DESIGN AND IMPLEMENTATION OF PULSE OXIMETER USING FUZZY LOGIC

MAJOR PROJECT REPORT SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY
In
Electronics and Communication Engineering

SUBMITTED BY
Richa Thakur
(2311254)
Vikram Singh
(2311258)
Mehak Sood
(2311259)

UNDER THE SUPERVISION OF

Chander Mohan
Co-Supervisor
Assistant Professor
ECE Department, ACE

Prof. Ashok Kumar
Supervisor
Head of the Department
ECE Department, ACE

SUBMITTED TO

Department of Electronics and Communication Engineering
Ambala College of Engineering and Applied Research, Ambala
Kurukshetra University, Kurukshetra.
2015

Department of Electronics and Communication Engineering
Ambala College of Engineering and Applied Research, Ambala
(*Affiliated With*)
Kurukshetra University, Kurukshetra – 136119
(Established by the state legislature Act XII of 1956)
("A" Grade, NAAC Accredited)

Ref No: ACE/ECE/15/ **Date:** 01/05/15

CERTIFICATE

Certified that this major project report entitled **"DESIGN AND IMPLEMENTATION OF PULSE OXIMETER USING FUZZY LOGIC"** is the bonafide work of "**Richa Thakur (2311254), Vikram Singh Anttal (2311258) and Mehak Sood (2311259)** of B.Tech 8th Semester, Electronics and Communication Engineering, Ambala College of Engineering and Applied Research, Devsthali, Ambala", who carried out the major project work under our supervision during Jan, 2015-April, 2015.

[redacted]

(Chander Mohan)
Assistant Professor
ECE Department, ACE

Supervisor

[redacted]

(Prof. Ashok Kumar)
Head of the Department
ECE Department, ACE

This is to certify that the above statement is correct to the best of our knowledge

[redacted]

(Vijay Kumar Anand)
Sr. Assistant Professor

[redacted]

(Sorabh Malhotra)
Assistant Professor
ECE Department, ACE

Head of Department

[redacted]

(Prof. Ashok Kumar)
ECE Department, ACE

CANDIDATES' DECLARATION

We, **Richa Thakur (2311254), Vikram Singh (2311258)** and **Mehak Sood (2311259)** hereby declare that the work which is being presented in the major project report entitled, **"DESIGN AND IMPLEMENTATION OF PULSE OXIMETER USING FUZZY LOGIC"** in partial fulfilment of requirements for the award of degree of B.Tech. (ECE) and submitted in the Department of Electronics and Communication Engineering, Ambala College of Engineering and Applied Research, Ambala, Kurukshetra University, Kurukshetra, is an authentic record of our own work carried by us under the supervision of Prof. Ashok Kumar (Head, ECE Department, Ambala College of Engineering and Applied Research) and Chander Mohan. The matter presented in this project report has not been submitted in this or any other University / Institute for the award of B.Tech Degree.

Signature of the Students *Dated*:

(Richa Thakur)

2311254

(***Vikram Singh Anttal)***

2311258

(Mehak Sood)

2311259

Department of Electronics and Communication Engineering

Ambala College of Engineering and Applied Research, Ambala

(*Affiliated With*)

Kurukshetra University, Kurukshetra – 136119

(Established by the state legislature Act XII of 1956)

("A" Grade, NAAC Accredited)

Ref No: ACE/ECE/14/ **Date:**

CERTIFICATE

Certified that this Major project report entitled **"DESIGN AND IMPLEMENTATION OF PULSE OXIMETER USING FUZZY LOGIC"** is the bonafide work of "**Richa Thakur (2311254), Vikram Singh Anttal (2311258) and Mehak Sood (2311259)** of B.Tech 8th Semester, Electronics and Communication Engineering, Ambala College of Engineering and Applied Research, Devsthali, Ambala", who carried out the major project work under our supervision during Jan, 2015-April, 2015.

Co-Supervisor	***Supervisor***
(Chander Mohan)	(Ashok Kumar)
Assistant Professor ECE Department, ACE	Head of the Department ECE Department,ACE

This is to certify that the above statement is correct to the best of our knowledge

Project Coordinators	***Head of Department***
(Vijay Kumar Anand) Sr. Assistant Professor	Prof. Ashok Kumar ECE Department, ACE
(Sorabh Malhotra) Assistant Professor ECE Department, ACE	

ACKNOWLEDGEMENTS

It is beyond our word power to acknowledge for the guidance to those who have cooperated and inspired us during the completion of this project. Still we would like to take this opportunity to express our sincere thanks to all those people who have, in various ways, helped in successful completion of this project. First of all we would like to thank almighty GOD who has given this wonderful gift of life to us. He is the one who is guiding us in right direction to follow noble path of humanity.

We must express our heartiest gratitude to respected **Dr. Ashawant Gupta**, Director, Ambala College of Engineering and Applied Research, Ambala for providing us the infrastructure to complete the project. We are thankful to respected **Dr. Amit Wason,** Principal, Ambala College of Engineering and Applied Research. We also thank **Prof. Ashok Kumar**, Dean Academics and Head, ECE Department, ACE for providing us all guidance to complete the project. It would be unfair if we do not mention the invaluable contribution and timely co-operation extended to us by Mr. Satpal Singh, Lab Supervisor, and ECE Department ACE.

We also give thanks to our parents who supported us in this pursuit. At the end we would like to thank our loving, caring and hardworking friends for their entertaining and knowledgeable discussions and incredible support in last four years, along with our science heroes without whose contribution probably science and technologies would not have reached here. Last but not the least we would like to express our utmost regards for the Electronics and Communication Engineering Department of our Institute.

(*Richa Thakur*)

2311254

(*Vikram Singh Anttal*)

2311258

(*Mehak Sood*)

2311259

ABSTRACT

The pulse oximeter is the most used device due to be a non-invasive method in measuring the oxygen level in blood. In the pulse oximeter, red and IR (Infrared) LEDs (light emitting diode) sent signals to the photodiode. An equation or relationship between the ratio (R) of signals (red and IR) received by photodiode with the oxygen saturation value (SpO2) needs to model the value of blood oxygen saturation. R (ratio) value is calculated according to absorption rates of signals, which are sent by red and IR LEDs in finger mounted oxygen saturation sensor. The obtained R value has been used as the input to fuzzy logic then SpO2 value has been calculated. Also, a linear relationship between SpO2 and R values has been created with absorption coefficients using linear regression method and compared with SpO2 values calculated by fuzzy logic method.

LIST OF FIGURES

TABLE

ABBREVIATIONS USED

AC	Alternating Current
DC	Direct Current
EAGLE	Easily Applicable Graphical Layout Editor
IC	Integrated Circuit
IR	Infrared
LED	Light Emitting Diode
LCD	Liquid Crystal Display
MATLAB	Matrix Laboratory
PCB	Printed Circuit Board
PIC	Programmable Interface Controller
SpO2	Saturation of Peripheral Oxygen
USB	Universal Serial Bus

Contents

Chapter 1

INTRODUCTION

1.1 Overview

Pulse oximeter is a medical instrument that can detect heart rate and oxygen saturation. Since it uses optical sensors so it is a non-invasive method used to measure the oxygen level in blood. The principal advantage of optical sensors for medical applications is that there is no electrical contact between the patient and the equipment. Pulse oximeter devices are generally used in the diagnosis of sleep disorders and respiratory diseases such as pneumonia, asthma, pulmonary diseases, chronic bronchitis, emphysema, congestive heart failure. Portable pulse oximeters are also useful for mountain climbers and athletes whose oxygen levels may decrease with high altitudes or with exercise.

The core theory behind the pulse oximeter is the variability of the absorption coefficient of photons going through human tissues at different wavelength. Since people are caring about the amount of oxygen saturation in our blood, the specific wavelength region should be settled which is the most sensitive to the oxygen in our blood. In our blood, oxygenated haemoglobin (oxy-Hb) and deoxygenated haemoglobin (deoxy-Hb), which can be used to measure human blood oxygen level, are stronger absorbers of light with wavelength in the range of 650nm-1000nm. In the pulse oximeter, red and IR (Infrared) LEDs (light emitting diode) having wavelengths around 660nm and 940 mm [1] respectively are used to send signals to the photodiode. The photodiode on the other side measures the intensity of transmitted light at each wavelength from which oxygen saturation is derived. It may be noted that the oxy –Hb absorbs both the lights differently. The oxy-Hb absorbs more IR light than red light, whereas the deoxy-Hb absorbs more red light than IR light. The pulse oximeter directly senses the absorption of red and infra red light and the ratio of pulsatile to nonpulsatile light at the red and infrared wavelengths are translated. These signals fluctuate in time because the amount of arterial blood that is present increases and decreases with each heartbeat. A microprocessor/microcontroller integrates the data, and through an elaborate calibration algorithm based on human volunteer data, the oxygen saturation can be estimated. An equation or relationship between the ratio (R) of signals (red and IR) received by photodiode with the oxygen saturation value

(SpO2) needs to model the value of blood oxygen saturation. Pulse oximeter is thus based upon two physical principles:

- The light absorbance of oxygenated haemoglobin is different from that of reduced haemoglobin, at the oximeter's two wavelengths, which include red and near infrared light.
- The absorbance of both wavelengths has a pulsatile component, which is due to the fluctuations in the volume of arterial blood between the source and the detector.

In this project report, all the related work along with the gaps in the study, problems formulation and the objectives to go with this project, the methodology opted for the project and its applications has been mentioned in chapter 2. Chapter 3 gives the overview about various components that have been used in the project also it mentions the various tools required for designing the PCB. In chapter 4 system architecture has been described. The architecture comprises of inter-relationship among different modules or functional units of the project. Chapter 5 describes the system design and analysis which includes the specification, mathematical model and step by step implementation of the system. Results and discussions have been mentioned in chapter 6. Chapter 7 includes the conclusions drawn and the future scope of the project.

1.2 Background of the Project

The central idea which motivated the design of project is to provide the user a low cost device that could be worn on wrist instead of conventional finger pulse oximeter. The designed oximeter should be reliable, comfortable to wear, easy to configure and operate, highly responsive and accurate. Hence the above mentioned characteristics lead to design a project that has the capability to provide the solution for the same i.e.

“Design and implementation of pulse oximeter to measure oxygen saturation based on fuzzy logic.”

1.3 Functional Block Diagram

The inter-relationship among different modules or functional units of the project comprises its architecture. The block diagram shown below illustrates all the detailed registers, units modules involved in the project design along with the control and data transfer. The Function Block Diagram (FBD) is a graphical language for programmable logic controller design that can describe the function between input variables and output variables.

The designed pulse oximeter comprises of two main parts- hardware and software part. In hardware part, the received signal from SpO2 sensor is processed with electronic circuits (current to voltage converter, filters, op-amps, and ADC). In software part, the fuzzy logic method is used for calculating oxygen saturation value. The step by step control transfer from one module to another and the corresponding functioning of the system design is as shown in the figure below:

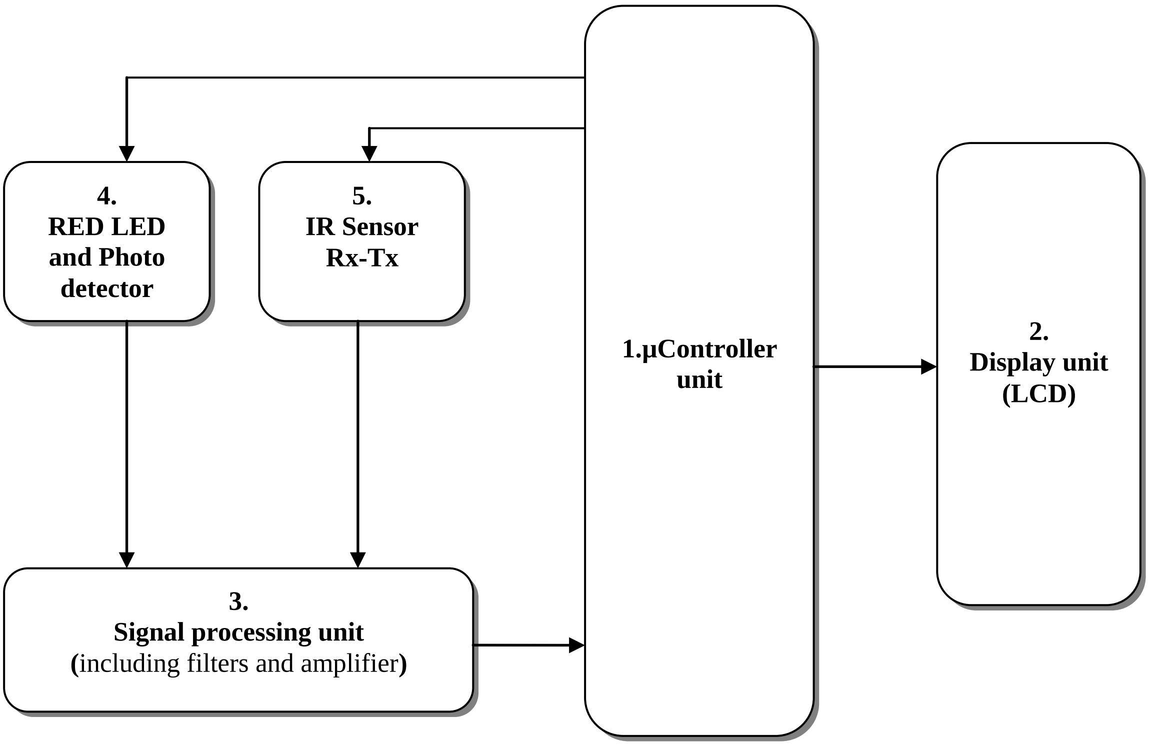


Figure 1.1 Block Diagram of the Project

1.4 Scope of the Project

The measurement of oxygen saturation level is required in various fields like medical, aviation, hill climbing etc. The designed pulse oximeter is useful in a situation where a patient's oxygenation is unstable including intensive care, operating, recovery and emergency, pilots in unpressurized aircraft, mountain climbers and athletes - for determining the effectiveness of or need for supplement oxygen. It is easy to configure and operate.

1.5 Applications of the Project

The various applications of pulse oximeter are as follows:

- **Medical applications**

  Pulse oximeter is generally used in the diagnosis of sleep disorders and respiratory diseases such as pneumonia, asthma, pulmonary diseases, chronic bronchitis, emphysema, congestive heart failure. People with cardiac or pulmonary disease may check on their blood oxygenation continuously, especially while jogging or exercising. Also before giving anaesthesia to a surgical patient the oxygen saturation level is checked.
- **Aviation applications**

  Pilots use them to determine if they need supplemental oxygen, especially when they fly a non-pressurized airplane.
- **Mountain climbing applications**

  When climbing on high altitudes, alpinists use pulse oximeters as well in order to check the oxygen saturation at different altitudes. Since the level of oxygen at higher altitudes is relatively less.

1.6 Advantages of the Project

The designed oximeter is more comfortable as compared to conventional finger or ear worn oximeters. Since usually, the sensor would press on a finger or the ear lobe which would sometimes cause discomfort or pain as the sensor would block the blood flow. Also the fuzzy logic used in the project provides more realistic results as compared to the oximeters designed earlier.

1.7 Limitations of the Project

Despite of various advantages the designed pulse oximeter has several limitations also which have been mentioned as follows:

- **Error due to presence of carboxy-haemoglobin or met-haemoglobin**

  The accuracy of pulse oximetry is excellent when the oxygen saturation is between the range of 70% to 100%, provided the only haemoglobin species present in the blood are deoxygenated haemoglobin and oxygenated haemoglobin. If carboxy-haemoglobin or met-haemoglobin is present in appreciable amounts, the accuracy is suspected. Since COHb and MetHb also absorb light at the pulse oximeter's two wavelengths and this leads to error in estimating the percentages of reduced and oxy-haemoglobin.

- **Loss of accuracy at low values**

  Measurement of SpO2 is less accurate at low values, and 70% saturation is generally taken as the lowest accurate reading.

- **Electrical interference**

  Electrical interference from an electrosurgical unit can cause the oximeter to give an incorrect pulse count (usually by counting extra beats) or to falsely register decrease in oxygen saturation. This problem may be increased in patients with weak pulse signals.

- **Erratic performance with irregular rhythms**

  Irregular heart rhythms can cause erratic performance. During aortic balloon pulsation, the augmentation of diastolic pressure exceeds that of systolic pressure. This leads to a double or triple-packed arterial pressure waveform that confuses the pulse oximeter so that it may not provide a reading.

1.8 Conclusion

This chapter gives the overview of the project. It mentions the problems that lead to the initiation of the project and the various prerequisites that are required. The main functional blocks of the project are introduced and a brief introduction is given. The scope of the project and advantages are discussed so as to highlight the utility of the project. The advantages and limitations of project are also mentioned.

Chapter 2

LITERATURE SURVEY AND PROBLEM OUTLINE

This chapter includes the literature survey providing the past work that has been done on the same platform and the gaps in the study. It also describes the problems that lead to design this project as well as the objectives and the methodology used to achieve the goal.

2.1 Related Work

Anan Wongjan et al. [2] proposed a Continuous Measurements of ECG and SpO2 for Cardiology Information System for the measurement of Spo2 and heart rate using electrocardiography (ECG) technique .This type of technique involves the use of two electrodes which has to be attached with the patient's body for the measurement of the heart rate and oxygen saturation. The output signal received from the type of signal is very weak because of the size of the electrodes and the large distance between the two electrodes but it has the advantage of easy of operation and can be attached in any part of body for taking output.

Collin Schreiner et al. [3] designed a Blood Oxygen Level Measurement with a chest-based Pulse Oximetry Prototype System which is capable of measuring the heart rate, temperature and oxygen saturation. It is based on the reflectance based pulse oximetry which requires a much more power than the transmittance based pulse oximetry as the amount of power reflected is much more less than that of power transmitted. It can be preferred over transmittance because poor blood circulation in the measurement area may also corrupt blood saturation readings. In some life-threatening situations blood circulation is limited to the torso and head only.

Radovan Stojanovic et al.[4] published Design of an Oximeter Based on LED-LED Configuration and FPGA Technology. This system is realized using digital filters and digital signal processing. Such type of systems are genuinely very accurate because proposed sensing technique is fully digital and provides good spectral sensitivity, increased resolution, satisfactory signal to noise ratio, low power consumption, in system flexibility, reconfigurability and miniaturization but the cost of system is very high since it involves FPGA.

Wei Chen et al. [5] gave Non-invasive Blood Oxygen Saturation Monitoring for Neonates Using Reflectance Pulse Oximeter. In this project Near Infrared Spectroscopy (NIRS) techniques are applied for enhancing the flexibility of measurements at different locations on the body of the neonates but has a disadvantage of high power consumption due to continuous glowing of light emitting diodes and the system is more prone to supply variations because the output of the sensor depends upon the intensity of the LEDs' light which is a function of supply voltage. The other factors which need improvement are power factor, advanced interaction and feedback control.

Grantham Pang et al. [6] proposed A Neo-Reflective Wrist Pulse Oximeter. It is a wrist worn oximeter with neo-reflective sensor module designed to send the sensor data to a nearby intelligent mobile phone using wireless data transmission. A concave structure is designed to house the sensor module components so as to obtain improved operational efficiency when collecting the wrist pulse signals. As compared to the conventional finger worn oximeters it is more costlier and also more hardware is required. During the calibration process, it is found that a second-order relationship between the R value and the oxygen saturation can provide better results with the data.

Jianchu Yao et al. [7] designed In- House pulse oximeter. In this project, two light-emitting diodes are used as excitation sources, acquire reflectance data with a photodiode, and send these raw photo-plethysmographic data to a personal computer via an RS-232 serial link. A Lab VIEW interface running on the personal computer processes these raw data and stores the results to a file. It allow a home monitoring system to authenticate the identity of a patient prior to uploading the patient's physiological data to a remote electronic patient record.

Sangeeta Bagha et al. [8] published A Real Time Analysis of PPG Signal for Measurement of SpO2 and Pulse Rate in International Journal of Computer Applications (0975 – 8887) Volume 36– No.11, December 2011. The peak detection of the PPG signal is done by advanced peak detector in Lab View. First a threshold value is set by analyzing the PPG signal. The threshold value should be above the amplitude of small peak of the PPG signal and below the true peak of the PPG signal.

The numbers of identified peaks in each 60-second recording is counted and it provides a measure of the pulse rate.

An electrocardiography (ECG) technique for measuring the spo2, ECG and Heart rate using the two electrodes has been presented in [2] which lack precision due to losses in ohmic contacts in electrode and the design of hardware is difficult due the placement of the electrode at precise distance from heart to the targeted part. Collin Schreiner et al. used the reflectance based pulse Oximetry technique in [3] which was highly uneconomic on the part of power requirement. FPGA is the other device which can be used as a high speed digital control unit dedicated to specified job of processing the signal and output display management as was done in [4] but genuinely such a devices are very expensive for such a requirements. In [5] a technique using the single infrared led based on Near Infrared Spectroscopy (NIRS) techniques is used for measuring the oxygen saturation but it lacks on the part of removing the static losses like flesh and the liquid absorption as well as the power required for higher wavelength LEDS like white or Ir is comparatively high .A Neo-Reflective Wrist Pulse Oximeter was given [6] which can send the data to nearby an intelligent mobile. In this the hardware output can be improved using a higher order filter. Similar to [4] another expensive hardware based technique was proposed by [7] using labview which along with the spo2 and heartrate also provides the other physical characteristics like temperature, humidity etc. This data is logged to the computer using max 232 ic and is stored in the database for future inspection. A Real Time Analysis of PPG Signal was done in [8] for measuring spo2 and heart rate but such a devices due to temperature sensitivity of electrode lack accuracy can deviate significantly.

2.2 Gaps in the Study

The analysis shows that every project has its own advantages and limitations. Some of the gaps in the study were as follows:

- The output signal received from the type of signal [2] is very weak because of the size of the electrodes and the large distance between the two electrodes.

- The reflectance based pulse oximetry [3] which requires a much more power than the transmittance based pulse oximetry as the amount of power reflected is much more less than that of power transmitted.
- The cost of the system [4] is very high due to the use of FPGA.
- Due to continuous glowing of light emitting diodes, the system [5] is more prone to supply variations because the output of the sensor depends upon the intensity of the LEDs' light which is a function of supply voltage. The other factors which need improvement are power factor, advanced interaction and feedback control.

2.3 Problem formulation

The limitations of various projects, discussed in the literature survey, are taken into consideration while designing of the project. Instead of using conventional methods for calibration, fuzzy logic has been implemented. The problems and the discomfort that occurred due to the finger worn pulse oximeters have been removed by using wrist worn oximeter. As we seen that [2] and [8] lack precision due to losses in ohmic contacts in electrode and the design of hardware is difficult due the placement of the electrode so it can be improved using pulse Oximetry rather than electrocardiography (ECG) technique on the other hand [3] and [7] which was highly uneconomic on the part of power requirement and due to FPGA's cost which can be reduced significantly by using microcontroller which is very economic. The model presented in better as it requires only one led but is much more undesirable due to inaccuracy of output due to static losses and high power losses which can be overcome by using the multiple leds for rejecting these unwanted losses in flesh and liquids.AS the bandwidth required for this system is 3 Hz which is very small so multiple ordered signal processing filter can be used to improve the output which was a limitation of the project given in [6].

2.4 Objectives of the Project

The main objectives of our project are as follows:

- To study literature survey.
- To study the microcontroller PIC 16F877A and its various interfacings.
- To simulate the circuit using PROTEUS.

- To implement the program using fuzzy logic.
- To design PCB using EAGLE.
- To synchronize between various modules.
- To design and develop the pulse oximeter.

2.5 Methodology

This section explains the methodology we used in implementation of our project. The main phases of our project were:

- **Requirement Analysis**

  In this phase, we discussed the various modules to be used and the basic model of our project.
- **Literature Survey**

  We studied the other similar projects that were implemented earlier. We tried to find out the gaps between them and their drawbacks, so that we can make a better design.
- **Investigation of Possible Implementation Methods**

  The third phase that we underwent was to come up with as many possible methods for implementation of our project that we could. None of the ideas were immediately rejected. A seemingly unfeasible idea may not be able to stand alone but it needs to be considered because certain parts of it may be applicable to another idea. To narrow down our choices we needed to find out which ones were the most feasible.
- **Circuit Simulation**

  Before designing the PCB, the circuit simulation was done with the help of PROTEUS. It is a virtual system modelling that provides animated components and microprocessor models to simulate the design.
- **PCB Designing and Testing**

  It was one of the most important phases of our project. In this, we designed the schematic and PCB layout using EAGLE. After designing the PCB, testing was done in order to check the connections as well as power supply and ground points.

- **Programming**

  In order to make the controller work according to the requirement, a program is needed. The programming was done using HI-TECH C COMPILER and it was burnt into the controller using PIC kit 2 and PIC burner.

2.6 Conclusion

In this chapter related work has been mentioned. It basically describes the various technologies used by the others in making similar projects. It also mentions the shortcomings and the gaps in their study. The objectives and the methodology used in designing the system have been also mentioned.

Chapter 3

COMPONENTS AND TOOLS USED

This chapter describes the various components that have been used in making the project. It also mentions the tools – EAGLE (For PCB Designing), MPLAB IDE and HI-Tech C Compiler, ISIS 7 Professional PROTEUS and MATLAB 2012b that have been used for designing the project.

3.1 Components Used

3.1.1Capacitor

A capacitor (originally known as a condenser) is a passive two terminal electrical component used to store energy electrostatically in an electric field. When there is a potential difference across the conductors (e.g., when a capacitor is attached across a battery), an electric field develops across the dielectric, causing positive charge +Q to collect on one plate and negative charge −Q to collect on the other plate.

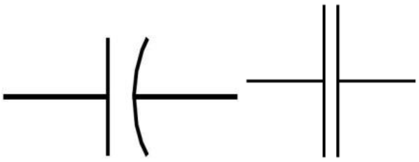


Figure 3.1: Capacitor Symbols [9]

3.1.2 Diodes

1N4007 is a member of 1N400x diodes.Diode is a rectifying device which conducts only if a certain threshold voltage or cut-in voltage is present in the forward direction (a state in which the diode is said to be forward-biased) and it behaves open circuited for the current flow from cathode to anode (a state in which diode is said to be reverse-biased).

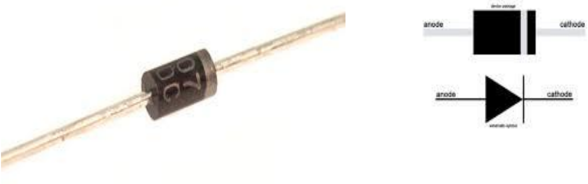


Figure 3.2: Diodes [10]

3.1.3 Voltage Regulator

The 7805 is a family of self- contained fixed voltage regulator integrated circuits. The 7805 is commonly used in electronic circuits to provide a regulated power supply. The 7805 line is positive voltage regulators as they produce a voltage that is positive w.r.to ground.

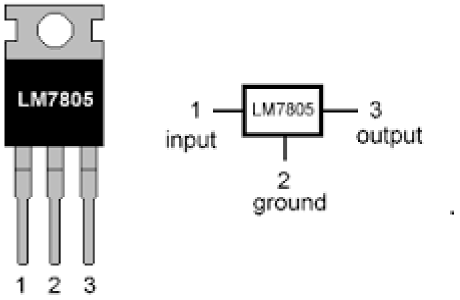


Figure 3.3: Voltage Regulator 7805[11]

3.1.4 Step Down Transformer

A transformer is an electrical device that transfers energy between two or more circuits through electromagnetic induction. A varying current in the transformer's primary winding creates a varying magnetic flux in the core and a varying magnetic field impinging on the secondary winding. This varying magnetic field at the secondary induces a varying electromotive force (emf) or voltage in the secondary winding. Making use of Faraday's Law in conjunction with high magnetic permeability core properties, transformers can thus be designed to efficiently change AC voltages from one voltage level to another within power networks.

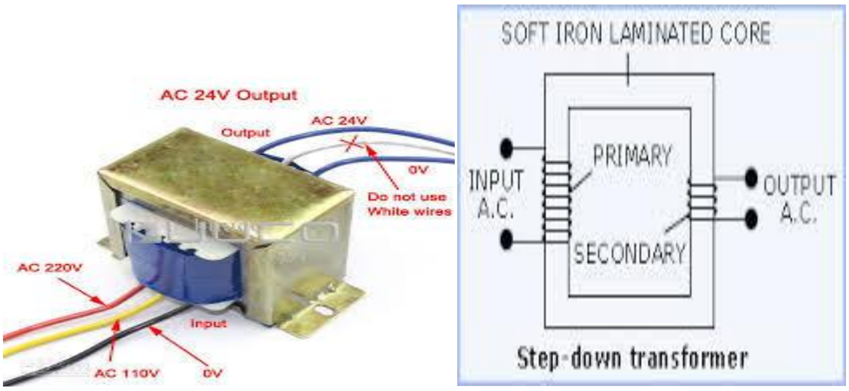


Figure 3.4: Step down Transformer [12]

3.1.5 Infra-Red Sensor

An infrared sensor is an electronic instrument which is used to sense certain characteristics of its surroundings by either emitting and/or detecting infrared radiation. All objects which have a temperature greater than absolute zero (0 Kelvin) possess thermal energy and are sources of infrared radiation as a result.Sources of infrared radiation include blackbody radiators, tungsten lamps and silicon carbide. Infrared sensors typically use infrared lasers and LEDs with specific infrared wavelengths as sources.

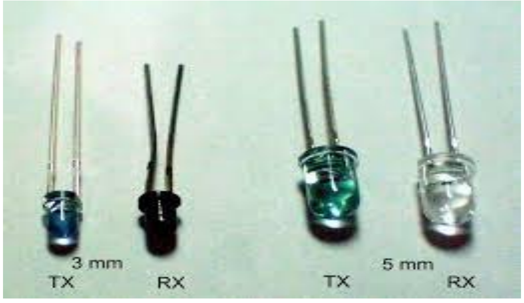


Figure 3.5: IR Sensor [13]

3.1.6 Comparator LM358

LM358 consists of two independent, high gain operational amplifiers in one package. Important feature of this IC is that we do not require independent power supply for working of each comparator for wide range of power supply. LM358 can be used as transducer amplifier, DC gain block etc. It has large dc voltage gain of 100dB. This IC can be operated on wide range of power supply from 3V to 32V for single power supply or from ±1.5V to ±16V for dual power supply and it also support large output voltage swing.

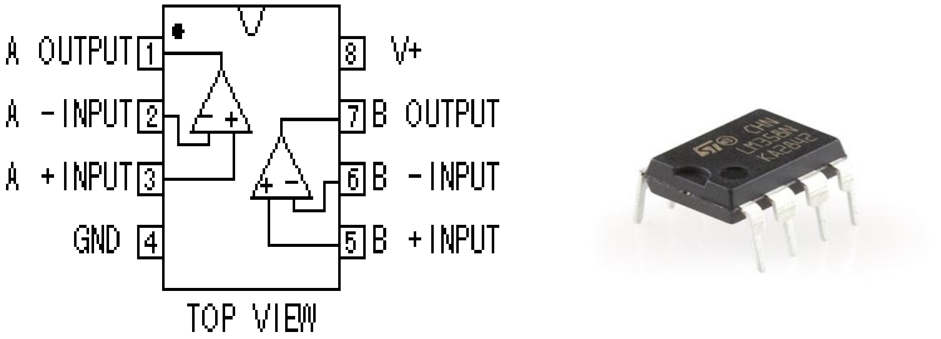


Figure 3.6: Comparator LM 358 [14]

3.1.7 Photodiode

Photodiodes convert light into current or voltage depending on its mode of operation. It is a PN junction or PIN structure. When a photon of sufficient energy strikes the diode it creates a free electron and a hole. Now the holes move towards the anode and electrons towards the cathode and a photocurrent is created.

Figure 3.1.7: Photo diode QSD2030F [15]

3.1.8 Liquid Crystal Display JHD 162A

The JHD162A display controller is a monochrome LCD module. The JHD162A has 16 pins and can be operated in 4-bit mode or 8-bit mode. Here we are using the LCD module in 4-bit mode. Before going in to the details of the project, let’s have a look at the JHD162A LCD module. The schematic of a JHD162A LCD module is given below.

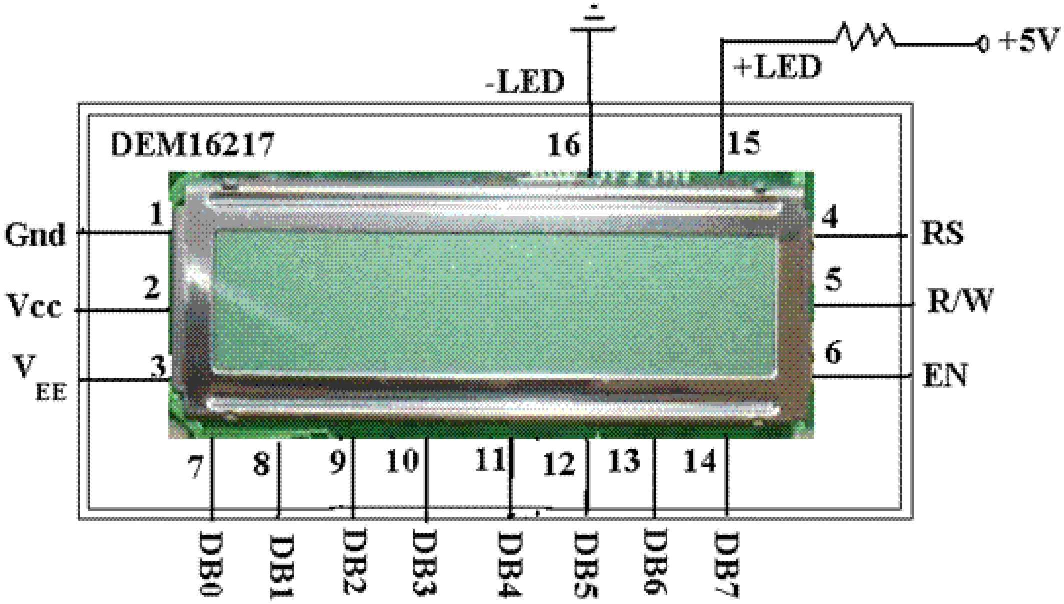


Figure3.8: LCD JHD 162A [16]

3.1.9 PIC 16F877A

The PIC16F877A CMOS FLASH-based 8-bit microcontroller is upward compatible with the PIC16C5x, PIC12Cxxx and PIC16C7x devices. It features 200 ns instruction execution, 256 bytes of EEPROM data memory, self programming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, a synchronous serial port that can be configured as either 3-wire SPI or 2-wire I2C bus, a USART, and a Parallel Slave Port.

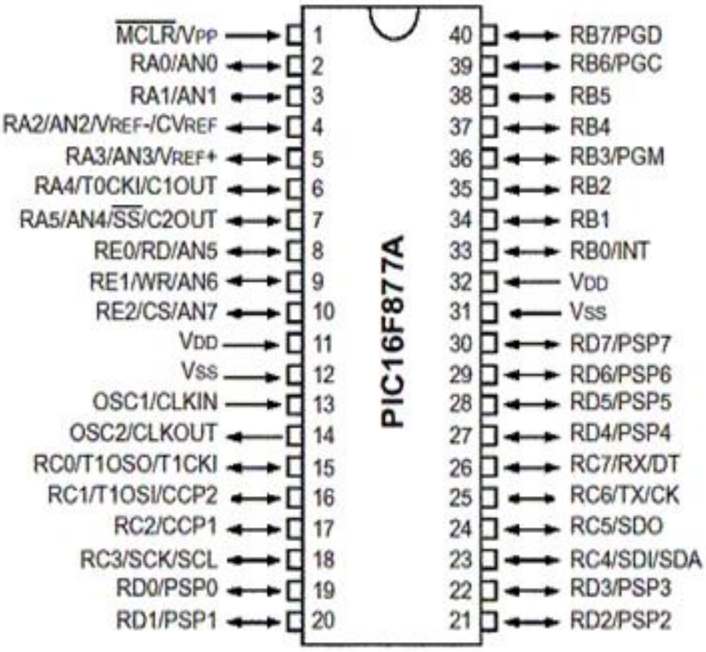


Figure 3.9: Pin Diagram of PIC 16F877A [17]

3.2 Tools Used

3.2.1 EAGLE

EAGLE stands for Easily Applicable Graphical Layout Editor. It is software [18] used to design an electronic schematic and layout of a printed circuit board. It consists of a schematics editor, a PCB editor and an auto router module. Schematic provides the functional flow and the graphical representation of an electronic circuit. Schematic mainly consists of Electrical connections (nets), Junctions, Integrated circuits symbols, discrete components like Resistors, I/O connectors, Power &ground symbols.PCB layout. To start laying out the printed circuit board, open the schematics in Eagles schematic editor and click on the board button.

3.2.2 MPLAB IDE V 8.85

MPLAB IDE [19] is a software program that runs on a PC (Windows, Mac OS, Linux) to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment (IDE), because it provides a single integrated environment to develop code for embedded microcontrollers. MPLAB IDE Integrated Development Environment brings many changes to the PIC microcontroller development tool chain. Unlike previous versions of the MPLAB IDE which were developed completely in-house, MPLAB IDE is based on the open source Net Beans IDE from Oracle. Taking this path has allowed us to add many frequently requested features very quickly and easily, while also providing us with a much more extensible architecture to bring you even more new features in the future.

3.2.3 HI-TECH C Compiler

HI-TECH C [20] is a world class brand of compilers featuring omniscient code generation whole program compilation technology designed for Microchip Technology's 8, 16 and 32-bit PIC microcontroller and PIC digital signal controller architecture. It is compatible with all microchip debuggers and emulators.

3.2.4 ISIS 7 Professional PROTEUS

Proteus 7.0 is a Virtual System Modelling (VSM) [21] that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and if attached to the PC, switches and buttons Proteus VSM comes with extensive debugging features, including breakpoints, single stepping and variable display for a neat design prior to hardware prototyping.

3.2.5 MATLAB 2012b

MATLAB (matrix laboratory) 2012b is a [22] multi-paradigm numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

3.3 Conclusion

This chapter gives the overview about various components that have been used in the project. Also it mentions the various tools required for designing the PCB-EAGLE, HI TECH Compiler, PROTEUS and implementing fuzzy logic-MATLAB.

Chapter 4

SYSTEM ARCHITECTURE

The architecture shows the flow of information among different sections of the projects. It describes the outlook of the project and the functional blocks of the system.

4.1 Project Description

The following are the important points regarding the project:

- The pulse oximeter is designed to calculate the oxygen saturation in the human blood and the heart rate.
- The complete project has been designed in the parts: pulse oximeter sensor, control unit and display unit.
- Pulse Oximeter sensor involves the three major parts, one is light emitters, second is photo receptors and the third is the instrumentation and signal processing unit.
- Light emitter includes two LEDs- one is red light emitter and other is IR light emitter whose intensity and function is controllable by the control unit. The intensity of reflected light varies from person to person.
- Photo receptors respond to the emitted light. It includes two photo sensitive devices one is photo diode and the other is Infrared receiver or detector.
- The output of the receivers goes to the signal processing unit which involves the trans-impedance circuit, band pass filter and an amplifier of gain 10000. This unit consists of a fourth order band pass filter with two stage amplifier.
- Control unit (PIC 16F877A) is fed with the output of the pulse oximeter sensor at the analog port. The voltage at the port is root mean squared and averaged so as to obtain the pulsating and non-pulsating components.
- The heart rate input is applied at the timer input which is continuously observed for 15 seconds and hence the heart rate can be calculated for 60second or 1 minute after multiplication by four. The bigger the interval of observation better will the accuracy but at the cost of compromised speed.

- The ratio of the root mean square and average value is used for both red and infrared led for calculating the R factor which will conclude to the oxygen saturation value.
- This value finally will be displayed on the liquid crystal display JHD 162A. The LCD is configured with 8 bits in the write only mode.

A simple and the basic most diagram has been shown in the block diagram of the system in figure number 4.1

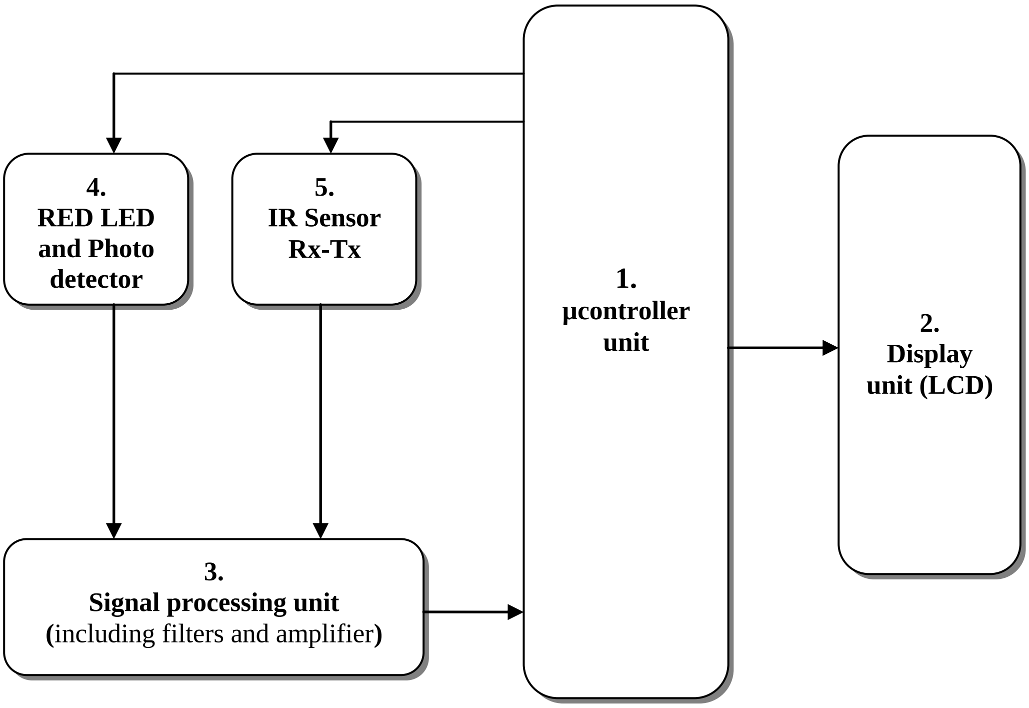


Figure 4.1: Block Diagram of the Project

4.2 Functional blocks of the Project

The inter-relationship among different modules or functional units of the project comprises its architecture. The block diagram shown below will illustrate all the detailed registers, units modules involved in the project design along with the control and data transfer The Function Block Diagram (FBD) is a graphical language for programmable logic controller design,[1] that can describe the function between input variables and output variables. A function is described as a set of elementary blocks. Input and output variables are connected to blocks by connection lines. Inputs and outputs of the blocks are wired together with connection lines, or links. The step by step control transfer from one module to another and the corresponding functioning of the system design is as shown in the figure below:

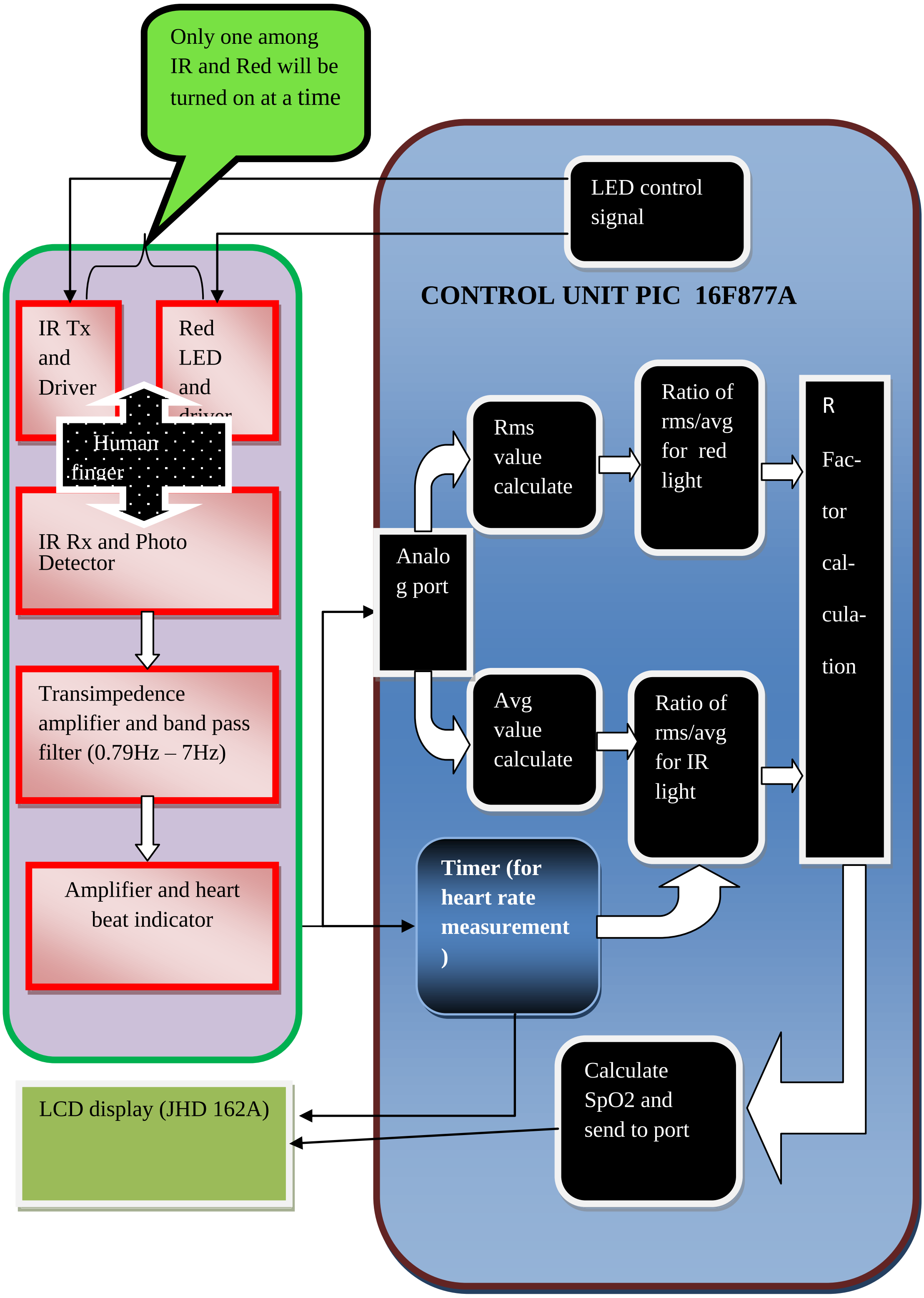

Only one among IR and Red will be turned on at a time
LED control signal
CONTROL UNIT PIC 16F877A
IR Tx and Driver
Red LED and driver
Human finger
IR Rx and Photo Detector
Transimpedence amplifier and band pass filter (0.79Hz – 7Hz)
Amplifier and heart beat indicator
Analog port
Rms value calculate
Avg value calculate
Ratio of rms/avg for red light
Ratio of rms/avg for IR light
R Fac-tor cal-cula-tion
Timer (for heart rate measurement)
Calculate SpO2 and send to port
LCD display (JHD 162A)

Figure 4.2: Architecture of the Project

4.3 Conclusion

This chapter mentions the generalised model of the project. The architecture describes the inter-relationship among different modules or functional units of the project. The step by step control transfer from one module to another and the corresponding functioning of the system design have been mentioned.

Chapter 5

SYSTEM DESIGN AND ANALYSIS

This chapter mentions the internal description of the project and the proposed mathematical model. It describes the workflow and step by step implementation of the project. It also mentions the specifications of the system.

5.1 Mathematical Modelling of System

A mathematical model is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed mathematical modelling. In general, mathematical models may include logical models. In many cases, the quality of a scientific field depends on how well the mathematical models developed on the theoretical side agree with results of repeatable experiments. Lack of agreement between theoretical mathematical models and experimental measurements often leads to important advances as better theories are developed.

5.1.1 Designing the signal processing circuit for the pulse Oximeter sensor

Response of a band pass filter:

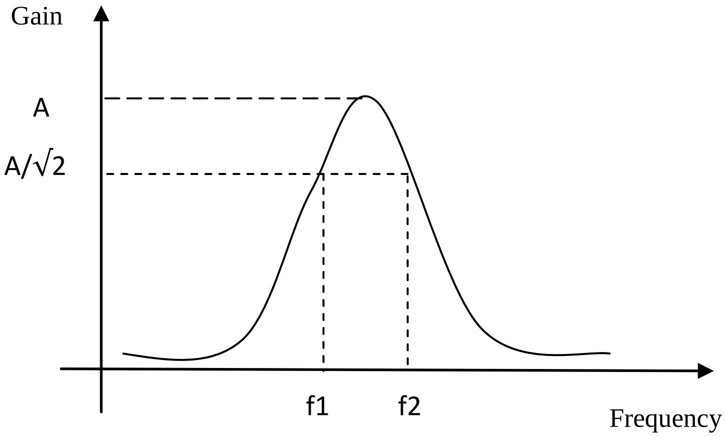


Figure 5.1: Magnitude Response of Simple Band Pass Filter

In order to realise a band pass filter, a low pass and a high pass filter is needed.
So a band pass is a second order filter. In this project two band pass filters have been used. So the order of the designed filter is four.
As the voltage the negative terminal will be given by equation (5.1):

$$V_n = \frac{R_F}{R_1+X_{c1}} = \frac{sR_1C_1}{1+sR_1C_1} * V_{IN} \qquad -(5.1)$$

$$Z_F = R_F||C_F = \frac{R_F}{1+sC_FR_F} \qquad -(5.2)$$

Hence the gain for non inverting amplifier can be calculated using equation 5.3:

$$\mathrm{Gain} = 1 + \frac{\mathrm{Z_F}}{\mathrm{R_0}} = \frac{\mathrm{R_0+sC_FR_FR_0+R_F}}{\mathrm{R_0(1+sC_FR_F)}} \qquad -(5.3)$$

$$\mathrm{Gain} = \left(1 + \frac{\mathrm{R_F}}{\mathrm{R_0}}\right)\left(\frac{1+{}^{(\mathrm{sC_FR_FR_0})}\!/_{(\mathrm{R_F+R_0})}}{1+\mathrm{sC_FR_F}}\right) \qquad -(5.4)$$

$$\mathrm{V_{OUT=}Gain * V_n} \qquad -(5.5)$$

$$\mathrm{V_{OUT=}} \left(1 + \frac{\mathrm{R_F}}{\mathrm{R_0}}\right)\left(\frac{1+{}^{(\mathrm{sC_FR_FR_0})}\!/_{(\mathrm{R_F+R_0})}}{1+\mathrm{sC_FR_F}}\right)\frac{\mathrm{sR_1C_1*V_{IN}}}{\mathrm{1+sR_1C_1}} \qquad -(5.6)$$

$$\mathrm{Transfer\ function} = \left(1 + \frac{\mathrm{R_F}}{\mathrm{R_0}}\right)\frac{\mathrm{sR_1C_1*(1+{}^{(sC_FR_FR_0)}\!/_{(R_F+R_0)})}}{\mathrm{(1+sC_FR_F)(1+sR_1C_1)}} \qquad -(5.7)$$

$$\mathrm{hence\ overall\ gain} = \left(1 + \frac{\mathrm{R_F}}{\mathrm{R_0}}\right) \qquad -(5.8)$$

$$\mathrm{H(s)} = \frac{\mathrm{s*}\left(\mathrm{s+\frac{(R_F+R_0)}{C_FR_FR_0}}\right)}{\left(\mathrm{s+\frac{1}{R_1C_1}}\right)\mathrm{(s+\frac{1}{C_FR_F})}} \qquad -(5.9)$$

This transfer function has two poles as understood from equation 5.9 and are calculated below:

$$s_{p1} = \frac{1}{R_1C_1} \qquad -(5.10)$$

$$s_{p2} = \frac{1}{R_FC_F} \qquad -(5.11)$$

which will give us cut of frequencies of the band pass filter as calculated below:

$$f_{c1} = \frac{1}{R_1C_1} \qquad -(5.12)$$

$$f_{c2} = \frac{1}{R_FC_F} \qquad -(5.13)$$

Chosen values of f_{c1} and f_{c2} using equation number 5.12 and 5.13 are .8 Hz and 3 Hz which can be obtained for following values of resistances and capacitances. As resistances has a wide range of values available for circuit design so will fix out capacitances first take

$$C_1 = 10\mu F$$

$$C_F = 1\mu F$$

hence the resulting values of resistances are:

$$R_1 = 19.895\ \text{k}\Omega \approx 20\text{k}\Omega$$

$$R_F = 53.05\text{k}\Omega \approx 56\text{k}\Omega$$

The two zeroes of the system lie at:

$$s_{z1} = 0$$

$$s_{z2} = \frac{(R_F+R_0)}{C_F R_F R_0} \qquad -(5.14)$$

The zeroes indicated above in equation 5.14 play a major part in the phase determination of a filter which is genuinely desired to be linear. As the multi stages will be used so for two identical stage system the transfer function become

$$H(s) = \left(\frac{s*\left(s+\frac{(R_F+R_0)}{C_F R_F R_0}\right)}{\left(s+\frac{1}{R_1 C_1}\right)\left(s+\frac{1}{C_F R_F}\right)}\right)^2 \qquad -(5.15)$$

For gain =100 the equation 5.15 which is the transfer function becomes

$$H(s) = \left(\frac{s*(s+1785.7)}{(s+5)(s+17.85)}\right)^2 \qquad -(5.16)$$

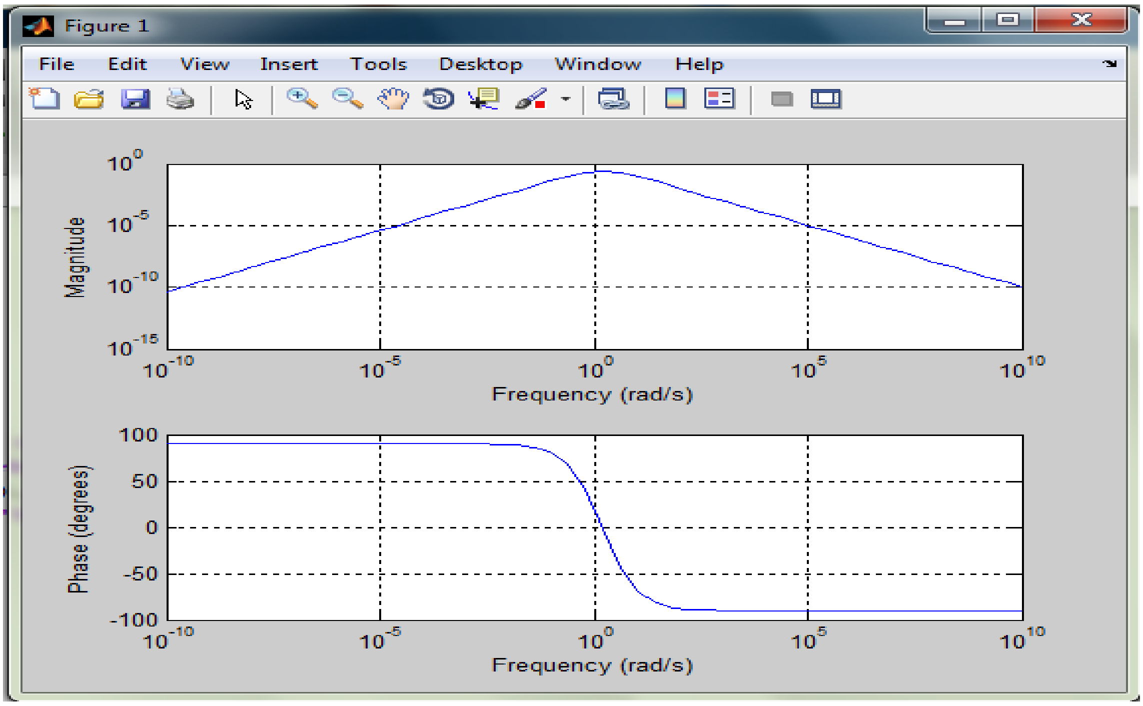


Figure 5.2: Frequency response of proposed second order band pass filter

Utilising final equation 5.16 (transfer function of the filter to be realized), the magnitude and the phase response of the system has been analysed as shown in fig.5.2. It has been plotted using the signal processing unit of MATLAB and it is a second order band pass filter.

5.1.2 Designing the digital system for calculating the oxygen saturation

Input received by the system will be analog which needs to be sampled. After sampling the analog input $V_A(t)$ will be converted to $V_A(nT)$ which is sampled at the rate of 1/T and finally is quantized to $V_A(n)$ which will be the value of our prime interest. As the received input has to wait for 15 seconds during which the pulse duration will be calculated denoted by T_P. This value will be used for the calculation of root mean squared and averaged values.

Number of samples will be denoted by $\mathrm{N}=\frac{\mathrm{T_P}}{\mathrm{T}}$ − (5.17)

Average value $\mathrm{V_{AVG}} = \sum_{\mathrm{I}=0}^{\mathrm{N}} \mathrm{V_A(i)}$ − (5.18)

Root mean square value $\mathrm{V_{RMS}} = \sum_{\mathrm{I}=0}^{\mathrm{N}} \mathrm{V_A}^2\mathrm{(i)}$ − (5.19)

Transfer function of the above stated systems:

$$\mathrm{H_{AVG}(n)}=\frac{1}{1+\mathrm{Z}^{-1}} \quad -(5.20)$$

But it’s not possible to calculate the transfer function of a root mean squarer as it’s a non linear device. But fortunately this can be realized practically.

$$\mathrm{R_I} = \frac{\mathrm{V_{RMS_I}}}{\mathrm{V_{AVG_I}}} \quad -(5.21)$$

The parameter I can be stated as

$$\mathrm{I} = \begin{cases} \mathrm{RED}, & I = 0 \\ \mathrm{IR}, & I = 1 \end{cases} \quad -(5.22)$$

Hence the R factor can be calculated as:

$$R = \frac{R_0}{R_1} \quad -(5.23)$$

Oxygen saturation: $\mathrm{SpO2} = \mathrm{A} + \mathrm{B} * \mathrm{R}$ − (5.24)

The two parameters A and B mentioned in equation 5.24 the oxygen saturation constants which are estimated using the surface of fuzzy logic. The diagram shown below can be used to realize the above stated filter which involves the use of root mean square value and average value calculation. The diagram shown below is an IIR realization of the system based upon above performed calculation with delay unit equal N.

Realisation of the system:

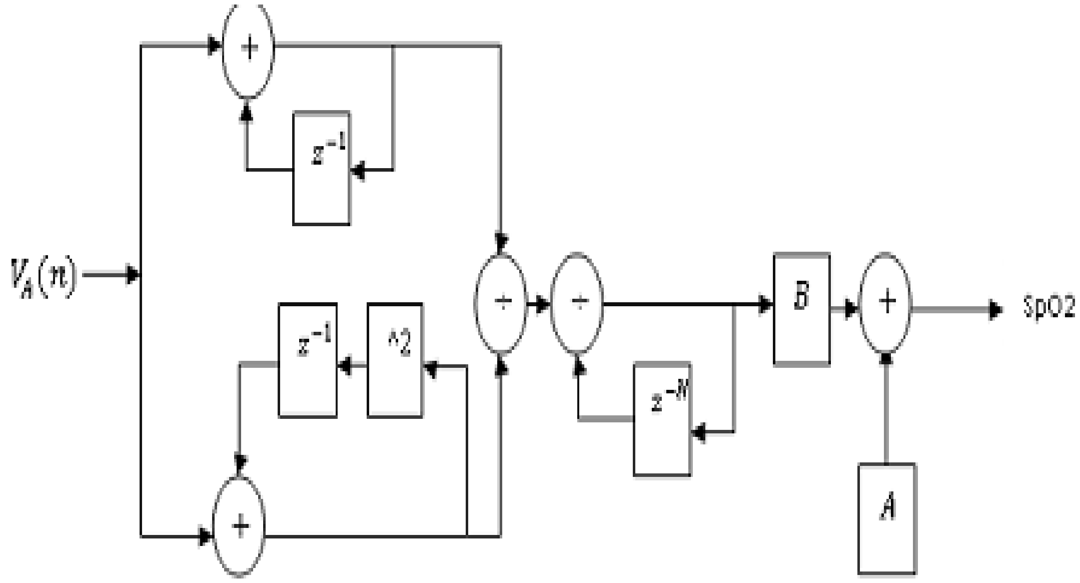


Figure 5.3: Realization of the Oxygen Saturation Measurement Digital System

5.2 Flowchart of the Project

A flowchart is a diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. A flowchart is described as "cross-functional" when the page is divided into different swim lanes describing the control of different organizational units. A symbol appearing in a particular "lane" is within the control of that organizational unit. This technique allows the author to locate the responsibility for performing an action or making a decision correctly, showing the responsibility of each organizational unit for different parts of a single process. Flowcharts depict certain aspects of processes and they are usually complemented by other types of diagram. For instance, Kaoru Ishikawa defined the flowchart as one of the seven basic tools of quality control, next to the histogram, Pareto chart, check sheet, control chart, cause-and-effect diagram, and the scatter diagram. Similarly, in UML, a standard concept-modeling notation used in software development, the activity diagram, which is a type of flowchart, is just one of many different diagram types. Average value The flowchart of the project is as shown below:

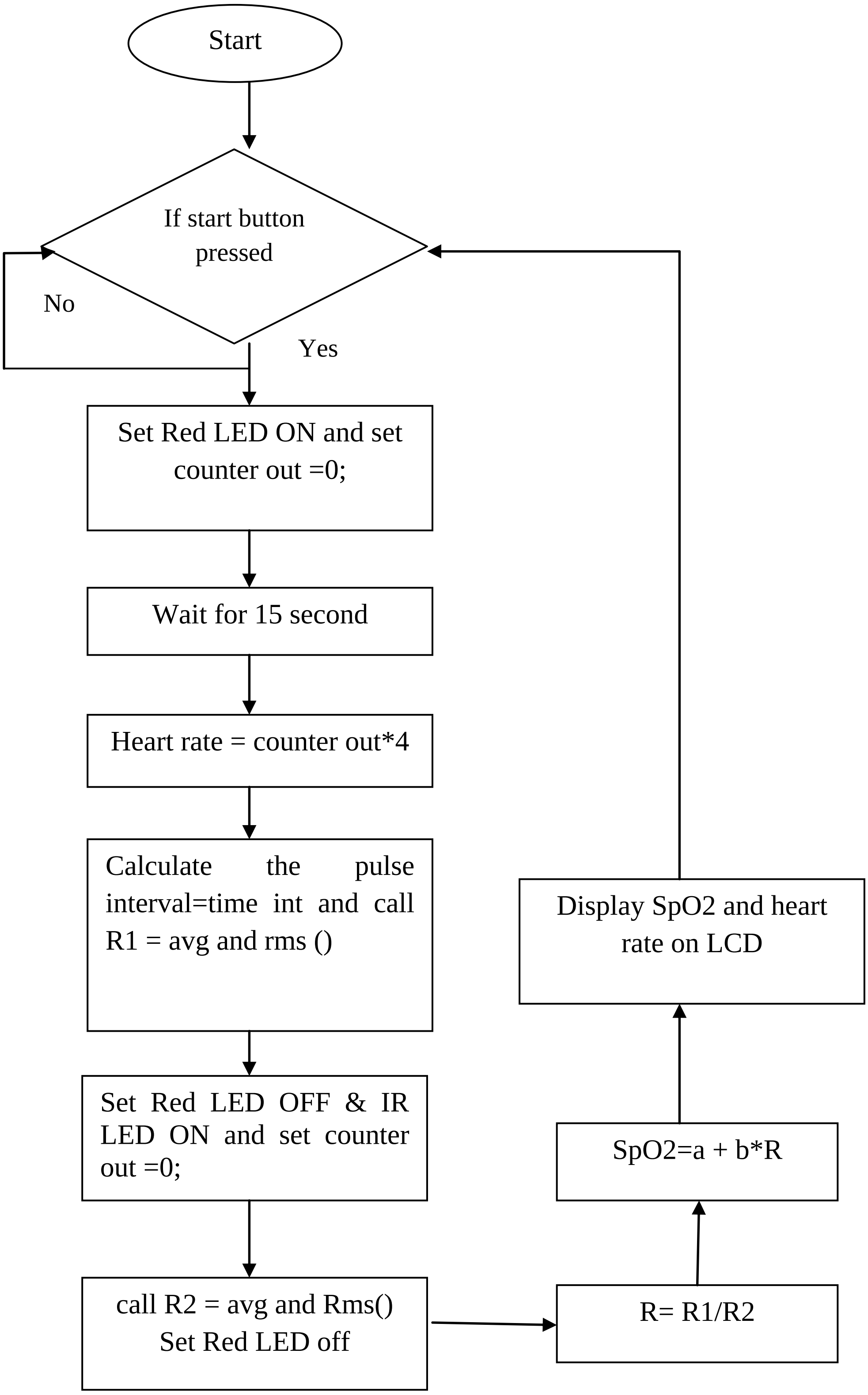

Start
If start button pressed
No
Yes
Set Red LED ON and set counter out =0;
Wait for 15 second
Heart rate = counter out*4
Calculate the pulse interval=time int and call R1 = avg and rms ()
Set Red LED OFF & IR LED ON and set counter out =0;
call R2 = avg and Rms()
Set Red LED off
R= R1/R2
SpO2=a + b*R
Display SpO2 and heart rate on LCD

Figure 5.4: Flowchart of the Project code (1)

The flow chart has been broken into two parts where first part involves description of the whole project and the second part is used just to find the ratio of the RMS to the average value which will be called by the first part wherever required. is calculated by taking the ratio of sum of all the samples and number of samples over a particular time interval whereas the RMS value is calculated by taking square root of mean of the squares of the samples taken. The ratio of average and RMS of Red LED to that of IR is calculated to find the R factor. The calculated R factor is returned to the first part where R factor has been used to compute the value of SpO2.

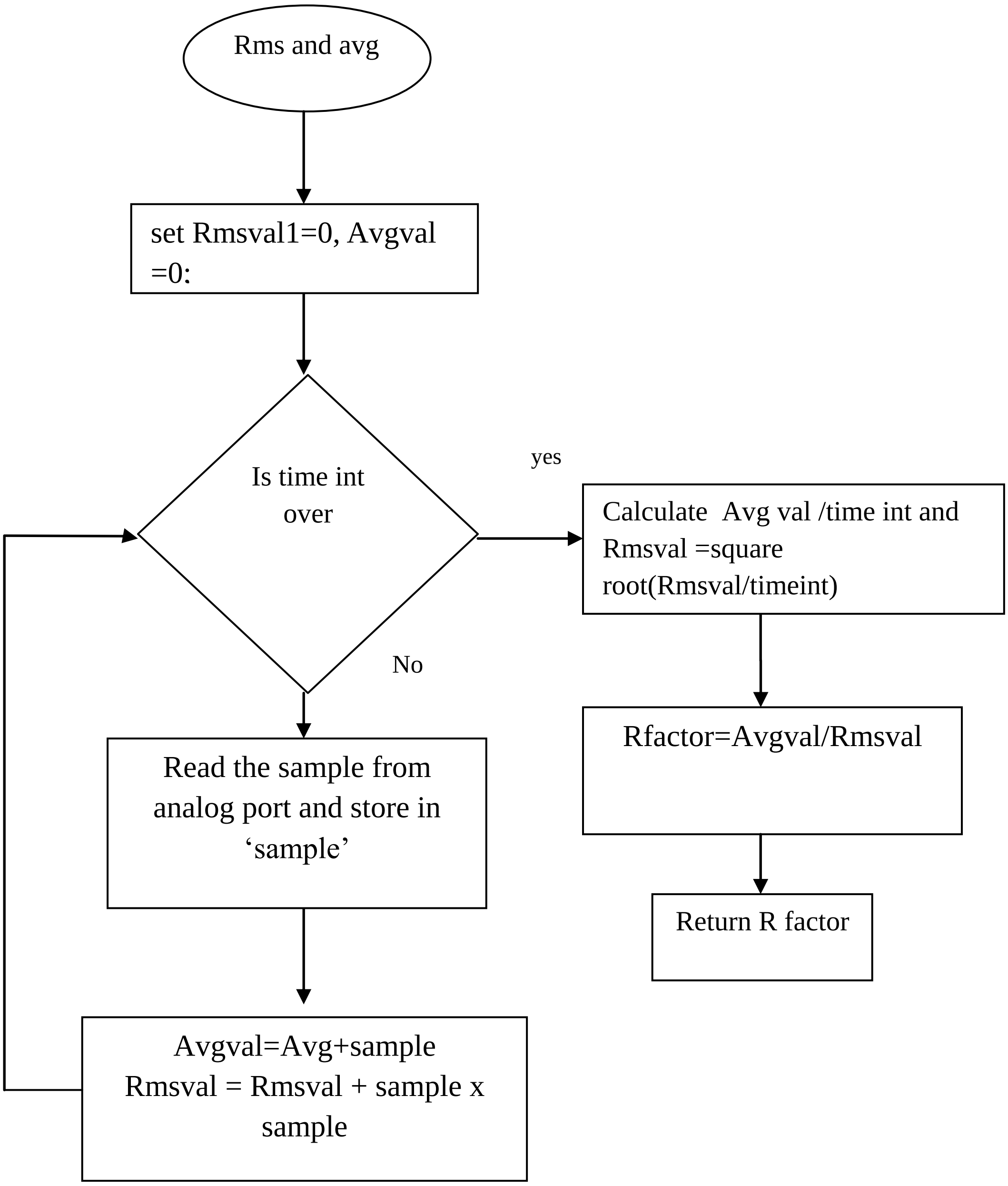

Rms and avg
set Rmsval1=0, Avgval =0:
Is time int over
yes
No
Calculate Avg val /time int and Rmsval =square root(Rmsval/timeint)
Rfactor=Avgval/Rmsval
Return R factor
Read the sample from analog port and store in 'sample'
Avgval=Avg+sample Rmsval = Rmsval + sample x sample

Figure 5.5: Flowchart of the Project code (2)

5.3 Algorithm of the Project

An algorithm is a step-by-step procedure for calculations. The algorithm of the project is as shown:

Algorithm used for oxygen saturation measurement SpO2 (R factor):

```
i)    Set the configuration bits timer and ADC registers
ii)   If start ==0 ; (start button is pressed).
      a). Redled_on=1; (set red LED on)
      b). Set count =0 ; initialize the counter
      c).call delay ; wait for 15 seconds
      d). Set heart rate =count * 4; (calculate heart beats in 1 minute or 60 seconds
      e). Calculate the pulse duration
      f). Call R1= rar(pulse duration) ;(calculate rms to average ratio for red light)
      g). set Redled_on = 0; and IRled_on = 1; ( turnoff red LED and switch on IR LED)
      h). Call R2= rar(pulse duration); (calculate rms to average ratio for IR light)
      i). Calculate SpO2=R1/R2;
iii)  Display heart rate and SpO2 on LCD.
iii)  Go to step iii
```

Algorithm for the calculation for RMS to Average ratio: R (pulse duration)

```
i)    Start
ii)   Set averagevalue=0 and rmsvalue=0;
iii)  Repeat until time <=pulse duration
          a)  Sample = an(0);(read samples from analog port pin 0;
          b). Calculate averagevalue=averagevalue + sample;
```

```
            c). Calculate rmsvalue=rmsvalue + sample*sample;
iv)     Set averagevalue=(averagevalue/pulse duration);
v)      Set rmsvalue=square root(rmsvalue/pulse duration);
vi)     R=rmsvalue/averagevalue;
vii)    Return R;
viii)   Stop
```

5.4 Conclusion

In this chapter the system design and analysis has been described. It also mentions the algorithm and flowchart of the project. The step by step control transfer from one module to another and the corresponding functioning of the system design have been mentioned.

Chapter 6

RESULTS AND DISCUSSIONS

The results and discussions of a project describe the software and hardware implementation. Before making the connections on the PCB, the similar connections have been made using animated components available on the software named PROTEUS.

6.1 Specifications of the system

The main specifications of the project have been mentioned in the table given below. The specifications of the system describe the operating voltage range, Clock Speed, response time, ultrasonic sensor range and memory specifications.

Table 6.1: Specifications of Project

SpO2	
Measurement range	70%-100%
Resolution	.1%
Measurement accuracy	Depends upon accuracy in heart rate
Heart rate	
Measurement range	30bpm – 200 bpm
Resolution	4
Measurement accuracy	Looses 6% for ± 1 bpm
Display	
Parameters	Heart rate,spO2
16 x 2 display with adjustable contrast.	
Mechanical Dimensions	90mm X 70 mm

Table 6.2: Patients' Demographics

Name	Class	Roll number	Spo2	Heartrate(bpm)
Sorab Datta	ECE 4^{TH}	2311242	97.2%	69
Ishana Banga	ECE 4^{TH}	2311237	96.8%	73
Jannat Bhola	ECE 4^{TH}	2311224	97.2%	78
Chander Mohan	Lecturer	Lecturer	96.3%	78
Puneet Sharma	ECE 4^{TH}	2311246	95.4%	74
Veer Singh	ECE 4^{TH}	2311244	99.0%	72
Sahil Kalra	ECE 4^{TH}	2311238	98.2%	72
Himanshu Dhiman	ECE 4^{TH}	2311255	98.6%	73

Healthy individuals at sea level usually exhibit oxygen saturation values between 96% and 99%, and should be above 94%. At 5,280 feet altitude (one mile high) oxygen saturation should be above 92%.An SaO_2 (arterial oxygen saturation) value below 90% causes hypoxemia (which can also be caused by anemia). Hypoxemia due to low SaO2 is indicated by cyanosis, but oxygen saturation does not directly reflect tissue oxygenation. The affinity of hemoglobin to oxygen may impair or enhance oxygen release at the tissue level. Oxygen is more readily released to the tissues (i.e., hemoglobin has a lower affinity for oxygen) when pH is decreased, body temperature is increased, arterial partial pressure of carbon dioxide ($PaCO_2$) is increased, and 2,3-DPG levels (a byproduct of glucose metabolism also found in stored blood products) are increased. When the hemoglobin has greater affinity for oxygen, less is available to the tissues. Conditions such as increased pH, decreased temperature, decreased $PaCO_2$, and decreased 2,3-DPG will increase oxygen binding to the hemoglobin and limit its release to the tissue.

Table 6.3: Effects of decreased oxygen saturation

Spo2	Effects
85% and above	No evidence of impairment
65% and less	Impaired mental function on average
55% and less	Loss of consciousness on average

6.2 Interfacings Used and Testing of the Project

6.2.1 Analog Port Interfacing and Testing

The output of the pulse Oximeter sensor will be analog which will be fed to the analog port of pic microcontroller. A snapshot of proteus simulated adc module is as shown in the diagram

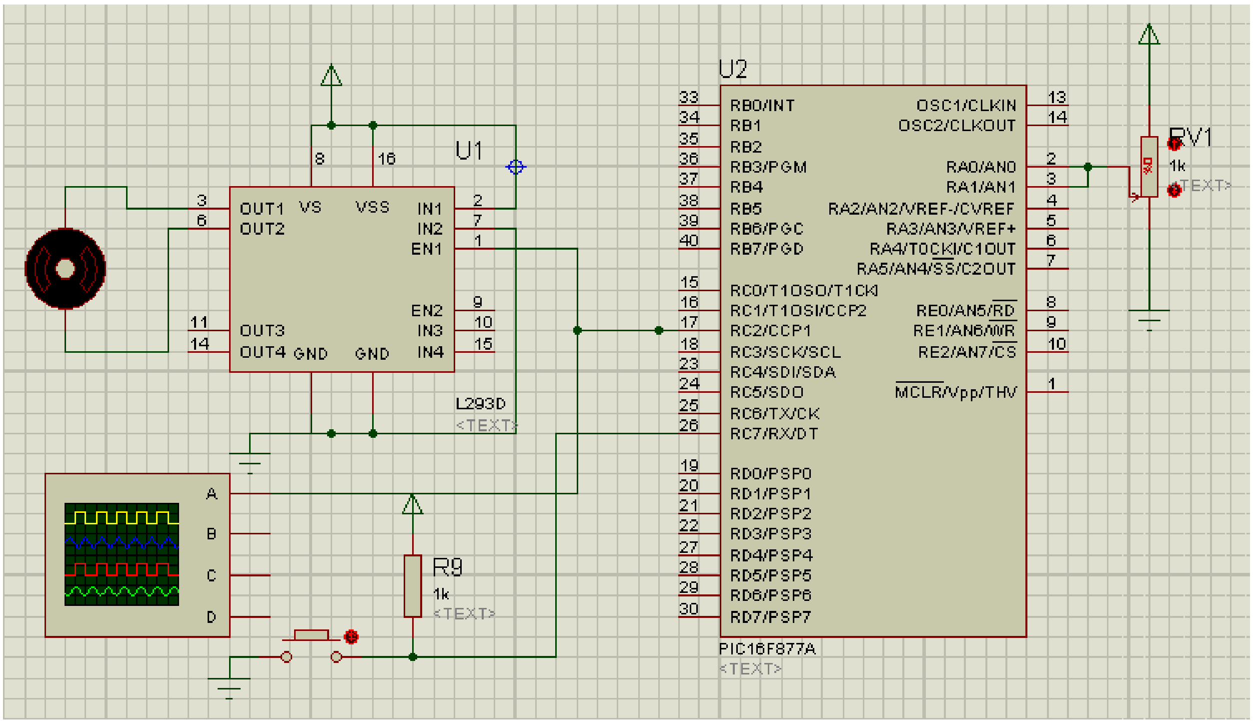


Figure 6.1: Reading analog value using built-in adc of pic microcontroller

ADC module of PIC microcontroller has 8 inputs for 40 pin devices.

The conversion of analog signal to PIC ADC module results in corresponding 10 bit digital number. PIC ADC module has software selectable high and low voltage reference input to some combination of VDD, VSS, RA2 and RA3.

In the following example project we will convert analog input to channel 1 to 10 bit digital number with low voltage reference (Vref-) 0v and high voltage reference (Vref+) 5V. The output will be displayed using 10 LEDs. You can change the Vref- and Vref+ by configuring the ADCON1 register.

Code: Testing of adc

```
#include<htc.h>

#define _XTAL_FREQ 10000000
```

```
#define _XTAL_FREQ 20000000
#define TMR2PRESCALE 4
long freq;
#define timer RC7

int PWM_Max_Duty();
PWM1_Init(long );
PWM1_Duty(unsigned int );
PWM1_Start();
PWM1_Stop();
void disptimer();
cmd();
data();
void main()
{
        TRISD = 0x00;
        TRISB = 0;
        TRISC=0b10000000;

        PWM1_Init(5000);
   PWM1_Start();


                ADCON0=0x05;
                ADCON1=0xc0;
                while(GO_DONE==1);
                int y;
```

```
            y=((ADRESH*256)+ADRESL);
            PWM1_Duty(y);



}

int PWM_Max_Duty()
{
  return(_XTAL_FREQ/(freq*TMR2PRESCALE);
}

PWM1_Init(long fre)
{
  PR2 = (_XTAL_FREQ/(freq*4*TMR2PRESCALE)) - 1;
  freq = fre;
}


PWM1_Duty(unsigned int duty)
{
  if(duty<1024)
  {
    duty = ((float)duty/1023)*PWM_Max_Duty();
    CCP1X = duty & 2;
    CCP1Y = duty & 1;
    CCPR1L = duty>>2;
  }
}

PWM1_Start()
{
  CCP1M3 = 1;
  CCP1M2 = 1;
  #if TMR2PRESCALAR == 1
    T2CKPS0 = 0;
    T2CKPS1 = 0;
  #elif TMR2PRESCALAR == 4
    T2CKPS0 = 1;
    T2CKPS1 = 0;
  #elif TMR2PRESCALAR == 16
    T2CKPS0 = 1;
    T2CKPS1 = 1;
  #endif
  TMR2ON = 1;
  TRISC2 = 0;
}

PWM1_Stop()
{
  CCP1M3 = 0;
  CCP1M2 = 0;
}
```

6.2.2 Interfacing and testing of LCD

JHD 16×2 Character LCD is a very basic LCD module which is commonly used in electronics projects and products. This type of LCD screen can display **2** lines with 16 characters each. Every character consists of 5×8 or 5×11 dot matrix. It can display all the letters of alphabet, Greek alphabet, mathematical symbols, punctuation marks etc. It is also likely to display symbols prepared by the user. Interfacing of LCD with PIC Microcontroller is as shown:

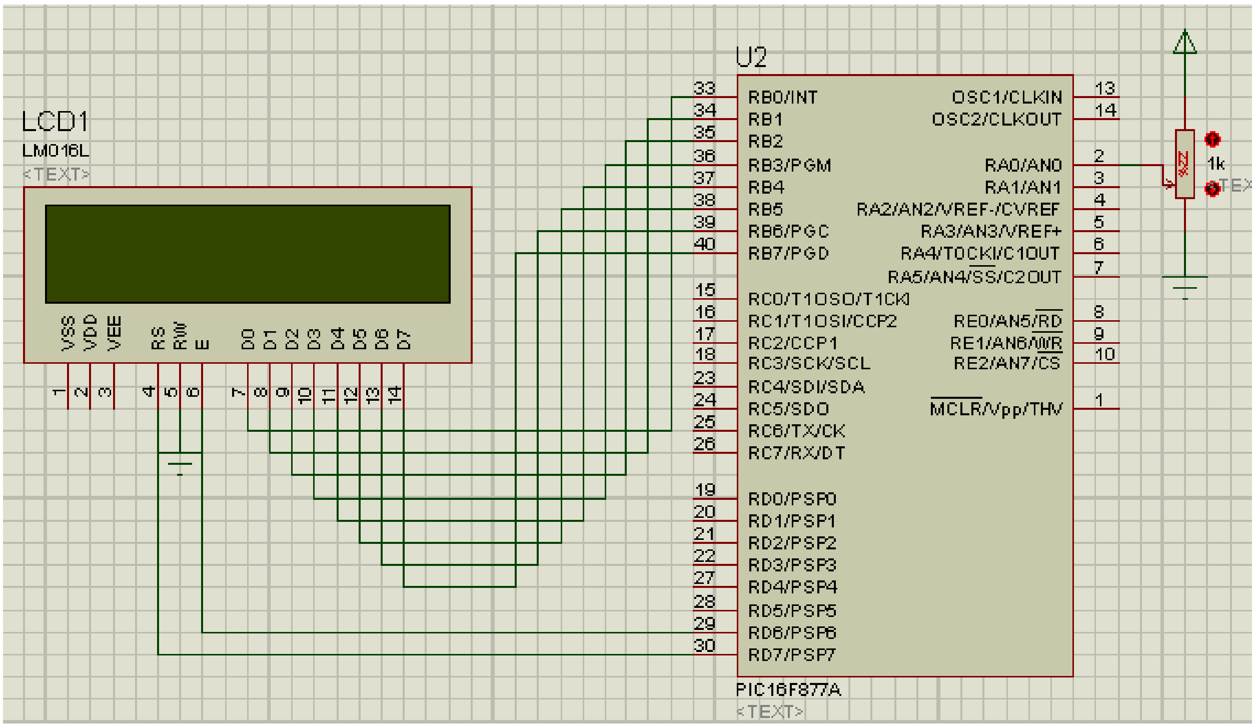


Figure 6.2: Interfacing lcd with pic microcontroller

It contains 2 rows that can display 16 characters. Each character is displayed using 5×8 or 5×10 dot matrix.

Code: Testing and interfacing of LCD

```
#include<htc.h>

__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_OFF & BOREN_OFF &
LVP_OFF & WRT_OFF & DEBUG_OFF & CP_ON);


#define _XTAL_FREQ 12000000


#define data_lcd PORTB
```

```
#define lcd_rs RD7

#define lcd_en RD6

cmd();

data();

void main()

{

        TRISB = 0;

        TRISD = 0;

        TRISA = 1;

        data_lcd = 0x38;

        cmd();

        data_lcd = 0x0C;

        cmd();

        data_lcd = 0x80;

        cmd();


                ADCON0=0x05;

                ADCON1=0xc0;

                while(GO_DONE==1);

                int y;

                y=((ADRESH*256)+ADRESL);

                int a,b,c,d,e;

                e=y*.48828;          /* stepsize, 5/1024 = 4.8828*10^-3 */

                a=e%10;

                e=e/10;

                b=e%10;

                e=e/10;
```

```
c=e%10;

e=e/10;

d=e%10;


data_lcd = 'T';

data();

data_lcd='e';

data();

data_lcd = 'm';

data();

data_lcd='p';

data();

/*data_lcd = 'e';

data();

data_lcd='r';

data();

data_lcd = 'a';

data();

data_lcd = 't';

data();

data_lcd = 'u';

data();

data_lcd = 'r';

data();

data_lcd = 'e';

data(); */

data_lcd=':';
```

```
                data();
                data_lcd = '-';
                data();
                data_lcd=' ';
                data();
                data_lcd=d+'0';
                data();
                data_lcd=c+'0';
                data();
                data_lcd=b+'0';
                data();
                data_lcd=a+'0';
                data();
                data_lcd = ' ';
                data();
                data_lcd='C';
                data();


}
cmd()
        {
        lcd_rs=0;       //RS(register select)selected in command mode
        lcd_en=1;
        __delay_ms(10);
        lcd_en=0;
        __delay_ms(10);
        }
data()
        {


        lcd_rs=1;       //RS(register select)selected in data mode
        lcd_en=1;
        __delay_ms(10);
        lcd_en=0;
        __delay_ms(10);
        }
```

6.3Schematics of the Project

6.3.1Power Supply Section

The power supply section provides a 5V DC output. The bridge rectifier section converts the 12V sinusoidal ac output of 12-0-12V transformer to 12V pulsating DC. The capacitor removes its ripples and smoothens the 12V pulsating DC which is fed to the input of 7805 voltage regulator. 7805 gives a constant 5V DC as the output which is the power supply for the whole circuit.

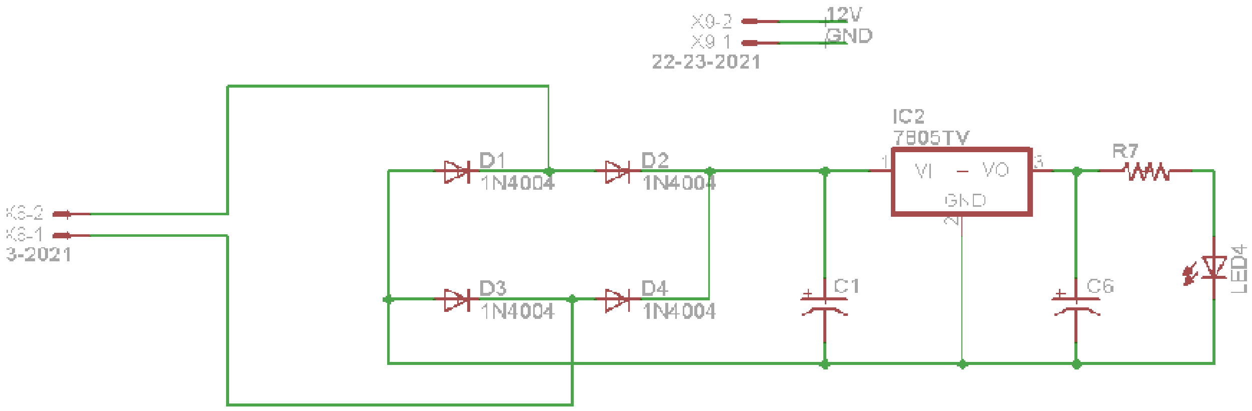


Figure 6.3: Circuit Diagram for Power Supply

6.3.2 Light Transmitter and Receiver Section

This section will convert the blood variation in our figure into the current variation. As the light will be emitted by the red and ir led which reflection from our figure is received by the photo diode and IR receiver.

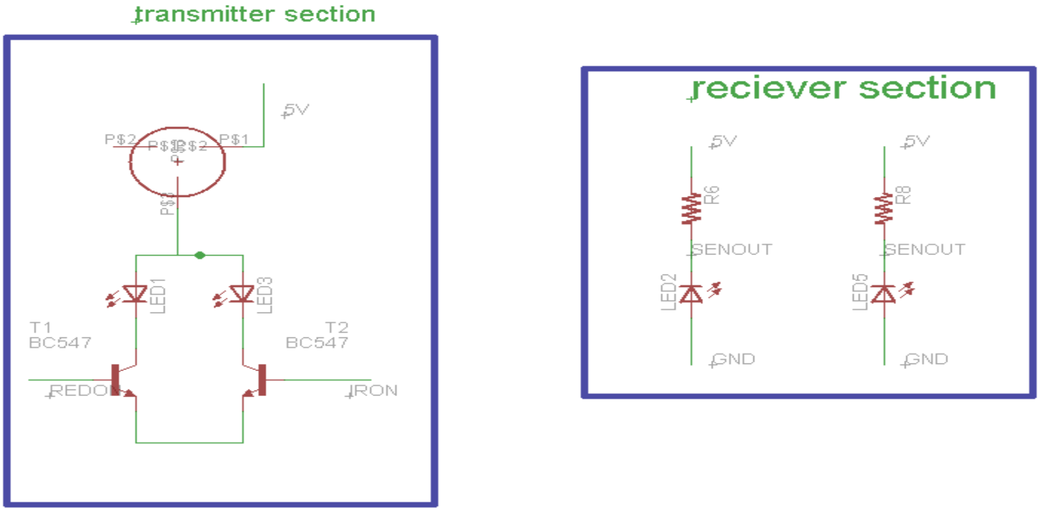


Figure 6.4: Circuit Diagram for led driver and photo detector

6.3.3 Signal Processing Circuit

As we are concerned with only the variation which is related to oxygen and heart rate hence we will design a filter which will focus the vision only to this range.

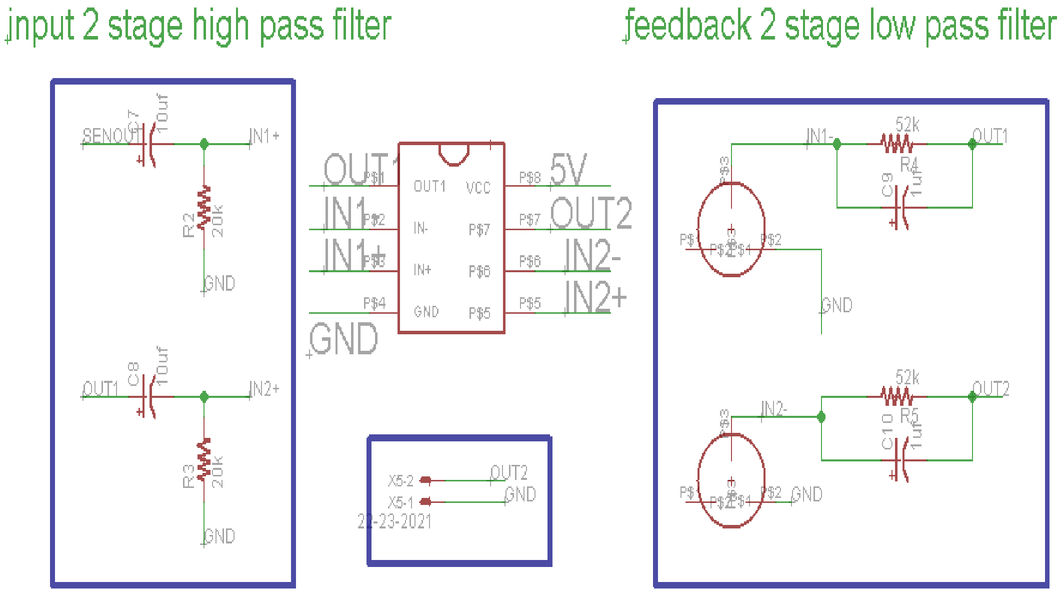


Figure 6.5: Circuit Diagram for signal processing unit

Genuinely the strength of the received signal is very low so the amplifier stage is employed and the variation of our interest lies only in a range of 0 Hz to maximum of 2 Hz to 3 Hz.

6.3.4 Controller section

This section involves the necessary reset circuit and oscillator circuitry along with a lcd which will display the heart rate and oxygen saturation output .

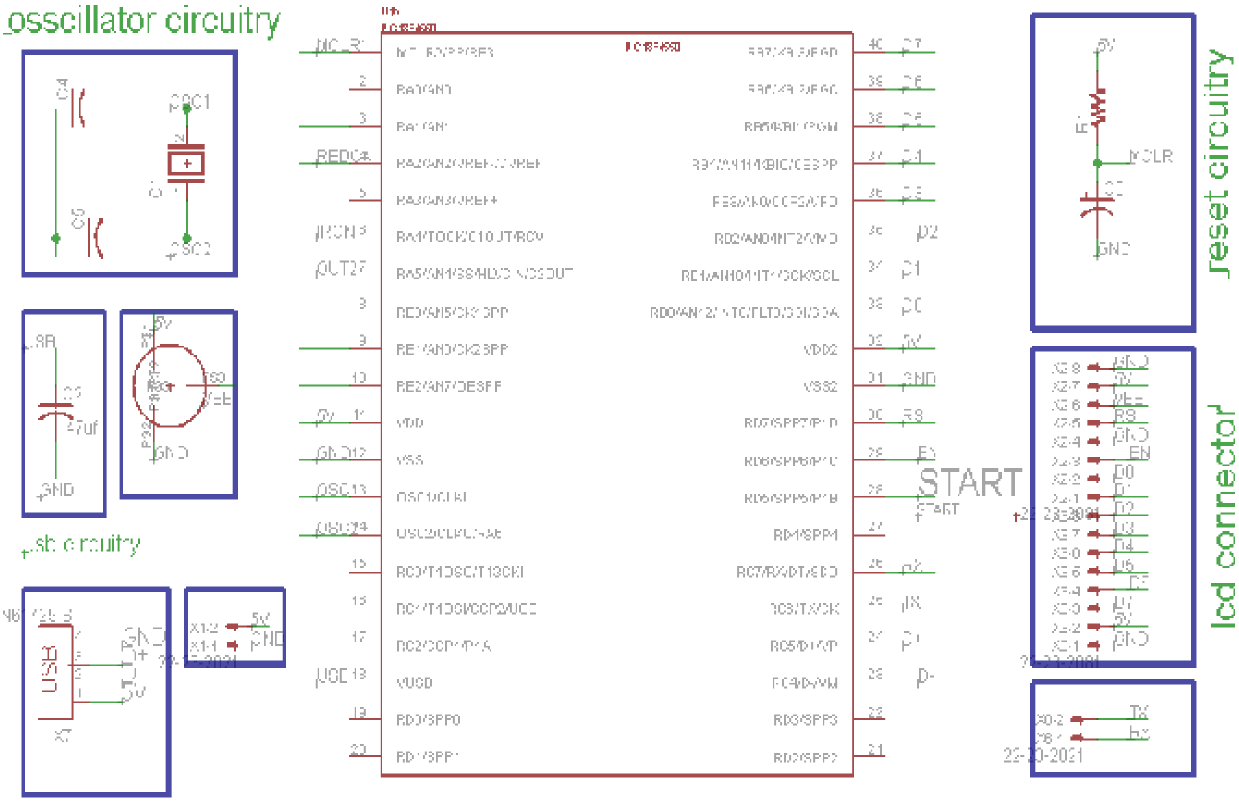


Figure 6.6: Circuit Diagram for Controller Section

6.4 Complete Circuit Diagram

The complete circuit diagram with all the modules and components interfaced for fulfilling the requisites of this project is shown below. The entire embedded system consists of three parts first is the pulse Oximeter section which gets the input preserved by the led and IR receiver from the light emitted by the transmitter. The next unit consists of amplifies as the input received is enormously weak and filter in only the informative part and fed it to the microcontroller analog channel. Microcontroller calculates the spo2 from the input signal. The heart rate and spo2 is displayed on the LCD. The LCD is also connected with the pot for intensity and contrast control. The crystal of 20MHz is attached for providing the clock to the microcontroller with two bypass capacitor.

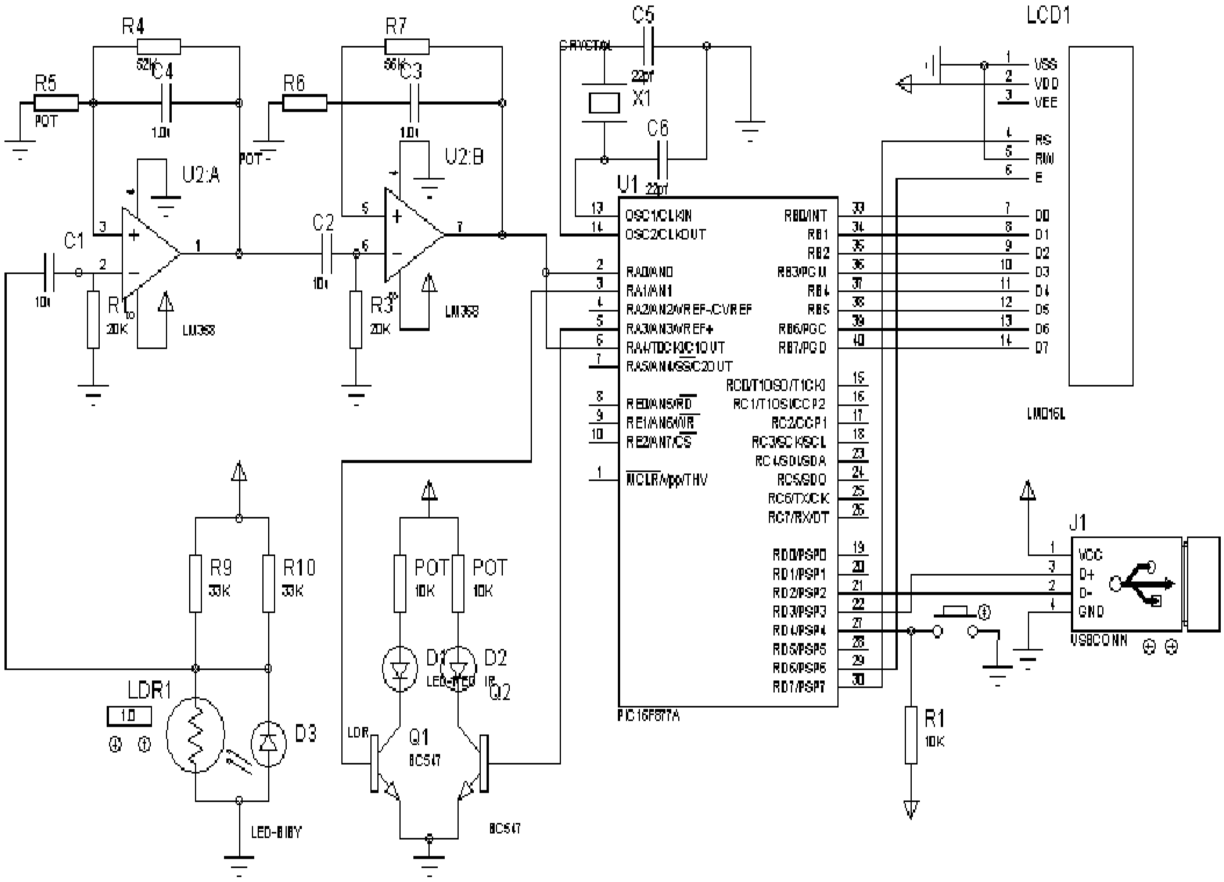


Figure 6.8: Complete Circuit Diagram of the Project

6.4 Conclusion

This chapter explained the whole result of the designing of the project. The various interfacings, design circuits, simulations and PCB designing are discussed in detail. We find that the both accuracy and speed cannot be achieved in this project to improve the one other has to be compromised. We tried to setup a trade between the two in which we are be able to achieve a response time of 17 second at max at a cost of sacrificing 6 % of accuracy at every heartbeat missed or excessed. The system is stable but if any change occurs in the nearby atmosphere to the sensor it may consider this also as a stimulus. The error in the oxygen saturation calculations due to heart rate are zero theoretically but may arise due to approximation and quantization which are extremely small.

.

Chapter 7

CONCLUSION AND FUTURE SCOPE

This chapter contains the conclusion of the entire project and also specifies its future scope. The conclusions mention the results that were supposed to be achieved from the project. It also tells about the limitations of the project that can be removed or minimized by future modifications.

7.1 Conclusion

The central idea which motivated the design of project is to provide the user a low cost device which is reliable, comfortable to wear, easy to configure and operate, highly responsive and accurate. Due to the use of fuzzy logic for the calibration, the above system is more accurate as compared to the earlier proposed systems.

7.2 Future scope

The conventional pulse oximeter uses two wavelengths (red and infrared) transmitted through a finger and a photo detector to analyse arterial haemoglobin oxygen saturation and pulse rate. The future advances in this field may include the use of multiple wavelengths of light to quantify methaemoglobin, carboxyhaemoglobin and total haemoglobin content in the blood. The use of electronic processes to improve pulse oximeter signal processing during condition of low signal to noise ratio can also be preferred. The new generation of pulse oximeters can be designed to provide accurate measurements in challenging situations such as low perfusion, the presence of motion artefacts and low arterial oxygen saturation and that will improve the patient monitoring system.

References

[1] en.wikipedia.org/wiki/Pulse_oximetry

[2] Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong

[3] http://cinc.mit.edu/current/preprints/340.pdf

[4] Sensors 2013, 13, 574-586; DOI:10.3390/s130100574

[5] Wei Chen ; Dept. of Ind. Design, Eindhoven Univ. of Technol., Eindhoven, Netherlands ; Ayoola, I. ; Oetomo, S.B. ; Feijs, ″Design, Automation & Test in Europe Conference & Exhibition”, 2010 8-12 March 2010, ISSN : 1530-1591.DOI: 10.1109.

[6] Grantham Pang and Chao Ma, 2014,”A Neo-Reflective Wrist Pulse Oximeter” Digital Object Identifier 10.1109/ACCESS.2014.2382179.

[7] Jianchu Yao, M.S. and Steve Warren,“In-House Pulse Oximeter Design” Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition Copyright, American Society for Engineering Education.

[8] Sangeeta Bagha and Laxmi Shaw, Dec 2011,” A Real Time Analysis of PPG Signal for Measurement of SpO2 and Pulse Rate” *International Journal of Computer Applications (0975 – 8887)* Volume 36– No.11.

[9] http://physics.stackexchange.com/questions/127667/why-is-it-that-the-cathodic-battery-symbol-in-an-equivalent-battery-circuit-does

[10] https://www.fairchildsemi.com/ds/1N/1N914.pdf

[11] https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf

[12] http://www.learningaboutelectronics.com/Articles/What-is-a-step-down-transformer

[13] http://www.engineersgarage.com/electronic-components/irsensor

[14] http://www.ti.com/lit/ds/symlink/snlm358.pdf

[15] www.osioptoelectronics.com

[16] en.wikipedia.org/wiki/Liquid-crystal-display

[17] www.microchip.com/pic16f877a

[18] http://www.cadsoftusa.com/

[19] en.wikipedia.org/wiki/Microchip_Technology

[20] http://www.labcenter.com/products/pcb/pcb_overview.cfm

[21] http://www.htsoft.com/

[22] http://in.mathworks.com/support/sysreq/sv-r2013a/

APPENDIX A

DATASHEETS

A.1 PIC 16F877A

A.2 Comparator LM 358

APPENDIX B

SOFTWARE USED

B.1 MPLAB IDE

The MPLAB IDE combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. The MPLAB development platform is easy-to-use and helping you quickly create embedded programs that work.

- Create a project, select the target chip from the device database, and configure the tool settings.
- Create source files in C or assembly.
- Build your application with the project manager.
- Correct errors in source files.
- Test the linked application.
- Debug the executable logic by watching program flow with the built-in simulator or in real time with in-circuit emulators or in-circuit debuggers.
- Make timing measurements with the simulator or emulator.
- Program firmware into devices with device programmers.

Create a Project

MPLAB includes a project manager which makes it easy to design applications for a PIC based microcontroller. You need to perform the following steps to create a new project:

- Start MPLAB and select the toolset
- Create a project file and select a CPU from the device database.

- Create a new source file and add this source file to the project.
- Add and configure the startup code for the PIC.
- Set tool options for target hardware.
- Build project and create a HEX file for Flash programming.

Start MPLAB

MPLAB is a standard Windows application and started by clicking on the program icon.

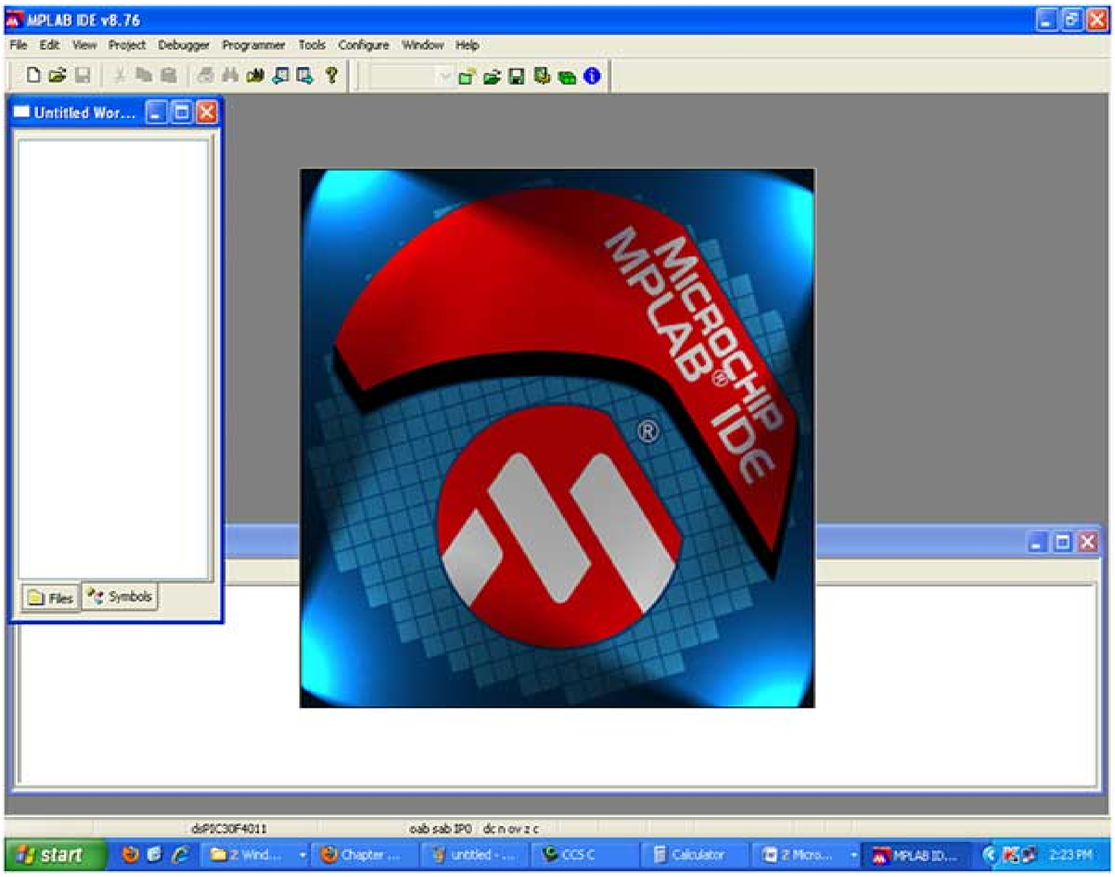
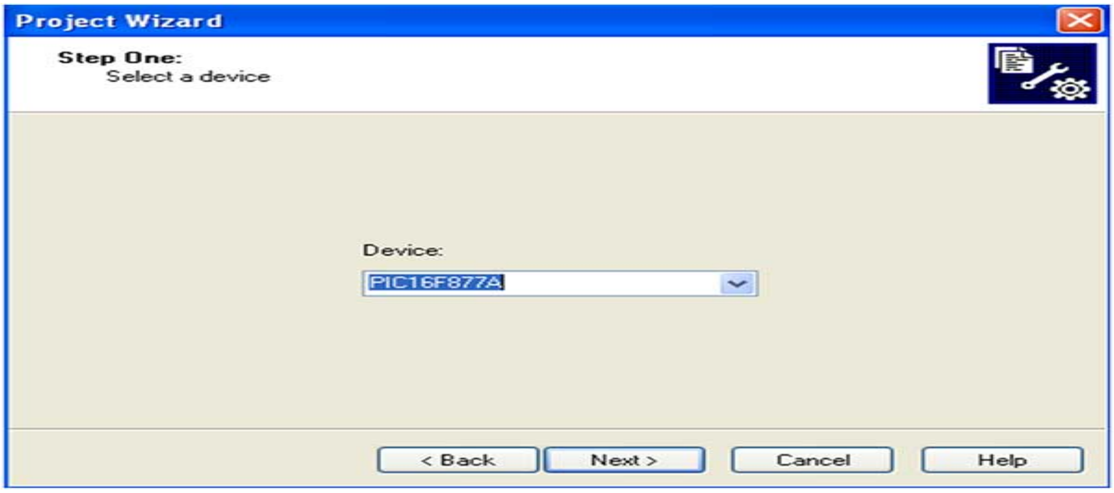


Figure B.1.1 Starting Window of MPLAB IDE

Create a Project File

To create a new project file select from the MPLAB menu Project – Project Wizard. This opens a standard Windows dialog that asks you for the new project file name. Now we can select Next button then the window shows as shown in below. Then we can select the device. Devices are selected by using the drop-down button in the device option.

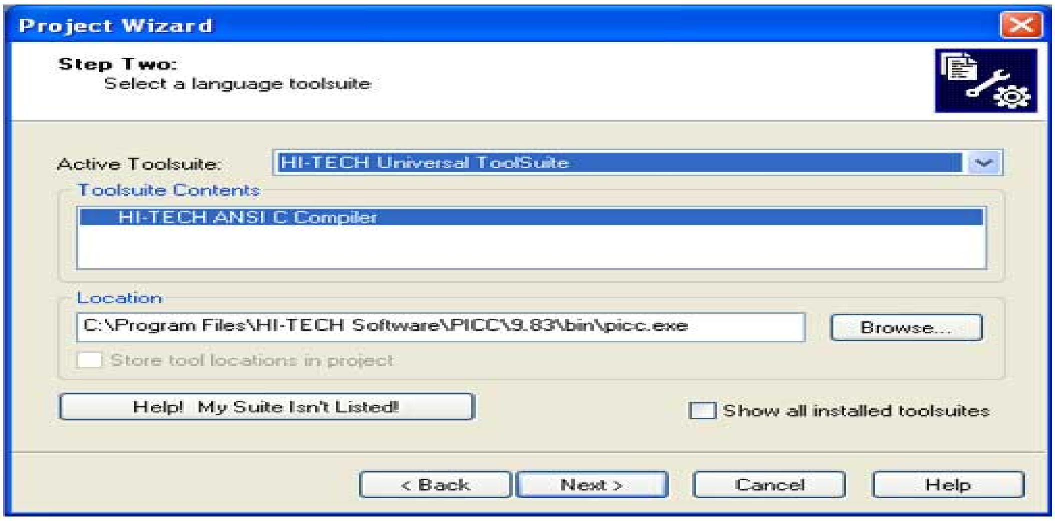


Device: PIC16F877A

Figure B.1.2 Select the Device (Microcontroller)

Select Language Tools

Step two of the Project Wizard sets up the language tools that are used with this project. Select Hi-Tech Universal Tool suite in the top pull down. Then you should see PICC, HLINK and LIBR show up in the Tool suite Contents box. You can click on each one to see its location. If you installed MPLAB IDE into the default directory, the Hi-Tech compiler executable will be:



Active Toolsuite: HI-TECH Universal ToolSuite

Toolsuite Contents: HI-TECH ANSI C Compiler

Location: C:\Program Files\HI-TECH Software\PICC\9.83\bin\picc.exe

□ Store tool locations in project

□ Show all installed toolsuites

Figure B.1.3 Select the compiler to be used and location of file.

Naming the Project

Step Three of the wizard allows you to name the project and put it into a folder. This sample project will be called MPLAB. Using the Browse button, place the project in a folder named MPLAB. Click Next>.

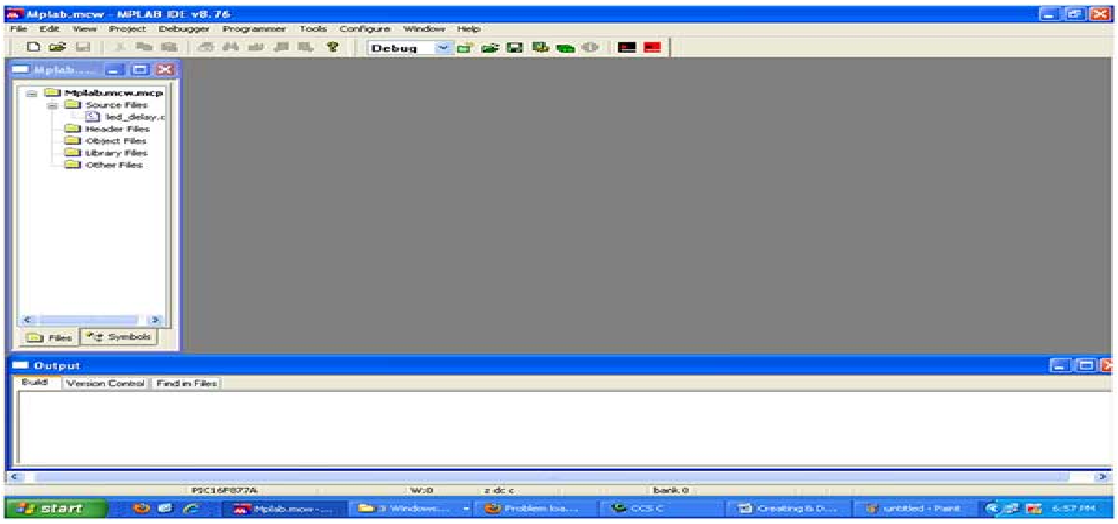
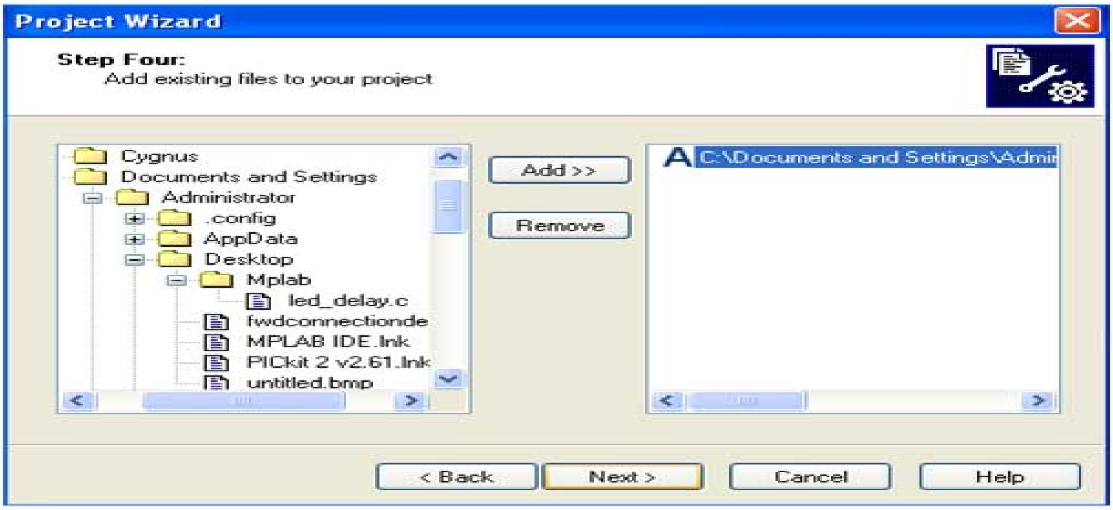


Figure B.1.4 Naming the project in MPLAB IDE

Adding files to Project

Step Four of the Project Wizard allows file selection for the project. A source file has not yet been selected, so we will use an MPLAB IDE template file. These files are in the MPLAB IDE folder, which by default is in the Program Files folder on the PC. Choose the file named led_delay.c. If MPLAB IDE is installed in the default location, the full path to the file will be

C:\Documents and Settings\Administrator\Desktop\Mplab\led_delay.c

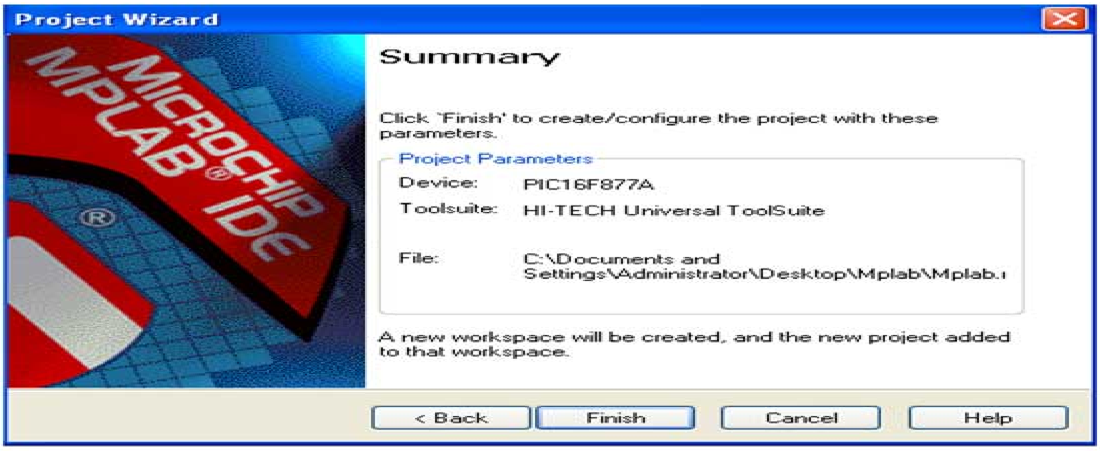


Project Wizard

Step Four:
Add existing files to your project

Figure B.1.5 Adding files to project in MPLAB IDE

Press Add>> to move the file name to the right panel, and the file name to enable this file to be copied to our project directory. Make sure that your dialog looks like the picture above, with both check boxes checked, then press Next> to finish the Project Wizard.



Summary

Click 'Finish' to create/configure the project with these parameters.

Project Parameters

Device: PIC16F877A

Toolsuite: HI-TECH Universal ToolSuite

File: C:\Documents and Settings\Administrator\Desktop\Mplab\Mplab.

A new workspace will be created, and the new project added to that workspace.

Figure B.1.6 Finishing window in MPLAB IDE

The final screen of the Project Wizard is a summary showing the selected device, the tool suite and the new project file name. After pressing the Finish button, review the Project Window on the **MPLAB IDE desktop**. If the Project Window is not open, then select View>Project.

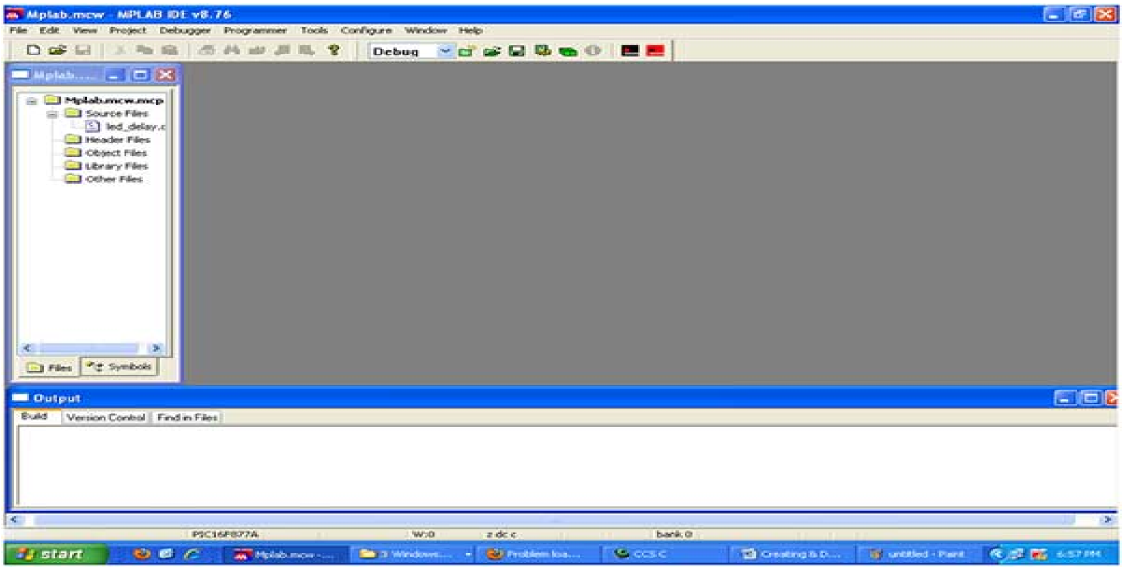


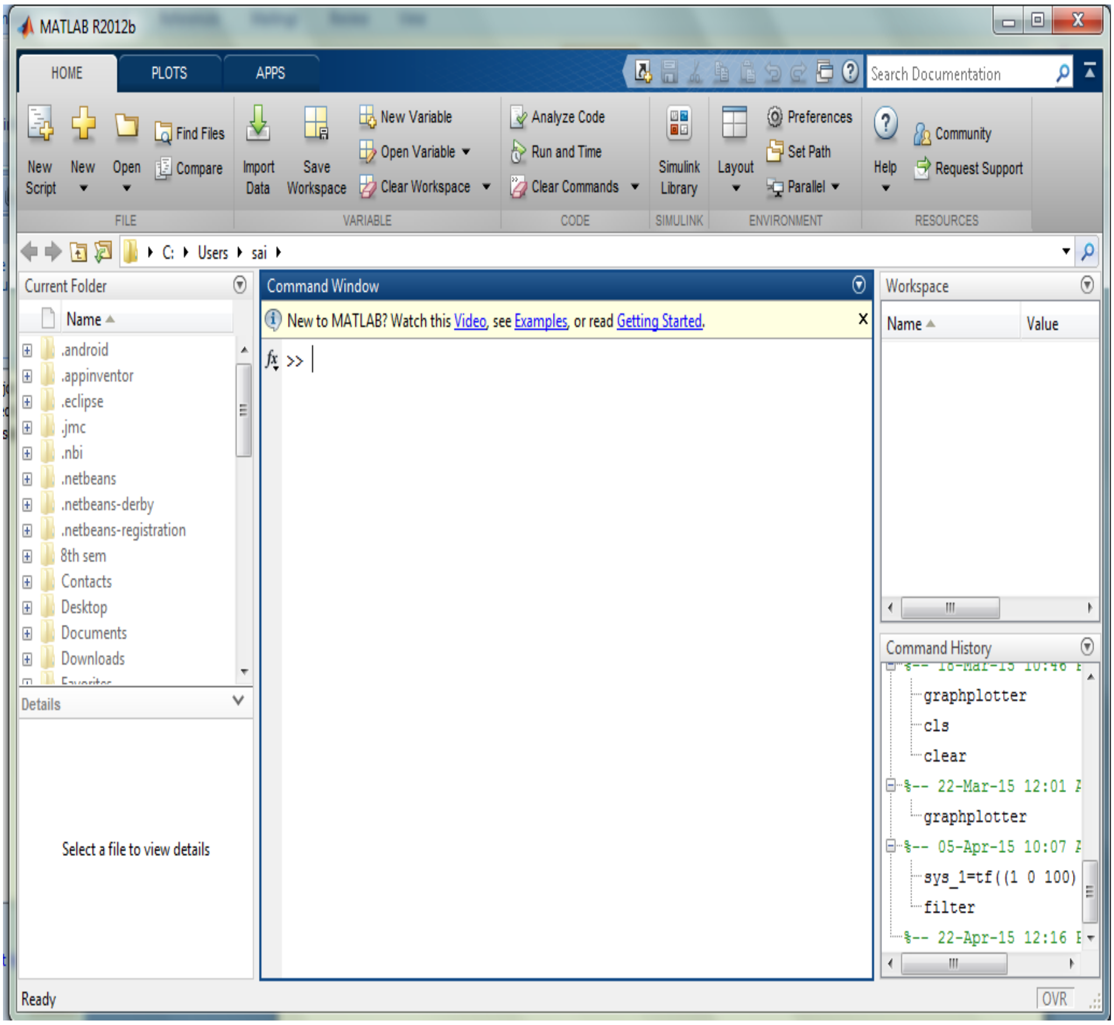
Figure B.1.7 Project window in MPLAB IDE

Note: Files can be added and projects saved by using the right mouse button in the project window. In case of error, files can be manually deleted by selecting them and using the right mouse click menu.

B.2 ANALOG FILTER DESIGN IN MATLAB

The Analog Filter Design block designs and implements a Butterworth, Chebyshev type I, Chebyshev type II, or elliptic filter in a high pass, low pass, band pass, or band stop configuration. The input must be a sample-based, continuous-time, real-valued, scalar signal. The design and band configuration of the filter are selected from the Design method and Filter type pop-up menus in the dialog box. For each combination of design method and band configuration, an appropriate set of secondary parameters is displayed.

Step1: Open MATLAB Window



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

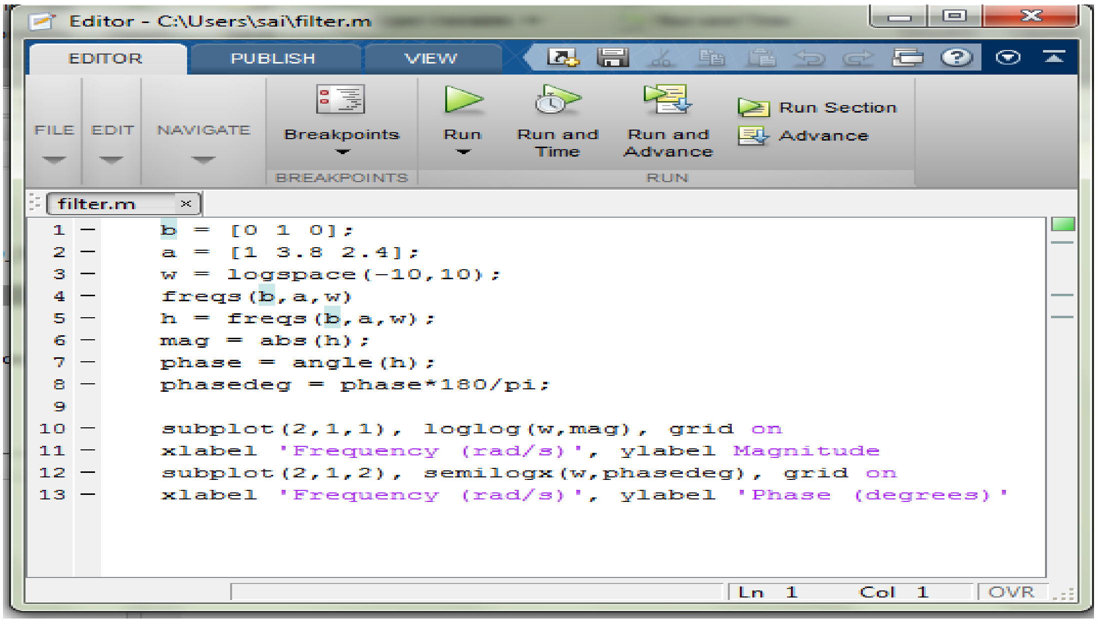
```
fx >>
```

```
%-- 18-Mar-15 10:46 
graphplotter
cls
clear
%-- 22-Mar-15 12:01 
graphplotter
%-- 05-Apr-15 10:07 
sys_1=tf((1 0 100)
filter
%-- 22-Apr-15 12:16 
```

Figure B.2.1: Main window of MATLAB

Step2 : Open New Script for Writing the Code for Filter Design.

Step 3: Writing the code into script window including the transfer function of the filter to be realized.



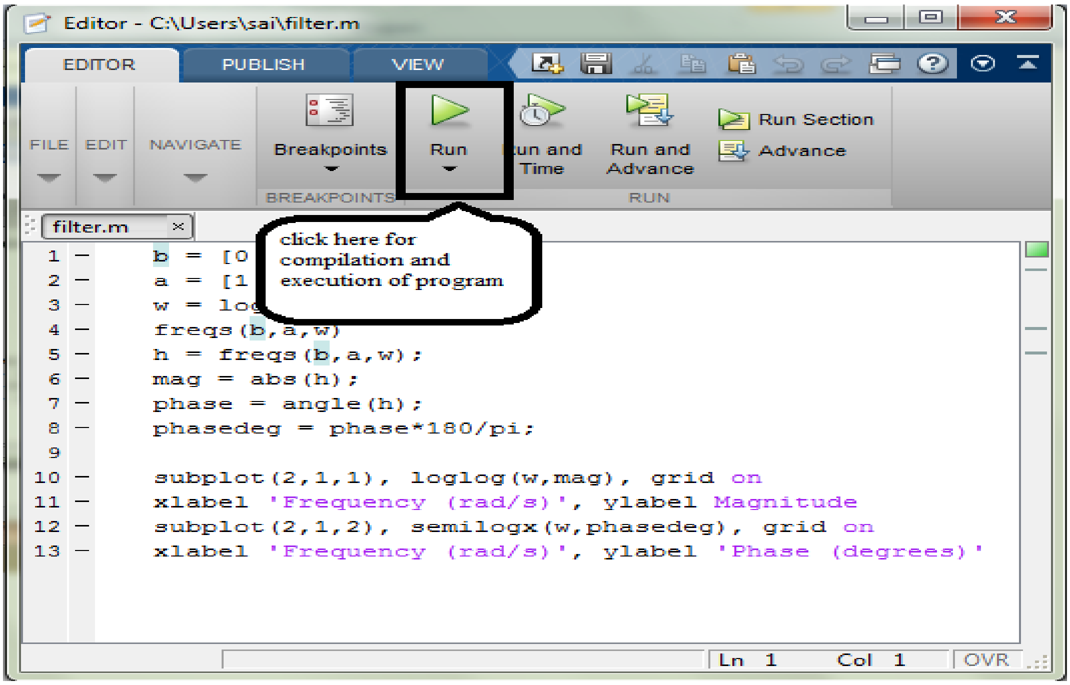
```
b = [0 1 0];
a = [1 3.8 2.4];
w = logspace(-10,10);
freqs(b,a,w)
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
phasedeg = phase*180/pi;

subplot(2,1,1), loglog(w,mag), grid on
xlabel 'Frequency (rad/s)', ylabel Magnitude
subplot(2,1,2), semilogx(w,phasedeg), grid on
xlabel 'Frequency (rad/s)', ylabel 'Phase (degrees)'
```

Figure B.2.3: Code Written on Script in MATLAB

Step 4: Execution of the code can be done in two ways

- Click on the run icon in the script window or
- Write the name of the file in command window and click enter



click here for compilation and execution of program

```
b = [0
a = [1
w = lo
freqs(b,a,w)
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
phasedeg = phase*180/pi;

subplot(2,1,1), loglog(w,mag), grid on
xlabel 'Frequency (rad/s)', ylabel Magnitude
subplot(2,1,2), semilogx(w,phasedeg), grid on
xlabel 'Frequency (rad/s)', ylabel 'Phase (degrees)'
```

Figure B.2.4: Code Compilation and Execution using Script Window in MATLAB

Step 5: Output window in MATLAB

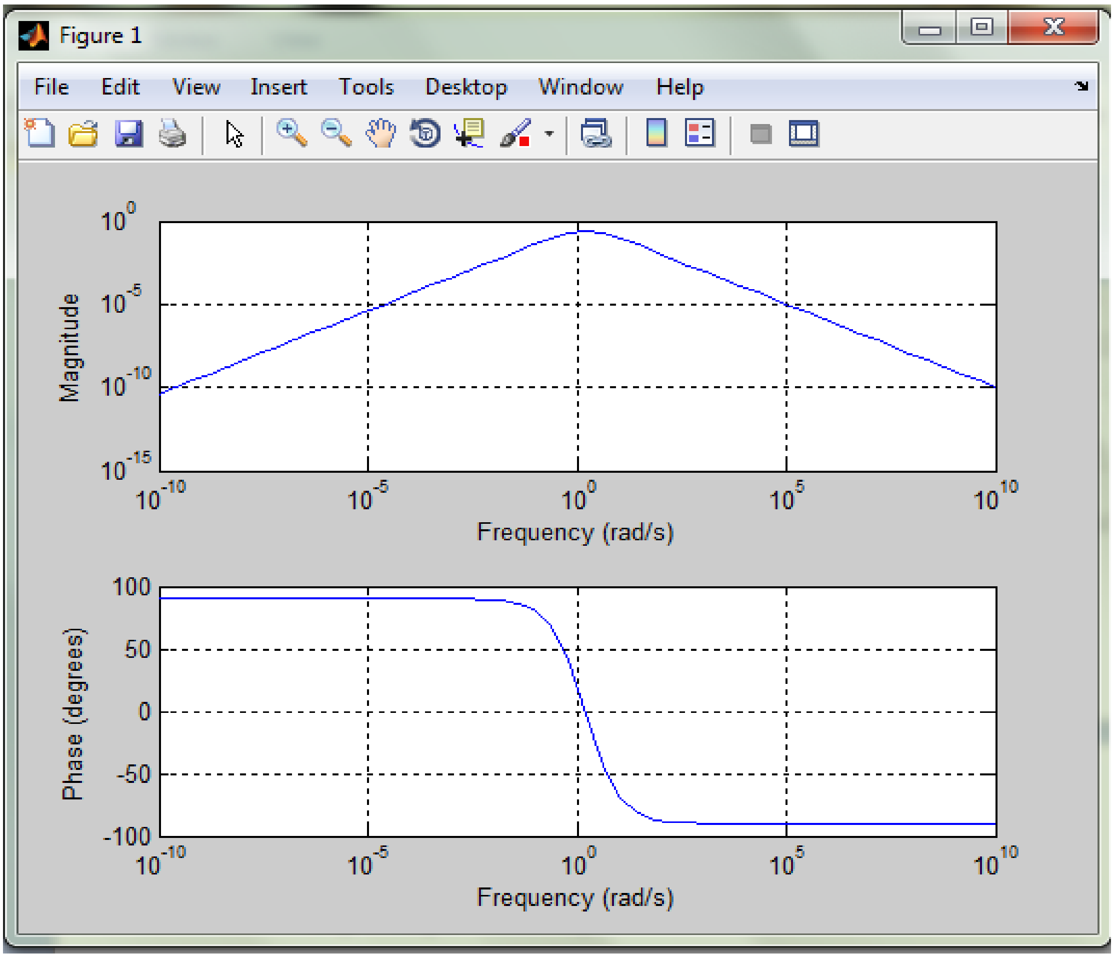


Figure B.2.6: Output window in MATLAB

B.3 DESIGNING PCB USING EAGLE SOFTCAD

The name EAGLE is an acronym, which stands for **E**asily **A**pplicable **G**raphical **L**ayout **E**ditor. It contains **Schematic editor**, **layout editor**, library editor modules with identical user interfaces.

WORKING WITH EAGLE

- Create new project
- Create new schematic in the project
- Find and place ("add") components
- Add all the components required… eg here I would be taking 4 diodes and making a connection between them.
- After placement and proper netting (Connection) switch to board editor for routing.

Step 1: Create new project and then new schematic under it

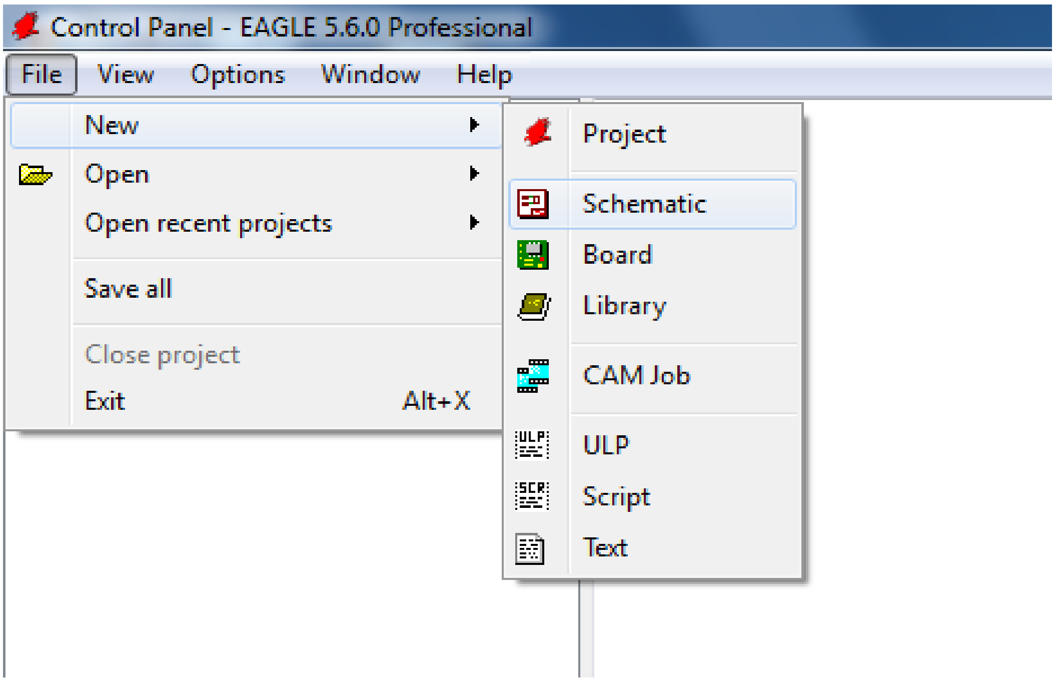
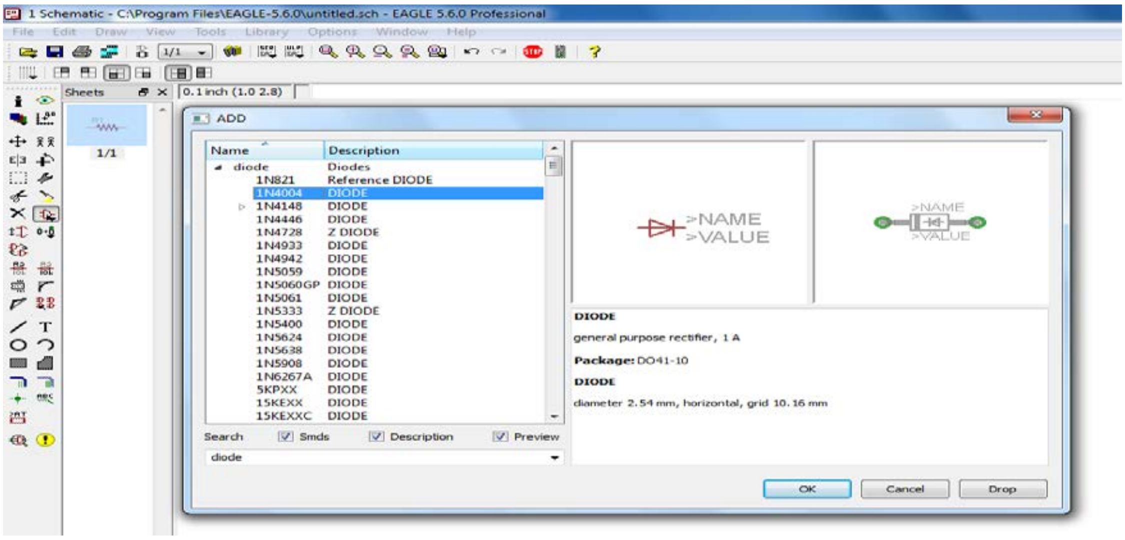


Figure B.3.1: Creating New Project and Schematic in EAGLE

Step 2: Find and place ("add") components if exactly the same device couldn't be found then a substitute for it can be used. Further even if this also doesn't work then you yourself will have to create a library.



1 Schematic - C:\Program Files\EAGLE-5.6.0\untitled.sch - EAGLE 5.6.0 Professional

0.1 inch (1.0 2.8)

Name	Description
diode	Diodes
1N821	Reference DIODE
1N4004	DIODE
1N4148	DIODE
1N4446	DIODE
1N4728	Z DIODE
1N4933	DIODE
1N4942	DIODE
1N5059	DIODE
1N5060GP	DIODE
1N5061	DIODE
1N5333	Z DIODE
1N5400	DIODE
1N5624	DIODE
1N5638	DIODE
1N5908	DIODE
1N6267A	DIODE
5KPXX	DIODE
15KEXX	DIODE
15KEXXC	DIODE

DIODE

general purpose rectifier, 1 A

Package: DO41-10

DIODE

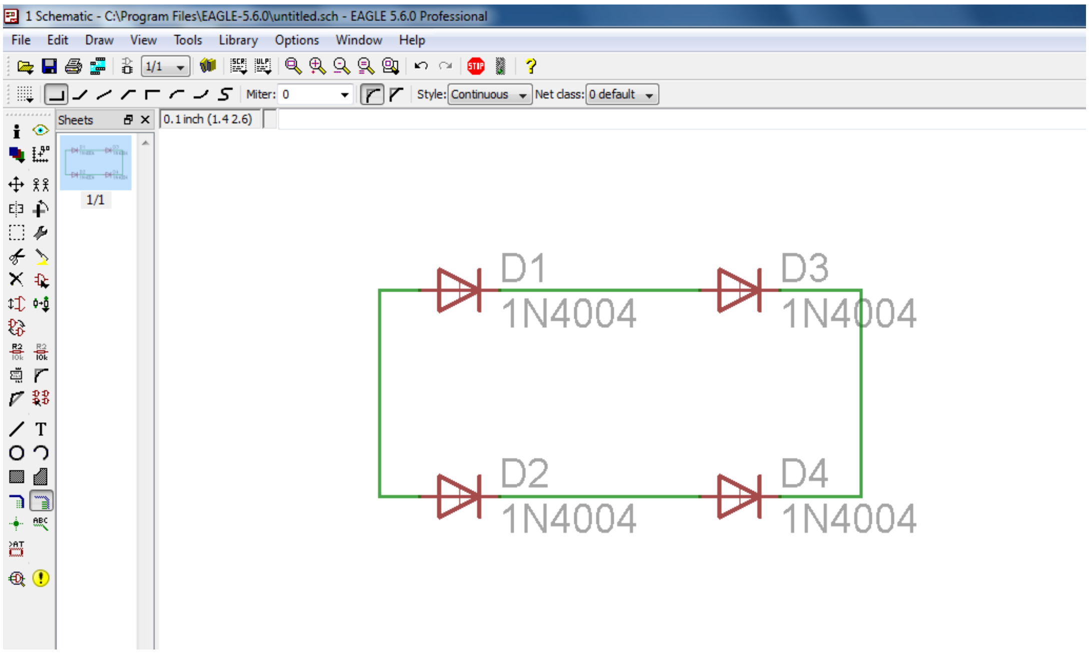
diameter 2.54 mm, horizontal, grid 10.16 mm

Search ☑ Smds ☑ Description ☑ Preview

diode

Figure B.3.2: Placing Components for Creating Circuit /Schematic in EAGLE

Step 3: Add all the components required to make the complete circuit on the schematic window using add click



1 Schematic - C:\Program Files\EAGLE-5.6.0\untitled.sch - EAGLE 5.6.0 Professional

Miter: 0 Style: Continuous Net class: 0 default

0.1 inch (1.4 2.6)

Figure B.3.3: Establishing the Connection Amongst all the Components

Step 4: Netting (Connection) switch to board editor for routing and establishing all the connections according to the circuit diagram and label can be used for reducing the intersection of the wires and understanding mess and multiple screen can be used for modularization of complex circuits.

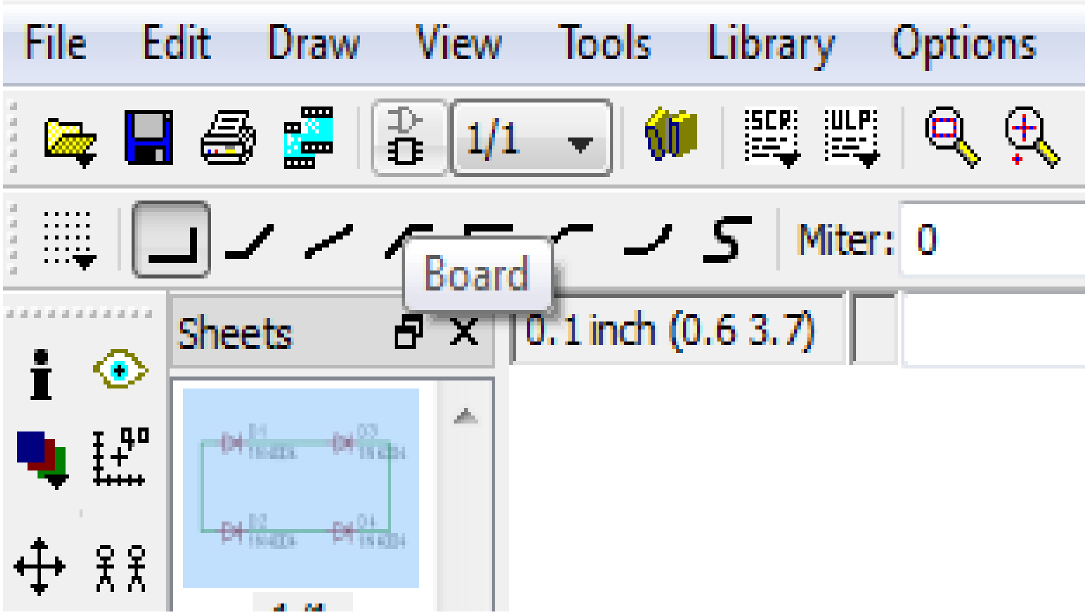


Figure B.3.4: Multiple windows of schematic in eagle

Step 5: Jump to board window from the schematic window for pcb design using switch to board click and place all the component’s footprints on to the board space provides and maximize the window

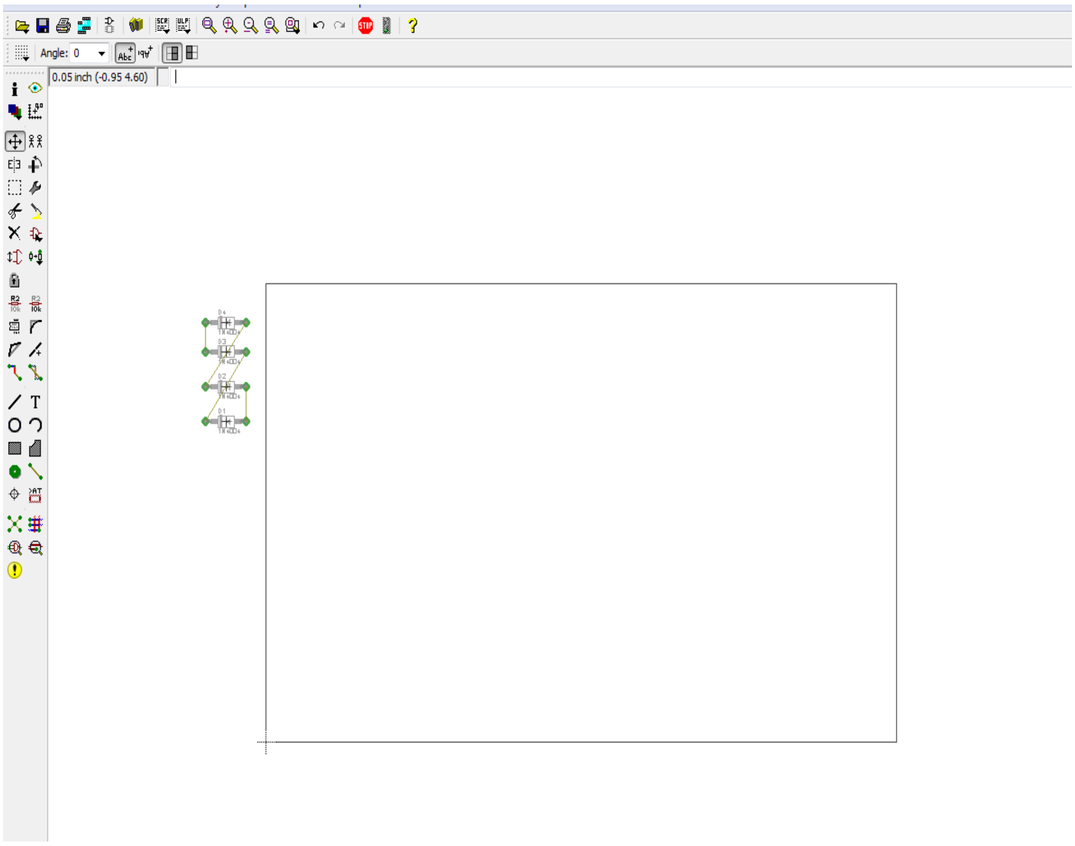


Figure B.3.5: Components and Board Screen in EAGLE

Now place components on board and route all the components in optimized way to reduce the area provided there is sufficient space for adequate soldering of PCB.

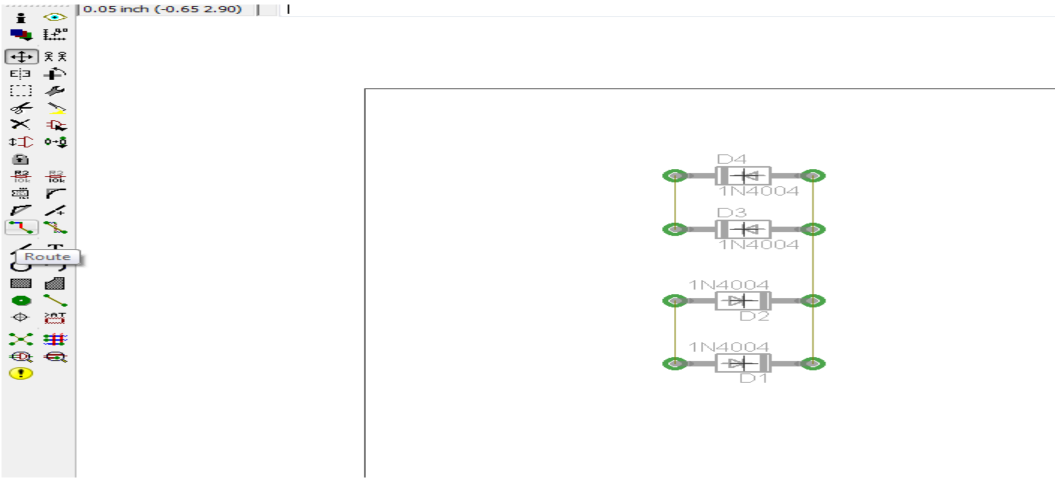


Figure B.3.6: Optimizing the Components Placement in the Board of EAGLE.

Step 6: Start the routing of the components without intersecting the unwanted tracks

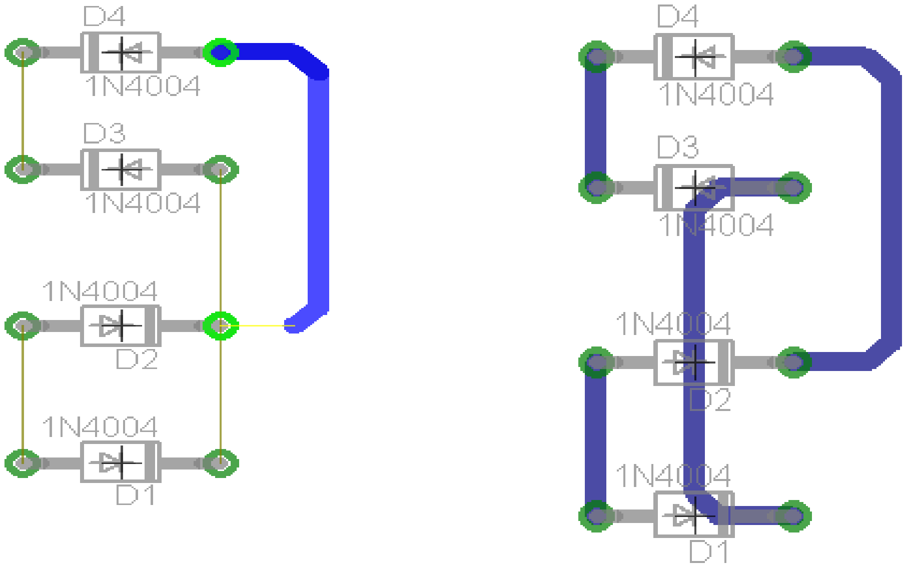


Figure B.3.7: Routing the Board in EAGLE

B.4 FUZZY LOGIC TOOLBOX

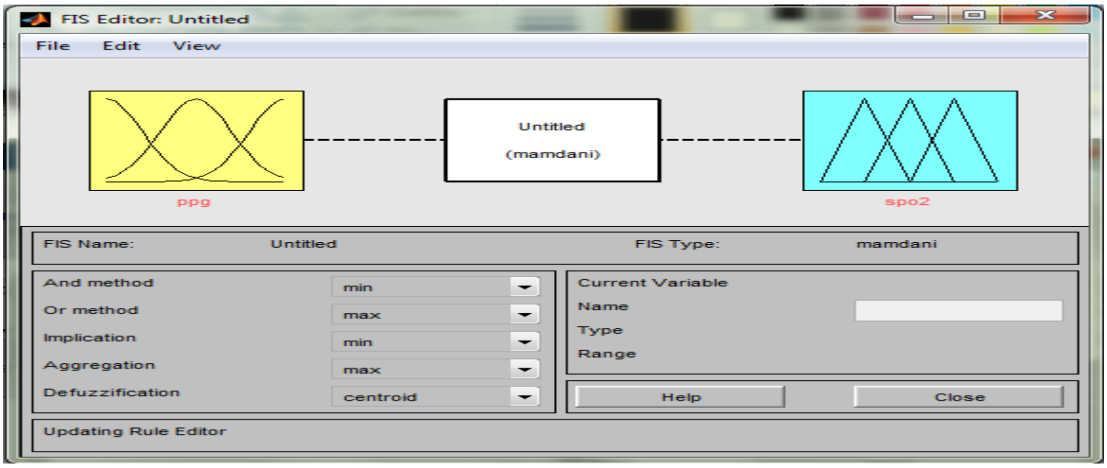
Fuzzy Logic Toolbox provides MATLAB functions, graphical tools, and a Simulink block for analyzing, designing, and simulating systems based on fuzzy logic. The product guides you through the steps of designing fuzzy inference systems. Functions are provided for many common methods, including fuzzy clustering and adaptive neuro-fuzzy learning. The toolbox let you model complex system behaviors using simple logic rules and then implements these rules in a fuzzy inference system. You can use it as a standalone fuzzy inference engine. Alternatively, you can use fuzzy inference blocks in Simulink and simulate the fuzzy systems within a comprehensive model of the entire dynamic system.

Key Features

- Specialized GUIs for building fuzzy inference systems and viewing and analyzing results
- Membership functions for creating fuzzy inference systems
- Support for AND, OR, and NOT logic in user-defined rules
- Standard Mamdani and Sugeno-type fuzzy inference systems
- Automated membership function shaping through neuro-adaptive and fuzzy clustering learning techniques
- Ability to embed a fuzzy inference system in a Simulink model
- Ability to generate embeddable C code or stand-alone executable fuzzy inference engines

Steps in fuzzy logic toolbox:

Step 1: Use Fuzzy Logic Toolbox to design fuzzy logic systems.



FIS Name: Untitled

FIS Type: mamdani

And method: min

Or method: max

Implication: min

Aggregation: max

Defuzzification: centroid

Current Variable

Name

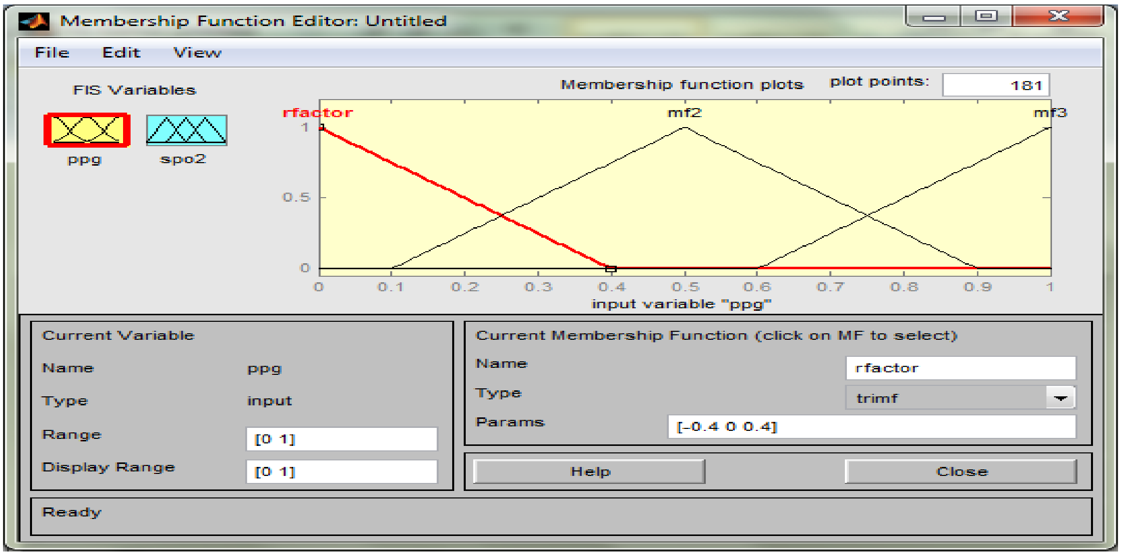
Type

Range

Updating Rule Editor

Figure B.4.1: Main Fuzzy Inference System Window

Step 2: Define membership functions and rules for fuzzy inference systems.



plot points: 181

Current Variable

Name ppg

Type input

Range [0 1]

Display Range [0 1]

Current Membership Function (click on MF to select)

Name rfactor

Type trimf

Params [-0.4 0 0.4]

Ready

Figure B.4.2: Window for Defining the Membership Function

Step 3: Simulate and analyze fuzzy inference systems.

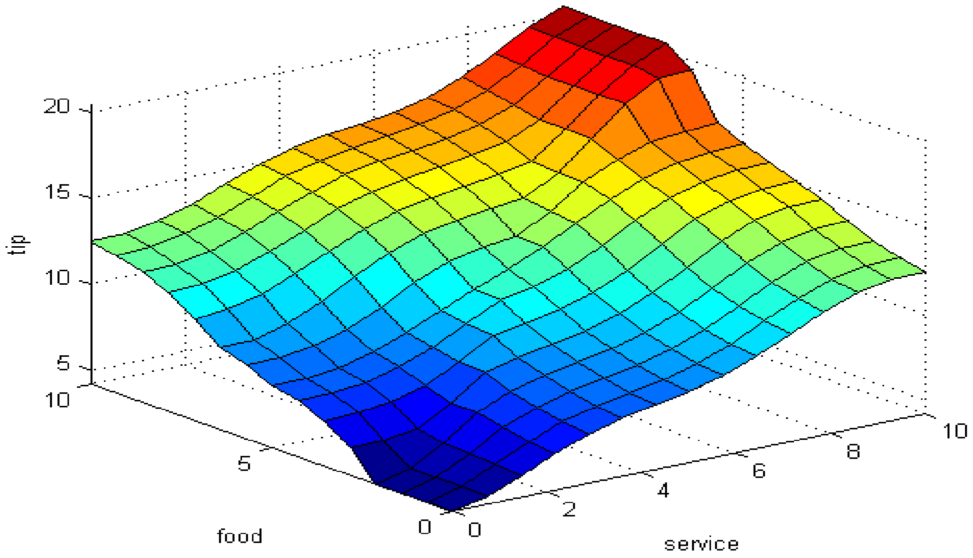
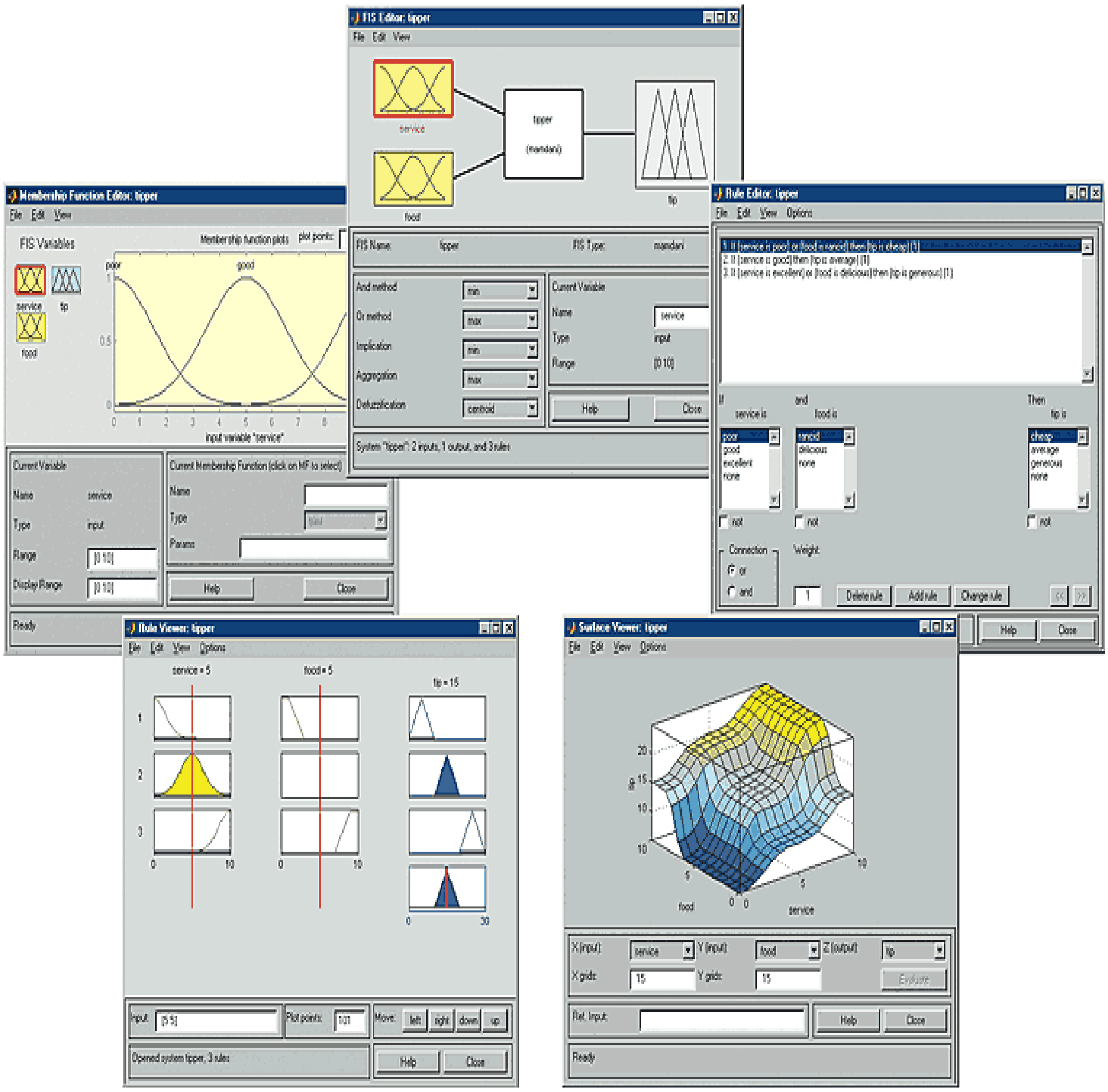


Figure B.4.3: Surface Indicating the Relationship Between the Input and Output Parameters

Building a Fuzzy Inference System

Fuzzy inference is a method that interprets the values in the input vector and, based on user-defined rules, assigns values to the output vector. Using the GUI editors and viewers in the Fuzzy Logic Toolbox, you can build the rules set, define the membership functions, and analyze the behavior of a fuzzy inference system (FIS). The following editors and viewers are provided:

- **FIS Editor** - Displays general information about a fuzzy inference system.
- **Membership Function Editor** - Lets you display and edit the membership functions associated with the input and output variables of the FIS
- **Rule Editor** - Lets you view and edit fuzzy rules using one of three formats: full English-like syntax, concise symbolic notation, or an indexed notation
- **Rule Viewer** - Lets you view detailed behavior of a FIS to help diagnose the behavior of specific rules or study the effect of changing input variables
- **Surface Viewer** - Generates a 3-D surface from two input variables and the output of an FIS.



FIS Editor: tipper

FIS Name: tipper
FIS Type: mamdani
And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid
Current Variable
Name: service
Type: input
Range: [0 10]
System "tipper": 2 inputs, 1 output, and 3 rules

Membership Function Editor: tipper

Current Variable
Name: service
Type: input
Range: [0 10]
Display Range: [0 10]
Current Membership Function (click on MF to select)
Ready

Rule Editor: tipper

1. If (service is poor) or (food is rancid) then (tip is cheap) (1)
2. If (service is good) then (tip is average) (1)
3. If (service is excellent) or (food is delicious) then (tip is generous) (1)

Connection: ● or ○ and
Weight: 1

Rule Viewer: tipper

Input: [5 5]
Plot points: 101
Opened system tipper, 3 rules

Surface Viewer: tipper

X (input): service
Y (input): food
Z (output): tip
X grids: 15
Y grids: 15
Ref. Input:
Ready

Figure B.4.4: All the Windows used for defining FIS Along with Input Output Relationship.

APPENDIX C

COMPONENTS USED

SEMICONDUCTORS

SYMBOL	QUANTITY	COMPONENT NAME	COST (in Rs)
IC	1	LM358,Comparator	20
D	4	1N4007,Rectifier Diode	4
D	3	Red LED	3
D	1	Photo Diode	10
Q	2	BC557,pnp Transistor	2

CAPACITOR (25 volt, electrolytic type)

SYMBOL	QUANTITY	COMPONENT NAME	COST (in Rs)
C	2	1 µF, Capacitor	4
C	3	10 µF, Capacitor	6
C	2	100 µF, Capacitor	2

MISCELLANEOUS

SYMBOL	QUANTITY	COMPONENT NAME	COST (in Rs)
MC	1	PIC 16F877A	190
IR	1	IR Sensor	20
LCD	1	JHD162A,LCD	130

RESISTOR (All 0.25 watt, 5% carbon)

SYMBOL	QUANTITY	COMPONENT NAME	COST (in Rs)
R	2	100 Ω, Resistor	2
R	2	22K, Resistor	2
R	3	1K, Variable Resistor	30
R	1	10K, Variable Resistor	10
R	2	56K, Resistor	2

APPENDIX D

D.1 Project Code

Code: Filter Design Code

```
#include<htc.h>

#include<math.h>

#include        <stdlib.h>

__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_OFF & BOREN_OFF &
LVP_OFF & WRT_OFF & DEBUG_OFF & CP_ON);

#define _XTAL_FREQ 20000000

#define IR_ON   RA2

#define RED_ON  RA1

#define rs              RD7

#define en              RD6

#define START           RD5

//LCD Data pins

#define data_lcd PORTB

unsigned int ones, hundreds, tens,i=0,pulserate,
pulsecount,r_red,r_ir,rfactor,adc_data,a,b,c,d,spo2;

void delay_refresh();

void countpulse();

void cmd( );
void data( );
int ratio();

void delay_refresh(){
 __delay_ms(5);
}
void countpulse(){
    int i=0,temp ,n=20;
    IR_ON=1;
    RED_ON=1;
    RED_ON=0;
    TMR0=0;
            for(i=0;i<20;i++)
                {

                temp=n;
                ones = temp % 10;
        temp = temp / 10;
        tens = temp % 10;

                data_lcd = 0x89;
                cmd();

                data_lcd=tens+'0';

                data();

                data_lcd=ones+'0';

                data();



                data_lcd='l';

                data();

                data_lcd = 'e';

                data();

                data_lcd='f';

                data();

                data_lcd = 't';

                data();

                __delay_ms(1000);  // Delay 1 Sec

                n--;

                }



                RED_ON=1;

                pulsecount = TMR0;

                pulserate = pulsecount*3;
```

```
        r_red=ratio();

        IR_ON=1;

        RED_ON=1;

        IR_ON=0;

        r_ir=ratio();

        r_red=rr_red*100;;

        rfactor=r_red / r_ir;

        spo2=(10000-12*rfactor)/10;

        a = spo2 % 10;

    spo2 = spo2 / 10;

    b = spo2 % 10;

        spo2 = spo2 / 10;

        c = spo2 % 10;

    d = spo2 / 10;

        data_lcd = 0xC5;          // SETTING THE CURSER TO 2 R,5C POSITION;

        cmd();

        data_lcd=d+'0';

        data();

        data_lcd=c+'0';

        data();

        data_lcd=b+'0';
```

```
                                        data();

                                        data_lcd='.';

                                        data();

                                        data_lcd=a+'0';

                                        data();

}

void cmd()

{

                    rs=0;

                    en=1;

                    __delay_ms(10);

                    en=0;

                    __delay_ms(10);

          }

void data()

{

                    rs=1;

                    en=1;

                    __delay_ms(10);

                    en=0;

                    __delay_ms(10);
```

```
}

int ratio()

{

            int pulsetime,sampletime,i,rms=0,avg=0,ratioval,rat;

            pulsetime=60000/pulserate;

            sampletime=pulsetime/20;

            for(i=0;i<sampletime;i++)

            {

                  ADCON0 = 0B00101101;

            ADCON1 = 0B10001110;

                  GO_DONE==1;                       //Start Conversion

          while(GO_DONE==1);;

          //Wait here until Conversion Complete

          adc_data = ADRESH ;

          adc_data = adc_data<<8;

          adc_data = adc_data | ADRESL;


          adc_data = adc_data*4.88/100;

                  rms=rms + adc_data*adc_data;

                  avg=avg + adc_data;

                  __delay_ms(20);
}

            rms=rms/sampletime;
            rms=sqrt(rms);
            avg=avg/sampletime;
            rms=rms*sampletime;
            ratioval=rms/avg;
            rat=ratioval;
            a = ratioval % 10;
            ratioval = ratioval / 10;
            b = ratioval % 10;
            ratioval = ratioval / 10;
            c = ratioval % 10;
            d = ratioval / 10;
            data_lcd = 0xC5;
            cmd();
            data_lcd=d+'0';
            data();
            data_lcd=c+'0';
            data();
            data_lcd=b+'0';
```

```
                data();

                data_lcd=a+'0';

                data();

                return rat;

}


void main() {

    CCP1CON = 0x07;                 // Disable Comparators

    TRISA = 0b00110001;             // RA4/T0CKI input, RA5 is I/P
only

    TRISB = 0x00;

    TRISD=0b00100000;;                  // RB5 input, rest output

    OPTION_REG = 0b00101000;        // Prescaler (1:1), TOCS =1 for
counter mode

    pulserate = 0;

    IR_ON=1;

    RED_ON=1;

    //Configuring the A/D Register's Inside PIC18F4550

  ADCON0 = 0B01000101;

  //Channel 3 &1  is Selected i.e RA0 and RA3 pin.

  //Anlog to Digital Converter Module gets on.

  ADCON1 = 0B10001110;
```

```
//Setting all Channels as Analog Channels

                data_lcd = 0x38;

                cmd();

                data_lcd = 0x0C;

                cmd();

                data_lcd = 0x80;

                cmd();

                data_lcd = 'H';

                data();

                data_lcd='e';

                data();

                data_lcd = 'a';

                data();

                data_lcd='r';

                data();

                data_lcd = 't';

                data();

                data_lcd='r';

                data();

                data_lcd = 'a';

                data();
```

```
        data_lcd='t';
        data();
        data_lcd='e';
        data();
        data_lcd = 0xC0;
        cmd();
        data_lcd = 's';
        data();
        data_lcd='P';
        data();
        data_lcd = 'O';
        data();
        data_lcd = '2';
        data();
        data_lcd = 0xCA;
        cmd();
        data_lcd='%';
        data();
    while(1)
{
                if(START == 0)
        {
        countpulse();
        IR_ON=1;
        RED_ON=1;
        __delay_ms(100);
        ones = pulserate % 10;
pulserate = pulserate / 10;
tens = pulserate % 10;
hundreds = pulserate / 10;
        data_lcd = 0x89;
        cmd();
        data_lcd=' ';
        data();
        data_lcd=hundreds+'0';
        data();
        data_lcd=tens+'0';
        data();
        data_lcd=ones+'0';
        data();
        data_lcd = 'b';
        data();
```

```
                        data_lcd='p';

                        data();

                        data_lcd = 'm';

                        data();

            }

}

}
```

D.2 Filter Design Code

Code: Filter Design Code

```
b = [0 1 0];                    // coefficients of the numerator of the transfer function
or zeroes
a = [1 3.8 2.4];                     // coefficients of the denominator of the
transfer function or poles
w = logspace (-10, 10);               // set the frequency scale as logarithmic
freqs (b,a,w)
h = freqs (b, a, w);
mag = abs(h);
phase = angle(h);
phasedeg = phase*180/pi;

subplot(2,1,1), loglog(w,mag), grid on                // plot magnitude
xlabel 'Frequency (rad/s)', ylabel Magnitude          // plot phase
subplot(2,1,2), semilogx(w,phasedeg), grid on
xlabel 'Frequency (rad/s)', ylabel 'Phase (degrees)'
```

D.3 DESIGNED PCB FOR THE PROJECT

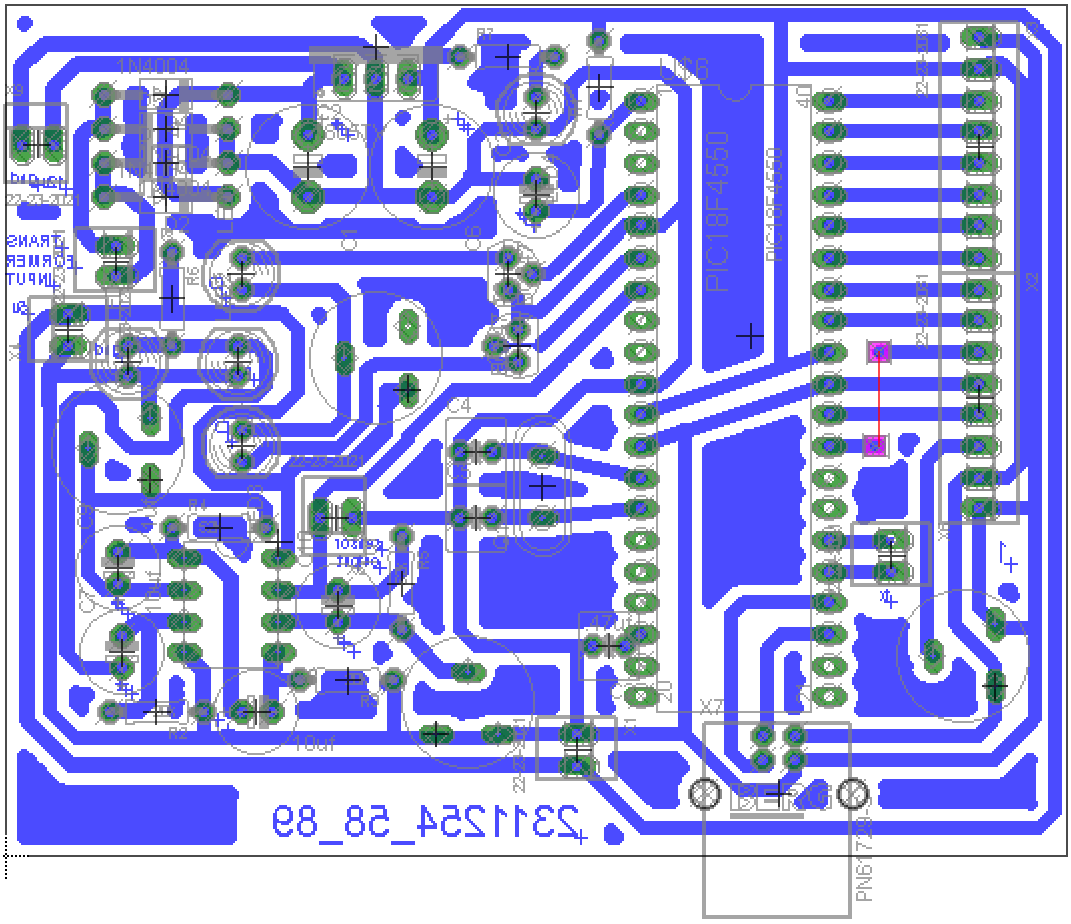


Figure D.4.1 Project PCB