TRIBHUVAN UNIVERSITY

Institute of Engineering

Department of Electronics and Computer Engineering

"PULSE OXIMETER"

By

Ranju Pant (059/BEX/431)

Samita Neupane (059/BEX/434)

Shishir Gautam (059/BEX/439)

Suveksha Thapa (059/BEX/446)

A PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE BACHELOR IN ELECTRONICS AND COMMUNICATION ENGINEERING

Lalitpur, Nepal

December 2006

TRIBHUVAN UNIVERSITY

Institute of Engineering

Department of Electronics and Computer Engineering

CERTIFICATE

We have pleasure in forwarding the project work of Ms. Ranju Pant, Ms. Samita Neupane, Mr. Shishir Gautam and Ms. Suveksha Thapa "**PULSE OXIMETER**" for the award of degree of Bachelor of Engineering in Electronics and Comunication of the Institute of Engineering, Pulchowk.

Ms. Ranju Pant, Ms. Samita Neupane, Mr. Shishir Gautam and Ms. Suveksha Thapa have completed the project work for full-prescribed period under Bachelor of Engineering in Electronics and Comunication and research project embodied the result of the investigation conducted during the period they worked as fulltime student of this department.

……………………….

Bikram Dangol

Supervisor

………………………

Bikash Shrestha

External Examiner

………………………

Assoc. Prof. M.N Ghimire

Internal Examiner

……………………………

Asst. Prof. R.L Rajbhandari

Project Coordinator

ACKNOWLEDMENT

We are very much thankful to the Department of Electronics and Computer Engineering, Pulchowk Campus for accepting our project proposal on "**Pulse Oximeter**" and granting us the opportunity of the work undertaken. We express our heart-felt gratitude for providing us with all the essential cooperation, valuable suggestion and the necessary resources during all the quarters of the completion of the project.

We are extremely grateful to our project coordinator, Mr. R.L Rajbhandari, and to our supervisor Mr. Bikram Dangol for their invaluable support and genuine guidance in all the happenings regarding the project. The constructive criticism is very appreciated.

Our sincere appreciation is extended to all staffs of the department for genuine cooperation. Finally, we express our earnest gratitude towards all the colleagues and teachers for the encouragement through potential discussions and suggestions.

Ranju Pant
Samita Neupane
Shishir Gautam
Suveksha Thapa

ABSTRACT

Health has been one of the most significant considerations in the fortification of human civilization. A lot of resource has been used in health sector to aid mankind fight diseases of all kinds. Different researches are ongoing for development of sophisticated high precision devices which make medical services reliable and accurate.

Pulse oximeter is one of such device popularly used in hospitals to monitor the saturation of oxygen concentration in blood cells for patients having risk of respiratory failure. Hemoglobin the colored substance in blood is a carrier of oxygen. It combines with oxygen to form oxidized hemoglobin. The absorption of light by hemoglobin varies with oxygen concentration. Hence by measuring the light transmitted (or calculating the attenuation of light) through the fingertip (or the earlobe) at two different wavelengths, one in the red and the other in the near infrared part of the spectrum, the oxygen saturation of the arterial blood in the finger (or ear) can be determined.

This project implements the theory of pulse oximetry using a sensor that senses the amount of light transmitted through the blood cells. Filters and amplifiers are used to attenuate unwanted noise and dc level and amplify the desired portion of the signal. Analog to digital converter is used to digitize the signal which will be fed to the microcontroller to calculate oxygen saturation in the blood through the conventional formulas developed. Finally the result i.e. the oxygen saturation will be displayed on LCD which is interfaced via the microcontroller.

For patients under intensive care pulse oximeter is indispensable as it is very important to monitor oxygen concentration in their blood in a continuous basis. In this project we have tried to implement all our knowledge gained to develop a system having high practical significance.

LIST OF FIGURES

LIST OF ABBREVIATIONS

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1Background

With the advancement of science new and sophisticated technologies have evolved which play a critical role in making medical services reliable and accurate. Biomedical instrumentation is an ever evolving field with a scope of development of high precision equipments that will help save lives of many people. Pulse oximetry is one such theory employed to develop a sensor capable of monitoring saturation of oxygen in blood cells. The principal advantage of optical sensors for medical applications is their intrinsic safety as there is no electrical contact between the patient and the equipment.

For patients at risk of respiratory failure, it is important to monitor the efficiency of gas exchange in the lungs, *i.e.* how well the arterial blood is oxygenated (as opposed to whether or not air is going in and out of the lungs). Preferably, such information should be available to the doctors in a continuous basis (rather than every few hours). Both of these requirements can be met non-invasively with the technology of *pulse oximetry*.

The project work undertaken here tries to explore one of the aspects of biomedical instrumentation i.e. to design a system capable of calculating the concentration of oxygen designated as SpO_2 in the arterial blood. Here we have tried to design an Embedded System to calculate SpO_2 .The main device used in our system is the transducer provided with a Red and IR Led on the transmitting side and a shared photodiode on the receiving side. The Red and the IR light that is transmitted through the finger is sensed by the photodiode. On calculating the ratio of light i.e. RED/IR it gives the corresponding value of oxygen saturation in blood. The main principle behind selection of RED and IR LED is the oxygen rich blood absorbs IR light while deoxygenated blood absorbs RED light to a greater extent. The varying amount of Red and IR light that is finally transmitted by through the finger helps to determine the concentration of oxygen in blood.

As the signal obtained by the photodiode is analog in nature it needs to be digitized before it can be processed by the microcontroller for further calculations. The obtained pulsatile signal is amplified and passed through low pass filter to remove noise and undesired signal components. Use of analog to digital converter digitizes the signal received and hence the microcontroller calculates the ratio of two lights transmitted through the finger. The microcontroller is programmed to calculate the corresponding value of SpO2. The final output will be displayed on the LCD (Liquid crystal display)

The important factors to be considered during calculation of SpO_2 are the amount of light received by the photodetector depends upon the attenuation of light by veins and tissues and also in the placement of the receiver with respect to the transmitter. The signal thus obtained is of low frequency and low amplitude so it needs to be amplified appropriately before it can be processed by the analog to digital converter. Lack of careful consideration of these factors may lead to error in our calculation. Here we have made use of microcontroller to process the signal received.

1.2 OBJECTIVES

The project aspires to attain the following objectives:

- To design an embedded system capable of determining the oxygen concentration in blood.
- To design cost-effective system.
- To familiarize the use optical sensors in biomedical instrumentation to implement a non invasive technique
- To enhance the programming skills with regard to SDCC
- To help familiarize the hardware realization of various signal processing stages
- To attain the mandatory award in the partial fulfilment of the requirement for the degree of Bachelor of Electronics and Communication Engineering.

1.3 LITERATURE REVIEW

1.3.1 Theory behind pulse oximetry

Haemoglobin a coloured substance in blood is a carrier of oxygen. The absorption of visible light by a haemoglobin solution varies with oxygenation. This is because the two common forms of the molecule, oxidized haemoglobin (HbO_2) and reduced haemoglobin (Hb) have significantly different optical spectra in the wavelength range from 500nm to 1000nm, as shown in Fig 1.3.1.

The oxygen chemically combined with haemoglobin inside the red blood cells makes up nearly all of the oxygen present in the blood (there is also a very small amount which is dissolved in the plasma). Oxygen saturation, which is often referred to as SaO_2 or SpO_2, is defined as the ratio of oxyhaemoglobin (HbO_2) to the total concentration of haemoglobin present in the blood (*i.e.* oxyhaemoglobin + reduced haemoglobin):

$$SpO2= \ HbO2/ [Total\ Hemoglobin]$$

Arterial SaO_2 is a parameter measured with oximetry and is normally expressed in percentage. Under normal physiological conditions arterial blood is 97% saturated, whilst venous blood is 75% saturated. It is possible to use the difference in absorption spectra of HbO_2 and Hb for the measurement of arterial oxygen saturation because the wavelength range between 600 nm and 1000 nm is also the range for which the is least attenuation of light by body tissues.

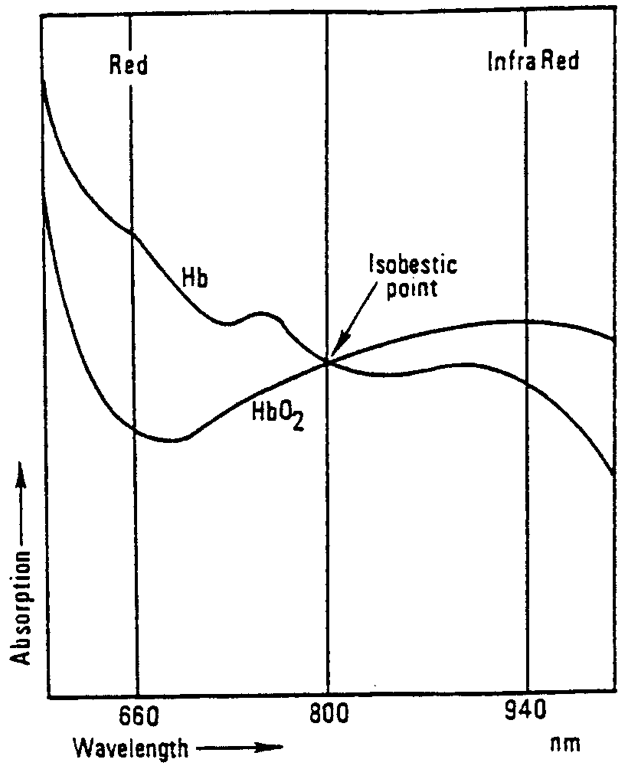


Fig: 1.3.1 Absorption spectra of oxygenated and deoxygenated blood

It is the recent development of *pulse* oximetry which has led to oximetry being accepted as a useful noninvasive technique for the measurement of arterial SaO_2. With pulse oximetry, only that part of the signal directly related to the inflow of arterial blood into the body segment is used for the calculation of oxygen saturation. The intensity of light transmitted across the fingertip, for example, varies as shown in Fig 1.1. A pulsatile signal, which varies in time with the heart beat, is superimposed on a d.c. level. Pulse oximetry assumes that the attenuation of light by the body segment can be split into the three independent components shown in Fig 1.3.2 (a): arterial blood, venous blood and tissues.

If we assume that the *increase* in attenuation of light is caused only by the inflow of arterial blood into the fingertip, we can calculate the oxygen saturation of the arterial blood by subtracting the d.c. component of the attenuation from the total attenuation, leaving only the cardiac synchronous pulsatile component for the dual wavelength determination of oxygen saturation.

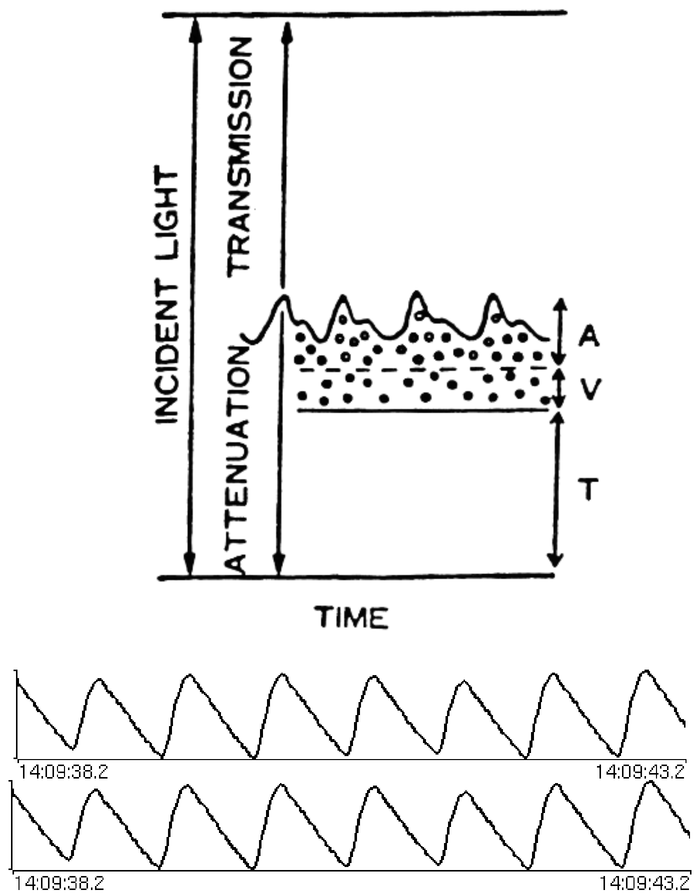


Figure1.3.2: (a) Transmission of light through the finger when the attenuation of light is caused by arterial blood (A), venous blood (V) and tissues (T). (b) and (c) show typical pulsatile signals detected in the intensity of detected light when light is shone through a finger.

.

1.3.2 Calibration of Pulse Oximeter

The first pulse oximeters, which were manufactured in the early 1980’s, used the Beer Lambert model to calculate SpO_2. However, the Beer Lambert law does not take into account the multiple scattering of light by the red blood cells. Although oximetry is a differential technique, the effect of scattering is only partially compensated for since scattering is wavelength dependent. A more efficient technique is to use the look up table derived from calibration studies on large numbers of healthy volunteers whose oxygen saturation is also measured invasively. Fig 1.3.2.1 shows two relationships, one using the Beer Lambert law and the other based on empirical data, between the ratio *R* and the oxygen saturation of the patient. Consequently, instruments based on the Beer Lambert law tended to give erroneous estimates of the true value of oxygen saturation (especially for SpO_2 values below 85%).

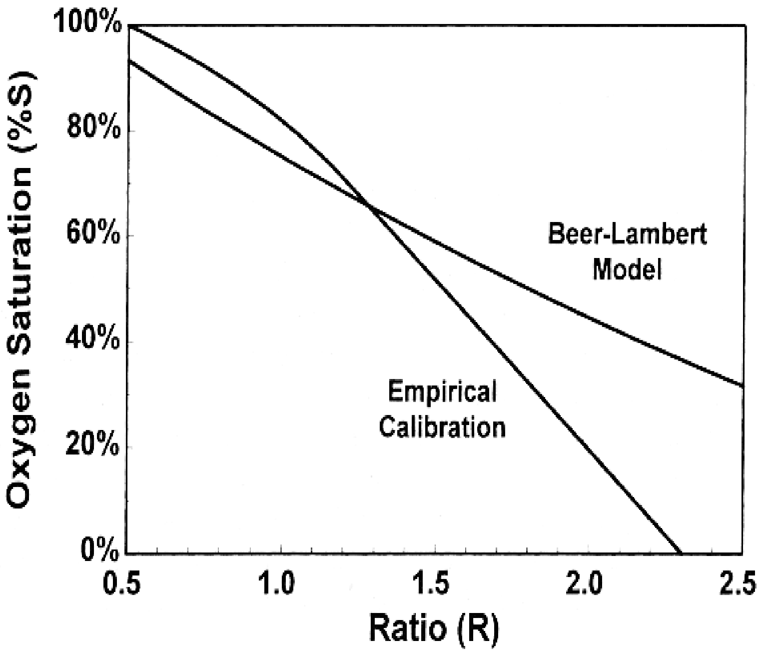


Figure 1.3: Two relationships between the ratio R and the Oxygen Saturation of the patient.

Oxygen saturation percent is calculated with each pulse detected, and thus the display is continually updated. SpO2 is generally measured via pulses detected using a finger sensor. However, for certain situations SpO2 may be measured at alternate sites including the earlobe, forehead and toes. Special sensors must be employed in these situations.
The minimum limit of the sensed SPO2 is taken as 40 and the maximum limit as 99.

1.4 OUTLINE OF THE REPORT

The report has been compiled in an organized manner so that it can be read linearly from front to back. Each chapter introduces a number of related concepts. The report initially gives an overview of the project with introduction to the related theoretical background. Further the report elaborates on the design and implementation of the algorithms used. Finally, it puts forward suggestions and prospects for the attainment of future enhancements.

Chapter 1 introduces the basic idea behind the commencement of the project and outlines the benefit it ensues. Chapter 2 gives an overview of the used to implement the system. Chapter 3 explains about the components used in our system and chapter 4 describes the overall design of the system.

CHAPTER 2
SDCC and ESD

2.2 ESD (Embedded System Design)

We are living in the Embedded World. We are surrounded by with many embedded products and our daily life depends largely on the proper functioning of these gadgets. Television, Radio, CD player of the living room, Microwave Oven of the kitchen, Card Readers, Access Controls, Palm Devices of the work space all enable us to do many tasks effectively. Apart from all these, many controllers embedded in the car take care of car operations between the bumpers. However we tend to ignore these controllers.

In recent days, we are showered with variety of information about these embedded controllers in many places. All kinds of magazines and journals regularly dish out details about recent technologies, new devices, fast applications which makes us believe that the basic survival is controlled by these embedded products. We all agree that the embedded products have successfully invaded our world.

But what really is this Embedded System???

Embedded systems are the systems that do one specialized task efficiently and in cost effective manner. The desktop computers that we use are designed to serve many purposes and applications. In contrast, embedded controllers carryout a specific work for which they are designed. Most of the time, engineers design these embedded controllers with a specific goal in mind. So they cannot be used in other places.

Embedded systems are found in a variety of common electronic devices, such as:

(a) Consumer Electronics -- cell phones, pagers, digital cameras, camcorders, videocassette recorders, portable video games, calculators, and personal digital assistants;
(b) Home Appliances -- microwave ovens, answering machines, thermostat, home security, washing machines, and lighting systems;
(c) Office Automation -- fax machines, copiers, printers, and scanners;
(d) Business Equipment -- cash registers, curbside check-in, alarm systems, card readers, product scanners, and automated teller machines;
(e) Automobiles -- transmission control, cruise control, fuel injection, anti-lock brakes and active suspension. One might say that nearly any device that runs on electricity either already has, or will soon have, a computing system embedded within it.

Theoretically, an embedded controller is a combination of a piece of microprocessor based hardware and a suitable software to undertake a suitable task. But today we have choices between a microprocessor and a microcontroller. Selecting a right microprocessor or microcontroller may turn out as a most difficult first step and is getting complicated as more devices continue to pop-up more often.

Any embedded system's functionality consists of three aspects: processing, storage, and communication. Processing is the transformation of data, storage is the retention of data for later use, and communication is the transfer of data. Each of these aspects must be implemented. We use *processors* to implement processing, *memories* to implement storage, and *buses* to implement communication.

Embedded systems have several common characteristics:

1) *Single-functioned:* An embedded system usually executes only one program, repeatedly. For example, a pager is always a pager. In contrast, a desktop system executes a variety of programs, like spreadsheets, word processors, and video games, with new programs added frequently.

2) *Tightly constrained:* All computing systems have constraints on design metrics, but those on embedded systems can be especially tight. A design metric is a measure of an implementation's features, such as cost, size, performance, and power. Embedded systems often must cost just a few dollars, must be sized to fit on a single chip, must perform fast enough to process data in real-time, and must consume minimum power to extend battery life or prevent the necessity of a cooling fan.

3) *Reactive and real-time:* Many embedded systems must continually react to changes in the system's environment, and must compute certain results in real time without delay. For example, a car's cruise controller continually monitors and reacts to speed and brake sensors. It must compute acceleration or decelerations amounts repeatedly within a limited time; a delayed computation result could result in a failure to maintain control of the car. In contrast, a desktop system typically focuses on computations, with relatively infrequent (from the computer's perspective) reactions to input devices. In addition, a delay in those computations, while perhaps inconvenient to the computer user, typically does not result in a system failure.

Modern design requires a designer to have a unified view of software and hardware, seeing them not as completely different domains. Three important trends have made such a unified view possible. First, integrated circuit (IC) capacities have increased to the point that both software processors and custom hardware processors now commonly coexist on a single IC. Second, quality compiler availability and average program sizes have increased to the point that C compilers (and even C++ or in some cases Java) have become commonplace in embedded systems. Third, synthesis technology has advanced to the point that synthesis tools have become commonplace in the design of digital hardware.

ESD focuses on design principles, breaking the focuses on the details a particular microprocessor and its assembly-language programming. While stressing a processor-independent high-level language approach to programming of embedded systems, it still covers enough assembly language programming to enable programming of device drivers. Such processor-independence is possible because of compiler availability.

Also there are various tools for Embedded System implementation. They are listed and explained below.

- Software Options
- Stand Alone Device Assemblers
- Stand Alone Remote Debugger
- Standalone Simulators
- In Circuit Emulators

Software Options:

- Integrated Development Environment(IDE)
- Standalone Remote Debuggers
- Standalone Simulators
- Device Assemblers

Of all these, IDE is the most sophisticated and useful tool suite that contains a kind of high level C compiler, debugging tool and other optimizing tools. We can develop our project using C language and get access into other tools like Simulators, Remote Debuggers, Optimization Tools to get a quick start into the project. However commercial systems are developed using assembly language.

The development process for embedded software follows a cycle:

1. Problem specification
2. Tool/chip selection
3. Software plan
4. Device plan
5. Code/debug
6. Test
7. Integrate

Stand Alone Device Assemblers:

These assemblers are cross assemblers using the standard PC. We can use any standard text editor to write the software program and then export it to the assembler to get the required system code, which is in Hex format.

Stand Alone Remote Debugger

This one is very important part of the development process. The debugger provides the means to look into the target code when it is getting executed in the microcontroller. It enables us to watch how the constants of the various registers, memory area and other parts of the microcontroller vary during code execution. Also we can execute a code upto a certain break point or execute the code step by step. The contents of the register or memory locations can also be viewed when desired. All these make the debugger the most important vehicle we should board for any serious development.

Standalone Simulators

These simulators act standalone and works in personal computers. They simulate the actual target microcontroller in the PC. They work in GUI and enables us to get the feel of the microcontroller without waiting for the actual and physical hardware. We can also control the program execution in many ways as executing up to a certain breakpoint, single step execution etc.

In circuit Emulators

In circuit Emulators are based on testing hardware along with debugging software. Normally they take charge of the target PCB hardware and captures all the signals at most of the microcontroller pins. The debugging part builds up required information from these signals. They are the costliest tools for any microcontroller.

Embedded Systems are really large in numbers, and those numbers are growing every year as more electronic devices gain a computational element.

2.2 SDCC (Small Device C Compiler)

SDCC (*S*mall *D*evice *C* *C*ompiler) is an open source, retargettable, optimizing ANSI-C compiler by Sandeep Dutta designed for 8 bit Microprocessors. The current version targets Intel MCS51 based Microprocessors (8031, 8032, 8051, 8052, etc.), Dallas DS80C390 variants, Freescale (formerly Motorola) HC08 and Zilog Z80 based MCUs. It can be retargeted for other microprocessors, support for Microchip PIC, Atmel AVR is under development. The entire source code for the compiler is distributed under GPL. SDCC uses ASXXXX & ASLINK, an open source retargetable assembler & linker. SDCC has extensive language extensions suitable for utilizing various microcontrollers and underlying hardware effectively.

In addition to the MCU specific optimizations SDCC also does a host of standard optimizations like:

- Global sub expression elimination,
- Loop optimizations (loop invariant, strength reduction of induction variables and loop reversing),
- Constant folding & propagation,
- Copy propagation,
- Dead code elimination
- Jump tables for switch statements.

For the back-end SDCC uses a global register allocation scheme which should be well suited for other 8 bit MCUs.

The peephole optimizer uses a rule based substitution mechanism which is MCU independent.

The data types supported are:

Type	Width	Default	Signed Range	Unsigned Range
ool	1 bit	Unsigned	-	0, 1
char	8 bits, 1 byte	Signed	-128, +127	0, +255
short	16 bits, 2 bytes	Signed	-32.768, +32.767	0, +65.535
int	16 bits, 2 bytes	Signed	-32.768, +32.767	0, +65.535
long	32 bits, 4 bytes	Signed	-2.147.483.648, +2.147.483.647	0, +4.294.967.295
float	4 bytes IEEE 754	Signed		1.175494351E-38, 3.402823466E+38
pointer	1, 2, 3 or 4 bytes	Generic		

The compiler also allows inline assembler code to be embedded anywhere in a function. In addition, routines developed in assembly can also be called. SDCC also provides an option (--cyclomatic) to report the relative complexity of a function. These functions can then be further optimized, or hand coded in assembly if needed. SDCC also comes with a companion source level debugger SDCDB, the debugger currently uses ucSim a freeware simulator for 8051 and other micro-controllers.

SDCC is not just a compiler, but a collection of tools by various developers. These include linkers, assemblers, simulators and other components. Here is a summary of some of the components. As SDCC grows to include support for other processors, other packages from various developers are included and may have their own sets of documentation.

- **sdcc - The Compiler:** This is the actual compiler, it in turn uses the c-preprocessor and invokes the assembler and linkage editor.

- **sdcpp - The C-Preprocessor:** The preprocessor is a modified version of the GNU cpp preprocessor. The C preprocessor is used to pull in #include sources, process #ifdef statements, #defines and so on.

- **2.9.3 as*xxxx*, aslink, link-*xxx* - The Assemblers and Linkage Editors:** This is retargettable assembler & linkage editor; it was developed by Alan Baldwin. John Hartman created the version for 8051, and Sandeep has made some enhancements and bug fixes for it to work properly with SDCC.

- **2.9.4 s51 - The Simulator:** S51 is a freeware, opensource simulator developed by Daniel Drotos. The simulator is built as part of the build process. It currently supports the core mcs51, the Dallas DS80C390 and the Phillips XA51 family.

- **2.9.5 sdcdb - Source Level Debugger:** SDCDB is the companion source level debugger. The current version of the debugger uses Daniel’s Simulator S51, but can be easily changed to use other simulators.

As development for other processors proceeds, this list will expand to include executables to support processors like AVR, PIC, etc.

CHAPTER 3
COMPONENTS USED

3.1 Sensor

Considering the fact that the accuracy of our system depends to a large extent on the output of the sensor, it indeed is the most important component. For our project we have used the Nellcor D-20 SPO2 sensor. It is a non- invasive and disposable optical sensor.

The use of optical sensors offers a number of advantages for: biomedical applications:

- Inherent safety
- Immunity to electrical interference
- No reference electrode required
- Optical sensing techniques can be envisaged for most analytes
- Robustness and Ease of sterilization

One of the most critical factors for ensuring reliable pulse oximetry readings is proper sensor selection and application. No single sensor is capable of monitoring all patients under all monitoring conditions. When choosing a sensor for a patient, we consider the following things

- Patient’s body weight
- Duration of use (long term, short term, spot check)
- Patient activity
- Infection control concerns

The Nellcor D-20 SPO2 sensor is intended for single use only and can measure SPO2 accurately for patients between 10-50kg.

The sensor has a seven pin D-9 male connector. Out of the seven pins two are for RED LED, two for IR, two for output and the remaining one is unused. The output is a shared photodiode detector.

The sensor is shown in Fig: 3.1

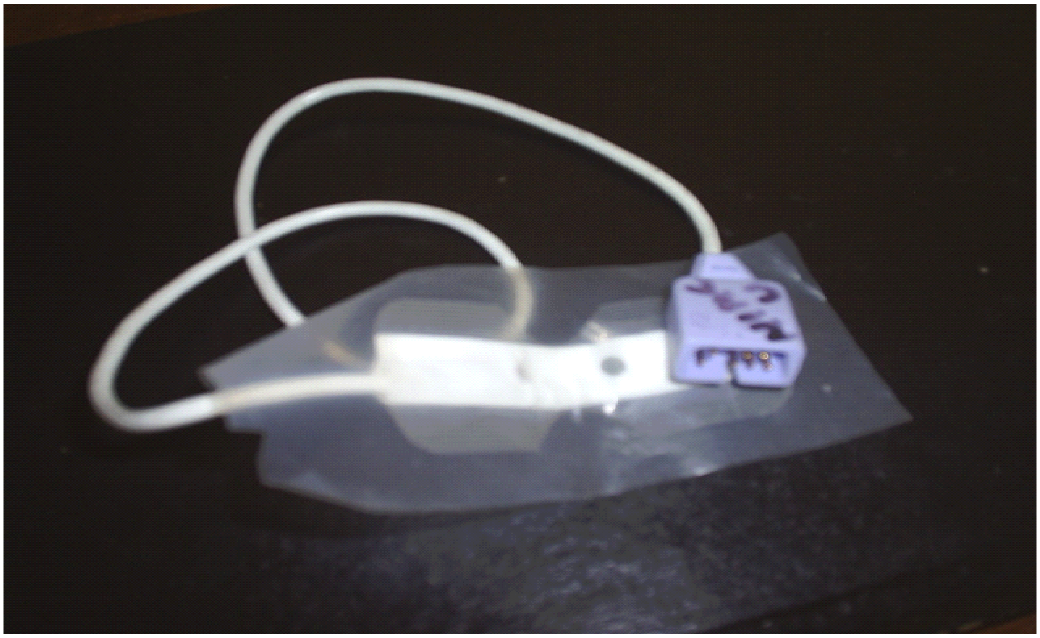


Fig 3.1.1: Nellcor D-20 Sensor

The pin configuration is shown below:

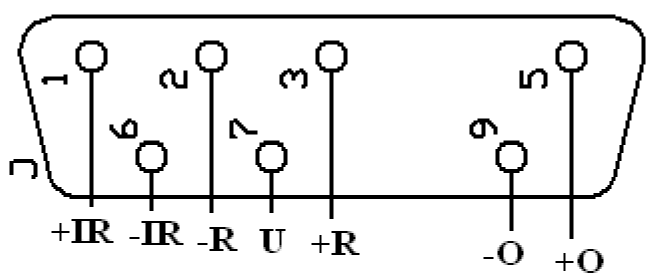


Fig 3.1.2 : Pin configuration

R = Red LED
IR = Infrared LED
O = Output
U= Unused
+ = positive terminal
- = negative terminal

Typical sensors are usually of transmission or reflectance type. The difference between the two types can be clearly seen from the Fig 3.3:

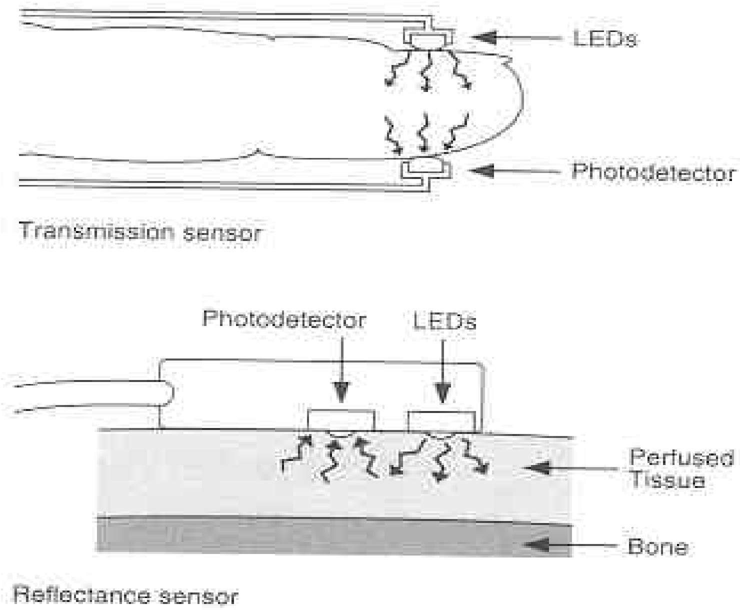


Fig 3.3 : Transmittance and Reflectance Sensor

Transmittance sensors must have the light source properly aligned with the photodetector. Reflectance sensors require proper alignment of the sensor against the surface of the skin. Our sensor is a transmittance one.

The D-20 sensor contains a dual light source and shared photodetector, which are used to measure the amount of oxygen that is combined with the hemoglobin. The source and receiver are on the opposite sides. The dual light source has a red and an infrared light for illumination. These light sources are used because each is absorbed differently by oxyhaemoglobin and deoxyhaemoglobin. The receiving end has a receiver.

In order to build a pulse oximeter several steps need to be in place. The probes need to be small and unobtrusive and this will affect how some of the steps are implemented.

ILLUMINATING:

- Small probes means that small light emitting devices need to be used.
- Red (Red (~660 nm) and Near Infra-Red (NIR, 660 nm) and Near Infra-Red (NIR, ~940 nm) 940 nm) wavelengths need to be emitted.
- Light Emitting Diodes (LEDs) are small and emit light at appropriate wavelengths.

Light Emitting Diodes (LEDs) are therefore, in principle appropriate.However, standard LEDs are not sufficiently powerful. Therefore special purpose LEDs has been designed with internal lensing to give a high intensity output.

DETECTING:

Photodiodes are the simplest solid-state optical detectors. When light falls on the p-n junction region an electron-hole pair is created. The hole and the electron are swept in opposite directions.. The resulting light current is seen as a large increase reverse current. This current needs to be turned into a voltage.

Here a single photodetector is used to receive signal from the RED and IR LED at different times.

For receiving suitable output from the sensor, we need to bias its diodes. Typically, the transmitting diodes are forward and receiving diode is reverse biased using suitable resistors.

The output of the sensor is a pulsatile signal very weak in amplitude and frequency and thus needs to be further processed.

3.2 Amplifier

Since the output of the SP02 sensor is very weak and noisy, the signal needs to be made noise free and also amplified to some extent. The output of the amplifier should be such that the ADC can easily detect and digitize it. To fulfill this requirement, we use an instrumentation amplifier INA217.

The INA217 is a low-noise, low-distortion, monolithic instrumentation amplifier. The current-feedback circuitry allows the INA217 to achieve wide bandwidth and excellent dynamic response over a wide range of gain. . Unique distortion cancellation circuitry reduces distortion to extremely low levels, even in high gain. The INA217 features differential input, low noise, and low distortion that provide superior performance. It also features wide supply voltage, excellent output voltage swing, and high output current drive.

The gain of IA can be set with a single resistor, unlike other circuits where gain adjusting involves manipulation of more than one component. Because the IA provides high SNR and high CMRR, it is found to be best for our application. It also provides isolation of patient from electric circuitry used. Thus the use of IA is advantageous from the safety point of view

FEATURES

- LOW NOISE: 1.3nV/√Hz at 1kHz
- LOW THD+N: 0.004% at 1kHz, G = 100
- WIDE BANDWIDTH: 800kHz at G = 100
- WIDE SUPPLY RANGE: ±4.5V to ±18V
- HIGH CMR: > 100dB
- GAIN SET WITH EXTERNAL RESISTOR
- DIP-8 AND SOL-16 WIDEBODY PACKAGES

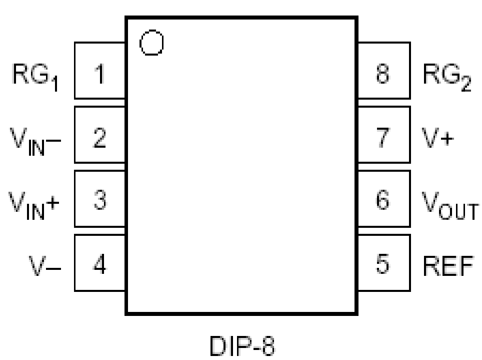


Fig: 3.2.1: Pin Configuration

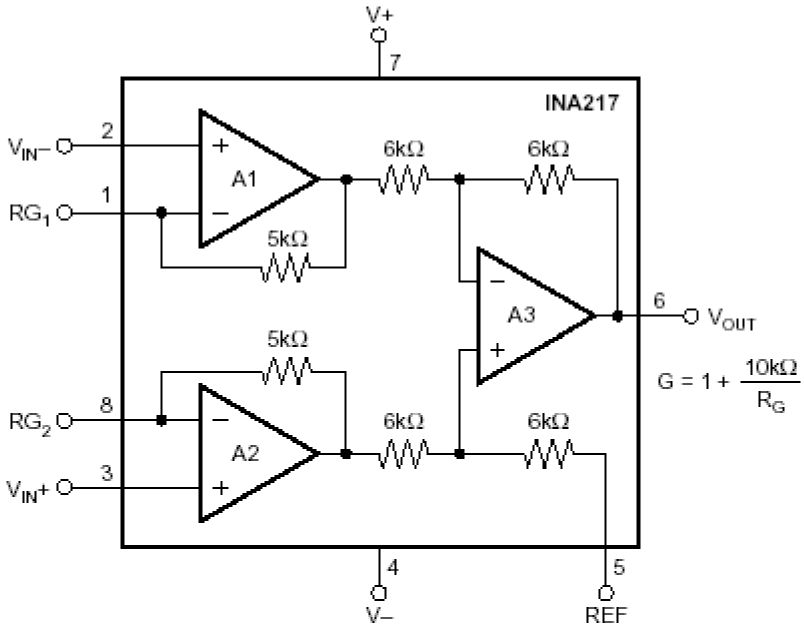
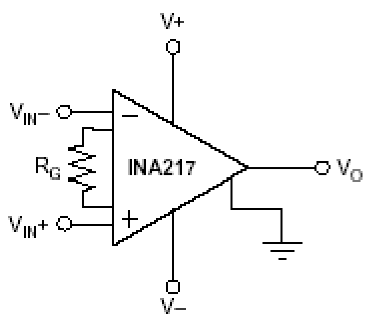


Fig 3.2.2 : Inside the IC



Gain= $1+10000/R_G$

Fig 3.2.3: Simplified diagram

For our system we have connected the output of the sensor to pin no 2. Pin no 3 and 5 are grounded and pin no 4 and 7 are used to supply suitable voltage. The output is received via pin no 6 whereas pin no 1 and 8 are used for gain adjustment.

3.3 Filter

The output of the photodiode detector is a pulsatile signal. This signal is very low in frequency. Therefore it becomes necessary to pass the signal through the LPF that passes low frequencies well, but attenuates (or reduces) frequencies higher than the cutoff frequency. Though the instrumentation amplifier reduces noise to a high extent, it is necessary to pass the signal through the LPF for better results.

One simple electrical circuit that will serve as a passive low-pass filter consists of a resistor in series with a load, and a capacitor in parallel with the load as shown in fig below . The capacitor exhibits reactance, and blocks low-frequency signals, causing them to go through the load instead. At higher frequencies the reactance drops, and the capacitor effectively functions as a short circuit. The combination of resistance and capacitance gives the time constant of the filter (T= RC). The break frequency, also called the turnover frequency or cutoff frequency (in hertz), is determined by the time constant:

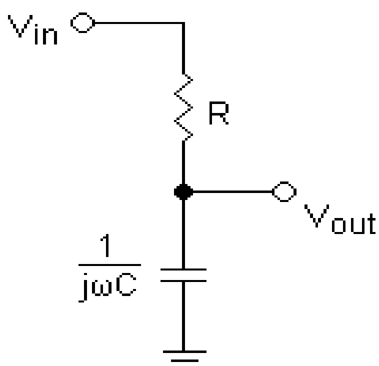


Fig 3.3.1: Low Pass Filter

Cut-off frequency $f_c=1/(2*\pi*T)=1/(2*\pi*R*C)$

An ideal low-pass filter completely eliminates all frequencies above the cut-off frequency while passing those below unchanged.However practical filters have transition region such that the frequencies ablove the cutt- off frequency are not eliminated at once but decay slowly.

In our system the LPF is implemented after the instrumentation amplifier. The output of the filter is passed to the input pin of the ADC.

3.4 Analog to Digital Converter (ADC)

Since the 89C52 microcontroller deals only with digital data, it is required to implement an ADC. The output of the filter forms the input to the ADC and the output of the ADC is connected to the microcontroller. For our project we have chosen the 0809 ADC. This ADC has total 8 inputs out of which only one is used.

The ADC0809 Data Acquisition Devices (DAD) implement on a single chip most of the elements of the standard data acquisition system. It is a monolithic CMOS device with an 8-bit analog- to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to wide range of applications including bio-medical instrumentation.

Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 VDC or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- Resolution 8 Bits
- Total Unadjusted Error ±1.2 LSB and ±1 LSB
- Single Supply 5 VDC
- Low Power 15 mW
- Conversion Time 100 μs

The basic pin configuration and functional block diagram of the ADC are shown below:

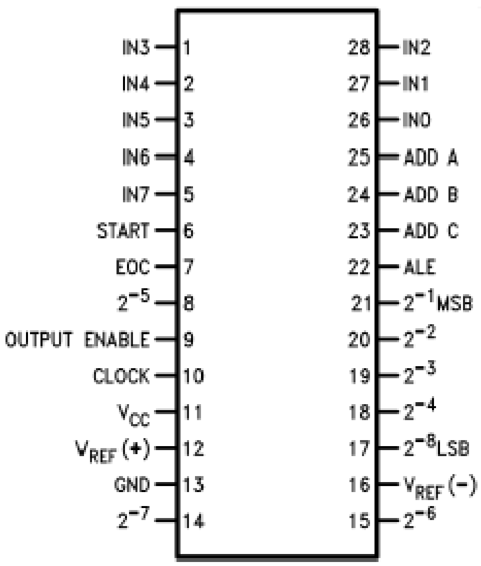


Fig 3.4.1: Pin Configuration

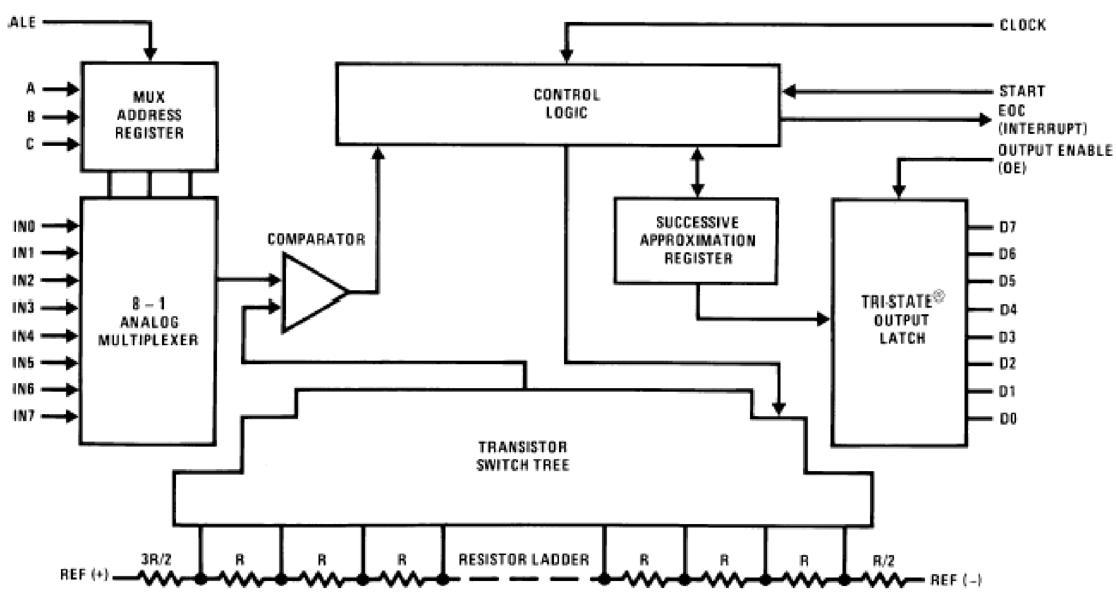


Fig 3.4.2 : Functional Block Diagram

The ADC0809, shown in the Fig 3.4.2 can be functionally divided into 2 basic sub-circuits. These two sub-circuits are an analog multiplexer and an A/D converter. The multiplexer uses 8 standard CMOS analog switches to provide for up to 8 analog inputs. The switches are selectively turned on, depending on the data latched into a 3-bit multiplexer address register.

The second function block, the successive approximation A/D converter, transforms the analog output of the multiplexer to an 8-bit digital word. The output of the multiplexer goes to one of two comparator inputs. The other input is derived from a 256R resistor ladder, which is tapped by a MOSFET transistor switch tree. The converter control logic controls the switch tree, funneling a particular tap voltage to the comparator. Based on the result of this comparison, the control logic and the successive approximation register (SAR) will decide whether the ext tap to be selected should be higher or lower than the present tap on the resistor ladder. This algorithm is executed 8 times per conversion, once every 8 clock periods, yielding a total conversion time of 64 clock periods.

When the conversion cycle is complete the resulting data is loaded into the TRI-STATE output latch. The data in the output latch can then be read by the host system any time before the end of the next conversion. The TRI-STATE capability of the latch allows easy interface to bus oriented systems.

The operation of these converters by a microprocessor or some control logic is very simple. The controlling device first selects the desired input channel. To do this, a 3-bit channel address is placed on the A, B, C input pins; and the ALE input is pulsed positively, clocking the address into the multiplexer address register. To begin the conversion, the START pin is pulsed. On the rising edge of this pulse the internal registers are cleared and on the falling edge the start conversion is initiated.

As mentioned earlier, there are 8 clock periods per approximation. Even though there is no conversion in progress the ADC0809 is still internally cycling through these 8 clock periods. A start pulse can occur any time during this cycle but the conversion will not actually begin until the converter internally cycles to the beginning of the next 8 clock period sequence. As long as the start pin is held high no conversion begins, but when the start pin is taken low the conversion will start within 8 clock periods.

The EOC output is triggered on the rising edge of the start pulse. It, too, is controlled by the 8 clock period cycle, so it will go low within 8 clock periods of the rising edge of the start pulse. Once EOC does go high this signals the interface logic that the data resulting from the conversion is ready to be read. The output enable (OE) is then raised high. This enables the TRI-STATE outputs, allowing the data to be read.

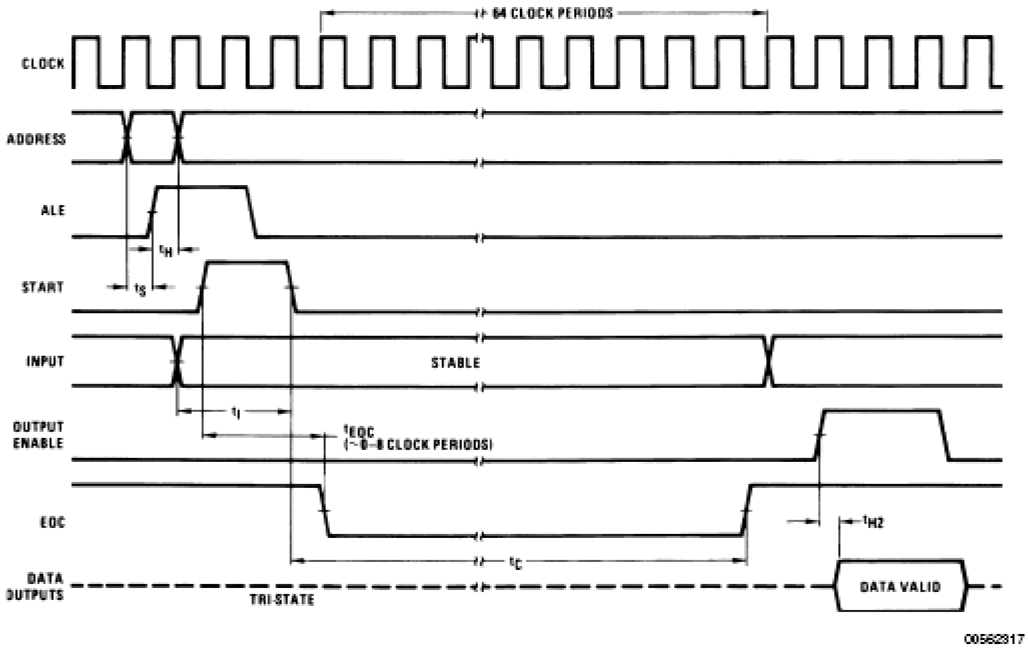


Fig 3.4.3: Timing Diagram

In our system, we have tied the ALE and START pin together and then connected it to the microcontroller. The OE is always enabled by connecting it to the Vcc and the End Of Conversion is checked to detect the completion of the conversion process. Since we are dealing only with one input, we ground the rest. Also the Address Lines are grounded. The reference voltage is provided by using a suitable resistor. The clock required for the ADC operation is provided by the microcontroller.

3.5 Liquid Crystal Display(LCD)

The LCD is used as a display unit in our project. It displays the SPO2 reading in percentage and provides a useful interface for the user. The LCD we have used is a single row 16 character display. The basic pin configuration is shown below:

Pins	Description
1	Ground
2	Vcc
3	Contrast Voltage
4	"R/S" _Instruction/Register Select
5	"R/W" _Read/Write LCD Registers
6	"E" Clock
7 - 14	Data I/O Pins

Table above shows the description for fourteen pins. But the LCD in total has 16 pins and the remaining two pins (15-16) are used for back light. However we have not used these pins.

The Ground and Vcc pins are used to supply the ground and supply voltage respectively.

The contrast pin is used to specify the contrast (or "darkness") of the characters on the LCD screen and has important significance when displaying many characters. In our system we have grounded this pin.

The "R/S" bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD. If the Bit is set, then the byte at the current LCD "Cursor" Position can be read or written. When the Bit is reset, either an instruction is being sent to the LCD or the execution status of the last instruction is read back (whether or not it has completed).

The R/W bit is used to select read or write operation. This line is pulled low in order to write commands or character data to the module or pulled high to read character or status information from the registers.

The "E" Clock is used to initiate the data transfer within the LCD.

The LCD deals with ASCII code that are send 4 or 8 bits at a time. Thus the interface is a parallel bus, allowing simple and fast reading/writing of data to and from the LCD. The transfer of data is via the Data I/O pins.

Fig 3.5.1 LCD

3.6 Microcontroller

A microcontroller is a single chip, self-contained computer which incorporates all the basic components of a personal computer on a much smaller scale. Microcontrollers are often referred to as single chip devices or single chip computers. The main consequence of the microcontroller’s small size is that its resources are far more limited than those of a desktop personal computer.

In functional terms, a microcontroller is a programmable single chip which controls a process or system. Microcontrollers are typically used as embedded controllers where they control part of a larger system. Microcontrollers are designed to be low cost solutions; therefore using them can drastically reduces part and design costs for a project.

Physically, a microcontroller is an integrated circuit with pins along each side. The pins presented by a microcontroller are used for power, ground, oscillator, I/O ports, interrupt request signals, reset and control. In contrast, the pins exposed by a microprocessor are most often memory bus signals (rather than I/O ports).

A microcontroller has seven main components:

- Central processing unit (CPU)
- ROM
- RAM
- Input and Output
- 'Timer
- Interrupt circuitry
- Buses

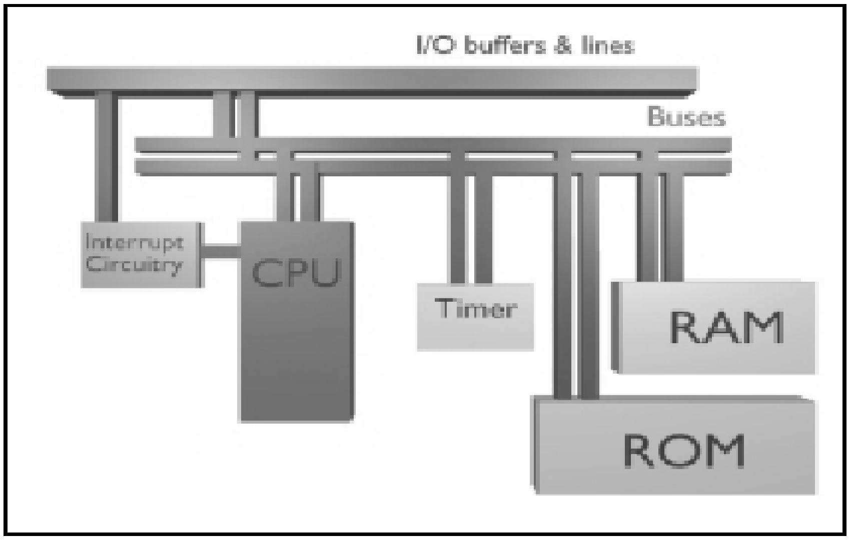


Fig 3.6.1: Block diagram of a microcontroller

Microcontroller forms the Central Control and Processing Unit of our project. It controls all the peripherals attached to it and does the entire processing task to get the desired output. Typically the sensor, ADC and LCD is interfaced to it. The software or the control program is stored in the microcontroller. Programming is done in high level C language and SDCC is used as the compiler. High-level programming definitely reduces the burden faced in assembly language programming, however the size of program is increased.

For out project, we have chosen the AT89C52 microcontroller.

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full-duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

The basic pin configuration is shown below:

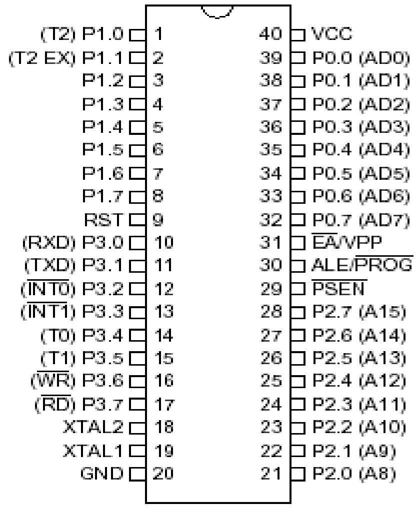


Fig 3.6.2 Pin Configuration

As can be seen, the 89C52 microcontroller has forty pins and a total four ports. Out of these four ports, our system utilizes Port 0 for ADC interfacing, Port 3 for LCD interfacing, Port 1 for controlling the ADC and LCD and finally Port 2 for interfacing the sensor. Pin no 9 is used as a RESET pin and pin no 30 is used to provide the clock for ADC operation. 3.579 MHz oscillatior is used to provide clock for the microcontoller.

CHAPTER 4
DESIGN

4.1 Block Diagram

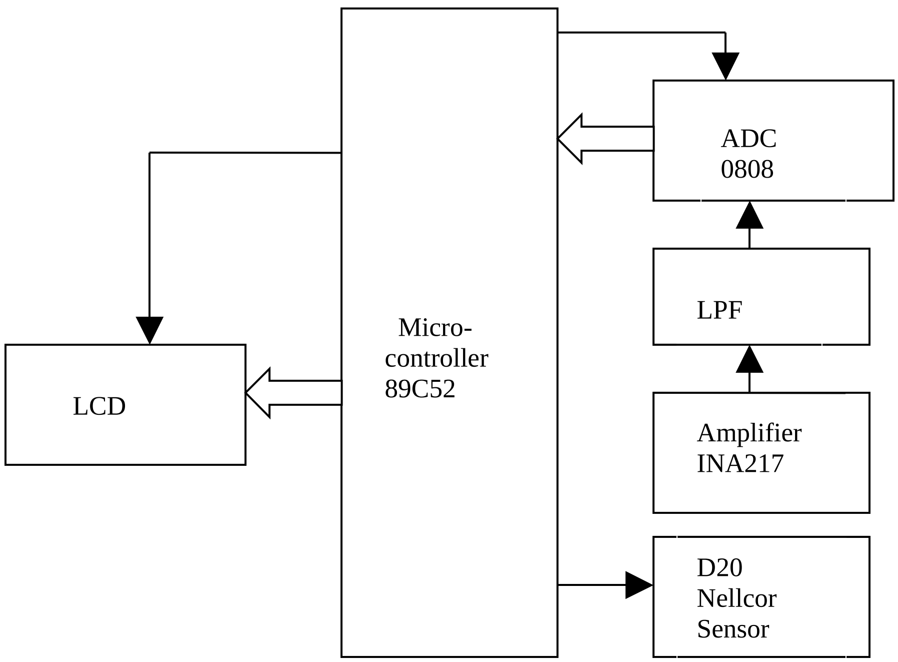


Fig 4.1.1: Block Diagram of the Designed System

The basic block diagram of the system is shown in Fig 4.1.1. The sensor generates the output voltage across the measuring site. The light sensor consists of two ends: one containing RED and IR LED and the other end containing photodetector diode that converts the intensity of received light into current. This current needs to be converted to voltage.

This signal is fed to the amplifier that amplifies the desired portion of the signal. The amplified signal is passed through a low pass filter to attenuate the high frequency signals.

This signal is fed to the ADC that converts the analog signal to its corresponding digital value. The digital value is stored in the microcontroller for the RED LED and the same process is carried out for the IR LED. The digital values for both RED and IR is used to calculate the ratio which is used to find the corresponding value of SpO2. Finally the result is displayed on the LCD.

4.2 Control Diagram

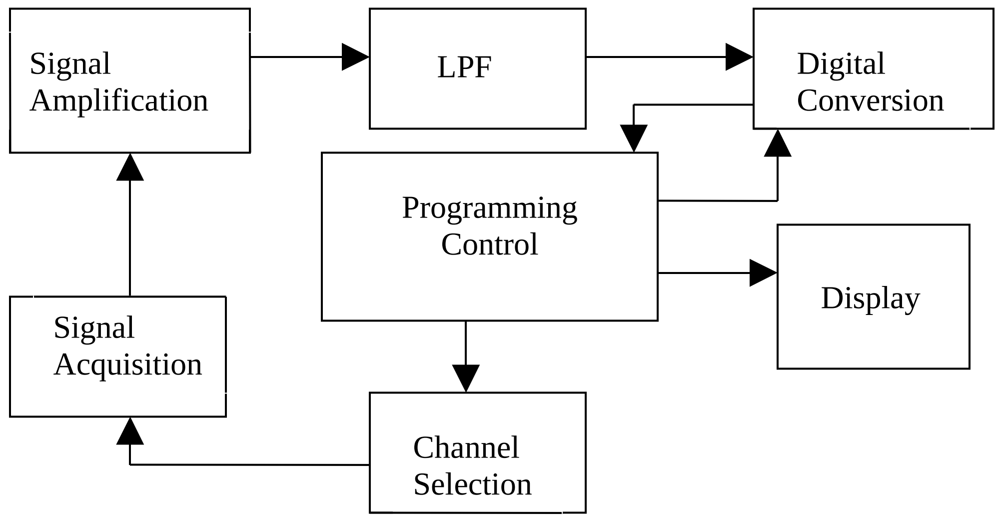


Fig 4.2.1 Control Diagram of the System

The Programming Control selects one of the two channels: Red and Infrared present in the sensor through the microcontroller. At first, the red is selected. Then the signal is passed through the instrumentation amplifier as the signal is of very low strength. This amplified signal contains high frequency components which have to be eliminated. This is achieved through the use of a low pass filter. This signal is then given to the ADC which digitizes the signal and the digital value corresponding to the Red is stored in the microcontroller. Similar process is carried for the Infrared also. The ratio of Red to infrared digitized value is then used to calculate the SPO2 value. The output is displayed on the LCD.

The overall process defined above is controlled by the program stored in the microcontroller.

4.2.1 Channel Selection

The programming control selects any of the two channels i.e. either Red or Infrared. The microcontroller transfers the control signal to the sensor. After getting the signal, one of the LED will glow.

4.2.2 Signal Acquisition

The next function to be performed is the detection of the signal generated. For this, the sensor has a photo diode that detects the signal. The light that is transmitted through the finger is attenuated by tissues and vein. Finally the amount of light that is able to be transmitted through the arteries is detected by the photo detector. After the output is detected, the signal is sent for further processing.

4.2.3 Signal Amplification

The output received by the photo diode is of very low strength. So, it needs to be amplified for further processing. In order to amplify the signal, the signal is passed through the instrumentation amplifier with desired gain.

4.2.4 Removal of unwanted Signal

To remove the unwanted high frequency signals, there is a use of low pass filter. This attenuates the high frequency components and passes only the low frequency components that are useful to the system.

4.2.5 Digital Conversion

The signal is now ready for the processing. This signal needs to be digitized so that the microcontroller can further process it. For digitizing purpose, the signal is fed to the ADC.

4.2.6 Programming Control

The programming control is done by the program stored in the microcontroller. This program controls the channel selection, ADC, calculation of SPO2 and also display to the LCD.

4.2.7 Signal Processing and Display

This is also controlled by the microcontroller. For each channel, the digital value of output voltage is stored in the microcontroller. The ratio of R and IR is calculated and the corresponding value of SPO2 is calculated. The result thus obtained is displayed on the LCD.

CHAPTER 5
IMPLEMENTATION OF DESIGN

5.1 Algorithm

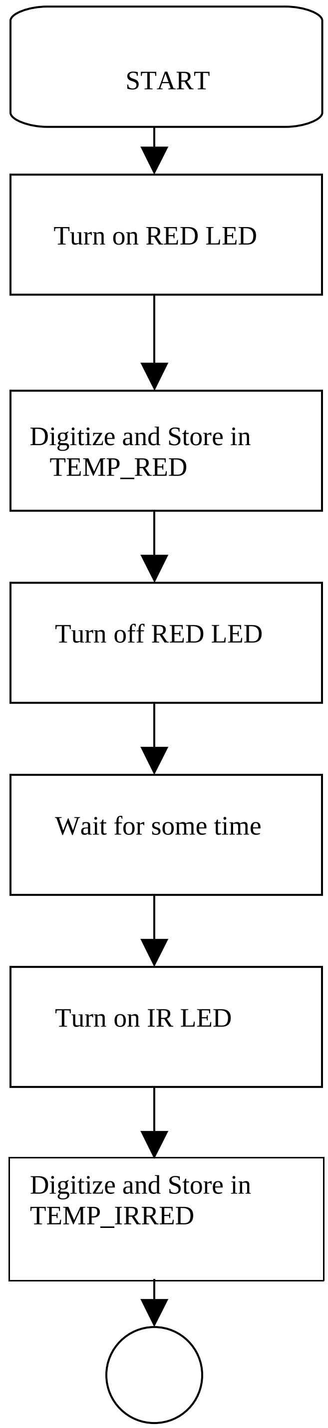
1. START
2. Turn the Red LED ON for some time.
3. Digitize and Store the output of Red LED in variable TEMP_RED.
4. Turn OFF the Red LED.
5. Wait for sometime.
6. Turn the IR LED ON for some time.
7. Digitize and Store the output of IR Led in variable TEMP_IRRED.
8. Turn OFF the IR LED.
9. Calculate the ratio y=(TEMP_RED)/TEMP_IRRED.
10. If the value of y is less than 0.50 or greater than 2.35 display the message SEARCH and go to START.
11. Else Calculate the SPO2 depending on the value of y as given below.

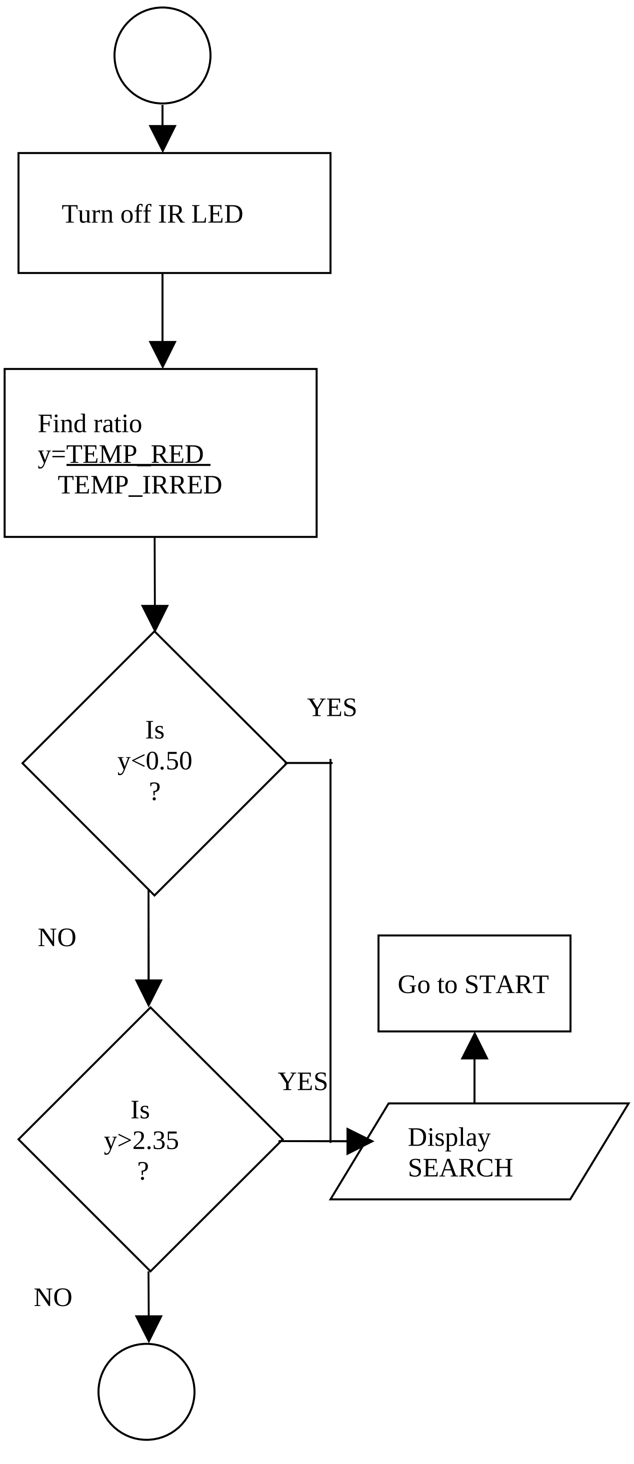
```
a)if (y>=0.50&&y<0.88)
      spo2=118.42-0.37*y

b) if(y>=0.88&&y<0.11)
      spo2=118-0.36*y
c) else
      spo2=146.64-0.62*y
```

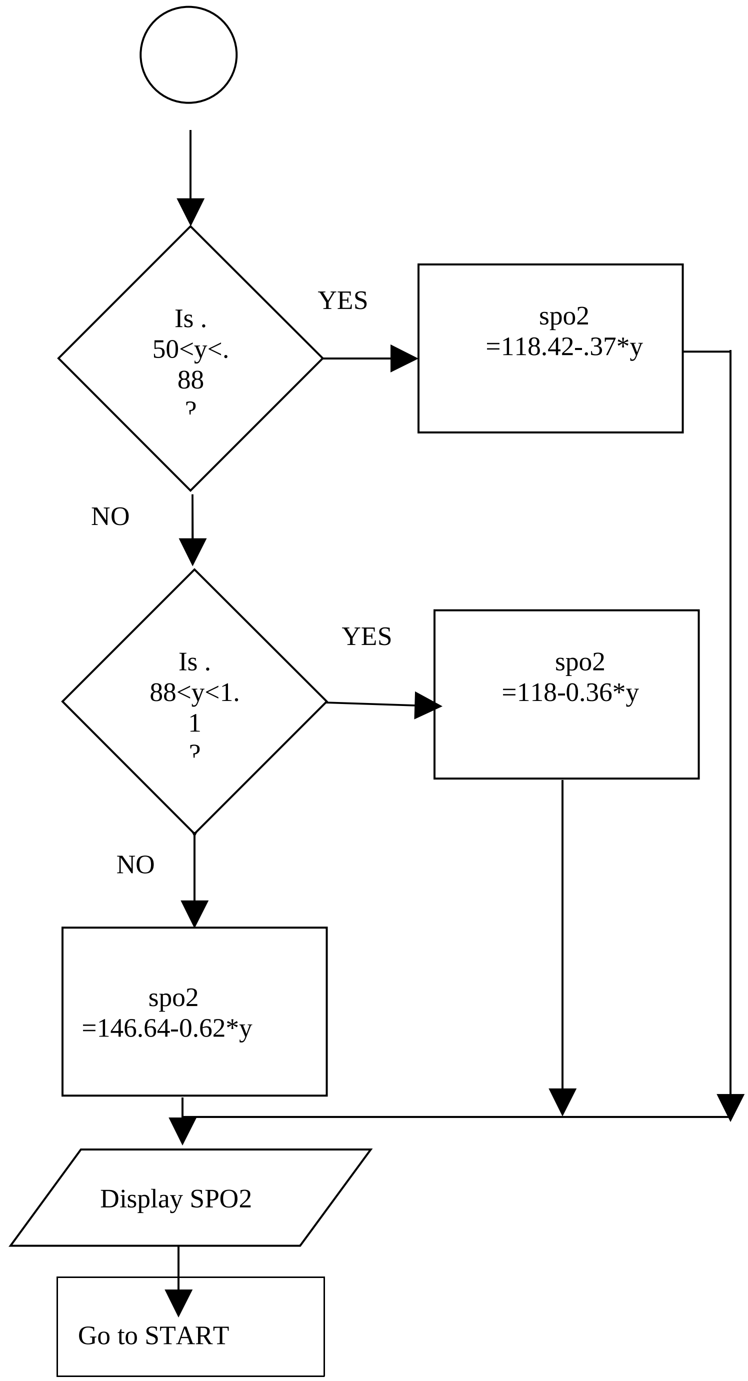
12. Display the value of SPO2.
13. Go to START

5.2 Flowchart




Turn off IR LED
Find ratio
y=TEMP_RED
TEMP_IRRED
Is
y<0.50
?
YES
NO
Is
y>2.35
?
YES
NO
Display
SEARCH
Go to START

I



Is .
50<y<.
88
?
YES
spo2
=118.42-.37*y
NO
Is .
88<y<1.
1
?
YES
spo2
=118-0.36*y
NO
spo2
=146.64-0.62*y
Display SPO2
Go to START

CHAPTER 6

EPILOGUE

6.1 LIMITATIONS

During the project development phase we encountered certain obstacles. Some of the major hindrance are listed below.

- The main problem faced during the project development phase was unavailability of the data sheet of the transducer. A lot of time was wasted in detecting the pin configuration of the sensor which proved to be a major hindrance in the project.
- Available transducer was disposable one so many experiments could not be carried out to verify our results.
- Low amplitude and frequency of the signal made it all the more difficult to process it as it was overcome by noise from the surrounding.

6.2 RECOMMENDATIONS

The team has put all its sincere efforts in the completion of the project; however there does remain certain room for future enhancements.

- Presently the system monitors the oxygen concentration in blood and displays it on the LCD. The project can be further modified to add a alarm system to indicate sudden fall in oxygen level.
- The project could be further augmented to calculate the heart beat of the patient as well.
- Use of efficient filters and amplifiers can remove noise from the system which interferes with the weak signal giving erroneous result.
- The efficiency of the project undertaken could be enhanced by using a finger clip sensor that would provide a more accurate reading.

6.3 MATERIALS REQUIRED & COST ESTIMATION

MATERIALS	COST (In NRs.)
Microcontroller AT89C52	200
ADC 0809	300
LCD	250
Amplifier INA217	
Capacitors	50
Miscellaneous	50
TOTAL	**850**

6.4 CONCLUSION

The project “Pulse Oximeter” has been completed as a mandatory task in the partial fulfillment requirement for the degree of Bachelor of Electronics and Communication Engineering.

The undertaken project, in essence was carried out as an application of biomedical instrumentation. The readily available pulse oximeter sensor was used to extract the signal from the finger .Use of microcontroller, analog to digital converters and different hardware and proper interfacing among these devices led to the development of the “Pulse Oximeter”. Different signal conditioning equipments like the filter and amplifier helped to strengthen the signal which was low in amplitude and frequency. The system thus designed was a stand alone system that can be used just by providing it with the power supply.

From the project so concluded, it is found that the results obtained was found to have some errors. This could be attributed to the interference of high frequency signals with the weak signal derived from the sensor. As the microcontroller does not support floating point operations certain manipulation had to be carried out to determine the ratio of RED/IR light. This factor limited the use of the pulse oximeter over a wide range. Some of the error in the system might be due to improper placement of the finger.

Irrespective of these errors our system works satisfactorily displaying the SpO2 level in the desired range. The team has put all its sincere efforts in developing the project; however there does remain certain room for future enhancements. Hence it can be concluded that most of the objective of the project has been achieved and we have been successful in developing the “Pulse Oximeter”.