**CHAPTER 1**

**INTRODUCTION**

**1.1 ABSTRACT**

With an improvement in technology and miniaturization of sensors, there have been attempts to utilize the new technology in various areas to improve the quality of human life. One main area of research that has seen an adoption of the technology is the healthcare sector. The people in need of healthcare services find it very expensive this is particularly true in developing countries.

As a result, this project is an attempt to solve a healthcare problem currently society is facing. The main objective of the project was to design a remote healthcare system. It’s comprised of three main parts. The first part being, detection of patient’s vitals using sensors, second for sending data to cloud storage and the last part was providing the detected data for remote viewing. Remote viewing of the data enables a doctor or guardian to monitor a patient’s health progress away from hospital premises.

The Internet of Things (IOT) concepts have been widely used to interconnect the available medical resources and offer smart, reliable, and effective healthcare service to the patients. Health monitoring for active and assisted living is one of the paradigms that can use the IOT advantages to improve the patient’s lifestyle. In this project, we have presented an IOT architecture customized for healthcare applications. The aim of the project was to come up with a Remote Health Monitoring System that can be made with locally available sensors with a view to making it affordable if it were to be mass produced.

Hence the proposed architecture collects the sensor data through PIC microcontroller and relays it to the cloud where it is processed and analysed for remote viewing. Feedback actions based on the analysed data can be sent back to the doctor or guardian through Email and/or SMS alerts in case of any emergencies.

**1.2 INTRODUCTION**

1. Human health monitoring system has gained popularity in recent years, paralleling the advances in the concept of the Internet Of Things.
2. This project presents the implementation of an inexpensive IOT-based health care monitoring system, within the framework of an assistive technology.
3. The system implementation is based on the PIC18F4520 microcontroller and it is designed for use by the elderly and people with disabilities. The model is very effective for rural areas people.
4. IOT devices produce large amount of data and information. IOT based health care services are getting better and less costly by recoding and collecting patients monitoring.

**1.3 LITERATURE SURVEY**

Patients in a coma state need to have a continuous update of Blood pressure, temperature, humidity, and urine level. Doing this manually can become almost impossible to keep updates of multiple patients at the same time. In order to address this situation, our system comes to the rescue; this system will collect the information of patients with the help of sensors. These sensors use WIFI to communicate this information to the internet. This system is powered by the Pic controller it includes a blood pressure monitoring unit and an ultrasonic sensor to check urine, temperature sensor, motion sensor, and an LCD display.

There are many continuous monitoring systems available in the medical field such as life scope vismo PVM-2703.A. LIFESCOPE VISMO PVM-2703This machine enables to monitor ECG, pulse respiration, NIBP (non -invasive blood pressure amplifier), temperature. A large touchscreen enables quick and intuitive operation. Some of the features are listed below:

• Simple operation

• Touch screen provides easy and intuitive operation

• 3 hours of continuous monitoring of battery power even though this machine provides good facilities to the health care system it contains some disadvantages too. It is very expensive. The medical staff needs to manually record the parameters for every ten minutes which can cause human errors while recording.

A number of reviews on the subject of Wireless Sensors techniques were done in the past either as part of research papers/technical reports on IOT based Health Monitoring System.

(a) First System Here, researcher designed health monitoring system using ATmega8 microcontroller with Wireless Body Area Sensor Network (WBASN). In this work, the sensors which are used here are Temperature sensor, Blood pressure sensor, Heart beat sensor. These sensors are placed on human body which are helps to monitor the health condition without disturbing the daily schedule of the patient and these health related parameters are then forwarded to physician’s server using long range wireless technology GSM. Health monitoring system consists of sensors, microcontroller, LCD display and GSM modem to transmit or receive health related data to or from the doctor. Similarly, at hospital same GSM modem is used. Hence, GSM modem helps in the establishment of network between patient’s server and doctor’s server. LCD (Liquid Crystal Display) display is providing to show the instant result to the patient. Here researcher used LM34 as temperature sensor, IR LED and red LED is used for heart rate monitoring and Pressure transducer or the sensor based on piezo-electrical material is used to measure the systolic BP and diastolic BP. Microcontroller reads data as given by the temperature sensor, blood pressure sensor and heart rate sensor and processing it gives the output in the form of digital and it gets directly display on LCD or it gets transmitted to the doctor’s server through GSM modem. This system gives exact and instant result with high accuracy which gets directly display on LCD. It takes max 4-5 sec to monitor the doctor’s server using GSM wireless technology .This system takes small amount of time to know the health condition of patient and then delivers the report to the doctor.

(b) Second System using same system, health parameters are sending by using RFID reader, Bluetooth, GSM and UMTS. This system gives facility to monitor the blood pressure of patient. The health parameter directly sends to the doctor using GSM and UTMS. Here, video guide is used. This video guide feature serves the patients age and his blood pressure correctly. This system consists of three parts: Touchpad, remote server and reading of the Tag ID and BPM. For reading the Tag ID and BPM, use a microcontroller unit (MCU) as a kernel. The client touchpad receive the blood pressure measurement (BPM) data a RFID through Bluetooth. Client touchpad send the data to the health parameter. Also, these health parameters are directly send to remote data center and remote data center to the doctor using GSM and UMTS wireless technology. Data gets transmitted in the form of the packets. This system helps to store previous data. Similarly, it takes less time to monitor the blood pressure of the patient.

(c) Third System shows the blood pressure monitoring system using microcontroller. This system includes motor control unit, Microcontroller ATmega328, LCD display. The pressure sensor is directly connected to the cuff, which is inflated or deflated via a motor and valve. ON and OFF switches of motor are controlled by the microcontroller at correct time. Due to changes in the ON and OFF switches of the motor, the wrist cuff gets inflated and deflated; this pressure is measured by the pressure sensor. Pressure sensor generates the health parameter in the analog sensor. The processing of analog sensor is done with the help of the microcontroller and gives digital output which is displayed on the LCD or on the Personal computer using RS232. Magneto resistive RAM (MRAM) stores the value of systolic and diastolic blood pressure and is directly connected to the microcontroller. Similarly, here no need have pumped the cuff by hand; all the system is controlled by the microcontroller. It is not required to calculate or observe blood pressure manually. Time consumption is very less compared to old system.

**SURVEY 1 :“Analysis and Monitoring of Coma Patients using Wearable Motion Sensor System*”.***

DESCRIPITION:

a) AUTHOR: Sneha Chowdary Koganti, Dr. H N Suma, Appaji M. Abhishek.

b) Publication: INTERNATIONAL JOURNAL OF SCIENCE AND RESEARCH (IJSR) ISSN (ONLINE): 2319-7064,VOLUME 4 ISSUE 9, SEPTEMBER 2015.

c) Concept: In this paper, Wearable Motion sensor system is used to monitor the body movements such as eye blink movement and hand movement to detect the conscious state of an individual. This system will be helpful in assisting the doctor about the health condition of the unconscious patient and alerting the doctor whenever care is required. This system will assist the doctor by giving an alarm about the health condition of the patient, when the set of vital signal recorded are out of the normal range. These results are displayed on the computer and on the Liquid crystal display (LCD).

d) Technique: Wearable Motion Sensor System.

e) Drawbacks: Sometimes due to the critical condition of the patient there will be a difficulty measuring the pulse at the finger.

**SURVEY 2:“Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound and Visual Perceptual Components”.**

DESCRIPITION:

a) AUTHOR: Charalampos N. Doukas, IliasMaglogiannis.

b) Publication: IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 15, NO. 2, MARCH 2011.

c) Concept: The main objective of this paper utilizes video, audio, and motion data captured from the patient’s body using appropriate body sensors and the surrounding environment, using overhead cameras and microphone arrays. Appropriate tracking techniques are applied to the visual perceptual component enabling the trajectory tracking of persons, while proper audio data processing and sound directionality analysis in conjunction to motion information and subject’s visual location can verify fall and indicate an emergency event.

d) Technique: Motion and sound data acquisition, human body visual tracking and sound processing and event detection.

**COMPARISON TABLE:**

**Table 1: Comparison of Existing Systems**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Authors** | **Paper Title** | **Publisher** | **Year** | **Method Used** | **Drawbacks** |
| 1 | Sneha Chowdary Koganti, Dr. H N Suma, Appaji M. Abhishek. | Analysis and Monitoring of Coma Patients using Wearable Motion Sensor System*.* | IJSR | 2015 | Based onWearable Motion Sensor System. | Sometimes due to the critical condition of the patient there will be a difficulty measuring the pulse at the finger. |
| 2 | Charalampos N. Doukas, IliasMaglogiannis. | Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound and Visual Perceptual Components. | IEEE | 2011 | Based onMotion and sound data acquisition, human body visual tracking and sound processing and event detection. | Range of Bluetooth is less. |

**1.4 NEED OF PROJECT**

* Remote health monitoring can provide useful physiological information in the home. This monitoring is useful for elderly or chronically ill patients who would like to avoid a long hospital stay. Wireless sensors are used to collect and transmit signals of interest and a processor is programmed to receive and automatically analyse the sensor signals.
* In this project, we have to choose appropriate sensors according to what you would like to detect and design algorithms to realize your detection. Examples are the detection of a temperature, monitoring heartbeat signals. Using a single parameter monitoring system an approach to a remote health monitoring system was designed that extends healthcare from the traditional clinic or hospital setting to the patient's home.
* During design the following characteristics of the future medical applications adhered:
* a) Integration with current trends in medical practices and technology,
* b) Real-time, long-term, remote monitoring, miniature, wearable sensors and long battery life of a designed device.
* c) Assistance to the elderly and chronic patients. The device should be easy to use with minimal buttons.

**1.5 AIM OF THE PROJECT**

* To develop IOT based human healthcare monitoring system for measurement of human body parameters such as temperature, systolic blood pressure, diastolic blood pressure, haemoglobin and heart beats.

**1.6 OBJECTIVES OF THE PROJECT**

* Old age patients and children with some diseases can be periodically monitored and doctors can easily understand the health status of patients time to time while at work.
* The specialist at a distance can monitor the patient’s condition so that he can save the life.
* IOT based monitoring system is used for monitoring the patient’s health condition perfectly.

**1.7 PLANNING**

**Table 2: Planning**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Month** | **Task** |
| 1 | July 2020 | * Formed the group. * Did the survey on problems related to health monitoring systems. * Found out the problems faced by the people. * Discussed different ideas with Guide related to Agriculture, Robotics, and Embedded. * We submitted 3 project ideas.  1. Next generation braille system implementation: mobile communication device for blind. 2. Landmine detection robot system using ARM. 3. IOT based human health monitoring system.  * Given the presentation on the above three project topics |
| 2 | August 2020 | * Final topic was selected: “IOT based human health monitoring system.” * Gave presentation on final topic. * Suggestions are given by the teachers. |
| 3 | September 2020 | * Literature survey * Block diagram implementation * Finalisation of components, downloading of datasheet of each component used for the project. |
| 4 | October 2020 | * Circuit diagram design. * Designed the Flowchart. |
| 5 | November 2020 | * Simulation on MPLAB IDE software. * To find the solutions for generated problems. * Layout |
| 6 | December 2020 | * PCB Design * Component mounting. |
| 7 | January 2021 | * Testing |
| 8 | February 2021 | * Real time programming. * Faults finding |
| 9 | March 2021 | * Troubleshooting and modifications if necessary. |
| 10 | April 2021 | * Preparation of report. |

**CHAPTER 2**

**HARDWARE DESIGN**

**2.1 INTRODUCTION**

**2.2 BLOCK DIAGRAM**

**POWER** **SUPPLY**

**HEART BEAT SENSOR**

**LCD DISPLAY**

**GSM/WIFI MODULE**

**TEMPERATURE SENSOR**

**PIC18F4520**

**GLUCOMETER**

**WEB SERVER**

**BLOOD PRESSURE SENSOR**

**DOCTOR/PATIENT**

**SWITCHES**

**Fig 1. Block Diagram of IOT Based Human Health Monitoring System.**

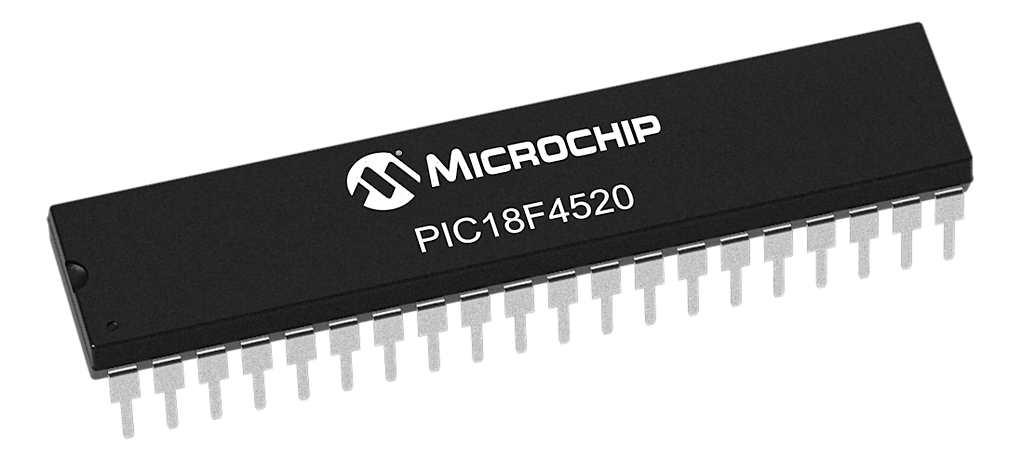
**2.2.1 BLOCK DIAGRAM DESCRIPTION**

To run the system first we need to connect microcontroller with the power supply as microcontroller is the main control unit. In input side, we have heartbeat sensor, temperature sensor and some manual buttons. On the other hand, output is shown in the LCD display. Moreover, GSM Module helps to send data in the cloud and when the data gets uploaded, we can check the output by using Laptop or Computer by log in to the server. First of all, a finger is placed in the heartbeat sensor and push button is also pressed so that the system can read data. After that, it shows result in the LCD display. Also, by pressing another push button, it can upload the output in webpage and APP and send text message through GSM module. Similar process is done with the HEART BEAT sensor. For this case, by pressing push button, data is send through GSM module and shows the ECG curve in the Web page and the APPs. This is all about the block diagram which shows the entire process of hardware.

**2.3 COMPONENTS REQUIRED**

**2.3.1 PIC MICROCONTROLLER (PIC 18F4520):**

A microcontroller is a compact integrated circuit designed to govern a specific operation in an [embedded system](http://searchenterpriselinux.techtarget.com/definition/embedded-system). A typical microcontroller includes a [processor](http://searchcio-midmarket.techtarget.com/definition/processor), [memory](http://searchmobilecomputing.techtarget.com/definition/memory) and input/output (I/O) [peripherals](http://searchmobilecomputing.techtarget.com/definition/peripheral) on a single chip. PIC microcontrollers are very popular and industrialists; this is only cause of wide availability, low cost, large user base & serial programming capability. In our project we are choosing a PIC18F4520 microcontroller because of its maximum speed, amount of RAM and sufficient number of I/O pins.

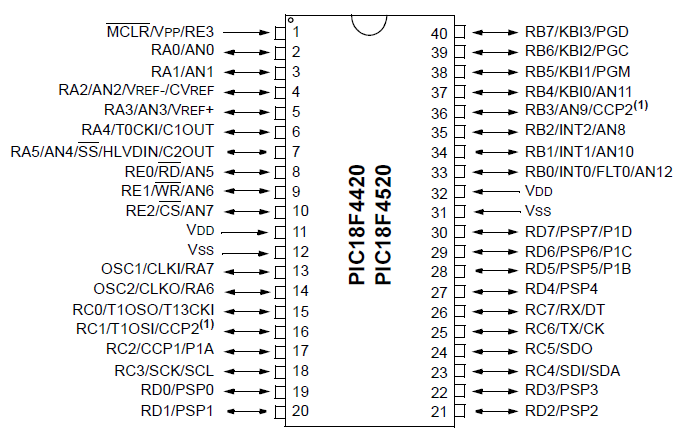
[](https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjz-Pzlu9zWAhVFQY8KHY_xDwQQjRwIBw&url=http://www.microchip.com/PIC18F4520&psig=AOvVaw23N7o3Tf8WSE-YJ_sTpzUf&ust=1507395422272622)

**Fig 2. PIC Microcontroller**

**Features:**

* Operating Frequency : DC – 40 MHZ
* Program Memory (Bytes) : 32768
* Program Memory (Instructions) : 16384
* Data Memory (Bytes) : 1536
* Data EEPROM Memory (Bytes) : 256
* Interrupt Sources : 20
* I/O Ports : Ports A, B, C, D, E
* Timers : 4
* Capture/Compare/PWM Modules : 1
* Enhanced Capture/Compare/PWM Modules : 1
* Serial Communications : MSSP, Enhanced USART
* Parallel Communications (PSP) : Yes
* 10-Bit Analog-to-Digital Module : 13 Input Channels
* Resets (and Delays) : POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
* Programmable High/Low-Voltage Detect : Yes
* Programmable Brown-out Reset : Yes
* Instruction Set : 75 Instructions, 83 with Extended Instruction Set Enabled
* Packages : 40-Pin PDIP 44-Pin QFN 44-Pin TQF

**PIN DIAGRAM:**

[](https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=&url=http://www.ebay.co.uk/itm/Microchip-PIC18F4520-DIP-40-PIN-I-P-MCU-Microcontroller-Brand-New-in-the-UK-/190784292178&psig=AOvVaw1OvxDZ5RbF9FbX19SrMZWu&ust=1507395390827001)

**Fig 3. Pin Configuration of PIC18F4520**

# 2.3.2 SIM800A Quad Band GSM/GPRS Serial Modem

# Description: SIM800A GSM Modem

# Fig 4. GSM/GPRS Modem

This GSM modem has a **SIM800A chip and RS232** interface while enables easy connection with the computer or laptop using the USB to Serial connector or to the microcontroller using the RS232 to TTL converter. Once you connect the SIM800 modem using the USB to RS232 connector, you need to find the correct COM port from the Device Manger of the USB to Serial Adapter. Then you can open Putty or any other terminal software and open an connection to that COM port at 9600 baud rate, which is the default baud rate of this modem.Once a serial connection is open through the computer or your microcontroller you can start sending the AT commands. When you send AT commands for example: "AT\r" you should receive back a reply from the SIM800 modem saying "OK" or other response depending on the command send.

SIM800 is a complete **Quad-band GSM/GPRS** solution in a LGA type which can be embedded in the customer applications. SIM800H support Quad-band 850/900/1800/1900MHz, it can transmit Voice, SMS and data information with low power consumption. With tiny size of 15.8\*17.8\*2.4 mm, it can fit into slim and compact demands of customer design. Featuring and Embedded AT, it allows total cost savings and fast time-to-market for customer applications.

**Features of SIM800A:**

* Bands: GSM 850MHz, EGSM 900MHz, DCS 1800MHz, PCS 1900MHz
* GPRS class 2/10
* Control via AT commands (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT commandset)
* Supply voltage 3.4-4.4V
* Coding schemes: CS-1, CS-2, CS-3, CS-4 Tx power: Class 4 (2W), Class 1 (1W)
* Small package: 23 \* 23 \* 3mm
* Low power: down to 1mA in sleep mode
* TCP/IP AT firmware
* Operating temperature: -40C to +85C
* Audio channels which include a microphone input and a receiver output.
* One SIM card interface.

**Modem Features:**

* High Quality Product (Not hobby grade)
* Quad-Band GSM/GPRS  850/ 900/ 1800/ 1900 MHz
* RS232 interface @ RMC Connector for direct communication with computer or MCU kit
* Configurable baud rate
* SMA connector with GSM Antenna.
* SIM Card holder.
* Built in Network Status LED
* Inbuilt Powerful TCP/IP protocol stack for internet data transfer over GPRS.
* Audio interface Connector
* Normal operation temperature: -20 °C to +55 °C
* Input Voltage: 5V-12V DC

**Interfacing with controller:**

**GSM module** is used in many communication devices which are based on GSM (Global System for Mobile Communications) technology. It is used to interact with GSM network using a computer. GSM module only understands **AT commands**, and can respond accordingly. The most basic command is “AT”, if GSM respond OK then it is working good otherwise it respond with “ERROR”. There are various AT commands like ATA for answer a call, ATD to dial a call, AT+CMGR to read the message, AT+CMGS to send the sms etc. AT commands should be followed by Carriage return i.e. \r (0D in hex), like “AT+CMGS\r”. We can use GSM module using these commands.

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with “ AT ” or “ at ” . That’s why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state), are also supported by GSM/GPRS modems and mobile phones. Besides this common AT command set, GSM/GPRS modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS – related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

Note that the starting “ AT ” is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS. However, some books and web sites use them interchangeably as the name of an AT command.

Here are some of the tasks that can be done using AT commands with a GSM/GPRS modem or mobile phone:

* Get basic information about the mobile phone or GSM/GPRS modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).
* Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subsriber Identity) (AT+CIMI).
* Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).
* Establish a data connection or voice connection to a remote modem (ATD, ATA, etc).
* Send and receive fax (ATD, ATA, AT+F\*).
* Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).
* Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.

**2.3.3 LM 35 Sensor**

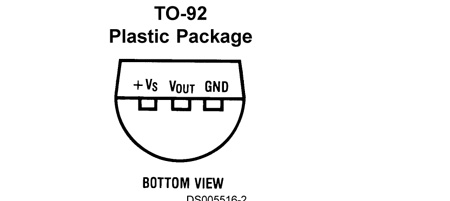
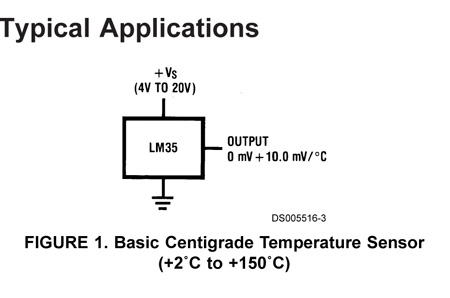
Outputs 10mV per Degree that can also be read directly on multimeter or read in to microcontroller. For example at 30 degree celcius it will output 300mV at linear scale.  
  
The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35D is rated to operate over a 0° to +100°C temperature range.

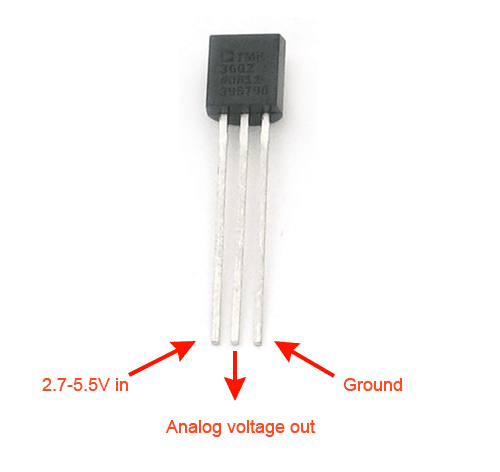
**General Description:**

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35’s low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range,while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

**Features:**

* Calibrated directly in Celsius (Centigrade)
* Linear + 10.0 mV/ C scale factor
* 0.5 C accuracy guaranteeable (at +25 C)
* Rated for full -55 to +150 C range
* Suitable for remote applications
* Low cost due to wafer-level trimming
* Operates from 4 to 30 volts
* Less than 60 A current drain
* Low self-heating, 0.08 C in still air
* Nonlinearity only 1/4 C typical
* Low impedance output, 0.1 W for 1 mA load





**Fig 5. LM35 Sensor**

**Applications:**

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is expecially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die’s temperature will not be affected by the air temperature.

**2.3.4 16\*2 LCD DISPLAY:**

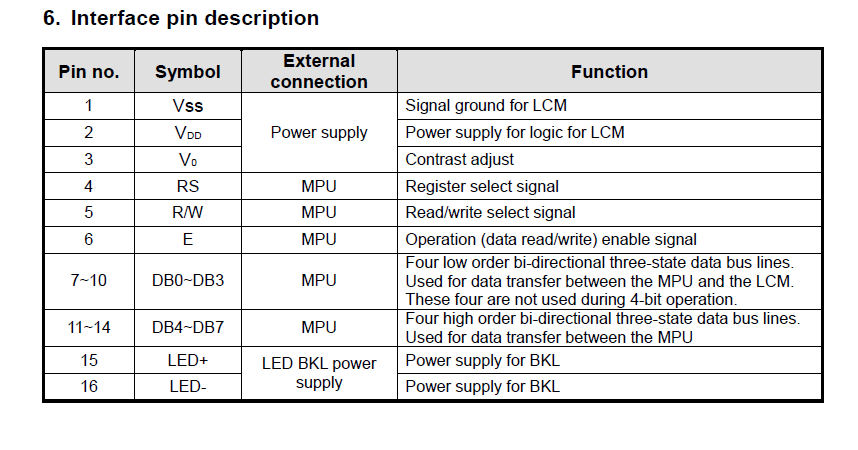
LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](http://www.engineersgarage.com/content/led)s. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even [custom characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

**Fig 6. 16\*2 LCD**

**Table 3: Pin description of 16x2 LCD**



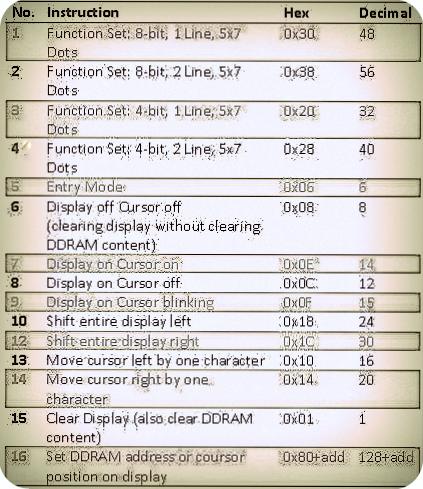
**V0 (Set LCD contrast)**  
Set LCD contrast here. Best way is to use variable resistor such as potentiometer. Output of the potentiometer is connected to this pin. Rotate the potentiometer knob forward and backward to adjust the LCD contrast.

**RS (Register Select)**  
There are two registers in every LCD.  
1. Command Register  
2. Data Register

**Command Register**  
When we send commands to LCD these commands go to Command Register and are processed there.  
Commands with their full description are given in the picture below.  
                                               When RS=0    Command Register is selected.  
**Data Register**  
 When we send data to LCD it goes to data register and is processed there.  
                                       When RS=1    Data Register is selected.  
 **RW (Read - Write)**  
                                              When RW=1   We want to read data from LCD.  
                                       When RW=0   We want to write to LCD.

**EN (Enable signal)**  
When you select the register (Command and Data) and set RW (read -  write) now its time to execute the instruction. By instruction i mean the 8-bit data or 8-bit command present on Data lines of LCD.  
This requires an extra voltage push to execute the instruction and EN (enable) signal is used for this purpose. Usually we make it en=0 and when we want to execute the instruction we make it high en=1 for some mili seconds. After this we again make it ground en=0.

Data which we send to our LCD can be any alphabet (small or big) , digit or ASCII character.  
  
**NOTE:** We can not send an integer, float, long, double type data to LCD because LCD is designed to display a character only. The 8 data pins on LCD carries only ASCII 8-bit code of the character to LCD. However we can convert our data in character type array and send one by one our data to LCD. Data can be sent using LCD in 8-bit 0r 4-bit mode. If 4-bit mode is used, two nibbles of data (First high four bits and then low four bits) are sent to complete a full eight-bit transfer. 8-bit mode is best used when speed is required in an application and at least ten I/O pins are available. 4-bit mode requires a minimum of seven bits. In 4-bit mode, only the top 4 data pins (4-7) are used.



**Fig 7. LCD Instructions**

Command 0x30 means we are setting 8-bit mode LCD having 1 line and we are initializing it to be 5x7 character display. Now this 5x7 is something which everyone should know what it stands for. Usually the characters are displayed on LCD in 5x8 matrices form, where 5 is total number of columns and 8 is total number of rows. Thus the above 0x30 command initializes the LCD to display character in 5 columns and 7 rows. The last row we usually leave for our cursor to move or blink, etc.

The character is displayed on LCD screen in 5x8 or 5x7 matrix, where 5 represents number of columns and 7, 8 represent number of rows. Maximum size of the matrix is 5x8. You can not display character greater then 5x8 dimension matrix. To display character greater than this dimension you have to switch to graphical LCDs. To learn about graphical LCDs here is a good tutorial [GRAPHICAL LCD WORKING AND PINOUT](http://www.microcontroller-project.com/128x64-graphical-lcd.html).

* The command 0x38 means we are setting 8-bit mode LCD having two lines and character shape between 5x7 matrix.
* The command 0x20 means we are setting 4-bit mode LCD having 1 line and character shape between 5x7 matrix.
* The command 0x28 means we are setting 4-bit mode LCD having 2 lines and character shape      between 5x7 matrix.
* The command 0x06 is entry mode it tells the LCD that we are going to use you'
* The command 0x08 dispalys cursor off and display off but with out clearing DDRAM contents.
* The command 0x0E displays cursor on and dispaly on.
* The command 0x0c dispaly on cursor off(displays cursor off but the text will appear on LCD)
* The command 0x0F dispaly on cursor blink(text will appear on screen and cursor will blink).
* The command 0x18 shift entire display left(shift whole off the text on the particular line to its left ).
* The command 0x1C shift entire display right(shift whole off the text on the particular line to its right).
* The command 0x10 Moves cursor one step left or move cursor on step a head to left whenever new character is displayed  on the screen.
* The command 0x14  Moves cursor one step right or move cursor on step a head to right whenever new character is displayed on the screen.
* The command 0x01 clear all the contents of the DDRAM and also clear the LCD removes all the text from the screen.
* The command 0x80 initialize the cursor to the first position means first line first matrix(start point) now if we add 1 in 0x80+1=0x81 the cursor moves to second matrix.

16x1 LCD displays 16 characters only. The first will appear on 0x80 second 0x81 third 0x82 and so on until last, the 16 once on address 0x8F.

**2.3.5 PIEZOELECTRIC BUZZER**

**MICRO BUZZER 5V DC / 20mA PCB TYPE**



**Fig 8. Piezoelectric buzzer**

**Features:**

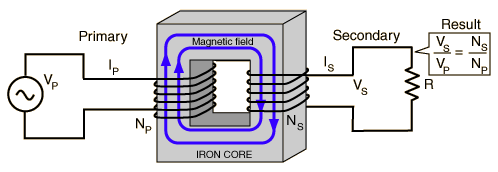
* + sealed: yes
  + operating power: 3-6V DC / 25mA
  + extremely compact, ultrathin construction
  + no electrical noise
  + low current consumption yet high sound pressure level

**Specifications:**

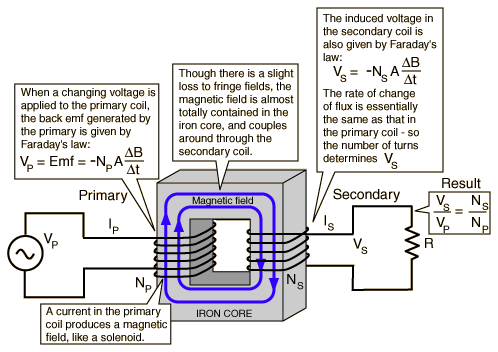
* tone type: single
* operating voltage: 3-6V DC
* rated voltage: 5V DC
* current consumption: 25mA
* osc. frequency: 3.2kHz
* sound level: 87dB
* connector type: pcb
* body color: gray
* weight: 0.056oz

# 2.3.6 Transformer

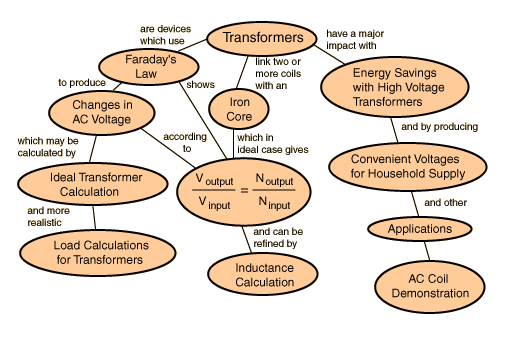
A transformer makes use of [Faraday's law](http://hyperphysics.phy-astr.gsu.edu/Hbase/electric/farlaw.html#c1) and the [ferromagnetic](http://hyperphysics.phy-astr.gsu.edu/Hbase/solids/ferro.html#c4) properties of an [iron core](http://hyperphysics.phy-astr.gsu.edu/Hbase/magnetic/elemag.html#c4) to efficiently raise or lower AC voltages. It of course cannot increase [power](http://hyperphysics.phy-astr.gsu.edu/Hbase/electric/powerac.html#c1) so that if the voltage is raised, the current is proportionally lowered and vice versa.



# Fig 9. Transformer and Faraday's Law



**Fig 10. Working of Transformer**



**Fig 11. Transformer concepts**

**2.3.7 Heart Beat Sensor/Blood Pressure Sensor:**



**Fig 12. Heart Beat Sensor/Blood Pressure Sensor**

Blood Pressure & Pulse reading are shown on display with serial out for external projects of embedded circuit processing and display. It shows Systolic, Diastolic and Pulse Readings. Compact design fits over your wrist like a watch. Easy to use wrist style eliminates pumping.

**Features :**

* Intelligent automatic compression and decompression
* Easy to operate, switching button to start measuring
* 60 store groups memory measurements
* Can read single or all measures
* 3 minutes automatic power saving device
* Intelligent device debugging, automatic power to detect
* Local tests for : wrist circumference as 135-195mm
* Large-scale digital liquid crystal display screen, Easy to Read Display
* Fully Automatic, Clinical Accuracy, High-accuracy
* Power by External +5V DC
* Serial output data for external circuit processing or display.

**Specification**

* Working Voltage: +5V, 200mA regulated
* Output Format :Serial Data at 9600 baud rate(8 bits data, No parity, 1 stop bits). Outputs three parameters in ASCII.
* Sensing unit wire length is 2 meters

**Sensor Pinouts**

* TX-OUT = Transmit output. Output serial data of 3V logic level, Usually connected to RXD pin of microcontrollers/RS232/USB-UART.
* +5V = Regulated 5V supply input.
* GND = Board Common Ground

**Note: Product does not require battery for operation. It is powered from external PCB as per above pinouts.**

### Blood Pressure Basics

Blood pressure is the pressure of the blood in the arteries as it is pumped around the body by the heart. When your heart beats, it contracts and pushes blood through the arteries to the rest of your body. This force creates pressure on the arteries. Blood pressure is recorded as two numbers— the systolic pressure (as the heart beats) over the diastolic pressure (as the heart relaxes between beats). The unit which measures this is called Sphygmomanometer.

Monitoring blood pressure at home is important for many people, especially if you have high blood pressure. Blood pressure does not stay the same all the time. It changes to meet your body’s needs. It is affected by various factors including body position, breathing or emotional state, exercise and sleep. It is best to measure blood pressure when you are relaxed and sitting or lying down.

**Classification of blood pressure for adults (18 years and older)**

|  |  |  |
| --- | --- | --- |
|  | **Systolic (mm Hg)** | **Diastolic (mm Hg)** |
| **Hypotension** | < 90 | < 60 |
| **Desired** | 90–119 | 60–79 |
| **Prehypertension** | 120–139 | 80–89 |
| **Stage 1 Hypertension** | 140–159 | 90–99 |
| **Stage 2 Hypertension** | 160–179 | 100–109 |
| **Hypertensive Crisis** | ≥ 180 | ≥ 110 |

High blood pressure (hypertension) can lead to serious problems like heart attack, stroke or kidney disease. High blood pressure usually does not have any symptoms, so you need to have your blood pressure checked regularly.

### Output Readings

Following are example output readings from sensor. Each reading consists of **15 bytes** at 9600 baud rate. The reading packet's last byte is always enter key character(0x0A in hex and 10 in decimal) so you can view each reading on new line. Also this character can be used to sync in microcontrollers after reach readings.

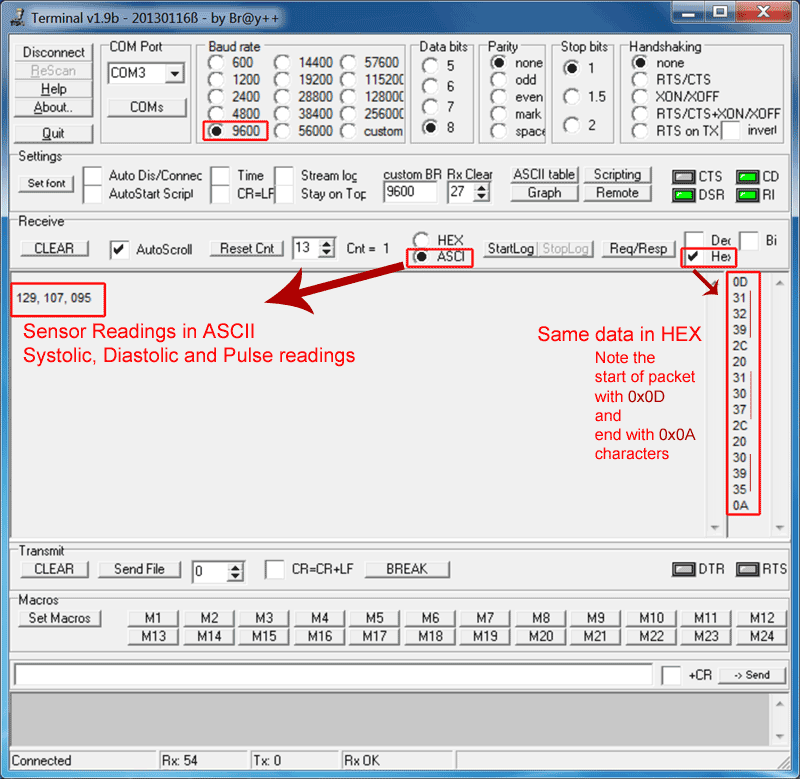
The output reading is 8bit value in ASCII format fixed digits, from 000 to 255.

Typical reading will be like below where the three values separated by comma and space.

* Systolic
* Diastolic
* Pulse

**129, 107, 095**

We have used Terminal software to view reading, you can use any other software also which allows connecting to COM port and view/process data.



**2.3.8 Glucometer:**

Glucometers, also known as [glucose meters](https://www.verywellhealth.com/how-accurate-is-your-glucose-meter-3289632), are highly sophisticated, requiring only a single drop of blood, and are conveniently sized and portable. They are small enough to take with you on-the-go, and based on your comfort level, can be used anywhere at any time. Whether you have type 1 or type 2 diabetes, [at-home blood-sugar monitoring](https://www.verywellhealth.com/why-i-test-my-blood-sugar-2242257) devices called glucometers can give you valuable information about whether your [blood sugar is too low](https://www.verywellhealth.com/recommended-blood-glucose-levels-for-diabetes-1087681) or too high, or in a good range for you. These portable electronic devices provide you with instant feedback and let you know immediately what your blood sugar is. Regular monitoring is a particularly helpful way to manage your diabetes and help control your blood sugar, so it's important to know how to properly use the device.



**Fig 13.** **Glucometer**

**Natural blood glucose regulation -**

Glucose (C6H12O6) is a carbohydrate whose most important function is to act as a source of energy for the human body, by being the essential precursor in the synthesis of AT (adenosine triphosphate). The energy stored in ATP can then be used to drive processes requiring energy, including biosynthesis, and locomotion or transportation of molecules across cell membranes. According to cellular requirements, glucose can also be used in the creation of proteins, glycogen, and lipids.

The blood glucose concentration is very tightly regulated. Human body has two hormones released by pancreas that have opposite effects: insulin and glucagon. Insulin is produced by beta cells of the pancreas while glucagon is produced by alpha cells.

The release of insulin is triggered when high levels of glucose are found in the bloodstream, and glucagon is released with low levels of glucose in the blood.

This blood glucose regulation process can be explained in the following steps:

1. After the glucose has been absorbed from the food eaten, it gets released in the bloodstream. High blood glucose levels triggers the pancreas to produce insulin. Insulin enables the muscle cells to take glucose as their source of energy and to form a type of molecule called glycogen that works as secondary energy storage in the case of low levels of glucose. In the liver cells, insulin instigates the conversion of glucose into glycogen and fat. In the fat cells of the adipose tissue, insulin also promotes the conversion of glucose into more fat and the uptake of glucose.

2. The pancreas will continue to release insulin and liver and fat cells continue to use glucose till the drop of concentration of glucose is below a threshold; in that case, glucagon will be released instead of insulin.

3. When glucagon reaches the liver cells, it initiates the conversion of glycogen into glucose, and fat into fatty acids, which many body cells can use as energy after the glucagon enables them to. The cells will continue to burn fat from the adipose tissue as an energy source, and follow with the protein of the muscles, until the levels of glucose increase again by the digestion of food, and that terminates the cycle.

**CHAPTER 3**

**METHODOLOGY**

**3.1 STEPS IN PROJECT IMPLEMENTATION**

• Finally system is tested and encountered error is omitted.

After the detail literature survey through the books, periodical, journal, magazine,

The steps taken in the implementation process are :

1. Circuit development and design.

2. Writing and developing code on MPLAB IDE.

3. Circuit simulation on Proteus Lite.

4. WIFI/GSM MOMEM testing and configuration using a PC.

5. Programming the Microcontroller.

6. Bread board testing of circuit.

7. Building and soldering circuit on a Vero Board.

8. Troubleshooting and testing.

9. Adding an alert system to the circuit.

10. Writing code to control the beeping of the buzzer.

11. Incorporating a 12 V rechargeable battery to serve as a secondary power source.

12. Developing a charging circuit for the battery.

13. Writing code to control charging of battery.

14. Simulating circuit with the new code.

15. Reprogramming the microcontroller.

16. Further troubleshooting and testing.

17. Casing design and construction.

18. Packaging.

19. Final testing of circuit.

**CHAPTER 4**

**SOFTWARE DEVELOPMENT**

**4.1 SELECTION OF MICROCONTROLLER DEVELOPMENT TOOLS:**

Once microcontroller is selected, selecting a perfect development tools is most important. For develop every microcontroller based system, a set of software and hardware tools are required. Software tools for editing and debugging and troubleshooting the microcontroller program. W9/hile hardware tools for burning computer code into microcontroller and testing microcontroller hardware. A good development tools must have following properties:

1. Simple to use.

2. Not many steps execution.

3. Inexpensive.

4. Must include basic functions like editor, debugger, compiler.

5. Must include power supply and basic hardware required and I/O pins connector facility.

6. Cross-platform development.

7. Must support different programming language and computer operating system.

**4.2** **EMBEDDED C**

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

**Features:-**

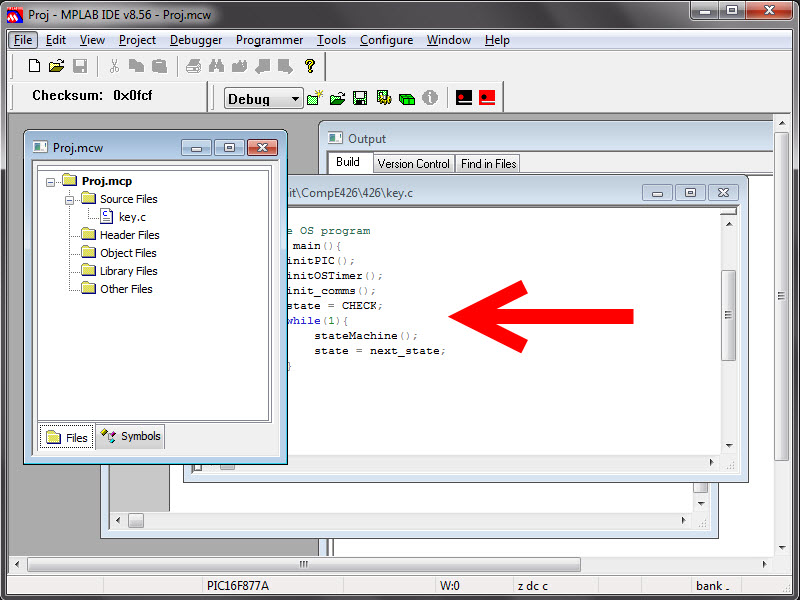
1. It is small and simpler to learn, understand, program and debug.
2. Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
3. C compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
4. C has advantage of processor-independence and is not specific to any particular microprocessor/microcontroller or any system.
5. As C combines functionality of assembly language and features of high level languages.
6. It is fairly efficient.
7. It supports access to I/O and provides ease of management of large embedded projects.

**4.3 MPLAB IDE SOFTWARE:**

MPLAB X IDE 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 X 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 X IDE is based on the open source NetBeans 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.

1. Open MPLAB IDE v8.56
2. From the 'Projects' tab, select the first option 'Project Wizard' Click on 'Next' in the
3. Select the desired PIC which you need to program or build your project on and click on 'Next'
4. Select the active tool suite you require; among the list of tool suites given (Usually the HI-TECH Universal tool suite is preferred, if installed)
5. Check if the ToolSuite contents listed contains a compiler suiting your programming needs("HI-TECH ANSI C Compiler" in the case of a HI-TECH Universal toolsuite) and click 'Next'
6. Create a new project file at your desired location in the desired name.
7. Take care that the project file is saved in the '\*.mcp' format and click 'Next'
8. In the next window , add any files you desire to add to your new project, if required. Else just skip this step by clicking 'Next'.
9. Now click 'finish' and your new project is created.
10. Now select the 'New' option from the 'File' tab.v
11. Select 'Save as' option from the 'File' tab and save the new file in the same folder in which you have created the project by selecting a suitable option from 'save as type'(depending on which type of program you're doing)
12. Go to the 'Project' tab and select the option 'Add Files To The Project' and add the file saved in the previous step you're doing programming in C
13. Assembly Source Files if you're doing programming in ASSEMBLY language etc.
14. Begin programming in the file.



**Fig 14. MPLAB IDE SOFTWARE**

**4.4 PROGRAMMER PIC KIT3 :**

The MPLAB PIC kit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 2, MPLAB ICD 3 and MPLAB REAL ICE).HID interface, say no more driver again Support windows 7 system USB (Full speed 12 Mbits/s interface to host PC) Real-time execution MPLAB IDE compatible (free copy included)Built-in over-voltage/short circuit monitor.Firmware upgradeable from PC/web download totally enclosed supports low voltage to 2.0 volts (2.0v to 6.0v range) Diagnostic LEDs (power, busy, error) Read/write program and data memory of microcontroller Erase of program memory space with verification Freeze-peripherals at breakpoint.

Program up to 512K byte flash with the Programmer-to-Go  
Material Plastic Housing Dimensions  
95 x 40 x 13 mm / 3.74 x 1.57 x 0.51 inch



**Fig 15. PIC KIT3**

**4.5 CCS COMPILER:**

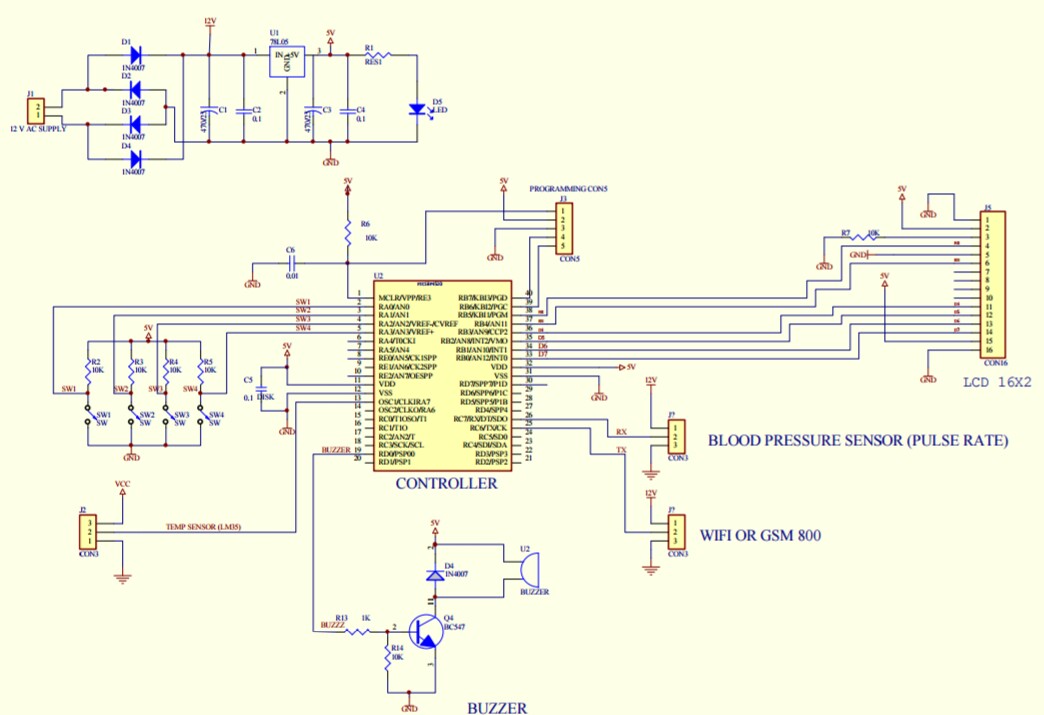
The CCS C Compiler for PIC10, PIC12, PIC14, PIC16, and PIC18 microcontrollers has 307 Built-in Functions to access PIC® MCU hardware is easy and produces efficient and highly optimized code. Functions such as timers, A/D, EEPROM, SSP, PSP, USB, I2C and more:

* Built-in libraries that work with all chips for RS-232 serial I/O, I2C, discrete I/O and precision delays.
* Serial I/O functions allow standard functions such as GETC() and PRINTF() to be used for RS-232 like I/O.
* Formatted printf allows easy formatting and display in HEX or decimal.
* Multiple I2C and RS232 ports may be easily defined.
* #use rs232() offers options to specify a maximum wait time for getc.
* Hardware tranceiver used when possible, but for all other occasions the compiler generates a software serial transceiver.
* Microcontroller clock speed may be specified in a PRAGMA to permit built-in functions to delay for a given number of microseconds or milliseconds.
* Functions such as INPUT() and OUTPUT\_HIGH() properly maintain the tri-state registers.
* Compiler directives determine if tri-state registers are refreshed on every I/O or if the I/O is as fast as possible.
* [**#USE SPI ()**](https://www.ccsinfo.com/content.php?page=compiler-features#usespi)
* Simple functions like READ\_ADC() to read a value from A/D converter.
* Source code drivers included for LCD modules, keypads, 24xx and 94xx serial EEPROM, X10, DS1302 and NJU6355 real time clocks, Dallas touch memory devices, DS2223 and PCF8570, LTC1298 and PCF8591 A/D converters, temperature sensors, digital pots, I/O expander and much more.
* 133 ready-to-run programs included.

**CHAPTER 5**

**CONNECTION DIAGRAMS**

**5.1 CIRCUIT DIAGRAM**

****

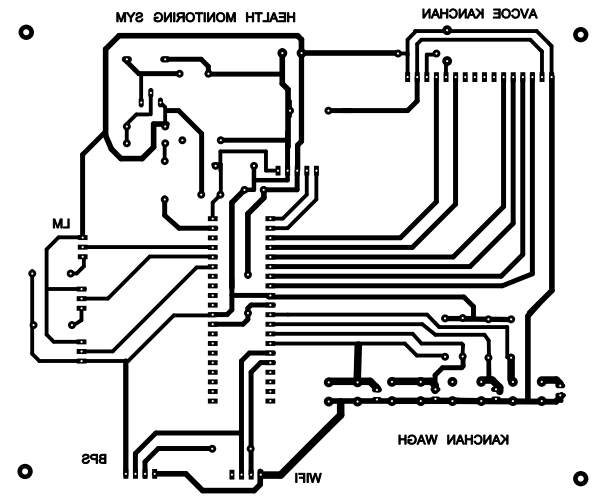
**Fig 16. Circuit Diagram**

**5.2 Layout:**

Layout designing with pencil sketch of component of conductors drawing which contain all relevant information for preparation of artwork layout is designing by taking paper for better accuracy.

**Rules for the layout:**

1. First rule s to prepare each and every PLB layout as view from the component side or top side.
2. Another important rule is not to start the designing of layout unless clear circuit diagram is available.
3. Develop the layout in the direction of signal flow and between the smaller components larger size component are to be placed.
4. Among the components larger size component should be placed first and then smaller component should be placed, and then all the component should be placed in such manner that the dishoarding of the other component will not occur.
5. In designing of PCB layout is very important to divide the circuit into functional subunit which helps in testing.
6. Masks the input, in output and power connection to appropriate point.
7. Use two different colors for drawing a layout of double sided PCB.
8. The rule for the width of conductor is as follows width of ground > width of supply> of the signal.

****

**Fig 17. Layout**

**CHAPTER 6**

**ALGORITHM AND FLOWCHART**

**6.1 ALGORITHM**

Step 1 : Start

Step 2 : Initialize sensors, controller, display & GSM module

Step 3 : Sensor starts to sense patient’s body parameter

Step 4 : Temperature sensor senses patient’s temperature

Step 5 : Heart beat sensor senses patient’s pulse rate, systolic & diastolic blood pressure

Step 6 : All parameters send to microcontroller

Step 7 : If any parameter goes above setting point

Step 8 : Buzzer blow, GSM turn on internet and send data to web server also send SMS to doctor

Step 9 : All parameters display on LCD

Step 10 : Stop

**6.2 FLOWCHART**

Start

Initialise sensors, controller, display & GSM module

Sensor starts to sense patient’s body parameter

Temperature sensor senses patient’s temperature

Heart beat sensor senses patient’s pulse rate, systolic & diastolic BP

All parameters send to microcontroller

If any parameter goes above setting point

Buzzer blow, GSM turn on internet and send data to web server also send SMS to doctor

All parameters display on LCD display

Stop

**CHAPTER 7**

**PROGRAMMING**

**7.1 PROGRAM**

#include <18f4520.h>

#DEVICE ADC=10

#fuses INTRC\_IO,NOPROTECT,BROWNOUT,NOMCLR,NOCPD,NOWDT,NOPUT,FCMEN

#use delay(clock=8000000)//,restart\_wdt)

#use rs232(baud=9600, xmit=PIN\_C6, rcv=PIN\_C7)

#byte adresh = 0x1e

#byte adresl = 0x9e

#bit adfm = 0x1f.7

#define BUZZ\_ON OUTPUT\_HIGH(PIN\_A2);

#define BUZZ\_OFF OUTPUT\_LOW(PIN\_A2);

#define RS\_HI OUTPUT\_HIGH(PIN\_B5);

#define RS\_LO OUTPUT\_LOW(PIN\_B5);

#define EN\_HI OUTPUT\_HIGH(PIN\_B4);

#define EN\_LO OUTPUT\_LOW(PIN\_B4);

#define D4\_HI OUTPUT\_HIGH(PIN\_B3);

#define D4\_LO OUTPUT\_LOW(PIN\_B3);

#define D5\_HI OUTPUT\_HIGH(PIN\_B2);

#define D5\_LO OUTPUT\_LOW(PIN\_B2);

#define D6\_HI OUTPUT\_HIGH(PIN\_B1);

#define D6\_LO OUTPUT\_LOW(PIN\_B1);

#define D7\_HI OUTPUT\_HIGH(PIN\_B0);

#define D7\_LO OUTPUT\_LOW(PIN\_B0);

#define sel\_ON OUTPUT\_HIGH(PIN\_D2);

#define sel\_OFF OUTPUT\_LOW(PIN\_D2);

#define LCD\_LINE\_1\_START\_ADDRESS 0x80

#define LCD\_LINE\_2\_START\_ADDRESS 0xC0

#define LCD\_LINE\_3\_START\_ADDRESS 0x90

#define LCD\_LINE\_4\_START\_ADDRESS 0xD0

#define LCD\_LINE\_1\_END\_ADDRESS 0x8F

#define LCD\_LINE\_2\_END\_ADDRESS 0xCF

#define LCD\_LINE\_3\_END\_ADDRESS 0x9F

#define LCD\_LINE\_4\_END\_ADDRESS 0xDF

#define LCD\_DATA\_WRITE 1

#define LCD\_CMD\_WRITE 0

int8 ucKeyPressed = 0;

int8 Send\_Sms\_Flag = 0;

int8 ucRxChar = 0;

int8 UCBLOOD\_SUGURF = 0;

int16 uiTemp = 0;

int8 ucresponsef = 0;

int8 ucrelayon\_f = 0;

int8 ucsendsmsf = 0;

volatile int16 hbyte = 0;

BYTE CONST ucWelcm\_1\_Array[17] = "IOT BASED PATIENT";//22

BYTE CONST ucWelcm\_2\_Array[17] = "HEALTH MONITORING";//23

BYTE CONST ucCITY2\_Array[17] = " ";//25

BYTE CONST ucCITY3\_Array[17] = "Temp: BS: ";//25

BYTE CONST ucCITY4\_Array[17] = "S: D: P: ";//25

#INT\_EXT

void Ex\_Int()

{

if(INPUT(PIN\_A1) == 0)

{

ucvibretflag = 1;

// Send\_Sms\_Flag = 1;

}

else

{

ucvibretflag = 0;

}

}

#INT\_RDA

void Rx\_int()

{

ucRxChar = getch();

if(ucSmsReadFlag == 1)

{

if(index == 24)//24)

{

index = 0;

ucSmsReadFlag = 2;

// RLY\_ON;

}

}

if(index < 76)

ucRxARR[index++] = ucRxChar;

if(ucsel\_hi == 1)

{

ucRxTimOut = 10;

}

}

void Send\_Sms\_Action(void)

{

sel\_OFF;

delay\_ms(300);

// if(ucsendsmsf == 1)

{

ucsendsmsf = 0;

printf("AT+CGATT?\r\n");

delay\_ms(500);

delay\_ms(500);

printf("AT+SAPBR=1,1\r\n");

delay\_ms(500);

//UART0\_SendStr("AT+CGATT?\r\n");

//Delay(1000);

delay\_ms(500);

printf("AT+HTTPPARA=\"URL\",\"api.thingspeak.com/update?api\_key=6F95OD8XLBJNJXSE");

printf("&field1=");

putc(((ui\_R\_vltg/10)%10)+0x30);

putc(((ui\_R\_vltg/1)%10)+0x30);

printf("&field2=");

putc(((ucvar1/100)%10)+0x30);

putc(((ucvar1/10)%10)+0x30);

putc(((ucvar1/1)%10)+0x30);

printf("&field3=");

putc(((ucvar2/100)%10)+0x30);

putc(((ucvar2/10)%10)+0x30);

putc(((ucvar2/1)%10)+0x30);

printf("&field4=");

putc(((ucvar3/100)%10)+0x30);

putc(((ucvar3/10)%10)+0x30);

putc(((ucvar3/1)%10)+0x30);

printf("&field5=");

putc(((ucvar4/100)%10)+0x30);

putc(((ucvar4/10)%10)+0x30);

putc(((ucvar4/1)%10)+0x30);

putc('"');

putc('\r');

putc('\n');

delay\_ms(500);

delay\_ms(500);

printf("AT+HTTPTERM\r\n");

delay\_ms(500);

printf("AT+IFC=1,0\r\n");

delay\_ms(1000);

printf("AT+CMGF=1\r\n");

delay\_ms(2000);

// printf("AT+CMGS=\"+919503561068\"\r\n");//prj transformer

printf("AT+CMGS=\"+919689039874\"\r\n");//919096350078

delay\_ms(2000);

printf("TEMP:");

putc(((ui\_R\_vltg/10)%10)+0x30);

putc(((ui\_R\_vltg/1)%10)+0x30);

printf("DC");

putc('\r');

putc('\n');

printf("BLOOD SUGUR:");

putc(((ucvar1/100)%10)+0x30);

putc(((ucvar1/10)%10)+0x30);

putc(((ucvar1/1)%10)+0x30);

printf("mg/dL");

putc('\r');

putc('\n');

printf("SYSTOLIC BP:");

putc(((ucvar2/100)%10)+0x30);

putc(((ucvar2/10)%10)+0x30);

putc(((ucvar2/1)%10)+0x30);

printf("mmHg");

putc('\r');

putc('\n');

printf("DYSTOLIC BP:");

putc(((ucvar3/100)%10)+0x30);

putc(((ucvar3/10)%10)+0x30);

putc(((ucvar3/1)%10)+0x30);

printf("mmHg");

putc('\r');

putc('\n');

printf("PULSE RATE:");

putc(((ucvar4/100)%10)+0x30);

putc(((ucvar4/10)%10)+0x30);

putc(((ucvar4/1)%10)+0x30);

printf("/min");

delay\_ms(100);

delay\_ms(100);

putc(0x1A);

delay\_ms(1550);

delay\_ms(1550);

delay\_ms(1550);

delay\_ms(1550);

index = 0;

ucsendsmsf = 2;

//.. ui1SecCNT = 0;

}

}

void main(void)

{

// int8 ucVar = 0;

SETUP\_ADC(ADC\_OFF); //disable ADC i/p

SETUP\_ADC\_PORTS(NO\_ANALOGS); //disable analog i/p

setup\_comparator(NC\_NC\_NC\_NC);

SETUP\_CCP1(CCP\_OFF);

SET\_TRIS\_A(0x01);//0000 1011

SET\_TRIS\_B(0x00);//0000 0111

SET\_TRIS\_C(0x80);//1000 0000

SET\_TRIS\_D(0xF1);//1111 0001

SET\_TRIS\_E(0x00);//0000 0000

SETUP\_TIMER\_1(T1\_INTERNAL|T1\_DIV\_BY\_8); //enables timer1

SET\_TIMER1(40536); // timer of 200ms (64286);//10msec

enable\_interrupts(INT\_RDA);

ENABLE\_INTERRUPTS(INT\_TIMER1);

ENABLE\_INTERRUPTS(INT\_EXT);

// setup\_wdt(WDT\_2304MS);

ENABLE\_INTERRUPTS(GLOBAL);

BUZZ\_ON;

delay\_ms(250);

BUZZ\_OFF;

INIT\_LCD();

while(1)

{

Get\_Key();

ADC\_CALL(0);

uiRvlt\_adc = (current\_adc\_val/2);

uiRvlt\_avrg += uiRvlt\_adc;

ucrvltgcnt++;

if(ucrvltgcnt == 10)

{

ucrvltgcnt = 0;

ui\_R\_vltg = uiRvlt\_avrg/10;

uiRvlt\_avrg = 0;

}

if(INPUT(PIN\_D5) == 0)

{

buzz\_ON;

delay\_ms(450);

buzz\_OFF;

ucsel\_hi = 1;

}

if(INPUT(PIN\_D4) == 0)

{

ucsel\_hi = 0;

sel\_OFF;

buzz\_ON;

delay\_ms(450);

buzz\_OFF;

ucsendsmsf = 1;

Send\_Sms\_Action();

ucvar3 = 0;

ucvar2 = 0;

ucvar4 = 0;

ucvar1 = 70;

ui1SecCNT = 1;

}

if(ucsel\_hi == 1)

{

sel\_ON;

}

else

{

sel\_OFF;

}

if((INPUT(PIN\_D7) == 0)&&(INPUT(PIN\_D6) == 0))

{

ui2SecCNT = 50;

if((UCBLOOD\_SUGURF == 0)||(UCBLOOD\_SUGURF == 3))

{

UCBLOOD\_SUGURF = 1;

}

else

{

UCBLOOD\_SUGURF = 2;

}

buzz\_ON;

delay\_ms(450);

buzz\_OFF;

}

if(ui2SecCNT == 1)

{

ui2SecCNT = 2;

if(UCBLOOD\_SUGURF == 1)

{

if(ucKeyPressed == 2)

{

if(ucvar1 < 160)

{

ucvar1++;

}

else

{

ucvar1 = 70;

}

}

if(ucKeyPressed == 1)

{

if(ucvar1 > 70)

{

ucvar1--;

}

else

{

ucvar1 = 160;

}

}

}

else if(UCBLOOD\_SUGURF != 4)

{

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(1,5,ui\_R\_vltg,2,0);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(1,12,ucvar1,3,0);

}

}

if(UCBLOOD\_SUGURF == 1)

{

LCD\_WRITE\_Const\_ARRAY(1,0,17,16);

LCD\_WRITE\_Const\_ARRAY(2,0,18,16);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(2,5,ucvar1,3,0);

}

if((UCBLOOD\_SUGURF == 2)||(UCBLOOD\_SUGURF == 0)||(UCBLOOD\_SUGURF == 3))

{

UCBLOOD\_SUGURF = 3;

LCD\_WRITE\_Const\_ARRAY(1,0,19,16);

LCD\_WRITE\_Const\_ARRAY(2,0,20,16);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(1,5,ui\_R\_vltg,2,0);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(1,12,ucvar1,3,0);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(2,2,ucvar2,3,0);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(2,8,ucvar3,3,0);

Show\_Float\_No\_ONLine\_At\_Offset\_IntDig\_FltDig(2,14,ucvar4,2,0);

}

if(ucRxTimOut == 1)

{

ucRxTimOut = 0;

ucresponsef = 1;

ucrelayon\_f = 1;

if(ucsel\_hi == 1)

{

buzz\_ON;

delay\_ms(450);

buzz\_OFF;

ucvar2 = ((((ucRxARR[1])\*100)+((ucRxARR[2])\*10)+(ucRxARR[3]))+0x30);

ucvar3 = ((((ucRxARR[6])\*100)+((ucRxARR[7])\*10)+(ucRxARR[8]))+0x30);

ucvar4 = ((((ucRxARR[11])\*100)+((ucRxARR[12])\*10)+(ucRxARR[13]))+0x30);

}

for(ucRxIndex = 0;ucRxIndex < 20;ucRxIndex++)

{

ucRxARR [ucRxIndex] = 0;

}

ucRxIndex = 0;

}

}

}

**CHAPTER 8**

**ADVANTAGES, DISADVANTAGES, APPLICATIONS**

**8.1 ADVANTAGES**

1. The people in need of healthcare services find it very expensive this is particularly true in developing countries. As a result, this project is an attempt to solve a healthcare problem currently society is facing.
2. The main objective of the project was to design a remote healthcare system.
3. Remote viewing of the data enables a doctor or guardian to monitor a patient’s health progress away from hospital premises.
4. This system is very affordable if it were to be mass produced.
5. Connected hospital equipment that notifies doctors of their current location, informs the hospital management of the replacement needs and monitors staff performance.
6. Prevention: Smart sensors analyze health conditions, lifestyle choices, environment and recommend preventive measures, which will reduce the occurrence of diseases and acute states.
7. IOT reduces costly visits to doctors and hospital admission and makes testing more affordable.
8. Medical data accessibility: Accessibility of electronic medical records allows patients to receive quality care and help healthcare providers make the right medical decisions and prevent complications.
9. Real-time data access and intelligent data integration.

10) Widely recognized technologies to be used for easy access.

11) Cost efficient technique and ubiquitous monitoring.

12) Easy addition of new sensors to existing system.

13) Portable and user-friendly.

**8.2 DISADVANTAGES**

1. Risk of failure: Failure or bugs in the hardware or even power failure can impact the performance of sensors and connected equipment placing healthcare operations at risk. In addition, skipping a scheduled software update may be even more.
2. While IOT promises to reduce the cost of healthcare in the long-term, the cost of its implementation in hospitals and staff training is quite high.
3. Difficult to implement in non-interactive group of networks.
4. Adoption of new technologies is difficult to equip with for elders.
5. Constant updation and upgradation of devices is needed.
6. Data may not be accurate always due to environmental interference by Wi-Fi - signals.
7. Inaccurate data due to wrong positioning of devices.
8. Multiple applications lead to complexity.

**8.3 APPLICATIONS**

1. It can be used in ICU's, operation theaters, monitoring of sugar level, oxygen level etc.
2. It can be also used to monitor the various parameters of a sick person in old age homes.
3. 24/7 health care can be provided. Beneficial for emergency conditions.
4. Tele consulting between general practitioner and specialists.

**CHAPTER 9**

**COSTSHEET**

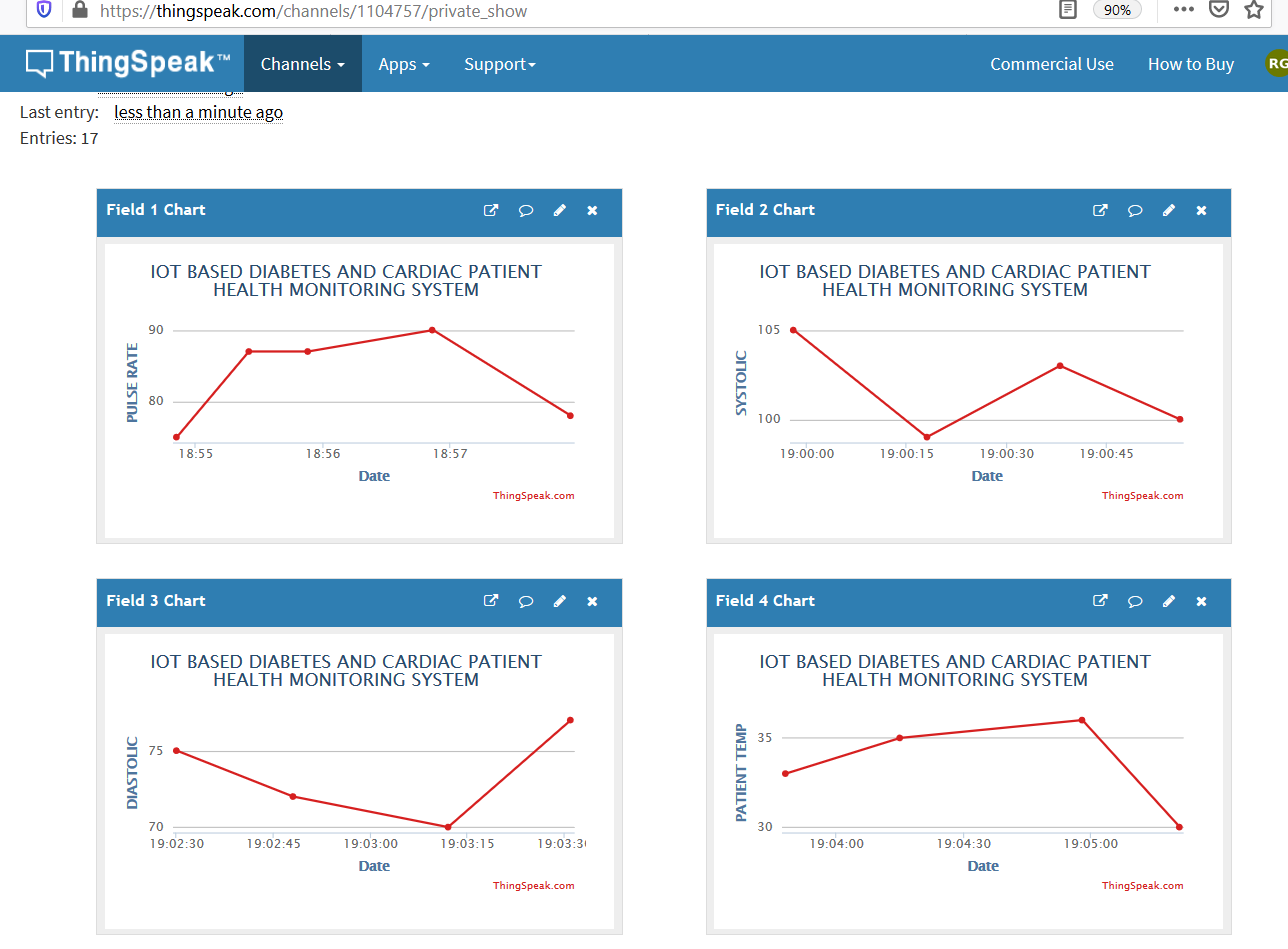
**9.1 COMPONENTS, SPECIFICATIONS AND THEIR COSTS**

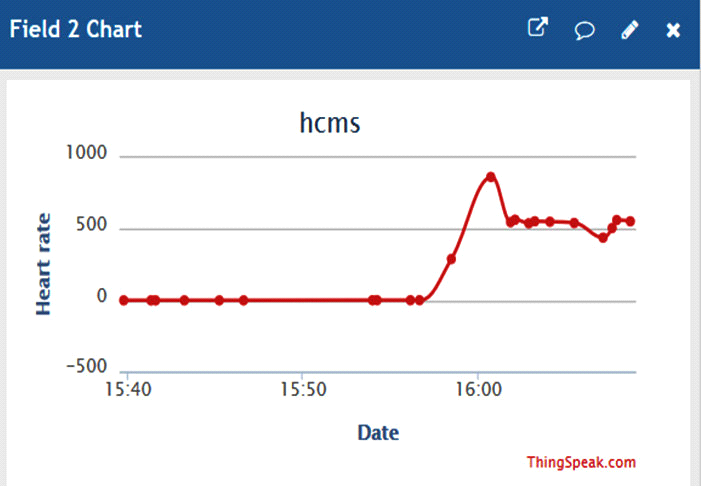
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COMPONENT** | **SPECIFICATION** | **QUANTITY** | **Cost (For 1)** | **Total cost** |
| Microcontroller | PIC 18F4520 | 1 | 630 | 630 |
| Transistor | BC547 | 1 | 5 | 5 |
| Buzzer | 5 V DC | 1 | 35 | 35 |
| Resistor | 1 K, 10 K, Random | 8 | 1 | 8 |
| Capacitor | 0.1uf, 0.01uf | 7 | 3 | 21 |
| Regulator IC | LM 7805 | 1 | 20 | 20 |
| Diode | 1N4007 | 7 | 3 | 21 |
| Blood pressure sensor | 5 V DC serially o/p | 1 | 4100 | 4100 |
| LCD display | 16\*2 | 1 | 320 | 320 |
| GSM module | GSM 800, 12 V AC or DC | 1 | 1550 | 1550 |
| Tact switches | 5 V DC | 4 | 8 | 32 |
| Temperature sensor | LM35 V DC | 1 | 80 | 80 |
| Capacitor | 450 uf/25 V | 1 | 10 | 10 |
| Capacitor | 1000 uf/25 V | 1 | 15 | 15 |
| Transformer | 12 V DC 2 A | 1 | 120 | 120 |
| LED | Upto 3 V 1 mA | 1 | 5 | 5 |
| PCB | 120\*80 MM | 1 | 100 | 100 |

**CHAPTER 10**

**SYSTEM OVERVIEW**

**10.1 SYSTEM OVERVIEW**

****



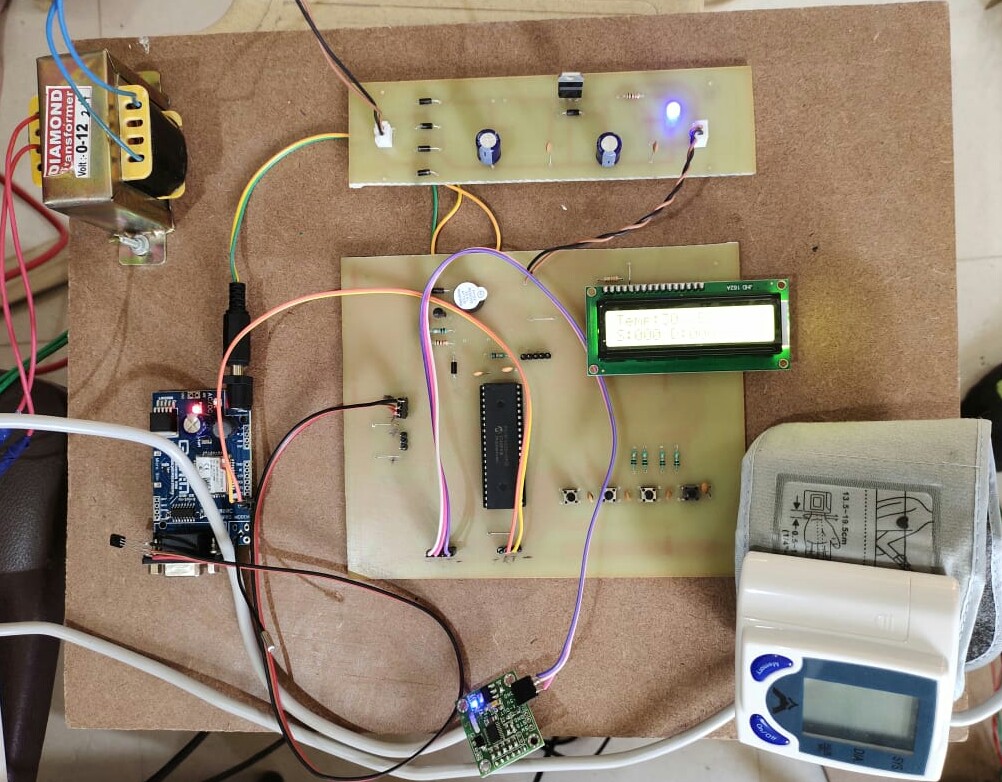
**Fig 18. System overview**

**CHAPTER 11**

**RESULTS**

**11.1 RESULTS :**

Thus, the proposed system could gather, reading of various important indications of the patient and after that evaluate at cloud then caution the doctor or concerned individuals about the health condition. It monitors the vital signs and sense abnormalities. These abnormalities alert the medical staff, it reduces the manual monitoring. The system uses IOT communication to send the data to cloud platform. This message protocol transmits the readings of important patient’s vital sense and helps a web interface to give a pictorial representation of information.



**Fig 19. IOT BASED HUMAN HEALTH MONITORING SYSTEM.**

**CHAPTER 12**

**CONCLUSION**

**12.1 CONCLUSION :**

* The main objective of the experiment was successfully achieved. All the individual modules like heartbeat detection module, temperature detection module, blood pressure viewing module, etc. gave out the intended results.
* The designed system modules can further be optimized and produced to a final single circuit. More important fact that came up during project design is that all the circuit components used in the remote health detection system are available easily.
* With the development in the integrated circuit industry, Micro Electro Mechanical Systems (MEMs) and microcontrollers have become affordable, have increased processing speeds, miniaturized and power efficient. This has led to increased development of embedded systems that the healthcare specialists are adopting. These embedded systems have also been adopted in the Smartphone technology. And with increased internet penetration in most developing countries through mobile phones, and with use of Internet of things (IOT) will become adopted at a faster rate. The Remote Health Care system utilizes these concepts to come up with a system for better quality of life for people in society.
* From an engineering perspective, the project has seen concepts acquired through the computer science and embedded study period being practically applied. The Electric circuit analysis knowledge was used during design and fabrication of the individual modules. Electromagnetic fields analysis used in the wireless transmission between microcontrollers and software programming used during programming of the microcontrollers to come up with a final finished circuit system.

**CHAPTER 13**

**REFERENCES**

**13.1 REFERENCES :**

[1]Ovidiu Apostu, Bogdan Hagiu, Sever Paşca, Wireless ECG Monitoring and Alarm System Using ZigBee”2011 The International Symposium on ADVANCED TOPIC INELECTRICAL ENGINEERING2068-7966/ATEE 2011.

[2]Warsuzarina Mat Jubadi, Siti Faridatul Aisyah Mohd ahak”,Heartbeat Monitoring Alert via SMS”, 978-1-4244-4683-4/09/$25.00 ©2009 IEEE.

[3]Goutam Motika, Abinash Prusty,” Wireless FetalHeartbeat Monitoring System Using ZigBee & IEEE 802.15.4 Standard”, 2011 Second International Conference on Emerging Applications of Information Technology, 978-0-7695-4329-1/11 $26.00 © 2011 IEEE DOI 10.1109/EAIT.2011.89

[4]S. M. Mahalle, P. V. Ingole, “Design and Implementation of Wireless Body Area Sensor Network Based Health Monitoring System”, International Journal of Engineering Research & Technology, Vol. 2 Issue 6, pp. 105-113, June 2013.

[5]“Healthcare Monitoring System Using Wireless Sensor Network”, D. Mahesh Kumar, Department of Electronics, PSG College of Arts and Science, Coimbatore -641 014. Volume 04, Issue 01 Pages:1497-1500 (2012), ISSN: 0975-0290