Patient Healthcare Monitoring System Using IoT

Nawal Naeem & Noor Ul Amin

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of
Bachelor of Science
in
Electronics
July, 2019

Department of Electronics

Quaid-i-Azam University, Islamabad

Pakistan

Declaration

It is hereby declared that the project thesis with the title "Patient Healthcare Monitoring System Using IoT" is submitted to the department of Electronics, Faculty of Natural Science, Quaid-i-Azam University Islamabad for the award of degree of Bachelors of Science in Electronics. The project work is carried under the kind supervision of Dr. Muhammad Zia.

Nawal Naeem

Approval Certificate

This is certified that thesis entitled **"Patient Healthcare Monitoring System Using IoT"** submitted by Nawal Naeem and Noor Ul Amin in partial fulfillment of the requirements for the degree of Bachelors of Science in Electronics, has been approved and accepted by the following

____

Dr. Muhammad Zia
Associate Professor
Department of Electronics
Quaid-i-Azam University, Islamabad Pakistan

____

Dr.Syed Aqeel Abbas Bukhari
Associate Professor
Chairman
Department of Electronics
Quaid-i-Azam University, Islamabad Pakistan

Abstract

In this project, we propose a smart solution, which monitors patient health parameters and shares with the health providers and care takers. The proposed project is aimed to design a smart patient health tracking system that uses sensors to track patient health and Bluetooth Module as well as Wifi module to share with their care takers and the health providers. Our solution monitors heart rate and temperature as measures of patient health. The sensors are connected to the server, which is Arduino UNO, an embedded development kit. The microcontroller acquires data from the sensors and uploads it to the application compatible to it. The fetched data must be processed in real time fashion.

ACKNOWLEDGEMENT

I would like to extend my Special thanks to worthy Supervisor Dr Muhammad Zia who stood by me throughout the working and implementation of this work. I would also like to thank my team member and the staff members who supported me in the completion of this task and the special thanks to Musharraf Raza and Hammad Dilpazir who helped me in crucial concepts of the project work. I had a great support of my family and friends who have guided, encouraged and upheld me throughout the time. Last but not the least, I'm whole-heartedly thankful to Almighty Allah, without His blessings, nothing is possible.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

SYSTEM OVERVIEW

CHAPTER 3

Implementation

CHAPTER 4

Working

CHAPTER 5

Conclusion and References

Chapter 1

Introduction

1.1 Introduction

In the recent years usage of electronic technology has increased for the need of upholding various sectors. One of the recent trends to provide better health care facilities is Biomedical. In traditional methods, doctors play a vital role in the check-up of patients. This modern approach reduces the time consumption because normally manual methods take a lot of time in registrations, appointments and checkups, also the report generation is time taking and due to this fatigue people tend to ignore the check-ups or postpone them. In developing countries, there is a lack of resources and management due to which important things are not wielded as they should be and hence the individuals have to suffer with a lot of health issues, which after a certain time become incurable leading to the death of the patients. [3]

1.2 Problem statement

Remote health care monitoring system provides ease for chronically ill and elderly patients who could not stay at hospitals for long time. Sensors are programmed with microprocessors from which, the data is collected, analyzed and transmitted. The sensors are selected carefully in order to meet the purpose of effective communication. Health care monitoring system should be installed in every hospital. Also the hospitals from the remote areas should also be engaged with this technology. Using one parameter, health care monitoring system which is an approach to a remote healthcare monitoring system is designed that extends healthcare from the local hospital or clinic to the patient's place or home. This system serves to collect pulse rate and temperature data and fluctuation in it. The data from this scheme parameter monitoring concept is then utilized for remote detection.

1.3 Remote patient healthcare monitoring system

With the passage of time, new technology is replacing old technology and bringing betterment and advancement in human life-style. Everything is becoming smarter and easier in use for routine life. Similarly, the use of remote patient healthcare monitoring system in daily life is bringing ease and efficiency. Introduction of remote patient health care monitoring system can reduce the requirement of patients of daily check-ups.

1.4 Importance

Health monitoring system serves the purpose of healthcare and record keeping. This is also helpful in maintaining real-time record which is helpful for doctors and health providers. This system also provides convenience to the patients, care takers and the health providers.

1.4.1 Operational efficiency

To make the system efficient and maximize the gain the system is monitored again and again and is checked whether system is in working condition. The system working is staked if the internet is not in the working condition.

1.4.2 Sensor access

The data from the sensors is fetched through Bluetooth, Zigbee or Wi-Fi depending upon the module selected. The sensors are compatible to the devices with which they are connected.

1.4.3 Efficient management

The whole system is setup in such a way that the data from the patient is easily fetched and accessible by health provides. There is no loss of connection otherwise the compatibility losses.

1.5 Aims and objectives

The aim of the project is to design a prototype for maintaining the healthcare systems in hospitals so that patients may get the aid in a click. Due to this technology, distant patients also have become very close and near to the doctors and health providers.

The main objectives of this project are:

- Investigation of architectural prototype
- Collection of Physiological data
- Evaluation of potential problems

1.5.1 Investigation of architectural prototype

Using the concepts as discussed earlier, this system is designed in such a way that it takes data from the remote patients and notifies it to the health provides. It helps them get proper medication in time to avoid any alarming circumstances.

1.5.2 Collection of physiological data

Physiological data such as blood pressure, temperature, heart rate and subjective patient data is collected by peripheral devices. Peripheral devices are termed as the devices which collect data and transmit it to the health providers or third parties via telecommunication devices.

1.5.3 Evaluation of potential problems

The data of the patients is evaluated and the caregivers or health providers or patients are apprised so that the necessary steps are taken in time.

1.6 Benefits

- It provides benefits to the patients in the remote areas, senior citizens, people having mobility issues and people having chronic diseases.
- It normalizes workload and balances time.
- It reduces mobility.
- Patients enjoy smart quality of treatment.

- It is efficient in each regard.
- It saves resources and reduces cost.
- It reduces crowd in hospitals.
- More health providers analyze a single patient.

1.7 Thesis plan

The thesis plan includes various areas of knowledge in the following chapters. The second chapter carries the information of literature review. In Chapter 3 we discuss Implementation setup. Working of the project is discussed in chapter 4. Finally, we provide conclusion and future work in chapter 5.

Chapter 2

Literature Review

2.1 System overview

Healthcare monitoring system is implemented for the ease and comfort of patients. In rural and even in some urban areas, due to lack of resources, the medical equipment and services are expired and the patients who need a lot of care and need to be looked after are not getting it properly. To eliminate this factor, people have used the approach of healthcare monitoring system in Biomedical. Using this approach they designed various smart healthcare monitoring systems. These systems generally comprise of three modules, where one is health monitoring section which consists of hardware components and are used to get the various physiological parameters using different sensors. The second module is emergency alert module which is applicable when an alarming situation arises and hence the concerning hospital and family is notified. This works as a certain threshold level is fixed and if crossed, family members and doctors are alerted. The third module bases on health status prediction section, in which the patient data is kept as record and can be used at the hour of need. Different wired sensors e.g. temperature sensor LM 35, DS18b20, heartbeat sensor, body movement sensor, pulse sensor etc. are used to fetch data. Using this system, a person can be treated and can get healthcare at home [4]. Next, we discuss pulse sensor, temperature sensor, microcontroller and its applications, HC-06 Bluetooth Module and ESP8266 Wifi module.

2.2 Pulse sensor

Pulse Sensor as shown in figure 2.1 is an enduring plug and play sensor for Arduino to measure heart rate. It is used in a lot of fields e.g. athletes, artists and students or developers use it to develop games and want to assimilate current heart rate data into their projects. The sensor hooks up with the finger or earlobe of the patient and is plugged right into the Arduino using some jumper wires. It also encompasses an open source application in order to graph the heart rate in real time and keep the record.

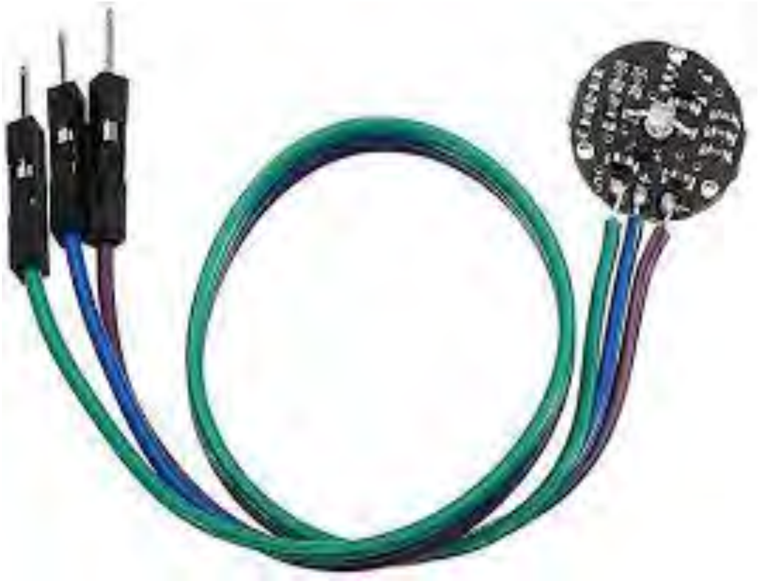


Figure 2.1: Pulse Sensor

Pulse Sensor Kit shown in figure 2.2 comprises of the following parameters:

1. It comprises of a Color-Coded cable which is 24 inch long, having (male) header clampers. This makes it very easy to embed the sensor into the project, and connects to an Arduino. This does not require soldering.
2. The Ear Catcher is sized according to the sensor perfectly. It is hot glued to the back of the sensor so that is can easily be worn on the earlobe.
3. 2 Velcro Dots. These are also perfectly sized according to the sensor. If we want to wrap Velcro strap round the fingertip, Velcro dots are useful.
4. 3 Transparent Stickers. On the front side of the Pulse Sensor, these stickers are used to prevent from oily fingers and sweaty earlobes.
5. Around the outside edge, Pulse Sensor consists of 3 holes which make it facile to stitch it into anything [1].

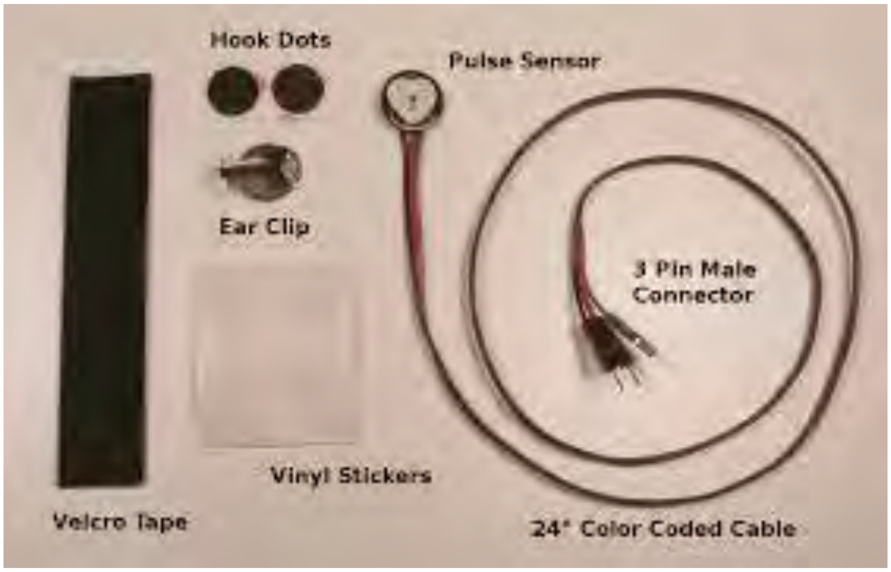


Figure 2.2: Pulse Sensor Components

2.3 DS18b20 Temperature Sensor

DS18b20 temperature shown in figure 2.3 comes up with a lot of such features. It is a programmable digital sensor which uses 1-wire method to communicate and operates at voltage range of 3V to 5V. Its temperature range is -55°C to +125°C with an accuracy of ±0.5°C. Its programmable Output resolution is from 9-bit to 12-bit. At 12 bit, its conversion time is 750ms. It contains a unique 64-bit address which enables multiplexing. It also has programmable alarm options. It is available as To-92, SOP and even as a waterproof sensor.

Pin Configuration

Sr no.	Pin Name	Description
1	Ground	Connect to ground of the circuit
2	Vcc	Powers the sensor
3	Data	Gives the temp. value using -1 wire method

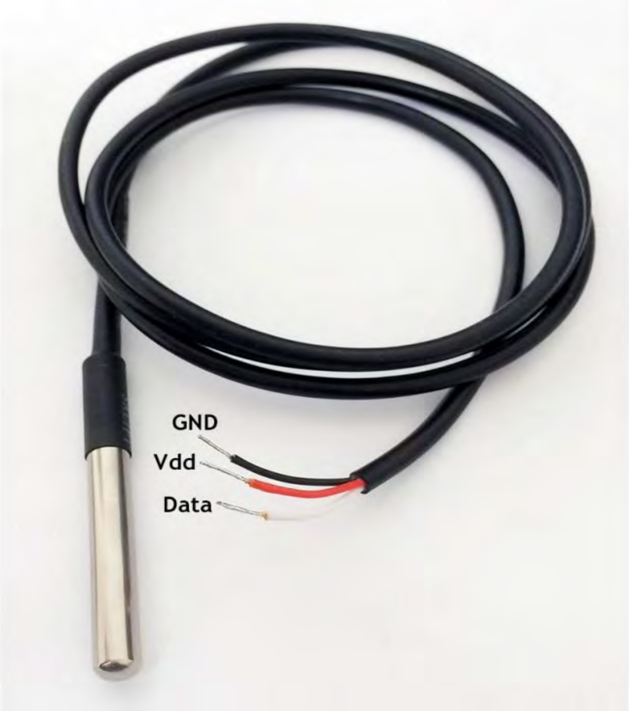


Figure 2.3: temperature sensor

2.4 Microcontroller

Electronic devices have become more functional and are cheap, compact and flexible as compared to their predecessors that were costly, covered more area and were able to conduct fewer tasks. Specialists always have strived to put forward a revolution to the electronic industry that involves minimum effort and gives maximum output. In the electronic industry microcontrollers are introduced in order to meet the purpose of making tasks handy. Microcontrollers make devices work according to our requirements and demands and are widely used in embedded systems. Arduino UNO as shown in figure 2.4 comprises of 14 digital I/O pins, 6 analogue pins, Atmega328 microcontroller and a USB interface. It reinforces serial type of communication by utilizing transmission (Tx) and receiver (Rx) pins.

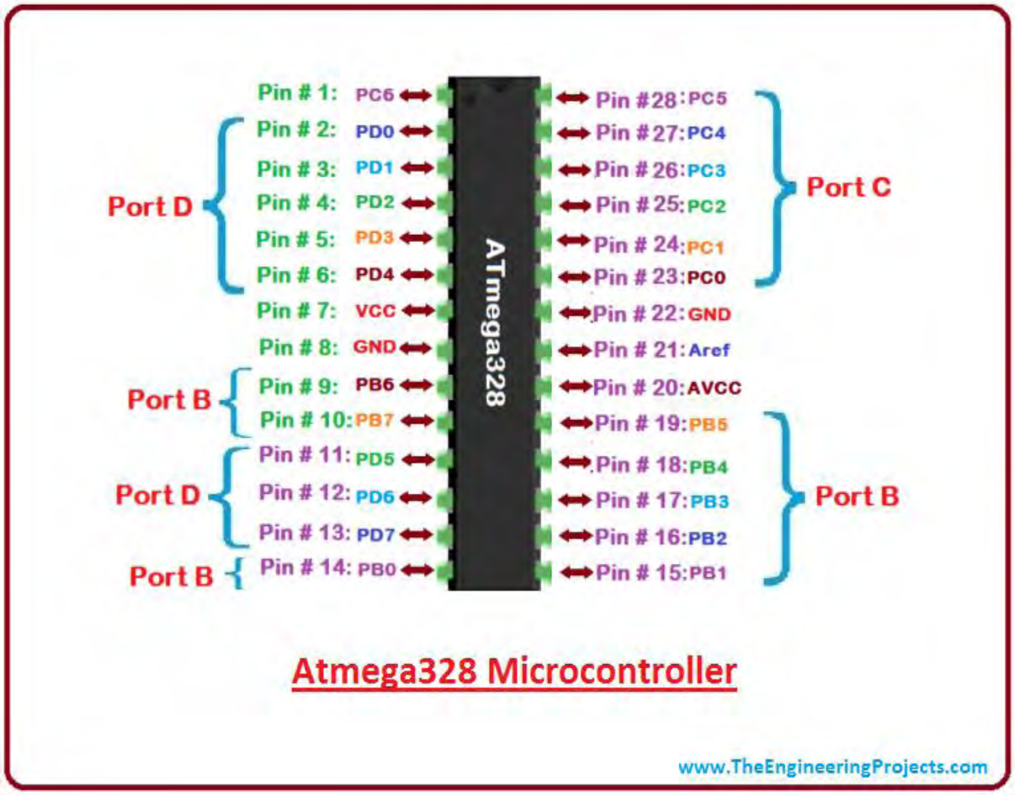


Figure 2.4: Pin configuration

There exist numerous versions of Arduino boards e.g. Arduino Due, Arduino Mega, Arduino Leonardo etc. and most common version of them is Arduino UNO and Arduino Mega. Arduino UNO as shown in figure 2.5 is the best option if one is interested to produce a project related with IoT, Robotics, Embedded systems or digital electronics. It is easy and cost effective. Since, it is an open-source platform so anyone can optimize and modify boards for better functionality because all the equipment is readily available. IDE (Integrated Development Environment) the software used by Arduino devices and its programming is done using C++ and C.

It is confusion among people regarding Arduino and Microcontroller. The former is an operating system with 40 pins chip and it has built-in processor and the hindmost is a board having micro-controller in the base of the boot loader and permits the access to I/O pins which burns the program very easily.

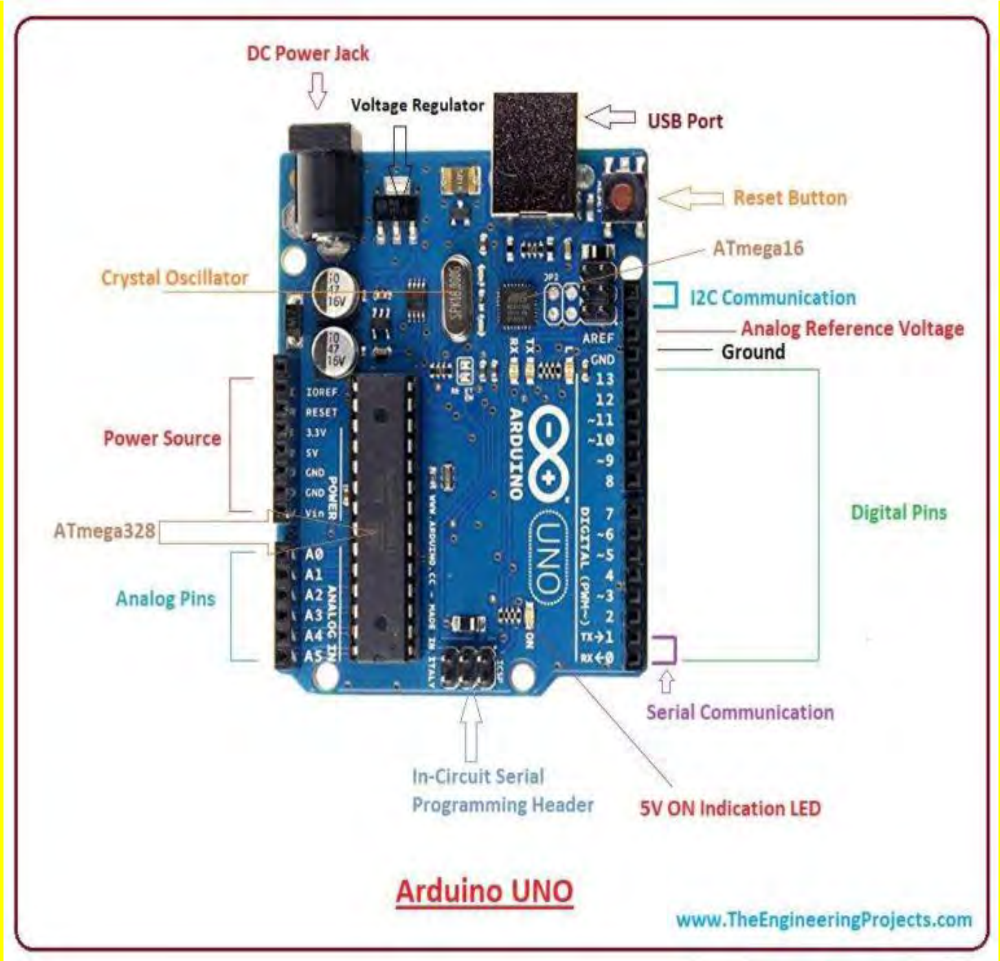


Figure 2.5: Arduino UNO

Introduction to Arduino Uno

- Arduino Uno is an open-source platform which is mainly based on AVR microcontroller Atmega328 which is a microcontroller board.
- First project of Arduino commenced in Interaction Design Institute Ivrea in 2003 by Massimo Banzi and David Cuartielles. Their intention was to provide a flexible and rather cheap way to students and professionals in order to sway numerous devices in real world.
- Arduino Uno, at present, comes with 6 analog input pins, a USB interface and 14 I/O digital pins to build connection with external devices or circuits. Out of 14 I/O pins, 6 pins are used for Pulse Width Modulation output.
- It empowers artificers to benchmark and perceive the electronic appliances in the real world.
- The Arduino board can directly be connected to PC via USB cable to send the code to controller by using IDE software and it contains all the attributes needed to run the controller which is developed in order to program Arduino.

IDE is compatible with environments like Windows, Linux, and MAC; where, Windows environment is most commonly used.

- AC to DC adopter or batteries can also be employed to supply power to the board, instead of USB.
- In terms of usage and functionality, Arduino Uno board is analogous to other boards of Arduino family.
- Different versions of Uno boards are available where Arduino UNO and Arduino Nano V3 are the most demanded versions with Atmega328 8-bit AVR Atmel microcontroller with 32 KB RAM memory.
- Micro SD card is employed in the board in order to store more data when the nature and functionality of the task goes compound.

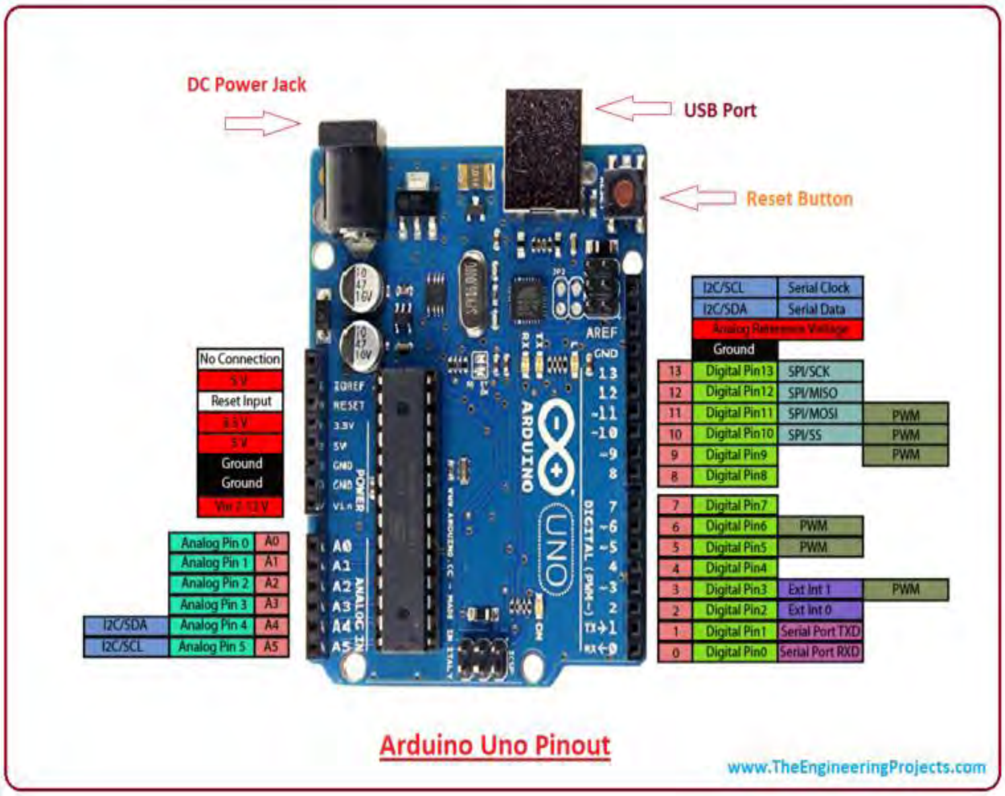


Figure 2.6: Arduino UNO Pinout

Applications

Arduino UNO has vast applications range. People use Arduino boards to develop electronic equipment that are used in technical work. Some applications of Arduino are as follows.

- Security System
- Home Automation
- Traffic Control System
- Robotics and Digital Electronics
- Parking Lot Counter
- Traffic Light Timer
- Medical Equipment
- Emergency Railway Lights
- Industrial Automation

A number of other microcontrollers are also accessible in market which is more substantial and cost effective as compared to Arduino. So, why should Arduino UNO be preferred?

The answer to this question is, Arduino has a link with a vast electronic industrial community that is developing and spreading the knowledge all over around the globe. Arduino has a quick support concerning to technical facets of electronic projects. When one elects Arduino board over other microcontrollers, one does not need to worry about any miscellaneous peripherals and gadgets for a lot of required functions are easily available on the board which makes one's work cost effective and also make one get rid of a lot of technical proficiencies [2].

2.5 HC-06 Bluetooth Module

HC-06 Bluetooth module as shown in figure 2.7 is a Bluetooth chip based on protocol standards. It needs voltage control to operate and its Operating voltage: is 3.3V. Its Baud rates are 1200, 2400, 4800, 9600, 19200, 38400, etc. and can also be set by users. Its Size is 28mmx15mmx2.35mm. While pairing its current is 20 ~ 30 mA and 8 mA after it is paired. It is used to meet various purposes e.g. for the GPS navigation system, mining control system, industrial site, utility meter reading systems etc. It can seamlessly connect with various Bluetooth devices e.g. notebook computer, computer and Bluetooth adapters, PDAs etc.

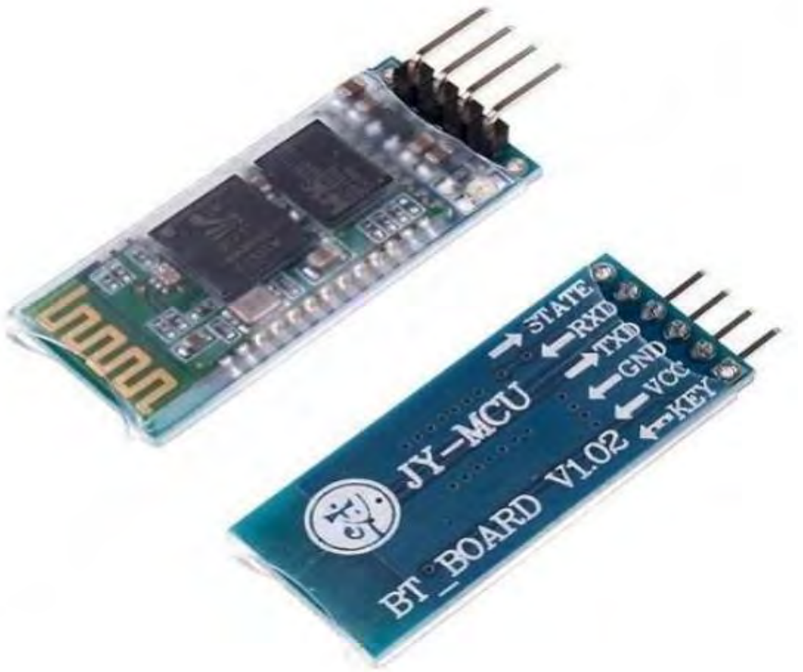


Figure 2.7: HC-06 Bluetooth Module

Now, we speak about the Base Module of Bluetooth HC-06 which is stable and of reliable quality. Since the reverse power does not work so, it is powered in anti-reverse manner. Its input voltage on which it is functional is 3.6-6V and it prohibits connection at 7V and above. For various modules, the common Bluetooth module can be installed e.g. HC05, HC06, HC07, BC04 and other pin definitions compatible to Bluetooth module. Pin-outs include TX, RX, STATE (if any) pin, KEY (if any) common pin, and in this way it is quite easy to use. The size of the module is 1.52*3.57 (cm).

Datasheet:http://www.sgbotic.com/products/datasheets/wireless/hc06_datasheet.pdf [5]

2.6 ESP8266 Wi-Fi Module

ESP8266 as shown in figure 2.8 is a Wifi module, produced in Espressif Systems. It is very cost effective microchip with Transmission Control Protocol/Internet Protocol stack having microcontroller capabilities. It is chiefly used in IoT based systems.
It comes with the characteristics as follows:

- General Purpose I/O(16 pins)
- 2.4Ghz Wi-Fi
- Pulse Width Modulation
- Universal Asynchronous Receiver Transmitter

- Analog to Digital converter
- Inter-integrated circuit serial communication protocol

Pin Configuration

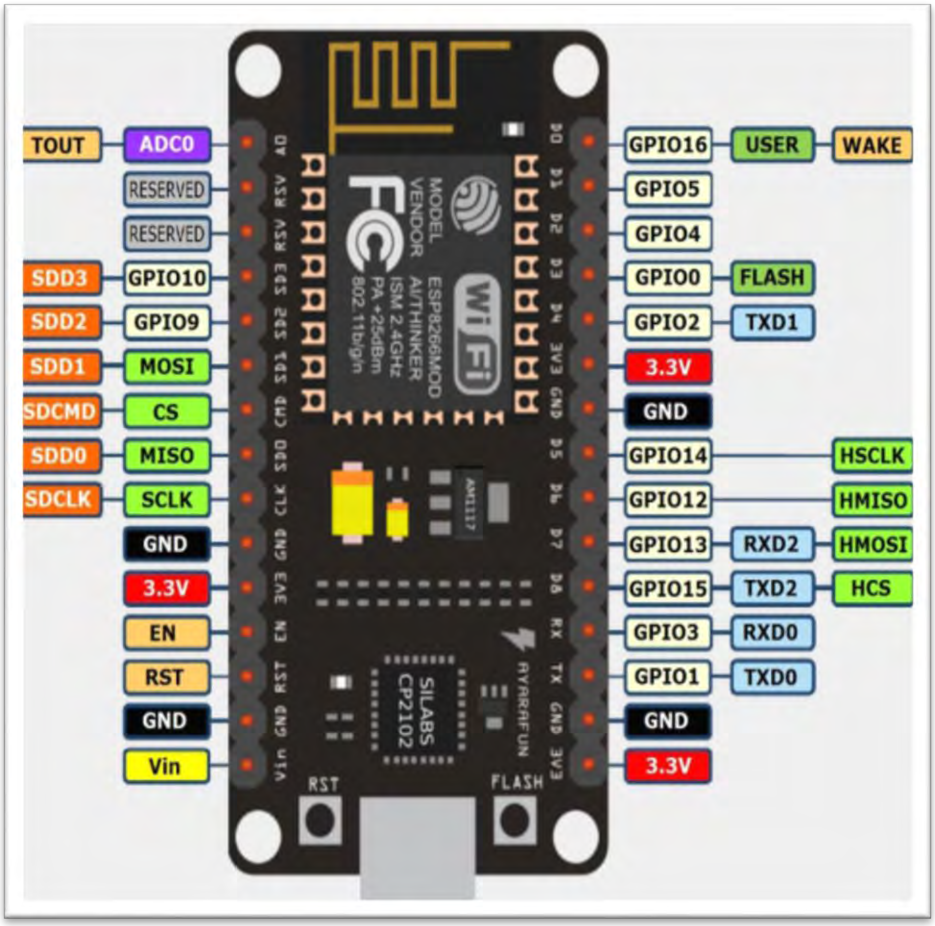


Figure 2.8: ESP8266 Wi-Fi Module

2.7 Machine to machine communication

Machine to machine communication has an aim to establish a link between different machines i.e. PC to PC, mobile to PC, mobile to mobile, mobile to microprocessor and so on so forth. It refers to direct relationship between electronic devices using wired or wireless communication channels. To accomplish this task of the establishment of connections, various concepts and IPs are used. There are different ways to make machine to machine communication successful such as, socket programming or Message queuing telemetry transport (MQTT).

2.7.1 Socket programming

Socket programming is way of connecting two machines using internet protocol. One socket listens to an IP while the other establishes a connection. The socket at the server side acts as a listener, while that at the client side tends to build a connection.

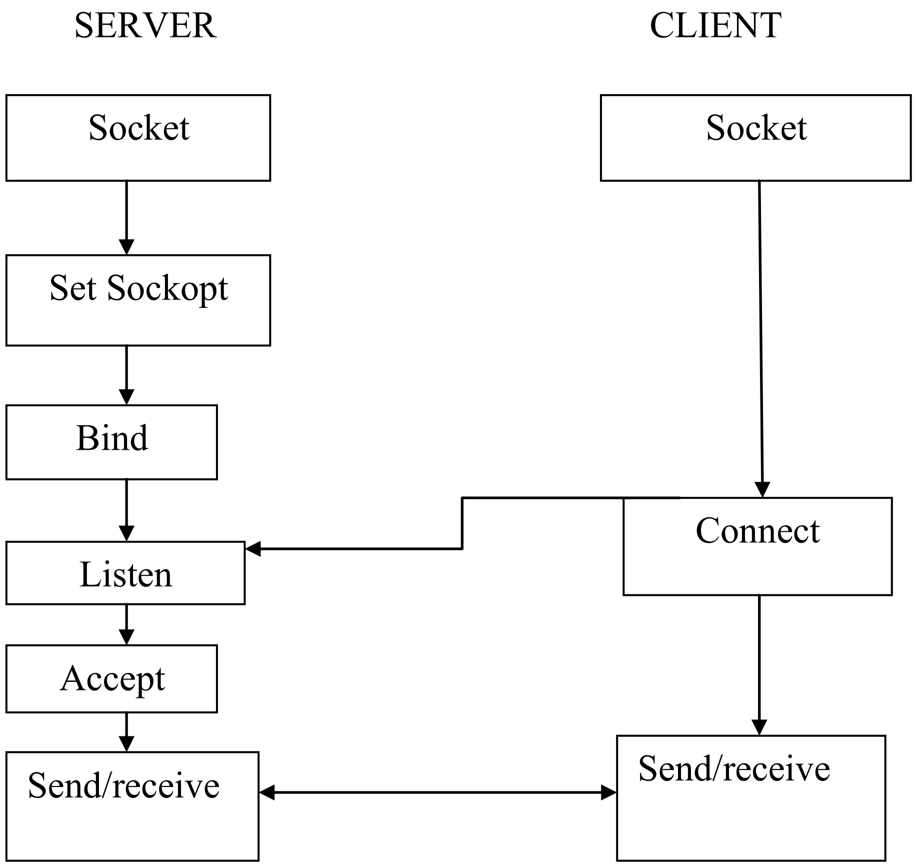


Figure 2.9: server to client

Stages for server

Socket Creation: socket creation includes the following parameters

- **Sockfd:** It is an integer type of socket descriptor.
- **Domain:** It is communication domain. It may have either IPv4 or IPv6 protocol.
- **Type:** It carries the information of communication type e.g. TCP or UDP.

- **Protocol:** The protocol value of internet protocol is 0, which is same as the number that appears on field of protocol in the header of a packet of IP.

Setsockopt: It is helpful in reuse of port and address. It prevents from error such as “address not valid”.

Bind: This step is performed after socket creation. It is useful to bind socket with address and port number which is specified in address.

Listen: This keeps the server socket in an inactive mode, where server waits for the client and establishes a connection when client approaches.

Accept: At this step, the connection is produced between server and client and they both are ready for the data transfer.

Stages for Client

Socket connection: The procedure is exactly the same as in the server side.

Connect: Connect function calls addresses specified in the address [7].

2.7.2 Message Queuing Telemetry and Transport (MQTT)

It is a machine to machine IOT connectivity protocol. It is small in size, using low operating power, has minimized data packets and makes the distribution of information efficient to one or more receivers.

The procedure of MQTT comprises of basic concepts, which are as follows:

- Publish/ subscribe
- Message
- Broker
- Topic

In the system, where publish/subscribe concept is used, the device generates or publishes a message on the topic. The message is information or data which is to be conveyed.

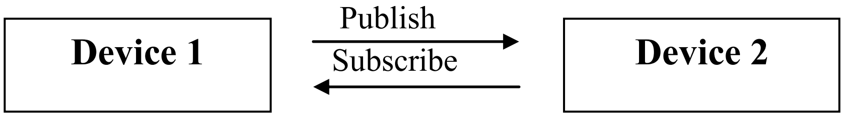


Figure 2.10

The publisher is the end, from where the data or information is published on the broker or database, and the subscriber can access the data by subscribing to the broker. All the data is fed on the broker which is accessible by both the publisher and subscriber. Also, the publisher can act as a subscriber and a subscriber can be a publisher at the same time, depending upon the requirements of knowledge of data. Both the subscribers and publishers can be one or more than one, depending upon the number of users. For instance, speaking about this project, a number of users can subscribe to broker also, a broker may have more than one publishers.

- **How the data is published on the broker**
  The data is published on the broker via the help of a topic. The **Topic** can be a random name or word e.g., test, check, measure, pressure etc. Broker uses this topic as a filter in order to make decision whom to give subscription.
- **How the data is subscribed by the broker**
  The subscribers request broker when they require data. The broker first realizes or understands whether the subscriber is a valid subscriber. If the result is true and the subscriber is a valid one, the broker provides the required data to the subscriber [6].

2.8 Communication between hardware and software

The hardware of the system comprises of a number of parameters, such as, microcontroller, sensors, mobile phone and PC. A number of software is required which provides a link between the whole setup. When the data is fetched from the patient using sensor, it is then fed on the microcontroller or application from where it travels to the database. The processed data remains in the database and can be accessed at any time. This data is on a distance of a single click. The data can be volatile, i.e. if not required, can be removed. It is kept in the database, so that, whenever the doctor wants to access it, he can easily get it. The android application is designed in such a way that it carries all the required parameters on it. It is handy to use, and readily extract out data.

The android application is an easy way to make things easy to handle. Since, in today's life, people have become so comfortable with the android applications, that if the database is made compatible with the android application, it is easy to use for everyone i.e., from doctor to a layman.

Chapter 3

Implementation

3.1 Implemented system overview

Implementation knowledge with results of respective execution and the software design of the system with the hardware components with their specifications are discussed in the proceeding sections. The entire hardware assembly is arranged and developed in a fashion that it connects with its software successfully. This project comprises of two modules. One is the flow of data from sensors to the PC through Arduino and the other is from Arduino to android Application via HC-06 Bluetooth module and ESP8266 Wifi module. They communicate with each other in order to share the information.

3.2 Connections of modules

3.2.1 Connection of Pulse Sensor with Arduino

Pulse Sensor, as discussed earlier, establishes connection with Arduino UNO via the pins. It has 3 main pins: positive, negative and S. Positive pin operates at 5V, negative pin is at ground and S is with A0 pin of the Arduino.

3.2.2 Connection of HC-06

The positive pin of HC-06 is connected to 3.3V supply pin of ESP8266 and ground pin of HC-06 is connected to ground pin of ESP8266. TX pin of HC-06 is connected to Pin7 of the Arduino UNO and RX pin of HC-06 is connected to Pin8 of the Arduino as specified in the code.

3.2.3 Connections of Temperature Senor (DS18b20)

The positive pin of DS18b20 is connected to 3.3V supply pin of Arduino UNO through a resistance of 330 ohms. The data pin of the sensor is connected to pin4 of the Arduino and the ground pin of the sensor is connected to the ground pin of Arduino.

3.3 Android Application

The android application for the project can be designed using any environment e.g. Blynk, android studio, MIT app inverter, eclipse etc. Android studio is an integrated development environment used specifically for the development of android applications. It can be downloaded and used both in windows and Linux operating systems. It has replaced the Eclipse android development which was used as an android development platform earlier. The Blynk application interfaced in the project is designed using Blynk platform which is an easy application interfacing and development platform in which modules can be easily interfaced. MIT app inverter is a user friendly development environment. Any environment which is used for this purpose has two programming platforms, i.e. front end which includes activities related to the designing of menu, buttons and icons etc., whereas back end includes activities related to the data reading, Bluetooth and Wifi interfacing programming which get the data from the sensors used in the project and show it. Also the data can be updated and retrieved all the time it is needed. Android application development platforms provide a flexible and easy way for the designing and development of android applications [8].

3.4 Blynk Application

Blynk is an application platform used for interfacing different modules and getting data. To design Blynk application in this project system, first of all go to Blynk platform and give a name for the project application. Then click on the device choose and select NodeMcu ESP8266, then set the connection type to Wifi then click on create button, an authentication code will sent to email which is used in programming later. Then click on the gauge and do the required setting, pin number and limit etc. By this way we can do the required setting for different buttons added and then Wifi module is interfaced to get data into the application.

Chapter 4

Working

4.1 Working statement

The working principles of this project depend upon the need of people. People who suffer from chronic diseases or diseases in which patients cannot move from one place to another need a permanent solution. And the solution to this problem is attempted to build in this project.

4.2 Work Scope

Since, electronics has become an emerging field. On industrial level, the projects of data monitoring are most demanded, because it makes lives handy and convenient. This is the reason why this title is taken up for this project.

4.3 Working methodology

So, talking about the working methodology of this gadget, we have to acknowledge its importance first. To understand the working of this product is very handy. One just has to reach out to the problem in a single click. The working of this project is discussed here.

The main purpose of this project is to collect the data from patients. This can only be done with the help of sensors. Sensors, as discussed earlier, can be wired or wireless. Wired sensors are easy to implement as the sensor used here is a pulse sensor which measures up heart rate and temperature sensor which measures up temperature of the body. The sensors are connected to the Arduino, when the finger is placed on the sensors the Arduino begins to collect data from the sensors and displays it on the serial monitor.HC-06 Bluetooth module is connected to the Arduino can send the data of sensor to any platform where Bluetooth module can be operated.

In this project, the Bluetooth module connects the Arduino and android application in order to feed data on the application. The ESP8266 Wifi module is also connected with Arduino and the data can be collected through Blynk application which is an android application interfaced and developed through Blynk platform. The application which collects data through HC-06 Bluetooth module is used when either locally data monitoring is required or when internet is not properly working, where as Blynk application which collects data through Wifi module is used to get data at very long distance , through which data can be monitored far away.

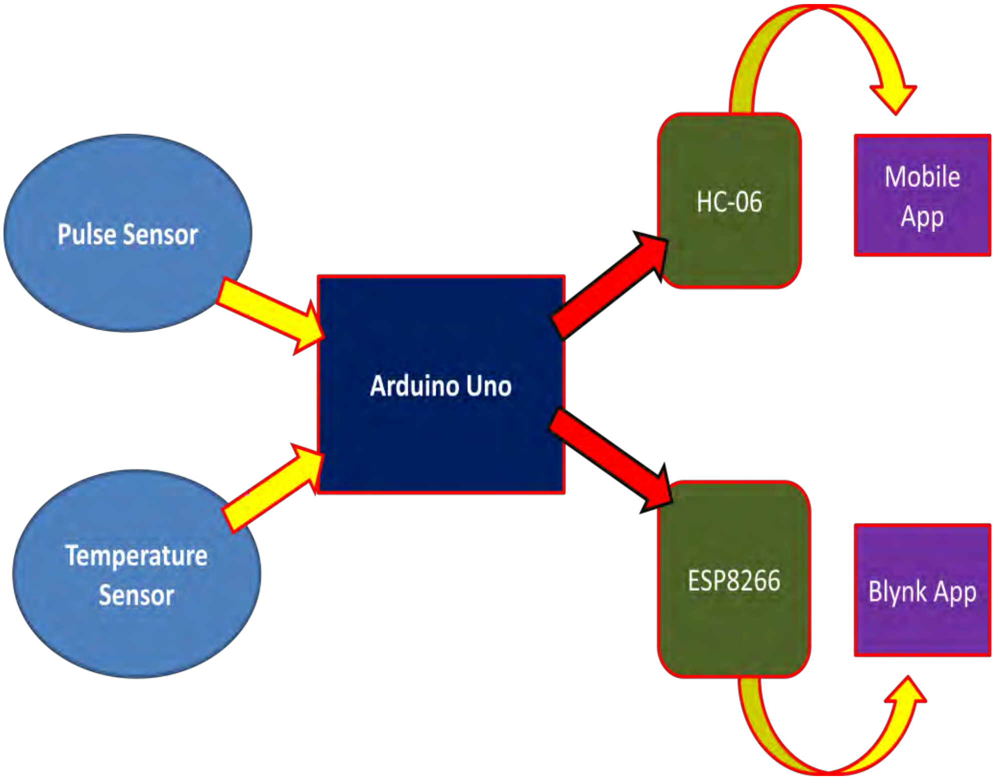


Figure 4.1(a): System Block Diagram

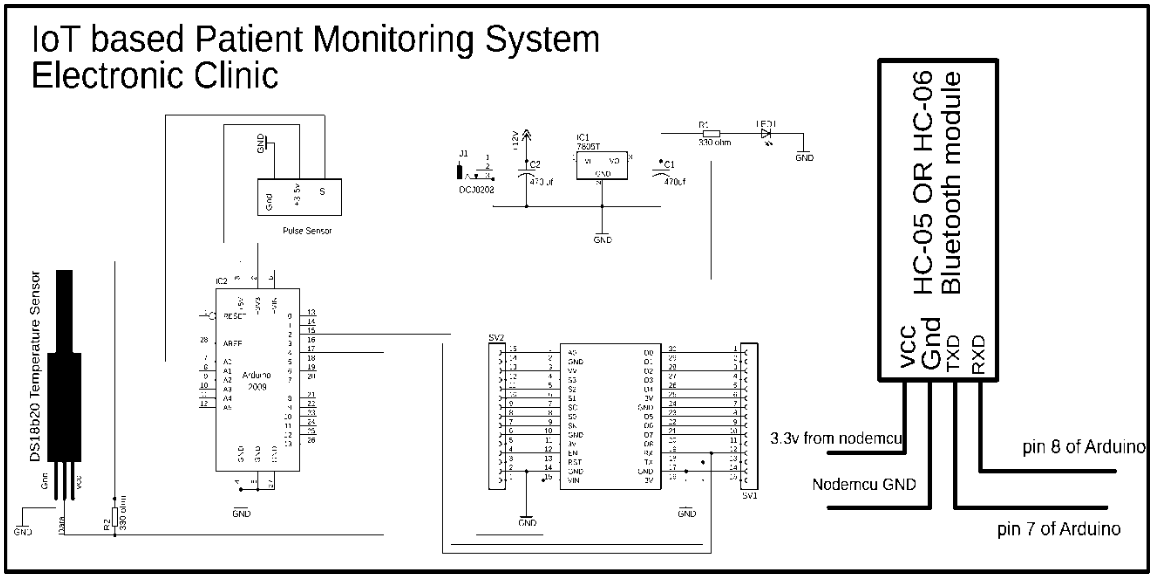
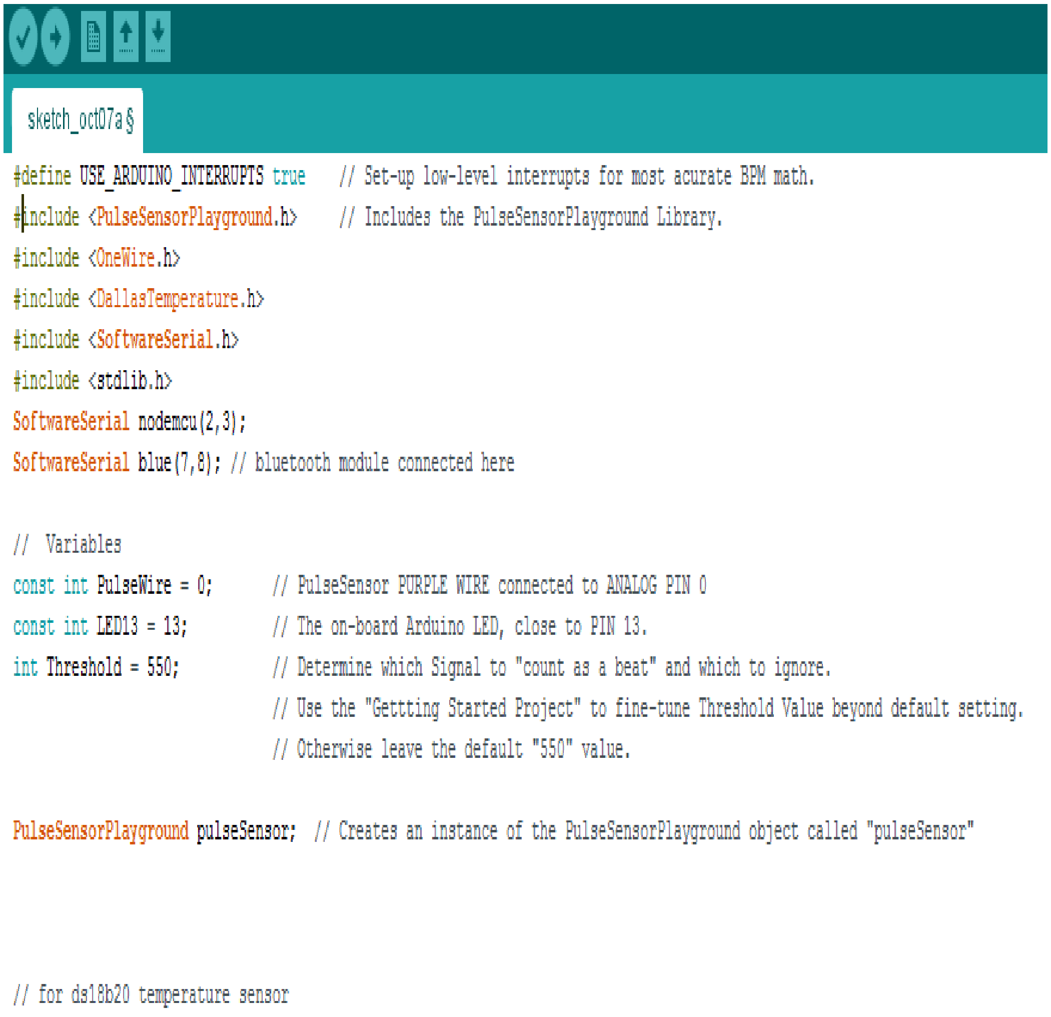
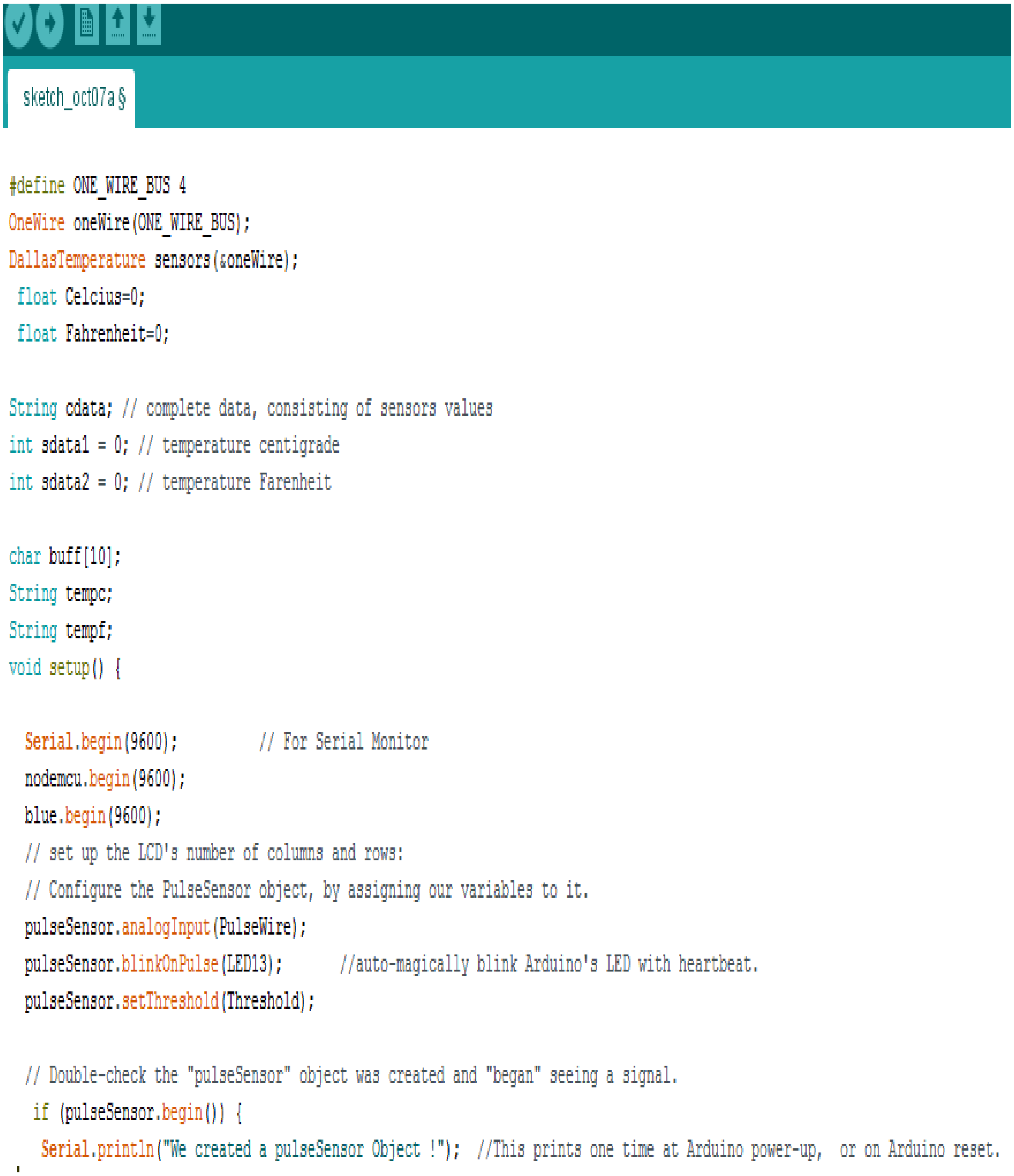


Figure 4.1(b): System Schematic



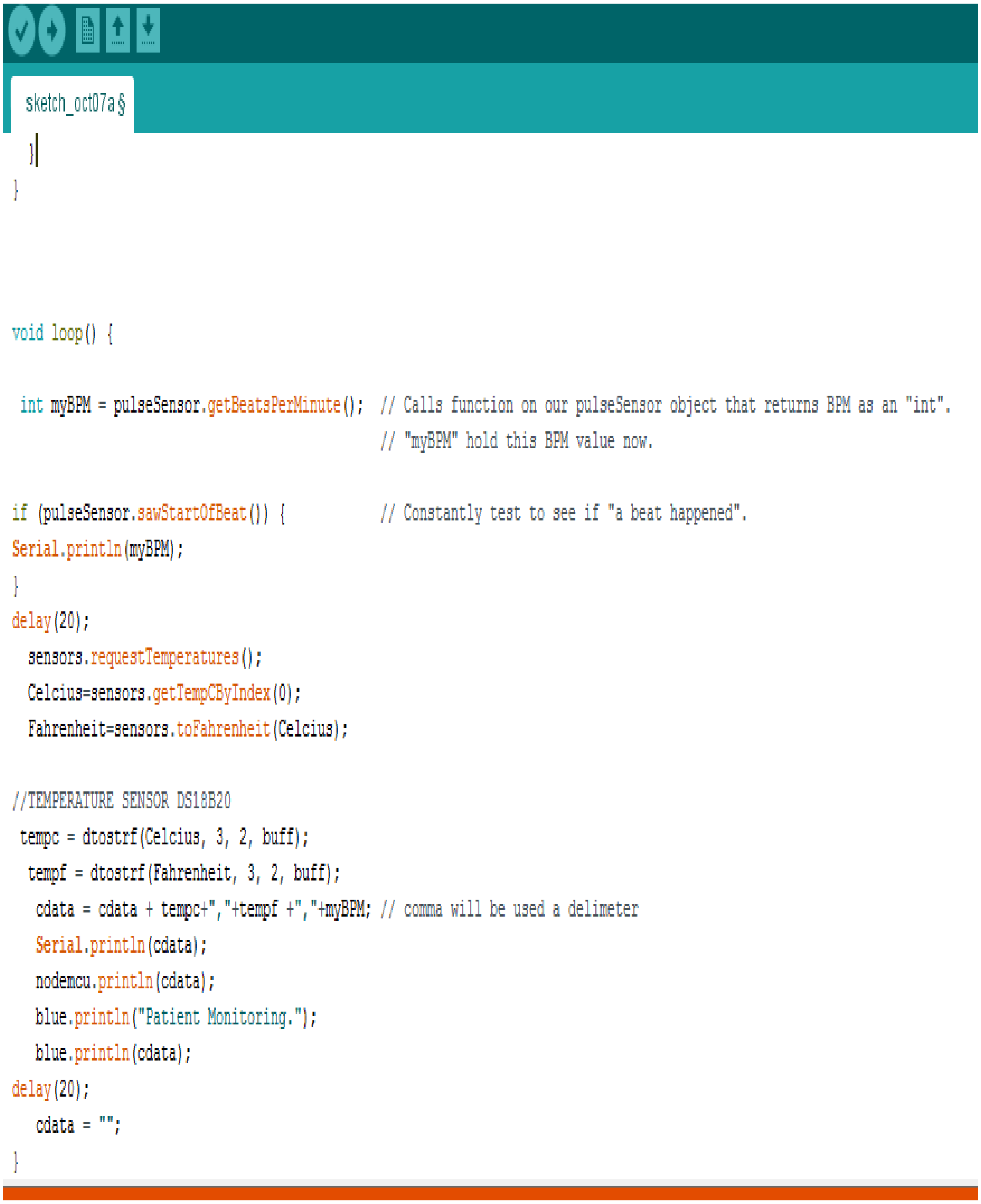
```
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include <stdlib.h>
SoftwareSerial nodemcu(2,3);
SoftwareSerial blue(7,8); // bluetooth module connected here

//  Variables
const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.
int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.
                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.
                               // Otherwise leave the default "550" value.

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"



// for ds18b20 temperature sensor
```

Figure 4.2(a): code for data from sensors



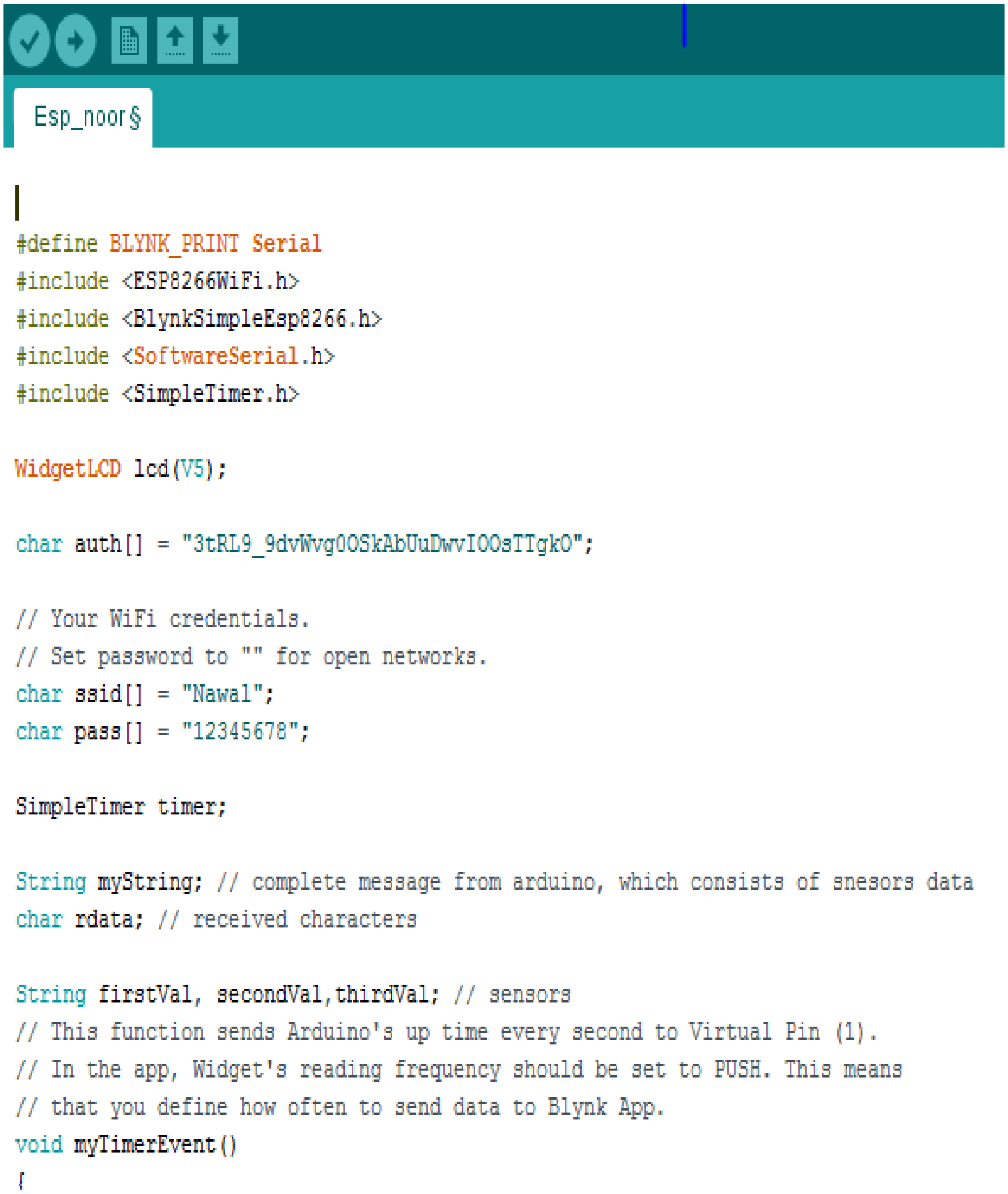
```
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
 float Celcius=0;
 float Fahrenheit=0;

String cdata; // complete data, consisting of sensors values
int sdata1 = 0; // temperature centigrade
int sdata2 = 0; // temperature Farenheit

char buff[10];
String tempc;
String tempf;
void setup() {

  Serial.begin(9600);         // For Serial Monitor
  nodemcu.begin(9600);
  blue.begin(9600);
  // set up the LCD's number of columns and rows:
  // Configure the PulseSensor object, by assigning our variables to it.
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.
  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" object was created and "began" seeing a signal.
   if (pulseSensor.begin()) {
    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.
```

Figure 4.2(b): code for data from sensors



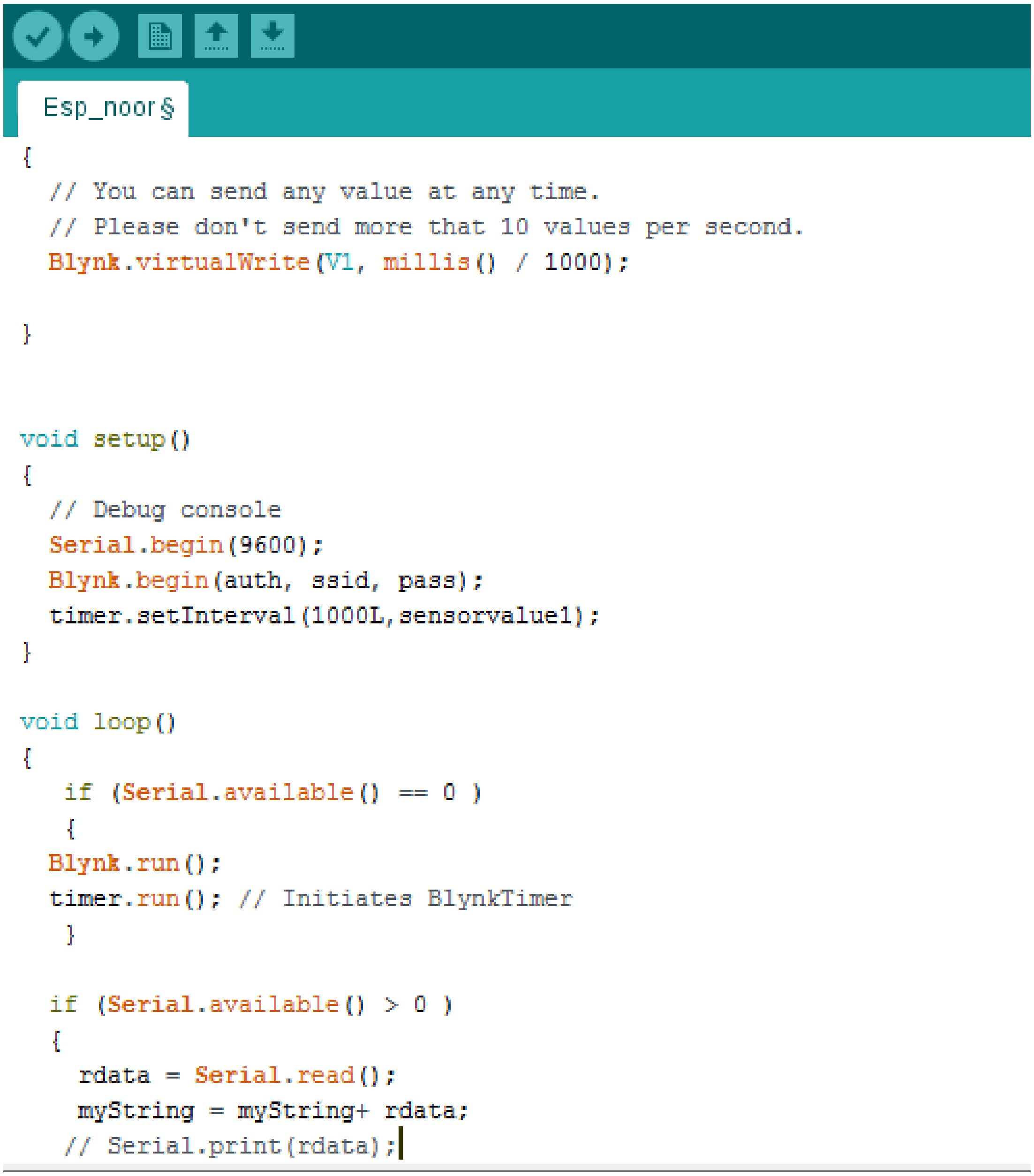
```
  }
}



void loop() {

 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".
                                               // "myBPM" hold this BPM value now.

if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".
Serial.println(myBPM);
}
delay(20);
  sensors.requestTemperatures();
  Celcius=sensors.getTempCByIndex(0);
  Fahrenheit=sensors.toFahrenheit(Celcius);

//TEMPERATURE SENSOR DS18B20
 tempc = dtostrf(Celcius, 3, 2, buff);
  tempf = dtostrf(Fahrenheit, 3, 2, buff);
   cdata = cdata + tempc+","+tempf +","+myBPM; // comma will be used a delimeter
   Serial.println(cdata);
   nodemcu.println(cdata);
   blue.println("Patient Monitoring.");
   blue.println(cdata);
delay(20);
   cdata = "";
}
```

Figure 4.2(c): code for data from sensors



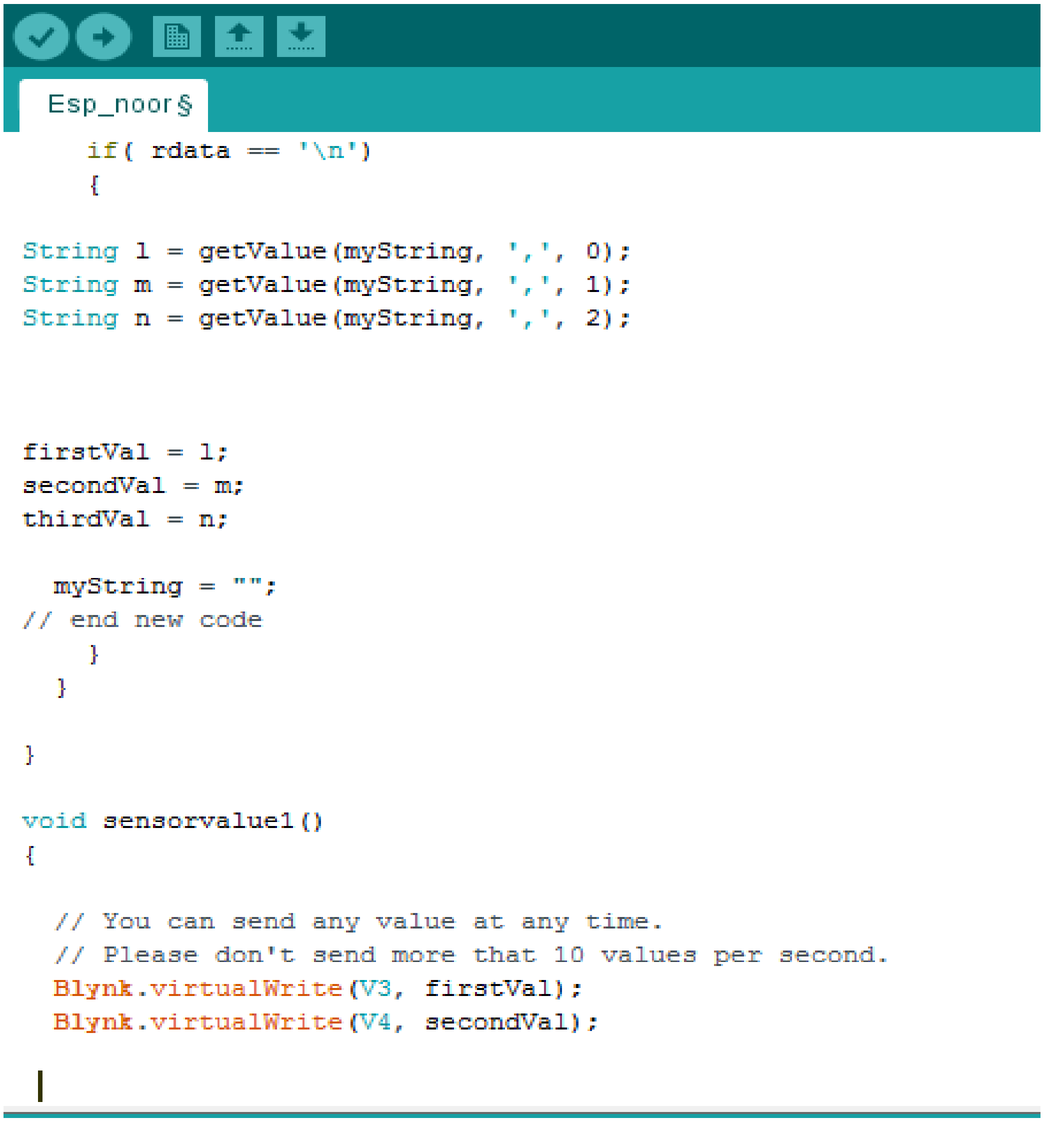
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>

WidgetLCD lcd(V5);

char auth[] = "3tRL9_9dvWvg0OSkAbUuDwvIOOsTTgk0";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Nawal";
char pass[] = "12345678";

SimpleTimer timer;

String myString; // complete message from arduino, which consists of snesors data
char rdata; // received characters

String firstVal, secondVal,thirdVal; // sensors
// This function sends Arduino's up time every second to Virtual Pin (1).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void myTimerEvent()
{
```

Figure 4.3(a): code for ESP Module with Blynk Application



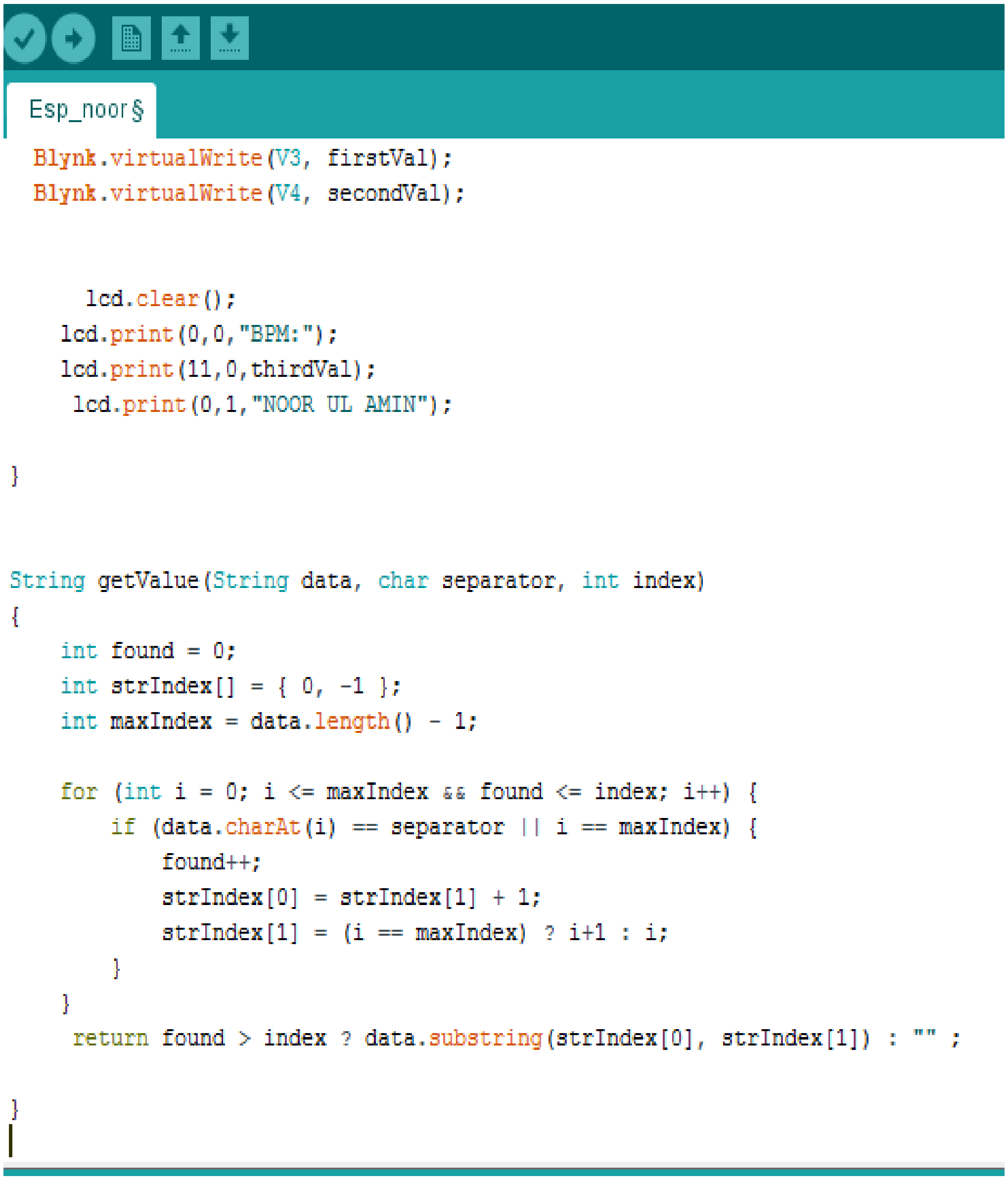
```
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V1, millis() / 1000);

}


void setup()
{
  // Debug console
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L,sensorvalue1);
}

void loop()
{
   if (Serial.available() == 0 )
   {
  Blynk.run();
  timer.run(); // Initiates BlynkTimer
   }

  if (Serial.available() > 0 )
  {
    rdata = Serial.read();
    myString = myString+ rdata;
   // Serial.print(rdata);
```

Figure 4.3(b): code for ESP Module with Blynk Application



```
    if( rdata == '\n')
    {

String l = getValue(myString, ',', 0);
String m = getValue(myString, ',', 1);
String n = getValue(myString, ',', 2);



firstVal = l;
secondVal = m;
thirdVal = n;

  myString = "";
// end new code
    }
  }

}

void sensorvalue1()
{

  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V3, firstVal);
  Blynk.virtualWrite(V4, secondVal);

```

Figure 4.3(c): code for ESP Module with Blynk Application



```
  Blynk.virtualWrite(V3, firstVal);
  Blynk.virtualWrite(V4, secondVal);


      lcd.clear();
    lcd.print(0,0,"BPM:");
    lcd.print(11,0,thirdVal);
     lcd.print(0,1,"NOOR UL AMIN");

}


String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
     return found > index ? data.substring(strIndex[0], strIndex[1]) : "" ;

}
```

Figure 4.3(d): code for ESP Module with Blynk Application

4.4 Project deliverables

The results of this project is smart health care monitoring system based on IoT which comprises of an android application, Blynk application, a pulse sensor, temperature sensor, and a complete setup.

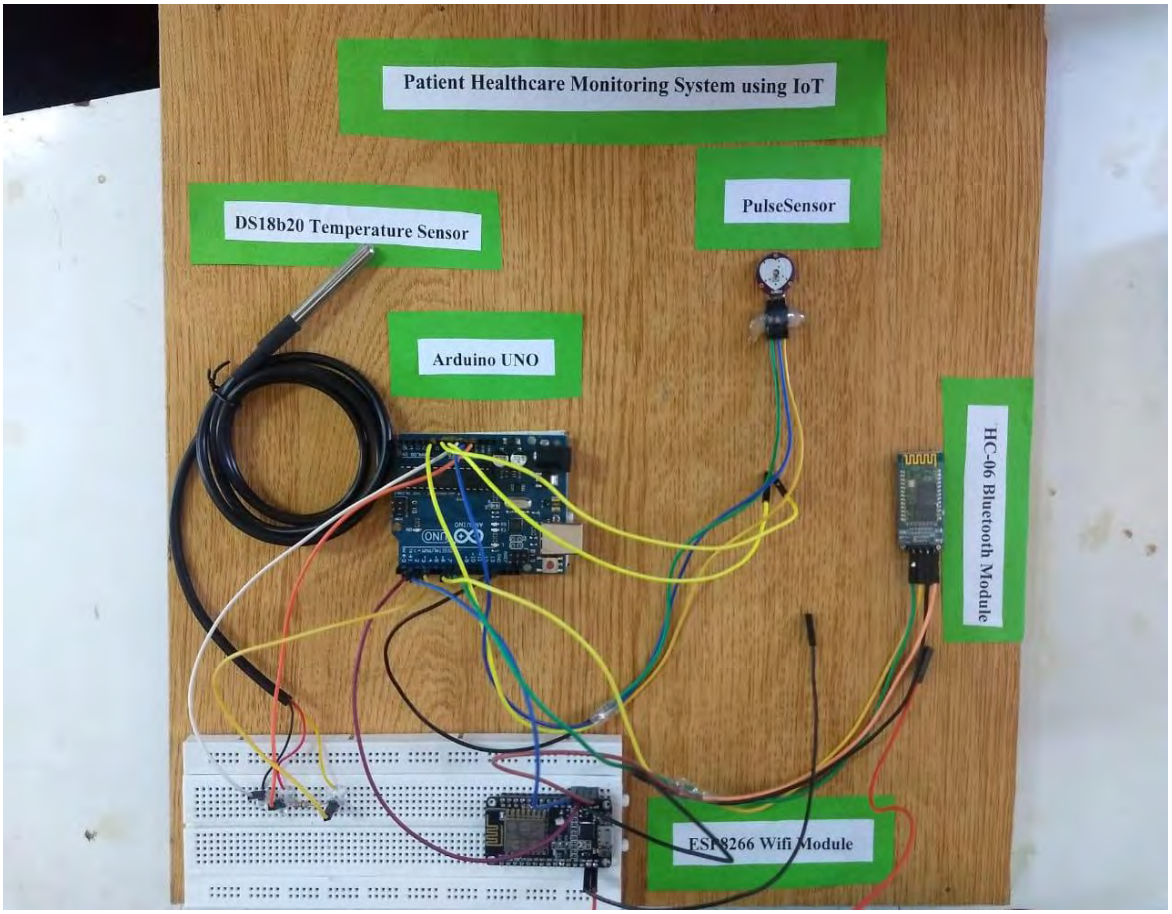
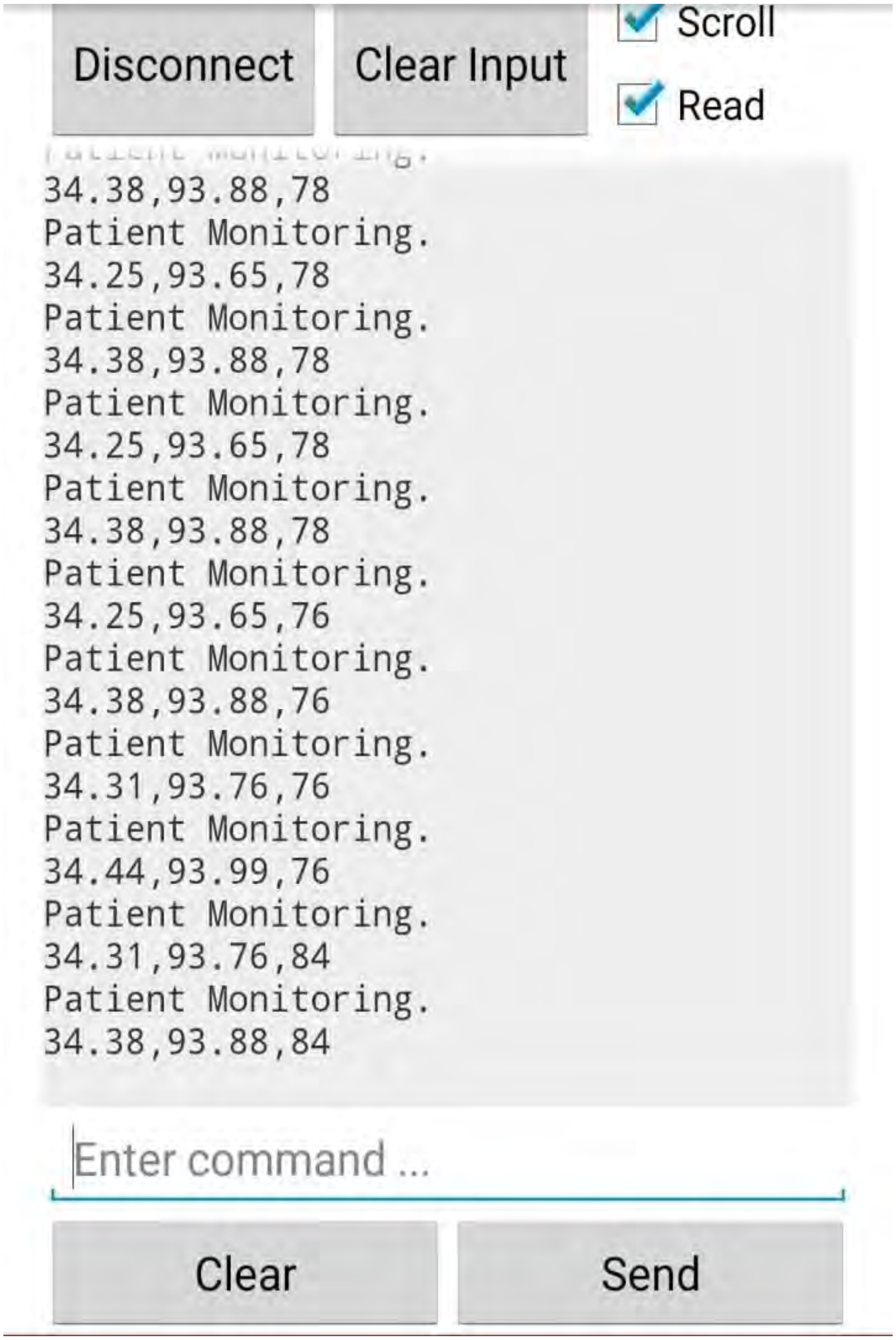
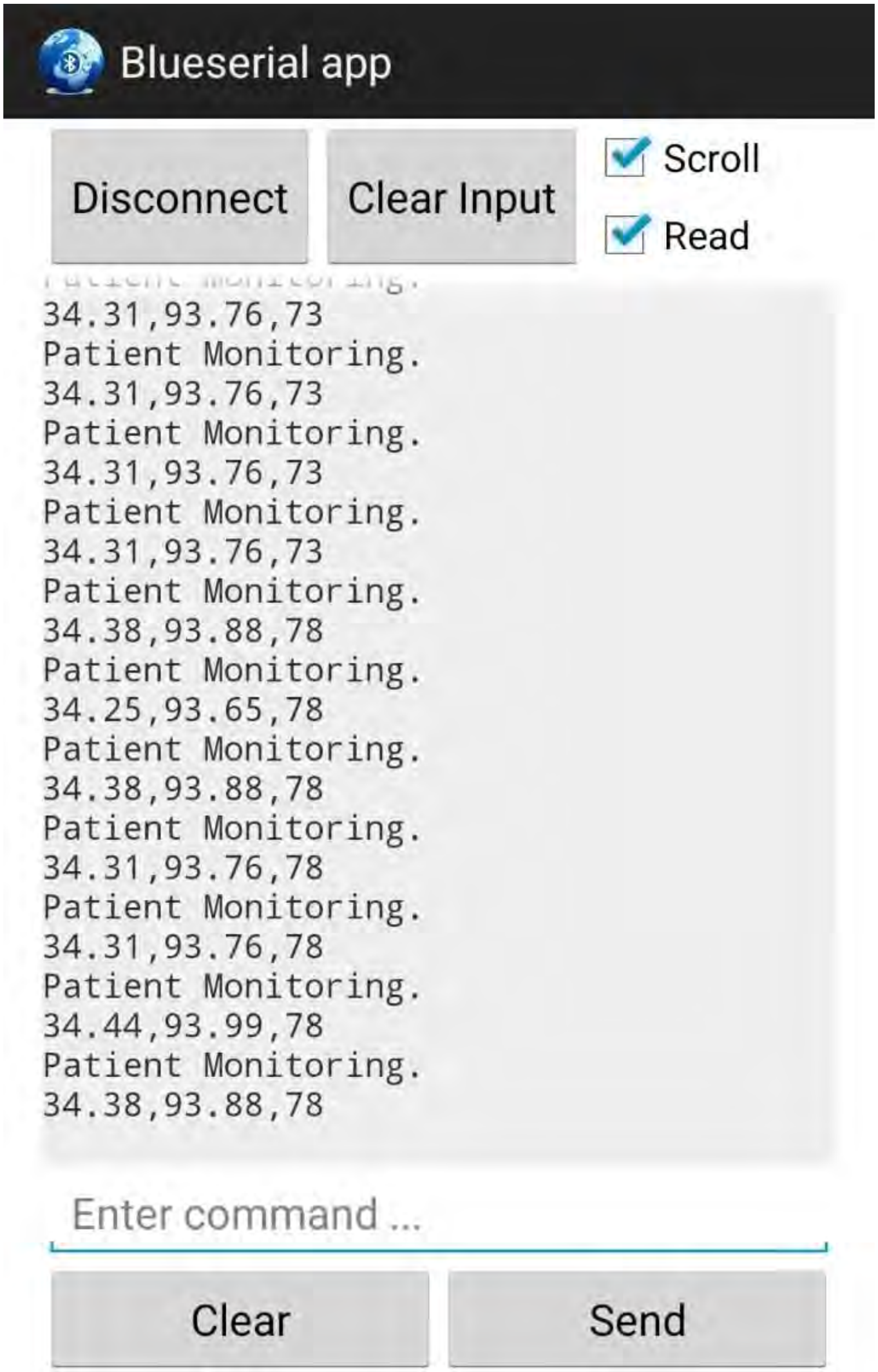

Patient Healthcare Monitoring System using IoT
DS18b20 Temperature Sensor
PulseSensor
Arduino UNO
HC-06 Bluetooth Module
ESP8266 Wifi Module

Figure 4.4: Final Setup



```
Patient Monitoring.
34.38,93.88,78
Patient Monitoring.
34.25,93.65,78
Patient Monitoring.
34.38,93.88,78
Patient Monitoring.
34.25,93.65,78
Patient Monitoring.
34.38,93.88,78
Patient Monitoring.
34.25,93.65,76
Patient Monitoring.
34.38,93.88,76
Patient Monitoring.
34.31,93.76,76
Patient Monitoring.
34.44,93.99,76
Patient Monitoring.
34.31,93.76,84
Patient Monitoring.
34.38,93.88,84
```

Figure 4.5(a): Results

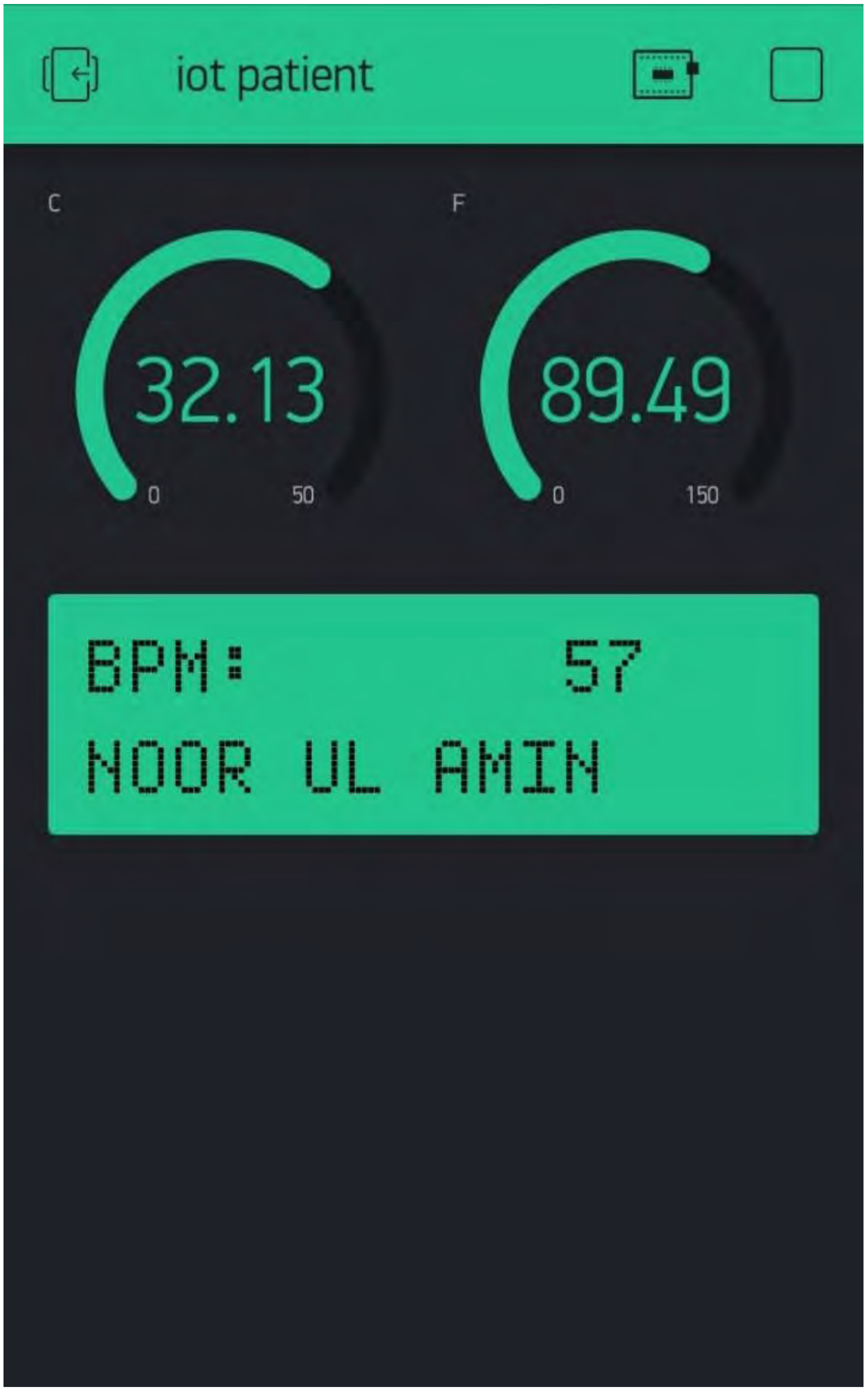


☑ Scroll

☑ Read

```
Patient Monitoring.
34.31,93.76,73
Patient Monitoring.
34.31,93.76,73
Patient Monitoring.
34.31,93.76,73
Patient Monitoring.
34.31,93.76,73
Patient Monitoring.
34.38,93.88,78
Patient Monitoring.
34.25,93.65,78
Patient Monitoring.
34.38,93.88,78
Patient Monitoring.
34.31,93.76,78
Patient Monitoring.
34.31,93.76,78
Patient Monitoring.
34.44,93.99,78
Patient Monitoring.
34.38,93.88,78
```

Figure 4.5(b): Results



```
BPM:          57
NOOR UL AMIN
```

Figure 4.5 (c): Results

4.5 Applications and limitations

Its applications include the data can be accessed and processed in a smart manner. The data can be taken either locally through android application using Bluetooth or very far away through Blynk application using Wifi module. Its limitations include that there may be moisture or externalities which may cause fluctuations in the readings of the sensors, also if the Bluetooth connection is lost the data is lost. The unavailability of Internet is also one of the limitations of the project system.

Chapter 5

Conclusion

5.1 Conclusion

In a nutshell, remote patient healthcare monitoring system is of great importance in regard to monitor patients smartly and efficiently, where both the patients and doctors can reach out each other without travelling up to long distances. The data can be easily fetched and processed from the patients in real time fashion.

5.2 Future work

It is recommended that, multiple sensors can be implemented and more than one or two services can be taken from this gadget at one time. The data can be stored in a database. Also this can further be furnished and taken to more advanced level using wireless sensors e.g. smart watch.

5.3 References

[1]https://www.generationrobots.com/media/DetecteurDePoulsAmplifie/PulseSensorAmpedGettingStartedGuide.pdf

[2] https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html

[3] https://www.irjet.net/archives/V5/i4/IRJET-V5I4375.pdf

[4] https://www.ripublication.com/ijaer18/ijaerv13n15_40.pdf

[5] http://www.sgbotic.com/products/datasheets/wireless/hc06_datasheet.pdf

[6] https://1sheeld.com/mqtt-protocol/

[7]https://www.geeksforgeeks.org/socket-programming-cc/

[8] https://developers.android.com/studio