Padmabhushan Vasantdada Patil Pratishthan's
College of Engineering

A MINI-PROJECT REPORT

ON

"SMART HEALTH MONITORING SYSTEM"

Submitted by:-

SNEHAL GHADGE **VU4F1718099**
RAFAT ANSARI **VU4F1718102**
JAGRUTI KHICHI **VU4F1718101**

Department of Information Technology

Padmabhushan Vasantdada Patil Pratishthan's

College of Engineering

Sion-Chunabhatti Eastern Express Highway

Mumbai - 400022

2019-2020

Padmabhushan Vasantdada Patil Pratishthan's

College of Engineering

Sion-Chunabhatti Eastern Express Highway

Mumbai - 400022

Department of Information Technology

CERTIFICATE

This is to certify that the mini-project work assigned by the University of Mumbai in the subject "Wireless Sensor Network" has been completed in a satisfactory manner within the premises of the Institute during the academic year 2019-2020 under the project titled **"SMART HEALTH MONITORING SYSTEM"** which is a bona fide work carried out by

SNEHAL GHADGE	**VU4F1718099**
RAFAT ANSARI	**VU4F1718102**
JAGRUTI KHICHI	**VU4F1718101**

....	....
Dr. Seema Ladhe	Dr. Alam N. Shaikh
Head of Department	**Principal**

....	....
Mrs. Danashri Konnur	**External Examiner**
Project Guide	

CONTENTS

ACKNOWLEDGEMENT

ABSTRACT

CHAPTERS:

1. INTRODUCTION...6

1.1 PROBLEM STATEMENT..6

1.2 NEED OF THE SYSTEM...6

1.3 LITERATURE SURVEY..7

2. SYSTEM STUDY..8

2.1 PROPOSED SYSTEM...8

2.2 ADVANTAGES OF THE PROPOSED SYSTEM.................................8

2.3 HARDWARE SPECIFICATIONS..9

2.4 SOFTWARE SPECIFICATIONS..17

3. SYSTEM IMPLEMENTATION...20

3.1 BLOCK DIAGRAM...20

3.2 CIRCUIT DIAGRAM...20

3.3 CODE...21

3.4 OUTPUT...25

4. CONCLUSION..28

5. REFERENCES...29

ACKNOWLEDGEMENT

We owe our gratitude to many people who have supported us throughout this journey. We would firstly like to express our heartfelt gratitude towards our respected Principal **Dr.Alam Shaikh** and our Head of Department **Dr. Seema Ladhe** for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our project guide **Mrs. Danashri Konnur** for her exemplary guidance, monitoring and constant encouragement throughout the course of the project.

Furthermore we would like to acknowledge with much appreciation the crucial role of the staff of SNL lab for giving the equipment and software for completion of the project "**Health Monitoring System**".

Last but not the least, we would like to thank our group members who contributed extensively throughout the semester and made this project possible. We would also like to extend our thank for the guidance given by other supervisor as well as the panels especially in our Wireless Sensor Network Project.

ABSTRACT

IOT is one of the booming field in forthcoming years and is starting to play a major role in every aspect of life. Present day's IoT brings the gadgets together and assumes a fundamental part in different methodologies like smart home mechanization, savvy urban areas, vehicle parking, traffic control, brilliant industries, smart environment, agribusiness fields, health monitoring systems and so on. IOT helps in connecting people by empowering their health and wealth in a smart way through smart gadgets. Recent improvements in wireless sensor networks have created a new trend in Internet of Things. In the health care centres, it is a challenge to monitor the health condition of the patients for 24 hours. Smart health has proved to be an important application for that matter. Patients with abnormal health conditions can be quickly monitored through smart health monitoring system. This type of system can be encountered through biomedical sensors that continuously monitor the activity and condition of the patients in a predictable method. The main aim of this work is to provide an extensive research in capturing the sensor data's, analyzing the data and alerting patient's caretaker based on different health parameters. In this way the system diminishes the exertion of specialists and paramedical staff to screen the patient for 24 hours and furthermore lessens the time and cost of support.

1. INTRODUCTION

1.1 PROBLEM STATEMENT

- In India, near about 20% of the total population loses their lives due to interrupted health monitoring i.e. in most of the hospitals, doctor visits patients either in morning shift or in evening shift or maybe, in both shifts.
- However, Patient's health can become critical in between that interval and such scenario can unfortunately put patient’s life at stake.
- Health care is also important at rural areas or villages where people need to travel to distant hospitals in order to get informed of their health condition.
- Hence, designing a Smart health monitoring system that can monitor the health parameters of the patients and also alerts the doctor/caretaker whenever critical heart rate or body temperature is observed is a necessity.

1.2 NEED OF THE SYSTEM

- India confronting an issue of absence of Doctors, from the most recent report of the Medical Council of India (MCI) we have 10.4 lakh specialists enlisted in Register of India Medical.
- In that 10.4 lakh, the 80% of specialists presently serves the patient at same time. It is 8.32 lakh specialists may be truly available for dynamic support to patient. In India specialists and patient proportion is around 1:1568 at odds with the World Health Organization standard of 1:1000.
- The 60,000 doctors and 28,000 post graduation doctors are moved on from different colleges once a year. There are around 11.65 lakh medical caretakers enlisted in Nursing Council of India. In that just 42% are in dynamic service.
- As per the suggests an attendant patient proportion of 1:1 in intensive care unit, 1:3 in the general care unit and 1:6 in the emergency ward. On the off chance 45 nurses are require for the every 15 patient, as 15 will work in every shift. You additionally require 30% leave save.
- This is the reason there is an immense deficiency of medical attendants. We require double the current number of nursing experts to adjust the diminishing patient-nurture proportion.
- In this way, it is clear there is just a single doctor for 2000 patients and requires twofold sum paramedical staff for present existing staff. It is impractical to rise to the patient and specialist proportion and multiplying the paramedical staff.
- Smart Health Monitoring System through IoT approach is utilized to decrease the endeavours of the doctors and paramedical staff. This strategy is likewise comfort for the patient since it lessens the enormous hardware, which presently utilized as a part of ICU's.

1.3 LITERATURE SURVEY

Designing of a continuous smart health monitoring system is the hot topic for researchers. A remote healthcare monitoring system has more advantages for those who are living in rural areas and not able to reach the hospital centres on time and from the other aspect, the strain on hospital medical resources like doctors, patients, and wards would also have decreased.

The continuous healthcare monitoring system is generally relying on wireless sensor network which decreases the rate of energy consumption and extend the coverage area for communication. Smart healthcare monitoring and giving more attention to people health is the difficult task that people must be aware of.

The development of sensors has brought huge facilities to the hospital environment. Sensor is used for the evaluation of different signs like ECG, motion, temperature, blood pressure and heart beating. Moreover, the different technologies and technological standards used for data access and storage, visualization and healthcare analysis techniques are essential parts of a continuous healthcare structure.

The development of electronic healthcare monitoring platform has changed the traditional way of healthcare method. Compromised IOT into these systems have increased adaptability, intelligence, and interoperability. The modern smart phones have the ability to generate continuous monitoring services by customizing the related applications.

Furthermore, in the literature review some of the proposed IOT architectures for healthcare monitoring are also discussed:

“IOT based smart healthcare kit”: A model designed for immediate medical situations to provide assistance to medical services. INTEL GALILEO second generation board was used which could collect data from the sensors touched to patients and were able to send the data over the internet for further temperature clarification and LM35 sensors were used for recording patient health.

“Implementation of a healthcare monitoring system using Raspberry pi”: The primary purpose of this model was to track temperature of patient body and the heartbeat of sick one at runtime. Physical parameters are concentrated and many users can access it. B+ model of Raspberry pi is used, through which healthcare parameters are focused and supervised.

“Real-time wireless health monitoring”: A system developed which has the below different components:

A. ECG Electrodes, a device which is attached to patient body like chest, arms, legs at the time of testing stage that track the electrical impulses when the heart beats.

B. LM35, which is used for sensing of body temperature.

C. Blood pressure sensor, which measures the patient blood pressure.

“Secured Smart Healthcare Monitoring System based on IOT”: In this system for collecting sensing data, Microcontroller PIC16F877A is used and delivered the data through the internet of things. The data is saved and doctors are able to access the recorded data from any place, anytime by using smart devices and track the health conditions of the relevant patients.

“Patient Health Monitoring System (PHM) Using IoT devices”: This system is used to collect the health data and values from patients. Various sensors are there for collecting diverse parameters. Sensors are used for blood pressure, heartbeats and body temperature of a patient and Wi-Fi or Bluetooth is the transmission device. After every 60 seconds record of data is updated.

2. SYSTEM STUDY

2.1 PROPOSED SYSTEM

Monitoring patient health whenever doctors/caretaker is not in the vicinity has always been a serious issue. With the number of patients rising each day and the lack of medical professionals in the region, it is laborious to analyze patient health status all the time. Thus, we need to find an IOT solution to this problematic situation.

One of the approaches is to monitor the health state of the patient and screen it to doctors or paramedical staff through the IoT, as it is hard to screen the patient for 24 hours. So here the patient health parameters such as Pulse rate, Respiratory rate, Body Temperature, Position of the body, ECG and so on can be measured by utilizing the Non-invasive sensors.

To keep a constant track of patient's health condition, a variety of sensors can be used. This work presents a smart health monitoring system that uses biomedical sensors to check patient's health condition and uses internet to inform the concerned. To keep this project simple, the biomedical sensors used in the proposed system are pulse sensor and temperature sensor. Pulse rate sensor can measure the heart rate of a person per minute which is typically around 70 beats per minute in normal state. Furthermore, the LM35 temperature sensor will measure body temperature.

These biomedical sensors are connected to Arduino UNO controller to read the data which is in turn interfaced to an LCD display/serial monitor to see the output. The "Thingspeak" named cloud is utilized here to store the detected data into the server. From this server, data is converted into JSON link in order to send an SMS alert through IFTTT applet services whenever the system observes abnormal heart rate or abnormal human body temperature to the concerned doctor/caretaker.

The proposed IOT system can be used by the health care centres to monitor physiological statuses of patients through sensors by collecting and analyzing their information and then sending the analyzed patient's data remotely to the concerned doctor/paramedical staff whenever the system detects a critical reading.

2.2 ADVANTAGES OF THE PROPOSED SYSTEM

- The patients can monitor their health conditions at any time from any location and need not visit the hospitals all the time.
- So the status can be determined from any place of the world online and the doctor can examine the patient health conditions all the time by using different smart devices such as tablets, smart phones, and laptops.
- The family members can also track patient's health in an emergency situation.
- The illness people are not required to stand in a long queue for doctors, thus save more time and cost.
- Through this system distance barrier is eliminated.

2.3 HARDWARE SPECIFICATIONS

The components used in design and connections of the project are as follows:

1. ARDUINO UNO
2. LM35 TEMPERATURE SENSOR
3. PULSE SENSOR
4. ESP8266 WI-FI MODULE
5. LIQUID CRYSTAL DISPLAY
6. BREADBOARD
7. RESISTORS
8. JUMPER WIRES

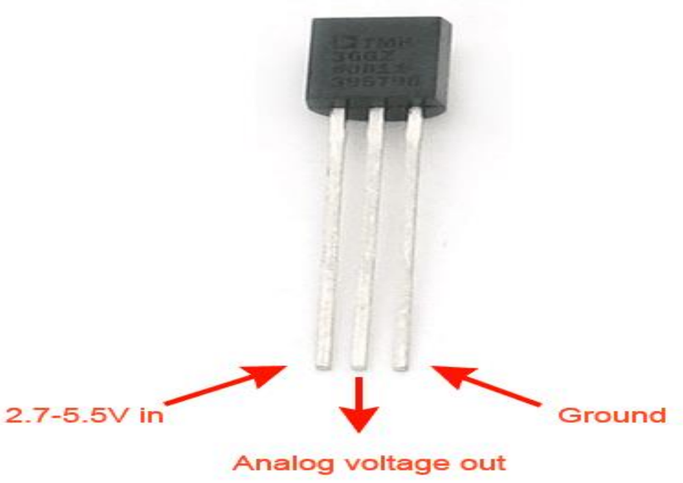
1. ARDUINO UNO

Arduino Uno is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328. First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world. The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output. It allows the designers to control and sense the external electronic devices in the real world. Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board.

Features of the Arduino Uno Board:

- It is an easy USB interface. This allows interface with USB as this is like a serial device.
- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
- It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
- It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes it easy to help in debugging projects.

2. LM35 TEMPERATURE SENSOR



LM35 is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius. The advantage of lm35 over thermistor is it does not require any external calibration. The coating also protects it from self-heating. Low cost (approximately $0.95) and greater accuracy make it popular among hobbyists, DIY circuit makers, and students. Many low-end products take advantage of low cost, greater accuracy and used LM35 in their products. Its approximately 15+ years to its first release but the sensor is still surviving and is used in any products. LM35 can measure from -55 degrees centigrade to 150-degree centigrade. The accuracy level is very high if operated at optimal temperature and humidity levels. The conversion of the output voltage to centigrade is also easy and straight forward. The input voltage to LM35 can be from +4 volts to 30 volts. It consumes about 60 microamperes of current. Lm35 has many family members a few names are LM35C, LM35CA, LM35D, LM135, LM135A, LM235, LM335. All LM35 family members work on the same principles but temperature measuring capacity varies and also they are available in many packages (SOIC, TO-220, TO-92, TO).

Features of LM35:

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical

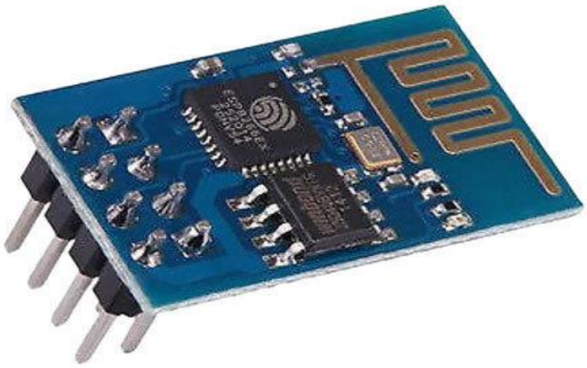
3. PULSE SENSOR

Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart rate data into their projects. The sensor clips onto a fingertip or earlobe and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time. An alternate name of this sensor is heartbeat sensor or heart rate sensor. The working of this sensor can be done by connecting it from the fingertip or human ear to Arduino board. So that heart rate can be easily calculated. The pulse sensor working principle is very simple. This sensor has two surfaces, on the first surface, the light-emitting diode & ambient light sensor is connected. Similarly, on the second surface, the circuit is connected which is accountable for the noise cancellation& amplification. The LED is located above a vein in a human body like ear tip or fingertip, however, it must be located on top of a layer directly. Once the LED is located on the vein, then the LED starts emitting light. Once the heart is pumping, then there will be a flow of blood within the veins. So if we check the blood flow, then we can check the heart rates also. If the blood flow is sensed then the ambient light sensor will receive more light as they will be reproduced by the flow of blood. This small change within obtained light can be examined over time to decide our pulse rates.

Features of Pulse sensor:

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V
- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit.
- Diameter: 0.625”

4. ESP8266 WI-FI MODULE

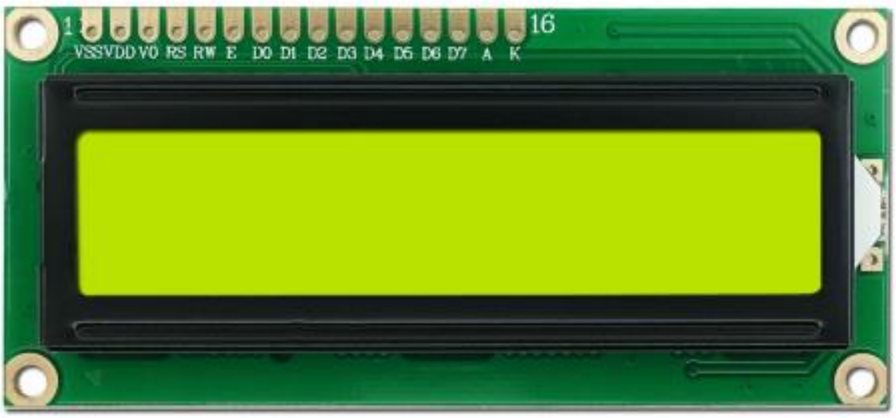


The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Features of ESP8266 Wi-Fi Module:

- Low cost, compact and powerful Wi-Fi Module
- Power Supply: +3.3V only
- Current Consumption: 100mA
- I/O Voltage: 3.6V (max)
- I/O source current: 12mA (max)
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports serial communication hence compatible with many development platform
- Can be programmed using Arduino IDE or AT-commands or Lua Scrip

5. LIQUID CRYSTAL DISPLAY



LCD (Liquid Crystal Display) screen is an electronic display module and can be found in 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 and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations 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. A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

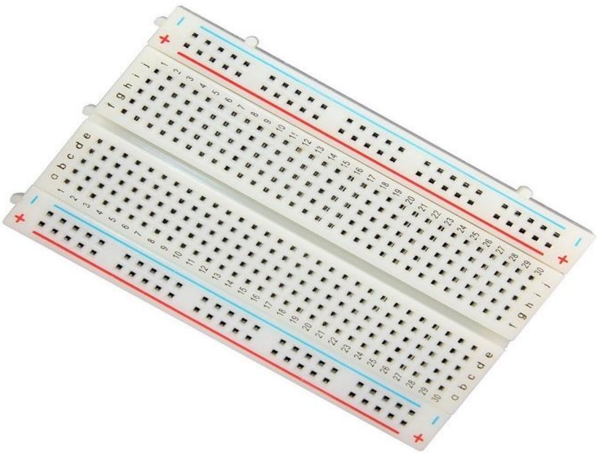
Command Register: The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task. Examples like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happens in the command register.

Data Register: 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. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

Features of LCD Display:

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.
- Each character is build by a 5×8 pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters
- Available in Green and Blue Backlight

6. BREADBOARD



Breadboard is a solder-less device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and holes on the top of the board. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically. The breadboard has many holes into which circuit components like ICs and resistors can be inserted. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

7. RESISTORS

A Resistor is a passive two-terminal electrical components that implements electrical resistance as a circuit elements. Resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. It is used to restrict the current flow in a circuit. The resistance toward the flow of current will result in the voltage drop. These devices may provide a permanent, adjustable resistance value. The value of resistors can be expressed in Ohms

8. JUMPER WIRES

A **jumper wire** (also known as jump wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. **Male-to-male jumper wires** are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you'll need. The difference between each is in the end point of the **wire**. **Male** ends have a pin protruding and can plug into things, while **female** ends do not and are used to plug things into.

Male-to-Male Jumper Wires

Male-Female Jumper Wires

2.4 SOFTWARE SPECIFICATIONS

The software tools utilized during the course of the project are:

1. ARDUINO IDE
2. THINGSPEAK
3. IFTTT

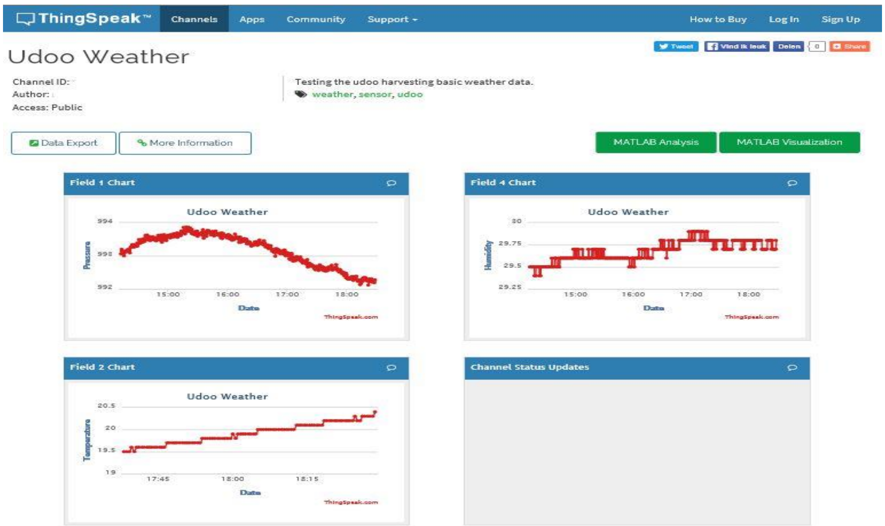
1. ARDUINO SOFTWARE (IDE)

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. This IDE comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

It is used to write and upload programs to Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE Extension.

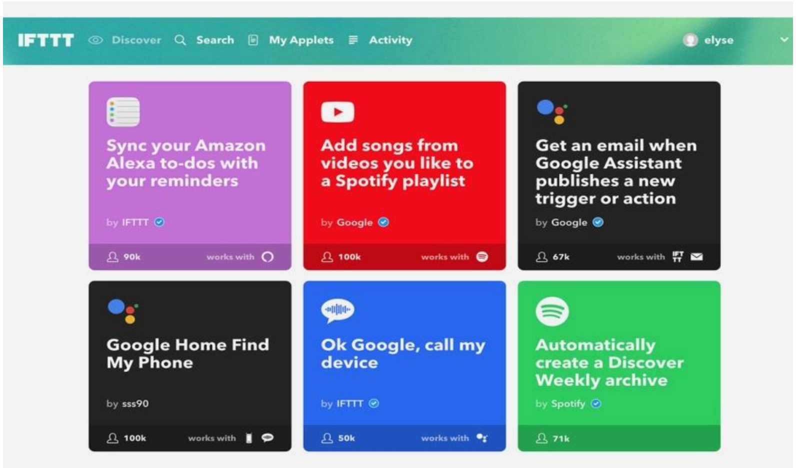
The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

2. THINGSPEAK



ThingSpeak is IoT Cloud platform where you can send sensor data to the cloud. You can also analyze and visualize your data with MATLAB or other software, including making your own applications. The ThingSpeak service is operated by MathWorks. In order to sign up for ThingSpeak, you must create a new MathWorks Account or log in to your existing MathWorks Account. ThingSpeak is free for small non-commercial projects. ThingSpeak includes a Web Service (REST API) that lets you collect and store sensor data in the cloud and develop Internet of Things applications. It works with Arduino, Raspberry Pi and MATLAB (premade libraries and APIs exists) But it should work with all kind of Programming Languages, since it uses a REST API and HTTP.

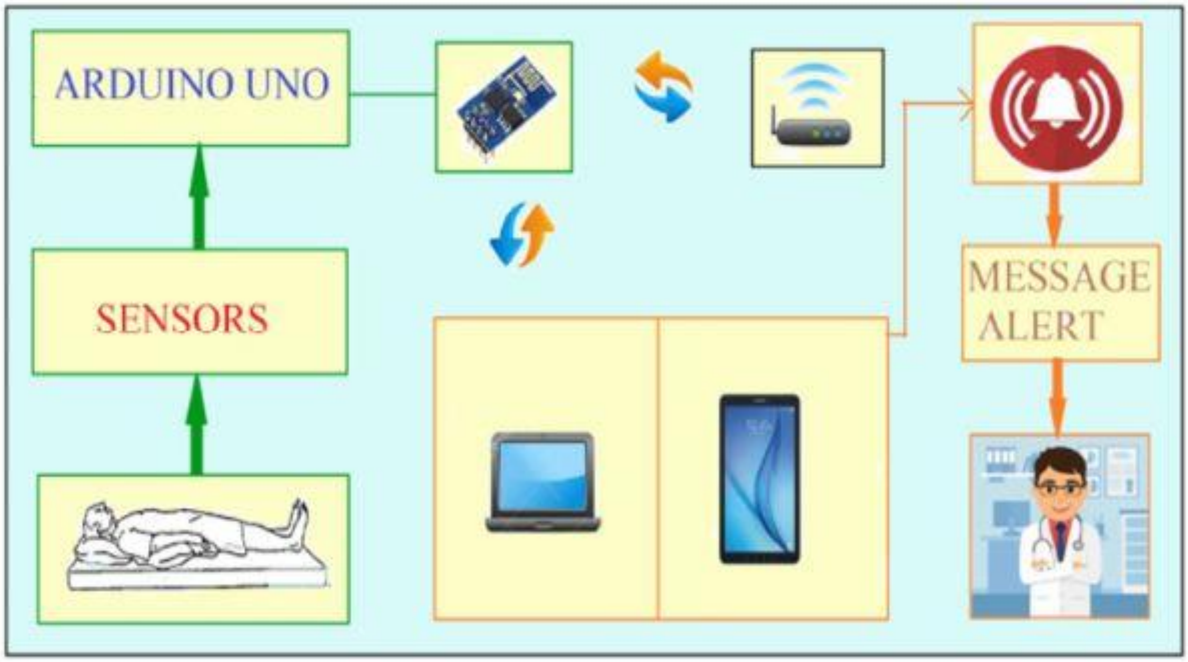
3. IFTTT



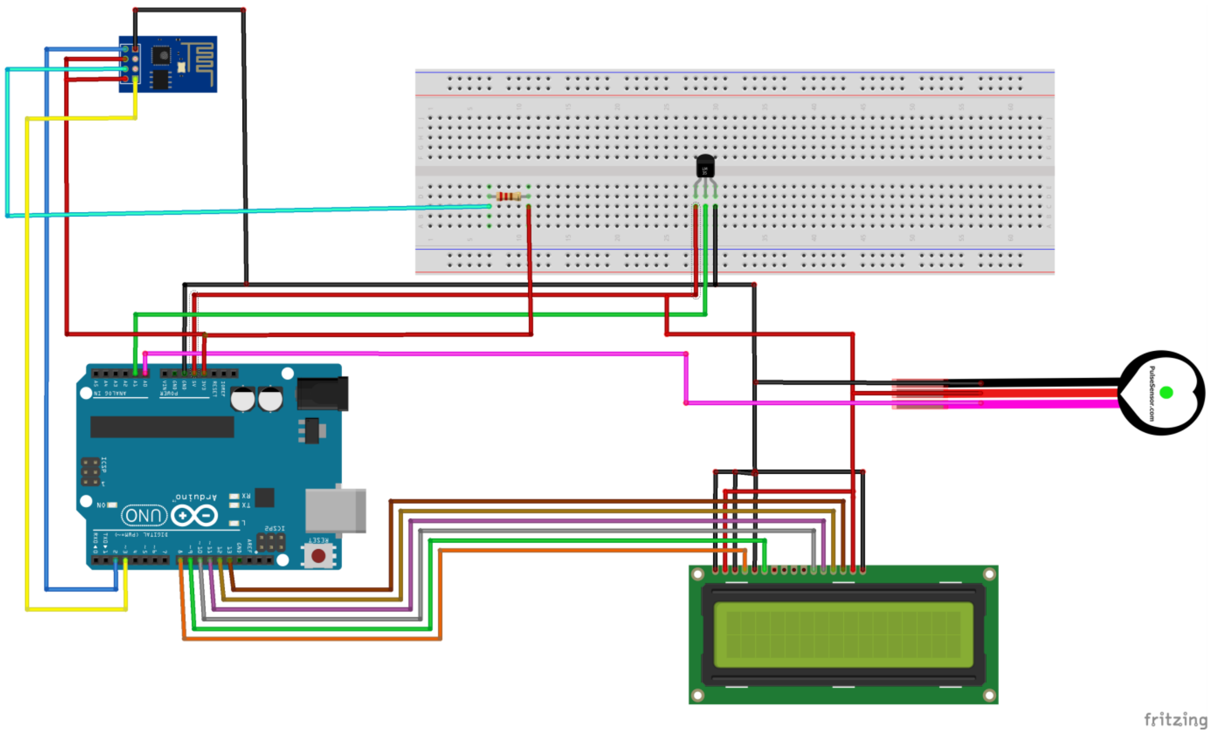
This is the web based service provider which can be used for server to server communication. It helps to create simple conditional statement which is called as applets. There are many applets in this IFTTT server. These applets pop up when there is a trigger in that connected platform like cloud or other social media’s and other channels, we can connect the thing speak cloud with this IFTTT for trigger. Services such as Webhooks and Android SMS can be used to trigger an SMS whenever a certain condition is met.

3. SYSTEM IMPLEMENTATION

3.1 BLOCK DIAGRAM



3.2 CIRCUIT DIAGRAM



3.3 CODE

```
#define USE_ARDUINO_INTERRUPTS true
#include <SoftwareSerial.h>
#include<LiquidCrystal.h>
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

String APIKEY = "PZSIOY00O197S6XY" ;
String SSID = "samsungDUOS"  ;   // "SSID-WiFiname"
String PASS =  "ansari.rkzrksm6.."; // "password"
String IP =  "184.106.153.149"    ;  // thingspeak.com IP Address

// Variables
const int PulseWire = A0;
const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.
int Threshold = 550;
String BPM;
String temp;

// Instances
SoftwareSerial esp8266(2,3);      // arduino RX pin=2  arduino TX pin=3
PulseSensorPlayground pulseSensor;
LiquidCrystal lcd(8,9,10,11,12,13);

byte degree[8] =
{
0b00011,
0b00011,
0b00000,
0b00000,
0b00000,
0b00000,
0b00000,
0b00000
};

//===============Sending BPM to ThingSpeak====================//

void thingSpeakWrite(){
  String cmd = "AT+CIPSTART=\"TCP\",\"";                  // TCP connection
  cmd += IP;                              // api.thingspeak.com
  cmd += "\",80";
  Serial.println(cmd);
  esp8266.println(cmd);

  if(esp8266.find("Error")){
    Serial.println("AT+CIPSTART error");
    return false;
  }

  String getStr = "GET /update?api_key=";   // prepare GET string
  getStr += APIKEY;
  getStr +="&field1=";
  getStr += BPM;
```

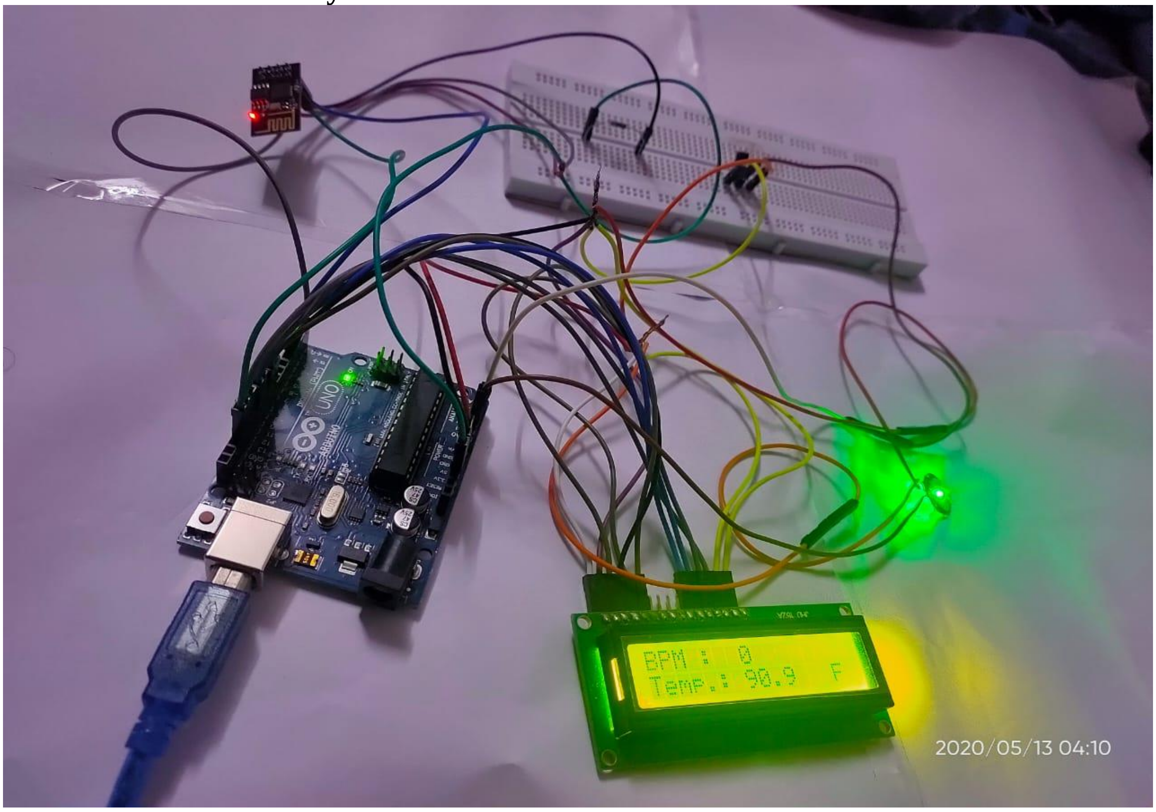
```
  getStr +="&field2=";
  getStr += temp;
  getStr += "\r\n\r\n";

  // send data length
  cmd = "AT+CIPSEND=";
  cmd += String(getStr.length());
  esp8266.println(cmd);
  Serial.println(cmd);
  delay(80);

  if(esp8266.find(">")){
    esp8266.print(getStr);
    Serial.print(getStr);
  }
  else{
    esp8266.println("AT+CIPCLOSE");
    Serial.println("AT+CIPCLOSE");
    return false;
  }
  return true;
}

//===========================Doing the Setup===============================//

void setup() {
  Serial.begin(9600); // enable software serial
  esp8266.begin(115200);
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.
  pulseSensor.setThreshold(Threshold);

  lcd.begin(16,2);
  lcd.createChar(1, degree);
  lcd.setCursor(0,0);
  lcd.print(" Health Monitor");
  lcd.setCursor(0,1);
  lcd.print("    System    ");
  delay(2000);

  if (pulseSensor.begin()) {
    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,
or on Arduino reset.
  }

//NOTE: For running the code first time uncomment the following lines//

/*  esp8266.println("AT+RST");         // Enable this line to reset the module;
  delay(2000);
*/

  if(esp8266.available()){
    connectWifi();
  }
  else{
```

```
    Serial.println("Wi-Fi Connection Error!");
  }
}

//=========================Connecting to Wi-Fi================================//

boolean connectWifi(){
  Serial.println("AT");
  esp8266.println("AT");
  delay(4000);

  Serial.println("AT+CWMODE=1");
  esp8266.println("AT+CWMODE=1");   // set esp8266 as client
  delay(2000);
    //showResponse(1000);

  Serial.println("AT+CWJAP=\""+SSID+"\",\""+PASS+"\"");
  esp8266.println("AT+CWJAP=\""+SSID+"\",\""+PASS+"\"");   // set your home router SSID and
password
  delay(5000);
    //showResponse(5000);
  if(esp8266.find("OK")){
    Serial.println("Setup completed");
    return true;
  }
  else{
    return false;
  }
}

//======================Looping through the readings========================//

void loop() {
   float reading=analogRead(A1);
   float tempC=reading*(5.0/1023.0)*100;
   float tempF = (tempC * 9.0 / 5.0) + 32.0;
   char buffer1[10];
   char buffer2[10];

   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("A beat happened");
    }
   delay(20);
   BPM = dtostrf(myBPM, 4, 1, buffer1);
   temp = dtostrf(tempF, 4, 1, buffer2);// Print the value inside of myBPM.

    if (isnan(tempF)) {
      Serial.println("Failed to read from Temperature sensor");
      }
    else{
      Serial.println("BPM="+String(BPM)+" BPM");
      Serial.println("Temp="+String(temp)+" *F");
```

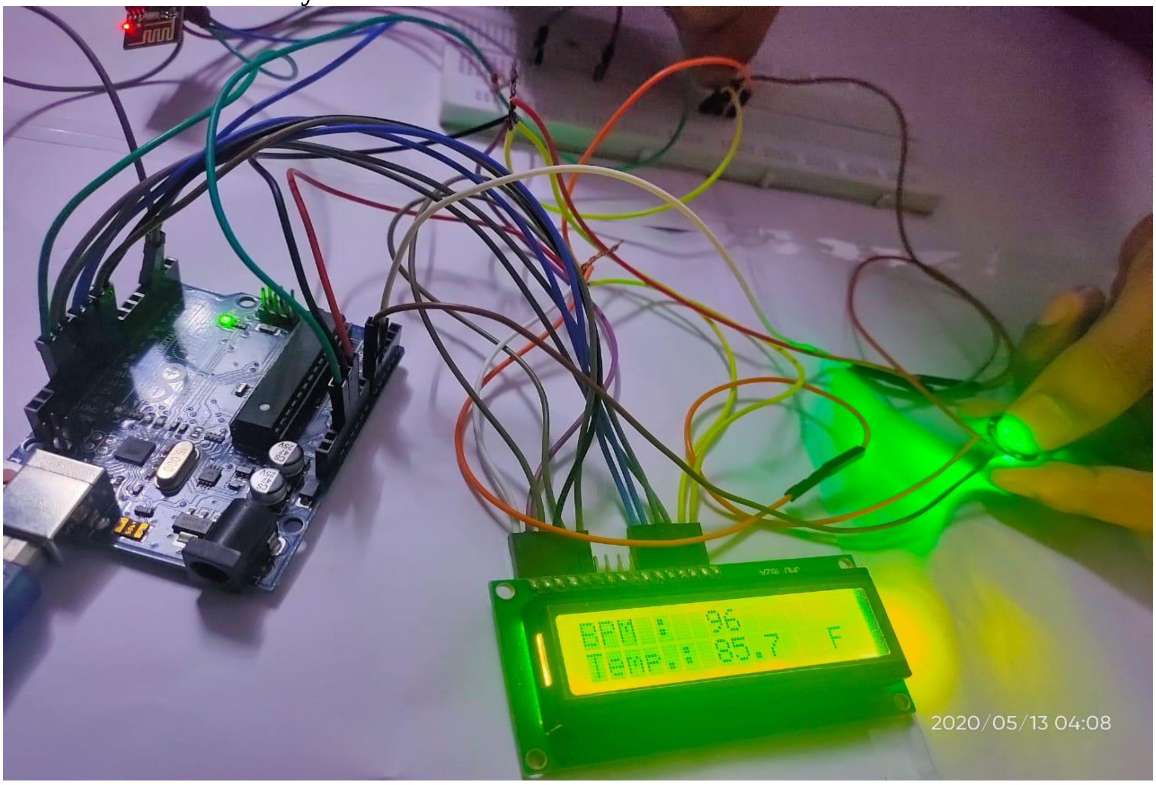
```
      thingSpeakWrite();
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("BPM :");
      lcd.setCursor(7,0);
      lcd.print(myBPM);
      lcd.setCursor(0,1);
      lcd.print("Temp.:");
      lcd.setCursor(7,1);
      lcd.print(temp);
      lcd.setCursor(13,1);
      lcd.print(" F");
    }
  //thingspeak needs 15 sec delay between updates,
  delay(15000);
}
```

3.4 OUTPUT

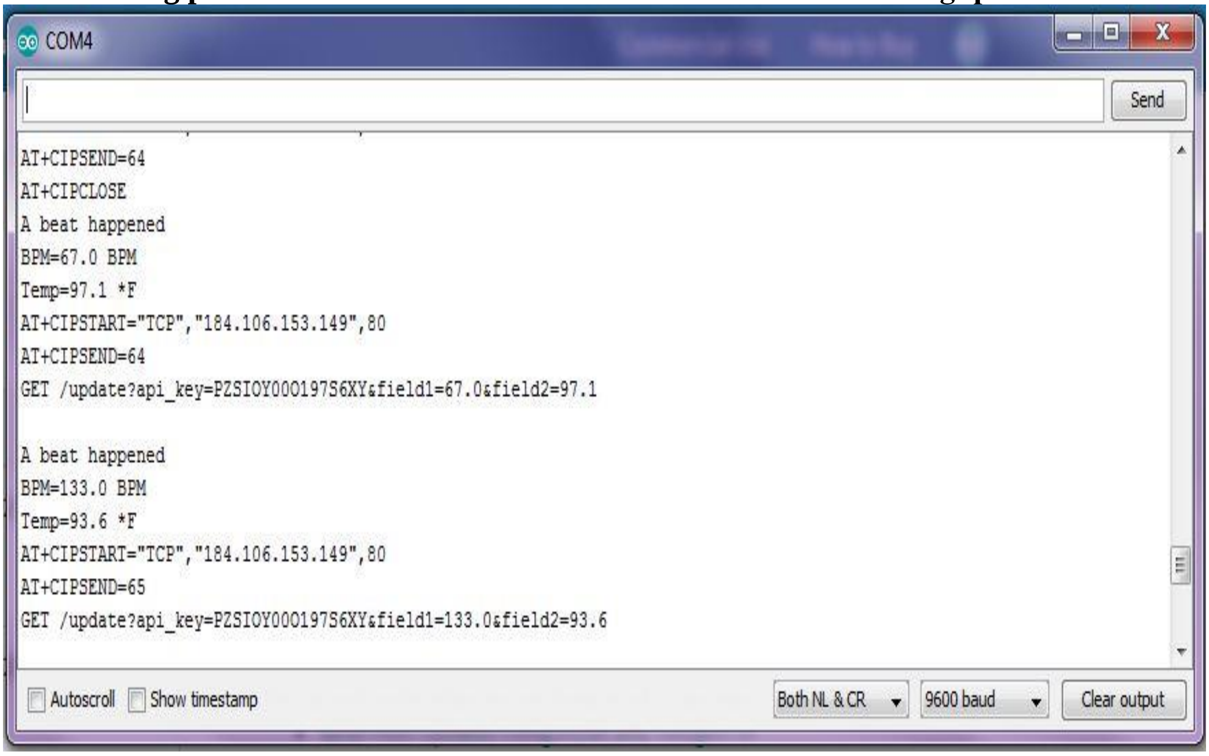
1. When no human body is in contact with the sensors.



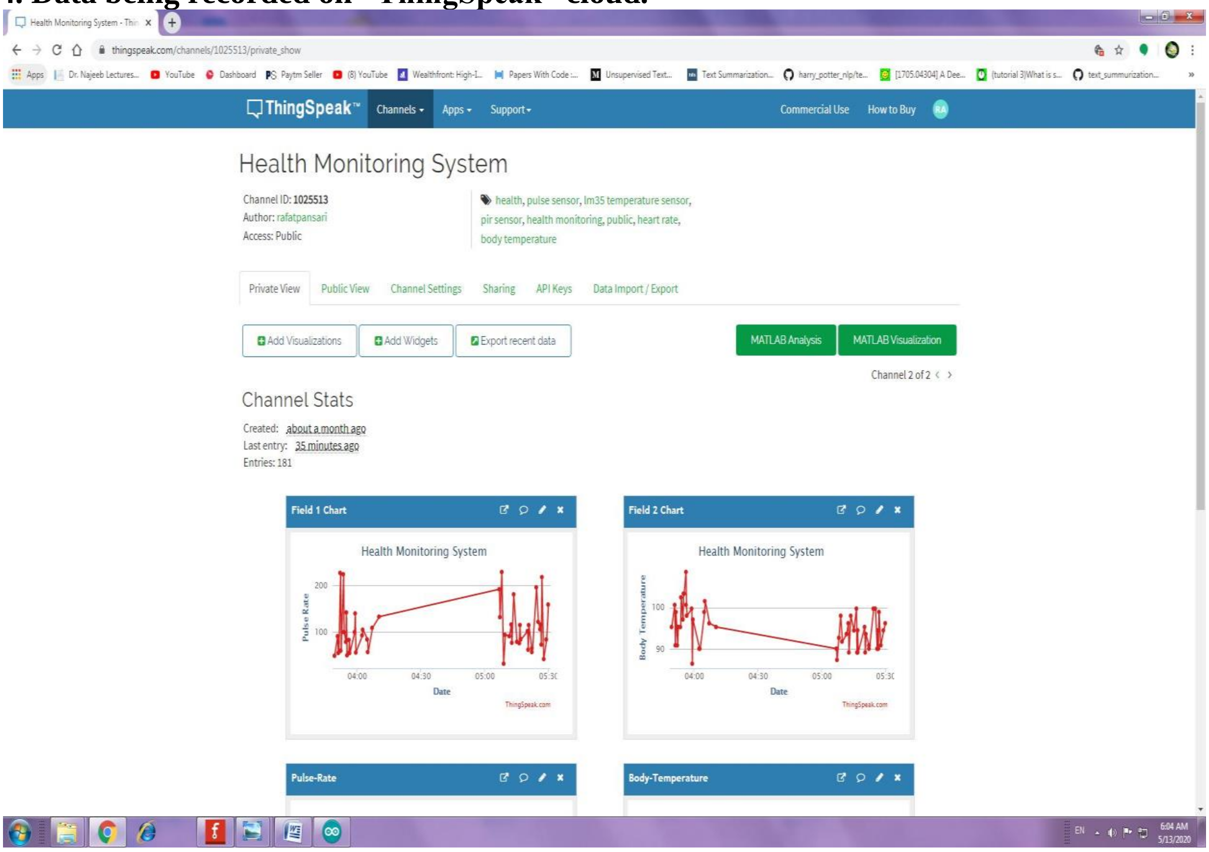
2. When a human body comes in contact with the sensors.



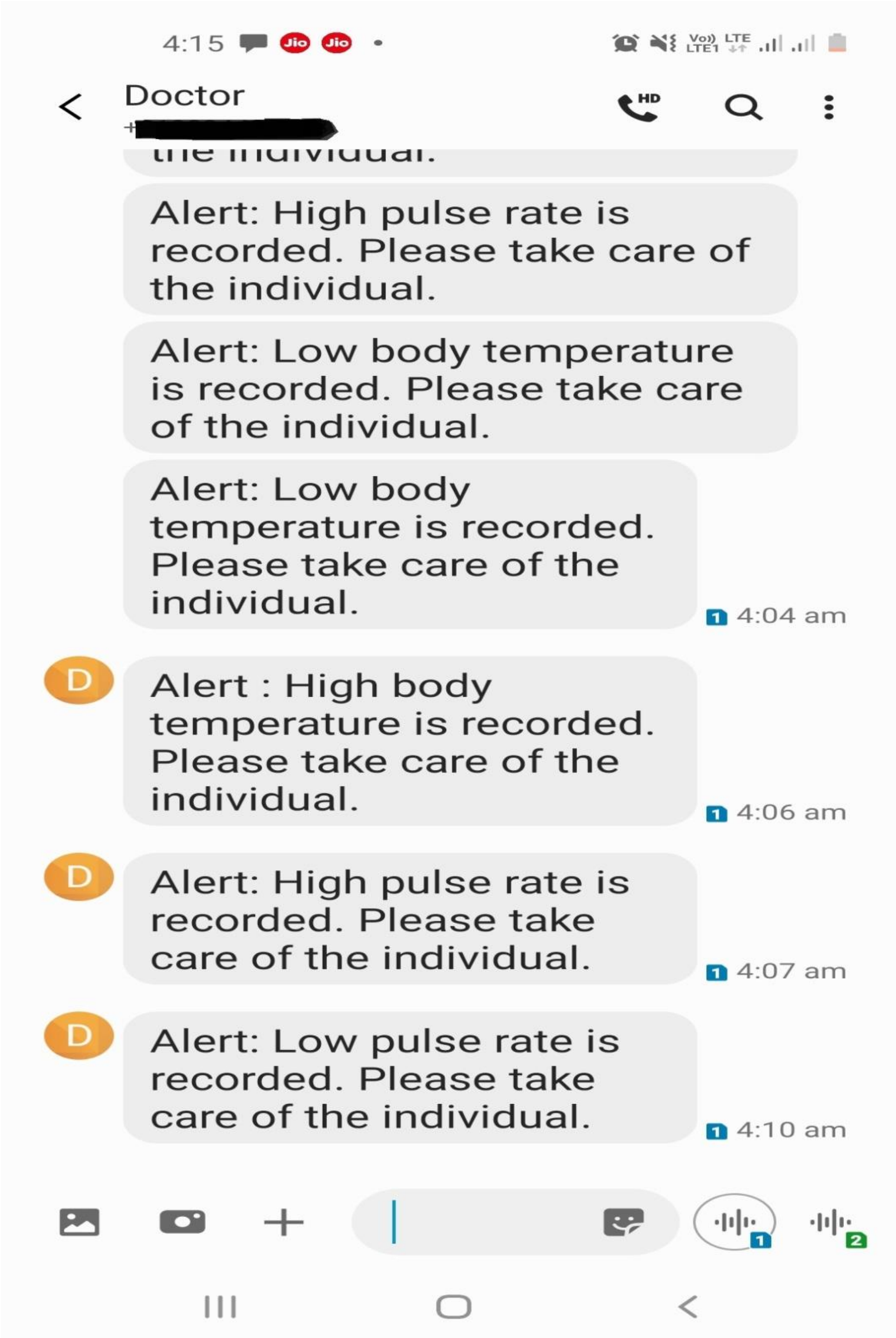
3. Data being printed out on serial monitor and sent further to “ThingSpeak” server.



4. Data being recorded on “ThingSpeak” cloud.



5. An SMS being received on the doctor/caretaker phone if any abnormal health parameter is recorded.



4. CONCLUSION

In this paper, we have proposed and implemented a Smart Health Monitoring System. It is working successfully. Through this system, Doctors can screen the patient's condition without the need of being in patient’s vicinity all the time. By using biomedical sensors, we have monitored patient’s data viz. temperature and heart beat rate. This data is further uploaded on the ThingSpeak server. From the server, the data is converted into JSON and any abrupt changes occurring in patient's health status is notified to the concerned Doctor or Paramedical staff through an SMS which is automatically sent by the system. For future work, we can increase the functionality of system by adding more biomedical sensors and by making our system readings more accurate.

5. REFERENCES

[1] Mohammad Dawood Babakerkhell, Nitin Pandey, "Analysis of Different IOT Based Healthcare Monitoring Systems", IJITEE, April 2019.

[2] Tarannum Khan, Manju K. Chattopadhyay, "SMART HEALTH MONITORING SYSTEM", IEEE, 2017.

[3] Ananth S, Sathya P and Madhan Mohan P, "Smart Health Monitoring System through IOT", International Conference on Communication and Signal Processing, April 4-6, 2019.

[4] S Pradeep Kumar, Vemuri Richard Ranjan Samson, U Bharath Sai, P L S D Malleswara Rao, K Kedar Eswar, "Smart Health Monitoring System of Patient Through IoT", International conference on I-SMAC, 2017.

[5] Prof(Dr).Jayant Shekhar, Mr.Desalegn Abebaw , Dr. Mesfin Abebe Haile, Md.Ahmed Mehamed, Mr.Yohannis Kifle, "Temperature and Heart Attack Detection using IOT(Arduino and ThingSpeak) ", International Journal of Advances in Computer Science and Technology, November 2018.