**HEALTH MONITORING SYSTEM**

A PROJECT REPORT

Submitted By

**SIIDHARTH S (17BEC1042)**

**KISHORE NITHIN S (17BEC1068)**

**V SURYA (17BEC1225)**

A project report submitted to

**PROF. RAJESH KUMAR**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

in partial fulfilment of the requirements for the course of

**CSE 3009: INTERNET OF THINGS**

in

**B.Tech.**

Electronics and Communication Engineering

and

Computer Science Engineering



Vandalur - Kodambakkam Road, Chennai - 600 127

**WINTER SEMESTER 2019-2020**

**31st MAY 2020**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**HEALTH MONITORING SYSTEM”** is a bonafide work of **SIDDHARTH S (17BEC1042), KISHORE NITHIN S (17BEC1068) & V SURYA (17BEC1225)** who carried out the Project work under my supervision and guidance.

**PROF. RAJESH KUMAR**

VIT University, Chennai

Chennai – 600 127.



Vandalur - Kodambakkam Road, Chennai - 600 127

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SERIAL NO.** | **NAME** | **PAGE NO.** |
| 1 | BONFIDE CERTIFICATE | 2 |
| 2 | ABSTRACT | 4 |
| 3 | ACKNOWLEDGEMENT | 5 |
| 4 | INTRODUCTION | 6 |
| 5 | PROPOSED SYSTEM | 7 |
| 6 | HARDWARE SYSTEM | 11 |
| 7 | SOFTWARE SIMULATION | 12 |
| 8 | ANALYTICS IN CLOUD PLATFORM | 13 |
| 9 | MOBILE APPLICATION | 14 |
| 10 | WEB APPLICATION | 15 |
| 11 | COMPONENTS USED | 15 |
| 12 | CODE | 19 |
| 13 | CONCLUSION | 24 |
| 14 | REFERENCES | 25 |

**ABSTRACT**

Health monitoring is the major problem in today’s world. Due to lack of proper health monitoring, patient suffer from serious health issues with extreme health issues leading to even death. India is a land the second most populous country in the world and is thus no surprise that it houses many people with different health problems throughout the country. This calls for a cheap and affordable health monitoring system which can be used to send data like patients’ blood pressure, pulse rate, oxygen levels, temperature etc. directly to the doctor to check for any abnormalities without having to frequent hospitals. Currently, Internet of Things (IoT) is at its zenith and is now capable of connecting billions of devices and services at any time in any place, with various applications. With emerging high-speed networks, the ease of use of IoT is going improve significantly. This technology can be integrated with a health monitoring system and thus can provide real-time monitoring of patient from anywhere in the world by a doctor. There are lots of IoT devices now days to monitor the health of patient over internet. Health experts are also taking advantage of these smart devices to keep an eye on their patients. With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry.

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **PROF. RAJESH KUMAR,** Associate Professor Senior, School of Computer Science and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Jagadeesh Kannan. R,** Dean of School of Computer Science and Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Head of the Department **Dr. Justus. S** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**INTRODUCTION**

The modern world is competitive with everyone giving their best to reach the pinnacle of success. This often comes with a negligence on their health and prolonging their visits to hospital even when their health is in bad shape. A health monitoring system gives way for busy people and anyone to send data like body temperature, pulse rate, oxygen levels etc. to their doctors without having to visit hospitals. In today’s era, health problems are increasing day-by-day at a high pace. The death rate of 55.3 million people dying each year or 151,600 people dying each day or 6316 people dying each hour is a big issue for all over the world. Adding on to this, there are numerous death occurrences in ambulances on their way to hospitals due to improper nursing. The deaths in ambulances are often accused with a case of improper treatment. Hence it is the need of hour to overcome such problems. We, therefore, proposing to integrate a health monitoring system using a microcontroller board and wireless sensors technology by designing a system which included different wireless sensors to receive information with respective human body temperature, blood oxygen level, heart rate etc. that will be undoubtedly further transmitted on an IoT platform which is accessible by the user via internet and doctors anywhere in the world. An accessible database is created about patient’s health history which can be further monitored & analyzed by the doctor if necessary. This data storage can be saved on the server permanently or can be reset via the software.

**PROPOSED SYSTEM**

In this project, we made an **IoT based Health Monitoring System** which records the patient heart beat rate and body temperature and also send an email/SMS alert whenever those readings go beyond critical values using Arduino UNO as the developer board. Pulse rate and body temperature readings are recorded over ThingSpeak and Google sheets so that patient health can be monitored from anywhere in the world over internet. A panic will also be attached so that patient can press it on emergency to send Email/SMS to their relatives.

**COMPONENTS REQUIRED:**

* Arduino Uno and Programming Cable
* ESP8266 Wi-Fi module
* LM35 temperature sensor
* MAX30100 sensor
* Push button
* 10k Resistor
* Male-female wires
* Breadboard

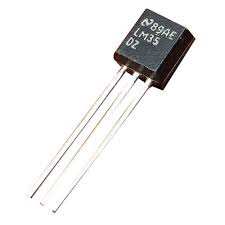
ARDUINO UNO



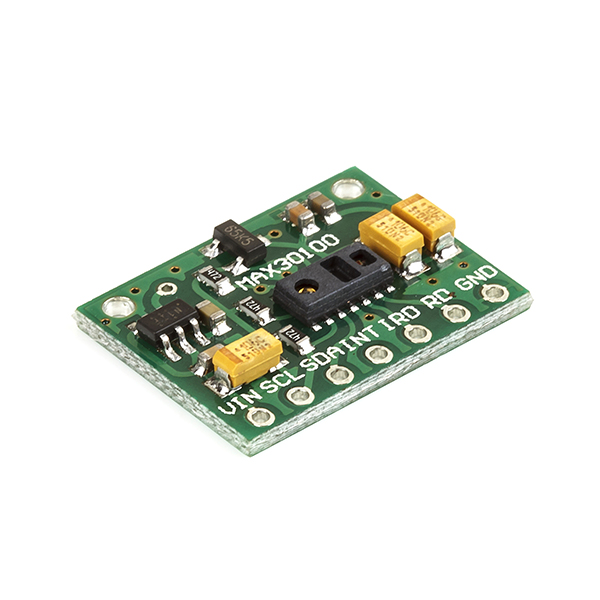
ESP8266 Wi-Fi MODULE



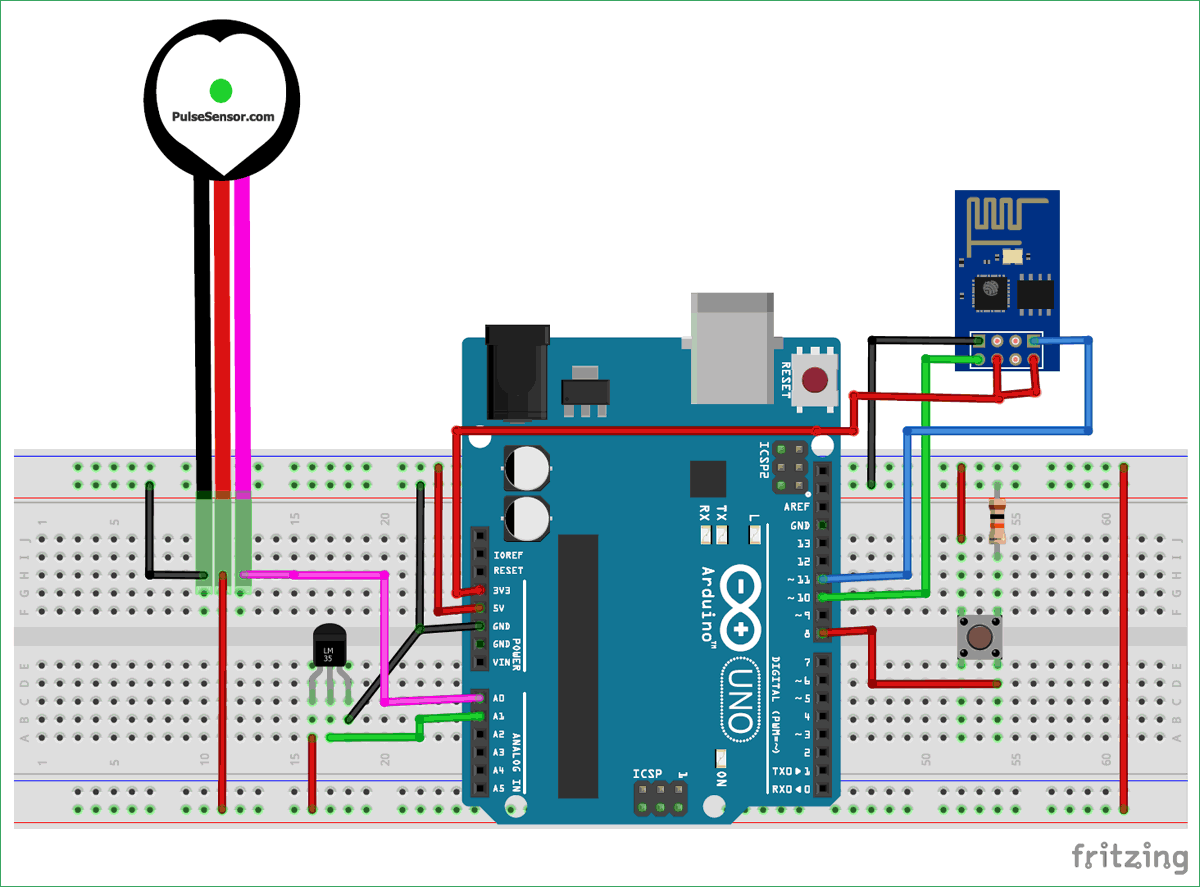
LM35 TEMPERATURE SENSOR



MAX30100 SENSOR



**SCHEMATIC:**



Connections are as follows:

* Signal pin of pulse sensor -> A0 of Arduino
* Vcc pin of pulse sensor -> 5V of Arduino
* GND pin of pulse sensor -> GND of Arduino
* Vout of LM35 -> A1 of Arduino
* Tx of ESP8266 -> Pin 10 of Arduino
* Rx of ESP8266 -> Pin 11 of Arduino
* CH\_PD and Vcc of ESP8266 -> 3.3 V of Arduino
* GND of ESP8266 -> GND of Arduino
* Push button -> Digital Pin 8 of Arduino

**HARDWARE SYSTEM**

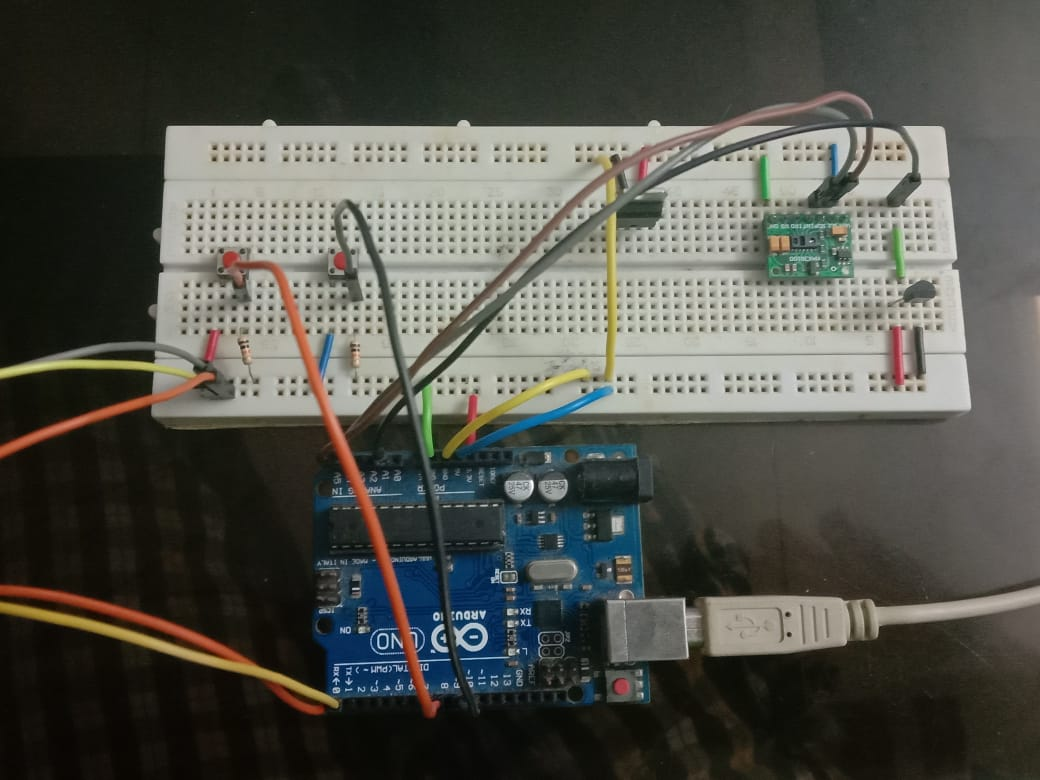


Figure 1(a): Hardware Setup

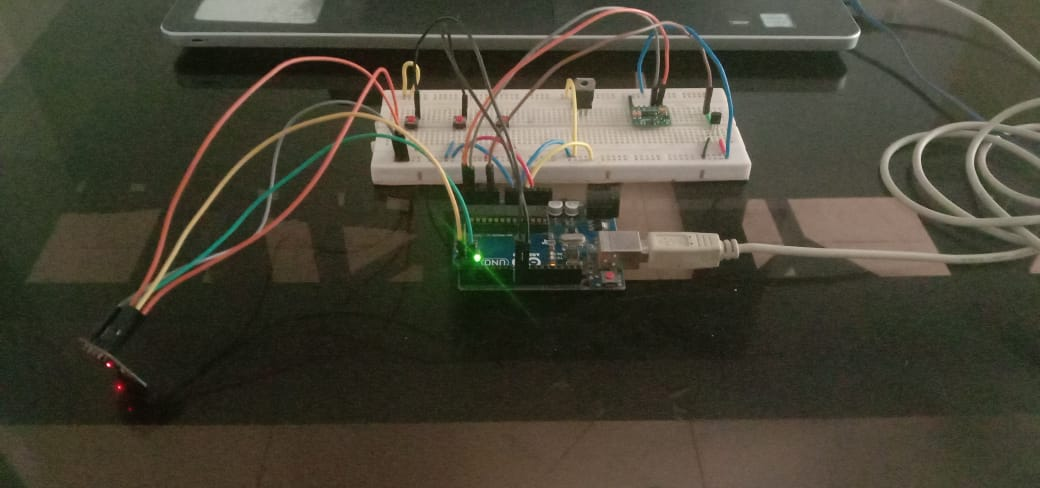


Figure 1(b): Arduino running with LED on WiFi Module blinking

**ARDUINO SERIAL WINDOW**

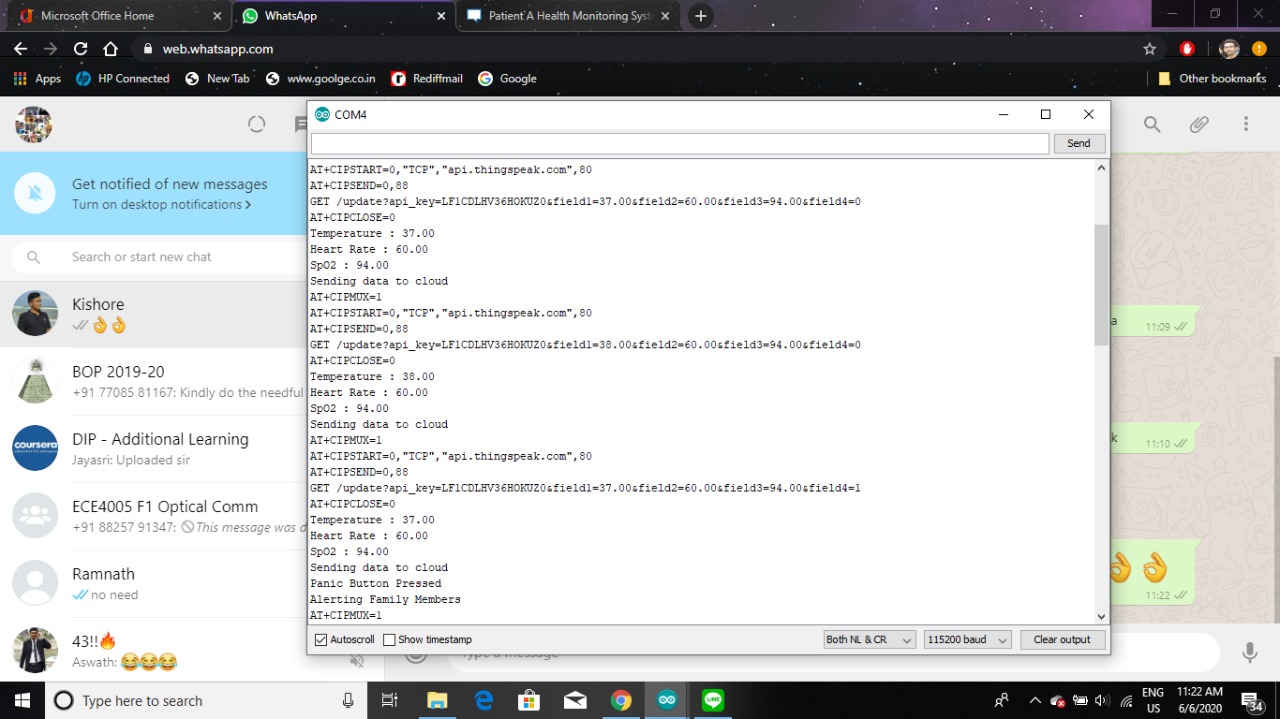


Figure 2: Arduino Serial Window

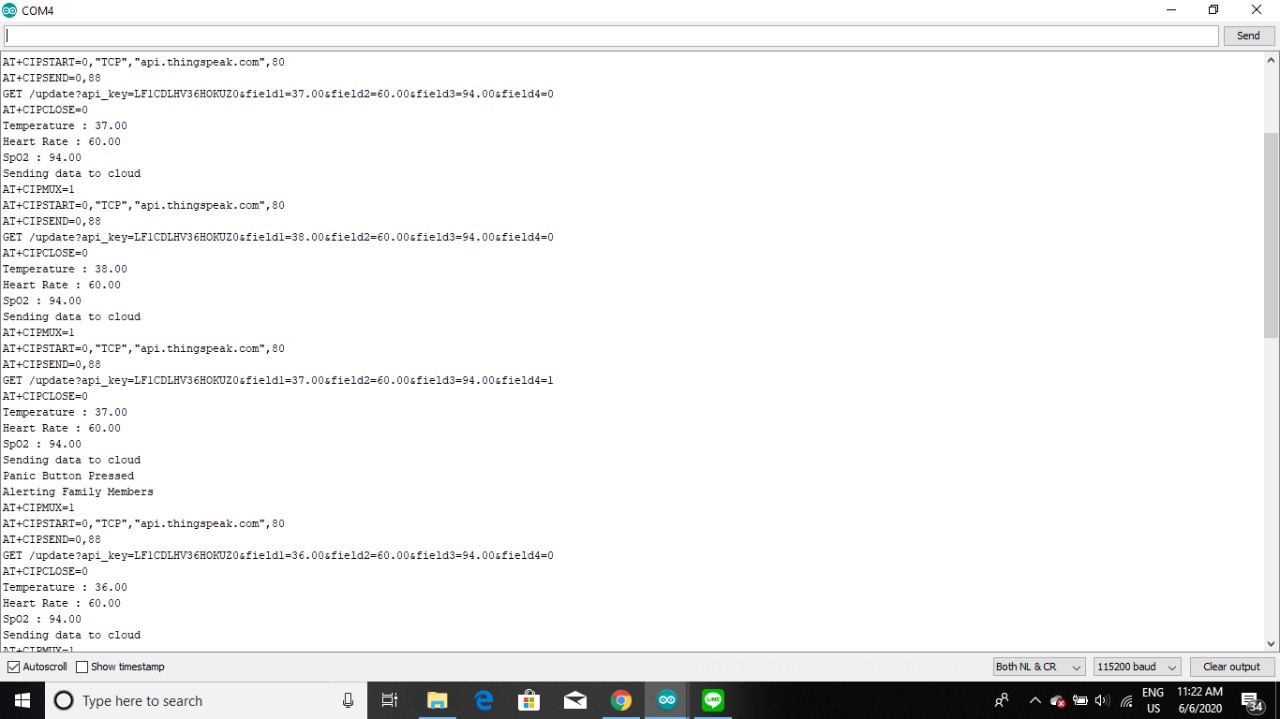


Figure 3: Arduino Serial Window Data

**ANALYTICS IN CLOUD PLATFORM.**

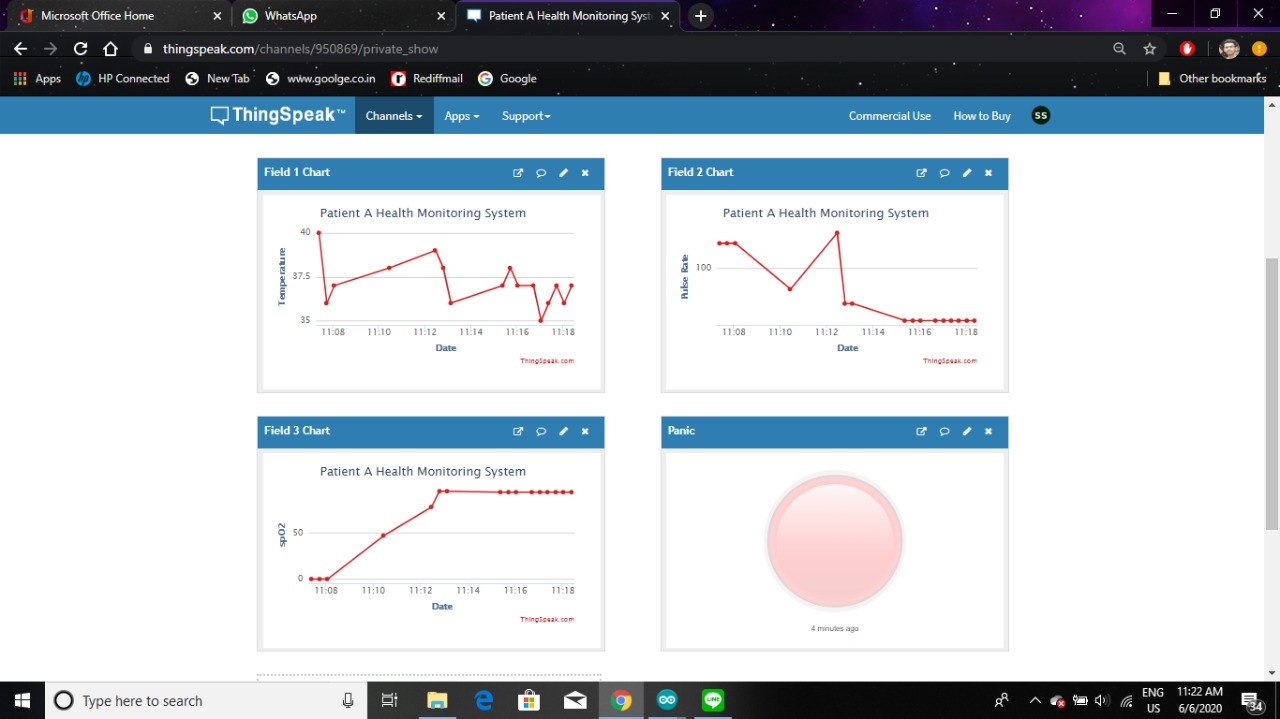


Figure 4: Analytics in ThingSpeak

**MOBILE APPLICATION**

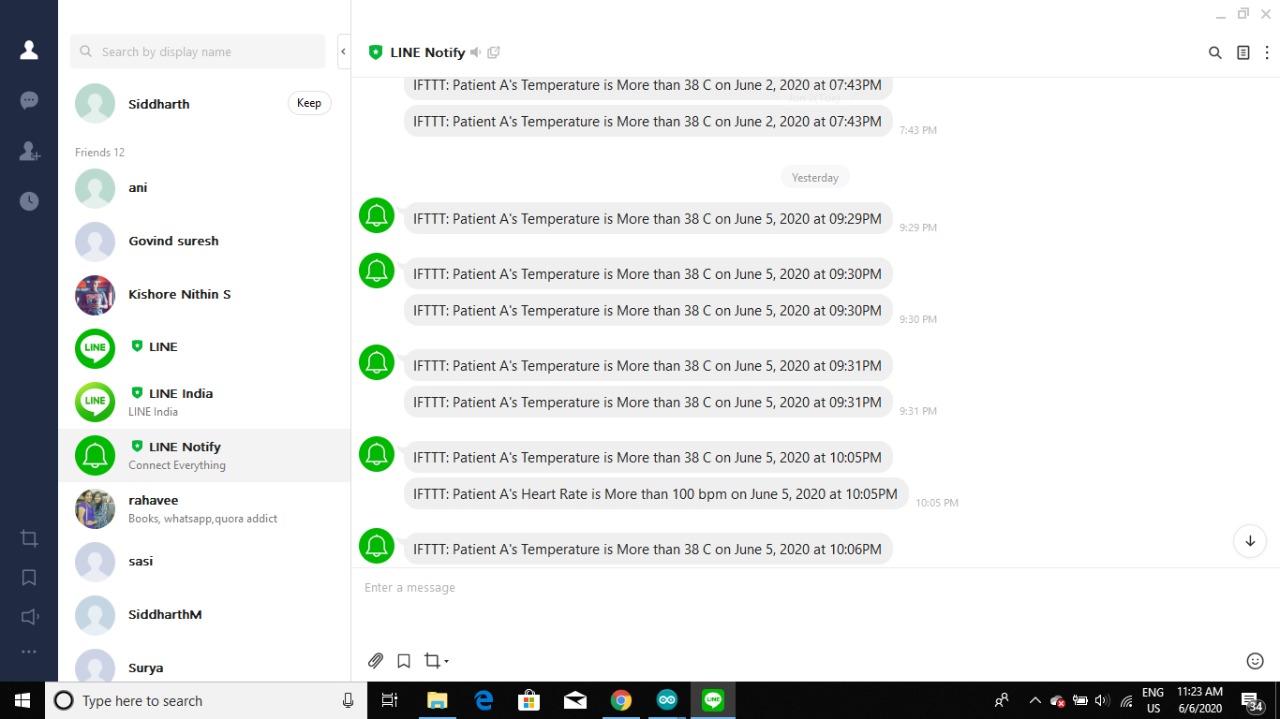


Figure 5: Data of patients accessible through LINE App

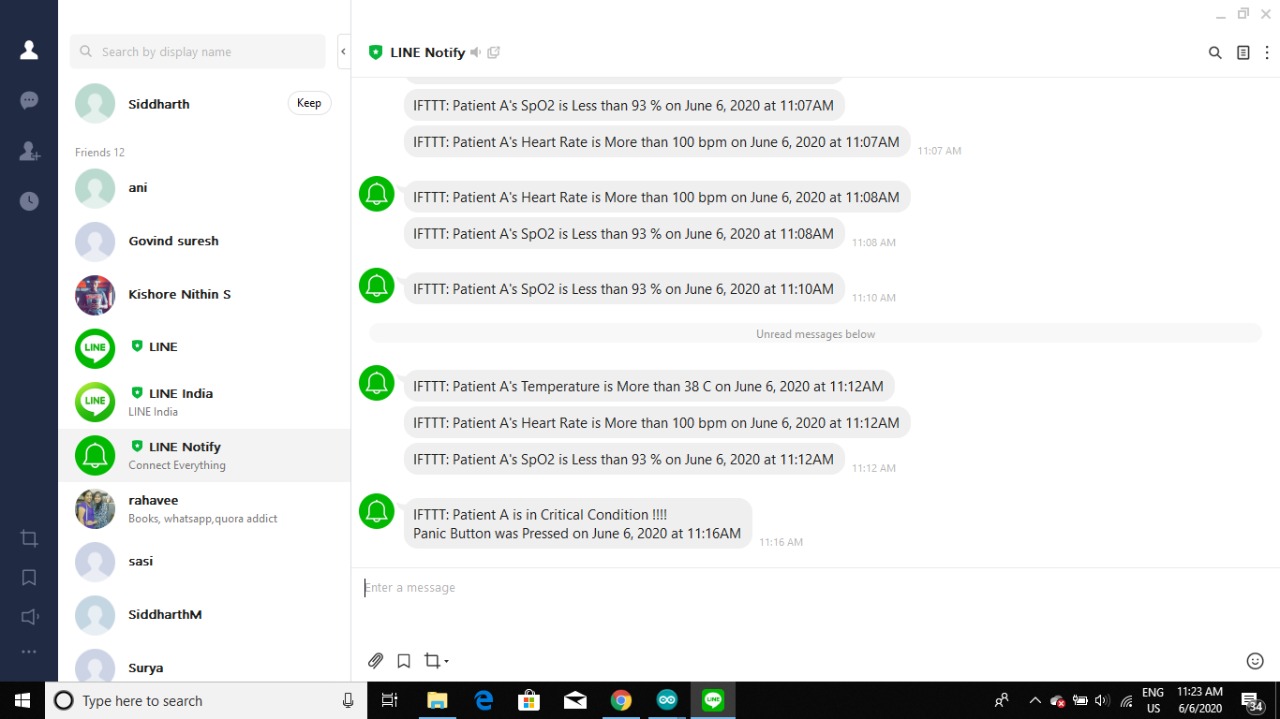


Figure 6: Alert generated when push button is high

**COMPONENTS USED**

1. **Arduino UNO:** **Arduino Uno** is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.
2. **ESP8266 WiFi 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.
3. **LM35 Temperature Sensor:**  LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.
4. **MAX30100 Pulse Oximeter:** The MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times. The MAX30100 sensor measures heart rate and SpO2 content from the blood. Oxygen enters the lungs and then is passed on into blood. The blood carries oxygen to the various organs in our body. The main way oxygen is carried in our blood is by means of hemoglobin. Small beams of light pass through the blood in the finger, measuring the amount of oxygen. It does this by measuring changes in light absorption in oxygenated or deoxygenated blood. Oxygen concentration in the blood can be calculated from the ratio between the absorption red light and IR light by the hemoglobin. Heart rate is detected by the change of blood volume throughout the finger, that is then quantified by the amount of light that passes through the finger.
5. **ThingSpeak API:** ThingSpeak is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_Things) (IoT) application and [API](https://en.wikipedia.org/wiki/API) to store and retrieve data from things using the [HTTP](https://en.wikipedia.org/wiki/HTTP) and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates. ThingSpeak has integrated support from the numerical computing software [MATLAB](https://en.wikipedia.org/wiki/MATLAB) from [MathWorks](https://en.wikipedia.org/wiki/MathWorks), allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks. Other features of ThingSpeak includes:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.
* Automatically act on your data and communicate using third-party services like Twilio or Twitter.

1. **IFTTT:** If This Then That, also known as IFTTT, is a freeware web-based service that creates chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest. IFTTT employs the following concepts:

* Services (formerly known as channels) are the basic building blocks of IFTTT. They mainly describe a series of data from a certain web service such as YouTube or eBay. Services can also describe actions controlled with certain APIs, like SMS. Sometimes, they can represent information in terms of weather or stocks. Each service has a particular set of triggers and actions.
* Triggers are the "this" part of an applet. They are the items that trigger the action. For example, from an RSS feed, you can receive a notification based on a keyword or phrase.
* Actions are the "that" part of an applet. They are the output that results from the input of the trigger.
* Applets (formerly known as recipes) are the predicates made from Triggers and Actions. For example, if you like a picture on Instagram (trigger), an IFTTT app can send the photo to your Dropbox account (action).
* Ingredients are basic data available from a trigger—from the email trigger, for example; subject, body, attachment, received date, and sender’s address.

**CODE**

#include <Wire.h>

#include "MAX30100\_PulseOximeter.h"

String mySSID = "Its ours";

String myPWD = "qwerty12345";

String myAPI = "LF1CDLHV36HOKUZ0";

String myHOST = "api.thingspeak.com";

String myPORT = "80";

String myFIELD1 = "field1";

String myFIELD2 = "field2";

String myFIELD3 = "field3";

String myFIELD4 = "field4";

#define temp\_pin A1

#define PERIOD 1000

float avg\_body\_temp = 0;

float avg\_pulse\_rate = 0;

float avg\_SpO2 = 0;

int panic = 0;

int panic\_button = 8;

long count = 0;

PulseOximeter pox;

unsigned long firstMillis;

unsigned long secMillis = 0;

uint32\_t ts = 0;

void setup()

{

Serial.begin(115200);

pinMode(temp\_pin, INPUT);

pinMode(panic\_button, INPUT);

espData("AT+CWMODE=1", 1000);

espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"", 1000);

pox.begin();

pox.setIRLedCurrent(MAX30100\_LED\_CURR\_7\_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop()

{

pox.update();

firstMillis = millis();

if(firstMillis - secMillis <= 20000)

{

if(millis() - ts > PERIOD)

{

++count;

avg\_body\_temp = ((avg\_body\_temp \* (count-1)) + calc\_body\_temp()) / count;

avg\_pulse\_rate = ((avg\_pulse\_rate \* (count-1)) + pox.getHeartRate()) / count;

avg\_SpO2 = ((avg\_SpO2 \* (count-1)) + pox.getSpO2()) / count;

ts = millis();

}

}

else

{

avg\_body\_temp = round(avg\_body\_temp);

avg\_pulse\_rate = round(avg\_pulse\_rate);

avg\_SpO2 = round(avg\_SpO2);

panic = digitalRead(panic\_button);

String sendData = "GET /update?api\_key=" + myAPI + "&" + myFIELD1 + "=" + String(avg\_body\_temp) + "&" + myFIELD2 + "=" + String(avg\_pulse\_rate) + "&" + myFIELD3 + "=" + String(avg\_SpO2) + "&" + myFIELD4 + "=" + String(panic);

disp\_in\_site(sendData);

Serial.print("Temperature : ");

Serial.println(avg\_body\_temp);

Serial.print("Heart Rate : ");

Serial.println(avg\_pulse\_rate);

Serial.print("SpO2 : ");

Serial.println(avg\_SpO2);

Serial.println("Sending data to cloud");

if(panic == 1)

{

Serial.println("Panic Button Pressed");

Serial.println("Alerting Family Members");

}

count = 0;

avg\_body\_temp = 0;

avg\_pulse\_rate = 0;

avg\_SpO2 = 0;

secMillis = firstMillis;

}

}

String espData(String command, const int timeout)

{

String response = "";

Serial.println(command);

long int t = millis();

while((t + timeout) > millis())

{

while (Serial.available())

{

char c = Serial.read();

response += c;

}

}

return response;

}

void disp\_in\_site(String data)

{

espData("AT+CIPMUX=1", 1000);

espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+ myPORT, 1000);

espData("AT+CIPSEND=0," +String(data.length()+4),1000);

Serial.println(data);

espData("AT+CIPCLOSE=0",1000);

}

float calc\_body\_temp()

{

return (analogRead(temp\_pin) \* 0.48828125) - 7;

}

void onBeatDetected()

{

}

**FUTUTRE WORK**

Health Monitoring System is an up-and-coming system and has high potential to revolutionize health sector. Some simple changes that can increase the efficiency of the system will be to use high precision sensors. Adding on to the sensors currently used in the system more sensors like blood pressure sensors, sensors to check insulin levels in blood etc., can be used in the system to improve its functionality. The data collected by these systems can be used to train machine learning models and can be used to predict health issues that may come in the future. To add on another feature to this system, medicine alert system can be integrated to this system and medicine logs can be sent to doctors.

**CONCLUSION**

Health Monitoring Systems can make our lives easier and it needs to be made affordable buy all means. By using this system, the healthcare professionals can monitor, diagnose, and can be treat patients better all the time. Patients can be monitored remotely when they are in transit on their way to hospitals and doctors can be ready with necessary arrangements to treat the patient without any delay. The health parameters data are stored and published online. Hence, the healthcare professional can monitor their patients from a remote location at any time. Our system is simple but is prone to errors due to the inaccuracies of the sensors used.

Oximeter is very hard to use and requires external modification before use. The future work of the project is very essential in order to make the design system more advanced and very accurate is which a primary requirement for these kinds of systems. In the designed system the enhancement would be connecting more sensors to internet which measures various other health parameters and would be beneficial for patient monitoring i.e. connecting all the objects to internet for quick and easy access. This system also increases the transparency in nursing as we can check the data to see if patients were given the necessary treatments.

**REFERENCES**

* K. Navya, Dr. M. B. R. Murthy, “A Zigbee Based Patient Health Monitoring System”, Int. Journal of Engineering Research and Applications Vol. 3, Issue 5, Sep-Oct 2013, pp.483-486
* Health Monitoring Systems using IoT and Raspberry Pi – A Review - <http://www.kresttechnology.com/krest-academic-projects/krest-mtech-projects/IOT/Mech%20IOT-2017-18/IOT%20Basepaper%202017-18/35.Health%20Monitoring%20Systems%20using%20IoT%20and%20Raspberry%20Pi.pdf>
* IoT Based Patient Monitoring System using ESP8266 and Arduino - https://circuitdigest.com/microcontroller-projects/iot-based-patient-monitoring-system-using-esp8266-and-arduino