IoT based Health Monitoring System for Elderly People

Submission Date: 18th April 2017

Mohammed Fathe Mubin (13101216)

Faysal Ahmed (13101225)

Md. Rafiul Islam (13101172)

Department of Computer Science and Engineering

Supervisor:

Amitabha Chakrabarty, Ph.D

Assistant Professor

Department of Computer Science and Engineering

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researchers conducted by others are mentioned in references.

Signature of the supervisor

____

Dr. Amitabha Chakrabarty

Department of Computer Science and Engineering

BRAC University

Signature of the authors

____

Mohammed Fathe Mubin

(13101216)

____

Faysal Ahmed

(13101225)

____

Md. Rafiul Islam

(13101172)

Table of Contents

List of Figures

Abstract

Internet of Things (IOT) is a where data collection is done via devices which communicates with each other and stores those data in cloud. From which it is possible to extract, analyze and send data way faster with efficiency. IOT has paved the way to new possibilities in different areas and industries. This indicates from smart home appliances to the smart self-driven car, to remotely observe and control different objects. Healthcare has been secluded from this revolution for long as its diversity and heterogeneity. Health monitoring of elderly is one of the most crucial subject in modern era health care. And it is a new challenge to remotely monitor the health condition. In this paper, we will depict the current situation of the technology of the health monitoring projects based on IOT and propose how we can improve the actual implementations of the monitoring system for the health care of the elderly people. We will also discuss how we can make the concerning people aware with precision if any alarming situation is spotted so that instant action can be taken thus ensuring reduction of casualty.

Acknowledgement

Firstly, we would like to thank the Almighty for giving us the opportunity, perseverance, determination and intellect to complete this project.

A big and warm gratitude goes to our supervisor Dr. Amitabha Chakrabarty for helping us all the way to complete our task successfully in time. He always pushed us to the edge, gave us healthy challenges and helped us through our thick and thins. It was a pleasure working under his guidance and supervision.

Finally, we would like to thank all the faculty members of the Department of Computer Science and Engineering of BRAC University from whom we have gained knowledge and assistance to complete our thesis work.

Chapter-1

Introduction

In the introduction part we will discuss what is Internet of things all about. How important Health Monitoring is in modern era of health care. And more importantly, why IOT should be integrated to health monitoring to improve the quality of service in health care sector? In the later parts we will discuss how we implemented the system.

1.1 Introduction

The term "Internet of Things" is considered to be first coined in the starting of this century when work was done on MIT Auto-ID Center [1], to make a smart identification technology which will help to reduce the error rate subsequently increasing efficiency and to automate. But since then, the concept of IOT has evolved rapidly in various ways, as now with the help of this huge number small networks which can remain connected to each other and can directly send data to the main network without any human interaction.

Quality of service in healthcare has always been under constant criticism in the modern era, as it is a very touchy subject. Health monitoring specially for elderly people is a concern and as most people in the modern times are job holders and have so hectic life. It is difficult to manage to keep a constant watch on the elderly of the house. Keeping a nurse or housekeeper is also a very costly issue nowadays. In this situation, remote health monitoring based on IOT can help to solve the problem.

IOT is providing the means by which it is possible to collect and analyze data remotely without any human interaction. So, this indicates that it is possible to detect and prevent any future hazard with precision and possible to aware the concerning authority like the family member or the physician if there is any alarming situation. The basic two reasons IOT is important for this project is firstly it is automated, so no human interaction is needed. And secondly, because of automation the process have less chance of having errors i.e. having a more efficient system indicating a better quality in service.

In this paper we would describe how we collected and analyzed data using Thinkspeak server. Also how it was possible to monitor all the collected data from Thinkspeak server. Lastly, how we made reaction panel to cope if any alarming situation is looming in future so it can be preventable.

1.2 Objective

- Making an automated system which will help to monitor host remotely is our primary objective.
- Making an alarm or reaction system which will react whenever there is an alarming situation.
- Providing a way to remotely monitor the temperature, pulse, counting the bowel discharge in a day and also the mount of sleep of the patient via Thinkspeak.
- Analyzing the collected data using the built in Matlab of the Thinkspeak sever to detect future hazards.
- Sending alarming messages via e-mail and twitter to the concerning authority or people if any abnormality is detected.
- Contributing in the field of IOT to pave a way for future project in the technological development.

1.3 Motivation

The core to our motivation was to think ahead of our time and to make a contribution on the sector Internet of Things (IOT) which is undoubtedly the next big thing on the technological market. And IOT has proven itself to be a very handy possession. The idea of this project came to us by observing how easy it is to connect or integrate the normal home appliances with the internet. This led us to the secluded health sector which is still lagging behind from other industries in terms of technology. We observed many case even from our daily lives that in the hectic world we are living right now, it is difficult to keep a watch over everything. Specially, to keep an eye on the elderly people is difficult and costly as well. These real life situations gave us the ideas and motivation to integrate the two sectors so that it helps in developing such a project which will create an economic and efficient health monitoring system and will pave the way for future work on the field of IOT.

1.4 Thesis Outline

Chapter-1: In this chapter we gave the introduction to our ideas for this project, the motivation we got from and our objectives with this project.

Chapter-2: This chapter describes the literature review for this project. It discusses the concept of this project, related and similar works done (if any) and how we differ from them.

Chapter-3: In this chapter, we described the steps of method we followed to complete the project, which includes data collection, data storing, analysis with the data's etc.

Chapter-4: This chapter gives a through view on the result that is obtained after the analysis from the data.

Chapter-5: The conclusion is given in this chapter.

Chapter-6: The references are given in this chapter.

Chapter-2

Literature Review

This chapter discusses about the concept of the project we are trying to implement, what kind of work has already been done on before and how we are different and more improved than those projects.

2.1 Concept

Monitoring health of elderly people is basically model for monitoring using different sensors. The reason we chose elderly people for monitoring is because, elderly people are usually more vulnerable to sickness and other aging factors. So, usually it becomes difficult for working people to monitor the senior members of the family the whole time. Even if it is possible to take care of the elderly during they stay at home, it becomes rather difficult to observe their activities and condition during the working hours. Thus, it was eminent to come with a solution that is to make a health monitoring system which can observe the daily basic activities of elderly people. A threshold value will already be provided to the system. The system will collect data of daily activities through sensors which will be placed according to the needs of the system. The retrieved data will then be compared with the provided threshold values. If everything remains normal then further analysis will not be done. But whenever any anomalies or abnormalities are recorded then the data's are further analyzed using the appropriate predictive algorithm. Then with the help of those predictions the system will decide how serious the situation is. If it is really that serious then a message will be send to the concerning relatives about the recent condition of the patient and about the prediction of the system.

The basic things that will be tried to be monitored in this research are pulse rate, temperature, usage of wash room, as well as amount sleep of the patient. The sensors used in the research will work in following steps:

1. The sensors will send the data collected from the host in a regular basis after definite span of time.
2. The data thus collected will undergo a comparison with the given threshold value to the system.

3. If the data set concurs with the threshold value then the situation will be considered to be normal thus the system will not take any further action.
4. The data set thus retrieved if contains any abnormities, will then go for further analysis to predict how serious the condition of the person is. And also send a alarming message to concerning authorities.

2.2 Related Works:

An extensive research on the topic related to the system shows a very few of the related works could actually build their own preliminary framework and prototype of the system. Some of the works like the research conducted on the **Ambient Assisted Living (AAL)** [2] actually did more of a literature survey of the state of its present condition of the monitoring system via IoT. They also tried to identify and highlight the critical issues and the quality of service as well as the user driven experiences in their work. Some, other worked on showing or highlighting the importance of IoT in the health sector and some proposals for the health monitoring architectures.

Some related findings used specific models for the health monitoring aspect. Like the abstraction of **Model Driven Tree Reference Model (MDTRM)** [3], where they explained the necessity of this model in the health field as well as identifying the complexities of the models. They also benched marked the models which came really handy for the initial phase of this research.

Some other related model we found are **General Domain Model Architecture (GDMA)** [3], the health monitoring and sensing with cloud processing was also a helpful source behind the research, as it was useful for generating ideas to get raw data’s from wearable devices which are compatible and capable of measuring many physical value which we can use to obtain meaningful results.

Masimo Radical-7 [4], a health monitor for clinical environment helps to collect data and wirelessly transmits it for ongoing display. This provides high resolution display of information with higher graphical capabilities. It also has a touch based user interface. But as it can be already assumed how cost effective it is, it can’t send an alarm message to notify for any emergencies. **Free Scale Home Health Hub reference platform** [4] store patient data to cloud via various sensors, where the people related to the patient can have an access. This platform too can’t notify for any alarming situation to the people engaged with the patient.

Some surveys of ours also lead us to projects which even discussed to monitor the health whole area through wireless network sensors. [5][6] They also tried to share their ideas by giving a model of their framework like cloud based processing [7] and big data. [8]

2.3 Thesis Contribution:

Of all these ideas, models, frameworks and platforms that are surveyed for this research, we differ with all the above with a very basic and fundamental ways. The above researches conducted may use many monitoring variable on the contrary we are using more specific variable. Still, the basic functionality difference is that our motive to develop this project is to generate a functional response and to give a feedback to the relatives of the host, so that they can quickly take steps for the wellbeing of the host. The source of this response will be e-mail and social networking site twitter. But the basic idea of the research is totally of a different paradigm.

Though, it is true that health monitoring and the prediction from anomalies as well as giving a useful feedback to the use is neither a very easy process nor the framework a very easy to set foot to. But, this research will take the health monitoring which now one of the most exciting topics (related to IoT) to a very different level. The useful information gathered from the research conducted has provided very handy to carry out and provide a workable model which will be discussed in the next segment of the paper. Workable data is something which we had to toil to obtain but the data's were managed properly to continue the research.

Thus, most the related works done related to our project was highlighted in this segment and how our research differs from those works were also explained thoroughly. Some of those researches conducted really helped us understanding the importance of this sector and inspired to go the extra-mile on this field. The platform monitoring the daily activities of the host will be useful for the physician attending the host. The daily activities monitoring conducted by the system will help the physician to develop a conclusion and prediction to diagnose the host in a swift manner.

Chapter-3

Methodology and Implementation

This chapter discusses the hardware required for the project implementation, discusses with figure the circuit connections and pseudo codes for the implementation.

3.1 Hardware

To conduct this project we need sensors so that we can monitor the health condition of the elderly people. In order to do so, we chose four different criteria's to monitor the elderly people. The four criteria's we chose are body temperature, pulse rate of the host, how long the person is sleeping and the number of time a person is using the toilet thus calculating how much time the host is spending there. So, for those four criteria's we chose the following sensors.

1. Temperature Sensor

For collecting the temperature data from the host we chose the model **DS18B20.** The DS18B20 [9] digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line eliminating the need for an external power supply. This sensor is very easy to work with and provides with accurate data of temperature. The sensor provides data in Celsius temperature scale but we later converted it to the Fahrenheit scale for our convenience. Whenever the host touches the sensor, the sensor saturates itself with the body temperature of the host and shows that temperature.

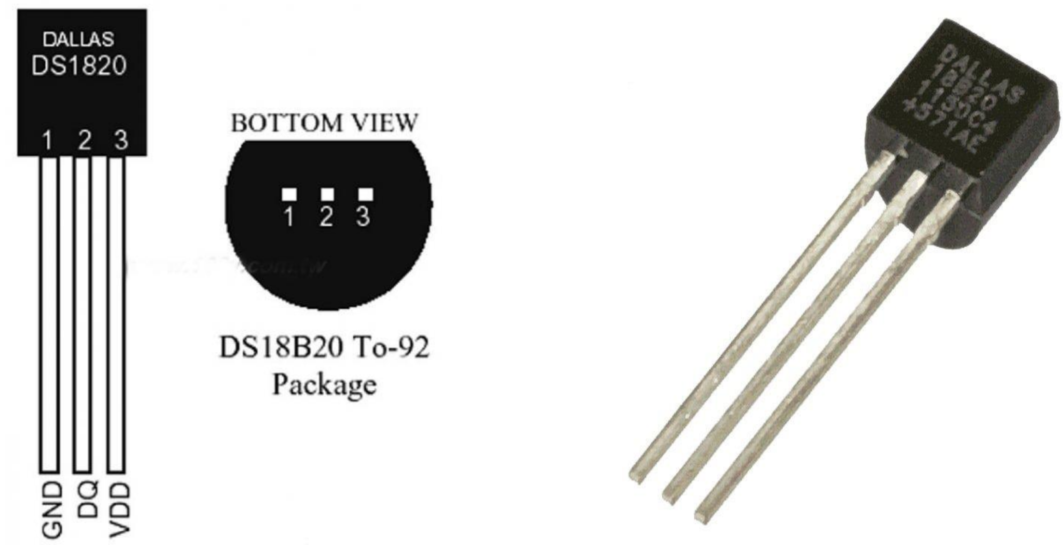


Fig 1: Temperature Sensor DS18B20

2. Pulse Sensor

For collecting the pulse rate data from the host we chose the model **SEN-11574** [10]. The Pulse Sensor is a plug-and-play heart-rate sensor for Arduino. It can be used by anyone who wants to easily incorporate live heart-rate data. Essence it is an integrated optical amplifying circuit and noise eliminating circuit sensor. It is very easy to use by clipping the Pulse Sensor to the hosts earlobe or fingertip and plug it into the Arduino. This sensor can give data’s like Pulse Rate, Cardio Graph and Inter Beat Interval. However for our convenience we only used the Pulse Rate per minute from the sensor. The data from the sensor can be retrieved from the host via the host’s fingertip or the lobe of the ear.

Fig 2: Pulse Sensor SEN-11547

3. **Ultra Sonic Sonar Sensor**

In order to collect the data of the number of times a host uses the toilet we chose the model **HC-SR04**. [11] The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1” to 13 feet. It operation is not affected by sunlight or black material like Sharp rangefinders are although acoustically soft materials like cloth can be difficult to detect. We set the ultra-sonic sonar sensor on the door of the toilet. In this way, whenever the door is moved by the host, it gives a notification and thus we can count how many times it is moved thus the number of time the toilet is used.

Fig 3: Ultra Sonic Sonar Sensor HC-SR04

4. **Pressure Pad**

A homemade cost effective pressure pad [12] using cardboard and foil paper. The card board gives the data in binary value. Whenever, there is pressure on the pressure sensor, then it gives 1 and whenever there is no pressure it gives a 0. So, until the host is asleep we can obtain a 1 and when he is awake and the pressure is off the value is back to 0.

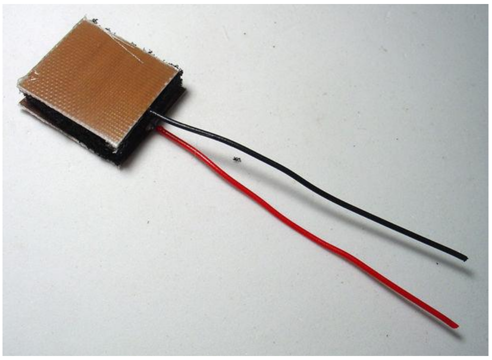


Fig 4: Pressure Pad

5. **Arduino**

We used Arduino Uno for the purpose of this project. All the sensors are connected with the Arduino. Arduino Uno [13] 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 quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. The sensors are powered from the Arduino Uno so is the Wi-Fi module. The Wi-Fi module is getting the data’s from the sensors through this Arduino Uno.

Fig 5: Arduino Uno

6. **Wi-Fi Module**

For the purpose of this project, we chose ESP-8266 [14]. The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller for our case the Arduino access to Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module is pre-programmed with an AT command set firmware, meaning, it can be simply connected with the Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers! The ESP8266 module is an extremely cost effective board. The module helps to transmit the sensor data to Thinkspeak cloud storage. It remains connected with the Arduino and connects with the server via internet and sends the data to that server.

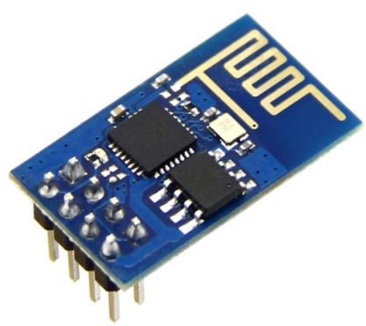


Fig 6: ESP-8266 (Wi-Fi Module)

3.2 Circuit Connection

3.2.1 Wi-Fi Module

The ESP-8266 has eight ports, one of them is the power port which is used to supply power to the module and this was connected with the Arduino’s 3.3v. ESP-8266 is very sensitive to high power so it was not connected to the 5v power besides it works better at 3.3v. Another port of the ESP-8266 is the ground port which is connected with the Arduino’s ground. ESP-8266 sends and receives data with 2 different ports these ports are called TX and RX ports. ESP-8266’s TX port was connected with Arduino’s digital port number 2 and ESP-8266’s Rx port was connected with Arduino’s digital port 3. There is a reset port on the ESP-8266 which was not used as we did not need it, there is a port called ch-pd on ESP-8266 which is connected to the Arduino’s 3.3v port. There are 2 GPIO ports on the ESP-8266, the GPIO1 is connected with the Arduino’s 3.3v and the other port, GPIO0 is left unused.

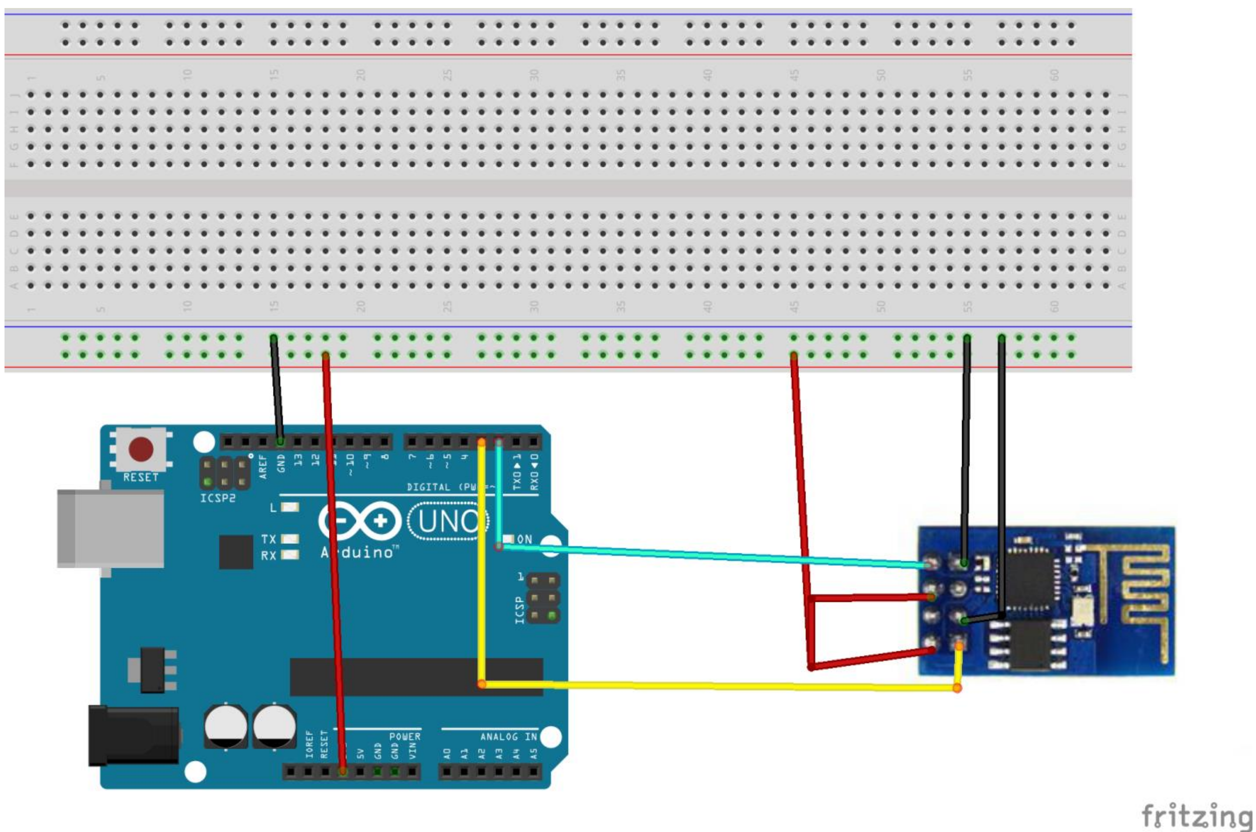


Fig 7: Circuit of ESP8266

3.2.2 Temperature Sensor

The DS18B20 has 3 pins to operate. One pin, VDD is for power and another pin is for ground. The DS18B20 can operate with power from 3.0v to 5.0v. The VDD port is connected with the Arduino's 3.3v through a 4.7ohm register. The ground port is connected with Arduino ground. The other port of the DS18B20 is the data port. This port is used to send temperature data to the Arduino, this port can also take power from the Arduino if it needs any extra. The data port of the DS18B20 is connected to the Arduino digital pin 5 through the same 4.7 ohm register. So, basically one end of the 4.7ohm register is connected with the VDD port and the other end of the register is connected with the data port. This register is put with both these ports so that overpower cannot harm the DS18B20 as the data port also may take power from the Arduino.

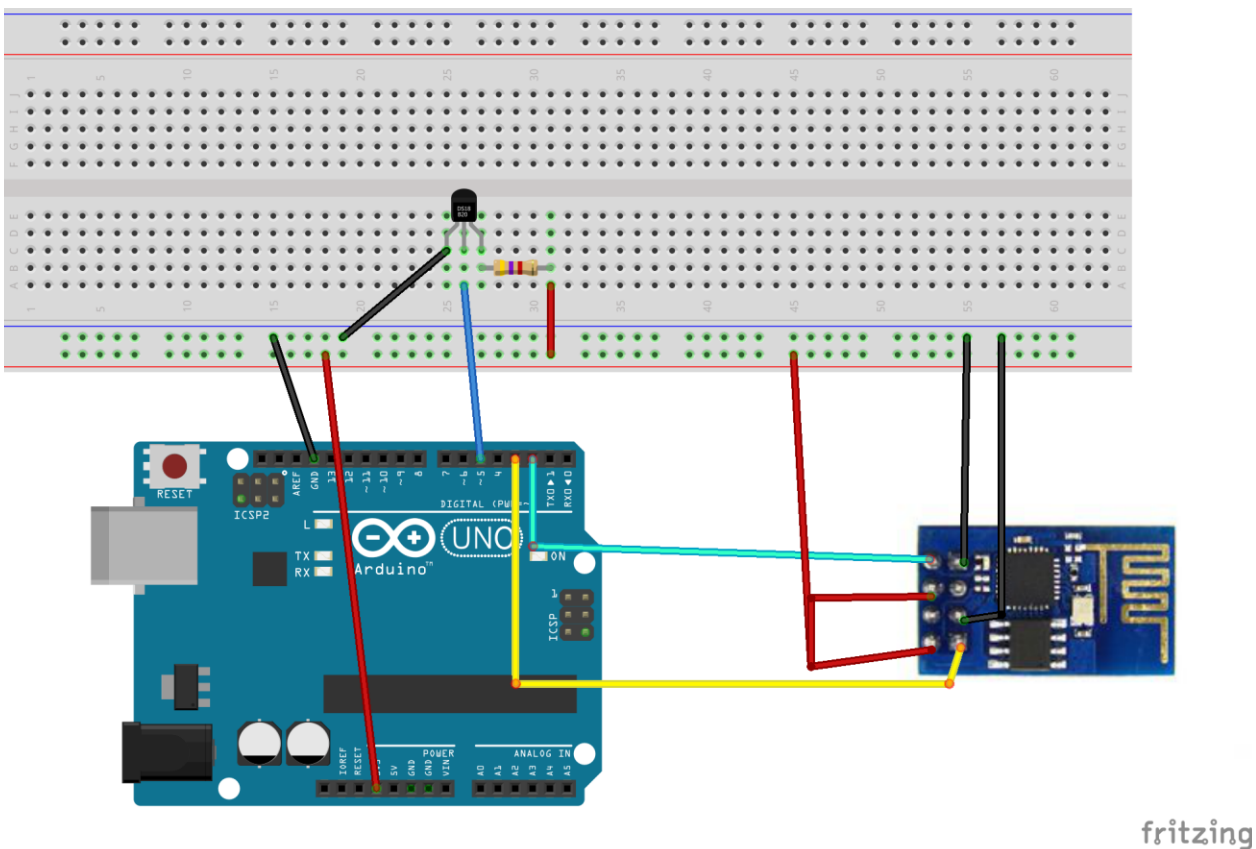


Fig 8: Circuit diagram of the DS18B20, temperature sensor.

3.2.3 Sonar Sensor

The sonar sensor we used is called HC-SR04. This sensor has 4 ports one of these four ports is the VCC which is connected with the Arduino's 5v connection. Another port of the sonar sensor is the ground port which is connected with Arduino's ground port. There is a port named TRIG on the sonar sensor which can cast ultra-sonic sound waves this is connected with Arduino's digital port 9 and the other port is called ECHO that is connected with the Arduino's digital port 10. To ensure that the sonar sensor is working, a LED light is attached with the Arduino.

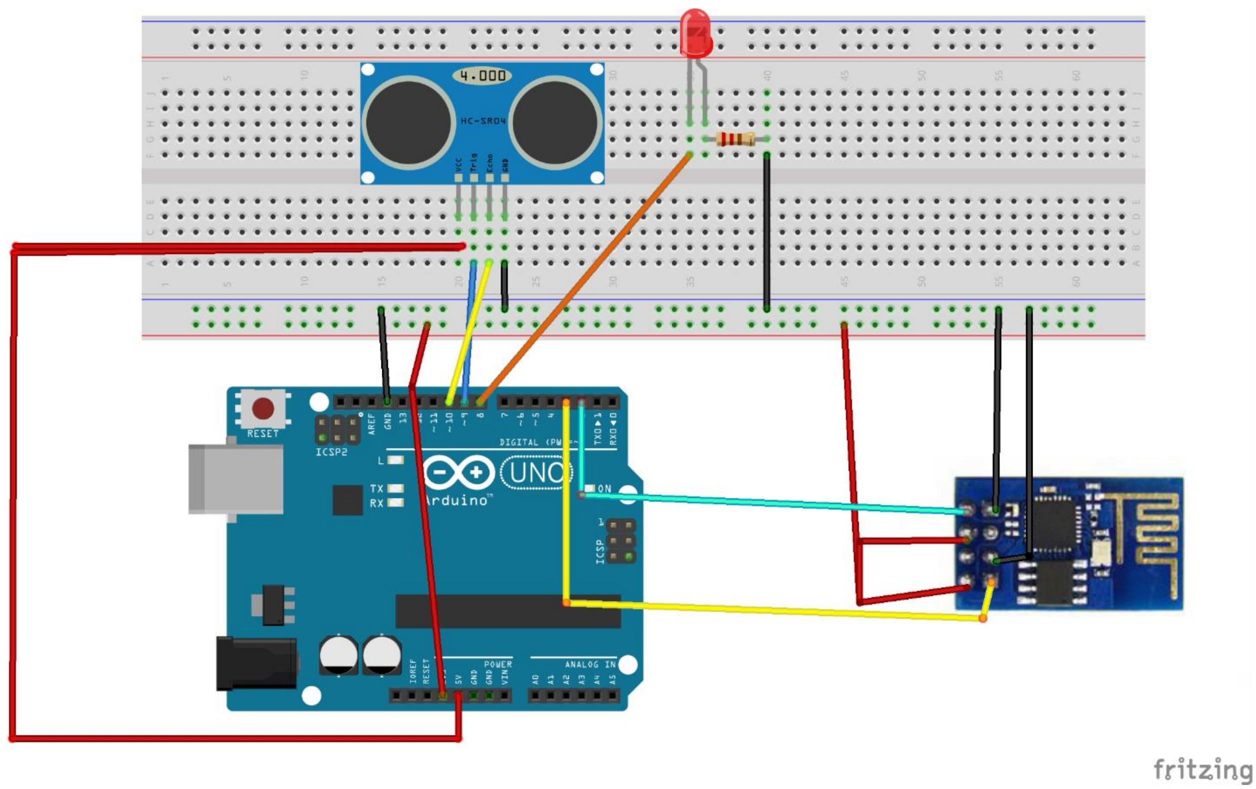


Fig 9: Circuit of Sonar Sensor (HC-SR04)

3.2.4 Pressure Pad

A pressure pad is made with card-board paper and conductive foil paper. Two separate parts of the pressure pad is connected with 2 different part of the Arduino. One part of the pressure pad is directly connected with the Arduino's 5v power supply and the other part of the pressure pad is connected to the Arduino's digital pin 7, a 1Kohm register is connected with this part of the pressure pad the other end of the register is connected to ground.

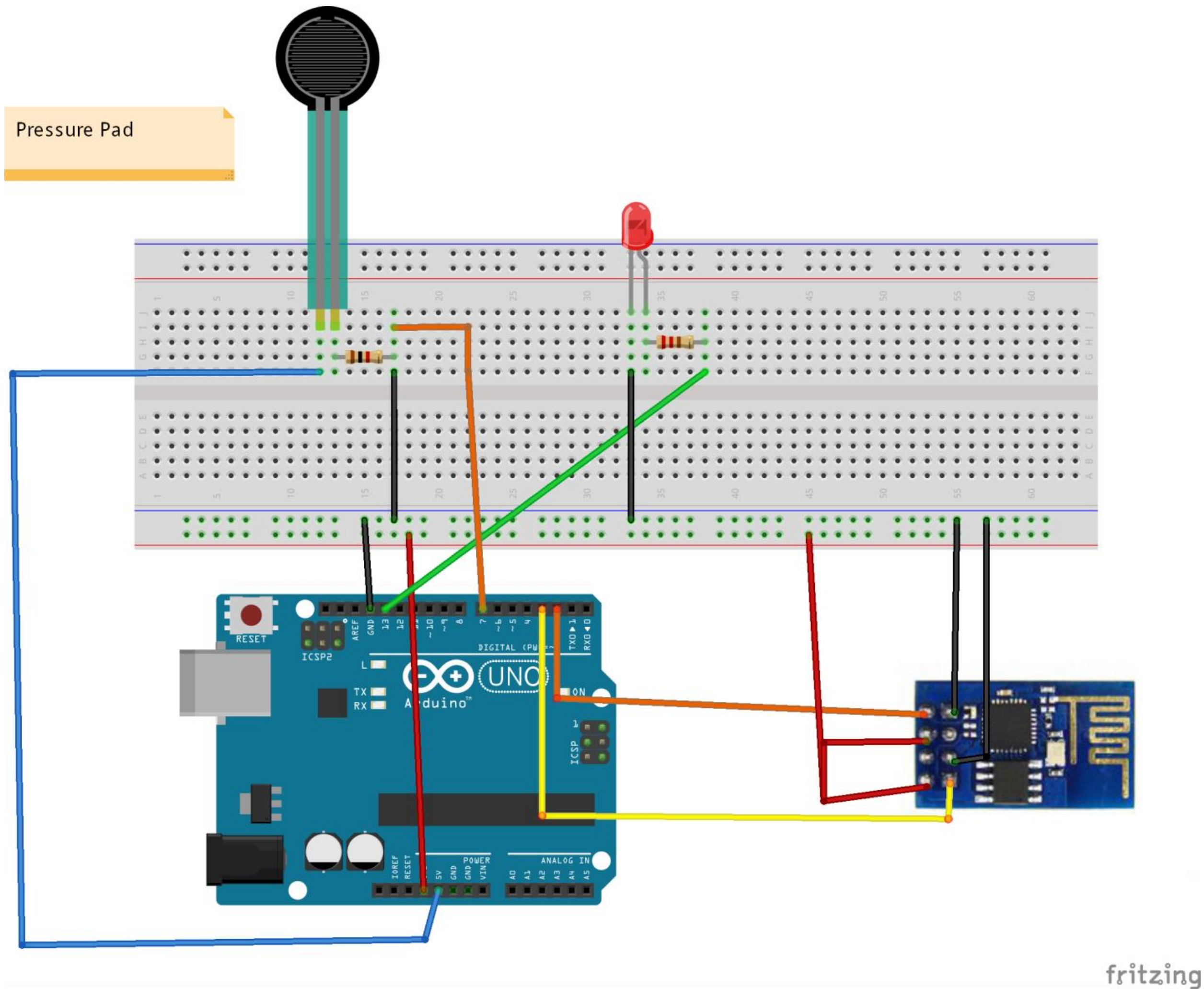


Fig 10: Circuit diagram of Pressure Pad

3.2.5 Pulse Sensor

The pulse sensor has three pins; one of them is the VCC pin to power the pulse sensor as usual. This pin is connected with Arduino's 5v power supply.

There is also a ground pin in the pulse sensor which is connected with the Arduino's ground port. The other pin of the pulse sensor is the data pin. This pin sends analog data to the Arduino so we had to attach this pin with the Arduino's analog pin 0. This way the Arduino will be converting the analog data to digital and we can avoid using an extra analog to digital converter.

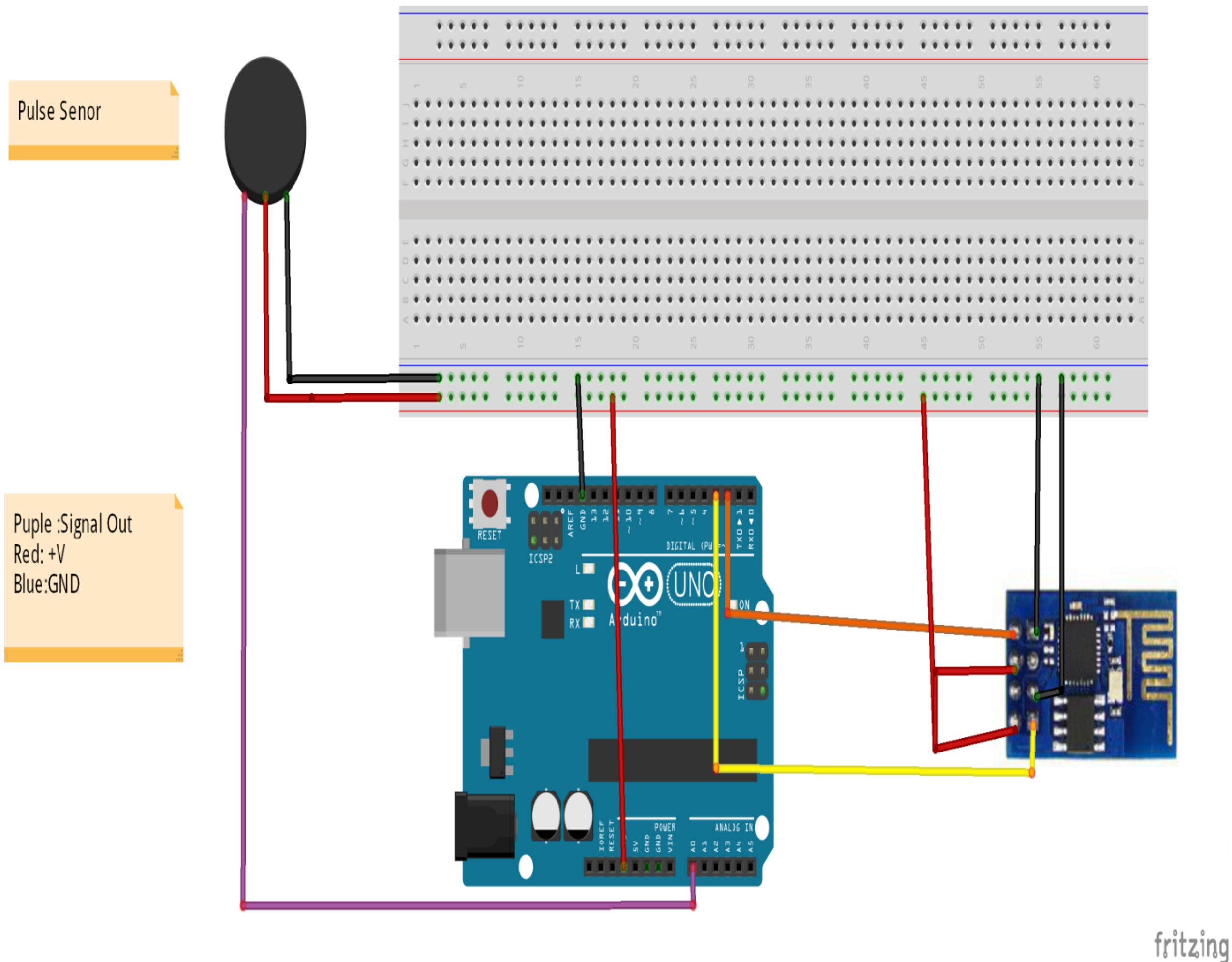


Fig 11: Circuit diagram of Pulse Sensor

3.3 Data Collection

After we are done connecting all the hardware and the Arduino, we can collect data from the sensors through the Arduino.

The temperature sensor is a small semi-circle hardware, to measure temperature the host just has to put their finger on the temperature sensor and the sensor will gradually adapt with the hosts temperature and take that data in the Arduino. The data that this sensor collects is in Celsius format so we have to convert it to Fahrenheit format.

The pulse sensor is a small round, disc shaped hardware, it has a green light in middle of the sensor, when the sensor receives power from the Arduino this green light will be lit. To measure the pulse of the host, they simply have to put their finger on the green light and the sensor will automatically send the pulse per minute data to the Arduino.

The pressure pad will be used to measure the sleep duration of the host. The pressure pad will be kept under the pillow of the host, so whenever the host lie down on the pillow the two separate part of the pressure pad would be connected so a high voltage will be counted on the Arduino and the duration of this high voltage is considered as the duration the host slept.

The sonar sensor was used to keep track of the host's toilet schedule. It is very important to measure if a person is having regular number of times that they need to visit the toilet. The sonar sensor will be placed inside the toilet room so that whenever the host is using the toilet the sonar sensor will detect his presence and count that to measure the number of times they visit the toilet in a day to determine their health situation.

3.4 Data Storage

The data that were collected from the sensors through the Arduino should be stored somewhere to analysis on those data. For that purpose Thinkspeak, a website that allows storing data and analysis them is used in this thesis. The ESP-8266 is used for connecting the Arduino with the internet and stores that necessary data's to Thinkspeak server. The ESP-8266 will receive data's from the Arduino and with the help of a Wi-Fi connection, the ESP-8266 will send the acquired data's to Thinkspeak server.

For different data's different types of channels are created in Thinkspeak. The temperature of the host can be continuously uploaded to Thinkspeak with the ESP-8266. From the temperature sensor, the temperature of the host will be saved at one channel and also the highest temperature measured will be stored too. This way we can alert the responsible person in terms of constant high temperature on the host.

Same goes for pulse sensor data's, pressure pad's data and the data collected from the sonar sensor. All of these are being stored in the Thingspeak server to analyze.

3.5 ThinkSpeak

Thinkspeak is an open source IoT application and API. It helps to store and retrieve data from things using the HTTP protocol all over the internet or from the local area network (LAN). ioBridge launched Thinkspeak in 2010 to support IoT based application. It allows its users to use MATLAB which they integrated to analyze and visualize the uploaded data without purchasing the license. Thinkspeak was written in ruby, has a Cross-Platform operating system and stores its data in cloud.

A user can have a free sign in Thinkspeak and after signing can open up unlimited channel under that account. Each channel has different fields and in those fields data's are displayed in graphical forms. User can analyze the data according to the need using the MATLAB. This manipulation of data is one of the most fascinating features of Thinkspeak.

Another important feature of Thinkspeak is it provides a react option to its users. The react option helps to give an instant reaction to the concerning authority whenever a definite condition is fulfilled via social networking site Twitter and even e-mail as well. This is one of the most unique features of Thinkspeak which helped us along the way to complete the project.

3.6 Steps of Implementation

For completing this project and to monitor the health criteria's of elderly people fully automatically, we had to follow some steps. There is a systematic way of doing all this things to operate the sensors the Wi-Fi module and to manage the data's in the Thinkspeak. The sequential steps of this thesis is stated below,

1. Individually collect data from the sensors through Arduino.

2. Send the data's to Thinkspeak with the ESP-8266.

3. Use Thinkspeak to sort the different data's into different channels.

4. Analyze the already stored data.

5. Display the data's to Doctors and concerning people for relative host.

6. Alarm the necessary person for abnormal health situation.

3.7 Implementation

After all the detail discussion about the hardware connections and introduction to Thinkspeak, in this segment we are going to follow the above methodology to complete the implementation of this thesis.

3.7.1 Individual data collection

As we are implementing a project based on IoT there are few things we have to ensure before the implementation. One of them is to ensure that it is fully automated i.e. without any human interaction. Secondly, it is better to form a nodal based architecture rather than wearable architecture. So, in our implementation each sensor performs as a node and data's are collected from these individual nodes rather than collectively.

The data collection from each node is done through simple Arduino codes. The temperature sensor can measure the temperature of the surrounding. At first, the temperature sensor gives only the room temperature. When the host directly touches the sensor, it will gradually adjust with the host's body temperature and that data is taken to the Arduino. For temperature sensor the data obtained is in Celsius scale. We have connected the pin 5 with arduino which shows the data. So, in the void loop section retrieve that data into a variable "temperature" and then we print that variable in order to see the temperature in the serial monitor [15][16].

In case of pulse sensor, there are 4 different tabs; the main tab we named is PulseSensorAmpedArduino_1.5.0, All serial handling, Interrupt and Timer Interrupt notes. In the main tab we collect the data from the analog zero or A0 pin. Whenever a heartbeat is discovered in the loop of the first tab [17], then in the Serial handling tab, beats per minute (BPM) and inter beat interval (IBI) is calculated. Then the print in the serial monitor is done for all the three different values. The visual serial plotting is then done on the second tab and we can see three different graphs for in the serial plotter [18]. The interrupt handles all the interrupt related issues which differs in different arduino board. We selected the calculation necessary for Arduino Uno. Timer notes give the detail idea of how interrupt works in arduino.

In case of sonar sensor, there are two speakers on the sonar sensors, these two speakers' works individually. One of the speakers send ultrasonic sound wave directly forward whenever the TRIG pin of the sensor receives data to send sound wave. This sound wave will be reflected by any obstacle in front and return back to the sensor, now the sensor will receive this wave with the second speaker. This wave data is put in an equation to determine the distance between the sonar sensor and the obstacle. [19]

In case of pressure pad we take the data from pin 7 and we have kept a count where whenever we get a push in the pad will generate a 1 and no pressure will be indicated as 0. In the loop section, we gave the above condition to check for a continuous amount of period. [20]

3.7.2. Sending data to Thinkspeak

In order to send data to Thinkspeak storage, we need the help of esp-8266. With the data we have in our disposal. The only thing need is to merge the esp-8266 code with every sensor arduino code will help in the sending the data to Thinkspeak. There are some common features of Thinkspeak that we need to ensure during the merging of the code. First, esp-8266 has a different set of language of its own. So, in order to send those data's we need to follow the language properly; secondly, every code will need the SSID and password to connect to the network. It will need the IP for the Thinkspeak website. Thirdly, it will need the channel ID and field number of that ID. And lastly, whenever we reading or writing data on the field each channel has its own unique data reading and writing number. These are the mandatory things to be integrated in the each sensors arduino code. [21]

Basic Codes of Esp-8266

AT+CWMODE: Setting WiFi mode.

AT+CWJAP: Connect to local router.

AT+CIFSR: Query for device IP.

AT+CIPSTART: ESP-8266 connecting to the required server as a client.

AT+CIPSEND: Sending data.

AT+CIPCLOSE: close of data transmission.

a) Modified pseudo code for temperature sensor

Initialize pins

Initialize SSID and Password

Initialize IP for connecting desired website.

Initialize the API key

Setup Method

Initialize Serial Baud

If

Esp8266 connects to wifi

Returns ok

Loop method

Sensor port request for Temperature data

Use update temperature method.

Connect WiFi method

Request connection from Esp8266 to WiFi network

If connected

Print ok

Return true

Else

Return false

Update Temperature method

Check connection.

If connected not found

Return error

Use IP and Intialize port

Start sending data

Resend.

b) Modified pseudo code for pulse sensor

Initialize pins

Initialize SSID and Password

Initialize IP for connecting desired website.

Initialize the API key

Declare Variable

Setup Method

Initialize Serial Baud

If

Esp8266 connects to wifi

Returns ok

use interrupt Setup method

Loop method

Use update beat method**.**

Print done

Connect WiFi method

Request connection from Esp8266 to WiFi network

If connected

Print ok

Return true

Else

Return false

Update beat method

Check connection.

If connected not found

Return error

Use IP and Initialize port

Start sending data

Resend.

Interrupt setup method

Enable interrupts

Read pulse signal

Calculate IBI

Compare IBI with signal

Calculate BPM

Reset variables

c) Modified pseudo code for ultrasonic sonar sensor

Initialize pins

Initialize SSID and Password

Initialize IP for connecting desired website.

Initialize the API key

Setup Method

Initialize Serial Baud

Initialize LED

If

Esp8266 connects to wifi

Returns ok

Loop method

clear trigpin

enable trigpin //generate ultrasonic waveform

calculate duration //from echopin

calculate distance

if distance is less or equals 20

turn on LED

turn count HIGH

else

turn off LED

turn count LOW

print distance

Use update distance method.

Connect WiFi method

Request connection from Esp8266 to WiFi network

If connected

Print ok

Return true

Else

Return false

Update Distance method

Check connection.

If connected not found

Return error

Use IP and Intialize port // Usually port used is 80

Initialize field of channel

Start sending data //CIPCLOSE

Resend

d) Modified pseudo code for pressure pad

Initialize pins

Initialize SSID and Password

Initialize IP for connecting desired website.

Initialize the API key

Declare Variable

Setup Method

Initialize Serial Baud //preferable 115200

If

Esp8266 connects to wifi   //using connect WiFi method

Returns ok

Loop method

read button state

if buttonState is HIGH

turn LED on:

set button variable high

else

turn LED off:

set button variable low

Use sleepOn method.

Print done

Connect WiFi method

Request connection from Esp8266 to WiFi network

If connected

Print ok

Return true

Else

Return false

SleepOn method

Check connection.

If connected not found

Return error

Use IP and Intialize port

Start sending data

Resend.

MatLab duration pseudo code

Declare channel ID

Initialize arrays

For all values of the array Do

Calculate duration between two intervals

Return duration

MatLab min,max,average calculation pseudo code

Declare channel ID

Initialize arrays

For all values of the array Do

Calculate maximum, minimum and average

Return values.

Data Retrieval

We can retrieve raw data from Thinkspeak using an option provided in it called the Export/Import Data. The updated data of the entire field in the corresponding channel can be downloaded. The data which we can retrieve are provided in the .CSV format.

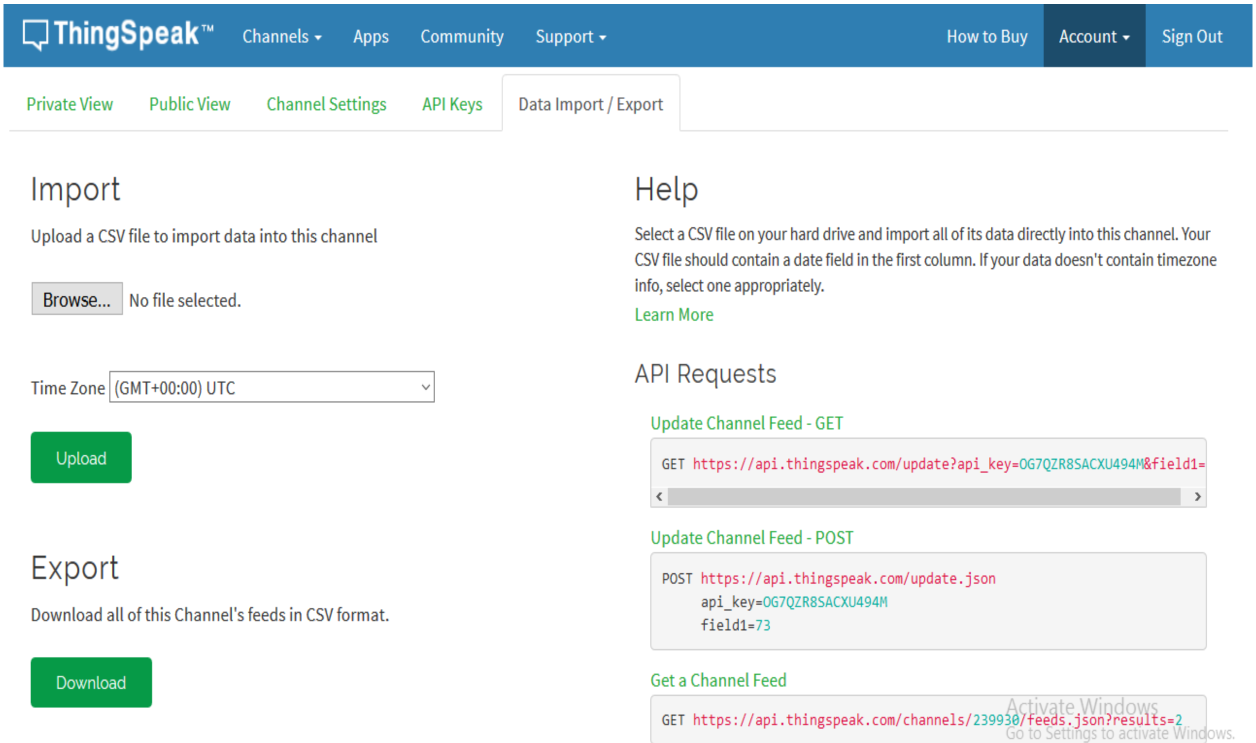


Fig 12: Retrieving data from Thinkspeak

Chapter-4

Result and Discussion

In this segment after implementing the code we will discuss how it is displayed in the Thinkspeak account of a user and how reaction or feedback is sent on the email and twitter. The data's we obtained after transmitting is saved in the cloud of Thinkspeak server and from there with the help the features provided from Thinkspeak we are able to display data and send alarming message.

4.1 Result Display

In the Thinkspeak server it takes 15 second delays for each data entry. The data entered in the storage is then graphically portrayed in the display. The data entered in the storage is channel and field specific. That means it will go the specific field of that channel that is given by the user. For, convenience of the doctor of the elderly we opened a separate channel for them. The patient who is under observation of that doctor is then placed on different fields of that doctor's channel. The following figure shows it:

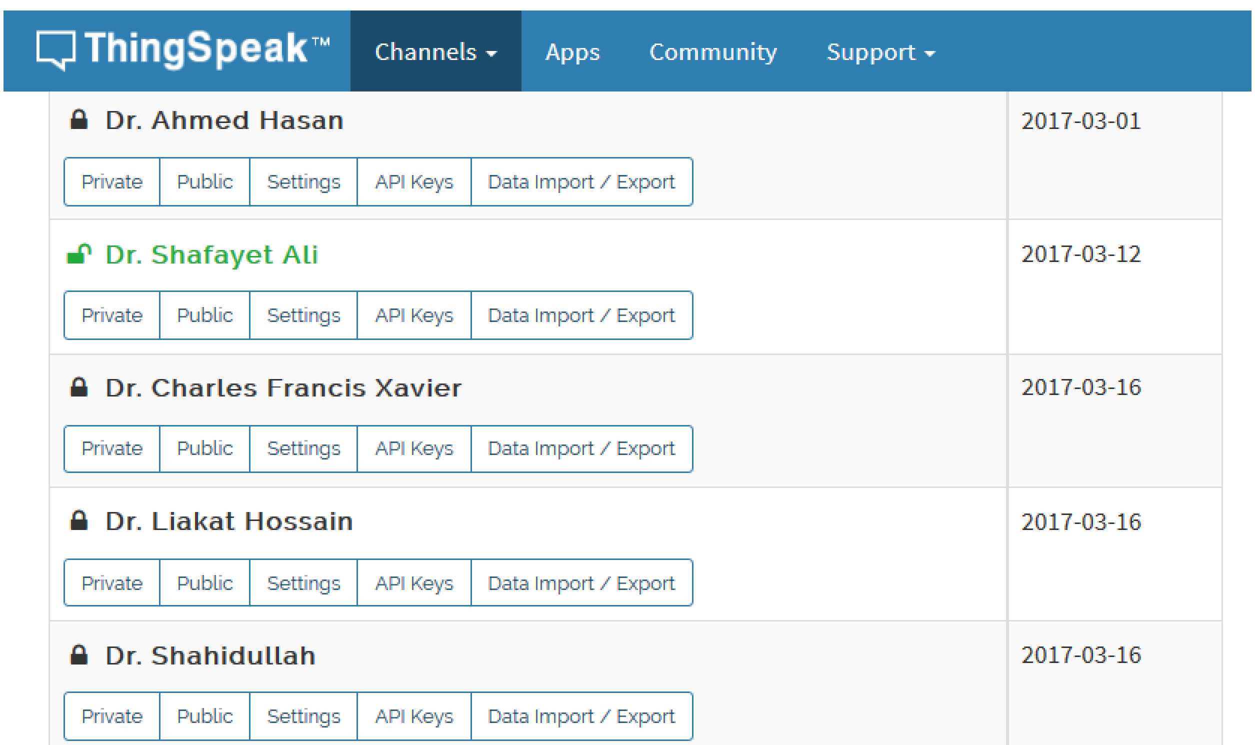


Fig13: Different Channels for different doctors

We gave multiple doctors in this case for multiple hosts. And the display is absolutely privacy protected. That is, the user can keep the channel open for public viewing or it can be made private viewing for convenience.

After selecting the specified channel the user can see the following field of the host's daily updated graphs for monitoring.

4.1.1 Temperature graph

As we can that temperature of a random is shown below. Here the peak value which was taken at an approximate time 2:40-2:50 PM GMT+6. The value shown here is seen to be slowly getting saturated with the body temperature and showing the peak value at about 98 degree Fahrenheit.

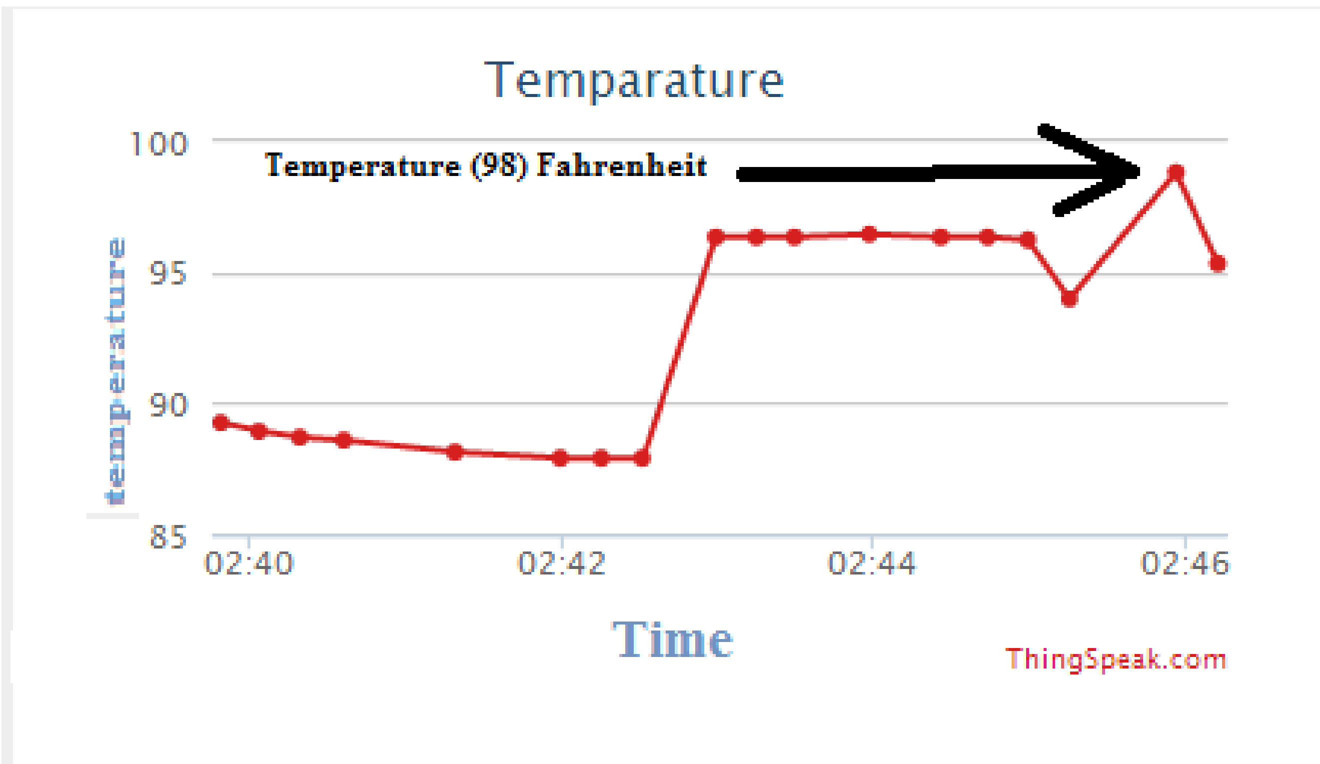


Fig 14: Temperature graph of a host

The temperature we send is in degree Celsius scale, which we convert using MatLab into degree Fahrenheit scale. So, our retrieved data for temperature remains in Celsius scale. The following figure shows the increase of temperature with a human touch and again goes back to room temperature when touch is removed along with the time of entry.

	A	C
1	created_at	Temperature
2	2017-03-01 04:36:35 UTC	96.5
3	2017-03-01 04:37:10 UTC	97.5
4	2017-03-01 04:37:43 UTC	87
5	2017-03-01 04:38:17 UTC	86
6	2017-03-01 04:38:52 UTC	86
7	2017-03-01 04:41:56 UTC	86
8	2017-03-01 04:42:30 UTC	86
9	2017-03-01 04:43:04 UTC	87
10	2017-03-01 04:43:38 UTC	93
11	2017-03-01 04:44:12 UTC	95.5
12	2017-03-01 04:44:47 UTC	96
13	2017-03-01 04:45:23 UTC	96
14	2017-03-01 04:45:55 UTC	96
15	2017-03-01 07:07:19 UTC	96
16	2017-03-01 07:07:48 UTC	96
17	2017-03-01 07:08:22 UTC	96
18	2017-03-01 07:08:58 UTC	96
19	2017-03-01 07:09:30 UTC	96
20	2017-03-01 07:10:04 UTC	96
21	2017-03-01 07:10:38 UTC	97
22	2017-03-01 07:12:25 UTC	98
23	2017-03-01 07:12:57 UTC	98.5
24	2017-03-01 07:13:31 UTC	98
25	2017-03-01 07:14:05 UTC	95.5

Fig 15: The retrieved temperature in Celsius scale

4.1.2 Pulse graph

The pulse graph which is taken at first gives three different values of parameters, which are already discussed above. The first figure shows the three different column, the first column being the pulse rate, the second being the IBI and the third being the pulse signal.

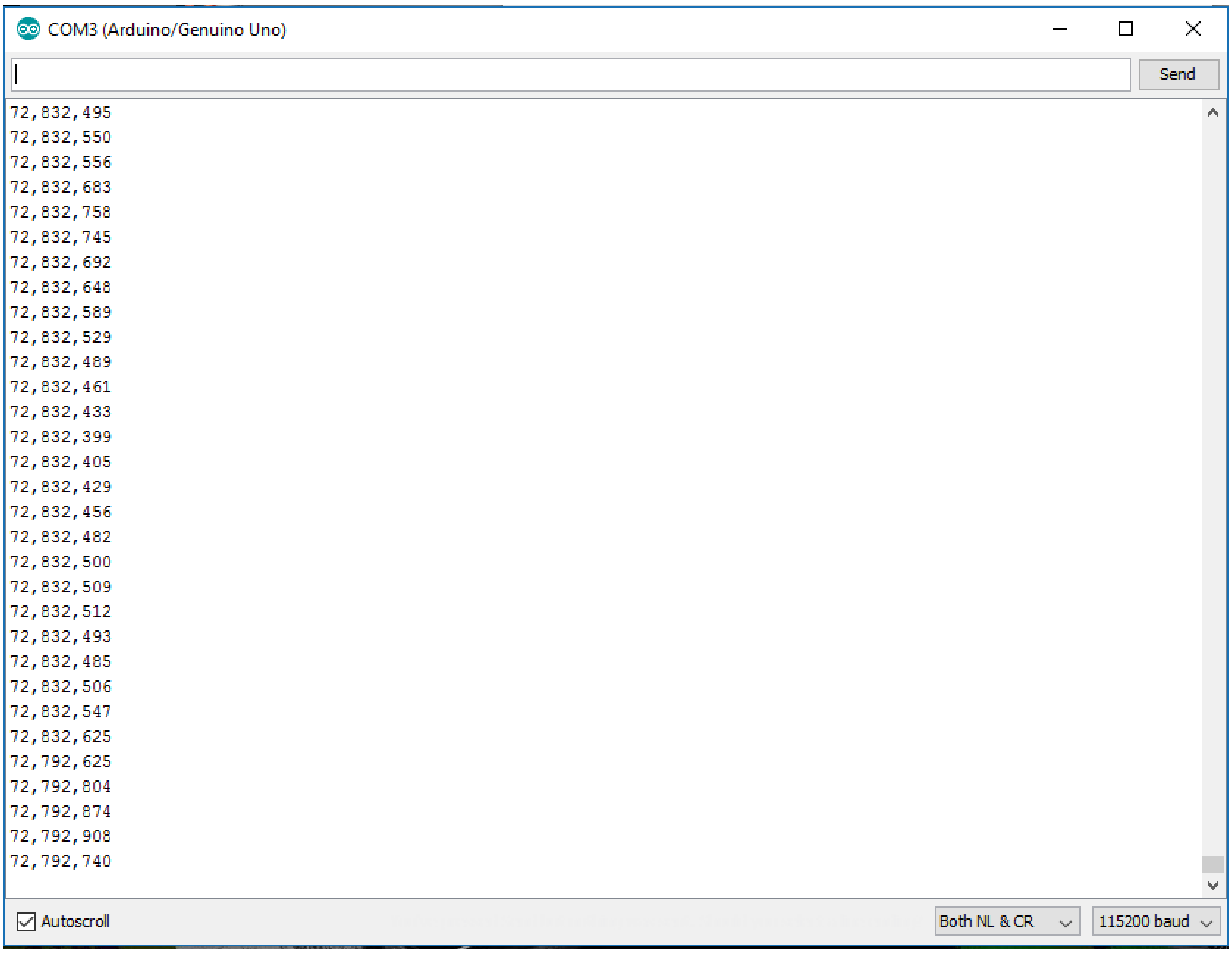


Fig 16: The initial values of from the serial monitor of arduino.

In the serial plotter of arduino the following graph is obtained. The blue colored indicates the pulse rate, the red color shows the Inter beat Interval (IBI) and the green graph shows the pulse signal.

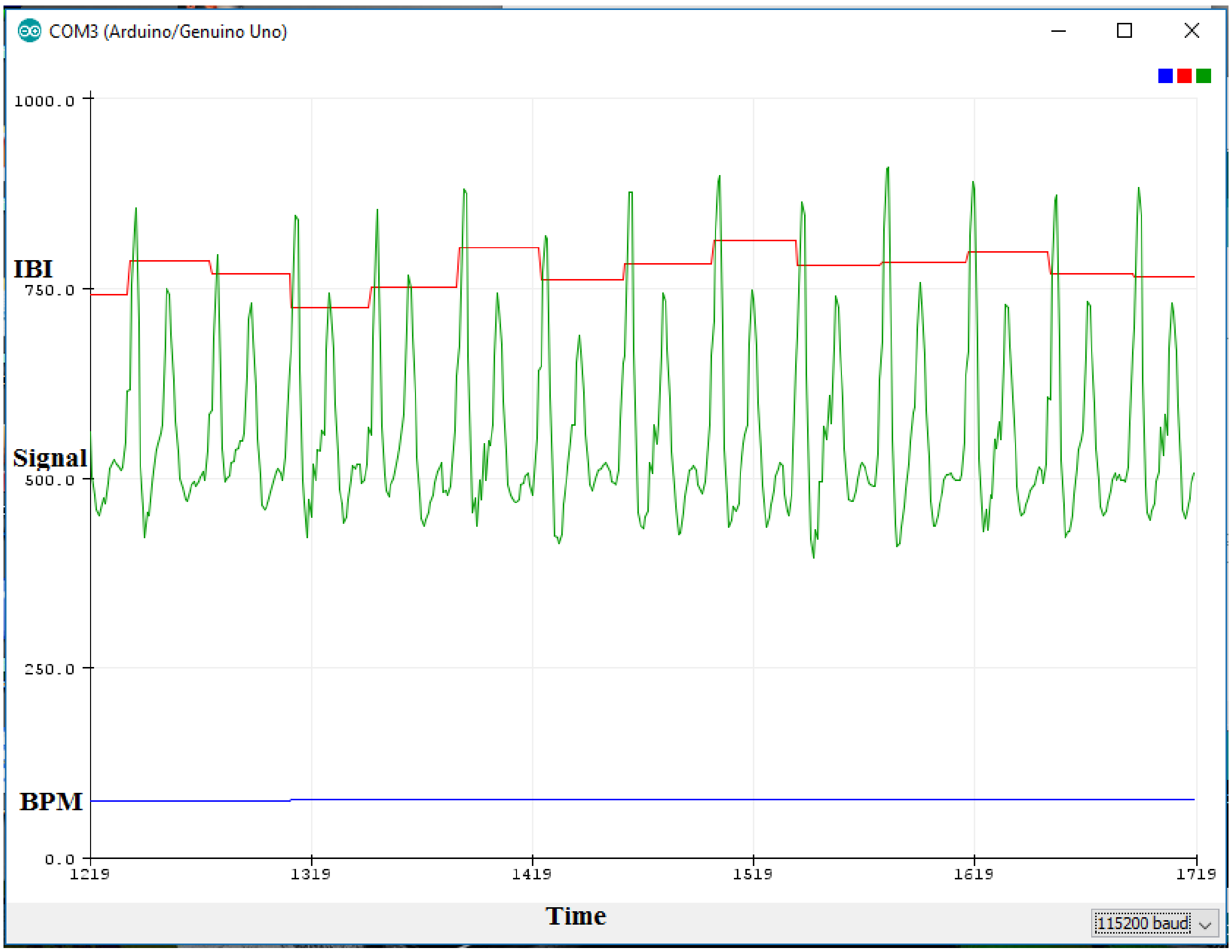


Fig 17: Serial plotter view of the data received from pulse sensor.

From this data set for the convenience of this project we only selected the pulse rate data as a criterion for monitoring. After implementing the ESP-8266 code with this pulse sensor data, it is formed in the Thinkspeak as following graph of a host. Initially there are some values which are a little high for noise factor. But, with time normal pulse starts to appear as we can see in the following graph.

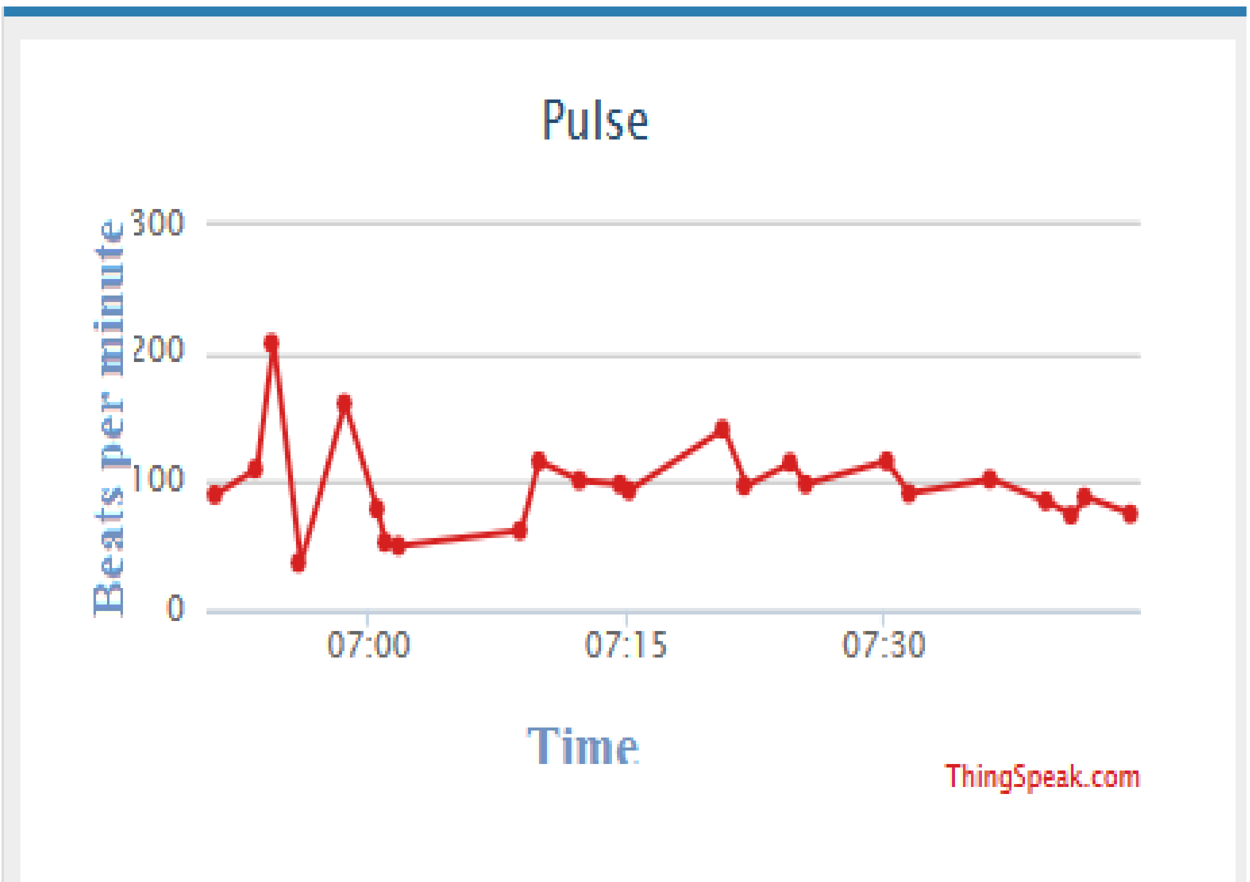


Fig 18: The Thinkspeak graph of a host corresponding to the plotter graph

We can see the corresponding raw data value that we retrieved from the Thinkspeak server in a .CSV format. Initially, the pulse rate is unusually high for high noise margin but with time the noise margin stabilizes.

	A	D
63	2017-03-05 01:48:11 UTC	98
64	2017-03-05 01:48:45 UTC	105
65	2017-03-05 01:49:03 UTC	104
66	2017-03-05 01:49:29 UTC	98
67	2017-03-05 01:50:13 UTC	94
68	2017-03-05 01:50:37 UTC	102
69	2017-03-05 01:50:55 UTC	94
70	2017-03-05 01:51:49 UTC	97
71	2017-03-05 01:52:33 UTC	99
72	2017-03-05 01:52:57 UTC	92
73	2017-03-05 01:53:31 UTC	102
74	2017-03-05 01:53:59 UTC	100
75	2017-03-05 01:54:47 UTC	100
76	2017-03-05 01:55:15 UTC	99
77	2017-03-05 01:55:39 UTC	106
78	2017-03-05 01:55:57 UTC	100
79	2017-03-05 01:56:45 UTC	103
80	2017-03-05 01:57:10 UTC	100
81	2017-03-05 01:57:28 UTC	103
82	2017-03-05 01:57:52 UTC	99
83	2017-03-05 01:58:26 UTC	101
84	2017-03-05 01:58:50 UTC	97
85	2017-03-05 01:59:54 UTC	95
86	2017-03-05 02:00:28 UTC	97
87	2017-03-05 02:00:52 UTC	92

Fig 19: The retrieved BPM data along with time duration

4.1.3 Sonar graph

The sonar sensor graph shown below is valued between 0to1. One indicating the number of times the host used toilet against the date.

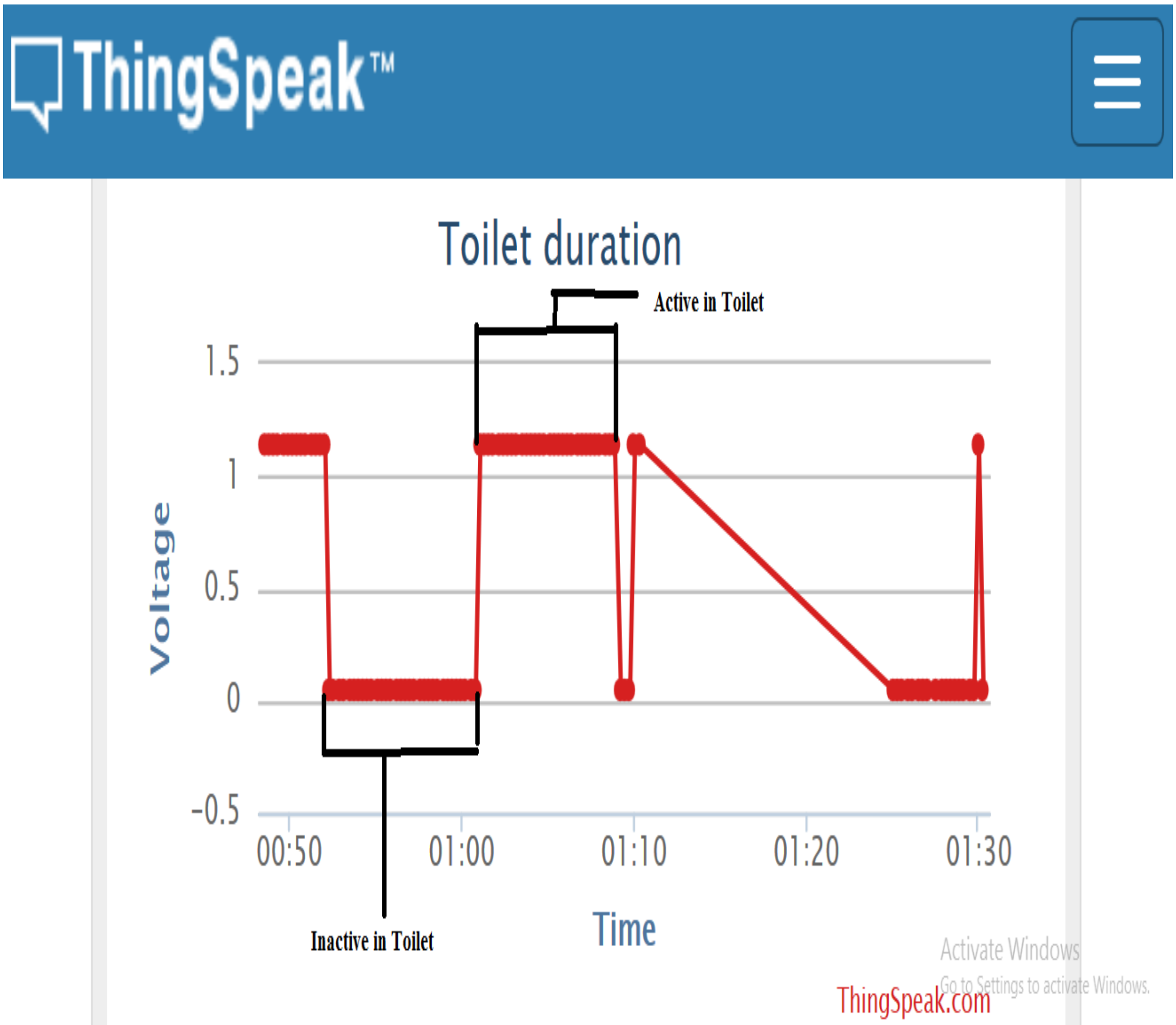
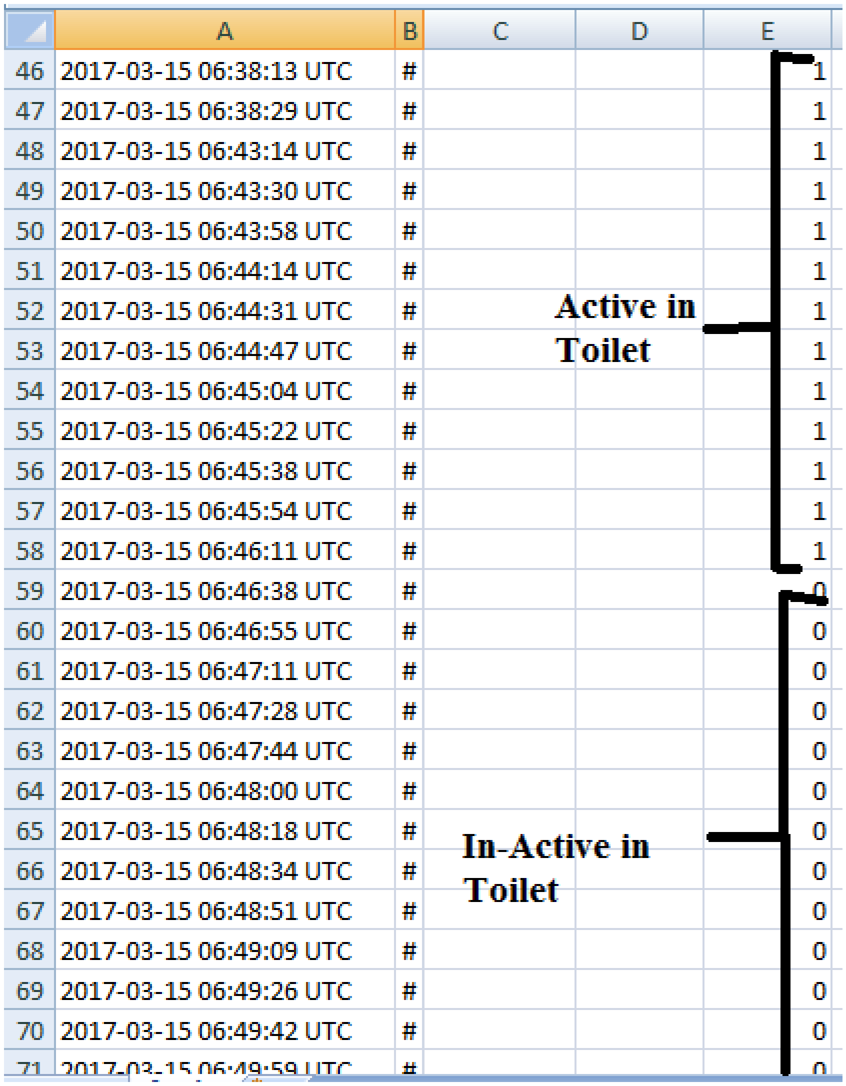


Fig 20: Graph used to indicate the duration of toilet

The following figure is the corresponding data retrieved of the above graph from Thinkspeak in .CSV format. The active and in-active session are marked in the following figure along with date and time.



	A	B	C	D	E
46	2017-03-15 06:38:13 UTC	#			1
47	2017-03-15 06:38:29 UTC	#			1
48	2017-03-15 06:43:14 UTC	#			1
49	2017-03-15 06:43:30 UTC	#			1
50	2017-03-15 06:43:58 UTC	#			1
51	2017-03-15 06:44:14 UTC	#			1
52	2017-03-15 06:44:31 UTC	#			1
53	2017-03-15 06:44:47 UTC	#			1
54	2017-03-15 06:45:04 UTC	#			1
55	2017-03-15 06:45:22 UTC	#			1
56	2017-03-15 06:45:38 UTC	#			1
57	2017-03-15 06:45:54 UTC	#			1
58	2017-03-15 06:46:11 UTC	#			1
59	2017-03-15 06:46:38 UTC	#			0
60	2017-03-15 06:46:55 UTC	#			0
61	2017-03-15 06:47:11 UTC	#			0
62	2017-03-15 06:47:28 UTC	#			0
63	2017-03-15 06:47:44 UTC	#			0
64	2017-03-15 06:48:00 UTC	#			0
65	2017-03-15 06:48:18 UTC	#			0
66	2017-03-15 06:48:34 UTC	#			0
67	2017-03-15 06:48:51 UTC	#			0
68	2017-03-15 06:49:09 UTC	#			0
69	2017-03-15 06:49:26 UTC	#			0
70	2017-03-15 06:49:42 UTC	#			0
71	2017-03-15 06:49:59 UTC	#			0

Fig 21: The retrieved activity duration data of Toilet

4.1.4 Sleeping time graph

The sleeping time is also similar as the sonar graph. One indicating that the person is sleep zone and 0 indicating the person is in non-sleeping zone i.e. awake.

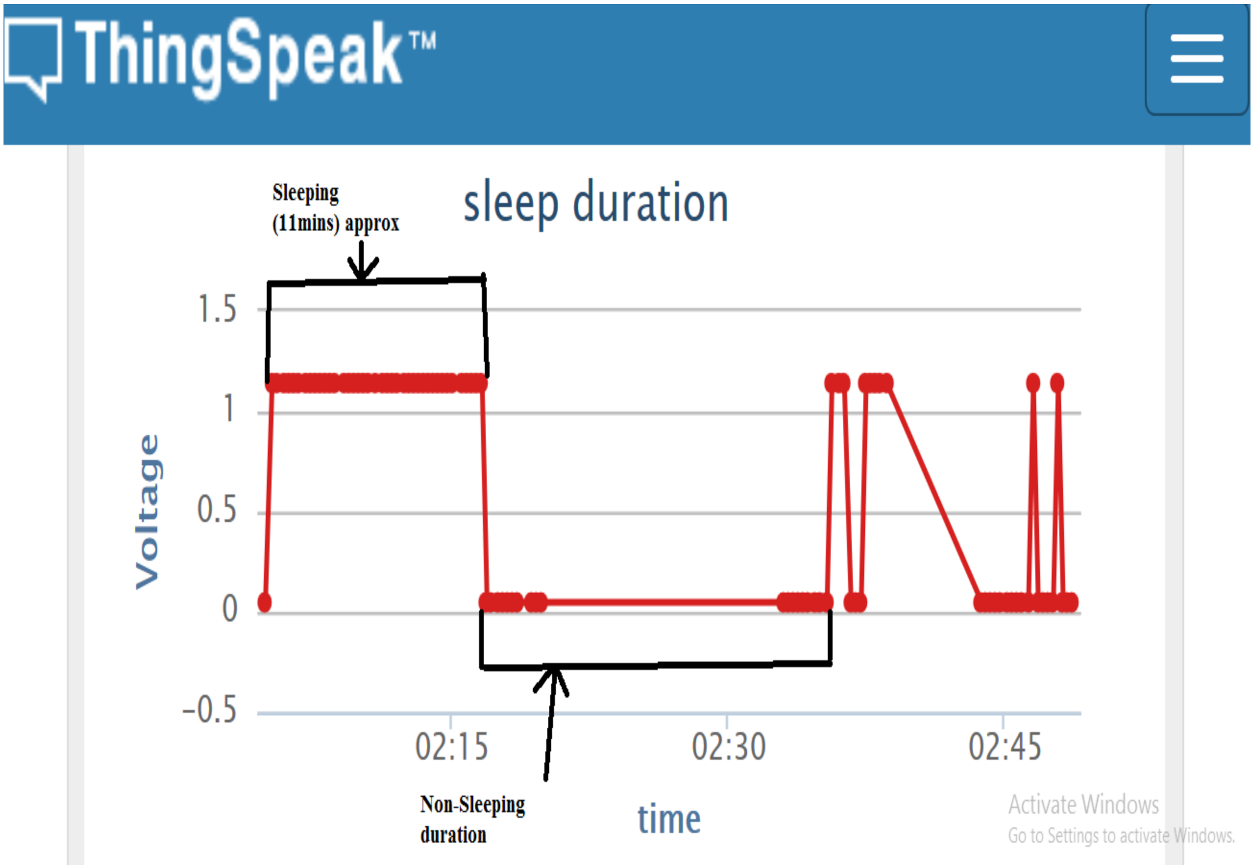


Fig 22: Sleep time of a random host

Corresponding to the graph which indicates the time duration of sleep, using MatLab provided by Thinkspeak we can measure the time actual duration of sleeping time.

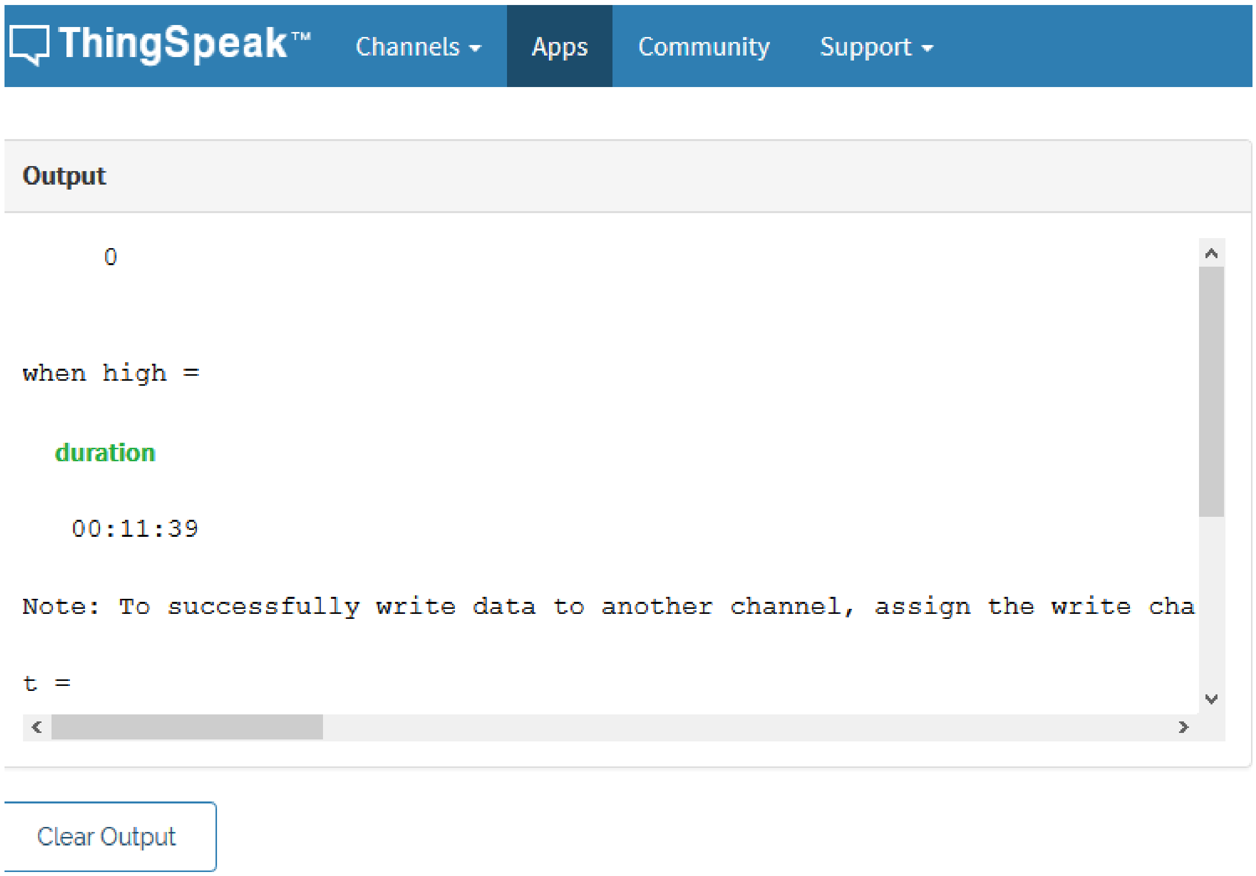
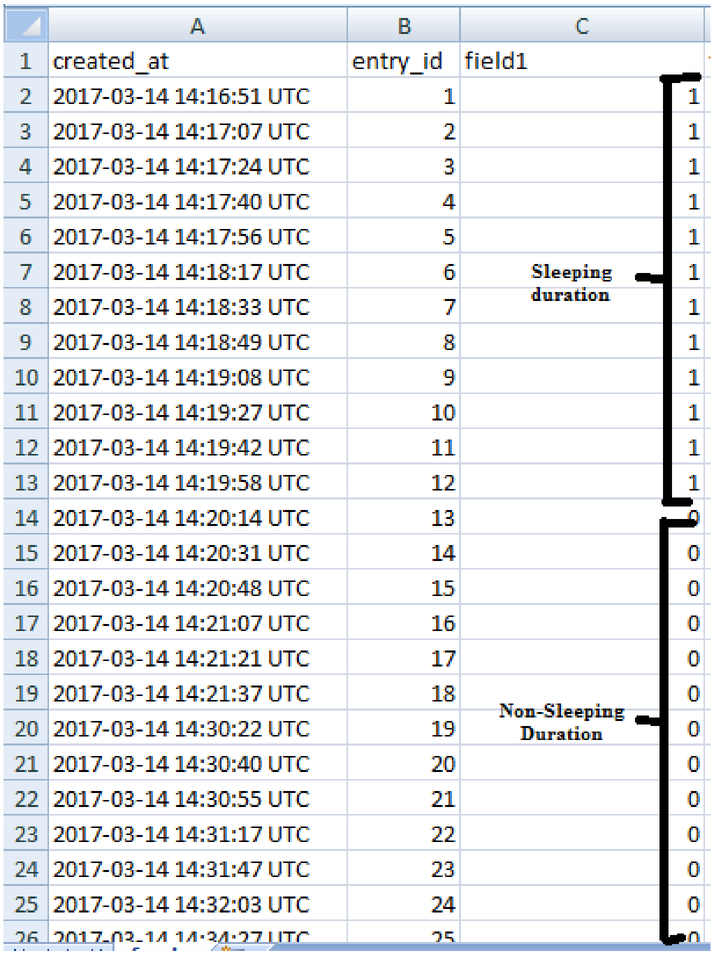


Fig 23: Duration Calculation of sleeping duration using MatLab

And from the retrieved data from obtained in .CSV file we can also see the resemblance and time duration very clearly. The 1 indicates sleeping duration and 0 indicating non-sleeping duration. The time of entry of the data is also provided along with the data.



	A	B	C
1	created_at	entry_id	field1
2	2017-03-14 14:16:51 UTC	1	1
3	2017-03-14 14:17:07 UTC	2	1
4	2017-03-14 14:17:24 UTC	3	1
5	2017-03-14 14:17:40 UTC	4	1
6	2017-03-14 14:17:56 UTC	5	1
7	2017-03-14 14:18:17 UTC	6	1
8	2017-03-14 14:18:33 UTC	7	1
9	2017-03-14 14:18:49 UTC	8	1
10	2017-03-14 14:19:08 UTC	9	1
11	2017-03-14 14:19:27 UTC	10	1
12	2017-03-14 14:19:42 UTC	11	1
13	2017-03-14 14:19:58 UTC	12	1
14	2017-03-14 14:20:14 UTC	13	0
15	2017-03-14 14:20:31 UTC	14	0
16	2017-03-14 14:20:48 UTC	15	0
17	2017-03-14 14:21:07 UTC	16	0
18	2017-03-14 14:21:21 UTC	17	0
19	2017-03-14 14:21:37 UTC	18	0
20	2017-03-14 14:30:22 UTC	19	0
21	2017-03-14 14:30:40 UTC	20	0
22	2017-03-14 14:30:55 UTC	21	0
23	2017-03-14 14:31:17 UTC	22	0
24	2017-03-14 14:31:47 UTC	23	0
25	2017-03-14 14:32:03 UTC	24	0
26	2017-03-14 14:34:27 UTC	25	0

Fig 24: Retrieved data of sleep duration in .CSV file

4.2 Reaction display

The reactions are displayed through the e-mail and social networking site Twitter. In Thinkspeak we have a privilege of using react and ThinkHTTP option. The react option works on channel and field specific. It requires the reference data of the host, whenever the reference data is crossed a tweet or message on e-mail will be sent. Tweet sending is relatively easy with react option of Thinkspeak but in order to send an e-mail a third party website Push Inbox is used for out project. ThinkHTTP helps to connect the react panel with the Push Inbox site. And then the message is then sent to the desired e-mail address through Push Inbox site.

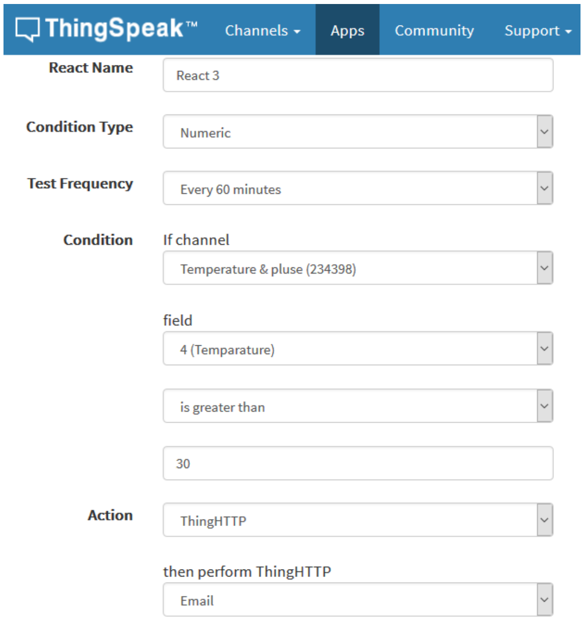


Fig 25: React for e-mail alarm send toThinkHTTP

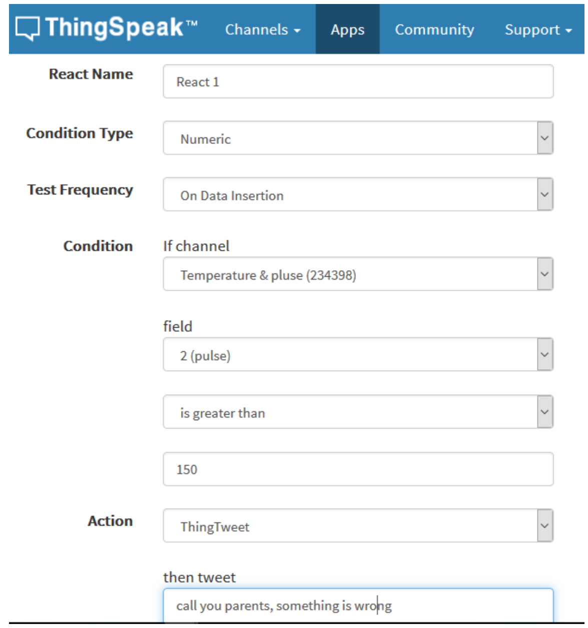
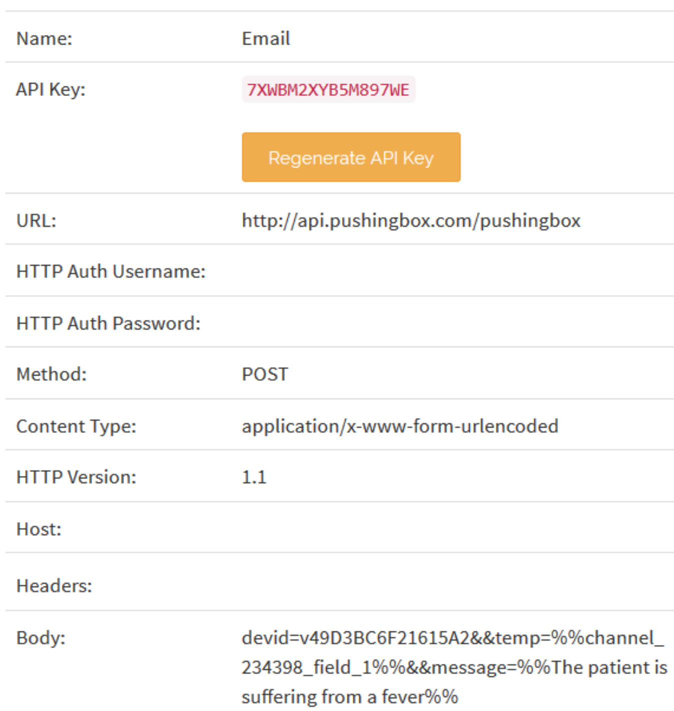


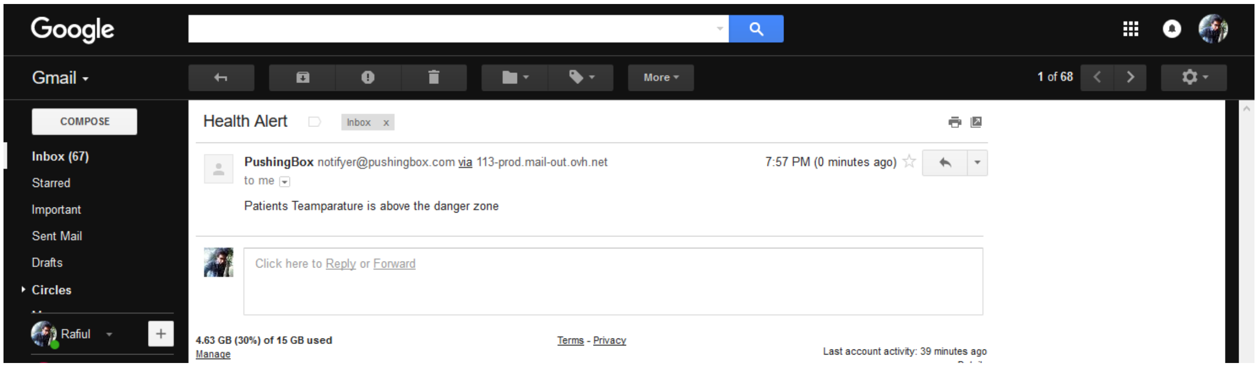
Fig 26: React for Tweet alarm



Name:	Email
API Key:	7XWBM2XYB5M897WE
	Regenerate API Key
URL:	http://api.pushingbox.com/pushingbox
HTTP Auth Username:	
HTTP Auth Password:	
Method:	POST
Content Type:	application/x-www-form-urlencoded
HTTP Version:	1.1
Host:	
Headers:	
Body:	devid=v49D3BC6F21615A2&&temp=%%channel_234398_field_1%%&&message=%%The patient is suffering from a fever%%

Fig 27: Message sending from ThinkHTTP to Push Inbox

Then the following figures shows our message sent to the concerning people related to the host. At first in the e-mail.



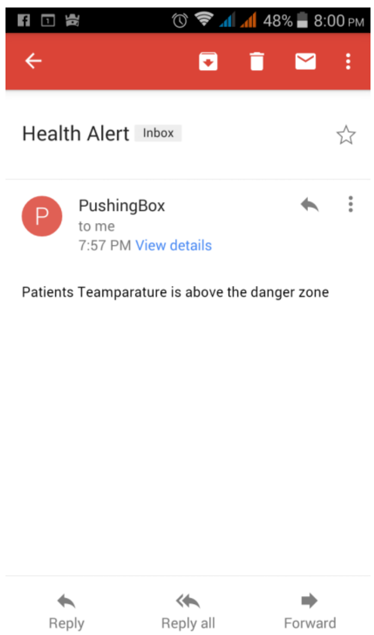


Fig 28: E-mail alarm message received from PushInbox

Secondly, the tweet message is shown below.

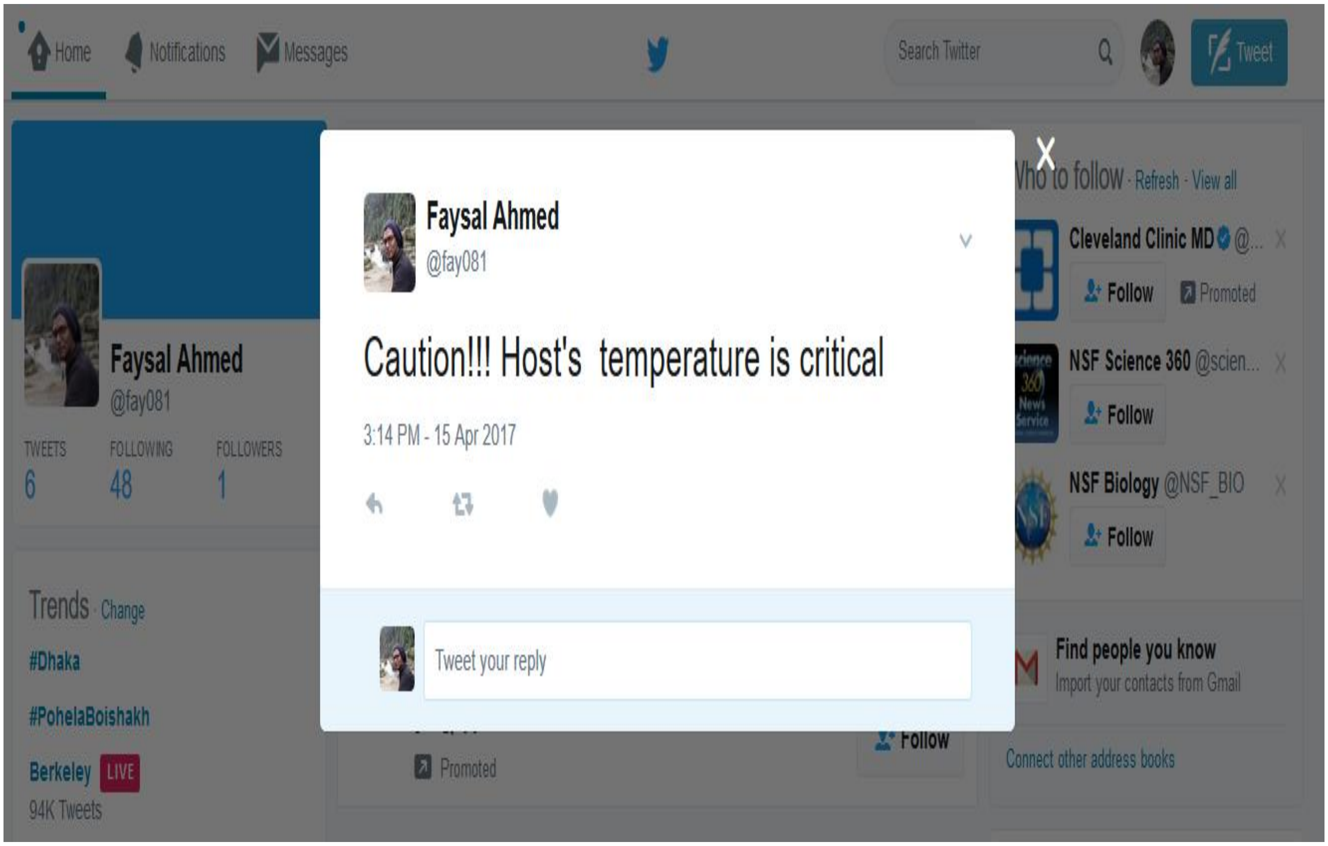


Fig 29: Tweet alarm message

4.3 Cost Analysis

The only cost that was needed for our project was the hardware cost, on the other had any other project that was conducted was needed more costly hardware with more limited features, like the **Masimo Radical-7** and **Free Scale Home Health Hub reference platform.** So, it is safe to say that our project is more cost effective, with more features.

Chapter-5
Conclusion

Our main objective in this project was to successfully monitor the basic four criteria's namely temperature, pulse, using of toilet and sleep and react during emergency situation without any human interaction. We wanted to make a mark on the field of IoT with the health sector. With the rise of IoT, the era of technology is moving towards a far superior dimension. In order to keep pace with the new technologies, this project can sure make way for the advancement in this sector. Though our model is tested and implemented, it will be difficult to continue the project without superior quality hardware support along with a lot of new integration. The real benefit of this work can only be fully realized when it can be implemented in full scale.

4.4 Challenges

There are three basic challenges we face during the project implementation.

ESP-8266 WiFi Module

The wifi module works in its own specific language. So, we had to face a lot problem regarding the ways of its language. Sometimes the wifi module itself cannot with the local network as a result the data sending got interrupted many a times. So, a better hardware support for wifi module is expected to send data smoothly.

Pulse Sensor

The data that is retrieved from the pulse sensor can give some error reading sometimes. Due to unavailability of better pulse sensor in our country we had to order it from USA. And the data comparison of the local and foreign sensor was noticeable. This delayed our implementation of the project in some ways as pulse sensor is an integral part of this project.

Data Analysis and React

The data analysis with MATLAB of Thinkspeak and react of the Thinkspeak gave us some challenges. Especially with alarm message, it was difficult to link with the Push Inbox and with twitter.

4.5 Future works

- Integrating more sensors for more specific data acquisition and analysis.
- Will be applicable in army services in active situation.
- Will be used to provide health service to rural areas in affordable price.
- Huge database will be built for doctors to diagnose people from different areas and cultures.

Our project can be considered as platform to develop in the field of IoT on the health sector. In developing countries like ours, this kind of innovative and cost effective project can improve the future of technology. So, we are looking forward to implement the project in order to make an impact in the new era of technology.

References

[1] Brock, D. L. (2001). The electronic product code (epc). *Auto-ID Center White Paper MIT-AUTOID-WH-002*.

[2] Ni, Q., García Hernando, A. B., & de la Cruz, I. P. (2015). The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors*, *15*(5), 11312-11362

[3] Ray, P. P. (2014, November). Home Health Hub Internet of Things (H 3 IoT): an architectural framework for monitoring health of elderly people. In *Science Engineering and Management Research (ICSEMR), 2014 International Conference on* (pp. 1-3). IEEE.

[4] Niewolny, D. (2013). How the internet of things is revolutionizing healthcare. *White paper*.

[5] Patel, S., Park, H., Bonato, P., Chan, L., & Rodgers, M. (2012). A review of wearable sensors and systems with application in rehabilitation. *Journal of neuroengineering and rehabilitation*, *9*(1), 21

[6] de Battista, N., Rice, J. A., Sim, S. H., Brownjohn, J. M. W., & Tan, H. P. STRUCTURAL HEALTH MONITORING OF CIVIL INFRASTRUCTURE USING WIRELESS SENSOR NETWORKS.

[7] Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., ... & Andreescu, S. (2015, June). Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In *Services Computing (SCC), 2015 IEEE International Conference on* (pp. 285-292). IEEE.

[8] Tyagi, S., Agarwal, A., & Maheshwari, P. (2016). A conceptual framework for IoT-based healthcare system using cloud computing. In *Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference* (pp. 503-507). IEEE.

[9]https://www.google.com/search?q=temperature+sensor+ds18b20&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwik4eD_4aHTAhUIsY8KHaKPAKYQ_AUIBigB&biw=1366&bih=669#imgrc=Iq8FIecPDinPqM:

[10] https://www.google.com/search?q=pulse+senspr+SEN11547&client=firefox-bab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwih8q7D46HTAhVBt48KHdpcABYQ_AUIBigB&biw=1366&bih=669#tbm=isch&q=pulse+sensor+&imgrc=XMR0igm9idepJM:

[11] https://www.google.com/search?q=ultrasonic+sonar+sensor&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwj5jPWD5KHTAhWIrY8KHeLkAKIQ_AUICCgB&biw=1252&bih=613&dpr=1.09#tbm=isch&q=ultrasonic+sonar+sensor+hc-sr04&imgrc=YAhJUyfVRQZKiM:

[12]https://www.google.com/search?q=pressure+pad&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiYv8DG5KHTAhXJrY8KHdWyDGYQ_AUICCgB&biw=1252&bih=613#tbm=isch&q=home+made+pressure+pad&imgrc=djV7UQ_Jtl-18M:

[13]https://www.google.com/search?q=arduino+uno&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwip2qiR5aHTAhUJQo8KHX-9AJAQ_AUICCgB#imgrc=YWmB1xWbeYqNrM:

[14]https://www.google.com/search?q=Esp8266&client=firefox-b-ab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwj5zZTR5aHTAhVEKY8KHcq0BEUQ_AUICCgB&biw=1252&bih=613#imgrc=Z2zdlsmWCMB6rM:

[15]https://create.arduino.cc/projecthub/everth-villamil-ruiz/temperature-sensor-ds18b20-3decfc

[16] https://create.arduino.cc/projecthub/everth-villamil-ruiz/temperature-sensor-ds18b20-3decfc

[17] https://pulsesensor.com/pages/code-and-guide

[18] https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino

[19] http://playground.arduino.cc/Main/UltrasonicSensor

[20]http://www.instructables.com/id/Use-a-DIY-Pressure-Plate-Switch-to-Automate-Your-H/

[21]https://circuitdigest.com/microcontroller-projects/sending-arduino-data-to-webpage