*A Project Report*

*on*

**Healthcare Monitoring System**

*Carried out as a part of the course Minor Project Submitted by*

***Mridul Jajodia***

***189303187***

***VI SEM B.Tech CCE***

*In partial fulfilment for the award of the degree*

*Of*

**Bachelors in Technology**

In

**Computer and Communications Engineering**



**Department of Computer & Communication Engineering,**

**School of Computing and IT,**

**Manipal University Jaipur,**

***June, 2021***

**CERTIFICATE**

This is to certify that the project entitled "**Healthcare monitoring system**" is a Bonafede work carried out as part of the course ***Minor Project*** , under my guidance by ***Mridul Jajodia,***  student of ***B.Tech VI SEM*** at the Department of Computer & Communication Engineering , Manipal University Jaipur, during the academic semester ***VI*** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer & Communication Engineering, at MUJ, Jaipur.

Place: Jaipur Dr. Vaishali Yadav

Date: 15/06/21 Signature of the Instructor (s)

**DECLARATION**

I hereby declare that the project entitled **“Healthcare monitoring system”** submitted as part of the partial course requirements for the course ***Minor project*** for the award of the degree of Bachelor of Technology in Computer & Communication Engineering at Manipal University Jaipur during the VI semester, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

Signature of the Student: Mridul Jajodia

Place: Jaipur

Date: 15/06/21

**ABSTRACT**

Healthcare is given the extreme importance now a- days by each country with the advent of the novel corona virus. So, in this aspect, an IoT based health monitoring system is the best solution for such an epidemic. Internet of Things (IoT) is the new revolution of internet which is the growing research area especially in the health care. With the increase in use of wearable sensors and the smart phones, these remote health care monitoring has evolved in such a pace. IoT monitoring of health helps in preventing the spread of disease as well as to get a proper diagnosis of the state of health, even if the doctor is at far distance. In this paper, a portable physiological checking framework is displayed, which can constantly screen the patient’s heartbeat, temperature and other basic parameters of the room. We proposed a nonstop checking and control instrument to screen the patient condition and store the patient information’s in server utilizing Wi-Fi Module based remote correspondence. A remote health monitoring system using IoT is proposed where the authorized personal can access these data stored using any IoT platform and based on these values received, the diseases are diagnosed by the doctors from a distance.

Keywords: Internet of Things, Health, Sensors.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **CONTENT** | **Page No.** |
| 1. | Introduction | 6 |
| 2. | Related work | 8 |
| 3. | Requirement analysis | 9 |
| 4. | Work Done | 18 |
| 5. | Conclusion | 22 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Particulars** | **Page No.** |
| 3.3.2.1 | LM35 | 14 |
| 3.3.2.2 | Pulse sensor | 15 |
| 3.3.2.3 | MQ135 | 15 |
| 3.3.2 | Final system design | 16 |
| 4.1.1 | Temperature output | 20 |
| 4.2.1 | SMS results | 21 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table** | **Particulars** | **Page No.** |
| 3.1 | Bolt module details | 10 |
| 3.2.2.1 | LM35 specifications | 15 |
| 3.2.2.2 | Pulse sensor specifications | 16 |
| 3.2.2.3 | MQ135 specifications | 16 |

1. **INTRODUCTION**
   1. **SCOPE OF WORK**

There are several fatal diseases like heart disease, diabetes, breast cancer, liver disorder, etc. in medical sector but the main concern of our developed system is to monitor the fundamental signs of all types of patients and the patient’s room environment. This model proposes a customized healthcare system that monitors the pulse and body temperature of patients as well as room humidity, CO, and CO2 gas level of patient’s room via sensors and transmits the data through Wi-Fi that enables the medical staffs to get data from the server. The data collected is also displayed in the form of a graph. Here I will be using the bolt iot and cloud platform as it is Wi-Fi enabled and provides everything.

* 1. **PRODUCT SCENARIOS**

Heart rate and body temperature are the two most significant indicators for human health. Heart rate is the per-minute number of heartbeats, commonly known as the pulse rate. To measure the pulse rate, an increase in the blood flow volume can be used by calculating the pulses. Normal heart rate ranges between 60 and 100 beats per minute for healthy people. The typical restful heart for adult males is roughly 70 bpm and for adult females 75 bpm. Female with 12 years of age and above, typically have higher rates of heart in contrast with males. The temperature of human body is simply the heat of body and the sum of heat radiated by the body is scientifically determined. The average person’s body temperature relies on different factors such as ambient temperature, the person’s gender, and his eating habits. In healthy adults, it is likely to range between 97.8 °F (36.5 °C) and 99 °F (37.2 °C). Different factors such as flu, low-temperature hypothermia, or any other illness may lead to a change in body temperature. In almost all illnesses, fever is a typical indicator. It is suggested that a healthcare should provide good room conditions to facilitate the patients. Some measures like room humidity, level of all gases like CO, and CO2 can determine the quality of room environment. The toxic gases and certain levels of humidity are very harmful to patients. For optimum comfort, the room humidity should be between 30 and 65%.

**2. RELATED WORK**

Some important works have been done in the field of medical science using IoT to monitor patient’s health. The works associated in this field are outlined as follows.

Tamilselvi et al. developed a health monitoring system that can monitor basic symptoms of a patient like heart rate, percentage of oxygen saturation, body temperature, and eye movement in IoT network. For this purpose, the system used Heartbeat, SpO2, Temperature, and Eye blink sensors as capturing elements and Arduino-UNO as a processing device. The developed system was implemented but no specific performance measures are described for any patient. Acharya et al. introduced a healthcare monitoring kit in IoT environment. The developed system monitored some basic parameters of human health like Heartbeat, ECG, body temperature, and Respiration. The major hardware components which are used here are pulse sensor, temperature sensor, BP sensor, ECG sensor, and raspberry pi. The data were collected from sensors and sent it to raspberry pi for processing and again transmitted it to IoT network. The major drawback of the system is that no interfaces for data visualization are developed.

Trivedi et al. suggested a mobile device regulated Arduino-based health parameter surveillance framework. The collected sensor data are analog and sent it to the board of Arduino Uno. By the integrated analog to digital converter, the recorded analog values are converted into digital data. Bluetooth transmitted the physical qualities to the developed device. The Bluetooth device used a module not covering a wide area. Kumar et al. developed an adaptive IoT safety monitoring device. The configuration of the framework is separated into 3 layers such as the control layer, the device layer, and the transport layer. A DS18B20 sensor was used for the measurement of body temperature in the control segment and a pulse sensor is used for the pulse measurement. The data were loaded from Arduino into the cloud through the Wi-Fi module and Ethernet shield on the transport layer. The framework layer finally collected the server details. However, Arduino Uno was used here, and hence, many sensors cannot be treated properly.

All works are mentioned in the references section.

**3. REQUIREMENT ANALYSIS**

* 1. **HARDWARE COMPONENTS:**

**Bolt Iot Wi-Fi Module:**

**General Specifications**

| **Parameters** | **Details** |
| --- | --- |
| Connectivity and Processing Module | ESP8266 with custom firmware |
| MCU | 32-bit RISC CPU: Tensilica Xtensa LX106 |
| Power | 5V/1A DC via Micro-USB port or 5V and GND pins |
| Operating Voltage | 3.3V |
| CPU Clock Frequency | 80 MHz |
| MCU Internal Memory | 64 KB of instruction RAM; 96KB of data RAM |
| MCU External Memory | 4 MB Flash memory [QSPI] |
| GPIO pins | 5 Digital pins [3.3V logic] |
| ADC | 1 pin 10 bit ADC [0-1V input] |
| PWM | All 5 Digital pins capable of PWM [Software PWM] |
| Dimensions | 35mm x 35mm |
| Boot Time | Less then 1 second |

Table 3.1 Bolt module details

**Pulse sensor:**

The heartbeat sensor is developed based on the plethysmography theory. It measures the change in blood volume through anybody’s organ that causes the light intensity to move through that organ. The timing of the pulses is more critical in systems where the heart pulse rate is to be tracked. The rate of heartbeats determines the distribution of blood volume, and the signal pulses are equal to the pulses of heartbeat when light is consumed by the blood.

**Temperature sensor:**

The LM35 series are accurate optimized temperature circuits with output voltage, which is linearly relative to the temperature in centigrade. The LM35 has a vantage point over Kelvin’s linear temperature sensors, as a realistic centigrade scaling does not allow the consumer to delete the huge constant voltage from the display.

**Gas sensor:**

For air quality control systems, the MQ-135 gas sensors are used for NH3, Nicotine, Benzene, Smoke, and CO2 detection as well as measurement. The MQ-135 sensor module comes with a digital pin that enables this sensor to work even without a microcontroller and is beneficial for detecting specific gases. The gasses in PPM are calculated using the analog pins. The analog pin is powered by TTL and works on 5 V, and hence it can be used with most modern microcontrollers.

**Bread board:**

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode). To learn more about individual electronic components

**Jumper wires:**

Jumper wires are simply wiring that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with [breadboards](https://blog.sparkfuneducation.com/what-is-a-breadboard) and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wires.

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. 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 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.

**3.2 SOFTWARE REQUIREMENTS**

**Bolt platform:**

Bolt is a fully integrated IoT platform for developers that helps them build IoT projects and products quickly and easily. It is a platform designed for Makers and Developers to build IoT Projects.

Bolt offers a Wi-Fi module and a cloud platform. The Bolt cloud, directly and via APIs, lets you store data, run analytics on it, and visualize the data in the form of graphs. You can send alerts via SMS and E-Mail when the measured values cross thresholds.

**VM Ware Workstation:**

VMware Workstation Pro is the industry standard for running multiple operating systems as [virtual machines](https://www.vmware.com/topics/glossary/content/virtual-machine) (VMs) on a single Linux or Windows PC to build, test, or demo software.

Here we have used VM ware to write our codes in Linux.

**Twilio:**

Twilio is an American cloud communications platform as a service company based in San Francisco, California. Twilio allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its web service APIs.

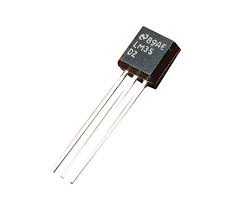
* 1. **SYSTEM DESIGN**
     1. **DESIGN GOAL**

Continuous online patient and patient’s room condition monitoring is the main idea of the proposed system. Therefore, the healthcare monitoring system utilizes the three-stage architectural features, namely (1) Sensor Module (2) Data Processing Module (3) Web User Interface.

The sensors are wired which are used to collect data from the patient’s body and the environment by gathering physiological signs. The collected data are then processed via a Bolt module and send to the gateway server. For the web user interface, Bolt platform is used for the graphical interpretation, and display of collected results. Bolt platform shows the current status and process of transactions. The HTTP protocol provides easy connectivity for the correspondence between a Wi-Fi module and the web server. The HTML user interface is updated every 15 s, allowing patients to be tracked in real-time.

The overall system architecture of the developed system is illustrated in Figure below, it can be seen that all the sensors are used to collect data from hospital environment. The sensors all are connected to a processing unit called Bolt module. Upon attaching these (temperature, heartbeat, gas) sensors, bolt works as a heart of the system. bolt collects sensor data and then wirelessly transfers them to IoT websites.

**SENSORS**

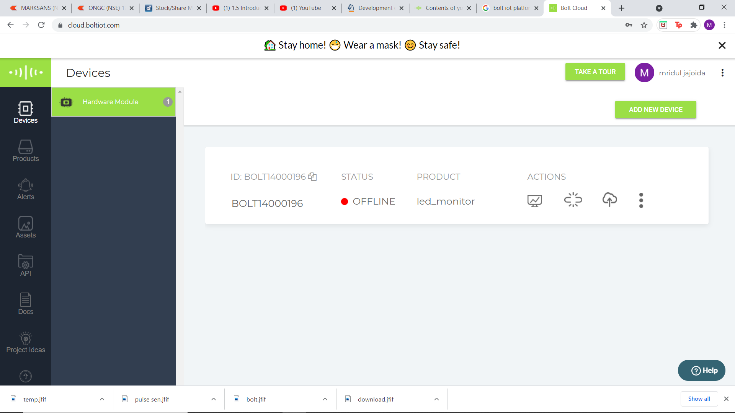
  

Pulse sensor MQ135 LM35

**BOLT MODULE**



**BOLT CLOUD PLATFORM**



**MEDICAL STAFF**



* + 1. **DETAILED DESIGN METHODOLOGY**

The system is implemented using the combination of hardware components. All the hardware components are assembled in the implementation phase. All the sensors are connected with bolt using physical pins. Bolt is used as a processing device as it has a built-in Wi-Fi module. For all sensors, the *V*cc and GND are connected with the *V*cc and GND pin of bolt. In the case of heart beat sensor, the signal pin is connected with D26 pin of bolt. The data pin of LM35 is mapped with D35 pin of microcontroller. These are the case with a specific patient. For room condition monitoring, the data pin of DHT11 is linked with bolt’s D14 pin. In the implementation, DHT11 is only considered for room humidity measurement. The digital out pin of MQ-135 is connected with bolt, for the measurement of toxic gases in room environment.

1. The temperature sensor has three pins which are connected to the microcontroller as shown in the below figure.

|  |  |  |
| --- | --- | --- |
| Pin Number | Pin Name | Description |
| 1 | Vcc | Input voltage is +5V for typical applications |
| 2 | Analog Out | There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C) |
| 3 | Ground | Connected to ground of circuit |

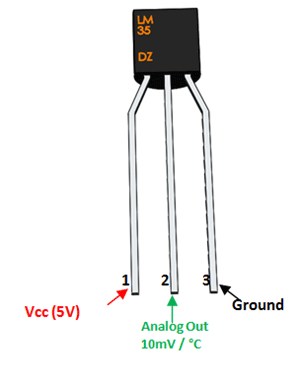


Fig.3.3.2.1 LM35

Table 3.2.2.1 LM35 specification

1. The pulse sensor has three pins connected to the bolt module as follows

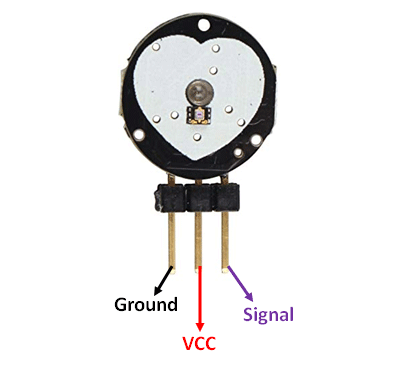


Fig.3.3.2.2 Pulse sensor

|  |  |  |  |
| --- | --- | --- | --- |
| Pin Number | Pin Name | Wire Colour | Description |
| 1 | Ground | Black | Connected to the ground of the system |
| 2 | Vcc | Red | Connect to +5V or +3.3V supply voltage |
| 3 | Signal | Purple | Pulsating output signal. |

Table 3.2.2.2 Pulse sensor specification

1. The gas sensor also has four pins connected as follows

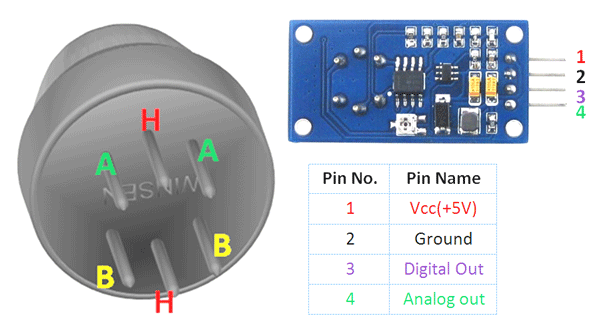


Fig3.3.2.3 MQ135

|  |  |  |
| --- | --- | --- |
| 1 | Vcc | Used to power the sensor, Generally the operating voltage is +5V. |
| 2 | Ground | Used to connect the module to system ground. |
| 3 | Digital Out | You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer. |
| 4 | Analog Out | This pin outputs 0-5V analog voltage based on the intensity of the gas. |

Table 3.2.2.3 MQ135 specification

1. After all the connections the complete system looks like the below figure

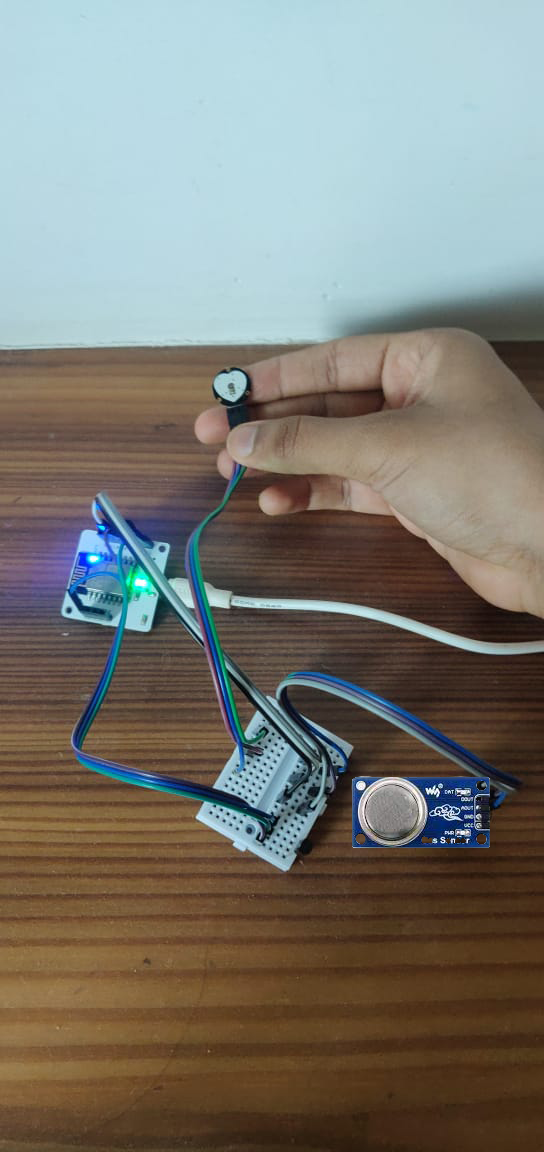


Fig3.3.2. Final System Design

**3.3.3 SENSOR CONDITIONS**

1. **Temperature sensor**

The value in the code is the raw temperature reading, obtained from the LM35 sensor, which is the voltage reading from the pin of the sensor, in this case we will convert this value to the temperature in degree Celsius, using the formula:

**Temperature = 100\*(sensor value)/1024**

< 300 = alert is sent in the form of SMS

Conditions

> 600 = alert is send in the form of SMS

1. **Pulse sensor**

The conditions of pulse sensor are as follows

< 57 = alert sent via SMS

Conditions

> 90 = alert sent via SMS

1. **MQ135 gas sensor**

The conditions of gas sensor are as follows:

If the sensor reads a value greater than 500 it means that the C02 in the environment is beyond normal range and due to this it is risky for the patients hence an alert is sent via SMS to the user.

1. **WORK DONE**
   1. **SOURCE CODE FOR ALL SENSORS**

**First the bolt credentials as well as twilio credentials are stored in a file**

At first, we create a file which will store all the credentials which can later be imported to the source code so that we don’t have to write it again and again. This file will contain –

1. SID: ID of my account in Twilio
2. AUTH\_TOKEN: authentication ID for my account
3. FROM\_NUMBER: the number generated by Twilio from which SMS will be sent
4. TO\_NUMBER: this is my number where the SMS will be sent
5. API\_KEY: this is my bolt account API key
6. DEVICE\_ID: this is the ID of my bolt microprocessor device.

**Code explanation**

 Now create one more file named temp\_sms.py. To do so you have to type sudo temp\_sms.py in the terminal. Now we will write main code to collect the data from the Bolt and send SMS if it crosses the threshold.

1. The algorithm for the code can be broken down into the following steps -
2. Fetch the latest sensor value from the Bolt device.
3. Check if the sensor value is in the range specified in our min and max values.
4. If it is not in range, send the SMS.
5. Wait for 10 seconds.
6. Repeat from step 1.

* In the code, we first have to import our credentials file which has all the credentials. The python json and time libraries are also imported in the same line. Since we have saved our credential file with the .py extension, we can directly import it.
* json is a python library used for handling all operations on JSON objects. JSON is nothing but a data communication format widely used on the Internet for sending/receiving data between a client and server.
* Now we will import Bolt python library which will let us fetch the data stored in Bolt Cloud. To send the SMS, the Sms library is also imported.
* In the above line, we are importing two objects. First one is SMS which will be used to send SMS alerts and the other one is Bolt which is used for accessing data from your Bolt device like the temperature reading.
* Now we will initialize two variables which will store minimum and maximum threshold value. You can initialize any minimum and maximum integer limits to them.
* This would send an alert if the temperature reading goes below the minimum limit or goes above the maximum limit.
* Now to fetch the data from Bolt Cloud, we will create an object called 'mybolt' using which you can access the data on your Bolt.
* Now to send an SMS, we will create an object of the same.
* Since we want to continuously monitor the temperature reading, we will enclose our logic to fetch, compare and send the SMS inside an infinite loop using the `while True:` statement.
* The code continuously fetches the temperature value using `analogRead` function. Since the sensor is connected to A0 pin of the Bolt, we will execute the analogRead() function on the pin A0.
* The response from the Bolt Cloud using the analogRead() function is in a JSON format, so we will need to load the JSON data sent by the cloud using Python's json library.
* The temperature value is inside a field labelled as "value" in the response. We can access the JSON values using the statement `sensor\_value = int(data['value'])`. This line also converts the sensor reading to integer data type for comparing the temperature range.
* This is enclosed inside a try-except block to handle any error that may occur in the code.
* The next line of code checks if the temperature reading is above the maximum limit or below the minimum limit. If it exceeds, then the SMS will be sent.
* The SMS to be sent will contain the text "The Current temperature sensor value is" followed by the temperature value.
* The response from Twilio will be stored inside the `response` variable.
* Once the temperature reading has been sent, we will need to wait for 10 seconds to get the next reading. For this, we will put the program to sleep once every loop iteration.
* The statement `time.sleep(10)` puts the program execution on hold for 10 seconds. This means that the program would not execute for a period of 10 seconds.

This will finish our code for the temperature sensor and send SMS whenever the value crosses threshold value.

Similarly, we will code for the pulse sensor as well as MQ135 sensor while setting different maximum and minimum limits.

**The Output for the code is as follows**

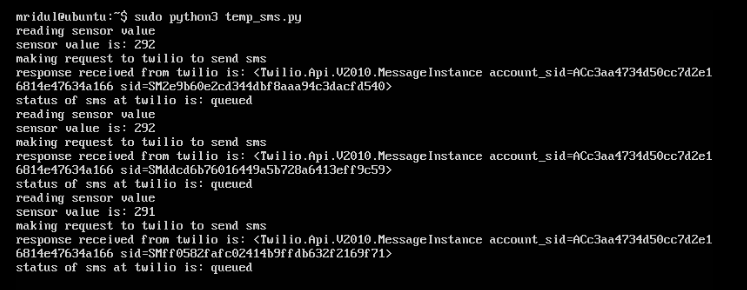
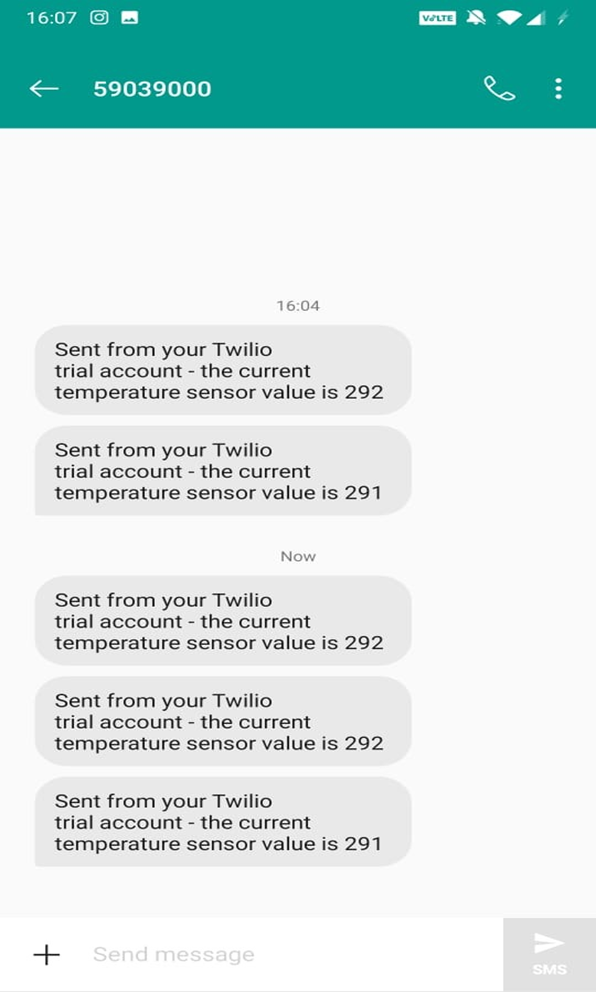


Fig4.1.1 Temperature Output

* 1. **SMS RESULTS**

**The SMS results are as follows**



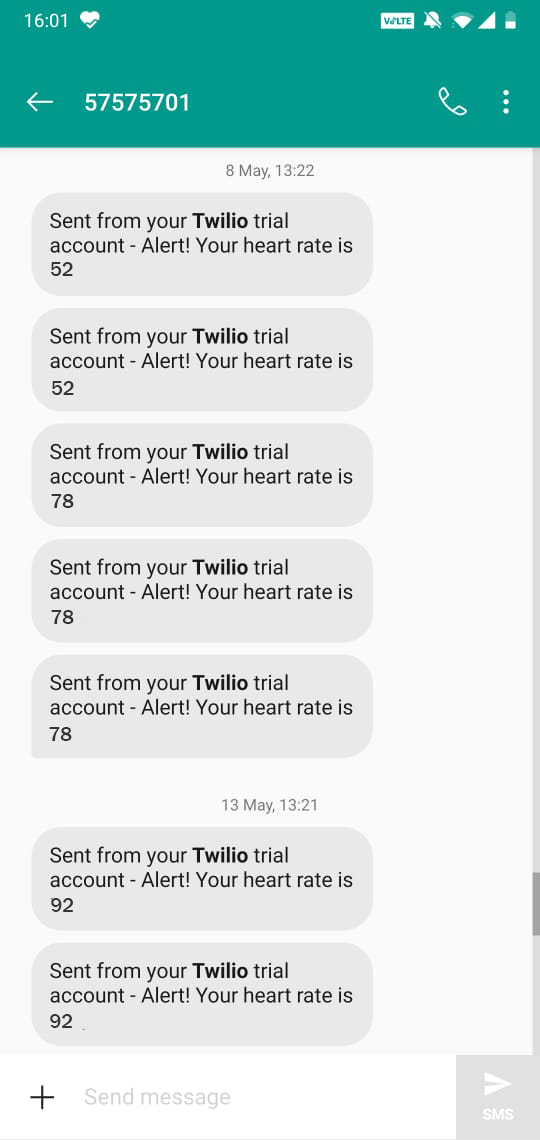


Fig4.2.1 SMS results

**5.CONCLUSION**

The system introduced smart healthcare to monitor the basic important signs of patients like heart rate, body temperature, and some measures of hospital room’s condition such as room humidity, the level of CO and CO2 gases. The rate of success between the observed data and actual data is approximately greater than 95% for all cases of the developed healthcare system. Authentic medical staff can view and track the data in real-time even though the patients perform the tests outside of the hospital. The system can also benefit nurses and doctors in situations of epidemics or crises as raw medical data can be analysed in a short time. The developed prototype is very simple to design and use. The system is very useful in the case of infectious disease like a novel coronavirus (COVID-19) treatment. The developed system will improve the current healthcare system that may protect lots of lives from death.

Although the system looks somewhat bulky, it will be a tiny device by proper manufacturing in the near future. The video feature can be added for face-to-face consultation between the doctors and patients. Some more measures which are very significant to determine a patient’s condition like the level of diabetes, respiration monitoring, etc. can be addressed as future work.

**6.REFERENCES**

* Boltiot course for IoT and ML training
* Md.Milon islam, ashikur Rahaman, Md. Rashedul islam, development of smart monitoring system,13 may 2020.
* Nooruddin S, Milon Islam M, Sharna FA. An IoT based device-type invariant fall detection system. Internet Things. 2020; 9:100130.
* Tamilselvi V, Sribalaji S, Vigneshwaran P, Vinu P, GeethaRamani J. IoT based health monitoring system. In: 2020 6th International conference on advanced computing and communication systems (ICACCS). IEEE; 2020. p. 386–9.
* Acharya AD, Patil SN. IoT based health care monitoring kit. In: 2020 Fourth international conference on computing methodologies and communication (ICCMC). IEEE; 2020. p. 363–8.
* Trivedi S, Cheeran AN. Android based health parameter monitoring. In: 2017 International conference on intelligent computing and control systems (ICICCS). IEEE; 2017. p. 1145–9.
* Kumar SP, Samson VRR, Sai UB, Rao PLSDM, Eswar KK. Smart health monitoring system of patient through IoT. In: 2017 International conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC). IEEE; 2017. p. 551–6.