VIRTUAL ASSISTANCE FOR EMERGENCY SITUATIONS USING IOT

Main Project Report

Submitted by

AKHILA MADHAVAN

Reg no: FIT20MCA-2006

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

Of

A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

JULY 2022

DECLARATION

I hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

Place: Angamaly **AKHILA MADHAVAN**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

CERTIFICATE

This is to certify that the project report titled **"VIRTUAL ASSISTANCE FOR EMERGENCY SITUATIONS USING IOT"** submitted by **AKHILA MADHAVAN**, **(Reg No: FIT20MCA-2006)** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

Project Guide	**Head of the Department**
Ms.Anju L	**Dr. Deepa Mary Mathews**

Submitted for the viva-voce held on.............................. at ....

Examiner :

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and co-operation of various individuals, which is received in abundance with the grace of God.

I hereby express my deep sense of gratitude to **Dr. Manoj George** Principal, FISAT and **Dr. C Sheela** Vice principal, FISAT for allowing me to utilize all the facilities of the college.

My sincere thanks to **Dr. Deepa Mary Mathews** HOD, Department of Computer Applications FISAT ,who had been a source of inspiration. I express heartiest thanks to **Ms.Anju L** my internal guide for his encouragement and valuable suggestion. I express my heartiest gratitude to my scrum master **Dr.Deepa Mary Mathews** and the faculty members in my Department for encouragement and constructive suggestions and comment during the project work.

Finally I wish to express my thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank GOD for everything.

ABSTRACT

Technology plays the major role in healthcare not only for sensory devices but also in communication, recording and display device. It is very important to monitor various medical parameters and post operational days. Hence the latest trend in Healthcare communication method using IOT is adapted. Internet of things serves as a catalyst for the healthcare and plays prominent role in wide range of healthcare applications.

In this project, an intelligent home-based platform, the Home Health-IoT, is proposed. In particular, the platform involves health care management system with enhanced connectivity for the integration of devices and services. The proposed platform seamlessly uses IoT devices (e.g., wearable sensors) with in-home healthcare services (e.g., telemedicine) for an improved user experience and service efficiency.

The Proposed System architecture for IOT Healthcare is consist of accelerometer, temperature sensor, heart beat sensor and a spo2 sensor. Sensors acquire the data of various parameters regarding patients' health using Node MCU mod ule and using the Internet of Things technology, stores that data and displays through the IoT mobile application, which provides access for remote monitoring .Whenever there is an emergency situation occurs an emergency alert will sent to minimum 3 contacts and notifications will sent to the Caretakers App. The Caretaker who uses the Mobile application can choose whether or not sent the details of patient to the hospital . If he chooses take the patient to the hospital then he can select hospital and Sent Details including the gps location of the patient.

Contents

Chapter 1

INTRODUCTION

When the Internet first came into existence, there was an Internet of computers. Each computer was connected and allowed to transfer data back and forth. When the population of Internet users started to grow and social media sites burst into popularity, it became an era of the Internet of People. Countless websites and apps (Facebook, Instagram, and Twitter just being the biggest examples) are available and used by a large percentage of the population to stay connected. So now that every device and every person is connected via the web, the next logical step is to start connecting things.

At first, adding inanimate objects to the connected web can seem like it has no real useful purpose. However, with technology quickly advancing as it has been, the Internet of Things has the "ability to provide smarter services to the environment as more data becomes available". There are boundless possibilities to what can be done with the Internet; for the future, they can work together to create homes and businesses with smart appliances and sensors. The internet of things (IoT) is the network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectiv- ity that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the in- frastructure of the information society".

The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. To implement this connection of physical objects via the web, embedded sensors are affixed to ordinary objects in order to collect data on them. Each of these objects are connected to the Internet, which is how all of the collected data is gath- ered to be analyzed, and ultimately, how these objects can be manipulated.

Nowadays, automation and Internet of Things are changing the world. The day-by-day development of the Internet of Things causes a revolution in modern technology, which makes our life easier and automated. The main idea of the proposed system is to continuous monitoring of the patients over internet.

Medical emergencies can happen anytime, anywhere. Victims in accidents, medical needs are sometime not able to speak for themselves. Also for emergency identification and communication devices and things like mobile phones, wallet /purses are either stolen or separated from victim because of some reasons A system which can help victim display their identity to doctors when in need and helps an individual keep a track of their respective members, creates a high chance for victim to survive a situation of medical emergencies. Medical devices which are IOT based are of different size, shape which contains high electronic equipment that are linked with network devices.

Due to a busy schedule and irregular lifestyle, health hazard is not an age dependent factor in the recent era. Also busy schedules and working pattern of individuals not allow them to take care the elderly or patients at home. Under these circumstances, Internet of Things has provided a much easier solution for remote real-time health monitoring of patients from the hospital as well as home. Use of Sensor reduces the human error, and the size of the system reduces the occupied space of the room. In-home healthcare services based on the Internet-of Things (IoT) have great business potential; however, a comprehensive platform is still missing.

Chapter 2

PROOF OF CONCEPT

The IoT is an ongoing huge trend in upcoming growing technologies that can leave a huge impact on the very single business trending and can be defined as the inter and intra connection of different yet unique objects and devices present within day today environment with benefits increasing as needs increases. By using IoT, Medical health concept uses sensors, computing, communication for healthcare is attracting more researchers to use this in various services. Various techniques flourishing IoT for social causes have been developed or still in progress. In last few years, researchers have taken a keen interest in IoT in healthcare system. Minimizing human effort to a certain extent by keeping it safe as well has become new trend in research world. It is not new concept still overgrowing, empowering and gaining attention in increasing order day by day. In this usage of electronic devices using cloud computing; private or public.

Many examples are there such as garage door openers which uses user identity as a main concept to send information to user mobile phones. All can see how healthcare using IoT is picking up speed in accelerating healthcare system network even though problems remain continuously. In paper [1], IoT is used for pediatric and elderly care and for their private health management. Though it has few drawbacks as in this there is no future in E-health and it doesn't maintain standard network model which has less chances of data protection. In paper [2], patient monitoring can be by staff anywhere and everywhere. But it has complex infrastructure which in emergency can cause problems. In paper [3], various interfacing is done between devices so that ER- doctors can use it without any problems. But it requires wide range of data to be accessed through app as this requires heterogeneity of database. In paper [4], it put forth idea of smart hotel management for monitoring health care system. The only difficulty in this is maintaining current and past medical information of patient with a good track of them.

A Remote health monitoring system is an extension of a hospital medical system where a patient's vital body state can be monitored remotely. Traditionally the detection systems were only found in hospitals and were characterized by huge and complex circuitry which required high power consumption. Continuous advances in the semiconductor technology industry have led to sensors and microcontrollers that are smaller in size, faster in operation, low in power consumption and affordable in cost.

This has further seen development in the remote monitoring of vital life signs of patients especially the elderly. The remote health monitoring system can be applied in many scenarios like A patient is known to have a medical condition with unstable regulatory body system. This is in cases where a new drug is being introduced to a patient. And if A patient is prone to heart attacks or may have suffered one before. The vitals may be monitored to predict and alert in advance any indication of the body status.

Remote health monitoring can provide useful physiological information in the home. This monitoring is useful for elderly or chronically ill patients who would like to avoid a long hospital stay. Wireless sensors are used to collect and transmit signals of interest and a processor is programmed to receive and automatically analyze the sensor signals.A cost-effective and fast responding alert mechanism is inevitable. Proper implementation of such systems can provide timely warnings to the medical staffs and doctors and their service can be activated in case of medical emergencies.

A wearable identity (bracelet, eyelet for watches or pendant) with basic medical history and emergency contact numbers will be provided to registered users.Along with GPS of the patient In the time of emergency, the person will get immediate medical help through Mobile Application and by sending alerts to emergency numbers. hospital approached. Incase of any emergency we need to contact caretaker or hospital through manually. Even if the patient is under camera surveilance it can't detect the temperature , heart beat etc. And after reaching hospitals , the staff needs to check everything related to the patient for providing proper identification and complete medical information of every emergency patient. Which requires more time in complex cases and the patient history may not be available at that time which enhance risk in giving treatments. During design the various characteristics of the future medical applications adhered .

a) Integration with current trends in medical practices and technology,

b) Real-time, long-term, remote monitoring, miniature, wearable sensors and long battery life of a designed device.

c) Assistance to the elderly and chronic patients. The device should be easy to use with minimal buttons

2.1 Existing System

Incase of any emergency patient or the guardian need to contact caretaker or hospital through manually. Even if the patient is under camera surveilance it can't detect the temperature , heart beat etc. And after reaching hospitals , the staff needs to check everything related to the patient for providing proper identification and complete medical information of every emergency patient. Which requires more time in complex cases and the patient history may not be available at that time which enhance risk in giving treatments.

2.2 Proposed System

Using the proposed system, data can be sent wirelessly to the Patient Monitoring System, allowing continuous monitoring of the patient. Contributing accuracy in measurements and providing security in proper alert mechanism give this system a higher level of customer satisfaction and low-cost implementation in hospitals. Thus, the patient can engage in his daily activities in a comfortable atmosphere where distractions of hardwired sensors are not present. Physiological monitoring hardware can be easily implemented using simple interfaces of the sensors with a Microcontroller and can effectively be used for healthcare monitoring. This will allow development of such low-cost devices based on natural human-computer interfaces.

The system we proposed here is efficient in monitoring the different physical parameters of bedridden patients and then in alerting the concerned medical authorities if these parameters bounce above its predefined critical values. Thus, remote monitoring and control refer to a field of industrial automation that is entering a new era with the development of wireless sensing devices. The Internet of Things (IoT) platform offers a promising technology to achieve the healthcare services, and can further improve the medical service systems. IoT wearable platforms can be used to collect the needed information of the user and its ambient environment and communicate such information wirelessly, where it is processed or stored for tracking the history of the user. Such a connectivity with external devices and services will allow for taking preventive measure (e.g., upon foreseeing an upcoming heart stroke) or provide immediate care (e.g., when a user falls and needs help).

2.2.1 Objectives of proposed system

• Continues monitoring of health parameters of patients or elderly people at home

• Remote monitoring of patient health via mobile application

• Reduce human error in accessing health parameters

• Access data for telemedicine system

• Reduce post operation periods at hospitals, hence the cost of treatments Reduce effort of doctors/care

• Automatic alert will send to care taker in case of emergency

Chapter 3

IMPLEMENTATION

Experimental prototyping using sensor units, microcontroller unit having IoT Connectivity. Temperature sensor, Spo2 sensor, heart beat sensor and accelerometer equipped with the module can automatically monitor the health parameters and detect any unhealthy situation of patient. Then situation alert is communicated by IoT to the care taker/ambulance drivers/doctors using the GSM module . Location of the patient is fetched using GPS module and communicated via IoT network.

Arduino microcontroller connected with sensors which are attached to the patient. All the sensors and location data sent from microcontroller to NodeMCU ,Which enables wifi connection setup with the Blynk Mobile application. Caretaker / guardian can log in to the mobile application to monitor patient's data at any point in time.

In case of emergencies, like temperature increase or heartbeat spike or Fall detection or Blood Oxygen level mismatches an SMS alert sent to minimum 3 emergency contacts and a notification to the caretaker app through mobile And at any point of time if the caretaker is unabe to handle the situation they can approach the hospital by simply click on the mobile application which follows Selecting Hospital and sending Details of the patient along with the GPS location of the patient is transferred to the hospital.So they can track patient's location and medical history which would helpful for immediate actions in case of emergencies.

While developing any micro-controller based electronic system, there are some steps which must be followed.

These steps are:

1. Firstly Deciding the sensing Parameters and they are Temperature,Fall Detection, Heartbeat rate, Blood Oxygen Level

2. Selection of efficient high quality Sensors ,they are MAX30102, MPU6050 and DHT11. For message/alert sening to multiple contacts GSM module used . And for fetching location GPS Module used (NEO 6M) .

3. Design of circuit diagram

4. Dangerous detection and alert sending

5. Identify the Emergency Numbers

6. Caretakers App Management and selection of Hospitals

3.1 ARCHITECTURE

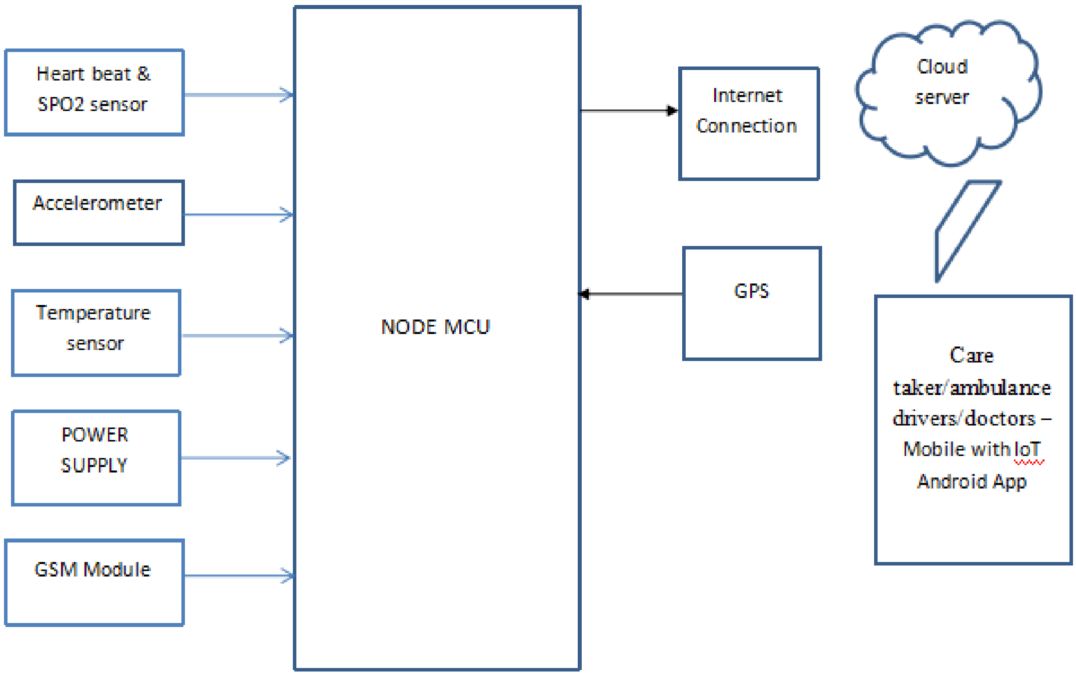


Figure 3.1: Block Diagram of Proposed System

The sensors help to transform the physical world data (e.g: temperature, pres- sure, humidity, etc) including human health data (heart rate, oxygen saturation, blood pressure, blood glucose, etc) to the digital world and the actuators trans- forms the digital data to physical actions (e.g: Infusion pumps,alert sending,health monitoring, dialysis system, etc). The IoT devices have sensors for receiving signals from the environment for analysis, or actuators for controlling the environment based on the inputs, or both sensors and actuators. These devices connect with each other through internet transfer and cloud storage for communication with similar devices and people.

3.2 HARDWARE REQUIREMENTS

3.2.1 Arduino Uno

Figure 3.2: Arduino UNO

Arduino Uno - ATMEGA328P

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package. This Arduino Uno is an original microcontroller board from Arduino officials 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. 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.

Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer. IDE is equally compatible with Windows, MAC or Linux Systems, however, Windows is preferable to use. Programming languages like C and C++ are used in IDE. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

TECHNICAL SPECIFICATIONS

Microcontroller : Microchip ATmega328P

Operating Voltage : 5 Volts

Input Voltage : 7 to 20 Volts

Digital I/O Pins : 14 (of which 6 can provide PWM output)

UART: 1, I2C: 1, SPPI : 1

Analog Input Pins : 6

DC Current per I/O Pin : 20 mA

DC Current for 3.3V Pin : 50 mA

Flash Memory : 32 KB of which 0.5 KB used by bootloader

SRAM: 2 KB, EEPROM : 1 KB

Clock Speed : 16 MHz

Length : 68.6 mm

Width : 53.4 mm

Weight : 25 g

3.2.2 NodeMCU

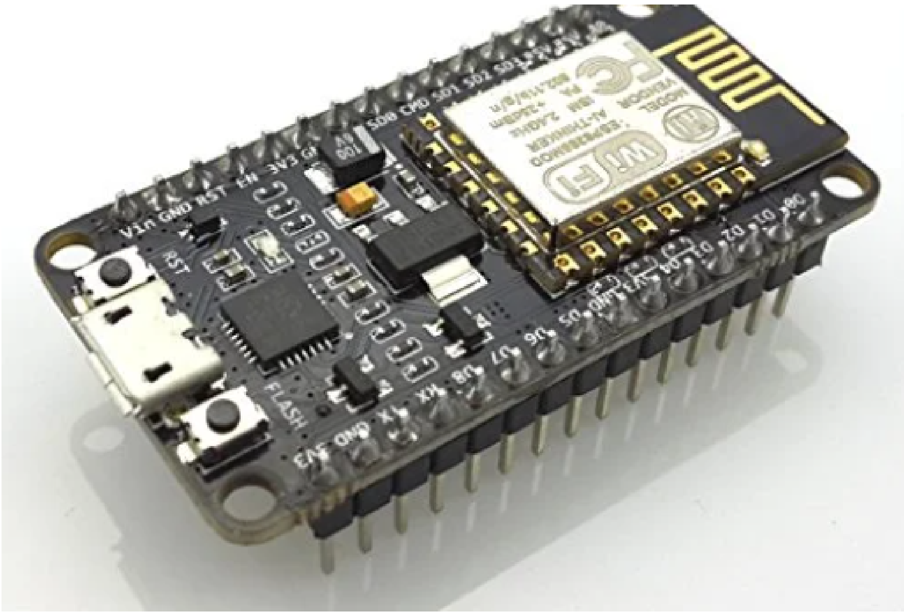


Figure 3.3: NodeMCU

The NodeMCU - ESP8266 has 30 pins in total out of which there are 17 GPIO pins. GPIO stands for General Purpose Input Output. There are the 9 digital pins ranging from D0-D8 and there is only one analog pin A0, which is a 10 bit ADC. The D0 pin can only be used to read or write data and can’t perform other options. The ESP8266 chip is enabled when the EN pin is pulled HIGH. When pulled LOW the chip works at minimum power. The board has a 2.4 GHz antenna for a long-range of network and the CP2102 is the USB to TTL converter. The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

There’s also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays. The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with an LDO (low dropout) voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA. It has three 3v3 pins along with 4 GND pins. The power supply is via the onboard MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin is used to directly supply the ESP8266. Moreover, it requires 80mA Operating Current and 20 µA during Sleep Mode.

TECHNICAL SPECIFICATIONS

Micro-controller: Tensilica 32-bit RISC CPU Xtensa LX106

Operating Voltage: 3.3V

Input Voltage: 7-12V

Digital I/O Pins (DIO): 16

Analog Input Pins (ADC): 1

UARTs: 1

SPIs: 1

I2Cs: 1

Flash Memory: 4 MB

SRAM: 64 KB

Clock Speed: 80 MHz

USB-TTL based on CP2102 is included onboard, Enabling Plug n Play

PCB Antenna

3.2.3 DHT 11

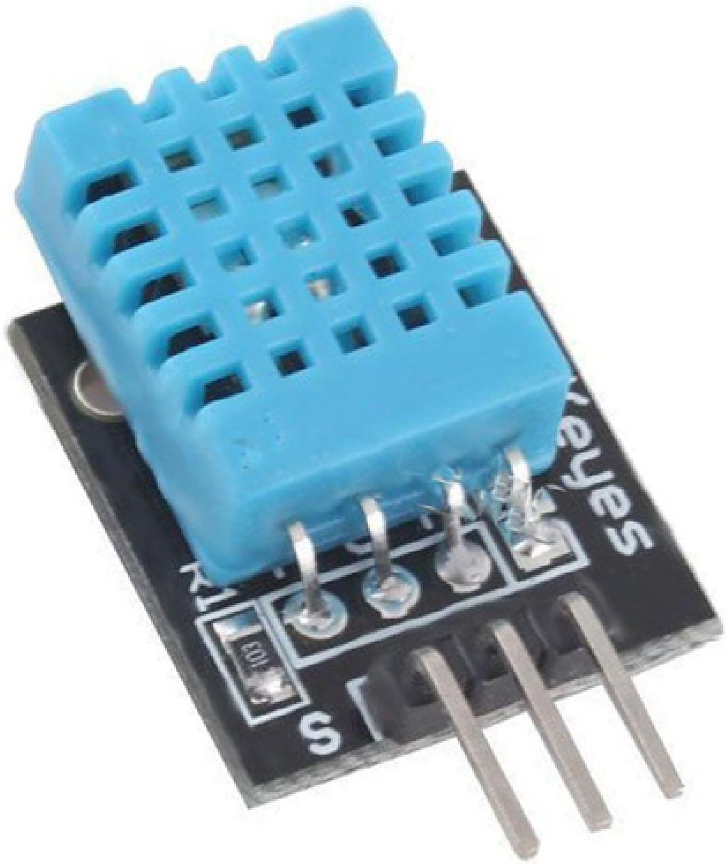


Figure 3.4: DHT11-Sensor

DHT11 Temperature Humidity Sensor features a temperature humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive type humidity measurement component and an NTC temperature measurement component, and connects to a high- performance 8-bit micro-controller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmer's in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

3.2.4 MAX 30102

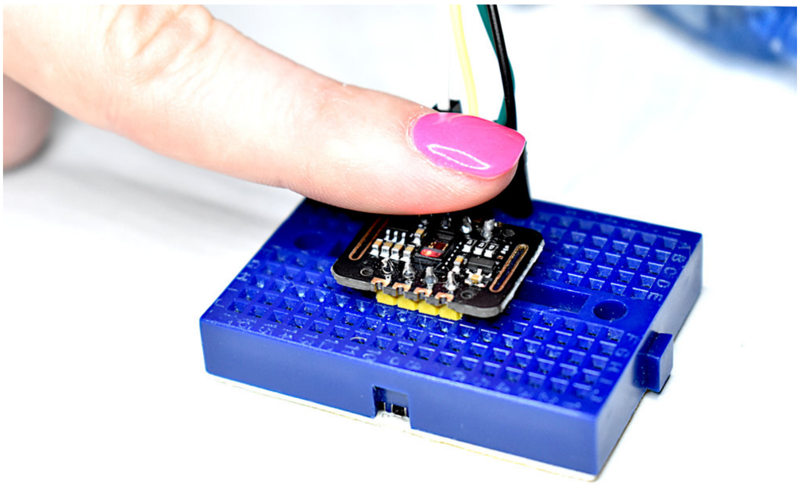
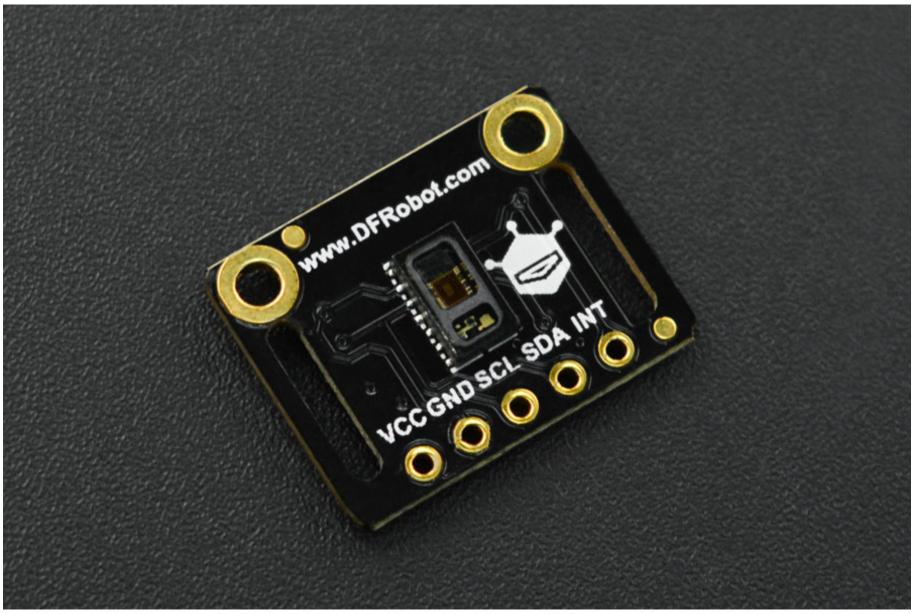


Figure 3.5: MAX 30102

The MAX30102 is an integrated pulse oximetry and heart-rate monitor module. It includes internal LEDs, photodetectors, optical elements, and low-noise electronics with ambient light rejection. The MAX30102 provides a complete system solution to ease the design-in process for mobile and wearable devices. The MAX30102 operates on a single 1.8V power supply and a separate 3.3V power supply for the internal LEDs. Communication is through a standard I2C-compatible interface. The module can be shut down through software with zero standby current, allowing the power rails to remain powered at all times.

The MAX30102 Sensor shines both the light through the skin and measures the reflection with the photodetector. This method of pulse detection through light is called Photoplethysmogram. The working of the sensor can be divided into two parts one is heart rate measurement and another is blood oxygen level measurement.



Heartbeat and Oxymetery Sensor

The oxygen in the hemoglobin has a specific characteristic, that it can absorb IR light. When the concentration of hemoglobin is more, the redder the blood. Which simply means it can absorve more IR light. As the blood is pumped through the veins in the finger the amount of reflected light changes creating an oscillating waveform. And by measuring this wave we can get the heartbeat reading. Blood oxygen level measurement works on the principle that Red and IR light varies depending upon the oxygen level in your blood. Deoxygenated blood absorbs more RED light while blood with more sufficient oxygen absorbs more IR light. By measuring the ratio between two we can measure oxygen level.

3.2.5 MPU 6050

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer. Micro machined accelerometers are increasingly present in portable electronic devices and video game controllers, to detect the position of the device or provide for game input.

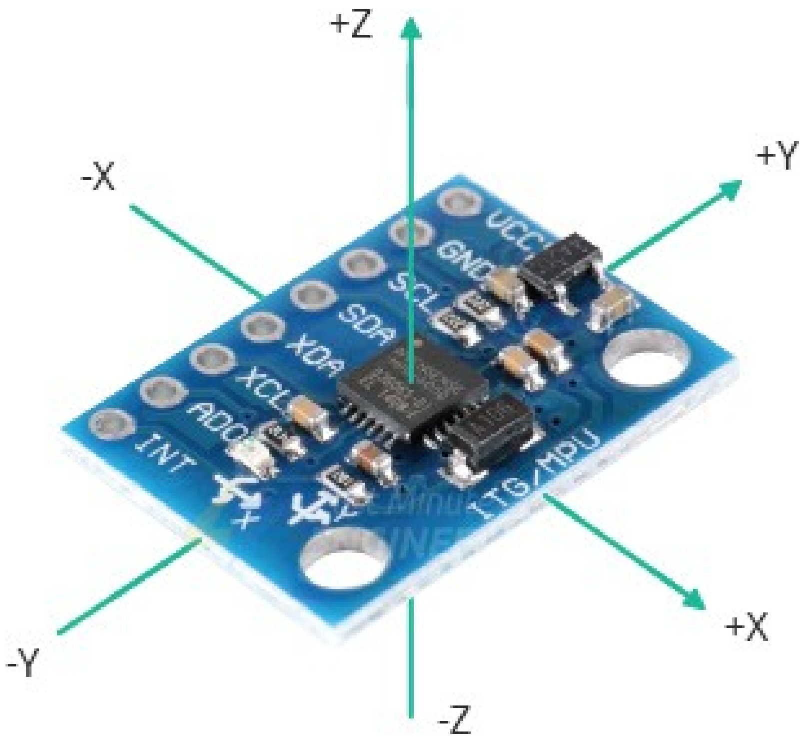


Figure 3.6: Fall detection sensor

Single- and multi-axis models of accelerometer are available to detect magnitude and direction of the proper acceleration (or g-force), as a vector quantity, and can be used to sense orientation (because direction of weight changes), coordinate acceleration (so long as it produces g-force or a change in g-force), vibration, shock, and falling in a resistive medium (a case where the proper acceleration changes, since it starts at zero, then increases).

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers. It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc. If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

3.2.6 GSM Module

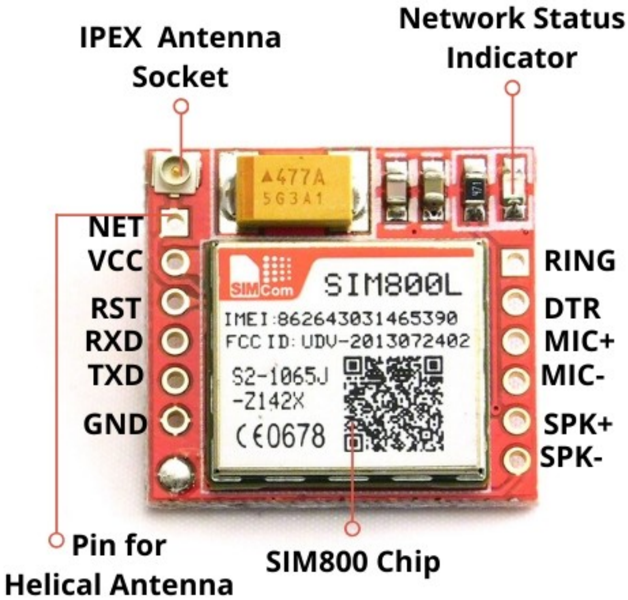


Figure 3.7: GSM Module

This GSM Modem can accept any GSM network operator SIM card and act just like a mobile phone with its own unique phone number. Advantage of using this modem will be that you can use its RS232 port to communicate and develop embedded applications. Applications like SMS Control, data transfer, remote control and logging can be developed easily.

The modem can either be connected to PC serial port directly or to any microcontroller. It can be used to send and receive SMS or make/receive voice calls. It can also be used in GPRS mode to connect to internet and do many applications for data logging and control. In GPRS mode you can also connect to any remote FTP server and upload files for data logging.

This GSM modem is a highly flexible plug and play quad band GSM modem for direct and easy integration to RS232 applications. Supports features like Voice, SMS, Data/Fax, GPRS and integrated TCP/IP stack.

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

3.2.7 GPS- NEO6M

Figure 3.8: GPS-NEO6M

The Global Positioning System (GPS) is global navigation satellite system which uses a constellation of between 24 and 32 Medium Earth Orbit satellites that transmit precise microwave signals, that enable GPS receivers to determine their location, speed, direction, and time. GPS has become a widely used aid to navigation worldwide, and a useful tool for map-making, land surveying, commerce, scientific uses, tracking and surveillance, and hobbies such as geo-caching and way marking. Also, the precise time reference is used in many applications including the scientific study of earthquakes and as a time synchronization source for cellular network protocols.

GPS has become a mainstay of transportation systems worldwide, providing navigation for aviation, ground, and maritime operations. Disaster relief and emergency services depend upon GPS for location and timing capabilities in their life-saving missions. The accurate timing that GPS provides facilitates everyday activities such as banking, mobile phone operations, and even the control of power grids. Farmers, surveyors, geologists and countless others perform their work more efficiently, safely, economically, and accurately using the free and open GPS signals.

A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include

- The time the message was transmitted
- Satellite position at time of message transmission

The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite using the speed of light. Each of these distances and satellites' locations define a sphere. The receiver is on the surface of each of these spheres when the distances and the satellites' locations are correct. These distances and satellites' locations are used to compute the location of the receiver using the navigation equations. This location is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS units show derived information such as direction and speed, calculated from position changes.

3.3 SOFTWARE REQUIREMENTS

3.3.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text.

The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port.

3.3.2 Features of Arduino IDE

- Multi-Platform Application Arduino IDE works on the three most popular operating systems: Windows, Mac OS, and Linux. Aside from that, the application is also accessible from the cloud. These options provide programmers with the choice of creating and saving their sketches on the cloud or building their programs locally and upload it directly to the board.
- Board Management Arduino IDE comes with a board management module, where users can select the board they want to work with at the moment. If they wish to change it, they can do so easily from the drop down menu. Modifying their selection also automatically updates the PORT infos with the data they need in relation to the new board.
- Straightforward Sketching With Arduino IDE, users can create programs called sketches that are built with a text editor. The process is a straightforward one though it has several bells
- The tool is armed with a board management module, wherein users can choose which board they want to use. If another board is needed, they can seamlessly select another option from the dropdown menu. PORT data is updated automatically whenever modifications are made on the board or if a new board is chosen.

3.3.3 Blynk IoT

At our core there is suite of powerful software products to build IoT apps, connect things, people, and data.Blynk is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen. Using the widgets, you can turn pins on and off or display data from sensors.

Currently, Blynk supports most Arduino boards, Raspberry Pi models, the ESP8266, Particle Core, and a handful of other common microcontrollers and single-board computers, and more are being added over time. Arduino Wi-Fi and Ethernet shields are supported, though you can also control devices plugged into a computer's USB port as well.While there are other platforms for controlling hardware over the internet (Particle, ThingSpeak, Temboo, IFTTT), Blynk is one of the most user-friendly I've seen yet, and it's also free and open-source under an MIT license.

3.3.4 C (Programming language)

C was originally developed at Bell Labs by Dennis Ritchie, between 1972 and 1973. It was created to make utilities running on Unix. Later, it was applied to re-implementing the kernel of the Unix operating system.During the 1980s, C gradually gained popularity. Nowadays, it is one of the most widely used programming languages.

With C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO).

C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming.

A standards compliant C program that is written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code; the language has become available on various platforms, from embedded micro-controllers to supercomputers.

3.4 MODULES

The project is divided into five modules.

1. Patient Monitoring

2. Dangerous deterioration

3. Alert Sending

4. Caretaker App Management

5. Hospital Telemedicine System

3.4.1 Patient Monitoring

In this module the device is continuously monitoring the health parameters of the patient. In recent times, several systems have come up to address the issue of remote health monitoring. The systems have a wireless detection system that sends the sensor information wirelessly to a remote server. This has multiple parameters being monitored at the same time. An example of such a system can be found in High Dependency Units (HDU), Intensive Care Units (ICU), during the surgery at a hospital theatre or Post surgery recovery units in Hospitals.

Several parameters that are monitored include Temperature , Fall Detection, Blood Oxygen level , Heart beat rate . The Multiparameter monitoring system basically proof that a patient is alive or recovering.

3.4.2 Dangerous deterioration

The data from the parameter monitoring systems was then availed for remote detection. The use of sensors detects the conditions of the patient and the data is collected and transferred using a microcontroller. Doctors and nurses or the caretakers need to visit the patient frequently to examine his/her current condition.

So here the wearable device is able to identify the dangerous situations of the patient without any human intervention .The dangerous situations are identified using setting threshold values for normal and abnormal conditions and limiting values for each monitoring parameters , like fall detected, temperature

3.4.3 Alert Sending

Patient Health Monitoring System is mainly focus on emergency medical support and continously diagnosing the health condition of the patients. Giving care and health assistance to the bedridden patients at critical stages with advanced medical facilities have become one of the major problems in the modern hectic world.

In hospitals or home the patient whose physical conditions must be monitored frequently as a part of a diagnostic procedure, the need for a cost-effective and fast responding alert mechanism is inevitable. Proper implementation of such systems can provide timely warnings to the medical staffs and doctors and their service can be activated in case of medical emergencies.

So whenever there is a dangerous situation occurs the GSM module will send alerts to the emergency contacts thus ensure the patient safety. Also it will also send notifica tions to the Health Monitoring App handled by Caretaker.So that the neighbours /Relatives / Friends can aware of the condition of the patient .

Using the proposed system, data can be sent wirelessly to the Patient Monitoring System, allowing continuous monitoring of the patient. Contributing accuracy in measurements and providing security in proper alert mechanism give this system a higher level of customer satisfaction and low-cost implementation in home based health monitoring .

3.4.4 Caretaker App Management

By the help of Blynk App create an UI for the application. Health Monitoring mobile application will work when connect with Nodemcu-Wifi module. The android application is downloaded to the caretakers phone and the Patient health monitoring device connects with the application , then all the health parameters reading can be send to the caretakers app wirelessly. Allowing continuous monitoring of the patient and simultaneous notifications will get on the app with the current state of the patient.

Thus, the caretaker can engage in his daily activities in a comfortable atmosphere where distractions of hardwired sensors are not present. Caretaker can add patient history, medical records , medicine names ,recent health hazards through the mobile application and in case of extreme emergency caretaker can send patient data to the concerned medical authorities. He can choose to opt send details of patient to the hospital followed by the select hospital button .

3.4.5 Hospital Telemedicine System

Nowadays,diseases are on the rise. Cardiac arrest is quoted as the major contributor to the sudden and unexpected death rate in the modern stress filled lifestyle around the globe. This module provides mobility to the doctor and the patient, by adopting a simple and popular technique, detecting the abnormalities in the bio signal of the patient and sending an SMS alert to the doctor through Global System for Mobile(GSM) thereby taking suitable precautionary measures thus reducing the critical level of the patient.

The post-operative patients can develop complications once they are discharged from the hospital. In some patients, the cardiac problems may reoccur, when they start doing their routine work. Hence the heart rate of such patients needs to be monitored for some time after their treatment. This helps in diagnosing the improper functioning of the heart and take precautions. Some of these lives can often be saved if acute care and treatment is provided within the so-called golden hour. So, the need for advice on first-hand medical attention and promotion of good health by patient monitoring and follow-up becomes inevitable.

Hence, patients who are at risk require that their health to be monitored frequently whether they are indoors or outdoors so that emergency treatment is possible. Telemedicine is widely considered to be part of the inevitable future of the modern practice of medicine.

When the situation which is not able to handled by the caretaker or the emergency contacts , the Caretaker can select the hospital in case of extended emergency , a message will sent through the caretakers app to the hospital which includes all the history and details of the patient with the current location of the patient. Doctor will able to prescribe the medicines to the long distance patient who needs help. Then all the data related to the patient will be transferred to hospital along with the gps location can be send

Chapter 4

RESULT ANALYSIS

The system architecture focuses on two parts of system. This proposed system for Virtual assistance is capable for ensuring the safety of the patients .The Proposed System architecture for IoT Healthcare is consist of accelerometer, temperature sensor, heart beat sensor and a spo2 sensor. Sensors acquire the data of various parameters regarding patients' health using Node MCU module and using the Internet of Things technology, takes The measurements continuously and stores that data and displays through the IoT mobile application. The system is able to provides access for remote monitoring and GSM module for sending alert /message for emergency contacts .The caretaker will able to select the hospital in case of critical situations through the mobile application and The Doctor can go through the patient history, and take timely decisions.

Along with the wearable identity, a mobile application will be available to access the user account as well as updating the medical information of the patient. The mobile application also provides a way to track the patient associated to the user account and gets historic data as well. An emergency feature to send assigned contacts an alert is also provided in the application as an IoT extension, a GPS tracing device can be requested to track a person using the same mobile application

4.1 Connection Diagram

The Circuit Diagram is the Graphical representation of the electrical Circuit. A pictorial circuit diagram uses simple images of components, while a schematic diagram shows the components and interconnections of the circuit using standardized symbolic representations. The presentation of the interconnections between circuit components in the schematic diagram does not necessarily correspond to the physical arrangements in the finished device. Unlike a block diagram or layout diagram, a circuit diagram shows the actual electrical connections. A drawing meant to depict the physical arrangement of the wires and the components they connect is called artwork or layout, physical design, or wiring diagram. Circuit diagrams are used for the design (circuit design), construction (such as PCB layout), and maintenance of electrical and electronic equipment.

4.2 Connection Diagram of System

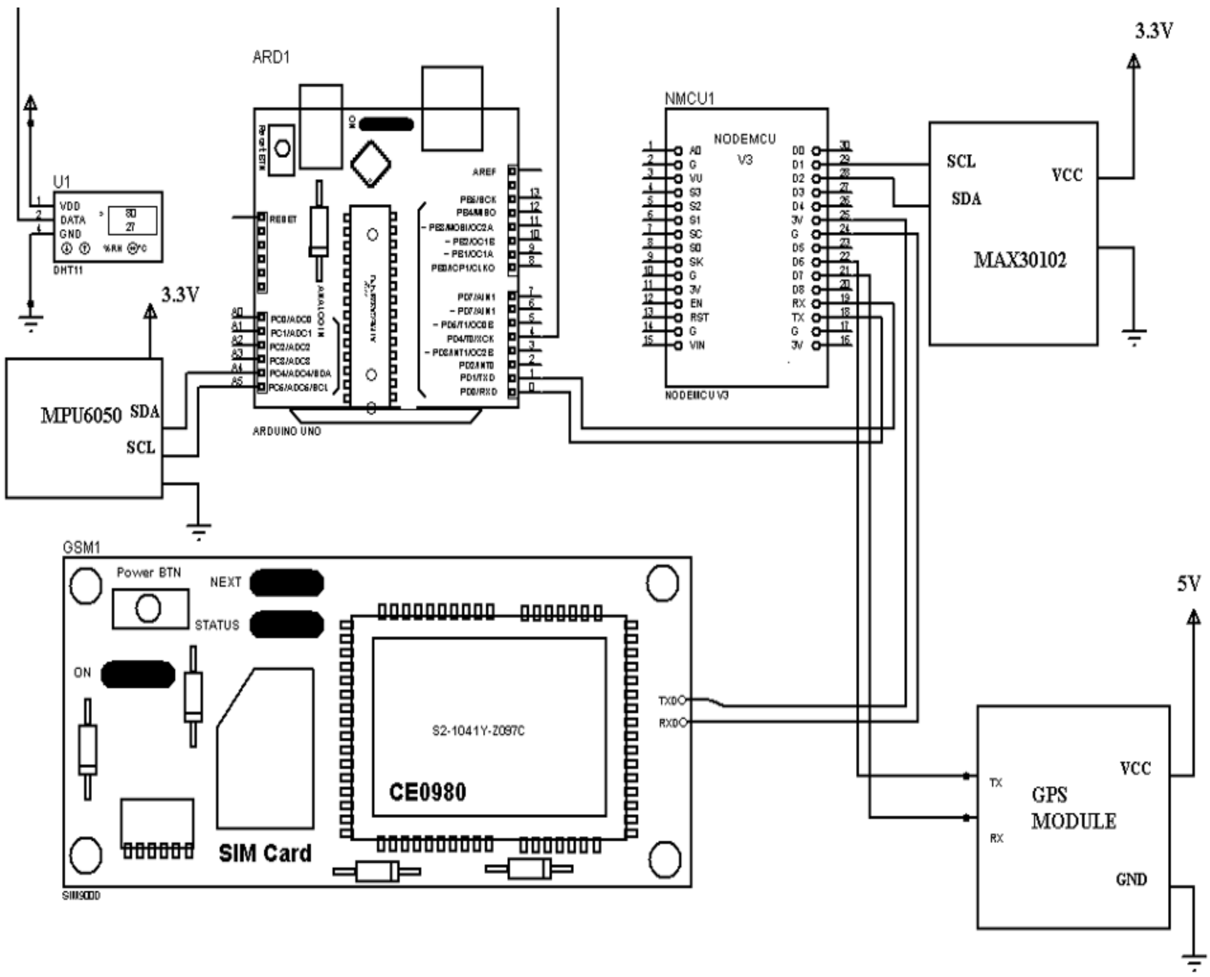


Figure 4.1: Cicuit Diagram

The proposed design of Virtual assistance for emergency situations using IOT uses an Arduino module which connects Temperature sensor and Fall detection sensor (DHT 11, MPU 6050) and NodeMCU moduleis connected with Heartbeat and Oxymetery sensor (MAX 30102) , The NodeMCU is needed to provide wifi connectivity to the design . A GSM and GPS module is enabled for live tracking and alert sending. The NodeMCU receives all the informations from the Arduino Uno and which sends all these details to the Health Monitoring App.

4.3 Testing

Hardware testing is the process of evaluating and verifying that a hardware product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

4.3.1 Test Cases

Testing is based on test cases. It describes which feature or service test attempts to cover. In test cases specify what you are testing and which particular feature it tests.

- Test the normal use of system
- Test the abnormal, but reasonable use of system
- Test the abnormal and reasonable use of system
- Test the boundary Conditions

4.3.2 Functionality Testing

Can be performed on hardware or software products to verify that your product functions exactly as designed. The general purpose of hardware and software functionality testing is to verify if the product performs as expected and documented, typically in technical or functional specifications. Developers creating a new product start from a functional specification, which describes the product's capabilities and limitations.

The prime objective of Functionality testing is checking the functionalities of the software system. It mainly concentrates on Mainline functions, Testing the main functions of an application. Basic Usability, It involves basic usability testing of the system. It checks whether a user can freely navigate through the screens without any difficulties. Accessibility, Checks the accessibility of the system for the user. Error Conditions, Usage of testing techniques to check for error conditions. It checks whether suitable error messages are displayed.

4.4 Hardware Test cases

SI.NO	PROCEDURE	TEST CASE DESCRIPTION	EXPECTED OUTPUT	ACTUAL RESULT	STATUS
1	Temperature Sensor-DHT11	Thermistor to measure the surrounding air and outputs a digital signal on the data pin	Value changes as the surrounding temperature changes	DHT11 sensor work normally as expected	PASS
2	Fall Detection-MPU6050 sensor	Measure Acceleration along three axes (X,Y,Z)	When the value goes beyond the threshold then Fall will Detected	When the sensor moves upside down it will exactly detect as FALL	PASS
3	Heartbeat Sensor-MAX30102	Measuring GPM-Gallons Per Minute	Continuosly monitor the Heart beat rate	MAX30102 sensor work normally as expected	PASS
4	Pulse Oxymetry-MAX30102 SPO2 sensor	Measuring the Oxygen level in the blood	Continuosly monitor the Oxygen level	MAX30102 sensor work normally as expected	PASS
5	GPS	Tracking Location of Patient	Give longitude and latitude of the location	Gives exact location details	PASS
6	GSM Module	For Sending Multiple messages to Multiple Numbers	Sending alerts to 3 emergency contacts and complete patient details to doctor	Sending alerts successfully to all these contacts	PASS
7	Health Monitoring app	Shows the real time measurements of the health monitoring parameters	All the 4 parameters are monitored and displayed and Notifications are shown.	Shown notifications and gives values of the health parameters.	PASS

Figure 4.2: Health Monitoring

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The main purpose of designing the Remote Patient Health Monitoring System to diagnose the health condition of the patients and timely alerts and information passing are correctly achieved. Giving care and health assistance to the bedridden patients at critical stages with advanced medical facilities have become one of the major problems in the modern hectic world, and this difficulty level reduced by using this proposed system .

The patients whose physical conditions must be monitored frequently as a part of a diagnostic procedure, the need for a cost effective and fast responding alert mechanism is inevitable.Thus the objective of developing monitoring systems is to reduce health care costs by reducing physician office visits, hospitalizations, and diagnostic testing procedure. Also the transparency of this system helps patients to trust it. Because of wireless data transmission over internet, health related data will be send to doctor's personal mobile or hospital . So, need not to go hospital every time and sending massage to the doctor gets immediate remedy related to the health condition.

It has the system structure with a high possibility of user satisfaction. Those needy people can use it without any disturbance, insecurity of losing something as in this system, they are their owner. It Integrate with current trends in medical practices and technology, Real-time, long-term, remote monitoring, miniature, wearable sensors and long battery life of a designed device. High level of assistance to the elderly and emergency patients.

The device is designes as very easy to use with minimal buttons so each and every people can make use of it .

By using the Proposed System one can ensure the security , safety of every registered patient and can reach the patient by tracking location . The system is portable because the device is made by Bio-sensors in wearable IDs. The Health Monitoring App helps to update the Medical history of patients ,thus recent activity/conditions can be added and previous/old history (if unwanted) can be removed from viewing . All the features are tested and validated and give expected results accurately.

5.2 Future Scope

As health care services are important part of our society, automating these services lessen the burden on humans and eases the measuring and diagnosing process. Many further improvements can be made in our system to make it better and easily adaptable such as adding more advanced sensors. It can be enhanced to detect and collect data of several anomalies for monitoring purpose such as home ultrasound, Brain signal monitoring, Tumor detection etc.The system can be extended from single patient monitoring to Multi patient monitoring system , so that many patients can be monitored by a caretaker .

The system can be extended by adding more features like linking the ambulance services, leading doctor's list and their specialist , hospitals and their special facilities and nearest hospital to the patient location through accessing google map facility etc. A two way communication can be availed that the Doctors can create awareness about diseases and their symptoms through the mobile application . Automatic calling can be implemented instead of alerts/messages. The IoT device can be combined with the cloud computing so that the database can be shared in all the hospitals for the intensive care and treatment. Also a video feature can be added for face to face consultation between the doctors and patients.

Chapter 6

APPENDIX

6.1 Source Code

6.1.1 Arduino.ino

```
#include "Wire.h"
#include <MPU6050_light.h>
#include <DFRobot_DHT11.h>
DFRobot_DHT11 DHT;
int dht = 4;
int count=0;
MPU6050 mpu(Wire);
void setup () {
  Serial.begin(9600);
 pinMode(dht, INPUT);
  Wire.begin();
  byte status = mpu.begin();
  Serial.print(F("MPU6050 status: "));
  Serial.println(status);
  while(status!=0){ }
 // stop everything if could not connect to MPU6050
```

```
  Serial.println(F("Calculating offsets,
  do not move MPU6050"));
  delay(1000);
  // mpu.upsideDownMounting = true;
  // uncomment this line if the MPU6050 is
  mounted upside-down
  mpu.calcOffsets();
  // gyro and accelero
  Serial.println("Done!\n");
}

void loop() {

   mpu.update();

DHT.read(dht);
int dhtvalue = DHT.temperature;
Serial.print("X");
Serial.print(dhtvalue);
Serial.println("#");
  Serial.print("accerelometer : ");
  Serial.print(abs(mpu.getAngleX()));
  Serial.println("#");

//Serial.print("X : ");
//Serial.print(dhtvalue);
//Serial.println("#");
```

```
    Serial.print("Y");
   Serial.print(abs(mpu.getAngleX()));
   Serial.println("#");

   if (abs(mpu.getAngleX()) > 50)
   {
   Serial.println("FALL");
   }
}
```

6.1.2 Node.ino

```
#include <Wire.h>
#include "MAX30105.h"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
SoftwareSerial myserial(D7,D8);
SoftwareSerial Myserial(D3,D4);
#include "heartRate.h"
static const int RXPin = 4, TXPin = 5;
// GPIO 4=D2(conneect Tx of GPS) and
GPIO 5=D1(Connect Rx of GPS
static const uint32_t GPSBaud = 9600;
//if Baud rate 9600 didn't work in your
case then use 4800
float spd;
float sats;
unsigned int move_index = 1;
TinyGPSPlus gps;
// The TinyGPS++ object
WidgetMap myMap(V0);
#include "heartRate.h"
MAX30105 particleSensor;
char auth[] = "Hkutsq2mDk7HIh75bl9advim-ab-P6Bq";
char ssid[] = "JioFi_245E218";
char pass[] = "g9r6ca6s7i";
int x, y, z, k,l,i,t,h,m,d,a;
String str, temp,lab1,lab2,lab3,lab4,lab5,lab6,lab7;
```

```
int xflag, yflag, zflag, bflag;
const byte RATE_SIZE = 4;
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatsPerMinute;
int beatAvg;
int avgbpm;
int slot,slot1;
BLYNK_WRITE(V3)
{
  slot = param.asInt();
}
BLYNK_WRITE(V4)
{
  slot1 = param.asInt();
}
void setup()
{
  Serial.begin(9600);
myserial.begin(9600);
Myserial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  Blynk.notify("NodeMCU connected to Blynk !");
Serial.println("Initializing...");
 if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
  {
    Serial.println("MAX30105 was not found.
    Please check wiring/power. ");
```

```
    while (1);
  }
  Serial.println("Place your index finger
  on the sensor with steady pressure.");
  particleSensor.setup(); /
  particleSensor.setPulseAmplitudeRed(0x0A);
  particleSensor.setPulseAmplitudeGreen(0);
}
void loop()
{
  Blynk.run();
  while (myserial.available() > 0)
    {
     if (gps.encode(myserial.read()))
        displayInfo();
  }
 long irValue = particleSensor.getIR();
   if (checkForBeat(irValue) == true)
  {
    long delta = millis() - lastBeat;
    lastBeat = millis();
    beatsPerMinute = 60 / (delta / 1000.0);
    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      rates[rateSpot++] = (byte)beatsPerMinute;
      rateSpot %= RATE_SIZE; //Wrap variable
      beatAvg = 0;
      avgbpm = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
```

```
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
    }
    for (byte i = 0 ; i < bufferLength ; i++)
  {
    while (particleSensor.available() == false)
      particleSensor.check();
    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample();
    so move to next sample
    Serial.print(F("red="));
    Serial.print(redBuffer[i], DEC);
    Serial.print(F(", ir="));
    Serial.println(irBuffer[i], DEC);
  }

  //calculate heart rate and SpO2 after
  first 100 samples (first 4 seconds of samples)
  maxim_heart_rate_and_oxygen_saturation
  (irBuffer, bufferLength, redBuffer, &spo2, &
  validSPO2, &heartRate, &validHeartRate);
  //Continuously taking samples from MAX30102.
Heart rate and SpO2 are calculated
every 1 second
  while (1)
  {
    for (byte i = 25; i < 100; i++)
    {
```

```
      redBuffer[i - 25] = redBuffer[i];
      irBuffer[i - 25] = irBuffer[i];
    }

    //take 25 sets of samples before calculating
    the heart rate.
    for (byte i = 75; i < 100; i++)
    {
      while (particleSensor.available() == false)
      //do we have new data?
        particleSensor.check();
        //Check the sensor for new data

      digitalWrite(readLED, !digitalRead(readLED));
      //Blink onboard LED with every data read
      irBuffer[i] = particleSensor.getIR();
      particleSensor.nextSample();
      //We're finished with this sample so
      move to next sample

//send samples and calculation result to
terminal program through UART
      Serial.print(F("red="));
      Serial.print(Spo2, DEC);

  }
    Serial.print(", Avg BPM=");
  Serial.print(beatAvg);
  avgbpm=beatAvg;
```

```
//Serial.print(avgbpm);
//Blynk.virtualWrite(V0,avgbpm);
 //Blynk.virtualWrite(V1,random(91,100));
if ((avgbpm >= 40)and(avgbpm <=100)
and irValue > 50000)
{
  Serial.println("BPM : ");
  Serial.print(avgbpm);
  Serial.print("<<<==========>>>");
  Blynk.virtualWrite(V0,avgbpm);
  delay(1000);
}

if((spo2 >= 80) and (sp02 <= 100))
{
  Serial.print("SP02 : ");
  Serial.print(sp02);
  Serial.print("<<<==========>>>");
  Blynk.virtualWrite(V1,sp02);
}
if (myserial.available())
{
  str = myserial.readStringUntil('#');
  Serial.print("str = ");
  Serial.println(str);

  //temperature
  if (str[0] == 'X')
  {
```

```
        Serial.print("t = ");
        i = 1;
        temp = "";
        while (str[i] != '\0')
        {
          temp = temp + str[i];
          i++;
        }
        Serial.println(temp);
        t = temp.toInt();
        Serial.print("TEMP : ");
        Serial.println(t);
        Serial.print("<<<==========>>>");
        Blynk.virtualWrite(V2,t);
       // Blynk.virtualWrite(V0, t);

        if(t > 36)
        {
            Serial.print("<<<==========>>>");
   Serial.print("Temperature Increased");
   Blynk.notify("Temperature Increased");
   Serial.println("<<<==========>>>");
emergency1();
delay(20000);
emergency2();
delay(20000);
emergency3();
        }
      }
```

```
    //humidity
    if (str[0] == 'Y')
    {
      Serial.print("h = ");
      i = 1;
      temp = "";
      while (str[i] != '\0')
      {
        temp = temp + str[i];
        i++;
      }
      Serial.println(temp);
      h = temp.toInt();
      Serial.print("H = ");
      Serial.println(h);
      Blynk.virtualWrite(V5, h);
      if(h>500)
      {
        Serial.print("<<<===========>>>");
  Serial.print("Fall detected");
  Blynk.notify("Fall detected");
  Serial.println("<<<===========>>>");
      }
    }
}
if(slot == 1)
{
  lab1 = "Name: SANKARANTAMBI";
```

```
  lab2 = "Age : 82";
  lab3 = "Cardiovascular : High blood pressure,
  Heart Attack, Dizziness";
  lab4 = "Respiratory: Asthma, Emphysema";
  lab5 = "Blood Group : A+";
  lab6 = "Doctor's Name: Dr. Sunny";
  lab7 = "Phone number: 4466 ";
  Blynk.virtualWrite(V5, lab1);
  Blynk.virtualWrite(V5, lab2);
  Blynk.virtualWrite(V5, lab5);
  Blynk.virtualWrite(V5, lab3);
  Blynk.virtualWrite(V5, lab4);
  Blynk.virtualWrite(V5, lab6);
  Blynk.virtualWrite(V5, lab6);

}
if(slot1 == 1)
{
  Blynk.notify("Message sent");
SendMessage();
}
}
void SendMessage()
{
  Serial.println("To Hospital");
  Myserial.println("AT+CMGF=1");
  delay(1000);
   Serial.println("Connecting to +919961591209");
  Myserial.println("AT+CMGS=\"+919961591209\"\r");
```

```
//Myserial.println("AT+CMGS=\"+919544706072\"\r");
  delay(1000);
   Serial.println("Connecting to Dr. Sunny");
  Myserial.println(" Hi Dr.Sunny, Your patient is
  in critical situation. Please go through
  the details\n
  --- Name: SANKARANTAMBI -\n Age : 82\n -
  Cardiovascular : High blood pressure,
  Heart Attack, Dizziness -
  Respiratory: Asthma, Emphysema -\n
  Blood Group : A+ - \n
  Location :- http://www.google.com/maps/place/");
  // The SMS text you want to send
  Myserial.println(Latitude, Longitude);
  delay(100);
   Serial.println("Message sent to Dr. Sunny");
   Myserial.println((char)26);
   // ASCII code of CTRL+Z
  delay(1000);

}

void emergency1()
{
  Serial.println("First Person");
  Myserial.println("AT+CMGF=1");
  delay(1000);
   Serial.println("Connecting to +919747100657");
 Myserial.println("AT+CMGS=\"+919747100657\"\r");
```

```
  // Myserial.println("AT+CMGS=\"+919544706072\"\r");
 delay(1000);
  Serial.println("Connecting to Adhitya");
 Myserial.println(" Hi Adhitya, Your Father
 Sankaranthambi needs your help.FALL DETECTED.
 Please visit him asap.");
 delay(100);
  Serial.println("Message Sent to Adhitya");
  Myserial.println((char)26);
}

void emergency2()
{
 Serial.println("Second Person");
 Myserial.println("AT+CMGF=1");
 delay(1000);
  Serial.println("Connecting to +917736510228");
 Myserial.println("AT+CMGS=\"+917736510228\"\r");
 delay(1000);
  Serial.println("Connecting to Afeefa");
 Myserial.println("Hi Afeefa, Your neighbour
 Sankaranthambi needs your help.FALL DETECTED.
 Please visit him asap.");
  Serial.println("Message sent");
  Myserial.println((char)26);

}

void emergency3()
```

```
{

  Serial.println("Third Person");
  Myserial.println("AT+CMGF=1");
  delay(1000);
   Serial.println("Connecting to  +919847587202 ");
  Myserial.println("AT+CMGS=\"+919847587202\"\r");
  delay(1000);
   Serial.println("Connecting to Abel");
  Myserial.println("Hi Abel, Your Uncle
  Sankaranthambi needs your help.FALL DETECTED.
  Please visit him asap.");
  delay(100);
   Serial.println("Message sent to Abel");
   Myserial.println((char)26);
  delay(1000);

}
void checkGPS(){
  if (gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
     // Blynk.virtualWrite(V4, "GPS ERROR");
     // Value Display widget  on V4 if GPS not detected
  }
}

void displayInfo()
{
```

```
  if (gps.location.isValid() )
  {
    float latitude = (gps.location.lat());
    //Storing the Lat. and Lon.
    float longitude = (gps.location.lng());

    Serial.print("LAT:  ");
    Serial.println(latitude, 6);
    // float to x decimal places
    Serial.print("LONG: ");
    Serial.println(longitude, 6);

    myMap.location(move_index, latitude,
    longitude, "GPS_Location");
    spd = gps.speed.kmph();
    //get speed
//          Blynk.virtualWrite(V3, spd);
//          sats = gps.satellites.value();
        //get number of satellites
       // Blynk.virtualWrite(V4, sats);

bearing = TinyGPSPlus::cardinal(gps.course.value());
// get the direction
        //Blynk.virtualWrite(V5, bearing);
}
 Serial.println();
}
```

6.2 Screen Shots

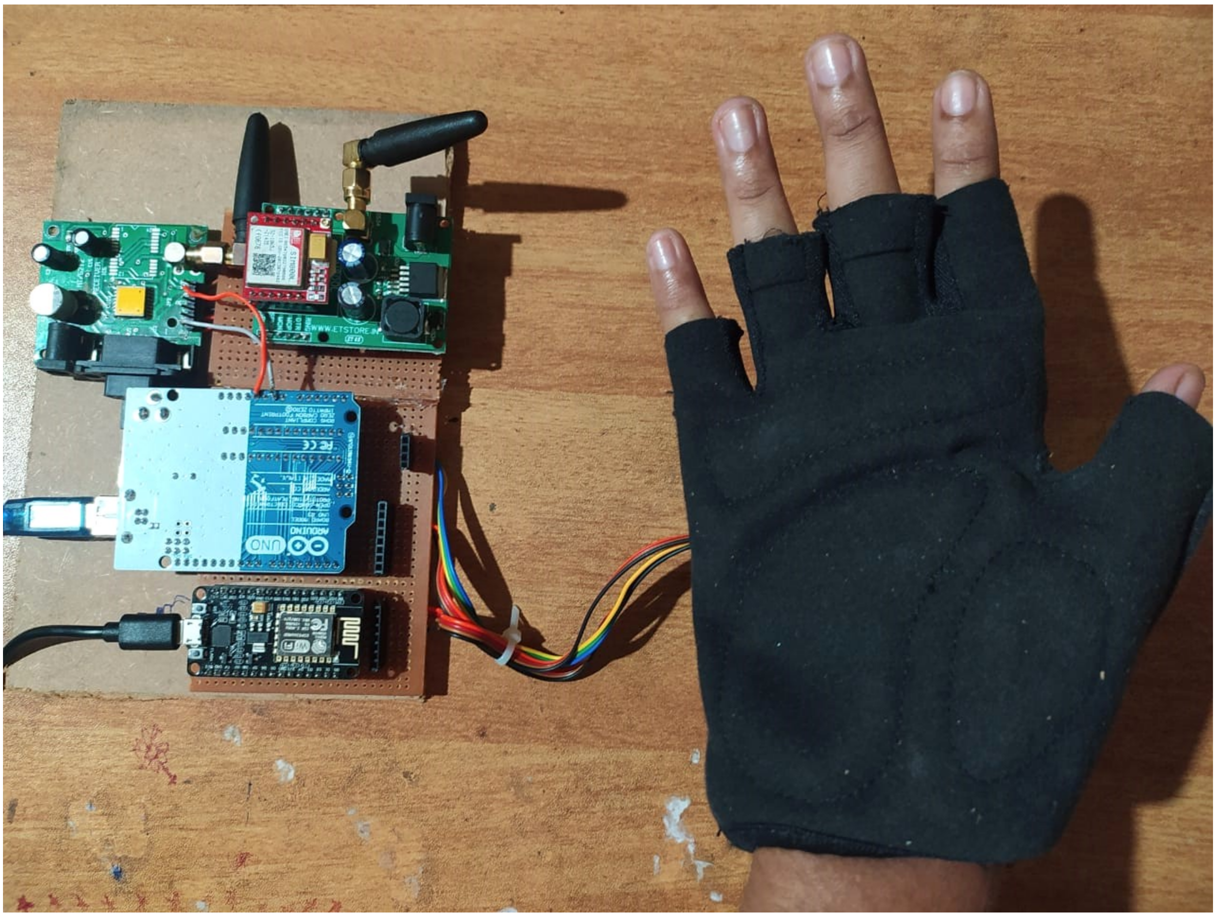


Figure 6.1: Health Monitoring- Wearable device

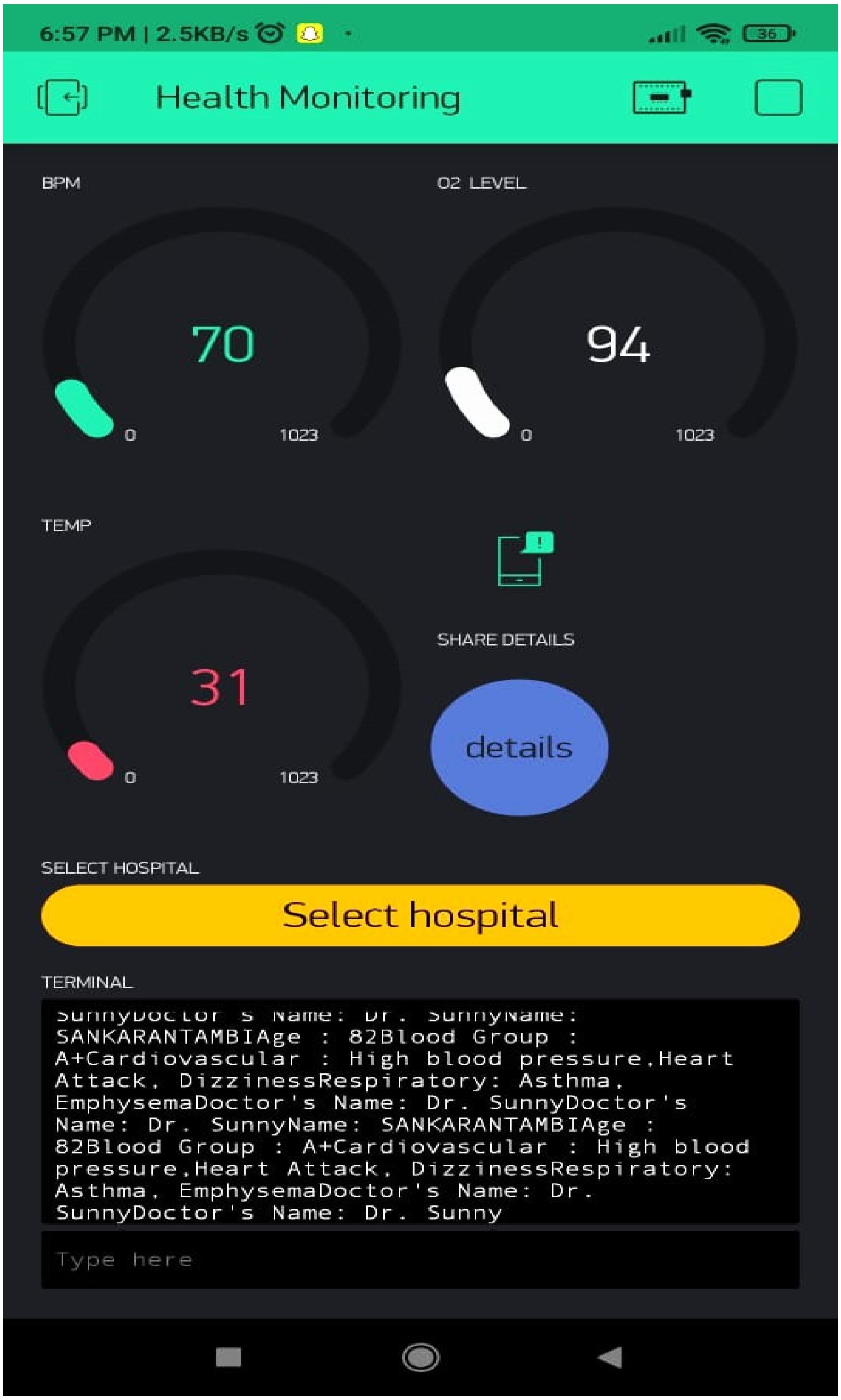


Figure 6.2: Health Monitoring App

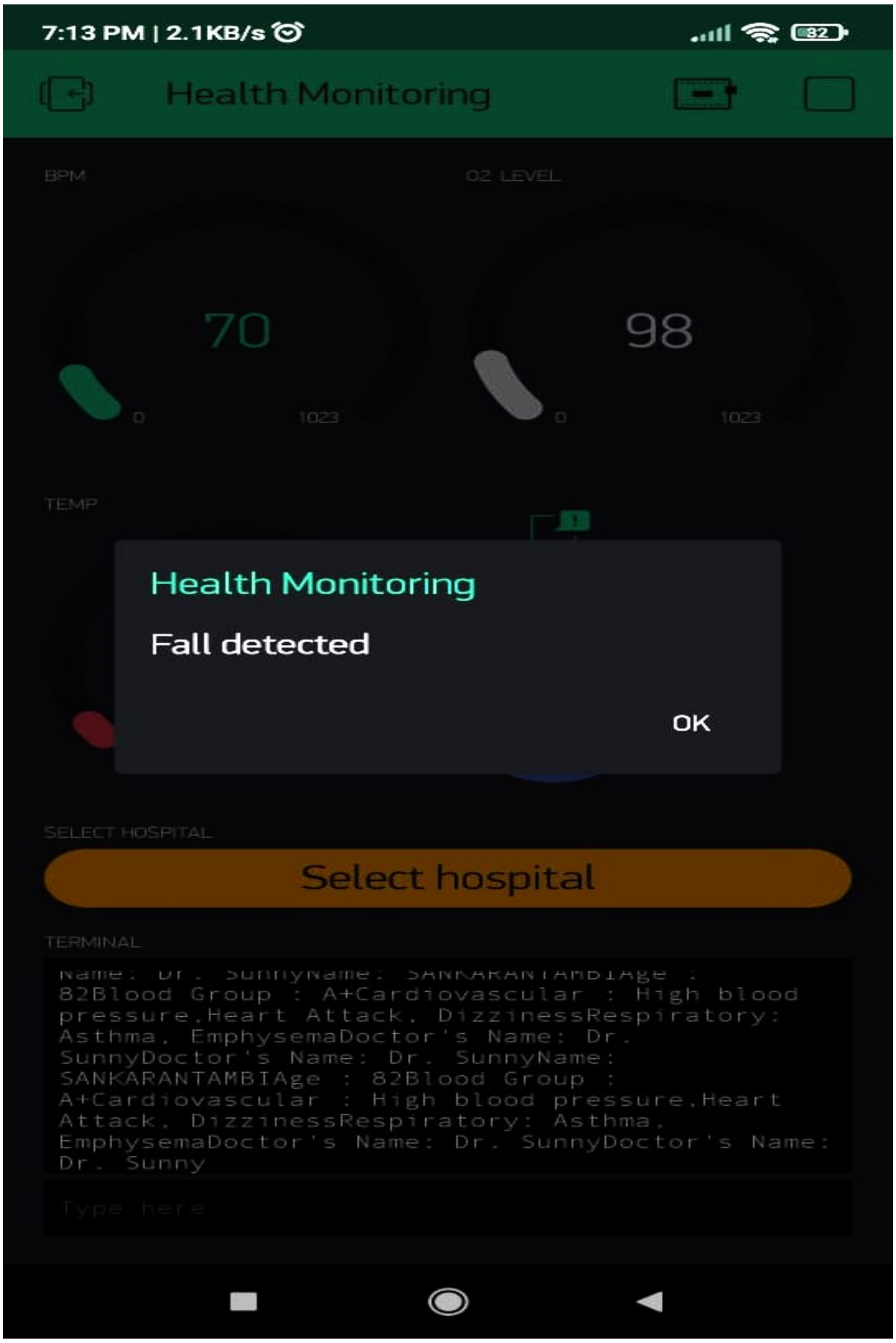


Figure 6.3: Fall Detected - Notification

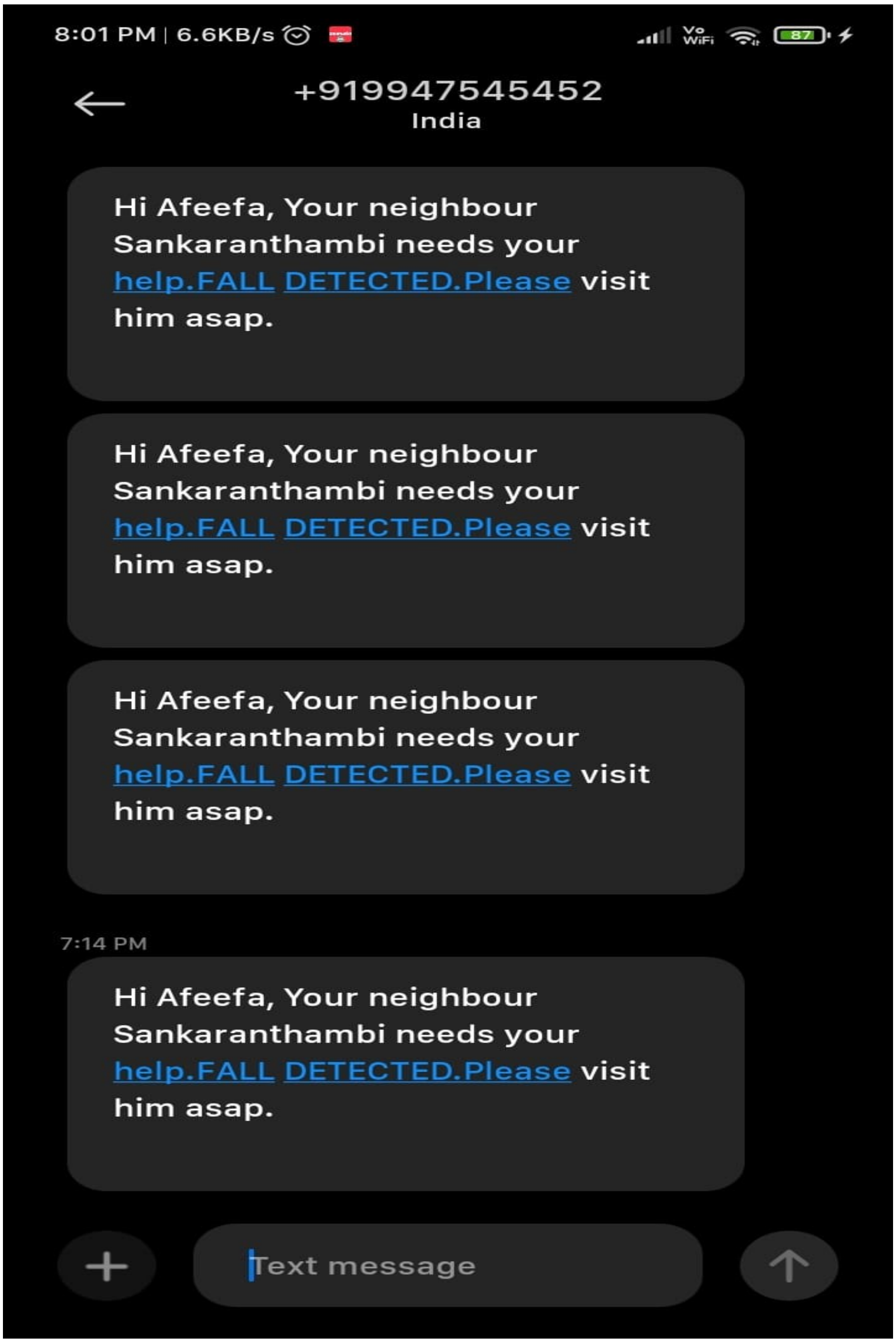


Figure 6.4: Alert to Emergency contact-1

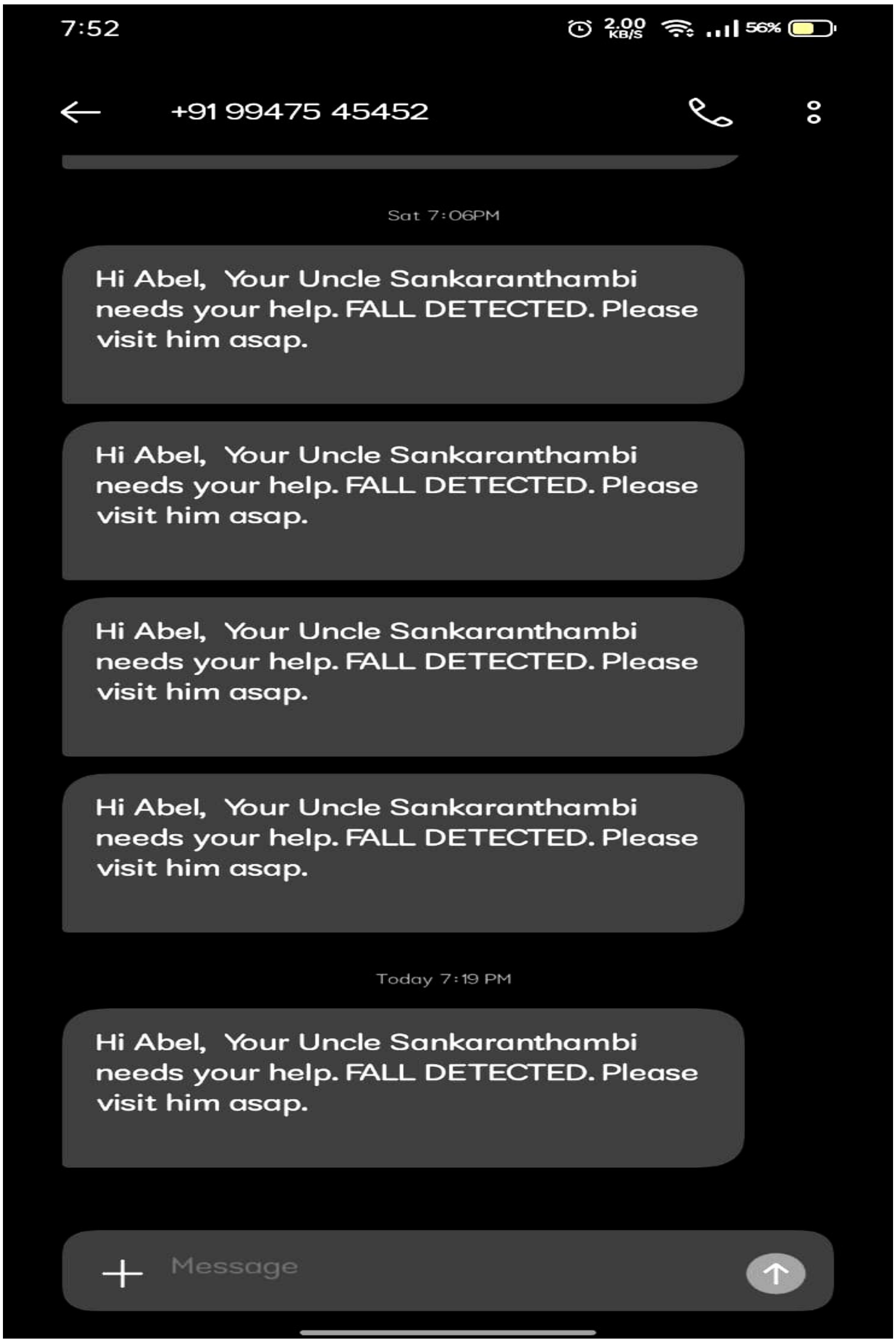


Figure 6.5: Alert to Emergency contact-2

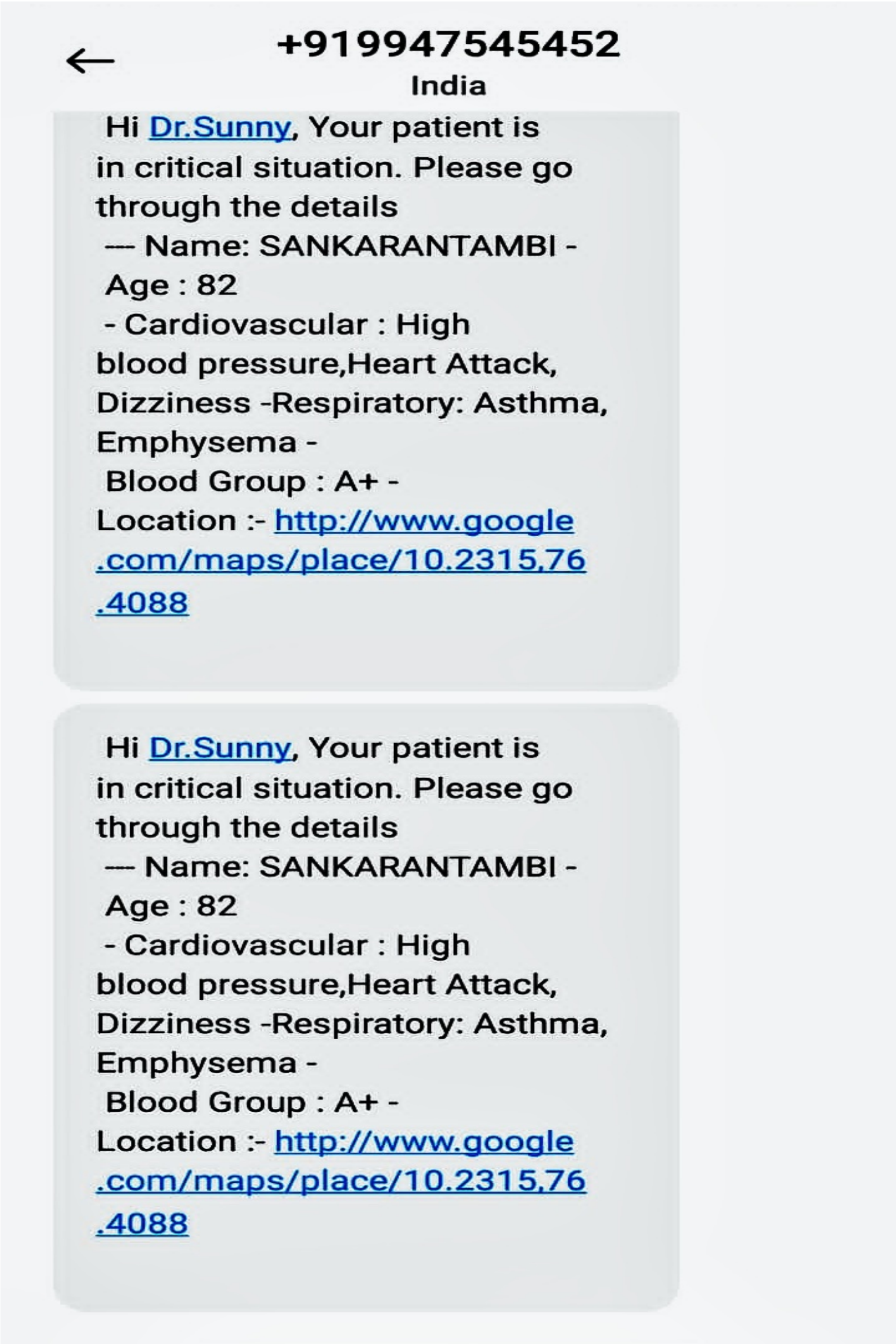


Figure 6.6: Alert to the Hospital

Chapter 7

REFERENCES

[1] S. M. Riazul Islam, (Member, IEEE), Daehan Kwak, MD. Humaun Kabir, Mahmud Hossain, And Kyung-Sup Kwak, (Member, IEEE) “The Internet of Things for Health Care: A Comprehensive Survey” 2015.2437951 VOLUME 3, 2015

[2] Punit Gupta, Deepika Agrawal, Jasmeet Chhabra, Pulkit Kumar Dhir Jaypee University of Information Technology Himachal Pradesh, India “IoT based Smart HealthCare Kit” ICCTICT.2016.7514585 March 2016

[3] Luca Catarinucci, Danilo De Donno, Luca Mainetti, Luca Palano, Luigi Patrono, Maria Laura Stefanizzi, and Luciano Tarricone “An IoT related Architecture for Smart Health Care Systems” 2327-4662 2015 IEEE

[4] M. Akiyama and R. Nagai, “Information technology in health care: E-health for Japanese health services,” A Report of the CSIS Global Health policy center, The Center for Strategic and International Studies and the Health and Global Policy Institute, Mar. 2012.

[5] Pioggia, ”Personal Health System architecture for stress monitoring and support to clinical decisions”, Computer Communications Vol.35, pp.1296-1305, 2017.

[6] ”Improvement and Enhancement in Emergency Medical Services using IOT” Kajal R.K Pandey 1, Ketan Arwat2, Ishaan Sharma3, Prof. Sonali Patil4 1,2,3SRTTC’s Faculty of Engineering, Pune University-Feb-2018

[7] ”Wireless Sensor Network Application for IoT based HealthCare System” Gurram Dheeraj Reddy, Shashank Chutke2, M.Sri Venkat Rami Reddy3, Dr. D. Nageswara Rao -Department of Electronics and Communication Engineering, Hyderabad, India.- February 2018, Volume 5, Issue 2

[8] ArunaDevi.S et al. ”SMART HUMAN HEALTH MONITORING SYSTEM BY USING IoT/ International Journal of Computer Science Engineering Technology (IJCSET) ISSN,2017