BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

HAND GESTURE CONTROLLED WHEELCHAIR AND HEALTH MONITORING SYSTEM

Submitted By

Zakia Sultana **ID: 666-49-15**

Qazi MD Atiqul Islam **ID: 666-49-11**

Supervised By

Mohammad Jahangir Alam
Assistant Professor
Department of Computer Science & Engineering
Southern University Bangladesh

Submitted to

Department of Computer Science & Engineering

Southern University Bangladesh

Arefin Nagar, Bayazid, Chattogram

HAND GESTURE CONTROLLED WHEELCHAIR AND HEALTH MONITORING SYSTEM

Submitted By: Zakia Sultana ID: 666-49-15

Qazi MD Atiqul Islam ID: 666-49-11

This project submitted to the Department of Computer Science & Engineering, Southern University Bangladesh in partial fulfillment of the requirement of the degree of Bachelor of Science in Computer Science & Engineering.

Board of Examiners

_____

Md. Jahangir Alam **Supervisor**
Assistant Professor
Department of Computer Science & Engineering
Southern University Bangladesh

_____

Md. Jahangir Alam
Head **Head**
Department of Computer Science & Engineering
Southern University Bangladesh

_____

Professor Dr. Md. Abul Mansur Chowdhury
Dean **Dean**
Faculty of Science & Engineering
Southern University Bangladesh

_____

Professor Dr. Mohammad Osiur Rahman **External**
Professor
Department of Computer Science & Engineering,
University of Chittagong

ACKNOWLEDGEMENT

We are thankful to Allah, the almighty for giving us blessings and mercy to work hard for completing this project. We are thankful to our honorable teachers at the Department of Computer Science & Engineering, Southern University Bangladesh for their cordial help and support throughout honoree course here.

We would like to express our special thanks of gratitude to our Supervisor Mohammad Jahangir Alam who gave us the golden opportunity to do this wonderful project on the topic Hand Gesture Controlled Wheelchair and health monitoring system. Which also helped us to do a lot of research and we came to know about so many new things, we are really thankful to them.

Lastly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

ABSTRACT

Patients with various physical disabilities use wheelchairs to help them move and easily meet their daily needs. But there are some cases where the movement of a wheelchair is dependent on another person as is the case with patients who lack the required arm strength and movement to properly push the wheels forward such as quadriplegics, paraplegics, stroke patients, elders etc .The automated wheelchairs an easy transportation for these physical disabled persons.

This project is an effort to help people with disabilities to move independently using Gesture Control System. This project is designed using the hardware and with the help of a simple hand gesture which includes an accelerometer (MPU6050) mounted on the hand glove that detects the angle of movement of the user's hand and transmits control signal to the receiver mounted on the wheelchair. This will translate the movement in the way the user needs. The wheelchair is designed to allow people with disabilities to travel safely and to maintain integrity in performing certain essential tasks in daily life; they can drive the wheelchair left, right, forward, backward and give emergency signal; we also developed a wireless system which gives us information about patients temperature, fall occurrence, pulse condition and oxygen level monitoring using a Wi-Fi module. By this we can provide real time information of the patients to the relatives or doctors over the internet. To give the patients proper and timely help, continuous monitoring of the patient is needed but these continuous monitoring systems are only available in hospitals ICU and that’s also monitored while the patient is on bed. This system will help the patients who are not in critical condition but their continuous health monitoring is needed by a doctor or family and the patients can also go around as they wish without any support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The number of handicapped/movement disabled people in Bangladesh is increasing due to the incremental rate of accidents. About 650 million people are suffering from several types of physical disabilities. A wheelchair is such a type of mechanical device, which is used by the elderly patients and the people who are physically unable to move properly due to some kind of accident or neural disease. Most of the electric wheelchairs used in Bangladesh are imported. Moreover, these wheelchairs are above the level of affordability of most common class people here in Bangladesh due to higher cost. Several works have been done in the wheelchair design project with various techniques to design low cost version. But those techniques are not efficient enough and not much affordable in terms of cost and complexity.

Some patients cannot use the wheelchair with their arms, because they cannot move utmost of the body corridor. This is where the demand for automated wheelchairs arises. In this city culture, the elderly persons are hospitalized due their health condition and there is no one to take care of them. When they are in home, one person is required to take care of them personally. Our project is very much helpful to those persons, since it helps in monitoring their health and informs the care taker. The body sensor network (BSN) technology is one of the recent technologies of IOT developments in the Health Monitoring system, where a patient can be monitored using a collection of tiny powered and lightweight wireless sensor nodes. At this generation, heart disease is one of serious diseases that may threaten human life. The electrocardiogram (ECG) plays major role in the prevention, diagnoses the abnormality of patients and rescue of heart disease. In progress has been made in the development of a remote monitoring system for ECG signals, the deployment of IoT services over communication network with new applications. We, therefore, are modelling a system having several wireless sensors which will measure health related information like body temperature, heart beat rate, ECG & fall detect, etc. and transmit over internet to be accessed by other user at remote location. This system also provide 3 times medicine alert & also give saline nil level alert.

This project is to develop a wheelchair which is useful and affordable to the physically disabled/challenged person using hand movement or by using hand gesture reorganization. With the help of the wheelchair, the physically disabled person can move himself to the desired location with the help of hand gestures that control the movement of the chair. This project aims to provide a feasible solution to those handicapped/disabled people who do not have the ability to move the wheelchair by themselves.

2. MOTIVATION

Individuals who are crippled or somewhat paralyzed regularly have trouble getting in and out of their homes without help. Usually individuals ought to utilize wheelchairs to move these persons after loss of motion or other physical feebleness. However, at whatever point they have to move around the house with the assistance of somebody else, that person's assurance can drop for asking for help all the time. Also people cannot always monitor their health in hospitals, which will cost a fortune if anyone wants to monitor their health all the time. The main point of this project was to plan a framework that will utilize existing innovation to a grandstand that it can advantage the people with inabilities and notify their relatives about their health over the internet.

3. OBJECTIVE

- Patients can freely move with the help of wheelchair.
- Patients can contact any person when in emergency.
- They will be self-independent.
- Caretaker/relatives will get live updates.

4. OVERVIEW OF THE PROJECT

Chapter 1: In this chapter, we have discussed introduction, motivation and organization of the project.

Chapter 2: In this chapter, we have discussed some topic related with application and preview related works of development.

Chapter 3: In this chapter, we have discussed about the requirement analysis, hardware requirement, System diagram and block diagram.

Chapter 4: In this chapter, we have discussed about all kind of hardware components that will be used in this project, circuit diagram.

Chapter 5: In this chapter, we have discussed all pages of the apps which show the expected result in brief.

Chapter 6: In this chapter, we have discussed the conclusion, limitation and future working scope of the system.

5. SUMMARY

In this chapter, we have discussed introduction, motivation and organization of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Unlike the 19th century challenges that people use to face with the nuances of underdevelopment and lack of facilities, today with the advancements in technology, newer opportunities and facilities have come up in a higher pace to serve the elderly population in various innovative ways.

When an unfortunate event affects the motor capacity of a person, it's necessary to use devices like wheelchairs that offer a means of transportation for patients with motors problems of the lower limbs. Tremendous changes have been made in the field of wheelchair technology. Still, indeed these significant advances haven't been suitable to help quadriplegics navigate wheelchair unassisted.

There are some limitations existing with traditional wheelchair system such as complexity in service when repaired, use of joystick may be difficult for challenged persons, the wheel chair cannot sense any obstacles' on the path, and to move the chair they need some external help. The limitations of the existing systems can be eliminated in this present system.

2.2 SOME RELATED WORKS

There have been several researches and developments on the hand gesture wheelchair system. In this system W.Zgallai et al. [1] worked on a smart wheelchair that will be able to move based on EEG and spectrum and they became fruitful to construct their model. They tried the model smart wheelchair utilizing Emotive EPOC headset that will offer assistance the blind and paralyzed individuals who are incapable to control parts of their body. It uses deep learning in order to recognize four distinctive movements from the recorded EEG signal, to move left, right, forward, and halt. Information from 10 volunteers appeared a success rate of 70% from the row EEG and 96% from the spectrum of the information from the frequency bins comparing to the delta, theta, alpha, beta, and gamma waves. Shayban et al. [2] later proposed a head gesture device using a sensor for the acceleration where the wheelchair is regulated by a joystick which is hard to control for people with disabilities. By changing the head, data is transmitted wirelessly to the micro-controller, and the wheelchair controls movement. In addition, an electronic wheelchair based on android has been suggested by R. J. Leela et al [3] for physically disabled people where the smartphone gives voice commands, and voices are translated to text. The microcontroller then receives texts and the proposed system movement is controlled using Bluetooth module with the aid of DC motors. In another paper, for disable and elderly people a RoboChair has been built by Gray et al. [4] which controls the RoboChair motion utilizing DSP motion controller so that it can control the motion of the RoboChair according to the user's intention.. The fundamental reason for

this exploration was to make an easy to understand environment for old individuals and individuals with incapacities P.R. Kumar et al. worked on a smart wheelchair that can be controlled by voice command [5]. Their Automated wheelchair studies the arrangement of surrounding area through a recited, directed journey that a person enters. Eventually, the chair will move to any formerly-updated position underneath the voice instruction. This traditional wheelchair will be helpful to people who don't have control of the lower part of their body. R. Alkhatib et al developed a machine learning-based wheelchair to support its user [6]. Here navigation was controlled by a computer vision system. I.G. Torres et al. worked on a cost efficient wheelchair to support peoples as though smart wheelchairs are available but most people cannot afford it. Their affordable Smart Wheelchair is for indoor use. This kit implements full-autonomy in the form of indoor navigation from one room to another and to predetermined docking locations through voice control. It also has semi-autonomous functions in the form of manual joystick control augmented with real-time collision avoidance and staircase detection. [7]. Z. Fan et al. worked on motion predictors for wheelchairs using autoregressive sparse Gaussian Process.[8]. They presented an integrated equipment and software for foreseeing the movement of a commercial PWC stage that does not require any physical or electronic alteration of the chair beyond plugging into an industry standard assistant input port. This framework uses an RGB-D camera and an Arduino interfaced board to capture motion information, counting visual odometry and joystick signals, through ROS communication. Future movement is anticipated utilizing an autoregressive sparse Gaussian process model. U. Garg et al. designed a special wheelchair for quadriplegia patients [9]. Using this IoT device people will be able to control the wheelchair using RF. To provide movement of wheelchair in wanted direction an accelerometer based hand glove is designed. Patient can transmit the requirements to the closer ones through hand motion utilizing RF transmission. Third mode gives the data on cloud which can be accessed by the specialist for determining problems. Rajesh Kannan Megalingam et al. researched on paper titled "Sensor-Based Gesture Control Wheelchair" [10] it works on the principle of gesture recognition by using Infrared Sensors. In this method, IR sensors are used for identifying the simple gestures to control the powered wheelchair to move in any direction. In the proposed prototype system, a gesture pad that includes IR sensors, MCU and power management circuit is designed for gesture recognition and identification and a controller for driving motors is implemented. The main problem that comes with IR is during daylight, its sensitivity is reduced, and hence causing problem for processing the further programs. And it is difficult to recognize exact gestures using IR sensors. M.R.R.M. Akanda et al. [11] built voice-controlled smart assistant and real-time vehicle detection system to make life easier for the blind peoples. This framework is completely voice controlled. Blind individuals can know the current area and can travel by walk and via transport to diverse places by utilizing this framework. They will get continuous instruction through speech which can guarantee the benefit flawlessly. Z. Jiao et al. designed a wheelchair that will be able to plan its path using adaptive polymorphic ant colony algorithm [12]. Using this technology, the wheelchair itself will be able to create its own path and find the shortest path for its user. The target path planning and obstacle path planning are respectively carried out by using the adaptive polymorphic ant colony algorithm.

2.3 SUMMARY

In this chapter, we have discussed some topic related with application and preview related works of development.

CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

The software requirement is description of features and functionalities of the target system. Requirement conveys the exception of users from the software product. The requirement can be obvious or hidden, know or unknown, expected or unexpected from client's point of view.

3.2 Methodology

This project was implemented concentrating to design a power wheelchair which has both control systems for the disabled having just lower limb or both lower limb and upper limb injury as well as provide some therapy facilities to support them to gain their stain back. So the whole system is divided into two portions. First one is controlling portion whereas the other is panic portion for seeking help in need.

3.3 System design

Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

3.4 Coding

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs of the UNO board in a smaller form factor.
The Arduino Nano is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline.
Install the Arduino Desktop IDE

To get step-by-step instructions select one of the following link accordingly to your operating system.

- Windows
- Mac OS
- Linux

- Portable IDE (Windows and Linux)
- ChromeOS (Arduino Create App) in the Chrome Web Store

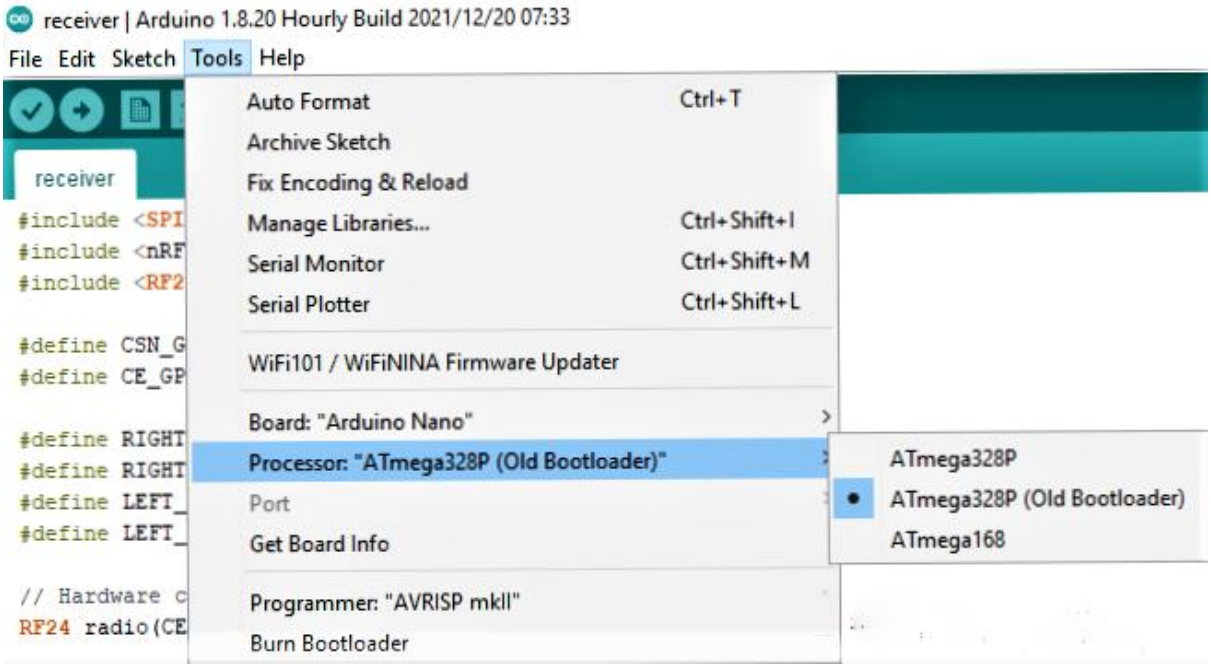

Figure 3.1: Arduino coding interface

3.5 System Diagram

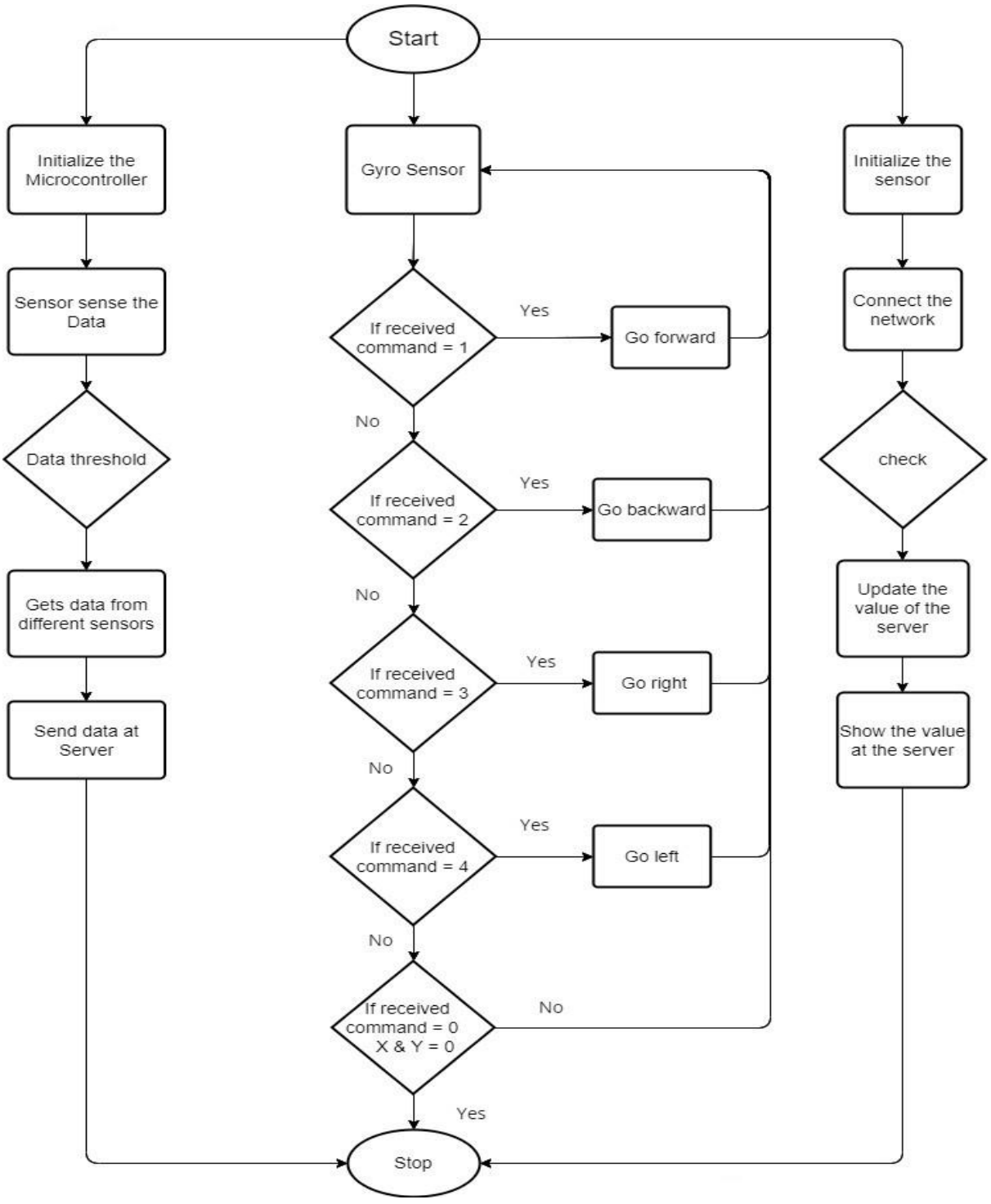

Figure 3.2: System Block Diagram

In the above figure 3.2, the whole system is shown through the diagram. From this we can see that when the system starts the gyro, other microcontrollers and sensors are initialized and after checking data from the sensors send to the server and after collecting the data the value is shown and the gyro works as per the instruction given by its user.

3.6 Block diagram

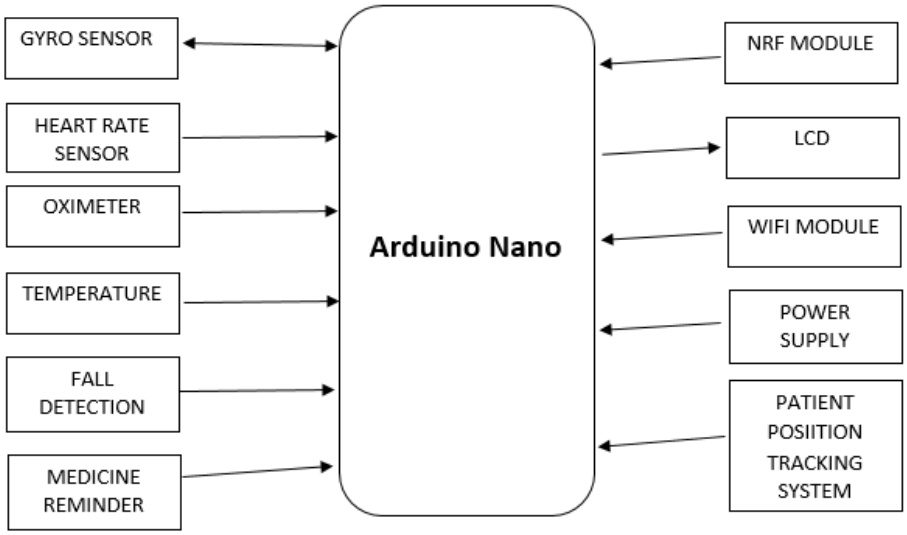

Figure 3.3: Transmitter Block Diagram

In the above figure 3.3, the gyro sensor sends the information about the movement; heart rate sensor, oximeter, temperature sends information by sensing the patient's nerves to the Arduino Nano. Fall detection works when the patient falls from the wheelchair. Medicine Reminder is about when a caretaker/relative inputs the preferred time for medicines and it displays in LCD. NRF module gives signals to Nano so that it connects with the receiver system. The WiFi module connects with WiFi and via WiFi the Nano displays other relevant information in LCD. Power supply is needed for the system to run. The patient position tracking system works for the location of the patient. All this information is stored temporarily in Nano.

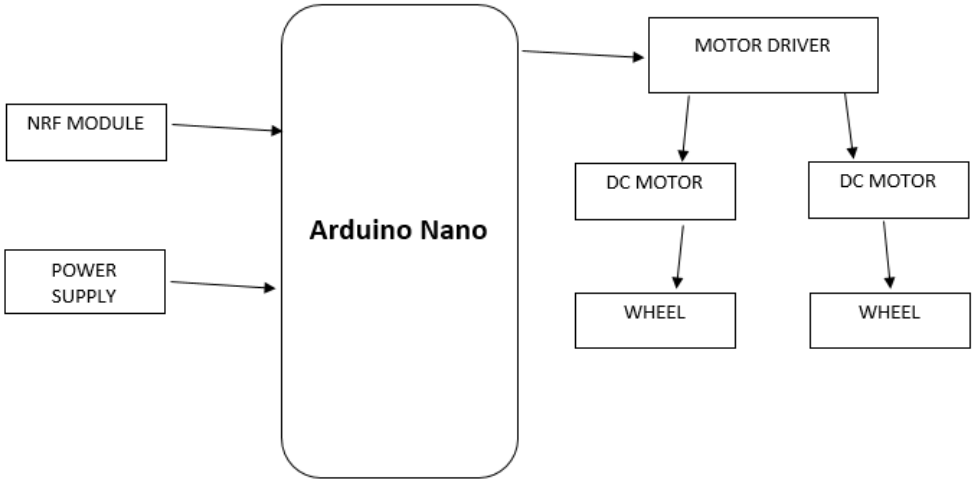

Figure 3.4: Receiver Block Diagram

In the above figure 3.4, once the power supply is switched the system starts operating. The NRF Module receives movement signals from the transmitter and passes it to Arduino Nano by which the motor drivers starts moving according to the direction signaled.

3.7 Necessity to use a diagram

- To develop understanding of how a process is done
- To study a process for improvement
- To communicate to others how a process is done
- When better communication is needed between people involved with the same process
- To document a process
- When planning a project

3.8 Summary

In this chapter, we have discussed about the requirement analysis, hardware requirement, block diagram.

CHAPTER 4
IMPLEMENTATION

4.1 INTRODUCTION

In these modern days, everyone has a mindset that they must be independent in all forms of life rather depending on other people and asking help from others. The fantastic revolution in technology has broadened up the chances of possible development in any field of progressive studies. Apart from the developments in hospital equipment's and pharmaceutical products, new catalyst in technological innovations like the internet has become a feasible platform for the elderly citizens. Wellbeing and extreme care can be manifested through applications on smart devices. The connectivity and communication gaps are bridged through these creative applications in a superfluous fashion. Now a days, many physically challenged and elderly people need wheelchair to make them easily assist in mobility from one place to another. So, they require an automated wheelchair to help in their mobility rather using the manual wheelchair which required someone to propel it. These disadvantages of manual wheelchair can be overcome by using automated wheelchair which is the electric wheelchair controlled using the joystick. Generally, these automated wheelchairs cost ranges from 60000 and more. So, we have developed a wheelchair with hand gestured recognition system which provides an advantage of mobility assistance to physically challenged people at low cost than conventional wheelchair available in the market.

4.2 COMPONENTS

One of the most touted benefits of smart wheelchair is providing peace of mind to the relatives of the patient, allowing the patient to move around, countering dangers such as fall from the wheelchair and seeking help in dangerous situation. Anyways now we are discussing the hardware components which are using for our project.

4.2.1Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3. x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

The Arduino Nano is very much similar to the Arduino UNO. They use the same Processor (Atmega328p) and hence they both can share the same program. One big difference between both is the size. UNO is twice as big as Nano and hence occupies more space on project. Also, Nano is breadboard friendly while Uno is not. To program an Uno Regular USB cable is needed; whereas for Nano a mini USB cable is needed.

The Arduino Nano is the preferred board for many projects requiring a small and easy to use microcontroller board. This will allow to make larger programs than with the Arduino Uno (it has 50% more program memory), and with a lot more variables (the RAM is 200% bigger).

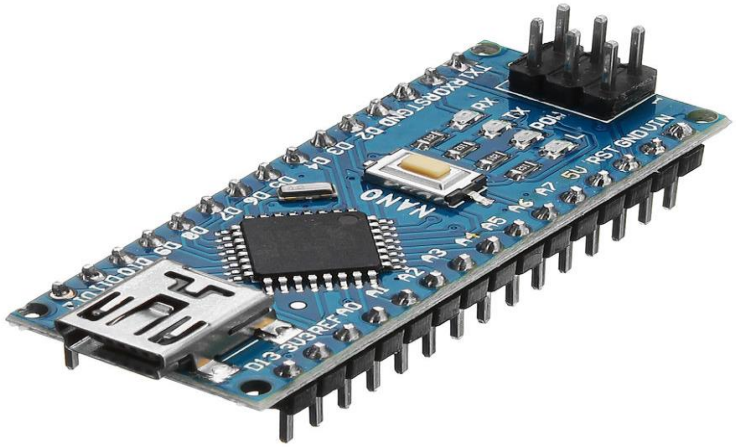

Figure 4.1 Arduino Nano

The Arduino board is designed in such a way that it is very easy for beginners to get started with microcontrollers. This board especially is breadboard friendly, and that's why it is very easy to handle the connections. The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on board).

The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows to upload new code to it without the use of an external hardware programmer. As It is based on the ATmega328 8-bit microcontroller by Atmel (Microchip Technology). The Atmega328 comes with a built-in bootloader, which makes it convenient to flash the Nano board with a program.[13]

Arduino Nano has a total of 36 pins. There are total 14 digital Pins and 8 Analog pins on Nano board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like pinMode() and digitalWrite() can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function like analogRead().Nano has a 16 MHz SMD crystal resonator, a mini USB-B port, an ICSP header, 3 RESET pins and, a RESET button.

4.2.2 MPU 6050

The MPU6050 can be a Small scale Electro-mechanical framework (MEMS), including a three-axis accelerometer and a three-axis gyroscope. It helps us to measure velocity, orientation, acceleration, displacement and other movement like features. It includes 16-bit analog to digital conversion hardware. Because of this integration, it captures three-dimensional movements simultaneously. It works on the power supply of 3V-5V. This

module uses the I2C module to connect to Arduino. The MPU6050 is less expensive, its main advantage is that it can successfully integrate with the accelerometer and gyro.[14]

There are eight PINOUT of MPU6050,which are described here:

Pin#	Pin Name	Description
01	Vcc	This pin used for Supply Voltage. Its input voltage is +3 to +5V.
02	GND	This pin use for ground
03	SCL	This pin is used for clock pulse for I2C compunction
04	SDA	This pin is used for transferring of data through I2C communication.
05	Auxiliary Serial Data (XDA)	It can be used for other interfaced other I2C module with MPU6050.
06	Auxiliary Serial Clock (XCL)	It can also be used for other interfaced other I2C module with MPU6050.
07	AD0	If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address.
08	interrupt (int)	This pin is used to indicate that data is available for MCU to read.

Table 4.1 : MPU6050 Pinout

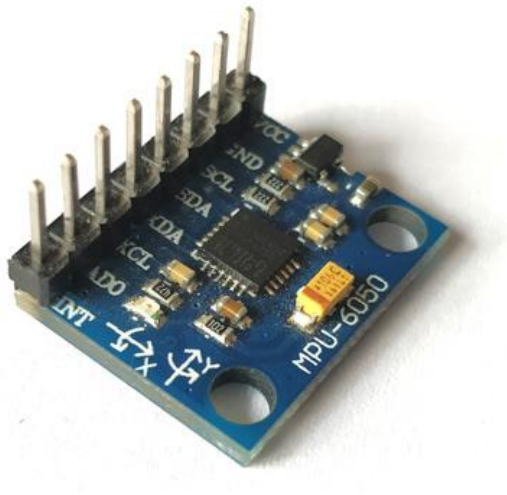

Figure 4.2 : MPU6050

4.2.3 nRF24L01

nRF24L01 is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst™ protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a SPI interface. Current consumption is very low, only 9.0mA at an output power of -6dBm and 12.3mA in RX mode. Built-in Power Down and Standby modes makes power saving easily realizable.[15]

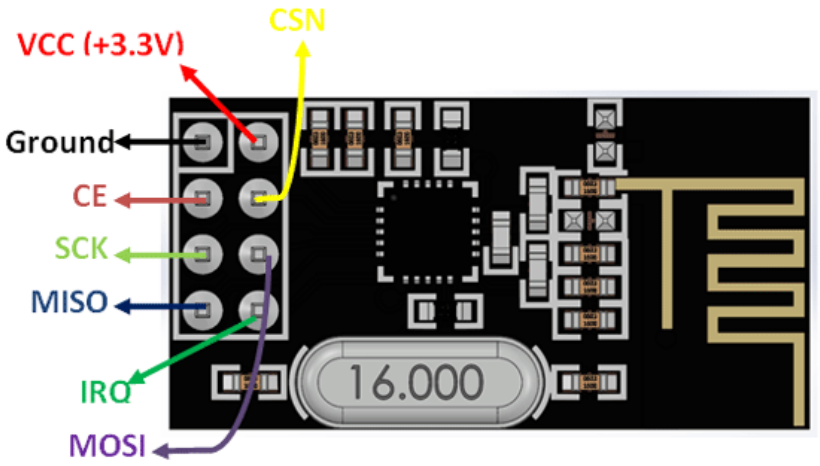

Figure 4.3: nRF24L01

Pin Number	Pin Name	Abbreviation	Function
1	Ground	Ground	Connected to the Ground of the system
2	Vcc	Power	Powers the module using 3.3V
3	CE	Chip Enable	Used to enable SPI communication
4	CSN	Ship Select Not	This pin has to be kept high always, else it will disable the SPI
5	SCK	Serial Clock	Provides the clock pulse using which the SPI communication works
6	MOSI	Master Out Slave In	Connected to MOSI pin of MCU, for the module to receive data from the MCU
7	MISO	Master In Slave Out	Connected to MISO pin of MCU, for the module to send data from the MCU
8	IRQ	Interrupt	It is an active low pin and is used only if interrupt is required

Table 4.2:nRF24L01 Pinout Configuration

4.2.4 L298N Motor Driver

L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors.78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.[16]

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

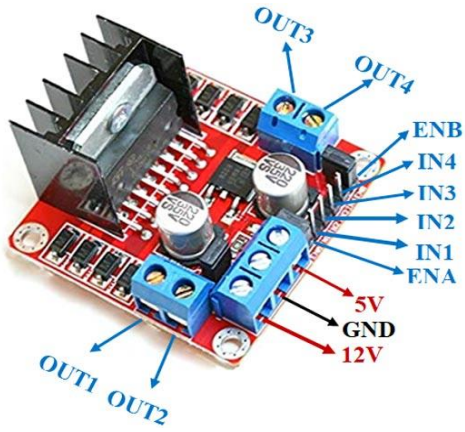

Figure4.4: L298N Motor driver

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B

OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

Table 4.3: L298N Module Pinout Configuration

4.2.5 Esp 8266

The ESP8266 is a very user friendly and low cost device to provide internet connectivity to projects. The module can work both as an Access point and as a station, hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible. It can also fetch data from internet using API's hence the project could access any information that is available in the internet making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. The ESP8266 module works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with circuits. However this version of the module has only 2 GPIO pins so we need to use it along with another microcontroller like Arduino.[17]

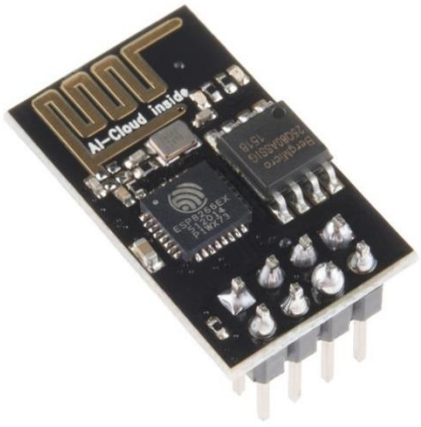

Figure 4.5 : ESP 8266

Pin Number	Pin Name	Normally used for	Alternate purpose
1	Ground	Connected to the ground of the circuit.	
2	TX	Connected to Rx pin of programmer to upload program.	Can act as a General purpose Input/output pin when not used as TX.
3	GPIO-2	General purpose Input/output pin.	
4	CH_EN	Chip Enable – Active high	
5	GPIO-0	General purpose Input/output pin.	Takes module into serial programming when held low during start up
6	Reset	Resets the module.	
7	RX	General purpose Input/output pin.	Can act as a General purpose Input/output pin when not used as RX [14]
8	VCC	Connect to +3.3V only	

Table 4.4: ESP 8266 Pinout

4.2.6 Motor

A motor is a machine that converts electrical energy into mechanical energy. In other words, electrical power is "battery" and mechanical power is "circular". To define the motor physically, the well-known "Fleming Left Hand Law" is a good way. Vehicle speed control is very important for robots and where the motor is used. In this project, we will control motor speeds with the Arduino Nano. In this way, you will learn to drive the motor and adjust the speed on the projects where you use it. To drive the motors we will use the L298N Motor Driver

Figure 4.6: Motor

4.2.7 Type B USB

This connector is designed to carry data and power in USB SuperSpeed applications. Cables with this connector are not backwards compatible with USB 2.0 or USB 1.1 devices; however USB 3.0 devices with this connection type can accept previous USB 2.0 and 1.1 cabling. Generally, the Type-B connector is the end of the standard USB cable that plugs into a peripheral device such as a printer, phone, or external hard drive, and is also known as the Type B-male. The port on the device itself is referred to as the Type B-female. USB cables that share the same type are universal. However, there are now different types of USB cables and ports that range in shape and size, and therefore, are not universal with other type USB cables without the use of an adapter.

Figure 4.7: Type B USB

4.2.8 LCD 16*2

I2C_LCD is an easy-to-use display module, it can make display easier. Using it can reduce the difficulty of make, so that makers can focus on the core of the work. Connect VO to the middle pin of the potentiometer – with it you can adjust the contrast of the screen display. Connect RS to D4 and R/W pin to GND, which means then you can write characters to the LCD1602. Connect E to pin6 and the characters displayed on the LCD1602 are controlled by D4-D7.

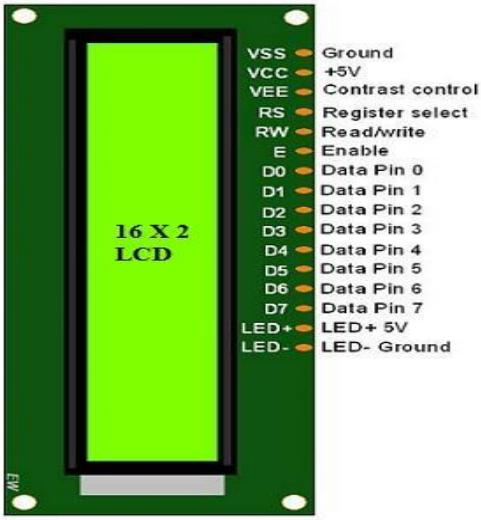

Figure 4.8 LCD 16*2

This is a 16x2 LCD display screen with I2C interface. It is able to display 16x2 characters on 2 lines, white characters on blue background. ... It means it only needs 4 pins for the LCD display: VCC, GND, SDA, and SCL. It will save at least 4 digital/analog pins on Arduino. Although it may be time taking because you need to understand and connect 16 pins of LCD to the microcontroller. So first let's understand the 16 pins of LCD module. We can divide it in five categories, Power Pins, contrast pin, Control Pins, Data pins and Backlight pins. At the heart of the adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required by the LCD display. The board also comes with a small trim pot to make fine adjustments to the contrast of the display.[19]

4.2.9 Voltage Regulator IC 7805

Voltage regulators are very common in electronic circuits. They provide a constant output voltage for a varied input voltage. In our case the 7805 IC is an iconic regulator IC that finds its application in most of the projects. The name 7805 signifies two meaning, “78” means that it is a positive voltage regulator and “05” means that it provides 5V as output. So our 7805 will provide a +5V output voltage.[18]

Figure 4.9:Voltage Regulator IC 7805

The output current of this IC can go up to 1.5A. But, the IC suffers from heavy heat loss hence a Heat sink is recommended for projects that consume more current.

4.2.10 ASM 1117

The AMS1117 is a type of linear voltage regulator that is used in the circuit for varying the voltage flow from the input to the output of the circuit.

We often use the power supplies like batteries or direct AC/DC supply which are normally in higher ranges like 12 V compared to the actual requirement for the circuit

Figure 4.10: ASM 1117

4.2.11 MAX 30100

The MAX30100 is a complete pulse oximeter and heartrate sensor system solution designed for the demanding requirements of wearable devices. The MAX30100 provides very small total solution size without sacrificing optical or electrical performance. Minimal external hardware components are needed for integration into a wearable device. The MAX30100 is fully configurable through software registers, and the digital output data is stored in a 16-deep FIFO within the device. The FIFO allows the MAX30100 to be connected to a microcontroller or microprocessor on a shared bus, where the data is not being read continuously from the device's registers.[20]

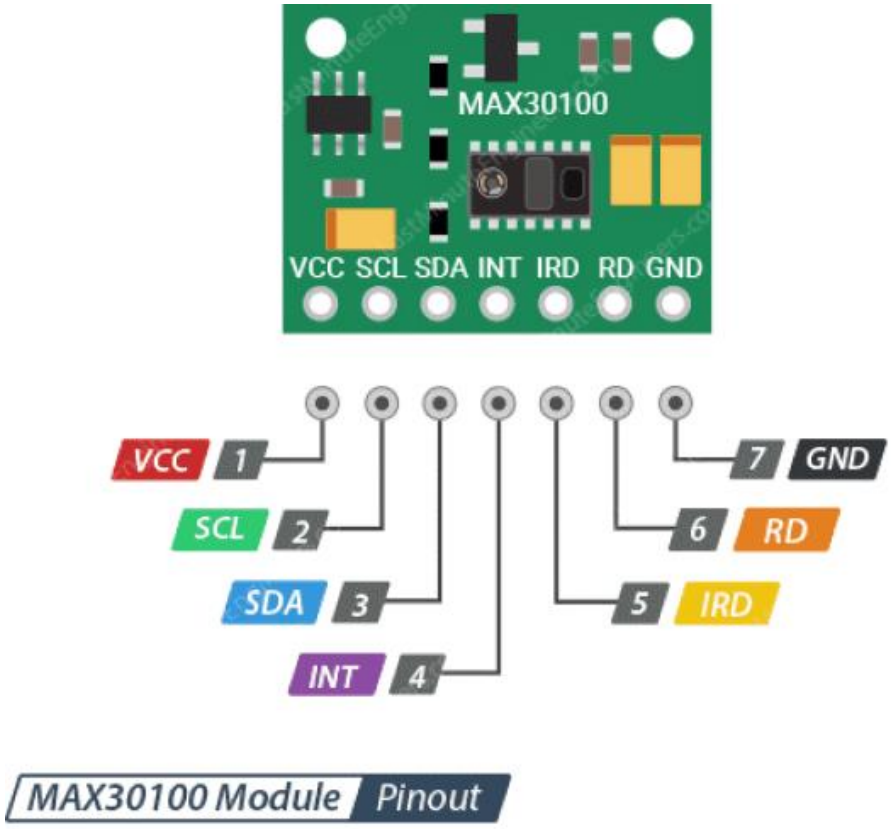

Figure 4.11: MAX30100

SN	PINS	DEFINITION OF PINS
1	VIN	Input voltage (1.8V to 5.5V)
2	SCL	IIC-SCL
3	SDA	IIC-SDA
4	INT	MAX30100INT
5	IRD	MAX30100 IR_DRV
6	RD	MAX30100 R_DRV
7	GND	Ground

4.2.12 Keypad

In embedded devices one of the essential part is Keypad is used to interact with embedded devices, Keypad is input devices which is used to give commend the devices, from calculator to computer input is given through keypad, in this tutorial you will know how to interface keypad with Arduino, Arduino is a mini computer which is user-friendly, we can hack anything using Arduino, let see how to interface 4x4 matrix keypad with Arduino.

Figure 4.12: Keypad

A keypad is a set of buttons arranged in a block or "pad" which bear digits, symbols or alphabetical letters. Pads mostly containing numbers are called a numeric keypad. Numeric keypads are found on alphanumeric keyboards and on other devices which require mainly numeric input such as calculators, push-button telephones, vending machines, ATMs, Point of Sale devices, combination locks, and digital door locks. Many devices follow the E.161 standard for their arrangement. Many laptop computers have special function keys which turn part of the alphabetical keyboard into a numerical keypad as there is insufficient space to allow a separate keypad to be built into the laptop's chassis. Separate external plug-in keypads can be purchased. Keypads for the entry of PINs and for product selection appear on many devices including ATMs, vending machines, Point of Sale payment devices, time clocks, combination locks and digital door locks.

4.2.13 Thermistor

A thermistor is a type of thermal resistor whose electrical resistance increase or decrease with change in temperature.Thus, the thermistors are thermally sensitive resistance whose resistance changes with change in the thermistor body temperature. The resistance of the thermistor changes with change in thermistor body temperature. The resistance of the thermistor does not vary linearly with change in temperature. The thermistor has non-linear resistance temperature curve. The resistance of the thermistor can be measured using resistance meter(Ohm-meter). By knowing the exact relationship between the change in the resistance with temperature,the temperature can be derived by measuring the resistance of

thermistor at particular temperature. The change in the thermistor resistance with temperature depends on the type of material used for thermistor construction.

Figure 4.13: Thermistor

4.2.14 GSM Module SIM800L

SIM800L GSM/GPRS module is a miniature GSM modem, which can be integrated into a great number of IOT projects. You can use this module to accomplish almost anything a normal cell phone can; SMS text messages, Make or receive phone calls, connecting to internet through GPRS, TCP/IP, and more! To top it off, the module supports quad-band GSM/GPRS network, meaning it works pretty much anywhere in the world. At the heart of the module is a SIM800L GSM cellular chip from SimCom. The operating voltage of the chip is from 3.4V to 4.4V, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.[21][22]

Figure 4.14 : GSM Module SIM800L

Pin Number	Pin Name	Description
1	NET	External antenna attachment pin
2	VCC	Power supply pin, 3.4V to 4.4V input
3	RST	Reset pin, pull low for 100ms to perform hard reset
4	RXD	Serial data input
5	TXD	Serial data output
6	GND	Module ground reference
7, 8	SPK	Speaker differential output
9, 10	MIC	Microphone differential input
11	DTR	Serial data terminal ready pin, pull high to enable sleep mode
12	RING	Interrupt output, active low

Table 4.5: SIM800L Pinout Configuration

4.3 CIRCUIT DIAGRAM

4.3.1 Health Monitoring Circuit Diagram

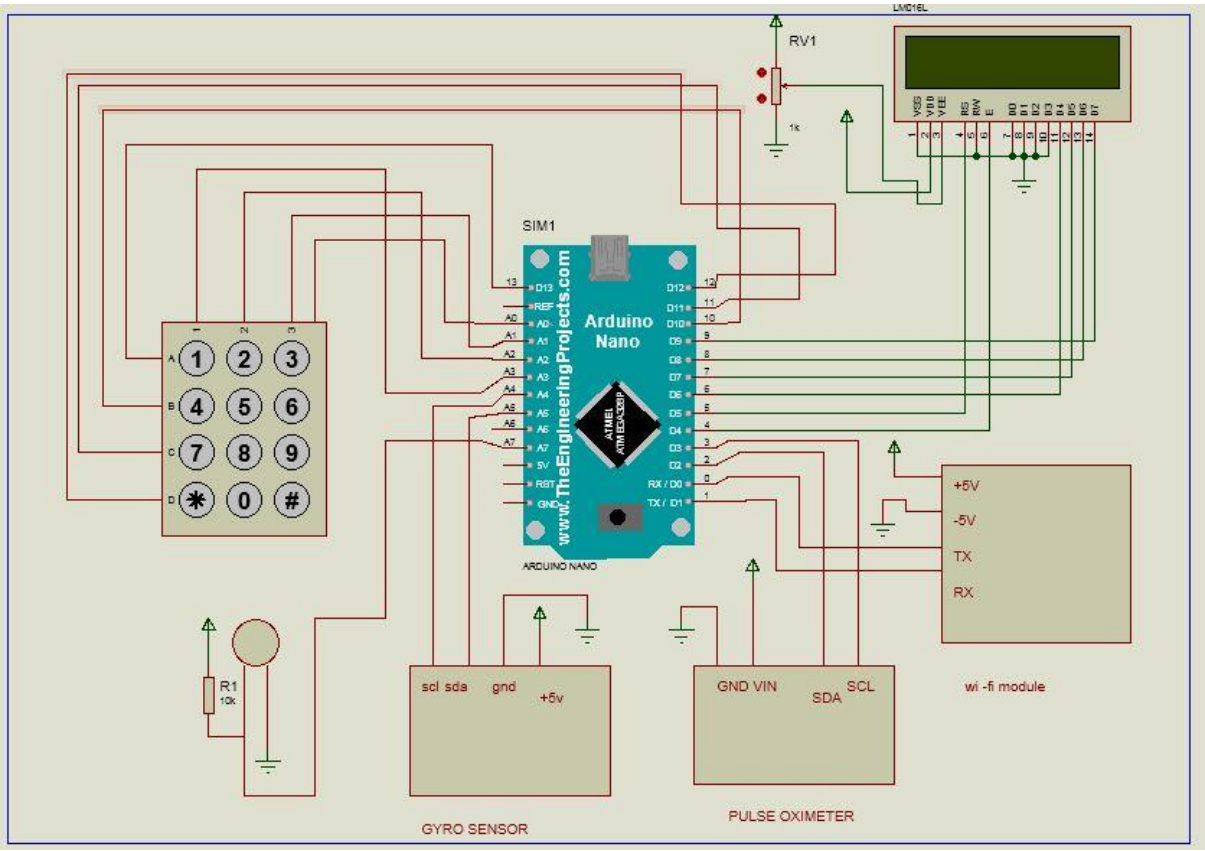

Figure 4.15: Health Monitoring Circuit Diagram

In the figure 4.15, all the connection of the modules those have been used in this system with the Arduino board is shown clearly in this diagram. All the VCC and GND connections of temperature, Gyro sensor, pulse sensor, Wifi module, Keypad, and LCD have been connected with the GND and VCC Connection of the Arduino. The Arduino-nano is started and simultaneously the Wi-Fi module is powered and the connection is established between the Wi-Fi module and access point through which can upload and access the sensor value over the internet. The control taken through the user is sent by Arduino nano and uploaded over the cloud server, then from the cloud server, the controller retrieves the data through a Wi-Fi module and performs the required action, depending on the control signal provided by the user.

4.3.2 Transmitter Circuit Diagram

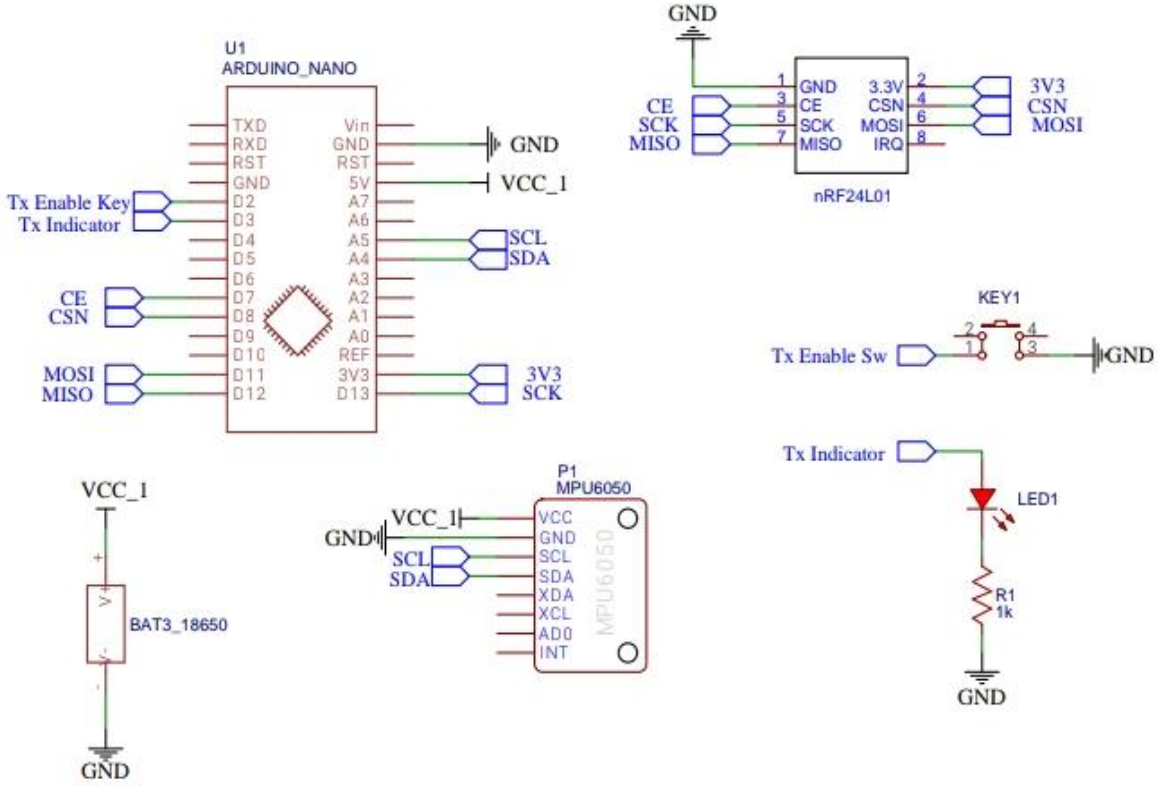

Figure 4.16: Transmitter Circuit Diagram

All the connection of this module used with Arduino nano are shown here. All the VCC and GND of gyro andnRF are connected with arduino's VCC and GND. Gyro gets 5v on the other hand nRF gets 3.3v and the pins are connected accordingly. After pressing the tx enable key the data works as per the instruction given by the user through the MPU 6050.

4.3.3 Receiver Circuit Diagram

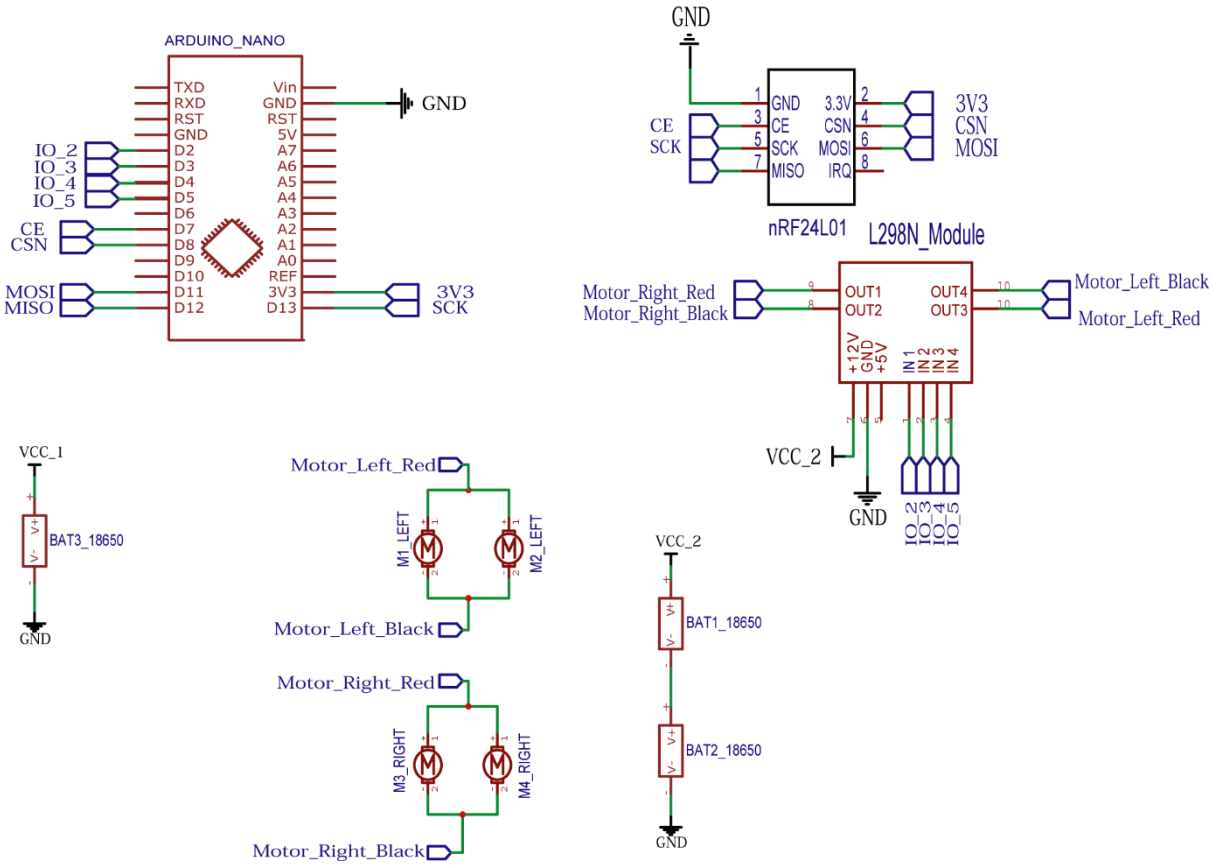

Figure 4.17: Receiver Circuit Diagram

All the connection of the modules those have been used in this system with the Arduino board is shown clearly in this diagram. The VCC and GND connection of nRF and L298N motor driver is connected with the VCC and GND connection of Arduino. The IO pins(IO 2 – IO 5) of the motor driver is connected with the digital pin of arduino (D2 – D5) . Four motors are connected in the output pins of the motor driver module. When the arduino is powered on the motor driver and nRF starts simultaneously; after receiving the command from the transmitter the motor driver moves according to the given command.

4.3.4 GSM Emergency Message Circuit Diagram

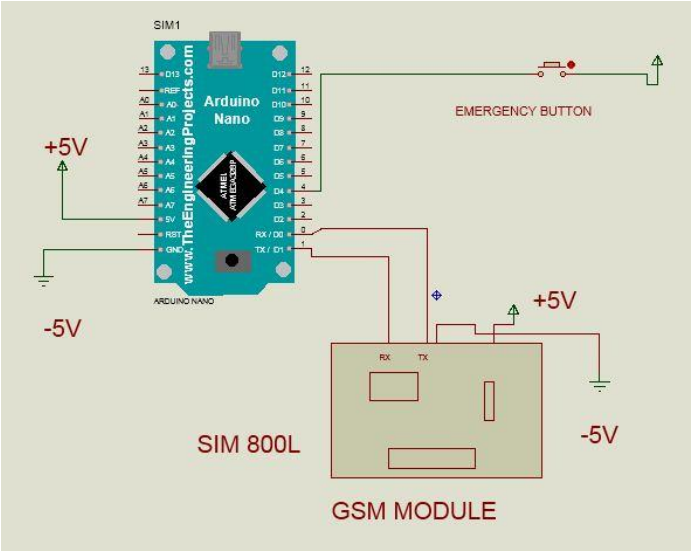

Figure 4.18: GSM Circuit Diagram

All the connection of the modules those have been used in this system with the Arduino board is shown clearly in this above diagram. The VCC and GND connection of gsm module is connected with the arduino's VCC and GND as well as the TX and RX. Button is connected with the D4 pin. When the arduino is on the gsm also starts and if the user presses the button an emergency message is sent to the given mobile phone number with gps location.

4.4 SUMMARY

In this chapter, we have discussed about all kind of hardware components that will be used in this project, circuit diagram.

CHAPTER 5

RESULT AND DISCUSSION

5.1 RESULT

After the implementation of the modules in our proposed system, it has to undergo the phase of experimentation and testing. The modules were tested for running the wheelchair forward, backward, left and right direction and the appropriate results were obtained. The following section includes the analysis of the obtained test results and its descriptions.

5.2 DISCUSSION

The developed device will provide movement assistance to the disabled individuals / aged people in their day-to-day operation thus increasing the level of comfort. This project describes the design and working of a system which is useful for disabled people to move around and provide their relatives with important information about their health and also they can get help with a single push of emergency button while facing danger. This device forms a bridge between the physically disabled people and the normal people.

Figure 5.1: Transmitter and Wheelchair

In the above figure 5.1, First we need to activate the system then the wheelchair is ready to move. Then we have to press the tx button so that the nRF in the wheelchair starts receiving the commands. It will move according to the command that is given by the hand gesture.

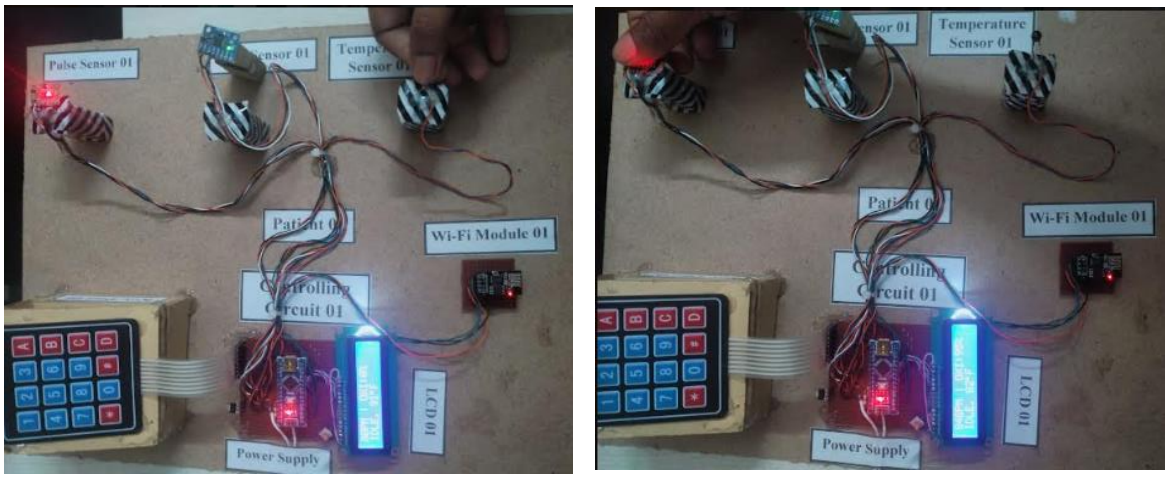

Figure 5.2: Health Monitoring System: Temperature Sensor & Pulse Sensor

In the above figure 5.2, First we need to activate all the systems shown here. If the Wifi connects with the WiFi Module of Health Monitoring System then it will send continuous data to the server.

When pulse sensor and temperature sensor is touched by the user; these sensors will collect their respective data and the display will show their data as we can see in the figure (5.2).

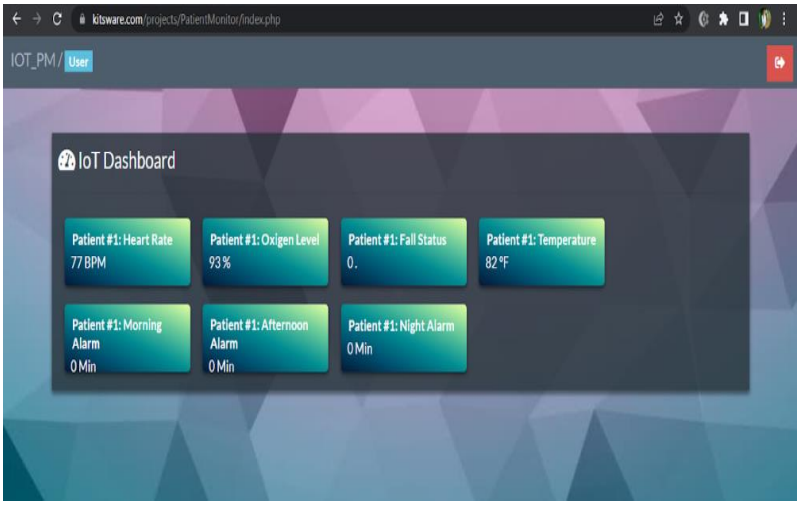

Figure 5.3: Values Displayed in Web.

In the above figure 5.3, also after activating the system, when we touch the temperature sensor and pulse sensor it displays the value received from these sensors in the Web. The data of these sensors are received by the server continuously so that it can provide accurate findings of the body.

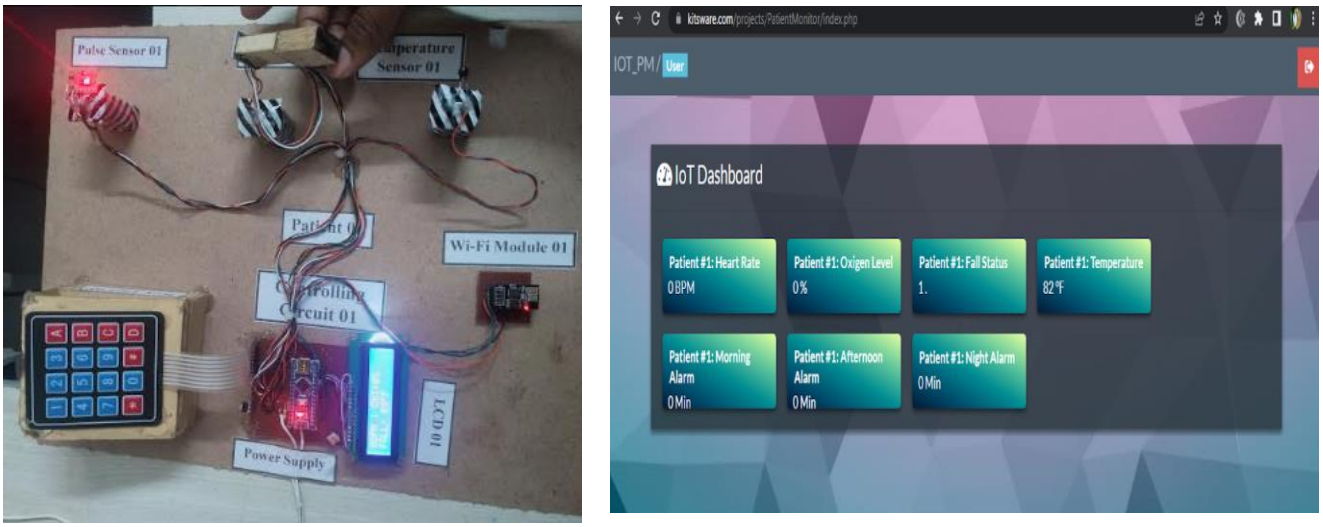

Figure 5.4: Fall Detection

In the above figure, when the person falls or trips over, the data is sent to the server and also it displays in the LCD Display.

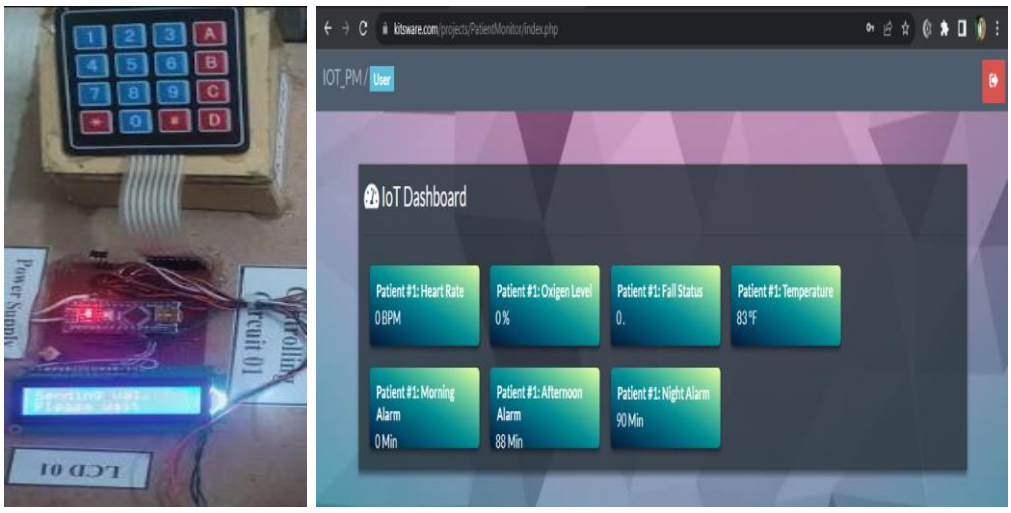

Figure 5.5: Medicine Reminder and data shown in web

After activating the system if we want to setup timing for medicine then we set the time in the keypad then these timing will go to the server and thus timing is shown in the webpage which we can see from the above figure.

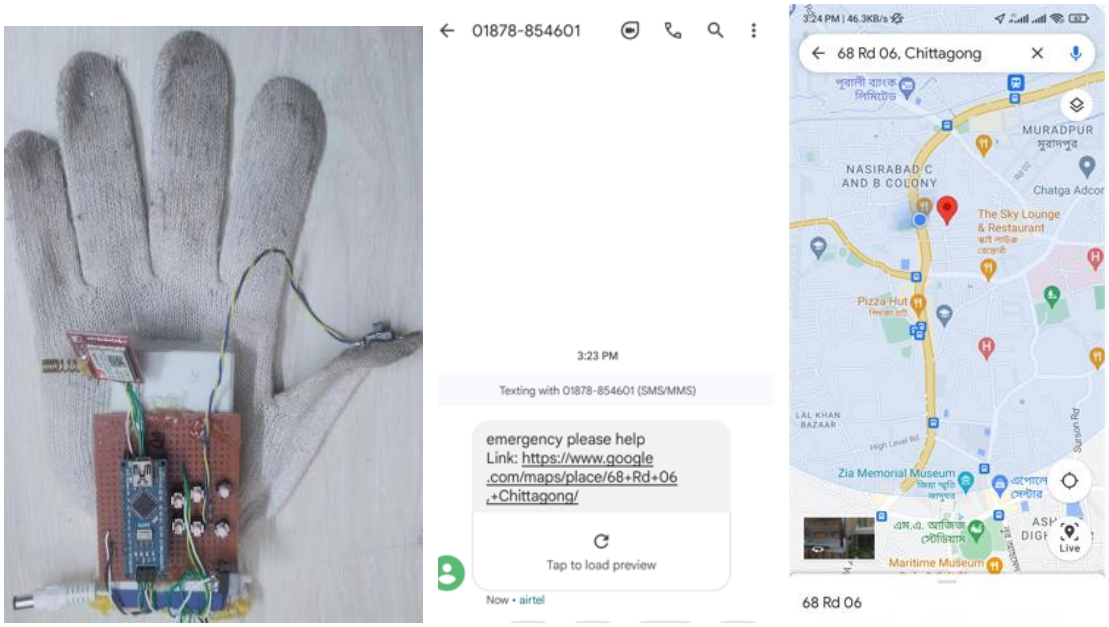

Figure 5.6: Emergency Message

In the above figure, when the patient is in need of some urgent help, he presses the button on his hand and through the GSM Module emergency message is sent to patient relatives with the goggle location.

5.3 SUMMARY

In this chapter, we have discussed all pages of the systems which show the expected result in brief.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

This scheme presents the smart wheelchair and health monitoring system for physically disabled people. This system will be efficient for those who are in need of physical assistance. By using the system, the user can move around and the guardian/responsible person can monitor his/her health through the system. In this study, the purpose and benefits of smart wheelchair and health monitoring system was demonstrated, with a particular emphasis on people with disabilities. The impact of disabilities was also researched and provided as evidence to support the beneficial effect of such system. The effectiveness of this system showed that disable people and their guardian/ responsible person can be rest assured.

6.2 LIMITATION

- We used two hands because LiPo batteries are expensive and also the weight will be increased if we would integrate the system in one hand.
- Charging could be an issue.

6.3 FUTURE SCOPES

1. In future the proposed framework will be customized into a SMD (surface mounted device) making it more convenient.
2. Run of communication will moreover be expanded to 100m with the capable remote gadgets.

REFERENCES

1. W. Zgallai, John Teye Brown, Afnan Ibrahim, Fatma Mahmood, Khulood Muhammad, Maitha Khalfan and Maryam Saleem, "Deep Learning AI Application to an EEG driven BCI Smart Wheelchair," 2019 Advances in Science and Engineering Technology International Conferences (ASET), 2019, pp. 1-5, doi: 10.1109/ICASET.2019.8714373

2. S. Nasif and M. A. G. Khan, "Wireless head gesture controlled wheel chair for disable persons," in 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2017, pp. 156–161.

3. R. J. Leela, A. Joshi, B. Agasthiya, U. Aarthiee, E. Jameela, and S. Varshitha, "Android based automated wheelchair control," in 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM). IEEE, 2017, pp. 349–353.

4. J. Gray, P. Jia, H. H. Hu, T. Lu, and K. Yuan, "Head gesture recognition for hands-free control of an intelligent wheelchair," Industrial Robot: An International Journal, 2007.

5. P. R. Kumar, K. Sumathi, V. S. Prithi, and S. S. Suriya, "Smart Assistance Library System for the Disabled: An IOT based UserFriendly Wheelchair," In 5th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 753-757, 2019.

6. R. Alkhatib, A. Swaidan, J. Marzouk, M. Sabbah, S. Berjaoui and M. O. Diab, "Smart Autonomous Wheelchair," In 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART), Paris, France, pp. 1-5, 2019

7. I. G. Torres, G. Parmar, S. Aggarwal, N. Mansur, A. Guthrie, "Affordable Smart Wheelchair," In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, Paper No. SRC07, pp. 1-6, 2019

8. Z. Fan, L. Meng, T. Q. Chen, J. Li, and I. M. Mitchell, "Learning Motion Predictors for Smart Wheelchair Using Autoregressive Sparse Gaussian Process," In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 713-718, 2018.

9. U. Garg, K. K. Ghanshala, R. C. Joshi and R. Chauhan, "Design and Implementation of Smart Wheelchair for Quadriplegia patients using IoT," In 2018 First International Conference on Secure Cyber Computing and Communication, pp. 106-110, India, 2018.

10. Rajesh Kannan Megalingam, VenkatRangan, Sujin Krishnan, AthulBalanEdicheryAlinkeezhil, "IR Sensor-Based Gesture Control Wheelchair for Stroke and SCI Patients", IEEE Sensors Journal Vol. 16, Issue: 17, Sept.1, 2016.

11. M.R.R.M. Akanda, M.M. Khandaker, T. Saha, J. Haque, A. Majumder, A. Rakshit, "Voice-Controlled Smart Assistant and RealTime Vehicle Detection for Blind People," In Advances in Electrical and Computer Technologies. Lecture Notes in Electrical Engineering, Vol. 672, Springer, Singapore.

12. Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, and S. Wang, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," In Journal of Computational Science, Vol. 25, pp. 50-57, 2018.

13. "Arduino nano - components101.com" https://components101.com/microcontrollers/arduino-nano (accessed june. 17, 2022).

14. " What is MPU6050? – Arduino Project Hub" https://create.arduino.cc/projecthub/CiferTech/what-is-mpu6050-b3b178 (accessed june. 17, 2022).

15. "nRF24L01 Wireless RF Module - Components101" https://components101.com/wireless/nrf24l01-pinout-features-datasheet (accessed june. 17, 2022).

16. "L298N Motor Driver Module Pinout - Components101" https://components101.com/modules/l293n-motor-driver-module (accessed june. 17, 2022).

17. "ESP8266 Wi-Fi MCU I Espressif Systems" https://www.espressif.com/en/products/modules/esp8266 (accessed june. 17, 2022).

18. "7805 Voltage Regulator IC - Components101" https://components101.com/ics/7805-voltage-regulator-ic-pinout-datasheet (accessed june. 17, 2022).

19. "Standard LCD 16x2 Display - Thingbits Electronics" https://www.thingbits.in/products/standard-lcd-16x2-display (accessed june. 17, 2022).

20. "Interfacing MAX30100 Pulse Oximeter and Heart Rate Sensor with Arduino - lastminuteengineers.com" https://lastminuteengineers.com/max30100-pulse-oximeter-heart-rate-sensor-arduino-tutorial (accessed june. 17, 2022).

21. "SIM800L GSM / GRPS module - Nettigo" https://nettigo.eu/products/sim800l-gsm-grps-module (accessed june. 17, 2022).

22. "Send Receive SMS & Call with SIM800L GSM Module & Arduino - lastminuteengineers.com" https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial (accessed june. 17, 2022).

APPENDIX

SOURCE CODE

Tx.ino

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Wire.h>
#include "MPU6050.h"
#define CSN_GPIO    8
#define CE_GPIO     7
// Hardware configuration
RF24 radio(CE_GPIO, CSN_GPIO);                    // Set up nRF24L01 radio on SPI bus plus pins 7 & 8
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_tgx, gy, gz;
#define TX_ENABLE_KEY     2
#define TX_LED            3
const byte Address[6] = "00001";
intPot_Val_Y = 0,Pot_Val_X = 0, Up_key = 0, Dn_key = 0, Left_key = 0, Right_key = 0;
unsigned char Tx_command = 0,Speed_index = 0,Tx_Enable_Flag = 0,TX_Key_Pressed = 0;
unsigned char Tx_Array[2];
void setup() {
  Serial.begin(115200);
  pinMode(TX_ENABLE_KEY, INPUT_PULLUP);
  pinMode(TX_LED, OUTPUT);

  radio.begin();
  radio.openWritingPipe(Address);
  radio.setPALevel(RF24_PA_MAX);
  radio.stopListening();
```

```
  radio.write(&Tx_command, sizeof(Tx_command));


  Wire.begin();
  Serial.println("Initializing I2C devices...");
  accelgyro.initialize();
  // verify connection
  Serial.println("Testing device connections...");
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050
connection failed");
  Tx_Array[0] = 0;
  Tx_Array[1] = 0;
}
void loop()
{
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  if(!(PIND & 0x04)&&(TX_Key_Pressed==0))
  {
     TX_Key_Pressed = 1;
     if(Tx_Enable_Flag==0)
     {
       Tx_Enable_Flag = 1;
       PORTD |= 0x08;
       //Serial.println("A");
     }
     else
     {
       Tx_Enable_Flag = 0;
       PORTD &= 0xF7;
       //Serial.println("B");
     }
     //PORTD |= 0x08;
     Serial.println("Pressed");
```

```
}
else if((PIND & 0x04)&&(TX_Key_Pressed==1))
{
  TX_Key_Pressed = 0;
  Serial.println("Released");
}

if((ay<=-4000)||((ay>=4000)))
{
  if((ax>=-4000)||((ax<=4000)))
  {
    Serial.print("B , ");
    if((ay<=-4000))
    {
       Tx_command = 1;                    // forward
       Speed_index = (ay + 4000)/-2000 + 1;
       if(Speed_index>5)
       {
         Speed_index = 5;
       }
    }

    if((ay>=4000))
    {
       Tx_command = 2;                    // Backward
       Speed_index = (ay - 4000)/2000 + 1;
       if(Speed_index>5)
       {
         Speed_index = 5;
       }
    }
  }
```

```
  else
  {
     Tx_command = 0;
     Speed_index = 0;
  }
}
else if((ax<=-4000)||((ax>=4000)))
{
  if((ay>=-4000)||((ay<=4000)))
  {
    if((ax<=-4000))
    {
       //Serial.print("H , ");
       Tx_command = 4;                    // Right
       Speed_index = (ax + 4000)/-2000 + 1;
       if(Speed_index>5)
       {
         Speed_index = 5;
       }
    }

    if((ax>=4000))
    {
       Tx_command = 3;                    // Left
       Speed_index = (ax - 4000)/2000 + 1;
       if(Speed_index>5)
       {
         Speed_index = 5;
       }
    }
  }
```

```
  else
  {
     Tx_command = 0;
     Speed_index = 0;
  }
 }
 else
 {
    Tx_command = 0;
    Speed_index = 0;
 }
 Serial.print(Tx_command);
 Serial.print(" , ");
 Serial.println(Speed_index);
 if(Tx_Enable_Flag)
 {
  Tx_Array[0] = Tx_command;
  Tx_Array[1] = Speed_index;
  radio.write(&Tx_Array, 2);    // 1st byte = Direction , 2nd Byte = Speed
 }
 delay(100);
}
```

Rx.ino

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#define CSN_GPIO    8
#define CE_GPIO     7
#define RIGHT_FORWARD     5  // Left side motor forward
#define RIGHT_BACKWARD    4  // Left side motor backward
#define LEFT_FORWARD      3  // Right side motor forward
#define LEFT_BACKWARD     2  // Right side motor backward
// Hardware configuration
RF24 radio(CE_GPIO, CSN_GPIO);                    // Set up nRF24L01 radio on SPI bus plus pins 7 & 8
const byte Address[6] = "00001";
unsigned char Received_Command = 0,Speed_index = 0,Run_Stop_Mode = 0;   //  Run_Stop_Mode -> 0 = Stop , 1 = Run;
unsigned char Rx_Array[2];
unsignedintRun_Stop_Counter = 0;
void setup() {
  Serial.begin(115200);
  pinMode(RIGHT_FORWARD,OUTPUT);   //left motors forward
  pinMode(RIGHT_BACKWARD,OUTPUT);   //left motors reverse
  pinMode(LEFT_FORWARD,OUTPUT);   //right motors forward
  pinMode(LEFT_BACKWARD,OUTPUT);   //right motors reverse
  radio.begin();
  radio.openReadingPipe(0, Address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
  Serial.println("START");
}

void loop()
```

```
{
  if (radio.available())    // If the NRF240L01 module received data
  {
    delay(1);
    //radio.read(&Received_Command, 1);
    radio.read(&Rx_Array, 2);
    Received_Command = Rx_Array[0];
    Speed_index = Rx_Array[1];
    Serial.print(Received_Command);
    Serial.print(" , ");
    Serial.println(Speed_index);
  }
  if(Run_Stop_Mode==0)    // Stop
  {
    digitalWrite(RIGHT_FORWARD,LOW);
    digitalWrite(RIGHT_BACKWARD,LOW);
    digitalWrite(LEFT_FORWARD,LOW);
    digitalWrite(LEFT_BACKWARD,LOW);
    Run_Stop_Counter++;
    if(Run_Stop_Counter>=((5-Speed_index)*100))   // Speed_index = 1 -> Min speed,   5 -> Max
Speed.
    {
      Run_Stop_Counter = 0;
      Run_Stop_Mode = 1;
    }
  }
  else if(Run_Stop_Mode==1)    // Run
  {
    Run_Stop_Counter++;
    if(Run_Stop_Counter>=((Speed_index)*100))   // Speed_index = 1 -> Min speed,   5 -> Max
Speed.
    {
```

```
  Run_Stop_Counter = 0;
  Run_Stop_Mode = 0;
 }
 if(Received_Command == 1)          //move forward(all motors rotate in forward direction)
 {
  digitalWrite(LEFT_FORWARD,HIGH);
  digitalWrite(RIGHT_FORWARD,HIGH);
 }
 else if(Received_Command == 2)     //move reverse (all motors rotate in reverse direction)
 {
  digitalWrite(LEFT_BACKWARD,HIGH);
  digitalWrite(RIGHT_BACKWARD,HIGH);
 }
 else if(Received_Command == 3)     //turn right (left side motors rotate in forward direction, right side motors rotates in backward direction)
 {
  digitalWrite(LEFT_FORWARD,HIGH);
  digitalWrite(RIGHT_BACKWARD,HIGH);
 }
 else if(Received_Command == 4)     //turn left (right side motors rotate in forward direction, left side motors rotates in backward direction)
 {
  digitalWrite(RIGHT_FORWARD,HIGH);
  digitalWrite(LEFT_BACKWARD,HIGH);
 }
 else if(Received_Command == 0)     //STOP (all motors stop)
 {
  digitalWrite(RIGHT_FORWARD,LOW);
  digitalWrite(RIGHT_BACKWARD,LOW);
  digitalWrite(LEFT_FORWARD,LOW);
  digitalWrite(LEFT_BACKWARD,LOW);
 }
```

```
   else                          //STOP (all motors stop) , If any other command is received.
   {
     digitalWrite(RIGHT_FORWARD,LOW);
     digitalWrite(RIGHT_BACKWARD,LOW);
     digitalWrite(LEFT_FORWARD,LOW);
     digitalWrite(LEFT_BACKWARD,LOW);
   }
  }
}
```

Gsm.ino

```
#include <SoftwareSerial.h>
#include <Wire.h>
SoftwareSerialgsm(10, 11); //ARDU RX, TX -> BT TX, RX
#define MPU_addr 0x68
#define buzz  A1
#define relay A2
#define butt1 7
#define butt2 8
#define butt3 9
#define butt4 4
#define butt5 5
#define butt6 6
#define led   13
//#define LOCATION "\nLink: https://www.google.com/maps/place/68+Rd+06,+Chittagong/"
#define number1 "01878854601"
#define number2 "01636302244"
#define number3 "01875752347"

int count;
boolsmsFlag = 0;
```

```
void setup() {
  Serial.begin(9600);
  gsm.begin(9600);
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  pinMode(buzz, OUTPUT);
  pinMode(relay, OUTPUT);
  pinMode(butt1, INPUT_PULLUP);
  pinMode(butt2, INPUT_PULLUP);
  pinMode(butt3, INPUT_PULLUP);
  pinMode(butt4, INPUT_PULLUP);
  pinMode(butt5, INPUT_PULLUP);
  pinMode(butt6, INPUT_PULLUP);
  pinMode(led, OUTPUT);
  digitalWrite(led, 0);
  GSMinit();
  digitalWrite(led, 1);
}

void loop() {
  int gas = analogRead(A0);
  bool fall = readGyro();
  digitalWrite(relay, 1);
  if (fall) count++;
  else count = 0;
  if (count > 30 && count < 50) digitalWrite(buzz, 1); // adjust time
  else if (count > 50) {                          // adjust time
    digitalWrite(relay, 0);
```

```
  sendSMS3("Emergency Help");
  digitalWrite(buzz, 0);
  while(fall) fall = readGyro();
 }
 if (gas > 200) {
  digitalWrite(buzz, 1);
  digitalWrite(relay, 0);
  sendSMS3("Alcohol Detect");
  digitalWrite(buzz, 0);
  while (gas > 200) gas = analogRead(A0);
 }
 if (!digitalRead(butt4)) {
  digitalWrite(buzz, 1);
  digitalWrite(relay, 0);
  sendSMS3("Emergency Help");
  digitalWrite(buzz, 0);
  while (!digitalRead(butt4));
 }
 if (!digitalRead(butt5)) {
  digitalWrite(buzz, 1);
  digitalWrite(relay, 0);
  sendSMS3("Emergency Help");
  digitalWrite(buzz, 0);
  while (!digitalRead(butt5));
 }
 if (!digitalRead(butt6)) {
  digitalWrite(buzz, 1);
  digitalWrite(relay, 0);
  sendSMS3("Emergency Help");
  digitalWrite(buzz, 0);
  while (!digitalRead(butt6));
```

```
  }

  Serial.println((String)"B4:" + digitalRead(butt4) + "\tB5:" + digitalRead(butt5)
              + "\tB6:" + digitalRead(butt6) + "\tF:" + fall + "\tG:" + gas + "\tTime:" + count);
  delay(100);
}
boolreadGyro() {
  intAcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 14, true);
  AcX = Wire.read() << 8 | Wire.read();
  AcY = Wire.read() << 8 | Wire.read();
  AcZ = Wire.read() << 8 | Wire.read();
  Tmp = Wire.read() << 8 | Wire.read();
  GyX = Wire.read() << 8 | Wire.read();
  GyY = Wire.read() << 8 | Wire.read();
  GyZ = Wire.read() << 8 | Wire.read();
  AcX = map(AcX, -32768, 32767, -90, 90);
  if (AcX< -10 || AcX> 10) return true;
  else return false;
}

voidGSMinit() {
  delay(5000);
  gsm.println("AT");
  response();
  gsm.println("ATE1");
  response();
  gsm.println("AT+CMGF=1");
  response();
```

```
  gsm.println("AT+CNMI=1,2,0,0,0");
  response();
  }
void sendSMS3(String txt) {
  Serial.println("Sending SMS 1.");
  sendSMS(number1, (String)txt + LOCATION);
  Serial.println("Sending SMS 2.");
  sendSMS(number2, (String)txt + LOCATION);
  Serial.println("Sending SMS 3.");
  sendSMS(number3, (String)txt + LOCATION);
  smsFlag = 1;
}

voidsendSMS(String number, String txt) {
  gsm.print("AT+CMGF=1\r\n");
  response();
  gsm.print("AT+CMGS=\"");
  response();
  gsm.print(number);
  gsm.print("\"\r\n");
  response();
  gsm.print(txt);
  gsm.write(0x1A);
  gsm.print("\r\n");
  delay(3000);
  response();
}
void response() {
  while (!gsm.available());
  if (gsm.available()) {
    gsm.readString();
```

```
  }
}
```

Health.ino

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
#include "MAX30100_PulseOximeter.h"

#define DEBUG 0 // 1 for serial monitor, 0 for wifi

#define REPORTING_MS      1000
#define SERVER_UPDATE_MS 15000

#define link   "WEB=http://kitsware.com/projects/PatientMonitor/update_machine.php?"

#define buzz 2

LiquidCrystallcd(9, 8, 7, 6, 5, 4);
PulseOximeter pox;
SoftwareSerialgsm(A6, 3);

charhexaKeys[4][4] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byterowPins[4] = {A3, A2, A1, A0};
bytecolPins[4] = {13, 10, 11, 12};
Keypad key = Keypad(makeKeymap(hexaKeys), rowPins, colPins, 4, 4);

constintMPU_addr = 0x68;
```

```
int16_tAcX, AcY, AcZ;
int16_tGyX, GyY, GyZ, Tmp;
bool fall;
uint32_ttsLastReport = 0;
int bpm, spo2, temp, level;
longct;
String sLevel = "LOW";

int alarm1, alarm2, alarm3, timer, ss, count, mm;
boolisAlarm, smsFlag;

void (*RESET) (void) = 0x00;

void setup() {
  delay(1000);
  Serial.begin(9600);
  lcd.begin(16, 2);

  gsm.begin(9600);
  GSMinit();

  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  pox.begin();

  pinMode(buzz, OUTPUT);

  Serial.println("CON=iot,12345678,");
  ct = millis();
}
```

```
void loop() {
   fallDetector();
  pox.update();

  char k = key.getKey();
  if (k == 'A') setAlarm();

  if (millis() - tsLastReport> REPORTING_MS) {
    bpm = pox.getHeartRate();
    spo2 = pox.getSpO2();
    readTemp();

    lcd.setCursor(0, 0);
    lcd.print((String)bpm + "BPM | OXI:" + spo2 + "%    ");
    lcd.setCursor(0, 1);
    fall ?lcd.print("FALL!") : lcd.print("IDLE.");
    lcd.setCursor(6, 1);
    lcd.print((String)temp + (char)223 + "F ");
    if (smsFlag == 0) {
      if (fall == 1) sendSMS("Fall Detect");
      else if (temp >= 104)  sendSMS("High fever");
      else if (bpm >= 150) {
        sendSMS("High BPM");
        RESET();
      }
      else if (spo2 >= 10 && spo2 <= 80) {
        sendSMS("Low SPO2");
        RESET();
      }
    }

#if DEBUG == 1
    Serial.println((String)bpm + " BPM | OXI_LEVEL: " + spo2);
```

```
    fall ?Serial.println("FALL DETECT!") : Serial.println("IDLE.");
    Serial.println((String)temp + "dF | Level: " + level + "\n");
#endif

    if (isAlarm == true) {
      count = 0;
      ss++;
      if (ss == 60) {
        ss = 0;
        mm++;
      }

      if (alarm1 != 0 && mm == alarm1) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(">> ALARM 01");
        lcd.setCursor(0, 1);
        lcd.print("MORNING");
              sendSMS("MORNING ALARM");
        alarm1 = 0;
      }
      if (alarm2 != 0 && mm == alarm2) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(">> ALARM 02");
        lcd.setCursor(0, 1);
        lcd.print("AFTERNOON");
        sendSMS("AFTERNOON ALARM");
        alarm2 = 0;
      }
      if (alarm3 != 0 && mm == alarm3) {
        lcd.clear();
```

```
        lcd.setCursor(0, 0);
        lcd.print(">> ALARM 03");
        lcd.setCursor(0, 1);
        lcd.print("NIGHT");
        sendSMS("NIGHT ALARM");
        alarm3 = 0;
      }
    }

    tsLastReport = millis();
  }

#if DEBUG == 0
  if (millis() - ct> SERVER_UPDATE_MS) {
    Serial.println((String)link
                   + "bpm1=" + bpm
                   + "&oxi1=" + spo2
                   + "&temp1=" + temp
                   + "&fall1=" + fall
                   + "&p1a1=" + alarm1
                   + "&p1a2=" + alarm2
                   + "&p1a3=" + alarm3
                   + "&slvl1=" + sLevel + "$");
    smsFlag = 0;
    ct = millis();
  }
#endif
}

voidGSMinit() {
  lcd.clear();
  lcd.print(F("GSM INIT..."));
```

```
  delay(5000);
  delay(5000);
  delay(5000);

  gsm.print("AT\r\n");
  delay(1000);
  gsm.print("ATE0\r\n");
  delay(1000);
  gsm.print("AT+CMGF=1\r\n");
  delay(1000);
  gsm.print("AT+CNMI=1,2,0,0,0\r\n");
  delay(1000);
  lcd.clear();
}

voidsendSMS(char *msg) {
  digitalWrite(buzz, 1);
  lcd.clear();
  lcd.print(F("Sending SMS"));
  lcd.setCursor(0, 1);
  lcd.print(F("Please wait"));

  gsm.print("AT+CMGF=1\r\n");
  delay(500);
  gsm.print((String)"AT+CMGS=\"" + number + "\"\r\n");
  delay(500);
  gsm.print(msg);
  gsm.write(0x1A);
  gsm.print("\r\n");
  delay(3000);

  digitalWrite(buzz, 0);
  lcd.clear();
```

```
  smsFlag = 1;
}

voidsetAlarm() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("A=Morning B=Noon"));
  lcd.setCursor(0, 1);
  lcd.print(F("C=Night D=Exit"));

  while (true) {
    char k = key.waitForKey();
    String value = "";

    if (k == 'D') break;
    if (k == 'A') {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(F("Enter [Min]:"));

      value = "";
      while (k != '#') {
        k = key.waitForKey();
        if (k >= '0' && k <= '9') {
          value += k;
          lcd.setCursor(0, 1);
          lcd.print(value);
        }
      }
      alarm1 = value.toInt();
      lcd.setCursor(0, 1);
      lcd.print(F("D. Exit"));
    }
```

```
else if (k == 'B') {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("Enter [Min]:"));

  value = "";
  while (k != '#') {
    k = key.waitForKey();
    if (k >= '0' && k <= '9') {
      value += k;
      lcd.setCursor(0, 1);
      lcd.print(value);
    }
  }
  alarm2 = value.toInt();
  lcd.setCursor(0, 30);
  lcd.print(F("D. Exit"));
}
else if (k == 'C') {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("Enter [Min]:"));

  value = "";
  while (k != '#') {
    k = key.waitForKey();
    if (k >= '0' && k <= '9') {
      value += k;
      lcd.setCursor(0, 1);
      lcd.print(value);
      lcd.display();
    }
```

```
      }
      alarm3 = value.toInt();
      lcd.setCursor(0, 1);
      lcd.print(F("D. Exit"));
    }
  }
  if (alarm1 == 0 && alarm2 == 0 && alarm3 == 0) isAlarm = false;
  elseisAlarm = true;
  lcd.clear();
}

voidreadTemp() {
  float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;
  int Vo = analogRead(A7);
  float R2 = 10000 * (1023.0 / (float)Vo - 1.0);
  float logR2 = log(R2);
  float T = (1.0 / (c1 + c2 * logR2 + c3 * logR2 * logR2 * logR2));
  T = T - 273.15;
  temp = ((T * 9.0) / 5.0 + 32.0) + 4; // change 0 if needed temp value manual
}

voidfallDetector() {
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 14, true);
  AcX = Wire.read() << 8 | Wire.read();
  AcY = Wire.read() << 8 | Wire.read();
  AcZ = Wire.read() << 8 | Wire.read();
  Tmp = Wire.read() << 8 | Wire.read();
  GyX = Wire.read() << 8 | Wire.read();
```

```
  GyY = Wire.read() << 8 | Wire.read();
  GyZ = Wire.read() << 8 | Wire.read();
  Wire.endTransmission(true);

  GyX = map(AcX, -32768, 32767, -90, 90);
  GyY = map(AcY, -32768, 32767, -90, 90);
  GyZ = map(AcZ, -32768, 32767, -90, 90);

  if (GyZ< -10) fall = 1;
  else fall = 0;
}

void beep(int d) {
  digitalWrite(buzz, 1);
  delay(d);
  digitalWrite(buzz, 0);
}
```

Auth.php

```
<?php
        session_start();
        include 'config/init.php';
        include 'lib/inputProcess.php';
        $db= new Database();

        $un= input_filter($_POST['username']);
        $pwd= input_filter($_POST['password']);
        $db->query("SELECT * FROM `user` WHERE `username`=? AND `password`= ?");
        $user=$db->fetchArray([$un,$pwd]);
        if(!empty($user)){
                $_SESSION['user']=$user;
                header('location: index.php');
        }else{
                $_SESSION['loginErr']=true;
```

```
                header('location: login.php');
        }
?>
```

Customize-admin.php

```
<?php
        session_start();
        include 'config/init.php';
        include 'lib/inputProcess.php';
         $db= new Database();
 ?>



<?php
        if(!isset($_SESSION['user'])){
                header('location: login.php');
        }else{
                extract($_SESSION['user']);
                if($usertype!='admin'){
                        header('location: unauthorized.php');
                }
        }
 ?>
```

Login.php

```
<?phpsession_start(); ?>
<?php include 'config/init.php'; ?>
<?php
        $header= new Templete('common/header');
        $footer= new Templete('common/footer');
 ?>
<?php echo $header ?>
        <div id="login-form" class="container p-3">
        <?php if (isset($_SESSION['loginErr'])): ?>
                <div class="alert alert-danger" role="alert">
```

```
                <i class="fas fa-exclamation-triangle"></i>The username or password is
wrong
                </div>
                <?php unset($_SESSION['loginErr']); ?>
        <?phpendif ?>
        <form action="auth.php" method='POST'>
        <div class="form-group">
        <label for="username">Username</label>
        <input name="username" type="text" class="form-control" id="username" aria-
describedby="username help" placeholder="Enter username" required="required">
        <small id="username help" class="form-text text-muted">Please input a valid user
name</small>
        </div>
        <div class="form-group">
        <label for="exampleInputPassword1">Password</label>
        <input name="password" type="password" class="form-control"
id="exampleInputPassword1" placeholder="Password" required="required">
        </div>
        <button type="submit" class="btnbtn-primary">Submit</button>
        </form>
</div>
<?php echo $footer ?>
```

Logout.php

```
<?php
        session_start();
        unset($_SESSION['user']);
        header('location: index.php');
?>
```

Stat-machine.php

```
<?php
        include 'config/init.php';
        $db = new Database();
```

```
	$db->query('SELECT * FROM `switchview`');
	$switchs=$db->fetchAll();
	foreach ($switchs as $key => $switch)
		echo $switch->code.'='.$switch->value.'<br>';
?>
```

Unauthorized.php

```
<?php
	if(!isset($_SESSION['user'])){
		header('location: login.php');
	}else{
		extract($_SESSION['user']);
	}
 ?>


<?php
	$header= new Templete('common/header');
	$footer= new Templete('common/footer');
 ?>
<?php echo $header ?>
<div class="jumbotron not-found">
<h1 class="display-4">Unauthorized Access</h1>
<p class="lead">You don't have appropriate permissions to access this page</p>
<hr class="my-4">
<p>Please contact the admin</p>
</div>
<?php echo $footer; ?>
```

Update-machine.php

```
<?php
	include 'config/init.php';
	include 'lib/inputProcess.php';
	$db= new Database;
	$machine_ip=$_SERVER['REMOTE_ADDR'];
	echo $machine_ip;
```

```
        $db->query("UPDATE `sensors` SET `value`= ? WHERE `name` = 'ip'");
        $db->execute([$machine_ip]);
        if(!empty($_REQUEST)){
                print_r($_REQUEST);
                foreach ($_REQUEST as $key => $value) {
                        $key=input_filter($key);
                        $db->query("SELECT * FROM sensors WHERE name = ?");
                        $tempsensor= $db->fetchArray([$key]);
                        $db->query("SELECT * FROM switchview WHERE code = ?");
                        $tempswitch= $db->fetchArray([$key]);
                        if(!empty($tempswitch)){
                                $db->query("UPDATE `switchview` SET `value`= ? WHERE
`code` = ?");
                                $db->execute([$value,$key]);
                        }else if(empty($tempsensor)){
                                $db->query("INSERT INTO `sensors` (`name`, `value`)
VALUES (?,?)");
                                $db->execute([$key,$value]);
                        }else{
                                $db->query("UPDATE `sensors` SET `value`= ? WHERE
`name` = ?");
                                $db->execute([$value,$key]);
                        }
                }
        }
 ?>
```

Index.php

```
<?php
        session_start();
        include 'config/init.php';
        include 'lib/inputProcess.php';
        $db = new Database();
?>
```

```
<?php
	if(!isset($_SESSION['user'])){
		header('location: login.php');
	}else{
		extract($_SESSION['user']);
	}
 ?>

<?php

 /*=====================================
 =	Switch Control		=
 =====================================*/

	if(isset($_GET['switchtoggle'])){
		$id= input_filter($_GET['switchtoggle']);
		$db->query('UPDATE `switchview` SET `value`= !value WHERE `serial`=
?');
		$db->execute([$id]);
	}

 /*=====  End of Switch Control  ======*/

  ?>


<?php
	$header= new Templete('common/header');
	$footer= new Templete('common/footer');
 ?>

<?php echo $header ?>
```

```
<div class="container">
<div class="view sensors-view">
        <span class="title"><i class="fas fa-tachometer-alt"></i>IoT
Dashboard</span>
        <?php if($usertype=='admin'):  ?>
            <a href="customize-admin.php#sensorview" class="btnbtn-success
float-right">
                <i class="fas fa-cog"></i>
            </a>
        <?phpendif; ?>
        <hr>
        <?php
            $db->query('SELECT * FROM `sensorview`');
            $sensors=$db->fetchAll();
            //print_r($sensors);
         ?>
        <div class="d-flex flex-wrap" >
            <?phpforeach ($sensors as $key => $sensor): ?>
                <div class="sensorbox">
                    <h3 class='title'><?php echo $sensor->title ?></h3>
                <?php
                    $db->query('SELECT * FROM `sensors` WHERE id=
?');
                    $sensordevice= $db->fetch([$sensor->sensorId])
                ?>
                <p class="value"><?php echo $sensordevice->value; ?><?php
echo $sensor->unit; ?></p>
                </div>
            <?phpendforeach ?>
        </div>
</div>
```

```
<div class="view switch-view" style="display:none;">
		<span class="title"><i class="fas fa-toggle-on"></i>IoT Control</span>
		<?php if($usertype=='admin'):  ?>
			<a href="customize-admin.php#switchview" class="btnbtn-success
float-right">
				<i class="fas fa-cog"></i>
			</a>
		<?phpendif; ?>
		<hr>
		<?php
			$db->query('SELECT * FROM `switchview`');
			$switchs=$db->fetchAll();
			//print_r($switchs);
		 ?>
		<div class="d-flex flex-wrap" >
			<?phpforeach ($switchs as $key => $switch): ?>
				<a class="switch" href="<?php echo
$_SERVER['PHP_SELF'].'?switchtoggle='.$switch->serial ?>">
					<div class="switchbox">
						<h3 class='title'><?php echo $switch->title
?></h3>
						<?php if ($switch->value==1): ?>
							<p class="text-center small"><i
class="far fa-circle text-success"></i> ON</span>
						<?php else: ?>
							<p class="text-center text-muted
small"><i class="far fa-dizzy"></i> OFF</span>
						<?phpendif ?>
					</div>
				</a>
			<?phpendforeach ?>
		</div>
</div>
```

```
<?php
	$db->query("SELECT * FROM `configuration` WHERE `item`='showMap'");
	$showMap=$db->fetch();
?>
<?php if($showMap->value=='1'):?>
	<div class="view Geoloaction-view">
		<span class="title"><i class="fas fa-map-marked-alt"></i>
Geolocation</span>
		<?php if($usertype=='admin'):  ?>
			<a href="customize-admin.php#mapView" class="btnbtn-
success float-right">
				<i class="fas fa-cog"></i>
			</a>
		<?phpendif; ?>
		<hr>
		<?php
			$db->query("SELECT `value` FROM `sensors` WHERE
`name`='lati'");
			$lati_row= $db->fetch();
			$lati=$lati_row->value;
			$db->query("SELECT `value` FROM `sensors` WHERE
`name`='longi'");
			$longi_row= $db->fetch();
			$longi=$longi_row->value;
		?>
		<iframe width="100%" height="300"
src="https://maps.google.com/maps?width=100%25&amp;height=600&amp;hl=en&amp;q=
<?php echo $lati;?>,<?php echo
$longi;?>+(TEST)&amp;t=&amp;z=15&amp;ie=UTF8&amp;iwloc=B&amp;output=embed"
></iframe>
	</div>
<?phpendif;?>
</div>
```

```
<?php if($usertype=='admin'):  ?>
        <a href="customize-admin.php#userview" class="add-user btnbtn-light">
                <i class="fas fa-user-plus"></i>
        </a>
<?phpendif; ?>


<?php echo $footer ?>
```