Major Project Report on

SMART ASSISTIVE SHOE FOR BLIND PEOPLE

Submitted in partial fulfilment for the
Degree of Bachelor of Technology in
Information Technology

Submitted by
SONALI JADHAV(1513015)
MITALI KAMBLI(1513017)
PRIYANKA MOGIDALA(1513026)

Under the guidance of
PROF. SUJATA KULLUR

USHA MITTAL INSTITUTE OF TECHNOLOGY
SNDT Women's University,
Santacruz(West),Mumbai-400049

2019-2020

Approval Sheet

This is to certify that **Sonali Jadhav, Mitali Kambli and Priyanka Mogidala** have completed the Mini Project report on the topic **"Smart Assistive Shoe For Blind People"** satisfactorily in partial fulfillment for the Bachelor's Degree in Information Technology under the guidance of Prof.Sujata Kullur during the year 2019-2020 as prescribed by S.N.D.T. Women's University.

Guide

Prof. Sujata Kullur

Head Of Department

Dr. Sanjay Shitole

Principal
Dr.Sanjay Pawar

Examiner 1

Examiner 2

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

SONALI JADHAV(1513015)
MITALI KAMBLI(1513017)
PRIYANKA MOGIDALA(1513026)

Date:

Contents

List of Figures

Chapter 1

Introduction

The main objective of our project is to provide an acoustic assistance to the blind people and also to deal with the problems faced by them to walk like normal human beings. Our project focuses to provide overall measures object detection and real-time assistance. The smart shoe will detect the nearby objects or obstacles and simultaneously send a message to the receiver in vibration form. So it helps the visually impaired person to acquire the extra knowledge about the obstacle around them without any help or any guidance. With the rapid advances of modern technology, both in hardware and software front have brought potential to provide intelligent navigation capabilities. We are presenting Smart Shoe project that enables the visually impaired users with mobility impairment to avoid obstacles. By leveraging existing robotics technologies, our system detects obstacles such as curbs, and staircases in the ground or even moving objects, and transmits obstacle information through haptic feedback (vibrations and beeps).

1.1 Objectives of the Study

The main objective is to design algorithms that are robust enough to reliably detect and warn for any obstacles that can appear in front of the user on the road area. Blindness, low vision, visual impairment and vision loss have dramatic impacts on individuals experiencing such disabilities. These carry with them physiological and psychological outcomes, hence impacting the quality of life and depriving such individuals from performing many of the Activities of Daily Living.

Chapter 2

Review of Literature

We are motivated with the thought of helping people who are facing difficulties because of their blindness to perform their daily activities. Hence we are developing this innovative solution called as Smart Shoe. Smart Shoe consists of sensors, microcontroller, actuators etc. to detect the obstacles coming their way. The following are the papers that we referred for our project :

Shubham Rastogi (2017), Smart Assistive Shoe, Department of Electronics and Communication, Moradabad : This paper focuses on designing a device for visual impaired (or blind) people that would help them to travel independently and also with more ease. It also offers a numerous applications in the medical field to provide a better responsive mate to the visually impaired.[1]

Hrushikesh Gosavi (2015), Obstacle Detection For Visually Impaired, International Research Journal : The paper is based upon the survey about the existing solutions meant for autonomous mobility for the visually impaired people. In this paper, they have proposed a novel design, Smart Shoes with sensors embedded in them to guide a visually impaired person fluidly and to alert him/her of the obstacles that lay ahead of him in his path.[2]

Anuradha Dubey (2014), Smart Device for Blind People, Dept. of Electronics and Communication, Gorakhpur : This paper presents an idea about dealing the problems faced by visually impaired individuals through assistive device in form of shoes. According to the WORLD HEALTH ORGANIZATION (WHO) survey in 2014 there is estimated 285 millions of people are visually impairment in which 37 millions of people are BLIND across the Globe over 15 millions of people are from INDIA. Where INDIA contributes to 21 percent of total blind population thats why most of the blind people dependent on other people for their activities. They have introduced a assistive shoe for blind people which will help them in their needed activities.[3]

Chapter 3

Problem statement

To design The "Integrated Smart Shoe" for visually impaired people which will detect and notify the obstacles in a way, using embedded systems technology. Artificial Vision is the most important part of human physiology as 83 percent of information human being gets from the environment is via sight. The statistics by the World Health Organization (WHO) in 2014 estimates that there are 285 billion people in world with visual impairment, 39 billion of people which are blind and 246 with low vision. The oldest and traditional mobility aids for persons with visual impairments are the walking cane (also called white cane or stick) and guide dogs. The drawbacks of these aids are range of motion and very little Information conveyed. With the rapid advances of modern technology, both in hardware and software front have brought potential to provide intelligent navigation capabilities. Recently there has been a lot of Electronic Travel Aids (ETA) designed and devised to help the blind people to navigate safely and independently. Also high-end technological solutions have been introduced recently to help blind persons navigate independently.

To design The " Integrated Smart Shoe " for visually impaired people which will detect and notify the obstacles in a way, using embedded systems technology. To provide intelligent navigation capabilities in the Integrated Smart Shoe with the rapidly advancing modern technology. To introduce high end technological solutions for helping the blind people navigate independently.

Chapter 4

Proposed Solution

This project presents a prototype model and a system concept to provide a smart shoe for blind people. This system is intended to provide overall measures object detection, and Real-time Assistance. This shoe contains Hardware components such as NodeMCU (Controller), Ultrasonic sensors, Power Supply (Transformer, Diode, capacitor, resistors, LED), Arduino Nano, PCB and Software components such as Kikad Software (PCB designing), Arduino IDE Software (Controller Programming including IOT programming), Blynk Software (IOT Mobile application). The shoe will be design to detect the obstacles around them and give blind person the tactile feedback so they can get rid of the obstacles and choose the right path to move on. The shoe will be use to guide the blind people, which is fitted with array of ultrasonic sensor around the sole.

Chapter 5

Components

5.1 Hardware Components

1. NodeMCU (Controller)
2. Ultrasonic sensors
3. Power Supply(Transformer,Diode,capacitor,resistors,LED)
4. PCB
5. Arduino Nano
6. Nylon cable ties to attach all the components on the shoe

5.2 Software Components

1. Kicad Software(PCB designing)
2. Arduino IDE Software (Controller Programming inculding IOT programming)
3. Blynk Software (IOT Mobile application)

5.3 Introduction to components

1. ESP-12E module

- NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.
- 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.
- Theres 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.
- For the sake of simplicity, we will make groups of pins with similar functionalities.

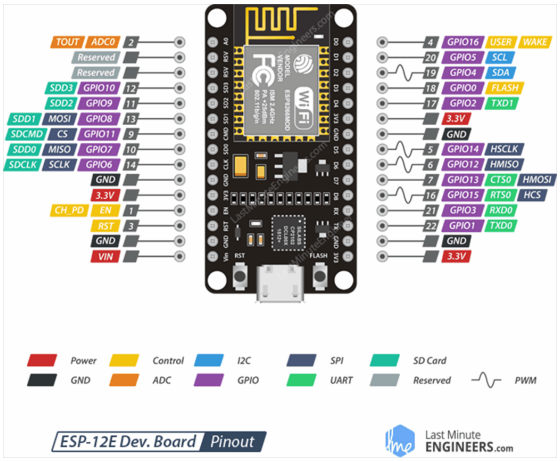

Figure 5.1: ESP8266

2. Ultrasonic Sensor

- The ultrasonic sensor belongs to a category of sensors that emits ultrasound i.e. sound of frequency more than 20 kHz.
- Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm.
- The basic principle of work:
    1. Intially, a trigger pulse is given as an input to the ultrasonic sensor.
    2. The ultrasonic sensor will emit a short ultrasonic burst signal.
    3. This burst signal will travel through the air and will hit an object and then it bounces back to the sensor resulting in an output pulse.
    4. Test distance = (high level time * velocity of sound (340M/S) / 2)
- The sensor consists of four pins:
  (1) VCC : It is used to provide 5V power to the sensor.
  (2) Trigger :- Takes in Input Pulse to trigger
  (3) Echo : It is used to receive the Output Pulse i.e. the echo from the object detected.
  (4) Ground :It connects sensor to the ground.

HC-SR04 Ultrasonic Sensor - Working

- The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below.
- As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like

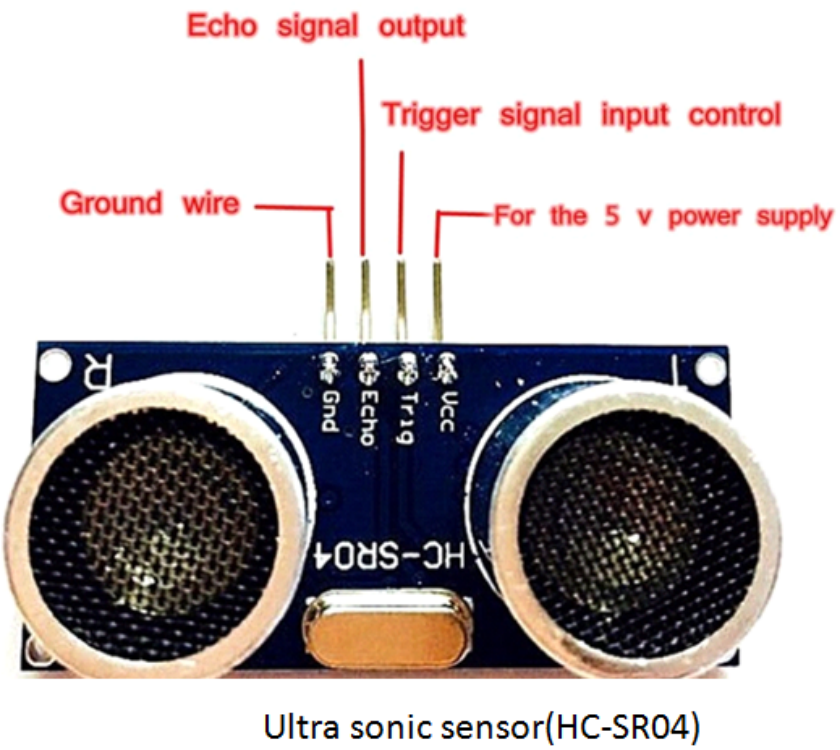

Figure 5.2: Ultrasonic Sensor

projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that
Distance = Speed x Time

- Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

- Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules

Figure 5.3: Ultrasonic Sensor Working

includes ultrasonic transmitters, receiver and control circuit.

- The basic principle of work:

  1. Using IO trigger for at least 10us high level signal,
  2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
  3. IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time * velocity of sound (340M/S) / 2.

- Wire connecting direct as following:

  1. 5V Supply
  2. Trigger Pulse Input
  3. Echo Pulse Output
  4. 0V Ground

- Electric Parameter:

    1. Working Voltage DC 5 V
    2. Working Current 15mA
    3. Working Frequency 40Hz
    4. Max Range 4m Min Range 2cm
    5. MeasuringAngle 15 degree
    6. Trigger Input Signal 10uS TTL pulse
    7. Echo Output Signal Input TTL lever signal and the range in proportion
    8. Dimension 45*20*15mm

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: ¡15
- Operating Current: ¡15mA
- Operating Frequency: 40Hz

How to use the HC-SR04 Ultrasonic Sensor

- HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.
- Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor.

- The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available).
- The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller.
- To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return.
- Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.
- The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

3. Lithium-ion Rechargeable Battery

- A lithium-ion battery or Li-ion battery is a type of rechargeable battery.
- Lithium-ion batteries are commonly used for portable electronics and electric vehicles and are growing in popularity for military and aerospace applications.
- Features:
    1. Power : 3.7V
    2. Current Rating: 2200mah
    3. Rechargeable
- Specifications :
    1. Size of Battery: 18*65mm
    2. Place of Original: Henan Province
    3. Material: lithium manganate/NCM
    4. Nominal Capacity: 1800mAh
    5. Nominal Voltage: 3.7V
    6. Charge Current: 0.2-0.5C@900mAh
    7. Discharge Current: 0.2-0.5C@900mAh
    8. Internal Resistance: 50m
    9. Cycle Life: above 250cycles
    10. Cell weight: Approx 40g/pc
    11. Operating Temperature: Charge:0C 45C; Discharging: -20C 60C
    12. Storage temperature: -10C 45C
- Applications :
    1. 3C (consumer) Digital Electronic Products: Power Bank; Bluetooth Speaker, Electronic Toys

2. Electric Products in Daily life : LED Bulbs , Mini Fan, LED light, Emergency Lamp, Outdoor LED light, Pos machine, Intelligent Lock
3. Salor Series Products: Solar Street Light, Solar Garden Light,Solar Bags
4. Electric Vehicles: E-bilke, Electric Scooter

Figure 5.4: Lithium-ion Rechargeable Battery

4. Voltage Regulator IC

- Voltage regulator is any electrical or electronic device that maintains the voltage of a power source within acceptable limits.
- The voltage regulator is needed to keep voltages within the prescribed range that can be tolerated by the electrical equipment using that voltage.
- Here in our circuit we are converting/regulating the voltage form 9 Volt to 5 Volts since the controller works on 5volts.
- Features:

1. Input voltage range 7V- 35V
2. Current rating Ic = 1A
3. Output voltage range Vmax=5.2V Vmin=4.8V

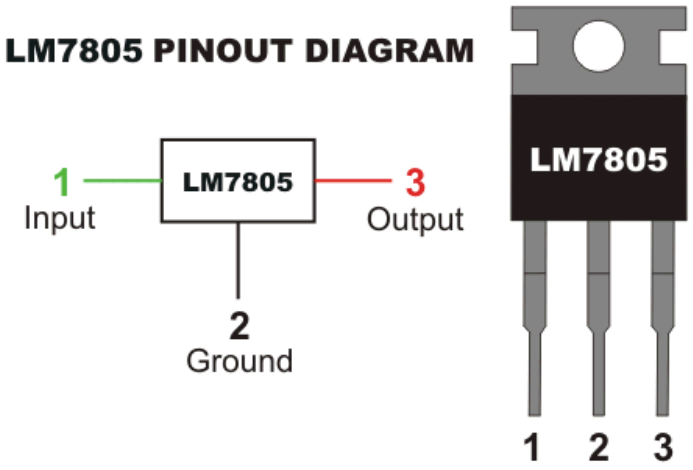

Figure 5.5: Voltage Regulator IC

5. Arduino Nano

- The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P; 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.

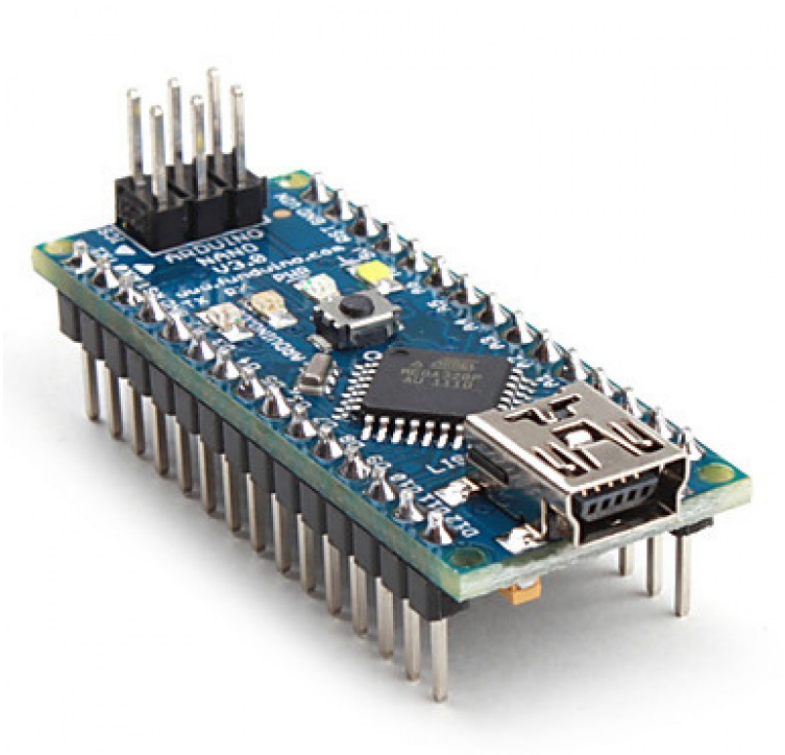

Figure 5.6: Arduino Nano

6. ARDUINO 1.8.10

- The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board.
- It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
- Language : C

Figure 5.7: ARDUINO 1.8.10

7. PCB Designing Software : KiCad

- KiCad is a free software suite for electronic design automation.
- It facilitates the design of schematics for electronic circuits and their conversion to PCB designs.
- KiCad was originally developed by Jean-Pierre Charras.
- It features an integrated environment for schematic capture and PCB layout design.

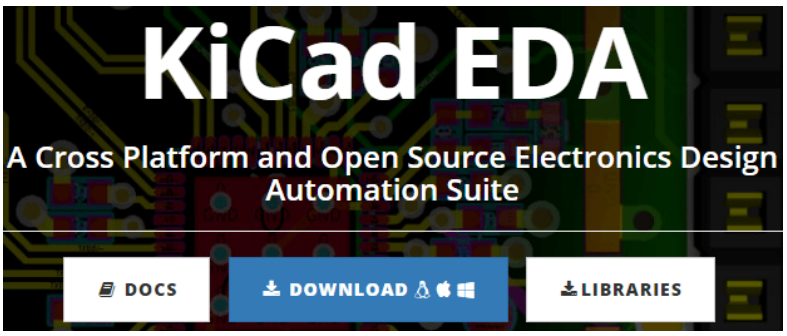

Figure 5.8: PCB Designing Software : KiCad

8. IOT Software : Blynk

- Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.
- It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

Figure 5.9: IOT Software : Blynk

COST OF THE COMPONENTS :

- Shoe : 500/-
- Arduino Nano : 399/-
- ESP-8266 Wi-Fi module : 360/-
- Ultrasonic sensors : 325 * 3 = 975/-
- Speaker : 300/-
- Jumper wires : 200/-
- Lithium-ion batteries : 100 * 2 = 200/-
- Nut/Bolts : 60/-
- Box : 100 * 2 = 200/-
- Brackets : 70/-
- Battery holders : 120 * 2 = 240/-
- Switch : 114/-
- Buttons : 95 * 2 = 190/-
- **Total cost = 3808/-**

Chapter 6

Implementation

6.1 Shoe : Tx 1 (Transmitter Section 1) :

- This section consists of 3 ultrasonic sensors,Node MCU and Battery.
- One sensor is used to detect the obstacles on the left side and the other sensor to detect the obstacle on the front side and the third sensor is used for pothole direction and it will also alert the blind person about the same.
- In this circuit, we have used a Micro-Controller named as Node MCU ESP8266 which will calculate the distance between the obstacle and the shoe detected by the ultra sonic sensor and it works as a transmitter to send these data to the Remote which acts as a receiver in our system.
- In this circuit, the Components used are as follows:
    1. Node MCU
    2. Capacitor
    3. Regulator IC
    4. Register
    5. Power LED
- We are using lithium batteries which are rechargeable and it posseses of 8 volts.

- The controller requires 5 Volts to perform its operation.
- since the controller runs on 5 volt , the above components are used for the regulation of the voltage i.e it converts the voltage from 8 to 5 volts.
- The circuit also consists of power switch which is used to turn on or turn off the circuit.
- THE ABOVE MENTIONED ENTIRE CIRCUIT WILL BE MOUNTED ON THE SHOE.
- The connection part of the kit is done according to the pcb design which is designed in the Kicad software.

Figure 6.1: Shoe

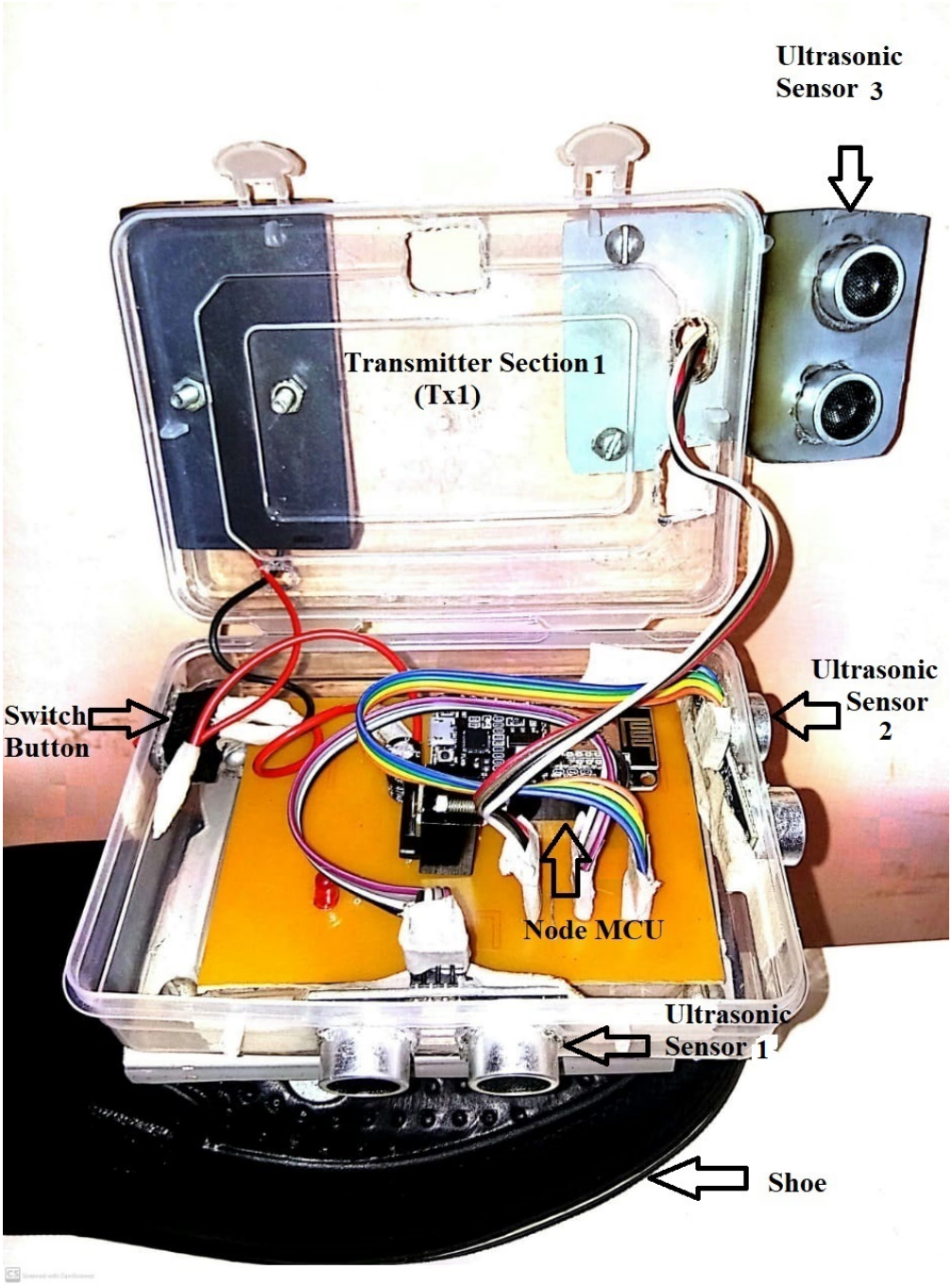

Figure 6.2: Shoe

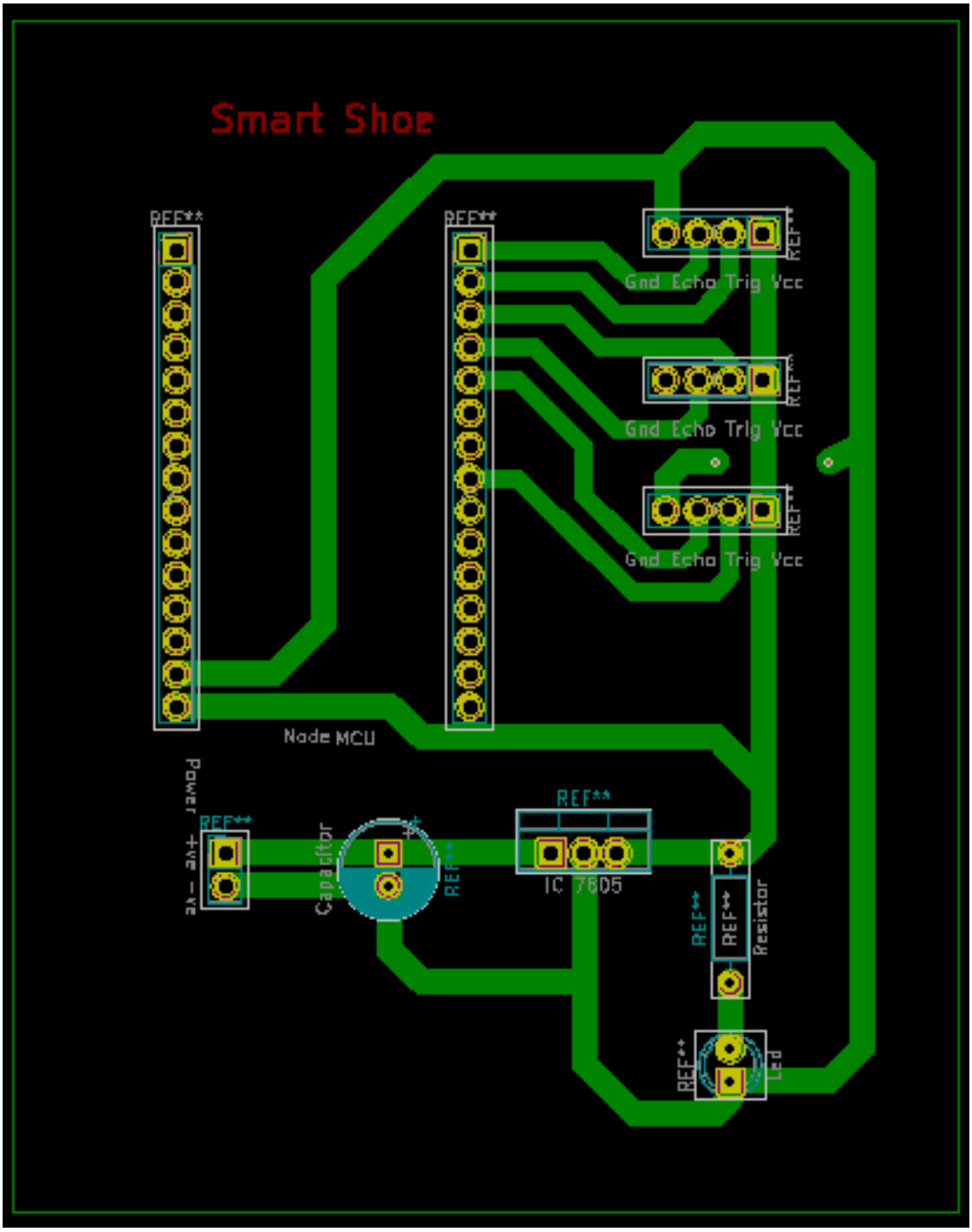

Figure 6.3: pcb design

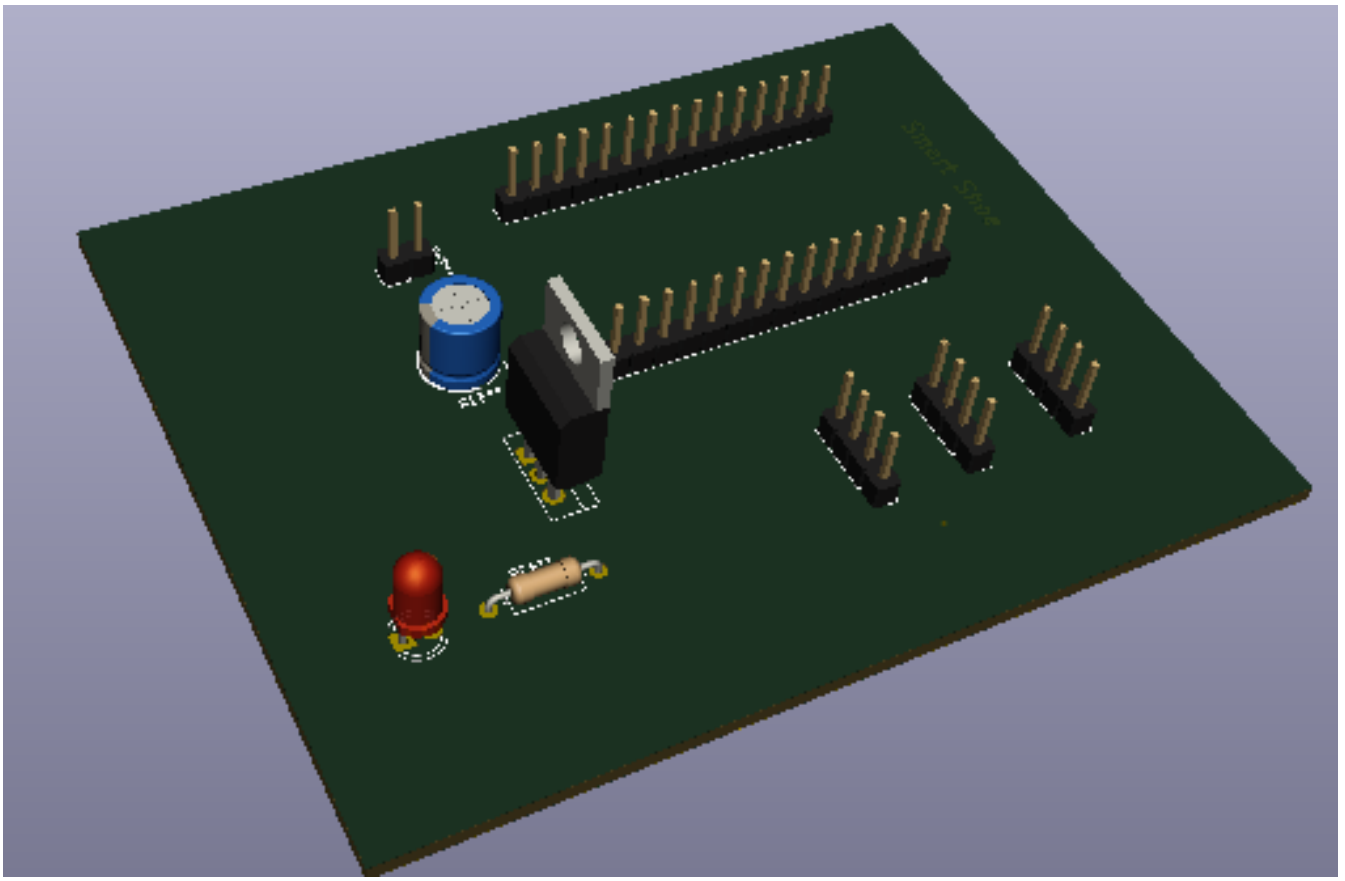

Figure 6.4: 3d view

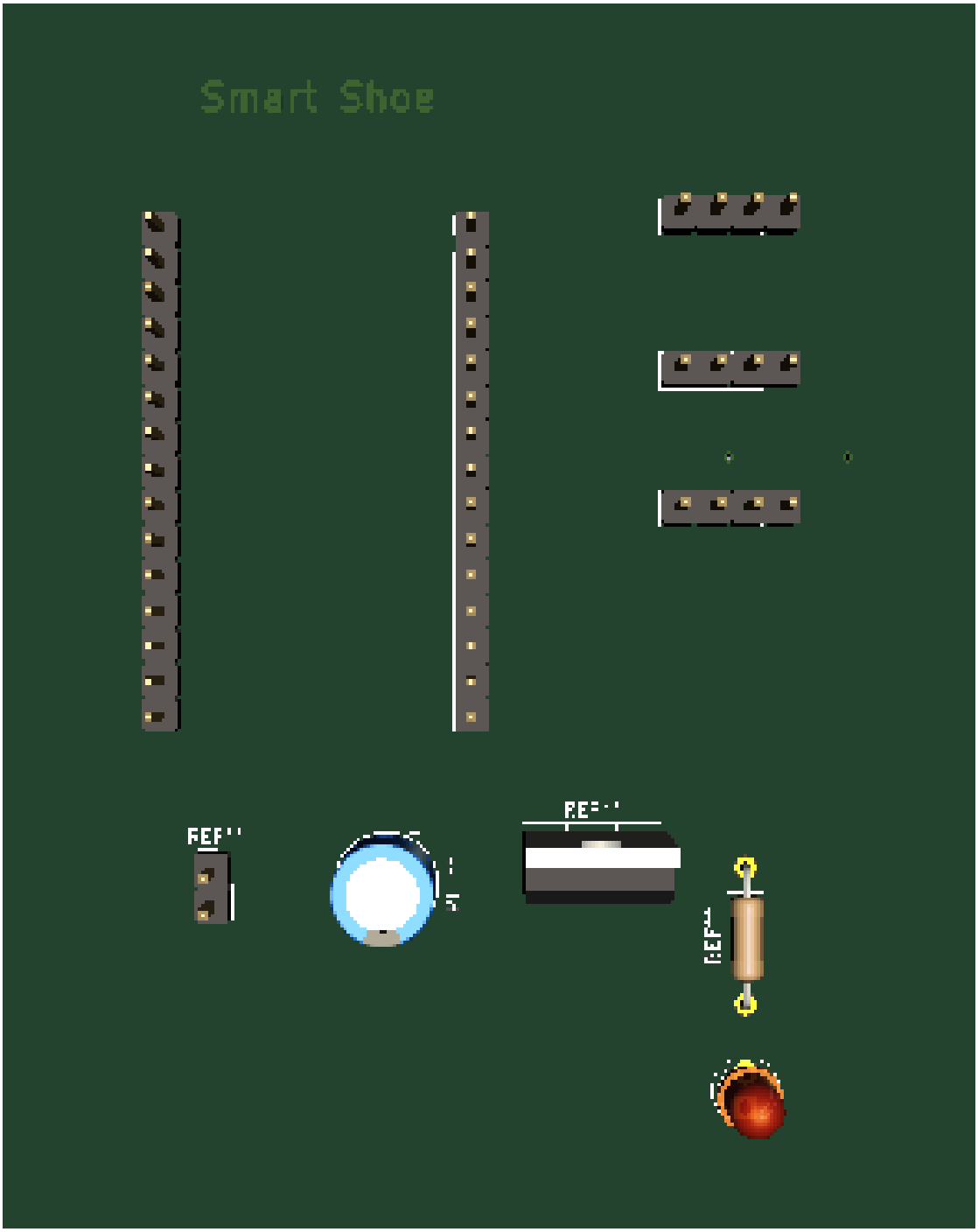

Figure 6.5: 2d view

6.2 REMOTE : Rx 1 / Tx 2 (Receiver Section for Section 1) and (Transmitter Section 2):

- This box is basically referring to remote control which will be held by the blind person.
- The remote will receive the signals from the transmitter i.e Shoe and it acts a receiver in our project.
- Through this remote, the visually impaired will access the smart shoe.
- In this circuit ,we are using Lithium Rechargeable Batteries.
- It consists of HELP SWITCH , RESET SWITCH and POWER SWITCH.
- The HELP switch is used to send the HELP MAIL with the location of the blind person to their family members whenever they need any sort of help and in case of any emergencies.
- The power switch is used to turn on and turn off the circuit. Similarly, the Reset switch is used to reset the circuit.
- The Remote consists of Node MCU , Arduino Nano Controller, Help switch, Battery,Text-to-speech module, speaker.
- In this circuit, all the signals coming from the transmitter will be received by our Node MCU and it will be processed.
- After the signal is processed by Node MCU, it will send these signals to Arduino Nano Controller.
- Then Arduino Nano Controller processes it and transmits these signals to the text-to-speech module.This module will further make the text audible to the visually impaired with the help of speaker.
- In case of any help , we need to press the help switch twice since it is a push up button.

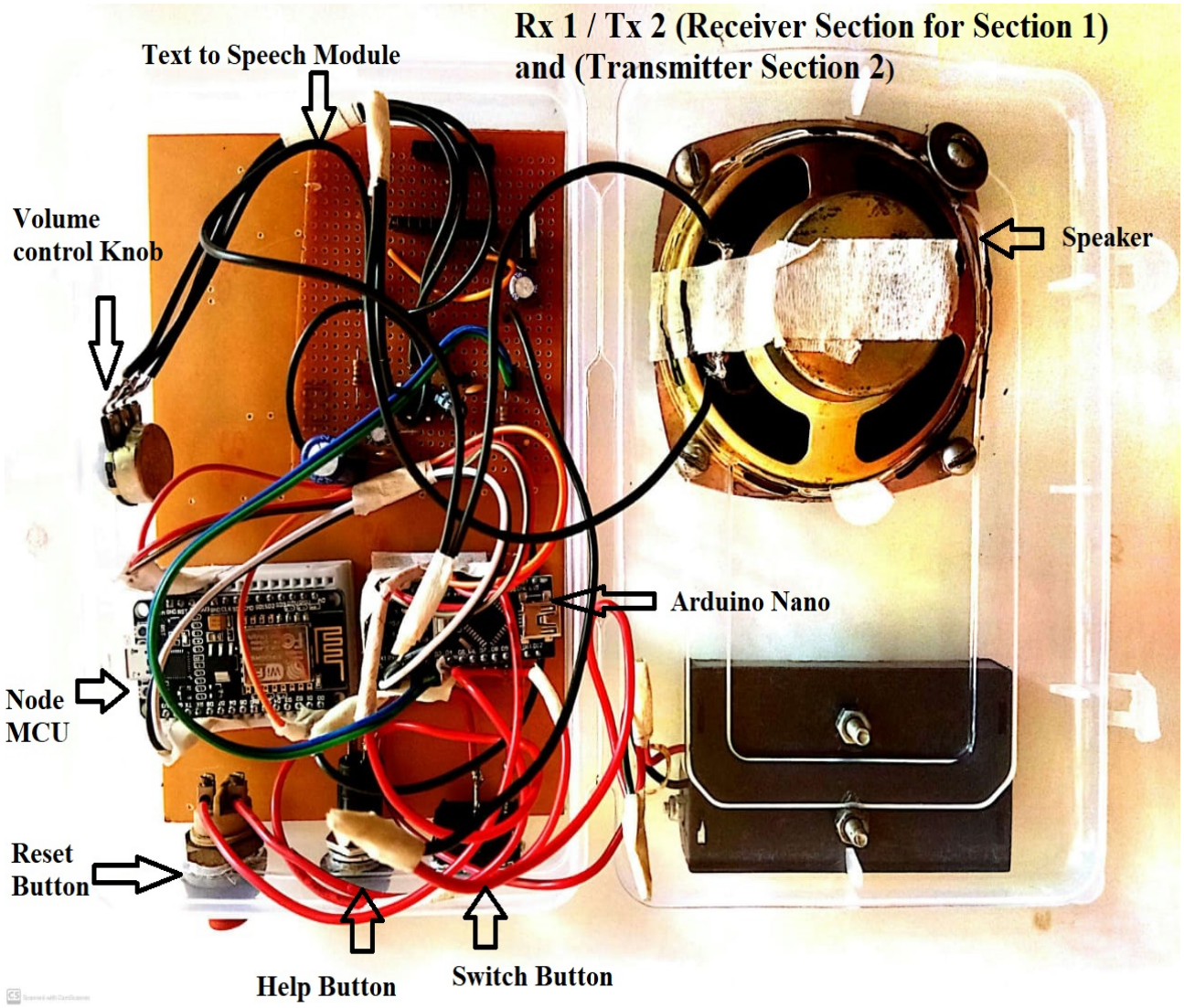

Figure 6.6: Remote

- When we press the button, the help mail will be continuously sent to the family member of the blind person until you release the button.
- After this stage, it requires a reset because after sending the mail the controller drains more power and it can go into flash mode.
- Flash mode means that the code is not able to run , hence we need to reset the controller with the help of reset switch.

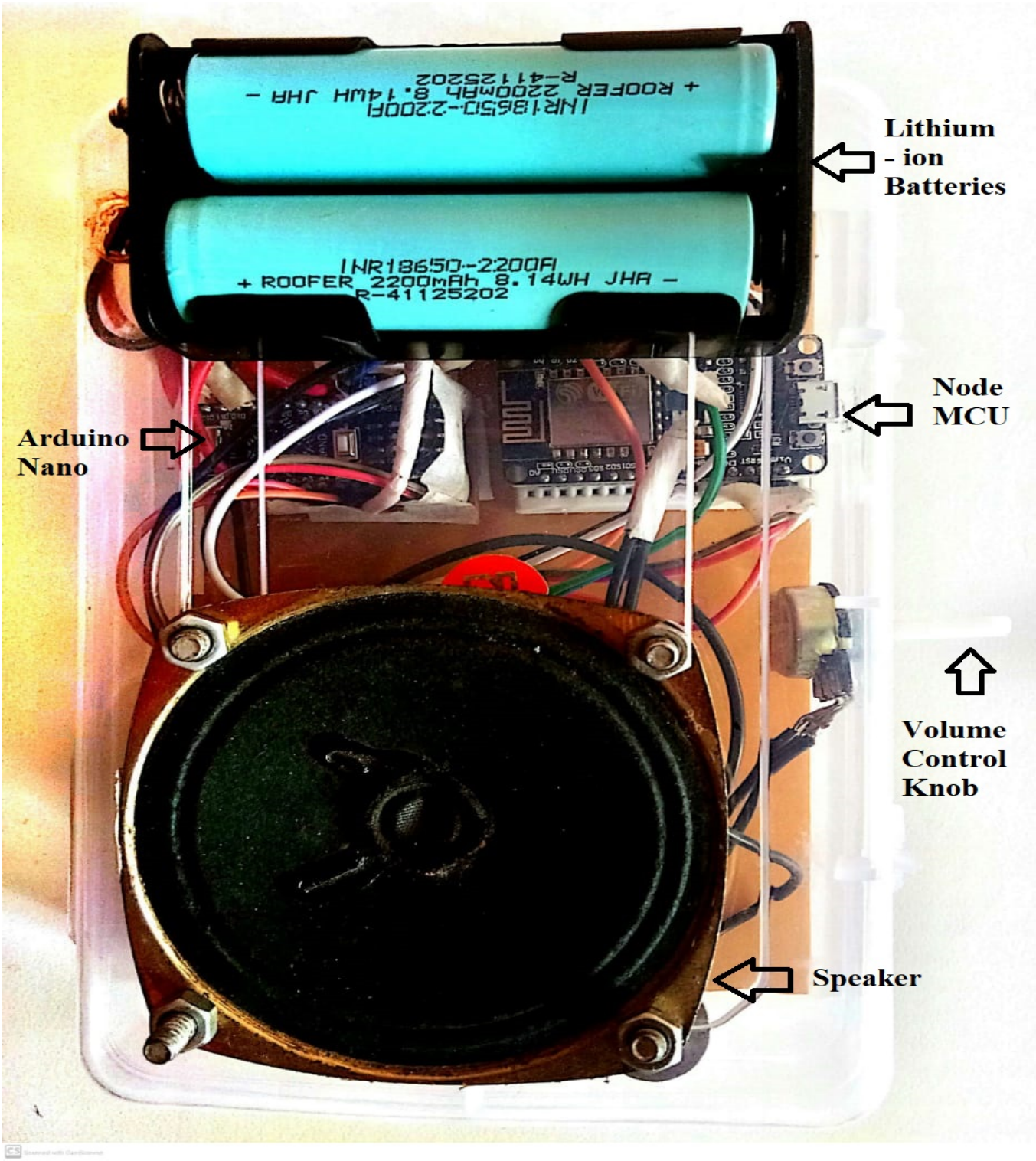

Figure 6.7: Remote

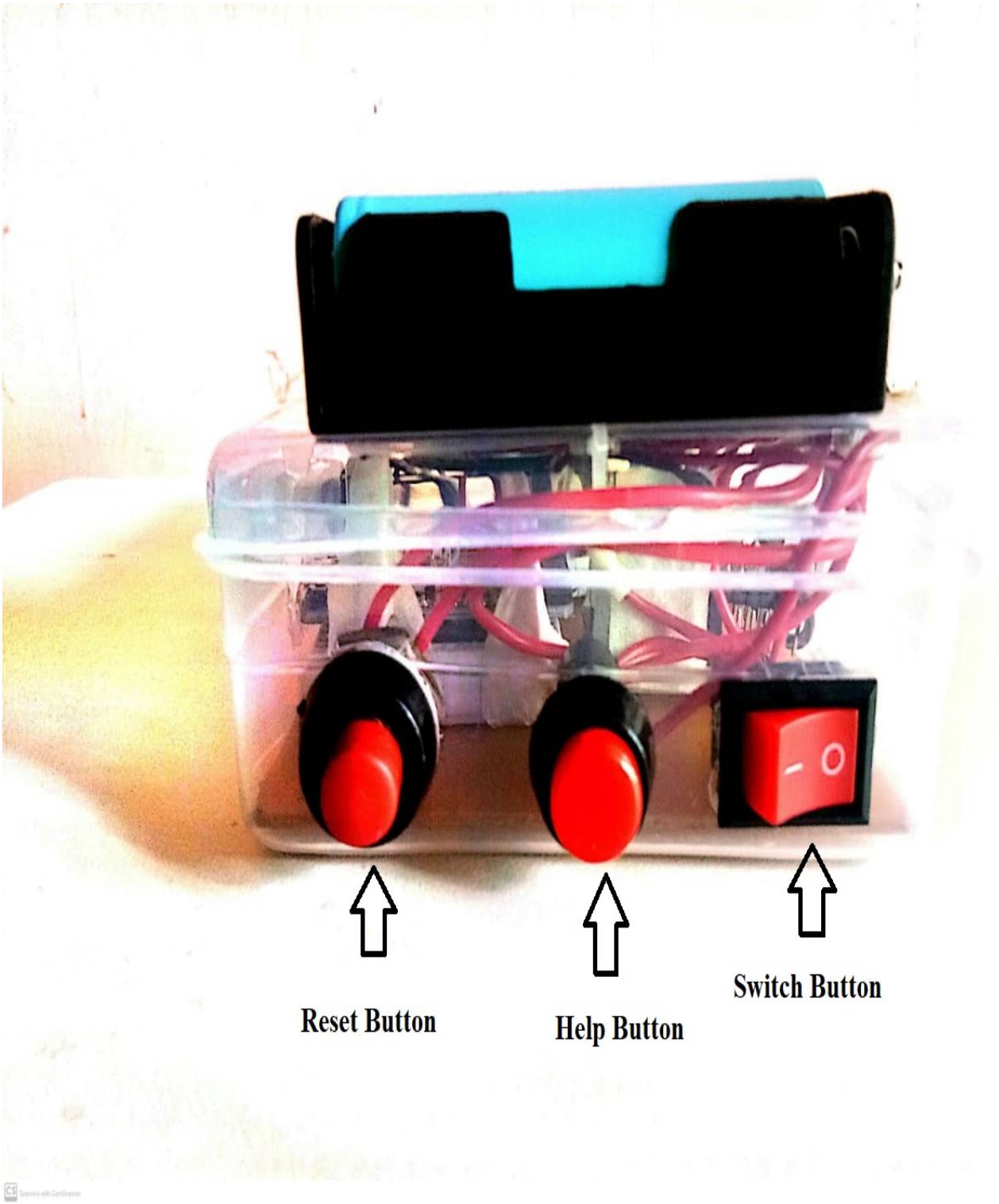

Figure 6.8: Remote Buttons

6.3 BLYNK APPLICATION : Rx 2 (Receiver Section 2):

- *widgetBridge bridge1(v2)* is the command we have used to connect two devices/controllers through which we are sending our data from one controller to another controller. *(v2)* is the virtual pin.
- After this command, we have to write
  BLYNK CONNECTED()
  bridge1.setAuthToken(".....") to the code of transmitter section/shoe controller.
- This token is similar to the token of controller of the remote i.e. this token is for the controller to which we have to send the data.
- So we have connected the shoe controller and remote controller through Blynk application using the "authentication token".

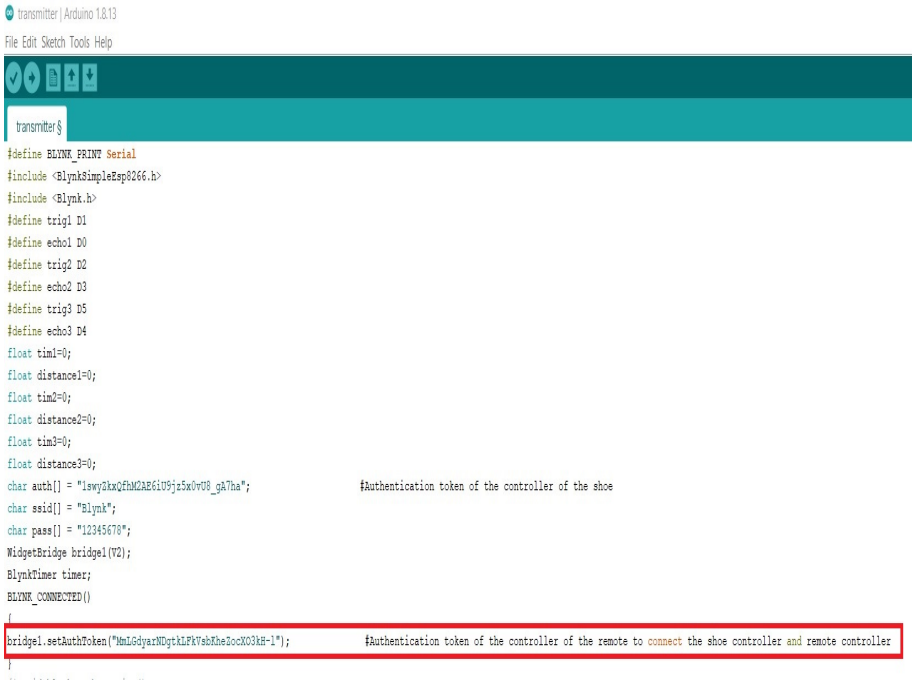

Figure 6.9: code of the transmitter section

Figure 6.10: code of the receiver section

- After pressing the help switch of the remote, the help mail will be send to the relative of the blind person through Blynk application.
- We have to provide the mail ID of the realtive of the blind person in the code.
- After clicking on the link in the help mail, the relative person will get the location of the blind person.

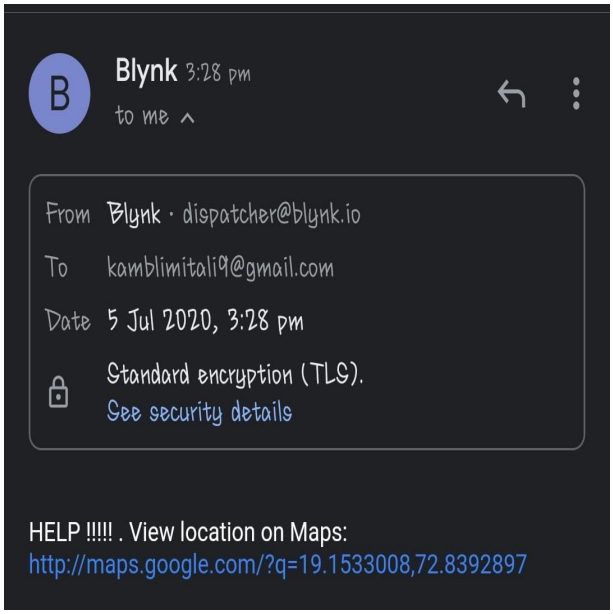

Figure 6.11: Help mail

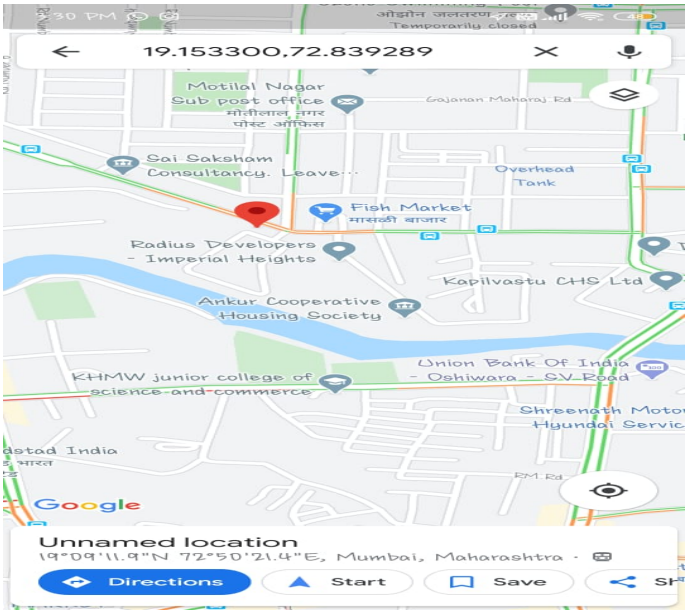

Figure 6.12: Location of the blind person

6.4 IMPLEMENTATION :

- To get connected to the controllers in the circuit, we have to turn on our Wi-Fi.
- All the three sensors would be always detecting for obstacles around as they are programmed to.
- If ultra 1 and ultra 2 doesnt detect obstacles on left/right or front, ultra 3 would keep detecting the ground.
- If the shoe comes to a place where there is no floor detected ahead for example, while walking down the stairs ultra 3 would detect no floor ahead before stepping and send an alert signal to the remote.
- The text-to-speech module in the remote will alert the blind person.
- We need to give external power supply to the remote through power bank because, the lithium-ion batteries in the remote would only provide power supply to the text-to-speech module i.e. for the speaker and not the controller.
- In the remote, we have to insert the cable of power bank into Node-MCU for power supply. After connecting the cable, we switch on the kit.
- We can see the two red lights turned on in the kit after we switched it on. One is the power LED and the other one is the reception LED.
- The reception LED will start blinking as soon as it connects to the network and starts receiving data from the shoe of the blind person wearing it.
- If there are no obstacles around or in front of the blind person, our speaker would keep saying Go.
- Whenever an obstacle is detected it takes 5 seconds to update because, the controller of the shoe is communicating to the controller of the remote and the controller of the remote is forwarding it to the Arduino nano controller which will communicate with the text-to-speech module.

Circuit Diagram:- Text to speech Module:

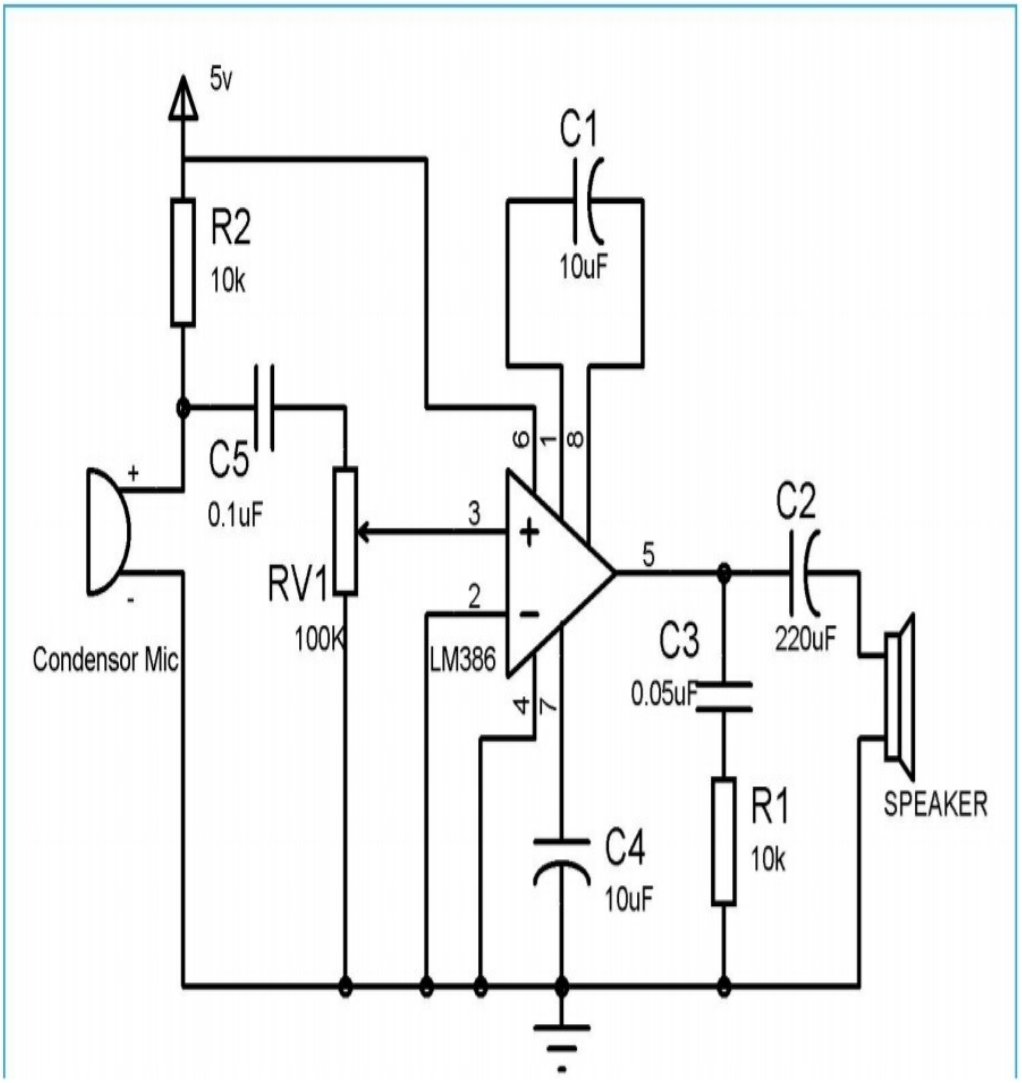

Figure 6.13: Text to Speech module

- Then the text-to-speech module will convert the text into speech and make the text audible through the speaker.
- If ultra1 detects an obstacle on its left/right, it will give output as Not through the speaker. Not means the user can go straight as there is no obstacle in the front of the user.
- If there is an obstacle in front of the user, ultra2 will give output as Left for the user to turn left. If both the sensors i.e. ultra1 and ultra2 detect obstacles in front of them, it will give output as Stop and alert the user to stop.
- The main purpose of ultra3 is to detect the depth.
- As long as it is detecting the floor, it wont give any instructions. But if it detects slope in front of the shoe it will say Alert to the user.
- It is mainly useful while walking down the stairs to alert the user. It is also helpful for pothole detection.
- In case of emergency, the blind person can press the help button on the remote.
- The Blynk app on the phone of the blind person will locate the blind person through the GPS of his phone after he presses the help button.
- The Blynk app will send an emergency help mail to the family with the location it detected of the blind person.
- NOTE: Before Starting the System, we've set the Wifi Parameters as shown Below :
  SSID : Blynk
  Password : 12345678
  Then we've to turn on the Internet and Turn on the Hotspot. And then turn the GPS on and open the blynk application by using the following details.
  Login details of the Blynk app:
  LoginID : kamblimitali9@gmail.com
  Password : mitali

6.5 PCB DESIGNING :

- PCB Designing means Printed Circuit Board used for designing electronic circuit.
- Process of designing PCB is as below:
  - STEP 1: PCB Layout Designing
    - First before getting into designing the designer need a circuit diagram.
    - After a circuit diagram is ready, the 1st step is to design the PCB layout of the circuit diagram which can be made by PCB designing Software.
    - There are many PCB designing Softwares available like KiKad, Eagle, Proteus, etc.
    - We are using KiKad Software for PCB designing.
    - PCB design of the Circuit is shown in the figure below:

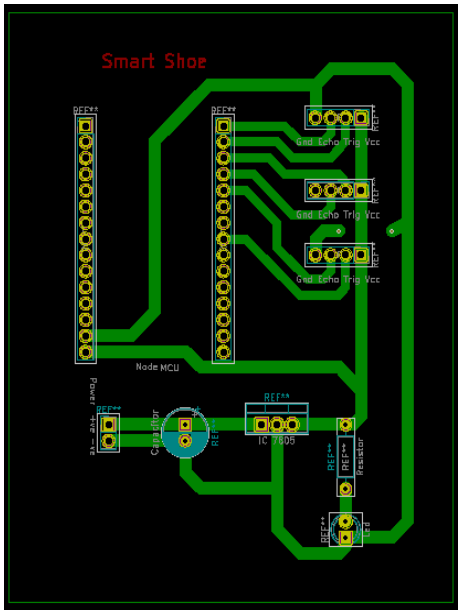

Figure 6.14: pcb design

– STEP 2: PCB Layout Printing

  * PCB layout needs to be printed with Laser printer only (No Ink-jet Printer) on the Glossy paper/Normal White Paper as shown in the figure below:

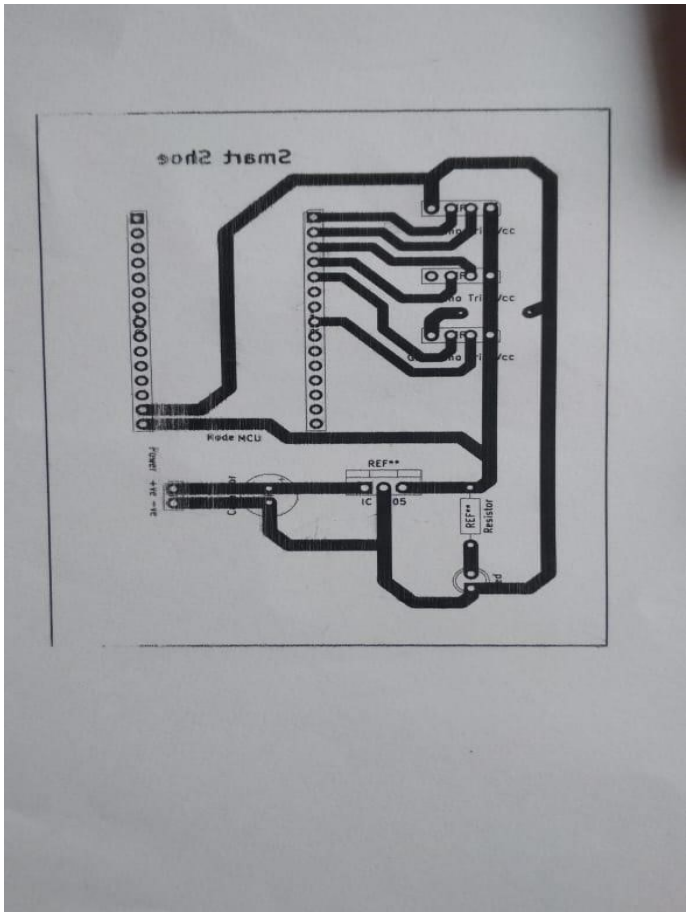

Figure 6.15: pcb design print

– STEP 3: Copper Clad Board

  * Before printing on the copper clad board it needs to first cut into the required size and wash it well before using it as shown in the figure below:

Figure 6.16: Copper Clad Board

– STEP 4: Printing on Board

  * With the help of Iron (Cloths press Iron) press the paper on the board for an half an hour for getting print as shown in the figure:

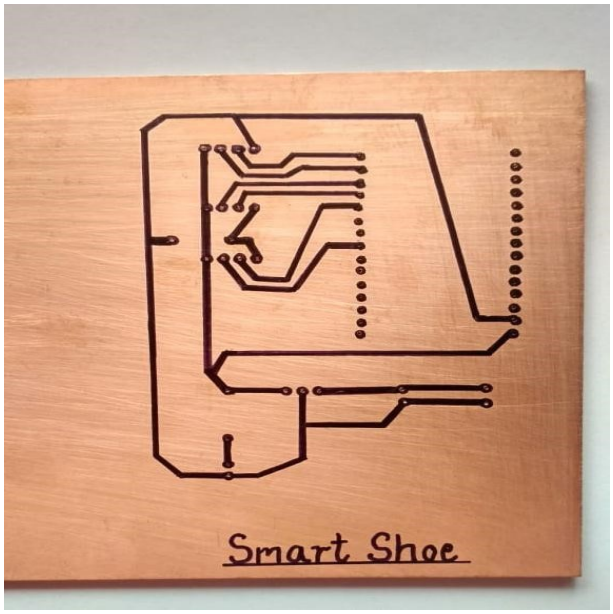

Figure 6.17: Printing On Board

– STEP 5: Etching Process

  * Removing the not require/excess copper from the board without damaging the tracks is called as Etching Process.
  * For this process, take Ferrous Chloride solution into Water (take some amount of water into the tray so that the pcb may fully dipped into that water).
  * Mix the solution well and then Dip the board into it.

  * NOTE : dont dip any conductive material into it, use pen or pencil to remove the PCB as shown in the figure:

Figure 6.18: Adding Water in the tray

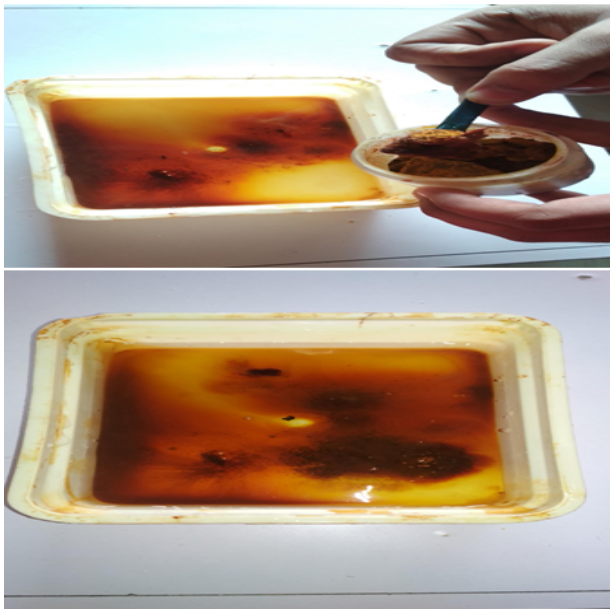

Figure 6.19: Adding Ferrous Chloride Solution/PCB Etching Agent

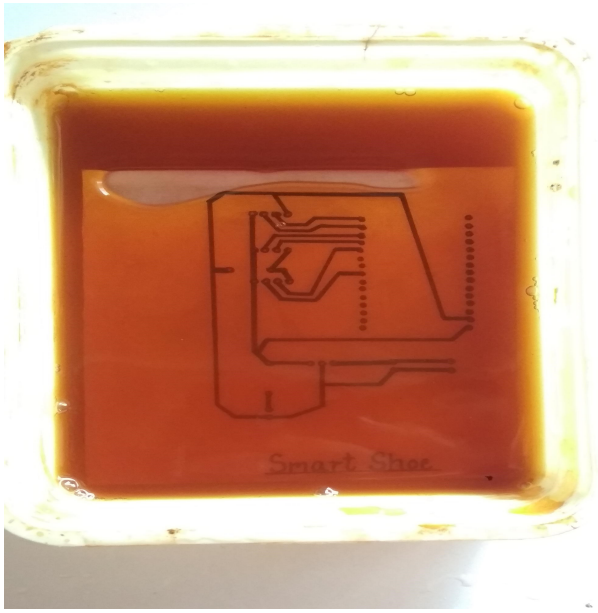

Figure 6.20: Dipping Board into the solution

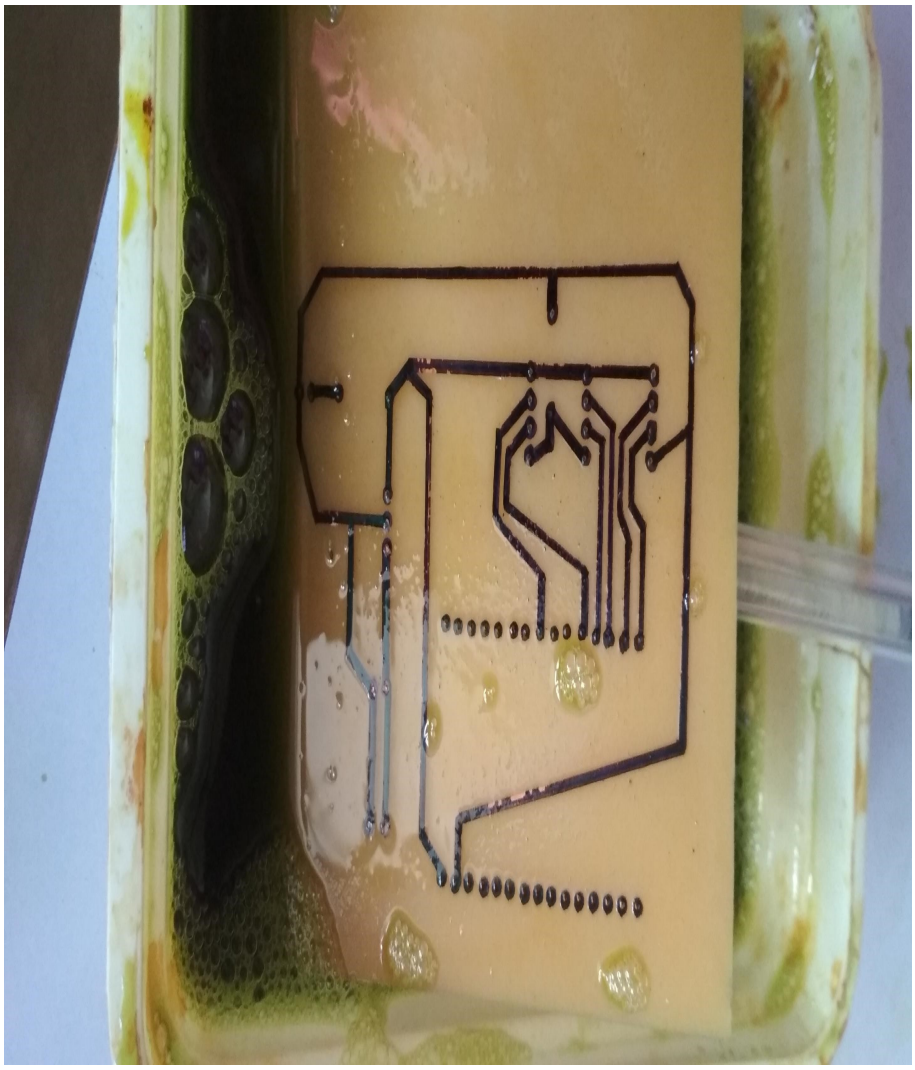

Figure 6.21: Removing Board From the solution as all the copper is Removed from the PCB except the printed part.
(Colour of the solution changes from Yellow/Orange to Greenish. It indicates the one of the sign of completion of the etching Process)

* After that Wash the Board with scrub and water and remove the print ink. You will get board as shown in the figure below:

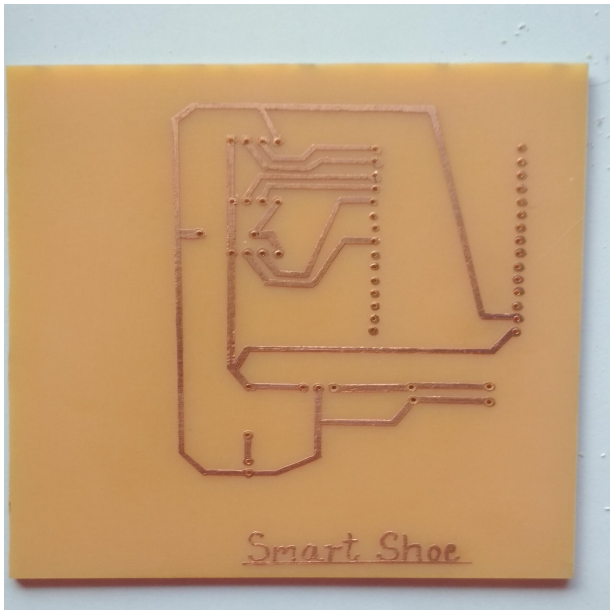

Figure 6.22: The board is ready to use

* After that the next step is drilling the holes of the PCB.

– STEP 6: Testing process

* In the testing process, we need to check the each and every track of the PCB if it is having break anywhere, if so then make it correct with the help of the solder.
* Note: This testing is also called as the Continuity Testing.

– STEP 7: Component Soldering

* Soldering the components as per the circuit diagram as shown in the figure:

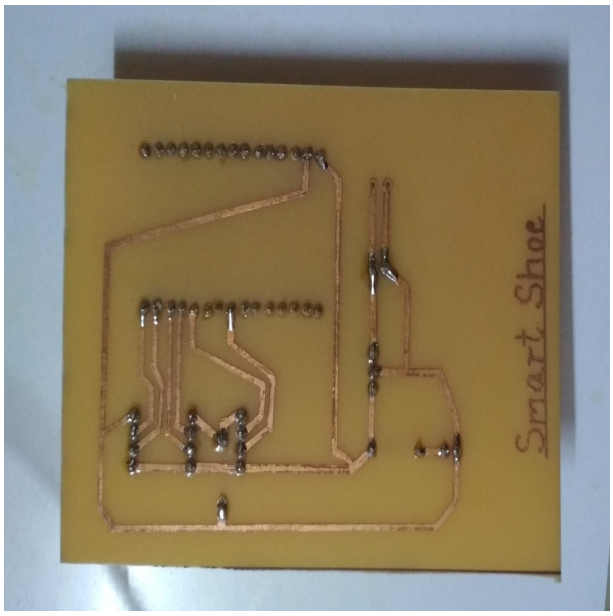

Figure 6.23: Drilling holes to pcb board

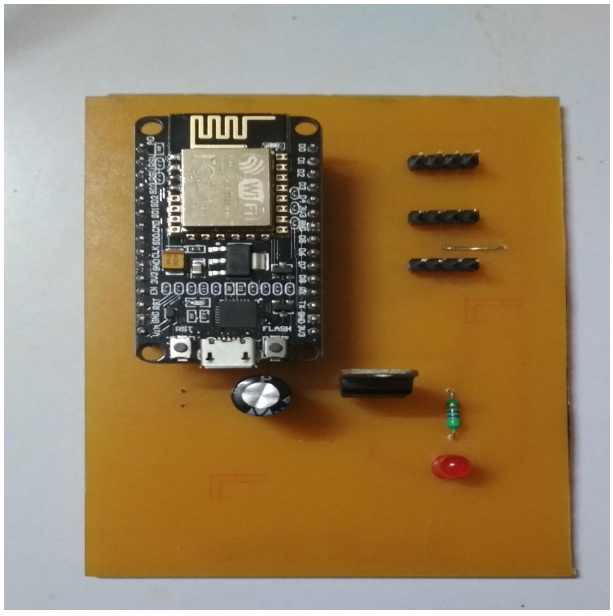

Figure 6.24: Attach the components to pcb board according to the pcb design

Chapter 7

Code :

7.1 Transmitter Section :

define BLYNK_PRINT Serial *//initialization of blynk serial communication*
include ⟨BlynkSimpleEsp8266.h⟩ *//initialization of controller node mcu with blynk*
include ⟨Blynk.h⟩ *//initialization of blynk application*
define trig1 D1 *//defining ultrasonic1 trig pin is conected to D1 of node mcu*
define echo1 D0 *//defining ultrasonic1 echo pin is conected to D0 of node mcu*
define trig2 D2 *// defining ultrasonic2 trig pin is conected to D2 of node mcu*
define echo2 D3 *// defining ultrasonic2 echo pin is conected to D3 of node mcu*
define trig3 D5 *// defining ultrasonic3 trig pin is conected to D5 of node mcu*
define echo3 D4 *// defining ultrasonic3 echo pin is conected to D4 of node mcu*
float tim1=0; *// denoting a variable to store the value of time and making it 0*
float distance1=0; *// denoting a variable to store the value of distance and making it 0*
float tim2=0; *// denoting a variable to store the value of time and making it 0*
float distance2=0; *// denoting a variable to store the value of distance and making it 0*
float tim3=0; *// denoting a variable to store the value of time and making it 0*
float distance3=0; *// denoting a variable to store the value of distance and making it 0*

```
char auth[ ] = "1swyZkxQfhM2AE6iU9jz5x0vU8_gA7ha"; // Auth token for connection of the controller to the blynk app
char ssid[ ] = "Blynk"; // SSID name for the connection of the controller to wifi
char pass[ ] = "12345678"; //pass of ssid for the connection of the controller to wifi
WidgetBridge bridge1(V2); // connecting widget bridge to virtual pin no V2
BlynkTimer timer; // initialization of blynk timer
BLYNK_CONNECTED() // command for starting the connection of bridge
{
bridge1.setAuthToken("MmLGdyarNDgtkLFkVsbKheZocXO3kH-l"); // token of device which is to be connected
}
/* void blynkAnotherDevice()
{
if(distance1≥10&&distance2≤10)
{
bridge1.digitalWrite(D1,HIGH); //if distance1≥10&&distance2≤10 then set the D1 pin of the remote controller High else set it low
delay(100);
}
else
{
bridge1.digitalWrite(D1,LOW);
delay(100);
}
if(distance2≥10&&distance1≤10)
{
bridge1.digitalWrite(D2,HIGH); //if distance2≥10&&distance1≤10 then set the D2 pin of the remote controller High else set it low
delay(100);
}
else
{
```

```
bridge1.digitalWrite(D2,LOW);
delay(100);
}
if(distance1≤10&&distance2≤10)
{
bridge1.digitalWrite(D1,HIGH); //if distance1≤10&&distance2≤10 then set the D1 and D2 pin of the remote controller High else set it low
bridge1.digitalWrite(D2,HIGH);
delay(100);
}
else
{
bridge1.digitalWrite(D1,LOW);
bridge1.digitalWrite(D2,LOW);
delay(100);
}
if(distance3≤40)
{
bridge1.digitalWrite(D3,HIGH); //if distance3≤40 then set the D3 pin of the remote controller High else set it low
delay(100);
}
else
{
bridge1.digitalWrite(D3,LOW);
delay(100);
} }*/
void blynkAnotherDevice() // making a void function
{
if(distance2≤20) //measuring the distance value of ultrasonic for detection of obstacle
{
bridge1.digitalWrite(D1,HIGH); // if obstacle detected then making value of other
```

controller high on pin D1
delay(100); *// wait for 100 millisecond*
}
else if(distance1≤20) *//measuring the distance value of ultrasonic for detection of obstacle*
{
bridge1.digitalWrite(D2,HIGH); *// if obstacle detected then making value of other controller high on pin D2*
delay(100); *// wait for 100 millisecond*
}
else if(distance1≤10&&distance2≤10) *//measuring the distance value of ultrasonic for detection of obstacle*
{
bridge1.digitalWrite(D1,HIGH); *// if obstacle detected then making value of other controller high on pin D1*
bridge1.digitalWrite(D2,HIGH); *// if obstacle detected then making value of other controller high on pin D2*
delay(100); *// Wait for 100 millisecond*
}
else if(distance3≥60) *//measuring the distance value of ultrasonic for detection of obstacle*
{
bridge1.digitalWrite(D3,HIGH); *// if obstacle detected then making value of other controller high on pin D3*
delay(100); *// Wait for 100 millisecond*
}
else *// if none of the above conditions are true then making the value of D1,D2,D3 low of the other controller*
{
bridge1.digitalWrite(D1,LOW);
bridge1.digitalWrite(D2,LOW);
bridge1.digitalWrite(D3,LOW);
delay(100);

```
}
}
void setup() // void setup for defining the mode of the connected pins
{
pinMode(trig1,OUTPUT); // defining trigger pins as output pins of ultrasonic sensor
pinMode(echo1,INPUT);  // defining echo pins as input pins of ultrasonic sensors
pinMode(trig2,OUTPUT);
pinMode(echo2,INPUT);
pinMode(trig3,OUTPUT);
pinMode(echo3,INPUT);
Serial.begin(9600); // defining baudrate speed for serial communication as 9600 bits/sec
Blynk.begin(auth, ssid,pass );// start the connection of the blynk using credentials mentioned above
timer.setInterval(1000L,blynkAnotherDevice); // start the timer of the blynk
}
void loop() // void loop for running the program again and again
{
Blynk.run(); // Start the blynk process
ultra1(); // calling ultrasonic codes for executing from below
ultra2();
ultra3();
timer.run(); // timer started
}
void ultra1() // code for ultrasonic sensor 1
{
digitalWrite(trig1,LOW);  // making trigger pin low
delayMicroseconds(2); // wait 2us
digitalWrite(trig1,HIGH); // making trigger pin high
delayMicroseconds(10);  // wait 2us
digitalWrite(trig1,LOW); //making trigger pin low
delayMicroseconds(2); // wait 2us, this generates the pulses on the trigger pin
```

```
tim1=pulseIn(echo1,HIGH); // counting the pulse and storing in variable
distance1=tim1*0.0340/2; // calculating the distance by formula and storing it
}
void ultra2() //same as ultra 1
{
digitalWrite(trig2,LOW);
delayMicroseconds(2);
digitalWrite(trig2,HIGH);
delayMicroseconds(10);
digitalWrite(trig2,LOW);
delayMicroseconds(2);
tim2=pulseIn(echo2,HIGH);
distance2=tim2*0.0340/2;
}
void ultra3() //same as ultra 1
{
digitalWrite(trig3,LOW);
delayMicroseconds(2);
digitalWrite(trig3,HIGH);
delayMicroseconds(10);
digitalWrite(trig3,LOW);
delayMicroseconds(2);
tim3=pulseIn(echo3,HIGH);
distance3=tim3*0.0340/2; }
```

7.2 Receiver Section :

```
# define BLYNK_PRINT Serial
# include ⟨ESP8266WiFi.h⟩
# include ⟨BlynkSimpleEsp8266.h⟩
# define B D0 // defining the B as button connected on pin no D0
int val=0; // defining a variable int to store the value
char auth[ ] = "MmLGdyarNDgtkLFkVsbKheZocXO3kH-l";
char ssid[ ] = "Blynk";
char pass[ ] = "12345678";
String latitude; // defining string to store the value of latitude String
String longitude; // defining string to store the value of longitude
void setup()
{
Serial.begin(9600);
pinMode(D1,INPUT);
pinMode(D2,INPUT);
pinMode(D3,INPUT);
PinMode(B,INPUT);
Blynk.begin(auth,ssid,pass);
}
BLYNK_WRITE(V6) //connection of GPS widget on virtual pin no V6
{
GpsParam gps(param); //initialising the GPS
Blynk.virtualWrite(V6, param[0].asString()); //Making a pointer [0] to store data
on same pin V6 of GPS
Blynk.virtualWrite(V6, param[1].asString()); //Making a pointer [1] to store data
on same pin V6 of GPS
latitude = param[0].asString(); // store the value of pointer[0] in variable
longitude = param[1].asString(); // store the value of pointer[1] in variable
// Print 6 decimal places for Lat, Lon
Serial.print("Lat: "); // serial print all the data on serial monitor
Serial.println(gps.getLat(), 7);
```

```
Serial.print("Lon: ");
Serial.println(gps.getLon(), 7);
}
void loop()
{ Blynk.run();
If(B==HIGH) // if condition is true then enter the loop
{
if(digitalRead(D1)==HIGH && digitalRead(D2) == LOW) // if D1 and D2 is high then enter the loop
{
Serial.write(1); // serial writing 1 value for nano controller for text to speech conversion
Serial.println("Walk Ahead"); //print data on serial monitor
delay(100);
}
else if(digitalRead(D2)==HIGH && digitalRead(D1) == LOW) // if condition is true then enter the loop
{
Serial.write(2); // serial writing 2 value for nano controller for text to speech conversion
Serial.println("Turn Left"); // print data on serial monitor
delay(100);
}
else if(digitalRead(D3)==HIGH) // if condition is true then enter the loop
{
Serial.write(3); // serial writing 3 value for nano controller for text to speech conversion
Serial.println("Slope Ahead"); // print data on serial monitor
delay(100);
}
else if((digitalRead(D1)==HIGH && digitalRead(D2) == HIGH) ||
(digitalRead(D2)==HIGH && digitalRead(D1) == HIGH) ) // if condition is true then enter the loop
```

```
{
Serial.write(4); // serial writing 4 value for nano controller for text to speech conversion
Serial.println("Stop"); // print data on serial monitor
delay(100);
}
else // if none of the above condition is true then enter the loop
{
Serial.write(5); // serial writing 5 value for nano controller for text to speech conversion
Serial.println("All ok"); // print data on serial monito
delay(100);
}
}
else
{
Serial.print("Help"); // print data on serial monitor
String toSend; // defining string to store the value
toSend += " HELP !!!!! ";
toSend += ". View location on Maps: ";
toSend += "http://maps.google.com/?q=";
toSend += latitude; toSend += ",";
toSend += longitude; // writing the values in the string
Blynk.email("kamblimitali9@gmail.com", "Subject: Smart Shoe", toSend); // send mail command
}
}
```

7.3 Arduino Nano (Remote Section) :

```
# include "Talkie.h" //Header file for text to speech Module
# include "Vocab_US_Large.h" //Header file for US accent speech
# include "Vocab_Special.h" // Header file for tts library
# include "Vocab_Soundbites.h"
int value1=0; // defining the value of variable int to store value
Talkie voice; // initialising tts module
void setup()
{
Serial.begin(9600);
}
void loop()
{
if(Serial.available() ⟩ 0) // if serial data is available then enter the loop
{
value1=Serial.read(); // read the incoming value
if(value1 ==1) // if condition is true then enter the loop
{
// Serial.println("Walk Ahead");
voice.say(sp2_NORTH); // spell the written word (sp2,sp3,sp4 are the speed of
saying word)
}
else if(value1 ==2) // if condition is true then enter the loop
{
//Serial.println("Turn Left");
voice.say(sp2_LEFT); // spell the written word
}
else if(value1 ==3) // if condition is true then enter the loop
{
// Serial.println("Slope Ahead");
voice.say(sp2_ALERT); // spell the written word
}
```

```
else if(value1 ==4) // if condition is true then enter the loop
{
//Serial.println("Stop");
voice.say(sp2_STOP); // spell the written word
}
else if(value1 ==5) // if condition is true then enter the loop
{
//Serial.println("All ok");
voice.say(sp2_GO); // spell the written word
}
}
}
```

Chapter 8

Concept of working

8.1 Tx 1 (Transmitter Section 1) :

In this section the used components are ultrasonic sensors, Controller/NodeMcu, GPS module and Battery (For Power Supply).This section will be mounted on the shoe of the blind person.In this circuit we have three ultrasonic sensor as ultra 1,ultra 2 and ultra 3 shown in the figure above. Ultra1 is used for detecting the obstacle in left /Right of the blind person and ultra 2 is used for detecting the obstacle in the Front of the blind person and ultra 3 is used to detected the Path ahead (Hole,Pothole or deep Step) in front of the blind person.This ultrasonic sensors has two pins Trig and Echo .Trig pin is used to send the pulses to the sensor from the controller and echo pin is used to detect the obstacle (when the obstacle is present the wave is transmitted form trig and will touch the obstacle and return back and will be received by the echo pin) this distance will be measured by controller. Gps Module is connected to the controller which will give the location of the Shoe (indirectly the location of the blind person).

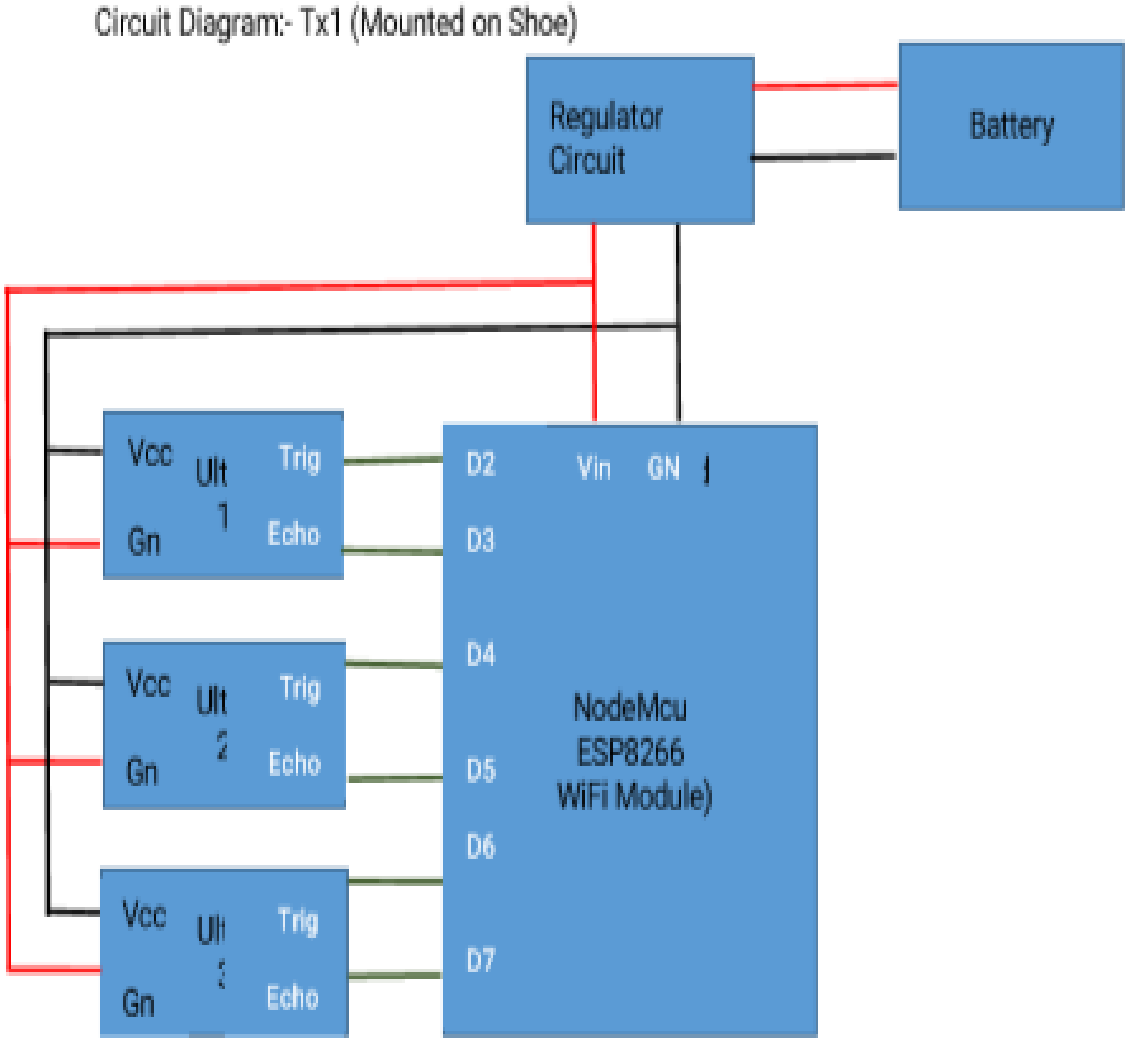

Figure 8.1: Tx1

8.2 Rx 1 / Tx 2 (Receiver Section for Section 1) and (Transmitter Section 2):

This section will receive the details from the Transmitter Section for further processing.This data will be transmitted wireless/via Internet.If the obstacle is present in any of the three ultrasonic sensors this will detected by the controller and the controller will give commands to text to speech module as Turn Right/Left if the obstacle is present at left /Front or vice- versa and if Stop Go back if the all side obstacle is detected and Stop if only step and no obstacle detected .In this section we have one Help switch used for the blind person if he/she presses this switch then a help mail is send on their family members mail id with the location(via Blynk app) of the blind person.

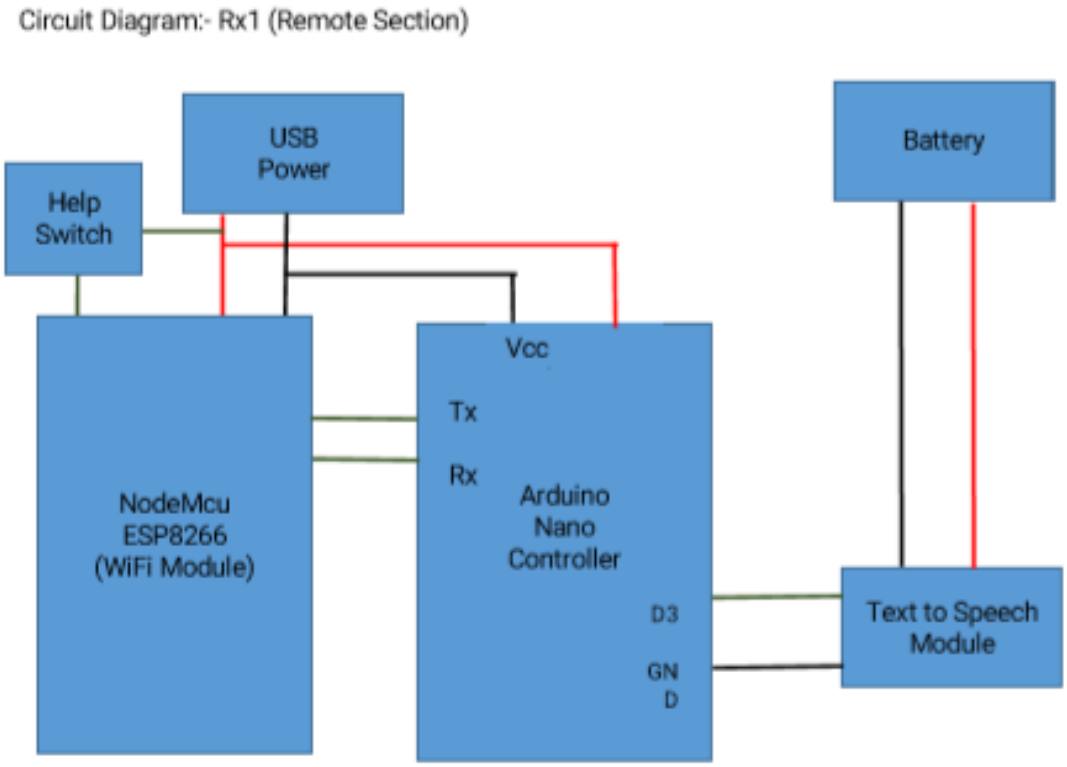

Figure 8.2: Tx2

8.3 Rx 2 (Receiver 2):

This section is a moblie app Blynk used to support the IOT For sending data form one section to other and the help Mail.

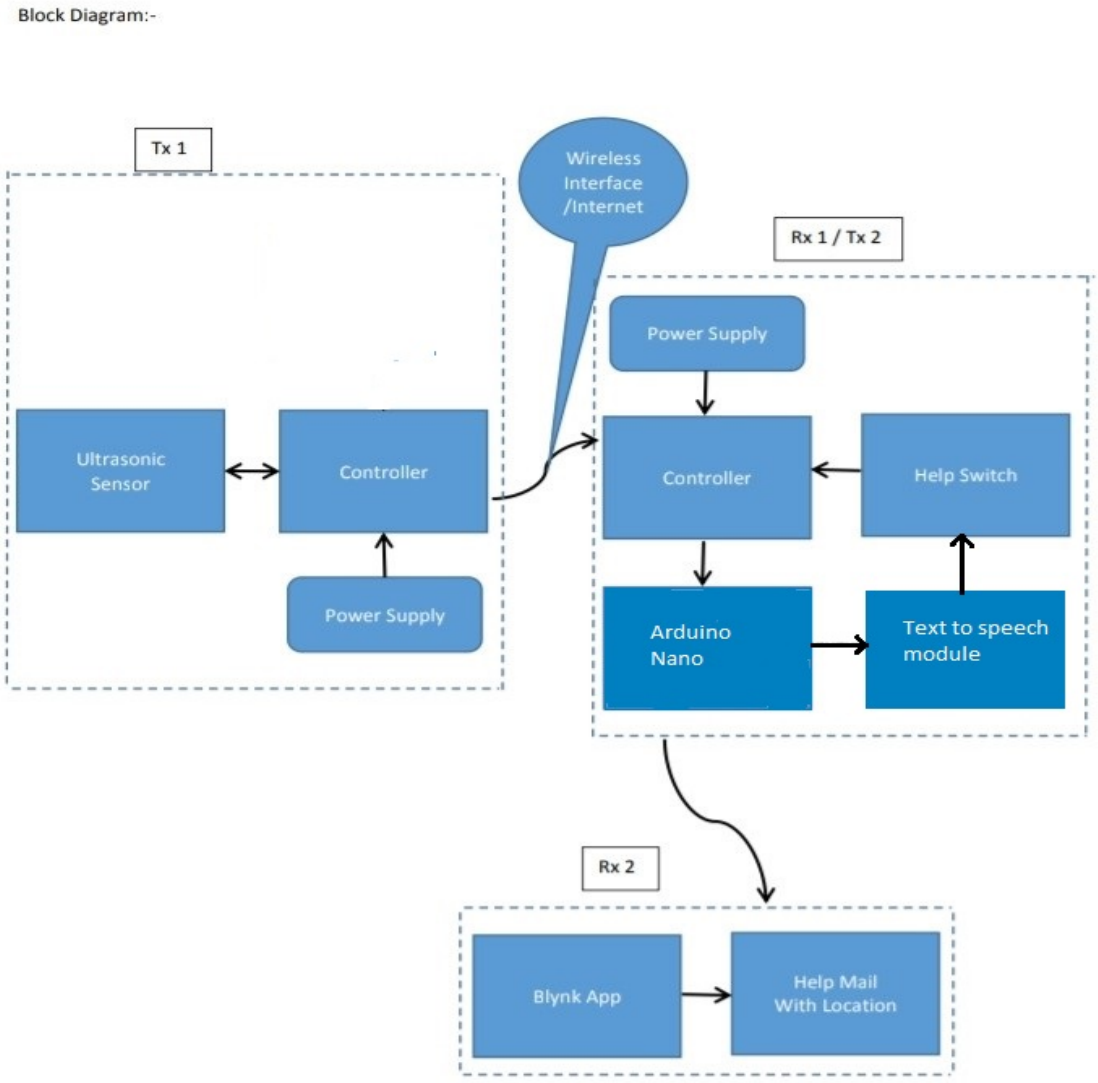

Figure 8.3: block diagram

Chapter 9

Future Scope And Conclusion

9.1 Future Scope :

In our system we can make the following advancement in the future which are as follows :

- We can add Iot based Health monitoring of the blind person : This system can be integrated with our current system in future.
- We can also make prosthetic leg rather than shoe which will be a robotic prosthetic leg for the people who are blind and handicapped (leg) as well,which will be helpful for them to not only for detecting obstacles but also helpful in walking by their own.
- We can also make a system in which a GPS location of the destination has to written and the Smart Shoe for blind person will automatically tell the blind person about that route without getting touch by any one i.e safely to the destination. This system can be easily integrated with our project with some minimum changes.
- We can add more help switch in the system for the blind person in order to not only getting help from the family via mail with location but also from police, ambulance and some community.

- The main focus of integration of this system with our system is to provide more security to the visually impaired person.

9.2 Conclusion :

- In order to make use of latest technology, we have proposed a project of IOT based smart blind shoe system for the blind preson.
- We have designed a wearable shoe on which electronic kit is proposed.
- Main goal of this proposed system is to provide navigation assistance for this visually impaired person and security to the blind person.
- Sensors will detect obstacles and remote system will announce the commands according direction.
- HELP switch is also mounted on the remote for getting a help to the blind person by the family i.e mail will be sent to the family member's mail id with location of that person.
- Our approach is to make easy application to make visually impaired person to live independently.

References

- Shubham Rastogi (2017), Smart Assistive Shoe, Department of Electronics and Communication, Moradabad — Visual Impairment — Ultrasound. [online] Scribd.
  Available at: https://www.scribd.com/document/344087375/Smart-Assistive-Shoes-for-the-Blind-People [Accessed 2 Jul. 2019].

- Hrushikesh Gosavi (2015), Obstacle Detection For Visually Impaired, International Research JournalIn Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility, ACM (2006), pp- 293294.

- Mohd Helmy Abd Wahab, Amirul A. Talib, Herdawatie A. Kadir, Smart Cane: Assistive Cane for Visually-impaired People, Journal: IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011.

Acknowledgement

We would like to thank our project guide Prof. Sujata Kullur for her enormous co-operation and guidance. We express our gratitude for their valuable time while making this project. All the inputs given by her have found a place in the project. The technical guidance provide by her has been benefitial and made the project successful. We are also thankful to our principle Dr.Sanjay Pawar, our HOD Dr. Sanjay Shitole and all the staff members of Information Technology department who have provided us various faciitiesand guided us to develop this project. Finally, we would like to thank the teachers of our college and friends who have helped us while working on this project.

Sonali Jadhav

Mitali Kambli

Priyanka Mogidala