Smart Navigation Cane for Visually Impaired People

By

Capt Aamir Sajjad

Capt Fahad Shafique

Capt M. Sohail Mushtaq

Capt Jawad Ijaz

Submitted to the Faculty of Department of Electrical Engineering,

Military College of Signals, National University of Sciences and Technology, Islamabad

in partial fulfillment for the requirements of B.E Degree in

Electrical Engineering

JUNE 2018

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

“Smart Navigation Cane for visually impaired people”

is carried out by

Capt Aamir Sajjad, Capt Fahad Shafique, Capt Sohail Mushtaq, Capt Jawad Ijaz

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical Engineering from National University of Sciences and Technology (NUST), Islamabad.

Approved By:

Signature: ______________________________

Supervisor: **Asst Prof Dr. Col Adil Masood**

MCS, Rawalpindi

ABSTRACT

Smart Navigation Cane for visually impaired people

Currently, visually impaired individuals use a standard cane as a tool for steering them after they move from one place to a different. Although, the standard cane is the most widespread cane employed these days by the visually impaired individuals, it couldn't facilitate them to find dangers from all levels of obstacles. Keeping that in view, we have a tendency to propose a replacement intelligent system for guiding people. World Health Organization square measure visually impaired or partly clear-sighted. The system is employed to alter visually impaired folks to maneuver with constant ease and confidence as a clear-sighted folks. Additionally, the system helps in police investigation the potholes. The system is coupled with a GPS module to pin-point the placement of the visually impaired person. Moreover, the compass can offer the direction info furthermore gives heading angle info. A buzzer and voice recognition module also are value-added to the system. The entire system is meant to be tiny, lightweight and is employed in conjunction with the cane. The results have shown that the blinds that used this method may move safely

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of the sole effort of our group, comprising of **Capt Aamir Sajjad, Capt Fahad Shafique, Capt Sohail Mushtaq, Capt Jawad Ijaz**. No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere, nor does it include any verbatim of the published resources which could be treated as a violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

Dedicated to all those

who lead us on the journey

from ignorance to knowledge.

ACKNOWLEDGEMENTS

All of our gratitude is extended to Almighty Allah, the most beneficent, whose blessings empowered us to tread this journey.

We feel indebted to the benign faculty of Department of Electrical Engineering, who raised us to these standards of knowledge; and especially to Asst. prof. Dr Col Adil Masood, whose unending support, motivation and guidance made us capable of turning the concept of this project into reality.

We also extend our thankfulness to our families and friends, who kept our moral spirits high in the times of need and without their unending support and guidance, completion of this project was impossible.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Background

1. Globally, the amount of individuals of all ages living with sight loss is calculable to be 285 million,out of them thirty-nine million are visually impaired in step with the globe Health Organization (WHO). Among several constraints envisaged by a visually impaired person, the challenge of navigation is very critical. Usually visually impaired individuals think about help of argus-eyed persons to search out their means or would like to have Nursing assistant for the person to follow a specific path. This suggests that the bulk of visually impaired individuals cannot notice their means autonomously in an unknown space. Generally visionless persons use a white cane or walking cane.
2. In Electronic destined technology, energy waves square measure emitted ahead, then it's mirrored from obstacles, additionally to the present, the sensible cane are coupled with the GPS system is employed by visually impaired persons to work out and verify the current location and with the assistance of compass can advance its heading angle towards its next points.

1.2 Overview

This project is basically a prototype that comprises of:

1. A Stick. To hold the whole circuit and sensors in place.
2. Raspberry pi. To overall controls and integrate all the components.
3. Ultrasonic Sensors. For detection of obstacle.
4. Speech IC. To give voice output feedback.
5. GPS . To determine the position of blind people.
6. Magnetometer. To determine the direction of movement towards next waypoint.
7. Vibrator. To give vibration feedback.

8 .Rechargeable Battery. To power up the whole system

1.3 Problem Statement

Way finding for blind people is a really big issue now-a-days. There are machines and sticks available for them but they are either not so accurate or they only work inside and also very costly. So, for this problem we decided to design a compact stick in sleek design with more accuracy plus efficiency.

1.4Approach

The basic design approach of this project is that there will be two parts of hardware. One will comprise of the programming and algorithmic structure of Raspberry pi, and the other will be the integration of Raspberry Pi with the other components (GPS, magnetometer, ultrasonic Sensors) of the system.

1.5 Objectives

Our project will be able:

a. To detect different hurdles or obstacles with the help of ultrasonic sensors and hence making it very easy and safe for blind people to find the right way.
b. To navigate through multiple waypoints to reach the destination safely.
c. In case the obstacle is detected he can easily deviate and gets to its right path for further movement towards his destination.

1.6 Academic Objectives

Our project will help us to enhance and polish our practical skills on:

a. Ultrasonic Sensors
b. Raspberry pi
c. GPS module
d. Magnetometer

Chapter 2: Background

2.1 Overview of Existing Literature:

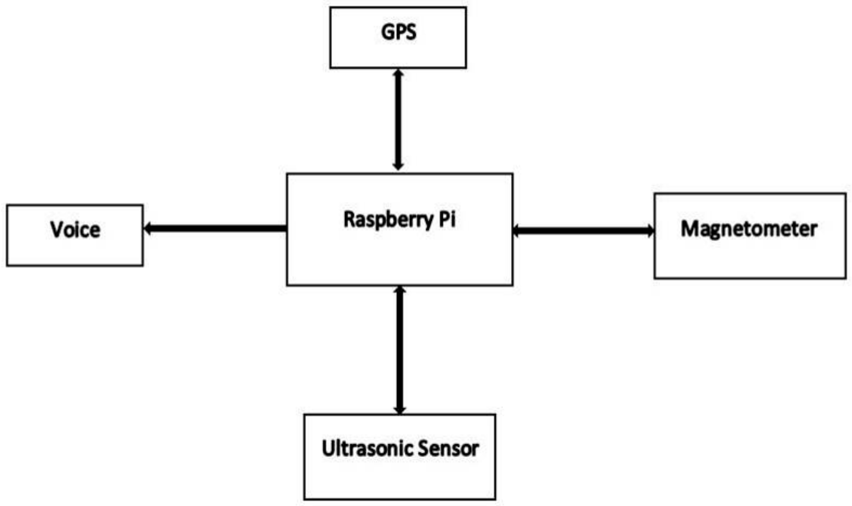
A number of navigation systems for aiding the blinds are developed already. These developed systems are classified into 2 categories. The first cluster is Electronic Travel Aids (ETAs) and the second cluster is Electronic Orientation Aids (EOAs). ETAs are designed to form a secure journey by determining the hurdles by using supersonic and proximity sensing element. EOAs are designed to sight desired destination with the help of GPS and placement of service which is based on location. A stick for the blind people was developed which works with the help of GPS, integrated with ultrasonic and proximity sensing element for determining the obstacle.

2.2 Problem Formulation:

Way finding and navigation up till destination for blind people is a really big issue now-a-days. There are machines and sticks available for them but they only detect the obstacle or can give the current location of blind person but cannot guide the individual towards his destination. So, for this problem we decided to design a compact stick in sleek design that is capable to navigate the visually impaired person using multiple waypoints towards his destination

Chapter 3: Design

3.1 System Architecture:



3.2 Design Specifications:

Following are the specification summary of the proposed design:

3.2.1 Raspberry Pi :

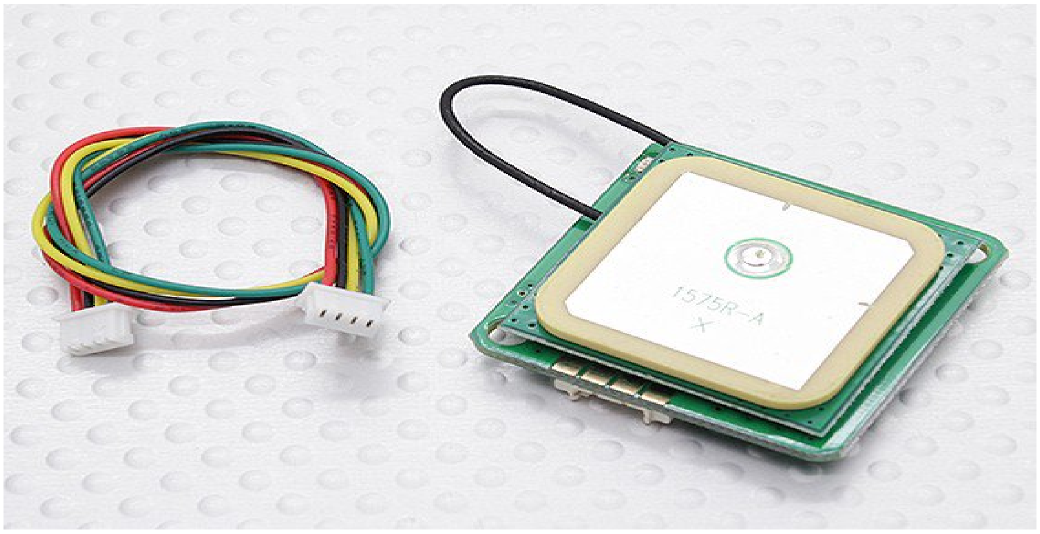
Raspberry is a brain of our prototype.

- Handheld computer with powerful processing speed.
- Less power consumption.
- Collect the data from all components.
- Process the data collected to generate alert and sends the alert as voice output

3.2.2 Magnetometer

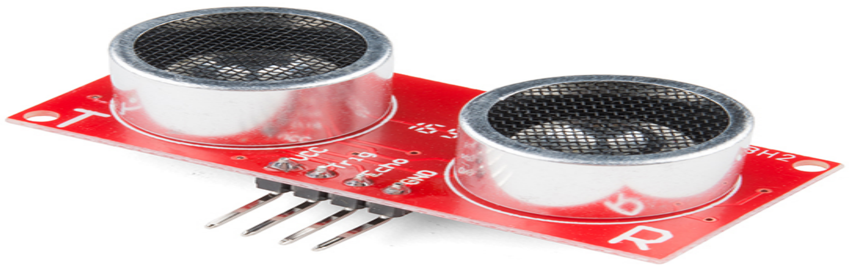
- The Compass Module 3-Axis HMC5883L is a low- magnetic field sensing device with a digital border.
- The compass module transforms any magnetic field into the output of a differential voltage on three axes.
- This voltage shift is raw digital output which can then be utilized to calculate headings or sense magnetic fields coming back from completely different directions.
- The module is meant to be used with a huge variety ofmicrocontrollers with altogether different voltage needs.

3.2.3 GPS (NEO 6M)



- It is aGPS module that may be used with Respberry Pi.
- GPS module provides the best available location information, giving better performance when integrated with Raspberry Pi.
- Operating temperature range: -40 TO 85°C.
- Module can save the data when the power is shut down

3.2.4 US SENSOR



- This is an HC-SR04 inaudible sensing element.
- This economical sensing element gives (3cm to 400cm) a non-contact measurement practicality.

- Each HC-SR04 module includes associate degree US sensing transmitter, a receiver and a controlling circuit.
- It has four pins: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground).

3.3 Special Skills Required

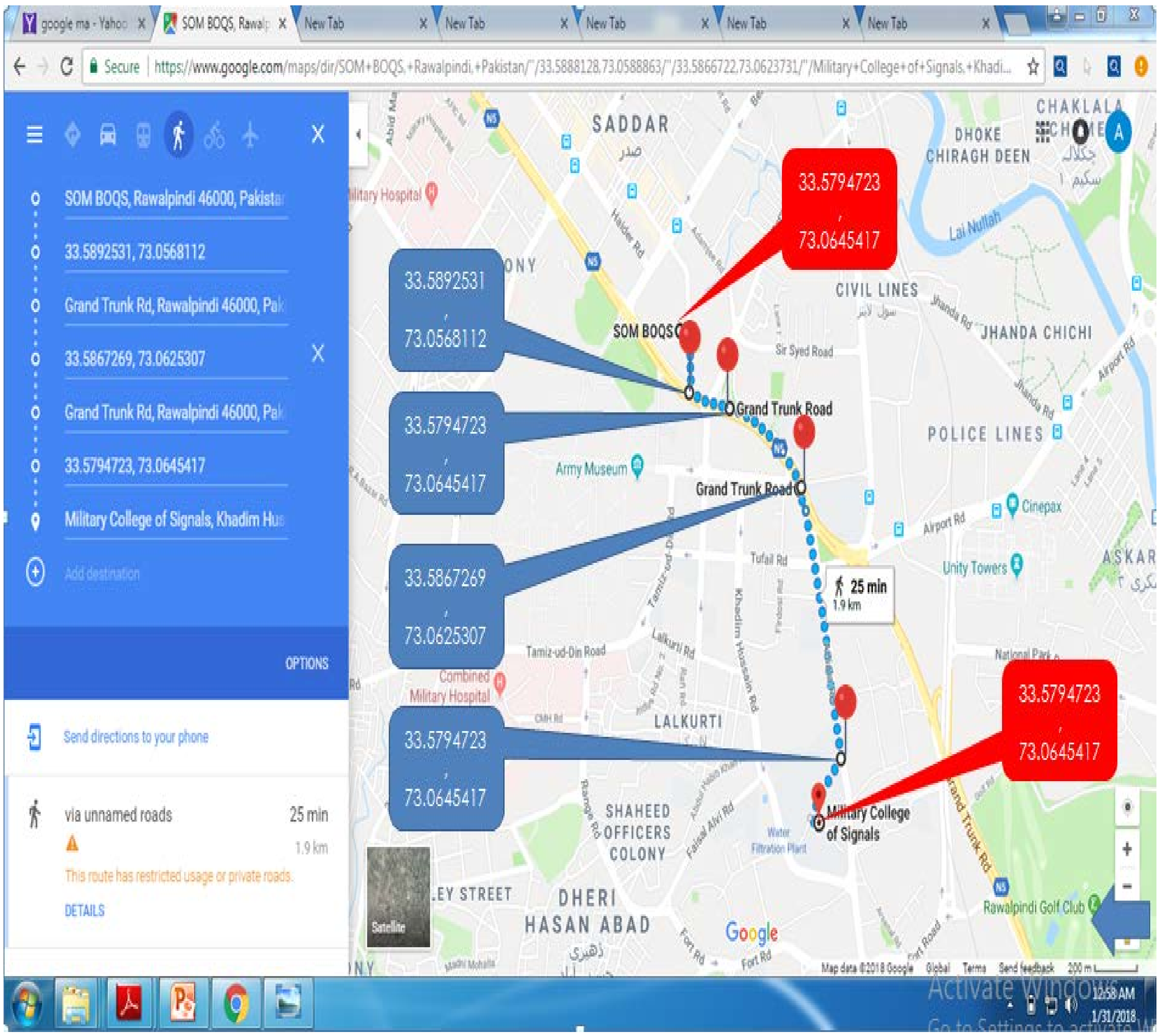
The following specialized skills are required to work on this project:

- Raspberry pi
- Python Programming

Chapter 4: Development of Algorithm

4.1 Entering the Way Points

- Selection of Route e.g. SOM to MCS
- Division of route into Multiple Way Points



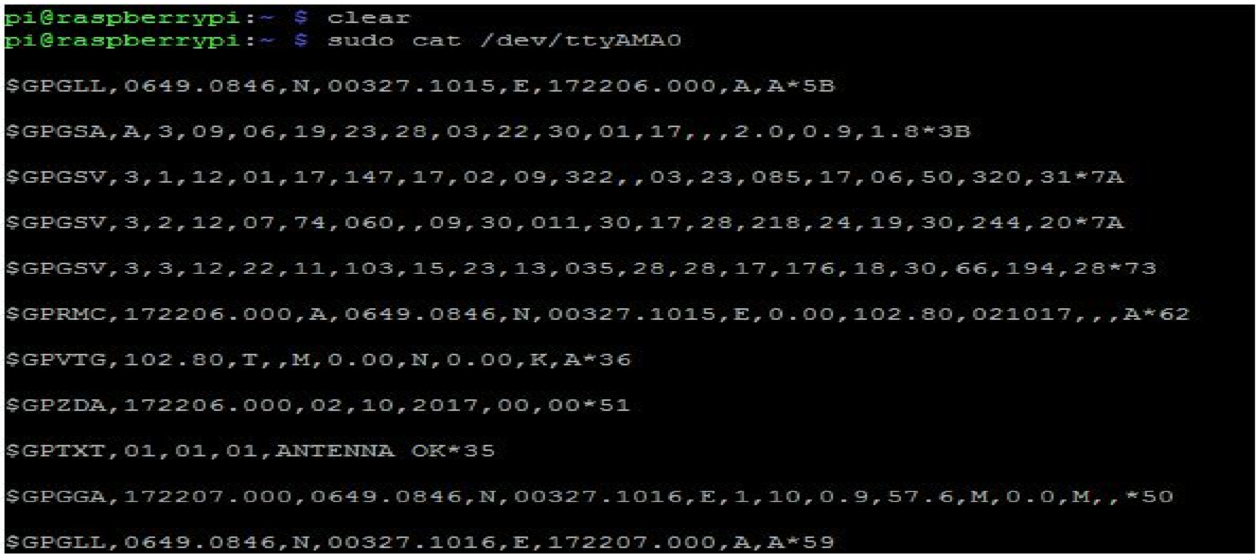
https://www.google.com/maps/dir/SOM+BOQS,+Rawalpindi,+Pakistan/'/33.5888128,73.0588863/'/33.5866722,73.0623731/'/Military+College+of+Signals,+Khadi...
SOM BOQS, Rawalpindi 46000, Pakistan
33.5892531, 73.0568112
Grand Trunk Rd, Rawalpindi 46000, Pak
33.5867269, 73.0625307
Grand Trunk Rd, Rawalpindi 46000, Pak
33.5794723, 73.0645417
Military College of Signals, Khadim Hu
Add destination
OPTIONS
Send directions to your phone
via unnamed roads 25 min 1.9 km
This route has restricted usage or private roads.
DETAILS
33.5892531 , 73.0568112
33.5794723 , 73.0645417
33.5867269 , 73.0625307
33.5794723 , 73.0645417
33.5794723 , 73.0645417
33.5794723 , 73.0645417
SOM BOQS
Grand Trunk Road
Grand Trunk Road
25 min 1.9 km
Military College of Signals
SADDAR
CIVIL LINES
POLICE LINES
JHANDA CHICHI
LALKURTI
SHAHEED OFFICERS COLONY
DHERI HASAN ABAD
Army Museum
Combined Military Hospital
Rawalpindi Golf Club

8

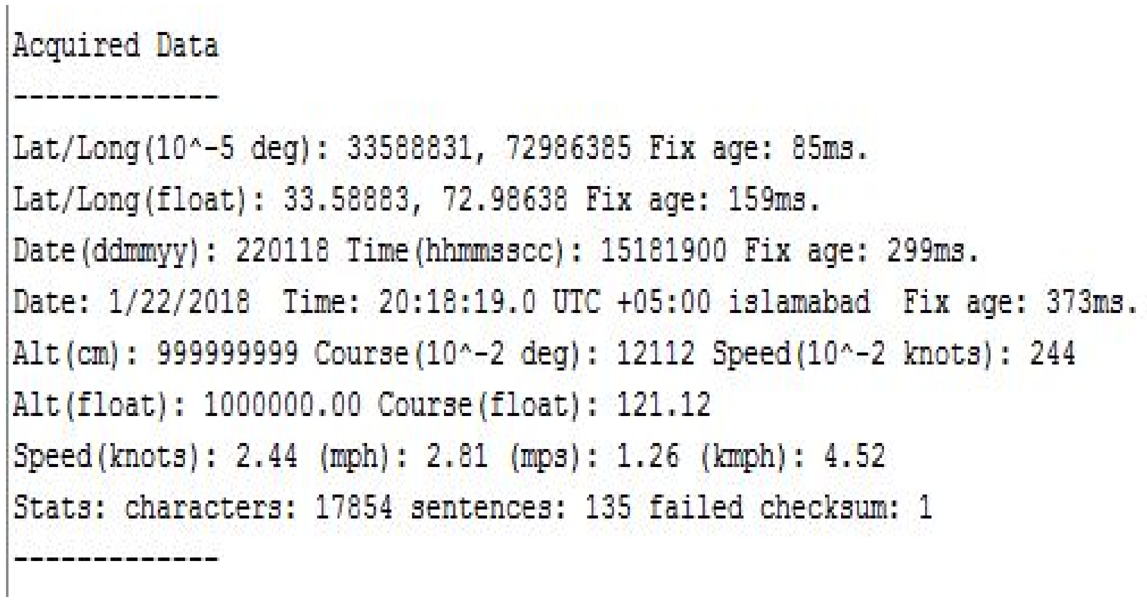
4.2 Finding the Current Location (Through GPS):

4.2.1 <u>Basic Principle of GPS</u>

- GPS provides statistics in a Specific Format consisting of Multiple Sentences. Every Sentence starts from $GPGGA and contains the coordinates, time and different helpful info.
- GPGGA is remarked as Globally Posn Sys Fix Data. We may extract coordinate from $GPGGA string by investigating the commas within the string. Suppose you get GPGGA string and save it in an array, then Latitude may be found after 2 commas and line of Long will be found after four commas

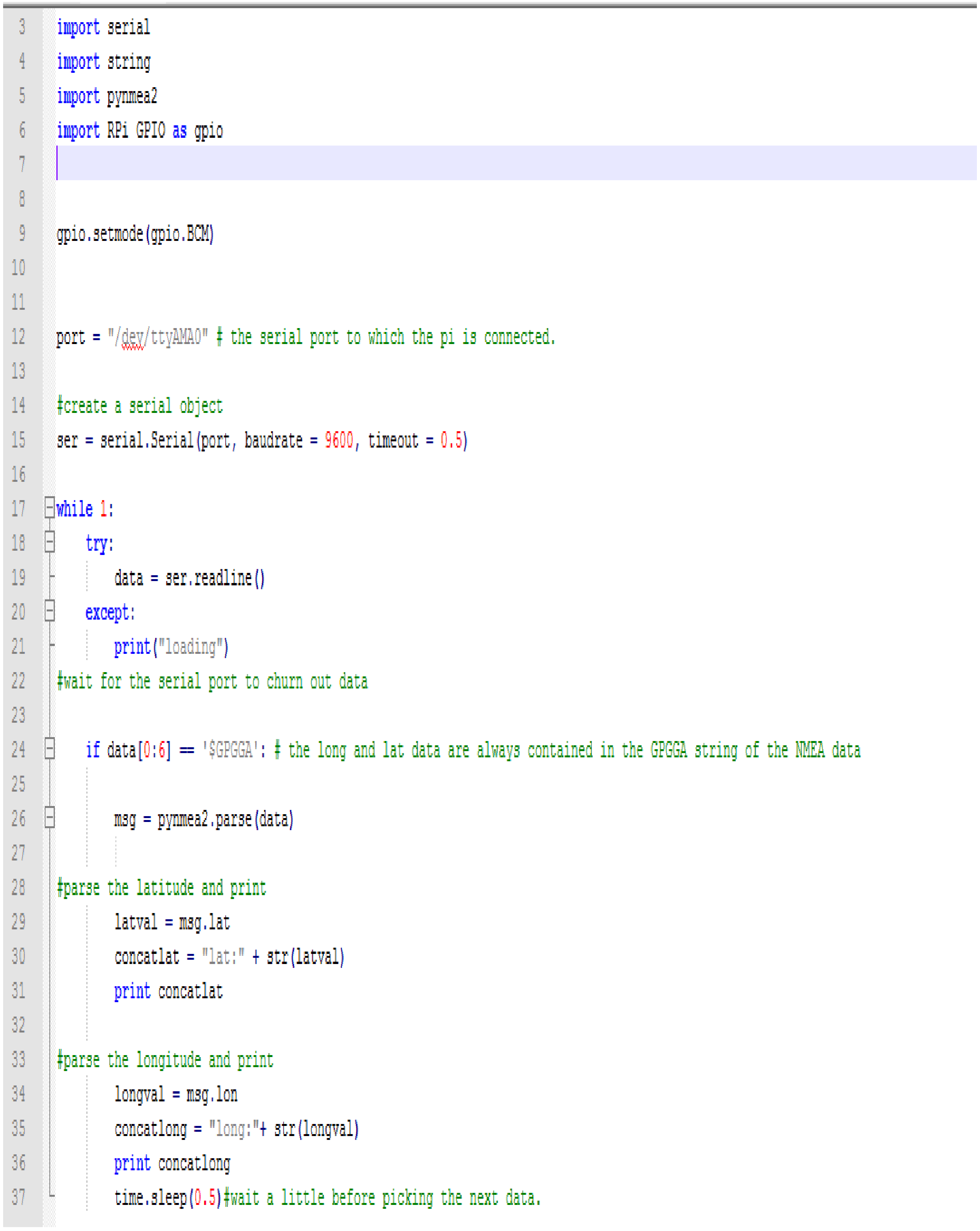


```
pi@raspberrypi:~ $ clear
pi@raspberrypi:~ $ sudo cat /dev/ttyAMA0

$GPGLL,0649.0846,N,00327.1015,E,172206.000,A,A*5B

$GPGSA,A,3,09,06,19,23,28,03,22,30,01,17,,,2.0,0.9,1.8*3B

$GPGSV,3,1,12,01,17,147,17,02,09,322,,03,23,085,17,06,50,320,31*7A

$GPGSV,3,2,12,07,74,060,,09,30,011,30,17,28,218,24,19,30,244,20*7A

$GPGSV,3,3,12,22,11,103,15,23,13,035,28,28,17,176,18,30,66,194,28*73

$GPRMC,172206.000,A,0649.0846,N,00327.1015,E,0.00,102.80,021017,,,A*62

$GPVTG,102.80,T,,M,0.00,N,0.00,K,A*36

$GPZDA,172206.000,02,10,2017,00,00*51

$GPTXT,01,01,01,ANTENNA OK*35

$GPGGA,172207.000,0649.0846,N,00327.1016,E,1,10,0.9,57.6,M,0.0,M,,*50

$GPGLL,0649.0846,N,00327.1016,E,172207.000,A,A*59
```



```
Acquired Data
-------------
Lat/Long(10^-5 deg): 33588831, 72986385 Fix age: 85ms.
Lat/Long(float): 33.58883, 72.98638 Fix age: 159ms.
Date(ddmmyy): 220118 Time(hhmmsscc): 15181900 Fix age: 299ms.
Date: 1/22/2018  Time: 20:18:19.0 UTC +05:00 islamabad  Fix age: 373ms.
Alt(cm): 999999999 Course(10^-2 deg): 12112 Speed(10^-2 knots): 244
Alt(float): 1000000.00 Course(float): 121.12
Speed(knots): 2.44 (mph): 2.81 (mps): 1.26 (kmph): 4.52
Stats: characters: 17854 sentences: 135 failed checksum: 1
-------------
```

4.2.2 Code:

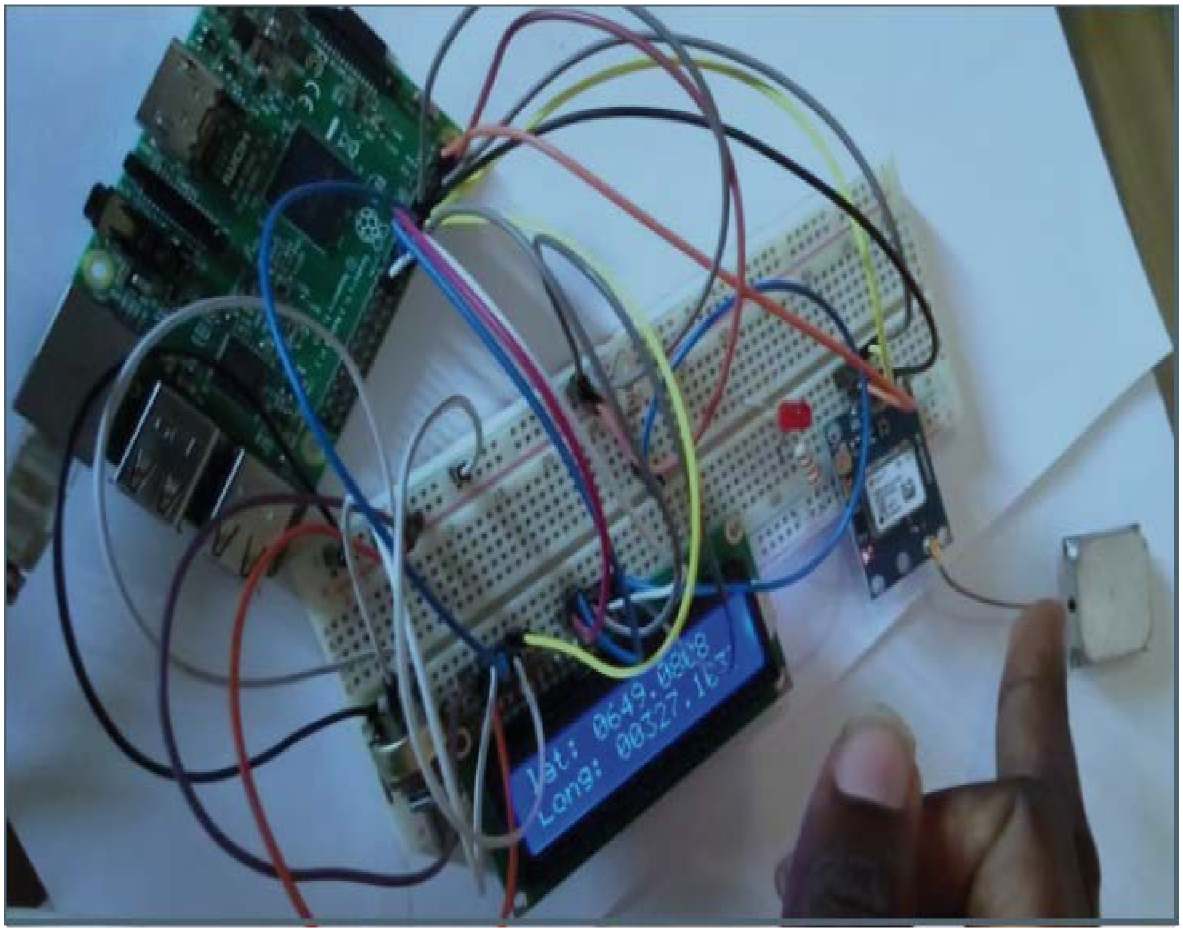


```python
import serial
import string
import pynmea2
import RPi GPIO as gpio


gpio.setmode(gpio.BCM)


port = "/dev/ttyAMA0" # the serial port to which the pi is connected.

#create a serial object
ser = serial.Serial(port, baudrate = 9600, timeout = 0.5)

while 1:
    try:
        data = ser.readline()
    except:
        print("loading")
#wait for the serial port to churn out data

    if data[0:6] == '$GPGGA': # the long and lat data are always contained in the GPGGA string of the NMEA data

        msg = pynmea2.parse(data)

#parse the latitude and print
        latval = msg.lat
        concatlat = "lat:" + str(latval)
        print concatlat

#parse the longitude and print
        longval = msg.lon
        concatlong = "long:"+ str(longval)
        print concatlong
        time.sleep(0.5)#wait a little before picking the next data.
```

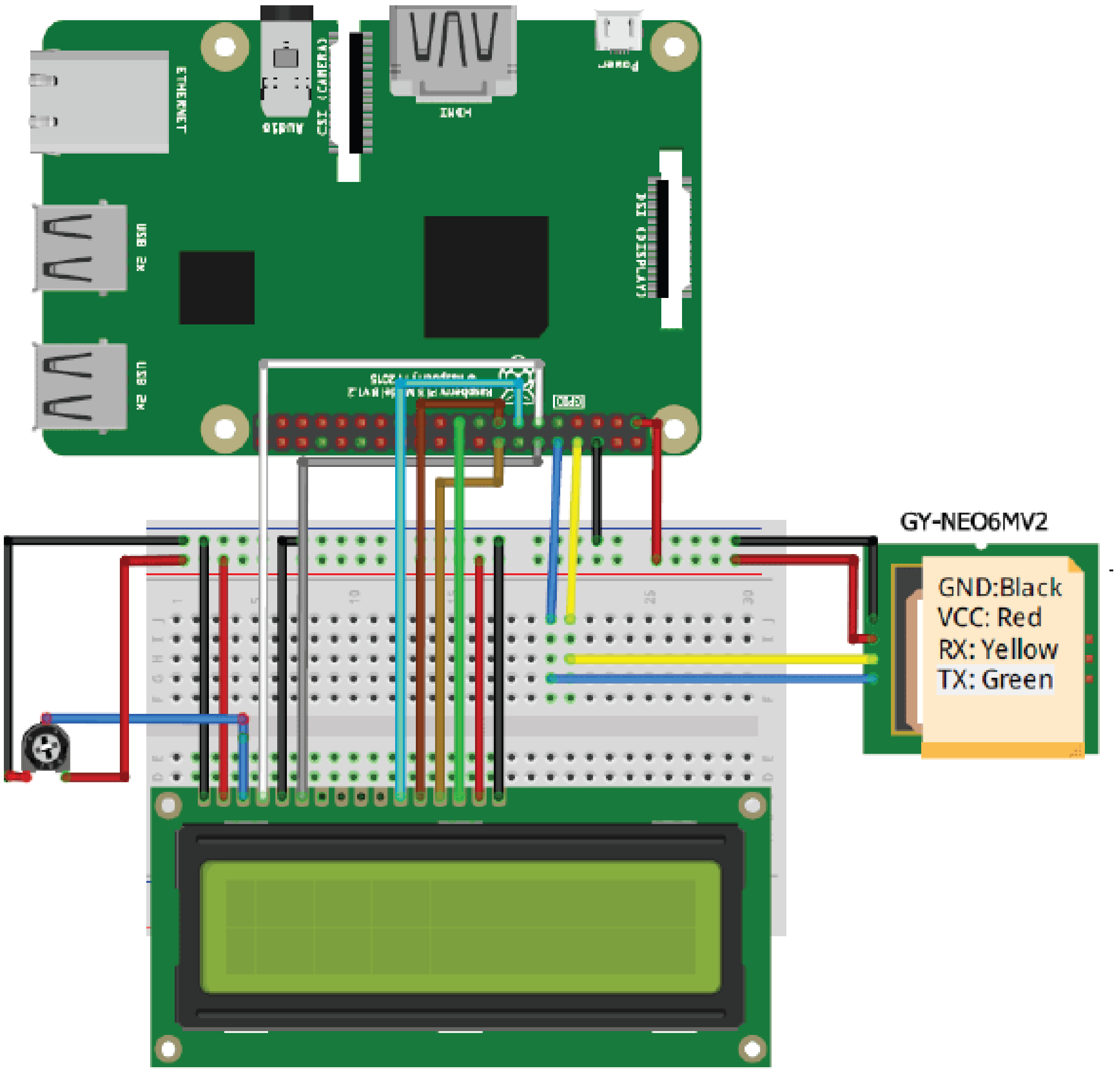
4.2.3 <u>Interfacing with Respberry Pi</u>

4.2.3.1 Preparing the Respberry Pi to Communicate with GPS

- Updating the Respberry Pi.
- Setting the UART in Respberry Pi.
- Disabling Respberry Pi Getty Service.
- Activating TTV AMAO.
- Installing mini COM and pynmea 2.
- Installing the LCD library



4.2.3.2 Connection for Respberry Pi GPS Module Interfacing



This product has huge applications in the health sector. Currently there is no efficient product present in the market that can assist the blind people and can help them perform everyday tasks.

- It is sad to know that even in this day and age we have not done enough to solve the problems that are faced by the blind people.

- The introduction of this product in the health sector can help compensate the visually impaired people. This will boost their confidence and enable them to move around without the assistance of a helper.

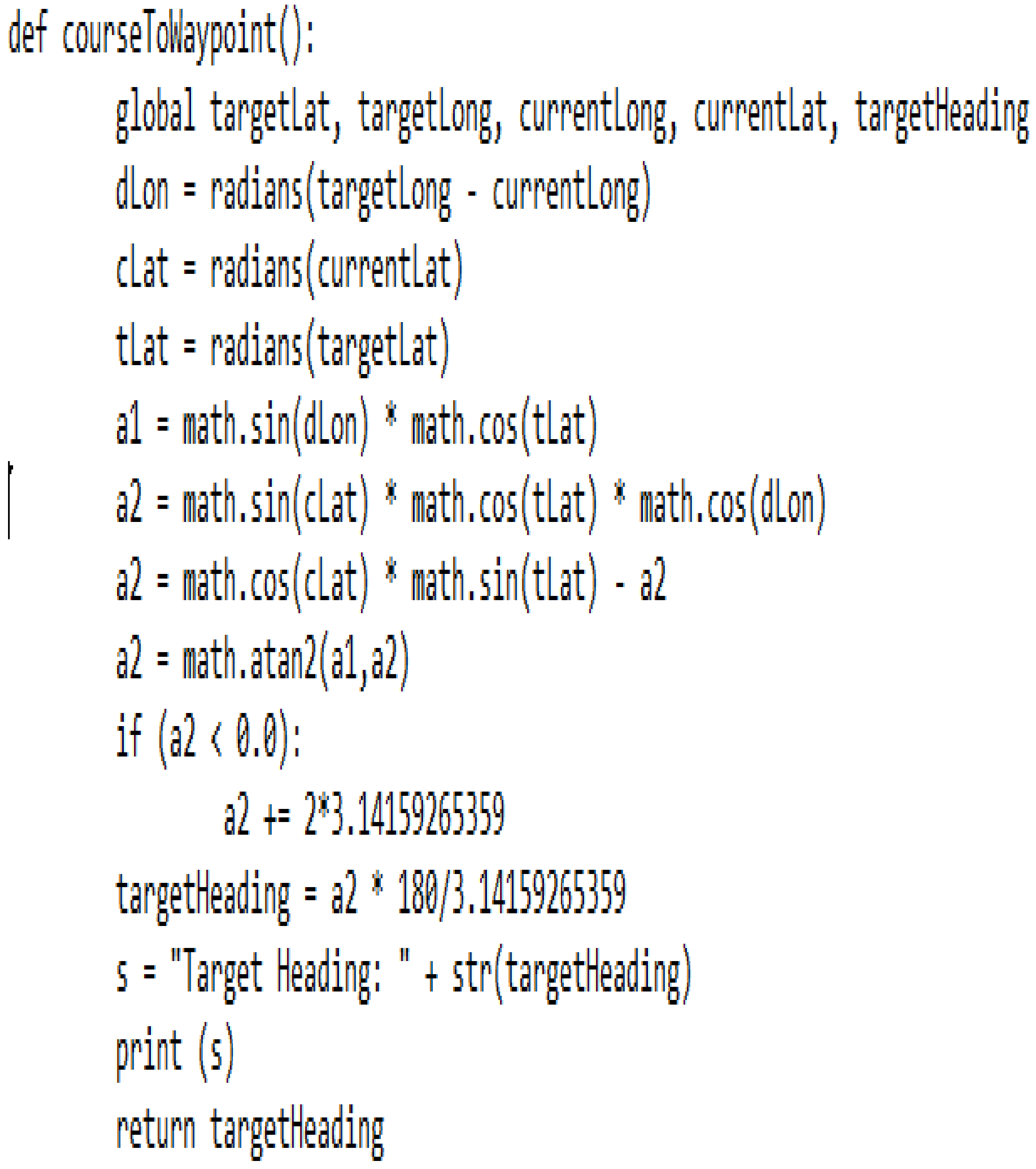
4.3 CALCULATING TARGET BEARING:

- Target Bearing is found using Coordinates of Current Location(GPS) and 1st Way Point
- Formula Applied:

Formula: $$\theta = \operatorname{atan2}(\sin\Delta\lambda \cdot \cos\varphi_2 , \cos\varphi_1 \cdot \sin\varphi_2 - \sin\varphi_1 \cdot \cos\varphi_2 \cdot \cos\Delta\lambda)$$

where φ_1,λ_1 is the start point, φ_2,λ_2 the end point ($\Delta\lambda$ is the difference in longitude)

➢ Application of Formula in Code:

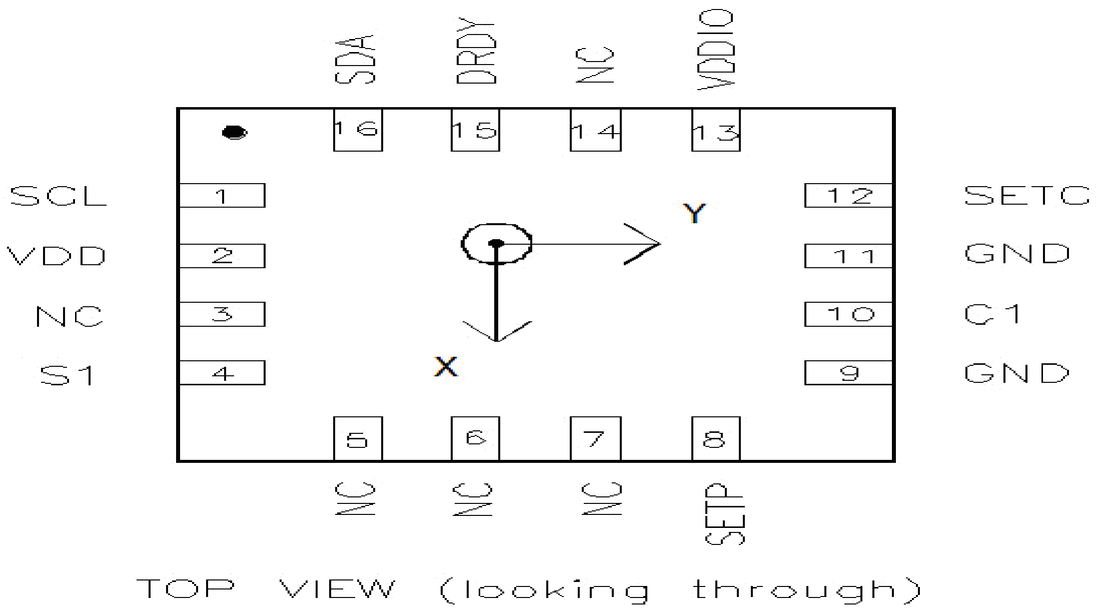


```
def courseToWaypoint():
    global targetLat, targetLong, currentLong, currentLat, targetHeading
    dLon = radians(targetLong - currentLong)
    cLat = radians(currentLat)
    tLat = radians(targetLat)
    a1 = math.sin(dLon) * math.cos(tLat)
    a2 = math.sin(cLat) * math.cos(tLat) * math.cos(dLon)
    a2 = math.cos(cLat) * math.sin(tLat) - a2
    a2 = math.atan2(a1,a2)
    if (a2 < 0.0):
        a2 += 2*3.14159265359
    targetHeading = a2 * 180/3.14159265359
    s = "Target Heading: " + str(targetHeading)
    print (s)
    return targetHeading
```

4.4 FINDING CURRENT BEARING (Magnetometer)

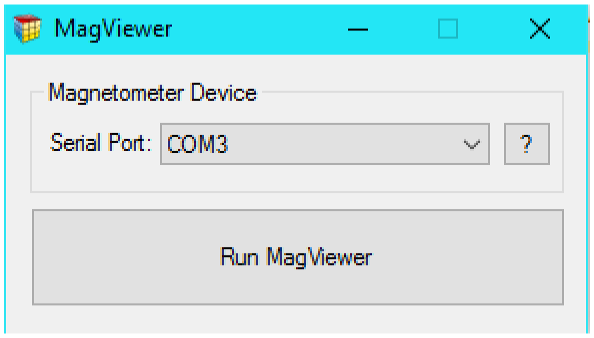
4.4.1 <u>Working Principle</u>

- The Compass Module 3-Axis HMC5883L be a low-field magnetic sensing device with a digital interface.
- The compass module converts any field to a differential voltage output on three axes.
- This voltage shift is that the raw digital output worth, which might then be used to calculate headings or sense magnetic fields coming back from completely different directions

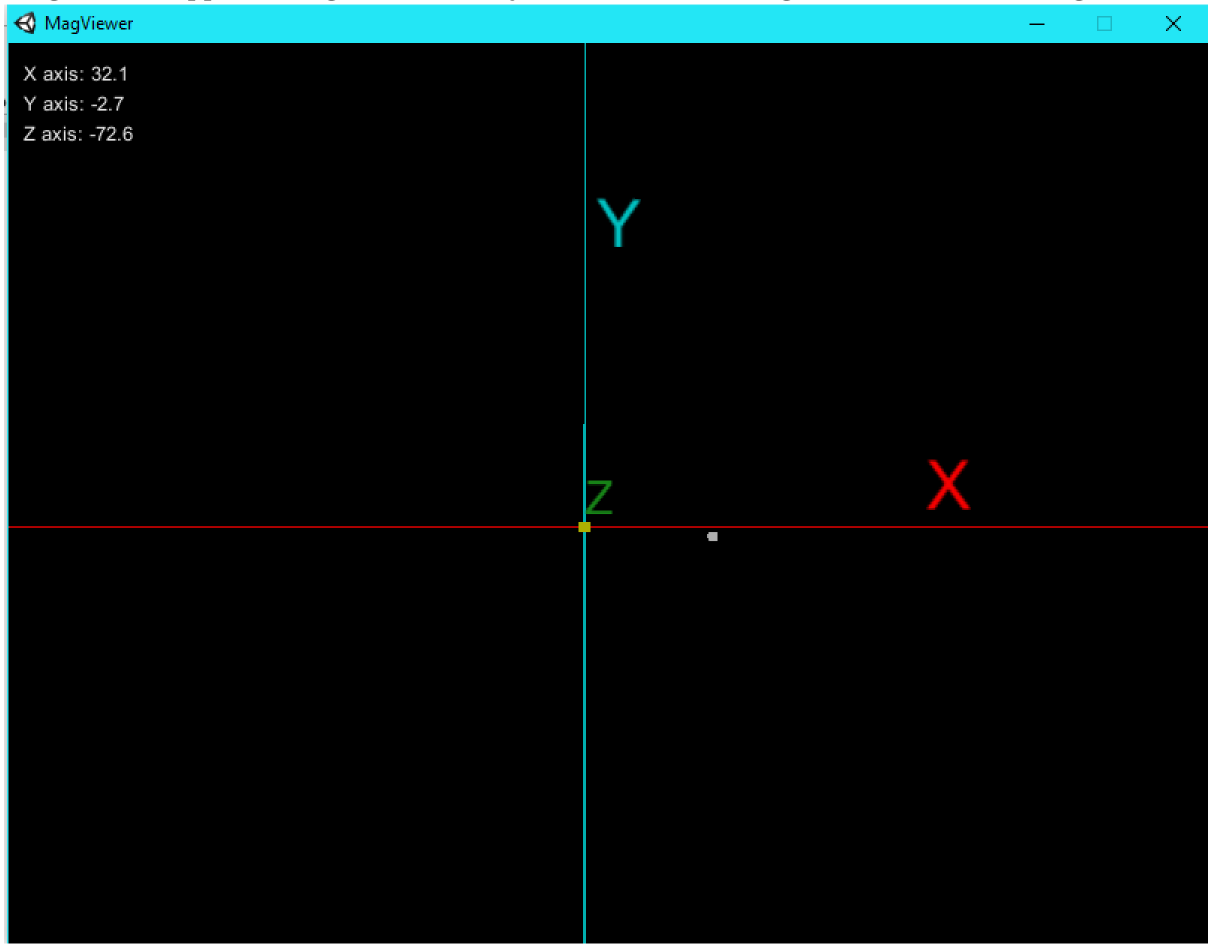


4.4.2 Calibration Process

1) Connect Magnetometer with Arduino according to configuration:
2) Open Magsensor3 Arduino code and upload it in Arduino.
3) Now open MagViewer.exe Application. Select COM Port on which Arduino is connected. Then press run.

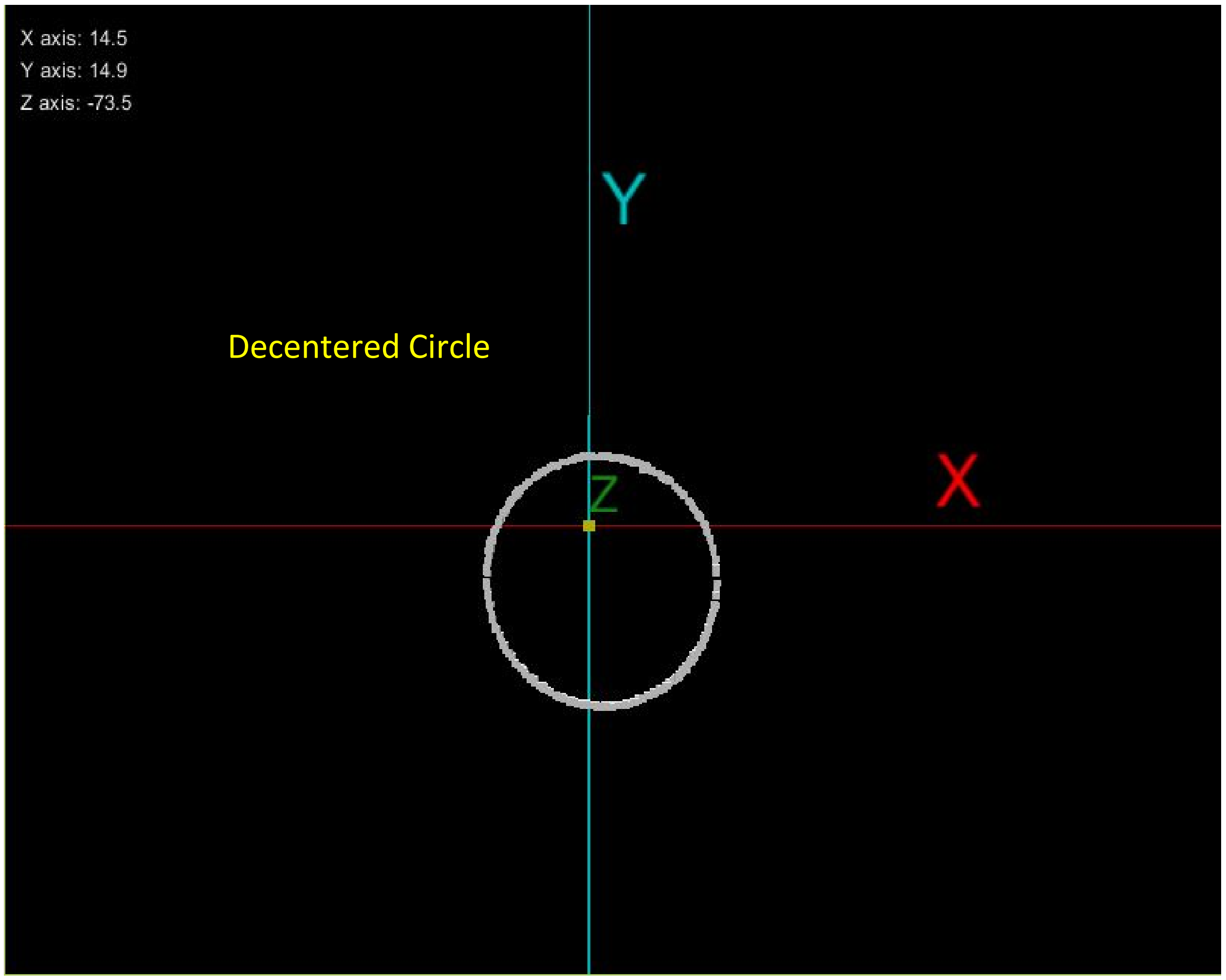


4) MagViewer Application get started. Adjust the window along Z-axis as shown in figure.



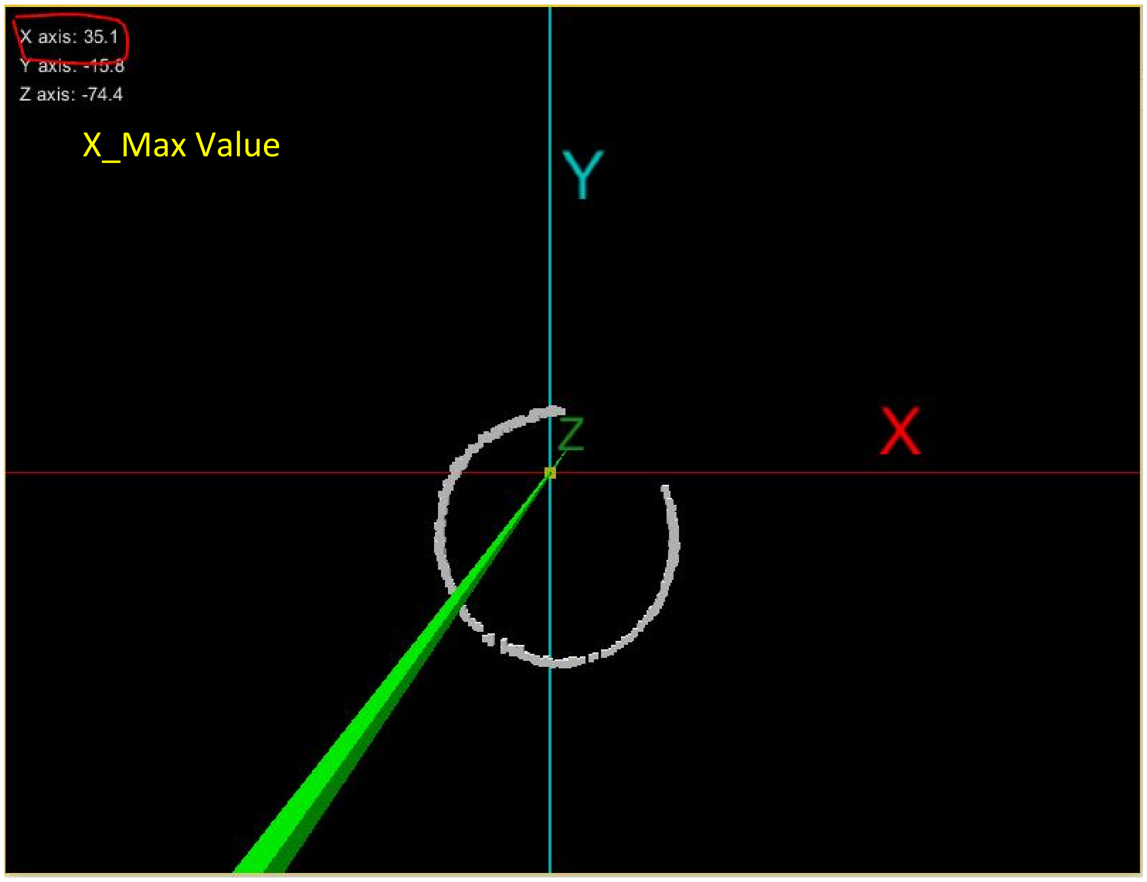
X axis: 32.1
Y axis: -2.7
Z axis: -72.6

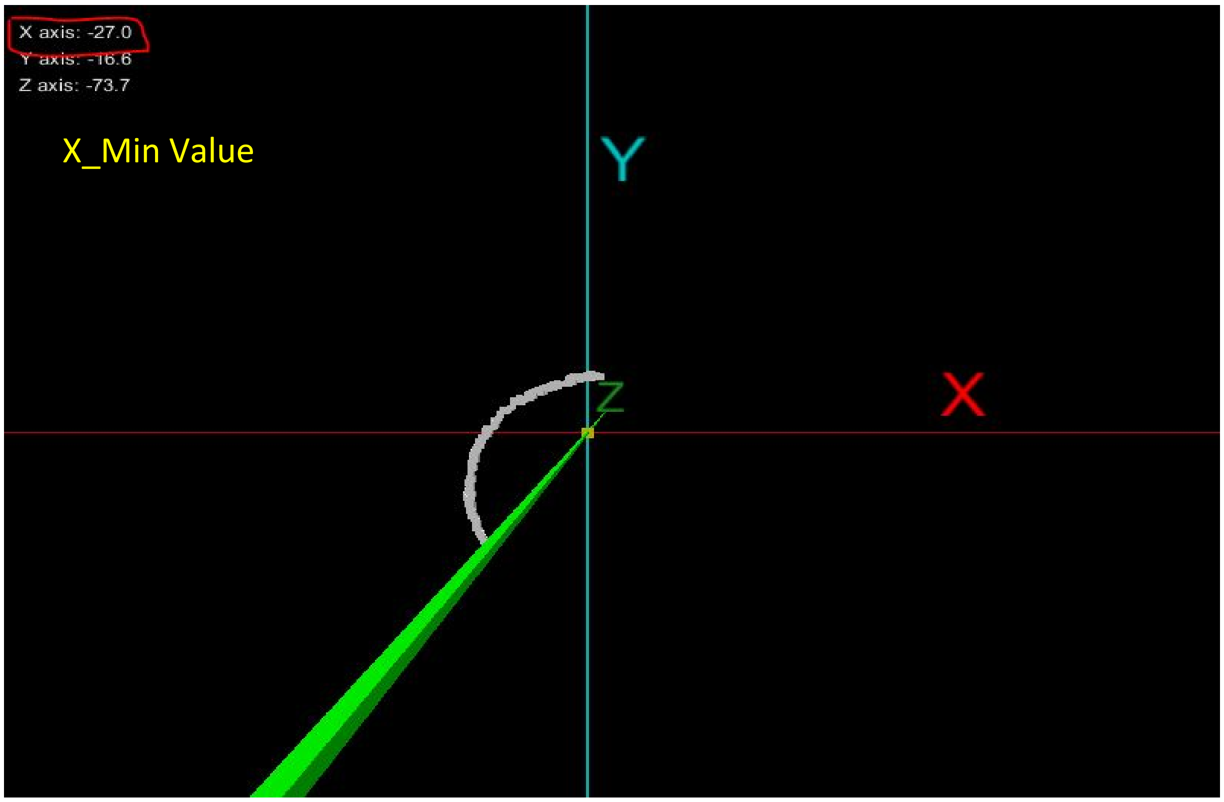
5) Now rotate Magnetometer along Z-axis to get the circle.


X axis: 14.5
Y axis: 14.9
Z axis: -73.5

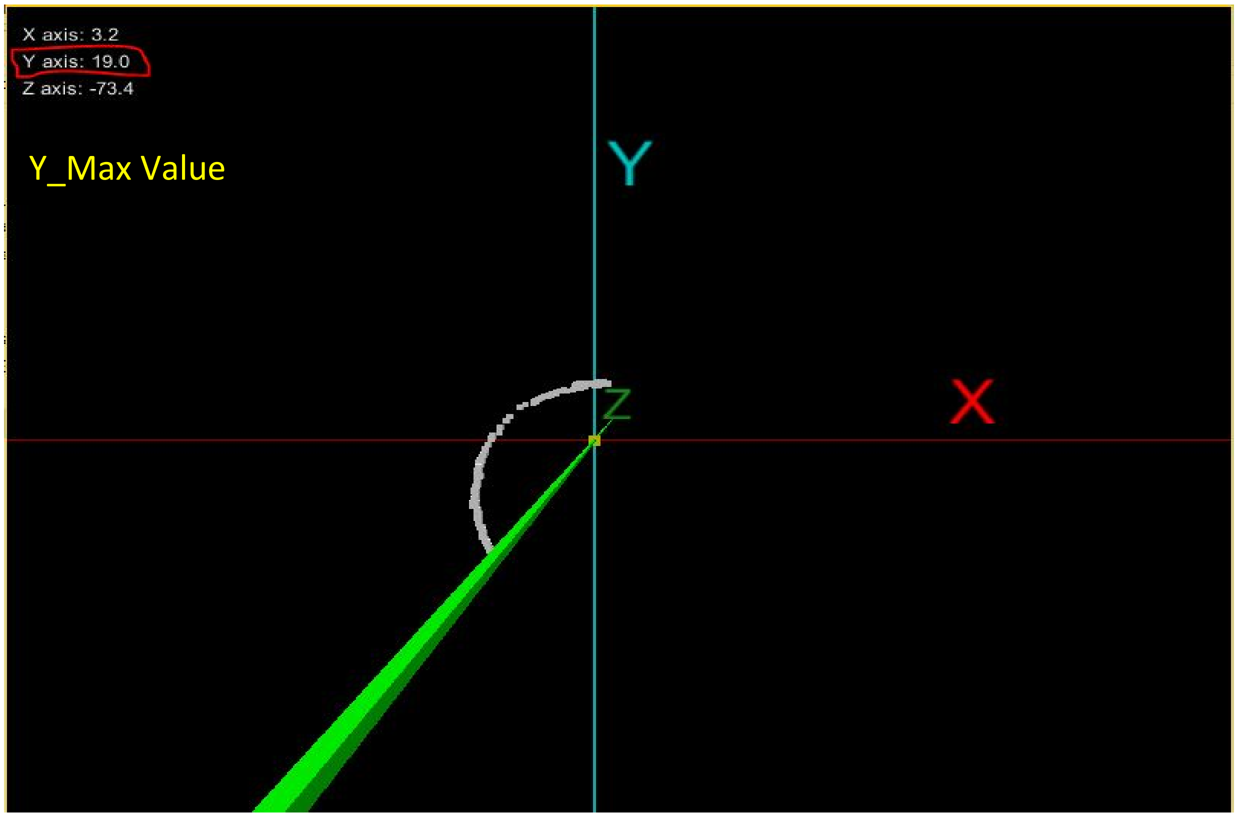
6) It can be seen the circle of plotted values is decentered. So it is required to calibrate it.

7) For calibration, get the values of X and Y axis along Z axis. Note down the max and min values of X and Y axis. In our case we got following values.

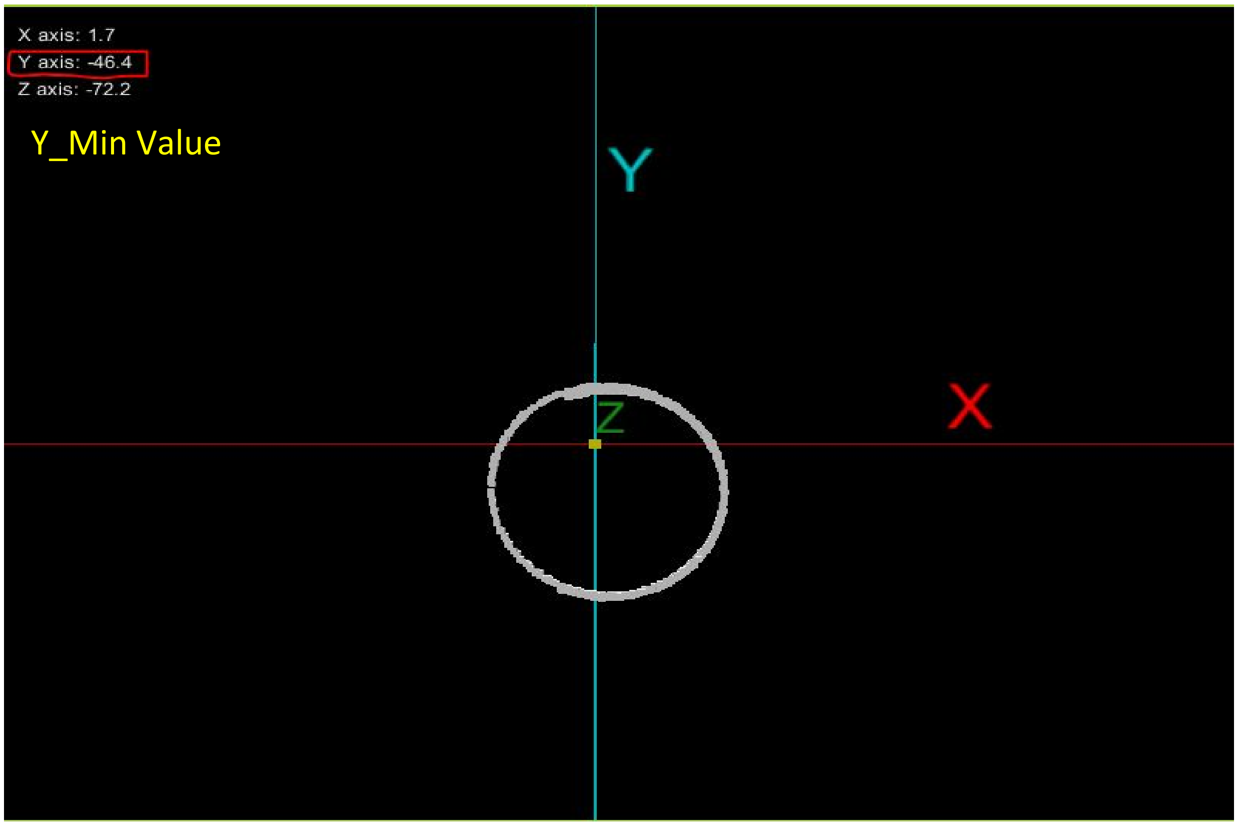

```
X axis: 35.1
Y axis: -15.8
Z axis: -74.4
```


X axis: -27.0
Y axis: -16.6
Z axis: -73.7

X_Min Value


X axis: 3.2
Y axis: 19.0
Z axis: -73.4

Y_Max Value


X axis: 1.7
Y axis: -46.4
Z axis: -72.2

8) Now we need to find offset values of X and Y axis to center the circle along Z axis.

9) The basic principle is that center of X and Y axis should be at 0.

10) Current center point of X axis is:

    (X_Max + X_Min)/2 = (35.1 - 27)/2 = 8.1/2 = 4.05

11) So X_offset = -4.05 (To make X center at zero)

12) Similarly, Current center point of Y axis is:

    (Y_Max + Y_Min)/2 = (19 – 46.4)/2 = -27.4/2 = -13.7

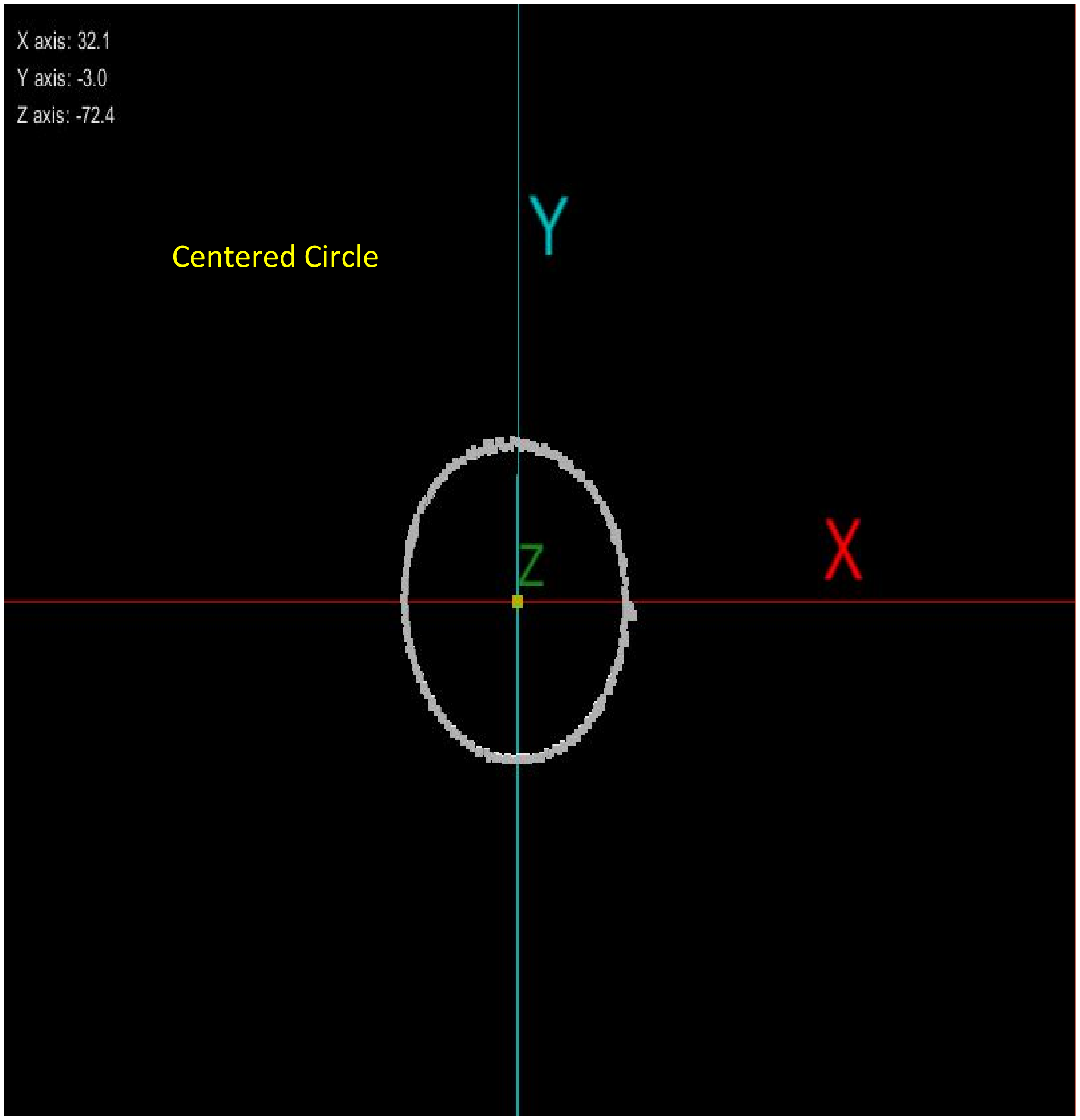
13) So Y_offset = +13.7 (To make Y center at zero)

14) Now put these offset values in Magsensor_Calibrated Arduino Program.

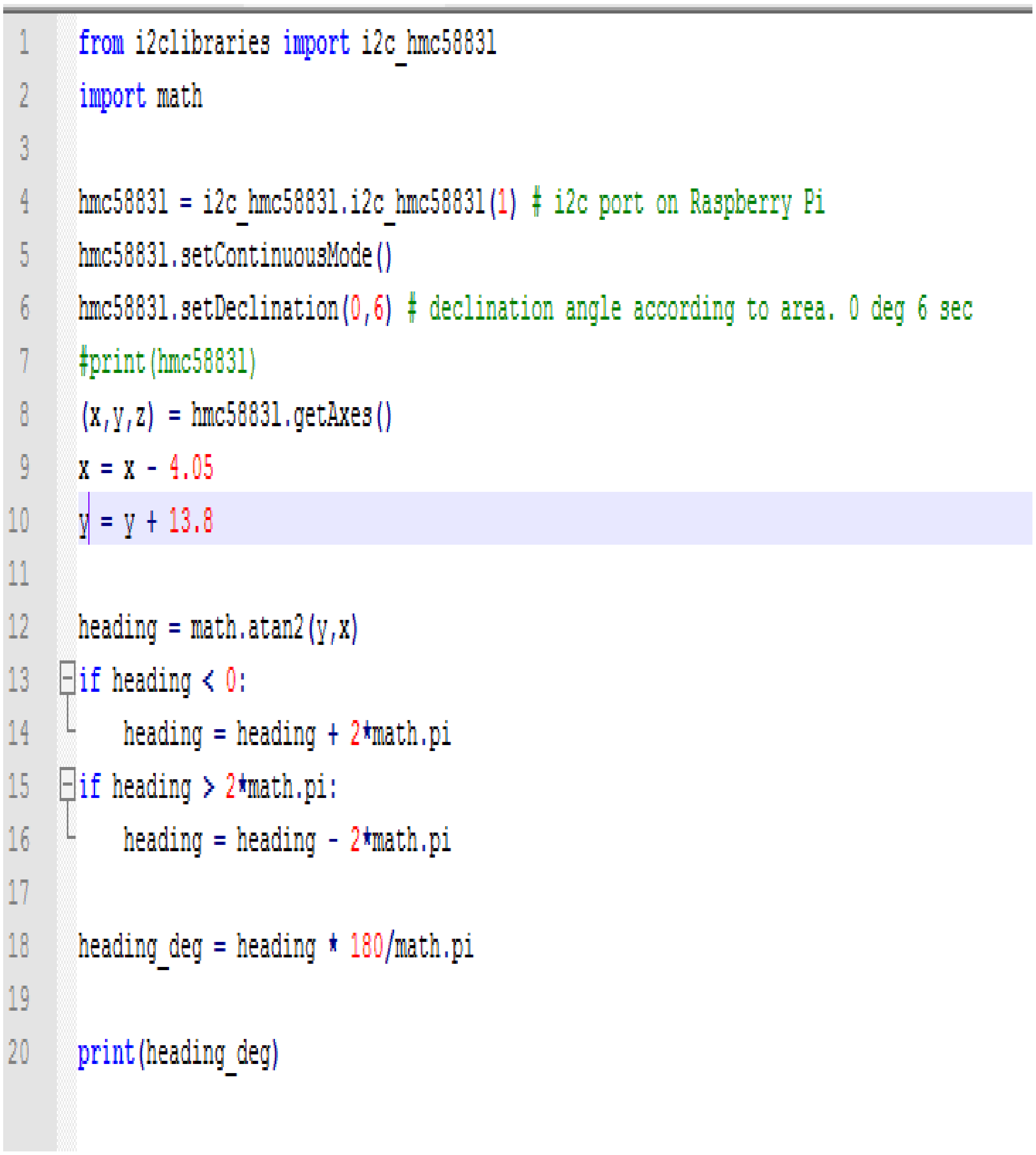
15) Upload that program in Arduino and start MagViewer again.

16) Now again rotate the sensor along Z axis to get circle.

17) It can be seen that now circle is at center.


X axis: 32.1
Y axis: -3.0
Z axis: -72.4
Centered Circle
Y
Z
X

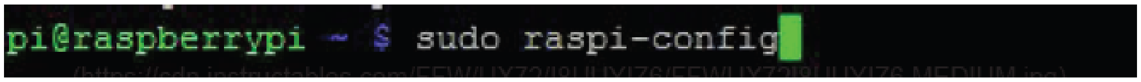
4.4.3 Code:



```python
from i2clibraries import i2c_hmc5883l
import math

hmc5883l = i2c_hmc5883l.i2c_hmc5883l(1) # i2c port on Raspberry Pi
hmc5883l.setContinuousMode()
hmc5883l.setDeclination(0,6) # declination angle according to area. 0 deg 6 sec
#print(hmc5883l)
(x,y,z) = hmc5883l.getAxes()
x = x - 4.05
y = y + 13.8

heading = math.atan2(y,x)
if heading < 0:
    heading = heading + 2*math.pi
if heading > 2*math.pi:
    heading = heading - 2*math.pi

heading_deg = heading * 180/math.pi

print(heading_deg)
```

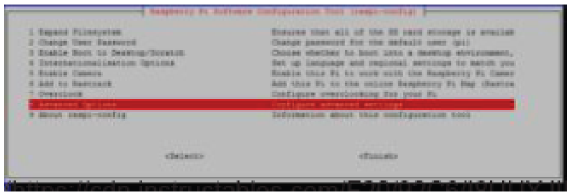
4.4.4 Interfacing with Respberry Pi

Step 1: Configuring RPi2

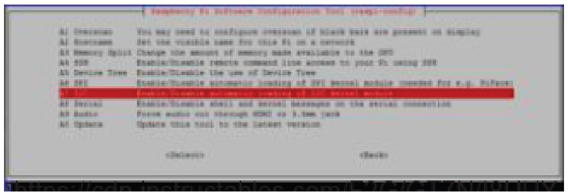


```
pi@raspberrypi ~ $ sudo raspi-config
```

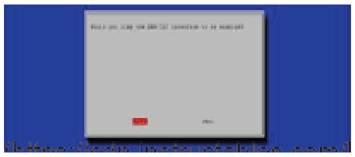
(https://cdn.instructables.com/FEW/UX72/I8UUXIZ6/FEWUX72I8UUXIZ6.MEDIUM.jpg)



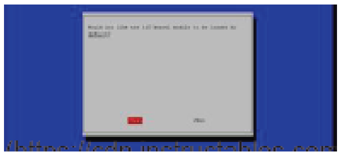
```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem              Ensures that all of the SD card storage is availab
2 Change User Password           Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment,
4 Internationalisation Options   Set up language and regional settings to match you
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camer
6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastra
7 Overclock                      Configure overclocking for your Pi
8 Advanced Options               Configure advanced settings
9 About raspi-config             Information about this configuration tool

              <Select>                      <Finish>
```



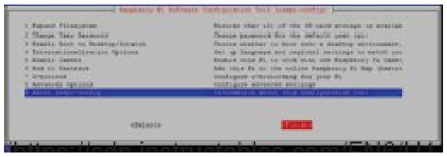
```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan     You may need to configure overscan if black bars are present on display
A2 Hostname     Set the visible name for this Pi on a network
A3 Memory Split Change the amount of memory made available to the GPU
A4 SSH          Enable/Disable remote command line access to your Pi using SSH
A5 Device Tree  Enable/Disable the use of Device Tree
A6 SPI          Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C          Enable/Disable automatic loading of I2C kernel module
A8 Serial       Enable/Disable shell and kernel messages on the serial connection
A9 Audio        Force audio out through HDMI or 3.5mm jack
A0 Update       Update this tool to the latest version

              <Select>                      <Back>
```



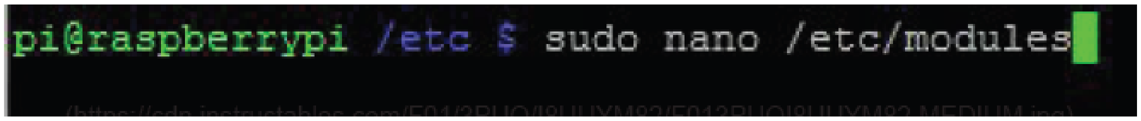
```
Would you like the ARM I2C interface to be enabled?

     <Yes>          <No>
```



```
Would you like the I2C kernel module to be loaded by
default?

     <Yes>          <No>
```

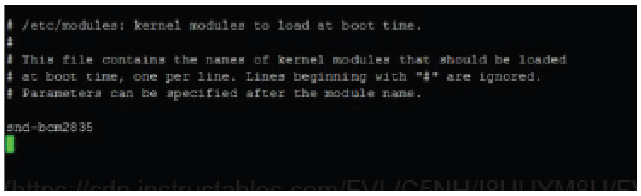


```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem              Ensures that all of the SD card storage is availab
2 Change User Password           Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment,
4 Internationalisation Options   Set up language and regional settings to match you
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camer
6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastra
7 Overclock                      Configure overclocking for your Pi
8 Advanced Options               Configure advanced settings
9 About raspi-config             Information about this configuration tool

              <Select>                      <Finish>
```

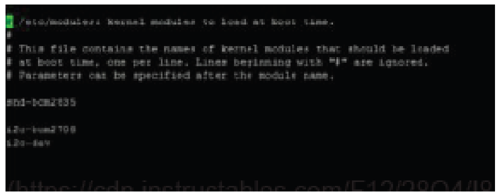


```
pi@raspberrypi /etc $ sudo nano /etc/modules
```

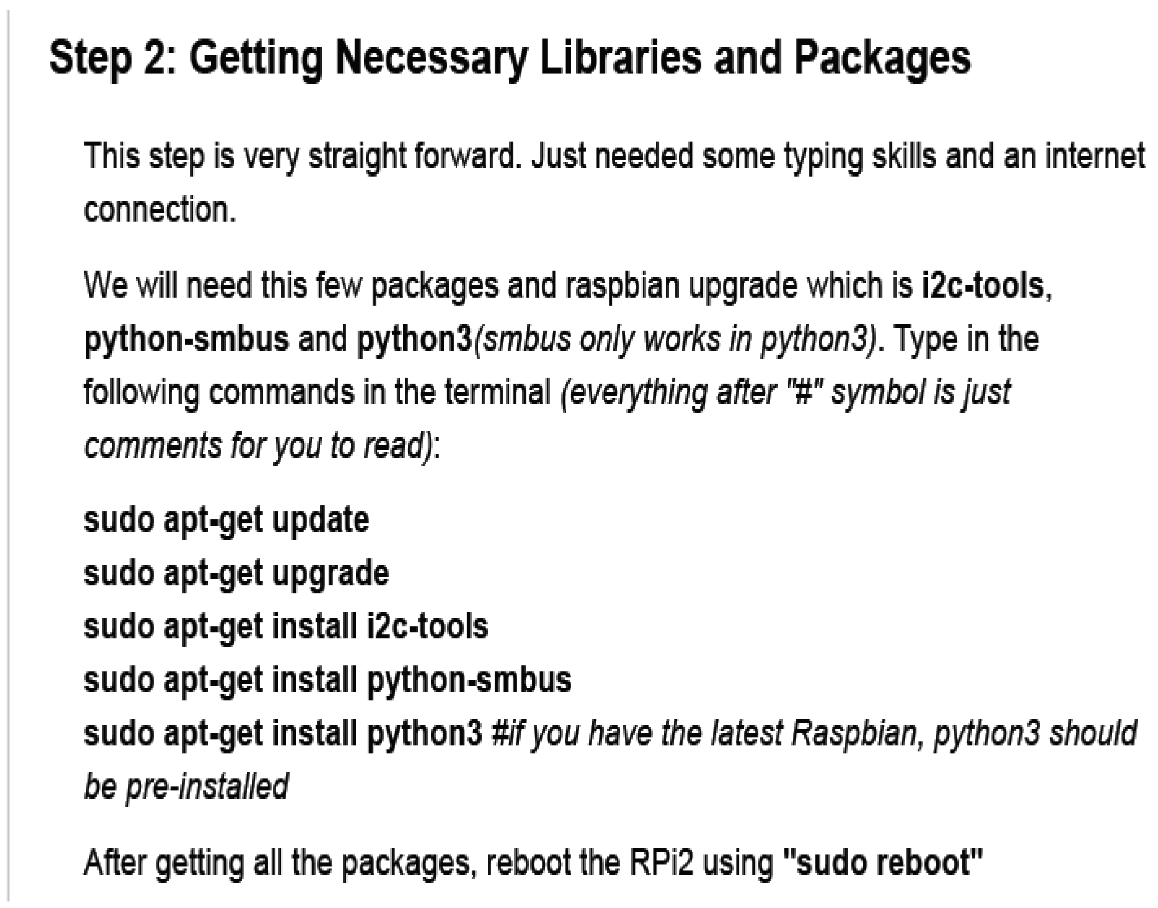
(https://cdn.instructables.com/F01/2BUQ/I8UUYM82/F012BUQI8UUYM82.MEDIUM.jpg)



```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
```



```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835

i2c-bcm2708
i2c-dev
```



Step 2: Getting Necessary Libraries and Packages

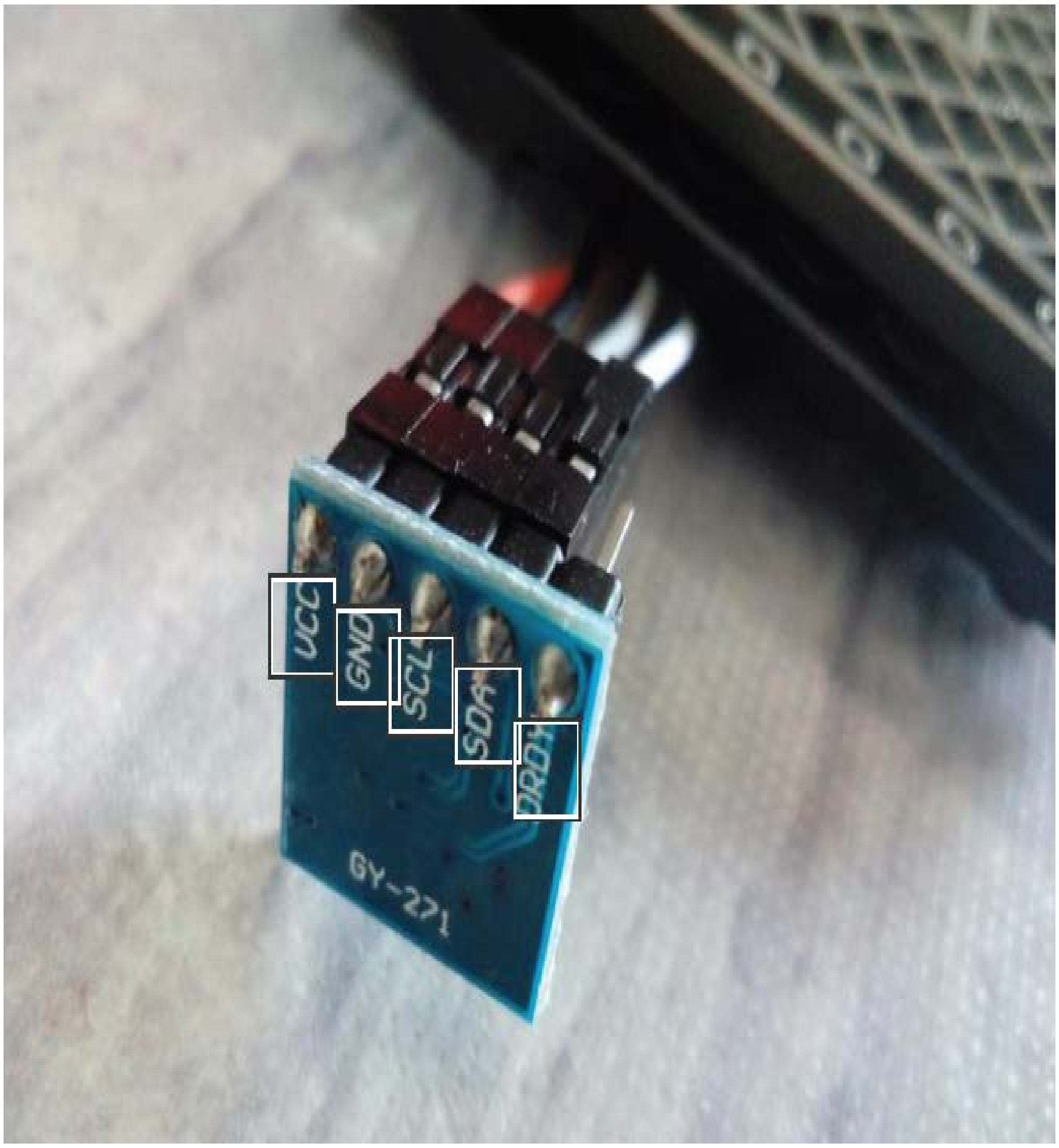
This step is very straight forward. Just needed some typing skills and an internet connection.

We will need this few packages and raspbian upgrade which is **i2c-tools**, **python-smbus** and **python3***(smbus only works in python3)*. Type in the following commands in the terminal *(everything after "#" symbol is just comments for you to read)*:

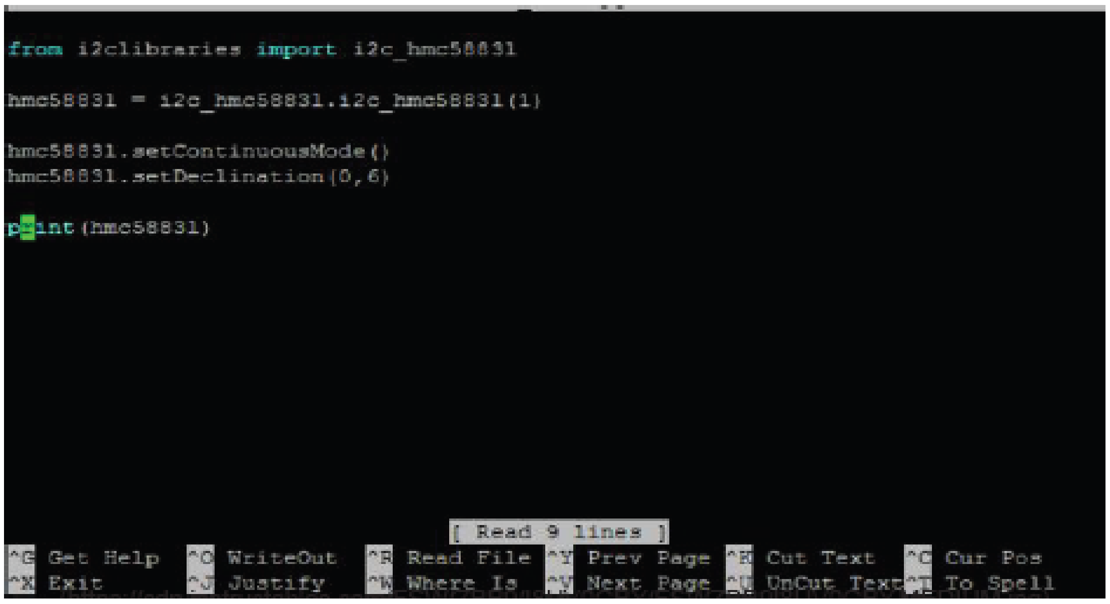
```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install i2c-tools
sudo apt-get install python-smbus
sudo apt-get install python3 #if you have the latest Raspbian, python3 should be pre-installed
```

After getting all the packages, reboot the RPi2 using **"sudo reboot"**

Step 3: Hardware Wiring

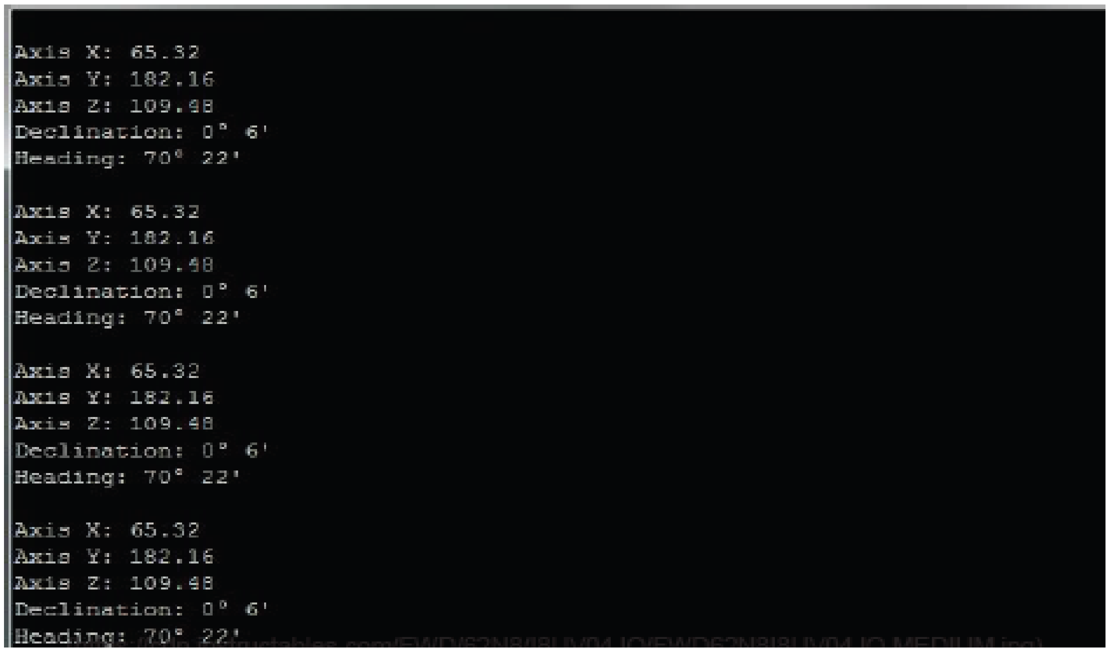


Step 4: Coding



```
from i2clibraries import i2c_hmc58831

hmc58831 = i2c_hmc58831.i2c_hmc58831(1)

hmc58831.setContinuousMode()
hmc58831.setDeclination(0,6)

print(hmc58831)
```

```
                              [ Read 9 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```



```
Axis X: 65.32
Axis Y: 182.16
Axis Z: 109.48
Declination: 0° 6'
Heading: 70° 22'

Axis X: 65.32
Axis Y: 182.16
Axis Z: 109.48
Declination: 0° 6'
Heading: 70° 22'

Axis X: 65.32
Axis Y: 182.16
Axis Z: 109.48
Declination: 0° 6'
Heading: 70° 22'

Axis X: 65.32
Axis Y: 182.16
Axis Z: 109.48
Declination: 0° 6'
Heading: 70° 22'
```

4.5 ESTIMATE HEADING ANGLE (Desired Direction)

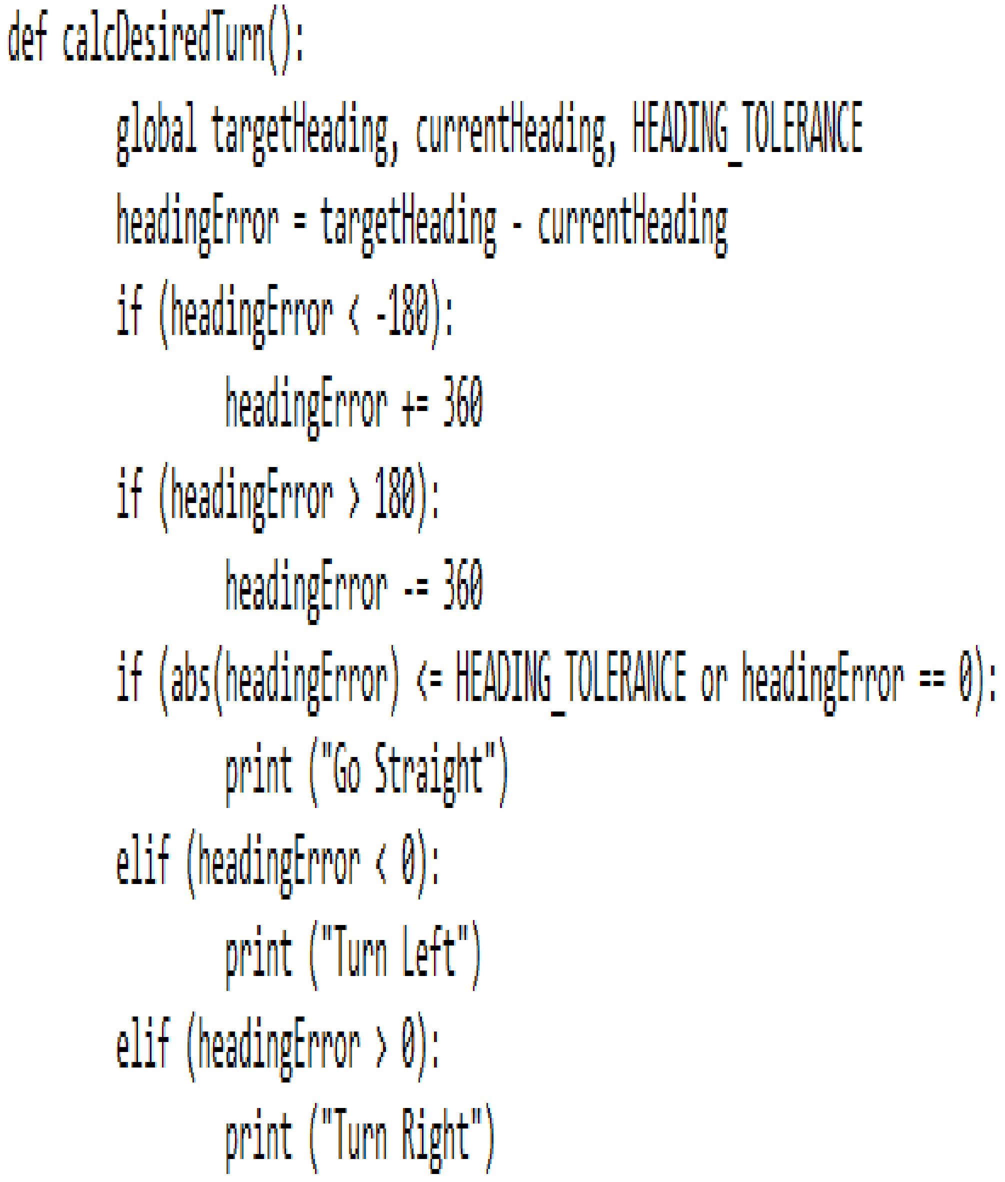
- Direction Difference = target direction – current direction
- <u>Wrap Direction Difference</u>

```
If (direction difference < -180)
Direction difference =+ 360
If (direction difference > 180)
Direction difference = -360
```

- <u>Find desired turn</u>

```
If (direction difference < 0)
Desired turn = LEFT
If (direction difference > 0)
Desired turn = RIGHT
Else
 Desired turn = STRAIGHT
```

- **Code**



```
def calcDesiredTurn():
    global targetHeading, currentHeading, HEADING_TOLERANCE
    headingError = targetHeading - currentHeading
    if (headingError < -180):
        headingError += 360
    if (headingError > 180):
        headingError -= 360
    if (abs(headingError) <= HEADING_TOLERANCE or headingError == 0):
        print ("Go Straight")
    elif (headingError < 0):
        print ("Turn Left")
    elif (headingError > 0):
        print ("Turn Right")
```

4.6 Repetition Until way Point Achieved

- Formula For Distance Between Two Points

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \cdot c$$

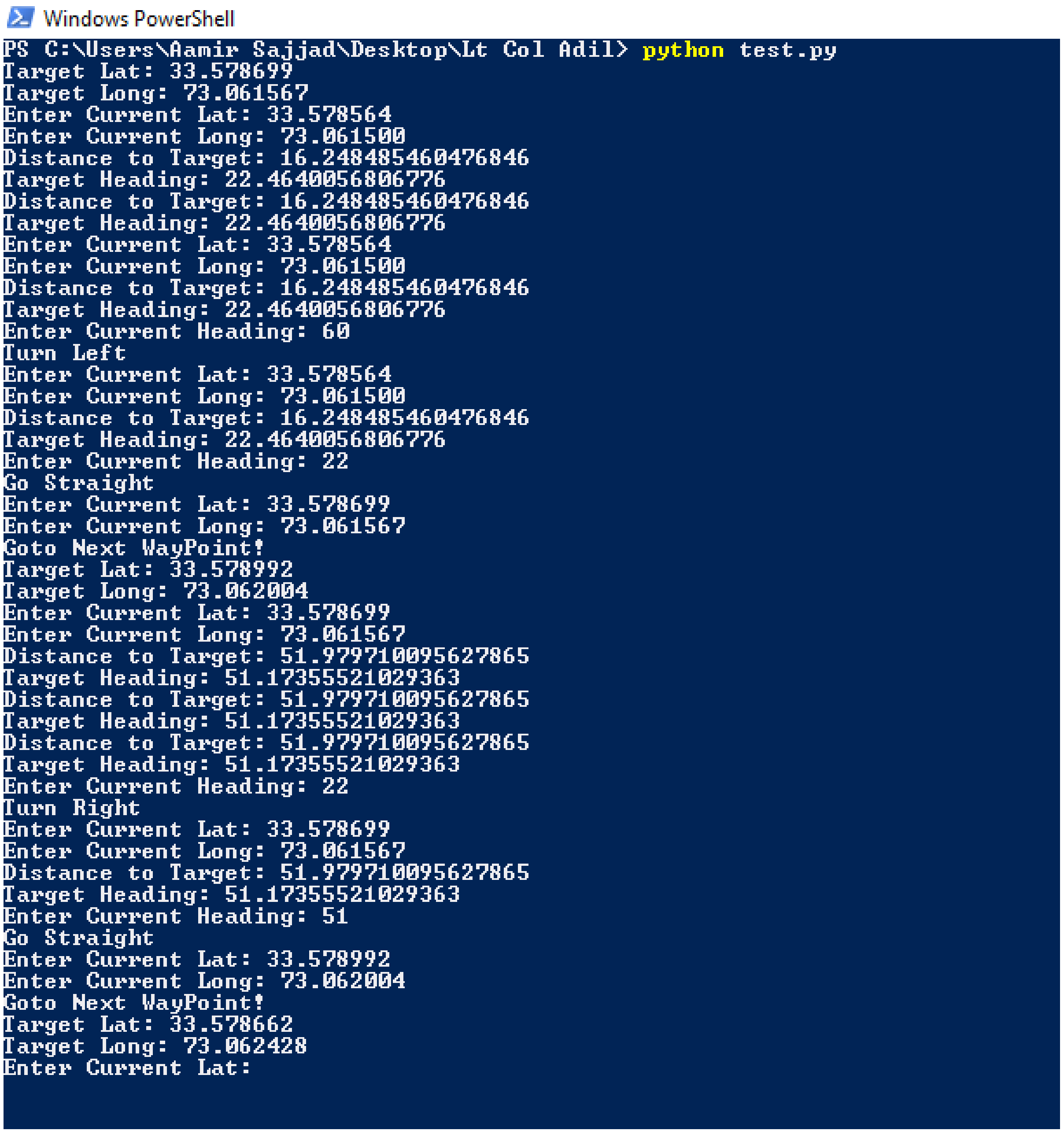
where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km);

note that angles need to be in radians to pass to trig functions!

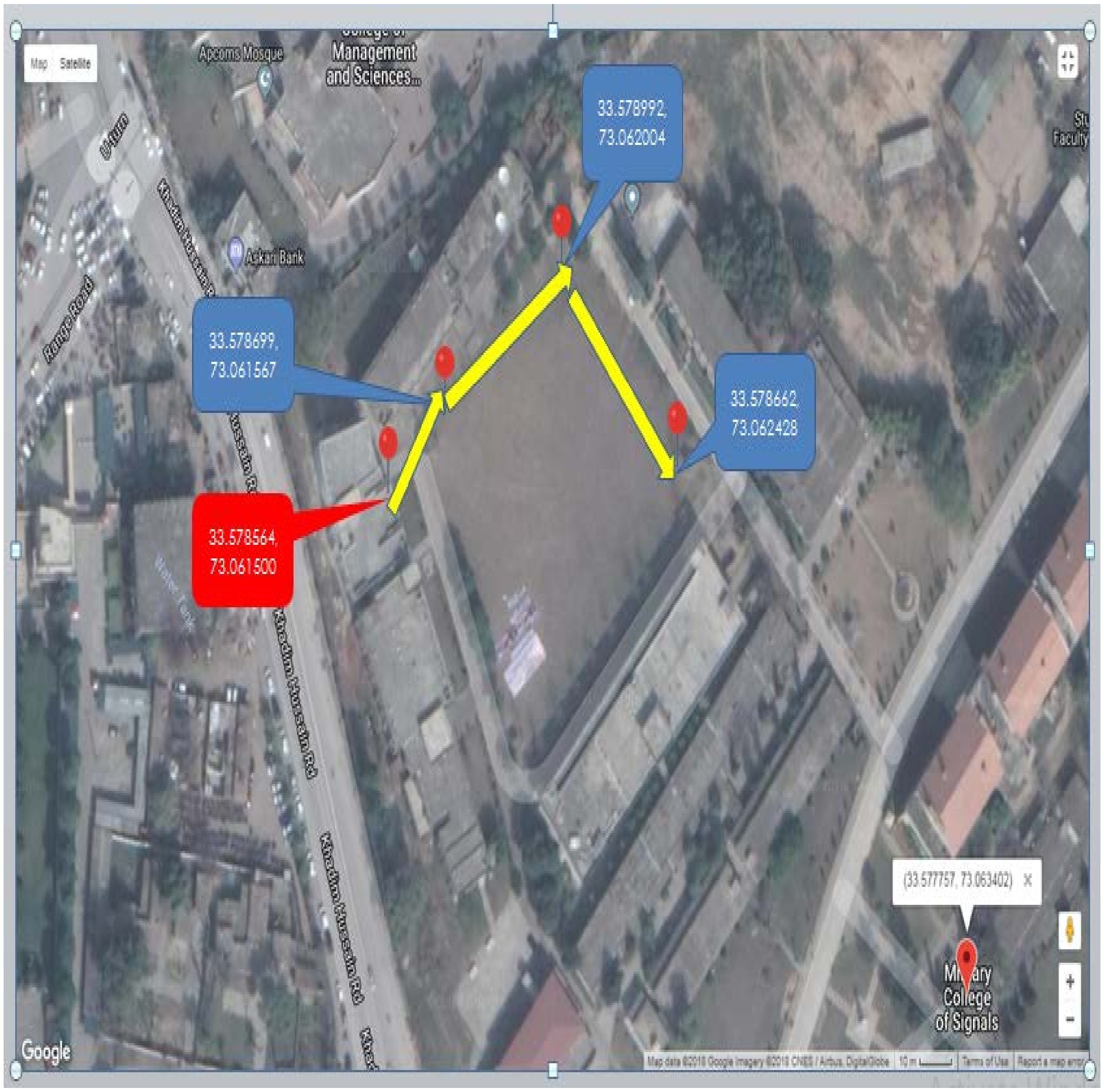
- If distance is 0 then load next waypoint, and repeat above steps. Else repeat above steps without loading next waypoint.
- If distance is 0 and waypoint was last, then Destination Reached.
- Way Point Distance Code.

Chapter 5: Demonstration Code / Hardware Implementation

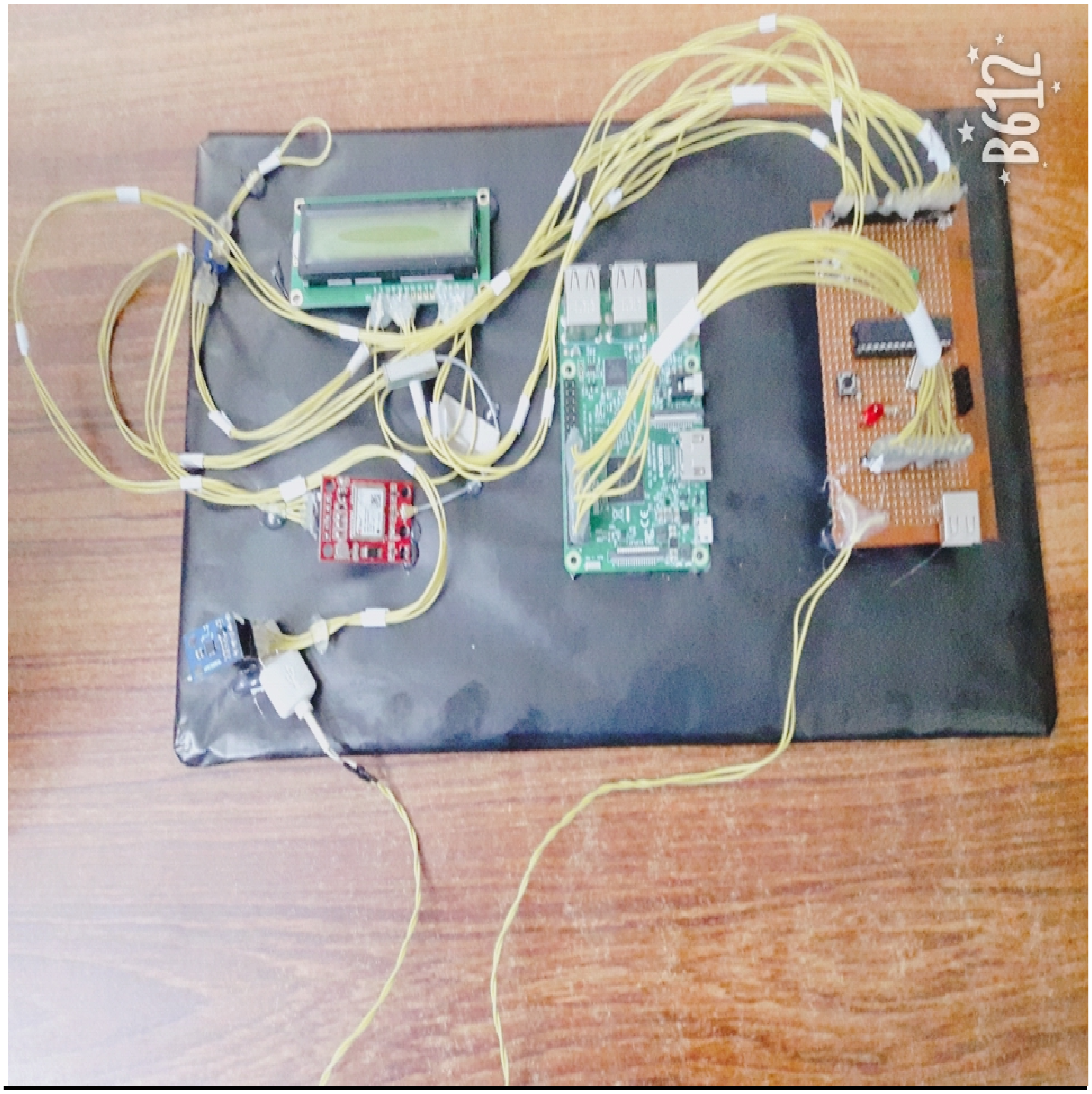
5.1 Demo Code


Windows PowerShell

```
PS C:\Users\Aamir Sajjad\Desktop\Lt Col Adil> python test.py
Target Lat: 33.578699
Target Long: 73.061567
Enter Current Lat: 33.578564
Enter Current Long: 73.061500
Distance to Target: 16.248485460476846
Target Heading: 22.4640056806776
Distance to Target: 16.248485460476846
Target Heading: 22.4640056806776
Enter Current Lat: 33.578564
Enter Current Long: 73.061500
Distance to Target: 16.248485460476846
Target Heading: 22.4640056806776
Enter Current Heading: 60
Turn Left
Enter Current Lat: 33.578564
Enter Current Long: 73.061500
Distance to Target: 16.248485460476846
Target Heading: 22.4640056806776
Enter Current Heading: 22
Go Straight
Enter Current Lat: 33.578699
Enter Current Long: 73.061567
Goto Next WayPoint!
Target Lat: 33.578992
Target Long: 73.062004
Enter Current Lat: 33.578699
Enter Current Long: 73.061567
Distance to Target: 51.979710095627865
Target Heading: 51.17355521029363
Distance to Target: 51.979710095627865
Target Heading: 51.17355521029363
Distance to Target: 51.979710095627865
Target Heading: 51.17355521029363
Enter Current Heading: 22
Turn Right
Enter Current Lat: 33.578699
Enter Current Long: 73.061567
Distance to Target: 51.979710095627865
Target Heading: 51.17355521029363
Enter Current Heading: 51
Go Straight
Enter Current Lat: 33.578992
Enter Current Long: 73.062004
Goto Next WayPoint!
Target Lat: 33.578662
Target Long: 73.062428
Enter Current Lat:
```


Map
Satellite
Apcoms Mosque
College of Management and Sciences...
Askari Bank
U-turn
Range Road
Khadim Hussain Rd
Water Park
33.578992, 73.062004
33.578699, 73.061567
33.578662, 73.062428
33.578564, 73.061500
(33.577757, 73.063402)
Military College of Signals
Google

5.2 Hardware Implementation



5.3 End Product

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Various applications exist that perform the functions of a smart cane but there is no application which gives the navigation feature while remaining offline. Our project has the capability to navigate and takes the individual from the start point to its destination through multiple way points. Furthermore, this project has the ability to operate in an offline mode but way points will be added manually before the start of journey.

6.2 Future Work

This idea can be polished further by adding the feature of taking voice input from the blind person and dividing the route into multiple way points automatically, which will remove the dependency of blind people on manual insertion of the way points.

Chapter 7: References

https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/

http://www.instructables.com/id/Arduino-Powered-Autonomous-Vehicle/

http://robotshop.com/letsmakerobots/fundamentals-a-gps-guided-vehicle

https://sites.google.com/site/wayneholder/self-driving-rc-car

http://www.instructables.com/id/Boat-Autopilot/

https://makezine.com/projects/make-37/gps/

https://www.latlong.net/

Chapter 8: Bibliography

[1] "Python Algorithms: mastering basic algorithms in the Python language", Choice Reviews Online, vol. 48, no. 10, pp. 48-5731-48-5731, 2011.

[2] C. Andrews, "Easy as Pi [Raspberry Pi]", Engineering & Technology, vol. 8, no. 3, pp. 34-37, 2013.