LANDMINE DETECTION ROBOT

by

Aakash A	20BEC1294
Karanam Venkata Sai Eswar	20BEC1266
Chanikya Annabathuni	20BEC1047
Bairampally Nikhilesh goud	20BEC1011

A project report submitted to

Dr. SOFANA REKA S

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE2010 – CONTROL SYSTEMS

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

NOVEMBER 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled "**LANDMINE DETECTION ROBOT"** is a bonafide work of **Aakash A-20BEC1294, Karanam Venkata Sai Eswar – 20BEC1266, Chanikya Annabathuni-20BEC1047, Bairampally Nikhilesh Goud- 20BEC1011** carried out the Project work under my supervision and guidance for **ECE2010 – CONTROL SYSTEM**

Dr. SOFANA REKA S

Assistant Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

The Landmine detecting robots are designed to cover maximum possible area of landmine field for detection of landmines. The detected landmines along with scanned and leftover area are represented on a visual map with accuracy in millimeters. This paper presents a prototype model of landmine detecting robot that is powerful yet low cost and easily controllable. A graphical user interface is developed for plotting the landmines, scanned & leftover area presentation, PID tuning and camera alignment. Emphasis is placed on the control of the differential drive robot in auto mode, semi-automode and the manual mode. Image processing technique is employed to find the accurate position of robot which provides the live reckoning feedback to the dead reckoning servo control of the robot. Metal detector is the sensor used to detect landmines. The graphical user interface for the remote terminal computer provides the effective control for the robot. The system is simple but powerful and intelligible to achieve the required results.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. SOFANA REKA S,** Assistant Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of School ofElectronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of Department **Dr. Mohanaprasad** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

A Chanikya

KARANAM VENKATA SAI ESWAR **CHANIKYA ANNABATHUNI**

Aakash A

`

AAKASH A **BAIRAMPALLY NIKHILESH GOUD**

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES

A few of the objectives of the given project are:

- Designing a low-cost mobile robot.
- Proposing a method to detect landmines by integrating more than one technique.
- Using effective motion planning algorithm.
- Ensuring the demining operators safety.

1.2 APPLICATIONS

Landmine detection is a process to determine whether a signal from GPR contains a landmine, whereas identification includes not only detecting a landmine in the signal but also determining the type of landmine. Therefore, identification is more involved than detection, providing more information about a landmine.

This Landmine Detection Robotic Vehicle uses a metal sensing coil to detect mines hidden under the surface of the earth. It's tracked design allows it to traverse across the roughest and wildest terrains. This Robot has been equipped with ultrasonic sensor to help it with obstacle sensing.

1.3 FEATURES

In this project, the implementation to develop landmine detection robot, has been done using the MathWorks Simulink software.

Simulink is a block diagram environment used to design systems with multidomain models, simulate before moving to hardware, and deploy without writing code. It helps to explore a wide design space and test the systems early with multidomain modelling and simulation by deploying models for desktop, real-time, and Hardware-in-the-Loop testing.

Simscape library:

Simscape enables you to rapidly create models of physical systems within the Simulink environment. With Simscape, you build physical component models based on physical connections that directly integrate with block diagrams and other modeling paradigms. You model systems such as electric motors, bridge rectifiers, hydraulic actuators, and refrigeration systems, by assembling fundamental components into a schematic.
Simscape add-on products provide more complex components and analysis capabilities.

Simscape helps you develop control systems and test system-level performance. You can create custom component models using the MATLAB based Simscape language, which enables text-based authoring of physical modeling components, domains, and libraries. You can parameterize your models using MATLAB variables and expressions, and design control systems for your physical system in Simulink. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape supports C-code generation.

Spectrum Analyzer:

The spectrum Analyzer object displays frequency-domain signals and the frequency spectrum of time-domain signals. The scope shows the spectrum view and the spectrogram view. The object performs spectral estimation using the filter bank method and Welch's method of averaged modified periodograms. You can customize the spectrum analyzer display to show the data and the measurement information that you need. The spectrum analyzer can display the power spectrum of the signal in three units, Watts, dBm, and dBW. To display the spectra of signals in the Spectrum Analyzer:

1. Create the spectrum Analyzer object and set its properties.
2. Call the object with arguments, as if it were a function.

Utilities:

This library contains required and commonly used blocks for setting simulation environment, interfacing with Simulink models, and generating custom components on-the-fly.The blocks in the Network Couplers library provide you with starting points for splitting your Simscape network into separate smaller networks interfaced to each other through Simulink connections.

2. DESIGN

2.1 FLOW CHART

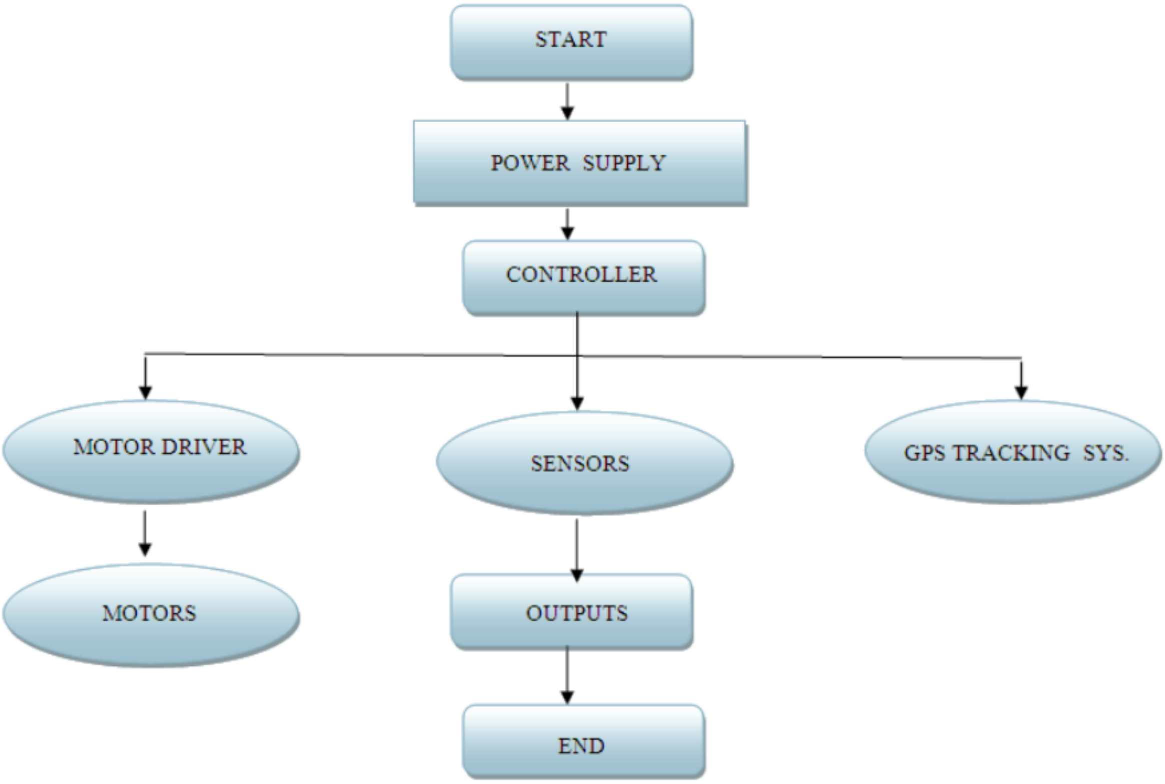


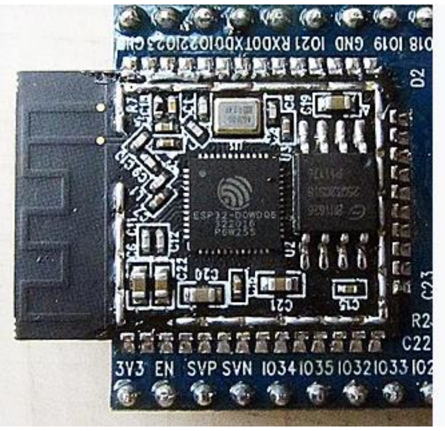
Figure. 1 Flow chart of the proposed idea

2.2 HARDWARE BLOCKS DESCRIPTION:

ESP 32 MICROCONTROLLER:

ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI or SDIO or I2C / UART interfaces.

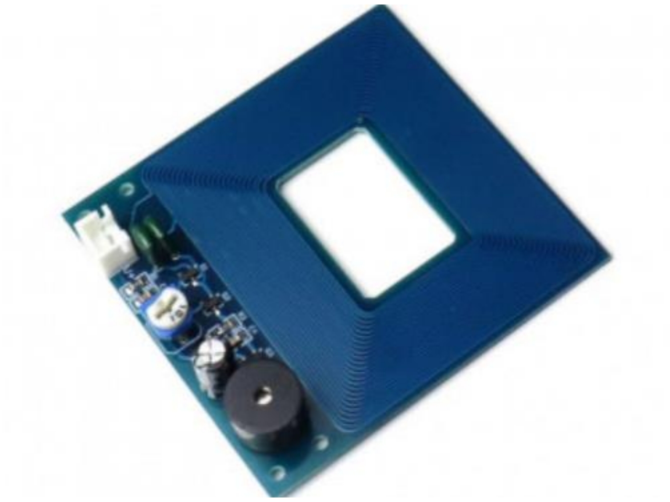
ESP32 is highly-integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. ESP32 adds priceless functionality and versatility to your applications with minimal Printed Circuit Board (PCB) requirements.



A88 METAL DETECTOR:

This is a module specifically designed to detect metal. The module operates by inducing currents in metal objects and responding when it occurs. A nice onboard buzzer signals when it detects something and an onboard potentiometer allow adjustment of sensitivity.

The power cables of the Metal detector non-contact metal induction detection module will need soldering on for the module to function, positive to the outside of the module and negative between the potentiometer and an electrolytic capacitor.



NEO 6M GPS MODULE:

The NEO-6M GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, you can monitor the status of the module. Thanks to the data backup battery, the module can save the data when the main power is shut down accidentally. Its 3mm mounting holes can ensure easy assembly on your aircraft, which thus can fly steadily at a fixed position, return to Home automatically, and automatic waypoint flying, etc.

3. SOFTWARE –CODING AND RESULTS

3.1 CODE / IMPLEMENTATION SETUP

```
#include <WiFi.h>
#include <TinyGPS++.h> // library for GPS module
#include <SoftwareSerial.h>
const char* ssid = "vivo";
const char* password = "12345678";
TinyGPSPlus gps;  // The TinyGPS++ object
SoftwareSerial ss(4, 5);
float latitude , longitude;
int year , month , date, hour , minute , second;
String date_str , time_str , lat_str , lng_str;
int pm;
#define AOUT_PIN 36
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2
  pinMode(2, OUTPUT);
  digitalWrite(2, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP())
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(0, OUTPUT);
  pinMode(2, OUTPUT);

  digitalWrite(5, 0);
  digitalWrite(4, 0);

  digitalWrite(0, 1);
  digitalWrite(2, 1);

   int value = analogRead(AOUT_PIN); // read the analog value from sensor

  Serial.print("Metal detector ");
  Serial.println(value);
}

void loop() {
//gps
  while (ss.available() > 0) //while data is available
    if (gps.encode(ss.read())) //read gps data
    {
      if (gps.location.isValid()) //check whether gps location is valid
      {
        latitude = gps.location.lat();
        lat_str = String(latitude , 6); // latitude location is stored in a string
        longitude = gps.location.lng();
        lng_str = String(longitude , 6); //longitude location is stored in a string

    }
```

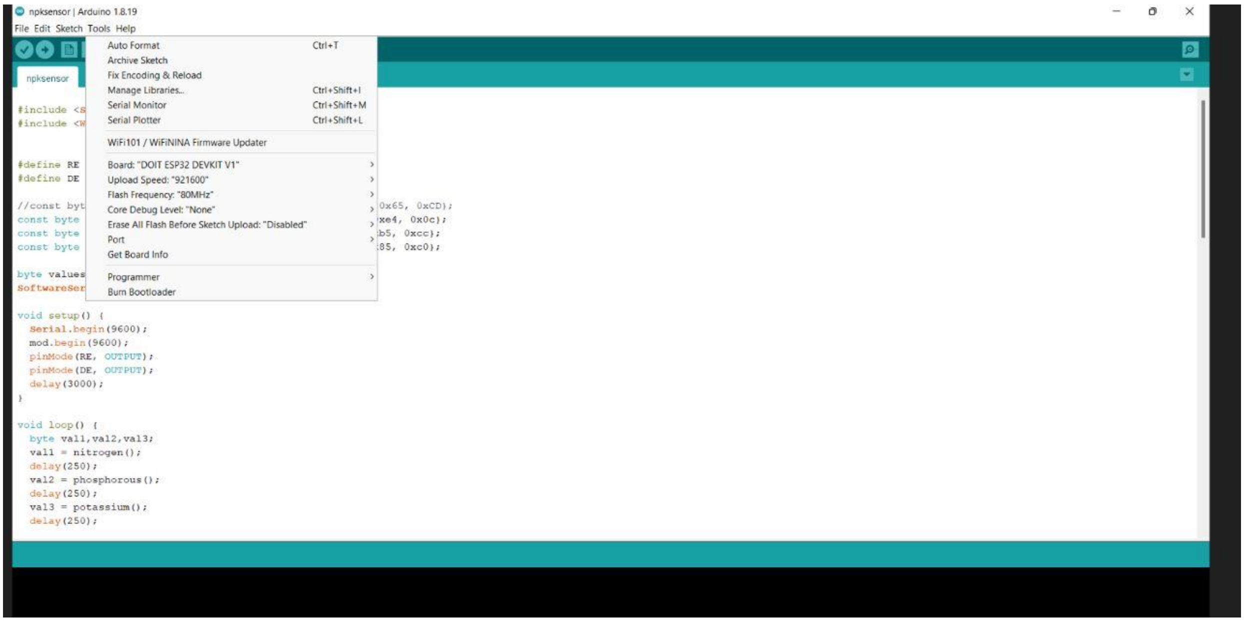
```
if (gps.date.isValid()) //check whether gps date is valid
   {
     date_str = "";
     date = gps.date.day();
     month = gps.date.month();
     year = gps.date.year();
     if (date < 10)
       date_str = '0';
     date_str += String(date);// values of date,month and year are stored in a string
     date_str += " / ";
     if (month < 10)
       date_str += '0';
     date_str += String(month); // values of date,month and year are stored in a string
     date_str += " / ";
     if (year < 10)
       date_str += '0';
     date_str += String(year); // values of date,month and year are stored in a string
   }
   if (gps.time.isValid())  //check whether gps time is valid
   {
     time_str = "";
     hour = gps.time.hour();
     minute = gps.time.minute();
     second = gps.time.second();
     minute = (minute + 30); // converting to IST
     if (minute > 59)
     {
       minute = minute - 60;
       hour = hour + 1;
     }
     hour = (hour + 5) ;
     if (hour > 23)
       hour = hour - 24;   // converting to IST
     if (hour >= 12)  // checking whether AM or PM
       pm = 1;
     else
       pm = 0;
     hour = hour % 12;
```

```
 if (hour < 10)
        time_str = '0';
      time_str += String(hour); //values of hour,minute and time are stored in a string
      time_str += " : ";
      if (minute < 10)

        time_str += '0';
      time_str += String(minute); //values of hour,minute and time are stored in a string
      time_str += " : ";
      if (second < 10)
        time_str += '0';
      time_str += String(second); //values of hour,minute and time are stored in a string
      if (pm == 1)
        time_str += " PM ";
      else
        time_str += " AM ";
    }
  }
   Serial.println("latitude");
  Serial.println("27.2046 N");

       delay(10000);
       Serial.println("longitude");
       Serial.println("77.4977 E ");
       delay(10000);

  WiFiClient client = server.available(); // Check if a client has connected
  if (!client)
  {
    return;
  }
  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n <!DOCTYPE html> <html> <head> <title>GPS
DATA</title> <style>";
  s += "a:link {background-color: YELLOW;text-decoration: none;}";
  s += "table, th, td </style> </head> <body> <h1  style=";
  s += "font-size:300%;";
  s += " ALIGN=CENTER> GPS DATA</h1>";
  s += "<p ALIGN=CENTER style=""font-size:150%;""";
```

```
s += "> <b>Location Details</b></p> <table ALIGN=CENTER style=";
s += "width:50%";
s += "> <tr> <th>Latitude</th>";
s += "<td ALIGN=CENTER >";
s += lat_str;
s += "</td> </tr> <tr> <th>Longitude</th> <td ALIGN=CENTER >";
s += lng_str;
s += "</td> </tr> <tr>  <th>Date</th> <td ALIGN=CENTER >";
s += date_str;
s += "</td></tr> <tr> <th>Time</th> <td ALIGN=CENTER >";
s += time_str;
s += "</td>  </tr> </table> ";

s += "</body> </html>";

// Check if a client has connected

if (!client) {
  analogWrite(5, 0);
  analogWrite(4, 0);
  digitalWrite(0, 1);
  digitalWrite(2, 1);
  return;
}

// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
  delay(1);
}

// Read the first line of the request
String req = client.readStringUntil('\r');
Serial.println(req);
client.flush();
```

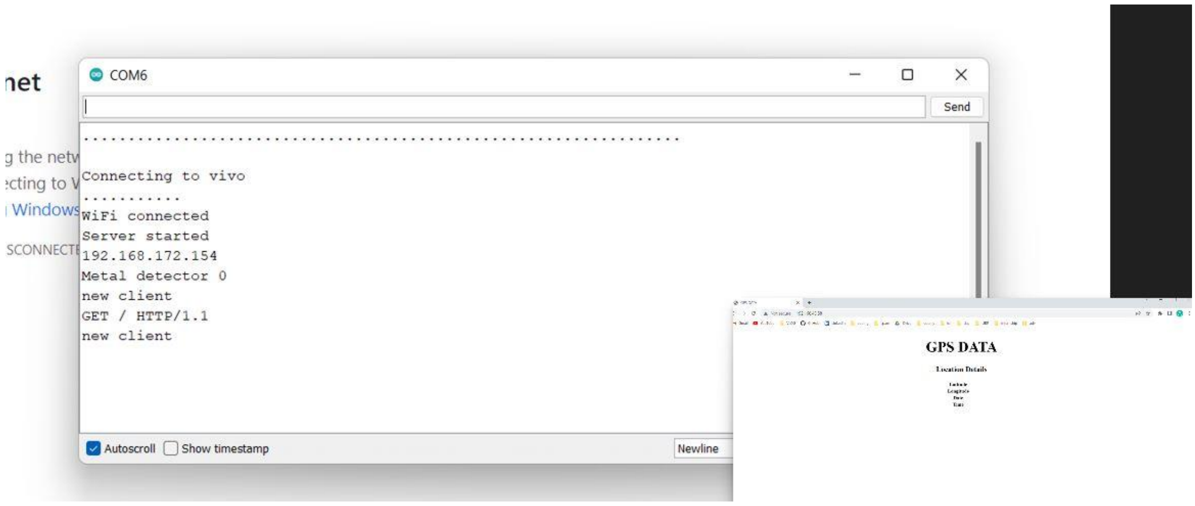
```
// Match the request
int motorASpeed = 1023;
int motorBSpeed = 1023;
int motorAForward = 1;
int motorBForward = 1;
if (req.indexOf("/engines/") != -1) {
  String parameters = req.substring(13);
  int separatorPos = parameters.indexOf(",");
  int httpPos = parameters.indexOf(" HTTP");
  String leftText = parameters.substring(0,separatorPos);
  String rightText = parameters.substring(separatorPos + 1, httpPos);
  Serial.println("[" + leftText +"][" + rightText + "]");
  int left = leftText.toInt();
  int right = rightText.toInt();
  if (left < 0) {
    motorAForward = 0;
  } else {
    motorAForward = 1;
  }
  if (right < 0) {
    motorBForward = 0;
  } else {
    motorBForward = 1;
  }
  analogWrite(5, abs(left));
  analogWrite(4, abs(right));
  digitalWrite(0, motorAForward);
  digitalWrite(2, motorBForward);
} else if (req.indexOf("/index.html") != - 1 || req.indexOf("/") != - 1) {
  client.print("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n");
  client.print("<html><head>");
  client.print("</head><body>");
  client.print("<script type='text/javascript' src='http://www.squix.org/blog/smartcar.js'></script>");
  client.print("<a href='#' onclick='move(\"f\");'>forward</a><BR/>");
  client.print("<a href='#' onclick='move(\"b\");'>backwards</a><BR/>");
  client.print("<a href='#' onclick='move(\"l\");'>left</a><BR/>");
  client.print("<a href='#' onclick='move(\"r\");'>right</a><BR/>");
  client.print("<div id=\"dmEvent\"/>");
  client.print("<div id=\"vector\"/>");
  client.print("</body></html>");

  analogWrite(5, 0);
  analogWrite(4, 0);
  digitalWrite(0, 1);
  digitalWrite(2, 1);
  return;
 }
 client.flush();

 // Prepare the response
 s += "</html>\n";

 // Send the response to the client
 client.print(s);
 delay(1);
 Serial.println("Client disonnected");
 delay(200);
 // The client will actually be disconnected
 // when the function returns and 'client' object is detroyed
}
```

3.2 SOFTWARE RESULTS :

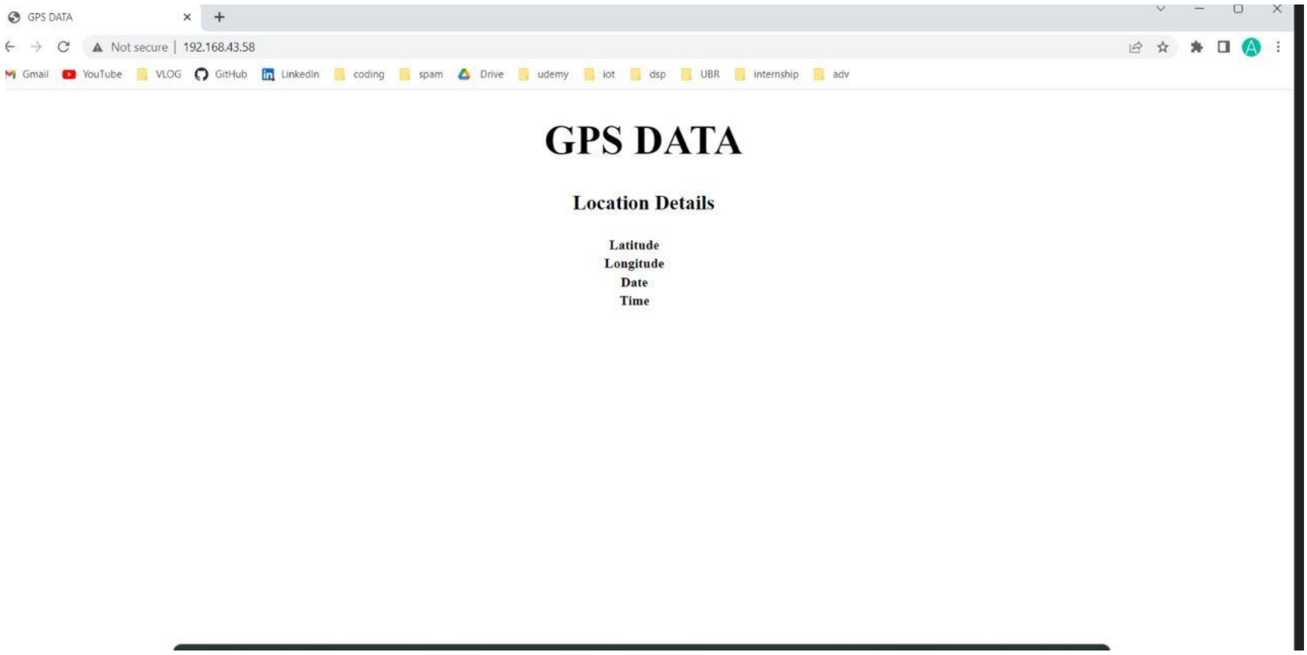

npksensor | Arduino 1.8.19
File Edit Sketch Tools Help
npksensor
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Programmer
Burn Bootloader

Board: "DOIT ESP32 DEVKIT V1"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Core Debug Level: "None"
Erase All Flash Before Sketch Upload: "Disabled"

```
#include <S
#include <W

#define RE
#define DE

//const byt                                          0x65, 0xCD};
const byte                                           xe4, 0x0c};
const byte                                           b5, 0xcc};
const byte                                           85, 0xc0};

byte values
SoftwareSer

void setup() {
  Serial.begin(9600);
  mod.begin(9600);
  pinMode(RE, OUTPUT);
  pinMode(DE, OUTPUT);
  delay(3000);
}

void loop() {
  byte val1,val2,val3;
  val1 = nitrogen();
  delay(250);
  val2 = phosphorous();
  delay(250);
  val3 = potassium();
  delay(250);
```



```
...................................................................

Connecting to vivo
...........
WiFi connected
Server started
192.168.172.154
Metal detector 0
new client
GET / HTTP/1.1
new client
```



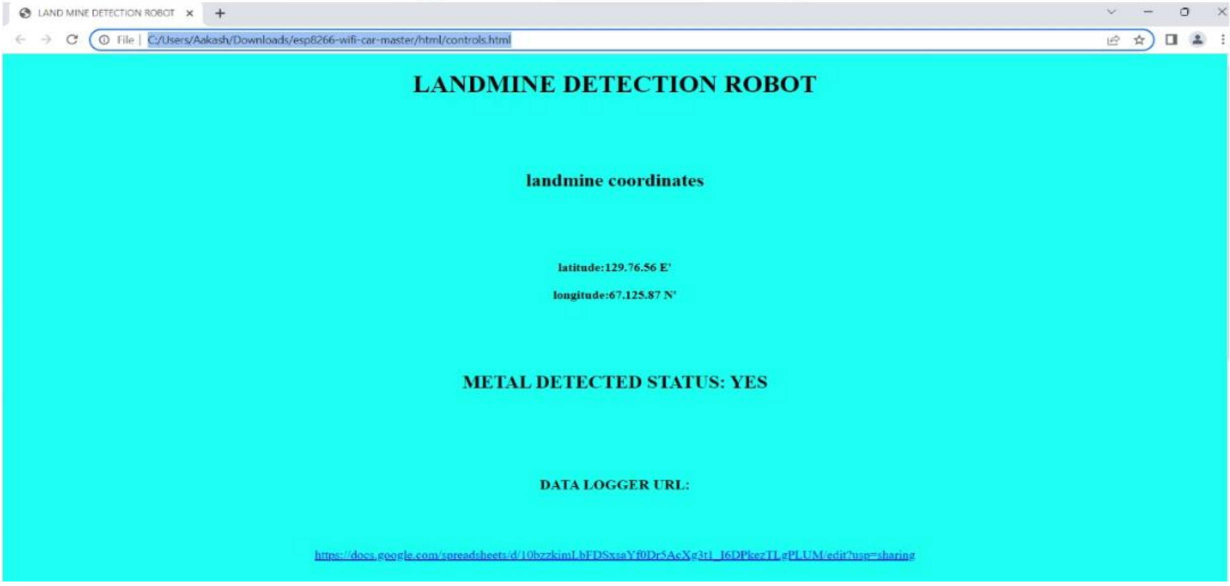
GPS DATA

Location Details

Latitude
Longitude
Date
Time

3.3 GLIMPSE OF WEBSITE

WEB CONTROLLER



LAND MINE DETECTION ROBOT
File
C:/Users/Aakash/Downloads/esp8266-wifi-car-master/html/controls.html

LANDMINE DETECTION ROBOT

landmine coordinates

latitude:129.76.56 E'

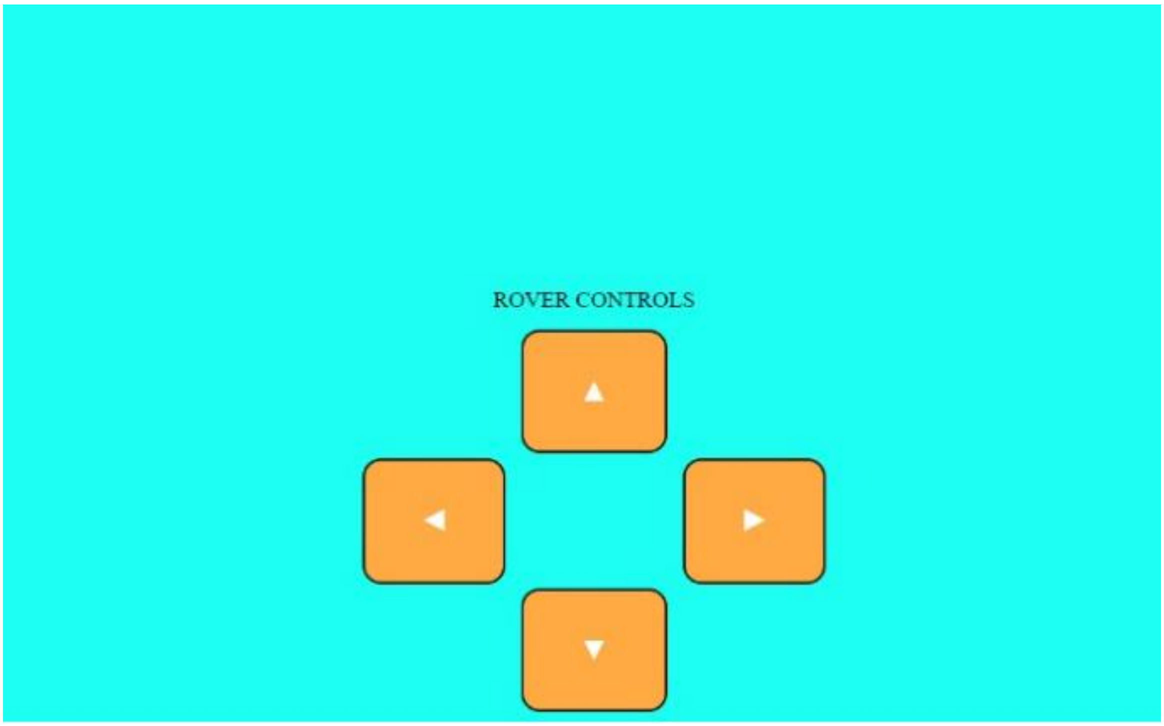
longitude:67.125.87 N'

METAL DETECTED STATUS: YES

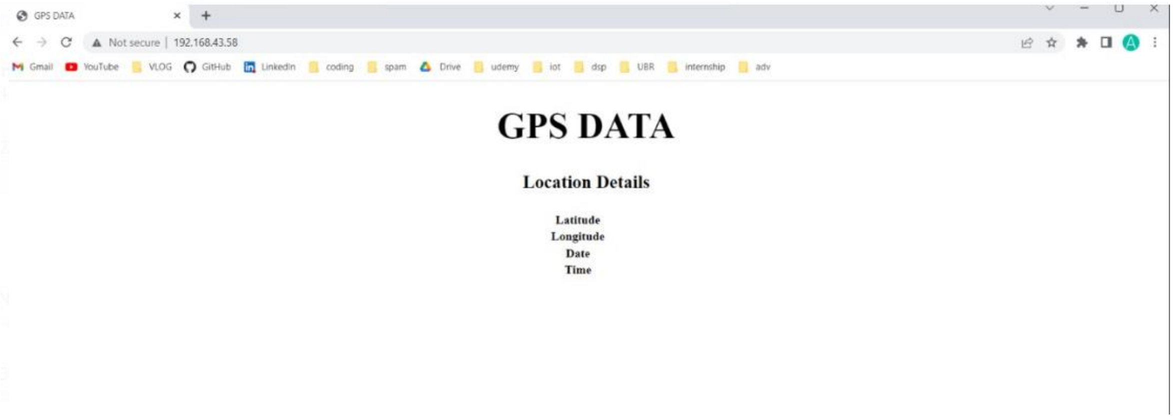
DATA LOGGER URL:

https://docs.google.com/spreadsheets/d/10bzzkimLbFDSxsaYf0Dr5AcXg3t1_I6DPkezTLgPLUM/edit?usp=sharing

ROVER CONTROL



WEB SERVER



GPS DATA

Location Details

Latitude
Longitude
Date
Time

4 . SIMULINK AND HARDWARE CONSTRUCTION

4.1 SIMULINK

Metal Detector Circuit

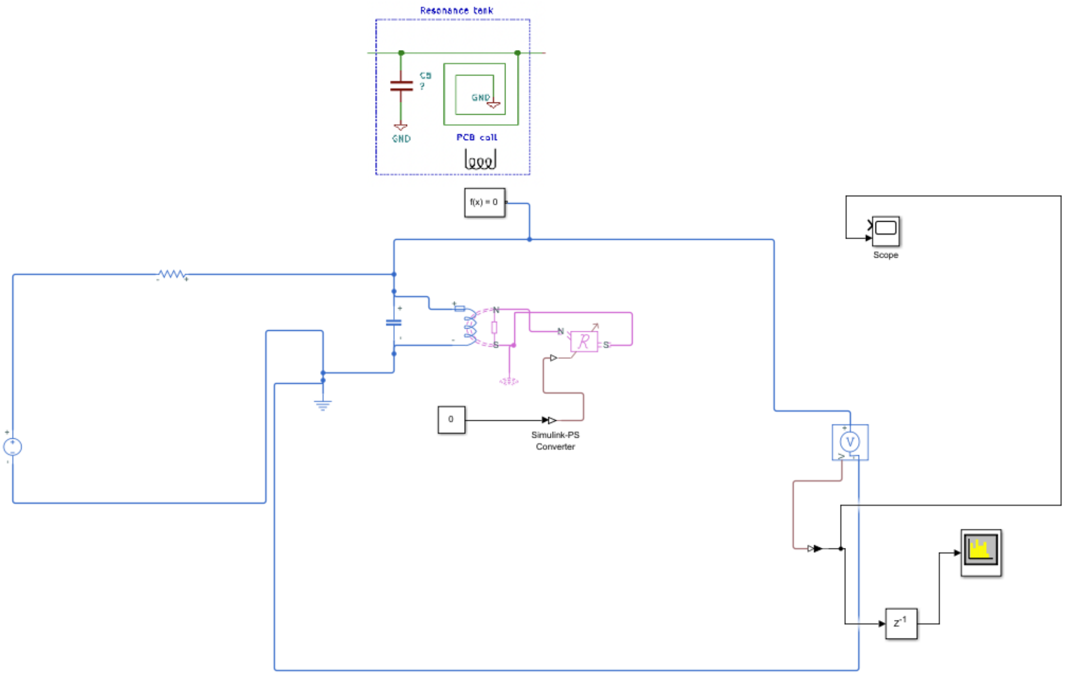
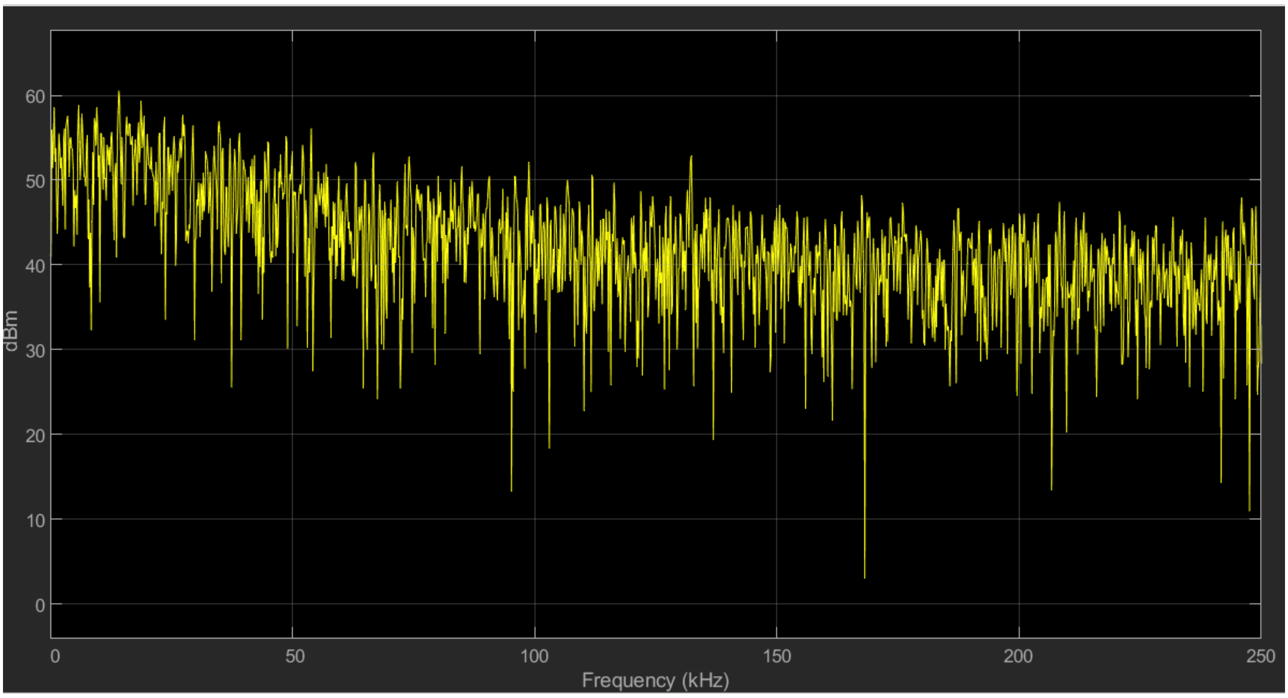
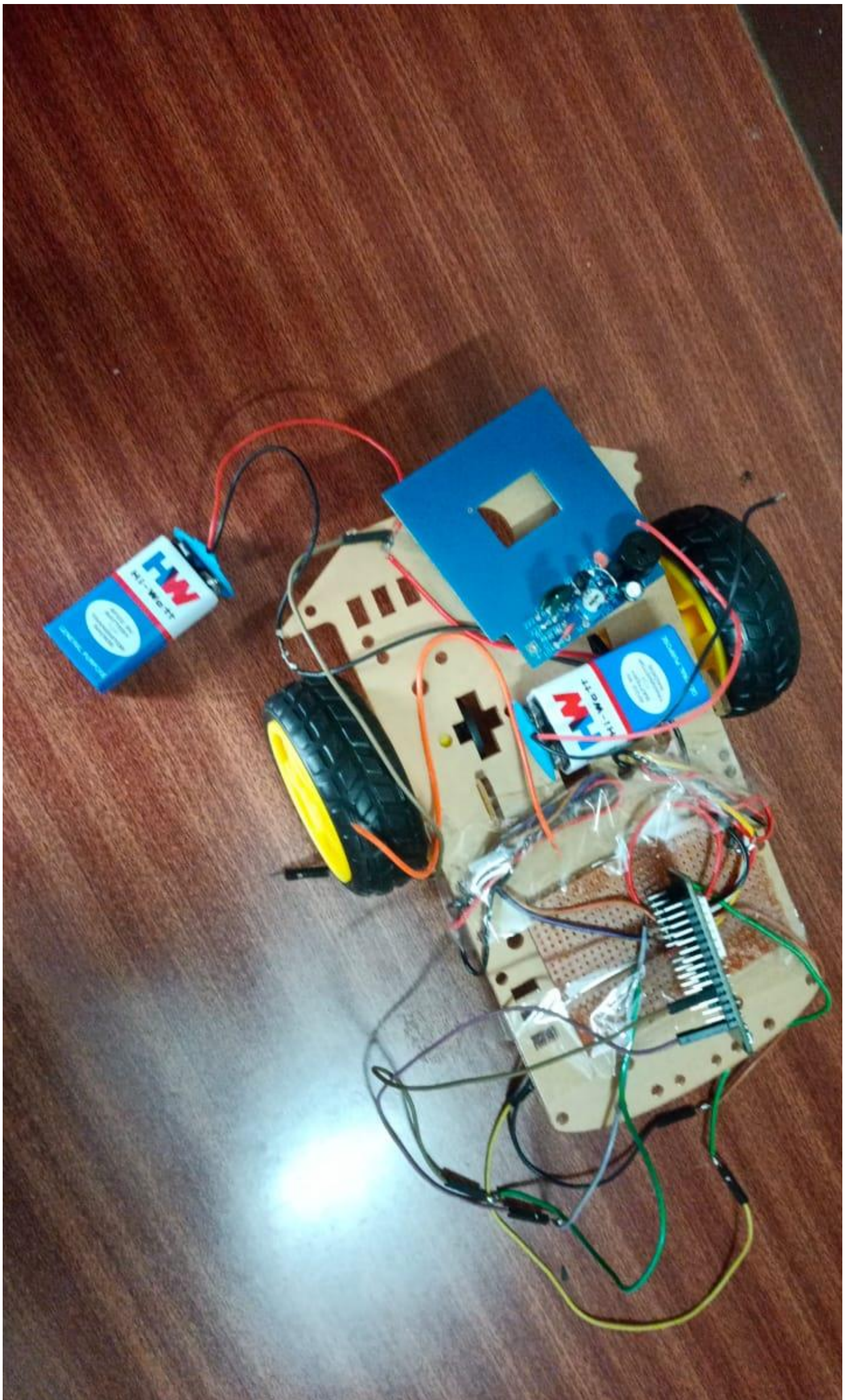


Figure: schematic of metal detector circuit

Spectrum Analyzer



4.2 HARDWARE CONSTRUCTION



5. CONCLUSION AND FUTURE WORK

5.1 CONCLUSION AND INFERENCE

This report has described overall design for wheeled robot for land mine detection purpose and implementation. The wheeled robot is less expensive, robust and it is a helpful tool in for military for surveying and monitoring purpose. Nothing should be more important than the lives and safety of our country's army men who risk their lives for our safety from external enemies. There have been many cases of fatalities and injuries due to explosion of landmines. Till date a lot of research and development has been done and different types of landmine detection robots have been developed each having its own advantages and disadvantages. The variation in these robots is based on the controller or processor used, sensor interfaced, GPS tracking system and the locomotion technique used.

5.2 FUTURE WORK

The future scope is concentrated on the improvement of the body designs by placing suspension system to over shock from the uneven surfaces. The robot is equipped with a camera for monitoring the condition of the robot. The power system is developed by replacing the battery with the solar panels to produce continuous power. The robot is equipped with a robotic arm for the diffusion purpose.

6. REFERENCES

[1] Habib M.K., "Mine detection and sensing technologies-new development potentials in the context of humanitarian demining," in Industrial Electronics Society,

[2] Zhenjun He, Jiang Zhang, Peng Xu, JiahengQin andYunkai Zhu, "Mine detecting robot based on wireless communication with multi-sensor".

[3] Schwarz, A, Zalevsky, Z, and Sanhedrai, Y.,"Digital camera sensing and its image disruption with controlled radio frequency transmission/reception".

[4] Christ.P,Neuwinger.B,Werner.F,Ruckert, U. "Performance analysis of the nRF24L01 ultra low-power transceiver in a multi-transmitter and multi-receiver scenario," in Sensors.

[5] Carullo A; Parvis, M., "An ultrasonic sensor for distance measurement in automotive applications".

[6] M. Parrilla, J. J. Anaya, and C. Fritsch ,"Digital signal processing techinques for high accuracy ultrasonic range measurement" .

[7] Brown C, Zoubir A.M, Chant, I.J, Abeynayake, C., "Landmine detection using single sensor metal detectors," in Acoustics, Speech, and Signal Processing.

[8] Sarm,G,H, Niti, G Ramanan, Manivanna and Mehta,K.,Bhattacharjee A, "Reliability studies on high current power modules with parallel mosfets".

[9] Jaradat M A, Bani Salim M N and Awad F H (2012), “Autonomous Navigation Robot for Landmine Detection Applications”.

[10] Kuo-Lan Su, Hsu-Shan Su, Sheng-Wen Shiao and Jr-Hung Guo (2011), “Motion Planning for a Landmine-Detection Robot”, Artificial Life and Robotics.

[11] Minh Dao-Johnson Tran, Canicious Abeynayake, Lakhmi C Jain and Lim C P (2010), “An Automated Decision System for Landmine Detection and Classification Using Metal Detector Signals”, Innovations in Defence Support Systems.

BIODATA

Name : KARANAM VENKATA SAI ESWAR
Mobile Number : 6281489479
E-mail : Karanam.saieswar2020@vitstudent.ac.in
Permanent Address : Balajiraopeta, ongole, Andhra Pradesh, 523001

Name : CHANIKYA ANNABATHUNI
Mobile Number : 8333850010
E-mail : chanikya.annabathuni2020@vitstudent.ac.in
Permanent Address : Itha nagar, Tenali, Andhra Pradesh, 522201

Name : BAIRAMPALLY NIKHILESH GOUD
Mobile Number : 8985537571
E-mail : bairampally.nikhilesh2020@vitstudent.ac.in
Permanent Address : nalanda autostand, mahabubnagar, Telangana, 509001

Name : AAKASH A
Mobile Number : 9176311830
E-mail : aakash.a2020@vitstudent.ac.in
Permanent Address : Ambal Nagar, Ramakrishna Avenue, Chennai, 600122