PROJECT REPORT

TWO WHEELER ANTI-THEFT

SYNDICATE

Lt KADNM Athukorola
SLt Utkarsh Mishra
SLt Senthil Ganesh P
SLt Praful G

PROJECT GUIDE

Lt Cdr Udit Chibber

Certificate of Merit

This is to certify that under mentioned officers of O-171 have successfully completed the project **"Two wheeler anti-theft"** and have been able to achieve the desired results as a partial fulfillment of requirements for 'L' specialization at INS Valsura.

Lt KADNM Athukorola	(NRL3564)
SLt Senthil Ganesh	(53274-A)
SLt Utkarsh Mishra	(53263-A)
SLt Praful G	(53285-A)

The project report compiled by them is an indication of the successful completion of the project under my guidance.

(Udit Chibber)
Lt Cdr
52844-Z

Acknowledgement

1. We would like to express our sincere thanks to our project guide, Lt Cdr Udit Chibber for his continuous guidance, support and motivation in working out the project.

2. We are grateful to SI FTP, Lt Cdr K Poswal and the staff of FTP for the necessary support and assistance.

(Senthil Ganesh P)
SLt
53274A

(Uthkarsh Mishra)
SLt
53263A

(Praful G)
SLt
53285A

(KADNM Athukorala)
Lt
NRL3564

ABSTRACT

Nowadays rate of vehicle theft is very high all through the world and the situation are even worse in developing countries. Therefore, protection of vehicles with an intelligent, reliable, effective and economical system is very important. The existing technologies for vehicle security have a number of limitations including high false alarm rate, easy deactivation and high cost. In this project an Anti-Theft Vehicle Security System has been designed and implemented utilizing Global Positioning System (GPS) and Global System for mobile communication (GSM) technology to track the vehicle. The cutting edge technology is capable to protect, monitor and track the vehicle even within a minute.

INDEX

CHAPTER 1

INTRODUCTION

The project titled “Two Wheeler Anti-Theft ” aims in prototyping of a security and tracking system for automobiles. The ESP 32 microcontroller is chosen for this system so as to allow any future improvisations as well as to support many features at an optimum cost. The GSM and GPS modules are also used to track the vehicle on any device connected to network apart from the SMS of the location being sent to the mobile number at a time as specified by the user. The location of the vehicle is shown on a web page which can be seen by different people. This does not only help in case of theft but can also be used to monitor the vehicle.

CHAPTER 2

BLOCK DIAGRAM AND MATERIALS USED

2.1 Block Diagram

The overall concept of the project is depicted below in the form of a block diagram as shown in Fig 1.

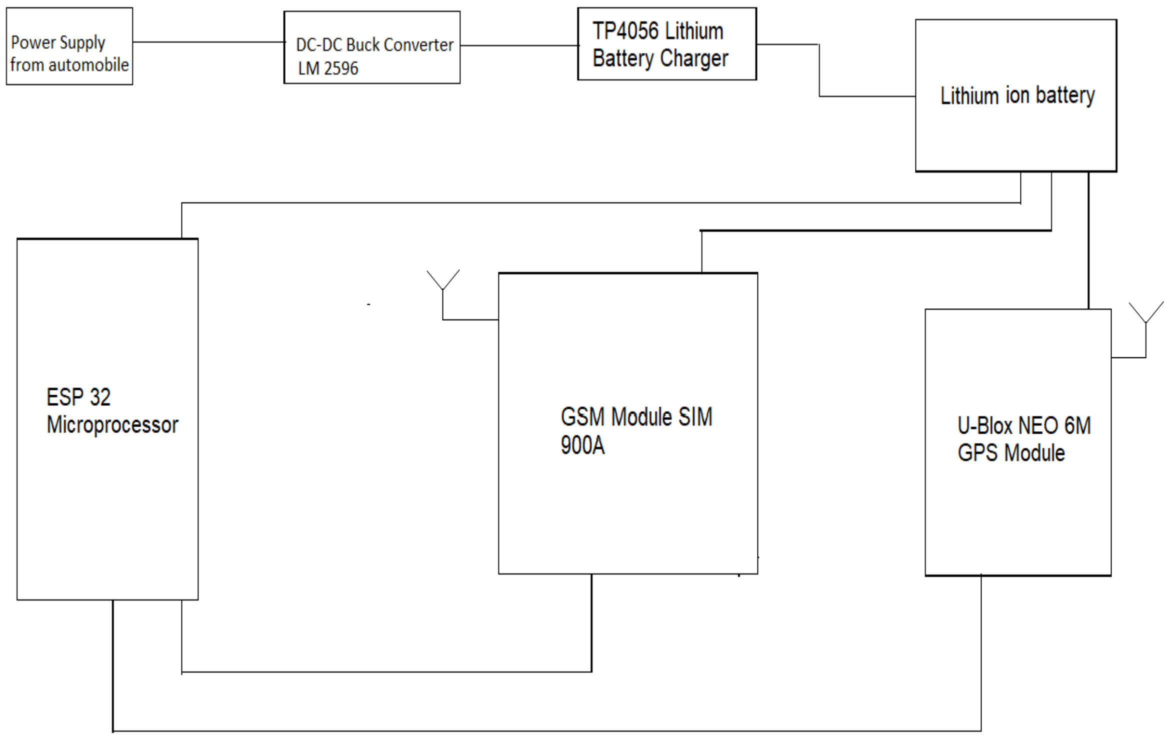

Fig 1 : Block diagram

2.2 ESP 32 Module

ESP32 (shown in Fig 2) is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtens LX6 microprocessor in both dual-core and single-core variations and includes in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems.

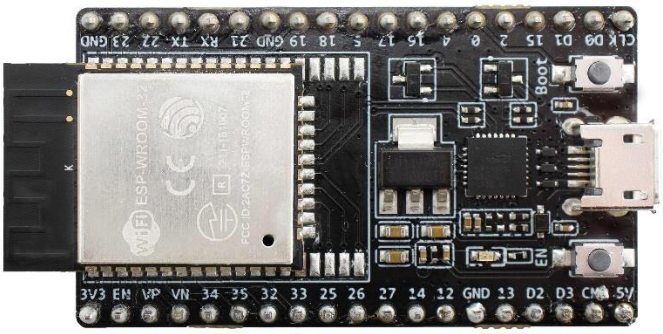

Fig 2 : ESP32 Module

Components with specific features have been utilized by the syndicate. Details of the components used are given below

(a) Robust Design. ESP32 is capable of functioning reliably in industrial environments, with an operating temperature ranging from –40°C to +125°C. Powered by advanced calibration circuitries, ESP32 can dynamically remove external circuit imperfections and adapt to changes in external conditions.

(b) High Level of Integration. 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.

(c) Ultra-Low Power Consumption. Engineered for mobile devices, wearable electronics and IoT applications, ESP32 achieves ultra-low power consumption with a combination of several types of proprietary software. ESP32 also includes state-of-the-art features, such as fine-grained clock gating, various power modes and dynamic power scaling.

Hardware Specifications

SRAM : 520MB

ROM	:	448KB
Operating Voltage	:	2.3V – 4.6V
Output Current	:	500mA
Processor	:	32-bit LX6 microprocessor

The Pinout of ESP 32 is depicted in Fig 3.

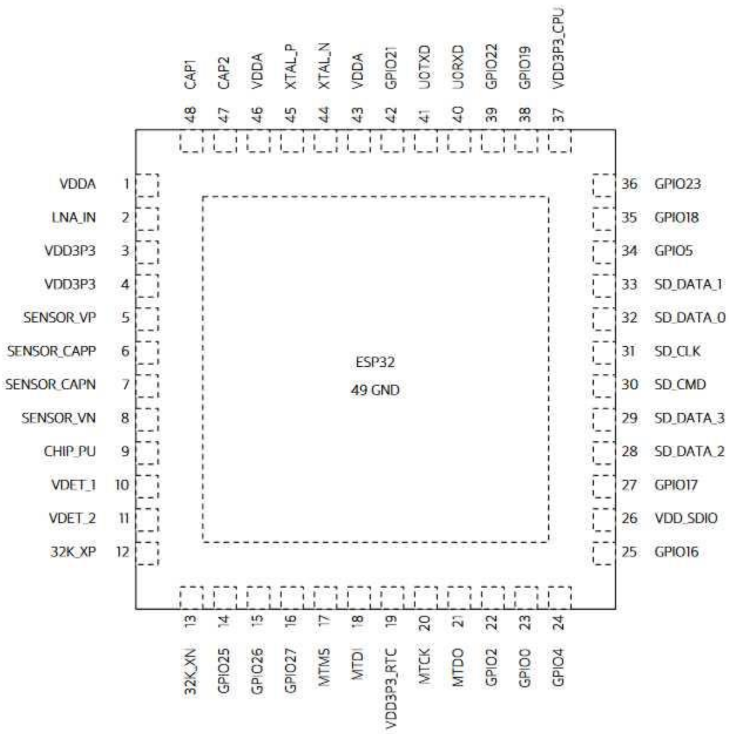

Fig 3 : ESP32 Module pin diagram

2.3 GPS Module

GPS (or global positioning system), is a satellite-based navigation system made up of a network of 24 satellites orbiting the earth at all times. This system, when the proper signals are received, allows a user to tell where they are located. Depending on many different factors, the accuracy can range from as close to 6 feet to as far away as 300 feet. GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's approximate location. Hence this will help to give us the location of the vehicle at any time. The GPS module along with the antenna is shown in Fig 4.

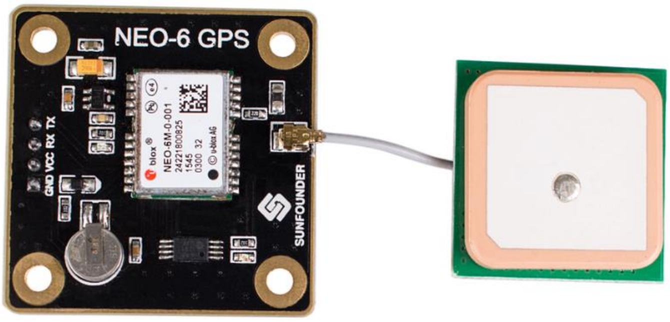

Fig 4 : NEO 6M GPS Module

The GPS module used in the project is U-Blox NEO 6M GPS module with EEPROM. The 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. The data backup battery can save the data when the main power is shut down accidentally. The EEPROM is used to save the configuration settings and the module can be configured at any point of time. It also has a rechargeable battery for back up.

Hardware Specifications

Supply voltage	:	3.6 V
Data rate	:	5Hz location update rate
Baud rate	:	4800 Bd - 11520 Bd
Antenna type	:	18 mm * 18 mm GPS antenna
Sensitivity	:	162 dBm tracking sensitivity
Hot start	:	1 s
Cold start	:	38 s

2.4 GSM Module

A GSM modem is a device which can be either a mobile phone or a modem device which can be used to make a computer or any other processor communicate over a network. The GSM module is shown in Fig 6.

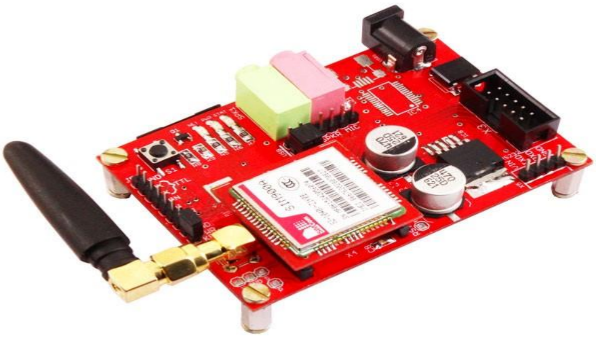

Fig 6 : GSM Module SIM 900A

. A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator. It can be connected to a computer through serial, USB or Bluetooth connection.

A GSM modem can also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. GSM modem is usually preferable to a GSM mobile phone. The GSM modem has wide range of applications in transaction terminals, supply chain management, security applications, weather stations and GPRS mode remote data logging.

The GSM module used in the project is GSM module – SIM 900A. The GSM/GPRS TTL UART Modem, from rhydoLABZ, is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with selectable interfacing voltage, which allows you to connect 5V & 3V3 microcontroller directly without any level conversion chips. The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface.

Hardware Specifications

Input voltage : 5 V – 12 V DC

Dual band : 900 MHz / 1800MHz

Baud rate : 9600 Bd - 115200 Bd

The Pinout of GSM Module SIM 900A is shown in Fig 7.

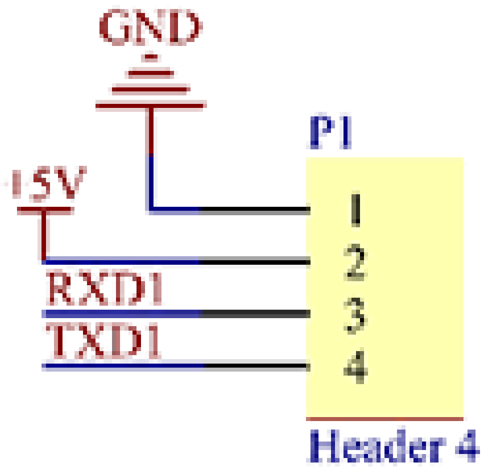

Fig 7 : GSM SIM 900A pin diagram

2.5 DC - DC Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage from its input to its output. The LM2596 DC-DC Buck Converter is shown in Fig 8.

Fig 8 : LM2596 DC-DC Buck Converter

. It is a class of switched-mode power supply typically containing at least two semiconductors and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors are normally added to such a converter's output and input.

LM2596 DC-DC buck converter step down module is used in the project in order to step down the power supply voltage so that a suitable voltage is given to other components. DC-DC Buck Converter Step Down Module LM2596 Power Supply is a step-down switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. The input voltage to the module given by the power supply is 12V and it down converts this voltage to 5V for the lithium battery charger module.

Hardware specifications:

Input Power supply	:	4V – 35V
Output Voltage	:	1.25V – 30V
Output current	:	3A (MAX)
Switching Frequency	:	150KHz

2.6 Lithium battery charging Module

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. The TP4056 Lithium Battery Charger is shown in Fig 9.

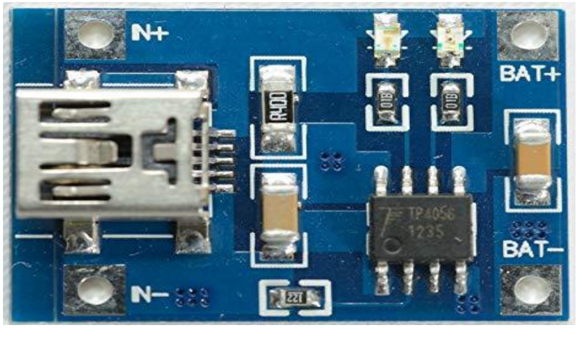

Fig 9 : TP4056 Lithium Battery Charger

Furthermore, the TP4056 can work within USB and wall adapter. No blocking diode is required due to the internal PMOSFET architecture and have prevent to negative Charge Current Circuit. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor.

Hardware Specifications:

Input Supply Voltage	:	4V – 8V
Input Current	:	500uA (MAX)
Charge Voltage	:	4.2V
Programmable current	:	1000mA

CHAPTER 3

SOFTWARE PROGRAMMING AND INTERFACING

3.1 Proposed System

The proposed system should have the following features. The tracking and monitoring of the vehicle at all times, so it should be programmed in such a way that when the vehicle is moved after a particular period of time (which is set at 30 minutes) for a particular distance (which is set at 150m) we should receive an alarm in the form of an SMS on our mobile phones and the vehicle's location can be tracked. Technically it means that the ESP 32 microprocessor should be programmed in such a way that the GSM and GPS modules work at the particular time. The anti-theft system is programmed such that the GPS sends the location of the vehicle to a google spread sheet which contains the longitude and the latitude of the location. These readings are then fed to the google spreadsheet and subsequently to html webpage which is shown in the web page. An ariduino IDE has been used to programme the ESP 32 module so that the anti-theft system could work in this way.

3.2 Code

(a) The following code has been used to programme the microcontroller to perform the required function of sending the SMS and configuration of the GSM module

```
#include <TinyGPS++.h>

TinyGPSPlus gps;

String latitude;

String longitude;

float lat1=0;

float lon1=0;

float lat2=0;
```

```
float lon2=0;
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
const long interval = 60000;
unsigned long previousMillisGSM = 0;
const long intervalGSM = 180000;
bool alert =LOW;
void setup() {
     pinMode(LED_BUILTIN, OUTPUT);
  // initialize both serial ports:
  Serial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600);
  Serial.println("Starting....");
   blinkLed ();
 // for (int x=0; x<5 ;x++){
    checkGSM();
}
void loop() {
  sendData();
  lat1 =gps.location.lat();
  lon1= gps.location.lng();
  currentMillis = millis();
  if((lat1-lat2)>0.0002){            //Condition for activation of system
    previousMillis = currentMillis;
    Serial.print("time reset");
    lat2=lat1;
    if(alert){
      sendSMS1 ();
      alert =LOW;
```

```
}
 }
 if((lat2-lat1)>0.0002){
 previousMillis = currentMillis;
 Serial.print("time reset");
 lat2=lat1;
 if(alert){
   sendSMS1 ();
   alert =LOW;
 }
 }
 if((lon1-lon2)>0.0002){
 sendSMS1 ();
 previousMillis = currentMillis;
 Serial.print("time reset");
 lon2=lon1;
 if(alert){
   sendSMS1 ();
   alert =LOW;
 }
 }
if((lon2-lon1)>0.0002){
 sendSMS1 ();
 previousMillis = currentMillis;
 Serial.print("time reset");
 lon2=lon1;
if(alert){
   sendSMS1 ();
   alert =LOW;
 }
```

```
    }
    blinkLed ();
    if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    alert =HIGH;
    Serial.print("time exceed");
  }
  if (currentMillis - previousMillisGSM  >= intervalGSM) {
    previousMillisGSM = currentMillis;
    checkGSM() ;
  }
}
void sendData(){                                   //GSM Configuration
  Serial2.print("AT+HTTPINIT\r\n");
  smartDelay(1000);
  Serial2.print("AT+HTTPPARA=\"CID\",1\r\n");
  smartDelay(1000);
  Serial2.print("AT+HTTPPARA=\"REDIR\",0\r\n");
  smartDelay(1000);
  Serial2.print("AT+HTTPSSL=1\r\n");
  smartDelay(1000);
Serial2.print("AT+HTTPPARA=\"URL\",\"https://script.google.com/macros/s/AKfycbwn2ch3mF_TDmEmcoPlsAqVjHnuQE_BwcFyuHPlxUjZDFbmnI/exec?lat=");
  Serial2.print(gps.location.lat(),6);
  Serial2.print("&lon=");
  Serial2.print(gps.location.lng(),6);
  Serial2.print("\"\r\n");
  smartDelay(1000);
  Serial2.print("AT+HTTPACTION=0\r\n");
  smartDelay(15000);
```

```
while (Serial2.available()) {
   Serial.write(Serial2.read());
  }
  Serial2.print("AT+HTTPREAD\r\n");
  smartDelay(8000);
  while (Serial2.available()) {
   Serial.write(Serial2.read());
  }
  }
static void smartDelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (Serial.available())
      gps.encode(Serial.read());
  } while (millis() - start < ms);
}
void checkGSM1() {
  Serial2.print("AT\r\n");
  Serial2.print("AT\r\n");
  smartDelay(2000);
    while (Serial2.available()){
         Serial2.read();
         }
  Serial2.print("AT+SAPBR=2,1\r\n");
  smartDelay(1000);
  char incomming;
  String response="";
```

```
    int cnt=0;


  while (Serial2.available()){

      cnt++;

      incomming = Serial2.read();

      if(incomming=='\r'){ Serial.print("r"); }

        else if(incomming=='\n'){ Serial.print("n"); }

          else if(incomming==' '){ Serial.print("$"); }

            else {Serial.print(incomming);}

       Serial.println(cnt);

      response += incomming;

      }

    if(response=="AT+SAPBR=1,1\r\n" ){

      Serial.print("response ok");

      }

    else {

        Serial.print("respnoce is deference =");

        Serial.println(response);

          }

  }

  void check_AT_Response (){

          String response;

          char incomming;

          while (Serial2.available()){

              Serial.print( Serial2.read());

              }

          while (1){

            Serial2.print("AT\r\n");

            smartDelay(3000);

            while(Serial2.available()){
```

```
                incomming = Serial2.read();

                Serial.write(incomming);

                response += incomming;

                }

          if(0 <= response.indexOf("AT")){

                      Serial.println("Responding GSM");

                      blinkLed ();

                      break;

            }

          else{

            Serial.print(" GSM Not Responding \n Check Wiring \n");

            blinkLedError ();

            response ="";

              }

        }

}

bool send_GSM_command  (String Command, String Response, long
timeOut , int atempt){

        char incomming;

        while (Serial2.available()){

            Serial.print( Serial2.read());

            }

        for (int i =0;i< atempt ; i++){

          Serial2.print(Command);

          smartDelay(timeOut);

          while(Serial2.available()){

                incomming = Serial2.read();

                Serial.write(incomming);

                Response += incomming;

                }
```

```
            if(0 <= Response.indexOf(Response)){

                        Serial.println("Responding to " + Command );

                        blinkLed ();

                        return true;

              }

              else{

                 Serial.println(" GSM Not Responding to" + Command );

                  blinkLedError ();

                  Response ="";

                }

           }

         return false;

}

void sendSMS1 (){                                    //To send SMS

           Serial.print("Sending massage");

         Serial2.print("AT+CMGF=1\r\n");

         smartDelay(3000);


Serial2.print("AT+CMGS=\"+919489546610\"\r\n");

                        smartDelay(3000);

Serial2.print("Bike         Security         Alert.Track         On
https://script.google.com/macros/s/AKfycbzEHxd246tagbX9pMQQEY7B0wAC
ApaNTKNxG6ndXvCa0gSSqTw/exec");

                 //   Serial2.print(" Bike location = \n");

                 //   Serial2.print(gps.location.lat(),6);

                 //   Serial2.print(", ");

                 //   Serial2.print(gps.location.lng(),6);

                      Serial2.write(0x1A);

                      Serial2.print("\r\n");

                      smartDelay(3000)

                          while (Serial2.available()){
```

```
                              Serial.write(Serial2.read());

                              }

   Serial.print("Sending ok")

     }

  void checkGSM() {

  check_AT_Response ();

  send_GSM_command  ("AT+CPIN?\r\n","AT+CPIN?\r\n\r\n+CPIN:  READY",
  2000,5 );

   if(send_GSM_command  ("AT+CREG?\r\n","AT+CREG?\r\n\r\n+CREG:  0,1",
2000,5 )){

                    }

  else if (send_GSM_command ("AT+CREG?\r\n","AT+CREG?\r\n\r\n+CREG:
  0,5", 2000,5 )){

                       }


  send_GSM_command("AT+SAPBR=3,1,\"Contype\",\"GPRS\"\r\n","AT+SAPBR=
  3,1,\"Contype\",\"GPRS\"\r\n\r\nOK", 2000,5 );

  send_GSM_command("AT+SAPBR=3,1,\"APN\",\"airtelgprs.com\"\r\n","AT+
  SAPBR=3,1,\"APN\",\"airtelgprs.com\"\r\n\r\nOK", 2000,5 );

  send_GSM_command ("AT+SAPBR=1,1\r\n","AT+SAPBR=1,1\r\n\r\nOK",
  2000,5 );

  send_GSM_command ("AT+SAPBR=2,1\r\n","AT+SAPBR=2,1\r\n\r\n+SAPBR:
  1,1", 2000,5 );

  }

  void blinkLed (){

    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is
  the voltage level)

    delay(200);                        // wait for a second

    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making
  the voltage LOW

    delay(200);

    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is
  the voltage level)

    delay(200);                        // wait for a second
```

```
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW

  delay(200);

}

void blinkLedError (){

  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)

  delay(3000);                       // wait for a second

  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW

}
```

3.2 Webpage Designing

(a) Code for updating the GPS data to the Google spread sheet

```
// GET request syntax:

// https://script.google.com/macros/s/<gscript id>/exec?header_here=data_here

//-----------------------------------------------

/**

* Function doGet: Parse received data from GET request,

  get and store data which is corresponding with header row in Google Spreadsheet

*/

function doGet(e) {

  Logger.log( JSON.stringify(e) );  // view parameters

  var result = 'Ok'; // assume success

  if (e.parameter == 'undefined') {

    result = 'No Parameters';

  }

  else {

    var sheet_id = '1mZqDgUnR8YF95XBty2dv9NZOybfa4nEzdDcngXIMI5w';
       // Spreadsheet ID
```

```
      var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
         // get Active sheet

      var newRow = sheet.getLastRow() + 1;


      var rowData = [];

      rowData[0] = new Date();
                     // Timestamp in column A



  for (var param in e.parameter) {

        Logger.log('In for loop, param=' + param);

        var value = stripQuotes(e.parameter[param]);

        Logger.log(param + ':' + e.parameter[param]);

        switch (param) {

          case 'lat': //Parameter

            rowData[1] = value; //Value in column B

            result += 'Written on column B';

            break;

          case 'lon': //Parameter

            rowData[2] = value; //Value in column C

            result += ' ,Written on column C';

            break;

          default:

            result = "unsupported parameter";

        }

      }

      Logger.log(JSON.stringify(rowData));

      // Write new row below

      var newRange = sheet.getRange(newRow, 1, 1, rowData.length);

      newRange.setValues([rowData]);

    }

    // Return result of operation
```

```
  return ContentService.createTextOutput(result);
}
/**
* Remove leading and trailing single or double quotes
*/
function stripQuotes( value ) {
  return value.replace(/^["']|['"]$/g, "");
}
//-----------------------------------------------
// End of file
//-----------------------------------------------
```

(b) Webpage Design code

(i) HTML code for the webpage

```
!DOCTYPE html>
<html>
  <head>
  <base target="_top">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

  <title>GPS Tracker... </title>

  <h1> Tracking and Security System for automobiles </h1>
 <button onclick="getLocation()">Track Vehicle</button>
 <button onclick="PauseTracking()">Pause Tracking</button>
 <button onclick="changeVehicle()">Change Travel Mode</button>

    <p id="demo"></p>
```

```
    <p id="demo1"></p>

    <p id="demo2"></p>




  </head>

  <body>


  <img src="https://auto.ndtvimg.com/bike-images/colors/yamaha/fz-s-v20-fi/yamaha-fz-s-v20-fi-lightning-cyan.png" alt="Trulli" width="450" height="333">


  <script type="text/javascript">

       var i = 0, len = frames.length, loc =10,myLat=20.02, myLon=79.9200, rld =20,vehicleType=0 ,bikePositionLat=22.9200, bikePositionLon=79.0000;


       function ChangeSrc()

              {

document.getElementById('frame').src="https://www.google.com/maps/embed?pb=!1m25!!!!!!!!!!!!!4m10!3e"+ vehicleType +"!4m4!2s" + myLat +"%2C+"
+myLon+"!!!!4m3!3m2!1d"+bikePositionLat+"!2d"+bikePositionLon+"!!!!!";

                              //document.getElementById('frame').src ="https://www.google.com/maps/embed?pb=!1m26!1m12!1m3!1d1892240.577366377!2d77.6117651528495!3d22.134271621163325!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!4m11!3e0!4m4!2s22.0000%2C78.000000!3m2!1d22!2d78!4m4!2s22.5%2C79.5!3m2!1d22.5!2d79.5!5e0!3m2!1sen!2sin!4v1543030007408" width="600" height="450" frameborder="0" style="border:0";

             setTimeout('ChangeSrc()', (rld*1000));

             setTimeout('ReadFromGoogleDrive()', (rld*1000));


              }

 </script>


 <script>
```

```
        var x = document.getElementById("demo");
        var x2 = document.getElementById("demo2");
         function getLocation()
            {
             ChangeSrc();
        if (navigator.geolocation) {
  navigator.geolocation.watchPosition(showPosition);
                        }
                    else {
  x.innerHTML = "Geolocation is not supported by this browser.";}
                        }
 function showPosition(position)
             {
                  myLat = position.coords.latitude;
                  myLon = position.coords.longitude;
x.innerHTML="My  Latitude:  "  +  position.coords.latitude  +"<br>  My
Longitude: " + position.coords.longitude;


             }
  </script>


  <script>
        function PauseTracking()
             {
             rld=60;
             }


  </script>


  <script>
```

```
        function changeVehicle()

             {

             if(vehicleType==2)

                   {vehicleType=0;}

                   else{vehicleType=2;}

             ChangeSrc();

             }


 </script>


 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.
js"></script>


 <script>

       function ReadFromGoogleDrive() {


         var xhttp = new XMLHttpRequest();

         xhttp.onreadystatechange = function() {

         if (this.readyState == 4 && this.status == 200) {

         var res=this.responseText;

         var result1=res.split(',');


         bikePositionLat =result1[1];

         bikePositionLon =result1[2];


document.getElementById('demo1').innerHTML = ("Bike Position Lat
="+result1[1] +"<br> Bike Position Lon ="+result1[2] )+"<br> <br>
Distance (Km):   "+((Math.sqrt((Math.pow((bikePositionLat -
myLat),2)+Math.pow((bikePositionLon - myLon),2))))*115);

    }

  };
```

```
  xhttp.open("GET", "https://script.google.com/macros/s/AKfycbygag__aKVU2WLj3GY3ea3jlJb31MUyj0iGUeA2HVgE5dWDPsN7/exec", true);

  xhttp.send();


                      }

 </script>


 <iframe src="" name="frame" id="frame" width="600" height="450" frameborder="0" style="border:0" allowfullscreen></iframe>

  </body>

</html>
```

(ii) Code for converting the html code to Google Spread sheet

```
function doGet() {
  return HtmlService.createHtmlOutputFromFile('as1');

}

function as1(){

  return "succes";
}
```

(c) The following code has been written to transfer the Google spread sheet readings to the html webpage

```
function doGet(e) {


 // Logger.log( JSON.stringify(e) );  // view parameters



  var result = 'Ok'; // assume success

  var sheet_id = '1mZqDgUnR8YF95XBty2dv9NZOybfa4nEzdDcngXIMI5w';      // Spreadsheet ID

  var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();      // get Active sheet


  var ss = SpreadsheetApp.getActiveSpreadsheet();
```

```
  var sheet1 = ss.getSheets()[0];



  var values11 = sheet1.getSheetValues(sheet.getLastRow(), 1,
sheet.getLastRow() , 3);

  result = values11;



  return ContentService.createTextOutput(result);

}


//function stripQuotes( value ) {

//  return value.replace(/^["']|['"]$/g, ""); //* Remove leading
and trailing single or double quotes

//}
```

CHAPTER 4

PCB PRINTING AND INTERGATION

4.1 PCB Designing

Once the circuit diagram is made and the ESP 32 microprocessor is programed it has to be realized into a printed circuit board. So we design a schematic diagram which could be used to print the PCB. This job has been done by using Eagle PCB design software where we can make the schematic diagram of the circuit and then convert it into a printed circuit layout. The PCB Schematic is shown in Fig 10.

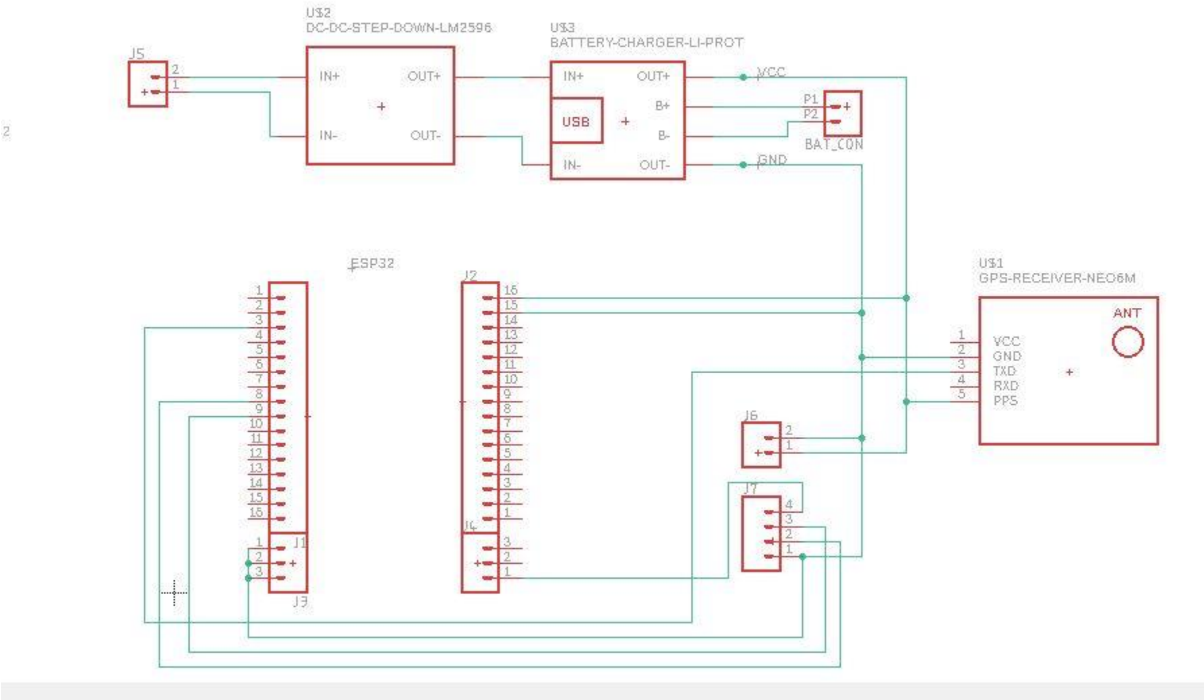

Fig 10 : PCB schematic diagram

The PCB layout designed in Eagle Software is shown in Fig 11.

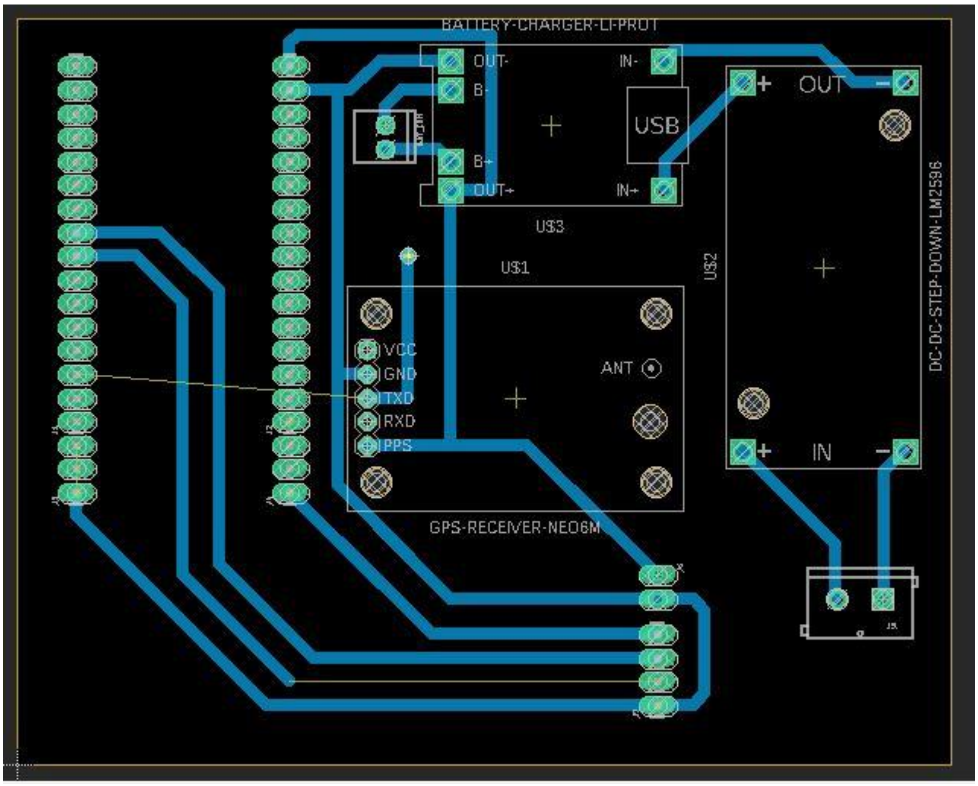

Fig 11 : Printed Circuit Layout

The software helps us to make the PCB more efficient changing the PCB parameters, like increasing the width of the copper tracks, as per the requirements of the PCB.

The printed circuit layout is then given to the PCB prototyping machine which gives us the final PCB.

CHAPTER 5

RESULTS AND DISCUSSIONS

The system was designed, developed and tested for the desired results.The final prototype designed as a PCB and implemented in the Two Wheeler is shown in Fig 12 and Fig 13. A screenshot of the Webpage designed is shown in Fig 14 and the screenshot of the tracking happening is shown in Fig 15.

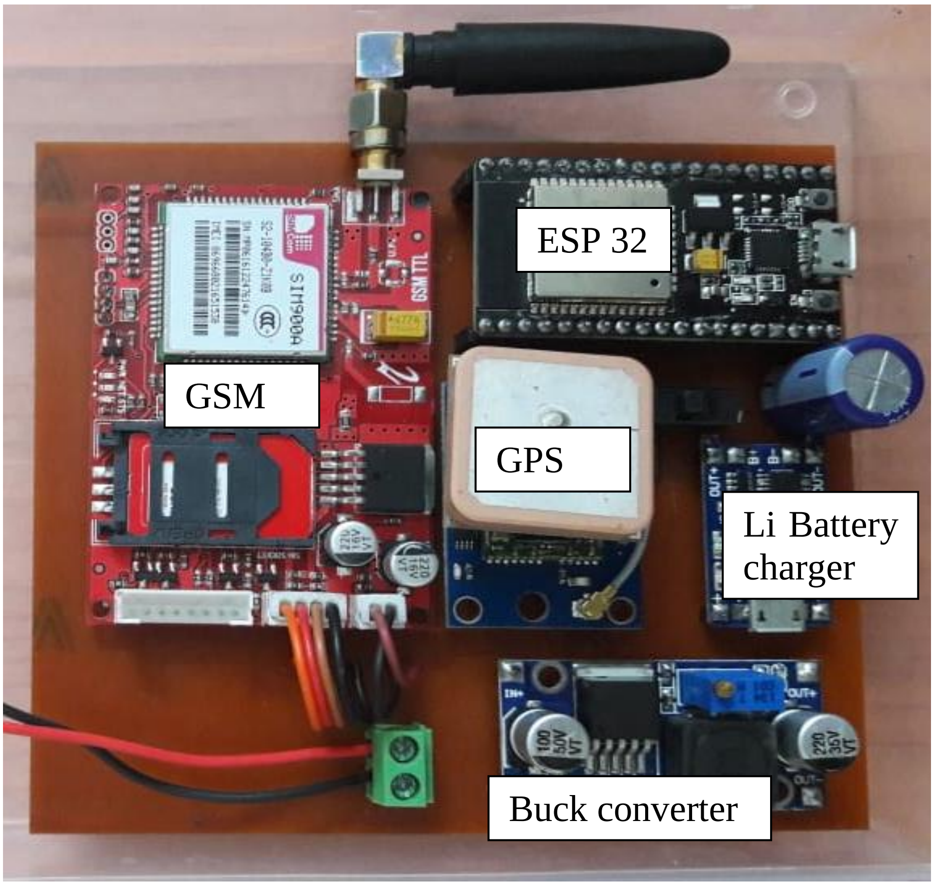

Fig 12 Two Wheeler Anti-Theft

Fig 13 System Implementation in the Two Wheeler

Fig 14 : Screenshot of the webpage

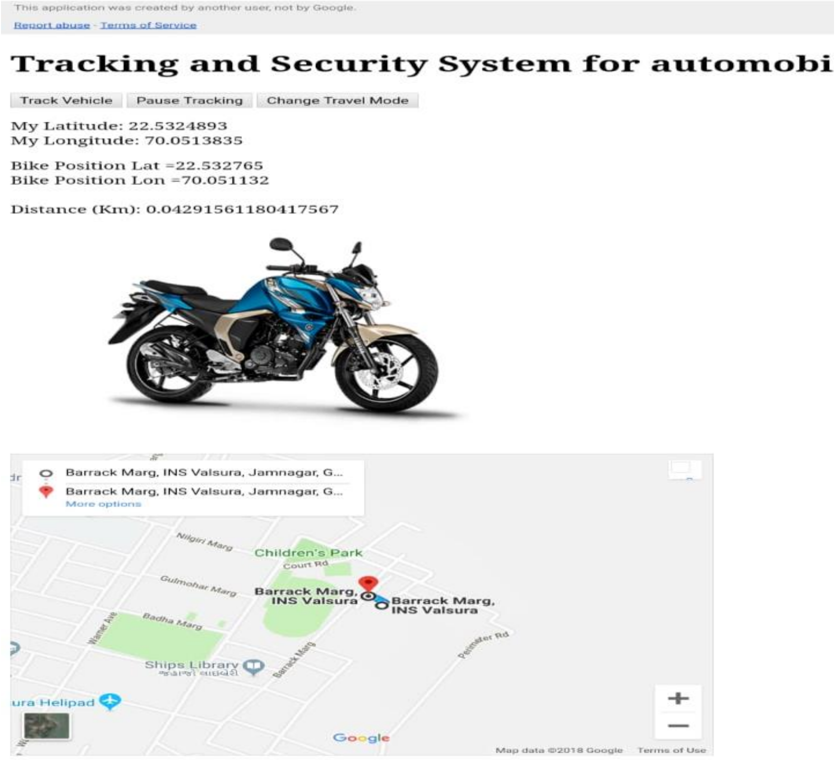

Fig 15 : Screenshot of tracking of the vehicle in Webpage

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

"Two wheeler Anti-Theft " is one of the most convenient way of knowing the location of one's vehicle. Thereby tracking and monitoring the vehicle at any time. Since it is economically feasible, user friendly and less complicated it can be used by everyone with a mobile phone or a laptop.

The prototype was developed with an aim of commercializing the the system as a product. The developed prototype may be further improvised by purchasing the Google map API so as to appreciate greater benefits with respect to the maps used in the Webpage. Many such product may be developed and fitted on various vehicles and can be simultaneously tracked on a single webpage or multiple webpages can also be developed.

CHAPTER 7

BIBLIOGRAPHY

1. Karan Siyal and G Gugapriya, " Anti-theft vehicle locking system using CAN", IJST, Dec 2016.

2. D Narender Singh and K Tejaswi, "Real time vehicle theft identity and control system based on ARM 9", IJLTET, Jan 2013

3. K Sruthi et. al " Anti-theft tracking system and security system for automobiles using GSM and ARM", IJEDR, 2016

4. Shamugnathan J and B C Kavitha, "Tracking and theft prevention system for two wheeler using android", IJETT, March 2015

5. Shweta K Narkhede et.al, "Two wheeler security system", IRJET, May 2017

6. www.w3schools.com,

7. Manual of GSM SIM 900A

8. Manual of U-BLOX NEO 6M GPS

9. Manual of ESP 32 microcontroller