Live Location Tracking of objects using GPS

By
Retta Sivaji
on the completion of Summer Research Internship
Under the supervision of Prof.Santos Kumar Das,
NIT ROURKLEA
MAY 2019-JUNE 2019
Application Number: SIP190410

***Abstract*- Now -a-days ,it is important to have a multidisciplinary Intelligent System(MIS) in the defense applications. Wireless sensor networks (WSNs) find extensive applications in the modern day world starting from the environment monitoring to the tracking of vehicles and security. We can detect the position of where the abnormality or accidents in case of vehicles has been occurred and the proposed GPS based tracking system can be installed in vehicles or any other objects and can be continuously tracked by the authorized persons .**

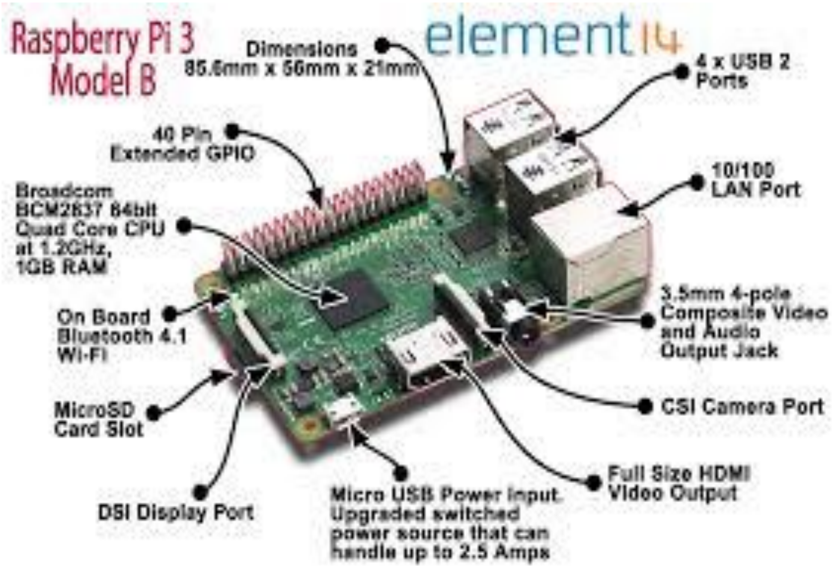
I. Introduction

Eventhough most location tracking devices are available in the market but all those are developed using SIM to notify specified persons using some android applications. But in our system , some authorized persons can login and track the specified vehicles or objects .We can use it for tracking of a fleet of vehicles or tracking of soldiers or any vehicles in the defense applications. The system is built with Raspberry Pi 3 and a Gtop GPS and GNSS module .

Introduction about the Hardware and Software parts of the Demo

Raspberry pi is a small and affordable computer. By connecting peripherals like keyboard ,mouse,display to the Raspberry Pi, it will act as a mini personal computer .Raspberry pi is popularly used for real time Image/video processing ,IOT based applications and Robotics applications.

Specifications of Raspberry Pi 3 B:



Gtop GPS module

- 33 tracking/ 99 acquisition-channel GPS/GLONASS receiver
- Supports QZSS, SBAS(WAAS, EGNOS, MSAS, GAGAN*) ranging
- Ultra-High Sensitivity: -165dBm
- High Update Rate: up to 10Hz(Note1)
- 12 multi-tone active interference canceller(Note2) [ISSCC 2011 Award -Section 26.5] (http://isscc.org/doc/2011/isscc2011.advanceprogrambooklet_abstracts.pdf)
- High accuracy 1-PPS timing support for Timing Applications (±10ns RMS jitter)
- AGPS Support for Fast TTFF (EPO in flash™ Enable 7 days/14 days)
- EASY™(Note2): Self-Generated Orbit Prediction for instant positioning fix
- AlwaysLocate™(Note2) Intelligent Algorithm (Advance Power Periodic Mode) for power saving
- LOCUS (Embedded Logger Function) (Note3)
- Gtop Firmware Customization Services
- GPS+GLONASS Consumption current(@3.3V):
  - Acquisition for GPS+GLONASS: 34mA Typical
  - Tracking for GPS+GLONASS: 29mA Typical
- RoHS compliant
- CE, FCC Certification

Table below lists each of the NMEA output sentences specifically developed and defined by MTK for use within MTK products.

Table-1: NMEA Output Sentence

Option	Description
GGA	Time, position and fix type data.
GSA	GNSS receiver operating mode, active satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
RMC	Time, date, position, course and speed data. The recommended minimum navigation information.
VTG	Course and speed information relative to the ground.

Among all these we need GPGGA string to find the location using latitude and longitude values at certain time.

Example to Understand the GPGGA String

$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65

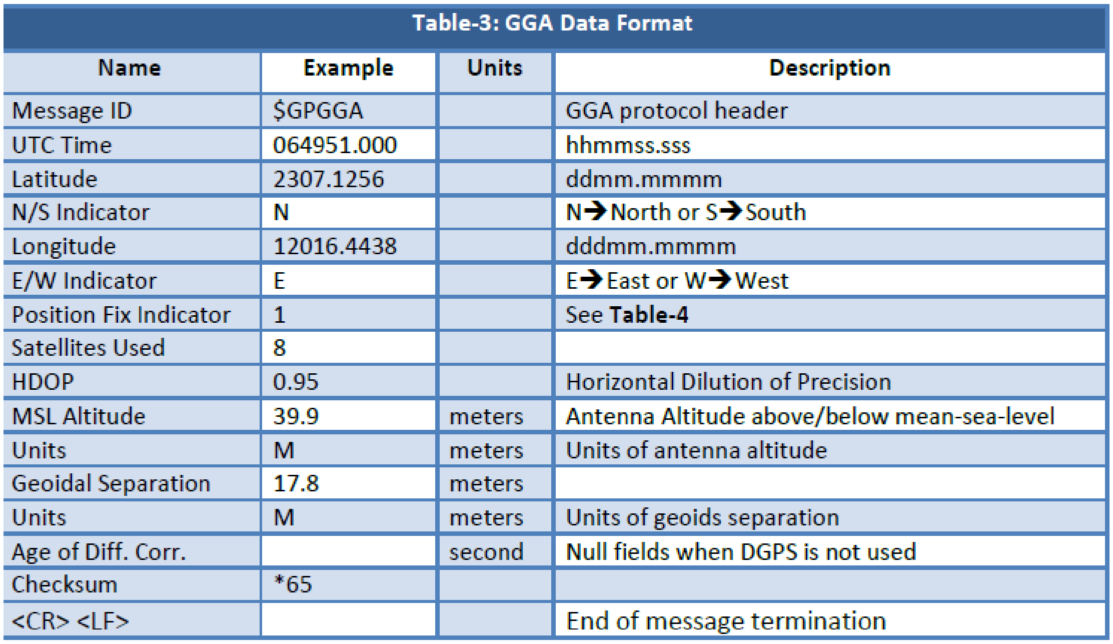
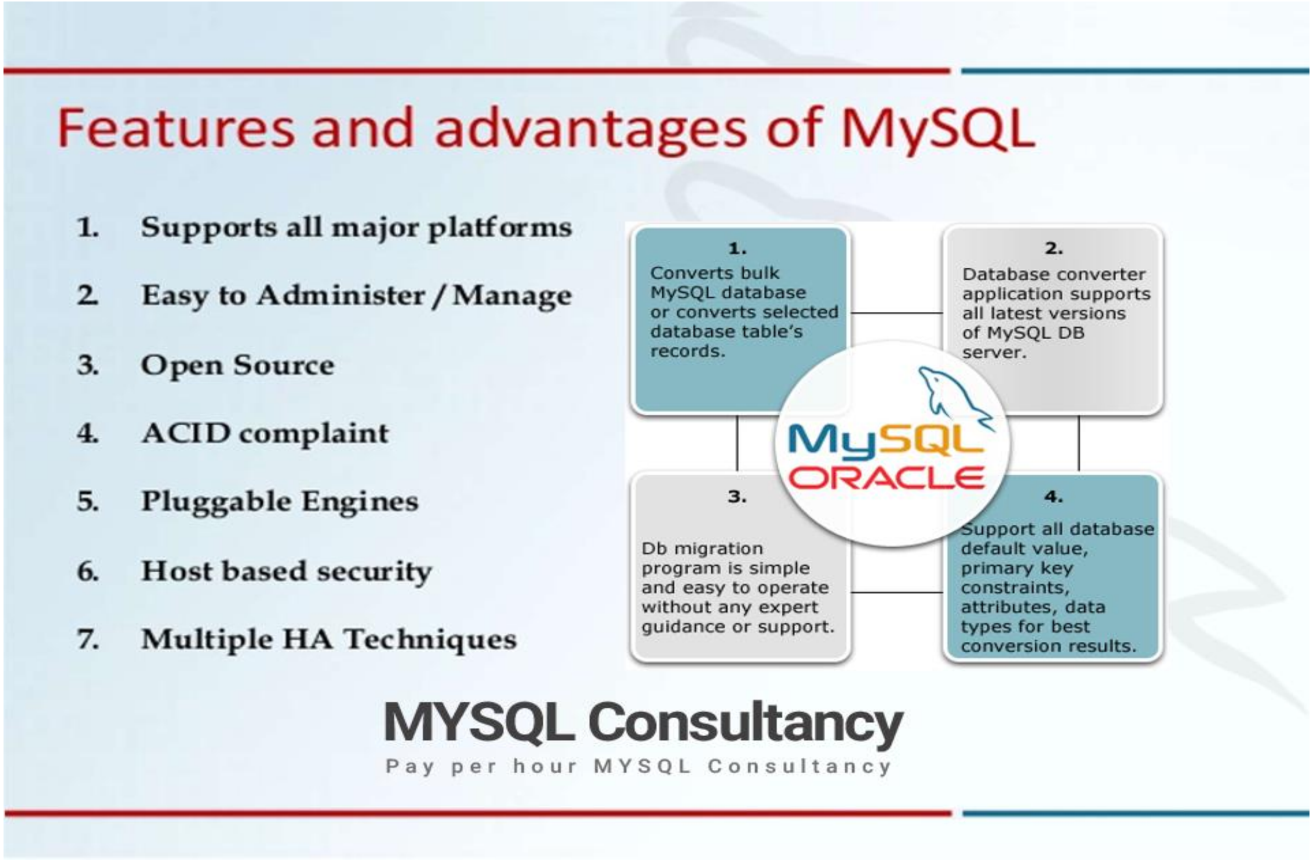


Table-3: GGA Data Format

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	064951.000		hhmmss.sss
Latitude	2307.1256		ddmm.mmmm
N/S Indicator	N		N→North or S→South
Longitude	12016.4438		dddmm.mmmm
E/W Indicator	E		E→East or W→West
Position Fix Indicator	1		See **Table-4**
Satellites Used	8		
HDOP	0.95		Horizontal Dilution of Precision
MSL Altitude	39.9	meters	Antenna Altitude above/below mean-sea-level
Units	M	meters	Units of antenna altitude
Geoidal Separation	17.8	meters	
Units	M	meters	Units of geoids separation
Age of Diff. Corr.		second	Null fields when DGPS is not used
Checksum	*65		
<CR> <LF>			End of message termination

MySQL Database:



Features and advantages of MySQL

1. Supports all major platforms
2. Easy to Administer / Manage
3. Open Source
4. ACID complaint
5. Pluggable Engines
6. Host based security
7. Multiple HA Techniques

1. Converts bulk MySQL database or converts selected database table's records.

2. Database converter application supports all latest versions of MySQL DB server.

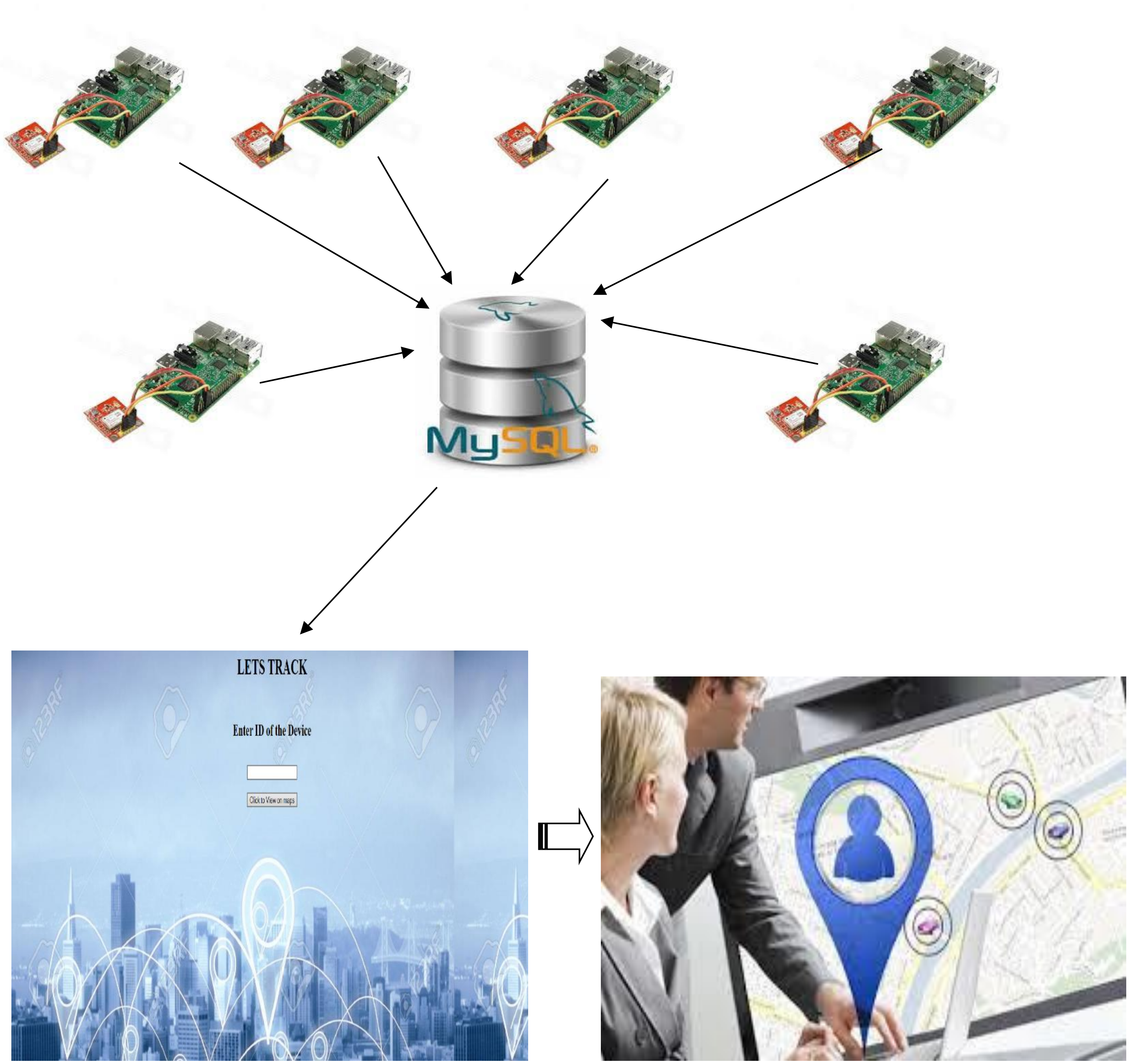
3. Db migration program is simple and easy to operate without any expert guidance or support.

4. Support all database default value, primary key constraints, attributes, data types for best conversion results.

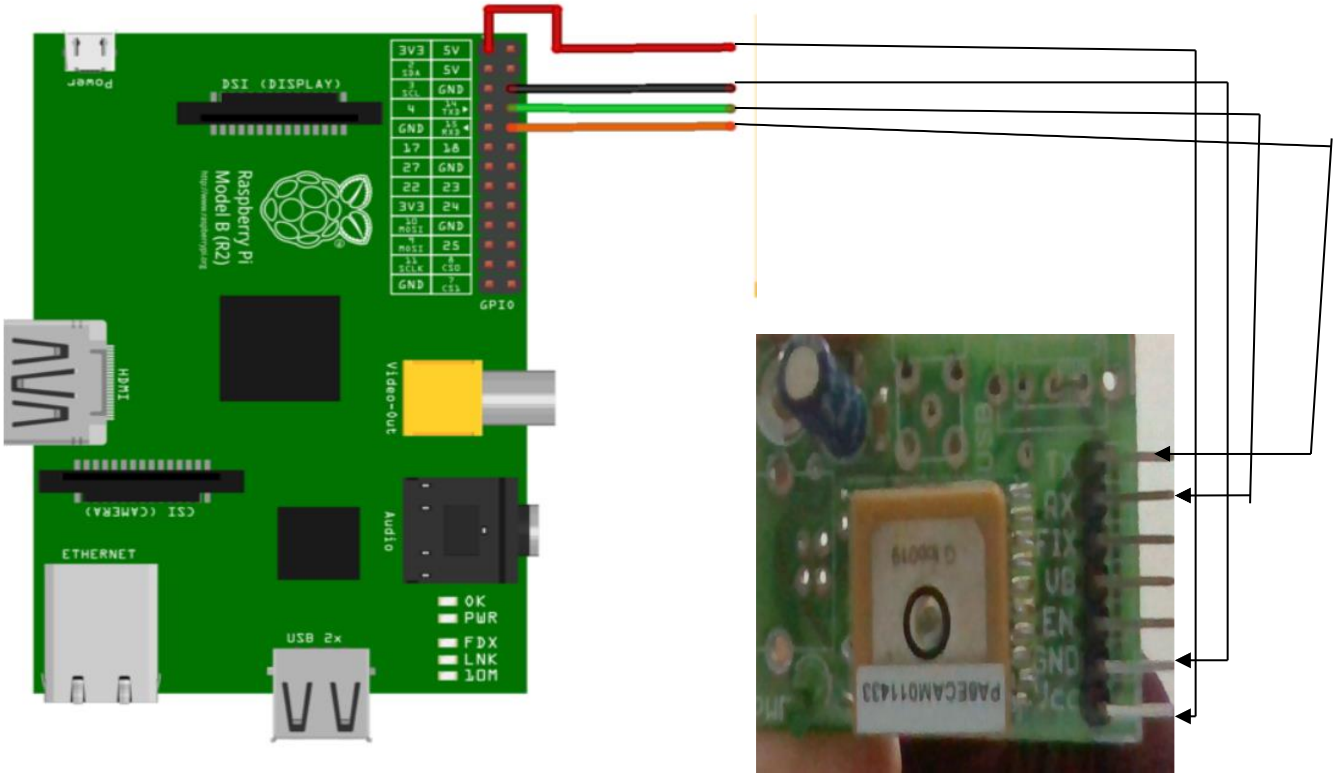
MYSQL Consultancy

Pay per hour MYSQL Consultancy

II. SYSTEM METHODOLOGY



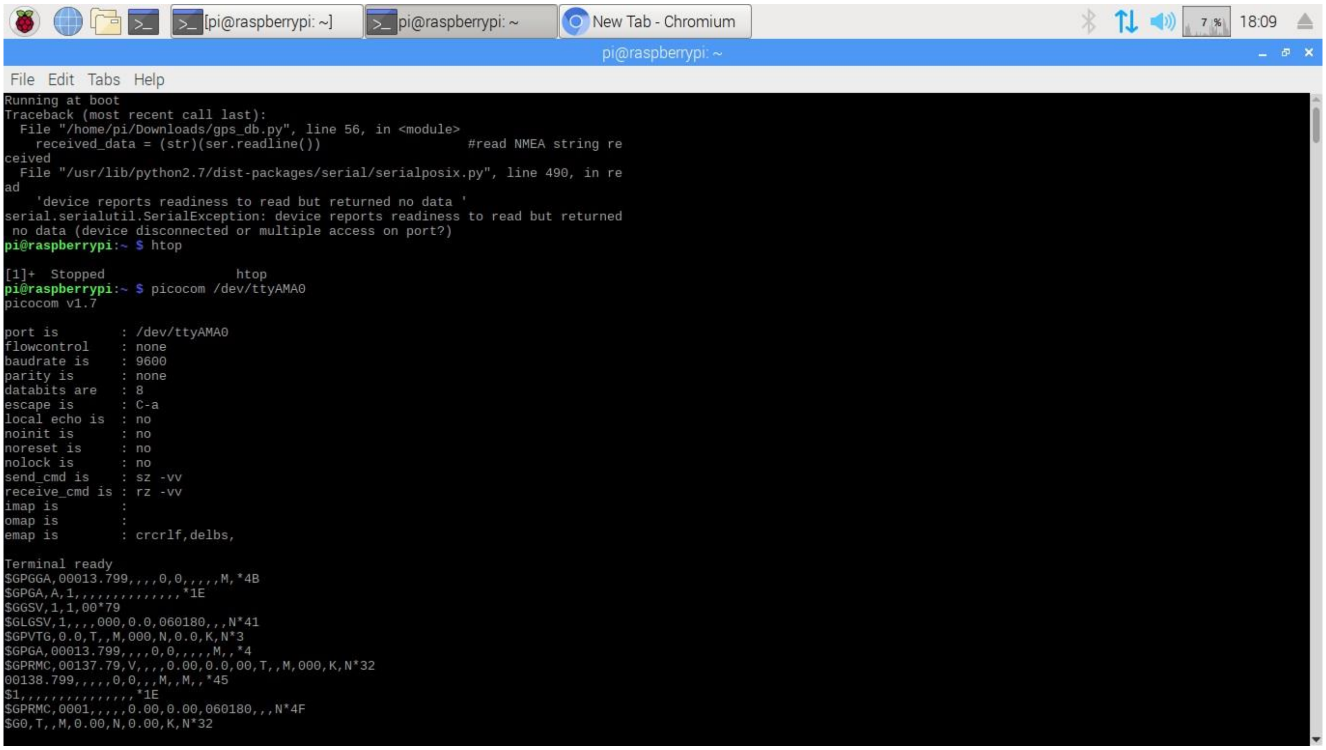
III. Interfacing Circuit



GPS	RASPBERRY PI
VCC	3.3 V
GND	GND
TX	RX
RX	TX

Implementation

1.Enable the serial port and install the gps libraries on the pi using gpsd clients command.
2.To check whether GPS module is working or not ,
i. Install picocom using command " pip install picocom"
ii.Type the command sudo picocom /dev/ttyAMAO or sudo picocom /dev/ttyS0 based on the port using.



```
Running at boot
Traceback (most recent call last):
  File "/home/pi/Downloads/gps_db.py", line 56, in <module>
    received_data = (str)(ser.readline())                   #read NMEA string re
ceived
  File "/usr/lib/python2.7/dist-packages/serial/serialposix.py", line 490, in re
ad
    'device reports readiness to read but returned no data '
serial.serialutil.SerialException: device reports readiness to read but returned
 no data (device disconnected or multiple access on port?)
pi@raspberrypi:~ $ htop

[1]+  Stopped                 htop
pi@raspberrypi:~ $ picocom /dev/ttyAMA0
picocom v1.7

port is        : /dev/ttyAMA0
flowcontrol    : none
baudrate is    : 9600
parity is      : none
databits are   : 8
escape is      : C-a
local echo is  : no
noinit is      : no
noreset is     : no
nolock is      : no
send_cmd is    : sz -vv
receive_cmd is : rz -vv
imap is        :
omap is        :
emap is        : crcrlf,delbs,

Terminal ready
$GPGGA,00013.799,,,,0,0,,,,,M,*4B
$GPGA,A,1,,,,,,,,,,,,,,*1E
$GGSV,1,1,00*79
$GLGSV,1,,,,000,0.0,060180,,,N*41
$GPVTG,0.0,T,,M,000,N,0.0,K,N*3
$GPGA,00013.799,,,,0,0,,,,,M,,*4
$GPRMC,00137.79,V,,,,0.00,0.0,00,T,,M,000,K,N*32
00138.799,,,,,0,0,,,M,,M,,*45
$1,,,,,,,,,,,,,,,,*1E
$GPRMC,0001,,,,,0.00,0.00,060180,,,N*4F
$G0,T,,M,0.00,N,0.00,K,N*32
```

3.Install the Mysqldb and Apache server and related libraries on your Raspberry Pi.

Create a Database using command “CREATE DATABASE GPSDB”
Then create the table gps with fiels datetime ,latitude and longitude.
To show the latest updated position in the google maps,
ALTER TABLE gps ADD sno INT PRIMARY KEY AUTO_INCREMENT

CODE:

The following code will fetch the latitude and longitude values and sent it to the database named GPSDB

```
#!/usr/bin/env python
import serial             #import serial pacakge
from time import sleep
import webbrowser         #import package for opening link in browser
import sys               #import system package
import  MySQLdb

db = MySQLdb.connect(host="localhost", user="root",passwd="sivaji17205", db="gps")
cur = db.cursor()
def GPS_Info():
    global NMEA_buff
    global lat_in_degrees
    global long_in_degrees
    nmea_time = []
    nmea_latitude = []
    nmea_longitude = []
    nmea_time = NMEA_buff[0]              #extract time from GPGGA string
    nmea_latitude = NMEA_buff[1]          #extract latitude from GPGGA string
    nmea_longitude = NMEA_buff[3]         #extract longitude from GPGGA string
    print("NMEA Time: ", nmea_time,'\n')
    print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,'\n')

    lat = float(nmea_latitude)              #convert string into float for calculation
    longi =float(nmea_longitude)            #convertr string into float for calculation

    lat_in_degrees = convert_to_degrees(lat)    #get latitude in degree decimal format
    long_in_degrees = convert_to_degrees(longi) #get longitude in degree decimal format

#convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
    decimal_value = raw_value/100.00
    degrees = int(decimal_value)
    mm_mmmm = (decimal_value - int(decimal_value))/0.6
    position = degrees + mm_mmmm
    position = "%.4f" %(position)
    return position



gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyAMA0")    #Open port with baud rate
GPGGA_buffer = 0
NMEA_buff = 0
lat_in_degrees = 0
long_in_degrees = 0

try:
    while True:
        received_data = (str)(ser.readline())                   #read NMEA string received
        GPGGA_data_available = received_data.find(gpgga_info)   #check for NMEA GPGGA string
        if (GPGGA_data_available>0):
            GPGGA_buffer = received_data.split("$GPGGA,",1)[1]  #store data coming after "$GPGGA," string
            NMEA_buff = (GPGGA_buffer.split(','))               #store comma separated data in buffer
            GPS_Info()                                          #get time, latitude, longitude
            print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')
            sql = ("""INSERT INTO gpsdb(datetime,lat,longi) VALUES
(%s,%s,%s)""",(nmea_time,lat_in_degrees,long_in_degrees))
            try:
                print ("Writing to database...")
                # Execute the SQL command
                cur.execute(*sql)
                # Commit your changes in the database
                db.commit()
                print ("Write Complete")

            except:
                # Rollback in case there is any error
                db.rollback()
                print ("Failed writing to database")

            cur.close()
            db.close()
except KeyboardInterrupt:
     webbrowser.open(map_link)                          #open current position information in google map
     sys.exit(0)
```

The following code will Show the latest position of the device in google maps

```
<?php
 $db = mysqli_connect('localhost','root','isdr@430','GPSDB')
 or die('Error connecting to MySQL server.');
?>
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style>
      /* Always set the map height explicitly to define the size of the div
       * element that contains the map. */
      #map {
        height: 100%;
      }
      /* Optional: Makes the sample page fill the window. */
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
    </style>
  </head>

  <body>
    <div id="map"></div>

      <?php
//Step2
$query = "SELECT * FROM  gps order by sno desc limit 1 ";
mysqli_query($db, $query) or die('Error querying database.');

//Step3
$result = mysqli_query($db, $query);
$row = mysqli_fetch_array($result);

echo $row['datetime'] . ' ' . $row['latitude'] . ': ' . $row['longitude'] . ' '  .'<br />';

?>

    <script>
      var map;
      function initMap(){
          var myLatlng = {lat: <?php echo $row['latitude']?>, lng: <?php echo $row['longitude']?>};
          console.log(myLatlng);
          map = new google.maps.Map(document.getElementById('map'), {
          center: {lat:  <?php echo $row['latitude']?>, lng: <?php echo $row['longitude']?> },
          zoom: 20
        });
           var marker = new google.maps.Marker({
           position: myLatlng,
           map: map,
           title: 'Click to zoom'
         });
      }
    </script>
```

```
     <?php
// mysqli_close($db);
?>

   <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD_4DEH5XzKKS36yrosKOJGHqewoTTvrmQ&callback=initMap
"
   async defer></script>
  <script language ="javascript">
   setTimeout(function(){
   window.location.reload(1);
   },10000);
  </script>
 </body>
</html>
```

This page will automatically reloaded for every 10 sec as depicted in the setTimeout function.If you want to change it,you can.

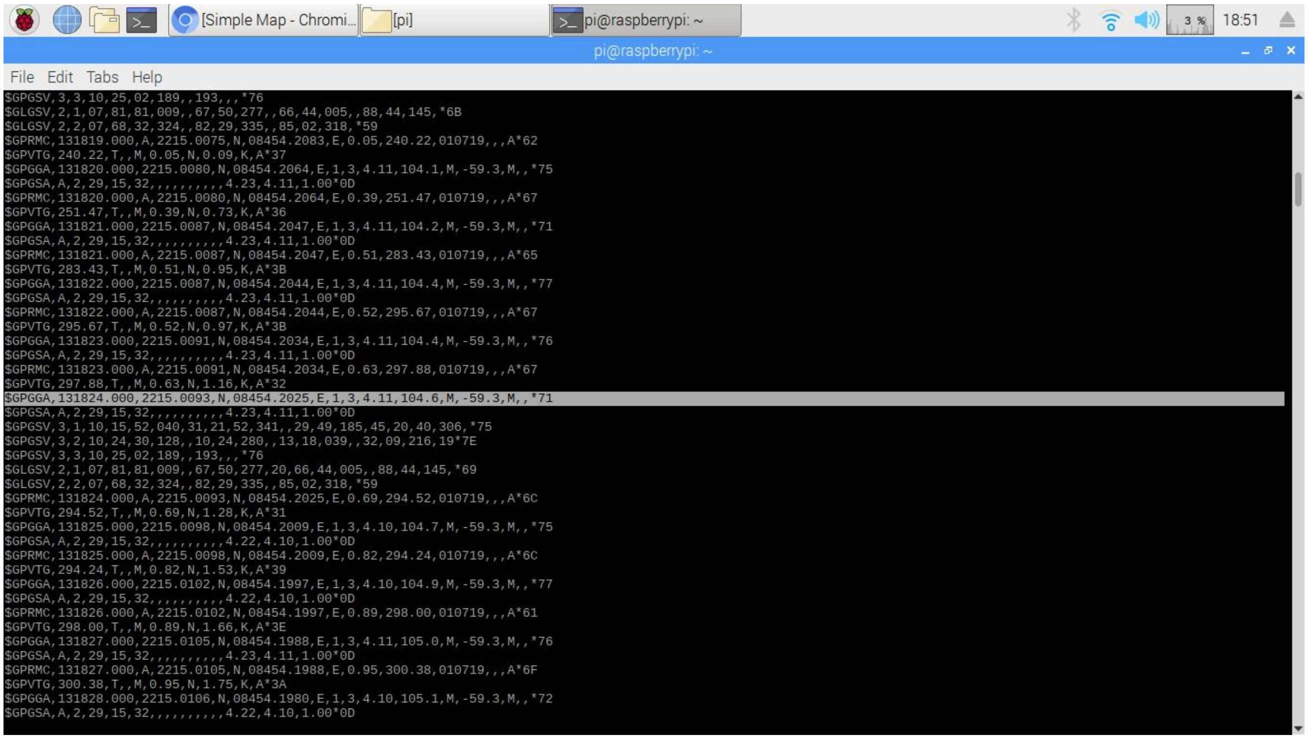
Running the program at startup on RaspberryPi:

```
1. sudo nano /home/pi/.bashrc
```

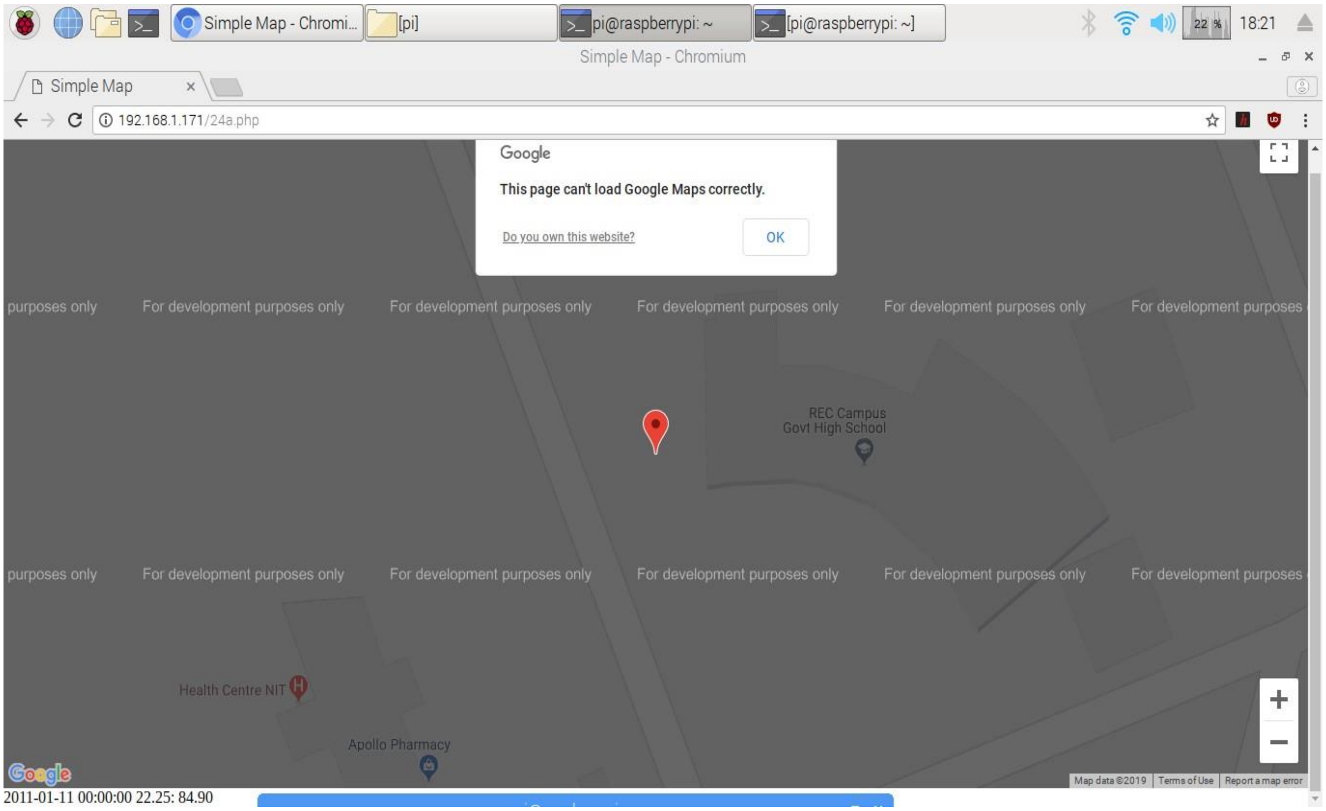
2.Go to the last line and paste the following by replacing the sample.py wioth your program name

```
echo Running at boot

sudo python /home/pi/sample.py

3.Then reboot using sudo reboot command
```

WORKING MODEL AND RESULTS



```
$GPGSV,3,3,10,25,02,189,,193,,,*76
$GLGSV,2,1,07,81,81,009,,67,50,277,,66,44,005,,88,44,145,*6B
$GLGSV,2,2,07,68,32,324,,82,29,335,,85,02,318,*59
$GPRMC,131819.000,A,2215.0075,N,08454.2083,E,0.05,240.22,010719,,,A*62
$GPVTG,240.22,T,,M,0.05,N,0.09,K,A*37
$GPGGA,131820.000,2215.0080,N,08454.2064,E,1,3,4.11,104.1,M,-59.3,M,,*75
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPRMC,131820.000,A,2215.0080,N,08454.2064,E,0.39,251.47,010719,,,A*67
$GPVTG,251.47,T,,M,0.39,N,0.73,K,A*36
$GPGGA,131821.000,2215.0087,N,08454.2047,E,1,3,4.11,104.2,M,-59.3,M,,*71
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPRMC,131821.000,A,2215.0087,N,08454.2047,E,0.51,283.43,010719,,,A*65
$GPVTG,283.43,T,,M,0.51,N,0.95,K,A*3B
$GPGGA,131822.000,2215.0087,N,08454.2044,E,1,3,4.11,104.4,M,-59.3,M,,*77
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPRMC,131822.000,A,2215.0087,N,08454.2044,E,0.52,295.67,010719,,,A*67
$GPVTG,295.67,T,,M,0.52,N,0.97,K,A*3B
$GPGGA,131823.000,2215.0091,N,08454.2034,E,1,3,4.11,104.4,M,-59.3,M,,*76
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPRMC,131823.000,A,2215.0091,N,08454.2034,E,0.63,297.88,010719,,,A*67
$GPVTG,297.88,T,,M,0.63,N,1.16,K,A*32
$GPGGA,131824.000,2215.0093,N,08454.2025,E,1,3,4.11,104.6,M,-59.3,M,,*71
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPGSV,3,1,10,15,52,040,31,21,52,341,,29,49,185,45,20,40,306,*75
$GPGSV,3,2,10,24,30,128,,10,24,280,,13,18,039,,32,09,216,19*7E
$GPGSV,3,3,10,25,02,189,,193,,,*76
$GLGSV,2,1,07,81,81,009,,67,50,277,20,66,44,005,,88,44,145,*69
$GLGSV,2,2,07,68,32,324,,82,29,335,,85,02,318,*59
$GPRMC,131824.000,A,2215.0093,N,08454.2025,E,0.69,294.52,010719,,,A*6C
$GPVTG,294.52,T,,M,0.69,N,1.28,K,A*31
$GPGGA,131825.000,2215.0098,N,08454.2009,E,1,3,4.10,104.7,M,-59.3,M,,*75
$GPGSA,A,2,29,15,32,,,,,,,,,,4.22,4.10,1.00*0D
$GPRMC,131825.000,A,2215.0098,N,08454.2009,E,0.82,294.24,010719,,,A*6C
$GPVTG,294.24,T,,M,0.82,N,1.53,K,A*39
$GPGGA,131826.000,2215.0102,N,08454.1997,E,1,3,4.10,104.9,M,-59.3,M,,*77
$GPGSA,A,2,29,15,32,,,,,,,,,,4.22,4.10,1.00*0D
$GPRMC,131826.000,A,2215.0102,N,08454.1997,E,0.89,298.00,010719,,,A*61
$GPVTG,298.00,T,,M,0.89,N,1.66,K,A*3E
$GPGGA,131827.000,2215.0105,N,08454.1988,E,1,3,4.11,105.0,M,-59.3,M,,*76
$GPGSA,A,2,29,15,32,,,,,,,,,,4.23,4.11,1.00*0D
$GPRMC,131827.000,A,2215.0105,N,08454.1988,E,0.95,300.38,010719,,,A*6F
$GPVTG,300.38,T,,M,0.95,N,1.75,K,A*3A
$GPGGA,131828.000,2215.0106,N,08454.1980,E,1,3,4.10,105.1,M,-59.3,M,,*72
$GPGSA,A,2,29,15,32,,,,,,,,,,4.22,4.10,1.00*0D
```



This page can't load Google Maps correctly.

2011-01-11 00:00:00 22.25: 84.90

CONCLUSION

Location tracking plays a major role in smart Vehicles and Defense Services.Our Model can be easily installed in vehicles and or any other systems to track by the authorized persons like owner of a fleet of vehicles or it can be worn by soldiers to track them by the specified personalities.It can also be implemented using NodeMcu to make it portable to wear.It can be affordable.

REFERENCES:

https://www.raspberrypi.org/
https://sunrobotics.co.in/download/globaltop-pa6e-cam-datasheet-v0b.pdf
https://www.instructables.com/id/Raspberry-Pi-the-Neo-6M-GPS/
https://circuitdigest.com/microcontroller-projects/raspberry-pi-3-gps-module-interfacing
https://www.electronicwings.com/raspberry-pi/gps-module-interfacing-with-raspberry-pi
https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/
https://console.cloud.google.com/google/maps-apis/overview
https://www.mysql.com/
https://www.w3schools.com/sql/sql_ref_mysql.asp
https://www.w3schools.com/php/
https://developers.google.com/maps/documentation/javascript/tutorial
https://www.dataplicity.com/
https://wingoodharry.wordpress.com/2015/01/05/raspberry-pi-temperature-sensor-web-server-part-2-setting-up-and-writing-to-a-mysql-database/