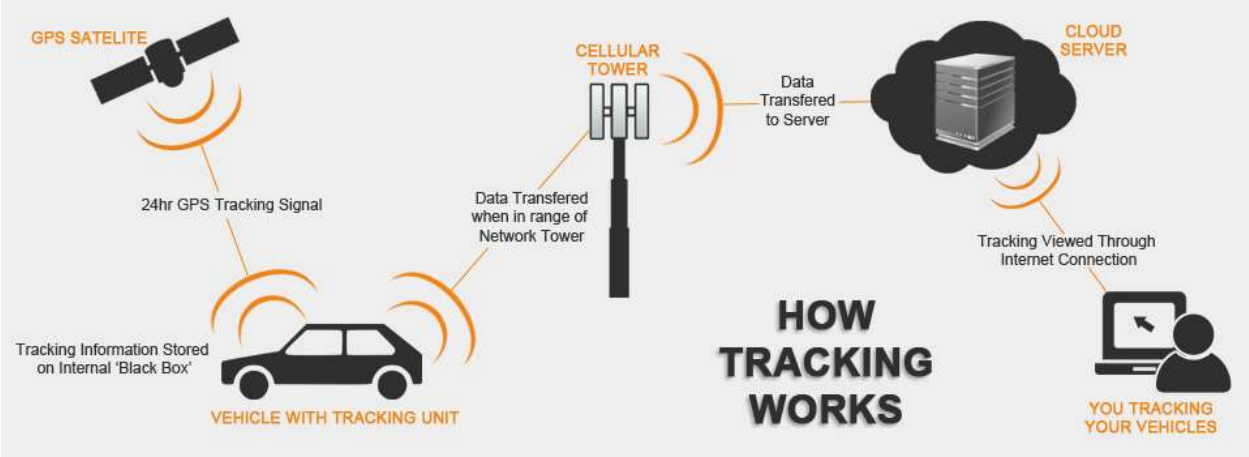
VEHICLE TRACKING SYSTEM

Summer Internship



Submitted by:

U.Tejas Kini - 160115735095

Deepak Raya Velgapuni - 160115735091

Aishwarya Kasim - 160115735062

ACKNOWLEDGEMENTS

Firstly, we would like to thank Dr.N.V. Koteswar Rao, Professor and Head of the Department of Electronics and Communication for giving us this opportunity to explore our innate abilities. We would also thank “Mytrah Energy (India) Limited”, which gave us the chance to prove ourselves and help us apply the knowledge we have and also, for providing an outstanding environment to carry out our Project.

We would like to convey our sincere thanks to:

- Mr. Kiran Nair, DGM, Head,WRD
- Mr. Prasad Gaitonde, Sr.Manager,WRD
- Mr. Harry Justin, Dy.Manager,WRD
- Mr. Jos Angel George, Team Lead, WRD
- Mr. Paul Joe Mampilly
- Ms. Jasmine U Varikkathu

for inspiring and guiding us. Only with their constant assistance have we achieved so much. Finally, we thank Mr.U.V.Kini for bridging the gap between the company and us and, with whose support our dreams were realizable.

ABSTRACT

Working of any system in an industry involves various aspects pertaining to it. Efficiency, viability, cost-effectiveness, age and sustainability are some among them. For a tracking system, the main aspect is its efficiency. For example, when we take a look at the present companies which depend solely and by large on GPS tracking are definitely Ola and Uber. Everyday thousands of people book cabs which are done by using GPS system. With that being said, there are also high chances of errors to occur. To ensure that it does not happen on such a large scale is an arduous task in itself. Hence, there is a need for constant maintenance and analysis of the system. And, for a tracking system, keeping in mind the present day technology which is used in it, also requires constant and consistent monitoring of it to ensure high efficiency.

The Project ‘Vehicle Tracking Module for Wind Mast Tracking’ main purpose is to find a thread to bind the present technology involving Vehicle Tracking and the need to overcome the problems that may crop up in the process.

TABLE OF CONTENTS

DESCRIPTION OF THE PROJECT

Aim of the Project:

To create a commercially viable and cost effective Vehicle Tracking System.

Overview of a Tracking System:

In easier terms, a system which tracks the location of an object is called a Tracking System. That being said, the device encompasses several stages which forms the base of this Project. In the later sections, the technical requirements of the Project and the methods used in creating such a device are elaborated.

Necessity of the Project:

With the advancement of technology, mankind has taken a quantum leap. Companies have grown boundless and so have the threats. For an established company such as Mytrah Energy (India) Limited, the procurement of raw materials and goods from different places has always been a need. The problem lies there. Whilst the goods are being transported from a place to the industrial site, the security of theirs is of our main concern. Hence, the solution to this problem is a Tracking Device. When attached to a good to be transported, this device helps us to locate the position of the vehicle and also, that of the good.

INTRODUCTION

In the field of communications, there are mainly two parts namely: Transmitting section and Receiving section. Thus, in our Project the present-day GPS system serves as a transmitting unit. The GPS-GSM module along with the Arduino-Uno act as a medium of transmitting data to the receiving unit which is: The Server.

The basic idea is to know the location of the vehicle and track its movement and store its history of travel in a database. The main reason of doing so,as already stated, is security of the goods being transported and delivered. Since the company uses third party vehicles in obtaining the goods, this system acts as a safe bet to monitor the process. Viability and cost-effectiveness are given the most importance in this Project.

METHODOLOGY

Hardware Components used:

- Arduino Uno r3 (ATMEGA 328 Microcontroller)
- SIMCOM SIM808 (GPS/GPRS/GSM MODULE)
- GPS receiver antenna

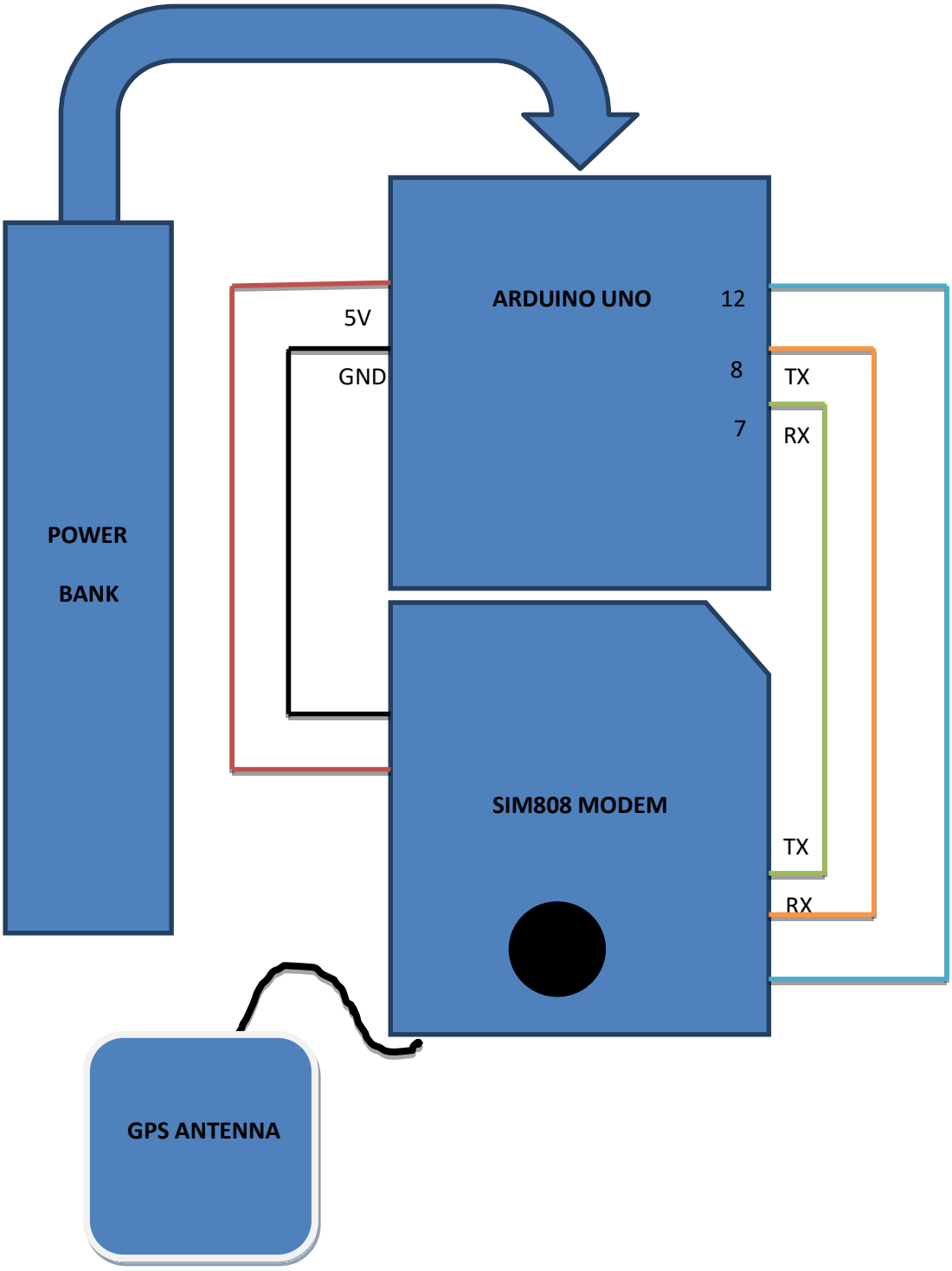
Software used:

- JavaScripts, HTML, CSS
- Node.js
- Google Maps API
- Mongo DB

Power requirements and specifications:

- Ambrane 26800mah power bank
  - Battery Type: Li-ion
  - Number of ports: 2
  - Output Power: 5V/2A
  - Warranty: One year
  - Weight: 448 grams

Block Diagram of the Module:

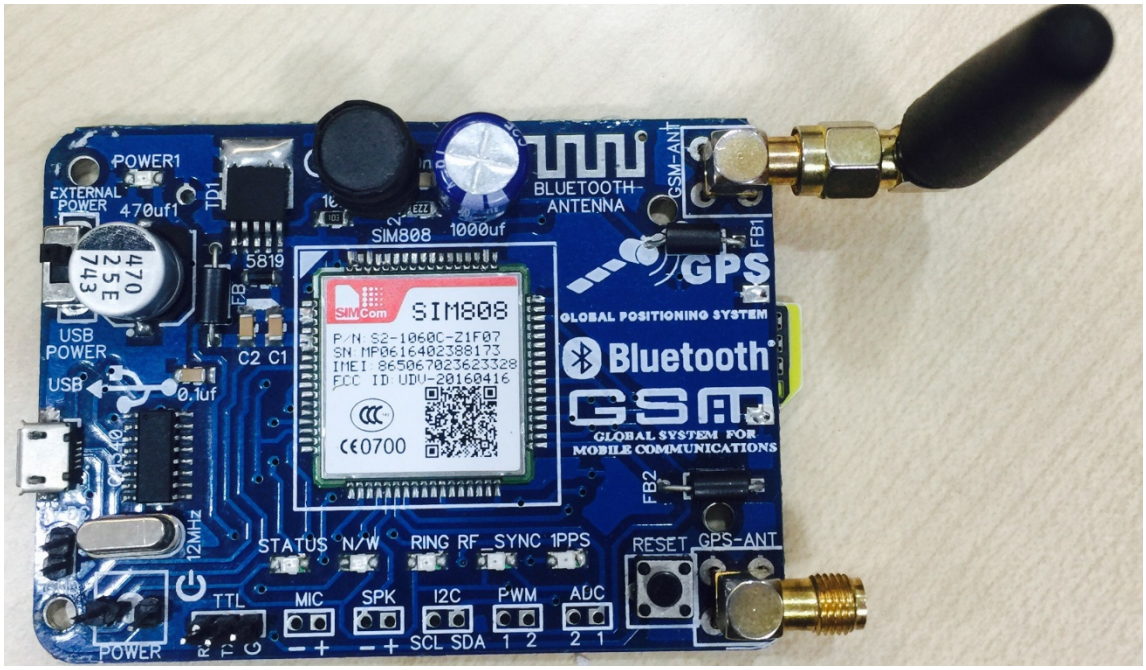


Arduino Uno board:

- The pins (digital 7 and 8 with reference to the Project) on the upper right side of the board are used.
- Battery is connected to the port as seen on the upper left side.
- The power to the GPS/GSM module is given using the power section to the pins +5V and is grounded using the GND pin.

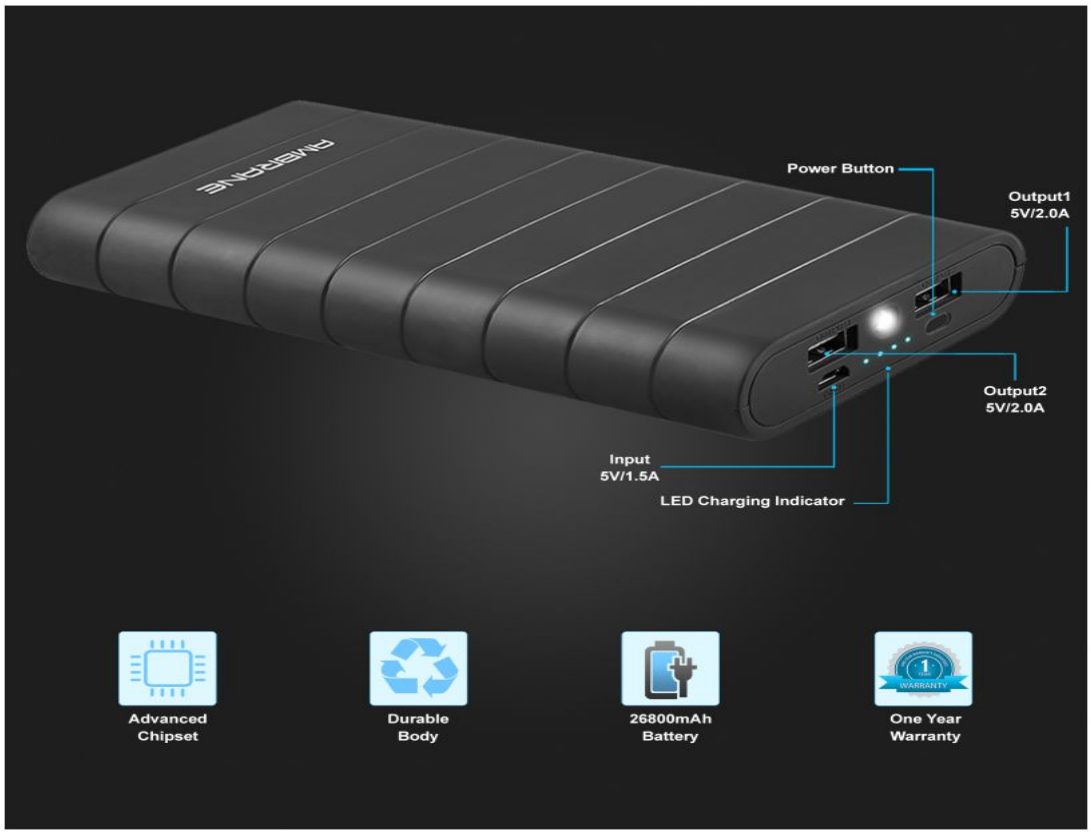
.

SIM 808 GPS/GSM Module:



- As clearly seen above, the module which is used is SIM808 GPS/GSM.
- It has two ports for antenna; one for GPS and the other for GSM.
- Supply can be given to it via USB port provided or by using the pins below it.
- The SIM is placed below the module where the provision is given.
- The LEDs on the module glow when the module is given power.
- The STATUS LED glows green when power is supplied.
- The N/W LED glows blue with a delay of 3 seconds which shows that the module is ready to connect.
- The SYNC LED glows when it is connected to the Mobile Network.
- The IPPS LED glows when it is connectecd to the GPS satellite network.

Power Bank:

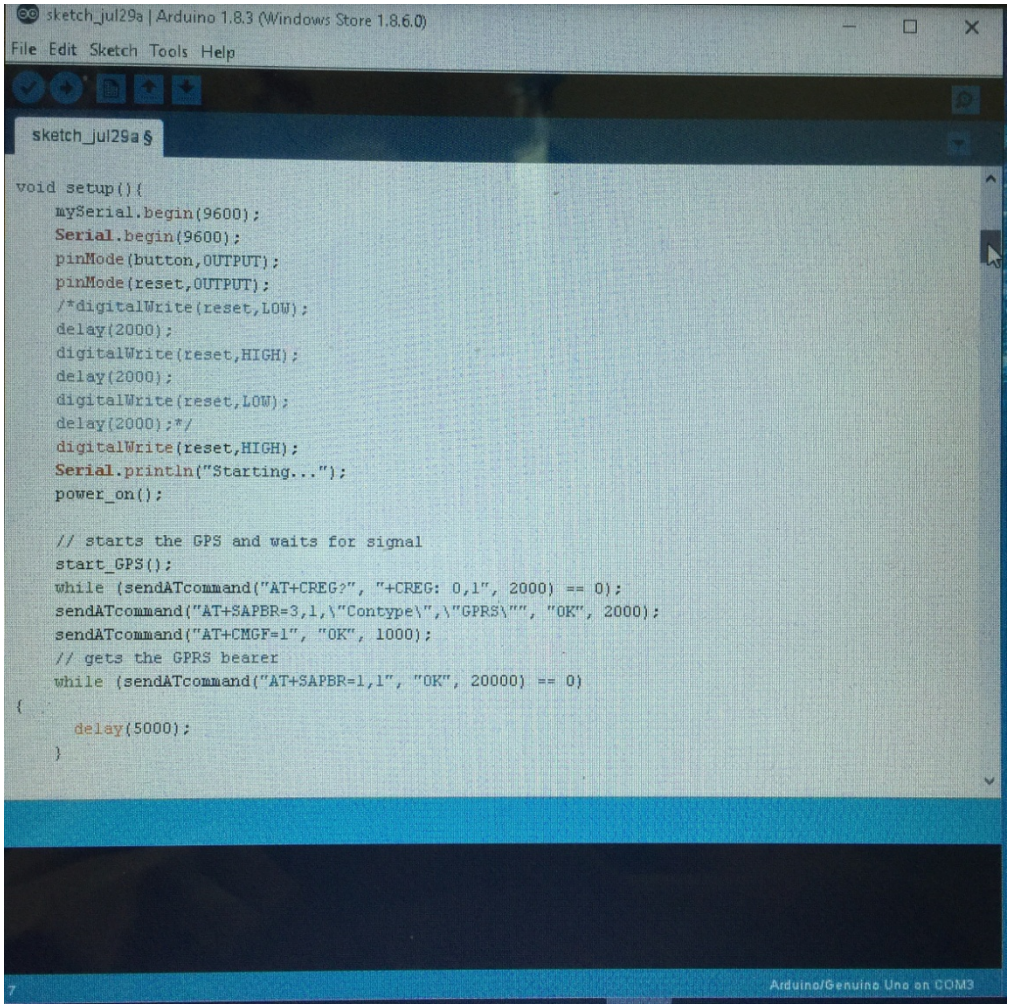


\

- Ambrane K-8 Knock Out Power Bank is used in the Project which has a capacity of 26800 mah.
- This battery can run the module upto a week.
- In the case of the Project, wind masts are brought from far away places. Hence, the vehicle not only travels long distances but also takes more amount of time to reach the destination.
- Thus, this power bank is used to support this case.

Programs used:

1. Arduino Code:



The complete code is given in the link here:

https://drive.google.com/open?id=0B4j0hDSuTk8pWEpWX2U1YkVxcUFDUTRuQXA0WEhpa1JvbzNv

2. The Server Code (Node.js) :

```
const express = require('express')
const app = express()
const body_parser = require("body-parser")
const mongoose = require("mongoose")

app.use(body_parser.urlencoded({ extended: false }))
app.use(express.static('html_files'))

mongoose.connect('mongodb://localhost:27017/maps', (error)=> {
        console.log(error)

});
var location_instance_schema = new mongoose.Schema({
    latitude: Number,
    longitude: Number,
m    traker_id: String,
    timestamp: Number
})

var location_instance = mongoose.model('location_instance',
location_instance_schema);
app.get('/test', function (req, res) {
    var test = new location_instance(
        {
            latitude: 17.42,
            longitude: 74.33,
            traker_id: 'Test',
            timestamp: 20170203234828809399
        }
    );
```

The complete program is given here:
https://drive.google.com/file/d/0B4j0hDSuTk8pU3BNd3ZDSkRHM1NEQnhwUjVEZjNlaXd0RURN/view?usp=sharing

3. The HTML file for present location of the vehicle:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Vehicle location</title>
        <meta name="viewport" content="initial-scale=1.0">
        <meta charset="utf-8">
        <style>
            #map {
                height: 100%;
            }
            html, body {
                height: 100%;
                margin: 0;
                padding: 0;
            }
        </style>
        <script src="https://code.jquery.com/jquery-3.2.1.js"></script>
    </head>
    <body>
        <div id="map"></div>
         <fieldset style="position:fixed; top:0px; right:0px; width:180px;
background-color:ivory">
            <label> Tracker id: </label>
            <input type = "text" id = "tracker" /><br><br>
            <input type = "button" value = "Get positon" id = "button"/>
        </fieldset>

        <script>
```

The complete program is given here:

https://drive.google.com/file/d/0B4j0hDSuTk8paWRKMm84dDNxeEJDYnJYb3FNc3F0TzQ3aVA0/view?usp=sharing

4. History of the Vehicle location program:

```
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
        <meta name="authors" content="" value='Aishwarya, Deepak and Tejas'>
        <meta charset="utf-8">
        <title>Vehicle History</title>
        <style>
            #map {
                height: 100%;
            }
            html, body {
                height: 100%;
                margin: 0;
                padding: 0;
            }
        </style>
        <script src="https://code.jquery.com/jquery-3.2.1.js"></script>
    </head>
    <body>
        <div id="map"></div>
        <fieldset style="position:fixed; top:0px; right:0px; width:180px;
background-color:ivory">
            <label> From: </label>
            <input type= "text" id = "timestamp1" /><br><br>
            <label> To: </label>
            <input type= "text" id = "timestamp2" /><br><br>
            <label> Tracker id: </label>
            <input type= "text" id = "tracker" /><br><br>
            <input type= "button" value= "Get route" id= "button"/>
        </fieldset>
        <script>
```

The complete program is given here:

https://drive.google.com/file/d/0B4j0hDSuTk8pVV9TdG9zNjR5M1p5WXE5VThTMGxHM01nSm5j/view?usp=sharing

PROCEDURE

Hardware Connections:

- As seen in the block diagrams, the Arduino Uno is connected to the GPS/GSM module having three main connections:
    - +5V pin of Arduino to Vin of GPS/GSM module.
    - Common ground between both the components.
    - Pin 7 and 8 of Arduino to Tx and Rx of GPS/GSM module respectively.
    - The antennas of GPS/GSM are fixed at their respective ports on the GPS/GSM module.
- The final connection is between the battery and the Arduino Uno port.

Programming:

Part 1: Arduino Program

- The program for Arduino as shown in the figure must be uploaded on the Microcontroller using Arduino IDE.
- This is used to get the GPS location from the satellites.
- The location is sent via SMS to a number specified in the program and also uploaded on a web server using AT commands.
- The SoftwareSerial library is used to communicate with the module.

Part 2: Node.js code

- This is used to create a http server.
- All the required HTML files are inlcluded in this code.

Part 3: HTML code

- In this code, the location is plotted on a map using Google API.
- The index.html file shows the present location of the vehicle.
- The history.html file shows the past locations of the vehicle with a route connecting all the locations.

Part 4: History of the position of the vehicle

- After plotting the present location of the vehicle on the map, the next step is to plot its history of travel.
- For this we use the mentioned program with which the final output would be a map on which the points of location of the vehicle are plotted.
- A text box will be seen on the output (web page) where the user is given the freedom to see the location of the vehicle for a given specific period of time (called Timestamps).

RESULT

Part 1:Arduino Output

- The Output of the Arduino program is seen on the serial monitor of the Arduino IDE.
- The Result includes the status of the SIM808 GPS/GSM module along with a set of data which is in NMEA format.

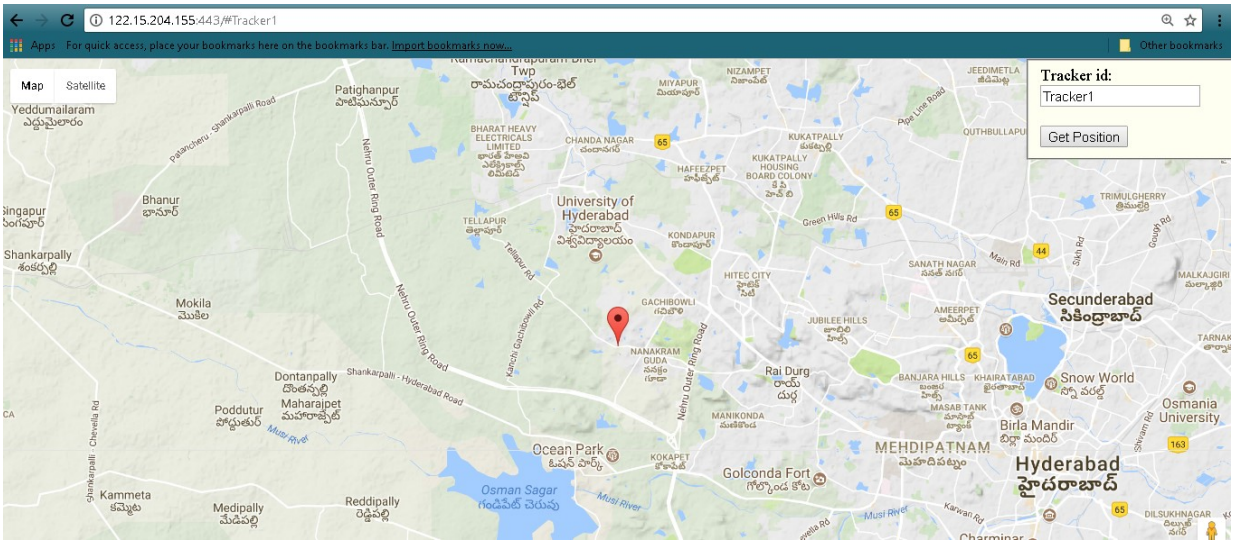
```
File  Edit  Format  View  Help
AT+HTTPINIT
AT+HTTPPARA="CID",1
AT+HTTPPARA="URL","http://122.15.204.155:443/set_location
/17.427360/78.331453/Tracker1/20170629075616.000
"
AT+HTTPACTION=0
Done!
AT+HTTPTERM
AT+CMGS="+918297382023"
▯
AT+CGNSINF
+CGNSINF: 1,1,20170629075628.000,17.427357,78.331442,590.800,0.06,2.0,1,,1.0,1.4,0.9,,12,7,,,44,,

0
AT+HTTPINIT
AT+HTTPPARA="CID",1
AT+HTTPPARA="URL","http://122.15.204.155:443/set location
/17.427357/78.331442/Tracker1/20170629075628.000
"
AT+HTTPACTION=0
Done!
AT+HTTPTERM
AT+CMGS="+918297382023"
▯
AT+CGNSINF
+CGNSINF: 1,1,20170629075637.000,17.427367,78.331452,591.100,0.22,25.0,1,,1.0,1.4,0.9,,12,7,,,44,,

0
AT+HTTPINIT
AT+HTTPPARA="CID",1
AT+HTTPPARA="URL","http://122.15.204.155:443/set_location
/17.427367/78.331452/Tracker1/20170629075637.000
```

- The information corresponding to the +CGNSINF are the variables we obtain during the tracking process.
- The format is in accordance to the variables enlisted in the Arduino code which are based on NMEA and AT commands.

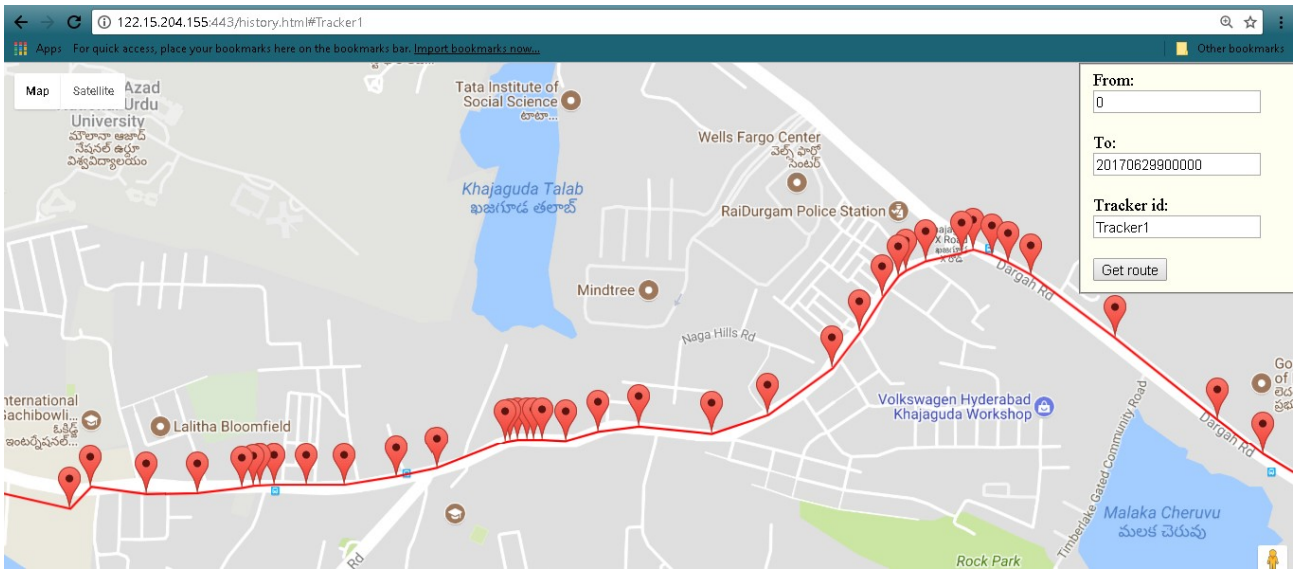
2. ***Final Maps Output:***
   - After the receiving the position of the vehicle through GPS, transmitting them GSM to the web server, the final output is obtained on to the map using Program 3.
   - The URL to check the current location of the vehicle is http://122.15.204.155:443/
   - When opened, an option of entering the Tracker ID is present.
   - In that, enter the name of the tracker which is by default the id of the module. (In this case, the Tracker ID is: Tracker1



The location is, hence, plotted on a map using Google APIs.

3. ***History Output:***
   - When the vehicles travels over a distance, the set of data received from the GPS satellite via the module is stored in a database. (Here MongoDB is used.)

- The URL to check the history of vehicle location is: http://122.15.204.155:443/history.html/



- The history of the vehicle is depicted as above.
- The Timestamp is entered in the box as seen on the page.

CONCLUSION

To begin with, the readily available GPS tracking devices are generally costly. To address this issue, the Project ‘Vehicle Tracking Module for Wind Mast Tracking’ is made cost effective*. Another feature in a GPS tracking system is that certain software has to be purchased to see the location of the object meant to be tracked. In this Project, no software was bought. The process by which the Project is developed purely based on Research and the guidance provided in the Company. The final output obtained is user friendly and easily accessible.

All in all, the Project at the end of the day is a Prototype. As in any industry or R&D, no progress is ever made without effort and consistent development.

REFERENCES

1. www.arduino.cc
2. Google maps API JavaScript documentation
3. www.W3schools.com
4. Nodejs.org
5. Simcom AT command set document
6. www.nmea.org(for format of GPS information received)
7. www.mongodb.com (reference for database)
8. Coursera (web development course)
9. Tutorials point
10. Stackexchange.com
11. Arduino.cc forum
12. www.elecrow.com
13. Jsfiddle.net
14. Api.jquery.com
15. Developers.google.com
16. Stackoverflow.com
17. www.npmjs.com
18. Arduino.stackexchange.com
19. Github.com
20. Expressjs.com