Real- Time Vehicle Tracking System

A Thesis

Submitted to the Department of Electrical and Electronics Engineering

Of

BRAC University

By

Saniah Ahmed (12321077)

Shandee Rahman (12121163)

Sudip Evans Costa (12121134)

Supervised by

Dr. A.K.M Abdul Malek Azad

Professor

Department of Electrical and Electronic Engineering

BRAC University, Dhaka.

DECLARATION

We hereby declare that our research titled "Real Time Vehicle Tracking System", a thesis submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelors of Science in Electrical and Electronics Engineering, is our own work. The work has not been presented elsewhere for assessment. The materials collected from other sources have been acknowledged here.

Date: 17.12.2015

Signature of Supervisor

…………………………………….

Dr. A. K. M. Abdul Malek Azad

Signature of Author

...…...…………………………….

Sudip Evans Costa

Signature of Co-Authors

…………………………………..

Saniah Ahmed

…………………………………...

Shandee Rahman

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our thesis supervisor Dr. A.K.M Abdul Malek Azad, Professor, Department of Electrical and Electronic Engineering (EEE), BRAC University, for his inspiration, idea and guidance for completion of the thesis. Regards to our Project Engineer, Sheri Jahan Chowdhury for her support throughout the whole thesis work. We are extremely thankful to the IT department of BRAC University for helping us enormously to complete our thesis.

ABSTRACT

Vehicle tracking system is a well-established technology in this era which is used by fleet system and owner of vehicle all over the world. It is a very safe and reliable technology. In our thesis we are going to design a system which is used for tracking and positioning of any vehicle by using Global Positioning System [GPS] and Global System for Mobile Communication [GSM]. We will be primarily focusing on tracking a Solar Assisted Rickshaw Van using Arduino Uno R3 and GSM module sim908. The design is an embedded application, which will continuously monitor a moving vehicle and report the status of vehicle on demand. For doing so the Arduino Uno R3 is interfaced serially to a GSM modem and GPS receiver. The GSM modem is used to continuously send the position of the vehicle from remote place. The GPS modem that uses satellite technology for its navigation system will continuously give data like longitude, latitude, speed , distance travelled etc. When the request by user is sent to the number at the modem in the form of SMS, the system automatically sends a return reply to the mobile indicating the position of the vehicle in terms of latitude and longitude via SMS. We will also view the position of vehicle on a digital mapping i.e. on Google map with the help of software via Internet. For this we will be using the software XAMPP and Google Map API. XAMPP is a free and open source cross-platform web server solution stack package consisting mainly of the Apache HTTP Server, MySQL database and interpreters for scripts is written in PHP and Perl programming languages. The Google Maps API allow for the embedding of Google Maps onto web pages using a JavaScript interface which is designed to work on desktop browser application. The MySQL database is used to store all the data of the GPS and Google Map API is used display the location information through a Google Map. We will also be able to control the vehicle if it is stolen.

Table of Contents

LIST OF FIGURES:

LIST OF TABLES:

ABBREVIATIONS:

GPS- Global Positioning System

GSM- Global System for Mobile Communication

SMS- Short Message Service

AVL- Automatic Vehicle Location

GIS- Geographic Information System

RF- Radio Frequency

AGPS- Assisted Global Positioning System

RFID- Radio Frequency identification

API- Application Programming Interface

APN- Access Point Name

SRAM- Static Random Access Memory

EEPROM- Electrically Erasable Programmable Read only Memory

PWM- Pulse-width Modulation

ICSP- In-Circuit System Programming

DC- Direct Current

PHP- Hypertext Preprocessor

HTTP- Hypertext Transfer Protocol

HTML- Hypertext Markup Language

IDE- Integrated Development Environment

CSS- Cascade Style Sheet

DBMS- Database Management System

SQL - Structured Query Language

NoSQL- Not Only SQL

RDBMS- Relational Database Management System

JDBC- Java Database Connectivity

ODBC- Open Database Connectivity

DML- Data Manipulation Language

IBM- International Business Machine

GET- Graduate Engineer Trainee

2D- Two Dimensional

XML- Extensible Markup Language

NMEA- National Marine Electronics Association

POST- Power-On Self-Test

IDCOL – Infrastructure Development Company Limited

Chapter 1: Introduction

1.1. Introduction to Tracking System

The vehicle tracking system is a total security and fleet management solution. It is the technology used to determine the location of a vehicle using different methods like GPS and other navigation system operating via satellite and ground based stations. Modern vehicle tracking system use GPS technology to monitor and locate our vehicle anywhere on earth, but sometimes different types of automatic vehicle location technology are also used. The vehicle tracking system is fitted inside the car that provides effective real time location and the data can even be stored and downloaded to a computer which can be used for analysis in future. This system is an essential device for tracking car any time the owner wants to monitor it and today it is extremely popular among people having expensive cars, used as theft prevention and recovery of the stolen car. The data collected can be viewed on electronic maps via internet and software.

The device includes modern hardware and software components that help to track and locate automobiles both online and offline. A tracking system comprises of mainly three parts- vehicle unit, fixed based station and database with software system [1].

The vehicle unit incorporates the hardware part that is the Arduino, GPS and GSM modem kept inside the vehicle that is to be tracked. The unit is mainly based on a modem that receives signals from the satellite with the help of GPS antenna. This modem then converts the data and sends the vehicle location information via SMS as well as a mobile application named 'VTS' which is synchronized with the web page and to a server which can be displayed on digital mapping.

Fixed Based station consists of a wireless network system that receives and transfer the information to the data center. The based station contains software and geographic map useful for locating the vehicle. Maps of every city are available in the based station that has an in-built Web Server.

Database and Software are used to give the location that is the coordinates of each visiting point that is saved in the database, which can be later displayed in a screen using Google maps.

However, to view the location the vehicle has travelled the user has to connect themselves to the web server.

Vehicle Security is a primary concern for all vehicle owners. Owners as well as researchers are always looking for new and upgraded vehicle security systems. For the modernization of technology it is now possible to track and closely monitor vehicle in real time as well as to check the history of vehicles' movements. One has to be thankful to Vehicle Tracking System that has helped enormously to maintain the security of the vehicle by tracking its activities at regular time interval. The system uses Global Positioning System [GPS], to find information about the location of the vehicle that is to be monitored and then send the latitude and longitude to the monitoring center through satellite. At the monitoring center different software is used to display the vehicle on the Google map. This is how our system tracks automobiles in real time. Due to real-time tracking facility, vehicle tracking systems have become increasingly popular among owners of vehicles as they are able to monitor their vehicle continuously.

The tracking hardware is installed inside the vehicle in such a manner that it is not visible from outside the vehicle. Hence, it works as a secret unit which continuously sends the coordinates to the monitoring center.

Monitoring center Software helps the vehicle owner with a view of the vehicle location on an electronic map. The user can use any browser to connect to the server and monitor the targeted vehicle on Google Map. Thus it saves the user from the hassle of calling the driver to know the vehicle's location as it is now possible to track vehicle online.

The vehicle tracking system is one of the biggest technological advancements for security purpose. It enables the owner to virtually keep an eye on his vehicle any time and from anywhere in the world.

A vehicle tracking system combines the installation of an electronic device in a vehicle with purpose-designed software to allow the owner to monitor the vehicle, collecting data in the process from the field and deliver it to the base of operation. Urban public transportation companies are an increasingly common user of tracking systems, particularly in large cities.

Vehicle tracking systems are also popular among people as a theft prevention and retrieval device. When the vehicle is stolen, the location information send by tracking unit can be used to find the location and coordinates can be sent to police for further action. Police can simply follow the signal given by the tracking system and locate the stolen vehicle.

When used as a security system, a Vehicle Tracking System may serve as either an addition to or replacement for a traditional car alarm. Some vehicle tracking systems make it possible to control vehicle remotely, including block doors or engine in case of emergency. Due to the advancement in technology Vehicle tracking Systems that can even detect unauthorized movements of the vehicle and then alert the owner. This gives an edge over other pieces of technology for the same purpose.

The device consists of a module that communicates independently and has the ability to work simultaneously with the local GSM network (DTMF, SMS, GPRS and GSM) [1]. It provides effective real time location. Using this system we can get accurate location of the vehicle that is to be tracked.

The tracking system is installed on the vehicle (car, bus, truck).The device is controlled by mobile phone that makes it possible for the user to communicate with the system wirelessly. The vehicle tracking system has a SIM slot that uses a GSM SIM to receive and transmit SMS. The user can send a SMS from a mobile phone and locate the vehicle that is to be tracked. We have also created a flat file database that is stored on a host computer system that contains a 'flat file'. A flat file can be either a text file or a binary file, for our system we have used text file as our flat file. We have also used the software XAMPP which make it possible to access the text file and view the position of the vehicle on the Google map from anywhere on earth. The system also provides the facility to prevent theft of the vehicle. So we can divide the whole operation of the vehicle tracking system into two parts-

1. Tracking the location of vehicle

2. Providing protection of vehicle

The system consists of GPS receiver which provides real time location of vehicle. This real time data is stored in the database. The GSM module is directly connected to the microcontroller which is use to send and receive the SMS and send the data to the database. GSM module takes the data and sends this data to the user's mobile phone that has sent the request for the coordinates of the position. This data consist of longitude, latitude and a link and when we click on the link we are able to see the location on Google map.

This system also has another special feature which provides not only the location of vehicle but also protection. If the vehicle is stolen, after knowing the location of the vehicle, the owner can follow the car and give the information of location to the police for faster recovery of the stolen car.

In our thesis we have used Arduino UNO R3 as a microcontroller that is used for interfacing to various hardware peripherals. The current design is an embedded application, which will continuously monitor a moving vehicle and report the status of the vehicle on demand. For doing so the Arduino is interfaced serially to a GSM Modem SIM908 and GPS Receiver. The GPS receiver will continuously give the data that is the latitude and longitude which indicates the position of the vehicle. A GSM modem is used to send the position (Latitude and Longitude) of the vehicle from a remote area. The GPS modem gives many parameters as the output. The data is sent to the user's mobile phone who has demanded for the position of the vehicle; it also saves the data in the database for future use.

GPS tracking unit is a device that uses the Global Positioning System to determine the precise location of a vehicle or other asset to which it is attached. A GPS receiver is operated by a user on Earth, it measures the time taken by radio signals to travel from four or more satellites to its location, it then calculates its distance from each satellite, and from this calculation it determines the longitude, latitude, and altitude of that position. By following triangulation or trilateration methods the tracking system determines the location of the vehicle easily and accurately [1]. Trilateration is a method of determining the relative positions of objects using the geometry of triangles. To "triangulate," a GPS receiver accurately measures the time taken by the satellite to make its brief journey to Earth(less than a tenth of a second) and hence measures its distance from the satellite using the travel time of the radio signal. To determine the distance between it and the satellite, the measured time is multiplied by the speed of a radio wave that is 300,000 km (186,000 miles) per second [1]. The coordinates of latitude and longitude can be send to the user on request via SMS, or it may be transmitted and stored in the database, using a cellular or satellite modem that is the GSM modem embedded in the unit. This enables the user to display the asset's location on the Google map either in real time or later whenever the user wants the data for further analysis.

1.2. Impact of introducing tracking system in vehicle

Vehicle tracking systems are commonly used by fleet operators for fleet management functions such as fleet tracking, routing, dispatch, on-board information and security [1]. Commercial fleet operators as well as urban transport companies use this system for various purposes that include monitoring schedules of buses in service, triggering any change of buses' destination and manage per-recorded announcements for passengers.

The American Public Transportation Association estimated that, at the beginning of 2009, around half of the public buses in the United States were already using a tracking system to trigger automated stop impaired customers and to do any important announcements to the passengers already on board [1]. This also includes external announcements (triggered by the opening of the bus's door at a bus stop, announcing the vehicle's route number and destination, primarily for the benefit of visually identifying the next stop, as the bus approaches a stop.

As the vehicle follows its route the data are collected and is stored in a computer system where it is compared with the location the vehicle was scheduled to be in at that very moment. It also updates the driver telling him how early or late he is at any given time making it easier for the driver. This device also helps to provide customers with real-time information such as time required for the arrival of the next bus at a given stop, it helps to save passengers time as it provides the passenger with updated information. Transport companies that offer this kind of information assign a unique number to each stop, and passengers can obtain information by entering the stop number on the transit system's website.

To provide accurate information to passengers some transit agencies have uploaded a map on their website, with icons indicating the current locations of buses in service on each route. There are also some transit companies that keep this information only for their own use which can be accessed by their employees only. Other applications include monitoring driving behavior of the driver which is helpful if the company have appointed any new driver.

The vehicle tracking system is also used as an anti-theft system. If a vehicle is stolen the owner can easily track the car and will be able to control the car by turning off the engine and lock the door of the vehicle via SMS. After the introduction of vehicle tracking system the chances of recovering the car has increased enormously.

Vehicle tracking system has made people's life much easier than before. Now, owner of cars can easily track their car from any corner of the world and control it if it is stolen. Passengers can even get notification about schedule of public transport that saves their precious time.

1.3. Block Diagram

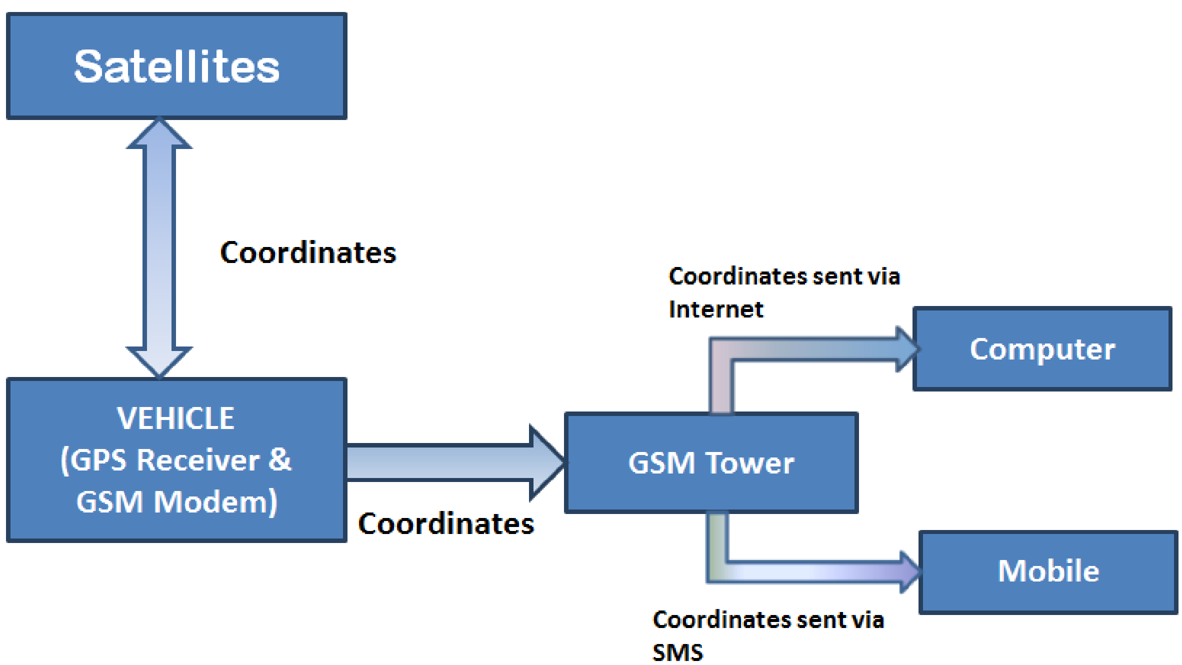


Fig.1.1: Block Diagram of Vehicle Tracking System

The block diagram of vehicle tracking system shows how our system actually works. The vehicle tracking unit is installed inside the vehicle that is to be tracked. The GPS receiver receives the coordinate from the satellite which is then send to the GSM tower by the GSM modem. The coordinate is then sent to a computer via internet where it is stored in the database for displaying the location on Google map. The user can also see the location of the vehicle in a mobile phone, when the user sends an SMS to the GSM modem in the vehicle, the GSM modem send another SMS back to the user with the coordinates of the location of the vehicle along with a Google map link.

1.4. Overview of Content

The following chapters portray the work that has been accomplished and the impact of introducing tracking system in the economy. The second chapter gives an overview of the history and literature of tracking system and different technologies that can be used in a tracking system. The third chapter gives an overview of the whole system with existing features. The fourth chapter introduces the database, Google Map and the software that we have used. The fifth chapter specifies the working of the system and the ways we have used to display the result. The sixth chapter gives an overview of the result analysis and the advantages along with the limitation of our vehicle tracking system. In the seventh chapter we are going to conclude and suggest future plans for this system.

Chapter 2: System Overview

2.1. History and Literature Review

Global Positioning Systems (GPS) were designed by the United States Government and military, which the design was intended to be used as surveillance. The GPS was invented as a collaborative effort by the United States' Department of Defense and Dr. Ivan Getting as a means to create a satellite course-plotting system, primarily used for navigation purposes [2].

At that time, the GPS project cost approximately $12 billion for the design and launch of 18 satellites, six in each of the orbital planes spaced 120 degrees apart, and their ground stations. GPS uses these satellites as reference points to determine and give the accurate geographical positions on map.

The idea for a global positioning system was initially planned to be used by military and intelligence organizational during the Cold War, with the introduction of the project stemming from the Soviet-launched spacecraft Sputnik. Since its introduction in the 1960s, GPS has developed into a larger and more advanced satellite network constellation that orbits Earth at fixed points in space to send signals to anyone with a GPS receiver. The signals carry a time code and geographic data point that enables us to display a device's exact position anywhere on the planet [2].

The design of GPS is partly similar to the design of ground-based radio navigation systems, such as LORAN and the Decca Navigator, developed in the early 1940s and were used during World War II. Additional inspiration for the GPS system came when the Soviet Union launched the first Sputnik in 1957 [3]. A team of U.S. scientists led by Dr. Richard B. Kershner were monitoring Sputnik's radio transmissions. They discovered that, because of the Doppler Effect, the frequency of the signal being transmitted by Sputnik was higher as the satellite approached and lower as it moves away from them. They realized that since they knew their exact location on the globe, by measuring the Doppler distortion it was possible to pinpoint where the satellite was along its orbit.

The first satellite navigation system was first successfully tested in 1960. It delivers a navigational fix approximately once per hour using a constellation of five satellites. In 1967, the U.S. Navy introduced the timation satellite which demonstrated the ability to place accurate clocks in space that is the technology used by the GPS system. In the 1970s, the ground-based Omega Navigation System, based on signal phase comparison, became the first world-wide radio navigation system [3].

In February 1978 the first experimental Block-I GPS satellite was launched. The GPS satellites were initially manufactured by Rockwell International and are now mass-produced by Lockheed Martin.

In 1983, after Soviet interceptor aircraft shot down the civilian airliner KAL 007 in restricted Soviet airspace, killing all 269 people on board, U.S. President Ronald Reagan announced that the GPS system would be made available for civilian uses once it was completed. Hence, the government signed a treaty to allow civilians to buy GPS units also only the civilians would get precise downgraded ratings.

The oldest GPS satellite still in operation was launched in August 1991. By December 1993 the GPS system achieved initial operational capability and a complete constellation of 24 satellites was in orbit by January 17, 1994 [3].

In the initial period of tracking only two radios were used to exchange the information. One radio was attached to the vehicle while another at base station by which drivers were enabled to talk to their masters. Fleet operator could identify the progress through their routes.

The early technology also has some limitation. It was restricted by the distance which became a hurdle in accuracy and better connectivity between driver and fleet operators. Base station was dependent on the driver for the information and a huge size fleet could not have been managed depending on man-power only [1].

The scene of vehicle tracking underwent a change with the arrival of GPS technology. This reduced the dependence on man-power. Most of the work of tracking became electronic. Computers proved a great help in managing a large fleet of vehicle. This also made the information authentic. As this technology was available at affordable cost all whether small or big fleet could take benefit of this technology.

Because of the accessibility of the device computer tracking facilities has come to stay and associated with enhanced management. Today each vehicle carries tracking unit which is monitored from the base station. Base station receives the data from the unit. All these facilities require a heavy investment of capital for the installation of the infrastructure of tracking system for monitoring and dispatching.

Today's GPS applications have vastly developed. It is possible to use the Global Positioning Systems to design expense reports, create time sheets, or reduce the costs of fuel consumption. We can also use the tracking devices to increase efficiency of employee driving. The GPS unit allows us to create Geo-Fences about a designated location, which gives us alerts once the driver passes through that location. This means we have added security combined with more powerful customer support for our workers [1].

Nowadays GPS units are great tracking devices that help fleet managers stay in control of their business. The applications in today's GPS units make it possible to take full control of any company. It is clear that the tracking devices offer many benefits to companies, since we can build automated expense reports anytime. GPS units do more than just allow companies to create reports. These devices also help to put an end to thieves. According to recent reports, crime is at a high, which means that car theft is increasing. If we have the right GPS unit, we can put an end to car thefts because we can lock and unlock our car anytime we want to. GPS is small tracking device that is installed in a car and it will supply feedback data from tracking software that loads from a satellite.

In this paper GPS based vehicle navigation system is implemented. This is done by fetching the information of the vehicle like location, distance, etc. by using GPS and GSM. The information of the vehicle is obtained after every specified time interval defined by the user. Then this periodic information of location is transmitted to monitoring or tracking server. This transmitted information is displayed on the display unit by using the Google earth to display the vehicle location in the electronic Google maps.

This system uses Global Positioning System (GPS) which is used to receive the coordinates of latitude and longitude form the satellite during the critical information. We all know that tracking system is now-a-days a very important in modern world. This system can be used in the monitoring our car, also in tracking the theft of the vehicle and in many more other applications. This system uses microcontroller, Global Positioning System (GPS) and Global System for Mobile Communication (GSM). Only one GPS device is used in this system and GSM enable a two way communication process. GSM modem is provide with a SIM card which uses the same and regular communication process as we are using in regular phone.

From the above mentioned vehicle tracking techniques we can say that each technique is appropriate with its function but in some system we need continuous net access and this system can go down if net fails. In the first system the GPS tracks the vehicle location and send it to the controller and the Google maps display the location of the Vehicle on the display unit, this system is useless without net because the location of vehicle can only be presented by the Google maps. In the other system an SMS of the coordinate of the location is send to the user on request which does not require an access to internet. By considering all these factors the upcoming implementation should introduce many more facilities which will make the system user friendly and efficient.

2.2. Different Technologies used in Tracking System

2.2.1. Active and passive tracking

Several types of vehicle tracking devices exist. Typically they are classified as "passive" and "active". "Passive" devices store GPS location, speed, heading and sometimes a trigger event such as key on or off, door open or closed. Once the vehicle returns to a predetermined point, the device is removed and the data downloaded to a computer for evaluation. Passive systems include auto download type that transfer data via wireless download. "Active" devices also collect the same information but usually transmit the data in real-time via cellular or satellite networks to a computer or data center for evaluation.

Passive trackers do not monitor movement in real-time. When using a passive GPS tracker, you will not be able to follow every last move that a tracked person or object makes. Instead, information that is stored inside of a passive tracker must be downloaded to a computer. Once tracking details have been downloaded, it is then possible to view tracking details.

After we have gathered all of the information we need from a passive tracker, we can place the tracker back on the same (or different) vehicle. Aside from the fact that a passive tracking device is entirely reliable, the main reason people choose passive trackers is that these devices are less expensive than active trackers. Most passive GPS tracking devices are not attached to a monthly fee, which makes these trackers affordable.

In contrast to passive devices, active GPS trackers will allow one to view tracking data in real-time. As soon as we place an active tracker on a vehicle, we will be able to view location, stop duration, speed, and other tracking details from the comfort of your home or office. Active GPS trackers are ideal when it comes to monitoring vehicle that need to be tracked at regular time interval.

While active tracking devices are more expensive than passive devices (most come with monthly fees), this expensive is usually justified. An active GPS tracker that comes with a reliable interface (and excellent tracking software), and you will be able to track anything or anyone quickly and efficiently.

When most people picture a GPS tracking device, they are picture a real-time tracker. These trackers can be attached to any object while a person monitors all activity from a home computer. For example, if you were to place a real-time tracker on a vehicle, you could then watch as the vehicle makes stops, takes alternate routes, and sits idling – all in real-time. GPS trackers that work on a real-time basis are usually considered "active" trackers, while those that do not include real-time tracking are considered "passive" trackers.

There are many advantages associated with a real time tracker. The most important advantage is that the GPS locater is convenience. Rather than waiting to download data to a computer (as is the case with most passive trackers), a tracker that works in real-time does not require any waiting. Since real-time trackers come with software that allows a user to track an object in real-time, watching any object's progress is simply a matter of sitting at a computer.

Many modern vehicle tracking devices combine both active and passive tracking abilities: when a cellular network is available and a tracking device is connected it transmits data to a server; when a network is not available the device stores data in internal memory and will transmit stored data to the server later when the network becomes available again.

Historically vehicle tracking has been accomplished by installing a box into the vehicle, either self-powered with a battery or wired into the vehicle's power system. For detailed vehicle locating and tracking this is still the predominant method; however, many companies are increasingly interested in the emerging cell phone technologies that provide tracking of multiple entities.

2.2.2. Different Types of Tracking System

There are three main types of GPS vehicle tracking that are widely used. They all use active devices. They are:

1. Automatic Vehicle Location (AVL) system
2. Assisted Global Positioning System (AGPS)
3. Radio Frequency Identification (RFID)

- **Automatic Vehicle Location (AVL) system-** AVL system is an advanced method to track and monitor any remote vehicle with the device that receives and sends signals through GPS satellites. AVL comprises of Global Positioning System (GPS) and Geographic Information System (GIS) in order to provide the real geographic location of the vehicle. AVL system consists of PC-based tracking software to dispatch, a radio system, GPS receiver on the vehicle and GPS satellites. Among the two types of AVL, GPS-based and Signpost-based, GPS-based system is widely used. The tracking method uses GPS satellite to locate the vehicle equipped with GPS modem by sending satellite signals. The accuracy of the tracking method depends on the AVL system which provides the vehicle location with the accuracy of about 5m to 10m. The information transmitted by the tracking system to the base station is location, speed, direction, mileage, start and stop information and status of vehicle. The information of the vehicle is often transmitted to the central control system (base station) from the vehicle after every 60 seconds. If the base station receives the data, it displays it on a computerized map. GPS receiver on the vehicle receives the signals of its geographic location. Then the receiver sends that data plus speed, direction, etc. to the base station via a radio system. If AVL system is used to track a vehicle the average cost of per vehicle is $1 to $2 per day. The system can provide additional services like: vehicle route replay facility, external sensor data, speed alerts. The system also has some limitation; using the AVL system we cannot get accurate, complete and sufficient satellite data in dense urban areas or indoors and when transmission is blocked by natural obstructions (heavy tree cover) or many buildings. It can also occur in RF-shadowed environments and under unfriendly Radio Frequency (RF) conditions. Sometimes, a position fix can be impossible.

- **Assisted GPS (AGPS) system-** In AGPS system, a terrestrial RF network is used to improve the performance of GPS receivers as it provides information about the satellite constellation directly to the GPS receivers. AGPS uses both mobiles and cellular networks to locate the accurate positioning information. AGPS is used to overcome some limitations of GPS. With unassisted GPS, locating the satellites, receiving the data and confirming the exact position may take several minutes. The tracking method of AGPS uses GPS satellites to track the vehicles. A GPS receiver in vehicle is always in contact with 4 satellites (3 satellites determine latitude, longitude and elevation and the fourth provides element of time) hence it never fails to detect the location of a vehicle. Location of the vehicle is provided with accuracy of between 3m and 8m, and speed of 1km using this method. Information like Vehicle location, average speed, direction, path traversed in a selected period and alerts (Engaged/Unengaged, speed limit, vehicle breakdown and traffic jam) are delivered by the tracking system to the base station. The system provides continuous updates after every 10 seconds while the vehicle is in motion. It also provides data storage for up to 1 year. The location is retrieved from the GPS device and relayed as a SMS using the cell phone by the Client Node to the Base station. This system is more expensive than the AVL system as it gives continuous update of the vehicle location. If the user needs update after every 10 seconds then the subscription for this system is $1.33 per day per vehicle and if the user needs update after every 5 seconds it is

$1.67 per day per vehicle. The system can provide further services like atomic time (Accurate Time Assistance). There is a "panic" button. When pressed, you can contact an operator and he or she will help you out or keep you safe from accidents or hijacks. The system has also some limitations as GSM network is used to transmit data from the vehicle to the base station, the cost of sending SMS is a major concern to be considered.

- **Radio Frequency Identification (RFID) System-** RFID is an automatic identification method using devices called tags to store and remotely retrieves data. RFID uses radio waves to capture data from tags. The tracking method of RFID is comprised of three components: tag (passive, semi passive and active), reader (antenna or integrator) and software (middleware). RFID tag which contains microelectronic circuits sends the vehicle information to a remote RFID reader which is then read via the software. This system provides the location of the vehicle with the accuracy of 4m to 6m. Information such as location of the vehicle, mileage and speed are delivered by the tracking system to the centre. The information is updated every one minute. The information is sent to and received from RFID tags by a reader using radio waves. RFID reader, basically a radio frequency (RF) transmitter and receiver, is controlled by a microprocessor or digital signal processor (DSP). RFID reader with an attached antenna reads data from RFID tags.

Chapter 3: Hardware Components

3.1. Introduction

The core function of our project is to develop a tracking system that is cost effective so we have made use of the following components that has effective operation and usage. In this section, hardware part, i.e. hardware components used for the project are discussed in details. The heart of the project that is, as microcontroller we used Arduino Uno. Initially, we worked on getting geo coordinates i.e. latitude and longitude and for this we made use of the GPS module to capture location, speed and time of last received data in accordance. Then using GSM technology, the captured data already sent to the web server is stored and for this we have used SIM908 module.

3.2. Components used:

3.2.1. Arduino Uno R3

3.2.2 SIM 908 Module

3.2.3 GPS and GSM Antenna

3.2.4 Battery

Details of each hardware components used are described below.

3.2.1. Arduino Uno R3:

Microcontroller used for our project is Arduino Uno R3. The R3 is the third, and latest, revision of the Arduino Uno. The Arduino Uno is a microcontroller board based on the ATmega328. The ATmega328 has 32 KB (with 0.5 KB occupied by the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It is simply connected to a computer with a USB cable. The V_{in} is the input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). The 5V pin outputs a regulated 5V from the regulator on the board. The microcontroller board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the V_{in} pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. So it is advised not to do so. Maximum current draw is 50 mA [5]. An Arduino board is based on a AVR microcontroller chip and when the board with nothing wired or attached to it consumes around 80ma of 5 volt current. The Clock speed of the Arduino is 16 MHz so it can perform a particular task faster than the other processor or controller. The AVR chip is clocking at 16 MHz continuously no matter what the code is doing, it never 'halts' so its current consumption is basically independent of the code you have it execute. Only if you put the AVR chip into one of its 'sleep modes' can you halt code execution and drastically cut current consumption for the AVR chip, however the rest of the other components on the Uno will continue to draw their normal current consumption. Also the Arduino does not provide any 'sleep mode' examples so one will have to look for other user supplied coding example. Arduino board supports I2C and SPI communication. The Arduino software includes wire library for I2C and SPI library for the SPI communication [6].

Fig.3.1: Arduino Uno R3

POWER SAVING:

Various power-saving options for running the project from battery power are listed below[7]:

- Turn off internal modules in software that are not needed (e.g. SPI, I2C, Serial and ADC).
- Turn off brownout detection.
- Turn off the watchdog timer.
- Put the processor to sleep.
- Do not use inefficient voltage regulators - if possible run directly from batteries.
- Don not use power-hungry displays (e.g. indicator LEDs, backlit LCDs).
- Arrange to wake the processor from sleep only when needed.
- Turn off (with a MOSFET) external devices (e.g. SD cards, temperature sensors) until needed.

3.2.2. SIM908 Module:

SIM908 module is a complete Quad-Band GSM / GPRS module which combines GPS technology for satellite navigation. It has a SIM application toolkit where SIM card can be inserted. The compact design which integrated GPRS and GPS in a SMT package significantly saves both time and cost for one to develop GPS enabled applications. A modem GSM & GPRS with SIM908 module allows to create data connections on the GSM network through a standard USB interface. The cellular modems, particularly USB-stick ones, are now at very affordable prices. However, they are limited: they are explicitly designed for Internet connections, so one cannot use it as a normal modem and so implement, for example, a point to point data communications with them.

To switch ON the cellular module, the microcontroller has to put high the line ON/OFF (pin 1 on connector). This saturates the T2 transistor that drives to low the line PWR of GSM. SIM908 is designed with power saving technique so that the current consumption is as low as 1.0mA in sleep mode (GPS engine is powered down). The range of DC005 voltage input is 5 - 26V, when use the 5V power as the power, it is needed to make sure that the power supply can provide 2A current.

The SIM908 module has two different serial ports on board, one for the cellular section of the module and one for the GPS section. The serial port on cellular allows the full management of SIM908 module, therefore it can be used to configure and communicate with the GPS receiver, in order to call for data about satellite status and geographical positioning and to transfer them to the microcontroller. This is the approach followed in the design of this project.

All the GPS function is controlled by AT command via serial port. This module uses AT command to execute user’s desired functions. While using the GPS function, two AT commands are send to open the GPS function, and the commands are AT+CGPSPWR=1 and AT+CGPSRST=1 respectively; two instructions are used to power GPS and reset GPS. And then, the GPS TTL level interface will send data out and the baud rate is 115200 by default [8].

Fig.3.2a: SIM 908 Module (front side)

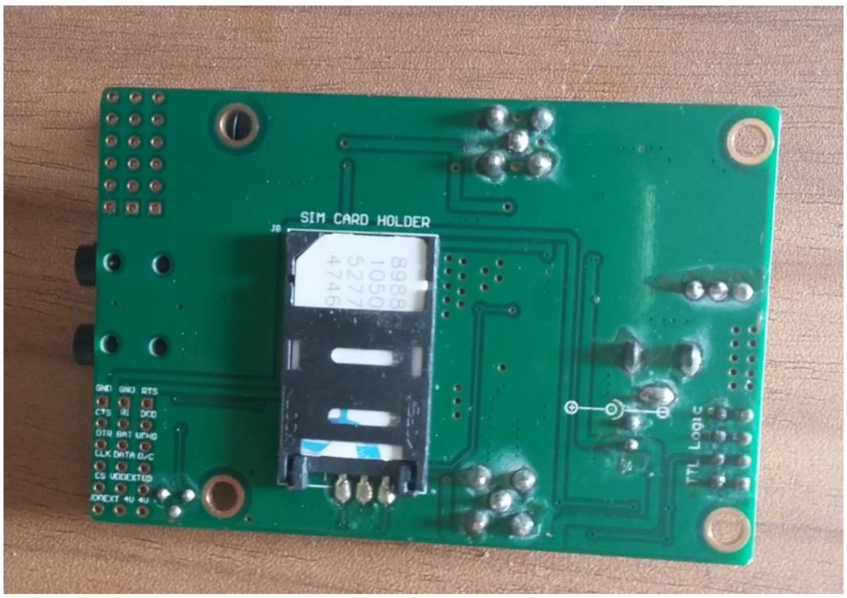


Fig.3.2b: SIM 908 Module (back side)

3.2.3. GPS and GSM Antenna

GPS Antenna:

This GPS antenna draws about 10mA and will give you an additional 28 dB of gain. It got a 5 meter long cable so it will easily reach wherever it is needed to. The antenna is magnetic so it will stick to the top of a car or truck or any other steel structure. Its operating frequency range is 1575.42±1.023 MHz and voltage range is 2.5V- 5.5V and corresponding current range is 6.6 mA - 16.6 mA [9].

GPS signals are extremely weak and present unique demands on the antenna so the choice of antenna plays an important role in GPS performance. A GPS unit needs to have a clear, unobstructed sky view, to best receive the microwave signals that allow it to communicate with satellites. GPS Down/Up converter used for very long cable runs. This GPS antenna that receives the GPS signal, converts it to a lower frequency which is then sent down the cable. Next to the GPS receiver is an up converter that converts the signal back to the original frequency and delivers it to the GPS receiver.

Fig. 3.3: GPS Antenna

GSM Antenna:

GSM communications are dependent on antennas. The antenna is what allows communications signals to be sent and received. The antenna that we have used in our project provides operation at both GSM Quad Band Frequencies with +2dBi gain [10]. This antenna operates in Quad Band 890/960, 1710/1880 MHz Frequencies and it’s an omni-directional [11].

Fig.3.4: GSM Antenna

3.2.4. Battery

We have used a sealed Lead- acid battery of 12V which has nominal capacity of 7.2mAh. It is rechargeable and has maintenance -free operation and low discharge functionality.

Fig.3.5: Lead Acid Battery

Chapter 4: Software Component

4.1. Method Followed

In this project, we built a vehicle tracking device which will be imbedded into a vehicle and monitoring that vehicle in real time. In tracking device, GPS receiver receives the data or information mainly latitude and longitude of the particular vehicle from the satellite which information is transferred over mobile phone via Short Message Service (SMS) by using GSM modem. GSM modem is connected with Arduino Uno R3 microcontroller. The positional information is also transferred to a remote server over Hypertext Transfer Protocol (HTTP) connection. HTTP server stores the acquiring positional data in the database. A web application had been built using Hypertext Preprocessor (PHP), JavaScript or jQuery which is a library of JavaScript, Hypertext Markup Language (HTML) with embedded Google Map. The JavaScript runs in the browser and integrates this information into Google Maps through Google Maps Application Programming Interface (API) which displays the position on a map. Since the geo positional data is retrieved every second and the maps updated at the same interval, thus a real time vehicle tracking is achieved.

4.2. Arduino IDE

For designing the vehicle tracking system we need to have knowledge about various programming languages to ensure the communication between microcontroller and GSM SIM module and to store and retrieve data of vehicle's location into the web application through Google Map. For receiving data from the satellite and sending data into the database, Arduino microcontroller and SIM908 module had been programmed by using Arduino IDE software.

Arduino IDE software is a open source software which is used for compiling the program into the microcontroller. In this software C- programming language has been used for code. The coding has mainly two parts – void setup () which is known as preparation for the program, runs only once and another part is void loop () which is known as execution of the program. We had written some function to get the authentication of SIM, GPS data, send the data to the script over HTTP. We had used AT command for SIM908 to communicate with the Arduino and the server or mobile device.

For monitoring the location of the vehicle into a google map, a web application had been built. Developing that web site, we need to know about Xampp server, PHP, HTML, JavaScript, MySQL.

4.3. XAMPP

XAMPP stands for Cross Platform, Apache, MySQL PHP and Perl. It is a simple Apache distribution and it is easy for developers to create a local web server for testing purposes. All needed is to set up a web server – server application (Apache), database (MySQL) and scripting language (PHP).

PHP is open source and widely used scripting language. PHP scripts are executed on the server. PHP files contain text, HTML, Cascade Style Sheet (CSS), JavaScript and PHP code. While the PHP code is running on the server, the result is returned as a plain HTML to the browser. PHP can read, write, delete, open, modify, and create files on the server [12].

JavaScript is a programming language of the web page. In this project, we had used jQuery which is JavaScript library. jQuery makes much easier to use JavaScript on a website. jQuery takes a lots of common tasks that require a lots of lines of JavaScript to execute and combine them into method so that we can easily call with a single line of code [13].

4.4. Database

Data is known fact that can be recorded and have implicit meaning [14]. A database is a collection of related data which is organized so that it can be manipulated, updated and stored very easily. A database is designed, built and populated with data for specific purpose. A database management system (DBMS) is a software system to create and manage the database. It helps the users or programmers to update, retrieve, create and manage the data in a proper way. DBMS interacts with the user, other applications, and the database itself to capture and analyze data. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. This three foundation elements provide concurrency, data integrity, security and uniform administration procedures. Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

- **Data definition** – Creation, modification and removal of definitions that define the organization of the data.
- **Update** – Insertion, modification, and deletion of the actual data.
- **Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
- **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

Advantages of a DBMS

1. Controlling redundancy in data storage and in development and maintenance efforts
2. Sharing of data among multiple users
3. Blocking unauthorized access to data
4. Providing persistent storage for program objects
5. Providing storage structures for efficient query processing
6. Enables backup and recovery services.
7. Providing multiple interfaces to different classes of users.
8. Representing complex relationships among data.
9. Enforcing integrity constraints on the database.
10. Permitting actions using active rules

DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase and IBM DB2. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS.

There are two major classes of database management systems (DBMS) popular today, namely SQL (Structured Query Language) and NoSQL (Not Only SQL) systems. NoSQL refers to a non-relational manner in which data storage and retrieval is handled, as opposed to the relational manner in which SQL based solutions approach this. A NoSQL database does not always maintain referential integrity SQL is a standard language for making interactive queries from and updating a database such as IBM's DB2, Microsoft's SQL Server, and database products from Oracle, Sybase, and Computer Associates. SQL is a programming language for Relational Databases. SQL comes as a package with all major distributions of Relational Database Management System (RDBMS).

SQL comprises both data definition and data manipulation languages. Using the data definition properties of SQL, one can design and modify database schema, whereas data manipulation properties allows SQL to store and retrieve data from database.

SQL uses the following set of commands to define database schema:

CREATE

Creates new databases, tables and views from RDBMS.

DROP

Drops commands, views, tables, and databases from RDBMS.

ALTER

Modifies database schema.

SQL is equipped with data manipulation language (DML). DML modifies the database instance by inserting, updating and deleting its data. DML is responsible for all forms data modification in a database. SQL contains the following set of commands in its DML section −

- SELECT/FROM/WHERE

- INSERT INTO/VALUES
- UPDATE/SET/WHERE
- DELETE FROM/WHERE

These basic constructs allow database programmers and users to enter data and information into the database and retrieve efficiently using a number of filter options.

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS [15].

User (Presentation) Tier − End-users operate on this tier and they know nothing about any existence of the database beyond this layer. All views are generated by applications that reside in the application tier. The presentation layer communicates with other tiers by outputting results to the browser tier and all other tiers in the network.

Application (Middle) Tier – At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

Database (Data) Tier – At this tier, the database resides along with its query processing languages. Here the information is stored and retrieved. This keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

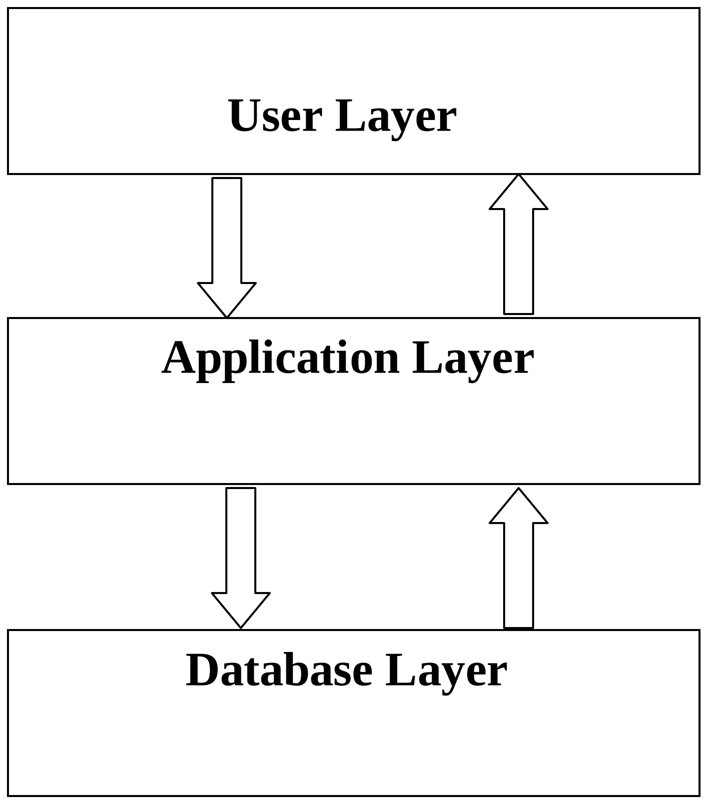


Figure 4.1: Database Architecture

In our project we had used a text file to store the positional data of the vehicle. By using AT command SIM908 module is connected through HTTP to the server and GET method is used to store the data into text file in PHP code. This data can be retrieved or fetched into the Google maps by using Google maps API to show the visual location of the vehicle.

4.5. Google Map

Google has developed a Google maps for the computer or mobile mapping service. It offers 2D map, satellite map and 360 panoramic views of streets (Street View).

Google Maps provides high-resolution satellite images. With the introduction of an easily searchable mapping and satellite imagery tool, Google's mapping engine prompted a surge of interest in satellite imagery. Sites were established which feature satellite images of interesting natural and man-made landmarks, including such novelties as "large type" writing visible in the imagery, as well as famous stadia and unique earth formations. Although Google uses the word "satellite", most of the high-resolution imagery is aerial photography taken from airplanes rather than from satellites [16].

Google Maps uses JavaScript extensively. As the Google Maps code is almost entirely JavaScript and XML, some end-users reverse-engineered the tool and produced client-side scripts and server-side hooks which allowed a user or website to introduce expanded or customized features into the Google Maps interface. For developers, Google launched the Google Maps API (Application Programming Interface) in June 2005[17]. This allows developers to integrate Google Maps into their websites and mobile apps. Start by creating an API Key, it will be bound to the web site and directory you enter when creating the key. Creating your own map interface involves adding the Google JavaScript code to your page, and then using JavaScript functions to add points to the map.

For using the Google maps in our website, we have found Google Map API key from this site **https://console.developers.google.com.** As Google provides several types of map view, we had used MapTypeId property to view ROADMAP (2D map), SATELITE (photographic map).

Chapter 5: System Design and Analysis

5.1. System Design

Vehicle tracking device is made up with Arduino Uno R3, SIM908 module including GPS and GSM antenna. The core part of tracking system is microcontroller Arduino Uno. The geo location of a vehicle can be captured through GPS receiver and that data will be transmitted to the web server by using GSM technology. That data will be stored in a database. For monitoring the location of the vehicle on the map, we had developed a web application. We made up this web application with PHP, HTML and JavaScript using XAMPP software. For storing the location data, a text file had been used. We had also developed a mobile application to view the location of vehicle in a mobile device by using Android Studio.

The SIM908 module is initialized to start gathering geo location data from the satellite; device initialization is done using AT commands and includes GPS and GSM module; to turn on the GPS, first it is powered on and put in reset mode. Then the module become ready for receiving coordinates from satellite. The GPRS is next turned on; the process includes GPRS power on, setting APN of service provider, initiating HTTP protocol, and setting protocol method (Get method). Device initialization process may take up to 1 minute to worm up and calculate the accurate position. In case of network un-availability, the acquisitioned GPS coordinates and other data such as time and speed are stored temporarily until the network returns back to service then the stored coordinates are sent with their time stamp and speed. SIM908 requires 2A peak current. So, external power supply like 12V-2A battery is used to provide the power.

GPS antenna and GSM antenna are connected to the port of SIM908 module. The module and Arduino have a common ground. We had uploaded the program into the Arduino microcontroller which program is written in C programming language. Uploading program into Arduino is done by using Arduino IDE software.

Fig 5.1: Design of System

5.2. Algorithm

At first, we did the experiment of our work with the GPS module which is connected to the Arduino Uno microcontroller to get latitude and longitude of a vehicle. For doing that, we had used Arduino IDE software to program the code. In the coding we had included two libraries softwareSerial.h and TinyGPS.h. The flowchart of GPS coding is given below.

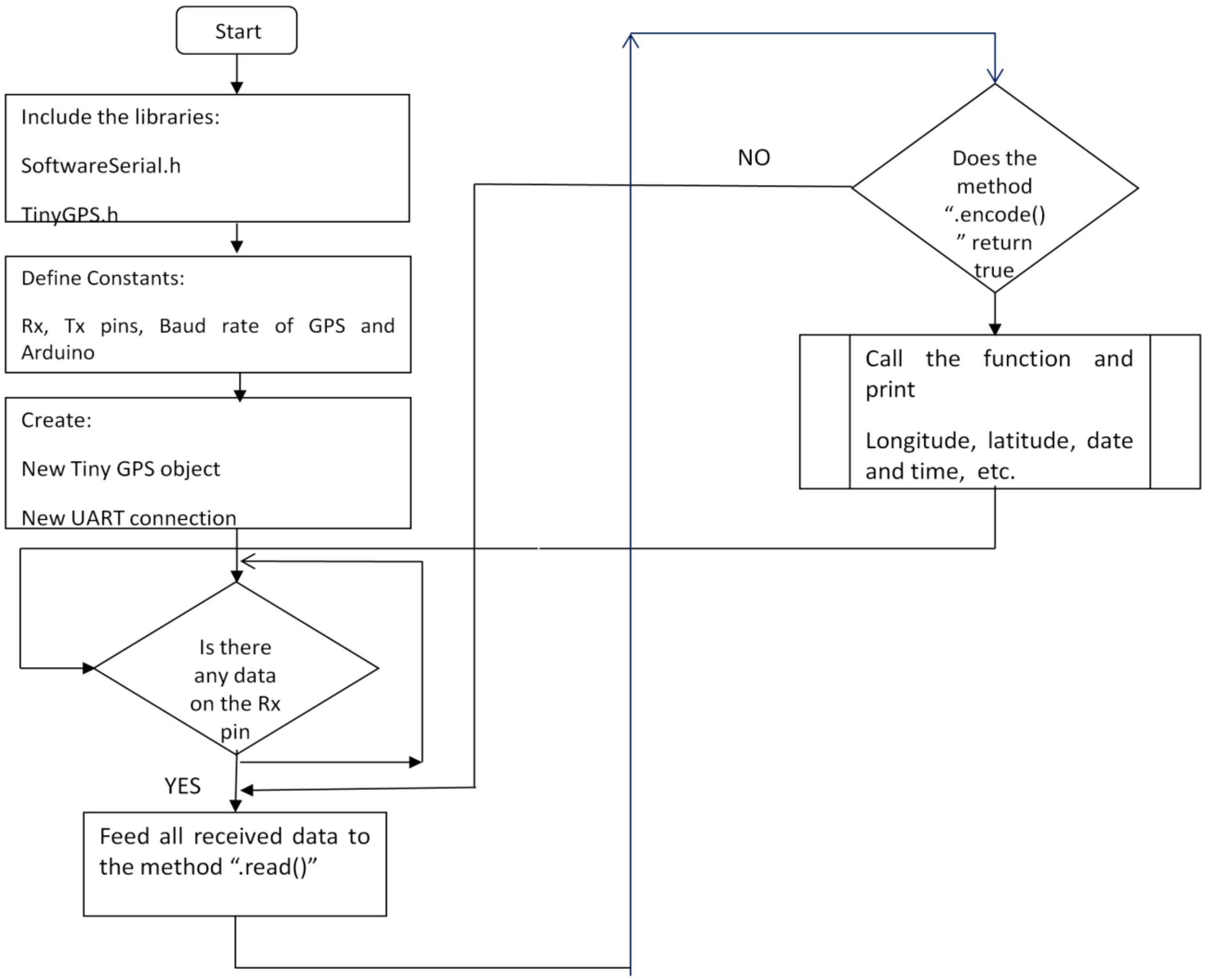


Fig 5.2: Flowchart for GPS coding

After initialization those libraries, we had created a new Tiny GPS object. If the receiver received a data, that data will be fetched into the ".read()" method. Then, if ".encode ()" method return to true, longitude, latitude data will be printed into serial monitor.

Then we have used SIM908 module with Arduino Uno microcontroller for sending location data of vehicle to the user via GSM network. For communication with mobile device and SIM908 module, AT command has been used in programming. The flowchart of GSM coding is given below.

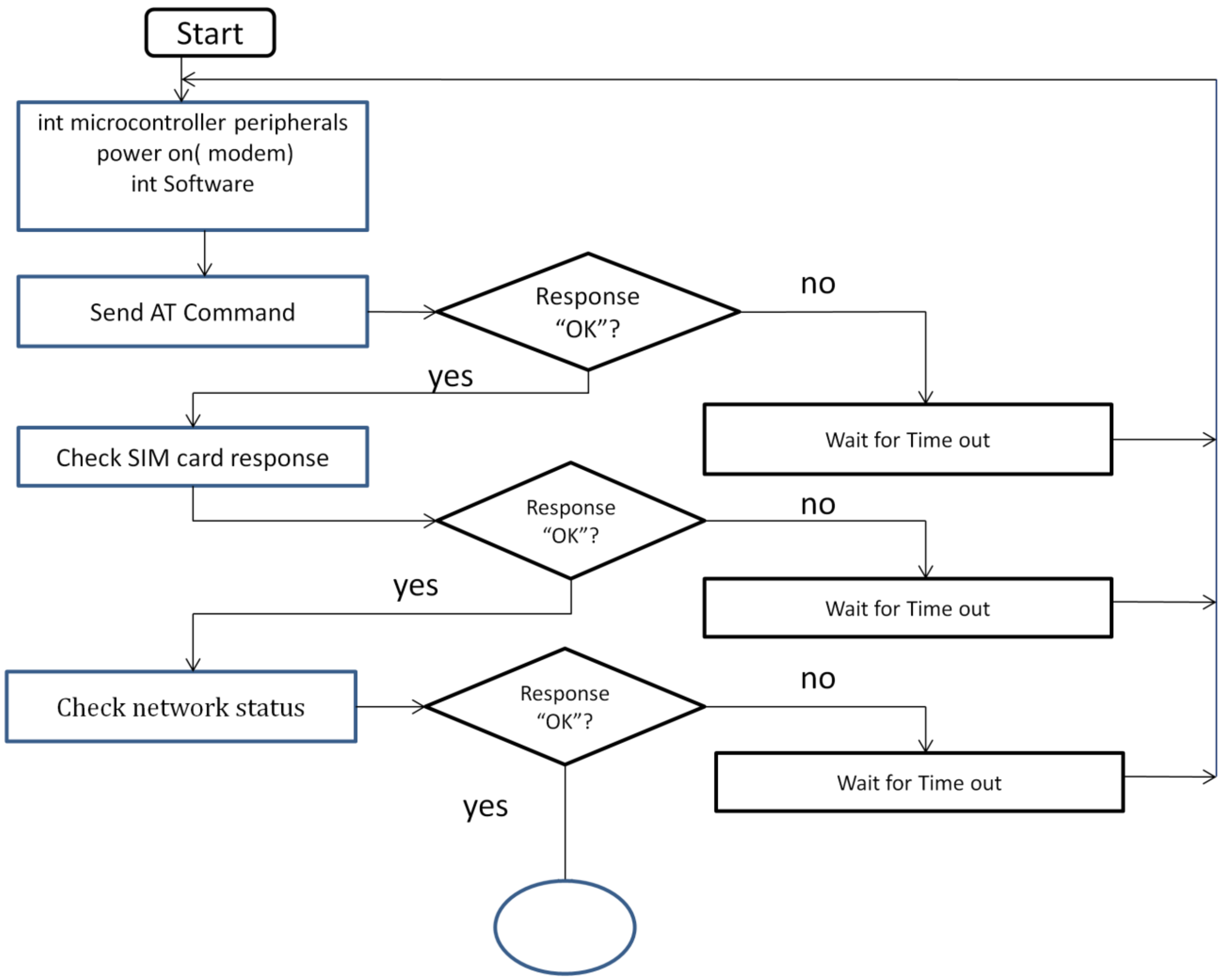


Fig 5.3: Flowchart of GSM coding

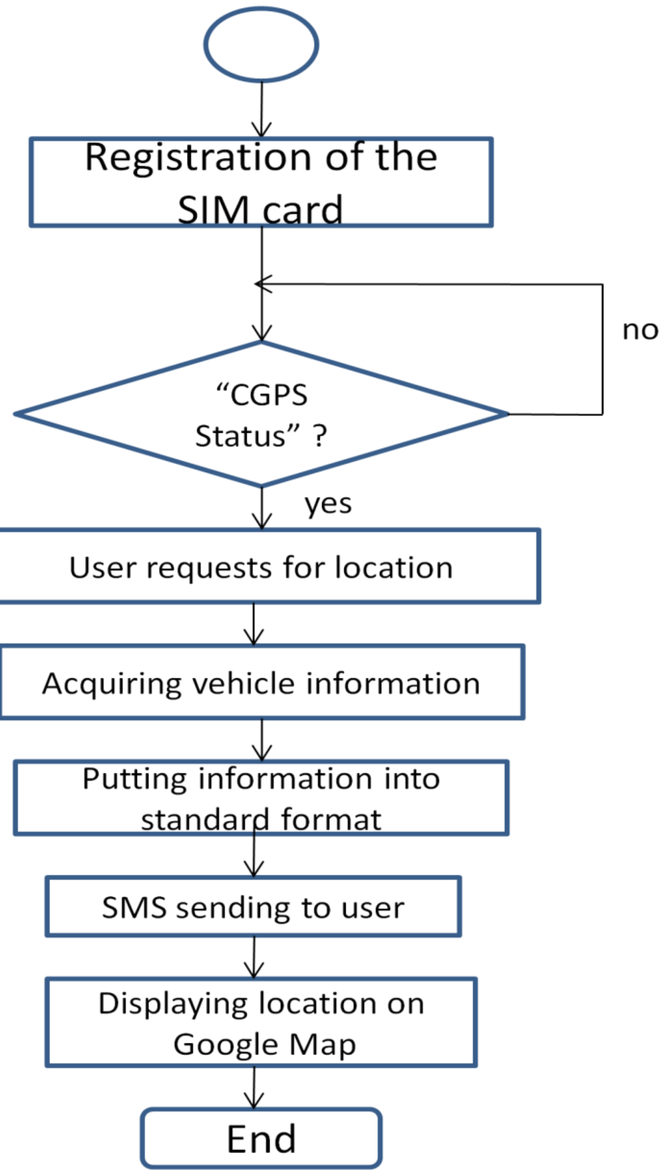


Figure 5.4: Continuation from previous coding

When SIM908 module is powered on, it sends the AT command. If the response of corresponding AT command is OK, it will check the response of SIM card. Then it will check the response of network status. After the confirmation of network status, it will check the GPS status. If the GPS receiver receives the geo location data, the user sends a SMS to the SIM908 module and the module will provide the information of location of the vehicle to the user via SMS.

For visualization of the location of vehicle into the Google map, we had developed a web application. XAMPP software is used to create a web server. PHP script, HTML is written which will show the web page into the browser. We had developed a coding for communication with SIM908 module and internet protocol. The flowchart of sending and receiving data is given below:

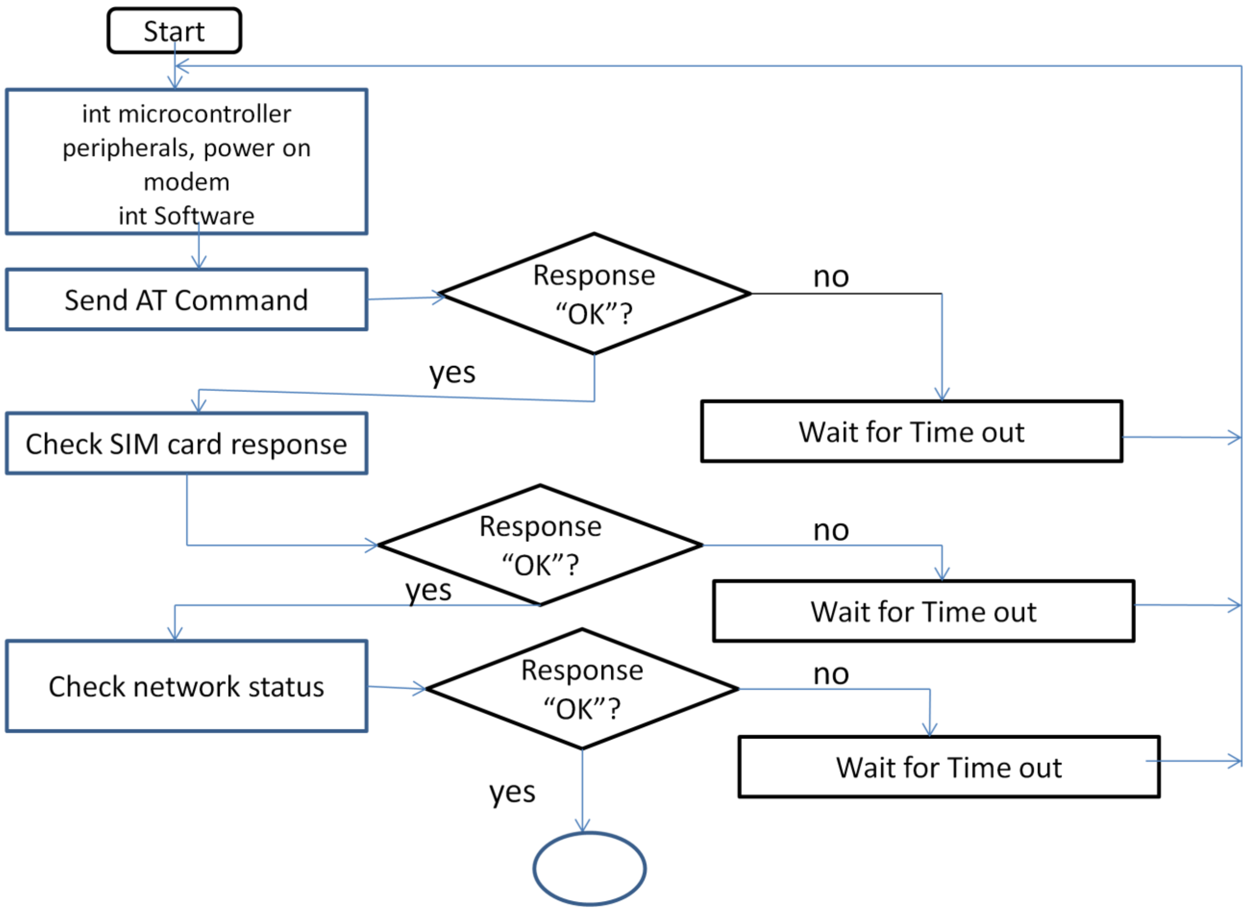


Figure 5.5: Flowchart of sending and receiving data

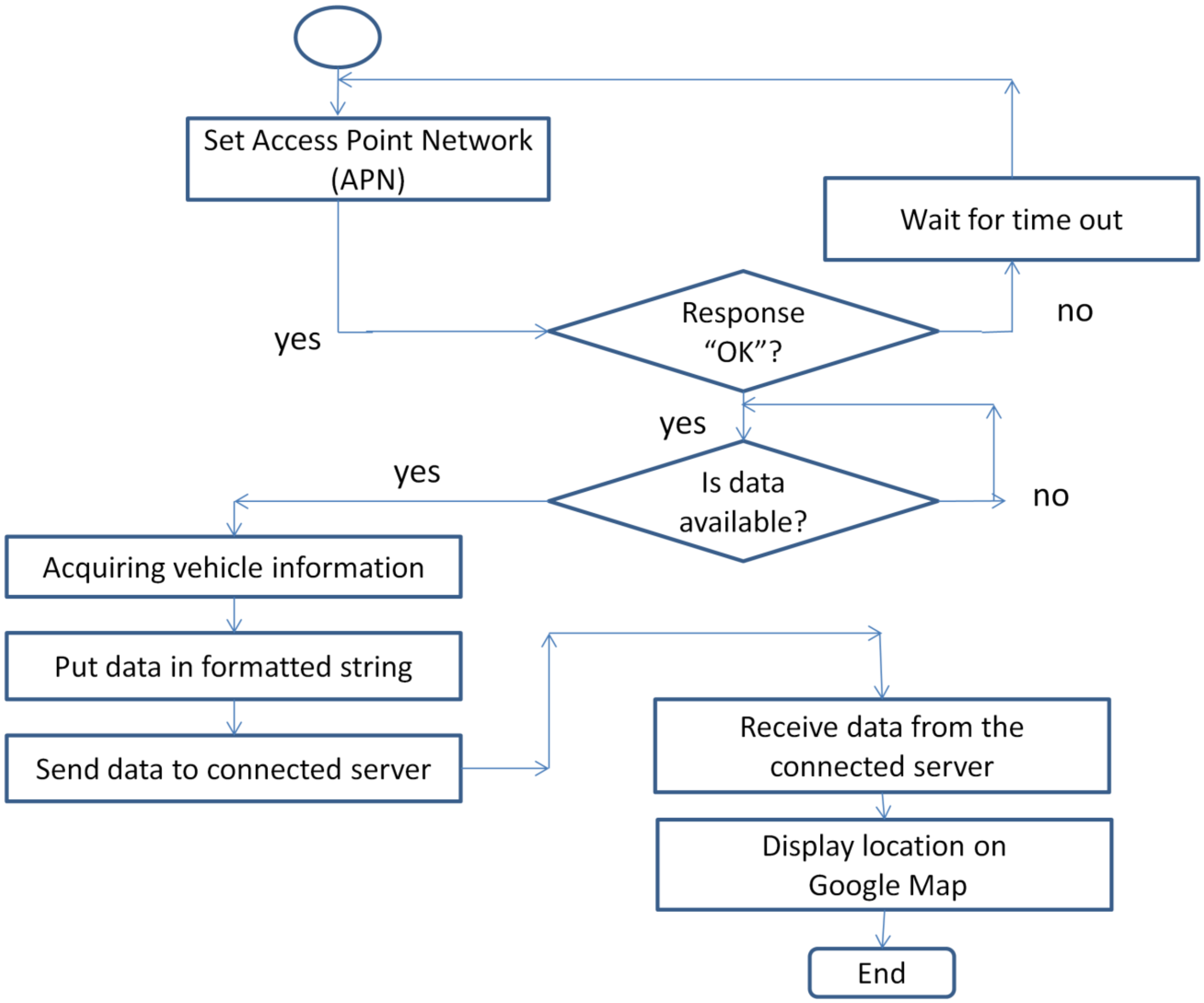


Figure 5.6: Continuation of previous coding

SIM908 module including GPS antenna and GSM antenna is connected with Arduino Uno microcontroller. That module will send AT command. If the response of AT command is OK, it will check the whole network status for confirmation of registration. After that, it will check the GPS status and acquire the GPS information of the vehicle. Then it will check the HTTP request for internet. That data will be transferred to the database of the server.

We had written the PHP script for web page. In this script Google Map API is embedded. We got the Google Map API key from **https://console.developers.google.com.** For storing the data into database, we have used a text file. By using GET and POST function in the PHP script, data has been fetched into the Google map to show the location of vehicle. The flowchart of PHP file is given below.

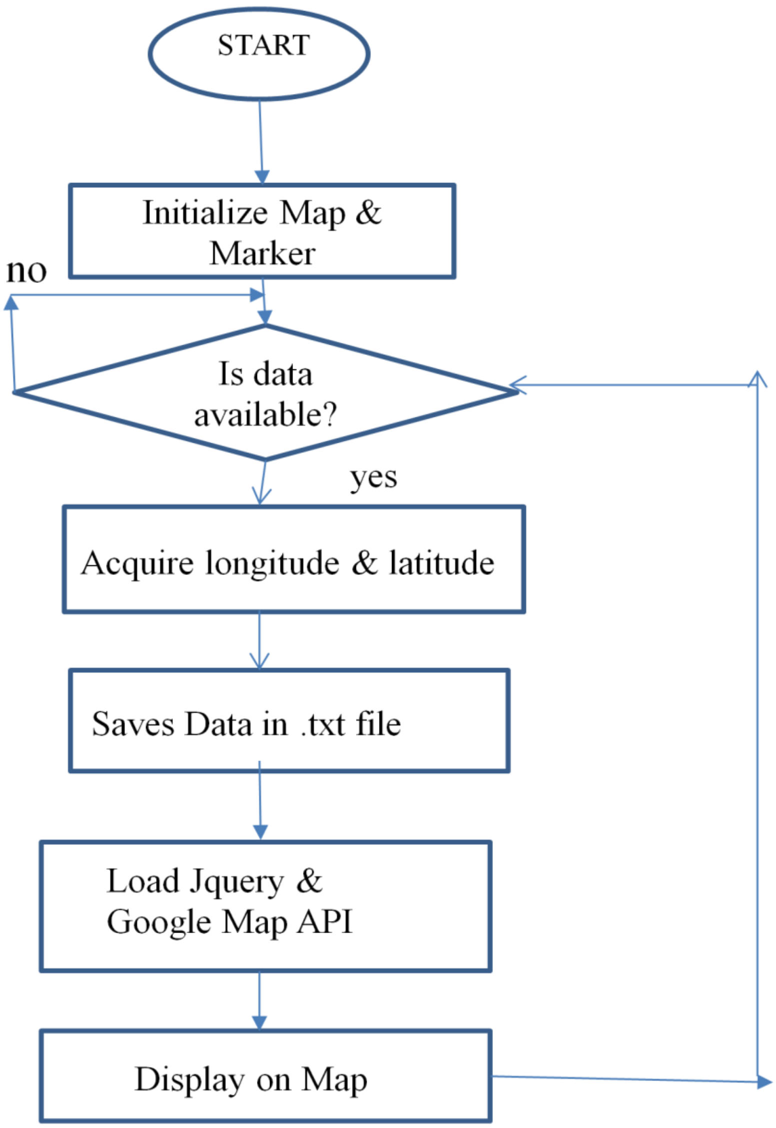


Fig 5.7: Flowchart of PHP file

In the PHP script, Map and marker have been initialized. If the PHP script gets the longitude and latitude, it will store into the text file. JQuery and Google Map API has been load into the script which will help to show the location of the vehicle into the Google Map in real time.

5.3. Working of the System

In vehicle tracking system, GPS receiver receives the location data like latitude and longitude of a vehicle and send them by using a HTTP request to web server. Then browser is used to load the PHP webpage which contain Google maps to show the location of the vehicle in real time.

At the initial stage, SIM908 module has been powered up by using 12V battery rather than using 5V from Arduino. While the SIM908 module draws 2A peak current, a huge voltage drop across it and then SIM908 module has automatically been shut down. Therefore we need to use the external source to provide power to the system. Moreover, as our tracking system is used for the car, so we can use the car battery to provide power to the system.

SIM908 module is run by using AT command in the program of Arduino. Initially, the network registration is done by using AT+CREG and set Access Point Name (APN), user name and password. After that turn on the GPS power supply using the command CGPSPWR and current GPS location information can be gotten using AT+CGPSINF command.

After getting the data of vehicle’s location we have used two methods to send the data to the user end. If the user or owner of the vehicle sends a SMS to the mobile number which SIM card is used into the SIM908 module, it will continuously send the SMS to the user’s mobile of the current location. Another way, by using HTTP request to web server the data has been sent. For sending SMS, AT+CMGS command is used in the program and AT+CMGR command is used to read the SMS which has been sent from the user. For setting the configuration of GPRS, AT+SAPBR command has been used. To access the internet HTTP service, AT+HTTPINIT is used to initialize HTTP service, HTTPPARA command is used to set the parameters value.

We had developed a web application to view the location of the vehicle in the map in our working system. HTML is written to show the web page in the browser including with PHP and JavaScript for making the web page more dynamic. In the PHP script, it contains Google embedded map that shows the map and given parameters to the user. Google Map API key is used for embedding the map into the PHP script. The script is also used to handle POST and GET functions to fetch or store the data into database.

5.4. Display of Result

5.4.1. Google Map

With the Google Maps API, we add maps based on Google Maps data to the web application. The API automatically handles access to Google Maps servers. We also use API calls to add markers. We set the center location of map by using google.maps.LatLng (). As Google Maps API provides different kind of maps view, we used ROADMAP for this web application. For fetching the data into the Google Map, we have used getElementByID () method.

5.4.2. SMS

For monitoring the vehicle location, we have also included the feature which will send SMS to the user according to user request. SMS will be included the value of latitude and longitude of the vehicle. A link is also attached with the SMS, so that the user can see the location by using Google map.

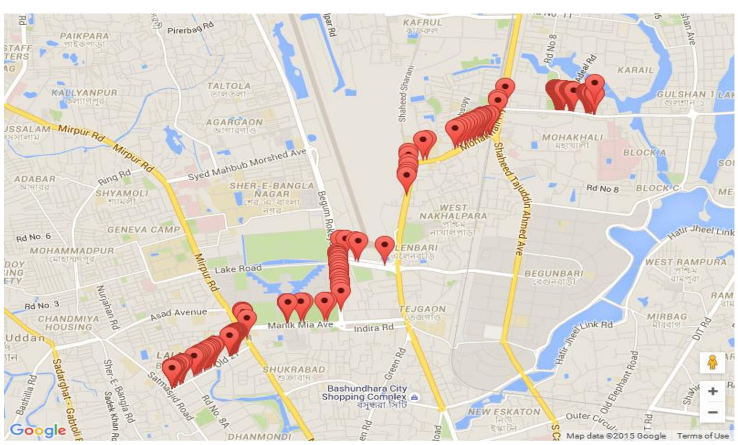
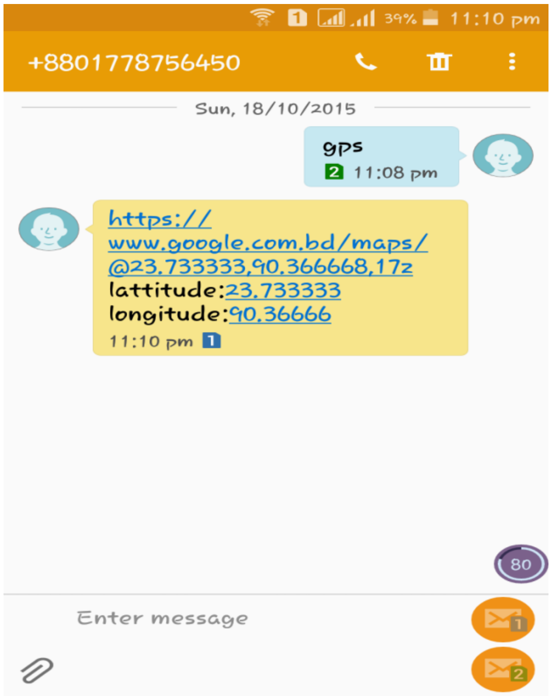


Fig 5.8: View of location on website through Google map

The SMS received by the user and view on Google Map:



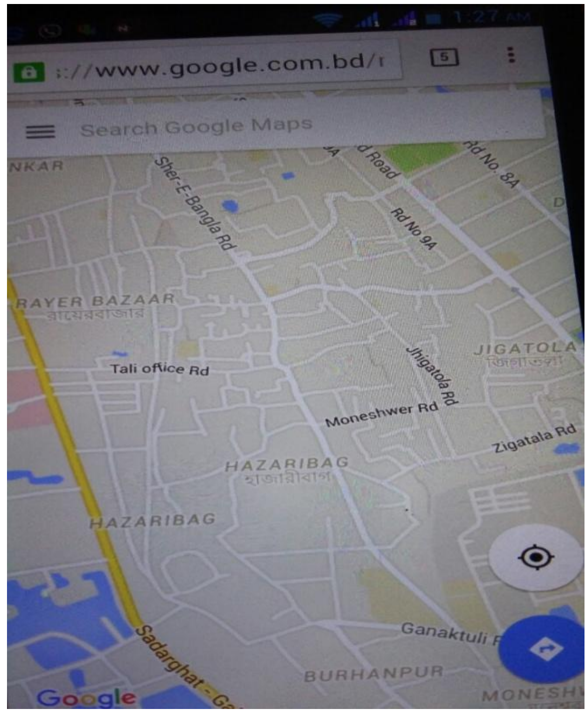


Fig. 5.9.a: SMS with a Google Map link

Fig.5.9.b: View of the location

5.4.3. Mobile Application synchronized with the Web page:

Mobile application has been built up by using Android Studio Software. This app is actually embedded with the web site so that user can comfortably view the map into mobile device. Android application is built with Java programming language. In Android Studio there is different kind of API for making the application. In the coding, MyActivity class we have created a new object called WebView. For getting the permission of using internet connection, we had to include permission in androidmanifest.xml file.

<uses-permission android:name=”android.permission.INTERNET”/>

We included our web page URL into MyActivity class.

mWebView.loadUrl("http://115.127.80.43/gps/")

For opening the web page into mobile application instead of browser, we had included new class MyAppWebViewClient. The below figure show the maps of vehicle location using mobile application.

Fig 5.10: View of location on Mobile Application

Chapter 6: Result Analysis

6.1 Data Analysis

The more detail location view on the Google Map is given below.

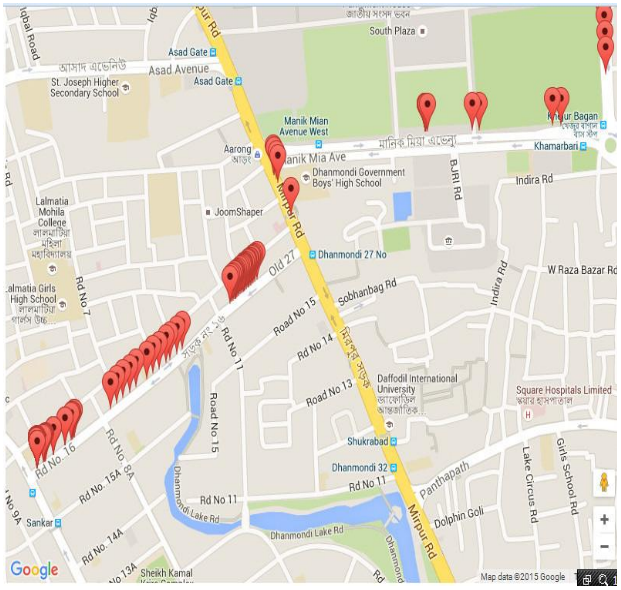


Fig.6.1a. Dhanmondi to Manik Mia Avenue

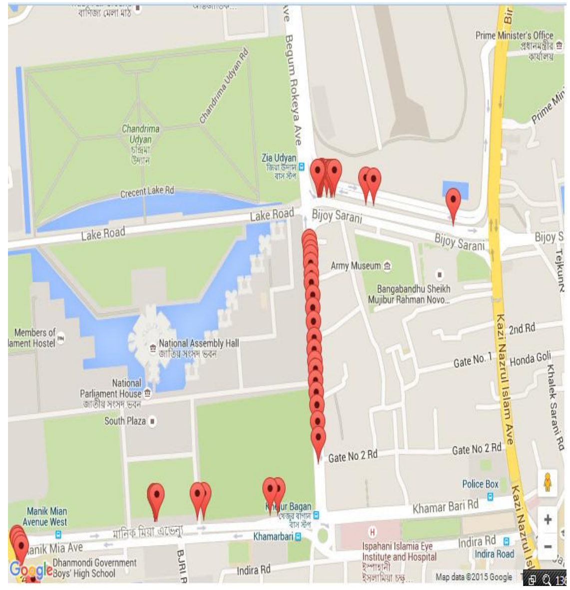


Fig.6.1b Manik Mia Avenue to Bijoy Sarani

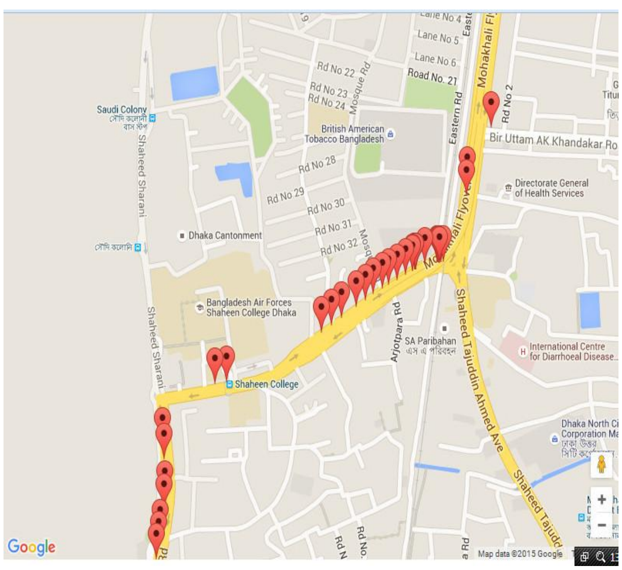


Fig. 6.1c Bijoy Sarani to Jahangir Gate

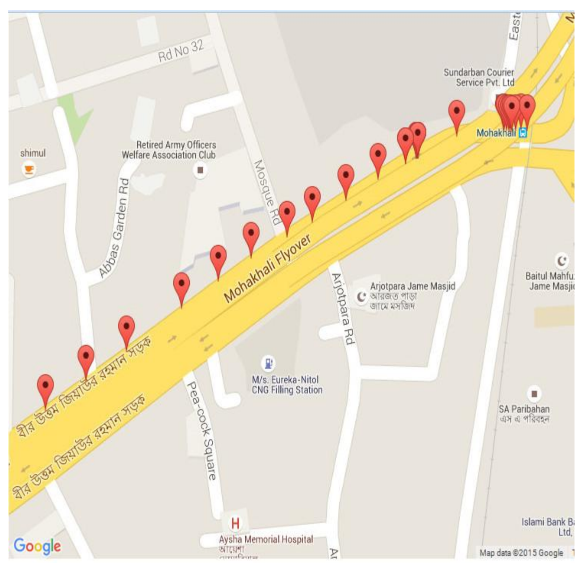


Fig. 6.1d Jahangir Gate to Mohakhali Flyover (Railgate)

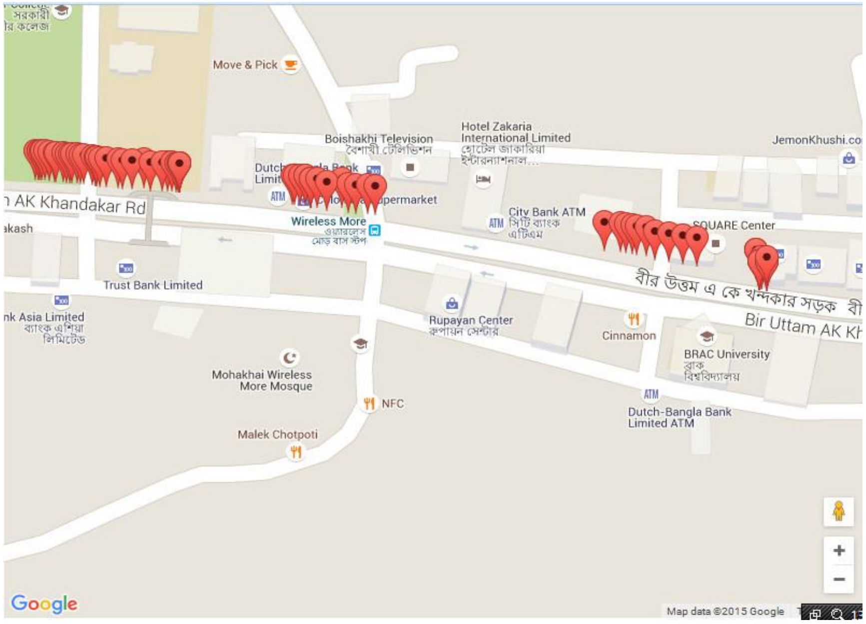


Fig.6.d Mohakhali Railgate to BRAC University

6.2 Delay Analysis:

While testing the whole setup we also observed the delay analysis. That is we observed the time it takes for the in-vehicle device to reply or send an SMS in response to end user’s demand. For such observation of this delay analysis, we used SIM cards of four different operators and recorded the delay for each of the corresponding SIM cards in pair. Four different operator SIM cards that we have used are Banglalink, Grameenphone, Robi and Airtel. Different sets of data are recorded and corresponding average time is considered to study the delay analysis. The results are shown below:

- **SIM in GSM module (fixed):** Banglalink

SIM in GSM Module	SIM in user end	Delay
Banglalink	Banglalink	1 minute 49 seconds
Banglalink	Robi	1 minute 52 seconds
Banglalink	Grameenphone	1 minute 55 seconds
Banglalink	Airtel	2 minutes 10 seconds

Table: 6.1a Delay Analysis

- **SIM in GSM Module (fixed):** Grameenphone

SIM in GSM Module	SIM in user end	Delay
Grameenphone	Grameenphone	1 minutes 58 seconds
Grameenphone	Robi	2 minutes 7 seconds
Grameenphone	Airtel	2 minutes 9 seconds
Grameenphone	Banglalink	2 minutes 14 seconds

Table: 6.1b Delay Analysis

- **SIM in GSM Module (fixed):** Airtel

SIM in GSM Module	SIM in user end	Delay
Airtel	Airtel	1 minute 26 seconds
Airtel	Grameenphone	1 minute 30 seconds
Airtel	Banglalink	1 minute 37 seconds
Airtel	Robi	1 minute 41 seconds

Table: 6.1c Delay Analysis

- **SIM in GSM Module (fixed):** Robi

SIM in GSM Module	SIM in user end	Delay
Robi	Robi	1 minute 36 seconds
Robi	Banglalink	1 minute 37 seconds
Robi	Airtel	1 minute 41 seconds
Robi	Grameenphone	1 minute 42 seconds

Table: 6.1d Delay Analysis

From the above analysis, it is observed that there is nearly 1.5 minutes to 2 minutes delay for each set of data of different operator and this time delay seem to be less significant. Network dependency on the GSM service provider of this GSM based technology is one of reasons for such delay. This is because while the vehicle is traveling, it passes through certain areas of poor network coverage.

6.3. Benefits of Vehicle Tracking System:

The in-vehicle tracking device or unit working along with a central server and a software, which let the user or owner of a car to know the whereabouts of his own vehicle, surely comes with several benefits. The GPS and GSM installed inside the vehicle fetches its location information and send it to owner on regular intervals according to user's preferences, in order to remain up-to- date all the time. As all the relevant information is displayed on the screen, it is very convenient for the user to monitor and take any actions in case of an emergency.

Also monitoring discourages dangerous and inefficient driving practices of drivers which lead to increased vehicle security and driver safety [18]. The vehicle tracking system plays a vital role if it is used in any companies or organization for any kind of delivery purposes. Since the driver is being aware of the fact that the car is constantly being monitored so one would be careful while driving and take shortest possible route to reach destination right on time [19]. This system can also be named as an anti-theft tracking system as this advanced yet affordable system ensures the recovery of stolen vehicles too. If the car does not get to designated location or being used by unauthorized user, the location can be traced and then notified to police to reach the unauthorized location where the vehicle is residing and thus this vehicle tracking system ensures car safety as well [18].

6.4. Limitations:

While this advanced technology based tracking system can benefit users, company or any organization, there are also some limitations to using this vehicle tracking devices. Often GPS takes time to connect with the network due to poor weather conditions. For the GPS to work properly, it needs to have a clear view of the sky. That is it is unlikely to work indoor or may even have problem outside where it has no clear path of transmitting to and receiving signal from satellites. Therefore, due to obstacles like tall buildings or such infrastructure which block view of the sky, often causes multipath error to the receiving signal of the GPS receiver. As a result, location seems to appear to jump from one place to another leading to inaccurate results. Thus incorrect values of latitude and longitude are sent to the server, for displaying in the Google map on error being initialized.

Chapter 7: Conclusion and Future work

7.1. Conclusion

In our thesis we have developed a vehicle tracking system that is flexible, customizable and accurate. The GSM modem was configured and we tested and implemented the tracking system to monitor the vehicle's location via SMS and online on Google map. To display the position on Google map we have used Google map API. The Arduino is the brain of the system and the GSM modem is controlled by AT commands that enable data transmission over GSM network while the GPS provide the location data. Whenever the GPS receives a new data it is updated in the database and hence we are able to see the location on the Google map. We thought of designing a real time vehicle tracking system in our thesis keeping the scenario of Bangladesh in mind where vehicle theft is rapidly increasing. Our device can provide good control on carjacking. The system provides accurate data in real time that makes it possible for the user to track the vehicle and it also enable an early retrieval if the car is stolen. Implementation of GPS tracker in vehicle can certainly bring revolutionary change in developing country like Bangladesh where there is very high urban as well as rural vehicular transition every day. There can be various other applications that can be built over our existing platform. Hence, we have designed our system in such a way that upgrading this system is very easy which makes it open for future requirement without the need of rebuilding everything from scratch, which makes our system even more efficient. This thesis has widely increased our knowledge of GPS and also improved our programming skills. We have also ensured the reliability of our system through various field tests that we have done during our thesis and the initial results that we obtained through our prototype are very promising. This makes our thesis complete, robust and we can even think of commercialization of this system in future.

7.2. Scope of development:

Our aim is to implement this tracking device into more 15 units. Basically, this one is a pilot project of a bigger one; which BRAC Health will be assisting in the IDCOL (Infrastructure Development Company Limited) PROJECT which has been proposed with the aim to implement 15 units. With the succession of each unit being manufactured properly and the quality of service it provides after completion of manufacturing, we will go for mass production. That is 15 units will be installed with this tracking device in order to monitor or track down each one of its location simultaneously. The features are actually not only limited to what we added but also more technical features can be developed to enhance this tracking purpose. For example, a camera can be placed inside the car along with this tracing device, with the aim to protect the vehicle from unauthorized users. On installing a camera inside, it will enable to have photos of rightful owner along with audio and video, with the photos being sent over the M2M link. Even the videos or pictures of current location of the vehicle can be sent using such technology to track down the vehicle, if it gets to an unauthorized user or to an unauthorized location. The mobile application that we have done in synchronization with the web page can be further modified by including more advanced features with existing ones. Therefore, a wide range of applications can be implemented to make this tracking system a more desirable and beneficial as well.

References:

[1] “TRACKING SYSTEM USING GSM, GPS & ARM7” by ASHUTOSH UPADHAYA ,SAMIR BOTHRA, RASHMI SINGH, SHIVANSHU GUPTA

[2] http://www.teletrac.com/fleet-management/topics/history-gps-tracking

[3] http://www.fleetistics.com/history-gps-satellites.php

[4]- A Review on GSM and GPS Based Vehicle Tracking System by Dinesh Suresh Bhadane, PritamB.Bharati, Sanjeev A.Shukla, MonaliD.Wani, KishorK.Ambekar.

[5]https://forums.oneplus.net/threads/top-5-features-of-arduino-uno-r3-my-first-article-here.330321/

[6] https://www.arduino.cc/en/Main/ArduinoBoardUno

[7] http://www.gammon.com.au/power

[8] http://www.sabreadv.com/simcom-gsm-sim908/

[9]http://www.simcom.ee/modules/gsm-gprs-gps/sim908/

[10]http://web.ics.purdue.edu/~ecalais/teaching/geodesy/GPS_observables.pdf

[11] http://www.rfsolutions.co.uk/acatalog/info_ANT_GHEL2R_SMA.html

[12] http://php.net/

[13] http://www.w3schools.com/jquery/default.asp

[14] Fundamental of Database System 5th edition- Elmasri/ Navathe

[15]http://www.tutorialspoint.com/dbms/sql_overview.htm

[16] https://en.wikipedia.org/wiki/GoogleMaps#Google_Maps_API.

[17] "Google Maps Tutorial Article," [Online]: Available:https://developers.google.com/maps/articles/phpsqlsearch_v3?csw=1#createtable .

[18] http://www.trackcompare.co.uk/benefits-of-tracking-systems

[19] http://www.globefleet.com/en/aboutus/benefits-of-vehicle-tracking-system.html

Appendix-

C-programming code of Arduino:

```
#include <SoftwareSerial.h>

#include <TinyGPS.h>

/* This sample code demonstrates the normal use of a TinyGPS object.

   It requires the use of SoftwareSerial, and assumes that you have a

   4800-baud serial GPS device hooked up on pins 4(rx) and 3(tx).

*/

TinyGPS gps;

SoftwareSerial ss(0, 1);

static void smartdelay(unsigned long ms);

static void print_float(float val, float invalid, int len, int prec);

static void print_int(unsigned long val, unsigned long invalid, int len);

static void print_date(TinyGPS &gps);

static void print_str(const char *str, int len);

void setup()

{

  Serial.begin(9600);

Serial.print("Testing TinyGPS library v. "); Serial.println(TinyGPS::library_version());

  Serial.println("by Mikal Hart");

  Serial.println();

  Serial.println("Sats HDOP Latitude  Longitude  Fix  Date       Time     Date Alt    Course Speed
Card  Distance Course Card  Chars Sentences Checksum");

  Serial.println("          (deg)     (deg)      Age                      Age  (m)    --- from GPS ----  ---- to
London  ----  RX    RX        Fail");
```

```
  Serial.println("--------------------------------------------------------------------------------------------------------------------------------");
  ss.begin(9600);
}
void loop()
{
  float flat, flon;
  unsigned long age, date, time, chars = 0;
  unsigned short sentences = 0, failed = 0;
  static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;

  print_int(gps.satellites(), TinyGPS::GPS_INVALID_SATELLITES, 5);
  print_int(gps.hdop(), TinyGPS::GPS_INVALID_HDOP, 5);
  gps.f_get_position(&flat, &flon, &age);
  print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6);
 print_float(flon, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);
  print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
  print_date(gps);
  print_float(gps.f_altitude(), TinyGPS::GPS_INVALID_F_ALTITUDE, 7, 2);
  print_float(gps.f_course(), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
  print_float(gps.f_speed_kmph(), TinyGPS::GPS_INVALID_F_SPEED, 6, 2);
  print_str(gps.f_course() == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " : TinyGPS::cardinal(gps.f_course()), 6);
print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned long)TinyGPS::distance_between(flat, flon, LONDON_LAT, LONDON_LON) / 1000, 0xFFFFFFFF, 9);
```

```
  print_float(flat == TinyGPS::GPS_INVALID_F_ANGLE ?
TinyGPS::GPS_INVALID_F_ANGLE : TinyGPS::course_to(flat, flon, LONDON_LAT,
LONDON_LON), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);

  print_str(flat == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " :
TinyGPS::cardinal(TinyGPS::course_to(flat, flon, LONDON_LAT, LONDON_LON)), 6);

  gps.stats(&chars, &sentences, &failed);

  print_int(chars, 0xFFFFFFFF, 6);

  print_int(sentences, 0xFFFFFFFF, 10);

  print_int(failed, 0xFFFFFFFF, 9);

  Serial.println();


  smartdelay(1000);

}

static void smartdelay(unsigned long ms)

{

  unsigned long start = millis();

  do

  {

    while (ss.available())

      gps.encode(ss.read());

  } while (millis() - start < ms);

}

static void print_float(float val, float invalid, int len, int prec)

{

  if (val == invalid)

  {
```

```
    while (len-- > 1)
      Serial.print('*');
    Serial.print(' ');
  }
  else
  {
    Serial.print(val, prec);
    int vi = abs((int)val);
    int flen = prec + (val < 0.0 ? 2 : 1); // . and -
    flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
    for (int i=flen; i<len; ++i)
      Serial.print(' ');
  }
  smartdelay(0);
}
static void print_int(unsigned long val, unsigned long invalid, int len)
{
  char sz[32];
  if (val == invalid)
    strcpy(sz, "*******");
  else
    sprintf(sz, "%ld", val);
  sz[len] = 0;
  for (int i=strlen(sz); i<len; ++i)
    sz[i] = ' ';
```

```
if (len > 0)
    sz[len-1] = ' ';
  Serial.print(sz);
  smartdelay(0);
}
static void print_date(TinyGPS &gps)
{
  int year;
  byte month, day, hour, minute, second, hundredths;
  unsigned long age;
  gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
  if (age == TinyGPS::GPS_INVALID_AGE)
    Serial.print("********** ******** ");
  else
  {
    char sz[32];
    sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
        month, day, year, hour, minute, second);
    Serial.print(sz);
  }
  print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
  smartdelay(0);
}
static void print_str(const char *str, int len)
```

```
{
  int slen = strlen(str);
  for (int i=0; i<len; ++i)
    Serial.print(i<slen ? str[i] : ' ');
  smartdelay(0);
}
```

C-programming code of Arduino for GSM:

```
int8_t answer;

int onModulePin= 2;

int counter;

char aux_str[150];

char aux;

char frame[200];

char latitude[15];

char longitude[15];

char altitude[6];

char date[16];

char time[7];

char satellites[3];

char speedOTG[10];

char course[10];

int x = 0;

char N_S,W_E;

void setup()

{


   //pinMode(onModulePin, OUTPUT);

   Serial.begin(115200);


   delay(3000);

   delay(3000);
```

```
  Serial.println("Starting...");
  delay(30000);
  while (sendATcommand("AT+CREG?", "+CREG: 0,1", 2000) == 0);
  // sets APN , user name and password
  sendATcommand("AT+SAPBR=3,1,\"Contype\",\"GPRS\"", "OK", 2000);
  sendATcommand("AT+SAPBR=3,1,\"APN\",\"gpinternet\"", "OK", 2000);
  sendATcommand("AT+SAPBR=3,1,\"USER\",\"\"", "OK", 2000);
  sendATcommand("AT+SAPBR=3,1,\"PWD\",\"\"", "OK", 2000);
// gets the GPRS bearer
   while (sendATcommand("AT+SAPBR=1,1", "OK", 20000) == 0)
  {
    delay(5000);
  }



  delay(3000);

 // sets APN, user name and password
//   sendATcommand("AT+CGPSPWR=1", "OK", 2000);
//   sendATcommand("AT+CGPSRST=0", "OK", 2000);

  while( (sendATcommand("AT+CGPSSTATUS?", "2D Fix", 5000) ||
        sendATcommand("AT+CGPSSTATUS?", "3D Fix", 5000) ||
        sendATcommand("AT+CGPSSTATUS?", "Unknown", 5000)) == 0 );
```

```
}
int8_t get_GPS()
{
   int8_t counter, answer;
   long previous;
   // First get the NMEA string
   // Clean the input buffer
   while( Serial. Available() > 0) Serial. Read();
   // request Basic string
   sendATcommand("AT+CGPSINF=0", "AT+CGPSINF=0\r\n\r\n", 2000);
   counter = 0;
   answer = 0;
   memset(frame, '\0', 100);    // Initialize the string
   previous = millis();
// this loop waits for the NMEA string
do
{
      if(Serial.available() != 0){
         frame[counter] = Serial.read();
         counter++;
         // check if the desired answer is in the response of the module
         if (strstr(frame, "OK") != NULL)
         {
            answer = 1;
         }
```

```
        }
        // Waits for the asnwer with time out
    }
    while((answer == 0) && ((millis() - previous) < 2000));
    frame[counter-3] = '\0';     \\ '\0' where it finds the token you supplied, and
                                        returns a pointer to the start of the string.
// Parses the string
    strtok(frame, ",");
    strcpy(longitude,strtok(NULL, ",")); // Gets longitude
    strcpy(latitude,strtok(NULL, ",")); // Gets latitude
    strcpy(altitude,strtok(NULL, ".")); // Gets altitude
    strtok(NULL, ",");
    strcpy(date,strtok(NULL, ".")); // Gets date
    strtok(NULL, ",");
    strtok(NULL, ",");
    strcpy(satellites,strtok(NULL, ",")); // Gets satellites
    strcpy(speedOTG,strtok(NULL, ",")); // Gets speed over ground.
  return answer;
    // convert string to integer and add it to final float variable
deg = atof(aux);
    strcpy(aux, strtok(NULL, '\0'));
    minutes=atof(aux);
    if (deg < 100)
    {
        minutes += deg;
```

```
        deg = 0;
    }
    else
    {
        minutes += int(deg) % 100;
        deg = int(deg) / 100;
    }

    if( deg < 0 )
    {
        neg = true;
        deg*=-1;
    }
void loop()
{

 delay(1000);
delay(1000);
 Serial.print("AT+CMGF=1\r");              // AT command to select SMS format.
 delay(100);
 Serial.println("AT + CMGS = \"+880XXXXXXXXXX\"");   // AT command to send SMS
                                                       to recipient's mobile number,
                                                        in international format.
delay(100);
 Serial.println("Here is my latitude and longitude;");
```

```
 Serial.println(latitude);
 Serial.println(longitude);  // message to send

   delay(100);
 Serial.println((char)26);                       // End AT command with a ^Z, ASCII code 26
  delay(100);
 Serial.println();
  delay(5000);
}
// Clean the input buffer
    while( Serial.available() > 0) Serial.read();

    Serial.println(ATcommand);    // Send the AT command
    x = 0;
    previous = millis();

    // this loop waits for the answer
    do
{
        if(Serial.available() != 0){
            response[x] = Serial.read();
            x++;
            // check if the desired answer is in the response of the module
            if (strstr(response, expected_answer1) != NULL)
            {
```

```
            answer = 1;
          }
        }
        // Waits for the asnwer with time out
    }while((answer == 0) && ((millis() - previous) < timeout));
  //response[x] = 0;
  //Serial.println(response);
    return answer;
}
```

Code of Receiving and Sending Data in Database:

```
 int8_t answer;
int onModulePin= 2;
int counter;
long previous;

//http://gprsdata.orgfree.com/repostdata.php?latitude=0.000000&longiwude@0.0<0000

//char frame[200];
//#include <dht.h>
//#define dht_dpin A1
//dht DHT;
//int sensorValue;

//
//int co2;
//int temp;
//int humidity;
//int loudness;
//int co;

char aux_str[150];
char aux;
char co3[50];
//int x = 0;
```

```
char frame[200];


char lattitude[15];

char longitude[15];

int x = 0;

char N_S,W_E;

void setup()

{


    //pinMode(onModulePin, OUTPUT);

    Serial.begin(115200);


   // delay(3000);

    //delay(3000);


    Serial.println("Starting...");

////    power_on();


    delay(6000);


    // sets the PIN code

////    sendATcommand("AT+CPIN=****", "OK", 2000);
```

```
    // sets APN , user name and password
    sendATcommand("AT+SAPBR=3,1,\"Contype\",\"GPRS\"", "OK", 2000);
    sendATcommand("AT+SAPBR=3,1,\"APN\",\"gpinternet\"", "OK", 2000);
    sendATcommand("AT+SAPBR=3,1,\"USER\",\"\"", "OK", 2000);
    sendATcommand("AT+SAPBR=3,1,\"PWD\",\"\"", "OK", 2000);

    // gets the GPRS bearer
    {
      delay(5000);
    }

   // delay(3000);

    delay(3000);

   // sets APN, user name and password
    sendATcommand("AT+CGPSPWR=1", "OK", 2000);
    sendATcommand("AT+CGPSRST=0", "OK", 2000);

    while( (sendATcommand("AT+CGPSSTATUS?", "2D Fix", 5000) ||
           sendATcommand("AT+CGPSSTATUS?", "3D Fix", 5000) ||
           sendATcommand("AT+CGPSSTATUS?", "Unknown", 5000)) == 0 );

}
```

```
int8_t get_GPS(){

    int8_t counter, answer;
    long previous;

    // First get the NMEA string
    // Clean the input buffer
    while( Serial.available() > 0) Serial.read();
    // request Basic string
    sendATcommand("AT+CGPSINF=0", "AT+CGPSINF=0\r\n\r\n", 2000);

    counter = 0;
    answer = 0;
    memset(frame, '\0', 100);    // Initialize the string
    previous = millis();
    // this loop waits for the NMEA string
    do{

        if(Serial.available() != 0){
            frame[counter] = Serial.read();
            counter++;
            // check if the desired answer is in the response of the module
            if (strstr(frame, "OK") != NULL)
```

```
        {
          answer = 1;
        }
      }
      // Waits for the asnwer with time out
    }
    while((answer == 0) && ((millis() - previous) < 2000));

    frame[counter-3] = '\0';

    // Parses the string
    strtok(frame, ",");
    strcpy(longitude,strtok(NULL, ",")); // Gets longitude
    strcpy(lattitude,strtok(NULL, ",")); // Gets latitude
   // strcpy(altitude,strtok(NULL, ".")); // Gets altitude
    strtok(NULL, ",");
    //strcpy(date,strtok(NULL, ".")); // Gets date
    strtok(NULL, ",");
    strtok(NULL, ",");
      //strcpy(speedOTG,strtok(NULL, ",")); // Gets speed over ground. Unit is knots.
    //strcpy(course,strtok(NULL, "\r")); // Gets course

    convert2Degrees(lattitude);
    convert2Degrees(longitude);
```

```
    return answer;
}

int8_t convert2Degrees(char* input){

    float deg;
    float minutes;
    boolean neg = false;

    //auxiliar variable
    char aux[10];

    if (input[0] == '-')
    {
        neg = true;
        strcpy(aux, strtok(input+1, "."));

    }
    else
    {
        strcpy(aux, strtok(input, "."));
    }

    // convert string to integer and add it to final float variable
    deg = atof(aux);
```

```
strcpy(aux, strtok(NULL, '\0'));
minutes=atof(aux);
minutes/=1000000;
if (deg < 100)
{
    minutes += deg;
    deg = 0;
}
else
{
    minutes += int(deg) % 100;
    deg = int(deg) / 100;
}

// add minutes to degrees
deg=deg+minutes/60;


if (neg == true)
{
    deg*=-1.0;
}

neg = false;
```

```
if( deg < 0 ){
    neg = true;
    deg*=-1;
}



while(1){
    size=size+1;
    cifra=numero%10;
    numero=numero/10;
    parteEntera[size-1]=cifra;
    if (numero==0){
        break;
    }
}
    input[indice]=parteEntera[i]+'0';
    indice++;
}

input[indice]='.';
indice++;

numeroFloat=(numeroFloat-(int)numeroFloat);
```

```
    for (int i=1; i<=6 ; i++)
    {
        numeroFloat=numeroFloat*10;
        cifra= (long)numeroFloat;
        numeroFloat=numeroFloat-cifra;
        input[indice]=char(cifra)+48;
        indice++;
    }
    input[indice]='\0';


}

void loop(){

 get_GPS();
 delay(1000);
   sprintf(frame, "visor=false&lat=%s&lon=%s", lattitude, longitude);
 //delay(1000);

//    strtok(frame, ",");
//    strcpy(co2,strtok(NULL, ".")); // Gets longitude


  //
```

```
  //sprintf(frame, "visor=false&co2=$",co2);
   //Serial.print(frame);


  // Initializes HTTP service
   answer = sendATcommand("AT+HTTPINIT", "OK", 6000);
   if (answer == 1)
   {
       // Sets CID parameter
       answer = sendATcommand("AT+HTTPPARA=\"CID\",1", "OK", 5000);
       if (answer == 1)
       {
           // Sets url ?latitude=%s&longitude=%s   , latitude, longitude

sprintf(aux_str,"AT+HTTPPARA=\"URL\",\"http://115.127.80.43/gps/index.php?latitude=%s&l
ongitude=%s",lattitude,longitude);
           Serial.print(aux_str);


           //delay(30000);


           answer = sendATcommand("\"", "OK", 5000);
           if (answer == 1)
           {
               // Starts GET action
               answer = sendATcommand("AT+HTTPACTION=0", "+HTTPACTION:0,200",
30000);
               if (answer == 1)
```

```
        {

          Serial.println(F("Done!"));
        }
        else
        {
          Serial.println(F("Error getting url"));
        }

      }
      else
      {
        Serial.println(F("Error setting the url"));
      }
    }
    else
    {
      Serial.println(F("Error setting the CID"));
    }
  }
  else
  {
    Serial.println(F("Error initializating"));
  }
```

```
    sendATcommand("AT+HTTPTERM", "OK", 5000);


    delay(100);

}

/*void power_on(){

    uint8_t answer=0;

    // checks if the module is started
    answer = sendATcommand("AT", "OK", 2000);
    if (answer == 0)
    {
        // power on pulse
        digitalWrite(onModulePin,HIGH);
        delay(3000);
        digitalWrite(onModulePin,LOW);

        // waits for an answer from the module
        while(answer == 0){
            // Send AT every two seconds and wait for the answer
            answer = sendATcommand("AT", "OK", 2000);
        }
```

```
    }

}*/

    uint8_t x=0,  answer=0;
    char response[100];
    unsigned long previous;

    memset(response, '\0', 100);    // Initialize the string

    delay(100);

    // Clean the input buffer
    while( Serial.available() > 0) Serial.read();

    Serial.println(ATcommand);    // Send the AT command

    x = 0;
    previous = millis();

    // this loop waits for the answer
    do{
```

```
        if(Serial.available() != 0){
            response[x] = Serial.read();
            x++;
            // check if the desired answer is in the response of the module
            if (strstr(response, expected_answer1) != NULL)
            {
                answer = 1;
            }
        }
        // Waits for the asnwer with time out
    }while((answer == 0) && ((millis() - previous) < timeout));

  //response[x] = 0;
  //Serial.println(response);

    return answer;
}
```

PHP Code-

```
        !empty($_GET['time']) && !empty($_GET['satellites']) &&

        !empty($_GET['speedOTG']) && !empty($_GET['course'])


*/


        function getParameter($par, $default = null){

                if (isset($_GET[$par]) && strlen($_GET[$par])) return $_GET[$par];

                elseif (isset($_POST[$par]) && strlen($_POST[$par]))

                        return $_POST[$par];

                else return $default;

        }


        $file = 'gps.txt';

        $lat = getParameter("latitude");

        $lon = getParameter("longitude");

        $time = getParameter("time");

        $sat = getParameter("satellites");

        $speed = getParameter("speedOTG");

        $course = getParameter("course");

        $person = "\n".$lat.",".$lon.",".$time.",".$sat.",".$speed.",".$course;


        echo "

                DATA:\n
```

```
                    Latitude: ".$lat."\n
                    Longitude: ".$lon."\n
                    Time: ".$time."\n
                    Satellites: ".$sat."\n
                    Speed OTG: ".$speed."\n
                    Course: ".$course;

            if (!file_put_contents($file, $person, FILE_APPEND | LOCK_EX))
                    echo "\n\t Error saving Data\n";
            else echo "\n\t Data Save\n";
        }
        else {

?>

<!DOCTYPE html>
<html>

<head>

        <!-- Load Jquery -->

        <script language="JavaScript" type="text/javascript" src="jquery-1.10.1.min.js"></script>

        <!-- Load Google Maps Api -->
```

```
<!-- IMPORTANT: change the API v3 key -->

<script
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDPDiBcj4JK9Admz6_uYuVzeEB9J196z6s&sens
or=false"></script>

<!-- Initialize Map and markers -->

<script type="text/javascript">
        var myCenter=new google.maps.LatLng(41.669578,-0.907495);
        var marker;
        var map;
        var mapProp;

        function initialize()
        {
                mapProp = {
                  center:myCenter,
                  zoom:15,
                  mapTypeId:google.maps.MapTypeId.ROADMAP
                  };
                setInterval('mark()',5000);
        }

        function mark()
```

```
{
    map=new
google.maps.Map(document.getElementById("googleMap"),mapProp);
  var file = "gps.txt";
  $.get(file, function(txt) {
    var lines = txt.split("\n");
      for (var i=0;i<lines.length;i++){
        console.log(lines[i]);
        var words=lines[i].split(",");
        if ((words[0]!="")&&(words[1]!=""))
        {
          marker=new google.maps.Marker({
            position:new
google.maps.LatLng(words[0],words[1]),
            });
          marker.setMap(map);
          map.setCenter(new
google.maps.LatLng(words[0],words[1]));
          document.getElementById('sat').innerHTML=words[3];

    document.getElementById('speed').innerHTML=words[4];

    document.getElementById('course').innerHTML=words[5];
        }
      }
      marker.setAnimation(google.maps.Animation.BOUNCE);
  });
```

```
		}

		google.maps.event.addDomListener(window, 'load', initialize);
	</script>
</head>

<body>
	<?php
		echo '

		<!-- Draw information table and Google Maps div -->

		<div>
			<center><br />
				<b> SIM908 GPS position DEMO </b><br /><br />
				<div id="superior" style="width:800px;border:1px solid">
					<table style="width:100%">
						<tr>
							<td>Time</td>
							<td>Satellites</td>
							<td>Speed OTG</td>
							<td>Course</td>
						</tr>
						<tr>
```

```
                                        <td id="time">'. date("Y M d - H:m") .'</td>

                                        <td id="sat"></td>

                                        <td id="speed"></td>

                                        <td id="course"></td>

                                </tr>

                        </table>

                        </div>

                        <br /><br />

                        <div id="googleMap" style="width:800px;height:700px;"></div>

                </center>

        </div>';

    ?>

</body>

</html>


<?php } ?>
```