Abstract

A Tracking Device is an electronic device installed in any vehicle or laptop or any device to enable the owner or a third party to track the vehicle's location. Offline location tracking is an emerging trend that locates users even if they are not connected to the internet.

The work focused on android based technology by modifying GPS for offline location tracking. This project proposed to design a tracking system that works using GPS and GSM technology, which would be the cheapest source of vehicle tracking and it would work as anti-theft system.

It is an embedded system which is used for tracking and positioning of any other device by using Global Positioning System (GPS) and Global system for mobile communication (GSM). This design will continuously monitor a moving Device or laptop or vehicle and report the status of the Device on demand. For doing so a microcontroller is interfaced serially to a GSM Modem and GPS Receiver.

A GSM modem is used to send the position (Latitude and Longitude) of the device from a remote place. The GPS modem will continuously give the data i.e. the latitude and longitude indicating the position of the device. The same data is sent to the mobile at the other end from where the position of the vehicle is demanded by using an android application. When the request by user is sent to the number at the GSM modem, the system automatically sends a return reply to that mobile indicating the position of the vehicle in terms of latitude and longitude in real time.

1.Introduction

Location tracking is a process of determining the precise location. These tracking greatly impact to find location of a device, person or other asset to which it is attached and to record the position of the asset at regular intervals. With the rapid growth of online resources, user can easily track anyone or anything.

As a result, the location tracking being done progressively. Current focuses in the field includes online maps with features like online location tracking using GPS, point plotting on graph based on user requirements, online tracking of stored path. By using the concept of Google Tiles and Open street maps (.mbtiles) with the use of MOBAC Tools, this creates mbtiles file for storing the location

Many approaches have been made using internet to track the location. However, online location tracking is the improvement of network problems. Online location tracking can be seen as a way for users to track their own location. To track the location online we require information regarding maps in the form of kml files provided by Google and using the concept of parsing, we proposed the android based technique.

Offline location tracking mechanism which can overcome the problems of online location tracking such as no network area or low signal strength. With the popularity of smart phones network traffic gets increased, it is increasingly becoming important to track the location offline.

Typically, the number of location tracking maps that people uses. This makes it difficult in no network area or weak signal strength to track the location. Offline location tracking is an emerging trend that locates users even if they are not connected to the internet. The work focused on android based technology by modifying GPS for offline location tracking. These tracking greatly impact to find location and to record the position of the asset at regular intervals.

Purpose:

The main purpose of this project is to provide a cheap and offline tracking system which should help in tracking vehicles as well as other assets of end user. Proposed system will be most useful for location tracking application and will also provide the authentication. This system will track the location automatically according to user reminder task and will also overcome the traditional desktop-based approach of obtaining information offline.

Need:

Most of the tracking system are online i.e. they required internet connection for tracking location of assets which is not convenient due to low network signals in rural areas. Hence a offline tracking system with appropriate authentication is required to track vehicles and other asset.

2.Project plan

2.1 Roles and Responsibility:

Roles and responsibility of every member in group in given below:

ROLES	RESPONSIBILITIES
Yash Parmar	Requirement gathering, Arduino Programming, Application Development, Testing.
Aditya Nangare	Requirement gathering, Testing, Application Development, UI implementation.
Niraj Rasal	Requirement gathering, UI implementation, Testing, Application Development.
Pratik Korde	Requirement gathering, Arduino Programming, Application Development, Testing.

Table 2.1 Roles and Responsibilities

2.2 Software Model:

- For this project, we use iterative model.
- Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented.
- At each iteration design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through is to develop a system through repeated cycles and in smaller portions at a time.

2.3 Schedule of Project:

- Estimated start date: 3 Feb 2019
- Actual start date: 7 Feb 2019
- Estimated end date: 27 March 2019
- Actual end date: 30 March 2019

3.Project Requirements

3.1 Hardware Requirements:

1. Smart Phone:

a. Minimum RAM 512GB (Recommended 2GB)

b. Minimum operating system lollipop 5.0

3. Arduino Uno:

a. Arduino UNO R3.

4. GSM Module:

a. GSM SIM800L.

5. GPS Module:

a. Neo-6-m GPS Module

3.2 Software Requirements:

1. Android Application
2. Message Interface\

3.3 Environmental Requirements

1. Android OS for Smartphone
2. Android Studio

3.4 Technologies used:

1. Embedded C language
2. Android programming
3. Database: MySQL 5.5
4. Java Programming Language
5. Extended Mark-up Language (XML)

3. Arduino Uno:

Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

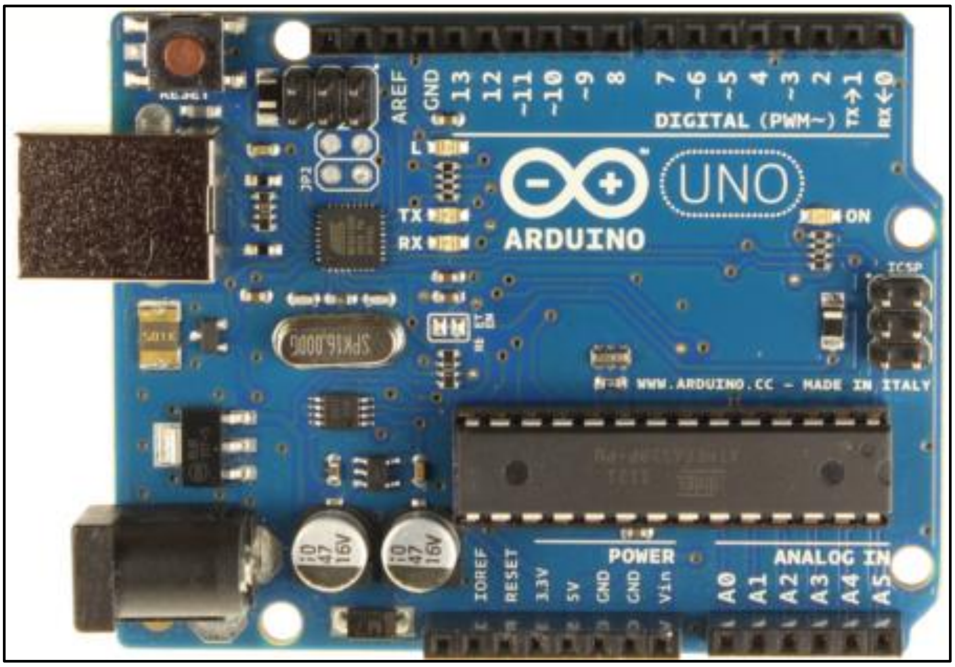


Fig 3.1 Arduino UNO

The Arduino boards are able to read inputs - light, proximity or air quality on a sensor, or an SMS or Twitter message - and turn it into an output - activating a motor, turning on a light, publishing content online or trigger external events. You can tell your board what to do by writing code and uploading it to the microcontroller on it using the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

VCC:

Digital supply voltage. GND: Ground Port B (PB7:0): XTAL1/XTAL2/TOSC1/TOSC2 Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers have symmetric drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator issued as chip clock source, PB7...6 is used as TOSC2

Port C (PC5:0):

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET:

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBLF use is un programmed, PC 6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

Port D (PD7:0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D Output Buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET:

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBLF use is un programmed, PC 6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset

Port D (PD7:0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output Buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

4.2.1 Key features:

- Inexpensive and flexible hardware
- Cross-platform
- Open source and extensible hardware
- Open source and extensible software
- Simple programming environment

3.6.GSM Module:

The GSM modem is a specialized type of modem which accepts a SIM card operates on a user’s mobile number over a network, just like a cellular phone.

Booting the GSM Module:

- Insert the SIM card to GSM module and lock it.
- Connect the adapter to GSM module and turn it ON.
- Now wait for some time (say 1 minute) and see the blinking rate of “status LED‟ or “network LED‟ (GSM module will take some time to establish connection with mobile network)
- Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection

GSM Module:

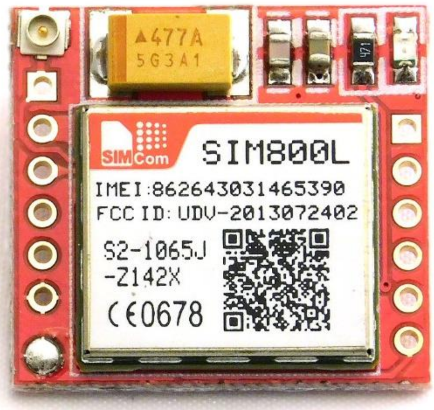


Fig 3.2 GSM-Module.

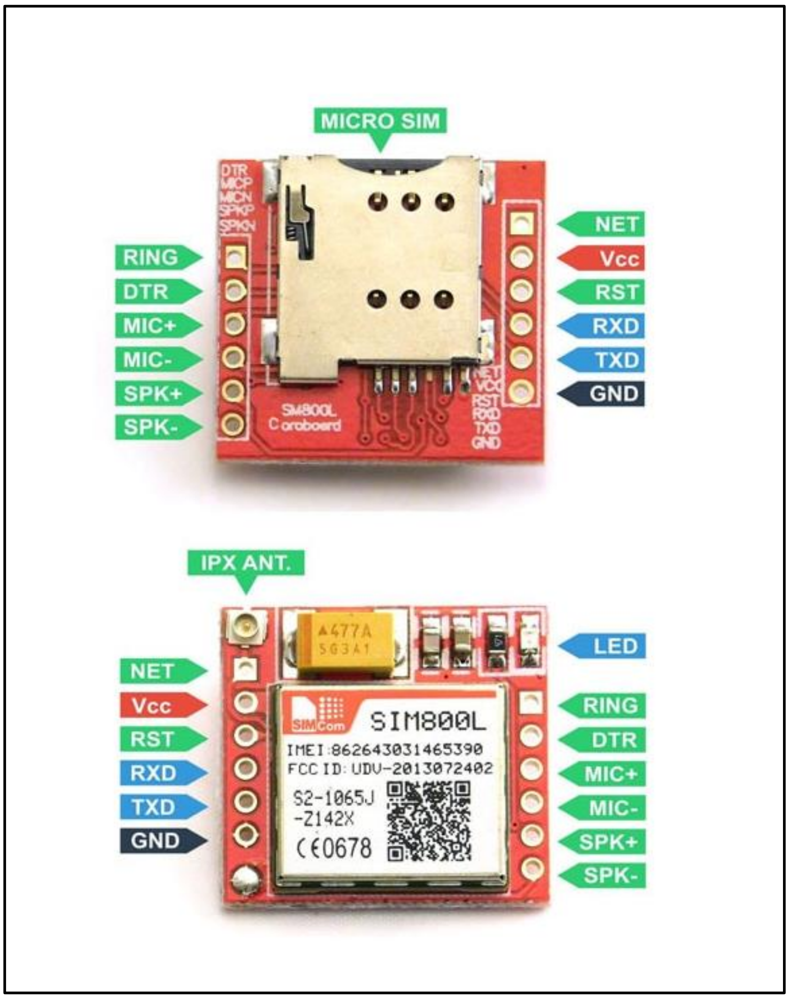


Fig 3.3 GSM-Module Pin Diagram.

Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

Connecting GSM Module to Arduino:

There are two ways of connecting GSM module to Arduino. In any case, the communication between Arduino and GSM module is serial. So, we are supposed to use serial pins of Arduino (Rx and TX). So, if you are going with this method, you may connect the TX pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to TX pin of Arduino. You read it right? GSM TX –> Arduino Rx and GSM Rx –> Arduino TX. Now connect the ground pin of Arduino to ground pin of GSM module! So that is all! You made 3 connections and the wiring is over! Now you can load different programs to communicate with GSM module and make it work.

The problem with this connection is that, while programming Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will not be loaded successfully to Arduino. So, you have to disconnect wiring in Rx and TX each time you burn the program to Arduino. Once the program is loaded successfully, you can reconnect these pins and have the system working!

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.

Features of GSM:

- Dual band GSM/GPRS 900/1800MHz.
- Configurable baud rate.
- SIM card holder.
- Built in network status LED.
- Inbuilt powerful TCP/IP protocol stack for internet data transfer over GPRS

AT Command for GSM Module:

Command	Description
AT+CMGD	DELETE SMS MESSAGE
AT+CMGF	SELECT SMS MESSAGE FORMAT
AT+CMGL	LIST SMS MESSAGES FROM PREFERRED STORE
AT+CMGR	READ SMS MESSAGE
AT+CMGS	SEND SMS MESSAGE
AT+CMGW	WRITE SMS MESSAGE TO MEMORY
AT+CMSS	SEND SMS MESSAGE FROM STORAGE
AT+CMGC	SEND SMS COMMAND
AT+CNMI	NEW SMS MESSAGE INDICATIONS
AT+CPMS	PREFERRED SMS MESSAGE STORAGE
AT+CRES	RESTORE SMS SETTINGS
AT+CSAS	SAVE SMS SETTINGS
AT+CSCA	SMS SERVICE CENTER ADDRESS
AT+CSCB	SELECT CELL BROADCAST SMS MESSAGES
AT+CSDH	SHOW SMS TEXT MODE PARAMETERS
AT+CSMP	SET SMS TEXT MODE PARAMETERS
AT+CSMS	SELECT MESSAGE SERVICE

Table 3.1 AT Commands According to GSM SIM800L

GPS module:

GPS Technology the Global Positioning System (GPS) is a satellite based navigation system consists of a network of 24 satellites located into orbit.

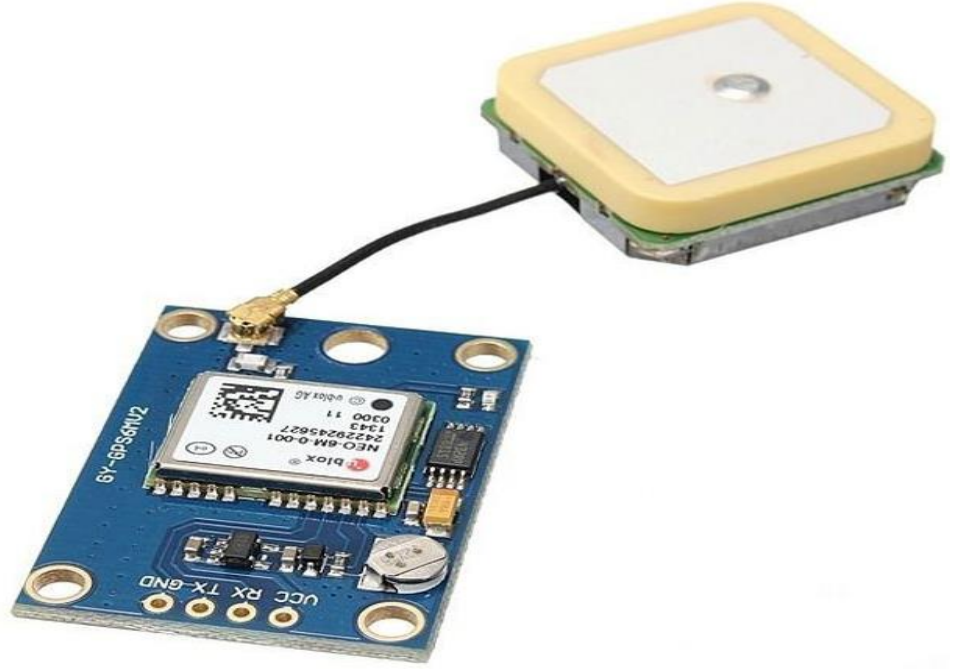


Fig 3.3 GPS-Module.

The system provides essential information to military, civil and commercial users around the world and which is freely accessible to anyone with a GPS receiver. GPS works in any weather circumstances at anywhere in the world. Normally no subscription fees or system charges to utilize GPS. A GPS receiver must be locked on to the signal of at least three satellites to estimate 2D position (latitude and longitude) and track movement. With four or more satellites in sight, the receiver can determine the user's 3D position (latitude, longitude and altitude).

Once the vehicle position has been determined, the GPS unit can determine other information like, speed, distance to destination, time and other. GPS receiver is used for this research work to detect the vehicle location and provide information to responsible person through GSM technology. GPS module sends the data related to tracking position in real time, and it sends so many data in NMEA format (see the screenshot below). NMEA format consist several sentences, in which we only need one sentence. This sentence starts from $GPGGA and contains the coordinates, time and other useful information. This GPGGA is referred to Global Positioning System Fix Data. Know more about Reading GPS data and its strings here. We can extract coordinate from $GPGGA string by counting the commas in the string. Suppose you find $GPGGA string and stores it in an array, then Latitude can be found after two commas and Longitude can be found after four commas. Now these latitude and longitude can be put in other arrays.

RS232:

In telecommunications systems used today RS-232 is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit terminating equipment, originally defined as data communication equipment), such as a modem.

The RS-232 standard is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pinout of connectors. The current version of the standard is TIA-232-F Interface between Data Terminal Equipment and Data Circuit Terminating Equipment Employing Serial Binary Data Interchange, issued in 1997.

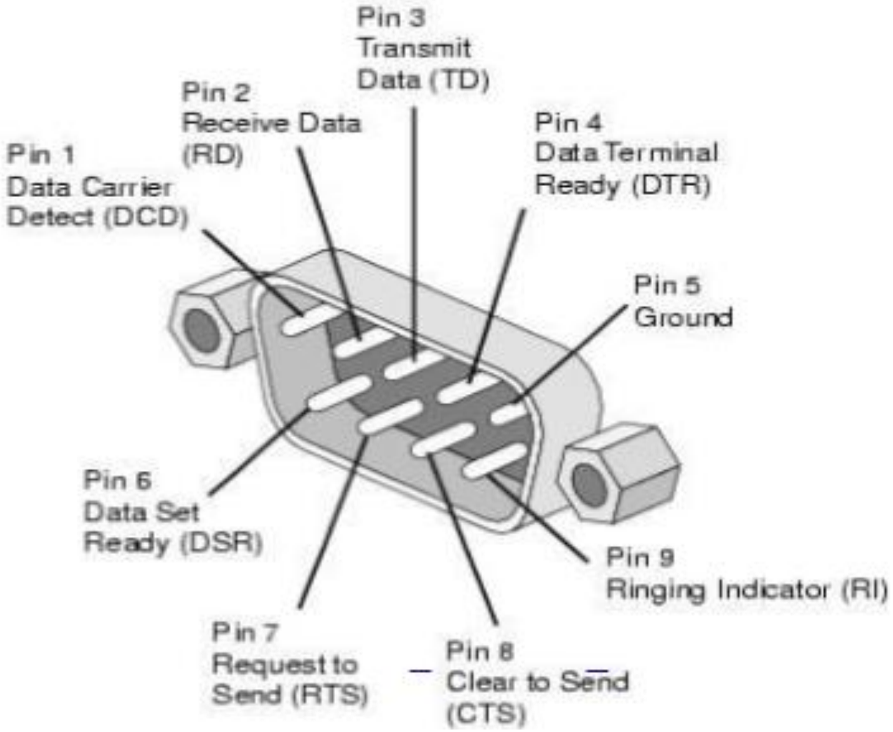


Fig. 3.5 RS232

An RS-232 serial port was once a standard feature of a personal computer, used for connections to modems, printers, mice, data storage, power supplies, and other peripheral devices. However, RS-232 is hampered by low transmission speed, large voltage swing, and large standard connectors.

In modern personal computers, USB has displaced RS-232 from most of its peripheral interface roles. Many computers do not come equipped with RS-232 ports and must use either an external USB-to-RS-232 converter or an internal expansion card with one or more serial ports to connect to RS-232 peripherals. Nevertheless, RS-232 devices are still used, especially in industrial machines, networking equipment and scientific instruments.

4.Project Design

The modem sends the Longitude and Latitude position of any vector from an unknown place. When the user demands or requests, the system accordingly delivers a reply to the device indicating the location of a vehicle with respect to the latitude and longitude values. Microcontroller's internal memory i.e. ROM (Read-Only Memory) consists of the code. Processing of the instruction is done using instruction set and acts as an ally between GSM and GPS. GPS only transmits data whereas GSM imparts as well as receive data. The microcontroller is linked to the GPS pins and a serial connection is established between the GSM pins and microcontroller.

The data is received by the GPS receiver from numerous satellites i.e. processing of data containing GPS results from diverse satellites is done by the microcontroller which is then acquired by the GSM modem. The receiver sends the data to microcontroller which is then processed by the microcontroller and catches only latitude and longitude numerical values blocking information such as interval, elevation, alias of satellite, verification etc. On the user end the GSM enabled device such as smartphone acts like an SMS receiver and acting as a SMS sender is a GSM modem connected serially to the microcontroller. In terms of power, GSM uses 12 volts whereas microcontroller circuitry and GPS use 5 volts which are provided by a 12 volts 3.2 Ampere battery.

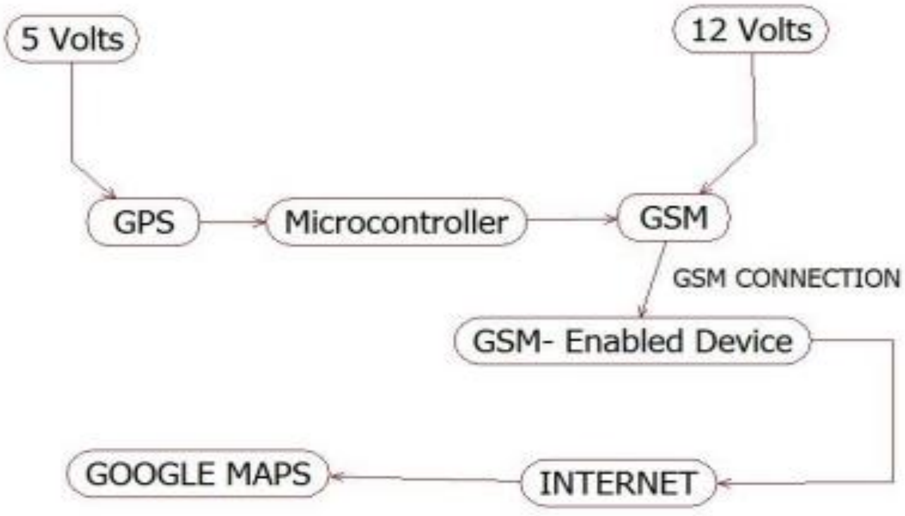


Fig 4.1 Block Diagram

4.1. Risk Projection

Enforcing security in location tracking application is essential. The system stores the rea time location of the user asset and hence any leak of information should be avoided. Only authorized user must get permission to track the vehicle or other asset and user should agree to the terms and condition of the application.

A risk analysis applied on a offline tracker system has been conducted. Out of 12 examined risks, 6 were classified as low and 6 as high, i.e., most of the identified risks were deemed as moderate. The risks classified as high were either related to the human factor or to the software components of the system. The results indicate that with the implementation of standard security features, new, as well as, current risks can be minimized to acceptable levels albeit that the most serious risks, i.e., those derived from the human factor, need more careful consideration, as they are inherently complex to handle.

The components that are likely to cause risks are –

- GPS module
- Arduino
- Mobile device (App)
- Network

The risks can be categorized into – software, hardware and human caused risks.

SOFTWARE:

1. Inadequate authentication

2. Software vulnerability in OS/API.

3. Database connectivity

4. Inappropriate app behavior (crashes, incorrect information)

5. UI issues

6.displaying inaccurate location

7.Low GPS signal.

HARDWARE:

1. Physical devices connectivity issues.
2. low battery in system.
3. Inadequate physical security.
4. loosen connection of wires in Arduino.

HUMAN:

1. Sloppy end users
2. Disgruntled employees
3. Poor password selection
4. Sending multiple request to tracker.

4.2 Feasibility of the Project:

Feasibility study is an important phase of the System Development Life Cycle where the proposed system is viewed to determine the suitability and feasibility of the system i.e.to determine whether or not the system is worth doing.

An important outcome of the preliminary investigation phase is that the package required is feasible and meets the efficiency criteria. The workability standard.

The various types of feasibility are.

4.2.1 Technical Feasibility:

This evaluation determines

- Whether the technology needed for the proposed system is available.
- How this technology integrated within the organization?
- Technical evaluation must also access whether the existing system can be upgraded to use the new technology & whether the organization has expertise to use

The project is based on wireless technology for communication and embedded systems which are currently in-phase with the technology. The android application which is deployable to all mobile operating systems makes it reach to most of the users. Therefore, the project is very much favored by technology.

4.2.2 Operational Feasibility:

- Operational feasibility determines how acceptable the application is with the organization.
- One of the main objectives of developing a user-friendly application is that the user does not face any problem while operating with the system.
- The 'Track my device' app can control all the user's devices centrally. The app is operational in and works remotely in all environments with internet being the only dependency. The Arduino controls all the sockets require to be connected to all the components through various pins. Thus, the project is easily operational in any environment with appropriate interfacing.

4.2.3 Financial Feasibility:

This evaluation looks at the financial aspects of the project. To carry out an economical study it is necessary to place actual money values against any activities needed to implements the condition system.

The project consists of basic hardware and electronic components such as Arduino Uno, GPS module, GSM module, battery which are affordable and making it financially economical to implement.

4.3 Behavioral and Function Description

In this project, Arduino is used for controlling whole the process with a GPS Receiver and GSM module. GPS Receiver is used for detecting coordinates of the vehicle, GSM module is used for sending the coordinates to user by SMS. And an optional 16x2 LCD is also used for displaying status messages or coordinates. We have used GPS Module SKG13BL and GSM Module SIM900A.

When we ready with our hardware after programming, we can install it in our vehicle and power it up. Then we just need to send a SMS, “Track Vehicle”, to the system that is placed in our vehicle. We can also use some prefix (#) or suffix (*) like #Track Vehicle*, to properly identify the starting and ending of the string, like we did in these projects: GSM Based Home Automation and Wireless Notice Board.

Sent message is received by GSM module which is connected to the system and sends message data to Arduino. Arduino reads it and extract main message from the whole message. And then compare it with predefined message in Arduino. If any match occurs then Arduino reads coordinates by extracting $GPGGA String from GPS module data (GPS working explained above) and send it to user by using GSM module. This message contains the coordinates of vehicle location.

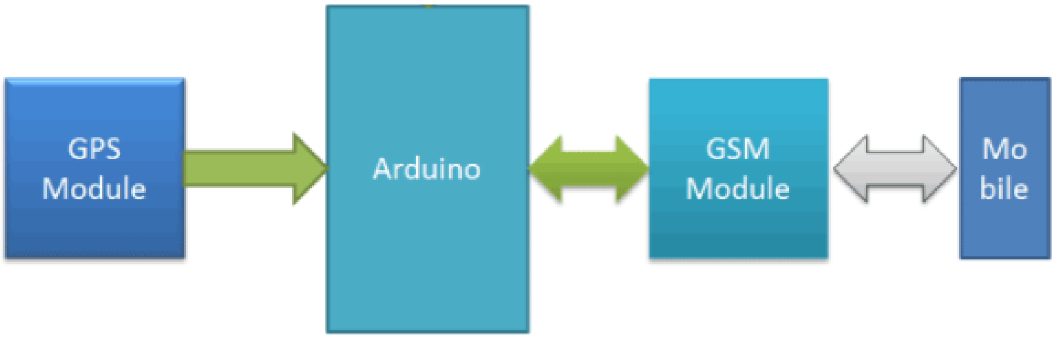


Fig. 4.3.1 Functional Diagram

4.3.1 Concept and Overview:

This vehicle tracking system takes input from GPS and send it through the GSM module to desired mobile/laptop using mobile communication. Proper Authentication is also provided to android application therefore only authorized person can access the device location. username and password is required for each session to track location.

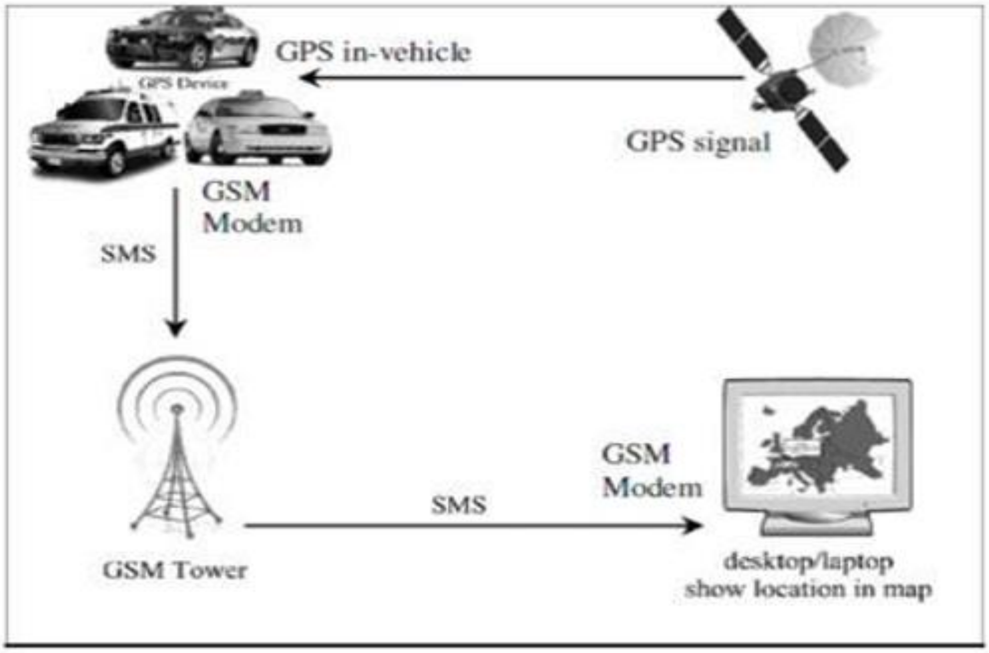


Fig 4.3.1 Working Module

This vehicle tracking system takes input from GPS and send it through the GSM module to desired mobile/laptop using mobile communication. Vehicle Tracking System is one of the biggest technological advancements to track the activities of the Device.

The security system uses Global Positioning System GPS, to find the location of the monitored or tracked vehicle and then uses satellite or radio systems to send to send the coordinates and the location data to the user mobile. Due to this technology vehicle tracking systems are becoming increasingly popular among owners of expensive vehicles.

4.4 DIAGRAMS:

UML Diagram:

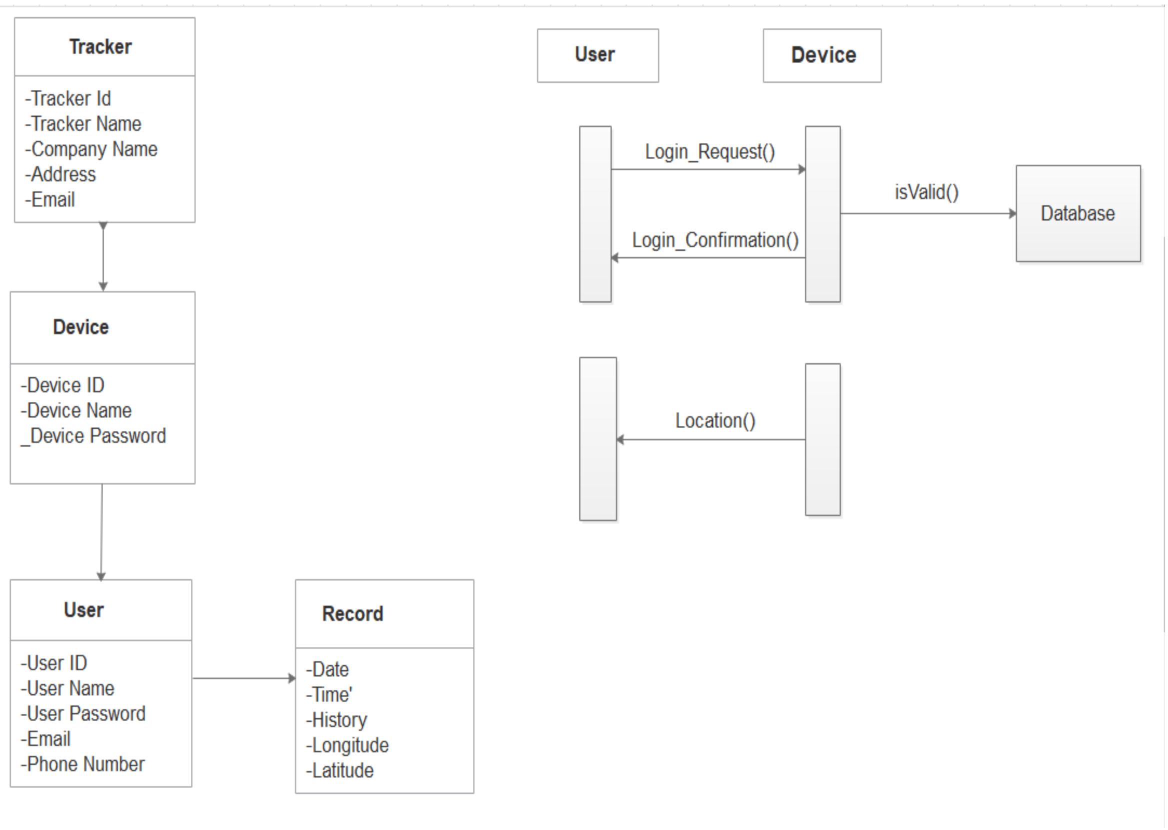


Fig. 4.4.1 UML Diagram

Sequence Daigram:

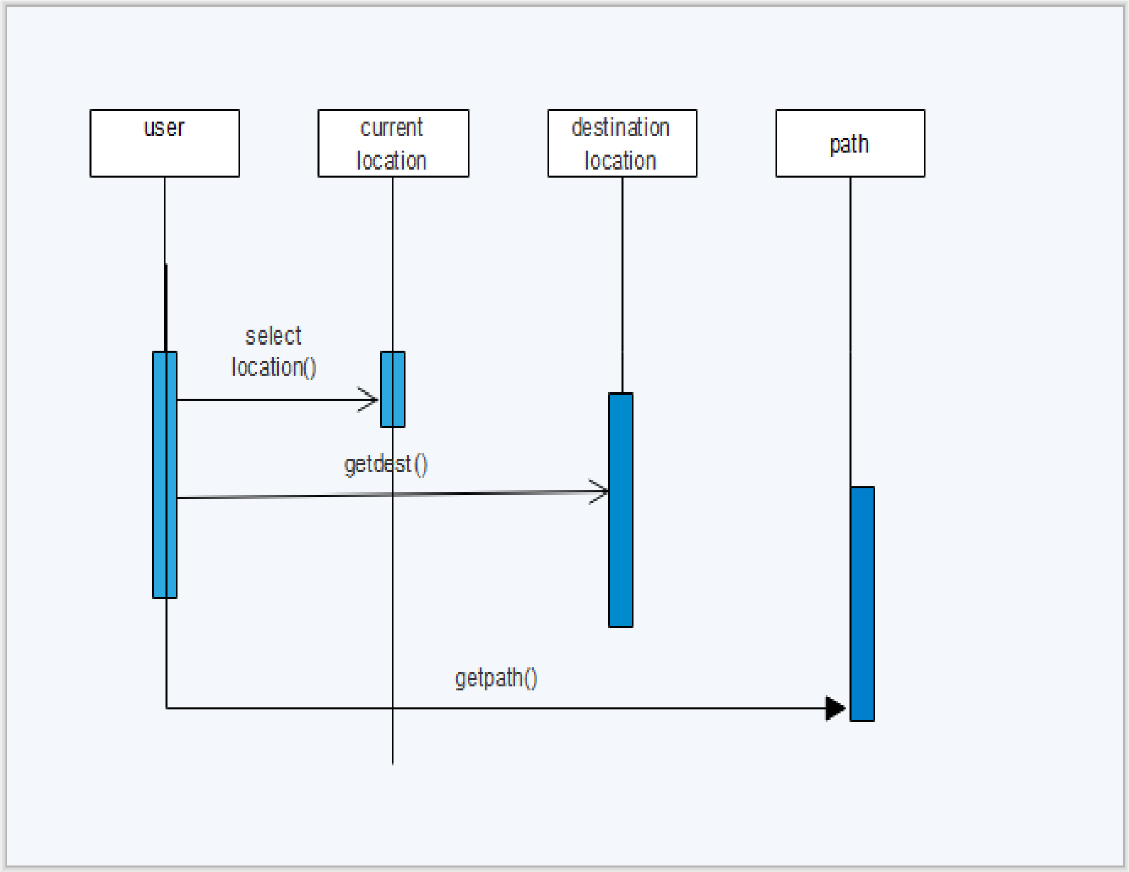


Fig. 4.4.2 Sequence Diagram

Activity Diagram:

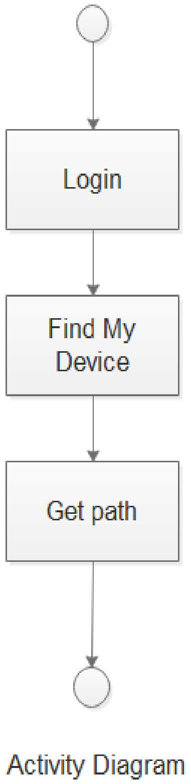
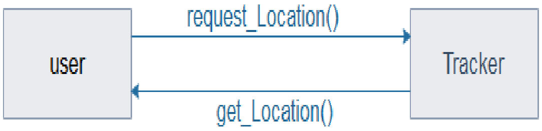
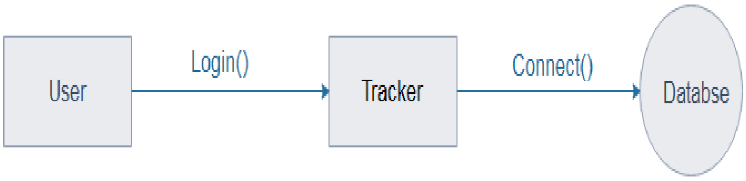


Fig. 4.4.3 Activity Diagram

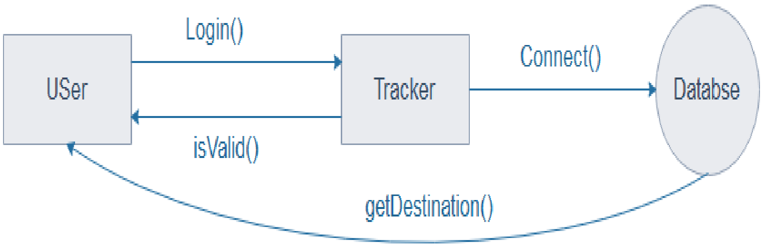
4.4.4 Data Flow Daigram:



DFD level 0

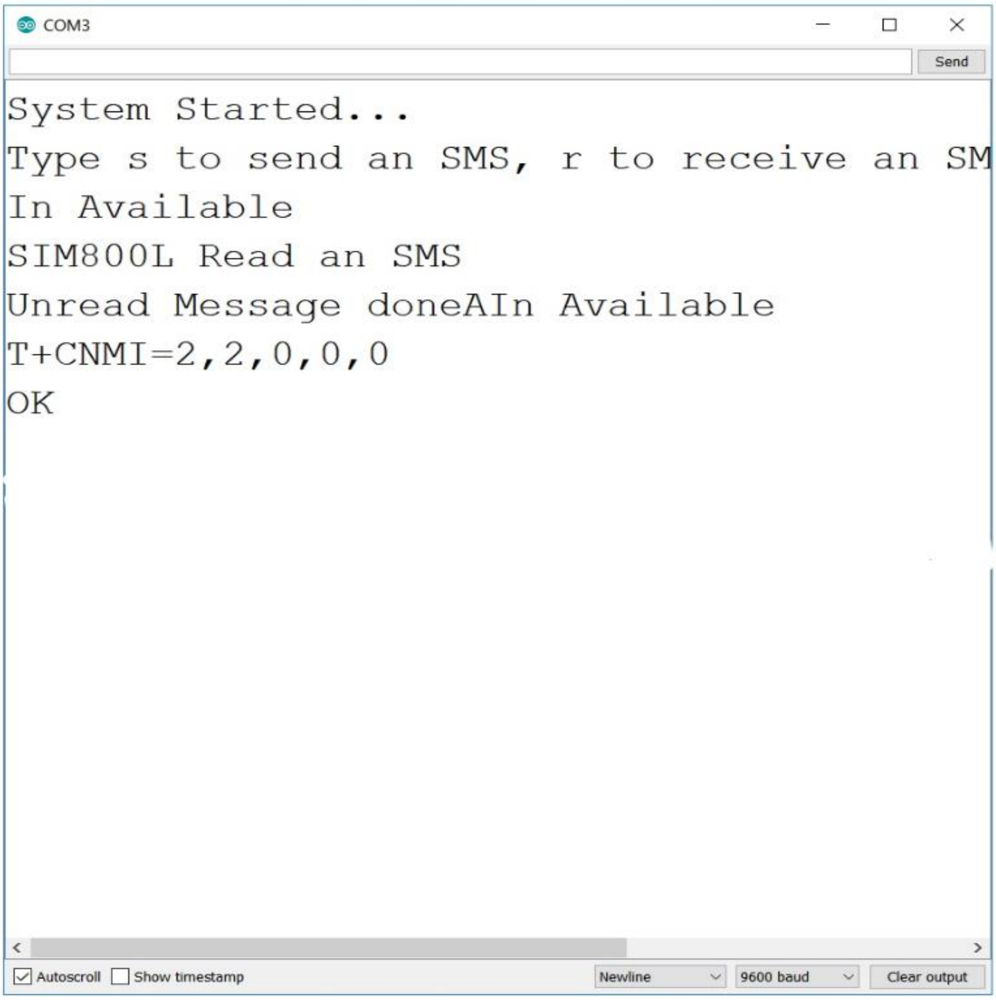


DFD level 1



DFD level 2

Screen-Shots:



```
System Started...
Type s to send an SMS, r to receive an SM
In Available
SIM800L Read an SMS
Unread Message doneAIn Available
T+CNMI=2,2,0,0,0
OK
```

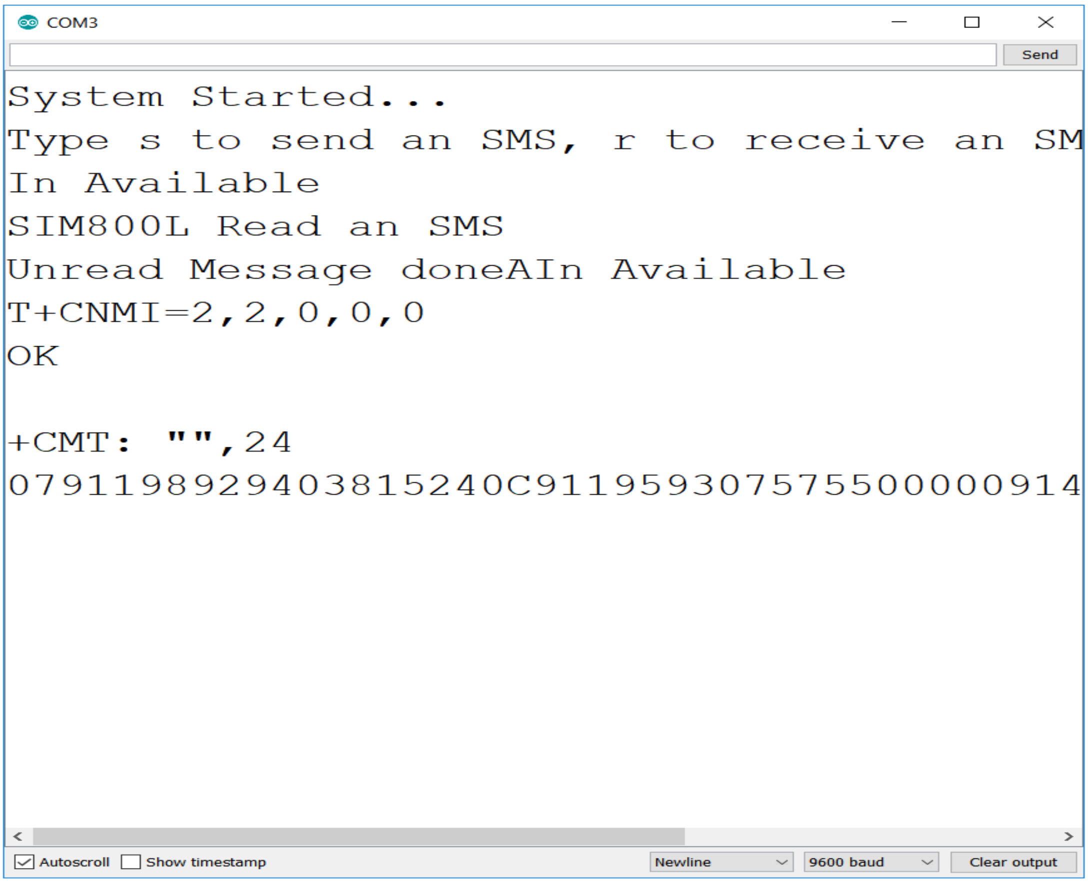
☑ Autoscroll □ Show timestamp Newline 9600 baud

Fig 5.1 Send and Receive

This is the screenshot after the system is started. In this there are two things;

1. Send SMS to the registered Number.
2. Receive SMS from any number to device.

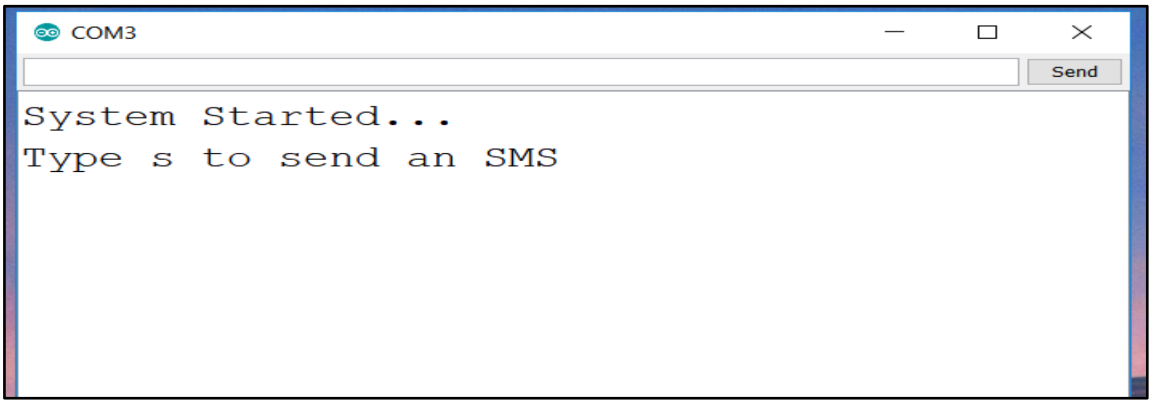
The device can send the SMS to registered number, for that the user 1st need to register his number. This device is able to receive the SMS from any mobile number. For send the SMS we need to press the ‘S’ key. Then device will send the longitude and latitude to the register number. This is used for testing purpose weather the device is sending SMS or not. For receiving SMS we need to press ‘R’ key.



```
System Started...
Type s to send an SMS, r to receive an SM
In Available
SIM800L Read an SMS
Unread Message doneAIn Available
T+CNMI=2,2,0,0,0
OK

+CMT: "",24
07911989294038152 40C91195930757550000914
```

Fig 5.2 Receiving message

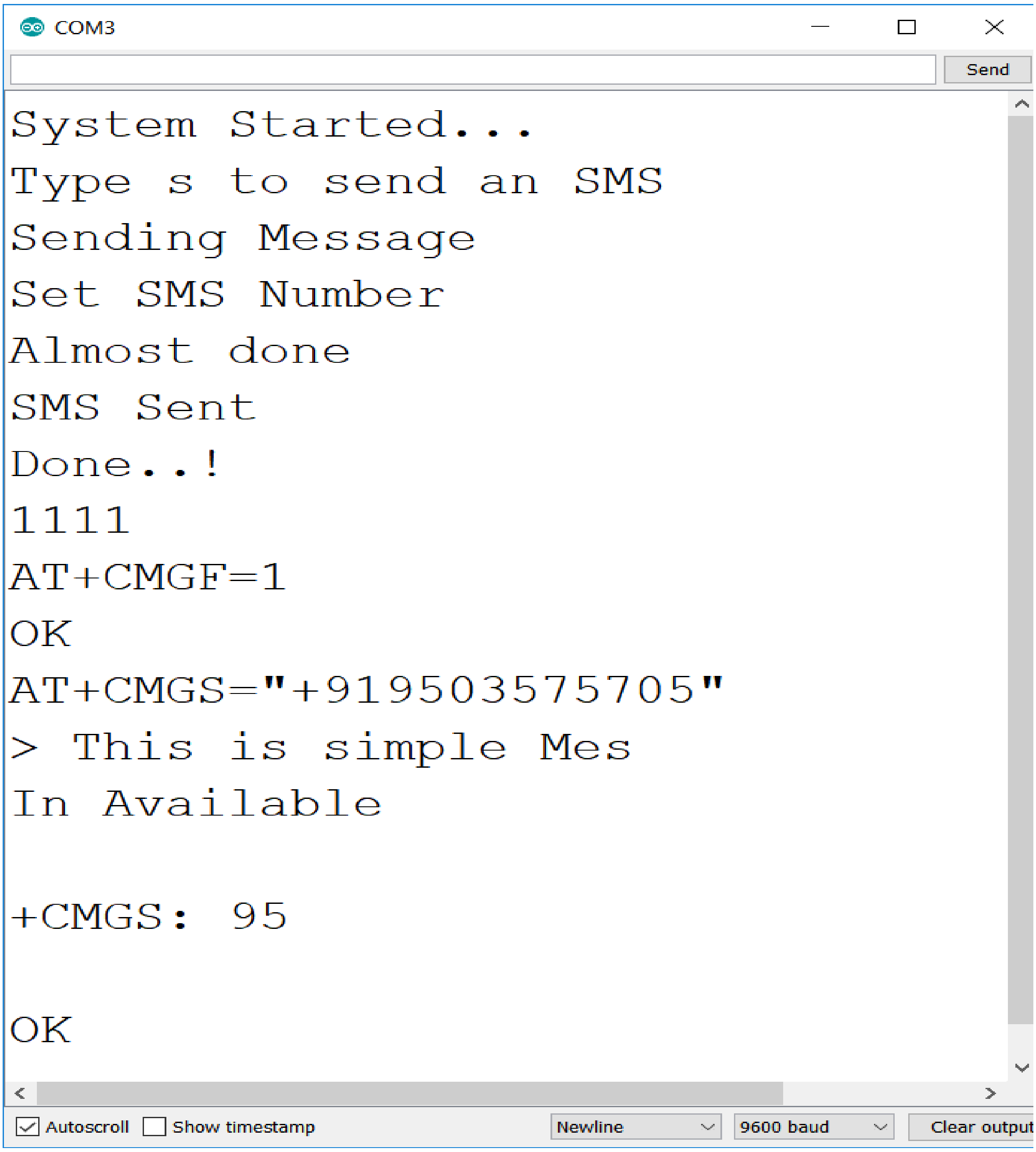
In this screen-shot we received the raw data. After pressing the key ‘R’ the receiving code is performed correctly is checked. This are some raw data send via GPS and GSM.



```
System Started...
Type s to send an SMS
```

Fig 5.3 System Started

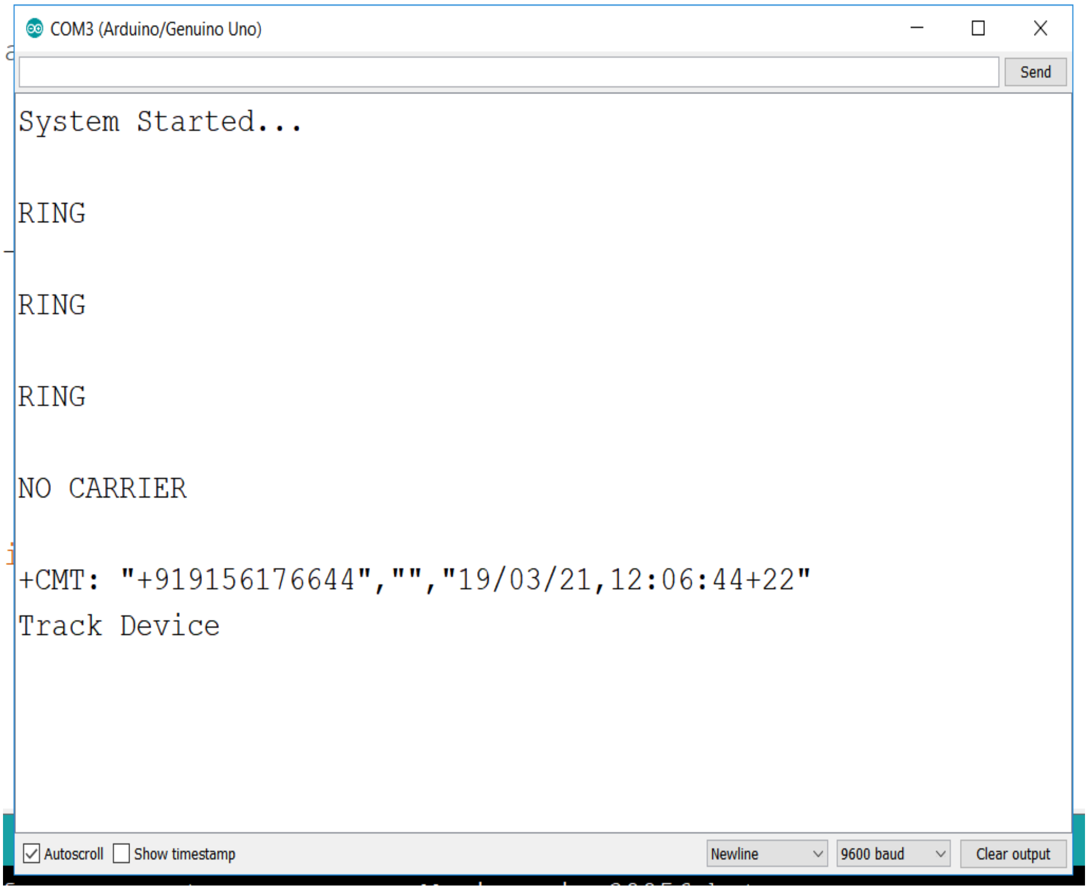
This screen appears after the system is started. In this we press 'S' to send SMS to the specified device. The message is longitude and latitude of our current location. Then that message which content longitude and latitude is fetched by our app which is Find_Device. Then app fetch that coordinated and provide the current location to the user.



```
System Started...
Type s to send an SMS
Sending Message
Set SMS Number
Almost done
SMS Sent
Done..!
1111
AT+CMGF=1
OK
AT+CMGS="+919503575705"
> This is simple Mes
In Available

+CMGS: 95

OK
```

Fig 5.4 Send Normal Message

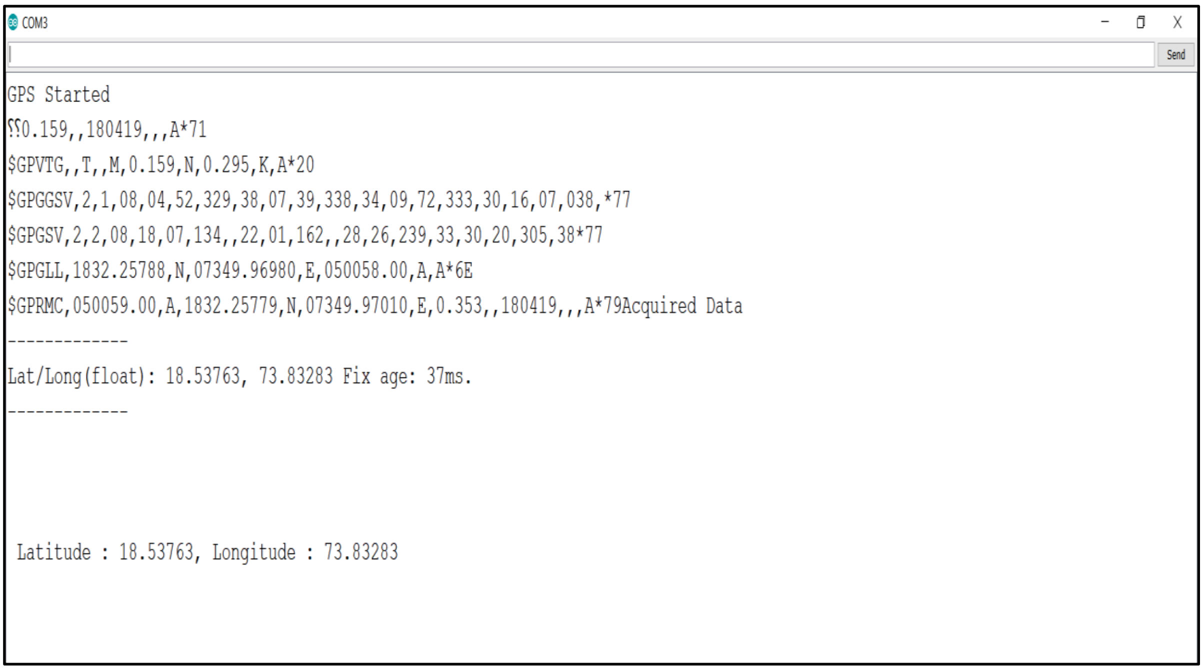
When system is started, the option screen appears which ask for receiving and sending the SMS. After pressing the 'S' key the normal message is sent to the registered user number.



```
System Started...

RING

RING

RING

NO CARRIER

+CMT: "+919156176644","","19/03/21,12:06:44+22"
Track Device
```

Fig 5.5 Making call

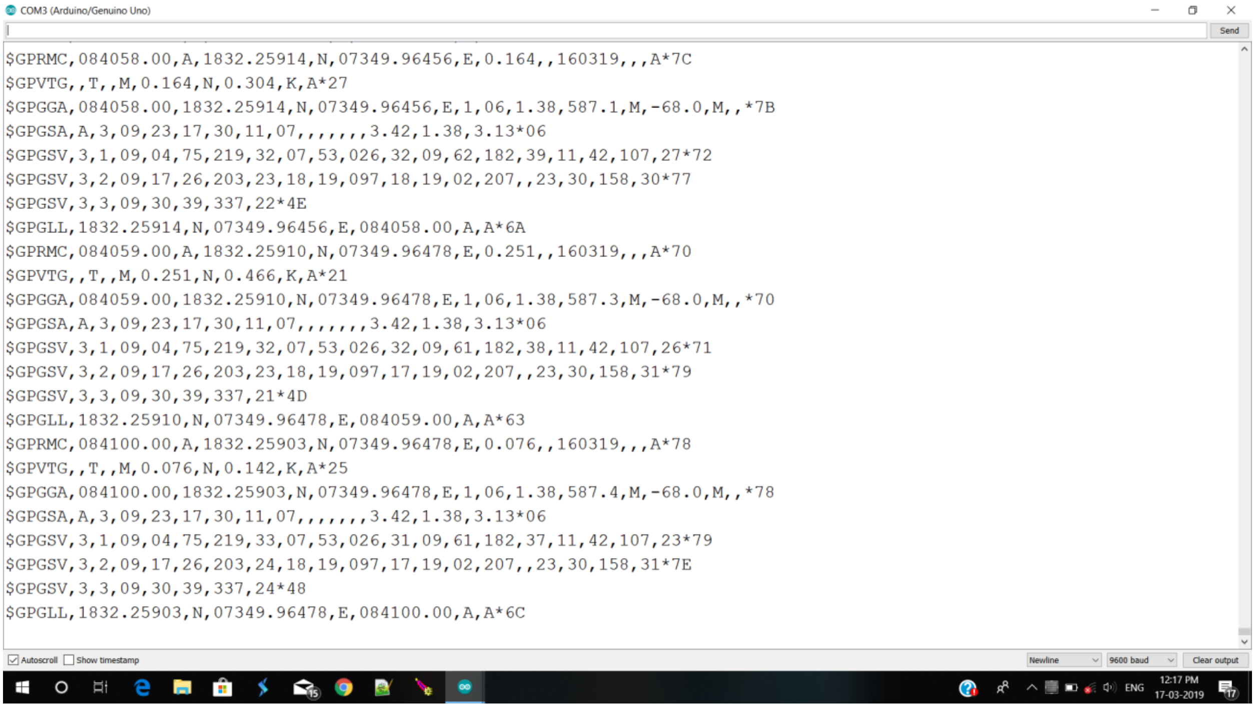
There is an option to make the call. After pressing the ‘C’ key, then the call is placed to the specified number. And the pre-define voice message will be played. If anyone try to make the call to the number which is placed in our device then the emulator will display such kind of output.



```
GPS Started
⸮⸮0.159,,180419,,,A*71
$GPVTG,,T,,M,0.159,N,0.295,K,A*20
$GPGGSV,2,1,08,04,52,329,38,07,39,338,34,09,72,333,30,16,07,038,*77
$GPGSV,2,2,08,18,07,134,,22,01,162,,28,26,239,33,30,20,305,38*77
$GPGLL,1832.25788,N,07349.96980,E,050058.00,A,A*6E
$GPRMC,050059.00,A,1832.25779,N,07349.97010,E,0.353,,180419,,,A*79Acquired Data
-------------
Lat/Long(float): 18.53763, 73.83283 Fix age: 37ms.
-------------



 Latitude : 18.53763, Longitude : 73.83283
```

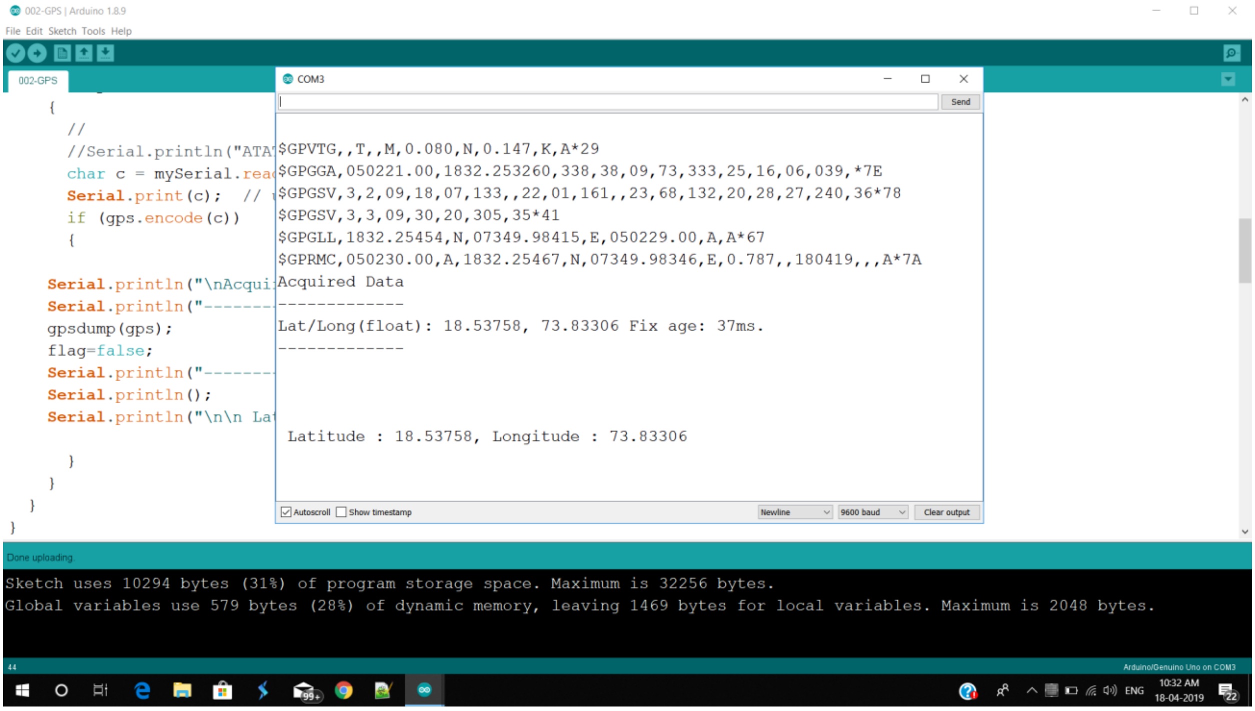
Fig 5.6 Fetching Data

When the device is started, if the user ask to track the device the coordinated is fetched and shown differently. GPS provides combine longitude and latitude which is separated and shown differently. Longitude and latitude and then fetched by the app which provide the current location of the device.



```
$GPRMC,084058.00,A,1832.25914,N,07349.96456,E,0.164,,160319,,,A*7C
$GPVTG,,T,,M,0.164,N,0.304,K,A*27
$GPGGA,084058.00,1832.25914,N,07349.96456,E,1,06,1.38,587.1,M,-68.0,M,,*7B
$GPGSA,A,3,09,23,17,30,11,07,,,,,,,3.42,1.38,3.13*06
$GPGSV,3,1,09,04,75,219,32,07,53,026,32,09,62,182,39,11,42,107,27*72
$GPGSV,3,2,09,17,26,203,23,18,19,097,18,19,02,207,,23,30,158,30*77
$GPGSV,3,3,09,30,39,337,22*4E
$GPGLL,1832.25914,N,07349.96456,E,084058.00,A,A*6A
$GPRMC,084059.00,A,1832.25910,N,07349.96478,E,0.251,,160319,,,A*70
$GPVTG,,T,,M,0.251,N,0.466,K,A*21
$GPGGA,084059.00,1832.25910,N,07349.96478,E,1,06,1.38,587.3,M,-68.0,M,,*70
$GPGSA,A,3,09,23,17,30,11,07,,,,,,,3.42,1.38,3.13*06
$GPGSV,3,1,09,04,75,219,32,07,53,026,32,09,61,182,38,11,42,107,26*71
$GPGSV,3,2,09,17,26,203,23,18,19,097,17,19,02,207,,23,30,158,31*79
$GPGSV,3,3,09,30,39,337,21*4D
$GPGLL,1832.25910,N,07349.96478,E,084059.00,A,A*63
$GPRMC,084100.00,A,1832.25903,N,07349.96478,E,0.076,,160319,,,A*78
$GPVTG,,T,,M,0.076,N,0.142,K,A*25
$GPGGA,084100.00,1832.25903,N,07349.96478,E,1,06,1.38,587.4,M,-68.0,M,,*78
$GPGSA,A,3,09,23,17,30,11,07,,,,,,,3.42,1.38,3.13*06
$GPGSV,3,1,09,04,75,219,33,07,53,026,31,09,61,182,37,11,42,107,23*79
$GPGSV,3,2,09,17,26,203,24,18,19,097,17,19,02,207,,23,30,158,31*7E
$GPGSV,3,3,09,30,39,337,24*48
$GPGLL,1832.25903,N,07349.96478,E,084100.00,A,A*6C
```

Fig 5.6 Raw Data

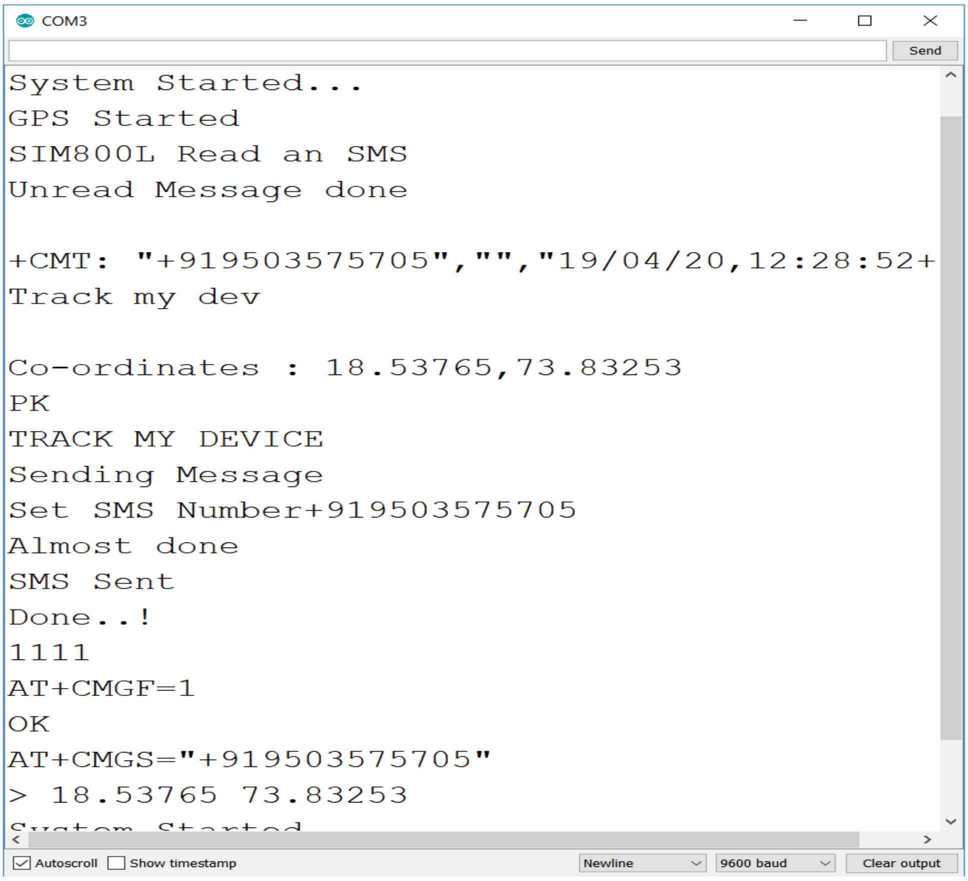


```
    {
      //
      //Serial.println("ATA
      char c = mySerial.rea
      Serial.print(c);  // 
      if (gps.encode(c))
      {

    Serial.println("\nAcqui
    Serial.println("-------
    gpsdump(gps);
    flag=false;
    Serial.println("-------
    Serial.println();
    Serial.println("\n\n La

      }
    }
  }
}
```

```
$GPVTG,,T,,M,0.080,N,0.147,K,A*29
$GPGGA,050221.00,1832.253260,338,38,09,73,333,25,16,06,039,*7E
$GPGSV,3,2,09,18,07,133,,22,01,161,,23,68,132,20,28,27,240,36*78
$GPGSV,3,3,09,30,20,305,35*41
$GPGLL,1832.25454,N,07349.98415,E,050229.00,A,A*67
$GPRMC,050230.00,A,1832.25467,N,07349.98346,E,0.787,,180419,,,A*7A
Acquired Data
-------------
Lat/Long(float): 18.53758, 73.83306 Fix age: 37ms.
-------------



 Latitude : 18.53758, Longitude : 73.83306
```

```
Sketch uses 10294 bytes (31%) of program storage space. Maximum is 32256 bytes.
Global variables use 579 bytes (28%) of dynamic memory, leaving 1469 bytes for local variables. Maximum is 2048 bytes.
```

Fig 5.7 Fetched Data in seconds



```
System Started...
GPS Started
SIM800L Read an SMS
Unread Message done

+CMT: "+919503575705","","19/04/20,12:28:52+
Track my dev

Co-ordinates : 18.53765,73.83253
PK
TRACK MY DEVICE
Sending Message
Set SMS Number+919503575705
Almost done
SMS Sent
Done..!
1111
AT+CMGF=1
OK
AT+CMGS="+919503575705"
> 18.53765 73.83253
System Started
```

Fig 5.8 Getting location

This is the actual working of the system. In this, when the device is started the gps get start then the user’s registered number is displayed. If we press track my device button from the app, then the device will fetch the coordinates from the raw data and will display the coordinates and that coordinates are send to the user’s number. Then that coordinates is taken by the application to track the location of that device.

Android Application:

Android application is needed to track the location on maps and also to provide authentication. First user has to register the device name, mobile number and password. all values are stored in database. User has to enter username and password to track the location of device. Google map API is used to however display the location on map, the displayed location is not exactly accurate due to limitation of GPS module, but the device will be in range of displayed location. The android application will also show user the distance and the path to reach the device while tracking. Application will require internet access in order to track the location

The tracker will send its current location to application when user sends “track my device” message to the tracker. Android application is created by using android studio and only able to function in android devices version 5.0 or above. SQLite database is used for storing user information

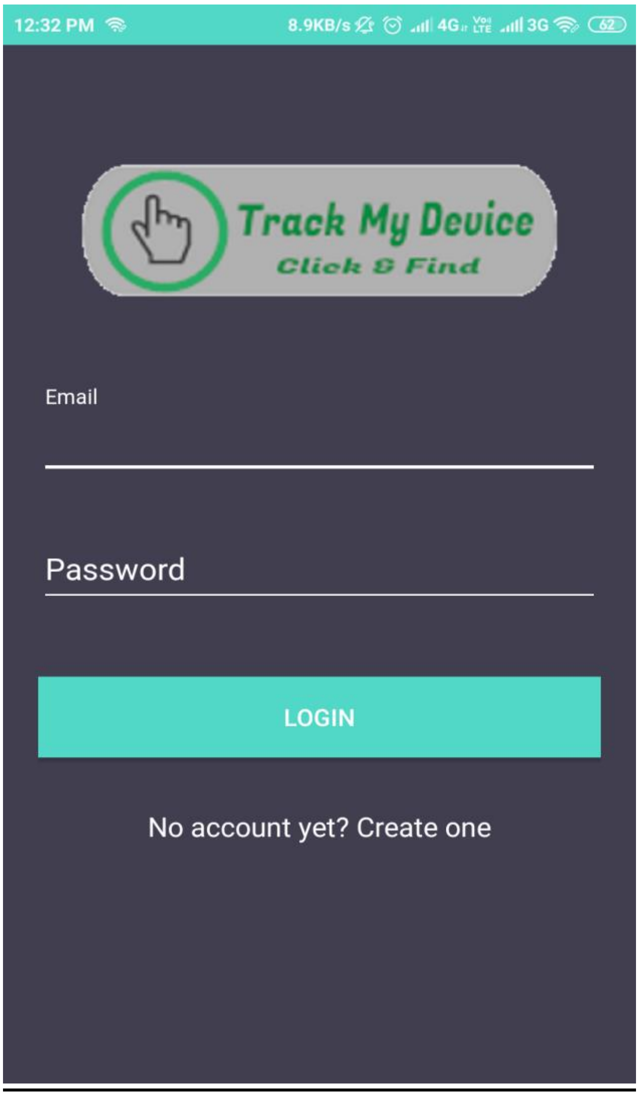


Fig 5.9 Login Screen

This is the login screen of the application. When the app is open the login screen will appears if the user is not logged in then, if the user is already logged in then the home page of the application will occur. For login we takes to parameter;

1. User Name/ Email Address
2. Password

User name or email is used to identify the user and no duplicate user is allowed to log in, means duplicate values like same name is not allowed. User name should be unique and password must be of 6 character long.

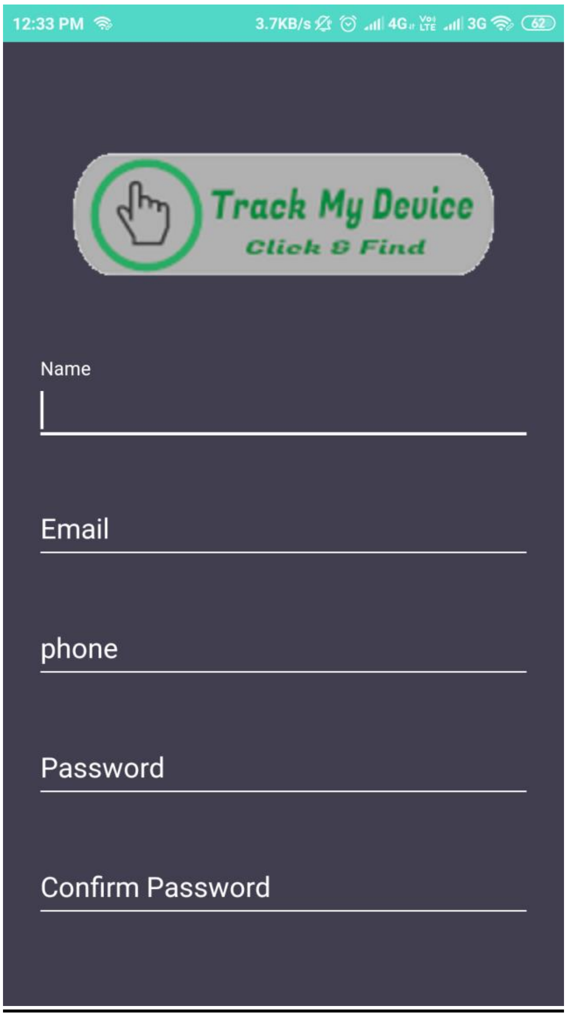


Fig 5.10 Registration screen

If user has not created the account then he first need to create the account by clicking on the link ”No account yet? Create one”. After clicking that link the user will be redirected to the registration form page where he/she needs to fill the following details. After filling the details correctly user need to sign in to the system.

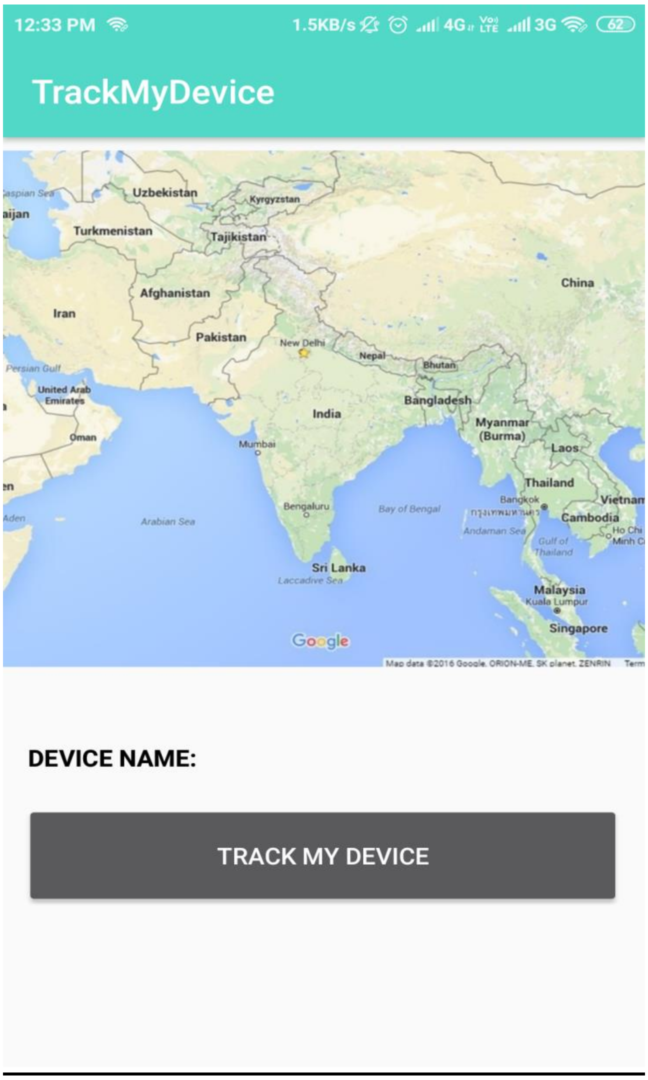
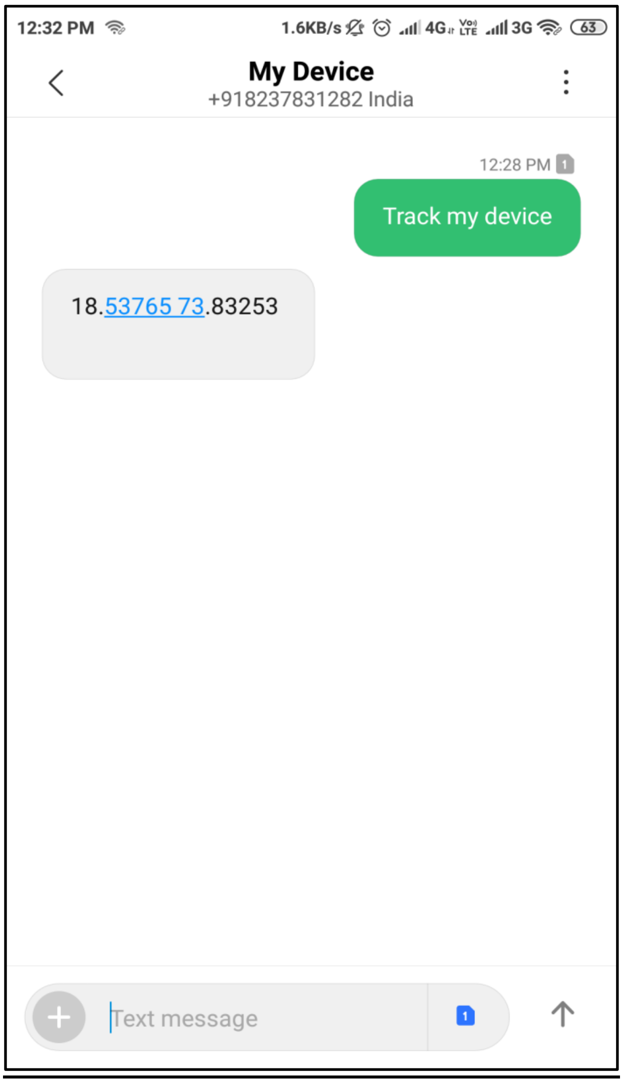


Fig 5.11 Main Screen

This is the main screen. Once the user is logged in then the home screen or main screen will occur which has two things;

1. Device Name
2. Track my device Button

After loggin to the system the app will show the device name which user has registered for the device. If user wants to track his/her device then he/she simply needs to click on track my device button, then the device will send the coordinated in the form of message to the registered number as shown in fig 5.12.



My Device
+918237831282 India

12:28 PM

Track my device

18.53765 73.83253

Fig 5.12 Receiving coordinates

After clicking on track my device button the device will send the coordinated to the registerd number as shown below. This coordinated are then fetch by the application in the form of longitude and latitude, the space between the coordinates are use to separate the longitude and latitude. This coordinates are fetch by the application to find the location of the device as shown in fig 5.13.

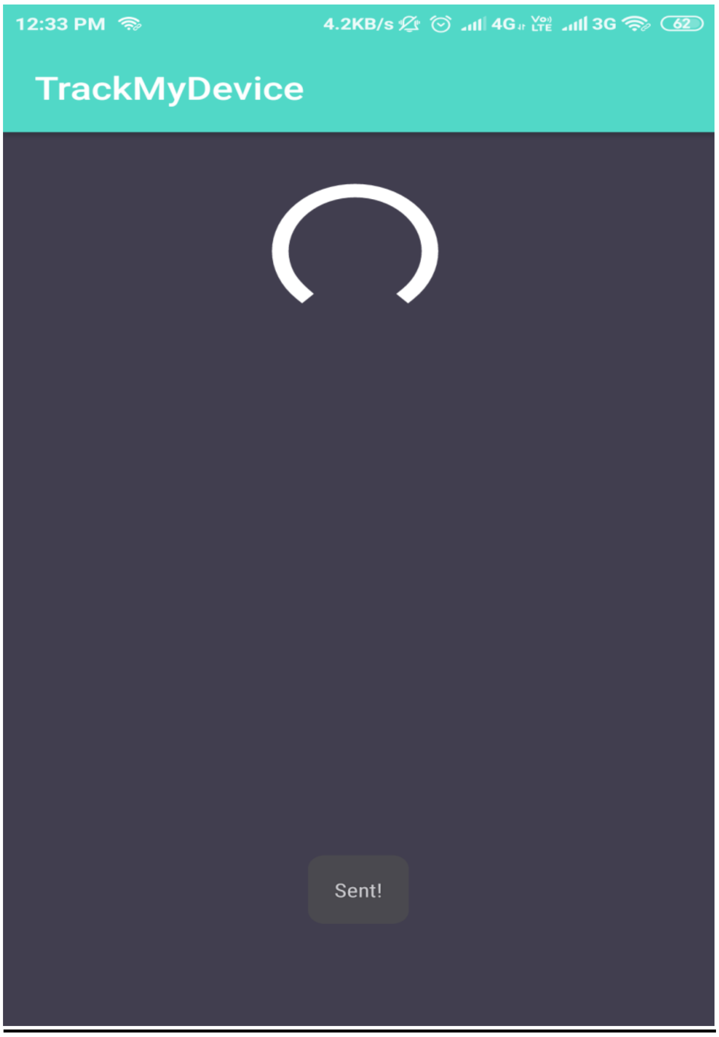


Fig 5.13 Fetching the coordinates

When device sends the coordinates to the registered number then our app fetches that coordinates in longitude and latitude. The following image show the fetching work which is running in background and the “sent!” toast messgae indicates that the message is send to the number. After fetching the coordinates then app redirect to the location of the device.

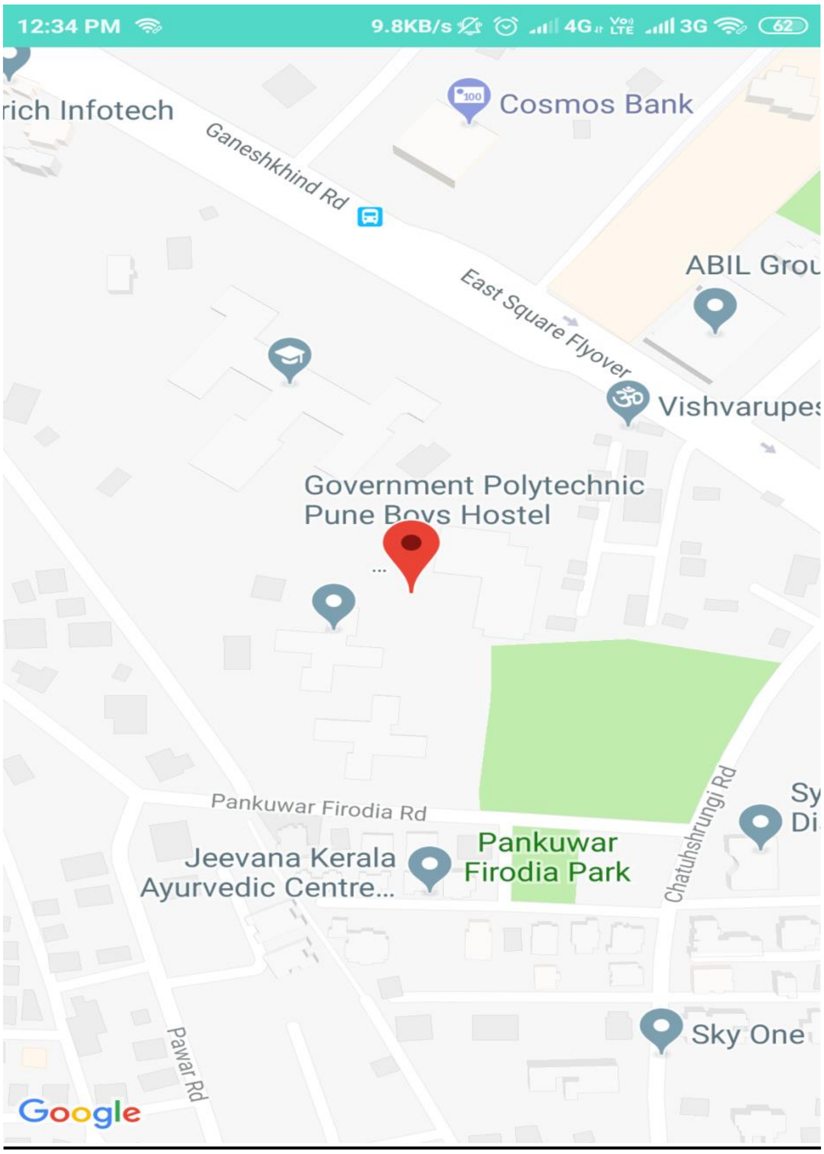


Fig 5.14 Location

After fetching the coordinates form inbox, app separate the coordinates in longitude and latitude. When user Clicks on find my device then the location of the device is shown, as shown in the above fig 5.14. This fig show the current location of our device which is checked while testing the software.

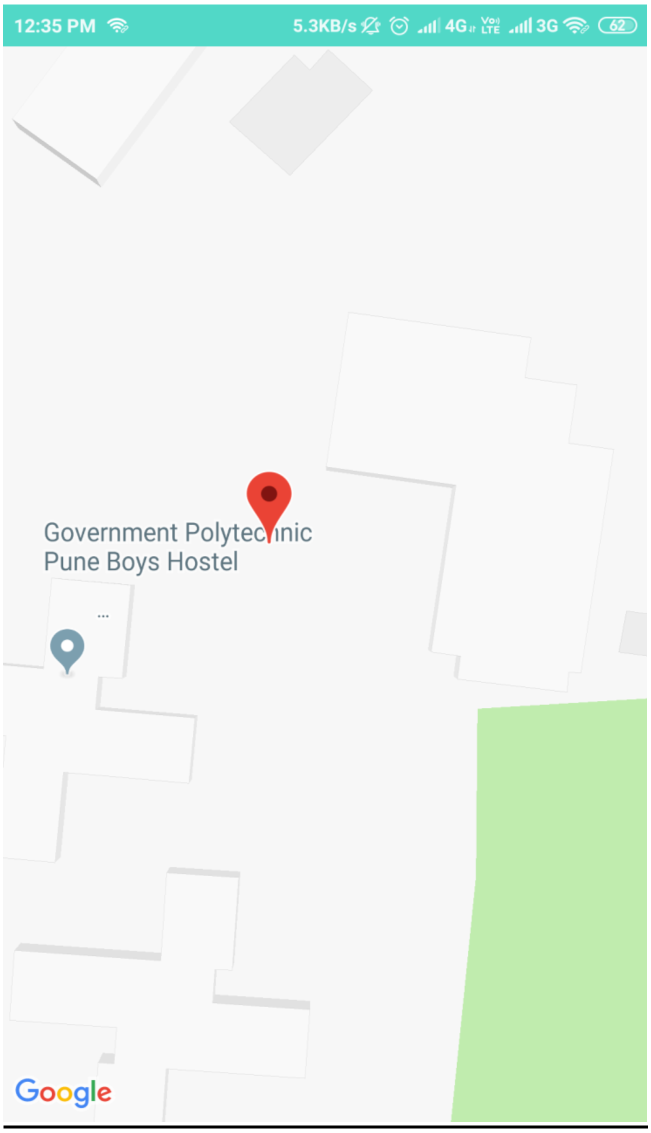


Fig 5.15 Location(zoomed)

System Testing

5.1 Testing Plan:

Sr. No.	Type of Test	Description	Software Components
1	Requirement Testing	This testing is required because we need to verify whether our requirements are able to solve the current problem or not	Complete Software including GUI
2	Unit testing	This testing allows us to test individual modules before integrating them together to form a single software	Data Pre-processing
3	Integration	This test is important to check whether the modules are giving the same results after integration as before	All adjacent modules
4	Performance	This test is important to calculate the efficiency of the software also helps us to find any performance issue related to the system	All the software components individually
5	Security	We have performed this test to check whether privacy is maintained	Wave data
6	Volume	We have performed this test to judge the embedding in a better way	Wave data pre-processing
7	Compliance	This test is performed in order to check whether we are implementing and meeting the defined standards	GUI Components

Table No. 2 – Testing Plan

Conclusion

Provide more delicate and easy to use. Tracker helps to track the device as fast as possible. Location is just given in 4 to 5 seconds. This system will help us to find the device in more efficient manner. As some device which require net access to track the location of the device, in this device there is no need for giving the net access. The coordinates are send via SMS to the user’s registered number.

Future Scope

- By adding acolometer we can continuously track the location of the device.
- Battery capacity can increased which will help us for more sustainability of device.
- We can use external battery source from other devices battery such as laptop, cars, etc.
- Location history can be stored.
- More security function will be provided.
- By adding motion sensor current state of device can be determine.
- Motion sensor would alert the owner that the device is moved.
- Vehicle tracking system makes better fleet management and which in turn brings large profits.
- Better scheduling or route planning can enable you handle larger jobs loads within a particular time.
- Vehicle tracking both in case of personal as well as business purpose improves safety and security, communication medium, performance monitoring and increases productivity.
- So in the coming year, it is going to play a major role in our day-to-day living
- Main motto of the project is to incorporate different types of sensors so that they help in decrease the chances of theft which we can stop from occurring.

References

- IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 51, NO. 2, APRIL 2004
- international journal of recent scientific research vole 6,issue,6, pp.4805-4808,,june 2015[4806],
- [IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015]
- [1 st IEEE International Workshop on Machine to Machine Communications Interfaces and Platforms 2013]
- [IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 51, NO. 2, APRIL 2004]
- [IEEE SENSORS JOURNAL, VOL. 7, NO. 7, JULY 2007]
- www.arduino.com/about ¬ www.cooking-hacks.com