International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 325 - 330

Real Time Vehicle Tracking on Google Maps using Raspberry Pi Web Server

Dhiraj Sunehra
Department of Electronics and Communication Engineering
Jawaharlal Nehru Technological University Hyderabad
Telangana, India
e-mail: dhirajsunehra@yahoo.co.in

Abstract**—**The use of automobiles is necessary to ship goods and products from one place to another. The users face several problems due to delay in the delivery of goods. This delay may be due to choosing wrong or longer routes by the driver. The Global Positioning System (GPS) is being used for fleet management, stolen vehicle recovery, surveillance and mapping applications. In this paper, implementation details of a vehicle tracking system (VTS) on Google Maps are presented along with experimental results. The vehicle tracking device (VTD) placed in the vehicle consists of a GPS receiver to acquire geographic coordinates, and a GSM/GPRS module to send the vehicle coordinates to a web server. The GPS and GSM modules are controlled using the Arduino microcontroller. Raspberry Pi is used as a web server to store the vehicle position and display it on the Google maps.

Keywords*- Automobile tracking; GPS/GSM/GPRS technology; Raspberry Pi; Arduino Uno; Google maps API*

*****

I. INTRODUCTION

With the advancements in technology, automated vehicle tracking systems (VTS) have become very prominent. The implementation details of a VTS that can display the vehicle position on Google maps, is presented in this paper. The GPS, GSM/GPRS modules controlled by a Arduino microcontroller are placed inside the vehicle. The vehicle position obtained from the GPS module is sent to a remote web server using GSM. The Raspberry Pi microcontroller board is used as a web server and it displays received vehicle coordinates on the Google maps. The vehicle position is updated every 60 s as the vehicle is moving.

Several authors have reported in literature their work related to vehicle tracking. Aravind et al proposed a low cost VTS which uses wireless sensor technology [1]. The Qualnet network simulator is used to perform simulations for different scenarios. SeokJu Lee et al developed a smart phone application for real time vehicle tracking. Google maps API is used to locate the vehicle in a map using the smart phone application [2]. Hoang Dat Pham et al used GPS and GSM technologies to transmit the vehicle location to user’s phone [3]. Moloo and Digumber developed a web application using PHP and MySQL for GPS tracking using low-priced mobile phones. Google maps API is used for location visualization [4]. Almomani et al proposed a GPS vehicle tracking system which sets speed and geographical limits [5]. Manoharan proposed a software architecture to locate lost and misplaced devices [6].

The system proposed in this paper helps in monitoring the vehicle location from anywhere in the world through the Raspberry Pi based web server. Displaying the coordinates on Google maps enables the users to understand the location of vehicle easily.

II. HARDWARE IMPLEMENTATION

Figure 1 shows the block diagram of VTS. It contains a VTD consisting of GPS receiver, Arduino microcontroller and GSM module, Raspberry Pi (server) and user laptop. The GPS receiver and GSM modules are interfaced to Arduino microcontroller.

A. Salient Features of Arduino Microcontroller

The Arduino Uno is an open source, extensible software microcontroller containing ATmega328. Arduino board can communicate with software running on our computer. It has 14 digital I/O pins, 6 analog inputs, a USB, power jack, and reset button (Fig.2) [7]. It can be attached to a computer using a USB cable. The I/O pins accept or provide digital or analog voltages between 0 and 5V. Arduino receives the vehicle coordinates from GPS and sends them to GSM/GPRS module.

B. Vehicle Location Identification using GPS Receiver

A GPS receiver can compute user position in three dimensions using the measurements of pseudorange from a minimum of three satellites and their satellite positions. As the receiver and satellite clocks are not completely synchronized; a fourth satellite data is needed to compute the receiver’s clock offset. The GPS receiver module provides raw data which can be converted to the desired format for post-processing. Almost all receivers generate GPS data in National Marine Electronics Association (NMEA) format. Each NMEA sentence is prefixed with a $ symbol, a 4-5 letter code signifies the information contained within the sentence and comma separates each value. NMEA data provides various output messages. One of the commonly used NMEA sentence is GPRMC which provides the GPS position, velocity, and time information. The receiver position is computed relative to World geodetic System 1984 (WGS84) ellipsoid.

A sample $GPRMC sentence is as follows:

$GPRMC,114016,A,1736.798,N,07852.073,E,026.8,085.5,230215,003.4,W*6A

The description of various fields in the above GPRMC sentence can be found elsewhere [8]. From the data, the following two fields are extracted by the microcontroller:

1736.798,N Latitude 17 deg 36.798' N
07852.073,E Longitude 78 deg 52.073' E

Figure 3 shows the interfacing of GPS receiver with the Arduino microcontroller.

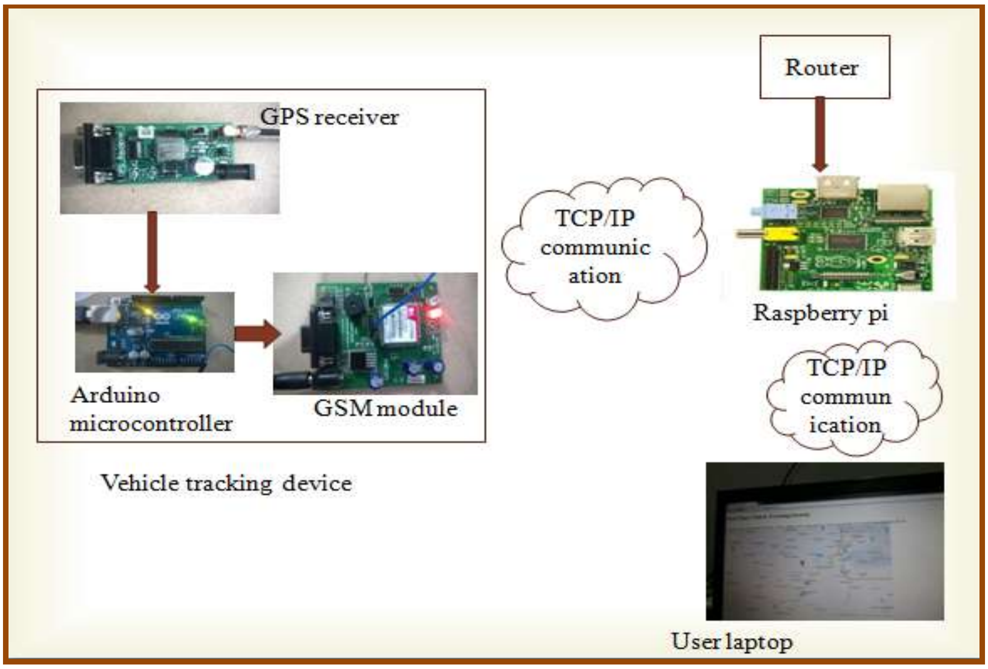

Figure 1. Block Diagram of Proposed Vehicle Tracking System using Raspberry Pi Web server

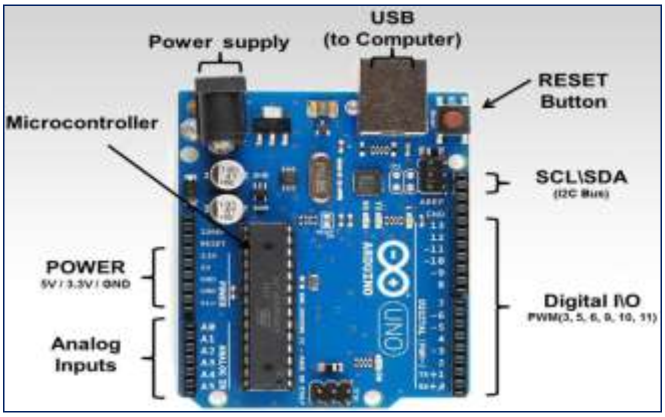

Figure 2. Arduino ATMega328P Microcontroller board

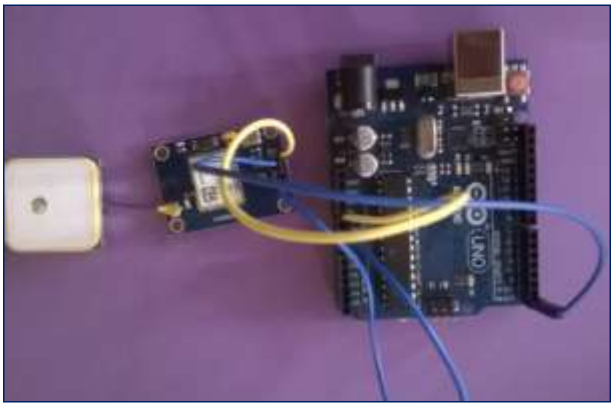

Figure 3. GPS receiver interfacing with Arduino board

C. *Features and Interfacing of GSM/GPRS Module*

The SIM900A is a Global System for Mobile Communication (GSM/GPRS) modem. It can be used for mobile voice, SMS, data transmission and consumes less power consumption [9]. GSM SIM900A module shown in Fig. 4 is used to send vehicle location to Raspberry Pi web server using GPRS technology. GSM SIM900A module's receive (Rx) pin is connected to pin 3 of Arduino board, and ground pin (GND) is connected to GND of Arduino.

Figure 4. GSM SIM900 module

D. *Salient Features of Raspberry Pi Model B+ Board*

The Raspberry Pi is a small but powerful ARM based microcontroller board. The Raspberry Pi uses a Broadcom BCM2835 system on chip (SoC), which has an ARM1176JZF-S700MHz processor, Video Core IV Graphics Processing Unit (GPU), and 512 MB of RAM. As it does not have a built-in hard disk, a SD card slot is provided for booting and storage. There are USB ports, SD card slot, audio jack, HDMI slot and GPIO pins on the Raspberry Pi [10]. The powerful graphics capabilities and HDMI video output make it suitable for multimedia applications. The Raspberry Pi model B+ board is shown in Fig. 5.

Raspberry Pi is good to build a cheap web server to be used as a testing environment or for storing files. In this work, a Wi-Fi router is used to access the internet for Raspberry Pi. The program for reading the vehicle GPS coordinates and

ISSN: 2321-8169
325 - 330

storing the values in a text file is written in PHP language and stored in Raspberry Pi memory.

Figure 5. Raspberry Pi model B+ board

III. SOFTWARE TOOLS

Various software tools such as Arduino Integrated Development Environment (IDE), Extra PuTTY and PHP language are used in the implementation. In Arduino IDE, software code is written to read and send the GPS values to the web server and is dumped into the Arduino board. Extra Putty is used to communicate with the server. PHP is used to write the web programs to display the vehicle position on Google maps.

A. *Arduino Development Environment*

The Arduino is connected to the computer using USB. The Arduino community calls a program as a sketch [7]. Arduino IDE contains a text editor used to write the program in C/C++, and after compilation, the program is dumped in to the board. Arduino IDE tool sketch is shown in Fig. 6.

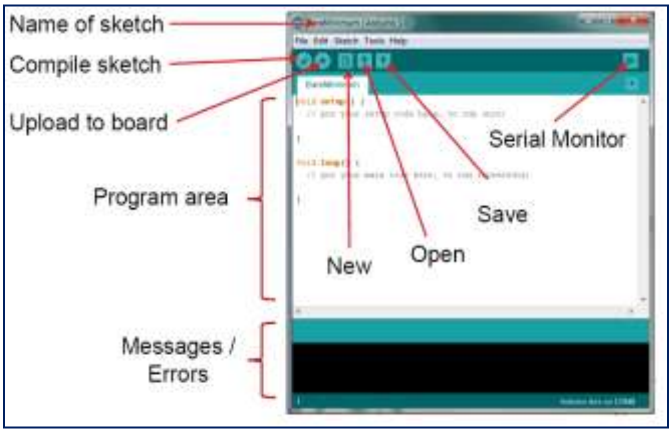

Figure 6. Arduino IDE Tool Sketch

B. *ExtraPuTTY*

ExtraPuTTY is a software tool used to initialize Raspberry Pi board, by entering the IP address of the board. It supports several network protocols, Secure Copy (SCP), Secure Shell (SSH), Telnet, rlogin, and raw socket connection [11]. It can also connect to a serial device via serial port. Figure 7 shows how to initialize a Raspberry Pi by entering IP address of the device in ExtraPuTTY. After entering into Raspberry Pi with IP address, one has to login with already created directory and with password as shown in Fig. 8.

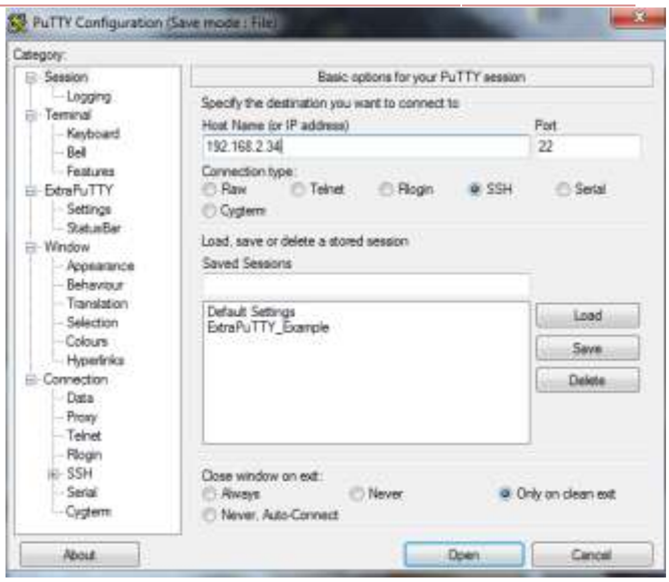

Figure 7. Initializing Raspberry Pi board using PuTTY

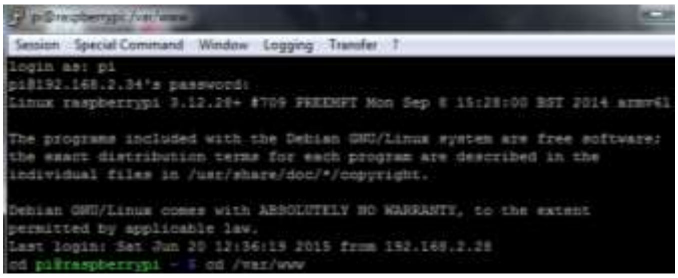

Figure 8. Login into Raspberry Pi board using PuTTY

C. *PHP Language*

PHP is an open source general-purpose scripting language which can be used for web development and can be embedded into HTML. The abbreviation for PHP now is: Hypertext Preprocessor. When the PHP code is interpreted and executed, the web server sends the resulting output to its client. The distinguishing feature in PHP with respect to client-side JavaScript is that the code is executed on the server that generates HTML, which is then sent to the client. The client receives the results of running that script, but doesn't gets information about the underlying code [12].

IV. EXPERIMENTAL RESULTS

A. *Experimental Setup and Flowchart*

Figure 9 shows the experimental setup of in-vehicle tracking device (VTD). The GPS and GSM modules are interfaced to the Arduino microcontroller board. A 12V rechargeable battery is connected to the Arduino microcontroller and GSM module. The program for receiving the GPS coordinates and to establish the TCP/IP communication through GSM/GPRS, to send these values to the remote server is written in Embedded 'C', tested using Arduino IDE software and dumped into the microcontroller. The software is programmed to send the vehicle location every 60 seconds, so as to update the vehicle location on Google maps.

Figure 10 shows the experimental setup at user monitoring site. It includes Wi-Fi router, Raspberry Pi and user laptop. Wi-Fi router is used to access internet for Raspberry Pi. The program for reading the vehicle coordinates and storing these values in a text file is written in PHP language and stored in Raspberry Pi memory. Also the code for reading the text file values and displaying the vehicle location on Google maps is written in PHP language. Figure 11 shows the flow chart depicting various events that takes place in the operation of the VTS.

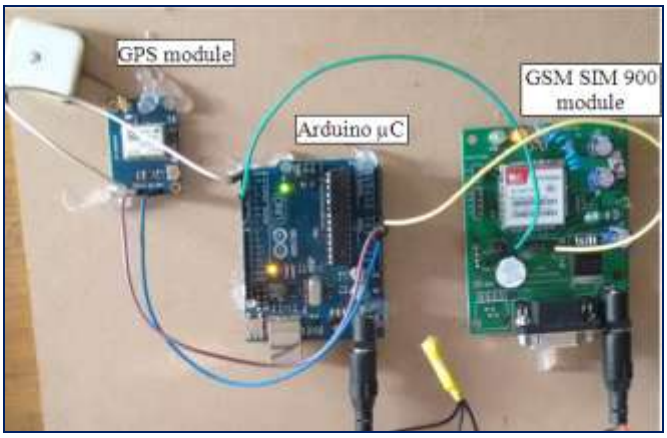

Figure 9. Experimental Setup of Vehicle Tracking Device

Figure 10. Experimental setup at User Monitoring Site

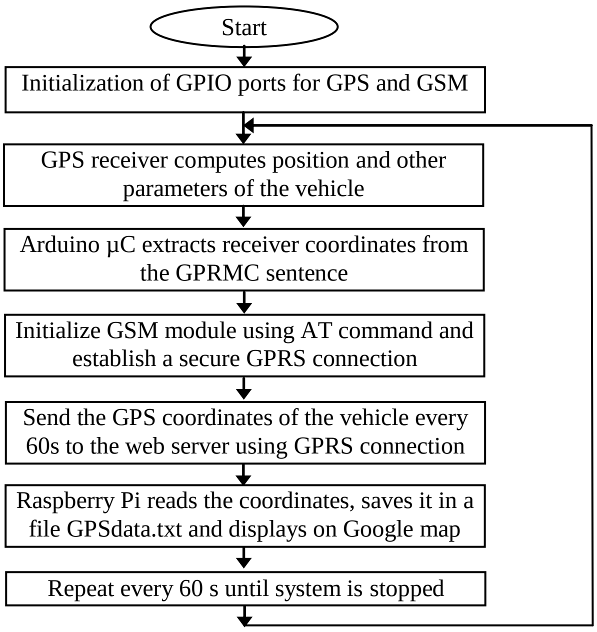

Figure 11. Flowchart showing sequence of events that take place during the operation of VTS

B. Experimental Results

Figure 12 shows the window displaying GPS coordinates stored in the Raspberry Pi. The latitude and longitude values which come from the VTD are stored in a text file, i.e. home_auto.txt file. As the vehicle is moving, these values are overwritten in that text file over a period of time. Also, one more text file is created to store all the values which are coming from the VTD, i.e. GPSdata.txt file.

Figure 13 shows the GPS coordinates stored in the file GPSdata.txt. To display vehicle location on Google maps, code is written in PHP language. This PHP program reads the GPS coordinate values from text file and maps these values as vehicle location on Google maps. By entering the Raspberry Pi IP address and the Google maps file name (google.php) in the browser, we can obtain the vehicle location.

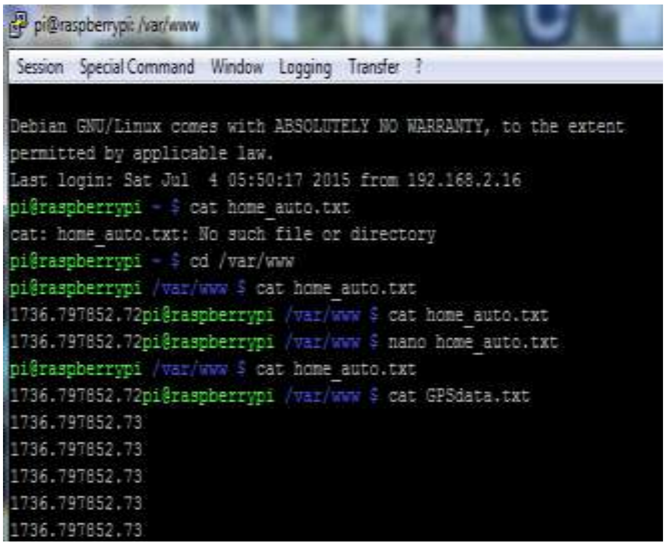

Figure 12. Flowchart showing sequence of events that take place during the operation of VTS

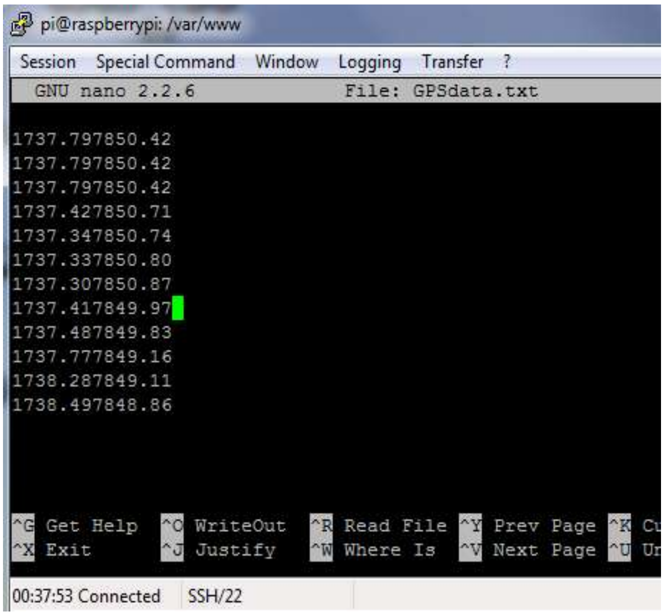

Figure 13. Reading the GPS coordinates in Raspberry Pi

A vehicle tracking experiment is conducted by traveling in a car from Dilsukhnagar to Koti in Hyderabad, India and the coordinates are monitored along the route at various points

ISSN: 2321-8169
325 - 330

using Google maps. Figures 14-18 shows Google map location of the vehicle at various locations along the route, viz. Sai Baba Temple Road (Dilsukhnagar), Malakpet police station, Nalgonda crossroads, Esamia bazar road, and Koti.

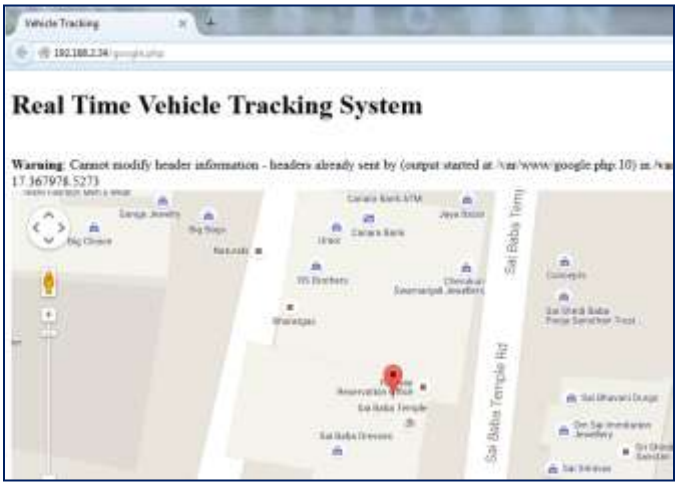

Figure 14. Vehicle location displayed on Google maps when it is near Sai Baba temple road, Dilsukhnagar

Figure 15. Vehicle location displayed on Google maps when it is near Malakpet Police Station

Figure 16. Vehicle location displayed on Google maps when it is near Nalgonda crossroads

The GPS coordinates, viz. latitude and longitude are updated in text file (home_auto.txt) for every 60 seconds and the corresponding vehicle location will be displayed on the Google map by refreshing the browser.

Figure 17. Vehicle location displayed on Google maps when it is on Esamia Bazar Road

Figure 18. Vehicle location displayed on Google maps when it is near Koti

V. CONCLUSIONS AND FUTURE SCOPE

The implementation and experiment details of a vehicle tracking system on Google maps using Arduino and Raspberry Pi microcontrollers are presented. Arduino microcontroller is used in the VTD and transmits the GPS coordinates to a Raspberry Pi based remote web server. The program to receive GPS coordinates and send these values to the web server by establishing TCP/IP communication is written in Embedded 'C' in Arduino IDE. The vehicle's geographic coordinates from the VTD are stored in a text file in Raspberry Pi. The vehicle's GPS coordinates are updated for every 60 seconds in the server and the corresponding location is shown on Google maps. Prototype of the VTS is implemented using low-cost electronics and tested for satisfactory operation.

This implementation can be extended for tracking of multiple vehicles at the same time. An android application could be developed to show the vehicle location in a smart phone. Also, a subscription system could be developed to send alerts to subscribers, when vehicle is at a certain distance or time from the user location.

ISSN: 2321-8169
325 - 330

REFERENCES

[1] K.G. Aravind, Tapas Chakravarty, M. Girish Chandra, P. Balamuralidhar, "On the architecture of vehicle tracking system using wireless sensor devices", International Conference on Ultra Modern Telecommunications & Workshops (ICUMT), St. Petersburg, Oct. 12-14, 2009, pp.1-5, IEEE.

[2] SeokJu Lee, Girma Tewolde, Jaerock Kwon, "Design and Implementation of Vehicle Tracking System using GPS/GSM/GPRS Technology and Smartphone Application", IEEE World Forum on Internet of Things (WF-IoT), Seoul, March 6-8, 2014, pp.353-358.

[3] Hoang Dat Pham, Micheal Drieberg, Chi Cuong Nguyen, "Development of Vehicle tracking system using GPS and GSM modem", IEEE Conference on Open Systems (ICOS), Kuching, Dec. 2-4, 2013, pp.89-94.

[4] Raj Kishen Moloo, V.K. Digumber, "Low-Cost Mobile GPS Tracking Solution", International Conference on Business Computing and Global Informatization (BCGIn), Shanghai, July29-31, 2011, pp.516-519.

[5] Iman M. Almomani, Nour Y. Alkhalil, Enas M. Ahmad, Rania M. Jodeh, "Ubiquitous GPS vehicle tracking and management system", IEEE Jordan Conference on Applied Electrical and Computing Technologies (AEECT), Amman, Dec.6-8, 2011, pp.1-6.

[6] S. Manoharan, "On GPS Tracking of Mobile Devices", International Conference on Networking and Services (ICNS), Valencia, April 20-25, 2009, pp.415-418, IEEE.

[7] Tenere Technologies (P) Ltd., "Embedded System Design using Arduino" (www.tenere.in).

[8] G. Rajendran, M.Arthanari, M. Sivakumar, "A Simplified NMEA Sentence Generator for the Simulation of GPS Tracking", Global Journal of Computer Science and Technology, Vol. 10, Issue 14, November 2010, pp. 19-26.

[9] Rhydo Technologies (P) Ltd., "SIM900-RS232 GSM/GPRS Modem User Manual", Cochin, Kerala, India, 2011 (www.rhydolabz.com).

[10] Adafruit Learning System, "Introducing the Raspberry Pi Model B+", March 2015 (https://learn.adafruit.com).

[11] https://en.wikipedia.org/wiki/PuTTY.

[12] http://php.net/manual/en/intro-whatis.php.