# Cost Effective Real Time Vehicle Location and Status Monitoring System

Fahad Mirza, Masum Rana

***Abstract*— Location Based Services (LBS) has become prevalent and indispensable over the last few years. Application ranges from military to personal security and comfort. It has been using in vehicles, for fleet management, car theft situations etc. But all these services are expensive, in terms of, the device itself and the monthly service charge. The devices consist of expensive components. And continuous transmission of data from device to server increases data transmission cost. In this paper we try to build such device from scratch, composed of basic hardware, to reduce the device cost and implement an algorithm to reduce the number of data transmission, while continuing record accurate representation of real time vehicle position and status of vehicle by communicating with ECU, in server side. A web application has been developed using open source softwares and free version of Google Maps API. It can send Geofence alerts through SMS. The device has a memory flash, to temporary store location information, and an 8-bit AVR microcontroller with bootloader, to update its firmware on the air anytime, anywhere.**

***Index Terms*— LBS, Real Time Tracking, GPS, GSM, GPRS, ECU, NMEA, Google Map, PHP, MySQL, Latitude, Longitude, OBD-II, Bootloader.***

## I. Introduction

Effective transportation and logistics has becomes a very important part of business due to the continuously increasing of oil price. Several efforts have been taken to deploy methods of making transportation even more efficient to reduce cost. Automatic vehicle location (AVL) is one of the methods to reduce the cost by knowing in real-time the current location of a vehicle. Also the current and historical movement of people and equipment is a useful tool, when looking to develop more efficient work practices by removing wasted time or movements [1].

Real-time tracking and management of vehicles has been a field of interest for many researchers [4 – 9]. Some based on GIS environment [4] [5], some systems use short message service (SMS) technology [7 – 9]. But the operation cost of most of these systems is higher which prevents from widespread use. The objective of this research is to reduce the cost of the tracking system using the latest technologies and making it available to the common people.

One of the reasons of high cost is data update method. There are many methods in existing systems. In polling methods, device updates data when server sends the request. Polling is useful for occasional data display, but not feasible for real-time application. Periodic data update from device to server is more common approach in data update method. But in this approach a significant amount of unnecessary data can be sent to the server if small interval value is use. On the other hand if we use large interval values then most of the data will miss. So update value should be dynamic. In the proposed algorithm we try to merge all the factors which will update only crucial points.

Supervise and record engine's running status parameters such as fuel injection pulse width, fuel injection advance angle, EGR valve opening, etc. will make great sense to evaluation and optimization of engine control strategies which lead to improve fuel economy. ECU [10] controls these parameters. All cars built since January 1, 1996 [11] have On-Board Diagnostics (OBD-II) [12] systems. OBD-II provides access to data from the ECU.

Along with tracking, an effective ECU monitoring system is developed in this paper for remote fault diagnosis. This system is fairly convenient and meanwhile guarantees protocol adaptability, quality of communication and security of data.

The rest of the paper is as follow: In Section II, we propose the system architecture. Section III shows implementation of the system prototype and some screenshots of our work. In Section IV, we analyze the result and discuss the low cost solution. We conclude with our achievements, limitations and future work in Section V.

## II. System Architecture

Figure 1 shows basic principles of our system. The system is composed of three parts – a tracking device, a server part and a web application.

### *A. GPS Tracking*

The tracking device comprises of three main modules: the GPS module, the GSM module, and the backend system that collects the information and process it. The tracking unit is designed to be powered by the vehicle battery. However, a Li-Ion battery is built into the device as an emergency backup.

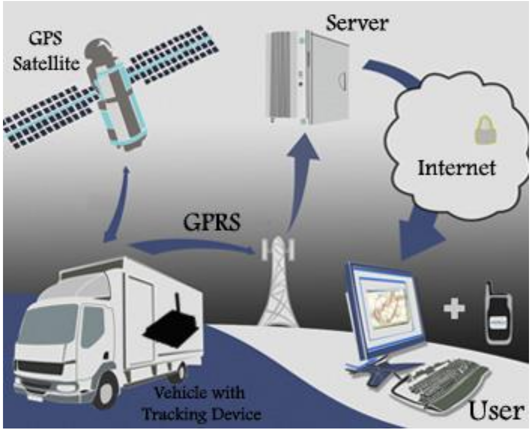

Figure 1. System Architecture

The device is attached with the vehicle and gets the position from GPS satellite in real-time. CPU packages GPS and status information along with the International Mobile Equipment Identity (IMEI) [13] number as its own identity to the server, into complete data packets, and sends them to wireless module.

*B. Server*

The server is a personal computer that receives this information. When server has analyzed the data received, it saved the data into database. The database formats the information in a special form that can search and display using Google Map.

In addition, the database provides the ability to dial up with a multi-user to trace a fleet of vehicles at the same time. This makes the tracking software more reliable and efficient.

*C. Web application*

Our system provides visualization of routes/tracks via a front-end of a web-based interface with details such as speed and direction heading to etc. The features are:

**1. Live Tracking**: It allows any registered user to have a real time Google Map view of the position of devices corresponding to the user. It can allow simultaneous view of several devices at the same time or a single device view.
**2. Static Recorded Tracking:** Recorded track on a user's device can be statically viewed on Google Maps view based on time and date.
**3. History Playback (Animated Tracking):** Via Google maps and the web-based interface, the users have an animated view of a track previously carried out by the device.
**4. Track Reports:** Reports and charts for each device and each track can be downloaded in PDF and Excel format. Tracks can be visualized on Google Charts with information such as speed, altitude, time, coordinates and name of location.
**5. Alerting Services using SMS and Email Alerts:** Users can also set alarm parameters.

## III. Design and Implementation

Our device can be divided in to five modules: central control, wireless communication, GPS, memory, vehicular communication access. Central control module is responsible for coordinating all modules [Figure 2].

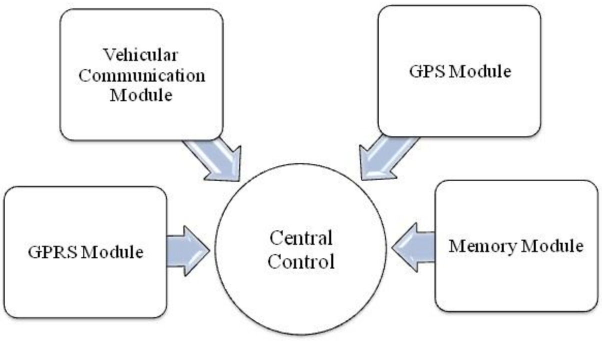

Figure 2. Hardware Configuration of the device

Based on system structure, main functions of the device are:
**1. Network Management:** Log on to GPRS network, connect to server, test network connection, maintain network state, re-connect when lose connection, log off, etc. Most of these actions are handled automatically.
**2. GPS (Global Positioning System):** Obtain GPS data through satellites.
**3. Data Collection:** Collect running state information (such as real-time speed of vehicle, mileage and fuel etc.) and update data of ECU.
**4. Data Storing:** In the absence of GPRS connection store the information into high capacity nonvolatile memory.
**5. Data Encryption:** To prevent malicious intrusion to system and consequent data interception, communication between device and servers must be encrypted. Security of data is of great importance to ECU for manufacturers.

*A. Hardware Specification*

For our system we use SIM548C [14] module, which contains GSM, GPRS and GPS module, all-in-one. It is a low cost module compare to others available in the market and capable of working in any GSM network around the world.

For central control we use a powerful 8-bit AVR RISC microcontroller, ATmega128A [15], which is a low power MCU with 128k bytes RAM. The SPI interface is connected with memory module. To meet the requirement of uninterrupted record, large-capacity memory chip is necessary. So we use ATMEL's 16 mega bit flash memory, AT45DB161D [16].

We also use Sparkfun's OBD-II to UART [17] [Figure 3] module to communicate with vehicles ECU.

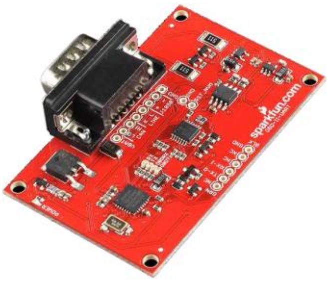

Figure 3. OBD-II to UART module

*B. Device Firmware*

The firmware of the GPS Tracking module is written and compiled using an open source AVR compiler, AVR Studio [18]. The firmware performs three phases, the initialization, the GPS position and vehicle status reading, and the collected data formatted and transmitted to server via GPRS networks.

In the GPS position reading phase, the GPS data pass through an algorithm (discussed later) before send them to the server. This will ensure the low cost. Then the NMEA-formatted data is parsed and convert to our own format. The format includes the device ID, session ID, time in UTC format, flags, latitude, longitude, speed and date. Then communicate with ECU and read vehicle status like mileage, fuel, engine temperature, rpm, pressure, also read if there any trouble codes and converts these to our own format. All these information, along with the IMEI number, are bundled together, ended with the character “\n”, and transmitted to a fixed IP address at a fixed port for specific server through GPRS.

To update the firmware anytime, anywhere, over the air, a bootloader [19] has been included in the microcontroller. Microcontroller’s Reset pin is connected with one of the GPIO pins of SIM548C. When we want to update the firmware, we send “AT+SGPIO” command through web application in transparent mode to GSM module. Module then pull down the GPIO pin connected to CPU’s Reset pin and activate the bootloader. Then we provide the hex file from server side to CPU through serial port (UART). A prototype of the board is showing in Figure 4.

Figure 4. Device Prototype

*C. Valid Point Algorithm*

The goal of this algorithm is to reduce redundant data updates, which will reduce bandwidth consumed and number of bytes transferred (i.e. cost to the users). After getting GPS data it checks the speed. If it is lower than a threshold speed (i.e. standing still) then data won’t update until the speed is higher than the threshold value. Once done, then it enters into second phase. If the device moving in a straight line then the points along the lines can be discarded because they won’t add any additional path information. In straight line only two points are needed, starting and ending point. All the points between them can be replaced with a connected line. It will nearly intersect all the points. Change in the direction is measured by difference in slope of two points [Figure 7].

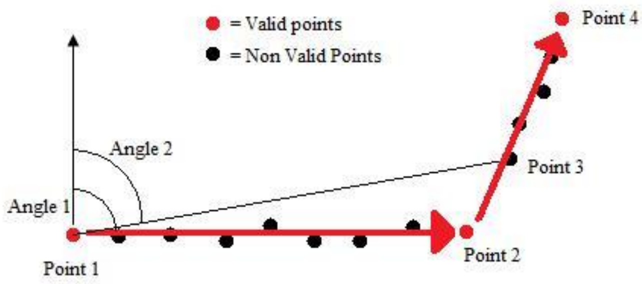


Figure 7. Calculation of valid points

The first data (Point 1) is a valid point so this sent to the server and set as Last Valid Point. By default the device start with a three seconds interval. After three seconds a new point will appear and go through the algorithm. But as there is not enough information to determine this point is valid or not, algorithm will return null output. Though it will save the point for future calculation to determine whether it is valid point or not. When the third point appears the difference in slope between first and second point and first and third point is calculated. If it exceeds the define threshold value then the third point will be update or else second point will be replace by third point as Last Point available. When there is a change in direction (Point 3) which surpasses the threshold value, the previous point (Point 2) will be the valid point and sent to the server, which is the end of the straight line. Now Point 2 will be the Last Valid Point and Point 3 will be the Last Point. If the device is in straight line more than a threshold time then a valid point is created in between the line. Also if the GPS data is invalid or has less accuracy then data won’t be update. This algorithm is executed every time when a GPS data is calculated.

*D. Software Specification*

To view the current position of the device a web based application has been developed. By this application user will be able to view the live and history tracking. PHP and Yii Framework [20] were used to develop the web application.

The database was created using MySQL [21]. It contains three tables. The devices table, the users table, and the data logs table. Each table has an ID. The devices table contains a list of tracking devices, the device name, and the user ID. The users table contains the devices users, and password. The data logs table contains the device ID, the UTC (Universal Time Coordinates) time and date, the position records for the tracking device (longitude, latitude, and altitude), speed and the vehicle status (fuel, mileage, engine temperature, rpm, pressure). The purpose of the website is to provide a web interface that can be used to view the vehicle current position information on Google maps, as well as view the other information. It can also analyze and monitor the vehicle status at specific time and date, or during a period of time. This will give a real time tracking of the vehicle on Google maps.

## IV. Results and Analysis

*A. Cost Effective*

Efforts have been taken to reduce the total cost of the system including device and services. By using a single module the device cost and by using Valid Point algorithm and free Google map API and HTTP protocol the service cost has been reduced dramatically. Starting from small business to large enterprise, the device and service cost can be affordable. They can also set up their own customized web-based remote monitoring system easily.

GPRS was used for data transfer instead of SMS. Several SMS based vehicle tracking systems were mentioned at the beginning. SMS based tracking system is neither efficient nor cost effective. In most countries the cost of GPRS is cheaper than SMS by a factor of 20 to 100.

To reduce the total system cost, a single GSM/GPRS/GPS module was used instead of separate devices. It requires only external power and an antenna for GPS and GSM. Beside the cost, this approach saves extra PCB space of the system. One part of the service cost was reduced by integrating free Google map into the service provider's site using the Google map API and customizing the service. Countries where there is a lack of proper GIS based map, it would be time consuming and extremely expensive to develop a map solution.

To test the algorithm, the device use in several trips initially without using algorithm i.e. the device sends the GPS data periodically. And then complete the same trips with using the algorithm. [Figure 5] clearly shows that the algorithm can detect the direction change i.e. valid points. Table 1 shows the estimate cost savings.

For the purpose of analysis, we also tested our system against one of the local system, Finder [22], over the same track. The difference is based on one primary factor namely the number of GPS positions the applications send to the server. It has been noted that for *Finder* application, GPS positions were sent to its server and from there that there was processing whether to store the data or not. But in our system calculation is handled in the device.

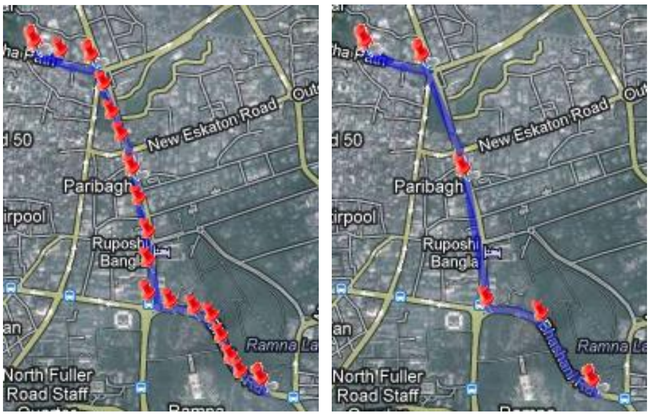


Figure 5. Comparing trip with and without algorithm

Table 1. Financial Savings by using algorithm

| Trip Number | Total Points | Total Valid Points | Saved Bytes | Financial Save |
|---|---|---|---|---|
| 1 | 102 | 34 | 7684 | Tk. 0.15 |
| 2 | 258 | 51 | 23391 | Tk. 0.45 |
| 3 | 813 | 126 | 77631 | Tk. 1.51 |
| 4 | 1342 | 165 | 133001 | Tk. 2.60 |
| 5 | 2405 | 222 | 246679 | Tk. 4.82 |

B. Experimental Testing

In order to test the vehicle tracking system, the tracking device prototype was installed in a car by connecting the power jack with a 12V power from the car battery. We track more than one vehicle at the same time: one at Faridpur, one at Narayanganj and one at Dhaka [Figure 6].

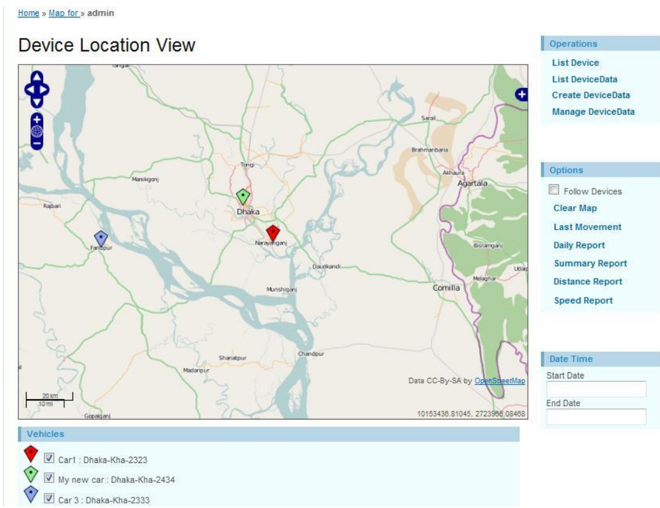


Figure 6. Live tracking more than one vehicle

The accuracy of the GPS position varies from place to place, but the difference was rather small for such a low-cost system. It actually matched the accuracy of *Finder* and the real track.

Any user could get a clear overview where the device has been. There was no big difference in the order of several meters at all, unless the device is in a building.

In addition to this, Geofence alerts were tested. One Geofence zone was set up at Uttara. When approaching to around 50 meters to that point, a text message was sent to a mobile number. The automatic logging was also tested by calling the device with another mobile phone, which blocks the GPRS connection. After ending the call all the stored data has been send to the server.

## V. Conclusion

In this paper we presents a cost effective vehicle tracking system with remote maintenance of vehicular electronic control, using commodity hardware and open source software, suitable for wide range of applications all over the world. The tracking process is within an accurate and acceptable range, and since it allows managers to supervise vehicle status (i.e. fuel, temperature and door status); the system provides reliable and precise information about the amount of work by all employees, so the administrator will make sure that his/her fleet is working in location and being monitored efficiently and effectively.

We reduce both initial investment cost as well as the operational cost in terms devices and services. We use a low cost module where GSM, GPRS and GPS integrated all together. By using free Google Maps APIs, HTTP protocol, intelligent logging, and an intelligent positioning calculation, the service cost has been reduced dramatically, providing most services provided by existing systems.

In future we will more focus on threshold values to evaluate most appropriate combination for the algorithm. Also the proposed system, though cost-effective, has some limitations. The system can give false GPS position, also no data at all, if confined in a building or under a bridge where there is no possibility to capture a GPS position. So in the future, we will plan to integrate Dead Reckoning (DR) [23] system, which has good positioning precision in short term and some sensors like accelerometer and gyroscope to calculate the position of vehicle in absence of GPS data.

## Acknowledgement

The authors would like to acknowledge the contribution of Mr. Kamruzzaman Kanok in making valuable suggestions in technical field.

## References

[1] McGlinchey, G.J., 2009. *Indoor Wireless Location Determination*. Computer Science Thesis. University of Ulster.

[2] A. Goel and V. Gruhn , “Fleet Monitoring System for Advanced Tracking of Commercial Vehicles”, Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2006), pp. 2517-2522, Taipei, Taiwan, 08.10.2006 - 11.10.2006

[3] M. Ahmad, et al, “Real time fleet monitoring and security system using GSM network”, Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbon, Portugal pp. 498-503, 2006.

[4] O. Aloquili, A. Elbanna, and A. Al-Azizi, ”Automatic vehicle location tracking system based on GIS environment”, Software, IET Volume 3, Issue 4, pp. 255 – 263, 2009.

[5] Al-Bayari, B. Sadoun, “New centralized automatic vehicle location communications software system under GIS environment”, *International Journal of Communication Systems*, Vol 18, Issue 9, April 2005, pp. 833–846

[6] Fan, X., W. Xu, H. Chen, and L. Liu, “CCSMOMS:A Composite Communication Scheme for Mobile Object Management System”, *20th International Conference on Advanced Information Networking and Applications*, Volume 2, Issue 18-20, April 2006, pp. 235–239

[7] ArcView Tracking Analyst: ‘Complete tracking solutions’. ESRI White Paper, ESRI Press, 1999

[8] Hapsari, A.T., E.Y. Syamsudin, and I. Pramana, “Design of Vehicle Position Tracking System Using Short Message Services And Its Implementation on FPGA”, *Proceedings of the Conference on Asia South Pacific Design Automation*, Shanghai, China, 2005

[9] Tamil, E.M., D.B. Saleh, and M.Y.I. Idris, “A Mobile Vehicle Tracking System with GPS/GSM Technology”, *Proceedings of the 5th Student Conference on Research and Development (SCORED)*, Permala Bangi, Malaysia, May 2007

[10] Wikipedia. What is ECU? Available: http://en.wikipedia.org/wi-ki/Engine_control_unit

[11] OBD-II, History of OBD-II, Available: http://www.obdii.com/b-ackgrou-nd.html

[12] Wikipedia. What is OBD? Available: http://en.wikipedia.org/wi-ki/Onboard_diagnostics

[13] Network System Architects, Inc. What is an IMEI? Available: http://www.nmea.org/pub/ 0183/ index.html

[14] SimCom, SIM548C Module, Retrieved 25/3/2008, Available: *www.simcom.us/act_admin/supportfile/SIM5X8C_HD_V1.01.pdf*

[15] ATMEL. ATmega128A Microcontroller. Available: www.atme-l.com/Images/doc8151.pdf

[16] ATMEL. AT45DB161D Flash memory. Available: www.atme-l.com/Images/doc3500.pdf

[17] Sparkfun Electronics. OBD-II UART Module. Available: http://www.sparkfun.com/products/9555

[18] ATMEL. AVR Studio 5. Available: http://www.atmel.com/mic-rosite/avr_studio_5/default.aspx

[19] Fischl. AVRUSBBoot. Available: http://www.fischl.de/avrusb-boot/

[20] Yii Framework. What is Yii Framework? Available: http://www.yiiframework.com/

[21] MySQL AB. MySQL Documentation, Available: http://dev.my-sql.com/doc/

[22] Monico Ltd. Finder-lbs. Available: http://www.finder-lbs.com/

[23] Hong-Jiao Ma, Yong-Hui Hu, Hai-Bo Yuan, Wei Guo: Design and Analysis of Embedded GPS/DR Vehicle Integrated Navigation System', International Conference on Embedded Software and Systems Symposia, July 2008, pp. 265 - 270