Integrating Social Network Services with Vehicle Tracking Technologies

Ahmed ElShafee
Assistant Professor
Ahram Canadian University
6th October, Egypt

Mahmoud ElMenshawi
B.Sc. Computer Science
Ahram Canadian University
6th October, Egypt

Mena Saeed
B.Sc. Computer Science
Ahram Canadian University
6th October, Egypt

ABSTRACT

This paper gives design, and implementation of a newly proposed vehicle tracking system, that uses the popular social network as a value added service for traditional tracking system. The proposed tracking system make use of Google maps service to trace the vehicle, each vehicle has an account that contains a posts of Google maps that display the vehicle location on real time mode. A hardware module is inside the vehicle that uses Global Positioning System (GPS) – to detect vehicle location- and Global system for mobile communication (GSM) – to update vehicle location in vehicle account on social network -. System uses the well-known Arduino microcontroller to control GSM-GPS Modem. The proposed system can be used for a broad range of applications such as traffic management and vehicle tracking/ anti theift system, and finally traffic routing and navigation. it can be applied in many business cases, like public transportation, so passengers can track their buses, trains, by following the vehicle account on social network. It also can be used in private business sector as an easy and simple fleet tracking and management system , or can be used by anyone who wants to track his car, or to find his way in case he get lost.

Keywords

Vehicle Tracking, GSM, GPS, Microcontrollers, Twitter, Google maps.

1. INTRODUCTION

1.1 Overview

Vehicle tracking systems [1], this term covers a range of products which, uses communications technology, or a combination of technologies, identify vehicle land report and, its real-time location and present this information to vehicle trackers on a remote server through internet. Vehicle tracking systems are commonly used by fleet operators for fleet management functions such as routing, dispatch, on-board information and security. Other applications include monitoring driving behavior, such as an employer of an employee, or a parent with a teen driver. Vehicle tracking systems are also popular in consumer vehicles as a theft prevention and retrieval device. Police can simply follow the signal emitted by the tracking system and locate the stolen vehicle.

social network sites (SNSs) [2] such as Twitter, Facebook, have attracted millions of users, many of whom have integrated these sites into their daily practices. Till now there are alot of SNSs, with various technological affordances, supporting a wide range of interests and practices. The great benefit behind SNSs that they help strangers connect based on shared interests, political views, or activities. They attract people based on common language or shared racial, sexual, religious, or nationality-based identities. Most SNSs incorporate new information and communication tools, such as mobile connectivity, blogging, and photo/video-sharing.

1.2 Features and benefits vehicle tracking system

Vehicle Tracking service [3] [4] is a GPS based solution that provides instant location information to the vehicle owner/authorized person through web/SMS with other flexibilities. The basic feature of popular vehicle tracking systems is locating real-time position of the vehicle. There are many other value added features and services, like applying rules on the vehicles (e.g.: speed limit, No Go Area, etc.), securing the vehicles by adding security features like remote immobilization, panic alarm. Some of the other features are: Speed Violation Alert/Report (speed of all vehicles can be controlled, monitored, and hence when violated, immediate contact can be made to reduce such violation). Area Alarm (an area can be assigned to vehicles, the owner/authorized person will be notified, if the rule is violated); 'No-Go' area (A 'No-Go' area can be created for the vehicles, the owner/authorized person will be notified, if the rule is violated); in addition, the car owner can find out whether the ignition is on or off; if on, then whether the vehicle is moving or stationary, thus vehicles can be monitored even if they are switched off.

1.3 Features and benefits of social network

Most social networking websites are supported by paid advertisements that appear on member pages. Because of this, most social networking websites do not carry membership charges and offer free services to all users. The main goal of social networking websites are designed to allow members to connect and communicate with one another - so features of these sites foster interaction, activity, and of course, community.

2. SYSTEM ANALYSIS

2.1 Problem definition

The proposed vehicle tracking system is an open system that uses a free and open source software and is composed of commodity hardware that is easy-to-find. Our system is composed of four components, a GPS/GSM Tracking Device, a web server with database, social network, and finally Map. The GPS tracking device is an embedded system that transmits location information to the server through GPRS networks. The server is a personal computer that receives the information and put it in the database. Twitter is selected as value added service to common vehicle tracking system, vehicle presented on the social network as virtual profile that users can simply follow vehicle account. Vehicle send tweets in regular bases, that contains a link to a map showing the current location of the vehicle.

2.2 Proposed system feature

The proposed vehicle tracking system provides the following features;

- Vehicle position Tracking system
- Intelligent Transportation System (ITS)
- Fleet Management System
- Vehicle anti-theft system.

2.3 Proposed system objectives and scope

- Exploring GPS based tracking systems
- Developing Automatic Vehicle Location system using GPS for positioning information and GSM/GPRS for information transmission with following features:
- Acquisition of vehicle's location information (latitude and longitude) after specified time interval.
- Transmission of vehicle's location and other information (including ignition status, door open/close status) to the monitoring station/Tracking server after specified interval of time.
- Server is capable to place the latitude and longitude on Google maps, preparing a simple URL for a map containing the current location of the tracked vehicle
- Server posts the vehicle status to vehicle account on the social network
- Now users who follow vehicle profile will find vehicle feeds attached with a Google map showing vehicle current location.

3. SYSTEM DESIGN

3.1 Proposed System layout

Overall system is partitioned into four major units.

- In-Vehicle unit
- Server
- Social network
- User interact with the system using web browser through vehicle account on the social network, Figure 1 shows system layout

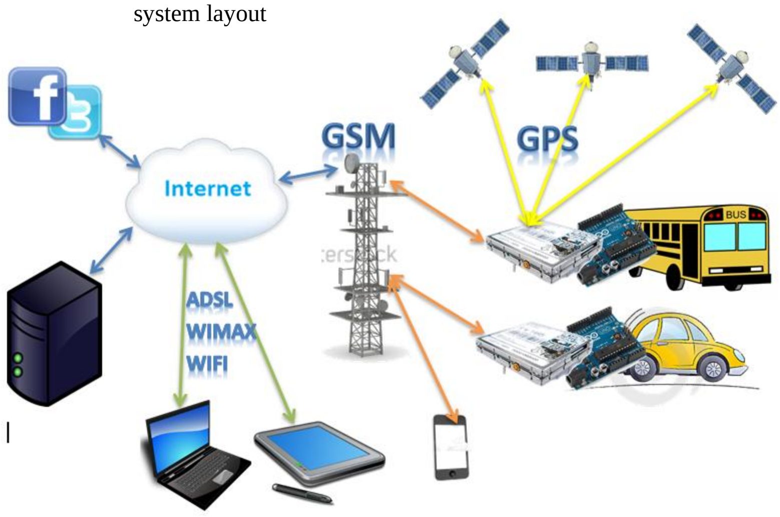


Fig.1 Proposed system layout

3.2 In-Vehicle Unit functions and components

This is major part of the system and it will be installed into the vehicle. It is responsible for capturing the following information for the vehicle.

- Current location of vehicle
- Speed of vehicle
- Door open/close status
- Ignition on/off status

In-vehicle unit is also responsible for transmitting this information to Tracking Server through the internet. To achieve all these functionalities In-Vehicle unit uses following modules.

3.2.1 GPS module

GPS [5] module is responsible of capturing the current location and speed. Location and speed data provided by GSM/GPS module need some fragmentation to be compatible with Google maps format. CPU is required to process this raw data. SiRF Star III single-chip GPS receiver is used which comes integrated with GM862-GPS. GPS receiver can also provide information of altitude, time of last reported location, status of GPS last reported location, number of satellite used to compute current location information along with location and speed. System truncate the vehicle coordinates and time. Other data provided by GPS receiver is used to determine the validity of location information, and will be ignored.

3.2.2 Central Processing Unit

CPU captures raw data from GSM/GPS receiver to extract the required vehicle location and speed information. CPU is also responsible for monitoring vehicle door open/close, engine status on/off and controlling the vehicle ignition on/off status. CPU is also responsible of establishing connections between GSM/GPS module and remote server, through internet over the GSM network, CPU sends detected vehicle location, speed, door status, and engine status. On other hand, CPU process commands being sent from the remote server to control the vehicle like ignition on/off. The microcontroller selected to serve as CPU for In-vehicle unit is Ateml's ATmega328. a popular microcontroller based called Arduino[6] uses the Atmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, and power connector. It contains everything needed to support the microcontroller; it can simply programmed through USB connection, and can be directly interfaced to GSM/GPS module, door, generator, and ignition

3.2.3 Data Transceiver

For real time tracking of vehicle, a reliable wireless network is required to transmit data to remote server. Existing GSM network is selected because of its broad coverage, and its cost effective rather than to deploy own network for transmission of vehicle information.

For data transmission over GSM network, GSM modem is required as a data transceiver module. GM862-GPS [7] GSM/GPRS modem is selected to transmit data over GSM network because of its features and capabilities. GM862-GPS provides AT commands interface i.e. all functions can be accessed using AT commands. AT commands can be sent to it using UART serial interface.

3.3 Design of In-Vehicle Unit

In-Vehicle unit consists of two main modules, the Telite GM862-GPS GSM/GRPS modem, and Arduino microcontroller board. Figure 2, shows the block diagram of in-vehicle unit.

GPS antenna should be directed toward sky in a correct computation of the GPS satellites location to be able to receive GPS satellites signals. Arduino microcontroller communicates with modem through a Simple UART serial interface. Arduino process raw data received from GPS/GSM modem, then transmits this information to remote server using GSM/GPRS modem over the internet through GSM network. Microcontroller controls the operation of GSM/GPRS modem through serial interface using AT commands. To assure a reliable transmission and receiving of data an external GSM antenna is required by the GSM/GPRS modem.

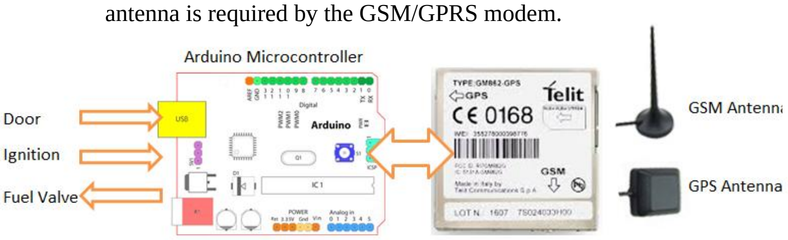


Fig.2 block diagram of in-vehicle unit.

Arduino microcontroller receives commands and information passed from remote server through GPS/GSM modem, then passes this information to which analyses received information and performs action accordingly (i.e. turns on/off ignition of vehicle, transmits current location, etc).

Arduino microcontroller is directly connected to vehicle different components, as a normal digital sensors or actuators, like ignition on/off circuitry and door status output of vehicle. Information packet sent to server also contains status information of these I/O ports.

3.3.1 GM862-GPS Arduino shield card

As mentioned before, Arduino microcontroller communicates with GM862-GPS through UART serial interface [8]. The first step is to design and implement extended PCB with GM862-GPS modem that can be installed over the Arduino microcontroller, called GM862-GPS Arduino shield. Which acts as an upgrade module that can easily interface the GM862-GPS module to Arduino microcontroller. The following figure shows the designed GM862-GPS shield

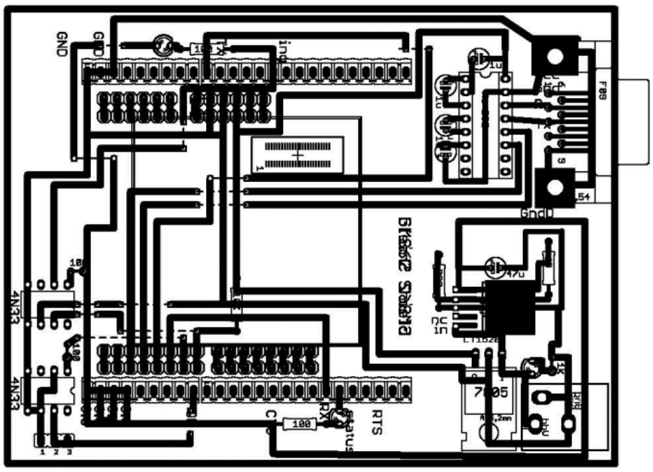


Fig.3 GM862-GPS shield

3.4 In-Vehicle Unit Software Design

Arduino Microcontroller is the Central Processing Unit for In-Vehicle unit, it controls all its operations.

A software program written in C language, that is compiled then burnt into the mictodontroller's flash memory. Complete software consists of small modules as shown in the Figure 4.

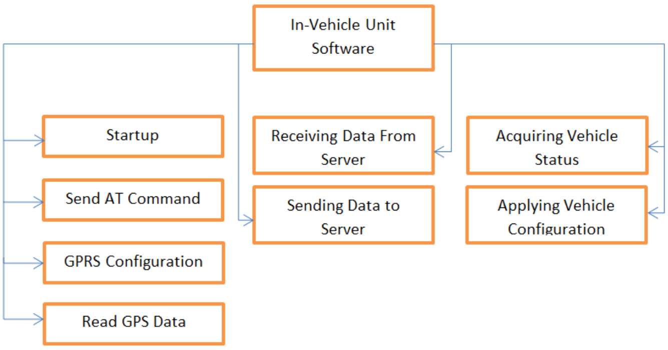


Fig.4 block diagram of in-vehicle unit.

All these modules are implemented as subroutines in the software. Each subroutine performs series of its designated tasks. Flow chart of each subroutine is described below.

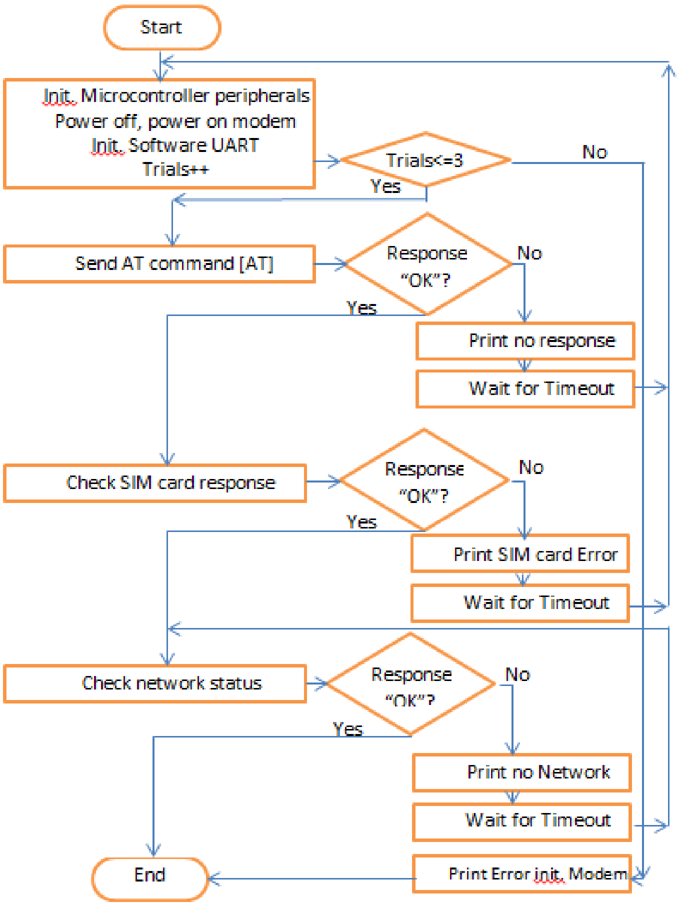


Fig.5 Startup routine flow chart.

3.4.1 Startup- subroutine

Startup routine is executed only when device is powered on and reset. It initializes all hardware of the In-Vehicle unit and configures GM862-GPS. It powers up the GM682-GPS modem then performs various tests to ensure that modem is working properly and is ready to use. If microcontroller failed to communicate with modem, alarm led set on, and sends error description to its serial interface.

Figure 5 shows the startup subroutine flowchart. The flow chart shows that the subroutine starts with initializing peripherals of the microcontroller.

All peripherals in use need to be initialized in this step. After initializations Microcontroller power up the GM862-GPS modem then it starts modem checking process. Microcontroller sends “AT” command to the modem using “Send AT Command” subroutine.

All AT commands sent to the modem are sent using this subroutine. If the device responds with “OK”, it means microcontroller can communicate with module. If device doesn’t respond after expiration of timeout modem is restarted, and the microcontroller sends “no response” message to its serial port (device console port) . If the modem doesn’t respond to microcontroller for three successive trials, and problem then persists definitely something in hardware is damaged. Then microcontroller sends “error initializing modem” message, then go to halt mode.

After receiving “OK” response from the modem, which refers to modem status that is connected and ready. Then various parameters of modem need to be initialized, and checked, SIM presence is checked by sending command “AT+CPIN?” If device responds with “+CPIN: READY” message, SIM is ready to be useed. Microcontroller will consider any other response message as an error. Microlcontroller will send “SIM card error” message to console port, and module will be restarted after expiration of timeout.

When SIM card check is OK,, microtroller is going to make sure that modem is connected to network or not. Network status can be tested with command “AT+CREG?” If module responds with “+CREG: 0, 1” module is connected to network and ready to communicate over the network. If any other response is received module keeps on checking for network status until it is connected. Once it made sure that module is connected to network, subroutine is ended.

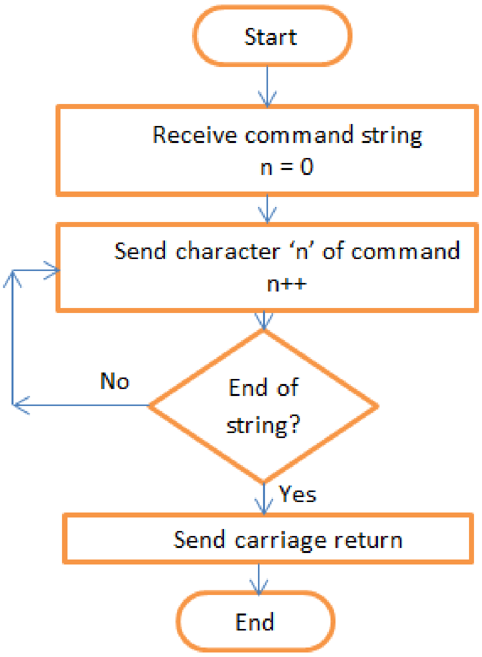


Fig.6 Send AT command routine flow chart.

3.4.2 *Send AT commands - Subroutine*

This subroutine is the basic routine that handles all the communications with GM82-GPS modem. This routine accepts AT command as input string argument or parameter then sends it (character by character) to the modem followed by carriage return (‘\r’) as a command terminating character. Figure 6 shows routine flowchart During the startup routine, a soft serial port (UART) is initialized on two pins which is connected to modem serial port, as shown in figure 3. The communication baud-rate (between microcontroller and modem) is specified during the initialization process too. Transmit buffer is a software register of UART. As soon as a 8-bit data is written into the transmit buffer. It will be transmitted through UART at the specified baud-rate. Each character of command string will be sent in same way. After sending the command characters, microcontroller terminates the command by sending carriage return to the modem. Response received from the modem will be handled in another subroutine.

3.4.3 *GPRS configuration - Subroutine*

The in-vehicle unit sends vehicle information for the server through internet using GPRS service [9]. The first step is to configure modem.

Figure 6 shows the required steps to configure the GM682/GPS module for GPRS data transmission. First step is to define GPRS context, which means identify the internet entry point interface of your network provider. Hence microcontroller sends the following command “AT+CGDCONT” with some parameters to identify network entry point interface in order to gain access to the internet and define the value of IP address of the module as follows;

AT+CGDCONT=1, “IP”, “wap.vodafone.com.eg”, ”0.0.0.0”, 0, 0.

First parameter is context id, it is possible to define up to 5 contexts. Next parameter is communication protocol, third parameter is APN assigned by network server provider.

The next step is to set the parameters for Quality of service. Commands used are ;

“*AT+CGQMIN= 1,0,0,0,0,0*” and

“*AT+CGREQ=1,0,0,3,0,0*”.

These parameters are recommended by manufacturer of the GM862-GPS module.

Network service provider provides a user name and a password to authenticate the network connection, so the next step is to set user name and password for current GPRS context. Commands used are;

“*AT#USERID=WAP”* and

“*AT#PASSW=WAP”.*

Next step is to configure the TCP/IP stack, which mainly sets the minimum packet size, data sending timeout and socket inactivity timeout. Command used for configuring TCP/IP stack is:

“AT#SCFG=1,1,140,30,300,100”.

The first parameter of command is connection identifier; 2nd parameter is the context identifier for which stack is being configured. the 5th parameter (300) is the minimum number of bytes that will be sent in one packet. The last parameter (100) is the inactivity timeout, connection timeout, and data sending timeout.

Next step of the subroutine is to configure the firewall settings, which allows certain computers to connect to the the module. In this case server IP address will be provided to firewall so that Tracking server can connect to In-Vehicle unit. Command used for firewall settings is;

AT#FRWL=1,”server ip”, “subnet mask”

Server IP address is to the Tracking server address on the internet and subnet mask.

Last step is activate current GPRS context through the following command;

"*AT#SGACT=1, 1.*"

First parameter is context id to be activated and next parameter is status i.e. 1 for activation and 0 for deactivation.

3.4.4 *Read GPS data- Subroutine*

Microcontroller requests current location from GM862-GPS by sending the following command "*AT$GPSACP*". Microcontroller waits for modem reply as follows;

"$GPSACP:080220,4542.82691N,01344.26820E,259.07,3,2.1,0.1,0.0,0.0,270705,09"

Microcontroller extract the latitude and longitude from the modem message, which are the 2nd and 3rd parameters "4542.82691N,01344.26820E" in the following format "ddmm.mmmm N/S" for latitude and "dddmm.mmmm E/W" for longitude. dd: degree (0-90 for latitude) and (0-180 for longitude), mm.mmmm: minutes (0- 59.9999 minutes), N/S:

North / South, and E/W: East / West. Microcontroller converts [10] these values first to decimal values instead of degree value in order to send these coordinates to server. Figure 7 shows the subroutine flow chart.

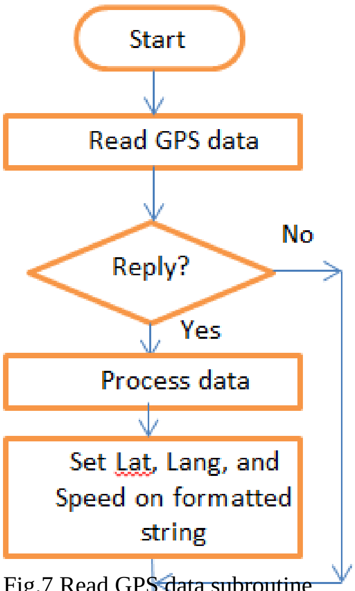


Fig.7 Read GPS data subroutine

3.4.5 *Receiving data from server- Subroutine*

Microcontroller is always listening to GM-862-GPS module data stack, checking for its buffer, if any data is available, it starts processing data saved in data stack buffer. Microcontroller decide what is server asking for then call the appropriate subroutine, like acquiring vehicle status, or acquiring vehicle location, or applying a command to vehicle. Figure 8 shows subroutine flowchart.

3.4.6 *Sending data to server- Subroutine*

In order to send data over the internet (IP network), application needs an interface to medium access and physical layers, which named as socket. The socket is an interface contains three main entries, the transport protocol type (TCP/UDP) the server IP address, and the port number .This subroutine starts with opening socket for currently configured TCP/IP stack [11]. Command used to open socket for configured embedded TCP/IP stack is "*AT#SD=1, 0, 80,"server address"*".

The first parameter is connection identifier of TCP/IP stack, 2nd is protocol i.e. 0 for TCP and 1 for UDP, the 3rd parameter is the port number, and the last parameter is the IP address/host name of Tracking server. If command returns the response CONNECT; connection is accepted. Now the connection to the server is established and ready to send data to the server broadcast and activated on the module. When GPRS connection is alive, module can't accept AT commands and GPS data can't be read from module. To return to At command mode socket is suspended using escape sequence +++.

If In-Vehicle unit is configured for continuous transmission of vehicle information after regular intervals then microcontroller will automatically update server with vehicle status including vehicle location (latitude, and longitude), and vehicle sensors status like door, and ignition status. Figure 9 shows the flowchart for this subroutine.

3.4.7 *Acquiring vehicle status - Subroutine*

In this subroutine the microcontroller collects the vehicle sensors status like the door, and ignition status. The second part of this subroutine is to get the vehicle location and speed, so microcontroller should call the "Read GPS data" subroutine. Finally the information string received from Read GPS data subroutine is appended with status of I/O ports then sent back to tracking server after resuming socket connection. All above steps are repeated otherwise module waits for incoming requests from Tracking server. Figure 10 shows subroutine flow chart.

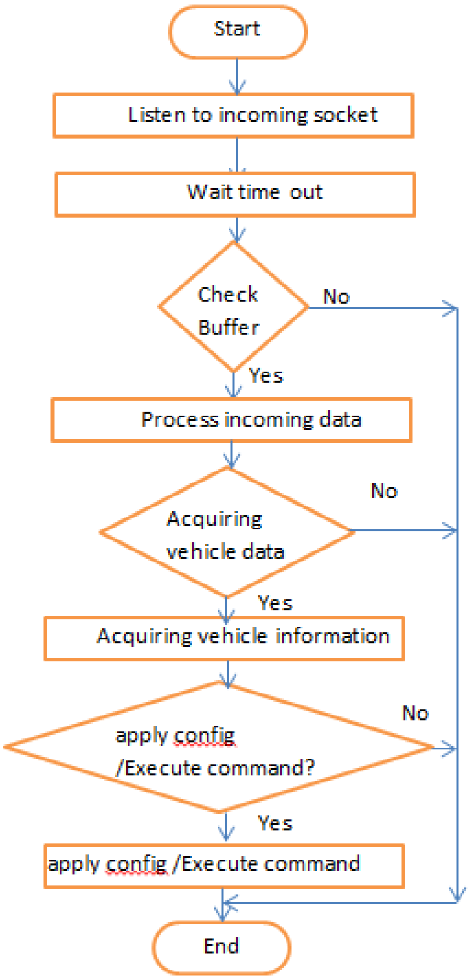

Start
Listen to incoming socket
Wait time out
Check Buffer
No
Yes
Process incoming data
Acquiring vehicle data
No
Yes
Acquiring vehicle information
apply config /Execute command?
No
Yes
apply config /Execute command
End

Fig.8 Receiving data from server- Subroutine flow chart.

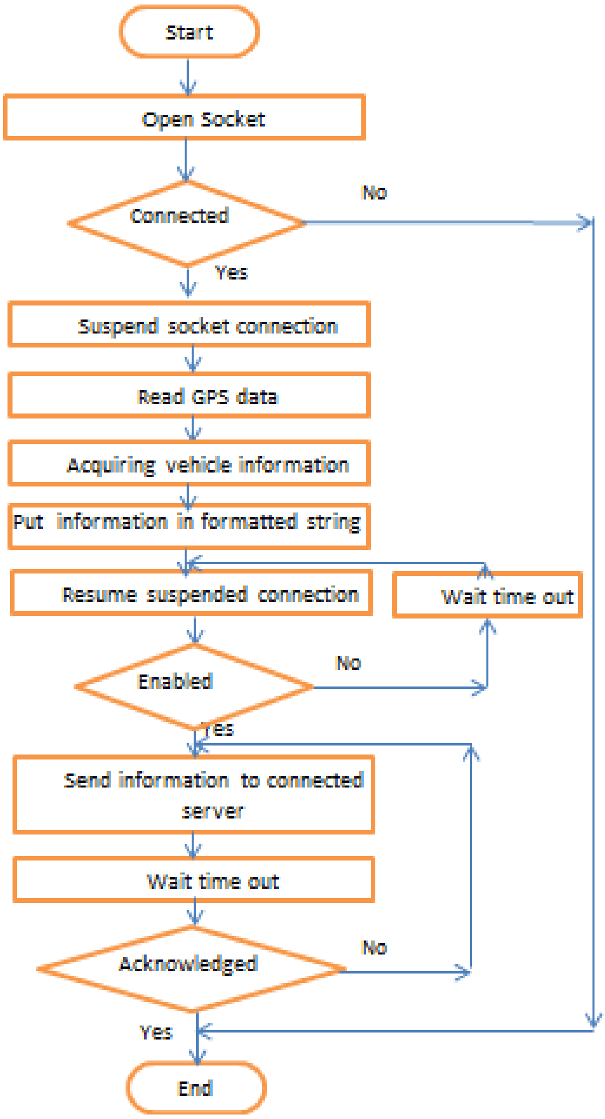

Start
Open Socket
Connected
No
Yes
Suspend socket connection
Read GPS data
Acquiring vehicle information
Put information in formatted string
Resume suspended connection
Wait time out
Enabled
No
Yes
Send information to connected server
Wait time out
Acknowledged
No
Yes
End

Fig. 9 Sending data to server- Subroutine

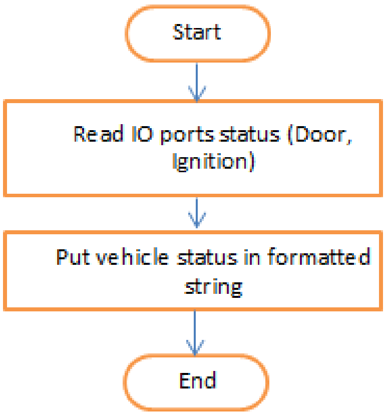

Start
Read IO ports status (Door, Ignition)
Put vehicle status in formatted string
End

Fig.10 Acquiring vehicle status – Subroutine flowchart

3.4.8 *Applying vehicle configuration/ Execute vehicle Command - Subroutine*

Server can send request for vehicle shutdown, restart the In-Vehicle unit. Microcontroller process the received data from server, then accordingly execute the proper action. Figure 11 shows subroutine flow chart.

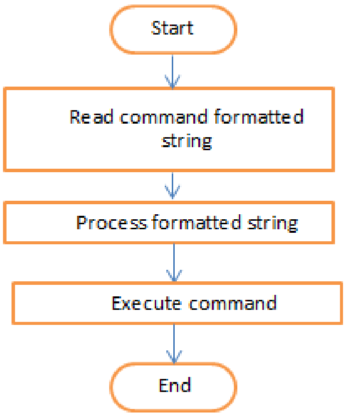


Fig.11 Applying vehicle configuration/ Execute vehicle Command – Subroutine flow chart

3.4.9 *Main Routine of In-Vehicle Unit*

The main routine just calls all above subroutines. When in-vehicle unit starts up, the microcontroller calls startup subroutine once, if the GM862-GPS modem successfully started, microcontroller calls the GPRS configuration subroutine, finally microcontroller enters an endless loop.

In this loop, microcontroller listens to data being received from tracking server by calling "receiving data from server" subroutine, if any data is received microcontroller starts processing requests then take necessary action.

If the in-vehicle unit is configured to continuously update tracking server with vehicle status, microntroller will read GPS data, acquire vehicle status, then finally send these data to tracking server in regular bases, based on unit configuration. All subroutines are implemented in C language. Compiler used Arduino IDE software . Figure 12 shows main subroutine flow chart.

3.5 Tracking Server

3.5.1 *Tracking Server software design*

Tracking server maintains all received information from all In-Vehicle units installed in different vehicles into a central database. This database is accessible from internet through user friendly interface to authorized users through a web application. Here all vehicle updates are available, like vehicle location, door status, ignition status, and authorized user can send commands to in-vehicle unit like shut down the vehicle or restart the in-vehicle unit. Authorized users control vehicle accounts on twitter social network, and can automatically make server posts vehicle updates (vehicle location) to vehicle's account on twitter. Vehicle location is automatically placed on Google maps, which make it easier for tracking the vehicle by vehicle trackers. Tracking Server consists of four major parts.

(i) Communication Software with GM862-GPS
(ii) Communication Software with Twitter social network
(iii) Database
(iv) Web Interface

3.5.2 *Web Interface software module*

As described in previous section Tracking Server maintains all information in a database. To display this information to authorized End users, front end software is required. The Authorized end users are the persons who have installed the In-Vehicle unit in their vehicle and also the system administrators who are managing Vehicle Tracking System. Server is designed to handle many In-Vehicle units at once, each unit presents a car in the tracking server. Each in-Vehicle unit has a unique identifier that identify the vehicle to server and their authorized users. Whenever In-Vehicle unit is installed, information about that vehicle is stored in the database. Web interface supports this functionality. As the tracking server will be available on the internet, access to the vehicle information should be restricted to the authorized users.

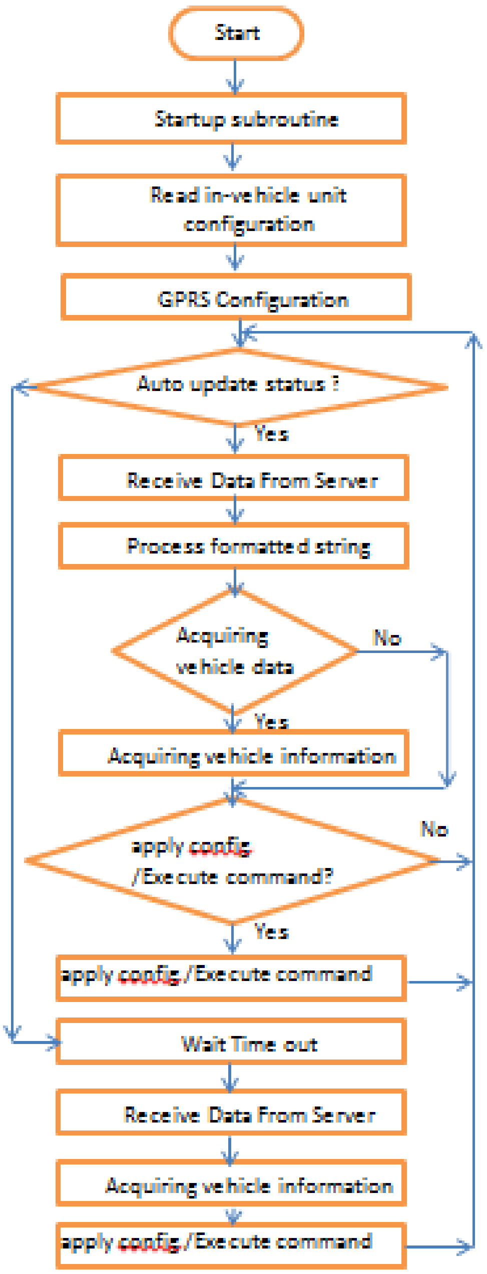


Fig.12 In-Vehicle unit main program

3.5.3 *Database module*

Database is designed to store all received vehicle information (vehicle updates), information about In-Vehicle units and users of the system. Figure 13 show the ER diagram of tracking server database.

3.5.4 Communication Software with GM862-GPSsoftware module

GM862-GPS is GSM/GPRS modem that was used in the In-Vehicle unit. From tracking server point of view it's a seamless TCP/IP communication protocol. Server simply listens to pre-defined socket port, after receiving information form in-vehicle system, it extract the vehicle location and vehicle status, and save it in database. If server is configured to posts vehicle location to vehicle account on twitter, server also forwards a Google map traking the vehicle location on it. Authorized end users who are authorized to access vehicle account on server can monitor vehicle sensors status, and send commands to in-vehicle unit by forwarding these information to in-vehicle unit as a reply to the same socket that in-vehicle unit opened before to communicate with tracking server.

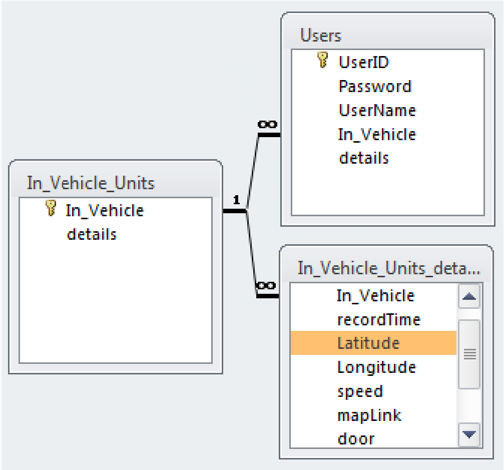


Fig. 13 tracking server DB ER diagram

3.5.5 *Communication Software with twitter social network software module*

As mentioned before tracking server can be configured to automatically post vehicle location on vehicle's twitter account. To do so user needs first to create an application via a vehicle twitter account with Read/Write permission, and of course the twitter's account password is known to tracking server, and saved in its database. To gain access to vehicle twitter's account we used The OAuth 2.0 authorization framework, which enables a third-party application to obtain limited access to an HTTP service. If user is storing protected data on original user's behalf, they shouldn't be spreading their passwords around the web to get access to it. OAuth is used to give users access to data while protecting original users accounts' credentials. Finally authors use PHP Library to support OAuth for Twitter's REST API. Figure 14 shows the flow chart of that software module.

3.6 System Data Flow

Main program listens to incoming TCP/IP connections from in-vehicle units, and creates separate thread for each incoming connection, which allows any number of In-Vehicle units to connect to server at once.

4. System Testing And Results

The proposed system is verified by testing it after integration of all components of the system. The following section contains a few testing information of each module.

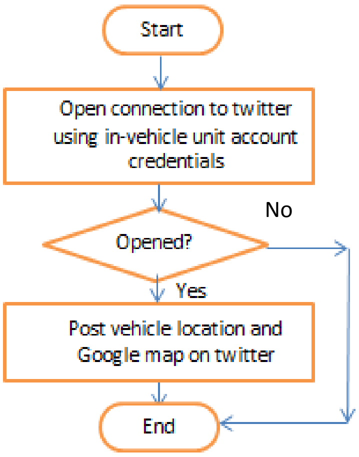

Start
Open connection to twitter using in-vehicle unit account credentials
Opened?
No
Yes
Post vehicle location and Google map on twitter
End

Fig.14 tracking server and twitter subroutine

4.1.1 Testing In-Vehicle Unit

GM862-GPS interface board was connected to microcontroller board through a serial cable. When In-Vehicle unit is powered on, it executes Startup routine. At first it reads and displays the existing configuration of the system. Next step, the microcontroller configures the GM862-GPS. It tests the communication interface by sending "AT" command. GM862-GPS responded with "OK" message which shows that interface is working. +CPIN: READY response shows that SIM card is ready and +CREG:0, 1 response shows that module is connected to network. Figures 15 shows in-vehicle unit.

4.1.2 Testing Tracking Server

In order to test server, an i7 PC was configured to act as a server. Apache server was installed to make it act like server. MySQL database server was installed.

4.1.3 Web Interface Testing

After logging to the website it displays vehicle location placed on Google maps, and the status of vehicle sensors like door, and ignitions status. User can send commands to in-vehicle unit to restart it or to shut down the vehicle. Figure 16 shows web interface.

4.1.4 Twitter social network integration

Each in-vehicle unit has its account on twitter social network. in-vehicle unit update vehicle information on tracking server in frequent basis, and server inturn update vehicle location on twitter's vehicle profile. The following figure 17, shows vehicle account on twitter social network.

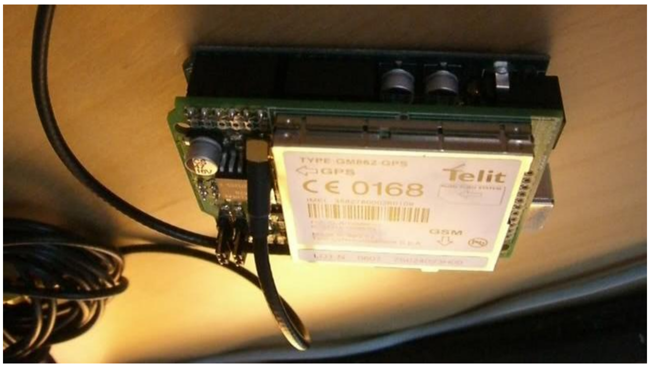


Fig.15 In-Vehicle Unit

5. CONCLUSION

This paper propose a new vehicle tracking and security system, that make use of social network as a value added service for traditional tracking systems. For vehicle tracking in real time, in-vehicle unit and a tracking server is used. The information is transferred to Tracking server using GSM/GPRS module on GSM network by direct TCP/IP connection with Tracking server through GPRS. Vehicle information is recorded in tracking server database. This information like vehicle location (on google maps), and vehicle status (door, and ignition) is only available to authorized users of the system via web interface over the internet. User can send different commands to in-vehicle unit (restart, shut down) to remotely controls his vehicle, which can be used as vehicle security and tracking system. Tracking server posts vehicle location placed on a Google maps to vehicle's twitter account, which make the vehicle followers easily find targeted vehicle, which can be applied to public transportation tracking. Currently In-Vehicle unit was implemented with Arduino. Microcontroller board which is connected to GM862-GPS through extension board named GM862-GPS shield.

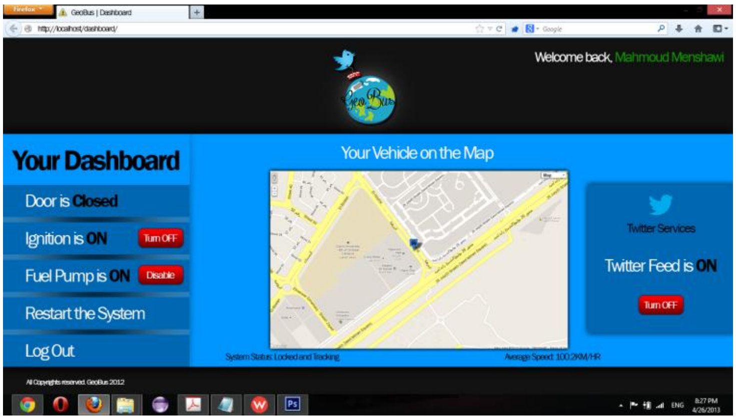
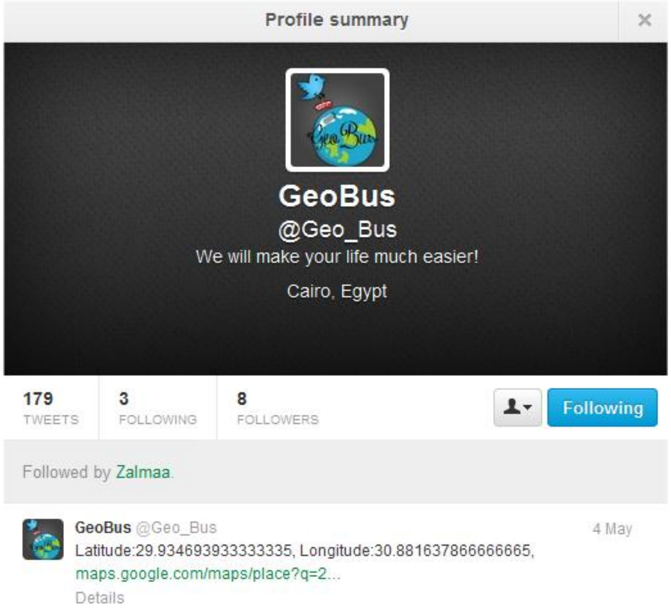


Fig. 16 proposed web interface



GeoBus
@Geo_Bus
We will make your life much easier!
Cairo, Egypt

Followed by Zalmaa.

GeoBus @Geo_Bus 4 May
Latitude:29.934693933333335, Longitude:30.881637866666665,
maps.google.com/maps/place?q=2...
Details

Fig.17 Twitter posts of vehicle location "#GeoBus"

6. ACKNOWLEDGMENTS

This project won the 1st prize of ACU “Make it Tweet” competition Fall 2011 which was organized by Ahram Canadian university. Then won the 1st prize of “Microsoft Geeks” competition 2012, organized by Microsoft Egypt, and hosted in Ahram Canadian University, Spring 2012.

The following video stream show development and operating process of prototype.

http://youtu.be/mVTgX3bpx9Y
http://youtu.be/cxjfSAub1iI

7. REFERENCES

[1] Muruganandham , P.R.Mukes, “Real Time Web based Vehicle Tracking using GPS”, World Academy of Science, Engineering and Technology 37 2010

[2] boyd, d. m., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), article 1.

[3] "Transportation District's Automatic Vehicle Location System" [online:] http://www.itsdocs.fhwa.dot.gov/ JPODOCS/REPTS_TE/13589.html

[4] "Vehicle Tracking Systems Overview" [Online:] http://www.roseindia.net/technology/vehicletracking/ VehicleTrackingSystems.shtml

[5] J.B. TSUI, "Fundamentals of Global Positioning System Receivers", 1st Edition. John Willey & Sons Inc., 2000.

[6] "Arduino microcontroller", [online:] http://Arduino.cc

[7] "Telit Wireless Solutions", GM862-GPS Modem, 2008.

[8] "Telit Wireless Solutions" "GM862-GPS Hardware user guide". 1vv0300728 Rev. 8 - 20/09/07

[9] "GPRS (General Packet Radio Service), HSCSD & EDGE" [online:] http:// www.mobile-phones-uk.org.uk/ gprs.htm

[10] R. Parsad, M. Ruggieri, "Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems", London, ARTECH HOUSE, 2005.

[11] T. Halonen et al, "GSM, GPRS and EDGE Performance", 2nd Edition, Chichester, John Willey & Sons Ltd., 2003.

Ahmed M. ElShafee, Held a Bachelor degree in Electrical Engineering from Faculty of Engineering, Alexandria University, Masters' of science in Electrical Engineering from Faculty of Engineering, Alexandria University, Ph.D. degree in Electrical Engineering from Faculty of Engineering, Alexandria University.
He published many scientific papers, international conferences in Egypt, France, Dubai, Namibia, and India.
He won The Best Young Scientist Award as per the conference council recommendation (National Radio Science Conference 2001), Alexandria, Egypt, for his paper entitled “Rotor Enhanced Block Cipher (REBC)”.
He worked in telecommunication engineering field (Operations and Research & Development) for more than 8 years. Now he works as Assistant Professor, in Faculty of Computer Science and Information Technology (Ahram Canadian University), and as Researcher in Information Technology Research & Consultation Center (ITRCC), Ahram Canadian University.
Mahmoud ElMenshawy, Hold B.Sc. in Computer Science from Faculty of computer science and information technology, Ahram Canadian University. He works as Technical Support Professional at Stream Global Services since 2011.
Mena Saeed, Hold B.Sc. in Computer Science from Faculty of computer science and information technology, Ahram Canadian University