Real Time Web based Vehicle Tracking using GPS

Muruganandham , P.R.Mukesh

***Abstract*—**Tracking systems were first developed for the shipping industry because they wanted to determine where each vehicle was at any given time. Passive systems were developed in the beginning to fulfill these requirements. For the applications which require real time location information of the vehicle, these systems can't be employed because they save the location information in the internal storage and location information can only be accessed when vehicle is available. To achieve Automatic Vehicle Location system that can transmit the location information in real time, Active systems are developed. Real time vehicular tracking system incorporates a hardware device installed in the vehicle (In-Vehicle Unit) and a remote Tracking server. The information is transmitted to Tracking server using GSM/GPRS modem on GSM network by using SMS or using direct TCP/IP connection with Tracking server through GPRS. Tracking server also has GSM/GPRS modem that receives vehicle location information via GSM network and stores this information in database. This information is available to authorized users of the system via website over the internet

***Keywords*—**GSM, GPS, GPRS, GM862

I. INTRODUCTION

THE roots of Vehicle Tracking Systems lie in shipping industry. When large fleet of vehicles were spread out over the vast expanses of ocean, the owner corporations often found it difficult to keep track of what was happening. They required some sort of system to determine where each vehicle was at any given time and for how long it travelled. The need of vehicle tracking in consumer's vehicle rose to prevent any kind of theft because Police can use tracking reports to locate stolen vehicle. [11] Initially vehicle tracking systems developed for fleet management were passive tracking system. In passive tracking system a hardware device installed in the vehicle store GPS location, speed, heading and a trigger event such as key on/off, door open/closed. When vehicle returns to a specific location device is removed and data downloaded to computer. Passive systems also included auto download type that transfer data via wireless download but the system was not real time. [10, 11]

Passive systems weren't useful to track consumer's vehicle for theft prevention. Real time tracking system was required that can transmit the collected information about the vehicle after regular intervals or at least could transmit the information when required by monitoring station. Active systems were developed that transmit vehicle's data in real time via cellular or satellite networks to a remote computer or data centre. [10, 11]

Many vehicle systems that are in use now days are some form of Automatic Vehicle Location (AVL). It is a concept for determining the geographic location of a vehicle and transmitting this information to a remotely located server. The location is determined using GPS and transmission mechanism could be a satellite, terrestrial radio or cellular connection from the vehicle to a radio receiver, satellite or nearby cell tower. Other options for determining actual location, for example in environments where GPS illumination is poor, are dead reckoning, i.e. inertial navigation or active RFID systems or cooperative RTLS systems. After capture, the tracking data can be transmitted using any choice of telemetry or wireless communications systems. GSM is the most common used service for this purpose. [10, 11]

A. Objective

- 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 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.
- Developing a web based software to display all transmitted information to end user along with displaying location of vehicle on a map.

II. SYSTEM DESIGN AND IMPLEMENTATION

Overall system is partitioned into two major design units.

- In-Vehicle unit
- Tracking Server/Monitoring Station.

A. In-Vehicle Unit

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

Muruganandham, Asst.Prof., Department of ECE,PG Research lab, Sona College of Technology, Salem, Tamil Nadu, India (+91-9443825711, e-mail : muruga_salem@rediffmail.com)

P. R. Mukesh., lecturer, Department of Aeronautics, Research Lab, Bannari Amman Institute of Technology, Sathy, Erode, Tamil Nadu, India (+ 91-9486872922,e-mail:pr.mukeshphd@gmail.com)

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

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

B. GPS Receiver

In-Vehicle unit uses GPS receiver to capture the current location and vehicle speed. Location and speed data provided by GPS is not in human understandable format. This raw data needs to be processed to convert it into useful information that can be displayed by a beacon on the map. CPU is required to process this raw data. SiRF Star III single-chip GPS receiver is used which comes integrated with GM862-GPS which is GSM/GPRS modem used for data transmission. GPS receiver can also provide information of altitude, time of GPS fix, status of GPS fix, and number of satellite used to compute current location information along with location and speed. GPS fix means last reported location. For tracking purpose only location and speed data is required for transmission. Other data provided by GPS receiver is used to determine the validity of location information.

C. Central Processing Unit

The raw data provided by the GPS receiver is captured by the CPU and processed to extract the required location and speed information CPU is also responsible for monitoring the door/open close status of vehicle and controlling the ignition on/off status of the vehicle. CPU holds all the required information that is to be transmitted to remote server. It also controls data transmission module to exchange information with remote server. It actually acts as a bridge between GPS receiver, vehicle and remote server. It receives commands sent by server through data transmission/receiving module and performs corresponding action required by server. As the processing required in the In-vehicle unit is not computationally intensive therefore any low end microcontroller can be used as a CPU. The microcontroller selected to serve as CPU for In-vehicle unit is Microchip's PIC18F248. This is 8-bit microcontroller and runs at speed of 20 MHz which is enough speed for the system.

D. Data Transceiver

When all required information is extracted and processed, it needs to be transmitted to a remote Tracking Server which will be able to display this information to the end user. For real time tracking of vehicle, reliable data transmission to remote server is very important. Wireless network is required to transmit vehicle information to remote server. Existing GSM network is selected to transmit vehicle information to remote server because of broad coverage of GSM network. It is also cost effective rather than to deploy own network for transmission of vehicle information. For data transmission over GSM network GSM modem is required. GSM modem can send and receive data SMS text messages and GPRS data over GSM network. GM862-GPS 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 by use of AT commands. AT commands can be sent to it using serial interface. It has built in UART that accepts the AT commands and modem performs the function as required by AT command received.

III. DESIGN OF IN-VEHICLE UNIT

In-Vehicle unit is designed using OEM module Telit GM862-GPS GSM/GPRS modem and microcontroller PIC18F248 manufactured by Microchip. Figure 1 shows the block diagram of In-Vehicle unit.

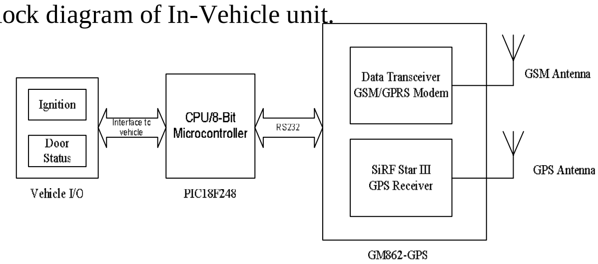

Fig. 1 In-Vehicle Unit Block diagram

GPS antenna receives signals from GPS satellites and it must face towards sky for correct computation of the current location by GPS receiver. Location data is transferred to microcontroller through serial interface. After processing of the data provided by GPS receiver, microcontroller transmits this information to remote location using GSM/GPRS modem. Microcontroller controls the operation of GSM/GPRS modem through serial interface using AT commands. External GSM antenna is required by the GSM/GPRS modem for reliable transmission and receiving of data. When modem receives any command sent by tracking server, it passes this information to microcontroller which analyses received information and performs action accordingly (i.e. turns on/off ignition of vehicle, transmits current location, restarts GPS receiver, restarts whole system etc). Some of microcontroller I/O ports are connected to vehicle ignition on/off circuitry and door status output of vehicle. Information packet sent to server also contains status information of these I/O ports.

A. GM862-GPS Interface Board Design

First step in circuit design of In-Vehicle unit is to design interfacing circuit for Telit GM862-GPS so that it can be interfaced with microcontroller. Telit GM862-GPS is provided of the following interfaces:

- GSM Antenna Connector
- Board to Board Interface Connector
- SIM Card Reader
- GPS Antenna Connector

GSM, GPS antennas and SIM card are not important from design point of view as they can be just installed into connectors. Only important is board to board interface connector which provides interface for external devices to the modem.

IV. IN-VEHICLE UNIT SOFTWARE DESIGN

Microcontroller is acting as Central Processing Unit for In-Vehicle unit. All operations of the In-Vehicle unit are to be controlled by the microcontroller. Microcontroller needs instructions to operate the whole system. These instructions are provided to microcontroller by writing the software into microcontroller's flash memory. It reads the software instruction by instruction and performs the action as required by instruction. Complete software is broken down into small modules as shown by the Figure 2.

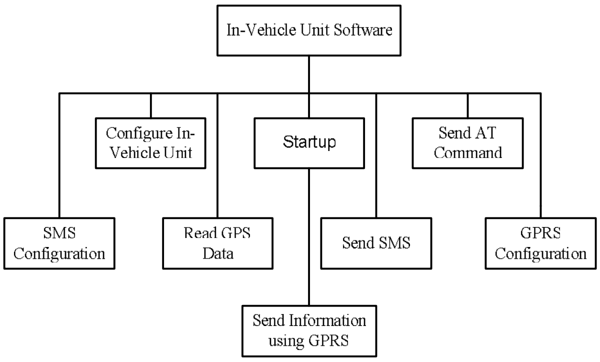

Fig. 2 Breakdown of In-Vehicle software

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.

A. Subroutine- Send AT Command

This subroutine is the basic routine which handles all the communication with GM82-GPS.

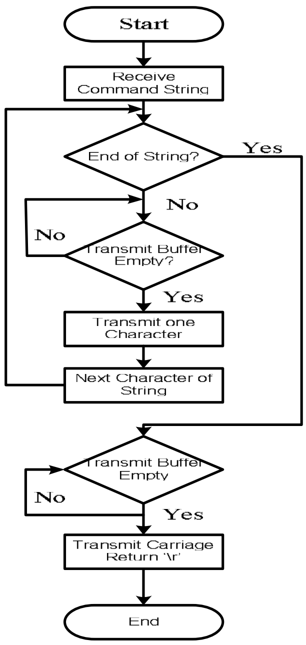

Fig. 3 Flow chart of subroutine Send AT commands

This routine accepts the string containing "AT" command input in its parameters and sends this string character by character to module. GM862-GPS accepts carriage return ('\r') as a command terminating character. As this character is received it sends back the response to microcontroller. Figure 3 shows the flowchart

As shown in the flow chart routine checks each character of string, if the character is not null, it will check the transmit buffer contents. If transmit buffer is empty it will write new character into the buffer. Transmit buffer is a hardware register of UART. As soon as a 8-bit data is written into the transmit buffer, UART hardware transmits that character at the specified baud rate. Each character of command string will be sent in this way. When null character is found, it specifies end of string and routine terminates by sending carriage return to the module. Response received from the module will be handled in another subroutine.

B. Subroutine- Startup

Startup routine is executed only when device is powered on. It initializes all hardware of the In-Vehicle unit and configures GM862-GPS. It performs various tests to ensure the GM862-GPS is working properly and is ready to use. Figure 4 shows the flowchart.

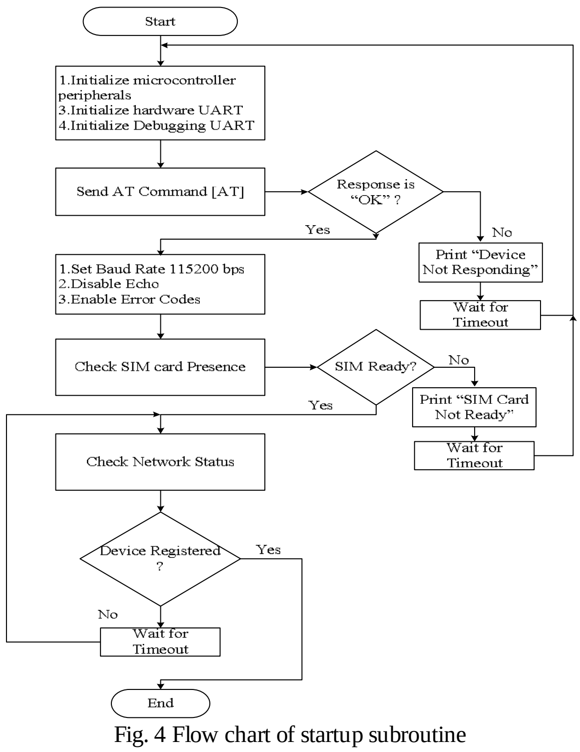

Fig. 4 Flow chart of startup subroutine

As shown in the flowchart subroutine starts with initializing peripherals of the microcontroller. All peripherals in use need to be initialized in this step. After initializations of local peripherals, GM862-GPS needs to be tested. Microcontroller sends "AT" command to GSM module using subroutine Send AT Command.

All commands sent to module 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 routine is restarted. If problem persists definitely something in hardware is damaged. After receiving "OK" response from module various parameters of module need to be initialized. SIM presence is checked by sending command "AT+CPIN?" If device responds with "+CPIN: READY" message, SIM is ready to use. Any other response message will be considered as an error and routine will be restarted after expiration of timeout.

When SIM card is ready, it is important to test whether module 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 data can be sent over network. If any other response is received module keeps on checking for network status until it connects to network. Once it makes sure that module is connected to network, subroutine is terminated.

C. Subroutine- Read GPS Data

GPS controller is by default powered on when module is switched on. Figure 5 shows the flow chart for Read GPS Data subroutine. As shown in the flow chart subroutine first of all checks whether GPS controller is powered on? To check this "AT$GPSP?" is sent to the module. If it responds with $GPSP: 0 it is not powered up. If it is not already powered up; it can be switched on by sending "AT$GPSP=1". Once GPS controller is powered up location information can be read from it by sending "AT$GPSACP".
The module responds with a long NMEA sentence. The information of interest is latitude, longitude, speed, number of satellites used in calculating latitude and longitude. This information is extracted from the received response and saved in formatted string.

This string can be later on passed to Send SMS subroutine to send it to remotely located Tracking Server.

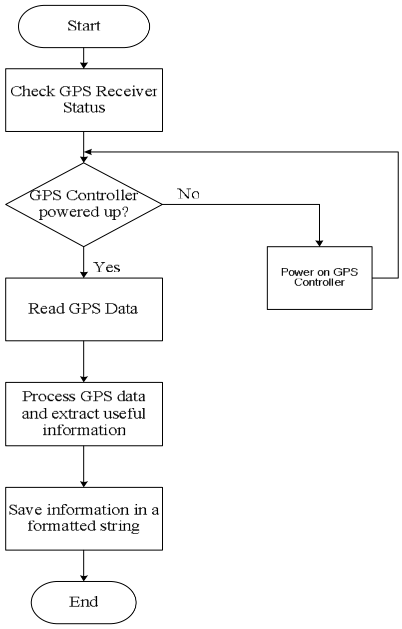

Fig. 5 Flow chart of subroutine Read GPS Data

D. Subroutine- Send SMS

This subroutine accepts message string as input parameter which needs to be transmitted. Subroutine adds a terminating character Ctrl-Z at the end of message string as shown in the Figure 6

Then it checks whether module is in Text SMS mode. It can be checked by sending command "AT+CMGF?" If module responds with "+CMGF: 0" it is in PDU mode. Mode can be changed to text by sending command "AT+CMGF=1". To send an SMS module requires destination phone number that is sent to module using command "AT+CMGS= da" where da represents the destination phone number. This phone number will be read from microcontroller internal memory which is stored during programming. After sending destination number module waits for prompt ">".

When prompt appears message string is sent using Send AT Command subroutine. If message sent successfully, module responds with +CMGS: <mr> where mr is message reference number. If any error occurs subroutine tries to resend the message until it is successfully sent.

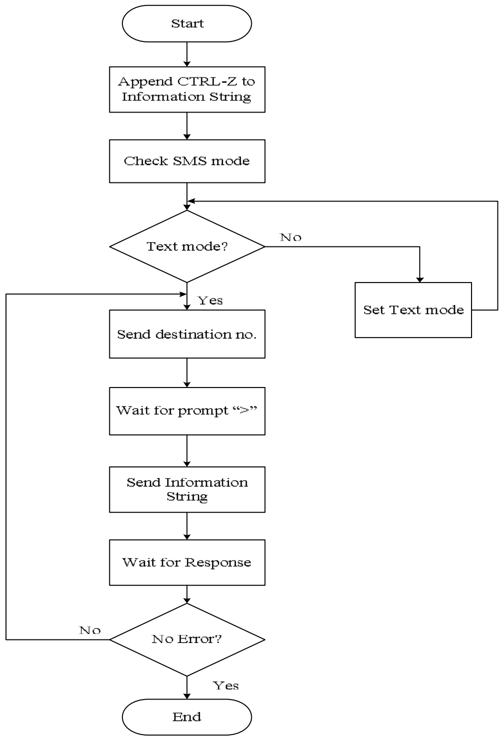

Fig. 6 Flow chart of subroutine Send SMS

E. Subroutine- SMS configuration

SMS configuration subroutine is call after startup routine. It is basically called once after powering up the In-Vehicle unit like startup routine.

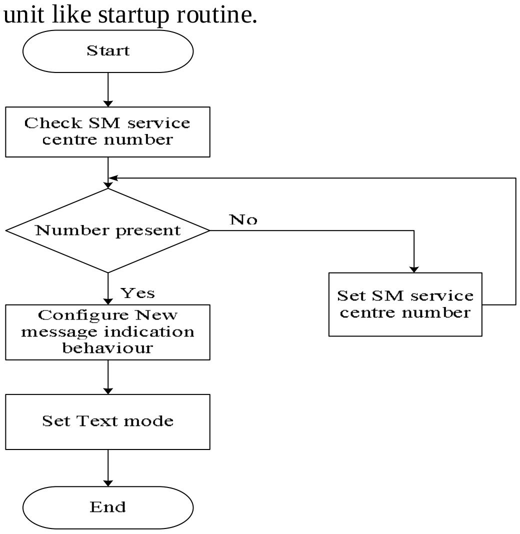

Fig. 7 Flow chart of subroutine- SMS Configuration

It can be part of startup routine but it is separated because it does configuration of the module related to SMS only. Figure 7 shows the flow chart.This subroutine checks the SMS service centre number by sending the command "AT+CSCA?" Service centre number is required because SMS is routed to destination via SMS service centre.The module responds with "+CSCA: number". If no number is present it can be saved in module by sending the command "AT+CSCA= number, type" type could be 145 if number is in international number format (i.e. it begins with +) or it could be 129 if number is in national format. When new message is received by module an unsolicited indication is generated. This indication may be sent to microcontroller, buffered if microcontroller is busy or discarded. In this case new message must be immediately sent to microcontroller or buffered if microcontroller is busy. This configuration can be done by sending command "AT+CNMI=1, 1, 0, 0, 0" when GSM modem receives a new message it will send "+CMTI: "SM", message index no" where message index no is location of message in memory and it can be then read by sending command "AT+CMGR=message index no". After configuring new message behavior module is set to Text mode for SMS. It can be done by sending command "AT+CMGF=1". All configuration related to SMS is finished and subroutine terminates.

F. Subroutine- Configure GPRS

When GPRS service is available, it is cost effective and more efficient to transmit vehicle information through GPRS. In order to connect to GPRS, it needs to be configured. Figure 8 shows the steps required to configure the GMS module for GPRS data transmission. First step in configuration of GPRS is to define GPRS context. It is set of information to identify the internet entry point interface provided by the ISP. With these parameters the GPRS network identifies the ISP to be used to gain access to the internet and defines the value of IP address of the GPRS device once connected.

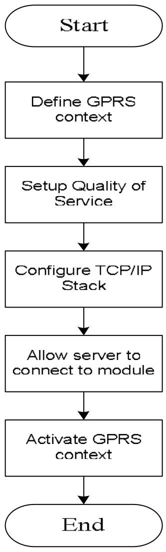

Fig. 8 Flow chart of subroutine configure GPRS

The command sent for defining GPRS context is AT+CGDCONT=1, "IP", "payandgo.o2.co.uk", "0.0.0.0", 0, 0. First parameters is context id, it is possible to define up to 5 contexts. Next parameter is protocol used for communication, third parameter is APN assigned by network server provider. In next step subroutine sets 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 GSM module. Along with APN network service provider also provides user name and password to connect to ISP. Next step is to set user name and password for current GPRS context. Commands used are AT#USERID=payandgo and AT#PASSW=password. Next step configures the TCP/IP stack. It basically 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. First parameter of command is connection identifier; next parameter is context identifier for which stack is being configured. 300 is the minimum number of bytes that will be sent in one packet. Next parameters are inactivity timeout, connection timeout, and data sending timeout. Next step of the subroutine is configures the firewall settings. It allows certain computers to connect to 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 will be the IP address of Tracking server and subnet mask can be provided to allow access to range of computers. Last step is activate current GPRS context. Command is 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.

G.Subroutine-Send Information Using GPRS

When In-Vehicle unit is configured to send information using GPRS, all activities of In-Vehicle unit are controlled by this subroutine.

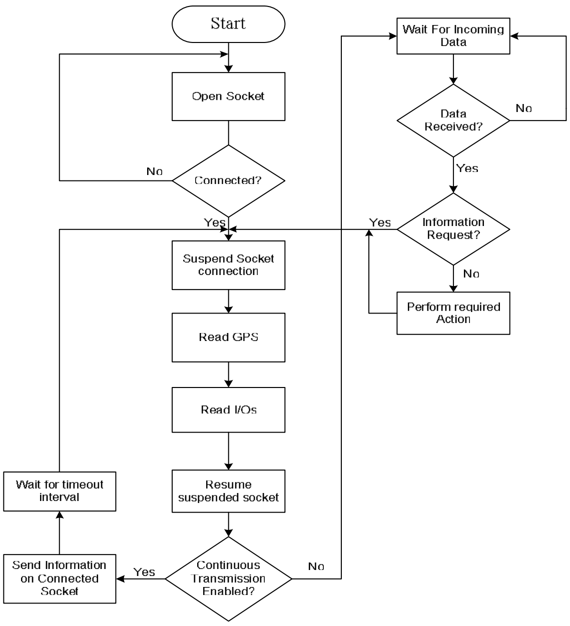

Fig. 9 Flow chart of subroutine Send Information using GPRS

Figure 9 shows the flowchart for this subroutine. In order to send data over IP network application needs an interface to physical layer. This interface is named as socket. This subroutine starts with opening socket for currently configured TCP/IP stack. Command used to open socket for configured embedded TCP/IP stack is AT#SD=1, 1, 6534. First parameter is connection identifier of TCP/IP stack, 2^{nd} is protocol i.e. 0 for TCP and 1 for UDP. Next two parameters are port number and IP address/host name of Tracking server respectively. If command returns the response CONNECT; connection is accepted. Data can be sent now. After getting connection, socket is suspended using escape sequence +++ to bring module in command mode. Socket remains connected while it is suspended. When GPRS connection is alive, module can't accept AT commands and GPS data can't be read from module. Once module is in command mode this subroutine calls the routine Read GPS data which provides the information string that is to be sent to Tracking Server. Next step is to read I/O ports of microcontroller to get vehicle's door and ignition status. Information string received from Read GPS data subroutine is appended with status of I/O ports. Socket connection is resumed and information is sent to Tracking server on this socket. If In-Vehicle unit is configured for continuous transmission of vehicle information after regular intervals, all above steps are repeated otherwise module waits for incoming requests from Tracking server. If location request is received above steps are repeated and if any other command is sent by the server according action is taken. Server can send request for vehicle shutdown, changing the data transmission from GPRS to SMS or changing the continuous transmission to polling or vice versa, restart the In-Vehicle unit. This subroutine ends only when In-Vehicle unit is restarted by Tracking server.

H. Main Routine of In-Vehicle Unit

Main routine just calls the subroutines described in previous sections. With start of main routine call is made to Startup routine that initializes all peripheral and In-Vehicle unit configurations. It checks for stored configuration to decide whether data transmission should be through GPRS or SMS. If configuration says for GPRS, call is made to GPRS configuration routine and then GPRS data sending routine is run. If configuration is for SMS, configuration is done and In-Vehicle unit starts sending the vehicle information to Tracking server via SMS either continuously after regular intervals or it waits for commands from Tracking server as SMS. GM862-GPS is configured in such way that whenever new SMS arrives, and indication is received by microcontroller with message identifier. This message is read by microcontroller and corresponding action is performed as shown in Figure 10.

All subroutines are implemented in C language. Compiler used to generate machine language code for PIC18F248 is CCS PICC.

V. TRACKING SERVER

Tracking server maintains all information received from all In-Vehicle units installed in different vehicles into a central database. This database is accessible from internet to authorized users through a web interface. Authorized users can track their vehicle and view all previous information stored in database.

Tracking server has a GSM/GPRS modem attached to it that receives SMS from In-Vehicle units and sends those messages to the server through serial port. Tracking server saves this information into database.

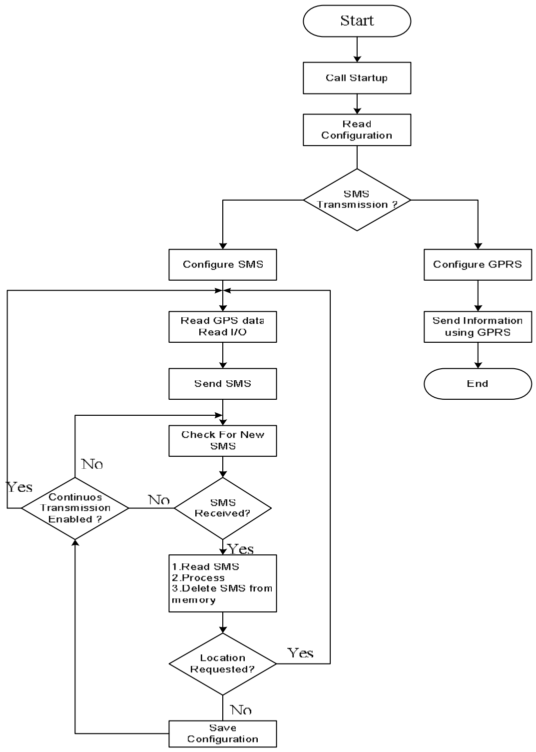

Fig. 1 Flow chart of Main program

Design of Tracking Server is partitioned into four major parts.

(i) Hardware design for GSM/GPRS Modem (GM862-GPS)
(ii) Communication Software for GM862-GPS
(iii) Database
(iv) Web Interface

A. Web Interface Design

As described in previous section Tracking Server maintains all information in a database. To display this information to end users front end software is required that can display all information to the end user. End user is the user of system who has installed the In-Vehicle unit in his vehicle and also the administrator of the system who is managing Vehicle Tracking System. There may be a number of vehicles installed with In-Vehicle units therefore server must be able to manage and distinguish information sent by all In-Vehicle units. For this purpose information must be available to server about all vehicles that are installed with In-Vehicle units. Whenever In-Vehicle unit is installed, information about that vehicle is stored in the database. Web interface must also support this functionality. Since web interface will be accessible over the internet therefore access must be restricted to authorized users only. Therefore information about all users of the system must be stored in database.

B. Database Design

Database is designed to store all received vehicle information, information about In-Vehicle units and users of the system. Information to be stored in the database is

- Information about users of the system
- Information about vehicles
- Information about received from vehicles

C. GM862-GPS Interface Board for Tracking Server

GM862-GPS is GSM/GPRS modem that was used in In-Vehicle unit. The same modem is used on server side to exchange information with In-Vehicle units through SMS. Vehicle information sent using SMS on GSM network is received by this modem. Tracking server can also send commands for In-Vehicle units using this modem. Same interface board is used on this side. GM862-GPS interface board is connected to the serial (COM) port of server. Server can communicate with modem using AT commands. To send and receive data using this modem a software is required that can send AT commands to module.

D. Design of Communication Software for GM862-GPS

The software that is to be designed will provide communication interface to the GM862-GPS modem attached to computer’s serial port. It will control the operations of GM862-GPS.

This software must be able to support following functions

- Configuration of GM862-GPS for sending and receiving SMS
- Receiving the SMS.
- Processing received SMS and saving information into database
- Sending SMS to in vehicle unit as required by user
- Accepting TCP/IP connections from In-Vehicle units
- Exchanging information with In-Vehicle units through internet

GM862-GPS will be configured in such a way that whenever new SMS arrives, GM862-GPS will send the information about SMS to the serial port. Software will be listening at serial port; it will read the SMS from GM862-GPS memory and extract the information from SMS. After extracting the information SMS will be deleted from GM862-GPS by software and information will be written to the database.

Design requirements suggest that following objects are part of the system.

- GM862-GPS Modem
- Serial Port
- Vehicle Info
- TCP/IP Socket
- Database

This analysis yields following classes in the system.

E. Data Flow

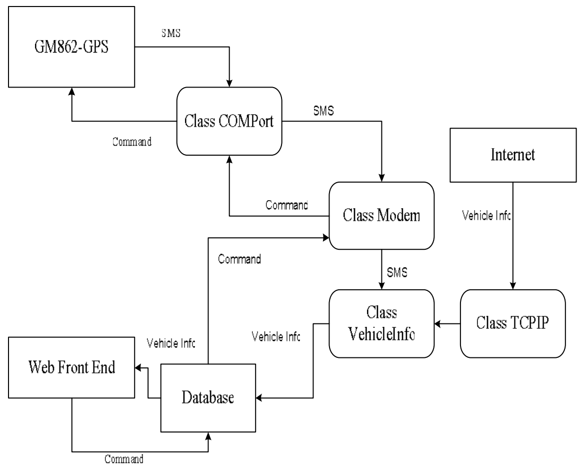

Fig. 11 Data flow of communication software

F. Software Flow

Figure 12 shows the flow chart of main program. Main program listens for SMS and handles all communication with In-Vehicle units using SMS. It creates a separate thread for listening to TCP/IP connections, which receives incoming 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.

VI. System Testing And Results

System design needs to be verified by testing after integration of all components of the system. PCB designed for In-Vehicle unit and server side was assembled. After integrating all the components, system was tested.

A. Testing In-Vehicle Unit (SMS Configuration)

GM862-GPS interface board was connected to microcontroller board through a serial cable.

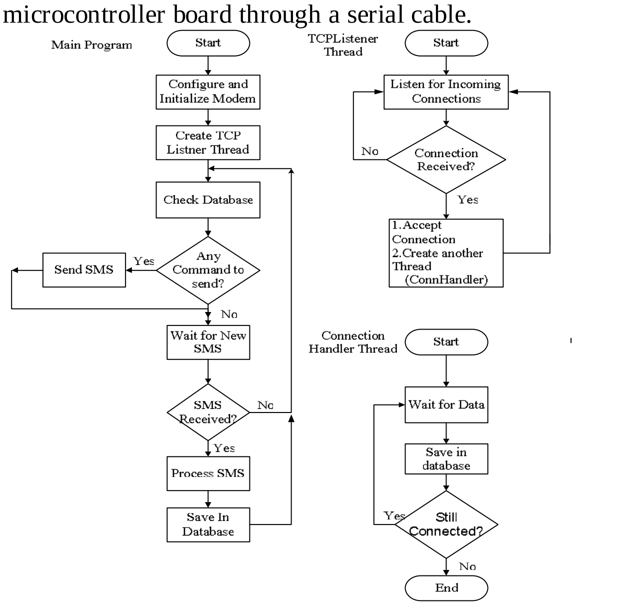

Fig. 12 Flowchart of communication software for GM862-GPS

Debugging serial port of In-Vehicle unit was connected to a laptop's COM port to see the debugging messages printed by microcontroller on HyperTerminal during its operation. This laptop and debugging COM port is just for debugging purposes, in real time there is no need to connect laptop to In-Vehicle unit.

After connecting the GSM antenna and GPS antenna to the In-Vehicle unit system was powered on. Following logs of microcontroller operation were captured from HyperTerminal.

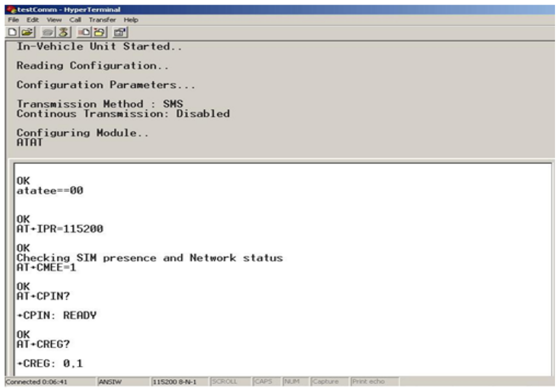

Fig. 13 Results of execution of Startup routine

When In-Vehicle unit is powered on it executes Startup routine. It first reads and displays the existing configuration of the system. In next step microcontroller is configuring the GM862-GPS. It first 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.

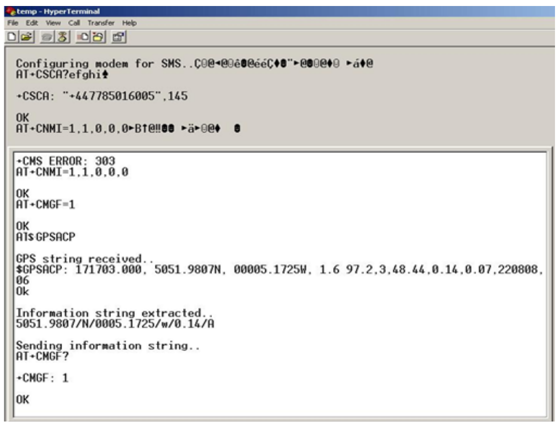

Fig. 14 Results of execution of SMS Configure routine

B. Testing Tracking Server

In order to test server, laptop was configured to act as a server. GM862-GPS COM was connected to COM port of laptop. Apache server was run on laptop to make it act like server. MySQL DBMS was installed. After running the

Communication software for GM862-GPS following results were observed.

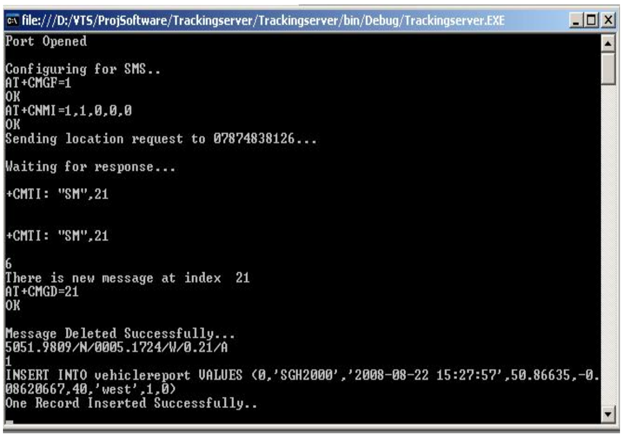

Fig. 15 Logs of Tracking Server

C. Web Interface Testing

Since server is setup on the local machine. Website was opened in internet explorer. After logging to the website it displayed the page as shown in Figure.

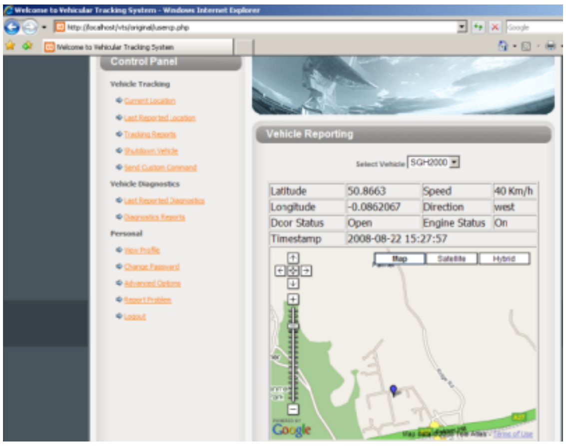

Fig. 16 Pointing out current location of vehicle

VII. CONCLUSION

The results presented in this paper contain execution of Startup routine, execution of SMS Configure routine, Logs of Tracking Server and Pointing out current location of vehicle. For vehicle tracking in real time, in-vehicle unit and a tracking server is used. The information is transmitted to Tracking server using GSM/GPRS modem on GSM network by using SMS or using direct TCP/IP connection with Tracking server through GPRS. Tracking server also has GSM/GPRS modem that receives vehicle location information via GSM network and stores this information in database. This information is available to authorized users of the system via website over the internet. Currently In-Vehicle unit was implemented with two boards. Microcontroller board was externally connected to GM862-GPS interface board. Single board can be designed to incorporate Microcontroller circuitry on the GM862-GPS interface board. It will reduce the overall size of In-Vehicle unit and it will also reduce the number of components so will the cost.

REFERENCES

[1]. G. T. French (1996) Understanding the GPS. 1st Edition. Bethesda, GeoResearch Inc.

[2]. J.B. TSUI (2000) Fundamentals of Global Positioning System Receivers. 1st Edition. John Willey & Sons Inc.

[3]. GPSImages[online:] http://www.gpsvehiclenavigation.com/GPS/images.php

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

[5]. R. Steel et al (2001) GSM, cdmaOne and 3G Systems. Chichester, John Willey & Sons Inc.

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

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

[8]. Telit Wireless Solutions (2008) GM862-GPS Modem

[9]. Microchip (2007) PIC18FXX8 Datasheet

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

[11]. Vehicle Tracking Systems Overview [Online:] http://www.roseindia.net/technology/vehicle-tracking/VehicleTrackingSystems.shtml

[12]. Telit Wireless Solutions (2007) GM862-GPS Hardware user guide. 1vv0300728 Rev. 8 - 20/09/07