TCP/IP Based Vehicle Tracking and Fuel Monitoring Using Low Power Microcontroller

M. Shanmugasundaram, D. Karthikeyan, K. Arul Prasath
and R. Sri Raghav

Vellore Institute of Technology University, Vellore-632014, Tamilnadu, India
Correspoding author: mshanmugasundaram@vit.ac.in

Abstract

This paper is for transportation security and fleet management. An embedded device is designed and fabricated into PCB. The embedded device having SIM908, integrated GSM/GPS module, is kept inside vehicle which acquires the speed and location of the vehicle and send data to the server. The fuel level detection circuit calculates the fuel level from the fuel gauge which is present in all the vehicles. Connecting to internet requires additional money deducted by internet provider and the more use of battery. In order to be efficient, SMS function is included to start/stop internet connection. When the owner needs to track the vehicle, a SMS is sent to the device which then establishes the connection to the TCP server through GPRS. Data such as location speed and fuel level is sent to the server and stored in a database. The web page and android application designed for this purpose tracks the vehicle three dimensionally and show them in integrated map. Emergency call function is designed. The driver can call to the police, ambulance and the owner of the vehicle in emergency situations. The owner can dial a call to the vehicle at situations when the driver goes beyond the limits like high speed driving and wrong direction.

1 Introduction

Managing the fleet is difficult in these days. High fleet maintenance cost, security risks and raising fuel costs are the challenges faced by the owners of the fleet. Vehicle tracking system with fuel monitoring and emergency call function designed at low costs provides the solution for those problems. The challenges faced when designing this system includes size of the tracker device, cost of design, power constraint, server and software maintenance cost. The growing field of electronics provides us components very small in size, more efficient in working and is available at low costs. With these newest technologies a hardware device is designed.

2 Methodology

2.1 Hardware

SIM908 is a quad-band Global System for Mobile Communication (GSM) module which includes Global Positioning System (GPS) function for satellite navigation. Time and costs are saved because of its compact design which integrates General Packet Radio Service (GPRS) and GPS in a single SMT package. It allows tracking our assets seamlessly at any location and anytime with signal coverage. The input supply voltage range is 3.3 to 4.5 Volts. SIM908 is controlled via AT commands. The SIM908 is connected to micro-controller through Universal Asynchronous Receiver/Transmitter (UART). The integrated Transfer Control Protocol, TCP/IP stack makes it easy to establish TCP/IP connection. The data transfer supports two modes transparent mode and Non-transparent mode. In the transparent mode, the data received are sent serially to micro-controller automatically.

LP38501AT-ADJ is the linear voltage regulator. Flex Cap LDO's feature unique compensation that allows the use of any type of output capacitor with no limits on minimum or maximum ESR. This ultra-low dropout linear regulator responds very quickly to step changes in load, which makes them suitable for low voltage microprocessor applications. The input can be in the range of 3 volts to 5.5 volts. Output current is 3A. The output can be varied from 0.65 volts to 5 volts. The Voltage regulator has five pins and is surface mounted component. The adjust pin acts like feedback and helps the regulator to maintain the required output voltage effectively. The enable pin is used to start the voltage regulator.

TVS arrays are designed to protect sensitive electronics from damage or latch-up due to ESD and other voltage-induced transient events. SMF05C is

the surface mounted IC which is very small is size and provides up to five connections which can be used in parallel to the circuit to protect from the electrostatic discharge.

The MSP430F5419A an ultra-low power microcontroller is used in this project. It has four UART, SPI and I2C interfaces which can be used to add more features in future. There are 100 in which 87 are general purpose input/output pins.

The port 1 and port 2 has internal pull up resistors connected which can be enabled when required. It has 128KB flash memory which can be used for storing the collected data as well as the code.

The target board MSP-TS430PZ5x100 is designed by Texas Instruments for the purpose of testing and developing MSP430 F5 series micro-controller and its applications. The board has a JTAG port which is connected to the MSP430 FET USB programmer. The USB programmer is used to dump the codes in the micro-controller and can act as a simulator. External crystal of 32 KHz is used in addition to the inbuilt 25 MHz crystal. Lower the frequency of the crystal higher the accuracy the will be[1].

ADS1110 is a 16 bit ADC which will give a more accurate output in digital format. This is continuously self – calibrating Analog to digital converter with differential inputs is very in size. The ADS1110 uses I2c serial interface and operates from a power supply ranging from 2.7V to 5.5V

The entire tracker device design can be made as small as the size of a human's palm when the components mentioned above are used. They are small and power efficient. A PCB with two to three layers will be required to make such a small device.

2.2 Software

The webpage is designed with the Hyper Text Markup Language (HTML) and Personal Home Page (PHP) language. MySQL is used as database software to store the details of the users. Each user can have up to five number of vehicle which can be tracked at the same time. The user has to login, in order to track the vehicle owned. The three dimensional vehicle information shows the vehicle's direction, altitude, latitude, longitude and speed.

The website "*www.dkarthik.net23.net*" is designed. The website has access to the web tracker application. The android application can be downloaded from the website. The website is hosted by *000webhost.com* and is 24x7 online. But, the tracker application link, "*www.myproject.ttl60.com*", provided under the *Products and Services/Online Tracker* of the created website is hosted by

the home server in which the computer acts as the server. So, it works only when the computer used as the server is provided internet connection with IP address same as router's IP. If not port forwarding will be required to make server host the page online. The GPS Gate Server uses Internet Information Service (IIS) to host the webpage designed. IIS can only host one page at a time. Unlike Apache which can host more than one webpage at a time.

With the android application designed the user can track their vehicles at any time and in any place with their smart phones. The integrated Google maps will show the user the location of the vehicle marked in red color. The android application is created in ECLIPSE software which allows the program to access the GPS data from the tracker device to the phone and start tracking the vehicle.

Code Composer Studio V5 is used to write the micro-controller's code for the tracker, fuel level detection and emergency call. Cadsoft Eagle V6.3 is used to design the PCB.

2.3 Working Methodology

When the SMS is received, the GPS module starts collecting the NMEA data and sends it to the GSM. The NMEA data has latitude, longitude, speed, altitude, time date and direction of the vehicle. The collected data is sent to the server through a GPRS internet connection. A TCP/IP connection is established between the server and the Device. The data sent to the Gps gate server is stored in a database of that particular user which is identified by
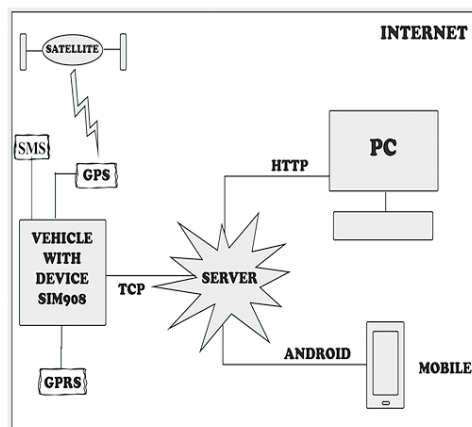
Figure 1   Working methodology

the IMEI number. User has to login to the website. The hosted website has embedded Google maps in vehicle can be tracked in real time. The connection can be closed with command given from website or through SMS. The micro-controller runs in the low power mode to reduce the power requirements by disabling interrupts and crystal oscillators.

3 System Structure

The six pin SIM card holder, net light indicator microphone and speaker are connected to SIM908 with the supporting circuits as suggested in the hardware design manual.

The net light indicator shows the network registration status of SIM card. The micro-controller is connected serially through UART to the SIM908 device. Both the device and modem supports Complementary Metal Oxide Semiconductor (CMOS) level hence no level shifters are required. The SIM908 is Data Communication Equipment (DCE) whereas the micro-controller is the Data Terminal Equipment (DTE). The Request to Send (RTS) and Clear to Send (CTS) line are data hardware flow control lines. When the micro-controller has to send the data the RTS is enabled and the modem acknowledges back with the enabled CTS signal. The Ring Indicator (RI) will be enabled for about 300ms when a call is received. Data Terminal Ready (DTR) is enabled has to be pulled down to show the DCE that DTE is ready to communicate.
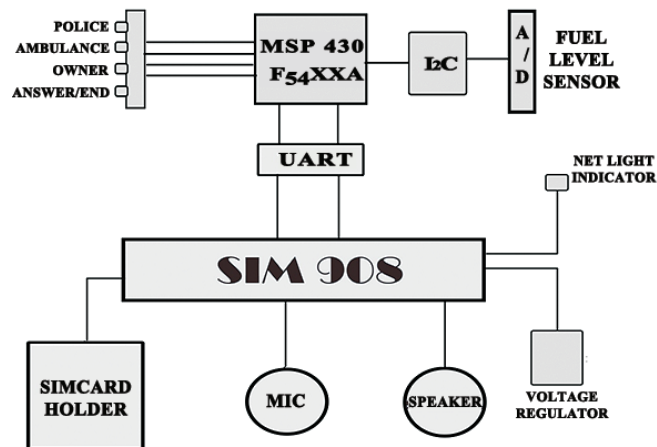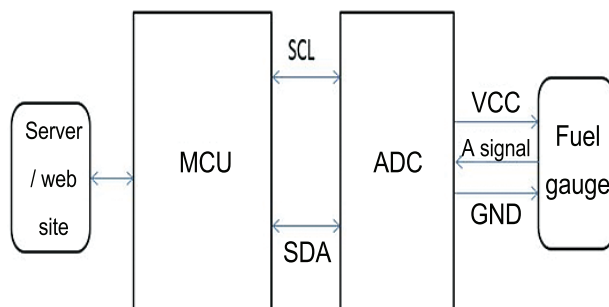

Figure 2 Basic structure of system

Figure 3   The overview of fuel level detector

3.1 Fuel Level Detection

The Analog to Digital (A-D) converter is connected to the micro-controller through Inter-Integrated Circuit (I2C) interface. The input to the converter comes from the fuel gauge kept inside the vehicle. Working of the A-D converter requires micro-controller programming. When the analog data is received by the A-D converter, the digital data is sent to the micro-controller and stored in a character array which is then converted to the fuel level in liters.

3.2 Emergency Call

Four button switches are connected to the second port of the micro-controller with the internal pull up resistors enabled. When the switches are pressed the corresponding calls are made to the respective people such as police, ambulance and owner of the vehicle. The switches can be set on the dash board of the vehicle, whereas the device is embedded inside the vehicle.

4 Microcontroller Programming Methods

The commands given to the SIM908 device starts with "AT+". Hence, the commands are called AT commands. These are stored inside the micro-controller and have to be sent to the SIM908 during the flow of the program upon meeting certain conditions. AT commands are stored in character array and sent serially to the SIM908 device. Each command sent the SIM908 acknowledged back with characters "OK". If some other characters are received then there is some error in the code or real difficulties caused in real time.

4.1 Emergency Call Algorithm

AT Commands are sent from micro-controller to the SIM908 device. Once an AT command is sent, next command is sent only when positive acknowledgement "OK" is received.

1. Initialize the variables, stop watch dog timer and enable interrupts.
2. Send the command "AT" to check the communication is successful. Modem replies back "OK"
3. Set the baud rate of 115200 by sending "AT+IPR=115200". After this command the program enters into the interrupt.
4. To answer the call the command "ATA" is sent when the button4 is pressed when the device is ringing. This same button is used for ending an ongoing call. "ATH" is sent to end the call when the button is pressed during the call period.
5. To dial a particular number the command "ATD<number>" is sent. Mobile number of the person to be called should be given instead of <number>. Three numbers should be called for three different buttons. Button1, dials to ambulance, Button2, dials to police, Button3, dials to owner of the vehicle.
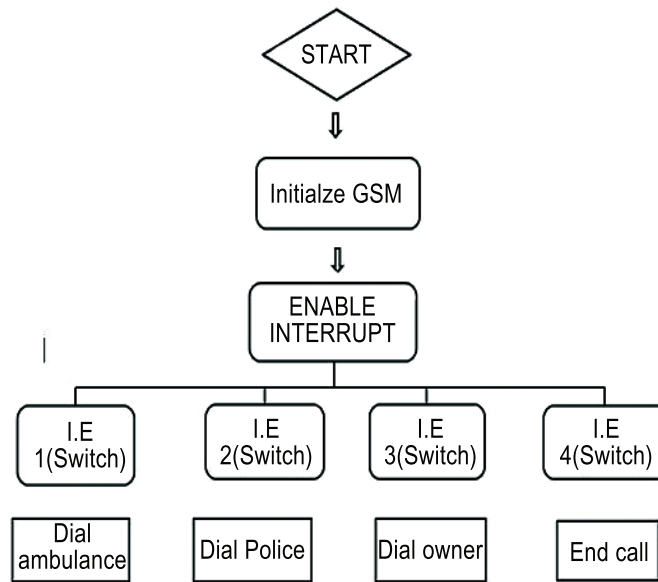
Figure 4   The working of emergency call function

4.2 GPS Data Collections Algorithm

Each command is sent only the positive acknowledgement "OK" is received for the previous command.

1. When SMS message containing "START" is received power on the GPS device with the command "AT+CGPSPWR=1".
2. First time after the device is powered on, the device has to be reset in cold mode. It takes 40 seconds in good satellite coverage to fix the location three dimensionally. Hence, the command "AT+CGPSRST=1" is sent. Waits for 40 seconds after sending this command.
3. To check whether the location is fixed or not. "AT+CGPSSTATUS" command is in a time interval of 20 seconds until location fixed acknowledgement is received.
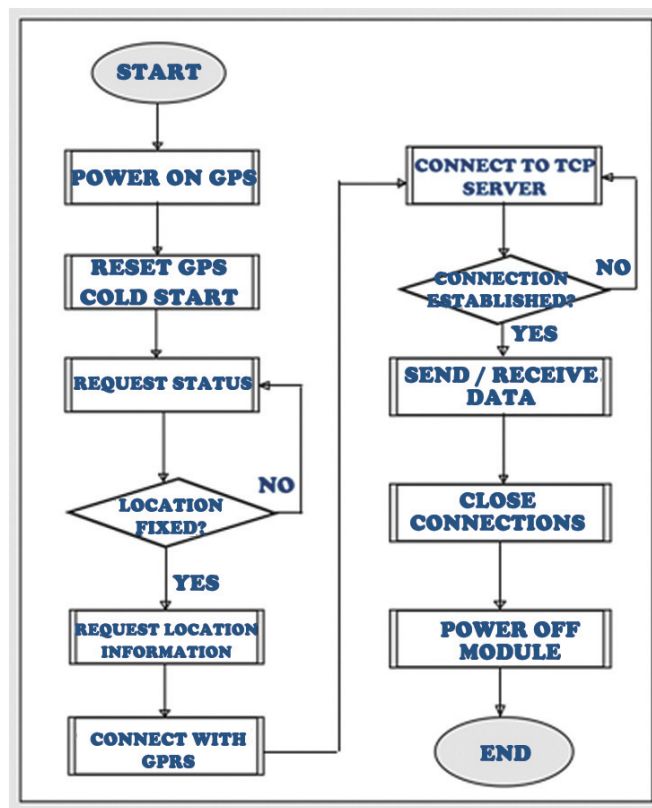

Figure 5   Flowchart of GPS and GPRS

4. Once the location is fixed, to get current GPS location the command "AT+CGPSINF=32" is sent. There are different formats in which the GPS NMEA data is received from the modem. The required format is chosen and the command for the particular format has to be sent to acquire the NMEA data.

5. The acquired data is not in the format the server needs. The format needed by the server is also not in the list of supported formats. Hence re-arranging, the acquired data is to be done and the check sum has to be added before the data is sent to the server.

6. By calculating the number of commas, the data can be stored in the different character array and can be merged together in the single character array. This is achieved by using number of for loops and conditional statements.

4.3 GPS Tracker One Server Protocol

4.3.1 Start tracking

This command is sent from the server to the tracker device. It has rules. The rules like at which time interval the device has to keep sending the location information[4]. The syntax of start tracking will be as follows

"$FRCMD,IMEI,_StartTracking,,Rule1=value1,Rule2=value2,...*XX"

4.3.2 Stop tracking

After the start command, the device keeps sending the location of the vehicle at certain time intervals. This command should stop the tracking. When this command is received, the micro-controller program should be written in a way to stop the Tracking and turn off the GPS and GPRS connections and enter low power mode[4]. Has syntax as follows

"$FRCMD,IMEI,_StopTracking*XX"

4.3.3 Send message

This command is sent from the micro-controller to the server. GPS data and fuel level is sent to server with this command[4]

Syntax is as follows

"$FRCMD,IMEI,_SendMessage,,DDMM.mmmm,N,DDMM.mmmm,E,AA.a,SSS.ss,HHH.h,DDMMYY,hhmmss.dd,valid,var1=value,var2=value...*XX"

Where "*XX" is the checksum.

Figure 6 Flowchart of fuel level sensor mechanism

4.4 Fuel Level Algorithm

1. Initialize the variables in unsigned integer.
2. Initialize i2c clock, set baud rate, enable interrupt.
3. Generate an acknowledgement and Not Acknowledgement signal.
4. Start I2C bus.
5. Stop I2C bus.
6. Read data from ADS1110 Data register and convert it with the fuel level value in liters.
7. Store the fuel level in the character array "fuel []". This is sent later with the send message command to the server.

4.5 GPRS Algorithm

1. To set the GPRS active, the command "AT+CGATT=1" is sent.
2. Start the GPRS connection by the command "AT+CGDCONT=1,"IP"," <apn_name>"". <apn_name> is provided by the internet service provider.
3. To start task and set APN, User name, password the command "AT+CSTT="<apn_name>,"<username>", "<password>""" is sent to SIM908.

4. To bring up wireless connection with GPRS the command "AT+CIICR" is sent.
5. To get local IP Address, the command "AT+CIFSR" is sent.
6. For knowing the current connection status the command "AT+CIPSTATUS" is sent. If reply is "CONNECT OK" the GPRS connection is successful.
7. To add an IP head at the beginning of package received "AT+CIPHEAD" is sent.
8. To start TCP connection, the command "AT+CIPSTART="TCP", <IP address>, <port>" is sent.
9. AT+CIPSTART" starts the TCP connection to the server.
10. Send the data through the TCP connection the command "CIPSEND" is sent. When ">" is returned by the SIM908, the data to be sent to the server should be written within five seconds. After five seconds the connection to the server will be closed.
11. If some data is received from server, it'll be sent to the micro-controller automatically by SIM908 because it's connected in transparent mode. From the received data, the required commands and data to the server are sent back using "AT+CIPSEND".

4.6 Android Applications

The application for the android phones is developed with the help of eclipse software and the language used is Java language[5]. Google map is integrated into the application. This is done by receiving API key from Google and generating a MD5 key from ECLIPSE[6]. Tracking the fleet involves the following steps

1. Java script code in android application sends the Asynchronous Java script (AJAX) request to the PHP script on the server every minute or on demand by the user using an event.
2. The PHP script will then obtain the last record for that user/vehicle from the database table and sends the latitude and longitude to the java script code running on the client side.
3. The application has embedded maps which shows the location of the vehicle

The application developed can be downloaded from the website "www.dkarthik.net23.net". This is the designed webpage to track the vehicle.

4.7 Web Page Design

The webpage is developed using HTML, scripting language like PHP and database by MySql. The Google map is embedded into the web application to view the fleet. These maps can be integrated by using the Google's API key. The web application is hosted in a home server whereas the website containing the information is hosted by "000webhost.com".

4.8 Result

The result obtained matched our design goals. PCB design worked properly without having much Electro-Magnetic interference and Electro-Static Discharge. The designed PCB and the assembly of the entire hardware is shown in the Figure 7. The GPS and GSM antennas are connected to the SIM908. The target board containing micro-controller is connected to the SIM908 in PCB board serially.

Figure 8 shows the tracked vehicle as red marker in the maps inside application website. The red marker marks the correct location *Kannamangalam*, near Vellore, where the device was set. The fuel level detected is displayed on the webpage in liters.
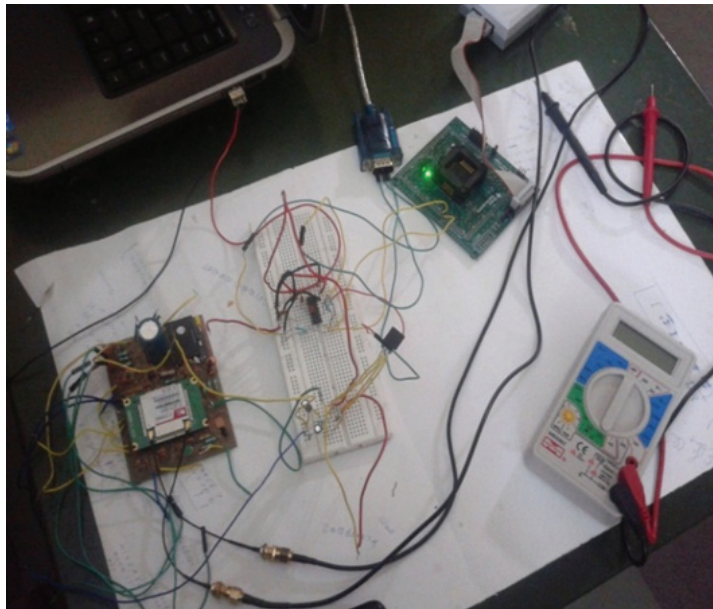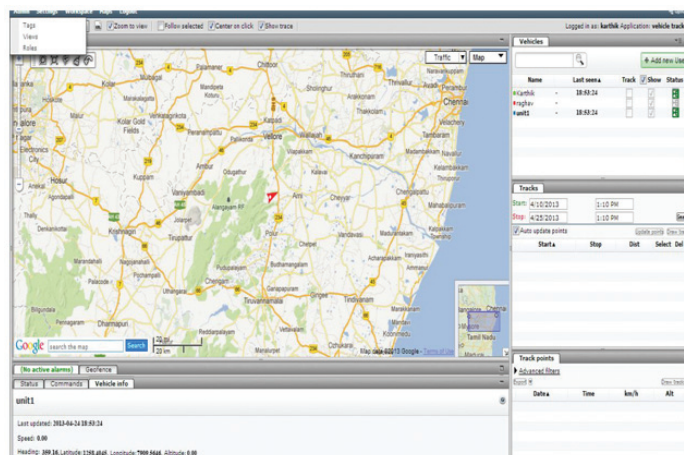

Figure 7   Hardware setup

Figure 8   Vehicle tracker output

5 Conclusion

The emergency call function has been tested and implemented in real time. When the buttons are pressed the calls are dialed/received as per the project's motive. The GPRS connection and the SMS function are utilized successfully to start and stop the data usage and connect to the internet. The TCP connection can be established successfully with the server. Location and fuel level can be shown in the website as well as in the android application. Thus, the tracker device works in real-time. As the further development, more number of sensors can be included to know more about the vehicle's condition and more SMS functions can be added. So that when the connectivity is lost the client can still know status about the vehicle.

References

[1] Sachin S. Aher and Kolkate R. D, "Fuel Monitoring and Vehicle Tracking Using GPS, GSM and MSP430F149" published - *international Journal of Advances in Engineering & Technology,* July 2012.

[2] Texas instruments web site and forum [Online] Available: http://www.ti.com

[3] The official Microsoft IIS site and forum [Online] Available: http://www.iis.net

[4] *SIMCOM's manual SIM908 Hardware Design V1.00*, June 2011

[5] The Gpsgate website and forum. [Online] Available: http://www.gpsgate.com

[6] The android application developer team. [Online] Available: http://www.developer.android.com

Biographies

M. Shanmugasundaram obtained a BE in Electrical and Electronics Engineering from Manonmaniam Sundaranar University, Thirunelveli, in 1999, and an M.E in Embedded System Technologies from Anna University, Chennai, in 2006. Currently he is a senior assistant professor in VIT University, Vellore. His research interests include scheduling in real time system and fault-tolerant computing.

D. Karthikeyan obtained his BTech degree in Electronics and Communications Engineering from VIT University, Vellore, in 2013. Currently he is working as software Engineer. His area of interests include real-time systems and networks.

K. Arul Prasath obtained his BTech degree in Electronics and Communications Engineering from VIT University, Vellore, in 2013. Now he is doing Master degree in Communication Engineering in the same university. His research interests include microcontroller, Automobile and communication network.

R. Sri Raghav obtained his BTech degree in Electronics and Communications Engineering from VIT University, Vellore, in 2013. Currently he is doing his master in Business Administration. His area of interests include software development for various platform.