Smart Transportation System

J. Dani Reagan Vivek¹, R. Gokul Prashanth², B. Ganesh Prabhu³

¹Assistant professor, ²³UG students

Department of Electronics and communication Engineering
MEPCO SCHLENK ENGINEERING COLLEGE
SIVAKASI

Abstract –This paper describes the design and implementation of the hardware and software for tracking the automobile which is capable of monitoring its own location. The unit also has the ability to report its location to a central server. The firmware for the tracking unit was implemented in the Arduino programming language and runs natively on Arduino Mega 2560 architecture. A PHP application was developed to enable central logging of locations, as well as exporting this data to Google Maps and Google Earth. The system was successfully implemented and a number of field tests were performed to validate specifications. The system is able to accurately track a vehicle or asset for an extended period of time. A unit’s position could be visualized using either Google Earth or Google Maps. SMS control and monitoring was possible for the majority of the units operating time.

1. Introduction

This Project talks about the design of a hardware for Tracking automobile, and a Web application for a Smart Transportation System. The Application reads the location of the bus for every stipulated time and provides a close proximity location of the automobile with the seats availability form the central server. The main objective of this project is to enhance public transport system to a digitalized and a smart system and to notify the user with the accurate and a brief status of the automobile and also to have a persistent monitoring of the automobile.

The challenges during this development were: (i) to develop a device with the ability to track and report its current location. (ii) to design the device, placing components in such a way as to minimize the device’s size and cost, as well as maximize power efficiency; (iii) to create a firmware implementation for the designed hardware, which has the ability to communicate with existing asset tracking software technologies, by using predetermined data protocols and access mediums; (iv) to develop and implement intelligent monitoring software, with the ability to react on external variables that are relevant to the asset being tracked.

The project description identified a number of challenges that had to be successfully solved for a complete implementation, these include the items such as The ability for a unit to determine its own location. The ability to manage its own power source and monitor power utilization. The ability to receive commands and execute the appropriate action. The ability to communicate with an external host to upload location updates for centralized monitoring.

2. Literature study

Localisation is a large research area [4] but this paper will mostly focus on GSM-based methods. GSM based location finding is in no way a new technology, it has been improved in a number of ways in the last couple of years [11], but is only now reaching maturity. The basic premise has stayed the same: a GSM module looks up the base stations in its current range in a central database and through a number of algorithms try and triangulate its own position.

The downfall for this method is that a number of base stations need to be in range with little to no interference for an accurate estimation [12]. This, combined with the fact that not even an approximate fix can be achieved when no base station is in range, creates some major drawbacks to this method. It was instead decided to implement this design using GSM (global system for mobile), as well as GPS technology. Since GPS has near global coverage, this eliminates the above problem almost completely. GPS technology while still susceptible to jamming [10] has seen many improvements in robustness over the last couple of years. Many vendors produce modules resistant to this type of attack but the ease of use and availability of jammers is a significant threat to GPS based tracking technology.

3. Design and Implementation

The first step in designing this system was to develop the individual system units. The system functional block diagram as determined from the user requirements.

The system is composed of a number of different units. Arduinomega 2560 gathers input from a GPS (global positioning system), Accelerometer and GSM (global system for mobile) modem. The block diagram is shown in the Figure 3.1. The combination...
of these units allows the system to operate in a stand-alone fashion as well as to ascertain its own location and transmit it to a central location for logging.

To detect physical impact on vehicles and mishandling of packages, an impact sensor was needed. The impact experienced in a vehicle collision can easily exceed 10G; it was decided to equip the unit with a 7G accelerometer. This was adequate since only detection was needed. The accelerometer was set up to operate independently from the microprocessor and only triggered an alert when a predefined acceleration value was exceeded. This enabled the system to issue immediate emergency updates to the monitoring server in the event of a high impact collision or rough handling of a monitored package.

It was decided to use a digital accelerometer that could operate independently as to reduce processing overhead. The MMA7455 accelerometer was chosen; it has all the above mentioned features and communicates over the I2C bus. The microprocessor chosen ideally would need built-in hardware controls for I2C.

The GPS accuracy was accepted to be within 10 meters as the primary use would be for vehicle tracking. This is easily obtained with most GPS units on the market. An update interval of no less than once every ten seconds was required which was also easily obtained. The GPS receiver was designed to integrate directly to the SIM800a GSM modem and allowed using only one UART channel for communication to both units. These two units were first prototyped to ensure connectivity before being incorporated into the design. A firmware upgrade for the SIM800a was needed to attain compatibility with the GPS receiver. Once connectivity between these two units was confirmed, basic tests were performed to test functionality.

For SMS and GPRS connectivity the SIM800a GSM modem was used. Only the basic features of this unit were used and no developed code was uploaded to the unit. Communication between the modem and a computer was first confirmed before incorporating it into the design. A number of tests were done to confirm valid operation of the unit. These tests included testing SMS and GPRS connectivity as well as reading model specific information. The proposed model is shown in the Figure 3.2.

The data output by the GPS was in NMEA 0183 (National Marine Electronics Association) format and had to be converted to longitude and latitude positions before being processed further. The location updates could then be sent to the server for logging. The GPS used was set up to operate with a passive GPS antenna but the design could easily be modified to allow for both passive as well as active antennas.

The hardware implementation of this project is shown in the Figure 3.3.

For Arduino mega 2560, it was decided to not use the ARM processor on the SIM800a GSM modem; this was done so the GSM modem could be turned off, saving power. The goal then was to choose a microcontroller with I2C and UART hardware interfaces, as well as a clock speed high enough to handle all the needed processing. The Arduino mega 2560 from Arduino was chosen for this component; it matched the above mentioned requirements and was fairly well priced. In addition, the hardware and software needed for development was easy to obtain. The unit was tasked with the following. The GSM modem enabled communication to and from the unit using both SMS messages as well as GPRS data connections. Both of these were essential since logging information could be sent to a central location for fairly little cost periodically, or requested immediately for the cost of two SMS messages.
GPRS is a packet oriented mobile data service on the 2G and 3G cellular communication system’s global system for mobile communication (GSM).

GPRS usage is typically charged based on volume of data transferred, contrasting with circuit switched data, which is usually billed per minute of connection time.

GPRS extends the GSM Packet circuit switched data capabilities and makes the following services possible: SMS messaging and broadcasting, “Always on” internet access, Multimedia messaging services (MMS), Push-to-talk over cellular (PoC), Instant messaging and presence, Point-to-point (P2P) service: inter-networking with the Internet (IP), Point-to-multipoint (P2M) service and point-to-multipoint group calls.

GPRS supports the following protocols: Internet Protocol (IP) - In practice, built-in mobile browsers use IPv4 since IPv6 was not yet popular. Point-to-Point Protocol (PPP) - In this mode PPP is often not supported by the mobile phone operator but if the mobile is used as a modem to the connected computer, PPP is used to tunnel IP to the phone. X.25 connections - This is typically used for applications like wireless payment terminals.

The protocol for communicating with the server was HTTP (hypertext transmission protocol) over TCP/IP (transmission control protocol over internet protocol). This ensured message integrity and easy integration into existing systems. The default update server could be set on a per unit basis using simple SMS commands. In addition, a unit could be queried for a location update to a mobile phone via a specially formatted SMS.

To understand the NMEA message structure, let’s examine the popular $GPGGA message. This particular message was output from an RTK GPS receiver.

$$GPGGA,181908.00,3404.7041778,N,07044.3966270,W,4,13,1.00,495.144,M,29.200,M.010.0000*40$$

All NMEA messages start with the $ character, and each data field is separated by a comma. GP represents that it is a GPS position (GL would denote GLONASS). 181908.00 is the time stamp: UTC time in hours, minutes and seconds. 3404.7041778 is the latitude in the DDMM.MM format. Decimal places are variable. N denotes north latitude. 07044.3966270 is the longitude in the DDDMM.MM format. Decimal places are variable. W denotes west longitude. Four denotes Quality Indicator:

1 = Uncorrected coordinate
2 = Differentially correct coordinate (e.g., WAAS, DGPS)
4 = RTK Fix coordinate (centimetre precision)
5 = RTK Float (decimetre precision).

13 denotes number of satellites used in the coordinate. 1.0 denotes HDOP (horizontal dilution of precision). 495.144 denotes altitude of the antenna. M denotes the units of altitude (e.g. Meters or Feet). 29.200 denotes geoidal separation (subtract this from the altitude of the antenna to arrive at the Height Above Ellipsoid (HAE)). M denotes units used by the geoidal separation. 1.0 denotes age of the correction (if any). 0000 denotes correction station ID (if any). *40 denotes checksum. The $GPGGA is a basic GPS NMEA message. $GPGSA – Detailed GPS DOP and detailed satellite tracking information (e.g. individual satellite numbers). $NGNSA for GNSS receivers. $GPGSV – Detailed GPS satellite information such as azimuth and elevation of each satellite being tracked. $NGNSV for GNSS receivers. $GPGST – Estimated horizontal and vertical precision. $NGGST for GNSS receivers.

4. Results:

The main test for the developed prototype was a drive from Virudhunagar to Sivakasi in a vehicle equipped with the tracking unit. These could be displayed in Google Maps or Google Earth with ease and allowed visualizing the exact location of the vehicle with timestamps attached to each location point. The automobile tracked inside the Mepco Schleck engineering college pointing ECE block was retrieved using the web interface and displayed using Google Earth as shown in the Figure 4.1. The data could be easily visualized and allowed tracking the unit every step of the way. The system could accurately track assets for an extended period. The GPS location reported by the unit was almost always accurate to within 10 meters and no loss of signal was experienced during the testing phase.
Signal loss however is possible but wasn’t experienced during testing.

![Web interface using Google Earth](image)

Figure 4.1: Web interface using Google Earth

The SMS commands used by the unit functioned as expected but sometimes had a delay as long as 120 seconds. This delay is considered acceptable as GSM network latency is unavoidable and not outside acceptable limits for a system of this nature. The measured latitude and longitude position taken in various places inside our college campus is shown in the Figure 4.2

![GPS position taken in various places inside Mepco college campus](image)

Figure 4.2: GPS position taken in various places inside Mepco college campus

5. Conclusion

A proof of concept system was developed to assist small businesses and individuals with asset and vehicle tracking. The system eliminated the need for a third party to maintain the infrastructure and manage the system. An integrated system consists of embedded tracking units capable of standalone operation as well as determining their own location and reporting this via SMS or GPRS, firmware for the tracking units written in Arduino tasked with all client side functionality and server side monitoring software written in PHP tasked with logging location updates and exporting this to either Google Maps or Google Earth, allowing the user to track multiple units in real time.

The system could accurately track assets for an extended period. The GPS location reported by the unit was almost always accurate to within 10 meters and no loss of signal was experienced during the testing phase. Signal loss, however, is possible, but was not experienced during testing.

6. Reference


