PPU College of Engineering and Technology

The Home of Competent Engineers and Researchers

Collage of Engineering and Technology,
Collage of Information Technology and Computer Engineering

Communication, Computer and Automobile Engineering Program

Bachelor Thesis

Graduation Project

Intelligent Traffic with Automatic Violation System

Project Team

Khalid I. Baradia – Communication Engineering
Abdullah Z. Ja'abary – Communication Engineering
Hassan A. Ja'afreh – Computer engineering
Mustafa F. Zaitown – Automobile Engineering

Project Supervisor/s

Dr. Murad A. Abusbaih
Dr. Zuhdi H. Salhab

Hebron – Palestine
May 2013

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة والتكنولوجيا

كلية تكنولوجيا المعلومات وهندسة الحاسوب

دائرة الهندسة الكهربائية وهندسة الحاسوب، دائرة الهندسة الميكانيكية

اسم المشروع

Intelligent Traffic With Automatic Violation System

أسماء الطلاب

خالد برادعيه عبدالله الجعبري حسن جعافرة مصطفى زيتون

بناء على نظام كلية الهندسة والتكنولوجيا واشراف ومتابعة المشرف المباشر على المشروع ومتابعة أعضاء اللجنة المستحنة، تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية والحاسوب ودائرة الهندسة الميكانيكية، وذلك استكمالا لمتطلبات درجة البكالوريوس في تخصصات: هندسة الاتصالات والالكترونيات، هندسة الحاسوب وهندسة السيارات.

توقيع المشرف

17/6/13

توقيع اللجنة المستحنة

--- --- ---

توقيع رئيس الدائرة

Dedication

To our families and our parents whose constant inspiration and encouragements drives us to become all that we can be.

Our appreciation goes to our teachers and all staff of the faculty of engineering.

We give thanks for them for being our mentors, and our friends; it is indeed a privilege for us to be their students.

Khalid I. Baradia

Abdulla Z. Ja'bary

Hasan A. Ja'freh

Mustafa F. Zaitown

Acknowledgement

We would like to thank our mentors Dr. Murad Abusbaih and DR. Zuhdi Salhab, who read our numerous revisions of this document, and who were generous enough to give us the support we needed to make this project.

We must not forget to thank all the instructors in the Electrical and Mechanical Engineering Departments for their great impact in our education.

Special thanks to Eng.Abeer Imtair, Eng. Rawan Abu Aisheh, Eng. Sami Salamine and Eng. Ra'fat K. Junadi, for all his care and valuable advices.

We thank the Hebron municipality for its logistical support, and belief in our abilities.

We acknowledge the Palestine Polytechnic University for giving us the opportunity to experience real engineering and exhibit some of the things we've been taught over the course of the last five years.

Finally we can't forget to acknowledge our great parents who gave so much to offer us the education we wanted. We truly cannot ever repay them for the love and assurance they entrusted in us.

Khalid I. Baradia
Abdulla Z. Ju'abary
Hasan A. Ju'afreh
Mustafa F. Zaiteon

Abstract

This project aims to plan and build a system to reduce the number of vehicle accidents happening every day due to the illegal speed. System will present a model for vehicles speed monitoring, and sending violation reports automatically using GSM network based on real time localization system GPS. In additional, an automatic system based on sensors used to alarm the ambulance about where the accidents occurred, exactly when they are happening.

The system based on sensing the location of the vehicle using a database. When the vehicle's speed exceeds a pre-defined speed limit, the violation occurred, recorded and sent as a message to the corresponding driver, then the mechanical part of the system will reduce the vehicle speed automatically.

يقوم هذا المشروع على بناء نظام يهدف إلى تقليل عدد حوادث السير الناجمة عن الزيادة الغير قانونية في سرعة المركبات، من خلال استخدام نظام التوقيع الكوني (GPS) وشبكة الاتصالات الخليوية (GSM) وابتكار نظام ميكانيكي يهدف الى تقليل سرعة السيارة للحد القانوني المسموح به.

يقوم هذا المشروع أيضا على بناء نظام في داخل المركبات لتحسس الحوادث، وعند وقوع الحادث يعمل هذا النظام على إعلام مراكز الإسعاف الأولي عن موقع هذا الحادث، بهدف الإسراع في عملية إسعاف المصابين.

يمثل هذا المشروع جهدا مشتركا ما بين وزارة الاتصالات ووزارة المواصلات والجهات التابعة لها، يستهدف المشروع القطاع العام من خلال المساعدة في تقليل حوادث المرور ، ويقدم الدعم التقني والفني لقطاع المواصلات من خلال نظام تحرير مخالفات السير الالكتروني.

Table of Contents

TABLE OF FIGURES

List of tables

1

Chapter one

Introduction

Chapter 1:
Introduction

1.1 Overview

This project aims to plan and build a system, which will be used to reduce the amount of vehicle accidents, which occur every day because the illegal vehicle speed. We propose a model for the speed monitoring of vehicles, and create an automatic violation system using GSM network-based on real-time localization system GPS. This model is based on sensing the location of the vehicle depends on the built in database. When the vehicle speed exceeds the speed limit, then the violation occurred and recorded.

Despite all attempts to prevent traffic accidents, it's still happening. It doesn't stop reckless driver from the commission of doing something that may take stock of himself throughout his life.

Prevention is better than cure, this project works with this statement for facing the biggest disease "traffic accidents". It Provides a new mechanism for stopping increasing of the speed more than the limit, and this dramatically reduces the traffic accidents, but if it happened; and this is the rate of a few, the system can help in speed up the rescue operation, and solve the delay problem in the ambulance arriving to the accident location by automatic alarm the emergency center about the location of accident once it occurred.

1.2 Project Objectives and Motivation

Malaria, AIDS, Cancer, Hepatitis, these are the most deadly diseases. Traffic accidents are going to become with these diseases. Studies and statistics have proven that the traffic accidents increase from year to year. Driving at a high speed, delay in the ambulance to transport injured to the hospital in the reasonable time; making the accidents more fatal.

Hundreds of people are killing every day, and thousands injured on roads. Men, women or children who are walking, riding to school or work, will never return home, leaving behind shattered families and communities. Millions of people each year will spend long weeks in hospitals after severe crashes, and many will never be able to live, work or play as they used to do.

Driving more than the speed limit of the streets cause the proportion of 31% of fatal traffic accident. Delayed ambulance access to the accident makes it worst and fatal.

Statistics proved that the number of traffic accidents is moving towards an annually increasing, accompanied by a steady increase in the number of injuries, deaths and material losses, as shown in figure (1.1). Therefore, we can appreciate the direct physical loss (damage to vehicles) and indirect (environment and health damage), etc [1].

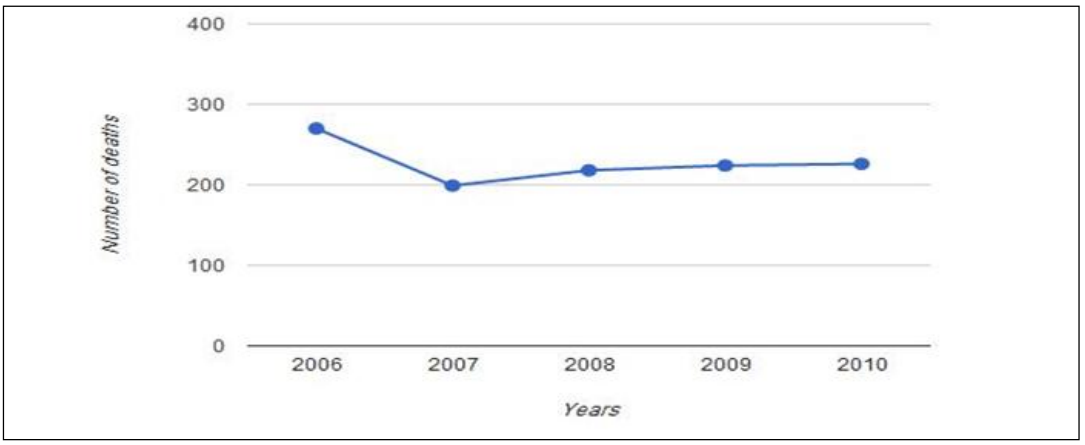


Figure 1.1: *Annual traffic accidents deaths*

This project aims to develop a system to reduce the number of accidents occurred every day, especially by illegal speed, in addition to develop an automatic system to detect the transport violation without needing external devices such as Cameras , Radar, etc.

This system will alarm the driver that his life is in danger, alarm him about his violation, and maintain an automatic system to alarm the ambulance where the accidents occurred, exactly when they are happening.

1.3 Approach

This project requires deep knowledge in GPS system with suitable coordinate system to read live coordinates from where the vehicle seat. This will make the sophisticated vehicle guidance and advisory systems such as ITWAVs available and enable.

Since the GPS receivers have some accuracy errors, caused by high buildings, tunnels, etc., we need to design a Correction algorithm, which will match the current value of GPS with the reference point stored in database.

Android programming language will be used. It is known as a common programming language for mobile programming. Thus, the system will be applicable and could be used with a wide range of devices such as PDAs, Galaxy phones, etc.

The android navigator will be interfaced with a microcontroller device, and this device is used to make interfacing with whole system sensors.

This system has a central system API connected with the database, and the GSM modem built on PC to receive violations and store it in the database. The central system will be programmed using C# language and SQL server database.

When the speed increased above the specified speed of the road, the violation occurs, and then control signal sent to the microcontroller, which activate the break solenoid motor, and deactivate the accelerator pedal.

1.4 Assumptions and Requirements

The following assumptions are considered in this system:

1) Each vehicle will contain an In-vehicle GPS navigator, which will receive data from GPS system, process it, and compare the result with the stored database.
2) Each vehicle will contain an Arduino Microcontroller, which will make an interface between mechanical system and GPS navigator.
3) Each vehicle must contain multiple piezoelectric sensors, which will send signal to the microcontroller when the accidents occurred.
4) Each vehicle must contain APPS, ABS system, which will be used to decrease the vehicle speed as it exceeds speed limit.
5) Data that received from GPS and processed, in addition to the street speed limit will collect as a packet, and then send over GSM network using GPRS modem to the central server.
6) Emergency Server: which will receive longitude and latitude data from GPS once the accident occurred, and determine the exact location where the accident happened, then send Ambulance to that location.

1.5 Literature Review

The idea can be seen in different systems .however, the proposed system is expected to provide better accuracy at lower cost.

The following section will summarize the previous related works, and describe its principle of operation, then display the advantages, disadvantages, and the differences between our system and those systems.

1.5.1 Vehicle Information and Communication System (VICS) in Japan [2]

This system uses roadside beacon unit, which is a transmitter device located on the speed limit sign .This unit transmits signals contain information regarding the road speed limit. The vehicle receive this information, and when it exceeds speed limit the system makes warring to the driver .This system uses mobile communication according to update the speed limit of the road .

Advantages:

1) This system is simple, when the vehicle appears the system works.
2) The easy of updating the speed limit by using the mobile communication.

Disadvantages:

1) The cost of this system, if we need to consider all roads in the city.
2) The cost of maintain.

Differences:

1) This system uses a transmitter device located on the speed limit sign, while our system depends only on GPS system.
2) This system has criteria only for alarm the driver when exceeding speed limit, while our system develop a violation system.
3) This system Just warning when the vehicle exceeds the speed limit, without speed monitoring.

1.5.2 Active Distance Support System (ACDIS) [3]

This system uses the distance sensor located in the nose of vehicle, and this supports longitudinal guidance. For example, if we have two vehicles, second behind the first, the second one must not beyond the safety distance with the first one , the distance sensor gives the distance between the two vehicles to the monitoring speed system ,which controls the force feedback Pedal .

Advantages

1) Force the accelerator pedal when driver falls below the safety distance.
2) Distance sensor is low cost, so we can apply this system for all vehicles.
3) Distance sensor inside the vehicle, no need to fix anything out vehicle like sensors on the traffic sign as infrared beacon.

Disadvantages

It just gives the solution for the safety distance between vehicles but not monitoring the speed of the vehicle when it exceeds the speed limit.

Differences

1) This system decreases the number of accidents by conserve the fixed distance between two vehicles, while our system decrease the number of accidents by develop a design depend on the speed of vehicles.
2) This system requires distance sensor in each movement vehicle, while our system requires GPS receiver in each vehicle.

1.5.3 Intelligent Speed Adaptation (ISA) [4]

This system uses GPS receiver with roadside beacons , it's the same as VICS system but it contain GPS receiver to match the coordinate that received from roadside beacons .

The roadside beacon sends a data to the vehicle .The data is formatted of latitude and longitude and its corresponding road sign speed limit. The controller compares the GPS coordinate with the latitude and longitude that received from road sign beacon, when the current position reaches the position of data, corresponding alert is done.

Advantages

1) It contains GPS receiver that makes the system more accurate.
2) Easy to update the speed limit using mobile communication.
3) Low cost.

Disadvantages:

1) The cost of this system if we need to mention all roads in the city.
2) The cost of maintain.
3) Just warning when the vehicle exceed the speed limit.

Differences:

1) This system using roadside beacons devices located on the speed limit sign, and using GPS receiver to achieve high accuracy, which increase the system cost, while our system depends only on GPS system in addition to correction algorithm which used to increase the system accuracy.
2) This system has criteria only for alarm the driver when exceeding speed limit, while our system develop a violation system.
3) This system Just warning when the vehicle exceeds the speed limit, without speed monitoring.

1.6 System Schedule

The following table, describes the overall stages of the system:

Table 1.1: *Overall system timing table for the first semester*

Timing Table (in Weeks)																
Task/Week	**Summer**	**First semester**														
		1th	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**11th**	**12th**	**13th**	**14th**	**15th**
Selecting The idea of the project																
Search and collecting information																
Preparing for the project																
Studying the Principles																
System and requirement analysis																
In vehicle Map design																
Crashing system design																
Software and Server design																
Mechanical System design																
System parts collection																
Conclusion																
Documentation writing																
All members	Communication Team	Computer Team	Mechanical Team													

Table 1.2: *Overall system timing table for the second semester*

Task/Week	**Second semester**														
	1th	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**11th**	**12th**	**13th**	**14th**	**15th**
Pay the hardware component															
Preparing software skills for the project															
Build up the software															
Testing the software															
Testing the hardware															
Testing the system															
Documentation writing															

All members Communication and computer team Mechanical Team

2

Chapter Two

Theoretical Background

Chapter 2:
Theoretical Background

2.1 Introduction

This chapter presents the basic theoretical information, by focusing on the technologies and components that the system needs, and learn about some devices that will be used in our system. In order to use them in an appropriate way and take the desired advantages of them, so make it easy for the reader to understand and interact with this system.

This Chapter will be divided into two parts: electrical and mechanical part. It shortly displays the most important ideas about the system technologies that will be used, and the detailed information will be displayed in the next chapters.

We will talk about GPS technology, GSM/SMS technology, and a collection of hardware components that will be used in this system includes GPS navigator, GSM/GPRS modems, and Arduino microcontrollers.

Software that will be used presented by programming languages that will be used in the Mobile phones or Tablets as a client programming language, in addition to the central system programming language, and microcontroller and SMS/GPRS modem programming commands.

Part A: Electrical elements

2.2 GPS Technology

2.2.1 Definition

The global positioning system (GPS) is a system Consists of constellation of 24 well-spaced satellites that makes a revolution every 12 hours, used for determining the location of people. The 24 satellites broadcasting data that allows many users to determine their spatial position simultaneously. Navigation System with Time and Ranging (NAVSTAR) is a global positioning system follow American Defiance Ministry that most application based on it [5].

2.2.2 GPS system overview

The first GPS satellite was launched in 1978, a full constellation of 24 satellites was achieved in 1994, and each satellite is built to last about 10 years. Replacements are constantly being built and launched into orbit. A GPS satellite weighs approximately 2000 pounds and is about 17feet across, with the solar panels extended. Transmitter power is only 50 watts or less [6].

2.2.3 GPS system components

The GPS technology consists of the following components [7]:

- **The Earth**: it is the major component of GPS, and the space immediately above. The mass of the Earth holds the satellites in orbit.
- **Earth-circling satellites:** Twenty-four solar powered radio transmitters, forming a constellation such that several are visible from any point on Earth at any given time. The satellites are at middle altitude of 20200 kilometers (Km); each is in a 12-hour orbit.

- **GPS Receivers:** it consists of an antenna whose position the receiver reports, and electronics to receive the satellite signals, in addition to a microcomputer to process the data that determines the antenna position, and recording position values.
- **The United States Department of Defense:** for developing and maintaining GPS satellites.

2.2.4 GPS in our System

We will use GPS to determine the real time location of the moving vehicle, using GPS receiver built in the android navigator device. This operation needs GPS receiver plus some processing to calculate the position. The navigator device takes GPS measurements, and compares it with the saved GPS coordinates or stored database in GPS navigator device to find in which road the vehicle move.

2.3 GSM Technology

2.3.1 Overview

GSM (Global System for Mobile Communications) is the world's most popular wireless phone technology. More than one billion people around the world use it.

GSM offers unparalleled global roaming capabilities, as well as the truest voice quality in wireless. It’s easy-to-use data capabilities offer service up to the fastest wireless broadband service currently possible [9].

There are several technologies that companies used to provide wireless services. While there may never be one standard technology worldwide, GSM is now used in 219 countries and territories serving more than three billion people and providing travelers with access to mobile services wherever they go.

2.3.2 SMS Technology

Short Message Service (SMS) is a text messaging service component of phone, web, or mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices. [11]

SMS is the most widely used data application in the world, with 3.6 billion active users, or 78% of all mobile phone subscribers [12]. The term "SMS" is used as an acronym for all types of short text messaging and the user activity itself in many parts of the world. SMS is also employed in direct marketing, known as SMS marketing.

2.3.3 SMS working criteria [13]

Once a message is sent, it is received by a Short Message Service Center (SMSC), which must then get it to the appropriate mobile device, to do this, the SMSC sends a SMS Request to the home location register (HLR) to find the roaming customer. Once the HLR receives the request, it will respond to the SMSC with the subscriber's status: inactive or active where subscriber is roaming.

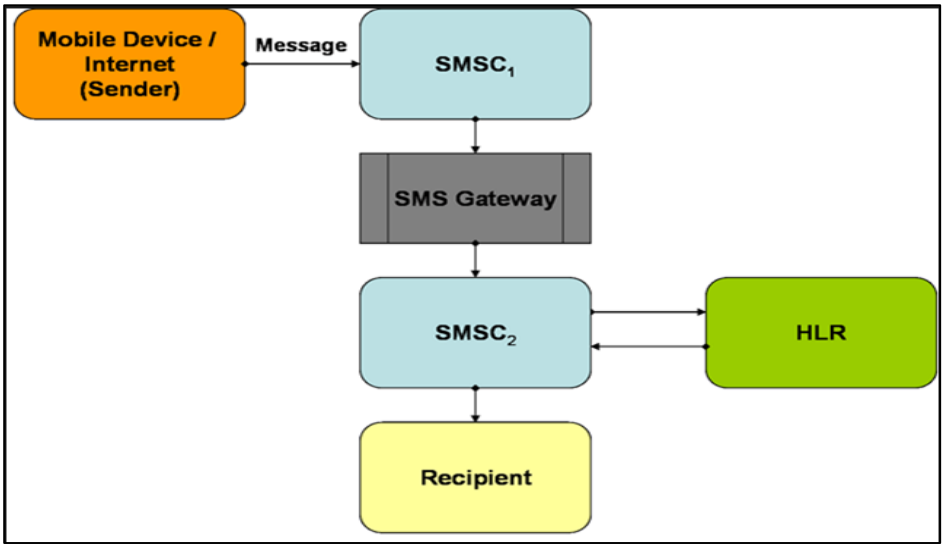


Figure 2.1: SMS working criteria

If the response is "inactive", then the SMSC will hold onto the message for a period. When the subscriber accesses his device, the HLR sends a SMS Notification to the SMSC, and the SMSC will attempt delivery.

The SMSC transfers the message in a Short Message Delivery Point to Point format to the serving system. The system pages the device, and if it responds, the message is delivered.

The SMSC receives verification that the message was received by the end user, then categorizes the message as "sent" and will not attempt to send again.

2.3.4 SMS Protocols

The SMS protocol stack for the CDMA mode of operation is illustrated in Figure (), shaded areas indicate the protocol elements covered in this standard [14].

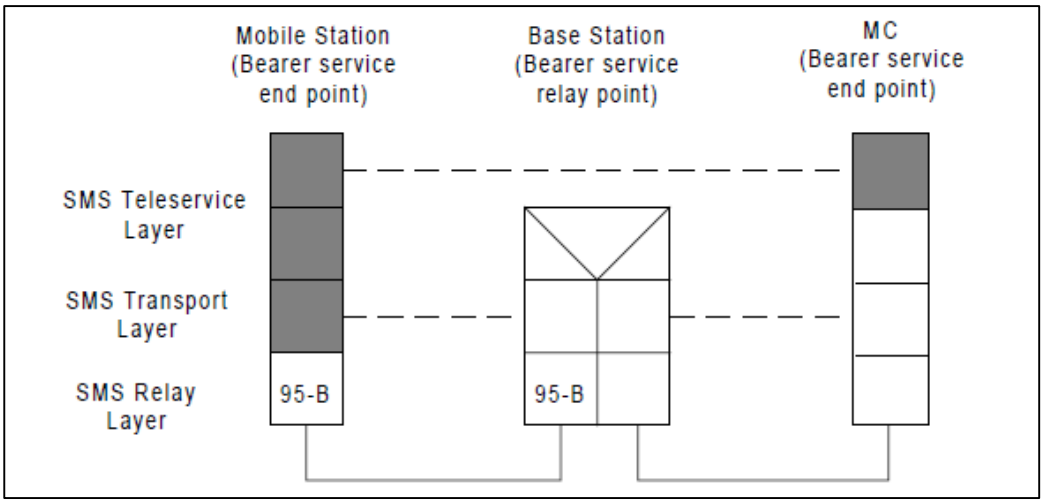


Figure 2.2: SMS Protocol stack

The SMS bearer service is the portion of the SMS system responsible for delivery of messages between the MC and mobile user equipment. The bearer service is provided by the SMS Transport Layer and the SMS Relay Layer.

The SMS Transport Layer is the highest layer of the bearer service protocol. The Transport Layer manages the end-to-end delivery of messages. In an entity serving as a relay point, the Transport Layer is responsible for receiving SMS Transport Layer messages from an

underlying SMS Relay Layer, interpreting the destination address and other routing information, and forwarding the message via an underlying SMS Relay Layer.

The SMS Relay Layer provides the interface between the SMS Transport Layer and the Link Layer used to carry short message traffic. This standard addresses the SMS Relay Layer in mobile stations and base stations and their interfaces to the Link Layers.

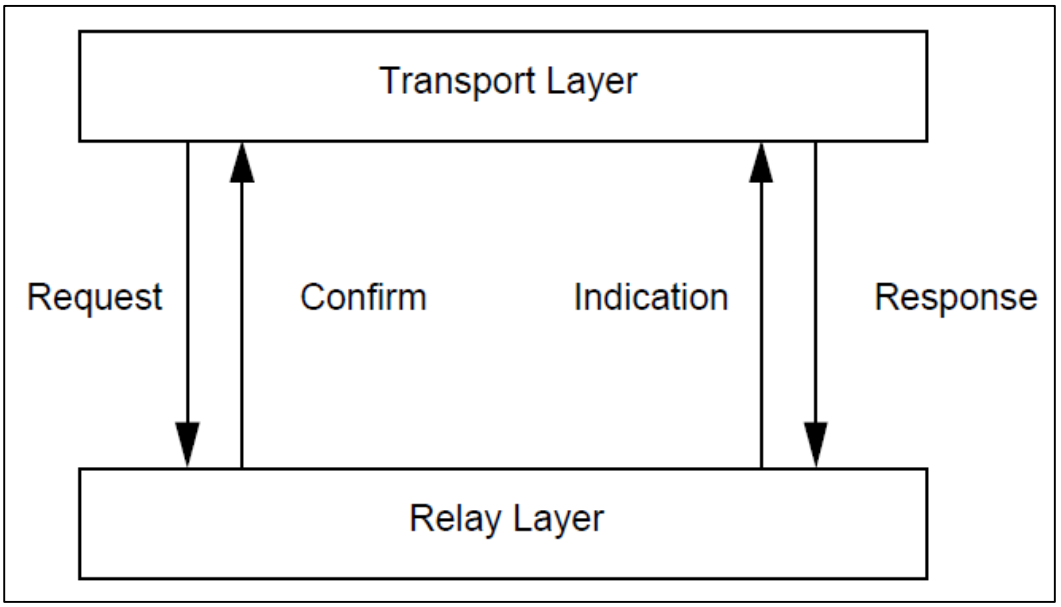


Figure 2.3: *SMS Service layers*

The SMS Teleservice Layer resides in a Bearer Service end point. The Teleservice Layer supports basic SMS functions through a standard set of sub-parameters of the Transport Layer's Bearer Data parameter. The Teleservice support defined in this standard is restricted to definition of the Teleservice messages and their contents, and definition of the minimum set of procedural requirements necessary to ensure compatibility. The message type is indicated in the Message Identifier sub-parameter.

2.3.5 Requirements for using SMS

- An application with a connected GSM/GPRS modem.
- An available GSM/GPRS network.
- A SIM card with an activated GPRS service.

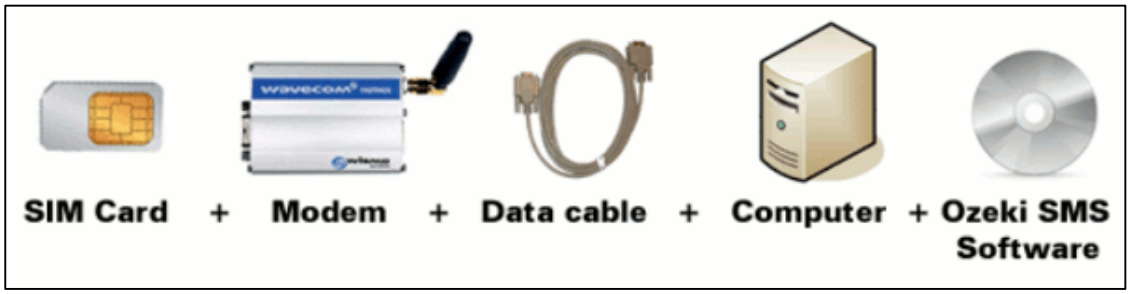


Figure 2.4: SMS connection components

2.3.6 Advantages of SMS

1) **An SMS is personal**: Unlike an e-mail, an SMS is much more likely to be read by a person at any one time, since the majority of people have their mobile phones at arms reach 24 hours a day. Of course the same also applies to a phone call [15].
2) **Messages are instantly recorded**: Unlike a phone call, an SMS message is automatically stored where it can be re-read. This proves particularly useful in the case of fairly detailed information that might otherwise be forgotten. Unlike e-mail, SMS as it currently stands is relatively SPAM free. Although this may change over the next year or so, at present it is the ideal communication channel to cut through the clutter. As a result, marketing departments worldwide are climbing aboard to try and target their customers in this one-on-one manner that has message opening rates soaring way above any other medium.
3) **An SMS is Discreet**: Unlike a phone call you still know when an SMS has arrived. The discreet nature of text messaging ensures you stay in touch with minimal disturbance.
4) **SMS leads to smaller phone bills:** An SMS is far cheaper than a phone call yet in most instances you will convey just as much information as you would have if you had called.

5) **SMS is cheaper**: when compared with voice messaging or web access, SMS is non-intrusive and hence messages will be obtained in a discrete fashion. SMS does not

disturb you if you are busy with an important work. SMS has high integration capabilities and so many software programs help you to give text alerts to cell phones if any urgent conditions exist. The advantages of SMS are cost effectiveness, comfort, opinion expression, and spontaneity. The main advantage of SMS is its inexpensiveness. When compared with an international call that costs you several dollars, by sending an SMS the same communication can be done with a significantly lesser amount. SMS is found as the most comfortable way to communicate [16].

2.3.7 SMS in Our system

In this system SMS will be used to send/receive messages over all the system. By sending a message which contains the violation data from vehicle to the central server, in addition to send alarm signal to the emergency center when crashing occurred.

On the other side, SMS modem will be used at the central server and emergency server to receive a messages, and to send the reply again.

2.4 Bluetooth technology

2.4.1 Introduction

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band (2400–2480 MHz) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security. Created by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

2.4.2 Bluetooth ranges

Bluetooth is a standard wire-replacement communications protocol primarily designed for low power consumption, with a short range (power-class-dependent, but effective ranges vary in practice; see table below) based on low-cost transceiver microchips in each device. Because the devices use a radio (broadcast) communications system, they do not have to be in visual line of sight of each other, however a quasi optical wireless path must be viable.

Table 2.1: Bluetooth Ranges

Class	Maximum permitted power		Range (m)
	(mW)	(dBm)	
Class 1	100	20	~100
Class 2	2.5	4	~10
Class 3	1	0	~1

2.4.3 Bluetooth in our system

In this system, bluetooth will be used to transfer control signals from android mobile phone to the Arduino microcontroller, which will be used later to activate or deactivate the mechanical system.

2.5 Sensors and other equipment needed in the system

2.5.1 Modems

1) GPS android Mobile/Tablet device

A GPS navigation device is any device that receives Global Positioning System (GPS) signals by using GPS receiver built in for the purpose of determining the device's current location on Earth. GPS devices provide latitude and longitude information, and some may also calculate altitude [17].

This device supported by android operating system that we want to use in order to support our project objectives.

Android is an open source platform and it is released under open source license. The Android operating system software stack consists of Java applications running on a Java based object oriented application framework. Libraries written in C, include the surface manager, Open Core media framework and SQlite relational database management system.

This device has a high speed performance that's good for running android operation system and for fast determining vehicle location by using SQLite data base application.

The project basically depends on the GPRS technology, and this technology needs data SIM card and a modem.

2) GPRS/GSM modem [18]

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable.

A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer.

Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. Computers and microcontrollers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands.

In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, you can do things like:

1) Reading, writing and deleting SMS messages.
2) Sending SMS messages.
3) Monitoring the signal strength.
4) Monitoring the charging status and charge level of the battery.
5) Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low (only about six to ten SMS messages per minute).

2.5.2 Microcontroller

Definition of Microcontroller

A Microcontroller is a small computer on single chip, used to control objects, processes and events. A Microcontroller is similar to the microprocessor, both Microcontroller and microprocessor contain processing unit, in different of microprocessor, the Microcontroller contain memory and programmable input/output peripherals on single chip [19].

The Microcontroller is very common component in modern electronic system; a

Microcontroller is device, which integrates a number of the component of a microprocessor system onto a single chip (IC).

Microcontroller are designed for embedded application which is essentially the whole circuit board, and can execute one instruction at a time , for this reason your program must have the ability to run while the Microcontroller is run .

Components of Microcontroller

The next figure show the most component of any microcontroller
CPU, memory, input/output peripherals, and bus

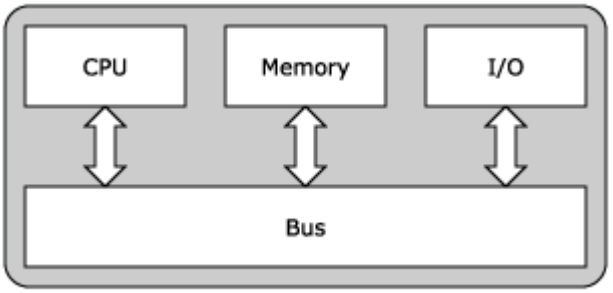


Figure 2.5: Main components of a microcontroller

Microcontrollers will also contain other devices such as, Timer module to allow the microcontroller to perform tasks times. Serial I/O port to allow data to flow between the microcontroller and other devices such as a PC or another microcontroller. ADC to allow the microcontroller to accept analogue input data for processing.

Microcontroller's building blocks explained

Memory Unit

Memory is part of the microcontroller used for data storage, it has two types: read only memory-ROM, and random access memory-RAM.

- Read only memory – ROM:

ROM used to permanently save the program being executed , microcontroller commonly use 16-bit addressing , which means that they are able to address up to 64kb of memory , ROM have many type ,such as erasable programmable Rom -EPROM, one time programmable Rom -OTPROM .

- Random access memory RAM:

RAM is used for temporary storing data and intermediate results created during the operation of the microcontroller.

Central Processing Unit CPU

CPU is a unit, which monitors and controls all processes inside the microcontrollers; it consists of subunits, such as instruction decoder, arithmetical logical unit -ALU, accumulator.

Input/output Ports

There are several types of ports , input , output ,or bidirectional ports , when work with ports , first of all it is necessary to choose which port we need to work with , and then to send data to , or take it from the port . When working with port, it acts like a memory locations.

Bus

Bus consists of 8 , 16 or more wire , there are two types of buses: the address bus, and the data bus. The address bus consists of as many lines as necessary for memory addressing. It is used to transmit the address from the CPU to the memory. The data bus is as wide as the data, in our case it is 8 bits or wires wide. It is used to connect all circuits inside the microcontroller.

Arduino Microcontroller

Arduino is an open source board microcontroller, Arduino is designed to make electronic more accessible to artists, designer, hobbyists and anyone interested in creating interactive objects or environments.

An Arduino board can be purchased pre-assembled, or because the hardware design is open source, built by hand. The hardware consists of a simple open hardware design for the

Arduino board with an Atmel AVR processor and on board input/output support. The software consists of a standard programming language compiler and the boot loader that run on the board.

Arduino hardware is programmed using a wiring-based language; the Arduino project received an honorary mention in the Digital Communities category at the 2006.

Arduino input/output peripherals

The Arduino Mega 2560 is based on the ATmega2560 microcontroller. It has 54 digital input/output pins, 14 pins of digital input/output pins can be used as PWM outputs, 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, USB connection, power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Figure (2.7), show main component on the Arduino board [20].

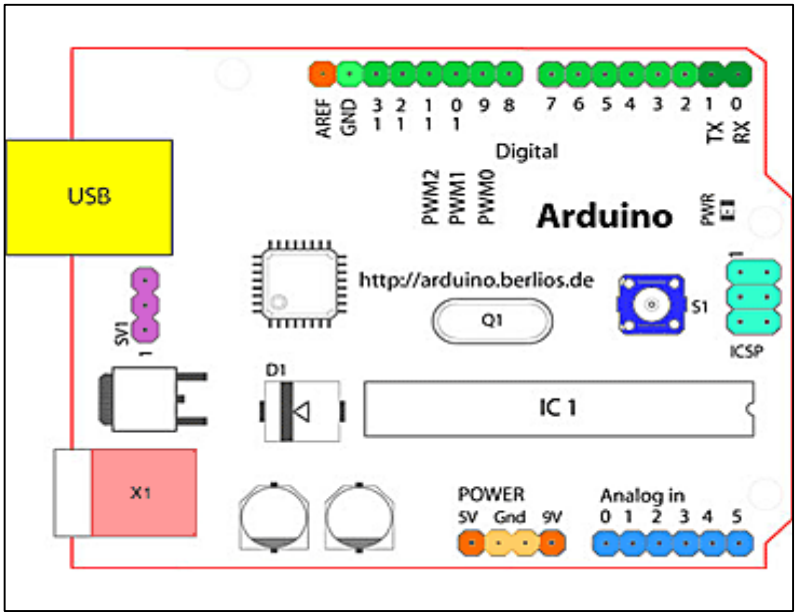


Figure 2.6: Arduino board components

Microcontroller in Our system

In this system, we will use Arduino mega as microcontroller and processing unit, which will be fixed inside vehicle , and programmed to be ready to send and receive data to/from GPS navigator tablet. In addition, it will be used to send/receive data to/from mechanical system.

Micro controller will be responsible for all computational process as in this system, such as making loops rules, and voltages amplitude conversion forms, to have suitable forms for any connected components.

2.5.3 Android Sensors [21]

Most Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data with high precision and accuracy, and are useful if you want to monitor three-dimensional device movement or positioning, or you want to monitor changes in the ambient environment near a device. For example, a game might track readings from a device's gravity sensor to infer complex user gestures and motions, such as tilt, shake, rotation, or swing. Likewise, a weather application might use a device's temperature sensor and humidity sensor to calculate and report the dew point, or a travel application might use the geomagnetic field sensor and accelerometer to report a compass bearing.

The Android sensor framework lets you access many types of sensors. Some of these sensors are hardware-based and some are software-based. Hardware-based sensors are physical components built into a handset or tablet device. They derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change. Software-based sensors are not physical devices, although they mimic hardware-based sensors. Software-based sensors derive their data from one or more of the hardware-based sensors and are sometimes called virtual sensors or synthetic sensors. The linear acceleration sensor and the gravity sensor are examples of software-based sensors. Table 1 summarizes the sensors that are supported by the Android platform.

Few Android-powered devices have every type of sensor. For example, most handset devices and tablets have an accelerometer and a magnetometer, but fewer devices have barometers or thermometers. In addition, a device can have more than one sensor of a given type.

The Accelerometer Sensor

It is a Motion detection (shake, tilt, etc.), which measures the acceleration force in (m/s^2) that is applied to a device on all three physical axes (x, y, and z), including the force of gravity [22].

The accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device. For example, an accelerometer at rest on the surface of the earth will measure an acceleration g = 9.81 m/s^2, straight upwards, due to its weight. By contrast, accelerometers in free fall or at rest in outer space will measure zero. Another term for the type of acceleration that accelerometers can measure is g-force acceleration.

Accelerometers have multiple applications in industry and science. Highly sensitive accelerometers are components of inertial navigation systems for aircraft and missiles. Accelerometers are used to detect and monitor vibration in rotating machinery. Accelerometers are used in tablet computers and digital cameras so that images on screens are always displayed upright [23].

Accelerometer sensor in our system

Our system depends on Accelerometers sensor to detect accidents .Once they happened, the accelerometers sensor listens to the mobile vibration then making an interrupt over the system, next the mobile/tablet sends the location data to the emergency server.

2.6 System software

2.6.1 Overview

Now days, computers are able to perform different tasks in different fields, but they can't do that by themselves! They need detailed procedures for each task. That what we called a computer program.

There are a lot of programming languages that are used to build out a computer program, choosing any language depends on the tasks of the program.

This section will introduce an overview about the software and programming methods which will be used in the system, includes software inside vehicle which includes Android programming language, and AT commands for GPRS modems, or at the server part which includes C# programming language, and database building.

2.6.2 Android programming language

Android is a Linux-based operating system designed primarily for touch screen mobile devices, such as smart phones and tablet computers, in much the same way that PCs run Microsoft Windows as their operating system. Android developed by Google in conjunction with the Open Handset Alliance. Initially developed by Android Inc, whom Google financially backed and later purchased in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance [24].

Android was built from the ground-up to enable developers to create compelling mobile applications that take full advantage of all a handset has to offer. It was built to be truly open. Furthermore, it utilizes a custom virtual machine that was designed to optimize memory and hardware resources in a mobile environment. Android is open source; it can be liberally extended to incorporate new cutting edge technologies as they emerge. The platform will continue to evolve as the developer community works together to build innovative mobile applications.

Millions of people already use Android because it makes your mobile device so much more powerful and useful. On Android, the home screen, web browser, email and everything in between are designed to make your life easier. And because Android is open, you can create a unique mobile experience that's just right for you.

Android in our project

In this system, Android will be used as an operating system for the GPS navigator, and after completing this system design, we expect to have a special android application, which will be able to receive data from GPS, compare it with stored database, and then determine if the driver commit a violation or not. Using Android programming language, a suitable user interface design will be developed.

2.6.3 Mobile Database

Definition

A mobile database is a database that can be connected by a mobile computing device over a mobile network. The client and server have wireless connections. A cache is maintained to hold frequent data and transactions so that they are not lost due to connection failure. A database is a structured way to organize information. This could be a list of contacts, price information or distance travelled [4] .

The use of laptops, mobiles and PDAs is increasing and likely to increase in the future, with more and more applications residing in the mobile systems. While those same analysts can't tell us exactly which applications will be the most popular, it is clear that a large percentage will require the use of a database of some sort. Many applications such as databases would require the ability to download information from an information repository and operate on this information even when out of range or disconnected.

Need for mobile databases

- Mobile users must be able to work without a wireless connection due to poor or even non-existent connections.
- Applications must provide significant interactivity.
- Applications must be able to access local device/vehicle hardware, such as printers, bar code scanners, or GPS units (for mapping or Automatic Vehicle Location systems).
- Bandwidth must be conserved (a common requirement on wireless networks that charge per megabyte or data transferred).
- Users don't require access to truly live data, only recently modified data.
- Limited life of power supply (battery).
- The changing topology of network.

Three parties of mobile database

As shown in figure (2.7), mobile databases typically involve three parties: fixed hosts, mobile units, and base stations. Fixed hosts perform the transaction and data management functions with the help of database servers. Mobile units are portable computers that move around a geographical region that includes the cellular network (or "cells") that these units use to communicate to base stations. Base stations are two-way radios, installations in fixed locations that pass communications with the mobile units to and from the fixed hosts. They are typically low-power devices such as mobile phones, portable phones, or wireless routers.

When a mobile unit leaves a cell serviced by a particular base station, that station transparently transfers the responsibility for the mobile unit's transaction and data support to whichever base station covers the mobile unit's new location.

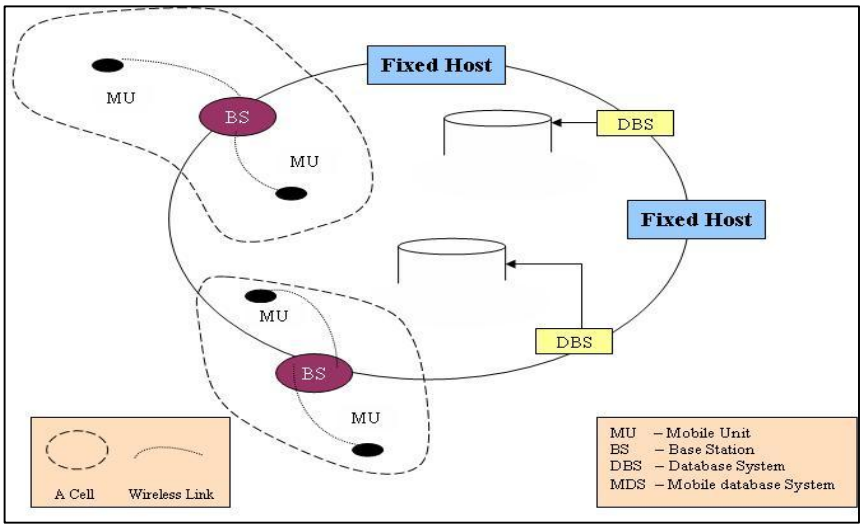


Figure 2.7: *Parties of mobile database*

2.6.4 SQlite Database

Definition

SQlite is an open source embedded relational database. Originally released in 2000, it was designed to provide a convenient way for applications to manage data without the overhead that often comes with dedicated relational database management systems. SQlite has a reputation for being highly portable, easy to use, compact, efficient, and reliable.

One advantage of having a database server inside your program is that no network configuration or administration is required. Both client and server run together in the same process. This reduces overhead related to network calls, simplifies database administration, and makes it easier to deploy your application. Everything you need is compiled right into your program.

SQlite Benefits

SQlite is quite versatile. It is a database, a programming library, and a command-line tool, as well an excellent learning tool that introduces relational databases. There are indeed

many ways to use it in embedded environments, websites, operating system services, scripts, and applications. For programmers, SQlite is like digital duct tape, providing an easy way to bind applications and their data. Like duct tape, there is no end to its potential uses. In a web environment, SQlite can help with managing complex session information. Rather than serializing session data into one big blob, individual pieces can be selectively written to and read from individual session databases. SQlite also serves as a good stand-in relational database for development and testing: there are no external networking to configure, or usernames and passwords to bother with. SQlite might also serve as a cache, hold configuration data, or because of its binary compatibility across platforms, even work as an application file format.

SQlite in Our System

In this System, SQlite will be used as an in-vehicle database, which will contain all streets GPS coordinates, Streets names, and streets speed limits.

This database will be used later to compare it with the data taken from GPS receiver to determine the current location of the vehicle and the road speed limit, which will be used to determine if the driver commit a violation or not .

2.6.5 C Sharp and Microsoft SQL Server [25]

Overview

C# (Pronounced "C sharp"), is as Microsoft simple, modern, object-oriented, and type-safe programming language derived from C and C++." Targeted at the web development community, where Java is the dominant language, C# is available as part of Microsoft's visual studio. NET. In addition to C#, visual studio .NET also supports Visual Basic, Visual C++, and Jscript [26].

C# is a new programming language from Microsoft designed specifically to target the .NET Framework. Microsoft's .NET Framework is a runtime environment and class library that dramatically simplifies the development and deployment of modern, component-based appellations [27].

C# was designed by Anders Hejlsberg (creator of Turbo Pascal and architect of Delphi), Scott Wiltamuth, and Peter Golde. Described in the C# Language Specification as a "simple, modern, object-oriented, and type-safe programming language derived from C and C++, C# bears many syntactic similarities to C++ and Java .

C# has no runtime library of its own. instead, C# relies on the Vast class library in the .NET Framework for all its needs, including console I/O, network and file handling , collection data structure, and many other facilities .

The main architecture of C# can be described in:

1) **Common Language Runtime (CLR)**

CLR is a modern runtime environment that manages the execution of user code, providing services such as JIT compilation, memory management, exception management, debugging and profiling and support, and integrated security and permission management.

2) **Compiler**

.NET compiler function do not target a spesific native processor. Instead, they consume source files and produce binary files containing an intermediate representation of the source constructs, expressed as acombination of metadata and Common Intermediate Language (CIL). In order for these binaries to be executed, the CLR must be present on the target machine.

Microsoft intermediate language shares with java byte code the idea that it is a low level-language with a simple syntax, which can be very quickly translated into native machine code. This also has significant advantages:

- Platform independence: mean that the same file containing byte code instruction can be placed on any platform, so by compiling to IL you obtain platform independence for .NET.
- Performance improvement: instead of compiling the entire application in one go, the JIT compiler simply compiles each portion of code as it is called, so when code has been compiled once, the resultant native executable is stored until the application exits so that it does not need to be recompiled the next time that portion of code is run.
- Language interoperability: the use of IL facilitates language interoperability, simply put you can compile to IL from one language, and this compiled code should then be interoperable with code that has been compiled to IL from another language. [16].

2.6.6 Microsoft SQL Server

Overview

Microsoft SQL Server is a relational database that runs on the NT operating system. SQL, or Structured Query Language, is a widely accepted industry standard for defining, changing, and managing data and controlling how changes to the database are made by using tables, indexes, keys, rows, and columns to store data. SQL was developed from the ideas of Dr. Edgar (Ted) F. Codd of International Business Machines (IBM) who helped develop the relational model while working at IBM's research labs in the 1970s.

Data base is a software product whose primary function is to store and retrieve data as requested by other software applications, those can be on the same computer or those running on another computer across a network (including the Internet). There are at least many different editions of Microsoft SQL Server aimed at different audiences and for different workloads, ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time.

SQL Server is Microsoft high-end client-server data base and is closely integrated with Microsoft visual studio and the Microsoft office system, SQL Server support symmetric multiprocessing hardware, and major open standard communication protocols.

Architecture of MS SQL Server

Microsoft SQL Server contains different layers and services [28]:

- Protocol Layer

Protocol layer implement the external interface to SQL Server, any operation invoked on SQL Server are communicated to it via a Microsoft defined format, called tabular data stream, which is an application layer protocol, used to transfer data between a data base and client.

- Data Storage

Database is the main unit of data storage; it is a collection of tables with type columns. SQL Server supports different data type, including primary types such as integer, character string.

- Buffer Management

SQL Server buffer pages in memory-RAM to minimize disc input/output, any page been buffered in memory and the set of all buffered on memory called buffer cache, the buffer cache is manage by buffer manager. Any operation reading from or writing to any page copies it to the buffer cache, Subsequent reads or writes are redirected to the memory copy , rather than the on disc version .The page is update on the disc by the buffer manager only if the in-memory cache has not been referenced for same time .

- Logging and Transactions

SQL Server ensures that any change to the data is ACID-compliant, it uses transactions to ensure that the data base well revert to a known consistent state.SQL Server implement transactions using a write-ahead log.

- .Concurrency and locking

SQL Server allow multiple clients to use the same data base concurrently , it needs to control concurrent to shared data , to ensure data integrity , SQL Server controls concurrent access using locks , locks can be applied on different levels of granularity , or entire tables ,page ,or even on a per row basis on tables .

Part B: Mechanical Part

2.7 Power train control module

The PCM controls operation of all of the engine and transmission outputs. It is on a bracket attached to floor under the centre console. The data link connector is the access point for the scan tool. The connector is under the instrument panel at the left side of the steering column. It is a multi-pin connector with integral mounting bracket [28].

Signal inputs used by the PCM include:

- Coolant temperature.
- Intake air temperature.
- Boost/baro pressure.
- Optical/fuel temperature sensor.
- Crankshaft position sensor.
- Accelerator pedal position sensors (APP).
- Cruise control.
- A/C request.
- Vehicle speed sensor.
- Automatic transmission fluid pressure manual valve position switch.
- Transmission input speed sensor.
- Transfer case low range switch.
- Transmission fluid temperature sensor.

2.8 Accelerator Pedal Position Sensor (APPS)

2.8.1 Accelerator Pedal and (APPS)

The accelerator pedal is a device, used in many types of vehicles, that allows an operator to modulate engine power remotely. It is generally paired with a brake pedal, and sometimes a

clutch, enabling a driver to control the speed of the vehicle almost exclusively with his feet. An accelerator pedal is typically connected to a throttle directly, either by cables or, electronically, to a computer that mechanically adjusts the throttle based on pedal input.

The APP sensor module contains two potentiometers (a device for measuring unknown voltage or potential difference by comparison to a standard voltage). Each of the APP sensors sends a varying voltage to the ECM. By monitoring the output voltage from the Accelerator Pedal Position (APP) module, the ECM can determine fuel delivery based on the accelerator pedal position (driver demand) [29].

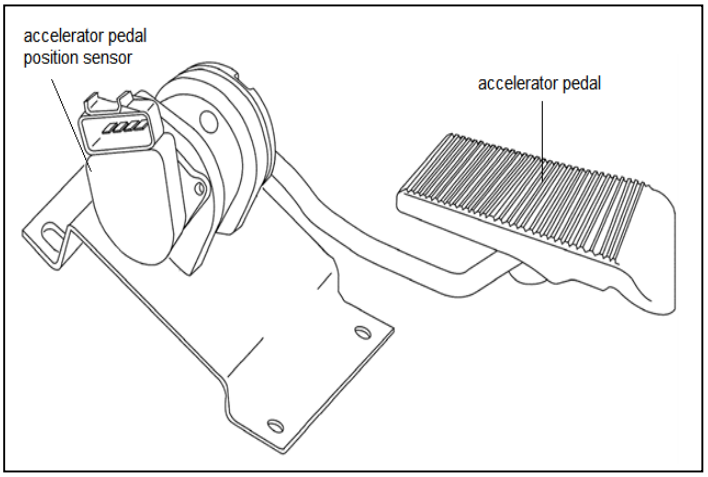


Figure 2.8: *Accelerator pedal and APP sensor*

2.8.2 Types of APPS

There are two commonly used types of position sensor, Potentiometer and Hall Effect.

Potentiometer type's sensors comprise of an arc resistive material in two separate tracks with a wiper arm which slides across the track producing a progressive change in resistance, this signal is sent to the ECU which calculates the angle of the component the sensor is attached to. In some instances one of the two sensor tracks is used as a redundant sensor which produces half the voltage of the first, this sensor is used as a second independent signal which is used for trouble shooting and also provides a more accurate reading.

The Hall Effect angle of rotation sensors use a semi-circular magnet disc which generates a magnetic flux, this flux is returned to a Hall Effect sensor via soft conductive elements and a shaft. As the magnetic disc rotates the change in flux is picked up by the Hall Effect sensor, this information is used by the ECU to calculate the angle of movement.

2.8.3 Construction

In this application a contact-less sensor is used, that is based on the inductive principle. This sensor consists of a stator, that contains a field coil, reception coils as well as an electronic for evaluation (Fig) and a rotor that is be formed from one or more closed conducting loops with a definite geometry [30].

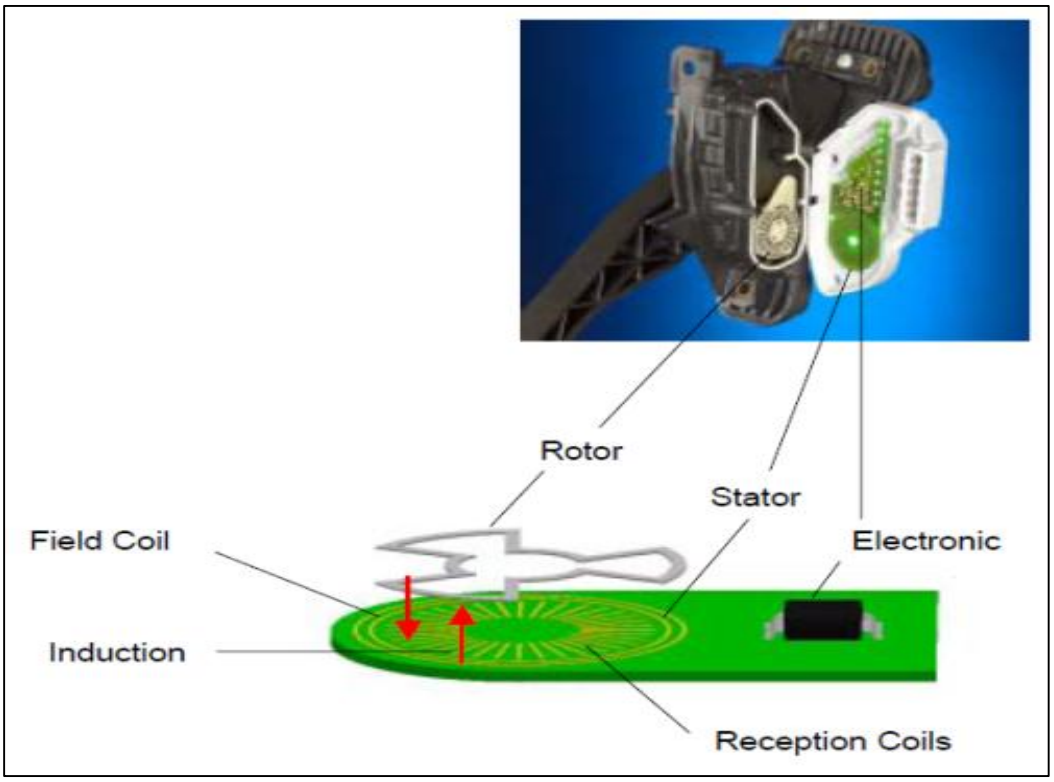


Figure 2.9: *Inductive APP sensor construction*

2.8.4 Function

By supplying an alternating voltage to the field coil, a magnetic field is produced, which induces voltages in the reception windings. A current is also induced in the conduction loops of the rotor, which influences the exciter field.

Dependent upon the position of the rotor relative to the stator windings, A/C amplitudes are produced. These are rectified in the electronic rectification unit and sent as a D/C voltage to the control unit. The evaluated signal is sent to the throttle position motor as a pulsed signal. The characteristics of this signal is dependent upon the attitude of the accelerator pedal.

2.8.5 Consequence of failure

If the accelerator pedal sensor fails it can cause the following fault symptoms:

- Engine has only a slightly position idle position
- Engine doesn't react to movement of the accelerator pedal
- Vehicle goes into emergency running mode
- Malfunction indicator lamp illuminates

A failure can have different causes:

- Damaged wire or connection on the an accelerator pedal sensor
- Faulty earth / ground // faulty voltages supply
- Defective electronic rectification in the sensor
- Mechanical defect

2.8.6 Output and input signal representation

1. Output Signal from PCM representation: with this measurement the voltage supply of the sensor is checked. Ignition on / off.

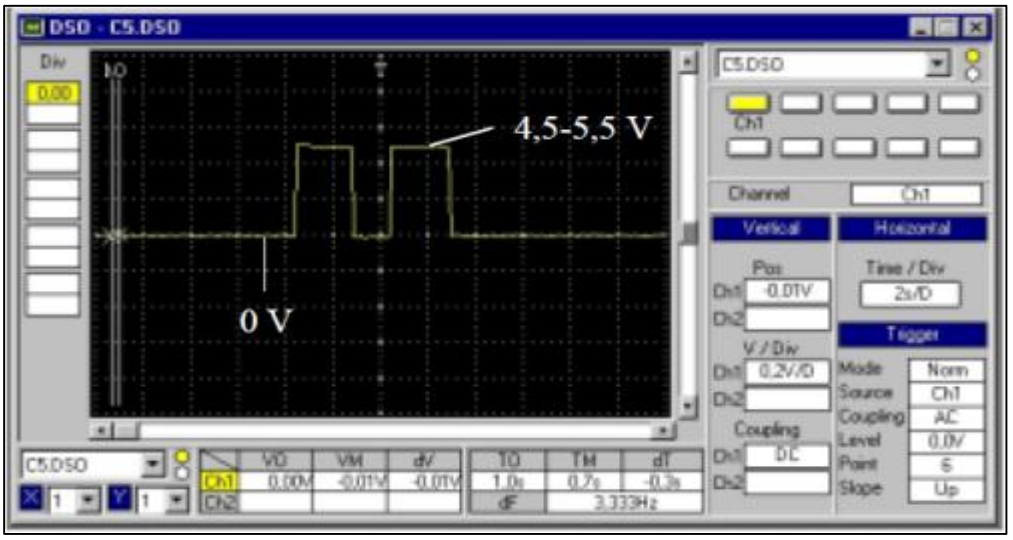


Figure 2.10: *Output Signal from PCM*

2. Input Signal to PCM representation (Ignition on, pressing pedal and releasing): the increase and decrease of the signal are dependent on the speed, with which the pedal is pressed and released. [31].

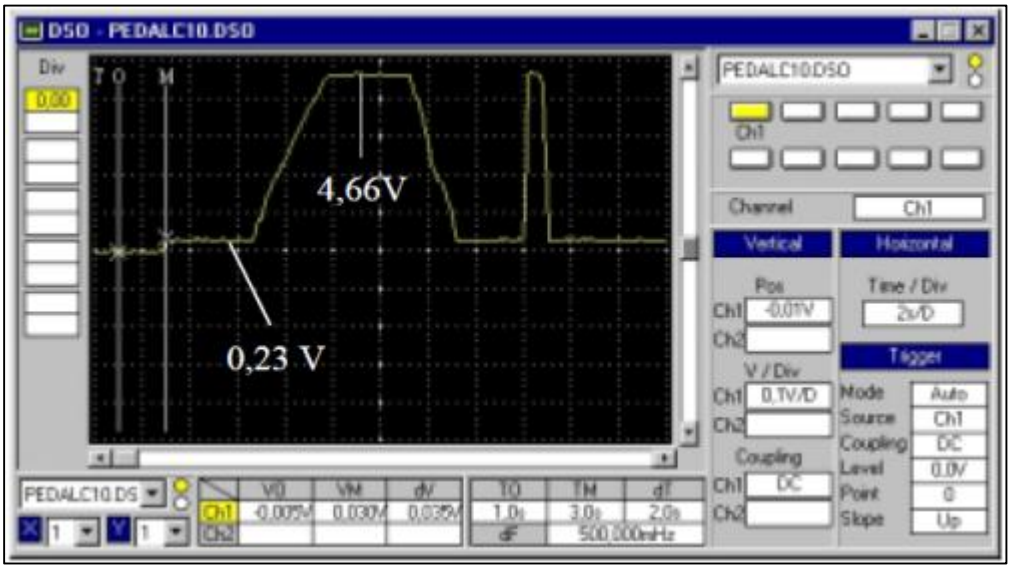


Figure 2.11: *Input Signal to PCM representation (Ignition on, pressing pedal and releasing)*

2.8.7 APPS in gasoline and diesel engine

Beginning with the first engine vehicles, there has always been a need for the driver to adjust engine output in order to control speed. In gasoline-powered internal combustion engines, the accelerator pedal adjusts the amount of air allowed into the combustion chamber, with the corresponding supply of fuel being regulated by a carburetor or fuel injection. In early designs, the pedal itself was directly tied to a butterfly valve, located either in the carburetor itself or the throttle body that could let more or less air in.

Many modern engines use a drive by wire system, in which there is no direct physical connection between the pedal and the throttle. Rather pedal pressure is translated by a computer, which regulates air intake in response to driver input, while maximizing efficiency. Detractors of this design claim the driver loses a degree of control when a computer is introduced into the equation, but car manufacturers contend technology has reached the point where there is no loss of what some people refer to as driver feel [32].

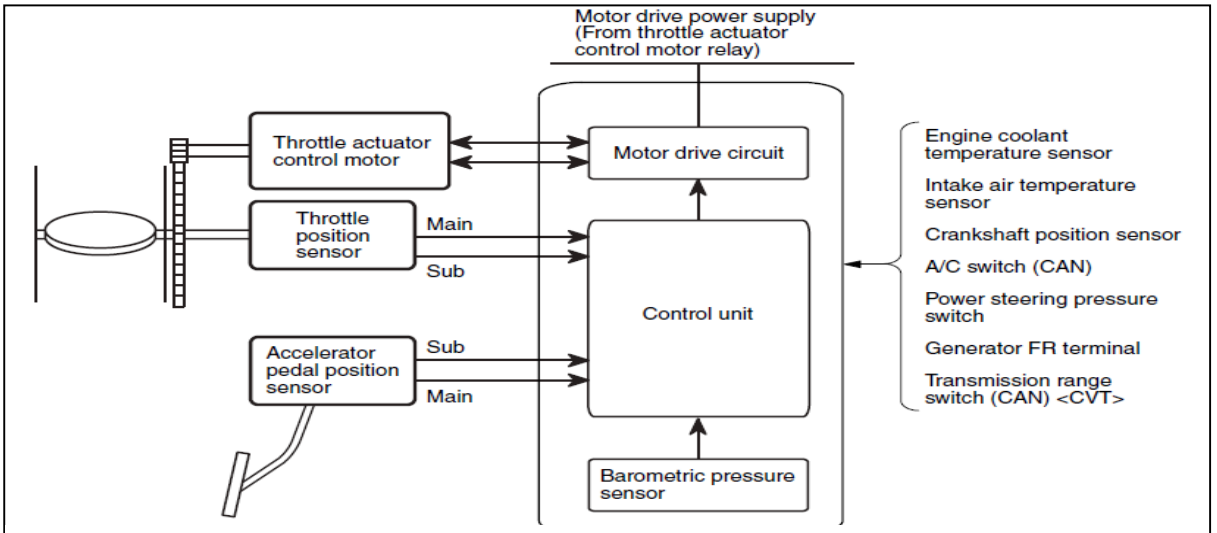


Figure 2.12: *The APP sensor in gasoline engine*

The accelerator pedal of a diesel engine functions differently. Instead of controlling the flow of air, it adjusts the amount of fuel entering the combustion chamber. In a diesel engine, it is the compression of the fuel that causes it to ignite, as opposed to the introduction of air.

Therefore there is no actual throttle. For the driver, however, the effect of pushing the pedal down is the same.

2.8.8 APPS in this system

In this system the first step of reduce vehicle speed is cut the output voltage from two APPS potentiometers.

2.9 Traction Control System

A traction control system (TCS), also known as anti-slip regulation (ASR), is typically (but not necessarily) a secondary function of the anti-lock braking system (ABS) on production motor vehicles, designed to prevent loss of traction of driven road wheels. When invoked it therefore enhances driver control as throttle input applied is miss-matched to road surface conditions (due to varying factors) being unable to manage applied torque.

Intervention consists of one or more of the following:

1) Reduces or suppress spark sequence to one or more cylinders
2) Reduce fuel supply to one or more cylinders
3) Brake force applied at one or more wheels
4) Close the throttle, if the vehicle is fitted with drive by wire throttle
5) In turbo-charged vehicles, a boost control solenoid can be actuated to reduce boost and therefore engine power.

Typically, traction control systems share the electro-hydraulic brake actuator (but does not use the conventional master cylinder and servo), and wheel speed sensors with the anti-lock braking system.

Operation

When the traction control computer (often incorporated into another control unit, like the anti-lock braking system module) detects one or more driven wheels spinning significantly faster than another, it invokes ABS ecu to apply brake friction to wheels spinning with lessened traction. Braking action on slipping wheel(s) will cause power transfer to wheel axle(s) with

traction due to the mechanical action within a differential. All-wheel drive AWD vehicles often have an electronically controlled coupling system in the transfer case or transaxle engaged (active part-time AWD), or locked-up tighter (in a true full-time set up driving all wheels with some power all the time) to supply non-slipping wheels with (more) torque.

Use of traction control

In road cars: Traction control has traditionally been a safety feature in premium high-performance cars, which otherwise need sensitive throttle input preventing spinning driven wheels when accelerating, especially in wet, icy or snowy conditions. In recent years, traction control systems have become widely available in non-performance cars, minivans, and light trucks.

In race cars: Traction control is used as a performance enhancement, allowing maximum traction under acceleration without wheel spin. When accelerating out of turn, it keeps the tires at optimal slip ratio.

In off road vehicles: Traction control is used instead or in addition to the mechanical limited slip or locking differential. It is often implemented with an electronic limited slip differential, as well as other computerized controls of the engine and transmission. The spinning wheel is slowed down with short applications of brakes, diverting more torque to the non-spinning wheel; this is the system adopted by Range Rover models in the mid 1990's, for example. This form of traction control has an advantage over a locking differential, as steering and control of a vehicle is easier, so the system can be continuously enabled. It creates less stress on drive-trains, particularly important to vehicles with an independent suspension, generally weaker compared to solid axles.[citation needed] On the other hand, only half of the available torque will be applied to a wheel with traction, compared to a locked differential, and handling is less predictable.

Traction control in cornering

Traction control is not just used for improving acceleration under slippery conditions. It can also help a driver to corner more safely. If too much throttle is applied during cornering, the drive wheels will lose traction and slide sideways. This occurs as understeer in front wheel drive vehicles and over steer in rear wheel drive vehicles. Traction control can prevent this from happening by limiting power to the wheels. It cannot increase the limits of grip available and is used only to decrease the effect of driver error or compensate for a driver's inability to react quickly enough to wheel slip.

2.10 Anti-lock braking system

An anti-lock braking system (ABS) is a safety system that allows the wheels on a motor vehicle to continue interacting attractively with the road surface as directed by driver steering inputs while braking, preventing the wheels from locking up (that is, ceasing rotation) and therefore avoiding skidding.

Since initial widespread use in production cars, anti-lock braking systems have evolved considerably. Recent versions not only prevent wheel lock under braking, but also electronically control the front-to-rear brake bias. This function, depending on its specific capabilities and implementation, is known as electronic brake force distribution (EBD), traction control system, emergency brake assist, or electronic stability control (ESC).

ABS Operation

The anti-lock brake controller is also known as the CAB (Controller Anti-lock Brake).A typical ABS includes a central electronic control unit (ECU), four wheel speed sensors, and at least two hydraulic valves within the brake hydraulics. The ECU constantly monitors the rotational speed of each wheel; if it detects a wheel rotating significantly slower than the others, a condition indicative of impending wheel lock, it actuates the valves to reduce hydraulic pressure to the brake at the affected wheel, thus reducing the braking force on that wheel; the wheel then turns faster. Conversely, if the ECU detects a wheel turning significantly faster than the others, brake hydraulic pressure to the wheel is increased so the braking force is reapplied,

slowing down the wheel. This process is repeated continuously and can be detected by the driver via brake pedal pulsation. Some anti-lock system can apply or release braking pressure 16 times per second.

The ECU is programmed to disregard differences in wheel rotate speed below a critical threshold, because when the car is turning, the two wheels towards the centre of the curve turn slower than the outer two. For this same reason, a differential is used in virtually all road going vehicles.

If a fault develops in any part of the ABS, a warning light will usually be illuminated on the vehicle instrument panel, and the ABS will be disabled until the fault is rectified.

The modern ABS applies individual brake pressure to all four wheels through a control system of hub-mounted sensors and a dedicated micro-controller. ABS is offered or comes standard on most road vehicles produced today and is the foundation for ESC systems, which are rapidly increasing in popularity due to the vast reduction in price of vehicle electronics over the years.

Modern electronic stability control (ESC or ESP) systems are an evolution of the ABS concept. Here, a minimum of two additional sensors are added to help the system work: these are a steering wheel angle sensor, and a gyroscopic sensor. The theory of operation is simple: when the gyroscopic sensor detects that the direction taken by the car does not coincide with what the steering wheel sensor reports, the ESC software will break the necessary individual wheel(s) (up to three with the most sophisticated systems), so that the vehicle goes the way the driver intends. The steering wheel sensor also helps in the operation of Cornering Brake Control (CBC), since this will tell the ABS that wheels on the inside of the curve should brake more than wheels on the outside, and by how much.

2.11 Electronic Brake force distribution

Electronic brake force distribution (EBD or EBFD), Electronic brake force limitation (EBL) is an automobile brake technology that automatically varies the amount of force applied

to each of a vehicle's brakes, based on road conditions, speed, loading, etc. Always coupled with anti-lock braking systems, EBD can apply more or less braking pressure to each wheel in order to maximize stopping power whilst maintaining vehicular control. Typically, the front end carries the most weight and EBD distributes less braking pressure to the rear brakes so the rear brakes do not lock up and cause a skid. In some systems, EBD distributes more braking pressure at the rear brakes during initial brake application before the effects of weight transfer become apparent.

2.12 Brake Assist

Basically, a brake assist system monitors the driver's use of the brake pedal, automatically sensing an attempt to stop the car as a result of panic. It then generates very high braking power, even when the driver is only pressing lightly on the brake pedal. When this is used together with anti-lock braking systems, it results in faster and safer braking.

2.12.1 Brake actuator

In order to reduce the speed of the car, the brakes should be activated, to a chive this goal the brake actuator will be used, by pulling the brake pedal when it receive signal from the controller.

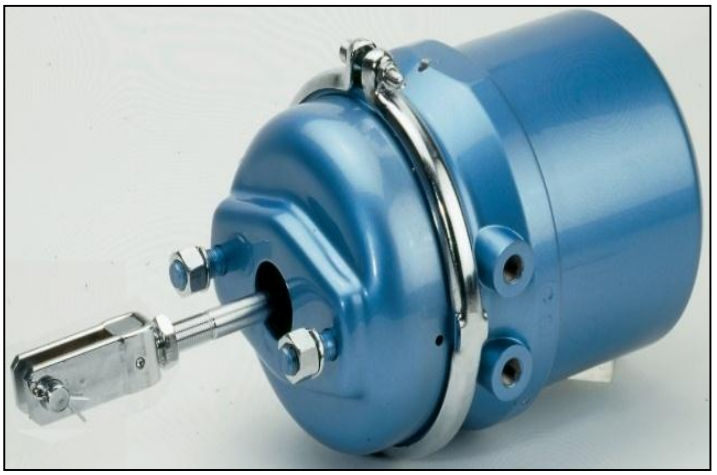


Figure 2.13: *Brake actuator*

3

Chapter Three

Conceptual Design

Chapter 3:

Conceptual Design

3.1 Introduction

This chapter will describe the system main parts and the design concepts in some details; it will talk about system general block diagram, the system main components, system main flow chart, system data flow diagram, and the system main functions.

3.2 General system block diagram

As shown in figure (3.1) the general system block diagram; the main parts of the system are: GPS receiver, where data and locations coordinates will be created, GSM network which is the connection medium through which data will be transferred, Central server where the data will be received and processed and mechanical system where the over limited vehicle speed will return to the speed street limit.

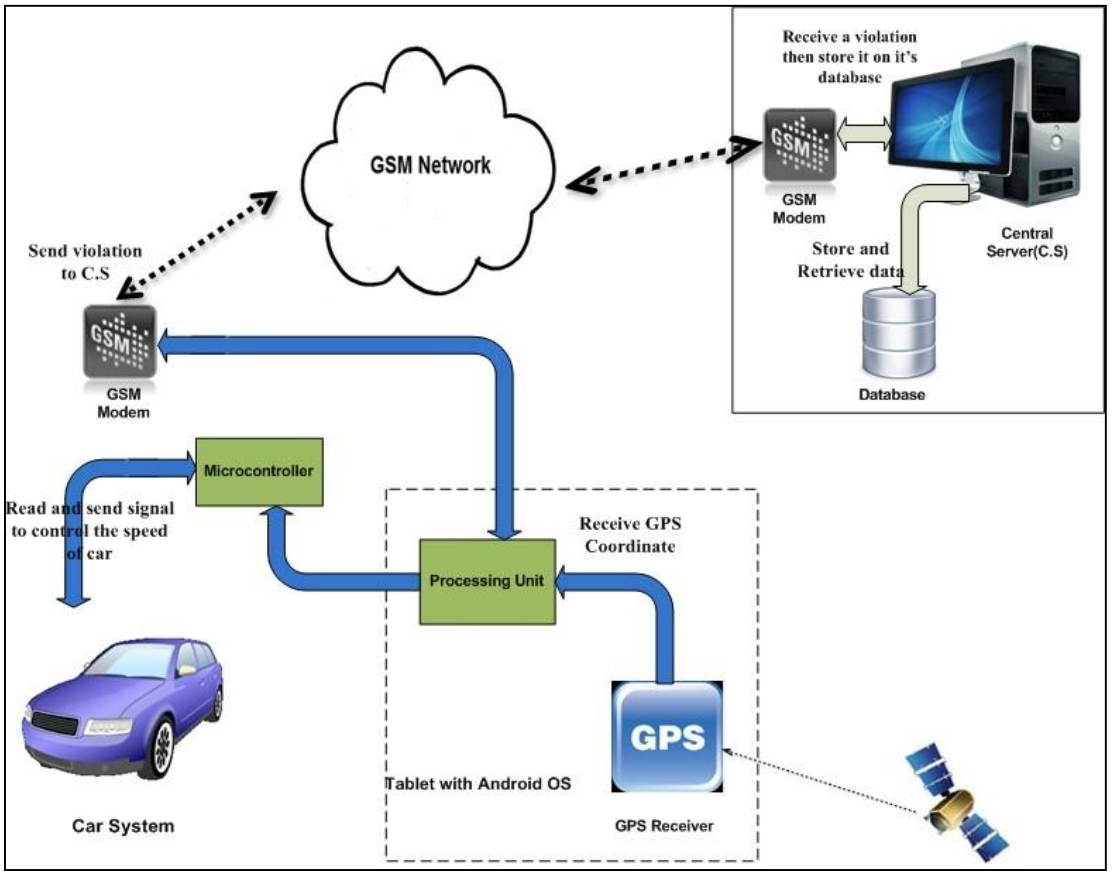


Figure 3.1:*The General System Design*

3.3 System main components

The main components of the system are the GPS Navigator (Samsung galaxy phone), microcontroller, GSM modem and Central server, as shown in Figure (3.2). This section will describe all of them in more details, and the options for these components will be described.

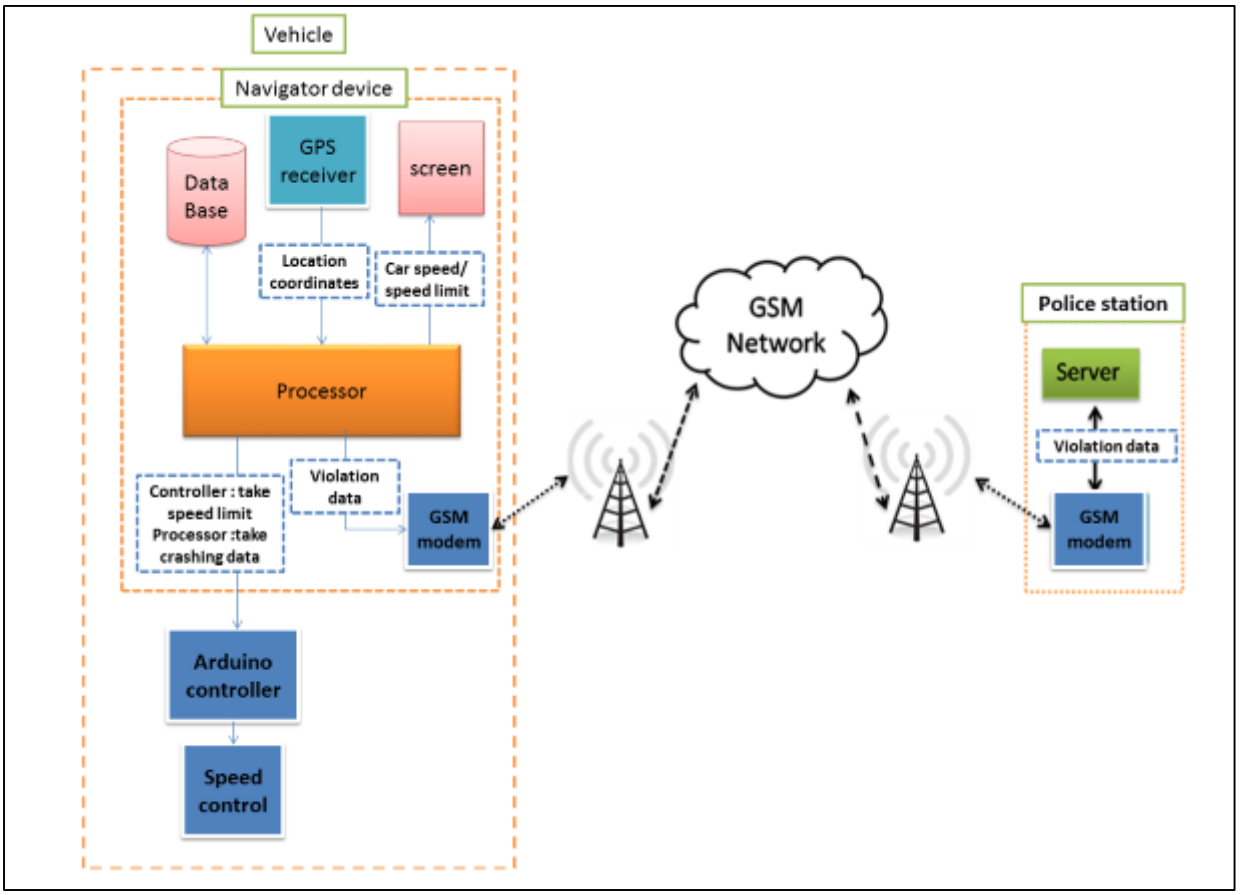


Figure 3.2:*System main component*

3.3.1 GPS Receiver

Overview

GPS receivers use signals from satellites which orbit the Earth at known positions. Each satellite has a unique identification code and sends a signal which the GPS receiver software can use to calculate the distance from the device to those satellites.GPS receivers have many types:

Type 1: Not-self-contained receivers (without screen), also known as RS232 receivers or also GPS mice. This type needs a computer (often a Pocket-PC or Palm PDA) and according to the

program in order to visualize the actual position of the GPS receiver. The link between the GPS and the computer can be wireless (Bluetooth), by means of a cable or via a card-slot or sleeve.

Type 2: Self-contained receivers (with screen). Here the computer is integrated in the GPS receiver. It is a mapping device which can be handheld or mounted in boat, car or plane.

Type 3: tablets, PDA's, and Mobile Phones with incorporated GPS receiver built in. these devices used for in-car navigation system. This type is different from the others as it include a built in GPS receiver.

3.3.2 Galaxy S1 Mobile phone

The system will use the Galaxy S1 mobile phone, because the cost and complexity of using separated GPS receiver, screen and microcontroller devices[35].

Figure 3.3:*Galaxy S1*

The following table shows the main specification of this device:

Table 3.1:*Galaxy S1 Mobile phone Specifications[36]*

Specifications	
CPU	PowerVR SGX540, 1 GHz Cortex-A8
OS	Android OS, v2.1 (Eclair), upgradable to v2.3 (Gingerbread)
General	2G Network GSM 850 / 900 / 1800 / 1900 3Gnetwork HSDPA 900 / 1900 / 2100
Speed	HSDPA, 7.2 Mbps; HSUPA, 5.76 Mbps
SIM Card	Mini Sim

Connectivity	GPRS/EDGE/WIFI/Bluetooth/GPRS
Memory	8/16 GB storage, 512 MB RAM, 2 GB ROM
WLAN	Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot (Android 2.2)
3G	Built in, Support phone calls and 3G Network
GPS	Support
Data Transfer	USB cable / Bluetooth
I/O Port	1*Mini USB port, 1*Headphone, 1*DC-IN JACK, 1*TF card reader, 1*HDMI, 2*SIM card slot
Audio	3.5mm headphone jack
	High-quality stereo loud speaker
	Built-in microphone

3.3.3 GSM/GPRS Modem

A GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone.

When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages.

A GSM modem exposes an interface that allows applications to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. To perform these tasks, a GSM modem must support an “extended AT command set” for sending/receiving SMS messages.

A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, A GSM modem could also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on the computer.

GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. The MODEM is the soul of such modules.

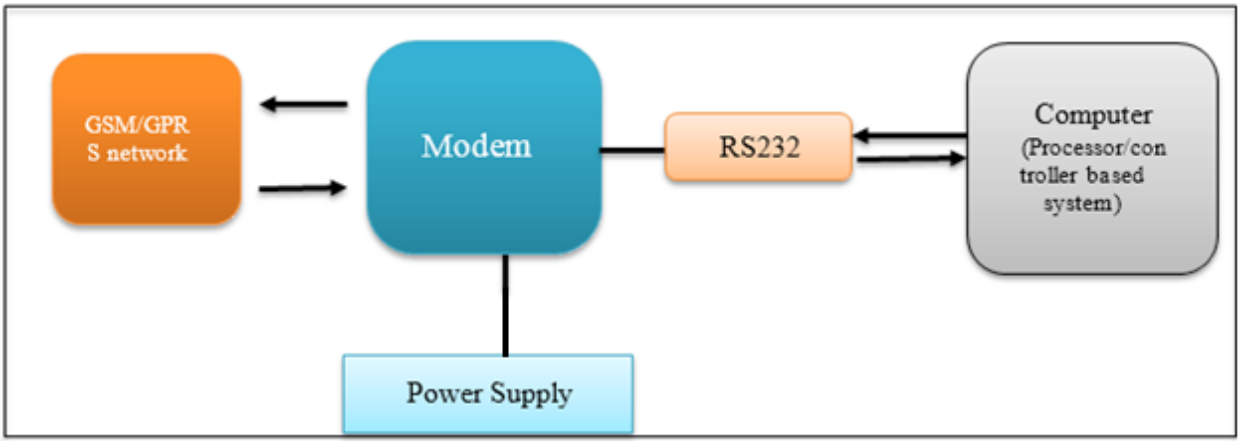


Figure 3.4:*GSM/GPRS module*

3.3.4 Wireless Modems

Wireless modems are the modem devices that generate, transmit or decode data from a cellular network, for establishing communication between the cellular network and the computer. These are manufactured for specific cellular network (GSM/UMTS/CDMA) or specific cellular data standard (GSM/UMTS/GPRS/EDGE/HSDPA) or technology (GPS/SIM). Wireless modems like other modem devices use serial communication to interface with and need compatible AT commands for communication with the computer (any microprocessor or microcontroller system)[37].

3.3.5 Telecom ZTE MF 180 USB GSM/GPRS modem[38]

In this system, and at the server side, the ZTE MF 180 USB GSM/GPRSmodem will be used to send/receive messages:

Table 3.2: *ZTE MF 180 USB GSM/GPRS modem Specifications*

Specification	Description
Network and Band compatibility	HSDPA/UMTS 850, 2100 MHz GSM/EDGE 850, 900, 1800, 1900 MHz
Size and Weight	75 x 26 x 10mm. Approx 28g
Data Rate	HSDPA Mode up to 3.6 Mbps
	UMTS Mode up to 384 Kbps
	EDGE Mode up to 236.8 Kbps
Power requirements	5V with current 100mA – 450mA max
Interface	USB 2.0
Operating systems	Windows 7, Vista & XP SP2 & SP3 Apple Mac OS X 10.4.11 and above
Functions	High speed wireless data, internet, MicroSDHC™card, SMS
Memory card	2GB MicroSD™up to 32GB MicroSDHC™

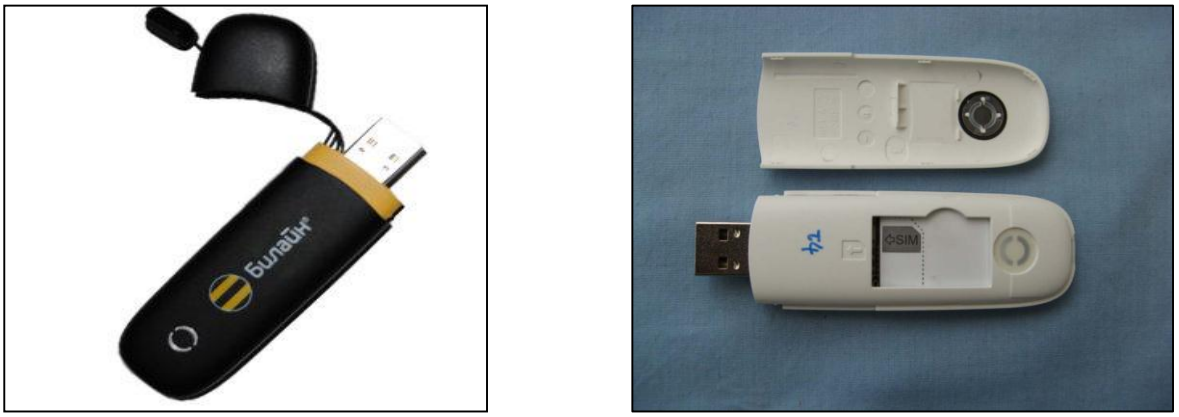


Figure 3.5:*ZTE MF 180 USB GSM/GPRS modem*

3.3.6 Arduino Microcontroller

Overview

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino hardware is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment.

3.3.7 The Arduino Mega 2560

Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 . It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), anddigitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions.

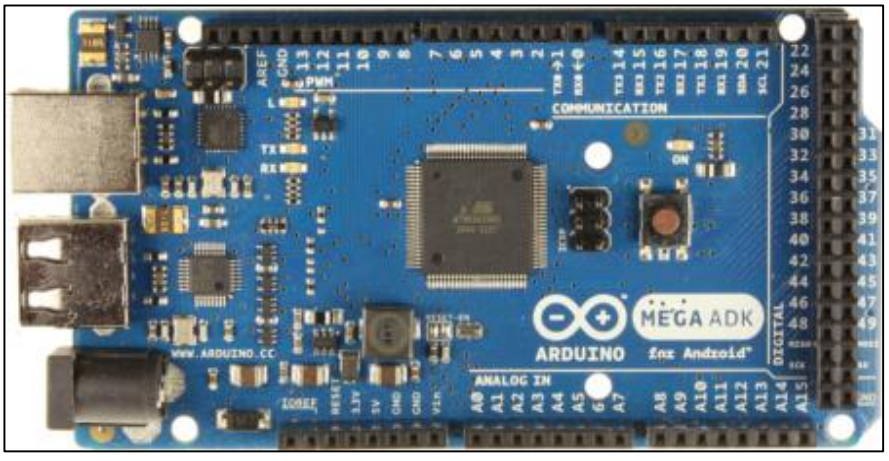
Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. AnATmega16U2 on the board channels one of these over USB and provides a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer .

Programming

The Arduino Mega can be programmed with the Arduino software.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on the Mega2560 and Duemilanove / Diecimila.



3.3.8 Why Arduino ATmega2560

1) It has a USB host interface to connect with Android based phones, and a power supply boost converter to charge up the phone from DC power, while its plugged into the ADK .
2) The Mega ADK is compatible with all of Android's Accessory Development Kit.
3) It is easy to connect any sort of sensors, LED, motor, etc. to your Android based phones.
4) Easy programming it with the Arduino software.

3.4 ITWAVs working principle

ITWAVs is a generic term for an in-vehicle technology system, which assists drivers to keep to, or below the sign-posted speed limit on public roads at all times.

By using GPS technology and on-board maps, which are linked to a speed zone database, ITWAV system knows where the vehicle located, and what is the speed limit for that road at all times.

The vehicle speed is then compared to the speed zone database, which includes the speed limit for that section of road. If the vehicle is travelling above the speed limit, ITWAVs device warns the driver using visual and audible alerts, and then the driver can choose to reduce the vehicle speed, or allow the vehicle to automatically reduce its speed to match the prevailing speed limit, but in this case the automatic violation will be sent to record on the server, while alarm message will appear on the user screen, which will tell him about his violation.

3.5 System Flowchart

Figure (3.7), illustrates the system main flow chart .Detailed algorithms will be provided in chapter 4 later on.

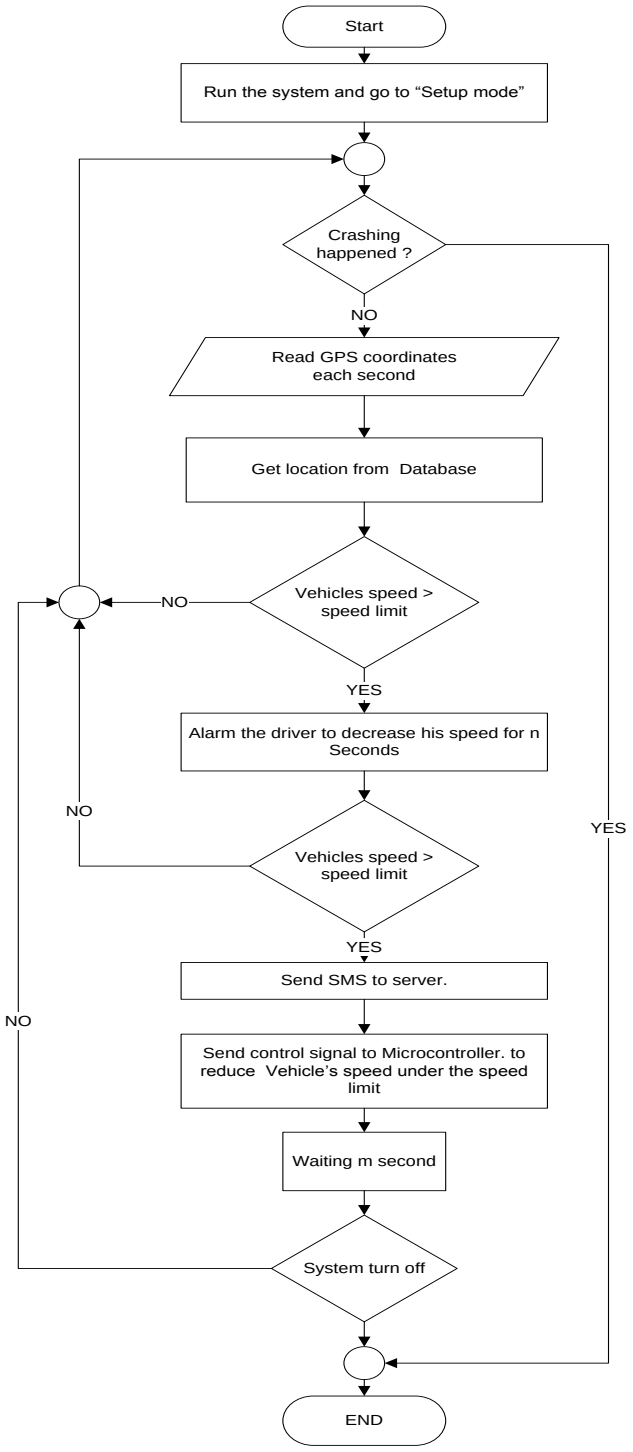


Figure 3.6:*System Flowchart*

The following steps, shows the details of each previous flowchart entity, (Tablet functions):

1. To run the system, the driver must turn on the Android navigator then run the system application (ITWAVs).

2. The driver must fill his registration data (username, password) when the vehicle is stationary.
3. The in vehicle GPS receiver starts to read GPS coordinates, and compares them with the stored database to determine the current road location, and its speed limit.
4. If the vehicle exceeds the speed limit of the street, the system will display alarm message on the screen for known period of time, which claims the driver to manually decrease his speed through this time.
5. Unless the driver responds to the alarm message, the system will go to the automagically decreasing speed of the vehicle until it reaches the zone speed limit.
6. This will be done by sending control signal from a microcontroller to the mechanical system, which will gradually start the decreasing operation.
7. The system waits signal from the microcontroller, which means that the decreasing operation done and reads the next GPS new coordinates.
8. When the vehicle turns off, the system ended.

3.5.1 Set up mode

- Set up mode responsible for prevents any unwanted person from driving the vehicle without applying our system.
- This system prevents the users from driving the vehicle until entering his username and password.

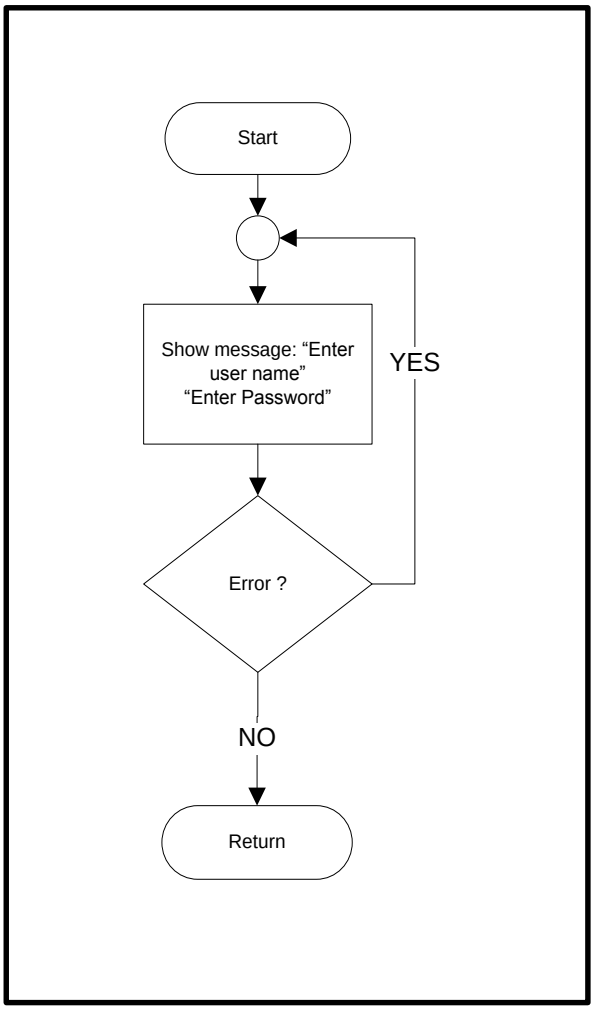


Figure 3.7:*Setup mode flowchart*

3.5.2 Comparing the current GPS coordinates with the stored database

- the next step is to know the current road that the vehicle located on it .
- It is easy to know the speed limit of the current road stored inside data base.

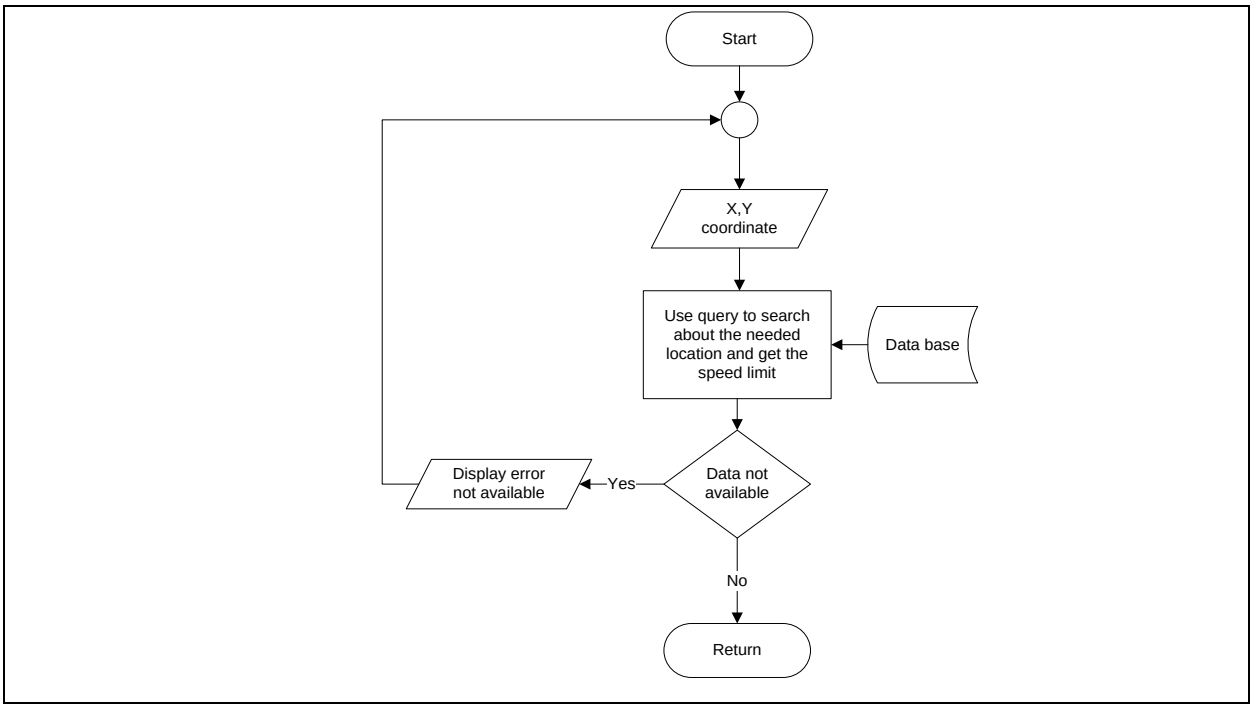


Figure 3.8:*Comparing the current GPS coordinates with the stored GPS database*

3.5.3 Sending data

- After the speed violation occurred , i.e. the driver exceeds the speed limit, and he doesn't decrease his speed through the counter time, android navigator will send messages to multi recipients as the following :

  1- The first recipient is "server": which will receive the SMS with special format, which will be explained in details later in this chapter.
  2- The second recipient is "microcontroller": which will receive the control signal that will activate the mechanical part to gradually slow down the vehicle until it reaches the speed limit.

3.5.4 Arduino Flowchart

The following flowchart shows the functions related to Arduino microcontroller in our system. Arduino will receive data from android via Bluetooth , and check whether the data is for activate the braking system or accelerator pedal .According that , the microcontroller do some tasks will explained in next chapter .

Arduino will receive, send, and process the data as the following:

1) The setup function initialize the selected pins as input mode or output mode, and initialize the needed variables.
2) The loop function detect if there is data available from Bluetooth, deactivate the a accelerator pedal when in wrong id or password is enter, reduce the speed of car by break System.

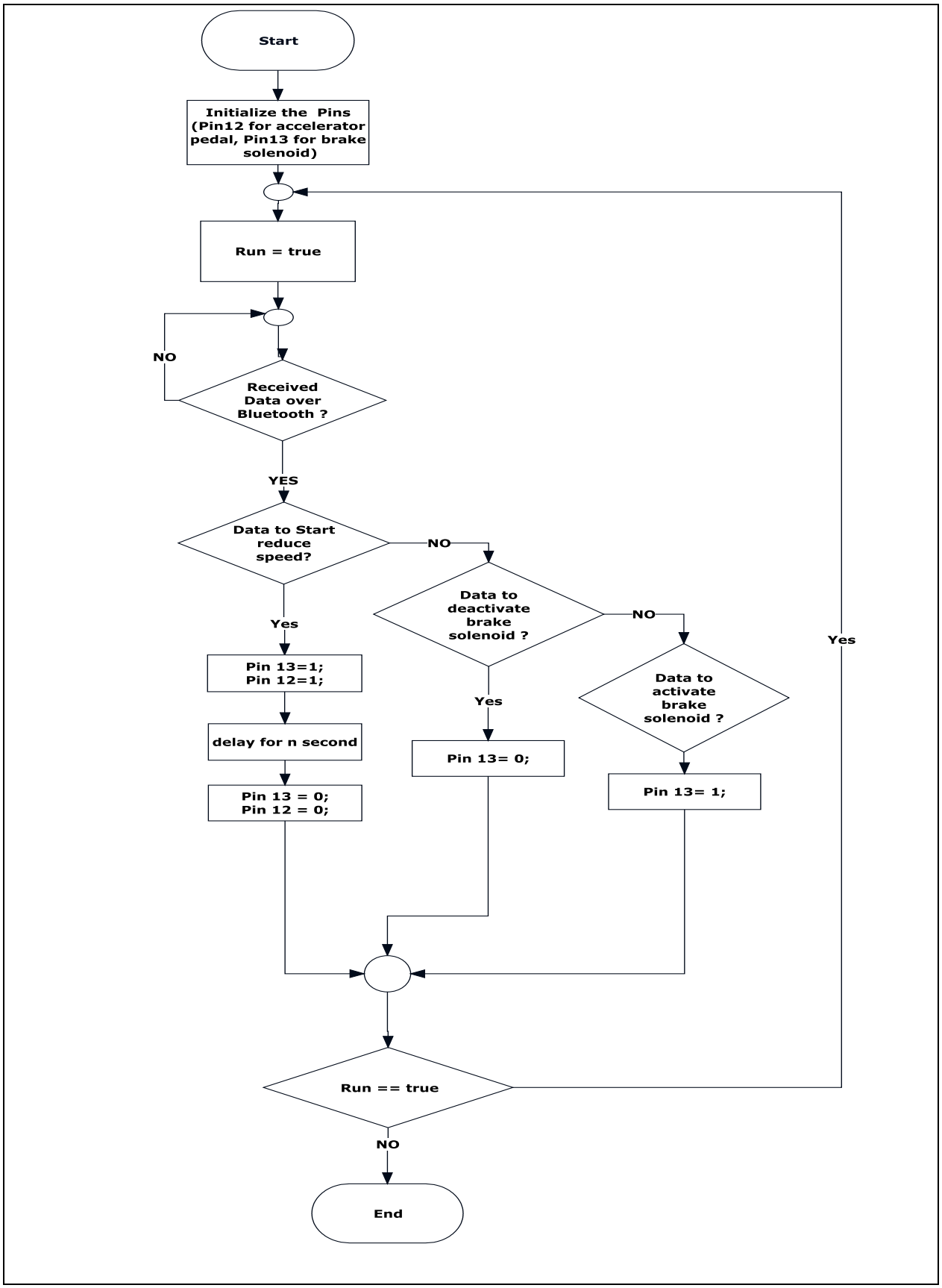


Figure 3.9:*Arduino Flow Chart*

3.6 Crashing System Design

The second part of this system is to deal with the traffic accidents. The system uses accelerator sensor built in mobile phone to detect that the crashing occurred.

As the name suggested, the aim of the system is to monitor accidents, and when there is a crash, it calls for help on emergency number. So, the first thing for this system is to monitor the accident. The second thing is when the accident is detected; it finds its location, so that rescue can be provided at that location. And the third task is to send the location to the emergency number so that they can provide help.

The following figure shows the crashing system block diagram:

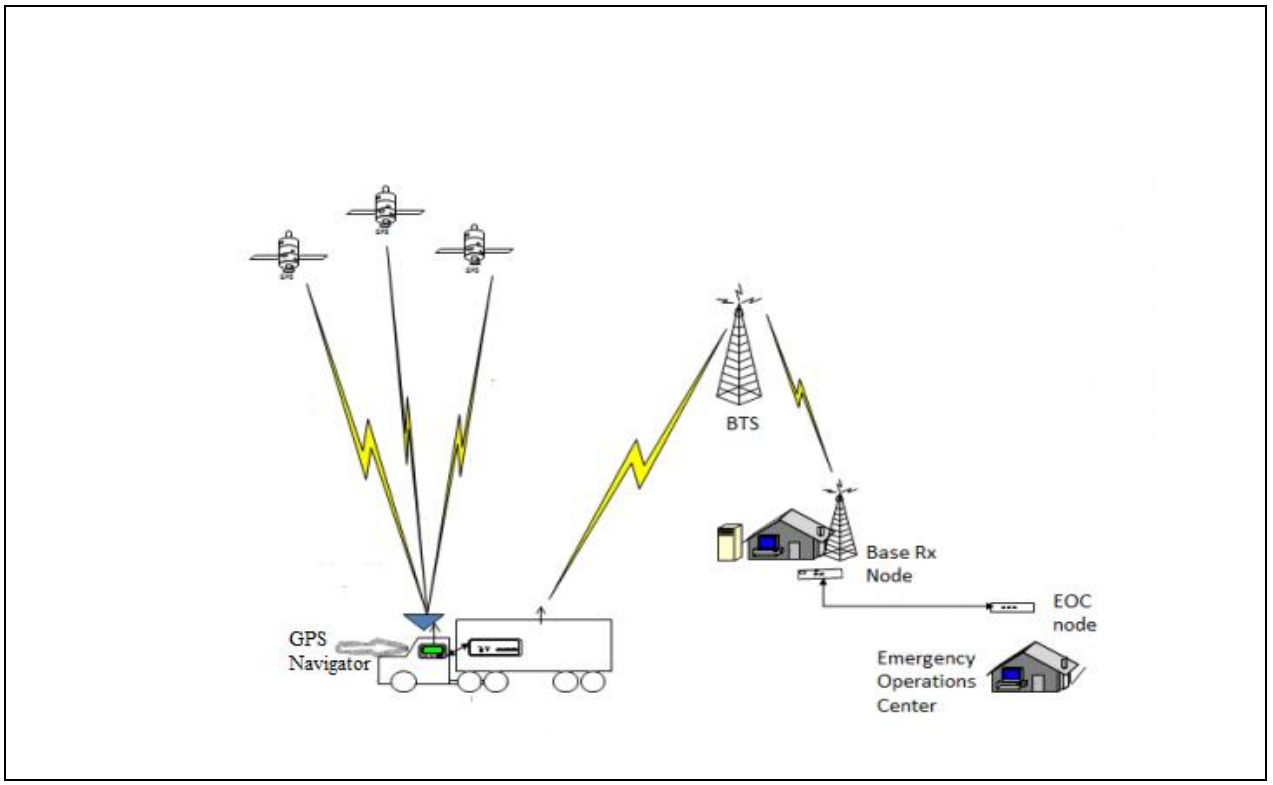


Figure 3.10:*Crashing System Block diagram*

3.7 Central Server System Design

The third component of the system is the central server system. The central server has the ability to receive the violation or any formatted massage from the system on the car, and processed, Analysis the violation to be able to store it on the driver database, and provide the person who work on the server with clear system interface, and provide other available services like statistics for violations, and violations for any drivers.

The application on the server can be programmed using one of the common programming language, like Java, C#, C++, Visual Basic, and many other choices that all has its own features. In this system it will be programmed with C# language, because it provides many features, like easy to write code, clear user interface, and built- in library to provide serial port.

3.7.1 Receiving and storing violations

The server waiting to receive any violation or formatted massage from the car, These violations need to processed and analyzed by the server to get information about the driver, car, street, date of violation, type of violation, and other attribute from the received massage, after that store it on its own database.

The central server side is described as shown in figure (3.12),Where the main parts of the received process are:

- First: find the available serial port, open the selected port.
- Initialized the GSM modem to be able to receive SMS.
- Then wait for incoming massage violation received by GSM modem from the GSM modem in the car. The system accepts only formatted violation massage, and rejects any unformatted massage.
- Then when the violation gets accepted, it will be processed and analyzed to generate the violation and store the violation on the data base .
- Finally, generate the violation report which can be direct printed or store in folder.

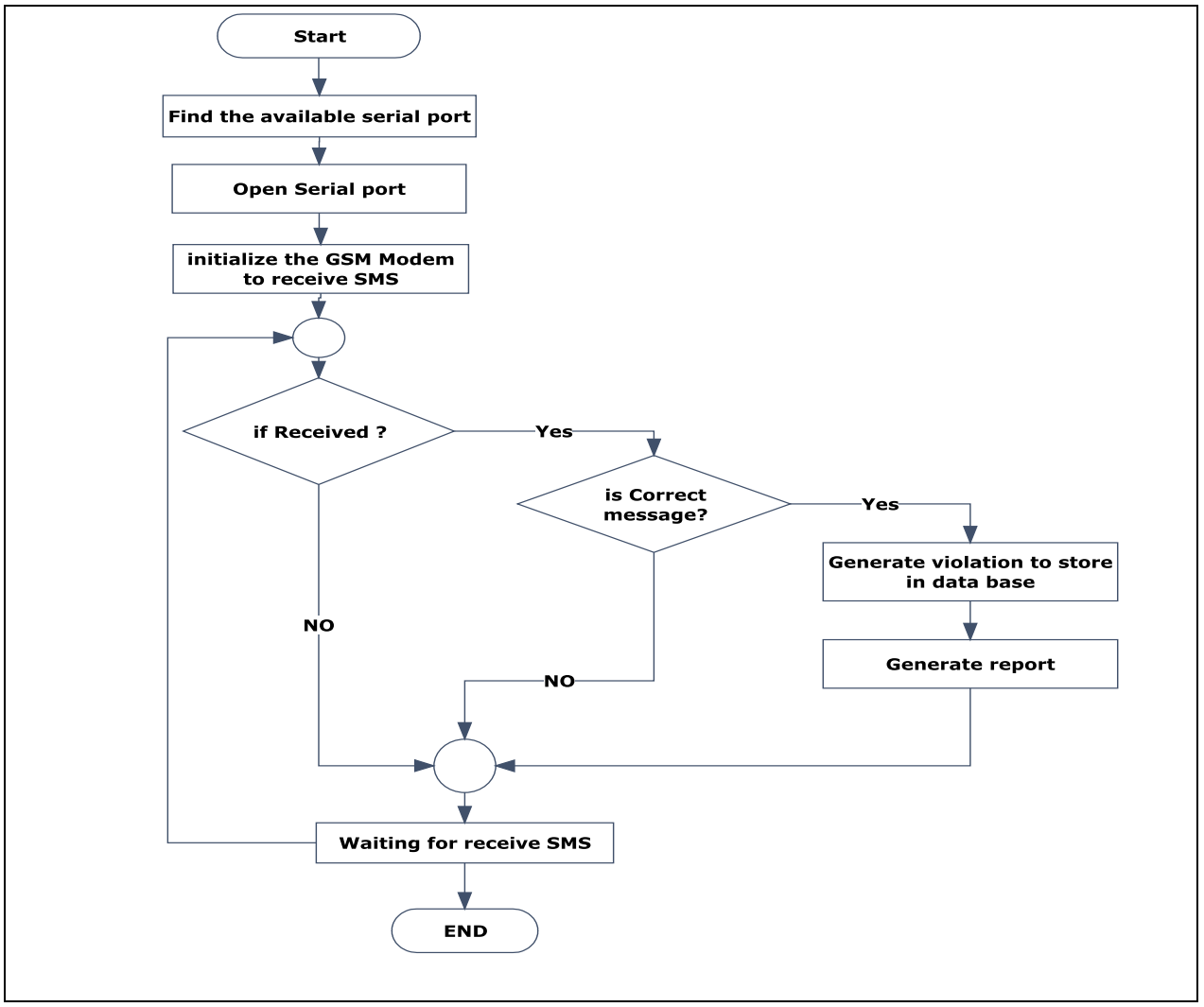


Figure 3.11:*Server receiving violation flow chart*

3.7.2 Violation massage

This massage is sent from the android device in the car to the central server. It is sent only after violation detection.

The Format of the violation massage incoming from the system in the car are :

Driver ID $ Car ID $ Violation cost $ Location $ violation type

The massage fields are :

- Driver ID : The Identification number of the driver who drive the car, or car owner.
- Car_ID : Contains the Identification number of the car .
- Violation cost: the cost of the violation.
- Location : The country -Street where the violation occur .
- Violation type : Contains the type of the violation .

  The delimiter between the parameter in the massage are '$' .

3.7.3 Entity-Relationship Model (ER Model)

ER Model describes the driver database as high-level conceptual data, as shown in figure 3.13:

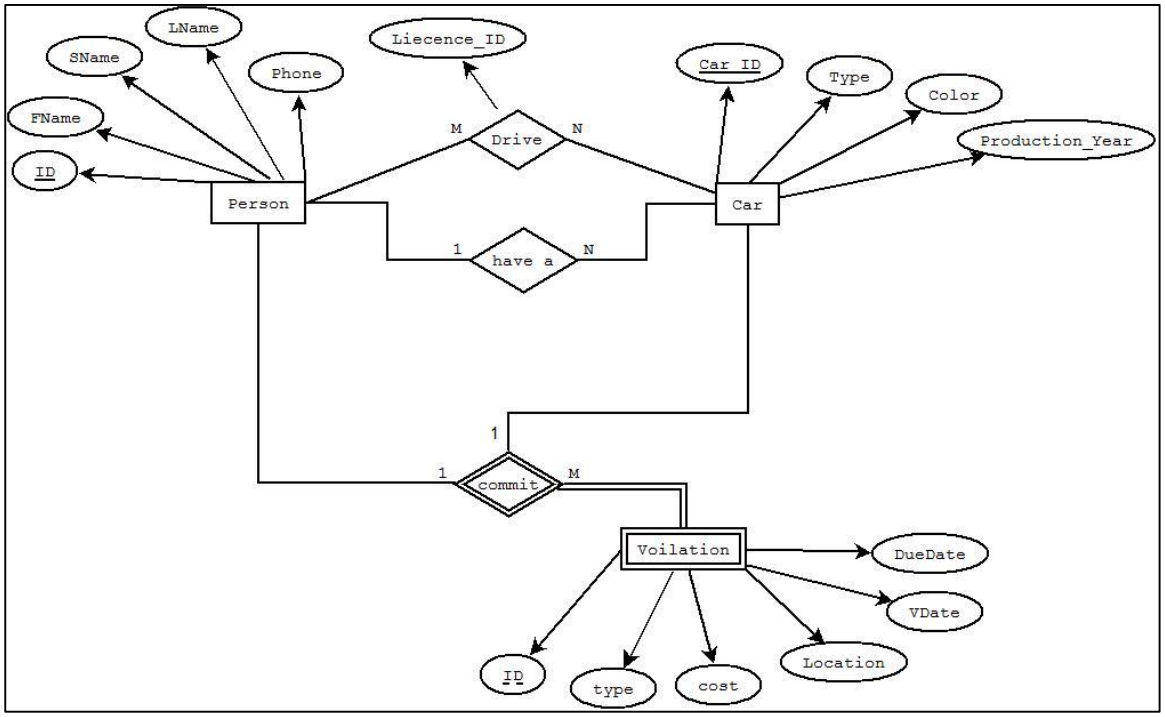


Figure 3.12: *Server ER model*

Person table stores information about personal information of any person Registered in the system such as Person's ID, First name, second name, Last name, and person phone . The Car table contain information about a registered car such as Identification number of the car, Type, Color, Rroduction_Year of the car. The violation table store information about the violation such as ID, type, cost, location, Date when violation occur, last date to pay the violation (Due Date).

Mapping a conceptual schema into a logical (relational) schema

The relational model represents the database as a collection of relations. Informally, each relation resembles a table of values or, to some extent, a "flat" file of records.

The second step of building database, is mapping conceptual schema into a relational schema, by use a seven-step algorithm to convert the basic ER model constraints - entity types (strong and weak), binary relationships (with various structural constraints), n-ary relationships, and attributes (simple, composite, and multivalve) - into relations. Figure 3.24 Schema diagram for the server database**,** show the tables produced by mapping to relations.

Person

ID	Fname	Sname	Lname	Phone

Car

Car_ID	Type	Production year	Color	OID

Violation

PID	Type	VDate	Cost	Location	DueDate

Drive

DID	CID	LiecenceID

The algorithm used to convert to relational model have seven-step : mapping of regular entity types, mapping of weak entity types, mapping of binary 1:1 relationship types, mapping of binary 1:N relationship types, mapping of binary M:N relationship types, mapping of multivalued attributes, mapping of N-ary relationship types.

Relation schema using the data type of each attribute :

This schema show the tables have the attribute with its data type :

- Person (ID: integer, Fname: string, Sname: string, Lname: string, Phone: integer).
- Car (Car_ID: integer, Type: string, Color: string, Production_year: date, OID: integer).

- Violation (ID: integer, PID: integer, CID: string, Type: string, Vdate: Date, Cost: integer, location: string, DueDate: Date).
- Drive (DID: integer, CID: integer, Liecence_ID: integer).

The final step of mapping, decides the foreign key and primary key between tables, then draw arrow , the tile of arrow represents the foreign key and the head of the arrow represents the primary key, Figure 3.14 Referential integrity constraints displayed on the server relational database schema, show the tables after mapping and foreign key and primary key is drawn.

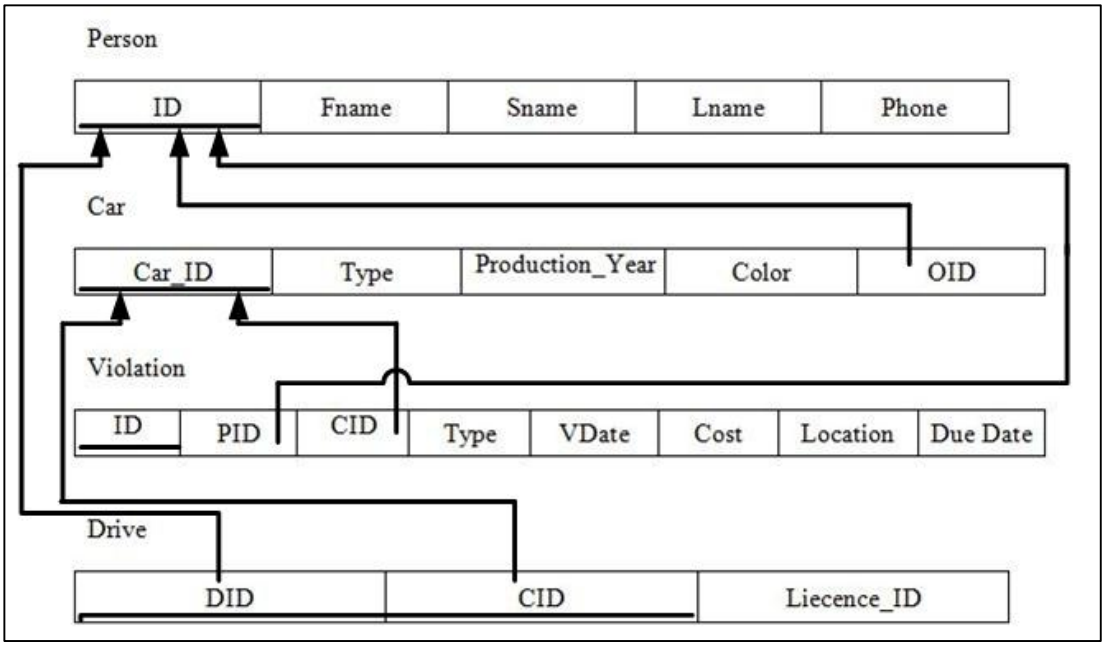


Figure 3.13: *Referential integrity constraints displayed on the server*

3.7.4 Central server function

1. Receive data from GSM, which is a violation that contains: ID of the driver, location of the violation occurred, Cost of violation, car_ID, type of violation.
2. Process received data: the data received will be processed in the same way, to get all attribute contain.

3. Create user interface: to make our system usable and easy for the system server side user to; a clear interface must be created, and all services that available by the system will be set.
4. Record violation: after process receive violation, create a new insertion query to record the violation in the database.
5. generate violation reports which can be direct printing or store in folder as pdf.
6. generate statistics about any city, most street danger.

An example of insertion:

INSERT into violation (Driver_ID, Car_ID, Type, VDate, Cost, Location, DueDate) VALUES (987654321, "22-22222-22", "traffic violation Speed", "22/11/2012-12:40 AM", 350, "Hebron-Alsalam Street", "22/12/2012:12:40AM").

7. View driver violation: the system must be able to save any violation received and recorded it, and have clear way to retrieve this violation.

8. Waiting new connection: during all process of the system, it must still able to receive any new violation come from any car, process them, and store the violation in the database, without any reflection in the processes sequence or connections.

4

Chapter Four
Detailed Design

4.1 Functional Block Diagram.
4.2 Hardware System Design
4.3 Crashing system.
4.4 Location determination algorithm
4.5 Software system design
4.6 Database
4.7 Server Functional requirements
4.8 Serial manager module
4.9 Print manager module
4.10 database interface module
4.11 System use cases
4.12 suggested methods to reduce speed
4.13 accelerator pedal position sensor APPS
4.14 Brake actuator

Chapter 4:

Detailed Design

4.1 Functional Block Diagram

The following figures shows in details the interconnection between the system elements:

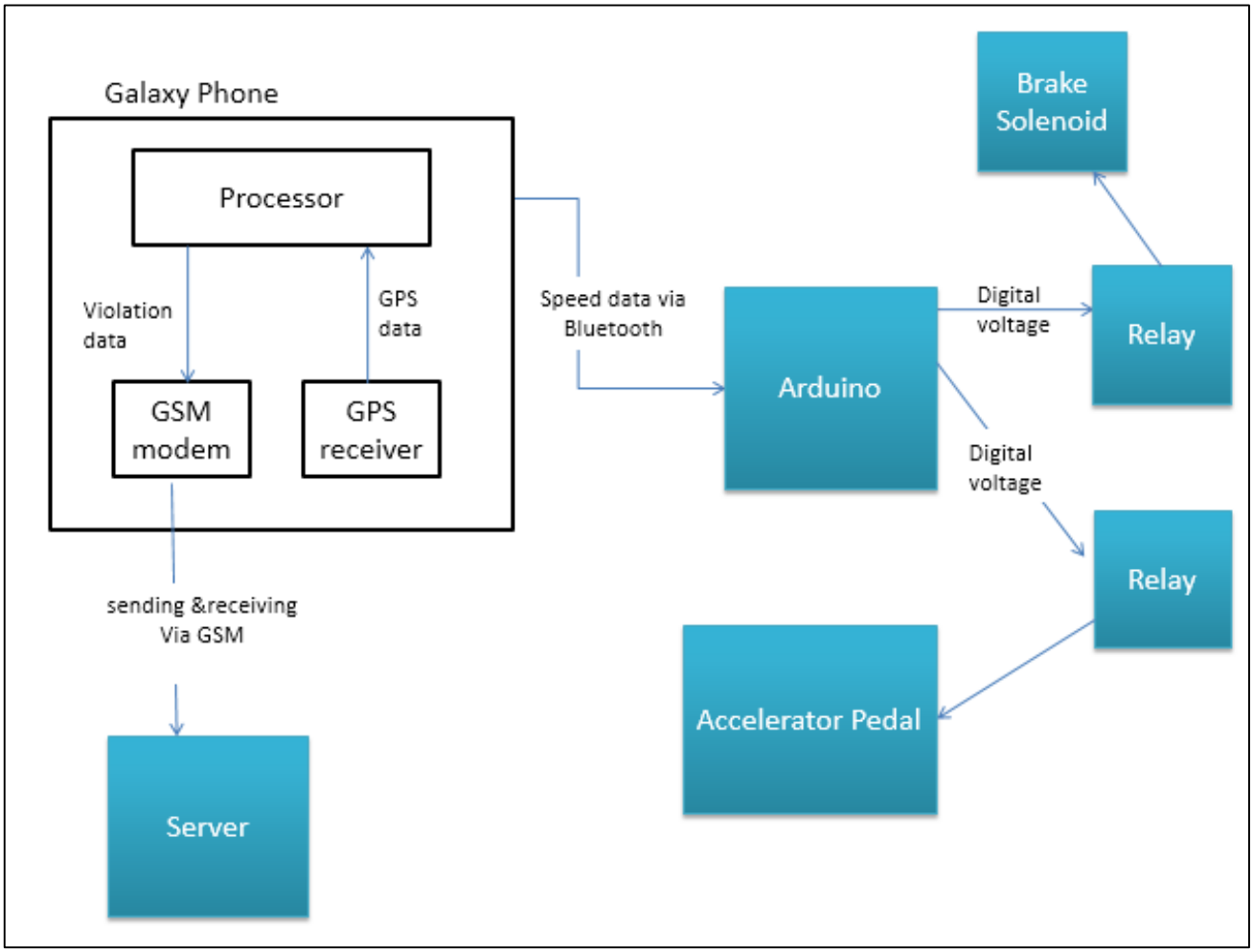


Figure 4.1: *Functional block diagram*

4.2 Hardware system Design

4.2.1 Connection between Mobile phone and Arduino microcontroller via Bluetooth

Introduction

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band from 2400–2480 MHz) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security. Created by telecom vendor Ericsson in 1994 [39]. It was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

Bluetooth is managed by the Bluetooth Special Interest Group, which has more than 18,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics [40]. Bluetooth was standardized as IEEE 802.15.1.

4.2.2 Interfacing Arduino Microcontroller

In this system, Arduino microcontroller based on ATmega 2560 chip will be used, connected with Android device, and mechanical system.

Arduino microcontroller links between the Android device and the other system. It receives data from Android device and process it to control the speed reduction system via Bluetooth connection, and provide android device by the electricity via USB as shown in the following figure:

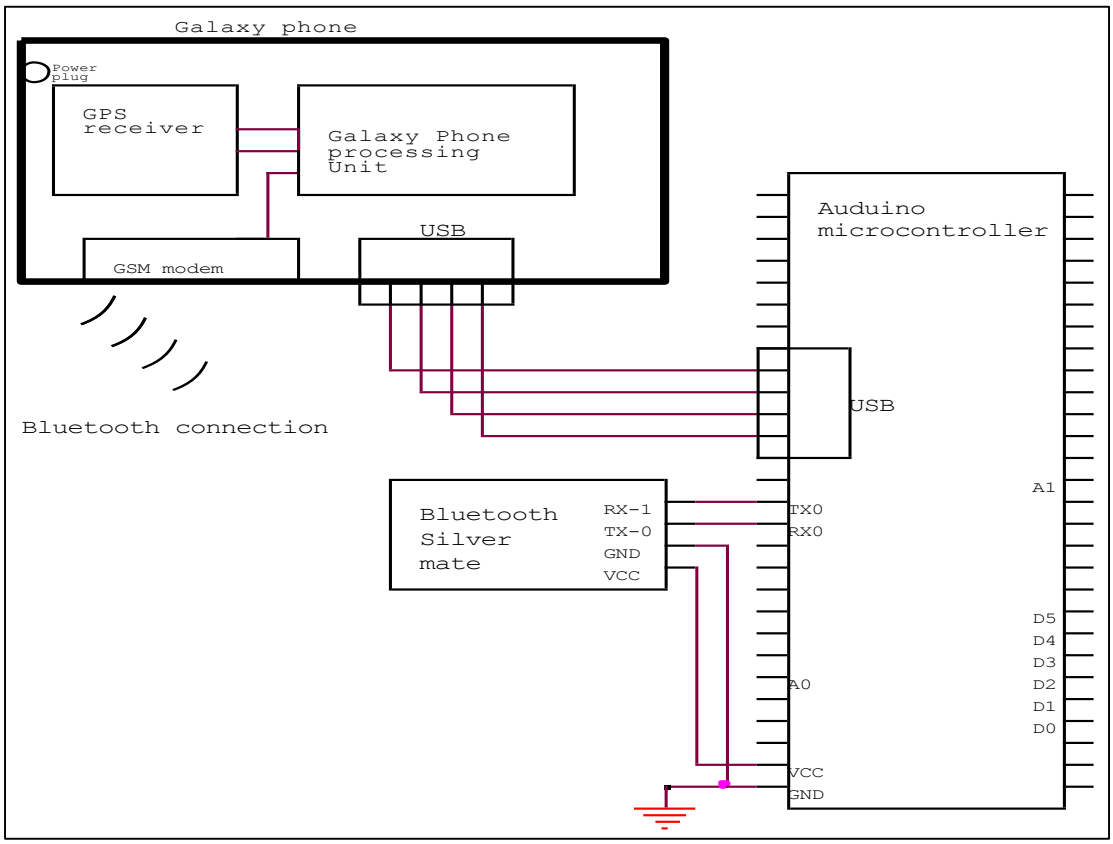


Figure 4.2: *Interfacing Arduino with Android Device*

4.2.3 GSM/GPRS modem Interfacing

In this System, GPRS Modems are the devices that involve machine-to-machine communications, shown in details in chapter 3. AT commands will be used to control the GPRS modems.

AT commands

AT is the abbreviation for Attention. These commands came from Hayes commands that were used by the Hayes smart modems. The Hayes commands started with AT to indicate the attention from the MODEM. The dial up and wireless MODEMs need AT commands to interact with a computer.

AT commands with a GSM/GPRS MODEM or mobile phones can be used to access the following information and services:

1. Information and configuration pertaining to mobile device or MODEM a SIM card.
2. SMS services.
3. MMS services.
4. Fax services.
5. Data and Voice link over mobile network.

The Hayes subset commands are called the basic commands and the commands specific to a GSM network are called extended AT commands.

Command, Information response and Result Codes

The AT commands sent by the computer to the MODEM/mobile phone. The MODEM sends back an Information response i.e. the information requested by the AT command. A Result Code follows this. The result code tells about the successful or failed execution of that command. Figure (4.3) shows the block diagram of the execution steps of the AT commands.

There are also unsolicited Result Codes that are returned automatically by the MODEM to notify the occurrence of an event. For example, the reception of SMS will force MODEM to return an unsolicited result code.

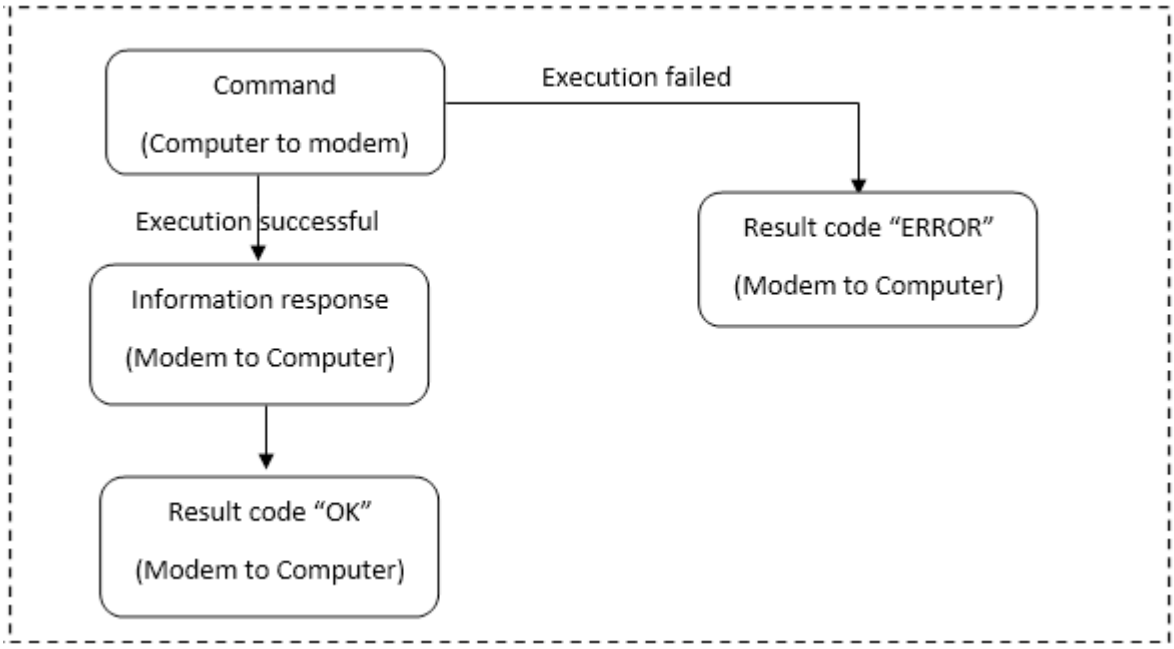


***Figure 4.3: The** execution steps of the AT commands*

Different AT commands Result Codes:

The following table show the most important AT commands result codes, which may appear when configuring the GSM/GPRS modems.

***Table 4.1:** AT Command result codes*

RESULT CODE	DESCRIPTION
OK	Successful Execution of a command
ERROR	Execution of a command failed
+CMS ERROR	Message service failure, is returned with an error code
Unsolicited Result Codes:	
+CMT	Notify forwarding of a new SMS to computer
+CMTI	Notify receipt of SMS status report of a new message and its location in memory to computer

Sending data from In-vehicle Mobile over GSM network to the receiver GSM modem:

When a computer program needs to connect to a local or wide area network such as the Internet, GSM, it uses a special software, which opens the network connection for the programs via computer ports, to read and write data over the network. It is important to know that these ports are software, not hardware.

4.3 Interfacing ZTA MF180 MODEM with Windows platform

Windows operating system comes with an application called **HyperTerminal** for data communication through serial port of the computer. The interfacing of the GSM/GPRS module with the serial port of the computer.

4.4 Crashing System

Accelerometer Sensor are used to construct transducers for a large number of different applications. Accelerometer Sensor generate an electrical charge in response to mechanical movement, or vice versa, produce mechanical movement in response to electrical input.

4.4.1 Theory and Modeling

The basic theory behind Accelerometer Sensor based on the electrical dipole. At the molecular level, the structure of a Accelerometer Sensor is typically an ionic bonded crystal. At rest, the dipoles formed by the positive and negative ions cancel each other due to the symmetry of the crystal structure, and an electric field is not observed. When stressed, the crystal deforms, symmetry is lost, and a net dipole moment is created. This dipole moment forms an electric field across the crystal.

In this manner, the materials generate an electrical charge that is proportional to the vibration applied. If a reciprocating force is applied, an AC voltage is seen across the terminals of the device. Accelerometer sensor are not suited for static or dc applications, because the electrical charge produced decays with time due to the internal impedance of the sensor, and the input impedance of the signal conditioning circuits.

Accelerometer sensor shown in figure (4.4),is an electronic device that generates a voltage when it is physically deformed by a vibration, sound wave, or mechanical strain. Similarly, when you put a voltage across accelerometer sensor, it vibrates and creates a tone [42].

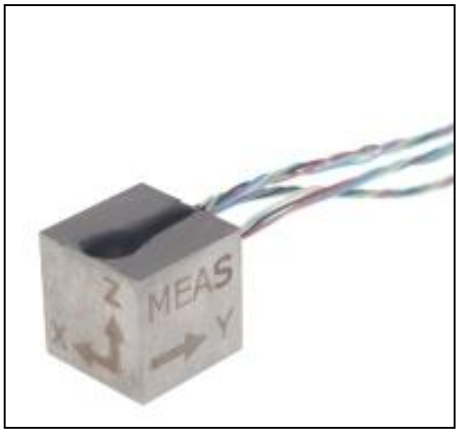


Figure 4.4: *Accelerometer Sensor*

4.5 Location Determination Algorithm

4.5.1 Overview

In order to determine the speed limit of the street where the car located, the system must match the vehicle location i.e. the current GPS reading with the pre-stored value in database. In this system, we will develop a geometrical algorithm, which will be used to approximate the current GPS data to the closest street.

4.5.2 Problem Statement

The goal is to identify the route taken by a traveler equipped with a GPS data logger. Assuming that the traveler is moving along a transportation network, we can restrict ourselves to a network representation of the spatial environment. The network is coded as a set of nodes connected by directional links. A path is a set of connected links.

The map-matching problem is to find the best estimated of the path that is the best estimation of the route that was actually taken by the user. There are two sources of errors. First, the GPS data loggers have their own limitations depending on the environment (e.g. canyon streets, tree canopies, etc.) and their accuracy. Second, the coded network (database) which approximates the physical world.

Most of the existing map-matching algorithms [43], [44], [45], [46], are more focusing on the accuracy than the computational speed of the algorithms. Therefore, our concern is slightly different from the existing literature.

First, we assume in this algorithm that the only source of data concerning the traveler is a stream of 2D coordinates collected by a GPS logger embarked in a car. In particular, we do not use a DR device and we will not use information about the heading nor the speed of the vehicle.

Second ,the measurement of the overall error as the distance between the GPS points and the coded network is not transferable from one study to another, since it highly depends on the resolution of the coded network. Therefore, the comparison of the algorithms can only be performed on the same data sets. Ideally, the performance of the algorithms should be measured as the ratio of routes that were correctly matched, which in turn would require a tedious manual checking (e.g. using street names).

For these reasons, we will focus only on the operational performance of the algorithms, that is how much faster that real-time they can process large volumes of data with reasonable matching errors (i.e. the routes are checked visually on a map). Note that most embarked GPS navigation systems have map-matching abilities.

To summarize, we have 1) a standard network representation of the transportation system, directed links and "polygons" describing their curvature as will be explained later, and2) a stream of GPS coordinates recorded for every time. The problem is to find the path in the network that is the closest to the GPS points and in minimum computations.

4.5.3 Proposed Algorithm

Let *G (V, E)* be the directed graph describing the road network. *V* is the set of vertices or nodes and *E* is the set of edges or links. Let Q_i, Q_i' be the set of points given by the GPS data stream, set of corrected GPS points respectively , $i = 1...T$, Each GPS point consists of a pair of coordinates (*xi*, *yi*).

In order to minimize the GPS reading error, a small shift perpendicular to the link direction will be introduced, according to the end driving side of the studied area. Therefore, in the computation of the distance, an oriented link *AB* is replaced by $A'B'$ where $A' = A + \Delta$, and $B' = B + \Delta$, Δ is chosen to reflect the average physical distance from each edge of the road. In this system, Δ = 5m in each direction, since the average GPS error is about 10m. Figure 4.5 illustrates the mentioned parameters.

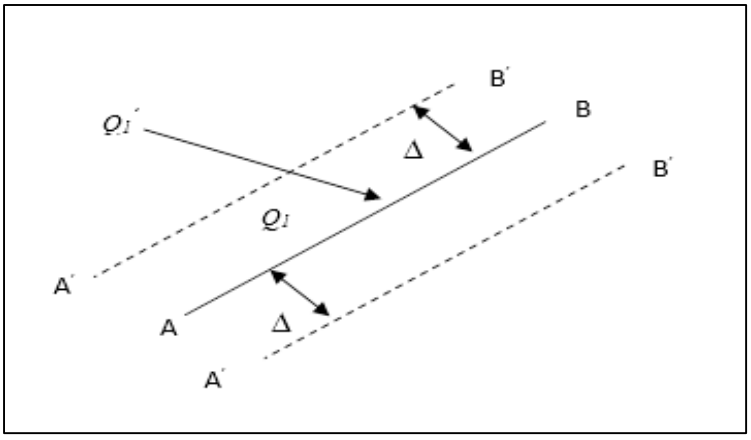


Figure 4.5: *Distance between a point and a road segment*

The most important thing that must be searched is how to satisfy the algorithm objectives with the least computational complexity, and minimum time. So, the Algorithm initialized by distributing the working region into N_i polygons, each polygon is restricted by the lowest and highest GPS points.

Inside each polygon, distribute the streets into S_j internal polygons, each one is restricted by the lowest first and highest end GPS points, whereas a diagonal link can be connected between this two points.

Technically, this algorithm sorts each group of streets into one polygon, which help in decreasing the searching process inside the database. In other words, the searching process will be divided into two stages: first, inside the N polygons to determine in what region the vehicle site (N_i), and the second is the searching process inside the N_{th} polygon, to search about the S_j street.

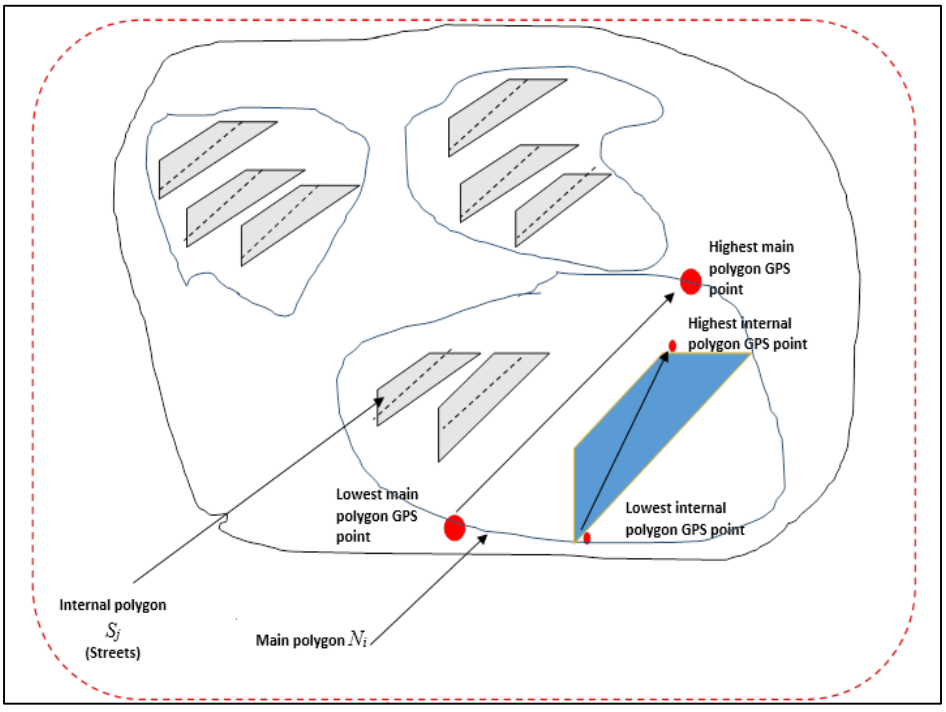


Figure 4.6: *Technical algorithm description*

4.5.4 Building mobile database

As mentioned before, this algorithm depends on determining the vehicle location with minimum possible time, so lowest computation process must apply. Therefore, this concept must apply when building streets database, by applying the following steps:

1. Distribute the intended region into polygons as mentioned before, each with the lowest and highest GPS points. *For example*, in the model applied for this system, we consider that "West-Bank" is the working region, and divided it into polygons, one of these polygons is "Hebron" polygon, with GPS points: [31°27'56.10"N , 35° 4'52.45"E] ~ [31°36'34.10"N , 35° 7'11.61"E], as shown below.

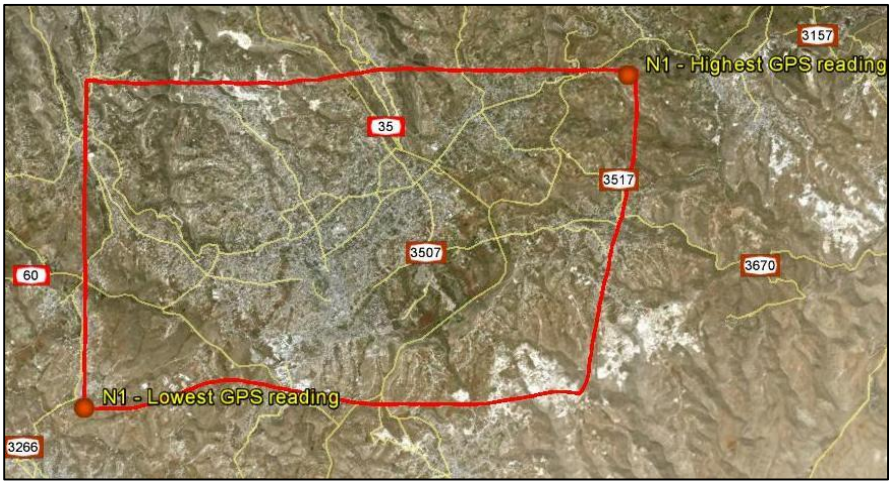


Figure 4.7: *Hebron Polygon*

2. Inside Each polygon, arise streets polygons, each one determining by two diagonal GPS points as shown in the following figure, and for each street Expand them by (5m) on each direction (East, West, South, North), in order to minimize the probability of error, as shown in figure 4. 8.

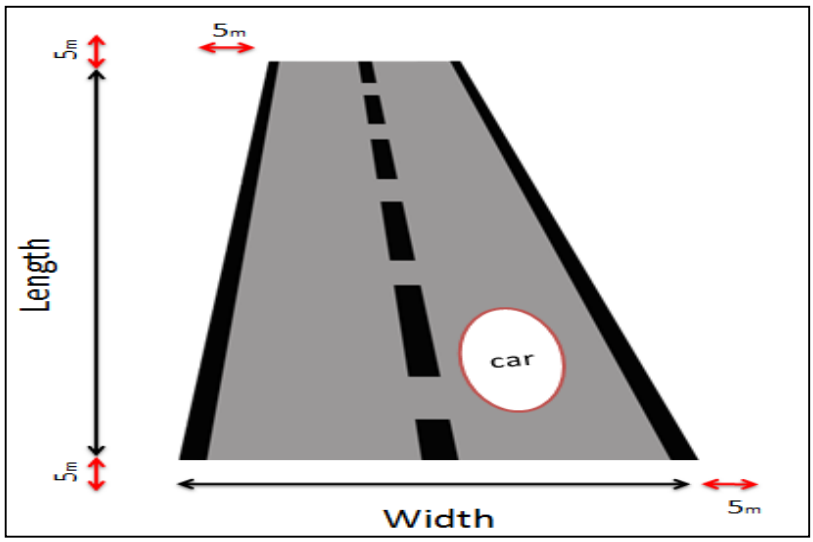


Figure 4.8: *Expand Street by 5m in each direction*

3. From the principle of GPS coordination, the GPS point value will increase or decrease as the direction of moving. Benefit from this principle; expand the diagonal line from the start GPS-stored data, to the end on the opposite side, as shown in figure 4.9, in order to contain the max. GPS coordinates in each sub-polygon.

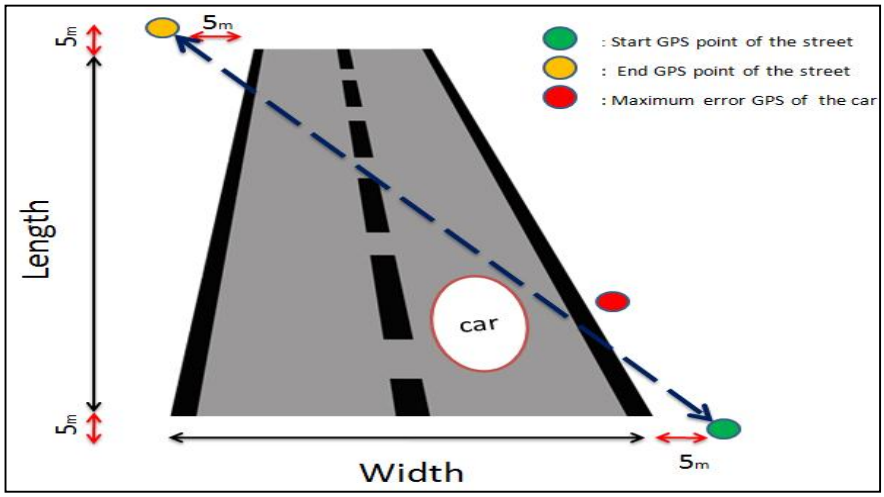


Figure 4.9: *Expanded street with start- end GPS points moreover, the max GPS error point of the car*

In our model, a part of sub-polygon appears in the following figure:

Figure 4.10: *Hebron Sub-polygon*

4. In order to make a comparison, choose latitude coordinates arranged from [Max latitude -Min latitude] for each diagonal sub-polygon, where Max. and Min. latitude represent the start and end GPS of each sub-polygon, that will be stored in the database .Then the system compares the latitude coordinate of vehicle location with these latitudes range stored in database.

5. The problem here, that there will be many streets have the same latitude coordinates .To solve this problem: the system will compare the vehicle GPS location (latitude and longitude) with the street range latitude and longitude, never to get two street having the same range of latitude and longitude.

6. Built in database of each main or sub polygon, will be presented by 4 points (two for latitude and the same for longitude).

Table 4.2: *Sample of streets database*

SPEED LIMIT	ID	END POINT (Longitude)	FISRST POINT (Longitude)	END POINT (Latitude)	FIRST POINT (Latitude)
50	S1	350602	350600	313325	313320
60	S2	350601	350553	313318	313310

7. GPS coordinates form conversion from (degrees) to (XY) form will be used here, in order to simplify the comparing process, using the conversion equation presented in equation (3.1).

4.5.5 Breaking Route into pieces

When the streets are too long, or do not have a linear shape, these streets will be divided into pieces, each one has 4 points.

Query will be used to determine the car location by holding the current GPS data, and then compare it with the database.

The following flow-charts show the Map-Matching algorithm:

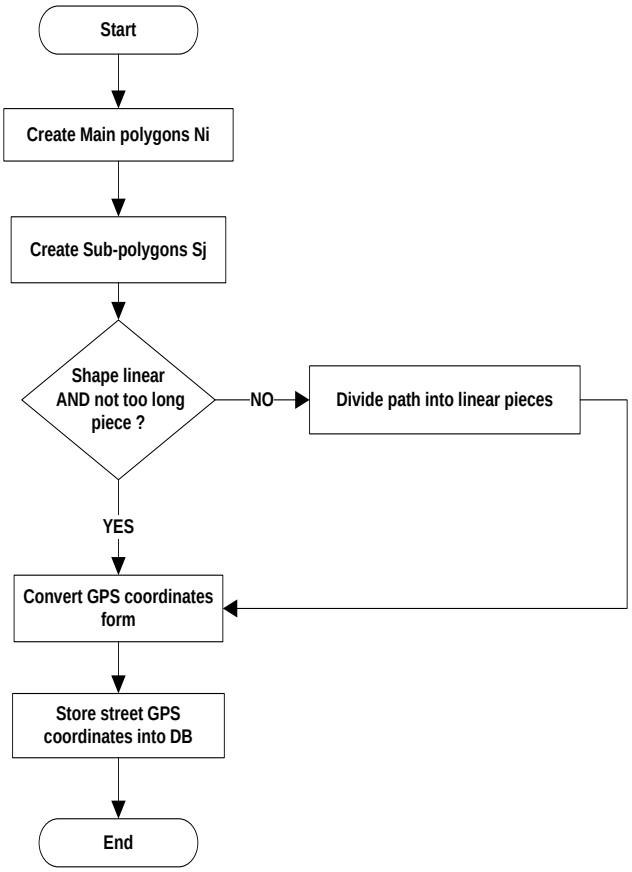


Figure 4.12: *Breaking Route into pieces flowchart*

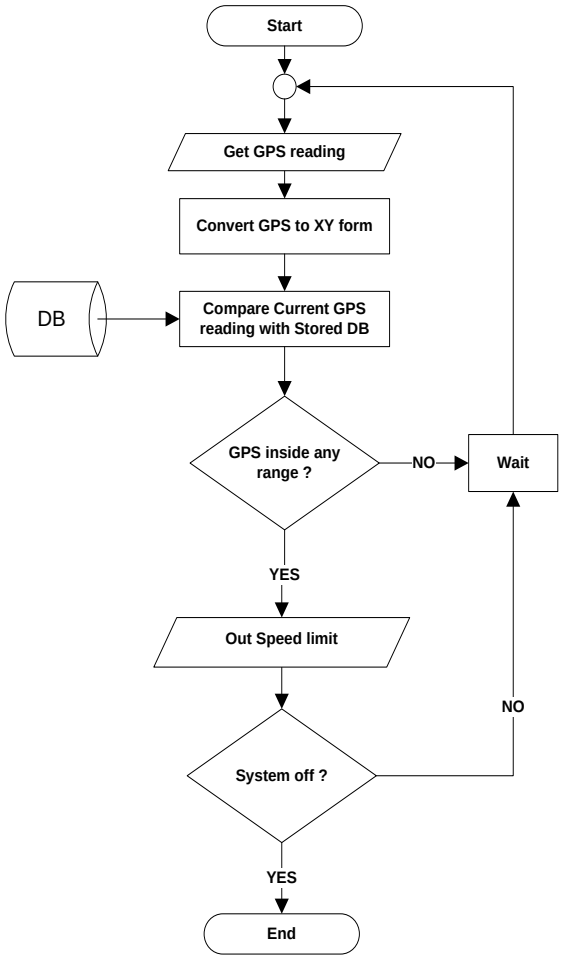


Figure 4.11: *Map-Matching algorithm flowchart*

4.6 Software System Design

We have a primary table with the name "Poly", it has 5 attributes, Tname describes the name other table in the data base, Xs describes the X point start of table, Ys describes the Ypoint start in the table, Xn describes the X point end in the table, Yn describes the end Y point in the table.

The primary table decide how many region are the system can monitor.

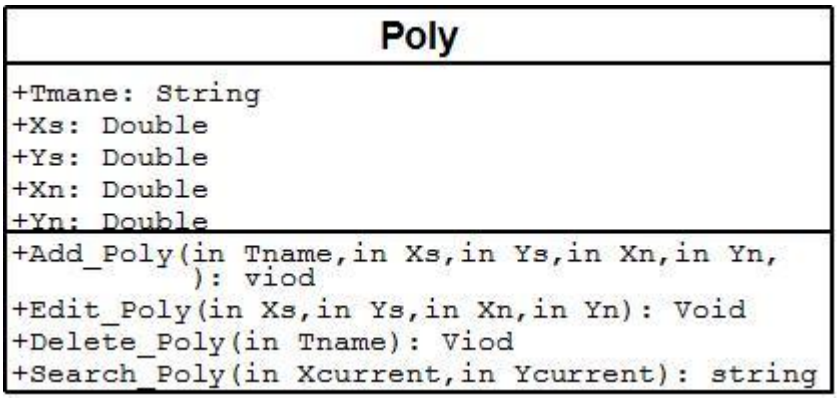


Figure 4.13: *UML Class For the primary table*

The other tables have the same name on the Tname on primary table, and have 7 attributes: ID, Xs, Ys, Xn, Yn, SpeedLimit, StreetName, when add new entry to other tables, Xs,Ys, Xn, Yn, must be between the range of the major entry in the primary table.

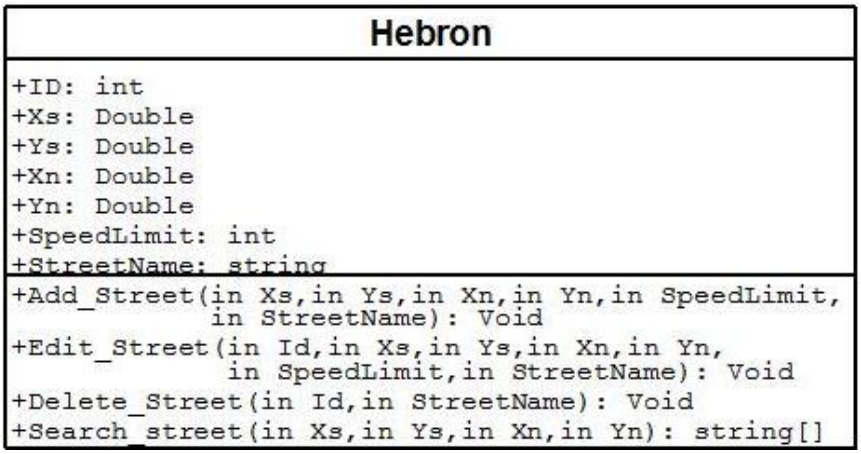


Figure 4.14: *UML Class For Table Hebron*

After the location is determined, speed limit and the name of the street will be returned, by this mechanism, it is an easy to add a new region to the system.

4.7 Database

4.7.1 UML diagram for server data base

UML (Unified Modeling Language) is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology.

UML methodology is being used extensively in software design and has many types of diagrams for various software design purposes. Class diagram can be considered as an alternative notation to ER diagram.UML deferent.

A database is employed within the software to ensure that the violation is saved, and to insure that the user can properly retrieve the history of any violation.

It is also used to add new user and Edit user in the system. The Database UML Diagram and design is presented in the UML Class diagram in figure 4.15.

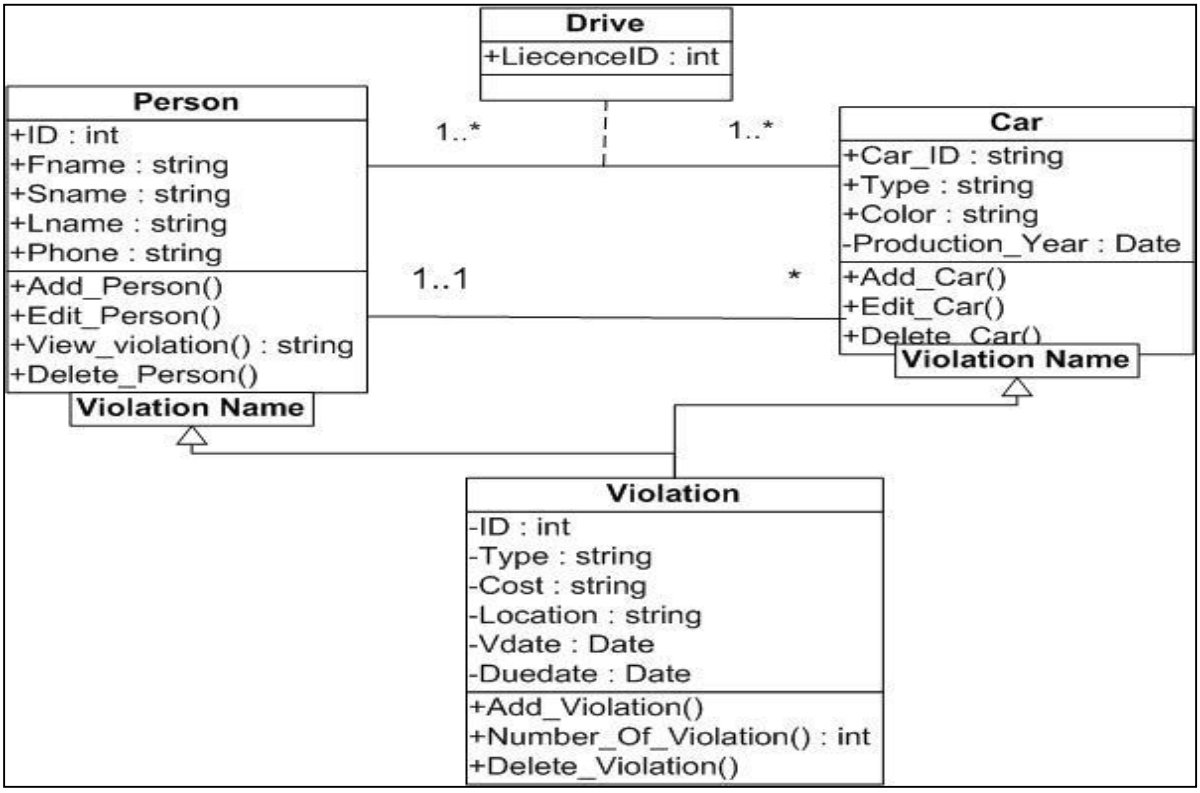


Figure 4.15: *UML class diagrams notation for server conceptual schema*

4.8 Server Functional Requirements

The server software within the project has the following functional requirements:

- The software must be able to communicate with the database, GSM Modem.
- The software must be able to process data taken from GSM Modem, and convert it into suitable understandable data.
- The software must be able to check the valid violation massage and reject other.
- The software must have an attractive graphical user interface suitable for add new user: driver and car .
- The software must be able to store the violation into the database automatic.
- The Soft Ware must be able add, edit, delete user on the system.

- The software must be able to generate violation reports as PDF file.
- The software must be able to generate receipt voucher.
- The software must be able to print violation reports, receipt voucher.

4.9 Serial manager module

This component implements a standard serial communication program. It contains information regarding the port number, parity bits, stop bits, the number of bits for data transfer in the attribute serial manager. In addition to that, it contains functions specific to the SMS functionality, and functions to send and receive SMS as shown in figure 4.16.

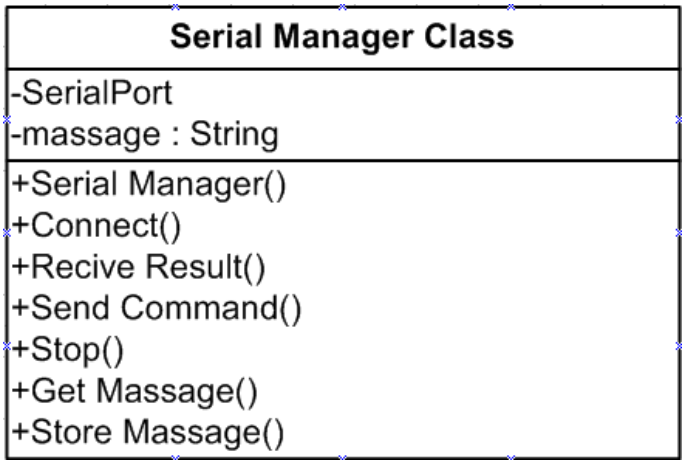


Figure 4.16: *Serial manager Module UML Class Diagram*

The main components for this module are:

- Connect: This function will initiate a new port connection and sets its configurations, then establish a connection. It will also send the following command to ensure that the modem and computer seeing each other : AT
- Stop : This function will end the serial communication across the specified port.
- Serial Manager : This constructor function finds the installed serial ports.
- Send Command : This function will send commands to the GSM modem over serial port.

- Get Massage : This function Will Split the massage from the Result and build a violation string.
- Store Massage: This function will take the violation string and store it data base.
- Receive Result: When data is received on the serial port, an event is raised and Receive Result is called.

4.10 Print Manager module

This component implements a interface a class for print the violation reports. It contains attribute for print the document, and attribute for store the reports. In addition to that, it contains functions specific to print the violation reports directly on line, save the violation reports as PDF file on ITWAVs folder, collect the violation reports, and event is called when print online is selected, as shown in figure 4.17.

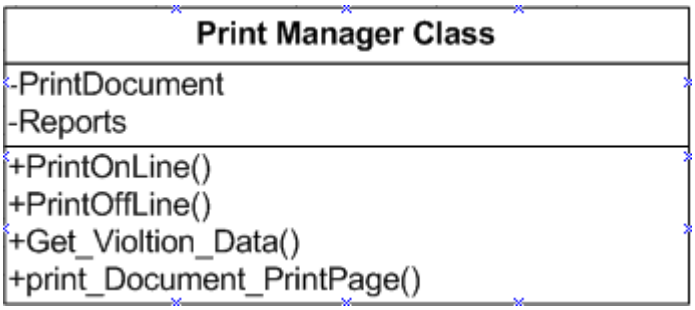


Figure 4.17: *Print Manager Module UML Class Diagram*

The main components for this module are:

- Print Online: This function will print the violation reports directly to printer.
- Print Offline : This function will generate and save the violation reports as PDF file.
- Get Violation Data : This function will collect the violation information.
- Print Document Print Page: This event will called when print online is selected.

4.11 Data Base interface module

This component implements a interface from communication with data base . It contains attribute for read result from SQL query, connection for data base, and attribute for SQL command. In addition to that, it contains functions specific to insert, update, delete, view the tables on the data base as shown in figure 4.18.

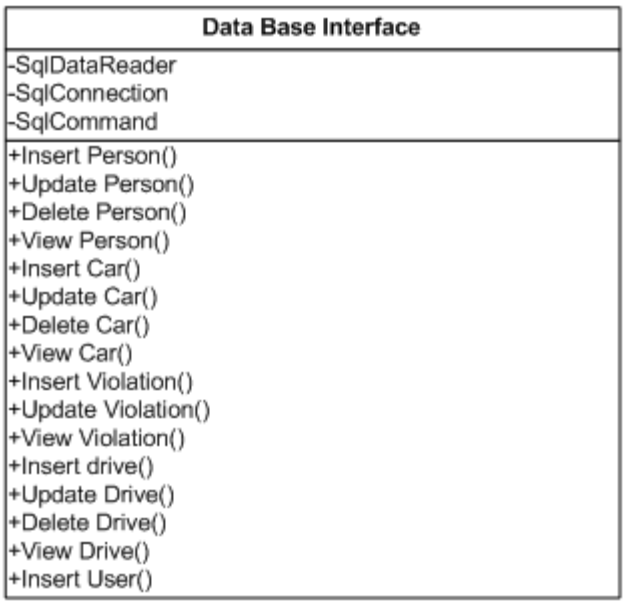


Figure 4.18: *Data Base Interface Module UML Class Diagram*

The main components for this module are:

- Insert Person: This function will execute a query to insert person in database.
- Update Person: This function will execute a query to update person in database.
- Delete Person : This function will execute a query to delete person in database.
- View Person : This function will execute a query to view person in database.
- Insert Car : This function will execute a query to insert car in database.
- Update Car : This function will execute a query to update car in database.
- Delete Car : This function will execute a query to delete car in database.

- View Car : This function will execute a query to view car in database.
- Insert Violation : This function will execute a query to insert violation in database.
- Update Violation : This function will execute a query to update violation in database.
- Delete Violation : This function will execute a query to delete violation in database.
- Insert Drive : This function will execute a query to insert drive in database.
- Update Drive : This function will execute a query to update drive in database.
- Delete Drive : This function will execute a query to delete drive in database.
- View Drive : This function will execute a query to view drive in database.
- Insert User : This function will execute a query to insert user in database.

4.12 System use cases

Details of how the system outside users, and the system actions will interact with each other during the process of the system, will be described in the system use cases clearly in this section.

1) **Server Use case:**

From Figure 4.19, we can note that:

- Three actors mainly interact to server system, the admin ,user and the GSM modem.
- The admin who will start the server application.
- The GSM Modem receives the violation data using GPRS, sends it to server application and stores violation on database.
- The user can Add, Edit, Delete, on his account, and pay the violation by the admin.
- The admin can add, edit and delete users stored in database.
- The admin can print and view the violation reports for any user.

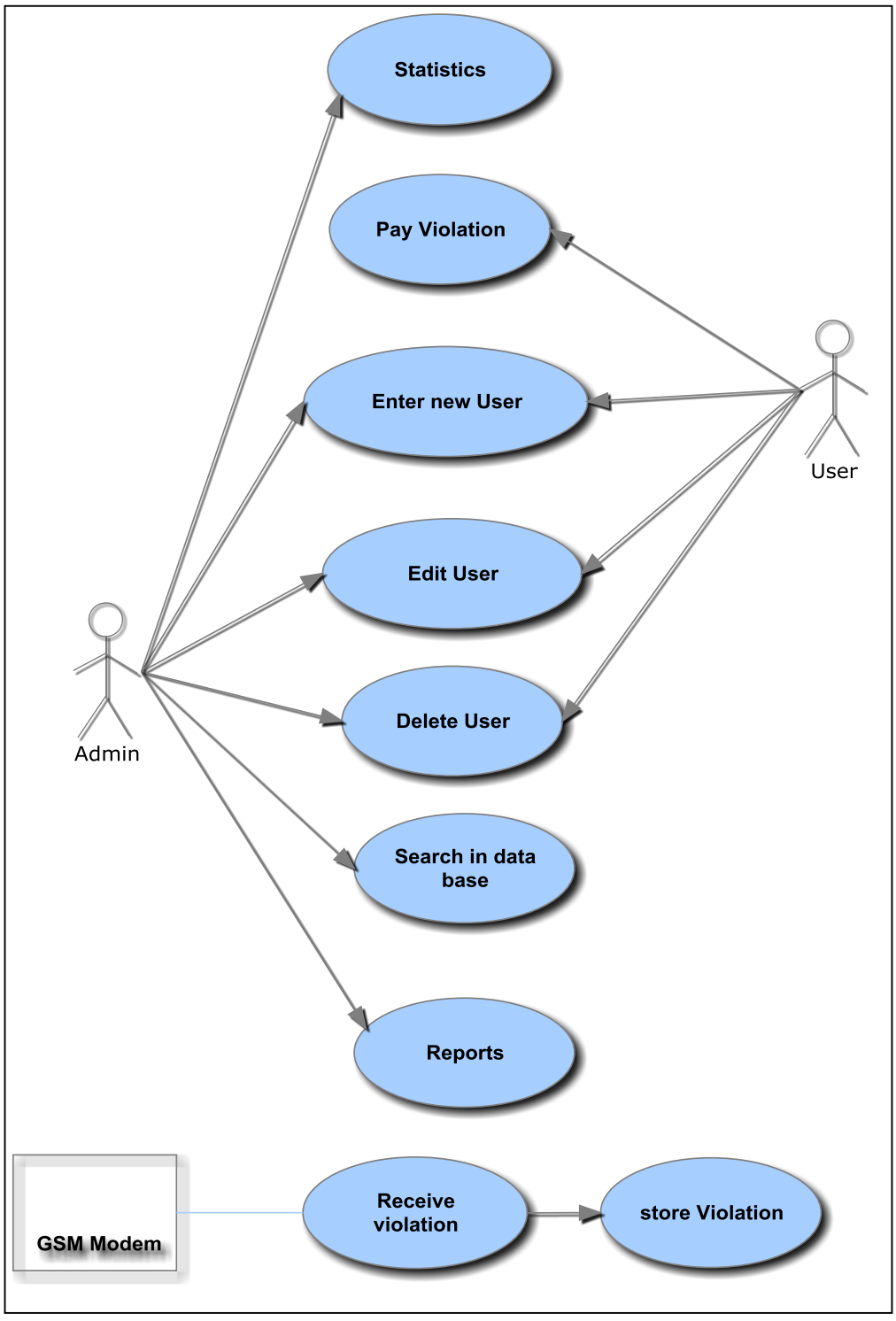


Figure 4.19: *Server Use case*

2) Client user case

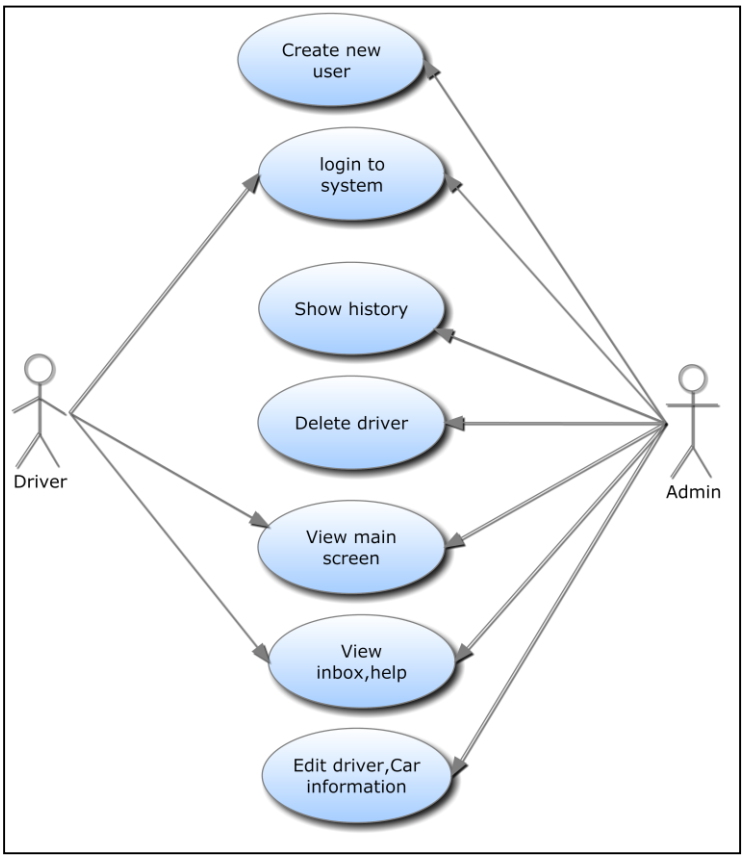


Figure 4.20: *Client use case*

From the figure 4.20, we can note that:

- Two actors mainly interact to the Android system, the admin, and the driver.
- The driver log in the system to enable the switch to run the car.
- The driver after log into the system can edit the password of user, and can edit the setting of the application.
- After log in, no action is required from the driver, the user could view status screen only.
- The admin can edit all passwords of users; edit the data of road in database.
- The admin can change the role of the system, such as time of alarm.
- The admin can creates new user, edit and delete, the users in the system.

Part II: Reduce vehicle speed technique

4.13 Suggested methods to reduce speed

Before selecting the appropriate brake actuator, there were many options to reduce vehicle speed, some of them:

1) Insert throttle with throttle motor in exhaust pipe: but this method leads to increase the reflected pressure which affect on engine operation, and may stop the engine completely if the engine in the exhaust stroke.
2) Insert throttle with throttle motor in air cleaner intake hose before intake manifold: but this method has problem which is just controlling the intake air, and ignore other variables. The accelerator pedal is depressed the same time potentiometer gives reading to the ECU, and there is no air entering to the intake system, so the computer gives "out of rang" reading.
3) Use linear DC Motor: this method can cause conflict with the brake pedal, and control of dc motor is not easy and complex.
4) Inserting hydraulic piston: this method required space to install hydraulic pump, oil tank and hydraulic piston. In the other hand it has long response time and expensive.

After deep searching, the decrease of speed will go through this sequence operation. First decrease the APPS output voltage (driver demand) as the accelerator pedal depressed. After that activate the brake actuator (starter solenoid), by signal coming from controller.

4.14 Accelerator pedal position sensor (APPS)

When the driver depress on the accelerator pedal, two potentiometers installed on the pedal to measures the angle, two potentiometers used to enhance the accuracy of reading, the schematic diagram are shown in the next diagram.

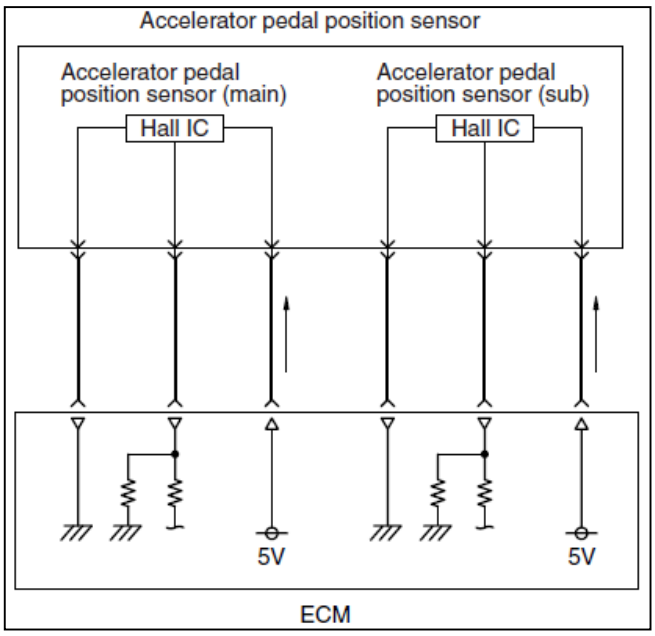


Figure 4.21: *Schematic diagram for APPS*

The two potentiometers are supplied by (5V) and the third pin connected to ground, but the difference in the output voltage pin.

The first one (main) has the following characteristics, when the pedal is not depressed the output voltage is (1V) and when the pedal is fully depressed the output voltage is (4V).

The second potentiometer (sub) has the following characteristics, when the pedal is not depressed the output voltage is (0.43V) and when the pedal is fully depressed the output voltage is (2V) [43].

Finally the output voltage from the tow potentiometers goes to ECU that analysis the angle of the accelerator pedal that controls the air entering in gasoline engine and the fuel entering in diesel engine.

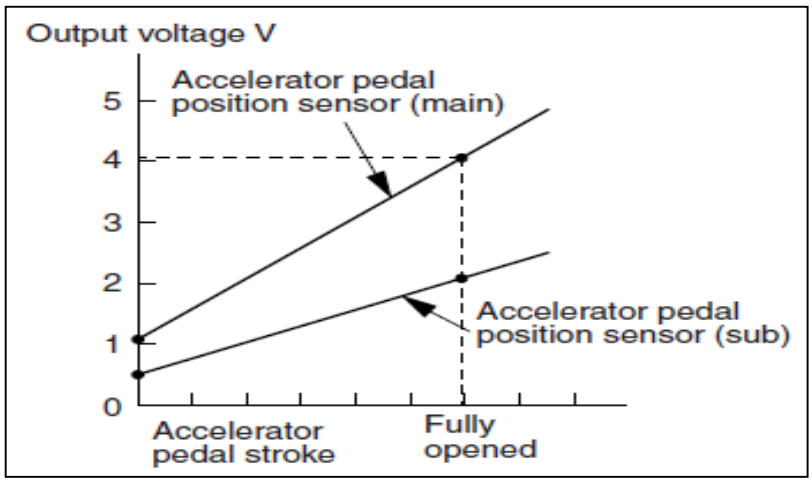


Figure 4.22: *APPS voltage output with accelerator pedal stroke*

The first step of reduce vehicle speed is objection the APPS output voltage by giving an ECU the voltage of the ideal position (accelerator pedal doesn't depressed), i.e. an ECU should receive the voltage of (1V) from the main potentiometer, and receive (0.43V) from the sub potentiometer, even that the accelerator pedal is depressed, this occur by using Analogue Digital CMOS SPDT Switch (ADG1419).

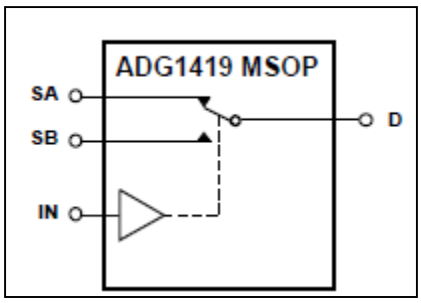


Figure 4.23: *Functional block diagram*

The above figure shows ADG1419 functional block diagram, IN input pin controls the switch to choose the output on D to be SA or SB, when IN logic is 0 the output on D is SA, and when the IN = 1 the output on D is SB [44].

EN	IN	Switch A	Switch B
0	X	Off	Off
1	0	On	Off
1	1	Off	On

Figure 4.24: *ADG1419 Truth table*

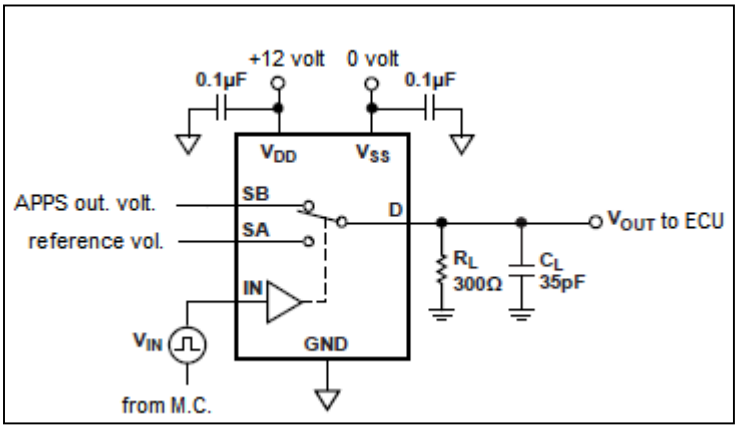


Figure 4.25: *ADG1419 Pin connection*

Due to lack of an ADG1219, The search for alternatives such as normal relay, which have the same characteristic but low response time, which is not significant.

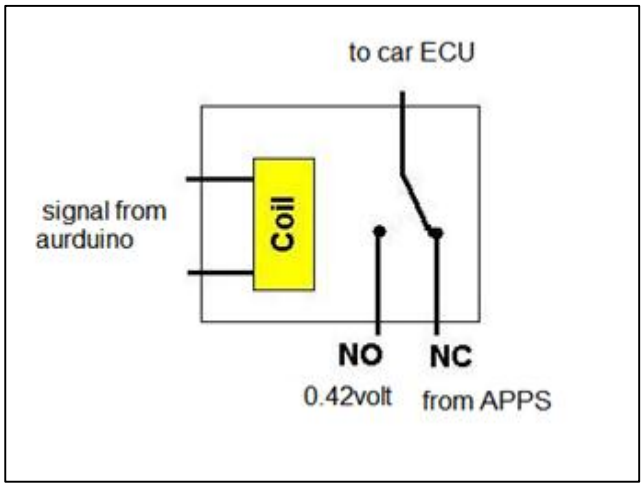


Figure 4.26: *Relay control APPS*

4.15 Brake actuator:

In order to reduce the speed of the car, the brakes should be activated, to a chive this goal the brake actuator will be used, by pulling the brake pedal when it receive signal from the controller.

In this system the choose of brake actuator is starter solenoid. The starter solenoid works as a powerful electric relay - when activated, it closes the electric circuit and sends the battery power to the starter motor. At the same, the starter solenoid pushes the starter gear forward to mesh with the engine flywheel. A typical starter solenoid has one small connector for the control wire and two large terminals: one for the positive battery cable and the other for the starter motor [5].

4.15.1 Starter motor

A starter is an electric motor that turns over or "cranks" the engine to start it.
A starter consists of the very powerful DC electric motor and the starter solenoid that is usually attached to the motor. Inside, a typical starter motor has the electric windings (coils) attached to the starter motor housing and the armature (the rotating part) that is connected through the carbon brushes in series with the windings. On the front end of the armature, there is a small gear that attached to the armature through an overrunning clutch [5].

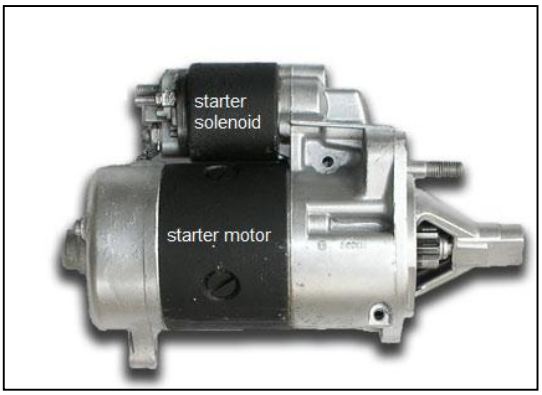


Figure 4.27: *Starter motor with starter solenoid*

4.15.2 Brake actuator installation

In the beginning of searching, team project try to find method and actuator that doesn't affect or opposed on the brake pedal normally operation.

A specimen is fixed perpendicular on starter solenoid shaft, which make pulling force on the brake pedal when receive signal from controller, in the same time, if the driver decide to depress on the brake pedal it doesn't oppose him, because the specimen is above the pedal.

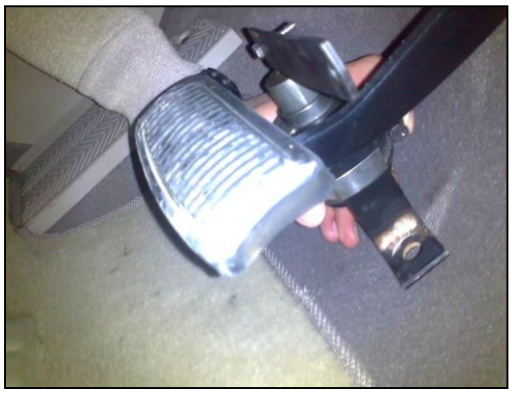


Figure 4.29: *Brake actuator installation*

Figure 4.28: *Different side of brake actuator*

The previous figures describes the installation of the brake actuator, and how does it never object on driver depressing on brake pedal, the vertical distance between the brake actuator hand and brake pedal can be adjusted.

It is important to know that the starter solenoid signal is (12V) DC, but output signal from the ordino controller is (5V), to solve this problem a relay will be inserted in the circuit, the signal from ordino are connect to coil of solenoid, the normally open switch when it close will be close the circuit and cause the brake actuator work, figure below shows the schematic diagram of relay and brake actuator.

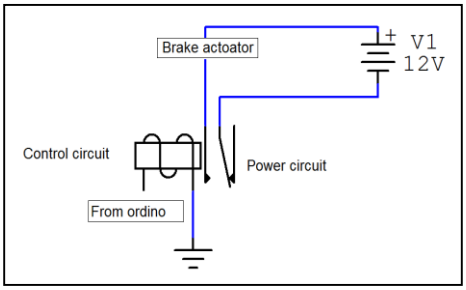


Figure 4.30: Brake actuator relay

4.16 Reduce speed flowchart and block diagram

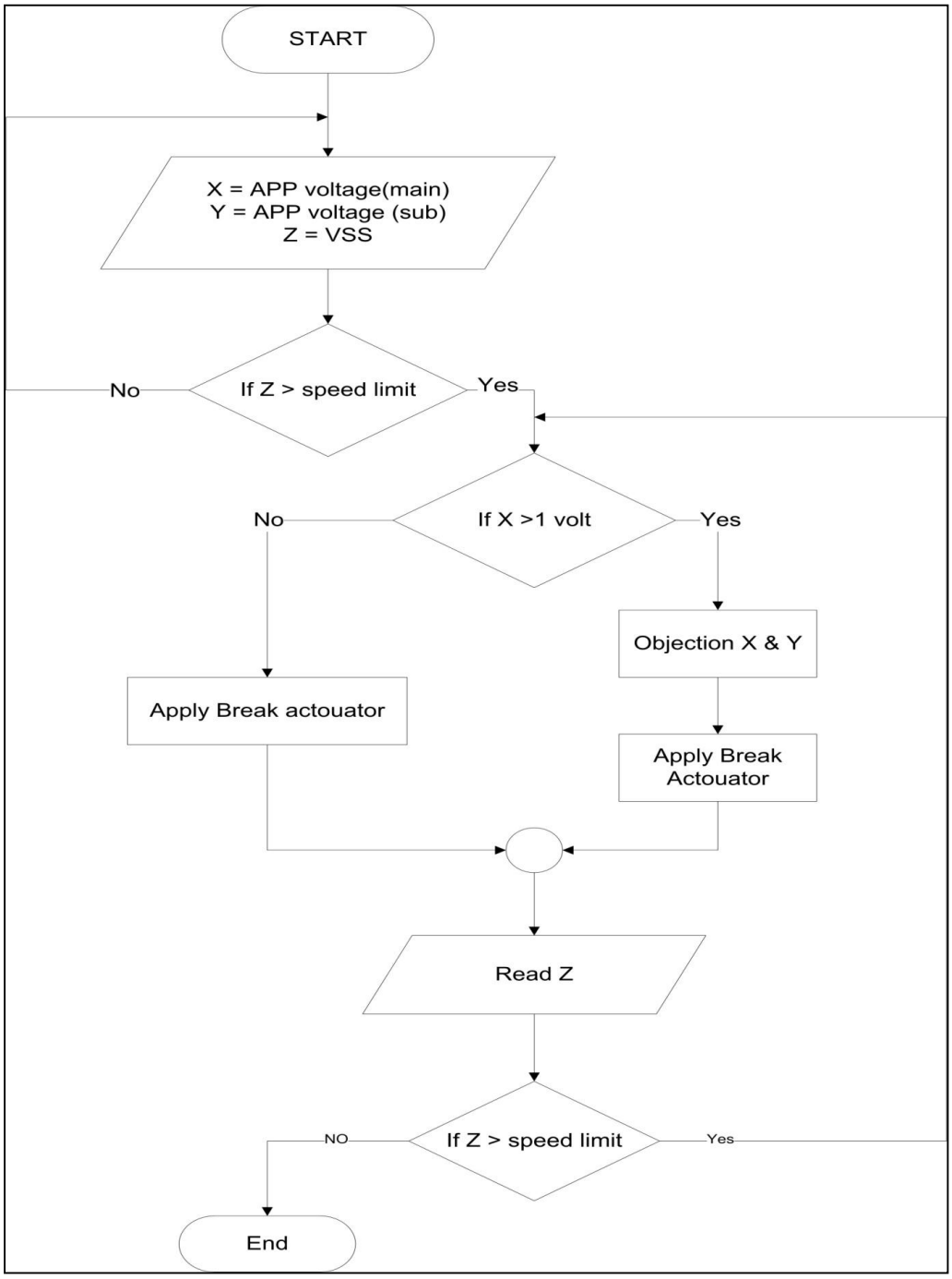


Figure 4.31: *Reduce speed flowchart*

5

Chapter Five

System Implementation

5.1 Introduction

5.2 Hardware system Implementation

5.3 software System Implementation

5.4 Arduino Software

Chapter 5:
System Implementation

5.1 Introduction

This chapter describes the hardware and software implementation of the system. The system can be divided into two main parts: first is the Implementation of the software functions, objects, and methods at the mobile side, and at the server side, then the second part is implementation of an entire interconnected set of components and software for the purpose of determining proper functions and achieving the desired goals of the system.

5.2 Hardware System Implementation

5.2.1 Electrical system implementation

The next figure shows the hardware component before connected together. **Android USB shield** used to connect Galaxy phone with Arduino microcontroller to charge the phone, and connect Bluetooth mate silver with Arduino. **Bluetooth mate silver** used to send the data from android to Arduino, while **Arduino** used to control the mechanical part. **USB cable** used to connect galaxy phone with Android USB shield. The **wires** used to connect Bluetooth mat shield with Arduino serial port. Finally, **Galaxy phone** the most important part, which has the system application.

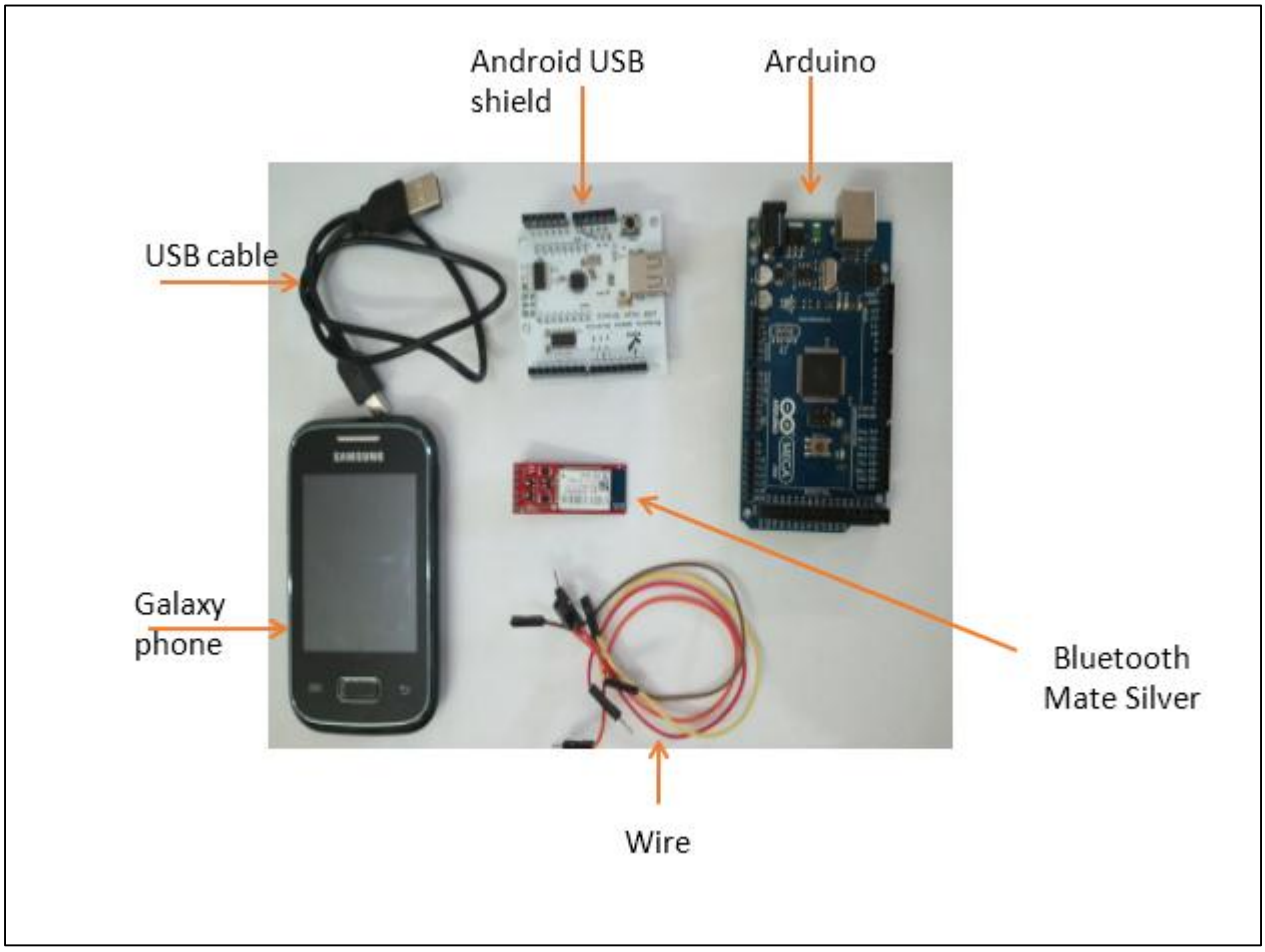


Figure 5.1: *Electrical subsystem components*

The connection between components without mechanical part shown in the next figure. Firstly, the galaxy phone make Bluetooth connection with Bluetooth mate silver and send data when violation occurred, after that, the Bluetooth silver mate sends the data to the Arduino by four wires, the first and second wires connect VCC and GND Bluetooth pins to VCC and GND Arduino pins for giving voltage to the Bluetooth mate silver, third wire used to connect the TX-0 Bluetooth pin to RX0 Android USB shield pin, and the fourth wire used to connect RX-1 Bluetooth pin to TX0 Android USB shield pin. The third and fourth wires are important to make connection between Bluetooth mate silver and Android USB shield for sending and receiving data between them. Finally, Arduino microcontroller do some tasks according to the data sent, in addition to make a connection and control in mechanical part, which will be explained later.

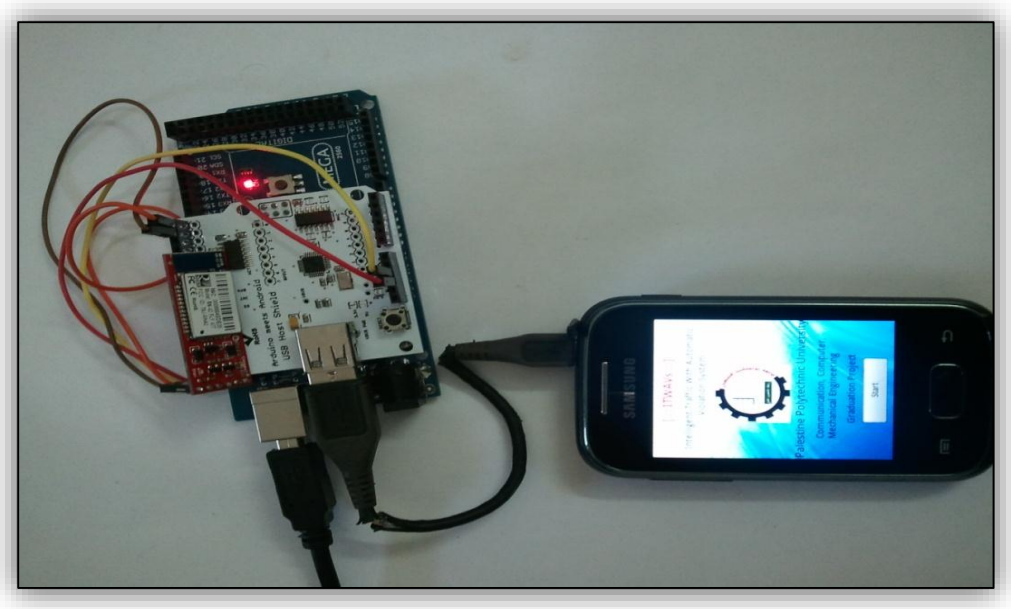


Figure 5.2: *Electrical system connection*

5.2.2 Mechanical System implementation

1. APPS

Returning the auto data and tolerance program to know the type of accelerator pedal, by insertion the specification of the vehicle.

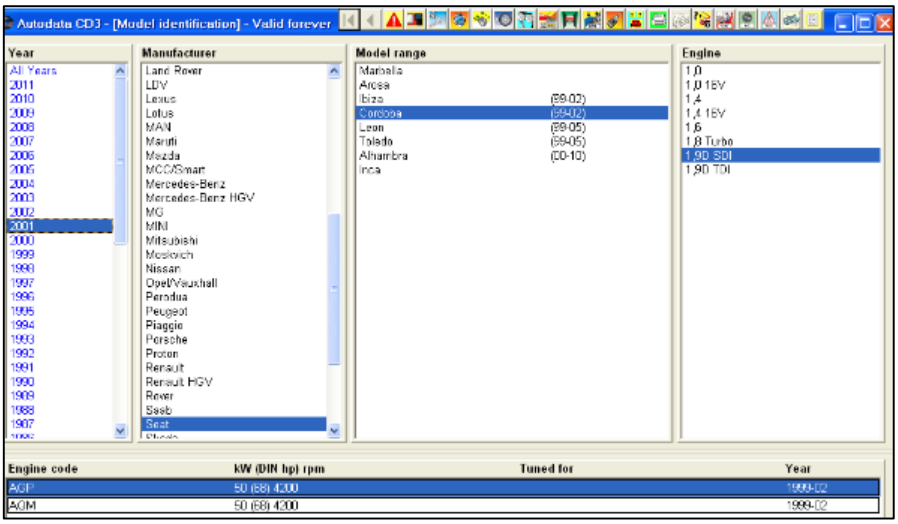
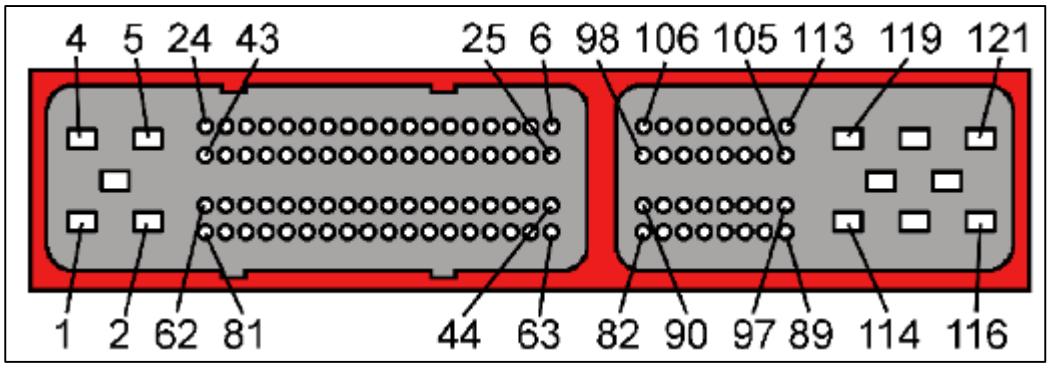


Figure 5.3: *Insert car specification in auto data*

After that go to electrical wiring diagram, showing the pins definition and test result to identify each wires in APPS.



Th

Figure 5.4: *Vehicle ECU pins*

Table 5.1: *Test result for APPS*

Component	From	To	Condition	Value
APPS	12 (-)	Battery	Ignition on,	5v
APPS	50 (-)	Battery	Ignition on,	0v
APPS	51 (-)	Battery	Ignition on,	0v
APPS	70 (-)	Battery	Ignition on, AR	0v
APPS	69 (-)	Battery	Ignition on, AR	0.1-0.5v
APPS	63 (-)	Battery	Ignition on, AR	0v
APPS	63 (-)	Battery	Ignition on, AFD	2.8v
APPS	70 (-)	Battery	Ignition on, AFD	2.8v
APPS	69 (-)	Battery	Ignition on, AFD	3-4.5v
APPS	54 (-)	Battery	Ignition on,	0v

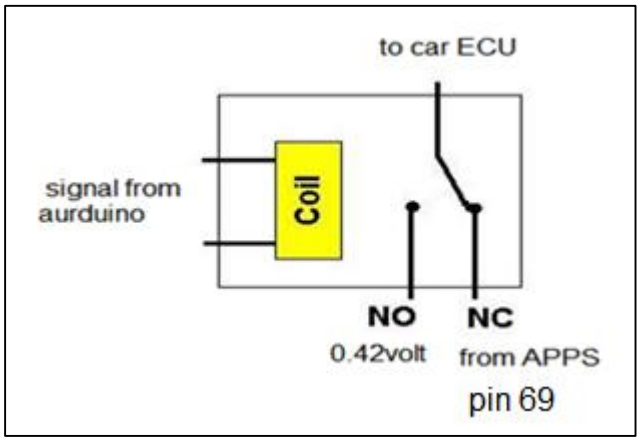


Figure 5.5: *Relay to control APPS*

Where:

APPS: accelerator pedal position sensor

AR: accelerator released

AFD: accelerator fully depressed

From this table we can note that the APPS has one potentiometer, pin 12 is supply voltage from ECU, pin 69 is the output from sensor to ECU, pin 63 and 70 for kick down position (kick down switch).

The first step in reducing the speed is to cancel the APPS operation, by giving the ECU an idle reading, this achieves by using relay, when relay receiving signal from Arduino it will click, the needle move from NC to NO position, which lead to entering voltage from output source, and the ECU read 0.43 volt which indicate idle position.

The following circuit used to control APPS, where the wires color meanings describes in the table 5.2.

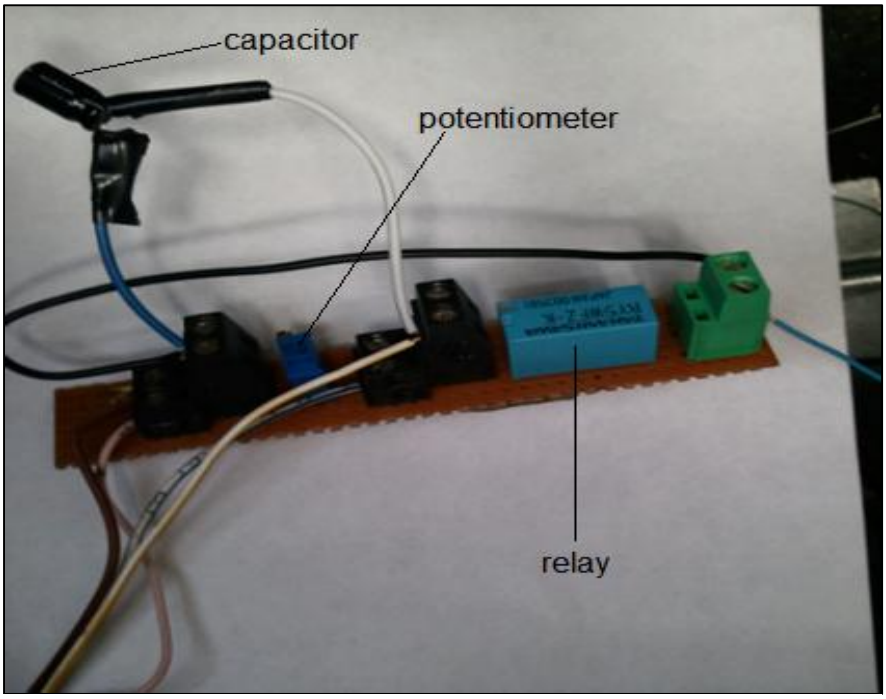


Figure 5.6: *APPS control circuit*

The potentiometer used to adjust the required voltage (o.43 volt), it connects with relay NO position.

Table 5.2: *APPS control circuit-wiring description*

wire color	*wire description*
Brown	Negative from car ECU
Pink	5 volt from car ECU
Blue/White	Output from APPS
Yellow/ White	Input to car ECU
Black	Negative for relay
Blue	Signal from Arduino

Where:

Blue: negative

White: input to car ECU

The first problem that we faced that when the accelerator pedal sensor reach signal from the Arduino (give 1 to relay), it still disconnected after that the Arduino give 0 to relay, the car ECU take an error from this sensor, and the accelerator pedal doesn't work (stuck in idle position).

By reference to the output signal from the APPS to the ECU, the signal change linearly as the driver depressed not step as relay do, to achieve this goals, a capacitor must be connected to the relay common.

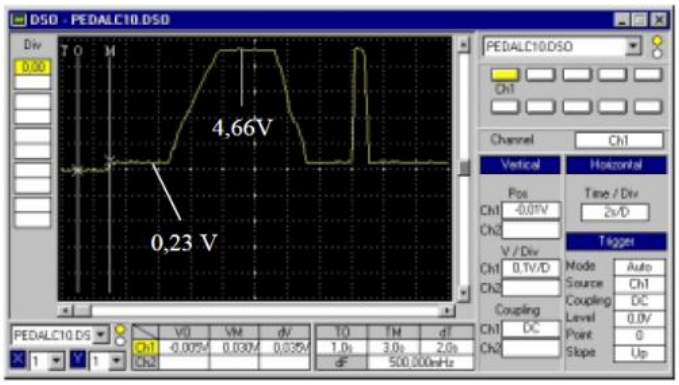


Figure 5.7: *Input Signal to ECU representation (Ignition on, pressing pedal and releasing)*

2. Brake actuator

The amount of speed that will be reduced not too high, and reach to zero speed not acceptable, so the actuator choose to be one-step to reduce the vehicle speed. The car was chosen so that it contains to brake assist system, which monitors the driver's use of the brake pedal, automatically sensing an attempt to stop the car because of panic. It then generates very high braking

power, even when the driver is only pressing lightly on the brake pedal. When this is used together with anti-lock braking systems, it results in faster and safer braking.

Note: 12-volt DC wire was taken directly from vehicle battery.

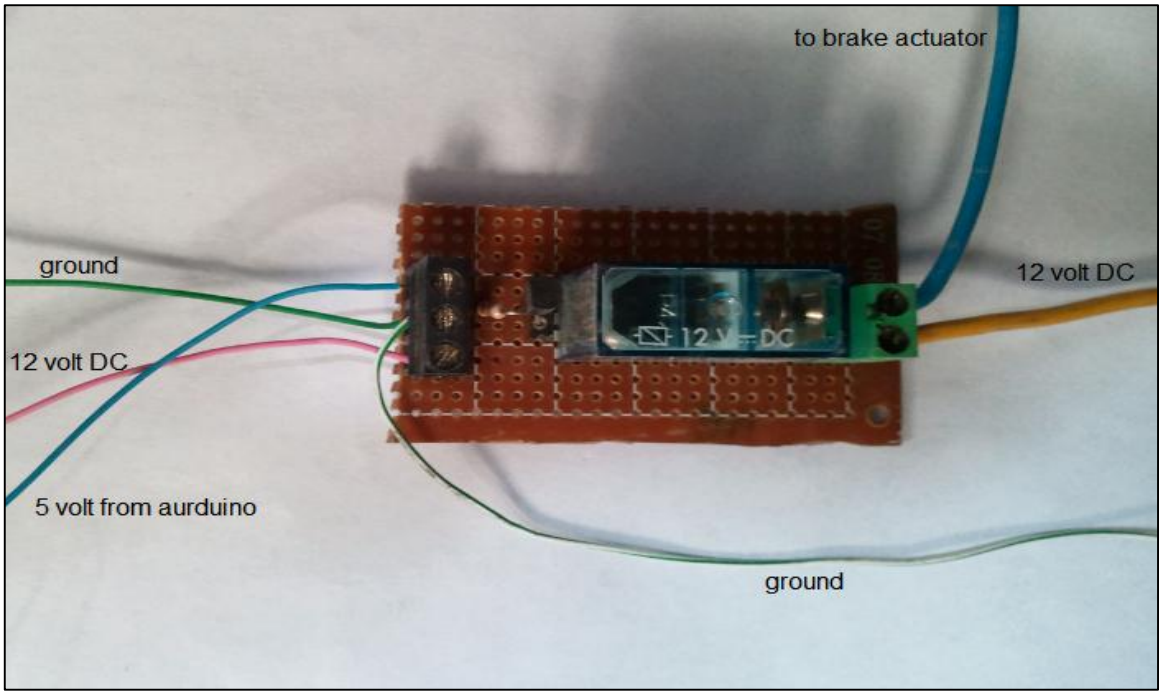


Figure 5.8: *Brake solenoid relay*

The brake solenoid operate with 12 volt DC, but the Arduino gives 5 volt signal, to connect the solenoid with Arduino a relay with transistor must be used, connect the transistor emitter with 12 volt, collector with 5 volt and base with ground.

When signal comes from Arduino, the transistor allow 12 volt to pass through relay coil, which lead to click relay and operate the break actuator.

Figure 5.9: *Brake solenoid installation*

For the installation of brake solenoid, base inserted under it to work efficiently. The two screws used for prevent the moving part get out from it place.

5.2.3 The whole system implementation

Whole system component connected together as shown in the next figure.

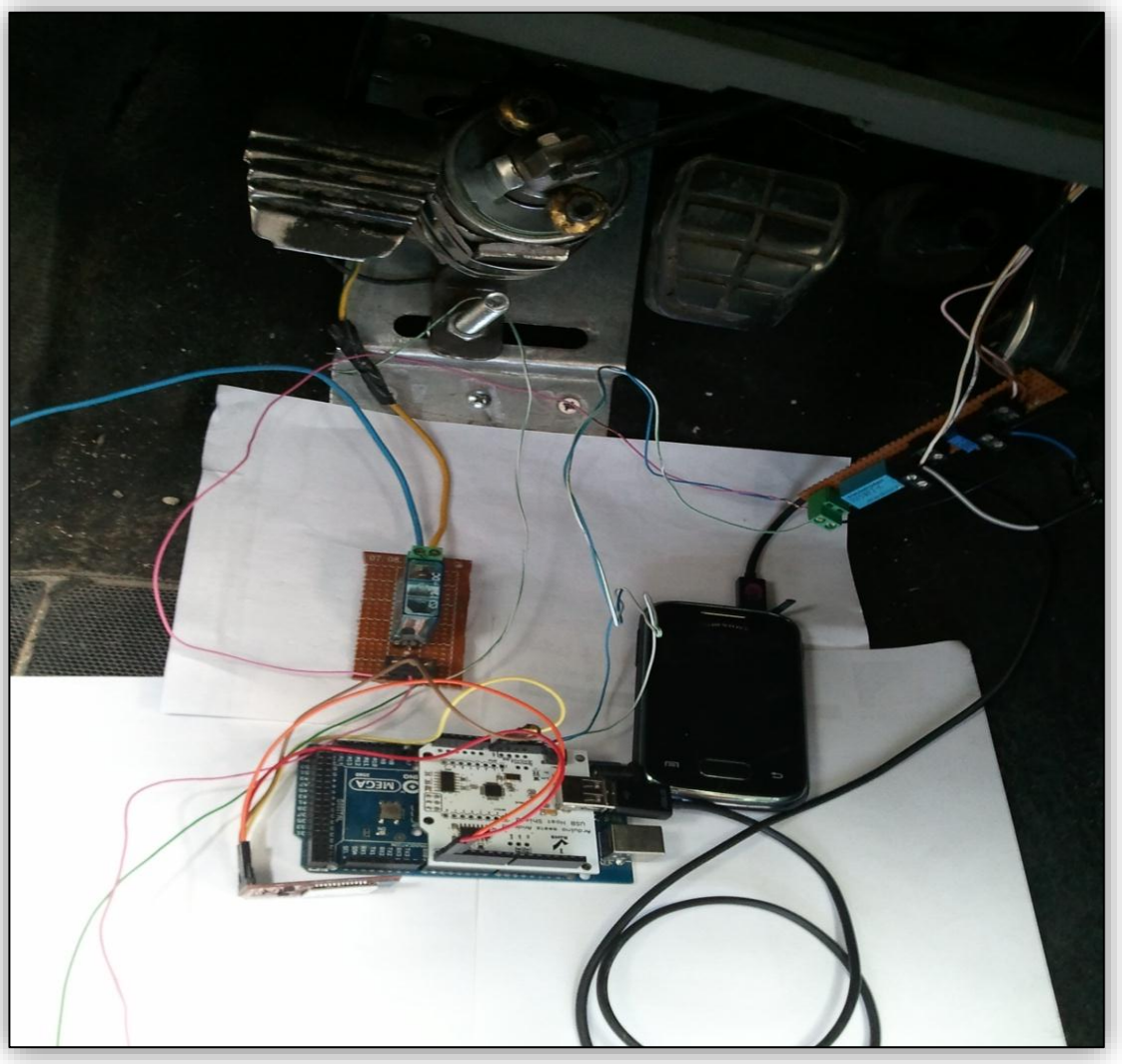


Figure 5.10: *ITWAVs system components interconnection*

5.3 Software System Implementation

5.3.1 Mobile Phone System Implementation

Galaxy S1 mobile phone is the main element which used to process the signals received from the GPS satellites, accelerometer signals for crashing detection sensor, and the Bluetooth

connection with the microcontroller when connection stage needed.

In this System, mobile phone will perform six main operations:

1) Run the system and insert the data needed to define drivers in the system by signing up new driver, sign in by setting username and password.
2) Getting the correct coordinates and speed by receiving GPS signals.
3) Searching in built in database about current street and its speed.
4) Checking that the car speed is within the speed limit, once the current car speed is greater than speed limit, the system gives a chance to the driver to reduce his speed by waiting few seconds, after that the system test again and if the driver doesn't reduce his speed the system enter to the fifth operation.
5) Sending SMS to the server and sending signals over Bluetooth to the microcontroller and reduce the car speed.
6) Detecting weather accidents happened or not, and sending accident location data to the first aid unit when accidents happened.

Mobile application has already been built using the Eclipse environment, to use this application it must be installed to the mobile, which will be performed by connecting the mobile phone (Galaxy S) by the USB cable to the PC at which the workspace of Eclipse is created. Then ".apk "file must be copied to the mobile SD card. Finally, the application must install to be ready to use by the driver.

5.3.2 Mobile Classes Implementation:

A detailed description for Mobile Application will be implemented by describing its main classes, methods, and attributes used in this system, where the mobile application classes are:

Hello class:

This class will be used when starting the application, as greeting screen, as shown in the following figure:

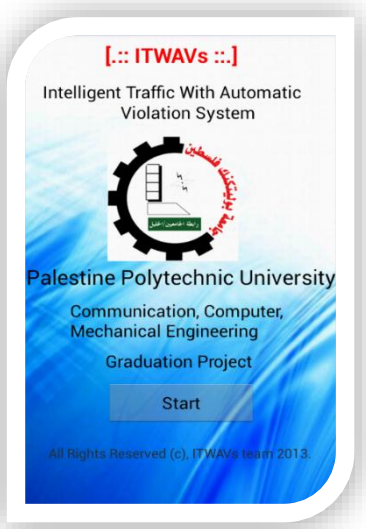


Figure 5.11: *Hello class*

The following objects are used in this class:

MediaPlayer: which is a media player object that plays music at the beginning of the application.

```
MediaPlayer.create(getBaseContext(), R.raw.roh);
mp.start();
```

onCreate: Called when the activity is starting, this is where most initialization should go.

```
super.onCreate(savedInstanceState);
```

Start buuton: this button used to enter the next step of this application, by activate intent which will move to the Welcome intent.

```
button1.setOnClickListener(new OnClickListener()
```

1. Welcome class

This class gives the user two options, admin and driver , where each option move the user to new screen as shown later, then after choose the required option, two buttons are used, one for move to the required class, or exit all application.

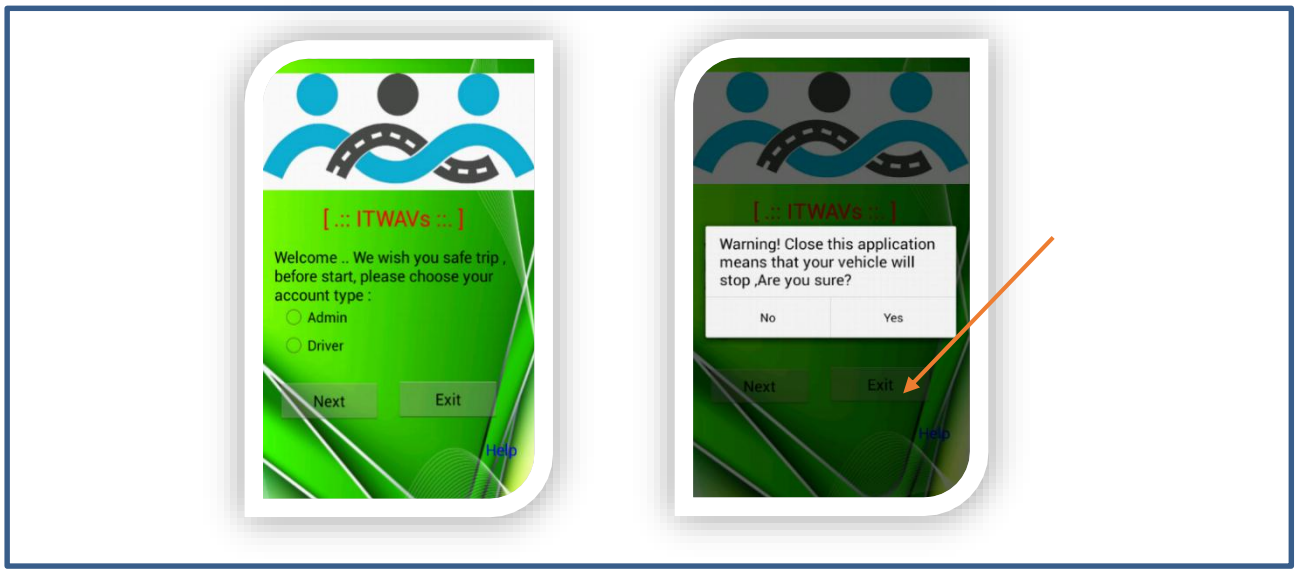


Figure 5.12: *welcome class*

The following objects are used in this class:

Radiogroup: this method used to make two choices, Admin or driver.

```
RadioGroup g1 = (RadioGroup) findViewById(R.id.radioGroup1)
g1.setOnCheckedChangeListener(new OnCheckedChangeListener()
```

DialogInterface: this method appears when Exit button clicked, and used to warn the user that the system will shut the vehicle down.

```
DialogInterface.OnClickListener dialogClickListener = new
DialogInterface.OnClickListener()
```

2. Register and Sign in class:

The main purpose of this activity is the car and application security, no one can move to the next activity and run the car without having account. The first time used this application the driver doesn't have account, so he must has the secret code to make new account. Once he enter the correct code, fill his new username and password and press button Sign up, and has new account. After that he can enter the next activity and run the car by filling the username and password field, and press the button sign in.

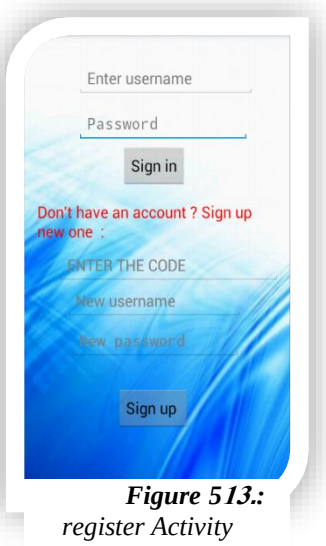


Figure 513.: *register Activity*

The following objects are used in this class:

Btnsignup: this button used to read input string on username and password edit texts, then open the users database, write the captured data, then close the database.

```
btnsignup.setOnClickListener(new OnClickListener()
```

Btnsignin: this button used to retrieve the drivers database, then compare the entered data with all database entries, until find the similar elements to move to the next intent.

```
Button btnsignin = (Button) findViewById(R.id.signin);
```

3. Main screen activity :

In this activity the system can get the car speed and location, determine in which street the car is located, get the speed limit and the name of current street, determine whether the car speed is higher than the speed limit or not, and has the ability to detect the accidents .

The following objects are used in this class:

LocationListener: used to get the GPS coordinates (longitude , latitude and speed).

```
class MyLocationListener implements LocationListener .
public void onLocationChanged(Location location)
public void onProviderDisabled(String provider)
public void onProviderEnabled(String provider)
public void onStatusChanged(String provider, int status,
Bundle extras)
public void onResume()
public void onPause()
```

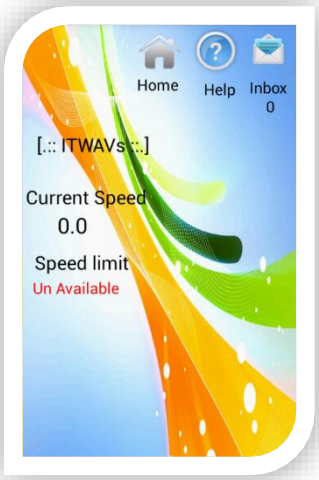


Figure 5.14:
Main screen activity

TextToSpeech: used for convert text to speech.

```
public void onInit(int status)
public void say(String string)
```

SensorManager : it communicates with the accelerator sensor which used for detect the accident once it happened.

```
private final SensorEventListener mSensorListener = new SensorEventListener()
public void onSensorChanged(SensorEvent se)
public void onAccuracyChanged(Sensor sensor, int accuracy)
```

In this class the system deals with GPS database. The next figure shows the SQLight GPS database.

Table 5.3: *Streets database inside mobile*

	speed	name	fl	fg	sl	sg
1						
2	40	Electrod Street	31.5487	35.0948	31.5513	35.09709
3	70	Rass Aljora	31.54782	35.09739	31.5525	35.09842
4	50	Ein sara1	31.5322	35.098	31.5396	35.0993
5	60	Ein Sara2	31.5422	35.0975	31.5478	35.0991
6	0	new	31.505499	35.08964	31.50584	35.08991
7	40	1	31.5074	35.08953	31.50948	35.09054
8	50	2	31.50609	35.08663	31.507709	35.08923
9	60	3	31.50609	35.0823	31.50764	35.08576
10	70	4	31.507699	35.08065	31.50843	35.081957
11	50	5	31.50877	35.080618	31.510848	35.08406
12	60	6	31.51104	35.08431	31.51284	35.08793
13	40	7	31.51331	35.08839	31.51434	35.090151
14	50	8	31.51044	35.09056	31.51458	35.09144
15	5	PPU	31.50529	35.08962	31.5072	35.09046
16	3	Home	31.64881	35.05879	31.64895	35.059
17	3	Home2	31.64892	35.05899	31.64901	35.05902
18	60	Bit-Ummar	31.62797	35.10141	31.63096	35.11509
19	70	Al-Eltifafy	31.60045	35.11004	31.61577	35.1145
20	65	Halhul	31.57171	35.09795	31.58826	35.10371

Where **fl** means the first latitude point in the street, **fg** the first longitude point in the street, **sl** the end latitude point in the street, **sg** the end longitude point in the street, **name** is the street name, and **speed** is the speed limit of the street .

```
	private final static String DATABASE_NAME = "mydb.db";
	private final static String DATABASE_PATH =
/data/data/com.example.ITWAVs/databases/";
	private final static String TABLE_GPS = "gpsdata";
	private final Context context;
	private DatabaseHelper DBHelper;
	private SQLiteDatabase db;
	public GPSDB open()
	public void close()
```

Once the car moves, the current speed field changes to the car speed, and when the car enter the available street that exist inside the database, the speed limit field is changing from *unavailable* to the street name and speed limit of the current street, as shown in next figure.

```
public Cursor getStreet(double lt, double lg) {
Cursor mCursor = db.query(TABLE_GPS, new String[] { "name",
"speed" },lt + ">= fl and " + lt + "<= sl and "  lg + ">= fg and
" + lg+ " <= sg", null, null, null, null);
if (mCursor != null) {
mCursor.moveToFirst();
      }
return mCursor;
             }
```

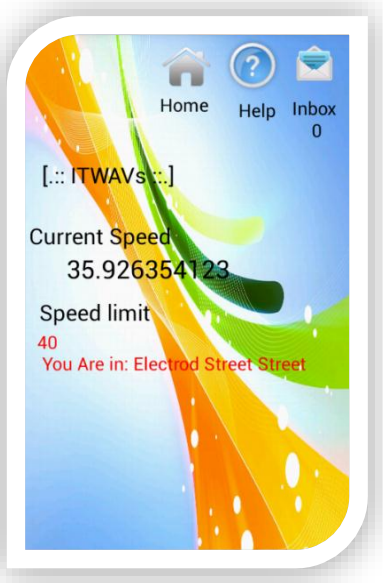


Figure 5.15:
reading database activity

When the driver exceeds the speed limit ,the application gives a five seconds to him to reduce his speed, and if the time is up and the car speed still higher than speed limit, the violation will be occurred, and the application will go to the SMS activity, and finally to the Bluetooth activity as shown in the next figure.

The *accidents detection* is the additional part in the main screen activity. When the accident happened, the accelerator sensor that is located inside mobile will detect it as shown in figure (5.16), then the application will go to the crashing activity which has a button called "*I'm OK*", when it pressed then no action happened, and after counter it will returned to the main screen activity, which mean that the problem which happened doesn't need ambulance, but if this button doesn't pressed, then after some counter, it will send SMS to the first aid center.

```
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE)
mSensorManager.registerListener(mSensorListener,mSensorManager.getDefaultSensor(Sensor.T
YPE_ACCELEROMETER),SensorManager.SENSOR_DELAY_NORMAL);
```

```
mAccel = 0.00f;
mAccelCurrent = SensorManager.GRAVITY_EARTH;
mAccelLast = SensorManager.GRAVITY_EARTH;
```

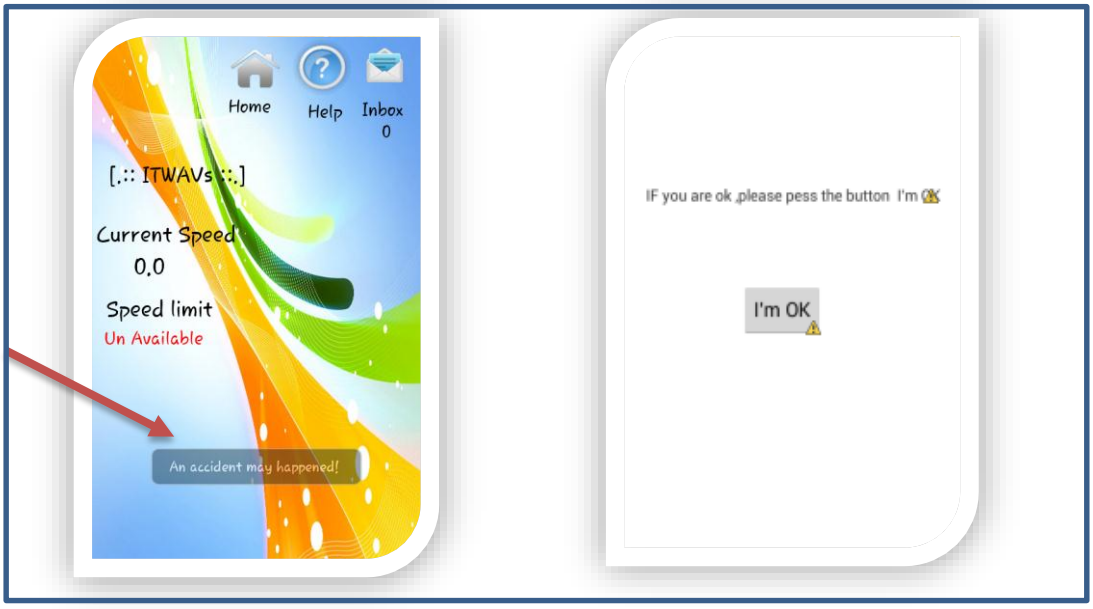


Figure 5.16: *Crashing system activity*

4. SMS activity :

This activity runs when the violation occurred. It will send violation data in specific format:

```
sendSMS(" SIM Number ", "987654321$22-2222-22$500$Hebron$Speed");
```

There are two filed in sending SMS : first one is the *"SIM Number"*, this number is the 3G modem number at the server that receiving the violation. And the second one is SMS massage data: driver identity number, car number, violation cost, driver live region, and violation type is the five component of SMS message, these components will fragment by insert dollar sign *"$"* between components, where the server knows that *"$"* means comma. Figure (5.17) shows the SMS activity.

The following objects are used in this class:

smsmanager: this method used to activate mobile GSM modem, and send SMS message.

```
      private void sendSMS(String phoneNumber, String message)
sendSMS("0597048646","987654321$22-2222-22$444$Hebron-
"+"mainscreen.mspeed.toString()"+"$Speed");
```

CountDownTimer: this is used to wait n seconds before move from SMS class to the Bluetooth class.

```
new CountDownTimer(3000, 1000)
```

Intent: this is used to move from SMS class to Bluetooth class.

```
      Intent I=new Intent(SMS.this,MainScreen.class);
      startActivity(I);
```

Figure 5.17: *SMS activity*

5. Bluetooth Activity:

In this activity the application sends signal to the microcontroller via Bluetooth. After sending violation SMS, the application must reduce the car in three steps: First one, the activity will make connection between mobile and Bluetooth modem, after that the Bluetooth activity sends signal to the Bluetooth modem. In the second step, the Bluetooth modem transfer the data signal to the Arduino microcontroller by serial connection between them. Finally, the data signal can control the car mechanical part, and reduce the car speed.

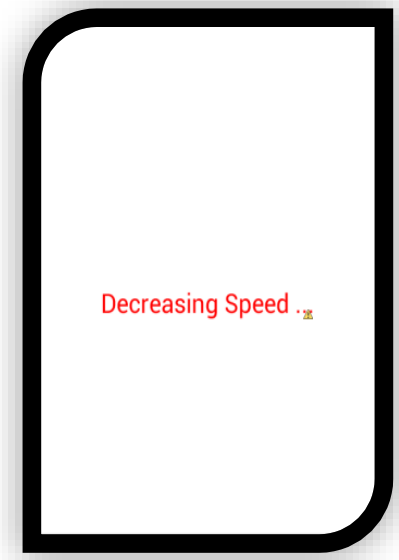


Figure 518.: *Bluetooth activity*

The following objects are used in this class:

BluetoothSocket : used to open Bluetooth socket to send data from android device to Bluetooth modem .

```
      private BluetoothAdapter btAdapter = null;
```

```
private BluetoothSocket btSocket = null;
private OutputStream outStream = null;
private static String address = "00:06:66:4E:DE:35";
private void sendData(String message)
```

CountDownTimer: used for waiting n seconds before move from Bluetooth class to the Main screen class.

```
new CountDownTimer(8000, 1000);
```

Intent: used to move from Bluetooth class to Mainscreen class .

```
ntent i= new Intent(Bluetooth2.this,SMS.class);
startActivity(i);
```

5.3.3 Server System Implementation

Server application has already been built using c# programming language; there are five main operations in the system:

1. Initialize the GSM Modem and connect using Serial port.
2. Receive violation form client and generate reports.
3. Add, Edit, and Delete users from the system.
4. Search about users in the system.
5. Pay violations from the system and generate the recipient voucher.

A detailed description for the server will be implemented by describing its main methods and attributes used in this system, where the server's application forms are:

1. **Welcome Form :**

This form will be used when starting the application, as greeting screen, as shown in the following figure.

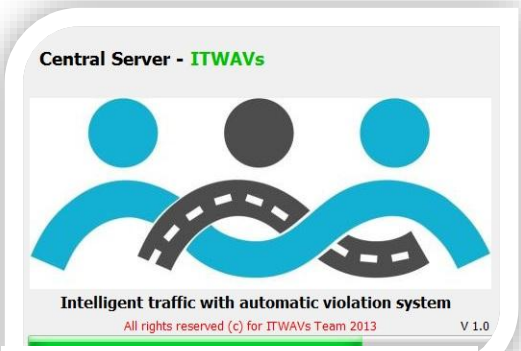


Figure 519.: *ITWAVs Server System Welcome Screen*

The following objects are used in this Form:

Progress Bar : used to open database for the first time .

2. **Login Form :**

This form will be used to enter username and password to sign in to the system, or register new account, as shown in the figure (5.20) .

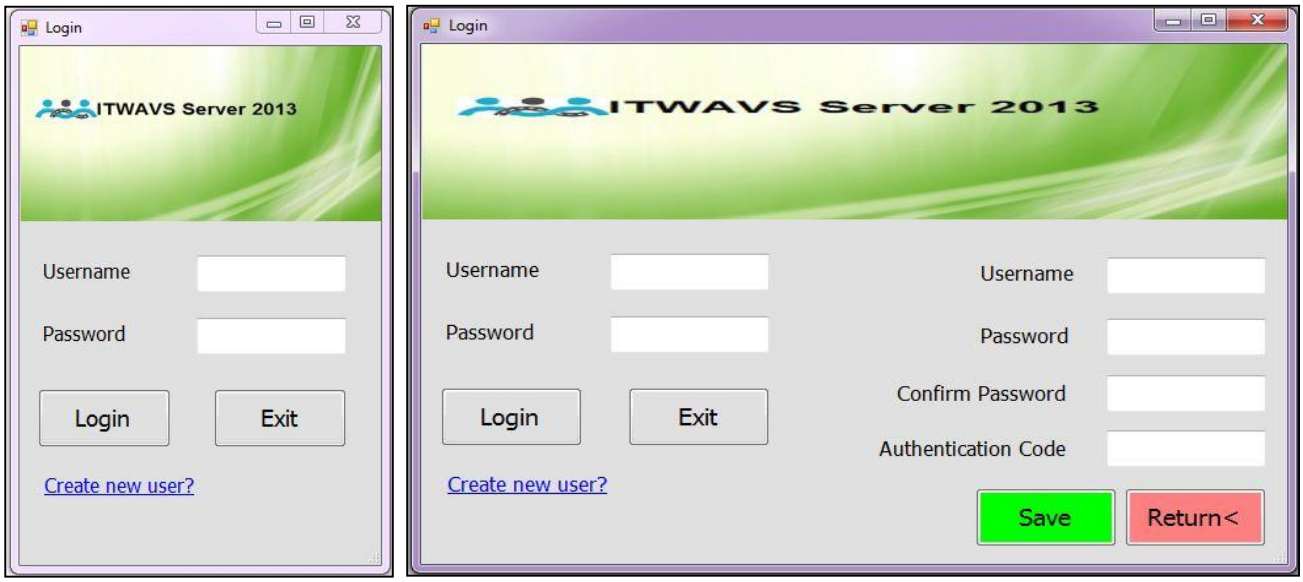


Figure 5.20: *ITWAVS Server System Log In Screen*

The following objects are used in this Form:

Login : this method used to get the username, password and sign in to the system.

create new user : this method used to get the collection of information about the new user and create new account in the system.

3. Main Form :

This form will be used to reach all processes created in this system, as shown in the next figure.

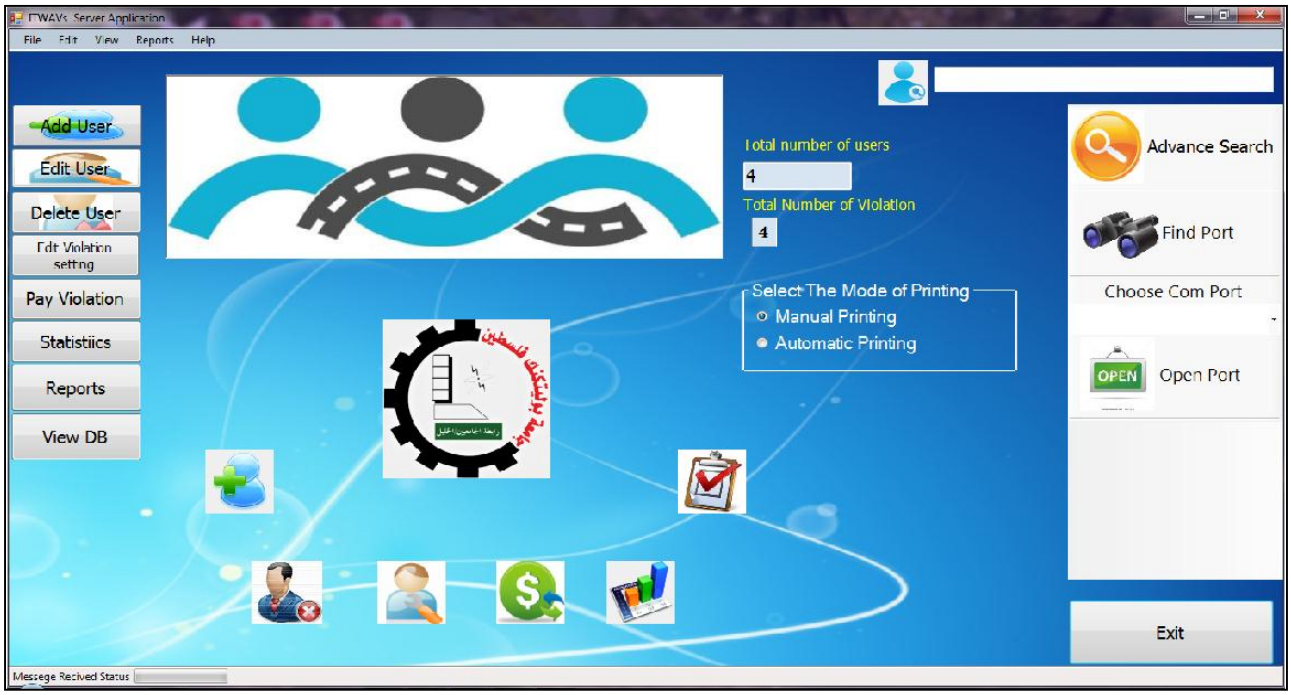


Figure 5.21: *ITWAVS Server System Main Screen*

The following objects are used in this Form:

Find port: this method used to search for all available serial port for GSM modem.

Initialize port: this method used to open the selected serial port that connected to GSM modem, and initialized the GSM modem to receive SMS.

Printing Mode: this method has two option, one for manually printing as pdf file, and the other for automatically printing directly on printer.

serial manager : this class used to listen on the opened serial port, send AT command to GSM modem, receive the acknowledgment form GSM modem, generate the violation, and store it in data base.

DataBase interface: this class used to connect to data base, have many method to: add, edit, delete, view, for the all the object in the data base.

The default mode of printing violation reports is manual printing, which is generate a PDF reports and store it in hard disk, you can change the mode of printing by select the mode from the main screen, when select automatic printing, a message box will be appear to notifying about the default printer, if press ok, automatic printing will be activated, as shown in figure 5.22,and figure 5.23 :

Figure 5.22: *ITWAVS Server System Select Printing mode*

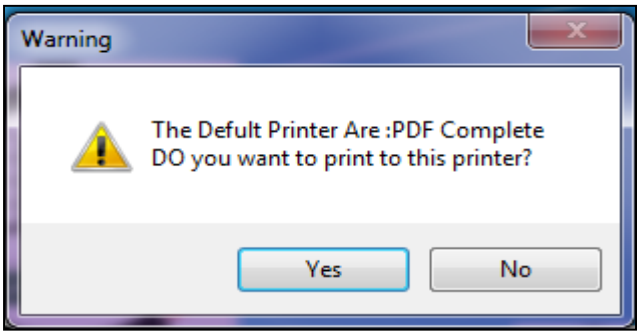


Figure 5.23: *ITWAVS Server System Message box default printer*

After finishing initializing GSM modem, and select the printing mode, the ITWAVs server well be ready to receive violation form client.

4. Add Vehicle Form :

This form used to add new vehicle to the system, by entering the owner and vehicle information, as shown in the next figure :

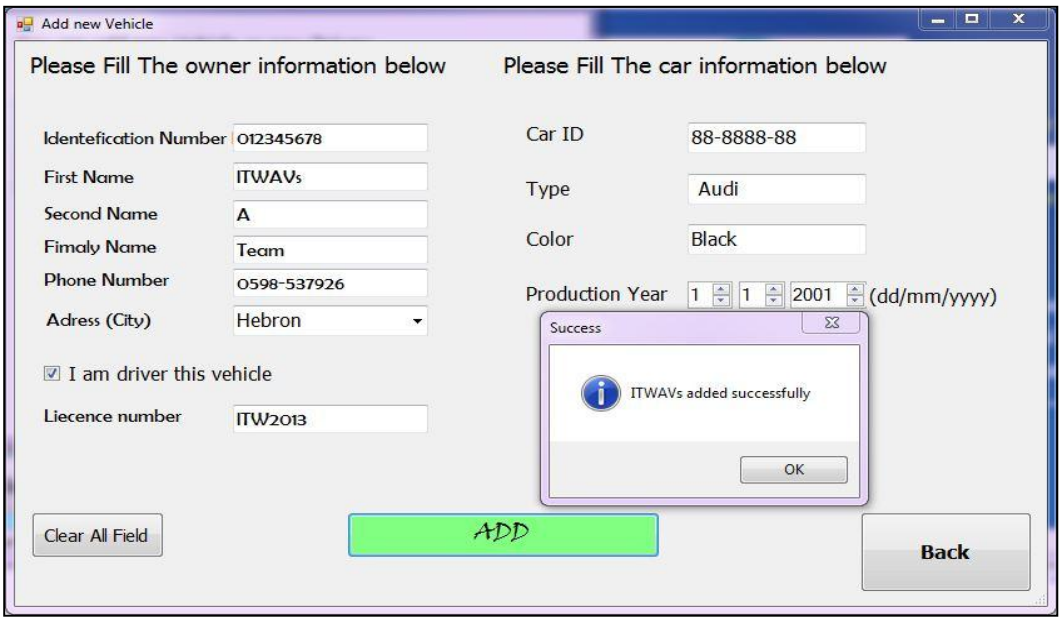


Figure 5.24: *ITWAVS Server System Add new vehicle*

The following objects are used in this Form:

clear all field : this button used to clear all field in the form from data.

add vehicle : this method used to get owner and vehicle information, check the validity of information, insert new vehicle to database with all insertion operation necessary.

ID Checker: this method invoked automatically when any changes occur in field identification number that allow the user to enter only nine numeric number.

5. Add Driver Form :

This form used to add new driver to the system, get information from field, as shown in the next figure:

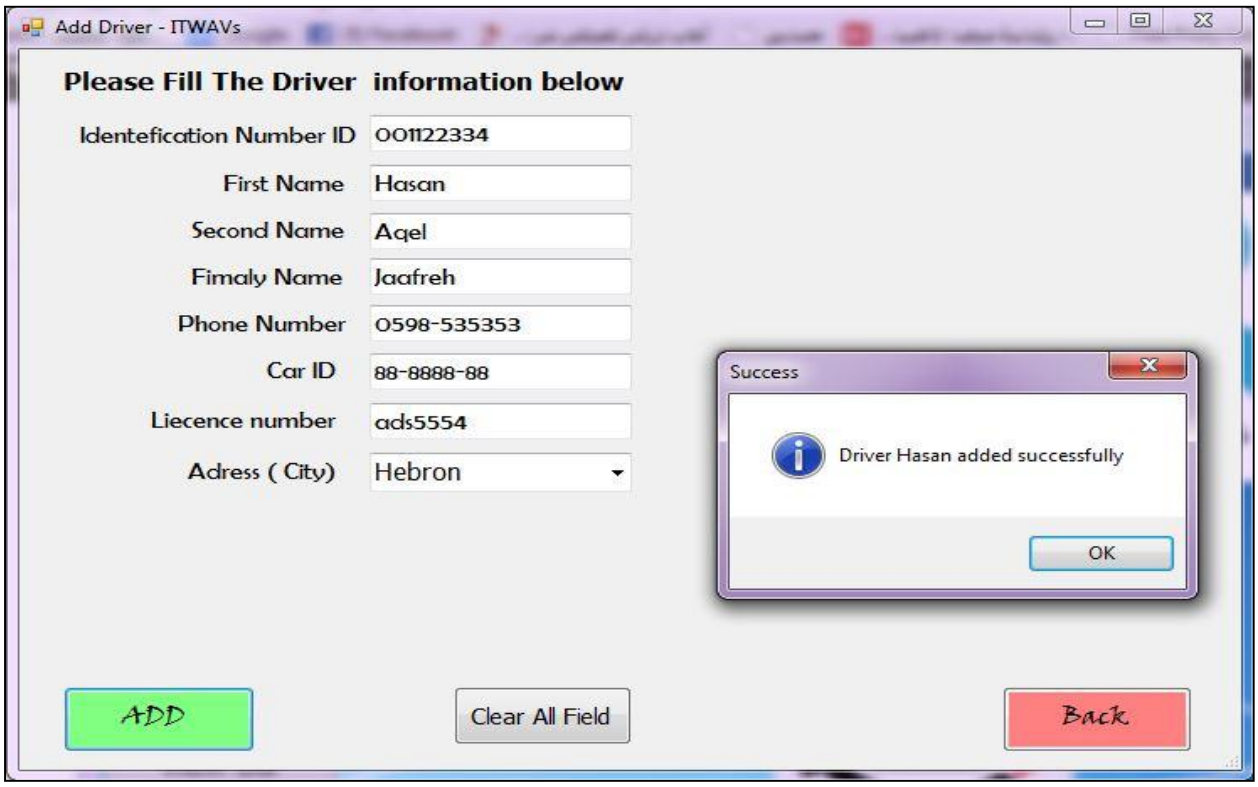


Figure 5.25: *ITWAVS Server System Add new driver*

The following object are used in this form :

clear all field : this button used to clear all felid in the form from data.

Add Driver : this method used to add new driver to the system, collect necessary information to check the validity of the data are entered by the user.

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

6. **Edit user Form :**

This form used to modify existing users' information in the system, modify existing vehicle's information in the system, get information from field, as shown in the next figure:

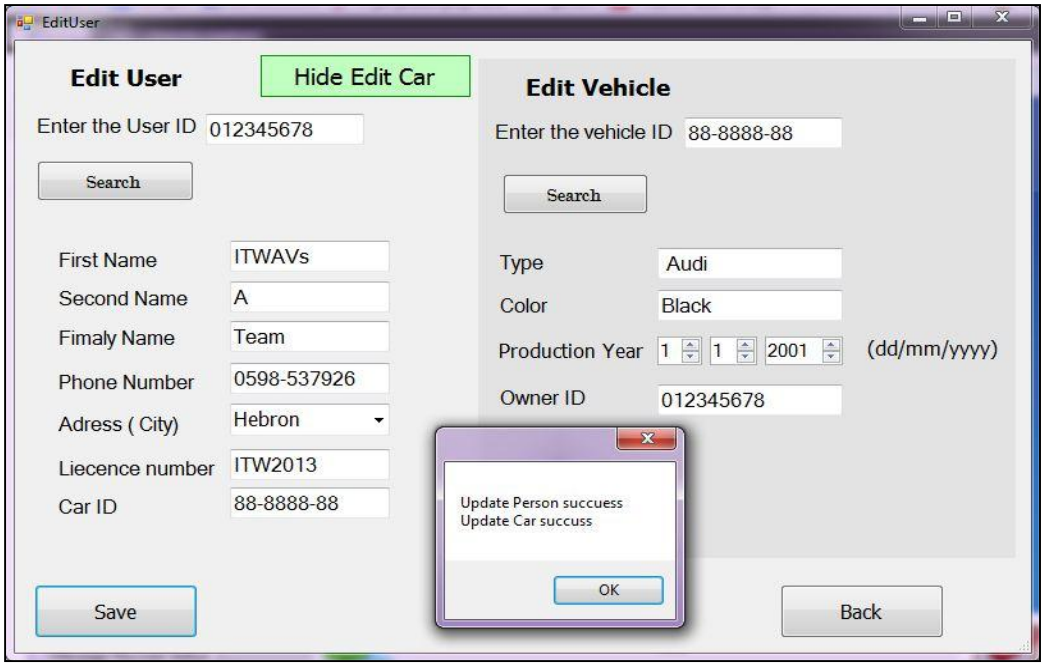


Figure 5.26: *ITWAVS Server System Edit users*

The following object are used in this form :

user search : this method used to search for users using user id and return the result on the specify field.

vehicle Search : this method used to search for vehicle using vehicle id and return the result on the specify field.

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

update : this method used to collect and check the user and vehicle information and update the user and vehicle information.

7. Delete user Form :

This form used to Delete existing users in the system, allow you to select the mode of user : driver or owner, in the both mode, enter the user id, and if owner is selected enter the car id. the system can delete only the users with no violation Paid violations, as shown in the next figure:

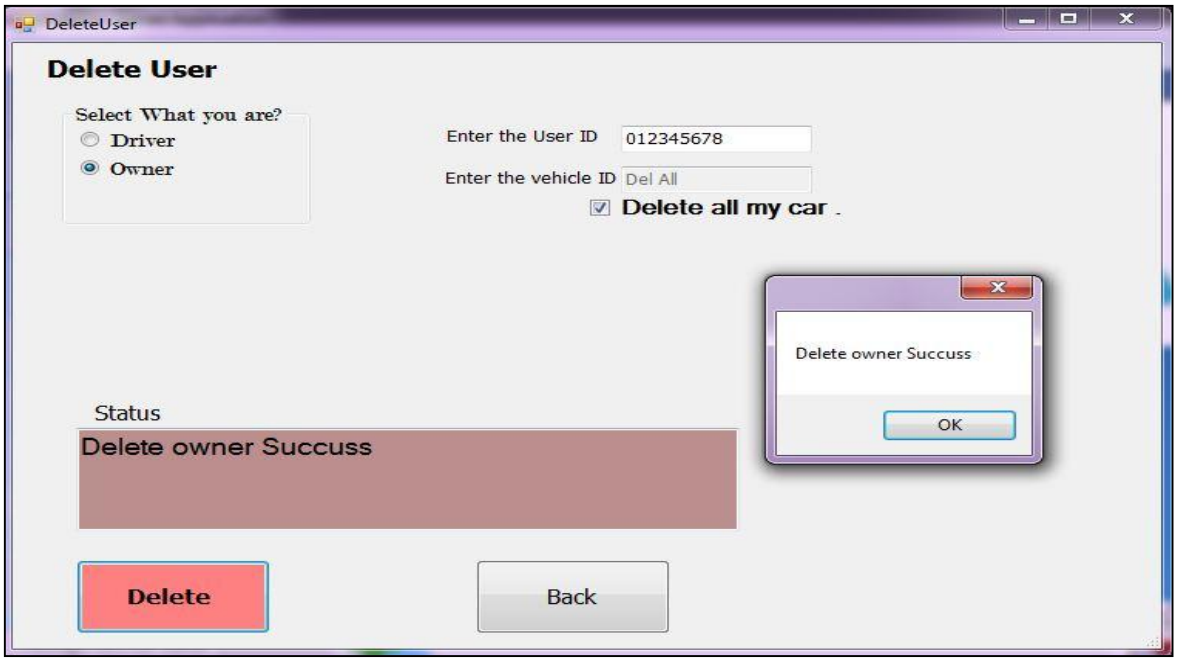


Figure 5.27: *ITWAVS Server System Delete Users*

The following object are used in this form :

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

Delete method : this method used to check if the user are found, and to delete the selected user from the system if the user have no violation or all violation are paid.

8. Pay Form :

This form used to permit you to enter the user id and vehicle id, after search for users' violation, select the violation to pay it, you can save receipt voucher and print it, as shown in figure 5.28 :

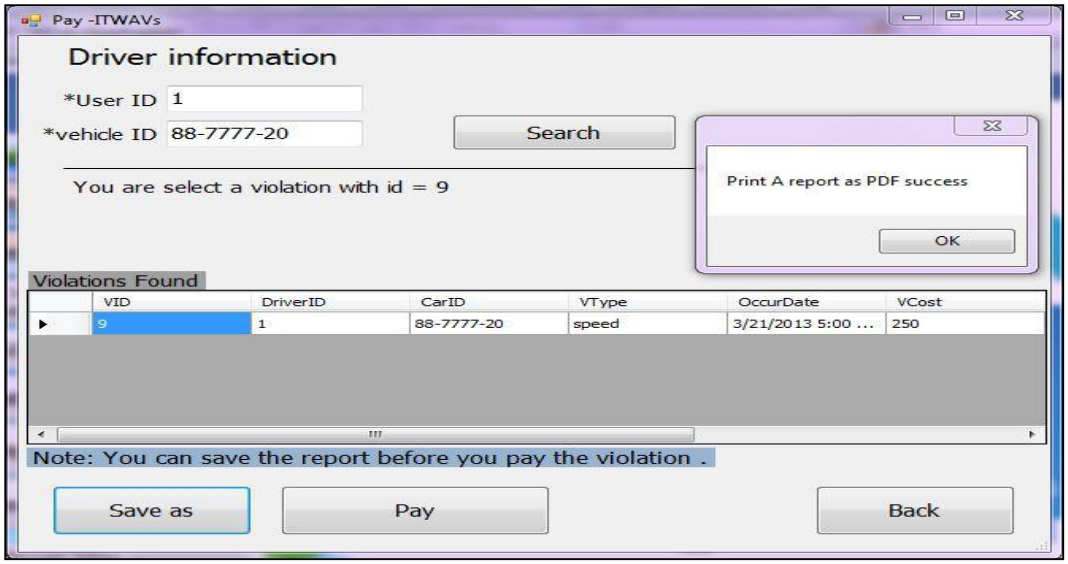


Figure 5.28: *ITWAVS Server System Driver information*

The following object are used in this form :

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

search: this button used to search for violations are not paid for the specify user and vehicle and view it's on form.

Save as : this method used to generate receipt voucher for the selected violation and store the receipt voucher as pdf file in the selected folder.

Pay : this method used to pay the violation, generate receipt voucher and directly printing to printer, and set the violation as paid.

9. Statistics Form :

This form used to view Statistics about violation and street, you can select the city and find the statistics about the city : how many violation occur in the city , how many user in the city, most street danger, and general statistics about the system, as shown in figure 5.29 :

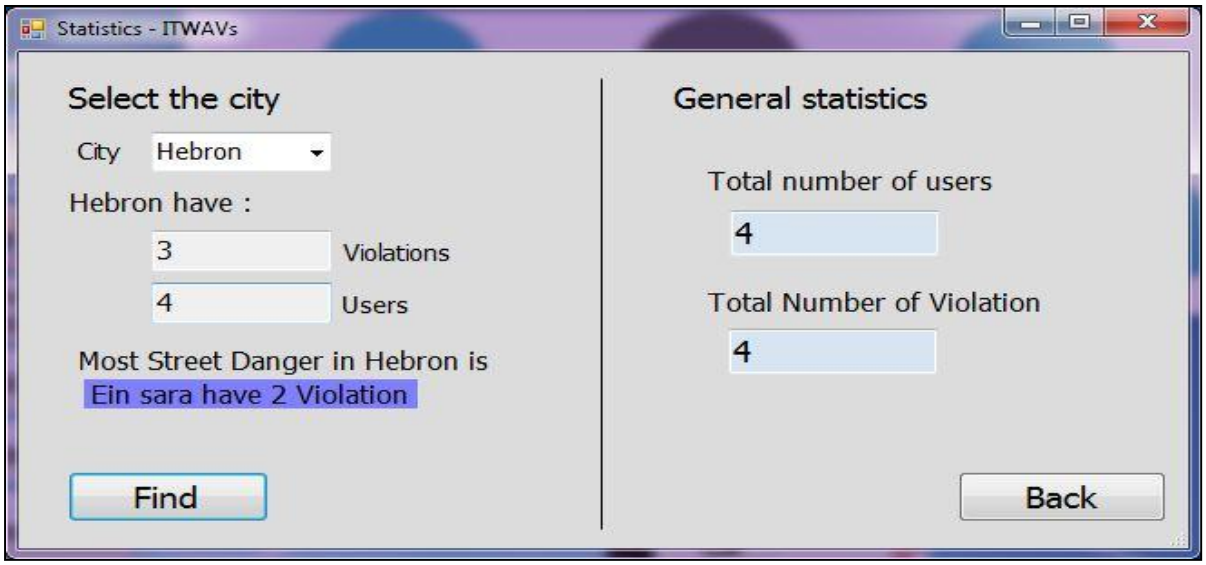


Figure 5.29: *ITWAVS Server System Statistics*

The following object are used in this form :

Find City : this method used to find all supported city in the system, and all supported street in the system.

statistics : this method used to find how many violation, user and most street danger in the selected city, in additional to find total number of user and violation in the system.

10. Report Form :

This form used to open the archive traffic violations and view, print earlier reports, this form permit you to enter the driver id and search, then select the violation, you can save it as PDF report or direct print on printer, as shown in figure 5.30 :

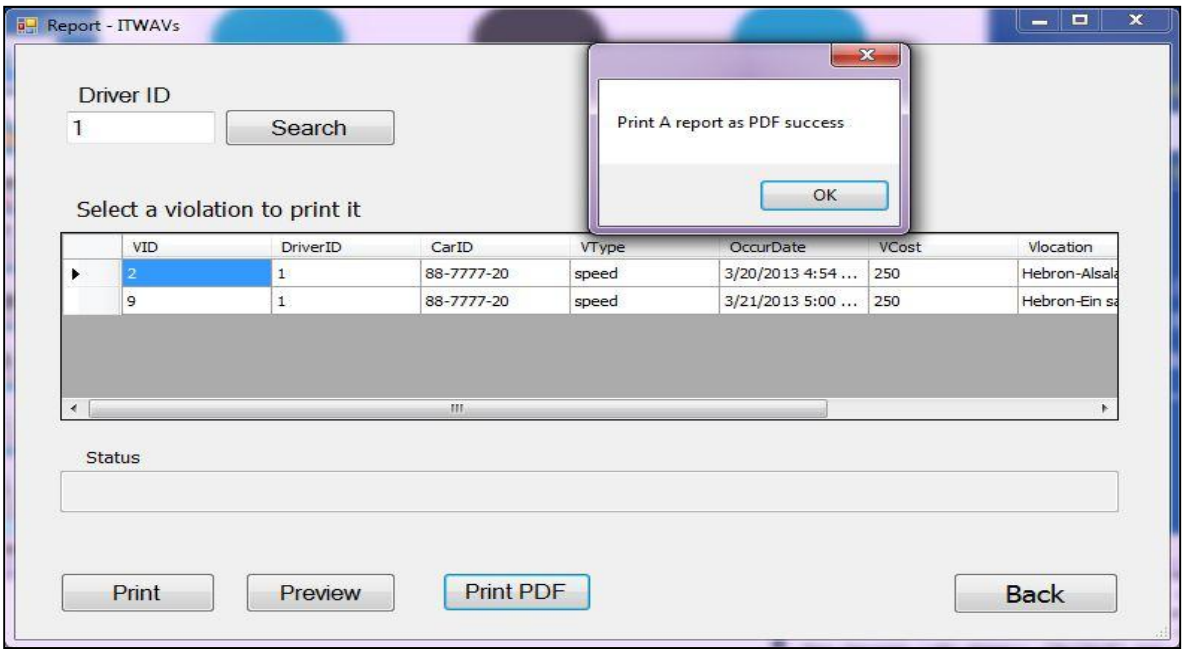


Figure 5.30: *ITWAVS Server System Reports*

The following object are used in this form :

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

search : this method used to search for violation using id, and view it.

report generator : this method used to generate a report for the selected violation.

print : this button used to print the report for the selected violation on the printer.

preview : this button used to preview the report for the selected violation.

printPDF: this button used to save report for the selected violation as PDF file in the ITWAVS folder.

11. View Form :

This form used to view database, view tables, show you data base tables', as shown in figure 5.31:

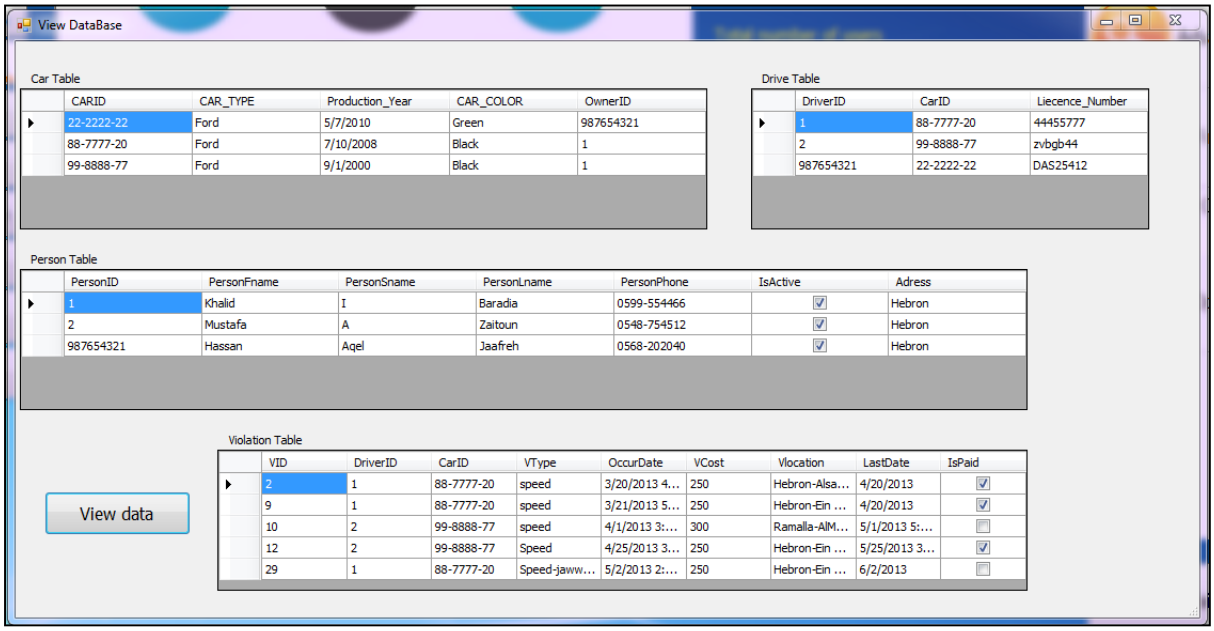


Figure 5.31: *ITWAVS Server System View Data Base*

The following object are used in this form :

View Data : this button used to get all data from data base and view it on tables in this form.

12. Find User Form :

This form used to search for users, show the user information and have button that well go to advance search, as shown in figure 5.32:

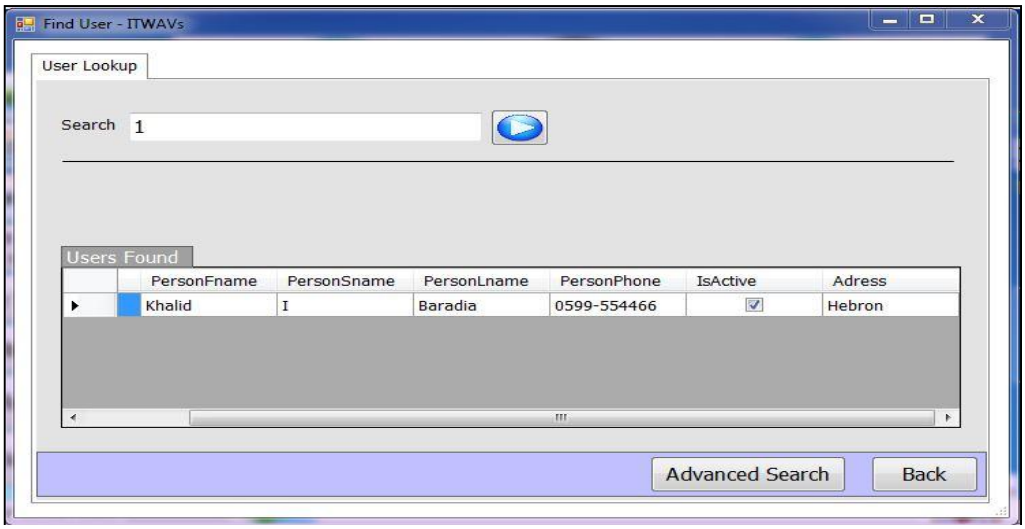


Figure 5.32: *ITWAVS Server System Find user*

The following object are used in this form :

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

search : this button used to search for user using user id and view the result in data grid view.

advanced search : this button used to move to the advanced search.

13. Advance Search Form :

This form used to search for user, vehicle, violation, this form allow you to select the type of search, enter the id, and view the result, as shown in figure 5.33 :

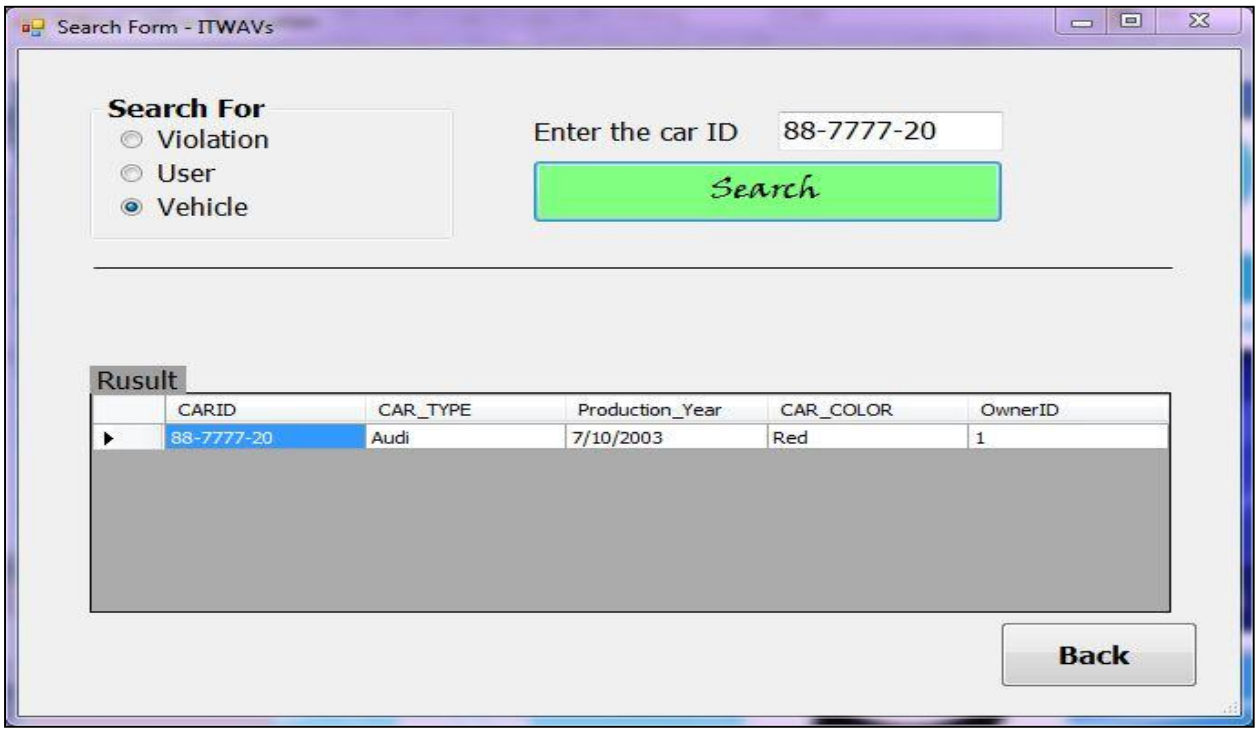


Figure 5.33: *ITWAVS Server System Advance Search*

The following object are used in this form :

ID Checker: this method invoked automatically when any changes occur in field identification number, that allow the user to enter only 9 numeric number.

search : this button used to search for user, violation, vehicle, depend on the search mode, and view the result on data grid view.

5.4 Arduino software

There are two main operations in the system. The first one is receiving the data coming from mobile via Bluetooth. Second is dealing with the hardware component to reduce the car speed.

```
char incomingByte;  // incoming data
int  LED = 13;      // LED pin

void setup() {
  Serial.begin(9600); // initialization
  pinMode(LED, OUTPUT);
  Serial.println("Press 1 to LED ON or 0 to LED OFF...");
}
void loop() {
  if (Serial.available() > 0) {  // if the data came
    incomingByte = Serial.read(); // read byte
    if(incomingByte == '0') {
       digitalWrite(LED, LOW);  // if 1, switch LED Off
       Serial.println("LED OFF. Press 1 to LED ON!");  // print message
    }
    if(incomingByte == '1') {
       digitalWrite(LED, HIGH); // if 0, switch LED on
       Serial.println("LED ON. Press 0 to LED OFF!");
    }
  }
}
```

6

Chapter Six

System Testing and Performance

6.1 Overview

6.2 Sub-System testing

6.3 Installation and preparing the system

6.4 Testing scenarios

Chapter 6:
System Testing and performance

6.1 Overview

In this chapter, the whole testing stages will be described, including a test of an entire interconnected set of components and software, for the purpose of determining proper functions and achieving the desired goals of the system. We will talk about testing of each part of the system, and describe its scenarios.

6.2 Sub-System Testing

The main aim of this testing part is to test the main operations in the system. There are five main operations in the system. The first one is to enter the system by setting username and password, second is getting the correct coordinates and speed, then the third is searching about current street and speed limit in database. The fourth operation is testing that the car speed is within speed limit, once the car speed greater than speed limit, the system gives a chance to the driver to reduce his speed by waiting few seconds, after this the system test again and if the driver doesn't reduce his speed, the system enters to the fifth operation is sending violation SMS to the server via GSM network, and send signal to the microcontroller via Bluetooth for doing some tasks that will be explained later, sixth operation is detecting weather the accident happened or not, and if it's happened the system sends accident location data to the first aid unit.

6.3 Installation and preparing the system

At this stage, all system parts must be ready to be tested at the two sides, mobile application at the car, and the desktop application at the server.

6.3.1 Mobile application

An application has already been built using the Eclipse environment, to use this application, it must be installed at the mobile, by connect the mobile phone (Galaxy S) by the USB cable connection to the PC at which the workspace of Eclipse is created, then the following file must be copied to the mobile SD card. C:\workspace\ITWAVS\bin\ITWAVS.apk .Open the ".apk" file, a message will appear at the screen of the mobile as shown in figure(5.1), show the name of the application and the permutations needed for it to works probably. In our application we need to access the GPS location to get the location of the car, and we need a full internet access for the communication between mobile (client) and PC (server) while posting data.

Click the install button to install the application as shown at figure(6.1), then when the installation is completed, click the open button to start the application.

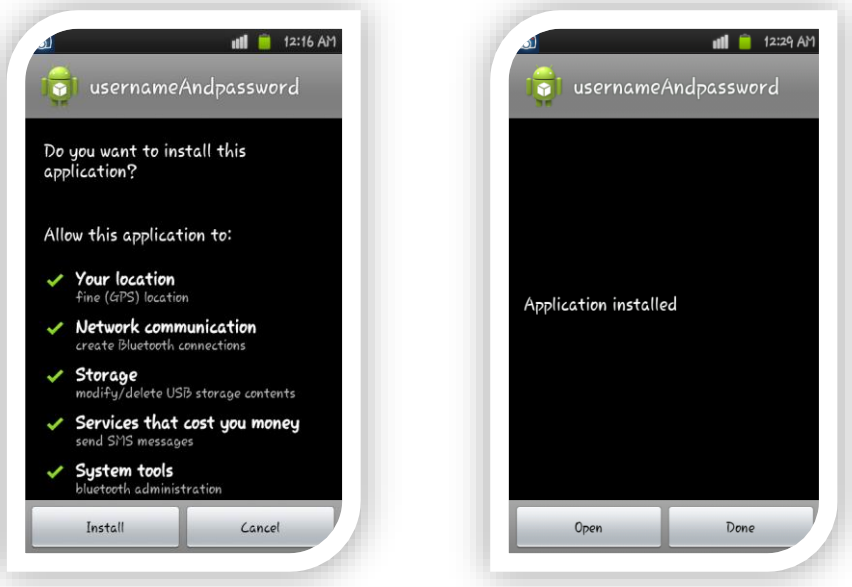


Figure 6.1: *Installing Application on Galaxy*

Now, the application is on mobile and it ready to use by the driver.

6.3.2 Server Application

An application has already been built using c# programming language. The system has two different methods to start up the application:

First method:

I. Open project "ITWAVs Server System" from the project path that was created using the Microsoft visual studio 2010 program. As shown in figure :

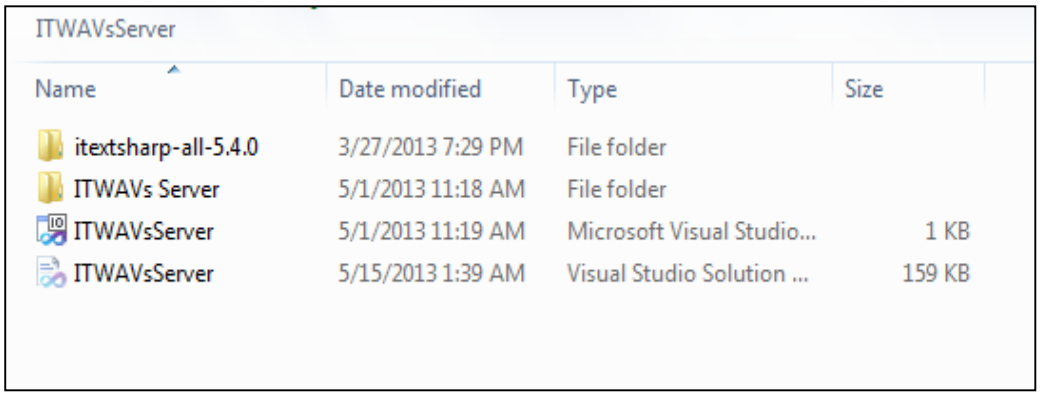
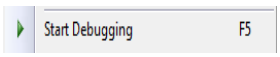


Figure 6.2: *Server path file*

II. Then after start it, a screen for the c# 2010 environment will be open with the project solution explorer that contain our system, then we debug the system to start it by click on Debug , or 

press F5.

second method :

I. Open “ITWAVs Server System” folder that was created using the Microsoft visual studio 2010 program. As shown in figure :

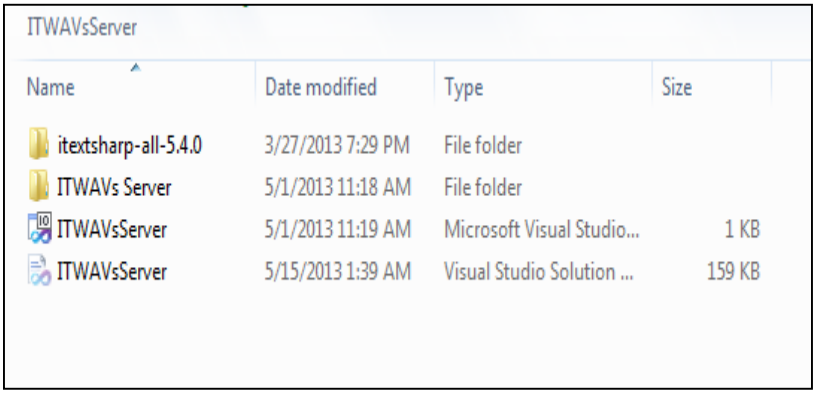
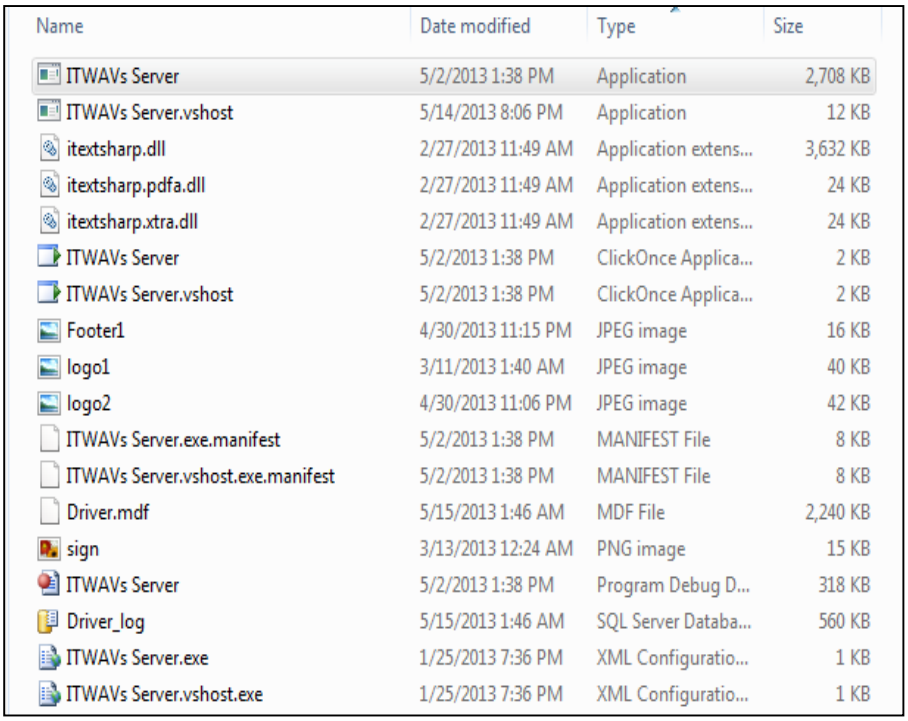


Figure 6.3: *ITWAVs Server System Project Folder*

II. Then open the debug folder from the bin folder, and open the "ITWAVs Server.exe" application, as shown in figure :



Name	Date modified	Type	Size
ITWAVs Server	5/2/2013 1:38 PM	Application	2,708 KB
ITWAVs Server.vshost	5/14/2013 8:06 PM	Application	12 KB
itextsharp.dll	2/27/2013 11:49 AM	Application extens...	3,632 KB
itextsharp.pdfa.dll	2/27/2013 11:49 AM	Application extens...	24 KB
itextsharp.xtra.dll	2/27/2013 11:49 AM	Application extens...	24 KB
ITWAVs Server	5/2/2013 1:38 PM	ClickOnce Applica...	2 KB
ITWAVs Server.vshost	5/2/2013 1:38 PM	ClickOnce Applica...	2 KB
Footer1	4/30/2013 11:15 PM	JPEG image	16 KB
logo1	3/11/2013 1:40 AM	JPEG image	40 KB
logo2	4/30/2013 11:06 PM	JPEG image	42 KB
ITWAVs Server.exe.manifest	5/2/2013 1:38 PM	MANIFEST File	8 KB
ITWAVs Server.vshost.exe.manifest	5/2/2013 1:38 PM	MANIFEST File	8 KB
Driver.mdf	5/15/2013 1:46 AM	MDF File	2,240 KB
sign	3/13/2013 12:24 AM	PNG image	15 KB
ITWAVs Server	5/2/2013 1:38 PM	Program Debug D...	318 KB
Driver_log	5/15/2013 1:46 AM	SQL Server Databa...	560 KB
ITWAVs Server.exe	1/25/2013 7:36 PM	XML Configuratio...	1 KB
ITWAVs Server.vshost.exe	1/25/2013 7:36 PM	XML Configuratio...	1 KB

Figure 64.: *ITWAVs Server Debug Folder*

The next figure shows the welcome screen of the server application:

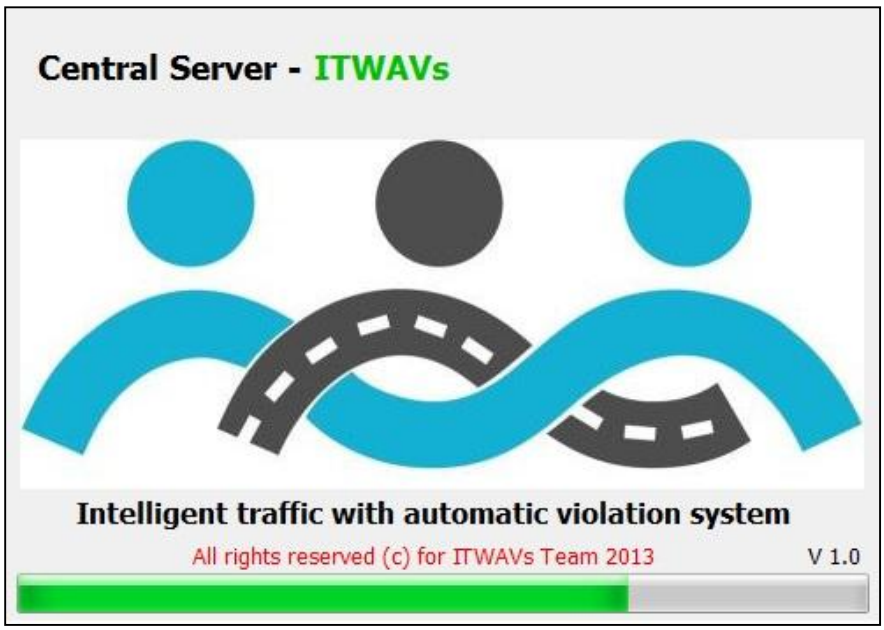


Figure 65.: *Server welcome screen*

6.4 Testing Scenarios

In this section, they are two demos, one for the sub system without any connection with microcontroller, and the second demo contains the whole system with microcontroller.

6.4.1 First Demo

In this demo, the scenario of starting the system without mechanical part, including all the necessary steps, which include:

1. **At mobile side :**

Configuration:

- Download the mobile application and open it, then give it to the car driver.
- Fill the registration information and click sign in.
- After registration, the system enable the GPS, the mobile will start receiving GPS signals and calculate its coordinates, then compare them with pre-stored database to get the street name and speed limit.
- When the driver exceeds the speed limit, the application gives five seconds to him to reduce his speed, and if the timer is become off while the car speed still higher than the street speed limit, then the violation will be occurred, and SMS message sent from mobile to the server.
- During that, if an accident happened, the crashing sensor inside mobile will detect it, and then gives the driver twenty seconds to click the button 

if nothing happened to him. Once the time is off and the button not clicked, the car location will be sent to the first aid unit.

The next figure, shows the application activates and classes that will be shown on Android mobile:

Figure 6.6: *Mobile phone application activities*

2. At the server side:

Configuration:

- Open the server application from Microsoft visual studio 2010 program.
- Fill the user name and the password and sign in.
- After signing in, the application will enter to the main screen window.

- Open the serial port and continue to initialize the GSM modem.
- After that, the server will be able to receive SMS data violation.
- When receiving the data violation, the serial manager get the data violation, check the validity of violation, generate the violation, and store it in the database.
- Finally, the application generates a report for violation. Which will be directly printed by printer or saved as pdf file to ITWAVs folder.

Figure 67: *Server main screen*

Test 1

The first test is applied over a small distances"one street", and without using a car. A person carries the mobile, open the application and start walking with a medium speed around 4.5 kmph, where street speed limit is 5 kmph.

Results:

- Test time: the time is approximately 3minutes.
- GPS error : NO GPS error in this test , once the person enter the street the application gives the street name and speed limit quickly .
- Number of times that the person exceeds speed limit : five times.
- Number of times that the person exceeds speed limit and not reduce his speed : two times ,and SMS was sending from mobile to server .
- Time that SMS required to reach server : one second .
- Time from exceeding speed limit to SMS reach server : 8 seconds , five seconds that the system wait to give the driver chance to reduce his speed , two seconds to sending SMS from mobile ,and one second to reach server.
- Time that the application pause after sending SMS : 10 seconds.
- Server required at maximum 0.5 second as a Time to process and perform the operations when received SMS message.

Errors

While testing the system the following errors happened:

- When the violation occurred, the mobile sends many SMS massages to the server instead of sending only one message (loop problem).
- At the server, there are error occurred after receiving many messages at the same time, where the system stop receiving.
- Crashing sensor at mobile does not works successfully on mobile application.

Challenges

The mentioned errors bring the following challenges:

- To give a perfect solution to solve the problem of sending many messages for the same violation.
- Solve the multi messages unacceptable at the server.
- Make crashing sensor works in the application.
- Do test by using car.
- Do test in many street instead of just one street.

Test 2

The second test is applied over a small region “three streets” using a car. A driver carries the mobile, open the application and start driving with a medium speed around 35 kmph, and the three streets speed limit is 40, 50, and 55 kmph consecutively.

Results:

- Test time: the test time is approximately one minute.
- GPS error: 3-6 meter GPS error in this test.
- Number of times that the person exceed speed limit: three times.
- Number of times that the person exceed speed limit without reduce his speed: two times, and SMS sent from mobile to server.
- Time needed by SMS to reach server, one second.
- Time from exceeding speed limit until SMS reaches server : 8 seconds , five seconds that the system wait to give the driver chance to reduce his speed, two seconds to send SMS from mobile, and one second to reach server.
- Time that the application pause after sending SMS: 20 seconds.
- Time required to process and perform the operation by the server when the SMS message received maximum 0.5 second.

Errors

While testing the system the following errors happened:

- Overlap between streets make exception error on mobile application, because of query problem.
- When crashing happened, the application detect it, but do nothing, and it does not send SMS location data to the first aid unit.

Challenges

The mentioned errors bring the following challenges:

- Solve overlap problems.
- Sending SMS location data when the crashing happened.

Test 3

The final test is applied over a small region “five streets” using a car. A driver carries the mobile, open the application and start driving with an average speed around 40kmph, and the streets speed limit is 45,50, 55,60,and 65 kmph.

Results:

- Test time: the test time is approximately 3 minutes.
- GPS error: 2-4 meter GPS error in this test.
- Number of times that the person exceeds speed limit : 7 times.
- Number of times that the person exceeds speed limit without reduce his speed: four times ,and SMS sent from mobile to server .
- Time required by SMS to reach the server : one second .

- Time from exceeding speed limit until SMS reaches the server: 8 seconds: five seconds that the system wait to give the driver chance to reduce his speed, two seconds to send SMS from mobile, and one second to reach the server.
- Time that the application pause after sending SMS: 20 seconds.
- Server required at maximum 0.5 second Time to process and perform the operation.

Errors

In the final test they are just one error, it is just GPS coordinates error, the error is around 3-6 meters, which was solved by using location determination algorithm that mentioned in chapter 4.

6.4.2 Demo Two (With mechanical part)

In this section, the scenario of starting the system with all its parts will be mentioned, including all the necessary steps, which include:

1. **At mobile side :**
   - Download the mobile application and open it, then give it to the car driver.
   - Fill the registration information and click sign in.
   - After registration, the system enable the GPS, the mobile will start receiving GPS signals and calculate its coordinates, then compare them with pre-stored database to get the street name and speed limit.
   - When the driver exceeds the speed limit, the application gives few seconds to him for reducing his speed, and if the timer is become off while the car speed still higher than the street speed limit ,then the violation will be occurred, and SMS message sent from mobile to the server.
   - During that, if an accident happened, the crashing sensor inside mobile will detect it, then gives the driver twenty seconds to 

click the button if

nothing happened to him . Once the time is off and the button not clicked, the car location will be sent to the first aid unit.

- The additional things in this part is using the Bluetooth to send data from Android to Arduino to control the mechanical part and reducing car speed when the violation occurred.

The next figure, shows the application activates and classes that will be shown in Android mobile:

Figure 6.8: *Mobile phone activities*

2. **At the server side:**
    - Open the server application from Microsoft visual studio 2010 program.
    - Fill the user name and the password and sign in.
    - After sign in, the application will enter to the main screen window .
    - Open the serial port and continue to initialize the GSM modem.

- After that, the server will be able to receive SMS data violation.
- When receiving the data violation,the serial manager get the data violation, check the validity of violation, generate the violation, and store it in the database.
- Finally, the application generates a report for violation. which will be directly printed by printer or saved as pdf file to ITWAVs folder.

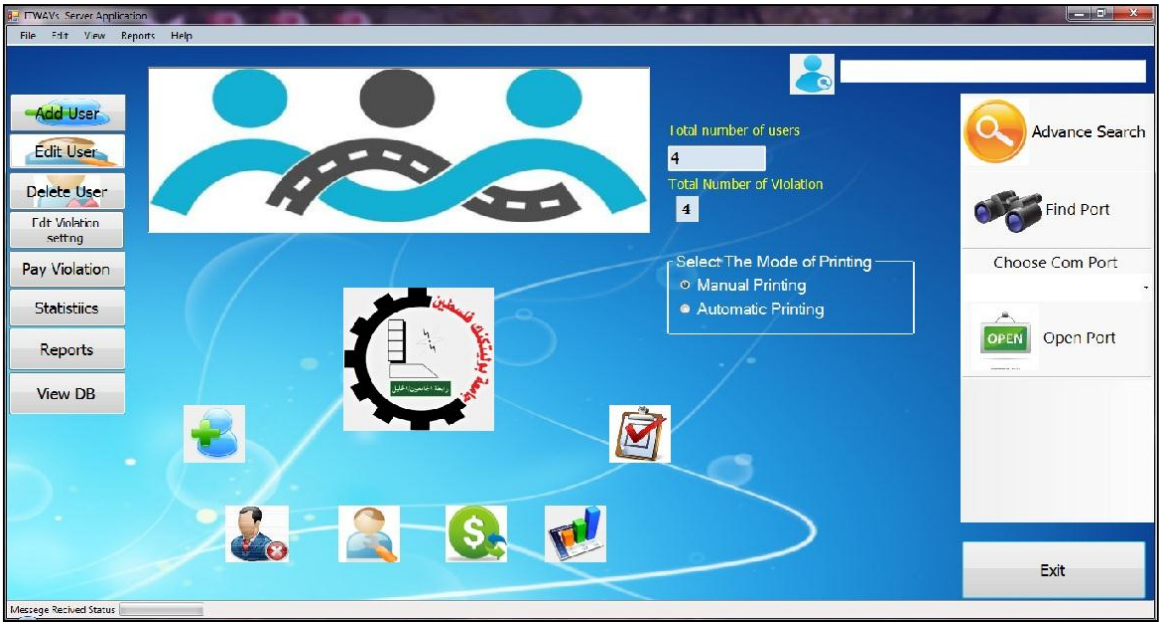


Figure 69.: *Server Main screen*

3. At the mechanical side:

- Connect brake solenoid relay with 12v wire from vehicle battery, ground wire, and output control signal wire from Arduino microcontroller.
- Connect accelerator pedal sensor relay with ground wire, and output control signal wire from Arduino microcontroller.

Test 1

The first test is applied over a small distance “one street”. A driver open the application and start driving with a medium speed around 40 kmph, and street speed limit is 50 kmph.

Results :

- Test time: the test time is approximately 15 minutes.
- GPS error: 3-6 meters GPS error in this test.
- Number of times that the driver exceeds speed limit : 6 times.
- Number of time that the driver exceeds speed limit without reduce his speed: 3 times, and SMS sent from mobile to server .
- Time required by SMS to reach the server one second.
- Time from exceeding speed limit until SMS reaches server : 8 seconds .
- Time needed by the application to send control data to the microcontroller via Bluetooth: part of second.
- Time where the application pause after sending control data via Bluetooth: 2 seconds.
- Time that the microcontroller activate brake actuator and deactivate the accelerator pedal: 4 seconds .
- Server required Time to process and perform the operations when SMS received: at maximum 0.5 second.
- After the mechanical system activated, the vehicle speed decreases about 10 kmph.

Errors

While testing the system, the following errors happened:

- After accelerator pedal deactivated, it doesn’t activate again .
- The system can’t decrease car speed to speed limit.

Challenges

The mentioned errors bring the following challenges:

- Successfully activate accelerator pedal after deactivate it.
- Decreasing car speed below speed limit.
- Do test in many street instead of just one street.

Test 2

The second test is applied over eight streets with different speed limits. A driver open the application and start driving with a medium speed around 40 kmph.

Results:

- Testing time: the test time is approximately 25 minutes.
- GPS error: 3-6 meter GPS error in this test.
- Number of times that the driver exceeds speed limit: 9 times.
- Number of times that the driver exceeds speed limit without reduce his speed: 5 times, and SMS sent from mobile to server .
- Time required by SMS to reach the server: one second .
- Time from exceeding speed limit until SMS reaches the server: 8 seconds.
- Time needed by the application to send control data to the microcontroller via Bluetooth: part of second.
- Time where the application pause after sending control data via Bluetooth: 20 seconds.
- Time that the microcontroller activate brake actuator and deactivate the accelerator pedal: 7 seconds .
- After activate the mechanical system, the vehicle speed decreases about 20 kmph.
- Server required Time to process and perform the operations when SMS received: at maximum 0.5 second.

Errors

While testing the system, the accelerator pedal works approximately about 90 % truth. The GPS error is around 3-6 meter, which was solved by using location determination algorithm that mentioned in chapter 4.

7

Chapter Seven

Conclusion and Recommendation

Chapter 7:
Conclusion and Recommendations

7.1 Introduction

This chapter describes the real learning outcomes have been acquired during the work on the project, we will mention what we achieved in this project, and the conclusion for All things that we have done, also we will talk about the challenges that we faced and ending with recommendations needed for the future work.

7.2 Problems

Many problems, challenges, and issues have been raised during the work on the project. Many experiments, suggestions, ideas and researches have been carried out to deal with the different situations. Some of these problems are:

1) The Availability of the quantity and the quality of some of the Project's equipment.
2) The Israeli Restrictions on the imported equipment and the delay that occurs accordingly.
3) Many problems have been raised while receiving GPS signals on mobile phone, due to the restrictions on the GPS in Palestine, and the difficulty in using GPS over mobile networks, which required very high cost.
4) Problems with transfer data over USB, due to the high android version required, which mean new mobile type, with high price.

7.3 Acquired Learning Outcomes

After accomplishing, the project tasks many talents and abilities have been achieved as:

1) We have learnt Android programming language that is used to program the mobile phone application, and how to deal with Android developer tolls program, which we depend on it to build a mobile application to receive an accurate location of the car, and use mobile database query to get speed limit and name of the street, and display it on the screen.
2) We have learnt C# programming language and database building.
3) We have learnt how to interface the microcontroller with android system, and mechanical system.
4) We have learn how to control vehicle speed, by building a mechanical system for this purpose.
5) We have developed our abilities in troubleshooting and problem solving.

7.4 Recommendations for future work

At the end, some ideas can be given to develop the system or extend its duties and functions, and some recommendations can be given to avoid the problems that may happen in the future as:

1) Adding automatic update property to the system, by making it possible to extend, edit or delete GPS coordinates by the server.
2) Develop a live connection between mobile phone, and server, so any change in mobile will change on server automatically and vice versa.
3) Applied this system by using 3G or LTE networks.
4) Develop a website, which give driver the ability to check his violations, pay it and communicate with admins.
5) Controlling vehicle speed by deal with vehicle computer.

ABBREVIATIONS

A

ABS: Anti-lock Braking System

AC: Alternating Current

ACDIS: Active Distance Support system

ADC: Analog-to-Digital Converter

ALU: Arithmetical Logical Unit

API: Application Programming Interface

APPS: Accelerator pedal position sensors

APPS main: Main potentiometer of Accelerator pedal position sensors

APPS sub: Sub potentiometer of Accelerator pedal position sensors

C

CDMA: Code division multiple access

CIL: Common Intermediate Language

CLR: Common Language Runtime

CPU: Central Processing Unit

COM: Component Object Model

CRC: Cyclic Redundancy Check

CTPS: Closed Throttle Position Sensor

D

DC: Direct Current

DM: Data Minus

DP: Data Pulse

E

***ECM*:** Engine Control Module

ECU: Engine Control Unit

EDGE: Enhanced Data rates for GSM Evolution.

EEC: Electronic Engine Control

EGM96: Earth Gravitational Model

EGM2008: Earth Gravitational Model 2008

EOP: End of Packet

ER/ERD: Entity-Relationship Model

ETC: Electronic throttle control

F

FTP: File Transfer Protocol

G

Gbps: Gigabit per second

GPRS: General Packet Radio Service

GPS: Global Positioning System

GSM: Global System for Mobile

H

HLR: Home Location Register

HSDPA: High-Speed Downlink Packet Access

HSPA: High Speed Packet Access

HTML: Hyperactive Text Markup Language

HTTP: Hypertext Transfer Protocol

I

IACV: Idle Air Control Valve

IBM: International Business Machines

IC: Integrated Circuit

ID: Identifier

ICSP: In Circuit Serial Programming

I/O: Input Output

IP: Internet Protocol

ISA: Intelligent speed adaptation

J

JIT: Just-In-Time

K

Kbps: kilo bit per second

Km: kilometer/s

L

LSB: Least Significant Bit

M

M: Meter

Max: Maximum

Mbps: Mega bit per second

MHz: Megahertz

Min: Minimum

MMS: Multimedia-Messaging Service

Modem: Modulator-demodulator

N

NAD83: North American Datum of 1983

NAVSTAR*:* Navigation Satellite Time and Ranging

NRZI: Non-Return-to-Zero Inverted

O

OS: Operating System

P

PC: Personal Computer

PCM: Power-train Control Module

PCMCIA: Personal Computer Memory Card International Association

PDA: Personal Digital Assistant

PWM*:* Pulse-Width Modulation

R

RAM: Random Access Memory

ROM: Read Only Memory

RX: Receiver

S

SGSN: Serving GPRS Support Node

SIM: Subscriber Identity Module

SMS: Short Message Service

SOP: Start of Frame Packet

SQL: Structured Query Language

T

TCP: Transmission Control Protocol

TDMA: Time Division Multiple Access

TPS: Throttle Position Sensor

TX: Transmitter

U

UART: Universal Asynchronous Receiver/Transmitter

UML: Unified Modeling Language

UMTS: Universal Mobile Telephone Service

URL: Uniform Resource Locator

USB: Universal Serial Bus

V

VICS: Vehicle Information and Communication System

VPN: virtual Private Network

VSS: Vehicle Speed Sensor

W

WAP: Wireless Application Protocol

WGS84: World Geodetic System 1984

WML: Wireless Markup Language

WOT: Wide-Open Throttle

TABLE OF FIGURES

List of tables

References

[1] [Online]. Available: http://www.who.int/en/.

[2] [Online]. Available: http://en.wikipedia.org/wiki/Vehicle_Information_and_Communication_System.

[3] G. Omaima Bamasak, "Monitored Intelligent Car Speed Adaptation system," Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz, Jeddah, Saudi Arabia, 2011.

[4] F. Samantha Jamson, "Intelligent speed adaptation litrature review and scoping study," The University of leeds and Mira ltd., London, 2006.

[5] M. kennedy, "The Global Positioning System and GIS An Introduction," Fanrica,, United States of America, 1996.

[6] [Online]. Available: http://en.wikipedia.org/wiki/Global_Positioning_System.

[7] [Online]. Available: http://www.articlesnatch.com/Article/Gps-System-Components/1004987.

[8] [Online]. Available: http://www.nowsms.com/faq/what-is-a-gsm-modem.

[9] CNN, "The text message turns 20," December 3, 2012.

[10] [Online]. Available: http://communities-dominate.blogs.com/brands/2011/01/time-to-confirm-some-mobile-user-numbers-sms-mms-mobile-internet-m-news.html.

[11] [Online]. Available: http://www.webopedia.com/TERM/S/short_message_service.html.

[12] "Short Message Service," 3GPP2 and its Organizational Partners.

[13] [Online]. Available: http://www.funsms.net/advantages_sms.htm.

[14] [Online]. Available: http://www.streetdirectory.com/travel_guide/140316/technology/benefits_of_sms.html.

[15] A. Mahdavifar, "Automatic Veicle Location systems," World Academy of Science, Engineering and Technology, 2009.

[16] [Online]. Available: http://www.codeproject.com/Articles/38705/Send-and-Read-SMS-through-a-GSM-Modem-using-AT-Com.

[17] [Online]. Available: mikroElektronika. http://www.mikroe.com/chapters/view/64/chapter-1-introduction-to-microcontrollers/. mikroElektronika..

[18] [Online]. Available: http://arduino.cc/en/. Arduino. [Online] [Cited: 9 28, 2012.].

[19] [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html.

[20] [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html.

[21] [Online]. Available: http://www.freescale.com/webapp/sps/site/overview.jsp?code=DRSNSAXLRTN.

[22] M. L. Murphy, "Start Application Mainfest," in *Begining Android*, United states of America, The Expert's Voice, 2009, pp. 9-93.

[23] G. Coffman, "Microsoft SQL Server. SQL Server 7 : The Complete Reference. s.l.," Brandon A.Nordin, 1999.

[24] C. Hall, "Jayaraman, Radhika and Sethupathi, Madhavi," Indiana : Dean Miller, 2002.

[25] P. A. B. a. N. Drayton, "C# in a Nutshell. : O'Reilly & Associates," United States of America, 2002.

[26] [Online]. Available: http://www.mikroe.com/chapters/view/64/chapter-1-introduction-to-microcontrollers/. mikroElektronika. .

[27] [Online]. Available: http://en.wikipedia.org/wiki/Powertrain_control_module.

[28] [Online]. Available: http://www.diycardoctor.com/accelerator_pedal_position.htm .

[29] [Online]. Available: http://askpete-hella.com/2011/01/04/accelerator-pedal-position-sensor/.

[30] [Online]. Available: http://askpete-hella.com/2011/01/04/accelerator-pedal-position-sensor/.

[31] [Online]. Available: http://autospeed.com/cms/title_The-Bosch-MEMotronic-System-Part-1/A_108379/article.html .

[32] [Online]. Available: http://www.alibaba.com/product-gs/581555098/Build_in_3G_Dual_sim_Android.html.

[33] [Online]. Available: http://www.gsmarena.com/samsung_i9000_galaxy_s-3115.php.

[34] [Online]. Available: http://www.reflexiona.biz/shop/lang-en/-shields/94--cellular-shield-con-sm5100b.html.

[35] A. Technologies, "Telecom MF180 USB Modem User Guide," Aglient Technologies, 2009.

[36] "Bluetooth traveler," hoovers.com, 9 April 2010..

[37] H. Newton, "Newton's telecom dictionary," Flatiron Publishing, New York, 2007.

[38] M. Specialties, "Shock & Impact Testing Piezoresistive MEMS mV Output, DC Response Low Noise, Shielded Cable," www.meas-spec.com, 08/30/2012.

[39] Y. Ochieng, "MAP-MATCHING IN COMPLEX URBAN ROAD NETWORKS," Centre for Transport Studies, Department of Civil and Environmental Engineering, London, 2006.

[40] M. A. Quddus, "High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications," University of London, London, 2006.

[41] F. Marchal, "Efficient map-matching of large GPS data sets - Tests on a speed monitoring experiment in Zurich," Institut for Verkehrsplanung, 2004.

[42] L. Friese, "Updating the Spatial Alignment Attributes of Digital Maps Using GPS Points," Princeton University, May, 2005.

[43] [Online]. Available: http://www.justanswer.com/chevy/1tv7u-06-duramax-lag-issues-when-step.html.

[44] [Online]. Available: http://www.alldatasheet.com/datasheet-pdf/pdf/310815/AD/ADG1419.html.

[46] [Online]. Available: http://www.engine-light-help.com/speed-sensor.html.

[47] [Online]. Available: http://en.wikipedia.org/wiki/Powertrain_control_module.

[48] [Online]. Available: http://www.diycardoctor.com/accelerator_pedal_position.htm.

[49] [Online]. Available: http://autospeed.com/cms/title_The-Bosch-MEMotronic-System-Part-1/A_108379/article.html .

[50] [Online]. Available: http://www.cdxetextbook.com/electrical/ignition/conBreakComp/ignitionswitch.html.

[51] [Online]. Available: www.tegger.com/hondafaq/mainrelayoperation/index.html.

[52] [Online]. Available: http://www.justanswer.com/chevy/1tv7u-06-duramax-lag-issues-when-step.html.

[53] [Online]. Available: http://www.samarins.com/glossary/starter.html.

[54] [Online]. Available: http://www.samarins.com/glossary/starter.html.

[55] [Online]. Available: http://www.alldatasheet.com/datasheet-pdf/pdf/8825/NSC/LM2917.html.

[56] [Online]. Available: http://www.alldatasheet.com/datasheet-pdf/pdf/310815/AD/ADG1419.html.

[57] [Online]. Available: http://www.alldatasheet.com/datasheet-pdf/pdf/310815/AD/ADG1419.html.

[58] [Online]. Available: http://www.nowsms.com/faq/what-is-a-gsm-modem.

[59] G. Blewitt, "Basics of the GPS Technique: Observation Equations," Department of Geomatics, University of Newcastle, Newcastle upon Tyne, NE1 7RU, United Kingdom, 1997.

[60] [Online]. Available: http://www.wireless.att.com/learn/why/technology/mobile-broadband-and-GSM.jsp.

[61] A. Sicher, "GPRS Technology Overview," Dell Computer Corporation, 2002.

[62] Corporation, " Communications Using GPRS," SCADA-RTU.

[63] [Online]. Available: http://www.yokogawa.com/iab/appnotes/iab-app-gprs-en.htm.

[64] [Online]. Available: http://www.developershome.com/sms/howToReceiveSMSUsingPC.asp.

[65] [Online]. Available: http://www.makeuseof.com/tag/how-to-use-your-gprs-cell-phone-as-a-modem/.

[66] [Online]. Available: http://www.ehow.com/list_7628002_advantages-gprs.html.

[67] "USB Data Packet Structure," Future Technology Devices International Limited (FTDI), United Kingdom, 2011.

[68] F. Ivis, "Calculating Geographic Distance: Concepts and Methods," Canadian Institute for Health Information, Toronto, Ontario, Canada.

[69] R. Johnson, "Spherical Trigonometry," West Hills Institute of Mathematics.

[70] R. Khondker Shajadul Hasan, "Cost Effective GPS-GPRS Based Object Tracking System," Proceeding of the international Multiconfrence of Engineering and Computer Scientists, Hpng Kong, 2009.

[71] P. Bhumkar, "Intelligent Car System for Accident Prevention Using ARM-7," *International Journal of Emering Technology and advanced Engineering*, p. 527, 2012.

[72] J. Karki, "Signal Conditioning Piezoelectric Sensors," Texas Instruments, Texas, 2003.

[73] Z. company, "Connect Me 3G and Wi-Fi Connection Manager, User Guide," Telecom ZTA, 2010.

Table of Contents