IMPLEMENTATION OF ANDROID VERSATILE DRIVE INTERFACE IN A VEHICLE MODEL

AHMAD NAZIRUL ARIFF BIN A MAJID

Thesis submitted in partial fulfilment of the requirements for the award of Bachelor of Mechatronics Engineering (Dual Degree Programme UMP-HsKA, Germany)

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

&

Fakultät für Machinenbau und Mechatronik
HOCHSCHULE KARLSRUHE TECHNIK UND WIRTSCHAFT

MARCH 2016

ABSTRACT

Versatile Drive Interface is important as a platform to the user to monitor the health of the car and as an infotainment to controlling the interior features of the vehicle. Furthermore the increase of the number of vehicles on the road is proportional to the increase of the vehicle's services. With the data provided by the on board diagnostic system in the vehicle can let the users somehow notice the vehicle's problems and can prevent or predict the next service. With more complex and advance technologies in the automotive industries means more features will be add into a vehicle and with the help of the Versatile Drive Interface (VDI), users are know more convenient to control the features from inside or outside of their vehicles. This, on one hand, allows for providing more convenient and modern method by implement the Android technology This thesis involved the design process and the development process of the Android based software application and the hardware part and more focusing on the local manufactured vehicle in Malaysia. Finally, this thesis present a prototype of the implementation of a Versatile Drive Interface (VDI) in a vehicle model.

ABSTRAK

'Versatile Drive Interface' merupakan asas yang penting bagi pengguna untuk memantau prestasi kereta dan sebagai info hiburan untuk mengawal kemudahan dalaman sesebuah kereta. Tambahan pula dengan penambahan jumlah kenderaan di atas jalan secara langsung menambahkan jumlah servis kenderaan. Dengan data yang disediakan oleh diagnostic dalaman kereta membolehkan pengguna menghalang atau mengagak servis seterusnya. Dengan adanya industry automotive yang semakin maju dan complex, ini bererti semakin banyak teknologi yang ditambah di dalam kenderaan dan dengan adanya system 'Versatile Drive Interface (VDI)' membolehkan pengguna lebih selesa mengawal dari dalam atau luar kenderaan. Hal ini membolehkan untuk menyediakan cara yang lebih selesa dan moden dengan melaksanakan teknologi Android. Tesis ini melibatkan proses mereka bentuk dan proses pembangunan perisian aplikasi Android dan system perkakasan dan memfokuskan kepada pengeluar kederaan tempatan di Malaysia. Akhir kata, tesis ini mempersembahkan sebuah prototaip dalam perlaksanaan 'Versatile Drive Interface (VDI)' dalam model kenderaan.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

VDI	Versatile Drive Interface is a tablet specific mobile concept which is used to provide a driver's preference for interfacing with vehicles.
Android	An open source platform designed for mobile devices.
OS	Operating system used for controlling device.
Android SDK	Developing kit for Android platform.
JRE	The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine and other components to run applets and applications written in Java programming language.
IDE	Integrated development environment (E.g. Eclipse, OpenCV, etc.)
Android Studio	An IDE tool which focusing on Android application in Java language which developed by Android team itself.
Eclipse	An IDE tool which consist of workspace and plug-ins. It can be used to develop applications in Java and other programming languages
Open source software	Computer software developers of which have made source code of the program available and license it under license which gives to software provides the rights to study, change and distribute the software to anyone and for any purpose. (St. Laurent, Andrew M, 2008)
AVD	Android Virtual Device is a tool for running Android OS along with applications.
XML	Extensible Markup Language is a specification for storing information.
GUI	Graphical user interface.
ADT	Android development tool, package of tools which are used to develop Android application.
OBD-II	On-Board Diagnostic (OBD) –II is the improvement over OBD-1 which provides the list of vehicle parameters to monitor and encode the data.
OBD-II adapter	A device use to interpret the data from the OBD-II port.

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

In Malaysia, the usage of mobile application inside the vehicle is still not widely implemented compared to the foreign automotive industries. With the implementation of VDI inside a vehicle can reduce time, saving cost and provide better vehicle experience. An important factor for the success of automotive interfaces for drivers and passengers is the system's ability to take the high contextual nature of traveling by car into account. In the tradition of contextual interfaces, successful in-car interfaces should react to what is going on in and around the car. Thus, in-car systems need to have access to data about the car, its interior and its surrounding. This contextual data can be used in different ways, two of which are most prominent: Applying car context data has the potential to support the creation of new user interfaces that make traveling by car safer and result in an overall better user experience.

1.2 PROJECT MOTIVATION

The capability of mobile phones to act as central part of such toolkits or context measuring platforms has already been exploited by before smartphones entered the market. Since then, these devices gained even more value for such a purpose. Modern smartphones are platforms with increasing computing capabilities, power supply and a great potential for connectivity, e.g., through Bluetooth. A wide range of smartphone apps is available on market places that use the ability of a smartphone to display car data. Torque for example, is an application that provides temperature information, RPM gauges

and speed timings. These apps nevertheless focus on displaying data and on limited functionality such as timing, but not on providing car data for interactive prototypes. So based on the mobile application inventions used in the vehicle nowadays, the VDI is developed to improve the driver preferences and the conventional mobile application in vehicle.

1.3 PROBLEM STATEMENT

There are only few applications which provide data monitoring for the vehicle. Receiving the data from the OBD-II from the vehicle requires a detail researched as it is one of the most difficult task in this project. The communication and the speed of the data transfer between the Android devices must be synchronise with the OBD-II to avoid data loss.

Most of local cars still using uncomplicated circuit system which can be used throughout this project. The modelled circuit must have the same characteristic as the actual one to achieve the goal which is to simulate the real car system for the central locking and power window. The software interface between the controller and the Android device have to communicate in transferring and receiving the data in the same manner.

The communication system is one of the issue where each of the system have their own communication make it hard to combine into a single application. As the Bluetooth only can connect to a single device at a time, this is one of the challenge in this project.

1.4 PROJECT OBJECTIVES

The aim of this project to develop a user-vehicle interface using Android technology which focusing on local vehicle. The VDI consist of two main features which are the OBD-II reader and car interior control (door lock and power window). The OBD-II Reader establishes a wireless Bluetooth communication link between an Android device and the automobile's On-Board Diagnostic system (OBD). The objectives for the OBD-II Reader are shown below:

1. To develop an application to communicate with vehicle on board diagnostic computer with Android device.
2. To design and develop an Android application with GUI which able to display these features:
    a) The real time vehicle engine information such as Power Supply Voltage, Engine RPM, Throttle Position, Air Intake Temperature, Speed and Engine Coolant Temperature.
    b) Extra features of the Android application such as Compass, GPS detection, Trip List and Diagnostic/Clear Troublecodes.

The second part of the VDI consist of controlling the central locking system and the power window using a microcontroller (Arduino). The objectives are as below:

3.1 To develop an Android application to be able to control the interior features of a vehicle which have:
- Search and connect the paired Bluetooth device.
- The digital buttons to control the lock and unlock of the central locking system and power window.

3.2 To develop a model circuit for the central locking and power window system based on the actual circuit of local cars.

1.5 LAYOUT OF THESIS

Chapter 1 is a brief introduction that explains the motivation and the main objectives of this project. Chapter 2 corresponds to the literature review which is the related researched based on the project and about the existing work on the field. It contains a brief explanation of the existing smartphone applications and an analysis of the existing research works on using smartphones for vehicular information and control system. Chapter 3 contains the methodology which is the work process until the completion of the project.

Chapter 4 explains the results and discussion obtained from the finalise application and circuit which has been tested according to the requirements. Chapter 5 concluded the whole project.

CHAPTER 2

LITERATURE REVIEW

2.1 OVERVIEW OF AVAILABLE MOBILE APPLICATIONS

The potential to use mobile phones for prototyping in the vehicle has already been demonstrated by Navet (Navet et al, 2005) and Dongho (Dongho Choi et al, 2005) for example, used mobile phones and GPS sensors to provide micro entertainment when the car is waiting in front of a traffic light. Flach (Flach et al, 2011) propose CarMa, a system that provides automotive parameters and supports the inclusion of apps for tuning the car to personal needs and different road situations.

The most common vehicular feature that the drivers use with the smartphone is routing. People are being used on getting the routes from a smartphone application and get the indications when the driver does not know how to arrive to a desired destination (Gianpoulo M. et al, 2009). A widely used application that implements this feature is Google Maps, despite it is not only focused in automotive.

One of the most famous vehicular oriented applications is Waze (Waze Mobile, 2009). This application is mainly a social network focused only in driving. It uses collaboratively made road maps in order to calculate routes and guide the users to the desired destination. This application uses the GPS system to locate the users and calculate the current speed. From this data, the application is able to compute the traffic situation from the users' state and allow other users to recalculate the routes according to this data.

In figure 2.1 and 2.2 there is a capture of the application map with the calculated route, the road events and the location of the other users that are currently using the application.

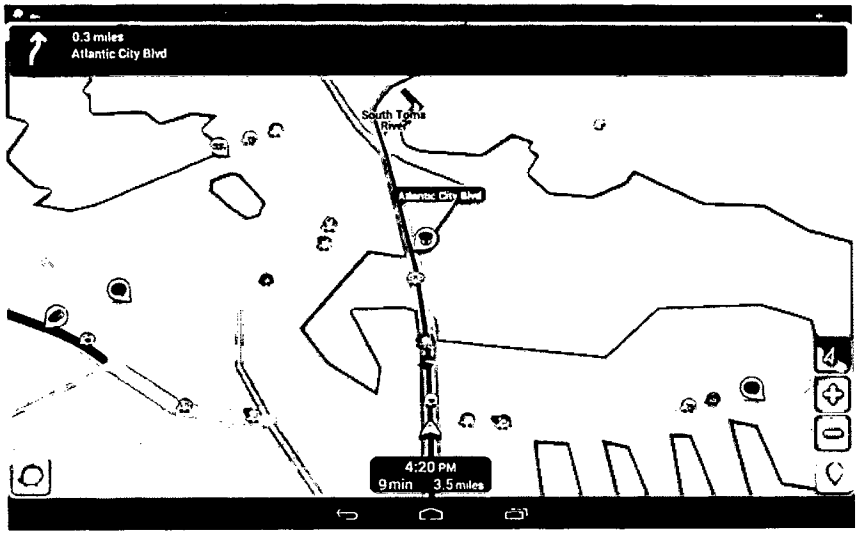

Figure 2.1: Waze application.

Source: Waze Mobile 2009

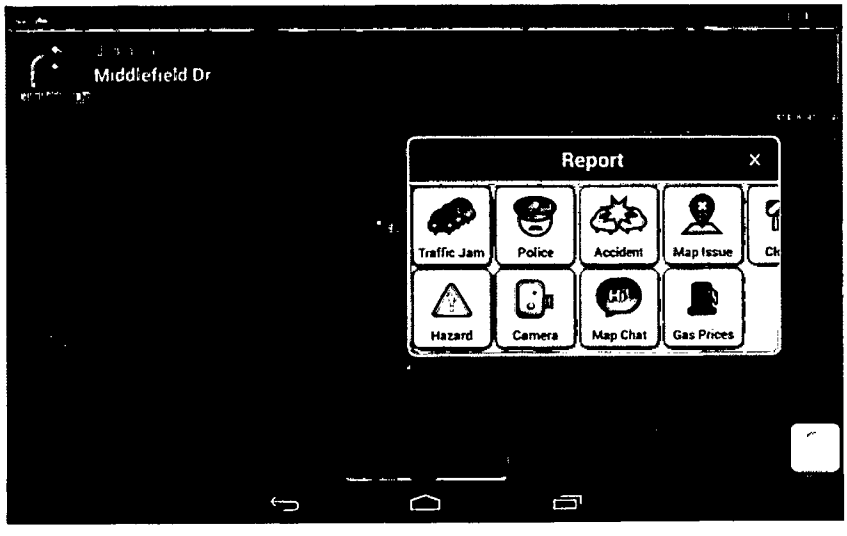

Figure 2.2: Features inside the Waze application

Source: Waze Mobile 2009

Apart from the applications that improve the user navigation and road information there are other vehicle oriented applications that focus on reading data from the car as well as reading failure and trouble codes. This type of applications are commonly developed to use an On-Board Diagnostics second generation (OBD-II) to Bluetooth device to extract the information from the car Engine Control Unit (ECU) or Powertrain Control Module (Siva Karthik, 2013). The most known application in this field is the Torque Pro (Ian.H, 2011). This application is able to read data (e.g. vehicle speed, engine speed, intake air temperature, mass air flow) and Diagnostic Trouble Codes (DTC) from the OBD-II interface. In figure 2.3 there are two screen captures of the Torque Pro where we can see some real time data displays and the trouble codes list. What is interesting in this kind of applications is the possibility of getting real time data from the vehicle by only connecting a simple device on the OBD-II port of the vehicle. This is an economic way to improve the vehicle dashboard features wirelessly with a simple smartphone through the Bluetooth or Wi-Fi interface.

Figure 2.3: Torque Pro application

Source: Torque Pro, 2011

Other than the navigation and vehicle information, vehicle infotainment application is widely used in more advanced car as discussed by Vagesh (Vagesh K., 2013) and Simon (Simon I., 2014) in their research paper about the use of the vehicle infotainment in automotive industry. This infotainment provide much interesting driving experience towards the user. Another interesting application developed by application developer named Graeme Richardson which is 'VDI' (G. Richardson, 2011) is focusing on the driver preferences for interfacing with vehicles. It have the features to unlock/lock the doors, controlling the power windows, controlling radio channel and sound system, weather forecast and internet browsing and others. This application is more complicated and advanced and it will be useful for the user to control most of the vehicle's features using the smartphone/tablet.

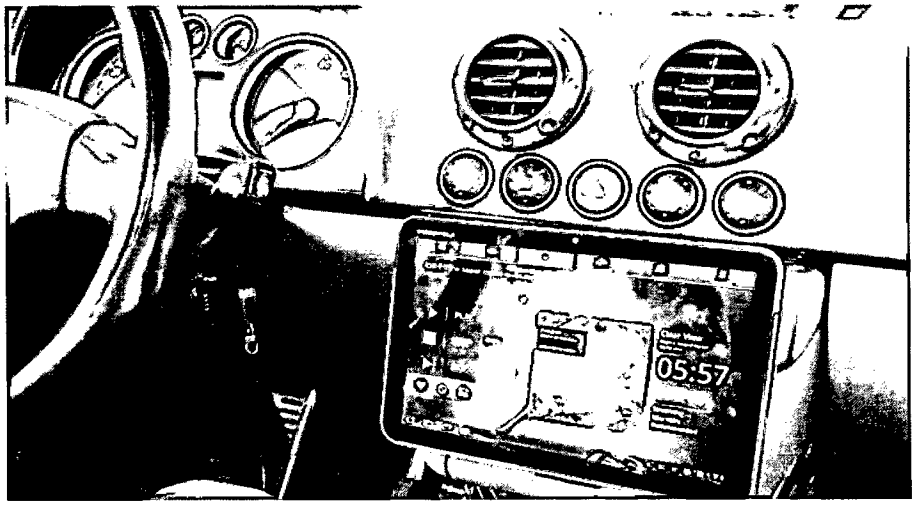

Figure 2.4: VDI placement inside a car.

Source: G.Richardson 2011

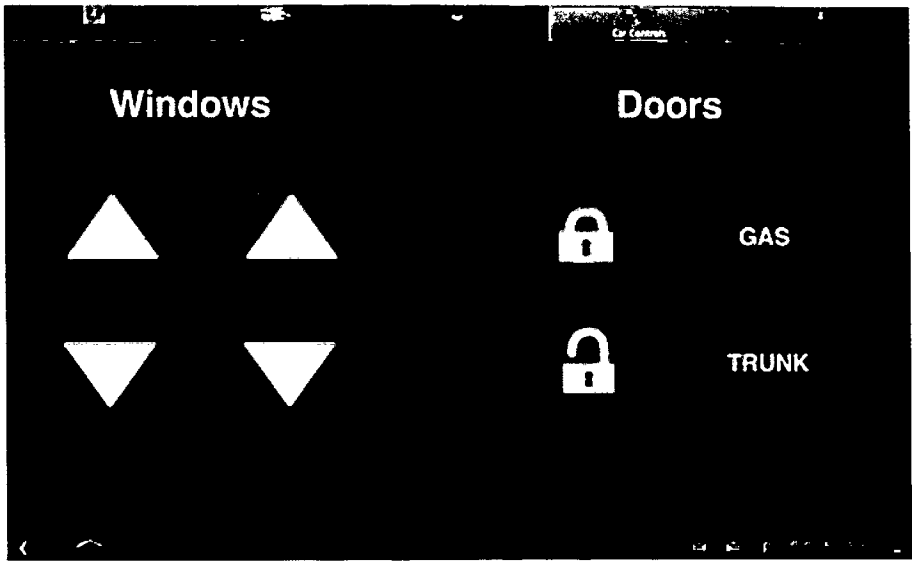

Figure 2.5: Features in the VDI application

Source: G.Richardson, 2011

2.2 ON-BOARD DIAGNOSTIC II (OBD-II)

The OBD-II (On-board Diagnostics) standard describes an interface to a car's ECU (engine control unit) as discussed by Christobal (R.Christobal, 2011). The ECU collects data from different sensors within the car's combustion engine which describe certain engine parameters. OBD-II specifies the connector and communication protocol to access this data, which is primarily done for diagnostic and repair purposes and this process is a part of the vehicle-vehicle communication described by Harding (Harding, 2014). It grants access to real-time data (e.g., speed, rounds per minute) and so called trouble codes (i.e. everything that is responsible for the check engine light to be activated). The data is not intuitively accessible and requires a certain amount of refinement before further use. For example the rounds per minute value is encoded in two bytes, which is calculated by the formula (256 * value of first byte + value of second byte) / 4 (R.Christobal, 2011). To access the OBD-II data we used two devices based on the ELM327 microcontroller. This microcontroller marketed as accessories to Android devices to monitor engine status, read and reset trouble codes or to calculate average fuel consumption.

2.2.1 OBD-II Protocols

An OBD2 compliant vehicle can use any of the five communication protocols which are commonly worldwide. Below is the list of each protocol.

1. ISO 15765-4 (CAN-BUS)

The most modern protocol, mandatory for all 2008+ vehicles sold in the US. Uses pins 6 and 14 (referenced to signal ground), communication is differential. Four variants of ISO15765 exist. They differ only in identifier length and bus speed:

- ISO 15765-4 CAN (11 bit ID, 500 Kbaud).
- ISO 15765-4 CAN (29 bit ID, 500 Kbaud).
- ISO 15765-4 CAN (11 bit ID, 250 Kbaud).
- ISO 15765-4 CAN (29 bit ID, 250 Kbaud).

Fiat/Alfa/Lancia used also fault-tolerant CAN-BUS at 50 kbaud, not compatible with OBD2 standard.

2. ISO 14230-4 (KWP2000)

Very common protocol for 2003+ vehicles using ISO9141 K-Line. Uses pin 7. Two variants of ISO14230-4 exist. They differ only in method of communication initialization. All use 10400 bits per second.

- ISO 14230-4 KWP (5 baud init, 10.4 Kbaud).
- ISO 14230-4 KWP (fast init, 10.4 Kbaud).

3. ISO 9141-2

Older protocol used mostly on European vehicles between 2000 and 2004. Uses pins 7 and optionally 15.

4. SAE J1850 VPW

Diagnostic bus used mostly on GM vehicles. Uses pin 1, communication speed is 10.4 kB/sec.

5. SAE J1850 PWM

Diagnostic bus/protocol used mostly on Ford. Uses pins 1 and 2, communication signal is differential and it's rate is 41.6kB/sec.

2.2.2 ELM 327

In Section 2.1.1 we have explained the On-Board Diagnostics standard that all the vehicles must implement. However, the standard has the drawback that any communication protocol is not directly compatible with common computers or mobile devices. To overcome this, Elm Electronics developed a chip, the ELM327, to act as a bridge between the OBD connector and the standard RS232 serial interface. This chip is an appropriately programmed PIC18F2480 microcontroller from Microchip Technology (Elm Electronics, 2016).

The ELM327 chip is ready to be connected directly to the OBD connector with some external hardware that is already specified in the datasheet. As we can see in the ELM327 pinout (figure 2.6) there are pins for each OBD connector pin. We can also see that there are two pins (RS232 Rx and RS232 Tx) that are in charge to carry the RS232 communication to the computer or another device. The ELM327 chip has also several pins to be connected directly to the LEDs that will indicate the transmitting or receiving through the OBD or RS232 interfaces.

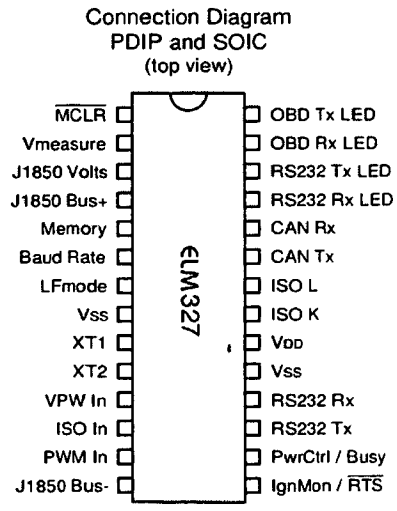

Figure 2.6: The Pinout of the ELM327 chip

Recently, RS232 interfaces are not much supported in laptops that are the computers most likely to be connected to an OBD interface. For this reason the most of ELM327 implementations include a bridge to adapt the RS232 to a most supported interface. Some implementations use USB but the most recent are implementing Bluetooth or Wi-Fi interfaces in order to be able to communicate to the ELM327 wirelessly.

In figure 2.7 there are two examples of the common implementation of the ELM327. On the left there is an implementation by adapting the RS232 to USB and on the right there is the implementation by adapting the RS232 to Bluetooth. To be used with smartphones, the most common implementations are using Bluetooth or Wi-Fi interfaces. In this project the Bluetooth will be used to communicate with the OBD interface so the application will be developed to use an ELM327 to Bluetooth adapter.

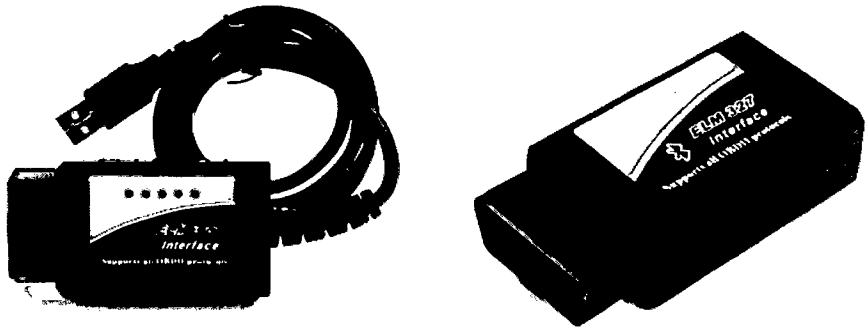

Figure 2.7: ELM327 implementation in USB and Bluetooth

Source: Elm Electronics, 2016

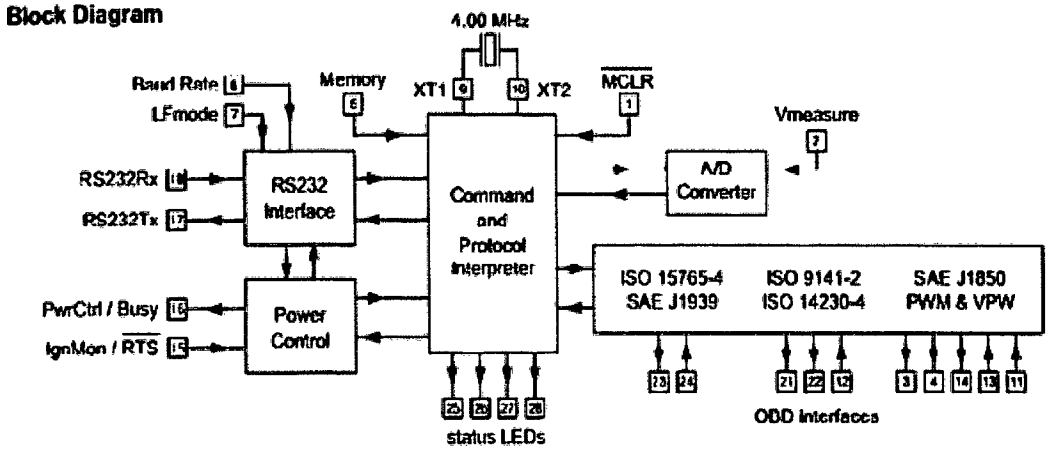

Figure 2.8: ELM327 Block Diagram

Source: Elm Electronics, 2016

The figure 2.8 represents the block diagram of an ELM327 device. As you can see the RS232 interface can either be serial USB or Bluetooth interface. The status LED in the device start glowing when- it is plugged in and is ready to receive commands. It has a small memory which saves data like the protocol being used, query times and so on. The reason for using the ELM327 device is that most importantly it is reliable. Also, the application is intended to be a fuel saving application so, it is wise to choose something that is cheap and reliable at the same time. Thus, the ELM device fulfils both these requirements. There are ELM devices with both USB and Bluetooth communication. In this case, we chose the Bluetooth version since it is more feasible to communicate with the android device and easier to handle for the users. If it was a PC, then the USB version would be appropriate.

2.2.4 J1962 Connector

The OBD-II standardization forced to replace the wide variety of diagnostic connectors into a common 16-pin connector called J1962, which is generally in the same place on most vehicles, under the steering wheel or near the driver seat. In figure 2.9 there is how the connector looks like in a car.

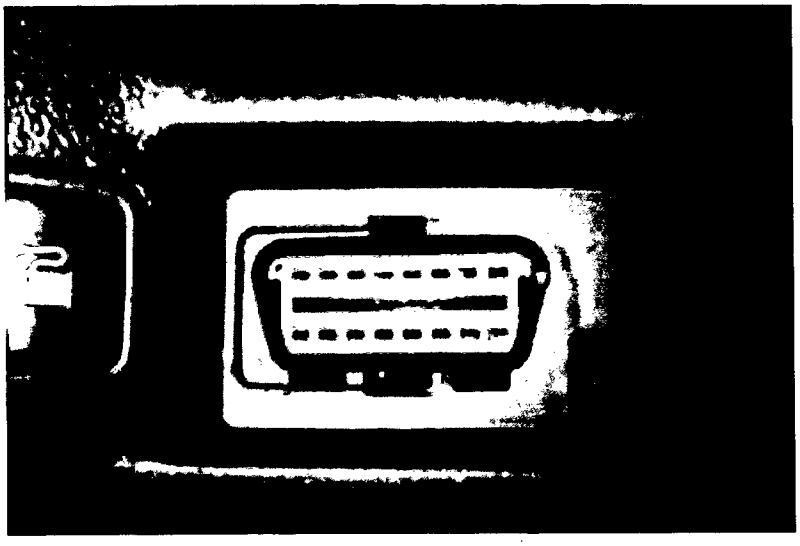

Figure 2.9: The location of the J1962 connector in a Car

As we can see, there are some pins that are manufacturers discretionary, so that, the manufacturers can use them for any purpose. In the J1962 connector there are different pins dedicated to different communication standards. That is because the OBD-II standard allows five different signal protocols. Most of the vehicles use only one protocol, which can be deduced by looking at the J1962 connector and seeing which pins are present and which not (M.J.Kim et al, 2010).

2.2.5 Diagnostics Information

The Engine Control Unit (ECU) is the device that will provide the data to the OBDII/EOBD interface. As discussed by M.Jyothi (M.Jyothi et al, 2012) and Alexander et al (Alexander H., 2013) the method to request the data is defined by the SAE J1979 standard and this standard lists a big amount of parameters that can be available from the ECU. Not all the parameters are required to be implemented and the manufactures can include their own proprietary ones that are not listed in the J1979 standard. To request the parameters through the OBD interface the J1979 standard defines the parameter identification numbers (PIDs) to address them. By requesting and retrieving data through the OBD interface using PIDs, we can have access to real time performance data and to the recorded diagnostic trouble codes (DTCs). As well as the manufacturers can

implement their proprietary parameters, they can also define and implement their own DTCs to freely enhance the OBD code set.

2.3 CAR CONTROL

Car Control application consist of two parts which is the central locking system control and power window control. The main goal is to simulate the real circuit connection and control the system using Arduino microcontroller.

2.3.1 Central Locking System

In some cars that have power door locks, the lock/unlock switch actually sends power to the actuators that unlock the door. But in more complicated systems that have several ways to lock and unlock the doors, the body controller decides when to do the unlocking. Below are some of the ways to unlock car doors:

- With a mechanical key.
- By pressing the unlock button inside the car.
- By using the combination lock on the outside of the door.
- By pulling up the knob on the inside of the door.
- With a keyless-entry remote control.
- By a signal from a control centre.

In the figure below, the power-door-lock actuator is positioned below the latch. A rod connects the actuator to the latch, and another rod connects the latch to the knob that sticks up out of the top of the door. When the actuator moves the latch up, it connects the outside door handle to the opening mechanism. When the latch is down, the outside door handle is disconnected from the mechanism so that it cannot be opened (InfoSpace LLC, 1998).