

## DEVELOPMENT OF OBD-II DRIVER INFORMATION SYSTEM

I. Aris<sup>1</sup>, M.F. Zakaria<sup>2</sup>, S.M. Abdullah<sup>1</sup> and R.M Sidek<sup>1</sup>

<sup>1</sup>Dept. of Electrical and Electronics Engineering, Universiti Putra Malaysia, 43400, Malaysia.

<sup>2</sup>Kolej Universiti Teknologi Tun Hussein Onn, 86400 Parit Raja, Johor, Malaysia

Email: ishak@eng.upm.edu.my

### ABSTRACT

*An on-board diagnostic II (OBD-II) is a standard diagnostic software management that is installed in a power train control module (PCM). It provides some useful data to the driver. There are four different devices using the OBD-II protocol exists in the market. These devices are used as a driver information system. They are personal digital assistant (PDA) Dyno/OBD-II scan tool, CarChip Fleet, DriveRight 600, and ScanGauge. Each of these four devices has some limitation in term of supporting all features for live data monitoring, trip information, diagnostic trouble code (DTC) scanning system, and data logging system. Thus, there is a need for a device that supports all these features together with scheduled maintenance reminder. This paper describes the design and development of driver information based on OBD-II protocol using embedded system architecture. Its hardware and software systems were designed based on four design considerations: upgrade capability, high data storage capacity, back-up capability, and user friendly. The proposed system was successfully interfaced and tested with the PCM of Hyundai Getz car.*

**Keywords:** OBD-II, diagnostic scan tool, trip information, data logging, driver information system

### INTRODUCTION

Every automobile is equipped with an electrical instrumentation panel that is used as a driver information centre, formerly known as a dashboard. It contains various gauges and indicators that provide valuable information to the driver [1]. Gauges provide scaled indication of the system condition such as distance, engine speed, vehicle speed, and fuel level. Whereas, the indicator lights supply information of something is being turned on or warn the driver about system malfunctioning problems. However, this instrumentation panel has limitation in providing more information on specific areas such as malfunctioning problems, trip information, scheduled maintenance reminder and data logging system.

The information displayed on the dash board is retrieved from the electronic control unit (ECU) of the vehicle. The ECU that can provide this information normally has on-board diagnostic (OBD) software (known as a diagnostic management system). This management system performs diagnostic testing, record the result of the tests, and request the test fail actions [2].

Currently, most of the ECUs are equipped with on-board diagnostic II (OBD-II) standard. The OBD-II has been installed inside the engine management ECU (known as PCM) in the United State since 1996. An example of the OBD-II block diagram for General Motors PCM is shown in Figure 1. With the success of the OBD-II, European countries require all petrol cars since 2001 and diesel cars manufactured from 2003 onwards, to equip with European onboard diagnostic systems (EOBD) that equivalent to OBD-II standard. The OBD-II had been standardized by the Society of Automotive Engineers (SAE) and supported by the International Standardization Organization (ISO). This standardization includes common terms and acronyms, a common diagnostic link connector and location, common diagnostic test modes (generic and enhanced), a common scan tools, a common diagnostic trouble codes, and a common communication protocol standard [3].

OBD-II data can be collected through one of five communication protocols that are listed in Table 1. The format of the request command has to follow the SAE J1979 standard (common diagnostic test mode). The OBD-II standards were primarily developed to regulate vehicles emission level standards. Many supporting systems were developed to improve the quality and performance of the instrumentation panel available in a car. Currently, there are four equipments available in the market that can be used as a driver information system that supports some features such as live data (gauges) monitoring, trip information, data logging, and DTC scanning

system. They are PDA Dyno/OBD-II scan tool, CarChip Fleet, DriveRight 600, and ScanGauge. These devices are handy and suitable to be located near to dashboard area.

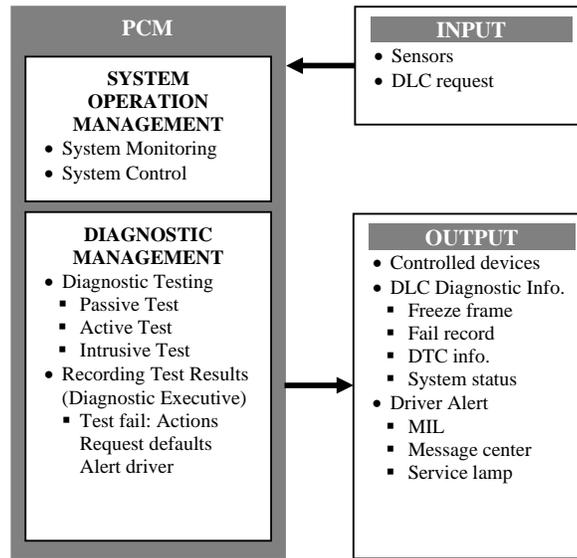


Figure 1: General Motors PCM Management System [2]

Table 1: Communication Protocol [3][4][5]

No	Protocol	Speed
1	SAE J1850 (PWM)	41.6 kbps
2	SAE J1850 (VPW)	10.4 kbps
3	ISO 9141-2	10.4 kbps
4	ISO 14230-4 (KWP 2000)	10.4 kbps
5	ISO 15765-4 (CAN)	500 kbps

The PDA Dyno/OBD-II scan tool is a handheld computer of diagnostic scan tool that enables access to the vehicle’s diagnostic information. Since it was designed for mechanics, therefore it only provides additional data of fuel economy information and selected sensor data logging [6]. The drawback of this system is that it is expensive and no trip information is provided.

CarChip Fleet and DriveRight 600 are more powerful for data logging where they are capable of doing the complete data logging that is good for fleet management [7][8]. The CarChip Fleet can log vehicle and driver performance including DTC and ‘accident’ data for every trip. However, it does not provide trip information except the trip start and stop times. In addition, it does not have live data display on liquid crystal display (LCD). However, its limitation is resolved by the DriveRight 600. The DriveRight 600 has data logging feature similar to the CarChip but does not support for DTC monitoring and recording.

The most suitable equipment for solving the limitation of the instrumentation panel is ScanGauge. It provides the live data display, trip information, and DTC scanning system [9]. In its trip information, it offers the data logging for one trip data of previous day only and does not support for accident data recording. Another downside of this system is that its graphical user interface is not user friendly.

Each of these four equipments has some limitations in term of supporting all features for live data monitoring, trip information, data logging, and DTC scanning system. Thus, there is a need to have a special dedicated system that supports all these systems together with scheduled maintenance reminder. This reminder would replace a current sticker maintenance reminder and prompt drivers about its due maintenance time. The proposed system would be using an embedded system architecture and multimedia card flash memory device to execute its operations and functions. In this proposed prototype system, the ISO 9141-2 is selected as a diagnostic communication protocol for collecting OBD-II data. This is due to the fact that the test car (Hundai Getz) only has ISO 9141-2 diagnostic communication protocol. However, the other protocols can be developed without changing application software of the system because the OBD-II interface hardware is using a universal protocol of OBD-II interpreter chip.

## APPROACH AND METHODS

### System Overview

The driver information system (DIS) is attached in a vehicle through OBD-II connector. Normally, this connector is located underneath a steering. The system will collect, process, and then display the OBD-II data on LCD. It also provides a data logging service. The data logging system uses a multimedia card (MMC) that has to be programmed in .csv format. The file system format of the card is based on FAT16. Thus, the data logging file can be directly opened in the PC via MMC card reader that is attached in USB slot. The file can be viewed either in Microsoft Excel or in Notepad software for further analysis. The general block diagram of the proposed system is depicted in Figure 2.

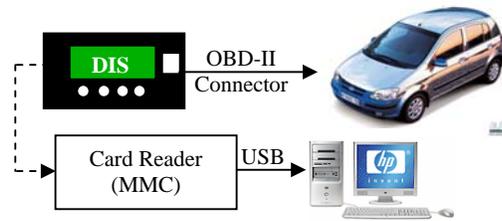


Figure 2: Proposed DIS Block Diagram

### System Requirement

The system requirement is based on output of system features. Figure 3 shows the system features that are divided into two main categories: display and record functions. There are five tasks that would be displayed on the LCD. First, live data task that provides real time gauges reading. The live data is limited to display an engine speed (rpm), a vehicle speed (km/h), a coolant temperature, and a battery voltage reading. Second task is DTC scanning system that offers four functions: MIL and DTC monitoring status, confirmed and pending codes display, freeze frame data (snapshot data during fault occurred), and DTC reset. Third task is trip information that would supply instantaneous or average fuel consumption (exist if the vehicle has mass airflow sensor), elapsed time, and distance for every trip. Fourth task is maintenance reminder that reminds and displays the maintenance schedule according to the vehicle's manual (user's guide) book. In this prototype system, this reminder tells only for engine oil and spark plug replacement. Fifth task is clock system that provides time, day, month, and year information.

In record function, there are three tasks involved in the data logging system. First task is trip information report that provides a summary of every trip. Second, accident log task that records the relevant data for 60s before and after hard deceleration or fully stop. Third task is maintenance reminder data that supply the day and distance remaining of maintenance scheduled.

Beside the system features, the system was designed based on four considerations: upgrade capability, high data storage capacity, data back-up capability, and user friendly. The upgrade capability means the application software can be modified easily without any further modification in system hardware. Meanwhile, the high data storage capacity is necessary to store more data and supporting system data where it can be directly read or written by the PC. Then, the back-up capability system is essential for data logging system, where the accuracy time clock and other recording data are not lost when there is no power. Finally, user-friendly features were designed on the output display and data recording.

The LCD with navigation buttons was designed in a user-friendly manner. The memory is a flash card of MMC that capable of handling huge data recording and providing the program codes. The MMC uses default block setting (512 bytes). Therefore, it needs a buffer memory to handle the temporary data and provide the temporary program codes of the microcontroller unit. The data logging system and clock display need accuracy clock data that is provided by a battery back-up real time clock (RTC) unit. This RTC unit keeps the time and date running when the main power of the system is switched off. The microcontroller unit has capability to collect the OBD-II and clock data through ISO9141-2 interface unit and RTC unit. These two data have to be processed into valuable information and sent to the LCD or data logging system.

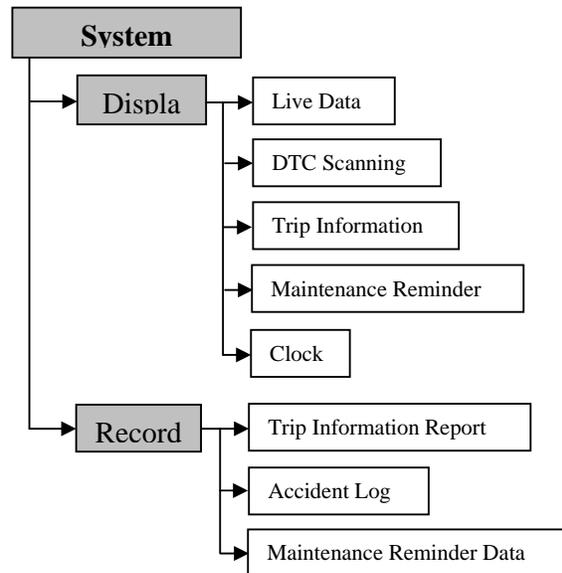


Figure 3: Proposed DIS Features

## System Architecture

The above system requirements give basic idea of designing the proposed system architecture. The system needs two input hardware units of data collection (OBD-II interface unit and RTC unit), one output hardware display of processed data (LCD unit), one user interaction hardware input (navigation buttons unit), one program codes and data recording hardware unit (MMC unit), and one system of buffer memory unit. All of these units are controlled by a microcontroller. The main power supply (5V) for this system is using the vehicle power supply (12V battery voltage) that directly taken from diagnostic connector. The overall architecture of this system is depicted in Figure 4.

The OBD-II interface unit uses an interpreter chip that would automatically establish the communication between the ECU and microcontroller. In this system prototype, the hardware design is limited to support for ISO 9141-2 operation only. Through this interface unit, the OBD-II data can be collected using an UART register function in the microcontroller.

The MMC interface unit is used to let 3.3V of MMC voltage level to be interfaced with 5V of microcontroller. The communication protocol is based on the serial peripheral interface (SPI) and its operation is under SPI mode.

The buffer memory unit is needed to hold the temporary program and data from MMC where the microcontroller assumes that buffer as ROM and RAM. This memory connection is using the same data and address bus with LCD. Thus, the microcontroller assumes that LCD as a RAM unit. The LCD would provide the graphical user interface (GUI) of the system. The GUI is controlled by four navigation buttons.

The time and date would be provided by RTC unit. RTC would communicate with the microcontroller through two serial wires of I<sup>2</sup>C communication protocol.

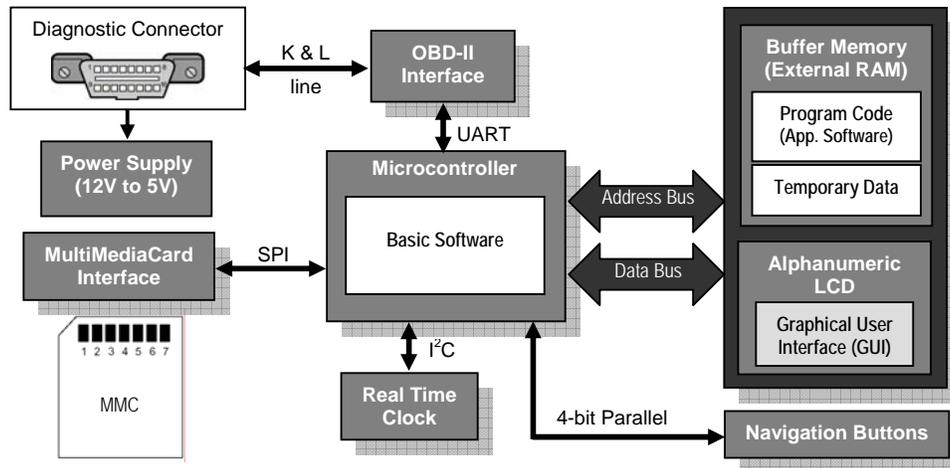


Figure 4: Proposed DIS Architecture

### Software Design

The software of the driver information system was designed and divided into two levels: low-level and high-level software. The low-level is subroutines software that interacts directly to hardware. It has capability to do serial or parallel communication by following the hardware’s communication protocol and flow timing. This software also simplifies the lengthy command especially in MMC operation. The input data will be stored in the buffer memory that later would be retrieved by high-level especially for OBD-II, RTC, and MMC. Besides that, the low-level also provides the initialization subroutine and other subroutines to reduce the complexity of high-level software. The overall software interconnection is illustrated in Figure 5.

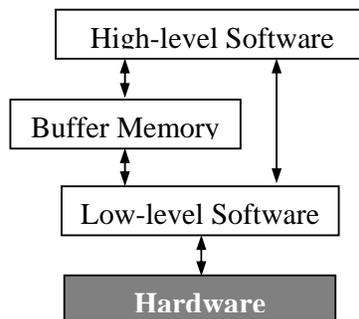


Figure 5: Software Interconnection

The high-level software consists of two-implementation software: basic and application. Basic software is located inside the microcontroller that must be designed with less than 8KB size of program code. The main function of this software is to load the application software that is located in MMC based on FAT16 file system. It is a fundamental to the system, works as BIOS (basic input output system) of current technology of PC system. This software just interacts with LCD, navigation buttons, external RAM, and MMC.

Application software is the main programming of the system, which is run from external RAM with 24KB size of program code allocation. It would interact with the same units of basic software plus two units of RTC and OBD-II. Due to the fact that this software originated from MMC, thus it could be updated easily by replacing or reprogramming the program codes in MMC. The application software was designed based on the system features of display and record (data logging system) functions. The data logging software was designed to handle one file of maintenance reminder, ten file of accident log for ten suspected accident event, and some file of trip information report for every trip. The trip is defined after the vehicle start moving.

## RESULTS

The system prototype was designed and fabricated based on the previous requirement and design. This prototype was tested into two sections: hardware system test and software system test. The hardware testing is co-operated with low-level software and some simple operation testing software. A special test routine written in Assembly Language was developed to carry out the functionality test of the hardware. This testing objective is to verify the hardware circuit and interfacing compatibility as well as the functionality of low-level software. All hardware interfacing were successfully tested.

The software testing was concerned more on to the system operation of the basic software and application software. The basic software operated successfully where it could load the application software at address 2000h in buffer memory. The size of the basic software is 2619 bytes (30% of 8KB internal flash memory) whereas the size of beta version of application software is 8129 bytes.

The prototype system was successfully tested on Hyundai Getz 2004 vehicle model as shown in Figure 6. The test was carried out for one trip cycle.



Figure 6: Proposed DIS is interfaced with Hyundai Getz car's ECU

The .csv file size of the trip information report, accident log, and maintenance reminder are below 512 bytes, respectively. The testing results of trip information summary can be viewed in Microsoft Excel or Notepad as shown in Figures 7 and 8. This trip information file was named as 18050745.csv where the '1805' is date and time captured during engine start (i.e. 18<sup>th</sup> May at 7.45 am).

	A	B	C
1	TRIP SUMMARY		
2	Time Start	AM 07:45:01	
3	Date Start	18-05-2006	
4	Time Stop	AM 08:56:34	
5	Date Stop	18-05-2006	
6	Elapsed Time	[01:11:33]	
7	Distance	56KM	
8	Idling Time	[00:10:21]	
9	Running Time	[01:01:12]	
10	Max. Speed	120KM/H	
11	Max. Tachometer	3200RPM	
12			
13			

Figure 7: Trip Information Summary in Microsoft Excel

```

TRIP SUMMARY
Time Start,AM 07:45:01
Date Start,18-05-2006
Time Stop,AM 08:56:34
Date stop,18-05-2006
Elapsed Time, [01:11:33]
Distance,56KM
Idling Time, [00:10:21]
Running Time, [01:01:12]
Max. Speed,120KM/H
Max. Tachometer,3200RPM
    
```

Figure 8: Trip Information Summary in Notepad

## DISCUSSION

The system has capability to operate as desired and only support to ISO 9141-2 diagnostic protocol. However, this system can be used to other protocols by adding another converter circuit in OBD-II hardware interface unit without any modification on software.

The MMC memory size used in this project is around 128MB (250880 sectors). The MMC is used to hold the program codes and data. The file system (FAT16) needs 528 sectors (270336 bytes), whereas the program code was allocated 60 sectors (30720 bytes). The rest of 250292 sectors are for the data logging system.

In this project, the size of one data .csv file is below 512 bytes. However, in FAT16 file format this file would be allocated into one cluster (2KB). The maintenance reminder file needs one cluster and the accident log file needs ten clusters. The rest of clusters would be allocated to the trip information report that is enough for 25281 trips recording.

The types of data that are displayed on the DIS LCD have been discussed in Section System Requirement. The information is updated during vehicle movement every 5 seconds.

## CONCLUSION

The design and development of a driver information system based on OBD-II protocol using embedded system architecture was presented. It is an accessory tool that provides the information related to the vehicle and trip information to the driver. The proposed DIS supports services such as monitoring live gauges data, trip information, DTC scanning system, maintenance reminder, clock system and data logging system. The system was successfully tested on Hyundai Getz (ISO 9141-2 compatible vehicle). The proposed system is equipped with the 128MB MMC that is able to record for 25281 trip reports.

## ACKNOWLEDGMENTS

The authors would like to thank the Ministry of Science and Innovation, Malaysia for sponsoring this work under CNG/DI Engine & Transmission – Electronic Control Unit (ECU) and Its Diagnostic Kit, IRPA 03-02-04-0561-PR0030/10-06.

## REFERENCES

- [1] Erjavec J. (2005). *Automotive Technology, 4th ed.* New York: Thomson Delmar Learning.
- [2] Hollembeak, B., (2005). *Classroom Manual for Automotive Fuels & Emissions.* New York: Thomson Delmar Learning.
- [3] Society of Automotive Engineer. (2003). *On-Board Diagnostics for Light and Medium Duty Vehicles Standards Manual, (2003) ed.* SAE International: USA.
- [4] Ribbens W. B. (2003). *Understanding automotive electronics, 6th ed.* MA: Newnes.
- [5] Tom N. (2005). *Automotive Protocols & Standards*, Magazine of Motor, July ed. USA: Hearst Business Publishing.
- [6] *PDA-Dyno and OBD II Scan Tool Operating Manual.* (2006). Citing Internet sources URL <http://www.nology.com/pdfandzipfiles/pdadynousermanual.pdf>
- [7] *DriveRight Catalog.* (2006). Citing Internet sources URL [http://www.davisnet.com/catalog/download/PR81\\_0206.pdf](http://www.davisnet.com/catalog/download/PR81_0206.pdf)
- [8] *CarChip Catalog.* (2006). Citing Internet sources URL [http://www.davisnet.com/drive/products/CarChip\\_Catalog.pdf](http://www.davisnet.com/drive/products/CarChip_Catalog.pdf)
- [9] *ScanGauge Manual.*(2006). Citing Internet sources URL [http://www.scangauge.com/SGIIManual .pdf](http://www.scangauge.com/SGIIManual.pdf)