

Efficient Low-power Scanner Free On-Board Diagnostics and Monitoring System for Vehicles

S. Sheraz Mohani
Dept. of Engineering,
Sciences &
Technology,
Iqra University,
Karachi

M. Waseem
Dept. of Telecom.
Engg.,
Sir Syed University of
Engineering &
Technology
Karachi, Pakistan

Mir Asif
Dept. of Electronic
Engg.,
Sir Syed University of
Engineering &
Technology,
Karachi, Pakistan

Zeeshan Asim
Dept. of Electronic
Engg.,
Sir Syed University of
Engineering &
Technology,
Karachi, Pakistan

ABSTRACT

The diagnostics and monitoring of the running condition of a car engine is very important and smart way to enhance the efficiency and maintenance of the engine. In the modern era almost all modern vehicles include an On Board Diagnostic (OBD) system. The OBD system can always monitor the running condition of engine. When the OBD system detects a malfunction in any of sensors or engine parameters, OBD regulations require the Electronic Control Unit (ECU) of the vehicle to save a standardized Diagnostic Trouble Code (DTC) about the information of malfunction in the memory. These DTC's can be processed and read by connecting external scanner to OBD system which then shows the diagnostic results. But this project removes the need of external scanner.

The purpose of this study is to make the information of different sensors of engine and parameters of vehicle freely available to drivers or mechanics on an embedded screen device. This system eliminates the need of any external scanner required to show the diagnostic results of OBD system and solves these problems by providing many useful functions such as real time display of sensor data, diagnostic trouble code (DTC) reading and displaying information of different sensors and parameters of engine in a simple and an easily understandable language for the user (driver).

General Terms

Monitoring system, On-board diagnostics, vehicle engine monitoring.

Keywords

Model-based diagnosis, On-board diagnostics system, vehicle parameters monitoring.

1. INTRODUCTION

On-Board Diagnostics, or OBD, in an automotive context, is a generic term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or a repair technician access to state of health information for various vehicle sub-systems. OBD system consists of a computer chip interfaced with different sensors in the car. These sensors update their information to the controller in OBD system that transfers this data in the form of DTC's (Diagnostic Troubleshooting Codes). This data is then read by the scan tool. Early instances of OBD would simply illuminate a malfunction indicator light, or MIL, if a problem was detected—but would not provide any information as to the nature of the problem. Modern OBD implementations use

a standardized fast digital communications port to provide real time data of sensors, DTC's, which allow one to rapidly identify and remedy malfunctions within the vehicle. But this is achieved by connecting an external scanning tool which is an expensive device and it is not easily decodable in simple language by an owner or a car technician. Comprehensive description of the DTC is done by using software on a PC or laptop.

The aim of the research is focused to design a low cost and low power diagnostic system for vehicles that can monitor the status and process real time data of engine parameters and sensors connected, and also display its status and information on LCD in an easy and understandable language for the user of the vehicle. The proposed system is so designed that it must eliminate the need of any external scanning tool or connection to a PC or laptop. Moreover, the proposed system should be capable of extracting the necessary data from the vehicle in order to use it in a meaningful and useful way by using the mounted sensors.

2. SYSTEM BLOCKS

The building blocks of the proposed diagnostics system model are explained in this section. The backbone of the system is considered to be the communication with the system done using the custom protocols on serial transmission; and the firmware is based on C programming. The system takes sensor-data and sends it to a master controller using a distinct frame format used for identification and information, the data contains.

The OBD (on board diagnostic) frameworks made are perplexing and oblige outer scanner interfaced with no constant powerful implication of the glitch in the vehicle, so the thought is to actualize a prototype framework which gives the client continuous data of the breakdown and which is straightforward. The utilization of DTC's (symptomatic inconvenience codes) will be carried out on inner level and the client is demonstrated the issue in a basic dialect on a showcase. Sensors that are really basic for a legitimate vehicle working will be interfaced including fuel level, temperature, alternator sensing, brake oil level, clutch check, state of brakes and radiator fan state are vital things to be incorporated.

In the proposed model AVR series of microcontrollers is used, which contain built-in peripheral UART and ADC. The initialization and selection of modes for both the peripherals according to the requirement is to be achieved before any further work can be done. The on-board UART in the AVR is

used in asynchronous mode and for the proposed system to work efficiently it is proposed that the ADC is used in single ended conversion mode.

Initially UART attribute setting was unknown that includes baud rate setting, frame format setting, UART running at asynchronous double speed or normal speed mode and synchronization problems. Some of the problems that can be encountered in building a custom made OBD is that ADC testing might yield inaccurate values of conversion results due to wrong setting of pre-scale clock frequency of ADC and the problem in getting 8-bit resolution from 10bit converted result. All these problems were solved by referring to different literatures on AVR microcontrollers and the datasheets of the controllers. According to the industrial standard, programming of microcontroller is to be done in C-language directives. For individuals with experience with assembly language, switching from assembly language programming to C-language is a challenge. For this purpose a major research was done using internet and reference books on C-programming and header files searching and a compatible C compiler for AVR series of microcontrollers.

3. DATA ACQUISITION AND SIGNAL CONDITIONING

The available sensors are analog in nature which changes their resistance against the change in parameter measured. Making the working signal conditioning circuit whose input is compatible for the ADC of micro-controller. For better and accurate results with a wide range of data taken from the sensor, its signal conditioning was required and it could not be used directly with the ADC as it is by default having a very small range of data values. For the operations to be function correctly it is desired to convert analog output values of sensors into a range of 0 to 5 volts for ADC input. For this purpose, proper selection of signal conditioning circuit was to be done. It is proposed that an operational amplifier be used as a difference amplifier which gives output of 0 to 5 volts corresponding to minimum and maximum signal value of sensor, as explained by Table 1 below. When sensors were interfaced with the signal conditioning circuit connected with the ADC of microcontroller, the identification of data coming from different sensors can be considered as a major task to complete. Figure 1 and 2 shows the signal conditioning circuits and Figure 3 shows the interconnections between signal conditioning circuits with the ADC.

Table 1. ADC results for different input values

Analog Input (volts)	Digital value (Output of ADC) (8 bit value)	Calculated value of input based on digital output
0	00000000 = 0	0
0.5	00011000 = 24	0.468
1	00110100 = 52	1.01
1.5	01001000 = 72	1.4
2	01100100 = 100	1.95
2.5	01111010 = 122	2.38
3	10010111 = 151	2.94
3.5	10101101 = 173	3.38
4	11000110 = 198	3.86

4.5	11011111 = 222	4.33
5	11110110 = 246	4.8

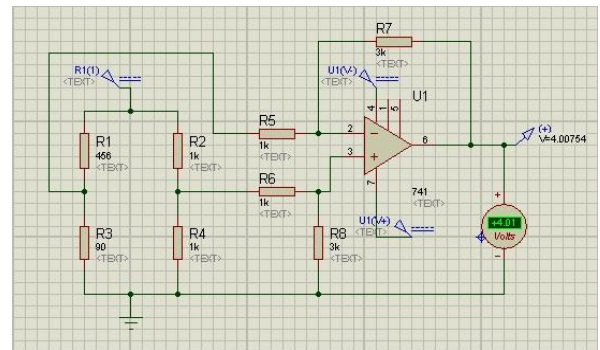


Fig 1: Signal conditioning circuit when temperature is high

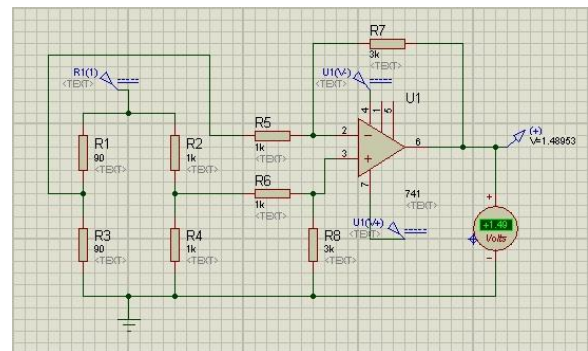


Fig 2: Signal conditioning circuit when temperature is Low

When different sensors and parameters are to be sensed, they need identification so that data from each sensor is sensed distinctly and can be displayed well. For this purpose, a unique code is sent from sub controller to main controller before sending data frame of that sensor. In this way main controller gets to know that the upcoming data frame will be the data of which sensor.

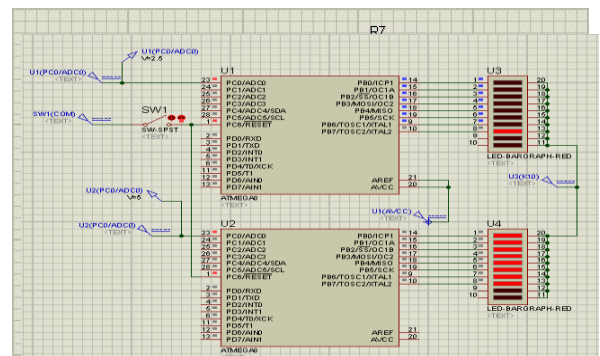


Fig 3: The output of signal conditioner is fed to ADC for analog to digital conversion of data

To add a graphical user interface instead of the old scanner method, the proposed system uses a (20*4) character LCD to display the information relevant to the working of the system providing a great experience to the user how OBD system is performing different operations. This feature adds a great

interest of the user as it shows the sensor which the system is using, its result and indicates its state as “ok” or “not ok” or “high” or “low” etc. This new feature is a great milestone to be achieved as it is for the first time that an OBD system is claiming a scanner free system.

One particular hurdle in the development of the prototype model can be the information of codes required for write operation on LCD, the working of LCD controller, the character support and the address location of each block in which a character can be written. Detailed study on LCD controllers, reading datasheets of different character LCD are available options to achieve a modern use of LCD. Moreover, another problem can be in reading the busy flag of LCD (Reading busy flag condition is essential because of the fact that LCD processing speed is slower than microcontroller). Literature shows that one method to solve this problem is by adding delays between different I/O operations on LCD instead of reading busy flag.

The access of internal EEPROM of Atmega 162 is required by the system when the capability to save last obtained results is to be added to the system. For this, 512 bytes internal memory of the controller is accessed through the C-language programming and results are saved in the EEPROM whenever they are obtained from the sensors. This feature makes OBD a kind of black box for the car which can save the last results obtained from the car sensors.

Allocation of memory blocks was required in which data of different sensors is to be saved. Then a complete read/write algorithm was to be integrated within functions dealing with I/O operations on sensor. This would allow real time saving of data in EEPROM in allocated memory blocks. This problem was solved by making a write and read function separately for EEPROM and then while taking data from sensors, these functions were called to save and retrieve saved data from EEPROM when required.

4. EXPERIMENTAL RESULTS

The proposed model was backed up by practical demonstration, for which it was to install the proposed system on a car. A working OBD is required to be shown on a car giving real time results so that it can perform functions it is designed for, on a real car. There are certain problems involved as there is a difference in the car voltage level (12V) and the system voltage level (5V). To solve this problem, a relay box is made to effectively isolate different voltage lines and get the required output voltage level required for proposed system from the different car parameters, as shown here by figures 4 and 5.

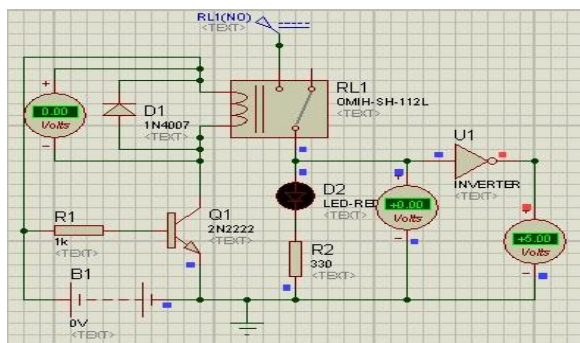


Fig 4: Circuit when Relay is OFF

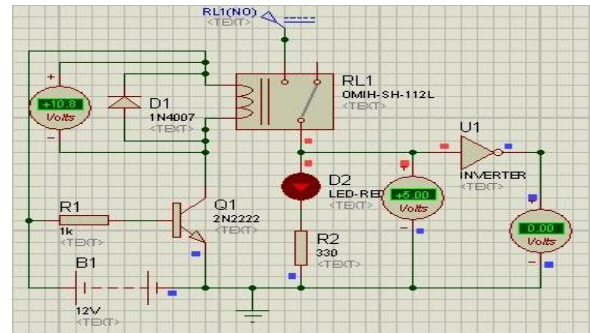


Fig 5: Circuit when the relay is on

Secondly, wires were required from the sensors installed in the car. For this the description of wires from the alternator, brake oil level sensor and connections from compressor and brakes should be known. The experimentation was successful completed with installation of system achieved on Suzuki Khyber car.

The whole main controller function is to be implemented on a 32bit MPC5554 processor but due to its unavailability, we did it on 8-bit RISC microcontroller. Experiment shows that enhancing controller enhances the efficiency by many folds. The enhancement was implementing the main controller functionalities using 32bit processor or controller. So to achieve this, SPARTAN 3E STARTER KIT containing X3s500 FPGA from Xilinx was chosen. Serial port and LCD of the board is programmed using EDK(Embedded Development Kit) tools. The soft core processor used is Micro-blaze which is a 32bit RISC architecture processor. By using the Micro-blaze, LCD and DTE (Data Terminal Equipment) serial port, we will enable the FPGA kit to act like the main controller as it functions and by connecting the sub-controllers with FPGA, complete OBD system can be implemented.

5. CONCLUSION

This paper introduces automobile diagnostic technique and on-board diagnosis (OBD) system. A low power, efficient and scanner free, automobile fault diagnostics system based on OBD system design scheme is presented in this paper. The performance of each module is tested on vehicles, and the results show that the communication of this system is reliable, the diagnostic result is accurate, and the performance is good and easy for the human to interact with the system.

6. REFERENCES

- [1] Grimaldi, C. and Mariani, F., "OBD Engine Fault Detection Using a Neural Approach," SAE Technical Paper 2001-01-0559, 2001, doi:10.4271/2001-01-0559.
- [2] Jim Luyckx, Mohan Sethi, Mike Clancy. "Completely integrated diagnostic platform using industry standard PCs and internet update". SAE 2004-01-0674
- [3] Yang Hai, Feng Jing,Zhou Xing-li. "Development of multifunctional diagnosis instrument for electronically controlled diesel engine" Vehicle Engine Feb.2006 No.1
- [4] Tian Ying, Chen Pei-hong, Nie Sheng-fang. "Development of ECU malfunction diagnosis instrument for electronic control engine". Feb.2005 No.1

- [5] Wang Xian-wen. "OBDII-based automotive scanning monitor". Techniques of Automation & Application
- [6] California Air Resources Board "Technical status update and proposed revisions to malfunction and diagnostics system requirements application to 1994 and subsequent California passenger cars, light-duty trucks, and medium-duty vehicles-(OBDII)". CARB Staff Report, 1991.
- [7] SAE.J1850 "Class B data communication network Interface".SAE, 1994
- [8] ISO/WD14230-1, "Road vehicles diagnostic system keyword protocol 2000 physical layer".1997
- [9] ISO/WD14230-2, "Road vehicles diagnostic system keyword protocol 2000 data Link".1997
- [10] ISO/WD14230-3, "Road vehicles diagnostic system keyword protocol 2000 application layer".1997