Driver Safety Assistance System

Introduction

The goal of this project is to build an embedded system that can be interfaced with any OBD-2 enabled car. This system by extracting the real time diagnostic and performance data from On-board diagnostics (OBD) provides status of the various vehicle subsystems and guides the driver to develop safe driving habits.

Motivation

The idea came from the "snapshot device" which is developed and deployed by auto insurance companies for measuring the driver's behavior and differentiate the auto insurance rate depending on how safely the driver drives. Since 1996, all cars sold in the United States have been required to implement the On-board diagnostics (OBD-II) standard so our system is capable of communicating with almost all the cars irrespective of the manufacturer. This on-board diagnostic provides all the information needed to monitor and provide feedback so that it helps the driver to improve his driving habits.

The project is done in a two members team and the hardware part is implemented by me and the software parts are implemented by team member Taejoon Byun.

Learning outcomes

1. Learnt to design a real time embedded system to extract data from cars OBD-II connector.
2. Learnt how to communicate and interact with multiple microcontrollers by interfacing ATMEGA32 microcontroller with ELM327L programmed microcontroller
3. Learnt to interface ATMEGA32 microcontroller with 4x2 LCD
4. Build circuits for voltage regulation – Developed a Circuit to extract 12V power supply from the car battery and convert it to 5V, which is needed for ATMega32, ELM327L and MCP2551
5. Learnt to establish communication by interfacing ELM327L with CAN transceiver - MCP2551
6. Understanding procedures involved in Embedded System Development - circuit design, debugging

Product Description

Project is based around a chip ELM327L, that is designed to handle OBD-II interpretations. It converts 5 vehicle communication formats in to ASCII strings and it transmits the data in RS232 serial format. Block diagram how different components are connected in the project is shown below in figure 1.

Extracting data from cars On-board diagnostics: The OBD-II standard-SAE J1962 specifies the type of connector, pinout, signaling protocol and messaging format which is standard across vehicle manufactures. Block marked 1 in figure 1, represents OBD-II connector provided in the car which enables interface to diagnostics data of the car. Since this is a female OBD-II connector, it is necessary to use a male connector to get access to its pins. Because this has additional pins which provide, manufacturer specific detail which is not necessary for this project, OBD-II to DB9 converter is used which give access to only the signaling protocols and power. Voltage regulator MC78L05BP is connected to pin 9 of DB9 connector which provides access to car battery.

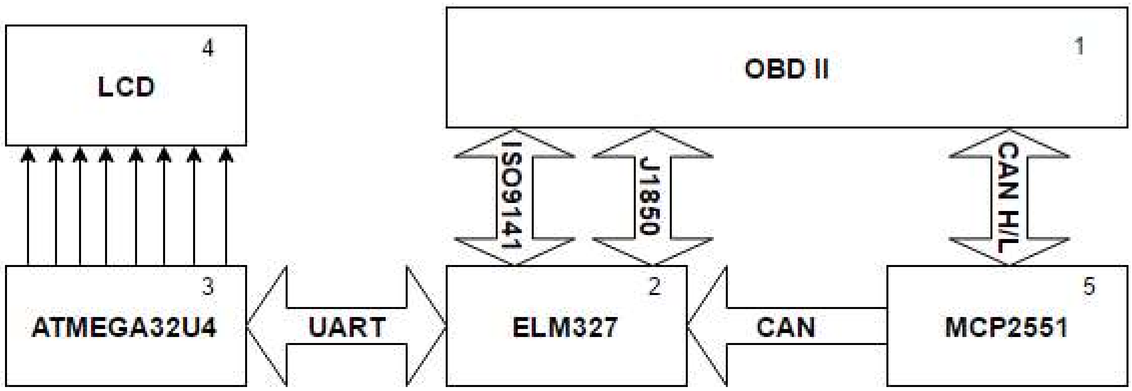

Figure 1 Block diagram of devices connected

Interfacing ELM327L with CAN transceiver - MCP2551: The signals extracted via a OBD-II connector is passed on to OBD-II interpreter ELM327L (marked 2 above in block diagram). Apart from Power and Ground signals, DB9 connector has CAN Low and CAN High signals which contains diagnostic information. CAN H and CAN L signals use Differential signaling method to transfer the data, where the signal is represented by a differences between CAN L and CANH. If the difference is positive and larger than a certain minimum voltage it is a "1" and if the difference is negative it is a "0". Since this generates a differential output, there is a need of CAN protocol controller as an interface between physical bus and ELM327L. The MCP2551, a high-speed CAN transceiver (marked 5 above in block diagram) is used for this purpose.

Interfacing ATMEGA32 microcontroller ELM327L: Once the data is available in ELM327L, required data can be accessed by using AT commands. The microcontroller ATmega32 (marked 3 above in block diagram) is interfaced with ELM327L as shown below in figure 3, communication schematic. ELM327L and ATmega32 controller communicates with each other using standard RS232 serial communication. Rx pin of ELM327L is connected to Tx pin of ATmega32 and Rx pin of ATmega32 is connected to Tx pin of ELM327L. Both the devices are set to same baud rate - 38400 baud. While connecting the power supply, it is necessary that both the microcontroller and ELM327 are powered on with the same power supply. It is also important to note that, even though ATmega32 is operating with 16MHz crystal and ELM327L operating with 4 MHz crystal, serial communication can be established by setting same baud rate both the controllers.

Interfacing ATMEGA32 microcontroller with 4x2 LCD: The LCD (marked 4 in figure 1, block diagram) is configured for 8-bit character display and following pins are connected to the connections mentioned in the table.

Pin	Connections	Description
1	V_{SS}	Ground
2	V_{DD}	Supply voltage for logic (+3.3V)
3	V_0	Supply voltage for contrast
4	RS	Register Select signal. RS=0: Command, RS=1: Data
5	R/W	Read/Write select signal, R/W=1: Read R/W:=0: Write
6	E	Operation Enable signal. Falling edge triggered.
7-10	DB0-DB3	Four low order bi-directional three-state data bus lines.
11-14	DB4-DB7	Four high order bi-directional three-state data bus lines
15	A	Backlight Anode (+3.0V)
16	K	Backlight Cathode (Ground)

Figure 2 Interfacing LCD display with the micro controller

As explained above, overall communication schematic is as shown in figure 3

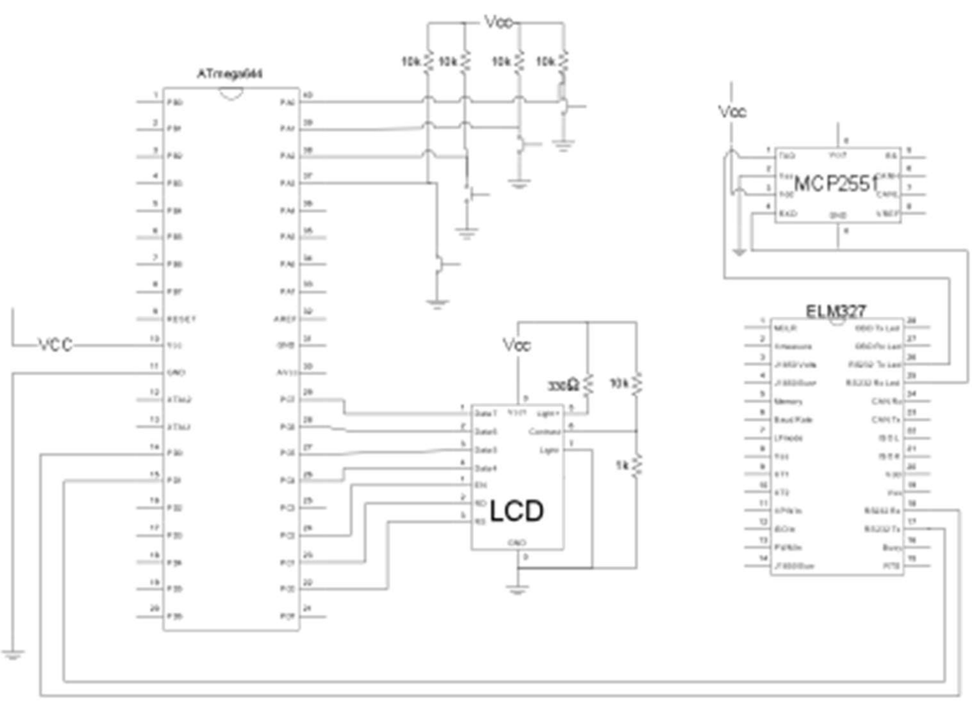

Figure 3 Communication Schematic

Voltage Regulation: The microcontroller target board which consists of ELM327L, MCP2551 CAN transceiver require 5V for its their operation. So the voltage regulator MC78L05BP is used to regulate car's 12V to 5V. A voltage of approx. 0.1 V should be connected to V0 pin of the display as shown above to adjust contrast. 1M Ohm potentiometer is used to adjust V0 voltage to 0.1 V. Circuit containing all the connections is as shown in figure 4 OBD-II to hardware schematic.

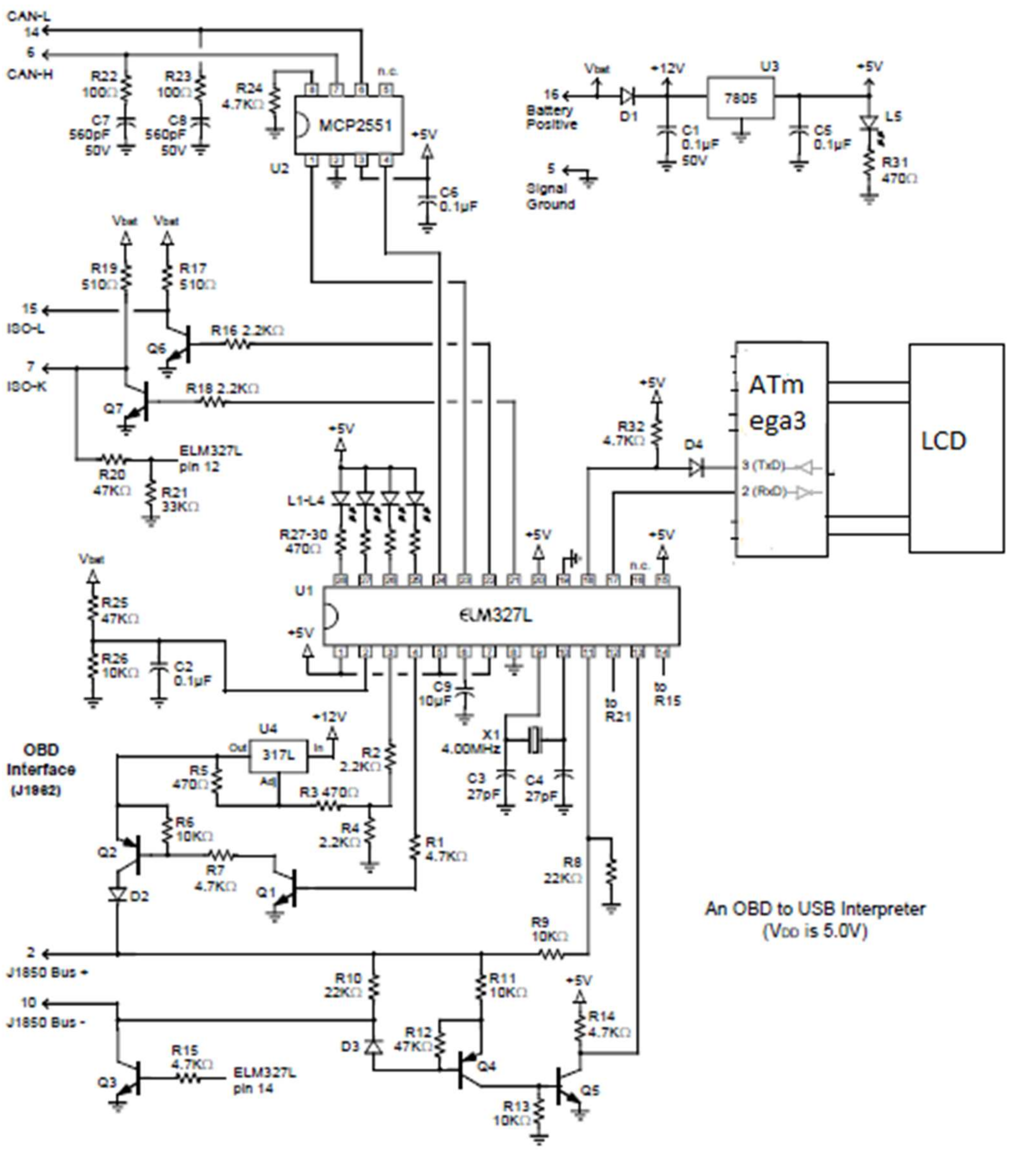

Figure 4 OBD-II hardware schematic

As shown above in figure 4, ELM327L accesses the data from ODB-II via CAN, ISO and J1962 messaging protocol. Accessed data is transferred to micro-controller using RS232 serial communication protocol and in turn it is displayed on an LCD. The circuit shown at the top right corner is a voltage regulator which provides power to the components.

Difficulties faced during project execution:

1. Since the supporting circuitry needed for ELM327L to communicate has many components and is complex, PCB development was challenging. This complexity resulted the risk of short circuit.
2. During the testing phase, after completing initial interface testing between ELM327L and ATmega32, OBD-II in the car stopped functioning possibly because of short circuit. Which became a major roadblock to test the implementation.

Because of these difficulties, the outcomes of the project were different from the objectives. Due to the unavailability of the actual data from the car, device is not programmed to provide any feedback to the driver and the data extraction was not achieved. However, the communication between ATmega32 and ELM327 was establiched where ELM responds to the command sent by the microcontroller via RS232.

Learning objective not realized and Future Work

1. Establishing communication between ELM327L and CAN transceiver - MCP2551.
2. By using OBD-II data such as Vehicle speed, Accelerator pedal position, engine idle times, warning codes, product should check for aggressive starts and hard braking and guide the driver to develop safe driving habits.

The project implements following topics introduced in the course:

1. Setting up UART Serial communication – ELM237L communicates with ATmega32 microcontroller using serial communication.
2. Busy waiting technique is used in serial communication where AT commands are sent to ELM327L requesting data and wait for some amount of time and display connection failed in case ELM327L des not send back the data.
3. Program is designed in such a way; it accesses those data which are likely to change quickly very frequently and other parameter which are less likely to change less frequently. For example, Vehicle speed is more likely to change compared to mileage. So those parameters, which are more likely to change are assigned high priority and other are assigned low priority.
4. Working with digital I/O – Port B is configured as output to display LCD characters. Various 0 and 1 combinations of 8 channels of poet B display characters as per the requirement.
5. External 4MHz oscillator is setup to provide clock signals to the microcontroller ELM327L.
6. Configuring and using timer to generate waiting delay before printing an error message on LCD in case connection fails.

References

1. Cornell University Final Projects: OBDII Project https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2009/ama64_maa66/ama64_maa66/index.html
2. Progressive® Snapshot® device: https://www.progressive.com/auto/snapshot/
3. OBD (On-board diagnostics): https://en.wikipedia.org/wiki/On-board_diagnostics
4. OBD-II PIDs - http://obdcon.sourceforge.net/2010/06/obd-ii-pids/