Techno-Science

Scientific Journal of Mehmet Akif Ersoy University

www.dergipark.gov.tr/sjmakeu

WEB BASED TRACKING OF VEHICLE FAULT AND PERFORMANCE DATA ON OBD II

Ahmet Ali SÜZEN[1*], Kıyas KAYAALP[1]

[1] *Department of Computer Technologies, Applied Sciences University of Isparta, Turkey*

ARTICLE INFO

Article History

Received : 24/09/2018
Revised : 30/09/2018
Accepted : 30/09/2018
Available online : 30/09/2018

Keywords

Arduino
OBD II
Vehicle tracking
Web Based

ABSTRACT

OBD (On-board Diagnostic) is the standard used by the SAE (Society of Automotive Engineers) in 1988 to monitor the data flow of sensors from external sources. OBD was updated as OBD II standard in 1990. Nowadays, monitoring of the flow of sensor data in vehicles is done via OBD-II standard. The OBD-II connection, which must be used in all vehicles, is a window that opens out of a vehicle.In this study, the data received via OBD II via Bluetooth was transferred to the Web via Arduino Uno development card. The error codes and speed-fuel information produced by the vehicle during the journey were transferred to the software in the Web environment with the arduino with wifi module. The software developed with ASP.NET converts incoming error codes into meaningful texts. This allows the first intervention to be carried out by the experts in the event of a failure during the driving. The internet required for the operation of the system is provided by sharing the personal access point of the driver's mobile device internet (hotspot).

1. INTRODUCTION

Electronic control units (ECU) used in vehicles are also the black box of the vehicles. Parameters for vehicle safe use and the parameters to be used in undesired situations during vehicle use are stored on the ECU. For a vehicle equipped with an ECU, situations such as the occurrence of an error, temporary or permanent condition, or the components that affect it, whether it affects the use of the vehicle are important and such information is continuously recorded and the driver is informed on the trip computer screen in the vehicle [1].

The ECU also controls precise control of the engine's main features, such as ignition timing, fuel injection functions, variable valve timing, idle speed, as well as other vehicle systems such as the transmission, brake and steering systems. ECU's total control of these systems never means that the system will work perfectly. Both external factors and internal factors can cause temporary or permanent problems in vehicles. Some of these problems may not interfere with the operation of the vehicle. However, as soon as possible to diagnose a possible fault is vital for both driving safety and vehicle safety [2].

It is always important that the faults that occur in the vehicles are given in a simple way. It is a basic principle to give general information instead of instantaneous details for the drivers, but only for signs of faults. The details of the fault are important only for the expert repairman. However, some faults may not be directly related to the part of the fault. This situation can be compared to human diseases. The symptom of a disease can be seen as a condition of an organ that is not directly related to the disease. The situation is the same for faults occurring in vehicles. It is not necessary for the driver to be informed in detail by the vehicle in such cases as an expert is required for this to be understood [1-2].

The output required for the transfer of this error information stored in the ECU is OBD, which is an abbreviation for “On-Board Diagnostic” words [1]. The output required for the transfer of this error information stored in the ECU is OBD, which is an abbreviation for on-board diagnostical words. Although the history dates back to 1988, especially in all vehicles produced in 2000 and later, there are OBD connectors. The OBD connector is the external door of the ECU, a black box of a vehicle. The OBD connector used today is called the OBD-II or OBD-2. The OBD-I connector is not standardized between

* Corresponding Author: ahmetsuzen@isparta.edu.tr

To cite this article: Suzen, A., Kayaalp, K. (2018). Web based tracking of vehicle fault and performance data on OBD II. *Techno-Science* vol. 1, no. 1 p. 13-16.

firms, and therefore the OBD-II connector, which is a standard (universal) connector in 1996, has started to be used [3].

Communication with OBD-II can be wired or wireless. OBD-II adapters are used for this purpose. In this study, a commercially available OBD-II adapter with bluetooth communication is used. The main purpose of this study is to create a Web based fault tracking system by means of data collection device which we designed by using Arduino development card, bluetooth module and wifi module.

2. METHODS

2.1. Service-PID codes for OBD-II

The SAE J1962 is an accepted standard for the OBD-II connector and communication. For OBD-II, two connection types A and B type have been determined. The type A connector is used for 12V and the type B connector for vehicles using 24V supply voltage. Two connector types are shown in Figure 1. Both consist of 16-pin (2x8) D-shaped connectors [4].

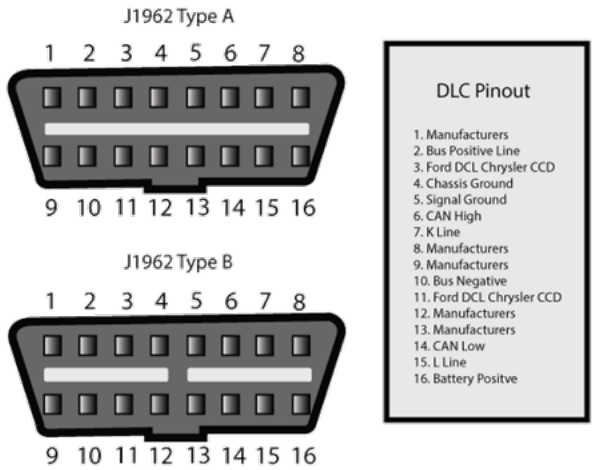

Fig. 1. *OBD-II connector wiring* [5]

As shown in Figure 1, the OBD-II connector has battery (power supply), communication and manufacturer pins. The manufacturer pins are special pins that allow certain special data to be received and that the ECU can be partially programmed, which changes according to each manufacturer and acts as a second key.

No significant direct data can be obtained through the ECU. For this purpose, a special parameter identification (OBD-PID) has been created for OBD. The SAE J1979 standard for OBD-PID has been adopted [6].

A total of 10 service parameters were determined in 2017 for OBD-PID. These service parameters are hex codes that can be requested from the ECU, such as displaying the available data, displaying the registered diagnostic error codes, clearing diagnostic error codes and stored values, displaying the test results, testing vehicle components, displaying vehicle information, displaying permanent diagnostic error codes. Service parameters may vary depending on the manufacturer. PID hex code is generated after service hex code (Table 1). The service and PID hex code correspond to the requested response from the ECU again as the hex code. Further information can be found in Ref [6].

Table 1. *Service-PID hex code examples* [6]

Service-PID hex code	Description	Formula
01-04	Calculated engine load (%)	$\frac{100}{255}A$
01-0C	Engine rpm (rpm)	$\frac{256A+B}{4}$
01-0E	Timing advance (° BTDC)	$\frac{A}{2}-64$
01-5E	Engine fuel rate (L/h)	$\frac{256A+B}{20}$

The 4-byte A, B, C, and D hex values come from the ECU according to the Service-PID hex code. The values of A and B in Table 1 refer to desimal numbers. The obtained values of A and B are converted to decimal numbers to obtain the desired information. C and D values are used in other service-PID hex codes.

2.2. Bluetooth adapter for OBD-II

The most common OBD-II adapter on the market is bluetooth communication. These adapters are called ELM-327, which is the code used for the integration. MCP-2551 is used for communication of ELM-327 with CAN. The communication with the ECU can be achieved with these two integrated tools. Wired or wireless communication can be used to transmit this communication to an external device. The adapter used in this study is ELM 327 integrated and has bluetooth communication [7].

Fig. 2. *Bluetooth OBD-II adapter* [8]

It is possible to find many software used with OBD-II adapter. These softwares generally show specific instant data. Manufacturer's software is designed to perform all operations on ECU. Figure 3 shows various software interfaces.

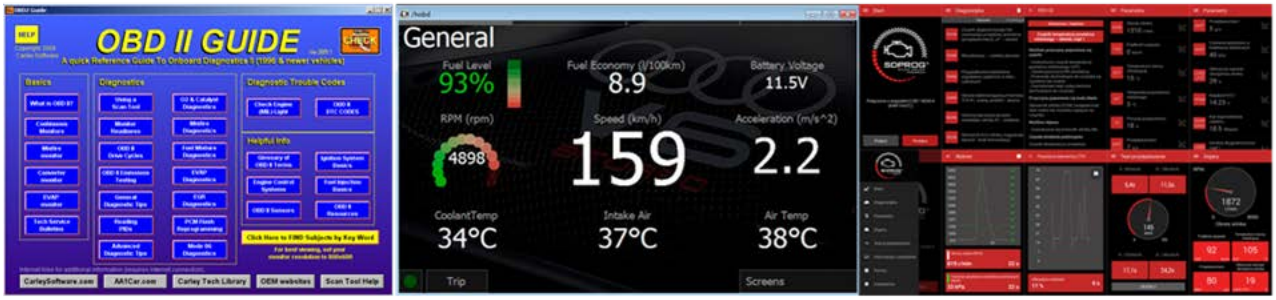

Fig. 3. *OBD-II softwares* [9-11]

In this study, a simple controller and software which can communicate with the OBD-II adapter via bluetooth and transfer the data obtained via wifi to the remote device has been created.

2.3. OBD-II Control Device

In this study, vehicle data were requested at 1 minute intervals via Arduino. The data was transferred to Arduino with the OBD-II Bluetooth adapter pluged into the vehicle. The data were processed with arduino and recorded in the database on the Web. Thanks to the Web interface, data recorded in the database can be analyzed. The HC-05 bluetooth module is used for communication with the OBD-II bluetooth adapter and the ESP-8266 wifi module is used for data transfer to the Web (Figure 4).

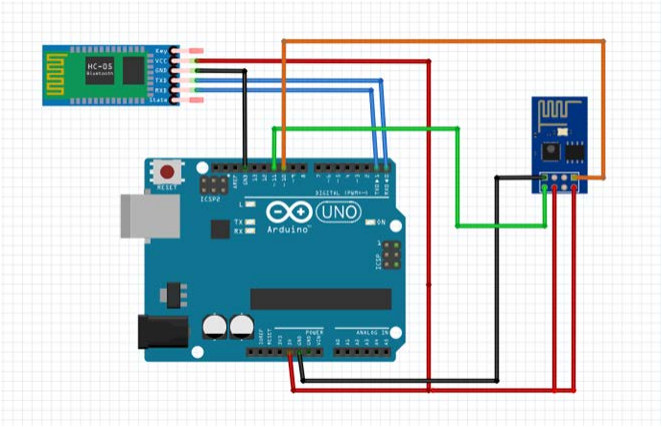

Fig. 4. *OBD-II control device*

For the Arduino software to work, the Initialization operation must be performed first. ELM327 works with AT (Attention) commands. Therefore, it is necessary to perform the Initialization process with AT commands and make the device ready. On the Arduino IDE, this process was performed with the following code block (Figure 5)

After the service-PID hex code sent to the ECU, the ECU receives a response. For example, when the hex code 01-5E is sent, a hex-code, aa-bb-cc-dd, is returned. The values here can be converted to decimals to obtain the required values A, B, C and D for the formula. In order to obtain the registered error codes on the vehicle, serial port listening is performed. A 5 digit error code (XXX XX) comes from this listening. The error definitions corresponding to this error code are listed in the Diagnostic Trouble Code (DTC) list as standard and total 3170 for year 2017. Error codes can also be defined by matching them on the Web. A DTC code is encoded as shown in figure 6.

```
SoftwareSerial elm327(0, 1);
      elm327.begin(9600);
      elm327. println ("ATZ");
            delay(2000);
      elm327. println ("ATEO");
            delay(1000);
      elm327. println ("ATL0");
            delay(500);
      elm327. println("ATH1");
            delay(500);
      elm327. println ("ATSP 5")
            delay(500);
      elm327. println ("01 00");
            delay(500);
```

Fig. 5. *AT code block*

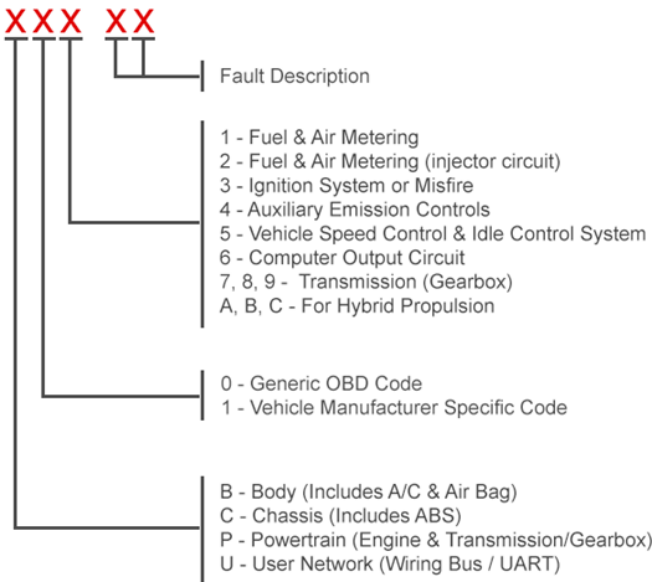

Fig. 6. *Explanation of the DTC codes* [12]

3. RESULTS AND DISCUSSION

It is provided to monitor the data coming from the vehicle in the Web interface of the actual study. The web interface is developed using C# programming language with ASP.NET. The vehicle data received with the Arduino development card is displayed as a querystring

parameter as shown in on the web page with the wifi module:

http://localhost:60904/DataSave.aspx?Gas=10&EngineTemp=40&Power=2000&Speed=0&DTC=P0157

These parameters are taken from the browser address section and saved in the database. Technical data can be accessed via the interface via the interface shown in Figure 7.

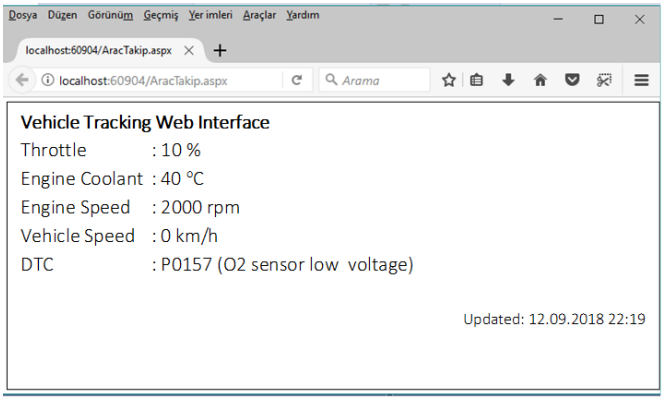

Fig. 7. *Vehicle tracking Web interface*

4. CONCLUSIONS

The work carried out via OBD-II bluetooth device allows the monitoring of data about the vehicle on the Web with the help of Arduino. With this vehicle tracking software, it is possible to control the DTC error codes, speed, temperature and power stored in the vehicle in web environment. The system provides quick and accurate measures to prevent possible problems during driving or when the technical expert is away. It is aimed to provide emergency measures against the error codes during vehicle driving with expert support. In addition to this, it is planned to carry out a process such as making a transaction that the driver can do about the error codes obtained by using artificial intelligence algorithms in future studies. Today, when autonomous vehicles are developed, such applications are expected to become a necessity in the near future.

ACKNOWLEDGEMENTS

This paper was developed and derived from the paper presented by the authors at the 1st International Vocational Science Symposium.

REFERENCES

[1]. Baek, S. H., & Jang, J. W. (2015). Implementation of integrated OBD-II connector with external network. *Information Systems*, *50*, 69-75, DOI: 10.1016/j.is.2014.06.011.

[2]. Yang, L., Zhang, S., Wu, Y., Chen, Q., Niu, T., Huang, X., ... & Hao, J. (2016). Evaluating real-world CO2 and NOX emissions for public transit buses using a remote wireless on-board diagnostic (OBD) approach. *Environmental pollution*, *218*, 453-462, DOI: 10.1016/j.envpol.2016.07.025

[3]. OBD Advisor, *https://www.obdadvisor.com/difference-obd1-obd2-scanners/,* accessed on 2018-09-20.

[4]. SAE J1962, *https://www.sae.org/standards/content/j1962_201207/,* accessed on 2018-09-20.

[5]. Obd Connector Wiring, *http://osomeweb.com/obd-connector-wiring.html*, accessed on 2018-09-20.

[6]. SAE J1979, *https://www.sae.org/standards/content/j1979_201202/,* accessed on 2018-09-20.

[7]. ELM 327, *https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf,* accessed on 2018-09-20.

[8]. OBD-II, *https://tr.pinterest.com/pin/387802217909976705/*, accessed on 2018-09-20.

[9]. OBD Software-1, *https://scantoolcenter.com/blog/the-unique-obd-nissan-software/,* accessed on 2018-09-20.

[10]. OBD Software-2, *http://www.archer-soft.com/en/blog/7-best-obd-apps-your-car,* accessed on 2018-09-20.

[11]. OBD Software-3, https://www.elexpress.pl/product-eng-480-Diagnostic-program-SDPROG-OBD2-WIN-iOS-Android-KOD-ID39A.html, accessed on 2018-09-20.

[12]. OBD-II DTC List, *http://www.totalcardiagnostics.com/support/Knowledgebase/Article/View/21/0/complete-list-of-obd-codes-generic-obd2-obdii--manufacturer*, accessed on 2018-09-20.