American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-12, Issue-3, pp-162-172
www.ajer.org
Research Paper Open Access

An experimental study for reading dynamic data of vehicles

Trinh Ngoc-Hung[1] and Nguyen Dinh-Dung[2, *]
[1] *Faculty of Vehicle and Energy Engineering, Le Quy Don Technical University, 100000 Hanoi, Vietnam*
[2] *Faculty of Aerospace Engineering, Le Quy Don Technical University, 100000 Hanoi, Vietnam*
** Corresponding Author: Nguyen Dinh-Dung,*

ABSTRACT: *The operation of vehicles, particularly cars, is continuously monitored through physical parameters (intake-air temperature, engine speed, coolant temperature, etc.). These parameters are determined and transmitted through sensors to the vehicle's central control unit through the vehicle's internal communication protocol. This paper presents the design of the vehicle's dynamic data reading circuit through the OBD-II interface. This circuit is designed based on the Can Bus Shield circuit and Arduino. In real-time, this circuit observes and records parameters such as engine speed, vehicle speed, and engine temperature - these parameters are used for further studies related to monitoring, analyzing, and diagnosing possible failures. Based on the test results, it can be concluded that the vehicle's dynamic data reading circuit operates steadily and can identify automobile parameters in real-time.*
KEYWORDS: *Can bus shield, OBD-II, Diagnostic Tool, Fault detection, Arduino.*

Date of Submission: 10-03-2023 Date of acceptance: 23-03-2023

I. INTRODUCTION

The purpose of obtaining dynamic data is to help users get information when driving a vehicle. Every car is equipped with a dashboard that is used as a driving information center. Gauges provide information about the system status, such as distance, engine speed, vehicle speed, and fuel level. The car's dashboard display provides valuable information that warns the driver about system malfunctions, reminds him/her of scheduled maintenance, and memorizes data in the system. When a car is damaged, the ECU, which receives signals from the sensors, will report the error code "Check Engine" to the control panel, helping the mechanic detect the system's error codes and fix them quickly.

Modern OBD-II gateway systems have become more complex and recently adopted a standardized digital communication port to provide real-time data and a standard series of diagnostic error codes. Modern motor vehicles are highly sophisticated machines in combination with electrical, electronic, and mechanical engineering developments. Traditional "trial and error" diagnostics are no longer effective in meeting such vehicles' maintenance and repair needs. Therefore, this poses a challenge in keeping up with technological trends, and therefore, it is imperative to apply a new diagnostic method with high efficiency on new-generation motor vehicles. The research presents the design of the vehicle's dynamic data reading circuit through OBD-II gateway systems to facilitate and improve the early detection of faults and malfunctions related to emission control components.

Researchers worldwide have deployed vehicle dynamic data monitoring to improve efficiency, engine utilization, and vehicle performance. In research[1], Gilman demonstrated a driver assistance system- called Driving Coach that monitors specific parameters to increase fuel efficiency depending on the vehicle type. Meanwhile, Szalay et al. used two different reading methods (CANBus and FMS CANBus) and concluded that these measurements are almost identical[2].

To prevent problems from happening and diagnose potential failures, Kushiro et al. built a diagnostic model based on the correlation between error codes through the OBD (On-Board Diagnostics) interface[3]. Sik has used CAN and OBD circuits combined with GPS to create a predictive model that helps drivers choose routes to avoid traffic jams or find parking spaces[4]. To save fuel and reduce emissions, D'Agostino used data from the OBD on the hybrid vehicle to determine the mode of operation with the electric motor or the internal

combustion engine[5],[6]. With the development of science and technology, more and more studies have been carried out on monitoring and remotely controlling vehicle parameters in real time[7], [8], [9].

So, there needs to be more research on using dynamic data parameters for in-vehicle diagnostics via the OBD-II port. This study presents an experiment to get dynamic data from sensors through the ECU via the OBD-II port to serve fault diagnosis of system clusters on cars, which offers many new insights as we programmed our code on Arduino software, calculated and selected the communication circuits with the vehicle's Can bus network. We also retrieved dynamic data sets to PC to monitor and diagnose car parameters. Like the engine and powertrain assemblies, error and error-clearing codes are displayed on the "Check Engine" control panel.

This paper presents the calculation and selection of dynamic data reading circuit to directly retrieve sensor parameters from the ECU of Toyota Vios 2016 using the OBD-II protocol. The Can Bus Shield and Arduino circuits were utilized to receive data from the OBD-II port on the medium that sends the data through the computer for analysis. This device allows people to observe and record dynamic data sets and evaluate the measured signal values from the sensors for technical diagnostics.

II. METHODS AND MATERIALS

2.1. Principles of reading dynamic data

The OBD-II PID (Vehicle Diagnostic Parameter ID)- a code for requesting data from the vehicle was used as a diagnostic tool.

The SAE J1979 standard defines multiple OBD-II PIDs. All vehicles and trucks sold in North America must support a subset of these codes, primarily for state-mandated emissions tests. Manufacturers also define additional PIDs specific to their vehicles.

In 1996, light vehicles (under 8,500 pounds or 3,900 kg) were the first to be authorized, followed by medium vehicles (14,000 pounds -8,500kg or 3,900 pounds - 6,400 kg) in 2005. Both must be accessed through a standard data link connector defined by SAE J1962.

Heavy vehicles (more than 14,000 pounds or 6,400 kg) manufactured after 2010 for sales in the United States are authorized to support OBD-II diagnostics through SAE Standard J1939-13 (circular diagnostic connector) under CARB in title 13 CCRD 1971.1. Some North American heavy-duty trucks use the SAE J1962 OBD-II diagnostic connector, commonly used for passenger cars, especially Mack and Volvo Trucks. However, they use the 29-bit CAN identifier in contrast to the 11-bit header used by passenger cars). There are 10 diagnostic modes described in the latest OBD-II SAE J1979 standard (Table 1). Prior to 2002, J1979 referred to these services as "modes".

Table 1. Ten diagnostic modes according to SAE J1979

Service/Mode (hex)	Describe
01	Show current data
02	Freeze frame data display
03	Display stored diagnostic error codes
04	Clear diagnostic error codes and stored values
05	Oxygen sensor monitoring and test results (not only CAN)
06	Other component/system monitoring and test results (Oxygen sensor monitoring and testing results for CAN only)
07	Display pending diagnostic error codes (detected during the current or last driving cycle)
08	Control the operation of the vehicle component/system
09	Request vehicle information
10	Error code (DTC), (DTC cleared)

During the research process, we selected vehicle-specific parameters for monitoring, analysis and diagnosis. These parameters are shown in Table 2.

Table 2. Selection of parameters for monitoring, analysis and diagnosis

PID (hex)	PID (Dec)	Data bytes returned	Description	Minimum value	Maximum value	Units	Recipe
04	1	1	Calculated engineload	0	100	%	$\frac{100}{255}A$
05	2	1	Engine coolant temperature	-40	215	^{0} Celsius	A - 40
10	3	2	Massair flowsensor(MAF) air flowrate	0	655.35	Gram/sec	$\frac{256\,A+B}{100}$
11	4	1	Throttle position	0	100	%	$\frac{100}{255}A$
14	5	2	Oxygen sensor	0	1.275	Volt	$\frac{100}{255}A$
0F	6	1	Intake air temperature	-40	215	^{0}Celsius	A - 40
5C	7	1	Engine oil temperature	-40	210	^{0}Celsius	A - 40

While researching and testing cars, we have selected 07 parameters as follows: Calculated engine load, the time before the top dead center of the engine, intake air flow sensor, and running time after starting the engine. Monitoring and analyzing engine speed sensors, vehicle speed, and engine cooling temperature through observation and comparison between the display panel and computer are coincident. When any sensors are removed from the system, the signal will disappear and vice versa. The purpose of dynamic data collection is to help us monitor and analyze from the minimum to maximum value threshold. An error signal will be sent to the system if the value exceeds the allowed limit.

2.2. Dynamic data reading system

The dynamic data reader receives the signal from the car through the OBD-II port. Thanks to the Can Bus Shield circuit, it communicates with the Can bus network in the car. It processes the signal transmitted to the Arduino circuit. The center receives the signals sent to the computer to monitor and evaluate the dynamic data set for the technical diagnostic process (Fig. 1).

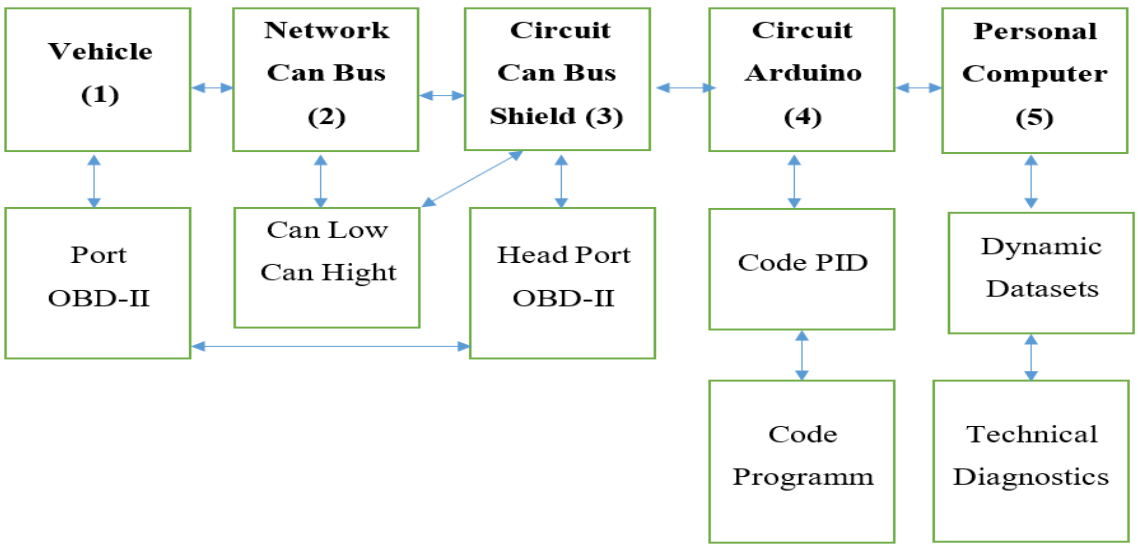

Fig. 1. Diagram of dynamic data communication principles

A car's inner facilities are similar to a human body. The CAN (Controller Area Network) communication network is a two-wire high-speed serial network technology. As the most commonly used communication network in cars today, the CAN bus is a central nervous system that allows electronic communication and control (ECU) like parts of the body. Information is sensed by signals from sensors mounted around the vehicle via the CAN bus (including two wires, CAN Low and CAN High) to control car systems (Fig. 2).

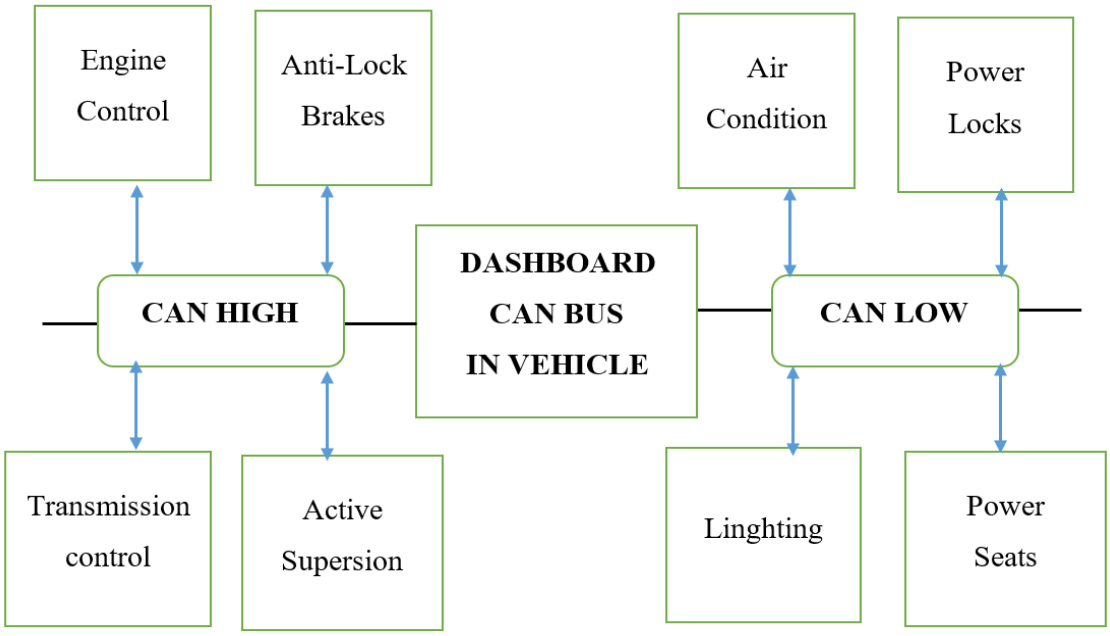

Fig. 2.CAN bus network in vehicle

In this study, the device reads the OBD-II dynamic data via CAN Bus Shield and uses the Arduino platform to transmit the data through the computer. The interface responsible for controlling the CAN bus signals on the car is defined by the CAN Bus Shield for Arduino (Fig. 1).Through the Can bus circuit, the signal management output on the vehicle is CAN LOW and CAN HIGHT (Fig. 2).

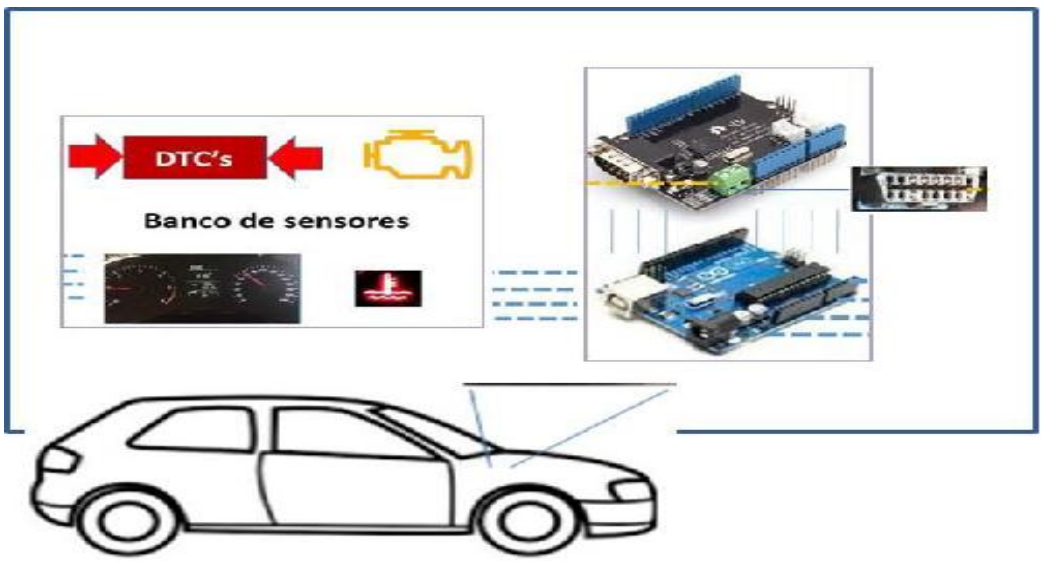

Fig. 3. System overview diagram

The general diagram of the developed system is divided into two parts: the data collection phase and the graphical interface to visualize and monitor some dynamic parameters for technical diagnostics (Fig. 3).

The input signal from the car via the OBD-II port is transmitted through the CAN bus network through two wires, CAN Low and CAN High. The Can Bus Shield circuit is where the input signal communicates with the car CAN bus network through the Arduino circuit using the CAN controller- MCP2515 with SPI interface. The Arduino circuit is the central circuit responsible for receiving the output signals and controlling the pre-processing of CAN messages. It also has a CAN transceiver- MCP2551 that processes signal from cars, as shown (Figure 4).

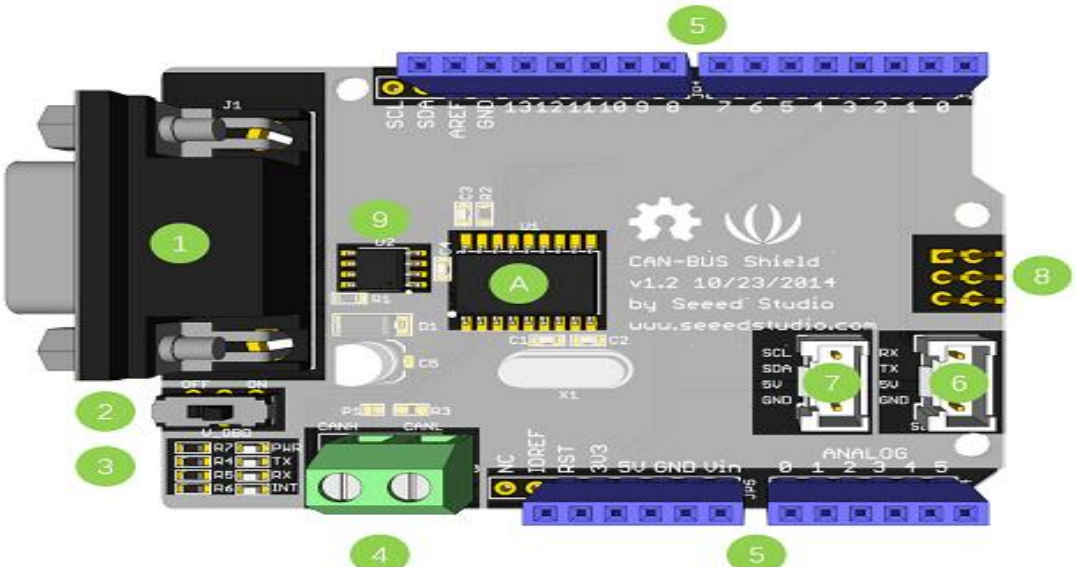

Fig.4. CAN Bus Shield

1- DB9 Interface connects to OBD-II Interface via DBG-OBD Cable; 2- V_OBD gets its source from the OBDII Interface (from DB9); 3- Led indicator includes PWR: power, TX: flashing when data is being sent, RX: flashing when data is being received, INT: data interrupt; 4- Terminals CAN_H and CAN_L; 5- Output pin Arduino UNO; 6- Serial Grove connector; 7- I2C Grove connector; 8- ICSP pins; 9- IC - MCP2551, high speed CAN transceiver; 10- IC - MCP2515, standalone CAN controller with SPI interface

The Arduino circuit (Fig. 5) is an output signal connected to a PC through a port combined with Code programming language and OBD PID encoding to send signals to the Can bus on the car. Through the ECU, the returned sensor signals are displayed on the computer through the dynamic data acquisition device described above.

Fig. 5. Arduino circuit

The Arduino circuit will send data signals back and forth from the car through the Can Bus Shield converter circuit to the computer to process the signals. The Arduino Uno is an electronic board based on the ATmega328P microprocessor. The circuit has 14 digital input/output pins (six of which feature PWM), six

analog inputs, a 16 MHz quartz crystal, a USB connection, a DC power outlet, an ICSP header, and a restart button.

III. EXPERIMENTAL SETUP

3.1. Hardware of the dynamic data reading system

The dynamic data reading circuit hardware is designed based on the CAN Bus Shield circuit (Fig. 6) and Arduino. The dynamic data reader will connect the Can Bus to the Arduino. The ODB Interface represents the device when collecting data from the ECU (Engine Control Unit) for technical diagnostics.

Fig. 6. The overview of the dynamic data reading system

3.2. Software and algorithm

Arduino needs external software executed in the Computer to write programs on the development board. This software, commonly known as the Arduino IDE, is relatively simple and based on processing. The latter is an open-source programming language and Java-based IDE and is often used as a vehicle for teaching interactive and multimedia digital design projects.

LabView software is used to get the results from the Arduino to display the monitoring interface, monitor the dynamic data set for technical diagnosis and display the car's malfunctions (Fig. 7).

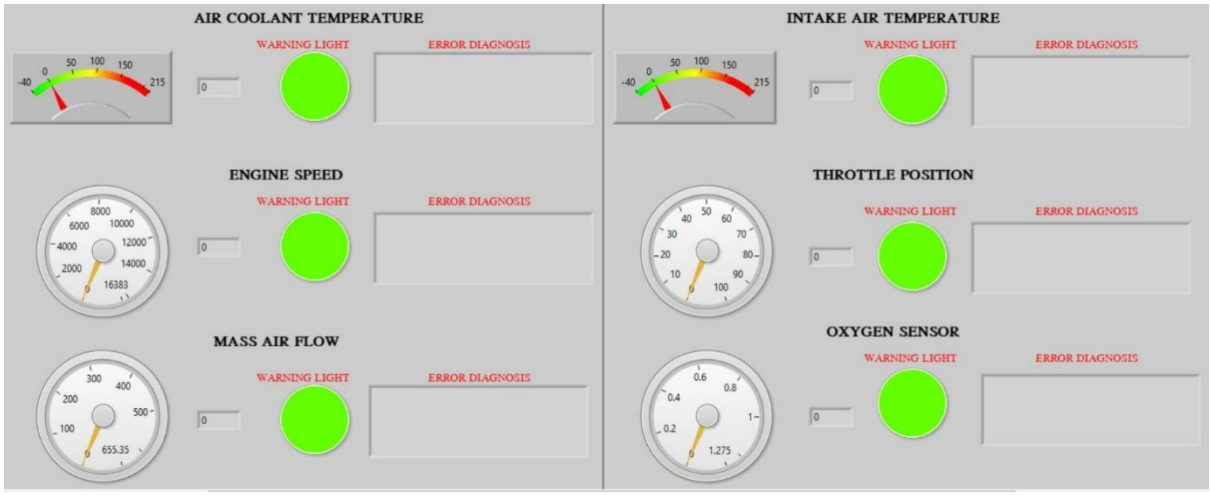

Fig. 7. Displaying dynamics parameters

The algorithm used for the circuit to read dynamic data through the OBD-II port is shown as follows:

Step 1: Declare access parameters to ECU

Step 2: Put the Can command on the car's LAN (Can Low and Can High) through the OBD-II port.

Step 3: Set the Baud rate and the Void taskCanRec command to initialize the baud rate of the Can Bus system.

Step 4: Set the mask (Init_Mask) and filter (Filt Unsigned char) to ensure the data from the device reaches the endpoint.

Step 5: Check the received data (CheckReceive Void).

Step 6: Get ID on the device via ECU (Get CanId Void).

Step 7: Send data onto media (CAN.sendMsgBuf).

Step 8: Receive data and print parameters and values (Serial.Print)

Step 9: Display the results on the graph (Serial Plotter).

Step 10: Export the data to the computer screen shown through the interface.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In our testing, we used a 2016 Toyota Vios to retrieve data from the ECU via the vehicle's OBD-II transmission (Fig. 8). The CAN Bus Shield circuit and the Arduino platform were used to transmit data through the computer to get the values of the parameters. The test was carried out in three cases: i) Increase engine speed while the vehicle is stationary, ii) Increase vehicle speed, and iii) Vehicle on the road. The ambient temperature was 16 at 9 a.m, and the entire project lasted for 15 minutes or 900 sec.

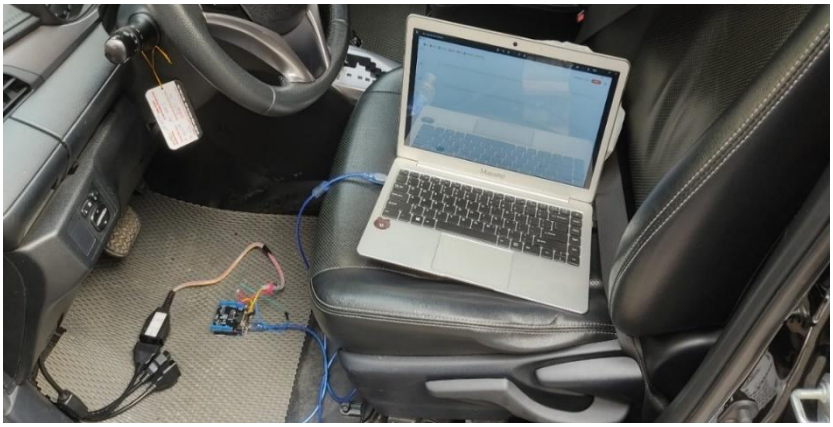

Fig. 8. Experiment test on Toyota Vios 2016

The experimental results are shown in Figure 9-11 as follows:
Case 1. Increase engine speed while the vehicle is stationary

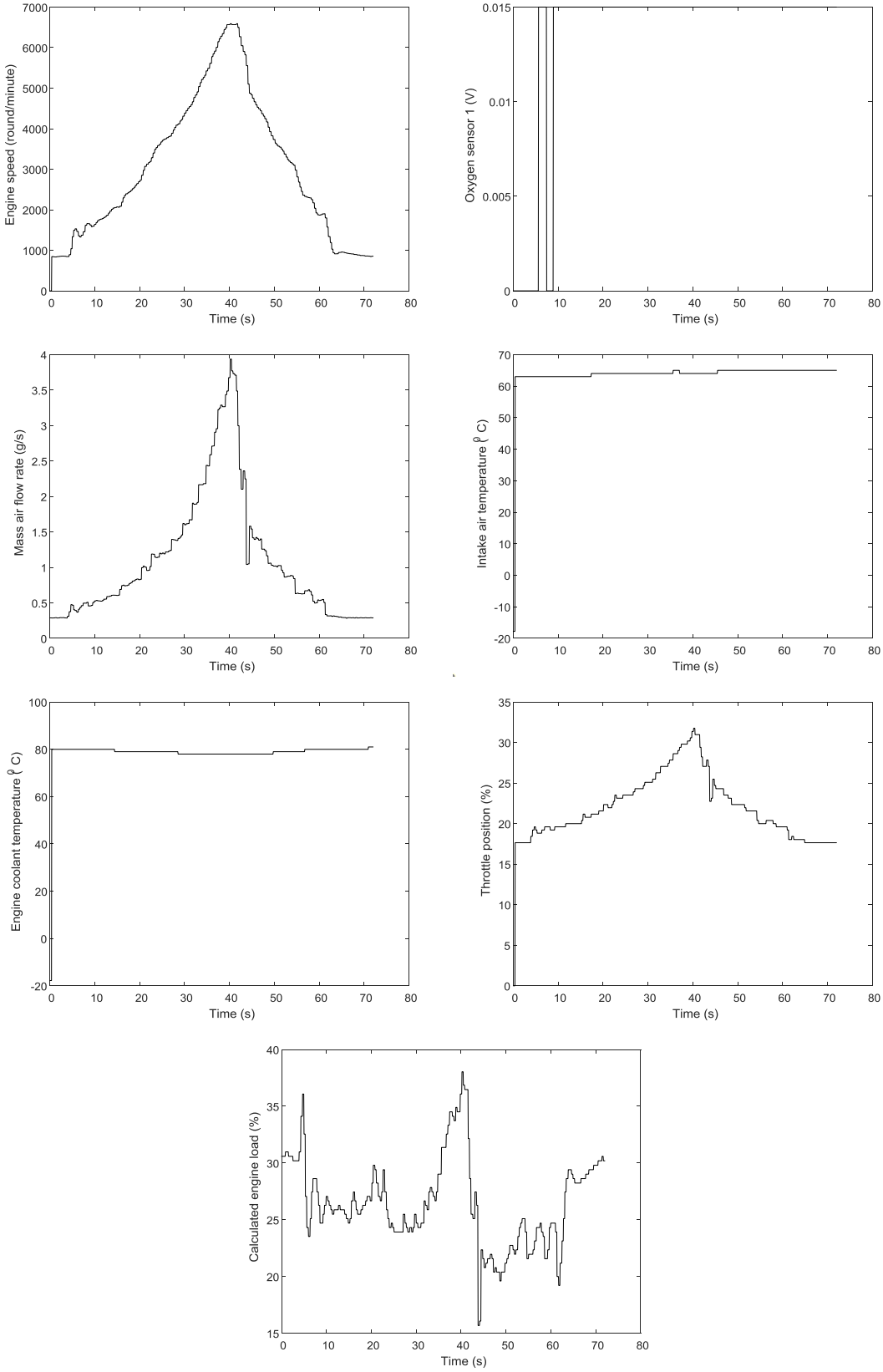

Fig. 9. All parameters in case 1

Case 2. Increase vehicle speed

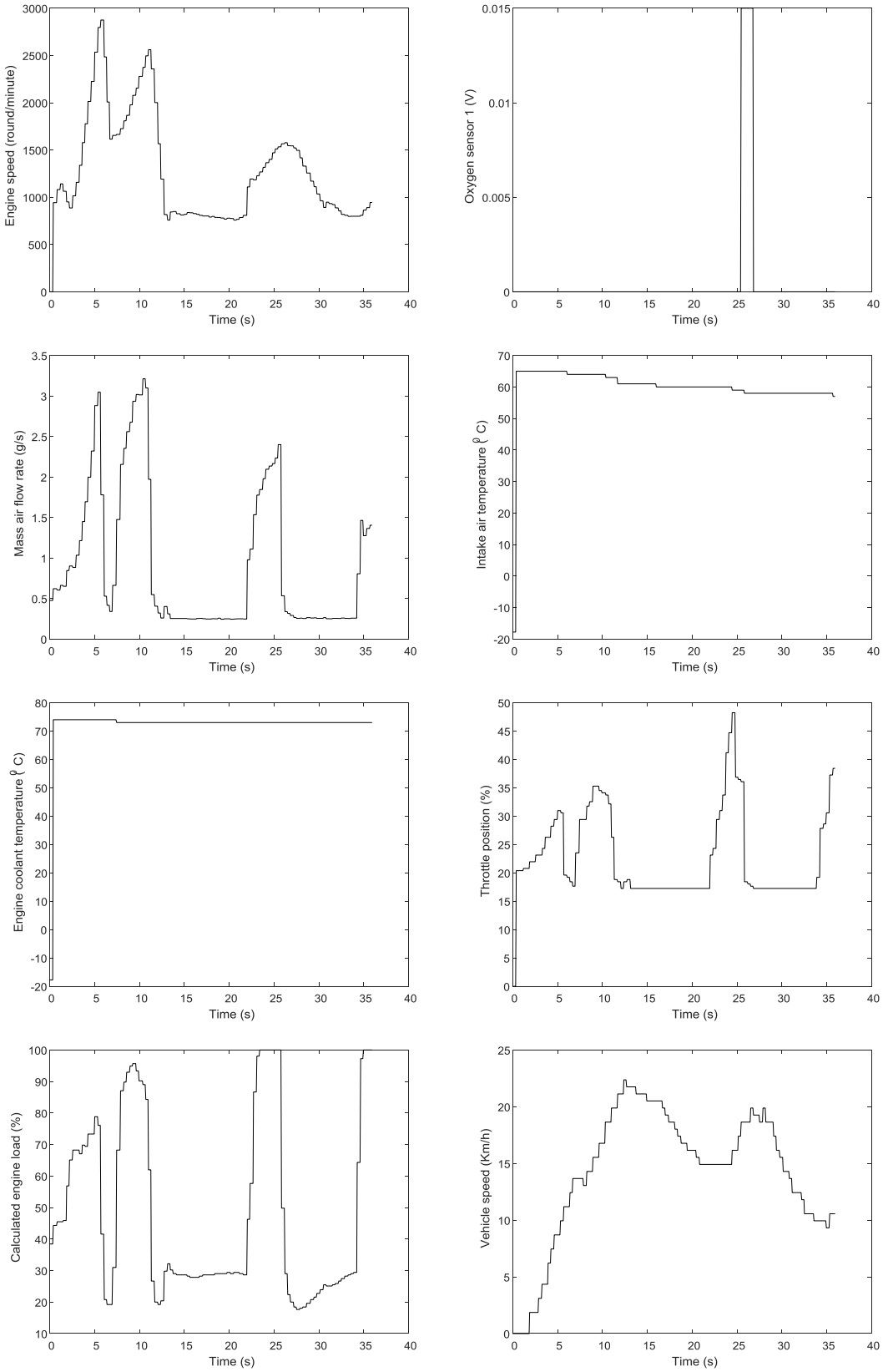

Fig. 10. All parameters in case 2

Case 3. Vehicle on the road

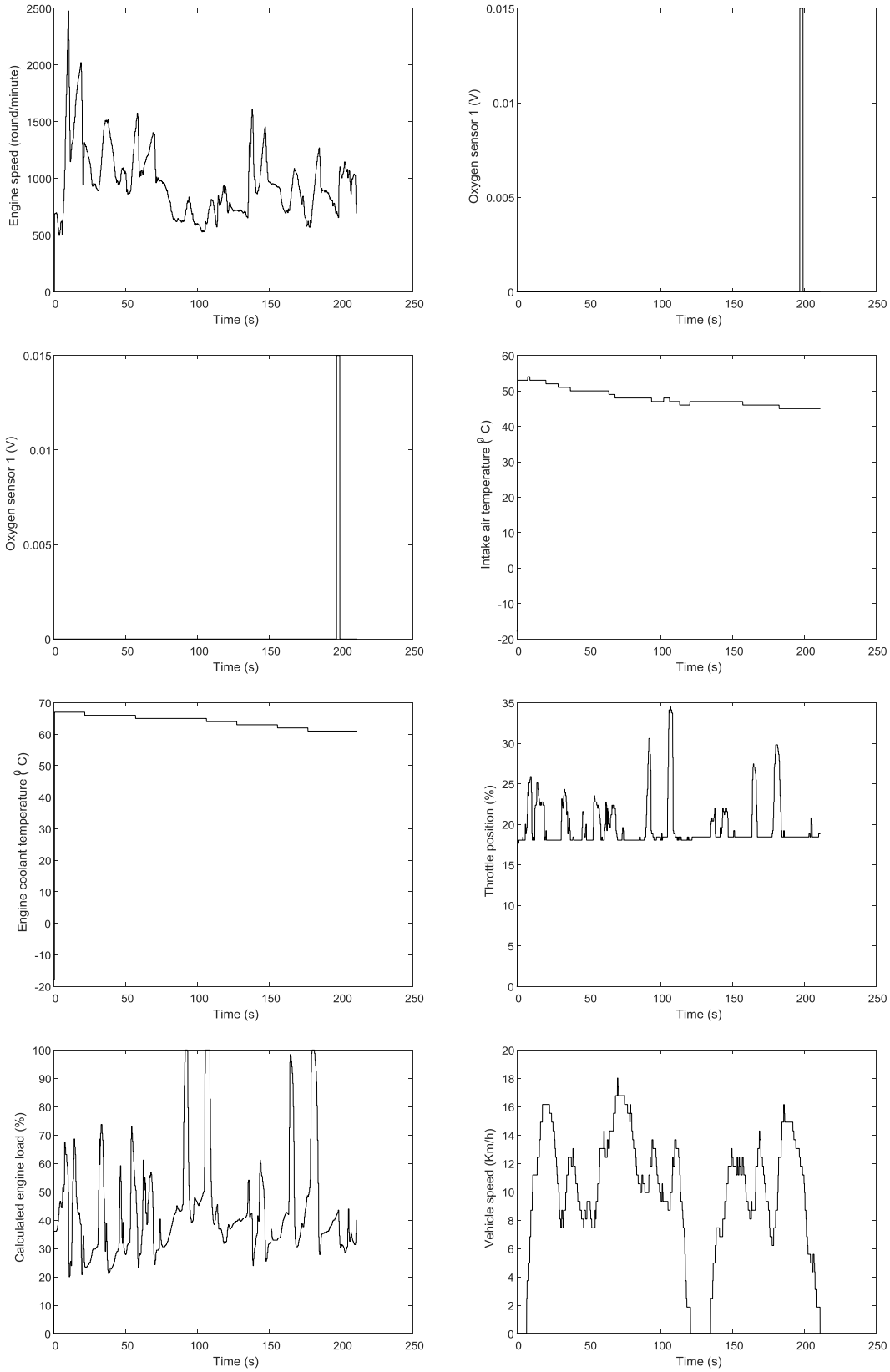

Fig. 11. All parameters in case 3

The calculated engine load and coolant temperature were monitored during testing, comparing these parameters on the computer and the vehicle's dashboard. The results show that these parameters on the dashboard and the computer are the same, which shows that the dynamic data reading circuit is working correctly.

When the vehicle moves with the calculated engine load ranging from 20% to 70%, it is considered the time when the average throttle position opens from 18% to 25% as the vehicle is not accelerating on the road. The measured mass air flow rate ranges from 0.3g/s to 1.5g/s at a time when the oxygen sensor voltage is 0V. When the intake air temperature around50^0C then the average coolant temperature is from 60^0C to 68^0C.

During the test, the car was driven on both roads and highways. The test results show that the values of the parameters displayed on the computer (via the dynamic data reading circuit) and on the control panel are the same, specifically, the engine load parameters, throttle position, and coolant temperature.

Although this study has achieved specific theoretical and experimental results, there are still some problems in the research process, such as the Arduino circuit that frequently burns and the occasionally unsynchronized transmission and reception speed of the CAN Bus Shield circuit. The cause of these limitations is that the quality of the Arduino circuit is not high, and the sampling time is not reasonable (sampling time varies from 0.01s to 0.1s, after performing the test many times, we choose a sampling time of 0.02s). Therefore, to serve the following research, the choice of Arduino circuit supplier is significant, improving research efficiency.

With the obtained results, we continue to conduct further studies based on this study. Based on the observed dynamic data set, we monitor emissions quality journeys, fuel consumption, etc., to serve the technical diagnostics of clusters and systems when damages occur. This will help technicians and drivers to operate, repair, maintain, and use the vehicle more efficiently.

V. CONCLUSION

This study presented an experimental study for reading dynamic parameters in a car through the OBD II port. The hardware of this data reader is designed based on the CAN Bus Shield and Arduino circuit. At the same time, the software is compiled on the Arduino V2.0.3 IDE environment and displays the diagnostic interface via LabView software. After running the test on the 2016 Toyota Vios, we obtained dynamic data parameters to check and monitor the engine load calculation, throttle position, and coolant temperature. Through the test results, we conclude that the dynamic data reading circuit has been operating stably, and the parameters are obtained in real-time. However, specific difficulties were encountered during the research process, such as the burned Arduino circuit and the Can Bus Shield circuit's transmission and reception speeds needing to be synchronized.

REFERENCES

[1] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki, “Personalised assistance for fuel-efficient driving,” *Transp. Res. Part C Emerg. Technol.*, vol. 58, pp. 681–705, 2015, doi: 10.1016/j.trc.2015.02.007.

[2] Z. Szalay *et al.*, “ICT in road vehicles — Reliable vehicle sensor information from OBD versus CAN,” in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 469–476. doi: 10.1109/MTITS.2015.7223296.

[3] N. Kushiro, Y. Oniduka, and Y. Sakurai, “Initial Practice of Telematics-Based Prognostics for Commercial Vehicles: Analysis Tool for Building Faults Progress Model for Trucks on Telematics Data,” *Procedia Comput. Sci.*, vol. 112, pp. 2155–2164, 2017, doi: 10.1016/j.procs.2017.08.244.

[4] D. Sik, T. Balogh, P. Ekler, and L. Lengyel, “Comparing OBD and CAN Sampling on the go with the SensorHUB Framework,” *Procedia Eng.*, vol. 168, pp. 39–42, 2016, doi: 10.1016/j.proeng.2016.11.133.

[5] M. D’Agostino, M. Naddeo, and G. Rizzo, “Development and validation of a model to detect active gear via OBD data for a Through-The-Road Hybrid Electric Vehicle,” *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 6618–6623, 2014, doi: 10.3182/20140824-6-ZA-1003.01166.

[6] G. Charalampidis, A. Papadakis, and M. Samarakou, “Power estimation of RF energy harvesters,” *Energy Procedia*, vol. 157, pp. 892–900, 2019, doi: 10.1016/j.egypro.2018.11.255.

[7] D. Rimpas, A. Papadakis, and M. Samarakou, “OBD-II sensor diagnostics for monitoring vehicle operation and consumption,” *Energy Reports*, vol. 6, pp. 55–63, 2020, doi: 10.1016/j.egyr.2019.10.018.

[8] M. A. Hamed, M. H. Khafagy, and R. M. Badry, “Fuel Consumption Prediction Model using Machine Learning,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, pp. 406–414, 2021, doi: 10.14569/IJACSA.2021.0121146.

[9] Medashe Michael Oluwaseyi and Abolarin Matthew Sunday, “Specifications and Analysis of Digitized Diagnostics of Automobiles: A Case Study of on Board Diagnostic (OBD II),” *Int. J. Eng. Res.*, vol. V9, no. 01, pp. 91–105, 2020, doi: 10.17577/ijertv9is010045.