JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Research and Development of a Remote Vehicle Diagnostic Device via Mobile Application

Ngoc Le Nguyen[1], Minh Khoi Nguyen[2], Duy Thong Nguyen[3], Thanh Phat Le[3], Thanh Tuyen Nguyen[3*]

[1]*Thaco Auto Company Limited, Vietnam*
[2]*Ces Plasma Environment And Energy Technology Company Limited, Vietnam*
[3]*Ho Chi Minh City University of Technology and Education, Vietnam*

[*]Corresponding author. *Email: tuyennt@hcmute.edu.vn*

ARTICLE INFO

Received: 15/05/2025
Revised: 09/06/2025
Accepted: 20/06/2025
Published: 28/11/2025

KEYWORDS

CANBus;
Car Diagnostics;
Diagnostic Trouble Code DTC;
OBD-II;
IoT Automotive;
Firebase Realtime Database.

ABSTRACT

The rapid development of the automotive industry has significantly increased the number of vehicles, thereby driving a higher demand for vehicle diagnosis and maintenance. However, existing diagnostic devices are still limited by their connection range and high cost. This study aims to develop a novel diagnostic solution that offers more flexible and cost-effective connectivity. The research begins by analyzing and synthesizing technical documents on OBD-II systems, Android application development tools, and real-time databases. A diagnostic module, using an ESP32 microcontroller is then developed to read and process CAN signals from the vehicle's OBD-II system. The module is designed to be compact, cost-effective, and non-intrusive to the vehicle's hardware. Additionally, an Android application was developed with functionalities such as data reading, Diagnostic Trouble Code (DTC) reading, clearing all DTCs, and component activation testing. This application enables users to interact bidirectionally with the remote diagnostic module via a real-time database, without being restricted by distance. The application functions similarly to a multi-functional diagnostic tool, offering effectively unlimited connection range and fast response times (0.4–0.6 seconds). This research successfully delivers a flexible diagnostic solution for vehicles, addressing common limitations of existing diagnostic devices.

Doi: https://doi.org/10.54644/jte.2025.1906

1. Introduction

On-Board Diagnostics (OBD) has become an important tool in monitoring and reporting technical problems, supporting maintenance and repair processes. However, traditional diagnostic tools are often expensive and require direct connection, requiring the user to be present next to the vehicle to perform the test. This creates inconvenience when the vehicle experiences issues at a considerable distance from the repair facility, leading to increased costs and prolonged processing times.

In recent years, numerous international studies have explored the development of mobile applications and wireless connectivity for OBD-II systems, such as Bluetooth, Wi-Fi, and GSM technologies, to support remote diagnostics and provide more convenient, flexible solutions.

Muhammad Nadeem Iqbal et al. (2017) developed an on-board diagnostic kit and a remote online diagnostic system for European-standard vehicles. The system uses GSM technology to establish remote connections with the Electronic Control Unit (ECU), enabling a server to conduct diagnostics and provide repair instructions to the user. However, this research primarily focuses on domestic European vehicles, limiting its global applicability [1].

A study by Pirapuraj Ponnampalam et al. (2021) proposed a smart diagnostic system in which an Internet of Things (IoT) device connects to the vehicle's OBD-II port to transmit data to a server, providing vehicle status updates via a mobile application. The system also facilitates a network connection between customers and service stations, enabling quotation requests and remote repair

support. Despite its practicality, the system emphasizes routine maintenance over in-depth remote diagnostics [2].

Another study by Ahmed J. Abid et al. (2014) introduced a remote vehicle monitoring and support system that allows for remote monitoring, fault diagnosis, CO emission control, and traffic condition analysis. A notable feature of this study is its ability to predict potential faults based on sensor data and fault history while alerting drivers about safety concerns. However, the combination of this system onto mobile devices to increase flexibility has not yet been implemented [3].

In Vietnam, some studies have initiated the application of OBD combined with Bluetooth and Wi-Fi to enable remote vehicle status monitoring, making it easier for users and technicians to access vehicle data [4], [5]. However, these solutions primarily perform basic diagnostic functions and lack deep IoT integration, which restricts remote vehicle interaction.

To address these limitations, our research proposes a remote diagnostic device incorporating IoT technology.

2. OBD-II data frame

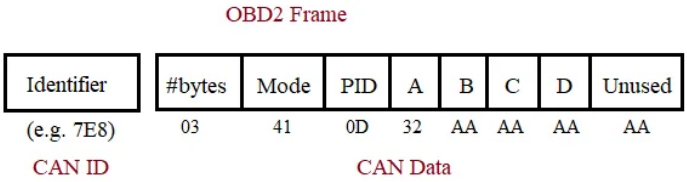


Figure 1. *OBD-II frame*

The standard 11-bit OBD-II data frame structure, used in the CAN protocol (ISO 15765) includes key fields, as shown in Figure 1: ID (0x7DF for requests, 0x7E8–0x7EF for ECU responses), length (03–06 bytes), mode (as shown in Table 1, defined by SAE J1979), PID (to specify the requested parameter), and data (up to 4 bytes: A, B, C, D). The data bytes are interpreted based on the PID and mode to extract actual sensor values, such as engine speed or oxygen sensor voltage. [6]

Table 1. *10 diagnostic modes on the OBD-II system.*

Mode (hex)	Describe
01	Show live data (real-time data)
02	Show freeze frame data
03	Show stored Diagnostic Trouble Codes (DTCs)
04	Clear DTCs
05	Show oxygen sensor monitoring test results
06	Show On-Board Monitoring Test Results
07	Displays pending fault codes during the current drive cycle or the most recent last drive cycle
08	Check the operation of components/systems
09	Request vehicle information
0A	Permanent DTCs (not supported by all vehicles)

3. Recommended mobile device and applications

3.1. Overview of the communication system between the module and the mobile application

We propose a system consisting of CAN signal from OBD-II (1), a TJA1050 CAN signal conversion circuit (2), ESP32 (3), Wi-Fi and mobile data (4)(5), mobile application (6), and a real-time database built on Firebase Realtime Database (7), as shown in Figure 2. Specifically, ESP32 processes requests from the application, sends and receives data over the CAN, then send it to Firebase. The mobile

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

application interacts with the user, transmits and displays data from Firebase. Data on Firebase will be continuously updated to ensure timely diagnosis and display via the Internet connection.

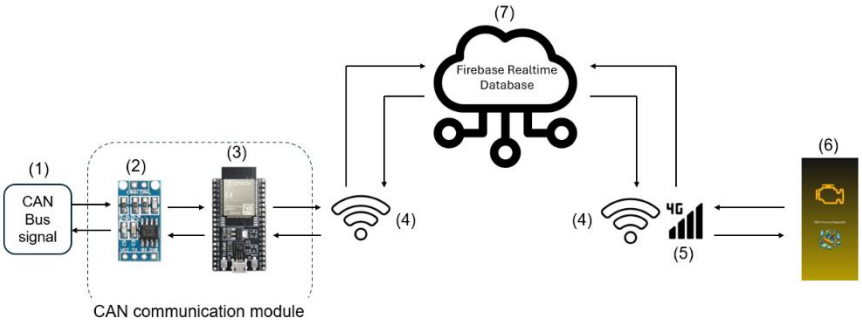


Figure 2. *System overview block diagram*

Figure 3 shows the circuit diagram of CAN communication module via OBD-II port. It uses the module ESP32 microcontroller to communicate with the CAN network via the TJA1050 CAN signal conversion circuit. The 12V power from the OBD-II port is regulated to 5V to supply power to the ESP32 and TJA1050. The TJA1050 connects to CAN High and CAN Low, converting the CAN signal into digital data for further processing by the module ESP32.

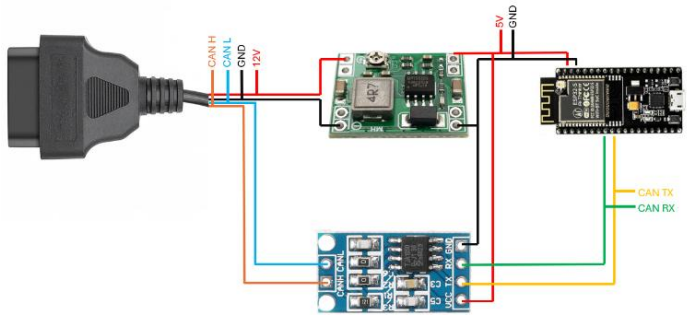


Figure 3. *Circuit diagram of CAN communication module via OBD-II port*

The module measures 90x45 mm and is encased in a 3D-printed plastic shell, featuring LED indicators for power and Wi-Fi connection status, as shown in Figure 4. When the Wi-Fi connection is successful, the corresponding LED will light up. The device is compact, easy to install, uses power from the OBD-II port, and does not interfere with the vehicle hardware.

Figure 4. *CAN communication module via OBD-II port*

3.2. Block diagram on ESP32

Figure 5 shows the block diagram of the main processing program on module ESP32. Before connecting to Firebase, ESP32 needs to elestasblish a new Wi-Fi connection. After starting, ESP32 automatically connects to a saved Wi-Fi network. If the connection is successful, it skips the initialization block (1) and starts initializing CAN and Firebase communication (2). Otherwise, if the Wi-Fi connection fails, ESP32 will initialize Bluetooth to configure a new Wi-Fi. Users can connect to ESP32 via a mobile app with Bluetooth, scan, and select an available Wi-Fi network. After receiving the correct SSID and password, ESP32 connects to the Wi-Fi network, stores this Wi-Fi in Flash memory

for use in subsequent connections, then disconnects Bluetooth, initializes CAN and Firebase to perform the required functions.

After connecting to Wi-Fi and Firebase, ESP32 read the "Function" value from Firebase to determine the function to perform. Depending on the request, ESP32 gets data from the OBD-II port and sends it to Firebase. For real-time functions, such as reading live data and drawing graphs, ESP32 continuously updates the data. For other functions, such as DTC reading and clearing or component activation, it only performs once and resets the "Function" variable after completion.

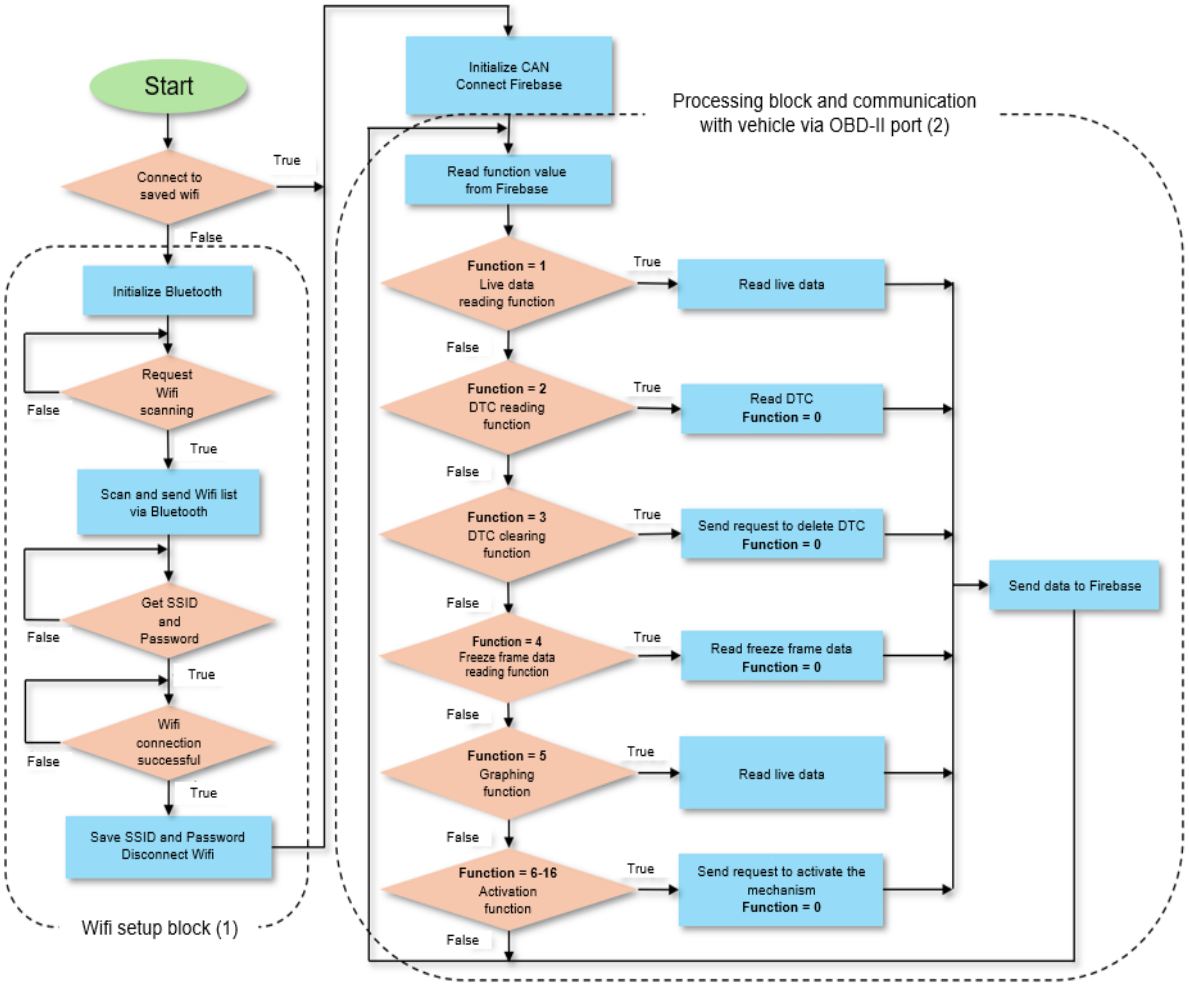


Figure 5. *Block diagram of the main processing program on ESP32*

3.3. Mobile applications

3.3.1. Live data reading function

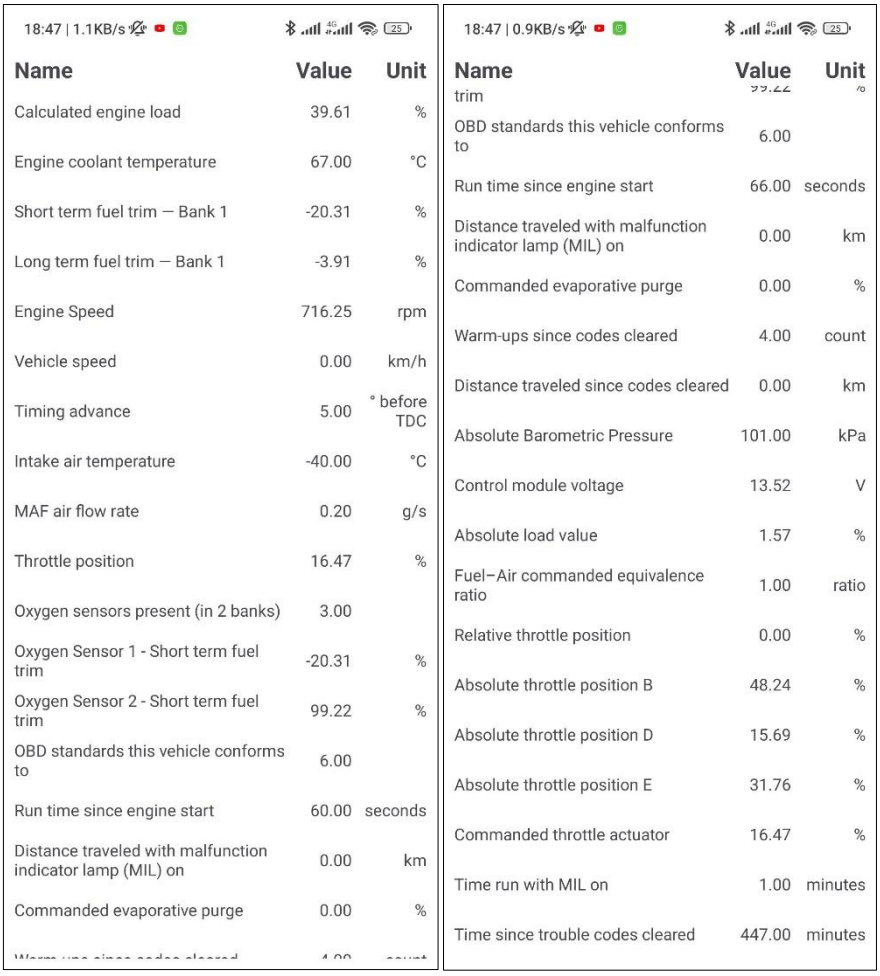
When the live data reading function is selected, the application initializes Firebase connection and sends a "Function = 1" to Firebase for the module to perform this task. After collecting the data, the values are converted to actual values using formula (1):

$$\textbf{\textit{physical\_value = offset + scale * raw\_value\_decimal}} \text{ [7]} \quad (1)$$

where ***raw_value_decimal*** is the feedback data from the ECU with a length of 8-32 bits, translated into decimal, and then calculated according to the ***scale*** and ***offset*** values corresponding to each specified Parameter ID (PID).

Figure 6 shows the information displayed, includes the name, value, and unit of the parameters. This data is automatically updated on Firebase, enabling continuous monitoring and detection of anomalies in engine operation.

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn



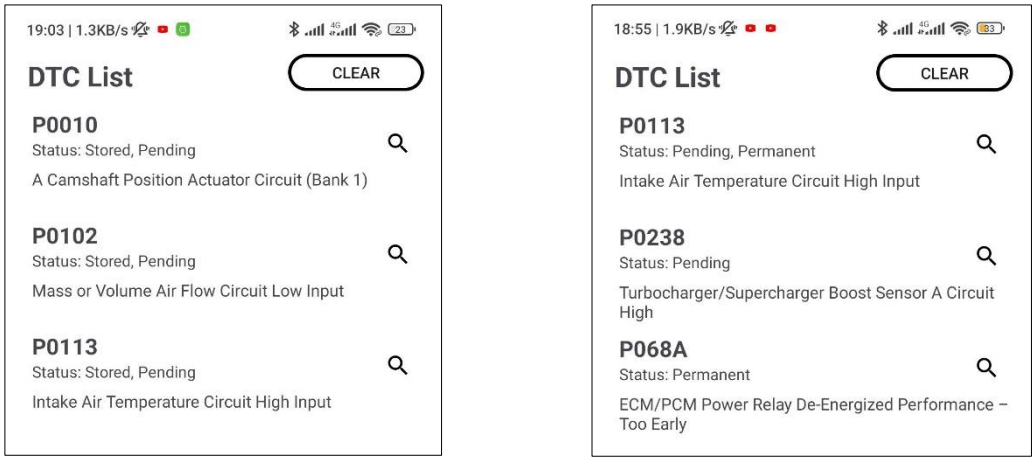
Name	Value	Unit
Calculated engine load	39.61	%
Engine coolant temperature	67.00	°C
Short term fuel trim — Bank 1	-20.31	%
Long term fuel trim — Bank 1	-3.91	%
Engine Speed	716.25	rpm
Vehicle speed	0.00	km/h
Timing advance	5.00	° before TDC
Intake air temperature	-40.00	°C
MAF air flow rate	0.20	g/s
Throttle position	16.47	%
Oxygen sensors present (in 2 banks)	3.00	
Oxygen Sensor 1 - Short term fuel trim	-20.31	%
Oxygen Sensor 2 - Short term fuel trim	99.22	%
OBD standards this vehicle conforms to	6.00	
Run time since engine start	60.00	seconds
Distance traveled with malfunction indicator lamp (MIL) on	0.00	km
Commanded evaporative purge	0.00	%

Name	Value	Unit
trim		
OBD standards this vehicle conforms to	6.00	
Run time since engine start	66.00	seconds
Distance traveled with malfunction indicator lamp (MIL) on	0.00	km
Commanded evaporative purge	0.00	%
Warm-ups since codes cleared	4.00	count
Distance traveled since codes cleared	0.00	km
Absolute Barometric Pressure	101.00	kPa
Control module voltage	13.52	V
Absolute load value	1.57	%
Fuel–Air commanded equivalence ratio	1.00	ratio
Relative throttle position	0.00	%
Absolute throttle position B	48.24	%
Absolute throttle position D	15.69	%
Absolute throttle position E	31.76	%
Commanded throttle actuator	16.47	%
Time run with MIL on	1.00	minutes
Time since trouble codes cleared	447.00	minutes

Figure 6. *Live data reading results on Toyota Vios 2007*

3.3.2. DTC reading and clearing function

a. DTC reading function



(a) *(b)*

Figure 7. *DTC reading results on Toyota Vios 2007 (a) and Ford Focus 2019 (b)*

When DTC reading function is selected, the application sends "Function = 2" to read the DTC. The DTC list is retrieved from Firebase and classified into three states: “Stored”, “Pending”, and “Permanent” as shown in Figure 7. The DTC status can vary between vehicle manufacturers.

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

The application also supports searching for DTC information on the Internet; simply tap the search icon to switch the browser with the syntax "DTC [Diagnostic Trouble Code]" as shown in Figure 8. This function is a great support in providing more detailed information about DTC as well as inspection and repair methods.

Figure 8. *Internet DTC search function*

b. DTC clearing function

As Figure (9-10) shows, when the DTC clearing function is used, the application will send "Function = 3" to Firebase allow to delete all DTC and stored data. The test results on a Ford Focus 2019 show that all DTCs have been deleted, except for "Permanent", which are only deleted automatically once the damage is repaired.

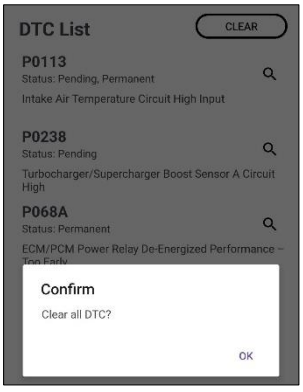


Figure 9. *DTC clearing function*

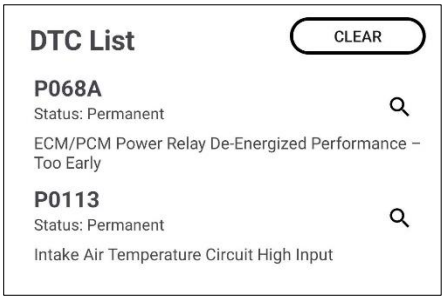


Figure 10. *Results after clearing DTC on Ford Focus 2019*

3.3.3. Freeze frame data reading function

When the freeze frame data reading function is selected, the application sends "Function = 4" to Firebase. The application displays supporting parameters including DTC, name, value, and unit (see Figure 11). The collected data will also be converted into actual values according to formula (1). The

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

number of parameters from freeze frame data may be less than the live data because not all are saved when an error occurs.

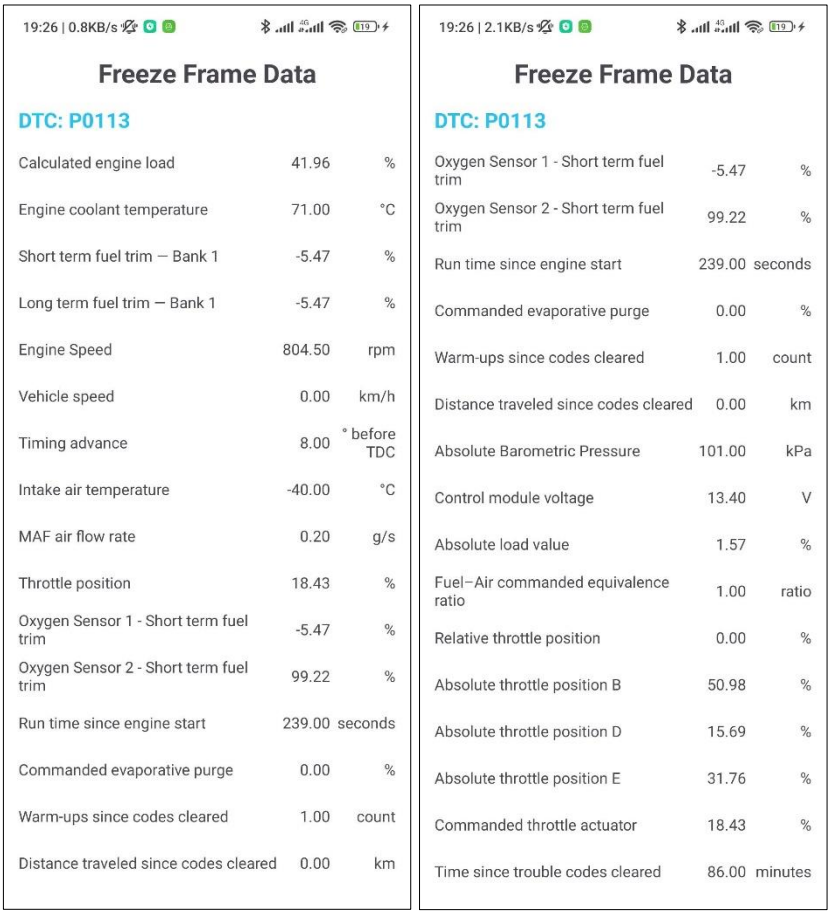


Figure 11. *Freeze frame data reading results on Toyota Vios 2007*

3.3.4. Graphing function

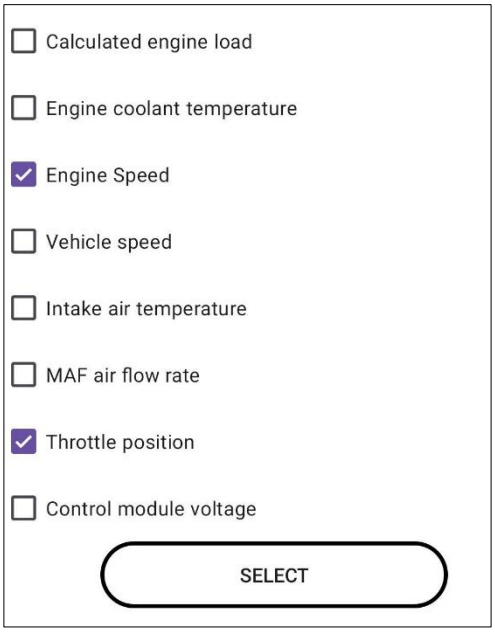


Figure 12. *Select display parameters*

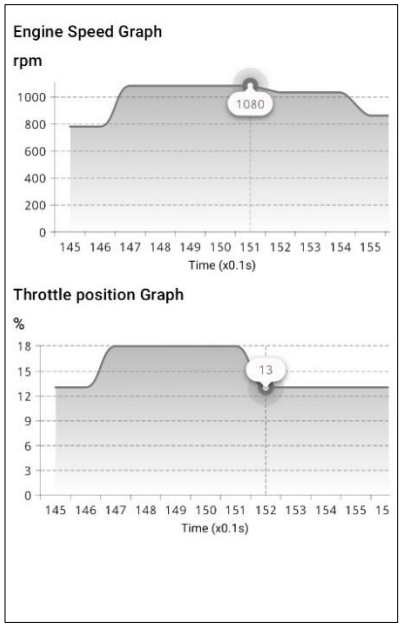


Figure 13. *Engine speed and throttle position graph on Toyota Vios 2007*

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

In the graphing function, the application initializes Firebase and sends a request "Function = 5" to the module to read live data for graphing. The application will automatically update data from Firebase for the selected PIDs as soon as there is a change, with an update time of no more than 10 ms [8]. Figure 12 displayed the interface for selecting display parameters. Data from ESP32 is sent every 400 ms. To improve performance, the graph update on the application occurs every 100 ms (see Figure 13).

3.3.5. Component activation test function

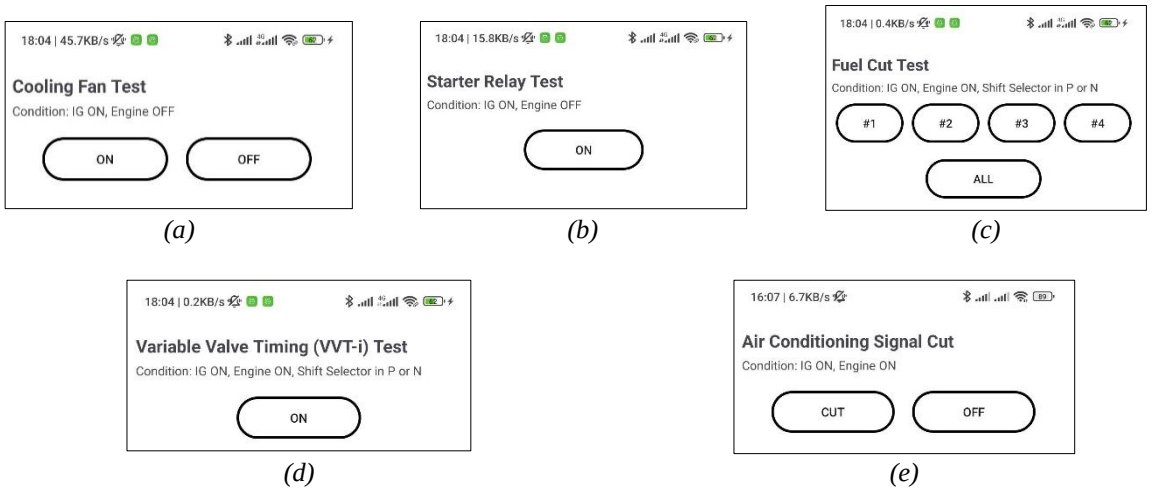


Figure 14. *Component activation test function: (a) Cooling fan test; (b) Starter relay; (c) Fuel cut test; (d) Variable Valve Timing test and (e) Air conditioning signal cut*

The component activation test function allows testing of five systems: cooling fan, starter relay, fuel system, variable valve timing, and air conditioning signal, as shown in Figure 14. Each system has one or two operating modes.

The application sends a Function value from 6 to 16 to request activation of the actuators.

+ Cooling fan: Function = 6 (on), Function = 7 (off).

+ Starter relay: Function = 8 (on).

+ Fuel cut: Function = 9 (cylinder 1 cut-off), Function = 10 (cylinder 2 cut-off), Function = 11 (cylinder 3 cut-off), Function = 12 (cylinder 4 cut-off), Function = 13 (all cylinders cut-off).

+ Variable valve timing: Function = 14 (on).

+ Air conditioning signal cut: Function = 15 (cut), Function = 16 (on again).

Each activation operation comes with specific conditions to ensure safety and does not affect other systems.

4. Results

4.1. Evaluation of test results on Toyota Vios 2007 and Ford Focus 2019

Figure 15. *Testing equipment on Toyota Vios 2007*

Figure 16. *Testing equipment on Ford Focus 2019*

After finishing the algorithm on ESP32 and developing the mobile application functions, we conducted tests on Toyota Vios 2007 and Ford Focus 2019 (see Figure (15 - 16)) by disconnecting the sensors to create simulation DTC. The results showed that the application could accurately detect DTC and display related information, confirming the device's performance in actual error conditions. The results are shown in the Table 2.

Table 2. *Test results on Toyota Vios 2007 and Ford Focus 2019*

Function	Toyota Vios 2007	Ford Focus 2019
Live data reading	Read and display 30 current parameters on the engine	Read and display 34 current parameters on the engine
Freeze frame data reading	Read and display 26 freeze frame data parameters	Read and display 32 freeze frame data parameters
DTC reading	Read and display 3 DTCs in “Pending” and “Stored” states	Read and display 3 DTCs in different states. “Pending” and “Permanent” state
DTC clearing	100% successful deletion of DTC and stored data	100% successful deletion of DTC and stored data
Component activation test	Successful activation on 5 component	None

The application developed in this paper is capable of reading and displaying all current parameters, DTC in all modes, and freeze frame data on both Toyota Vios 2007 and Ford Focus 2019. The DTCs are displayed with full information and fault status, allowing users to look them up on the Interne. This is a significant improvement over previous research, which could only read some current data and DTC in storage mode.

During the testing process, the DTC clearing success rate reached 100%, and the operations were performed entirely through the mobile application over the internet, without any limitations on connection range. The data obtained is highly reliable, compared with specialized equipment and the company's repair documentation during the algorithm development process.

In addition, the application also has the component activation test function with five components/systems on the Toyota Vios 2007, including the cooling fan, starter relay, fuel system, variable valve timing system, and air conditioning system signal. This function is a highlight, only available in high-priced diagnostic tool, which have not been integrated into current mobile diagnostic devices. The module is compactly designed, conveniently connected to the vehicle's OBD-II port and takes advantage of the 12V power source from this port.

4.2. Evaluate the interface and main functions of the application

Table 3. *Compare the functions on the application with the main functions on multi-functional diagnostic device.*

Number	Function	On other diagnostic tools	On the application
1	Live data reading	✓	✓
2	Display data in graph	✓	✓
3	DTC Reading	✓	✓
4	DTC Clearing	✓	✓
5	Freeze frame data reading	✓	✓
6	Component activation test	✓	✓ (On Toyota Vios)
7	Diagnose all vehicle component/systems	✓	X
8	Remote operation	✓ (Connect the device to the vehicle and control via third-party applications)	✓ (Develop directly on the application)

Table 3 compares the main functions of the diagnostic application proposed by us and the other diagnostic tools currently available on the market. The application has been designed to cover all the basic functions of a diagnostic tool, including:

+ Live data reading: Accurately displays engine parameters, including name, value and unit.

+ DTC reading and clearing: Provides DTC descriptions and status, along with Internet lookup functionality.

+ Freeze frame data reading: Record parameters at the time of fault detection.

+ Display data as graphs: Allows displaying up to three parameters simultaneously, enhancing visualization.

+ Component activation test: This function allows testing of component without input signals or dismantling parts, providing convenience to the user.

Although the current functions are only applicable to the engine control system, the research product allows for remote diagnostic operations via the internet, unlike current other diagnostic tools, which require the use of additional third-party software.

4.3. Evaluation of data update time on ESP32

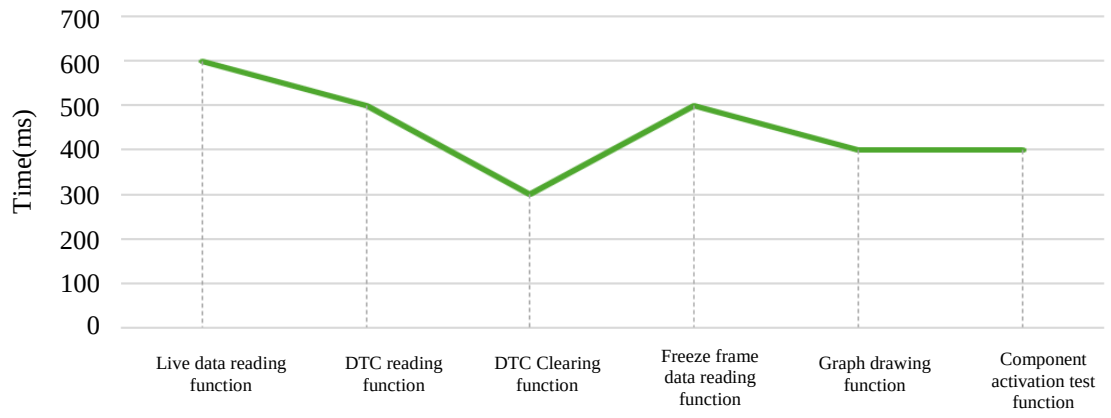


Figure 17. *Average data update time on ESP32*

Figure 17 displayed data transmission time between ESP32 and Firebase. It is affected by many factors, including processing time on ESP32, packet size, speed and stability of internet connection. In the test conditions, Internet speed is maintained stable, so the two main factors affecting data transmission time are mainly packet size and processing time on ESP32.

The packet size varies depending on the operating mode. The live data reading and freeze frame data functions generate packets of up to 1 MB, while the DTC reading and graphing functions have faster update times due to their smaller sizes. For the DTC clearing and component activation test functions, the processing time is shortest because only a request message is sent.

The live data reading and graphing functions require continuous real-time data updates. Under stable internet conditions, the processing time is 0.6 seconds and 0.4 seconds, respectively, which is not too high but still not really optimal, and the data displayed on the application is not continuous.

4.4. Evaluate data update time on the application

To evaluate the data update time on the application, we debugged the application on Android Studio software. Under stable internet conditions, the recorded data update time fluctuated from 1 to 2 ms. This delay is considered very small and completely acceptable, meeting the requirements for data transmission speed and interaction between the application's functions and module.

5. Conclusions

Remote diagnosis on automotive is an essential need in the context of the strong development of the automotive industry. The solution developed in this study not only brings convenience to users and

technicians, but also supports organizations and authorities in managing and monitoring vehicles remotely, serving many different purposes. This study has successfully developed an application that operates stably on the Android operating system, integrating the basic functions of the diagnostic device. The application allows communication with the module via the internet, ensuring low latency and meeting the requirements of the diagnostic system. Test results on Toyota Vios 2007 and Ford Focus 2019 show that the system achieves high accuracy in reading operating parameters and DTCs of the engine control system, while also accessing advanced functions of the multi-function diagnostic device. In addition, this paper proposed a new solution: combining the OBD-II self-diagnosis system with the IoT platform, solving the problem of accessing vehicle data and remote diagnosis. Beside the achieved results, the current system still has some limitations, such as the diagnostic function does not cover all systems on the vehicle, the component activation test function is limited in the number of components, and the application cannot perform diagnosis without an internet connection. Due to limited experimental conditions, the application scope is not yet widely applied to other vehicle models. From the achieved results, this paper has opened up many potential development directions, including expanding the diagnostic database, integrating early damage prediction functions, and adding the ability to update vehicle status and traffic conditions. These improvements promise to improve the efficiency and flexibility of the system, while better meeting the practical needs of users and the automotive industry in the future.

Acknowledgments

This work was supported by the facilities and research environment provided by Ho Chi Minh City University of Technology and Education, Vietnam.

Conflict of Interest

The authors declare no conflict of interest.

REFERENCES

[1] M. N. Iqbal *et al*., "Diagnostic tool and remote online diagnostic system for Euro standard vehicles," *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 2017, pp. 415-419, doi: 10.1109/ITOEC.2017.8122328.
[2] U. Maalik and P. Ponnampalam, "Intelligent vehicle diagnostic system for service center using OBD-II and IoT," *International Conference of Science and Technology*, South Eastern University, Sri Lanka, 2021, pp. 209-214.
[3] A. J. Abid and R. S. A. A. Waily, "Vehicle remote support and surveillance system," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 10, pp. 55–63, 2014, doi: 10.33762/eeej.2014.95513
[4] "ELM 327 Bluetooth multifunction diagnostic device," (in Vietnamese), [Online]. Available: https://obdvietnam.vn/thiet-bi-chan-doan-da-nang-elm-327-bluetooth-3001.html. [Accessed: Mar. 1, 2024].
[5] "FOBD error code reader device," (in Vietnamese), [Online]. Available: https://obdvietnam.vn/thiet-bi-doc-loi-fobd-3064.html. [Accessed: Mar. 1, 2024].
[6] "OBD2 frame format or message structure," [Online]. Available: https://www.rfwireless-world.com/Terminology/OBD2-Frame-format.html. [Accessed: Mar. 13, 2024].
[7] "OBD2 PID overview [lookup/converter tool, table, CSV, DBC] HEX," [Online]. Available: https://www.csselectronics.com/pages/obd2-pid-table-on-board-diagnostics-j1979. [Accessed: Mar. 14, 2024].
[8] "Choose a database: Cloud Firestore or Realtime Database," [Online]. Available: https://firebase.google.com/docs/database/rtdb-vs-firestore. [Accessed: Mar. 25, 2024].

Ngoc Le Nguyen received his B.E. degree in Automotive Engineering from Ho Chi Minh City University of Technology and Education (HCMUTE) in 2024. His studies focused on the development of technologies and applications in the automotive field. He is working at Thaco Auto Company Limited.
Email: lenguyen45911@gmail.com. ORCID: https://orcid.org/0009-0002-8509-1417

Minh Khoi Nguyen received his B.E. degree in Automotive Engineering from Ho Chi Minh City University of Technology and Education (HCMUTE) in 2025, Vietnam. His studies focus on the development of technologies and applications in the automotive field. He is working at Ces Plasma Environment And Energy Technology Company Limited.
Email: minhkhoixh057@gmail.com. ORCID: https://orcid.org/0009-0001-1615-8545

Duy Thong Nguyen is currently an second-year student majoring in Automotive Engineering Technology at Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His studies focus on the development of technologies and applications in the automotive field.
Email: 23145206@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0006-7679-0872

Thanh Phat Le is currently an second-year student majoring in Automotive Engineering Technology at Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam. His studies focus on the development of technologies and applications in the automotive field.
Email: 23145165@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0006-6537-1111

Thanh Tuyen Nguyen was born in Binh Dinh, VietNam in 1989. He received the B.S. degree in Automotive Engineering from HoChiMinh City University of Technology and Education, VietNam, in 2012 and the M.S. degree in Automotive Engineering from University of Technology and Education, VietNam, in 2015. From 2011 to 2013, he also worked at Hocdelam Group, VietNam. His fields of study are Automative Control, Automotive Electrics, Air-conditioning Engineering, Battery Management in Electric Vehicles, EV, and HEV Technologies.
Email: tuyennt@hcmute.edu.vn. ORCID: https://orcid.org/0009-0003-7135-2804