PROGRESS IN ENGINEERING APPLICATION AND TECHNOLOGY
e-ISSN: 2773-5303

PEAT

Vol. 7 No. 1 (2026) 323-327
https://publisher.uthm.edu.my/periodicals/index.php/peat

Mobile Application for Vehicle Maintenance and Scheduling System

Nur Qamarina Balqis Ballia[1], Aimi Syamimi Ab Ghaffar[1]*

[1] Department of Electrical Engineering Technology, Faculty of Engineering Technology. Universiti Tun Hussein Onn Malaysia,Muar,84600,Pagoh,Johor,MALAYSIA

*Corresponding Author: aimi@uthm.edu.my
DOI: https://doi.org/10.30880/peat.2026.07.01.037

Article Info

Received: 12 January 2026
Accepted: 20 February 2026
Available online: 25 May 2026

Keywords

Vehicle scheduling system, Mit App, Car system

Abstract

Vehicle maintenance is essential to ensure reliable performance, reduce the risk of breakdowns and extend the lifespan of the vehicle. Traditional maintenance practices often rely on manual tracking and delayed diagnostic, which can lead to missed service intervals and unexpected issues. This project proposes the enhancement of an integrated system that combines real-time battery and mileage monitoring with intelligent maintenance scheduling through a mobile application. The system utilizes an ESP32 microcontroller and an OBD-II adapter to continuously collect vehicle battery and mileage data. The data is transmitted to a user-friendly Android application and developed using MIT App Inventor via Bluetooth. The app tracks odometer readings and calculates a maintenance schedule based on the type of engine oil used and driving patterns. In addition, the application provides automatic service alerts, the system is designed to be low cost, efficient and scalable offering vehicle owners a smart accessible tool for proactive vehicle health management. Performance evaluation confirms the accuracy of the sensor data and the reliability of service reminders, making this solution both practical and effective for everyday use.

1. Introduction

In recent years, the demand for smarter and more efficient vehicle maintenance solutions has increased alongside advancements in automotive electronics and mobile technologies. Traditional approaches, such as manual logbooks and basic dashboard warning indicators, often provide limited or delayed information regarding vehicle condition, which can result in missed maintenance and unexpected breakdowns [1],[2] .These limitations highlight the need for more accessible and real-time vehicle monitoring systems.

With the growing availability of wireless communication, low-cost microcontrollers such as the ESP32, and mobile application development platforms, vehicle maintenance systems can be significantly improved. Previous studies have demonstrated that integrating OBD-II data with microcontroller-based systems enables effective monitoring of vehicle parameters, including battery status and mileage information [2], [3]. By presenting this information through a mobile application, users are able to monitor essential vehicle conditions more conveniently. Therefore, this project builds upon these developments by combining ESP32-based hardware, OBD-II data acquisition, and mobile connectivity to provide a practical system for real-time battery monitoring and mileage tracking, helping vehicle owners maintain their vehicles more effectively and reduce the risk of unexpected issues.

1.1 Problem Statement

This project focuses on solving common vehicle maintenance problems caused by the lack of real-time battery monitoring and unreliable manual tracking methods. Traditional approaches provide limited information and often alert users only at critical stages, which can lead to unexpected breakdowns, delayed servicing, and increased repair costs. To address this, the project proposes a smart system using an ESP32 and OBD-II interface to continuously monitor battery health and vehicle mileage. The system is integrated with a user-friendly mobile application that provides real-time updates, mileage tracking, oil change suggestions, and automated maintenance reminders based on driving habits and oil type. By evaluating the accuracy of the battery.

1.2 Objectives

The objectives include:

- To enhance a sensor node to continuously monitor vehicle battery status using ESP32, and OBD-II.
- To enhance a user-friendly mobile application allowing vehicle owners to monitor vehicle battery status, track real-time mileage, suggest calculating next service dates based on oil type and receive timely.
- To evaluate the system performance in terms of battery monitoring accuracy and maintenance schedule generation.

2. Materials and Methods

This project utilizes both hardware and software materials to develop a smart vehicle monitoring and maintenance system. An ESP32 microcontroller is used as the core processing unit due to its low cost, high processing capability, and built-in wireless connectivity, enabling continuous data acquisition and real-time communication. An OBD-II interface module is employed to access vehicle parameters such as battery voltage and mileage directly from the vehicles onboard diagnostics system, ensuring accurate and reliable data collection. To support battery health monitoring, voltage sensing components are integrated with the ESP32 to measure battery status continuously and detect potential issues before critical failure occurs. On the software side, a mobile application developed using MIT App Inventor serves as the user interface, allowing vehicle owners to view real-time battery status, track mileage, receive oil change suggestions, and obtain automated maintenance reminders. Bluetooth or Wi-Fi communication modules are used to transmit data wirelessly from the ESP32 to the mobile application, ensuring timely updates and ease of access. Together, these materials form a comprehensive system that addresses the limitations of traditional maintenance methods by providing accurate, real-time monitoring and intelligent maintenance scheduling.

2.1 System Block Diagram

In Fig 1, the system operates by collecting vehicle data through the OBD-II interface, which is connected directly to the vehicle diagnostic port. A power supply provides the required voltage to the ESP32 (Sender), ensuring stable operation. The ESP32 (Sender) receives data from the OBD-II module, processes the information, and transmits it wirelessly to the ESP32 (Receiver). The ESP32 (Receiver) then forwards the received data to the mobile application via Bluetooth communication. Finally, the MIT App displays the vehicle and battery information in a user-friendly format, allowing users to monitor vehicle status and maintenance-related data in real time.

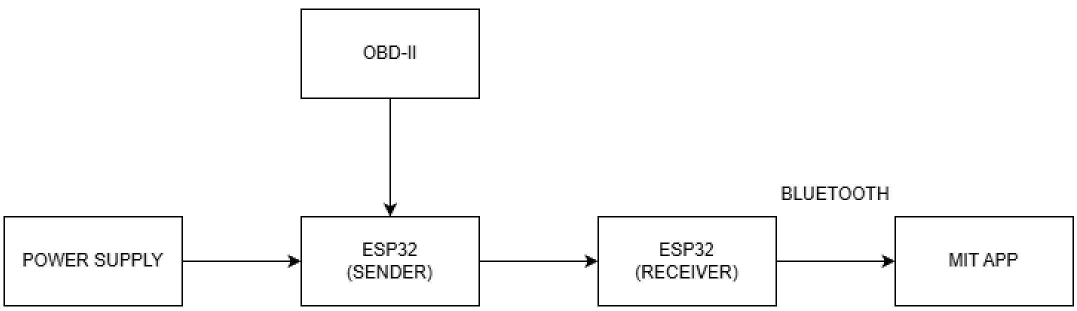
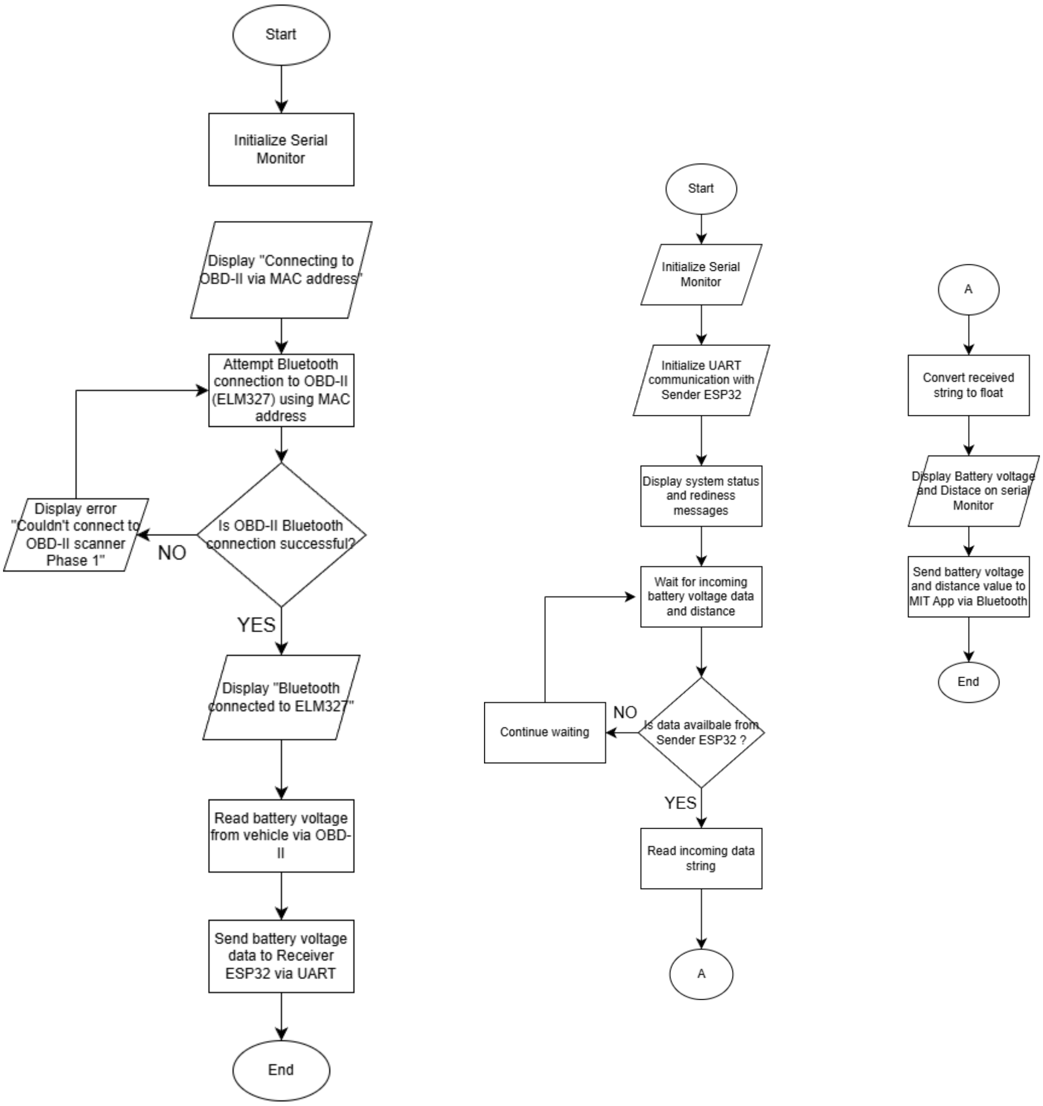


Fig.1 Block Diagram of the system

2.2 Hardware Development

Fig.2 shows that the sender ESP32 flowchart summarizes the process of connecting to the vehicle's OBD-II system via Bluetooth, initializing the ELM327 module, and reading the battery voltage. If any connection or initialization step fails, the system stops to prevent errors. Once successful, the sender reads the battery voltage data and transmits it to the receiver ESP32. Fig.3 shows that the receiver ESP32 flowchart summarizes the process of initializing serial and Bluetooth communication, then continuously waiting for data from the sender. When battery voltage data is received, it is converted into a numerical value and transmitted to the mobile application via Bluetooth, allowing real-time monitoring of the vehicle battery status.



(1)

(a) Sender (b) Receiver

Fig.2 System Flowchart

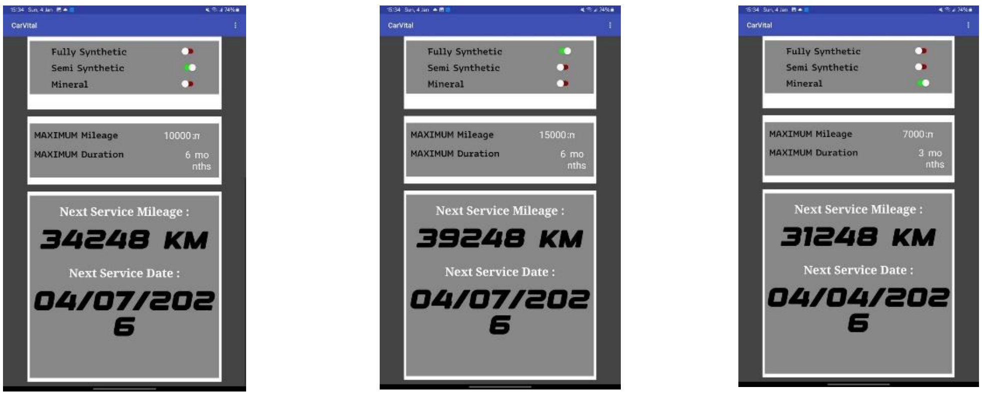
3. Results, Analysis and Discussion

In Fig.3 shows, how the MIT App Inventor application interface to incorporates a value battery monitoring system. When the vehicle's battery health reduces or drops below acceptable levels, the MIT app immediately alerts users using the alert warning, as shown in Fig.3(a). The ESP32 microcontroller may receive and continually display real-time battery voltage data on the smartphone screen when the MIT app successfully develops wireless connection with it via Bluetooth connection, as shown in Fig.3(b). Also, in Fig.3(c) it illustrates the distance value measuring capabilities, showing how real-time mileage data is concurrently recorded and tracked via the serial monitor interface for purposes of checking.

(a) **(b)** **(c)**

Fig.3 Bluetooth connected Battery Alert (a), Battery Monitoring (b) and Distance value (c)

In Fig.4 shows each engine oil type screen when it is chosen. As shown, each oil type has its own maximum mileage and duration displayed when the mileage value has been calculated. In Fig.4 shows that the application is able to automatically schedule the maintenance date and mileage for each type by calculating usage patterns.



(a) **(b)** **(c)**

Fig.4 Maintenance Scheduling (a) Fully synthetic (b) Semi-synthetic (c) Mineral

4. Conclusion

This project successfully developed a smartphone application integrated with an ESP32-based hardware system to monitor vehicle battery voltage and track engine oil mileage using OBD-II data. The results show that the system is capable of obtaining real-time battery voltage readings from the vehicle and transmitting the data wirelessly to a mobile application with reliable performance. This enables users to continuously monitor the condition of the vehicle battery through a smartphone interface. In addition, the system supports oil mileage tracking by using distance-related data obtained from the vehicle, allowing users to estimate when engine oil servicing is required based on mileage usage. By presenting this information in a clear and user-friendly MIT App Inventor application, the system helps vehicle owners better manage routine maintenance and reduce the risk of battery-related issues. Overall, the developed system provides a practical and convenient solution for vehicle condition monitoring and basic maintenance awareness.

Acknowledgement

The author would like to thank the Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia for their support.

References

[1] *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems*. IEEE, 2017.
[2] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156739.
[3] K. Iskandar, A. Tambayong, M. R. F. Mulya, S. C. Elfanlie, and M. G. Herlina, "Mobile-Based Car Diagnostic Application Using Onboard Diagnostic-II Scanner," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 14, no. 2, pp. 129–141, Dec. 2023, doi: 10.21512/comtech.v14i2.9138.
[4] M. Ismail and R. Ahmed, "A Comprehensive Review of Cloud-Based Lithium-Ion Battery Management Systems for Electric Vehicle Applications," 2024, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2024.3446880.
[5] V. S. Malokar Mukesh R Dhale Nikita V Gawande Dipti S Nimkande Ajay P Kandarkar Sangharatna Gawande, "Battery Monitoring System using ESP8266& Arduino IOT Cloud Prof.Atul Ghute," *International Journal of Research and Analytical Reviews*, 2022, [Online]. Available: www.ijrar.org
[6] Sandhya M, "Design and Implementation of Vehicle Tracking System Using Esp32," 2024.
[7] W. Zhe Min and R. Muslim, "A Study on Common Problem on Smart Parking System at AEON Seremban 2," *Progress in Engineering Application and Technology*, vol. 3, no. 1, pp. 268–278, 2021, doi: 10.30880/peat.2022.03.01.028.
[8] P. Delgado, "Vehicle Monitoring Using OBD2 and Alexa (SmartCar). Final Year Project Report for the Degree of Bachelor in Electronics Engineering", doi: 10.13140/RG.2.2.23578.26562.
[9] D. Rimpas, A. Papadakis, and M. Samarakou, "OBD-II sensor diagnostics for monitoring vehicle operation and consumption," *Energy Reports*, vol. 6, pp. 55–63, Feb. 2020, doi: 10.1016/j.egyr.2019.10.018.