OPEN ACCESS
International Journal of Innovations in Science & Technology

Design And Implementation Of Black Box For Automobiles Using Esp 32

Maaz Khan, Arbab Masood Ahmad, Waseem Sajjad, Usman Khan Khalil
Department of Computer Systems Engineering University of Engineering & Technology Peshawar, Pakistan
***Correspondence**: muhammadmaazyousafzai@gmail.com, arbabmasood@uetpeshawar.edu.pk, waseemsajjadismaila1@gmail.com, usmankhankhalil09@gmail.com

Citation | Khan. M, Ahmad. A. M, Sajjad. W, Khalil. U. K, "Design And Implementation Of Black Box For Automobiles Using Esp 32", IJIST, Vol. 07 Issue. 02 pp 1083-1097, June 2025

Received | April 05, 2025 **Revised |** May 29, 2025 **Accepted |** May 31, 2025 **Published |** June 02, 2025.

A black box system in vehicles acts as an important tool that records important information to make vehicles safer, investigate accidents, and even improve the overall performance of the vehicles. This study will present the Black Box System that has been developed using ESP32 microcontrollers for cars to enhance data collection and analysis in automotive fields using technology. The Black Box System or Event Data Recorder (EDR) is an important tool in the enhancement of road safety, investigation of accidents, and evaluation of the performance of a vehicle. The system utilizes ESP32 as the main microcontroller since it is cost-effective, efficient, and can be programmed in multiple ways. It comprises several sensors and data acquisition modules to collect key parameters including speed, acceleration, geographical location, engine, and various diagnostic information about the vehicle. This paper also presents a detailed overview and integration of the system into Hardware and software parts of the automobile. A user-friendly interface facilitates data retrieval and analysis, supporting applications in fleet management, driver behavior monitoring, and accident investigations. The study focuses on the responsibility and protection of personal data, as well as ways of protecting personal data from misuse and violation of the law. Therefore, using ESP32 technology in the vehicle's Black Box System is a great improvement towards road safety and vehicle monitoring. By ensuring data security and privacy, this system provides the users with a complete data set to support a decision-making process for self-employed drivers and other organizations.

Keywords: Black Box System, ESP32, Sensors, Microcontroller, Automobile, Event Data Recorder, Vehicle Safety, IoT Integration

Introduction:

Black boxes are devices originally developed for the aviation industry to enhance flight safety by recording data on aircraft operations and pilot performance. These are known as Flight Data Recorders (FDRs) and Cockpit Voice Recorders (CVRs) and became indispensable in the investigation of accidents and enhancement of operations' safety. Considering the success of the black box technology in the aviation industry, its applicability in automobiles was considered to reduce the rate of accidents and analyze them.

The growth in the technology of vehicles led to the increased use of Event Data Recorders (EDRs), the other name for the black box. These are gadgets that are meant to record various parameters of the car including speed, acceleration, braking pattern, and geographic position of the car which can be crucial in analyzing the events of an accident, monitoring the driver's behavior, and improving the efficiency of the automobile. According to the World Health Organization (WHO) globally, every year over 1.3 million people die, and millions of others are injured in road traffic accidents making it the biggest source of casualties, especially to the group of people of ages between fifteen and twenty-nine years [1]. The data indicates a growing demand for technological features like black box systems to enhance road safety. This work presents an automobile Black Box System designed using ESP32 microcontrollers. In contrast to using platforms such as Arduino, the ESP32 provides the advantage of higher computing power, inbuilt Wi-Fi, as well as Bluetooth capabilities, all in a cost-effective solution for real-time data acquisition. The system also requires several sensors that allow measuring vital values including speed, acceleration, GPS coordinates, engine status, and driver's actions and behavior.

The following research questions have been developed for the paper:

- In what ways do the black box systems enhance the safety of the roads and assist in case of an accident?
- What technological, legal, and privacy challenges hinder the widespread adoption of black box systems in automobiles?
- How can ESP32-based systems offer a more efficient, scalable, and secure solution compared to traditional microcontroller platforms?

In addressing these questions, this paper aims to demonstrate how modern black box technology can revolutionize traffic safety by enhancing driver accountability, reducing traffic fatalities, optimizing vehicle system efficiency, ensuring compliance with legal requirements, and upholding individuals' privacy rights.

Literature Review:

The automotive black boxes are named so because their origins are linked to the aircraft industry, using such systems to record data and analyze accidents. Thus, the application of this technology to automobiles is still in its development stage due to the developments in microcontroller technology, integration of sensors, and data communication.

Previous Studies and Existing Technology:

The research conducted provided detailed information on developing and integrating a car's Black Box System with Arduino technology. These include accelerometers, GPS modules, microphones, and cameras that help in acquiring essential data like speed, acceleration, performance of the engine, and geographical coordinates. It is important to monitor data in real-time; interfaces must be friendly and data protection should be possible using SD cards and external drives for storing data. It is particularly useful in fleet management, accident reconstruction, and insurance claims to give insight into the vehicle and its usage by the driver. [1]

In the same study, another system was designed, namely the Automotive Smart Black-Box Monitoring System, which focuses on real-time detection of accidents and sending immediate alerts. This system uses a microcontroller to record various vehicle parameters, storing the data on a digital memory card and also backing it up to cloud storage. It incorporates gas detectors, vibration sensors, temperature sensors, Zigbee, GSM, and GPS modules to trigger alerts to emergency services upon detecting abnormal conditions. According to author(2023), this proposed system outperformed traditional techniques such as RFID-based monitoring, Support Vector Machines (SVM), and even deep learning models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in enhancing road safety and facilitating accident investigations [2].

Another study developed an IoT-based Vehicle Black Box System equipped with cameras and various sensors to monitor both vehicle performance and driver behavior. The system can send alerts to emergency services and relevant stakeholders via SMS and email, ensuring a quick response in the event of a mishap. Notably, the system emphasizes IoT connectivity, which enhances data availability and enables real-time telemetry. According to author(2024), this integration of sensors and IoT technologies improves both accident response and vehicle diagnostics by maintaining continuous communication with cloud-based services [3].

One of the subsequent research projects by authors(2020) proposed an Automobile Black Box System that employs both Raspberry Pi and Arduino controllers to manage multiple sensors for accident investigation. The system incorporates data encryption mechanisms to prevent data tampering;an essential consideration in black box technology. By using dual controllers, the design enhances data accuracy and ensures secure storage, thereby enabling more reliable and tamper-proof accident analyses [4].

Identified Gaps in Current Research:

Thus, although there has been a considerable improvement in the development of automotive black box systems there exist some limitations:

- **Standardization and Legal Compliance:** However, unlike aviation black boxes, there is no standardized set of procedures or legal requirements for implementing black box systems in automobiles. Challenges that can be attributed to privacy include the issues of constant surveillance and sharing of information.
- **Data Security and Privacy:** While data privacy is addressed in both studies, implementing enhanced encryption and secure transmission protocols is essential to safeguard against breaches of data protection regulations.
- **Integration with Advanced Driver-Assistance Systems (ADAS):** Current research is largely focused on the post-accident phase, especially during the developmental stage. There exists an opportunity to incorporate the black box into ADAS to give immediate feedback to the driver as well as alert him/her to any potential mishap that may happen in the future.
- **Scalability and Cost**: Arduino-based systems are inexpensive and very flexible, though they have limitations in terms of scalability for high-volume automotive applications. There is a need to conduct further studies that help define other innovative models that can have lower costs and still offer the essential features.
- **Real-Time Data Utilization:** While Arduino-based black box prototypes and commercial vehicle monitoring systems have real-time monitoring capabilities, there is very little consideration for how such data collected during the drive can be used in real-time to modify behavior or take correction actions during the drive[5].

In conclusion, the integration of black box systems in automobiles has some potential to increase safety on roads and proper investigation of accidents, but nevertheless, there are some concerns regarding privacy issues, legal issues, and technical issues that should be examined adequately to attain better outcomes.

Objectives:

Our main objectives are as follows:

Crash Response/investigation:

- Automatic detection of impact and enhancing the ability to analyze accidents.
- Emergency Response using GPS/GSM modules Vehicle management.
- Logging vehicle data i.e., diagnosing vehicles through error codes, speed,and fuel logging.
- Keeping a record of locations the vehicle traveled to.

Anti – Theft Measure:

- Real-Time Car Tracking
- Emergency alert SOS

Data Storage and Communication:

- Data storage at the time of impact

Novelty Statement:

This study introduces a new black box system for cars that prioritizes improved safety monitoring, local data storage, and affordability [6]. The suggested architecture makes use of the ESP32 microcontroller, which provides better processing power and integrated connectivity at a lower cost than traditional systems based on Arduino or Raspberry Pi. The integration of gas (MQ-7) and alcohol (MQ-3) sensors is a significant breakthrough that makes it possible to detect dangerous in-cabin circumstances, which is not often possible in low-cost implementations. Furthermore, the system offers a dependable solution that functions without requiring real-time cloud connectivity by prioritizing data privacy through limited data access and secure SD card storage.

Methodology:

The methodology encompasses both hardware and software components, with a particular focus on data acquisition, accident detection, and data security. The identified methodology includes hardware and software elements, in particular, the focus is on data acquisition, accident identification, and data security.

System Architecture:

The architecture of the Black Box System integrates multiple sensors connected to an ESP32 microcontroller for real-time data capture. Key components include:

GPS Module for real-time location tracking.The ESP32 microcontroller was selected for this system due to its combination of high performance and low cost, making it ideal for embedded automotive applications. It features a dual-core processor that supports real-time data processing from multiple sensors simultaneously. Additionally, its integrated Wi-Fi and Bluetooth capabilities simplify wireless data transmission and remote monitoring without requiring additional modules[7]. The ESP32 also offers a flexible development environment with wide community support, extensive libraries, and compatibility with popular platforms such as Arduino IDE and MicroPython, facilitating faster prototyping and system integration.

Gas Sensor (MQ-7) for detecting harmful gas concentrations in the vehicle.

Accelerometers (MPU-6050) for monitoring speed and sudden decelerations.

SD Card Module for data storage.

LCD Display for real-time feedback to the driver.

Buzzer for auditory alerts in case of emergencies.

The hardware connections of the system are illustrated in Figure 1 below:

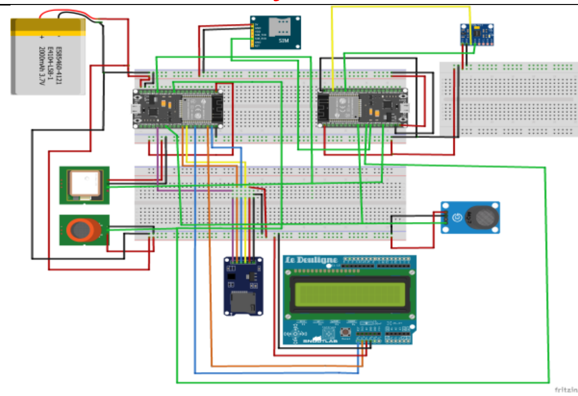


Figure 1. Hardware Schematic of the ESP32-based Black Box System

Data Flow and Accident Detection Process:

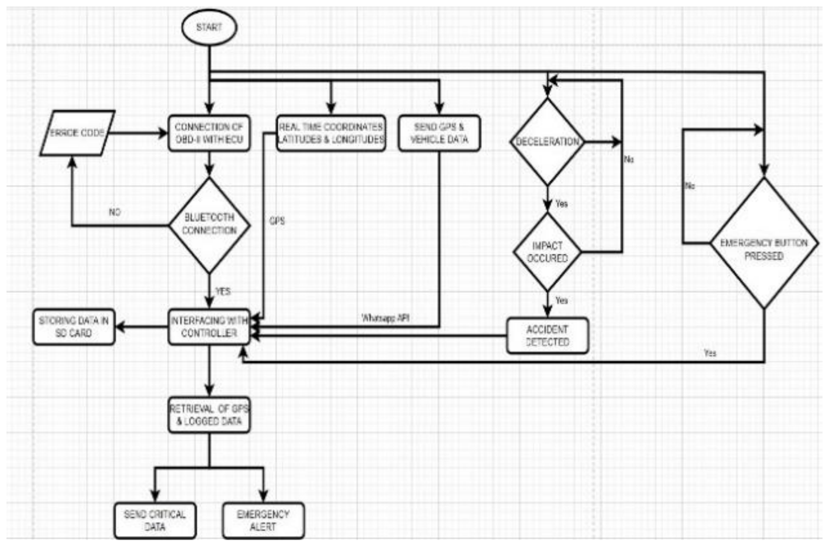


Figure 2. Flowchart of Data Collection, Accident Detection, and Emergency Response

The system employs a combination of GPS and GSM modules for data transmission and emergency notifications. The flowchart in Figure 2 illustrates data acquisition, accident detection, and alerting process. Key steps include:

1. **Data Acquisition:** Real-time coordinates and vehicle diagnostics are collected via OBD-II and GPS modules.
2. **Accident Detection:** The MPU-6050 accelerometer, which detects acceleration along multiple axes, is used by the system to continuously check for both abrupt hits and unexpected decelerations. Impact detection is differentiated by abrupt, high-frequency acceleration spikes that take place over a very short period of time, whereas deceleration is recognized when a notable decrease in forward acceleration beyond a predetermined threshold. The accelerometer is set up to distinguish between usual braking patterns and the abrupt shocks that occur during crashes. Impact detection is accomplished by carefully analyzing accelerometer data; no more sensors are needed [8].
3. **Data Storage & Transmission:** Collected data is stored in an SD card and transmitted to emergency contacts via GSM.

Process Flow Diagram:

Our system is divided into 2 major parts:

PART 1: Accessing Vehicle Parameter over Bluetooth:

The process flow of the first part starts by checking the Bluetooth connection of the ESP-32 with the OBD-II. The process stays at this block until the connection is successfully established. After this, the ESP-32 starts to access data from the external sensors as well as the

GPS module. All the acquired data is stored in the SD-card module and displayed on an LCD as well [9].

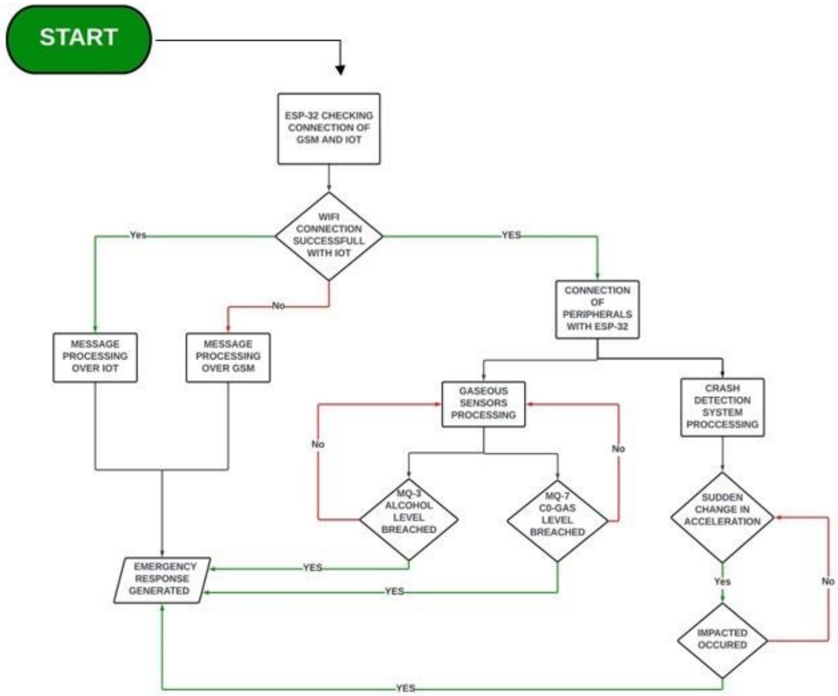


Figure 3. Process Flow Diagram Part 1

PART 2: Generating Emergency Response over Wi-Fi/GSM:

The process flow of the first part starts by checking the connection of the ESP-32 with the Wi-Fi. If a Wi-Fi connection is established, then all the emergency alert generation process is done over the WhatsApp API. If the Wi-Fi connection isn't available, then the system shifts to a GSM module for generating emergency alerts over SMS. The Crash detection block shows how MPU-6050 is utilized to detect the impact of the vehicle[10].

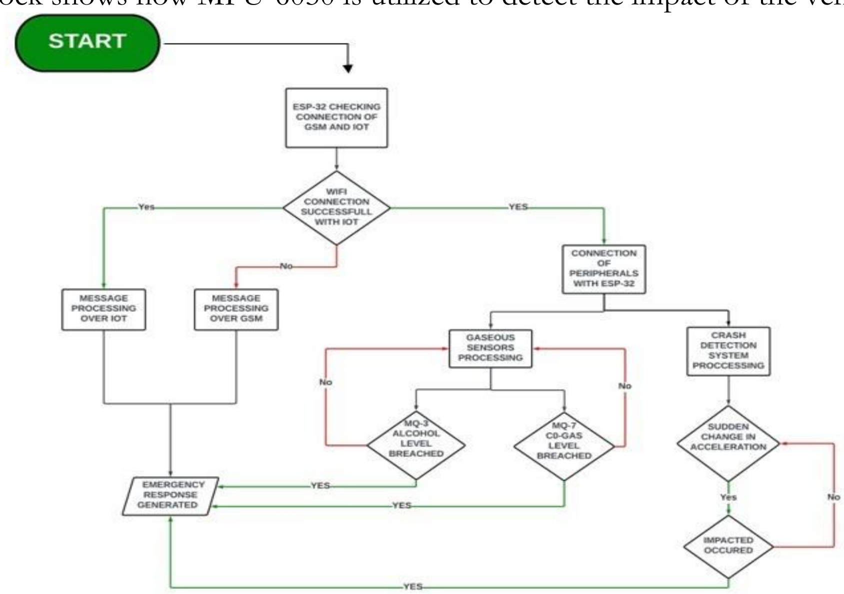


Figure 4. Process Flow Diagram Part2

Working Principle:

The Black Box which we have designed for automobiles is based upon three basic principles that govern its working and purpose of use. These principles are as follows:

- Data Collection from the ECU of the vehicle using an OBD-II connector that was interfaced with our microcontroller.
- Installing External sensors in the vehicle and interfacing them with the microcontroller.
- Utilizing the GPS module and WhatsApp API or GSM modules that are interfaced with the Microcontroller for generating Emergency Response for the vehicle by using its Tracking and Data Transmission abilities.

The Black Box starts its process execution from the Engine of the vehicle. ECU behaves as the brain of a vehicle. Now to access the data of the ECU, we linked the OBD-II connecter to the DLC port provided in the vehicle. To do processing on the data accessed from the ECU. We need to transfer the ECU's data to our microcontroller. For that, we used the built-in Bluetooth module of our microcontroller and transferred our ECU's data from the OBDII connector to our microcontroller using Bluetooth protocol.

We also needed a few external sensors for creating a completely functional black box therefore we included sensors like MPU-6050, MQ3 & MQ7, etc in our system as well. All the sensors were interfaced with the microcontroller and their data along with the data of the ECU's sensors is stored inside a micro-SD card mounted on the SD card module which is also interfaced with the microcontroller.

The Black Box would process all these data variables and then generate an Emergency Response, which includes using WhatsApp API and GPS module interfaced with the microcontroller to access the positional coordinates of the vehicle and then utilizing the API to send an SMS alert if WIFI connection is present else using GSM module to send that data along with the emergency alert via SMS to the designated numbers.

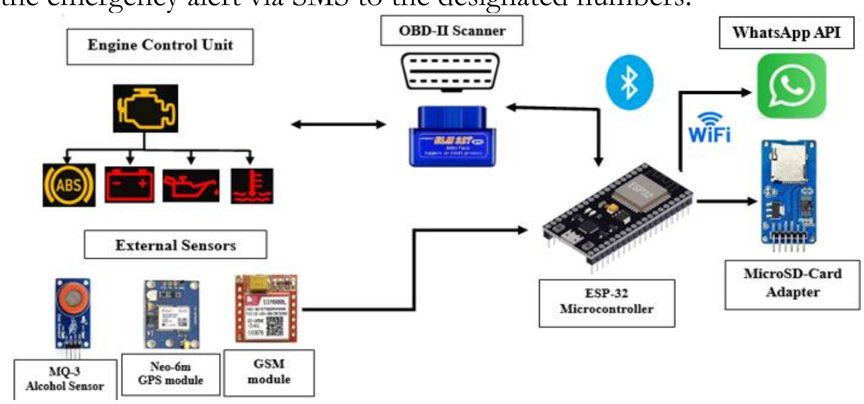


Figure 5. Methodology figure

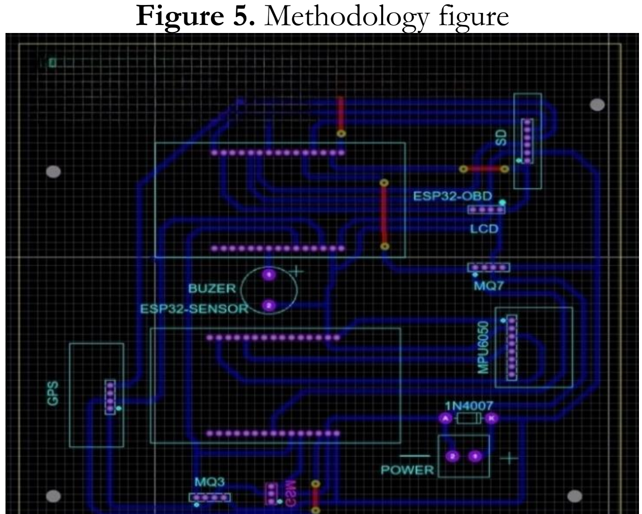


Figure 6. PCB layout PCB fabrication involved the following steps:

Proteus Schematic:

We used Proteus for drawing the schematic layout of our project. Some components that we used were not present in the proteus software libraries, so we used the dimensions to create the desired components.

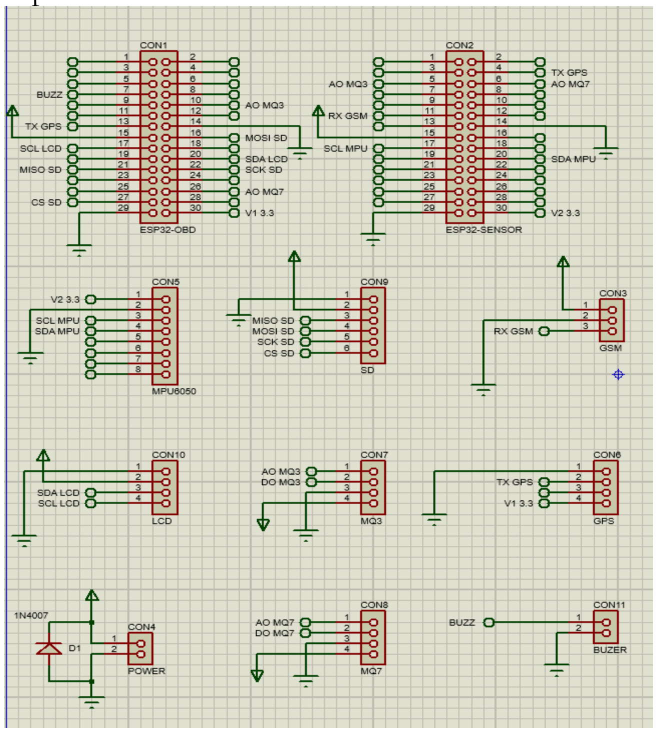


Figure 7. Proteus Schematic

PCB Layout:

PCB design refers to the process of creating a printed circuit board (PCB) layout that serves as the foundation for electronic circuits. It involves converting a schematic diagram into a physical design that incorporates all the necessary components, traces, and connections required for the circuit to function properly.

- Creating schematic diagrams on software (we used Fritzing for this purpose).
- Creating a PIN-to-PIN diagram on Proteus.
- Designing custom components in Proteus for our PCB.
- Placement of all the components in the specified board space and connecting them by routing.
- Print the PCB layout
- Drilling the PCB layout on the Copper sheet and using $FeCl_2$
- Soldering and connecting all components

For our system, we designed a PCB that was a single-layer PCB having 4 jumper wires.

Results & discussion:

System Performance and Data Accuracy:

The ESP32-based Black Box System was successfully developed and tested under various driving conditions. The system demonstrated high reliability in capturing real-time data and accurately detecting accident scenarios. The assembled hardware prototype is shown in Figure 8 below.

The prototype includes the following components:

ESP32 microcontroller for processing data.

- GPS module for real-time location tracking.
- Gas sensors to detect harmful gases inside the vehicle.

- SD card module for storing logged data. Once the ESP32 accessed all the parameters from the black box, the next step was to begin logging that data. To log data onto a storage device we utilized a Micro SD card module. The Micro SD-card module was interfaced with the ESP-32 microcontroller following our need for accident detection and vehicle parameter monitoring. The Micro SD-card module operates on two types of protocols for communication: 1) SD mode and 2) SPI mode (Serial Peripheral Interface)

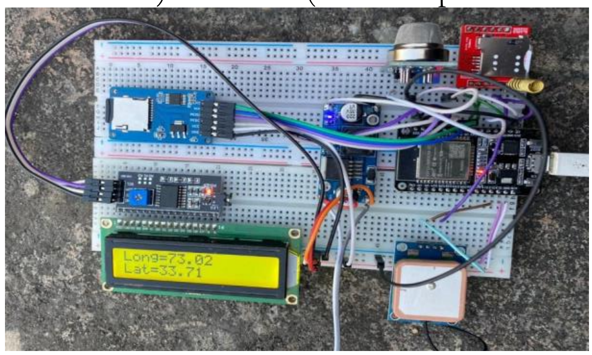


Figure 8. Prototype of the ESP32-based Black Box System

- **LCD:** The I2C LCD component is used in applications that require a visual or textual display.

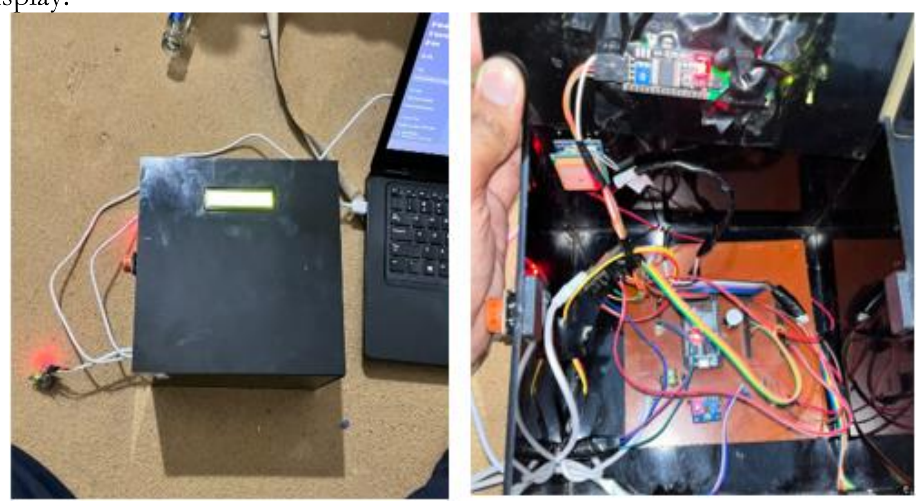


Figure 9. Hardware Results (Complete Prototype)

Impact on Accident Analysis and Insurance Claims:

The deployment of black box systems in automobiles has significantly improved the accuracy and speed of accident investigations:

Accident Reconstruction: Data from the black box provides clear insights into the vehicle's behavior before, during, and after a collision. Information such as speed, braking force, and steering input allowed for precise reconstruction of accident scenarios, which proves invaluable in determining the causes of accidents and assigning liability.

Insurance Claims Processing: Black box data has become a critical factor in **streamlining insurance claims**. Insurance companies can access objective, time-stamped data to verify claims, reducing the risk of fraud. This has resulted in faster claims processing and, in some cases, lower premiums for drivers who demonstrate safe driving habits.

Legal and Ethical Implications:

While black box systems offer clear benefits in terms of road safety and accident analysis, they also raise significant legal and ethical concerns:

Data Privacy: The continuous collection of sensitive data, including location and driving habits, raises questions about data ownership and privacy rights. Thus, the ownership of the data must be spelled out, whether it belongs to the driver, the manufacturer, or the insurer, and under what circumstances the data can be transferred or sold.

Privacy: This is an issue of ethical concern when drivers do not know how much data is being collected or what the data will be used for. This means that there should be clear policies and the consent to be given by the drivers should be well understood and unambiguous.

Legal Aspect: The system was made to meet certain legal requirements including the General Data Protection Regulation (GDPR) in the European Union. Proper measures of data protection, such as encryption and storage, as well as limiting access to this data were put in place.

Legal concerns: In most cases, black box data can be useful to determine who was at fault in an accident, however, the meaning of the data can be open to debate. Issues such as the validity of the data or the relevance of certain driving maneuvers may be raised, thus, the need to develop common guidelines for the use of data in legal cases.

Comparison with Existing Systems:

When comparing this system with the traditional black box system (that is, a system that does not use any control algorithms and is based on Arduino or OBD-II data logger), the following advantages of the ESP32-based system can be noted:

- **Enhanced Processing Power:** It is also important to note that ESP32 has a dual-core processor, which enables fast data processing and reaction to occurrences of an accident.
- **Wireless Interfaces:** The inclusion of Wi-Fi and Bluetooth connections ensures that there is no need for additional communication interfaces for data transfer to the cloud storage.
- **Cost-Effectiveness and Scalability:** It was found that the proposed system is cost-effective compared to other systems, with better flexibility for vehicular applications ranging from personal use to commercial use of vehicles.

Future Implications and Technological Advancements:

These black box systems could be improved by implementing IoT as well as machine learning algorithms into them. For instance:

- **Predictive Analytics:** The data on past driving could be used to predict when an accident might occur due to; behavior or conditions that have been associated with accidents.
- **Integration with Autonomous Vehicles:** In the context of the development of the automotive industry towards autonomous cars, black box systems will become indispensable tools to assess the performance of the algorithms for self-driving and to explain the reasons for failures.

Overall, the development of the Black Box System which is based on ESP32, is one of the most important innovations in automotive safety systems. Aside from its application in the analysis of accidents, insurance claims, and monitoring of driver behavior, the system also raises concerns regarding the proper and legal ways of dealing with vehicular data.

System Testing and Validation:

The system was thoroughly tested under various real-world and simulated conditions to validate its functionality, data accuracy, and emergency response capabilities. Testing focused on OBD-II parameter retrieval, sensor responsiveness, GPS/GSM communication, and system integration for event detection. The ESP32 platform's flexibility and performance enabled seamless interaction with all modules and ensured reliable data acquisition and transmission.

This system’s novelty lies not only in its affordability and integrated design but also in its ability to detect hazardous in-cabin conditions. The addition of gas (MQ-7) and alcohol (MQ-3) sensors enables proactive safety interventions, rarely found in other low-cost implementations. Furthermore, the emphasis on local data storage via secure SD card logging provides a privacy-focused alternative to always-online systems, making it suitable even in connectivity-challenged areas.

OBD-II Parameters:

Interfacing OBD-II with ESP32:

The system establishes a connection between the OBD-II module and ESP32 using the ATZ command, which resets the module. The ESP32 receives raw hexadecimal values which are parsed (typically from byte positions 1 to 3) to extract and validate meaningful diagnostic data.

```
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

Connected to ELM327
RAW:ATZELM327 v1.5>    SLICED:TZ
Welcome
RAW:0105SEARCHING...    SLICED:ING    FINAL:-40C
```

Figure 10. Bluetooth connected

```
Output   Serial Monitor ×
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

Couldn't connect to OBD scanner - Phase 1
```

Figure 11. When OBDII is not connected to ESP32

Engine RPM:

The command for RPM uses PID 0C. Raw values are extracted from byte positions 9–11 and 11–13. These are then converted from hexadecimal to decimal and processed using the standard RPM formula:

$$\text{RPM} = \frac{(A * 256) + B}{4}$$

Where A and B are the parsed byte values. Units are appended to finalize the result.

```
Output   Serial Monitor ×
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

04:25:22.165 -> RAW:010C410C0AE9>    SLICED:698    FINAL:698 rpm
04:25:22.165 -> Appending to file: /RPM.txt
04:25:22.199 -> Failed to open file for appending
04:25:24.167 -> RAW:010C410C0FD2>    SLICED:1012    FINAL:1012 rpm
04:25:24.167 -> Appending to file: /RPM.txt
04:25:24.214 -> Failed to open file for appending
04:25:26.194 -> RAW:010C410C1430>    SLICED:1292    FINAL:1292 rpm
04:25:26.194 -> Appending to file: /RPM.txt
04:25:26.194 -> Failed to open file for appending
04:25:28.188 -> RAW:010C410C1339>    SLICED:1230    FINAL:1230 rpm
04:25:28.188 -> Appending to file: /RPM.txt
04:25:28.188 -> Failed to open file for appending
04:25:30.191 -> RAW:010C410C1792>    SLICED:1508    FINAL:1508 rpm
04:25:30.191 -> Appending to file: /RPM.txt
04:25:30.191 -> Failed to open file for appending
04:25:32.196 -> RAW:010C410C154A>    SLICED:1362    FINAL:1362 rpm
04:25:32.196 -> Appending to file: /RPM.txt
```

Figure 12. Engine RPM

Coolant Temperature:

Using PID 05, the coolant temperature is derived by parsing bytes 9–11, converting from hex to decimal, and applying the formula:

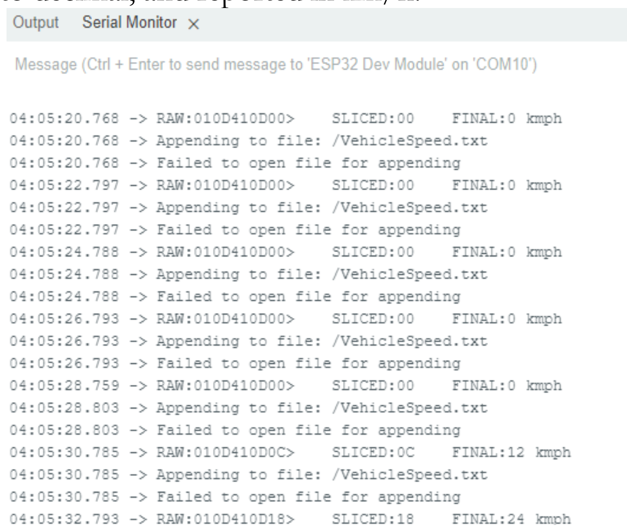
$$\text{Coolant } (°\text{C}) = A - 40$$

```
Output   Serial Monitor ×
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 70 >    SLICED:70    FINAL:72C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
RAW:010541 05 71 >    SLICED:71    FINAL:73C
```

Figure 13. Coolant

Vehicle Speed:

PID 0D provides vehicle speed. The relevant bytes are extracted (typically 1–11), converted from hex to decimal, and reported in km/h.



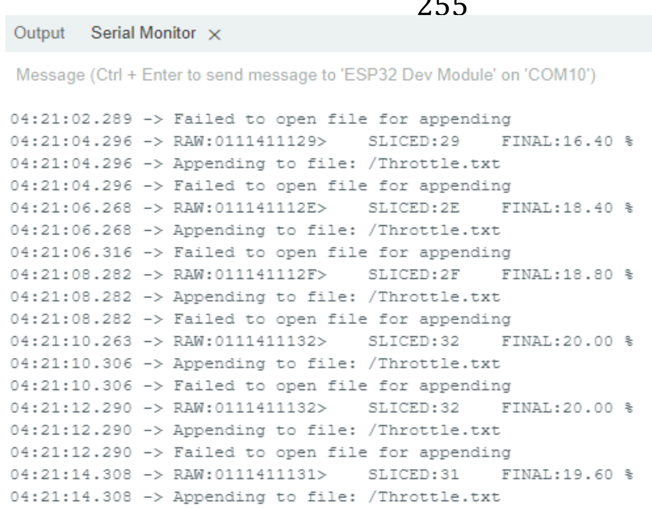
```
Output    Serial Monitor ×
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

04:05:20.768 -> RAW:010D410D00>    SLICED:00    FINAL:0 kmph
04:05:20.768 -> Appending to file: /VehicleSpeed.txt
04:05:20.768 -> Failed to open file for appending
04:05:22.797 -> RAW:010D410D00>    SLICED:00    FINAL:0 kmph
04:05:22.797 -> Appending to file: /VehicleSpeed.txt
04:05:22.797 -> Failed to open file for appending
04:05:24.788 -> RAW:010D410D00>    SLICED:00    FINAL:0 kmph
04:05:24.788 -> Appending to file: /VehicleSpeed.txt
04:05:24.788 -> Failed to open file for appending
04:05:26.793 -> RAW:010D410D00>    SLICED:00    FINAL:0 kmph
04:05:26.793 -> Appending to file: /VehicleSpeed.txt
04:05:26.793 -> Failed to open file for appending
04:05:28.759 -> RAW:010D410D00>    SLICED:00    FINAL:0 kmph
04:05:28.803 -> Appending to file: /VehicleSpeed.txt
04:05:28.803 -> Failed to open file for appending
04:05:30.785 -> RAW:010D410D0C>    SLICED:0C    FINAL:12 kmph
04:05:30.785 -> Appending to file: /VehicleSpeed.txt
04:05:30.785 -> Failed to open file for appending
04:05:32.793 -> RAW:010D410D18>    SLICED:18    FINAL:24 kmph
```

Figure 14. Vehicle Speed in Km/hr

Throttle Position:

PID 11 is used to extract throttle position data. Parsed values are converted from hexadecimal to percentage using the formula:

$$\text{Throttle } (\%) = \frac{(A * 100)}{255}$$



```
Output    Serial Monitor ×
Message (Ctrl + Enter to send message to 'ESP32 Dev Module' on 'COM10')

04:21:02.289 -> Failed to open file for appending
04:21:04.296 -> RAW:0111411129>    SLICED:29    FINAL:16.40 %
04:21:04.296 -> Appending to file: /Throttle.txt
04:21:04.296 -> Failed to open file for appending
04:21:06.268 -> RAW:011141112E>    SLICED:2E    FINAL:18.40 %
04:21:06.268 -> Appending to file: /Throttle.txt
04:21:06.316 -> Failed to open file for appending
04:21:08.282 -> RAW:011141112F>    SLICED:2F    FINAL:18.80 %
04:21:08.282 -> Appending to file: /Throttle.txt
04:21:08.282 -> Failed to open file for appending
04:21:10.263 -> RAW:0111411132>    SLICED:32    FINAL:20.00 %
04:21:10.306 -> Appending to file: /Throttle.txt
04:21:10.306 -> Failed to open file for appending
04:21:12.290 -> RAW:0111411132>    SLICED:32    FINAL:20.00 %
04:21:12.290 -> Appending to file: /Throttle.txt
04:21:12.290 -> Failed to open file for appending
04:21:14.308 -> RAW:0111411131>    SLICED:31    FINAL:19.60 %
04:21:14.308 -> Appending to file: /Throttle.txt
```

Figure 15. Throttle

External Parameters:

GPS and GSM:

- **GPS**: Connected to ESP32 pin 13 (TX), the GPS provides real-time longitude and latitude.
- **GSM (SIM800L)**: Connected via TX2 and RX2 with the SoftwareSerial library to enable communication on Serial2. GSM is configured to text mode using AT+CMGF=1. Coordinates received from the GPS are sent via SMS to a predefined emergency contact.

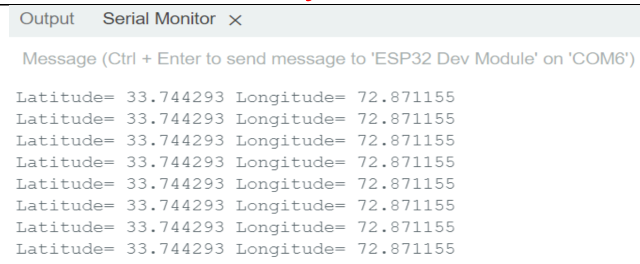


Figure 16. GPS Output

Gas Sensors and Emergency Response:

Sensors were calibrated and tested with threshold triggers. If any gas concentration exceeds safety limits, the following occurs:

- A buzzer is activated for 5 seconds.
- The GPS location is sent to the emergency contact via GSM.
- A welcome message confirms GSM module activation on vehicle start.

1. **Smoke Detection (MQ-7):**

Smoke presence triggers emergency SMS alerts.

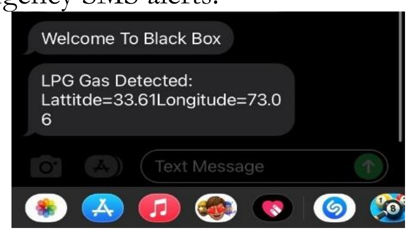


Figure 17. Smoke Detected Using GSM

2. **Carbon Monoxide Detection (MQ-7):**

CO levels are monitored and transmitted via a Wi-Fi-based alert system.

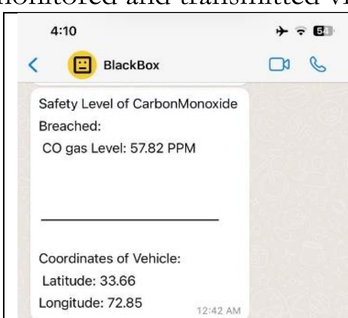


Figure 18. Carbon Monoxide Detected Using Wi-Fi

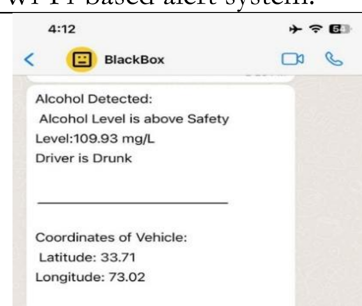


Figure 19. Alcohol Detected

3. **Alcohol Detection (MQ-3):**

If alcohol concentration exceeds 100 mg/L, an alert is sent via GSM including the driver's condition and real-time location.

Conclusion:

In this research, an ESP32-based Black Box System for automobiles was proposed for the purpose of increasing road safety, facilitating better analysis of accidents, and improving the performance monitoring of vehicles. The system was also highly accurate in recording important parameters like speed, acceleration, GPS location, and the status of the vehicle's engine as well as being able to detect accidents and send out alerts.

Key findings include:

- **Accurate Data Capture and Analysis:** It also ensured the documentation and transmission of important information previous, during, and even after accidents occurred which helped in reconstructing the accident scene and was useful in insurance claims.

- **Enhanced Road Safety and Driver Monitoring:** Through constant monitoring of the drivers and the performance of the car, the system is useful in ensuring that the drivers are more accountable and kept off the road, hence reducing the rate of accidents.
- **Legal and Ethical Considerations:** On the positive side, the need for the system cannot be dismissed, but the studies revealed a couple of concerns that are related to data protection, ownership, and legal concerns.

In this case, the ESP32 microcontroller was found to be economical, efficient, fast, connected, and scalable compared to the other systems. The use of SD cards and cloud storage was beneficial in the case of accidents as the data could be accessed easily and had a backup in case of damage.

Future Work:

Despite such a solid environment for vehicular data recording and analysis, there are still several directions that need to be researched to improve the system:

1. **Integration with AI for Predictive Analytics:**

It may also be useful in future studies to continue this research by analyzing driving data and machine learning to predict possible accident causes due to risky driver behavior or unfavorable conditions. Predictive analytics could help send signals to the drivers about the possible dangers of an incident and thus eliminate it.

2. **Advanced IoT Integration:**

Extension of the System's IoT integration is the possibility of effective Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V) communications. This can help in increasing traffic management, better implementation of anti-collision gadgets, and evolution of intelligent transportation systems.

3. **Autonomous Vehicle Monitoring:**

Future studies may examine how the black boxes could track and analyze the functioning of algorithms in autonomous vehicles that the automotive industry is currently developing. This would ensure that there is the ability to answer for some of the decisions made by the system in case of an accident as well as be able to detect some of the failures in the system.

4. **Enhanced Data Privacy and Legal Frameworks:**

It is recommended that future research should pay attention to the creation of clear legal norms and ethical standards for using vehicular data. This involves understanding how data can be made anonymous, how the encryption standards can be enhanced, and how the company can meet data protection standards of the world such as the GDPR.

5. **Integration with Health Monitoring Systems:**

The next possible features of the black box could be biometric indicators of the state of the driver: heart rate, fatigue, and alcohol levels. This data could also be used to identify instances of impaired driving and possibly prevent such incidences or respond to them as appropriate.

6. **Wireless Data Transmission and 5G Integration:**

The enhancement of the 5G network for better and faster data transmission could improve the system's capability to transmit and send a large amount of data in real time, especially in high-speed or remote driving environments.

Thus, based on the current ESP32-based Black Box System, it can be stated that there is a good foundation for monitoring vehicles and analyzing accidents; further development of AI, IoT, and autonomous technologies will create vast opportunities to enhance road safety and vehicles' performance. They will not only improve specific drivers' safety but will also promote the general welfare of society by minimizing the cases of traffic accidents and encouraging the use of efficient and safe roads.

Acknowledgments:

We are profoundly grateful to all those who helped us in this research project, especially the faculty and staff of the University of Engineering and Technology, Peshawar, Pakistan.

The manuscript has not been published or submitted to other journals previously.

Author's Contribution:

It is acknowledged that all authors have contributed significantly and that all authors agree with the content of the manuscript.

Conflict of interest: There exists no conflict of interest for publishing this manuscript in IJIST.

References:

[1] A. T. Rajendran Thanikachalam, Rajendran Thavasimuthu, Godwin John J, Maria Arockia Dass J, Nithya T, "Design and Implementation of a Car's Black Box System using Arduino," *Int. Res. J. Multidiscip. Technovation*, vol. 6, no. 3, 2024, doi: 10.54392/irjmt24320.

[2] K. M. P. Josephinshermila, S. Sharon priya, S. G. P. Ramakrishna hegde, and B. Veerasamy, "Accident detection using Automotive Smart Black-Box based Monitoring system," *Meas. Sensors*, vol. 27, p. 100721, 2023, doi: https://doi.org/10.1016/j.measen.2023.100721.

[3] A. Shinde, S., Choudhari, S., Khaladkar, A., & Randive, "Car Black Box System," *Int. Res. J. Innov. Eng. Technol.*, vol. 8, no. 4, pp. 194–199, 2024, doi: https://doi.org/10.47001/IRJIET/2024.804027.

[4] S. Sethuraman and S. Santhanalakshmi, "Implementing Vehicle Black Box System by IoT based approach," *Proc. 4th Int. Conf. Trends Electron. Informatics, ICOEI 2020*, pp. 390–395, Jun. 2020, doi: 10.1109/ICOEI48184.2020.9142906.

[5] M. J. Prasad, S. Arundathi, N. Anil, H. Harshikha, and B. S. Kariyappa, "Automobile black box system for accident analysis," *2014 Int. Conf. Adv. Electron. Comput. Commun. ICAECC 2014*, Jan. 2015, doi: 10.1109/ICAECC.2014.7002430.

[6] J. M. Hu, J. Li, and G. H. Li, "Automobile anti-theft system based on GSM and GPS module," *Proc. - 5th Int. Conf. Intell. Networks Intell. Syst. ICINIS 2012*, pp. 199–201, 2012, doi: 10.1109/ICINIS.2012.86.

[7] D. P. Raja, G. Barkavi, S. Aishwarya, K. Keerthana, and V. Vasudevan, "Alcohol Detection and Emergency Alert System using IoT," *Proc. - 2022 6th Int. Conf. Intell. Comput. Control Syst. ICICCS 2022*, pp. 395–400, 2022, doi: 10.1109/ICICCS53718.2022.9788419.

[8] M. N. Mohammed, Y. Ghanesen, S. Al-Zubaidi, M. A. M. Ali, O. Ismael Al-Sanjary, and N. S. Zamani, "Investigation on Carbon Monoxide Monitoring and Alert System for Vehicles," *Proc. - 2019 IEEE 15th Int. Colloq. Signal Process. its Appl. CSPA 2019*, pp. 239–242, Apr. 2019, doi: 10.1109/CSPA.2019.8696001.

[9] D. A. H. and F. S. J V Moniaga, S R Manalu, "Diagnostics vehicle's condition using obd-ii and raspberry pi technology: study literature," *J. Phys. Conf. Ser.*, vol. 978, p. 012011, 2018, doi: 10.1088/1742-6596/978/1/012011.

[10] R. Malekian, N. R. Moloisane, L. Nair, B. T. Maharaj, and U. A. K. Chude-Okonkwo, "Design and Implementation of a Wireless OBD II Fleet Management System," *IEEE Sens. J.*, vol. 17, no. 4, pp. 1154–1164, Feb. 2017, doi: 10.1109/JSEN.2016.2631542.