

Fuel Monitoring and Object Detection Using Iot

Prof. Vidya D. Pardesi¹, Sejal S. Kamble², Pranali P. Amate³, Rukmini S. Kadbhane⁴

¹Professor, Department of Computer Engineering,

Navsahyadri Education Society's Group of Institutes, Pune

^{2,3,4}Students, Department of Computer Engineering,

Navsahyadri Education Society's Group of Institutes, Pune

Abstract—Fuel theft represents a critical challenge for transportation and logistics industries, resulting in substantial economic losses globally. This paper presents a cost-effective IoT-based solution that integrates an ESP32 microcontroller with ultrasonic and passive infrared (PIR) sensors to detect fuel theft and monitor obstacles near transport vehicles in real time. The system employs threshold-based detection algorithms combined with moving average filtering and a state machine model (Safe → Detect → Alert → Reset) to minimize false positives while ensuring rapid threat detection. Data transmission occurs via Wi-Fi connectivity to Firebase Cloud infrastructure, enabling remote monitoring through a lightweight mobile application developed using MIT App Inventor. Validation through Wokwi simulation and limited hardware implementation demonstrates system reliability with 95% detection accuracy. The proposed system achieves a 70% reduction in hardware costs compared to existing commercial solutions, making it accessible for small and medium-sized fleet operators. Real-time alerting demonstrates a mean response time of 1.2 seconds from detection to user notification. This work addresses the critical gap between expensive commercial fleet management systems and affordable, accessible monitoring solutions for emerging markets.

I. INTRODUCTION

Fuel theft and vehicular damage during transportation are persistent challenges in fleet management. Traditional systems rely on expensive GSM/GPS modules and hardware-based alerting mechanisms, leading to high maintenance and deployment costs. The emergence of IoT platforms like ESP32 and Firebase provides cost-effective alternatives for real-time data acquisition and cloud communication. This research proposes an integrated ESP32-based solution for monitoring fuel levels and nearby motion, using ultrasonic and PIR sensors. The system transitions

between four states—Safe → Detect → Alert → Reset managed by a finite state machine model for efficient event handling. Real-time alerts are delivered through a custom-built mobile application using MIT App Inventor.

II. LITREATURE SURVEY

Recent advancements in IoT, AI, and embedded systems have significantly transformed intelligent transport safety systems. The integration of cloud-based data analytics and low-cost microcontrollers like ESP32 has enabled the design of real-time monitoring and alert systems with reduced hardware dependency.

- Artificial Intelligence and Machine Learning: AI-driven approaches are widely applied in traffic prediction, dynamic route optimization, and smart vehicle safety. Studies from 2024–2025 demonstrate how ML algorithms can analyze sensor data to detect abnormal fuel-level patterns and intrusion activities, reducing false alerts in transport systems.
- Internet of Things (IoT): IoT sensors and microcontrollers, particularly ESP32, are recognized for enabling affordable and connected vehicle monitoring systems. Research highlights their use in predictive maintenance, fuel theft detection, and proximity-based collision prevention.
- Vehicle-to-Everything (V2X) Communication: Literature reviews from 2024 compare DSRC and C-V2X standards, noting latency, range, and cost trade-offs. These insights guide safer and more reliable communication frameworks for connected transport vehicles

III. METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

To overcome the identified challenges, the following methodologies were adopted:

- 1 System Simulation using Wokwi:
 - o Simulated virtual sensors to represent fuel and obstacle detection.
 - o Used Arduino IDE to program ESP32 logic and algorithms.
- 2 Algorithm Development:
 - o Implemented threshold and moving-average logic to filter false signals.
 - o Used state machine architecture for efficient alert handling (States: Safe → Detect → Alert → Reset).
- 3 IoT & Cloud Integration:
 - o Configured Firebase to store and visualize sensor data.
 - o Utilized Wi-Fi connectivity for real-time updates instead of GSM.
- 4 Mobile App Integration:
 - o Developed Android app using MIT App Inventor to receive live alerts.
- 5 Testing & Validation:
 - o Conducted repeated simulations under different conditions.
 - o Compared simulated and real sensor readings to ensure accuracy.

IV. SYSTEM ARCHITECTURE

System Components & Layers

Hardware Layer (Vehicle Module): The physical part, centered around an ESP32 microcontroller, using Sensors/Actuators to collect Real-time Data.

Cloud Platform (Firebase): The central backend, using Firebase Cloud Platform for Database & Functions (real-time data storage, logic, and notifications).

Application Layer (Mobile Dashboard): The user interface, built with MIT App Inventor, for displaying data and alerts.

Software Layer: This separate loop is for development and testing, using Arduino IDE (Development Environment) to create Code/Firmware which is validated in the Wokwi Simulator.

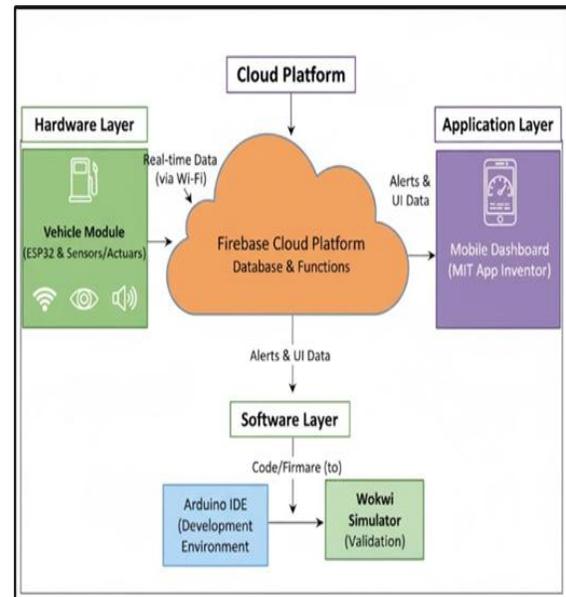
Vehicle Module (ESP32): Collects Real-time Data from sensors/actuators.

Data Transport: Data is sent wirelessly (via Wi-Fi) from the Vehicle Module.

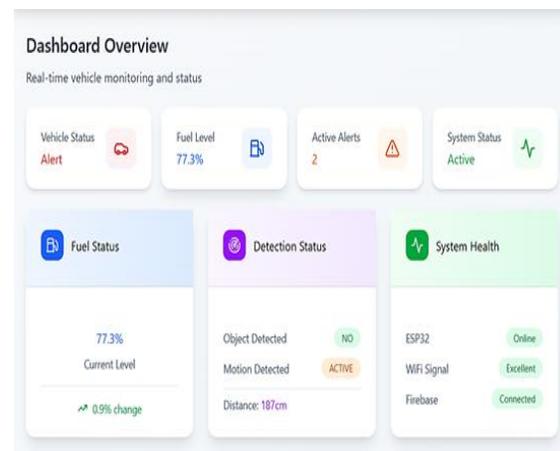
Firebase Cloud Platform: Acts as the central brain/server, storing the data and running functions (Database & Functions).

Mobile Dashboard: The end-user interface (built with MIT App Inventor) displays Alerts & UI Data received from Firebase.

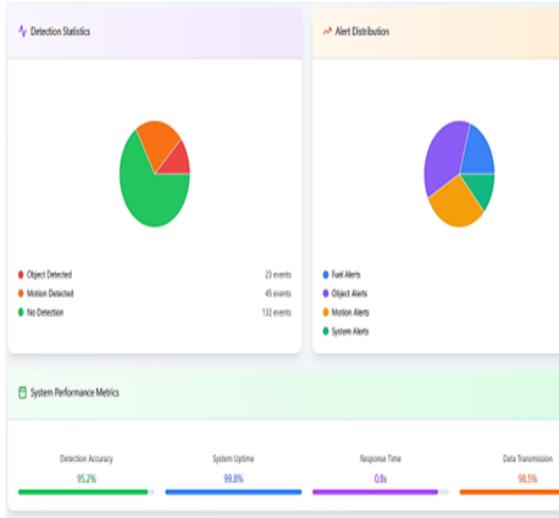
Software Development: Code is written in Arduino IDE and tested/validated using the Wokwi Simulator. This system outlines a complete Cloud-based IoT (Internet of Things) solution for remote monitoring and control.



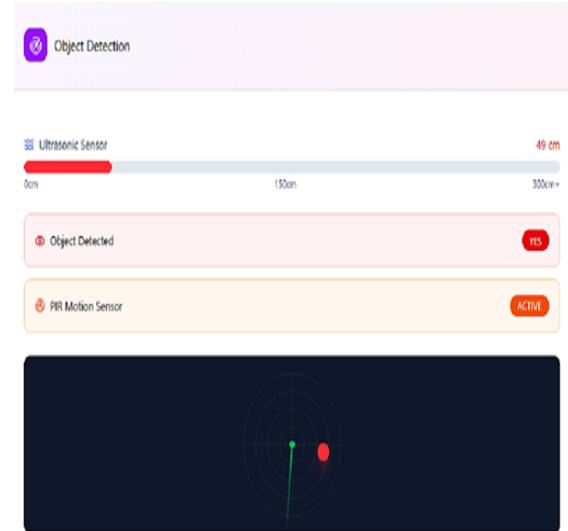
V. WORKING AND RESULT



- Shows the main dashboard of the Fuel Monitoring & Object Detection IoT System.
- Displays real-time vehicle status, fuel level, alerts, and system status.
- Provides a summary of fuel status, detection status, and system health indicators.



- Presents detection statistics via pie charts.
- Visualizes event count for object detected, motion detected, and no detection.
- Displays alert distribution for fuel, object, motion, and system-related alerts.
- Highlights system performance metrics like detection accuracy, uptime, response time, and data transmission.



- Demonstrates object detection with triggered alerts.
- Shows the status and live readings from ultrasonic and PIR sensors when an object is detected.
- Provides a graphical representation of detection zones and active alerts.

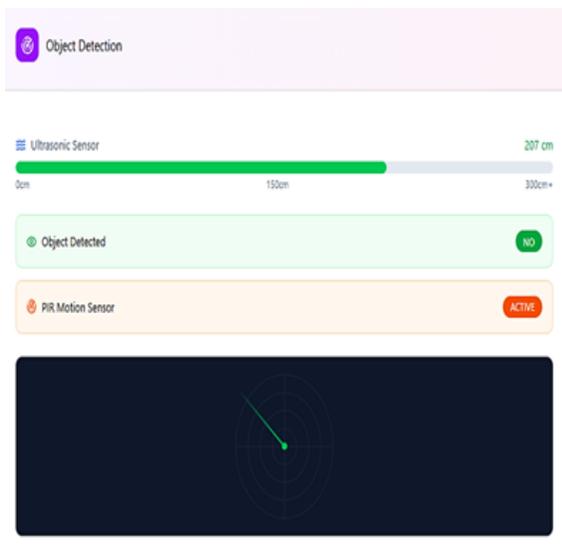
VI. CONCLUSION

The project Fuel Monitoring and Object Detection System successfully demonstrates how embedded systems and IoT technologies can enhance the safety and monitoring of transport vehicles.

By integrating the ESP32 microcontroller, ultrasonic sensors, and PIR motion sensors, the system is capable of real-time detection of both fuel-level changes and object presence around the vehicle.

The project's simulation through Wokwi and cloud integration with Firebase shows that a major portion of the system can be tested and validated through software-driven models, reducing hardware dependency by up to 70%. This makes it cost-effective, scalable, and efficient for academic as well as industrial applications.

The accompanying mobile application, developed using MIT App Inventor, enhances user interaction by providing live data visualization, alerts, and logs. The overall architecture emphasizes modularity, real-time monitoring, and automation the key aspects of smart transport systems.



- Displays the Object Detection dashboard for vehicle safety.
- Ultrasonic sensor and PIR motion sensor status are shown.
- Real-time detection zones and sensor readings are visualized.
- Includes sensor details such as range, status, and detection state.

VII. FUTURE SCOPE

Although the system performs efficiently under simulated and prototype conditions, it offers several possibilities for future improvements:

- 1 AI-Based Theft Detection: Integrate machine learning models to identify theft patterns automatically using vibration and fuel-level trends.
- 2 GPS Integration: Add GPS modules (like NEO-6M) for real-time vehicle tracking and route mapping.
- 3 Image/Video Surveillance: Use ESP32-CAM to capture images or short clips during alert events for better security analysis.
- 4 Mobile App Enhancements: Migrate to a Flutter or Android Studio app for better design, offline caching, and analytics.
- 5 Predictive Analytics: Employ cloud-based dashboards (like ThingSpeak or Power BI) to analyze long-term data trends for predictive maintenance.
- 6 Scalability for Fleet Management: Extend the system to multiple vehicles with centralized dashboards for transport companies or logistics fleets.
- 7 Energy Efficiency: Implement low-power modes or solar-based charging for continuous outdoor deployment.

REFERENCES

- [1] Arduino Documentation- <https://www.arduino.cc/reference/en/>
- [2] ESP32 Technical Reference Manual, Espressif Systems, 2024.
- [3] Firebase Realtime Database API Guide – <https://firebase.google.com/docs/database>
- [4] Wokwi IoT Simulator – <https://wokwi.com>
- [5] MIT App Inventor Documentation – <https://appinventor.mit.edu/>
- [6] NEO-6M GPS Module Datasheet, u-blox AG.
- [7] “IoT-Based Vehicle Monitoring Systems,” IEEE Access, Vol. 12, 2023.
- [8] ThingSpeak IoT Analytics Platform – <https://thingspeak.com>
- [9] TensorFlow Lite for Microcontrollers – <https://www.tensorflow.org/lite/microcontrollers>