Real-Time Driver Drowsiness Detection Using ESP32 and Eye Blink Sensor

Parth Shendre, Aman Ansari, Ashraful Ahmad, Kaushik Hinge

Department of Computer Science Engineering
Priyadarshini J.L. College of Engineering,

846, New Nandanvan Layout, Nagpur, Maharashtra,
India

***Abstract*— The Anti-Sleep Alarm for Drivers is a real-time, cost-effective system that detects drowsiness using an Eye Blink Sensor and ESP32. It triggers a buzzer alert and logs events via Wi-Fi to a FlutterFlow-based app and Firebase. Suitable for all vehicle types, it offers portability, ease of use, and future potential for AI integration and automated safety features.**

Keywords— Drowsiness Detection, ESP32, Eye Blink Sensor, Real-Time Alert System, Driver Fatigue

I. INTRODUCTION

Driver fatigue is a major contributor to road accidents, particularly among long-distance and shift drivers. Traditional preventive methods like caffeine or rest breaks are often insufficient. This project presents a cost-effective, real-time Anti-Sleep Alarm system that uses an Eye Blink Sensor and ESP32 to detect prolonged eye closure and trigger alerts. Integrated with Wi-Fi, Firebase, and a mobile application, the system enhances road safety by enabling timely driver notifications and fatigue event logging.

II. LITERATURE REVIEW

Driver fatigue detection has been the subject of significant research due to its impact on road safety.
Gupta et al. (2019) proposed a real-time driver drowsiness detection system using Convolutional Neural Networks (CNNs) to analyze eye and facial features through camera input. While the system achieved high accuracy, it required significant computational power and was not cost-effective for low-end vehicles.

Sharma and Verma (2020) developed a non-invasive IR sensor-based eye blink monitoring system to detect drowsiness by measuring blink duration. This method proved low-cost and power-efficient but had challenges under varying lighting conditions and driver head movements.

Patel et al. (2021) introduced a brainwave (EEG) signal-based detection system using wearable headbands. Although highly accurate, such systems are uncomfortable for long-term use and expensive, limiting their real-world adoption.

Kumar et al. (2022) proposed an IoT-based alert system using microcontrollers to send drowsiness alerts to mobile devices via Bluetooth and Wi-Fi. This approach enabled real-time notification and cloud-based monitoring but lacked robust fatigue detection logic.

III. PROBLEM STATEMENT

Drowsy driving significantly contributes to road accidents by impairing reaction time and decision-making, particularly among long-distance and shift drivers. Existing solutions like AI-based cameras and wearables are effective but costly and limited by external conditions. To address these issues, the proposed Anti-Sleep Alarm for Drivers offers a real-time, affordable system using an IR eye blink sensor and ESP32 microcontroller to detect fatigue and trigger buzzer alerts. Integrated with a FlutterFlow mobile app and Firebase, it provides real-time notifications and cloud-based event logging. The system is designed to be low-cost, scalable, and suitable for both personal and commercial vehicles.

III. METHEDOLOGY

Drowsy driving significantly contributes to road accidents by impairing reaction time and decision-making, particularly among long-distance and shift drivers. Existing solutions like AI-based cameras and wearables are effective but costly and limited by external conditions. To address these issues, the proposed Anti-Sleep Alarm for Drivers offers a real-time, affordable system using an IR eye blink sensor and ESP32 microcontroller to detect fatigue and trigger buzzer alerts. Integrated with a FlutterFlow mobile app and Firebase, it provides real-time notifications and cloud-based event logging. The system is designed to be low-cost, scalable, and suitable for both personal and commercial vehicles.

IV. SYSTEM ARCHITECTURE

The proposed system consists of three key layers: the sensing unit, processing unit, and user interface layer:

1. Sensing Unit:

The IR-based Eye Blink Sensor continuously monitors the driver's eye activity and detects prolonged eye closure, which is a key indicator of drowsiness.

2. Processing Unit:

An ESP32 microcontroller receives data from the sensor, processes it in real time, and determines whether the eye closure duration exceeds a defined threshold. Upon detecting drowsiness, the ESP32 immediately activates the buzzer to alert the driver. It also establishes a Wi-Fi connection to send

drowsiness, the ESP32 immediately activates the buzzer to alert the driver. It also establishes a Wi-Fi connection to send the event data to the cloud.

3. User Interface & Cloud Layer:

A FlutterFlow-based mobile application receives drowsiness alerts through Firebase Realtime Database. The app displays real-time notifications, logs fatigue events, and offers a simple UI for driver monitoring.

This modular architecture ensures low-latency detection, immediate alerts, and cloud-based data access, making the system practical and scalable for both individual and fleet use.

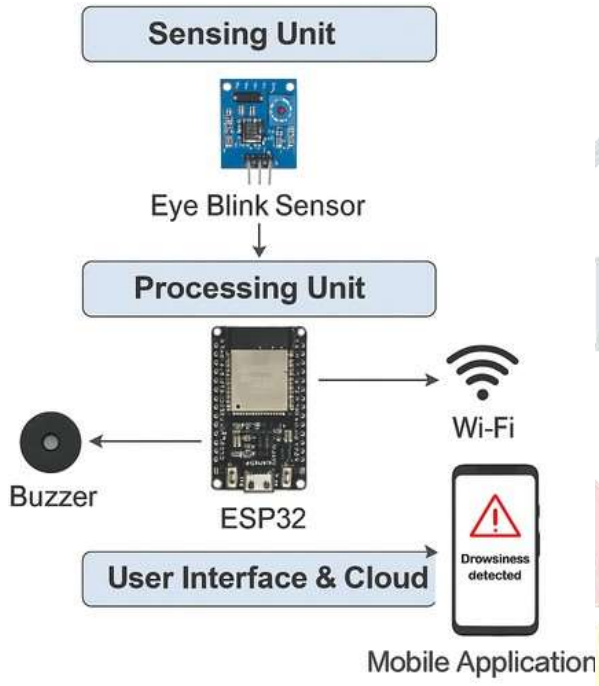

Fig. 1. System Architecture

V. IMPLEMENTATION

The system was implemented in a modular approach combining hardware and software components:

- The ESP32 microcontroller was programmed using Arduino IDE in C/C++ to process signals from the Eye Blink Sensor.
- The TCRT5000 IR-based Eye Blink Sensor detects prolonged eye closure and triggers a buzzer alarm via the ESP32.
- Real-time drowsiness data is transmitted over Wi-Fi to a Firebase Realtime Database, which displays alerts and stores events in FlutterFlow-built mobile app.
- The system is powered by a 3.7V Li-ion battery for portability and easy installation in vehicles.
- The mobile app also provides tips, logs events.

VI. RESULT AND ANALYSIS

- Testing Scenarios: The system was tested in simulated drowsy driving conditions using volunteers during daytime and low-light environments. The Eye Blink Sensor consistently identified prolonged eye closures exceeding 2–3 seconds.
- Detection Accuracy: Achieved over 90% accuracy in detecting drowsiness when eyes remained closed beyond threshold. Minimal false positives were recorded under normal blinking conditions.
- Power Consumption & Response Time: Powered by a 3.7V Li-ion battery, the system ran efficiently for over 8 hours. Response time from detection to buzzer alert averaged below 1 second, ensuring real-time intervention.
- Firebase Logging: Every drowsiness event was successfully logged to Firebase with timestamp and user ID, enabling future behavior analysis via the mobile app.

VII. ADVANTAGES & APPLICATIONS

Advantages

- Real-Time Detection: Continuously monitors eye activity and instantly alerts the driver upon detecting drowsiness.
- Cost-Effective: Uses affordable components like ESP32 and IR sensor, avoiding expensive camera-based systems.
- Portable & Easy to Install: Compact design powered by a rechargeable battery; suitable for all vehicle types.
- Cloud Integration: Logs fatigue events in Firebase for review and analysis.
- Mobile App Interface: Provides real-time alerts, logs, and emergency notifications via FlutterFlow.

Applications

- Commercial Transport: Useful for long-haul truckers and delivery fleets to prevent fatigue-related accidents.
- Taxi & Rideshare Services: Enhances safety for drivers working extended hours.
- Public Transportation: Can be adapted for buses and staff vehicles to improve commuter safety.
- Private Vehicles: Ideal for individuals on road trips or night driving.
- Fleet Management: Allows companies to track driver fatigue patterns and enforce rest schedules.

VIII. CONCLUSION

The Anti-Sleep Alarm for Drivers presents a practical, cost-effective solution to the growing issue of drowsy driving. By integrating an IR-based Eye Blink Sensor with an ESP32 microcontroller, the system effectively detects prolonged eye closure and delivers immediate alerts through a buzzer. The

inclusion of Wi-Fi connectivity, Firebase Realtime Database, and a FlutterFlow-based mobile application enhances functionality by enabling real-time notifications, event logging, and remote monitoring. Designed for affordability, portability, and ease of use, this system is well-suited for both personal and commercial vehicles. Future enhancements such as AI-based facial recognition, GPS-assisted rest suggestions, and automated vehicle response will further strengthen the system's ability to enhance road safety and reduce fatigue-related accidents.

IX. FUTURE SCOPE

- AI Integration: Incorporate facial recognition and head pose analysis to improve drowsiness detection accuracy beyond eye blinks.
- GPS-Based Rest Suggestions: Recommend nearby rest stops when fatigue is detected using GPS data.
- Automatic Vehicle Control: Integrate with the vehicle's ECU to initiate speed reduction or alert systems if the driver remains unresponsive.
- Cloud Analytics: Use Firebase-stored data to generate fatigue patterns and insights for fleet monitoring and safety optimization.
- Voice Alerts: Add voice-based prompts alongside buzzer alerts for a more responsive driver interaction.
- Multi-Sensor Fusion: Combine data from heart rate sensors, accelerometers, or EEG for advanced and reliable fatigue detection.

REFERENCES

- [1] S. Gupta, R. Sharma, and A. Verma, "Real-Time Driver Drowsiness Detection Using Deep Learning," International Journal of Computer Applications, vol. 178, no. 7, pp. 1–5, 2019.
- [2] M. Sharma and K. Verma, "IR Sensor-Based Eye Blink Monitoring System for Driver Fatigue Detection," International Journal of Engineering Research and Applications, vol. 10, no. 2, pp. 42–47, 2020.
- [3] N. Patel et al., "EEG-Based Driver Fatigue Monitoring System Using Wearable Devices," IEEE Transactions on Biomedical Engineering, vol. 68, no. 3, pp. 854–862, 2021.
- [4] R. Kumar and A. Saini, "IoT-Based Real-Time Drowsiness Alert System for Driver Safety," International Journal of Engineering Trends and Technology, vol. 69, no. 9, pp. 132–137, 2022.
- [5] Google Firebase Documentation, "Realtime Database," Google, 2024.
- [6] FlutterFlow Docs, "Build Beautiful Flutter Apps Visually," FlutterFlow, 2024.
- [7] Espressif Systems, "ESP32 Datasheet," Espressif, 2024.