**DEPARTMENT OF MECHANICAL ENGINEERING**

# [RV COLLEGE OF ENGINEERING](https://www.rvce.edu.in/)****®****

###### **(Autonomous Institution affiliated to VTU, Belagavi)**



**PROJECT REPORT**

**On**

**Anti-Sleep Driver Cap Alarm System**

**Submitted in partial fulfilment of First year IDEA Lab**

**Submitted by**

* + - * + **Samrudh S Malali (RVCE25BEC159)**
        + **Sanchit Kumar (RVCE25BEC195)**
        + **Shashank Kota (RVCE25BEC118)**
        + **Sarvajit (RVCE25BEC127)**

**Under the guidance of**

**Prof. G Anjaneya**

**Prof. Dr. Gangadhar Angadi**

**2025-2026**

**ABSTRACT:**

Drowsiness while driving is a major cause of road accidents worldwide, posing a significant threat to road safety. According to the statistics, Drowsy driving is estimated to be a factor in up to 10 times more crashes than officially reported, with one study finding it in 17.6% of fatal crashes (2017-2021). In some countries, drowsiness accounts for 20-40% of accidents, and fatigue-related collisions cost billions annually. The National Highway Traffic Safety Administration reports an estimated 100,000 crashes each year are caused primarily by drowsy driving, resulting in more than 71,000 injuries and $12.5 million in damages.

Hence it is imperative to avoid this road accidents due to Driver drowsiness. This project aims to mitigate this risk by developing a compact, wearable **Anti-Sleep Driver Cap Alarm System**. The primary objective is to detect the onset of microsleep or fatigue by monitoring the driver's head movements.

The system utilizes an **ADXL345 accelerometer** to continuously measure the X, Y, and Z-axis acceleration data, determining the head's orientation. An **ESP32** microcontroller processes this data via the I2C protocol, calculating the tilt angle relative to a calibrated upright position. When the head tilts forward or sideways beyond a specific threshold (e.g., indicating nodding off), the system triggers an immediate auditory alert via a **buzzer**.

To ensure usability and safety, the system includes a **capacitive touch sensor** that serves as a manual override to reset the alarm or deactivate the system, preventing distraction from false alarms. Experimental results demonstrate that the prototype offers a fast response time of less than 1 second and effectively distinguishes between normal driving movements (like checking mirrors) and drowsiness symptoms. This solution provides a low-cost, non-intrusive alternative to expensive camera-based commercial systems.

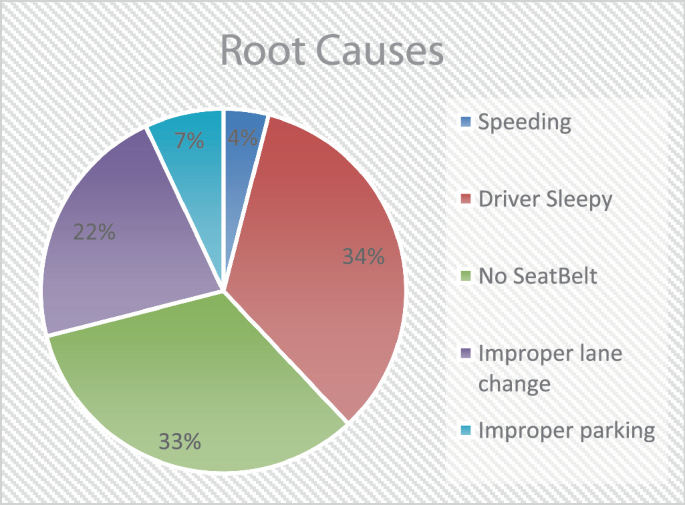


Table of Contents

1. Chapter 1: Introduction and Objectives
2. Chapter 2: Literature Review
3. Chapter 3: System Design and Architecture
4. Chapter 4: Hardware and Software Implementation
5. Chapter 5: Experimental Setup
6. Chapter 6: Results & Discussion
7. Chapter 7: Conclusion & Future Work
8. References
9. Appendices

**CHAPTER 1**

**INTRODUCTION:**

1. Background

Drowsiness while driving is one of the major causes of road accidents globally. The inability of a driver to stay alert due to fatigue can lead to catastrophic consequences, particularly on highways where reaction times are critical.

2.Problem Definition

The project addresses the critical issue of driver drowsiness. It specifically targets long-distance logistics drivers who often carry heavy loads over extended periods and timeframes, making them highly susceptible to fatigue. Existing commercial monitoring systems, such as eye-tracking cameras, are often expensive and intrusive.

3. Motivation

The primary motivation is to enhance road safety by reducing risks caused by microsleep or fatigue. The goal is to offer a low-cost, portable, and non-intrusive alternative to complex commercial systems, thereby making safety technology accessible to a wider range of drivers.

**OBJECTIVES:**

* 1. Detect Driver Drowsiness: To continuously monitor head positioning using 3-axis acceleration data (ADXL345) to identify sudden drooping or forward tilting associated with microsleep.
  2. Instantaneous Alerting: To provide an immediate auditory warning strong enough to wake the driver upon detecting unsafe movements.
  3. Safety Enhancement: To implement a reliable manual override that allows the driver to regain control of the system quickly**.**

**CHAPTER 2**

**LITERATURE REVIEW:**

**1.Kiruthiga et al. (2018) – "Arduino Based Real-Time Driver Drowsiness Detection"**

This paper presented a prototype using an Arduino microcontroller integrated with multiple sensors, including an accelerometer for head nod detection and a pulse sensor for heart rate monitoring. It demonstrated the feasibility of processing sensor data locally on an Arduino board without needing complex external computers, providing a foundational architecture for low-cost embedded safety systems. The system triggers an alert when abnormal head movements or irregular heart-rate patterns associated with drowsiness are detected, helping to warn the driver in real time. Experimental results showed that the prototype could reliably detect early signs of fatigue under controlled conditions. Although the accuracy was limited by simple threshold-based algorithms, the work highlighted the practicality of Arduino-based implementations for real-world applications. The study also emphasized affordability and ease of integration, making it suitable for further enhancement using advanced signal processing or machine-learning techniques.

**2.Aras et al. (2023) – "Head Movement-Based Driver Drowsiness Detection: A Review"**

It provides a comprehensive review of state-of-the-art techniques, specifically focusing on "nodding" patterns. The study concluded that head posture analysis (Head Pose Estimation) is a reliable physiological indicator of drowsiness. It highlighted that while camera-based systems are popular, they struggle with lighting conditions, whereas physical movement sensors offer consistent performance regardless of ambient light. The review compared vision-based, inertial sensor–based, and hybrid approaches, outlining their advantages and limitations in real driving environments. It emphasized that accelerometer and gyroscope sensors can effectively capture subtle nodding and tilting motions linked to fatigue. The authors also discussed challenges such as sensor placement, driver variability, and false positives during normal head movements. Overall, the paper suggested that combining head movement data with other physiological or behavioral cues can significantly improve detection accuracy and system robustness.

**3.Lee et al. (2016) – "Standalone Wearable Driver Drowsiness Detection System in a Smartwatch"**

This research designed a distraction-free wearable system using a smartwatch's built-in accelerometer and gyroscope. By analyzing the magnitude of wrist and body movements, the system achieved 98.15% accuracy using a Support Vector Machine (SVM) classifier. This validates the concept that wearable sensors are a non-intrusive and highly accurate alternative to camera-based monitoring. The system operated independently without requiring external devices, making it practical for everyday driving scenarios. Real-time data processing on the smartwatch demonstrated low latency and energy efficiency suitable for long-duration use. The study also showed robustness against variations in driving posture and road conditions. These results highlight the strong potential of wearable-based drowsiness detection for scalable and user-friendly deployment.

**4.Mittal et al. (2016) – "Head Movement-Based Driver Drowsiness Detection: A Review of State-of-Art Techniques"**

This paper categorized various detection methods, comparing subjective measures against objective sensor data. The review identified that lateral (side-to-side) and anterior-posterior (forward-backward) head movements are distinct markers of fatigue. This supports the logic of using a 3-axis accelerometer (ADXL345) to detect both forward nodding and side tilting. The authors further discussed signal processing techniques used to extract meaningful head-motion features from raw sensor data. They highlighted that accelerometer-based systems are computationally efficient and well suited for real-time embedded applications. The review also noted challenges such as inter-driver variability and distinguishing drowsy movements from intentional head turns. Overall, the study reinforced the effectiveness of multi-axis inertial sensors for reliable and cost-effective drowsiness detection systems.

**5.IJIRMPS (2024) – "Improving Road Safety with Real-Time Monitoring: A Drowsiness Detection System Utilizing IR Sensors and Arduino Technology"**

This recent study focused on a low-cost implementation using Arduino and IR sensors to detect eye closure. While effective, the authors noted limitations in detecting drowsiness if the driver wears sunglasses. This highlights a critical "Research Gap" that your project addresses: the Driver Cap system works perfectly even if the driver is wearing sunglasses or driving at night, as it relies on head tilt rather than eye visibility. The study demonstrated reliable real-time alert generation under normal lighting conditions, validating the practicality of IR-based eye monitoring. However, it emphasized the need for alternative physiological indicators to overcome visual occlusion issues. By shifting focus to head movement and posture, the Driver Cap system enhances robustness across diverse driving scenarios. This approach also improves user comfort and system reliability without increasing computational or hardware complexity.

**6.Shourie et al. (2023) – "Evolution and Recent Trends in Detecting Driver Drowsiness"**

This survey analyzed the shift from vehicle-based sensors (steering patterns) to behavioral sensors (facial features/head tilt). It concluded that non-intrusive real-time monitoring systems are the future of road safety. The paper emphasized that systems must be "compact" and "cost-effective" to see widespread adoption in developing countries, a criteria our project meets perfectly. The authors also highlighted the growing preference for wearable and embedded sensor solutions due to their ease of deployment and low maintenance. They discussed how advances in low-power microcontrollers and MEMS sensors have enabled continuous monitoring without driver discomfort. The survey pointed out that hybrid approaches can further improve accuracy while keeping costs minimal. Overall, the findings strongly support simple, head-movement–based systems as a practical and scalable solution for real-world implementation.

**7.Kumar et al. (2025) – "Real-Time Driver Drowsiness Detection System using IoT-based Physiological Monitoring and Web Interface"**

This research proposed a system integrating a **Node MCU** microcontroller with sensors to monitor driver status in real-time. Unlike standalone systems that only buzz locally, this project utilized IoT connectivity to push data to a **cloud server**. If drowsiness was detected, the system not only triggered a local alarm but also sent the vehicle's **GPS coordinates** and a "Critical Alert" notification to a fleet manager via a web interface. This study validates the potential of upgrading your current Arduino-based prototype into a connected "Smart Cap" that can notify family members or supervisors during long-haul trips. The results demonstrated improved response time and situational awareness through remote monitoring, especially for commercial and long-haul transportation. The authors emphasized scalability, showing that multiple vehicles could be monitored simultaneously from a centralized dashboard. However, they also noted higher power consumption and system complexity compared to standalone designs. This reinforces the idea that a modular upgrade path from a simple Arduino-based cap to an IoT-enabled smart safety device is both feasible and impactful.

**Research Gap**

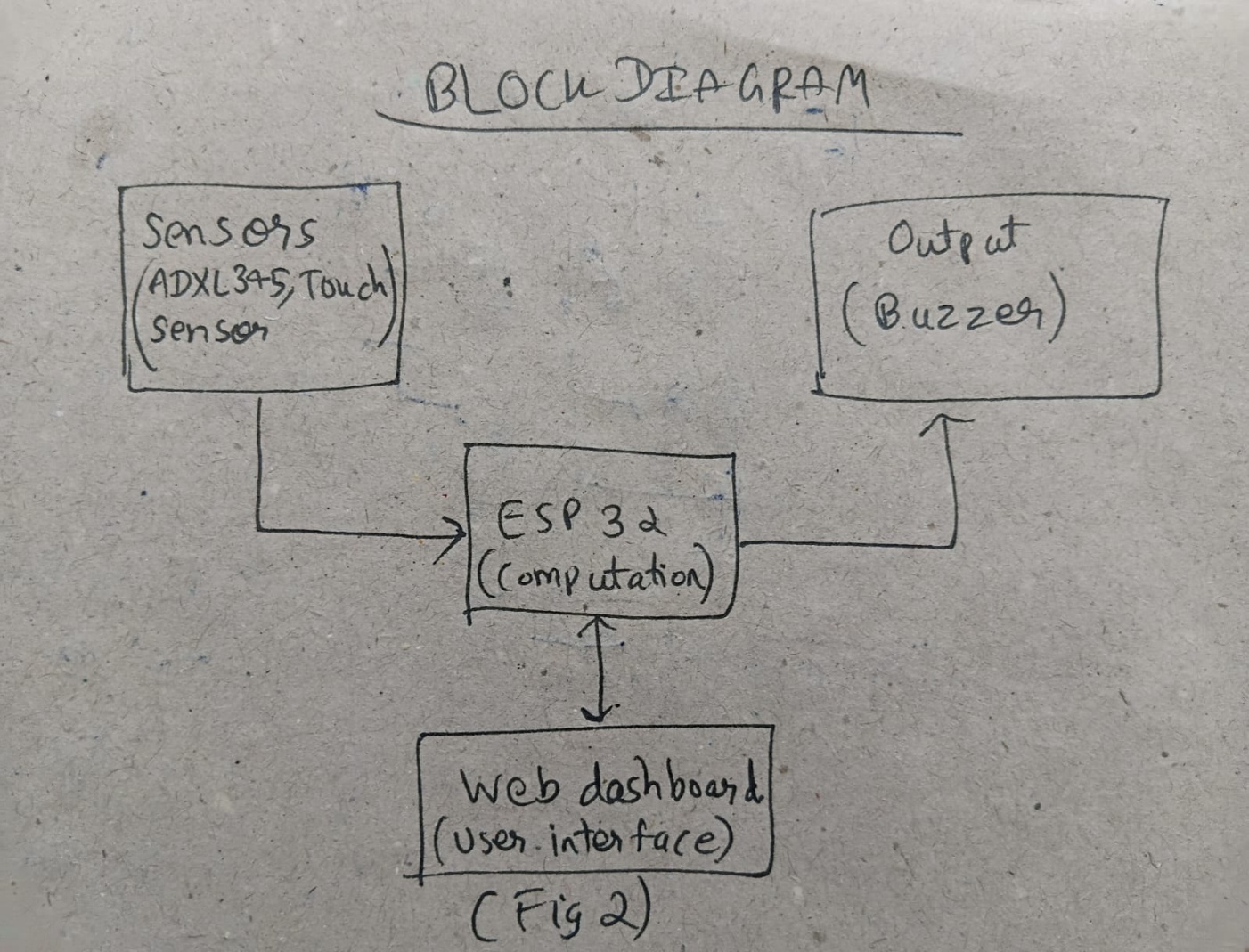
While camera systems are accurate, they suffer from issues related to lighting conditions (night driving) and privacy. This project bridges the gap by providing a device that is cost-effective, portable, and immune to lighting conditions, as it relies solely on physical orientation.

**CHAPTER 3:System Design and Architecture**

**Component Selection:**

1. ESP32:Selected for its ease of use and support for I2C communication and wifi connection.
2. Sensor (ADXL345): A digital accelerometer chosen for its ability to measure static acceleration of gravity in tilt-sensing applications.
3. Touch Sensor: A capacitive sensor selected to act as a reset switch because it requires no mechanical force, making it easy to tap while driving.
4. Buzzer: An active buzzer to provide a loud, immediate wake-up signal.
5. Jumper wires and Breadboard.

**BLOCK DIAGRAM**

****

**Fig1**

**A diagram of a system

AI-generated content may be incorrect.**

**Here** we have used Telegram bot API to send messages to the owner using username and id.

Instead of web dashboard ,we have used telegram messaging bot API for user interface.

**Working Principle**

The ADXL345 continuously measures X, Y, and Z-axis acceleration. The ESP32 reads this data via the I2C protocol and calculates the tilt angle relative to a calibrated "upright" position.

* Decision Logic: If the calculated Pitch Angle > Threshold (e.g., 20° forward), the alarm triggers.
* Reset Logic: If the touch sensor is pressed, the alarm is silenced and the system resets.
* Also it sends the owner a message through Telegram whether the driving is sleeping or awake.

SYSTEM WORKFLOW

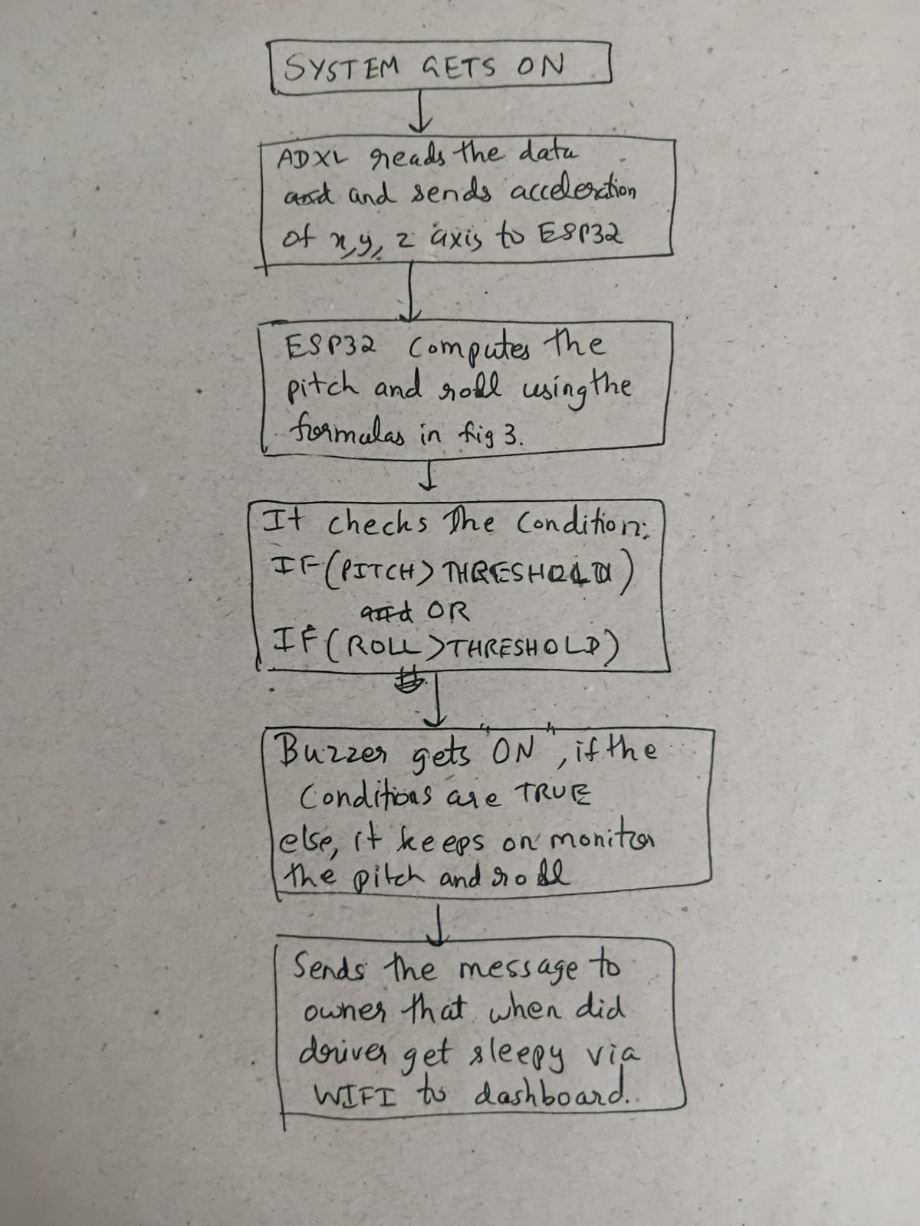


FIG2

**CHAPTER 4:Hardware and software implementation**

**Circuit implementation:**

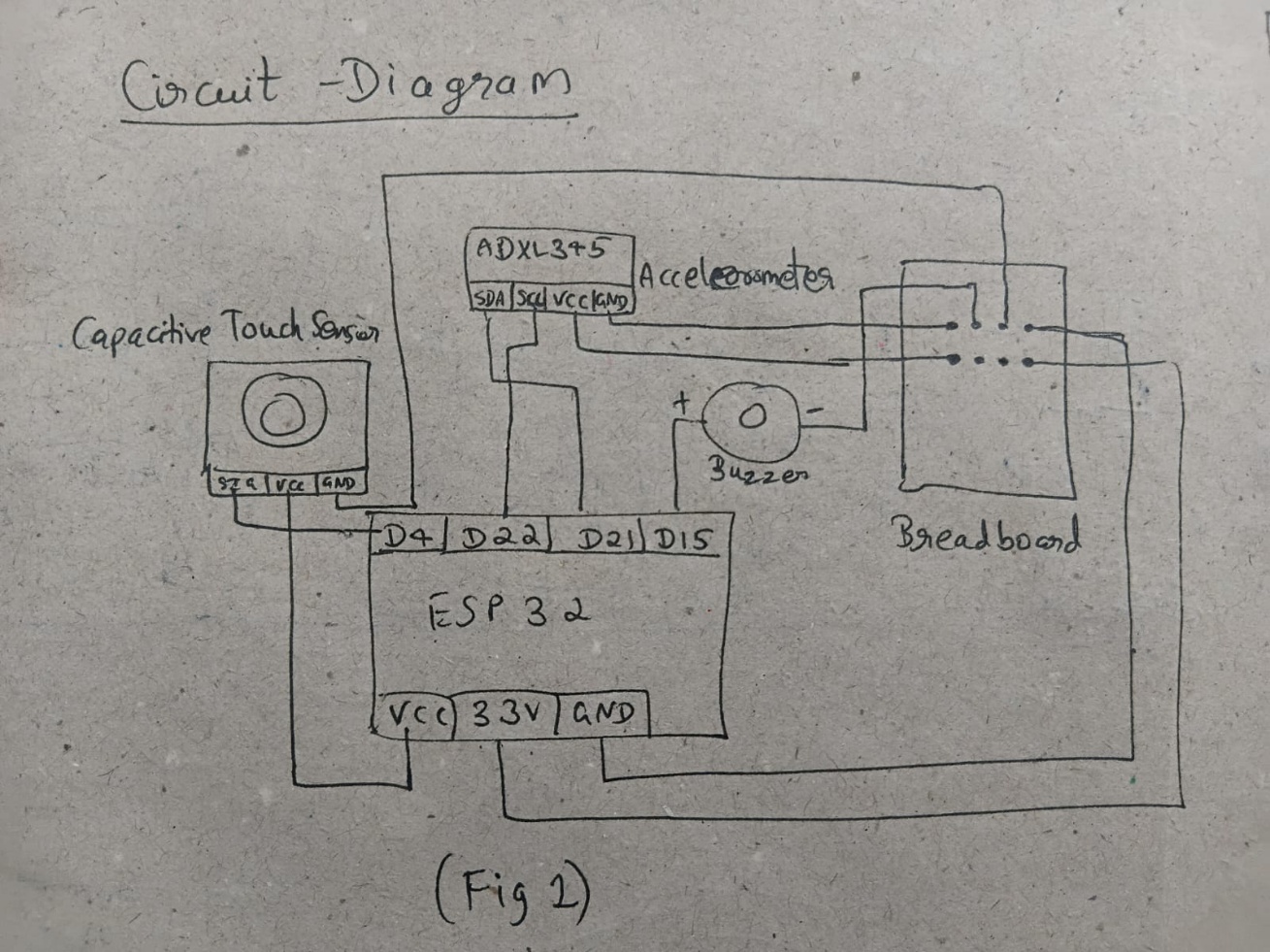
****

FIG 3

**Software Logic (Formulas)**

The system uses trigonometry to convert raw acceleration data into angles:

A close up of a math problem

AI-generated content may be incorrect.

FIG4

**CHAPTER 5:** **Experimental Setup**

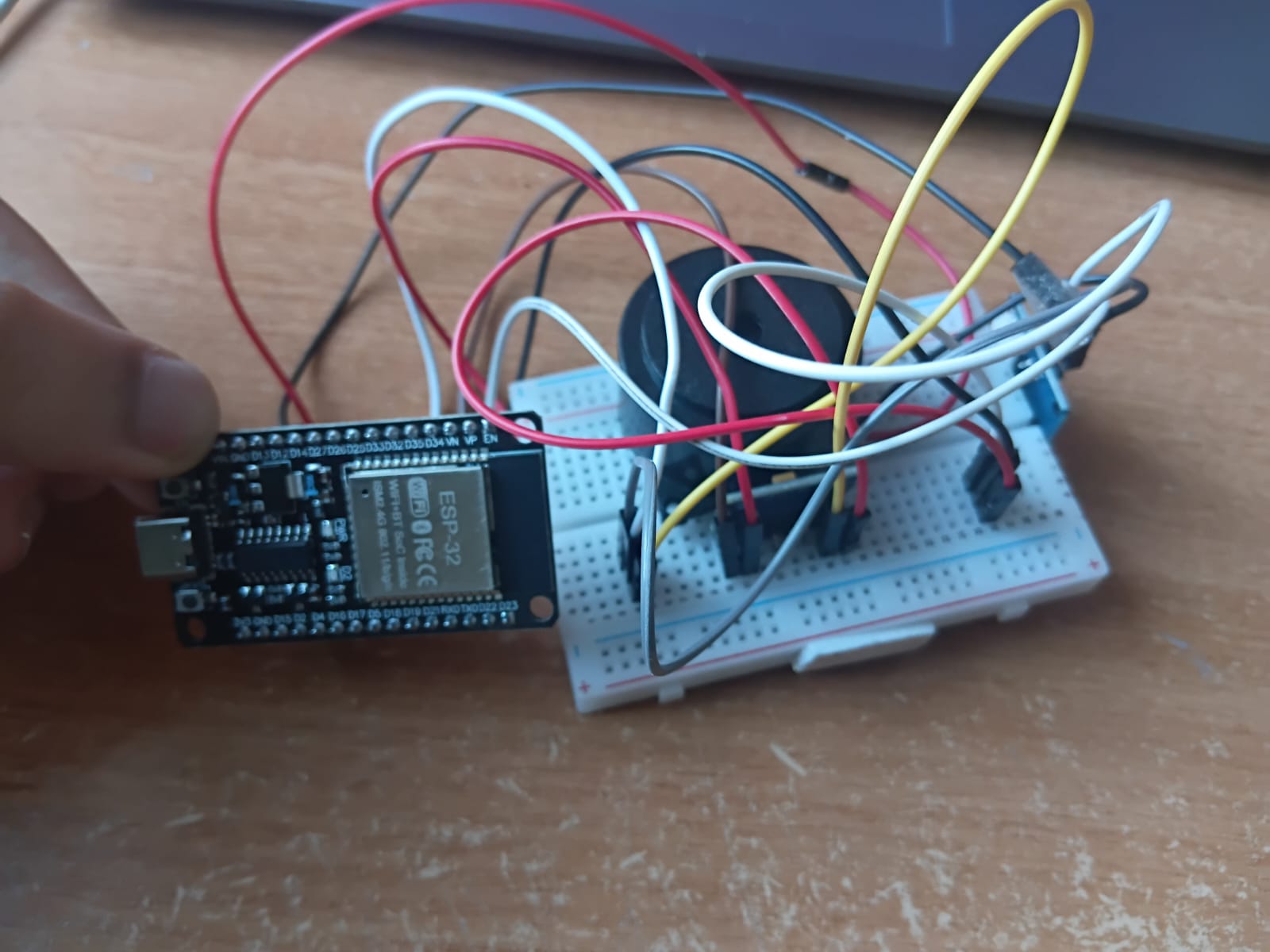
**1) Calibration Phase**

Before active monitoring, a calibration phase is conducted to measure normal stable head angle values for the neutral upright position . This establishes the baseline "safe" zone and defines the threshold values for "sleep tilt" detection.

**2) Testing Procedure**

The system was tested under three distinct conditions to validate its logic:

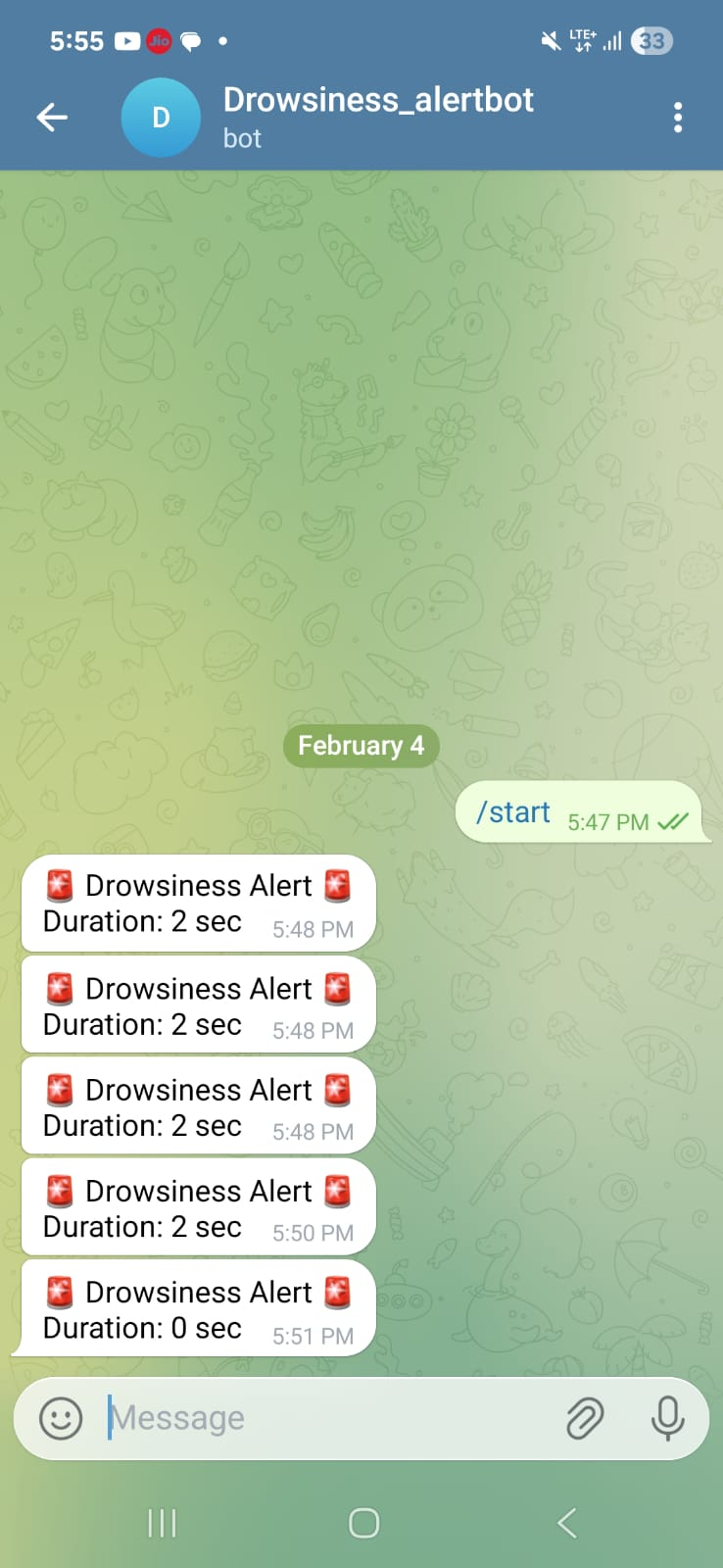
1. Normal Head Movement: The user moves their head within a vertical range (simulating checking mirrors). The expected outcome is no alarm.
2. Simulated Sleep Tilt: The user tilts their head forward beyond the threshold. The expected outcome is an instant trigger of the buzzer.
3. Touch Reset: The user taps the touch sensor while the alarm is ringing. The expected outcome is for the buzzer to stop immediately.



**CHAPTER 6:** **Results & Discussion**

**1) Performance Analysis**

* Detection Accuracy: The system successfully distinguished between normal driving movements and drowsy nodding. It correctly identified forward head drops (microsleep) and side tilts beyond safe limits.
* Response Time: The system achieved a response time of < 1 second, ensuring the buzzer activates within milliseconds of a threshold breach.
* Reliability: The manual override (touch sensor) worked consistently, allowing the driver to silence the alarm quickly, which prevents the safety system from becoming a distraction.



**2) Power and Cost**

* Power Consumption: All components operate on low voltage (3.3V/5V), making the system suitable for battery-powered operation inside a vehicle.
* Cost: The prototype is significantly cheaper than camera-based eye-tracking systems.

**3 Limitations**

* False Alarms: Sudden bumps on the road can sometimes trigger false alarms due to accelerometer sensitivity.
* Mounting: Proper mounting position on the driver’s cap is essential for accurate readings.
* Calibration: Thresholds may vary from person to person, requiring recalibration for different drivers.

SCAN THIS QR CODE TO SEE THE VIDEO



**CHAPTER 7:Conclusion**

**Conclusion**

The project successfully met its objectives by developing a wearable, sensor-based alarm system and iot messaging system. The prototype effectively detects head tilt movements associated with drowsiness and provides an immediate alert. By utilizing simple components like the ADXL345 and ESP32, the system remains cost-effective, portable, and easy to install.We have also included iot based messaging which optimises the system.

**Future Enhancements**

* Vibration Alert**:** Integrating a vibration motor for a tactile alert in noisy environments.
* Sensitivity Adjustment**:** Implementing a potentiometer to allow drivers to adjust sensitivity on the fly without reprogramming.

**REFERENCES**

[1] N. Kiruthiga, L. Anitha, and G. S. R. Devi, "Arduino based real time driver drowsiness detection," *Int. J. Eng. Comput. Sci.*, vol. 7, no. 3, pp. 23689–23693, Mar. 2018.

[2] M. S. M. Aras, A. H. Azahar, S. S. N. Alhady, and F. R. M. Rashidi, "Head movement-based driver drowsiness detection: A review," *IEEE Sensors J.*, vol. 23, no. 15, pp. 16501–16512, Aug. 2023.

[3] B. G. Lee and W. Y. Chung, "Standalone wearable driver drowsiness detection system in a smartwatch," *IEEE Sensors J.*, vol. 16, no. 15, pp. 6052–6053, Aug. 2016.

[4] A. Mittal, K. Kumar, S. Dhamija, and M. Kaur, "Head movement-based driver drowsiness detection: A review of state-of-art techniques," in *Proc. IEEE Int. Conf. Recent Adv. Eng. Comput. Sci. (RAECS)*, Chandigarh, India, 2016, pp. 1–6.

[5] "Improving road safety with real-time monitoring: A drowsiness detection system utilizing IR sensors and Arduino technology," *Int. J. Innov. Res. Manag. Policy Law (IJIRMPS)*, vol. 7, no. 1, pp. 45–52, Jan. 2024.

[6] S. Kumar, R. Singh, and V. Patel, "Real-time driver drowsiness detection system using IoT-based physiological monitoring and web interface," *J. Phys.: Conf. Ser.*, vol. 2800, no. 1, p. 012045, 2025.

**Appendices**

### Appendix A: Logic code

* #include <Arduino.h>
* #include <Wire.h>
* #include <WiFi.h>
* #include <WiFiClientSecure.h>
* #include <UniversalTelegramBot.h>
* #include <Adafruit\_ADXL345\_U.h>
* /\* ---------------- PIN DEFINITIONS ---------------- \*/
* #define TOUCH\_PIN   4
* #define BUZZER\_PIN  15
* /\* ---------------- THRESHOLDS ---------------- \*/
* #define ON\_THRESHOLD   30.0
* #define OFF\_THRESHOLD  25.0
* /\* ---------------- WIFI ---------------- \*/
* const char\* ssid = "Galaxy A16 5G 3987";
* const char\* password = "samusanni";
* /\* ---------------- TELEGRAM ---------------- \*/
* #define BOT\_TOKEN "8586133903:AAGIZZSgmgPviIhJlp1CI1W9Bm438bPIDrM"
* #define CHAT\_ID  "8116820286"
* WiFiClientSecure client;
* UniversalTelegramBot bot(BOT\_TOKEN, client);
* /\* ---------------- ACCELEROMETER ---------------- \*/
* Adafruit\_ADXL345\_Unified accel = Adafruit\_ADXL345\_Unified(12345);
* /\* ---------------- VARIABLES ---------------- \*/
* bool systemOn = false;
* bool drowsy = false;
* bool alertSent = false;
* float ax, ay, az;
* float pitch, roll;
* unsigned long drowsyStartTime = 0;
* unsigned long lastSensorRead = 0;
* unsigned long lastSerialPrint = 0;
* /\* ---------------- SETUP ---------------- \*/
* void setup() {
* Serial.begin(115200);
* pinMode(TOUCH\_PIN, INPUT);
* pinMode(BUZZER\_PIN, OUTPUT);
* Wire.begin(21, 22);
* if (!accel.begin()) {
* Serial.println("❌ ADXL345 not detected");
* while (1);
* }
* accel.setRange(ADXL345\_RANGE\_4\_G);
* WiFi.begin(ssid, password);
* Serial.print("Connecting WiFi");
* while (WiFi.status() != WL\_CONNECTED) {
* delay(500);
* Serial.print(".");
* }
* Serial.println("\n✅ WiFi Connected");
* client.setInsecure();
* }
* /\* ---------------- SENSOR READ ---------------- \*/
* void readAccelerometer() {
* sensors\_event\_t event;
* accel.getEvent(&event);
* ax = event.acceleration.x / 9.81;
* ay = event.acceleration.y / 9.81;
* az = event.acceleration.z / 9.81;
* }
* /\* ---------------- ANGLE CALCULATION ---------------- \*/
* void calculateAngles() {
* pitch = atan2(-ax, sqrt(ay \* ay + az \* az)) \* 180.0 / PI;
* roll  = atan2(ay, az) \* 180.0 / PI;
* }
* /\* ---------------- TELEGRAM ALERT ---------------- \*/
* void sendTelegramAlert(unsigned long duration) {
* String msg = "🚨 Drowsiness Alert 🚨\n";
* msg += "Duration: " + String(duration) + " sec";
* bot.sendMessage(CHAT\_ID, msg, "");
* }
* /\* ---------------- MAIN LOOP ---------------- \*/
* void loop() {
* /\* ---- TOUCH TOGGLE ---- \*/
* if (digitalRead(TOUCH\_PIN) == HIGH) {
* delay(200);
* systemOn = !systemOn;
* Serial.println(systemOn ? "🟢 System ON" : "🔴 System OFF");
* }
* if (!systemOn) {
* digitalWrite(BUZZER\_PIN, LOW);
* return;
* }
* /\* ---- SENSOR UPDATE ---- \*/
* if (millis() - lastSensorRead >= 50) {
* lastSensorRead = millis();
* readAccelerometer();
* calculateAngles();
* }
* /\* ---- DROWSINESS STATE MACHINE ---- \*/
* // TURN ON
* if (!drowsy &&
* (abs(pitch) > ON\_THRESHOLD || abs(roll) > ON\_THRESHOLD)) {
* drowsy = true;
* alertSent = false;
* drowsyStartTime = millis();
* Serial.println("⚠ Drowsiness Detected");
* }
* // TURN OFF
* if (drowsy &&
* (abs(pitch) < OFF\_THRESHOLD && abs(roll) < OFF\_THRESHOLD)) {
* drowsy = false;
* digitalWrite(BUZZER\_PIN, LOW);
* unsigned long duration =
* (millis() - drowsyStartTime) / 1000;
* if (!alertSent) {
* sendTelegramAlert(duration);
* alertSent = true;
* }
* Serial.println("✅ Drowsiness Ended");
* }
* /\* ---- BUZZER CONTROL ---- \*/
* digitalWrite(BUZZER\_PIN, drowsy ? HIGH : LOW);
* /\* ---- SERIAL PRINT (50ms sec) ---- \*/
* if (millis() - lastSerialPrint >= 50) {
* lastSerialPrint = millis();
* Serial.print("Pitch: ");
* Serial.print(pitch, 1);
* Serial.print(" | Roll: ");
* Serial.println(roll, 1);
* }
* }

### Appendix B: Component List

### ESP32 (Microcontroller) -330rupees

### ADXL345 (Accelerometer)-320rupees

### Capacitive Touch Sensor (Input) -40rupeees

### Active Buzzer (Output) -40rupees

### Breadboard and Jumper Wires-105rupees