Driver Fatigue System

ECE445 Design Document - Spring 2025

Project # 76
Julio Cornejo
Vincent Ng

Professor: Arne Fliflet
TA: Maanas Agrawal

Contents

1. Introduction

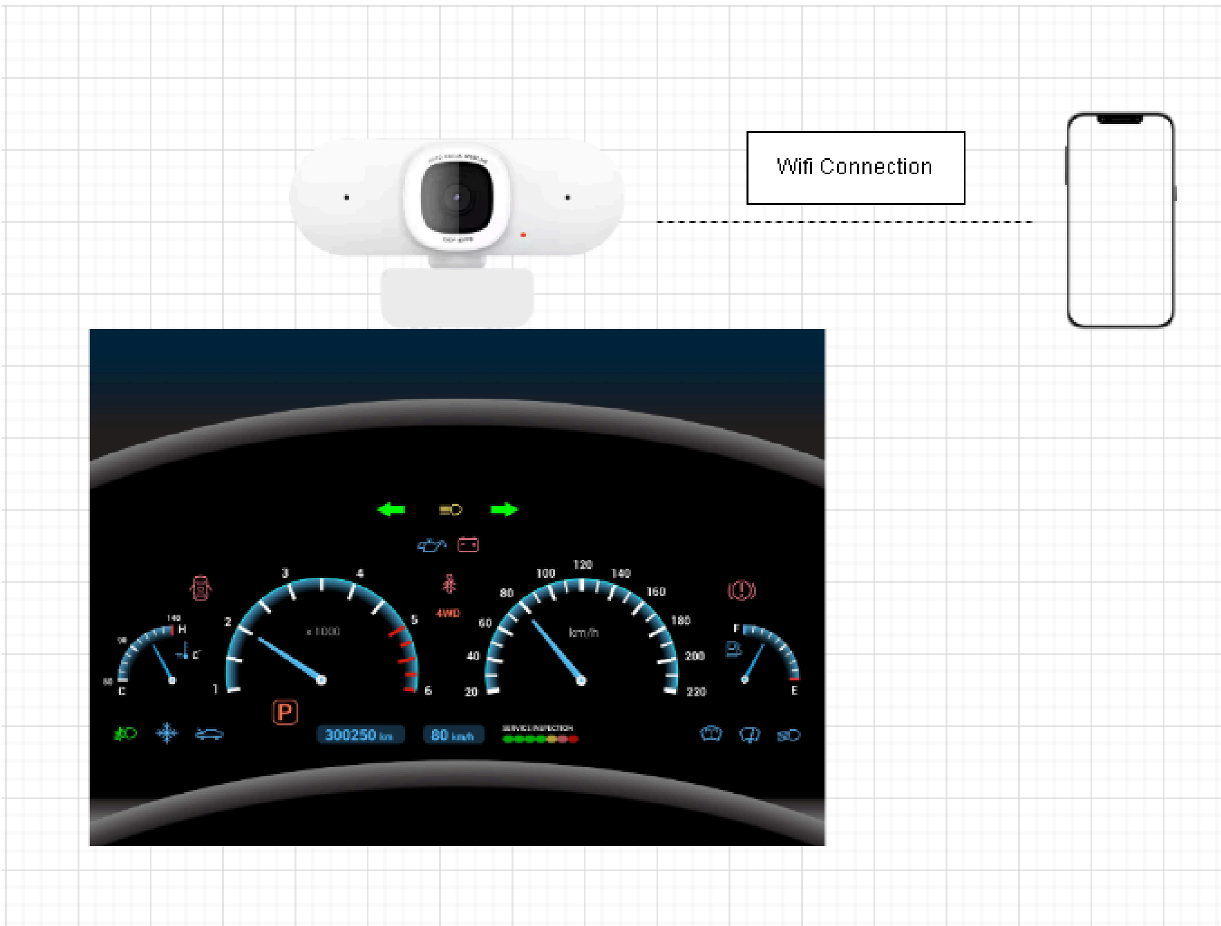
Problem:

When driving for prolonged amounts of time, some key body movements and facial changes can be made due to drowsiness. The drowsiness, if unmonitored, can pose dangerous conditions for other drivers and the drivers themselves. Intoxication while driving is also a rampant issue; there is no universal breathalyzer that prohibits driving based on BAC. I propose this device that uses facial recognition, eye-level detection sensors, and cameras to detect symptoms of fatigue along the road while also prohibiting intoxicated drivers from proceeding to drive. This device can monitor head position, yawns, and register long blinks. It can also track the driving duration and eventually register all these symptoms if it detects fatigue. Once enough triggers are set, an app interface can assess your tiredness or driving incapability via WiFi transmission. It can suggest and locate the nearest rest stop or call emergency contacts (set by the user). When specific drowsiness scores are reached, the user's BAC and live drowsiness rating will be displayed with in-house buzzer systems.

Solution:

The system revolves around an algorithm that makes use of a variety of sensors and cameras. One is a breathalyzer that measures the blood alcohol concentration of the driver. Using the software, we monitor the live value and set triggers to call emergency contacts and monitor until a safe concentration for resumable driving. Drowsiness and tiredness can be detected in various ways ranging from yawn frequency, long blinks, and head tilts. They happen suddenly. Most traditional cars use your position in a lane to track tiredness. Tracking head movement and analyzing the face for more key indicators precisely provides more information that can be essential to identify when a driver needs to step away from the wheel and request a ride elsewhere. The PCB can be housed in a small, compact shape like a cube that sits over any dashboard with detachment features to trigger the BAC sensor correctly.

Visual Aid:



S

Figure 1: Simple Device Overview

High-Level Requirements:

- Driver Inactivity Detection Accuracy: The system must detect eye closure and blink duration with at least 90% accuracy, using an EAR threshold of 0.2 for detecting closed eyes and registering blinks that last ≥15 frames as prolonged closures. The false positive rate for open eyes must remain below 5%.

- BAC Measurement Precision: The device must measure Blood Alcohol Concentration (BAC) with an accuracy of ±1% BAC.

- User Interface Accessibility: The system must provide a mobile/web-based interface that updates drowsiness and BAC data in real-time (latency ≤1 second) and helps users contact emergency contacts within 5 seconds when the drowsiness score surpasses a certain threshold.

2. Design

Physical Design:

The physical design of our device will be similar to that of a driver facing dash cam. The housing of our design will be a rectangular box made out of plastic, with cutouts for the camera module, display, led, and breathalyzer tube. There will also be a mounting mechanism for the device so that it can sit securely on the drivers dashboard. In the current time frame of the project, we are not yet at the stage of having a prototype built, but once we do, an image will be attached here for reference.

Block Diagram:

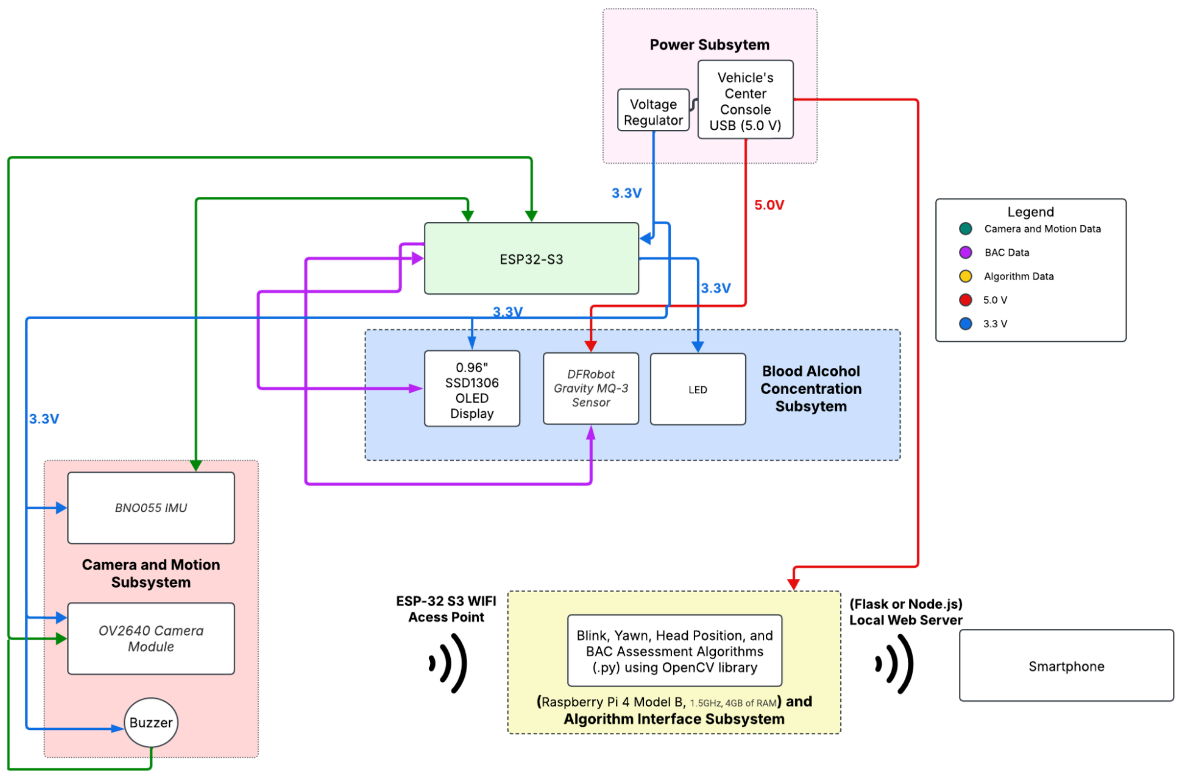


Figure 2: Diagram of How Each Subsystem is Connected

Subsystem Overview:

Subsystem 1: Blood Alcohol Concentration System
Overview:
The functionality of this subsystem revolves around the MQ-3 breathalyzer sensor responsible for measuring the BAC of a potential driver. Accompanying the sensor is logistical device tools such as a display and LED. This module is connected to the ESP32 microcontroller and communicates directly with our designed algorithm for an automated emergency notification system if intoxication is detected. The BAC will also be locally displayed on the unit's OLED screen. A light will light blue when the device is ready to test an individual's BAC.

Requirements:

- The device must measure Blood Alcohol Concentration (BAC) with an accuracy of ±0.01% BAC, within a detection range of 0.00% - 0.20% BAC.

- The device must ensure the baud rate transmission is 9600 for the MQ-3's functionality.

- The LED accurately reflects the readiness and warm-up time of the MQ-3 to provide accurate measurements.

Requirements and Verification Table:

Requirements	Verification
● The device must measure BAC with an accuracy of ±0.01% BAC within a detection range of 0.00% - 0.20% BAC.	● Perform calibration tests using known alcohol concentrations and compare readings with a certified breathalyzer. ● Cross-checking each of the individual MQ-3 sensors since we ordered a few for faultiness.
● The device must maintain a baud rate of 9600 for MQ-3 communication.	● Monitor serial transmission using an oscilloscope to verify correct baud rate operation.
● The LED should accurately reflect the readiness and warm-up time of the MQ-3.	● Measure time from power-on to ready-state using a stopwatch and validate against the MQ-3 datasheet specifications.

Subsystem 2: Power System
Overview:
Our primary power source for our power subsystem comes directly from a car's center console. Most cars have USB ports that provide 5V of power. Coupled with a voltage regulator to step down to 3.3V, we can power our ESP32 and sensor/camera modules. The MQ-3 is powered directly by the 5V source.

Requirements:

- Maintain overall voltage stability within a tolerance of 0.1V (+/-).
- Successful step-down conversion to 3.3V for smooth operation of the BNO055 IMU (Head position sensor), OV2640 Camera Module, SSD1306 OLED Display, and LED.
- Must provide consistent and stable 5V to operate the MQ-3's critical sensor.

Requirements and Verification Table:

Requirements	Verification
• Maintain overall voltage stability within a tolerance of ±0.1V.	• Measure voltage fluctuations using a digital multimeter (DMM) under different load conditions.
• Ensure successful step-down conversion to 3.3V for all necessary components.	• Use a DMM to verify stable 3.3V output under normal operating conditions.
• Provide a consistent and stable 5V output for MQ-3 operation.	• Test voltage stability under varying load conditions and monitor power fluctuations using an oscilloscope.

Subsystem 3: Camera and Motion System
Overview:
This subsystem is the most critical component for sensing fatigue. Both sensors, BNO055 IMU for head movement and OV2640 Camera Module for eye blink and yawn detection, are essential inputs for our computer vision algorithm; both sensors are powered by a 3.3V power source and interact and transmit data directly to the ESP32. A buzzer is provided for temporary alertness in emergency-level drowsiness.

Requirements:

- Capable of capturing images with sufficient clarity for detecting blinks and yawns.
- To ensure real-time processing, the camera should interface with the ESP32 and quickly transmit image data via WiFi to the Raspberry Pi.
- The camera should support JPEG image compression for efficient data transfer.
- Minimum image resolution: 640x480 pixels and frame requirement of 30fps.
- Non-intrusive (low dB) buzzer and alert-focused sound.

Requirements and Verification Table:

Requirements	Verification
● Capture images with sufficient clarity to detect blinks and yawns.	● Evaluate image resolution and sharpness using controlled lighting conditions. ● Sample test of an individual blinking X amount of times in a tets period 60 sec. Compare results with algorithm testing.
● Camera must interface with ESP32 and transmit data via WiFi with minimal latency.	● Measure transmission delay using a network packet analyzer. ● Compare the speed and algorithm running process with I2C connection before testing with WiFi.
● Support JPEG image compression for efficient data transfer.	● Verify image file sizes and compression ratios using software analysis.

• Maintain a minimum resolution of 640x480 pixels and 30fps frame rate.	• Capture and analyze sample frames to confirm compliance with the required resolution and frame rate.
• Buzzer should emit a non-intrusive alert sound.	• Measure buzzer sound levels in decibels (dB) using a sound meter. Place the nozzle of the sound meter above the encaving of the active buzzer. • Most active buzzers can be lowered by attaching a resistor using the correct amount of resistance. Validate with a 1k, 10k, etc. for the optimal db count using a sound meter.

Subsystem 4: Algorithm and User Interface Subsystem

Overview:

This subsystem will handle the processing of all sensor data, run the necessary algorithms to assess drowsiness levels based on eye blink frequency, head movements, and breathalyzer readings, and display the results on a web app accessible from a smartphone. It will communicate wirelessly with the ESP32 microcontroller via WiFi using a client-server access point connected to the Raspberry Pi to receive real-time sensor and camera data. The algorithm will analyze the data, calculate the drowsiness level, and send this information to the web app via another access point, a web-server host using Flask or, potentially, Node.js as our framework. Using WiFi once more, we can display and manipulate all the data for analytics on the smartphone. The web app will provide a real-time drowsiness score to the user and display analytics.

Requirements:

- Using an efficient algorithm, the system must calculate a fair drowsiness score from real-time data (eye blinks, head movement, BAC levels), providing instant feedback on driver fatigue.

- The ESP32 should transmit drowsiness data via WiFi with a minimum baud rate of 9600 bps.

- The web app should provide real-time feedback on the driver's drowsiness score, display analytics (e.g., blink frequency, head movement), and send alerts when fatigue levels are critical.

- The system must log drowsiness data for historical analysis and send push notifications or emergency contact alerts when the drowsiness score exceeds a defined threshold. [Pi Link]

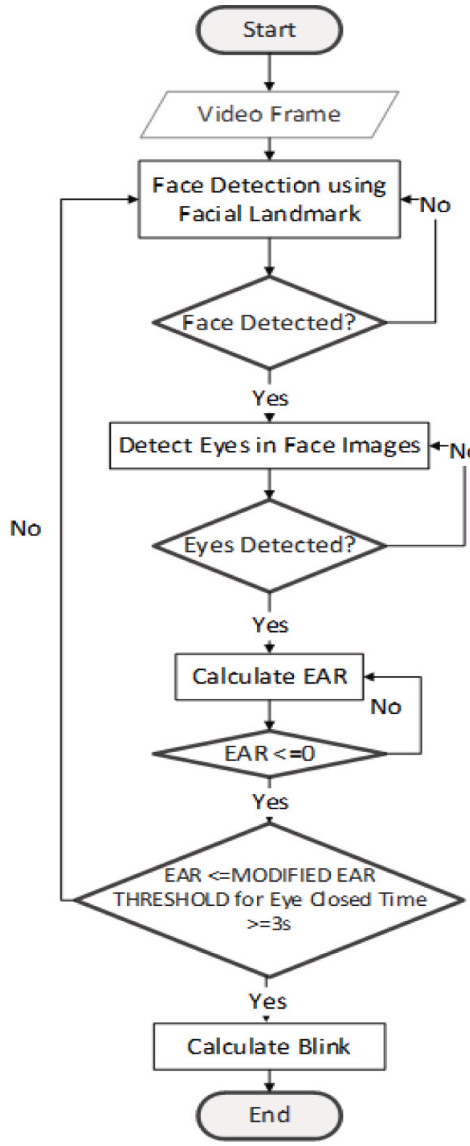


Figure 3: Flow Chart of EAR algorithm

Requirements and Verification Table:

Requirements	Verification
• Calculate a real-time drowsiness score based on sensor inputs.	• Validate computed scores against manually analyzed fatigue behavior datasets. • For test subjects, we can have a self-reported driving quiz where the user assesses a level where they feel incapable
• ESP32 must transmit drowsiness data via WiFi with a baud rate of at least 9600 bps.	• Monitor WiFi transmission rates using a network packet analyzer.
• Web app should provide real-time feedback on drowsiness score and analytics.	• Conduct usability tests to ensure data is displayed with minimal delay. • A user should be able to query into our database to see the times a certain score of their choosing is submitted during their driving times.
• System should log drowsiness data and send emergency alerts when a defined threshold is exceeded.	• Simulate drowsiness conditions and verify proper alert generation and logging. • Use Twilio API and simulate a call to ourselves via

Tolerance Analysis:
One of this project's most critical and challenging components is the accurate detection of eye closure and blink duration using EAR. The system must reliably differentiate between normal blinking and drowsiness-induced prolonged eye closures while accounting for environmental factors such as low lighting, camera angle variations, and false detections. Reliable measurements are crucial as not to falsely alert drivers when driving on the road.
The Eye Aspect Ratio formula is given below:

$$EAR = \frac{\left(\left|P_2 - P_6\right| + \left|P_3 - P_5\right|\right)}{2\left|P_1 - P_4\right|}$$

P1-P6 are the eye landmarks detected by the OV2640 camera module

We can set the EAR threshold to:

Eyes open: 0.2-0.3

Eyes partially closed: 0.15-0.2

Eyes closed: < 0.15

However, these thresholds are not set in place and can be adjusted once the algorithm testing begins so that we can implement a more refined threshold.
Given our EAR thresholds, we can also register blinks with these conditions:
EAR drops below 0.2 for three consecutive frames (assuming a 30 FPS camera feed).
If EAR remains below 0.2 for more than 15 frames, it is flagged as a sign of drowsiness.

Tolerance Analysis Considerations:
Component Accuracy and Environmental Factors:

Camera Resolution and Precision:
The OV2640 camera module has a resolution of 640x480 pixels at 30 FPS, but low-light conditions may affect its accuracy.
The sensor has an approximate accuracy variance of ±5% in detecting key facial landmarks due to environmental lighting and head orientation.

Measurement Noise and Latency:
EAR calculations rely on landmark detection accuracy, which may fluctuate due to camera latency (typically 10-20 ms per frame under normal conditions).
Noise filtering techniques (e.g., median filtering) will be used to smooth EAR readings to reduce variability.

EAR Threshold Tolerance:
A deviation of ±0.02 EAR from the expected threshold could lead to false positives or missed detections.
A key risk is misclassifying normal long blinks or head tilts as drowsiness. Therefore, we can introduce an adaptive threshold based on the driver's historical blink patterns, adjusting the EAR threshold dynamically.

Weighted Drowsiness Scoring Model:
This is our scoring model that is designed to improve drowsiness detection accuracy by incorporating multiple fatigue indicators. Instead of relying solely on blink duration or frequency, this model assigns different weight values to various fatigue-related behaviors, such as yawning and head tilting, based on their severity and impact on alertness. By aggregating these factors into a single score, the system can more accurately differentiate between temporary fatigue and actual drowsiness. This approach minimizes false positives while ensuring that prolonged drowsiness patterns trigger necessary alerts, making the system more reliable for real-world driving conditions. Whenever a certain drowsiness score threshold is reached, the system will contact the driver's emergency contact.

The drowsiness score is calculated using a weighted formula shown below:
Assume we monitor over a **60-second window** and detect the following:

- **Blink Duration**: 6 long blinks → **High risk**
- **Blink Frequency**: 20 blinks → **Moderate risk**
- **Yawns Detected**: 4 yawns → **High risk**
- **Head Tilt Angle**: 30° tilt for 5+ sec → **Moderate risk**
- **Head Nods**: 2 detected → **Low risk**

Now, applying the weights that are currency variable and are going to be changing :

Drowsiness Score = $(6\times4.2) + (20\times1.0) + (4\times7.5) + (1\times10) + (2\times5) = (6\times4.2) + (20\times1.0) + (4\times7.5) + (1\times10) + (2\times5) = 25.2+20+30+10+10=95.2= 25.2 + 20 + 30 + 10 + 10 = 95.2=25.2+20+30+10+10 =$ **95.2**

3. Cost

Cost Analysis:

Note that we are considering the cost of the items with the retail price where we acquired the part, excluding ECE discounts.

Part	Manufacturer	Quantity	Price	Subsystem
Raspberry Pi 4 Model B 2GB	Raspberry Pi Ltd	1	$53.99	Algorithm and Interface
Micro-HDMI Male to HDMI Female Adapter	Microware	1	$6.16	Algorithm and Interface
32GB MicroSD Card	Samsung	1	$13.05	Algorithm and Interface
ESP32-S3 Development Board	HiLetgo	1	$16.53	Algorithm and Interface
OV2640 Camera Module	STMicroelectronics	1	$11.99	Camera and Motion
Active Piezo Buzzer Alarm	GFORTUN	1	$0.89	Camera and Motion
MQ3 Alcohol Sensor Module	Reland Sun	1	$2.67	BAC Concentration
0.96" SSD1306 OLED LCD	HiLetgo	1	$6.99	BAC Concentration
LED	HiLetgo	1	$0.89	BAC Concentration
LD117 Voltage Regulator	STMicroelectronics	4	$0.80	Power Subsystem
Total			$116.36	

Our project has a heavy computation component relying on algorithm creation and data transfer via WiFi. For this reason, many of the estimated labor hours involve client-server setup, so the ESP-32 and the Pi can efficiently and wirelessly communicate. The algorithms will require extensive image processing, so searching and modifying a functional algorithm is one of the largest time-sinks. Assuming an hourly-paid entry-level engineer position, we take an hourly pay of $45. With a rough estimate of 310 hours based on the schedule and planning, our costs come out to:

Total Cost: Labor Cost + Part Cost = ((310 hours) x ($45)) + $116.36 = $14066.36

4. Schedule

Week	Task	Hours	Engineer
2/24	Designing PCB in KiCad. Seek PCB Review + Approval	15-20	Julio and Vincent
3/3	Connecting MQ-3, camera, and sensors. Implementing I2C and WiFi communication. Running Test Python algorithm on the Pi Data transmission via WiFi.	30-40	Julio
3/10 3/17	Setting up OpenCV and TensorFlow. Writing Program in Python. PCB Review and Pass Audit Training and testing blink/yawn detection.	50-60	Julio
3/24	Assemble and finalize casing design via CAD and 3D print the result.	35-45	Vincent
3/31	Solder components onto the PCB. Ensuring all components work together. Work on Performance optimization	35-45	Vincent
4/7 4/14	Web App Development Designing the UI for real-time data. Setting up Flask/Django server.	40-50	Julio and Vincent
Total		**310**	

5. Ethics and Safety

Our project's main ethical and safety concerns are privacy and data security.
The system collects sensitive biometric data, including facial recognition patterns, eye movement, and breathalyzer readings. According to the ACM Code of Ethics (Principle 1.6: Respect Privacy), developers must ensure that user data is stored securely and only accessed for its intended purpose. To mitigate potential privacy concerns:
Data will be processed locally on the ESP32-S3 or Raspberry Pi, avoiding unnecessary cloud storage. Encrypted communication protocols (e.g., HTTPS, TLS) will transmit data to the mobile application.

Another ethical concern highlighted by IEEE is Principle 2: Avoid Harm to Others, emphasizing that systems must be designed to avoid discrimination or bias based on race, gender, or disability. We will train our model on a diverse dataset to ensure fairness, including individuals of different ethnicities, backgrounds, and facial features. Not only that, but the EAR threshold will also be adjustable for individuals, allowing for an individualized experience.

Since our device also has a miniature built-in display, we have decided to keep it as minimal as possible to uphold safety guidelines and maintain a safe driving experience. Alerts, buzzer warnings, and the user interface must be designed to minimize distractions.
According to NHTSA guidelines, visual displays should be simple and non-intrusive.
Alerts should not require extended interaction from the driver while operating the vehicle.
The buzzer should produce audible but non-alarming signals to avoid panic responses.

6. Citations and References

Dewi, C., Chen, R.-C., Jiang, X., & Yu, H. (2022, April 18). Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. PeerJ. Computer science. https://pmc.ncbi.nlm.nih.gov/articles/PMC9044337/

Pandey, D. (2021, April 21). Eye aspect ratio(ear) and drowsiness detector using dlib. Medium. https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706

Raspberry pi documentation - getting started. Raspberry pi documentation. (n.d.). https://www.raspberrypi.com/documentation/computers/getting-started.html

Shekari Soleimanloo, S., Wilkinson, V. E., Cori, J. M., Westlake, J., Stevens, B., Downey, L. A., Shiferaw, B. A., Rajaratnam, S. M. W., & Howard, M. E. (2019, September 15). Eye-blink parameters detect on-road track-driving impairment following severe sleep deprivation. Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine. https://pmc.ncbi.nlm.nih.gov/articles/PMC6760410/

Ucl. (2022, May 6). Blink and you miss it!. UCL News. https://www.ucl.ac.uk/news/2005/jul/blink-and-you-miss-it