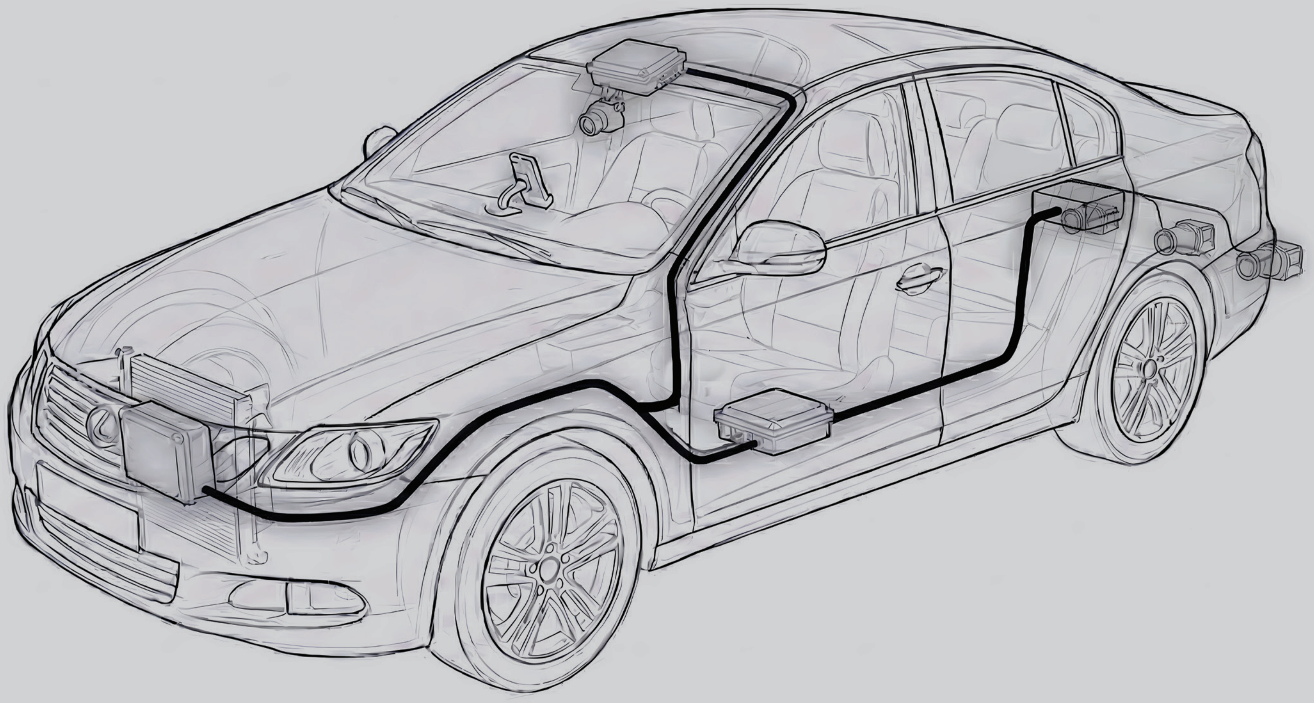


Drivora

Universal Advanced Driver Assistance System

Design Manual

Version 1.0

© 2026 Drivora Project Team

Table of Contents

---

1. Introduction

Drivora is a vehicle-mounted advanced driver assistance system designed to improve driver awareness through sensor fusion, local processing, and a mobile dashboard interface. The system combines ultrasonic sensing, inertial sensing, and camera-based lane monitoring to detect front collision risk, rear blind-spot risk, lane departure, and vehicle instability.

This document describes the technical design of the system. It is intended for engineers, evaluators, and maintainers who need a concise view of the architecture, major subsystems, communication methods, and implementation decisions.

1.1 Design Objectives

The design was developed with the following objectives:

- provide real-time hazard warnings with low driver distraction,
- support multiple vehicle types through configurable geometry and calibration,
- operate locally without requiring constant internet connectivity,
- synchronize important data to the cloud when available,
- keep the driver interface simple and glanceable,
- separate sensing, processing, and user interface responsibilities across subsystems.

1.2 Scope

This manual covers the current Drivora implementation, including the vehicle-edge hardware units, central processing unit, communication pathways, mobile application, warning logic, and diagnostic behavior. It does not describe user-level operation in detail; that information belongs in the user manual.

1.3 System Context

The current design uses a Raspberry Pi-based brain unit as the central processor, ESP32-C3-based sensing units for front, rear, and center/lean measurements, a Raspberry Pi Camera Module for lane monitoring, a local Wi-Fi hotspot for app communication, and Firebase for account and incident data handling.

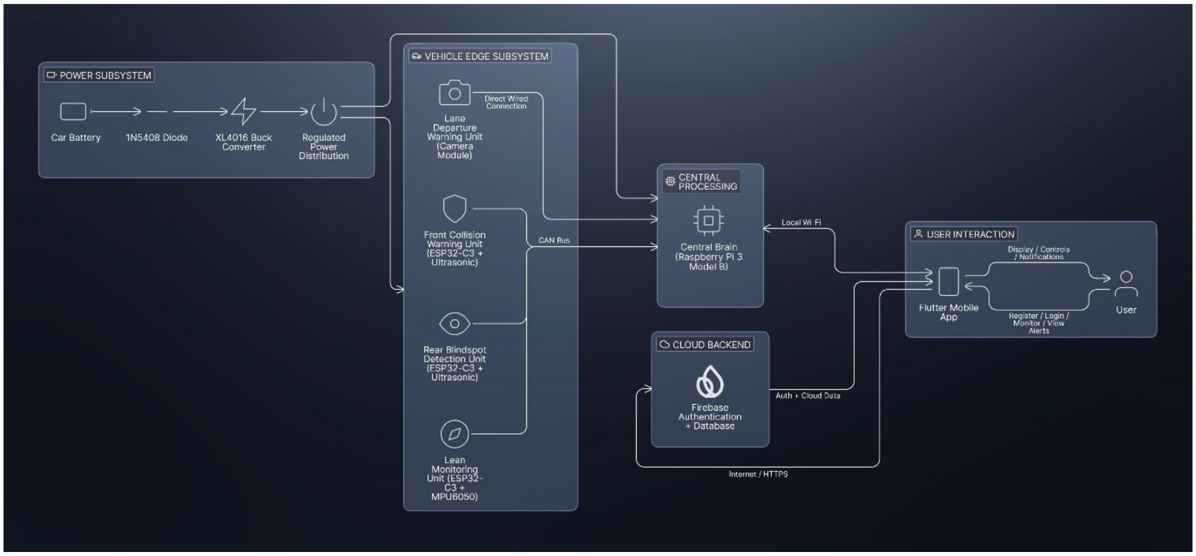


Figure 1: High-level architecture diagram of the full Drivora system.

---

2. System Overview

Drivora follows a distributed architecture. Vehicle-mounted sensor units gather raw data, the brain unit fuses and classifies warnings, and the Flutter application displays the processed result to the user.

2.1 Main Subsystems

Subsystem	Role
Vehicle-edge subsystem	Front, rear, center/lean, and camera/lane sensing on the vehicle
Central processing subsystem	Raspberry Pi brain unit, CAN bridge, warning fusion, buzzer control
Mobile application subsystem	Dashboard, statistics, settings, and setup workflow
Cloud backend subsystem	Firebase authentication, database storage, and sync support
Power subsystem	Vehicle 12 V input, protection, and regulated 5 V distribution

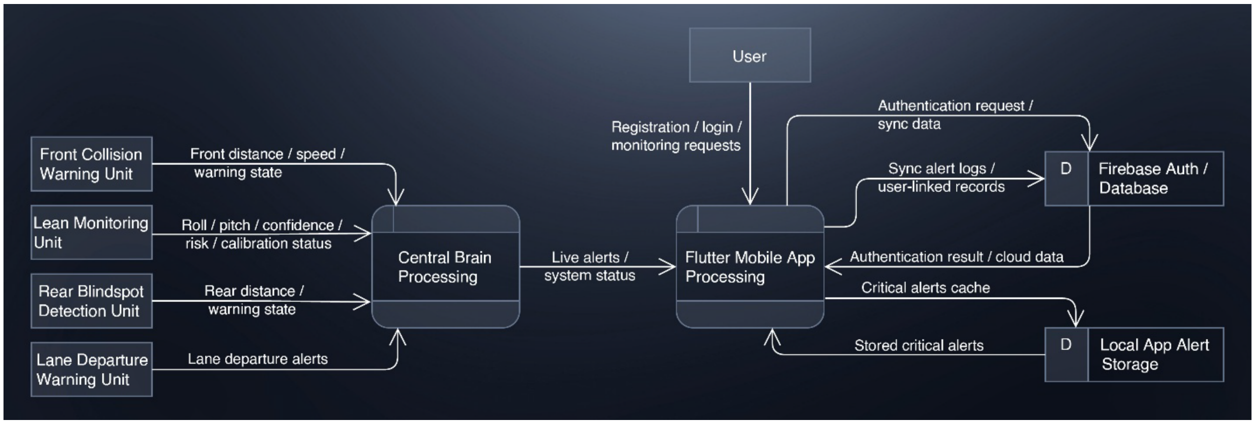


Figure 2: Dataflow Diagram

2.2 System Data Flow

The system flow is as follows:

1. Sensor units collect raw measurements from their local sensing hardware and then process them using their dedicated ESPs.
2. Processed data is transmitted to the central brain through the communication layer.
3. The brain filters the incoming data and applies warning fusion logic.
4. The brain controls the buzzer and provides status data to the app.
5. The mobile app presents the current vehicle state, warning conditions, and recent history.
6. Selected account and incident data are synchronized with Firebase when connectivity is available.

2.3 High-Level Design Choice

The main design choice is to keep safety decision logic in the brain unit rather than in the mobile app. This ensures that warning generation continues even if the app is closed, delayed, or disconnected. The app is therefore a presentation and configuration layer, while the brain remains the decision layer.

---

3. Hardware Design

The hardware design is divided into five functional units: front sensing, rear sensing, center/lean sensing, camera/lane sensing, and the brain unit.

3.1 Front Unit

The front unit detects forward obstacle risk using ultrasonic sensing hardware. It is mounted near the front grille or bumper region, depending on vehicle geometry and available mounting space.

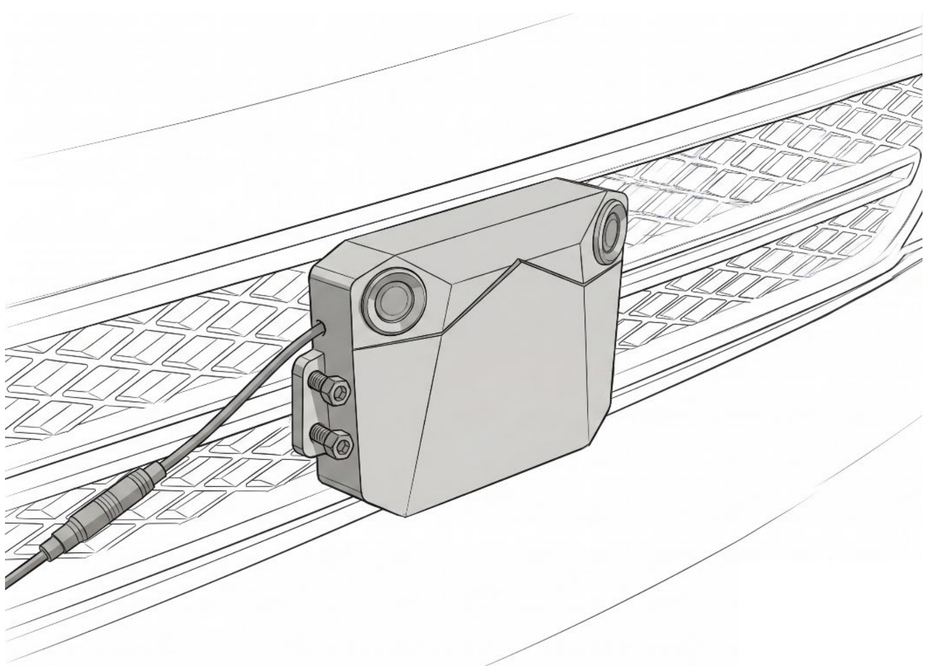


Figure 3: Front unit placement on the vehicle

The front unit is designed to report distance and warning state to the brain with minimal latency. The unit enclosure should protect the electronics from water splash, dust, and vibration.

Figure 4: Front Unit Final Assembly

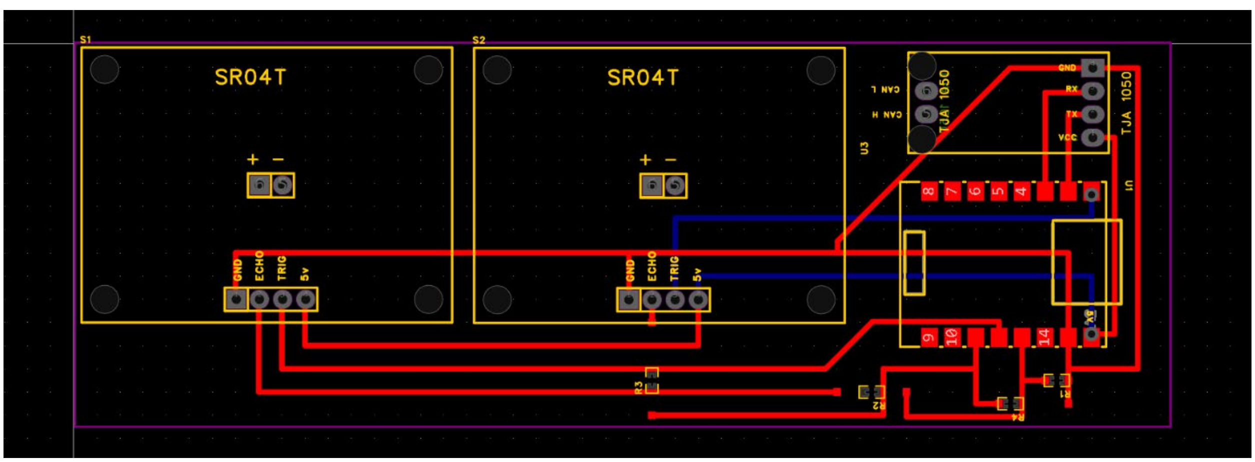


Figure 5: Front Unit PCB Layout

Key Electronic Components

Component	Function
ESP32-C3 SuperMini	Local controller and CAN communication
2x JSN-SR04T Ultrasonic Sensor Modules	Front obstacle distance measurement
TJA1050 CAN Transceiver	Physical CAN bus interface
Waterproof Connector	Vehicle wiring interface
Few Resistors	Voltage dividers

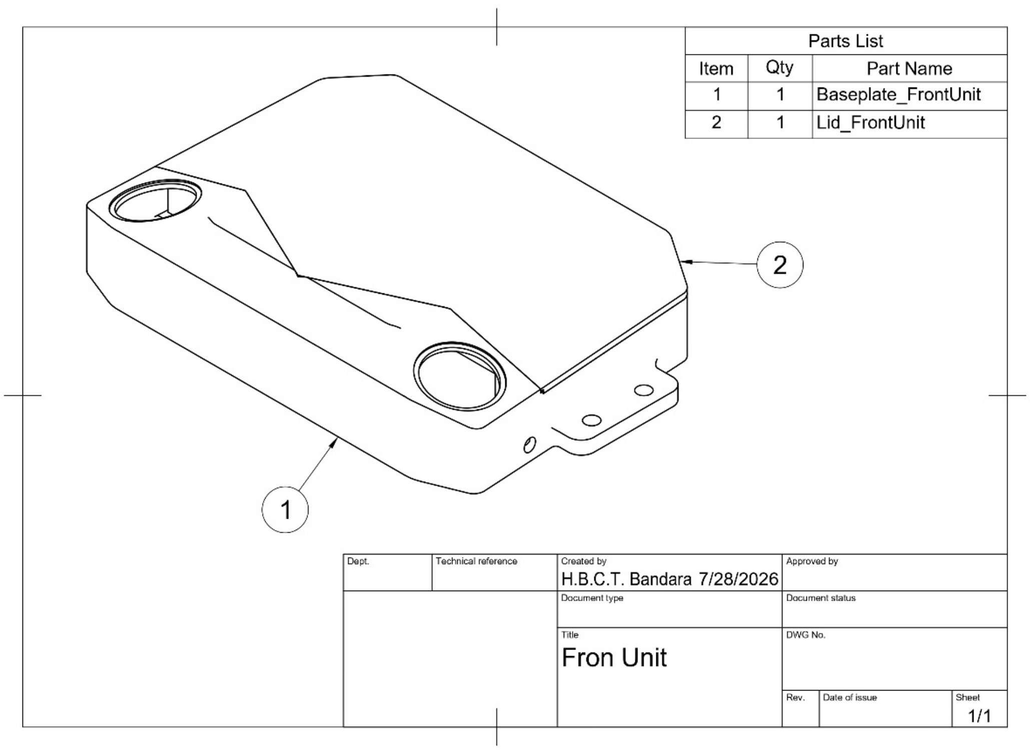


Figure 6: Front Unit Enclosure - Isometric View

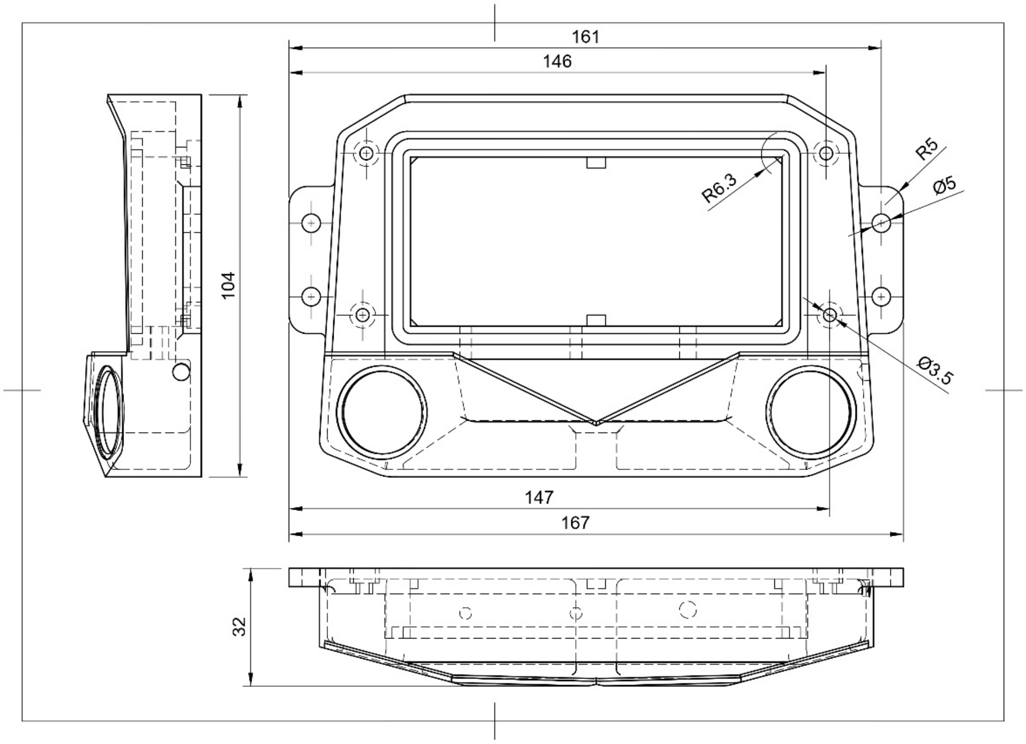


Figure 7: Front Unit Base Plate - Orthographic Views

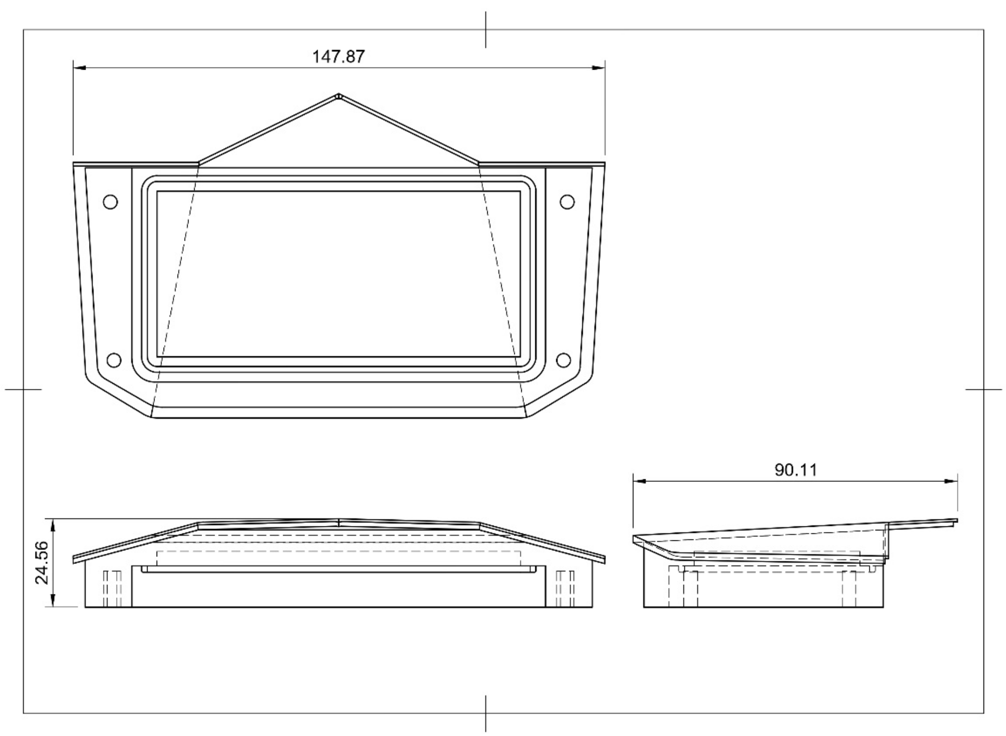


Figure 8: Front Unit Lid - Orthographic Views

3.2 Rear Unit

The rear subsystem uses three ultrasonic probes to monitor rear-left, rear-center, and rear-right blind-spot regions. The rear hub is placed inside the rear cargo area, while the probes are mounted through the rear bumper.

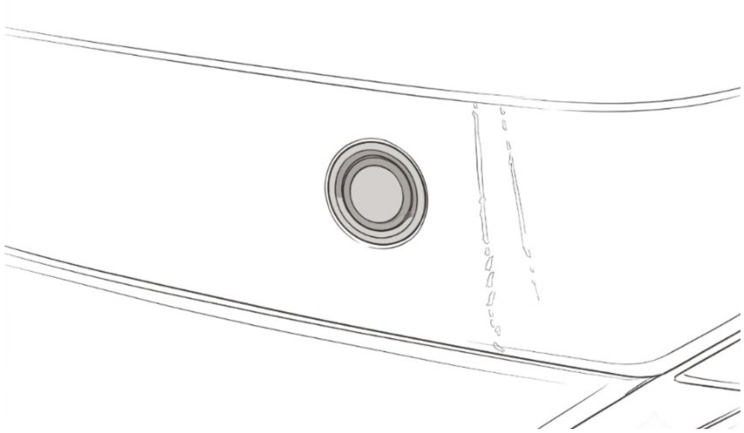


Figure 9: Rear sensor probe placement on the bumper

The rear layout is intended to preserve spatial separation between the three detection zones while keeping the wiring short and mechanically protected.

Figure 10: Rear Unit Final Assembly

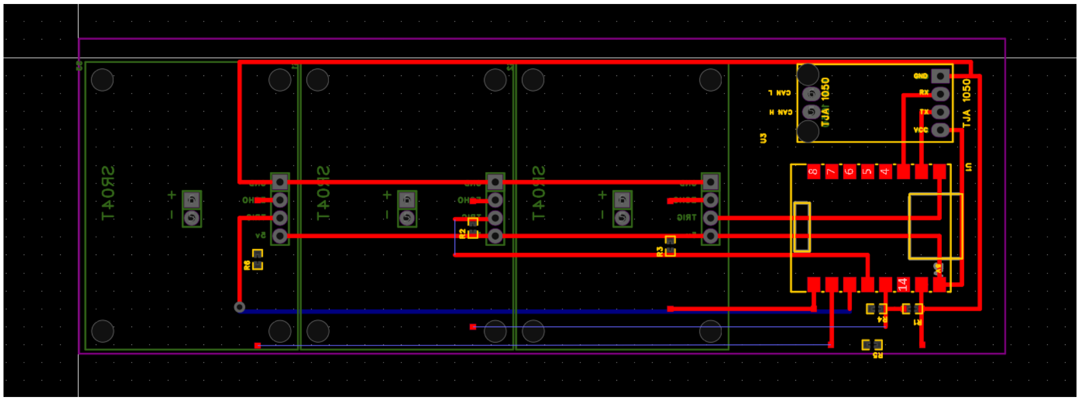


Figure 11: Rear Unit PCB Layout

Key Electronic Components

Component	Function
ESP32-C3 SuperMini	Local controller and CAN communication
3x JSN-SR04T Ultrasonic Sensor Modules	Rear obstacle distance measurement
TJA1050 CAN Transceiver	Physical CAN bus interface
Waterproof Connector	Vehicle wiring interface
Few Resistors	Voltage dividers

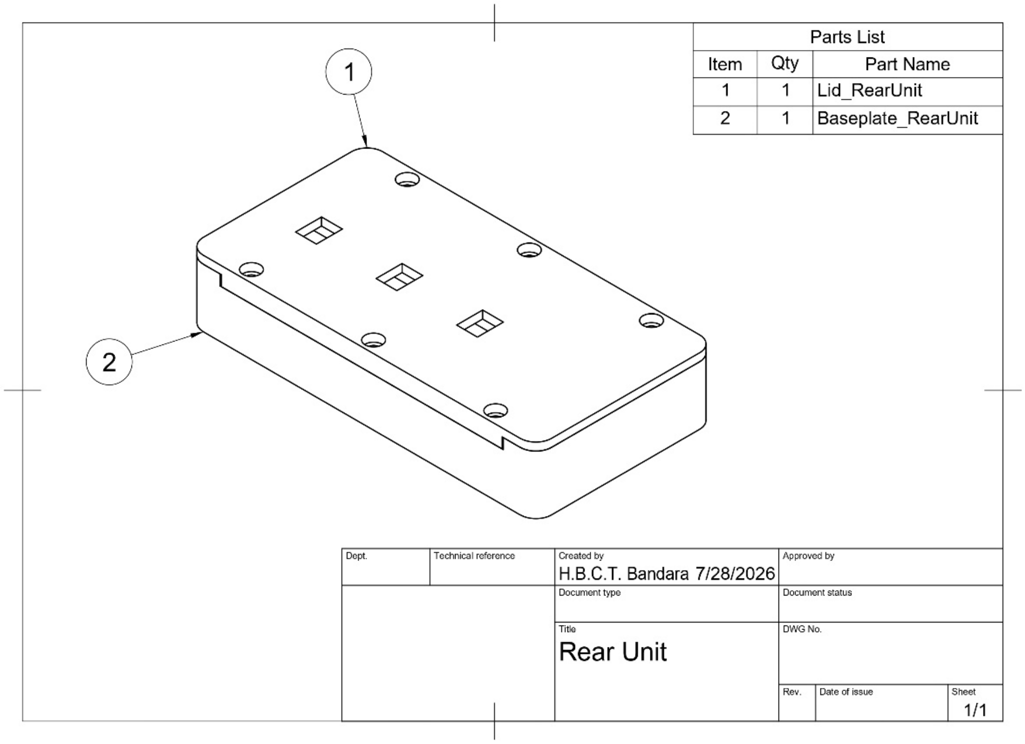


Figure 12: Rear Unit Enclosure - Isometric View

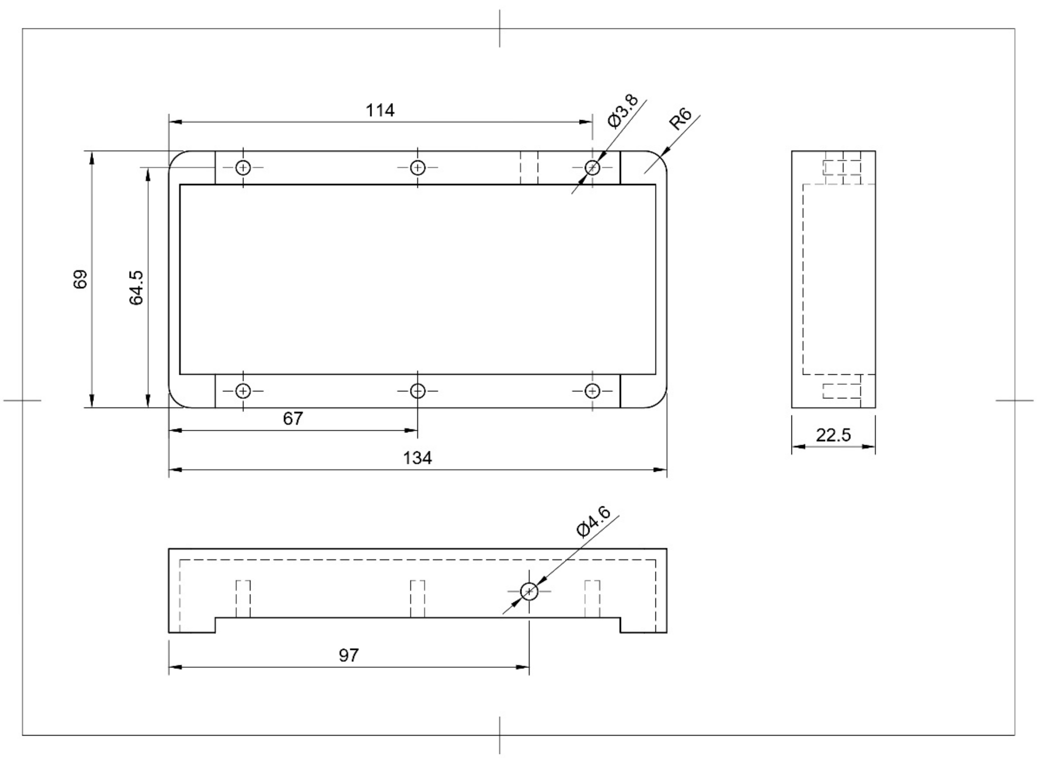


Figure 13: Rear Unit Base Plate - Orthographic Views

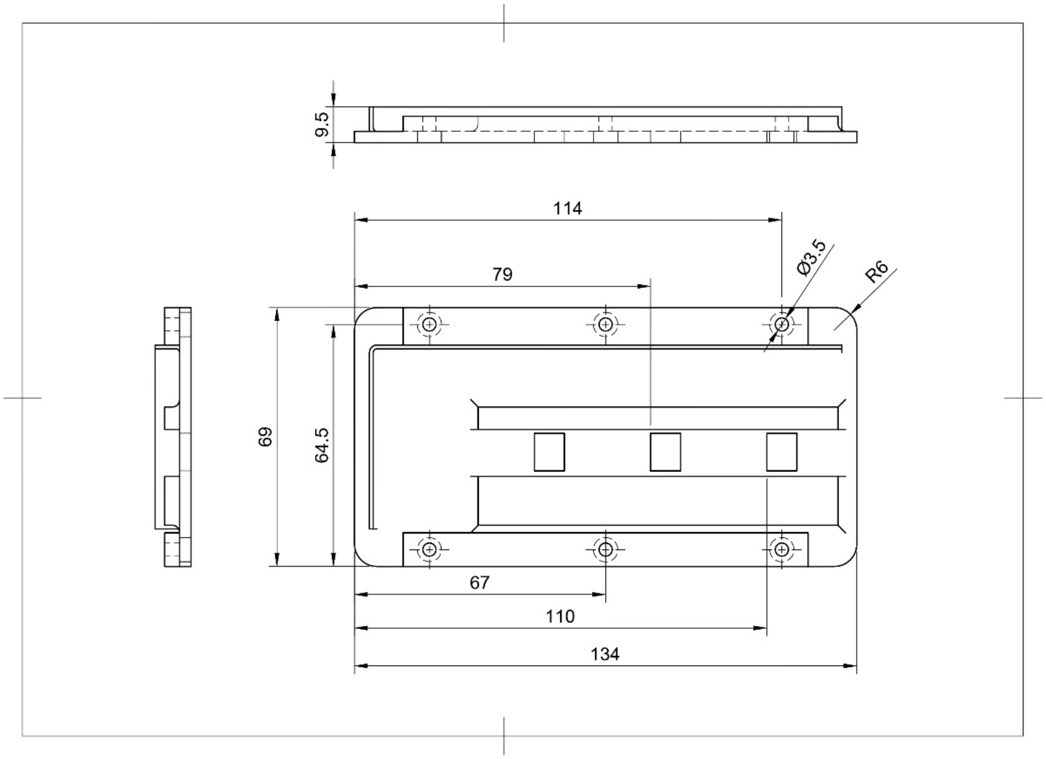


Figure 14: Rear Unit Lid - Orthographic Views

3.3 Center / Lean Unit

The center/lean unit uses an IMU to estimate roll and pitch changes. This information is used to detect unusual vehicle body movement or stability risk after calibration. The unit must be mounted on a rigid surface with the correct orientation.

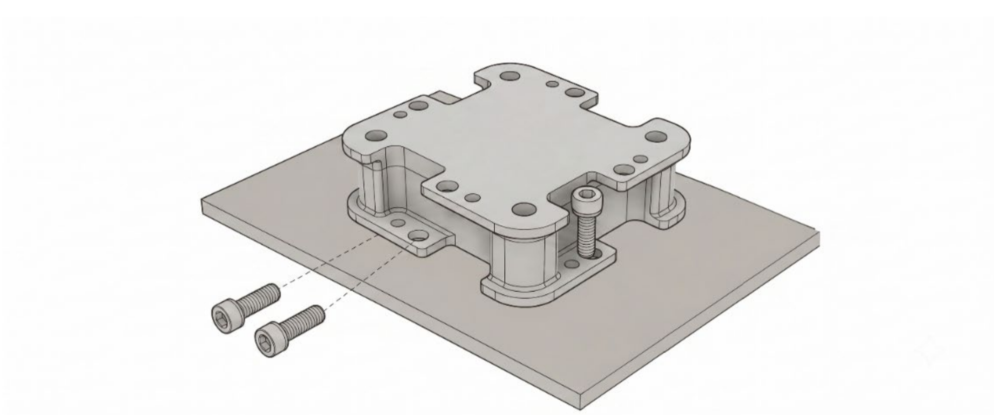


Figure 15: center/lean unit mounting orientation

This unit is sensitive to mounting quality. Any remounting or vehicle transfer requires recalibration.

Figure 16: Center Unit Final Assembly

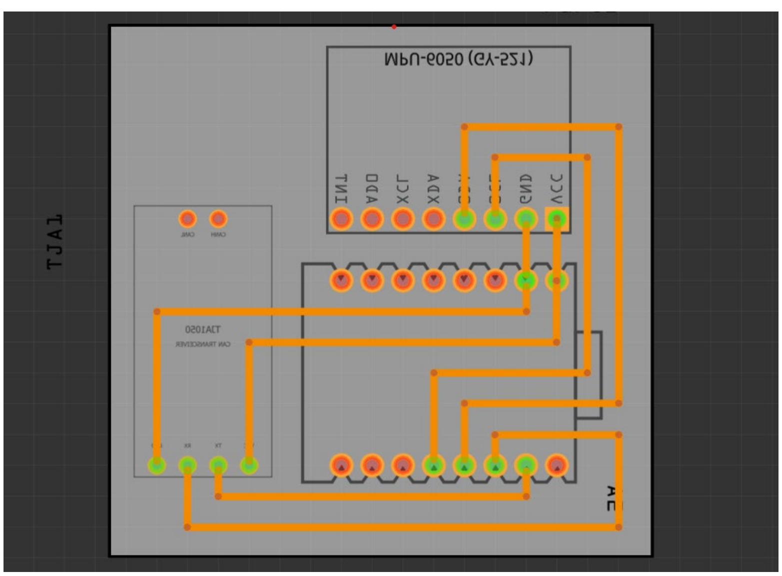


Figure 17: Center Unit PCB Layout

Key Electronic Components

Component	Function
ESP32-C3 SuperMini	Local controller and CAN communication
MPU6050 IMU	Roll and pitch measurement
TJA1050 CAN Transceiver	Physical CAN bus interface
Waterproof Connector	Vehicle wiring interface
Few Resistors	Voltage dividers

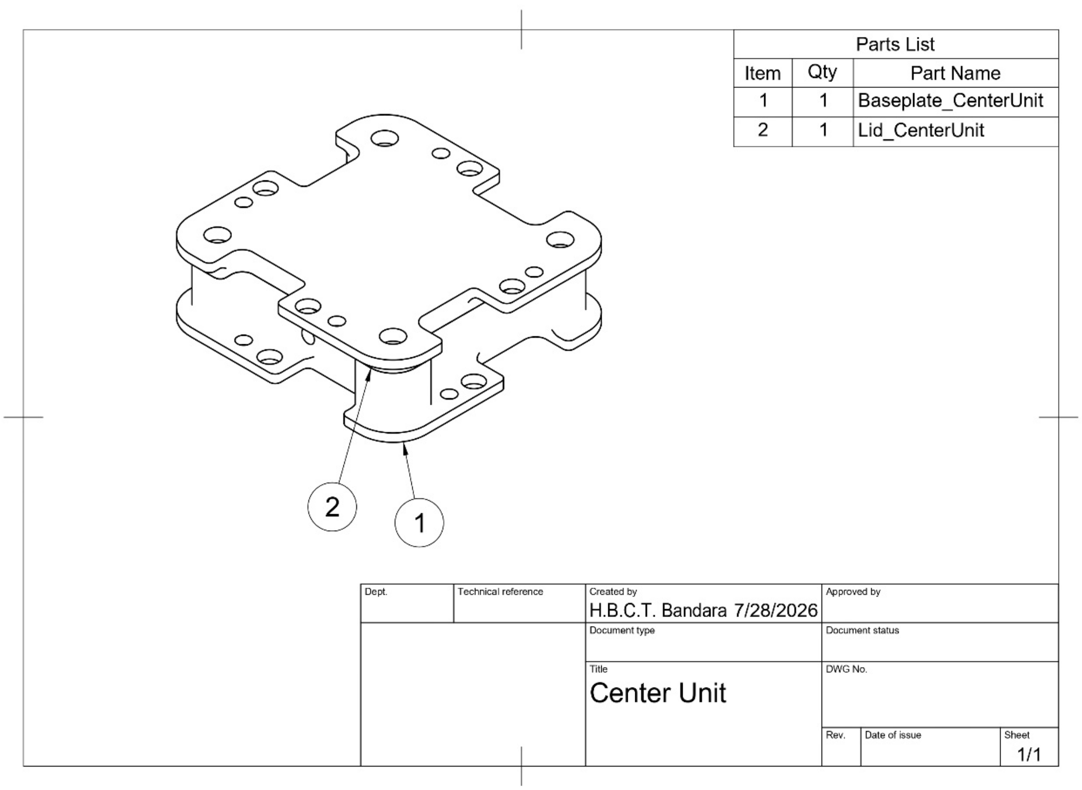


Figure 18: Center Unit Enclosure - Isometric View

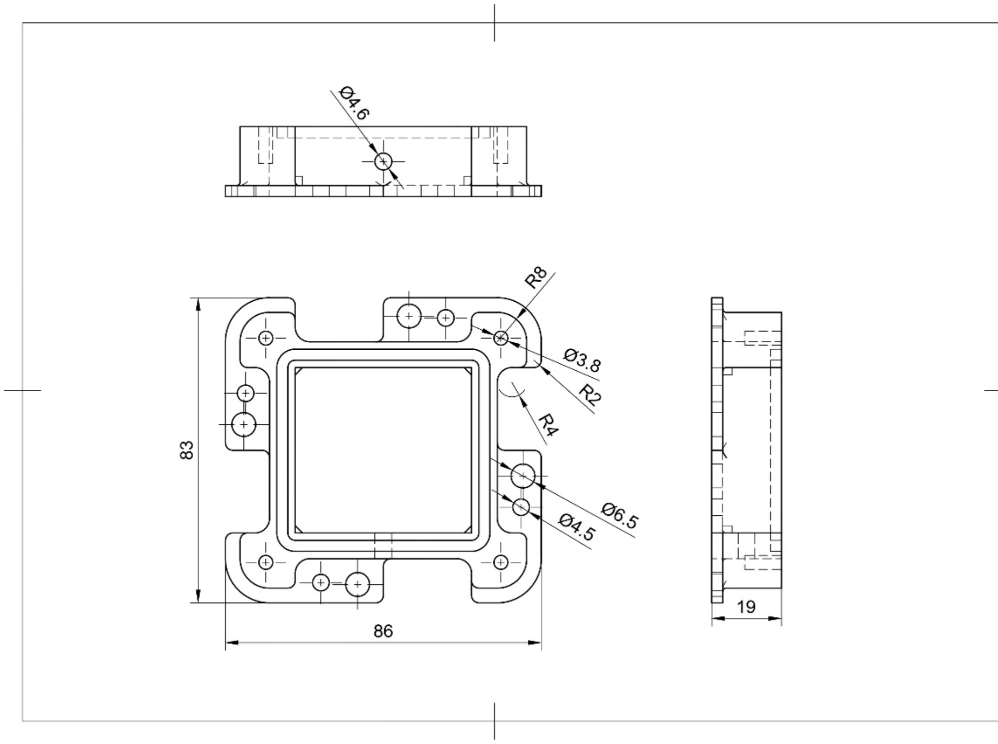


Figure 19: Center Unit Base Plate - Orthographic Views

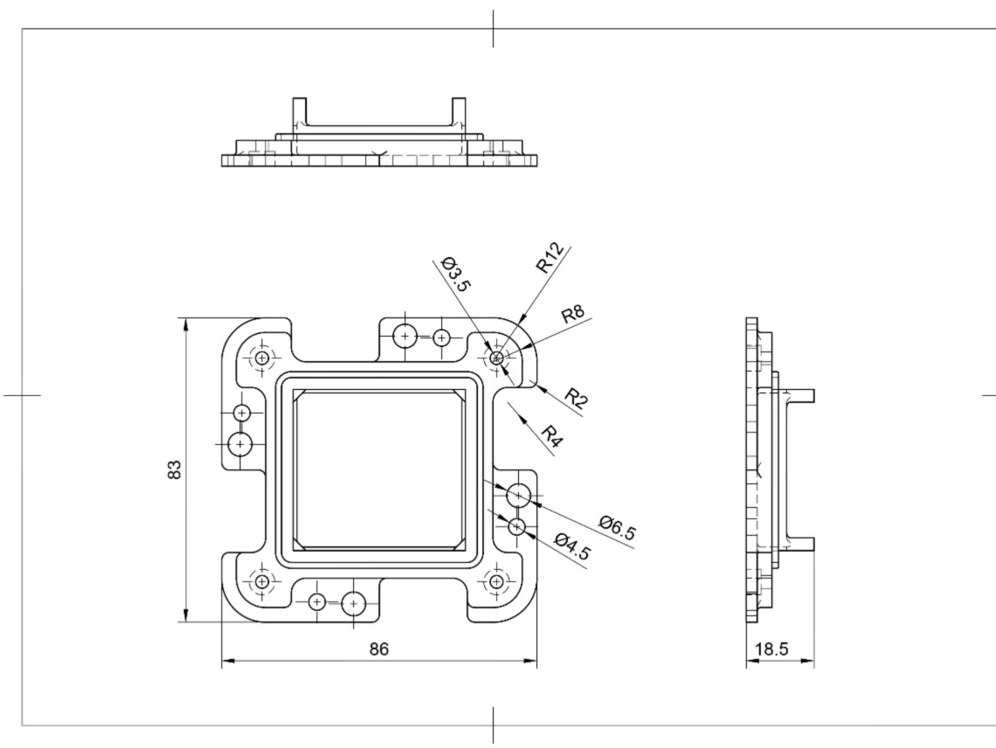


Figure 20: Center Unit Lid - Orthographic Views

3.4 Brain Unit + Camera Unit (Brain Module)

The brain unit is the central processor of the system. It performs the following roles:

- receives sensor and camera data,
- runs warning fusion logic,
- selects the primary hazard state,
- controls the buzzer,
- manages local Wi-Fi communication,
- sends data to the mobile application,
- stores and forwards incident data.

The camera unit is used for lane departure detection and camera-based validation of forward road information. It is connected directly to the brain unit and aligned toward the road and lane area.

The camera should be mounted so that road markings are visible, the horizon is stable, and the driver's forward visibility is not obstructed.

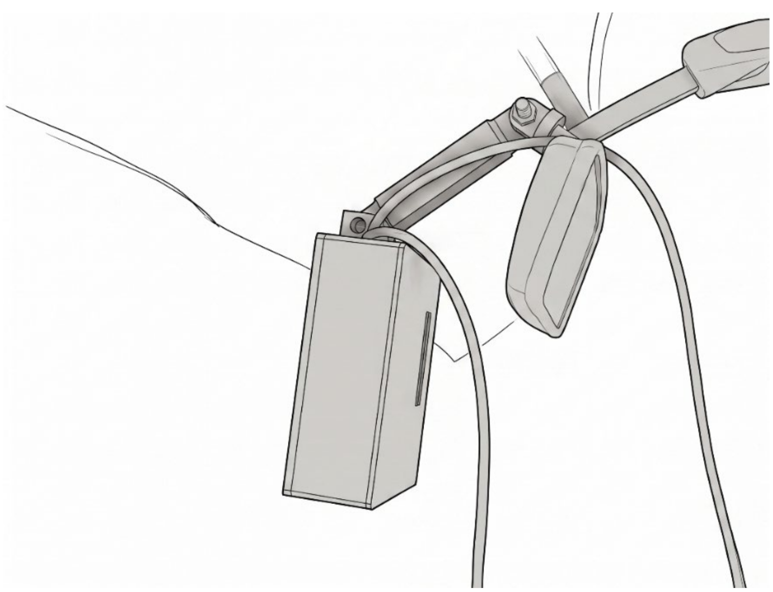


Figure 21: Brain Module mounting position near the rear-view mirror.

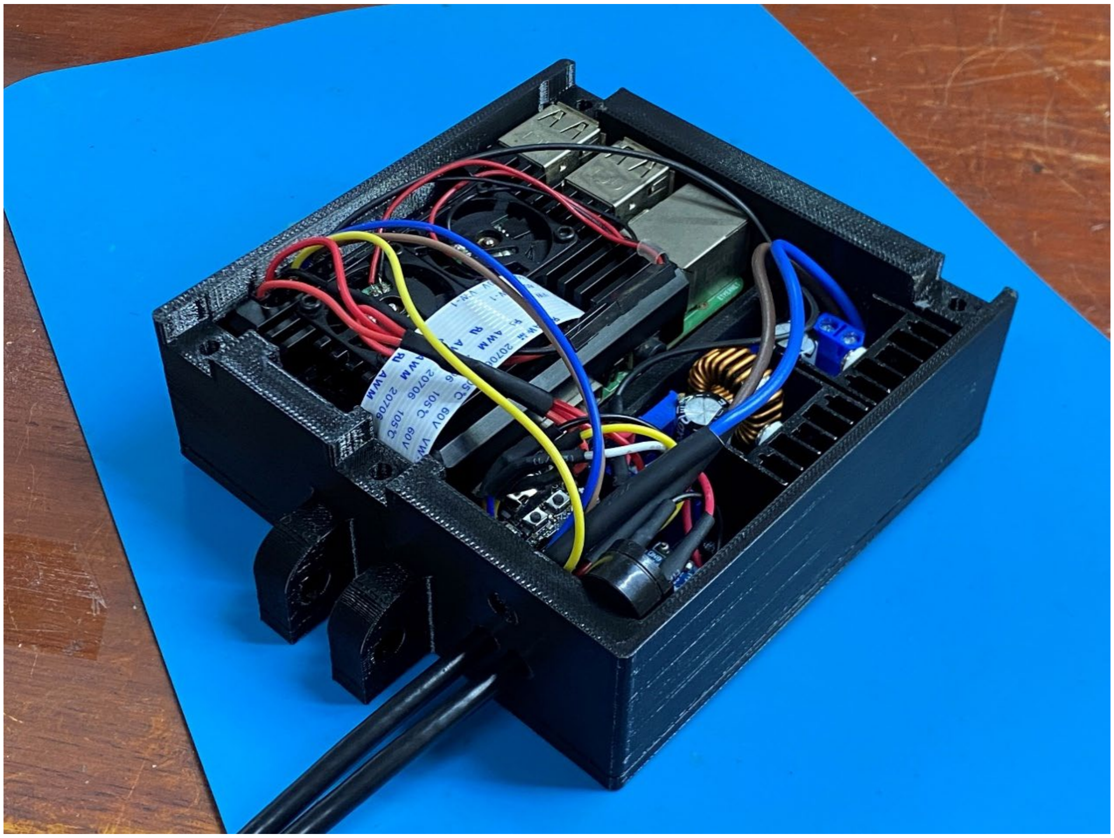


Figure 22: Brain Module Final Assembly

Key Electronic Components

Component	Function
Raspberry Pi 3 Model B	Central processing unit
Raspberry Pi Camera Module	Lane image acquisition
ESP32-C3 CAN Bridge	CAN interface for Raspberry Pi
Active Buzzer	Audible warnings
2x Cooling Fans	Thermal management
12 V – 5 V Buck Converter	Main power regulation
Toggle Switch	Manual power control
Reset Push Button	Raspberry Pi reset
Status LED	System status indication

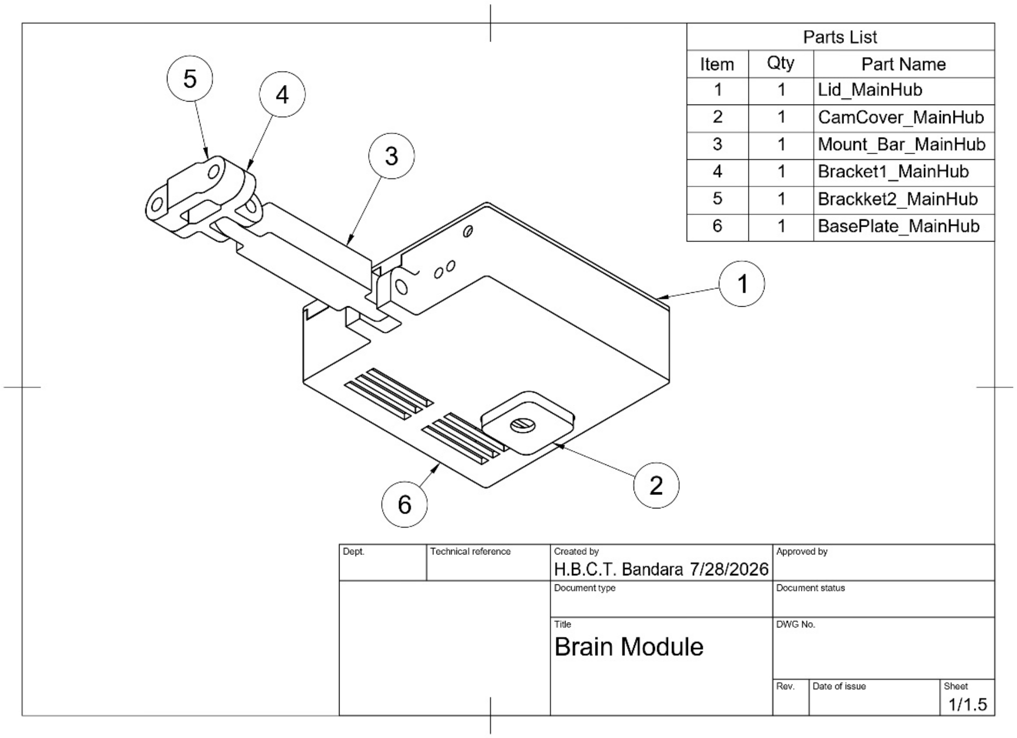


Figure 23: Brain Module Enclosure - Isometric View

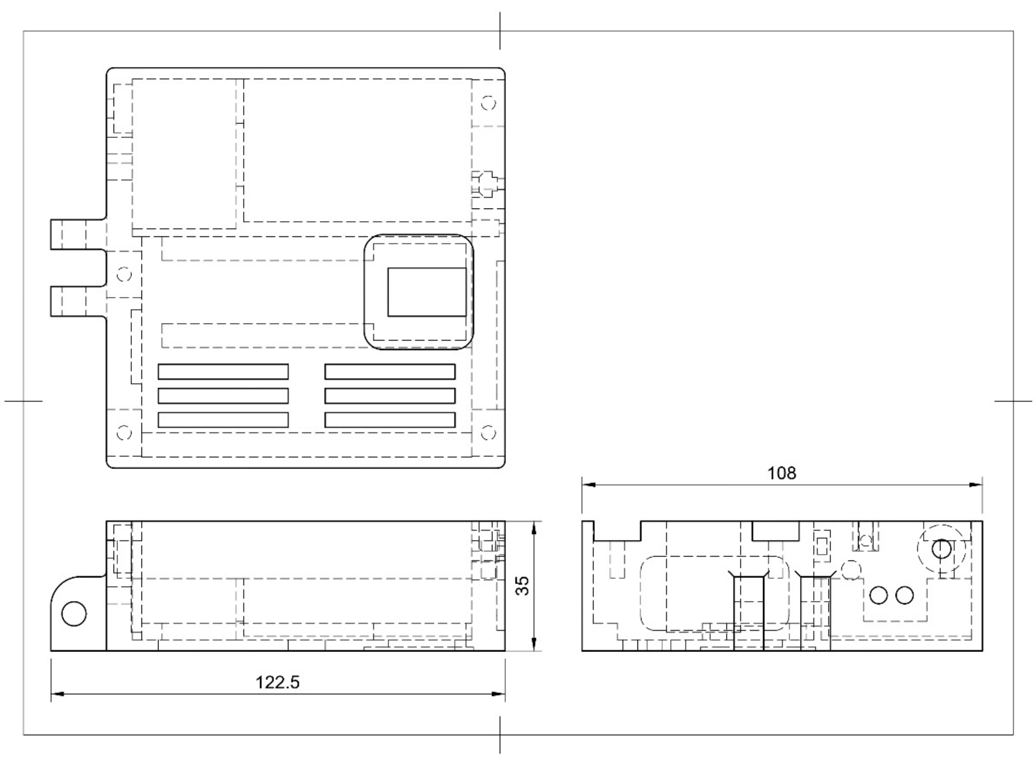


Figure 24: Brain Module Base Plate - Orthographic Views

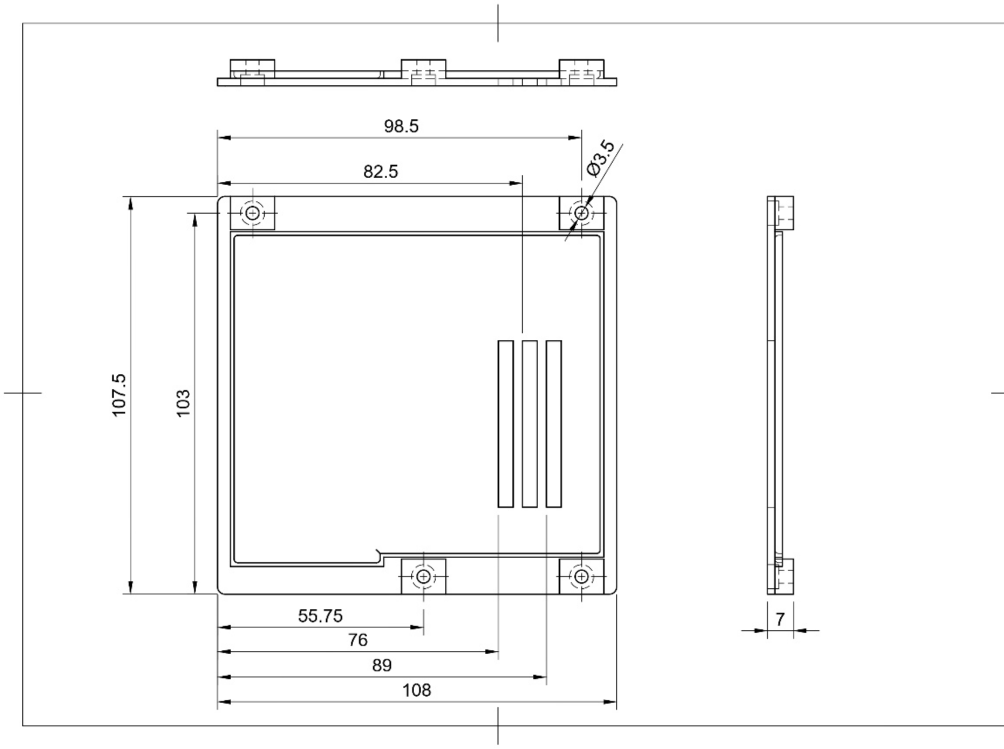


Figure 25: Brain Module Lid – Orthographic Views

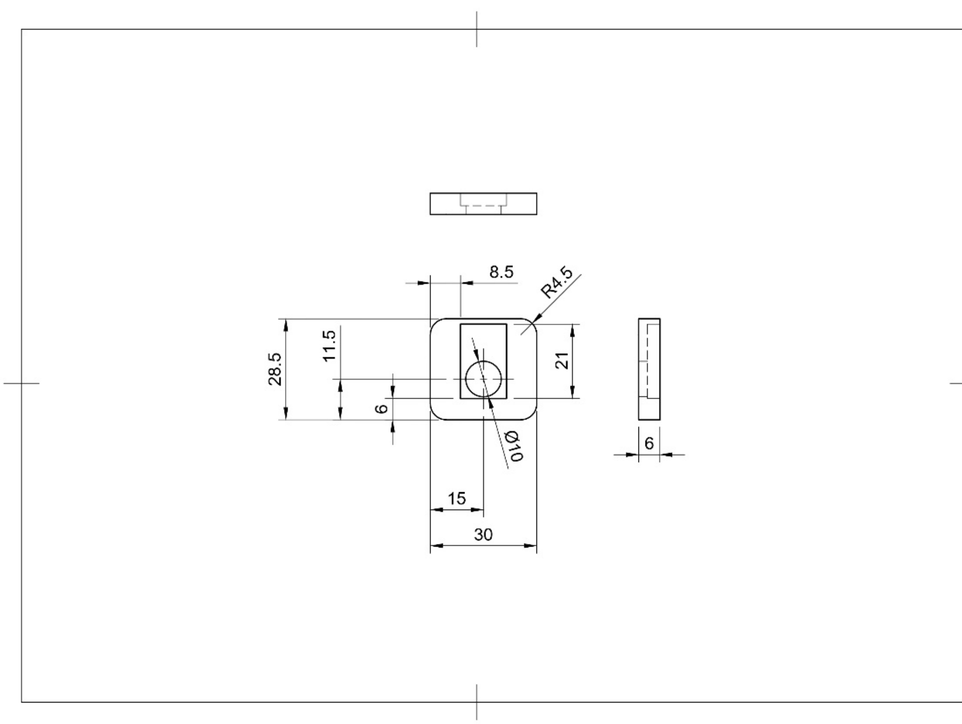


Figure 26: Brain Module Camera Cover – Orthographic Views

3.5 Environmental Protection

The front and center units use thermal and water-resistant enclosures with sealing around cable exits. Waterproof connectors are used for exposed connections, and ultrasonic probes are selected for outdoor mounting positions. The rear hub remains inside the vehicle, while only the rear probes are exposed through the bumper.

Figure 27: Rubber O-ring gaskets inside Front and Center units

4. Power and Installation Design

Drivora is powered from the vehicle's 12 V battery and converted to a regulated 5 V supply for the electronics. The power design includes protection before the conversion stage to reduce the risk of damage from incorrect wiring or transient conditions.

4.1 Power Path

The simplified power path is:

Vehicle 12 V battery → Protection Stage → Buck Converter → Regulated 5 V Distribution

4.2 Power-On Behavior

The system supports two startup modes:

- automatic startup when connected to the vehicle key power line,
- manual startup using the physical switch on the brain unit.

The automatic mode is preferred for normal use because it aligns the system state with the vehicle ignition state. The manual switch is mainly for testing, recovery, and service situations.

4.3 Shutdown and Restart Strategy

The app-based shutdown/restart option is preferred when available. Physical switching should only be used when the app is unavailable or when a complete power cycle is required. This reduces the risk of inconsistent service states in the brain and camera services.

4.4 Installation Notes

Installation should preserve both mechanical stability and electrical reliability. The main points are:

- front and center units require secure mounting and clear sensor exposure,
- rear probes require precise bumper placement and equal spacing,
- the camera must remain stable and properly aligned,
- the brain unit should be mounted where it is accessible but protected,
- wiring must be routed away from moving parts, high-heat areas, and sharp edges.

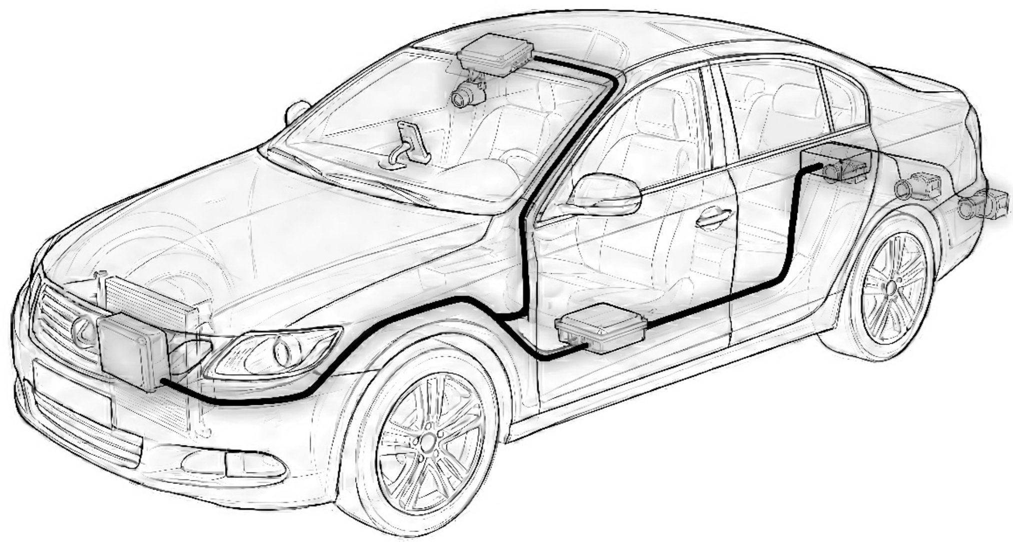


Figure 28: recommended mounting locations on the vehicle

4.5 Rear Bumper Drilling

The rear sensor probes require three bumper openings for left, center, and right placement. Hole size and spacing must match the probe hardware and vehicle body design. The mounting position should be finalized before drilling, since misalignment can affect rear detection coverage.

4.6 Installation Constraint

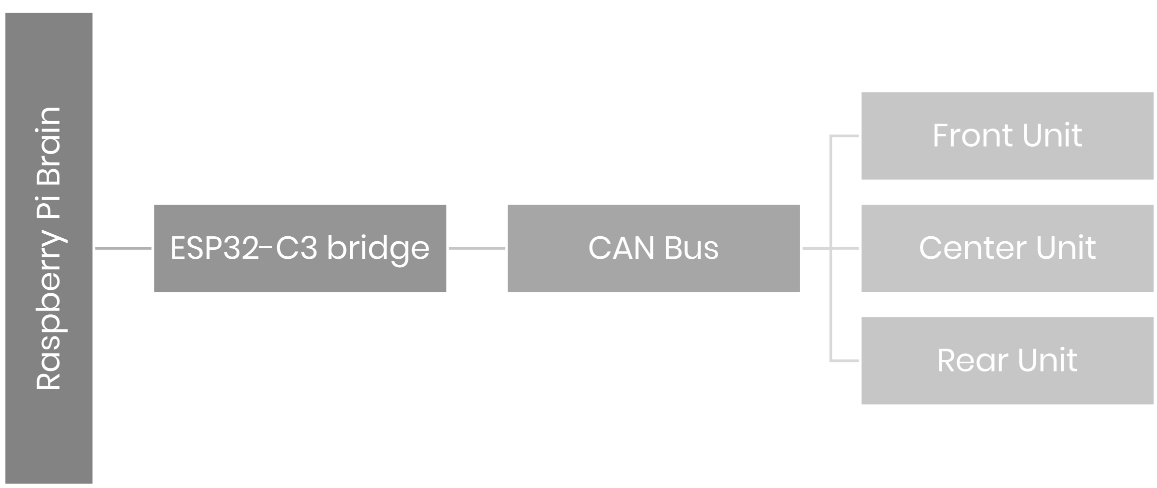
The system should be installed only after vehicle power is isolated and the mounting plan is confirmed. Repositioning a unit after installation may require both mechanical adjustment and full recalibration.

5. Communication Design

Drivora uses a mixed communication architecture to separate real-time vehicle sensing from user-facing application traffic.

5.1 CAN Communication

Drivora uses Classical CAN for communication between the Raspberry Pi brain and the vehicle-edge sensing units. The CAN bus is used for the front unit, rear unit, and center/lean unit. The camera/lane unit is not carried on CAN; it is handled separately by the Raspberry Pi camera service and UDP-based telemetry. The brain interfaces with the CAN bus through an ESP32-C3 CAN bridge connected over UART.



5.1.1 CAN Bus Configuration

Parameter	Value
CAN type	Classical CAN
Identifier format	Standard 11-bit
Data length	8-byte frames
Bit rate	500 kbps
Byte order	Little-endian for multi-byte values

The brain treats CAN-based unit status using timeout logic. A unit is marked stale after 300 ms without a valid update and offline after 1000 ms. This status model is used in both the dashboard and the warning fusion logic.

5.1.2 CAN Frame Map

Frame ID	Direction	Function
0x100	Unit → Brain	Center/lean main telemetry
0x101	Unit → Brain	Center/lean debug telemetry
0x110	Brain → Unit	Center configuration A
0x111	Brain → Unit	Center calibration command
0x112	Brain → Unit	Center configuration B
0x200	Unit → Brain	Front main telemetry
0x201	Unit → Brain	Front debug telemetry
0x210	Brain → Unit	Front sensitivity preset
0x300	Unit → Brain	Rear main telemetry
0x301	Unit → Brain	Rear debug telemetry
0x302	Unit → Brain	Rear distance telemetry
0x310	Brain → Unit	Rear sensitivity preset

5.1.3 Payload Summary

The brain code confirms the following payload structure:

Center / Lean Unit

- 0x100: risk level, roll angle, pitch angle, confidence, and calibration flag
- 0x101: critical roll threshold, critical pitch threshold, vehicle type, load condition, and debug counters

Front Unit

- 0x200: front state, filtered distance, closing speed, and raw distance
- 0x201: debug flags and internal counters
- 0x210: front sensitivity preset sent from the brain

Rear Unit

- 0x300: left, center, right, and overall rear warning state
- 0x301: rear debug counters and release-state telemetry
- 0x302: filtered distance values for left, center, and right probes plus nearest-sensor index
- 0x310: rear sensitivity preset sent from the brain

5.1.4 Configuration Flow

At startup and whenever settings change, the brain broadcasts vehicle configuration to the sensing units. The lean unit receives vehicle type, load condition, track width, wheelbase, and vehicle height through separate configuration frames, while the front and rear units receive their sensitivity presets. The brain also sends a dedicated lean calibration command when center calibration is requested.

5.1.5 Implementation Notes

The CAN bridge receives and forwards JSON-encoded frame data over the serial link to the Raspberry Pi. On the brain side, each received frame updates the corresponding subsystem state, which is then used for fused warning generation, dashboard status display, and incident logging. The lane/camera service remains separate and sends its data through UDP rather than CAN.

5.2 Local Wi-Fi Communication

The Flutter application communicates with the brain through a local Wi-Fi hotspot hosted by the Raspberry Pi. This keeps the live driving interface independent of external internet access.

The local Wi-Fi link is used for:

- dashboard updates,
- setup wizard communication,
- settings changes,
- statistics synchronization,
- service status reporting.

5.3 Firebase and Local Storage

Firebase is used for user authentication, vehicle profile storage, and incident synchronization. Local storage is used to preserve important records when internet access is unavailable. When connectivity returns, the stored incident data can be synchronized.

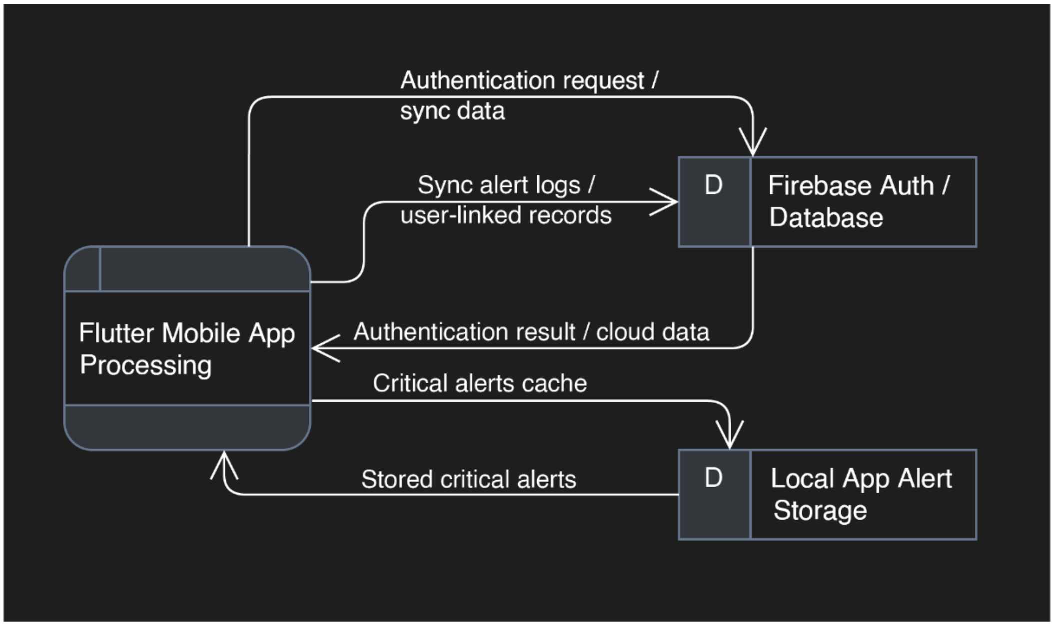


Figure 29: Communication flow between app, local storage, and Firebase

5.4 Communication Strategy

The communication design follows a layered approach:

1. low-level sensing data remains on the vehicle network,
2. fused warning results are exposed to the app,
3. account and profile data are synchronized to the cloud,
4. local data is retained first to avoid loss during offline operation.

6. Software and Firmware Design

The software design separates responsibilities across the embedded units, the Raspberry Pi brain, and the mobile application.

6.1 Front ESP32-C3 Firmware

The front unit firmware reads the ultrasonic sensors, filters the measured distance, estimates closing speed, and transmits telemetry at regular intervals. The firmware also supports debug frames for technical verification.

6.2 Rear ESP32-C3 Firmware

The rear unit firmware manages three rear probes and reports left, center, and right zone activity. It applies filtering and warning latching so that short transient readings do not cause unstable alert behavior.

6.3 Center ESP32-C3 Firmware

The center unit firmware reads IMU data, computes roll and pitch, and applies calibration parameters based on the vehicle profile. This unit reports stability-related risk information to the brain.

6.4 ESP32-C3 CAN Bridge Firmware

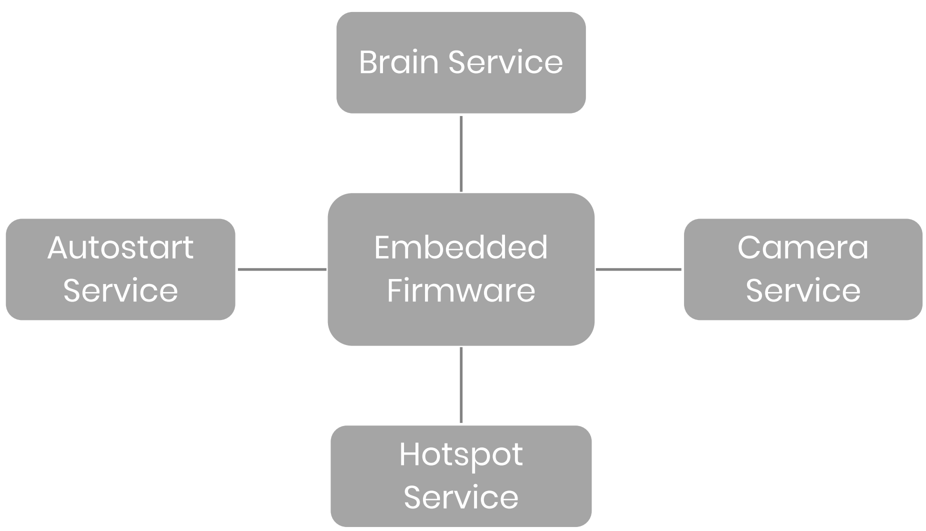
The CAN bridge firmware operates as the interface between the Raspberry Pi and the CAN bus. It forwards frames between the processing layer and the sensing units using TWAI/CAN at the configured bus speed.

6.5 Raspberry Pi Brain Service

The brain service is the core runtime component of Drivora. It performs configuration synchronization, warning fusion, incident classification, buzzer control, and app communication.

Its main tasks are:

- receive sensor frames,
- interpret unit status,
- fuse hazard conditions,
- select the primary warning,
- drive audio alerts,
- expose data to the mobile app,
- manage event records.



6.6 Camera Service

The camera service uses the Raspberry Pi Camera Module and image-processing logic for lane departure tracking. It exchanges status and telemetry with the brain service.

6.7 Autostart Services

System startup is managed through service automation so the hotspot, brain process, and camera process can launch automatically when the system powers on. This reduces manual recovery steps after ignition cycles.

7. Mobile Application Design

The mobile application is the user-facing interface for live monitoring, setup, and configuration.

7.1 Application Role

The app does not make final safety decisions. Its role is to display system state, provide setup controls, and present recent history in a clean driver-friendly layout.

7.2 Top-Level Pages

Page	Purpose
Dashboard	Live driving view with fused warning display
Statistics	Driver score, event count, and incident history
Settings	Vehicle profile, sensitivity, sound, connectivity, and reset

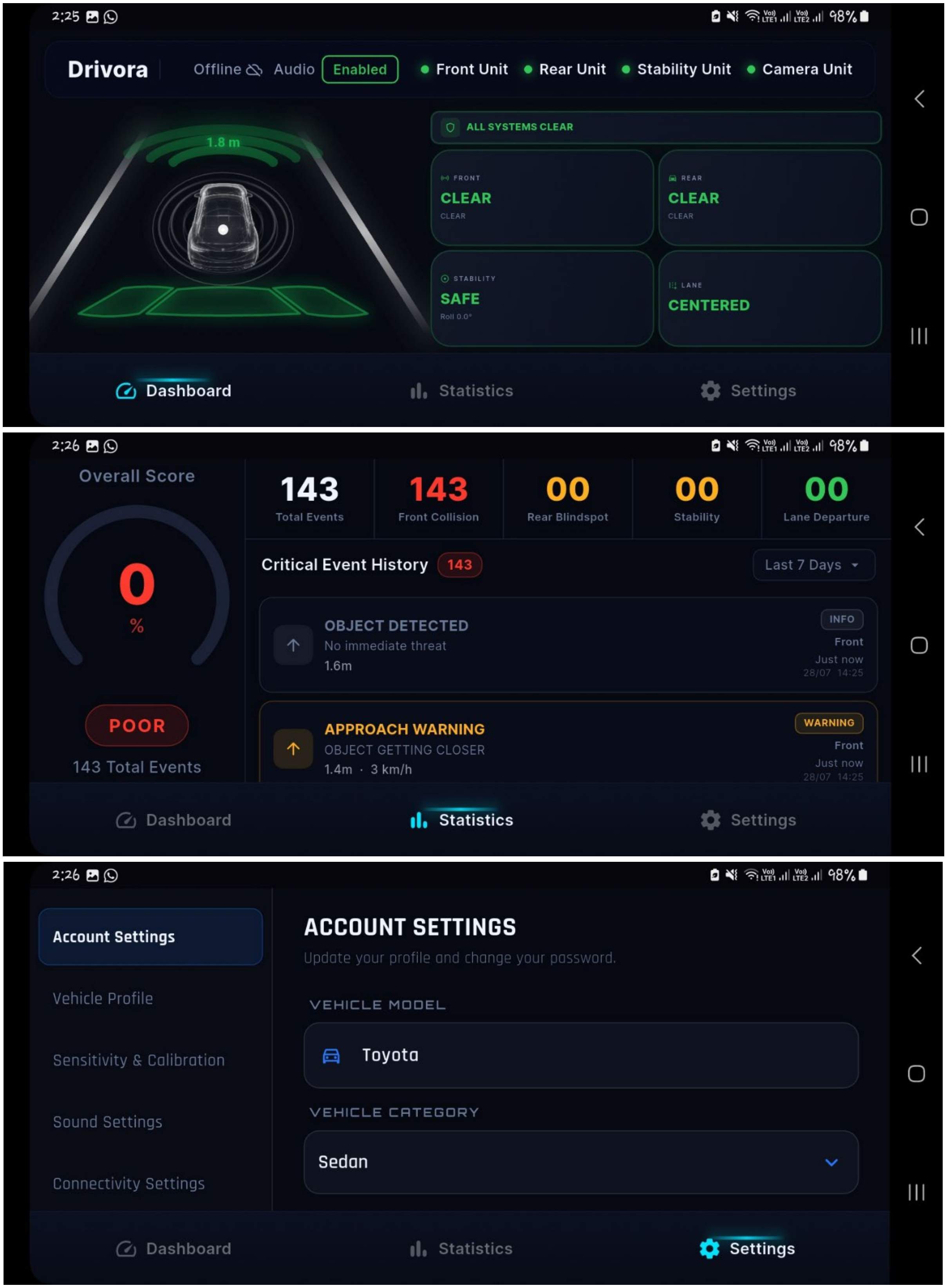


Figure 30: Three main pages of the mobile app

7.3 Dashboard Design Principles

The dashboard is designed for quick glanceability while driving. The main design decisions are:

- show one primary fused warning prominently,
- use color-coded unit status dots,
- keep supporting text short,
- preserve a spatial vehicle visualization,
- avoid cluttering the screen with equal-priority alerts.

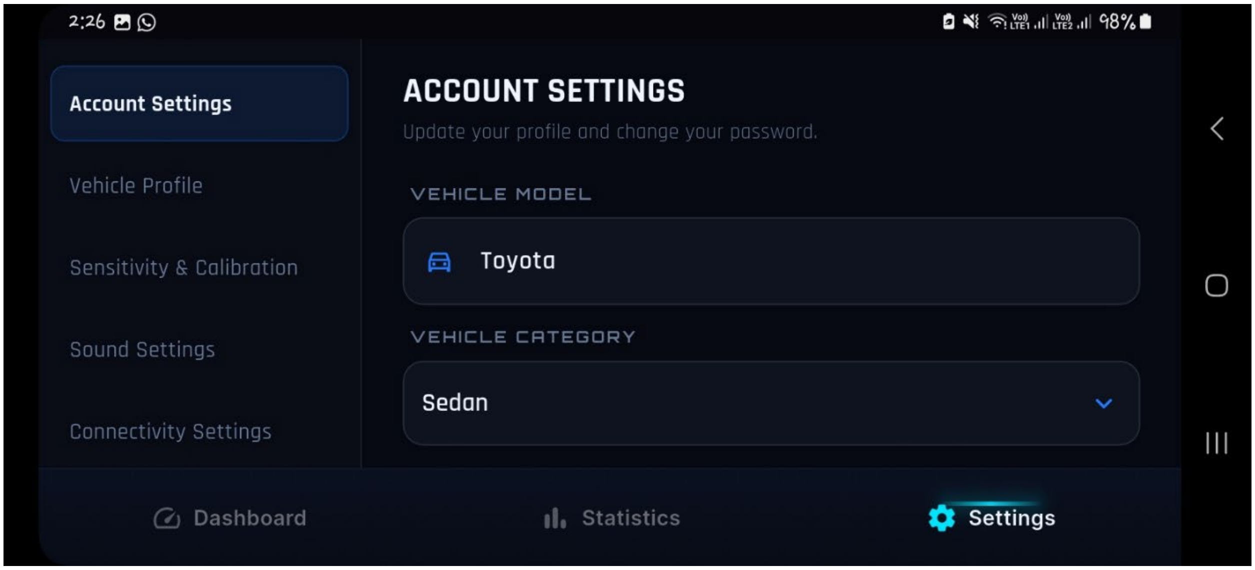
7.4 Statistics Page

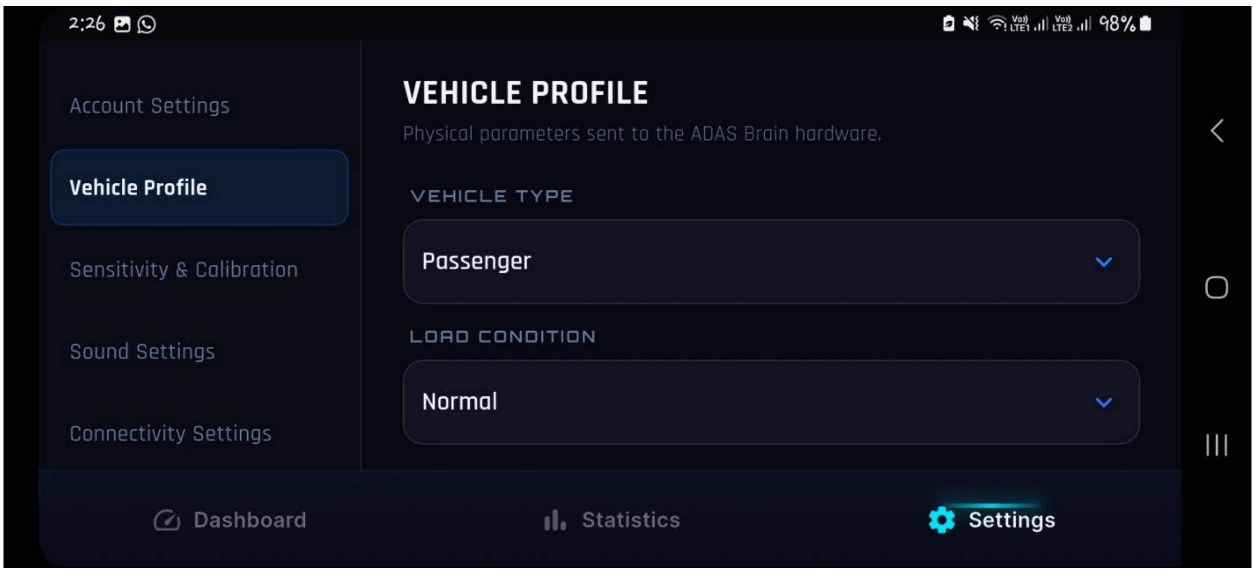
The statistics page summarizes recent driving events and provides a compact view of warning history and driver score trends.

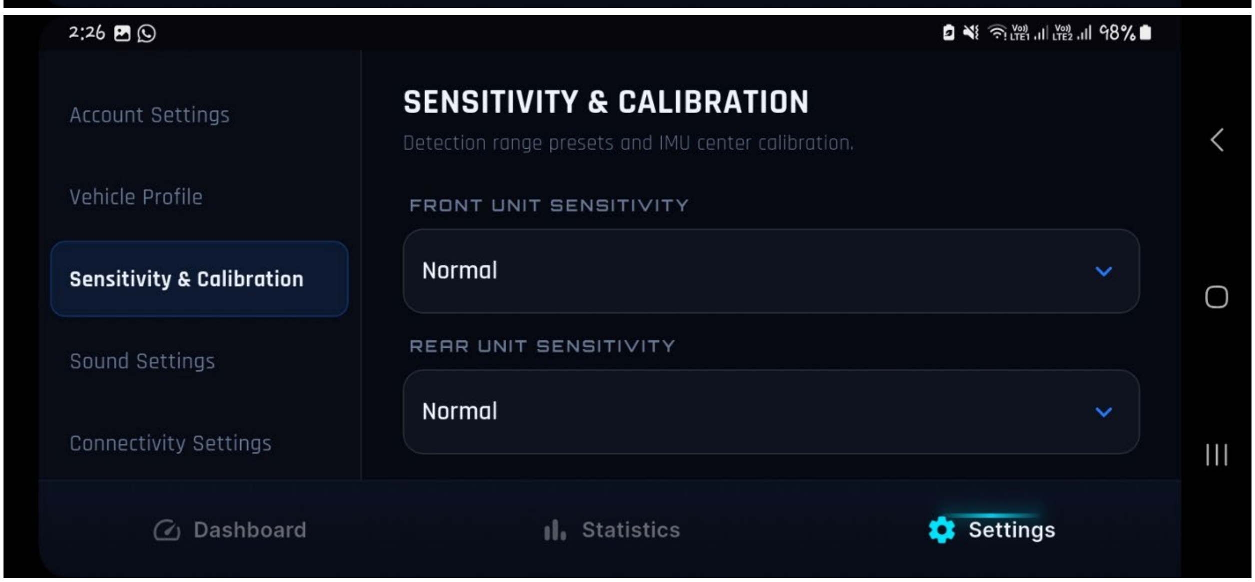
7.5 Settings Page

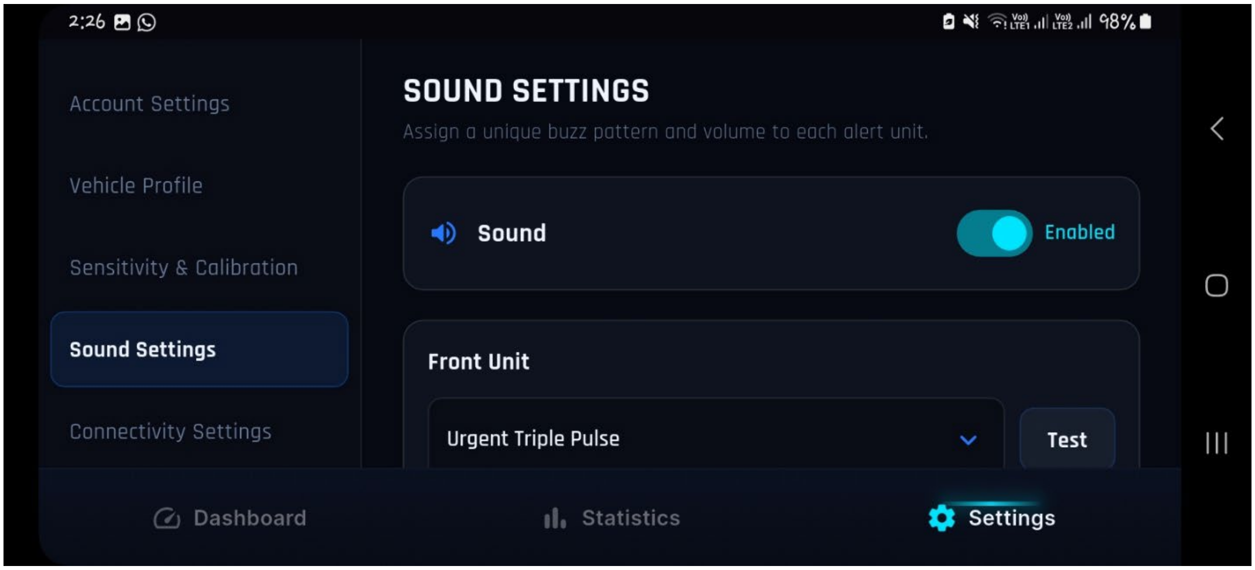
The settings page provides access to:

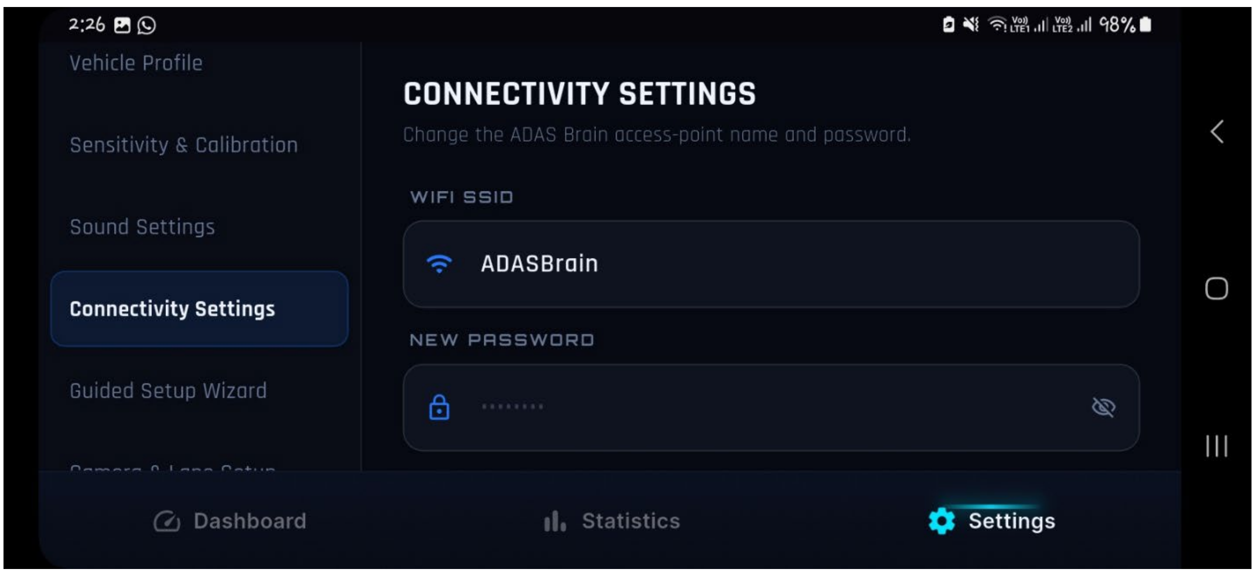
- vehicle profile,
- sensitivity presets,
- sound settings,
- connectivity settings,
- guided setup wizard,
- system reset.

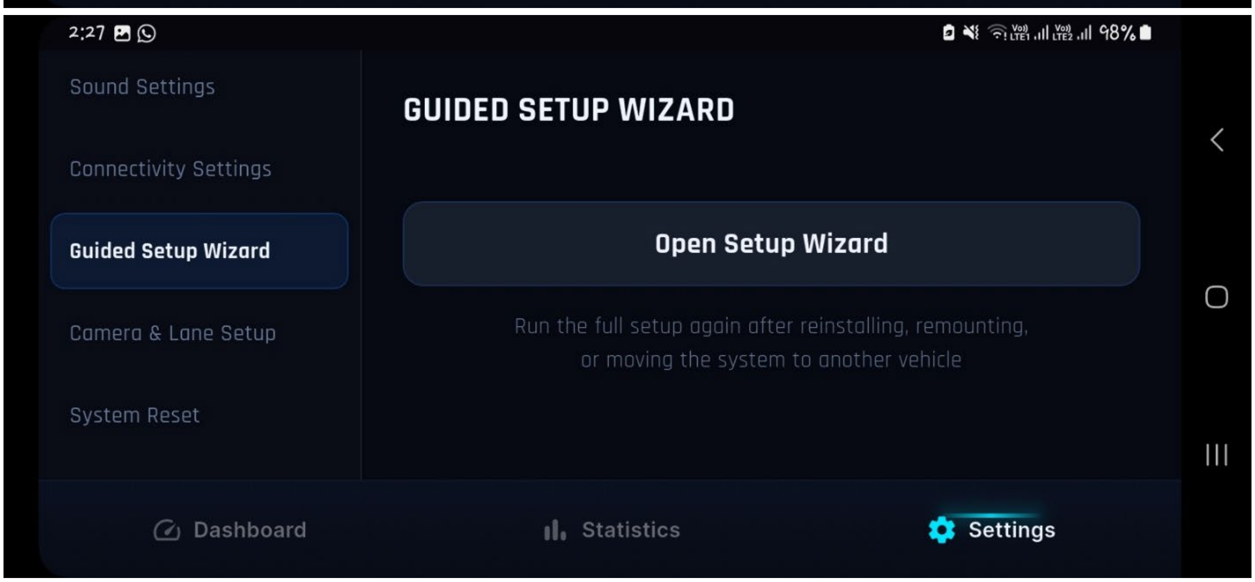

2:26
98%
Account Settings
Vehicle Profile
Sensitivity & Calibration
Sound Settings
Connectivity Settings
ACCOUNT SETTINGS
Update your profile and change your password.
VEHICLE MODEL
Toyota
VEHICLE CATEGORY
Sedan
Dashboard
Statistics
Settings


2:26
98%
Account Settings
Vehicle Profile
Sensitivity & Calibration
Sound Settings
Connectivity Settings
VEHICLE PROFILE
Physical parameters sent to the ADAS Brain hardware.
VEHICLE TYPE
Passenger
LOAD CONDITION
Normal
Dashboard
Statistics
Settings


2:26
98%
Account Settings
Vehicle Profile
Sensitivity & Calibration
Sound Settings
Connectivity Settings
SENSITIVITY & CALIBRATION
Detection range presets and IMU center calibration.
FRONT UNIT SENSITIVITY
Normal
REAR UNIT SENSITIVITY
Normal
Dashboard
Statistics
Settings


2:26
98%
Account Settings
Vehicle Profile
Sensitivity & Calibration
Sound Settings
Connectivity Settings
SOUND SETTINGS
Assign a unique buzz pattern and volume to each alert unit.
Sound
Enabled
Front Unit
Urgent Triple Pulse
Test
Dashboard
Statistics
Settings


2:26
98%
Vehicle Profile
Sensitivity & Calibration
Sound Settings
Connectivity Settings
Guided Setup Wizard
CONNECTIVITY SETTINGS
Change the ADAS Brain access-point name and password.
WIFI SSID
ADASBrain
NEW PASSWORD
Dashboard
Statistics
Settings


2:27
98%
Sound Settings
Connectivity Settings
Guided Setup Wizard
Camera & Lane Setup
System Reset
GUIDED SETUP WIZARD
Open Setup Wizard
Run the full setup again after reinstalling, remounting,
or moving the system to another vehicle
Dashboard
Statistics
Settings

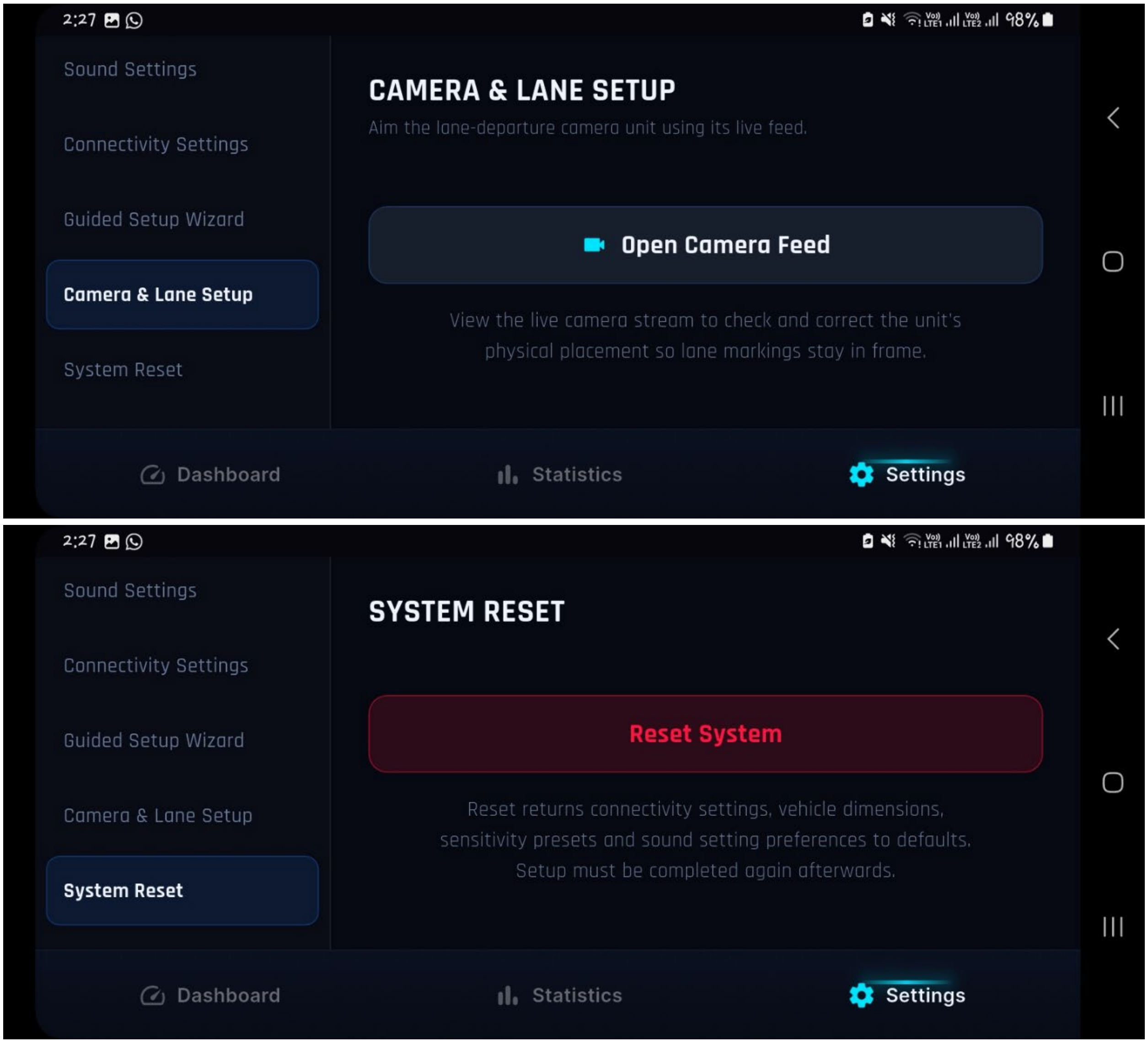


Figure 31: Settings page screens

8. Safety Logic and Incident Handling

Drivora uses the brain unit as the safety decision layer. The mobile application displays the output, but the brain performs warning fusion, priority selection, and incident classification.

8.1 Warning Fusion and Priority Handling

The brain evaluates the current states from the front, rear, center/lean, and lane subsystems and selects one primary fused warning at a time. This prevents conflicting alerts on the driver interface.

Priority	Condition	Fused Warning
1	Lean risk level = HIGH	High Lean Risk
2	Front state = WARNING and vision validated/recent	Front Collision Warning
3	Rear overall state = WARNING	Rear Blindspot Warning
4	Lane state = LEFT	Left Lane Departure
5	Lane state = RIGHT	Right Lane Departure
6	Lean risk level = CAUTION	Lean Caution
7	Front state = APPROACHING and vision validated/recent	Object Approaching
8	Rear overall state = CAUTION	Rear Caution
9	Front state = OBJECT and vision validated/recent	Object Ahead
10	Rear overall state = OBJECT	Rear Object Detected
—	No active condition	All Clear

The system always chooses the highest-priority active condition from the table above.

8.2 Critical Incident Rules

Incident logging is separate from live warning display. A condition is recorded only when it stays active long enough to satisfy the confirmation window.

Event Type	Trigger Condition	Confirmation Time
Front critical	Front warning remains active with valid vision confirmation	700 ms
Rear critical	Rear warning remains active	700 ms
Lean critical	Lean risk remains high	500 ms
Lane critical	Lane departure remains active	500 ms
Multi-hazard	Two or more serious conditions active together	500 ms

Additional rule:

- A cooldown period of 5000 ms is applied before the same type of incident can be logged again.

8.3 Driver Score Model

The driver score is a rolling indicator derived from recent incident history. It is used for summary statistics and trend visibility rather than direct vehicle control.

9. Maintenance, Diagnostics, and Limitations

9.1 Maintenance

Regular checks should focus on connector condition, sensor cleanliness, cable strain relief, and mounting stability.

Recommended checks:

- inspect waterproof connectors,
- clean ultrasonic sensor faces,
- verify camera lens clarity,
- confirm the IMU mount has not shifted,
- check whether the system needs recalibration after any remounting.

9.2 Diagnostics

If a fault occurs, technical debugging should begin with:

1. checking power rails,
2. checking connector integrity,
3. confirming service status on the brain unit,
4. verifying CAN communication,
5. confirming that the mobile app is connected to the correct local Wi-Fi network.

9.3 Known Limitations

The current implementation is a driver assistance prototype and does not replace driver attention or vehicle control. Performance depends on sensor placement, vehicle geometry, vibration, weather, and installation quality.

Important technical constraints include:

- ultrasonic performance depends on object surface and angle,
- lane detection depends on visible road markings and lighting,
- remounting components may require recalibration,
- the app cannot be treated as the safety decision source.

---

10. Conclusion

Drivora is designed as a modular, locally processed ADAS platform with separate sensing, processing, and presentation layers. The architecture supports vehicle-specific configuration, real-time warning fusion, and offline-first incident handling.

The current design prioritizes:

- reliable local operation,
- low driver distraction,
- hardware modularity,
- cloud synchronization only when available,
- clear separation between control logic and UI logic.

This design provides a practical technical basis for the implemented system and supports future extension or refinement.

11. Appendices

Appendix A: Setup Profile Fields

Field	Purpose
Vehicle type/category	Identifies the vehicle class
Track width	Used in geometry-related logic
Wheelbase	Used for vehicle profile calculations
Vehicle height	Used for profile estimation
Load condition	Supports stability-related behavior
Sensitivity presets	Controls warning thresholds
Calibration status	Indicates completed alignment
Connectivity state	Indicates Wi-Fi configuration status
Sound preferences	Controls buzzer behavior
Incident history	Stores warning events and sync status

Appendix B: Status Color Meanings

Color	Meaning
Green	Online / clear
Yellow	Stale / caution
Red	Offline / warning / critical

Appendix C: Revision History

Version	Date	Description
1.0	July 2026	Initial design manual draft

Project Repository

GitHub URL - https://github.com/cepdnaclk/e21-3yp-Drivora

Project Team

Name	Registration Number
H.B.C.T. Bandara	E/21/050
H.M.P.D. Bandara	E/21/052
K.M.S. Darshana	E/21/077
A.P.S. Nirmal	E/21/269

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya