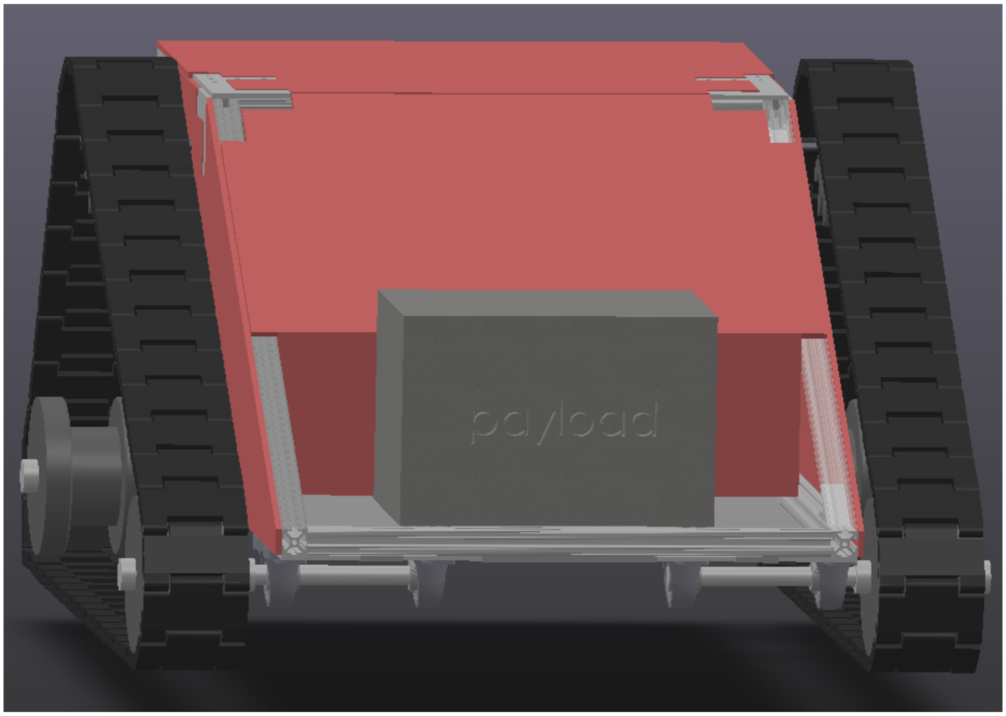
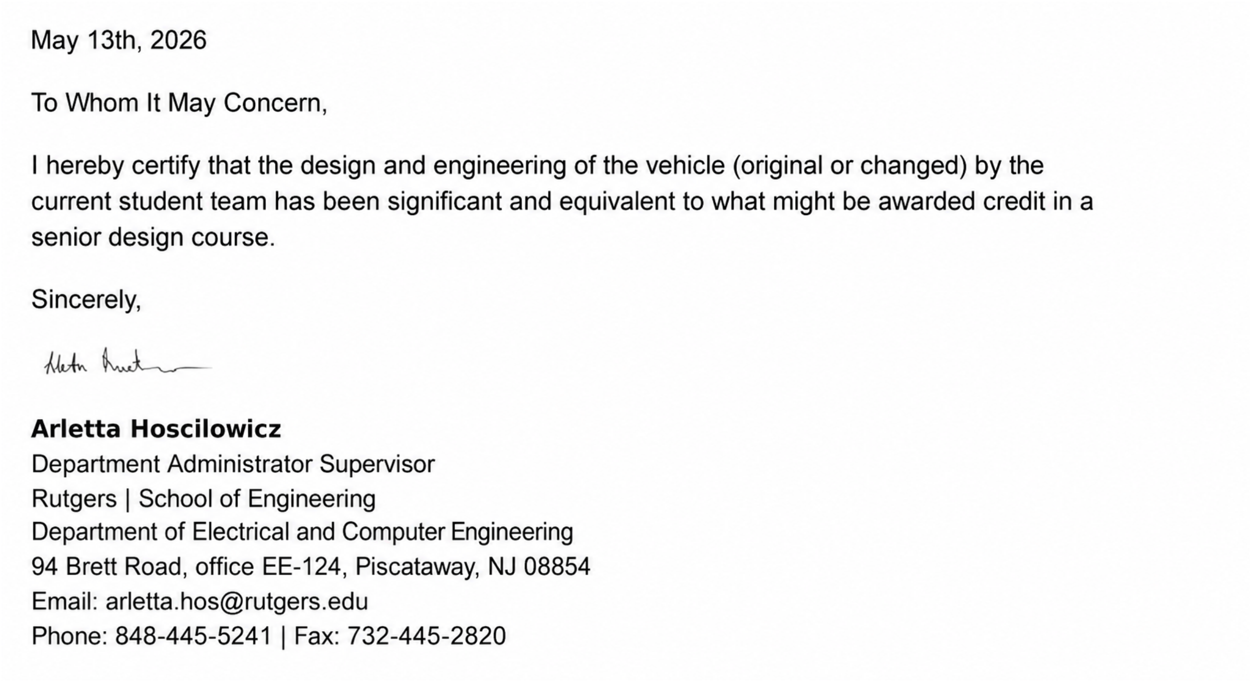
# Rutgers IGVC Robotics Team
# IGVC 2026 Design Report
# Scarlet Rover

Rutgers University

May 13th, 2026





May 13th, 2026

To Whom It May Concern,

I hereby certify that the design and engineering of the vehicle (original or changed) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Sincerely,

**Arletta Hoscilowicz**
Department Administrator Supervisor
Rutgers | School of Engineering
Department of Electrical and Computer Engineering
94 Brett Road, office EE-124, Piscataway, NJ 08854
Email: arletta.hos@rutgers.edu
Phone: 848-445-5241 | Fax: 732-445-2820

| **Electrical** | **Mechanical** | **Software** |
|---|---|---|
| Chance Reyes | Nicholas Duchatellier | Will Wands |
| Nikhil Reddy | Edward Dieguez | Rushika Thandra |
| Nikita Shmuylovskiy | Venkata Sai Donepudi | |
| Soham Joshi | | |
| Israel Lopez | | |
| Gavril Lomotey | | |
| Jerry Rafael | | |

## Contents

# 1 Design Process, Team Identification and Organization

## 1.1 Introduction

Rutgers IEEE is a student organization composed of multiple divisions, each focused on different areas of engineering. Within this organization, the Rutgers IGVC team represents the Intelligent Ground Vehicle Competition (IGVC) division, dedicated to the design and development of highly autonomous ground vehicles. Our focus is on applying robotics concepts such as autonomous navigation, computer vision, and system integration to create reliable and efficient systems. Through participation in IGVC, we aim to provide hands-on experience that reflects real-world engineering practices.

## 1.2 Organisation

The team is structured into several sub-teams, including electrical, mechanical, and software, each responsible for specific components of the vehicle. Members collaborate on sub-projects and meet regularly to track progress and address challenges, while full team meetings are held to discuss overall development and integration. Rutgers University has previously competed in IGVC in 2011, 2012, and consistently from 2022 through 2026, and the team is committed to continuing this involvement. Our goal is to foster growth in robotics and encourage students to pursue innovation in autonomous systems.

## 1.3 Design Process

The process began with post-analysis of the previous Rutgers IGVC vehicle. Several priorities were identified, such as improving power reliability, reducing wiring complexity, increasing subsystem modularity, strengthening weather resistance, improving software debugging capability, and simplifying maintenance and repair.

# 2 Mechanical Design

## 2.1 Overview

The mechanical team designed the vehicle to be more robust, modular, and adaptable than the team's previous mechanical design. The new platform prioritizes structural rigidity, ease of maintenance, sensor visibility, weather resistance, and strong outdoor terrain performance. A major improvement is the modular drivetrain system, which allows the robot to operate with either wheels or treads depending on the terrain and testing needs.

## 2.2 Chassis and Frame

The chassis was designed around a tank-inspired modular frame, with the Husky A300 from Clearpath serving as one of the main references. The frame supports both wheeled and treaded configurations, allowing the robot to be adapted for different course conditions without requiring a complete redesign. This modular approach improves maintenance, testing flexibility, and future upgradeability while keeping the structure strong enough for outdoor IGVC terrain.

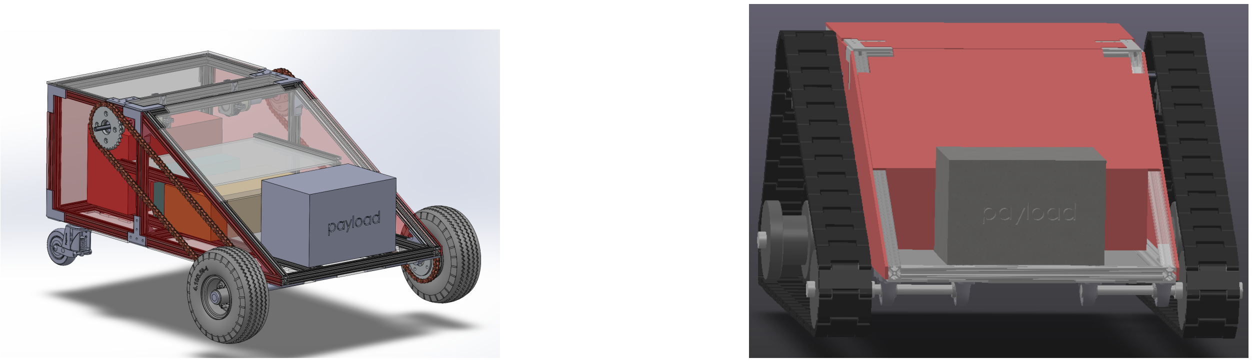


Figure 1: Left: Scarlet Rover configured with wheels. Right: Scarlet Rover with tracks installed.

### 2.3 Drivetrain

The drivetrain was selected to prioritize controllability, reliability, and outdoor terrain traversal rather than maximum speed. The system is designed to be modular, allowing the vehicle to switch between wheels and treads depending on the environmental conditions. The wheeled setup provides smoother and more efficient motion on pavement and flat surfaces, while the treaded setup provides increased traction and stability in rainy conditions. This flexibility allows the robot to maintain stable autonomous operation across the different surfaces commonly encountered during IGVC competition events.

### 2.4 Sensor Mounting

Sensor placement was optimized to maximize perception quality while minimizing vibration and occlusion. Proper sensor placement is critical because perception quality directly affects localization, mapping, lane detection, and obstacle avoidance performance.

### 2.5 Weatherproofing

The vehicle is designed for outdoor operation under varying environmental conditions. Weatherproofing considerations include enclosed electronics compartments, controlled cable routing, reduced connector exposure, and active thermal management.

## 3 Electronic and Power Design

### 3.1 Overview

The electrical framework of the Scarlet Rover is designed to handle safe power distribution, sensor management, and proper control over the rover during operation. To facilitate this, the majority of the electrical system is composed of custom printed circuit boards to handle tasks like thermal management, efficient board-to-board communication, and better fault protocols. The custom boards are able to communicate with one another and the NUC using a Controller Area Network (CAN) bus, imitating the robust system present in most automotive systems in industry. These custom boards and firmware allow for smoother operation and make room for upgrades in the future.

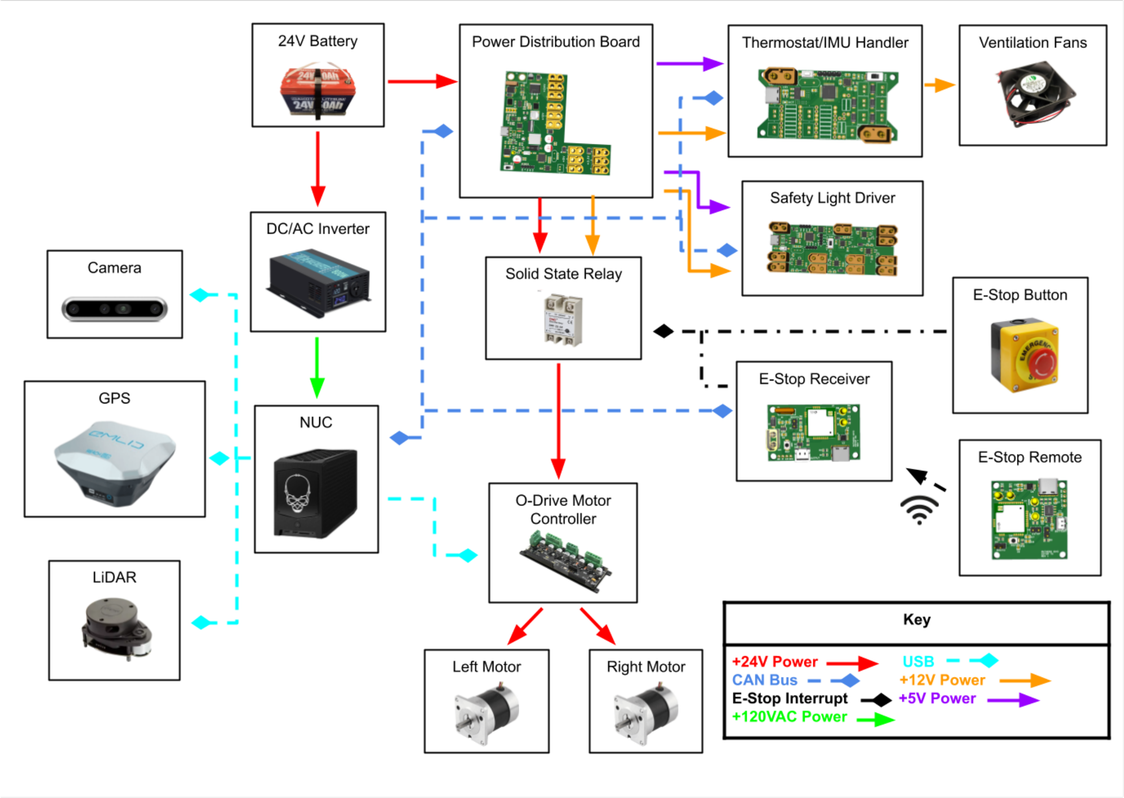


Figure 2: High-level electronic and power interaction diagram for Scarlet Rover.

### 3.2 Power Distribution

The electrical system and drivetrain are powered by a 24V 50Ah lithium battery, which should last more than long enough for the 5-minute run required of the rover. The power distribution board is custom-designed and utilizes a protected power path feeding multiple voltage rails for drivetrain systems, sensors, embedded hardware, and compute systems. Protection mechanisms include reverse polarity protection, transient suppression, overcurrent protection, and regulated rail generation for four different output rails: 24V, 12V, 5V, and 3.3V. The modular PCB architecture significantly simplifies debugging and future upgrades.

### 3.3 Motor Control

The Scarlet Rover possesses two 24V DC motors managed by an ODrive connected via USB to the NUC. The ODrive is powered by 24V from a solid-state relay controlled by the E-Stop, so that power to the motors is cut off when a fault is detected. Through the ODrive, the NUC can accurately control the velocity and orientation of the robot with much simpler programming than what would have been required if the ODrive were absent.

### 3.4 Safety and Emergency Stop

The emergency stop system is composed of a receiver onboard the rover and a remote in the operator's hand. The E-Stop uses LoRa communication to maintain reliable long-range wireless communication between the handheld emergency stop controller and the robot. Motor power is maintained by a solid-state relay that the E-Stop breaks upon firing. A secondary onboard emergency stop button provides redundant local shutdown capability. If communication is lost or any subsystem enters an invalid operating condition, the robot transitions toward a safe, stopped state.

### 3.5 Thermal Control

In order to allow the rover to operate in high-temperature conditions, a custom PCB is present to record and maintain heat inside the rover. NTC thermistors are used within an operational amplifier array to calculate the surrounding temperature at each motor, the battery, and the ambient air inside the robot. The STM32 on the PCB can then read this information and use PWM to control up to five ventilation fans to expel extra heat from the rover, avoiding niche failure states. The board also contains an MPU-6050 IMU for extra positional and rotational information, which is passed through the CAN bus to assist the NUC's navigation in combination with the other sensors.

### 3.6 Safety Light Indication

The Scarlet Rover contains an LED indicator system to notify the surroundings of its current operating state. The indicator lights are controlled by an STM32 on a custom PCB linked with the CAN bus across the rest of the rover, allowing the lights to update in accordance with the current condition of operation. The rover has 10 controllable lights driven by 12V from the power distribution board.

| **LED Color** | **Condition** |
|---|---|
| Green | Standby: powered, nominal operation, not moving |
| Yellow | Moving: navigating, autonomous operation, nominal operation |
| Red | E-Stopped: fault detected or abnormal state |

Table 1: Safety light operating states.

# 4 Software Systems

## 4.1 Overview

The software stack is built using ROS2 and follows a modular distributed-node architecture designed for scalability, debugging, and subsystem isolation. The system uses a combination of C++ and Python depending on performance requirements.

Core infrastructure includes:

- RCLCPP / RCLPY
- OpenCV
- TF2
- Nav2 Costmap 2D
- RViz2

## 4.2 Perception

The perception system processes sensor data to identify lanes, obstacles, and navigable space. Camera and depth data are converted into representations usable by the navigation and planning systems.

## 4.3 Behavior and Arbitration

The autonomy stack separates perception from behavioral decision-making using a layered arbitration system. Obstacle-avoidance logic operates alongside lane-following and waypoint-navigation systems to generate safe autonomous motion commands.

The navigation system incorporates Vector Field Histogram-style obstacle analysis and command blending techniques to maintain smooth and reactive motion.

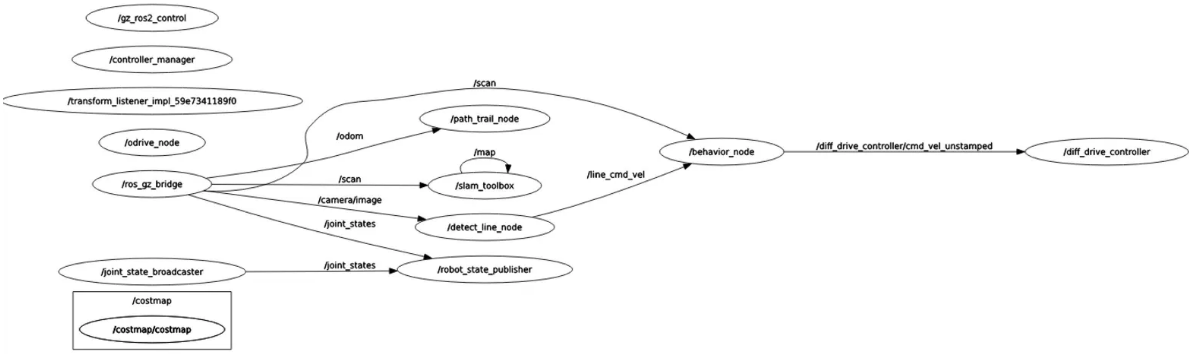


Figure 3: ROS2 node graph showing the active autonomy pipeline during simulation. Lane detection commands flow from the `/detect_line` node through the `/behavior` node to `/ros_gz_bridge`, with sensor data from `/camera/image` and `/scan` feeding perception and SLAM through `/slam_toolbox`.

### 4.4 Localization, Mapping, Planning, and Controls

The autonomy system combines localization, mapping, path planning, and controls into one connected navigation pipeline. Localization estimates the vehicle's position and heading using data from GPS, IMU, wheel odometry, and perception-based information. This pose estimate allows the robot to understand where it is relative to the course, obstacles, and target waypoints.

Using the localization data and real-time sensor observations, the mapping system builds a representation of nearby free space and obstacles. The path planner then uses this map to generate safe trajectories toward the next waypoint while avoiding cones, lane boundaries, and other obstacles. Once a path is selected, the control system converts the planned trajectory into drivetrain commands for the motor controller while respecting acceleration limits, steering behavior, and safety requirements.

Together, these systems allow the vehicle to move autonomously by continuously estimating its position, updating its surroundings, planning a safe route, and sending motion commands to the drivetrain.

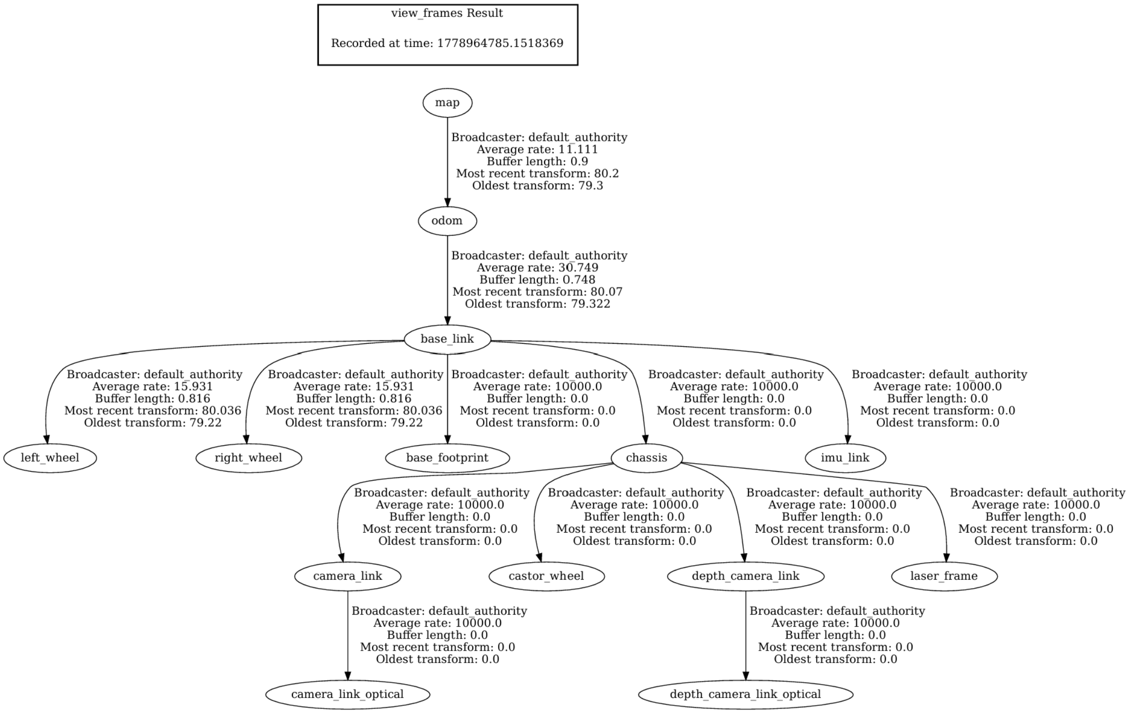


Figure 4: TF2 transform tree showing the full frame hierarchy from `map` through `odom` and `base_link` to all sensor frames, including `camera_link`, `depth_camera_link`, `laser_frame`, and wheel frames.

### 4.5 State Management and Logging

The robot software includes a centralized state-management infrastructure coordinating subsystem behavior and enforcing safe operating conditions.

Global operating states include Disabled, Manual Control, Autonomous, and Fault / Shutdown.

Extensive logging infrastructure records sensor data, localization estimates, drivetrain states, perception outputs, and debugging telemetry for post-run analysis.

### 4.6 Simulation and Debugging

Simulation is heavily utilized throughout development using Gazebo Sim and ROS2 Control integration.

The simulation environment includes:

- Simulated LiDAR
- RGB cameras
- Depth cameras
- IMU simulation

- Occupancy grids
- Obstacle environments

RViz2 is used extensively for debugging localization, mapping, perception outputs, and autonomous behavior. The Gazebo world features a curved white-line lane with traffic cones and block obstacles placed within the lane boundaries, providing a representative test environment for perception and navigation validation.

## 5 Vehicle Analysis

### 5.1 Performance Metrics

| **Metric** | **Target** | **Measured Value** |
|---|---|---|
| Maximum Speed | >1 MPH | 1.2 MPH |
| Battery Runtime | >10 min | 1.1 hours |
| Obstacle Detection Range | >2 meters | 12 meters |
| Emergency Stop Range | >150 ft | >450 ft |
| Emergency Stop Response Time | <100 ms | 50 ms |
| Localization Accuracy | <20 cm | 14 cm |
| Ramp Completion | Successful | Successful |

Table 2: Scarlet Rover performance metrics.

### 5.2 Failure Points and Resolutions

| **Category** | **Failure Point** | **Mitigation** |
|---|---|---|
| Mechanical | Loose fasteners from vibration | Use locking hardware and threadlocker. |
| Electrical | Short circuit or overcurrent event | Protected rails and eFuses. |
| Electrical | Voltage transient or reverse polarity | TVS protection and reverse polarity protection. |
| Software | Sensor dropout | Monitor node health and disable autonomous motion if required. |
| Software | Incorrect obstacle classification | Tune thresholds and validate under multiple environmental conditions. |

Table 3: Failure points and mitigation strategies.

# 6 Conclusion

The 2026 Rutgers IGVC vehicle represents a major improvement in systems integration, modularity, safety, and reliability compared to previous generations of the Rutgers platform. The robot combines a protected electrical architecture, modular ROS2 software stack, custom embedded hardware, and robust mechanical systems into a unified autonomous vehicle platform designed for reliable operation during the IGVC competition.