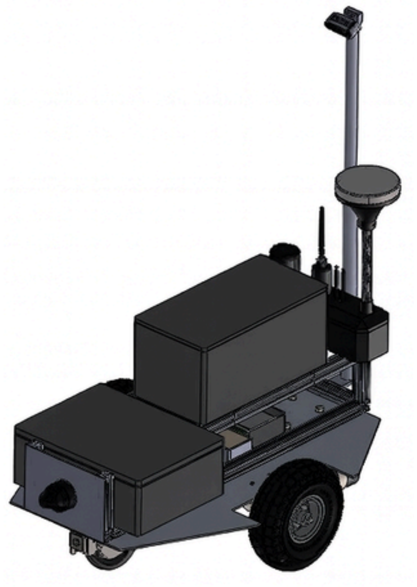
Karabuk University

EVRENOS IGVC 2026

Design Report

Auto-Nav Challenge



Statement of Integrity

I certify that the design and engineering of the vehicle (EVR-03) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.This report was prepared by the team EVRENOS under my guidance.

Prof. Dr. Raif BAYIR
Director of Karabuk University Robot Technologies Application and Research Center
rbayir@karabuk.edu.tr
+90 532 305 74 30

Software

Ali Haydar Duzenli
2110213071@ogrenci.karabuk.edu.tr
Ismail Samurkas
2210206267@ogrenci.karabuk.edu.tr

Mechanical

Emre Sermet
2110212010@ogrenci.karabuk.edu.tr
Ugur Unal
2110212002@ogrenci.karabuk.edu.tr

Electrical

Fahrettin Komurcu
2210219023@ogrenci.karabuk.edu.tr
Ulas Kapucuoglu
2010216003@ogrenci.karabuk.edu.tr
Orkun Furkan Aydilek
2410219051@ogrenci.karabuk.edu.tr

Shayan Yousefi Lame - Team Captain
2210216503@ogrenci.karabuk.edu.tr
sayan.yusefi@outlook.com
+90 536 994 07 19

February 25, 2026

Table of Contents

Introduction

The Karabuk University Evrenos UGV (Unmanned Ground Vehicle) Team is participating in the IGVC 2026 Auto-Nav Challenge with our autonomous ground vehicle, 'EVR-03'. Our team brings together the expertise of undergraduate students from various engineering disciplines, focusing on autonomous technologies. Our vehicle has been specifically designed and manufactured to perfectly meet the demands of the IGVC AutoNav course. Alongside its autonomous capabilities, the principles of maximum functionality, simplicity, and durability have been prioritized in its mechanical and electronic architectures. The vehicle’s autonomous software architecture is built upon the ROS2 system, providing a modular and reliable framework. For environmental awareness, lane tracking is performed via an Intel RealSense camera, while obstacles are detected using a 3D LiDAR sensor; these data streams are fused to generate a safe navigation route. Precise localization of the vehicle on the course is ensured through RTK-GPS support. This integrated system has been optimized to enable the vehicle to navigate flawlessly under IGVC’s open-field conditions, and extensive open-field testing has been conducted under similar simulated environments provided by our team.

Team Assignment Table

Embedded Design	Mechanical Design and Manufacturing	Autonomous Software development
Shayan Yousefi Ulas Kapucuoglu	Emre Sermet Ugur Unal	Ismail Samurkas Ali Haydar Duzenli
Embedded Software		
Orkun Furkan Aydilek Fahrettin Komurcu		

1. System and Subsystem Requirements

1.1 System Engineering Process

As the Evrenos Team, we developed our vehicle for the IGVC Auto-Nav competition using the V-Model Systems Engineering approach. Our primary objective was to ensure that every capability of the vehicle is directly derived from the competition requirements, while also designing it for applications that extend beyond the scope of the competition, making it an easily integrable platform for fields such as transportation and security. To enhance maneuverability within the course dimensions, we have integrated a compact version of the software stack from our military UGV platform, EVR-01 which has 12 months of field-tested reliability. This heritage allows for more precise control and functional adaptability on the track. In the initial phase, we identified priority critical requirements, such as safety and autonomous systems, by taking the IGVC 2026 rules into account. Before proceeding to physical assembly, we tested our sensor placements and algorithms within the Gazebo environment, specifically by simulating the optical characteristics of the course. During the implementation stage, we designed our software infrastructure to be compatible with the ROS2 Humble distribution to ensure modularity. Finally, we verified compliance with the Auto-Nav regulations by subjecting the system first to component-level testing and then to full-scale field tests on a custom test track consisting of asphalt surfacing, white lines, and obstacles-constructed by our team.

V1

V2

V3

1.2 System Requirements

Mechanical

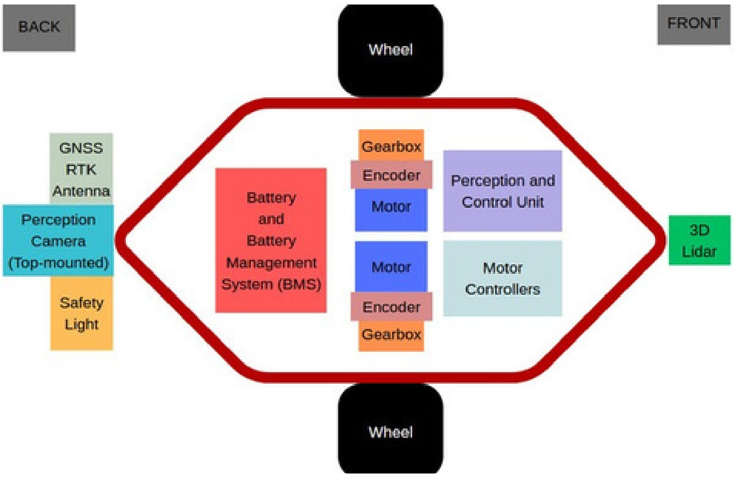
The mechanical drive system must be engineered to navigate course inclines, carry the required payload, and perform skid-steer maneuvers while maintaining a velocity near the maximum speed limit of 5 MPH. Accordingly, theoretical motor torque calculations based on motor power must be conducted and subsequently verified using a motor test bench to ensure empirical accuracy. Another critical requirement is the design of a skid-steer vehicle architecture. Therefore, the vehicle must be designed to comply with the minimum dimensions outlined in the specifications, while compactly housing all electronic components. No extraneous space should be left within the chassis, with the exception of the designated payload area. Such void spaces would unnecessarily increase the vehicle's overall dimensions, thereby hindering the effective operation of the mid-wheel differential drive system.

Safety

The safety system must be capable of monitoring potential overheating and fire hazards via NTC thermistors during battery charging and discharging cycles. Furthermore, to prevent accidents resulting from control system failures or environmental factors, the vehicle must support E-STOP (Emergency Stop) activation via on-board manual switches, wireless remote control, and the ground control station (GCS). The system must also incorporate a fail-safe mechanism to ensure protection during communication loss. Additionally, the vehicle indicates its current operational status distinguishing between manual and autonomous modes through an integrated light signaling system. This provides external situational awareness, allowing observers to identify potential navigation errors and intervene safely.

Electronic

The electrical and electronic system's primary requirement is to possess a communication infrastructure capable of supporting both autonomous and manual driving modes. Secondly, the motor control system must accurately manage speed and directional data, with verification provided through encoder feedback. Our validation plan involves monitoring the communication system for potential data loss, while the motor control system will be assessed by comparing real-time encoder data against target velocity values. Based on these control tests, the objective is to ensure uninterrupted communication and to maintain motor speeds within a tolerance range of ±5%.



1.2.1 Requirements Guiding the Design

To achieve high-fidelity environmental perception, the integrated use of LiDAR and camera sensors is targeted. YOLO based object detection packages will be integrated to ensure high accuracy in obstacle and lane detection. Because the mid-wheel differential drive architecture allows for highly dynamic rotational movements, the perception pipeline is specifically optimized for high speed processing and minimal latency to ensure immediate system response. For localization precision, the system is designed to achieve an error margin of ±4 cm using RTK-GPS, which is planned to be verified through real time position validation on the test track.

Driving Logic

To ensure continuous vehicle motion, we aim to develop a Python-based control and state machine architecture compatible with the Nav2 framework. To reliably track global waypoints and execute rapid maneuvers in the presence of unexpected obstacles, we have adopted an approach focused on optimizing local path planning parameters and performing real-time evaluation of sensor outputs.

Key Performance Indicators

We plan to evaluate vehicle performance using key metrics, including velocity stability, GPS accuracy, and operational runtime. To maintain a velocity deviation of less than 5%, ensure GPS localization precision within ±4 cm, and achieve an average operational duration of 480 minutes, our objectives focus on optimizing control algorithms and energy management systems.

2. Mechanical Design

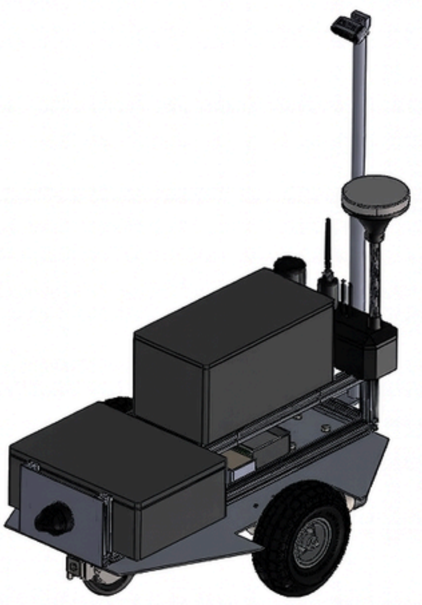
2.1 Mechanical Design Principle

The mechanical system is engineered to meet IGVC requirements with a minimal spatial footprint, prioritizing rapid maneuverability and energy efficiency. The architecture focuses on maximum functionality through a lightweight yet robust design, ensuring the vehicle can carry a payload nearly equal to its own weight. Designed for versatility and logistical ease, the system is highly modular, allowing for full assembly or disassembly within one hour. Every component, from the chassis frame to the motor specifications, has been meticulously selected to ensure structural durability and high performance across various operational terrains.

2.2 Mechanical Subsystems

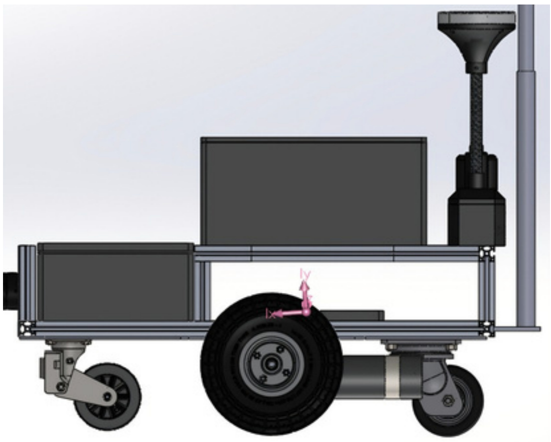
2.2.1 Chassis

In the chassis design, a 5mm aluminum plate has been selected as the primary base structure for mounting the motors, the corresponding wheels, the motor drivers and the battery pack. Fabricating a single-piece base plate was deemed the optimal choice to maximize structural strength and integrity. Crucially, the overall footprint of this base plate employs a diamond geometry rather than a traditional rectangular shape. This diamond configuration significantly enhances the vehicle's maneuverability; the tapered corners drastically reduce the vehicle's sweep radius during turns, allowing it to navigate tightly spaced obstacles and avoid corner collisions much more effectively than a standard rectangular chassis. Furthermore, precision-machined mounting holes for the motors and various components were incorporated to ensure high accuracy and alignment during mechanical assembly. To prevent any flex in this base plate and to provide secure mounting points for the enclosures housing the electronic components and the required cinder block payload, a two-tiered frame was constructed using 20x20mm aluminum extrusions. This highly modular architecture grants critical operational flexibility to the electronics, software, and mechanical sub-teams; by simply removing a few bolts, the entire upper frame along with all electronic components can be swiftly detached from the motor and drivetrain assembly.



2.2 Mechanical Subsystems

The vehicle's drivetrain utilizes two Motion Tech EC series 200W wheelchair motors (3800 RPM base speed with a 32:1 reduction ratio). The selection of these motors was mathematically validated to ensure the estimated 55 kg vehicle can seamlessly conquer the maximum 15% course inclines. Overcoming the combined gravitational pull and rolling resistance on a 15% gradient requires a total tractive force of approximately 96 N. With 30 cm diameter wheels , this translates to a required torque of 7.2 Nm per drive wheel. Given the motor's geared output speed of approximately 118 RPM, each motor is capable of delivering up to 16.1 Nm of continuous theoretical torque. This provides a safety factor greater than 2.0, guaranteeing unimpeded mobility on inclines. Given the mid-wheel differential architecture, the motors are mounted as low and perfectly centralized as possible to optimize the center of gravity. Furthermore, spring-loaded front and rear casters maintain continuous drive-wheel traction on uneven terrain while significantly dampening camera tower vibrations for stable perception. As illustrated below, the rear caster and central wheels are spaced equidistantly for optimal kinematic stability, whereas the front caster primarily functions as a dynamic shock-absorbing mechanism. For the wheels, 30 cm wheels with a crowned (rounded) tread profile were selected. This specific profile was intentionally chosen to minimize lateral scrubbing friction during skid-steer maneuvers. Given the high-friction nature of the asphalt course, reducing the wheel's contact patch is critical to significantly lower the stress and high torque requirements on the motors. This design ensures agile rotational maneuvers without compromising the excellent forward traction inherent to the asphalt surface, particularly on course inclines. To ensure a robust and highly precise mechanical transmission, a custom CNC-machined wheel hub was designed. The shaft of this fabricated hub is securely connected to the motor's keyed shaft, ensuring a positive mechanical engagement that prevents rotational slippage under high torque loads.



2.3 Target vs. Measured Value Analysis

A critical design objective was to prevent the lower motor assemblies from bottoming out on course inclines, as well as to mitigate excessive upward pitching of the front chassis to avoid severe frontal impacts during descents. The conducted kinematic analyses have successfully validated that the vehicle experiences no ground interference, ensuring smooth and unimpeded mobility across varying gradients.

3. Safety

To ensure safety during the transport, parking, and charging phases of the robot, current measurement systems and NTC temperature sensors are actively utilized. Data obtained from these sensors are continuously analyzed to provide real-time monitoring of risks, such as overheating or battery instabilities, that may occur particularly during the charging process. During the operational phase, while the vehicle is navigating the course, the Emergency Stop (E-STOP) system remains active. This safety mechanism can be triggered from three distinct points: the physical E-STOP button mounted on the vehicle, the emergency switch on the remote control unit, and the system interface at the ground control station (GCS). This multichannel architecture enables the immediate disconnection of power or the transition to a failsafe mode via both mechanical and wireless communication, thereby maximizing course safety. The monitoring network, established with NTC temperature sensors to mitigate the risk of overheating and fire, automatically triggers the valve of the fire extinguisher via the emergency control board when a critical threshold is exceeded. This autonomous intervention system initiates the discharge of fire-suppressing fluid while simultaneously notifying the user through a visual alert on the system interface to ensure immediate situational awareness. Analyses conducted based on system requirements demonstrate that the target values of the safety mechanisms are in full alignment with the measured empirical data. For instance, the target latency values for the E-STOP signal transmission, measured in milliseconds, were successfully achieved during testing, confirming that the system provides instantaneous response during critical moments. Upon reviewing data such as the activation success rate of the fire suppression system and the margin of error for the temperature sensors, it is evident that the vehicle maintains operational safety even under challenging course conditions and is fully prepared for all identified risk scenarios.

4. Electrical/Electronic Design

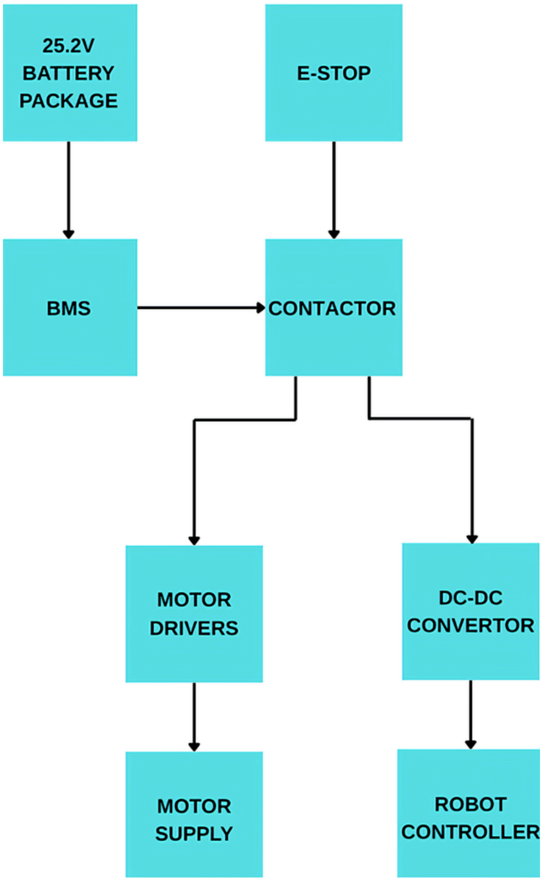
The designed vehicle is capable of both autonomous and manual operation. Its core electronic components are engineered to control the vehicle’s velocity and heading by processing sensor data and incoming commands. Integrated into this architecture, the safety system intervenes in the event of an anomaly by disconnecting system power, ensuring the most secure intervention possible.

4.1 Description of the Significant Power and Electronic Component

The vehicle’s electronic architecture is built on a sophisticated tri-tier processing system designed to balance autonomous intelligence, real-time telemetry, and precise motor control. At the core of the autonomous navigation, NVIDIA Jetson Orin Nano’s processes high-fidelity data from LiDAR and RTK-GPS sensors to execute path-planning algorithms. Parallel to this, a Raspberry Pi 5 manages the multimedia subsystem, capturing and transmitting real-time camera feeds to the ground station to facilitate low-latency manual operation. Acting as the critical bridge between these high-level processors and the physical actuators, an STM32 microcontroller serves as the primary control unit. It aggregates directional commands from both the autonomous algorithms and manual remote-control inputs, translating them into precise signals for the motor drivers. Furthermore, the STM32 maintains a closed-loop control system by utilizing encoder feedback, ensuring accurate speed regulation and high-stability movement during all operational modes.

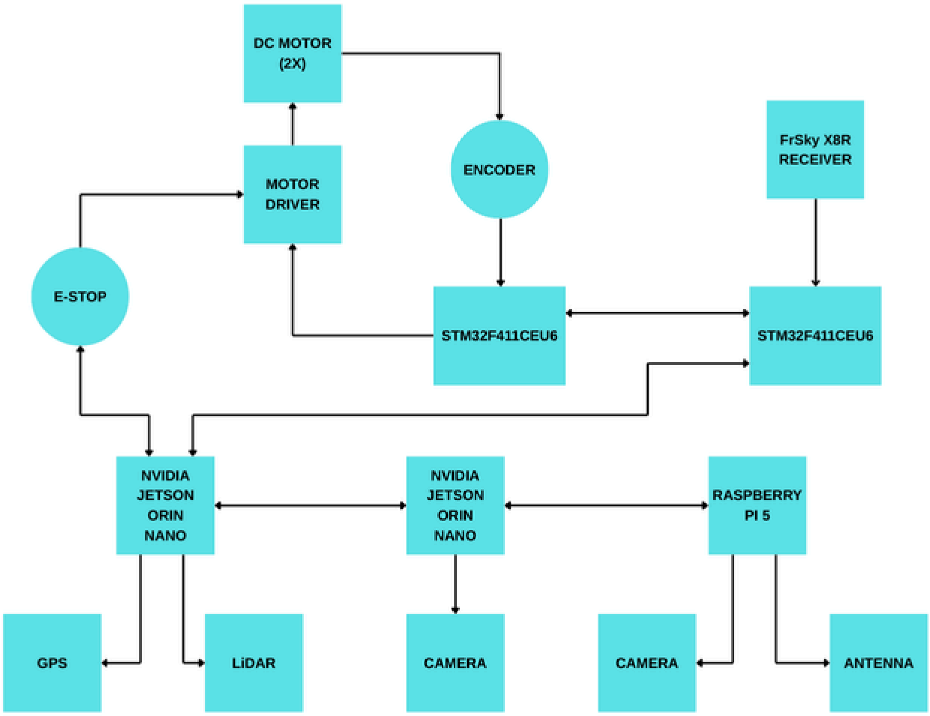
4.1.2 Power Distribution

The power for the vehicle's motors is supplied by the 25.2V main battery pack. This battery pack, consisting of 6 series and 10 parallel Li-ion cells, can provide 140A continuously and 250A peak (instantaneous) current. It has a discharge capacity of 28Ah and is balanced via a passive BMS. Considering this capacity and the power consumption of the motors, the vehicle can travel for 480 minutes without stopping at an average speed. To reach the 28Ah charge capacity, the main battery pack must be charged for 130 minutes with a specially designed 25.2V 12A charger. Other than the battery pack feeds all electronic components and sensors with the help of DC-DC converters. The battery pack is connected to the system via contactors, and in a potential emergency scenario, the power can be cut off by a command from the E-STOP board.



4.1.3 Electronics Suite

The system incorporates specialized electronic components to support both autonomous and manual operational modes. By employing a multi-processor architecture, computational tasks are effectively distributed. This approach significantly enhances both system stability and real-time execution capabilities. The utilized components are illustrated in detail in the schematic diagram below.



The system architecture integrates two NVIDIA Jetson Orin Nano and one Raspberry Pi 5 onboard computers, which exchange data via the USB protocol. In addition to these processors, two STM32F411CEU6 microcontrollers are utilized, communicating with each other through an I2C interface in a master-slave configuration. To bridge the communication between the processing units and the microcontrollers, the master STM32F411CEU6 is interfaced with the NVIDIA Jetson Orin Nano using the USB protocol. The Raspberry Pi 5 is equipped with a Logitech C920 camera sensor to transmit the vehicle's front-view imagery to the utilized interface. Additionally, it facilitates communication with the ground station interface through an integrated antenna system. Two NVIDIA Jetson Orin Nano units are integrated into the system for deployment in autonomous operations. These onboard computers have been preferred for their high AI processing capabilities and advanced GPU architecture, which allow for the real-time execution of complex computer vision and sensor fusion algorithms with optimized power consumption. To facilitate image processing, one of these units is interfaced with a Intel RealSense camera sensor. The second unit is connected to an Emlid Reach M+ RTK-GPS sensor and a Unitree 4D LiDAR sensor to enable the execution of ROS2 algorithms. Furthermore, the Jetson Orin Nano is designated to monitor and manage E-STOP safety protocols while the system operates in autonomous mode. The system contains two STM32F411CEU6 microcontrollers. This microcontroller has been preferred for its high dynamic efficiency and hardware-based FPU (Floating Point Unit), which allow for the precise execution of complex motor control algorithms with optimized power consumption. The master STM32F411CEU6 microcontroller in the system is utilized for two primary functions. One of these is to enable manual control of the system. For this purpose, it is interfaced with the FrSky X8R receiver of the FrSky Taranis X9D Plus transmitter. Its second function is to interpret motor control commands received from the NVIDIA Jetson Orin Nano onboard computer during autonomous operations and transmit them to the slave STM32F411CEU6 microcontroller. The slave STM32F411CEU6 microcontroller in the system controls the DC motors used to provide movement. PWM signals from the microcontroller are first sent to the motor drivers. Based on these incoming signals, the motor drivers control the DC motors to enable system movement. Additionally, by utilizing feedback from the encoders connected to the motors, this microcontroller can regulate motor speed via PWM signals within a closed-loop control system. The E-STOP system is a specially designed board. This board stops the movement of the motors in the system based on data sent from the Nvidia Jetson Orin Nano computer when operating in autonomous mode, and from the FrSky Taranis X9D Plus transmitter when operating in manual mode.

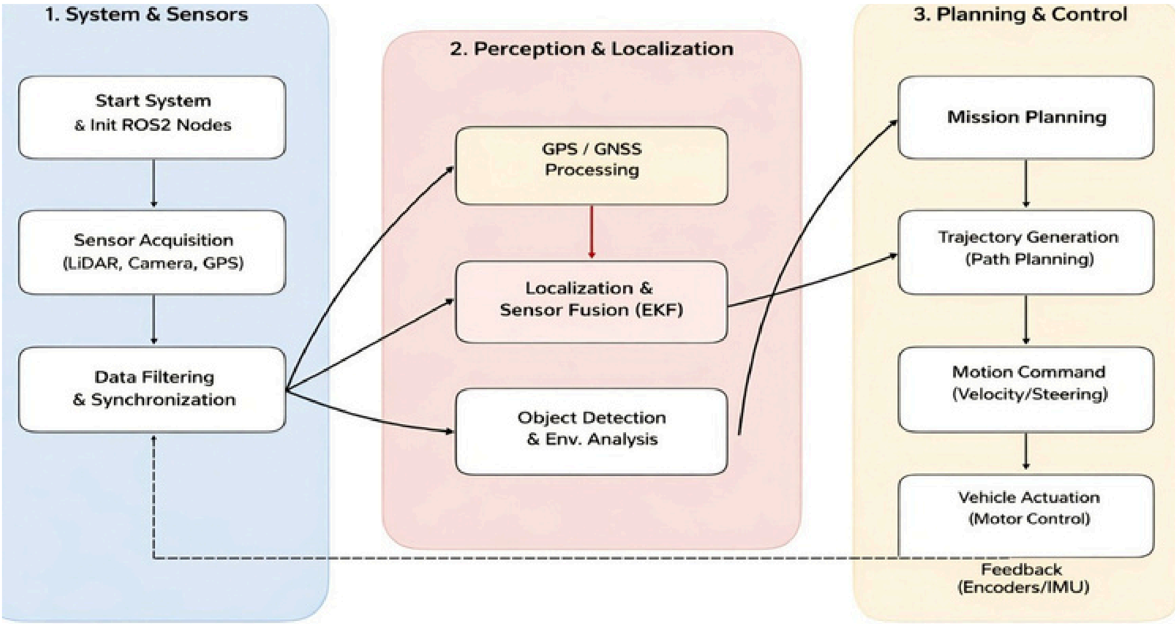
4.2 Target vs. Measured Value Analysis

For the first requirement, the communication infrastructure, uninterrupted USB communication between the Nvidia Jetson Orin Nano and the STM32F411CEU6, and error-ree I2C communication between the STM32 microcontrollers must operate. As a result of the measurements made, USB communication was established stably, and no data loss was observed on the I2C line. For the other requirement, the motor control system, the DC motors must reach the desired speed values, and the speed control must be verified with encoder data. It has been observed that motor speeds are regulated with a Fuzzy control algorithm using encoder data and operate within an approximate ±3–5% tolerance range of the target values. This ensures the vehicle can move safely during autonomous driving.

5. Perception

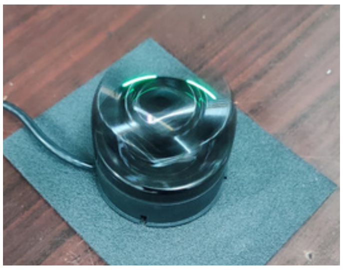
5.1 Software System Description

The autonomous vehicle software is implemented using ROS2 Humble and follows a modular node-based architecture. The system processes sensor data through localization, perception, navigation, and control layers to generate safe motion commands. Sensor inputs such as LiDAR, camera, GPS, and encoder data are fused using localization algorithms to determine the vehicle’s position. Processed environmental data is used for obstacle detection and path planning. Navigation algorithms generate motion commands, which are transmitted to low-level controllers to actuate the vehicle hardware. The overall system workflow is illustrated in the figure below.



5.2 Perception and Localization

Intel Real Sense camera



Unitree 4D LiDAR

The system is equipped with an advanced sensor suite to successfully complete the course in the AutoNav competition. This suite includes a 360° LiDAR for high-resolution 3D mapping, a Intel RealSense camera for visual perception and lane tracking, an IMU for orientation and acceleration estimation, and an RTK-supported GPS module for global positioning. Thanks to this combination of sensors, the vehicle can perceive its surroundings and determine its position within the course. In situations where GPS data is insufficient, odometry calculations utilizing both LiDAR and IMU data are activated. Thus, positional drifts are minimized. Working in integration with this sensor data, the Nav2 navigation infrastructure and the Python-based State Machine mentioned in section 6.3 Scripted Route System enable the vehicle to detect obstacles and generate alternative maneuvers while following waypoints.

5.2.1 Mapping and Navigation with Nav2

The vehicle's navigation system is built on the ROS2 Navigation (Nav2) stack. Since the vehicle operates in unknown and dynamic environments, it uses the SLAM (Simultaneous Localization and Mapping) method instead of a predefined map. The system creates a cost map using sensor data and performs path planning by selecting the lowest-cost trajectory using Dijkstra or A* algorithms. The LiDAR sensor detects physical obstacles such as barrels, while the camera detects lane markings and environmental boundaries. The system performance was validated through simulation and experimental testing as described in Section 8.2 Simulation Environment and Testing.

5.3 Teleoperation

Although the system is autonomous, this is the manual software component of the vehicle used by the team for testing, transport, and especially in emergency situations. The user sends speed and direction commands to the vehicle via the remote controller. These commands have the highest priority within the ROS2 network. Additionally, a hardware-based Emergency Stop (E-STOP) system is present, which operates completely independently of the software. When the E-STOP is triggered, the power to the motor drivers is physically cut off. During the IGVC Auto-Nav mission, the teleoperation software is completely disabled, and the vehicle executes all decisions and movements without any human intervention.

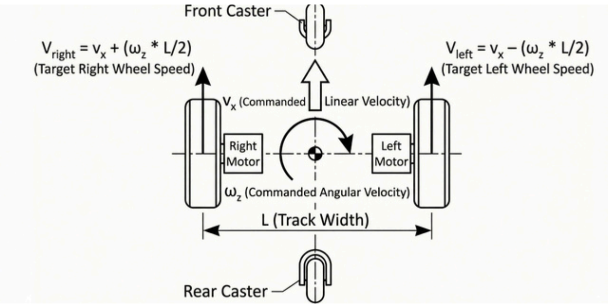
5.4 Ground Control Station (GCS) Interface

Whenever the system is operational, the custom-developed Ground Control Station (GCS) interface is actively utilized. Through this interface, the operator can simultaneously monitor the low-latency live video feed from the onboard Logitech C920 camera and track other critical telemetry data such as the vehicle's speed, battery levels, and real-time positioning from a single, unified screen.

6. Driving Logic

6.1 Drive by Wire Control

General movement commands (Twist messages: linear.x, angular.z) are received from Nav2 or the teleoperation system and converted into the physical movement of the vehicle. Since the vehicle does not have a steering mechanism, it utilizes a differential drive architecture. The propulsion system consists of two motorized main wheels positioned opposite each other on the center axis. The two passive caster wheels located at the front and rear are used solely for balance and load distribution and are not included in active drive control. Directional changes in the vehicle are achieved through the velocity difference between the right and left drive wheels. To convert the target robot velocities into individual wheel commands, the inverse kinematic model for a differential drive is given by in this context. In this context, Vr and Vl represent the target linear velocities for the right and left wheels respectively. The variable Vx represents the commanded linear velocity (derived from linear.x), Wz represents the commanded angular velocity (derived from angular.z), and L represents the track width, which is the distance between the two drive wheels. Both drive wheels are controlled by independent Fuzzy control loops using encoder feedback. The microcontroller maintains system stability by continuously calculating the error between the target velocity and the measured actual velocity. Fuzzy parameters were optimized using the incremental tuning method during experimental tests and simulation studies. To prevent sudden current draw and mechanical strain, velocity commands are applied using a trapezoidal (soft-start) velocity profile. This approach ensures that the vehicle performs controlled acceleration and deceleration.

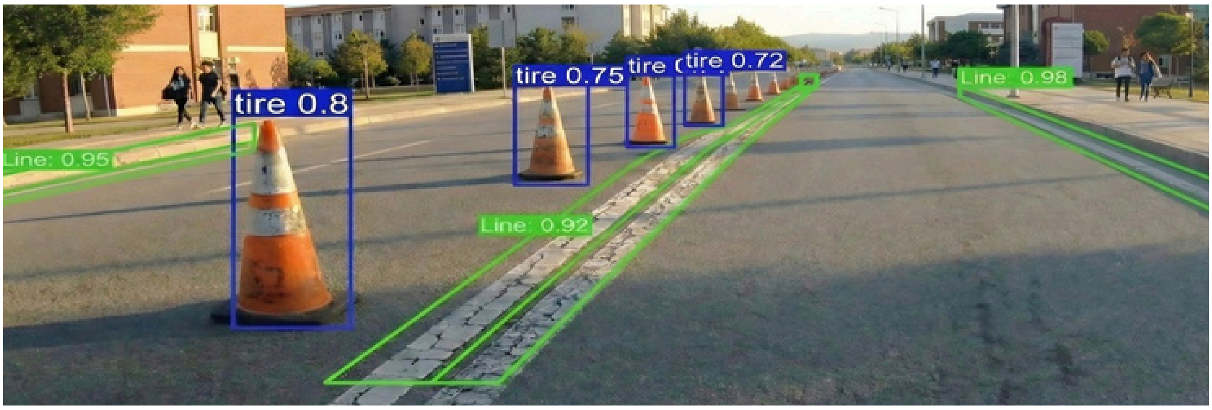


6.2 Object Detection and Classification

6.2.1 YOLO models introduction

The object detection module is built upon the YOLO v12 (You Only Look Once) architecture, which offers a balance of real-time performance and high accuracy. Fundamentally, the YOLO v12 architecture, which is a deep learning algorithm, has been utilized. The images in the dataset created by the team were first labeled, and then the fine-tuning process was performed using the Google Colab platform. Basically, there are 3 main classes in the dataset, and these classes are transmitted to the navigation layer located in ROS2.

1.Obstacles: Barrels, ramps, etc.
2.Lines: White lane lines on the asphalt.
3.Potholes: Circular markings located on the course.



6.3 Scripted Route System

A Python-based State Machine has been developed to manage the Auto-Nav course. This system operates on the ROS2 Humble and Nav2 infrastructure. While the vehicle relies on global waypoints using RTK-GNSS supported positioning under normal conditions, LiDAR and camera are used simultaneously for environmental perception and safety. While the vehicle proceeds towards the global waypoints determined by RTK-GNSS, the LiDAR sensor continuously updates Nav2's local costmap. In this way, even if the GPS signal is perfect, when the vehicle detects an obstacle on its route, it prioritizes LiDAR and camera data to perform an autonomous avoidance maneuver and realigns with the RTK route after clearing the obstacle.

7. Key Performance Indicators

7.1 Key Performance Indicators For Software

The software performance of the vehicle has been analyzed in 3 main stages:

1. Speed Stability & Limit Compliance:
The vehicle's continuous speed control is monitored via ROS2 log records and encoder feedback to ensure strict adherence to the competition's 5 MPH maximum speed limit. For a target cruising speed of 3.5 MPH, the acceptable deviation was set at ±15%. Real-time test results indicate that the vehicle typically maintains its speed within ±5% of the target. Occasional transient fluctuations up to ±7% were observed during sudden incline changes or sharp maneuvers. This performance guarantees stable progression and strict speed limit compliance without any loss of driving control.

2. GPS Navigation Accuracy:
The vehicle's accuracy in navigating to the 2 designated waypoints on the course is evaluated using log data from multiple test drives. The baseline requirement was a 90% success rate in reaching both coordinates within the acceptable tolerance radius. During the final physical tests, the vehicle successfully localized and navigated to both waypoints in 92% of the trial runs. The minor deviations observed in the remaining runs were largely due to transient GPS signal drops, which the system managed without critical failure.

3. System Uptime & Reliability:
System stability against software or hardware-induced anomalies during the mission is verified through comprehensive system logs. The team's target was a continuous, error-free operation rate of at least 90%. Over the course of the final endurance tests, the system achieved a 96% uptime. The remaining 4% accounted for minor, recoverable anomalies (such as brief ROS2 node timeouts) that did not lead to mission failure, further validating the overall robustness of the software architecture.

7.2 Key Performance Indicators For Electronics

The performance of the vehicle's electronic system has been evaluated considering communication reliability and motor control accuracy. Communication performance was measured by examining the accuracy of data packets sent between system components, while motor performance was analyzed by comparing encoder feedback data with target speed values. In the evaluations performed, the aim was to ensure no data loss during communication and for motor speeds to operate within a ±5% tolerance range. In the tests conducted, it was observed that these criteria were met, and it was concluded that the vehicle could offer a reliable and competitive performance on race courses.

7.3 Key Performance Indicators For Mechanics

The foremost Key Performance Indicator (KPI) of the vehicle is its mid-wheel differential drive architecture. This configuration provides a critical maneuverability advantage within the tightly spaced and demanding environments of the IGVC course. In fact, achieving this superior mobility is the fundamental reason we transitioned to this specific architecture, moving away from the design of our initial test vehicle, EVR-01. The second most critical KPI is the hybrid chassis system, which integrates an upper frame constructed from aluminum extrusions onto the primary sheet metal base. This specific structural design effectively eliminates chassis flex while optimizing the spatial arrangement of all components. Crucially, this low-profile architecture allows us to maintain the center of gravity as low as possible, keeping it closely aligned with the wheel axis. Any vertical expansion or raising of the vehicle would inherently shift the center of gravity upwards, leading to a severe loss of dynamic stability and overall balance.

8. Analysis of Complete Vehicle

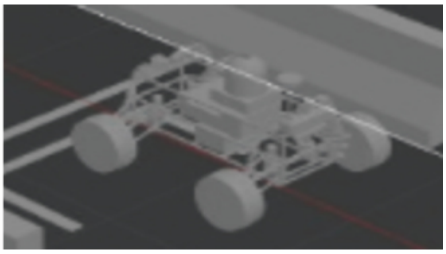
8.1 Mechanical and Electrical Components Failures

During the vertical obstacle tests of our initial prototype vehicle, which was specifically developed for software testing, the original axle system salvaged from an old automobile and adapted for the reducer shaft could not withstand the peak torque and shattered. Consequently, the mechanical team executed a complete redesign of the shaft system based on rigorous stress analyses, ensuring its capacity to endure the maximum torsional stresses generated by the drivetrain. The universal joints, functioning as critical intermediate transmission elements, were engineered to the 1G standard and precisely manufactured inhouse using a CNC router following custom CAM programming. Although this software testing vehicle has now accumulated approximately 150 hours of operational time, the newly fabricated joints and axle system have exhibited absolutely zero deformation. In addition to this, electrical noise generated in the motors could sometimes leak into the control circuit via the motor driver. As a solution, filters and optocouplers were added, and the suppression of the generated interference was tested first in simulation tools and then in real-life conditions.

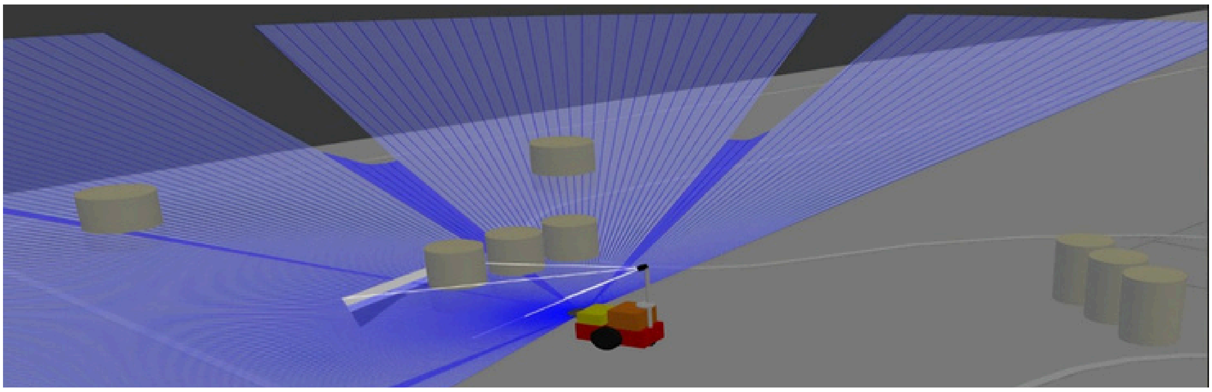
8.2 Simulation Environment and Testing

The simulation software utilized during the vehicle development process is the open-source Gazebo simulation environment, where real-world robot behaviors can be modeled with high fidelity. The software team has completed the simulation and validation of the vehicle's autonomous driving systems on this platform. Operating in integration with the ROS2 Humble distribution, Gazebo simulates both sensor data and vehicle kinematics. Throughout the project lifecycle, the hardware and software designs have evolved iteratively. Initially, a four-wheel-drive (4WD) vehicle was developed. Both the simulation tests (initial 4WD simulation) and the physical manufacturing (actual initial 4WD model) of this first iteration were successfully completed. Currently, the team is focusing on developing a new, smaller, and more compact vehicle. For this new iteration, the robot model in Gazebo was configured as a differential drive mobile platform featuring two centrally positioned drive wheels and two passive caster wheels (Gazebo simulation of the new vehicle). The differential drive controller plugin was utilized to accurately represent the motion characteristics of the new vehicle. For both iterations, a URDF file was used to define the robot's physical structure, wheel configuration, and sensor placements. Virtual data generated from the LiDAR, camera, and IMU sensors were configured to closely mimic real-world behaviors. This simulation infrastructure enabled comprehensive testing of the Nav2 navigation stack and SLAM algorithms under realistic operating conditions across various complex map scenarios. This approach allowed for the fine-tuning of the autonomous navigation system and the validation of sensor placements prior to deploying the system onto physical hardware.

(actual initial 4WD model)



(initial 4WD simulation)



(Gazebo simulation of the new vehicle)

8.2.1 Software Testing, Bug Tracking and Version Control Processes

All software development processes carried out for the system are hosted and managed on a GitHub organization. Collaborative coding with multiple participants was intended. During the project process, Docker containerization infrastructure was actively used to ensure development environment consistency and accelerate integration. This choice optimized collaboration by minimizing the installation processes for team members. In this way, time efficiency was prioritized. The Python-based State Machine and navigation algorithms were tested on the vehicle's URDF model in the Gazebo simulation environment before being deployed to the physical vehicle (NVIDIA Jetson Orin). As a precaution against system failures, the computer on the vehicle (Jetson Orin) is regularly backed up. Additionally, installation scripts that automatically configure ROS2 packages and libraries have been created.

8.3 Physical Test and Difference From Simulations

MOSFETs and resistors were used to balance the voltage of the cells in the passive balancing system designed for the vehicle's main battery pack. The discharge time of each parallel branch was analyzed to determine the optimum setting for battery longevity. Additionally, the topology designed to prevent voltage differences in each parallel cell from affecting the balancing system was simulated to verify its operation for all cells. Consequently, the designed passive balancing Battery Management System (BMS) operated without problems, first in the LTSpice simulation tool and then in real life. The autonomous driving software, validated in the simulation environment, was tested on the real vehicle. While our vehicle could stably perform waypoint tracking and navigation tasks with ideal sensor data in the Gazebo simulation, a deviation of approximately 4% was detected in the vehicle's positional accuracy in real tests due to environmental effects such as GNSS signal fluctuations, sensor noise, and ground conditions. Despite this deviation rate, the vehicle is able to perform route tracking safely.

9. Cyber Security Analysıs

Ensuring the security of our autonomous vehicle’s communication infrastructure is a top priority, as any vulnerability in data transmission could lead to a total system failure or unauthorized control. To achieve a resilient operation for the IGVC competition, our team conducted an in-depth cybersecurity analysis specifically focused on our Ubiquiti hardware stack and the TCP/IP protocol suite.

9.1 Telemetry and Signal Layer Security

For communication, the vehicle utilizes high-performance Ubiquiti Bullet M5 and PowerBeam M5 hardware. While these devices offer a stable 150+ Mbps bandwidth over a 15 km range, we recognized that such a powerful signal footprint significantly expands our physical attack surface. To manage this risk, we implemented an RF Containment strategy: we used the AirOS interface to intentionally dial down the Transmit Power (TX Power) to the minimum level required for the course. Additionally, we isolated the communication link by configuring it as a dedicated Point-to-Point (PtP) bridge. By enforcing WPA2-AES encryption and locking the topology to this PtP mode, we ensure all telemetry data is encrypted and unauthorized devices are physically blocked from even attempting to associate with our bridge.

9.2 Access Control and Authentication

Our analysis showed that simply hiding the network name (SSID cloaking) is not enough to stop a determined attacker from trying to force a connection to our PowerBeam M5 station. To prevent unauthorized entry, we strictly enforced WPA2-AES encryption paired with a high-entropy, complex passphrase. This creates a mandatory authentication layer that requires a secure "handshake" between the Bullet M5 (Client) and the PowerBeam M5 (Station). As a result, even if our signal is detected, no external device can establish a physical Layer 2 association without the correct credentials, effectively shielding our internal Local Area Network (LAN).

9.3 Middleware Security and ROS 2 Integrity

Moving to the software level, we evaluated the inherent vulnerabilities of the ROS2 middleware. Since the default DDS (Data Distribution Service) protocol broadcasts discovery data openly, the system is naturally prone to interference and manipulation by unauthorized actors. An intruder gaining network access could potentially hijack the communication to inject false velocity commands into /cmd_vel or feed noise into the /scan topic, leading to erratic vehicle behavior. To mitigate these middleware-level risks, we integrated the SROS 2 (Secure ROS) framework into our stack. Our approach involved generating unique X.509 certificates for every critical node to enforce authentication and encryption. To further harden the system, we isolated our traffic using a non-default ROS_DOMAIN_ID and customized our DDS XML profiles. These configurations allow us to restrict ROS2 communication strictly to the internal loopback interface and the designated Operator Control Unit (OCU) IP, which effectively blocks all unauthorized external multicast traffic.