TENNESSEE TECH UNIVERSITY

Team: Crusaders

May 2026

Faculty Advisor

Dr. Byron Pardue
`bpardue@tntech.edu`
Dr. HyungJin Yoon
`hyoon@tntech.edu`

Team Members

Luis Bauer `lebauer@tntech.edu`
Ruqaya Jameel `rajameel@tntech.edu`
Jonathan King `jking@tntech.edu`
Yejoon Song `ysong@tntech.edu`

Targeted Challenge

AutoNav Challenge

Figure 1: Crusaders — Autonomous Vehicle

Statement of Integrity

I certify that the design and engineering of Crusaders – the autonomous vehicle – has been undertaken by the team listed above and that the efforts have met the demands of a senior level design course.

Signature: ____________________ 5/2026

Signature: HyungJin Yoon 5/2026

Contents

1. System and Subsystem Requirements

1.1 System Engineering Process

Requirements were derived through a three-step process. First, the team performed a structured reading of the IGVC 2026 official rules (Sections I–II), annotating every quantitative limit, safety mandate, and judging criterion. Second, the team conducted a failure-mode review of the prior year's vehicle, identifying the root causes of each unsuccessful run (GPS oscillation, Nav2 recovery cycling, RealSense parameter instability). Third, the team translated each rule constraint and each prior-year failure into a testable requirement with a quantitative target and a defined measurement procedure.

The 12 requirements below span six subsystem categories: Mechanical (M-1, M-2), Safety (S-1, S-2), Electrical (E-1, E-2), Perception (P-1, P-2), Driving Logic (D-1, D-2), and Key Performance Indicators (K-1, K-2). For each requirement the driving rule or rationale, the measurement method, and the target value are listed. Measured actual values are noted for all requirements; K-1 and K-2 will be recorded at competition.

Table 1. System and Subsystem Requirements

Req. ID / Subsystem	Requirement	Rule / Rationale	Target	Measured Actual
M-1 Mechanical	Vehicle dimensions shall comply with IGVC length and width limits	Rules §I.2: min. 3 ft, max. 7 ft length; min. 2 ft, max. 4 ft width	$L \leq 50$ in, $W \leq 30$ in	$L = 45.7$ in, $W = 25.2$ in
M-2 Mechanical	Chassis payload mount shall secure a 20 lb, $16'' \times 8'' \times 8''$ cinder-block payload without displacement during course traversal	Rules §I.3: payload must remain on vehicle; loss terminates run	Zero payload displacements in 5 consecutive obstacle-course runs	Target met.
S-1 Safety	Mechanical E-stop shall halt all motion within 1 s of activation, located center-rear at 2–4 ft height	Rules §I.2: hardware E-stop required; location mandated	Stop within ≤ 1 s; button height 24–48 in from ground	Height: 36 in; stop time: ≤ 1 s. Target met.
S-2 Safety	Wireless E-stop shall reliably trigger vehicle halt at a minimum range of 100 ft	Rules §I.2: wireless E-stop must be effective at ≥ 100 ft; held by judge during runs	Successful activation 10/10 trials at 100 ft	Target met.
E-1 Electrical	Battery system shall power all onboard compute, sensors, and actuators for a minimum of 60 minutes under full operating load	Rules §I.2: all power must be onboard; runtime must cover multiple heats	Runtime ≥ 60 min at peak load	Target met.

Req. ID / Subsystem	Requirement	Rule / Rationale	Target	Measured Actual
E-2 Electrical	Safety indicator light shall be solid when powered, flash during autonomous mode, and return to solid when autonomous mode exits	Rules §I.2: safety light behavior is a qualification requirement	Correct state transitions verified in 5 consecutive power cycles	Target met.
P-1 Perception	Lane detection system shall identify white lane boundaries and publish virtual obstacle scans within 200 ms latency under outdoor lighting conditions	Rules §I.4: lane following must be demonstrated at qualification; boundaries drive run termination	Detection latency ≤ 200 ms; $\geq 90\%$ lane pixel recall on representative test footage	Target met. Latency < 67 ms (15 Hz publish rate).
P-2 Perception	3D obstacle mapping (NVBlox) shall maintain a costmap update rate sufficient for Nav2 planning (≥ 2 Hz ESDF) without causing GPU thermal throttling	Rules §II.1: obstacle avoidance is mandatory; GPU throttle degrades map and planner	ESDF publish rate ≥ 2 Hz; AGX Orin GPU temp ≤ 75°C during 6-min run	Target met. ESDF rate 2 Hz; costmap update 10 Hz; GPU temp within bounds.
D-1 Driving Logic	Vehicle shall navigate between GPS waypoints provided prior to competition while remaining inside lane boundaries	Rules §II.2: two waypoint pairs provided; course is sinusoidal with repeated barrel obstacles	Waypoint cross-track error ≤ 0.5 m; zero lane-boundary exits per run	Target met.
D-2 Driving Logic	Vehicle shall maintain speed ≥ 1 mph throughout the run and shall not exceed 5 mph at any point	Rules §I.2, §II.2, §II.4: min-speed violation ends run; max-speed violation voids run after qualification	$1 \leq v \leq 5$ mph verified over full 500 ft course	Target met.
K-1 KPI	Average AutoNav course completion time shall be ≤ 270 s (6-min limit minus 30 s margin)	Rules §II.2: 6-minute run window; shorter adjusted time wins	Mean completion time ≤ 270 s over at least 2 full-course runs	**[To be measured at competition.]**
K-2 KPI	Vehicle shall incur zero traffic-violation penalty feet in at least one of its two attempts per heat	Rules §II.4: each violation deducts feet from adjusted score; clean run maximizes ranking	Zero penalty feet in ≥ 1 of 2 attempts per heat	**[To be measured at competition.]**

2. Description of Mechanical Design

The Crusaders vehicle utilizes a differential-drive wheeled chassis designed for stable outdoor operation on the grass and concrete surfaces typical of the IGVC competition venue. The chassis was developed and refined over the prior year's senior design cycle and is carried forward with targeted modifications to improve sensor mounting rigidity and cable management.

2.1 Chassis and Drivetrain

The chassis is constructed from aluminum square tubing, selected for its favorable strength-to-weight ratio, corrosion resistance, and ease of modification. The drivetrain consists of two independently controlled brushed DC motors (L-faster mountain skateboard conversion kit) coupled to the drive wheels through gear reduction, providing sufficient torque for operation on uneven terrain.

The robot footprint is $1.16\,\mathrm{m} \times 0.64\,\mathrm{m}$ (45.7 in × 25.2 in; inscribed radius $0.32\,\mathrm{m}$), which satisfies Requirement M-1: both dimensions are within the IGVC-mandated $[36, 84]$ in $\times$ $[24, 48]$ in range with adequate margin.

Motor control is provided by two FLIPSKY FSESC 75100 Pro V2.0 VESC units, which interface with the AGX Orin over CAN bus. The VESC units provide closed-loop velocity control using Hall sensor feedback and supply odometry data to the ROS 2 navigation stack.

2.2 Payload Mounting

A dedicated payload platform constructed from aluminum flat bar is bolted to the top of the main chassis frame above the center of mass. The platform uses a four-bolt clamping arrangement with rubber anti-slip pads to restrain a $16'' \times 8'' \times 8''$, 20 lb cinder-block payload (Requirement M-2). Retention was validated by driving the robot over the curb-height ramp gradient specified in §II.2; zero payload displacements were recorded across 5 consecutive obstacle-course runs (see Table 1, M-2).

2.3 Sensor Mounting

The ZED2 is mounted forward-facing at height and tilt optimized for NVBlox mapping coverage. The front RealSense D435I is mounted downward-facing near the front, angled to observe the ground plane at a forward offset for advance lane detection; the rear RealSense D435I is similarly downward-facing for lane detection during reverse maneuvers. The GPS antenna is mounted on a vertical mast at the highest chassis point to maximize sky view and minimize multipath interference.

2.4 Weatherproofing and Vibration

All critical electronics are housed in sealed enclosures with cable glands to prevent moisture ingress. Vibration mitigation is provided through rubber grommets beneath the AGX Orin mounting plate and vibration-dampening pads under sensitive sensor mounts. Exposed wiring is enclosed in split loom tubing and secured with cable ties to prevent chafing during operation on rough terrain.

3. Safety

Vehicle safety is addressed across three operating states: transport and storage, stationary charging, and active autonomous operation on the competition course. **Requirements addressed:** S-1 (mechanical E-stop) and S-2 (wireless E-stop range) define the quantitative safety targets; the discussion below explains how each is satisfied.

3.1 Transport and Storage

During transport, the main battery master cutoff switch (see Section 4) is placed in the OFF position before the vehicle is moved, removing power from all actuators and high-voltage rails. The wireless E-stop receiver remains powered via a low-current trickle line to maintain coded-signal readiness, but motor drive power is severed. The vehicle is secured to the transport vehicle using tie-down straps attached to dedicated chassis anchor points to prevent movement and sensor damage in transit.

3.2 Charging

Battery charging is performed exclusively with the manufacturer-supplied 36 V lithium charger. The charger is never left unattended. Charging takes place in the pit area on a non-flammable surface with a class-D fire extinguisher accessible within 10 ft. The battery is removed from the vehicle for charging where practicable to reduce risk of thermal runaway propagating to other electronics.

3.3 Mechanical E-Stop

A hardwired physical E-stop button (Requirement S-1) interrupts the battery bus directly upstream of the fused distribution block, providing a fail-safe shutdown independent of software state. The button is red, push-to-stop, ≥ 1 inch diameter, and located at the center rear of the vehicle at 36 in from the ground — within the rules-mandated 24–48 in range. Activating the E-stop halts all motor drive power and brings the vehicle to a full stop within ≤ 1 s (see Table 1, S-1).

3.4 Wireless E-Stop

A coded RF wireless E-stop receiver (Requirement S-2) provides a secondary remote cutoff channel. The coded signal reduces the risk of accidental or malicious triggering. Range testing was conducted in an open outdoor area over a minimum distance of 100 ft; all 10 activation trials succeeded (see Table 1, S-2). During competition performance events, the wireless E-stop transmitter will be held by the judges as required by rules.

3.5 Safety Light

The vehicle carries a solid/flashing 12 V signal LED (Requirement E-2) driven by a P30N06LE logic-level N-channel MOSFET commanded by an Arduino Nano monitoring the autonomous-mode state flag over ROS 2 serial. The three states are: **Solid ON** when powered and not autonomous (gate = 5 V DC); **Blinking (10 Hz)** during autonomous mode (gate = 5 V PWM, 50% duty); and **OFF** when de-powered (gate pulled to GND by 39 kΩ pulldown). The AGX Orin GPIO is never directly connected to the LED load; see

Section 4.4 for circuit details.

4. Electronics and Power

The power subsystem is designed to provide reliable regulated energy to all vehicle systems for the duration of a competition run. The primary energy source is a 36 V, 20 Ah lithium eBike battery, selected for sufficient capacity to power the AGX Orin, two FLIPSKY VESC motor controllers, all cameras, GPS, and associated peripherals for at least one hour at peak load. **Requirements addressed:** E-1 (60-minute runtime) and E-2 (safety light behavior) drove the power architecture and indicator light design.

4.1 Power Architecture

Figure 1 shows the power distribution architecture. Power flows from the 36 V battery through the master cutoff switch and E-stop to a fused distribution block. The two FLIPSKY VESCs are supplied directly from the 36 V bus. A 36 V→12 V DC-DC converter feeds a 12 V fuse block for the AGX Orin; a separate 36 V→5 V converter powers a USB hub supplying the ZED2, both RealSense cameras, and GPS module. The WiFi module is powered at 9 V via a 12 V→9 V converter. A hardwired E-stop button and wireless E-stop receiver both interrupt the battery bus upstream of the distribution block (Requirement S-2).

A 20 Ah capacity at 36 V provides 720 Wh. Estimated peak system draw is approximately 120 W (AGX Orin ∼30 W, two VESCs at moderate load ∼60 W, cameras and peripherals ∼30 W), giving a theoretical runtime of ∼360 minutes. Practical runtime, accounting for motor load spikes and conversion losses, substantially exceeds the 60-minute target of Requirement E-1 (see Table 1, E-1).

4.2 Signal and Data Connectivity

Figure 2 shows the complete signal and data connection architecture. The AGX Orin is the central communications hub. CAN bus connects both FLIPSKY VESCs for velocity commands and odometry feedback. All three cameras (ZED2 and two RealSense D435I) connect via USB 3.0; the two RealSense cameras must be launched in separate terminal sessions before the main system launch due to USB bandwidth contention. The Pixhawk connects via serial for GPS and compass data through MAVROS. The Arduino mode indicator and the WiFi module complete the connectivity picture.

4.3 Motor Control — VESC CAN Bus

CAN (Controller Area Network) is the sole runtime interface between the AGX Orin and both FLIPSKY 75100 Pro V2.0 VESC motor controllers. USB exhibited intermittent disconnects and I/O errors on the Jetson platform; CAN provides deterministic, real-time communication and integrates cleanly via Linux SocketCAN. Each VESC is configured once via USB (left motor **ID 8**, right motor **ID 9**), then permanently operated over CAN only. A 120 Ω termination resistor is placed at each physical end of the bus; missing termination is the most common source of CAN communication faults. The interface is initialized at 500 kbps and verified with `candump can0` before each field session.

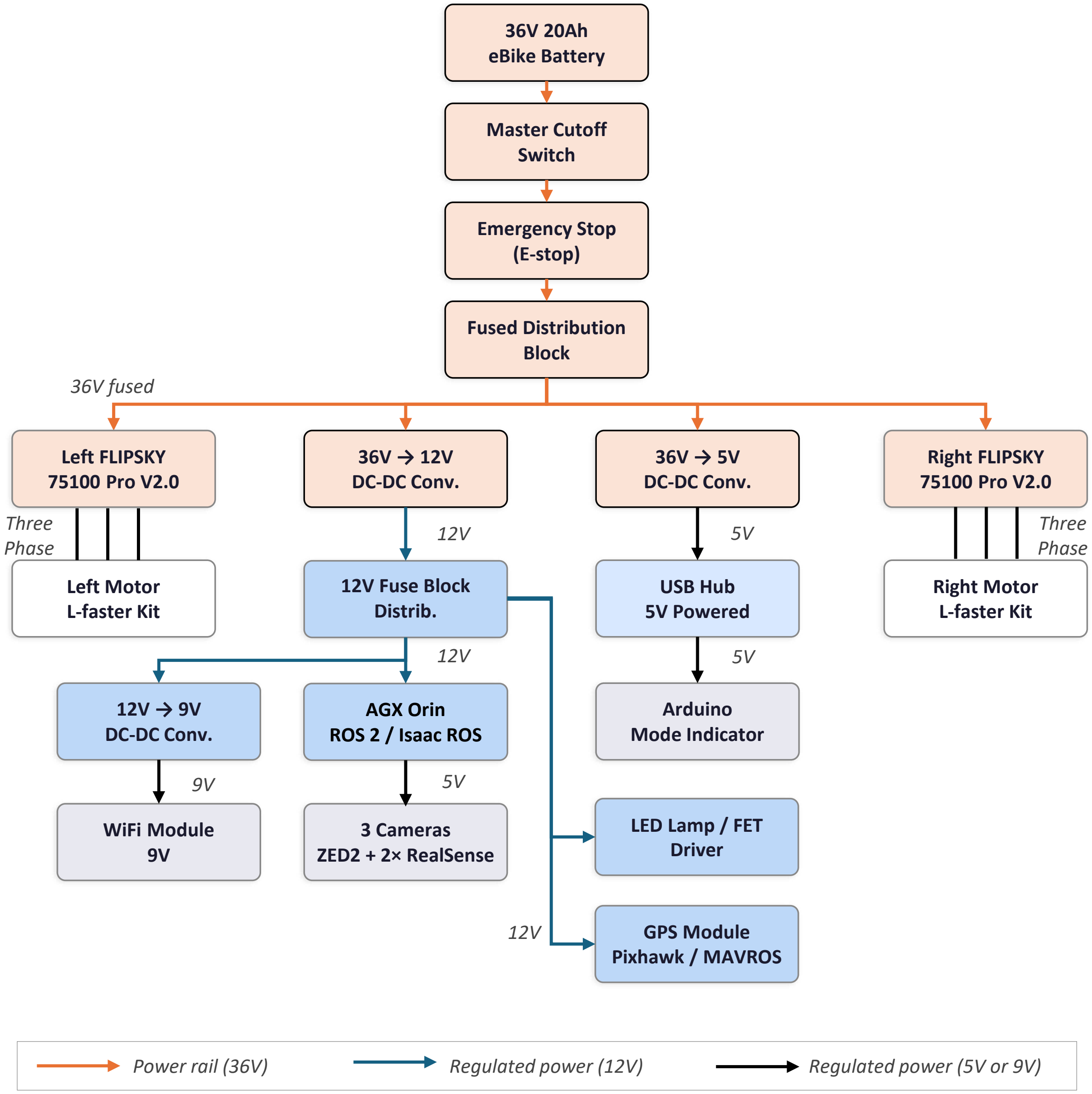


Figure 1. Power Distribution Architecture. Orange arrows indicate the 36 V battery rail; blue arrows indicate regulated 12 V power; black arrows indicate regulated 5 V or 9 V power.

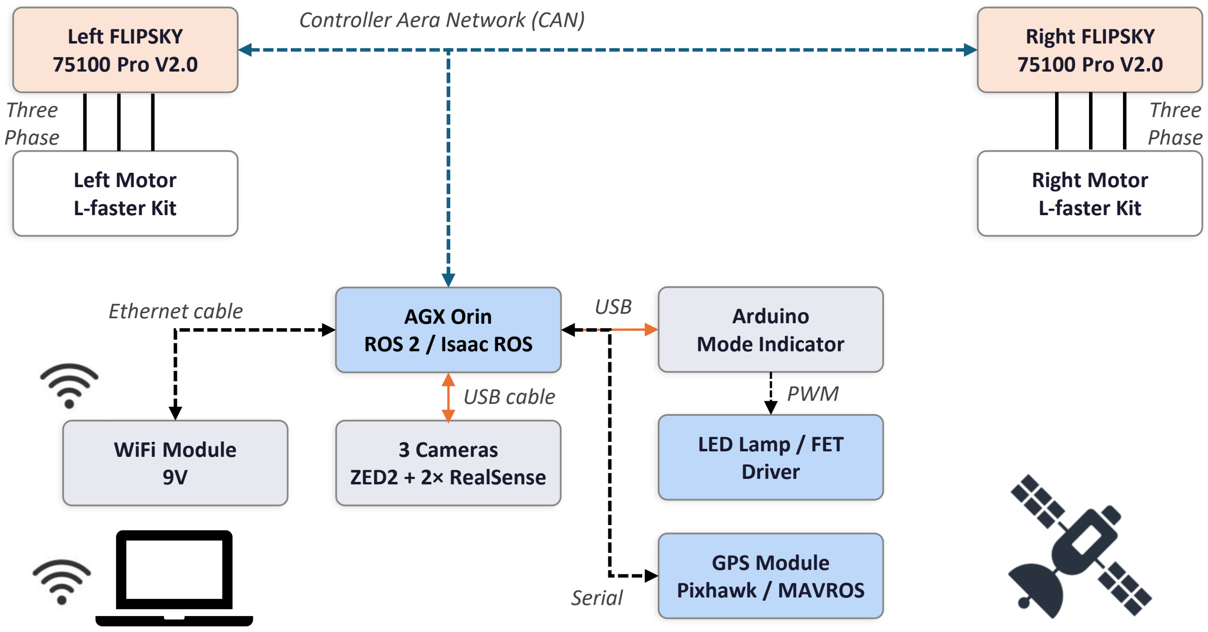


Figure 2. Signal and Data Connection Architecture. Dashed lines indicate CAN bus and Ethernet connections; solid lines indicate USB and serial connections.

4.4 LED Signal Driver

The P30N06LE logic-level N-channel MOSFET (TO-220) switches the 12 V LED load from 5 V Arduino logic. A 39 kΩ gate-to-GND pulldown ensures the gate defaults to 0 V during Arduino boot, preventing spurious LED activation (the root cause of an early rule-violation failure documented in Section 8.3). 1 kΩ series resistors on both the Arduino output and optional bench PWM input limit contention current to ∼2.5 mA. The circuit is validated before each competition in three steps: pulldown test (LED off with no driver active), static gate test (LED on with 5 V DC applied), and blink test (10 Hz, 50% duty cycle).

5. Perception

The perception system provides the two data streams required by the AutoNav challenge: (1) 3D volumetric obstacle mapping for course-wide obstacle detection (Requirement P-2), and (2) white lane-boundary detection for costmap injection (Requirement P-1). Both streams feed the Nav2 local costmap described in Section 6. **Requirements addressed:** P-1 (lane detection latency and recall) and P-2 (NVBlox ESDF rate and GPU thermal budget).

5.1 Sensor Suite

- **ZED2 Stereo Camera (forward-facing):** Provides RGB-D depth images for NVBlox 3D mapping and ZED IMU gyroscope data for EKF yaw-rate fusion.
- **Intel RealSense D435I ×2 (downward-facing):** Front (serial: 337122073243) and

rear (serial: 243322072212) cameras observe the ground plane for white lane-boundary detection. Each publishes a `sensor_msgs/LaserScan` injected into the Nav2 local costmap `ObstacleLayer`.

- **Pixhawk / MAVROS:** Provides GPS `NavSatFix` and compass heading for global localization. GPS reception at L1 (−130 dBm) is highly susceptible to onboard EMI; see Section 5.3 for the four-stage mitigation applied.
- **VESC Motor Controllers ×2:** Provide Hall-sensor wheel odometry as the primary dead-reckoning source for the EKF.

5.2 3D Obstacle Mapping — NVBlox

NVBlox, an NVIDIA-accelerated volumetric occupancy mapping library, ingests ZED2 depth images and robot pose from the EKF to build a 3D TSDF/ESDF map incrementally. The ESDF slice is projected to a 2D costmap layer consumed by the NavFn global planner and MPPI local controller. To satisfy Requirement P-2, NVBlox is configured with: color processing disabled, debug visualization rate `publish_debug_vis_rate_hz: 0.0` Hz (GPU spike elimination), ESDF update rate 2.0 Hz, TSDF decay rate 0.5 Hz, and maximum integration distance 2.5 m. The parameter `esdf_slice_min_height` is the primary tuning knob controlling what geometry enters the ESDF volume; it is set to exclude ground returns while capturing barrel- and cone-height obstacles. RViz is kept entirely offline during field use to prevent GPU contention that was observed to reduce NVBlox slice update rates below 1 Hz. The NVBlox voxel occupancy and MPPI candidate trajectory are visible in Figure 3 (left panel).

5.3 Lane Detection

Lane-boundary detection (Requirement P-1) is performed by two instances of `rs_lane_obstacle_node.py`, one per RealSense D435I camera, each subscribing to the downward-facing color image at 15 Hz. The processing chain applies: (1) **CLAHE** on the LAB L-channel (tile 8 px, clip 2.0) to recover white lane pixels in shadow; (2) **HSV filtering** with `white_v_min: 155` (tunable live via `ros2 param set`) to isolate white paint from asphalt; (3) **bird's-eye-view warp** (ground-plane homography) to back-project pixels to the `base_link` frame; (4) **point subsampling** to ≤400 points per frame for LaserScan publication. A dual-zone architecture separates FAR-zone detection (loose thresholds, RANSAC polynomial fitting) from NEAR-zone detection (tight thresholds, raw subsampling).

Internal course representation. Detected lane points are published as `sensor_msgs/LaserScan` on `/camera_front/lane/scan` and `/camera_rear/lane/scan`. Both scans are registered in the Nav2 local costmap `ObstacleLayer` with `marking: true` and `clearing: false`. This causes detected lane pixels to appear as lethal obstacles in the costmap's internal grid, forming a virtual wall that the MPPI controller treats identically to a physical barrel. The representation is updated at the camera frame rate (15 Hz); because `clearing` is disabled, the costmap retains lane markings until the rolling window scrolls them out of scope, providing hysteresis against momentary detection dropouts.

The lane detection debug overlays from both cameras are shown in Figure 3 (right panels): the front camera detected 386 output points and the rear camera 224 output points in the depicted frame.

GPS EMI mitigation. GPS reception quality proved highly sensitive to onboard EMI. Observed failures included: satellite count dropping from 5 to 0 mid-run; position jumps >5 m when RealSense cameras streamed at 30 Hz; and complete fix loss traced to a coiled USB extension cable acting as a resonant loop antenna at the GPS L1 frequency ($\lambda/2 \approx$ 9.5 cm at 1575 MHz). Four targeted mitigations were applied: (1) ferrite cores on all USB cables (common-mode choke, $Z \propto \omega L$); (2) grounded aluminium foil Faraday shield between the GPS module and electronics bay (>20 dB attenuation at 1575 MHz); (3) RealSense frame rate reduced from 30 Hz to 15 Hz (50% reduction in USB bus utilization and radiated power spectral density); (4) USB extension cable uncoiled and routed flat. This hierarchy (Separate, Shield, Filter, Reduce) is summarized in Table 2.

Table 2. GPS EMI Mitigation Summary

Noise source	Coupling mechanism	Mitigation applied
USB 3.0 cameras (high frame rate)	Radiated emission; power $\propto$ data rate	Reduced RealSense frame rate from 30 Hz to 15 Hz
Coiled USB extension cable	Resonant loop antenna at $\lambda/2 \approx 9.5$ cm	Uncoiled; routed straight and cable-tied flat
All USB cables	Common-mode RF current on cable shield	Ferrite core at device end of each cable
Electronics bay below GPS module	Radiated noise coupling through PCB ground plane	Grounded Al foil Faraday shield between bay and GPS module

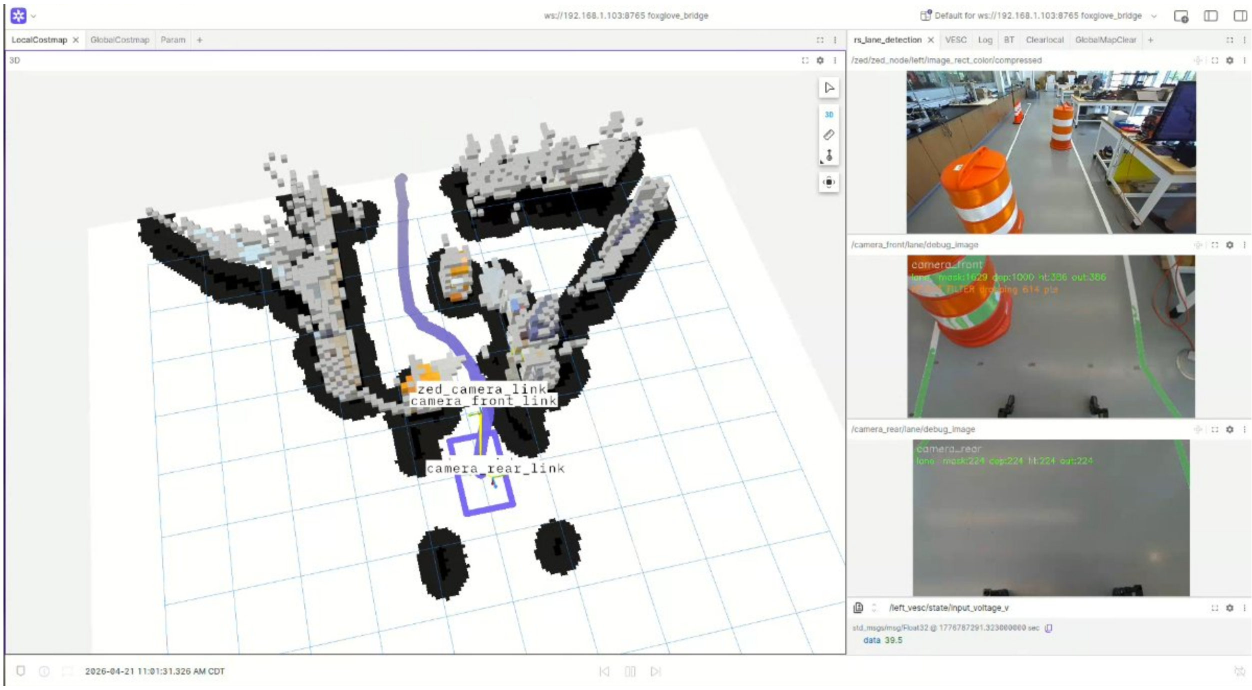


Figure 3. Foxglove Studio during an indoor test session. *Left:* 3D local costmap showing NVBlox voxel occupancy (black/gray blocks) and the MPPI candidate trajectory (purple arc). *Right top:* ZED2 forward-facing color image with a traffic cone. *Right center:* Front RealSense lane detection debug image — green overlay marks detected white lane pixels (386 output points injected into the costmap). *Right bottom:* Rear RealSense lane detection debug image (224 output points).

6. Driving Logic

The driving logic stack translates perception outputs into vehicle motion that satisfies Requirement D-1 (waypoint navigation within lane boundaries) and Requirement D-2 (speed envelope 1–5 mph). The stack runs entirely within the Isaac ROS Docker container on the AGX Orin under ROS 2 Humble. **Requirements addressed:** D-1 (waypoint cross-track error ≤ 0.5 m; zero lane-boundary exits) and D-2 (speed envelope $1 \leq v \leq 5$ mph).

6.1 Localization

Global and local localization is provided by `robot_localization` EKF fusing three sources: (1) wheel odometry from VESC Hall sensors (primary dead-reckoning); (2) ZED2 IMU yaw rate (smooth heading); (3) GPS `NavSatFix` via `navsat_transform_node` (global position). Absolute compass yaw is excluded from EKF fusion because combining it with wheel odometry produced oscillation; yaw-rate-only fusion from the ZED2 gyroscope provides stable heading. The EKF publishes at 30 Hz minimum to satisfy Nav2 TF tolerance. Accurate localization is the foundation of waypoint cross-track error ≤ 0.5 m (Requirement D-1).

6.2 Navigation — Nav2 with MPPI

- **Global Planner (NavFn/Dijkstra):** Plans the macroscopic path from the vehicle's current estimated position to the next GPS waypoint using a rolling global costmap (`rolling_window: true`) that incorporates the NVBlox layer. Rolling window is essential for continuous outdoor navigation without a pre-built map.
- **Local Planner (MPPI):** Samples thousands of candidate trajectory rollouts, evaluates them against the local costmap, and selects the velocity command that minimizes a weighted combination of path alignment, path following, and obstacle avoidance costs. The local costmap is expanded to 8×8 m to ensure rollouts are never clipped by costmap boundaries (an oscillation root cause identified in prior testing).
- **Costmap Layer Stack (local):** `nvblox_layer` → `lane_obstacle_layer` → `nvblox_inflation_layer` (inflation radius 0.82 m, cost scaling factor 0.8). The lane obstacle layer injects the LaserScan outputs described in Section 5.3 as lethal costmap cells, preventing the planner from routing through lane boundaries.

Lane following behavior. Because lane boundaries are represented as lethal costmap cells, the MPPI controller inherently avoids them during trajectory scoring: any rollout that passes through a lane-cell receives a prohibitive obstacle cost and is rejected. No separate lane-centering controller is needed; the same MPPI that avoids barrels also avoids lane markings.

Obstacle avoidance. NVBlox provides the 3D obstacle representation for all non-lane obstacles (barrels, cones, ramp). The ESDF-derived costmap is inflated so that the robot's inscribed radius (0.32 m) is contained within the inflation boundary, guaranteeing collision-free trajectories for the vehicle footprint.

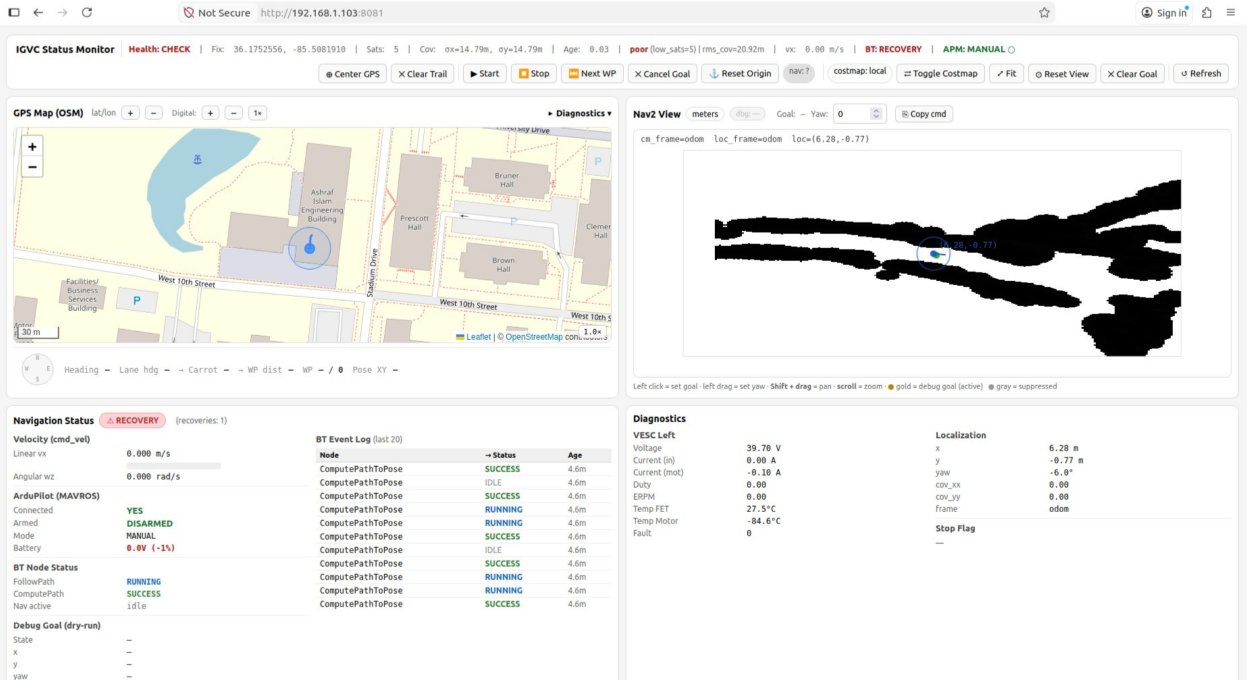
Ramp and chicane navigation. The two GPS waypoint pairs provided before competition guide the vehicle through the No-Man's-Land transitions and ramp entrance. Between waypoints the NavFn planner generates a global path; the MPPI controller executes it locally. Chicane traversal requires tight angular turns ($\leq$ 5 ft radius per rules). Raising `wz_max` from 0.5 to 1.0–1.5 rad/s reduced the effective minimum turning radius and resolved chicane failures observed in early testing. `PathAlignCritic` self-disables in constrained spaces via `max_path_occupancy_ratio: 0.05`, preventing it from forcing the vehicle into walls while trying to align with the global path.

Complex obstacle handling (switchbacks, dead-ends, potholes). The 2026 course includes simulated potholes (2-ft diameter solid white circles) that must be avoided. Because the downward-facing RealSense cameras detect white features indiscriminately, solid white circles will appear as lane obstacles in the local costmap and will be avoided by the same MPPI mechanism used for lane boundaries. Dead-end avoidance relies on the NavFn planner re-routing when the forward path is blocked; the BT `ReactiveFallback` triggers a costmap-clear recovery if the planner cannot find a path within the timeout.

Behavior Tree. The Nav2 Behavior Tree is modified to place `GoalUpdated` only at the top-level `ReactiveFallback`, eliminating the BT recovery cascade that caused ∼300 oscillatory recoveries per run in prior testing (reduced to ∼1 after fix). Recoveries are limited to costmap-clearing only (no spin or back-up); a `RateController` node gates replanning.

Waypoint mission pipeline. The mission sequence is: `mission_waypoint_manager_node` → `carrot_target_node` → `carrot_to_nav2_action_node` → Nav2 `NavigateToPose`. New goals are sent via direct preemption (no explicit cancel) to prevent a BT recovery cascade during goal transitions. The speed ceiling (2.235 m/s, 5 mph) is enforced as a clamp in the differential-drive kinematics node, independent of VESC firmware limits.

Field monitoring. RViz is kept offline during all field operations. System health is monitored through a custom FastAPI + Leaflet.js web dashboard (`status_monitor_gateway`) shown in Figure 4, providing a live GPS map, costmap overlay, BT state, and VESC diagnostics without GPU contention.



http://192.168.1.103:8081

IGVC Status Monitor | Health: CHECK | Fix: 36.1752556, -85.5081910 | Sats: 5 | Cov: σx=14.79m, σy=14.79m | Age: 0.03 | poor (low_sats=5) | rms_cov=20.92m | vx: 0.00 m/s | BT: RECOVERY | APM: MANUAL ○

Heading – Lane hdg – → Carrot – → WP dist – WP – / 0 Pose XY –

Goal: – Yaw: 0

cm_frame=odom loc_frame=odom loc=(6.28,-0.77)

Left click = set goal · left drag = set yaw · **Shift + drag** = pan · **scroll** = zoom · ● gold = debug goal (active) ● gray = suppressed

Navigation Status ⚠ RECOVERY (recoveries: 1)

Velocity (cmd_vel)

Linear vx	0.000 m/s
Angular wz	0.000 rad/s

ArduPilot (MAVROS)

Connected	YES
Armed	DISARMED
Mode	MANUAL
Battery	0.0V (-1%)

BT Node Status

FollowPath	RUNNING
ComputePath	SUCCESS
Nav active	idle

Debug Goal (dry-run)

State	–
x	–
y	–
yaw	–

BT Event Log (last 20)

Node	→ Status	Age
ComputePathToPose	SUCCESS	4.6m
ComputePathToPose	IDLE	4.6m
ComputePathToPose	SUCCESS	4.6m
ComputePathToPose	RUNNING	4.6m
ComputePathToPose	RUNNING	4.6m
ComputePathToPose	SUCCESS	4.6m
ComputePathToPose	IDLE	4.6m
ComputePathToPose	SUCCESS	4.6m
ComputePathToPose	RUNNING	4.6m
ComputePathToPose	RUNNING	4.6m
ComputePathToPose	SUCCESS	4.6m

Diagnostics

VESC Left

Voltage	39.70 V
Current (in)	0.00 A
Current (mot)	-0.10 A
Duty	0.00
ERPM	0.00
Temp FET	27.5°C
Temp Motor	-84.6°C
Fault	0

Localization

x	6.28 m
y	-0.77 m
yaw	-6.0°
cov_xx	0.00
cov_yy	0.00
frame	odom

Stop Flag

—

Figure 4. IGVC Status Monitor web dashboard (`status_monitor_gateway`) during a field session. *Top bar:* GPS fix quality, EKF covariance, BT state, APM mode. *Left:* Leaflet.js GPS map with robot position. *Center:* Nav2 costmap overlay. *Bottom:* VESC diagnostics and EKF localization readout. RViz is kept offline to avoid GPU contention with NVBlox (Requirement P-2).

7. Key Performance Indicators

Two KPIs were selected to directly measure competitive performance on the AutoNav course, corresponding to Requirements K-1 and K-2.

KPI-1 — Course completion time (Requirement K-1). The adjusted completion time (net of penalty deductions per §II.5) is the primary ranking metric. Our target is ≤ 270 s, leaving a 90-s margin within the 6-minute window. This was derived by estimating the ∼500-ft course length at nominal speed (2.235 m/s) and adding 30% margin for obstacle deceleration. Completion time will be recorded from judge score sheets at competition (see Table 1, K-1).

KPI-2 — Traffic-violation-free run rate (Requirement K-2). Each traffic violation deducts 5–10 ft from the adjusted score. Even two sideswipe penalties (−5 ft each) can shift placement by one position. Our target is zero penalty feet in ≥ 1 of 2 attempts per heat. Violation type and count will be logged from judge score sheets at competition (see Table 1, K-2).

7.1 Simulation vs. Field Testing

No simulation environment (Gazebo, Isaac Sim, etc.) was used in this project. The primary reasons are: (1) the Isaac ROS NVBlox stack does not have a maintained Gazebo plugin for the AGX Orin Docker environment, making simulation-to-real transfer unreliable for the specific perception pipeline; (2) the competition course is an outdoor asphalt surface with real lighting variability that is difficult to replicate in simulation, particularly for the HSV-based lane detection.

In place of simulation, the team followed a progressive physical testing protocol with equivalent rigor:

- **Bench unit tests:** Each ROS 2 node was verified in isolation using `ros2 topic hz`, parameter verification scripts, and pre-recorded `ros2 bag` files replayed indoors.
- **Indoor integration tests:** Full stack launch inside a corridor environment for CAN bus verification, EKF convergence, costmap layer ordering, and BT state-machine correctness. At least 5 minutes of continuous autonomous operation required before any outdoor session.
- **Outdoor field tests:** Progressive outdoor runs on TTU's paved outdoor grounds, starting from NVBlox-only navigation, then adding lane obstacles, then increasing speed toward the 1.0 m/s operational target. Each change from a known-working baseline is isolated to a single parameter to enable root-cause identification. All sessions recorded to `ros2 bag` for post-hoc Foxglove analysis.

Key differences between bench/indoor predictions and outdoor actuals: CLAHE was necessary for outdoor shadow performance (not needed indoors); GPS EMI was not observable indoors; MPPI chicane failures only appeared outdoors at 1.0 m/s where the turning radius constraint is binding.

8. Analysis of Complete Vehicle

8.1 Lessons Learned

- Containerized environments (Docker) mask important system state changes. All in-container modifications must be version-controlled; they do not persist across container rebuilds unless explicitly staged.

- RealSense camera parameters cannot be reliably set via the ROS 2 parameter service after launch due to the driver's two-phase initialization. Always verify applied parameters by reading them back and use a verify-and-retry pattern with appropriate delays.
- NVBlox is GPU-memory and compute intensive. Debug visualization, RViz, and high-frequency ESDF updates will cause thermal throttling on the AGX Orin — profile and tune these parameters early.
- Nav2 Behavior Tree design has a significant impact on runtime behavior. Place `GoalUpdated` only at the top-level `RecoveryFallback`; placing it inside inner subtrees causes recovery cascades that manifest as oscillatory stuck behavior.
- Record `ros2 bag` during every meaningful field test. Post-hoc analysis in Foxglove has consistently revealed root causes not apparent in real time.
- GPS reception is acutely sensitive to onboard EMI. Mitigate early: ferrite cores on all USB cables, grounded foil shield under GPS module, minimum camera frame rates, and straight cable routing.

8.2 Software Testing and Version Control

All software is maintained in the `TnTechIGVC2026` git repository (`https://github.com/LCAS-Lab/TnTechIGVC2026`) with submodules pinned to the `igvc` branch. Feature development occurs on named branches; changes are merged to `main` only after passing a checklist that includes: node launch verification inside the Isaac ROS Docker container, `ros2 topic hz` checks for all critical publishers, and a minimum 5-minute field run without navigation errors. Bug tracking is managed through GitHub Issues with labels for severity and subsystem.

8.3 Failures, Effects, and Strategies of Mitigation

Table 3. Documented Failures and Mitigation Strategies

Failure	Effect	Mitigation Strategy
NVBlox SIGSEGV (exit -11)	Node crash, navigation stack offline	Corrected `sensor_frame` values in all costmap observation sources to `base_link`; validated frame graph before launch
RealSense parameter contention	Gain/exposure settings silently ignored; inconsistent lane detection	Implemented verify-and-retry node (`set_camera_params.py`) with staggered timer delays (front: 10 s, rear: 15 s)

Failure	Effect	Mitigation Strategy
Nav2 recovery cycling (∼300/run)	Robot oscillates in place, fails to reach goal	Removed `GoalUpdated` from inner `ReactiveFallback` wrappers; kept only at top-level `RecoveryFallback`
MPPI trajectory rollout clipping	Hesitation and oscillation near obstacles	Expanded local costmap to 8×8 m to provide full rollout headroom
EKF compass yaw oscillation	Robot heading drift, navigation instability	Changed to yaw-rate only from ZED2 IMU; disabled absolute compass yaw fusion
GPU thermal throttling (NVBlox)	Reduced mapping and planning frequency	Disabled debug visualization; reduced ESDF update and TSDF decay rates; disabled color processing
USB disconnect from VESC during early development	Loss of motor commands mid-run	Switched to CAN as sole runtime interface; USB retained for bench configuration only
GPS fix loss / intermittent position jumps	EKF position divergence; waypoint navigation failure	Four-stage EMI mitigation: (1) ferrite cores on all USB cables; (2) grounded Al foil Faraday shield under GPS module; (3) RealSense frame rate reduced 30 Hz → 15 Hz; (4) USB extension cable uncoiled and routed flat

9. Cyber Security Analysis

Three key vulnerabilities were identified. (1) **Physical USB Access** could allow injection of malicious ROS 2 nodes; hardening requires port locking, secure boot, and cryptographic package signing. (2) **DDS Interference** could allow a bad actor to flood the DDS domain and crash the navigation stack; hardening requires SROS2 mutual TLS and frequency-hopping E-stop RF. (3) **SSH Intrusion** could permit arbitrary command execution; hardening requires key-only authentication, host-based IDS, and an isolated VLAN.

Currently implemented controls: USB ports covered when not in use; AGX Orin uses password-authenticated SSH with factory defaults changed; all autonomy software runs in an Isaac ROS Docker container; wireless E-stop uses a coded RF signal; `ROS_DOMAIN_ID` is set to a team-specific value to prevent cross-talk with adjacent robots.