



## Cookeville Crashers - ATNA Tennessee Technological University



**Submitted:** May 15<sup>th</sup>, 2025  
**Team Captain:** Kyle Madigan ([kdmadigan42@tntech.edu](mailto:kdmadigan42@tntech.edu) – 585-643-0209)  
**Faculty Advisors:** Dr. Hyung Yoon ([hyoon@tntech.edu](mailto:hyoon@tntech.edu))  
Dr. Byron Pardue ([bpardue@tntech.edu](mailto:bpardue@tntech.edu))  
**Target Challenge:** AutoNav  
**Statement of Integrity:** The design and engineering of ATNA have been significant and equivalent to what might be awarded credit in a senior design course.

Signature: 

Date: 14 MAY 26

Signature: 

Date: 14 May 2026

<b>Team Roster</b>			
<b>Role</b>	<b>Name</b>	<b>Email</b>	<b>Major</b>
<b>Electrical &amp; Software</b>	Kyle Madigan	Kdmadigan42@tntech.edu	ME
	Rene Rooker	Mrooker42@tntech.edu	ME
<b>Mechanical</b>	Matthew Bickel	Mjbickel42@tntech.edu	ME
	Abdullah Almatrafi	Afalmatra42@tntech.edu	ME

## **Table of Contents**

1. Abstract----- 1

2. System and Subsystem Requirements ----- 1

3. Mechanical Design ----- 2

4. Safety ----- 5

5. Electrical/Electronic Design----- 6

6. Perception ----- 8

7. Driving Logic----- 10

8. Key Performance Indicators----- 12

9. Analysis of Complete Vehicle ----- 13

10. Cyber Security Analysis----- 14

## 1. Abstract

ATNA, which stands for All-Terrain, Nearly Autonomous, is one of Tennessee Tech’s autonomous ground vehicles designed for the Intelligent Ground Vehicle Competition AutoNav challenge. The vehicle was developed by a four-student mechanical engineering team and required the integration of mechanical design, electrical systems, onboard sensing, embedded control, and autonomous navigation software into one working robotic platform. ATNA uses a centered differential-drive layout with two independently driven wheels, front and rear spring-loaded casters, a 36 V battery system, VESC motor controllers, an NVIDIA Jetson AGX Orin, two RPLiDAR S2 sensors, a ZED 2i camera, and a Pixhawk/GPS system. The final platform is approximately 38 in. long by 30 in. wide and was designed to operate within the size, speed, obstacle, safety, and navigation expectations of the AutoNav course.

The design approach focused on building a reliable manually controlled vehicle first, then adding autonomous capability in stages. This allowed the drivetrain, power system, emergency stops, RC control, sensor communication, and motor feedback to be tested before enabling full autonomous operation. During indoor development, SLAM Toolbox, Nav2, and RViz were used to validate LiDAR sensing, mapping, costmap generation, obstacle avoidance, and autonomous motion toward selected goals. For competition use, the autonomy system is being transitioned toward GPS waypoint navigation, LiDAR-based obstacle avoidance, and ZED camera-based lane detection so the vehicle can navigate the outdoor AutoNav course without a prior map.

At the current stage of development, ATNA has demonstrated safe manual control, emergency-stop functionality, LiDAR-based obstacle detection, real-time mapping, autonomous path planning, and autonomous driving in a controlled test environment. The vehicle has successfully navigated between closely spaced obstacles during indoor testing, showing that the drivetrain, perception system, planning stack, and motor control system can operate effectively as a unified autonomy platform. Continued development will focus on outdoor validation, GPS waypoint navigation, camera-based lane detection, and full AutoNav course testing.

## 2. System and Subsystem Requirements

The system and subsystem requirements for ATNA were developed from the IGVC AutoNav rules and the team’s goal of building a safe, testable autonomous ground vehicle. Since AutoNav requires the vehicle to stay within lane boundaries, avoid obstacles, navigate toward GPS waypoints, and operate at controlled speeds, the requirements were divided into perception, driving logic, and key performance indicators. The table below summarizes each requirement, the design driver behind it, and how it will be measured.

Category	Requirement	Rule / Design Driver	Measurement Method
Perception	Detect obstacles	AutoNav obstacle avoidance	Verify barrels/obstacles in RViz costmap
Perception	Detect white lines	AutoNav lane boundaries	Test white tape detection outdoors
Driving Logic	Generate autonomous motion	Navigate lanes, obstacles, and GPS goals	Confirm /cmd_vel output and goal progress
Driving Logic	Control vehicle speed	IGVC 5 mph max speed	Measure speed with odometry/tachometer
KPI	Avoid narrow obstacles	Course gaps/switchbacks	Drive through measured obstacle gaps
KPI	Verify E-stop	IGVC safety requirement	Trigger E-stops; confirm no motor power
KPI	Verify runtime	Complete repeated runs	Timed battery test under load
KPI	Reach GPS waypoints	AutoNav waypoint navigation	Measure stopped distance from GPS goal

The vehicle was designed specifically to fit the AutoNav challenge. Each requirement was chosen because it affects whether ATNA can safely navigate the course, avoid obstacles, remain within the lane boundaries, and move toward the required GPS waypoints. The requirements were also written so that they can be measured during testing instead of being judged only by observation.

For perception, the aim was to give the vehicle enough information to understand the course around it in real time. ATNA uses two RPLiDAR S2 sensors for obstacle detection and local mapping, with one LiDAR mounted at the front of the vehicle and one at the rear. This layout was chosen to improve obstacle awareness around the vehicle and support costmap-based obstacle avoidance. The ZED 2i camera is mounted on the vertical mast for lane detection because the AutoNav course uses white lane markings that must be identified visually. Together, the LiDAR and camera systems allow the vehicle to separate obstacle detection from lane detection instead of relying on a single sensor for all perception tasks.

For driving logic, the requirements focused on turning perception data into safe vehicle motion. The robot uses ROS 2 on the Jetson to process sensor data, generate navigation commands, and send motor commands to the drivetrain. During indoor testing, SLAM Toolbox, Nav2, and RViz were used to validate LiDAR-based mapping, costmap generation, obstacle avoidance, and autonomous motion toward selected goals. For competition, the driving logic is being transitioned toward outdoor GPS waypoint navigation with LiDAR-based obstacle avoidance and camera-based lane detection. This approach allows the vehicle to use GPS for course-level navigation while still reacting locally to obstacles and lane boundaries.

The key performance indicators were selected because they give the team measurable ways to judge whether the vehicle is becoming competition ready. Obstacle avoidance, GPS waypoint accuracy, vehicle speed, and E-stop response are all important because they directly affect AutoNav performance and safety. During indoor testing, ATNA successfully planned around obstacles and drove toward selected goals, including through obstacle gaps of approximately 3 ft. This is narrower than the expected minimum AutoNav obstacle spacing, which shows that the LiDAR, costmap, planning, and drivetrain systems are already working together. Outdoor KPI testing will focus on lane detection, GPS waypoint accuracy, full-course navigation, and validating that the robot maintains safe speed and reliable E-stop behavior in competition-like conditions.

### 3. Mechanical Design

ATNA's mechanical design was built around creating a strong, serviceable, and compact autonomous ground vehicle platform for the IGVC AutoNav course. The vehicle uses a centered differential-drive layout with two independently driven wheels, one spring-loaded caster at the front, and one spring-loaded caster at the rear. This layout was selected because it allows the robot to turn about its own center point, simplifies the drivetrain control model, and reduces the space needed to maneuver through obstacles, switchbacks, and narrow course sections. The final vehicle footprint is approximately **38 in. long by 30 in. wide**, which keeps the platform within the IGVC size limits while still providing enough space for the drivetrain, battery, electronics, sensors, safety equipment, and required payload.

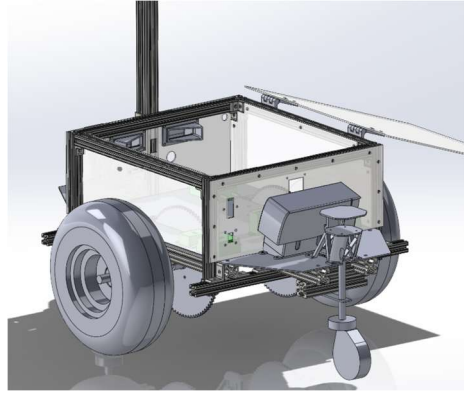


Figure 1 - Full CAD assembly of ATNA showing the overall vehicle layout, sensor mast, drivetrain, electronics cage, and trimmed chassis corners.

The main frame is built from 8020 aluminum extrusion. The lower drivetrain frame uses **1.5 in. × 1.5 in. 8020 rails**, while the electronics cage and upper structure use **1 in. × 1 in. rails** to reduce weight. 8020 was selected instead of welding because it is modular, adjustable, and easier to modify during development. The frame corners were also trimmed to reduce the vehicle's swept path during turns. A **1/8 in. aluminum base plate** was waterjet cut and mounted to the lower frame. This plate supports the electronics cage, drivetrain-related components, battery holder, sensor mounts, and vertical mast. The electronics cage houses the Jetson, relays, DC-DC converters, VESCs, Arduino, beacon control circuit, cooling fan, and wireless E-stop relay.

The drivetrain uses two independent axle sections made from a modified go-kart axle. Each axle section is supported by two pillow block bearings and drives one **14 in. treaded tire**. Power is transferred through a **#35 chain drive**, with a **10-tooth motor sprocket** driving an **80-tooth axle sprocket** for an **8:1 reduction ratio**. This lowers top speed and increases wheel torque, matching the competition need for controlled low-speed navigation.



Figure 2 - ATNA drivetrain showing the independent axle section, chain drive, sprockets, pillow block bearings, and drive wheel.

The motors are mounted using custom PAHT-CF carbon-fiber-reinforced nylon mounts with slotted adjustment holes for chain tensioning and sprocket alignment. 3D-printed sprocket covers were added to reduce debris entering the electronics cage from below and to protect nearby wiring from the moving drivetrain. The vehicle is supported by one **40 lb spring-loaded caster** at the front and one at the rear. These casters stabilize the robot while allowing the centered drive wheels to control motion. The spring-loaded design helps the vehicle maintain ground contact over ramps, uneven pavement, and small terrain changes. The electronics housing is enclosed with laser-cut frosted plexiglass panels mounted to the 8020 frame. Foam rubber gasket material was added around panel and lid contact surfaces to improve water resistance. The hinged lid is held closed with cabinet magnets, allowing the cage to stay protected while remaining accessible for testing and repairs.

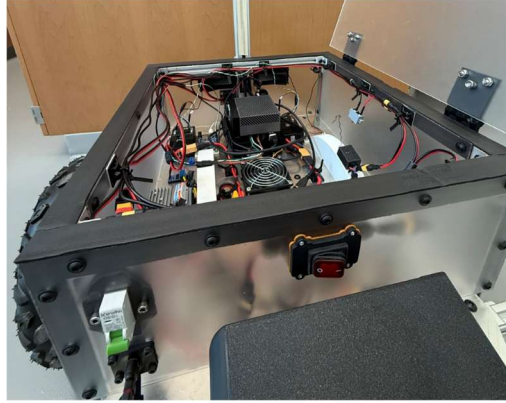


Figure 3 - Electronics cage with frosted plexiglass panels, gasket material, and hinged lid for weather resistance and serviceability.

The sensor and safety hardware are mounted to support both perception and competition operation. The front and rear RPLiDAR sensors are mounted along the vehicle centerline for obstacle coverage, while the ZED 2i camera is mounted about **4 ft above the ground** for lane detection. The Pixhawk enclosure, GPS module, E-stop, and beacon light are mounted near the top of the vehicle for visibility, accessibility, and reduced interference from the drivetrain and electronics.

Several major components, including the Flipsky motors, VESCs, 36 V battery, Jetson AGX Orin, Pixhawk/GPS, LiDAR sensors, and ZED camera, were purchased ready-made and integrated into the custom platform. Their size, mounting needs, cooling requirements, wiring access, and sensor placement strongly influenced the final frame, base plate, electronics cage, and mast design.

### Mechanical Requirement Comparison

Requirement	Target Value	Actual Value	Results
Vehicle size	2–4 ft W, 3–7 ft L, <6 ft H	30 in. W × 38 in. L, <6 ft H	Meets IGVC size limits
Payload support	Carry 20 lb. payload	Designed for 20 lb; drive-tested with 200 lb person	Exceeds payload requirement
Low-speed drivetrain	≤5 mph, smooth low-speed control	8:1 reduction, 420 ERPM minimum, 4.97 mph measured	Meets speed/control target
Tight maneuvering	Near zero-radius turning	Centered differential drive turns about vehicle center	Meets maneuverability target
Electronics protection	Protected, serviceable enclosure	Plexiglass panels, gaskets, hinged lid, protected wiring	Meets weather/serviceability goal
Ramp/terrain handling	Stable ramp transition	Successfully drove over homemade ramp	Meets ramp stability target

Overall, the mechanical design shows that ATNA was built around the AutoNav requirements rather than simply assembled around available parts. The frame provides a strong and adjustable foundation, the differential drivetrain supports tight maneuvering and controlled motion, the housing protects the electrical system from outdoor conditions, and the sensor mast places the perception and safety components where they can operate effectively. The measured vehicle size and current drivetrain testing show that the platform meets the major mechanical requirements needed for continued outdoor validation and competition preparation.

## 4. Safety

Safety was treated as a major design requirement because ATNA is a battery-powered autonomous vehicle with moving drivetrain components, high-current electrical systems, and the ability to operate without direct human control. The safety design uses multiple layers, including a main breaker, main power switch, mechanical E-stop, wireless E-stop, RC motor arming logic, software zero-command behavior, speed limits, drivetrain guards, protected wiring, and a visible beacon light.

When the robot is transported, parked, or charged, the team follows procedures that keep the drivetrain disabled and prevent accidental power-up. During transport, the battery is removed and the chassis is secured with a ratchet strap to prevent wheel movement. When parked or in standby, the physical E-stop is pressed, the main power switch and breaker are turned off, and the RC transmitter is turned off. The robot does not use a separate mechanical parking brake, so parked-state safety is based on removing electrical power and preventing the system from being armed accidentally. During charging, the battery can be charged on or off the vehicle, but the team normally disconnects it and charges it separately. The charger is rated at 42 V and 2 A, and charging from about 30 V to 41.4 V takes roughly 3 hours. The team uses two batteries, allowing one to charge while the other is used for testing.

During course operation, the most important safety features are the mechanical and wireless E-stop systems. Both E-stops interrupt the 12 V relay coil that enables the motor power circuit. When either E-stop is activated, the relay opens and immediately removes power from the VESC motor controllers and drive motors. The system is also designed so that the wireless E-stop must be intentionally re-enabled after the mechanical E-stop has been released, preventing the motors from automatically regaining power if the mechanical E-stop is accidentally bumped, tripped, or reset. The Jetson, sensors, and other logic electronics remain powered, which allows the robot to stop moving while still preserving diagnostic information for troubleshooting.

Software safety provides an additional layer below the hardware E-stop system. The RC transmitter includes a motor arm/disarm channel, and motor commands are not allowed through unless the vehicle is armed. If the vehicle is disarmed, the RC signal becomes stale, or a stop condition is detected, the command path sends zero ERPM commands to both VESC motor controllers. This is not treated as the primary emergency stop, but it helps prevent unintended motion during startup, testing, and mode switching. Software ERPM limits also keep the vehicle within the IGVC maximum speed requirement.

The beacon light provides a visible indication of vehicle state. It turns on during startup and changes behavior based on operating mode, including autonomous mode. This helps team members recognize when the robot is powered, ready, or capable of autonomous motion, especially during outdoor testing.

## Safety Requirement Comparison

Requirement	Target Value	Actual Value	Results
Mechanical E-stop	Immediate motor power cutoff	Cuts 12 V relay coil to VESC/motor circuit	Validated; motor power does not auto-restore without wireless reset
Wireless E-stop	Stop from $\geq 100$ ft	Tested from 200+ ft; cuts same relay coil	Exceeds range target
Prevent unintended motion	Zero motor command unless armed	RC arming logic and stale-signal zero ERPM implemented	Validated during powered testing
Limit speed	$\leq 5$ mph	ERPM limits and speed modes implemented	Supports safe operation
Drivetrain guarding	Guard moving chain/sprockets	3D-printed sprocket covers installed	Reduces contact and wiring risk
Park/transport safety	Disable and secure robot	E-stop, power off, breaker off; battery removed and strapped for transport	Procedure established
Charging safety	Charge without robot activation	Battery usually removed; 42 V, 2 A charger; $\sim 3$ hr charge	Safe charging procedure established
State indication	Visible powered/autonomous status	Beacon controlled by Jetson/Arduino	Improves field awareness

Overall, ATNA's safety system uses both hardware and software layers. The mechanical and wireless E-stops provide the primary safety function by immediately removing motor power through the relay coil circuit. The RC arming switch, stale-signal timeout, zero ERPM command behavior, and speed limits provide software-level protection during normal operation. The main breaker, power switch, battery handling procedure, drivetrain guards, protected wiring, and beacon light further reduce risk during transport, charging, testing, and competition use. Current testing shows that the robot can be powered, driven, stopped, transported, and charged using repeatable safety procedures, while final outdoor testing will continue to validate safe behavior during full autonomous operation.

## 5. Electrical/Electronic Design

ATNA's electrical and electronic system was designed to safely power the drivetrain, onboard computer, sensors, communication hardware, and support electronics needed for autonomous operation. The system uses a **36 V, 14 Ah battery** as the main energy source, with separate voltage branches for the drivetrain, computing hardware, sensors, and beacon light. The major electronic components include the NVIDIA Jetson AGX Orin, two VESC motor controllers, two brushless drive motors, two RPLiDAR S2 sensors, a ZED 2i camera, a Pixhawk 4 with M8N GPS, an RC receiver, an Arduino beacon controller, DC-DC converters, relays, breakers, fuses, and the mechanical and wireless E-stop hardware. These major electrical systems were fully integrated using separate **5 V, 12 V, 24 V, and 36 V** power lines to support the different loads on the robot.

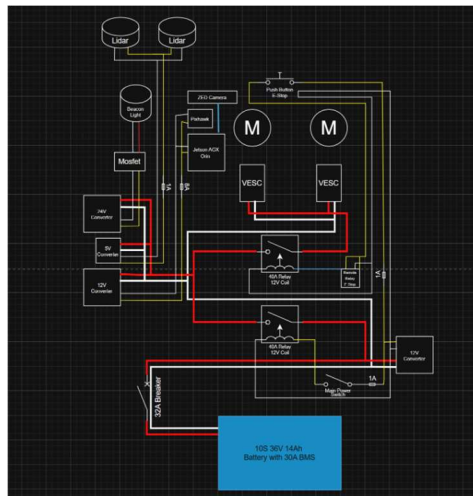


Figure 4 - ATNA power distribution diagram showing the 36 V battery input, main breaker, main switch, relay-controlled motor circuit, and 5 V, 12 V, 24 V, and 36 V branches.

The main power source is an externally mounted **36 V, 14 Ah battery** with a built-in BMS. The battery connects to the robot through a waterproof panel-mounted connector on the side of the electronics cage. From there, power passes through a **32 A main breaker** before reaching the main switch and the rest of the distribution system. The main switch controls power to the full vehicle. When the switch is off, all vehicle systems are off. When the switch is on, the logic electronics can power up, but the motors only receive power if both the mechanical and wireless E-stop systems allow the motor relay to close. This was an important design choice because it allows the robot's computer and sensors to remain powered for startup and debugging while keeping the drivetrain disabled until the vehicle is intentionally enabled.

The power system is divided into four voltage levels. The **36 V branch** powers only the VESC motor controllers and drive motors. The **24 V branch** powers the beacon light. The **12 V branch** powers the main switch circuit, E-stop system, Jetson, and Pixhawk. The **5 V branch** powers the LiDAR sensors. The ZED camera and Arduino are powered through the Jetson rather than through separate voltage branches. This separation made the system easier to protect and troubleshoot because each major group of components could be fused, tested, and isolated more clearly.

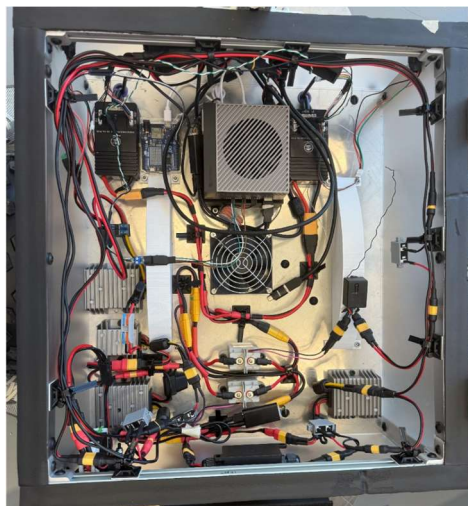


Figure 5 - Interior of ATNA's electronics cage showing the Jetson, VESCs, relays, DC-DC converters, wiring, and protected component layout.

The Jetson AGX Orin is the main onboard computer and runs ROS 2, sensor processing, manual/autonomous control logic, and motor command generation. The Jetson communicates with the VESCs over CAN, which was selected for reliable motor command and feedback communication. The Pixhawk communicates over serial and provides RC input, IMU data, and GPS capability, while the LiDARs, ZED camera, and Arduino beacon controller communicate through USB.

Circuit protection and wire routing were major design considerations. The 36 V motor circuit uses **10 AWG wire** and XT90 connectors, while lower-voltage branches use smaller-gauge wiring and XT60 connectors. The Jetson is protected by an **8 A fuse**, most non-motor branches use **1 A breakers**, the VESC circuit is protected by the main breaker, and the DC-DC converters include internal protection. The VESCs are mounted inside the electronics cage near the drivetrain, but their motor phase wires are routed immediately downward through the aluminum base plate to reduce high-current wiring inside the cage and limit EMI effects on nearby signal wiring.

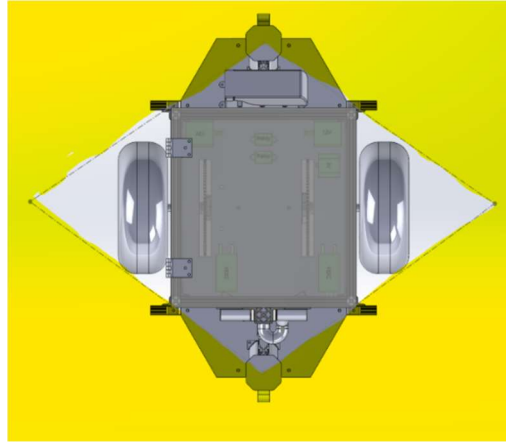
The power system has an estimated capacity of **504 Wh** based on the 36 V, 14 Ah battery. Current calculations estimate about **10 A** draw from the battery at full motor throttle, giving an expected runtime of at least **one hour**, depending on terrain, acceleration, turning, sensor load, and Jetson power draw. The team uses two batteries so one can charge while the other powers the robot. The charger is rated at **42 V and 2 A**, and charging from about **30 V to 41.4 V** takes roughly **3 hours**.

### Electrical/Electronic Requirement Comparison

Requirement	Target Value	Actual Value	Results
Battery capacity	$\geq 1$ hr runtime	36 V, 14 Ah; $\sim 10$ A estimated full-throttle draw	Expected runtime meets target
Voltage rails	36 V, 24 V, 12 V, 5 V	Separate branches for motors, beacon, electronics, and LiDARs	Meets subsystem power needs
Circuit protection	Main + branch protection	32 A main breaker, 8 A Jetson fuse, 1 A branch breakers	Protects major loads
Motor power isolation	Stop motors, keep logic powered	E-stops cut VESC/motor relay only	Preserves diagnostics during stop
Motor communication	Reliable Jetson-to-VESC control	CAN bus communication implemented	Supports commands and feedback
Sensor integration	LiDAR, camera, GPS/IMU available	RPLiDARs, ZED 2i, Pixhawk 4, M8N GPS installed	Supports AutoNav sensing
Beacon status	Visible operating-state indication	24 V beacon controlled by Arduino/MOSFET	Improves field awareness
Battery recharge/swap	Removable battery, practical recharge	Two batteries; $\sim 3$ hr charge from 30 V to 41.4 V	Reduces testing downtime

## 6. Perception

ATNA's perception system is designed for the IGVC AutoNav course, where the vehicle must detect lane boundaries, avoid obstacles, and navigate through the course using onboard sensors. The system uses two RPLiDAR S2 sensors for obstacle detection and a ZED 2i camera for lane detection. The LiDAR sensors were prioritized first because they provide reliable obstacle data for mapping, costmaps, and local avoidance. The ZED camera is mounted on the sensor mast for machine-vision-based detection of the white course lines. Together, these sensors allow ATNA to treat obstacle detection and lane detection as separate perception tasks instead of relying on one sensor for the entire environment.



*Figure 6 - Effective LiDAR coverage from the front and rear RPLiDAR sensors used for obstacle detection around the vehicle.*

Obstacle detection is handled using the front and rear RPLiDAR sensors. The front LiDAR identifies obstacles in the robot's forward path, while the rear LiDAR improves awareness around the full vehicle footprint. Each LiDAR publishes scan data into ROS 2, where the scans are transformed into the robot's coordinate frame and used by the navigation stack to create obstacle costmaps. These costmaps mark occupied areas around the vehicle and allow the driving logic to plan paths around barrels, walls, and other objects. During indoor testing, the LiDAR system was able to detect obstacles, support real-time mapping, and allow the robot to plan through narrow obstacle gaps without contact.

Lane detection is handled using the ZED 2i camera mounted approximately 4 ft above the ground on the sensor mast. The camera is intended to identify the white lane markings used in the AutoNav course and provide boundary information to the driving logic. The planned processing approach is to use machine vision to isolate the white line markings from the camera image, filter the result based on expected lane color and geometry, and estimate the lane position relative to the robot. This lane information will be used to keep the vehicle inside the course boundaries while GPS and LiDAR guide the robot toward waypoints and around obstacles. The camera hardware is installed, but final outdoor lane detection validation is still part of the remaining competition preparation.

The internal representation used by ATNA's driving logic is a local map or costmap built from the vehicle's sensor data. During indoor testing, SLAM Toolbox and Nav2 were used to build and update a map in real time from LiDAR and odometry data. This helped the team validate sensor transforms, obstacle detection, costmap generation, and autonomous motion in a controlled environment. For outdoor AutoNav operation, the perception system is being transitioned toward a competition-focused representation using GPS waypoints for course-level navigation, LiDAR costmaps for local obstacle avoidance, and camera-detected lane boundaries to constrain the vehicle inside the course. This representation is updated continuously as the robot moves, so new obstacles and lane information can affect the path being generated by the driving logic.

### Perception Requirement Comparison

Perception Requirement	Target Value	Most Recent Actual Value	Results
Obstacle detection	≥8 ft detection range	Front/rear RPLiDARs tested in RViz/costmaps	Validated indoors
Lane detection	3 in. white lines	ZED 2i installed; outdoor validation pending	Still in progress
Local representation	Live costmap for driving logic	SLAM/Nav2 costmaps tested indoors	Supports obstacle-aware planning
Perception-to-motion validation	Avoid obstacles without contact	Navigated ~3 ft obstacle gaps indoors	Validated core perception pipeline

Overall, ATNA’s perception system was designed around the main needs of AutoNav: identifying obstacles, detecting lane boundaries, and providing usable information to the driving logic. The LiDAR-based obstacle detection system has already been tested successfully with ROS 2, Nav2, and indoor autonomous navigation. The next major perception task is completing outdoor validation of the ZED camera lane-detection system and combining it with GPS waypoint navigation and LiDAR-based obstacle avoidance for full competition operation.

### 7. Driving Logic

ATNA is targeting the **AutoNav** challenge, so the driving logic is designed around lane following, obstacle avoidance, GPS waypoint navigation, and controlled low-speed motion. The vehicle is not targeting the SelfDrive challenge, so pedestrians, stop signs, and road-rule behavior are not part of the current driving logic. ATNA’s driving system uses the perception output from the LiDAR sensors, ZED camera, GPS/IMU system, and wheel odometry to decide where the robot should move next. The Jetson AGX Orin runs the ROS 2 software stack and converts those decisions into motor commands for the differential-drive drivetrain.

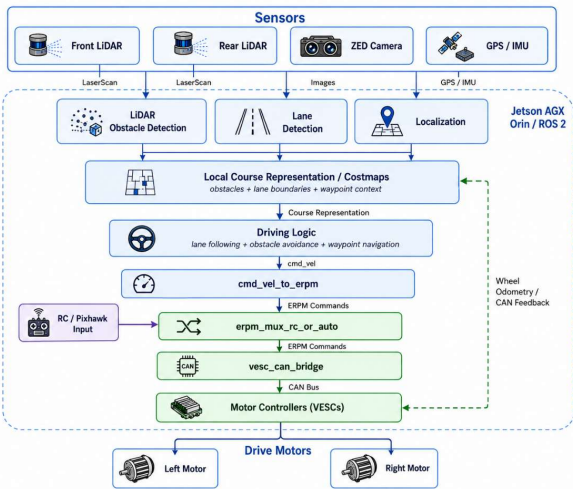


Figure 7 - Perception and driving-logic flow for ATNA, showing LiDAR obstacle data, ZED lane detection, GPS/IMU localization, local costmap creation, waypoint navigation, and drivetrain command output.

The main driving logic is organized around a local course representation. LiDAR data is used to mark obstacles in the local costmap, camera data is used to identify the white lane boundaries, and GPS/IMU data provides course-level position and heading information. The robot then uses this information to choose motion commands

that keep the vehicle moving toward the next waypoint while avoiding obstacles and staying inside the lane boundaries. During indoor development, RViz, Nav2, and SLAM Toolbox were used to test this same general flow by selecting goal poses and confirming that the robot could build a map, avoid obstacles, and drive toward a goal. For competition, the goal source is being transitioned from RViz-selected goals to GPS waypoints, while LiDAR obstacle avoidance and camera lane detection provide the local constraints needed for AutoNav operation.

Lane following is handled by treating the detected white course lines as boundaries that the robot should not cross. The ZED camera is mounted high on the mast so it can see the course markings ahead of the vehicle. The lane detection system will estimate the position of the left and right lane markings relative to the robot and provide a centerline or boundary constraint to the driving logic. When both lane boundaries are visible, the vehicle can bias its desired path toward the center of the lane. When one boundary is partially missing or blocked, the system can continue using the visible boundary, recent path history, GPS waypoint direction, and obstacle information to maintain a safe course direction.

Obstacle avoidance is handled primarily through the front and rear RPLiDAR sensors. The LiDAR scans are transformed into the robot frame and used to update the local costmap. Obstacles such as barrels, walls, or course objects are treated as occupied regions that the planner should avoid. If a direct path to the waypoint is blocked, the driving logic selects a path around the obstacle while staying within the lane constraints. This same approach is used for more complex course features such as switchbacks, center islands, dead ends, traps, and potholes. In those situations, the robot does not need to identify every feature by name; instead, it treats them as combinations of lane boundaries, blocked regions, and open drivable space. The planner chooses the safest available path through the local costmap while continuing to make progress toward the next GPS waypoint.

GPS waypoint navigation provides the high-level direction for the course. The competition system is intended to load the required GPS waypoints and drive toward them in sequence. The Pixhawk/GPS system provides position and IMU information to the Jetson, while wheel odometry from the VESCs provides local motion feedback. As each waypoint is reached within the target tolerance, the driving logic advances to the next waypoint. This lets the robot follow the overall course route without needing a prior map of the competition field. This is the intended competition operation: the robot should follow a list of four GPS waypoints and advance from one waypoint to the next as it moves through the course.

Vehicle motion is generated through the same differential-drive command path in both manual and autonomous operation. The driving logic outputs a desired forward velocity and turn rate as `/cmd_vel`. The `cmd_vel_to_erpm` node converts that desired motion into left and right ERPM commands using the vehicle's wheel radius, track width, gear ratio, pole-pair count, and software speed limits. The `erpm_mux_rc_or_auto` node selects between RC commands and autonomous commands based on the current operating mode. The selected commands are sent to the `vesc_can_bridge`, which communicates with the VESC motor controllers over CAN and publishes wheel feedback for odometry. This structure allows the robot to use the same drivetrain model for remote control and autonomous driving.

Ramp and rough-surface behavior are handled through a combination of perception, mechanical design, and conservative driving logic. The LiDAR sensors are mounted approximately 14 in. above the ground, while the AutoNav ramp is expected to be approximately 12 in. high, so the ramp should fall below the main LiDAR scan plane rather than appearing as a full-height impassable obstacle. This helps prevent the local costmap from incorrectly treating the ramp as a barrier. The ZED camera provides the better sensing path for identifying the ramp visually, while the drivetrain and mechanical layout allow the vehicle to physically climb it. Mechanically, the 14 in. treaded drive wheels and front/rear spring-loaded casters help the vehicle remain stable over ramp transitions. From the driving-logic side, ATNA is commanded at controlled low speeds so it approaches ramps,

potholes, and uneven terrain safely. If an object or terrain feature is detected as impassable, the local planner can route around it; if it is identified as passable, the drivetrain has enough torque and ground contact to continue through it at a controlled speed.

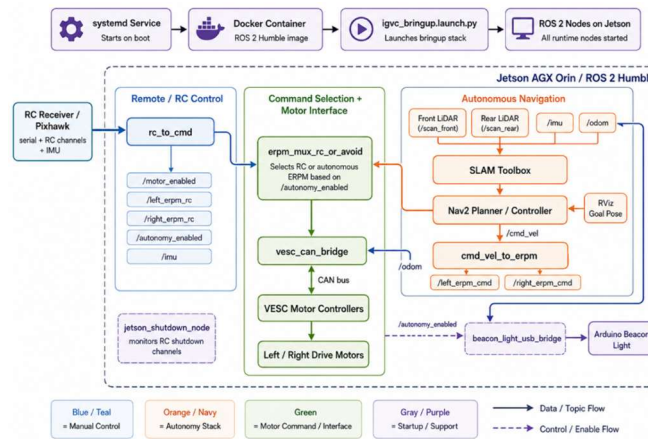


Figure 8 - Autonomous command flow from perception and planning to /cmd\_vel, ERPM conversion, command selection, CAN communication, and VESC motor control.

### Driving Logic Requirement Comparison

Requirement	Target Value	Actual Value	Results
Autonomous command generation	Convert autonomy output to drivetrain commands	Nav2 /cmd_vel converted to ERPM and sent to VESCs	Command path validated
Obstacle avoidance	Avoid obstacles without contact	Planned through ~3 ft indoor obstacle gaps	Validated indoors
Lane following	Follow 3 in. white lane markings	ZED installed; outdoor lane-control validation pending	Detected, but still in progress
GPS waypoint navigation	Reach waypoint within 1.5 ft	Pixhawk/GPS installed; outdoor testing pending	In progress
Controlled motion	≤5 mph max with odometry feedback	ERPM limits and VESC odometry implemented	Supports safe AutoNav motion

Overall, ATNA’s driving logic was built to turn perception data into safe, low-speed motion commands. The indoor autonomy testing proved that the robot can use LiDAR, odometry, costmaps, Nav2 planning, and VESC motor control to drive toward goals while avoiding obstacles. The competition version uses the same core command pipeline, but adds GPS waypoint navigation and camera-based lane boundaries so the robot can operate on the outdoor AutoNav course. The remaining work is focused on outdoor integration and validation rather than changing the basic driving architecture.

### 8. Key Performance Indicators

ATNA’s key performance indicators were selected to measure whether the robot is becoming competitive for the IGVC AutoNav course. The selected KPIs focus on the vehicle’s ability to move through the course safely, detect and avoid obstacles, follow the lane boundaries, reach GPS waypoints, maintain the required speed range, and operate long enough for repeated course attempts. These indicators were chosen because they directly connect to the AutoNav scoring goals and to the major technical risks of the vehicle: perception reliability, drivetrain control,

navigation accuracy, safety response, and battery endurance. The IGVC-based design targets include staying within white lane boundaries, avoiding obstacles, navigating GPS waypoints, maintaining an average speed of at least 1 mph, and staying below the 5 mph maximum speed limit.

### KPI Requirement Comparison

KPI	Measurement Method	Target	Actual Value	Results
Obstacle avoidance	Drive through measured obstacle gaps	Navigate $\geq 5$ ft gaps	Navigated $\sim 3$ ft indoor gap	Exceeds indoor target
GPS waypoint accuracy	Measure stopped distance from waypoint	Stop within 1.5 ft	GPS installed; outdoor validation pending	In progress
Lane detection	Test white tape/paint outdoors	Detect 4–6 in. white lines	ZED installed; validation pending	Detected; following in progress
Speed control	Measure with odometry/tachometer	1–5 mph	4.97 mph max measured in RC mode	Meets max speed target
E-stop response	Trigger mechanical/wireless E-stops	Immediate motor power cut	VESC/motor relay power cut validated	Meets safety target
Runtime	Timed battery test under load	$\geq 1$ hr	$\sim 10$ A full-throttle estimate from 36 V, 14 Ah battery	Expected to meet target
Ramp capability	Drive over test ramp	Stable climb/transition	Homemade ramp test passed	Meets ramp target

The current KPI results show that ATNA has a strong base platform and a working indoor autonomy stack. The most validated results so far are manual control, E-stop behavior, LiDAR-based obstacle avoidance, autonomous path planning, and ramp testing. Indoor testing showed that the robot could generate a map, plan around obstacles, and drive toward selected goals, including through obstacle gaps of approximately 3 ft. The remaining competition-critical KPIs are outdoor GPS waypoint accuracy, ZED camera lane detection, full course speed validation, and full battery runtime testing under representative outdoor conditions. These remaining tests will determine how ready the robot is for the final AutoNav course, but the completed results show that the drivetrain, safety system, perception stack, and motor-control pipeline are already functioning together as a complete autonomous vehicle platform.

## 9. Analysis of Complete Vehicle

ATNA was built and tested through a staged integration process. The team learned that the most difficult part of the project was not any single subsystem, but making the mechanical, electrical, and software systems work together reliably. Because autonomy could not be tested until the frame, drivetrain, power system, E-stops, motor controllers, sensors, and Jetson were all functioning together, the team tested each subsystem as it was added. Power distribution, E-stop behavior, RC control, motor response, LiDAR sensing, odometry, and autonomous navigation were all checked individually before being used as part of the full vehicle.

Several lessons were learned during construction. The 8020 frame was useful because it was modular and adjustable, but tool access and blocked extrusion channels became harder to manage once more components were installed. Some brackets had to be trimmed or adjusted during assembly, and chain alignment, wire routing, waterproofing, and sensor mounting all required small changes after the initial design. These changes slowed construction, but they improved the final vehicle by making it more serviceable and easier to troubleshoot. The

drivetrain also improved through testing. Early design work considered 3D-printed gears, but the team switched to metal sprockets and #35 chain for better reliability. The PAHT-CF motor mounts included slotted holes, which allowed the motors to be adjusted for chain tensioning and sprocket alignment. This helped prevent repeat chain-alignment issues. The drivetrain was also tested on a homemade ramp, where the 14 in. treaded wheels and spring-loaded casters kept the vehicle stable over the transition.

The most significant electrical/control issue was inconsistent feedback from one VESC motor controller. The controller could still drive the motor, but wheel-speed feedback did not always update continuously, which caused odometry drift. The team responded by checking CAN communication, tuning VESC feedback settings, adding software feedback timeouts, and updating odometry parameters using measured wheel radius and track width. Low-speed motor behavior also required tuning. Commands below about 420 ERPM caused motor jitter, while higher minimum commands made slow driving less useful. The final tuning improved low-speed control and made the robot more predictable for AutoNav testing.

Software testing was handled in layers using ROS 2 tools, RViz, terminal logs, and GitHub version control. Individual nodes were tested first by checking topic inputs and outputs, including RC channel data, motor arming, ERPM commands, VESC feedback, LiDAR scans, and odometry. Code changes were made incrementally, tested one at a time, and preserved through GitHub. The ROS 2 workspace runs in Docker on the Jetson, and the main launch process is started through systemd, which reduced startup issues caused by duplicate nodes or missed manual steps.

Simulation-based testing was limited but useful. Before the drivetrain was complete, MATLAB and Arduino-based simulations were used to test differential-drive logic, joystick mapping, speed scaling, deadband behavior, and command smoothness. Full Gazebo simulation was not used because the team prioritized real hardware integration. RViz became the main validation tool for checking LiDAR transforms, costmaps, odometry, and planned paths. Physical testing showed that real vehicle behavior required more tuning than the calculations predicted. Wheel radius, track width, ERPM limits, motor response, and feedback quality all affected the final drivetrain behavior. Sensor alignment also mattered more than expected because small LiDAR transform errors caused obstacles to appear in the wrong location. Once these parameters were tuned, ATNA successfully mapped indoor environments, planned around obstacles, and drove toward selected goals, including through obstacle gaps of approximately 3 ft. Overall, the completed testing showed that ATNA's mechanical platform, electrical system, safety system, and ROS 2 control stack are functioning together, with the main remaining work being outdoor GPS waypoint navigation, ZED lane detection, speed testing, runtime testing, and full AutoNav validation.

## 10. Cyber Security Analysis

Although ATNA is a student competition vehicle, it uses many of the same technologies that create cyber security risks in modern robotic and autonomous vehicles. The robot includes wireless control systems, remote access to a Linux-based onboard computer, third-party software libraries, and multiple internal communication protocols. These systems are appropriate for a prototype, but if the vehicle were developed beyond competition use, the cyber security design would need to be hardened before wider deployment. The three main vulnerabilities identified for ATNA are wireless/remote access, unsecured internal communication, and software supply-chain or update risk.

### **10.1 Wireless and Remote Access Vulnerability**

ATNA uses wireless systems for RC control, wireless E-stop operation, and remote development access to the Jetson. These systems are necessary for safe testing and debugging, but they also create possible entry points for unauthorized access, signal interference, or unexpected commands. The biggest concern is not only someone accessing the Jetson, but also interference with a wireless link that could affect robot behavior during operation.

Before wider deployment, the vehicle should use stronger wireless authentication, encrypted remote connections, disabled default access settings, and key-based login for remote access. Development tools and remote services should not be left open during normal operation. The RC and wireless E-stop systems should also be evaluated for interference resistance and fail-safe behavior so that communication loss always causes the vehicle to enter a safe stopped condition.

### **10.2 Unsecured Internal Communication**

The robot relies on several internal communication paths, including CAN communication between the Jetson and VESC motor controllers, serial communication with the Pixhawk, and USB connections to sensors and support controllers. These links work well for a student prototype, but they do not provide strong authentication or encryption. If an attacker gained physical or network access to the system, they could potentially inject false motor commands, sensor data, or mode-selection signals.

To harden this issue, a production version should authenticate critical messages and separate safety-critical communication from general development traffic. Motor commands, arming commands, and mode-selection signals should be checked against safety rules before being accepted. The system should also monitor for unusual command values, unexpected message rates, and communication loss. A dedicated safety controller could also be added so that drivetrain power cannot depend only on software running on the main computer.

### **10.3 Software Supply Chain and Update Risk**

ATNA's software stack depends on ROS 2, Linux, Docker, Python libraries, sensor drivers, VESC tools, and other third-party software. This is practical for development, but it creates a supply-chain risk because vulnerabilities can come from outdated packages, unverified dependencies, or untested updates. A software update or dependency change could also break vehicle behavior if it is deployed without testing.

Before wider deployment, the software should use controlled releases instead of active development branches. Docker images should be versioned, scanned for vulnerabilities, and built only from trusted sources. Jetson security updates should be applied on a planned schedule, but tested before being used on the vehicle. Code changes should go through review, version control, and vehicle-level testing before deployment. Unused services and packages should also be removed to reduce the attack surface.