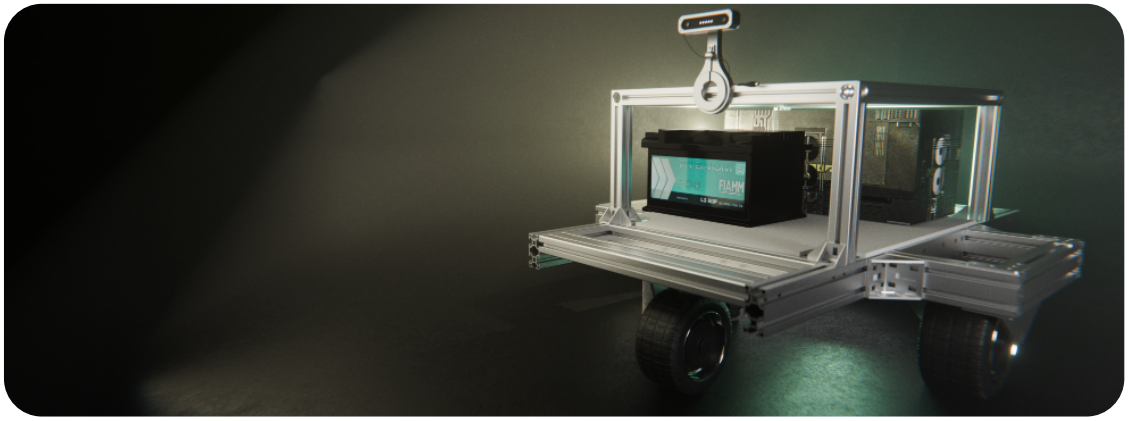
33rd International Ground Vehicle Competition



The Queen Bee

Team Members

Nitin Chandrasekhar	nchandr18@charlotte.edu	**Team Lead**
Sylvester Pudelko	spudelko@charlotte.edu	Electrical Sub-Team
Ryan Guthrey	rguthrey@uncc.edu	Electrical Sub-Team
Govind Menon	pmenon4@charlotte.edu	Software Sub-Team
Samuel Woodring	swoodri2@charlotte.edu	Electrical Sub-Team
Rushil Dasari	rdasari2@charlotte.edu	Mechanical Sub-Team

Faculty Advisors

Amirhossein Ghasemi	ah.ghasemi@charlotte.edu
Sam Shue	slshue@charlotte.edu

I certify that the design and engineering on The Queen Bee by the individuals on this team is significant and equivalent to what might be awarded credit in a senior design course.

Amirhossein Ghasemi

Date Submitted: May 15, 2025

Contents

1 Project Overview

1.1 Introduction

49er Robotics is a student-led organisation from the University of North Carolina, Charlotte (UNCC) that provides a platform for students within the university to discover, explore, and develop their passion for robotics by participating in multiple competitions year-round. The team participating in the International Ground Vehicle Competition (IGVC) in both the AutoNav (*AutoNav*) and *Self-Drive* challenges consists of senior members of the organisation. The team meets every Tuesday and Thursday from 6:00 to 8:00 pm.

1.2 Design Process

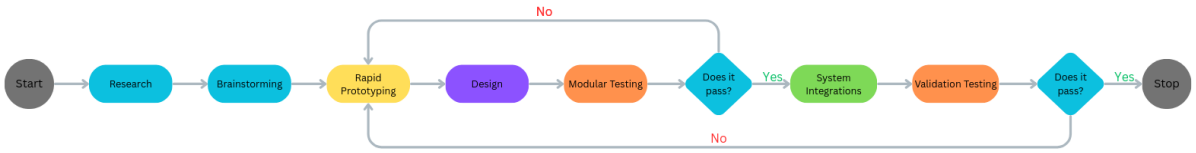


Figure 1: Design Cycle

As shown in the infographic above, the team followed a closed-loop flow that allows for modular understanding of each subsystem in the robot. This engineering process was beneficial in helping the team debug any issues that were encountered on the robot in a effective and decisive manner. The team met two days a week to work on the robot while also discussing future steps. The workflow helped to streamline the meetings by focusing the attention of the group on problems that needed immediate debugging to ensure no problem goes unaddressed. The next few sections will explore the requirements and design assumptions set forth to define what a functional and competition-worthy robot looks like. As shown in the infographic above, the team followed a closed-loop flow that allows for modular understanding of each subsystem in the robot. This engineering process was beneficial in helping the team debug any issues that were encountered on the robot in an effective and decisive manner. The team met two days a week to work on the robot while also discussing future steps. The workflow helped streamline the meetings by focusing the attention of the group on problems that needed immediate debugging to ensure no problem went unaddressed. The next few sections will explore the requirements and design assumptions set forth to define what a functional and competition-worthy robot looks like.

1.2.1 Design Assumptions

The design process was initiated with a review of prior IGVC competition vehicles and a thorough analysis of the 2026 competition rules to identify key constraints and performance targets.

The primary assumption driving early design decisions was software development taking the longest time. Keeping in line with this decision, the team listed some requirements to follow when designing and building the robot.

- **Simple Kinematic Model**: The robot shall have a simple kinematic model designed to drive it. It was agreed that a simple and fast-to-deploy model would best align with the team's goals for this season, and so it was decided to use a three-wheel differential drive with two, independently-mounted, rear driven wheels and one front castor wheel. This will be measured by controlling robot via twist commands and performing driving maneuvers over the course of

ten minutes to confirm uninterrupted and steady movements, with a target of zero interruptions or inconsistencies in vehicle motion throughout the test.

- **Dimensional and Payload Compliance:** The robot shall conform to the size constraints set forth in the IGVC rules, measuring between three to seven feet in length, two to four feet in width, and stay below a maximum height of six feet at all times. In addition, the mechanical structure of the robot shall support a securely mounted payload of twenty pounds with no structural deformation or payload displacement during operation. This requirement will be verified by placing an equally weighted object on the robot and driving around for a minimum of fifteen minutes, with intermittent inspections to confirm no structural deformations or payload displacement.

- **Motor Control Interfacing:** Each rear driven wheel shall be independently controlled via a dedicated motor controller. Keeping with the primary assumption, this approach foregoes the need for custom motor hardware development while still offering steering and navigation control. This will be measured by controlling the wheels via motor controllers at different speeds. Also, the robot must be able to vary its speed from one to five miles per hour.

- **Onboard Power Distribution**: All electrical power shall be distributed onboard the vehicle from a single centralized power system, with no external power sources or base stations permitted during operation. The system shall provide stable, regulated voltage to all computing, sensing, and actuation components simultaneously without brownouts or resets under full autonomous load. This shall be verified by monitoring system voltage across all rails during a minimum thirty-minute full autonomous load test, with a target of zero brownouts, resets, or voltage deviations exceeding five percent of the nominal supply voltage.

- **Mechanical Emergency Stop**: In compliance with the rules, the robot shall have a hardware-based emergency stop (e-stop) that immediately halts the robot regardless of its software state. This requirement will be tested with random activations of the mechanical e-stop during the robot's testing period and verifying a full vehicle stop has occurred with no consequent damage to any hardware on the robot.

- **Wireless Emergency Stop**: The robot shall, also in compliance with the rules, have an additional hardware-based e-stop that brings the robot to an immediate stop at a minimum range of one hundred feet. This requirement will be tested with another series of random activations of the wireless e-stop at varying distances from the robot within the aforementioned range constraint and verifying a full vehicle stop has occurred with no consequent damage to any hardware on the robot.

1.2.2 AutoNav Requirements

The AutoNav challenge requires the competing robot to autonomously navigate around an outdoor obstacle course with a given time constraint while maintaining its speed within a range of one to five miles per hour, remaining within the lanes, and avoiding obstacles. To tackle this challenge, the robot requires some baseline perception and maneuverability capabilities that have been agreed upon by the team and are listed below.

- **Lane Detection**: The robot shall detect and identify the track's boundary at all times. This requirement needs to be met infallibly at every testing point since failure would result in disqualification. The testing will consist of constant detection of white lines under different lighting conditions with success being defined as successful lane detection for more than eighty percent of the robot's run time.

- **Obstacle Detection**: The robot shall detect static obstacles such as construction barrels and simulated potholes. Like lane detection, successful object detection throughout all test and competition runs is critical. Verification will include placing competition-equivalent objects around the robot in varying lighting conditions and angles, with a target of ninety percent detection across all trials.

- **Lane Following**: The robot shall drive within the track lanes at all times while maintaining a legal minimum average speed of one mile per hour. Testing for this requirement will include setting up a test track and performing trial runs with the goal of zero boundary violations and average speed between one and five miles per hour.

- **Waypoint Navigation**: The robot shall advance through the track with the help of preset GPS waypoints. This requirement will be verified by performing navigation trials where the robot must reach the assigned waypoints while avoiding obstacles and staying within lane boundaries. An eighty percent success rate will be considered satisfactory for meeting this requirement.

In addition to the requirements listed above, the team established two key performance indicators to evaluate the robot's overall performance on the AutoNav course: course completion time and speed compliance. Course completion time will be measured using the adjusted time scoring criteria outlined in the rules, whereby the team will establish a target adjusted completion time based on trial runs before the competition, with success qualified by consistently achieving or improving upon the target across successive trials. Speed compliance will be measured by verifying that the robot maintains a legal average speed between one and five miles per hour throughout each run, with a target of one hundred percent compliance across all runs.

1.2.3 Self-Drive Requirements

The Self Drive challenge demands a higher level of navigational complexity compared to AutoNav, requiring the robot to detect and respond to dynamic objects such as pedestrians, perform lane changes, classify road signs, and execute parking maneuvers. Given the expanded scope of this challenge, the team revisited and updated its requirements to reflect the additional capabilities the robot must satisfy, as listed below.

- **Lane and Line Detection**: The robot shall detect and distinguish both white and yellow lane markings with the results displayed on an onboard GUI as required by the rules. This requirement shall be considered satisfied during trial runs when the robot can identify and separate white and yellow track lines, with the same displayed on the GUI, with a success rate over ninety percent.

- **Sign and Dynamic Object Detection**: The robot shall detect stop signs of varying sizes and configurations along with dynamic and static objects such as pedestrians and potholes. This shall be verified through a testing procedure equivalent to that used for the AutoNav obstacle detection requirement, extended to include moving objects and sign classification across a minimum of ten trials per object type.

- **Rules of the Road Compliance**: This robot shall obey rules of the road to keep in line with the purpose of this challenge. This will be verified using the scoring criteria provided in the rules for this challenge aiming for a pre-determined target score that needs to be consistently reached or improved upon.

- **Lane Changing and Parking**: This robot shall safely perform lane changing and parking maneuvers while staying within the boundaries of the course to prevent disqualification of the robot. This requirement will be satisfied with independent testing of the lane changing and

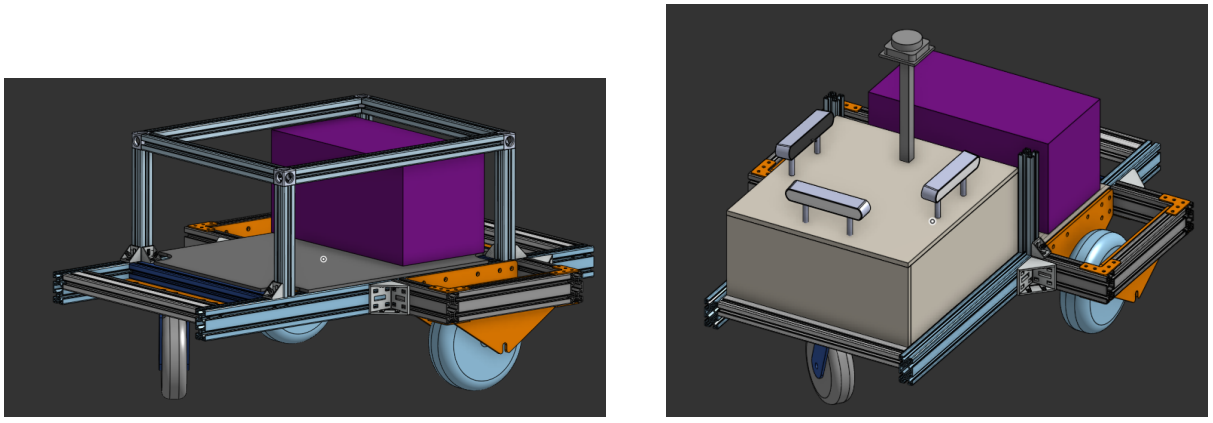
dynamic pedestrian driving logic with a target of zero lane-crossing violations across all test runs.

For the key performance indicators, the team has decided on: function test and full course completion scores. The former shall test how consistently the robot is able to complete all the expected self-drive functions autonomously during test runs. The latter, full course completion score, shall be scored based on how many functions the robot can successfully complete in one full autonomous run and aiming for a target of fifteen out of the required twenty-one functions.

2 Mechanical Design

2.1 Mechanical Overview

This section will deal with investigating the mechanical makeup of the Queen Bee. The mechanical subteam's design decisions were driven by the requirements mentioned in Section 1.2.1 that the team had decided upon. Therefore, the final robot shall have a sturdy chassis that can support the payload, be within the size constraints set forth in the rules, and have a three-wheel differential-drive system. To assist in the initial design, the team made use of the online computer-aided design (CAD) software system Onshape. This software allowed the subteam to engage in rapid prototyping of the several mechanical subsystems that are a part of the vehicle. This effect also influenced part manufacturing, as many specific parts could be designed and 3D-printed immediately for testing. The Queen Bee measures three feet in length, two-and-a-half feet in width, and nearly two feet in height. Attached below are the initial and final CAD designs of the robot.



(a) Preliminary CAD

(b) Final CAD

Figure 2: CAD Iterations of The Queen Bee

2.2 Significant Mechanical Components

2.2.1 Drivetrain

The drivetrain is made entirely of aluminum extrusion that was cut in-house to fit the team's needs. The reason this material was chosen was due to the amount of raw stock 49er's Robotics had available, and it is able to withstand the weight of the payload. The chassis is sized to be as slim as possible to accommodate the rear wheels, which require extensions on either side. To connect pieces of extrusion, M8-sized holes were drilled and tapped on the sides of the extrusions and screwed

together. To attach the castor wheel onto this frame, a piece of carbon fiber was first connected to the frame, as shown in the figure below.

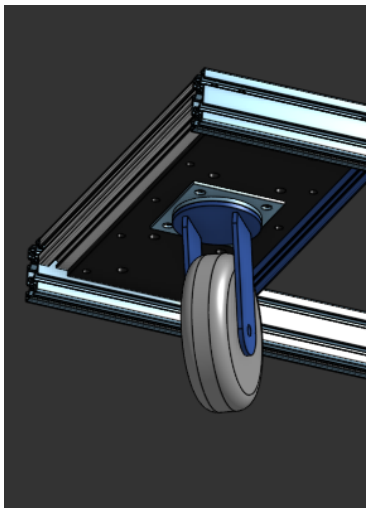


Figure 3: Castor Wheel Attached to Frame Via Carbon Fiber

For the rear wheel assembly, the frame was extended width-wise and two aluminum plates, cut in-house, were attached to the frame, as shown in Figure 2b. These aluminum plates had two slots on their bottom and allowed the scooter wheels to slide into place and then be fastened with M10 nuts on both sides of the wheel. The extended pieces of aluminum were not attached together like the main frame to allow for the wheels to be moved across the chassis during testing. Instead, they were attached using right-angle aluminum brackets on the top and bottom of the extrusions, while they were attached to the chassis using sturdier brackets, both of which were acquired ready-made. A closer view of the left wheel has been attached below.

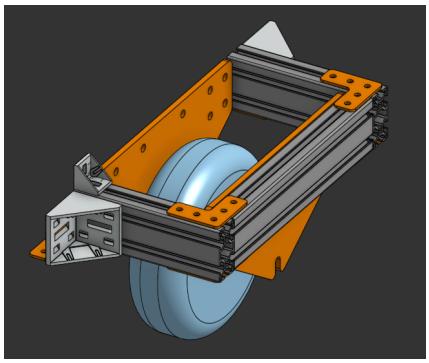


Figure 4: Left Rear Wheel Assembly

During chassis testing, the mechanical subteam confirmed that the robot can withstand the payload weight and maneuver in a steady manner for the entirety of its run.

2.2.2 Housing

The electronics and power system are housed in a wooden box made of ready-made cut plywood, while the sensors are installed on the top panel as shown in Figure 2b. The platform the payload rests on is made of the same material. To access the electronics, ready-made triangular hinges were acquired and installed between the back and top wooden panels. Initially, the wood was attempted to be laser-cut in-house; however, the resultant pieces always had slanting edges, causing the robot

to, one, look shoddy and, two, not allow for symmetric camera placement, which brought forth perception errors during the initial test run. Smaller hinges were acquired at first, and custom hinge adapters were designed and printed for use on the vehicle. However, during testing it was discovered that the M4 screws used to install them, with nuts, were not strong enough to support the weight of the top panel and were weak against any vibrations that would reach them. The electrical box and payload platform were first attached to the frame by drilling holes on the edges of the wooden panels and screwing them together using sliding slot-nuts that were ready-made. While this worked for the payload, the vibrations that would reach the electrical box would disturb the information the vehicle received from the cameras. To combat this, the mechanical subteam installed a suspension system, which will be further discussed in the sections below. The electrical power system that breathes life into the robot and the control system for vehicle motion are covered in Section 4.

2.2.3 Suspension

A major roadblock the mechanical subteam ran into late in testing was the lack of vibrational dampeners present on the robot. Trying to adhere to mechanical simplicity, the subteam decided not to rebuild the chassis to accommodate wheel suspension and instead focus on the most affected area: the electrical housing box. This box contains nearly all of the electronics, sensitive wires, and the cameras. Vibrations to this box could cause unwanted consequences, such as disconnected wires or poor video feed. Initially, thermoplastic polyurethane (TPU) shocks were designed and installed on the vehicle but did little to solve the problem. To combat this, the subteam decided to install four ready-made vertical shocks on the inside of the robot to dampen movement in the z-direction. With further testing, four ready-made linear rails were attached to the outside of the box via 3D-printed mounts to help constrain unexpected motion produced by the shocks.

2.2.4 Weatherproofing

The mechanical subteam determined the most severe weather the vehicle will endure is light rain and went about weatherproofing the robot accordingly. The sensitive electronics such as the motor controllers, Jetson, fuse box, relays, and power distribution board are inside the electrical box, providing protection. As for the ZED camera, LiDAR, GPS module, and status LED that are mounted on top of the panel, they are all IP-rated to withstand light rain. The mechanical e-stop and its voltage converter, both of which are mounted outside of the box, also have an adequate IP rating to not be affected by light rain.

3 Safety

3.1 Section Overview

This section will delve into the safety measures installed on the robot to ensure a clean run takes place with no damage to the vehicle's surroundings, chassis, sensors, or electrical hardware.

3.2 Safety Requirements

3.2.1 Mechanical E-stop

The mechanical e-stop is one of two e-stops that are installed on the robot, as both are requirements that the team agreed upon. Whenever the robot is not in use between trial runs or autonomous training, the e-stop is always turned on to ensure the battery does not lose unnecessary charge and

to prevent any unwanted electrical accidents. The mechanical e-stop is used to force-stop the robot at any moment regardless of the state of the robot and bring it to a full halt. This e-stop is the primary of the two e-stops that is used and has thus far not caused any harm to the circuitry or to the battery of the vehicle. These random activations have helped confirm that this e-stop fulfills its role on the Queen Bee and forces the robot to come to a full halt with no subsequent damage to any hardware on the robot.

3.2.2 Wireless E-stop

The wireless e-stop is the second of the two e-stops installed on the robot. During full-course autonomous runs, this e-stop is used to forcefully halt the robot from a distance to prevent any accidents during the tests. This e-stop works by the key fob sending a signal to a relay, which cuts off power from the battery to the rest of the circuit, thereby forcing the robot to come to a full halt. Through random activations of the e-stop during practice runs, as stated in 1.2.1, the range of this emergency stop has been confirmed to be at a minimum of one hundred feet.

3.2.3 Fuse Box

The fuse box is another stopgap designed to provide the vehicle with overcurrent, short-circuit, and overvoltage protection. The box contains different fuses for each part of the circuit to accommodate the specific current draws. This was useful during initial electrical testing, preventing the motor controllers from overcurrent surge, and will be useful during competition runs should any unforeseen accidents take place.

3.2.4 Relays

Throughout the circuit, there are relays used to secure safe and quick cut-off of power to the circuits. The first is a latching relay which is the wireless e-stop and is controlled by the wireless receiver. The e-stop ensures quick disconnects and immediate halting of movement from the robot. Next, there is one relay each for both motor controllers that ensure all phases of the motor controller are safely disconnected from the robot and prevent back-powering if power is suddenly disconnected and the robot is still in motion.

4 Electrical Design

4.1 Introduction

The electrical design of the robot went through multiple iterations, where different motors, motor drivers, and custom PCBs were tested before the final electrical design was settled on. The electrical design consists of both off-the-shelf components and custom-designed and soldered ones. Below, Figure 4.1 shows a simplified diagram of the complete electrical diagram of our robot.

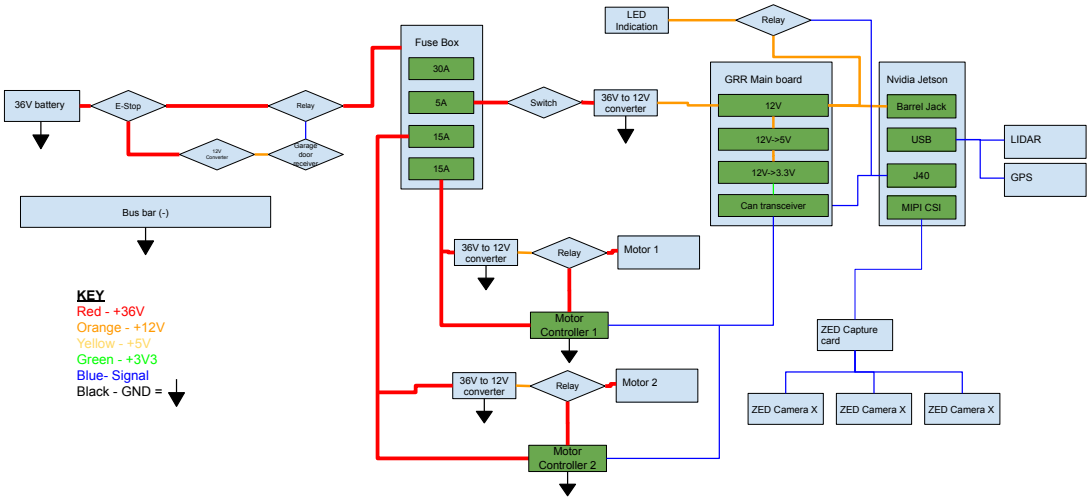


Figure 5: Complete Electrical Schematic of Robot

4.2 Significant Electrical Components

4.2.1 Battery

For our battery, we used a 36V 20Ah lithium iron phosphate battery which connects using an XT60 connector. This large battery allows us to have a long run time, and it is also able to keep up with our larger current demands. The battery also has a built-in BMS, which has safety features such as cutting power when the battery voltage is too low.

4.2.2 Wireless E-Stop

For our wireless E-Stop, a custom solution was created from off-the-shelf parts by one of our members. The design consists of a wireless garage door opener, a small 36V to 12V converter, and a latching relay. When the up button is pressed on the remote, it engages the relay. Even once power stops flowing to the relay, it stays latched until the down button is pressed, which applies a negative voltage and unlatches the relay, cutting power to the rest of our robot.

4.2.3 Motors and Motor Drivers

For our drivetrain, 36V scooter motors were chosen due to their motors being built into the wheels, allowing for a more plug and play solution. The motors also have physical drum brakes, which allows for a second form of braking along with the electronic braking. Each motor is controlled by a Flipsky

ODESC V4.2, which allowed for a lot of software customization and setup, such as tuning the PID values of the motors for optimal performance. It also included many features, such as over-current protection, and recharging of the battery from braking.

4.2.4 12V Power Distribution

For the 12V portion of our robot, which consists of our compute and sensors, an off-the-shelf 36V to 12V converter was used, which then gets connected to a custom-designed PCB. The purpose of this PCB is to provide 12V power distribution and also CAN communication. The 12V power gets distributed to our LED indicator for when our robot is running, as well as our compute, using XT30 connectors.

4.2.5 CAN Communication

For the CAN communication, 3.3V power is required. For this, a TPS565208 buck converter was put on the PCB along with supporting hardware to power a TCAN3414 CAN transceiver. All communication between our compute and motor controllers was performed over CAN.

4.2.6 Compute and Sensors

For the brain of our robot, an NVIDIA Jetson AGX Orin was used. This is powered from our 12V distribution board and has multiple sensors connected to it. The Jetson allows us a lot of headroom due to its powerful GPU and CPU combination. The sensors attached to it include three ZEDX stereo cameras, an RPLIDAR C4, and a u-blox ZED-F9P GPS module. The sensors combined together allow the robot to know where it is, detect lanes and cones, and complete the AutoNav portion of the competition.

To power our robot, we use a 36V 20Ah lithium iron phosphate battery connected through an XT60 connector. With this high voltage and large capacity, our robot is able to run for an estimated 4–5 hours continuously in our testing, before the built-in BMS shuts off the system when it detects the battery voltage is too low. We chose 36V since we wanted to use multiple scooter wheels as our drive base for the robot, and many of them run off 36 volts. The charge time for our robot is about 10 hours with a 2 amp charger, and 5 hours with a 4 amp charger.

Before power reaches the fuse box and the rest of the robot, we have two separate emergency shut-off switches on our robot. Our first is a physical E-stop button. The second is a wireless E-stop, which allows us to remotely cut power to our robot. The wireless E-stop was a custom design from one of our electrical team members. The design consists of a wireless garage door opener, a small 36V to 12V converter, and a latching relay. When the up button is pressed on the remote, it engages the relay. Even once power stops flowing to the relay, it stays latched until the down button is pressed, which applies a negative voltage and unlatches the relay, cutting power to the rest of our robot.

Once power passes the E-stops, it then goes to a fuse box for safe power distribution. Each fuse is sized accordingly to the load that it is going to. For example, each motor has a 15 amp fuse to accommodate the large power draw of the motors at full stall. Each motor is inside the hub of the wheel, the same as it is in a scooter. This provided us with an easier solution that was more plug-and-play for our setup. Each motor is controlled by a Flipsky ODESC V4.2, which allowed for a lot of software customization and setup, such as tuning the PID values of the motors for optimal performance. It also included many features, such as over-current protection. The main 36V to 12V converter has a 5 amp fuse.

4.3 Design Decisions

4.3.1 Battery and Voltage

When deciding on the battery, the battery voltage, capacity, chemistry, and safety features were all considered. The battery voltage was determined almost entirely by the type of motor we were trying to use, which were scooter motors. These motors ran on 36V, so we constrained ourselves to a 36V battery. The chemistry of the battery was chosen to be lithium iron phosphate due to it being much more stable than lithium polymer, and much less heavy than lead acid. The capacity was chosen to be 20 amp-hours due to us being unsure of how much power we would be drawing at first, so we went with a large-capacity one in case we drew a lot of current.

4.3.2 Motors

For our motors, we had already limited ourselves to 36V scooter motors. The reason being is because these eliminated the need for any chain or pulley to be needed and a separate wheel, so it allowed for a simpler and more compact design. When choosing motors, we had bought multiple different scooter motors and tried them out using a motor driver. In the end of our testing, we chose our specific motor for 2 reasons. The first reason being that it produced much more torque than the rest. The other motors would only spin on the ground if a kick start was given, which was not acceptable in our case. The second reason being that it had mechanical brakes, which gave us the possibility of designing a system of engaging our brakes if an E-stop was activated.

4.3.3 Motor Controllers

For the motor controllers, we had originally planned on using generic motor drivers that were found on Amazon. However, throughout the fall semester, while these were being used, they proved to be highly unreliable. The motor drivers would sometimes seem to quit working out of nowhere, and they allowed for very little control and did not have any smart features, as the only way to control the speed was by providing an analog voltage. We later switched to a Flipsky ODESC V4.2, due to it having native CAN support, motor calibration, PID tuning, and many safety features built in. With thorough testing, the team has been able to verify the motor controllers, as per the requirement in Section 1.2.1, can alter the speeds of the wheels efficiently and vary the speed of the robot from one to five miles per hour.

4.4 Power

4.4.1 Max Capacity, Run Time, Charge Time

With our chosen battery and connector, the max current it would theoretically be able to sustain would be 30 amps continuous. The main wire that was used from the battery to the fuse box was a 10 AWG wire. This allowed us to stay safely within the current limit of what we needed. On average, the robot will draw about 5 amps when going 5 miles per hour and running a full load on the Jetson. The electrical system was very overbuilt for the current it was meant to handle, in order to make sure the system is in no way ever overstrained, which may pose a safety risk.

When doing testing of our robot, we managed to get about 4–5 hours of continuous runtime from our robot, while it was driving around and performing its AutoNav tasks. Whenever the battery ran out, it was removed from the robot and connected to an external charger. Since our charger was limited to 2 amps, it took between 8–10 hours for the battery to fully recharge.

4.4.2 Safety Features

The electrical system contains multiple safety features, mostly concentrated in the 36V portion of the robot. To start, the battery itself contains a BMS. This makes sure all the cells of the battery are performing as they should, and it completely cuts power if the battery voltage drops too low.

There are two separate E-stops. The first is a physical E-Stop, while the second is a wireless E-Stop, which allows us to remotely kill the robot. Each portion of the robot that uses 36V has its own separate fuse, such as each motor and the main 36V to 12V converter.

Each motor also has a triple-pole triple-throw relay that goes to each phase of each motor. The idea is that the relay is powered by a separate, much smaller 36V to 12V converter, which, when it loses power, will disconnect each phase of the motor from the robot and motor controller. The idea behind this was to prevent possible back-powering of our system if the robot was to be in motion while the E-stop was activated. This would prevent any possible damage from occurring to the robot. Additionally, in software there are current limits that would completely turn off the motor driver if the current limit is exceeded by the motors.

5 Perception

5.1 Perception Overview

In this section, the logic behind how the Queen Bee recognizes lines, lanes, obstacles, and signs will be discussed in detail for both AutoNav and Self-Drive. This will further extend to how the robot recognizes and builds the map in the brain and uses that to train itself for future runs.

5.2 Software Architecture

The top priority of the software subteam was porting over ROS 2-based software to the Queen Bee's platform. Since the mechanical system will go through changes during the design cycle, the subteam's focus was on making the software architecture modular, which also helps in allowing the team to test in simulation without being dependent on a physical robot being built. The development pipeline was split into three parts: simulation, testing suite, and deployment. ROS 2 was the main software framework used for both the simulation and the physical robot.

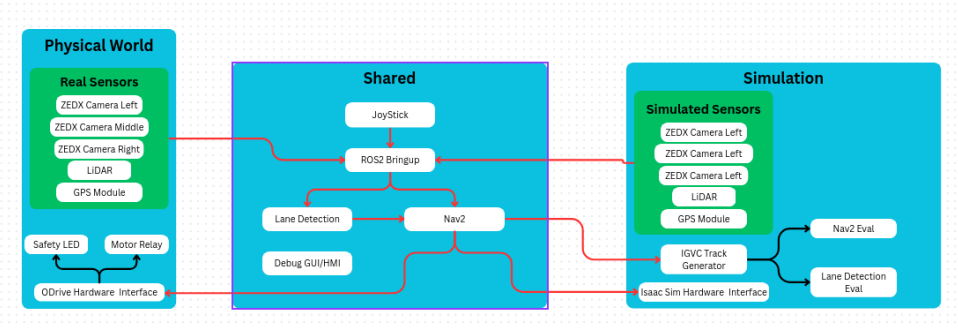


Figure 6: Queen Bee Software Architecture

5.2.1 Sensor Data

The sensors used on this robot were three ZEDX cameras, one Light Detection and Ranging (LiDAR) sensor, and one Global Positioning System (GPS) sensor, with the Jetson AGX Orin acting as the brain.

For lane detection, the team uses YOLOPv2, an open-source model for lane segmentation. The approach was centered around the idea that we did not want to have to tune HSV parameters throughout the competition and wanted a robust solution. First, the preprocessed ZEDX camera RGB frames are subscribed to and are then Gaussian blurred, and Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to the image. With this technique, the robot is able to extract lines in various conditions, which are then combined with the paired depth frame to map the location in an occupancy grid.
For odometry, the depth and inertial measurement unit (IMU) data from the ZEDX cameras and the LiDAR are fused with the GPS data using a ZED ROS 2 wrapper, which generates fast-updating and highly accurate odometry data. For both AutoNav and Self-Drive, a custom model was trained using open-source datasets in Roboflow, a software platform that generates synthetic images. The dataset contained over six thousand images and thirteen classes, some of which included people, traffic barrels, and stop signs. This model is then converted to an Open Neural Network Exchange (ONNX) file, an open-source format used for machine-learning models, and loaded through the ZED ROS 2 wrapper. During the Queen Bee's run, every time the model detects an object, the depth is calculated by the ROS 2 node and is then updated onto the robot's lane-detection costmap or occupancy grid. Due to the high-frequency costmap, dynamic objects like pedestrians are detected immediately, resulting in the brain communicating to the motor controllers to stop the vehicle until the pedestrian is no longer visible.

5.2.2 Perception Requirements

There were two major requirements that drove the software design for the Queen Bee: lane and object detection. By fusing the sensors as mentioned above, the robot is able to accurately detect and distinguish between multiple objects at the same time, which satisfies the aforementioned requirements. Images will be attached below to show the success the subteam has had with its object and lane detection model during test runs.

6 Driving Logic

6.1 Driving Logic Overview

The costmap, mentioned in Section 5.2.1, which is created by the fusion of ZEDX camera data, LiDAR data, and GPS waypoint data, is used by the robot to make decisions on how to maneuver the course in the most efficient manner. The driving logic for the Queen Bee is a combination of the Nav2 stack with the SmacHybrid Planner and Regulated Pure Pursuit. Using the costmap, the robot creates many routes using the nearest GPS waypoint by sampling carrot (look-ahead) distances over a distribution, and these are then weighted based on speed, distance from obstacles, and progress made. The selected path is then smoothed out and scaled, and finally the Jetson sends commands to the motor controllers over CAN on how to traverse the selected path.

6.2 Driving in AutoNav

For the AutoNav challenge, the depth data and object detection model that are running from the fused sensors are used to first detect lines and obstacles and second figure out how far away the robot is from them. For complex obstacles such as the ramp and potholes, detection is more dependent on the depth data collected, as this will inform the robot of any change in elevation from the road. Additionally, since the detection model is also trained on potholes, the costmap will be updated accordingly and the robot will map a route to avoid them. As mentioned in the previous section,

GPS waypoints are used to figure out the fastest route to progress across the track, and the robot moves accordingly by communicating with the motor controllers to dictate the speed of the two rear wheels.

6.3 Driving in Self-Drive

For the Self-Drive challenge, similar to AutoNav, the majority of the obstacles are detected and avoided with the help of the fused depth data and object detection model. The model is trained on various stop signs and detects them with great accuracy. For dynamic pedestrians, when the robot detects an object that is constantly changing its position in the costmap, it will come to a stop and wait until the object is no longer detectable and then continue.

6.4 Driving Logic Requirements

The two requirements for the two challenges that drove the design for the Queen Bee's driving logic are lane following, waypoint navigation, lane changing and parking, and compliance with the rules of the road. For lane following and changing to exist, the team decided to use the ZEDX cameras as they process the images they capture as point-cloud images, which is required to calculate the depth data and fuse that with the depth obtained from LiDAR. For compliance with the rules of the road, object detection is crucial, which can also be solved by feeding the input images from the ZEDX cameras into an object detection model. Finally, waypoint navigation can be solved with the installation of a GPS module. As of the most recent tests, the Queen Bee performs trial runs with no lane-crossing violations, and the GPS feedback quality has increased to now accurately track and locate waypoints.

7 Key Performance Indicators

7.1 AutoNav Indicators

For the AutoNav challenge, the team identified two key performance indicators: course completion time and speed compliance. Speed compliance is considered satisfied as long as the robot maintains a consistent average speed between one and five miles per hour throughout each run, in accordance with the IGVC 2026 rules. For course completion time, the team set a target of six minutes for the Queen Bee to complete a full competition run. This target was established following a review of previous competition results, as six minutes is approximately double the recorded time of the prior year's winning entry. The team has deliberately set this as an initial benchmark, with the intention of incrementally improving upon it as the development cycle progresses. As of the latest testing, the Queen Bee has recorded an average completion time of ten minutes, indicating meaningful improvement is still required to reach the six-minute target prior to competition.

7.2 Self-Drive Indicators

For the Self Drive challenge, the team identified two key performance indicators: function test score and full course completion score. The function test score measures the robot's ability to autonomously complete each of the sixteen individual Self Drive functions independently, while the full course completion score measures how many of those functions the robot can successfully execute in a single continuous autonomous run. As of the latest testing, the Queen Bee has demonstrated the ability to complete fourteen of the sixteen function tests autonomously, with the curved road

evaluation functions representing the two remaining outstanding tasks. For the full course completion score, the team set an initial target of completing fifteen of the twenty-one course functions in a single autonomous run. The Queen Bee has surpassed this target, achieving a best result of sixteen functions completed in one autonomous run. While the goal was accomplished, the team recognizes that further improvement is necessary and has set the next objective of progressively increasing the number of completed functions toward a full twenty-one function run prior to competition.

8 Analysis of Complete Vehicle

8.1 Lessons Learned During Design Cycle

One of the trials the team faced was a lack of mechanical members. This team mainly consists of electrical and software members, which caused delays in the design and building of the robot and certain crucial systems like suspension. It also forced the team to go through more mechanical iterations than needed and encroached on the software subteam's time to train and test the robot, since most of the time went into building or fixing parts of the robot. The team eventually overcame this hurdle by designing and installing a suspension system, but from future seasons onward the group should look to recruit more mechanical talent.

8.2 Key Failures and Mitigation Strategies

8.2.1 Motor Controllers

Early in the design phase, the team had decided to use Pololu brushless motor controllers due to their high power output. However, during initial electrical testing these controllers would keep overdrawing current and shut off. It took a long period of debugging and reading the Pololu forums to realize that the motor controllers had a lower current limit than what the motor required to run, resulting in an overcurrent surge, which is what caused them to shut down. We solved this problem by acquiring the Flipsky ODESC V4.2 motor controllers, which could handle the required current draw and had better protections compared to the Pololu motor controllers.

8.2.2 Teensy

The Pololu motor controllers communicated with the Teensy 4.1, which relayed the commands given by the Jetson and also provided sensor feedback. We had multiple Teensy 4.1s shut off during our debugging cycle. Due to time constraints, complete debugging of the failure could not take place; however, the current hypothesis is that the voltage converter powering it was unstable. When we switched over to the Flipsky ODESC V4.2, these motor controllers had an inbuilt CAN transceiver that allowed the team to bypass the Teensy 4.1 and let the motor controllers directly communicate with the Jetson. This resulted in the electrical subteam decommissioning the Teensy 4.1 from the electrical architecture.

8.2.3 Rear Driven Wheels

The first set of driven wheels acquired were from an e-scooter, most of whose in-hub motors expect a kick-start before moving due to low torque inducement. Therefore, the mechanical subteam searched for another in-hub motor-in-wheel that was closer to the required torque rating.

8.2.4 Electrical Box

As discussed previously in Section 2.2.2, the first iteration of the electrical box was made of laser-cut wood of incorrect dimensions. This resulted in a faulty electrical box that would neither close all the way nor allow for the correct mounting of many components like the ZEDX cameras. This was solved by acquiring custom-ordered cut wood from a hardware store.

8.2.5 Hinges

The initial hinges the team acquired were not the desired length. One of the solutions that was tested at that time was custom-designed hinge adapters that were 3D-printed in-house. The reason for installing hinges, which would go between the back and top wood panels of the electrical box, was to provide ease of repairability to the electrical box. However, these adapters were also undersized, causing the top wooden panel to detach itself from the electrical box during one of the test runs. So, the team decided to buy door hinges of the appropriate size, solving this issue.

8.2.6 Suspension

The initial iteration of the robot lacked any type of suspension, which resulted in noisier sensor feedback from the ZEDX cameras. Due to time constraints, a suspension system attached directly to the drivebase could not be installed, resulting in the team trying to dampen only the electrical box since that's where the cameras are mounted. The initial solution to this was installing 3D-printed dampeners made with TPU, which did little to solve the problem. The next iteration of solutions was four shock absorbers from goBILDA that were installed in the four corners of the electrical box, connecting it to the frame of the robot. While these shocks dampened movement in the vertical axis, there were still large movements in the horizontal and rotational axes. To combat this, four linear rails were attached to the outside of the electrical box and to the frame using custom-built 3D-printed mounts, which suppressed the unwanted movement.

8.3 Software Testing, Bug Tracking, Version Control

The software subteam wanted a comprehensive testing suite to get quantitative results on the performance of our code. This led to the creation of the IGVC track generator. This generator creates a unique track, object placement, and Universal Scene Description file that can be imported into Isaac Sim, the simulation platform that the team uses. For bug tracking and version control, the team employs GitHub and its services, where most of the version history is managed by the software lead.

8.4 Simulation-Based Testing

As mentioned previously, the simulation platform used by the team is NVIDIA Isaac Sim, as it provided very realistic physics and scene-rendering capabilities. All of the sensors on the robot have native and third-party support out of the box for this platform. The Onshape importer and robot rigging tools allowed for quick prototype designs in simulation before manufacturing them. Due to the majority of the team consisting of electrical and computer engineers, simulations were heavily relied on to validate designs.

For sensor simulation, three ZEDX cameras, one GPS module, two wheel encoders, and one LiDAR needed to be simulated. For the cameras, Stereolabs, the manufacturer, provided a Stereolabs extension that allows the simulated cameras to use the native ZED ROS 2 wrapper. This allows the simulated depth clouds and RGB images to closely resemble what the real robot experiences, and the simulation output was considered raw camera output and funneled through the wrapper like

the real Queen Bee. For the GPS module, there is a version of Isaac Sim made for drones called Pegasus Sim. By following the Omni extension made for that project, the GPS module was able to be simulated and published like it was on the real robot.

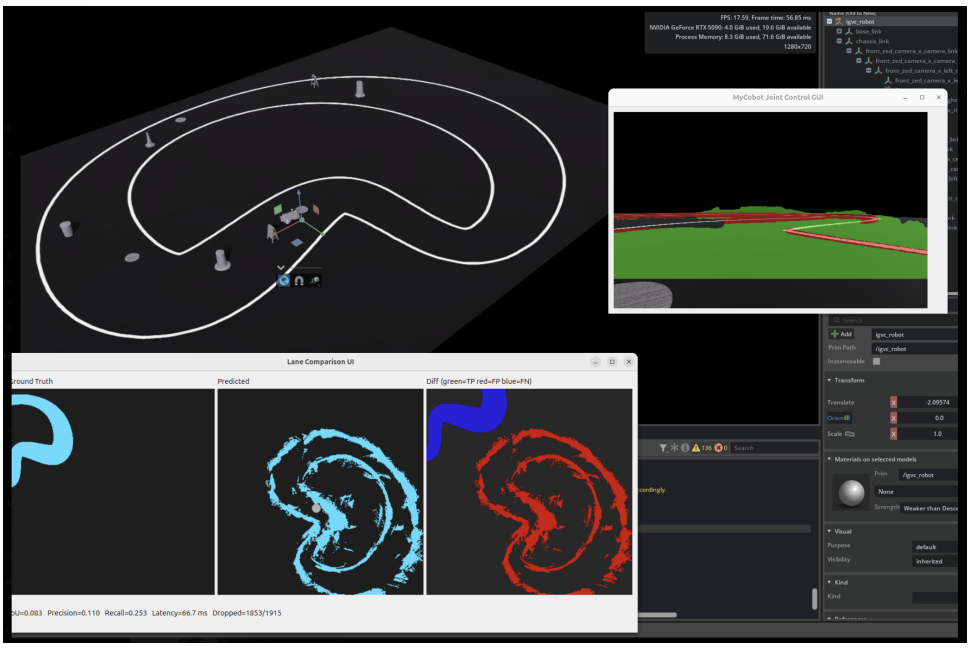


Figure 7: Simulated Sensor Feedback in Isaac Sim

8.5 Physical Testing and Performance

Due to the Isaac Sim platform allowing all the simulated sensors to publish sensor feedback as the real robot would, there is a negligible difference in actual performance from simulation predictions. While there are errors that do occur during testing, they are due to the difference in processing power between a PC and a Jetson.