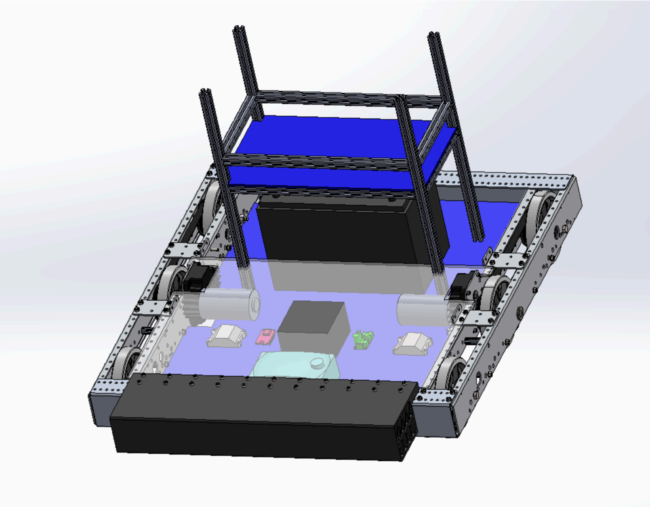
Texas A&M University - Kingsville

TEXAS A&M UNIVERSITY KINGSVILLE-ROBOHOG



5/15/2026

Team Captain :
Xavier Gonzalez
uni.xavieragonz@gmail.com

Team Members:
Jason Besancon (Jasonbesancon12345@gmail.com),
Ishan Prashant Pathak (Ishan_prashant.pathak@students.tamuk.edu),
Andrew Ybarra (Andrew.ybarra@students.tamuk.edu)

Advisors:
Dr. Haitham Adarbah
Dr. Dunren Che

Competitions:
AutoNav

Statement of integrity:

We, the undersigned, affirm that the contents of this report for the Intelligent Ground Vehicle Competition (IGVC) represent our original work. All designs, analyses, and conclusions presented were completed by our team in accordance with the rules and expectations of the IGVC. Any external sources, assistance, or references used have been appropriately acknowledged. We understand the importance of ethical conduct in engineering practice and certify that this report reflects our honest and independent efforts.

System and Subsystem Requirements

We redesigned the layout of our vehicle from last year's competition. We replaced a couple of electrical components as well as making more of the inclusions more permanent. The idea was to make the vehicle more reliable, as last year we had a few electrical issues such as our solid state relay shortening, and inconvenient problems like wires unplugging. Our vehicle's size is 3ft x 3ft as it will allow the vehicle to maneuver, when taking into account the minimum 5ft road clearance, with ease. The hardware and software was selected based on convenience with the Nvidia Jetson Orin, our main computer. This year we also included the VectorNav VN-200, which will be our GPS and IMU.

Our goal in participating this year is to build upon our strong foundation for future IGVC competitions, demonstrate our technical capabilities, and proudly represent the spirit of Texas A&M-Kingsville on the international stage.

Part	**Cost**
Jeston Orin	$2000.00
Inertial Measurement Unit (IMU) (3 Pack) x2	$10.99/each
Through Bore Encoder x2	$48.00/each
ESP32 (3 Pack)	$19.99
Fuse Block	$39.99
Boost Converter	$16.14
Buck Converter (2 Pack)	$8.99/each
100A Circuit Breaker	$10.79
Coaxial Power Plug x2	$9.99/each
E-Stop Button x2	$9.99/each
Solid State Relay	$24.95
Push Button Starter	$10.99
Inline Fuse Holder (6 Pack)	$8.99

Part	**Cost**
Realsense Depth Camera D435	Donated
Power Supply	$200.00
Voltage Regulator	$14.24
VectorNav VN-200	Donated

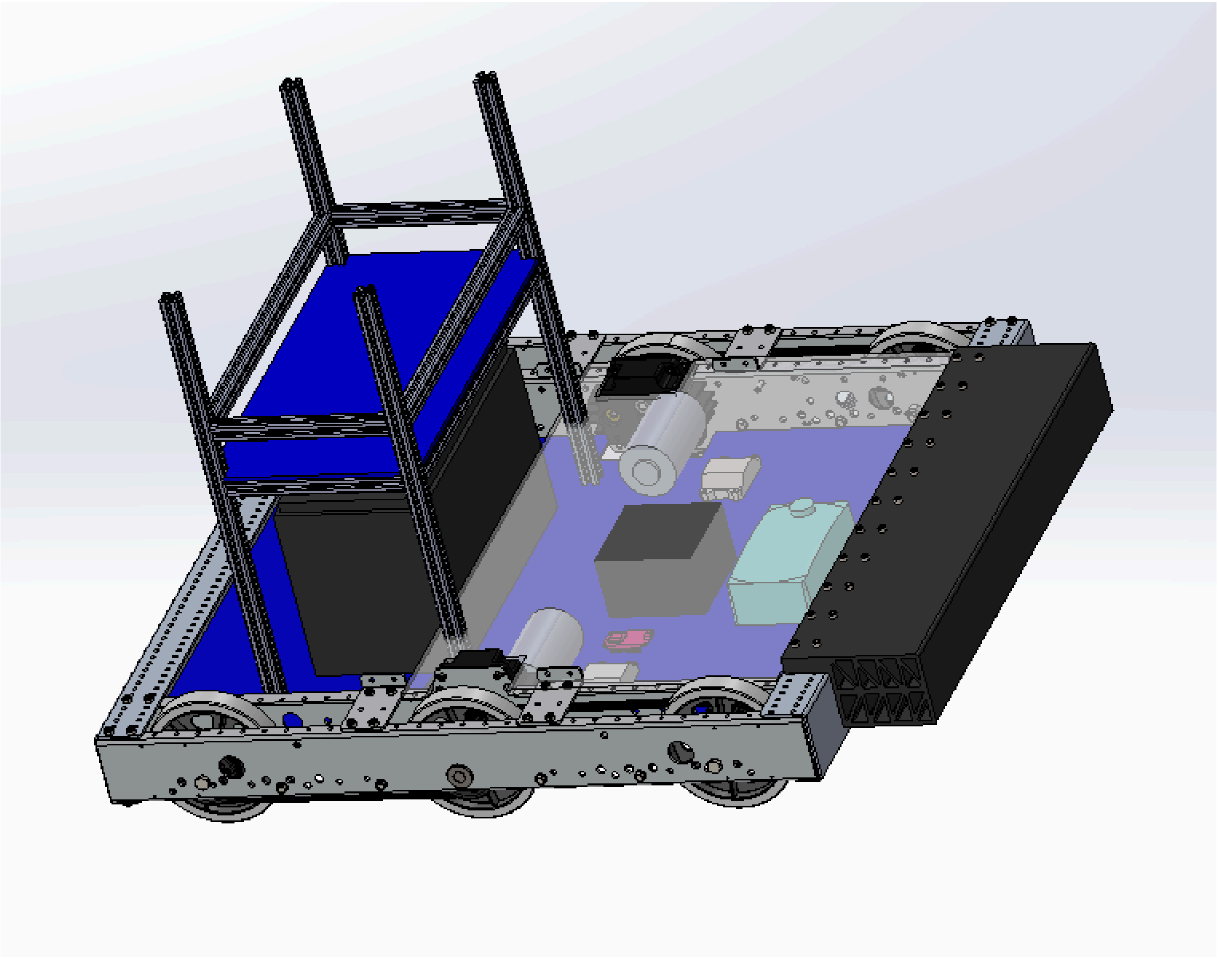
Mechanical Design

The Chassis/Drivetrain was made from a kit, AndyMark AM14U3 because of its durability and ease of design adaptations. The AndyMark AM14U3 drivetrain frame is a modular platform frequently used in competitive robotics. Its robust design, excellent strength-to-weight ratio, and durability make it ideal for supporting a variety of drivetrain components, ensuring both dynamic performance and energy efficiency. This chassis was selected for its ease of integration and maintenance, along with its ability to withstand high operational stresses and facilitate rapid design iterations, qualities that are essential for meeting the rigorous demands of autonomous vehicle competitions. The first part of the assembly process was to assemble the chassis to build up on.

For the base of the vehicle a fiberglass panel was manufactured to increase strength and efficiency. For the panels/floors of the vehicle, fiberglass panels were selected because of the strength, durability, and increased efficiency. Making the fiberglass panels involved cutting sheets of glass fiber to form layers then calculations must be done to know how much resin is needed for the manufacturing of the panel. These calculations involved finding the volume of the panel and then multiplying by the density of the resin to find how much mass of resin is needed. Once the resin is prepped the glass fiber is infused with resin using the vacuum bag method, where a vacuum pump causes the flow of the resin throughout the panel. The panel is then left to cure and then cut to size. The first iteration of testing used plywood so that the base could go through multiple iterations. Once the fiberglass panel was finished it was installed on the chassis as the base.

Manufacturing fiberglass panels is a significant innovation for our self-driving car project because it allows an optimization of both performance and efficiency from the ground up. Fiberglass has an impressive strength-to-weight ratio meaning it holds weight exceptionally well while reducing overall mass. By producing these panels in-house, we maintain strict control over material quality, enabling us to fine-tune the balance between durability and lightness. This not only improves the vehicle's dynamic performance and energy efficiency but also ensures that the body can withstand various stresses without compromising structural integrity. Additionally, in-house production reduces dependency on external suppliers and allows for rapid iterations and customizations suited for competition standards. Fiberglass panels also do well in harsh weather and climates making it a great choice for an outside vehicle.

For the placement of the components a Computer-Aided Design (CAD) model was used to conserve space and keep center of gravity acceptable. All CAD modeling was modeled in SolidWorks.



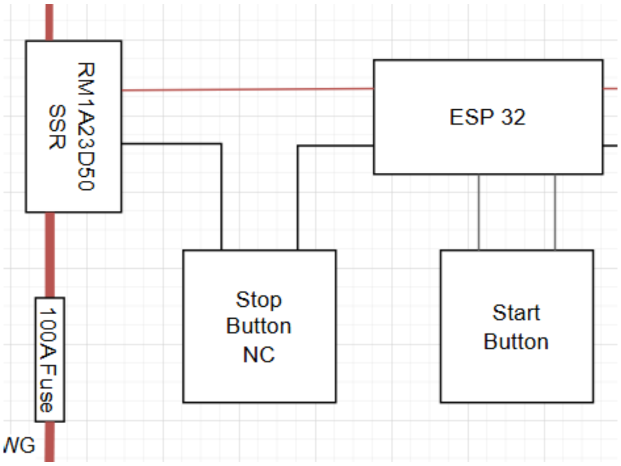
A second floor of the vehicle was designed using T slot extrusions and a fiberglass panel to keep the payload as well as to mount the camera someplace high. The next step in the process since the chassis, base, and electronic placement was completed, was to design the second story where the payload would sit. Some of the main details looked at included how to make it to where the battery would be accessible and what material to use. The battery was given enough space to be able to be taken in and out of the first story and brackets were utilized to fit the battery for minimal movement. For the supporting material metal T slot extrusions were selected because of their modularity, strength, and ease of reconfiguration. This allowed for quick testing and redesigning. Another Fiberglass panel was manufactured to use as the baseplate to hold the payload.

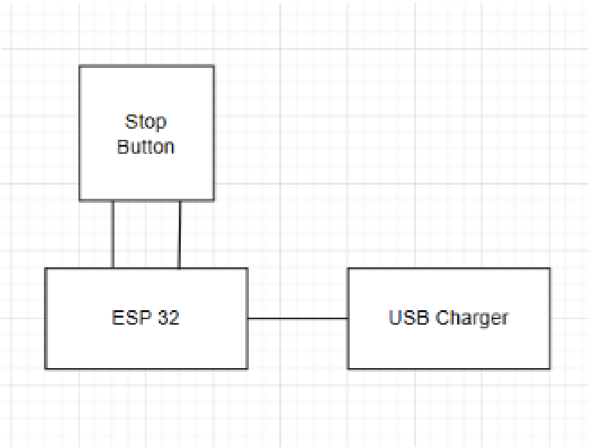
To weatherproof the vehicle an acrylic top was mounted as shelter, a marine sealant was used to seal the edges, and silica packets were used to absorb water in the air inside the electronic housing. With the robot being fully built, weatherproofing was now the focus. A clear acrylic sheet was used as shelter for the electronics. A marine adhesive sealant was chosen to seal up the electronics. This specific sealant was chosen because it is one-part, moisture-curing polyurethane-based sealant which has been engineered to form a watertight barrier that protects electronics from rain and environmental exposure. Its chemistry centers on functional isocyanate groups that react with ambient moisture, triggering a polymerization process that forms a dense, cross-linked polyurea network. This network not only provides robust adhesion to substrates such as fiberglass, metals, and plastics but also creates an elastic, cohesive barrier that resists water ingress and absorbs mechanical stresses, including vibration and thermal expansion. By harnessing moisture as a catalyst for curing, the sealant transforms even trace amounts of water into an essential component of its protective mechanism. For further protection from moisture silica packets were used since silica gel is a porous form of silicon dioxide with numerous microscopic cavities. These cavities, lined with silanol groups, attract and hold water molecules through hydrogen bonding, effectively reducing moisture in the environment and protecting electronics from humidity. Any other housing was made using a 3D printer and designed using SolidWorks.

Safety

Our IGVC robot is equipped with multiple safety devices to ensure reliable and secure operation during autonomous travel. A central fuse box protects the system’s electrical components from overload and short circuits, providing critical circuit isolation. The robot features two emergency stop mechanisms: a remote E-stop that allows team members to disable the system wirelessly from a safe distance, and a physical push-button E-stop mounted on the chassis for immediate manual shutdown. These safety features meet competition requirements and are vital to safe testing and operation around people and obstacles.

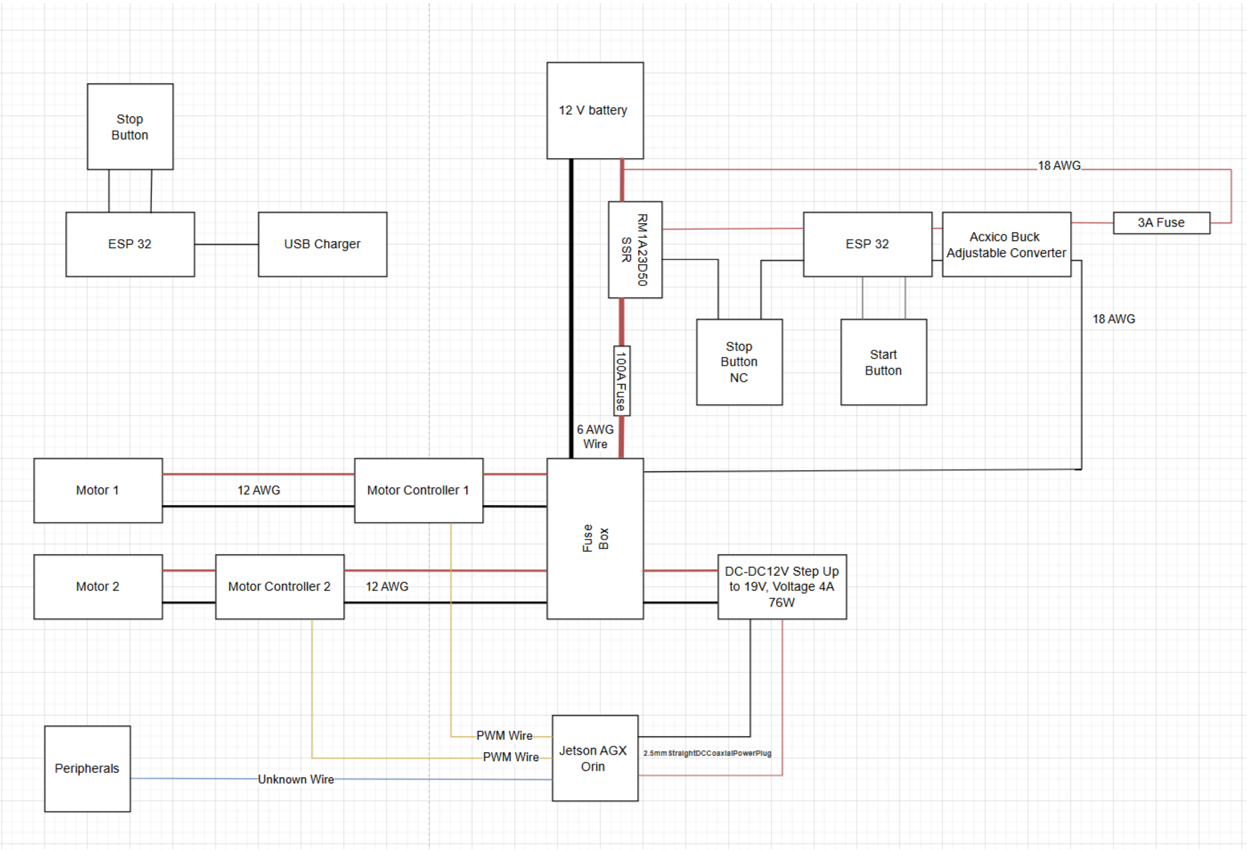
The stop button placed on the robot will interrupt the solid state relay's control power. Causing the entire robot to cease functioning. The wireless stop button will be connected via 2 ESP32s in ESP32NOW mode. The first ESP32 will be supplying the control signal to the relay and the second will be with the wireless EStop. The programing will also interrupt the control signal if the ESP32s lose connection which happens at a range of about 220 meters. This should make our EStop system functional up to 220 meters.





Electrical/Electronic Design

The primary power source of the IGVC Robot is a 12v LiFePo4 Battery. The primary feed of our battery can be interrupted by a solid state relay which will serve as our method for emergency cutoff. The primary feeds a fuse box from which all our electronic power branches to. The exception to this is an ESP 32 which branches off the main feed before our solid-state relay because it is responsible for turning the robot on and off.



There are 7 significant Power and Electronic Components:

- A DJLBERMPW 12v 100Ah battery Which will power our robot
- A simple fuse box This will allow us to protect all of our components from over-current
- A 250A solid state relay This allows us to turn the robot on and off in case of emergency
- A 19v 76w step-up power supply This allows us to power our Jettson Orin with a 12v Battery
- An adjustable Buck converter Allows us to lower 12v to 5v to power our ESP32
- The Jetson Orin AGX This Component is responsible for running our AI and communicating its control to the ESCs
- The ESP32 The ESP32 will control the Solid State Relay unless the onboard stop button is pressed.

Using CAD, the placement of the electronics, battery, and payload was designed. The battery and payload, being the heaviest of the objects being installed onto the robot, were carefully placed to try to keep the center of gravity as low and to the center as possible. To do this multiple iterations were tested. Finally the battery was placed towards the center and a second story of the robot was designed and manufactured so that the payload would be above the battery. This also was done so that the camera could be placed up high.

A DJLBERMPW 12v 100Ah battery is being used as our main powersource. Assuming an operational power of 300 watts. The ESP 32 and Jetson Orin should consume no more than 90 watts total in the worst-case scenario. Assuming each individual motor does not consume more than 105 watts the life of this battery would be 4 hours. 300 watts/12V = 25 amps. 100Ah/25 = 4 hours of runtime at maximum load.

Perception

Based on the past design reports published in the IGVC website, our team recognized a common use of the Light Detection and Ranging (LiDAR) sensor. Because of this, we decided to replace that component for the utilization of the infrared sensor within the Realsense camera to calculate distances. By stationing the camera at the front of the vehicle, it allows the program to recognize incoming obstacles, which then it proceeds to take action based on the situation. This method can be compared to the use of a Flash LiDAR, which captures a scene of distances based on the direction of where the sensor is pointed to as well.

RealSense D435 Camera provides: RGB image stream for object detection, Depth map for estimating obstacle distance using infrared sensing. IMU Output provides: Real-time orientation data for motion planning and trajectory estimation. All data is synchronized using ROS 2 topics and processed in realtime.

IMU data will be utilized by gathering pitch data of the vehicle to understand what slope it is on. This means we can make the vehicle speed up when on an upwards slope, and slow down on a downwards slope without going over the speed limit.

We decided to utilize a pipeline of machine learning models to recognize obstacles and lanes. As a frame is passed as the input to the pipeline, Semantic Segmentation is used to group up the pixels in a sort of “blob”, these will be used for recognizing lanes. Instance Segmentation will instead group up and classify objects, which will be used to recognize the different obstacles found throughout the course. The output results in the vehicle having a better understanding of its environment.

Last year we compared the location of HSV color pixels from the camera to locations of areas on the frame to see where the vehicle needed to steer. This time we are doing that, but using the output from the pipeline to determine steering.

Driving Logic

Our main processing unit is the NVIDIA Jetson Orin, which handles all inference and Robot Operating System (ROS) 2 node execution in realtime. It provides the Graphics Processing Unit (GPU) acceleration necessary to run deep learning models such as YOLOv8 while simultaneously managing sensor integration and decision-making processes. The software architecture for our IGVC robot is built using ROS 2 Foxy, running on the NVIDIA Jetson Orin platform and is structured around modular nodes that integrate perception, localization, decision-making, and control. The primary sensors include a RealSense D435 camera and IMU output, both interfaced via ROS. RealSense provides color (RGB) and depth information, which is used for object detection and distance estimation, while the IMU output is used for orientation and stability awareness. A state machine governs robot behavior for tasks such as lane following, object detection, and halting.

Lane following is implemented using custom OpenCV filters on the RealSense RGB stream. Infrared and depth data is used to differentiate between terrain features and obstacles. Mapping and trajectory estimation are integrated via a lightweight costmap generator. Scene data is updated incrementally to build a world model and avoid redundant re-processing.

Operating Modes: Manual Override Mode: PS4 joystick control; used during testing and emergency scenarios. Autonomous Navigation Mode: Object detection + lane following active; triggers trajectory generation. Stopped Mode: Triggered by detection of an object within halt range or E-Stop command.

The robot computes straight-line trajectories using IMU output orientation. The trajectory control node publishes cmd_vel commands based on the next waypoint. Navigation decisions are made by analyzing current scene, object locations, and depth map. A PID-style velocity controller is used to generate smooth acceleration and stopping behavior.

Creative Concepts: Real-time depth-aware halting via YOLO + infrared fusion, Multi-threaded ROS nodes for parallel camera and control processing, Adaptive parameter tuning using dynamic reconfigure for field flexibility. The integration of Jetson Orin allows fast GPU-powered inference alongside ROS 2 execution. YOLOv8 + depth fusion allows single-camera stop-and-go behavior without LiDAR. Safety stack includes heartbeat monitors, emergency stop override logic, and watchdog-controlled command flow.

Key Performance Indicators

Field Testing and Performance targets observed: Depth-based object stopping tested to within 100 ms halt accuracy. IMU-based heading tracking showed < 5-degree drift over 15 meters. YOLOv8 detection accuracy: > 90% precision on cones/barrels in test environment.

Analysis of Complete Vehicle

Software Testing: We conducted unit testing for detection, velocity publishing, and sensor node health. ROS 2 tools such as ros2 topic echo, ros2 bag, and rqt_graph were used for real-time debugging. Dynamic reconfigure was used for adjusting sensor thresholds during field tests.

Bug Tracking and Version Control: All software was maintained using Git and hosted on a private GitHub repository. Issues and features were tracked via GitHub Projects board. Commits required mandatory code reviews and CI checks for style and node launch sanity.

Simulation Testing (SIL): Basic testing with RVIZ2 for visual inspection of camera and IMU data. ROS bags from actual field runs were replayed for testing detection and decision logic. Planned integration of Gazebo for future closed-loop testing.