# Joyride: IGVC 2023 Self-Drive Design Report

Oklahoma State University

**Submitted:** 15 May 2023

| Team Member | Role | Email |
|---|---|---|
| Max DeSantis | Captain, Software | max.desantis@okstate.edu |
| Brenden Wickman | Electrical | brenden.wickman@okstate.edu |
| Isaac Castilleja | Software | isaac.castilleja@okstate.edu |
| Isabell Cook | Software | icook@okstate.edu |
| Kelvin Tran | Software | kelvin.tran@okstate.edu |
| Nathan Wilson | Software | natwils@okstate.edu |
| Jason Aquino | Software | jason.aquino@okstate.edu |
| Jack Funderburgh | Mechanical | jack.e.funderburgh@okstate.edu |
| Kale Downing | Mechanical | kale.downing@okstate.edu |
| Bailey DeSpain | Mechanical | bailey.despain@okstate.edu |
| Zeke Lappe | Mechanical | zeke.lappe@okstate.edu |
| Christopher Shropshire | Mechanical | christopher.j.shropshire@okstate.edu |
| Troy Willoughby | Mechanical | troy.willoughby@okstate.edu |
| Diego Colon | Graduate Mentor | dcolon@okstate.edu |
| Rushikesh Kamalapurkar | Faculty Advisor | rushikesh.kamalapurkar@okstate.edu |
| He Bai | Faculty Advisor | he.bai@okstate.edu |
| Weihua Sheng | Faculty Advisor | weihua.sheng@okstate.edu |

**Faculty Advisor Statement of Integrity**

I, Rushikesh Kamalapurkar, certify that the design and engineering of the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Sign: ________________________________________ Date: _______________

## 1 Introduction

Joyride is an autonomous vehicle developed by Oklahoma State University senior design students over the course of three semesters. Its foundation is a Polaris GEM e2 low-speed electric vehicle. The team designed, built, and installed a custom drive-by-wire (DBW) kit that interfaces with vehicle throttle, a power steering unit, and an electric-over-hydraulic braking unit. Furthermore, the team utilized the Robot Operating System (ROS) to develop remote control and autonomous navigation capabilities. As it is the team's first year competing, the primary objectives were foundational behaviors such as waypoint navigation, lane following, and obstacle avoidance.

The vehicle is comprised of several custom-printed circuit boards, an onboard computer, 2D LiDAR units, gigabit Ethernet cameras, a GPS/INS unit, wheel encoders, and a touchscreen interface.

## 2 Team Organization

Undergraduate senior design students worked on the project over three semesters from Spring 2022 to Spring 2023. Additionally, graduate student mentors assisted where possible. Some students participated for multiple semesters. Student work varied from 5 hours to 30 hours per week - cumulative effort was more than 3000 hours of labor.

The team was split into mechanical, software, and electrical teams. Each had a sub-team lead that delegated tasks, ensured timely delivery, and mediated design discussions. The mechanical team was responsible for sensor mounts, installing wheel encoders, and implementing autonomous braking capabilities. Electrical designed and built the custom PCBs and wiring harness that comprised the DBW system. Software used ROS to interface sensors, the onboard computer, and the DBW system to enable autonomous behavior. The teams utilized a shared sticky-note kanban board to track tasks and work distribution. Generally, each team member was given decision-making authority over their own task areas on the project.

## 3 Design Process and Assumptions

The primary goal of the team was to make a professional product capable of foundational autonomous behaviors. With that in mind, the team adopted a "do-it-right" as opposed to a "do-it-all" mentality toward each aspect of the project. The team also endeavored to provide robust documentation and transparency for the benefit of future Joyride and IGVC teams; this is accomplished by making all design and documentation elements freely open source and available. Primary ROS code is available at https://github.com/osu-igvc/joyride-ros-main.

Several assumptions were made about the vehicle's operation and environment. Namely, that the environment will be well-marked and that two GPS waypoints will be provided beforehand. The velocity control system assumes no wheel slip, no inclination, and an unchanging surface material.

## 4 Innovations

### 4.1 Drive-by-Wire

The team designed, built, and programmed a custom DBW system to interface with the vehicle. Two custom PCBs - the Drive Controller and Accessory Controller - allow for total control of braking, steering, throttle, lights, parking brake, and more. Steering and brake actuation is done by off-the-shelf components, but low-level control, communication, and spoofing of original systems is done by the custom boards.

### 4.2 Touchscreen Interface

Although other vehicles have user interfaces, Joyride features a touchscreen monitor attached to the dashboard with a built-from-scratch, feature-rich graphical user interface (GUI). The interface allows for a convenient, intuitive way to interact with the vehicle without requiring external hardware or technical savvy. The GUI is integrated with ROS and displays immediately relevant information concerning the vehicle state. It visualizes sensor feedback, streamlines preparation for vehicle testing, displays live system diagnostics, provides GPS waypoint interaction, and even plays music. Each of these features has a dedicated page that has been extensively tested and refined to maximize ease of use, functionality, and overall user experience.

### 4.3 Safety Takeovers

Joyride features numerous “safety-takeover” features that make it a safer, more experiment-friendly research platform. Should the driver press the brake or grab the steering wheel, the DBW system relinquishes control of the vehicle immediately. The system verifies the driver seat belt is engaged, gear selector and parking brake are in the correct position, and all systems are operational before taking over control. The E-stops not only disable the vehicle but also fully actuate the rear brake to stop the vehicle in approximately three feet. This is done fully in hardware included on the Drive Controller PCB. Also, note that the front and rear brakes are two separate hydraulic systems. The driver will always be able to stop the vehicle with the brake pedal, regardless of software or hardware malfunction.

### 4.4 Camera Shrouds

The forward-facing cameras are mounted inside the passenger cab, to improve aesthetics, simplify mounting, and improve weather resistance. This introduced significant dashboard reflection in the image due to the curved windshield. The team developed a novel 3D-printed camera shroud that greatly reduces reflection in image streams.

## 5 Mechanical Design

### 5.1 Overview

The mechanical team’s primary objectives were to install actuators, mount sensors, and otherwise modify the vehicle to be suitable for the competition. Reliability and maintainability are of utmost concern; notably, sensors must be replaceable with precise, repeatable mounting positions. The vehicle’s aesthetic is preserved and improved to lend greater professionalism. Over three semesters, several components were redone to improve reliability. The braking system, for example, underwent several iterations.

### 5.2 Vehicle Selection

The Polaris GEM e2 was selected due to its popularity within IGVC and other autonomous vehicle projects. Its proven reliability reduces time wasted on troubleshooting. This assumption has been validated, as the installation of DBW actuators has been less complex than previous autonomous vehicles at Oklahoma State University.

The suspension has remained unmodified to the original GEM. As for weather-proofing, the vehicle does not have fully-sealed doors. This is mitigated by selecting water-resistant (IP66+) sensors and connectors. Furthermore, critical electronics are installed inside the vehicle cab to reduce the probability of water ingress.

## 5.3 Drive-by-wire

The DBW system has three actuators: the stock front-wheel motor, a student-installed electronic power steering unit, and a student-installed autonomous brake.

### 5.3.1 Electronic Power Steering

An Allied Motion electronic power steering unit enables autonomous control of the steering column. It receives position and velocity commands over the CAN bus from the DBW system. The unit is prefabricated and a drop-in add-on, which was chosen due to simplicity and ease of interface.

### 5.3.2 Brake System

To improve safety and reliability, the team chose to separate the front and rear braking systems. This ensures the driver's brake pedal works, regardless of software or electrical failure. An electric-over-hydraulic master cylinder was installed to control the rear brakes, it receives a voltage signal from the DBW system and can actuate up to 1000 pounds per square inch. A pressure transducer is also installed to allow real-time monitoring and feedback for the velocity controller.

At five miles per hour, the autonomous brake was able to stop the vehicle in approximately three feet of travel distance.

### 5.3.3 Rear Wheel Encoders

Initial testing showed the onboard CAN speedometer was insufficient for velocity control feedback. The team developed quadrature wheel encoders to interface with the DBW system. A steel ring with 124 teeth is clamped to the brake drum, shown in Figure 1a. A ferromagnetic-sensitive quadrature encoder is secured to the rear of the wheel hub. The encoder ring is held in place by a double pin and bolt mechanism that applies radial force uniformly around the drum. This allows the mechanism to be easily serviced, replaced, or upgraded.

(a) Front of Encoder

(b) Back of Encoder

Figure 1: Wheel Encoder Assembly

## 5.4 Component Mounting

### 5.4.1 Cameras

For the object and lane detection, three Blackfly Flir cameras are being used. The mounts were designed to integrate into the overhead accessory rail behind the front windshield. A few of the design goals were to ensure repeatable orientation, optimize the FOV, keep the camera close to the windshield, and to not impact the rear-view mirror location. The design as seen in Figure 2 display all of those traits. To guarantee repeatable orientation, the corner mounts have overhangs that locate the track. There are overhangs on the front, back, and on one side to also ensure it is the same, left to right, on each side. The mount for the center camera integrates into the factory rear-view mirror mount. For more flexibility, there are multiple holes for the cameras to mount through. The reason for this is to get the camera as close to the windshield as possible to minimize windshield reflection/glare. The mounts were 3D printed out of ONYX because they needed a higher strength than the original PLA, but more importantly needed to withstand higher temperatures. Additionally, the ONYX finish meshes very well with the overall aesthetic of factory components on the car.

### 5.4.2 Camera Shrouds

The solution to the reflection issues was to enclose the camera FOV against the windshield. Eleven prototypes were made before a final design was decided upon. The shrouds are 3D printed out of black 95-A TPU. After the designs were near finalization, the printer settings were fine-tuned to get high quality print results. The results were highly effective in reducing the reflections, but the TPU material finish was slightly too glossy. To solve this, a few different spray finishes were tested. Black Plasti-Dip, flat spray paint, and flat spray enamel were used. The best results were found with the flat black spray enamel. The shrouds are independent of the cameras, and they are adhered to the windshield using a multi-purpose, clear silicone. The assembly can be seen in Figure 2 of how the mounts, cameras, and shrouds integrate together.

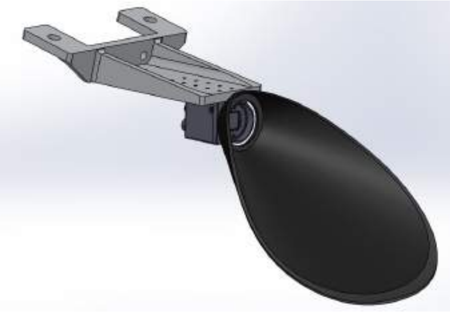


Figure 2: Rearview Camera Mount, Camera, and Shroud

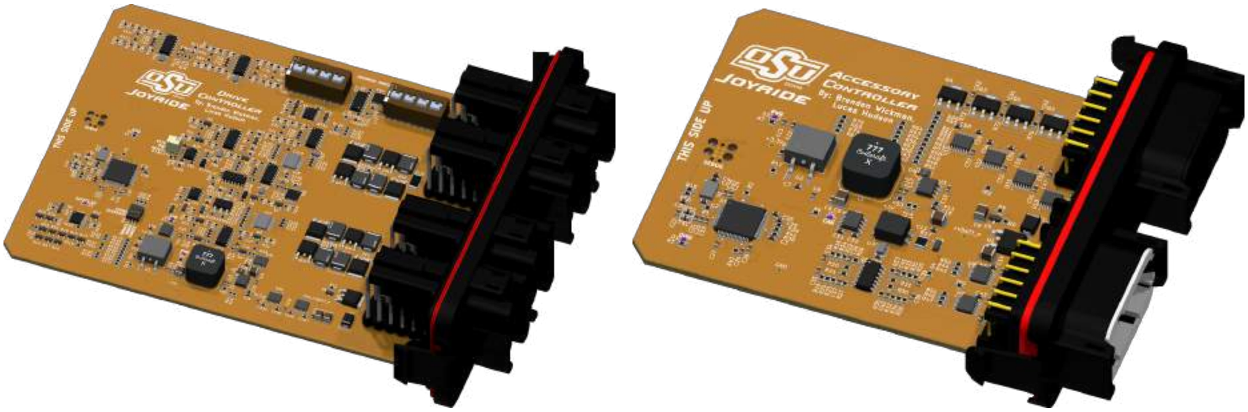
### 5.4.3 Miscellaneous

Additionally, the mechanical team fabricated 3D printed and metal mounts for the touchscreen display, front and rear 2D LiDAR, front 3D LiDAR, IMU, GPS antennas, beacon light, and door webbing.

## 6 Electrical Design

The vehicle has many electrical components within that are used not only to enable autonomous control of the vehicle but to create enhanced safety features and data collection. Work on custom PCBs started August of 2022 and the design for this semester will mark the third iteration of the current PCB architecture as seen in Figure 3.

The use of custom PCBs is an alternative to spending tens of thousands of dollars on an off-the-shelf designed and installed drive-by-wire system. Having a custom PCB not only ensures a greater educational outcome but an increased control of what types of data the custom drive-by-wire system has access to but also how the data is processed and passed along to other subsystems within the vehicle.



(a) Drive Controller

(b) Accessory Controller

Figure 3: Custom DBW PCBs

## 6.1 Drive Controller

The Drive Controller PCB is used to control functions that are directly related to the enabling of autonomy to the vehicle, as well as any sensor data that is considered important for safe vehicle operation. The Drive Controller is designed to be enclosed in a large Armor IPX enclosure paired with a 48 pin connectors as seen in Figure 3a.

### 6.1.1 Driver Feedback Capabilities

A ring lit LED push button is installed into the dash that is used to enter autonomous mode and, once the button has been pressed and the system has entered an autonomous state, the ring light around the outside of the button will illuminate green, indicating to the users of the vehicle that an autonomous state has been entered. There is also a RGB LED indicator installed into the dash that is used to represent the activity of the PCB and act as a quick, debug light that can present error codes in the form of light/color patterns.

### 6.1.2 Device Control Capabilities

The Drive Controller handles many types of signals and contains various hardware integrated logic sequences to ensure the system is always operating in a safe environment and has stable data being collected. The controller is capable of switching analog signal sources to turning on and off external relays.

**6.1.2.1 Signal Spoofing** In order to control the throttle and gear control signals an analog switch is used to switch between the factory signals and the signals generated by the Drive Controller. There are also physical switches on the PCB to allow for a manual signal override in case the PCB is ever to fail, the vehicle will still be able to be operated in manual mode.

**6.1.2.2 Braking Systems** The Drive Controller houses three braking systems. The first system is for the parking brake which consists of an H-Bridge to engage the brake. The remaining two systems are for rear braking, first being a system to generate a 0-12V analog signal that can control an electric-over-hydraulic trailer brake system, and second being an additional H-Bridge to control a linear actuator that can be used to control a master cylinder in the rear of the vehicle.

There will also be two brake pressure transducer inputs, one for the front and one for the rear brake system. These sensors will be used to ensure the systems are seeing an increase of pressure when the brakes are said to be applied.

**6.1.2.3 Wheel Speed Sensors** Each rear wheel on the vehicle will be equipped with a hall effect quadrature encoder that will be used to determine the vehicle's wheel speed and if it's moving in a forward or backward direction. Having wheel speed data is primarily for the velocity controller but also acts as another data point that can be used to ensure the vehicle is coming to a stop when the brakes are said to be applied. This data can also be used to determine if a wheel is locked up or skidding on pavement when the brakes are applied and the vehicle is not stationary.

**6.1.2.4 External Relays** The drive-by-wire system is powered from a relay that is turned on when the vehicles ignition is powered. In order for the onboard computer to properly shutdown, the computer must have a way of knowing if the vehicle is off without losing power.

To power the system after the vehicle is turned off an additional relay is placed in parallel with the existing ignition relay. The secondary relay is controlled by the Drive Controller and this gives the ability to the Drive Controller to keep itself powered once the vehicle is turned off without disturbing any factory power rails.

Now that the Drive Controller can keep itself powered, it can now also keep the relay powering the 48V inverter turned on resulting in the onboard computer staying powered after the vehicle is shutoff. The Drive Controller can now send a shutdown signal to the onboard computer and then properly shutdown the entire drive-by-wire system.

### 6.1.3 Future Proofed Design

The Drive Controller has various unused features built-in that can be used by future teams. These items are things such as front brake system pressure transducer input, and a second rear brake control system.

The front brake pressure transducer can be added to the vehicle to create a relationship between the front and rear braking system while the vehicle is in manual operation mode. This feature would be desired to improve the vehicles manual driving performance while coming to a stop since the system currently relies purely on front brakes. Creating a relationship between the two systems would allow for the Drive Controller to see the front brake systems pressure and match the rear brake system to that pressure to allow the user to have four wheel braking.

## 6.2 Accessory Controller

The Accessory Controller PCB is used to control functions that are not important to the vehicle's safe operation. These items mostly consist of lights and indicators or various other items which are required for the vehicle to operate in accordance with state and competition regulations. The Accessory Controller will be enclosed in a small Armor IPX enclosure paired with a 24 pin connector as seen in Figure 3b.

### 6.2.1 Driver Feedback Capabilities

A piezoelectric buzzer is installed underneath the dash and the Accessory Controller will activate the buzzer to indicate when the vehicle enters and exits autonomous mode. There is also a RGB LED indicator installed into the dash that is used to represent the activity of the PCB and act as a quick, debug light that can present error codes in the form of light/color patterns.

### 6.2.2 Device Control Capabilities

Various accessories that are required to stay in compliance with state and competition guidelines are listed below. All of the items are independently controlled by either a low-side or high-side self-protected output. A self-protected output is very important, this type of output means it can withstand many unwanted scenarios such as; over-voltage, over-current, overheating, and short to ground/12V. The accessory controller manages headlights, blinkers, brake lights, the horn, and the windshield wiper.

### 6.2.3 Future Proofed Design

The Accessory Controller has unallocated pins that are designed to be used in the future if the need arises. The design proof is delivered through making the unallocated pins a mixture of input and outputs that can later be connected to any type of device or sensor.

Extra pins that are designated as inputs will have the ability to read an analog or digital input from 0V to 12V with a max frequency of 12kHz. Outputs will consist of self-protected high and low-side outputs that can be used to power a 12V device or control a relay coil.

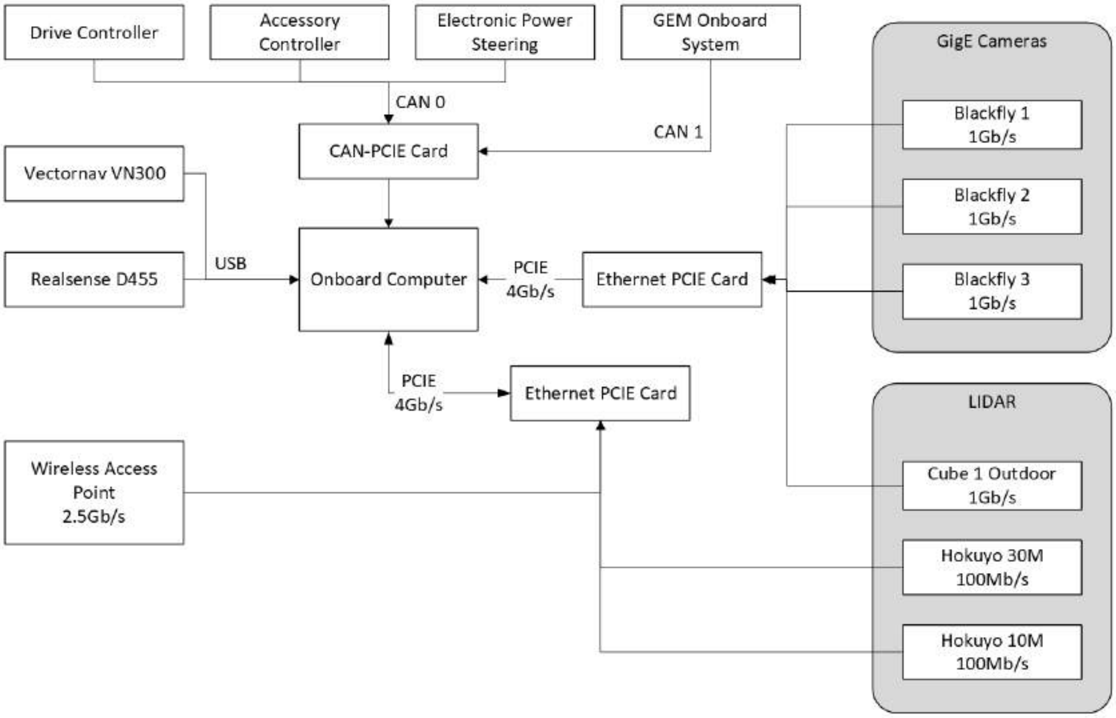


Figure 4: Hardware Overview

## 7 Software Design

The software stack is oriented around the Robot Operating System (ROS). More specifically, ROS2 Humble on Ubuntu 22.04 is being used. This was chosen due to its relative maturity compared to other ROS2 versions, but newer navigation features compared to ROS1.

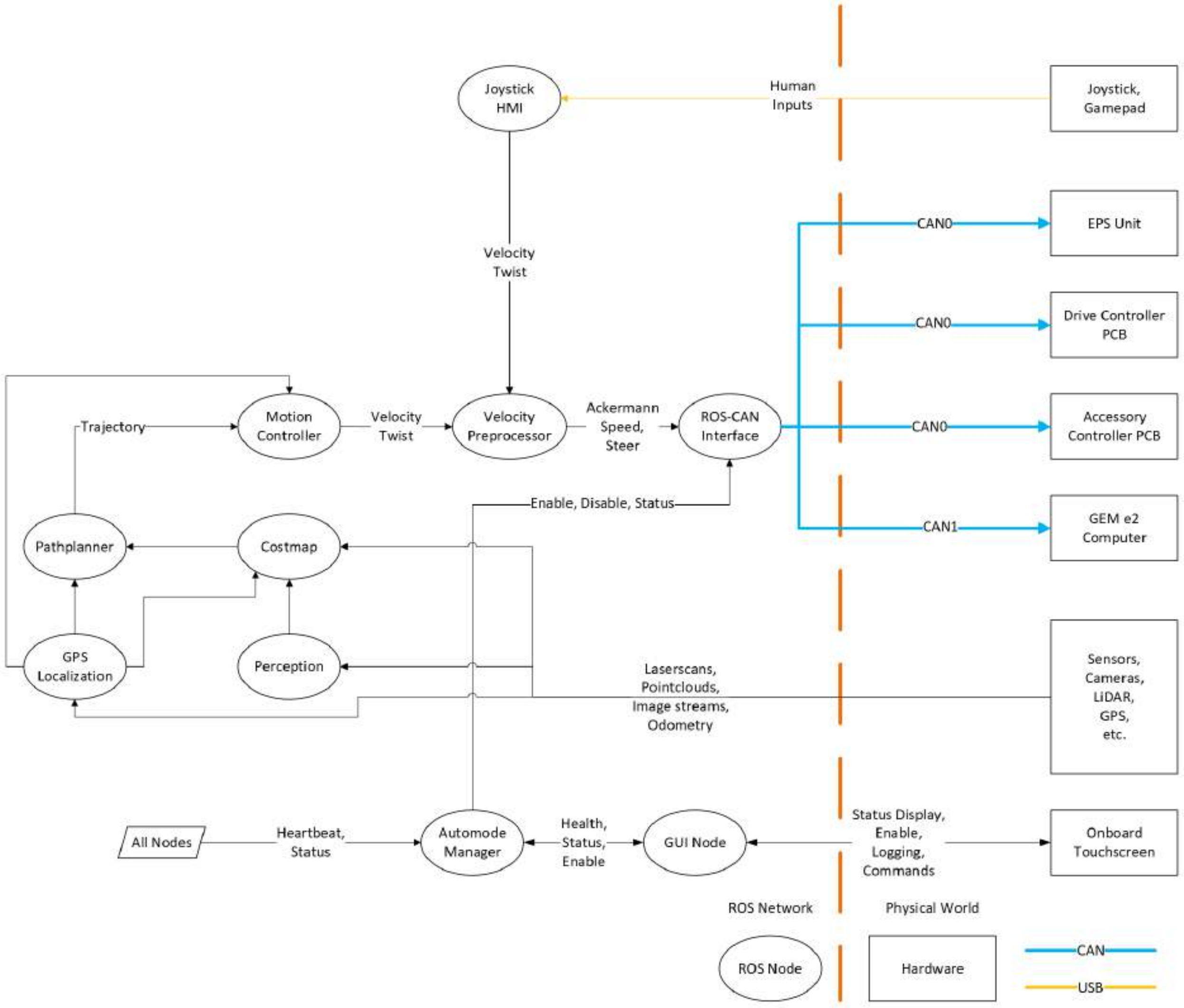


Figure 5: High-level software overview

## 7.1 Lane Detection

Per IGVC rules, lane detection is required to utilize traditional computer vision which encourages detection primarily based on shape and color. Furthermore, the lanes in the competition are comprised of a one-way, two-lane road that will be marked in three-inch white tape on asphalt. The lanes will be eight feet apart and contain a curve with a minimum twenty-foot radius. Taking these specifications into account the lanes can be detected by their light color on top of the dark asphalt.

Implementation of this involves getting the individual frames from the video stream and converting the color space of the frame from RGB, with axes representing red, green, and blue, to that of Lab, with axes representing lightness, red/green, and blue/yellow. The lightness channel is then extracted and a threshold is applied so that only pixels that are brighter than the threshold are collected. This set of light points is sent through Opencv's edge detector and then through a Hough Transform. The Hough Transform takes the edges and runs a line of best fit between the points collected by the edge detection. The lines with the highest probability of making up lines from the given edges are returned as a set of line segments. These segments are combined to form distinct lane lines along which samples are going to be extracted and converted into

3D space for use in navigation.

## 7.2 Obstacle Detection

The vehicle's obstacle detection utilizes traditional computer vision methods to extract the shape and location of an obstacle. The overall design for obstacle detection uses OpenCV's library to perform edge detection and use a support vector machine to determine an object within the view of the vehicle. Obstacle detection is separated into two categories: object detection and pedestrian detection. Both programs will extract a detected target frame-by-frame and display it on a blank frame for the onboard GUI to display. Afterward, the position of the detected object(s) will be transformed from 2D space into 3D space for obstacle avoidance.

### 7.2.1 Pedestrian Detection

For pedestrian detection, the vehicle uses a Histogram of Oriented Gradients algorithm with a support vector machine (SVM) to identify human-like objects for each frame from the camera. The HOG algorithm is a feature descriptor that computes the distribution of oriented gradients within a frame. A histogram is created for the gradient orientations where a support vector machine determines whether or not the cell(s) in the image contains a human-like object. OpenCV provides two different HOG feature descriptors for the HOG-SVM implementation: the default people detector and the Daimler people detector. The dataset that is used to train the default people detector is not disclosed in the documentation by OpenCV, but it is most likely trained from the INRIA Person Dataset, Caltech Pedestrian Detection benchmark, or KITTI Vision Benchmark Suite. The dataset used to train the Daimler people detector is the Daimler Pedestrian Detection Benchmark dataset. The Daimler people detector was chosen since its dataset contains pedestrian images from video recordings in real-world traffic scenarios which means that the images contain wider ranges of lighting conditions, scales, occlusions, and viewpoints. This allows for the pedestrian detection to operate under less constrained environments.

Unlike a more modern approach such as a neural network, this algorithm works well for a traditional computer vision algorithm. However, the limitations are apparent in the forms of significantly different classifications depending on lighting and targets having extraordinarily complex shapes to be classified. This leads to many false positives and false negatives classifications in some instances. Thus, our implementation has a parameter for manually fine-tuning the detector in different environments. The ROS node connecting the HOG-SVM algorithm will provide the pixel position of the object's center to the ROS network.

Figure 6: Pedestrian Detection Output

### 7.2.2 Object Detection (Pothole and Tire Detection)

For pothole and tire detection, a more simplistic approach was taken using edge detection algorithms to distinguish shapes within a frame. Using OpenCV's Canny edge detection function, the edges within a frame can be extracted onto a blank canvas. Furthermore, a region of interest was applied that will remove all detected edges outside this region to only allow detection in front of the vehicle.

The program will determine an object by area size within a complete shape from the Canny edge detector. The minimum and maximum area thresholds are parameters that can be adjusted for different environments or objects. This approach has many limitations due to the traditional computer vision approach where the road conditions will heavily affect the detection of the object if different contours intercept each other. Lighting is another limitation since low-light environments mesh edges together, making it more difficult to distinguish different objects. Under normal conditions, the edge detection approach works well in order to convert that object location from 2D space into 3D space for object avoidance. The ROS node connecting this program will be provided with the object's center location.

### 7.2.3 Sign Detection

The sign detection in the vehicle utilizes a convolutional neural network known as YOLO to classify road signs for proper vehicle avoidance and procedures. The pre-built neural network was trained using the YOLOv5s architecture with a large dataset of road signs within the world. This model quickly and accurately identifies road signs to swiftly place into a cost map for the vehicle to proceed accordingly. As with the previous detection algorithms, the ROS node connecting to the neural network will provide the pixel position of the object's center, classification type, and confidence level.

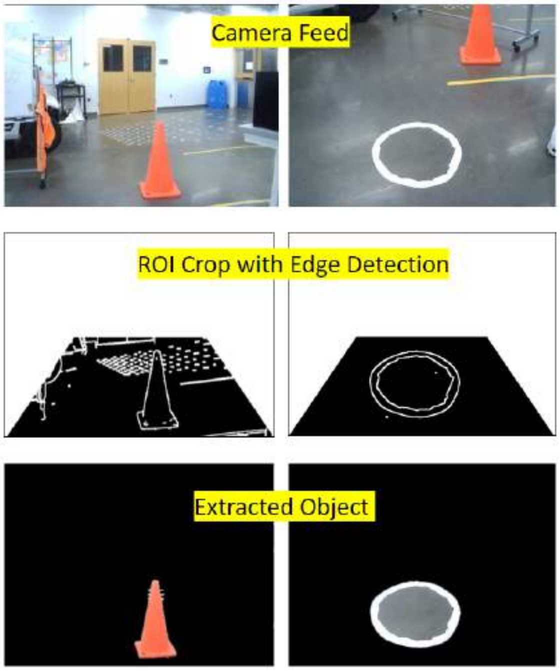


Figure 7: Obstacle Detection Output

## 7.3 Navigation

The team is utilizing ROS 2's Navigation 2 (NAV2) software stack to handle global planning and motion control. NAV2 is organized around behavior trees (BTs) that make requests of servers on the ROS network. For example, the *NavigateToPose* BT may request a new global path from the current position to a goal pose. This request is sent to the *planner* server, which computes and returns the requested path. BTs can be created and combined in whatever manner necessary to achieve desired behaviors.

### 7.3.1 State Estimation

The vehicle's state is provided by the VN300 inertial navigation system (INS) installed onboard. It provides latitude, longitude, and orientation estimates at 400 Hz, which are sampled at approximately 75 Hz. This data is converted into a world-fixed frame using the local-tangent-plane (LTP) approach. The LTP origin is set at the vehicle's initial position. A ROS node *NavSatOdom* performs this transformation constantly and publishes a live odometry transform to vehicle local frame. The *NavSatTransform* node allows any ROS node to convert between latitude and longitude and the LTP coordinates. Traditional ROS systems have both a *map* and *odom* frame according to ROS REP 105. To ensure compatibility, the system publishes a static transformation between both *map* and *odom* which treats them as identical frames.

### 7.3.2 Planning and Motion Control

Joyride utilizes a Hybrid-A* path planner implemented in the *SmacPlannerHybrid* ROS package. It generates a path between the current pose and the goal pose every second.

Once a path is generated, the vehicle must generate desired local velocity twists (linear $X$ 0-and angular $Z$) to follow the path. The twists are generated by the motion controller, which can implement any number of algorithms. The vehicle currently uses a variation of the Pure Pursuit algorithm, referred to as *Regulated Pure Pursuit*, provided by the *NAV2* package. Desired velocities are referenced to the center of the vehicle's rear axle. It is transformed into a wheel speed and wheel angle conforming to an Ackermann steering geometry before being transmitted to the DBW system.

### 7.3.3 Environmental Representation and Mapping

The environment will be represented as a two-dimensional costmap, often referred to as an occupancy grid in ROS nomenclature. *Lethal* obstacles are those that the vehicle can collide with, such as barrels and pedestrians. *Nonlethal* obstacles should be avoided to optimize performance, but are not safety-critical; lane markers and potholes are examples of this type. Lethal obstacles will be inflated to a much higher cost than nonlethal obstacles so the vehicle crosses a lane rather than strikes an object.

The map will be generated using a Simultaneous Localization and Mapping (SLAM) approach. The localization component will be provided by the INS, meaning SLAM is concerned only with building the map during operation. Obstacle information will come from two sources: cameras and LiDAR. LiDAR returns will be directly fed into SLAM. Camera frames will be interpreted for various obstacles and inserted into SLAM as pseudo-LiDAR.

## 7.4 Low-Level Control

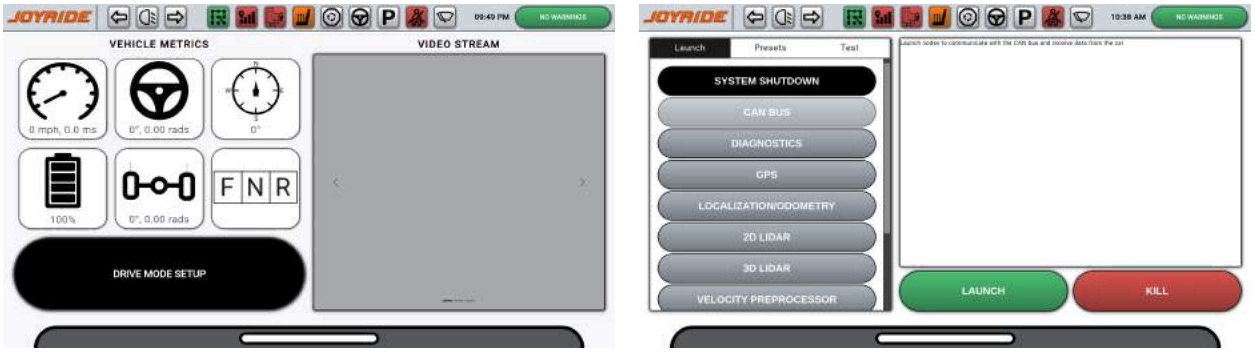
Vehicle motion is controlled by the Drive Controller PCB. It controls the throttle, brake, and gear according to measured and desired forward velocities. Wheel velocity is measured using the rear wheel encoders. The desired velocity is received over the CAN bus from the ROS network. By comparing the two, the controller selects the necessary gear and toggles between a throttle PID and a brake PID control loop. The controller ensures the vehicle stops before changing gear smoothly with actuators to reach the desired velocity.

### 7.4.1 Communication

ROS and the DBW system communicate over two CAN buses. One interfaces ROS to the custom PCBs and the power steering. The other connects ROS to the existing GEM CAN bus. A ROS node performs translation between CAN and ROS topics, allowing for data to be properly transmitted back and forth. A 2-port PCI-E CAN interface provides a direct CAN connection to the onboard computer. Sensors and other devices are connected on a local area network managed by the onboard computer.

## 7.5 Graphical User Interface

The GUI is made up of five pages: The dashboard, commands, sensor data, system health, and waypoints page. Each page has a status bar at the top that contains icons to indicate the status of the vehicle and its various systems. The dashboard is the main page of the GUI and contains immediately useful feedback about the vehicle. Each of the vehicle metrics has an associated animated icon that gives the user information at a glance. Additionally, the dashboard has a video stream section where the user can switch what video stream is currently being displayed. Finally, the dashboard has a quick setup button to ready the system for autonomous or joystick control. The commands page purpose is to streamline testing and preparing the vehicle. With the commands page, the user can launch individual nodes, presets, or unit tests at the press of a button. The sensor data page allows for easy visualization of incoming data. The page contains three different graphs that support overlaying plots for comparing data by simply dragging and dropping a data series' associated button atop the graph. The system health page makes keeping track of the ROS network and troubleshooting any issues that may arise much more manageable. Each ROS node sends its status to the system health page where it is color-coded and automatically sorted into categories in real-time. Moreover, there is a tab that monitors messages sent on ROS's logging topic and displays them with a timestamp and an associated message that is also color-coded. The final page of the GUI, the waypoints page, is how waypoints are set for waypoint navigation. A familiar setup lets the user set waypoints for the vehicle to travel to either by clicking on a map or by manually entering the coordinates. The map also shows the path the vehicle has traveled and the planned paths produced by the path planner.



Launch nodes to communicate with the CAN bus and receive data from the car

(a) Dashboard (b) Commands

| NAME | MESSAGE |
|---|---|
| left motor | OK |
| right motor | Warning |

| ID | NAME | MESSAGE |
|---|---|---|
| | left motor | OK |
| | right motor | Warning |

(a) Sensor Data (b) System Health

Figure 9: GUI Pages

Figure 10: Waypoints

## 8 Failure Modes and Safety

Vehicle state and diagnostics are constantly monitored to ensure safety. Malfunctions or human intervention will trigger either an emergency stop, return to manual control, or both. The E-stops (onboard and wireless) control a hardware signal that actuates the autonomous brake.

Vehicle status is shown on the touchscreen GUI at all times. DBW status is also displayed using two RGB LEDs mounted on the dash, which change color and flash to assist in diagnostics. A ROS node and the DBW system manage autonomous mode together. Any error or loss of heartbeat on either side will trigger an exit of autonomy. Diagnostics from each node are displayed on the GUI to visualize in real-time which nodes have errors. ROS2 lifecycle nodes are utilized to better manage startup and shutdown sequences.

Critical safety systems were tested outdoors under realistic scenarios. All E-stops are proven to be functional and to stop the vehicle. Wireless E-stop range is verified. Manual steering control can be resumed by grabbing the wheel. Manual brakes will overpower the throttle to stop the vehicle. At five miles per hour, the vehicle can stop in approximately three feet after E-stop is pressed.

| Failure | Likelihood | Resolution |
|---|---|---|
| Loss of Power | Low | Use manual brakes to stop vehicle. |
| DBW Malfunction | Low | Alert driver with GUI. Resume manual control. Override throttle with manual brake or E-stop if necessary. |
| Computer Malfunction | Low | Resume manual control. |
| Network Malfunction | Low | Alert driver with GUI. Resume manual control. |
| Sensor Malfunction | Low | Alert driver with GUI. Resume manual control. |
| ROS Node Failure | Medium | Automatically restart non-critical nodes. Alert driver and resume manual control if critical node. |
| Failure to detect lanes | Medium | Continue moving. Halt if uncertain for too long. |

## 9 Simulations

A graphical simulator was *not* used in the development of the vehicle due to manpower constraints. It is left as a task for future Joyride teams. Instead, the team utilized hardware-in-the-loop (HITL) testing through a combination of replaying live data (in the form of a ROSBAG) and spoofing data through non-graphical simulation. This resulted in significant physical testing being necessary to validate new algorithms. The existing ROS node *RVIZ2* is used to visualize performance during HITL tests.

## 10 Performance Testing

Throughout the design process we employed subsystem unit tests, integration tests, and outdoor testing to efficiently build and test the car as a whole. All integration and outdoor tests were performed with working E-stops, two people in the vehicle at all times, and in the presence of an observer holding the wireless E-stop. Unit tests of each subsystem were performed in an indoor lab space to assess their performance. Integration testing was performed indoors with the cart being operated on jacks. Outdoor tests were performed in a university-owned parking lot which offered more realistic conditions to test the cart.

Due to university rules, the team was unable to create a proper test course for the vehicle to validate lane following, intersections, and other scenarios requiring lane markings. Safety-critical systems were consistently re-evaluated during each outdoor test. Any failures were documented and discussed for later revision or redesign.

## 11 Initial Performance Assessment

As of May 15, 2023 the vehicle is capable of remote control with forward, reverse, and brake control functionality. GPS waypoint navigation is partially functional, but not reliable. Object and pedestrian detection are untested on the moving vehicle. Lane following is not implemented as lanes are not yet mapped into world space and the local cost map. Localization is functional and reliable.

Electrical and mechanical work is complete and all components are mounted. Communication between the DBW and ROS is functional and reliable.