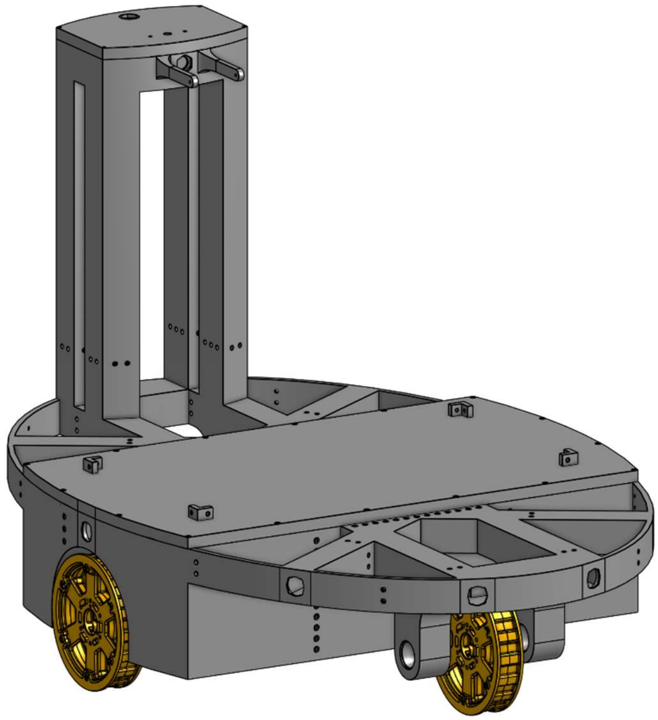
The Mobile Robotics Research Team
at
MILLERSVILLE UNIVERSITY

N.A.G.L.F.A.R.E.



Submitted May 15, 2022

Team Captain

Ian Troop | IPTroop@millersvill.edu

Team Members

Benjamin Weaver | BAWeaver@Millersville.edu
Dennis Nguyen | DENguyen@Millersville.edu
Hecmarys Cintron | BAWeaver@Millersville.edu
Jacob Garcia | JGGarci1@Millersville.edu
Joseph LaMontagne | JOLaMont@Millersville.edu
Matthew Way | MAWay1@Millersville.edu
Sofia Griffiths | KJGriffi@Millersville.edu
Zane Weaver | ZAWeaver@Millersville.edu

Faculty and Volunteer Advisors

John Wright Jr., Ph.D., CSTM, CLSSGB, CSCE, F.ATMAE | John.Wright@millersville.edu
John Haughery, Ph.D., CSCE | John.Haughery@millersville.edu
Jeff Pinegar | JPinegar@phoenixcontact.com
Ermias Wogari | ewogari@precisioncobotics.com

I, Dr. John Wright Jr., certify that the design and engineering of N.A.G.L.F.A.R.E. by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Signature: ______________________ Date: May 15, 2025

TABLE OF CONTENTS

1. DESIGN PROCESS AND TEAM ORGANIZATION

1.1. Introduction

The Mobile Robotics Research Team at Millersville University has been developing competitive mobile robots since 2001. During that time, the team has designed, built, and programmed 50 robots, winning over 50 awards in numerous national and international competitions, including seven national championships. In 2018, the team began developing vehicles for the Intelligent Ground Vehicle Competition, including this year's entry, which competed with seven different robots featuring several unique approaches, including historic core concepts such as distributed intelligence, ROS minimum viable product, Industrial hardware, and, for this platform, lowest viable cost with the objective of a competitive platform with a cost in the three figures instead of four or five. To achieve this objective, the platform was named N.A.G.L.F.A.R.E. (New Autonomous GPS and Line Following Advanced Robotics Experiment), named for the ship in Norse mythology that carries Hel's monsters to Asgard at Ragnarök, made from people's waste.

1.2. Team Organization

The Mobile Robotics Research Team is officially organized as a club at Millersville University. This enables us to receive substantial funding from the university and provides us with access to their labs. Clubs are organized with an elected officer cabinet responsible for managing club operations and administrative tasks. In addition, the advisors select a project lead for each platform in conjunction with the cabinet. The Project Lead maintains a unified vision for the project and is responsible for its successful completion. For R&D work, we have adopted an integrated team organization, facilitated by the interdisciplinary nature of our Automation and Robotics major, which enables all team members to possess the skills necessary to participate in every aspect of the platform.

Name	**Email**	**Position**	**Major**	**Class**	**Hours**
Ian Troop	IPTroop@Millersville.edu	Lead, Treasurer	AURO	2025	1000+
Dennis Nguyen	DENguyen@Millersville.edu		AURO, CSCI	2025	50+
Joseph LaMontagne	JOLaMont@Millersville.edu	Vice President	AURO	2026	20+
Matthew Way	MAWay1@Millersville.edu	Secretary	AURO	2026	20+
Jacob Garcia	JGGarci1@Millersville.edu		AURO	2025	20+
Zane Weaver	ZAWeaver@Millersville.edu	President	AURO, CSCI	2026	10+
Sofia Griffiths	KJGriffi@Millersville.edu	PR Chair	AURO	2026	10+
Benjamin Weaver	BAWeaver@Millersville.edu		AURO	2027	10+
Hecmarys Cintron	HECintro@Millersville.edu		AURO	2029	10+

AURO: Automation & Robotics Engineering Technology, Department of Applied Engineering, Safety & Technology
CSCI: BS, Computer Science, Department of Computer Science

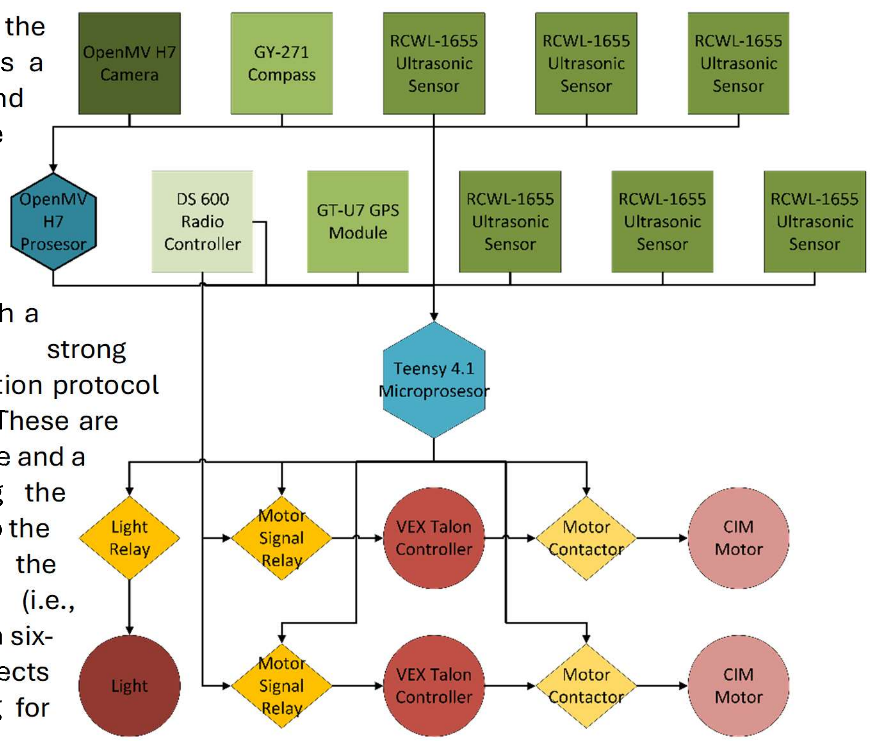
1.3. Design Assumptions and Process

Our designs always start on the competition field, reflecting on the past platforms, both ours and others. We have seen a consistent trend of robots costing thousands of dollars. When reflecting on the platforms we have developed over the years, there has been a persistent and increasing trend toward increasingly industrial and expensive platforms, culminating in a plan to build an entirely industrial vehicle. One extreme inspired us also to try the other: building a robot with a cost under $1,000. This design assumption implied others. First, the platform would use a low-cost microcontroller. Second, it would be accompanied by as few or cheap sensors as feasible.

Once we have established the restrictions of the competition and any added restrictions not a apparent in the rules our design process starts from the perceived most restrictive point for example here we started with motors we believed the cost to speed to torque relationship would be the most restrictive and that therefore if anything were decided or designed before obtaining a preliminary choice would almost certainly have to be redesigned. Next, we check our assumption by examining how the element fits in the design. If it offers any known restrictions that change previous assumptions, we chose low-cost brushless motors in our example since they provided the best balance on paper. When considering how this choice might add restrictions, it is apparent that motor controllers are now restricted to ones compatible with brushless motors, so that was what we investigated and calculated the needed specifications. This is where the recursive element of the process can be seen, where we repeat selecting the most restrictive aspect until the vehicle is complete, but it is missing a feedback element. Once we investigated brushless motor controls, it became clear they were prohibitively expensive. We needed to look back at our decision of the previous most restrictive element, where we were incorrect based on what we have learned. It is clear that motor controllers cost twice as much as motors and that motors and motor controllers will restrict each other, so motor controllers must come first or be considered one element in pairs. While limited to the first stage, this example demonstrates the process in a very structured way, helpful in explaining, but not in how the process flows as the project comes together. As more is designed, the recursion and feedback flows naturally from one to the next as relationships and interactions become established and clear.

2. SYSTEM ARCHITECTURE

At the center of our architecture is the Teensy 4.1 microcontroller giving us a good balance between cost and capability for its data collection we utilize six RCWL-1655 ultrasonic sensors giving it a rudimentary local map for pennies on the dollar along with a OpenMV H7 camera which allows us to run a vision system with a dedicated microprocess with strong microcontroller friendly communication protocol without needing a whole computer. These are accompanied by a GT-U7 GPS module and a GY-271 magnetometer, completing the sensor suite. All of these route data to the Teensy through either one of the receive/transmit two-pin protocols (i.e., GPIO, UART, I2C), or for the camera, a six-pin communication protocol that reflects the six ultrasonic sensors, allowing for



natural correlation between them. For Power, the Robot has a 12VDC LiFePO4 that runs directly through the E-Stop, allowing complete power disconnect. The 12V DC power source runs the motors, and lights and is stepped down to 5V DC for the sensors and processors. The 5V DC is distributed via a fused distribution block. A DS 600 radio controller is used for the remote E-Stop, allowing us to disconnect motor power and switch relays, enabling direct remote control or autonomous navigation with a single button push. The software is split between the OpenMV H7 which is detecting lines and determining which of six zones they fall in and the central controller Teensy microcontroller that takes the signal from the camera and combines it with the rest of the sensor suite to create a custom programed local map that allows the vehicle to make moment by moment decisions based on its environment.

3. INNOVATIONS IN VEHICLE DESIGN

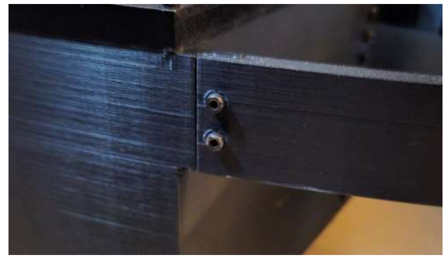
3.1. Cost

For our money, the feature that makes N.A.G.L.F.A.R.E. stand out from the pack the most is its price tag. Many teams work within tight budget constraints and are hard pressed to find another vehicle with a three-digit price tag that looks this clean. While keeping the cost under $1000 proved difficult, it presented an interesting challenge that led to a unique solution that might not have been attempted otherwise.

Item	Price	Unit	Qty	Total
PETG 3D Printer Filament	$12.99	Kg	15	$194.85
Motor and Gearbox	$66.42	Each	2	$132.84
Motor Controller	$49.99	Each	2	$99.98
OpenMV Cam H7 Plus	$85.00	Each	1	$85.00
Dakota LiFe 12V Battery	$49.50	Each	1	$49.50
Fused Power Distribution	$38.00	Each	1	$38.00
Hex Aluminum Hubs 1/2”	$4.49	Each	8	$35.92
6CH 2.4GHz Controller	$30.99	Each	1	$30.99
6" Omni-Wheels	$14.99	Each	2	$29.98
Lockable Teensy 4.1	$29.60	Each	1	$29.60
Ultrasonic Sensor	$4.40	Each	6	$26.39
6" Traction Wheels	$12.48	Each	2	$24.96
Teensy 4.1 Mount	$23.90	Each	1	$23.90
Flanged Bearing 1/2" Hex	$1.50	Each	8	$11.99
Clamping Shaft Collar	$0.99	Each	12	$11.88
20 AWG Wire	$0.01	In	1000	$11.65
Threaded Heat Inserts	$0.02	Each	500	$9.99
Din Rail Mount Adapter	$0.56	Each	16	$8.96
MakerFocus GT-U7 GPS	$8.00	Each	1	$8.00
Industrial Warning Light	$7.99	Each	1	$7.99
12V High Amp Contactor	$3.20	Each	2	$6.40
DIN Rail 35mm 16"	$3.10	Each	2	$6.20
8 Conductor Cable	$0.10	In	50	$5.00
5VDC Relay	$1.20	Each	4	$4.80
20 AWG Wire Pair	$0.02	In	300	$4.75
DC to DC Converter	$4.50	Each	1	$4.50
Screw Terminals	$0.27	Each	16	$4.30
E-Stop Push Button	$3.75	Each	1	$3.75
Metric Bolts	$0.01	Each	500	$3.61
Heat Shrink Tubing Kit 4:1	$0.02	Each	100	$2.35
Hex Stock 1/2"	$0.28	In	8	$2.22
DIN Rail Terminal Block	$0.22	Each	10	$2.20
12 AWG Wire Pair	$0.02	In	100	$2.17
GY-271 Compass Sensor	$1.75	Each	1	$1.75
Wire Ferrules Terminals	$0.01	Each	100	$1.00
Waterproof Cable Gland	$0.26	Each	2	$0.52

Total: $927.89

3.2. 3D Printed Chasie

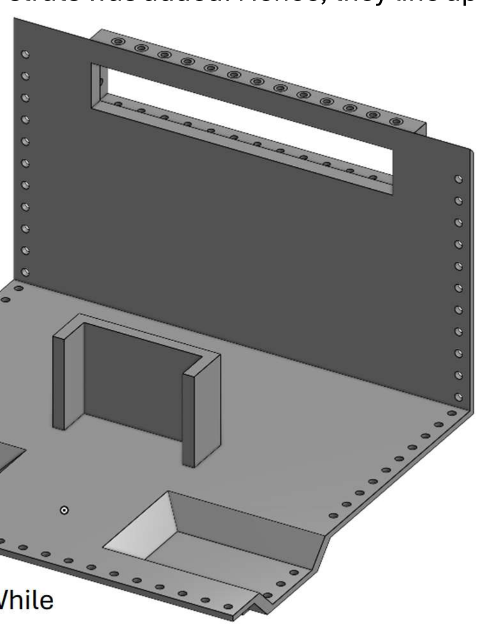


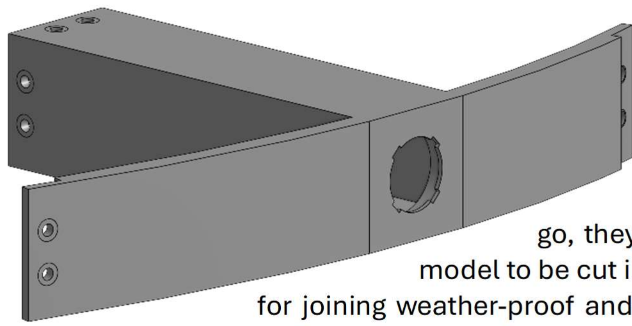
3D printing the chassis probably wouldn't have been a consideration without the need for a low-cost, lightweight chassis that could mount waterproof ultrasonic sensors while maintaining a clean look and a high degree of customizability. The 3D-printed PETG platform allows for a flowing look with integrated sensor mounting, showcasing what can be achieved with a modern 3D printer.

4. MECHANICAL DESIGN

The vast majority of the vehicle is 3D printed. The electronics and power discussed in their section are the few components purchased instead of being printed. The threaded heat set inserts are small brass threaded parts that have become commonplace in the 3D printing community. Simple cap bolts 3mm for all the frame connections (motor bolts), 2mm for some electronic mounting, and 5mm for the DIN rail. 35mm DIN rail consisting of two long sections in the main sections of the frame and one short section for the signal tower. Some electronics do not generally mount to DIN rails, so 3D printed adapter plates are mounted to universal DIN rail mounts for these components, allowing for easy interchangeable mounting. The vehicle utilizes wheels standard in FRC, driving on a pair of VEX six-inch traction wheels, each configured to be an inch wide, along with supporting six-inch VEX omni wheels on the front and back, with half-inch hex shafts running through hex flange bearings with shaft collars. While motors are powerful components, they are also mechanical. To meet the tight budget a pair of Vex CIM motors with 16:1Versa Planetary gear box giving a max speed just a hair bellow five miles per hour along with a modest amount of torque that would only be limited by the wheels as long as the robots weight does not exceed 60 pounds and would be more then sufficient for considerably more.

The mechanical design is where the impact of the low-cost solution is most evident. This begins with the motors. The CIM motors we use were selected for their low cost while meeting the minimum specifications, but they left little room for excess weight. Combined with the course challenges and budget restrictions, this limited our options, particularly when flexibility is desired. To achieve this, the entire frame was to be 3D printed. The frame design originated from a two-foot-wide, three-foot-long ellipse chosen to maximize maneuverability. From there, a central box was extruded towards the ground to maintain a very low stable center of gravity by placing the battery, payload, and most components near the center, extending towards the box from the ellipse, a network of struts was added. Hence, they line up with the position of the ultrasonic sensors, E-Stop, and signal tower to stiffen the frame, give wire routes, and support the omni wheels. These front and back sections do not extend towards the ground maintaining a clearance to climb over moderate obstacles when this is combined with the center drive wheels being slightly lower the omni wheels on either end and the intrinsic flex of a PETG frame gives the vehicle a suspension system capable of handling far more then what the course will throw at it. The low frame means that there is no where high enough to place the E-Stop and camera along with placing the compass and wireless devices close to potential noise from the motors to alleviate this a two-foot-nine-inch tall tower was added with a box at the top for the compass, GPS, safety light, remote control and E-Stop receiver, and the physical E-Stop, with a pair of arms extending from the front to mount the camera in a way that would allow for adjustable tilt. While





this tilts the vehicle back, it is not enough to introduce instability; instead, it increases the ability to climb over obstacles in the direction of travel. There was only one viable option to bring this vehicle from the screen to the real world. While a handful of 3D printers are large enough to print the frame in one go, they are both expensive and unreliable. This only left the model to be cut into blocks that fit on a more reliable printer. A method for joining weather-proof and strong sections had to be designed to achieve this. Initially, a solvent weld similar to PVC cement was considered; however, this would have made repairs or later minor edits almost impossible. Therefore, overlapping nesting connection points with a snug fit were held together with threaded inserts to pin the joints together. A half-inch overlap ensures the frame remains strong, allowing for comfortable use in rainy conditions and enabling the vehicle to function in a broader range of conditions than the average driver on the road seems capable of.

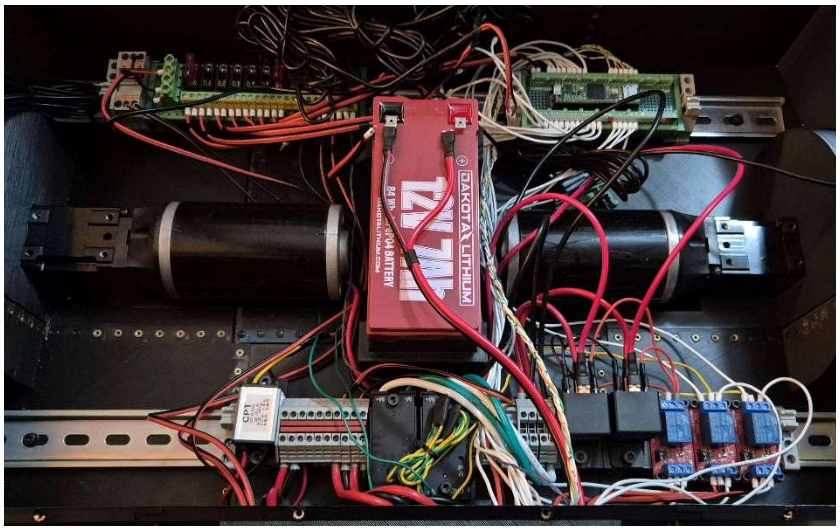
5. ELECTRONIC AND POWER DESIGN

The key goal with the electrical and power system was to use an industrial style of design while maintaining the tight budget. These two objectives are at odds, so compromises had to be made. For mounting components, we wanted to use DIN rail, as mentioned in the mechanical section, to have an industrial look and the adaptability benefits it offers. To do this, many components or breakout boards that could be affordably purchased in a DIN rail-mounted variant were acquired; failing that, universal DIN rail mounts with 3D printed adapter plates were used. In a similar vein we intend to use screw or push in terminals everywhere possible both to fit with the industrial styling and how secure the connections are. This was achieved by acquiring components or breakout boards that natively have screw terminals or by removing Dupont pins and replacing them with screw terminals of the same pitch. Between the DIN rail adapters and the loose screw terminals the restrictions introduced by the desire to have more industrial electronics are alleviated while keeping many of the benefits allowing for a more straightforward design process.

Since the vehicle is, at least electronically, assembled from preexisting components, much of the design process involves investing specifications of available components to assemble a selection capable of all required features within the given imperceptible restrictions. During our design process, several instances of the pattern were described to explain our process. To better detail our choices and their underlying logic, these decisions and reevaluations will be discussed as they are relevant. As detailed in the description of our design process, the first elements decided were the motors and motor controllers. Starting with brushless motors and determining that the controllers made them cost-ineffective, brushed CIM motors were selected based on the selection of the VEX Talon motor controllers. Both of these have been proven reliable by rigorous use by FRC teams since this pairing is among the most popular. With the motor voltage and available torque established, the next component to be determined was the battery. We needed a low weight. Hence, lithium of some kind was high in contention, and the reliability and value of Dakota LiFePO4 in one of their smallest sizes, with 7Ah at 12VDC. At the same time, this is small compared to their offerings with hundreds of amp hours. A full charge is still sufficient for a full day of testing and only takes 45 mins to charge while staying light at only 2.2lb and only costing $50. This is the point at which most of the fundamental mechanical aspects of the platform were chosen before moving back to electronic components.

The next electronic component that needed to be determined was the sensor suite. As mentioned in the design process section, sensor choices were limited to either one sensor per task or an exceptionally cheap sensor, and, where possible, both. The three primary sensors needs are object detection, marker detection, and GPS. There were two primary options for object detection: a LiDAR as a single sensor with a higher price or a bank of cheaper single-point sensors. A LiDAR capable of operating in outdoor conditions would cost at least a few hundred, taking up a significant proportion of the budget, leaving signal point range finders as the only viable option given our constraints. Within single-point sensors, there are two primary sensor types: light and sound, usually infrared and ultrasonic. Infrared sensors can be far more precise and accurate, but cheaper models are easily washed out in direct sunlight. The cost of high enough end sensors quickly approaches the price of a LiDAR really only leaving ultrasonic sensors while from a resolution and accuracy perspective they are somewhat limiting we felt they would still offer enough data to be viable and a bank of six RCWL-1655 ultrasonic sensors costing under $30 meant they were the only path to a sub $999 vehicle.

In a similar but far tighter paradigm, there existed options for sensors to detect pavement markings. Here, the expensive choice would be some kind of microcontroller with a camera, and the distributed, cheaper choice would be infrared line sensors. The two considerations here resulted in the opposite decision of object detection. The cost gap was between $20 and $80, making both financially viable. At the same time, the capabilities swung heavily towards the camera option since the infrared sensors would require cumbersome hoods to keep from washing out in sunlight. Three options were considered when selecting the camera. A Teensy/Arduino compatible camera sensor, a web camera with a Raspberry Pi with OpenCV, or an OpenMV camera. All three were investigated to see if they could drive the vehicle and the camera processing. In the end, the I/O requirements of other sensors meant that only a teensy could drive everything, and no viable sensor could be found for one, so the reliable standalone OpenMV H7 camera was selected for vision. With a Teensy 4.1, a low-cost microprocessor with 1024 Kbytes of RAM, a 600 MHz processor, and 42 I/O pins, as the central processor.



The last input required for navigation is GPS. At the same time, there are many options. Generally speaking, relatively few fit the budgetary requirements. To this, a GT-U7, a small GPS module designed for compatibility with Arduino, was quickly decided on and combined with a GY-271 magnetometer. This allows the vehicle to know where it is and its compass bearing. All this costs less than $10, giving it a value offer that is hard to beat.

The next concern was the safety system and, by association, the beginning of the power system. For the physical E-Stop, a standard mushroom button twist release with a pair of normally closed contacts, a purpose-designed E-Stop button. This button is placed between the battery and the system on both the positive and negative legs. This safely breaks the circuit and leaves less room for short circuits than breaking only one leg. A DS 600 6CH 2.4GHz radio controller was employed for the remote E-Stop. The radio controller allows the vehicle to be manually driven for convenience rather than carried or pushed. This and the E-Stop functionality are achieved by feeding the four non-driving channels off the controller into the Teensy to convert the PWM signal into a digital signal. Chanel three is used to drive a pair of relays that have the PWM input for each motor controller tied to COM and the corresponding signals from the

RC receiver tied to NC and the signals from the teensy tied to the NO so that when button three is pushed the motor controls switch between getting their signal from the controller and the navigation algorithm. Similarly, another relay drives a pair of 12V 50-amp contactors that the motors' positive legs pass through. This relay is activated, closing the circuit to the motors, when buttons four and five are pressed and button six, which is momentary, unlike the others, is not pressed. This allows the motors to be safely back driven and sensor readings to be taken while the vehicle is immobile. The last safety component is the signal light. The 12V signal light was designed with both a flashing and a steady function, with the difference being where the neutral wire goes on the board. To make use of this, another relay is used with the negative leg for the steady on connected to NO and flash connected to NC with the negative wire connected to COM so that when the relay is open, the light stays on, and when it is closed, the light flashes.

The only parts of the electrical system left are the power distribution. The power distribution starts from the battery with 12AWG, which, as previously established, is through the E-Stop. Once it returns to the main box, it lands in a set of terminal blocks with jumpers. From this set of terminal blocks, the 12V system is distributed to the motor controllers via the 12AWG, the light, the contactors, and a 12VDC to 5VDC converter via 22AWG. From the DC to DC converter the 5V system lands a fused power distribution module from which it is distributed with 22AWGaswell as all the signals to the Teensy, camera, ultrasonic sensors, and the RC receiver the negative line is also connected to a set of terminal blocks to act as a reference for the PWM signal.

6. SOFTWARE SYSTEM DESIGN

6.1. Overview

As described in the electronics section, the central brain of the system is a Teensy 4.1 with a sensor suite consisting of a OpenMV H7 camera, which is a microprocessor with image processing on board, six RCWL-1655 ultrasonic sensors, a GT-U7 GPS, and a GY-271 magnetometer, which are read into a Teensy 4.1. In addition to the sensors, the button inputs from the DS 600 Radio controller are sent to the Teensy. This results in a central controller with access to all the information on the robot, allowing for more options if changes are needed down the road or in the field.

6.1 Input Processing

The GPS communicates over UART. While the sensor generates a relatively large quantity of data, we only care about longitude and latitude. This gives the platform its position, but not the direction it travels. To complete this data with a heading, we augment the GPS with a compass. The compass uses I2C to communicate. Like the GPS, it provides more data than is needed since it reads in three axes. The data in this format isn't very useful, so we throw out the Z axis and convert the component heading to get an angle. Now that the robot knows where it is and where it is facing relative to north, the data becomes more useful, but we do more conversion to translate it into terms that make it easier to make decisions with. First, the read longitude and latitude are subtracted from the coordinates of the next waypoint. Using the results, we can use the Pythagorean theorem to determine the distance to the waypoint. Using arctan, it can also be converted to the angle to the waypoint. If the robot's heading is subtracted from this, we can determine the angle relative to the direction the vehicle is facing. While this tells us if we have reached the waypoint and how far we are from it, we do not know if the path towards it is clear.

To understand the vehicle's immediate surroundings, the robot uses the six ultrasonic sensors to see physical obstacles. The ultrasonic sensors each consist of two parts: the waterproof sound transducers, mounted through the vehicle frame, and the processing board, which is DIN rail mounted in the main chamber of the car. These parts are connected via a two-conductor cable integrated into the sound transducers. This cable carries the power from the board to generate the sound wave, and the signal generated by the sound wave bouncing off an object and colliding with the transducer. The boards are connected to the Teensy using two wires for Echo and trigger. These wires are used by pulsing the trigger pin and then using the PulseIn command that looks for a high transition and returns the time in milliseconds from the high transition to the low transition. The board generates this signal, which starts when it gets the trigger signal, sends the sound wave, and ends when it hears the echo. When all six ultrasonic sensors are combined, this gives us a low-resolution point cloud of physical obstacles.

The robot sees pavement markings using an OpenMV H7. The H7 comprises a microprocessor and a camera sensor with an integrated connector. Since this sensor isn't simple, it uses its program. This program looks for collections of pixels that match an expected hue, brightness, and shape, and then checks to see where this blob is in the image. Since the vehicle uses a single camera high up and towards the back, it can determine in what part of the point cloud from the ultrasonics it would fall at this point. We condense the point cloud to a bitmap, reducing the depth resolution to two. The camera can then use a six-parallel binary signal to send the data to the Teensy. This means there is no need for a clock or call and response. This also means that if the camera code is slower than the main algorithm, it won't cause significant problems.

6.2. Output Controls

The primary outputs are the motors. These are controlled using PWM signals, with the min signal being max speed reverse, the max signal being max speed forward, and the middle signal being stop. These motors, wheel motion, and overall motion do not have any dedicated observation and instead take advantage of the speed at which the algorithm can process. Since it recurses so fast, it will make another decision based on new data before a slight intervention in movement can have any meaningful impact. We like to call this dynamic recursion.

The secondary output consists of relays. The first controls whether the signal light flashes or is steady—the second controls whether the motor controllers receive signals from the Teensy or the radio controller. The last controls the motor contactors as a safety disconnect. All of these are controlled with simple digital signals from the Teensy.

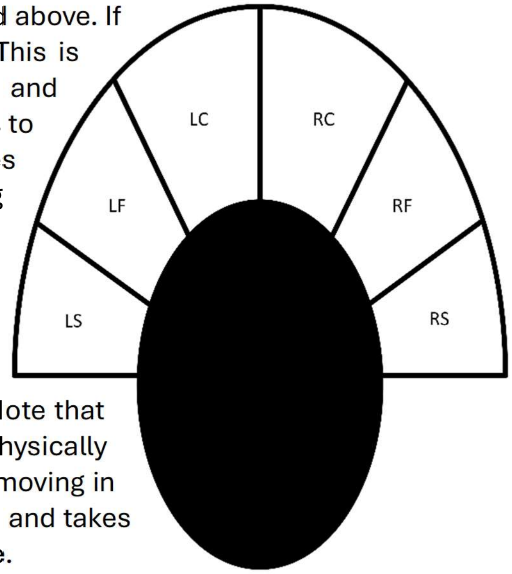
6.3. The Main Algorithm

The vehicle utilizes a custom local map format stored as a six-bit string. While six bits is a relatively limited resolution, the budget limitations on the sensor suite mean that six bits is not actually filtering out as much data as it might initially seem. This map is restricted almost exclusively to the front half of the vehicle. This offers the most valuable data while not overextending the sensor available to our budget.

To account for the fact that the vehicle cannot see behind it, the algorithm never tries to back up. Instead, the algorithm will track any obstacle it encounters with the side facing the center of the course. This allows it to navigate out of dead ends without object permanence because this approach allows the algorithm to deduce that since there should be an obstacle, either the line or something that was blocking the path next to the line, it on a known side therefore if there is a path forward it must lie in the other direction.

The final set of signals from the RC buttons are PWM signals, shorter pulses when off and longer pulses when on. The signals are read using the same pulse as the ultrasonic sensors and then interpreted into a bit based on their length.

The drive algorithm starts by taking all the data, as explained above. If a laptop is connected, it sends it over a serial connection. This is immensely useful for troubleshooting strange behaviors and confirming sensor operation. Next, the vehicle uses the buttons to set the relay controlling the safety light, whether the signal comes from the Teensy or the RC, and the motor contactors. If latching buttons four or five aren't active or momentary button six is active, the robot sets the stack light to solid and disconnects the motors, and if four and five are active and six is not, then the light is set to flash and the motors are connected. If toggle button three is active, then the light is set to solid, and the relay switches motor control to the RC; it does nothing if it is not. Note that button three is checked last and that these outputs are not physically toggled till after that, so the light will only flash if the robot is moving in autonomous mode. Next, the robot examines the local bit map and takes actions depending on what it sees as seen in the following table.



	LS	LF	LC	RC	RF	RS	Move
0	0	0	0	0	0	0	GPS
1	0	0	0	0	0	1	GPS-R
2	0	0	0	0	1	0	L1
3	0	0	0	0	1	1	L1
4	0	0	0	1	0	0	L2
5	0	0	0	1	0	1	L2
6	0	0	0	1	1	0	L2
7	0	0	0	1	1	1	L2
8	0	0	1	0	0	0	R1
9	0	0	1	0	0	1	L3
10	0	0	1	0	1	0	L3
11	0	0	1	0	1	1	L3
12	0	0	1	1	0	0	O3
13	0	0	1	1	0	1	L3
14	0	0	1	1	1	0	L3
15	0	0	1	1	1	1	L3
16	0	1	0	0	0	0	R1
17	0	1	0	0	0	1	L4
18	0	1	0	0	1	0	O4
19	0	1	0	0	1	1	L4
20	0	1	0	1	0	0	R3
21	0	1	0	1	0	1	L4
22	0	1	0	1	1	0	O4
23	0	1	0	1	1	1	L4
24	0	1	1	0	0	0	R2
25	0	1	1	0	0	1	L4
26	0	1	1	0	1	0	O4
27	0	1	1	0	1	1	L4
28	0	1	1	1	0	0	R3
29	0	1	1	1	0	1	L4
30	0	1	1	1	1	0	O4
31	0	1	1	1	1	1	L4
32	1	0	0	0	0	0	GPS-L
33	1	0	0	0	0	1	Straight
34	1	0	0	0	1	0	R4
35	1	0	0	0	1	1	O5
36	1	0	0	1	0	0	R3
37	1	0	0	1	0	1	O5
38	1	0	0	1	1	0	R4
39	1	0	0	1	1	1	O5
40	1	0	1	0	0	0	R2
41	1	0	1	0	0	1	O5
42	1	0	1	0	1	0	R4
43	1	0	1	0	1	1	O5
44	1	0	1	1	0	0	R3
45	1	0	1	1	0	1	O5
46	1	0	1	1	1	0	R4
47	1	0	1	1	1	1	O5
48	1	1	0	0	0	0	R1
49	1	1	0	0	0	1	O5
50	1	1	0	0	1	0	R4
51	1	1	0	0	1	1	O5
52	1	1	0	1	0	0	R3
53	1	1	0	1	0	1	O5
54	1	1	0	1	1	0	R4
55	1	1	0	1	1	1	O5
56	1	1	1	0	0	0	R2
57	1	1	1	0	0	1	O5
58	1	1	1	0	1	0	R4
59	1	1	1	0	1	1	O5
60	1	1	1	1	0	0	R3
61	1	1	1	1	0	1	O5
62	1	1	1	1	1	0	R4
63	1	1	1	1	1	1	Stop

*See the figure before the table
Rx: Turn right
Lx: Turn left

Ox: Turn towards the outside of the course
x: The magnitude of the turn

GPS: Follow the GPS
GPS-x: Follow GPS, excluding turns toward x

The final piece that brings the navigation together is that GPS is replaced with a slight turn towards the center of the course until it reaches the waypoint at the entrance to and after it reaches the waypoint at the exit of no man's land. This has the effect of following the inside line. As explained earlier, this is because it gives a known no-go side in ambiguous situations. The inside is desirable because at the course corners the navigation naturally turns with the course rather than turning against it, which can result in hitting the lane line straight on.

7. CYBERSECURITY AND RISK MANAGEMENT

The NIST Risk Management Framework (RMF) is a structured approach developed by the National Institute of Standards and Technology to manage organizational risk and ensure information system security. It is widely used for cybersecurity risk management across the government and private sectors.

RMF Steps:

1. Categorize: Define the information system and determine the impact levels (low, moderate, high) for confidentiality, integrity, and availability.
2. Select: Choose baseline security controls based on the impact level and tailor them to the specific system.
3. Implement: Apply the selected security controls and document how they are deployed.
4. Assess: Evaluate the effectiveness of controls to determine if they are implemented correctly and producing the desired outcome.
5. Authorize: Senior officials decide if the risk is acceptable and authorize system operation.
6. Monitor: Continuously monitor security controls, assess changes, and respond to new risks over time.

7.1. Threat Modeling: Rival Team Disrupting Robot Software

STRIDE Category	Threat Example	Impact
Spoofing	pretending to be a team member to access the system	Unauthorized access to the robot
Tampering	Altering configuration files or code	Robot malfunction, loss of competition functionality
Repudiation	Denying responsibility after sabotage	Difficulty tracing the source of the attack
Information Disclosure	Stealing source code or algorithms	Loss of competitive advantage
Denial of Service	Causing the robot to fail to start or execute	The robot becomes unusable during matches
Elevation of Privilege	Gaining admin access to override protections	Full system compromise

7.2. RMF Applied to This Scenario

Categorize: Classify the robot system as moderate-impact due to potential compromise of competition integrity and safety.

Select: Use chip versions that can only be written to and keep code on an encrypted thumb drive. This will prevent code theft, and any attempted code insertion will be evident due to the lack of a code base. Remote tampering is limited as the camera and Teensy do not have wireless capabilities. In addition, to prevent tampering, the robot should be kept bolted and never left unattended unless secured in the locked trailer.

Implement: We are using a lockable teensy and the code files for the OpenMV are uploaded as “final,” meaning they are not accessible to do more than delete from the camera. Both of these can only be interacted with using a physical cable. The code is kept on a Kingston IRONKEY Locker+ 50, a password-protected, encrypted thumb drive. With an off-site duplicate on the same model drive. All robot lids are secured with bolts, and of our relatively close and small team present, everyone knows not to or allow anyone to do anything more than look without the project lead or faculty advisor, since he is the only one with the thumb drive and its password to ensure system integrity. The robot must always be accompanied or locked in the team trailer on video surveillance, as the parking lot outside the tent has video surveillance.

Assess: These measures will be sufficient to maintain the platform's integrity. The only security risk we have not covered is the unavoidable weak point of the remote E-Stop. We place the platform's safe operation as paramount and acknowledge that the rules would not allow otherwise. Therefore, we are forced to accept this potential flaw.

Authorize: Based on our experience and the measures taken, we can proceed with the platform as presented.

Monitor: We will continue to monitor the platform's security. If any security risk is discovered, the physical copy of this document will be updated, and new measures will be established.

8. COMPLETE VEHICLE ANALYSIS AND TESTING

8.1. Lessons Learned During Construction and System Integration

One of the most significant challenges during development was slicing of the 3D model of the robot’s chassis. Creating a model was time-consuming, but the greater difficulty was engineering a design that could be segmented into printable chunks and then reassembled without loss of structural integrity or precision.

This platform highlighted the importance of distributable design process, where multiple contributors can work in parallel, and emphasized the need to eliminate bottlenecks in development. The reliance on a few contributors for the largest segment of the design created choke points that slowed progress. Future development efforts will prioritize better distribution of tasks and shared documentation to improve design iteration speed.

8.2. Top Hardware Failures and Mitigations

Potential hardware failures that would prevent successful competition performance include drivetrain, cracking structural components under stress, and loosening or detachment of sensor mounts. To mitigate these risks, the vehicle was designed with modularity as a top priority. Major components—such as motors, mounts, and electronic subsystems—can be swapped out independently. Thanks to the low material cost and short fabrication times of 3D printing, we are able to maintain a stock of spares, including many critical parts and in some cases nearly an entire second robot. This allows for rapid on-site replacement or repair, ensuring that no single point of failure will result in prolonged downtime during the competition.

8.3. Addressing Safety, Reliability, and Durability

The remote E-Stop was designed to fail to safe this means that any failure in the safety system will immobilize the robot. In addition, the remote E-Stop requires two buttons to be pressed to disengage the E-Stop on startup and pressing any button will disengage the motors. The motors being disengaged by the remote E-Stop combined with the fail to safe also prevents the motor from being back fed when the robot is disabled. The primary Estop brakes both positive and negative preventing most possible shorts and providing redundancy. Material selection was also a key consideration in ensuring durability. PETG was selected over PLA or ABS for structural parts due to its improved toughness and flexibility, which provide better impact resistance and reduce the likelihood of brittle failure. This decision contributed to a more robust platform that is better able to tolerate repeated use and minor collisions.

8.4. Hardware Failure Points and On-Site Recovery

The most likely hardware failure during operation is cracking of the front bumper due to obstacle impacts. This has been addressed by reducing risk through material selection and by pre-printing replacement bumpers and ensuring the design can be swapped quickly in the field. In general, the modular nature of the design allows any damaged part to be removed and replaced without requiring a full teardown of the vehicle. With a complete library of printable components and backups of all parts made possible by the low cost of the platform. This has prepared us to respond to virtually any hardware issue that might arise during competition days.

8.5. Predicted Vehicle Performance

- Speed: The drivetrain is geared for a maximum speed of approximately 5 mph, based on motor RPM and gear reduction ratios.
- Ramp Climbing: The drivetrain torque was specified to allow the vehicle to climb over a vertical curb. As a result, ramp navigation is not expected to be an issue.
- Reaction Time: Estimated control loop speed is about 50 Hz, providing sufficient reaction time for obstacles even with a decent amount of speed.
- GPS Navigation: The GPS is only used in no man's land and to determine when we have reached it. For the rest of the course the GPS is replaced with a slight turn towards the center of the course until it reaches. This has the effect of following the inside line or obstacle. As explained in section 6.3. When it is following the GPS it uses the ten turns used for object avoidance combined with to correct as the angle approaches dead on.
- Battery Life: The 7Ah Li-ion battery should support about four hours of testing and autonomous operation under normal load.
- Obstacle Detection Range: Obstacles are detected at up to two meters based on the range of the ultrasonic sensors and the wide-angle lens on the camera.
- Complex Obstacles: The system uses dynamic loop logic and edge-following behavior to navigate around dead ends, switchbacks, and central islands. Adaptive behavior and knowing that the path must be away from the inside line prevents the vehicle from becoming trapped.
- Failure Identification and Recovery: Failures are identified by the team having an intimate understanding of decision-making processes used by the algorithm combined with the ability to read all data out put by the sensors. This lack of a black box in the prosses due the program being written ground up and intimate familiarity with the frame allows the team to quickly notice, diagnose and fix any failures.
- Navigation Accuracy: The GPS sensor was tested while selecting it. The sensor can reliably navigate to a waypoint within 1 meter.

- Predicted vs. Actual Performance: Initial trial data supports the predicted values for speed, battery life, obstacle detection range, and GPS navigation. However, obstacle detection has proven less linear then expected but this is considered acceptable due to it fitting with how the data is used as described in section 6.1. Ramp climbing performance has exceeded expectations due to selecting a far higher bar.

8.6. Software Testing, Bug Tracking, and Version Control

The team uses a timestamp-based versioning system for software builds, with each version labeled in the format NAGLFARE.YY.MM.DD.HHMM using 24-hour time. This provides a clear chronological record of changes and ensures that updates are traceable and reproducible. All software is maintained in only two repositories—one active and one as a clean backup—to prevent accidental branching and data loss. Bugs are tracked manually and logged with revision comments at the top of each version indicating the issue, and resolution method. This lightweight system has been effective for a small, tightly integrated development team.

9. INITIAL PERFORMANCE ASSESSMENT

Initial performance assessments indicate the vehicle meets or exceeds several of its design targets. GPS navigation has proven consistent across mixed environments, and mobility over uneven terrain has been reliable. Obstacle detection triggers as expected, and the control loop handles most course scenarios without lag. Reaction time and system stability were consistent during prolonged testing sessions. The few discrepancies between predicted and actual behavior—such as bumper fatigue and GPS signal fluctuation—are being actively addressed. Overall, the vehicle appears competition-ready, with remaining improvements focused on robustness and fine-tuning of navigation and recovery behaviors.