IMPACT

5/3/2021

Team Lead

Robert McQueen rmcqueen@ltu.edu___

Team Members

Garrick Beaster gbeaster@ltu.edu____

Steven Brownlee sbrownlee@ltu.edu___

Samuel Huggins shuggins@ltu.edu____

Daniel Oliver doliver@ltu.edu_____

Dominic Peichel dpeichel@ltu.edu____

"I certify that the design and engineering of "IMPACT" by the 2020/2021 Lawrence Technological University Robotics Team has been significant and equivalent to what might be awarded credit in a senior design course".

Giscard Kfoury, PhD

Faculty Advisor

Bachelor of Science in Robotics Engineering

Email: gkfoury@ltu.edu

1. Introduction

Mission

The 2021 Lawrence Technological University (LTU) IGVC team's mission is to design, fabricate, and test an autonomous off-road vehicle that can perform lane detection, obstacle avoidance, and GPS waypoint navigation through the use of data acquisition and sensor fusion.

Vision Statement

The 2021 LTU IGVC team's vision is to utilize acquired knowledge from academic studies in a real world project. We aspire to create a team oriented culture to develop our engineering and professional skills.

2. Organization

IGVC Team Members				
Name	**Role**	**Standing**	**Major**	**Estimated Hours**
Garrick Beaster	Drivetrain Lead	Senior	Robotics & Mechanical	235
Steven Brownlee	Mechanical Design Lead	Senior	Robotics	215
Samuel Huggins	Sensors Lead	Senior	Robotics	260
Robert McQueen	Team Lead & Mechanical Design	Senior	Robotics & Mechanical	245
Daniel Oliver	Computing and Electrical Lead	Senior	Robotics	225
Dominic Peichel	Software and Controls Lead	Senior	Robotics	220

Table 1: LTU Team Member Organization and Estimated Hours Spent on the 2021 IGVC Vehicle

3. Innovations

3.1. Ackerman Steering

Although the Ackerman Steering system is an older technology that is not necessarily innovative in and of itself, the innovation comes from the ease in which it can seamlessly integrate with robot's software to make path planning easier to control. This novel approach, never before utilized in this competition, takes advantage of the functionality of the Ackerman system to streamline the writing of the software necessary for steering control. Ackerman steering allows for the fluidic, serpentine movement of the robot, as opposed to the shuddering or more erratic movements associated with differential steering. This also allows the stereovision to more accurately track the contours of the lines of the course.

3.2. Modular Payload Storage

The dimensions of the payload were given in the competition rules of being 8" x 8" x 16" and 20 pounds. The team wanted to be able to hold the weight and dimensions given to us but through further analysis of the frame and the initial design the team determined that the vehicle would allow for more space in the payload area. This addition to the vehicle would allow it to be used in many other applications where the size demands and payloads are different. The team designed a mounting station with a main section and four adjustable arms reaching out from the center as shown in Figure 1. This will allow the user to adjust the grip depending on the load desired to be carried onboard. The extension locks shown in blue in Figure 2 are adjustable on all four sides to allow the user to place the weight of the payload at the center, reducing any uneven loading of the frame.

Figure 1: Modular Payload System

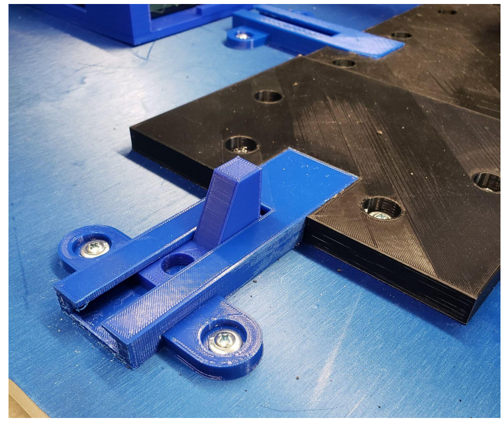


Figure 2: Extension Locks for Payload

3.3 Course Mapping

Our team is using data storage to create a global map from previous runs. When the vehicle first heads on the course it will use only sensor data to navigate and plan a path. It will then store the data from the run so it can make a better map during future runs and create a path plan using a Simultaneous Location and Mapping (SLAM) algorithm. With more data, our vehicle should be able to more efficiently run the course and potentially make more progress than it had on previous runs. Mapping will also allow us to move in reverse if needed despite not having any sensors facing backwards because the course behind the vehicle will be stored in the map. This concept of attempting to map the course and utilize it is completely new to our team and competition for this year.

4. Mechanical Design

4.1. Chassis Design

The chassis design is lightweight, but has a durable base in an attempt to keep it modular. The overall size was chosen to maximize the use of the ackerman steering, be able to hold all the necessary components, and allow for future modifications if required. The material selected was 6061 aluminum alloy because it is lightweight and durable. Although this material is more difficult to weld, the facilities at LTU granted us the ability to perform this task with ease. Without sacrificing the strength of the frame, the use of aluminum resulted in a significantly lower weight than if it had been made with steel. Once the base was designed, as shown in Figure 3, a FEA analysis was performed to ensure it would withstand the anticipated loads. The aluminum tubes on the bottom allowed us to mount a drive train with ease. The sensor mounting frame made of extruded t-slot aluminum (shown in black in Figure 4) enables easy modification and repositioning of sensors during testing. This grants us the ability to maximize the usefulness of our sensors in case our original position did not yield the intended results. Our batteries and payload are also able to be repositioned for ease of access and proper weight distribution. This innovation of modularity made it much easier to optimize our robot in every aspect of construction and assembly to get the most out of every component.

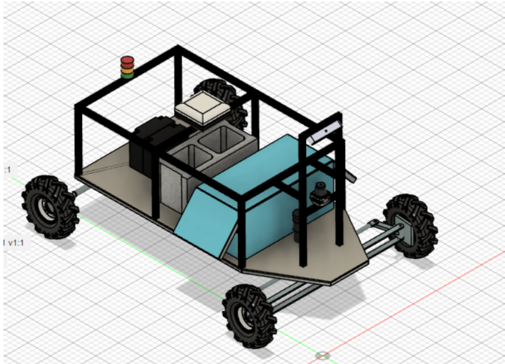


Figure 3: Original CAD Concept

Figure 4: Finished Vehicle

4.2. Drivetrain

The drivetrain was engineered to achieve two key functions. First, to withstand the many hours of testing necessary to achieve autonomous navigation. In order for the drivetrain to handle this testing period, it needed to be dependable, resilient, and low maintenance. Second, in order to provide the necessary stability of the sensors and other equipment on board, the ride needed to be steady and smooth (see Figure 5).

The drivetrain is suspended on a system of seven bearings in the front steering assembly and three in the rear drive assembly, with grease fittings for ease of maintenance. Sacrificial wear parts incorporated into the steering assembly allow a slight rotation along the face of each bearing, and sacrificial aluminum rods are engineered to endure the sliding contact friction that will occur throughout the testing and competition phases of the robot's lifecycle. Together, these components of the design allow the front steering system to have friction-controlled, limited compliance (see Figure 6).

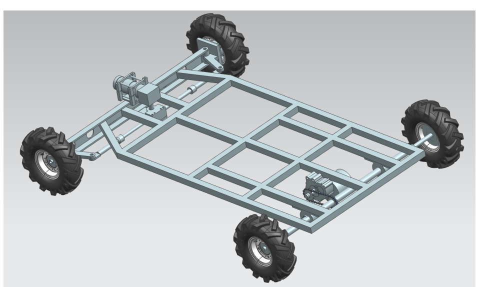


Figure 5: Initial Drivetrain Design

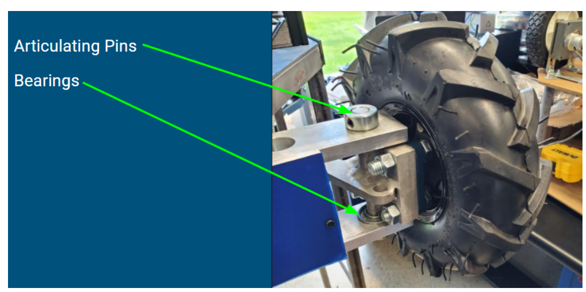


Figure 6: Front Steering Compliance

The Ackerman steering system is responsive and moves with low friction and no binding. The steering motor is a 24 volt, 42 mm motor with a right-angle reduction gearbox. It directly drives the rack and pinion. The rack end-connectors are adjustable for proper tire alignment with the frame. The steering links connect the rack to each hub, which are made out of PLA (3D printed). They are strong enough to withstand the normal operating parameters encountered during competition. Should the vehicle encounter forces that exceed the maximum torque value on the steering motor shaft, these links were designed to break, thus ensuring the integrity of the motor shaft and sparing the motor. For efficiency, they are easily accessible and can be replaced quickly (see Figure 7).

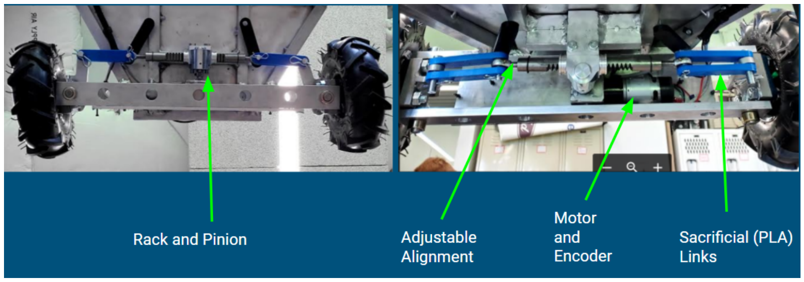


Figure 7: Front steering Assembly

The rear drive is a chain-driven system and is fully adjustable to allow for proper tire alignment with respect to the front tires. Also, the chain and the sprocket alignment and tension are fully adjustable (see Figure 8 and 9). The motor provides a low-velocity continuous torque but also allows for increased torque capabilities for larger payloads to suit the parameters of the overall project objectives.

The tires are pneumatic and have an aggressive tractor tine tread pattern. When fully deflated, the material properties and geometry of the tires dampen vibration and conform to the terrain giving them excellent traction and a smooth ride. The ratio between the weight of the vehicle to the elasticity properties of the tires mimics a well-balanced spring-damped suspension, when operating within the competition's parameters for the robot's velocity.

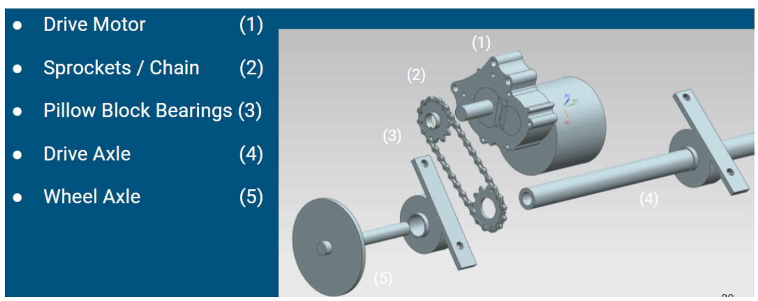


Figure 8: Exploded View Rear Drivetrain

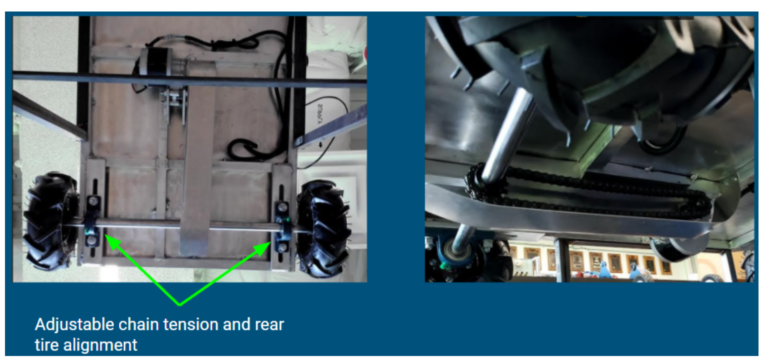


Figure 9: Rear Drivetrain Assembly

4.3 Sensor Mount Design

Each sensor mount was designed to allow for the best performance of the sensor. The camera is mounted using a custom lightweight modular mount. The mount allows the team to optimize the camera location and angle as required using an adjustable hinge (see Figure 10). The mount is positioned at the top of the sensor mounting frame in order to have an optimal field of view over the course. The LIDAR mount is positioned about two feet off the ground in order to optimally detect the obstacles that are in the path. It is mounted to the 20/20 aluminum beam with brackets that came with the LIDAR and drop in t-nuts as shown in Figure 11. Both the height and horizontal position can be changed during testing if needed. The GPS receiver is mounted at the center of the vehicle away from the other sensors so that there is no GPS signal interference (see Figure 12). Like the other sensor mounts, it is able to be repositioned if needed.

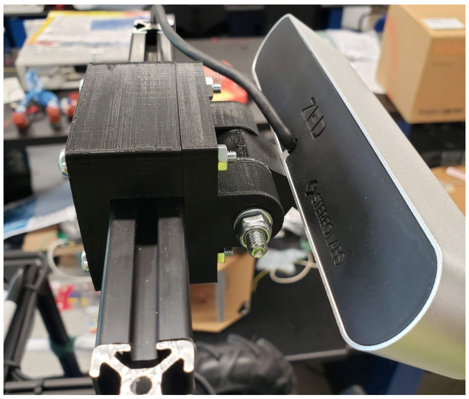


Figure 10: Camera mount.

Figure 11: LIDAR mount.

Figure 12: GPS receiver mount.

5. Electronic & Power Design

5.1 Overview

The power supply for the vehicle is two 12 volt batteries connected in series to create a 24 volt supply. The 24 volts will be used to power the computer, motors, monitor, sensors, and light stack. The Arduino receives 5 volts from the computer to power the IMU and wireless receiver. The Stereo camera runs on 5 volts from the computer as well. The team is using a power supply for the computer and fuse box to distribute power and protect all components from overcurrent . In Figures 13 and 14 below are the electrical diagrams for the power and communication for all computing components.

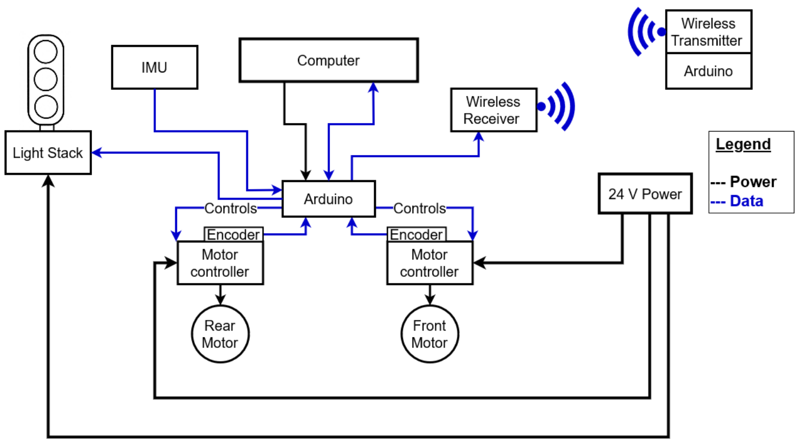


Figure 13: Components connected to the microcontroller.

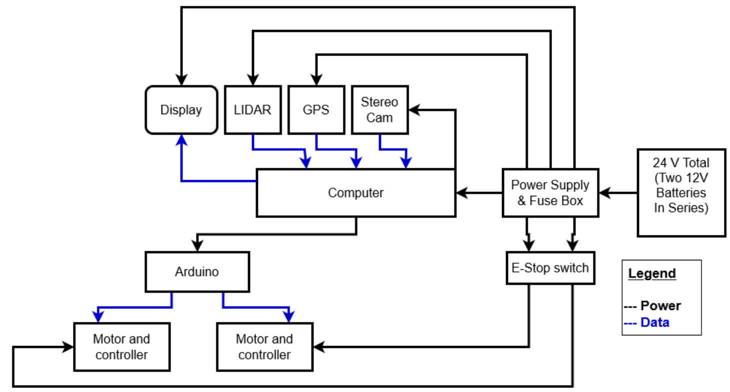


Figure 14 : Components connected to the computer

5.2 Sensors

The vehicle must be able to perform the three main tasks for Auto-Nav: lane following, obstacle avoidance, and GPS waypoint navigation. The vehicle will use a stereo camera, a LIDAR, a GPS receiver, and an IMU as the primary sensors to navigate the course. In order to select the sensors, the team researched sensors that had been used in past competitions, as well as sensors that are used in industrial applications. To select the best sensor for each application, each sensor was evaluated based on the requirements of the competition and the cost to our team.

5.2.1 3D Stereo Camera

In order to achieve lane following and path planning, the team decided to use a camera on the robot. The important factors considered were camera speed, the range of depth perception, and the resistance to glare from the sun. A comparison of the different options considered is shown in Table 2 below.

Camera	Price	Resolution	FPS	Depth Range	Power	Environment	Features
ZED	$350	1080p	30	0.3-25m	1.9W	Indoor/Outdoor	Long range 3D sensing
ZED2	$450	1080p	30	0.2-20m	1.9W	Indoor/Outdoor	Object Detection
Intel RealSense L515	$350	1080p	30	0.25-9m	3.5W	Indoor	Lidar Depth perception
Logitech c930e	$130	1080p	30	N/A	N/A	Indoor	Low-Light Correction

Table 2: Comparison of camera options.

Out of the options considered, the team decided to use the ZED Stereo Camera. This stereo camera provides a 1080p picture at 30 frames per second, and has a 110° viewing angle and a depth range of 0.5-25 meters. It was designed for both indoor and outdoor use, so it is resistant to both the weather and glare from the sun. It also has a USB connection for an easy connection to the PC unit. This sensor has been proven to work in competition, as it was used by the 2019 LTU IGVC team. Although the ZED2 is a higher quality camera with built-in object detection, our team determined that the benefits to the ZED2 were not worth the higher cost because the team planned to use a LIDAR for object detection.

5.2.2 LIDAR

In order to perform object detection and avoidance, the team decided to use a LIDAR. The important considerations were the accuracy, resolution, and weatherproofing of the sensor. A comparison of the different options considered is shown in Table 3 below.

Lidar	 Sick TIM561	 RPLIDAR A3M1	 Hokuyo URG-04LX
Price	$2,800	$600	$1,080
Detection Range	0.05-10 m	up to 20 m	0.02-5.6 m
Angular Resolution (degrees)	0.33	0.54	0.36
Aperature Angle (degrees)	270	360	240
Scanning Frequency	15 Hz	15 Hz	10 Hz
Ambient light resistance	80,000 lx	N/A	10,000 lx
Power Consumption	4 W	2.5 W	2.5 W
Voltage	9-28 V	5 V	5 V
Outdoor use	Yes	Yes	No
Weatherproofing	IP 67 (waterproof)	N/A	IP 64 (water-resistant)
Compatible with Ros	Yes	Yes	Yes

Table 3: Comparison of LIDAR options.

Out of the options considered, the team chose the Sick TIM561 LIDAR. It has a range of 10 meters and an aperture angle of 270 degrees. With an ambient light resistance of 80,000 lx and an IP67 waterproof rating, its outdoor performance is exceptional. It was donated to the team by Sick and is fast and accurate, providing data at 15 Hz and an angular resolution of 0.33 degrees. It will transmit data to the PC unit via Ethernet using TCP/IP.

5.2.3 GNSS Unit

In order to complete the waypoint navigation portion of the Auto-Nav challenge, the team decided to use a GPS receiver. The most important considerations for choosing the GPS were accuracy and weatherproofing. A comparison of the different options considered is shown in Table 4 below.

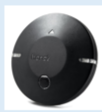
GPS	Price	Receiver Type	Signals Received	Accuracy	Weatherproofing	Voltage	Power
AtlasLink 	$5,000	Dual-frequency, multi-GNSS RTK	GPS, GLONASS, Bediou, WAAS	L-Band: 0.16m RTK: 20mm + 2ppm WAAS: 0.6m	IP67	7-32V	< 4.5W
Garmin 18x USB 	$70	Single-band	GPS, WAAS	WAAS: < 3 meters, 95%	IPX7	5V	0.55W
Here2 	$120	GNSS	GPS, GLONASS, Bediou, WAAS	2.5 m	N/A	5V	N/A

Table 4: Comparison of GPS receiver options.

The team decided to use the Hemisphere GNSS AtlasLink Smart Antenna. It was donated by Hemisphere in 2019, and features precise GPS location information with L-Band accuracy of 0.16 meters. In addition, the AtlasLink has an IP67 waterproof rating, making it suitable for the outdoor environment. The team obtained a sponsorship from them again for the 2021 vehicle in the form of a renewed subscription to their GPS services.

5.2.4 Inertial Measurement Unit (IMU)

The IMU is an Adafruit 9-DOF BNO055, it is used on the vehicle for compass navigation. The GPS does not give a compass heading so the IMU will give the vehicle direction. The IMU has 9 DOF: 3 axis magnetometer, 3 axis gyroscope, and 3 axis accelerometer. This will give us precise orientation information for the control of our robot. The IMU will be connected to the Arduino to retrieve information from the sensor. This will be used in conjunction with the GPS for accurate location and direction of the vehicle during competition.

5.3 Computing Unit

The components for the computer are housed in a custom designed box built by the team. Figure 15 shows the initial CAD design. The box has a sliding Plexiglas panel facing the back of the vehicle that can be removed to access the components inside. The box also has a 120 mm fan on either side, providing cross airflow of 70 CFM per fan. Above the fans are flaps made of Plexiglas that stop rain from entering the box through the fan holes without restricting air flow. To pass the connections from out of the box the side of the box is fitted with water tight connectors. This gives us easy access to all connections while maintaining the water resistance of the box. The completed box can be seen in Figure 16, mounted in the rear of the robot.

The team compared the specifications of the components used by the previous team to the specifications of prefabricated units such as an industrial computer (NI Compact DAQ Controller), a hobbyist computer board (Raspberry Pi), and an embedded computer (Jetson TX2). The team decided the custom built PC Unit will meet all of the vehicle's processing needs and reduce overall cost by not having to purchase new components. The current computer has performance that's better than a hobbyist board and comparable to an embedded computer, while still being less expensive than an industrial computer. The computer has a multi-core processor as well as general purpose I/O connections and ethernet functionality. The main board has an Intel i5-7500 quad core processor that will handle the demands of 3D course navigation and mapping. The graphics card uses Nvidia's Pascal architecture with 384 CUDA cores to do the encoding and decoding of video and image data. The computer has 8 GB of RAM and a 250 GB SSD that will be more than adequate for storing all of the information needed for path planning and mapping.

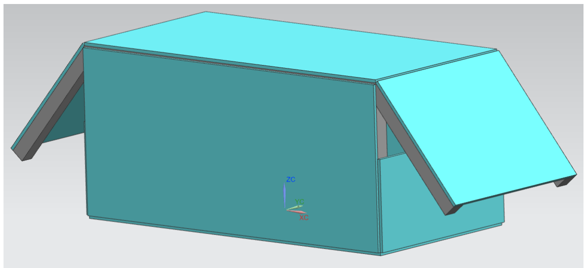


Figure 15: CAD model of computing box

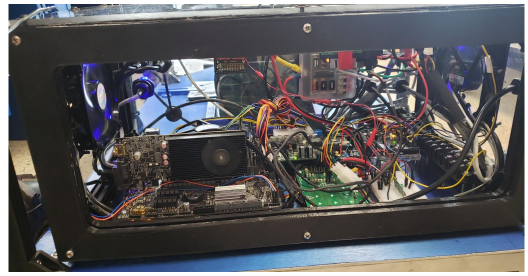


Figure 16: Completed Computing Box

5.4 Sunlight Readable Monitor

The Teguar TSD-45-12 is the sunlight readable monitor the team selected to use. The monitor is used for visual output of our computer and is vital for debugging the vehicle while testing and at competition. The monitor was made to be used in outdoor applications; it has a waterproof rating IP66 housing and has up to 800 nits of brightness for viewing in bright sun. This means that the screen is usable in the rain or in direct sunlight.

5.5 Motor Controllers

The motor controllers used are a SyRen 25A motor driver for the rear motor and a SyRen 10A motor driver for the front steering motor. The steering motor driver also has a Kangaroo x2 motor controller that allows the computer to read encoder feedback and set the range of motion of the steering motor.

5.6 Power Distribution

Our vehicle has two 12 volt batteries that are each 18 Ah. They are connected in series giving a total voltage of 24 volts. This gives a total power output of 432 Watt-hours. The total power needed for our vehicle is calculated to be 384 Watts. Table 5 below shows power consumption for each component used. The energy provided by the batteries will be sufficient to run our vehicle for approximately 67 minutes at max usage.

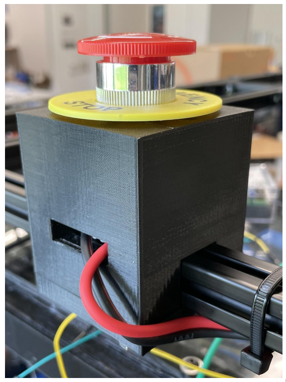
Component	Power Consumption	Operating Voltage	Source
Computer	110 W	24 V	Battery
LiDAR	4 W	24 V	Battery
Stereo Camera	2 W	5 V	Computer
GPS	3 W	24V	Battery
Monitor	8 W	24 V	Battery
Rear Drive Motor	250 W	24 V	Battery
Front Steering Motor	4 W	24 V	Battery
Light Stack	2 W	24 V	Battery
Arduino	1 W	5 V	Computer

Table 5. Power Requirements of Electrical Components

5.7 Safety Devices

5.7.1 Mechanical E-stop

The mechanical e-stop is placed in the center rear of our robot. It is directly wired between the fusebox and the motor controllers. This means that when the e-stop is engaged there is no power going to the motor controllers, but all other components such as the computer and safety light will still be active. It is mounted with a modular custom printed plastic mount, shown in Figure 17.



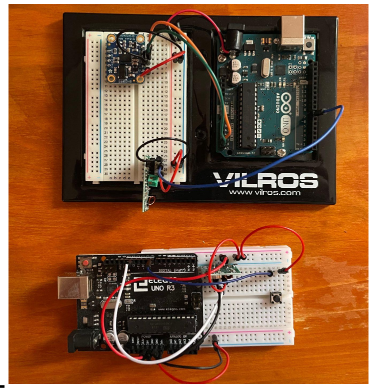


Figure 17 : Mechanical E-stop *Figure 18: Wireless E-stop Configuration*

5.7.2 Wireless E-stop

The wireless e-stop is a 315Mhz RF Transmitter and Receiver that connects to the Arduino on board. The transmitter will be connected to another Arduino that will continuously ping the onboard receiver. Any loss of signal will result in the vehicle immediately stopping until the signal is restored. The vehicle can also be triggered to stop by using the button connected to the transmitter Arduino. Both sides, receiver (upper) and transmitter (lower) are shown in Figure 18 above.

5.7.3 Safety Light

The safety light is the indicator of the state of the robot. The safety light is a tricolor (red, yellow, green) light tower. It runs off of 24 volts and has a high output so it can be seen in sunlight. It is controlled by our Arduino using a Texas Instruments LED controller EVM that was donated to the team as seen in Figure 19. It turns on when the vehicle turns on. The light will be continuously lit when the vehicle is powered and flashing during autonomous mode. To control this tricolor light tower the team needed a high side driver (a switch on the power side not the more commonly used switch on the ground). This is necessary because the lights have a shared ground. Using this board the team can control all three lights independently using PWM pins from our microcontroller. The safety light is placed on the highest point of the vehicle frame so that it is easily visible. It has a printed plastic modular mount allowing us to change its location if necessary.

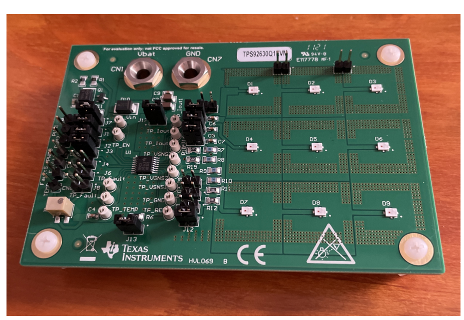


Figure 19:Texas Instruments LED controller EVM

6. Software/Control Strategy

6.1 Overview

The robot's program is developed in C++ using Visual Studio Code for any program nodes that require more intensive computing, and Arduino for simple calculations, safety features, and motor commands. To effectively manage multiple nodes of code, the Robotic Operating System (ROS) framework was used.

6.2 Obstacle Detection and Avoidance

For Obstacle detection, the team is using a Sick LIDAR and a ZED Stereo Camera. The Sick will be the primary sensor for detecting the obstacles, while the camera's primary purpose is to detect lanes. For local path planning, the Stereo Camera is referenced to verify the relative heading of the robot based on following lanes and then the LIDAR data dictates exactly the angle for the vehicle to move to stay within lanes and avoid obstacles.

6.3 Software strategy

Using the previously mentioned software, the code strategy denoted in Figure 20 was developed. This diagram shows the different nodes of code that will be running. The blue boxes are the sensors inputting data to the computer, namely the LIDAR, Stereo Camera, and GPS. The grey boxes denote different code blocks that will be running simultaneously. The arrows denote data being sent between nodes using ROS publishers and subscribers. The Robot node indicates the Arduino communication with the robot. The Robot Commander node outputs the desired heading and speed based on the path planning algorithm, and the Arduino sends the motor commands.

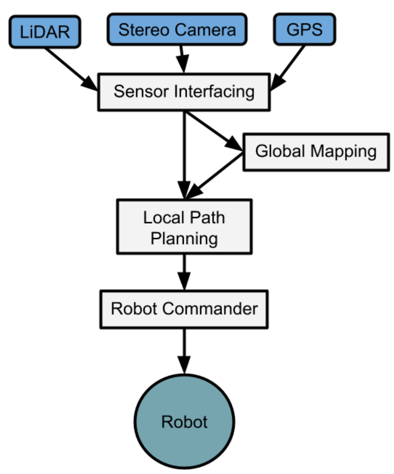


Figure 20. Program blocks

To control the motors the team is using two different control strategies. For the rear motor, which dictates power and speed of the robot, the team uses a boundary speed control. The vehicle moves at 5 mph if there are no obstacles around, but when an obstacle comes into view of the LIDAR the vehicle slows down based on the distance from the obstacle. This ensures that if the vehicle does not end up running into something, it will be moving at a slow enough speed so it won't damage the robot, the obstruction, or anyone or anything in the immediate vicinity. It also functions as a tactic to give the vehicle more time to steer around obstacles, especially when there are multiple obstacles to avoid.

The second strategy employed is a zonal position based strategy for the front steering motor. The robot takes in the camera data and filters it to find where the lanes are. From there, it separates the image into 10 different zones, 5 for the left lane and 5 for the right lane. Counting pixels in each of the zones tells the vehicle which general direction it should be heading from the following options: far left, short left, center, short right, and far right. The robot then takes in the LIDAR data to check the zone it expects to head towards to see if there are any obstacles in that zone. If there is an obstacle in the zone, the vehicle will attempt to head towards the next furthest zone, i.e. from short left to far left, short right to far right, etc., and check for obstacles there until it finds a zone and path that it can follow to stay within lanes and avoid obstacles. If the expected direction from the camera cannot be followed due to obstacles, the default direction will be straight forward.

6.4 Map generation

Using the gathered LIDAR data as well as a filtered 3D image, a cost map is created to give a direction for the vehicle to head while staying within lanes and avoiding obstacles within the range of the LIDAR and camera. Each run will store the cost map and overlay it on any previously stored cost maps to create a better prediction for where to go on a global scale rather than only using the data at instantaneous points in time.

6.5 Goal Selection and Path Generation

Each waypoint is programmed with an operating mode. When the first waypoint gets set, the robot uses the camera data to stay within lanes while avoiding obstacles with the LIDAR. If any previous runs have been completed, it will use the previous data to have a better long term prediction of where it should end up. During the section of no man's land the vehicle will follow GPS data to make it to the next waypoint while using the LIDAR to continue avoiding obstacles, however it will not check the camera for lane following as there will be no lanes to follow. During the lane following sections, the vehicle will follow the lanes and avoid obstacles, only using the GPS to check if it has made it to the waypoint.

6.6 Additional Creative Concepts

The team integrated ROS with Arduino as a means of communication between the main computer and the Arduino Mega. This allows transfer of data from the Inertial Measurement Unit (IMU) and motor encoders from the Arduino to the main computer. Since the computer has vastly more available RAM and computing power, this enables faster reaction time for the robot and less chance for program error.

7. Failure Modes and Failure Points

7.1 Vehicle Failure Modes

7.1.1 Software Safeguards

The Arduino controls our safety features as well as controlling the robot. This means that if an E-STOP, either wired or wireless, is hit, the Arduino will not send any motor commands to the motor controllers and will light up the LED to indicate that it is in E-STOP mode. It will also notify the main computer to clear up bandwidth between the computer and the Arduino. Once the E-STOP is cleared, the system will go into a standby mode until an operator sends a new start command. This ensures that if a problem with the robot occurs, it is immediately and safely stopped so that no one will be hurt and no property will be damaged.

7.1.2 Path Planning

With Ackermann steering, going backwards is a little less accurate than going forwards. To account for this, our path planning algorithm attempts to go through the course without going backwards. This could be problematic at a point where the vehicle cannot go any further forward due to being too close to an obstacle. In this case the vehicle will have to back up to continue along the course. To make the most accurate movement going backwards, the vehicle will go straight back. It has to check the mapped data in order to make sure this movement will not run into another obstacle since the vehicle does not have any sensors on the rear end of our robot.

7.2 Vehicle Failure Points

7.2.1 Mechanical

The front steering motor shaft is a ¼ inch in diameter. A set screw through the rack and pinion shaft mates the steering system to the motor. If the steering links fail to break as intended in an extreme over torque situation the small motor shaft could potentially shear. Because the steering motor can be removed and replaced in the matter of minutes an additional one will be available in reserve. Also additional steering links will be kept stocked because they are the intended failure mode engineered into the steering design to limit the potential of breaking the motor as specified above.

7.2.2 Electrical

The team has waterproof connectors on the side of the computing box to allow for ease of plugging and unplugging of sensors and electrical lines. The team also has a battery that can be quickly replaced with a backup so that the robot doesn't have significant downtime while charging the battery which can take several hours. Inside of our computing box is a fuse box that distributes the power to the various sensors and components. Each component is fused for proper safety. A fuse was also added coming directly from the battery, and a main shut off switch is placed on the computing box. While testing, the team experienced a wire coming loose that caused a short circuit on one of our motor controllers. The team has worked to ensure all wires are secure so this issue does not occur again. The team has also purchased an extra motor controller to make sure that a spare is available in case an accident occurs again. An extra Arduino is also available should the team run into any problems with our microcontroller. Should electrical issues occur at competition despite our preventative measures, the team will have extra wiring materials and equipment to troubleshoot and fix any electrical problems.

8. Simulations

8.1 FEA Analysis

Our team's accumulated knowledge allowed us to create a design that was durable, easy to build on, and reasonably light weight. The robust and modular design allows for additional carrying capacity and the ability to handle an increased payload with little deformation. The modularity of the frame allows additional options in the future if required. The team ran a Finite Element Analysis on the design to simulate the forces the frame would be subjected to. Tests were performed with the expected forces and with extreme forces acting on the frame, and the tests yielded favorable results. No redesign was required because the results proved our design was robust for loads well over anticipated levels. The FEA in Figure 21 shows the results of a 300 pound distributed force. This well exceeds the intended load of 100 pounds and resulted with a deformation of 0.460mm. The additional durability costs very little extra weight to attain and grants many benefits to the platform beyond the original scope of the project. The FEA analysis in Figure 22 shows a very hard hit to the right rear tire and the effect it would have on the frame. This would be equivalent to traveling over a large bump at the maximum allowed velocity of 5 mph and all the force taken by only one tire. The simulation yielded a deformation of 1.853 mm. This was a worst case scenario, because the robot could not go any faster and an obstacle any larger would be too tall or deep for the vehicle to traverse. It would be best to avoid such obstacles and the controls reflect this consideration. These tests prove our frame is designed to handle the assigned task, and able to handle more than the anticipated load for this competition.

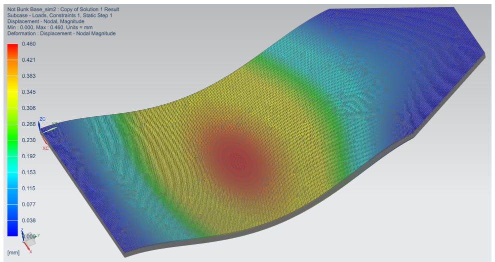


Figure 21: 300lb force distributed load on frame FEA

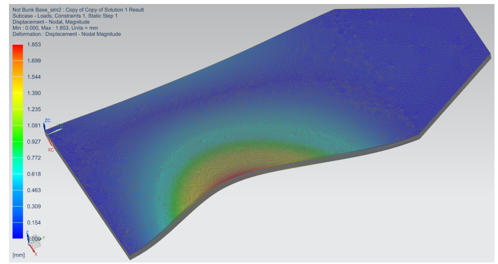


Figure 22: 200lb force bump to right rear tire FEA

9. Testing

The team has constructed the vehicle and tested it for strength and durability. It can take all required loads without failing, as well as additional loads that can be added into our modular payload system. Upon testing, the vehicle can carry an additional 50 pounds without any notable deformation to the frame. These deformations are small enough that the stress of the static load or impacts does not cause enough strain to cause permanent damage except in the most extreme situations. Those situations would not be classified as normal operations and would be avoided.

The TI LED controller is an evaluation module so the team was able to do initial testing of the LED control with the LEDs built into the board before wiring the board to the safety light. Once the board was tested to ensure that each light could be controlled individually the team then was able to wire the microcontroller and the safety light to the controller board. The lights were tested and confirmed to work as expected and needed per competition rules.

Initial testing of the robot motion has been completed. The team tested the robot moving and turning using a controller. This tested the communication between the computer, Arduino, and motor controllers, as well as range of motion and speed of the robot. Each motor controller works and the motion of the robot is as expected from basic joystick controls.

Testing of each sensor was also performed. The ZED camera shows views from both camera inputs into the program. The SICK TIM561 LIDAR has been communicating with little to no noise when tested inside. The GPS has been tested and after connection with Hemisphere for an updated subscription code the team found it works as intended.

10. Performance Assessment

Overall, the vehicle has been working as expected. There have been some issues with tuning the motor controller which were due to the limit switches being too close to the encoder causing electromagnetic interference. The team added shielded wires to help solve the issue and have been working to test the tuning cycle again. The next task the team has to work on is finalizing the path planning algorithm and driving the robot autonomously.