UCF

COLLEGE OF ENGINEERING
AND COMPUTER SCIENCE

The “NI-hicle”

Senior Design Project Documentation (Final)

Autonomous Vehicle Demonstrating Real-Time Navigation
Sponsored by Dr. Guo
Group 20 – *“The Knights of Ni (Navigation Independence)”*

THE KNIGHTS
OF
Ni
NAVIGATION
INDEPENDENCE

James Beckett, EE – Power Systems
Alex Jenkel, EE – Communications and Signal Processing
Juan Velasquez, EE – Communications and Signal Processing

Fall 2019 – Spring 2020

Table of Contents

List of Figures

List of Tables

1.0 Executive Summary

According to Moore's Law, the number of transistors on a microchip doubles every two years. Within the last decade Moore's Law has become more evident as technology has become more sophisticated, faster and powerful. This effect has been significant in the areas of cellular phones, automobiles, household electronics, etc. Cell phones have become personal computers, automobiles are able to maneuver on their own and household electronics can be accessed remotely. As a result, the world around us offers totally new families of products which are more efficient, reliable, interactive and safer products than most of us could have ever imagined. Nowhere is the trend more apparent than in the current transformation of the automobile. This idea of having fully autonomous vehicles seemed like science fiction, were an automobile would seemingly go from one destination to another without the aid of a human driver. This idea went from science fiction to become a reality due to the rapid increase in technological advancement. The beginnings of autonomous vehicles started in the early 20th century where the idea of a self-driving car was taking effect. One system that helped paved the way is the Global Positioning System (GPS). A GPS can be used via smartphone or as a standalone unit and can help a person navigate to any available destination. While GPS only provides guidance to a destination companies such as Google, Amazon, Domino's, Walmart and Tesla have been working on creating new technology to make vehicles fully automated. Advanced Driver Assistance Systems (ADAS) are now being included in automobiles to aid the driver with parking, blind spots, slowing or stopping the vehicle if necessary. While the automobile is not fully automated, beginning steps are taking shape in the development of fully automated vehicles. Through innovation and experimentation Google and Tesla have made this idea possible thus expanding opportunities to dwell on this newer technology. The idea to work, develop and possibly innovate was a key motivation in the selection of the project.

The goal of this project is to utilize a Real Time Operating System (RTOS) to implement navigational capabilities in a remote-controlled vehicle. This goal will be carried out using various onboard sensors such as radar, ultrasonic, optical and an onboard 3D camera connected to a Graphics Processing Unit (GPU) and a Central Processing Unit (CPU). The various sensors will feed their data to a central computing unit which will make the necessary decisions to either adjust the steering, adjust speed performances to accelerate or decelerate, avoid collisions or avoid obstacles and stop the vehicle. The expected outcome of this project is to have a vehicle that can autonomously traverse a course that could feature turns of various angles, twists, hazards, obstacles, uneven surfaces and even dead ends. The vehicle would be capable to navigate through the course as quickly as possible without colliding and avoiding obstacles all while adhering to the safety and road standards. Additional features of the course could include reconfigurability of the track, where the walls and obstacles could be translated to create a new track, and the vehicle would still be able to navigate through it accurately. The project will be featured on a 1/10 scale model Remote Control Car (RC). Safety, cost, and time constraints restrict the project to be done on a smaller scale. Translating the algorithm, hardware, and technology into a “real-world” problem looks as simple as having a processor that responds in real time to different flags and interrupts in a system while still operating efficiently and correctly. This work could easily be applied to existing technologies such as Roomba or other autonomous vehicles that are already in production.

2.0 Project Description

Nevada was the first state to authorize the operation of autonomous vehicles in 2011. Since that time, twenty-one other states have passed legislation related to autonomous vehicles. Florida’s legislation, passed in 2012, declared the legislative intent to encourage the safe development, testing and operation of motor vehicles and autonomous technology on public roads of the state and found that the state does not prohibit nor specifically regulate the testing or operation of autonomous technology in motor vehicles on public roads. Florida’s 2016 legislation expands the allowed operation of autonomous vehicles on public roads and eliminates requirements related to the testing of autonomous vehicles and the presence of a driver in the vehicle. [1]

The intended project is sponsored by UCF professor, Dr. Guo. Dr. Guo’s expectations are to take a remote-control toy car and convert it into a fully autonomous vehicle using various onboard sensors. Details surrounding the competition that the vehicle will be participating in are provided later in section 4.1.6, but to highlight: there will be two separate competitions—timed, single-car racing and competitive, two-car racing. Our project will aim to be superior in both of those racing contests. While speed is obviously paramount in any form of race, Real-Time Systems focus on accuracy and correctness in lieu of completing tasks the fastest.

The following sections are functional descriptions of the project and its parameters. Great detail is expounded so that absolute clarity can be achieved with the project’s standards, objectives, technologies, methodologies, and future interests. The end goal of this paper is to allow future participation in the competition using a new technology that is not widely used in the competition at this point in time.

2.1 Project Motivation and Goals

With the proliferation of autonomous vehicle technologies, a demand for real-time detection of obstacles has increased. Top companies are investing in this technology and are using various techniques to achieve automation. The most effective method for real-time processing of obstacle data is through utilization of a real-time operating system to process images and sensor data. Real time operating systems (RTOS) is an up-and-coming area of research that could be revolutionary within the computing and embedded systems domains. Because of the high demand by the automotive industry to pursue groundbreaking technology, which could eventually lead to self-driving automobiles, our group was motivated to pursue this project through the applications of embedded systems, linear control systems, communications, and due to an interest in the platform's competitive element. While this specific implementation of RTOS could be considered trivial, the work can easily be carried over into a more rigorous computing or scheduling setting for autonomous vehicles.

The primary goal of this project is to produce an autonomous vehicle with the capability to navigate a reconfigurable course without striking the course walls or another autonomous vehicle on the course. The vehicle should be capable of navigating the course without any outside assistance and without following a predetermined route. The secondary goal of this project is to provide designs and techniques that can be utilized by Dr. Guo in his research of autonomous vehicles and apply our design to the ongoing project at UCF. An ancillary goal for this project is to race in a competition. The goal of entering the competition would be to showcase the results of the project and compare those results with other educational institutions' vehicles.

2.2 Objectives

The objective of this project is to design a fully autonomous car that meets all of the customer requirements and performs its function adequately. This project will set itself apart by being as close to fully autonomous as achievable by not requiring any "learning input"—i.e. driving through a course first or having the route preplanned—and will instead use its various sensors to make navigational decisions. In addition, there is also a competitive factor in which multiple cars can race each other at events, which would require the vehicle to have the ability to navigate quickly while still guaranteeing that the group's vehicle will not collide with its competitor(s).

The architecture for our object avoidance techniques are two-fold. The primary input source for object-detection system is the ZED Stereo Camera, which has two lenses that offer a 3D perspective along with distance measurements and point cloud analysis. The secondary input source will be the myriad sensors that are associated with the project. These sensors include ultrasonic sensors, radar sensors, and a rotary encoder, and will be streaming their data to either the Jetson or the MCU, after which the MCU will send out steering commands to the speed controller and steering servos.

2.2.1 Autonomous Navigation

This project defines autonomy as being able to navigate quickly, accurately, and efficiently. Numerically this takes the form of:

- Navigating a course with a speed of 5 miles per hour
- When navigating through a course with multiple twists and turns, the vehicle will take the quickest route to complete the course
- The vehicle will stay at least 1 foot away from each wall
- The vehicle will continue to navigate through the course until the user issues the "kill" command

2.2.2 Object ranging and collision avoidance

The architecture for our object avoidance techniques are two-fold. The primary input source for object detection is the ZED Stereo Camera, which has the ability to identify objects' distances up to 20m away. Once this data is sent to the Graphics Processing Unit (GPU), the information will be processed and then forwarded to the microprocessor, which will be interfacing with the second part of the object avoidance scheme. The microcontroller will be in charge of making navigational decisions—such as adjusting speed and turning angles—and may need to utilize secondary sensors in order to form a more precise picture of its environment. There will be a programmable threshold such that if the camera data either comes back with error-ridden data, or if the camera detects an object within the threshold, the microcontroller will activate all of its secondary sensors so that the shorter distances can be accounted for (the camera will not measure distance accurately at distances less than 20 cm).

2.2.3 Peripheral Interfaces

The secondary proximity sensors along with the system diagnostics will be the peripheral interfaces that will provide the data to the MCU for analysis, corrective action and directional outputs. Final selection for the proximity sensors consists of radar, and ultrasonic sensor units. In addition, the 2.4GHz controller furnished with the vehicle chassis will be integrated into the design to facilitate a failsafe measure to shut down the autonomous vehicle in the event of malfunction.

2.3 Requirements Specifications

In order to meet the project and course goals, our design will need to fit all power, communication, and autonomous specifications within the dimensions of the vehicle chassis. A height restriction will also be considered to avoid air resistance that could potentially affect vehicle performance but will not strictly be considered a requirement specification. Communications specifications are summarized below in Table 1 and the specific requirements specifications are summarized below in Table 2.

Table 1 - Communication Specifications

Constraint	Definition	Quantity	Units
Wireless Communication	•Frequency	2.4	GHz
	•Response Time	100	ms
	•Number of Modes	2	--
	•Automation Reduction	[2,4]	SAE Autonomy Levels
UART	•Baud Rate	115200	bps
I2C	•Clock Rate	4	MHz

Table 2 - Requirements Specifications			
Constraint	**Definition**	**Quantity**	**Units**
Size	•Max Height	1	ft
	•Max Weight	15	lb
Autonomous	•Object Size Detection	6x12	in
	•Object Detection Range	1	m
	•Autonomy	4/3	SAE Autonomy Levels
Collision Avoidance	•Object Detection Response Time	1	second
	•Stopping Distance from 5mph	2	ft
	•Object Response Distance	3	ft
	•Minimum Distance from Obstacle	6	in
Real Time Navigation	•Max speed	10	mph
	•Acceleration Time from Full Stop	10	seconds
	•Stop Time from Max Speed	5	seconds
Power Management	•Source Voltage	11.1	V
	•Source Capacity	5000	mAh
	•Down Converted Voltages	1.8, 3.3, 5	V
	•Battery Type	Rechargeable	--
Liability	•User Adjustment to Autonomy Level	0/1	SAE Standard Level of Autonomy

2.3.1 Onboard Image Processing

Onboard image processing will be handled by a sponsor provided NVIDIA Jetson TX2. The Jetson TX2 will be attached to a carrier board to facilitate communications with the microcontroller and the external sensors. The ability of the Jetson TX2 to take input images and process them with a Real-Time Operating System will be instrumental in course navigation and object avoidance. The TX2's ability to process data quickly is enhanced by its multiple processors, accelerators, and large memory banks. Once the images have been thoroughly analyzed, the graphics information will be passed to the main microcontroller, which will then issue navigational commands.

2.3.2 Onboard 3D Camera

An onboard camera will be required to capture images that can be processed to interpret obstacles from the track and navigate the vehicle through a reconfigurable course. The onboard camera is a sponsor-provided ZED Stereo camera which will provide 3-dimensional images that will be deciphered by the image processor to navigate the reconfigurable course. The main features of the camera are its wide viewing angle, its high frame rate, and its ability to operate in ROS (discussed in section 3.4.1). The ZED Stereo Camera communicates through USB, so it will have to be read through the Jetson's USB port.

2.3.3 Proximity Sensors

Proximity sensors will be a combination of ultrasonic, radar and/or infrared sensors. These secondary sensors will serve a dual-purpose of corroborating the data that the ZED Stereo Camera measures, as well as acting as a failsafe measure in case the ZED Stereo Camera becomes inactive or unreliable while the vehicle is operational. An additional use for the secondary proximity sensors will be to enable the vehicle to avoid collisions with other autonomous vehicles during a competitive setting. For racing purposes, it could be beneficial to have at least one rear-facing sensor that can identify where the competitor-vehicle is, which in turn would allow for aggressive programming that would allow the project vehicle to cut off the other vehicles, giving our project an edge in the race.

2.3.4 Microcontroller

The onboard microcontroller will take inputs from the image processor, secondary sensors, battery and speed sensors to provide steering outputs and speed controls in order to avoid collisions and navigate a course in real-time. It goes without saying that the microcontroller should have the ability to communicate with all of the peripheral devices but is will also need enough GPIO channels to be able to read several HC-SR04 Ultrasonic Sensors. Currently the project will use five of these ultrasonic sensors, and so the microcontroller would either need five dedicated GPIO channels, or the PCB will need a multiplexer to cycle through which units are being read. Using a switch or multiplexer would be problematic as the throughput for reading the sensors one at a time would be astronomical compared to having enough channels to read each HC unit. It goes without saying that five GPIO pins is an insignificant amount of GPIO pins as most microcontrollers have on the range of fifty to 100 pins, but if a smaller microcontroller would be used in the future, this is a minimum for the ultrasonic sensors.

2.3.5 RF Communications

RF communications are already integrated into the remote-control unit operating at 2.4GHz of the sponsor-provided vehicle. The receiver that is included in the car chassis is a 3-channel receiver where all channels are centered around 2.4 GHz. This project will utilize the integrated communications medium in order to build a failsafe for the system which will help mitigate liability and maintain safe operation of the vehicle at all times. Normal operation of the RC car, where the user is in control of the car, utilizes two of the channels in the receiver to send speed and steering commands. This project will seize the paths of those signals to go towards the microcontroller instead of directly to the speed controller and steering servos, so that the user can simply utilize the given transmitter to issue a “kill signal” interrupt.

2.3.6 Power Supply

A power supply will be designed, fabricated and tested in order to take an input from a 11.1V_{DC} Lithium Polymer battery pack as well as to have the capability of accepting input voltages from 4.5 to 12V_{DC} and between 300 to 7500 mAh battery sources. The power supply will regulate the voltage and provide appropriate useable voltages to the various system components. Battery consumption is not a concern for this project as the LiPo battery will be rechargeable. It is only in our interest to not create such a large power draw that the battery will die in a short period of time so that we can effectively test it outside of a racing capacity.

2.3.7 Steering Controller

A PWM signal will be supplied from the MCU to the servo to move output spline through 180° range of angular rotation translated via steering linkage to 60° of steering rotation. The PWM signal will provide these inputs in order to steer the vehicle platform to avoid object collisions.

2.3.8 Drive Motor

The drive motor will be supplied 6W from the motor controller. The motor controller will determine the supplied current and polarity to propel the drive motor in forward or reverse rotation and determine the velocity of the drive motor. The rotation of the drive motor shaft will be translated to the vehicle tires via a geared transmission that will turn the vehicle platform’s main driveshaft. Rotation of the vehicle platform’s main driveshaft will drive another geared transmission that will rotate the axles of the vehicle platform’s wheels, thus rotating the attached wheels.

2.3.9 Motor Controller

The motor control will supply voltage and current based upon the internal BEC of the motor controller. This value for the sponsor-provided motor controller will be a max 6V at 1A. Polarity of the voltage will determine the rotational direction of the drive motor and the amount of current supplied to the drive motor will determine rotational velocity of the drive motor.

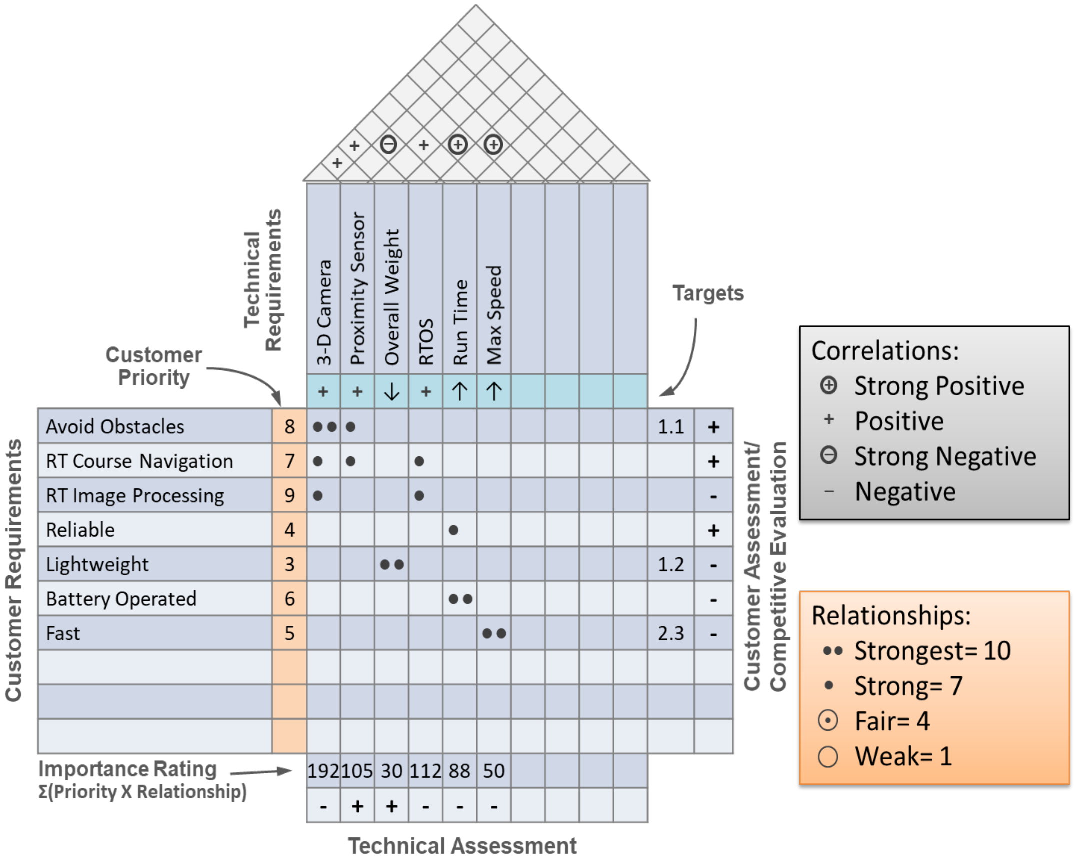
2.4 House of Quality Analysis

The house of quality diagram, shown in Figure 1, is a process utilized for project development which is based off customer requirements for product or process development. The requirements are weighed against capabilities and resources of those seeking to meet the customer’s requirements. The house of quality diagram below in Figure 1 indicates user requirements to include:

- Avoid obstacles
- Real-time course navigation
- Real-time image processing
- Reliability
- Lightweight
- Battery operated
- Fast

Technical requirements included to meet customer requirements include:

- 3-D camera
- Proximity sensors
- Overall weight reduction
- Real-time Operating System (RTOS)
- Run time
- Max speed



Customer Requirements	Customer Priority	3-D Camera	Proximity Sensor	Overall Weight	RTOS	Run Time	Max Speed	Targets	Customer Assessment/ Competitive Evaluation
		+	+	↓	+	↑	↑		
Avoid Obstacles	8	●●	●					1.1	+
RT Course Navigation	7	●	●		●				+
RT Image Processing	9	●			●				-
Reliable	4					●			+
Lightweight	3			●●				1.2	-
Battery Operated	6					●●			-
Fast	5						●●	2.3	-
Importance Rating Σ(Priority X Relationship)		192	105	30	112	88	50		
Technical Assessment		-	+	+	-	-	-		

Figure 1 - House of Quality Analysis

By comparing user requirements with technical requirements, we indicate the following:

- Strongest relationship between obstacle avoidance and the 3-D camera
- Strong relationship between real-time course navigation and 3-D camera

- Strong relationship between real-time processing and 3-D camera
- Strong relationship between obstacle avoidance and proximity sensors
- Strong relationship between real-time course navigation and proximity sensor
- Strongest relationship between lightweight and overall weight
- Strong relationship between real-time navigation and real-time operating system
- Strong relationship between real-time image processing and real-time operating system
- Strong relationship between reliable and run time
- Strongest relationship between battery operated and run time
- Strongest relationship between fast and max speed

3.0 Research Related to Project Definition

Autonomous Vehicles are a thriving emerging technology that already has had great impact, which may revolutionize transportation, while substantially enhancing traffic safety and efficiency. Market interest for autonomous vehicles currently is for the purpose of delivering goods or ride sharing with a long-term goal of reaching Level 4 or Level 5 autonomy within 10 years.

Some examples of emerging projects based on self-driving vehicles are: Google's self-driving car Waymo, Tesla's Autopilot control, Amazon's plans to use self-guided drones for deliveries, Domino's Pizza are actively developing electric vehicles to be able to deliver pizza in less than 30-minutes as well as drone-based systems for pizza delivery and megastores like Walmart are also investing in automated delivery technology and plan to go head to head with Amazon to provide fast and reliable deliveries without human intervention. The idea to create autonomous vehicles stems back to 1925, when Francis Houdina demonstrated a radio-controlled car through the streets of Manhattan. This idea continued to grow and in 1979 when The Stanford Cart used video processing to navigate a cluttered room without human input and in 1995 The VaMP autonomous vehicle drove about two thousand kilometers autonomously. By 2004 the DARPA Grand Challenge incentivized autonomous car development by offering $1 million to the team whose robotic vehicle could successfully navigate an obstacle course. No competitor was successful in completing the inaugural challenge; however, in 2005 five vehicles completed the race for a first prize of $2 million. The first-place winner of these successful competitors was Stanley, a vehicle designed by Stanford University and Volkswagen, using technology adapted from the Stanford Cart.

Since its inception, the idea to create autonomous vehicles is ongoing, avid, and ambitious. Remote controlled car competitions are popular projects that dwell on this idea of full automation, which allows to develop, work and create new technology.

3.1 Existing Similar Projects and Products

One of the most important technologies for self-driving cars is the Advanced Driver Assistance Systems (ADAS). It has become one of the fastest growing automotive electronic segments due to the desire to reduce vehicular accidents and fatalities. An ADAS is a combination of electric systems that are focused on automating and improving the vehicle’s systems to ensure safer driving. Conventional ADAS technology can detect some objects, do basic classification, alert the driver of hazardous road conditions and, in some cases, slow or stop the vehicle. This level of ADAS is great for applications like blind spot monitoring, lane-keep assistance and forward collision warning. These systems may also be used to judge the fatigue and distraction of the human driver and thus make precautionary alerts or to assess the driving performance and make suggestions regarding the same. These systems can take over the control from the human on assessing any threat, perform easy tasks (like cruise control) or difficult maneuvers (like overtaking and parking). The greatest advantage of using the assistance systems is that they enable communication between different vehicles, vehicle infrastructure systems and transportation management centers. This enables exchange of information for better vision, localization, planning and decision making of the vehicles. The evolution of ADAS sensor systems for driverless vehicles is changing the way radar systems are implemented. Moving from the simpler collision avoidance or adaptive cruise control to all round detection is presenting a significant challenge.

In order to get a fair comparison for the preexisting projects, four different projects are considered: a DIY car using an Arduino and basic sensor components, another project that uses a raspberry pi and a camera, a monstrous vehicle from MIT using a stacked-sensor approach, and the UCF F1/10 (UCF1/10) project that is already being developed and will eventually merge with our future design. Juxtaposing these projects present dichotomies of design, cost, simplicity, and success which will help influence the decisions and direction that the project takes as it is being developed.

3.1.1 Autonomous Do-It-Yourself (ADIY)

Considering the Autonomous Do-It-Yourself (ADIY) design shown in Figure 2, which will be viewed as the lower end of the cost spectrum, also being simplest, presents a lower bound for our project for cost and performance. The ADIY utilizes five separate ultrasonic sensors and an Arduino Uno. The creator of the ADIY design also added an extra feature, a button that operates as an override switch, but the button does not add any significant enhancements to the functionality of the project.

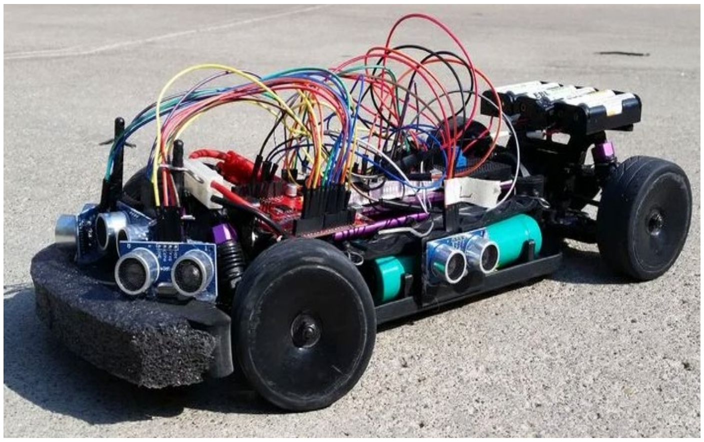


Figure 2 - The ADIY Project Final Design located at https://www.instructables.com/id/Autonomous-RC-Car/

The microcontroller used in the ADIY design is the Arduino Uno which was selected due to its relative ease of use and programming as well as its large amount of IO pins and available libraries. The sensor of choice was the HC-SR04 ultrasonic sensors which are relatively cheap, easy to use, and can provide distance measurements up to 400 cm. The ADIY design does not feature any secondary sensors but makes up for this by creating a 180° vision arc with strategic sensor placement. These sensors also provide simple measurement as the user only has to measure the time between pulses on the output pin.

3.1.2 Donkey Self-Racing Car

The next step up in complexity for existing projects is the Donkey Self Racing Car. Donkey is a high-level self-driving vehicle built on a library written in Python. It was developed with a focus on enabling fast experimentation and easy contribution. Donkey is the standard car that most dilettantes build first. The parts cost about $250 to $300 and takes a few hours to assemble. The Donkey Car has an onboard Raspberry Pi, a camera, and a servo controller unit that interfaces the Raspberry Pi to the actual servos and motors in the RC car. The project is shown below in Figure 3.

Figure 3 - Donkey Car Model found at https://makezine.com/projects/build-autonomous-rc-car-raspberry-pi/

The Donkey car represents a step up from the constant sensor-sampling scheme that they ADIY model utilized but is more computer science intensive. For the first drive-through of a track, the user must manually drive a lap, after which the Raspberry Pi collects and sends camera data to an Amazon server, which analyzes the frames and returns servo commands. Once the Raspberry Pi has all of its servo commands for a lap, it utilizes those commands to drive itself around and compete in the race.

3.1.3 F1/10 Competition Car

One of the most popular competitions for autonomous RC car racing is the F1/10 competition. The competition focuses on creating, designing, building and testing an autonomous 1/10th scale F1 race car, that is capable of speeds of 40 MPH. In this competition the use of LIght Detection and Ranging (LIDAR) is quite popular. LIDAR provides a 270-degree view of the surrounding area, which helps vehicles avoid collisions by detecting the obstruction ahead. The LIDAR works in accordance with other sensors, such as Inertial Measurement Units (IMUs), rotary encoders, and any other various sensors that the team deems necessary to operate the vehicle. Figure 4 below shows the prototype of the F1/10 vehicle with the LIDAR mounted on the front bumper.

Figure 4 - F1/10 Prototype from UCF Project utilizing LIDAR found at http://www.ece.ucf.edu/~zsguo/files/UnderGradPosition.pdf

The brain of UCF's Robotics Team F1/10 car (UCF1/10) is the Jetson Tx2, which is chosen for its incredibly powerful CPU, GPU, and accelerators. The Tx2 is capable of running a full operating system—in this case Linux—which allows the use team the ability to use ROS to monitor the sensors and then control the vehicle.

In addition to all of the aforementioned sensors and components, the UCF1/10 team uses the provided RF remote controller to operate as a switch for the programmed functions. When the switch is activated, the ROS protocols will run, and then the switch can be deactivated, and the vehicle will come to a full stop and will cease running.

3.1.4 MIT Rapid Autonomous Complex-Environment Competing Ackermann-steering Robot (RACECAR)

The final project comparison is to an MIT course-led project, called the “**R**apid **A**utonomous **C**omplex-**E**nvironment **C**ompeting **A**ckermann-steering **R**obot, or **RACECAR”** (jetsonhacks.com) shown in Figure 5. The RACECAR features just about every sensor that can be used in autonomous racing, including a LIDAR, a camera, an opto-isolator board, an IMU, and a visual odometer. In addition to all of the sensors housed on the RACECAR, the Jetson Tx2 is also used as the microcontroller for the project.

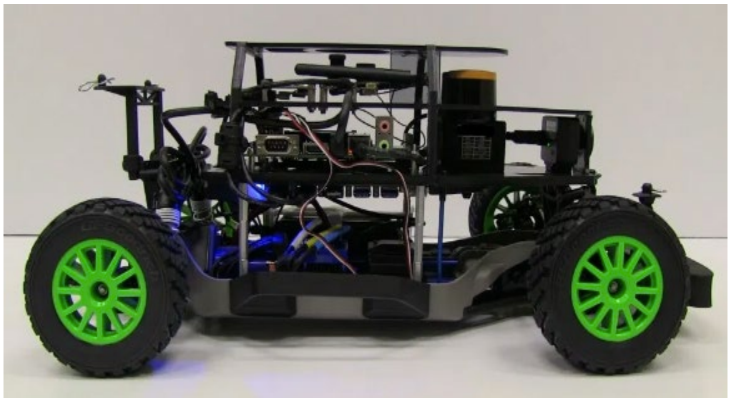


Figure 5 - MIT RACECAR Model, found at: https://www.jetsonhacks.com/2015/10/06/mit-racecar-walkthrough-nvidia-jetson-tk1/

This Mortal Engines-esque project has several custom 3D-printed pieces and required additional housing in order to accommodate all of the parts. Seeing as how the project is more involved than any of the other projects, and also requires special housing, the RACECAR provides the ideal case for our project if time and resources were not inhibitive.

While all these projects were able to build cars that can react to the environment around them, each had their own drawbacks. The ADIY model simply went in a straight line until the sensors gave a proximity alert and the Arduino would adjust the course. Even with this technique the car still crashed into a wall that was directly in front of the car. The Donkey car and most likely the RACECAR required course input so that the cars would have an idea of where they were going, which is assistive in racing scenarios but is unrealistic in the grand scheme of autonomous vehicles if every course had to be preplanned. The UCF1/10 project is closer to full autonomy as the LIDAR readings provide a wide range of measurements, and thus the Tx2 will be able to make an educated decision based on those readings and the other sensors. For our project, we will be taking all of these projects into consideration as we combine all of the pros and cons into our design.

Starting to take these projects into consideration for our design, another option that can be used in the F1/10 competition is the ZED 3D camera. The ZED is a passive stereo camera that reproduces the way human vision works. Using its two “eyes”, the ZED creates a three-dimensional interpretation of the scene by comparing the displacement of pixels between the left and right images. The ZED Stereo Camera is the first sensor to introduce indoor and outdoor long-range depth perception along with 3D motion tracking capabilities, enabling new applications in many industries: AR/VR, drones, robotics, retail, visual effects and more. For this project the ZED camera will be the main sensor that will help navigate the vehicle.

IEEE developed the “Autonomous RC-Car for Education Purpose in iSTEM Projects” which uses a scaled RC-Car platform with several sensors and Raspberry Pi embedded platform, to build an autonomous driving car in scaled indoor simulation environment. The RC-Car is capable of safely autonomous driving. Many existing algorithms are put together to provide the necessary functions of autonomous driving, such lane detection, obstacle detection, lane following, vehicle control etc. A pre-built four-wheel drive (4WD) chassis is used as a base on which following hardware components are fit: Raspberry Pi (rev C) for GPU and CPU computations, Wi-Fi 802.11n dongle to connect to Pi remotely, Motor driver IC L293D which can control two motors, Ultrasonic sensor to detect obstacles, Pi camera, batteries to provide power. The Raspberry Pi is a credit card-sized single-board computer.

3.2 Relevant Technologies

In the course of researching information surrounding the technology associated with autonomous vehicles, multiple interesting methods of converting existing commercial-off-the-shelf components designed to perform a function not necessarily related to autonomous vehicles into a component that could easily be converted for use in a functioning, semi-intelligent autonomous vehicle capable of autonomous course navigation and obstacle avoidance.

3.2.1 Vehicle Platform

The core technology for any autonomous vehicle is the vehicle chassis that will be converted from an operator input vehicle to an autonomous machine-controlled convoy. Any vehicle chassis to be considered must meet the engineering specifications outlined in section 2.3 Requirements Specifications as well as section 4.2 Realistic Design Constraints. As such, a smaller scale replica of a human-operated convoy was researched in order to replicate real-world conditions that our technology would influence, but at a smaller, more manageable and cost-effective scale. Our research migrated toward the remote-control vehicle enthusiast industry.

Upon performing research into the diverse remote-control vehicle industry, many trends were discovered. First, the vehicles were classified according to their size compared to a full-size vehicle. Available vehicle chassis can range from 1/16th scale vehicle up to 1/5th scale vehicle. By far the most versatile and popular size vehicle chassis was the 1/10th scale vehicle. This platform is capable of carrying the required weight while maintaining a smaller footprint for testing purposes in a limited area.

Second, vehicles were classified according to their propulsion method. Many remote-control vehicle manufacturers utilize electric motors to drive their vehicles, but gasoline and RC glow powered vehicles are also available. Gasoline motors in remote-control vehicles are generally integrated in the larger scale models, usually in the 1/8th to 1/5th scale due to the weight requirements of the heavier engine components, fuel storage requirements for a prolonged run time as well as the power to weight ratio output of the engine. These types of motors are generally two-stroke internal combustion engines that require the addition of synthetic oil to the gasoline in order to lubricate and cool the motor components during operation. RC glow engines use nitro fuel, a methanol-based fuel with nitromethane and oil added. The amount of nitromethane in the fuel is typically about 20% but could be anywhere in the 10% to 40% range or higher. Castor oil or synthetic oil is added to the fuel to provide lubrication and cooling. Models utilizing this type of propulsion are usually in the 1/10th scale range for models due to the power to weight ratio output by the motor and the run time associated with the fuel capacity. Due to the constraint of testing and demonstrating the autonomous vehicle in an enclosed space, the gasoline and nitro powered vehicles would not be optimal for our autonomous vehicle platform as the exhaust fumes would be prohibitive, toxic and potentially harmful within an enclosed space.

Third, vehicles are classified as to what terrain they are built to operate upon. There are on-road and off-road versions of vehicle models. There is also a subsection for rock crawling models as well as hybrids that are marginally effective at tackling both on-road and off-road terrain. This type of vehicle is equivalent to human sized rally vehicles utilized in races throughout Europe. This type of vehicle would be optimal for the purposes of an autonomous vehicle as the terrain conditions will fluctuate from testing to actual locality of utilization.

Lastly, vehicles were classified according to their drivetrain. Vehicle models were classified as either two-wheel drive or four-wheel drive versions. The four-wheel drive versions were typically designed for off-road or hybrid applications, though some on-road versions included four-wheel drive to allow better traction for racing applications. Two-wheel drive models were typically relegated to on-road uses or as general utilization over multiple roles. A typical two-wheel model would be utilized for backyard tracks or marginally uneven ground.

3.2.2 GPU Image Processing

The Graphics Processing Unit (GPU) is not only a powerful graphics engine but also a highly parallel programmable processor featuring peak arithmetic and memory bandwidth that substantially exceeds a dedicated Central Processing Unit (CPU). Graphics Processing Units can be utilized to apply texturing and pixel engines that were originally designed for 3-dimensional modeling and rendering, to many classic image-processing problems to provide speed increases over CPU-only implementations, without comprising image quality.

Compute Unified Device Architecture (CUDA) is a general architecture for parallel computing introduced by NVIDIA in November 2007. It includes a new programming model, architecture and instruction set oriented towards parallel computing. This allows pixels to be treated in parallel. In the CUDA programming framework, the GPU is viewed as a compute device that is a coprocessor to the CPU.

The GPU has its own DRAM, referred to as device memory, and executes a very high number of threads in parallel. More precisely, data-parallel portions of an application are executed on the device as kernels which run in parallel on many threads. In order to organize threads running in parallel on the GPU, CUDA organizes them into logical blocks. Each block is mapped onto a multiprocessor in the GPU. All the threads in one block can be synchronized together and communicate with each other. Because there is a limited number of threads that a block can contain, these blocks are further organized into grids allowing for a larger number of threads to run concurrently. CUDA also supports the use of memory pointers, which enables random memory-read and write-access ability. In addition, the CUDA framework provides a controllable memory hierarchy which allows the program to access the cache (shared memory) between GPU processing cores and GPU global memory. [7]

As shown in Figure 6 below, the input images are transferred from the CPU to the GPU where threads are allocated and CUDA parallel processing is performed on the input image before being sent back to the CPU as an output image.

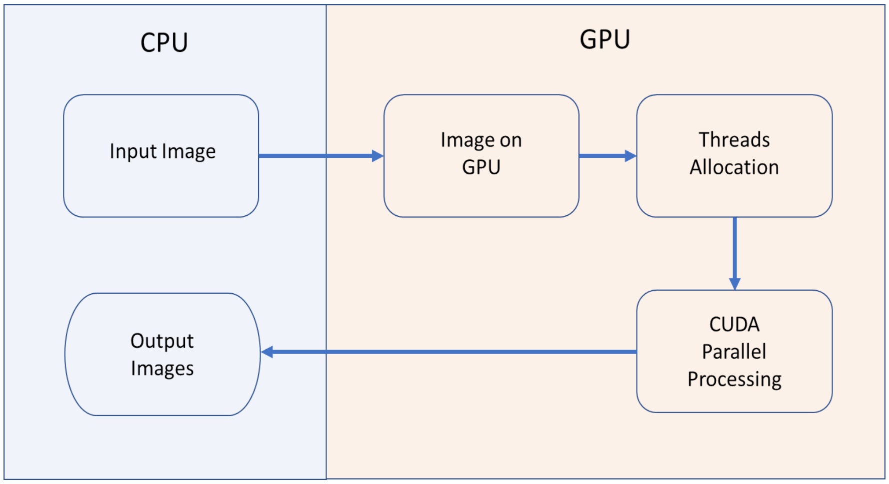


Figure 6 - GPU Image Processing Block Diagram

3.2.3 Real-Time Operating Systems (RTOS)

One of the potentially misleading ideas revolving around real-time operating systems is that real-time does not necessarily indicate “faster”. Real-time operating systems deals in scheduling processes such that no critical deadline is ever missed. Real-time operating systems serve applications that process data as it is received, sometime with minimal buffer delay. This technology will be vital to processing image data received from the ZED stereo camera in order to process images to facilitate object avoidance and course navigation in near time. From a competitive viewpoint, fast image processing will be required in order to navigate the course in a timely fashion while avoiding obstacles and fellow racers, and an advanced scheduling algorithm will be instrumental in achieving those goals.

3.2.4 Stereo Vision Systems

Human beings acquire information about the location and other properties of objects within an environment thanks to a powerful and sophisticated vision system. The perception of a third dimension (depth) occurs due to the difference between images formed in the retinas of the left and right eyes. In the process of image formation, the catches of each eye are not equal because they present a slight variation in the position of the observed objects, attributed to the separation between the eyes. Artificial stereo vision systems are generally inspired by the biological process to extract three-dimensional information from digital images, which can be used to perform 3D reconstructions, tracking, and detection of objects.

There are several devices that provide three-dimensional information, depending on the operating technology they can be classified into stereo vision sensors, structured light devices or sensors based on the principle of Time of Flight (ToF). These devices are used in several areas with multiple purposes, in Robotics they are employed as essential tools in navigation applications, three-dimensional parts review, among others.

However, depth data provided by stereo devices have errors attributed to several aspects related to cameras hardware and computational processes that are performed to obtain these values. It is possible to enumerate some sources of errors as hardware system error, camera calibration error, feature extraction and stereo matching errors. These inherent errors that such data present should be considered in the applications where depth data generated by 3D vision sensors are used, such an example is the Robotic Vision. In real applications, such as autonomous robotics, it is important to consider and treat those visual errors in order to achieve correct decision-making process during a navigation task, for example.

As said, humans can have a three-dimensional perception of the world through the eyes due to the difference observed in the images formed in left and right retinas. In the imaging process, the images sent to the brain from each eye are not the same, with a slight difference in the position of the objects due to the separation between the eyes, which form a triangle with the scene points. Thanks to this difference, by triangulation the brain can determine the distance (depth) that the objects are in relation to the observer position. The implementation of stereo vision in computers uses this basic principle to recreate a 3D scene representation based on the two images of it taken from different viewing points. This is known as stereo reconstruction. In order to do stereo reconstruction, a series of steps are necessary, as calibration, rectification, and further depth determination.

The calibration process estimates intrinsic and extrinsic parameters of the cameras. Intrinsic values include the focal length, principal point coordinates, radial and tangential distortion factors. They are commonly used to obtain images without distortions, caused by the lenses and camera construction process, and to obtain three-dimensional representations of a scene. On the other hand, extrinsic parameters relate the real-world reference systems and the camera, describing position and orientation of the device in the real-world coordinate system (i.e. rotation matrix and translation vector). In addition to the calibration (for each camera), may be developed a stereo calibration, this process allows obtaining information that relates the positions of the two cameras in space.

Stereo rectification is the process in which a pair of stereo images are corrected, so that, it appears that they had been taken by two cameras with row-aligned image planes as shown in Figure 2. With such process the principal rays of the cameras are parallel, that is, they intersect at infinity. This step facilitates the stereo disparity estimation, a fundamental process prior to the estimation of the depth map.

The stereo camera computes depth information using triangulation (re-projection) from the geometric model of non-distorted rectified cameras. Assuming the two cameras are co-planar with parallel optical axes and same focal length, the depth of each point is calculated. In this calculation, depth varies inversely proportional to the disparity between baseline distance and image distance. A depiction of a basic stereo vision system triangulation is shown below in Figure 7.

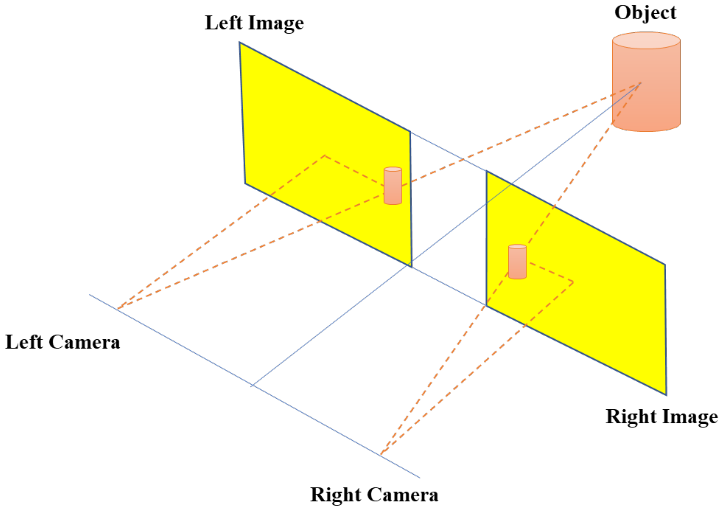


Figure 7 - Basic Stereo Vision Object Acquisition and Comparison

Stereo vision cameras acquire 3D values considering the left camera as the origin frame. Given the coordinates of a pixel in this image coordinate frame, these same coordinates are used to search their corresponding depth value Z in the depth map.

3.2.5 Radar Sensors

One of the leading technologies that is being used in the automobile industry is the radar-based safety system. Radar is being used for blind spot detection, automatic emergency braking, pedestrian automatic emergency braking and forward collision. Cameras and radar are now being used in the Advanced Driver Assistance Systems (ADAS) to provide lane-departure warnings and adaptive cruise control that allows the vehicle to follow the vehicle in front. As more ADAS systems become more advanced, they are expected to become government-mandated in the future following the recent introduction of legislation such as rearview video systems in vehicles and advanced emergency brake assist (AEB) for commercial vehicles.

Today, many automobile industries are now featuring rearview cameras and AEB systems to promote safe driving. Radar-based systems are used because they are based off radio waves which are not affected by temperature, humidity and is not affected by light. Radar can also be created to cover vast distances, depending on the power of the transmitter and the antenna size. They can easily differentiate between a moving and stationary object and can accurately measure the velocity, distance and exact position of any object. Radar typically consists of a transmitter, waveguides, antenna, receiver and a processing unit. An automotive radar usually works in the range of 77GHz and is divided into, short range – 0.5 to 20 meters, medium range – 1 to 60 meters and long range – 10 to 250 meters. The most acceptable radar technology in the automotive market is the Frequency Modulated Continuous Wave (FMCW) radar In FMCW radar, the transmitter and the receiver operate continuously and generate a carrier wave known as “chirp”. This chirp is transmitted by the “transmitter” and received back by the “receiver” after reflecting from the target object and allow radar to calculate its range, velocity, and position.

Many automotive radar systems use a pulse-Doppler approach, where the transmitter operates for a short period, known as the pulse repetition interval, then the system switches to receive mode until the next transmit pulse. As the radar returns, the reflections are processed coherently to extract range and relative motion of detected objects.

The BGT24M 24GHz radar sensor from Infineon Technologies, for example, can be used with an external microcontroller in an electronic control unit (ECU) to modify the throttle to maintain a constant distance from the car in front with a range of up to 20 meters.

Another approach is to use continuous wave frequency modulation (CWFM). This uses a continuous carrier frequency that varies over time with a receiver on constantly. To prevent the transmit signal from leaking into the receiver, separate transmit and receive antennas have to be used.

The BGT24MTR12 is a Silicon Germanium (SiGe) sensor for signal generation and reception, operating from 24.0 to 24.25 GHz. It uses a 24 GHz fundamental voltage-controlled oscillator and includes a switchable frequency prescaler with output frequencies of 1.5 GHz and 23 kHz. An RC polyphase filter (PPF) is used for LO quadrature phase generation for the down conversion mixer, while an output power sensor as well as a temperature sensor are integrated into the device for monitoring. The device is controlled via SPI and is manufactured in a 0.18 µm SiGe:C technology offering a cutoff frequency of 200 GHz and packaged in a 32-pin leadless VQFN package. However, the architecture is changing for driverless vehicles where in lieu of a local ECU, the data from the various radar systems around the vehicle are fed into a central high-performance controller that combines the signals with those coming from cameras and perhaps from Lidar laser sensors. The controller can be a high-performance general-purpose processor with graphic control units (GCUs), or a field programmable gate array where the signal processing can be handled by dedicated hardware. This places more emphasis on the analog front-end (AFE) interface devices that have to handle higher data rates and more data sources.

The type of radar sensor being used is also changing. 77 GHz sensors are providing both longer range and higher resolution. A 77 GHz or 79 GHz radar sensor can be adapted in real time to provide long-range sensing of up to 200 m in a 10° arc, for example for detecting other vehicles, but it can also be used in a wider, 30° sensing arc with a lower range of 30 m. The higher frequency gives higher resolution that allows the radar sensor system to distinguish between multiple objects in real time, for example detecting a number of pedestrians within that 30° arc, giving the controller for the driverless vehicle more time and more data.

The 77 GHz sensors use silicon germanium bipolar transistors with a 300 GHz oscillation frequency. This allows one radar sensor to be used for multiple safety systems such as headway alert, collision warning and automatic braking, and the 77 GHz technology is also more resistant to the vibration in the vehicle so that less filtering is required. The sensors are used to detect the range, speed and azimuth of a target vehicle in the vehicle coordinate system (VCS). The accuracy of the data depends on precise alignment of the radar sensor. A radar sensor alignment algorithm executes more than 40 Hz while the vehicle is running. In 1 ms it must calculate the misalignment angle based on data provided by the radar sensor, as well as the vehicle speed, the sensor's position on the vehicle, and its pointing angle. Software tools can be used to analyze recorded sensor data captured from road testing a real vehicle. This test data can be used to develop a radar sensor alignment algorithm that calculates sensor misalignment angles from raw radar detection and host vehicle speed using a least squares algorithm. This also estimates the computed angle’s accuracy based on the residual of the least-squares solution.

An analog front end such as the AFE5401-Q1 from Texas Instruments can be used to link the radar sensor to the rest of the automotive system. The AFE5401 is designed for the next generation of automotive radar applications where space constraints and increasing radar performance are driving a need for greater bandwidth, high integration and low power. Delivering twice the sampling rate and bandwidth spectrum of existing solutions, the AFE5401-Q1 enables quick position and speed discrimination of even the fastest moving scenes. It also requires 30 percent less power and 20 percent less board space enabling a small device for optimal in-car installation Four separate channels are simultaneously monitored by the device to determine the exact direction of the incoming radar signal. This allows the automotive radar system to make smart decisions about where an object is located, if it is moving and how soon a response needs to occur.

For a low-cost system, the AD8284 from Analog Devices provides an analog front end with a four-channel differential multiplexer (mux) that feeds a single channel low noise preamplifier (LNA) with a programmable gain amplifier (PGA) and an antialiasing filter (AAF). This also uses a single direct-to-ADC channel, all integrated with a single, 12-bit analog-to-digital converter (ADC). The AD8284 also incorporates a saturation detection circuit for high frequency overvoltage conditions that would otherwise be filtered by the AAF. The analog channel features a gain range of 17 dB to 35 dB in 6 dB increments, and an ADC with a conversion rate of up to 60 MSPS. The combined input referred voltage noise of the entire channel is 3.5 nV/√Hz at maximum gain. The output of the AFE is fed into a processor or an FPGA such as the IGLOO2 or Fusion from Microsemi or Cyclone IV from Intel. This can implement the 2D FFT in hardware using the FPGA design tools to handle the FFT and provide the required data on surrounding objects. This can then be fed into a central controller.

A critical challenge for the FPGA is the detection of multiple objects, which is more complicated for CWFM architectures than pulse-Doppler. One way is to vary the duration and frequency of the ramps and evaluating how the detected frequencies move in the spectrum with different steepness of frequency ramps. As the ramp can be varied at 1 ms intervals, hundreds of variations can be analyzed per second. The data fusion from other sensors can also help, as camera data can be used to discriminate between stronger returns from vehicles compared to weaker returns from people, and what sort of Doppler offset to expect.

Another option is multimode radar that uses CWFM to find targets at longer range on the highway, and short-range pulse-Doppler radar for urban areas where pedestrians are more likely to be detected. Radar is a very popular sensing technique that has become well established with automotive manufacturers and so is a leading technology for this approach. Bringing together higher frequency 77 GHz sensors with multi-mode CWFM and pulse-Doppler architectures, along with data from other sensors such as cameras is also presenting a significant challenge for the processing sub-systems. Solving these challenges in a safe, consistent and cost-effective way is essential to the continuing development of autonomous vehicles.

Collision avoidance -the detection of objects for collision avoidance is a key part of the safety requirements for self-navigation vehicles. While a multitude of CCD image sensors and cameras are readily available from a variety of sources, the ability to develop sensor algorithms quickly and effectively means that engineers need flexible platforms between the video sensor and the controlling processor. An ideal solution is to use FPGA technology that directly supports video streaming to detect edges, enhance images, and perform calculations in fast hardware to determine the speed, direction and proximity of approaching objects, and to perform threat assessment.

One ideal solution comes from Lattice Semiconductor with its LFE3-70EA-HDR60-DKN development system for 1080p, 60 fps video cameras. This platform includes a reference design and IP for use with the company’s LCMXO2-4000HE-DSIB-EVN image interface board, and its LF-9MT024NV-EVN Nanovesta camera headboard. This technology allows two image sensors to be merged into one video data stream (Figure 2) permitting depth perception and more accurate speed and position sensing as well as providing auto-white balance, 2D-noise reduction, and what is claimed to be the industry's fastest auto-exposure with support for up to 16 Megapixel resolution.

Freescale is the market leader in embedded radar solutions and has a highly skilled design team driving new innovations in embedded MCUs and 77 GHz millimeter wave integrated circuits specifically tailored for radar applications. Freescale introduced the next-generation of embedded radar-based products with the Qorivva MPC577xK MCU and the MR2001 77 GHz radar transceiver chipset. The radar transceiver chipset consists of a VCO (MR2001VC), a two-channel Tx transmitter (MR2001TX), and a three-channel Rx receiver (MR2001RX). These new products deliver a complete embedded radar system for automotive designs. These state-of-the-art radar solutions complement Freescale’s existing arsenal of high-performance ADAS solutions. For example, Freescale’s embedded camera/video offerings—such as the SCP220x image cognition processor (ICP) family and the Qorivva MPC5604E MCU—are designed for advanced vision processing to address object detection from vehicle to pedestrians. With the release of the new radar-based products, Freescale now offers a comprehensive ADAS solution for automotive manufacturers’ requirements.

Many current radar systems are based around the Qorivva MPC5675K MCU, including external FPGA, ADC, DAC, SRAM, and analog front-end for transmitting and receiving. The MPC577xK MCU is specially designed to enable a cost-effective radar system with increased performance in a single chip solution. The MPC577xK family attains its cost-advantage by providing high levels of digital and analog integration within a single 356 BGA package and removes the requirement of having an external FPGA, ADC, DAC and SRAM, thereby reducing the number of components required, the PCB size, and the complexity of software. In addition to the two 266 MHz e200z7 processing cores, the MPC577xK MCU also features a state-of-the-art signal processing toolbox (SPT). This contains the hardware modules required for processing sampled signals from short, medium and long-range radar applications. The SPT is a powerful processing engine containing high-performance signal processing operations driven by a specific instruction set. Its programmability ensures flexibility while removing the CPU from frequent scheduling of hardware operations, while still controlling and interacting with the processing flow.

- Highly integrated MCU reduces the total number of components required, the size of the PCB, and the complexity of the software
- Easy-to-use MCU with integrated FFT accelerator
- Large-density memories to support scalable radar applications
- Supports open-loop and phase-locked looped systems, enabling design flexibility
- Contained power dissipation improves efficiency
- Helps system manufacturers meet the functional safety ISO 26262 ASIL-D target

The MR2001 77 GHz radar transceiver chipset is an expandable, high-performance, three-package solution for automotive radar modules. The chipset consists of a VCO, a two- channel Tx transmitter, and a three-channel Rx receiver. These three parts are each packaged in a 6 mm x 6 mm fan-out wafer-level package on a 500 µm pitch.

This package technology is ideal for 77 GHz radar applications since it provides extremely low insertion loss and parasitic at frequencies up to 100 GHz. The packaged chipset simplifies the end user’s assembly of the radar module since there is no need for chip and wire assembly techniques for bare die. Furthermore, the chipset readily scales up to four Tx channels and 12 Rx channels, enabling a single radar platform capable of electronic beam steering across a wide field-of-view and supporting long-, mid-, and short-range radar applications over a full selection of vehicles—from budget to luxury.

- Scalable to two Tx channels and 12 Rx channels with simultaneous active channels to enable a single platform capable of electronic beam steering across a wide field-of-view
- Advanced packaging technology to ensure the highest performance and minimum signal interference on the customer PCB
- Low power consumption—2.5 W typical for the complete transceiver
- Supports fast modulation at 100 MHz/100ns
- Best phase noise performance <-93="" dbc/Hz="" at="" 1="" mHz,="" offset="" to="" improve="" target="">
- Integrated baseband filter and VGA saves system BOM cost

- Local oscillator at 38 GHz to lower the distribution loss and reduce system interference
- Bi-phase modulator on the transmitter chip supports rejection of parasitic signals

3.2.6 Ultrasonic Proximity Sensing

Ultrasonic proximity sensors are a common type of proximity sensor that works by emitting sound frequencies higher than the audible range of human hearing. The basic principle behind this type of sensor is that the sensor emits an ultrasonic pulse and receives it back. The time difference between transmission and reception is used to determine the distance traveled. Since the ultrasonic pulse will bounce off of an object, the distance travelled will indicate the distance to the object. Since ultrasonic proximity sensors utilize sound instead of light, they can be used where photoelectric sensors have difficulty, such as in strong sunlight. This type of sensor is also immune to common contaminants such as dust and moisture. This type of sensor would be susceptible to noise interference from any similar devices emitting pulses with the same sound frequency and potentially provide false readings to the microcontroller. This may be detrimental during a competition where multiple vehicles may be operating with similar sensors and their frequencies emissions may interact unfavorably.

3.2.7 Infrared Distance Finding

Infrared distance finders are inexpensive, relatively accurate at short distances (within 30 inches) and are fairly simple to incorporate into a project. They process distances via triangulation. The infrared range finder emits a pulse of infrared light which is reflected by the target. An enclosed CCD array receives the reflected light and determines the angle that the light is received. A corresponding value is then transmitted to a microcontroller. The output from an infrared range finder is non-linear due to the fact that the measured distance may increase or decrease linearly.

One major flaw in infrared range finders is when an object is below the minimum range for the sensor to detect it. Basically, if an object is too close to the sensor, then the light is reflected too quickly for the enclosed CCD array to read it and an inaccurate range reading is provided to the microcontroller. This may result in the sensor providing data to the microcontroller that would indicate an object is much farther away than it actually is and result in a collision.

3.2.8 Servo Motors

Servo motors have been available for a long time and are utilized in many applications. They are small in size but can carry a huge workload and are very energy efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services.

To fully understand how the servo works, you need to take a look at the intricacies within the design. Inside there is a pretty simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between its actual position and desired position. So, if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand.

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counterclockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position. The PWM signal effect on a servo motor is depicted bellow in Figure 8.

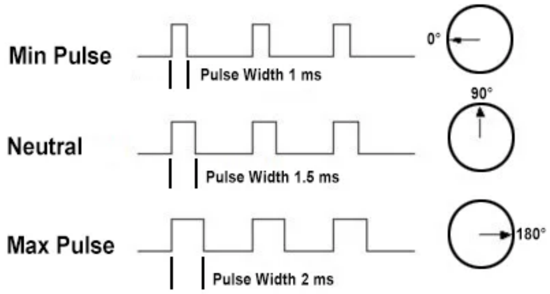


Figure 8 - PWM Signal effect on Servo Motor

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

There are two types of servo motors - AC and DC. AC servo can handle higher current surges and tend to be used in industrial machinery. DC servos are not designed for high current surges and are usually better suited for smaller applications. DC motors are less expensive than their AC counterparts. These are also servo motors that have been built specifically for continuous rotation, making it an easy way to get your robot moving. They feature two ball bearings on the output shaft for reduced friction and easy access to the rest-point adjustment potentiometer.

Servos are used in radio-controlled airplanes to position control surfaces like elevators, rudders, walking a robot, or operating grippers. Servo motors are small, have built-in control circuitry and have good power for their size. In food services and pharmaceuticals, the tools are designed to be used in harsher environments, where the potential for corrosion is high due to being washed at high pressures and temperatures repeatedly to maintain strict hygiene standards. Servos are also used in in-line manufacturing, where high repetition yet precise work is necessary.

3.2.9 PID Controller

The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output. The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The proportional gain (Kc) determines the ratio of output response to the error signal and the integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and set point. The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative time (Td) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (Td), because the Derivative Response is highly sensitive to noise in the process variable signal.

The process of setting the optimal gains for Proportional, Integral and Derivative to get an ideal response from a control system is called tuning. The gains of a PID controller can be obtained by trial and error method. In this method, the I and D terms are set to zero first and the proportional gain is increased until the output of the loop oscillates. As one increases the proportional gain, the system becomes faster, but care must be taken not make the system unstable. Once P has been set to obtain a desired fast response, the integral term is increased to stop the oscillations. The integral term reduces the steady state error but increases overshoot. Some amount of overshoot is always necessary for a fast system so that it could respond to changes immediately. The integral term is tweaked to achieve a minimal steady state error. Once the P and I have been set to get the desired fast control system with minimal steady state error, the derivative term is increased until the loop is acceptably quick to its set point. Increasing derivative term decreases overshoot and yields higher gain with stability but would cause the system to be highly sensitive to noise. With the use of MATLAB and having knowledge of the Ackermann's Function a PID controller is intended to be design in order to control the speed of the motor based on the inputs provided by the onboard sensors.

3.3 Strategic Components and Parts Selection

Strategic component and parts selection were conducted in order to fulfill engineering specifications, requirements and standards. Selected components are broken down into three subsections: platform, payload and peripherals. The platform will consist of the vehicle chassis and the components required to power, propel and steer the vehicle. The payload will consist of the electronic components required to process all peripheral data and provide motor and steering control signals to the platform. The peripherals will consist of all secondary sensors utilized to facilitate course navigation and collision avoidance.

3.3.1 Vehicle Chassis

Our vehicle chassis was provided by our sponsor, Dr. Guo in accordance with the competition standards outlined in Section 4.1.6. The sponsor-provided vehicle chassis is the Traxxas Ford Fiesta® ST Rally Radio Controlled 1/10th scale car. This specific vehicle chassis would meet our intended goals for autonomous vehicle as well as confirm our research into vehicle platforms. The hybrid of on-road and off-road capability as well as four-wheel drivetrain allows the autonomous vehicle to function over a broader range of terrains. The specifications provided by the manufacturer are summarized in Table 3 below. A picture of the component is highlighted in Figure 9 below.

Table 3 - Vehicle Chassis Specifications

Specification	**Value**
Length:	21.0 in (535mm)
Rear Track:	11.0 in (281mm)
Weight:	97.6 ounces (2.77kg)
Wheelbase:	12.8 in (324mm)
Overall Drive Ratio:	19.69
Gear Pitch:	48
Front Track:	11.0 in (281mm)
Center Ground Clearance:	0.82 in (21mm)
Height (overall):	8.1 in (206mm)
Drive System:	Shaft-Driven, Direct Drive 4WD

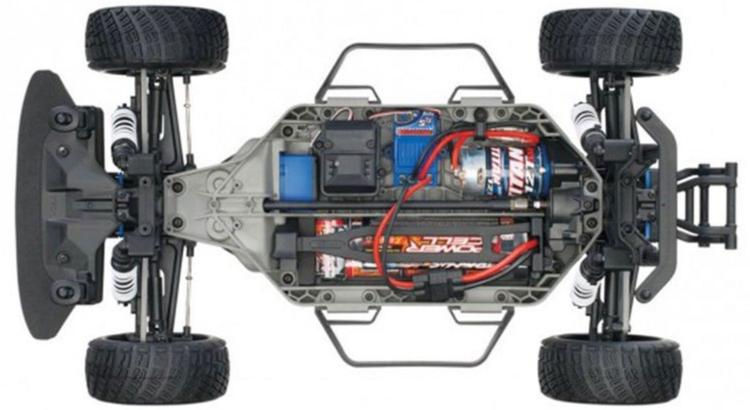


Figure 9 - Traxxas Ford Fiesta® ST Rally RC 1/10th Scale Vehicle

The specific standards and specifications that this vehicle chassis satisfies are summarized below in Table 4.

Table 4 - Vehicle Chassis Selection Matrix

Requirement Standard	Chosen Component Value	Satisfies
1/10 Scale rally car chassis	1/10 Scale	Competition Standard
Equivalent to Traxxas Model 74054	Model 74054	Competition Standard
Four-wheel or Two-wheel drive	Four-wheel drive	Competition Standard
Stock tires or equivalent	Stock tires	Competition Standard

3.3.2 Power System

The power management PCB will provide regulated DC power to all subsystems. The power system's input will be a rechargeable lithium-polymer (LiPo) battery pack operating between 11.1 V_{DC}, providing current of 5000 mAh. The required voltages for the various components of our system are 1.8V, 3.3V and 5V. In order to provide these regulated voltages, voltage regulators will be required to drop the input down to usable voltages for system components. Two types of voltage regulators can be used to accomplish this: linear and switching voltage regulators.

When selecting which type of voltage regulator to utilize, we determined that the ability to take a wide range of input voltages was vital as well as the ability to maintain high efficiency through only consuming input voltage as required. As such, we determined that a buck voltage regulator would be more suitable to our requirements. An output synchronous buck converter was chosen to facilitate the needed three stepped down voltages required by project's system components. The chosen buck voltage regulator is showcased below.

3.3.2.1 Texas Instruments TPS56339 Output Synchronous Buck Converter

The Texas Instruments TPS56339 is a 4.5V to 24 V input range, 3 A synchronous output buck convertor. The TPS56339 was chosen due to the requirement of taking a wide range of DC input voltages and converting them to a wide range of stable output voltages that will be utilized by the system. The synchronous output allows the TPS56339 output to be tailored to the needs of the circuit and is a robust convertor. The output of the regulator circuit can be tailored to our needs through the values set for the output resistor network. This allows quick change of the regulator in case of a circuit fault without having to replace an entire power system PCB. The manufacturer's specifications are summarized below in Table 5.

Table 5 - Delta PM05S018A Manufacturer Specifications

Specification	**Value**
Input Voltage:	4.5 to 32 V
Output Voltage:	Tailored to the user's needs via resistor network. 1.8V, 3.3V and 5.0V for our purposes.
Output Current:	3A max
Number of Outputs:	1
Length:	2.90 mm
Width:	1.60 mm

3.3.3 Drive Motor

A drive motor was provided by our sponsor, Dr. Guo, in accordance with the competition standards outlined in Section 4.1.6. The sponsor-provided drive motor is the Titan® 12T 550 (12-Turn) Brushed DC motor and is designed for increased power output with integrated cooling fan which pulls cool air through the motor to keep temperatures down. A cooler running motor allows faster speeds, longer runtime, and extended motor life. The manufacturer specifications are summarized in Table 7 below.

Table 6 - Traxxas Titan 12T Brushed DC Motor Specifications

Turns	12
Voltage	< 11.1 V_{DC}
Current	< 3500mAh

The specific standards and specifications that this drive motor satisfy are summarized below in Table 7.

Table 7 – Drive Motor Selection Matrix

Requirement Standard	Chosen Component Value	Satisfies
Brushless DC motor equivalent to Velineon® 3500 or lower	Velineon® 3500 Brushless Motor	Competition Standard

3.3.4 Motor Controller

To avoid collision or steer away from an obstacle, the speed of the motor needs to be either reduced or increased. To achieve speed control, a motor controller was taken into consideration.

A motor controller was provided by our sponsor, Dr. Guo, in accordance with the competition standards outlined in Section 4.1.6. The sponsor-provided motor controller is the Traxxas XL-5™ Electronic Speed Control. The specifications provided by the manufacturer are summarized below in Table 8.

Table 8 - Motor Controller Specifications

Specification	Value
Input Voltage:	4-7 Cells (4.8 to 11.1 V_{DC})
Case Size:	1.23” W x 2.18” D x 0.61” H
Weight:	2.44 ounces
Motor Limit:	15-turns (540 size) 12-turns (550 size)
On resistance forward:	0.007 Ω
On Resistance Reverse:	0.014 Ω
BEC Voltage:	6.0 VDC
BEC Current:	1A
Power Wire:	14 Gauge / 5”
Input Harness Wire:	26 Gauge / 9”
PWM Frequency:	1600 Hz
Thermal Protection:	Thermal Shutdown

The BEC voltage and current characteristics of a motor controller describe the maximum amount of voltage and current that can be fed directly to the motor via the attached battery pack. In order to feed more voltage or current directly to the motor, for increased performance and speed, a separate BEC circuit could be designed. This could enable the project to have a competitive edge over other racers but may decrease service life of the motor.

3.3.5 Steering Control

A steering controller was provided by our sponsor, Dr. Guo in accordance with the competition standards outlined in Section 4.1.6. The sponsor-provided steering controller is the Dual-Bell crank with Integrated Servo Saver with Traxxas high-torque ball-bearing waterproof 2056 servo motor for armature motion control. The specifications provided by the manufacturer are summarized below in Table 9.

Table 9 - Steering Controller Specifications

Specification	Value
Modulation:	Analog
Torque:	6.0V: 80.00 oz-in
Speed:	6.0V: 0.23 sec/60°
Dimensions:	2.17” W x 0.79” W x 1.69” H
Motor Type:	Brushed
Gear Type:	Plastic
Rotation/Support:	Single Bearing
60°	60°
Pulse Cycle:	2 ms
Pulse Width:	858-1670 µs
Connector Type:	J

3.3.6 MCU

The MCU is the brain of the design and takes the inputs from the sensors, interpret the data and provide output commands to the vehicle to navigate and avoid collisions. The MCU will be constantly taking in data that the GPU feeds it and will have to adjust the motor, speed controller, and steering servos accordingly. Additionally, the MCU will monitor any wireless communications and await a manual override signal—which will be supplied by the user (if necessary) as a failsafe technique—and will then “listen” to the user’s instructions in lieu of making its own.

3.3.6.1Atmel ATmega2560-16AU

This microcontroller, produced by Atmel, is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega2560 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed. As the ATmega2560 is based on Arduino and has readily available coding libraries, the ease of testing and coding makes this particular microprocessor attractive for utilization in the project. The large number of available development packages also facilitate coding and prototyping. The larger number of I/O pins when compared to the other microcontrollers outlined thus far will facilitate the large number of peripheral sensors required to provide object detection and course navigation in the autonomous vehicle. The lower maximum clock frequency versus the other highlighted microcontrollers may be a consideration due to the nature of autonomous vehicles and the need for fast processing to facilitate on the fly motor and steering control signals. Manufacturer specifications are summarized below in Table 10.

Table 10 – ATMEGA2560 Specifications Table

Specification	**Value**
Mounting Style:	SMT
Package/Case:	QFN-32
Core:	ARM Cortex M4
Data Bus Width:	8 bit
Maximum Clock Frequency:	16 MHz
Program Memory Size:	256 kB
Data RAM Size:	8 kB
Number of I/O's:	86 I/O
Operating Supply Voltage:	4.5 to 5.5 V
Interface:	2-Wire, SPI, USART
I/O Voltage:	4.5 - 5.5 V

3.3.7 Image Processor

The image processor we chose is the Jetson TX2, which will take raw data from the 3-D camera and provide data to be processed by the MCU to avoid collisions. Depending on the camera's specifications and capabilities, as well as the direction that the project goes in, the data will either be constant image frames or object distances. The image processor may be utilized to perform calculations on the frames to provide more detailed information about the rates at which objects are approaching and further assist the MCU in determining the most appropriate decision that it will make.

3.3.8 Audible Safety Device

Due to the minimal sound generated by an electric motor and small vehicle platform, an audible safety device was selected in order to alert nearby pedestrians of the presence of the autonomous vehicle. The Adafruit Audio FX Mini Sound Board is an efficient, cost effective means to alert nearby pedestrians of the presence of the autonomous vehicle and is configurable with up to 2MB of storage for various audible alerts recorded in compressed or uncompressed MP3 or WAV format. Manufacturer specifications are summarized below in Table 11. A visual depiction of the sound board is shown below in Figure 10.

Table 11 - Sound Board Specifications	
Specification	**Value**
Input Voltage:	3 to 5.5 V
Physical Dimensions:	1.9” L x 0.85” W
Nonvolatile Storage:	16 MB
Trigger Effect:	11 triggers
Control Communication:	UART

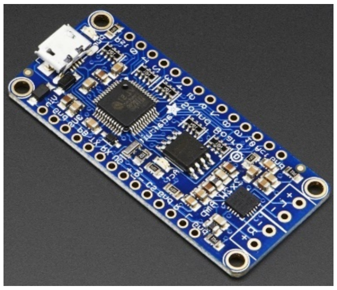


Figure 10 - Adafruit Audio FX Mini Sound Board

3.3.9 Failsafe Device

An electronic failsafe will be designed to mitigate liability associated with the operation of an autonomous vehicle. The failsafe will function in two separate ways. The first operation will be to act as a user-controlled override of the steering and speed functions of the vehicle. The existing remote-control functions provided with the initial vehicle will be integrated into our design, thus allowing an operator to seamlessly take control of the vehicle to avoid injury to individuals or damage to property. This option brings the autonomy level from a 4 down to a level 0 ([1]). The second operation of the failsafe will be as an electronic “kill switch” that immediately disconnects power to the motor, thereby disabling any powered vehicle movement, but still allowing the vehicle to process and steer away from obstacles. This operation is important in case the vehicle travels outside the range of the existing remote-control functionality present in the original vehicle. This option will alter the autonomy level from a 4 to a level 3).

3.3.10 3-D Imaging/Stereoscopic Camera

A stereoscopic camera was provided by our sponsor, Dr. Guo. In accordance with the competition standards outlined in Section 4.1.6, any type of camera may be utilized for navigation. The sponsor-provided camera was chosen for this product as a proof of concept for optical image directed, obstacle avoidance. The sponsor-provided camera is the ZED Stereo camera and is a 3-D sensor which contains depth perception and motion tracking functionality. The ZED device is composed of stereo 2K cameras with dual 4MP RGB sensors. It has a field of view of 110° and can streams uncompressed video at a rate up to 100 FPS in WVGA format. It is an UVC-compliant USB 3.0 camera backward compatible with USB 2.0. Left and right video frames are synchronized and streamed as a single uncompressed video frame in the side-by-side format. Several configurations parameters of on-board ISP (Image Signal Processor) as resolution, brightness, contrast, saturation can be adjusted through the SDK that is provided by ZED development team. This camera has a compact structure and reduced size, compared to other stereo cameras. These characteristics make it relatively simple to incorporate into robotic systems or drones. The manufacturer's specifications are summarized in Table 12 below. A picture of the component is highlighted in Figure 11 below.

Table 12 - ZED Stereo Camera Specifications	
Output Resolution	Side by side 2x (2208x1242) @15fps
Output Format	YUV 4:2:2
Field of View	Max. 110° (D)
Depth Range	1 m to 15 m (3.5 to 49 ft)
Interface	USB 3.0 - Integrated 1.5m cable
Active Array Size	4M pixels per sensor
Dimensions	175x30x33 mm (6.89 x 1.18 x 1.3'')
Weight	159g - 0.35 lb
Power	380mA / 5V USB Powered

Figure 11 - ZED Stereo Camera

This will be the featured sensor to interpret obstacles. This project features a 3D camera that has distance sensing capabilities. Ideally the camera will feed a constant stream of image data or distance measurements to the GPU or MCU, thus allowing the MCU to determine the appropriate course of action. After analyzing the cameras specifications, our group discovered that the camera supports multiple video qualities, as well as framerates, and is capable of interacting with Robotic Operating Software (ROS) and can provide output of a point cloud with distance measurements from 0.5 to 20m. This point cloud can be sent to a GPU to be further analyzed, and then the analysis can be later transferred to the MCU.

3.3.11 Radar Module

To aid in the aspects of collision avoidance, radar sensors will be implemented because of the range span that the sensors provide. Radar can propagate at high frequencies and are able to detect objects within millimeters of the vehicle. The DFRobot SEN0192 can detect movements in a room, yard, or even on the other side of a wall. It's a Doppler radar sensor that operates in the X-band frequency at 10.525 GHz and indicates movements with oscillations in its high/low output. Sensitivity is manually adjustable with a potentiometer on the back of the device, offering direct line of sight detection from roughly 8 to slightly over 30 ft (~2.4 to 9+ m). The manufacturer's specifications are summarized below in Table 13. A picture of the component is highlighted in Figure 12 below.

Table 13 – DFRobot SEN0192 Radar Table

Quantity	Minimum	Typical	Maximum	Units
Operation Frequency	10.520	10.525	10.530	GHz
Radiation Power	10	13	14	dBM
Receiver Signal Strength	X	140	X	μVpp
Noise	X	X	3	μVrms
Supply Voltage	4.75	5.00	5.25	VDC
Supply current, EN high (Enabled)	X	60	X	mA
Supply current, EN low (Disabled)	X	37	X	mA
Pulse frequency	X	2	X	kHz
Pulse duration	5	X	X	μs
Operating temperature	32	X	131	F
Weight	X	15.1	X	g

Figure 12 – DFRobot SEN0192 Detector Radar

A constraint that come with the DFRobot SEN0192 is that it is not capable of detecting the precise location of an object. The SEN0192 can only detect movement that comes across its path. To able to detect the location of an object a higher frequency is required. Automobile industries use frequencies of 77 GHz.

3.3.12 Ultrasonic Proximity Sensor

Ultrasonic proximity sensors are assumed to be the most reliable of the secondary proximity sensors to be utilized in the project. Their low cost, effective range and speed are suitable for use as backup collision avoidance sensors. Ease of programming and integrating is also a factor when selecting this type of sensor for its intended purpose.

The HC-SR04 sensor is ranging module that provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The module includes ultrasonic transmitters, receiver and control circuit. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back. If the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time x velocity of sound (340M/S) / 2. The specifications provided by the manufacturer are summarized below in Table 14. A picture of the component is highlighted in Figure 13.

Table 14 - HC-SR04 Table

Power	5 V
Current	15 mA
Temperature	-10 ~ +70 Celsius
Detecting range	2 ~ 400 cm
Measuring angle	15 degrees
Interface	RS232 (TTL), PWM
Dimensions	45 x 20 x 15 mm
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion

Figure 13 - HC-SR04 Sensor

3.4 Possible Architectures and Related Diagrams

Multiple computing architectures are utilized in robotics and autonomous vehicles. Some of the architectures familiar for use in this project are highlighted below.

3.4.1 Real-Time Operating System (RTOS)

Real-Time Operating Systems serve applications that process data as it is received, usually without buffer delay. This technology will be vital to processing image data received from the ZED stereo camera in order to process images to facilitate object avoidance and course navigation in near time. From a competitive viewpoint, fast image processing will be required in order to navigate the course in a timely fashion while avoiding obstacles and fellow racers. As shown in Figure 14, a real-time operating system will quickly take images of an object and process them in order to pass data to the vehicle for collision avoidance.

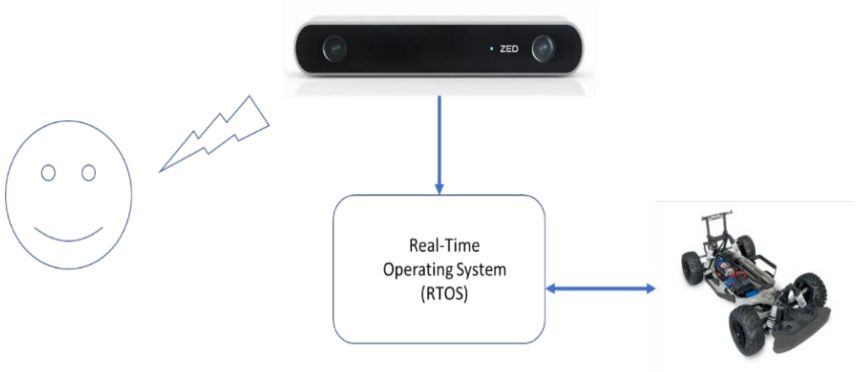


Figure 14 - Real-Time Operating System Object Detection Process

3.4.2 Ubuntu

Robotic Operating System (ROS—discussed below in section 3.4.4) will require Ubuntu or a similar Linux platform in order to run and process imaging data required for course navigation and object avoidance. Ubuntu will be loaded onto the Jetson Tx2 so that ROS can interface all of the sensors and give a working navigational computation that can be then communicated to the microcontroller.

3.4.3 Coding Language

The coding language used for this project will vary depending on how the processor is set up, as different companies will have their own IDE that will use different programming languages. The most common programming languages available through an IDE are C, C++, and Python. Out of these options, C would be the most preferred as the group is the most familiar with the language. As of this paper, the IDE provided will be S32 Design Studio, which offers a C-coding environment.

3.4.4 Robotic Operating System (ROS)

Robotic Operating System (ROS) is open source software licensed under BSD which provides libraries and tools to help software developers create robot applications. Robotic Operating System provides hardware abstraction, device drivers, libraries, visualizers, message-passing and more. The project’s potential utilization of Robotic Operating System would be for the main backbone of the communications between the image processor, sensors and microcontroller.

3.5 Parts Selection Summary

The part selection process was two-fold. As this project is sponsored by UCF professor Guo, some parts are provided and are not optional for utilization in the project. Other parts were selected specifically to meet the needs of supporting the provided parts in fulfilling power, object avoidance, course navigation and sensing duties required by the project. A summary of parts selection for those components that were not sponsor-provided are included in the following sections.

3.5.1 Power System

Linear voltage regulators will be designed and chosen to step down our input voltage of 11.1 V to usable 1.8V, 3.3V and 5V. These circuits will be integrated into a separate power system PCB in order to avoid potential signal interference with placing communications and power circuits on the same integrated PCB.

3.5.2 Microcontroller

A microcontroller will be selected that will be capable of receiving inputs from all external sensors and providing an adequate number of outputs to supply control signals to the motor and steering controls. The microcontroller will be run off of 3.3V and will need up to 155 mA of current. While this current rating seems especially high, the clock speed on the MK20DX128VFM5 warrants a high-power consumption. It goes without saying that the microcontroller could be replaced later on in the process if a microcontroller is presented that better suits the project's needs.

3.5.3 Ultrasonic Sensors

Ultrasonic sensors will be chosen that will be capable of detecting objects within a specified distance and feed that distance data to the microcontroller for interpretation and action. The HC-SR04 drives 5V at 15mA, which is also amplified by the number of sensors that we use. We will be using in the range of 6 to 10 sensors, which will pull in total anywhere from 90mA to 150mA. The low power consumption for even 10 sensors running in parallel is not necessary as the battery featured in the project will be rechargeable, but the power efficiency is helpful in testing.

3.5.4 Radar Sensors

Our group is considering using a radar as an auxiliary sensor to assist the microcontroller in conjunction with the ZED stereo camera to make decisions about where to navigate. The radar would be better than infrared as it is immune to luminosity that would negatively affect the infrared readings. The model selected to fill this need is the DFRobot SEN0192, which will drive 5 V at 37mA. Considering the high frequency of the signal it generates, it's amazing that the power consumption is so low.

3.5.5 Infrared Sensors

Infrared sensors will be chosen that will be capable of detecting objects within a specified distance and feed that distance data to the microcontroller for interpretation and action.

4.0 Related Standards and Realistic Design Constraints

Before designing a device, engineers must consider multiple restrictive factors which impact potential designs and restrict the ability to utilize specific components or methods of design. These standards are generally standardized by a governing body in order to maintain uniformity in design or to maintain safety for the end consumer of the engineered device. Realistic constraints must also be considered. Just because a design can include a multitude of features, does not mean that those features should be included. They may be cost restrictive, have minimal utility or negatively impact another portion of the design or the device's intended use by the end consumer. As such, this project will identify relevant standards and realistic design constraints that will shape how the autonomous vehicle is designed and what functions will be included in its final build.

4.1 Relevant Standards

Engineering standards are the key to design and implementation of an idea. Standards are what determine design flow in order to promote universal understanding of a specific function, capability or process of an engineering design. Without a common standard, the project design could be undecipherable from the collective engineering knowledge in use today.

4.1.1 Battery Standards

DS/EN 62952-2:2016 - Power Sources For A Wireless Communication Device - Part 2: Profile For Power Modules With Batteries [1]. IEC 62952-2:2016 specifies a profile for a power module containing batteries used as power source for wireless communication devices. This standard will dictate how we integrate the lithium polymer battery into our vehicle platform to power any onboard wireless communications devices.

IEEE 1625-2008 - IEEE Standard for Rechargeable Batteries for Multi-Cell Mobile Computing Devices [2]. This standard guides manufacturers/suppliers in planning and implementing the controls for the design and manufacture of lithium-ion (Li-ion) and lithium-ion polymer (Li-ion polymer) rechargeable battery packs used for multi-cell mobile computing devices. The provisions of this standard work together to define approaches to design, test, and evaluate a cell, battery pack, and host device to mitigate battery system failure. Additionally, this standard provides recommendations for end-user education and communication materials. This approach recommends the interfaces between subsystems (for example, cell, battery pack, host device, power adapter, etc.), and end users are as important to system reliability as is robust subsystem design and testing. This standard, therefore, includes subsystem interface design responsibilities for each subsystem manufacturer/supplier, and it provides messaging and communication provisions for end-user awareness. Therefore, the responsibility for total system reliability is shared between the designers/manufacturers/suppliers of the subsystems and the end user. Compliance to this standard requires adherence to all the provisions of the standard.

4.1.2 Printed Circuit Board Design Standards

The design will follow the standards of the IPC. These standards have had input from many individuals in many different industries worldwide in order to ensure reliability and minimum standards. This design will focus on the standards for IPC-2221B "Generic Standard on Printed Board Design" under the "Performance Classes" section. This section lays out three classes of design standards of printed circuit boards: class one, class two, and class three. Depending on the classification there are different levels of testing and functionality required.

4.1.2.1Class One

Class one printed circuit board standards are for "general electronic products" as described by the IPC. This is generally for any product that is not intended to have a long lifetime and is not an essential product for maintaining life. For example, this could be for simple cheaper mass-produced personal products such as flashlights or fitness trackers. The products that use this type of printed circuit board are typically cheaper but are not very robust.

4.1.2.2Class Two

Class two printed circuit board standards are for "dedicated service electronic products" as described by the IPC. This class of printed circuit board is meant for products that require some level of reliability and robustness. The products that use this type of printed circuit board are used for commercial and industrial purposes for products such as TVs and kitchen appliances. For this class of printed circuit board, it does not need to look neat and orderly, the board is only expected to work for the purposes it was designed for. In addition, the PCB should work for the expected lifetime of the product.

4.1.2.3Class Three

Class three printed circuit board standards are for "high reliability electronic products" as described by the IPC. These products are held to the highest design standard since they are usually for products that usually deal with maintaining or inducing the loss of human life. Some examples of products that use class three printed circuit board designs are missile systems for the military that must always hit the right target or pacemakers that are keeping the user alive. These printed circuit boards are the most expensive type of the three classes.

Because our autonomous vehicle is only a prototype with a limited budget, the design met a class one IPC design standard. If there was more time and money, the printed circuit board could have been designed to a class two. In addition, if mass production were already being planned, then a level two printed circuit board would help ensure a quality product that would function as expected in harsher environments for everyday use.

4.1.3 Wireless Standards

This standard has to do with wireless communications and what frequencies are allotted to the public and which are privatized. This project will be engineering wireless communication between the user and the vehicle, and so it is imperative that we utilize the proper frequency bands in order to avoid either receiving the wrong signals or interfering with other broadcasts.

ANSI X9.112-2016 - Wireless Management and Security - Part 1: General Requirements [3]. In today's world, both private and public sectors depend upon information technology systems to perform essential and mission-critical functions. In the current environment of increasingly open and interconnected systems and networks, network and data security are essential for the effective use of information technology. Privacy and regulatory requirements highlight this need. For example, systems that perform electronic commerce must protect against unauthorized access to confidential records and unauthorized modification of data. Wireless technologies are rapidly emerging as significant components of these networks. As such, data classification and risk assessments should be performed to determine the sensitivity of, and risk to, data transmitted over wireless networks. Various methods and controls should be considered for data that is sensitive, has a high value, or represents a high value if it is vulnerable to unauthorized disclosure or undetected modification during transmission over wireless networks. These methods and controls support communications security, for example by encrypting the communication prior to transmission and decrypting it at receipt. Note that data classification and risk assessments, regardless of whether data transmission is over wired or wireless environments, should be part of an organization's general security policy and best practices. Refer to Annex A Wireless Validation Control Objectives for further details. Part 1 of this Standard provides an overview of wireless radio frequency (RF) technologies and general requirements applicable to all wireless implementations for the financial services industry. Subsequent parts of this Standard will address specific applications to wireless technology and associated risks, as well as technologies, methods and controls that mitigate those risks. Note that other wireless non-radio frequency technologies, such as infrared and lasers are considered out of scope of this Standard.

4.1.4 Radar Standards

IEEE Std 521. IEEE Standard Letter Designations for Radar Frequency Bands. Since World War II, radar systems engineers have used letter designations as a short notation for describing the frequency band of operation. This usage has continued throughout the years and is now an accepted practice of radar engineers. Radar-Frequency letter designations are used for the following reasons:

1) They provide a convenient method for describing the band in which the radar operates without the need for awkwardly stating the limits of the frequency in numerical terms. For example, it is more convenient to say an L-band radar than a 12151400 MHz radar. This is especially important in titles of published papers on radar, in advertising of radar systems and components, or in any other situation where a short notation is desired.

2) In military radar systems, the exact frequency of operation cannot usually be disclosed, but it is permissible in many cases to describe the band in which it operates. The letter designations permit this.

3) Each radar-frequency band has its own particular characteristics. Thus, an X-band radar will be different from an S-band radar. The letter designations are often used in this manner to indicate the particular nature of the radar as it is influenced by its frequency. There are vast differences in characteristics, applications, and environmental constraints that distinguish radars in the different bands. It is the need to communicate concisely the whole set of characteristics which are shared by S-band radar, as distinguished from L-band radar, C-band radar, and the others, which requires the established usage of letter designations.

The Standard Letter Designations for Radar-Frequency Bands was first issued in 1976 and was written to remove the confusion that developed from the misapplication to radar of letter band designations of other microwave frequency users. This standard relates the letter terms in common usage to the frequency ranges that they represent. The 1984 revision defined the application of the letters V and W to a portion of the millimeter wave region while retaining the previous letter designators for frequencies. The 2002 revision included a change in the definition of millimeter wave frequencies to conform to the ITU (International Telecommunication Union) designation. The current (2019) revision keeps the same letter band designations but adds a "THz" band in recognition of the growing development of echolocation systems and associated technology in the region 300 GHz to 1000 GHz.

4.1.5 Microcontroller Standards

IEEE 1118.1-1990 - IEEE Standard for Microcontroller System Serial Control Bus [4]. A serial control bus for interdevice/intrabuilding as well as intrasite interconnection of microcontrollers is described. The bus, which is defined for (but not limited to) microcontrollers and devices with limited reprogramability, provides a multidrop bit-serial communication protocol that will allow the interconnection of distributed Independently manufactured devices. The protocol is optimized for instrumentation, distributed data acquisition systems, control devices, and test and measurement. Specifications for a common architecture, generic bus services, system wagement, data link, and several physical media are provided. The serial control bus expands upon BITBUS without making existing devices obsolete.

System reliability has been enhanced by the addition of a system management layer, and generic bus services have been expanded.

4.1.6 Safety Standards

The NHTSA has a list of multiple definitions and regulations on automated vehicles, as well as the history of car safety with an emphasis on how the future trend of automated vehicles. The biggest feature of the NHTSA site is that it defines different “levels” of automation. The National Highway Safety Administration (NHTSA) and Society of Automotive Engineers (SAE) have each published a formal classification system for automated vehicles. The NHTSA 14-13 system focuses on the capabilities of the vehicle control system and its ability to relieve the driver of driving responsibility. The SAE system is based on the amount of driver intervention and attentiveness required. Each level is briefly described below. The NHTSA system defines five levels, (Level: 0, 1, 2, 3, 4), of vehicle automation:

Level 0 – No-Automation. The driver is in complete and sole control of the primary vehicle controls (brake, steering, throttle, and motive power) at all times, and is solely responsible for monitoring the roadway and for safe operation of all vehicle controls. Vehicles that have certain driver support/convenience systems but do not have control authority over steering, braking, or throttle would still be considered “level 0” vehicles. Examples include systems that provide only warnings (e.g., forward collision warning, lane departure warning, blind spot monitoring) as well as systems providing automated secondary controls such as wipers, headlights, turn signals, hazard lights, etc. Although a vehicle with V2V warning technology alone would be at this level, that technology could significantly augment, and could be necessary to fully implement, many of the technologies described below, and is capable of providing warnings in several scenarios where sensors and cameras cannot (e.g., vehicles approaching each other at intersections).

Level 1 – Function-specific Automation: Automation at this level involves one or more specific control functions; if multiple functions are automated, they operate independently from each other. The driver has overall control, and is solely responsible for safe operation, but can choose to cede limited authority over a primary control (as in adaptive cruise control), the vehicle can automatically assume limited authority over a primary control (as in electronic stability control), or the automated system can provide added control to aid the driver in certain normal driving or crash-imminent situations (e.g., dynamic brake support in emergencies). The vehicle may have multiple capabilities combining individual driver support and crash avoidance technologies but does not replace driver vigilance and does not assume driving responsibility from the driver. The vehicle’s automated system may assist or augment the driver in operating one of the primary controls – either steering or braking/throttle controls (but not both). As a result, there is no combination of vehicle control systems working in unison that enables the driver to be disengaged from physically operating the vehicle by having his or her hands off the steering wheel AND feet off the pedals at the same time. Examples of function specific automation systems include cruise control, automatic braking, and lane keeping.

Level 2 - Combined Function Automation: This level involves automation of at least two primary control functions designed to work in unison to relieve the driver of control of those functions. Vehicles at this level of automation can utilize shared authority when the driver cedes active primary control in certain limited driving situations. The driver is still responsible for monitoring the roadway and safe operation and is expected to be available for control at all times and on short notice. The system can relinquish control with no advance warning and the driver must be ready to control the vehicle safely. An example of combined functions enabling a Level 2 system is adaptive cruise control in combination with lane centering. The major distinction between level 1 and level 2 is that, at level 2 in the specific operating conditions for which the system is designed, an automated operating mode is enabled such that the driver is disengaged from physically operating the vehicle by having his or her hands off the steering wheel AND foot off pedal at the same time.

Level 3 - Limited Self-Driving Automation: Vehicles at this level of automation enable the driver to cede full control of all safety-critical functions under certain traffic or environmental conditions and in those conditions to rely heavily on the vehicle to monitor for changes in those conditions requiring transition back to driver control. The driver is expected to be available for occasional control, but with sufficiently comfortable transition time. The vehicle is designed to ensure safe operation during the automated driving mode. An example would be an automated or self-driving car that can determine when the system is no longer able to support automation, such as from an oncoming construction area, and then signals to the driver to reengage in the driving task, providing the driver with an appropriate amount of transition time to safely regain manual control. The major distinction between level 2 and level 3 is that at level 3, the vehicle is designed so that the driver is not expected to constantly monitor the roadway while driving.

Level 4 - Full Self-Driving Automation (Level 4): The vehicle is designed to perform all safety-critical driving functions and monitor roadway conditions for an entire trip. Such a design anticipates that the driver1 will provide destination or navigation input but is not expected to be available for control at any time during the trip. This includes both occupied and unoccupied vehicles. By design, safe operation rests solely on the automated vehicle system

The SAE standard, J3016_201401 defines six levels, (Level: 0, 1, 2, 3, 4, 5), of vehicle automation:

1. Level zero (0) maintains that the driver is responsible for all aspects of driving, but the vehicle can provide automated warnings.
2. Level one (1) expects the driver to be able to perform all driving tasks at any time but be able to take advantage of assistance systems for steering or acceleration/deceleration systems such as cruise control, lane keeping, and parking assistance systems.
3. Level two (2) requires the driver to be able to detect when to take control over of any active automated system.
4. Level three (3) permits the driver, under limited conditions, to safely focus on tasks other than driving, but to be ready to take over when notified by the vehicle.
5. Level four (4) expands the scenarios that the automated vehicle can safely operate but requires the driver to determine when it is safe to do so. If the vehicle automation is appropriately activated, the driver may place their attention elsewhere.
6. Level five (5) requires no human intervention except to start the system and provide a destination.

For this project, the most appropriate level is a level 4, which is defined as being “capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.” ([1]) The current proposal is to have two sets of overrides which can demote the vehicle to either a level 3—where the “driver is a necessity, but is not required,” and “must be ready to take control of the car at all times with notice”—or a level 0 where the driver “performs all driving tasks”. The details of each failsafe technique are described above.

Preparing for the future of Transportation: Automated Vehicles 3.0 (AV 3.0 [5]. Preparing for the Future of Transportation: Automated Vehicles 3.0 (AV 3.0) builds upon Automated Driving Systems 2.0: A Vision for Safety (ADS 2.0). AV 3.0 expands the scope to all surface on-road transportation systems and was developed through the input from a diverse set of stakeholder engagements, throughout the Nation. AV 3.0 is structured around three key areas: Advancing multi-modal safety, reducing policy uncertainty, and Outlining a process for working with U.S. DOT. The U.S. DOT sees AV 3.0 as the beginning of a national discussion about the future of our on-road surface transportation system. The U.S. DOT is seeking public comments on the AV 3.0 (Federal Register Notice, DOT-OST-2018-0149 - comment period closed on December 3, 2018), Preparing for the Future of Transportation: Automated Vehicles 3.0 [ISBN 978-0-16-094944-9].

4.1.7 Programming Standards

IEEE 2050-2018 - IEEE Standard for A Real-Time Operating System (RTOS) For Small-Scale Embedded Systems [6]. A real-time operating system (RTOS) called μT-Kernel for small-scale embedded systems such as systems with a single chip microcomputer including 16-bit CPUs, systems with a small amount of ROM/RAM, and systems without a memory management unit (MMU) are specified in this standard. This set of standards entails portability at the source code level and affects how this project will be coded in order to facilitate transferring source code to a similar platform for evaluation or replication.

4.1.8 Competition Standards

As the project is sponsored by UCF with an ultimate goal of competing in the annual F1Tenth Racing Competition, the project must meet specific requirements set forth by the racing governing body.

Vehicle Classes - We will have two vehicle classes for the competition:

- F1/10 Restricted Class
- F1/10 Open Class

A team may participate in both classes if they choose to do so. It can have 2 different cars, one for each class, or use the same car for both classes (in which case, of course, that one car must meet the restrictions of the Restricted class). F1/10 Restricted Class will only allow cars which meet the following specifications:

- A 1/10 scale rally car chassis equivalent to the Traxxas model 74054 type is allowed.
- Four-wheel drive, and two-wheel drive versions are both allowed > in this class.
- Only the use of stock tires, or equivalent - in size and profile, is allowed. No special traction modifications are allowed, this includes:
  - Applying any liquids or gels of any kind to the stock tires
  - Using alternate racing tires
- Use of NVIDIA Jetson TX2 or an equivalent capability processor or anything of lower spec is allowed.
- Use of Hokuyo 10LX or an equivalent LIDAR range sensor or anything with a lower spec is allowed.
- Multiple LIDARS are allowed, as long as they are all equivalent to, or lower spec than, the Hokuyo 10LX.
- There are no restrictions on the use of cameras, encoders, or custom electronic speed controllers.
- Use of Brushless DC motor equivalent to Velineon 3500 or anything of lower spec is allowed.

It is up to the teams to demonstrate that they meet the above specifications for racing in the restricted class. F1/10 Open Class will allow cars which are outside the restricted specifications but still adhere to the following limitations:

- Car dimensions should be within 10% of the dimensions of the car required in Restricted class.
- (This is to make sure that it can fit comfortably in the racing track, and that it can compete with other cars in head-to-head race.)
- Only electric drive motors are allowed.

Race Classes - We will have two different racing classifications:

- Time Trial Race
- Head to Head Race

To be eligible to compete and win prizes in either classification, you must demonstrate the ability to remotely issue a stop signal to bring your car to a *safe* stop. This will typically mean that the car comes to a complete halt when it receives the signal. Issuing the stop signal must not require you to be in close proximity to the car; specifically, you will be seated at one end of the track, and the car at the other. This will be tested by the organizers prior to the race during the practice session

- Time Trial Race - Multiple heats are held where the goal of each car is to set as many laps as possible with the least (ideally no) amount of crashes. For each heat (typically 3-5 mins), we will log the number of laps set and your fastest lap time. Multiple heats allow you to change the car setup/parameters to try being more aggressive, or safe. Details about restart procedures, and crashes are announced prior to the race in the practice session since they depend on the track layout.
- Head-to-head Race - The organizers will decide your car's eligibility to go head-to-head. Here is the criteria that will help us decide that, and which you can replicate prior to competition day:
- Static obstacle avoidance: We will place cardboard/foam obstacles (with width and height similar to the restricted class dimensions) in the track at random locations and you need to demonstrate that the car can safely avoid them.
- Moving obstacle avoidance: we will slide the cardboard obstacles on the track, and your car must avoid these obstacles If your car collides often with the track or the obstacles, it won't be eligible for head-to-head racing.

The code you demonstrate is the code that will race. e.g. you can't demonstrate one code base at slow speed for obstacle avoidance, then run another at high speeds and crash.

We understand that the F1/10 head-to-head race is challenging, it is impossible for the rules to be comprehensive, so we will use 'common sense' to make that determination. For example, if your car appears to be rear-ending another vehicle and causing any perceivable damage, it may be removed from the race. We don't want cars to get damaged.

4.1.9 Autonomous Vehicle Standards

Autonomous Vehicles | Self Driving Vehicles Enacted Legislation

The NCSL is a third-party organization that reported on all the legislation passed at the state and national levels regarding autonomous vehicles. This will help the project when it comes to testing the vehicle as there are specific regulations about where the car can and cannot operate. Additionally, if the project happens to scale up into full-sized cars, this will be an excellent guide in making safety and equipment decisions.

Radio Control Radio Service (RCRS)

4.1.10 Robotics Standards

IEEE Standard Ontologies for Robotics and Automation. IEEE has published and maintained multiple standards of wireless communication. By utilizing these standards, our communications will be more effective and more organized. These standards will also give us a basis to guide us as we write our drivers and protocols.

4.2 Realistic Design Constraints

Realistic design constraints are tailored to the specific project. Examples of realistic constraints are those that directly impact how the system will be designed (e.g. the timeframe for this project is two semesters). An unrealistic design constraint would entail a factor that would affect electrical devices as a whole but may not specifically apply to this project (e.g. electrical devices can handle thousands of volts, ours can handle around eight volts)

4.2.1 Economic and Time Constraints

The first project constraint will be time. As the timeframe allotted is fixed and non-flexible, our team will have to allot and manage time efficiently in order to accomplish the scope of the project within the allowed timeframe. Our team will need to make sure that ample parts and models are ordered with enough time to have them shipped, prototyped, and tested before the project's presentation. Our goal is to have all of the PCB designs, sensors, hardware modifications, and wireless communication components ordered by the end of December. As the team progresses, this constraint may potentially be felt more than any other constraint in the design and implementation process. AS such, the team milestone matrix will need to be followed and adjusted accordingly to compensate for any unforeseen events that could delay a specific task.

Funding is also a concern as multiple components are cost prohibitive and being provided by our sponsor. Smaller components such as the PCBs, MCU and associated wiring will be financed by the individual team members and we will attempt to stay under $500. The team understands that additional expenditures may inevitably be required depending upon design, prototyping, and testing results. Even the most perfect designs may result in a requirement for rework or redesign to achieve a specific goal that may not be realized until later in the process.

4.2.2 Environmental, Social and Political Constraints

Environmental constraints for this project directly relate to the utilization of rechargeable batteries in the design. Whenever chemical storage batteries are utilized in a device, there are environmental concerns surrounding operation and disposal of the batteries contained therein. With the proliferation of autonomous vehicles and electric powered vehicles in general, the environmental impact on mining, global warming and industrial pollution generated from the increased need for specific materials required to create the batteries increases which also creates a large carbon footprint for the autonomous and electric vehicle industry as a whole.

Social constraints pertaining to this project involve the inherent desire of human beings to remain in control. This can be seen daily as commuters put forth greater expense to drive their own vehicles when ride sharing, and public transit will deliver them to their destination at a lower cost. The desire to be in control flows into society’s generalized feelings toward autonomous vehicles. General attitude towards autonomous vehicles range from the inane such as a belief that a human driver is better able to compensate for road conditions, to the insane where a belief that machines will develop a mind of their own and develop a desire to destroy mankind.

Political constraints will stem from prevalent social constraints in the short term and in environmental constraints in the long term. In the short term, political viewpoints will coincide with the prevalent societal views of the populace. Politicians will make decisions regarding autonomous vehicles in a way that will support the viewpoints of their constituents and their financial benefactors. In the long term, enough of a societal view that environmental impacts of fossil fuel driven vehicles and safety concerns that arise from inefficient driving capabilities of the populace will outweigh the social viewpoints of independence and control.

4.2.3 Ethical, Health and Safety Constraints

Ethical constraints will arise when decisions must be made by the autonomous vehicle. Suppose a child runs out in front of an autonomous vehicle and the car must make a decision to swerve to avoid the child. What if the decision left to the autonomous vehicle is to either swerve and miss the child, but upon swerving the car will strike an adult? The decision made by the autonomous vehicle will be considered through an ethical lens. Will the child’s life take precedence over the adult’s life? Will proportional property value be taken into account if the choice was between hitting a new Mercedes or crashing into a public building? Whatever way the autonomous vehicle decides, those decisions will be viewed through the lens of a population’s own ethical decision making and influence perception of the efficacy of the autonomous vehicle.

Health constraints will directly correlate to the general health precautions taken with any electrical devices. Low voltages utilized in the design of the project will minimize electrical shock concerns, but they will still be present, and precautions should be taken to avoid touching electrical components while the device is in operation. Another health concern related to electrical devices is the risk of lead exposure due to the presence of solder utilized in adhering and connecting electrical components inside a device. Lead exposure should be minimal due to the small quantities of solder utilized and an ongoing industry awareness of the risks of lead-based solder. However, precautions should be taken to avoid contact with the solder joints and components of the device in order to minimize possibility of lead exposure.

Safety is another constraint. Due to the autonomous nature of our project, our group must integrate specific safety precautions in order to mitigate liability in the case of interference or a “glitch” in the system. Also, due to the quiet nature of electric motors, we must devise an audible alert mechanism for bystanders to be able to recognize the autonomous vehicle while in operation. Another concern is that of the battery. Lithium Polymer batteries have notoriety for excessive heat during operation in enclosed spaces. The potential need for a cooling apparatus to keep the battery pack at a stable temperature may be warranted even though newer battery technology has made great strides in lowering operating temperatures of battery cells. Another safety concern is that of the speeds in which the vehicle is capable of traveling. It may become necessary to program a “governor” in order to lower the top operating speeds, even though this would be detrimental to our sponsor’s intended final goal of racing the autonomous vehicle in a competitive arena. This issue will need to be investigated further to ensure the team performs its due diligence in this regard and avoid any potential for physical harm to bystanders or property damage due to a collision at higher speeds.

4.2.4 Manufacturability and Sustainability Constraints

Manufacturability is a constraint in that any custom parts or components will have to be kept to a minimum in order to allow the project design to be recreated and implemented by the sponsor into an existing product. This will most likely come into play when the team is forced to either design and/or construct mounts for peripheral sensors. The team will strive to either implement available aftermarket parts to mount peripherals to the vehicle platform or design any custom parts in such a manner as to facilitate ease of manufacturing for the sponsor with tools and materials available to the College of Engineering (i.e. the Texas Instruments 3-D printing lab).

Physical dimensions of the host vehicle will dictate the size and scope of the electronics that we are able to install onto the platform. The vehicle’s chassis is not overly large—around .15 m^2—and so if more space is needed our team will have to find innovative solutions to solve the size problem. As discussed previously, the height of the vehicle should be as low as possible to maintain a lower center of gravity for turns and to reduce friction from wind drag. It will most likely be impossible to place the Lexan body that was included with the sponsor provided vehicle platform, so maintaining the lowest possible center of gravity is essential to prevent rollovers in a turn at speed during a competition. Any type of rollover onto an unprotected PCB and/or expensive peripheral sensors could be catastrophic to the operation of the vehicle, not to mention costly to the racing team.

Weight is a constraint due to the loading capacity of the host vehicle as well as to maintain a competitive edge for the end goal of racing the autonomous vehicle.

The weight capacity of the vehicle is mainly dependent upon the scale shock absorbers attached to the frame and the torque produced by the motor. If too much weight is applied to the vehicle, the risk of drag in corners or decreased speed could become detrimental to operation of the autonomous vehicle. As such, the car should be as light as possible but realistically it should maintain a weight less than 15 pounds in order to stay competitive and house all of the necessary components.

Computing Power/Memory Capacity is a constraint due to a large number of computations and visual processing that will be required for the object avoidance and mapping functionality of the project. The TX2 has a GPU with 8GB of memory on board, and so this will provide the ceiling for our processes in terms of memory capacity. The onboard MCU memory can range from 64kB to 1MB dependent upon the specific MCU utilized. Due to the nature of the embedded systems that the team will be utilizing, program storage size can also be a constraint that could adversely affect our design. If the onboard program memory size is insufficient to hold our code, we will either need to optimize the code further or select a more robust MCU for our purposes.

5.0 Project Hardware and Software Design Details

The following sections detail the design from an architectural standpoint. A thorough explanation is presented for every device, system, and subsystem involved in this project in hopes to explain the methodologies taken in designing the project. By showing our thought process and design choices, the aim is to allow future research interests to learn and develop the project based on the success and failure that we experience.

5.1 Initial Design Architectures and Related Diagrams

A proposed hardware diagram of the project’s main components is highlighted below in Figure 15.

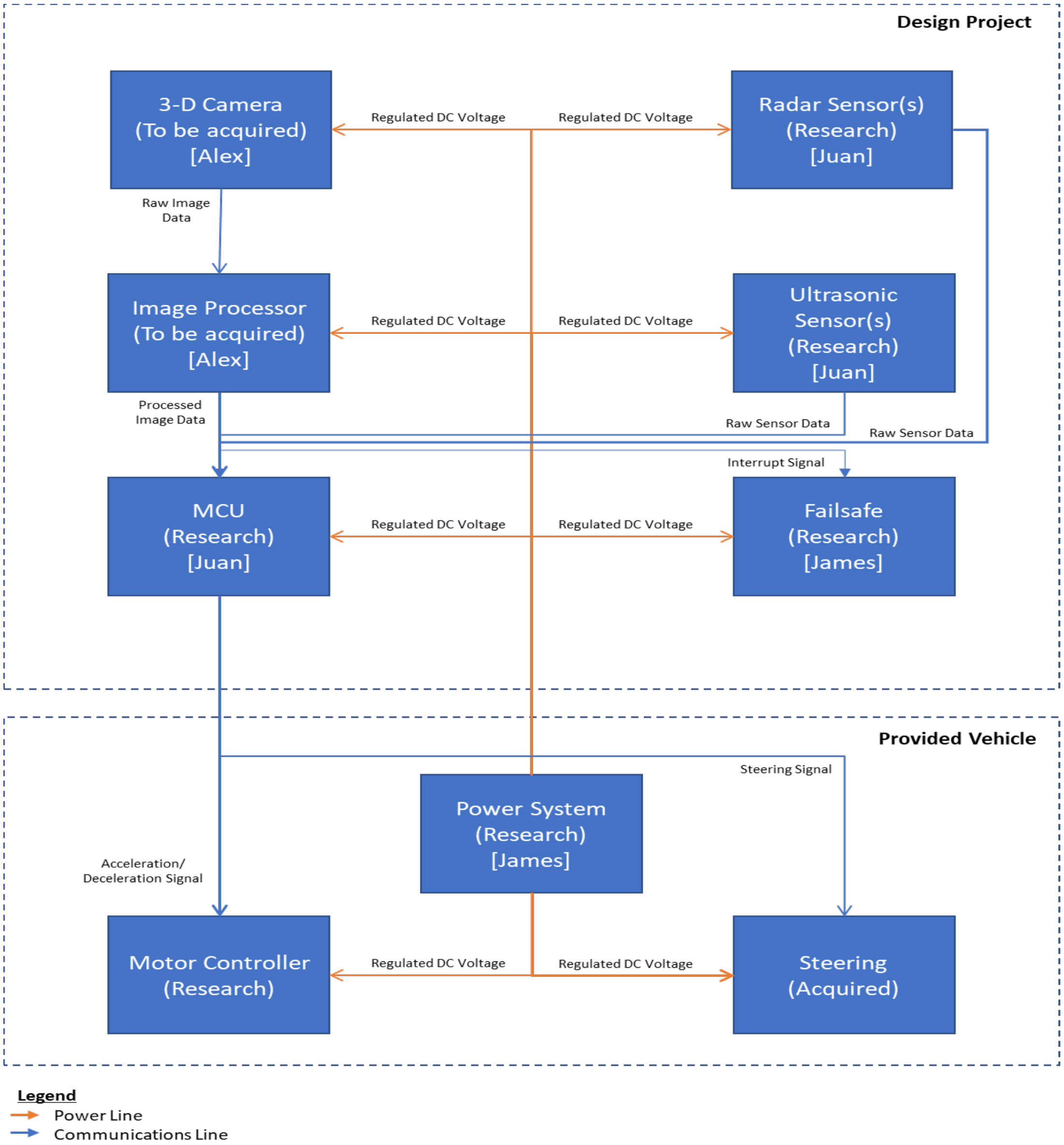


Figure 15 - Project Block Diagram

The project will consist of three sub-systems: the vehicle platform, the payload and the peripherals.

5.1.1 Vehicle Platform Diagrams

The basic block diagram of the vehicle platform sub-system is illustrated below in Figure 16.

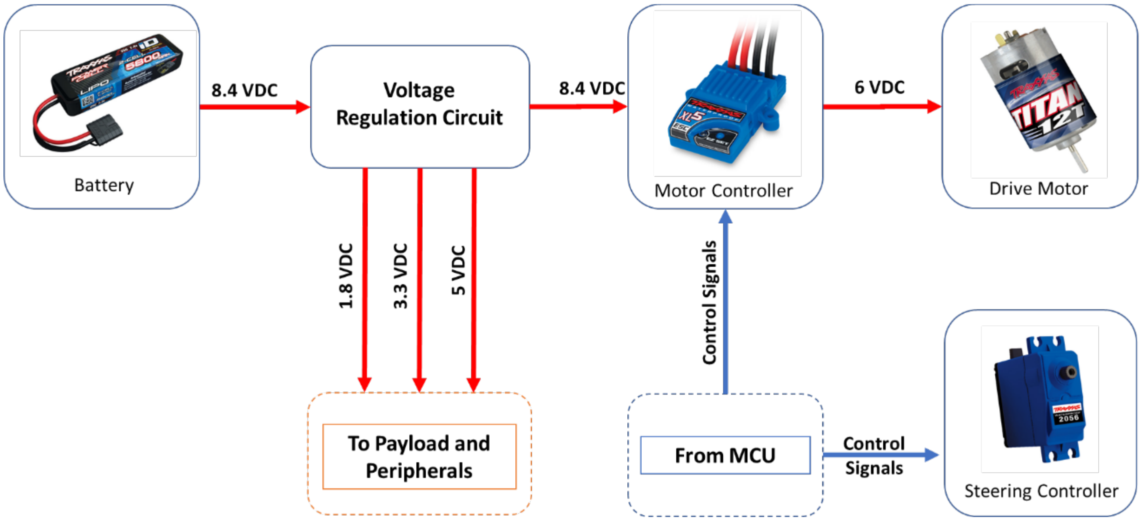


Figure 16 - Vehicle Platform Sub-System Block Diagram

5.1.2 Payload Diagrams

The basic block diagram of the payload sub-system is illustrated below in Figure 17.

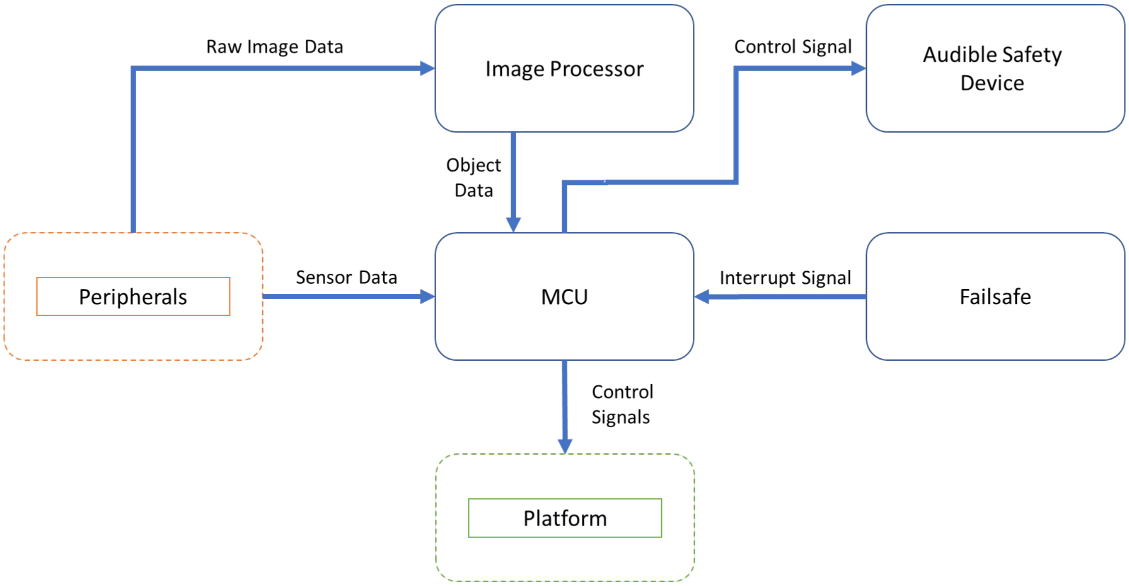


Figure 17 - Payload Sub-System Block Diagram

5.1.3 Peripheral Diagrams

The basic block diagram of the peripherals sub-system is illustrated below in Figure 18.

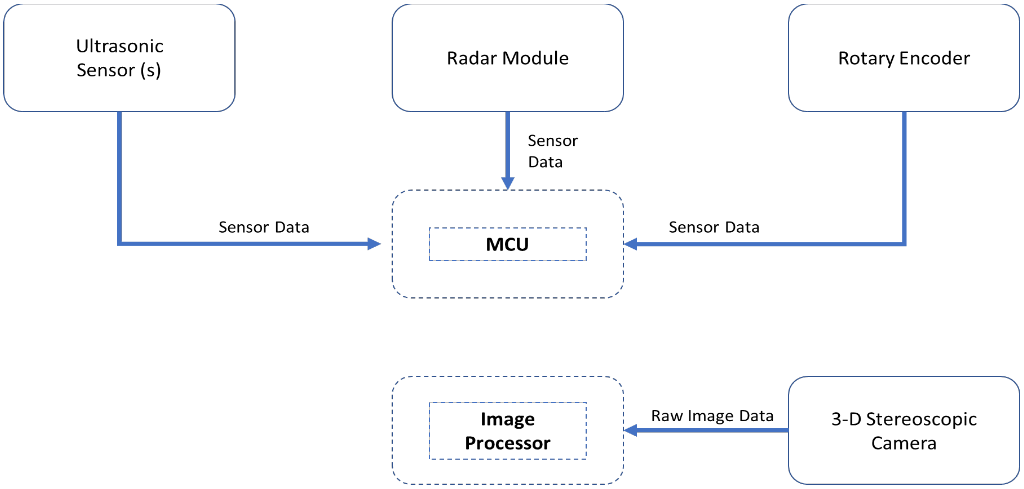


Figure 18 - Peripheral Sub-System Block Diagram

5.2 Vehicle Platform Sub-System

The vehicle platform sub-system consists of the motive and power components for the project. The vehicle platform will be the means in which our project will traverse the course and provides the actual motive and steering components, infrastructure and mechanics. The vehicle platform will also supply power to the payload and peripherals sub systems.

5.2.1 Power System

A lithium-polymer battery will be utilized to provide an input 11.1 VDC at 5800mAh to the power system. This input voltage and current will be regulated to three separate voltages that will be utilized by the project: 1.8V, 3.3V and 5V. These voltages will be provided to the payload and peripherals sub-systems via interconnecting wiring harnesses. The 11.1 VDC input will also be directly routed to the motor controller without being regulated in order to provide power the drive motor.

5.2.1.1Breadboard Test

As power system components have yet to arrive, breadboard testing of the completed power system is not feasible at this time. However, once constructed, the power system will be tested to ensure that regulated voltages of 1.8V, 3.3V and 5V are properly output by the power system in order to feed voltage and current to the vehicle platform, payload and peripherals sub-systems.

5.2.1.2Schematics

Three separate stepdown voltage regulators will be utilized to provide the 1.8V, 3.3V and 5V values required to power all components within the system. The schematic for the Power Systems board is provided below in Figure 19.

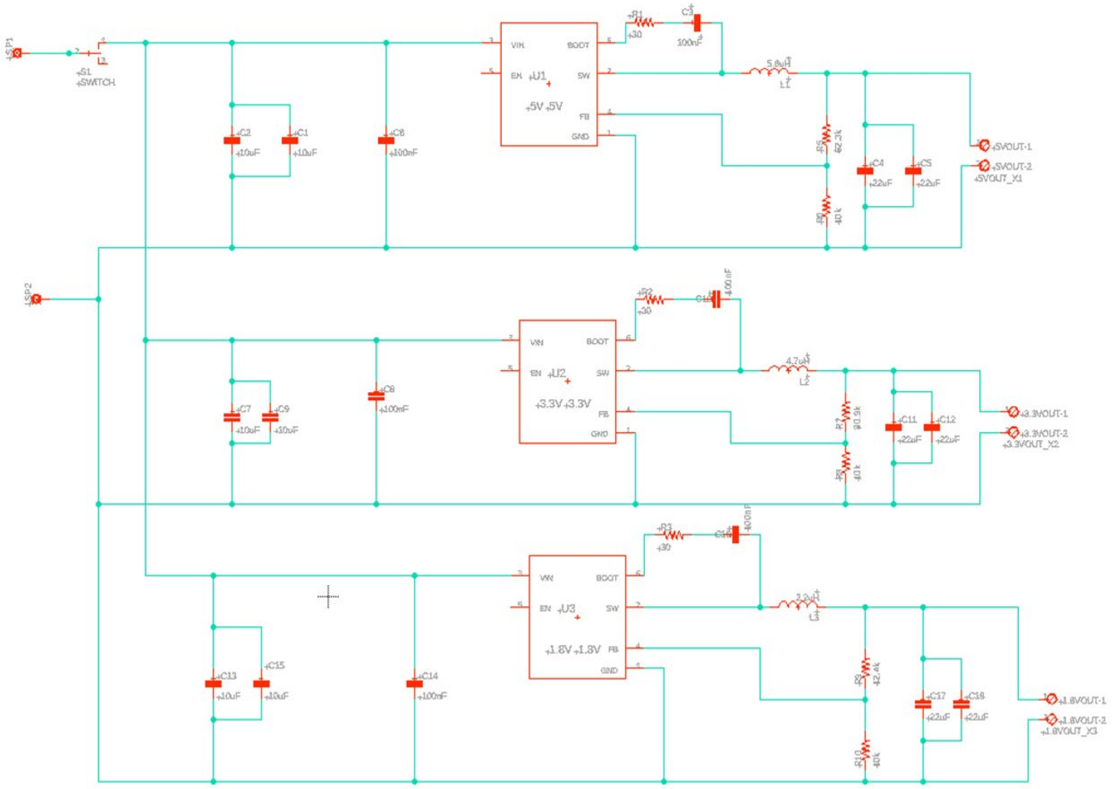


Figure 19 - Power System PCB Schematic

5.2.2 Drive Motor

The drive motor will be supplied 6V, 1A from the motor controller. The motor controller will determine the supplied current and polarity to propel the drive motor in forward or reverse rotation and determine the velocity of the drive motor. The rotation of the drive motor shaft will be translated to the vehicle tires via a geared transmission that will turn the vehicle platform’s main driveshaft. Rotation of the vehicle platform’s main driveshaft will drive another geared transmission that will rotate the axles of the vehicle platform’s wheels, thus rotating the attached wheels. Rotational power is supplied to all four tires at once (i.e. four-wheel drive).

5.2.2.1 Breadboard Test

To perform basic testing of the code required to provide a PWM signal to a DC motor for both forward and reverse operation, we setup a breadboard test with a basic DC motor, a microcontroller and a power supply. This basic test enabled the team to verify basic code that can be integrated into our final design to provide motor control inputs in order to accelerate or decelerate to avoid object collisions and navigate a course in real time. Basic breadboard testing for a DC brushed motor is illustrated below in Figure 20.

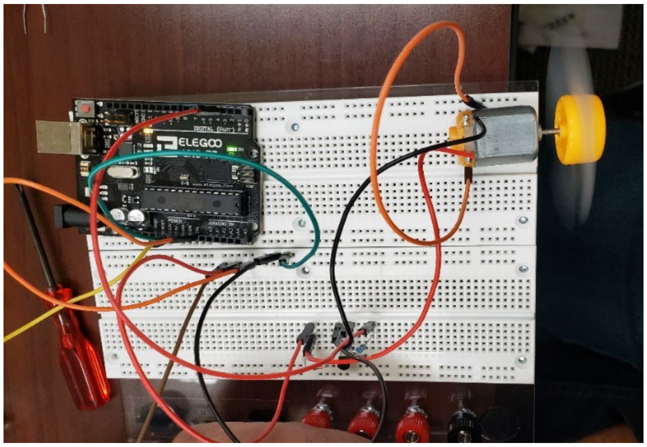


Figure 20 - DC Brushed Motor Breadboard Testing

5.2.3 Motor Control

The motor control will supply voltage and current based upon the internal BEC of the motor controller. This value for the sponsor-provided motor controller will be a max 6V at 1A. Polarity of the voltage will determine the rotational direction of the drive motor and the amount of current supplied to the drive motor will determine rotational velocity of the drive motor.

5.2.3.1Breadboard Test

The motor controller was tested to ensure a 6V, 1A output was generated as is required for the input of the sponsor provided 12 turn DC motor. The motor controller tested is capable of outputting a wide range of voltage and current which was tested to ensure intended operating voltages and current as well as potential for controlling a more powerful motor upgrade for competition. Breadboard testing of the DR10002 Motor Controller is illustrated below in Figure 21.

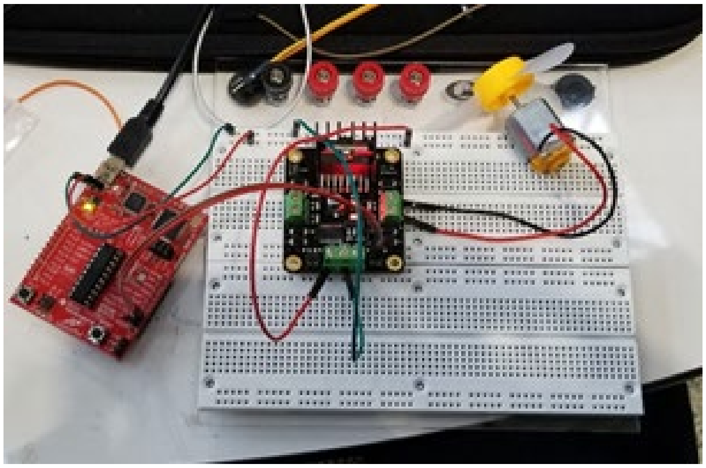


Figure 21 - DRI0002 Motor Controller Testing

5.2.3.2Schematics

A working schematic for the Traxxas motor controller was unable to be located during the course of our research. It was determined that the schematic was proprietary to Traxxas and they do not publish it.

5.2.4 Steering Control

The steering control circuit is testing via supply a PWM signal to the servo and ensuring full range of rotation. Testing will be done by issuing a wide range of amplitude values as inputs to verify that the characteristics are accurate and can be reliably manipulated.

5.2.4.1Breadboard Test

The steering controller will be tested to ensure 60° of rotation to provide steering for the vehicle platform. The steering controller in position is illustrated below in Figure 22.

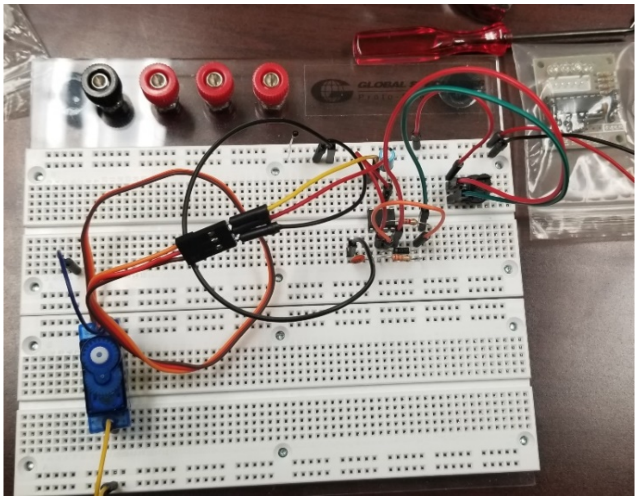


Figure 22 - Traxxas High Torque Servo - Steering Controller Testing

5.2.4.2Schematics

To test the servo, a basic servo tester circuit will be constructed and eventually utilized in the project design to provide steering signals sent by the MCU to the servo for rotation. The basic circuit is illustrated below in Figure 23.

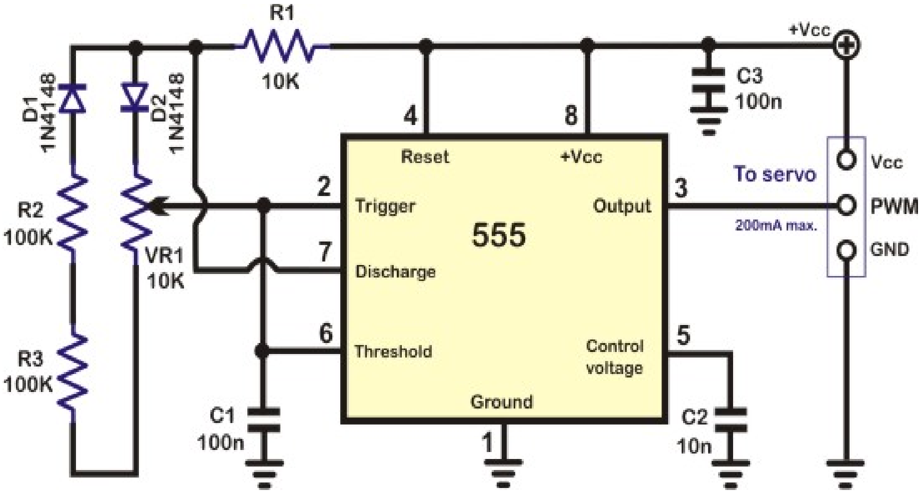


Figure 23 - Basic Servo Controller Test Circuit

5.3 Payload Sub-System

The payload sub-system will include the processing components of our system. These will include the image processor and MCU as well as the two safety modules. The payload subsystem will function as the “brains” of our autonomous vehicle through decision object recognition, decision making algorithms and output control signals to the vehicle platform subsystem. All autonomous “thinking” will be handled by this sub-system.

5.3.1 Image Processor

The image processor is a Jetson XT2 which is a sponsor provided component as dictated by the competition standards outlined in Section 4.3.3. The versatility and computing power required to detect objects and navigate a course is embodied by the Jetson and NVIDIA touts it as the premier board for accomplishing 3-D autonomous work.

5.3.1.1Breadboard Test

Due to the physical dimensions of this component, a breadboard test is not feasible, but a functional test will be conducted. The component is illustrated below in Figure 24.

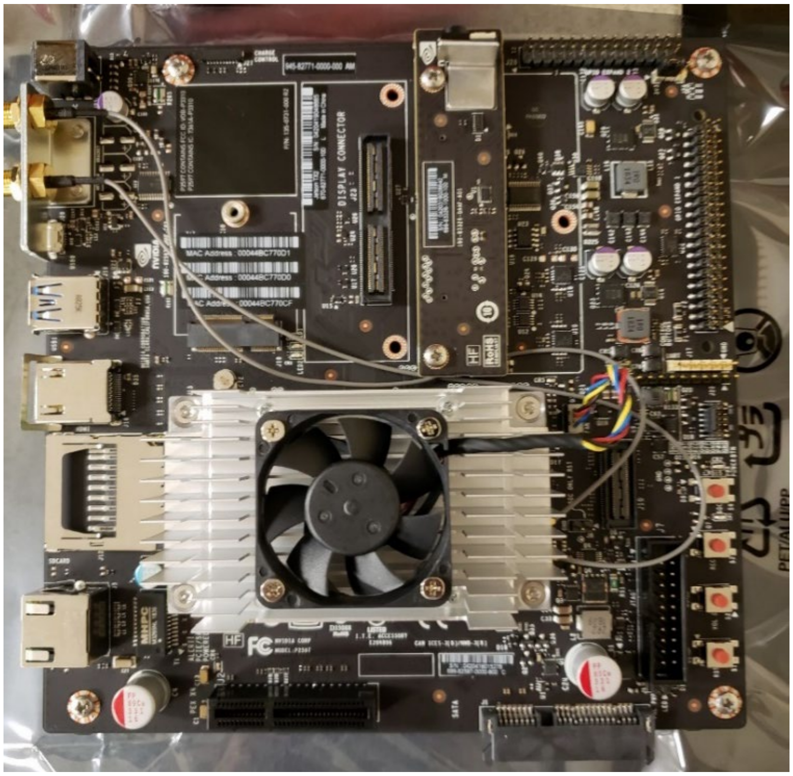


Figure 24 - Jetson Image Processor Testing

5.3.1.2Schematics

Wiring diagram and Signal Flowchart for the Jetson TX2 are illustrated below in Figure 25 and Figure 26 respectively.

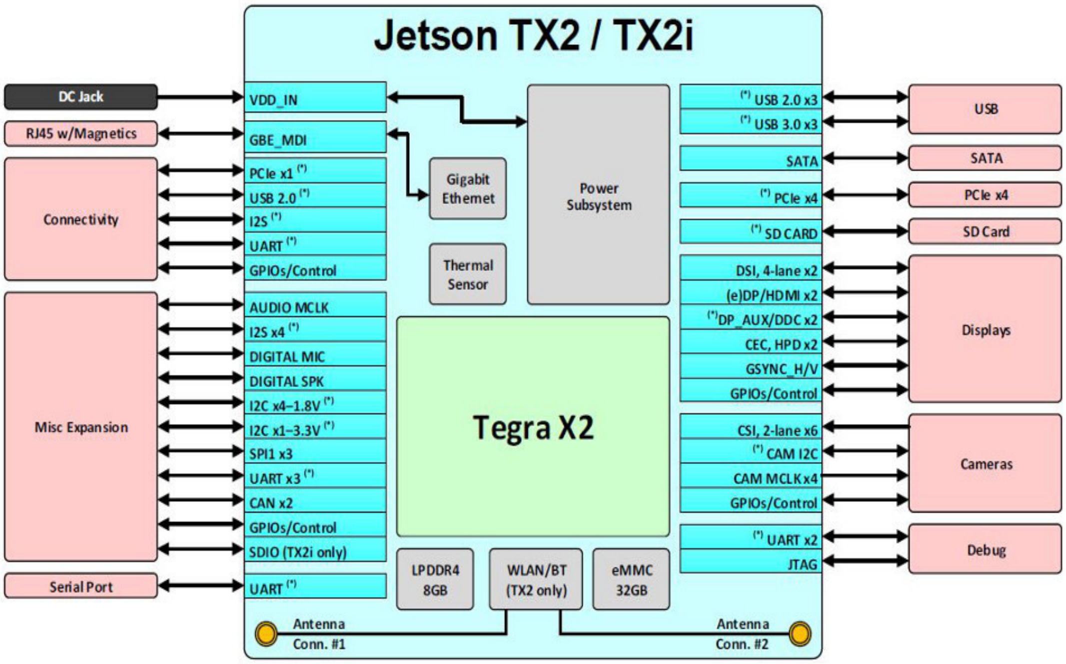


Figure 25 - Jetson TX2 Image Controller Wiring Diagram

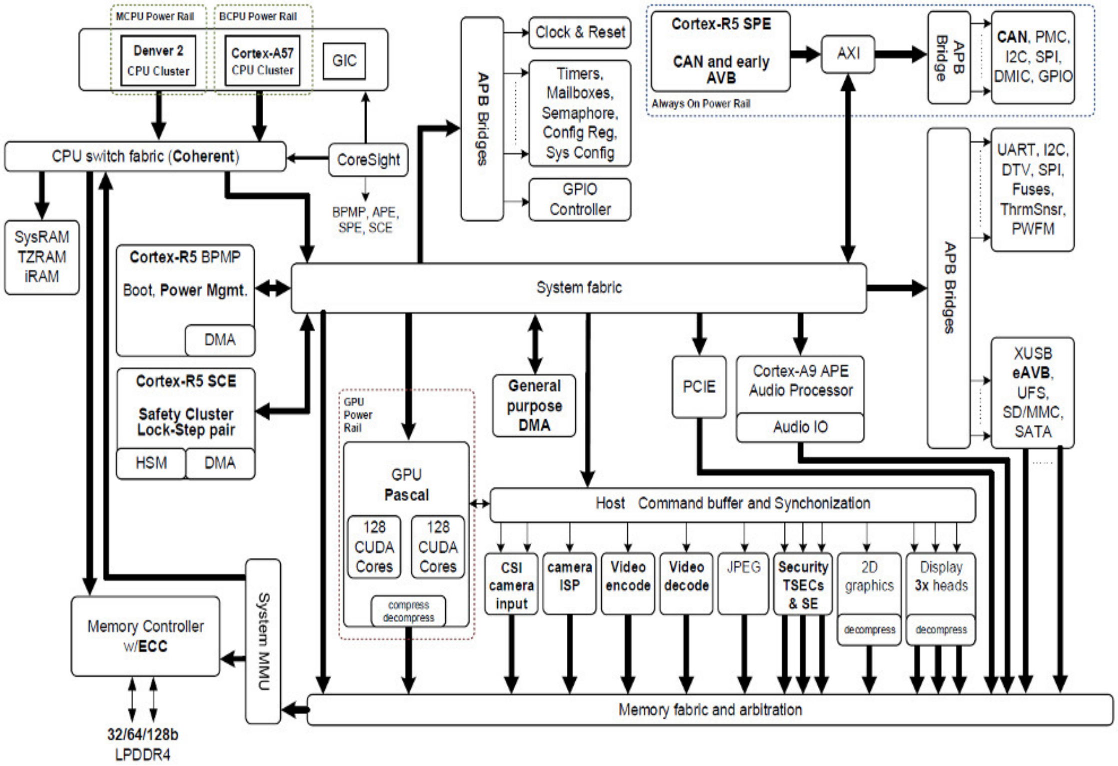


Figure 26 - Jetson TX2 Image Processor Signal Flowchart

5.3.2 MCU

The onboard microcontroller will take inputs from the image processor, secondary sensors, battery and speed sensors to provide steering outputs and speed controls in order to avoid collisions and navigate a course in real-time. It goes without saying that the microcontroller should have the ability to communicate with all of the peripheral devices, but it will also need enough ADC channels to be able to read several HC-SR04 Ultrasonic Sensors. Currently the project will use eight of these ultrasonic sensors, and so the microcontroller would either need eight ADC channels, or the PCB will need a multiplexer to cycle through which units are being read. This would be problematic as the throughput for reading the sensors one at a time would be astronomical compared to having enough channels to read each HC unit.

5.3.2.1Breadboard Test

As our group is still awaiting delivery of this specific component, breadboard testing will be conducted upon receipt of the part. The component is illustrated below in Figure 27.

5.3.2.2Schematics

A schematic for the sensor board is provided below in Figure 27.

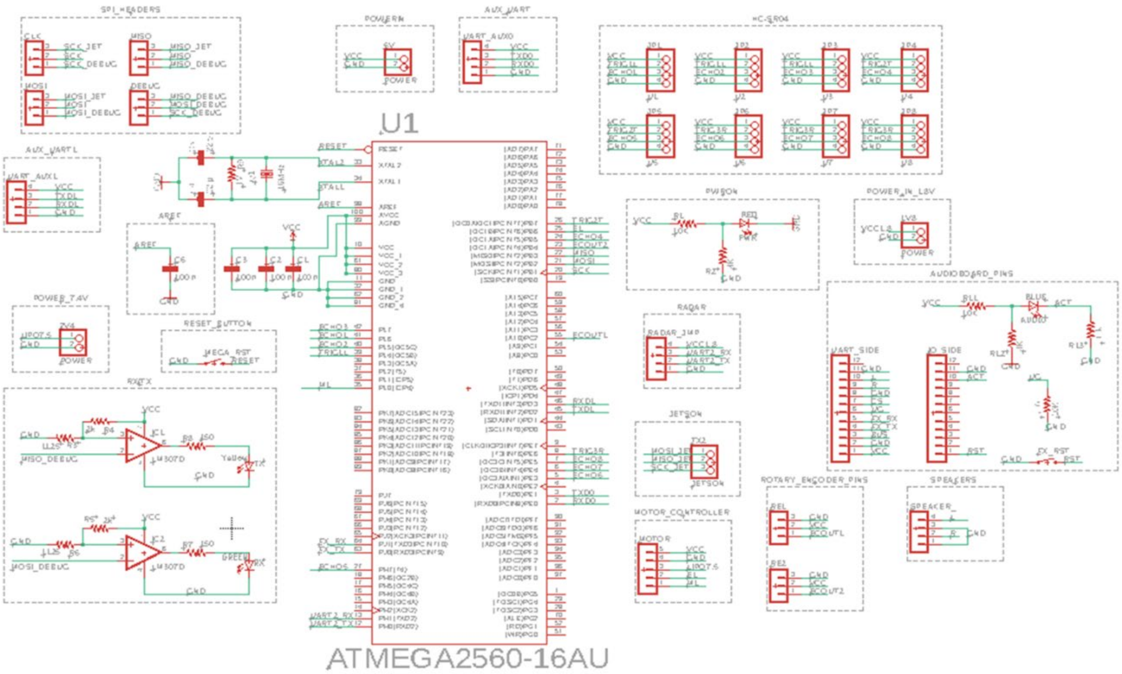


Figure 27 - Sensor Board Schematic

5.3.3 Failsafe

The proposed failsafe mechanism will incorporate the RF remote which was included with the sponsor-provided vehicle platform. The 2.4 GHz RF controller and receiver will allow a remote user to take control of the vehicle platform in the event of a “runaway” scenario in which a fault may cause the vehicle platform to accelerate or steer erratically. This will enable the project to maintain safety standards of autonomous vehicles as discussed previously in section 4.1.4.

5.3.3.1Breadboard Test

As the remote controller is wireless and the 2.4GHz receiver is integrated into the vehicle platform, a breadboard test is not feasible. However, an operational test will be conducted in order to ensure proper remote function of motor and steering control in the event of a need to take manual control of the autonomous vehicle The Traxxas 6519 2.4 GHz receiver is illustrated below in Figure 28.

Figure 28 - Traxxas 6519 2.4GHz, 3-Channel Receiver

5.3.3.2Schematics

Two channels will be utilized for the remote-control failsafe feature of the autonomous vehicle. One channel will be utilized for drive motor control and one will be utilized for steering control. The third channel will be utilized to transmit a “kill” signal in the event of erratic acceleration or steering due to any unforeseen issues with the autonomous vehicle’s operation. The sponsor provided 2.4GHz remote is unfortunately only a two-channel transmitter, so a three-channel remote must be utilized or the team will need to design a simple one channel transmitter to convey the kill command to the autonomous vehicle. The wiring is illustrated below in Figure 29.

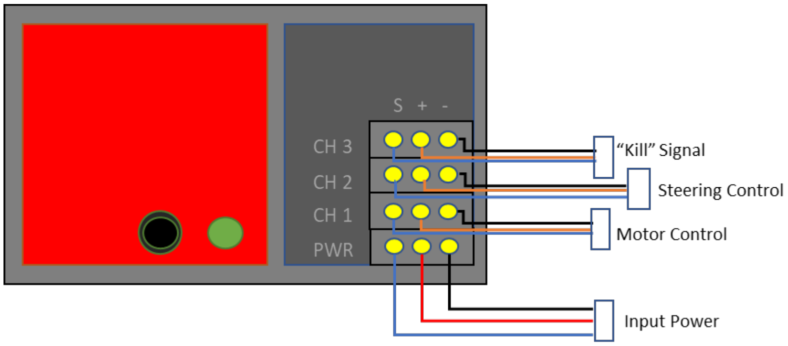


Figure 29 - Traxxas 6519 2.4GHz, 3CH Receiver Wiring Diagram

5.3.4 Audible Safety Device

The importance of the audible safety device cannot be understated. Due to the nature of silent operation inherent with electric motors and autonomous vehicles, it is essential to provide some measure of safety apparatus to prevent unwanted injury. To ensure proper operation of the audible safety device, it was tested via updating the firmware for the module, loading a compressed or uncompressed audio file to the onboard memory and then generating a trigger to play the recorded audio file.

5.3.4.1Breadboard Test

Power provided for the test was supplied by a micro USB connector. Supplied voltage was 5V. The test setup is illustrated below in Figure 30.

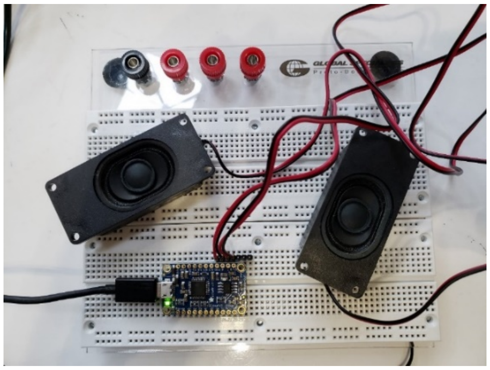


Figure 30 - Adafruit Sound Board Breadboard Test

5.3.4.2Schematics

The schematic for the Adafruit Audio FX Sound Board is illustrated below in Figure 31.

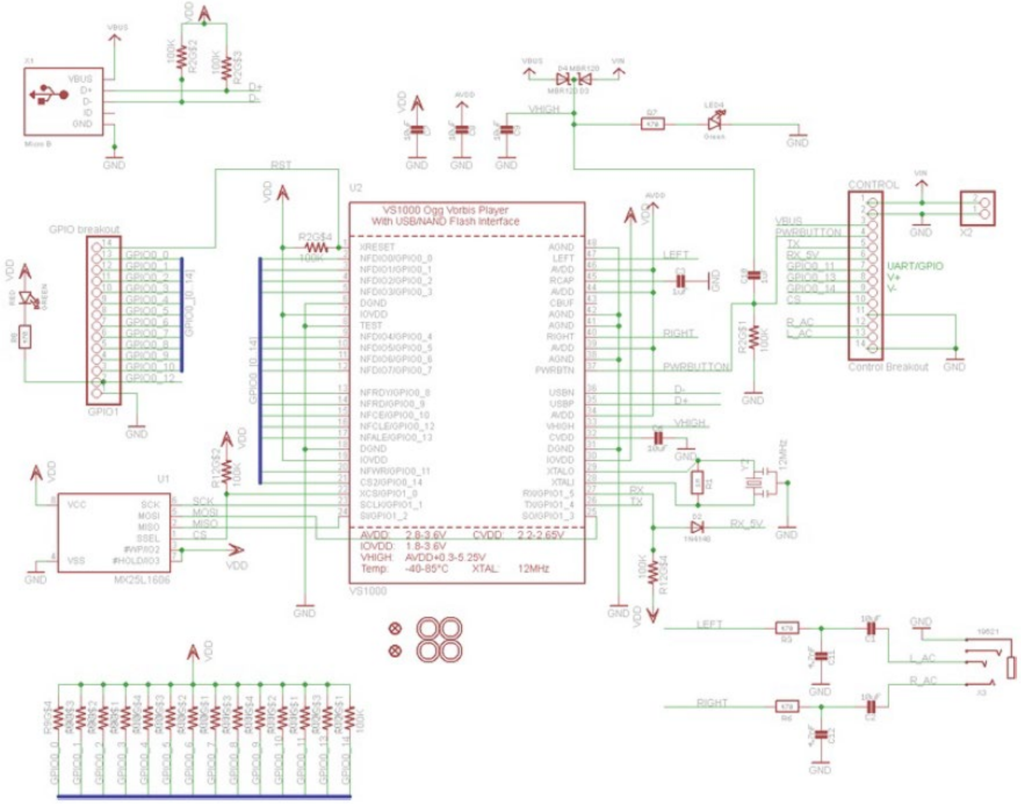


Figure 31 - Adafruit Audio FX Sound Board Schematic

5.4 Peripherals Sub-System

The peripherals sub-system is comprised of the sensors that will provide data to the MCU to interpret obstacles, facilitate course navigation and collision avoidance. The peripherals subsystem can be thought of as the “eyes and ears” of our autonomous vehicle by providing sensory input related to factors external to the autonomous vehicle (i.e. object detection). This subsystem will also be the only subsystem that does not require a PCB to be designed for integration. Even so, specific mounting hardware and wiring will be required to ensure safety, fit and proper operation of all components contained within the subsystem.

5.4.1 3-D Stereoscopic Camera

The ZED 3-D stereoscopic camera is the main peripheral sensor that will be primarily used for course navigation and object recognition. Due to the importance of this component and its requirement for an unobstructed view of the course ahead, the camera will need to be mounted at the tallest point of the vehicle platform. A custom mount will need to be created to attach the camera to the rear shock tower where the vehicle platform Lexan body would normally mount. The specific component placement is illustrated below in Figure 32.

.

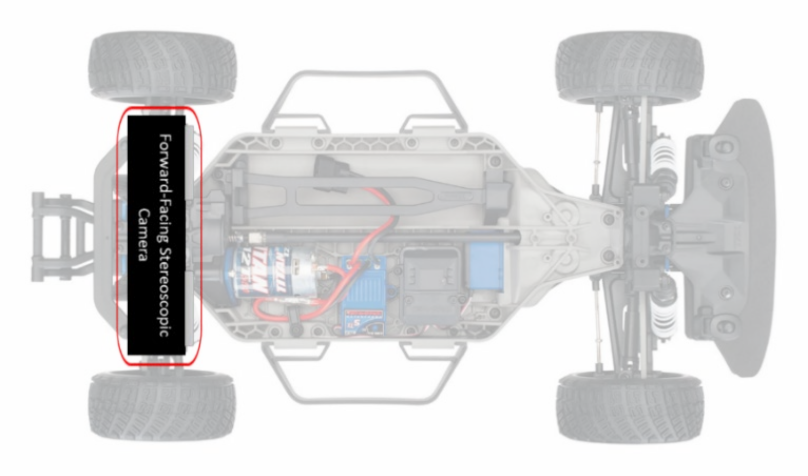


Figure 32 - Stereoscopic Camera Placement

5.4.1.1Breadboard Test

Due to the physical dimensions of this component, a breadboard test is not feasible, but a functional test will be conducted. The component is illustrated below in Figure 33.

Figure 33 - ZED 3-D Stereoscopic Camera

5.4.1.2Schematics

Schematics for the ZED camera were unavailable via research. The team reached out to the manufacturer but due to the proprietary and sealed nature of the camera, the team is not hopeful of receiving schematics. The camera itself is a self-contained unit without any user serviceable parts. The component is also fairly pricey, so the team has decided not to disassemble the component to reverse engineer it.

5.4.2 Radar Proximity Sensor

Specific placement of the radar module on the vehicle platform is illustrated below in Figure 34.

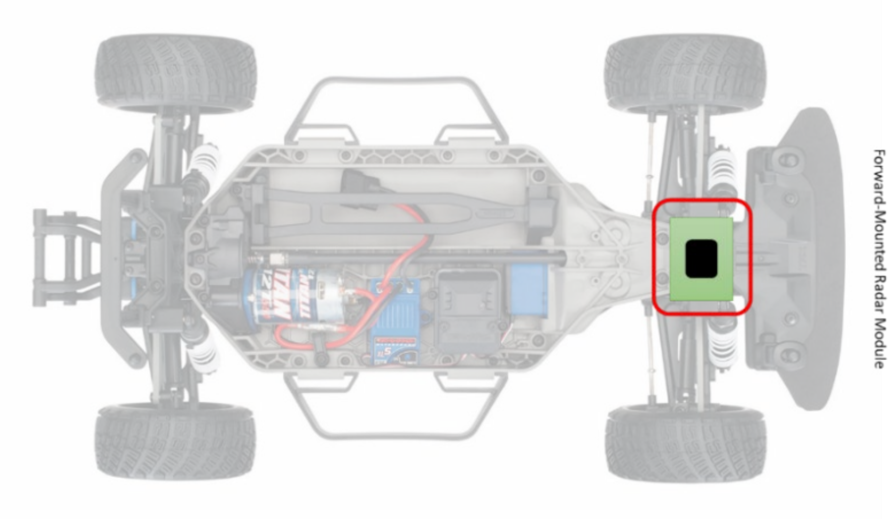


Figure 34 - Radar Proximity Sensor Placement

5.4.2.1Breadboard Test

The radar module was tested to ensure proper ranging and object detection. The breadboard testing is illustrated below in Figure 35.

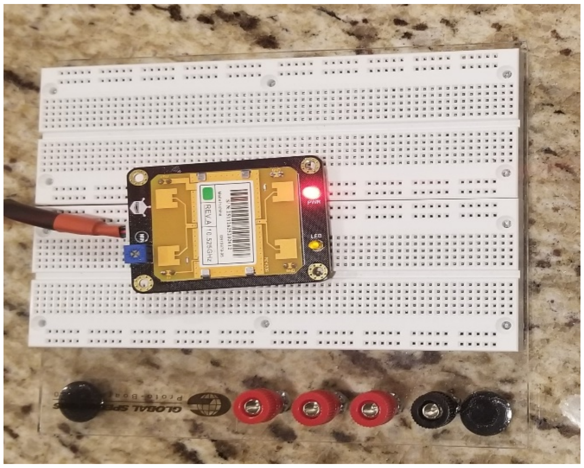


Figure 35 - Radar Module Breadboard Test

5.4.2.2Schematics

The radar module is comprised of a three-layer board and associated components. Schematics illustrating the three layers are given below in Figure 36, Figure 37 and Figure 38.

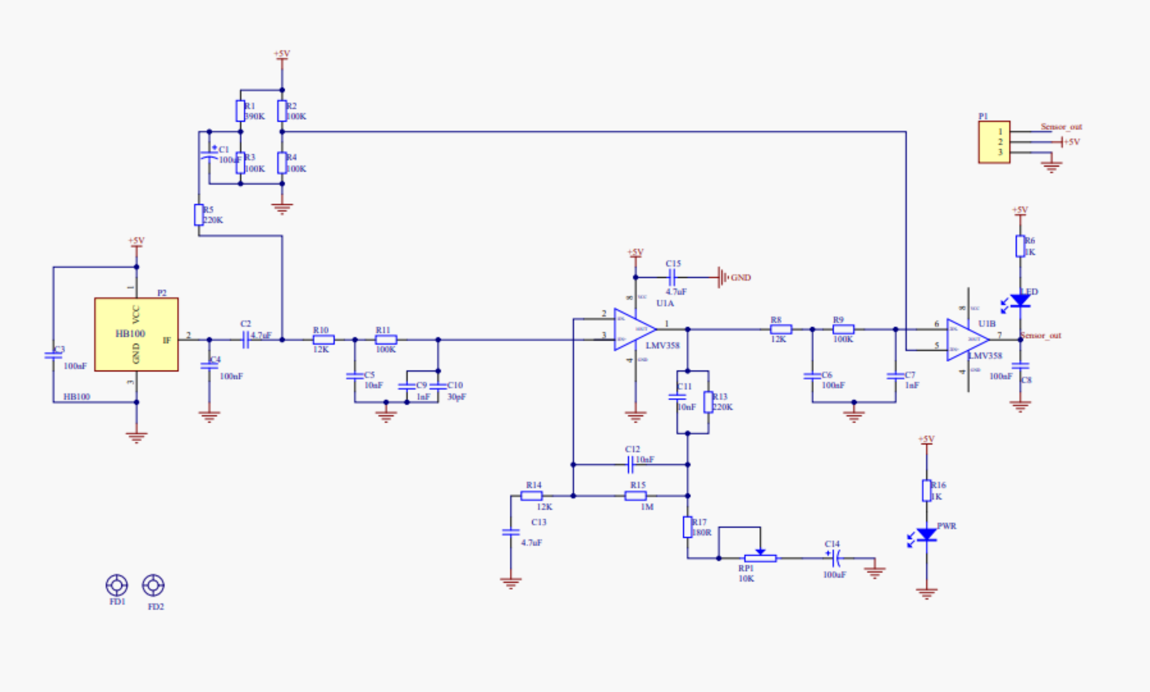


Figure 36 - Radar Module Layer 1

5.4.3 Ultrasonic Proximity Sensor

Specific placement of the proximity sensors on the vehicle platform is illustrated below in Figure 39.

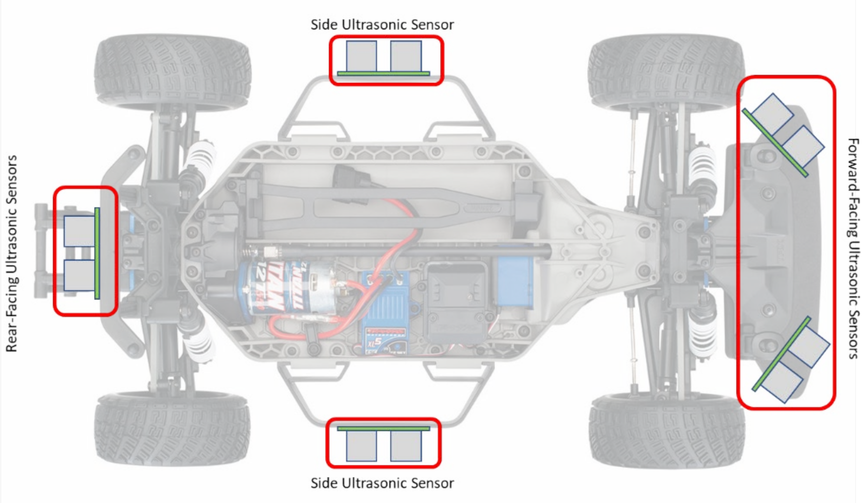


Figure 37 - Ultrasonic Proximity Sensor Placement

5.4.3.1 Breadboard Test

The ultrasonic proximity sensors were tested to ensure proper ranging and object detection. The breadboard testing is illustrated below in Figure 40.

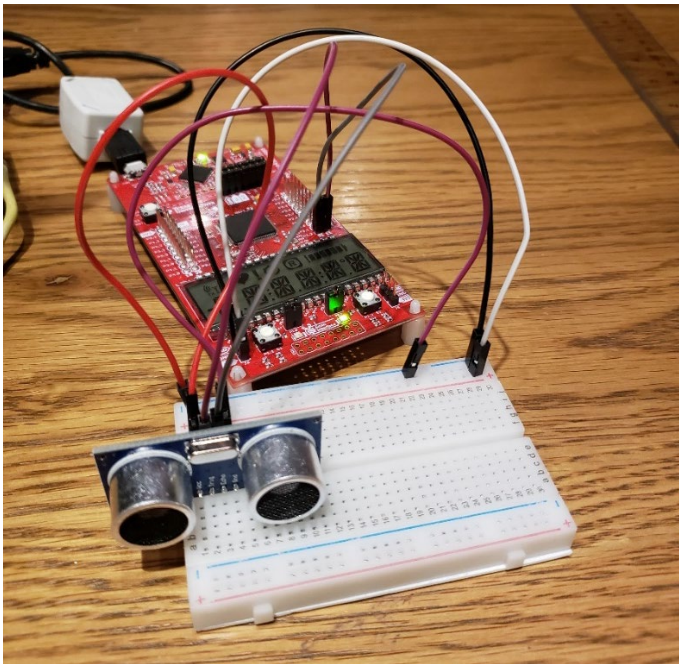


Figure 38 - Ultrasonic Proximity Sensor Breadboard Test

5.4.3.2Schematics

The schematic for the ultrasonic proximity sensor circuit is illustrated below in Figure 41.

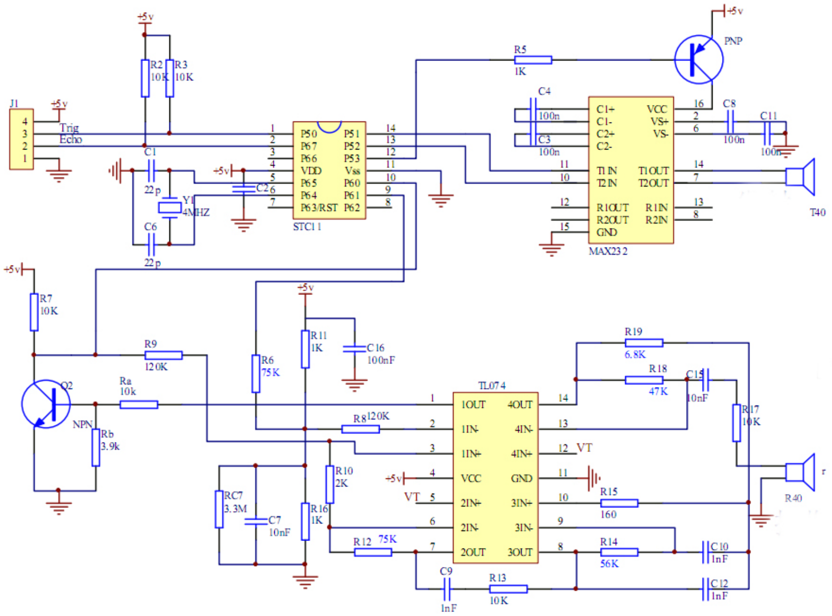


Figure 39 - Ultrasonic Proximity Sensor Circuit Schematic

5.5 Software Design

To create a good software design, research was done to achieve the desired set of objectives. Good software design definitions are summarized in table 15 below and will be utilized throughout the duration of the project:

Table 15 - Software Design Techniques

Maintainability	The ease with which changes can be made to satisfy new requirements or to correct deficiencies. Maintainability is achieved through:	•Clear and Concise Commenting • Condensing Code as Much as Possible • Utilizing Structured Format
Correctness	The degree with which software adheres to its specified requirements	• Having test plans that test all facets of the design
Reusability	The ease with which software can be reused in developing other software.	• Not using functions or IDEs that are only for certain OS
Reliability	The frequency and criticality of software failure, where failure is an unacceptable effect or behavior occurring under permissible operating conditions.	• Code doesn't use excess memory and frees whatever is possible • Code is thoroughly tested for bugs • Code manages variables appropriately
Portability	The ease with which software can be used on computer configurations other than its current one.	• If enough free space is available, more functions can be written that will allow other peripherals or architectures to use our code
Efficiency	The degree with which software fulfills its purpose without waste of resources.	• The order that the code runs on will be no more than O(N)—preferably O(1)

The software of the autonomous vehicle will be that of an embedded system, hence techniques such as reusability and portability will not be considered—since the UCF1/10 team uses a nearly identical set up. Instead the main focus of the software will be that of correctness, reusability, reliability, and efficiency. The project software behavior will be similar to that of a Roomba device, where the end user will only power the device, and then the vehicle will begin to navigate through a track, avoiding obstacles along the way. Since the project will feature different sensors, controllers and multiple processors, UART, SPI and Serial communications will be used throughout the project with as few devices on each protocol as possible. SPI communication will be featured on the ultrasonic and radar sensors due to its fast rate of communication. It is essential to have a high baud rate between the sensors and the MCU because the vehicle will be in constant motion and will need to update its positional data as quickly as possible, which in turn will help process the data that is being collected by the sensors.

The MCU will interpret the data and calculate the distances for any objects that are potentially spotted by each sensor. Once the data is collected the MCU will perform calculations and it will determine if either a threshold is triggered—signifying that evasive actions are necessary—or if the vehicle will be able to operate normally on the same path. Having many sensors in addition to allocating pins to specific sensors will help the efficiency of the software. The software will also feature UART communication that will be solely used for the rotary encoder. The rotary encoder will measure the revolutions of the motor and determine the speed of the car which will be in turn fed to the MCU. This will allow the MCU to determine how fast the vehicle is approaching an object as well as how much time the vehicle has until it will collide. Since UART has error checking, this will make the design be more accurate and efficient in calculating the distance of an obstacle and will also eliminate the potential for packets to be lost or dropped.

Serial communication will be used to communicate with the ZED Stereo camera. This communication protocol is used because the Jetson TX2 features USB ports and it will lessen the size of the software. The ZED Stereo Camera is also compatible with ROS, which will cut down on the amount of programming that we will have to do. The only parts of the ZED Stereo camera that we will need to program will be the minimum required to interface the ZED Stereo camera to the Jetson TX2.

The software will feature different Interrupt Service Routines (ISR) to handle the different evasive maneuvers the vehicle will exhibit. Some of the evasive maneuvers the vehicle could make are stop, produce a warning sound, steer either left or right to various degrees, accelerate and decelerate. Setting up the ISRs as functions will ease debugging the software as the various behavior-controllers will be centrally located within the ISR. This also allows more than one person to work on or to troubleshoot the code if need be.

Stopping the vehicle will need to happen once the vehicle gets to within a certain threshold distance away from an obstacle. The sensors will send the objects' distance data to the MCU which will then determine which action to take. According to the requirements from Table 1, the vehicle will enter the ISR for stop/decelerate/accelerate actions when the sensors detect and object that is three feet away. At three feet the vehicle will commence deceleration while the sensors gather additional data from the surroundings in case the vehicle will also need to turn to avoid the obstacle. If the object does not go below three feet, the MCU will send a command to the motor controller to accelerate, but if the object goes below two feet the ISR will break the connection to the motor controller to make the vehicle come to a complete stop. In addition, the software will feature a failsafe ISR that is intended to be a “kill switch” for the vehicle.

A kill switch is required for the competition and is specified by competition standards where the user will have the ability to stop the vehicle by flipping a switch or pressing a button. The ISR will be triggered by an output signal from the user’s remote control that is compatible with the OEM radio antenna. The end user should call the kill switch when the user sees a flashing LED mounted on the vehicle or if the user hears the warning sound clip. This LED will be triggered by the sensor data when the vehicle reaches the different thresholds. At two feet the LED will begin to flash slowly to let the user know that the vehicle should begin to stop. At six inches the LED—or possibly multiple LEDs—will flash rapidly to let the end user know that the vehicle should come to a complete stop within the aforementioned required time of five seconds. If the vehicle stops then the end user will disregard the waring LED. If the vehicle appears to continue going, then the end user will simply pull the switch trigger from the remote control.

To steer the vehicle in the correct speed and direction, measurements will be taken from the steering controller unit. Based on distance data received from the various sensors, speed read from the rotary encoder (or possibly calculated from changing distance data), and radius of the curve measured from distance data, the vehicle will be able to run an algorithm that correctly decides on which direction or speed to take. One of the main components that will be featured in the navigational ISR is the 3D camera. The camera has the ability to sense object distance and can create a point cloud. Software will be made to parse through the point cloud and generate a “collection” of objects along with their distances and size. Once both size (needed for steering direction) and distance (needed for reaction time) are found, the Jetson will send the data to the MCU. This will trigger the software to enter the ISR. The auxiliary sensors will also contribute data to have a more accurate calculation of the distances and will effectively act as a handshake with the ZED Stereo Camera. Once the path is determined, the MCU will send a command to the steering control to turn to the appropriate angle.

As a safety feature it was decided to include an audible device to make the vehicle stand out. Different sounds will be programmed when the vehicle performs an evasive maneuver, or to indicate different operation statuses. The vehicle will produce a warning sound when approaching an obstacle, when it has come to a complete stop, when the vehicle goes in reverse, as well as if any sensor or car component fails. This ISR was decided to be separate from the stop/accelerate/decelerate function to reduce the size of the function and ease of debugging, but the MCU will still be able to control the speed separately from the ISR.

5.6 Summary of Design

To summarize and provide a functional overview of the project, the design will be broken down into power flow, signal flow, programming overview and operational summary. Each component of the summary will include a basic walkthrough of the specific function as well as an overview of the intended inputs and outputs. When summarizing the design, it was decided to limit the amount of hard data that can be gleaned from specific sections pertaining to the operational process. Diagrams were created to reinforce the summary of the specific operational factors.

5.6.1 Power Flow

Input power will be supplied by a rechargeable Lithium Polymer battery supplying 11.1VDC at 5800mAh. This voltage and current will be routed to the power systems PCB where the 11.1V will be directed to three linear voltage regulator circuits that will convert the 11.1V input to 1.8V, 3.3V and 5V respectively. The 11.1V will also be routed through the power system PCB as an unregulated input that will be directed to the motor controller to provide voltage and current to the drive motor and to the image processor. The power system PCB will output 1.8V, 3.3V and 5V to the main payload PCB which contains the MCU and safety circuits. The MCU will utilize the 3.3V to power itself. The 1.8V, 3.3V and 5V voltages will also pass through the main payload PCB to be directed to the peripherals sub-systems.1.8V will be routed to the radar module. 5V will be directed to and utilized by the ultrasonic proximity sensors and 3-D stereo camera. System wide power flow is illustrated below in Figure 42.

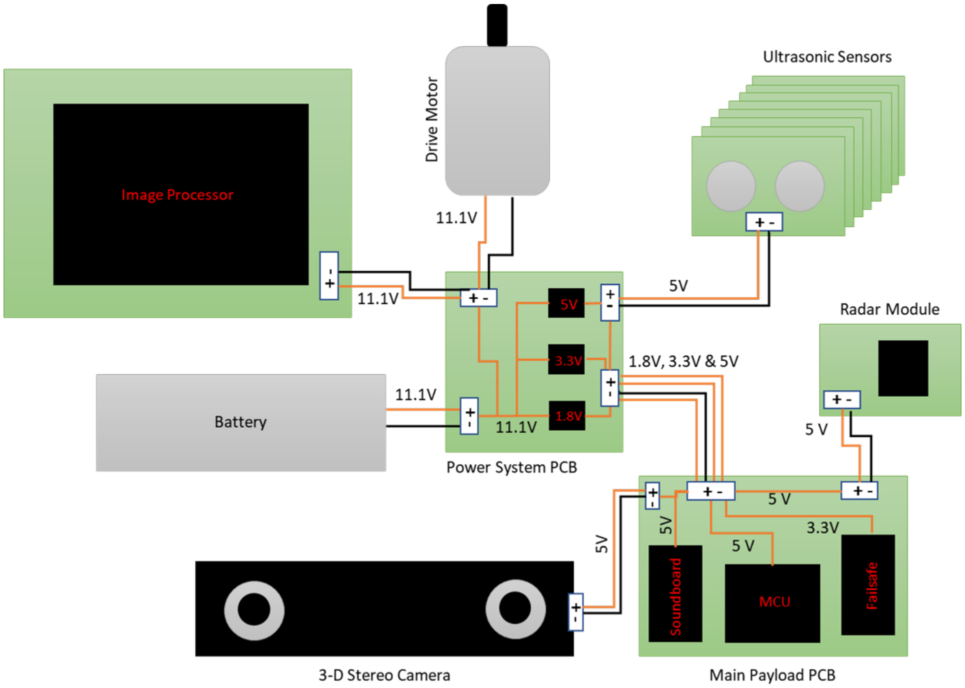


Figure 40 - Project Power Flow Diagram

5.6.2 Signal Flow

Raw image data will be transferred from the 3-D stereo camera to the image processor. Rectified image data will then be passed to the MCU. The MUC will also accept proximity data from the ultrasonic proximity sensors and the radar module to facilitate object detection. The MUC will then determine motor and steering output signals to facilitate collision avoidance. Signal flow throughout the system is depicted below in Figure 43.

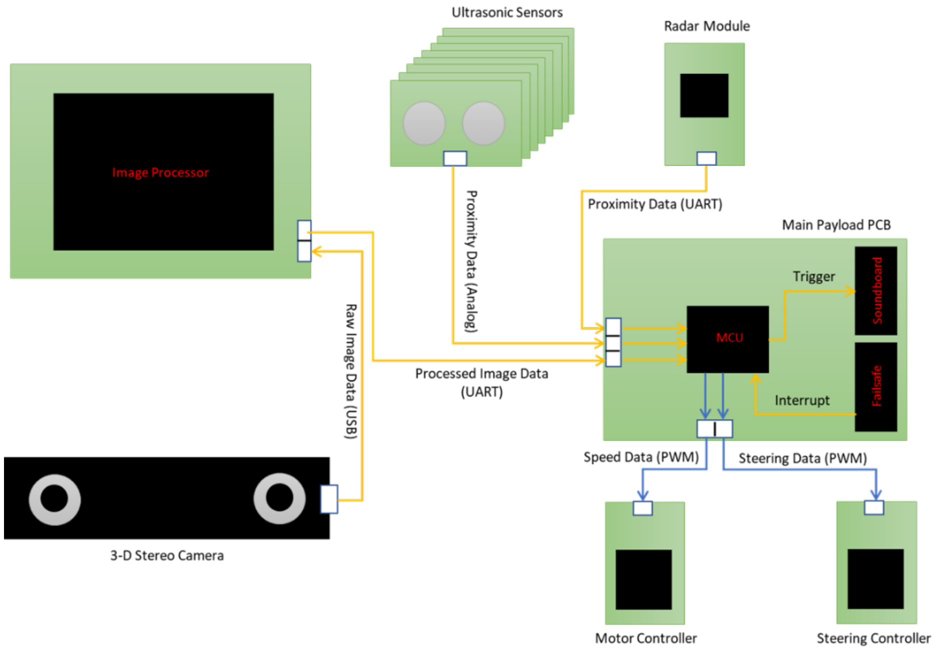


Figure 41 - Signal Flow Diagram

5.6.3 Programming Overview

Programming for the project will be two-fold. First, the Jetson NTX will need to be programmed to interface with the camera and MCU as well as to convert raw image data into rectified data that can be utilized to spatially recognize objects and facilitate course navigation input data to the MCU. The specific process for this programming is summarized in section 6.3 Final Coding Plan. Second, the chosen MCU will need to be programmed to interface with multiple peripheral sensors that operate with multiple communication protocols. The MCU will also need to be programmed to output motive and steering data signals that will be utilized by the motor control and steering control to facilitate course navigation and object avoidance. The MCU will also need to be programmed to randomly trigger the audible safety device in order to notify nearby pedestrians of the operation of the autonomous vehicle for safety.

5.6.4 Operational Summary

The basic operational summary of the system is as follows. The 3-D stereo camera, being the primary sensor, will utilize its stereoscopic vision system to detect and determine depth of images in order to measure distance from detected objects at longer ranges and to facilitate course navigation. The radar module will send proximity target data to the MCU in order to detect mid-range distance objects that are in the path of the autonomous vehicle. Ultrasonic proximity sensors will send data to the MCU in order to facilitate minimal range object detection. This data will be fed to the MCU in order to determine whether a motor or steering signal is required each cycle. The basic operations of the autonomous vehicle and its functional process is depicted below in Figure 44.

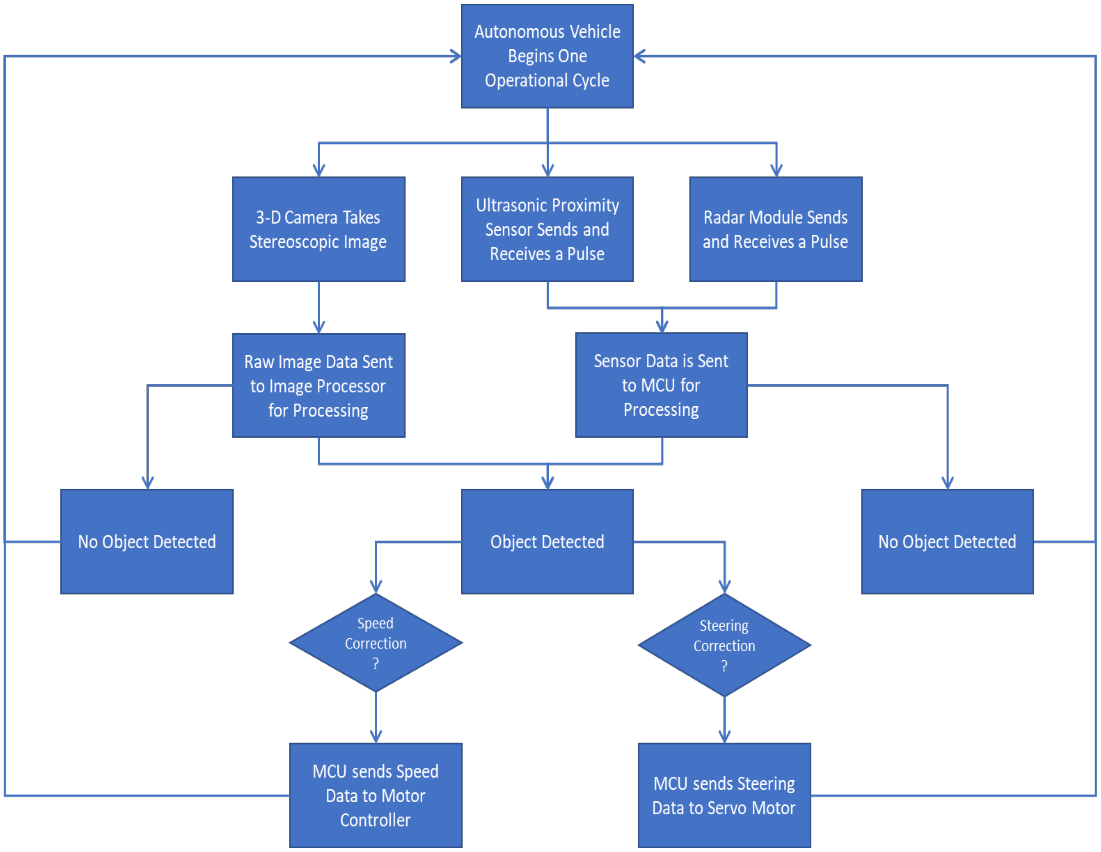


Figure 42 - Basic Operational Process Diagram

6.0 Project Prototype Construction and Coding

Once parts are received, construction of a prototype and initial coding for the image processor and MCU will begin. With the two potential MCUs chosen, a brief comparison may be made, although this should not be taken as an absolute decision for the MCU as problems could arise during the prototyping phase and changes may need to be made.

Additionally, mounts will need to be fabricated for the various peripherals as well as the PCB's themselves. These mounts will either be prototyped in CAD and then 3-D printed or an existing mounting scheme will be devised, researched or utilized. A custom mount runs the risk of decreasing manufacturability once the autonomous design is completed. This could be detrimental to our sponsor’s goal of integrating our design with an existing, parallel configuration. To minimize this possibility, all reasonable effort will be applied towards designing and/or utilizing a mounting solution that is commonly available or minimally difficult to replicate with facilities available to our sponsor.

A development board will most likely need to be purchased in order to program our MCU. In most cases, the MCU on the PCB will have to be bootloaded via another chip that has access to USB communication to our IDE. While this does pose as an extra intermediary step and will be less efficient than including a USB port on our PCB that can directly program the MCU through the JTAG pins is unnecessary, and in practice can allow other potentially malicious users to get their code into our device.

6.1 Integrated Schematics

Integrated schematics will be constructed to show power, data and signal interconnections between the three sub-systems. The PCB design for the MCU is shown below in Figure 45 to illustrate the layout for the MCU and the various sensors.

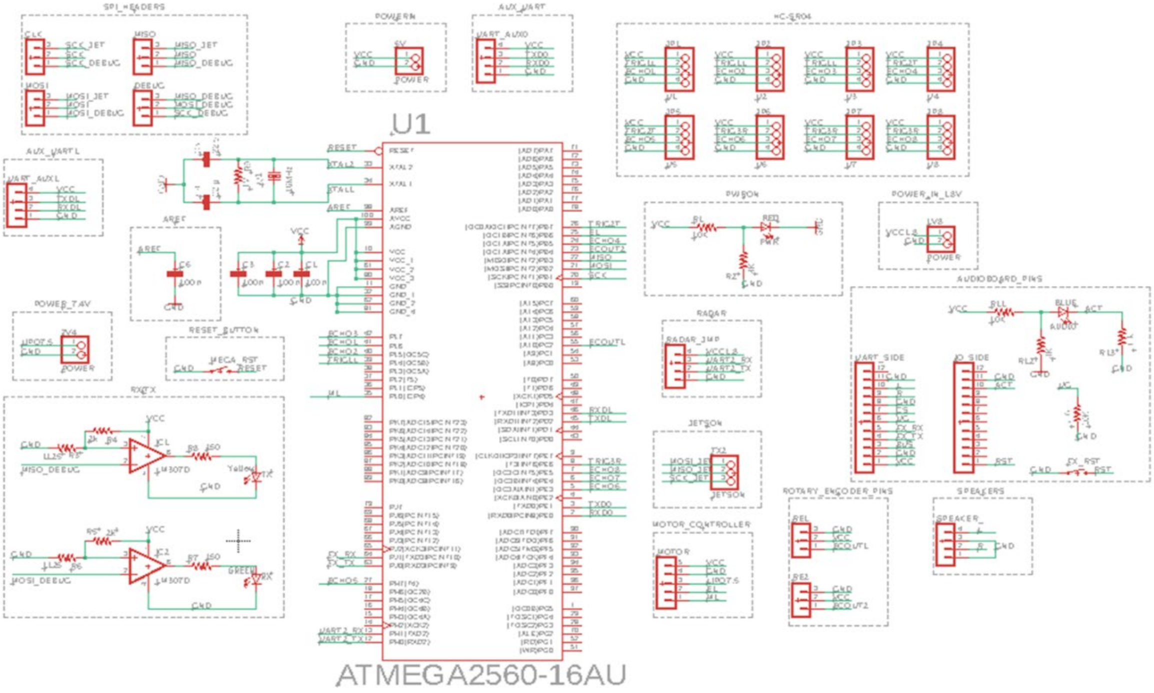


Figure 43 - MCU Schematic

The Jetson Tx2 will also have the ZED Stereo Camera in concordance with multiple sensors, in addition to having multiple connectors to the power supply, MCU, and radar subsystem. The following schematic in Figure 46 showcases the initial layout for the Tx2.

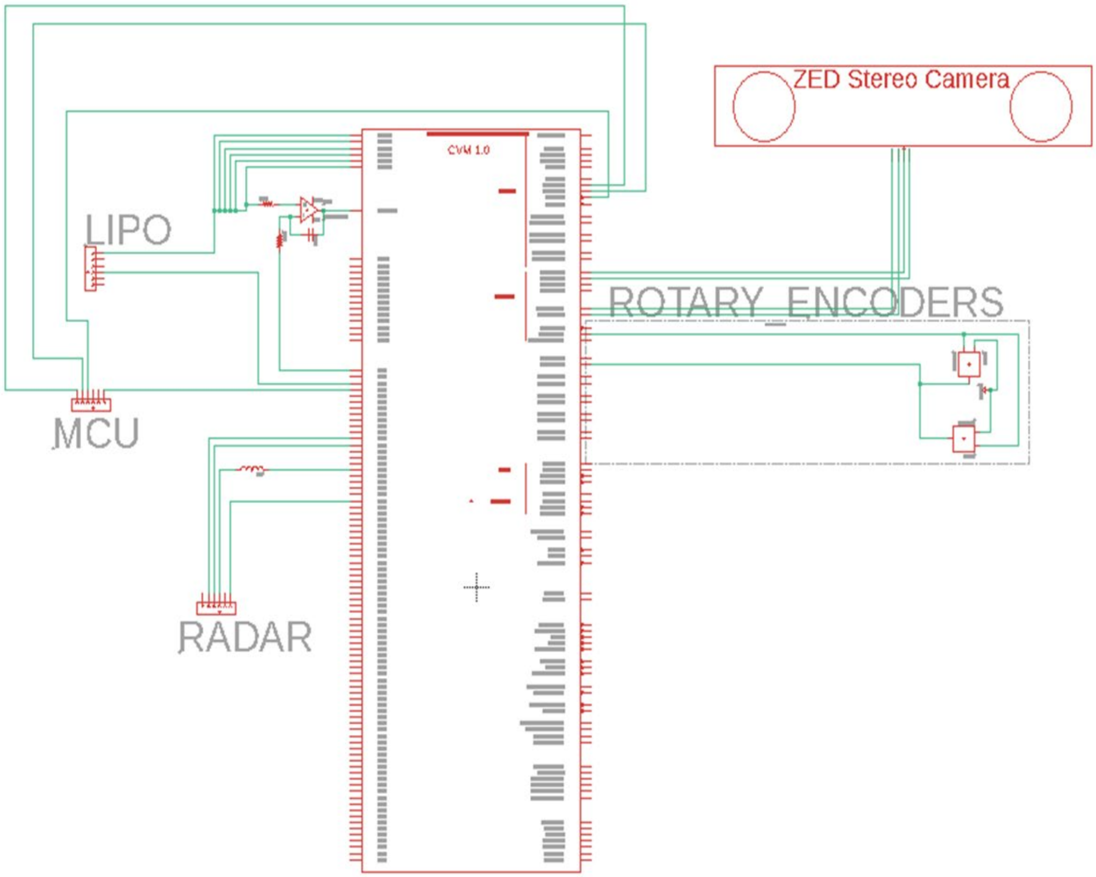


Figure 44 - Jetson TX2 and Peripherals Schematic

6.2 PCB Vendor and Assembly

The chosen PCB vendor will be PCBLayout.com and once final PCB designs are completed, an order will be placed to have the PCBs constructed. Team intention is to provide enough lead time in the design and implementation process to ensure that PCBs are received with enough time to solder and test designs before project deadline.

6.2.1 Power System PCB

The Power System PCB was designed with Eagle and the appropriate design files were sent to JLCPCB in China for manufacturing. Luckily, our team was able to design and have our v1.0 Power System PCB manufactured before the Coronavirus prohibited receiving PCB's from the supplier. The illustrations of the completed Power System PCB are provided below in Figure 47, Figure 48 and Figure 49.

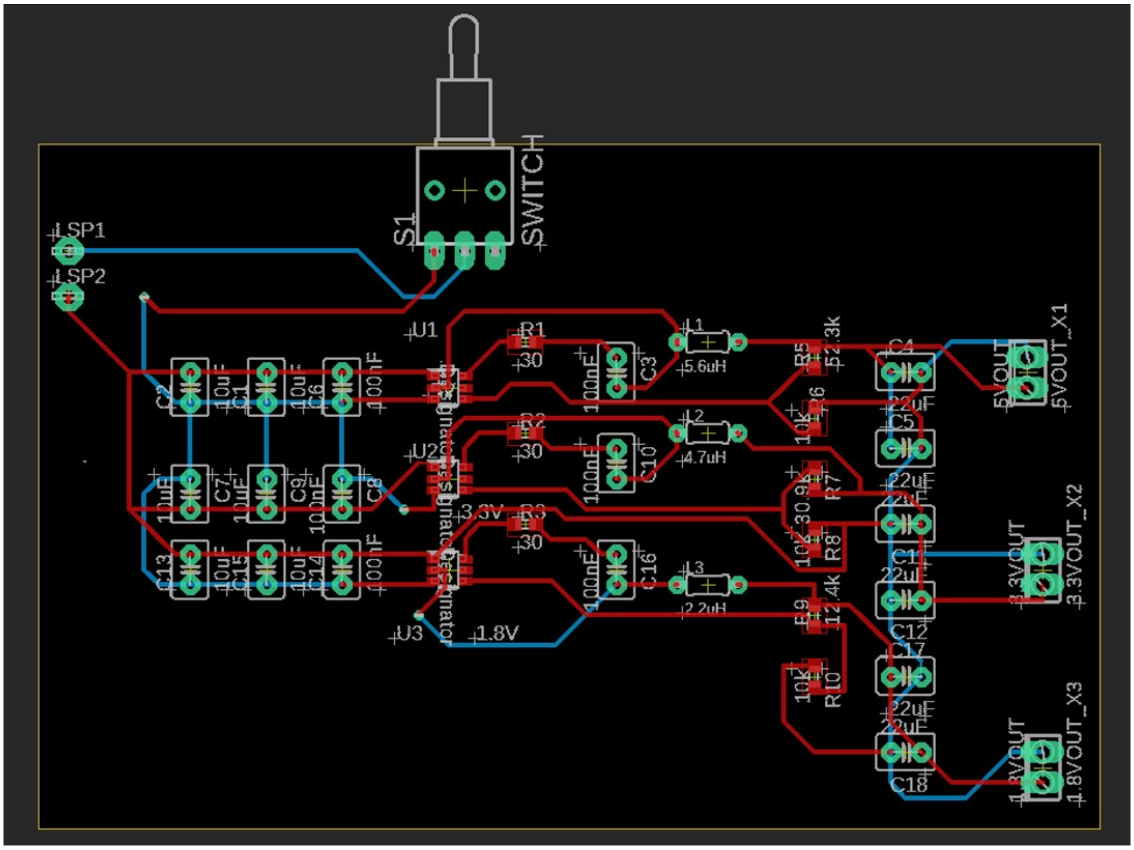


Figure 45 - Power System PCB Eagle File Diagram

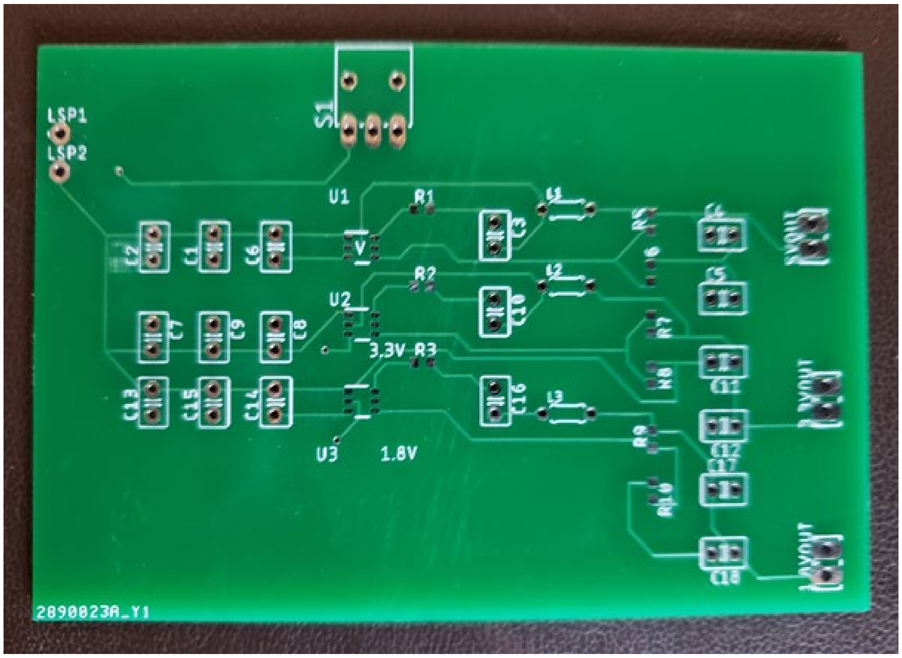


Figure 46 - Power System PCB Production Board (Unpopulated)

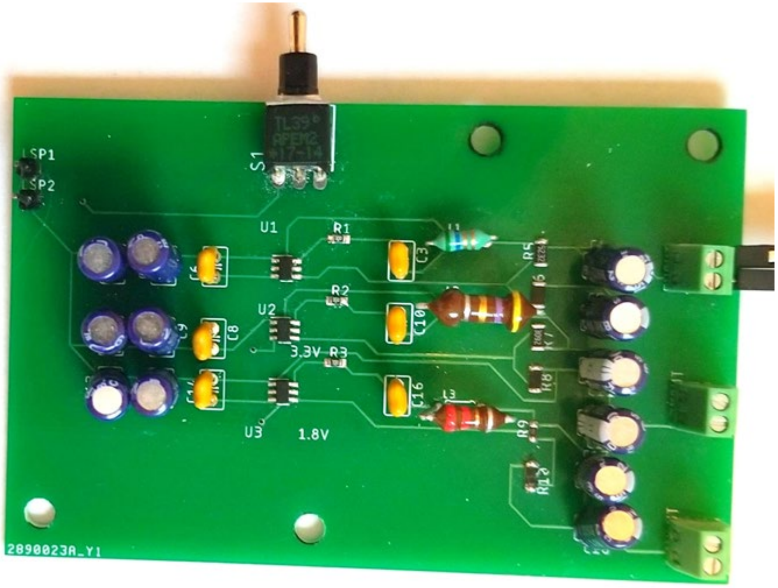


Figure 47 - Power System PCB Production Board (Populated)

6.2.2 Sensor Board PCB

The illustrations of the completed Sensor Board PCB are provided below in Figure 50, Figure 51, and Figure 52.

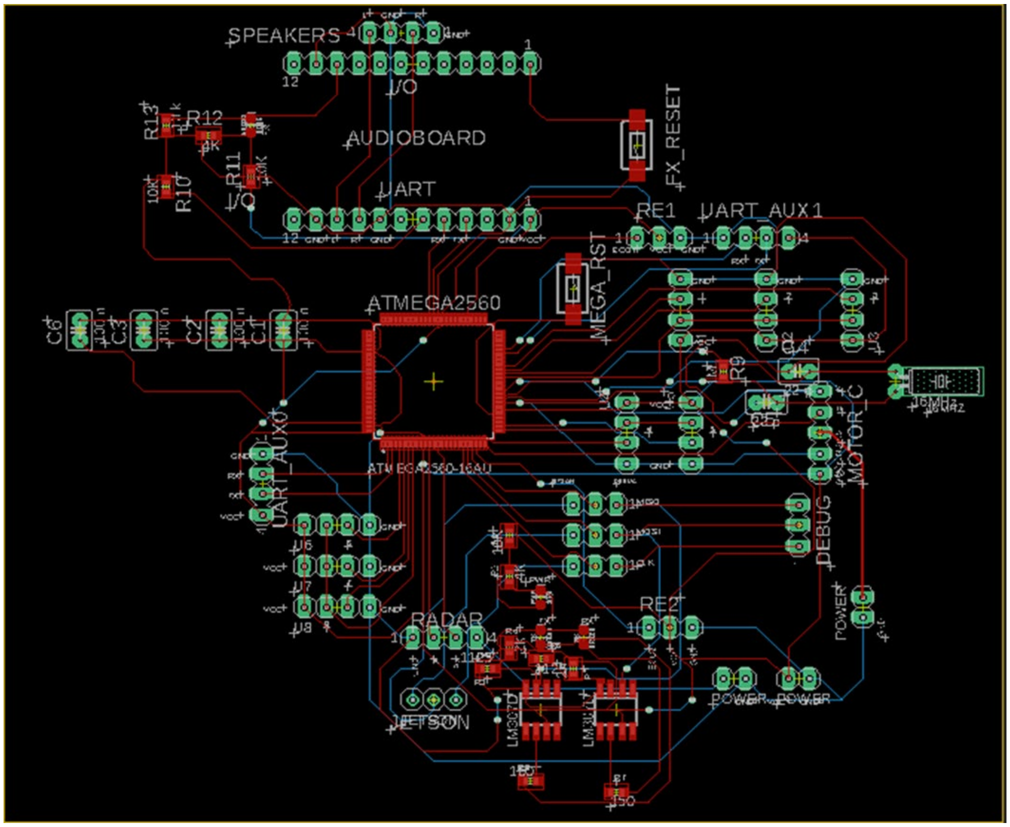


Figure 48 - Sensor Board PCB Eagle File Diagram

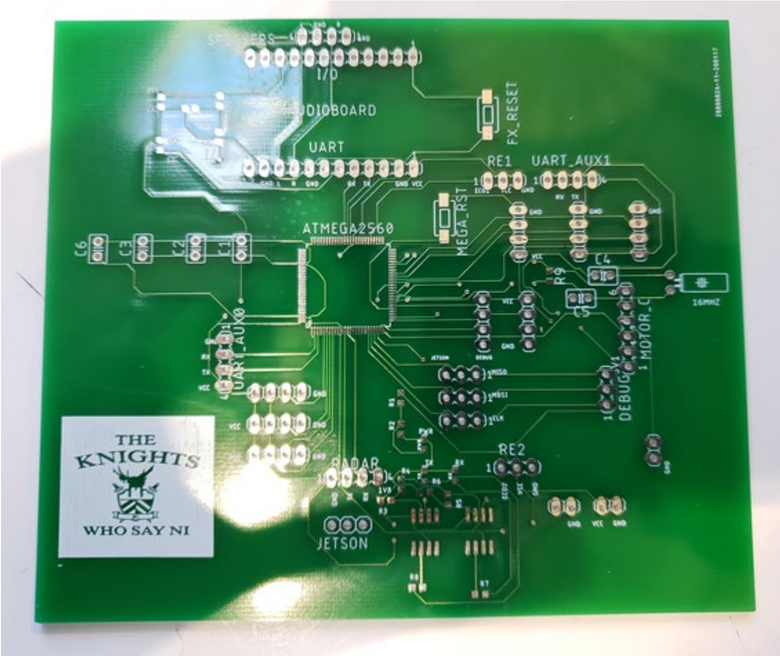


Figure 49 - Sensor Board PCB Production Board (Unpopulated)

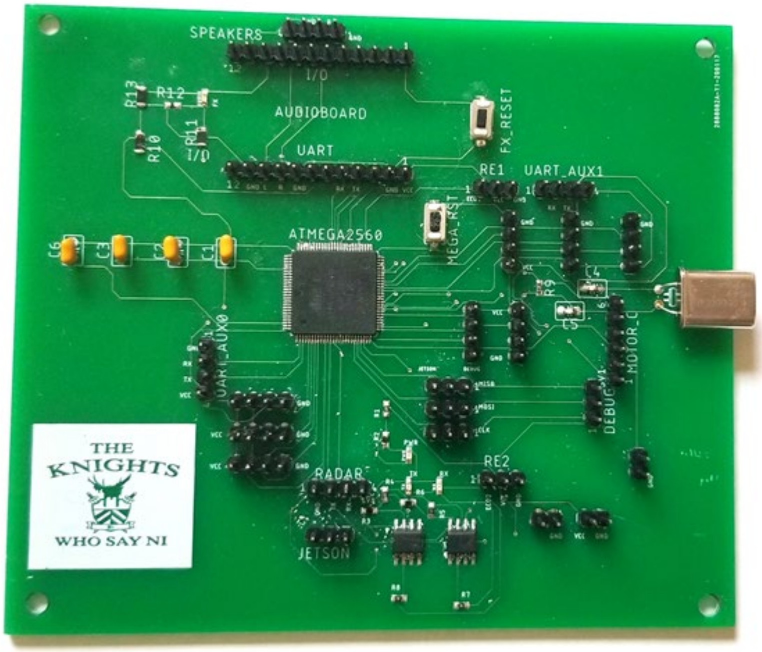


Figure 50 - Sensor Board PCB Production Board (Populated)

6.3 Final Coding Plan

For our project, only at most two modules will need to be programmed: the microcontroller that we choose and optionally the Jetson Tx2. The Jetson Tx2 is optional as the UCF1/10 team is currently using the Jetson, so it would be easier for the other team to incorporate our designs more seamlessly. The Jetson would also be assistive due to the fact that we can run ROS on the Jetson, not to mention that the Jetson has an incredibly powerful processor that can parse through and analyze the sensor data more efficiently. ROS is a framework that works with an operating system that is independent of a language and can be easily implemented into any modern programming language such as: Python, C++, Lisp, as well as having experimental libraries in Java and Lua. Because ROS was designed to be as thin as possible, it provides facility to use ROS with other robot software frameworks. ROS was selected as the main programming framework for its ease of testing via. The idea here is that the project can be accomplished (without the use of an RTOS, however) with our microcontroller of choice, but the processor would be less powerful, would have less memory, and would not be ROS-compatible.

As far as the programming language that will be used in this project, a few different options are presented. If our group ends up using the Jetson Tx2, the ROS programs will be written in Python, as it is much simpler to learn than C++, which is the only other native programming languages for ROS. Although it would be possible to include packages that allow for other programming languages, like Java, it is much simpler to go with Python and have many examples and documentation available to us. Python’s high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python offers a large library and modules that make just about every programming project realizable, and this library availability will make our lives much easier with this assignment. The basic algorithm is provided below in Figure 59.

Putting the flowchart into words, the car will first figure out if it is moving, or if it has stopped. If it is moving, it will adjust the speed and check for obstacles, where object avoidance comes into play. A boundary of 2 feet has been selected as the safe distance for our project. The car will determine—using the various sensors—if an object is at least two feet away before it will try to avoid it. In the alternate case where the car is stationary, two initial conclusions can be made: either the car has just started up and is running for the first time, or it approached a wall or another obstacle and had to stop to avoid a collision. In order to ply between these two scenarios, a quick systems check/debug run occurs, and then once the sensors are checked to determine it is safe to travel, the car will proceed.

The microcontroller will be the more difficult piece to program, as different microcontrollers will require different software and firmware to program, and until a firm selection is made it will be unknown what language or software will be needed to implement our design. Given the group's previous experience, C, Python, and Java will be preferred over any other language. If the MK20DX128VFM5 is the microcontroller that the team goes with, the S32 Design Studio is an IDE that is provided by NXP to program their ARM processors. This IDE uses a C-programming scheme and would be easy for the group to use. The flowchart below in Figure 53 provides a scope of how the program will function; it was created to be simple enough that anyone can see how the heart of the program operates.

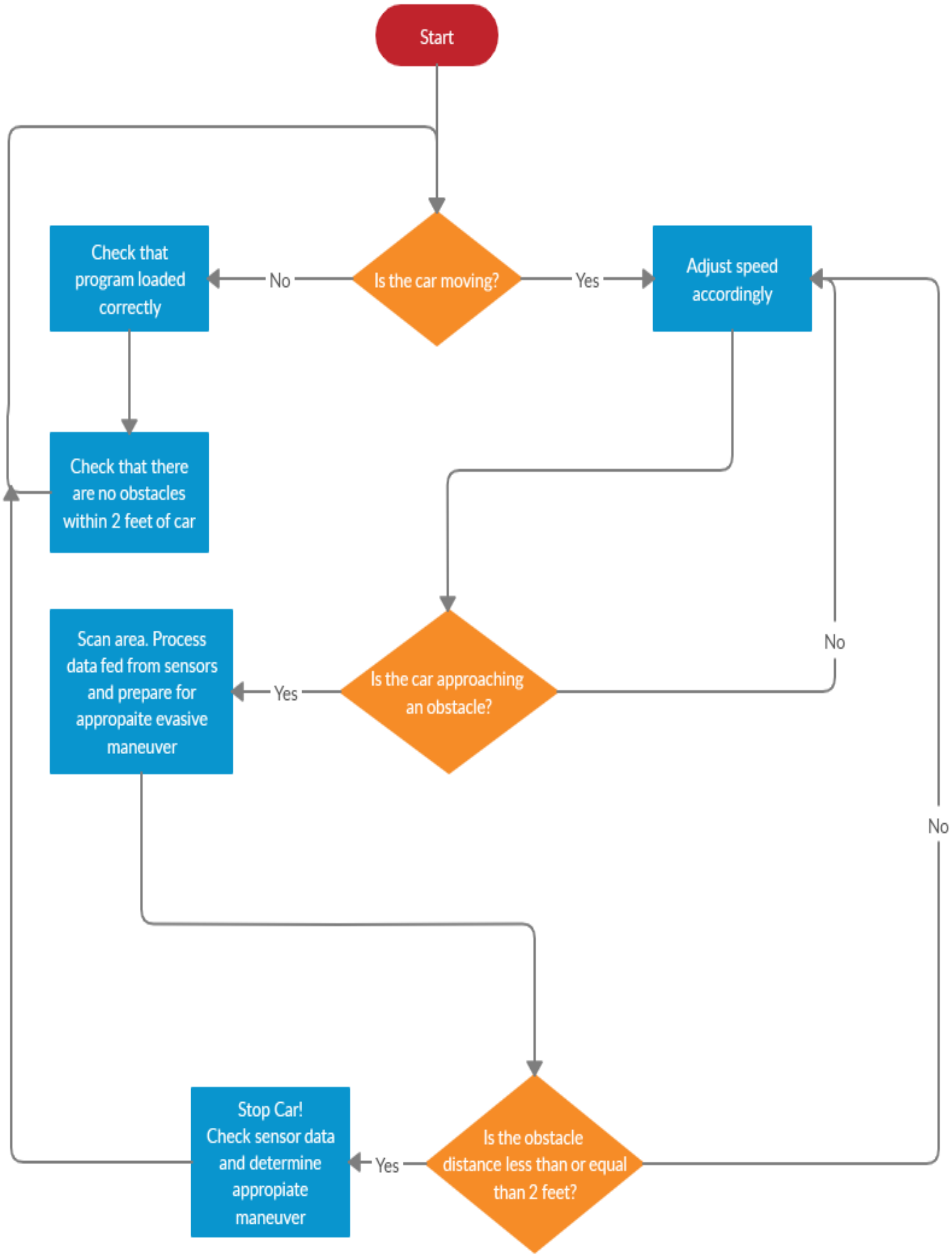


Figure 51 - Basic Algorithm flowchart

The code algorithm is based on the different sensor the vehicle will have. Each sensor will have a different threshold that will send a command telling the vehicle what action to take. For example, the ultrasonic sensor threshold will be at around three feet and at three feet the vehicle will begin to slow down. The threshold for radar will be at around two feet and at two feet the vehicle will come to a complete stop. The 3D camera will be programmed to recognize objects and provide navigation facilities. Since the vehicle will be in constant motion, an infinite loop was established to keep data flow constant.

7.0 Project Prototype Testing Plan

The autonomous vehicle consists of two major groups: Hardware and software. Each group was tested separately and then combined to ensure proper function. Hardware testing was done by setting predesign parameters -i.e. – engineering requirements – to verify functionality. Software design was done by using bench test by having predesign parameters. Once the hardware and the software were successful, they were merged to recreate the same results. The following sections elaborate more on hardware and software testing.

7.1 Hardware Test Environment

A proper test environment is required in order to test the hardware of the project. Due to the mobile nature and potential speed in which the vehicle platform can travel, a large space is required in order to test the project sufficiently. In a real-world situation with a full-scale autonomous vehicle, the test environment would require a large parking lot or test track. Due to the scaled down version of the project, at one tenth of the scale of a full-sized vehicle, we have determined that we should scale our test environment to roughly one tenth of the scale of an actual full-sized test environment. However, due to size constraints related to our final demonstration, we may need to scale down even further to present a realistic depiction of the vehicle's capabilities.

As such, the hardware testing environment will consist of two settings in order to evaluate acceleration, deceleration, stopping distance, object recognition, collision avoidance and autonomous navigation. To test autonomous course navigation ability and collision avoidance, a reconfigurable course enclosed on two sides by "walls" made of dryer vent tubing will be constructed as pictured below in Figure 54. The test track will be roughly oval shaped with physical dimensions of twenty feet width by forty feet length. The purpose of the course will be to provide a simulated race environment which will require the autonomous vehicle to navigate a course without hitting the side walls. During the final demonstration of the project, the team will need to preferably request an outdoor space in which to demonstrate the full capabilities of the project. Otherwise a scaled down version of the testing will have to be devised in order to showcase the vehicle's abilities as well as to demonstrate its autonomous capabilities. Depending upon available space constraints, the code may need to be adjusted to decrease sensor detection range or potentially some other design concession to facilitate the testing area. An example layout is provided below in Figure 55.

Figure 52 - Duct Tubing Wall for Test Environment

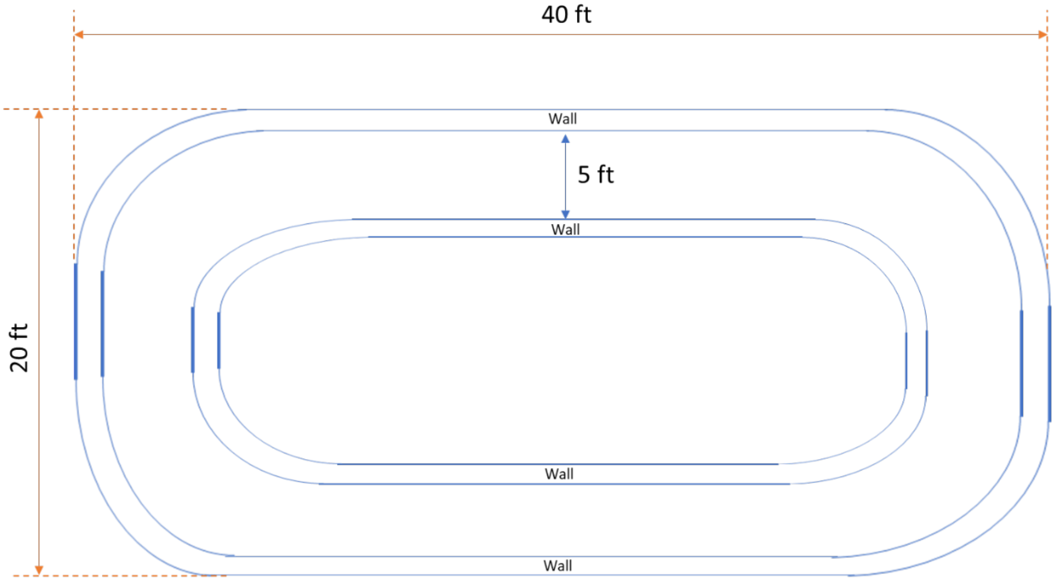


Figure 53 - Autonomous Navigation Test Environment

The test environment devised to test acceleration, deceleration and stopping ability will be an area of roughly twenty feet by five feet rectangular. Dryer vent walls will be placed along either side of the twenty feet length to provide navigational context for the autonomous vehicle as it travels the length of the area. An object will be placed approximately in the position depicted below in Figure 56 in order to provide an obstacle for the vehicle to recognize and interpret as an obstacle that requires deceleration or stopping.

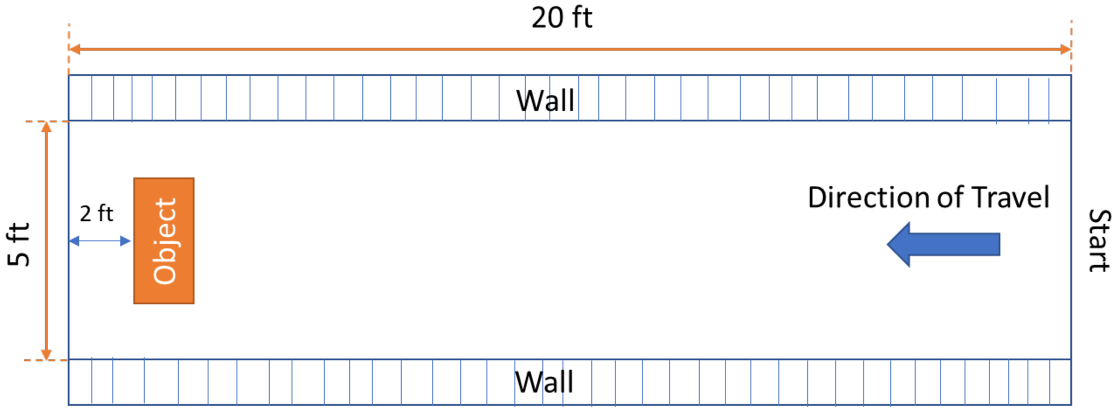


Figure 54 - Acceleration, Deceleration and Stopping Test Environment

7.2 Hardware Specific Testing

To create an autonomous vehicle, the components associated with it must provide reliable data and results to ensure a successful navigation. This is quintessential since the vehicle will be in constant motion. To ensure proper function of the autonomous vehicle, the components were individually tested for precision, accuracy and efficiency prior to being implemented in the system itself. One of the main components in the system are the different sensors it will carry. The ultrasonic sensor HC-SR04 was tested using the MSP430G2 development board for functionality and precision. Code composer studio was used to program the MSP430G2 to provide functionality to the HC-SR04 sensor. This is illustrated in Figure 57 and Figure 58.

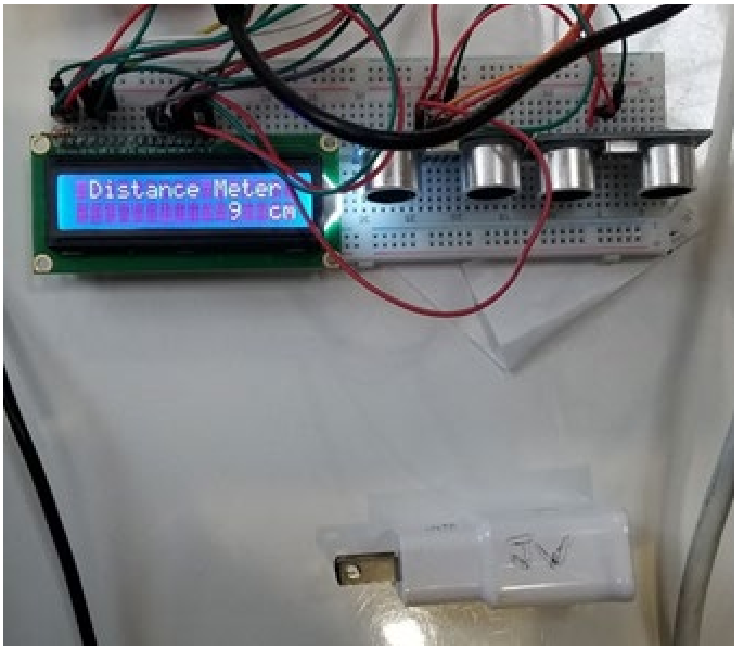


Figure 55 - Ultrasonic distance test 1

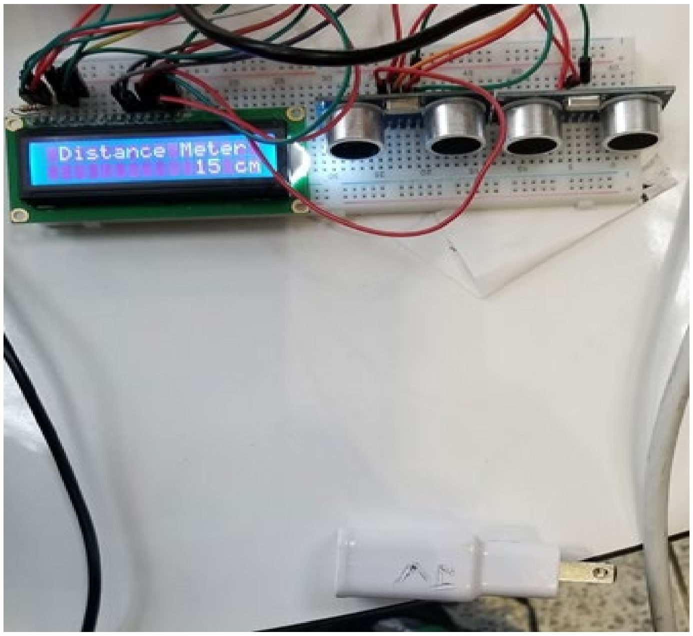


Figure 56 - Ultrasonic distance test 2

The other sensor that will be used by the system is a radar module. The radar sensor was used with the DFRobot SEN0192 radar module. To test this sensor, it was necessary to acquire the necessary Arduino library to test it. The SEN0192 module works with Arduino and the premise for its functionality is based on the Arduino library. Using a test program, the SEN0192 was tested. The Figures below showcase basic radar testing (Figure 59) and object detection testing (Figure 60).

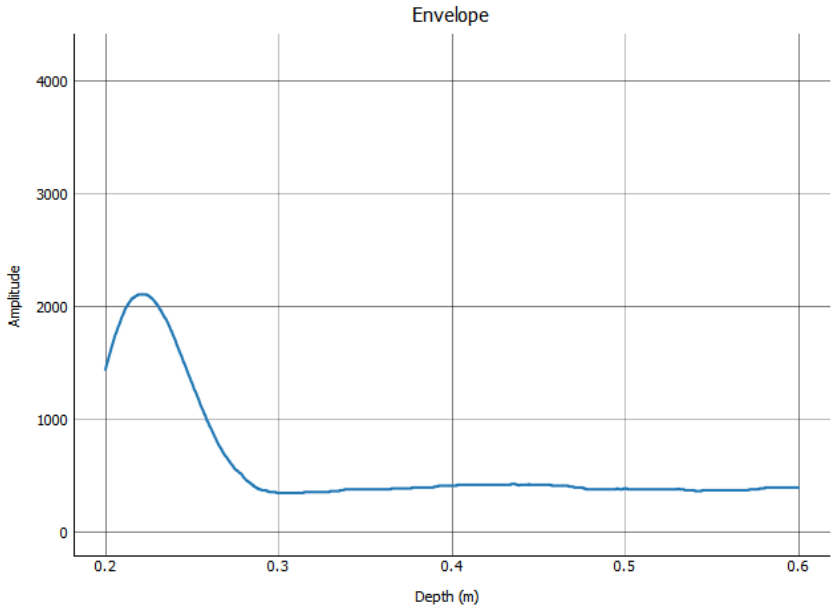


Figure 57 - Radar Module Basic test

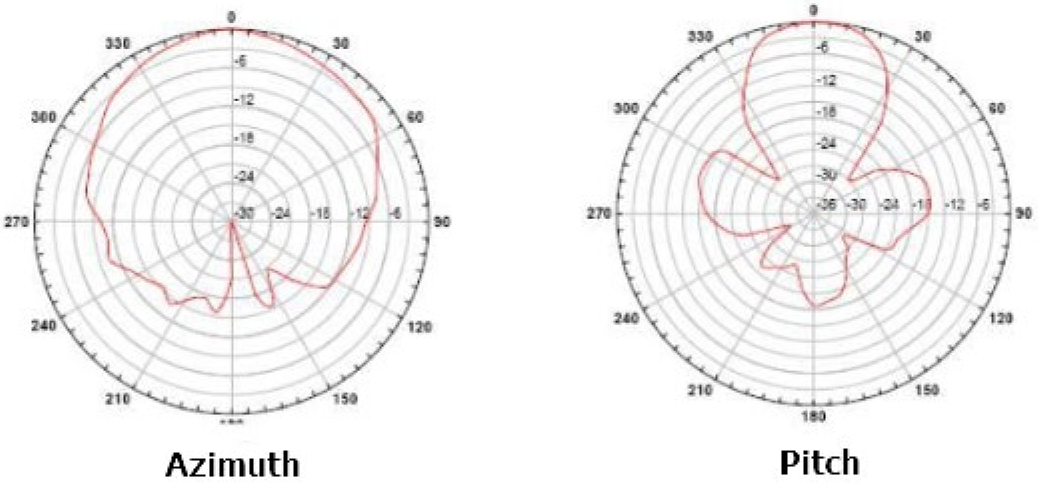


Figure 58 - Radar Module object detection test

One vital component of the project is the motor controller. Its function is to provide the drive motor control for the autonomous vehicle. As such, it was imperative that the functionality of the motor be tested and optimized. As the raw data is fed to the MCU, it will determine the signal needed for the motor controller to adjust the motor speed. To ensure proper function of the motor, a bench test was performed. Using the MSP430G2, and the DRI0002 dual motor controller a speed test of a generic motor was performed. The MSP provided a Pulse Width Modulated signal that was hard coded at different time rates to ensure proper function of the motor controller. The speed and the PWM cycles were recorded and plotted in Figure 61 below:

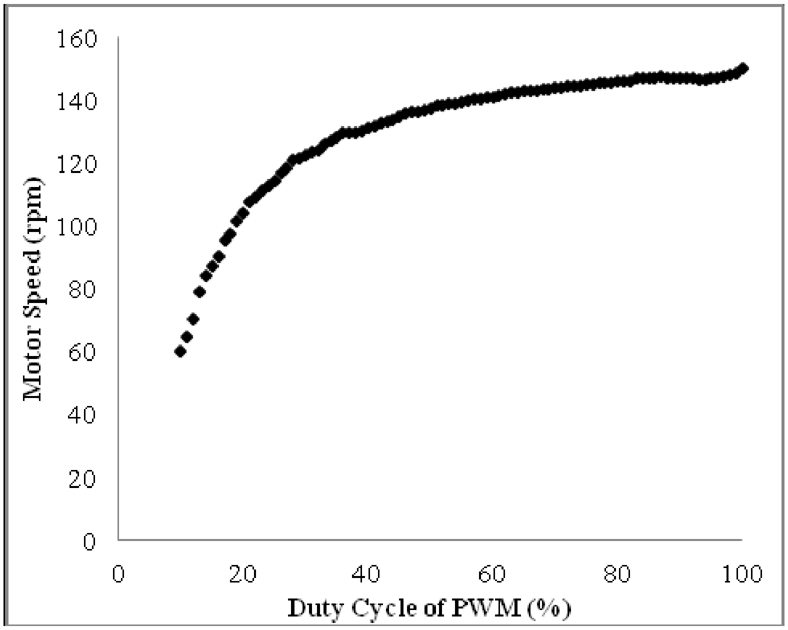


Figure 59 - Motor controller test data

7.3 Software Test Environment

The software test environment for this project will include multiple platforms to integrate the communications and data utilized by the various components. ROS is a framework that works with an operating system and can be easily implemented into any modern programming language such as: Python, C++, Lisp, as well as having experimental libraries in Java and Lua. Because ROS was designed to be as thin as possible, it provides facility to use ROS with other robot software frameworks. ROS was selected as the main programming framework for its ease of testing via the “rostest” which is an extension to roslaunch that enables roslaunch files to be used as test fixtures. As a fully running system has more complex behaviors than an individual ROS node, this allows to do full integration testing across multiple nodes. ROS can also be scaled for large runtime systems and for large development processes. This scaling feature would be of significant value when making the transition from a scale model into an automobile. The idea here is that the project can be accomplished (without the use of an RTOS, however) with our microcontroller of choice, but the processor would be less powerful, would have less memory, and would not be ROS-compatible.

As far as the programming language that will be used in this project, a few different options are presented. If our group ends up using the Jetson Tx2, the ROS programming will be done in Python, as it is much simpler to learn than C++, which is the only other native programming languages for ROS. Although it would be possible to include packages that allow for other programming languages, like Java, it is much simpler to go with Python and have many examples and documentation available to us. Python’s high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python offers a large library and modules that make just about every programming project realizable, and this library availability will make our lives much easier with this assignment.

7.4 Software Specific Testing

Software testing is a process to evaluate the functionality of a software application and the intent to find whether the developed software meets the specified requirements or not and to identify the defects to ensure that the products are defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. According to ANSI/IEEE 1059 standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item. There are different types of software such as: manual and automation. For our project, automated testing will only be done at the end of the project when every subsystem seems to agree with each other, and a prototype is put together.

Manual testing is the process of testing software by hand to learn more about it, to find what is and isn't working by observing the output behavior to a theoretical result known a priori. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing. Automation testing is the process of testing the software using an automation tool to find the defects. It is used to re-run the test scenarios that were performed manually, quickly, and repeatedly. Also, it is used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing. As the coding aspect begins to take shape and because of the nature of the project both types of testing techniques will be utilized when performing software tests. Once the MCU is programmed each component will undergo rigorous testing based on the set requirements. Manual testing will be conducted mainly on parts that required movement such as: the rotary encoder, ultrasonic sensors, radar sensors, audible system, failsafe system and steering control. Automation testing will be conducted mainly on components that required an automation tool to run.

In manual testing, each component will be tested to ensure they meet the set requirement. When performing a manual test on the HC-SR04 ultrasonic sensor, various distances of exact measurement will be tested to ensure proper function of the sensor. Once this is confirmed a test of the set required distance will be monitored to ensure that the sensor and the MCU are producing the expected results. In a similar fashion SEN0192 radar sensor will be manually tested by having a set of exact distances. Once the CU processes the data and determines the correct reading, the set required distance will be measure and verified. To manually test the rotary encoder the magnetic sensor will placed on the motor shaft to measure the revolutions of the motor. The vehicle will have to be powered to test the rotary encoder. The rotary encoder will provide speed data to the MCU to determine how fast the vehicle is approaching. In conjunction with the encoder the ZED camera will be utilized to verify that the rotary encoder is properly working. Testing at various set speeds the encoder will feed data to the MCU which will send the data to the camera. The camera will provide a time signature that will be used to calculate the time response versus the speed of the vehicle. The result will determine the proper function of the rotary encoder. The steering control will be manually tested by measuring the angles of the servo horn. The servo horn measures up to sixty-degree angle which is divided by having a thirty-degree angle to the right and a thirty-degree to the left. By moving the steering wheel, the steering control will measure the angle in which the horn is moving. Different set angles will be measured and fed to the MCU.

The fail system will be one of the most important components to manually test and verify because this is a requirement stated by the competition. The trigger of the remote control provided with the vehicle will be programmed to be the failsafe switch. Once triggered it will send a signal to the MCU to stop motor function. To ensure proper function of the trigger switch a button debouncing algorithm will be used to ensure the trigger is activated. The vehicle will be set in motion and the end user will press the trigger switch on the controller in which the vehicle will come to a stop. The audible safety system will be manually tested by moving the vehicle close to an object. The MCU will calculate the proximity of the object based on the sensor data and provided a signal to activate the audible system. Since the vehicle will be in constant motion it will be necessary to test with an automation tool. The Automation testing will be mainly performed on the MCU. Based on the data collected from manual testing of the components, different test scripts will be written to perform different tasks that will simulate the behavior of the various sensors featured on the vehicle. Since the limitations of the components will be known from manual testing, it will easier to identify errors in the automation tool. To test the MCU a script featuring different distances at a certain pace will be written to verify that the microchip can undertake this data and activate the appropriate sensor. Each sensor will have different thresholds (with hysteresis) that will become active once the threshold is met. These thresholds are feature on Table 1.

Once it is proven that an initial test is a success, the script will be either modified to include more sensors or a different script will be written for each component. This will be done to save time and to ease troubleshooting and debugging. The purpose of the scripts is to simulate best- and worst-case scenarios. Knowing the limitations of the components and the microchip will aide into designing and letting the end user know what the limitations of the vehicle are. Eventually one script will be able to simulate all the components in real time, meaning that it will simulate the car in motion and constantly feed data from the various sensors into the MCU.

8.0 Administrative Content

The old adage that “nothing is done until the paperwork is done”, rings true with this project. All administrative content pertaining to the project is contained within this section. In order to provide a proper administrative overview of the project process and progress, the team has devised a project milestone matrix that is feasible, realistic and capable of directing workflow for the design and implementation process. Also, in keeping with realistic cost constraints, the team is tracking major expenditures for the project as well as cost data for all sponsor furnished parts. The project bill of materials is included as a snapshot of group expenditures as the project progresses.

8.1 Milestone Discussion

Multiple milestones have been created to direct the progress flow for the project and are summarized in Table 16 below. These milestones have very little lag time due to time constraints inherent in the two-semester process.

Table 16 - Project Milestone Matrix	
Milestone	Due Date
Divide and Conquer 1.0	20 SEP 2019
Design Review	02 OCT 2019
Divide and Conquer 2.0	04 OCT 2019
Design Review	31 OCT 2019
60 Page Draft Document	01 NOV 2019
Design Review	12 NOV 2019
100 Page Document	15 NOV 2019
Design Review	30 NOV 2019
Final Document	02 DEC 2019
Order Parts	DEC 2019
Committee Selection	FEB 2020
Project Build	FEB 2020
Final Presentation	APR 2020

8.2 Budget and Finance Discussion

The main source for financing will be UCF professor Dr. Guo who is sponsoring the project. Dr. Guo will fund the majority of the parts summarized in Table 17 below. Minor parts will be purchased by the group members. The current bill of materials is summarized below in Table 17.

Table 17 - Bill of Materials Matrix

Part Name	Part Number	MFR	Vendor	Unit	Pricing	Qty
Ultrasonic Proximity Sensor	HC-SR04	WYPH	Amazon	10pc	$12.99	1
ARM Microcontrollers - MCU KINETIS 128K FLEX	MK20DX128VFM5	NXP	Mouser	EA	$6.13	1
Vehicle Chassis	74054-4	Traxxas	Traxxas	EA	$289.99	1
Jetson TX2	TX2	NVIDIA	NVIDIA	EA	$299.00	1
Stereo Camera	ZED	Stereolabs	Stereolabs	EA	$449.00	1
Carrier Board	ASG003	CTI	WDL Systems	EA	$174.00	1
Rotary Encoder	RS030	Sparkfun	Sparkfun	EA	$12.95	1
Radar Module	XM112	Acconeer	Mouser	EA	$74.95	1
Sound Board	2342	Adafruit	Adafruit	EA	$16.95	1
				Total	$1,335.96	

Appendix A – References

[1] National Conference of State Legislatures. Autonomous Vehicles, self-driving vehicles enacted legislation. Retrieved November 10, 2019. From http://www.ncsl.org/research/transportation/autonomous-vehicles-self-driving-vehicles-enacted-legislation.aspx

[2] DS/EN 62952-2:2016 - Power Sources For A Wireless Communication Device - Part 2: Profile For Power Modules With Batteries. Retrieved October 30, 2019. From https://webstore.ansi.org/Standards/DS/DSEN629522016-1646045

[3] IEEE 1625-2008 - IEEE Standard For Rechargeable Batteries For Multi-Cell Mobile Computing Devices. Retrieved October 30, 2019. From https://webstore.ansi.org/Standards/IEEE/IEEE16252008

[4] ANSI X9.112-2016 - Wireless Management And Security - Part 1: General Requirements. Retrieved October 30, 2019. From https://webstore.ansi.org/Standards/ASCX9/ANSIX91122016

[5] IEEE 1118.1-1990 - IEEE Standard for Microcontroller System Serial Control Bus. Retrieved October 30, 2019. From https://standards.ieee.org/standard/1118_1-1990.html

[6] Preparing for the future of Transportation: Automated Vehicles 3.0 (AV 3.0). Retrieved October 30, 2019. From https://www.regulations.gov/docket?D=DOT-OST-2018-0149

[7] IEEE 2050-2018 - IEEE Standard For A Real-Time Operating System (RTOS) For Small-Scale Embedded Systems. Retrieved October 30, 2019. From https://webstore.ansi.org/Standards/IEEE/IEEE20502018

[8] Marwa, C., Haythem, B., Ezahra, S., Mohamed, A. Image Processing Application on Graphics Processors (2014). *International Journal of Image Processing (IJIP)* Vol. 8 Issue 3. Retrieved November 1, 2019 https://pdfs.semanticscholar.org/282e/146a9085d66dffcda8a6f8239acc6dca1b58.pdf

Appendix B – Copyright Permissions

Copyrighted images are pending authorization for utilization as indicated in the text