UNIVERSITY OF CENTRAL FLORIDA

Final Report
Florida Solar Vehicle
Group 30

Hichame Boudi	Electrical Engineer
Billy Blanchard	Computer Engineer
Alyssa Fejer	Computer Engineer
Jesus Duran	Computer Engineer

Sponsored by

DUKE ENERGY®

Table of Contents

Table of Figures

Table of Tables

1.0 Executive Summary

The Florida Solar Vehicle Challenge was created as an interdisciplinary project to challenge students at the University of Central Florida to create a completely autonomous solar powered vehicle. It must be designed to detect and avoid objects and obstacles causing no harm to its surrounding environment. The vehicle must also be capable of transporting one passenger with a maximum payload of 120 pounds (lbs).

This semester one Electrical/Computer Engineering with Computer Science Team was formed to work in partnership with three Mechanical Engineering teams. Each team consists of senior level students who are enrolled at the College of Engineering and Computer Science at the University of Central Florida. Our team, consists of three Computer Science (CS) members, and four Electrical and Computer Engineering (ECE) members. As the ECE/CS unit, we will create a hardware/software suite responsible for power regulation, real-time object detection and avoidance using sensors/cameras, and simulation. This suite will then be transferrable into each one of the three Mechanical Engineering vehicle designs to be tested on a predetermined obstacle course at the University of Central Florida.

The budget for each team of this project is $1,500 with a total budget of $6,000 for all four teams sponsored and funded by Duke Energy. Each team also has a budget of $500 for any donated items. As a way to challenge the three Mechanical Engineering teams, they will all be constrained to use the same motor and batteries for their design. A final competition amongst the Mechanical Engineering teams will take place at the University of Central Florida at the end of the spring semester in April 2019. Our ECE/CS team will assist each team in completing the obstacle course for their unique vehicle designs.

This project has allowed the senior design students to accomplish a challenging technical project along with developing their communication skills and ability to coordinate and work with a group of not only those in their own discipline but two other disciplines as well. This paper documents the motivation, goals, objectives, requirements, research, system designs, budgeting, and testing processes of the Electrical/Computer Engineering with Computer Science Team.

2.0 Project Description

Every year thousands of people head to the east coast of Florida for the ultimate scenic drive down Daytona Beach. Starting in as early as 1902 with automobile and motorcycle races, beach driving has become one of the most popular and iconic activities beach goers have come to enjoy as part of their Daytona Beach vacation tradition. As time has gone on, this activity has become more and more controversial amongst locals as studies have been released showing the results of pollution and automobile use on this beautiful beach. As seen below in Figure 1, many beach goers enjoyed cruising down the beach in their classic cars.

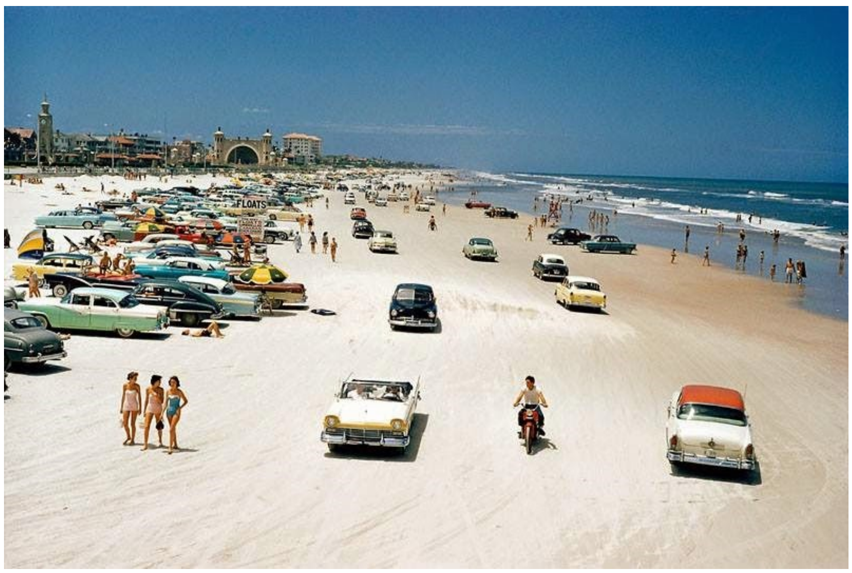


Figure 1. Tourist enjoying an afternoon drive along the Coast at Daytona Beach

Operating vehicles, including ATVs, on the beach can destroy wildlife habitat and be harmful or fatal to wildlife. This is one reason that, in many areas, beach-driving is strictly prohibited year-round to all but authorized personnel. The direct impact of vehicles on beaches largely relates to compaction and displacement of sand and beach soils; localized stripping of vegetation; damage to fauna; and creation of access tracks over dunes, across the foreshore and through vegetation. This in turn can lead to 'blow-outs' or dune erosion. Along with local environmental impacts beach driving has social impacts as well. Conflicts between vehicle users on beaches and pedestrians are major issues for managing beach access. These include, vehicles driving between pedestrians; vehicles crossing over dunes into pedestrian areas; dangerous driving; noise even in relatively remote coastal areas; and loss of overall beach experience.

With the rich history of vehicles on the beach and controversial environmental impacts has created a growing interest in a solution. This challenges engineers to create a solution that allows for beach transportation and a zero-emission/pollution impact. With Florida being the "Sunshine State" one of the best of these solutions is the use of solar energy to fuel an autonomous vehicle to transport a person down the 10-mile stretch of beach from Daytona Beach to the Ponce Inlet as shown below in Figure 2.

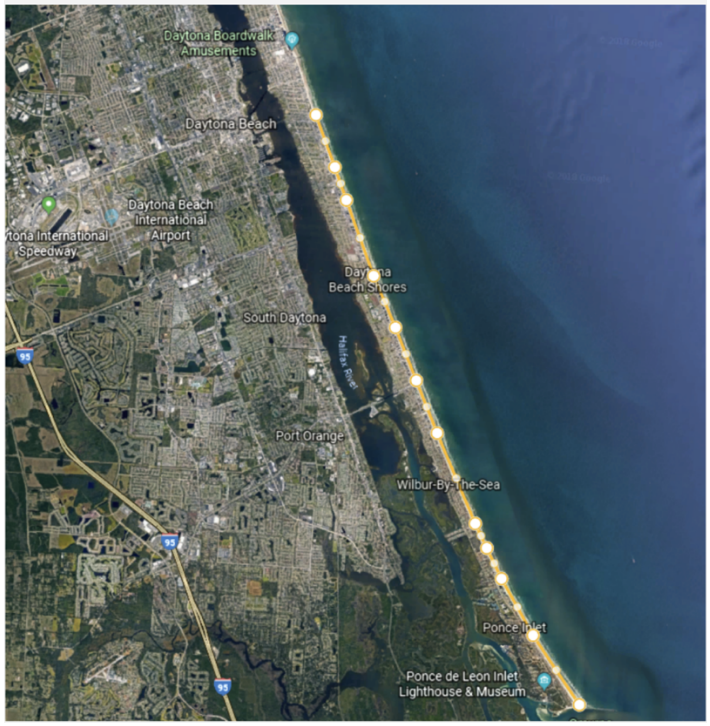


Figure 2. Map of 10-mile stretch from Daytona Beach to Ponce Inlet

Over the past ten years alternative energy sources have gained more and more attention within the transportation sector. The main push for this is the environmental impact. Solar panels work silently so they don't create noise pollution and they are zero-emission which means they don't create greenhouse gases, as gasoline engines do. Most importantly, solar energy is free, widely available, and grants a solar vehicle complete independence from foreign oil.

Solar cars use photovoltaic cells to convert sunlight into energy. Photovoltaic cells are the components in solar panels that convert the sun's energy to electricity. They're made up of semiconductors, usually silicon, that absorb the light. The sun's energy frees electrons in the semiconductors, creating a flow of electrons. That flow generates electricity that powers the motors of the vehicle to move across the sand down the beach.

Our solution to this challenge is to build a completely autonomous vehicle which can traverse the beach terrain without disturbing wild life or other beach goers and runs completely off of solar energy. We will test this proof of concept at the University of Central Florida within an obstacle course built by a professor adjunct. This light weight vehicle will be able to transport one person through a unknown obstacle course lasting at least 20-30 mins to ensure the system works on solar energy. This is the first step in testing if it is possible to build a vehicle able to traverse the 10-mile stretch of beach on the East Coast of Florida using object detection and avoidance to avoid any obstacles allowing for the full enjoyment of crashing waves and cool breezes the famous Daytona Beach has to offer.

2.1 Project Motivation and Goals

Every year renewable energy sources become more and more popular as the world realizes the permanent and lasting impact of using nonrenewable resources has on our environment. From economics to global warming, the immediate development of renewable resources is critical to the success of industry in the future. Renewable energy resources are clean, plentiful, and sustainable. The most obvious form of renewable energy in Florida is solar power. Solar energy is the most abundant renewable energy source available and modern technology has made significant strides to effectively convert and harness its power.

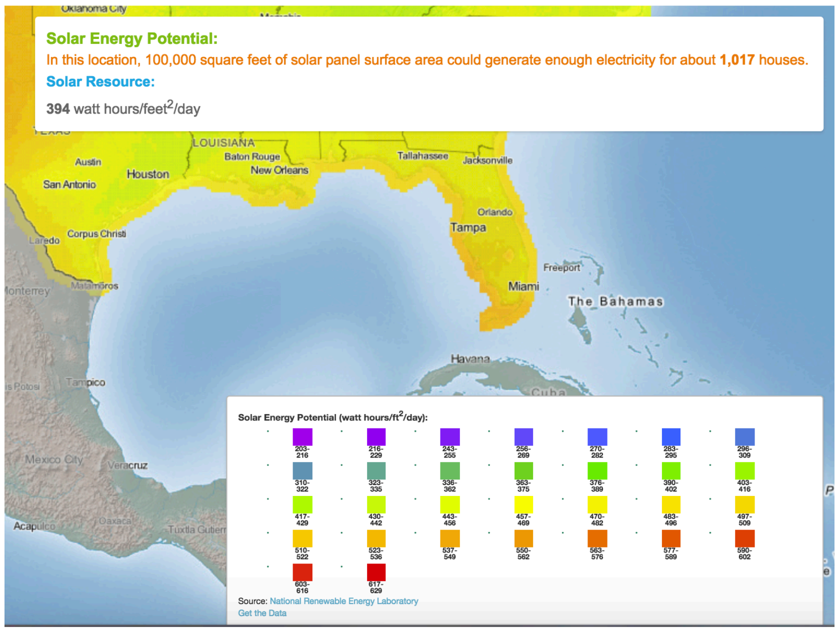


Figure 3: Solar Power Potential in the State of Florida (Energy.gov, n.d.)

As shown in Figure 3, almost the entire state of Florida has the solar energy potential to power over 1,000 homes with just 100,000 square feet of solar panel surface (Energy.gov, n.d.). With the increased interest in the solar energy potential in Florida solar projects and panels are becoming increasing more affordable and versatile. This motivates us to create completely solar reliant power system for our vehicle.

Also, with the recent advancements in technology, autonomous vehicles have become increasingly popular. Large automobile companies like BMW and Mercedes have already begun to produce semi-autonomous vehicles. Even Tesla, known for its technological advancements in electric powered vehicles, is set to launch its fully autonomous vehicle package by the end of this year. It is expected that 10 million driverless cars will be integrated in our roadways by 2020 (Autopilot: Full Self-Driving Hardware on All Cars, 2018). With our group graduating within the next year we want to be current on the recent developments to help us to be more competitive in the job market.

2.1.1 Broader Impacts

Autonomous vehicle software will have a huge impact on the future of transportation. Already across the country, several companies are testing this type of technology on real roads. Globally, an expansion of self-driving technology has the potential to save lives by avoiding human-caused collisions. According to the National Highway Traffic Safety Administration, over 37,000 people died in motor vehicle crashes in 2017 (Transportation, 2018). Although fatality rates are trending down over the past decade, thousands are still dying every year doing such a common task. Personal autonomous cars have the potential to drastically reduce the number of vehicular fatalities.

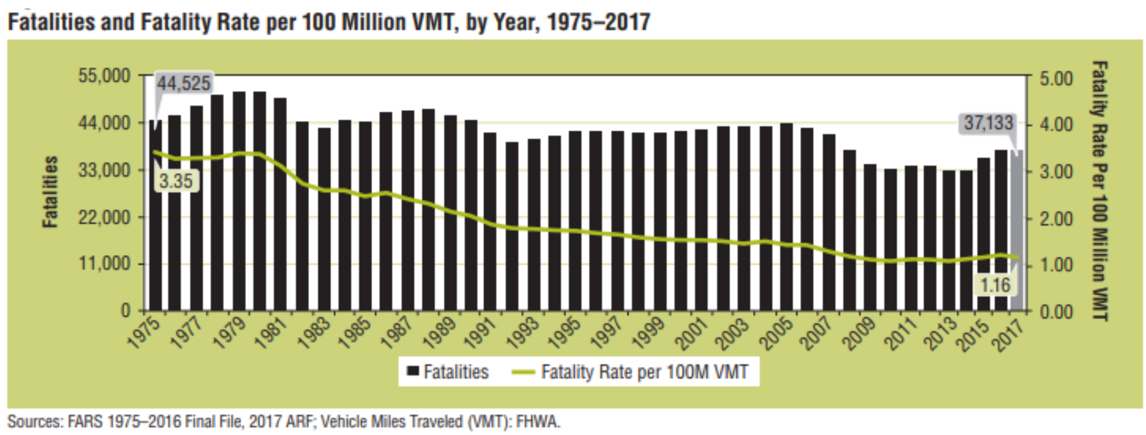


Figure 4. Fatality Rates from 1975 - 2017, National Highway Traffic Safety Administration (Analysis, 2018)

Locally, in addition to saving lives, self-driving cars allow for more accessible transportation options. Ride-sharing would no longer be dependent on a driver, allowing cars to drive unlimited by an operator. Commercially, autonomous vehicles can transport physical goods without the need for human oversight, increasing efficiency and reducing excessive labor costs.

This project involves navigating through an obstacle course created solely for testing this project. The layout will be makeshift, and unknown to us prior to running. Adjusting to the driving environment of a makeshift obstacle course allows us to consider unique approaches to maintaining position on the road. Perhaps these ideas could be applied to other autonomous solutions as an enhancement of existing methods.

Having an electric car replenished by solar energy is another huge goal for the future of transportation. With climate change reaching dire levels, transportation options that run on clean energy will become more important in the coming years. The ability for an electric car to not only drive itself, but to also recharge itself using solar power, would allow people to move safer, cleaner, and more efficiently than ever before (Transportation, 2018).

Autonomous vehicles that run on clean energy have the potential to be utilized across several different fields to solve a variety of problems.

2.2 Objectives

The objective of this project is to design and build a solar powered vehicle that is able to traverse an obstacle course created by Steven Flanders, adjunct professor at the University of Central Florida for the Mechanical Engineering Department. The vehicle must be able to complete the course and be fully autonomous while detecting and avoiding both stationary and moving objects, persons, or obstacles. The vehicle must also determine the appropriate response to execute after a figure has been detected in order to continue its path on the course.

A stretch objective for this project is two be able to reconstruct the course via simulation for path optimization to decrease the time the vehicle takes to complete the course, since it would know in advance where most of the obstacles would be located if they are stationary. This will prove to be a challenging task hence why it is a stretch objective.

2.2.1 Project Roles and Tasks Assignments

This is an interdisciplinary project; therefore, the work will be divided amongst the disciplines of mechanical engineering, computer science, and electrical and computer engineering. Each of the three Mechanical Engineering teams are composed of 7 senior undergraduate students. The ECE/CS Unit team is a separate team creating a software/hardware suite to plus into each Mechanical Engineering vehicle separately, ensuring full functional ability for each design. Each team is responsible for their own field of engineering including, research, design, and testing.

A group lead has been chosen for each group to act as the main point of contact amongst all four groups to stream line communication and questions. It is the role of the team leader to communicate to the others groups their group's needs, complications, and design implementations so the project can reach a finished working state. Steven Flanders has also been assigned as an adjunct professor by the Mechanical Engineering department to oversee the communication of all groups.

Each team lead will also be in charge of ordering for the groups. The Mechanical Engineering students will have all of their components ordered directly through UCF and the Mechanical Engineering Department. Since our ECE/CS team will need to purchase equipment sooner then our mechanical counterparts we will be ordering components on our own. We will self-organize and group the receipts together to ensure the budget is maintained. Since we need to test each component before programming can begin we need to order all components sooner than later. Our team will be reimbursed via the Electrical and Computer Engineering School.

As for the individual assignments for the ECE/CS group we have divided the project according to each student's strength. One CS student and one ECE Student (Bill) will work on programming and utilizing robot vision for object detection, recognition, and avoidance. The two of the CS students will work on Simultaneous Location and Mapping (SLAM) computer algorithms to try to have the computer read in the obstacle course and virtually simulate it while detecting and avoiding objects while optimizing path planning. Two ECE students (Hichame and Jesus) will design and test custom Printed Circuit Boards (PCBs) along with all components needed for power distribution throughout the vehicle. And the last ECE student (Alyssa) will work on component integration for both electrical, computer, and mechanical teams.

2.2.2 Solar Power System

The objective of the Solar Power system is to power each Mechanical Engineering team's solar vehicle using solar panels, charge controllers, two twelve-volt batteries for the motor system, and one fourteen-volt battery for the electronics and steering system. This system will involve a solar panel and batteries to store solar energy to be used when there is no sunlight being harvested or when the batteries are exhausted and need to be recharged. This will be tested through designing an obstacle course long enough in which it will cause each vehicle to exhaust all the power provided by the batteries. If the solar vehicle finishes the course then the batteries will have been recharged by the power supplied by the solar panels.

2.2.3 Autonomous Navigation

The objective of the autonomous navigation is to design a system of sensors and cameras that can detect stationary objects, moving objects, soft body, and hard body objects. The data collected by the sensors and cameras will then be transmitted to the solar vehicle to implement steering around those obstacles to complete the course. We would also like to try to program a path planning algorithm as the vehicle traverses through the course.

2.2.3.1 Obstacle Course

Our autonomous vehicle will navigate through an obstacle course, looping around repeatedly as much as possible. The winner of the competition will be the vehicle that can make the most loops around the track. The width of the track will depend on the dimensions of the vehicles' final design. The length of the course will also depend on the specific location of the course itself. The location may either be on a grass lot or an asphalt parking lot. The vehicle will also need to maneuver around obstacles to complete each loop. Because the course is to be constructed specifically for this project, and is not a preexisting road, the exact layout of the course will be unknown to us prior to the competition. However, we will have access to a list of the elements with which the vehicle will interact (such as the specifications of the obstacles, course walls, colors of the objects, etc.).

A successful completion of a course loops involves on a few specific tasks being accomplished. First, our system will need to correctly regulate the speed and movements of the vehicle around the track. Proper acceleration and steering will be important to making sure that the vehicle is navigating the course correctly.

Second, we will need to have the ability to scan the environment and understand the boundary of the course itself. This includes the ability to detect any potential obstacles in our way that could hinder the forward progress of the vehicle.

Third, our system will need to correctly generate specific instructions as to how the vehicle should navigate the analyzed environment. There is also a time component to generating these instructions. If the vehicle is in motion, it will need to know where to go next. A delay could result in either motion that is not being evaluated, or the vehicle constantly fluctuating between moving and halting.

Additionally, since this is organized to be a competition, the system should be designed to allow the fastest navigation possible. Specific hardware management, simultaneous localization and mapping, and an efficient, reliable navigation algorithm will be needed to successfully navigate the obstacle course.

2.2.3.2 Expected Obstacles

There are three main classes of obstacles that our vehicle will need to navigate around in order to complete the course.

The first type is the obstacle course border. The vehicle will be limited to movement within this defined, enclosed space. Our navigation system will need to process these borders and generate a viable path between both sides, preferably down the center. Any obstacle avoidance maneuvers will also need to account for the course walls, preventing a collision or movement out of the course bounds. These boarder walls will have a specific design associated with them, so that our computer vision component will be able to correctly recognize them.

The second type of obstacle to account for are stationary obstacles. These will be obstacles that have a fixed position within the course that will obstruct the forward progress of the vehicle and require an avoidance maneuver. These stationary obstacles may or may not be reoccurring, meaning that for each loop around the course, the vehicle will encounter the same object. An avoidance maneuver will only be taken if an object directly impedes the forward progress of the vehicle. Our vehicle will also encounter speed bumps, classified here as an obstacle our system may be capable of recognizing and processing as an obstacle, but one that will not impede the progress of the vehicle and as such should not influence navigation.

The third type of obstacle the vehicle will encounter are moving obstacles. These are objects that impede the forward progress of the vehicle but are either in a constant state of motion of stationary objects which appear irregularly in different locations, varying from lap to lap. Objects which are in perpetual motion, that impede progress, will halt the vehicle until removed. Static obstacles that the vehicle will recognize as new to the course, will need to be bypassed. Although response action is a simple stop, it means that our system will need to differentiate between a static and moving object throughout the entirety of the course run.

2.3.4 Engineering Development Process

The objective for our group as it relates to the engineering development process is to fully understand the scope of our project from all disciplines involved, including mechanical, electrical, computer, and computer science, and to effectively work as a team to build a working solar powered autonomous vehicle better then all groups whom have completed this project before us. We hope to gain an understanding of the level of communication, documentation, and execution that is expected of engineers by customers. We will break down the project into tenable sections that separate research, design requirements, prototypes, feasibility, and other project related benchmarks.

2.3.4.1 Anticipated Challenges

One problem we anticipate we may encounter is choosing hardware in conjunction with what algorithms we could use. Do we need specific hardware to integrate certain algorithms? Should our algorithms be modelled based on what hardware we have available to use? Our budget is limited, and we need to establish what we can do, with what we can afford, sooner rather than later.

Since the project has not been down in the way we are hoping to complete it this semester we are also concerned about the overall design of the solar vehicle. Solar power is used in many applications, but a fully solar powered autonomous vehicle is not one of them. To overcome this challenge, we are doing as much research as we can on existing systems and what others have learned from integrating electrical systems will solar power.

The power distribution of this system is the most critical challenge of the entire design next to avoiding objects. We have done research about recharging batteries from a solar panel, and the difference between our system and the ones we discovered was that our system will have two different voltage supplies pulling from the singular solar panel array. We are very constrained on size otherwise we would have two separate solar panel arrays connecting each power source to its own array. Since that is not an option we will be doing extensive testing on all components to ensure power is being distributed properly throughout the system to ensure safety and efficiency.

As mentioned above in the objectives section, we are hoping to utilize a new technology called SLAM (Simultaneous Localization and Mapping) algorithm. Simultaneous Localization and Mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. This is a growing and emerging field within autonomous technology. The challenge for us is to figure out how to utilize this within our system successfully. Being about to build the environment virtually as the solar vehicle interacts with it will be a huge accomplishment not only for this project, but in engineering in general. This proof of concept will lead the way for multiple integrations of this technology when an environment is unknown, which most of the time that is the case.

Another problem we anticipate if having enough time to test and integrate the ECE/CS Software/Hardware Suite on each one of the Mechanical Engineering vehicle designs. We are confronting this challenge by meeting with each group once a week to discuss vehicle progress and planning. Recently we discussed with them the need to finish building each vehicle as soon as possible. Ideally, we would like each group to finish their design and build leaving us with a month for system integration and basic testing.

2.3 Requirement Specifications

Requirements

Many of the requirement specifications and constraints for this project were provided by the University of Central Florida Engineering Department. The professor in the department wanted to challenge the students into creating an efficient durable design, while also ensuring the student had to spend their funds creatively to solve the project challenges. The tables below list the overall project requirements, specifications, and constraints.

Table 1. Project Requirements

Description	**Value**
Vehicle run time	≥ 20 mins
Capable of transporting one passenger	≤ 120-lbs
Top allowable speed	5 mph
Run completely on solar energy	100% Solar
Vehicle should not cause harm to environment	0 Emission
Vehicle should detect and avoid both stationary and moving obstacles	N/A
Solar Panel power output	+300 Watts
Store solar energy on battery/batteries	12V battery (x2)
Equip HD camera(s) for Computer Vision	+5 MP
Equip Ultrasonic sensors for object/distance detection	≤ 6 sensors
Equip IMU sensor (BNO055)	1 sensor
Equip NVIDIA Jetson TX2 for Image processing	1 board
Equip solar charge controller	2 controllers
Equip Printed Circuit Board	1 PCB

Specifications

Table 2. Project Specifications

Description	Value
Navigation algorithm to avoid unnecessary stopping	N/A
Reach maximum velocity in approximately 5 seconds	Accel. ≈ 0.45 m/s^2
Time to fully charge battery/batteries	≤ 12 hours
Noise level	≤ 100 dB
OpenCV for obstacle detection	≥ 6 x 4 inches
Object detection distance range	3 – 8 ft
Navigate within lane boundaries (Traffic Cones)	≤ 12 ft width
SLAM for path planning	N/A

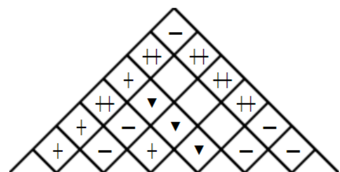
Constraints

Table 3. Project Constraints

Description	Value
Vehicle length	≤ 6 ft
Vehicle height	≤ 5 ft
Vehicle width	≤ 4 ft
Vehicle gross weight	≤ 500-lbs
Vehicle ground clearance	6 – 12 inches
Number of electric motors	1 motor
Solar Panel size	≤ 24 ft^2
Water resistant enclosure for electrical components	IP66 Standard
Total cost	≤ $1500.00

2.4 House of Quality Analysis

The House of Quality diagram on figure 4, defines the relationship between the customer requirements and the team/project capabilities by highlighting the trade-offs between meeting certain requirements. Keeping the project at a low production cost (<= $1500.00) is a challenging task due to the price of hardware components to fulfill the other requirements. One of the biggest impacts on keeping the project at a low cost is the obstacle avoidance requirement. This requires the use of a at least 6 ultrasonic sensors to cover the majority of blind spots in the path of the vehicle (forward and reverse).



				Column #	1	2	3	4	5	6	7
				Direction of Improvement:	▼		x	x		▲	▲
Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Demanded Quality	COST OF HARDWARE COMPONENTS	GPS MODULE	ULTRASONIC SENSORS	IMAGE PROCESSING	EMERGENCY STOP BUTTON	SOLAR PANEL	EFFICIENT ALGORITHM
1	9	17.6%	6	LOW COST	Θ	O	O	Θ	▲	Θ	▲
2	9	23.5%	8	AUTONOMOUS NAVIGATION	O	Θ	O	O	▲	▲	O
3	9	23.5%	8	AVOID OBSTACLES	Θ	▲	Θ	Θ	▲	▲	O
4	9	20.6%	7	SOLAR POWERED	Θ	▲	▲	▲	▲	Θ	▲
5	9	14.7%	5	EFFICIENT	▲	O	O	O	▲	O	Θ
				Target or Limit Value	<= $1500.00	1	>= 6	none	1	>=300 WATTS	N/A
				Difficulty	8	5	4	9	1	4	10
				Max Relationship Value in Column	9	9	9	9	1	9	9
				Weight / Importance	641.2	352.9	400	505.9	100	435.3	311.8
				Relative Weight	23.3%	12.8%	14.6%	18.4%	3.6%	15.8%	11.3%

Legend

- ++ strong positive correlation
- \+ positive correlation
- — negative correlation
- ▼ strong negative correlation

Direction of Improvement

- ▼ objective is to minimize
- ▲ objective is to maximize
- x objective is to hit target

- ▲ weak relationship
- O moderate relationship
- Θ strong relationship

Figure 5. House of Quality

Also, for high quality image processing, it is required to obtain a computational device with a high-performance CPU and GPU which can significantly increase the overall cost of the project. While not a requirement for ECE to provide a Solar Panel, in the overall scope of the project, it is one of the biggest impactors of the total cost due to the dollar amount per output wattage (Usually $1.00 per Watt). Some strong correlations can be draw when it comes to having the vehicle run efficiently between the navigation algorithm and the feedback data from the GPS module, ultrasonic sensors, and image processing unit.

3.0 Research related to Project Definition

This section reviews current products and relevant research done to design and build our ECE/CS Unit software/hardware suite that will be used by each of the Mechanical Engineering Teams to create a fully autonomous solar powered vehicle. Research on current products and technology will have the greatest impact on our design as cost is a factor. Designing an effective solar powered system has a potential to greatly increase costs if research is not done properly. Designing a cost-effective system is critical for this project, therefore we must utilize technologies that have already been vetted and tested.

The following topics were also included in our overall research Thorough research has also been conducted on possible components needed to develop an effectual photovoltaic system as well as a functioning navigation and object detection system in order to achieve autonomy. The researched components include sensors, cameras, microcontrollers, photovoltaic cells, batteries, voltage regulators, autonomous vehicle navigation, and image processing. This section will also discuss the final components selection for the initial design of the system.

3.1 Existing Similar Projects and Products

3.1.1 Tesla

Being the one of the first successful car companies since Chrysler and one of the only car companies utilizing Autonomy, Tesla is arguably the golden standard for practical applications of robot vision.

Figure 6. View from Computer Vision Front Camera on a Tesla Vehicle (Autopilot: Full Self-Driving Hardware on All Cars, 2018)

Straight from the Tesla website, all their cars utilize a similar (albeit more thorough) sensor suite as our Autonomous vehicle is planning on using: eight surround cameras with twelve ultrasonic sensors to complement “allowing for detection of both hard and soft objects at nearly twice the distance of [previous models] (Autopilot: Full Self-Driving Hardware on All Cars, 2018).” This setup, along with an additional radar sensor, is displayed in Figure 4. It has been extraordinarily successful, allowing for detection of objects in front (like in Figure 3), the classification of said objects (like the car and bus), and can also detect road lines, street signs, pedestrians and the objects around the vehicle with their respective distances away.

Tesla has a unique approach to their front vision using not one but three cameras to detect objects that pose an immediate threat to the vehicle and its operation: a wide camera, which detects things like traffic lights, pedestrians cutting into the path of the car, and other close range objects, a main camera which is in charge of the majority of object detection cases, and a narrow camera, which provides a focused long range view, perfect for high speeds (Autopilot: Full Self-Driving Hardware on All Cars, 2018). They also use more cameras in the rear and on the sides for additional object detection on the sides, working complementary to the twelve ultrasonic sensors in order to sense traffic along the sides of the vehicle and allowing it to navigate tighter quarters at higher accuracy.

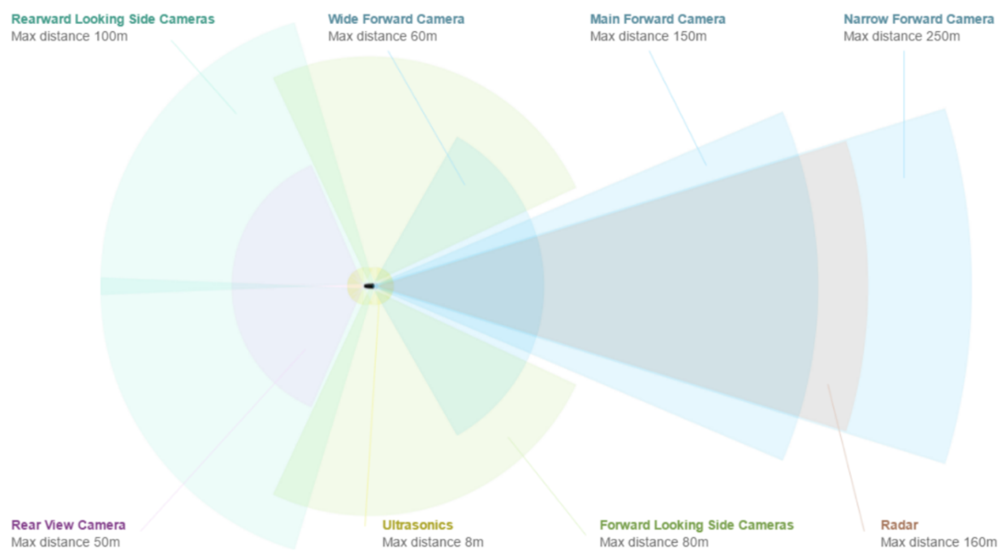


Figure 7. Degree of Angle coverage for each sensor and/or camera on Tesla Vehicles (Autopilot: Full Self-Driving Hardware on All Cars, 2018)

Even with all these features, Tesla autopilot is still seeking regulatory approval and is not even complete. While it does have some autonomous features, like auto cruise control, summoning, and exiting a highway, complete autonomy is a different beast entirely and there are several cases of where trial versions of this software resulted in deadly crashes.
All this being said, Tesla attempts to tackle a beast outside of the scope of our project: high-speed vehicle operation. Our solar vehicle will only be traveling at a max of 5mph, making it significantly easier to detect objects with only the use of a main camera and some ultrasonic sensors. Tesla has proven that, at least at low speeds, it tracking and detection algorithm is incredibly effective and can be detect, with extreme accuracy, the objects around it as well. It is difficult to fully dive into their algorithm because of proprietary restrictions, but Tesla proves that it is possible to have successful object detection with just cameras and ultra-sonics (Autopilot: Full Self-Driving Hardware on All Cars, 2018).

However, Tesla's onboard computer is many times more powerful than our Raspberry Pi, utilizing an onboard Deep Neural Network to process their plethora of sensors together to produce output. Since they are building a vehicle on a much larger scale this makes sense, but fundamentally the problem we face for this project is the same as theirs. This makes the differences in our vehicle and their vehicle more notable (Autopilot: Full Self-Driving Hardware on All Cars, 2018).

3.1.2 Waymo

Waymo is a lot like Tesla; they operate a car company that specializes in autonomous driving technology, touting over 10 million miles of real-world experience. They then go on to say that these 10 million miles are used to "teach" the cars as they continue. While they don't go on further than that, probably because of proprietary reasons, this alludes to some kind of neural net that learns based off past experiences, just like Tesla.

However, unlike Tesla, the Waymo website explains that through their sensors, the vehicle can see up to three football fields away in any given direction with respect to the vehicle. Tesla has cameras that specialize in views in different directions, and sensors which complement that, but they don't come close to this unidirectional power (Technology | We're building a safer driver that is always alert and never distracted., 2018).

Waymo accomplishes this via Lidar, which they tried and failed to patent. Lidar works kind of like radar, but instead of sending out radio waves it emits infrared light, measuring how long it takes them to come back after hitting nearby objects. After doing this millions of times per second, it compiles a sort of point map (Figure 8) of the surrounding environment, which is then used to calculate the best route for the vehicle to take (Technology | We're building a safer driver that is always alert and never distracted., 2018).

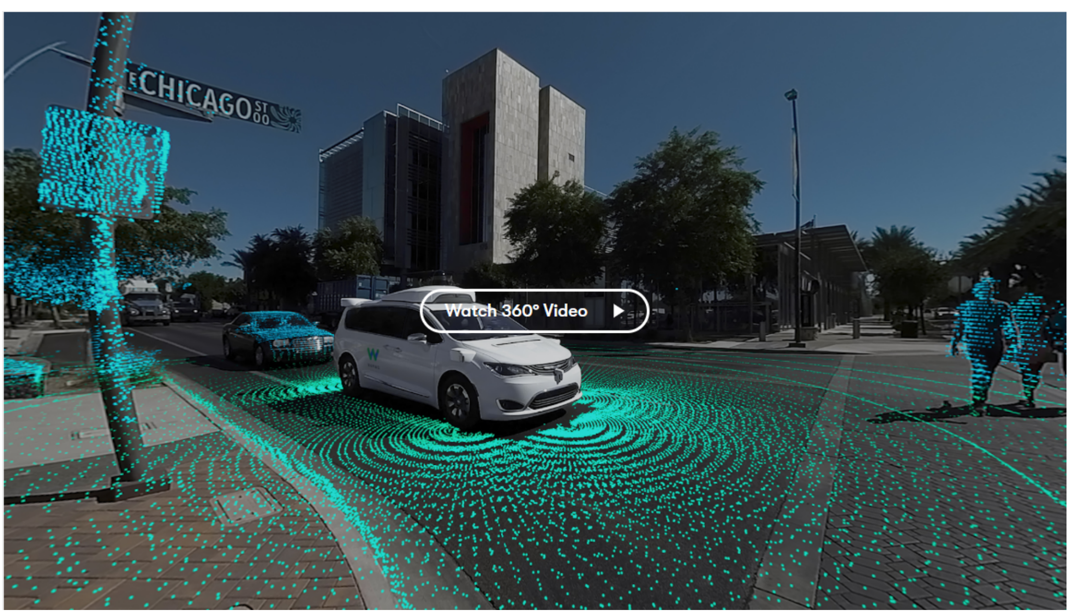


Figure 8. Example of Waymo Lidar Detection from a Vehicle

Tesla only uses radar, which is not nearly as accurate, mainly just for detecting fog and other inhibiting conditions that mess with the other sensors. After that, they rely on machine learning technology to process the 2D camera images into 3D data, unlike Lidar which comes 3D and ready for the algorithm to process off the bat (Autopilot: Full Self-Driving Hardware on All Cars, 2018).

While Lidar is great for this purpose, it has its drawbacks. First, it is expensive, costing more than $4,000 for one, and most cars need several. Another thing is that they aren't extremely resilient; to put it on a car it needs to withstand years of hard wear and tear, and since it is attached to the vehicle outside (kind of like a Google Street view car) it is difficult to do that. Still, most autonomous car companies insist that Lidar is the future, with the exception of Elon Musk and Tesla (Technology | We're building a safer driver that is always alert and never distracted., 2018).

3.2 Relevant Technologies

3.2.1 Solar Panels

This Solar Vehicle competition aims to make use of the unprecedented popularity of Solar Energy as a source of renewable energy. In 2017, the global solar market grew 29.3 percent, with nations installing 98.9 gigawatts of new capacity, according to data from the industry group SolarPower Europe (The World Added Nearly 30 Percent More Solar Energy Capacity in 2017, n.d.). Photovoltaic solar panels, referred as just solar panels, are composed of a number of photovoltaic cells that generate electricity by absorbing sunlight. They are generally fabricated using semiconducting materials, mainly Silicon and Gallium, with varying degrees of efficiency. Solar panels have been deemed expensive, heavy, and impractical to be considered for vehicular applications, but these traits keep getting reduced every year with new advances in solar technology.

The majority of today's solar cell market, about 95%, is comprised of crystalline silicon which comes in two categories, monocrystalline and polycrystalline (The World Added Nearly 30 Percent More Solar Energy Capacity in 2017, n.d.). *Monocrystalline* solar cells are made out of, very pure, silicon ingots and have some of the highest efficiency ratings (< 20%). The higher efficiency gives this type of solar cell a high price tag, highly influenced by the four-sided cutting process involved that wastes a high amount of silicon (The World Added Nearly 30 Percent More Solar Energy Capacity in 2017, n.d.). *Polycrystalline* solar cells are a the most popular choice due to the lower cost. The silicon is melted and molded into a square shape, with close to no silicone waste. The drawbacks are a lower efficiency rating of 13 to 16 percent and a lower heat tolerance that lowers its efficiency (The World Added Nearly 30 Percent More Solar Energy Capacity in 2017, n.d.).

Another solar cell technology that has grown in popularity in recent years is *Thin Film*. They are fabricated by layering a series of thin, semi-conducting material (including silicon), films. Thin Film cells can achieve efficiencies that range from 6 to 13 percent. There is a lower manufacturing cost involved with thin film technology, but it requires a far greater amount of space. In 2011, the price of Silicon dropped significantly, contributing to a drop in the price of solar PV modules to below $1 per watt (Davison, n.d.). A lower price on silicon, would allow companies that produce solar modules to add extra processing steps to increase the efficiency of the solar cells.

3.2.2 Distance Sensors

One of the major requirements for the project include, the ability to detect and avoid objects (See Requirements and Specifications). There is a wide selection of sensors that perform the task of detecting objects at various distances. These include infrared, passive infrared, inductive, capacitive and ultrasonic sensors. The first category of sensors are those with only close range detection. The most basic functionality of these sensors is Inductive. Inductive sensors are limited by the ability to only detect ferrous materials, those being made only from iron, and close sensing range (< 100 mm) (Kinney, 2001). Then there is Capacitive. *Capacitive* sensors, while detecting a variety of materials, operate at close ranges (< 100 mm) (Kinney, 2001).

The next category of sensors is those with a wider range of object detection, but with challenges in other areas. *Infrared* (IR) sensors can detect objects at a large ranges object by emitting infrared radiation and detecting it when it bounces back from an object. The downside to an IR sensor is the various environmental constraints that limit their functionality. IR sensors' performance is affected by lighting conditions of the environment, i.e. very bright/dark conditions, and objects that lack a reflective surface become harder to detect, especially objects with a dark surface (Kinney, 2001). *Passive Infrared* sensors (PIR) can detect objects in bright/dark environments by detecting a change in temperature from objects that radiate heat. The dynamic nature of the project does not provide the ideal environment for a PIR sensor application. *Ultrasonic* sensors detect objects by emitting high frequency (> 40 kHz) sound waves and sensing the wave reflected off the object. The distance is measured by using the speed of sound and the time that the wave takes to and from the object. Ultrasonic sensors can detect objects from centimeters to a few meters (~3 m) and are generally not affected by environmental conditions like light, dust, smoke, fog, and similar conditions. The downside to an Ultrasonic sensor is the inability to sense object with elevated sound absorption features, e.g. stuffed toys, and objects shaped in a way that reflect sound away from the receiver i.e. flat objects at an angle (Kinney, 2001).

3.2.3 Orientation Sensors

A third category of sensors takes inspiration from technologies found across all different types of scientific tools. Numerous systems today rely on the use of gyroscopes, accelerometers, and magnetometers sensor to measure different aspects of motion, whether it involves cars, airplanes, mobile devices, etc. An Inertial Measurement Unit (IMU) is a device based on multi-axis combinations of precision gyroscopes, accelerometers, magnetometers, and pressure sensors (Kinney, 2001). The importance of an IMU becomes evident in the ability to determine the orientation of the vehicle with respect to the geographical cardinal directions (points). This has been an important aspect of navigation dating many centuries back. IMUs can be used to measure the acceleration exerted by the vehicle on three axis (x, y, z) which can aid in tracking and adjusting the behavior of the vehicle due to changes in the terrain. Some IMUs offer additional features like temperature and atmospheric pressure measurements. There are a variety of IMU sensors on the market that offer a smooth integration into systems using the components that are being taken into consideration for this project. They also offer different ranges of degrees of freedom (DOF), which refers to the number of individual readings the IMU can perform (Kinney, 2001).

3.2.4 Microcontrollers

As technology grows in the field of electronics, so does the need for mini computers within systems to process minor computation and tasks. These computers are called microcontrollers. A microcontroller is a small, low-cost and self-contained computer-on-a-chip that can be used as an embedded system. They can be thought of as tiny computers that are added to any physical object or space to give it a brain (Microcontrollers – Types & Applications, 2018). Each microcontroller contains contain one or more computer processors, along with memory and programmable input/output peripherals — all in a single integrated circuit. Microcontrollers have less capability than a standard computer processor, but they make up for this with their low cost and compact size. Making microcontrollers perfect for personal electronic projects (Microcontrollers – Types & Applications, 2018).

In order to be able to determine which microcontroller will work the best with our solar vehicle, we needed to determine some of the key features of microcontrollers and what they do. Below are some of the specifications we used to determine the type of microcontroller needed.

- Bits: Microcontrollers are typically sold by the number of bits that they offer. This impacts the speed at which they are able to perform non-trivial computations (Califano, 2018).

- RAM: RAM is a fast-access memory that does not retain data in an absence of power. All MCUs come with certain amounts of RAM, which allows your microcontroller to quickly perform various actions. The more you have, the better, but the added RAM increases the cost of the MCU (Califano, 2018).

- Flash: Flash is computer memory that retains data in the absence of power. At least some of this is essential, and it’s very useful for features like offline storage (Califano, 2018).

- GPIO: GPIO stands for general-purpose input/output pins. These are the pins that you will use for connecting your sensors and actuators to the MCU and the internet. The number of pins can range from one to the hundreds, depending on the microcontroller (Califano, 2018).

- Connectivity: This is how the board (and application) connects to the Internet via Wi-Fi, Ethernet, or some other means. This is an important aspect of connected sensor applications, so we'll go over this topic in greater detail later (Califano, 2018).

- Power consumption: Power consumption is critically important for connected sensor applications, particularly when your device has to rely on something like a battery or solar power. This spec will tell you how power hungry the MCU is by default and whether or not it supports power-conscious programming techniques (Califano, 2018).

- Development tools and community: It's important that there is a mature set of tools, documentation, and community support to help build programs that will run on the MCU you select for your application (Califano, 2018).

For our vehicle we will be using sensors to determine objects in the vehicle path that we need to avoid. This is where another important and helpful element of microcontrollers is what is known as a development board comes in. A development board provides everything necessary to program the microcontroller. They're the perfect starting point for building connected systems. Development boards are printed circuit boards containing a microcontroller and the supporting components needed to program the microcontroller (Califano, 2018). They include things like a power source, support for connecting sensors. A development board is necessary or our application of sensors on the vehicle because development boards enable users to quickly connect sensors and actuators and their accompanying software facilitates the creation and deployment of code.

3.2.5 Graphical Processing Units

Throughout this report, it has been mentioned that one of several key requirements for our Solar-powered Vehicle is to detect and/or avoid common objects, that we would specifically describe as or define to be obstacles. Another one of the key requirements for the project is that it should sustain the ability to run autonomously off solar energy. In that, power-efficiency with both obstacle detection and obstacle avoidance, are somethings we were forced to take into serious consideration when choosing the hardware components that will be strategically used and implemented to aid in the design of this robot.

Research has been previously conducted by another University of Central Florida Senior Design team early as of February of this year. The team worked on a project that was closely related our solar vehicle experiment. The research led the team to become persuaded that a project topic of this caliber, alongside these complex requirements and constraints, could be completed using the Single-Board-Computer (SBC), the Raspberry Pi 3 Model B. After completing an extensive amount of research on the Raspberry Pi 3 computer and other embedded systems, we learned that the Pi 3 Model runs on a 4-core 64bit CPU with 1.2GigaHertz of processor speed (RASPBERRY PI HARDWARE, n.d.). This is ideally slow for the tasks we would expect the robot to accomplish.

After a careful and precautious effort of self-conducted research on several other hardware components, the Raspberry pi 3 Model B is arguably an ideal choice for completing this project. By far, the Pi 3 is the best choice to make when applying the most economical aspects; This single-board-computer comes at a low-cost, it is credit-card sized, and it is also very versatile when it comes to the OS selection. With that being stated, the Pi 3 is perfect for completing a prototype, or Proof-of-Concept (PoC) of the robot performing such tasks and possibly meeting the specific requirements mentioned however, both this computer's and graphics architecture lacks the necessary processing power for handling tasks such as image-processing and neural network training which will later be discussed.

Upon completing further research and narrowing down the choices of hardware, it was evident that one of the two Jetson TX Models, TX1 and TX2, would be more of an appropriate hardware component against the Raspberry Pi 3 Model B. Research has given the design team confidence in the Jetson TX2 embedded module. We strongly believe that the TX2 will significantly aid in accomplishing our complex tasks and meeting our specific performance requirements whereas, the Pi 3 would significantly lack on a production level.

3.3 Strategic Components and Parts Selections

3.3.1 Development Boards

As mentioned in the previous section, development boards have become an electronic necessity for most sensor applications for electronic projects. The good thing about development boards if that they have been around for years and have been redeveloped almost to perfection and versatile to work with almost any project you can create. The bad thing about development kits is how many options there are out there. For this project, research was done only on the most reliable and popular brands.

3.3.1.1 Arduino

The Arduino family of microcontroller development kits is an ideal choice when you want to communicate with a wide array of electronic components and manage the intercommunication on a swift, robust basis (ARDUINO UNO REV3, n.d.). Arduino offers numerous models of its popular microcontroller-based development kits. The most versatile board offered by Arduino is the Arduino Uno Rev3.

The Arduino Uno Rev3 Microcontroller does not come with an operating system, but it does have limited on-chip RAM/Flash memory (ARDUINO UNO REV3, n.d.). This system also does not have internet access, but is perfect for our application with sensors. This is because of the basic functionality needed from the microcontroller. To read in data collected by the sensors for object placement on the course. This board is also recommended for beginners creating their first electronic products. This is helpful since our group has never attempted a project like this before and there are large number of resources on the internet for programming and tips. More details regarding this particular development kit is shown below in Table 4.

Table 4. Technical Specifications for Arduino Microcontroller (ARDUINO UNO REV3, n.d.)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3.3.1.2 Raspberry Pi

The next most popular development kits on the market is made by Raspberry Pi. Raspberry Pi is capable of running an entire operating system on top of it (Linux, Raspbian, Windows or even Android), making it an ideal choice for applications where you need extra-powerful computing, or want to integrate multiple web-based services (Microcontrollers – Types & Applications, 2018). The most popular model from Raspberry Pi is the Raspberry Pi 3 Model B+.

Unlike the Arduino, the Raspberry Pi 3 Model B+ Microcontroller has internet capabilities, which means it can be controlled remotely. This system also has much more available memory compared to the Arduino, however a downside of this system is the power consumption. Mostly this system is recommended for enthusiasts who are building a project that requires significant computational power, and that have a background in computer programming (RASPBERRY PI HARDWARE, n.d.). For our application on this project with the sensors, it was discovered that the Raspberry Pi 3 Model B+ requires a sensor adaptor for data transmission. More details regarding this particular development kit is shown below in Table 5.

Table 5. Raspberry Pi 3 Model B+ Technical Specifications (RASPBERRY PI HARDWARE, n.d.)

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

3.3.1.3 Texas Instrument BeagleBone

Lastly, the next most popular brand for microcontrollers is BeagleBone by Texas Instruments. BeagleBone is mainly used for prototyping advanced applications like gesture recognition, face detection or even underwater exploration (BeagleBone Black Development Board, 2018). Most popular model of BeagleBone's microcontrollers is the BeagleBone Black.

Compared to Raspberry Pi, BeagleBone Black offers nearly 1.5 times more GPIO pins (65 digital pins), and comes with 4GB onboard storage memory, with no SD card required (BeagleBone Black Development Board, 2018). BeagleBone is an open-hardware, open-software computer which supports operating systems like, Ubuntu, Android (official) and Linux. This system is the closest thing to having a mini computer on your project. It offers much more interfacing capabilities than Raspberry Pi and Arduino. This system is recommended not only for advanced users, but for advanced systems. This is due to the power of the system, which also means this system pulls higher voltage and current then the Raspberry Pi 3 Model B+ and the Arduino. More details regarding this particular development kit is shown below in Table 6.

Table 6. Texas Instrument BeagleBone Black Technical Specifications (BeagleBone Black Development Board, 2018)

Board size	3.4" x 2.1"
DDR memory	512 MB
Development environment	Fully functional terminal interface directly in the browser and the ability to run Python, Ruby and INO Sketches directly in the Cloud9 IDE, in addition to JavaScript on Node. JS and in your web browser
Ethernet	On-chip 10/100 Ethernet
Memory	Memory: 4GB eMMC memory that's pre-loaded with Debian GNU/Linux™ distribution and that frees up your microSD card slot
Power Options	Via USB or 5V DC input
Price (USD) Per Unit	$55.00 (Suggested Retail Price)
Processor	1GHz AM3359 Sitara ARM Cortex-A8
USB	1-port USB 2.0 Host and 1-port USB 2.0

Summary

After reviewing all the options for developments and their technical specifications, our group decided on the Arduino Uno Rev3 as the correct microcontroller for our application. This choice was made due to budget and power constraints. We do not need an excessively powerful board for data collection from the sensors. We also considered the power consumption of the board since the whole vehicle needs to be powered by solar power. We will have multiple components attached to the batteries so low power consumption and current is a critical consideration for this project.

Lastly, we considered the ease of programming. Since none of us have previous experience with using a microcontroller development kit, the ease and available resources for Arduino projects was appealing. The below table shows a direct breakdown of the three microcontroller options. The option highlighted was the selected component shown below in Table 7 and Figure 8.

Figure 9. Arduino Uno Rev3 Microcontroller

Table 7. Microcontroller Comparison and Component Selection

Microcontroller	Arduino Uno Rev3	Raspberry Pi 3 Model B+	BeagleBone Black
Manufacturer	Arduino	Raspberry Pi	Texas Instrument
Size (inch)	2.70 x 2.10	3.4" x 2.1"	3.4" x 2.1"
Price	$22.00	$55.00	$55.00

3.3.2 Ultrasonic Sensors

For the project in scope, Ultrasonic Sensors were selected for the task of detecting and measuring distances to an object. The following criteria will be taken into consideration when selecting the sensors:

- Detect objects at large distances (> 3-ft)
- Measure distance to objects.
- Wide beam angle (>15^{0})
- Low cost (< $40.00 per unit)
- Unaffected by changes in lighting conditions

This section will focus on highlighting the features of some the most popular, reliable, and cost-effective Ultrasonic sensors on the market including, HC-SR04 by SparkFun Electronics Co, PING))) by Parallax Inc, and MB1000 by MaxBotix.

3.3.2.1 HC-SR04 by SparkFun Electronics Co.

The HC-SR04 Ultrasonic ranging module is the most popular option among beginner projects mostly due to it being quite inexpensive, $3.95 before taxes. According to the datasheet provided by the vendor, it provides 2cm – 400cm (0.07 – 13.1 ft) non-contact measurement (Kinney, 2001). The module includes ultrasonic transmitter, receiver, and control circuit with a resolution of 3mm. The operation of the sensor consists of sending a

40 kHz signal for 10 µs and listening for the echo. It comes with a working angle of 15^0. This particular sensor uses four pins for 5V supply, trigger pulse input, echo pulse output, and ground (Kinney, 2001). The datasheet does not provide any more information regarding the operation of the sensor including test data. In order to attain more information, a sensor would have to be purchased in advance for testing purposes.

3.3.2.2 PING))) by Parallax Inc

The Parallax PING))) ultrasonic distance sensor provides accurate distance measurements from 2 cm (0.8 inches) to 3 meters (~9 feet) (Kinney, 2001). It is also a popular choice providing more flexibility than the HC-SR04 but at a higher cost ($29.99 before taxes). It includes a bidirectional TTL pulse interface with a single I/O pin that can be used with microcontrollers that use 5V TTL or 3.3V CMOS. This sensor also operates by sending a 40 kHz burst with minimum duration of 2 µs, but 5 µs being more typical (Kinney, 2001).

The datasheet provides ample amounts of information giving the user realistic expectations. The sensor cannot accurately measure distance to an object that is more than 3 meters away, has a shallow angle that reflects the sound away from the receiver, or too small to reflect enough sound back to the sensor. The datasheet provides data for two different tests performed in the Parallax lab. The two tests provide the working angles at different distances for both a cylindrical object and a flat object respectively.

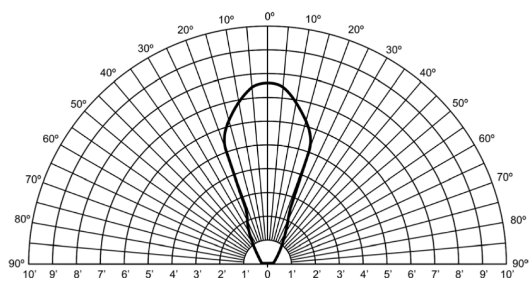


Figure 10. 3.5 in diameter cylinder, 4 ft. tall – vertical. 40 in sensor elevation (Kinney, 2001)

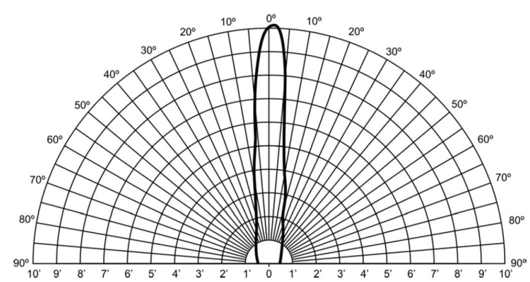


Figure 11. 12 x 12 in. cardboard, on 1" pole. 40 in sensor elevation (Kinney, 2001)

For round objects, it can be observed that the sensor offers a 40^0 total working angle up to 5.5 ft., gradually decreasing as the distance approaches 8 ft (Kinney, 2001). For flat objects, the working angle decreases `as the distance increases. Parallax provides example programs to get a project started along with accessories that can be used with the sensor, e.g. mounting brackets.

3.3.2.3 MB1000 LV-MaxSonar-EZO by MaxBotix

LV-MaxSonar-EZ Series by MaxBotix provides a high-performance range finder with 2.5V – 5.5V operating power. It can detect objects from 0 to 6.45 meters (~21 feet) and provide range information from 15 cm – 6.45 meters at a cost of $29.95 before taxes (Kinney, 2001). It provides readings at a rate of 20-Hz in a continuous manner. It includes interface output formats like pulse width output, analog voltage output, and RS232 serial output. The sensor comes with a 7-pin layout that allows flexible operations (Kinney, 2001). The datasheet provides three different wiring diagrams, pertaining to multiple sensors in a single system, in order to avoid interference (cross-talk) between sensors. This information covers all LV-MaxSonar-EZ Series; however, they differ on the beam pattern. The MB1000 has the widest beam in the series, and it is advertised for people detection applications. The following chart shows the beam characteristics of the sensor on a 30 cm grid. The dark line represents beam characteristics at 5V and the red dots the beam characteristics at 3.3V (Kinney, 2001).

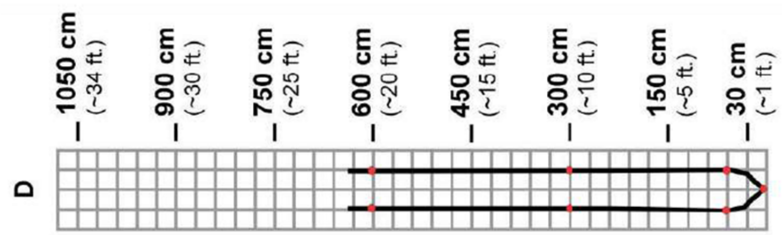


Figure 12. 11”. wide board (Kinney, 2001)

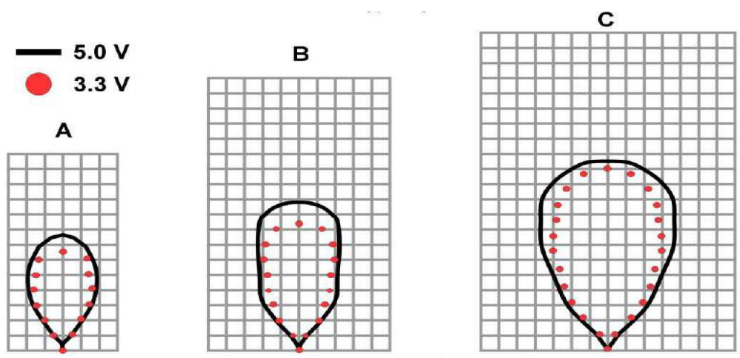


Figure 13. A: 6.1 mm dowel, B: 2.54 cm dowel, C: 8.89 cm dowel (Kinney, 2001)

The datasheet suggests that people detection pattern typically falls between charts A and B. Using the properties of a right triangle, the working angles of the sensor can be calculated as following. For ¼ in. diameter object, the effective working angle is ~43^0 up to 1.5 meters, gradually decreasing as distance increases. For 1 in. diameter object, the effective working angle is ~67^0 at 0.9 meters, gradually decreasing to ~25^0 by 2.7 meters. For a 3.5 in. diameter object, the effective working angle is ~53^0 up to 2.1 meters, gradually decreasing as distance increases. Flat objects can be detected at a distance of up to 6 meters with a minimal working angle (< 6^0) (Kinney, 2001).

Summary

The sensor chosen for our project application is the MB1000 Ultrasonic sensor shown below in Figure 14.

Figure 14. MB1000 (Kinney, 2001)

The following table summarizes the product and features of the HC-SR04, PING))), and MB1000 Ultrasonic sensors.

Table 8. HC-SR04 vs. PING))) vs. MB1000

	HC-SR04	PING)))	MB1000
Supplier	SparkFun	Parallax	MaxBotix
Operating Voltage	DC 5V	DC +5V	DC 2.5V - 5.5V
Operating Current	15 mA	30 mA	2 - 3 mA
Op. Frequency	40 kHz	40 kHz	42 kHz
Max Range	4 m	3 m	6.45 m
Min Range	2 cm	2 cm	0 cm
Working Angle	30 degrees	40 degrees	54.6 degrees
Dimension	45x20x15 mm	46x22x16 mm	22.1x19.9x15.5 mm
Weight	8.5 g	9 g	4.3 g
Op. Temperature	Unknown	0 - 70^0 C	-40^0 - 70^0 C
RoHS Compliant	Yes	Yes	Yes
Price	$3.95	$29.99	$29.95

3.3.3 Inertial Measurement Units (IMU)

This section will analyze some of the popular IMUs on the market. The criteria to be followed to select the appropriate IMU for this project is as follows:

- Degrees of Freedom
- Precision
- Cost (< $50.00)
- Documentation
- Support

The IMUs in scope are the 10-DOF Mems IMU sensor by DFRobot, GY-521 by Generic, and BNO055 9-DOF IMU breakout Board by Adafruit.

3.3.3.1 10-DOF Mems IMU Sensor V2.0 (SEN0140)

The SEN0140 is a very popular IMU featured in most self-balancing vehicles like the Segway Personal Transporter (Inertial Measurement Units (IMU), n.d.). It can be used in many applications like aircraft, balancing robots, indoor inertial navigation, and Human-Computer Interaction (HCI). It features a wide power input range from 3V – 8V, low noise low-dropout (LDO) regulator, I2C interface, and M3x2 mounting holes. It comes at the low price of $26.50 before taxes. The SEN0140 integrates the following 10 DOF sensors: ADXL345 Accelerometer, ITG3200 Gyro, HMC5883L Compass, and BMP280 pressure sensor (10 DOF Mems IMU Sensor V2.0 SKU: SEN0140, 2017).

The ADXL345 is a small, thin low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g manufactured by Analog Devices, Inc. The ITG3200 is a single-chip, digital-output, 3-axis MEMS gyro IC manufactured by InvenSense, Inc (Inertial Measurement Units (IMU), n.d.). It features enhanced bias and sensitivity temperature stability, reducing the need for user calibration. The HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with digital interface for applications such as compassing and magnetometry. It is manufactured by Honeywell International, Inc and enables a 1^0 to 2^0 compass heading accuracy. The overall size of the IMU is 28x18 mm and is compatible with Arduino micro-controllers (Inertial Measurement Units (IMU), n.d.). DFRobot provides sample codes to get the basic functions of the SEN0140 working, as well as resources for related projects using the same technology the sensor provides.

3.3.3.2 9-DOF Absolute Orientation IMU (BNO055)

The BNO055, manufactured by Bosch and supplied by Adafruit, is an IMU that takes a MEMS accelerometer, magnetometer and gyroscope and puts them in a single die with a high-speed ARM Cortex-MO based processor to digest all the sensor inputs and output data that can be used in quaternions, Euler angles or vectors (Inertial Measurement Units (IMU), n.d.). It offers an extensive number of features including: three axis orientation based on a 360^0 sphere, four-point quaternion output, three axis of rotation speed in radians, three axes of acceleration (gravity + linear), three axis of magnetic field sensing, three axis of linear acceleration, three axis of gravitational acceleration, and temperature measurement in Celsius.

The BNO055 is considered a System in Package (SiP), with an integrated triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a range of ±2000 degrees per second, a triaxial geomagnetic sensor, and a 32-bit cortex MO+ microcontroller running Bosch Sensortec sensor fusion software, all in a single package (Inertial Measurement Units (IMU), n.d.). For easier system integration it provides a digital bi-directional I2C and UART interfaces. The BNO055 has a voltage range of 2.4V - 3.6V at a price of $34.95 before taxes (Inertial Measurement Units (IMU), n.d.). The operating temperature ranges from -40^0 - 85^0 Celsius and a footprint of 3.8 x 5.2 x 1.13 mm (Inertial Measurement Units

(IMU), n.d.). Since it is a product from Adafruit, it comes with a lot of support for integration with systems using Raspberry Pi or Arduino microcontrollers.

3.3.3.3 6-DOF GY-521 MPU-6050

The GY-521 is a popular and rather inexpensive IMU sold by Generic that costs only $5.49 before taxes (Inertial Measurement Units (IMU), n.d.). It comes equipped with an MPU-6050, manufactured by InvenSense, Inc, which is the "working horse" behind this IMU. The MPU-6050 has an integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP). It has a dedicated I2C sensor bus that can directly accept input from an external 3-axis compass to provide a complete 9-DOF (Inertial Measurement Units (IMU), n.d.). It features six 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope and accelerometer outputs. The user programmable gyroscope has a range of ±250, ±500, ±1000, and ±2000 ^{0}/sec (dps) and the user programmable accelerometer has a range of ±2g, ±4g, ±8g, and ±16g. The GY-251 has an operating voltage range of 2.375V – 3.46V and a dimension of 4 x 4 x 0.9 mm (Inertial Measurement Units (IMU), n.d.).

Summary

The IMU sensor chosen for our project application was the 9-DOF BNO055 shown below.

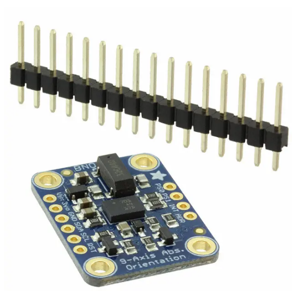


Figure 15. 9-DOF BNO055 (Inertial Measurement Units (IMU), n.d.)

The following table summarizes the features of the three IMU sensors that were chosen to perform an analysis on.

Table 9. SEN0140 vs. BNO055 vs. GY-521

	SEN0140	BNO055	GY-521
Supplier	DFRobot	Adafruit	Generic
Operating Voltage	3V - 5V	2.4V - 3.6 V	2.375V - 3.46V
Operating Current	6.6 mA	12.3 mA	4.1 mA
Low Power Mode	Yes	Yes	Yes
Interface	I2C	HID-I2C, I2C, UART	I2C
Deg. of Freedom	10	9	6
Dimension	28x18x _ mm	3.8 x 5.2x1.13 mm	4x4x0.9 mm

Op. Temperature	-40^0 - 85^0	-40^0 - 85^0	-40^0 - 85^0
RoHS Compliant	Yes	Yes	Yes
Price	$26.50	$34.95	$5.49

3.3.4 Battery Storage

All of the electrical components within our Solar Vehicle will be powered by batteries. In this section we will discuss the different types of batteries and their uses. Since our system in solar powered we already know that whichever battery we select will need to be rechargeable. Each of the mechanical engineering groups will be providing the battery for the motor of their vehicles. If any other batteries are needed for the system it will be provided by our group and then transported to each one of the mechanical engineering vehicles.

Something we need to keep inconsideration is that each battery needs to be powerful enough to supply power to all the electrical components attached to it. One battery power system will be connected to our motor subsystem and the other connected to the electronic and steering subsystem.

3.3.4.1 Battery - Lithium Ion

A lithium-ion battery is an advanced battery technology that uses lithium ions as component of its electrochemistry (Learning the Basics about Batteries, 2018). It uses a cathode, an anode and electrolyte as conductor. During discharge cycle, the ions move from the anode to the cathode passing by an electrolyte. Then, the charge changes its direction, and ions move from the cathode through the anode. Li-ion batteries use an intercalated lithium compound as one electrode material. Figure 22 below shows the process.

When the battery is discharging, and providing an electric current, the ions travel from the anode to the cathode, generating a flow of electrons from one side to the other. When plugging in the device, the opposite happens (Learning the Basics about Batteries, 2018). Lithium ions batteries use different materials as an electrolyte. the table below conclude the characteristics of li-ion with different electrolyte materials.

- Advantages of lithium-ion battery (Learning the Basics about Batteries, 2018)
  - High energy density
  - Low self-discharge
  - Cells provide very high current to applications
  - Low Maintenance

- Disadvantages of lithium-ion battery (Learning the Basics about Batteries, 2018)
  - Requires protection circuit to maintain voltage and current within safe limits.
  - Subject to aging, even if not in use
  - Transportation restrictions

– Expensive to manufacturer

Table 10. Characteristics of the four most commonly used lithium-ion batteries (Learning the Basics about Batteries, 2018)

Specifications	Li-cobalt	Li-manganese	Li-phosphate	NMC [1]
Voltage	3.60V	3.70V	3.30V	3.60/3.70V
Charge limit	4.20V	4.20V	3.60V	4.20V
Cycle life [2]	500	500–1,000	1,000–2,000	1,000–2,000
Operating temperature	Average	Average	Good	Good
Specific energy	150–190Wh/kg	100–135Wh/kg	90–120Wh/kg	140Wh/kg
Specific power	1C	10C, 40C pulse	35C continuous	10C
Safety	Average. Requires protection circuit and cell balancing of multi cell pack. Requirements for small formats with 1 or 2 cells can be relaxed		Very good, needs cell balancing and V protection	Good, needs cell balancing and voltage protection
Thermal runaway [3]	150°C (302°F)	250°C (482°F)	270°C (518°F)	210°C (410°F)
Cost	Raw material high	Material 30% less than cobalt	High	High
In use since	1994	2002	1999	2003
Researchers, manufacturers	Sony, Sanyo, FDK, Saft	NEC, Samsung, Hitachi	UT, QH, MIT A123, Valence	Sony, Sanyo, Nissan Motor
Notes	Very high specific	High power, average to high	High power, average	Very high specific

	energy, limited power; for cell phones, laptops	specific energy, power tools, medical, EVs	specific energy, higher self-discharge than other Li-ion	energy, high power; tools, medical, Evs

3.3.4.2 Battery – Lead Acid

Lead acid batteries are made from a lead alloy. A full discharge of lead acid battery causes an excessive demand on its strength, and every charge/discharge cycle minimize the capacity of the battery. Charging a lead acid battery is simple, but we must pay attention to the voltage limit. Using a low voltage protect the battery, but this produces poor performance. Using A high voltage improves performance but forms grid corrosion on the positive plate. There two type of lead acid battery: Deep-cycle Battery and Starter and Deep-cycle Batteries

- Advantages (Learning the Basics about Batteries, 2018)
  - Inexpensive and simple to manufacture
  - Low self-discharge
  - High specific power
  - Good low and high temperature performance

- Disadvantages (Learning the Basics about Batteries, 2018)
  - Low specific energy; poor weight-to-energy ratio
  - Slow charge; fully saturated charge takes 14-16 hours
  - Must be stored in charged condition to prevent sulfation
  - Limited cycle life; repeated deep-cycling reduces battery life
  - Not environmentally friendly

3.3.4.3 Nickel–Metal Hydride Battery

Nickel-Metal hydride battery is one of the most readily available rechargeable batteries for consumer use. They also offer higher performance at higher cost than lead-acid batteries. They have good cycle life and capacity. The only problem with this battery is high self-discharge. Nickel-Metal Hydride battery cell composed of a positive electrode and a negative electrode with a separator in between. The positive electrode contains nickel hydroxide. The negative electrode contains hydrogen that absorbs nickel alloys. The separator is alkaline electrolyte and a vented metal case.

- Advantages (Learning the Basics about Batteries, 2018)
  - Simple storage and transportation
  - Environmentally friendly
  - Nickel content makes recycling profitable
  - Wide temperature range

- Disadvantages (Learning the Basics about Batteries, 2018)
  - Limited service life
  - High self-discharge
  - Generates heat during fast charge and high-load discharge
  - Does not absorb overcharge well

3.3.4.5 Performance Comparison

- Energy Density and Specific Energy: Lead–acid has the lowest capacity at 35 Wh/kg and Ni–MH is around 75 Wh/kg. The Li-ion battery yields a much larger 150 Wh/kg specific energy. Li-ion batteries have an advantage in weight- and volume-sensitive applications (IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead Acid Batteries Used in Stand Alone Photovoltaic (PV) System, 2014).

- Cycle Life: All the three batteries have a good life cycle. Li-ion cells have the longest cycle life. At 100% DOD they typically last 3000 cycles at low discharge/charge rates and room temperature. Ni–MH batteries has around 500 cycles at 80% DOD at 0.2C charge/discharge and room temperature. For lead–acid batteries, 200 cycles are typical at 100% DOD (Learning the Basics about Batteries, 2018).

- Temperature Operating Range: For lead–acid batteries, charge and discharge temperatures should be limited to an operating range of −40–60◦C. Li-ion batteries have an operating range of −20–60◦C. Ni–MH have the narrowest operating range of −20–45◦C

Table 11. Battery Type Comparison and Component Selection

Specification	Lead-Acid	Ni-MH	Li-ion
Specific Energy (Wh/Kg)	30-50	60-120	100-150
Charge time	8-16h	2-4h	1-2h
Colombic Efficiency	~90%	~90%	99%
Cost	Low	Moderate	High

3.3.5 Voltage Regulator

Every electronic circuit is designed to operate at supply voltage, which is usually assumed to be constant. A voltage regulator provides this constant DC output voltage and contains circuitry that continuously holds the output voltage at the design value regardless of changes in load current or input voltage. In this section we will talk about two type of voltage regulator: linear and switching regulator. [3]

3.3.5.1 Linear Regulator

A linear regulator operates by using a voltage-controlled current source to force a fixed voltage to appear at the regulator output terminal. The control circuitry must monitor the output voltage, and adjust the current source to hold the output voltage at the desired value. The design limit of the current source defines the maximum load current the regulator can source and still maintain regulation. The output voltage is controlled using a feedback loop, which requires some type of compensation to assure loop stability. Most linear regulators have built-in compensation, and are completely stable without external components. Some regulators, do require some external capacitance connected from the output lead to ground to assure regulator stability (Simpson, 2011).

There are three type of linear regulator: Standard (NPN Darlington) Regulator, Low Dropout or LDO Regulator, and Quasi LDO Regulator.

Standard (NPN Darlington) Regulator uses the NPN Darlington configuration. It is a special arrangement of two standard NPN or PNP bipolar junction transistors (BJT) connected. The Emitter of the first transistor is connected to the Base of the second to produce a more sensitive transistor with a much larger current gain. A Darlington Transistor configuration, also known as a Darlington pair or super-alpha circuit, consist of two NPN or PNP transistors connected together so that the emitter current of the first transistor TR1 becomes the base current of the second transistor TR2. Then transistor TR1 is connected as an emitter follower and TR2 as a common emitter amplifier (Simpson, 2011).

The Low-dropout (LDO) regulator differs from the Standard regulator in that the pass device of the LDO is made up of only a single PNP transistor. They offer very low dropout, fast transient response, and excellent line and load regulation, and they have a very wide input voltage range (0.9 V to 80 V) (Simpson, 2011). Output currents range from 100 mA to 10 A, with positive, negative, and multiple outputs. The minimum voltage drop required across the LDO regulator to maintain regulation is just the voltage across the PNP transistor: VD(MIN) = VCE (Simpson, 2011).

Another regulator configuration that is becoming very popular in certain applications (like 5 – 3.3V conversion) is the quasi-LDO regulator. The quasi-LDO is so named because it is 'half way' between the NPN Darlington and the true LDO. The pass transistor is made up of a single NPN transistor being driven by a PNP. The minimum voltage drop required across the Quasi-LDO regulator to maintain regulation is given by: VD(MIN) = VBE + VCE (Simpson, 2011).

- Advantages of linear voltage regulator
  - Simple circuit configuration
  - Few external parts
  - Low noise

- Disadvantages of linear voltage regulator
  - Relatively poor efficiency
  - Considerable heat generation
  - Only step-down (buck) operation

3.3.5.2 Switching Regulator

A switching regulator is a voltage regulator that uses a switching element to transform the incoming power supply into a pulsed voltage, which is then smoothed using capacitors, inductors, and other elements. Power is supplied from the input to the output by turning ON a switch (MOSFET) until the desired voltage is reached. In recent years, switching regulators have become more popular because semiconductor technology is more advanced now. They provide very high switching speed and very high-power handling capabilities. Today, it is easy to design switched-mode power supply with an efficiency greater than 90% with low cost, small size, and light weight. Because they use power semiconductor devices, they can operate in either the on-state or the off-state. There are many types of switching regulator.

3.3.5.2.1 Buck Regulator (step down)

Buck regulator is the most common used switching regulator, which is used to down-convert a DC voltage to a lower DC voltage of the same polarity. This is essential in systems that use distributed power rails, which must be locally converted to 15 V, 12 V or 5 V with very little power loss. The Buck converter uses a transistor as a switch that alternately connects and disconnects the input voltage to an inductor (Simpson, 2011).

3.3.5.2.2 Boost Regulator (step up)

The main application of boost regulator is to regulate dc power supply which means to produce a dc output voltage greater in magnitude than the dc input voltage. The other application of this regulator is in the regenerative braking of dc motors. When the switch is on, the diode is reversed biased, thus isolating the output voltage, and the input supplies energy to the inductor. When the switch is off, the output receives energy from the inductor as well as from the input (Simpson, 2011).

3.3.5.2.3 Buck-Boost regulator

The main application of Buck-Boost regulator is in regulated dc power supply, where a negative output is desired. Its voltage gain can be greater than, equal or less than 1. Buck-Boost regulator can be obtained by cascading the step-down converter and the step-up converter (Simpson, 2011). The output to input voltage conversion ratio is the product of the conversion ratio of the two converters in cascade: $Vo/Vd = D/(1-D)$ so when D is greater than 0.5 it is Boost (Simpson, 2011). When D is less than 0.5, it is Buck. When D equal 0.5, it is unity gain. This regulator composed of a switch. When this switch is off, the input provide energy to the inductor and the diode is reversed bias. When the switch is on, the energy stored in the inductor transferred to the output.

3.3.5.2.4 CUK converter

CUK converter provides a negative polarity regulated output voltage. It is obtained by using duality principle on the circuit of Buck-Boost regulator. It requires two switching and uses two inductors and a capacitor to store and transfer energy from the input to the output. Its voltage gain can be greater than, equal or less than 1 (Simpson, 2011).

- Advantages:
  - High efficiency
  - Low heat generation
  - Boost/buck/negative voltage operation possible

- Disadvantages
  - More external parts required
  - Complicated design
  - Increased noise

3.3.5.2.5 Regulator Comparison

The linear regulator is easy to use, and it is a good choice for powering very low powered devices where the difference between input and output is very small. They are also simple and cheap. The only downside to using a linear regulator is that they are not efficient. On the other hand, switching regulators are highly efficient and can be divided into isolated and non-isolated partitions.

Table 12. Voltage Regulator Comparison and Component Selection

Regulator	Linear Regulator	Switching Regulator
Efficiency	Low to medium	High
Complexity	Low	Medium to high
Total cost	Low	Medium to high
Noise	Low	Medium to high

3.3.6 Cameras

This section will talk about the different camera options considered to aid in the image processing (recognition) that will determine path planning, autonomous navigation, object detection and avoidance.

Though there are several hardware components readily available, we will discuss the specifics needed that will help the robot perform its task down to a mission-critical response time. A few of the main specifics we had to take into consideration when selecting the hardware component for image recognition included but are not limited to: Frames per second(fps) measured or captured, field-of-view (fov) and sunlight sensitivity. For the fov and based on our calculations, we are primarily seeking a camera that contains a fov of 106.26 degrees from diagonal to diagonal.

When it comes to modern day television, films, motion pictures, and even the video that displays on the computer, frame rate is the number images or frames shown or projected per second (ileCAM30_TX2 - 3.4 MP GMSL Camera (supports Upto 15 Meters), n.d.). In the United States 30 fps is the professional frame rate for television. Field of view (fov) is essentially the angle in degrees view of what the camera sees from a specific focal point.

3.3.6.1 Standard Camera

Both the Jetson TX2 and TX1 development kits comes with a 5-mega pixel Camera Serial Interface (CSI) standard camera module, natively built onto the development module. The purpose for the camera is to act a sensor. Under close ranges, this camera can be used to detect humans, animals, and other objects. This immediate section will discuss the onboard TX development board camera module and compare it to other standard camera sensors and devices.

3.3.6.1.1 5MP CSI Camera module (Omni Vision OV5693)

Since this camera comes mounted onto the camera expansion our team will not have to worry about additional installation parts or software to make the camera work. It is expected that this 5MP camera module shall act as the ‘eyes’ of the autonomous solar vehicle. Given that the camera module comes already built into the TX2 embedded system we can benefit from the fact that there will be minimal latency with a quick response when it comes to detecting individuals amongst other objects. The camera is set in a fixed position on the TX2 development board so, we would only have to account for possible vibration, however, the solar vehicle robot will only be traveling at a maximum speed of 5mph. This possible issue of experiencing vibrations may be minimal.

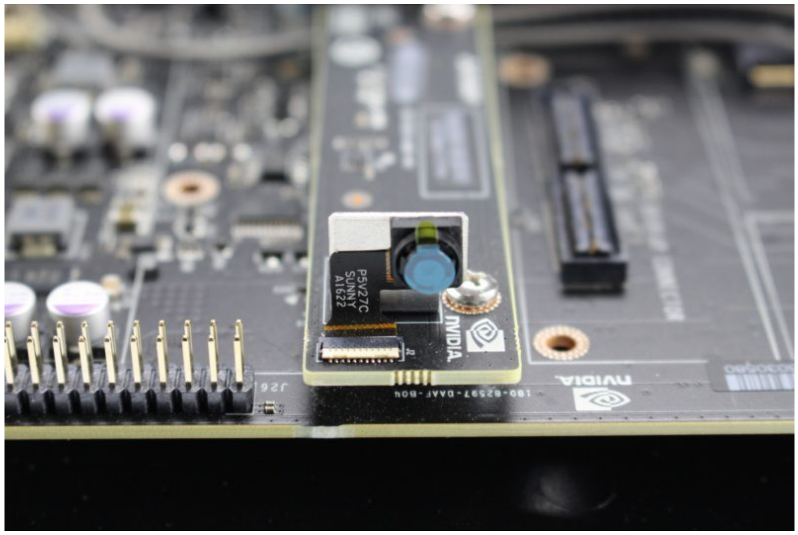


Figure 16. Close-up view Onboard OV5963 (NVIDIA, 2015)

The camera is set in a fixed position on the TX2 development board so, we would only have to account for possible vibration, however, the solar vehicle robot will only be traveling at a maximum speed of 5mph. This possible issue of experiencing vibrations may be minimal. The onboard camera has a diagonal field of view of 63 degrees, which in turn, is 126 degrees diagonal to diagonal this meeting our main camera vision requirement. The camera requires .250W to be active (NVIDIA, 2015). The camera module can operate effectively and take stable images as long as it is within the operating temperature range of 0 – 50 degrees Celsius. The fixed focus 5mp onboard camera module can view objects from 100mm to Infinity based on the specifications provided below.

Table 13. OV5963 Technical Specifications (NVIDIA, 2015)

- OmniVision® OV5680 CMOS image sensor
- Focus Type – Auto Focus (VCM driven)
- 2 lane MIPI® CSI-2 interface
- Output Format - RAW RGB (Bayer pattern)
- View Angle - 63 ° Diagonal
- Object Distance - 100mm to Infinity
- Effective Focal Length - 3.42 mm
- Max S/N ratio- 38 dB
- Dynamic Range - 73 dB
- Temperature Range
- Operation - -30° to 70° C
- Stable Image - 0° to 50° C
- Power requirements
- Active - 250 mW
- Module Dimensions (in mm)
- excluding flex cable - 8.5 x 8.5 x 5.15
- with flex cable - 20 x 9.6 x 5.75

3.3.6.1.2 NileCAM30_TX2 – 3.4MP GMSL Camera

The NileCAM30_TX2 comes with four boards including a TX2/TX1 base board and camera module itself. The NileCAM30_TX2 operates off of the AR0330 CMOS image sensor from ON Semiconductor (ileCAM30_TX2 - 3.4 MP GMSL Camera (supports Upto 15 Meters), n.d.).The 3.4Mega-pixel camera comes with both 3 and 15 meters of coaxial cable connected to a serializer and deserializer board. The cost of this camera module is $429USD.

The output format of the camera module is UYVY with an S-mount holder. The AR0330 is a 1/3" optical format CMOS Image sensor with fixed focus. With the lens being supplied at purchase, the display field-of-view (DFOV) for the 3.4MP resolution camera is 125 degrees. The camera uses 4-lane MIPI CSI-2 interface that plugs directly onto either the Jetson TX1 or TX2 development board. The maximum image transfer rate at HD (1280 x 720) resolution processes 60 fps while the full resolution is only processing at 24 fps. The operating temperature range for the camera module is -30 – 55 degrees Celsius (ileCAM30_TX2 - 3.4 MP GMSL Camera (supports Upto 15 Meters), n.d.).

3.3.6.1.3 e-CAM131_CUTX2 – 4K MIPI NVIDIA Jetson TX2/TX1 Camera

The e-cam131 is a 13MP camera solution for the Jetson TX1 and TX2 development boards. The board has 4-lane MIPI CSI-2 capabilities such that it can be housed or mounted to the camera connector on the Jetson TX2 and TX1 developer's kit. Base on e-con Systems, the e-CAM131 is a standard V4L2 device such that buyers the device's API to access the camera directly and control it as normal (NVIDIA, 2015).

The camera uses a 1/3.2" AR1335 CMOS image sensor with advanced pixel technology. The cost of the camera currently is $269USD (NVIDIA, 2015). Some of the key features of the e-CAM131 camera module is the 4-lane MIPI CSI-2 interface to connect with the development board, it connects perfectly to the Jetson TX1 and TX2 Developer's kit and lastly, it comes with an S-mount lens holder. The operating voltage for the camera module is 5v +/- 5%. The operating temperature range is listed at -30 degrees Celsius to 85 degrees Celsius with an approximate power consumption of 1.22W (NVIDIA, 2015).

The board is listed as a lightweight component weighing around 28 Grams with the lens included. The specification for this board does not list the DFOV, therefore we will not be considering this camera as that is one of the key features we are seeking.

3.3.6.1.4 Standard Camera Summary

The table shown below summarizes the features of each standard camera option available for our project.

Table 13. Standard Camera Comparison and Component Selection (NVIDIA, 2015)

Standard Camera	5MP CSI Camera(OV5693)	3.4MP NileCAM30_TX2	**13MP e-CAM131-CUTX2**
Size(mm)	20 x 9.6 x 5.75	30 x 30	75.03 x 40.18 x 25.6
Cost (USD)	On Board Camera	$429	$269
DFOV	63 (Diagonal)	125	Unknown
Power Consumption at 5V	250mW	2.9	1.25
CPU Interface	2-Lane MIPI CSI-2	4-Lane MIPI CSI-2	4-Lane MIPI CSI-2

3.3.6.2 Stereo Camera

Earlier in this document we discussed different methods for navigation and path planning. The most widely used computer system for this in everyday life is GPS or Global Positioning System. The problem with manipulating this technology for our project is the accuracy of the technology. Standard GPS has an accuracy of around 3 meters. This error field is too large for our application as we anticipate the obstacle being no wider then 2 meters.

A possible solution to this problem is using SLAM or Simultaneous Localization and Mapping. This is an emerging, open source technology, improving every day. SLAM (Simultaneous localization and mapping) is a technique used by robots and autonomous vehicles to build a map within an unknown environment, or to update a map within a known environment, while keeping track of their current location. This is done with a stereo camera.

A Stereo camera is a camera that has two lenses about the same distance apart as your eyes and takes two pictures at the same time (What is a Stereo Camera?, n.d.). This simulates the way we actually see and therefore creates the 3D effect when viewed. A standard camera cannot be used with SLAM because of the single lens. A single camera cannot handle depth perception. This is why a stereo camera is needed. The two lenses create a triangular point of view, allowing calculations to determine the distance an object is from the camera.

In this section, we will discuss different options for stereo cameras to use with our SLAM algorithm, allowing the solar vehicle to not only avoid objects, but potentially plan the most optimal path.

3.3.6.2.1 ZED Stereo Camera

The most advanced stereo camera available is the ZED from ZED Labs. Using advanced sensing technology based on human stereo vision, the ZED camera adds depth perception, positional tracking and 3D mapping to any application. ZED perceives the world in three dimensions. Using binocular vision and high-resolution sensors, the camera can tell how far objects are around you from 0.5 to 20m at 100FPS, indoors and outdoors (ZED Introduction, n.d.).

Figure 17. ZED Stereo Camera

The ZED camera can capture 2K 3D video with best-in-class low-light sensitivity to operate in the most challenging environments (ZED Introduction, n.d.). This is not a challenge for our project as we hope to operate the solar vehicle in bright sun conditions. ZED Labs does not release how the ZED stereo camera is constructed. This is important to make a fair comparison with other products on the market, but since they are not available we have to go by the information given by the company on the webpage. Below is a table with technical specifications listed from the website.

3.3.6.2.2 Tara USB 3.0 Stereo Camera

The Tara is a 3D stereo camera integrated with a MT9V024 stereo sensor. In an uncompressed format WVGA((2*752)X480 at 60 frames per second is supported by this camera sensor (Tara - USB 3.0 Stereo Vision Camera, n.d.). The Tara 3D stereo camera is ideal for the application we are attempting to develop which involves Machine or Robot vision, 3D video recording and Depth sensing (Tara - USB 3.0 Stereo Vision Camera, n.d.). The Tara models come with ROS support which demonstrates and provides adequate documentation for the Tara 3D stereo camera. The MT9VO24 is a CMOS active-pixel image sensor with high dynamic range (HDR) operation and a global shutter. At full resolution the sensor can display 60fps (Tara - USB 3.0 Stereo Vision Camera, n.d.).

3.3.6.2.3 Tara XL Stereo Camera

Another popular stereo lens camera being used for SLAM is the TARA XL. The TARA XL is a UVC- compliant 3D Stereo camera based on MT9V024 stereo sensor from ON Semiconductor at 60fps over USB 3.0 in uncompressed format. This Stereo camera provides two synchronized sensor frame data interleaved side by side to the host machine over USB 3.0 interface (Tara - USB 3.0 Stereo Vision Camera, n.d.).

The TARA XL has the capability to perform SLAM, but with research it is difficult to uncover documentation for algorithms or successful projects.

Within the hardware is an NVIDIA Compute Unified Device Architecture (CUDA) accelerated Standard Development Kit (SDK) that of course runs on the GPU of NVIDIA. The TaraXL uses the MT9V024 image sensor by OnSemi (Tara - USB 3.0 Stereo Vision Camera, n.d.). The CMOS active-pixel image sensor is 1/3-inch wide with a wide-VGA resolution. The default mode for output is 60 frames per second (fps) as a wide-VGA-size image (Tara - USB 3.0 Stereo Vision Camera, n.d.).

The cost of the TaraXL is $349USD. This camera is solely built and engineered for NVIDIA's GPU, more specifically, the Jetson TX2. Out of the box at 30 frames per second, the TaraXL provides depth mapping in 3D for developers to build custom solutions around (Tara - USB 3.0 Stereo Vision Camera, n.d.). The hardware is similar to that of its predecessor, the Tara, however, its firmware is where TaraXL shines. The TaraXL also comes with a ROS documentation for building applications.

The major difference between the TARA and the ZED is the strength of the lens and the internet connectivity. It also allows for full algorithm development. With this being the first application, our team is creating, it seems like this camera is more for the advanced programmer wanting no restrictions on program development. Again, there is limited information about the device, but listed below are the specifications taken for the product website.

- Houses 6-axis IMU
- USB 3.0 Interface and backward compatible with USB2.0 hosts with limited features and capabilities
- Unique ID for each camera
- WVGA 752(H) x 480(V) (360690 pixels) sensors output images in 8-bit/10-bit greyscale format
- Synchronous Parallel Monochrome 10bit video data at 60 fps per sensor
- Baseline distance of 60mm
- Depth ranges from 50cm to 300cm
- S-mount lens holder (M12 P0.5) with pre-calibrated S-mount lens pair
- External trigger support

Table 14. TARA XL Frame Rate Specifications (Tara - USB 3.0 Stereo Vision Camera, n.d.)

Frame Rate :

Format	Resolution	Over USB 3.0	Over USB 2.0 (only 8 bit)
WVGA	(2*752) x 480	60 fps	30 fps
VGA(Cropped)	(2*640) x 480	60 fps	30 fps
QVGA	(2*320) x 240	60 fps	60 fps

3.3.6.2.2 MYNT EYE Stereo Camera

The MYNT EYE Standard Edition utilizes the camera and the motion sensor to provide visually accurate SLAM results with higher precision, lower cost, simpler layout, along with the ability to achieve face and object recognition. The concept of combining binocular and IMU is the leading-edge technology in the current SLAM industry (MYNT EYE, n.d.). The description provided on the product website gives detailed information on the construction of the camera. It boasts, the ability to be used in Visual positioning navigation, including Visual real-time positioning navigation system of Driverless Vehicle and Robot, which is exactly the problem we are trying to solve with our solar vehicle (MYNT EYE, n.d.).

Figure 18. MYNT EYE Stereo Camera (MYNT EYE, n.d.)

In order to ensure the quality of the output data of the camera products, the MYNT EYE was calibrated with the binocular and IMU. The product has passed various hardware stability tests, such as high temperature and humidity continuous work and operation, low-temperature dynamic aging, high-temperature operation, low-temperature storage, whole-machine thermal shock, sinusoidal vibration and random vibration tests to ensure the stability and reliability of the product (MYNT EYE, n.d.). This is also a benefit to our application as the solar vehicle will not be equipped with a shock system and will be performing outdoors in high sunlight increasing the temperature the system is exposed to.

The developer also has multiple documents for installation and coding with autonomous projects.

Table 15. Stereo Camera Comparison and Component Selection

Stereo Camera	ZED	TARA XL	MYNT EYE
Manufacturer	Stereo Labs	e-con Systems	mynteyeai
Size (mm)	175x30	100 x 30 x 35	165x31.5x29.6
Cost	$449	$249	$249

3.3.7 GPUs: Single-Board-Computers vs. Supercomputers

In coursework, or self-developed projects, one may be familiar with the common processing performance measurements of a million-instructions-per-second or MIPS, however, this supercomputer's performance is measured in FLOPS, that is, floating-point operations-per-second. In the subsequent sections the Raspberry Pi 3, which is one type of an SBC along with the Jetson TX supercomputers, will be thoroughly broken down and explained in a manner that will convey the audience why the TX2, supercomputer, is the absolute better choice for completing our senior design project.

In that, the successive sections will express the researched information on both, the small Single Board Computer and supercomputer components, the Raspberry Pi 3 Model B and Jetson TX1 and TX2, respectively.

3.3.7.1 Raspberry Pi 3 Model B (SBC)

The Raspberry Pi Model B is an SBC, or Single-Board-Computer which essentially implies that the board functions completely as an actual computer which connects to the monitor of a computer. The Raspberry Pi also uses standard peripherals like a keyboard and mouse, for example. The inspiration of the Raspberry Pi family was to promote the Internet of Things (IoTs) exposing computer science to young people using a small high-performance computer at a low-cost budget (RASPBERRY PI HARDWARE, n.d.). The Pi out of the box is a small credit-card sized computer. The Pi 3 allows a user to install an operating system onto it at first-use. Although Raspbian is the official operation system for all

Raspberry Pi models, the Raspberry pi is essentially an open-source Linux-based computer where it can programmatically run in Python or Scratch. As one may already can tell, the Raspberry Pi 3 is the third generations of the Pi family.

As mentioned above, the Pi 3 is superb for demonstrating proof of concepts. A couple toplined features include the native Wi-Fi and Bluetooth capabilities, with a newly upgraded system-on-chip (SoC) Broadcom BCM2837 which enhances the processing speed of by %50 to that of the Raspberry Pi 2 (RASPBERRY PI HARDWARE, n.d.). The Raspberry Pi 2 has a processor speed of 900MHz where the Raspberry Pi 3 now has a processor speed of 1250MHz. The Pi 3 can be referred to as an actual desktop computer with its updated features operating at input power supply voltage of approximately 5volts at 2.5 amps (RASPBERRY PI HARDWARE, n.d.). Some additional, but relevant features we can conclude about the Pi 3 is that it being the latest model it was still able to maintain hardware size, GPIO structure and cost as its predecessors. Finally, the Model 3 can run the Robot Operating System (ROS) using Python and Scratch. The most widely used and native OS for the Pi is Raspbian.

Raspberry Pi 3 Model B Technical Specifications (RASPBERRY PI HARDWARE, n.d.)

- Broadcom BCM2837 64bit ARMv7 Quad Core Processor Powered Single Board Computer running at 1.2GHz
- 1GB RAM
- BCM43143 Wi-Fi on board
- Bluetooth Low Energy (BLE) on board
- 40pin extended GPIO 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source (now supports up to 2.4 Amps)

3.3.7.2 Jetson TX1

The Jetson TX1 is an Artificial Intelligence supercomputer embedded with modern-day technology for computer vision (CV), deep learning and GPU computing. The current price for the Jetson TX1 is $299. The TX1 contains NVIDIA Maxwell Architecture with 256 CUDA cores, Quad ARM A57 MPCore Processor at 1.73GHz, 16GB Flash storage and 4GB Low-powered double data rate (LPDDR) Memory (NVIDIA, 2015).

As a development kit, the TX1 comes pre-flashed with a Linux environment and a fully-loaded development platform for computing using AI. There is a 5MP fixed focus MIPI CSI camera module mounted onto the TX1 development board. The TX1 has 40 GPIOs, where 8 pins are signal specific, which can be used to power other development boards or microcontrollers that use 3.3 or 5.0 VDC. Some further features of the Jetson TX1 include an Encoder and Decoder (NVIDIA, 2015). The TX1 module processes (Encode & Decode) a JPEG at 600 Mega-Pixels per second. This comes in handy for deep learning and training. The imaging systems process up to 1400MP/sec with a support of up to 24MP sensor camera module that uses a dedicated YUV to RAW engine (NVIDIA, 2015).

The TX1 has a recommended operating input voltage range of 5.5 to 19.6 volts. The absolute maximum rating based on the TX1 module datasheet shows 30 input volts at a maximum of 3 amps (NVIDIA, 2015). The modules operating power is between the range of ~6.5 – 15W. The TX1 has a general power consumption of 10W under an input voltage of ~5.5 – 19.6 VDC. The operating temperature from the TTL or Thermal Transfer Plate is the range -25 to 80 degrees (Celsius) (NVIDIA, 2015).

3.3.7.3 Jetson TX2

The Nvidia Jetson TX1 and TX2 modules contains a powerful System-on-Chip (SoC) or processor called Tegra. In short, Tegra integrates an Advanced RISC Machine (ARM) architecture central processing unit (CPU), graphic processing unit (GPU), host-bridge and southbridge onto a computer. Concurrently, the Jetson Boards of this discussion are both able to run various software and operating systems including the Robot Operating System which again, is Linux-based.

Tegra (SoC)

Created by Nvidia this computer on a chip series integrates into a single package, a graphics processing unit (GPU), southbridge, (host-side) northbridge, ARM architecture CPUs and a controller of memory (NVIDIA, 2015). The Tegra chip is used on both.

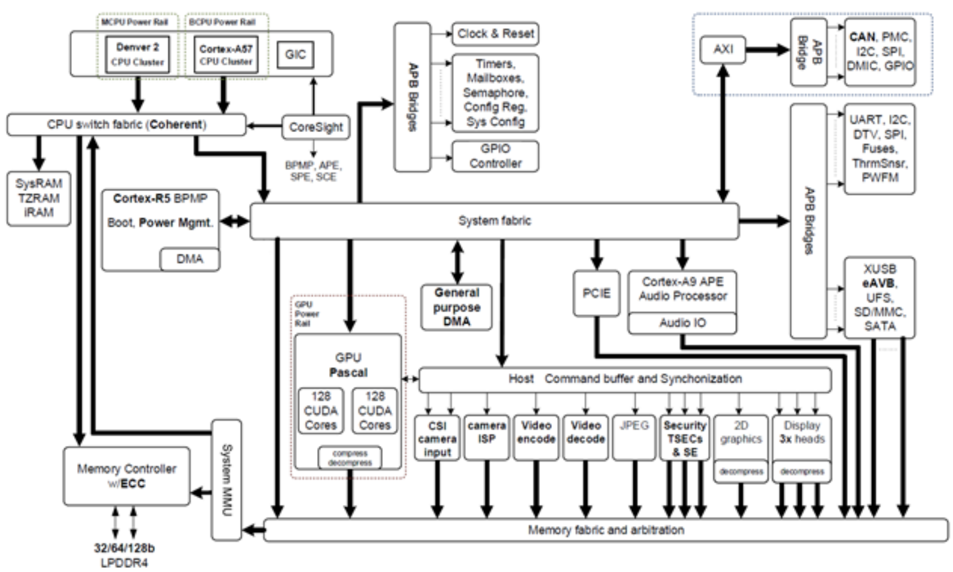


Figure 19. Tegra Parker Block Diagram (NVIDIA, 2015)

The Jetson TX2 is an Artificial-Intelligence supercomputer that is about the size of a single carton of cigarettes. This module is ideal for those interested in working on small, yet intelligent devices. Such devices may include but are not limited to smart camera devices, flying drones, hand-held medical device and lastly autonomous solar-powered vehicle robots. The AI system runs on Nvidia's Pascal GPU Architecture. It is listed that there are five technological advancements that the Pascal architecture is responsible for. [4]. Four of the five technological breakthroughs play a massive role that directly impacted our choice in going with the Jetson TX2 module for this project involving the autonomous robot. They are as listed below (NVIDIA, 2015):

- New artificial Intelligence (AI) algorithms
- 16 Nanometer FinFET for unprecedented energy efficiency
- An exponential boost in performance
- Maximum application scalability

New Artificial Intelligence (AI) Algorithms

Once the Jetson TX2 development kit is acquired we will begin the deep-learning process. According to Investopedia Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns (NVIDIA, 2015). In artificial intelligence, deep learning, which is also a subset of machine-learning, has networks with the ability to learn without any supervision from random or unlabeled data. Deep learning is also known as Deep Neural Network (DNN) or Deep Neural Learning (DNL) (NVIDIA, 2015).

The newly enhanced AI algorithms will allow the up to 256 (8-bit) instructions in the Pascal architecture thanks to having over 40 tera-operations per second of performance. Should we go into deep-learning this will enable and speed up the real-time responsiveness feature. For training, these advanced algorithms also allow over 20 teraflops of performance, which is essentially made for unique training.

One of the primary reasons why we selected the TX2 embedded platform is because of its deep learning capability. Choosing the TX2 was imperative for our design team's robot because we expect the robot to be able to perform image recognition. The image recognition will be conducted after using one of several deep learning frameworks.

16 Nanometer FinFET for unprecedented energy efficiency

The Pascal Graphics Processing Unit is currently the world's largest Fin Field-effect transistor. This is built on a substrate where a gate is stationed on two to four sides formed to create a double gated structure for more electrons to run freely, on the sides of a channel (NVIDIA, 2015). This Metal-oxide-semiconductor Field-effect-Transistor (MOSFET) fabrication consists of 150 billion transistors which promises that the fabrication technology will be most energy efficient for the described workloads that will also give the best performance for our computing needs.

An Exponential Leap in Performance

With having the most powerful architecture for High Performance Computing (HPC) inside a Graphics processing unit, Pascal is the ultimate reasoning why this TX2 is module is a supercomputer with unique performances delivered. Though we may not delve too much into neural network training the architectural system at hand delivers a boost up to 12 times in neural network training, however the seven times speedup of deep learning throughput is projected to save our team weeks of training the machine.

NVIDIA NvLink for Maximum Application Scalability

In the essence of ever using multiple GPU's for this project the technology of this architecture will delivers five times acceleration in bandwidth compared to that of the TX1 (NVIDIA, 2015). The NvLink high-speed bidirectional interconnect is responsible for this.

The key takeaway here is that this TX2 tiny module after considering its load from the list of specific tasks it has to accomplish/ it is incredibly power-efficient to that of the Raspberry Pi 3 and TX1 module, which makes this choice of hardware to be beyond ideal for the autonomous solar vehicle robot. While the TX2 module is one thing to install into intelligent devices, it also comes as a developer's kit. Within the dev kit we will find the TX2 board to be already pre-flashed or pre-installed with the Linux Integrated development environment. Another plus to the TX2 is that a Standard Development Kit of NVIDIA's Jetpack is too supported.

3.3.7.4 GPUs Comparison

The table below shows the feature comparison of the single board computers being considered for our project. The single board computer chosen for our project application the NVIDIA Jetson TX2.

Table 16. Single Board Computer and Supercomputer comparison Selection

	NVIDIA Jetson TX1	**NVIDIA Jetson TX2**	**Raspberry Pi 3 Model B**
CPU	ARM Quad- Core @ 1.73GHz	ARM Quad- Core @ 2GHz + NVIDIA dual-core @ 2GHz	4xARM Cortex @ 1.2GHz
GPU	256-core Maxwell @ .998GHz	256-core Pascal @ 1.3GHZ	Broadcom VideoCore IV@.4GHz
Memory	16GB	32GB	1GB
Camera (built in)?	YES	YES	NO
Ethernet	1000 BASE-T	1000 BASE-T	100 Base-t

USB	3.0 + 2.0	3.0 + 2.0	2.0
Encoder	4K30, (2x)1080p60	4Kp60,(3x)4Kp30,(8x)1080p30	1080p
Decoder	4Kp60, (4x) 1080p60	(2x) 4K60	1080p
Misc. I/O	UART, SPI, GPIOs	UART, SPI, GPIOs,I2C	UART, SPI
Thermals	-25 C to 80 C	-25 C to 80 C	0 C to 70 C
Voltage	5.5 to 19.6	5.5 to 19.6	3.3 t0o 5
Power	10W	7.5W	6W
Price (USD)	$199	$299	$35

3.3.8 Charge Controller

A solar charge controller is fundamentally a voltage or current controller to charge the battery and keep electric cells from overcharging. It directs the voltage and current hailing from the solar panels setting off to the electric cell. Generally, 12V boards/panels put out in the ballpark of 16 to 20V, so if there is no regulation the electric cells will damaged from overcharging. Generally, electric storage devices require around 14 to 14.5V to get completely charged. The solar charge controllers are available in all features, costs and sizes. The range of charge controllers are from 4.5A and up to 60 to 80A (Solar Charge Controller Types, Functionality and Applications, n.d.).

When it comes to solar panels, there are three different types of charge controllers.

- Simple 1 or 2 Controls: It has shunt transistors to control the voltage in one or two steps. This controller basically just shorts the solar panel when a certain voltage is arrived at. Their main genuine fuel for keeping such a notorious reputation is their unwavering quality – they have so not many segments, there is very little to break (Solar Charge Controller Types, Functionality and Applications, n.d.).

- PWM (Pulse Width Modulated): This is the traditional type charge controller, for instance anthrax, Blue Sky and so on. These are essentially the industry standard now (Solar Charge Controller Types, Functionality and Applications, n.d.).

- Maximum power point tracking (MPPT): The MPPT solar charge controller is the sparkling star of today's solar systems. These controllers truly identify the best working voltage and amperage of the solar panel exhibit and match that with the electric cell bank. The outcome is extra 10-30% more power out of your sun-oriented cluster versus a PWM controller. It is usually worth the speculation for any solar electric systems over 200 watts (Solar Charge Controller Types, Functionality and Applications, n.d.).

3.3.8.1 Pulse Width Modulation (PWM)

A PWM solar charge controller stands for "Pulse Width Modulation". These operate by making a connection directly from the solar array to the battery bank. During bulk charging, when there is a continuous connection from the array to the battery bank, the array output voltage is 'pulled down' to the battery voltage. As the battery charges, the voltage of the battery rises, so the voltage output of the solar panel rises as well, using more of the solar power as it charges. As a result, we need to make sure we match the nominal voltage of the solar panel array with the voltage of the battery bank (Store, 2016).

The PWM charge controller we discovered for our system is shown below. It is a 12V/24V, 20A Solar Charge Controller for LiFePO4 Batteries. We will be utilizing lithium ion batteries for the electronics system. This was the most cost effective, efficient, and safe option for a Lithium-Ion Solar Charge Controller.

Figure 20. Solar Charge Controller for Lithium Batteries (Power, 2018)

Bioenno Power 12V/24V, 20A Solar Charge Controller (Model SC-122420JUD) is a versatile controller for use in solar systems with an integrated LCD display, that is designed to charge LiFePO4 (Lithium Iron Phosphate) batteries (and AGM/SLA batteries)! This solar controller accepts either 12V/24V input from solar panels and can handle up to 20 Amps. Featuring our proprietary CC/CV (Constant Current/Constant Voltage) circuitry, the solar controller provides a regulated voltage output for charging 12V or 24V LiFePO4 (and AGM/SLA batteries). The solar controller also provides a regulated 12V or 24V output for electrical loads (depending on whether a 12V or 24V battery is used). Solar controllers are required for all solar systems in order to maintain a regulated output voltage to charge batteries and for maintaining a regulating output voltage for loads (Power, 2018).

3.3.8.2 Dual Battery PWM

Since we were aware we were going to need two charge controllers for the two sperate battery packs, needed for the power distribution of the solar vehicle, we researched the possibility of having one solar charge controller that could handle two batteries. For this we found Renogy 20 Amp PWM Dual Battery Charge Controller.

Figure 21. Dual Battery Solar Charge Controller

3.3.8.3 Maximum Power Point Tracking (MPPT)

An MPPT solar charge controller stands for "Maximum Power Point Tracking". It will measure the Vmp voltage of the panel, and down-converts the PV voltage to the battery voltage. Because power into the charge controller equals power out of the charge controller, when the voltage is dropped to match the battery bank, the current is raised, so you are using more of the available power from the panel. You can use a higher voltage solar array than battery, like the 60 cell nominal 20V grid-tie solar panels that are more readily available. With a 20V solar panel, you can charge a 12V battery bank, or two in series can charge up to a 24V battery bank, and three in series can charge up to a 48V battery bank. This opens up a whole wide range of solar panels that now can be used for your off-grid solar system (Store, 2016) (GreeSonic, 2018).

The MPPT charge controller we discovered for our system is shown below. We will be utilizing a lead-acid battery pack to power the motor system of the vehicle and we assume the batteries attached to the motor will drain faster then the battery pack installed from the electronics package. This is with it is critical to our design that we use a highly efficient MPPT solar charge controller.

Figure 22. MPPT Solar Charge Controller for Lead Acid Batteries (GreeSonic, 2018)

The GreeSonic MPPT1575 controller promises the tracking efficiency >99% and the charging efficiency is about 20% higher than the traditional PWM controller, which can make the cost of the whole system much lower. What's more, it's the most compact, waterproof, simple-operate. This charge controller also has features including and installed temperature sensor, automatic voltage recognition, and a strong durable aluminum case (GreeSonic, 2018).

Summary

For budget restrictions this seemed to be a great option for charge controller until after further researched we realized the major component differences between charge controllers for lead-acid batteries and lithium-ion batteries. The mixture of the two batteries on the system could be catastrophic. We also realized another issue due to the batteries being two different voltages. Due to these reasons the dual battery charge controller was not a viable option.

Ideally for the design of our vehicle we want to use a MPPT charge controller for both batteries. Besides how the three type of charge controller's function, there is also a drastic difference in price. The MPPT charge controller has better industry reviews and is better at maintaining a constant voltage to the battery cells for charging, but has a much higher price point.

Before choosing a certain type of charge controller, we first needed to review the type and voltage power of the battery power packs for both the motor and the electronics system. For the motor we chose two 12V Lead Acid Batteries and for the Jetson TX2 Development kit running the electronics system we chose a 3S Lithium Polymer Battery pack.

Since these batteries are different in both type and power pull we need two different charge controllers. This is due to the specific charging needs and constraints of each battery. It is dangerous to treat batteries of different types in the same way when it comes to power distribution.

3.4 Possible Architectures and Related Diagrams

3.4.1 System Block Diagram

The following block diagram shows a preliminary description of how the hardware components would interface with each other. The Solar Panel will be connected to a Battery Charge Controller. This controller would manage the amount of power delivered to the battery while monitoring the battery charge levels. This process will make sure the battery does not get overcharged which can cause the battery to explode and cause harm to the user and/or people around vehicle. The System will make use of two 12V batteries to store sufficient energy for the vehicle to run no more than 20 minutes. A PCB will be designed in order to control and distribute power to all of the electrical components used in the system.

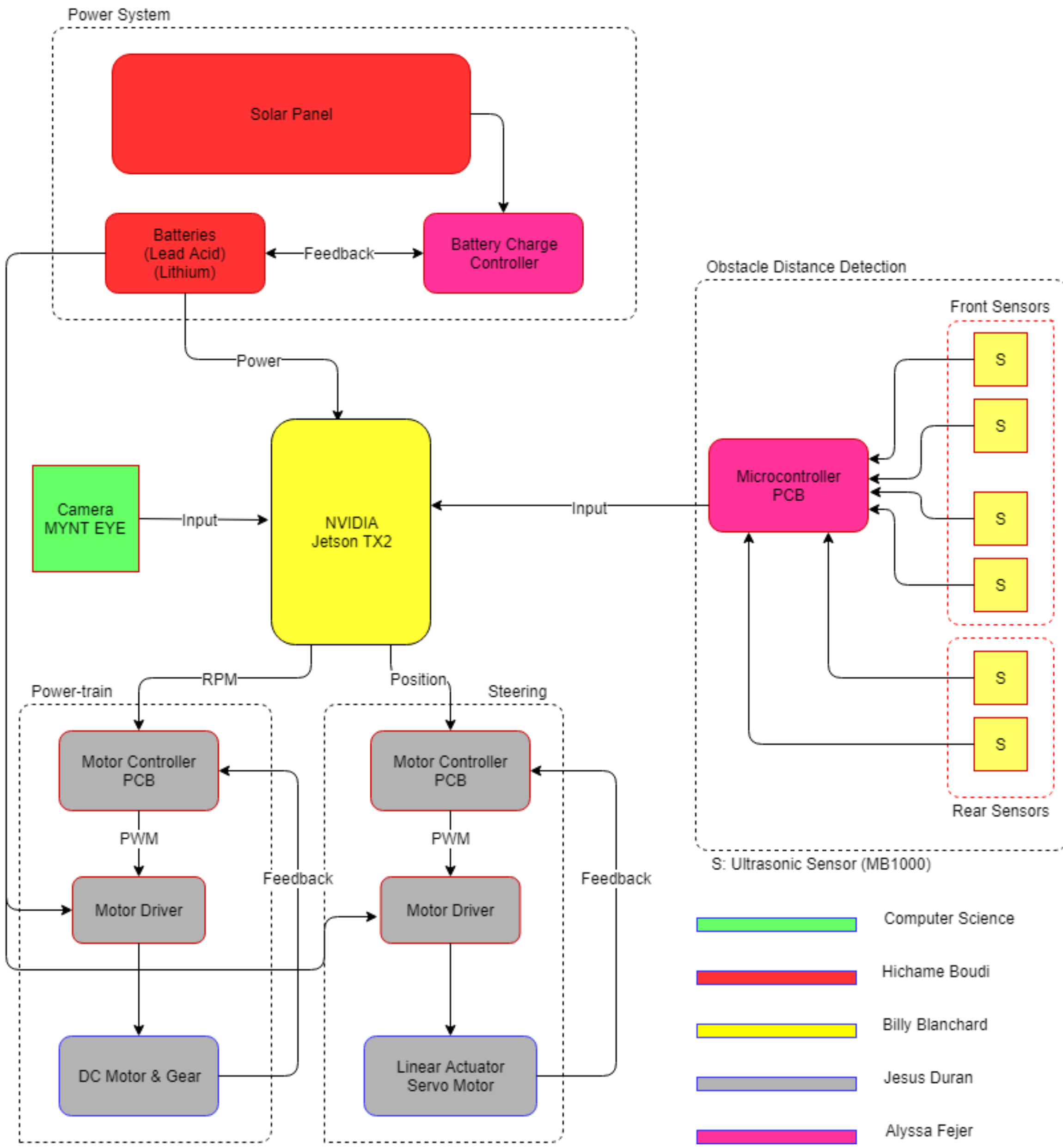


Figure 23. Solar Vehicle top level system

A Graphical Processing Unit (GPU) will be used to process images from an onboard camera. The camera will be used to send a visual representation of the surrounding environment to the GPU. Using Computer Vision libraries will aid in object recognition and avoidance by analyzing the captured images. Another component of the obstacle detection and avoidance subsystem consists of an array of Ultrasonic Sensors. The Ultrasonic Sensors data will be managed by a microcontroller. The microcontroller most important job will be to calculate the distance to the detected object using the data collected from the ultrasonic sensors. The data will be sent to the main computing device using communication protocols such as UART, SPI, and I2C. A DC motor and gear will make part of the powertrain subsystem.

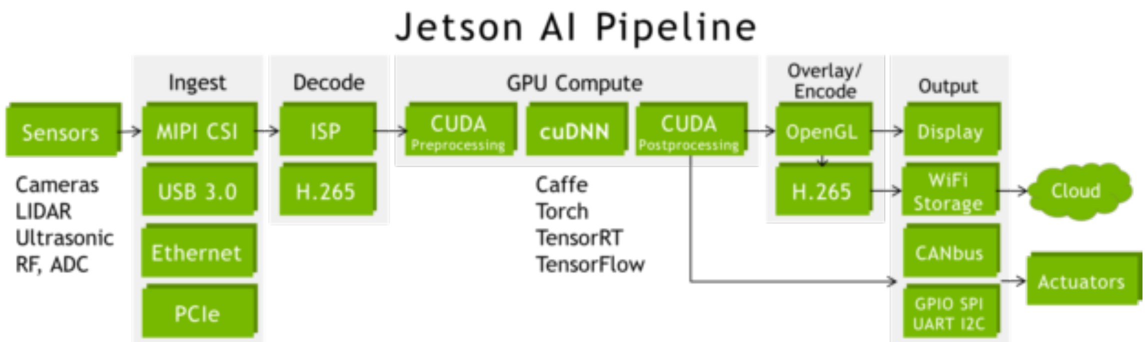


Figure 24. Jetson Sensor & Software suit to Linear Actuators

A motor controller will be used to monitor the speed and acceleration of the vehicle. As part of the sensor suite, an IMU will be used to relay information to the main computing device about the vehicle orientation. This data will be used for navigation purposes and monitoring the behavior of the vehicle in general. The selected Main Computing Device, will be in charge of processing camera and sensor data and determine the optimal path to navigate while avoiding obstacles. This will be achieved by interfacing with the motor controller and steering system. The steering system will be implemented using a linear actuator with position feedback.

Research associated with the components that make the system, was distributed among team members. The Mechanical Engineering teams will handle the solar panel selection, along with selecting a motor. The Computer Science team will handle researching algorithms related to computer vision and path planning. The Computer Engineering team will be in charge of the system's power distribution as well as hardware components needed to achieve the task of navigation and obstacle detection and avoidance.

3.5 Parts Selection Summary

3.5.1 Microcontroller

The current design for our solar vehicle, will only incorporate one Arduino Uno into the electronic system. The main purpose for the Arduino will be to handle communication between the ultrasonic sensors we are mounting to the outer frame of the vehicle for object detection and avoidance. We may also try to attach the RC Relay needed to control the linear actuator for steering if there are pins still available after we have attached all the sensors to the Arduino Uno.

The only challenge we foresee dealing with the microcontrollers is if we will need a pin extension to handle all the wiring for the ultrasonic sensors.

3.5.2 Camera

In the current design of our solar vehicle we account for the MYNT EYE stereo camera. As mentioned earlier in this report we are attempting two different ways to detect obstacles, avoid them, and stay on the path to complete the obstacle course.

The first way is to use computer vision with the ultrasonic sensors to recognize what is an obstacle in the path of the solar vehicle that it needs to avoid and the “walls” or boundaries that we will be using traffic cones for.

The second way is to write a SLAM (Simultaneous Location and Mapping) algorithm to detect our vehicle’s location amongst the obstacle course and map the obstacle course as vehicle completes the course. Ultimately optimizing the path from point-A to point-B, or from start to finish before the vehicle has to start its second lap.

To utilize SLAM the MYNT EYE stereo camera will be directly connected to the Jetson TX2 board for all processing, decision making, and power source. The camera will be mounted to the front of the vehicle so it will be able to the full span of the course in front of it. The camera will then use its triangular lens to calculate the distance between the vehicle and the world around it mapped out in key points or corner points.

3.5.3 Ultrasonic Sensor

Based on the research performed and analyzing the features of each sensor, the MB1000 Ultrasonic Sensor by MaxBotix will be selected for distance sensing. While not being the most economical sensor of the list, it provides the most flexibility out of all of them. It has the ability to work on a range of voltages (2.5V – 5.5V) while having the lowest operating current. The MB1000 also offers the best maximum and minimum detection range (0cm & 6.45m). One of the most desirable features the MB1000 possesses over the other two sensors, is the wide working detection angle (54.6^0). This can help eliminate blind spots around the vehicle and make the most use out of the sensor.

3.5.4 IMU Sensor

Analyzing the SEN0140, BNO055, and GY-521 features and the criteria by which the IMU sensor is to be selected, the Adafruit 9-DOF Absolute Orientation BNO055 IMU will be selected as the orientation sensor. While all three sensors offer similar features, the biggest deciding factor for choosing the BNO055 was the amount of documentation and support available for this sensor. Adafruit provides a plethora of material to help the user understand the inner workings of the sensor. Also, the BNO055 is backed by a large user-base with an active online presence.

3.5.5 DC Motor

This section gives a brief description of the motor selection. It is necessary to mention that researching and selecting a motor for the project was a task designed for the Mechanical Engineering part of the project. The ZXTDR, model M1020, electric motor, along with an appropriate gearing system, will be used as the powertrain of the vehicle. It is a high performance 24V, 500W, Brushed Electric Motor produced by Unite Motor and distributed by Amazon. It comes with an 11-tooth sprocket designed for a #25H chain. It has a rated speed of 2500 RPM at 27.4 Amps (2.2 Amps with no load) and a rated torque of 1.5 Nm. The peak efficiency of the motor is established at 24V (78 percent). The motor is capable of rotating in reverse by swapping the polarities. The motor can handle a maximum load of up to 200-lbs (90 kg) which makes it suitable for go karts, scooters, electric bikes, moped, and similar applications. The MY1020 comes at a price of $71.49 before taxes.

3.5.6 Battery

The decision on which battery the teams are to use on the motor sub system of the Solar Vehicle was made by the Faculty Adjunct Steven Flanders. The battery he chose was two ExpertPower 12V 7Amp Lead Acid Battery. We will connect these two batteries in series to get 24 V 7Ah to supply the voltage needed for the motor.

This battery can store 7 Ah at 12 V that equates to 84 Wh, which means it can supply a maximum of 12 V and 7 Amperes for about one hour, after 1 hour the battery will be completely drained. We are using a motor of 24V and 500 W so its current is almost 21 Amperes, so basically the battery will be completely drained in about 20 min or less. We have chosen this battery with lower capacity because one of the requirements is to use solar panel. We want to ensure the battery is drained quick enough so that the vehicle starts to pull power for the solar energy produced, instead of solely relying on the battery.

For the electronic sub system of our solar vehicle we have chosen to use a Tenergy Lithium-Polymer 14.8V 3000mAh Rechargeable Battery Pack. This battery is compact with a high heat resistance needed for the vehicle performing in full sunlight. This battery will provide power to the GPU, sensors, and steering system of the vehicle.

3.5.7 Charge Controller

We have decided to use two solar charge controllers. Each one will be specific to the power distribution system it is recharging. Since the batteries for the motor subsystem will be drained faster then the lithium-ion batteries we have decided to spend the extra money and use the GreeSonic MPPT1575. This will ensure the energy being sent to the lead-acid batteries is as efficient as possible so the batteries can be recharged once they are drained during vehicle performance on the obstacle course.

For the lithium-ion batteries suppling power to the electronics subsystem and steering we have chosen the Bioenno Power 12V/24V, 20A Solar Charge Controller specifically built for Lithium Polymer batteries. This option was most preferred due to its cost. PWM charge controllers are cheaper then MPPT due to the technology and efficiency. This charge controller may not be as efficient at the MPPT charge controller; however we assumed the lithium battery is most efficient with battery discharge and will not use as much power as the motor so efficiency with this charge controller was not necessary.

3.5.8 Voltage Regulator

We have decided to use switching regulator instead of linear regulator because linear regulator is inefficient and the difference between the input voltage and regulated output voltage is continually dissipated as heat. Switching regulator provides highly efficient, it dissipates almost no power, and it rapidly switch a series element on and off. It also generates output voltages that are higher than the input voltage or of opposite polarity.

We used Texas Instruments Webench tool to prototype a design with the proper DC-to-DC converter by inserting the minimum and maximum input voltage, the output voltage, and the output current. In this design, we are going to use two batteries of 12V in series to get 24V. We are also using the Jetson TX2 as a main component which needs between 3.2 to 19V with a maximum power of 12W. So basically, we need to convert 24V to 19V and keep in mind the maximum current which is 0.75 A. The Jetson will supply the voltage to sensors and ATmega328 and it already has an integrated DC to DC converter that will convert 19V to 5V.

We have decided to use TPS54308 synchronous Buck converter. It provides 4.5V to 28V input voltage range, and it includes two integrated switching FETs. It also achieves the high-power density and offers a small footprint on the PCB. It has an efficiency of 98.3%.

3.5.9 GPU

The design team made a conscious decision to go with NVIDIA’s Jetson TX2 embedded system. The Jetson TX2 is a powerful mainframe computer or what the industry may refer to as a “Supercomputer”. Not to be misleading, the TX2 is a low-powered embedded system operating well under 8 watts during general energy usage (NVIDIA, 2015). The Jetson TX2, which is designed by Nvidia, was released in March 2017[1]. What makes this embedded module a supercomputer is the cluster of hardware components embedded into its architecture. The TX2 contains Nvidia’s Pascal Graphics Processing Unit which is embedded with a 256-cores. The TX2 is also comprised with 8GB of Low Power Double Data Rate memory (LPDDR4) and a 16-core ARMv8 64-bit CPU. The LPDDR4 comes with a 64/128-bit interface (NVIDIA, 2015).

3.5.10 Solar Panels

The responsibility of selecting the solar panels is given to the Mechanical Engineering teams. This section will be updated when information about the selected solar panels becomes available.

Some information we need to ensure the mechanical engineering team is the restrictions of the solar panel voltage. A 12V solar panel can charge a 12V battery. A 24V solar panel or solar array (two 12V panels wired in series) is needed for a 24V battery bank, and 48V array is needed for 48V bank. If you try to charge a 12V battery with a 24V solar panel, you will be throwing over half of the panel's power away. If you try to charge a 24V battery bank with a 12V solar panel, you will be throwing away 100% of the panel's potential, and may actually drain the battery as well (Store, 2016).

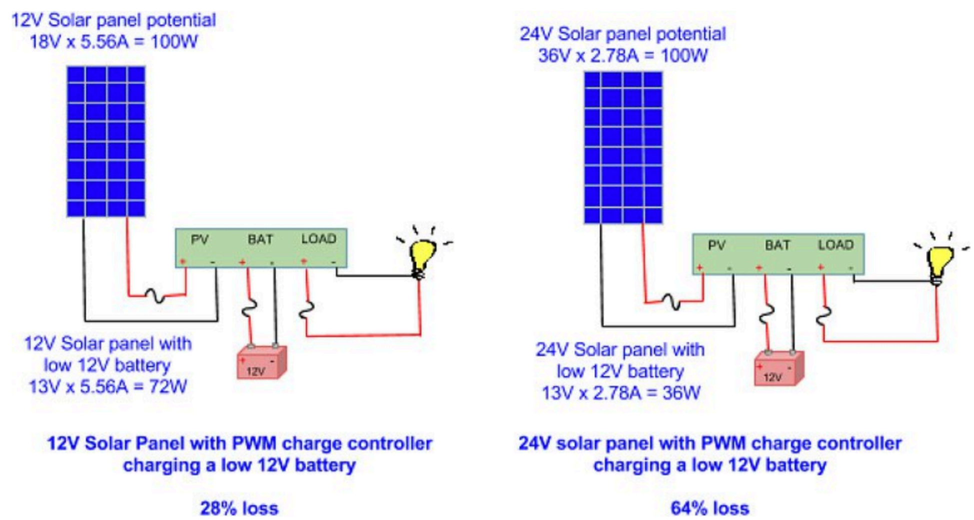


Figure 25. Solar Panel Efficiency based on Panel Voltage (Store, 2016)

The figure above shows the efficiency of a system with a PWM charge controller, however that detail is not important. The vital information is that the solar panels need to be either equal or more powerful than the power voltage being drawn from it.

3.5.10 Circuitry Materials

In our project, we are going to use many electronics components which are generally soldered onto the PCB. Our printed circuit board will support and electrically connect these components using conductive tracks. We will also use wires and electric tape to connect the circuit.

We will need DC to DC converters to change the DC voltage level. The goal of using converters is to match the power requirement and for the safety of the components. DC to DC converters will supply power to sensors, microcontrollers, and other components. In overall, the circuity materials will be purchased when needed.

3.5.11 Steering (RC Relays)

The complete design for the Solar Vehicle chassis is the responsibility of each of the Mechanical Engineering teams. We have talked to them about making informed decisions to assist both us and them in design, however when it comes to steering the vehicle each team has the flexibility to choose their own method. For the purposes of testing out prototype we are creating a steering system using an RC Relay and linear actuator.

4.0 Related Standards & Realistic Design Constraints

4.1 Relevant Standards

4.1.1 Coding Standards

One of the most essential parts of any engineering design involving autonomy or robotics are standards. The Robotic Operating System (ROS), which is the platform where we plan to natively construct and build our code and algorithms, is a giant system where individuals write large amounts of code. For adequate development in the ROS environment there are some guidelines or standards me must follow.

Source control is supported and strongly encouraged when working with ROS. Version source controls like Subversion, Bazaar, and Git are supported in the Ros community. When using a repository, github.com is used to host the main ROS code base. Machine-generated files, such as object (.o), auto-generated configure scripts or libraries(.a,.dll) should not be added to the source control. Only a minimal amount of code or build files should be added. *Svn add* adds a directory's entire content in a recursive manner (ROS, n.d.). *Make clean* should be performed before *svn add* (ROS, n.d.). Like saving your code often, you should commit your code often. Leave a descriptive message with each commit to the repository.

When bug tracking, a ticket should be opened or created whenever a bug is found. Task assignments are treated like bug trackers for every ROS package. Milestones will be assigned to each issued ticket or bug. Milestones are general releases or software updates. In summation, open a ticket if a bug is found. Tickets like emails, should have descriptive headings and subject lines. Provide the exact steps to reproduce that bug.

Built code in ROS gets placed into packages that can be collected into a solitaire repository. This is essentially the code layout.

In the root directory of a package there must be a manifest.xml file. A description, the author of the file and license should be included in the manifest.xml.

Cmake is the build tool. A CMakeLists.txt file must be included in every build package in the top directory (ROS, n.d.). Build files only go into packages with build steps. These files must also be included in a Makefile.

Library and Message are the two levels of testing. Standard unit-test frameworks are used at the library level. We use unit test in Python and gtest when working with C++.rostest is used at the message level. A system of ROS nodes is set up, then tested, then torn back down. Automatic testing is used in ROS at both levels

The Quality Assurance Process governs how all code should be documented. Any visible APIs at either the ROS-level or code-level should be documented (ROS, n.d.). A public web hosting site should be used for large data files or unit tests.

When writing code in Python, Python Enhancement Proposal 8 or PEP 8, is the coding Style Guide and standards which describes the specific coding conventions. PEP 8 can act more as a "common sense" standard approach rather than an actual strict style guide. PEP 8 prioritizes readability over consistency (Python, n.d.). The PEP 8 convention contains a plethora of guidelines and rules involving whitespace, indentation, tabs, maximum line length, comments, formatting, trailing comma usage, etc. These coding conventions are all mainly listen under Code layout.

Indentation under the PEP 8 style guide it is mentioned that 4 spaces should be used to infer an indentation level (Python, n.d.). In the context of Python, PEP 8 considers that no arguments should be on the same line as the functions name or first line. Also, further or noticeable indentation should be used such that, it is undoubtfully viewed as a continuation line. This is referred to as a hanging indentation (Python, n.d.). It is conventional to align continuation lines either vertically inside brackets, braces and parentheses or use a hanging indentation (Python, n.d.). It is highly recommended to use Python 2 command line interpreter *-tt* option (Python, n.d.). Invoking this command over the *-t* option turns what would be issued as warnings about code that convolutes spaces and tabs illegally to errors.

The preferred method of indentation is said to be spaces over tabs. If the code is initially, then tabs should be only used to maintain consistency. Python 2 allows the code to mixed with both tabs in spaces for indentation whereas Python 3 does not allow mixing of the two at all (Python, n.d.). Any given line should be limited to 79 characters (inclusively). In the case of long flowing blocks of text with comments or docstrings, the maximum limit of characters is 72 (Python, n.d.).

It should be discussed and agreed upon within the team before increasing the nominal line length up to 100 characters. In this case, docstrings and comments must still be wrapped at 72. This is rule is governed by the conservative Python standard library (Python, n.d.). Instead of using backslash for line continuation on long lines, the preferred method is to use Pythons implied line continuation within braces, brackets and parentheses. Backslashes are most beneficial or appropriate in the events of using multiple, long with-statements. Implicit continuation does not work when using with-statements (Python, n.d.). The same case applies with assert statements.

For years the conventional style was to break after binary operators, however this caused readability issues. The binary operators would get displaced across the screen and the operator would be apart from its operand. If the convention is done consistent locally, it is then allowed to break before or after a binary operator. This readability issue was corrected by mathematicians by simply swapping the conventional method. This resulted in more readable code (Python, n.d.). Both class definitions and top-level functions should be surrounded with double blank lines. Single blank lines should surround method definitions within a class. To indicate logical sections, single blank lines should be used scarcely. PEP 8 states that only one import statement should be allowed on a single line. Imports conventionally appear at the top of the files. Standard library related third party then the local application/library specific imports is the conventional order for import statements with a blank line between each import (Python, n.d.).

PEP 8 states that absolute imports should always be used. Readers become confused which names are available in the namespace when using the wildcard imports. Unlike most programming languages, both double-quote and single-quote strings literals are equivalent (Python, n.d.). There is no preference for one way over the other from PEP 8 (Python, n.d.). In the situation with parentheses and parameters, we should avoid extra whitespace immediately inside the brackets. Modern code requires modern comments. The first word should be capitalized in a comment unless it's a variable (Python, n.d.).

A cluster of complete sentences that all end with a period are called block comments. A single '#' on a line is used to separate block comments (Python, n.d.). Comments written on the same line as a statement is called an inline comment. Inline comments should be used in moderation. Avoid stating the obvious with inline comments. For example, b++ # Increment b. Capitalize all letters used in the camel-case or cap words that are acronyms. The letters 'l' lowercases el, 'I' capital eye, and 'O" should all be avoided as single character identifiers. The Cap Words convention is used for class names (Python, n.d.). Instance methods she use *self* for the initial argument.

For class methods *cls* is used as the first argument. Adding a single underscore usually corrects a function's name collision with a keyword. Function names should be lowercases followed by an optional underscore. Double leading underscores helps avoid Python's name mangling rules. The standards list continues further but the content discussed are the essential guidelines from PEP 8. The ultimate goal when employing a coding standard is to promote code readability amongst a team of programmers.

4.1.2 Battery Standards

IEEE 1361™-2014 which explain, "lead-acid battery test procedure, lead-acid battery tutorial, photovoltaic (PV) battery characteristics, PV battery selection, PV battery test procedure, PV systems" (Learning the Basics about Batteries, 2018). Using this standard, knowing the different type of lead acid battery and which to choose for best performance, safety, and efficiency. This standard mentioned that there are two type of lead acid battery: Vented lead-acid batteries and VRLA batteries. Stand-alone PV referred to a solar power device such as a solar panel. The standard explains that using a battery in this system is a must for extending operation of the system load during hours of darkness and below average solar resource.

This standard provides information on PV battery selection which means that we have to understand different battery types. Lead acid batteries are made for specific applications. For example, automotive batteries stay at full charge most of the time, UPS batteries does not tolerate charge and discharge cycle, and deep cycle batteries are designed to deliver electricity for a long time. Also, the material on which the battery was made plays a role because the current provided by the battery comes from a chemical reaction converted to energy (IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead Acid Batteries Used in Stand Alone Photovoltaic (PV) System, 2014).

It also provides information on PV battery characteristics. These characteristics are specific KWh cost which refers to operation costs of the battery divided by the stored kWh during its whole life, life time, efficiency, self-discharge, maintenance cost, and power (IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead Acid Batteries Used in Stand Alone Photovoltaic (PV) System, 2014).
It contains information on PV battery testing and evaluation. It analyzes the cycling test procedures for batteries used in standalone photovoltaic power systems. The objective of testing procedure is to evaluate the life time of the battery and its capacity.

4.1.3 Power Supply Standard

The International Electrotechnical Commission (IEC) and the associated International Organization for Standardization (ISO) are the principle agencies responsible for electrical safety standards. IEC 60950-1 is a standard applicable to all electric machines s with a rated voltage not exceeding 600 V. it is attended to protect electrical components from fire, dangerous temperature and mechanic stability. It also prevents from electric shock and injury to persons. This standard classifies equipment on how their power supply, and isolate secondary circuit for high ac voltage. There three classifications: class 1 for equipment that achieve their protection by using basic insulation and protective earth grounding, class 2 for equipment that achieve protection by using double or reinforced insulation, and class 3 for equipment that provide protection against electric shock with no hazardous voltage generated within the equipment.

4.1.4 IPC PCB (Printed Circuit Board) Standard

IPC, the Association Connecting Electronics Industries is an association which produces PCB-related standards. IPC standards are the electronics-industry-adopted standards for design, PCB manufacturing, and electronic assembly. These standards include requirements for organic PCB's and multichip modules, material considerations & board configurations and PC card form factors. IPC-2221A provides information on the generic requirements for organic printed board design. IPC-2222 provide information on the detailed requirements for organic rigid printed board design.

IPC-2223A establishes the specific requirements for the design of flexible printed circuit applications and its forms of component mounting and interconnecting structures. It also provides the types of a PCB including Single-sided flexible printed wiring containing one conductive layer, Double-sided flexible printed wiring containing two conductive layers, Multilayer flexible printed wiring containing three or more conductive layers, Multilayer rigid and flexible material combinations containing three or more conductive layers, Flexible or rigid-flex printed wiring containing two or more conductive layers.

These IP standards are very important and should be followed because they help the designer to build only the highest quality. They also help the designer, manufacturer, assembler, and tester to speak the same language.

4.1.5 Soldering Standard

The IPC J-STD-001 Requirements for Soldered Electrical and Electronic Assemblies has emerged as the preeminent authority for electronics assembly manufacturing. This standard provides information about materials, methods and verification criteria for producing high quality soldered interconnections. It also provides the process control methodology to ensure consistent quality levels during the manufacture of products.

4.2 Realistic Design Constraints

4.2.1 Economic and Time Constraints

The Solar Vehicle Competition was created with the purpose of challenging emerging engineers with relevant technologies that are, or will be, a common place like autonomous navigation and renewable energy. Part of the challenge includes setting economic constraints, aiming to reduce the cost of production which would in turn make the product a viable option for future consumers. The allocated budget, provided by Duke Energy Corporation, to develop the system is a total of $2000.00 per team. Each of the three teams involved will provide $500.00 for the development of the navigation and obstacle detection subsystem. This entails having to carefully select hardware components in a way that achieves the desired functionality without sacrificing the integrity and safety of the vehicle.

For major components such as the PCB, and charge controller, quality and reliability are a priority in order to not compromise the integrity of other hardware components used in the system. The obstacle detection and avoidance part of the system offers more flexibility for hardware components. The system will use low-cost (< $30.00) distance sensors. In the general sense, the price of the sensor is driven by the type of application, so there isn't necessarily a linear correlation between cost and quality of the sensor. The Graphical Processing Unit (GPU) will greatly impact the budget and will need to be selected in a way that meets only the minimum requirements necessary to detect obstacles in real time. The budget would also limit the camera selection to provide only the necessary resolution and video framerate. The project will be limited to using open source software as to not incur any additional expenses that could be associated with closed source software. An extensive amount of research would have to be performed before ordering hardware components since the limited budget does not yield an ample amount of room for discarding and reselecting components.

Time is one of the most important resources to manage for this project. The project's time constraint is set to two semesters (< 8 months) for full completion, with a final delivery date of the completed (fully operational) system in the last week of April, 2019. The first phase of the project involves the research of all the hardware components that will be used to meet the requirements of the project. This includes designing the system, ordering and testing the hardware components. Ordering and testing every hardware component before the December 3, 2018 can be a challenging task. Effectively managing the shipping times and availability of certain components, will play a major role in ensuring the requirements are met. Contingency plans will be put in place to compensate for any unforeseen delay.

The implementation phase (2^{nd} phase) of the project will take off in the second week of January and last until the final delivery date. During the first 2 weeks, the PCB design will be sent out to get manufactured. It will be imperative to have a fully tested PCB design since the manufacturing time, from sent to received, can take multiple weeks. A PCB redesign is possible, but it will put a strain on the overall timeline of the project. The system should reach its final development stages no later than March 1^{st}. This will allow sufficient time to integrate the system with the different Mechanical Engineering vehicles and to adjust components or connections as necessary. The structure describing every major task to be accomplished for this project and their respective due dates, is discussed in the Milestones section of this document.

4.2.2 Environmental, Social, and Political constraints

The autonomous solar vehicle is required to operate and traverse a path on Daytona beach, thus this presents an immediate identifiable environmental constraint. If unregistered or unmanned vehicles are not permissible on the beach this will change the dynamics of the project.

Another environmental constraint is that the robot is required to run off solar power. Both testing and possible showcase demoing can be blocked if the Florida does not permit. The solar panels that will be using must be powered with direct sunlight. Random weather patterns, storms, rain, or even cloudy weather can stump our power consumptions and possible charge banks. The majority of our power will need to come from the selected solar panels as we plan to do iterative testing under ideal test conditions, which will include maximum sunlight and minimal cloud cover to achieve, potentially the best test results.

The goal of this project is to design a fully autonomous solar vehicle which means this project will increase the safety of drivers. To accomplish this, the system should be used by the community, and making it as affordable as possible. Our system also must be comfortable to use.

Another social constraint concerns groups and individual within this project. Our team agreed that communication is the key of success. Any member of the team has any special situation, he must let us know.

The third social constraint is considered with the students in mind. The less time we can spend assembling, testing, and learning the concept, the less discouraged we are apt to be, resulting in a positive atmosphere and in a great learning experience.

Concerning political constraints, there are no political concerns because this design is intended to be an educational tool.

4.2.3 Ethical, Health, and Safety Constraints

This is an interdisciplinary project that, as it transitions through the design and development stages, will require that all parties involved conduct themselves in a professional manner while adhering to the highest principles of ethical conduct [8]. As engineers, we are expected to prioritize the health, safety, and welfare of the public. Engineers should operate only within their respective areas of study. Not having the correct qualifications for operating on a specific technical field can compromise the welfare of the public. Performing acts that purposely malicious and deceptive, violates the ethical standards by which engineers are held. For the scope of this project, research shall be presented in an objective manner. Components will be selected to comply with the necessary health and safety requirements.

Certain standards must be met in order to ensure the health and safety of the vehicle's user and people around it. The vehicle will use a lead-acid battery which proposes various hazards. Lead is a very toxic metal that, over-exposure to it, can cause various health problems such as anemia, weakness, and brain damage [9]. Appropriate warning labels will be put in place to educate the user on the potential health risks of lead exposure. Lead-acid batteries are also known to produce flammable hydrogen gas.

This will be addressed by keeping the battery in a well-ventilated area of the vehicle. Shorting the battery connectors can be very dangerous due to the high amperage running through the short. It can result in the battery exploding causing serious burns; therefore, the battery must be kept isolated from the user to avoid any contact that could result in a short-circuit. Fire hazard is something to be considered in electrical systems. A malfunctioning component or a short-circuit can cause a fire which can constitute a health risk to the user. Appropriate wire gauges, insulation of connectors, and shock mounts for circuit boards need to be in place to minimize the risk of a fire. Also, to avoid exposure to materials that could constitute health risks, components that are compliant with the Restriction of Hazardous Substances (RoHS) will be used for this project.

While the vehicle will not be operating on public roads, the safety of people in or around the vehicle is crucial and it poses constraints that cannot be ignored. The object detection system is an integral part of maintaining the safety of those around the vehicle. Having more than one way of detecting objects ensures that the vehicle will still detect an object in the off-chance one of the obstacle detection subsystems should fail. The object detection system will be composed of two subsystems, a camera using computer vision and multiple ultrasonic sensors, to ensure reliability.

The camera needs to have the appropriate range for a wide view of the surroundings while the ultrasonic sensors need to be strategically placed to minimize blind spots. Algorithms implemented in the system need to be fully tested in a controlled environment beforehand. The PCB needs to be designed in a way to safeguard the integrity of the hardware components. As a last line of defense, the vehicle will equip an emergency shut-off button in case there is a major malfunction in the system that could potentially put at risk the safety of the public.

4.2.4 Manufacturability and Sustainability Constraints

As far as manufacturability of our system, this is a big constraint. The mina purpose of this senior design project is proof of concept. After many of hours of research we have discovered that the problem we are attempting to solve has either never been done before within our exact description or something similar has been done, but with a much larger budget than ours. If we wanted to build a system that was easily manufactured then we would have to take assembly into account, along with who our consumer would be.

As for sustainability, our solar vehicle is intended to be highly sustainable. The ability to completely removing it away from the dependency or electric power or fossil fuels is a huge accomplishment. Not to mention the vehicle being autonomous. Technology is advancing every day in AI (Artificial Intelligence) to make things more autonomous. If this project is a success then we will be taking one step forward into the future advancements of technology.

5.0 Project Hardware and Software Design Details

This chapter will be discussing the different subsystems that compose the Solar Vehicle system. Every major component selected throughout this text, was selected to achieve the functionality of each of the subsystems. These include microcontrollers, GPU, PCB, battery, sensors, camera, motor drive controller, and charge controller.

The main system can be broken down into four major subsystems such as power, navigation, obstacle detection, and powertrain. The power subsystem will be in charge of managing and distributing the energy produced by the solar panel to the various hardware components in the system. This subsystem has a high degree of importance since the safe operation of the vehicle depends on how the power is distributed across the entire system. The power subsystem includes the solar panels, charge controller, PCB, and batteries.

The navigation subsystem will ensure that the vehicle can navigate and stay within the boundaries of the course. The navigation subsystem will include a camera for a visual representation of the surrounding environment and various algorithms to set optimal paths to navigate. An IMU sensor will also aid in navigation by relaying relevant information to the system about the orientation of the vehicle. The obstacle detection subsystem shares a component with the navigation subsystem and that is the camera, which will be used to detect objects. As a secondary aid to obstacle detection are the ultrasonic sensors, which can detect and calculate distances to obstacles around the vehicle.

The powertrain subsystem is composed of a DC motor and gearing system. In order to manage the amount of power delivered to the motor, a controller will be set in place along with a feedback system using a hall effect sensor. All the subsystems will be managed by a computing device which will serve as the brain of the entire system.

The following sections will discuss how the components will be integrated, the schematics related to the design, and breadboard test of the design.

5.1 Initial Design Architectures and Related Diagrams

After researching all the components individually, we got together to discuss our findings and to decide the best design to solve the challenge of building a solar powered autonomous vehicle. Then we had the opportunity to consult with a professional Electrical engineer to learn about industry best practices and to ensure we were not missing any crucial components. The figure below shows the estimated layout of how and where our hardware/ software suite will be placed on each vehicle.

As mentioned earlier in this report, the steering can be unique to each group. Therefore, a linear actuator was chosen as a tester for our vehicle prototype.

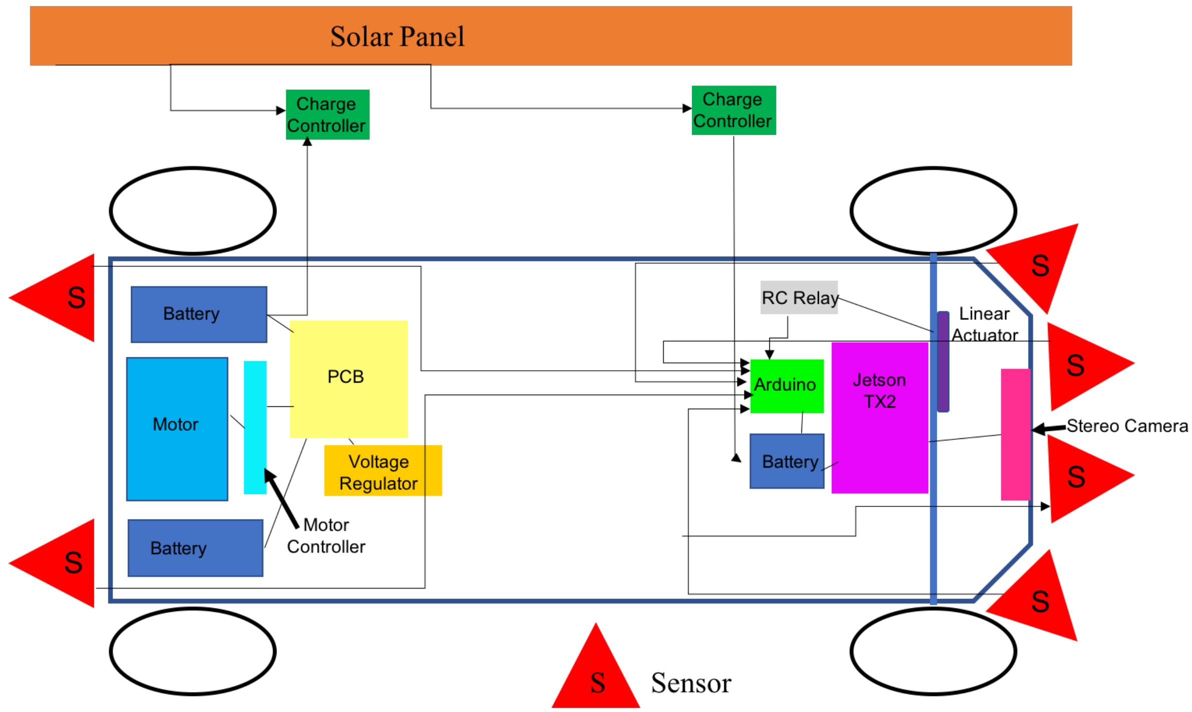


Figure 26. Hardware/Software Suite Design for Solar Vehicle

One of our most important takeaways from our consultation with a professional electrical engineering was how power the motor and lead-acid batteries will be. After careful consideration we opted to separate the power supplies to the two major components in the system. With our current design we will have lead acid batteries in the rear of the vehicle close to the motor for power efficiencies and another lithium-ion battery near the front to power all the smaller electronics.

5.2 Obstacle Detection

The obstacle detection subsystem operates using three major components: Arduino Uno Microcontroller, (Insert Make and Model here) Camera, and six MaxBotix MB1000 Ultrasonic Sensors.

The ultrasonic sensors will be placed in a way that covers the most amount of area in the path of the vehicle. The ultrasonic sensors will be connected to the Arduino Uno Microcontroller to process the ultrasonic sensor's data and calculate the distance to the objects. The calculated data will be sent to the Jetson TX2 board, via Serial Peripheral Interface, to be used by the obstacle avoidance algorithm.

The following diagram shows a top-level view of the obstacle detection subsystem.

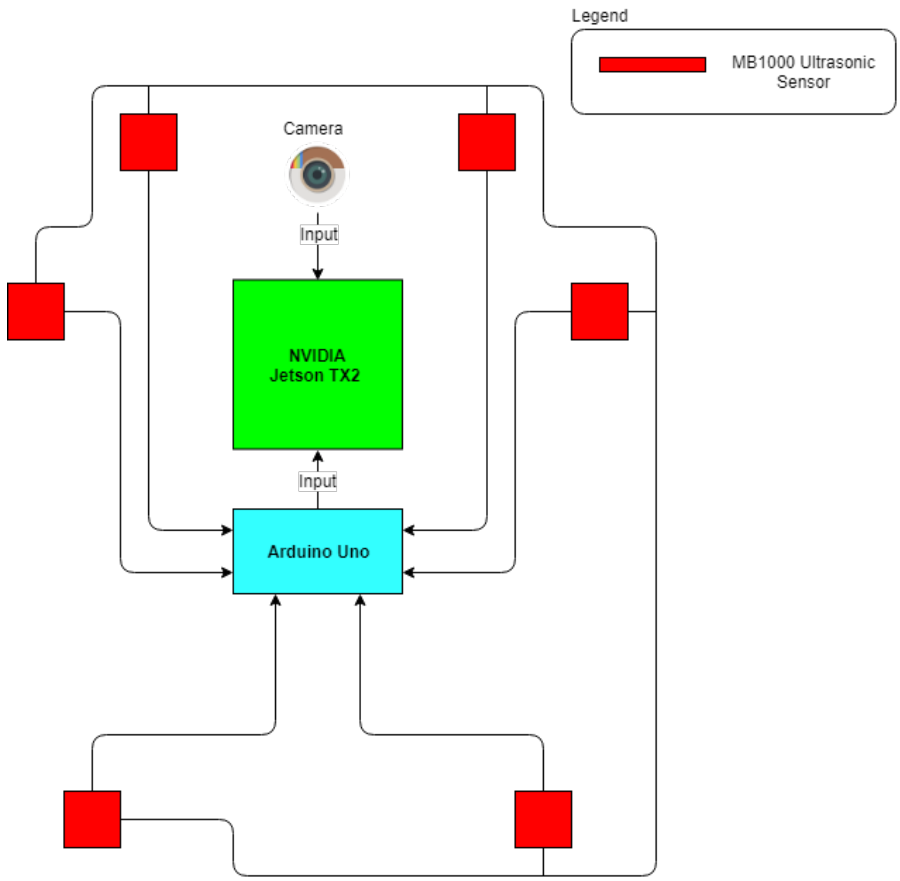


Figure 27. Obstacle detection top-level view

5.2.1 Ultrasonic Sensors

The MaxBotix LV-MaxSonar-EZ0 MB1000 Ultrasonic Sensors will be placed around the vehicle in a way that reduces blind spots. The position of the sensors is calculated taking into account the minimum working angle of the sensor which is approximately 43^0. The following two diagrams will give an approximate representation of the MaxBotix MB1000 beam projection and the exact location the sensors will need to be place to achieve an adequate obstacle detection performance.

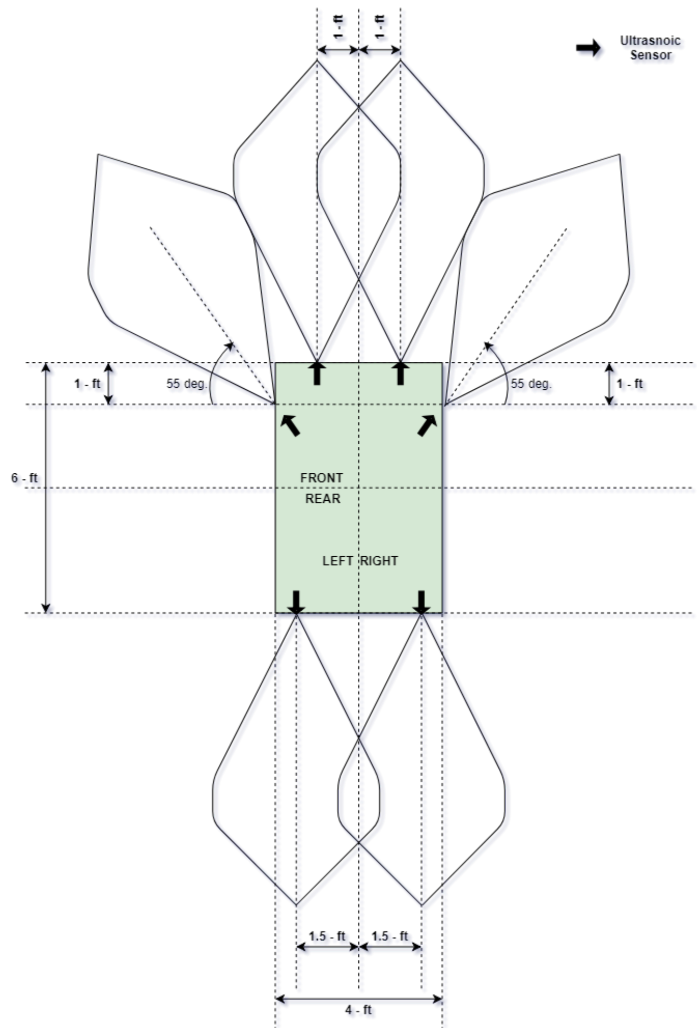


Figure 28. Top view of ultrasonic sensor location

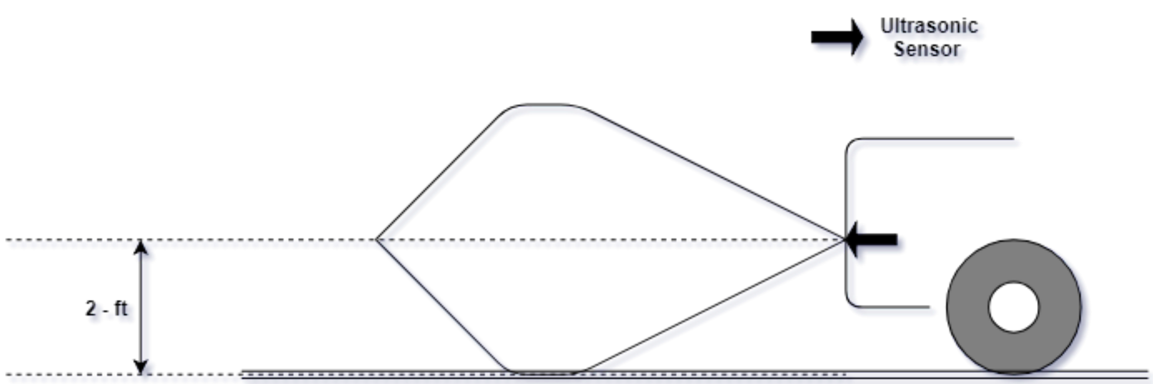


Figure 29. Side view of Ultrasonic Sensor elevation

Having multiple ultrasonic sensors in the same system can usually cause issues related to interference (cross-talk). MaxBotix provides multiple solutions to this issue which the company refers to as chaining. Their preferred method includes sequentially reading each sensor. The following schematic show the design implementation of this system using an Arduino Uno and three MaxBotix ultrasonic sensors.

5.2.1.1 Breadboard test

Each of the ultrasonic sensors were tested in the Senior Design Lab to ensure the functionality and accurate reading from each of the sensors. The 5V and GND pins of the sensor were connected to their respective 5V and GND pins of the Arduino Uno. The BW pin of the sensor was used as a trigger to start ranging. The PW pin of the sensor was then used to output to the Arduino a representation of the range. Using the time that it took for the signal to leave and comeback to the sensor, the distance to the object was calculated (in cm) by multiplying the time variable by 0.034 (speed of sound = 345 m/s) and dividing it by 2. The figure below shows the configuration used for the testing. The Jetson TX2 was connected to the Arduino Uno board using Serial Peripheral Interface (SPI), and the ultrasonic sensor was connected to the Arduino Uno.

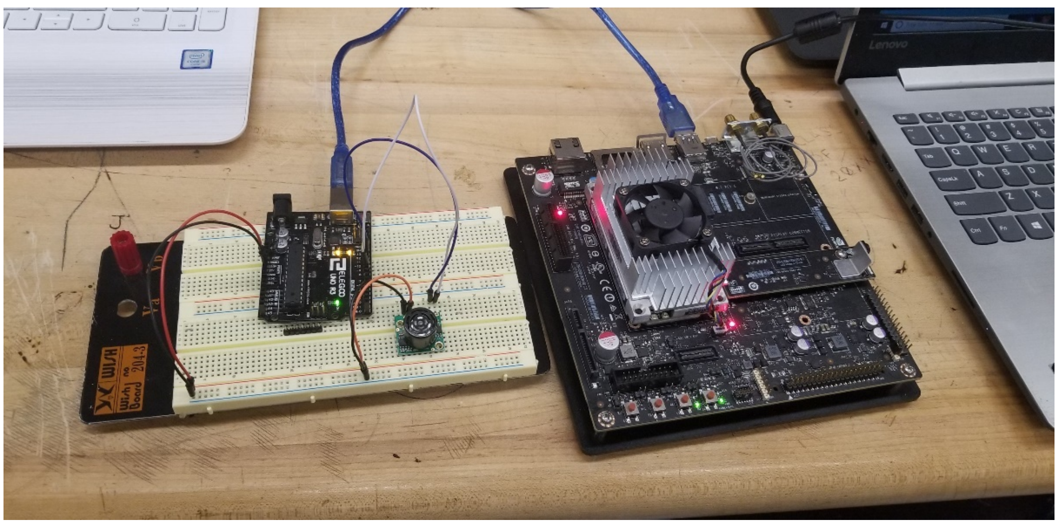


Figure 30. Ultrasonic Sensor Breadboard Test

The breadboard test showed that all the sensors had the same accurate reading. This was achieved by placing an object in front of the sensors and varying the distance. The effective working angle was also checked by moving the object to different positions within the same distance. All outputs were shown using the Arduino IDE's serial monitor.

5.3 Navigation

The navigation subsystem is composed of the hardware components that provide useful information to the navigation algorithm and allow automated steering of the vehicle. These components are the Inertial Measurement Unit (IMU) connected to the NVIDIA Jetson TX2, and a linear actuator connected to the steering system of the prototype controlled by the Jetson. This section will cover the details of how these components will be integrated into the system.

5.3.1 Inertial Measurement Unit (IMU)

The Adafruit BNO055 IMU will be connected to the NVIDIA Jetson TX2 to provide the navigation algorithm with meaningful information about the direction and acceleration of the vehicle. The low power requirements of the BNO055 will be provided by the Jetson. The communication between the IMU and the Jetson will be implemented using I2C interface. The figure below shows the schematic of the BNO055 IMU.

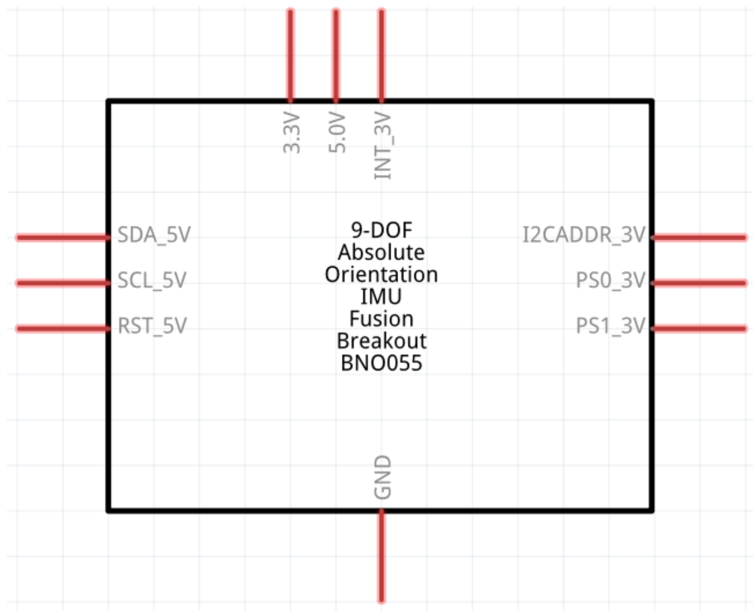


Figure 31. Adafruit BNO055 Schematic

The 5.0V pin of the IMU will connect to the 5.0 VDC Power pin of the Jetson (Pin 2). In a similar fashion, the ground pin of the IMU will connect to a ground pin on the Jetson (Pin 6). The I2C pins of the IMU are SDA_5V (data pin) and SCL_5V (clock pin). These to pins will connect to the Jetson's SDA1 and SCL1 respectively. These two I2C pins on the Jetson TX2 will provide 3.3V logic which matches the IMU's 3V to 5V logic capabilities.

The remaining pins on the IMU have various functions that could be utilized if the scope of the project needs adjustments. The RST_5V pin is used to reset the sensor. The INT_3V is used to generate an interrupt if a certain event occurs (i.e. movement is detected). The I2CADDR_3V pin is used to change default I2C address if more than one IC is present on the same I2C bus. The PS0_3V and PS1_3V pins are used to change the mode of the device to HID-I2C or UART.

5.3.2 Steering (Linear Actuator)

The system will have a 12” linear actuator connected to the steering tie rod for steering as represented in the figure above. The linear actuator’s 12V 3A power requirements will be achieved using a TPS54340 Step-Down DC/DC converter. The figure shown below shows how the linear actuator will adjust the direction of the wheels of the vehicle to steer.

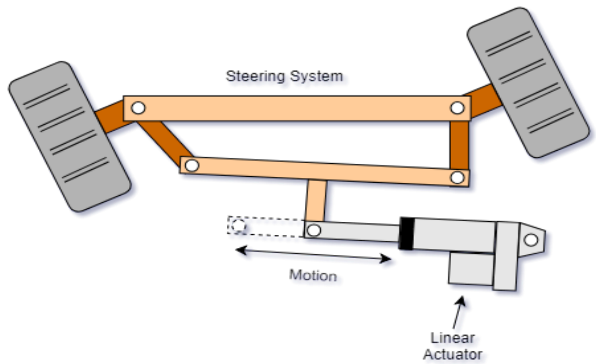


Figure 32. Top View of Steering System with Linear Actuator

The Step-Down converter has an input of 15VDC to 30VDC and output of 12V at 3A with a 94.5% efficiency. The Step-Down converter will be connected to an RC Relay that will be controlled by the NVIDIA Jetson TX2.

While a linear actuator has the advantage of handling high amounts of force, it has a major disadvantage which is keeping track of the shaft position. Keeping track of the position of the shaft must be present in order to manage the angle of the wheels with respect to the vehicle. The mid-point of the linear actuator (6” extension) should keep the wheels straight. Extending/contracting the shaft from the 6” point, turns the wheels left or right depending on the location of the liner actuator within the vehicle.

One of the options is a linear actuator with a built-in potentiometer that can be used to keep track of the shaft position. Another potential option is the use of a Time of Flight (ToF) infrared distance sensor to measure the extension of the linear actuator’s shaft. The latter approach would cut down on the budget at the cost of implementing a separate subsystem to keep track of the shaft position.

5.4 Power System

The power system to control the vehicle consist of a solar panel between 300W and 470W to charge two 12V 7Ah Lead-acid batteries and a 14.8V Lithium Polymer battery via charge controller. The two 12V Lead-acid batteries will be connected in series to achieve the 24V required by the brushed DC motor. The Lithium Ion battery will be connected to the NVIDIA Jetson TX2 and other sensitive hardware components used in the system.

The motor chosen for the vehicle is a 24V, 500W, Brushed Electric Motor. One major drawback to working with motors is the large amounts of electrical noise they produce on the power line of the system. This noise can interfere with our sensors and can even impair our electronics by causing voltage dips on our regulated power line. Large enough voltage dips can corrupt the data in electronic registers.

The main source of motor noise is the commutator brushes, which can bounce as the motor shaft rotates. This bouncing, when coupled with the inductance of the motor coils and motor leads, can lead to a lot of noise on your power line and can even induce noise in nearby lines.

Due to this problem we have decided to use two separate power sources for our vehicle design. One system to power the motor driver and motor of the vehicle and the other to power the electronics used for obstacle avoidance and navigation.

5.4.1 Charge Controller

The major efficiency of our system is the amount of energy created from the solar panel connected to the vehicle. To harness this energy and convert it into usable amperage to charge the two 12-volt batteries connected in series and the Lithium Ion Battery, a charge controller is used. Charge controllers are used to regulate the input voltage and/or current coming from a power source. They regulate the input power to charge the batteries effectively and does so safely by preventing the batteries from being overcharged. Without a charge controller, there is nothing keeping the batteries from being overcharged and expanding, heating up, or exploding. Charge controllers also prevent batteries from being undercharged, which could induce various degradation mechanisms depending on the battery (undercharging poses no issues for some batteries): reduction in capacity, copper dissolution, sulfation, early battery failure.

Since each Mechanical Engineering group is providing their own solar panels, the charge controller used for our system will need to be able to handle the highest wattage solar panel. As of right now the highest wattage Solar Panel being used is 350W. Taking this wattage we determined at least two 15-amp MPPT charge controllers are needed to charge the two 12V Lead Acid Battery pack for the motor and the lithium ion battery for the electronics.

As shown below in the figure, we will use two charge controllers for the two different battery banks in parallel to power our system. We need two different types of charge controllers for this system. One for the 24V Lead Acid battery pack used to power the motor and one for the 20V Lithium Ion battery pack to power the electronics.

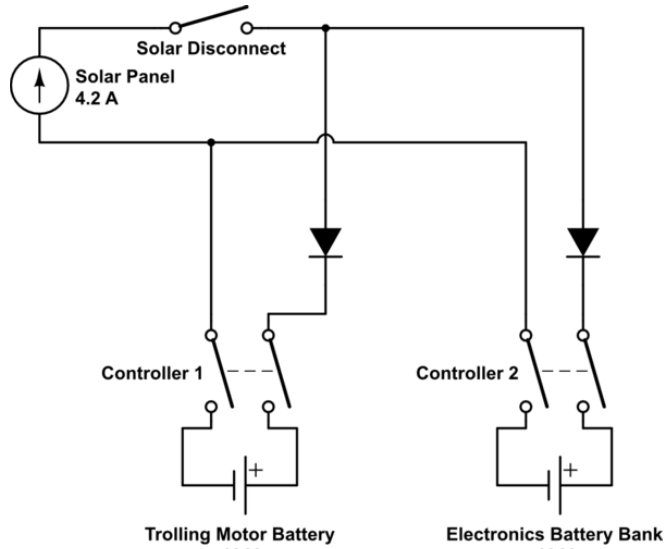


Figure 33. Solar Panel and Charge Controller Integration

Each charge controller is equipped with three sets of ports. One to connect to the solar panels, one to connect the battery and one called load. The LOAD or LVD output is use with smaller loads, such as small appliances and lights. The advantage is that the load terminals have a low voltage disconnect, so it will turn off whatever is connected to the load terminals and keep from running the battery down too far. The LOAD output is often used for small non-critical loads, such as lights. Since we will be utilizing the charge controller to maintain a power to the entire system of the vehicle the load terminal will not be used for either controller.

Figure 34: Standard Input/ Output Port options for Charge Controllers

5.4.2 Battery to Component Conversion

A major consideration for the power system of the Solar Vehicle is how power is distributed throughout all the components.

5.5 Powertrain

This subsystem controls the forward and backward movement of the vehicle. Each mechanical engineering group submitted a choice of their preferred motor. The ZXTDR, model M1020, electric motor, along with an appropriate gearing system, will be used to control the rear axle of the vehicle. This motor has a maximum draw of 21 amps at 2500 RPM.

5.5.1 Motor Controller

The motor controller subsystem consists of an Arduino connected to the NVIDIA Jetson TX2 using Serial Peripheral Interface (SPI). The Arduino will implement a Proportional-Integral-Derivative (PID) controller using a Hall Effect sensor as feedback. The Jetson will send information about the required rpms to the Arduino, and the Arduino implementation of a PID will continuously calculate the error value e(t) as the difference between a desired setpoint SP = r(t) and a measured process variable PV = y(t), and applies a correction based on proportional, integral, and derivative terms. The Arduino will also send information back to the Jetson about the current rpm of the motor.

A PID Library developed by Brett Beauregard is available to use for Arduino (Permission to use pending). This subsystem will use the MD30C 30A DC H-Bridge Motor Driver by Cytron. This driver will be connected to the battery and controlled by the PID controller implemented in the Arduino.

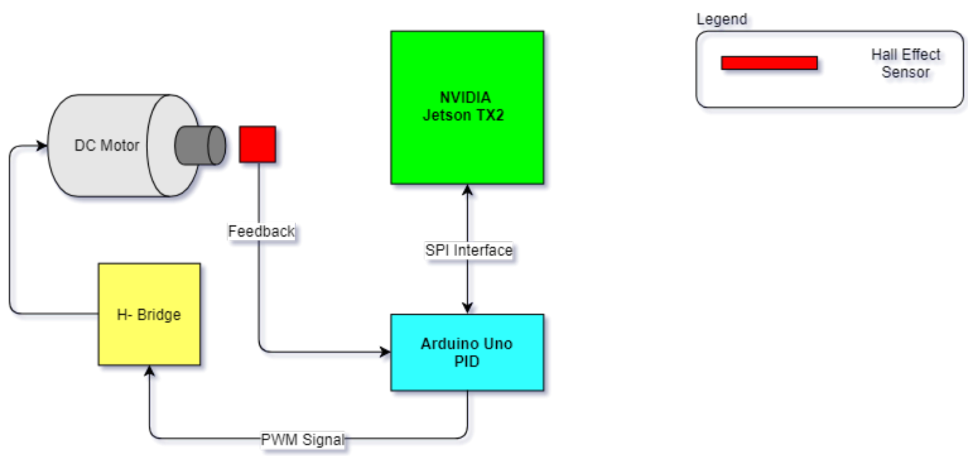


Figure 35. Motor Controller Subsystem

5.5.1.1 Breadboard Test

The proof of concept was tested using components that would resemble the actual system. A 5V DC motor was used, along with an L298N H-Bridge driver. A 3441 Hall Effect sensor was used to detect the magnet glued to a bottle cap that was attached to the motor

(See figure below). This prototype system was implemented using an Arduino Uno. The speed of the motor was regulated using Pulse Width Modulation (PWM). The RPM were calculated by detecting the magnet attached to the motor and calculating the time that it took for the magnet to be detected again. The figure below shows the breadboard prototype test.

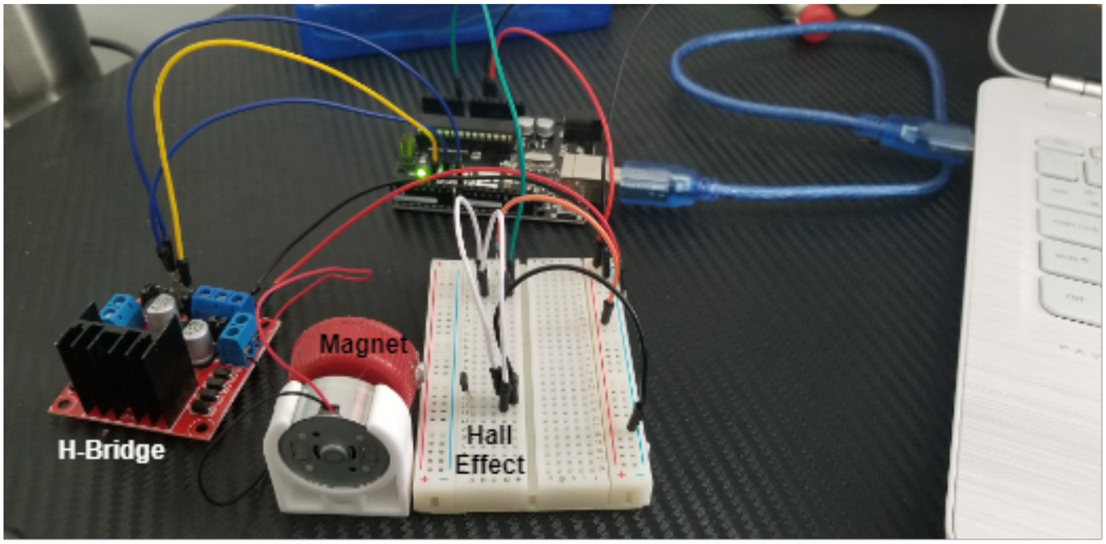


Figure 36. Motor Controller Prototype Breadboard Test

5.6 Software Design

5.6.1 Project Design Details

The solar vehicle's software portion will be implemented using a cluster of two ATMega328 microprocessors in parallel with both the Jetson TX2 developer's kit onboard camera module and GPU, which will rest on the Tegra system-on-chip. The microprocessor will take input serial inputs and produce serial outputs which then will be used as triggers or flags inside of the Arduino microcontroller's code.

The robot is projected to be completed to its entirety in early April or late March of the upcoming spring semester. With that, we are using a framework that will allow us to quickly develop new code that is written. In summation, there are several features or properties we expect the testing framework to have. The features of the software's framework should include easily accessible, robust simulation environment, accurate simulation environment, and unit test support.

5.6.1.1 Software Modules (Subsystems)

For the overall system to work as planned we have constructed a software module which consists of flow diagram chart or State machine to handle and control certain tasks. The software modules are broken down into software subsystem designs: System Check, Camera Read, Sensor Read, Stop, Acceleration, and Steering.

Based on the Vehicle Software State Machine in Figure 50, the software design flow will begin with a system check. After a validation of system check there will be a follow up read; a read of the onboard camera module. If the system check returns a negative validation i.e., if the system is NOT OK, then, log the error into a text file on the Jetson TX2 ROS topic to return which of the subsystems failed. Both the camera read, and ultrasonic sensors read will be occurring simultaneously using an algorithm that implements parallel-programming for the data reads. Ultimately, this read in parallelism is an act of fail-safe of object detection.

The camera is to detect objects and report them specifically using our OpenCV robot vision algorithms while the ultrasonic sensors are to read actually how far or close, rather a detected object is away from the physical solar vehicle. Specifically, if an object detected is within certain range (less than 5 feet) then the program will trigger a flag to the ATMega328 processor and go into an Object avoidance state. In the Object Avoidance state, this program design includes motor control using the Hall-effect sensor, and Arduino Uno microcontroller. While the objected detected is within 5 feet, the program's algorithm is set to reduce the speed of the solar vehicle. This reduction of speed is for the act of giving an object or obstacle detected time to move out of the vehicle's detection field of view.

If the object or obstacle detected has not moved on its own using our obstacle avoidance algorithm, the vehicle will make a critical selection to either steer left or steer right to avoid or move around the object. If an object is still detected but has reached our critical detection range of an object, then the robot will terminate immediately. Internally, the program has switched to an idle or Stop state. This abrupt termination of the solar vehicle is to be acting as an emergency shut off or power down switch since the solar vehicle is to be fully autonomous. While the robot is in the Stop or idle state, it shall then continue to read and sense data input from the TX2 onboard camera module.

The camera is designed to use both OpenCV and Caffe frameworks which will later be explained further in detail the actual design and implantation of these frameworks being used. In the essence where no object is being detected, we have a hall-effect sensor programmed in with the Arduino microcontroller acting as the motor controller sending serial feedback to the TX2 on how fast the robot is moving. While no object is detected the speed gradually increases topping out at a max speed of 5mph.

Since we are working with three different teams for the solar vehicle robot challenge, we have implemented a class using a data structure to store the different parameters or variables of each team. The generalized program function will consist of a single struct or class for RPM calculations for each team. We will pre-store three separate programs in memory on the Jetson TX2.

To access a specific program, we will toggle the user-Defined button (Vol) on the TX2 development board. The user-defined button will also be pre-programmed using ROS environment and packages. The program will be written in ROS with the pre-determined variables needed to calculate the RPM or top speed. The top speed for all robots is 5mph, however, each team will be using different tires, and steering controls.

The program will include the parameters of the tires (in inches), and dynamically calculate the RPM then send this information over a serial output to the other ATMega328 microprocessor. This dynamic programming gives us the max speed detection while setting the hall-effect sensor's parameter. Once the max speed is reached the program will continuously perform input reads from the camera and ultrasonic sensors.

This concludes the overall software module design, now we will go into each of the specific states to express the software design on how each state will be implemented.

5.6.1.1.1 System Check

As mentioned above the System Checks involves the Power, Arduino microcontroller, all six of the ultrasonic sensors and finally, the onboard camera module. This system check will be implemented using a tree-like data structure along with a linked-list data structure to handle dynamic programming. The root node will be the state of the System with subCheck being sub-roots of the tree structure.

Using a Breadth-First-Search (BFS) algorithm makes most sense to use in our software design because without any power nothing else can be moving. If there is no power coming from the TX2 development board we then cannot receive any data from the Arduino microcontroller, which in turn, will aid in powering the six ultrasonic sensors in parallel. The camera is the final subsystem check to be completed in the tree structure search.

5.6.1.1.1.1 Power

The Jetson TX2 has an operating voltage range of 5.5 -19.6V. We will have an external program running in the ROS environment. The programs tasks is to determine when the red LED on the Jetson TX2 development board turns off. This will be a simple program implemented using the built-in system clock on the Jetson Development board reading from J21 header of pin 5 for the clock.

Using pseudocode for power systems check:

- If J21 Header (pin 5) goes low after 10 onboard Serial Clock (SClk) seconds
  - Then Power check is good
- Else
  - Error log_# write error log to text file
- Recheck pin 5
  - While red LED is high (or on).

5.6.1.1.1.2 Arduino

The Arduino microcontroller check will be an internal program written using the tree data structure. There are several checks that needs to occur to make sure each microcontroller is good to perform. Given the analog voltage read from the above power system check, this voltage value will in turn, be an input parameter for comparison of the Arduino's output voltage using unit testing.

5.6.1.1.1.3 Sensors

We will be using six MaxBotix MB1000 ultrasonic sensors in parallel to cover all blind spots of the autonomous robot. This is imperative, so the code implemented will be done in either C++ or C-programming.

The choice to use C/C++ is for the latency to throughput comparison. Given that there is a trade off to the CPU with respect to the execution time and jobs processed we intend to sacrifice throughput to low-latency (execution time). There is a challenge our programmers are facing is that is the clock cycles between the microcontroller and Jetson TX2. The programmers are working on a solution that will correct this issue. We will parallel program with the Arduino and TX2 module with ROS to override the Arduino's clock cycle. In a nutshell, speedup the microcontrollers CPU clock cycle time or substitute it for the CPU clock cycle of that of the Jetson TX2's.

Implementation is currently pending and in premature design phase. While we have the six ultrasonic sensors in series (or parallel) we will perform an internal program check to confirm that each sensor (in series) has an operating voltage of 2.5 to 5.5v.

5.6.1.1.1.4 Camera

If the BFS leads to the camera being check this concludes that system checks is almost complete. The camera has a check of input detection and idle voltage of 25mW. The pseudocode for this is as follows:

- If (onboard_active) is HIGH
    - Continue
- Else
    - Error log# onboardCam to the text file
    - Goto Power check

5.6.1.1.2 Module Functionality

This section discusses the functionality of each of the subsystem's functionality. The camera, sensors, steering, stop and start will all be discussed briefly.

5.6.1.1.2.1 Camera Functionality

The onboard camera will be used as an input for object detection. We will be using the deep learning framework Caffe for neural-network training. The camera will be trained to recognize humans, birds, walls, boards, vehicles, turtles, and traffic or construction cones. This will all be convoluted into a package and stored onto the TX2 development board's RAM.

5.6.1.1.2.2 MB1000 Sensors Functionality

The six MB1000 sensors rely heavily on the ATMega microprocessors so the algorithm we intend to use will rely on Pulse Width Modulation converted to analog. Using the *anologWrite()* function and calculating the duty cycle on the input. The input sensors will also use if statements for object detected in close range. The sensors are strategically placed upon the vehicle's apparatus base on Figure 37: Obstacle detection top-level view.

All six cameras will be running in parallel on the microcontroller but will be using different object detection algorithms. For example, the two sensors in the front of the vehicle will be detecting not only how far away an object is but also will determine the next software state to enter; Stop or Steer left or Steer right.

The two MB1000 sensors on the vehicles side will be used for oncoming objects such as pedestrians, wildlife, or other moving vehicles. The algorithm to be implemented for side sensors will be involve close range triggers and require speedup of the motor controller or immediate steering using a mealy system.

The rear sensors are used to detect if an object is behind or approaching it from a distance of 5 feet or less. The rear sensors will strategically be sync with the front cameras in the state where an object detected in the front has become A) Too close. B) Not moving out of the way but is too close.

5.6.1.1.2.3 Stop Functionality

This Stop/Idle state will be implemented using a Mealy State algorithm. The Mealy algorithm from digital systems basically involves an emergency stop or reverse state. Essentially, if an object detected comes too close to the vehicle or if either of the six sensors detects an object to be within 3 feet within 3 seconds, then the robot's program will immediately return an Idle or Stop state for 4 seconds or 4000 sleep seconds, programmatically speaking before continuing the state machine's algorithm.

The robot will be programmed to stop in the event there is a sudden voltage drop detected in the microcontroller or Jetson TX2 developer's board. A sudden drop will be flagged via Arduino microcontroller. The pseudocode for this Mealy algorithm will closely be resembled as follows:

5.6.1.1.2.4 Acceleration

Acceleration state will require the motor controller to be controlled using the Arduino microcontroller. Acceleration will be also programmed in a manner that is team specific with respect to each of the solar vehicle competitors. Acceleration relies on the parameters explained earlier in the design phase, so we will apply the same concept and attach code to each user-defined setting on the Jetson TX2.

The robot will be programmed using the hall-effect sensor and Arduino to control the motor packaged together in the ROS environment. Relying on state algorithm the program should be able to respond accurately after passing each state and test accordingly.

5.6.1.1.2.5 Steering

Steering will be electronically controlled via a linear actuator however, we program set values using a HashMap and transfer function value to determine how each team's degree of steering will be implemented programmatically. Due to financial reasons with the university's accounting we are currently waiting to purchase a linear actuator to generate the transfer function with HashMap key values.

5.6.1.2 How It All Works (Using ROS)

5.6.1.2.1 Software Design Using ROS

As mentioned in the earlier software specific sections, ROS or Robotic Operating System is not an actual full-scaled OS. ROS is powerful in its own rights. The code functionality described in this entire section will be implemented then transferred to the ROS environment which is housed on the Jetson TX2 developer's board. ROS comes with some pretty neat tools and libraries which makes it easy for developers and ioTs projects come to life from design. The tools needed for our design will include the Computation Graph, nodes, and topics.

5.6.1.2.1.1 Communication Package (Computation Graphs)

ROS has the peer-to-peer network that essentially packages all data together using several concepts. The concepts as mentioned previously include nodes, topics, bags, services and messages to name a few.

5.6.1.2.1.1.1 Nodes

Nodes in ROS are not like nodes in tree data structures. Nodes generally handles computation in ROS. Every state described above in our functionality module design will be thrown into a node in ROS. For example, the motor controller, the sensor read, the camera read, the System Check states will all be in a node separately. All of these nodes will be implemented using a client library support of rospy or roscpp for Python or C/C++, respectively. An example implementation of how a state will be converted to a node in ROS written in Python is as follows:

Rospy.init_node('STOP') This creates and initializes the State Stop as a node in ROS in the python.

5.6.1.2.1.1.2 Topics

Nodes publish messages or information to topics in ROS. Topics are essentially key names that will be given to describe a certain type of information being passed. Using the example above in the Nodes section, a topic name would possibly be 'stop' or function name could be called 'stop'. This is a strong feature that will be heavily implemented since the Arduino and TX2 will be the brains of the robot running.

5.6.1.2.1.1.3 Bags

Bags will be crucial for our software design when it comes to CameraRead, SensorRead and System checks. Bags essentially hold and store the data that was previously read. Using the concept from previous section of *topics*, we can pass the data from the CameraRead, SensorRead and Systems power checks to the Arduino topics which reduces our programs latency.

5.6.1.3 Simulation

The simulation must be as accurate as possible to the environment in which will test the solar vehicle robot. Ros has several test environments that are commonly used for robotics. The simulations will occur on Ubuntu 14.04 using ROS. Gazeabo and turtlebot simulation being two starter simulation environments.

5.6.1.3.1 Robot Vision

We expect that simulating any robot vision algorithms accurately in a digital environment will be extraordinarily difficult. Most three-dimensional models depend on the use of polygons to render representations of a three-dimensional world.

Polygons are usually acceptable to use in many three-dimensional video games, since video games are works of art and do not necessarily try to match reality. But depending on the robot vision algorithm that we use, a polygonal universe might not represent reality with enough accuracy. The sharp edges, unnaturally flat surfaces, and lack of detailed shading apparent in many simulated universes may cause discrepancies in our algorithms' performance between simulated spaces and reality.

It has been suggested that we simply neglect robot vision algorithms in our simulation if we need to, and automatically assume a certain (data-driven) percentage of accuracy in robot vision algorithms if we need to. In this scenario, the robot vision algorithms would be tested separately, entirely separate from the rest of the simulation.

5.6.1.3.2 Physics

The robot's simulation environment must be in a three-dimensional space. The algorithms we intend to use needs to deal with the perturbations and movements of the solar vehicle robot, which will be challenging using a two-dimensional object.

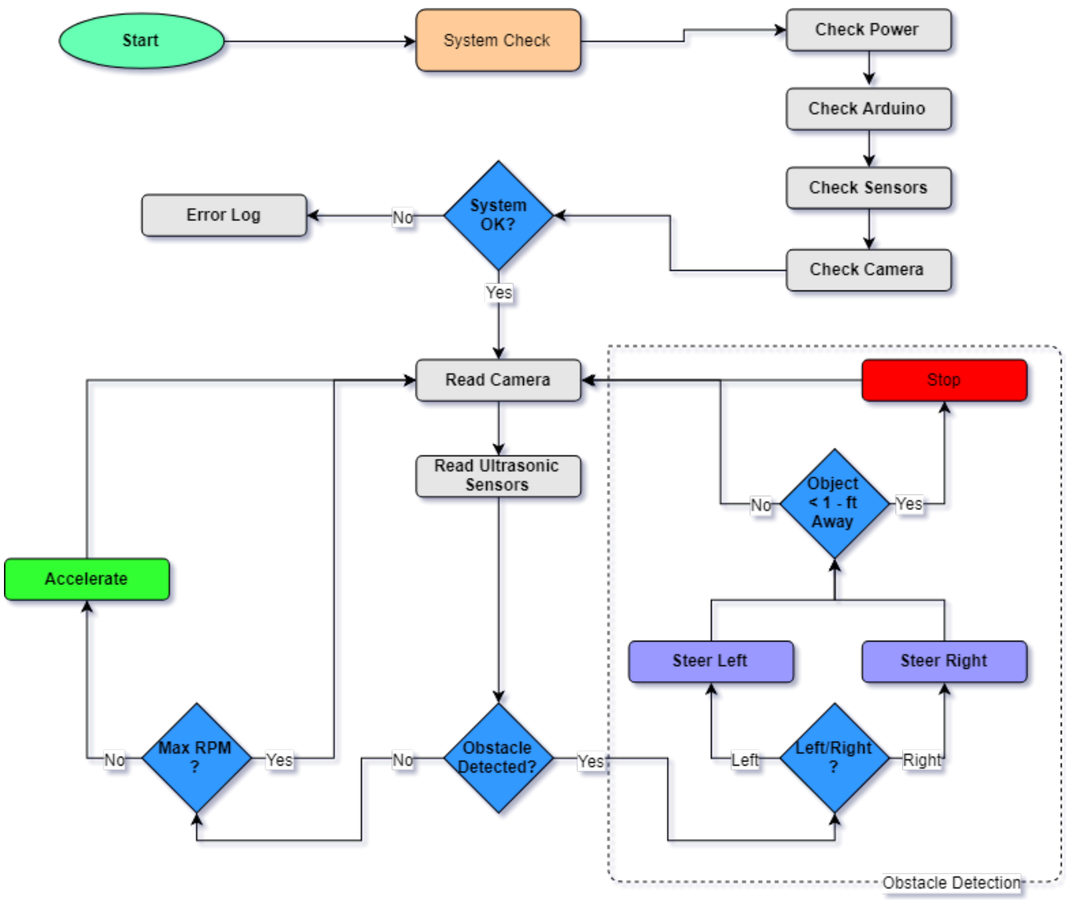


Figure 37. Vehicle Software State Machine

5.7 Summary of Design

This section covered the details of integrating the major hardware components into the system. The main system was divided into four subsystems that included obstacle detection, navigation, power, and powertrain.

The obstacle detection subsystem includes six MB1000 LV-MaxSonar-EZ0 ultrasonic sensors from MaxBotix. The six sensors will be placed around the vehicle covering most blinds spots in the in the intended path of the vehicle. A method know as chaining will be implemented to avoid noise (cross-talk) between sensors. This method is provided by the MaxBotix company and permission to use this method has been acquired. The sensors will be managed by an Arduino Uno. The Arduino will take the sensor data and calculate the distance to objects in front or around the vehicle. The data will be sent to the NVIDIA Jetson TX2 using I2C interface. Also, the power requirements of the Arduino Uno will be met by the Jetson development board.

The navigation subsystem includes an Inertial Measurement Unit (IMU) sensor, providing meaningful data to the navigation algorithm, and a linear actuator to steer the vehicle. The Adafruit BNO055 IMU will be connected to the Jetson TX2 using I2C interface. The sensor will provide data with regards to cardinal directions (Points). Data pertaining to the acceleration of the vehicle could still be utilized if need be. The linear actuator will be utilized to translate the navigation commands to the physical vehicle by turning the wheels left or right. The linear actuator will be connected to an RC Relay that would be connected to the Jetson Board. Ultimately, it will be mounted on the chassis of the prototype vehicle.

The power subsystem is composed of the solar panel, charge controller, and batteries. The solar panels will be supplied by the Mechanical Engineering teams and the solar panel's power output will vary from vehicle to vehicle. The charge controller is selected to meet the maximum power output of the selected solar panels. The charge controller will be connected to the two lead-acid batteries that will power the vehicle. A separate lithium ion battery will be used to power the more sensitive hardware components to avoid damage created by the back EMF created by the DC motor.

The powertrain includes the DC motor that would provide the forward and backward motion of the vehicle. The 24V 500W motor will be attached to the rear wheel axel of the vehicle. The acceleration and deceleration of the vehicle will be managed by an Arduino Uno implementing a PID controller. The Arduino will connect to the Jetson TX2 using I2C interface. An H-Bridge motor driver will sit in between the Arduino and the DC motor to manage the speed of the motor using Pulse Width Modulation (PWM). To complete the closed loop system of the Arduino PID controller a Hall Effect sensor will be used as feedback. The Hall Effect sensor will detect a magnet attached to the Motor shaft and actively calculate the RPM of the motor.

The NVIDIA Jetson TX2 would be acting as the brain of the system. It links to every major subsystem and it will be running the object detection and avoidance requirement of the project while navigating a closed course.

6.0 Project Prototype Construction and Coding

6.1 Integrated Schematics

The scope of the project requires the use of two Arduino Uno to fulfill the requirements. An over the counter Arduino has many components that are not necessarily essential to get the desired functionality. The Arduino Uno microcontroller was taken to “the drawing board” to remove the components that our project was not going to make use of. Using the Autodesk Eagle software, a new Arduino was designed including only the ATMEGA328P microprocessor and some basic but essential components. The figure below shows the Eagle CAD schematic for the ATMEGA. It was decided to include all the pinheads that would normally would come on the over-the-counter Arduino Uno board to allow for flexibility when it comes to building the prototype.

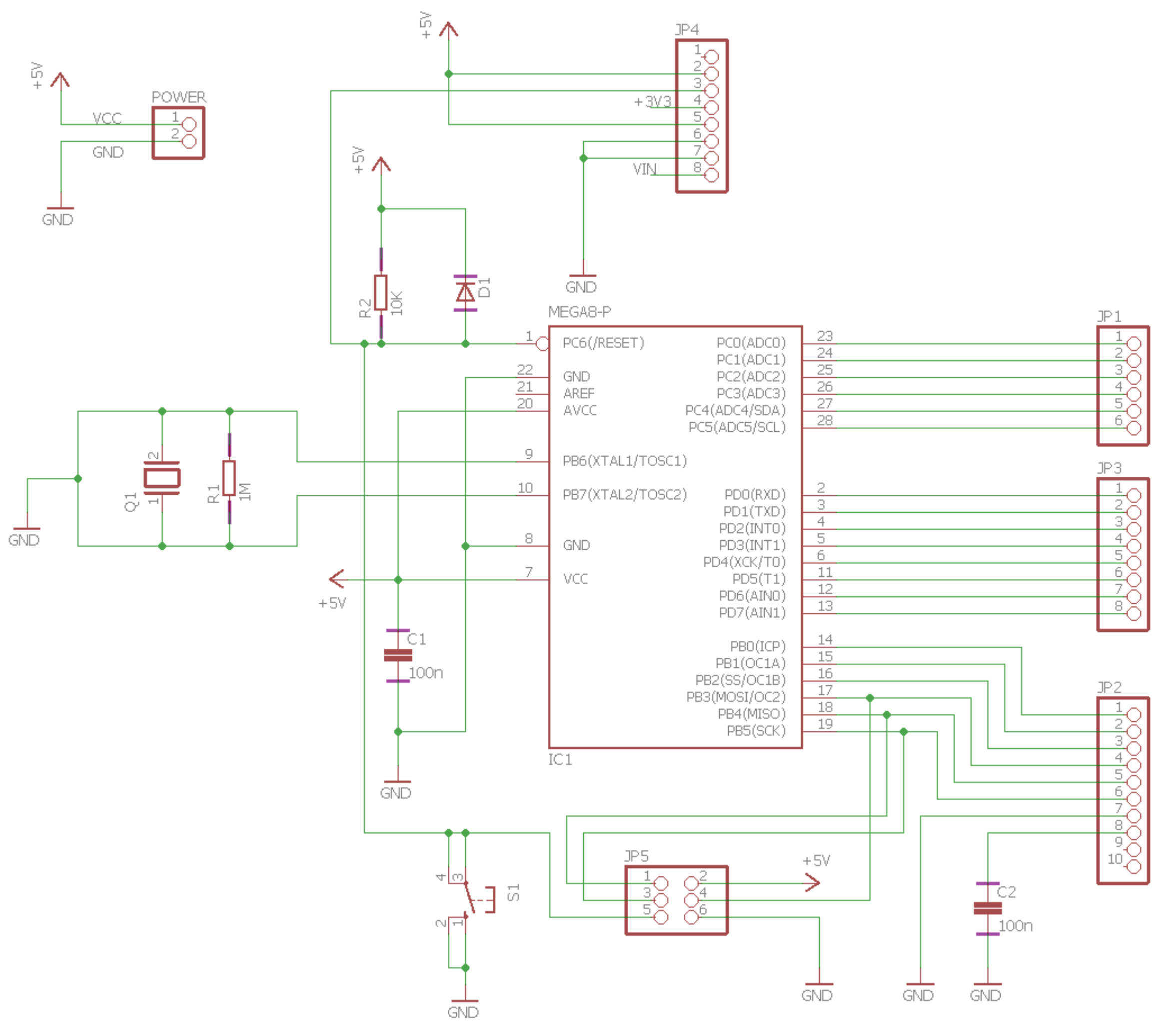


Figure 38. Arduino Uno Eagle CAD Schematic

The process of the design was a lengthy one that included many hours of researching how to use the CAD software and getting accustomed to the functionality of it.

6.2 PCB Vendor and Assembly

The Printed Circuit Board (PCB) consists of a substrate that contains tracks, lines and paths to connect different electronic components. It is made of a special material on which copper material is laminated that would create a path of conduction between the components. There are different types of PCB design that include single sided, double sided, multilayered, rigid, high frequency, etc.

Single sided PCBs are usually fabricated using only one layer of the base material. On one side, a metal lamination takes place to create the electrical connection between the components soldered on the board. Double sided PCBs are very similar to single sided PCBs. They contain a copper lamination on both sides of the substrate material. Multilayer PCBs can be a little more complex in design by containing a combination of single sided and double-sided boards. Insulation is usually put in place to provide protection between boards.

The PCB design for the project in scope will make use of a double-sided PCB board. The decision is made on the complexity of the design. To avoid overlap in the lines that connect the multiple components being used, a double layer provides access to the second side of the PCB to run connection. The connection between the layers is done through VIAs. The figure below shows the final PCB design of the ATMEGA328P that will be used for the project.

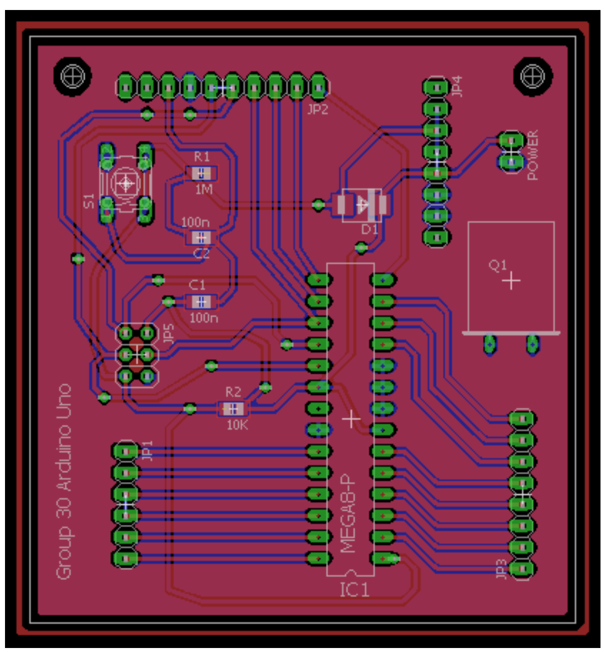


Figure 39. ATMEGA328P Final PCB Design

6.2.1 PCB Design

Designing a PCB has a set of rules that can aid in achieving a product that achieves a desirable functionality. Knowing what measurement units are being used is of great importance, as well as knowing how they are used and their common terminologies. When the PCB design phase is under way, it helps to lay out the board on a fixed grid. Snap grids are extremely useful as they components and tracks will “snap” into fixed grid positions. This feature is provided by the Eagle CAD software along with options to vary the grid size. The size of the tracks used in the design of the PCB is another aspect that can be easily overlooked when first getting into PCB design. Track size will usually depend on the electrical requirements of the design. It is not unusual to see PCB designs with multiple track sizes. Bigger tracks offer a lower direct current (DC) resistance and often times are easier and cheaper to manufacture.

Another feature that makes double-sided PCBs easier to design and layout are VIAS. They are a way to connect tracks from one side of the board to the other. VIAS are made using electrically plated holes called Plated Through Holes (PTH). Polygons are another feature that is being used in the PCB design of the project being discussed in this paper. Polygons are used to fill in desired areas of the board with copper which fills around tracks. It must be considered when the designing the PCB the clearance between tracks and VIAS to avoid shorts or any other problem that could arise from the manufacturing process. The location of components in the board may be one of the most important aspects to keep in mind when designing a PCB. A good component placement will ensure that the layout process doesn’t become more complicated than it would otherwise. Some important steps to follow when designing a placing component in the PCB are:

- Make use of a visible snap grid and default track/vias sizes.
- Put down all the components on the board before making any connections.
- If possible, divide the components into functional blocks.
- Layout critical tracks first.
- Blocks should be place and routed individually.
- Once blocks are completed, move into position on main board.
- Route any remaining power and signal connections.
- Do a final check.
- If possible, have someone else check your design.

7.0 Project Prototype Testing Plan

This project has a requirement of integrating the hardware component suite on the three separate vehicles. The availability of this vehicle might not be available in the time frame ideal for proper design and testing. A prototype will be obtained to create a base on which the functionality of the acquired components can be tested. This will allow us to start testing early and to adjust components or connections on the system as necessary.

7.1 Hardware Testing Environment

The vehicle being designed makes use of various hardware components that will be connected together to achieve autonomous navigation and obstacle detection. Multiple tests will take place ensuring every hardware component has the desired functionality and integrity required to fulfill its intended task. The hardware components will be tested individually before integrating into the prototype.

The integration process will be carried by testing the major subsystems before bringing integrating into the main system. Individual subsystems will be tested indoors. It is important to isolate the effects of indoor and outdoor testing on components that have a sensitivity to direct sunlight such as the camera. Hardware components like the NVIDIA Jetson TX2 and ultrasonic sensors can have negative effects when exposed to wet and/or very humid environments.

The indoor testing environment will consist of the University of Central Florida Senior Design Laboratory. The university provides lab access to students Monday through Friday 6:00am to 10:30pm and 7:00am to 5:00pm on weekends. The lab comes with ten working stations that provide the following equipment:

- Tektronix MSO 4034B Digital Mixed Signal Oscilloscope, 350 MHz, 4 Channel
- Tektronix TDS 2014B Digital Mixed Signal Oscilloscope, 100 MHz, 4 Channel
- Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
- Tektronix DMM 4050 6. Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply
- Dell OptiPlex 990 Computer

7.2 Hardware Specific Testing

7.2.1 Ultrasonic Sensors

The ultrasonic sensors will be initially tested indoors to ensure functionality and the expected working angle. There are three different chaining applications that would need to be fully tested to draw a conclusion on which to use. The position of the sensors calculated in section 5 will be tested on the prototype. There will be two sensors directly in front of the vehicle, two more sensors on the sides at a 55-degree angle, and two more sensors in the back of the vehicle. The subsequent testing should give information with regards to the sensor's ranging distance, effective working angle, consistency of readings, and power consumption.

7.2.2 Motor

The initial testing phase of the motor included a small prototype to test the proof of concept. A 5V DC motor was acquired along with an L298N H-Bridge driver and using a Hall Effect sensor to read the RPM. This prototype was tested using an Arduino Uno. The speed of the motor was controlled through Pulse Width Modulation (PWM). The next phase of motor testing will need to include the selected DC motor. The motor will be connected in the same way as the prototype ensuring that the power requirements are met. A PID library will be uploaded to the Arduino and the values of K_P, K_I, and K_D will need to be adjusted to achieve the desired motor control. Initially this test will be done in the Senior Design lab. The next phase will include mounting the motor on chassis of the prototype. Outdoor testing will proceed giving realistic results of the behavior of the motor control.

7.2.3 PCB (Arduino)

The PCB includes a simplified design of an Arduino Uno board. The design will be tested to follow the device's specifications. Some of the testing protocols that would follow are:

- Initial inspection of PCB board to ensure proper fabrication
- Successful initial power up
- Successful flash of the ATMEGA328P
- Pin functionality and current draw of 20mA to 40mA per pin
- Power supply pins deliver the correct amount of power

A simple program will be loaded into the ATMEGA328P to test the overall functionality of the board. The following test will include adding the components such as the ultrasonic sensors and running more complicated tasks. In the off chance the board does not meet the specified requirements, an evaluation will take place to pin point the problem. An additional re-design can take place.

7.2.4 NVIDIA Jetson TX2

The Jetson TX2 will be initially test indoors out-of-the-box with a flash of the latest software. For the sake of the solar powered robot we will run a stress test on the system-on-chip. The TX2 comes with two separate power settings with Max Q and Max P being maximum performance and maximum power consumption, respectively. Assuming all software has been loaded onto the Jetson TX2, we can run a performance (Max Q) test after changing the settings to max performance using the following command line:

- Sudo nvpmodel -m0
- Sudo /home/nvidia/jetson-clocks.sh

To monitor the GPU AND CPU performance we will then run the following command line and monitor the resources of interest.

- sudo /home/nvidia/tegrastats

7.2.5 Charge Controller

The initial testing for each charge controller will happen separately outdoors connected to one of the mechanical engineering groups Solar Panel PV Arrays and then connected to its corresponding battery. Before the charge controller testing can begin first each battery must be connected to its components it is powering to ensure the power is being distributed properly from the batteries to their corresponding subsystems.

Figure 40. Charge Controller Testing Setup with Solar Panel

The MPPT charge controller will connect from the solar panel array to the 24V Lead acid battery. Using a multimeter, we will measure the amount of voltage being fed into the lead acid battery. This number should be as close to 24V as possible. Since the lead acid battery pack consists of two 12V batteries in series they should pull around a consistent 24V from the solar panel array. If this number is measured on the multimeter then this test can be considered successful and all wiring will be disconnected.

The PWM charge controller will connected from the solar panel array to the 14.8V (4S) LiPo battery pack connected to both the linear actuator and to the Jetson TX2. Again, using a multimeter and the LCD screen on the charge controller we will measure the amount of voltage passing through the charge controller into the battery. This number should read close 14.8V but not higher to ensure there is no overcharging. If this number is measured on the multimeter then this test can be considered successful and all wiring will be disconnected.

Lastly, we will need to ensure that once the two charge controllers are connected in parallel to the solar panel array that they are still able to properly charge both the 24V lead-acid battery pack and the 14.8V LiPo battery pack. To do this we connect the charge controllers and its corresponding components to the solar panel array, ensuring the charge controllers are connected in parallel. If we accidentally connect the charge controllers in series there will be too large of a current passing through the wired connections and may damage the charge controllers. By connecting the charge controllers in parallel we can ensure the current is being split between the two devices. Again, using a multimeter and the LCD screen on the PWM charge controller we will measure the amount of voltage passing through each charge controllers into its corresponding battery packs.

7.3 Software Test Environment

There are three main classes of obstacles that our vehicle will need to navigate around in order to complete the course.

The first type is the obstacle course border. The vehicle will be limited to movement within this defined, enclosed space. Our navigation system will need to process these borders and generate a viable path between both sides, preferably down the center. Any obstacle avoidance maneuvers will also need to account for the course walls, preventing a collision or movement out of the course bounds. These boarder walls will have a specific design associated with them, so that our computer vision component will be able to correctly recognize them.

The second type of obstacle to account for are stationary obstacles. These will be obstacles that have a fixed position within the course that will obstruct the forward progress of the vehicle and require an avoidance maneuver. These stationary obstacles may or may not be reoccurring, meaning that for each loop around the course, the vehicle will encounter the same object. An avoidance maneuver will only be taken if an object directly impedes the forward progress of the vehicle. Our vehicle will also encounter speed bumps, classified here as an obstacle our system may be capable of recognizing and processing as an obstacle, but one that will not impede the progress of the vehicle and as such should not influence navigation 9.

The third type of obstacle the vehicle will encounter are moving obstacles. These are objects that impede the forward progress of the vehicle but are either in a constant state of motion of stationary objects which appear irregularly in different locations, varying from lap to lap. Objects which are in perpetual motion, that impede progress, will halt the vehicle until removed. Static obstacles that the vehicle will recognize as new to the course, will need to be bypassed. Although response action is a simple stop, it means that our system will need to differentiate between a static and moving object throughout the entirety of the course run.

The Robot Operating System commonly known as ROS is not a full-scaled operating system. It is in between a middleware and operating system for robotics. To be a bit more convincing, as of today, ROS only runs on Unix-based platforms. Although ROS is not a real-time framework, it is capable of being integrated with real-time code. Primarily, ROS is tested either on Mac OS X or Ubuntu systems.

An interconnected cluster of jobs or ROS processes that processes data on a peer-to-peer network is called the Computation Graph. It is stated that messages, services, nodes, Master, topics, and bags, all supply data to the computation graph in a variety of ways.

7.4 Software Specific Testing

7.4.1 Navigation Implementation

Autonomous vehicle navigation descriptions and algorithms rely on being able to track the vehicles final target and modeling areas surrounding the obstacles within ellipses that define where the vehicle is not allowed to go [3]. This brings up some changes we will need to consider when looking at his approach for our system.

The first is that we will have an undefined true final target. Our vehicle's goal will simply be forward for as long as possible. For our purposes, our final target will be the farthest set-point we are able to generate, which will vary by how far our camera will be able to reliably see. By default, our set-point should generate as to lead the car directly in between the two border walls. We should also assume that obstacles in the path will limit how far down a path we are able to see.

The second change we need to consider is modelling ellipses around obstacles to define their areas of influence. We may not be able to accurately define the areas around obstacles out of our vehicles field of view (their back-side, for example). Depending on what types of obstacles we will face, we should prioritize simply bypassing what we are capable of reading from our vehicle's field of view.

7.4.2 Algorithm: Set-Point Generation

As our vehicle reads the environment, we will need to generate set-points for it to follow through the course. As a general rule, we should try to keep the vehicle as close to the center of the course at all times, and default to creating the set-point midway between the course walls. However, there may be occasions where the vehicle is only able to see one wall at a time, such as when it is turning around a corner. In these cases, the vehicle should run adjacent to the only wall it is able read. This is to ensure that we only experience set-point creation in the context of the course, avoiding unpredictable situations. If there is ever a situation where we cannot see the course walls, we should halt.

This algorithm attempts to model how we can implement the logic above.

Input:

1) Number of course borders in view (0,1,2)
2) Distance between borders 86
3) All obstacles (with relevant features) in vehicle's field of view
4) Vehicle's dimensions and field of view

If (number of borders in view = 0)
    Error message. We should always have a course border in view of the vehicle.;

If (number of borders in view = 1)
    Generate set-point such that one side of the vehicle's field of view and the border's influence area intersect (or very close to each other; the idea is to have the vehicle run adjacent to the wall);
    Find most obstructing obstacle (Algorithm _);

If (obstacle exists && set-point causes vehicle FOV to intersect with obstacle influence area)
    Find distance between obstacle and border;
If (distance < width of vehicle)
    Choose direction opposite of wall;
Else
    Choose direction arbitrarily;
    Set-point becomes intersection point between vehicle FOV and obstacle influence area;
Else
    Set-point accepted;

If (number of borders in view = 2)
    Generate set-point such that it is in the middle of the walls;

Find most obstructing obstacle (Algorithm _);
If (obstacle exists && set-point causes vehicle FOV to intersect with obstacle influence area)
    Choose side with lowest distance between obstacle and both borders;
    Set-point becomes intersection point between vehicle FOV and obstacle influence area;
Else
    Set-point accepted;

Output: New set-point for vehicle to follow.

7.4.3 Obtaining the Most Obstructing Obstacle

Adouane provides this algorithm as way to figure out if any obstacle picked up by the vehicle's sensors disturb the path of the vehicle [3]. In the event of multiple obstacles laid out onto the course, the vehicle will need to initialize avoidance maneuvers by whichever obstacle is most in the way.

Input:

1) All obstacles (with relevant features) in vehicle's field of view
2) Desired minimum safe distance "offset" from the obstacles
3) Current final target

```
for (Each Obstacle),

        if (there exists an intersection point between vehicle's set-point and PEI of obstacle)
                add obstacle to ListObstructingObstacles;
        if(ListObstructingObstacles > 0)
                for (each obstructing obstacle)
                        find the obstacle with the minimum distance to vehicle;
                        if (two obstacles are the same distance from the vehicle)
                                choose obstacle with smallest distance between set-point and
                                the obstacle;
                        if(same distance between set-point and obstacle)
                                choose arbitrarily;
```

Output:

1) Index of the most obstructing obstacle

8.0 Administrative Content

8.1 Milestone Discussion

With this project being interdisciplinary there are certain advantages and disadvantages a typical Electrical and Computer Engineering Senior Design group does not have. The major advantage to working on an interdisciplinary project is that each group can solely focus on the sections of the project particularly geared toward their field of study. This means that the project as a whole should be above par when compared to the other singular disciplinary group projects.

The major disadvantage also deals with the fact of dealing with a larger group of people all from different engineering disciplines. Early on our group realized that streamlining communication across groups was going to be critical to the success of this project. The best way to do this was to assign a team lead to each Mechanical group. This way all communication for that team will only being going through the one person, so 21 people are not all trying to individually contact our ECE/CS group.

Next, we committed as a group to have at least one of us attend each of the Mechanical group meetings every week. This is guarantee that we are up to date on their progress and how their design of the vehicle will affect the design of our hardware/software suite.
Lastly, we constructed a tight schedule for our group to stick to. This will ensure every deadline is met, along with the Mechanical Engineering groups will understand the expectation for them to finish their design and build so to have enough time for integration and testing which will prove to be a challenge in itself before final demonstrations.

Table 17. Senior Design 1 Fall 2018 Semester Milestones

Semester 1 Week	Milestone (Task)	Start Date	Deadline(soft)	Deadline (Penalty)
02	Brainstorm (100 Ideas)	8/26/2018	8/31/2018	
03-05	Project Select/Roles	9/5/2018	9/19/2018	9/21/2018
04	Submit Project Report_1(10pg.)			
	Initial Project Document *Divide & Conquer*	9/10/2018	9/14/2018	9/22/2018
05-06	Update and resubmit Divide & Conquer File(s)	9/18/2018	9/28/2018	10/26/2018
07-11	Research Hardware/Software Packages	10/02/2018	11/02/2018	
07-11	SW Architectural Design/Layout	10/02/2018	11/02/2018	

10-11	Submit Project Report_2(60pg.)	10/22/2018	11/02/2018	11/02/2018
11-14	PCB Design/Layout Research	11/02/2018	11/23/2018	
12-13	Submit Project Report_3(100pg.)	11/05/2018	11/16/2018	11/16/2018
12-14	Integration_Testing_1 (SW & HW)	11/04/2018	11/19/2018	
14	Thanksgiving	11/22/2018	*No Classes*	
14-15	Sponsor Meeting (Stage-1 Demo) *ME Faculty*	11/27/2018	11/30/2018	
15	Finalize software and hardware design, Project Report_4(120pg)	12/3/2018	12/8/2018	

Table 18. Senior Design 2 Spring 2019 Semester Milestones

Semester 2 Week	**Milestone (Task)**	Start Date	Deadline(soft)	Deadline (Penalty)
Assume deadlines are due by 12PM EST.				
01	Assemble/Build Prototype	01/07/209	01/12/2019	
01-03	Acquire Hardware Components, Update MEA with Hardware Placement	01/12/2019	01/22/19	
03	Martin Luther King Day	1/21/2019	*No Classes*	
03-04	Integration Testing (SW & HW)	01/22/2019	01/31/2019	
08-10	Complete ECE prototype demo	3/1/2019	3/12/2019	
13	Start Integration with ME team	4/1/2019	4/8/2019	
15	Final Competition between ME teams	5/1/2019		

8.2 Budget and Finance Discussion

The table below contains the estimated finances for this project. The prices listed are taken from the manufacturer's web pages. The prices are not final and are subject to change once the final implementation begins.

For obstacle detection and avoidance, the vehicle will need six MaxBotix MB1000 LV-MaxSonar-EZ0 ultrasonic sensors. Two sensors for the front are of the vehicle, two at the back of the vehicle if reverse capabilities are implemented, and one on each side to avoid objects parallel to the vehicle. The cost of each ultrasonic sensor is $27.95 plus $2.95 in shipping and handling costs for a total of $170.69. We will need two Arduino Uno microcontrollers to manage the input from the ultrasonic sensors.

The Arduino boards will be responsible for handling the configuration of the software as well as running the software. The Arduino boards will come as a donation. An NVIDIA Jetson TX2 development board will be used as the brain of the system. It will read the input from the Adafruit BNO055 IMU and Camera Module (which comes integrated into the development kit), process the data for both navigation and object detection, and output necessary to the motor controller and steering system of the vehicle.

The cost of the NVIDIA Jetson TX2 development kit with a student discount is $319.22 including taxes and an additional $11.96 in shipping and handling costs for a total of $331.18.

A Printed Circuit Board (PCB) will be used to re-design the Arduino Uno board and removing components that will not be necessary in the scope of the project. The idea is to bring the power consumption of the board as low as possible. The expected estimate for the manufacturing and shipping of the PCB is set at $100.00.

As one of the requirements, the vehicle must run completely on solar energy, so a Solar Panel will be donated for the project; however, a battery pack will be needed to store the energy from the solar panel.

The selected battery pack consists of two lead-acid batteries that have a cost of $39.78 including taxes.

A linear actuator will need to be purchased for the steering system. The expected cost of the linear actuator is of $150.00 since a decision it's still to be made about whether a linear actuator with a built-in potentiometer is to be purchased or a shaft position subsystem will be implemented.

A voltage regulator is needed to step down the voltage form the system's 24V to the linear actuator's 12V. The expected cost of the voltage regulator is expected to be $50.00. A DC motor will be purchased for the test prototype in order to design the system as similar as possible to the intended vehicle.

The cost of the DC motor is $75.00. The DC motor will require a motor controller that will be connected to a motor driver. A Cytron Motor H-Bridge driver was selected since it can handle a max current of 80A and a constant operation of 30A. The cost of the Cytron Motor Driver $35.00 plus $9.00 in shipping and handling cost.

The MYNT Eye camera is being considered and accounted for in the budget. The camera offers unparalleled technology when it comes to image processing and Computer Vision. The cost of the MYNT Eye camera is $250.00.

Since the project is on the early stages of development, a miscellaneous cost has been added to the budget to compensate for any additional hardware that might be required such as prototype vehicle, wiring, additional microcontrollers, additional batteries, materials for solar panel mount and sensor brackets, etc.

Table 19. Quantity and Cost of Hardware Components

ITEM	QUANTITY	COST PER UNIT	TOTAL
Arduino Uno	2	Donated	$0.00
PCB Board	1	$100.00	$100.00
Lead-Acid Battery Pack	1	$39.78	$39.78
Solar Panel (Donated)	1	Donated	$0.00
Lithium Ion Battery	1	$59.99	$59.99
MaxBotix MB1000	6	$27.95 + $2.95 shipping	$170.69
Linear Actuator	1	$150.00	$150.00
IMU BNO055	1	$34.95	$34.95
NVIDIA Jetson TX2	1	$319.22 + $11.96 shipping	$331.18
MYNT Eye Camera	1	$250.00	$250.00
Voltage Regulator	1	$50.00	$50.00
Cytron Motor Driver	1	$35.00 + $9.00 shipping	$44.00
DC Motor	1	$75.00	$75.00
Miscellaneous	N/A	N/A	$194.41
TOTAL COST			**$1,500.00**

Appendix A: References

[1] Energy.gov, "Solar Energy Potential You are here," USA.gov, [Online]. Available: https://www.energy.gov/maps/solar-energy-potential. [Accessed 26 November 2018].

[2] "Autopilot: Full Self-Driving Hardware on All Cars," Tesla, 2018. [Online]. Available: https://www.tesla.com/autopilot. [Accessed 30 October 2018].

[3] U. S. D. o. Transportation, "U.S. DOT Announces 2017 Roadway Fatalities Down," NHTSA, 3 October 2018. [Online]. Available: https://www.nhtsa.gov/press-releases/us-dot-announces-2017-roadway-fatalities-down. [Accessed 29 October 2018].

[4] N. C. f. S. a. Analysis, "2017 Fatal Motor Vehicle Crashes: Overview," October 2018. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812603. [Accessed 30 October 2018].

[5] "Technology | We're building a safer driver that is always alert and never distracted.," Waymo, 2018. [Online]. Available: https://waymo.com/tech/. [Accessed 30 October 2018].

[6] "The World Added Nearly 30 Percent More Solar Energy Capacity in 2017," E360, Yale, [Online]. Available: https://e360.yale.edu/digest/the-world-added-nearly-30-percent-more-solar-energy-capacity-in-2017. [Accessed 06 November 2018].

[7] A. Davison, "Common Types of Solar Cells, Common Types of Solar Cells - What are Better Silicon, Monocrystalline, or Polycrystalline Solar Cells?," [Online]. Available: http://www.altenergy.org/renewables/solar/common-types-of-solar-cells.html. [Accessed 06 November 2018].

[8] T. A. Kinney, "Proximity Sensors Compared: Inductive, Capacitive, Photoelectric, and Ultrasonic," Machine Design, 01 September 2001. [Online]. Available: https://www.machinedesign.com/sensors/proximity-sensors-compared-inductive-capacitive-photoelectric-and-ultrasonic. [Accessed 28 October 2018].

[9] "Microcontrollers – Types & Applications," Elprocus, 2018. [Online]. Available: https://www.elprocus.com/microcontrollers-types-and-applications/. [Accessed 15 November 2018].

[10] J. Califano, "How to Choose a Microcontroller for IoT," IoT Zone, 9 June 2018. [Online]. Available: https://dzone.com/articles/how-to-choose-a-microcontroller-for-iot. [Accessed 15 November 2018].

[11] "RASPBERRY PI HARDWARE," RASPBERRY PI FOUNDATION, [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md. [Accessed 20 November 2018].

[12] "ARDUINO UNO REV3," Arduino, [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Accessed 21 November 2018].

[13] "BeagleBone Black Development Board," Texas Instruments, 2018. [Online]. Available: http://www.ti.com/tool/BEAGLEBK. [Accessed 21 November 2018].

[14] "Inertial Measurement Units (IMU)," Analog Devices, Inc., [Online]. Available: https://www.analog.com/en/products/sensors-mems/inertial-measurement-units.html. [Accessed 29 October 2018].

[15] "10 DOF Mems IMU Sensor V2.0 SKU: SEN0140," DFRobot, 20 December 2017. [Online]. Available: ttps://www.dfrobot.com/wiki/index.php/10_DOF_Mems_IMU_Sensor_V2.0_SKU:_SEN0140#More_Documents. [Accessed 31 October 2018].

[16] "Learning the Basics about Batteries," Battery University | Cadex Electronics Inc., 2018. [Online]. Available: https://batteryuniversity.com/learn/. [Accessed 25 October 2018].

[17] "IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead Acid Batteries Used in Stand Alone Photovoltaic (PV) System," *IEEE Std 1361-2014,* p. 1, 2014.

[18] C. Simpson, "Linear and Switching Voltage Regulator Fundamentals," 2011. [Online]. Available: http://www.ti.com/lit/an/snva558/snva558.pdf. [Accessed 20 October 2018].

[19] "ileCAM30_TX2 - 3.4 MP GMSL Camera (supports Upto 15 Meters)," System on Module Blog, [Online]. Available: https://www.e-consystems.com/gmsl-camera-for-nvidia-jetson-tx2.asp#ar0330-camera-module-features. [Accessed 26 November 2018].

[20] NVIDIA, *NVIDIA Jetson TX1 System-on-Module | Datasheet,* 2015.

[21] "What is a Stereo Camera?," VM Resource, [Online]. Available: http://www.vmresource.com/camera/cameras-general.htm. [Accessed 15 November 2018].

[22] "ZED Introduction," Stereo Labs, [Online]. Available: https://www.stereolabs.com/docs/getting-started/. [Accessed 15 November 2018].

[23] "Tara - USB 3.0 Stereo Vision Camera," e-con Systems, [Online]. Available: https://www.e-consystems.com/3D-USB-stereo-camera.asp. [Accessed 15 November 2018].

[24] "MYNT EYE," mynteyeai, [Online]. Available: https://mynteyeai.com/products/mynt-eye-stereo-camera. [Accessed 15 November 2018].

[25] "Solar Charge Controller Types, Functionality and Applications," Elprocus, [Online]. Available: https://www.elprocus.com/solar-charge-controller/. [Accessed 27 November 2018].

[26] A. E. Store, "WHAT IS A SOLAR CHARGE CONTROLLER," altE, 2016. [Online]. Available: https://www.altestore.com/store/info/solar-charge-controller/. [Accessed 26 November 2018].

[27] B. Power, "12V/24V, 20A Solar Charge Controller for LiFePO4 Batteries (SC-122420JUD)," Bioenno Power, 2018. [Online]. Available: https://www.bioennopower.com/products/12v-24v-20a-solar-charge-controller-for-lifepo4-batteries-sc-122420jud. [Accessed 26 November 2018].

[28] GreeSonic, "GreeSonic MPPT Solar Charge Controller," GreeSonic, 2018. [Online]. Available: http://greesonic.com/index.php?route=product/product&product_id=79. [Accessed 26 November 2018].

[29] "ROS," Open Source Robotic Foundation, [Online]. Available: http://www.ros.org/. [Accessed 23 November 2018].

[30] "Python," Python Software Foundation, [Online]. Available: https://www.python.org/. [Accessed 23 November 2018].

[31] P. Rawat, "How to build a vehicle simulation environment in Unity 3D," Medium, 11 September 2017. [Online]. Available: https://medium.com/@pallavrawat/how-to-build-a-vehicle-simulation-environment-in-unity-3d-a50efd52d4ee. [Accessed 27 October 2018].

[32] "The National Institute for Occupational Safety and Health (NIOSH)," Centers for Disease Control and Prevention, 11 April 2017. [Online]. Available: https://www.cdc.gov/niosh/topics/lead/health.html. [Accessed 08 November 2018].

[33] "Code of Ethics," National Society of Professional Engineers, [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed 08 November 2018].

[34] D. Levitan, "The solar efficiency gap," IEEE Journals & Magazine, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6203950. [Accessed 06 November 2018].

[35] T. Instruments, "Switch Regulator Fundamentals," September 2016. [Online]. Available: http://www.ti.com/lit/an/snva559a/snva559a.pdf. [Accessed 25 October 2018].

[36] O. Semiconductor, "1/3-Inch Wide VGA CMOS Digital Image Sensor," MT9V024/D datasheet, December 2017. [Online].

[37] "How does LiDAR work?," LiDAR UK, 2018. [Online]. Available: http://www.lidar-uk.com/how-lidar-works/. [Accessed 28 October 2018].

[38] L. Adouane, "Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures," Chapman and Hall/CRC, 2016. [Online]. Available: http://ebookcentral.proquest.com/lib/ucf/detail.action?docID=4460245. [Accessed 29 October 2018].

[39] K. Townsend, "Adafruit BNO055 Absolute Orientation Sensor," Adafruit Learning System, [Online]. Available: https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview. [Accessed 30 October 2018].

Appendix B: Copyright Permissions

JD

Jesus Duran

Today, 11:32 AM

Jessica Schulmeister <jschulmeister@maxbotix.com>

Reply all

Good Morning Ms. Schulmeister,

I'm a student at the University of Central Florida in the United States, and I'm currently involved in a Senior Design project that involves designing and implementing an autonomous vehicle. We will be using the LV-MaxSonar-EZ0 MB1000 ultrasonic sensor in our project. We wanted to get permission from MaxBotix to implement the chaining application diagram described in the datasheet. Thank you.

Best wishes,

Jesus Duran

Re: [MaxBotix Inc.] New customer order (8241) - November 15, 2018

JS

Jessica Schulmeister <jschulmeister@maxbotix.com>

Today, 11:59 AM

Jesus Duran; sales <sales@maxbotix.com>

Reply all

Inbox

Jesus,

Thank you for your e-mail. We would be happy for you to implement the chaining application diagram from our datasheet.

Please note that our products are not designed for use in safety-critical applications (such as life support) or other applications where a failure of the product could cause severe personal injury (see our terms and conditions for full details).

If you have any questions, don't hesitate to call our technical support team at 218-454-0766. We will be happy to help!

Best Regards,

Jessica Schulmeister

MaxBotix Inc.
Phone: +1 (218) 454-0766
Direct: +1 (218) 454-7332
Fax: +1 (803) 547-4758

Email: jschulmeister@maxbotix.com

Web: www.maxbotix.com

Figure 41. Ultrasonic Sensor Chaining Permission Request

Permission to use Arduino PID library

Brett Beauregard <br3ttb@gmail.com>
Today, 9:39 PM

it's released under the mit license. you can do whatever you want with it. good for you for asking when in doubt though.

Jesus Duran
Today, 8:47 PM
br3ttb@gmail.com

Reply all

Good evening Mr. Beauregard,

I'm a student at the University of Central Florida in the United States, and I'm currently involved in a Senior Design project that involves designing and implementing an autonomous vehicle. The vehicle will use a motor controller, so I wanted to ask you for permission to use your PID Library for Arduino.

Thanks so much,

Jesus Duran

Figure 42. Permission Request to use Arduino PID Library

B

BatteryU <BatteryU@cadex.com>
Wed 11/21, 5:15 AM
Hichame Boudi

Inbox

Action Items

Hi Hichame,

Yes, you may use the material as requested. Please cite source where appropriate.

Regards,

John Bradshaw - Marketing Communications Manager
Cadex Electronics Inc. | www.cadex.com
Vancouver | Minneapolis | Frankfurt
Tel: +1 604 231-7777 x319 | Toll Free: 1-800 565-5228

Follow us on Twitter: twitter.com/cadexelectronic
Join us on Facebook: facebook.com/cadexelectronics
Add us on Google+: plus.google.com/+Cadex

Figure 43. Permission Request to use the article "Learning the Basics about Batteries," Battery University | Cadex Electronics Inc

Rahn, Christopher <cdr10@psu.edu>
Tue 11/20, 11:32 PM
Hichame Boudi

Reply all

Inbox

Sure, thanks!

From: Hichame Boudi <hboudi@Knights.ucf.edu>
Sent: Wednesday, November 21, 2018 2:18 AM
To: Rahn, Christopher <cdr10@psu.edu>
Subject:

hello Dr Rahn

I am a student at University of Central Florida. I am currently working with a group on our senior design to create an autonomous car. I am requesting a permission to use the book " BATTERY SYSTEMS ENGINEERING " as a reference on my research paper.

Figure 44. Permission Request to use the book “Battery Systems” as a reference

Issa Batarseh <Issa.Batarseh@ucf.edu>
Tue 11/20, 11:04 PM
Hichame Boudi

Reply all

Inbox

Of course. The best of luck. Let me know if you need help
Issa

On Nov 21, 2018, at 2:24 AM, Hichame Boudi <hboudi@Knights.ucf.edu> wrote:

> hello Dr Batarseh,
>
> I am a student at University of Central Florida. I am currently working with a group on our senior design to create an autonomous car. I am requesting a permission to use the book " Power Electronic Circuits " as a reference on my research paper.
>
> best regards

Figure 45. Permission Request to use the book “Power Electronics Circuits” as a reference

Ned Mohan <mohan@umn.edu>
Fri 11/23, 1:06 AM
Hichame Boudi

Reply all

Inbox

You are most welcome.
Ned Mohan
Member: National Academy of Engineering

A sobering statistic: 1.3 billion people (1/6th of humanity) have no access to electricity.

On Wed, Nov 21, 2018 at 1:27 AM Hichame Boudi <hboudi@knights.ucf.edu> wrote:

> hello Dr Mohan,
>
> I am a student at University of Central Florida. I am currently working with a group on our senior design to create an autonomous car. I am requesting a permission to use the book " Power Electronics: Converters, Applications and Design " as a reference on my research paper.

Figure 46. Permission Request to use the book "Power Electronics Converters Applications and Design" as a reference