Florida Solar Beach Buggy Challenge

Sponsored by Duke Energy

UCF

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

Group 3

Cecilie Barreto	Computer Engineering	ceciliebarreto@knights.ucf.edu
Drew Curry	Electrical Engineering	drewcurry@knights.ucf.edu
Grace Yoo	Computer Engineering	graceyoo@knights.ucf.edu

Table of Contents

Table of Tables

Table of Figures

1.0 Executive Summary

The Florida Solar Beach Buggy Challenge was created to entice undergraduate students to expand their understanding and awareness of autonomy and solar energy by challenging them to develop a fully autonomous solar powered beach buggy. Our team is one of three teams selected to design and construct a beach vehicle that will be able to trek over 10 miles of the beach extending from Daytona Beach to the Ponce Inlet and return in a span of 8 hours, with no assistance or interference from humans while being completely powered by solar energy. It must be designed to detect and avoid objects and obstacles, such as conservation zones, animals, pedestrians, structures, etc. while traveling on its excursion, 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) and must not exceed a speed of 3 miles per hour.

Each team consist of interdisciplinary senior level students who are enrolled at the College of Engineering and Computer Science at the University of Central Florida. Our team, named the Gold Team, consists of six Mechanical Engineering (ME) members, two Computer Science (CS) members, and three Electrical and Computer Engineering (ECE) members. As the ECE unit, the responsibilities of photovoltaic system design will be adhered while supporting the CS unit with establishing a system for autonomy as well as the supporting the ME unit with powering their beach buggy design.

This project is sponsored and funded by Duke Energy with an overall budget of $2,000. This budget is expected to be shared amongst the interdisciplinary groups within each team. At the end of development, which is expected to be early December, all three teams will participate in beach buddy expedition intended to test whether each team was successful fulfilling all set requirements and instituting a truly autonomous and fully solar-powered beach buggy. Ultimately, Duke Energy provided students with a rare and remarkable opportunity to learn about autonomy and power of solar energy.

This paper documents the motivation, goals, objectives, requirements, research, system designs, budgeting, and testing processes of the Gold Team's Electrical and Computer Engineering group. Existing similar projects and other researched components will be discussed. Motives for specific development decisions will be explored and various project constraints will be explained in detail. This document will also briefly touch upon the contributions of the Mechanical Engineering and Computer Science group, all together formulating unique and effective autonomous solar beach buggy.

2.0 Project Description

Duke Energy, a utility company that has approximately 52,700 megawatts of electric generating capacity and provides electricity to approximately 7.4 million customers across the United States, is challenging students attending University of Central Florida College of Engineering and Computer Science in a unique challenge involving autonomy and solar energy. The overarching goal of this project is to engage CECS students in a fun and exciting Florida Solar Beach Buggy Challenge, while teaching students about the engineering design process, and promoting interest and awareness of solar energy. Several teams of students will be selected to complete in the Florida Beach Buggy Challenge. A basis requirement is that teams consist of interdisciplinary senior level students enrolled in a CECS senior design course.

Design objectives and constraints provided by sponsor are listed below.

1. Autonomously traverse a 10 mile stretch of beach from Daytona to Ponce Inlet (and return) within 8-hour time span.
2. Capable of transporting one passenger (Max payload: 120 lbs.)
3. Top allowable speed → 3 mph
4. Run completely on solar energy
5. Do no harm to environment and beachgoers
6. Detect and avoid both stationary and moving obstacles (e.g., rocks, docks, people, birds, turtles, etc.)
7. Budget: $2000

This group is tasked with completing the electrical and computer engineering scope of this project.

2.1 Project Motivation and Goals

The world is continuing to move toward renewable energy sources as the impacts of using nonrenewable resources on the environment are becoming more imminent. Renewable energy sources are clean, sustainable, and inexhaustible. They produce no polluting emission and emit no greenhouse gases. Although it is still costly to implement, solar energy is growing in use and the price to implement it is decreasing substantially. Solar energy is the most abundant renewable energy source available and modern technology has made significant strides to effectively convert and harness its power. Solar power is power generated or created from sunlight or heat of the Sun. Once implemented, solar power systems are low maintenance and ostentatious; and the use of solar power can significantly reduce one's electricity bills and save them a great extent of money. With more neoteric research, studies, data, and awareness, people are more progressively investing in solar energy. The beneficial environmental impact of clean energy also provides people with additional motives and incentives to move toward adopting the use of solar power.

In more recent years, autonomous vehicles have also become an increasingly popular subject. Many large automobile companies, such as Tesla, Mercedes, and BMW, have developed many semi-autonomous vehicles with hopes to produce a fully autonomous vehicle in the near future. An autonomous vehicle can also be described as self-driving, driverless, robotics, or unmanned due to its ability and design to sense its environment and navigate the terrain without human interference. There have been major breakthroughs in autonomous technology and it has the power to completely change the transportation industry. It is expected that 10 million driverless cars will be integrated in our roadways by 2020. Many prototypes have been developed and tested with safety, cost, and capabilities parameters in mind and with plans to incorporate autonomy in large-scale transportation vehicles, such as trucks, cargo ships, and aviation drones.

Our team is interested in combining these two topical areas of innovation to design and develop a completely autonomous, emission-less, solar-powered beach buggy for the Florida Solar Beach Buggy Challenge. Although our mission is to meet the goals and constraints of the Beach Buggy Challenge, a successfully implemented, solar-powered, autonomous beach buggy could be beneficial for scientists and researchers who need to effectively collect coastal data for time-consuming studies without having to manage or observe the vehicle. This project could also potentially benefit lifeguards or conservationists who wish to keep the beach clean. They can enhance the vehicle to detect and collect trash or litter along the beach and dispose of the material properly. One other marketable use of this project if completed successfully would be to assists consumers in transporting materials, such as tents, chairs, food, etc. and even children on a beach day.

2.2 Project Roles and Task Assignments

This is a multi-disciplinary project and the scope of work is divided amongst the disciplines of mechanical engineering, computer science, and electrical and computer engineering. The mechanical engineering team is composed of 6 undergraduate mechanical engineering students, the computer science team is composed of two undergraduate computer science students, and our team, ECE, is composed of the three of us. Each of the teams will be individually responsible for their field of engineering including researching, designing, and executing ideas for the project. Teams will also have to communicate to the others groups their needs, complications, and design implementations so the project can reach a finished working state. Communication is the absolutely the most important effort in this project. If all the separate teams cannot communicate and converse the overall objective of the project will be a failure. By effectively communicating our problems and solutions there is absolutely no obstacle we can't overcome.

2.2.1 Relevant Experience

Drew Curry, of our team, has incredibly relevant experience that will assist in the design and execution of the project. The experience is outlined below:

“A project that have helped me to prepare for this senior design course has been a project from the IEEE PES, power and energy society, here at UCF. The project is known as the Water Analogy Project and has been underway since the summer of 2017 and will be finished at the end of the spring semester 2018. This project introduced me into the use of microcontrollers completely.

The group that I was involved in for the project had most of its senior members graduating and therefore I was responsible for self-teaching myself about micro controllers. Pushing me to understand firstly what a microcontroller was and how it is operated, opened the door to much of the hardware and software capabilities that were possible with a Raspberry Pi. Furthermore, the project made me research how to operate code that would interface with the GPIO Pinout and PCB circuitry that would correspond to the software. For the duration of the project I was tasked with figuring out how to operate electronic valves, and through this I researched, designed, and programmed hardware and software that would cooperate with our system. The project forced me how to understand and read datasheets of electrical components and compare components with one another. This has given me an initial advantage for senior design allowing me to have some perspective with the direction of the hardware and software components of our current project. Additionally, the project has introduced me to soldering with wires and boards. This gives me much practice when It comes to our project. I will more than likely be soldering much of the hardware that will be implemented in our project. Upon designing, building, and installing our PCB I will be more prepared to solder components to and from the board due to this project.”

Grace Yoo, an undergraduate computer engineering senior, has relevant experience in single board computers and embedded systems obtained through her summer internship experience and current position as a College Work Experience Program (CWEP) student at Lockheed Martin Sensors & Global Sustainment. She has experience using a variety of communication protocols, such as User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Audio Video Bridging (AVB), as well as using communication analyzing tools, such as Wireshark. Grace is also familiar with software version control tools including GitHub and AccuRev. She has practiced developing software on both Windows and Linux environments, using virtual machines, Secure Shell (SSH) connection, MobaXterm, and Cygwin. She has also recently have been working with Google Test, a unit testing library for C++ programming, and Klocwork, a static code analysis tool. Grace also tinkers with microcontrollers and circuits in her spare time and taught herself how to solder and program in C++.

2.3 Objectives

The ultimate objective of this project is to design and develop a fully solar powered beach buggy that will be able to traverse a 10 mile stretch of the beach from Daytona to the Ponce Inlet and return within an 8-hour period without the need for human interference or guidance. The vehicle must be able to complete the excursion completely autonomously through the process of 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 journey along coast. The efficacious superimposition of photovoltaic, autonomic, and mechanic systems is the culmination of the project's objectives.

2.3.1 Solar Power System

Our objective is to design a solar power system capable of powering the vehicle designed by the mechanical engineering students assigned to our project for 10 hours whilst powering DC motors whose size and capabilities will be dictated by the mechanical engineering students as well. This system will involve a solar panel and a battery to store solar energy to be used when there is not sunlight being harvested in the moment to power the motor. The system will be guided through a charge controller to effectively convert as much energy as possible from the solar panels to the system. Further design implementations and complications will be talked about in other related sections of the paper.

2.3.2 Autonomous Navigation

Our objective in terms of autonomous navigation are to design and implement sensors that can detect both stationary objects, moving objects, soft body, and hard body objects. We handle the entire scope of implementing sensors and gathering data about the vehicles environment and are responsible for communicating that data to a programmable resource. We will also work in conjunction with the computer science students to analyze the data and design adequate course navigation.

2.3.3 Engineering Development Process

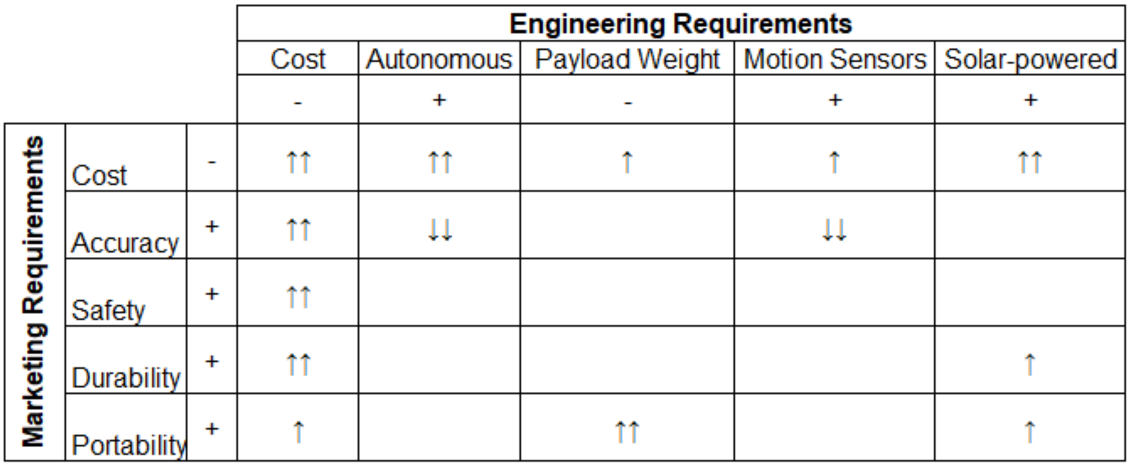
Our objective, in terms of the engineering development process, is to have a better understanding of the process to take a project from start to finish in the industry. We intend to understand the level of communication, documentation, and execution that is expected of engineers by customers. We will break down the project into tenable sections that separate research, design requirements, prototypes, feasibility, and other project related benchmarks.

2.4 Requirement Specifications

- The apparatus must be able to autonomously traverse a 10 mile stretch of beach from Daytona to Ponce Inlet (and return) within 8-hour time span
- The vehicle must be capable of transporting one passenger with a maximum payload of 120 lbs.
- The apparatus can travel no faster than 3 miles per hour
- The apparatus must be completely powered by solar energy
- The apparatus must do no harm to the environment and beachgoers
- The apparatus must be able to detect and avoid both stationary and moving obstacles (e.g., rocks, docks, people, birds, turtles, etc.)
- The apparatus must also be able to stay out of the ocean and keep from going into the non—beach parts of the environment (e.g., hotel resorts, grassy areas, etc.).
- The apparatus must effectively collect and convert solar energy into electricity
- The system should be designed to efficiently distribute power to the vehicle's motors, sensors, and electrical components as well as use very low amount of power when the vehicle is idle
- The apparatus should be big enough to carry one passenger, yet small enough to transport easily without the need for heavy duty machinery
- The speed of the vehicle should be monitored and throttled at 3 miles per hour, if the system senses the vehicle is moving too fast, it will activate the braking system
- Sensors should be actively surveying the environment in order to detect stationary or moving objects within 2 to 3 feet of the vehicle
- When the sensors detect a moving or stationary object, the system should be notified and alerted to either change the course of the vehicle or come to a complete stop.

2.5 House of Quality

The House of Quality table compares engineering requirements and marketing requirements to show the trade-off involved in satisfying both sets. It is an incredibly important decision-making tool that helped to establish what tradeoffs could and would be made when designing the vehicle. The house of quality table for this project is below.



			Engineering Requirements				
			Cost	Autonomous	Payload Weight	Motion Sensors	Solar-powered
			-	+	-	+	+
Marketing Requirements	Cost	-	↑↑	↑↑	↑	↑	↑↑
	Accuracy	+	↑↑	↓↓		↓↓	
	Safety	+	↑↑				
	Durability	+	↑↑				↑
	Portability	+	↑		↑↑		↑

Table 1: House of Quality

↑ = Positive correlation

↑↑ = Strong positive correlation

↓ = Negative correlation

↓↓ = Strong negative correlation

\+ = Increases the requirements

\- = Decreases the requirement

While all of these are requirements dictated by the project, the technical requirements fall into the engineering requirements section whereas the non-technical requirements are displayed in the marketing requirements section. There are engineering and marketing requirements that operate independently of each other. Most electronics will not have any effect on durability.

3.0 Research Related to Project Definition

This section will review all relevant research completed to design and develop our autonomous solar powered beach buggy system. Similar existing projects and products regarding autonomic and solar powered vehicles will be described, as this surrounds the main scope of our project. 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 but are not limited to sensors, GPS, cameras, microcontrollers, photovoltaic cells, batteries, and voltage regulators. Other research areas of interest include autonomous vehicle navigation, imagine processing, possible software middleware, bodies of water detection, and battery storage. This section will also discuss the final components selection for the initial design of the system and describe the observations from a recent site environment visit in which the project will be traversing.

3.1 Existing Similar Projects and Products

Innovation in the solar and autonomic fields is growing exponentially under a very competitive market. With environmental concerns breeching the surface and pressures for futuristic applications surrounding technology today, there are a variety of projects that already use the same idea of harnessing the power of autonomy and clean, renewable solar energy to power vehicles.

3.1.1 Fully Autonomous Vehicles on the Market

While fully autonomous vehicles are scarce in the market, as technology and research advances, they are beginning to play a more significant role in the commercial market. The concept of a fully autonomous vehicle hitting the market poses several safety issues that have yet to be resolved with the confidence required to roll them out. Several accidents have occurred in the testing of fully autonomous vehicles in public. Because of this, the vehicles on the market that boast being autonomous are actually not fully autonomous but have autonomous features.

Public transportation is one of the largest focus areas as municipalities aim to improve access to transportation in their cities, reduce traffic and accidents, and cut down on costs.

Additionally, research is being done incorporate autonomous vehicles in the delivery industry. Automated vehicles already exist on the consumer market. The lead in this industry is Tesla whose vehicles are equipped with “enhanced autopilot.”

3.1.2 Solar Vehicles on the Market

Just as prevalent in the industry as autonomous vehicles, if not more so, are environmentally conscious vehicles. The world is desperate to limit its impacts on the environment as research is exposing the damages that our day to day transportation habits are having on the earth. Due to feasibility of production and costs, the main focus is on electric vehicles (EVs), though advances have been made with solar powered vehicles.

Most notably has been the Stella Lux, which was built by a group of students studying at Eindhoven University of Technology. This solar vehicle has a maximum reported range of 300 km before resorting to stored battery power. The students worked to create a car that minimized energy consumption in order to maximize the use of the amount of energy that can be feasibly produced by solar panels. Resistance was a main factor in accomplishing this feat- air resistance and mechanical resistance. It is clear to see that extra consideration regarding these topics would need to be taken to account for the environment type of our project. The weight of the vehicle was also a huge factor in the design. The mass of the entire body is only 75 kg. Additionally, the solar roof was designed in such a way to maximize the solar yield. The resources available to the students at Eindhoven University were far more extensive than those available for this project. The scope was also much larger. [33]

Figure 1: Stella Lux Roof

Permission to use in progress. [33]

3.1.3 Autonomous Solar Vehicles

The idea of combining these two market products has also been produced, though not on a commercial level. Projects relating to autonomous solar vehicles has been performed by companies and students across the world. Most of the projects highlight the potential advantages these vehicles can offer and how they are implemented on small scales.

The innovator, engineer, and genius Elon Musk has shown interest in autonomous solar powered vehicles over the past ten years [4]. In a blog post by Elon Musk posted on the Tesla website titled "Master Plan, Part Deux", Musk talks about the future of Tesla and its interest in solar powered autonomous vehicles [4]. The article talks about delivering an engineering design process that will gradually reduce cost and improve efficiency of solar powered autonomous vehicles over the next ten years [4].

As a leader in innovation it is clear that Tesla, and by extension Elon Musk, want to provide this type of technology into the global market and the realm of practical engineering developments. As they continue their design process they will eventually achieve their goals and autonomous solar vehicles will be a regulatory good.

3.1.4 Autonomous Beach Buggy

This autonomous beach buggy challenge clearly reflects the emerging technologies of the 21st century. This project will expose all of the group members to solar power and autonomy which both arguably are driving many projects in all professions today. This Autonomous Beach Buggy project is an excellent opportunity and more importantly a stepping stone for future projects in this field of engineering and science.

3.2 Research of Relevant Technologies

This overreaching section will cover all the potential equipment and final equipment decisions for the project. The research will does not include all of the equipment that we discovered throughout our research but consist of the realistic possibilities that we considered using. This section will draw conclusion for the reasons why we chose these devices and equipment but will not talk about how we are explicitly implementing them. Device and equipment implementation will be discussed in another section.

3.2.1.3 Ultrasonic Sensors

In this section we will talk about the ultrasonic sensors that we will be considering for longer ranged detection. This includes object and people detection. Parried with our camera sensor and GPS, we will rely on this technology for navigation. The ultrasonic sensors compared feature different detection ranges, and detection angles specified by their beam pattern on their datasheets. We will only be referencing XL-MaxSonar sensors as there is such a diverse pool of ultrasonic sensors, we can achieve our requirements based on their library of sensors. The two sensor families can be divided into the 1XXX series and the 7XXX series. Each family series contains almost 50 different types of sensors independent of one another allowing for much cross comparison of sensory equipment. Additionally, the price ranges for the XL-MaxSonar sensors are reasonable compared to other sensors that claim to provide similar specs. This helps us to remain in budget and have additional funding for testing.

MB1XXX series – Indoor, wide beam angle, 10.5m

The MB1XXX XL-MaxSonar EZ series sensors are very similar indoor sensors that we are considering for close range proximity sensing. Coming from the same family they share very similar properties that will be lumped together, and their differences will be recognized when needed. All of these sensors feature a sensing distance of either 7.5 meters or 10.5 meters and almost all differ in their beam angle [8]. For example, the 1260 features the largest angle detection region and the longest-range detection region, where as the 1240 has a 7.5-meter range and the narrowest detection region [8]. Now although the 1260 features an overall larger area, the 1240 can be more specific and is tuned to detect specific things. All the sensors have features to detect specifically soft bodies, hard bodies, ignore clutter information, report the first thing detected, report all things detected. Some are aimed at providing information that appears directly in a straight line in front of the sensor, and others aim at detecting all objects in front. Both of these sensors could provide useful to the project and further discussion will determine which will be best to use. Our team will be deferring to the datasheet of the family to compare the sensors and draw conclusions.

MB7XXX series – Outdoor, low beam angle, environment sensors 10.5m

The MB 7XXX XL-MaxSonar WR/WRC series ultrasonic sensors are the outdoor equivalent of the MB1XXX sensors. Most of these sensors have differences that make them optimal for different detecting objectives, but still share some similarities. All of these sensors feature a detection range up to 7.65 meters or 10.68 meters [9]. Almost all of these sensors have very tight beam patterns that seem provide almost no peripheral vision [9]. Upon testing of a purchased sensor, we will determine the horizontal visibility and effectiveness. Each sensor comes with recommendations for which environment it would best be suited for. For example, MaxSonar recommends the MB7051 for small or soft target detection and is great for detecting people [9]. While the MB7052 is said to ignore small targets and only report the range of the largest acoustic return [9]. It is also said to ignore clutter such as rain, snow, electrical noise, and other acoustic noise. Lastly the MB7062 is said to have a 3 reading stability filter in the firmware [9]. The sensor is suited for applications requiring very accurate readings and ranges the first detectable target [9]. Obviously, this family of outdoor ultrasonic sensors has very a versatile selection and upon further discussion and possible testing the optimal sensor(s) will be selected for the project. Below is a table comparing the MB1XXX series and the MB7XXX series in just a few categories that relate to the project.

	MB1XXX Series	**MB7XXX series**
Environment	Indoor	Outdoor
Range	7.5 meters ~ 10.5 meters	7.5 meters ~ 10.5 meters
Beam Angle	Wide/Narrow	Wide/Narrow
Cost	$50 price range +/- 20%	$100 price range +/- 20%
Number of Parts	43,103	43,102
Voltage input	3.3V ~ 5.5V	3V ~ 5.5V
Power Consumption	3.4mA per	3.4mA per

Table 2: Comparison of sensors

HC-SR04 Sensor

The HC-SR04 is a cost-effective ultrasonic sensor that was immediately considered for use to detect close-range objects. The cost-effectiveness of the sensor allowed us to test it to assess the feasibility of this component for close range. For our test, the HC-SR04 was connected to an Arduino board and the data read from the HC-SR04 was displayed on the Arduino IDE's serial plotter. The following results show a partial of our output graph representing distance over time. Dividing the left column by 100 gives you values in meters and dividing the lower axis by 100 gives you seconds. The seconds category is arbitrary and only displays the time elapsed while the sensor was on.

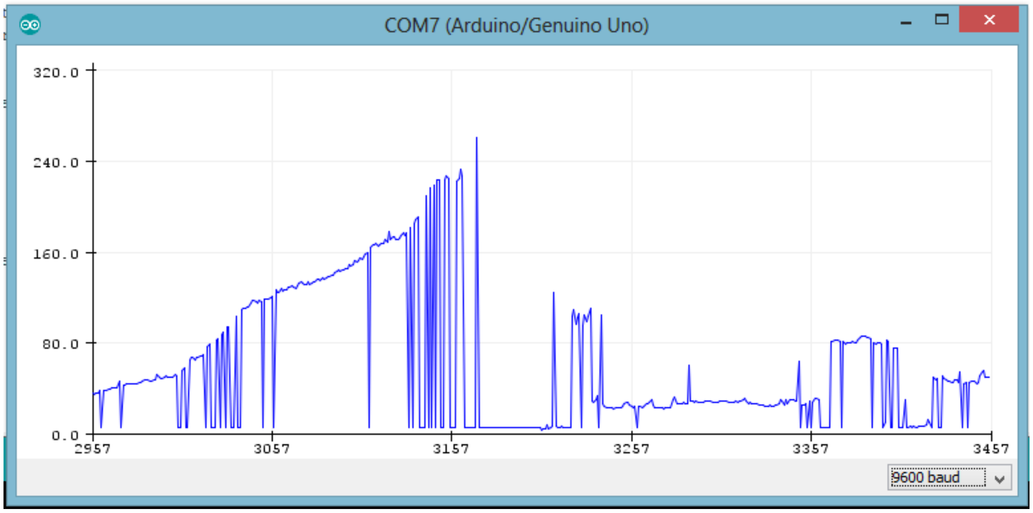


Figure 2: Screenshot of the Arduino Serial Plotter Results.

The device was tested in both indoor and outdoor environments. Though the HC-SR04 is rated for a range of 4 m, we were only able to acquire consistent and reliable readings for obstacles up to 2m away for both indoor and outdoor environments. There were also severe drops and spikes fairly often that did not accurately reflect the distance measured to the object. The dramatic drops can be seen in the serial plotter image results above. Because the overall graph has partial inconstancies scattered throughout its data we've concluded that effective software interpretation will be needed to analyze the data. We are also considering digital to analog converters to help ignore the drops and spikes.

3.2.2 Autonomous Vehicle Navigation

For large scale navigation our team has decided to utilize GPS tracking. Using software paired with a global positioning system we aim to create a virtual boundary for which our project will operate in. The GPS will aid in water avoidance and remaining on the beach. The GPS will help the project avoid the water through indirect means. Upon creating a virtual boundary, we will observe and calculate the ocean tides and maximum water rise to set an initial boundary for the GPS. This boundary will not be exact and may vary along the beach but will more or less act as a protection boundary for the system. GPS implementation will vary with versatility and remain open to interpretation from other sensory equipment to avoid water.

Additionally, sensory input will be a primary tool used for navigation. As the robot proceeds down the beach sensors will aim to change the navigation to avoid people, objects, and hazards. There are many difficulties regarding reliability and effectiveness of the sensory inputs, which we will aim to normalize through excessive testing and extra sensors. We will use inputs from multiple devices to reliably path and navigate the vehicle in order to meet the project requirements.

3.2.3 Image Processing

This section will discuss the various hardware options considered to assist in image processing that will handle object detection, path planning, and navigation. Several aspects of the hardware’s capabilities as they pertain to our needs will be address such as sensitivity of sunlight and frames per second captured.

3.2.3.1 Raspberry Pi Camera Board v2

The Raspberry Pi 3 comes with the Camera Module V2 that can take in information for the Pi. We will be utilizing the camera as a sensor for close range detection of objects and individuals. The camera will be mounted on the front of the project apparatus in anticipation that it will be most useful for navigation there.

With careful testing and unique programming, we will have the camera scanning for changes in color, shadows, and potentially movement. As testing progresses, we will find more uses for the Raspberry Pi camera in hopes that it will provide ample sensory input. If the project budget allows for additional microcontrollers we will consider the possibility of additional cameras for increased input. Depending on the range, and effectiveness of the cameras sensitivity, the camera may be on rotating motor and scan horizontally across the landscape. This would allow for increased vision of the immediate surroundings and further increase the navigation of the project.

The camera itself allows for high definition video and still photographs. As long as computations can keep up with live video feed, we will be using the camera to primarily gather video as a sensory input. We plan to identify color patterns consistent with the beach and aim to identify outliers of these color patterns. This in turn will let us identify obstacles, be it a person, inanimate object or even dangerous terrain that is impassible. Testing the capabilities of the camera is very important and must be done in an environment related to the project.

The camera may act additionally as an emergency stopping device. Being that safety is a top priority, the camera will act as one of the emergency devices. Not allowing for further mobility if there are hazards in the path of the object.

3.2.3.2 Pixy Cam

The Pixy Cam is an all-encompassing vision system with a powerful processor that is capable of being taught how to detect objects in its range of view. The Pixy Cam captures images at a rate of 50 frames per second and has 7 available data outputs (UART serial, SPI, I2C, USB, digital, analog) making it compatible with most microcontrollers. Software libraries are also provided for most popular microcontrollers.

The versatility and compatibility of the Pixy Cam for a multitude of uses and microcontrollers, and the fact that it already includes image processing software, makes it a very attractive choice from the standpoint of electrical engineers. It would allow us the opportunity to focus our efforts on the reliability of the data processing of our other sensors, as well as the harvesting and storage of solar energy. Its technical specifications are as follows:

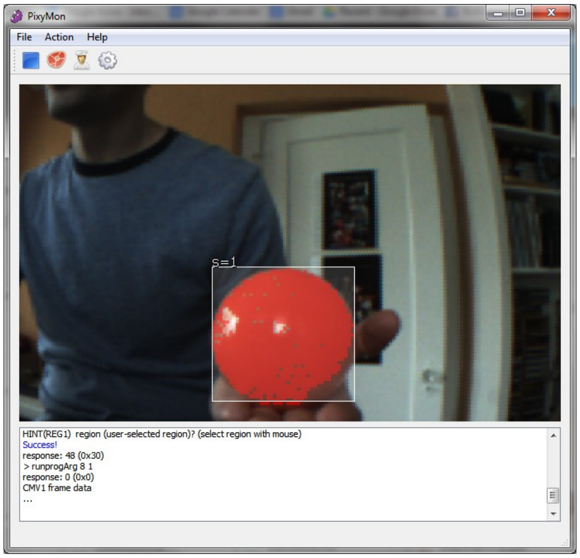
- Processor: NXP LPC4330, 204 MHz, dual core
- Image sensor: Omnivision OV9715, 1/4", 1280x800
- Lens field-of-view: 75 degrees horizontal, 47 degrees vertical
- Lens type: standard M12 (several different types available)
- Power consumption: 140 mA typical
- Power input: USB input (5V) or unregulated input (6V to 10V)

- RAM: 264K bytes
- Flash: 1M bytes
- Available data outputs: UART serial, SPI, I2C, USB, digital, analog
- Dimensions: 2.1" x 2.0" x 1.4
- Weight: 27 grams

For image capture, the Pixy Cam is supported by the Omnivision OV9715 image sensor, which is capable of operating within a temperature range of -30°C to +85°C and incorporates several advanced image processing functions such as exposure control, gain control, white balance, lens correction and defective Pixel correction.

Additionally, the Pixy Cam is able to locate and keep track of hundreds of items at once. It uses a connected components algorithm to determine where one object begins and another ends. It then compiles the sizes and locations of each object and reports them through one of its interfaces.

The price point of the Pixy Cam is around $60-$75, which is very reasonable considering its functionality.



```
HINT(REG1)  region (user-selected region)? (select region with mouse)
Success!
response: 48 (0x30)
> runprogArg 8 1
response: 0 (0x0)
CMV1 frame data
...
```

Figure 3: PixyCam Identifying Red Ball.

Permission to use in progress. [32]

In practice, the Pixy Cam uses a color-based filtering algorithm to detect objects. Color-based filtering methods are popular because they are fast, efficient, and relatively robust. Pixy calculates the color (hue) and saturation of each RGB (red green blue) Pixel from the image sensor and uses these as the primary filtering parameters. The hue of an object remains largely unchanged with changes in lighting and exposure. Changes in lighting and exposure can have a frustrating effect on color filtering algorithms, causing them to break. Pixy's filtering algorithm is robust when it comes to lighting and exposure changes, which is incredibly useful considering the variety of lighting and exposure that will be present in the beach environment [32].

3.2.3.3 Stereo Cameras

Stereo cameras offer a unique form of video capture. A stereoscopic camera allows images to be captured and rendered instantly in 3D. Utilizing a dual lens, a stereo camera can detect the foreground, background, and zero parallax [10]. All of these are important when trying to get correct picture and video. The foreground is the closest element that will appear, and the background is just the opposite, it determines the furthest image the camera will detect [10]. Lastly the zero parallax is like the space between, it resembles the screen and by default lies in between the background and the foreground [10]. We are considering implementing them for the purposes of object detection. The dual camera lens allows up to determine objects and colors on their own without the input of other sensors. We could potentially use this camera as a primary source of detection and then compare our sensor detection to it. They would both act as a reference to one another and let their outputs compare to find the true environment around the vehicle. Stereo cameras trick the human eye by taking the same picture but offsetting it by a factor of a few inches. This not only gives depth but a slight projection of the objects within the picture. Stereo cameras are used in short range detection scenarios in which it needs to determine objects and movement within a ten-foot range. A camera like this could provide a very reasonable and interesting way to object detection.

Figure 4: Example of a Simple Stereo Camera.

Permission to use in progress. [10]

3.2.4 Software

This section will explore the possibility of using a software middleware known as Robot Operating System or ROS. The ROS framework is widely used in automaton and robotics programs all around the world and is gaining more support and recognition over the years. The benefits and drawbacks of using the Python programming language as the basis of our software is also discussed in detail in this section.

3.2.4.1 Robot Operating System

ROS is a moderately new, open source, collaborative collection of tools, libraries, and conventions that is used in a wide variety of robotics projects. From large engineering companies such as Intel, to your everyday hobbyist or juvenile inventor in their garage, ROS is open to be used by anyone at no cost.

ROS is a financially responsible and production convenient option considering it was developed using the permissive BSD open-source license. This license states that the redistribution and use of the code (in source or binary forms), with or without modification, is permitted provided that a few conditions are met (These conditions will be further explained in Appendix B – Copyright Permissions). As long as these terms are resolved, any group can start their own code repository and maintain full ownership and control of their source code which will prevent any potential licensing roadblocks for our software development team.

ROS can be described more as a framework rather than an Operating System. It will simplify our task of software integration. ROS was designed for collaboration, making the job of code assimilation easier and diminishing the chances of collusion. It is easy to integrate with another robot software, such as OpenRAVE or Player, if needed. ROSTest is a built-in unit and integration testing framework that makes the construction and destruction of testing fixtures manageable.

The ROS framework can be implemented in Python, C++, Lisp, or any high-level programming language which allows our programming team to develop software in any language they prefer. ROS supports several different communication protocols, more commonly TCP/IP communication, and communication types, such as synchronous/asynchronous data streaming over topics or RPC-style communication over services. It is possible to incorporate real-time code with ROS, but ROS is not a real-time framework.

ROS is currently only available to run on Unix-based platforms, primarily in Ubuntu (Linux) and Mac OS X systems. Other Linux environments, such as Fedora, Gentoo, and Arch Linux, also have been successfully supported and integrated by the ROS community. A Microsoft Windows version has been possible, but it has not been fully developed or verified.

Another reason that steered our team to using ROS was its ability to support a plethora of sensors. ROS supports an assortment of 1D, 2D and 3D ranger finders, as well as a variety of cameras, audio/speech recognition software, pose estimation (GPS or IMU) software, and sensor interfaces. A more complete listing of ROS' supported devices, drivers, and interfaces will be provided in a source cited Appendix A [16].

We intended on using multiple Passive Infrared Radar (PIR) sensors, ultrasonic sensors, and a camera to detect and avoid objects and people while navigating the beach coast with assistants of a Global Positioning System (GPS). ROS not only with the tools to implement such a system, but also provides plenty of tutorials and documentations of features sensors, and common, stable, usable interfaces. Manipulating raw sensor data into usable information is one of the challenges of this project and ROS renders this objective more attainable.

The Robot Operating System is the ideal foundation for our software development. Not only is there an abundance of tools, documentation, tutorials, and support that will help us learn, develop, and troubleshoot our software; the framework is free, open-sourced, and internationally implemented across the world.

3.2.4.2 Python Programming Language

Python is a prevalent object-oriented, embeddable, high-level language used by many users for a range of applications, such as for academia, web development, GUI development, etc. The readability of Python is largely discussed a prominent advantage for software development. It is argued that this simplistic legibility and extensive libraries promotes a constructive workspace for programmers and aids with maintaining and enhancing their programs [17]. The clean, simple, and straightforward syntax is a result of Python's less verbose characteristic. Unlike other high-level languages like Java or C++, Python is more flexible and dynamic; it requires less code to develop a program in Python than to write the same program in another language.

To support its simplicity, Python downloads with a large, extensive, standard library that covers a myriad of uses and functionalities. Some include, but are not limited to, libraries that aid for document-generation, unit-testing, threading, databases, CGI, and image manipulation [17].

Python is also very extensible, meaning that it can be extended to incorporating other languages [17]. A software developer is not limited to only using Python in a project, they can integrate C, C++, or Java into their program if they deemed it necessary. Or they can integrate their Python program with other projects that are not written in Python. This flexibility aids to generate Python to more the Internet of Things (IOT) opportunities.

The Internet of Things is the concept that any device with an on/off switch can be connected to the Internet and to other devices. More specifically, a system of interrelated devices, either mechanical or digital, may be interlaced to transfer data over a network without the need for human interference [20]. This concept can be extremely useful for our team as we create a reliable autonomous system of communication between solar panels, power distribution device, motors, servos, Raspberry Pi or Arduino Uno microcontrollers, and an array of different sensors.

Python is also a portable language, meaning that you can run it on any machine after ensuring there is no operating system dependent features in the program. A programmer can interchange their program from a Windows workstation, to MAC or Linux workstation without any complications. This will be valuable perk for our team as our members work and develop software in the environment they deem fit per their programming experience.

Other programming languages, such as C++ or Java, need to be compiled before it can be executed, but this is not the case for Python. Python is described as interpreted, which means it is executed line by line rather than all at once [17]. This attribute of Python can be both beneficial and disadvantageous. This incremental execution can help with debugging and unit testing, but it can also result in slow execution time. Although this may be the case, it is well worth the tradeoff if computation speed can be spared.

There are other potential drawbacks of Python, such as having an underdeveloped database access layers, weak mobile computing and browsers presences, absence of commercial support, and restrictions in designs, but overall, for this project, Python is well-suited for our software objectives and goals [18]. The programming language is dynamic, embeddable, free and open-source, simple, portable, and works well with our intended framework ROS (Robot Operating System).

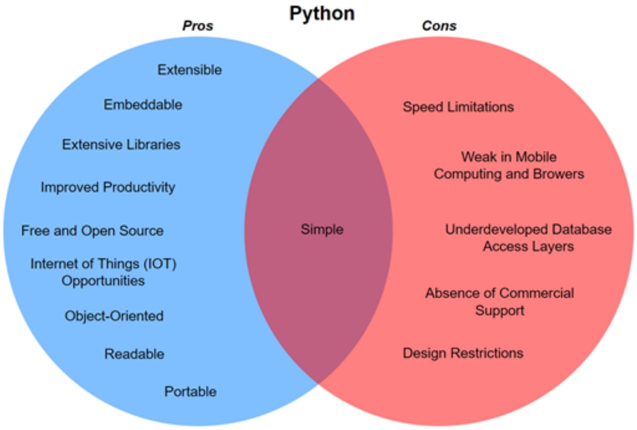


Figure 5: Python - Pros and Cons Venn Diagram

This research highlights Python as a strong contender to be selected as the sole programming language used to develop software for our project.

3.2.5 Microcontrollers

This section will discuss the information researched on microcontrollers, specifically the Raspberry Pi 3 Model B and the Arduino Uno. The microcontrollers' functionality, memory size, processing power, pin layout, compatibility, and cost will be described. A thorough overview of the microcontrollers is imperative during the selection process. The decision should keep processing power and relative cost in mind as it is necessary when exploring the potential roles and possibilities the hardware can contribute to the system.

3.2.5.1 Raspberry Pi 3 Model B

A Raspberry Pi microcontroller is a compact, credit-card sized, single-board computer that was originally designed to help educate individuals on both hardware and software components at a low-cost [26]. Over the years, Raspberry Pis have become faster, more efficient, and powerful without an inflation in price. Our team is planning to use a Raspberry Pi 3 Model B to direct and maintain our system of sensors and power supply for auto mobility.

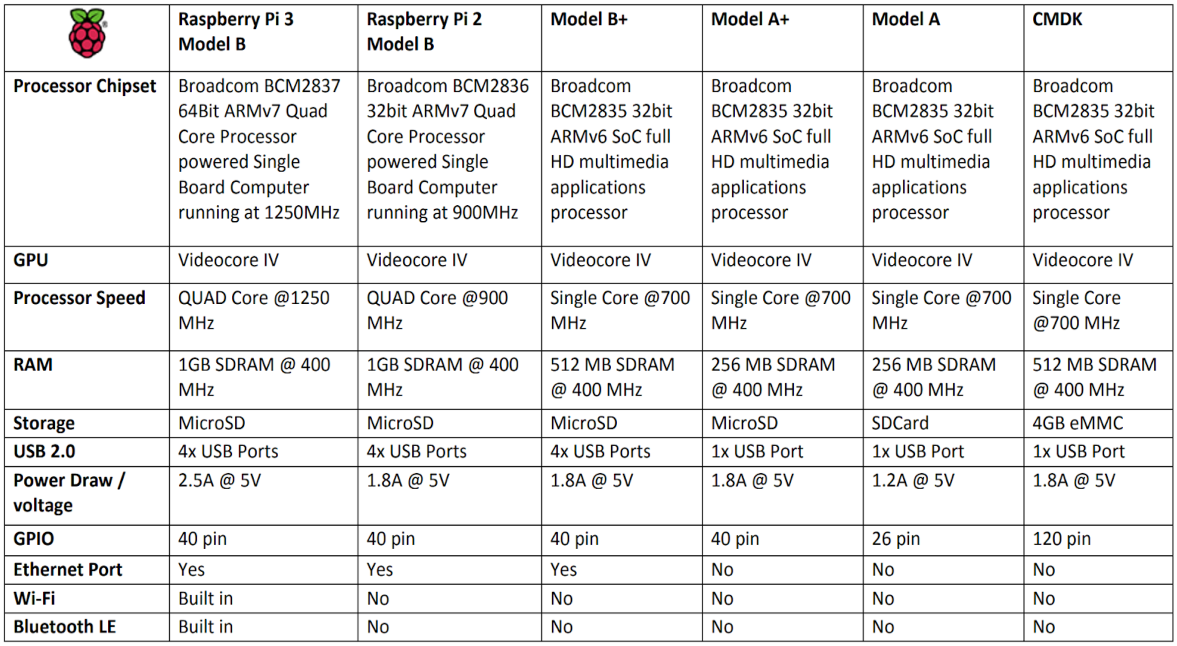
The Raspberry Pi was designed for a Linux operating system, the more popular options being Raspbian and Pidora [24]. Raspbian is based on the Debian operating system while Pidora is based on Fedora operating system. Although these operating systems are the most popular, the Raspberry Pi may run on other Linux-based operating systems, such as Ubuntu, Retro Pi, etc. The Raspberry Pi can be programmed using C, C++, Java, Scratch, Ruby, and Python; there languages come installed by default.

The Raspberry Pi 3 has two major upgrades compared previous generations of Pi's that our team may find useful; the upgrades include faster processor and a built-in Wi-Fi chip [25]. The Raspberry Pi 3 Model B has a Quad Core Broadcom BCM 2837 64-bit ARMv8 processor, improving the processor speed from 900 MHz (Pi 2) to 1.2 GHz (Pi 3) which will allow for more effective communication and response times. Additionally, the Pi 3 has a BCM43438 Wi-Fi chip built-in the board, making it Wi-Fi ready without the need for Wi-Fi adapters [25].

The Pi 3 is also open to Internet of Things (IoT) opportunities. This concept was briefly explained the Python section of this document, but in short, it is the concept that states any device with an on/off switch can be connected to the Internet and to a system of other devices [21]. This concept can be applied using the Pi 3's Bluetooth Low Energy (BLE) component on the broad.

The switched power source was also upgraded to supply 2.5 Amps instead of 2.0 Amps, which opens our design to power more devices over USB ports if necessary. Even with these upgrades and only a slight change in LEDs location, the Raspberry Pi 3 is backwards-compatible and was able to maintain its predecessors size, shape, connectors, and more importantly cost.

Our team decided to use the Raspberry Pi 3 Model B because of its economically priced cost, computation power and speed, modularity, Wi-Fi and Bluetooth capabilities, pocket-size shape, and availability. The Pi 3 can also be programmed using Python and be integrated with a system of sensors while using the Robot Operating System (ROS) framework.



	Raspberry Pi 3 Model B	**Raspberry Pi 2 Model B**	**Model B+**	**Model A+**	**Model A**	**CMDK**
Processor Chipset	Broadcom BCM2837 64Bit ARMv7 Quad Core Processor powered Single Board Computer running at 1250MHz	Broadcom BCM2836 32bit ARMv7 Quad Core Processor powered Single Board Computer running at 900MHz	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor
GPU	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Processor Speed	QUAD Core @1250 MHz	QUAD Core @900 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz
RAM	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	MicroSD	MicroSD	MicroSD	MicroSD	SDCard	4GB eMMC
USB 2.0	4x USB Ports	4x USB Ports	4x USB Ports	1x USB Port	1x USB Port	1x USB Port
Power Draw / voltage	2.5A @ 5V	1.8A @ 5V	1.8A @ 5V	1.8A @ 5V	1.2A @ 5V	1.8A @ 5V
GPIO	40 pin	40 pin	40 pin	40 pin	26 pin	120 pin
Ethernet Port	Yes	Yes	Yes	No	No	No
Wi-Fi	Built in	No	No	No	No	No
Bluetooth LE	Built in	No	No	No	No	No

Table 3: Raspberry Pi comparisons
Permission to use in progress. [25]

The Raspberry Pi 3 Model B specifications are summarized below. [23]

- Quad Core 1.2GHz Broadcom BCM2837 64-Bit CPU
- 1GB RAM
- BCM43438 Wireless LAN and Bluetooth Low Energy (BLE) on Board
- 40-Pin Extended GPIO
- 4 USB 2 Ports
- 4 Pole Stereo Output and Composite Video Port
- Full size HDMI
- CSI Camera Port for connecting a Raspberry Pi Camera
- DSI Display Port for connecting a Raspberry Pi Touchscreen Display

- Micro SD Port for loading your operating system and storing data
- Upgraded switched Micro USB Power Source up to 2.5 A

3.2.5.2 Arduino Uno

Arduino Uno is an open-source, electronic microcontroller supported by the ATmega328P microchip, which can be programmed using the Arduino Software IDE 1.0. This prototyping board is a great beginner board to learn both hardware and software implementation. The board has a USB connector, 14 digital input/output Pins, 6 analog inputs, a 16MHz quartz crystal oscillator, a power port, sending/receiving LEDs, and a reset switch.

The Arduino Uno can be powered in two ways, via the USB connection or with an external power supply, and the power source is automatically selected when either of the options is used. The Arduino board can also be powered through an AC-to-DC adapter for battery. If the USB-connector is not being used for power, the external power supply leads must be inserted into the GND and Vin Pin headers of the power connector. The recommended range of supplied power is 7-12 volts to avoid unstable power delivery or overheating/board damage. There are four power Pins located on the Arduino Uno, Vin, 5V, 3V3, and GND. More details on these Pins can be found on the Arduino datasheet [28].

The ATmega328 has 32 KB of flash memory for storing code with 0.5 KB reserved for the bootloader. The bootloader allows the programmer to directly upload a new Arduino program onto the board with ease. It also has 2 KB of Static Random Access Memory (SRAM) and 1 KB of Electrically Erasable Programmable Read-Only Memory (EEPROM). The EEPROM can be easily read and written to with the EEPROM library provided in the Arduino Software IDE [28].

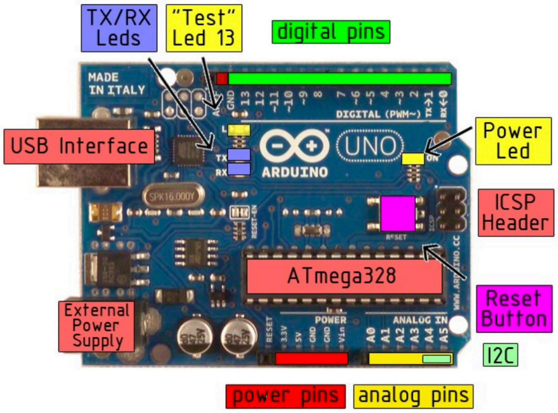


Figure 6: Arduino Uno Pinout and Various Board Information
Permission to use in progress. [28]

The Arduino Software IDE has a myriad of default functions that can be used to read and write to/from input and output Pins. Some functions include, but are not limited to, pinMode(), digitalWrite(), digitalRead(), attachInterrupt(), analogWrite(), etc. These embedded functions programming and communication more accessible to the programmer without the extensive knowledge of electrical circuitry [28].

The Arduino Uno can be integrated to communicate with a computer, another Arduino, or other microcontrollers using ATmega328 UART TLL (5V) serial communication, which is available using Pins 0 (RX) and 1 (TX). The broad also has Pins to support Serial Peripheral Interface (SPI) communication but this is not currently included in the Arduino language [28]. Many parts of the Arduino are interchangeable if they are damaged or defective at a reasonably affordable cost. The Arduino Uno can open our autonomous communication system to a variety of different sensors and will act as an effective communication point between the sensors and the Raspberry Pi.

The Arduino Uno specifications are summarized below. [27]

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) 0.5 KB for bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13

3.2.6 Research on Detecting Bodies of Water

A significant challenge of this project is ensuring that our vehicle can detect the ocean and keep from driving into it. Water damage could be incredibly detrimental not only to the frame of the vehicle but also to the electronics of the vehicle. Though significant considerations will be taken to ensure that components are water tight, issues will absolutely come to play if the vehicle is too critically submerged that it would be unable to navigate out of it. It is also not practical to rely solely on water proofing or tires that are able to handle these situations. Therefore, we must include this problem when selecting sensors for our vehicle.

As it stands, there are no sensors on the market that can reliably acknowledge a body of water. Additionally, when working on a beach near the ocean, it is difficult to rely on GPS solely to ensure that the vehicle will not drive into the ocean.

A study was performed by Jet Propulsion Laboratory in Pasadena, CA on daytime water detection by fusing multiple cues for autonomous off-road navigation. Water cues from color, texture, and range reflection were combined in order to detect bodies of water without relying on GPS. Acquiring water cues from color involved identifying the different factors that go into the color of a water's surface. Ultimately, it was their goal to focus on detecting only the sky reflections in water. They selected a representative subset of 7 images containing sky reflections in water in order to establish hue, saturation, and brightness thresholds. Using these thresholds, they were able to identify the area of a large puddle that was reflecting the sky. The water cue from texture targets regions of the image that have low texture- it is indicative of the features of a water surface. Analyzing range reflections for water cues involved performing stereo ranging on the images to detect reflections. Water is a reflective surface and typically, in the natural environment, would be the only reflective surface. See figures 8 and 9. The blue indicates 1 cue was used, the magenta indicates 2 cues were used, and the red indicates that all three cues were used.

Figure 7: Original Image of Puddle.
Permission to use granted by Larry Mathies, NASA [34]

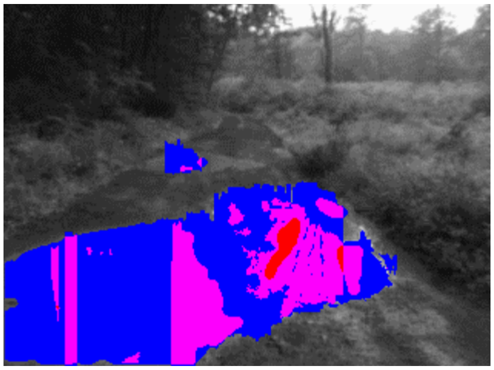


Figure 8: Image with Cues Overlaid.
Permission to use granted by Larry Mathies, NASA [34]

This research is incredibly informative in regard to how to overcome the issue of avoiding the ocean. It would be too difficult to implement, especially since all research on the topic applies to relatively still bodies of water which the ocean is not.

Another means of detecting the ocean that we considered were ultrasonic sensors. Ultrasonic sensors are often used for water detection, but in very controlled settings. Its typical uses include measuring water levels inside tank and other types of containers. This is a practical use for such a sensor because there are no other objects to worry about- the only possible thing for the ultrasonic to detect is the water. [34]

3.2.7 Battery Storage

The following two subsections independently verse positive reasons, with some comparison, for why the type of battery specified would be optimal for the project. The two types of batteries listed are lithium ion and lead acid. Overall a lithium ion battery is more effective and versatile, but due to budgeting reasons a lead acid battery is more likely to be the power supply. Upon additional funding or budgeting of materials, a lithium ion battery would be recommended for the project. Below is a table[1] representing a comparison between advantages of each battery type.

	Cost	**Usability**	**Maintenance**	**Charging**	**Energy Waste**
Lithium Ion	4	9	8	8	9
Lead Acid	8	6	5	5	4

Table 4: Battery Comparisons

The table above compares the Lithium ion battery and Lead acid battery. Higher score is better.

3.2.7.1 Battery – Lithium Ion

The battery chosen for the project has to meet as many requirements as possible and needs to coincide with the 250W solar panels. Due to the large amount of power required for the duration of the presentation and the availability of lithium ion batteries, we have decided to use a lithium ion battery for the power source. Although Nickel-Metal Hydride (Ni-MH) and Nickel-Cadmium (Ni-CD) batteries are similar to lithium ion with potentially lower prices, normal vendors of these batteries do not supply batteries with enough amp hours for our project at a reliable degree. We have also considered using a lead acid battery as our power supply to remain closer to our budget, but because of its lifetime, multitask limitations, recharge ability, and other extended factors, lead acid batteries can potentially reach a delivery limit and not supply ample power for an extended time period. Overall, we are considering lithium ion batteries that can supply between 20-amp hour and 100-amp hour. This range is variable due to our budget potential and power supply potential from the solar panel. If our budget doesn't allow for our low-end lithium ion battery, then we will reconsider the lead-acid battery and attempt to overcome complications and limitations that it may cause.

To further our decision, lithium ion batteries are first and foremost rechargeable. It is an absolute priority that our power supply be rechargeable such that our solar panels can be as productive as possible, and the power supply upkeep remains as maintenance free as possible. Additionally, this helps to avoid future costs and budget variability. Lithium ion batteries also have a large usability capacity that allows for much of the batteries supply to be used without damaging the total usability of the battery [1]. Lithium ion batteries have an overall low "memory effect" that allows for recharging to occur effectively [1]. In addition, they see little overall voltage drop off when much of their storage capacity has been used. This means they will output a steady stream of voltage at almost all charged states. Lithium ion batteries have fast and efficient charging and if there is a failure to completely recharge the battery, the damage to the battery is little to non-existent. This allows for the solar panels to freely charge the lithium ion battery in down time during any stage of its capacity. Lithium ion batteries are very energy efficient and can charge at high efficiency rates as compared to lead acid batteries that lost efficiency as they age.

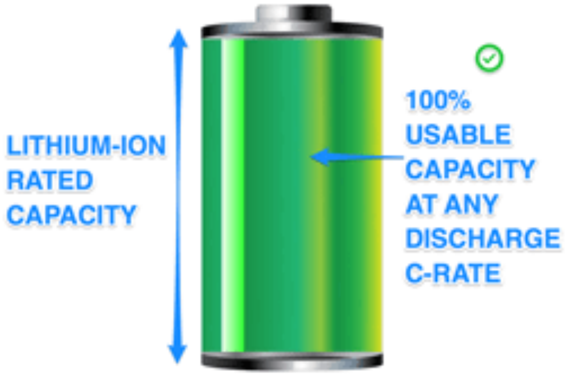


Figure 9: Lithium Ion Battery Capacity
Permission to use in progress.

Although lithium ion batteries are rather expensive, the popularity of them provides us with more choices regarding our amp hour specification. Although we would like to potentially have a battery that can complete the project without any solar aid, as to ensure it would reach its finish line in the case of overcast conditions, we have to remain reasonable and stay within budget. Because of this, we are considering and making calculations towards how much power our solar panels will be able to reliably provide. This in turn takes off power requirements from our battery supply.

3.2.7.2 Battery – Lead Acid

The lead acid battery is the second type of battery we have considered for this project. The lead acid battery can provide much of the same initial conditions that a lithium ion battery can provide, and this initially made the lead acid battery another potential candidate for the power supply. The number one reason we have been considering the lead acid battery is the price factor. Lead acid batteries provide some of the best price per amp hour for our needs. Recharging our power supply using solar panels can not only limit the charge time but the weather can limit a charging opportunity. Our project would benefit from the largest amp hour battery we can realistically use as to keep our charge time and opportunities at a minimum. Furthermore, lead acid batteries have a high tolerance to overcharging and considering the inconsistency of solar panel charge rates, this will allow our charge controller not only leeway in its performance but possibly lower levels of charge states for the lead acid battery [2]. Lead acid batteries that we would consider would be within the 100-amp hour to 200-amp hour range. A larger battery helps to prevent as many obstacles as possible when dealing with power.

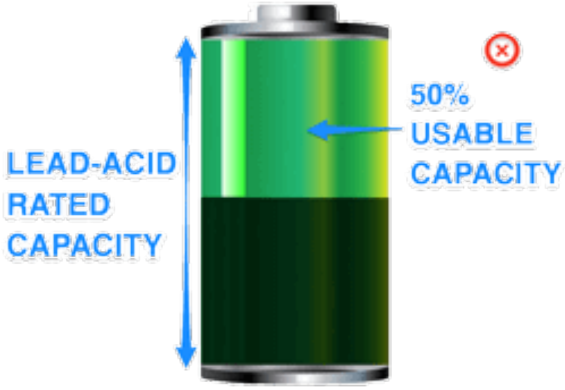


Figure 10: Lead Acid Battery Capacity
Permission to use in progress.

On the contrary lead acid batteries suffer a number of downsides when it comes to continuity, consistency, and prolonged reliability. Lead acid batteries suffer from a large effective capacity usability [1]. Lead acid batteries can start to see significant decreases in their total potential storage capability if their total charge is below 50%. Once lead acid batteries have their total storage capabilities limited the longevity of the battery is unpredictable [2]. This process takes a number of charge and discharge cycles within the battery. If we inevitably use a lead acid battery, we will more than likely not see much decrease in battery effectiveness over the duration of our project. But after a few hundred cycles of the battery there will be noticeable downsides and power supply reduction [1].

Overall lead acid batteries will represent a sufficient power supply but not a reliable power supply as compared to lithium ion batteries. In the end we will more than likely use a lead acid battery as a temporary power source for completion of the project. If the project were to have more budgeting lithium ion battery would see greater benefits than a lead acid battery.

3.2.8 Voltage Regulator

This section will be talking about the various voltage regulators we have been considering for the duration of the project. We will also talk about the differences between linear and switching regulators and the costs and benefits of both. Because there are multiple ways of interpreting the design specifications of the project, we will be considering the multiple ways to implement batteries with our voltage regulators and the multiple organizations of regulators based on our supply requirements. Lastly, we will talk about the implementation of these voltage regulators on a PCB which will be attached to the main PCB.

3.2.9.1 Linear and Switching Regulators

All electronic devices require a specific voltage to operate at and unfortunately it is neither feasible nor realistic to create batteries for each and every part in a project. This is where voltage regulators come in handy. Voltage regulators offer a steady DC voltage output that is independent of load current, temperature, and AC voltage variations. A voltage regulator automatically outputs a steady DC voltage value by utilizing feedback techniques to step down (buck) or step up (boost) the input voltage.

The two types of voltage regulators are Linear and Switching. Linear regulators only offer a buck regulation to step down the voltage and they use linear techniques to regulate the voltage and limit the entirety of the circuit to maintain the desired voltage output [7]. This type of limitation not only wastes energy but also produces excess heat, two things which we cannot ignore for this project. Additionally, linear regulators require a voltage that has to be some minimum amount higher than the output voltage. This voltage is called the dropout voltage. Most linear regulators are used for low power design circuits, so they don’t waste too much overall power because of their limiting factors. These types of regulators could be implemented into our project on small scales where we could build specific regulators tailored to the absolute needs of our sensors and peripherals only. When working with a budget and needed every we cannot consider situations such as these. We won't be using any linear voltage regulators in the project because most of our devices operate in larger voltage requirements where switching regulators would benefit.

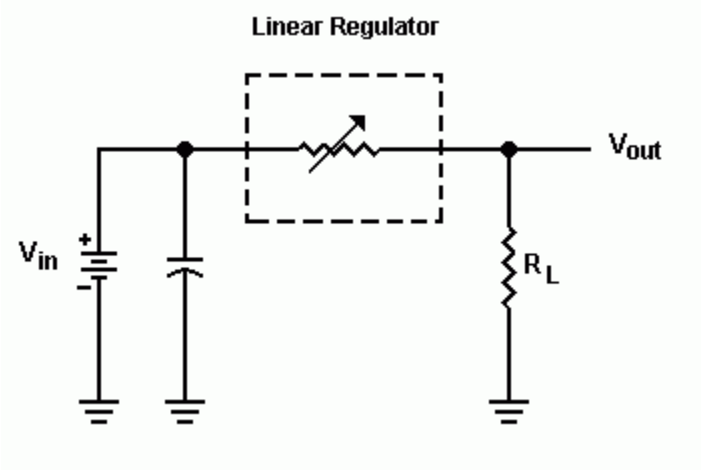


Figure 11: Basic Linear Regulator
Permission to use in progress.

Switching regulators are the far more popular and efficient voltage regulator of which we will be utilizing for our project. Firstly, switching regulators offer both the buck and boost configurations. Switching regulators operate on less power than linear regulators. They achieve this by rapidly switching on and off their storage elements allowing for charging and discharging of capacitors and inductors [7]. By using synchronous and non-synchronous switches (FETs) these input elements can store energy that can be become output voltage at different voltage levels [7]. Because of this continuous switching action and the high conductance of the switching elements don't allow for much power loss [7]. Resulting in an overall high-power efficiency system. Switching regulators can be used in low and high voltage circuits making them ideal for our project which requires at least two stages of voltage regulation. Because of their high-power conservation and buck/boost technology we will only be using switching regulators to step up and step down our power system.

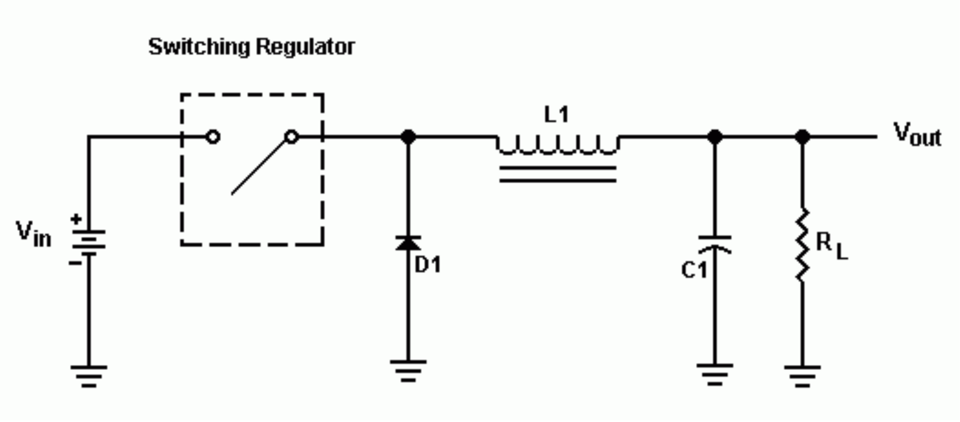


Figure 12: Switching Regulator
Permission to use in progress.

3.3 Strategic Components

The following sections discusses the strategic components selected based on the research performed in sections 3.1 and 3.2. Most of these components are decided upon before testing can occur. Some of these components may function better in different environments. We will be discussing first the applications and then the incorporations of these sensors into the project.

3.3.1 Solar Panel

For this project our sponsor has taken the liberty of already supplying us with solar panels. The solar panels provided are Suntech© solar panels, with model number: STP235-20/Wd. The picture below provides a screenshot of our solar panels. The solar panel itself is quite large in size and weight and provides 235W at maximum usage. But based on budgeting possibilities that may vary toward the later stages of this project we may be able to get an additional 235 solar panel of the same type. The solar panel itself will be mounted on the top of the beach buggy apparatus as to absorb as much sunlight as possible. The solar panel has a maximum current output at 7.79A and a maximum voltage output of 30.2V. With these limitations we will be utilizing a charge controller rated at 10A to ensure we have a strong enough controller to regulate the charging of the 12/24V batteries.

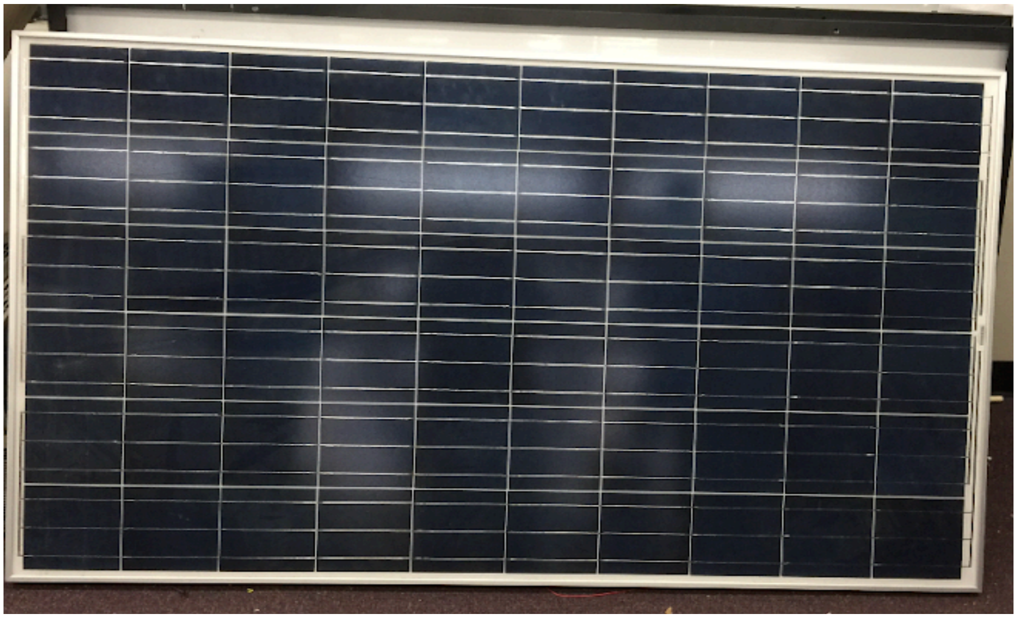


Figure 13: Suntech Solar Panel STP235-20/Wd

Model Number	STP235-2-/Wd
Rated Max Power (P_{MAX})	235W
Output Tolerance	0/+5%
Current at Pmax (I_{MP})	7.79A
Voltage at Pmax (V_{MP})	30.2V
Short-Circuit Current (I_{SC})	8.35A
Open-Circuit Voltage (V_{OC})	37.0V
Nominal Operating Cell Temp. (T_{NOCT})	45°C ± 2°C
Weight	18.2kg
Dimension	1640mm x 992mm x 35mm
Maximum System Voltage	1000V for IEC/ 600V for UL
Maximum Series Fuse Rating	20A

Table 5: Specifications table for Suntech solar panel STP235-20/Wd

A large portion of the energy that our solar panel will actually be able to contribute to our supply is highly dependent on th environment during the day of presentation. It is rather difficult to predict the weather conditions surrounding our presentation day, but we will take into consideration that our solar panel will only be able to contribute a fraction of the total potential power provided. We will consider the solar panel providing down to 30% of its optimal generation, anything below that and the solar panel would be closely producing less than about 20% power for the system. In this case we would claim the presentation day or testing day to be below operational. At this point we would reschedule for a day with a higher operational percentage. We will realistically assume that the solar panel will be able to provide about 70% on a sunny day, and up to 90%-100% during the middle of the day on a sunny day.

3.3.2 Sensors

Practical sensors that we will be using for our project include PIR (passive infrared sensor), ultrasonic, and camera. We decided to use these sensors due to a number of important reasons. Firstly, we considered budget for our sensors to instantly reject things that were not feasible. A LIDAR sensor and system was almost the first sensor we had to reject. Even the cheapest LIDAR sensors range in the $350 USD price range which, even though is within our budget, doesn’t provide ample data for a motion-based time varying system of this caliber.

Seeing as a LIDAR system would be the most effective sensor we could potentially obtain, we gathered that an array of PIR, IR, ultrasonic, radar, camera, and temperature sensors would give us the most information for mapping the area and scanning for arbitrary obstacles. After this we had to consider current research on the effectiveness of sensors that could work outdoors. This is the largest constraint for all of our sensors on this project, even including the LIDAR.

It was very difficult to find relatable information regarding the effectiveness of all our sensors for outdoor settings. Especially settings regarding Florida beaches. The varying weather patterns in Florida including rain, heat, humidity, mist, UV rays, and cloudiness make it difficult to produce a sensory system that will always work. Because of this we decided that ultrasonic sensors will be the most reliable and feasible sensors for the system. Ultrasonic sensors are least affected by weather patterns as compared to the other sensors we’ve considered.

The ultrasonic sensors will potentially use a sweeping motion to plot angles and distances of object from the apparatus. This data will be fed into our navigation system and used to plot out potential pathing’s down the beach. Even so ultrasonic sensors seem to lose potential range when they are not in optimal environments.

To avoid false data readings, we will buy a sensor that has double the suggested range in anticipation for falloff. Outdoor testing of the ultrasonic sensor will be required as to find the effective range. The possibility of machine learning and implementing computer algorithms is considered when interpreting the data. This is because not all data inputs can be interpreted properly. Because of the randomness of outdoor environments, we have to anticipate abstract data. We are aiming to use rationalization and anticipation when interpreting data, especially for moving objects.

Alongside ultrasonic sensors we will use PIR sensors for close range proximity sensing. These sensors will be utilized for detecting obstacles that the ultrasonic sensors could have possibly missed. Because they are infrared their range will be limited while on the beach to only a few feet, and thus will have only immediate proximity of the apparatus and be used as emergency sensors. As the pathing and navigation system operate the PIR sensors will act as red flags for emergency stopping and re-pathing. Testing of the PIR sensors in outdoor environments lets us determine the optimal range for accurate data. We will plan to have at least four PIR sensors on the beach buggy, one for each side of the vehicle.

After comparing all the PIR sensors our team will proceed with the wide angle PIR sensor. The sensor has the largest angle detection area, an adjustable sensitivity, remains within our power input and output profile, and still remains relatively low price. Once again, we will be using the sensor as a close-range proximity sensor that will cover the immediate area around the vehicle. We will be connecting the 4 sensors to our personal Atmega328 PCB and transmitting the sensor data to the Raspberry Pi 3 microcontroller for further navigation purposes. The PIR sensor will undergo testing to determine the optimal and reliable testing ranges for which we will operate it. These testing procedures are described in the testing portion of this project report. Our group will consider the other two PIR sensors if this wide-angle sensor proves to be unmanageable for the project.

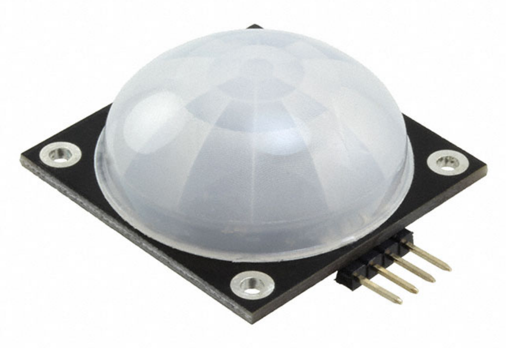


Figure 14: Wide Angle PIR Sensor
Permission to use in progress.

For this project we have decided to use the MB1260 indoor sensor for close proximity sensing on the front of the vehicle and the MB7051 for long range proximity sensing on the front of the vehicle. Both of these sensors have very large detection ranges making their probably reliability sensing range well within our targeted detection range. These two sensors in particular both range out to 10.5 meters and remain adjustable when incorporating into our system. Because the MB7051 sensor is built for outdoor environments it comes equipped with water proof standards and PCB standers for insulation making it great for our outdoor environment. Both of the sensors are rated for hard and soft body detection allowing us to detect not only objects but people. The MB7051 sensor has the option of adding on attachable components that improve the detection sensitivity, allowing us to utilize this if needed. Lastly both sensors feature three types of sensor output including Analog Voltage, Serial, and Pulse Width. These different types of information communicating techniques will allow us to experiment with the most effective and reliable way to transfer information.

Figure 15: MB1260 XL-MaxSonar & MB MB7051 XL-MaxSonar

Permission to use in progress.

3.3.3 Battery

For the project we have decided to use the Duracell ultra 12 v SLA sealed lead acid 50AH gel battery or one of equivalent quality. This battery was chosen due to its reliability related to the company. The battery allows for easy step up and step down to all accessories that we need for the project. The battery will be attached to both the charge controller and our PCB. The PCB will house connection ports that will coordinate power management to the motors and electrical equipment via our voltage regulators we have built and any purchased. This battery only features 50Ah's which poses a challenge to the project. A 50Ah battery will hopefully be enough energy combined with the solar panels but remains the lowest recommended rated battery. Selecting this battery lets us plan our budget accordingly by selecting the cheapest practical battery for the project. If later stages in the project allow our budget room for a stronger battery we will purchase a battery with more amp hours. We may even decide to upgrade the battery from a lead acid to a lithium-based battery. For this reason, we will be purchasing the battery in a later stage of the process to allow for changes.

3.3.4 Charge Controller

Our charge controller will be the device that will moderate and control the actual current and voltage coming from the solar panels. Between the two types of charge controllers, PWM (pulse width modulated) and MPPT (maximum power point tracking) we will also choose a MPPT charge controller. The charge controller has a voltage output rating and current input rating in line with what our batteries and solar panels require.

An MPPT charge controller would like to be chosen over the PWM charge controller. The MPPT charge controller is very versatile when it comes to larger wattage systems. The PWM charge controller operates on par with the MPPT when dealing with systems around the 1-10-watt range. When dealing with system of a larger wattage caliber an MPPT charge controller benefits. The MPPT has the advantage over a PWM because it can regulate its output voltage allowing for more amperage to flow, whereas when the PWM has too much voltage flowing through it simply limits or caps the voltage and wastes the remaining power instead of regulating and increasing the amperage. Overall this allows for more power flow to the system and battery utilizing the solar panel more and getting the maximum energy use that we can. In turn this obviously doesn't waste energy from the battery making us more solar dependent and increasing our efficiency and decreasing our pollution.

Unfortunately, due to budgetary constraints associated with the project we will be selecting a 12v 30A rated PWM charge controller. Overall this charge controller will be able to supply us with power and only lose efficiency in the upper 20% charge capacity of the lead acid battery.

Figure 16: QST 30 Amp PWM Smart Solar Charge Controller

Permission to use in progress.

The charge controller will be connected to both the solar panels and the battery. The charge controller will aim at regulating the solar panel input voltage down to a 12v voltage that we can charge our battery with. The charge controller will not only regulate the voltage but also be in synchronization with the batteries to allow for the solar panels to provide as much power to the circuit as they can. In other words, while the vehicle is not in operation, the charge controller will aid the solar panel to deliver all of its power to the battery. We will ensure that the battery doesn't become overcharged or undercharged for that matter. While the vehicle is in operation the solar panels will primarily be used to operate the circuit and the battery will make up for the rest of the required energy. In the case that we need to regulate the output current on the charge controller to keep the battery from becoming damaged we will set up communication lines that determine the charge rate or the duration of the charge. The complications dealing with the storage capacities and capabilities of the battery create complications that we much avoid ensuring that battery is not damaged.

3.3.5 Microcontrollers

The Raspberry Pi 3 Model B and the ATmega328P was selected to be integrated into our system design. The Raspberry Pi has the adequate processing power to manage imagine processing, computational navigation behavior as well as a GPS system. Accompanying this piece of hardware is the ATmega328P, a high-performance Microchip that can be found on an Arduino Uno. We have chosen to design a custom printed circuit board around this microchip to better suit the needs of our system instead of including a microcontroller with unused additional accessories. The section further explains our selection decision and how we plan to use these microcontrollers in our system.

3.3.5.1 Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B is the ideal “microcontroller” for our system because it is modular and considerably affordable given its computational power and relative speed. The compact board also has built in Wi-Fi and Bluetooth capabilities which is beneficial if our team decides to go with an IoT concept for our system communication. The Pi 3 can also supply power to a variety of devices through its USB port. The operating system on the Pi is also Linux-based which makes it open to the Robot Operating System (ROS) framework, and variety of programming languages, such as Python. The Raspberry Pi can be easy integrated into a communication system and is powerful enough to compute logic for situational responses given a set sensor data. The Raspberry Pi will be used in conjunction with our computer programming team, acting as their primary source for computing navigation.

The Raspberry Pi will be connected to our PCB for power and be connected to the ATmega328 chip for communication purposes from our sensors. The CS team will be using the Pi to coordinate navigational software. The Pi will also house our GPS system and any cameras that are not compatible with the ATmega328. We are choosing to work with a Raspberry Pi because our CS team has concerns that an ATmega328 will not provide sufficient computational power for the entire system. Our team has considered building our own PCB that would include the broadcom chip featured on the Raspberry Pi. Because we intend to use the other features on the Raspberry Pi development board we have been pending our decision on what to do. If our budgeting allows for a printable PCB we will probably create on with the broadcom chip and add the schematic to our PCB collection.

3.3.5.2 ATmega328P

Our team has decided to create a printed circuit board based off of the Arduino Uno microchip, the ATmega328. The high performance and low-cost of the microcontroller proves to be budget-friendly, fairly intuitive to use, and have a wide range of attachable sensors. The printed circuit board will effectively be the Arduino Uno without all of the unneeded components on the developer’s board. Our Printed circuit board will also contain a number of terminal banks to attach all of our sensors. We will be utilizing the analog, digital, and general purpose I/O Pins from the microcontroller to operate our sensors. This PCB schematic will be added onto our main PCB schematic as to eliminate wasted space and save money. The Atmega328 PCB will be connected to our power supply through a voltage regulator and also connected to the sensors through terminal banks on the PCB. The sensors will send unoptimized data to the ATmega328 chip and after some organization will be send via link to the Raspberry Pi for system navigation. The Atmega328 will be handling up to seven or eight sensors at a time and will not be performing to much computations because of this intense load.

3.3.6 Software

Our team plans on using the Robot Operating System for the basis of our software. It is the clear choice when working with robotics and most of its use is open source. ROS is a powerful and versatile open-source middleware framework for developing robotics applications. Its publish-subscribe messaging model, services, diverse libraries and packages, and bustling community have eased the robotics development cycle and made it feasible to develop modularized and versatile robotics software.

ROSs open-source nature also makes it ideal for incorporating into any robotics project. The various libraries and packages can be altered to suit the needs for any given project. Research into the potential drawbacks of utilizing ROS revealed that scalability is a common concern among developers given the nature of its messaging model. However, given the projected size of this project and the relatively small number of concurrent processes involved, this scalability factor is not a point of concern.

3.3.7 Image Processing

For this project our team has decided to use the Raspberry Pi Camera Board v2. Compared to the stereo camera option and the Pixy Cam, it is the clear choice when taking into account our limited budget. Though it will not provide the same built-in image processing features as the Pixy Cam, we are confident that our computer science student team members will be able to provide the same functionality with their own software and support from us. While stereo camera options will provide higher quality images for processing, we must consider that we will still be able to accurately and effectively avoid objects with the Raspberry Pi Camera Board v2 at a fraction of the costs.

Figure 17: Raspberry Pi 3 on Board Camera
Permission to use granted by Helen Lynn, Raspberry Pi foundation.

3.3.8 Circuitry Materials

The materials we will be utilizing for the electronics portion of the project will vary in quantity, but the overall types of materials will stay the same. Many of the basic components such as wires, electric tape, and solder will be generally used to connect and complete circuits. Additionally, we will introduce a PCB that will neatly organize much of the network communication.

We may require components such as voltage regulators and will definitely need DC to DC converters to supply power to our microcontroller(s), sensors, and other electrical components that can be used to supply power or communication to other components. Overall the circuitry materials will be purchased and decided upon when needed. Due to the environment and uncertainty of the nature of the project material needs will vary.

3.4 Possible Designs and Related Diagrams

Below is the top-level design for our scope of this project. A charge controller will be in place to regulate the voltage and current coming from the solar panel to ensure we do not over charge or damage our battery. The battery will supply power to the motors and will more than likely require a step-up voltage converter. The battery will also supply power to a PCB which will potentially step-down voltage to supply power to the sensors, camera and microcontroller. The microcontroller will be connected to the camera and will handle image processing. The microcontroller will also receive input from the PCB which is receiving input from the sensors. Additionally, the microcontroller will be connected to the GPS system.

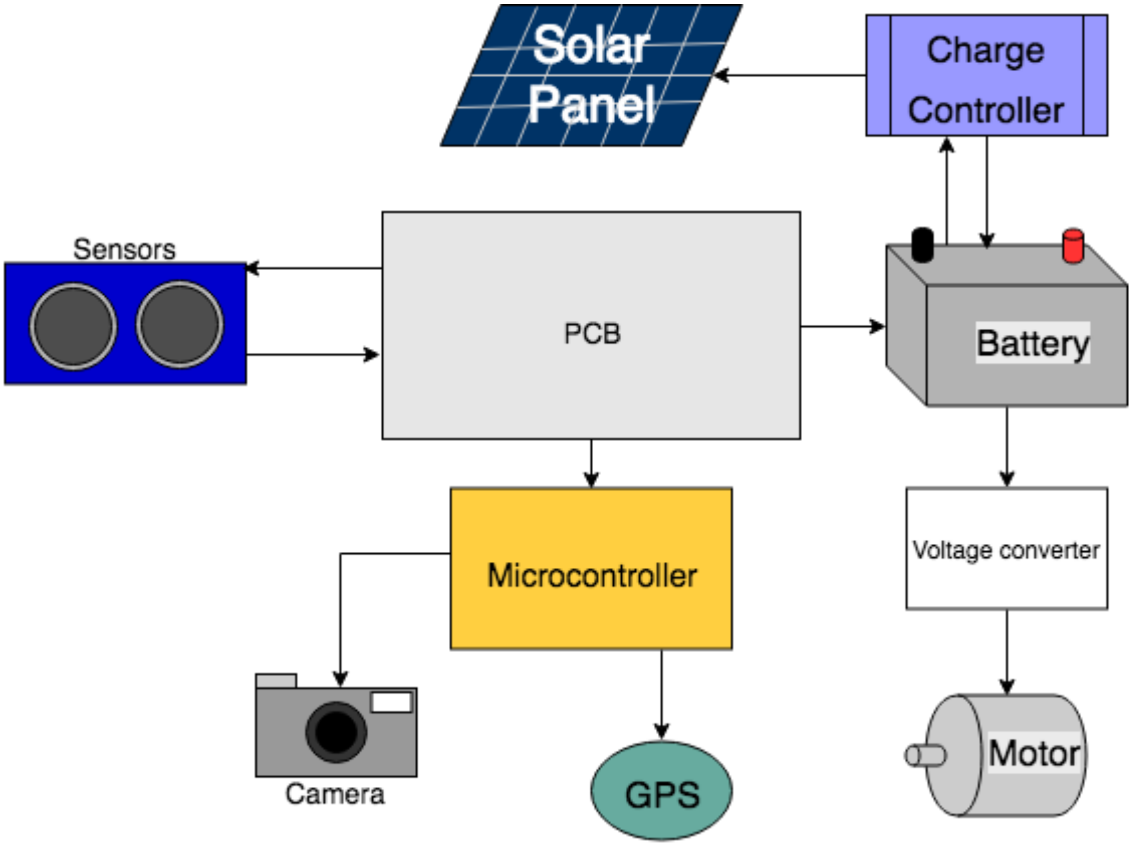


Figure 18: Possible System Diagram

As components are selected and testing is performed, the diagram will change and become more detailed.

3.5 Beach Site Visit

In order to get a better understanding of the environment we will be presenting on we had taken a day trip to the east coast. We walked about a mile of the beach taking note of the surrounding physical layout and the areas the beach goers would hang out in. The majority of the beach was flat with large soft sand areas. Only a small portion of the beach would be considered dangerous terrain because of the water.

Figure 19: Site Environment

During our observations at the beach we noted there were some largely congested areas of people where almost the entire width of the beach was filled with people. We anticipate areas like this to be the most challenging to navigate. Other than some congested areas, much of the beach had little no to no people, and only minor terrain to navigate. From our observations we deduced that the vehicle weight can sink the buggy into the sand causing more friction and that the presence of overcast can diminish the power generated by the solar panel.

The site visit was also completed in a time that would be more active with visitors than anticipated on the day of this challenge. The visit was done during typical spring break days, whereas the challenge with be completed in December 2018. This visit has shaped the remote and site testing plans giving insight into temperature and terrain.

4.0 Related Standards

This section discusses various related standards that were distinguished and considered during system design and development. These standards were used to guide the autonomous solar beach buggy project to meet milestones successfully, while ensuring the project's reliability, quality, and safety. More specifically, we will review relevant standards that involve software, hardware and safety; as well as discuss the design impact of implementing these standards into our system.

4.1 Relevant Standards

The following standards were explored and implemented during the development and design of the autonomous solar beach buggy:

Standard	**Scope**	**Publish Date**	**Description**
PEP 8	Software	5-Jul-01	Python Enhancement Proposal 8, a coding standard and Style Guide to organize code and increase readability [36].
IEEE Std 1361-2014	Hardware	16-Jun-14	This standard provides guidance to understanding lead-acid battery charging requirements in relation all stand-alone photovoltaic (PV) systems where PV is the only charging source. Stand-alone PV system parameters and operating conditions discussed in relation to battery characteristics and expected battery performance. This document aids in battery selection, evaluation PV system design, and test plan [35].

Table 6: Software standards

4.1.1 Coding Standards

Although setting a coding standard is not a requirement, it is an advantageous practice to implement to maximize readability and maintainability of our programmers' source code. By following a coding standard, the programming team will be able to understand and contribute to each other's code from a single source code repository more easily. The application of a coding standard will also increase productivity and encourage collaboration and teamwork.

PEP 8 or Python Enhancement Proposal 8 is a coding standard and Style Guide that describes specific conventions for writing code in Python. The coding conventions consist of rules and guidelines for import organization, indentation, formatting, tabs, maximum line length, line spacing, comments, naming conventions, etc. PEP 8 states that 4 spaces should be used to infer an indentation level [36]. Continuation lines should align either vertically inside parentheses, brackets and braces, or using a hanging indent [36]. Spaces are the preferred method for indentation over tabs, but tabs should be used solely to remain consistent with code that is already indented using tabs. All lines are limited to a length of 79 characters, with the except of long blocks of texts, such as docstrings or comments, which should be limited to 72 characters [36]. In Python code, it is permitted to break before or after a binary operator (+ or -) as long as the convention is consistent, but recent mathematicians prefer to break before the binary operators for increase readability [36]. Top-level and class definitions should be surrounded with two blank lines, while method definitions are surrounding by a single blank line. Other uses of blank lines can be used sparingly to indicate logical sections or to separate groups of relate functions. Imports should be stared on separate lines. Strings can be single-quoted or double-quoted but should remain consistent once a rule is selected. Block comments are to be indent to the same level as that code and indicated with a '#' and a single space [36]. Inline comments are sparingly permitted as well. Function and variable names should be lowercase, using underscores to separate words to import readability. Camel casing is only permissible where the style is already prevalent, to retain backwards compatibility. Method names and instance variables follow the same rule as the function naming rule stated above. In contrast, non-public methods and instance variables use one leading underscore before its name. Two leading underscores are used to invoke Python's name mangling rules to avoid name clashes with subclasses, but this rule faces some controversy. This rule may be followed or ignored following a discuss among our team members and a consensus is reached. Constants are written using all capital letters with underscores to separate words [36].

PEP 8 is not limited to the coding conventions describe above. More detailed descriptions and examples can be found in the sources stated in Appendix A [36]. The goal of employing a coding standard is to set a uniform pattern of writing code to ensure cohesion among our software team members.

4.1.2 Hardware Standards

IEEE Std 1361-2014 is a standard written to "aid in battery selection, evaluation, PV system design, and provide a test plan for evaluating the selected battery" [35]. This standard is relevant to our autonomous solar powered beach buggy project because it provides direction in selecting the best battery for a photovoltaic (PV) system, as well as guidelines to effectively charging batteries in PV systems in addition to evaluating battery performance during system execution. In our system, the we will be using the solar panel that was describes in section 3.3.1. Understanding the advantages and disadvantages of various lead-acid batteries in PV systems is imperative during battery selection process. Lead-acid batteries are designed distinctly for its intended systems, which include Starting Lighting Ignition (SLI), deep-cycle, stationary float service, and energy storages systems.

According to standard stated above, lead-acid batteries designed for deep-cycle systems perform the best in PV systems due to its high cycle-life and resistance to detrimental deficit-charge cycling. A deficit cycle is a condition where a battery's bank receives less charge each cycle and starts to sulfate, becoming resistance to holding charge. More specially, it occurs when a battery is charged "with less 'Amp-hour' than 'it is' required to return the battery to its initial state-of-charge. This results in a reduction in the battery state-of-charge" [35]. Solar powered systems are susceptible to deficit cycling due to its reliance sunlight availability. This availability is affected by weather, time of day, clouds, shadows, etc. and may not be predicted with extreme accuracy. The cycle-life of a battery is greater when the depth of discharge is shallower. It is important to understand a selected battery's design limitations, charge and maintenance requirements to attain the best cycle-life for the battery and avoid untimely failures.

Self-discharging is also a concern pertaining to lead-acid batteries. All lead-acid batteries experience some sort of self-discharge, an effect that can be described as a loss of charge at all times due to ongoing internal electrochemical reactions. This becomes an issue when batteries are left in storage without continuous charges flowing to it constantly and the most effective method to minimize self-discharge is to storage the battery at low temperatures. Temperature has a serious impact on the quality and performance of battery. Most lead-acid batteries are rated to operate at 25° Celsius (77° Fahrenheit). "Battery charge or regulation voltage needs to be adjusted by a range of typically -0.003 to -.005 V/°C/cell as the battery temperature changes for cycling applications" [35]. When battery temperature is expected to stray from the manufactural rated 25°C, the voltage regulation needs to be adjusted accordingly or the lead-acid battery will hold charge inefficiently, which may also cause damages. The equation below is the formula used to calculate the correct cell temperature-compensated regulation voltage for deviating battery temperatures.

$$V_r(\text{Temp Comp/cell}) = ((\text{Bat Temp} - 25\ ^\circ\text{C}) \text{ x } -0.005) + V_r/\text{cell @ } 25\ ^\circ\text{C}$$ [35]

A PV battery test procedure is also provided in the IEEE Std 1361-2014, which is used to evaluate whether the battery performance is acceptable with the current charge parameters in the PV system design. The procedure tests the following elements: initial battery charge, initial capacity, battery cycle, final capacity, and performance. The acceptable battery performance and charging parameters differ per individual project requirements, but based on “previous fielded stand-alone PV systems and laboratory testing… an initial capacity loss of up to 10% is not uncommon… If final battery capacity is less than 90% of initial capacity, then the test parameters, system design, or battery design may be improved” [35].

The standard also states the importance of recharge efficiency and charge acceptance. Charge acceptance is “used to describe the rate at which battery can be recharged” [35]. This rate can be measured by measuring the time required for a battery to reach 100% of its state-of-charge (SOC), or by calculating the number of Amp-hour (Ah) returned to the battery under fixed and constant conditions. Battery coulombic or Ah recharge efficiency is measured by dividing discharged Ah by charged Ah, and this is dependent on multiple factors, some of which include battery SOC, design, grid alloy, charge rate, history, age, and cycle. These rates are positively correlated with PV array-to-load ratio.

The PV A:L ratio is the average daily PV ampere-hours or watt-hours available divided by the average daily load hours or watt-hours. This ratio is critical in PV system design considering it determines if there is sufficient charge Ah or Wh to recharge the battery during mediocre sun availability. A low A:L ratio can mean there is insufficient sun exposure to generate charge for the battery, which can cause permanent loss of charge capacity. Adversely, a high A:L ratio can mean the battery is under the conditions of charged excessively. A PV charge controller would be beneficial under in this case. A charge controller is designed to prevent a battery from being excessively charged.

One of the main goals of a PV system is to ultimately achieve satisfactory autonomy. Autonomy, under a photovoltaic definition, is the amount of time that a PV system can provide energy to the load without receiving energy from the PV array. The length of autonomy is purely decided by the scope and needs of the project. It is recommended to avoid excessive autonomy due to increasing cost and difficulty. The IEEE Std 1361-2014 provides more in-depth information on battery quality, selection, testing, performance, and testing procedures as well as a battery’s process of sulfation, electrolyte management, hydration, mossing, thermal runaway, etc. This information can be found under the reference described in Appendix A [35].

4.1.3 Other Relevant Rules and Regulations

The autonomous solar-powered beach buggy is planned to navigate from Daytona Beach to the Ponce Inlet, on public property, therefore it is important to review and abide by the rules and regulations of the beach to ensure the safety of the public and wildlife.

Driving a vehicle on Daytona Beach is permitted in specific areas, given ideal weather and tides conditions, from sunrise to sunset during November 1 through April 30 (the expected time of the official test-run). The posted speed limit of 10 miles per hour is strictly enforced and traffic lanes are clearly marked for vehicles to follow [39]. If violated, a fine will be given to the operator of the vehicle. A vehicle's headlights must be on while driving, as well as with at least one front window down. Noise or music that ranges beyond 50 feet of the source is illegal. It is also illegal to disturb dune vegetation or drive/park in the conservation zone [39].

Figure 20: Daytona Beach Driving Map

Permission to use in progress. [38]

As this is a moving vehicle, transportation standards must be kept in mind. Road laws will not be directly applicable to this project, as the project will never reach a public road, but it in best practice to take into consideration the laws of the road for vehicles of this nature as they are in place for safety.

4.2 Design Impact of Relevant Standards

PEP 8 coding standard will be the guidelines in which our team's programmers will follow when they develop and write code, but ultimately slight deviations from this standard will be allotted if abiding by these conventions is deemed to take up too much time or if the software teams find it unprofitable. The sole purpose of abiding by this coding standard is to encourage collaboration and improve productivity though extending readability. If the coding standard is found to impede development or counter cohesion, the standard may be discarded or traded per the software team's discretion.

IEEE Std 1361-2014 provided our team with a variety of battery options that will work with PV systems. Ultimately, a battery was selected that was considered sufficient rather than best, given our monetary restrictions. Our electrical and hardware team will keep the battery self-discharging concern abreast and work towards developing a solar system that is efficient and will maximize battery performance. If we find battery storage necessary, it will be stored in a cool environment that will slow the effects of self-discharging.

Our team understands the need for voltage and charge regulation in order to prevent damaging the battery in our PV system as well and have incorporated two voltage converters and a charge controller in our design. The charge controller will be purchased to monitor and maintain the DC power being stored in the battery. One voltage converter will be designed within our Printed Circuit Board (PCB) to supply the appropriate voltage distribution to the ATmega328p and the Raspberry Pi and the other will be purchased to apply the appropriate voltage distribution to the motors. Temperature is also being considered during the development of our design since is it a major factor that affect battery charge, capacity, cycle, and performance. We have noted that up to 10 % of initial capacity loss is common and are working towards a system that will capitalize on the battery's full quality and potential performance. Although photovoltaic autonomy would be superb feat to accomplish, it is not needed in the scope of our project.

The autonomous solar-powered beach buggy does not classify as a standard vehicle therefore will not need to follow all the beach rules strictly, but all rules and laws regarding public health and wildlife will be enforced. Notably, the buggy will navigate the beach during the permitted operational hours and will not exceed 10 miles per hour. The vehicle will be designed to travel no more than 3 miles per hour, as described in the description. The vehicle does not have any windows or headlights by design, but also will not be operating beyond dusk so headlight may not be required. Noise should not range beyond 50 feet of the vehicle for it is designed to limit noise pollution during execution. The autonomous system is designed to detect and avoid both stationary and moving objects, while causing no harm to the environment or beachgoers, therefore it will logically avoid dune vegetation and conversation zones as well.

5.0 Realistic Design Constraints

This section discusses the design constraints that will guide the design and fabrication of the solar beach buggy. This section will cover economic, time, environment, social, political, ethical health, safety, manufacturability, and sustainability constraints. Each section will be addressed individually and directly related to the impacts they all have on the system.

5.1 Economic Constraints

The autonomous solar-powered beach buggy must be designed and developed in the most cost-effective manner due to it being under a strict budget of $2,000. The budget will have to be spread across all disciplines, covering the costs of, not only computational and electrical components, but mechanical components and construction as well. We expect to have some unaccountable obstacles arise during testing and implementation of the system and this may further complicate and constrict the budget. We plan to set aside a sufficient portion of the budget to mitigate the stress of these potential expenses. As development progresses and expenses are recorded and more defined, the quality of the system's components may be altered to remain within budget. These components include batteries, sensors, microcontrollers, and general mechanical devices.

The cost vs quality of the components needed by the autonomous system is positively correlated. The cost of sensors alone can range from $8,000 to $20. The more expensive devices tend to be more accurate and powerful. Our team is working with two other interdisciplinary senior design groups to decide what comprises the optimum combination of cost and quality.

5.2 Time Constraints

The project must be completed in a two-academic semester period, giving us roughly six months to collaborate with other teams, plan, design, test, and execute the project. Team members will have to work around individual, professional, and academic schedules, as well as long distance geographical commutes to meetings. Our team's timeline is even more constricted because of our delayed affiliation with the solar buggy project. Due to unforeseen circumstances, we were forced to forgo our first design project and change our senior design venture two months into the Spring academic semester. On top of this setback, it took an additional three weeks to communicate with the proper instructors and officially integrate with the other interdisciplinary groups working on this project.

5.3 Environmental Constraints

The project is required to be solar powered and thus presents an immediately identifiable environmental constraint. Not only can this impede on the day of presentation for the project but also can limit the testing of our system. The solar panels that we will be using are rated for an environment with theoretically 100% sunlight. Florida can have mixed weather on almost any day of the year. This can create cloudy weather, rain, thunderstorms, and other weather patterns that complicate power consumption readings and power storage potentials. We anticipate that we will need as much power generation from the solar panels as possible and therefore we will be testing under sunny conditions and intense sunlight for the best results.

5.4 Social and Political Constraints

Social constraints can be either external or internal. External social constraints could include private parties or individuals that impede with testing or tamper with equipment or the devices themselves. Other external social constraints include any individuals or parties we need to contact for building and welding the physical model of the apparatus itself.

Internal social constraints are limited to groups and individuals within the project itself. Our group has agreed that any emergency matters regarding family or personal reasons is immediately excused from pending group meetings. If individuals wish to remain conservative as to the reasons and details regarding why they weren't able to attend meetings, it is their prerogative to remain silent. Beyond health, work, and family matters, group members are asked to remain open and communicate their schedules and meeting times with other group members.

Political constraints are extremely unlikely but will be evaluated and addressed as they come into focus for the project.

5.5 Ethical, Health, and Safety Constraints

Safety will be a top priority when designing the apparatus. The chair will be easily mountable and demountable, possibly containing seatbelts or other devices aimed at securing the passenger as well as any equipment on the device. It will be a top priority that no passenger can get hurt physically or electrically by the apparatus. No sharp, or dangerous object will be protruding from the apparatus and all electrical equipment will be stored safely, and out of reach of passengers. The challenge will take place on a public beach. Because of this, the safety of potential beachgoers and wildlife must be taken into consideration as well.

Ethical and health constraints are extremely unlikely but will be evaluated and addressed as they come into focus for the project.

5.6 Manufacturability and Sustainability Constraints

The end product for this project will be a large apparatus capable of carrying a person up to the maximum weight of 120 pounds. To make the apparatus as aesthetically pleasing as possible the group will consider where to store sensors, electronic equipment, wires, microcontrollers, motors, batteries, and other equipment that cannot function under wet/unstable/dirty/hot environments.

The environment that the apparatus is expected to function in, the beach on a hot, sunny day, adds challenges in itself- increasing the need to worry about factors such as overheating and potentially even water damage. The apparatus will contain a chair for the passenger to ride in as well as possible other extremities for comfort. More contributions to the size physical design will be addressed as needed.

Sustainability also needs to be considered. We are going to be completing this challenge in a natural environment and must do our due diligence to ensure we leave the environment without impact from out vehicle. Special considerations will be taken to ensure we do not disrupt or injure wildlife.

6.0 Project Hardware and Software Design Details

This section discusses our initial design and the details of our selected components as they relate to our design. We will also attempt to explain part orientation within the project and how they will be coordinated with other systems.

6.1 Design Architectures and Related Diagrams

Below is the initial proposed system diagram for our scope of the buggy. The HQST 30 AMP PWM Smart Solar charge controller will be in place to regulate the voltage and current coming from the solar panel to ensure we do not over charge or damage our battery. The battery will supply power to the motors, which will be selected by the mechanical team, and will be connected for a 12 to 24 step up voltage converter. The battery will also supply power to our PCB which will contain the ATmega328p and a circuit to step down voltage from 12 V to 5 V. The PCB will supply the step-down voltage to the sensors, camera and Raspberry Pi 3. The PCB will also be receiving raw data from the sensors, which will be processed into usable data and transmitted to the Raspberry Pi. The Raspberry Pi will be connected to the camera and will handle image processing. The Raspberry Pi will also process the data received from the PCB and use this, the images received from the camera, and data gathered from the GPS system to control navigation.

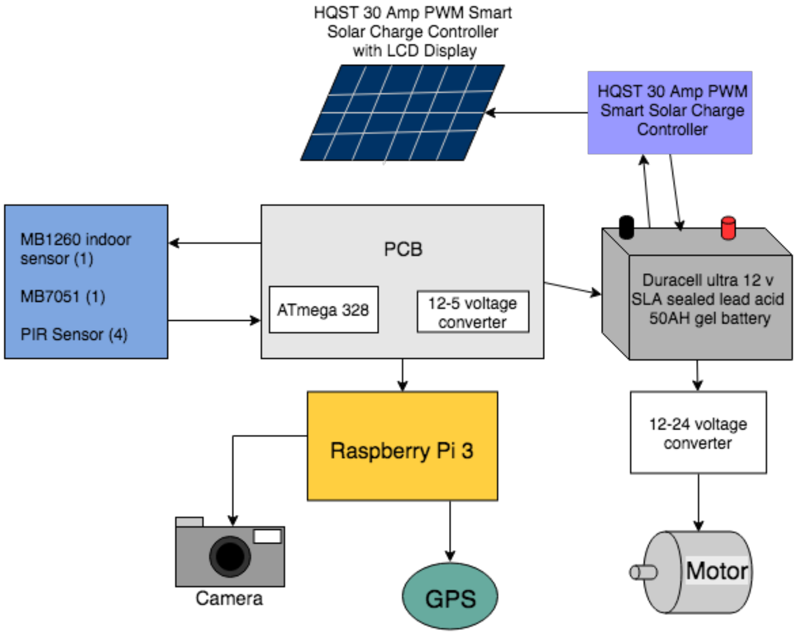
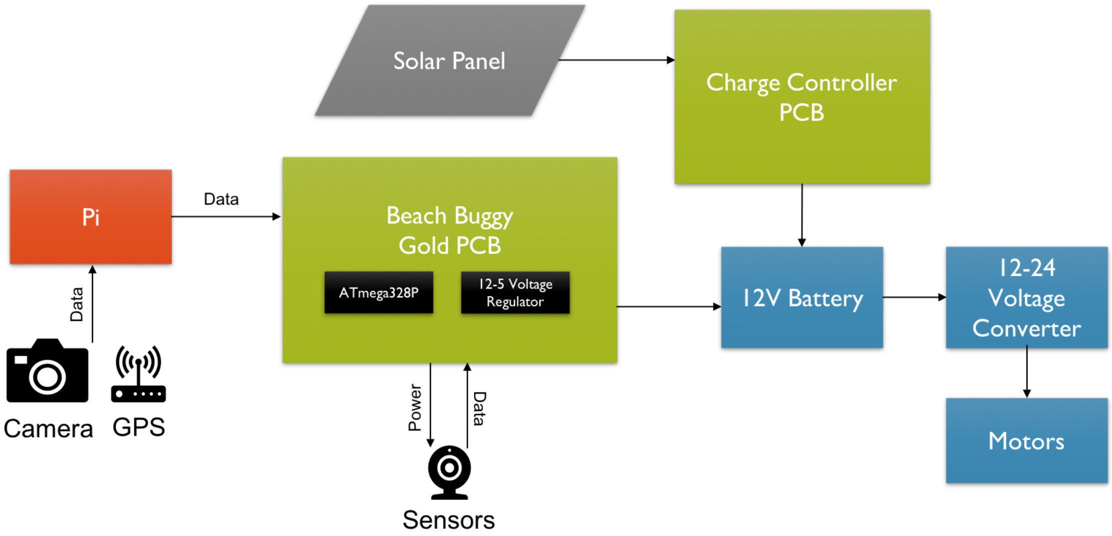


Figure 21: Initial System Design

The finalized system diagram did not differ greatly and can be seen in the following figure.



6.2 Sensors

For our project we plan to use three PIR sensors, two ultrasonic sensors, and one camera. The three PIR sensors will be used for close range proximity to account for object detection. We will position the sensors on the vehicle as to where there is as few dead spots as possible. The two ultrasonic sensors will be attached at the front of the vehicle and will be used for object detection at long and medium ranges. The camera will be mounted on the front of the vehicle for close range detection.

6.2.1 Long Range Sensors

There are two sensors that will be used to determine long range obstacles and they are both ultrasonic. The MB7051 and the MB1260 will be forward mounted on the vehicle.

The MB7051 is our high powered long range outdoor sensor and will be position on front nose of the vehicle and the MB1260 will be positioned next to it. The MB7051 will take in narrow long-range readings out to 10.5 meters directly in front of the vehicle. This narrow sensor will be able to tell the system what obstacles lay in front of it and whether or not it will be able to continue on its forward path. If this sensor detects objects and interferences the information will be sent back to the Atmega328 and then the Pi for course changes in it navigation routine.

The MB1260 is our wide angle indoor sensor. Although it claims to provide a range out to 10.5 meters, because of the nature and its testing environments it will not be able to reach its full potential. We will instead tailor back its detection range and use it as a wide are ultrasonic sensor for medium to close range proximity sensing that will work similarly to the forward camera mount.

Ultimately, we focused our project on close-range sensors only, allowing the computer science scope to handle the bulk of the long-range detection.

6.2.2 Close Range Sensors

The three PIR sensors will be mounted in a stationary position on the front, left, and right sides of the vehicle to ensure that all close-range objects will be detected and avoided. The sensors have a detection angle of 180 degrees and therefore will allow us so unsure we have no dead spots surrounding the vehicle.

The PIR sensors output a high state to one of its Pins by default if an object is detected. Just utilizing this we can determine consistency of the sensor detection. If we pair it with unique coding, we can create many different types of sensor detection that is much more intuitive than simply two state detection. We will further talk about the implementations and potentials of these sensors within our software section.

We will be using a camera for object detection. Our environment on Daytona Beach can be rather linear and intense for our ultrasonic and PIR sensors. Because the camera can detect visible light we will be able to distinguish between objects that appear on the same surface but have distinct colors. Beach towels, tents, and coolers are just a few examples of slender and flat objects that PIR and ultrasonic sensors would have difficulty detecting. The camera will function as a primary source of information for the navigation system.

Ultimately, the DS18 PRKDI4 is the brand of ultrasonic sensors selected to be used for the beach buggy. The specification of the reverse backup sensors is noted in Table This specific brand came with four sensors, and a color LED display that may be used to display distance between the sensor and an obstacle. The sensor are detachable units with waterproof socket connectors. They are quite durable, made from high quality components, and able to withstand extreme, elevated temperatures, which is ideal in a competition that takes place in a baking, humid beach environment.

Table 7: Specifications for DS18 PRKDI4

Specification	Rating
Voltage:	12V – 16V DC
DC rated current:	10 mA – 250 mA
Detection Distance:	0.3 m – 2.5 m
Working Temperature:	-30° – 80° Celsius
Ultrasonic Frequency:	40 KHz

The angle of reliable detection was tested to determine the best mounting placement of the sensors for autonomous navigation. This is necessary to reduce the size of the buggy’s blind spots and achieve successful object avoidance. It was found that there is an 15° angle of accurate detection radiating out from the center of the sensor, but only for the first meter. This is represented by the solid red cone are in Figure 6.

These automotive sensors are rated to detect almost 2.5m out, which proved to be true for a much narrower angle of detection beyond 1m, depending on the size of the object or obstacle. For a smaller object, the object had to be placed nearly centered to be detected. This is visually represented by the red-striped area in Figure 6. The overall dependable cone for short-range detection is 30° for the first meter, and approximately a cone of 10° degrees from 1m to 2.5 m.

The process of decoding the sensors’ binary messages was a tedious task due to the lack of documentation on this specific brand of sensors. Before deciphering the binary, the PWM signal produced from the sensors itself must be broken down into binary packets. Each port or channel sends the signal at the same frequency of ~40KHz. It is possible to send multiple signals over the same frequency because they are sent at separate times. Each port sends a signal about ~40ms between one another in a sequential loop. [5] In order to visually see the PWM signal, the data pin of the automotive sensors was connected to an oscilloscope, the result is shown in Figure 7. It was discovered that there are 17 voltage HIGHs or peaks of different lengths sent by the sensor, the first square pulse, being the longest pulse, indicated the beginning of a sensor's data. This starting pulse was typical around 800µs – 870µs long. The next 16 pulses were deciphered to form two bytes or 16 bits of sensor data. Pulses with the width 350µs or lower were bit 0, and pulses with pulse widths with 480µs or higher were bit 1. The first byte of data contains the distance the sensor detects an object in meters, and the second byte contains the address of the port that detected the specified distance. The mapping of detected distance to its respective sensors is crucial in formulating object avoidance algorithms.

Upon further evaluation, it was determined that the sensors were sending signals in Least-Significant Bit, or LSB bit order. Converting the binary data packet required reordering the bits in Most-Significant Bit, or MSB order per byte and then finding its complement. Then the binary number could be converted to decimal regularly.

6.3 Microcontrollers

As discussed, we will have two microcontrollers involved in the project. We will first discuss the implementation the Raspberry Pi developers board and then our Atmega328 PCB in the sections to follow.

6.3.1 Raspberry Pi 3

For the duration of the project we plan to primarily use a Raspberry Pi 3 model B as our microcontroller. The Raspberry Pi 3 consists of a 40 Pin GPIO which we will utilize alongside our sensors for input information. The Pi also carries a housing station for the additional camera attachment which we will be utilizing.

The Pi is driven off a 5v voltage source and uses roughly 750mA in computation and low power modes. The Pi can support up to almost 2.5A in deliverance to sensors and other devices, making it quite reliable. It has a rated 1.2 GHz processor making it powerful and fast enough for many of the computations we anticipate. The Pi also features I2C communication channels where we can use slave and master virtual communication to exchange information between sources.

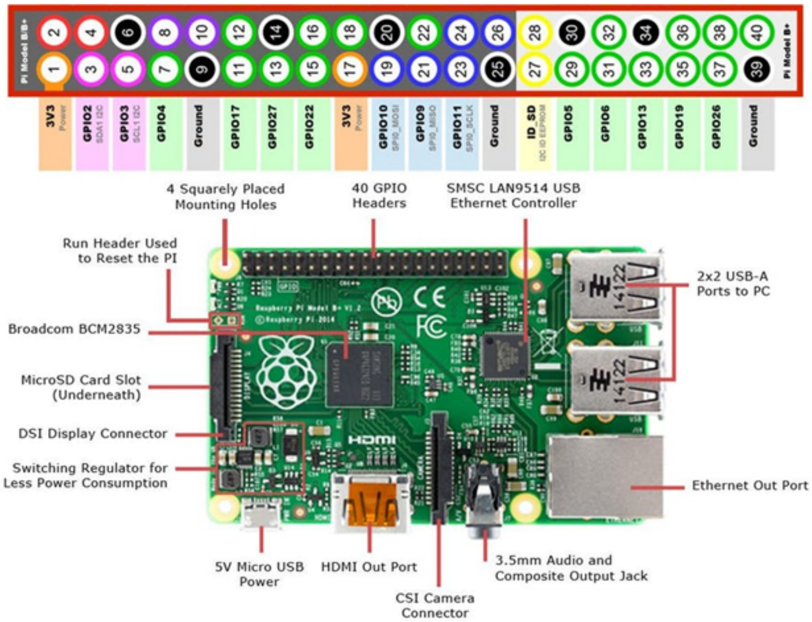


Figure 22: Raspberry Pi 3 GPIO Pinout and Various Board Information
Permission to use in progress.

Additionally, we have the UART configuration where we can use the RX/TX receive and transmit signal lines to communicate quickly between programs. All of this versatility in communication, and the powerful processing power for such a small device makes the Pi a great microcontroller for this project.

The Raspberry Pi also features wireless communication options between other electronic devices. We will be able to access the Pi without direct contact through our laptops and control programs and software as if we were connected to the Pi itself.

The Pi has a microSD slot for hard disk space storage allowing us for a modular storage. This gives us practically unlimited potential to store information relating to the project. Featured on the Raspberry Pi 3 I/O port is one Ethernet slot, four USB 3.0 slots, and one HDMI slot. The Pi has much versatility when it comes to customization on PCB of the Pi. Almost everything on the Pi's surface is solderable, giving us access to alter any hardware on the Pi if we deem it necessary.

As described most of these functionalities will be used or useful when communicating with the ATmega328. The Pi will function as the navigational microcontroller and house the main software functions for the project.

The Pi will also serve as the main processor for the computer science scope of the project.

6.3.2 ATmega328

The ATmega328 microchip comes preinstalled on the Arduino Uno development board. For these reasons the microchip and the development board share many of the similar properties. The Atmega328 can operate on 7 to 12 volts safely through USB connection or an external power supply, such as from an AC-to-DC power adapter or battery. The power Pins include Vin, 5V, 3V3, and GND. Vin is the input voltage Pin to the board when using an external power source, and conversely GND is the ground Pins. The 5V Pin is a regulated power supply used to power the ATmega328 or other components on the board. The 3V3 Pin is a 3.3 regulated volt supply that draws a maximum current of 50 mA.

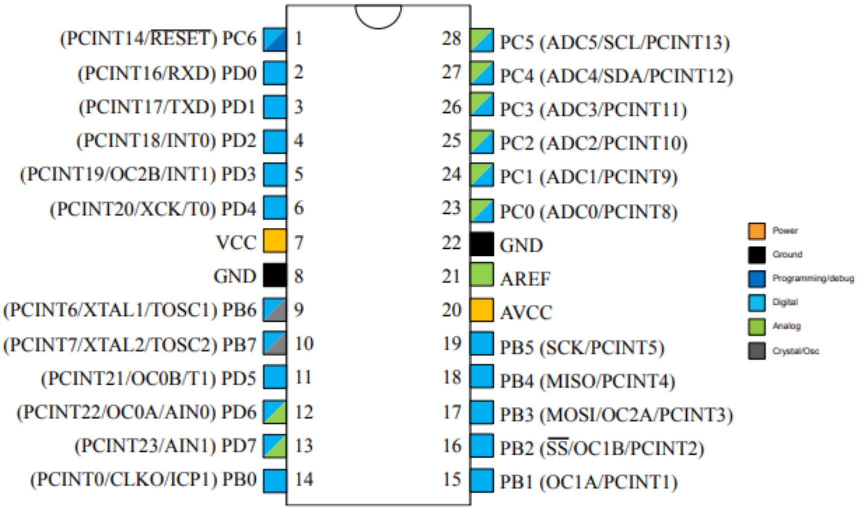


Figure 23: ATmega328 Chipset Only
Permission to use in progress.

The ATmega328 microcontroller uses 0.5 KB of its 32 KB of flash memory for its bootloader. The rest of the memory is used to store uploaded code. The ATmega328 has 2KB of Static Random Access Memory (SRAM) as well as an Electrically Erasable Programmable Read-Only Memory (EEPROM) library that can be used to read and write to its 1KB of EEPROM.

The ATmega328 has 14 digital Pins and 6 analog Pins. Pins 0 and 1 (serial Pins) are used for receiving (RX) and transmitting (TX) serial data respectively. Pins 2 and 3 are external interrupts, which can be programmed using the attachInterrupt() method to trigger an interrupt based on a clock, i.e. a low value, a rising or falling edge or a change in value. Pins 3, 5, 6, 9, 10, and 11 are 8-bit PWM output Pins that are used with the analogWrite() function. Pins 10 (SS), 11(MOSI), 12 (MISO), and 13 (SCK) are Pin that support Serial Peripheral Interface (SPI) communication, which is a synchronous serial communication interface provided by the underlying Arduino hardware. Pin 13 is just a build-in

LED connected to Pin 13 and this Pin can be useful for testing purposes. When the Pin is HIGH, the LED is on. Inversely, when the Pin is LOW, the LED is off.

The analog Pins can provide 10 bits of resolution which opens our project to adding a simple LCD display screen attached to the PCB. An AREF Pin and a Reset Pin is also available for use if necessary. The AREF Pin provides a reference voltage for the analog Pins and it used by the analogReference() method called in the Arduino IDE. The Reset Pin connects the LOW line to reset the microcontroller, but typically there is a reset button as well.

The ATmega328 PCB will act as the initial sensory input data operator. This means that all the peripheral sensor data will be sent to the Atmega328 and will undergo initial data deconstruction. This deconstruction will be responsible for analyzing the raw data input and categorizing it in our software for navigational purposes in the Raspberry Pi. The Atmega328 will additionally house am external oscillator if the sensors need to operate at a required frequency. This external oscillator will be attached to the PB6 and PB7 Pins for operation if needed. Further explanations and implementations of the ATmega328 will be addressed in the software portion of the project. Below is our schematic and of the Atmega328 microchip.

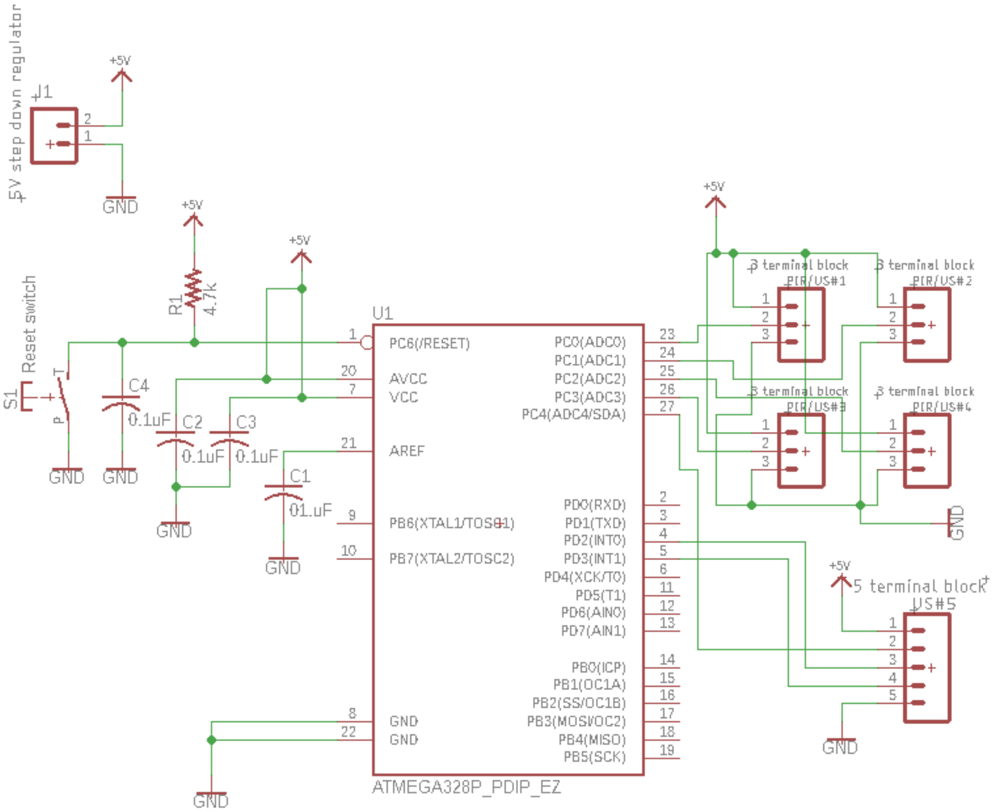


Figure 24: ATmega328 Microchip Schematic

This schematic features a 5v header Pin for power input into the microcontroller. In terms of power control, it also has a reset switch that has a pull-up resistor incorporated into it. The schematic has four three-Pin header banks that will be used for sensor input. The microcontroller will coordinate data between these header Pins. The schematic also features a five-Pin header that will be used in conjunction with our MB751 ultrasonic sensor for its three output information states. Capacitors C1 through C4 are used for noise cancelation and power orientation for the microcontroller. lastly Pin 9 and 10 offer and external oscillator connection even though the ATmega328 has its own internal oscillator. We will be revising this schematic before it is finalized.

Below is a PCB of the above schematic. This PCB will undergo final revisions before being ordered and manufactured. Potential changes to the PCB include drill holes, via holes, surface mount Pints or through hole Pin additions. The overall layout of the PCB may change orientation so that the routing of the connections would be more efficient. We plan to keep the printing that labels the PCB and intend to add a UCF knights logo to the board if allowed by the university.

6.4 Charge Controller

The solar panel has a current output at 7.79A with a maximum voltage output of 30.2V, supplying the system with 235W at maximum load. Considering these limitations, a charge controller design rated above 10A is utilized to ensure that the controller is powerful enough to regulate the charging of the 12V battery. The solar panel will only be able to supply partial energy while the beach buggy is on and under full load. For 12V systems the efficiency is above 96% and includes losses in the reverse battery protection MOSFETs.

The solar (MPPT) charge controller TIDA-00120 has been drafted from its schematic and created in Eagle CAD as a PCB, using the same required ICs. Some smaller components, such as resistor and capacitor values, were modified to adjust the design to best fit its intended use. Larger components such as the MOSFET's are modular for the board. By replacing the current MOSFET's with higher voltage and current ranged MOSFET's, a variable range of voltage and current values can be achieved. TI has recommended MOSFET's for this charge controller on its datasheet. Depending on mechanical requirements and power requirements these components can be easily replaced by soldering these higher rated surface mount MOSFET's. Additionally, this PCB features its own MCU to accommodate for optimization coding that will drive the charge controller. The code that drives the PCB operates on a specific tracking system that aims to regulate the output voltage based on previous measurements of power. If panel power increases, then the code modifies it to track in the same direction. Likewise, if the panel power decreases the code tracks in a decreasing manner. The PCB is powered directly from the solar panel via a voltage regulator on board. The circuit also features a low-dropout regulator that aims to help supply power to the device even if input and output voltages are similar.

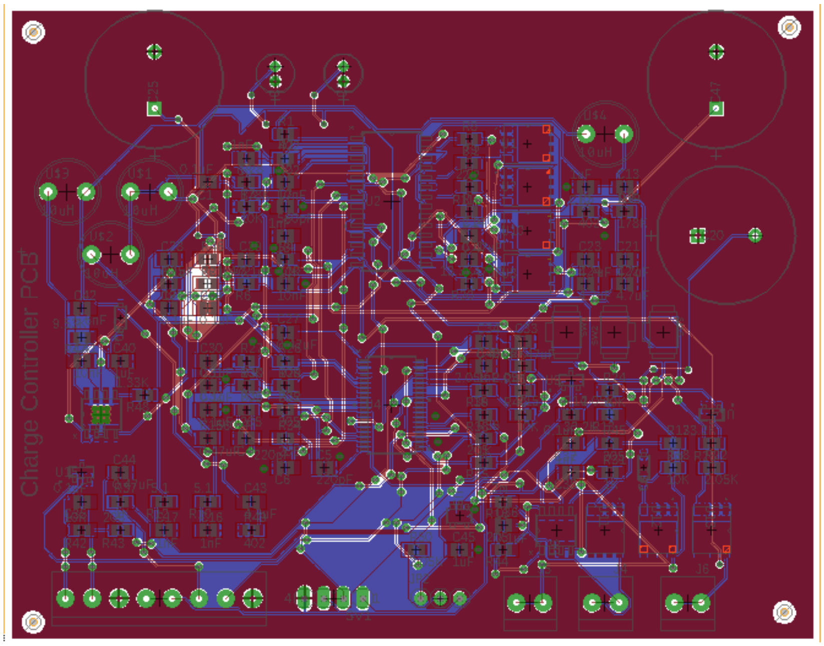


Figure 25: Charge controller PCB Design.

The solar panel is connected to a charge controller to monitor the charging of the power supply, preventing the battery from being over or under charged. The charge controller model selected was built using a schematic design on Texas Instrument's website

6.5 Battery Storage

As we've mentioned we will be deciding on our battery as late into the project as possible. This is due t the economic constraints put onto us. For now, we have decided to implement the 12v 50Ah battery as described before. The battery will be attached to the charge controller and supply the entire system. Power will be stepped up and stepped down accordingly for which components of the system require power. The battery will have to moderated for its total charge capacity because if the battery falls below a certain total threshold the batteries total capacity will be damaged. This would result in the battery not being able to receive a full charge in the future and lead to a deterioration of the system, making it needing replacement parts and upkeep. We have attempted to accomplish this by making calculations on the maximum run time on the battery and creating software that will take in the batteries charge as an input. Overall, we will ensure that the system does not lose its total charge.

Additionally, we need to ensure that the battery doesn't lose energy through is naturally degrading trickle charge and it doesn't become over charged from the solar panels. The charge controller has natural limiters that will be able to keep the battery from going to either of these extremes, but we will attempt to take additional measures to ensure these safety features.

6.6 Voltage Regulator

For this project we will be using a 12v battery. In order to properly distribute power throughout the system, we need to implement two voltage regulators. One to step down voltage for smaller components and one to step up voltage for the motors.

6.6.1 12-to-5 Regulator

We will need and have developed a switching regulator to step down the voltage from 12v to 5v for our microcontroller, PCB, sensors, and cameras. This step-down voltage will be accompanied by a header bank that will accommodate multiple pin locations, so we can connect as much as we need. Most of our 5v equipment in total will only be using about 2.5A ~ 3A. We have design our switching regulator to accommodate up to 5A allowing for extra components to be later attached and to assure that any fluctuations in current from the devices can be handled and supplied by the regulator. The diagram below shows our voltage regulator as a schematic.

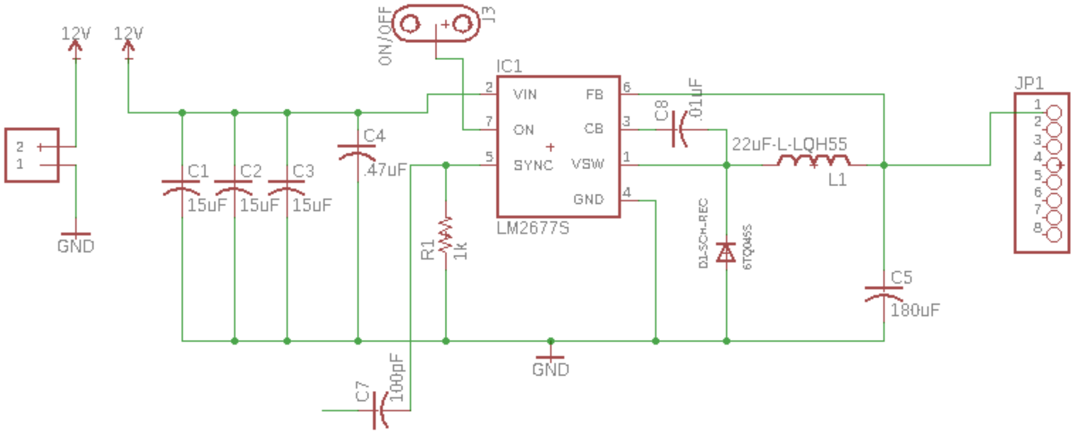


Figure 26: 12v – 5v Switching Regulator Schematic

The above schematic was based off the datasheet for the LM2677S voltage regulator. The schematic features the input header bank on the left side for the battery and the output header bank on the right for input to sensor equipment and the Atmega328 microcontroller PCB. The inductor within the circuit can be replaced to create different variations of this regulator. The capacitors C1 though C4 act as noise cancelation capacitors. The capacitor C7 is left open and not tied to ground. This is where we would add in an external oscillator if need be. Resistor R1 is for stability in the regulator. The diode prevents feedback current from damaging parts of the regulator and circuit. Capacitor C8 is used in conjunction with the regulator to help boost the voltage though the circuit. Capacitor C5 is used for stability and noise cancelation. Lastly there is an on/off switch built into the LM2677 that can be accessed via the on/off switch connected to Pin 7.

This PCB will undergo final revisions before being ordered and manufactured. Potential changes to the PCB include drill holes, via holes, surface mount Pints or through hole Pin additions. The overall layout of the PCB may change orientation so that the routing of the connections would be more efficient. We plan to keep the printing that labels the PCB and intend to add a UCF knights logo to the board if allowed by the university.

Several revisions we made to the PCB design. Due to a learning curve, many errors were made in the initial implementation of the designs in EAGLE Cad. Ultimately, the final PCB schematic can be seen in the following figure.

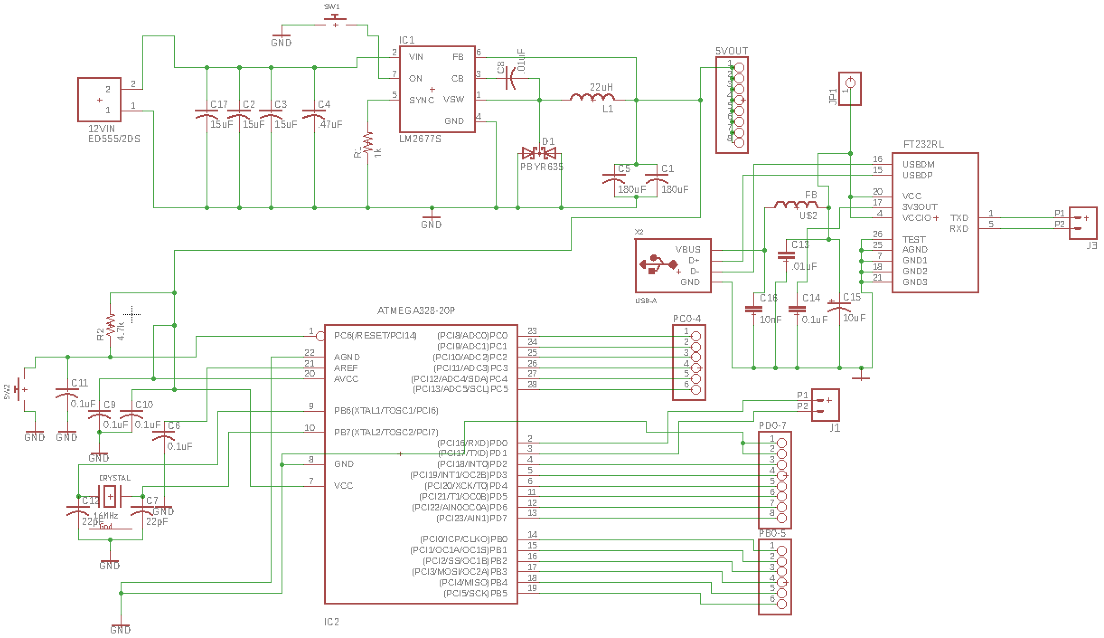


Figure 27: Final main PCB design

6.6.2 12-to-24 Regulator

Our other voltage regulator would be to step up our 12v battery to a 24v supply for the motors that wll drive the vehicle. This function of stepping up to a 24v from a 12v is slightly uncommon and really only presents problems in the amount of current we need to deliver. Since the vehicle will be dual axil, we will be responsible for powering both of the 24v 300w motors. This creates a huge complication for creating a switching regulator that can handle this degree of power. We have run into complications because of time and resources for finding information on switching regulator components of this caliber. Because of this our team is having to consider buying two 12v to 24v 360w regulators, or one 12v – 24v 720w regulator for the two motors in parallel. In general, the group is very conservative when determining the actual load demand by the motors. Upon further investigation into the project and actual testing in relatable environments we will be able to determine if the motors will be demanding their maximum power constraints. There are slight advantages to doing a step down from 24v to 12v rather than doing a boost from 12v to 24v. Brief research showed that due to electrical connections on the inductor, a system will save more energy if you step down the voltage. This will also help us to recover every drop of wattage from the solar panels. Below is a Picture of a 12v to 24v 360w regulator we have considered purchasing.

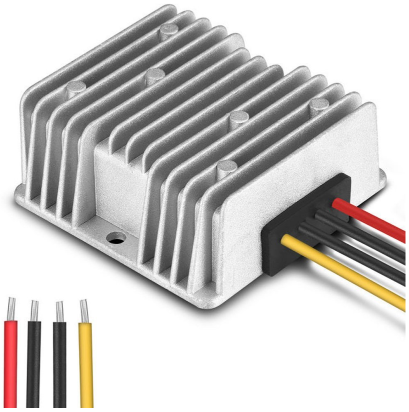


Figure 28: Aweking Waterproof DC/DC 12V Step Up to 24V 15A 360W Voltage Boost Converter

Permission to use in progress.

6.7 Robot Operating System (ROS)

ROS is a versatile middleware framework that is used for developing robotic applications. With a myriad of libraries and packages devoted to making robotic software development easier, this framework was selected to be basis of the beach buggy's autonomous navigation software. Our computer science partners are planning to use ROS to command the navigation of the vehicle after analyzing sensory data provided by our team from our system of sensors. Our microcontrollers must be ROS compatible and programmable using Python. ROS may be used to communicate between various nodes in our system through its anonymous publish and subscribe system, but this is up to the discretion of the computer science counterparts. A choice between asynchronous and synchronous communication is also offered by the ROS framework. Under any case, software development and testing support will be offered to our computer science team as needed.

6.8 Global Positioning System (GPS)

The computer science students into to implement a GPS module to aid in the navigation process. The Raspberry Pi will receive input as GPS coordinates from a remote device. Using the GPS module, the agent will identify its location in respect to the inputted coordinates and determine a "soft-path" to its destination. This path will give the agent a sense of direction, but its malleable path will be primarily defined by the agent's avoidance collision. The path calculation will constantly be re-evaluated as to consistently give the agent localized information in relation to its destination. We will assist the computer science students with this effort as needed as it relates to our scope of the project.

6.9 Master Printed Circuit Board (PCB)

Below is a Picture of our initial master PCB that will encompass all of the PCB's into one master board. This PCB features our 12v – 5v switching regulator and the ATmega328 microcontroller. The bear also features all of the terminal blocks that we would use for the sensors and input power. Being that this master PCB has only had minor revisions, further revisions will be performed before this board is ordered. We are also considering the size of our header blocks as well as the battery header blocks. All of our capacitors, resistors, and diodes are surface mounted, and we are considering partial through hole mountings for some if not all of the board. Further revisions will be performed to find the optimal solution. This board is as follows.

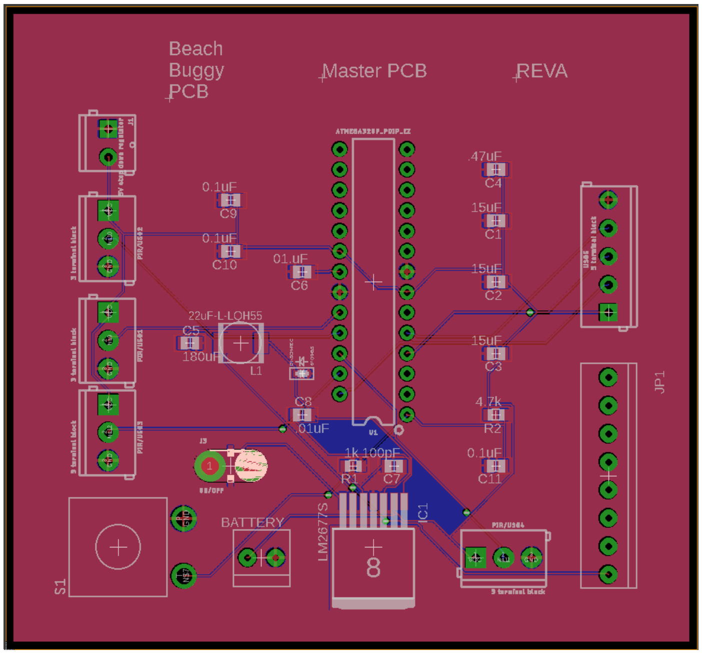


Figure 29: Initial Master PCB: REVA

6.10 Electrical Housing

In order to prevent our electrical equipment from becoming disabled or damaged, we will be housing it in a safe section of the vehicle. Once the vehicle schematic can be observed we will analyze where we can attach a sealed electrical box. We need the box to be off the ground and not within one foot of the sand, for it cannot drag on the ground or be hit by obstacles or terrain that the vehicle may pass over. Our team plans to not only insulate the box from outdoor hazards but make sure it has air flow running through the system. We plan to equip the box with at least two computer fans that will create a one-way air flow direction to ventilate the heat emitted by the electrical components. The box will be made of a protective but low insulative material such as acrylic, or other hard plastic materials.

7.0 Project Prototype Construction and Coding

This section will discuss the assembly of the prototype of the buggy. Due to ourscope of work, it must be completed in conjunction with the frame of the buggy being built, therefore we will be testing on a smaller prototype.

7.1 Parts Acquisition

For this project our sponsor has taken the liberty of already supplying us with a solar panel. See section ***3.3.1 Solar Panel*** in the Strategic Components section.

The following parts will be acquired based on the most reliable source offering the most competitive pricing:

- Raspberry Pi 3 Model B
- ATmega328p
- Duracell ultra 12 v SLA sealed lead acid 50AH gel battery

We will do extensive research to ensure we have our PCB manufactured by a reputable source offering competitive pricing. It is the backbone of our system and high importance will be set to ensure we minimize all risks involved in not having a proper PCB.

We used PCBWay, a full feature custom PCB prototype service based in Hangzhou, China, due to their quick turn around and competitive pricing. We ordered three separate PCBs from them- one was our first version of our main PCB containing the voltage converter and the ATmega328p, the second was our charge controller design, and the third was our final and functional version of our main PCB design.

Parts selection proved to be one of the more challenging aspects of PCB design. Special considerations needed to be taken when selecting parts, including, but not limited to, package size, voltage ratings, amperage ratings, and material. In designing the PCBs, we often found ourselves having to research appropriate replacements for ideal part selections due to lack of availability. Our primary parts suppliers were Mouser and Digikey, who both offer same day shipping on orders.

7.2 Assembly

Figure 31 shows the 3D printed frame that we will be using to do prototype testing. This small prototype will be primarily used for the initial development of the system, the learning of sensor integration, and how the data will be aggregated. The prototype will employ basic automatic motion and use sensory data to avoid collisions. Outside of how the steering will function, this agent will have little to no mechanical similarities to the final design but will be sufficient to implement and test our scope of work.

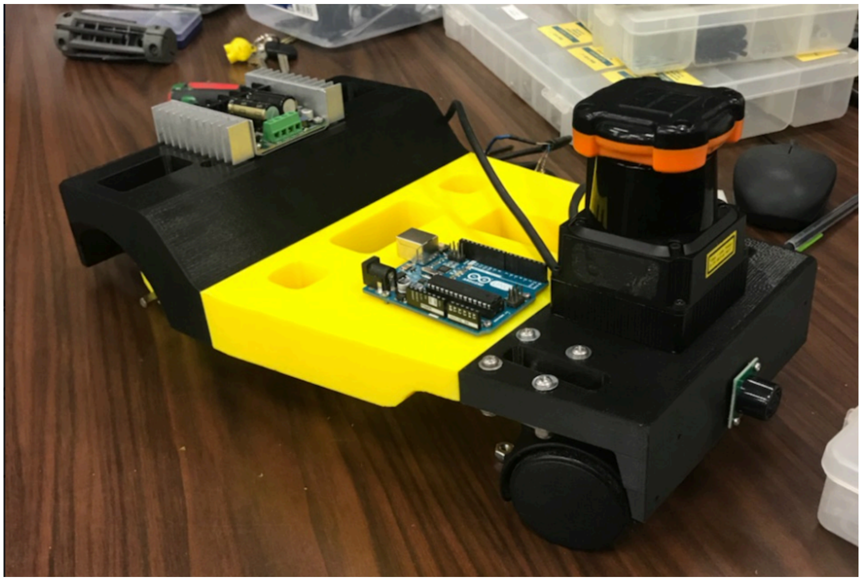


Figure 30: 3D Printed Frame

The prototype was crucial in proving proof of concept. It was equipped with a set of servos, a motor controller unit, an Arduino housing an ATmega328p chip to act as our main PCB, and a raspberry pi. A small-scale autonomous navigation script was written on the Pi and the prototype was able to detect and avoid objects. This same setup will be utilized in the Fall once the real frame of the buggy is built.

7.3 Alternate Design Ideas and Concepts

Several combinations of sensors and cameras were considered to form different subsystems. Therefore, the following sections will discuss ideas on how we wanted to develop the project but could not due to various reasons that will be identified.

Initially we were planning on implementing a computer vision software with the lidar system. We got inspiration to initially do this from other various projects on the internet. After realizing the cost of a LIDAR system, we had to scrap this idea and use simpler sensors combined in unique ways. We may be able to keep some type of computer vision implemented into the project once we arrive in our second semester of senior design. This would be great because it allows for the system to easily create boundaries to navigate through. Computer vision also creates a fantastic display for people to watch, which gives are vehicle the opportunity to have a user interface.

For this project we strongly considered using two batteries for the project. One battery would drive power to the system while the other battery would be charged by the solar panels. When the battery that was driving power to the system had to be replaced we would then switch over the newly charged battery in its place. Then resume charging the newly depleted battery with the solar panels. Although a great initial idea to effectively deliver power to the system and charge the batteries, we ran into complications in which we would had to swap the batteries between each other. Also, this idea of a dual battery was out of the budget in the end. For future projects this is defiantly something worth researching.

7.4 Final Coding and Development Plan

With the abundance of resources and tools available to software developers, it was determined by the CS team that system development and testing will be best performed through simulation software. These tools are purposefully tailored at mimic natural environments for a behavior-based robot, which will generate a realistic demonstration of how the beach buggy will operate in a real-world setting. Simulation also allows for more flexibility and accessibility due to its ability to easily alter tests parameters in order to obtain a quick experimental data. The CS team is confident that the amount of time spent on learning the tools will be compensated in quicker and more effective development and testing stages.

Once a well-tested foundation of software is available, uploading it on a scaled-down prototype may be prudent to develop and test the telemetric system and to demonstrate the overall systems progress in a real-time setting. The result may reveal unforeseen design flaws requiring logic and parameter changes, which may bring the team back to the simulator for further development and testing of the final beach buggy's software on the large-scale prototype. With this possible outcome in mind, a smaller, cheaper prototype would still be sensible to construct during the earlier stages of development for the purpose of demonstrating the system's capabilities while the real prototype structure is still under development. As environmental and autonomy testing becomes more complex, the software should be tested on the ME team's structural designed prototype for a more accurate representation of the system's behavior. This is crucial for the successful development of the final project, as the transmission of data cannot be accurately simulated in a virtual environment.

8.0 Project Prototype Testing

This section will review the environments in which both hardware and software components will udergo testing as well as discuss the accessible tools and equipment that will be used by the team for testing purposes. Additionally, the procedure for hardware and software specific testing will be defined. Each component will be tested individually for proper functionality before being integrated into a system. Software developed both the CS and CpE teams will be incorporated with hardware components and undergo unit, integration, and system testing before conclusively reaching acceptance testing. The intended plans for the project prototype testing process is generalized in the figure below.

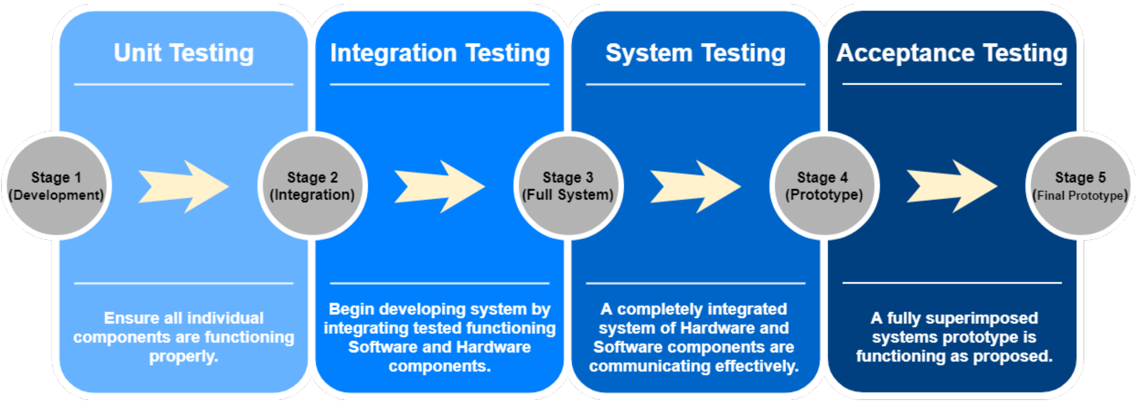


Figure 31: Project Prototype Testing Method

8.1 Hardware Test Environment

Our project has various hardware components interlaces into a system which is designed to detect and avoid objects, persons, and the ocean while navigating the stretch of the coast in a set amount of time. The hardware will be tested individually as well as when it is integrated in system, in both inside (In-Lab) and outside (Field) environments. We want to perform unit testing to ensure that the hardware component is working before integrating it into our system. This will instill the team with developmental practices and general troubleshooting procedures.

By testing indoor and outdoor conditions we will be able to isolate the effects of outdoor elements on certain parts of the system. The sensors and camera are expected to have the most drastic affects when comparing the results of the two environments. With the unknown effects of the Sun's bright natural light, the effectiveness of these devices would need to be thoroughly tested.

Another device that may be affected by natures elements is the Raspberry Pi and ATmega328. These devices are subject to overheating and malfunctioning when exposed to Florida's hot, humid, and occasionally wet weather. The placement and protective covering of these microcontrollers will need to be tested for all possible outside elements.

The solar panel must be tested outdoors to ensure that it is working properly and effectively converting solar energy into usable electricity that can be stored in a battery for future use.

The system will be tested both on and off the beach buggy. Since our mechanical team has yet to develop the buggy itself, most of the sensor testing will be performed off-vehicle. A substitute vehicle or vessel may be improvised when necessary until the beach buggy is ready for integration. The substitute model will fabricate the movement and communication of the future apparatus to the best of our ability with our provided budget.

8.1.1 Component and System In-Lab Testing

As students of the College of Engineering and Computer Science at the University of Central Florida, we have access to an abundance of testing resources and environments.

There is a lab referred to as the Senior Design lab with 10 stations that are equipped with the following:

- Tektronix MSO 4034B Digital Mixed Signal Oscilloscope, 350 MHz, 4 Channel
- Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply
- Dell Optiplex 960 Computer

This is the main environment that will be used for testing both our components and systems.

Figure 32: UCF Senior Design Lab

Additionally, we have access to the “Robotics Club at UCF”’s lab and will be completing component and hardware system testing there.

8.1.2 Component and System Field Testing

Our group plans to implement many forms of field testing for the autonomous system. In order to replicate the environment, we plan to recreate a track or sandbox that can model the beach that our system will be expected to perform on. We will recreate this type of sandy environment to test for sensory feedback on the reflection of the sand and shadow effects of the sand. This will help us to improve the navigation of the system by letting it become familiar with the beach and inevitably ignore any feedback the sand would negatively present.

Additionally, being able to test the motors as they traverse a sandy environment will let us understand maximum power requirements that motors would need. In turn this will give us more insight into how much battery storage we will need, as well as how many watts may need to be delivered in high stress instances. Other field testing may need interaction at a real beach for wide area feedback. We do not possess the funds to recreate a large-scale environment for all of the field testing.

We will attempt to test the system on multiple days with hopefully different weather forecasts, this way we can see performance response in multiple scenarios.

With this field testing combined with the in lab testing appropriate programming will be implemented to guide the system with as much precaution as possible.

Most of this filed testing was not required because of our parts selection.

8.2 Hardware Specific Testing

Each hardware component will be tested for proper functionality prior to system integration. The battery and solar panel will be tested for power distribution, ensuring each component receives a safe range of voltage for its specifications to prevent any under powering, damaging or overloading. The battery will be tested to verify it can effectively hold a charge and storage excess power generated by the solar panel. The solar panel will be tested to verify its production of electricity is sufficient enough to power all necessary device. The camera, sensors and microcontrollers will be tested for proper functionality before the start of development. All hardware components will be tested periodically to ensure that they are up to par when it is time for development or integration.

8.2.1 Microcontrollers

This section will describe the procedures in which the microcontrollers will be tested in order to verify proper functionality before the start of development or attempt to integrate into a system.

8.2.1.1 Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B will be tested for proper general functionality per the device's specifications.

Verify the following:

- Successful power up
- Micro-USB power adapter supplies 5V
- Successful boot up and display on computer screen using HDMI
- Proper terminal and file system functionality
- Successful connection using each USB 2.0 port
- Successful Wi-Fi signal
- Switched Micro Power Source supplies up to 2.5 Amps
- Check for temperature to prevent overheating

If the Raspberry Pi 3 Model B does not meet its proper design specifications, steps will be taken to replace or repair to component.

The Raspberry Pi 3 Model B utilized for this project passed all tests.

8.2.1.2 ATmega328P

The ATmega328P will also be tested for proper general functionality per the device's specifications. This microchip can be tested in multiple ways, but the simplest method may be through an Arduino Uno.

Verify the following on the Arduino Uno:

- Successful power up via USB connection
- Successful power up with external (non-USB) power connection
- External power supply is supplying from 6 to 20 volts
- 5V Pin is supplying 5 volts
- 3V3 Pin is supplying 3.3 volts with maximum current draw at 50 mA
- Internal pull-up resistor is of 20-50k ohm
- Digital Pins provide/receive at least 20 mA, maximum of 40mA
- LEDs indicators light up

Then run simple program such as blinking an LED once per second. If an Arduino Uno is not accessible, a simple circuit may be designed on a breadboard mapping to the ATmega328P and programmed to run the same program. If the

ATmega328P does not meet its proper design specifications, steps will be taken to replace or repair to component.

We completed this testing. All functions were working properly. We also found that in order for the ATmega328p to be programmable, we had to bootload the chip. This required using an Arduino Uno as a programmer to burn code onto the chip to allow it to be programmable.

8.2.2 Raspberry Pi Camera

The Raspberry Pi camera will be tested to ensure it communicates properly with the Raspberry Pi. We will also use the OpenCV framework to ensure it can be used to image processing and computer vision. Simple open source programs are available that we will run on through OpenCV to ensure functionality before we are able to begin testing the program designed specifically for this project. Testing and development will be led by the computer science team with assistance from us.

We ultimately did not handle this scope.

8.2.3 Sensors

In the following sections we will discuss the planned testing operations that we will perform on the PIR, ultrasonic, and camera modules. Each section will cover the basic input/output testing's as well as more specific testing needed for each device.

8.2.3.1 PIR Sensors

The PIR sensors will undergo both indoor and outdoor testing. We will be analyzing the range, consistency of vision, angle detection range, power consumption fluctuations, idle state conditions, full load conditions, and any other sensory testing rigors that may come to our attention. We will also be testing the sensors in fixed positions under different shadings, sun intensities, angles, and angle scopes. We will be testing the sensors for soft and hard body detection as well. The sensors will then undergo testing on our prototype model with all of the same pervious tests reperformed. We believe that the more testing we can perform with the robot the more effective we will be able to interpret its provided information. We will also test the PIR sensors when its range is limited and attempt to optimize its features to suit our needs as close-range sensors.

We ultimately did not adopt PIR sensors for our project.

8.2.3.2 Ultrasonic Sensors

The ultrasonic sensors will undergo both indoor and outdoor testing just as the PIR. Even though these sensors already have recommendations for what environments they are suited for. We need to completely understand the sensors and anticipate their outputs for what is actually there. Therefore, we will be doing fixed position testing in which we test for range, consistency of vision, angle detection range, power consumption fluctuations, idle state conditions, full load conditions, and any other sensory testing rigors that may come to our attention. We will be testing the sensors for soft and hard body detection as well. We will be testing the sensors under different shadings, sun intensities, angles, and angle scopes. The sensors will then undergo testing on our prototype model with all of the same pervious tests reperformed.

We completed this testing for our selected ultrasonic senors.

8.2.4 Power Distribution

The power distribution of the system has been the most challenging part of the project. Because of this we will be doing a large amount of testing on the system while under various load ranges from full load to no load. Additionally, we will perform these full load testing multiple times to gather independent data. These independent tests will provide us with the information across multiple power loads and give us a trustworthy expectation of the systems power consumption.

8.3 Software Test Environment

The communication between our photovoltaic system, sensors, and microcontrollers is an essential part in making our beach buggy successfully autonomous. Our detection system must be able to sense possible obstacles or figures and avoid them in a timely manner. The sensors need to send continuous raw data to the microcontrollers where the information will be interpreted to determine whether an object or obstacle is present or clear for forward movement. If a stationary or moving object is detected, the Raspberry Pi will determine the action that is required as a response, whether that mean turning the vehicles direction, slowing down, or coming to complete stop. If this process is not executed successfully, the data perceived by the sensors can be rendered useless and vehicle will fail to be autonomous. Our software test environment will primarily involve developing and testing the communication between our sensors, ATmega328 microcontroller, Raspberry Pi, and the charger controller connected to the motor, while incorporating the ROS framework. Appropriate situational responses will also need to be determined upon receiving data from the sensors and ROS will be the middleware that provides the vehicle with virtual commands for the responsive behavior. Software version control will be done using the tool GitHub. This tool allows the software team pull from a single repository to manage changes or updates to source code while programmers work on different tasks simultaneously. Software testing will occur under several environments, such as desktop/laptop, laboratory, and field/beach environments.

8.3.1 Desktop Environment

The programming team will be developing software on either their own personal computer, a laptop designated for development in the Robotics Lab, or a desktop stationed in the Senior Design lab. The integrated development environment (IDE) that is accompanied with the Arduino will be used for developing, testing, and uploading code on to the ATmega328. To program the Raspberry Pi, a variety of IDEs, such as IntelliJ, Geany IDE, Adafruit WebIDE, AlgoIDE, or Atom, may be used. The team will also work to incorporate the ROS middleware to execute the necessary virtual responses during development. The CS team is planning to utilize ROS in a Gazebo virtual environment to perform motility testing under this kind of environment.

Ultimately we utilized Arduino IDE and Thonny, a Python IDE to develop on the Pi and ATmega328p.

8.3.2 Lab Environment

As development progresses, the software will be uploaded onto the microcontrollers and the system will be tested in the lab environment similar to the testing environment described in section 8.1.1. Initially, the system will be tested with fixed objects, and upon successful detection, it will be tested with moving objects. After the goal of object detection is completed, the sensors and microcontrollers will be mounted on miniature mobile vehicle in order to confirm successful avoidance calculations, commands and communication. The autonomy system will be thoroughly tested and repeated to ensure proper functionality is observed. The Raspberry Pi should analyze the continuously streaming data from the sensors and be able to delineate when an object or obstacle is encountered. Subsequently, the Pi should logically determine the appropriate response to avoid the obstacle and command the vehicle's movement though the tools provided by ROS.

We implemented a small-scale version of object detection using Python instead of ROS. Our main intentions were to prove proof of concept and allow the Computer Science team to implement ROS and final object detection.

8.3.3 Field/Beach Environment

After the autonomous system proves to be operational in the laboratory, the same system will be tested in an environment similar to the field testing environment described in section 8.1.2. Adjustments to source code is expected to accommodate for outdoor conditions. A substitute sandbox environment may be necessary for quick spot testing to save our team transportation resources and valuable time. A nearer beach-like environment may also be used, such as a volleyball court or Lake Claire's man-made coast. It is important to replicate and test on an environment that will have similar terrain as Daytona Beach for more recreate similar errors that may occur at the beach and produce more accurate results.

8.4 Software Specific Testing

This section will discuss the software specific testing that was done in conjunction with the computer science team assigned to the project. The software specific testing includes individual unit tests, integration tests, system tests, and acceptance tests. The unit tests will test to ensure a program is working for a specific component; the integration tests will test to connect already approved component software together; the system tests will test whether the integration components work successfully in a full system; and the acceptance tests will test the final software version while it is incorporated in the complete vehicle prototype.

8.4.1 Unit Tests

Most unit tests can be performed on a personal laptop or desktop computer by a single individual. Unit testing consists of dividing your software application into 'units' that can be tested independently for proper functionality and results. Its purpose is to ensure that the program is performing as it is intended to function as well as to diagnose resulting errors or segmentation faults. Additionally, it may reveal faulty logic or missed cases in source code that may been overlooked. The CS team is planning to perform motility testing by utilizing ROS in a Gazebo virtual environment, while executing early image processing tests, such as basic movement detection, using OpenCV with a webcam. Unit testing for analyzing and processing sensory data will need to be physically performed. Initial unit tests will need to be executed on an isolated sensor component, such as by attaching a sensor to a microcontroller and collecting raw data through a connected laptop. These physical tests will naturally move to being connected to physical prototype in place of a laptop in further developments.

Unit tests were completed as the project was developed.

8.4.2 Integration Tests

As software surpasses unit testing and becomes ready to be incorporated into a system, integration testing will be performed. Integration tests ensure that the units are incorporated correctly and are communicating and functioning as proposed. These tests will be unable to utilize external software alternatives and will be entirely performed on vehicle prototype. The test essentially involves the successful communication of a series of devices, where more components may be added as they become available, in order to ultimately become a fully functional system.

Integration tests were completed as expected by ensuring communication between all applicable devices.

8.4.3 System Tests

System tests will be proposed in the later stages of development. These tests are performed on a completed system to ensure that the system behaves as expected. System tests are implemented after all components of a designed system were incorporated and completed integration testing. The photovoltaic system will be tested to ensure that power is effectively generated and efficiently stored, whereas the sensory system will be testing the system's ability to detect and avoid obstacles by analyzing and interpreting raw data as well as calculating and sending the appropriate commands to the vehicle to change its direction, speed, or path. The systems will be superimposed to overcome various physical environments, primarily outdoor beach terrains with diverse weather conditions. The effective processing of sensory information is one of the crucial task in overcoming the process of autonomy.

We were able to complete a small system test in a sandbox environment utilizing our prototype. The main system tests will be completed in the fall.

8.4.4 Acceptance Test

The acceptance test will be performed on the fully superimposed systems prototype of the solar-powered beach buggy. The final version of the software system will be uploaded on the microcontrollers and all hardware components will be mounted onto the vehicle. The autonomy system will be incorporated with the photovoltaic system and will be tested for functionality in the field. This test is truly a trial run of the designated course, where the all the requirements described will need to be met. The vehicle is expected to robotically traverse a 10 mile stretch of land from Daytona Beach to Ponce Inlet and return without the need of human guidance or input, while maintaining 3 miles per hour pace in addition detecting and avoid stationary and moving objects, animals, and persons. The product delivered after a successful outcome of acceptance testing will be the final prototype of the project.

This will be handled by the CS team once they have integrated our systems.

9.0 Administrative Content

This section contains all the relevant administrative content related to our project, including milstone tracking and budget discussion.

9.1 Milestones Discussion

A table of completed Spring 2018 milestones is listed below with their corresponding timeline to completion.

Spring 2018 - Senior Design I		
Milestone	**Start**	**End**
Senior Design Project Ideas	1/9/18	1/12/18
Interdisciplinary Project Request Submission	1/14/18	1/18/18
Initial Project Idea (Divide and Conquer) Document	1/23/18	1/28/18
Senior Design Bootcamp	1/28/18	1/28/18
Divide and Conquer Meeting	2/3/18	2/3/18
Revised Divide and Conquer Document	2/3/18	3/11/18
*Transition to Florida Solar Beach Buggy Challenge Project	2/26/18	3/13/18
Research	2/28/18	3/20/18
Table of Contents	3/6/18	3/20/18
Initial Meeting with Mechanical and Computer Science Teams	3/23/18	3/23/18
Draft Document	3/6/18	4/9/18
Design Printed Circuit Board	3/6/18	4/23/18
Design Prototype	3/6/18	4/23/18
Order Components	4/22/18	6/20/18
Final Revisions	4/24/18	4/26/18
Final Senior Design I Document	3/6/18	4/27/18

Table 8: Senior Design I Timeline

* This is the time period when our team officially transitioned to the Florida Solar Beach Buggy Project from a different sponsored project.

Listed below in Table 10 is the timeline for Senior Design II.

Summer 2018 - Senior Design II		
Milestone	**Start**	**End**
Test Components	6/25	6/27
Build Prototype	6/27	6/30
Test Prototype	6/30	7/7
Update Prototype	7/8	7/15
Finalize Project	7/15	7/25
Final Report	7/25	7/30
Final Presentation	7/25	7/25
Florida Solar Beach Buggy Challenge Competition	TBD	TBD

Table 9: Senior Design II Schedule

9.1.1 Defining Project

This phase was initially expected to be completed by February. This phase entails having a complete understanding of what is required of this project and what is expected from our sponsor. To accomplish this, we will have met with our sponsor to discuss budget and requirements. We will also have met with the other students working on the project, as this is an interdisciplinary undertaking, to divide work and establish a course of action. We will be able to definitively say what research, parts, components, and man hours will be needed. Additionally, we will be meeting regularly with our multidisciplinary teams in order to understand physical and virtual requirements and constraints that may limit the projects ability. Our expected time of completion was reasonable but, due to unforeseen circumstances, this phase of the project was not completed until March of 2018. The anticipated start date of this phase was pushing by a month, therefore causing our completion of the definition of the project to be pushed as well. We expected to be able to bridge this gap in the next phase of the project. Furthermore, as complications are discovered altercations to the project may occur.

9.1.2 Researching Project

For this phase, all research was expected to have been completed. This included information about possible components and designs for any physical devices, existing systems, and industry standards. We also hoped to have a full understanding of the requirements and expectations of our sponsor. Researching will help us to gain an understanding of our budget and how to manage our finances. We do not anticipate reaching outside of our budget unless absolutely necessary. If no later research can provide alternate solutions, and we have reached our budgeting limit, we may possibly use our own finances to fund small parts of the project. We initially expected this to be completed by the beginning of May. Further research will be done if needed. Testing and programming in later stages of the project may require research for methods and answer to potential problems that may incur. If research needs to be don't to find solutions to problem later in the project this will be a top priority, as we might not be able to continue with the project if physical and virtual complications arise.

Much research was completed after the initial anticipated end of this phase inorder to account for new situations and complications.

9.1.3 Design

At this point in the project we will be able finish research and plan a preliminary design for our product. This includes all physical components and potential computer coding. We anticipated our design to fluctuate as we understand limitations and requirements that our sponsors and their guests require. We also expected to be working with students in a different discipline, and need to coordinate to ensure our progress, research, and design are compatible. The design and research phase were expected to naturally overlap as we worked with all parties involved to ensure all requirements, technical or otherwise, are met. Our mechanical engineering team will be responsible for providing the physical framework of the project. This includes creating space and accessibility for the electronics and the sensors included. We will be working alongside them in order to advocate any concern or requirements that they need to account for. Lastly the physical design and layout of the electronics is somewhat arbitrary. Although our electronics will be networked together, as the framework of the device is managed and discovered upon creation, we will have to make in the moment decisions on the proper ways to network all of the electronics from the sensors, wires, microcontrollers, batteries, and other electrical components. This phase of the project was initially anticipated to be completed by the end of May. But further design will be implemented as the project continues.

Ultimately, the design changed many times and the design phase officially halted in early July when we were able to resolve all issues discovered through testing.

9.1.4 Prototype

This phase consists of building the prototype and put it through preliminary testing. We anticipated that we could run into obstacles and hope to identify and find any solutions as early as possible. We also anticipated that the building of this prototype would also allow for some preliminary testing that would allow confirmation for basic functionality for the product. Upon coordination with the mechanical team we will be able to build a prototype for preliminary testing. This will help us to model navigation and object detection. The prototype will be a fraction of the actual sized project and will be altered according to design requirements we have, and any requirements not yet discovered. This was initially anticipated to be completed by the end of June 2018 but will be completed before then as to get as much testing in as possible. We hope to have the 3D printed prototype model finish by the end of April, and ready for testing by the beginning of May.

The 3D prototype was completed by the beginning of May and we were able to handle some preliminary functions with it. Ultimately, most of the testing with the prototype tied into our next phase and was completed once other aspects of the project were ready to be tested.

9.1.5 Test

During this phase, we continued testing and evaluation of the product and its system. By this time, we initially hoped to focus on optimization of the product and creating solutions to any remaining problems. We hoped these tests would include a complete automated test without human interaction of the system, as well as a human machine interaction. Testing will be performed without a prototype, with a prototype, and with the final model. This will help us to gain multiple perspectives of the project and hopefully allow for intuitive ideas to come forth. This was initially anticipated to be started by May 2018 and completed by the end of July 2018 but will continue into the final months of the project as to ensure completeness of the system.

Our original assumptions of this point in the timeline are correct. We have finalized testing and were able to demonstrate our project.

9.1.6 Integration

Though not required by senior design standards, the last milestone of our project will be to assist the ME and CS team integrate our solutions. We will do this through August of 2018.

9.2 Budget and Finance Discussion

The budget for the autonomous solar-powered beach buggy is $2,000 divided amongst each discipline assigned to the project. With our team's essential materials alone, the budget is extremely limited. A significant struggle for our team lies in the debate of cost vs quality. We need our components, which includes sensors, batteries, microcontrollers, etc., to be powerful enough to develop an effective autonomous system for an outdoor environment without overtaking the entire budget. The other students affiliated with this project also need to be allocated enough funds to complete their goals and objectives as well.

Because the properties affected by these constraints greatly affect how effective the buggy will be at carrying its weight, the effectiveness of its object detection, and its ability to localize, our team has constructed a table to compare the various builds and the estimated costs associated with them. These builds are not absolute in that various components may be picked for upgrading from a lower tier. This chart, shown below, allows the team to view the various alternatives available at given costs if there was enough budget to improve on certain components. Additionally, this chart allows the team to present an argument for a greater budget, so that they may build a more effective, reliable, and efficient beach buggy by utilizing features that would greatly improve the motility and behavior of the buggy.

	Budget Build		***Intermediate Build***		***Optimal Build***	
Frame	Steel, 90-degree plates	$500	Steel, rectangular tubing	$700	Aluminum, round tubing, 3rd party welding	$1,200
Suspension	Struts, coils	$200	Struts, coils	$200	Struts, coils	$200
Tires	Basic Tires	$60	Basic Beach Tires	$120	Premium Beach Tires	$350
Motors	2750 RPM Electric Motor	$300	2750 RPM Electric Motor	$300	2750 RPM Electric Motor	$300
Sensors	2 Raspberry Pi cameras, 4 Ultrasonic sensors, GPS	$300	2D LIDAR, Camera, 4 ultrasonic, GPS	$1,400	3D LIDAR, HD Camera, 4 ultrasonic, GPS	$3,000
Solar Panels	ECE Panels (Bulky)	Donated	Commercial Panels	$700	Flexible Panels	$1,000
Battery	50 Ahr Lead Acid	$100	100 Ahr Lead Acid	$250	100 Ahr Lithium Ion	$1,000
Controllers	Raspberry Pi 3, ATMega 328P PCB Circuit Board	$250	Raspberry Pi 3, 2 ATMega 328P PCB Circuit Board	$300	NVIDIA Jetson TX2, 2 ATMega 328P PCB	$1,000

					Circuit Board	
Total Cost	**$1,710**		**$3,970**		**$8,050**	

Table 10: Budget breakup

The budget is subject to change as the teams' needs or priorities change, or unanticipated obstacles arise. Due to our restricted budget we will not be able to set aside an amount for unexpected potential expenses. The teams will have to carefully spend and record their expenses as well as revisit the budget often to ensure there is no overspending. The financial supply is incredibly limited considering the scope of the project.

Our team ended up maintaining within budget by subsidizing part of the project ourselves. With redesign of PCBs and a switch to building a charge controller, we were unable to remain within this allotted budget.

10.0 Additional Contributions

This section will describe the contributions that the other groups are adding to this project. After having conversed with the other groups and being provided ample design specifications we will attempt to summarize contributions without passing on specifics. Overall, this section will bring together the entirety of the project to describe it in specification.

10.1 Mechanical Engineering Team

The Mechanical team was responsible for designing and building the chassis of the vehicle, responsible for the motor research, tire research, and other mechanical topics for the project.

For all of the decisions that had to be made for the project, the mechanical team utilized a Pugh matrix. This allowed them compare all of the options for a certain aspect of the beach buggy vehicle. These aspects included criteria, importance weight, and rating scale. The rating scale was used in combination with the importance factor to develop a weighted rating for each criterion. These criteria weighted ratings were then summed and total to represent an overall value for that concept. They used this method for quantification of the options between selection and a comparison that will allow for the best decision to be made.

10.1.1 Chassis Design

The choice of materials used for the construction of the beach buggy were greatly influence the project budget and overall weight, so they have been selected carefully. For this project, steel, aluminum, titanium, and carbon fiber were considered. The mechanical team wanted for the frame to be affordable since the budget for this project is very limited and there are many components needed. Additionally, it is very practical for the beach buggy to be lightweight for obvious reasons that include less power to drive the vehicle itself. This is also necessary because the buggy is not being able to generate sufficient power by only using solar panels only. Again, a lighter buggy allows for less power to be demanded from the motors to move it through the sandy terrain.

10.1.1.1 Strength of the Vehicle

Integral strength is the next most important characteristic. The buggy must be capable of carrying a load and not failing structurally. The mechanical team also recognizes the difficultly in manufacturing a complex frame and therefor intends to make the frame simple but durable. This includes machinability and weldability. The mechanical team has put less emphasis on the durability noting that it is less of a priority due to the product only being used in testing and the day of the competition. From the research conducted, they concluded that steel and aluminum are both very capable materials for this purpose, but steel is slightly more favorable mainly due to its low cost and high strength.

10.1.1.2 Geometry of the Vehicle

Initially the mechanical team thought of three options that stood out above the rest when considering the main chassis geometry. The three chassis geometry design options included the X-beam chassis, the Ladder beam (aka Grilliage) Chassis, and the monocoque chassis. Each one of these offers an abundance of both advantages and disadvantages when diving and operating. With that being said, out of those advantages and disadvantages, what mattered most to the beach buggy project was that the design was low cost, easy to manufacture, had an adequate longitudinal stiffness, and had sufficient torsional stiffness. The mechanical team concluded that the X-beam chasses has amazing torsional stiffness and longitudinal stiffness due to its design but falls a little short in the areas of cost and easy manufacturability. The monocoque chassis has extraordinary torsional stiffness as well as longitudinal stiffness, but it is way too expensive and very difficult to manufacture with current resources. Thus, the Ladder beam chassis are very well rounded and sufficient in all areas which is why it was chosen as the main geometry type for the solar powered beach buggy. In reference to our project this type of ladder beam design effects where and how we can mount the electrical equipment. Each of the three different designs offers unique ways to mount the equipment. The ladder beam design will be re analyzed when electrical mounting becomes prevalent. Overall the electrical equipment will be housed in a dry concealed area, where it will not suffer from air temperature, water, sand, wind, and other unexpected environmental hazards.

10.1.2 Wheels

This paragraph will talk about the implementations that went into wheels for the vehicle. The wheels have little impact on the factors that relate directly to electrical except for some forces that create extra power consumption. None the less this short paragraph will help explain the model of the system better.

The wheel material is an important factor into the design for the Florida Beach Buggy, as it will ensure all power generated is not wasted through slippage. The material that the mechanical team has chosen must have characteristics of a high friction coefficient, large elasticity, low cost, water resistance, and light weight. These conditions combined will provide a suitable criterion for a wheel or tire, ensuring it can hold up to environmental conditions and stresses applied, while remaining at a reasonable cost. The criterion characteristics independently add to the whole importance of the vehicle. The friction force coefficient will determine the amount of slippage that may occur between material and the sandy terrain. Elasticity will affect the flexibility of the wheel or tire. This is important because the terrain is rough and may provide unknown obstacles. The buggy should ideally have a material that is ductile and not prone to cracking. When considering cost, it is important to find a suitable material that has a low cost, whether it be low cost due to large manufacturing, or cheap material cost. As far as water resistance, the wheels or tires must be water resistant and suitable for a salt and sand environment. Lastly, the weight of the material should be low, as the beach buggy is desired to be less than 300 lbs.

10.1.3 Motors

Probably the most impactful component of the system as they relate to electrical aspects are the motors. This section will explain the design process that implements the motors into the system.

When considering selection of the electric motor for the buggy, it is important to outline the basis upon which decisions were made. The mechanical team worked on this solution while taking into our considerations. We made clear that a 12v or 24v motor would be best and asked them Pick motors that would achieve the minimum speed and use the lowest amount of power at the same time. The criteria that controlled the weighting of each motor design followed specifically to the beach buggy's application and environment. When examining the needs analysis of the buggy, the motor must be at a competitive price for budget purposes. The motor must offer high efficiency relative to its ratings, instant torque at zero RPM, simple speed controlling ability, as well as high durability and heat dissipation. Since the motor will be in the open and not covered with any sort of surrounding, cooling trough natural convection should be adequate, hence its low importance weighting. Instant torque is an important characteristic because the force needed to overcome the static inertia of the buggy will need to be high. The options considered for the motor are AC, DC, brushed, and brushless configurations.

The mechanical team decided that it is in the team's best interest to choose a brushed, DC motor that requires an external power delivery system. This configuration best suits the low-cost constraint as well as the overall efficiency and build purpose of the product. Additionally, this type of motor works best for our electrical case. Supply a DC motor is easily achieved with a battery and no AC/DC converter is needed otherwise. Overall the mechanical team decided this type of motor was best suited for the project.

10.1.4 Steering

Part of our concern for the project was to tether the steering mechanism with the sensory input. In order to Pick out sensors and interpret potential coding we needed to understand how the vehicle would steer itself. The following paragraph will attempt to explain the implementation of

The steering mechanism that is implemented is vital to the mobility of the beach buggy. The mechanical team decided upon a variety of options that were available and the best considerations were one-wheel steering, two-wheel steering, and torque vectoring (motor control). They rated these options on their ability to meet the criteria of this project. The steering chosen must be low cost due to the conservative budget for this project and the high cost of other parts. Another important characteristic is that the steering method must not add any significant weight to the vehicle. The next most important goal of the design selection is that the buggy remains stable while turning. The steering method chosen must also be reliable and not break or malfunction during operation. After conversing with our group and voicing their concerns, the mechanical team decided the best option to be a torque vectoring style of steering. It is the cheapest and lightest option because it involves no additional components and is controlled through program control of the motors. It is also reliable because there are no mechanism Pieces that can be broken like those found in the other two options.

10.2 Computer Science Team

This section will attempt to integrate the computer science team's additions to the project. Much of their contributions will come during senior design two when we have electrical hardware submitting values that can be coded into sensible software. None the less these are some of the topics we will consider in our portion of the project as well, so that we can transition easier and keep communication simple between groups.

10.2.1 Telemetric Data Transfer

The computer science team is going to introduce telemetric data transfer in order to wirelessly transfer vehicular data to a remote device. This data will then be aggregated and displayed to the user of the device. The telemetric data to be sent to the device includes, but is not limited to:

- GPS coordinates.
- Vehicle speed.
- Battery life.
- Solar energy input.
- Path progression.

The collection of this data will be done through various interfaces between the sensors and the computer within the agent. These interfaces will provide streams of data from their respective sensor for the computer to process. The computer will need to take synchronous snapshots of each sensors data stream and aggregate the data to be packaged, serialized, and sent to the remote device. This device will then unpackage and display whatever data was sent to the user through some interface.

The computer will need some method for delivering the data. As the computer will reside on the agent itself, it would be ideal for this method to be of wireless means. There are a few methods available for interacting with remote devices: a direct ethernet connection, Bluetooth, and by using SSH to connect through a network. While a direct ethernet connection will be useful for initial set-up, it cannot be a serious data-deliverance candidate in the deployment of the agent in real-time due to the reasons previously discussed. Therefore, they will be focusing on Bluetooth and network connectivity for the agents telemetric deliverances.

10.2.2 Software State Machine

The software that the computer science team will be implementing for navigation will be a multi-state machine that has idle and running states. Until any input has been given to the machine, it will remain in an idle Waiting state. When a destination is inputted via the remote device connected to the system, it will begin determining its path using environmental data. Once this path has been defined, the machine will begin traversing to the inputted destination and will enter the Moving & Scanning state. While moving, the camera and various sensors will be searching for objects and environmental anomalies. Upon detection, the computer will determine if the object or anomaly is an obstacle; if yes, then it will attempt to move around the obstacle and determine its new path, otherwise, the agent will continue to move and scan for obstacles. While it is in the Moving & Scanning state, it will have a reference to its current GPS location. This will be used to determine if the buggy is too far off track or if it has reached its inputted destination. If the buggy has reached its destination, then it will return to the Waiting state; where it will wait until a new trajectory has been given. As the vehicle will be traversing to the targeted destination and back again, it will need to receive a destination two times; at the beginning of its journey, and at its original destination. This effect will be automated by creating a round-trip feature that tells the buggy that it will need to return to its original location at the end of its destination. The computer can easily perform this by storing the original location when the inputted destination is given and using it as its new destination once it has arrived at the inputted one.

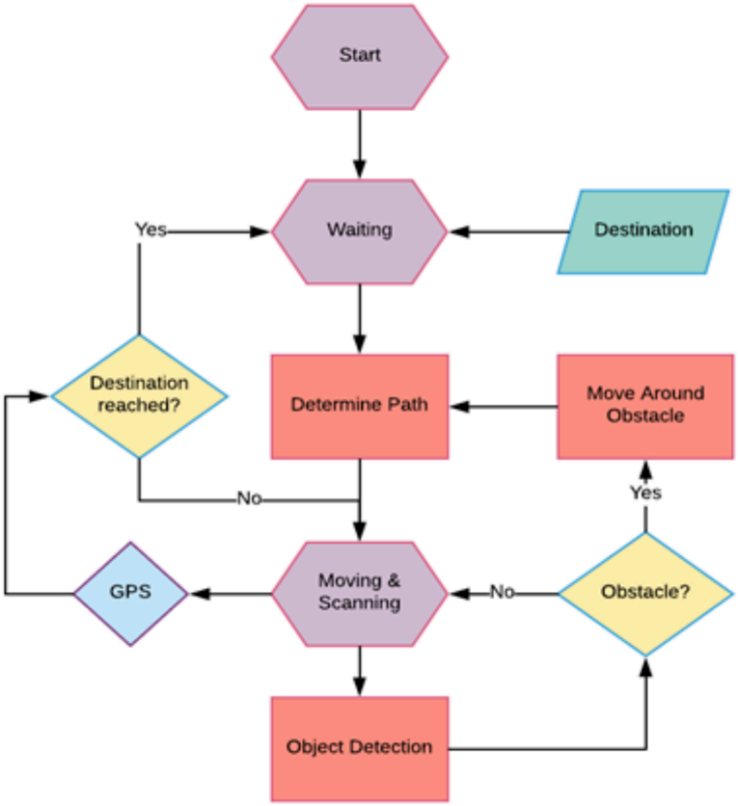


Figure 33: Software State Machine

11.0 Conclusion

The Florida Beach Buggy Solar Challenge poses many challenges. We are having to tackle solar conversion, long ange autonomous travel, and object processing on a large scale in a unique environment with an incredibly limited budget and serious time constraints. Our research has shaped our design and given us the confidence that we will be able to accomplish our scope of the project, which is to build the system that will spearhead navigation as well as the system that will convert solar energy into power that will be used to power our microcontrollers and motors.

We have implemented the sensors we are using for close-range object detection, which sends data through our PCB containing the ATmega328p to the Raspberry Pi 3. The Pi will be the central computing device of the system and will take this information, along with data passed in by the onboard camera, to control navigation in the following semester. The PCB also steps down the voltage input from our 12V battery to 5V in order to serve the ATmega328p, the multiple sensors, the camera, and the Raspberry Pi. Additionally, we have designed a charge controller to serve as the battery management system for the incoming power from our solar panel to the battery we selected. We have also selected a 12V to 24V converter in order to supply power to our motors.

We have concluded that, though we were successful in providing a proof of concept, we would have found more success that we been working on this project in the same time frame as the other scopes. We have done our best to make our implementations adaptable to many situations.

Appendix A: Sources

[1] "Lithium-Ion Battery Charging & Advantages – PowerTech Systems." *PowerTech Systems*, www.powertechsystems.eu/home/tech-corner/lithium-ion-battery-advantages.

[2] "BU-403: Charging Lead Acid." *Charging Information For Lead Acid Batteries – Battery University*, batteryuniversity.com/learn/article/charging_the_lead_acid_battery.

[3] *Lidar-Uk.com*, www.lidar-uk.com/how-lidar-works/.

[4] "Master Plan, Part Deux." *Tesla, Inc*, 21 July 2016, www.tesla.com/blog/master-plan-partdeux.

[5] "Photovoltaic (Solar Electric)." *SEIA*, www.seia.org/initiatives/photovoltaic-solar-electric.

[6] "Photovoltaic Cell." *Photovoltaic Cell - Energy Education*, energyeducation.ca/encyclopedia/Photovoltaic_cell.

[7] *Switching Regulator Basics*, www.rason.org/Projects/swregdes/swregdes.htm.

[8] "Ultrasonic Sensors and High Performance Proximity Sensors I MaxBotix." *MaxBotix Inc.*, www.maxbotix.com/.

[9] "MB1260 XL-MaxSonar-EZL0." *MaxBotix Inc.*, www.maxbotix.com/Ultrasonic_Sensors/MB1260.htm.

[10] "Knowing How a Stereoscopic Camera Works." *Knowing How a Stereoscopic Camera Works - Steve's Digicams*, www.steves-digicams.com/knowledge-center/how-tos/photo-accessories/knowing-how-a-stereoscopic-camera-works.html#b.

[11] "XL-MaxSonar® - EZ™ Series Datasheet." *Maxbotix.com*, www.maxbotix.com/documents/XL-MaxSonar-EZ_Datasheet.pdf.

[12] "About ROS." ROS.org, Open Source Robotics Foundation, www.ros.org/about-ros/.

[13] Wiki." Ros.org, Open Source Robotics Foundation, 01 Feb. 2017, 22:53:41, wiki.ros.org/DevelopersGuide#Copyright.

[14] Thomas, Dirk. "Wiki." Ros.org, Open Source Robotics Foundation, 22 May 2014, 16:14:17, wiki.ros.org/ROS/Introduction#Operating_Systems.

[15] R, Aaron M. "Wiki." Ros.org, Open Source Robotics Foundation, 21 June 2014, 11:53:27, wiki.ros.org/ROS/Concepts.

[16] "Wiki." Translated by OSE ROS Support, Ros.org, Open Source Robotics Foundation, 24 Apr. 2018, 02:29:52, wiki.ros.org/Sensors.

[17] DataFlair Team. "13 Unique Features of Python Programming Language." DataFlair, 30 Dec. 2017, data-flair.training/blogs/features-python/.

[18] DataFlair Team. “Advantages and Disadvantages of Python Programming Language.” DataFlair, 2 Jan. 2018, data-flair.training/blogs/advantages-and-disadvantages-of-python/.

[19] Krill, Paul, and InfoWorld. “A Developer's Guide to the Pros and Cons of Python.” InfoWorld, InfoWorld, 24 Feb. 2015, www.infoworld.com/article/2887974/application-development/a-developer-s-guide-to-the-pro-s-and-con-s-of-python.html.

[20] Rouse, Margaret, and Ivy Wigmore. “What Is Internet of Things (IoT)?” IoT Agenda, TechTarget, July 2016, internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

[21] Morgan, Jacob. “A Simple Explanation Of 'The Internet Of Things'.” Forbes, Forbes Magazine, 13 May 2014, 12:05, www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#5c79fae61d09.

[22] Clark, Jen. “What Is the Internet of Things, and How Does It Work?” Internet of Things Blog, IBM, 17 Nov. 2016, www.ibm.com/blogs/internet-of-things/what-is-the-iot/.

[23] “Raspberry Pi 3 Model B.” Raspberry Pi, Raspberry Pi Foundation, www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[24] “Raspberry Pi FAQs - Frequently Asked Questions.” Raspberry Pi, Raspberry Pi Foundation, www.raspberrypi.org/help/faqs/#softwareOS.

[25] Industries, Adafruit. “Raspberry Pi 3 - Model B - ARMv8 with 1G RAM.” Adafruit, Adafruit, www.adafruit.com/product/3055?src=raspberrypi.

[26] “What Is a Raspberry Pi?” Opensource.com, Red Hat, opensource.com/resources/raspberry-pi.

[27] “ARDUINO UNO REV3.” Arduino, Arduino, store.arduino.cc/usa/arduino-uno-rev3/.

[28] “Arduino Uno.” Octopart, datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf.

[29] “Wide Angle PIR Datasheet.” *Wide Angle PIR Datasheet*, Parallax Inc, media.digikey.com/pdf/Data Sheets/Parallax PDFs/28032_Web.pdf.

[30] “Mini PIR Motion Sensor Module.” *Seeedstudio*, www.seeedstudio.com/Mini-PIR-Motion-Sensor-Module-p-1729.html.

[31] “Sparkfun Open PIR.” *Digikey*, media.digikey.com/pdf/Data Sheets/Sparkfun PDFs/SEN-13968_Web.pdf.

[32] “CMUcam5 Pixy.” Overview - CMUcam5 Pixy - CMUcam: Open Source Programmable Embedded Color Vision Sensors, www.cmucam.org/projects/cmucam5.

[33] “Solar Team Eindhoven.” Solar Team Eindhoven, solarteameindhoven.nl/.

[34] Rankin, A. L., et al. “Daytime Water Detection By Fusing Multiple Cues For Autonomous Off-Road Navigation.” Transformational Science and Technology for the Current and Future Force, 2006, doi:10.1142/9789812772572_0023.

[35] "IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead-Acid Batteries Used in Stand-Alone Photovoltaic (PV) Systems." IEEE Std 1361-2014 (Revision of IEEE Std 1361-2003), Jan. 2014, p. 1. EBSCOhost, login.ezproxy.net.ucf.edu/login?auth=shibb&url=http://search.ebscohost.com.ezproxy.net.ucf.edu/login.aspx?direct=true&db=edb&AN=98472231&site=eds-live&scope=site.

[36] van Rossum, Guido, et al. "PEP 8 -- Style Guide for Python Code." Python.org, Python Software Foundation, 5 July 2001, www.python.org/dev/peps/pep-0008/.

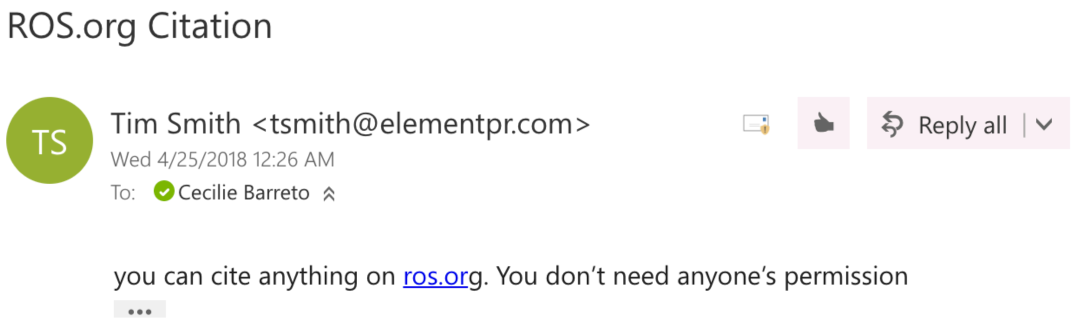
[37] Dixit, Chitrank. "PEP-8 Tutorial: Code Standards in Python." DataCamp Community, DataCamp, 25 Jan. 2018, www.datacamp.com/community/tutorials/pep8-tutorial-python-code.

[38] Volusia County Florida. Drive Safe Play Safe. Drive Safe Play Safe, Lifeguard Headquarters and Administration Center, 2014.

[39] "Welcome to an Engaged Community." Daytona Beach, FL - Official Website - Beach Information, City of Daytona Beach, www.codb.us/index.aspx?nid=407.

Appendix B: Copyright Permissions

All permissions not listed are pending.



ROS.org Citation

Tim Smith <tsmith@elementpr.com>
Wed 4/25/2018 12:26 AM
To: Cecilie Barreto

you can cite anything on ros.org. You don't need anyone's permission

[12][13][14][15]

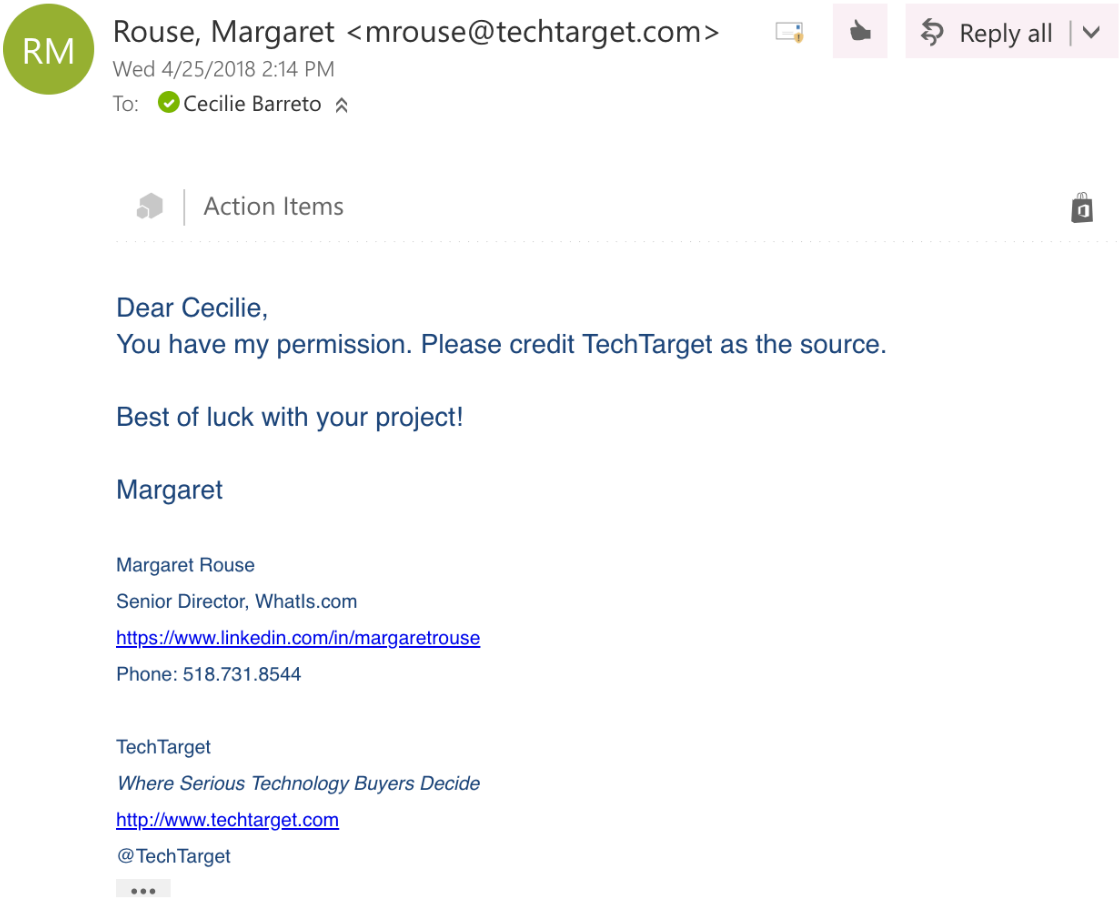
DataFlair Citation Request

Data Flair <info@data-flair.training>
Wed 4/25/2018 1:37 AM
To: Cecilie Barreto

Hi Cecilie.
We would be happy to see our articles in your paper. Do ensure to give credit to the material that you take from our website to DataFlair only.

Also share copy of the paper once it is published so that we can share with our employees as well.

[17][18]



Rouse, Margaret <mrouse@techtarget.com>

Wed 4/25/2018 2:14 PM

To: Cecilie Barreto

Dear Cecilie,
You have my permission. Please credit TechTarget as the source.

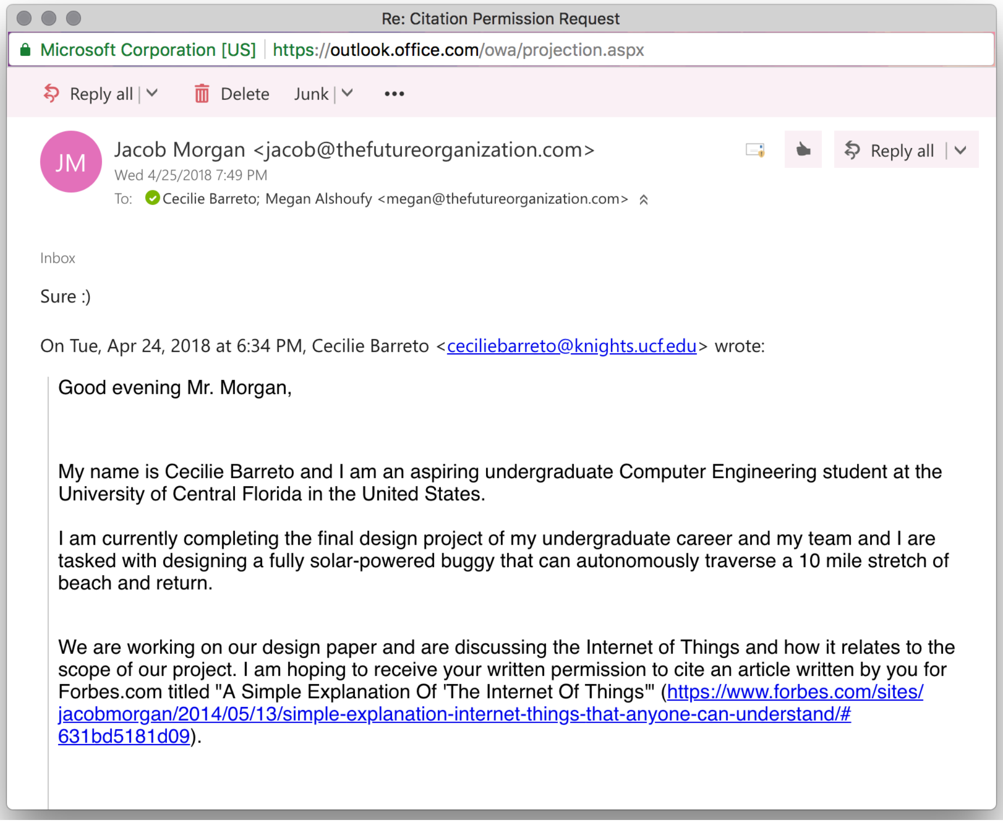
Best of luck with your project!

Margaret

Margaret Rouse
Senior Director, WhatIs.com
https://www.linkedin.com/in/margaretrouse
Phone: 518.731.8544

TechTarget
Where Serious Technology Buyers Decide
http://www.techtarget.com
@TechTarget

[20]



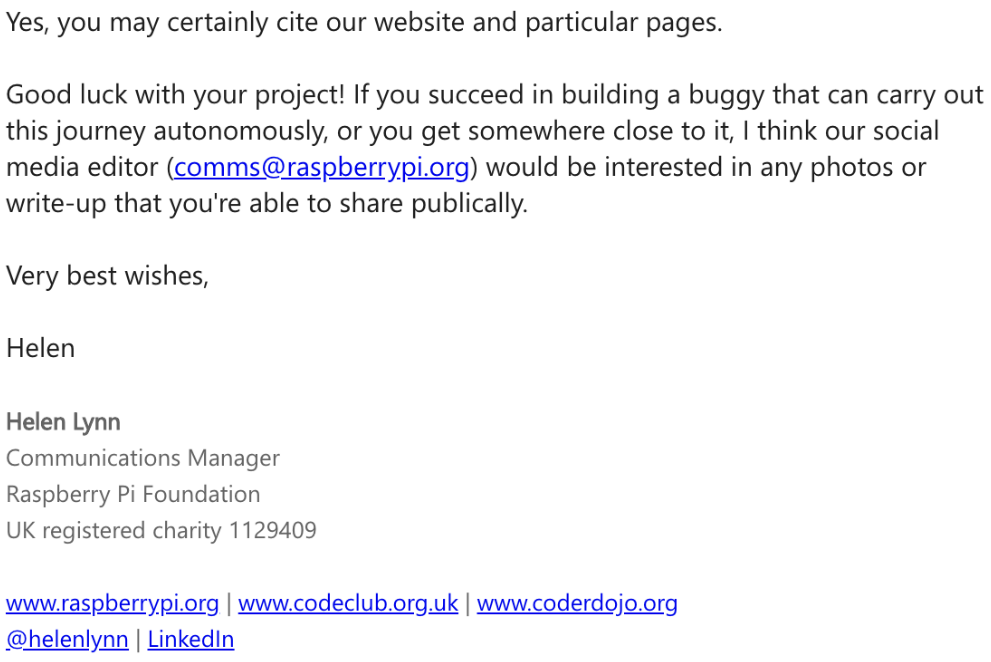
Jacob Morgan <jacob@thefutureorganization.com>
Wed 4/25/2018 7:49 PM
To: Cecilie Barreto; Megan Alshoufy <megan@thefutureorganization.com>

Sure :)

On Tue, Apr 24, 2018 at 6:34 PM, Cecilie Barreto <ceciliebarreto@knights.ucf.edu> wrote:

> Good evening Mr. Morgan,
>
> My name is Cecilie Barreto and I am an aspiring undergraduate Computer Engineering student at the University of Central Florida in the United States.
>
> I am currently completing the final design project of my undergraduate career and my team and I are tasked with designing a fully solar-powered buggy that can autonomously traverse a 10 mile stretch of beach and return.
>
> We are working on our design paper and are discussing the Internet of Things and how it relates to the scope of our project. I am hoping to receive your written permission to cite an article written by you for Forbes.com titled "A Simple Explanation Of 'The Internet Of Things'" (https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#631bd5181d09).

[21]



Yes, you may certainly cite our website and particular pages.

Good luck with your project! If you succeed in building a buggy that can carry out this journey autonomously, or you get somewhere close to it, I think our social media editor (comms@raspberrypi.org) would be interested in any photos or write-up that you're able to share publically.

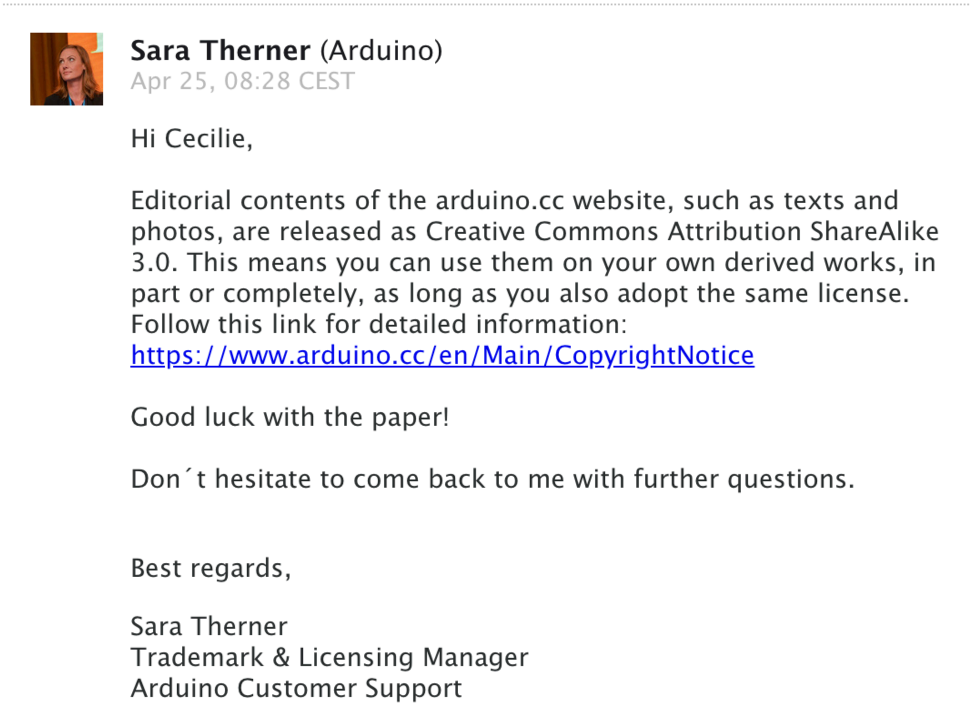
Very best wishes,

Helen

Helen Lynn
Communications Manager
Raspberry Pi Foundation
UK registered charity 1129409

www.raspberrypi.org | www.codeclub.org.uk | www.coderdojo.org
@helenlynn | LinkedIn

[23]



Sara Therner (Arduino)
Apr 25, 08:28 CEST

Hi Cecilie,

Editorial contents of the arduino.cc website, such as texts and photos, are released as Creative Commons Attribution ShareAlike 3.0. This means you can use them on your own derived works, in part or completely, as long as you also adopt the same license. Follow this link for detailed information: https://www.arduino.cc/en/Main/CopyrightNotice

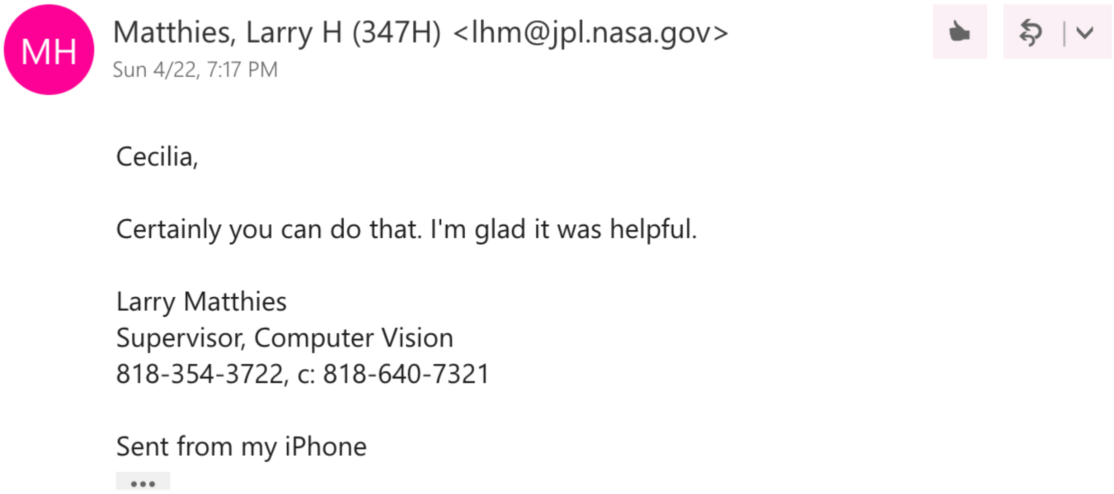
Good luck with the paper!

Don´t hesitate to come back to me with further questions.

Best regards,

Sara Therner
Trademark & Licensing Manager
Arduino Customer Support

[27]



Matthies, Larry H (347H) <lhm@jpl.nasa.gov>
Sun 4/22, 7:17 PM

Cecilia,

Certainly you can do that. I'm glad it was helpful.

Larry Matthies
Supervisor, Computer Vision
818-354-3722, c: 818-640-7321

Sent from my iPhone

[34]

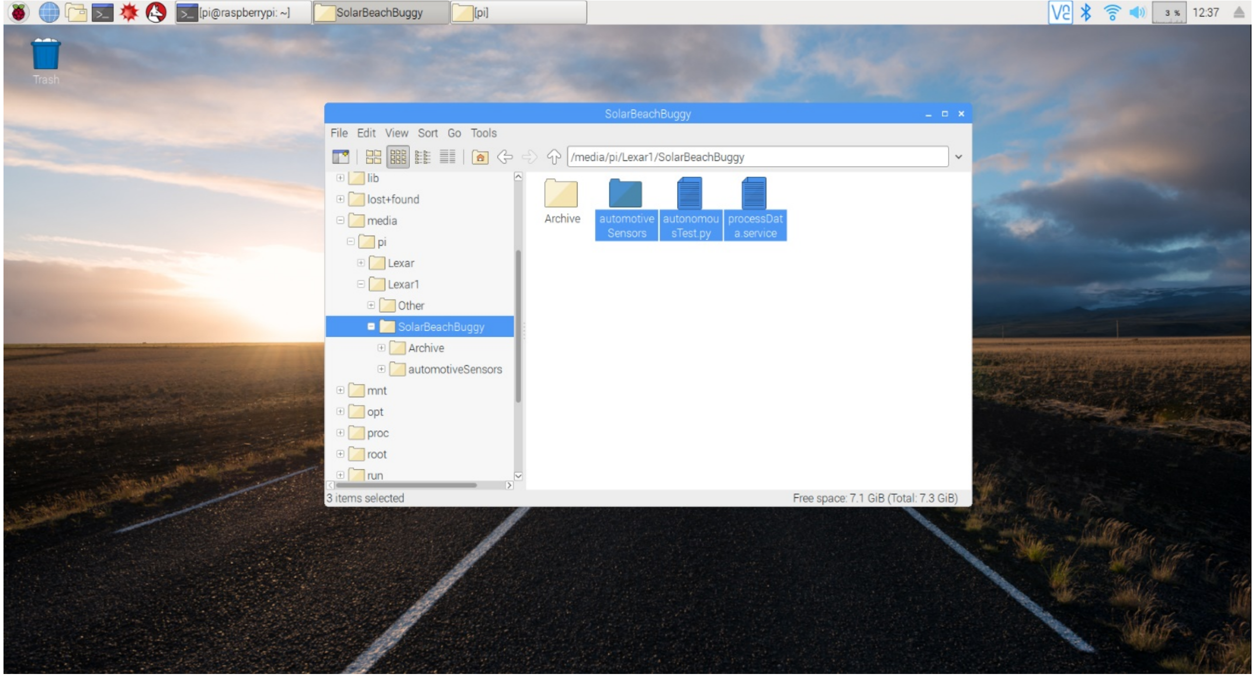
Appendix C: Software Procedures

Set-Up Pre-requisites:

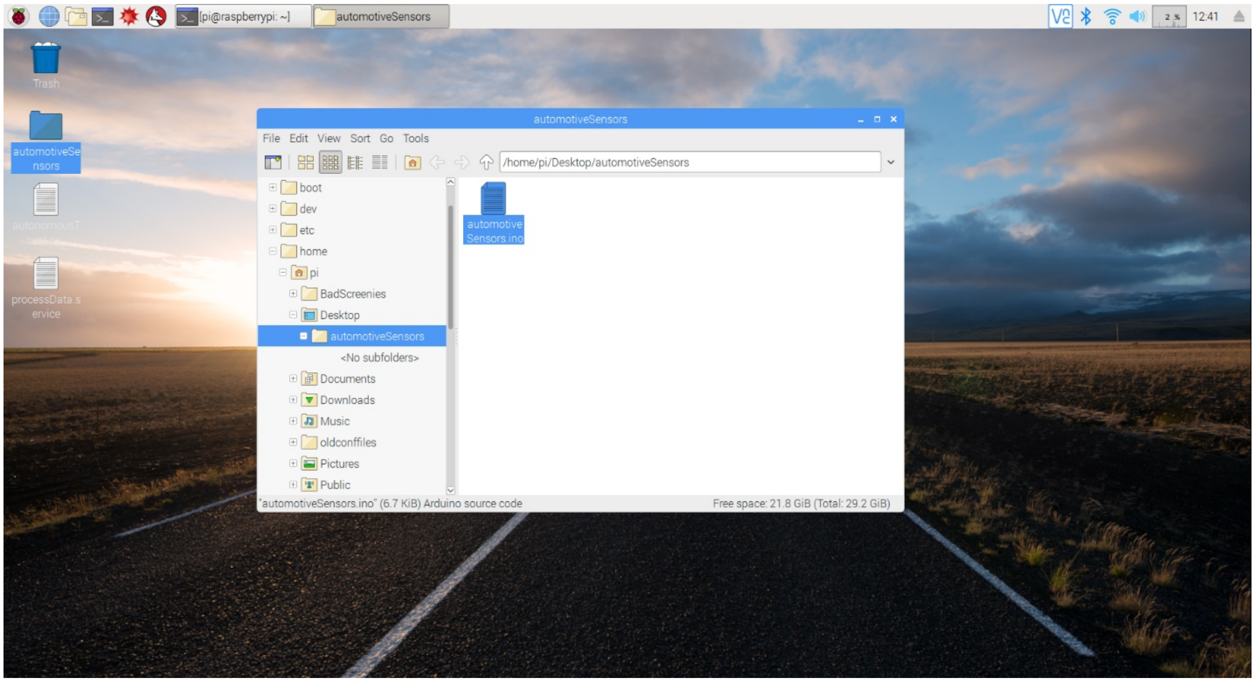
1. Make sure the Raspberry Pi has the following already installed:
    - Arduino IDE
    - Python 2.7
2. Connect the automotive sensors to the Arduino Uno.
3. Connect the Arduino Uno and the Raspberry Pi via serial USB.
4. Connect the Sabertooth Motor Controller to the Raspberry Pi.
5. Connect all necessary equipment to appropriate power sources.

Software Set-Up Procedure

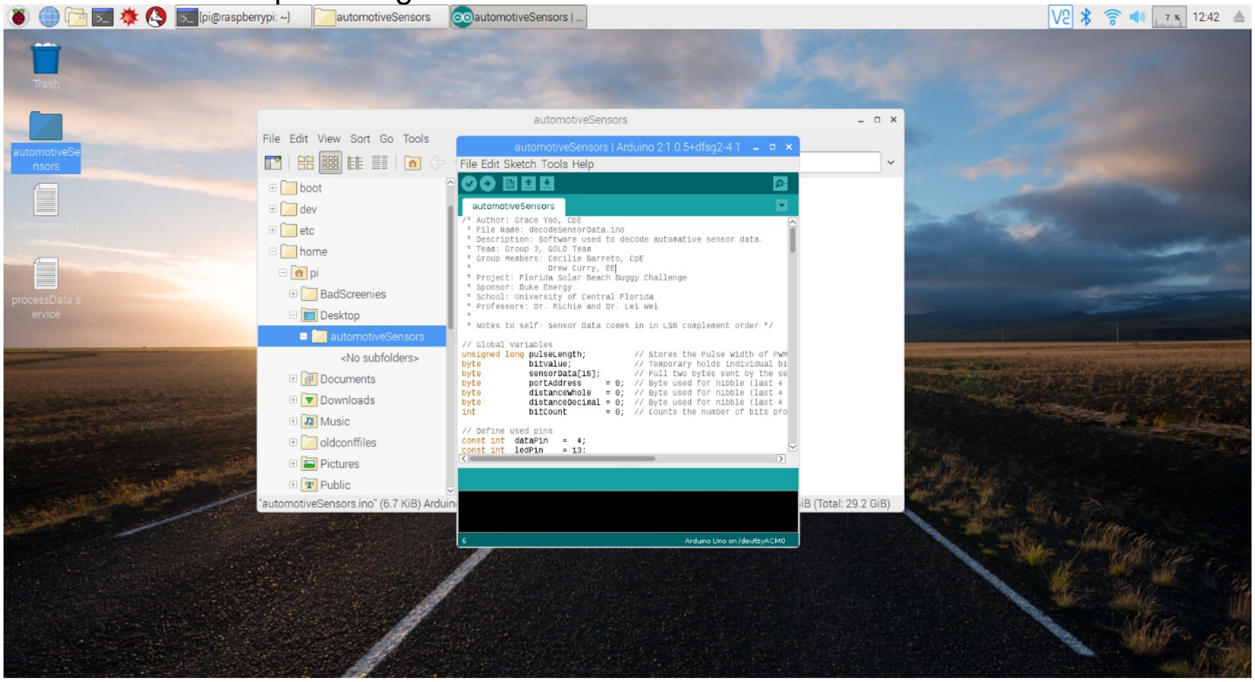
1. Upload the necessary software files onto the Desktop of the Raspberry Pi.



2. Navigate to the “automotiveSensors” folder and double click “automotiveSensors.ino” file.

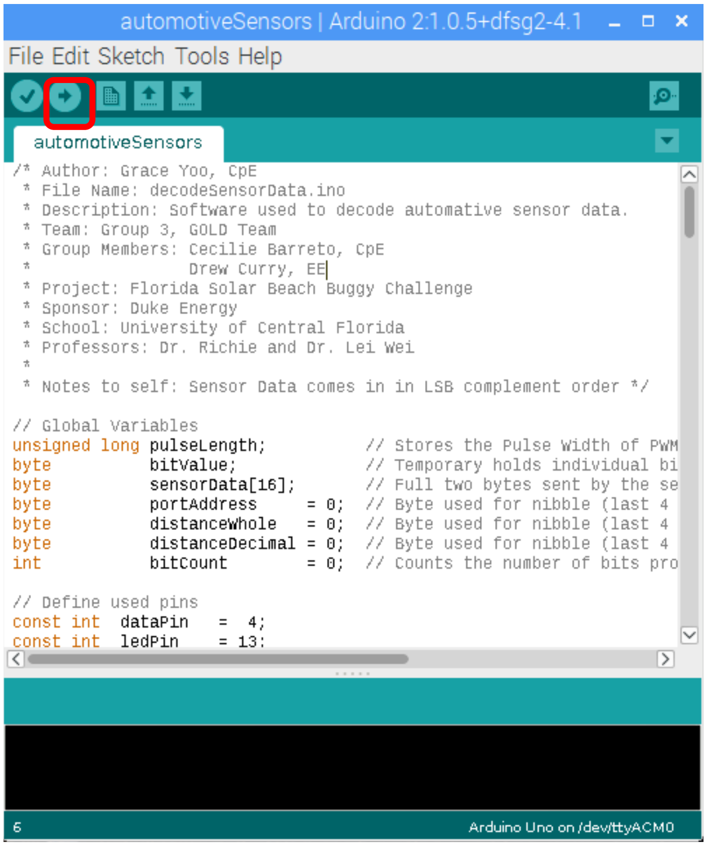


3. The file should open using the Ardunio IDE interface.



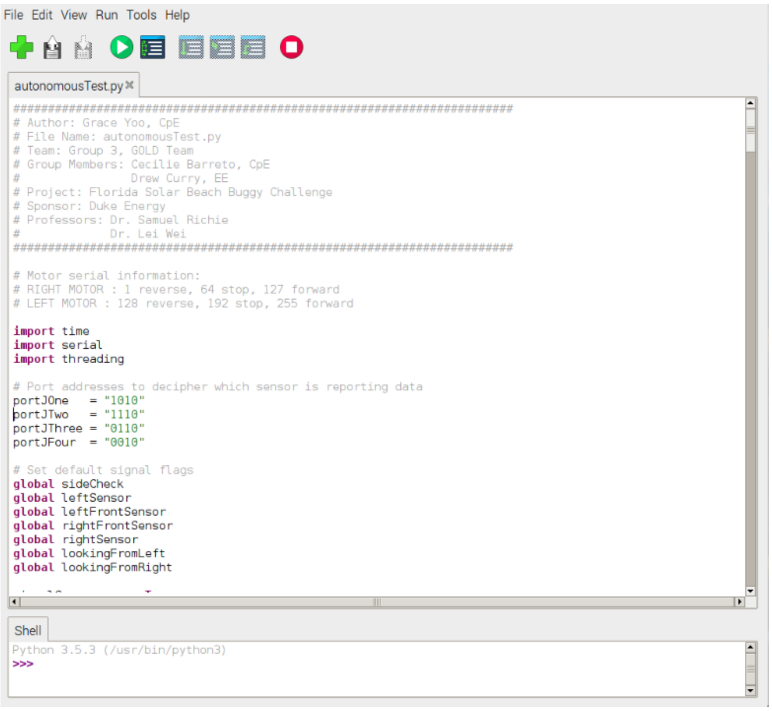
```
/* Author: Grace Yoo, CpE
 * File Name: decodeSensorData.ino
 * Description: Software used to decode automative sensor data.
 * Team: Group 3, GOLD Team
 * Group Members: Cecilie Barreto, CpE
 *                Drew Curry, EE
 * Project: Florida Solar Beach Buggy Challenge
 * Sponsor: Duke Energy
 * School: University of Central Florida
 * Professors: Dr. Richie and Dr. Lei Wei
 *
 * Notes to self: Sensor Data comes in in LSB complement order */

// Global Variables
unsigned long pulseLength;           // Stores the Pulse Width of PWM
byte          bitValue;              // Temporary holds individual bi
byte          sensorData[16];        // Full two bytes sent by the se
byte          portAddress     = 0;   // Byte used for nibble (last 4
byte          distanceWhole   = 0;   // Byte used for nibble (last 4
byte          distanceDecimal = 0;   // Byte used for nibble (last 4
int           bitCount        = 0;   // Counts the number of bits pro

// Define used pins
const int  dataPin   =  4;
const int  ledPin    = 13;
```

4. Click the Upload arrow to upload the code onto the Arduino Uno (outlined in red in the image below). Make sure the upload is successful before closing any windows.



```
/* Author: Grace Yoo, CpE
 * File Name: decodeSensorData.ino
 * Description: Software used to decode automative sensor data.
 * Team: Group 3, GOLD Team
 * Group Members: Cecilie Barreto, CpE
 *                Drew Curry, EE
 * Project: Florida Solar Beach Buggy Challenge
 * Sponsor: Duke Energy
 * School: University of Central Florida
 * Professors: Dr. Richie and Dr. Lei Wei
 *
 * Notes to self: Sensor Data comes in in LSB complement order */

// Global Variables
unsigned long pulseLength;          // Stores the Pulse Width of PWM
byte          bitValue;             // Temporary holds individual bi
byte          sensorData[16];       // Full two bytes sent by the se
byte          portAddress     = 0;  // Byte used for nibble (last 4
byte          distanceWhole   = 0;  // Byte used for nibble (last 4
byte          distanceDecimal = 0;  // Byte used for nibble (last 4
int           bitCount        = 0;  // Counts the number of bits pro

// Define used pins
const int  dataPin    =  4;
const int  ledPin     = 13;
```

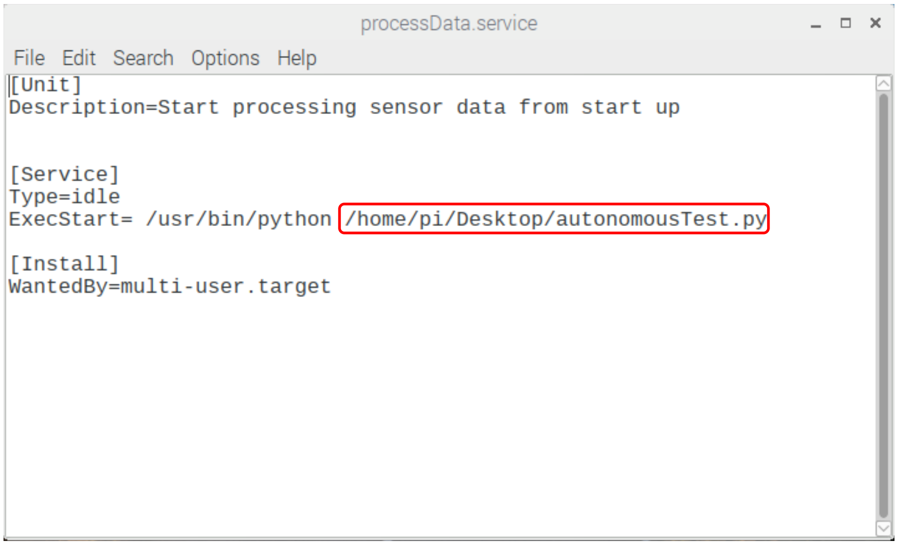
5. Double click the “autonomousTest.py” script. Make sure it is the latest version of the code. Close the window.



```
############################################################################
# Author: Grace Yoo, CpE
# File Name: autonomousTest.py
# Team: Group 3, GOLD Team
# Group Members: Cecilie Barreto, CpE
#                Drew Curry, EE
# Project: Florida Solar Beach Buggy Challenge
# Sponsor: Duke Energy
# Professors: Dr. Samuel Richie
#             Dr. Lei Wei
############################################################################

# Motor serial information:
# RIGHT MOTOR : 1 reverse, 64 stop, 127 forward
# LEFT MOTOR : 128 reverse, 192 stop, 255 forward

import time
import serial
import threading

# Port addresses to decipher which sensor is reporting data
portJOne   = "1010"
portJTwo   = "1110"
portJThree = "0110"
portJFour  = "0010"

# Set default signal flags
global sideCheck
global leftSensor
global leftFrontSensor
global rightFrontSensor
global rightSensor
global lookingFromLeft
global lookingFromRight
```

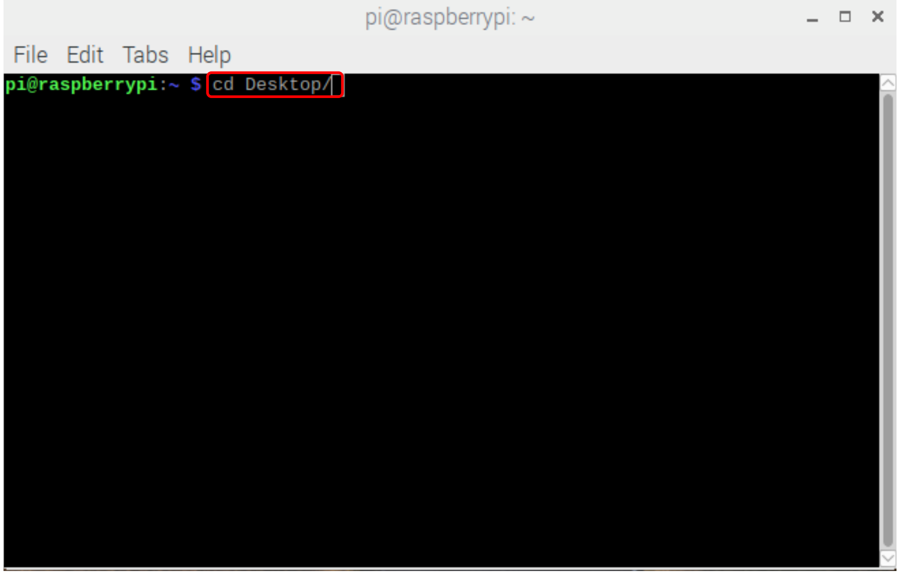
```
Python 3.5.3 (/usr/bin/python3)
>>>
```

6. Double click the “processData.service” startup script. Make sure the path is pointing to the autonomousTest.py python script described above. Close the window.



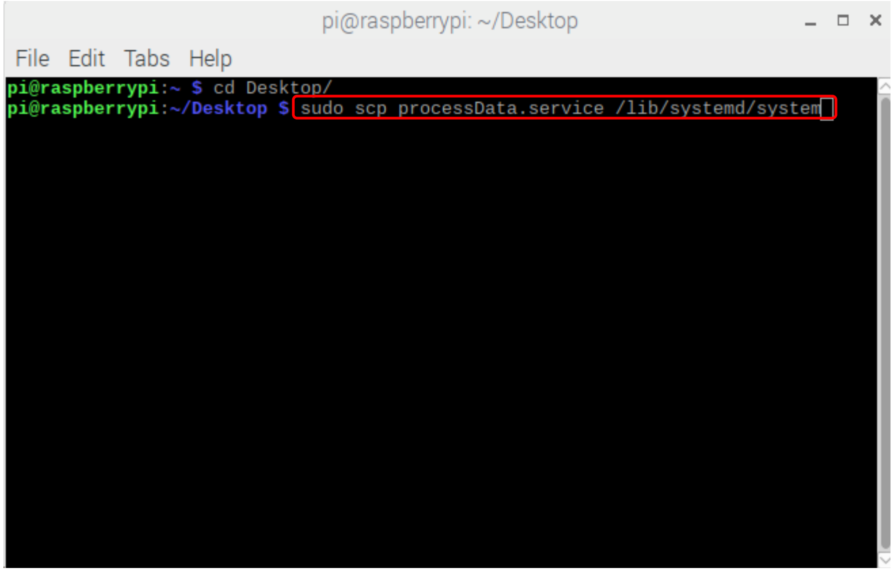
```
[Unit]
Description=Start processing sensor data from start up


[Service]
Type=idle
ExecStart= /usr/bin/python /home/pi/Desktop/autonomousTest.py

[Install]
WantedBy=multi-user.target
```

7. Open a new terminal window and navigate to Desktop by typing “cd Desktop/” in the command prompt. Then press Enter.



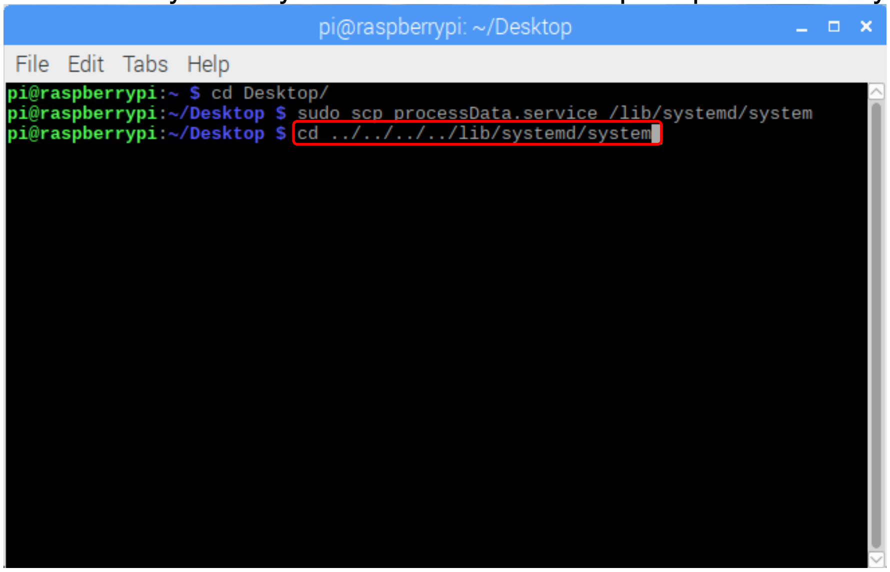
```
pi@raspberrypi:~ $ cd Desktop/
```

8. Secure copy the “processData.service” script on the Desktop to the file path /lib/system/system by typing “sudo scp processData.service /lib/system/system” in the command prompt followed by pressing Enter.



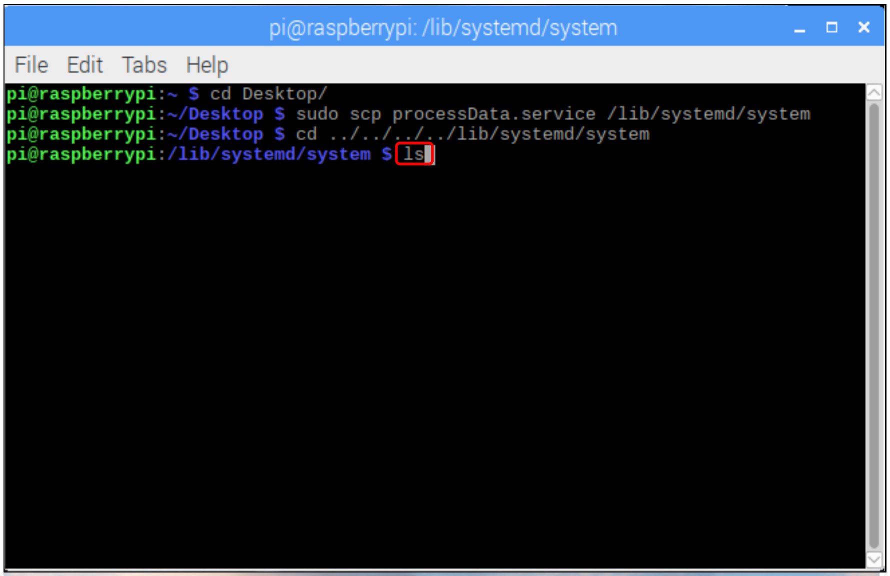
```
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ sudo scp processData.service /lib/systemd/system
```

9. Navigate to /lib/system/system/ through on the command prompt by typing “cd ../../../../lib/system/system” in the command prompt followed by pressing Enter.

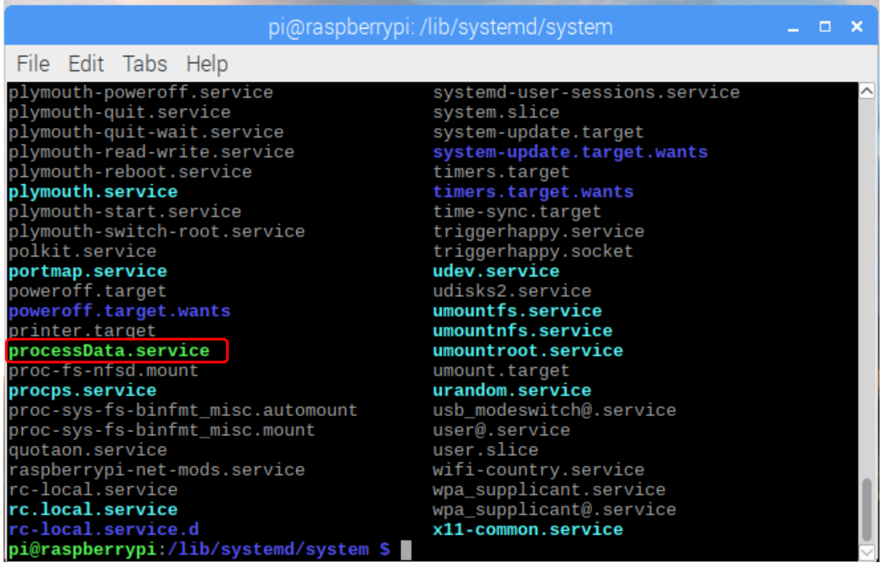


```
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ sudo scp processData.service /lib/systemd/system
pi@raspberrypi:~/Desktop $ cd ../../../../lib/systemd/system
```

10. After navigating to the /lib/systemd/system file path, type “ls” to view the contents of that folder. Confirm the copy was successful by locating processData.service in the current folder.

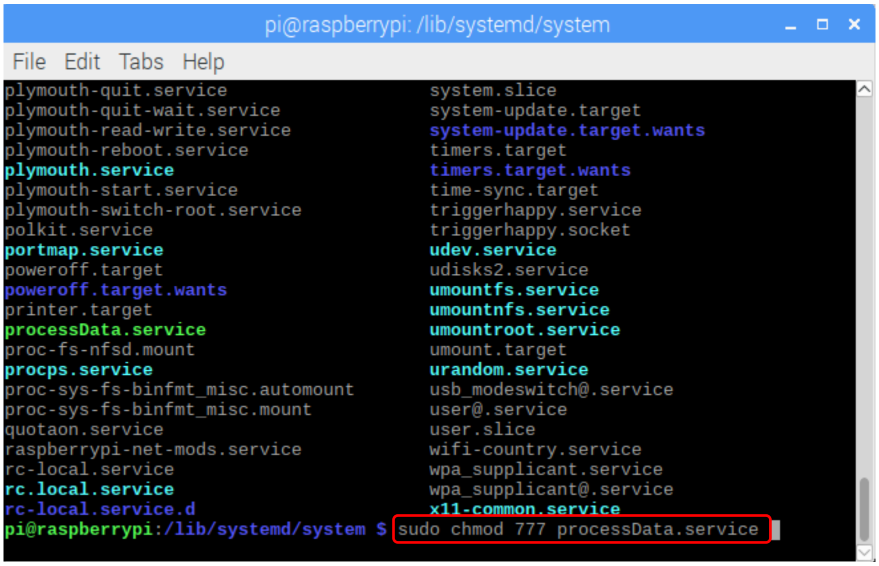


```
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ sudo scp processData.service /lib/systemd/system
pi@raspberrypi:~/Desktop $ cd ../../../../lib/systemd/system
pi@raspberrypi:/lib/systemd/system $ ls
```



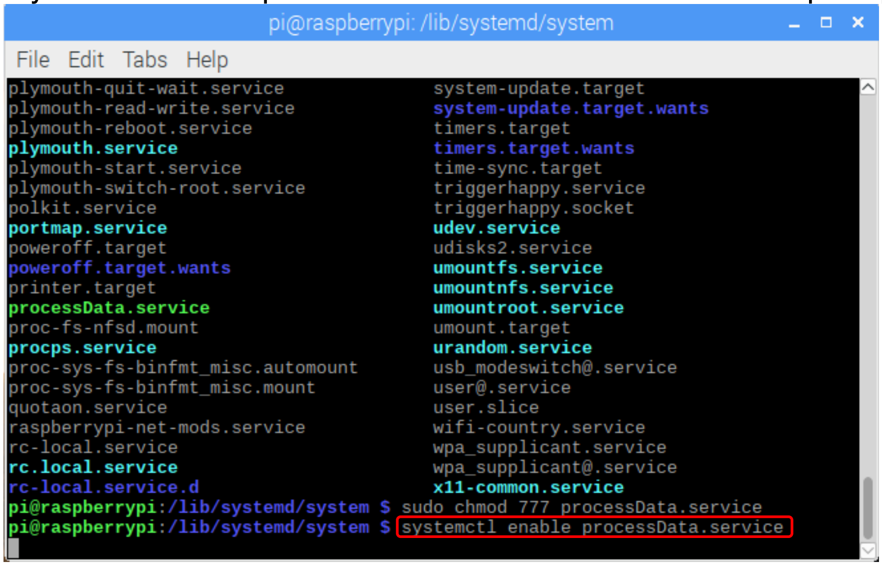
```
plymouth-poweroff.service               systemd-user-sessions.service
plymouth-quit.service                   system.slice
plymouth-quit-wait.service              system-update.target
plymouth-read-write.service             system-update.target.wants
plymouth-reboot.service                 timers.target
plymouth.service                        timers.target.wants
plymouth-start.service                  time-sync.target
plymouth-switch-root.service            triggerhappy.service
polkit.service                          triggerhappy.socket
portmap.service                         udev.service
poweroff.target                         udisks2.service
poweroff.target.wants                   umountfs.service
printer.target                          umountnfs.service
processData.service                     umountroot.service
proc-fs-nfsd.mount                      umount.target
procps.service                          urandom.service
proc-sys-fs-binfmt_misc.automount       usb_modeswitch@.service
proc-sys-fs-binfmt_misc.mount           user@.service
quotaon.service                         user.slice
raspberrypi-net-mods.service            wifi-country.service
rc-local.service                        wpa_supplicant.service
rc.local.service                        wpa_supplicant@.service
rc-local.service.d                      x11-common.service
pi@raspberrypi:/lib/systemd/system $
```

11. After verifying the copy was successful, change the permission of this file to allow it to be read, written, and executed by typing “sudo chmod 777 processData.service”



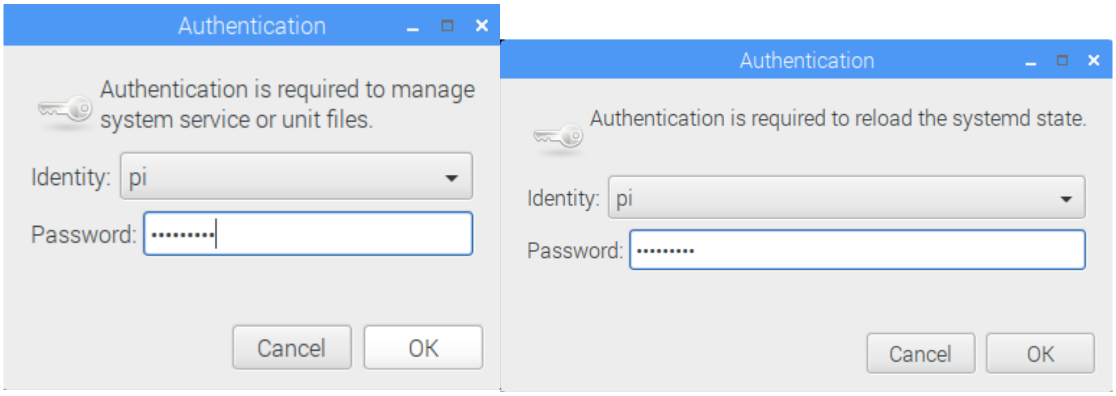
```
plymouth-quit.service                   system.slice
plymouth-quit-wait.service              system-update.target
plymouth-read-write.service             system-update.target.wants
plymouth-reboot.service                 timers.target
plymouth.service                        timers.target.wants
plymouth-start.service                  time-sync.target
plymouth-switch-root.service            triggerhappy.service
polkit.service                          triggerhappy.socket
portmap.service                         udev.service
poweroff.target                         udisks2.service
poweroff.target.wants                   umountfs.service
printer.target                          umountnfs.service
processData.service                     umountroot.service
proc-fs-nfsd.mount                      umount.target
procps.service                          urandom.service
proc-sys-fs-binfmt_misc.automount       usb_modeswitch@.service
proc-sys-fs-binfmt_misc.mount           user@.service
quotaon.service                         user.slice
raspberrypi-net-mods.service            wifi-country.service
rc-local.service                        wpa_supplicant.service
rc.local.service                        wpa_supplicant@.service
rc-local.service.d                      x11-common.service
pi@raspberrypi:/lib/systemd/system $ sudo chmod 777 processData.service
```

12. Next, enable the script to begin at bootup of the Raspberry Pi by typing "systemctl enable processData.service" in the command prompt



```
plymouth-quit-wait.service              system-update.target
plymouth-read-write.service             system-update.target.wants
plymouth-reboot.service                 timers.target
plymouth.service                        timers.target.wants
plymouth-start.service                  time-sync.target
plymouth-switch-root.service            triggerhappy.service
polkit.service                          triggerhappy.socket
portmap.service                         udev.service
poweroff.target                         udisks2.service
poweroff.target.wants                   umountfs.service
printer.target                          umountnfs.service
processData.service                     umountroot.service
proc-fs-nfsd.mount                      umount.target
procps.service                          urandom.service
proc-sys-fs-binfmt_misc.automount       usb_modeswitch@.service
proc-sys-fs-binfmt_misc.mount           user@.service
quotaon.service                         user.slice
raspberrypi-net-mods.service            wifi-country.service
rc-local.service                        wpa_supplicant.service
rc.local.service                        wpa_supplicant@.service
rc-local.service.d                      x11-common.service
pi@raspberrypi:/lib/systemd/system $ sudo chmod 777 processData.service
pi@raspberrypi:/lib/systemd/system $ systemctl enable processData.service
```

13. Entering this command will prompt the user for Authentication twice. The first authentication is to manage system service or unit files and the second authentication is to reload system state. Input the authentication password and click "OK."



Authentication is required to manage system service or unit files.

Identity: pi

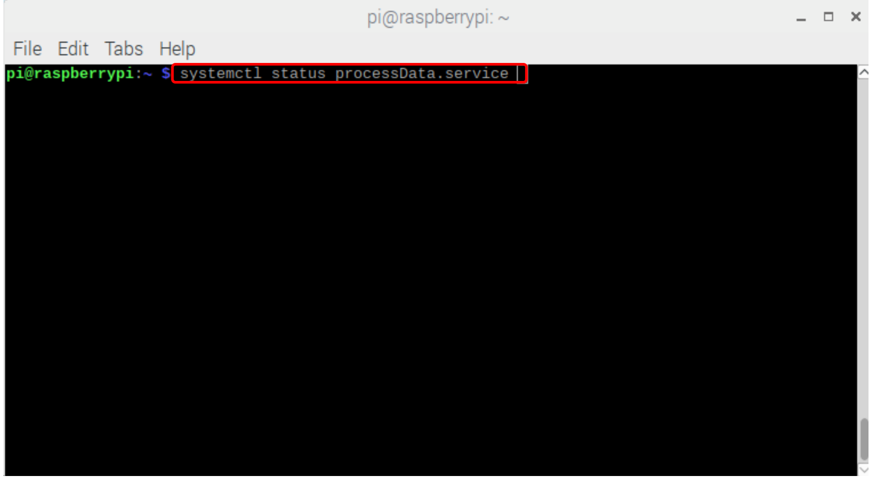
Password: •••••••••

Authentication is required to reload the systemd state.

Identity: pi

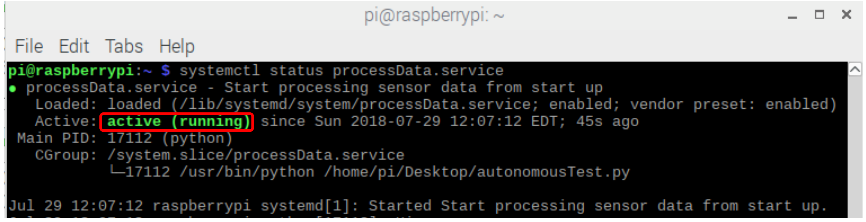
Password: •••••••••

14. After successfully enabling the “processData.service” startup script, restart the Raspberry Pi. Upon boot up, the python script should automatically run. In order to check the status of the startup script, open a new terminal window and type “systemctl status processData.service”



```
pi@raspberrypi:~ $ systemctl status processData.service
```

15. Verify the script is active (running), if the status of the script is inactive or failed, the startup was unsuccessful.



```
pi@raspberrypi:~ $ systemctl status processData.service
● processData.service - Start processing sensor data from start up
   Loaded: loaded (/lib/systemd/system/processData.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2018-07-29 12:07:12 EDT; 45s ago
 Main PID: 17112 (python)
   CGroup: /system.slice/processData.service
           └─17112 /usr/bin/python /home/pi/Desktop/autonomousTest.py

Jul 29 12:07:12 raspberrypi systemd[1]: Started Start processing sensor data from start up.
```