Florida Solar Vehicle Senior Design Project

Billy Blanchard, Hichame Boudi, Jesus Duran, and Alyssa Fejer

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450, USA

Abstract — The objective for this project is to design and build a fully solar powered autonomous vehicle while working with multiple engineering disciplines. Our goal is to fully understand the scope of our project from all disciplines involved, including mechanical, electrical, computer, and computer science. We hope to effectively work as a cohesive team to build a working and fully functioning vehicle better than all groups whom have completed this project before us. We will break down the project into tenable sections that separate research, design requirements, hardware and software components and how they relate to all disciplines involved, and final integration. We hope to gain an understanding of the level of communication, documentation, and execution that is expected of engineers by customers.

I. INTRODUCTION

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

This semester one Electrical/Computer Engineering team was formed to work in partnership with one Computer Science team and one Mechanical Engineering team. As the ECE team, we will create a hardware/software suite responsible for power regulation, i2c (inter-integrated circuit) communication and object avoidance using sensors. This suite will then be transferred to the Mechanical Engineering team's vehicle chassis along with the Computer Science team's robot vision and object detection software using a camera and Simultaneous Localization and Mapping (SLAM).

The budget for this project for both the Electrical and Computer Engineering team and the Computer Science team is $1,500 collectively and another $1,500 for the Mechanical Engineering team. All three teams are sponsored and funded by Duke Energy. Each team also has a budget of $500 for any donated items.

II. PROJECT MOTIVATION AND GOALS

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

Along with the advancements in solar power technology, autonomous vehicle software will also have a huge impact on the future, especially in the field of personal transportation. Already across the country, several companies are testing this type of technology on real roads. Globally, an expansion of self-driving technology has the potential to save lives by avoiding human-caused collisions. According to the National Highway Traffic Safety Administration, over 37,000 people died in motor vehicle crashes in 2017 [1]. Although fatality rates are trending down over the past decade, thousands are still dying every year doing such a common task. Personal autonomous cars have the potential to drastically reduce the number of vehicular fatalities.

Also, in addition to saving lives, self-driving cars allow for more accessible transportation options. Ride-sharing would no longer be dependent on a driver, allowing cars to drive unlimited by an operator. Commercially, autonomous vehicles can transport physical goods without the need for human oversight, increasing efficiency and reducing excessive labor costs [1].

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

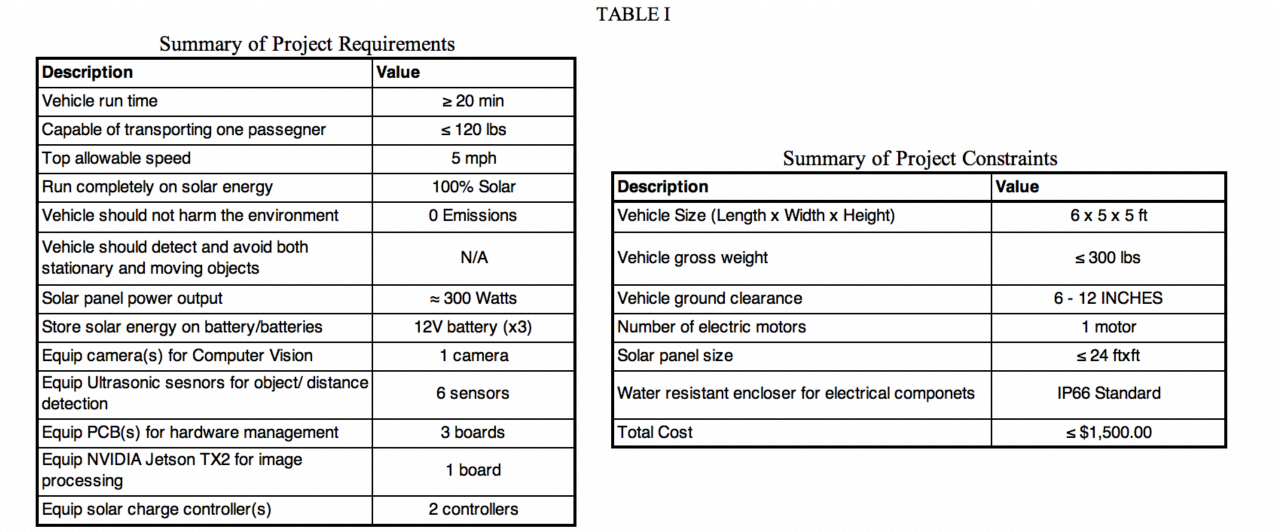


TABLE I

Summary of Project Requirements

Description	Value
Vehicle run time	≥ 20 min
Capable of transporting one passegner	≤ 120 lbs
Top allowable speed	5 mph
Run completely on solar energy	100% Solar
Vehicle should not harm the environment	0 Emissions
Vehicle should detect and avoid both stationary and moving objects	N/A
Solar panel power output	≈ 300 Watts
Store solar energy on battery/batteries	12V battery (x3)
Equip camera(s) for Computer Vision	1 camera
Equip Ultrasonic sesnors for object/ distance detection	6 sensors
Equip PCB(s) for hardware management	3 boards
Equip NVIDIA Jetson TX2 for image processing	1 board
Equip solar charge controller(s)	2 controllers

Summary of Project Constraints

Description	Value
Vehicle Size (Length x Width x Height)	6 x 5 x 5 ft
Vehicle gross weight	≤ 300 lbs
Vehicle ground clearance	6 - 12 INCHES
Number of electric motors	1 motor
Solar panel size	≤ 24 ftxft
Water resistant encloser for electrical componets	IP66 Standard
Total Cost	≤ $1,500.00

III. PROJECT REQUIREMENTS AND SPECIFICATIONS

This project was sponsored by Duke Energy with the overarching goal to engage CECS (College of Engineering and Computer Science) students in a fun and exciting interdisciplinary design challenge, while teaching students about the engineering design process, and promoting interest and awareness of solar energy. A summary of the specific design requirements, specifications and constraints given to all the CECS teams from Duke Energy can be seen above in Table I.

IV. PROJECT HARDWARE AND SOFTWARE DETAILS

The main system can be broken down into four major subsystems such as computer processing, powertrain and steering, obstacle detection, and power.

The computer processing subsystem uses nodes connected via a robot operating system (ROS) to process all the data collected and logically send instructions to receiving nodes to manage all functions of the vehicle. This subsystem is also interconnected to the computer science team's objectives as all computing is done within this subsystem. Computer science will combine their robot vision nodes with the nodes created by our Electrical and Computer Engineering team to have one complete system that controls the object detection, navigation, and movement of the vehicle. The computer processing subsystem consists of a mini computer or graphical processing unit.

The powertrain and steering subsystem manages the amount of power delivered to the motor and linear actuator. The amount of power delivered to each component will control how fast the vehicle is moving and in what direction the wheels are turned when navigating away from an object that is detected. In order to manage the amount of power that needs to be delivered to each component a feedback system is needed. A controller will be set in place along with a feedback system using a hall effect sensor. The powertrain and steering subsystem consists of an electric motor, hall effect sensor, linear actuator, PID controller, and motor drivers.

The obstacle detection subsystem manages the basic movement of the vehicle. The vehicle should be able to adjust the movement including speed and direction of the vehicle as both stationary and moving objects are detected. Primary object detection and navigation for this vehicle is the responsibility of the Computer Science team. The obstacle detection subsystem built by our Electrical and Computer Engineering team is meant to be the secondary mode of object detection and navigation. It is a critical requirement that the vehicle avoid objects as safety of the entire vehicle is each teams' number one priority. The object detection subsystem consists of sensors and a PCB.

The power subsystem collects and distributes the energy produced by the solar panels to the various hardware components in the system. This subsystem has a high degree of importance since the safe operation of the vehicle depends on how the power is distributed across the entire system to ensure all hardware components receive the necessary power to function properly. The power subsystem consists of the solar panels, batteries, charge controllers.

V. PROJECT BLOCK DIAGRAM

The following block diagram outlines a description of how the hardware components interface with each other. Each member of the Electrical and Computer Engineering team was assigned to lead a particular subsystem, with us all working together to complete the project as a whole.

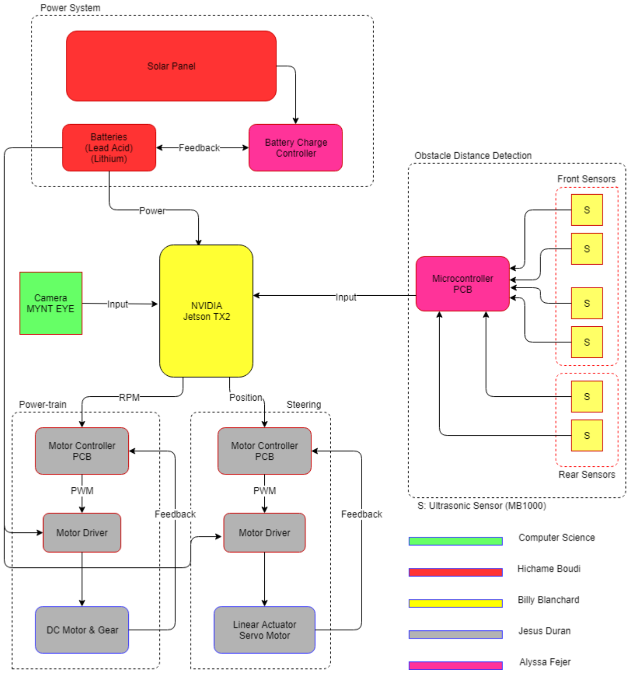


Fig. 1. Block Diagram

Since this is an interdisciplinary project we had to not only construct a block diagram to show the flow of how all the components will be connected, but we needed to physically design where on the vehicle the components would need to be housed. The Mechanical Engineering team was responsible for designing the chassis for the vehicle, so they had to account for the weight and placement of all of the electrical and computer components. This design can be seen below in Figure 2.

VI. COMPUTER PROCESSING SUBSYSTEM

Both ECE and CS teams made the decision to choose NVIDIA's Jetson TX2 embedded system for the GPU (graphical processing unit). The Jetson TX2 is a powerful mainframe computer or what the industry may refer to as a "Supercomputer". Not to be misleading, the TX2 is a low-powered embedded system operating well under 8 watts during general energy usage [2]. The TX2 contains Nvidia's Pascal Graphics Processing Unit which is embedded with a 256-cores. The TX2 is also comprised with 8GB of Low Power Double Data Rate memory (LPDDR4) and a 16-core ARMv8 64-bit CPU. The LPDDR4 comes with a 64/128-bit interface [2].

The Nvidia Jetson TX2 modules contains a powerful System-on-Chip (SoC) which integrates an Advanced RISC Machine (ARM) architecture central processing unit (CPU), graphic processing unit (GPU), host-bridge and southbridge onto a computer. Concurrently, the Jetson Boards of this discussion are both able to run various software and operating systems including the Robot Operating System which is Linux-based [2].

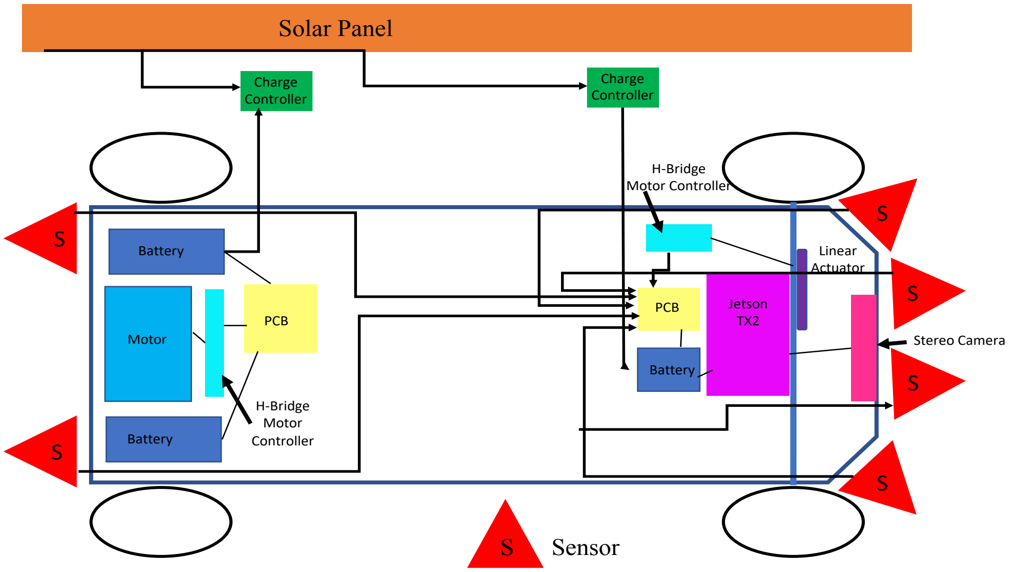


Fig. 2. Hardware Integration Diagram

VII. POWERTRAIN SUBSYSTEM

A. Motor

The Mechanical Engineering chose their preferred motor for this project. The ZXTDR, model M1020, electric motor, along with an appropriate gearing system, will be used to control the rear axle of the vehicle. This motor is a brushed DC electric motor with a maximum draw of 21 amps at 2500 RPM. Depending on the connections of the field to the power supply, the speed and torque characteristics of a brushed motor can be altered to provide steady speed or speed inversely proportional to the mechanical load.

B. Hall Effect Sensor

A Hall effect sensor is a device that is used to measure the magnitude of a magnetic field. Its output voltage is directly proportional to the magnetic field strength through it [3]. A hall effect sensor will be placed on the rear wheel axle of the vehicle to be used to track the speed or RPMs (rotations per minute) of the motor. This can be considered a feedback system of the motor.

Figure 3 below shows the implementation of the hall effect sensor on the axle. It uses a tooth gear to interrupt the magnetic signal between the sensor and the magnet. Hall-effect gear-tooth sensors detect changes in a magnetic field. When a gear tooth or other ferrous object approaches a magnet, it distorts the field surrounding that magnet. The presence or absence of ferrous targets are detected by measuring this distortion [3]. The most common method of measurement places a sensor on one magnet, and observes an increase in the magnetic field as a target approaches the magnet.

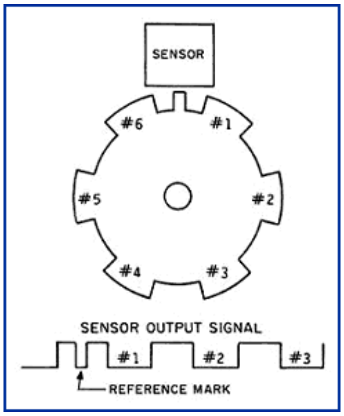


Fig. 3. Gear-tooth hall effect sensor diagram showing output from magnetic field detection [3]

C. Linear Actuator

The Mechanical Engineering team chose to have a 12" linear actuator connected to the steering tie rod for steering as represented in the Figure 4 below. The linear actuator's 12V, 3A power requirements will be achieved using a Step-Down DC/DC converter.

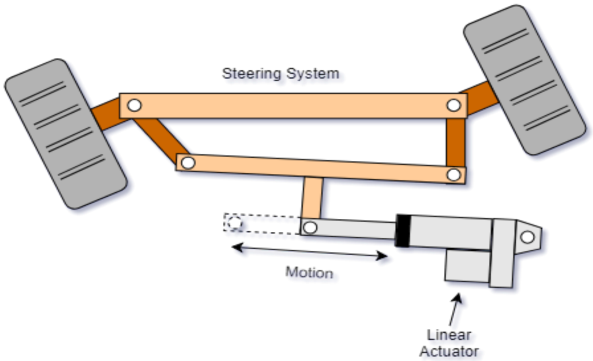


Fig. 4. Top View of Steering System with Linear Actuator

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

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

The linear actuator chosen by the Mechanical Engineering team has a built-in potentiometer that can be used to keep track of the shaft position. This will be used as the feedback for the steering system.

D. PID (Proportional, Integral, Derivative) Controller

PID (proportional integral derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller. PID control is a well-established way of driving a system towards a target position or level [4]. Consider the feedback system architecture that is shown in Figure 5 where it can be assumed that the plant is a DC motor whose speed must be accurately regulated.

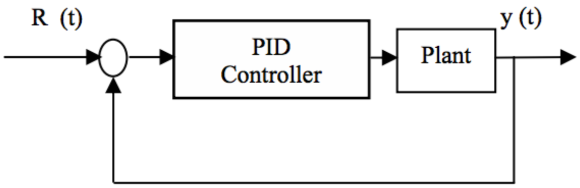


Fig. 5. Feedback system architecture [4]

The PID controller is placed in the forward path, so that its output becomes the voltage applied to the motor's armature; the feedback signal is a velocity, measured by the hall-effect sesnor; the output velocity signal y(t) is summed with a reference or command signal R(t) to form the error signal e(t). Finally, the error signal is the input to the PID controller [4].

E. H-Bridge Motor Driver

The H-bridge circuit derives its name from the full-bridge circuit shown in Figure 6. The motor forms the crosspiece in the "H". Speed and direction are controlled as current flows through the motor in the direction determined by the position of the switches in the bridge [5]. In this example, with switches "A" and "D" closed, the motor will operate in a clockwise (CW) direction. With "B" and "C" closed, the motor will operate in the counterclockwise (CCW) direction [5].

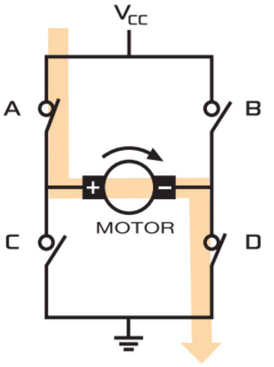


Fig. 6. Simplified H-Bridge Schematic [5]

An H-Bridge motor driver will sit in between the Arduino and the DC motor to manage the speed of the motor using Pulse Width Modulation (PWM). In the PWM implementation, the speed is controlled by the width of series of pulses of equal voltage. A Cytron Motor H-Bridge driver was selected since it can handle a max current of 80A and a constant operation of 30A.

VIII. OBJECT DETECTION SUBSYSTEM

A. Ultrasonic Sensors

One of the major requirements for the project include, the ability to detect and avoid objects. There is a wide selection of sensors that perform the task of detecting objects at various distances. These include infrared, passive infrared, inductive, capacitive and ultrasonic sensors. After much research was done on the many types of available sensors, the best type for our systems application was chosen to be ultrasonic sensors.

Ultrasonic sound vibrates at a frequency above the range of human hearing. Transducers are the microphones used to receive and send the ultrasonic sound [6].

The ultrasonic sensor used in the object detection subsystem uses a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse [6]. Figure 7 below shows a diagram of this process.

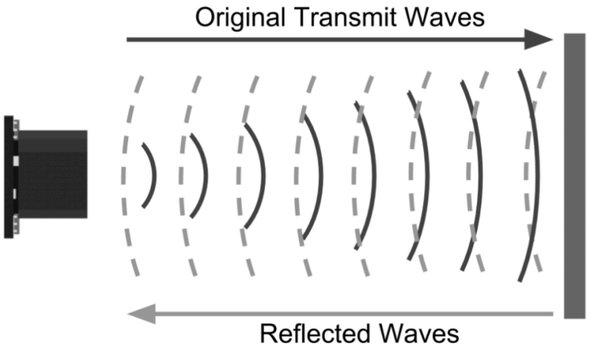


Fig. 7. Ultrasonic sensor transmitting and receive frequency waves to detect the distance of an object [6]

The main advantage to using ultrasonic sensors for our project is the reliability in any lighting environment. This means the sensors will always be able to strongly detect objects within its allowable range of 5 – 200 inches. The ultrasonic sensor used in this project is the MaxBotix MB1000 [6].

B. PCB (Printed Circuit Board)

The main requirement for ECE senior design is to design and build a printed circuit board (PCB). A printed circuit board (PCB) mechanically supports and electrically connects electronic components or electrical components using conductive tracks, pads and other features etched from one or more sheet layers of copper laminated onto and/or between sheet layers of a non-conductive substrate.

The PCBs used in this project are centered around the Atmega chip produced by Arduino. Then only the necessary connections that are need for electronic communication are added to the board design to ensure the least amount of space is used, with the proper distribution of power to the sensors, each needed 5V to function properly. Figure 8 below shows the schematic designed for the object detection subsystem.

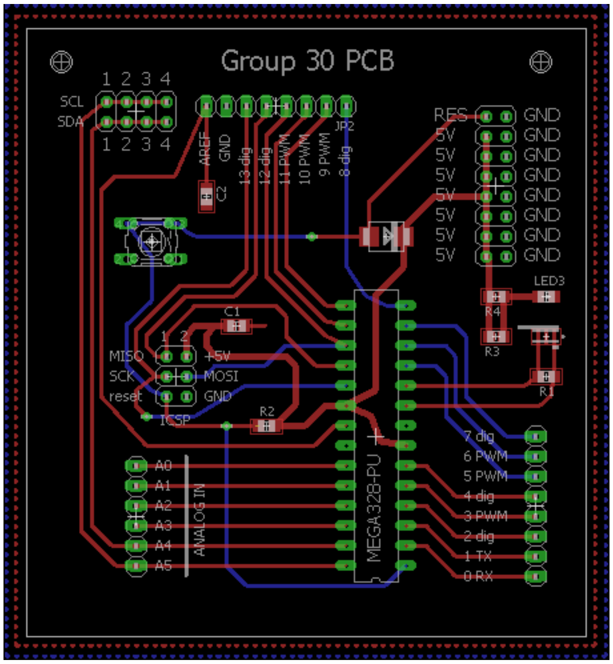


Fig. 8. PCB used to connect and manage communication of the ultrasonic sensors

As can be seen from the hardware integration diagram (Figure 2), six ultrasonic sensors are needed for object detection in the vehicle. Each sensor needs its own PWM line for communication and its own 5V power and ground via the general I/O (input/out) pins.

IX. POWER SUBSYSTEM

A. Solar Panels

Photovoltaic solar panels, referred as just solar panels, are composed of a number of photovoltaic cells that generate electricity by absorbing sunlight. They are generally fabricated using semiconducting materials, mainly Silicon and Gallium, with varying degrees of efficiency. The Mechanical Engineering team was responsible for providing the solar panels for this project.

The minimum power needed from the solar panels to ensure the entire system functions is 300W.

B. Batteries

One major requirement of this project is to design and build a completely solar powered vehicle. In an ideal world, it would make sense to connect all the electronic components directly to the solar panels. Even though this seems like a simple system it is unrealistic. Power created by the solar panels is to large and varying to be safely connected directly to electronics. Therefore, this system will involve solar panels and batteries to store solar energy to deliver constant power and to be used when there is no sunlight being harvested.

Another consideration unique to this system is the motor. The motor chosen for the vehicle is a 24V, 500W, Brushed Electric Motor. One major drawback to working with motors is the large amounts of electrical noise they produce on the power line of the system. This noise can interfere with the ultrasonic sensors and can even impair electronics by causing voltage dips on the regulated power line. Large enough voltage dips can corrupt the data in electronic registers. Because of this, one battery power system will be connected to the powertrain subsystem and the other will be connected to the computer processing subsystem, which is responsible for powering the object detection subsystem.

The battery power needed to power the powertrain subsystem is 24V, consumed by the motor. Two 12-V lead acid batteries connected in series to produce 24-V will be used and one 12-V lead acid battery is used to power the NVIDIA Jetson TX2. The NVIDIA Jetson TX2 distributes power to the PCBs, which then distribute power to each ultrasonic sensors.

C. Charge Controllers

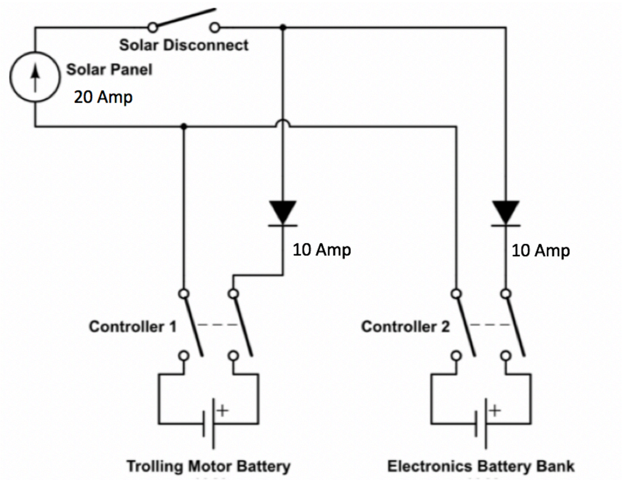
The major efficiency of the system is the amount of energy created from the solar panels connected to the vehicle. To harness this energy and convert it into usable amperage to charge the three 12-volt batteries, a charge controller is used. Charge controllers are used to regulate the input voltage and/or current coming from a power source. They regulate the input power to charge the batteries effectively and does so safely by preventing the batteries from being overcharged.

Without a charge controller, there is nothing keeping the batteries from being overcharged and expanding, heating up, or exploding. Charge controllers also prevent batteries from being undercharged, which could induce various degradation mechanisms depending on the battery (undercharging poses no issues for some batteries); reduction in capacity, copper dissolution, sulfation, early battery failure.

Since the Mechanical Engineering group is providing their own solar panels, with a max wattage of 350W it was determined that at least two 15-amp charge controllers are needed to charge the three batteries; the two 12V Lead Acid battery pack for the motor and the one 12V Lead Acid battery for the electronics.

With the need for two charge controllers to charge two separate battery systems a circuit was designed to ensure the highest power efficiency. The best way to connect the charge controllers is in parallel. This way the 20 amp being produced by the solar panel can be divided into a value within the charge controllers' limits without losing the voltage needed to recharge the batteries. Figure 9 shows the flow of power from the solar panels and then distributed to the charge controllers and battery systems.

Fig. 9. Solar Panel and Charge Controller Integration



IX. (ROS) ROBOT OPERATING SYSTEM

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. ROS was chosen for this project to help manage the multiple communication peripherals of each electronic component and how they interact with each other.

The key feature of ROS is the way the software is run and the way it communicates, allowing you to design complex software without knowing how certain hardware works. ROS provides a way to connect a network of processes (nodes) with a central hub. Nodes can be run on multiple devices, and they connect to that hub in various ways.

The main ways of creating the network are providing requestable services, or defining publisher/subscriber connections with other nodes. Both methods communicate via specified message types. Some types are provided by the core packages, but message types can be defined by individual packages.

X. HARDWARE/ SOFTWARE COMMUNICATION AND INTEGRATION

The solar vehicle's software portion will be implemented using a cluster of two PCBs featuring the ATMega328 microprocessors in parallel with the Jetson TX2 developer's kit, which will rest on the Tegra system-on-chip [2]. The microprocessor will take i2c (inter-integrated circuit) input and produce outputs which then will be used as triggers or flags inside of the PCB's code.

The ultrasonic sensors are placed in a way that covers the most amount of area in the path of the vehicle. The ultrasonic sensors are connected to the PCB to process the ultrasonic sensor's data and calculate the distance to the objects. The calculated data is then sent to the Jetson TX2 board to be used by the obstacle avoidance algorithm.

The MaxBotix LV-MaxSonar-EZ0 MB1000 Ultrasonic Sensors will be placed around the vehicle in a way that reduces blind spots. The position of the sensors is calculated considering the minimum working angle of the sensor which is approximately 43^0 [6]. Figures 10 and 11 show an approximate representation of the MaxBotix MB1000 beam projection and the exact location the sensors will need to be place to achieve an adequate obstacle detection performance.

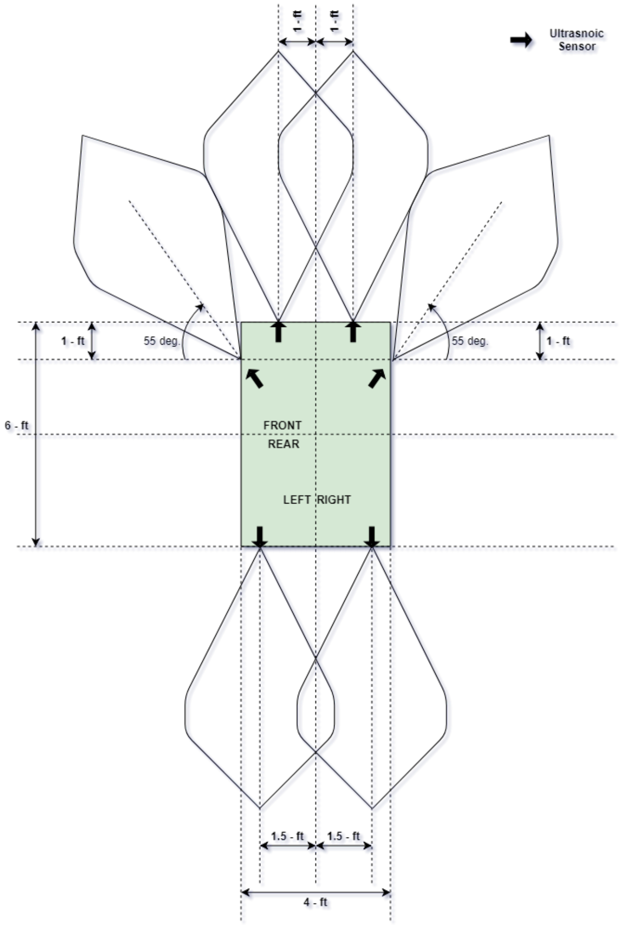


Fig. 10. Top view of ultrasonic sensor location

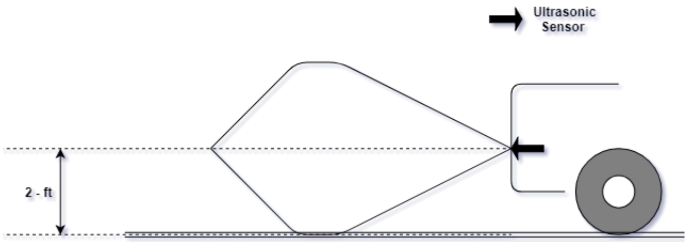


Fig. 11. Side view of Ultrasonic Sensor elevation

For the overall system to work as planned we have constructed a software module which consists of a flow diagram chart or state machine to handle and control certain tasks. The software modules are broken down into software subsystem designs: System Check, Camera Read, Sensor Read, Stop, Acceleration, and Steering. Based on the Vehicle Software State Machine in Figure 12, the software design flow will begin with a system check. After a validation of system check there will be a follow up read; a read of the onboard camera module.

If the system check returns a negative validation i.e., if the system is NOT OK, then, log the error into a text file on the Jetson TX2 ROS topic to return which of the subsystems failed. Both the camera read, and ultrasonic sensors read will be occurring simultaneously using an algorithm that implements parallel-programming for the data reads. Ultimately, this read in parallelism is an act of fail-safe of object detection.

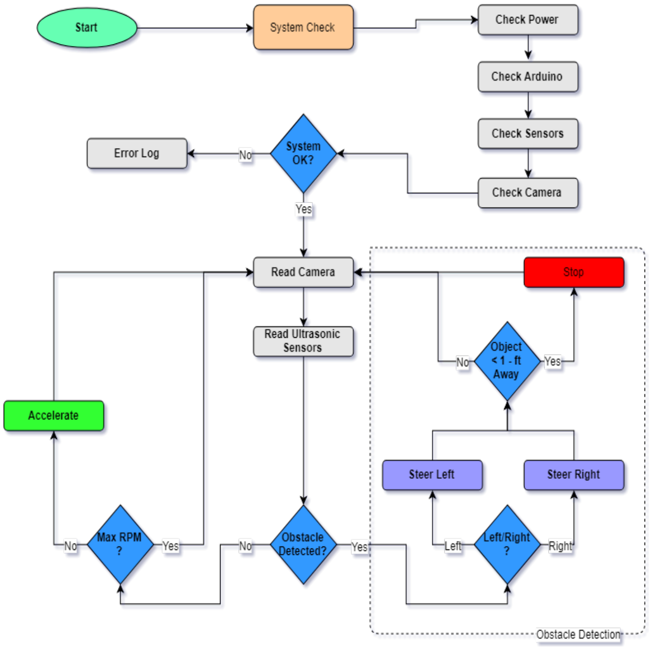


Fig. 12. Vehicle Software State Machine

Every state described above in the vehicle state machine will be programmed into a node in ROS. For example, the motor controller, the sensor read, the camera read, and the system check states will all be in a node separately. All of these nodes will be implemented using a client library support of rospy or roscpp for Python or C/C++, respectively.

In the Object Avoidance state, this program design includes motor control using the Hall-effect sensor, and PCB microcontroller. While the objected detected is within 5 feet, the program's algorithm is set to reduce the speed of the solar vehicle. This reduction of speed is for the act of giving an object or obstacle detected time to move out of the vehicle's detection field of view. If the object or obstacle detected has not moved on its own using our obstacle avoidance algorithm, the vehicle will make a critical selection to either steer left or steer right to avoid or move around the object.

If an object is still detected but has reached our critical detection range of an object, then the robot will terminate immediately. Internally, the program has switched to an idle or stop state. This abrupt termination of the solar vehicle is to be acting as an emergency shut off or power down switch since the solar vehicle is to be fully autonomous. While the robot is in the stop or idle state, it shall then continue to read and sense data input from the TX2 onboard camera module.

Since the vehicle is autonomous it will continue to run until the batteries are drained or the emergency stop button is pressed. The goal for this project is to run for at least 20 min before the batteries are exhausted.

REFERENCES

[1] "2017 Fatal Motor Vehicle Crashes: Overview," Traffic Safety Facts, Oct-2018. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812603.

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

[3] "Putting Teeth in Hall-Effect Sensors," Machine Design, 03-Apr-1998. [Online]. Available: https://www.machinedesign.com/archive/putting-teeth-hall-effect-sensors.

[4] F. Loucif , "DC MOTOR SPEED CONTROL USING PID CONTROLLER," Researchgate.net, rep., Jun. 2008.

[5] "Controlling DC Brush Motors with H-bridge Driver ICs," Innovations Imbedded. [Online]. Available: https://www.rohm.com/documents/11308/12928/100260.H-BRDG_WP_Jan09.pdf.

[6] R. Burnett, "Understanding How Ultrasonic Sensors Work," MaxBotix, 02-Dec-2018. [Online]. Available: https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm.