LAFAYETTE COLLEGE
ECE 491 - SENIOR DESIGN I
TEAM ALPHA - FINAL DESIGN PROPOSAL

Michala Dennis, Alexander Labell, Devin Arnold, Corin Nosow, Austin Pelayo,
Samuel Ehgartner, Drew Borek & Jack Horowitz

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

BACKGROUND

The MARGE System is an autonomous driveway security apparatus intended to perform automobile authentication and access restriction. The system consists of a solar powered charging station, referred to in this document as the "Docking Station", and a mobile robot referred to as the "Rover". These will be two separate devices. The system has four modes of operation, "Normally Closed", "Normally Open", "Scheduled" and "Remote Control." The "Normally Open" mode places the Rover on the Docking Station as its default state and will block unauthorized vehicles from entering by driving into the roadway and deploying a visual barrier. All the while, permitting authorized vehicles passage. The "Normally Closed" mode places the Rover in the middle of the road as its default state and it will move out of the way of vehicles to permit passage. The "Scheduled" mode will allow for the user to create a scheduled protocol for the "Normally Open" and "Normally Closed" modes. The "Remote Control" mode allows the user to directly control the Rover's movement, superseding commands normally given in the operating state from which it was functioning.

This final proposal, based on feedback from the customer and team development since our preliminary proposal, drafted in response to the Request For Preliminary Proposal (RFPP), details the final schema for the system.

MECHANICAL DESIGN

The MARGE System will have two mechanical components, the MARGE Rover and the Docking Station. Each system has been mechanically designed to ensure the most effective operation, and will be covered in the subsections below.

The Docking Station

The Docking Station will be a stationary structure responsible for charging the MARGE Rover, sensing the presence of incoming vehicles, providing navigational aid, and sending commands to the Rover. The Docking Station will be positioned on the side of the roadway on flat ground and must remain in the same spot during operation. The Station will be powered by a photovoltaic panel that will be placed separately by the user to achieve better performance.

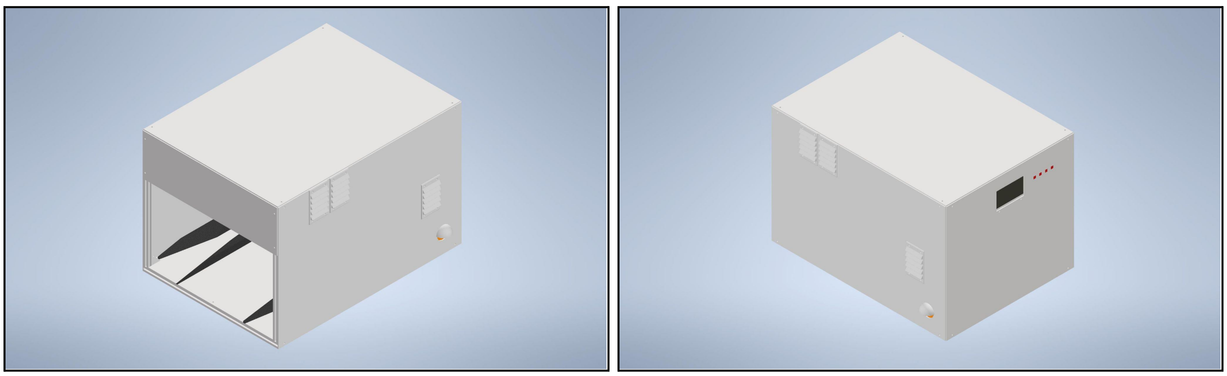


Figure 1a & 1b: Complete Model of Docking Station Front and Back

Physical Design:

The Docking Station's total dimensions will be 22 inches wide, 30 inches in length and 20 inches tall. The structure of the station will be broken up into three main sections. There will be an outer frame that will enclose all of the contents of the station. This frame will be constructed using European Standard T Slot beams and associated T Slot brackets. The T Slot beams are made of 2020 anodized aluminum which provides a sturdy and strong enclosure for the components inside. To complete the enclosure, side panels will be attached to all faces for additional structural support and environmental protection. These panels will be ⅛ inch thick polypropylene plastic sheets that are machined to fit the sizes of Docking Station faces, and include any necessary ventilation and cable routing. Polypropylene was chosen for its structural integrity, lighter weight, and cost effective advantages. It is also resistant to moisture, fatigue and corrosion, which aids in weather protection and resilience to wear and tear. The last main structural component of the Docking Station will be the floors and partitions, which will also be made out of the same polypropylene sheets. The Docking Station will be divided into three compartments. The lower level will contain the garage and energy storage sections. The garage will be 20" x 22" x 14" and is where the MARGE Rover can enter to charge and await interception of incoming vehicles. Charging contacts will be positioned on the floor of the garage and will have spring loaded pins that will be depressed when the Rover drives over them. The charging contacts will run a direct line under the floor of the Docking Station to the energy storage compartment, where it will connect to the Battery Charge Controller. The energy storage compartment will be 10" x 22" x 14" and will be where the battery and power controllers will be located, as well as the connection port for the solar panel which will be separated from the Docking Station structure. The upper layer will be for the computational components for vehicle sensing, communication and the user interface screen. A cable routing trough will connect the energy storage and computational sections due to the high dependence on power. The computational section will be 30" x 22" x 6" and have ventilation points included on the Docking Station body panels to prevent overheating. All side panels in the Docking Station have been carefully designed to allow for quick and easy access to the internals. All panels have four M5 screw attachment points that can be removed to open any side of the Station. This is also the case for the internal partitions between compartments and the vent covers on the walls of the Station. All electrical components will be bracketed and secured to the floors or walls of the structure, via a similar screw attachment design. Final Docking Station weight specification will be 70 pounds.

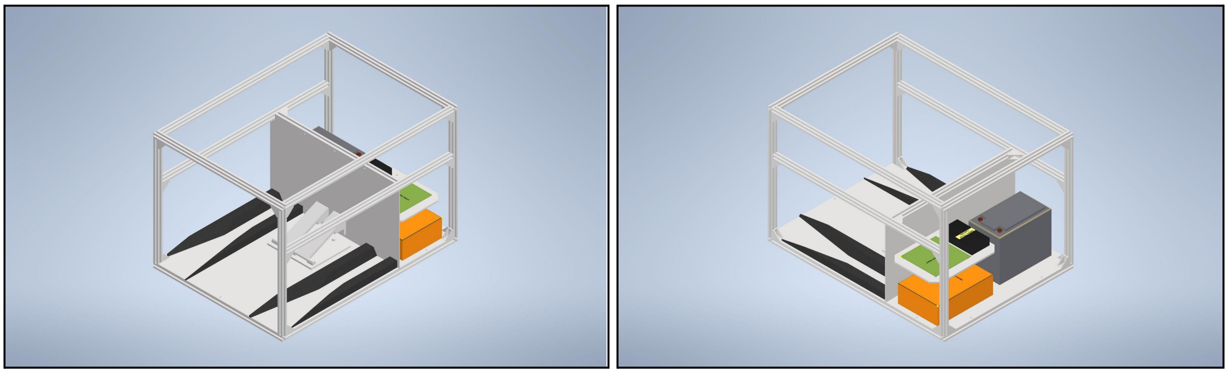


Figure 2a & 2b: Model of Lower Compartments of the Docking Station

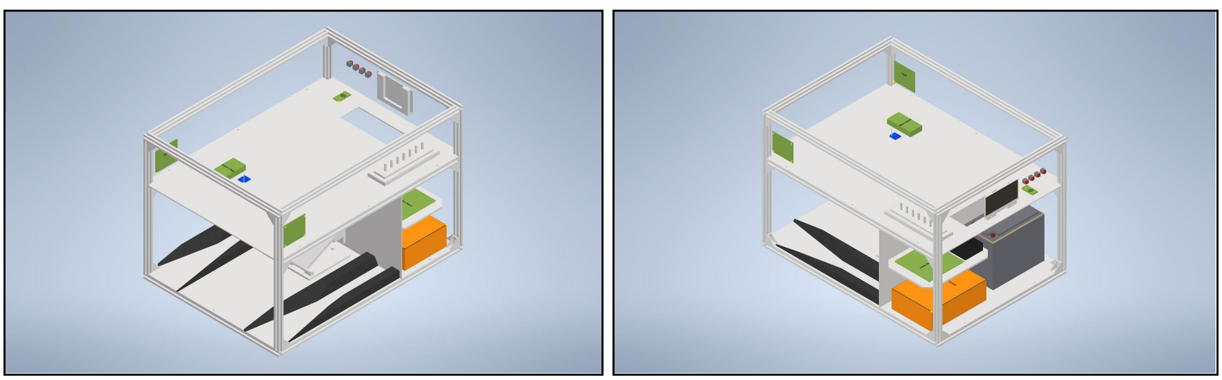


Figure 3a & 3b: Model of Upper Compartments of the Docking Station

Navigational Aid:

In addition to the MARGE Rover's navigation protocol, the Docking Station will include aids to assist in the Rover's ability to enter the garage and stay aligned with the charging contacts. The main navigational component will be the ArUco tag located on the rear wall of the garage, above the charging contacts. The MARGE Rover's navigation will be using computer vision to identify and move with reference to the tag. It is critical that the Rover has continuous line of sight with the tag, so an LED will be used to illuminate the tag during the evening.

In order to make the docking procedure more reliable and smooth, the Docking Station will be equipped with the Mechanical Guidance System (MGS). The MGS is a set of wheel channels that decrease in width as they extend further into the garage. The wheel channels have a 60 degree inclination on either side in order to take advantage of the omnidirectional wheels of the MARGE Rover (see "The MARGE Rover" subsection). The purpose of the MGS and its 60 degree inclined channel edges is to force the MARGE Rover to a predictable path and ensure the best possible chance of contact when attempting to charge. The structural components of the MGS will be 3d printed due to the more complex geometry required in the design and to reduce weight. The MGS parts will be fixed to the floor of the Docking Station by using screws.

The last navigational assistance feature of the Docking Station is the ramp that will be positioned at the entrance to the garage. It is not known exactly what kind of terrain the Docking Station will be situated on, so a very shallow ramp will be mounted on the front of the Station to allow the Rover a significantly easier time to enter the garage. The degree of the ramp will be kept to a minimum so that the Rover does not lose sight of the Aruco tag while approaching the garage.

<u>Energy Production and Storage:</u>

The Docking Station's source of energy will come from the photovoltaic panel that is to be placed nearby. The panel will be mounted to a 28" adjustable tilt stand for achieving more effective power generation at different latitudes. The photovoltaic panel's connection to the Docking Station will be through a port on the right side wall. The port will be covered by a half-hemisphere guard to prevent rain from entering the interior of the Station. Due to the complexity and geometry of the guard, it will be 3d printed for fabrication. The battery will sit at the back of the energy storage section and connect to the solar panel through the solar charge controller. Due to the weight of the Docking Station battery (10.62 lbs), an additional T Slot beam will be fixed underneath it to support the polypropylene floor plate. In addition to the battery and solar charge controller, the power distribution board that will feed all electrical systems will be located in the energy storage compartment. See the Electrical Design section for further discussion on these electrical systems.

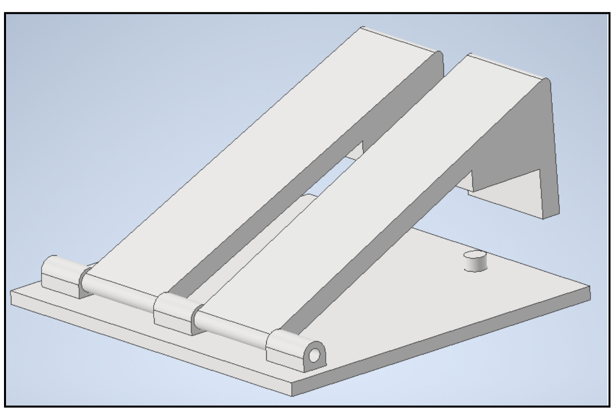


Figure 4: Spring-Actuated Charging Contacts

Computations and Processing:

The upper floor of the Docking Station will be dedicated to all computational and processing requirements. There are no partitions and the entire Docking Station length and width will be utilized for this purpose. The primary systems that will be stowed in this compartment are the Arduino Mega, the nRF transceivers, radar sensors and associated processing, the interface screen and buttons. The electronics that are prone to heat production will be situated in the path of airflow from the ventilation system discussed below, while others will be placed in accordance with their functions. The radar system will have one sensor placed on either side of the compartment. Both sensors are doppler radars which require a radome to function properly, therefore the bracketing and mounting plates will be made to a very specific thickness to allow the passage of the waves out of the Station. The screen will be placed at the rear of the compartment, in a bracket that will be mounted to the rear wall. The buttons will be attached to the right of the screen. Due to the size of the components versus the space available in the Docking Station's upper floor, the room for expansion and additional unforeseen components can be added with no issues. Additional brackets may need to be fabricated, but they will all be done by 3d printing, allowing implementation to be quick.

Cable Management and Electrical Grounding:

Cable management will be a priority for the internal components of the Docking Station. Numerous methods of routing cables and organizing them will be employed. The cable organizers will be 3d printed for quick fabrication as a result of the variable designs for specific purposes. As stated previously, there will be a wiring trough to feed cables from the battery and power distribution to the computational compartment. Cables will be grouped and tied from the distribution board up to the upper compartment where they are then split and sent to their required locations.

For the electrical grounding of the Docking Station, a chassis ground solution will be utilized. At the rear end of the computational compartment, there will be a bus bar for all grounded cables. This bus bar will have a single connection point to the T Slot frame of the Docking Station to avoid the potential for ground loops.

A main power switch will be located on the interior of the station, found in the energy storage compartment. This will cutoff all power to all systems on the Docking Station, acting as a main shutdown.

Ventilation and Environmental Protection:

Multiple measures have been put in place to negate overheating within the Docking Station. On a very basic level, the polypropylene plastic sheets that are covering almost all sides of the Station will be off-white in color. White is less absorbent of heat than other colors which should keep the Docking Station slightly cool, although it is unknown how much of an effect it will have on temperature. A more active approach to cooling will be in the form of ventilation paths on the upper and lower compartments of the Station. The vents will use a design that makes water near impossible to enter due to shielding that covers the vent openings from above, seen in the figure below. The same vent design will be used on the Rover. Due to the temperature analysis performed on the Docking Stations power distribution board, a fan will be implemented in the energy storage compartment.

The fan would be mounted on the right body panel of the Docking Station and pull air across the components from the left side vents and exhaust on the right side. The specific vent cover will be 3d printed, due to its small size, and potential for further rapid design adjustments. The user interface screen, which is slotted into the rear wall of the Station, will require a clear plastic cover in order to be protected by environmental hazards. Finally, as a means of preventing water from entering the vulnerable areas of the Docking Station, silicon strips will be attached to all seams of the Station's walls and partitions in order to form a gasket.

Figure 5: Ventilation Cover

<u>The MARGE Rover</u>

The Rover will be responsible for traveling between the Docking Station and the midpoint of the driveway. It will act as a responsive or passive indicator to cars entering the driveway and will be charged inside the Docking Station.

<u>Physical Design</u>:

The Rover chassis will be 14 inches in width, 18 inches in length, and 8 inches in height. This does not include the wheels or the added height from the cone when it is extended. The wheels will add an estimated 2 inches in height and 3.2 inches in width. There will be 4 wheels in total mounted to the outside of the Rover.

Two 6 inch wheels will be connected to the drive motors and will be the primary method of travel to and from the Docking Station. These wheels will be the furthest from the Docking Station. When traveling out towards the driveway, they will be oriented in the front of the Rover. On the way back to the Docking Station, the Rover will drive in the reverse direction, so they will be oriented in the back of the Rover. The other wheels will be two 4 inch omni-directional wheels that are not driven by any motors. These were chosen because of their ability to pivot side-to-side without dragging along the ground, making steering easier. They also compliment our Docking Station wheel channels well. They will be mounted to individual axles that are mounted on a designed suspension system.

On the end of the Rover facing the Docking Station, the top 3 inches of the corners will be chamfered (26.57 degree angle or 2 inches by 4 inches) to allow for easier alignment within the Docking Station before the charging contacts meet. The same chamfer will be within the Docking Station.

The walls of the Rover will be 3D printed in 4 segments that will be screwed down by their built-in brackets into the base plate. Both the base plate and the roof plate will be made of polypropylene plastic panels cut to size with necessary mounting holes drilled into them. An additional metal brace will be placed underneath the battery if bowing from the weight occurs during testing.

Wiring troughs are also designed onto the inside walls of the Rover to organize wires, keep them out of the way of internal components, and to give us a surface to tie them down to. There will also be mounted shelves on the back wall of the Rover above the motors to allow additional storage for boards and other components. Also on the inside walls of the Rover will be a 3D printed air vent to allow air intake and a fan that will blow warm air out of the chassis. The side of the Rover facing the Docking Station will have the female end of the charging contacts housed inside the chassis and sitting flush with the edge of the chassis to prevent debris from getting on the contact. The male end will be on the inside wall of the Docking Station and will insert into the female port of the contacts. These will be manufactured from a conductive metal by the college's metal shop technicians. They will make contact via the designed tension of the female port. No springs will be necessary because the width tolerance of the female port will be smaller than the width of the male end.

Above the charging contact on the Rover will be the Raspberry Pi camera that will be responsible for computer vision navigation using ArUco tag recognition based on the tag mounted on the Docking Station. This camera will be mounted in a clear box to allow line-of-sight but also protect from degree and weather conditions.

Attached to the bottom of the Rover will be a thin, flat plate of metal behaving as our ground plate. There will be one point of contact that reaches into the Rover chassis that all grounds will connect to. This will prevent ground loops.

Based on the weights of known components, materials, and estimates for unknown quantities of mounting hardware, the weight of the Rover will be around 25 pounds.

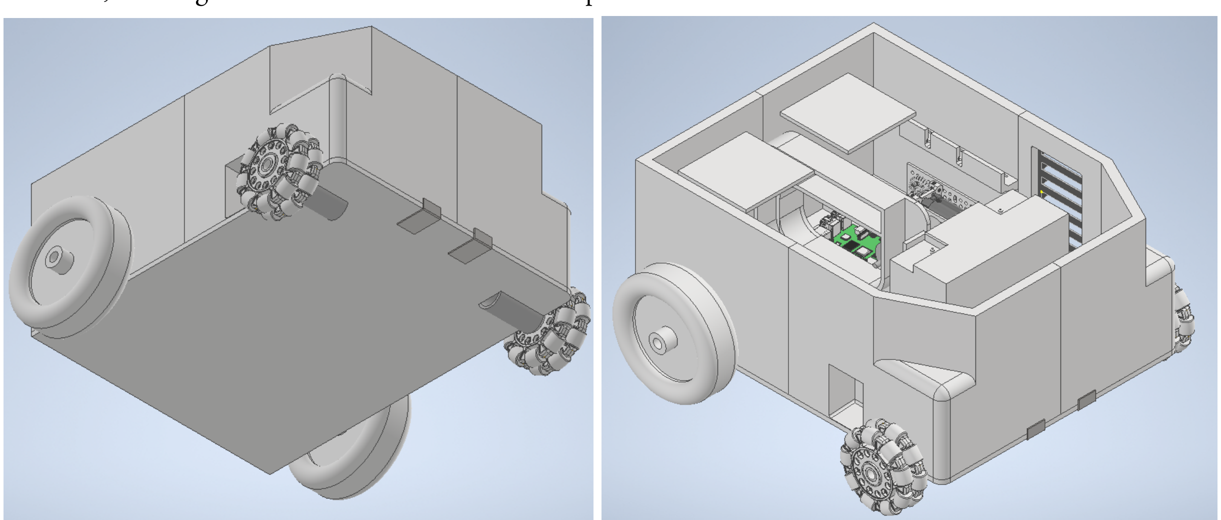


Figure 6a & 6b: Rover Bottom and Top View Without the Roof

Internal Components:
Inside the Rover will be a 12.8 volt, 10 amp-hour battery and an associated power distribution board and charge controller. Both drive motors will be connected to a pair of bevel gears and an axle mounted to a bracket meant to take the weight off of the motors shaft and onto the bearings of that bracket. There will be an Arduino Mega onboard that controls the transceivers and communication, motor signals, and the main FSM. For the motors, there will be a motor controller board. The Raspberry Pi, associated camera, cone extension motor, and the RF transceiver will be onboard as well.
The Rover will also include a main power switch located on the outside of the chassis in order to turn off all power to the Rover's components. All current that will flow through the Rover will go through this switch. It is located outside the chassis for easy access and safety and will be inside its own inlet in order to keep it out of the weather.

The main visual cue of the Rover will be our extendable cone. The cone consists of interlocking segments that will be vertically extended out of the top of the Rover when it is out in the driveway. The segments will have reflective tape on them to warn oncoming drivers. The cone's extension will be operated by a locking chain system that is driven by a motor. The system stores the chain in a horizontal position and is driven into and through a guiding channel to lock each segment in a vertical orientation. The cone will extend 18 inches high in addition to the Rover's ride height. A gear will drive our locking chains up when extending and back down into the Rover when retracting. The gear will be spun by a 12 volt motor with a built-in encoder and controlled by our onboard Arduino. The cone should fully extend in 6 seconds and fully retract in 6 seconds.

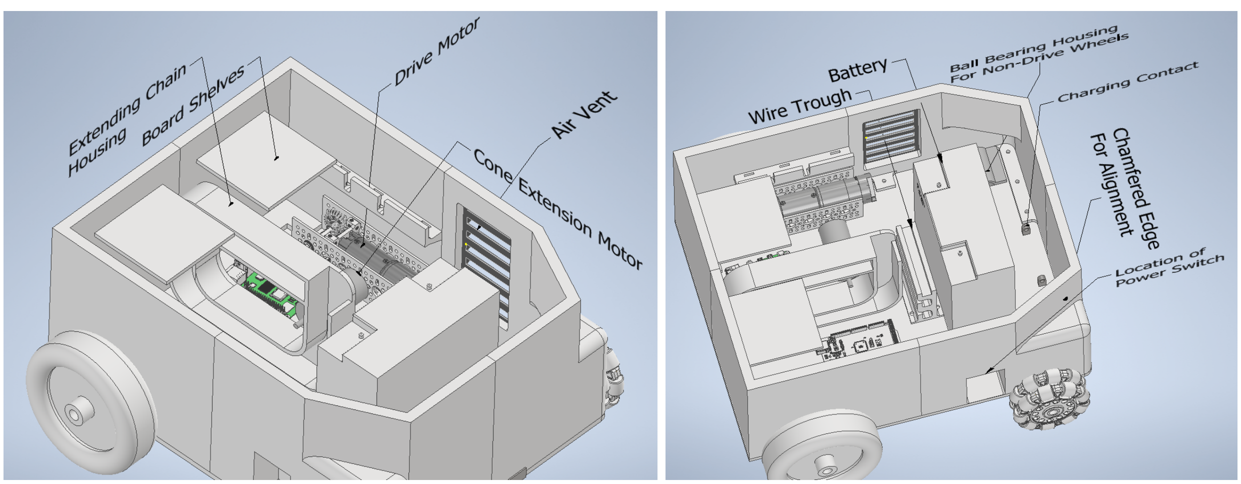
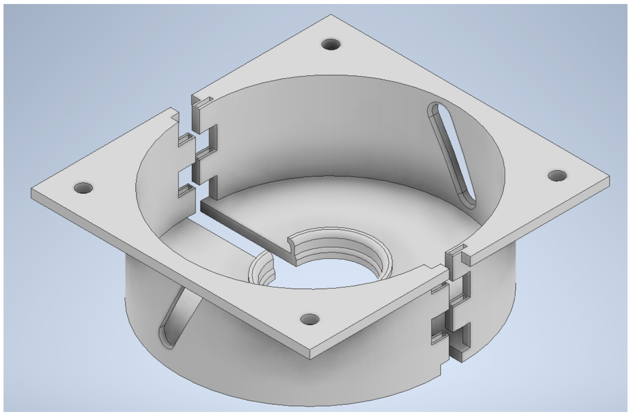


Figure 7a & 7b: Rover Internals



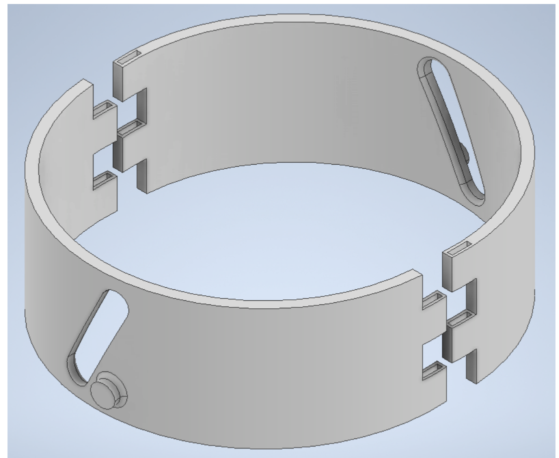


Figure 8a & 8b: Cone Roof Segment and Other Interlocking Segment

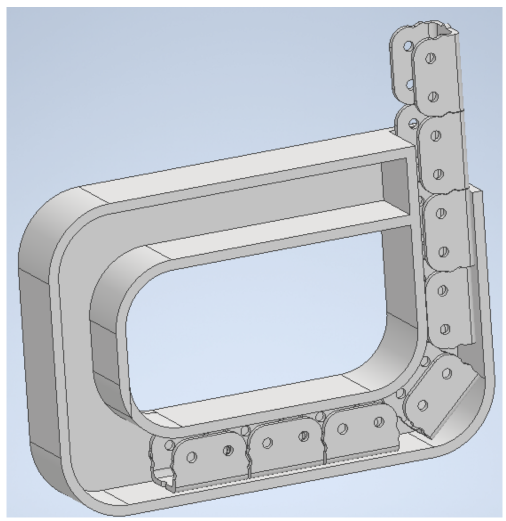


Figure 9: Chain System

Weather:

To keep water and debris out of the Rover, all gaps visible to the outside of the Rover will contain rubber gaskets or a “fender/awning” design to divert water, dust, etc.

To keep rain out of the cone segments on the roof of the Rover, the bottom cone segment will sit below the surface of the Rover like a bowl. It will have a drainage pipe on bottom that connects through to the bottom of the Rover’s chassis to drain out the bottom. The “bowl-like” cone segment will have a hole in the bottom to allow the locking chain to pass up through. To prevent water from getting down into the Rover through the chain’s hole, the chain will be covered by a waterproof material (like a tarp) going around it and attached to the chain passage hole and top segment of the cone.

The computer vision camera will be encased in a clear plastic container to allow sight but repel water. Also, the reason the female charging contact on the Rover is flush to the surface is because it will prevent water or debris from getting on the contact.

To combat hot weather, both the Docking Station and the Rover will have slotted vents and a fan on opposite sides of each other. The vent will have an overhang to allow water to run off of them rather than get inside the chassis. The fans will blow the hot air inside the chassis outside, creating a channel of airflow.

The power distribution boards will also be equipped with a heat sink to keep them at a reasonable operating temperature. A thermal analysis will be performed once the board is here and is able to be tested. We are also looking into adhesion methods. It is the only board modeled in metric units in order to maintain precision with the chip specifications. This component will be manufactured by the machine shop out of aluminum.

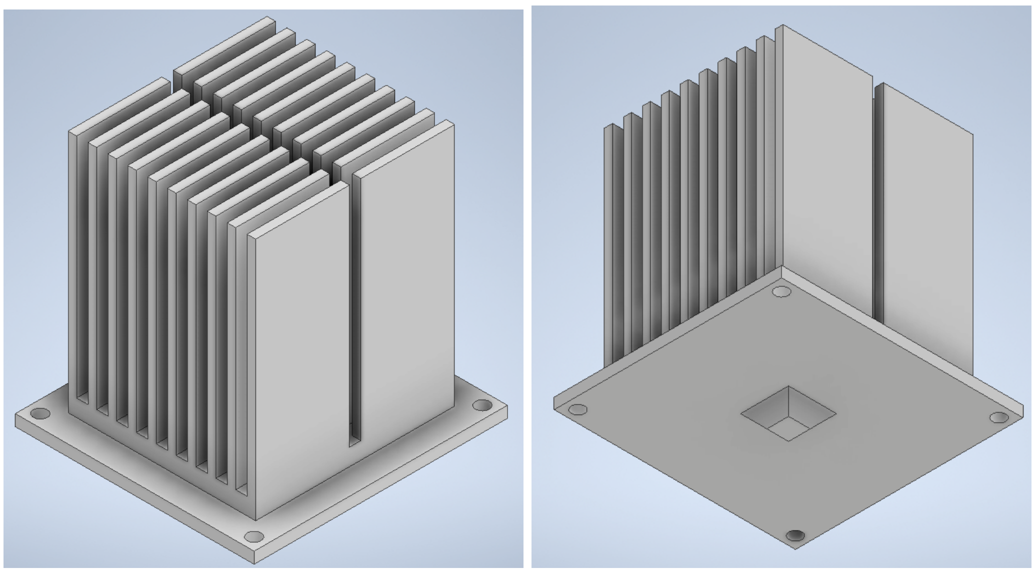


Figure 10a & 10b: Heatsink Models

ELECTRICAL DESIGN

Top-Level Electrical Design

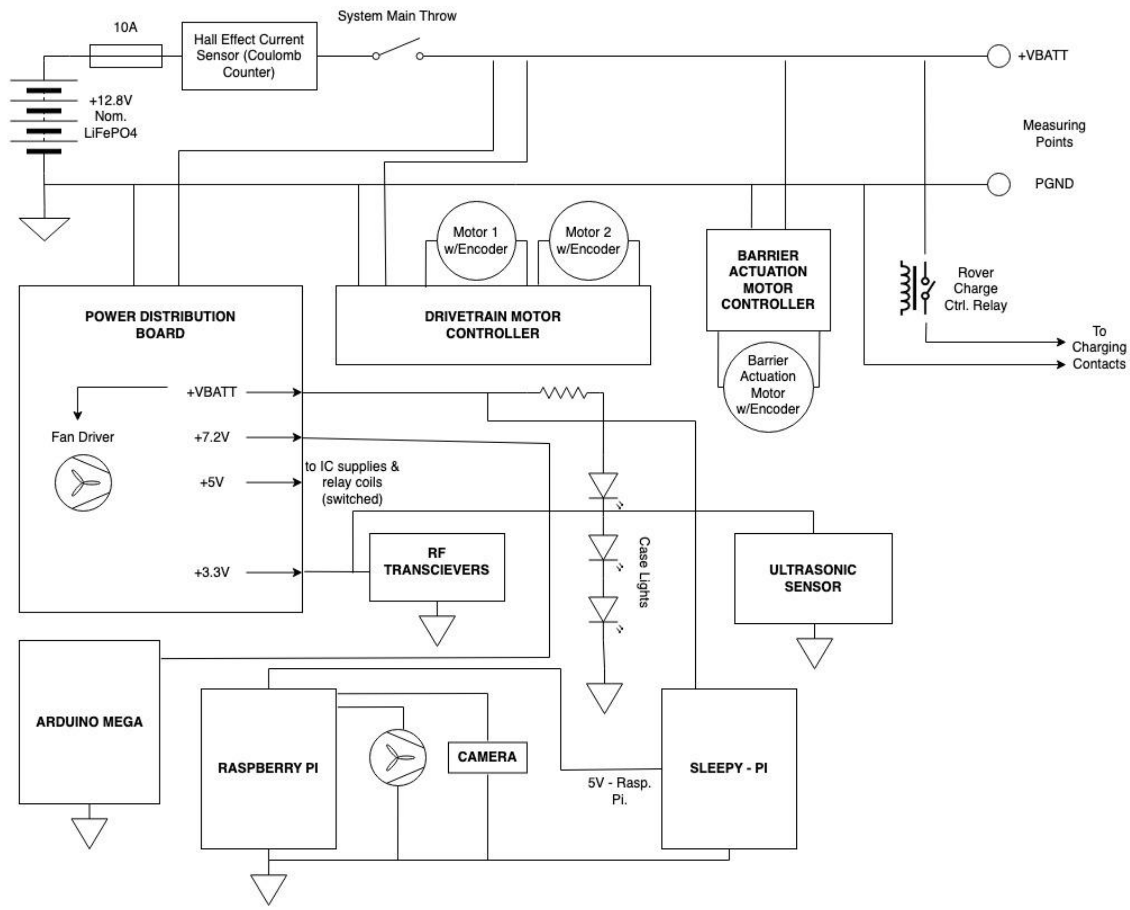


Figure 11 - Top-Level Rover Electrical Design

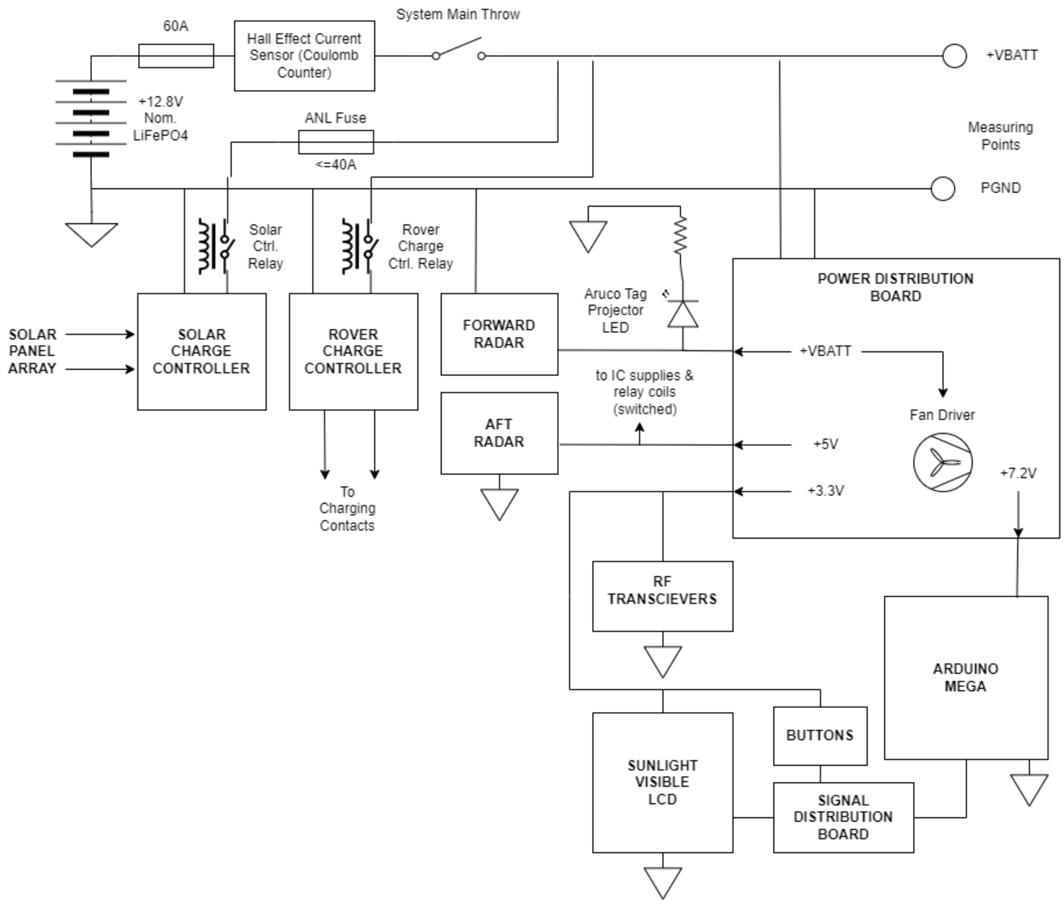


Figure 12 - Top-Level Docking Station Electrical Design

<u>Accumulator Systems</u>

The RFPP calls for a system featuring an solar powered charging station and an untethered mobile robot. In order to make this possible, our charging station, the Docking Station, and our robot, the Rover, both feature dedicated accumulator systems. Each will have its own lithium iron phosphate battery, the docking station with 50Ah of capacity and the rover with 10Ah. Both batteries comprise four sets of LiFePO4 cells in serial connection, with 3.2V nominal voltage each, summing to a battery nominal voltage of 12.8V. Both batteries feature a commercially designed battery management system which will be relied upon for thermal monitoring, and will be used in tandem with system fuses to protect against overcharge/discharge. Both batteries are manufactured by Eco-Worthy, a manufacturer specializing in LiFePO4 chemistry batteries for RV, solar and general outdoor applications.

For both the rover and the docking station, a surface-mounted hall effect sensor will be utilized as a coulomb counter. This serves three primary purposes. Firstly, it allows us to use it as a testing measure to understand power consumption of the system. Secondly, it allows us to know when the rover is charging. And lastly, it allows us to use amperage discharged/charged over time to calculate a state of charge (SOC) for the docking station and rover batteries. This is important information for the firmware team to make decisions as applicable during system operation, see the Embedded Control section below for more information.

Power Generation & Distribution

Solar Power:

The docking station will derive its source of power from a 100W monocrystalline solar panel manufactured by Renogy. Power will be delivered through a maximum power point tracking (MPPT) solar charge controller manufactured by the same company, provided in the same solar 'kit'. Power coming both to and from the charge controller will be fused with fuses included in the solar kit. Solar power incoming to the docking station battery is switched through a solar main relay. Solar power incoming to the docking station battery is switched through control by the firmware, allowing for additional safety and control of when the battery can be charged. Normal operation will call for the solar charge controller to control charging.

Rover Battery Charge Controller:

For charging the rover battery off of the docking station battery, a CV/CC LiFePO4 charge controller is employed. This is required for boosting the battery voltage of the docking station battery up to at most 14.5V to charge the rover battery. This controller features a two stage charge controlling algorithm for the bulk and absorb phases of the charging cycle, and has a controllable 2, 4 or 8A limit via an external potentiometer. This component will be located on the docking station, not the rover. Power being delivered to the rover will be switched at both ends to ensure that the exposed metallic contacts do not have a live voltage across them when not connected in the dedicated rover charging state.

Power Distribution Boards - Rover:

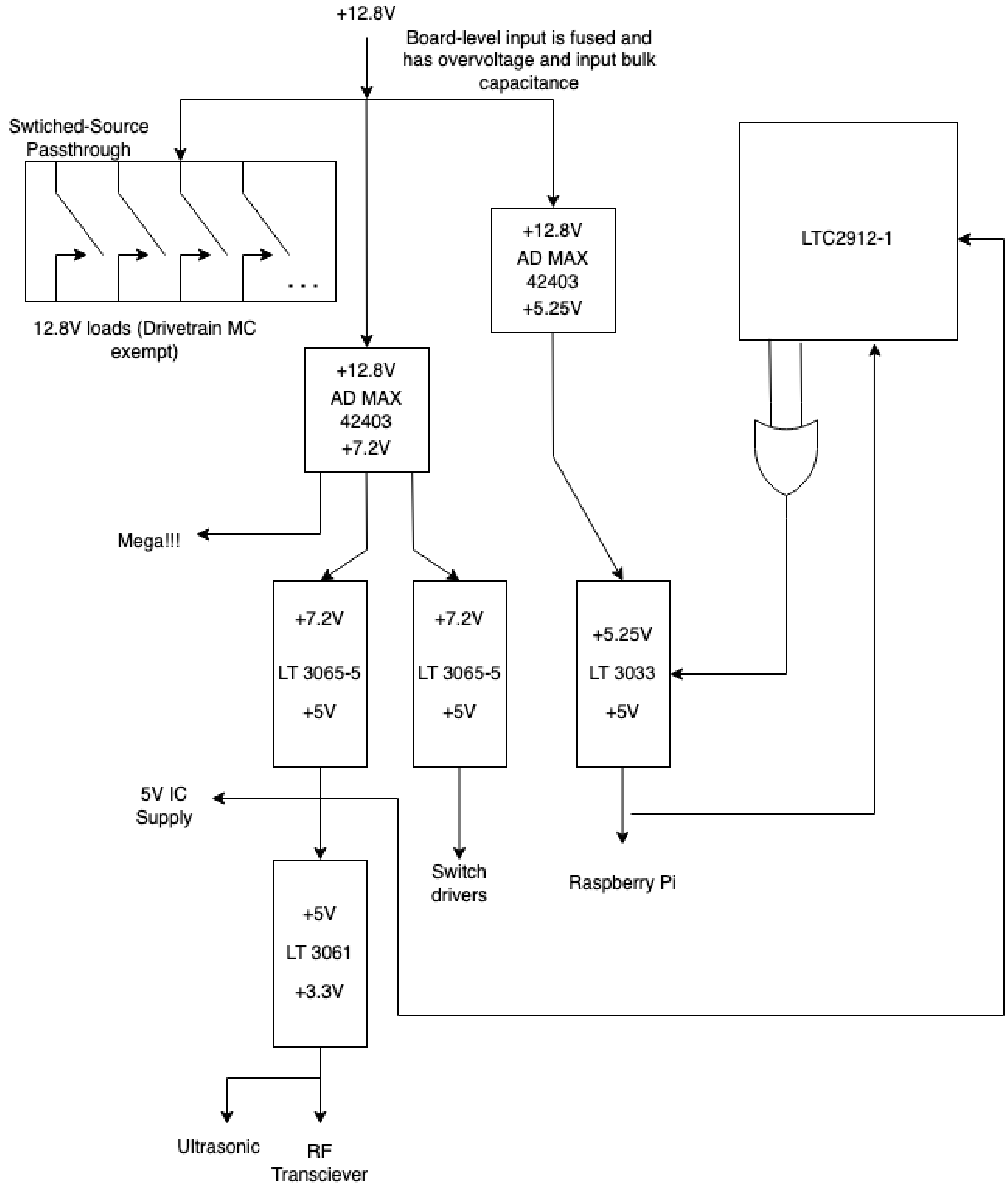
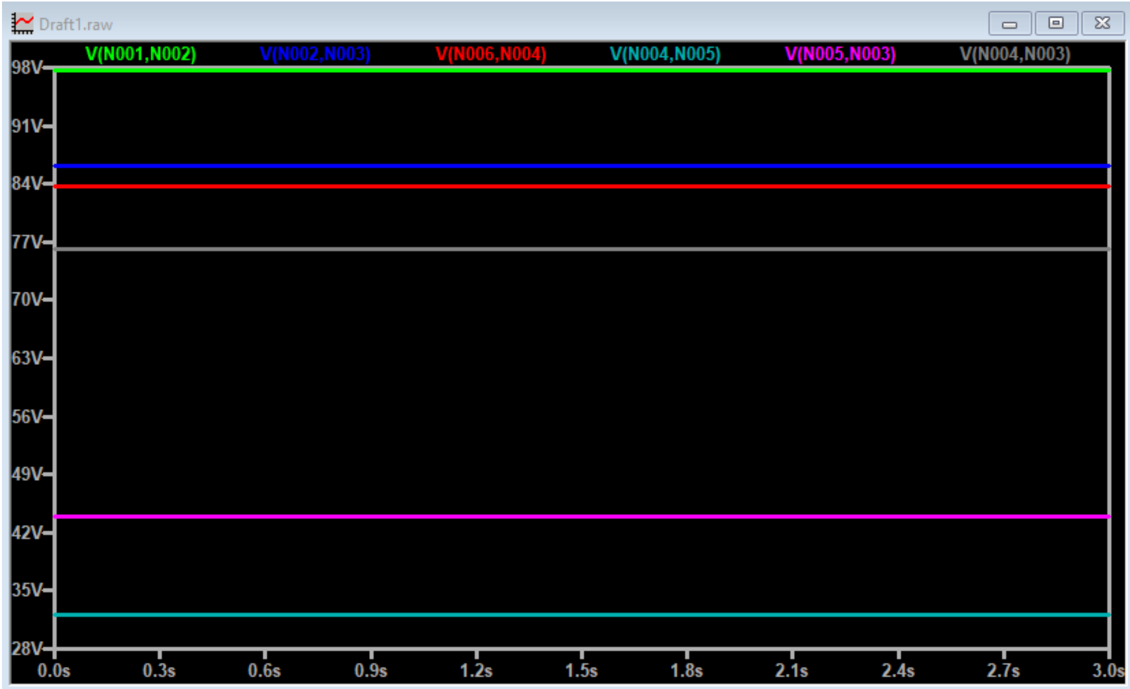


Figure 13 - Rover Power Distribution Top Level

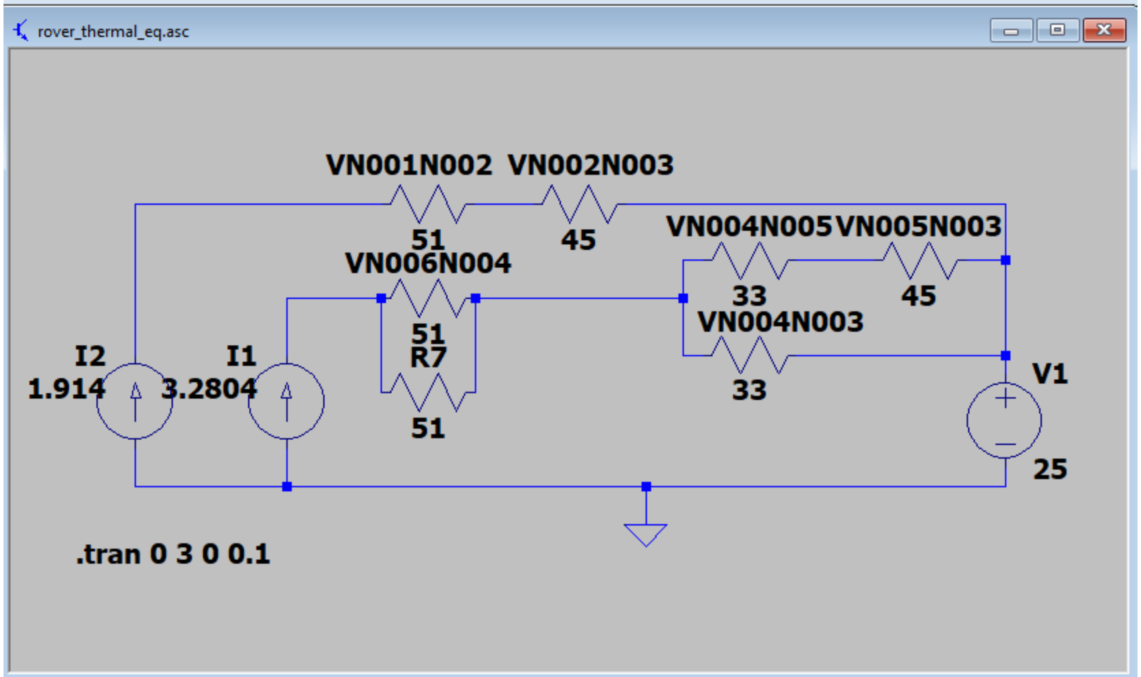
The rover power distribution board will feature six voltage regulators providing 4 different output voltage levels. Firstly, the battery power will be provided to loads not requiring regulation through a switched supply. These include the rover case lights, cooling fan and the sleepy-pi. Both the drivetrain and the barrier actuation motor controllers are exempt from this supply. This is due to the fact that the motor controllers themselves already act as switching for their respective loads. This way the high current levels that will be going through these boards can be avoided from also having to go through the power distribution board and the design can avoid unnecessary overcomplication.

Providing power to the regulating sources are two separate power paths, one specially designed for the Raspberry Pi logic chip, and one for everything else. Important to note is that this power path will no longer be used to power the Raspberry Pi. The Raspberry Pi will be powered by the regulation onboard the Sleepy-Pi board. The power structure will however, for the time being, remain on the board for the purpose of future expandability. For the Raspberry Pi power path, two regulators transform the battery voltage to a stable 5V supply output. The Raspberry Pi can consume up to 3A of current at 5V and has a +/-5% voltage supply tolerance. Additionally, since this component will only be used for our return navigation protocol, and otherwise will be put into a low power state by the companion board, the 'Sleepy-Pi', this load can be considered a stepping load. With these considerations in mind, the power supply network for this part needs to have a good transient response, extremely low output ripple, and high current capability (relatively). For this task, a system comprising a switching regulator powering a low-dropout (LDO) linear regulator has been designed. The Analog Devices MAX42403 is a switching regulator with a wide input voltage range, fully inclusive of the ~14.2-10V range our battery can output. It has an adjustable output, which has been set at 5.25V and can supply 3.5A of current in optimal thermal conditions. See following sections below for discussions of thermal performance. The Linear Technology LT 3033 LDO linear regulator has been chosen for this highly restrictive role of powering the logic chip. This regulator has a dropout of merely 90mV at no load and ranges to up to 200mV at a max load of 3A over temperature. The input voltage has been selected carefully to ensure that in the worst conditions, full load at high junction temperature, the dropout minimum requirement will not be violated. The adjustable output of the MAX42403 allows this precise output voltage; this will be further controlled with the voltage output adjustment feedback divider having 0.1% tolerance resistors. This will require careful testing when hardware arrives in the spring, although granted the design can perform as specified this power path for the Raspberry Pi should have 85.7% efficiency at maximum load, and 90.5% efficiency at a load of 1A. A high output capacitance will be installed at the output of the LT 3033 to reduce output ripple. Three different capacitors of varying orders of magnitude will be installed in parallel to improve performance over frequency (47uF, 10uF & 100nF). Critical to the reliable operation of the Raspberry Pi is keeping the input voltage within its specified tolerance, otherwise its operation can enter an undefined state and lead to system-level failure. An output power supply monitor/supervisor will be employed to control the output (ON/OFF) of the LT 3033. The LTC2912-1 will be used for this purpose. It has a pair of outputs that indicate under/over voltage respectively and can be put through OR'ing logic. This is what will be done to drive the shutdown pin on the LT 3033. The active low output will shutdown when both the active low undervoltage and overvoltage open-drain outputs on the power monitor go low. This is based on a voltage 'operating window' which is programmable via resistors through a procedure followed from the datasheet application information section. The reset will happen once the output voltage of the regulator has returned to the operating window and a programmable delay has occurred, which is 200ms. This is set by a capacitor. This power monitor solution should keep Raspberry Pi in a defined operating state and not stress the logic chip with over or under voltage conditions. This solution does not require active firmware control.

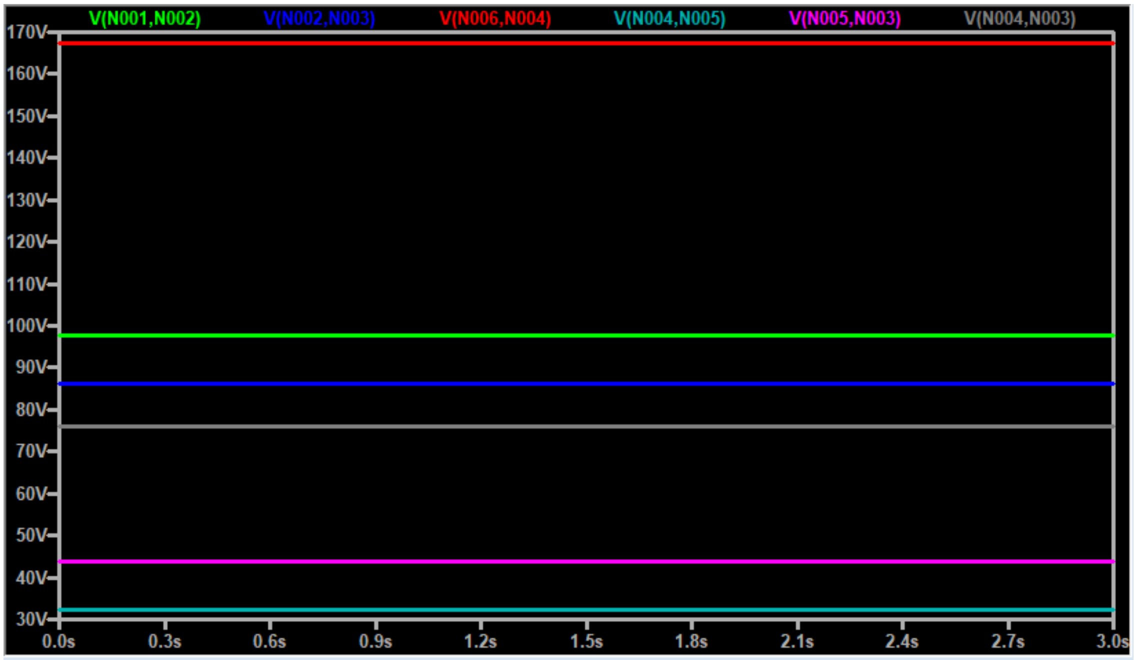
The other power path branches off another MAX42403 switching regulator, this time providing a 7.2V output. This supply will be used to power the Arduino Mega, and the other regulators down the chain. The Arduino Mega has an onboard regulator which supplies its logic chip with its required 5V. The input voltage range is rated as 7-12V normally, with the potential to overheat the chip if strayed too high above 12V and the potential to have undefined behavior if strayed too low below 7V. In order to maximize efficiency, but still be conscious of reliability, 7.2V was chosen to go just above the rated minimum. Choosing precise output voltage for this switching supply will follow the same method as its sister chip on the Raspberry Pi power path, 0.1% tolerance resistors. The ability of this supply to have an accurate adjustable output has an enormous impact on this design. There are several loads on the rover requiring a 5V supply voltage, these include driving switching relays and FETS and powering the array of IC's that will be scattered throughout the whole system. This supply is meant to accommodate expansion of electrical requirements as the project development progresses. Two LT-3065-5 linear regulators will be utilized for this purpose, each capable of outputting 500mA in optimal thermal conditions. One of these supplies will drive switching FET's and relays. This load type has been separated out due to its stepping nature and an effort to avoid the negative transient response effects this has on other loads. The other will supply all the IC supplies that require 5V, this includes the LTC2912-1 power monitor discussed above. Additionally, this linear regulator will supply another linear regulator, itself outputting 3.3V. This will be the LT 3061 which supplies the RF transceivers and ultrasonic sensor. This supply was chosen on the criteria of low output voltage ripple without the need for excessive output capacitance. Wireless communication in our system has been identified as a susceptible source of failure. This is not due to design decisions made by our team, but seen as a common trend in commercial products. Due to this, the RF Transceivers require a steady input voltage in an effort to reduce sources of error regarding these devices. Both these loads and the ultrasonic sensor draw below 15mA at maximum. The LT 3061 can supply up to 100mA in optimal thermal conditions, which allows expandability for powering IC's or other low-level loads requiring 3.3V supply.



Draft1.raw
V(N001,N002)
V(N002,N003)
V(N006,N004)
V(N004,N005)
V(N005,N003)
V(N004,N003)
98V
91V
84V
77V
70V
63V
56V
49V
42V
35V
28V
0.0s
0.3s
0.6s
0.9s
1.2s
1.5s
1.8s
2.1s
2.4s
2.7s
3.0s



rover_thermal_eq.asc
VN001N002
51
VN002N003
45
VN004N005
33
VN005N003
45
VN006N004
51
R7
51
VN004N003
33
I2
1.914
I1
3.2804
V1
25
.tran 0 3 0 0.1



Figures 14a, 14b & 14c - Rover Power Distribution Board Simulated Thermal Performance with/without Heatsink on R-VN006N004 & Thermal Equivalent Circuit

A full thermal analysis has been performed for this circuit board, utilizing thermal equivalent circuit techniques. In the figures above, the first graph shows temperatures in Celsius (shown as voltage for the electrical circuit equivalent) at the maximum load condition with a heatsink provided for the MAX42403 on the non-Raspberry Pi power path. The last figure shows it without the heatsink, and the middle figure shows the circuit model. Calculations were performed with each regulator outputting ideal voltages (no ripple, at specified output voltage by datasheet or adjustment resistor divider), rover battery voltage at nominal 12.8V where applicable, current draw at maximum rated by the respective datasheet, and the assumption that the PCB will provide no parallel thermal resistance. This was meant to be a truly worst case scenario.

The power dissipation for the Raspberry-Pi power path was calculated as 1.914W and the general path at 3.2804W. The results show generally 'warm' temperatures, as compared to the maximum rated junction temperatures. The worst performers are the pair of MAX42403 switching supplies, which is expected due to their position as the principle power suppliers for all other regulators. The root switching supply on the Raspberry-pi path will not require a heatsink, although it will run hot (~100C) at worst case. The other root switching supply will require a heatsink, running at nearly 170C at full load. Due to this, the proposal is to have a heatsink on both these parts to improve reliability. The first figure above shows the thermal performance with a heatsink on just the general power path root supply, with the assumption that the thermal resistance is equivalent to the thermal resistance of the chip (junction to ambient) itself. This heatsink is a custom-designed part made by the mechanical team, described in the mechanical design section above.

Power Distribution Boards - Docking Station:

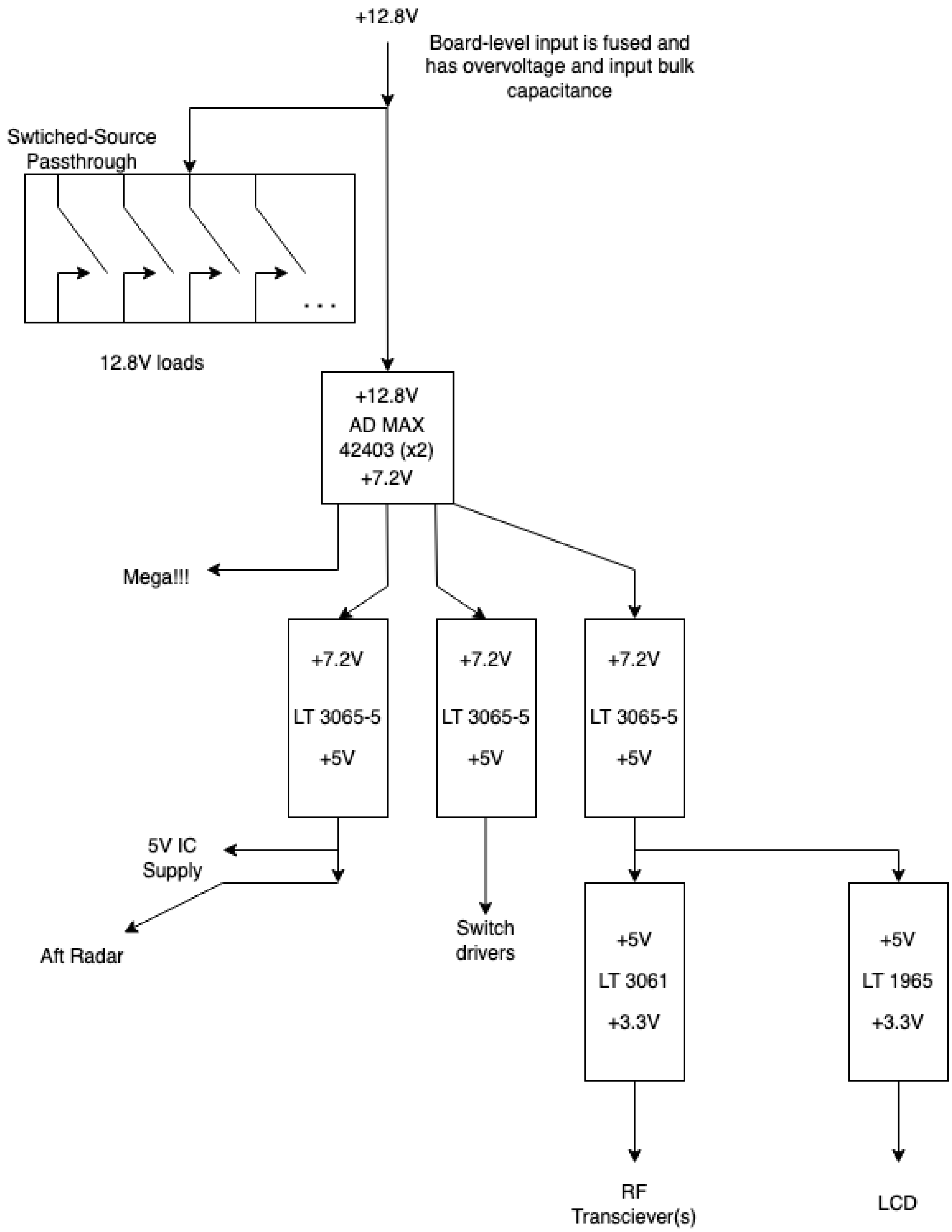
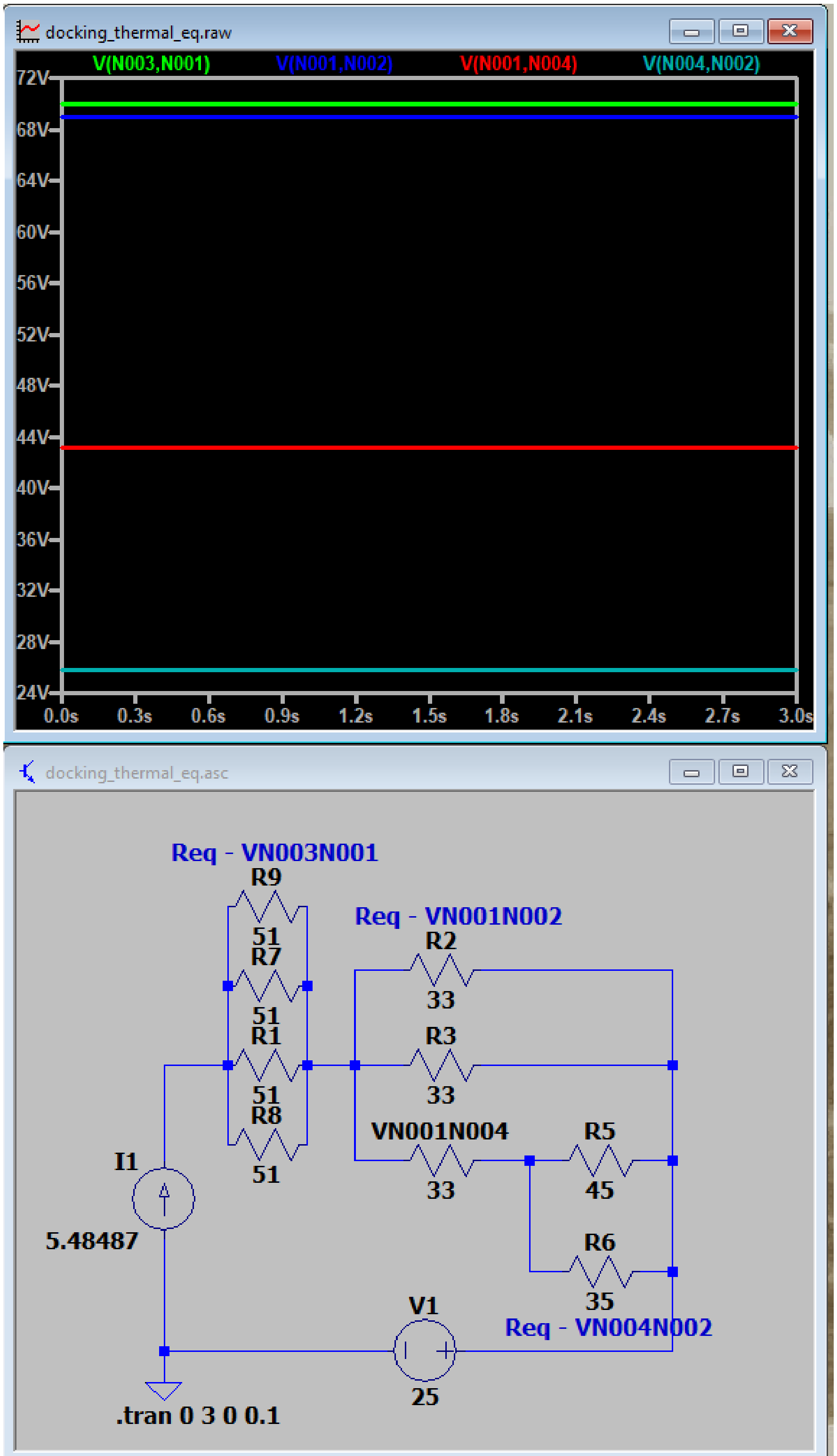


Figure 15 - Docking Station Power Distribution Board

The docking station power distribution board shares much in common with the rover board. All but one of the supplies are not new, and are carried over from the other board. While this was done with the aim to simplify the design process, it was not chosen until after the load requirements were characterized and the informed decision was made that the supply in question was able to meet requirements. This board features the same principle of switched supply of the battery voltage, powering loads such as the Aruco Tag Projector light (for nighttime visibility), forward radar module & case fan. Besides this, the docking station board features only one regulated power path as opposed to the two on the rover board. The same switching supply, the MAX42403, serves as the supply for the Arduino Mega and the root for all other supplies. This comes with the same pros and cons as on the rover board, particularly with the large downside of thermal performance. Due to this very large thermal performance deficit, two of these supplies will be parallelized to reduce the current supplied by each. This also will obviously double the current output capability, which allows for significant expandability. See below for the discussion on thermal performance. From this pair of switching root supplies, three LT-3065-5 5V supplies will be linked in parallel supplying a variety of loads. Like in the rover board, separate supplies will drive switching loads and IC supplies respectively. In addition to the list of loads for the IC driver however, is the aft radar module which consumes ~100mA in regular operation. A third linear regulator of this type is provided to drive the 3.3V linear regulators down the chain. The use of the LT-3065-5 in this case allows for significant expandability in terms of power consumption in the future, which is anticipated to affect the docking station more than the rover due to its solar supply allowing for this. To supply the RF transceivers, the same linear regulator, the LT 3061, will be used. However, to power the LCD a new linear supply, the LT 1965 will be used. This supply can push up to 1.1A and thus will be able to handle the ~200mA requirements of the display without dissipating too much heat.



Figures 16a & 16b - Docking Station Power Distribution Board Simulated Thermal Performance & Thermal Equivalent Circuit

A full thermal analysis has been performed for this circuit board, utilizing thermal equivalent circuit techniques, just as for the rover board. The above graph shows the thermal performance at the same worst case load and the same assumptions as with the rover thermal analysis. However, important to note is that the pair of MAX42403's (Req - VN003N001) each have a heatsink matching the thermal resistance of the chip (junction to ambient) itself. From these results, it is clear that this system will run much cooler than the rover generally, and agrees with the proclamation made earlier that there is significant room for expansion of loads on this power distribution structure.

Important to note for both thermal analyses, the layout of these boards will include attention to using the copper traces on the PCB's to improve thermal performance. This includes thermal vias and layout techniques included in the respective datasheet layout guidelines sections. Thus, these figures are considered to be inclusive of real-world performance due to their 'worst-case' nature.

Docking Process Hardware

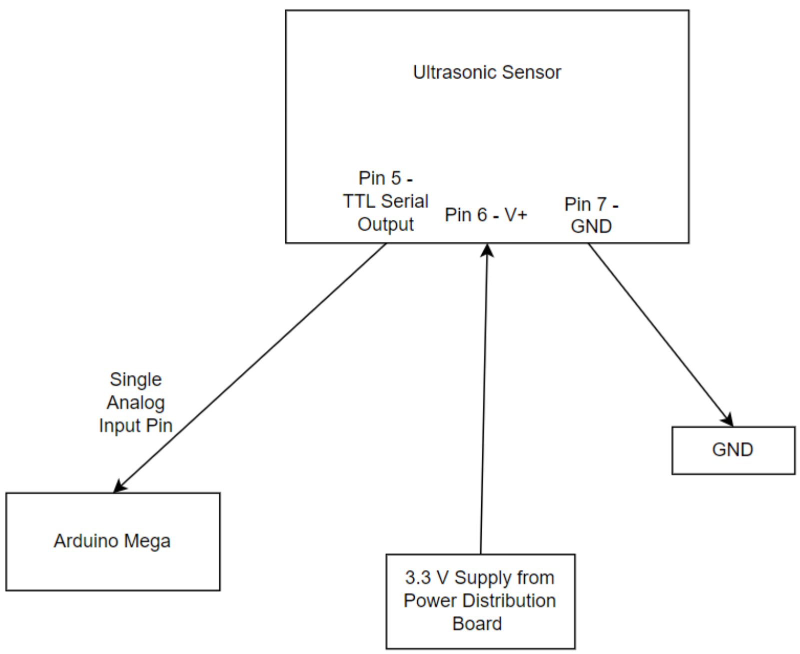


Figure 17 - Ultrasonic Sensor Top Level I/O

When the Rover reaches the entrance of the Docking Station during the navigation process of returning from the driveway, additional hardware is needed to help complete the docking process. To aid in this, a MB1613 HRLV-ShortRange-EZ1 ultrasonic sensor will be used. It will be mounted in a forward-facing direction located at the front of the rover in order to give proximity distance calculations between the front of the rover and the back wall of the docking station. This serves the purpose of allowing the rover to slow down as it approaches the back wall when it crosses certain distance thresholds detected by the sensor. The inside of the docking station is only about 14 inches deep, so the maximum sensing distance was set for about 2 feet. This sensor accounts for a maximum of 5 meters, plenty for the required distance. In addition, the sensor needed to have a relatively quick response time, of which this has one of 100 milliseconds, which is plenty for its purpose. Finally, due to the space restrictions on the rover, the sensor needed to be rather small so it didn't take too much space up at the front. This meets that requirement as well, with the dimensions being 1.5" x 0.875" with a depth of 3.18 mm.

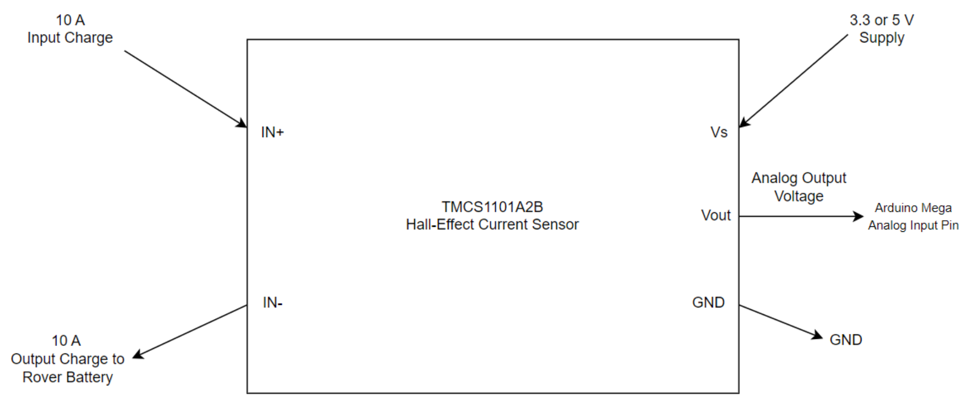


Figure 18 - Hall-Effect Sensor Top-Level I/O

Additionally, a Texas Instruments surface-mounted hall-effect current sensor will be implemented into the battery charging system on the rover to detect when charge is coming into the battery. This will serve as a signal if the rover is properly connected to the charging system at the back of the docking station or not. It has been selected to be able to intake a large amount of current maximum of 22.5 amperes since the discharge coming from the docking station battery will have a large amount of current of around 10 amperes. Something that's always a concern with a large amount of current running through a piece of hardware like this is the potential loss of charge going to the battery due to high temperatures. However, at a worst case scenario of maximum temperature of 150 °C, the maximum power dissipation loss would only be about 1.2 Watts. Additionally, on the PCB board the sensor will be placed on, there are a few things that will be done to help improve thermal performance. Large copper planes will be used for the input current path as well as the isolated power planes. Thermal via farms will be placed around the isolated input current. Finally, overall heavier copper PCB construction for the board will be necessary.

<u>Display Unit</u>

The display unit needed to meet one major requirement, which was being fully sunlight visible. For full sunlight visibility, a potential screen unit needed to have a brightness of 1000 nits. The display unit will be the E50GE TFT LCD display from Focus LCDs. With a brightness of 1100 nits, it meets the requirement for full sunlight visibility. The LCD display has a TFT controller, the ST7262, which is capable of being interfaced with Arduino, and there are existing GFX libraries for RGB displays specifically and GFX in general that can be used to create the GUI on the display. The display panel has its pin-out circuitry in a 40-pin FPC at the bottom of the display, which can be connected with through existing FPC connector boards. The major requirement when making this connection is it needs to be handled in a way to reduce any potential stress as much as possible. Due to recent unexpected developments, a full connection layout between the display and the Arduino is not available for this proposal.

<u>Drivetrain - Relevant Code: *NFPA 70: National Electrical Code Article 430*</u>

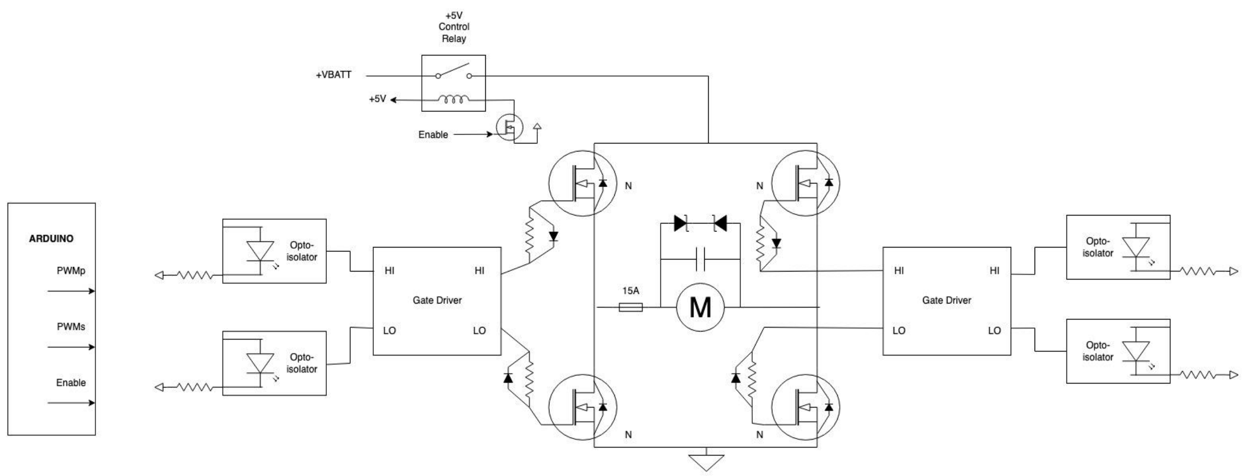


Figure 19 - Drivetrain Motor Controller Diagram - Single H-Bridge

The rover drivetrain comprises a pair 5203 Series Yellow Jacket brushed DC motors. These motors have a 99.5:1 gear ratio producing a stall torque of 133.2kg*cm at 9.2A/12V supply. These motors will be driven with a differential drive control schema, where modulation of the wheel velocity for each side constitutes the turning mechanism. Each motor will be driven by the dedicated drivetrain motor controller. This board is under development as an inter-team effort with Team Beta. The board features a pair of H-bridge motor drivers utilizing exclusively N-channel MOSFET surface mount devices, a safety control relay switching all power to both H-bridges, and a dedicated voltage regulator producing 5V to power onboard IC's. This decision was made to make the board as "all in one" as possible for ease of use by future senior design teams. The board will have overvoltage and overcurrent protection for each H-bridge; a zener diode across the motor leads in reverse polarity configuration, and a properly sized fuse inline with the positive motor lead. Fuse sizing was based on research performed regarding this type of application, arriving at a figure of a line fuse rated for 1.7x maximum current, which would be the stall current of 9.2A. The fuse is thus rated at 15A which is just below the 15.64A calculation. Power to the pair of H-bridges is additionally switched through a dedicated relay to enable further control of supplying current to the motors. This additionally serves as a failsafe precaution in case the MOSFET gate control is broken down for any reason. The relay current carrying capacity is rated significantly higher than the maximum expected current (stall current of Team Alpha motor) due to two reasons: Firstly, to enable future teams expandability when using this design. Lastly, since the relay has an inductive load (mostly) there will be a sizable surge current to start up these motors and this is to handle this transient startup surge current.

Crucial to the design of this board is protection of the control circuitry and compatibility with power FET's. Specifically, for protection of the Arduino. It is important that these power-FET's can actually be driven fully, which is not going to be the case from the outputs on the Arduino, and that in the event of a gate-breakdown the Arduino is protected. For these reasons, a gate driver and an opto-isolator will be installed in series with the FET gates. The gate driver is a half bridge dual driver/charge pump. Meaning one gate driver will drive a full half-bridge, thus two drivers per H-bridge and four of these chips on the board as a whole. This will provide the power to charge the gate to source capacitance when turning on the transistors and perform required voltage level shifting. Based on the power and voltage requirements for switching, an exclusively NMOS implementation will be adopted. Before the gate driver is the opto-isolator with a push-pull totem pole output. This part has 5kVrms isolation rating and will serve as the final fail-safe to protect the Arduino in the case of a serious motor controller failure.

Each individual motor will be driven by a pair of PWM signals in junction with an always on (when conduction through the motor is desired) signal from the Arduino. The always on signal will be delivered to the low-side switch of the main conduction path. The principal signal is the primary control signal for modulating current, delivered to the conducting FET of the high-side pair of the H-bridge. The phase shifted signal is used to control the low-side FET off the main conduction path. This will establish a conducting path from power to ground through the motor during the ON cycle, and a discharge path during the OFF cycle.

An important part of the design process is choosing the PWM frequency, for which there are several considerations. Firstly, a higher PWM frequency will yield less power loss. However, there is a finite limit determined by the switches used in the H-bridge design. It is additionally limited by the desire to run the motors in a constant conduction mode (CCM). Research has indicated that a minimum switching frequency of approximately 5*1/(2*pi*tau), where tau is the inductive time constant, is highly desirable to reduce output ripple. Thus, the minimum frequency can be defined in terms of the inductive time constant, L/R where L is the motor armature inductance and R is the armature resistance. The two motors vary significantly in R but are relatively constant with respect to L. Motor #2 (marked on the product label with a dot) has a time constant of ~52 microseconds, and motor #1 ~37.2 microseconds. These impose restrictions on the minimum frequency of a 100% duty cycle PWM frequency, 15.314KHz and 21.39KHz respectively. Regarding duty cycle, the output voltage ripple will be at a maximum at a precisely 50% duty ratio, so this must be avoided.

Additionally, due to the fact that the switching timing will never be perfect, a PWM dead-zoning technique will be adopted. The dead-zoning will involve the phase shifted signal having a delay time on both its clock edges so that its ON time is shorter than the ON time of the normal signal by twice the delay period. This will prevent a potential period of time where a half bridge may be shorted. This deadtime is implemented in hardware on the half-bridge gate drivers.

For a multitude of reasons, some bulk input capacitance between the motor leads is required. This is to reduce voltage ripple, absorb back-emf and protect the switching FET's on the H-bridge. Selecting the appropriate input capacitance involves many factors, including the drain to source stress (Vdss) of the FET's, wire leads inductance and the equivalent series resistance (ESR) added into the circuit by this capacitance. An application note from Infineon on selecting bulk capacitance for single phase DC brushed motors proved supremely helpful for this part of the design process, providing a derivation for the following equation:

$$CDCLINK = \frac{1}{fPWM} \times \left(\frac{IMOTavg \times (1 - DC) \times DC}{\text{VSpp} - (IMOTavg + 0{,}5 \times IMOTpp) \times ESR} \right)$$

Figure 20 - Brushed DC Single-Phase Motor Input Bulk Capacitance Formula

All the information required to make this determination is not known yet, and is planned to be finalized shortly after submission of this document.

Lastly, the drivetrain motor controller will feature a pair of current sense amplifiers for measuring current flow into each motor. This is at the request of the firmware team. Part selection has not been finalized but the TI-INA240 is the current top candidate.

<u>Visual Barrier Actuation</u>

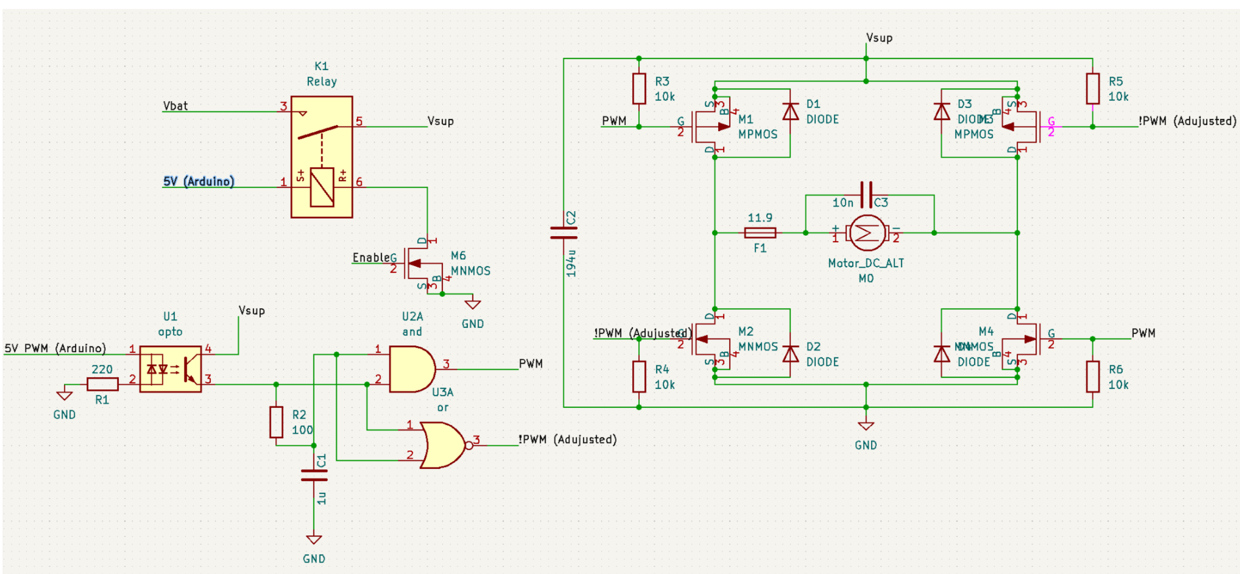


Figure 21 - Custom Barrier Actuator Driver Diagram

Onboard the rover is a visual cone that raises and lowers depending on the current operating mode (and position) of the robot. This cone is attached to a vertically raising chain stored horizontally within. This chain is actuated by a DFRobot FIT0186 12V gear motor connected to a custom controller. The controller will utilize lock anti-phase drive since it allows for PWM to control the entire bridge. Since the Arduino MEGA is the most important device on the rover, protection is instituted by sending the generated PWM signal through an opto-coupler. This PWM signal is then sent through a low pass filter and operated on by Schmitt Trigger AND and NOR gates, returning a PWM and opposite PWM simultaneously. A great benefit of this method is that by only using one PWM signal through one opto-coupler there is no worry about timing discrepancies in manufactured parts (by having to use a second coupler). By filtering, we create dead time, providing necessary shoot-through protection to the system.

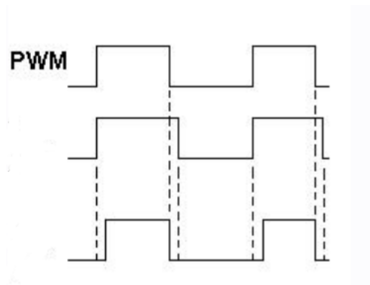


Figure 22 - Filtered PWM with Dead Time

The controller includes pull up/down resistors, as well as a control relay which affects the battery connection. Additionally, there is a line fuse which was determined to have an intended rating of 1.7 x average current. In this case, average current is seen as stall current, which is 7, leading to a required fuse rating of 11.9. Across the motor is a 10 nF capacitor in order to reduce ripple current through the bridge. Lastly, there is an input capacitor of 194uF which is intended to account for any spikes generated by the bridge. Calculations were completed expecting a 5% voltage ripple at a 30kHz switching frequency, both of which are subject to change. Additionally, at present, it is intended that Alpha & Omega Semiconductor AO4616 MOSFET pair chips will be used due to its maximum power and current ratings, as well as its acceptable r_{dson} value.

Switching frequency was determined by first calculating the time constant of the motor. Converting this to frequency yielded a minimum switching frequency of 8191.78 Hz. The value 30kHz was arrived at since it is well above ultrasonic ranges and should cause minimal audible noise.

The FIT0186 includes a quadrature hall encoder, which will interface directly with the Arduino MEGA through its interrupt and digital pins. Knowing this, the chain can be manipulated to always arrive at and reset from the same value throughout operations.

<u>Sensors</u>

Vehicle Identification Sensors:

In order to detect and authenticate vehicles, the system will employ two separate systems. For detecting authorized vehicles, a solution akin to a garage door 'clicker' will be used. This beacon will operate off the user pressing a button to send a 24GHz radio signal to a RF transceiver onboard the docking station. This signal will serve as the sole means of authenticating an authorized vehicle and is the motivation behind the authorized vehicle sensing distance. The device onboard each authorized vehicle will be a variant of the remote controller design, which is discussed in further detail in the appropriate section below.

Paramount to meeting the requirements of the RFPP is the ability to detect the presence of a vehicle, regardless of its status of authorization. This needs to be done to intercept unauthorized vehicles and to allow exiting vehicles passage. This task will be accomplished by a pair of radars, referred to as the forward and aft radars. The forward radar is an Omni-Presence OPS243-A doppler radar, operating about 24GHz in the NATO-defined K-band. This radar will detect vehicles and report their velocity as a pair of data points; heading and speed. This information will be utilized by the firmware team to analyze if a vehicle is present. The range of this radar on a reflective target, i.e.: a metallic vehicle approaching, is the motivation behind the unrecognized vehicle sensing distance. However, to allow exiting vehicles passage another, weaker, radar will be employed. This is the job of the aft radar, which is a Seeed Studio 24GHz millimeter-wave doppler radar. This radar, as opposed to the forward radar, has a wide viewing angle and short range (~30m maximum on reflective targets). It will be facing opposing the forward radar's line of sight and serve as a detector and interrupt to the current state of operation to allow the exiting car passage.

Navigation Sensors:

Critical to the computer vision system for return navigation is the system camera. This is the official Raspberry Pi Camera module V2. It is powered off the Raspberry Pi board and thus will only interface the rest of the system through the Pi. The camera will be the sole means by which the Aruco tag can be detected and used for navigation.

The final step of the return navigation system is the ultrasonic sensor previously described in the Docking Process Hardware. The ultrasonic sensor, placed at the front of the rover, will ping against the back wall of the docking station and give proximity distance calculations to the rover, signaling to slow down when it crosses set distance thresholds.

Current Sensors:

As previously mentioned in the Docking Process Hardware section, a surface-mount, hall-effect current sensor from Texas Instruments will be placed into the charging circuit going to the battery on the Rover. In addition to its previous purpose described there, it will also serve as a Coulomb counter to keep track of the electric charge and energy passing through the charging circuit on the Rover. This will be done by taking the analog voltage output from the sensor and converting it back into current in firmware. Additionally, a different, more robust hall-effect current sensor will be used on the docking station as a Coulomb counter due to the larger amount of current capable of being discharged out of the docking station than coming into the rover. This sensor will be the HCSP-1BS open-loop chassis-mounted current sensor from Piher. The current of the charging wire going out of the docking station has a much larger current of around 60 amperes, so to account for this the selected hall-effect current sensor will have a much higher maximum current intake of 200 amperes. Since this is a much larger amount of current intake, temperature concerns will be higher than for the TI current sensor. However, since this is a more over-the-top sensing solution, and since it's a through-hole and not a surface-mount, it is much more acclimated for the amount of current expected to be ran through it, and there's not the problem of the power dissipation that would come from running the current through a resistor. The other concern would be whether or not the hole through which the charging wires go through is large enough. The expected wire size are 8-gauge wires, which have a diameter of 3.264 millimeters. The hole of this current sensor is 6 mm x 20.5 mm, so the wires will fit through without problem. Similar to the TI current sensor, the output will be an analog output voltage that can be connected to an Arduino analog input pin, where the voltage calculation output can be calculated back into a power reading through firmware.

<u>Connectivity Design & Signal Integrity</u>

A crucial part of any electronics design is in the physical implementation of off-board wiring. This involves signal & power integrity, and organization. See the mechanical design section above for discussion on wire organization. Important for signal integrity is to avoid electromagnetic interference in proximity to digital communication lines. This will be mitigated by two main practices: Firstly, power lines (particularly those charging/discharging the battery) for both the docking station and rover will be located away from digital communication cables. Secondly, digital lines will be assembled or purchased in a twisted pair with shielding surrounding the cables in addition to a second layer of insulation. It is common practice to wind a wire pair 1.5x per inch, as this allows the electromagnetic fields produced to destructively interfere optimally to reduce their overall negative effect, the phenomenon known as crosstalk. For power integrity, an important note involves keeping the positive and negative power leads to the respective power distribution boards as close as possible to one another, and the leads as short as possible. This is to reduce the wire-self inductance so that the combination of low-ESR input capacitance and the wire lead self-inductance cannot start resonating and causing stability problems.

In the docking station, due to the LCD having its connections on a flex-pcb, a signal distribution board will be utilized to make effective connections between itself, the GUI controlling buttons, and the Arduino.

Additionally, per the customer's request as of 11.06.23, the RF communication transceivers will be implemented on a custom-designed board with the LoRa chipset. This board will be standardized and have one unit present for the Rover, Docking station, remote controller & one per vehicle authentication beacon. Due to how recent this change is, further details on board development are not available for this proposal.

<u>Distances Defined in RFPP - Amendments to Preliminary Proposal</u>

Due to changes in design during development since the preliminary proposal, the following amendments will be made for final specifications provided to the customer as requested:

1. The Marge System Setback (MSSB) is proposed to be 30m for the Normally Open mode.
2. The Marge System Setback (MSSB) is proposed to be 10m for the Normally Closed mode.

3. The Recognized Vehicle Sensing Distance (RVSD) target will be 500m.
4. The Unrecognized Vehicle Sensing Distance (UVSD) will be 120m.

<u>EMBEDDED CONTROL</u>

<u>Hardware</u>

The hardware for the embedded systems within the project will require

- RFM96W LoRa Radio Transceivers
- Arduino Nano
- USB Receptacle Power Only
- BQ25306RTER (Battery Charging IC)
- DMN2050L-7 (Nmos Transistor)
- Battery
- LED push buttons (x7)
- RGB LED
- Raspberry Pi Camera Module V2
- Raspberry Pi 4 4GB
- Arduino Mega
- Arduino Uno
- Bq2920x (BMS)

The remote incorporates five distinct components in order to build the remote. The goal for the remote is to create a PCB that can utilize an A spec (for user remote control) and a B spec (for vehicle authentication).

1) User Interface

Figure 23 - Expected design of the remote the user will have for the MARGE system (A-spec).

The user interface will incorporate seven buttons linked to a different task or mode. Each of the modes will be defined below. For the hardware, the team has chosen to use push buttons with an LED as an indicator light to indicate the mode after a user button is pressed. Upon a successful signal sent between the RFMs, the light will turn off. The Arduino Nano on the remote will only wake up from a low-power "sleep" mode upon a user input.

2) Arduino Nano

In order for the remote control to be as robust as possible, the team has chosen to use an Arduino Nano. The Arduino will be the "brains" for the remote and will control timings, power management (battery balancing), and the user interface. Due to already purchasing an Arduino, the team has elected to do a through hole design in the PCB and use the previously purchased breakout board.

3) RFM95

Our team has decided to implement an RFM95 as a LoRa radio transceiver on the 915MHz band. Our team will utilize this part in order to establish long range communication between the remote, the rover, and the docking station. LoRa technology works by modifying frequency shift keying (FSK) by sweeping between the two frequencies and transmitting a signal based on a rising or falling edge. The remote will interface with the RFM95 by utilizing a three-wire SPI connection.

4) Battery Balancing System

The battery balancing system will utilize two NMOS transistors and two 10KΩ resistors in order to create the hardware for the active balancing system. The Arduino will be programmed to use its on- board ADC to read the different voltages of the battery and to see if they are out of 1% tolerance.

5) Bq2920x (BMS IC) , BQ25306RTER (Battery Charging IC) & USB C Receptacle

The remote will incorporate a battery charging IC, the BQ25306RTER, and will be connected to a power-only USB receptacle in the downstream format to provide the user with the ability to charge the battery of the remote. The BQ25306 supports 4.1-V to 17-V input voltage range so the USB will be powered with the standard 5 volt wire. The bq2920x will handle the battery balancing for the two batteries within the remote. The bq2920x is also able to do over voltage protection, and overcurrent protection along with automatic cell balancing.

6) Communication Protocol

Using the RFM95 transceiver as described in 3), the MARGE subsystem will have a remote-controlled sensing distance (RCSD) of ~1000 feet.

Navigation Hardware

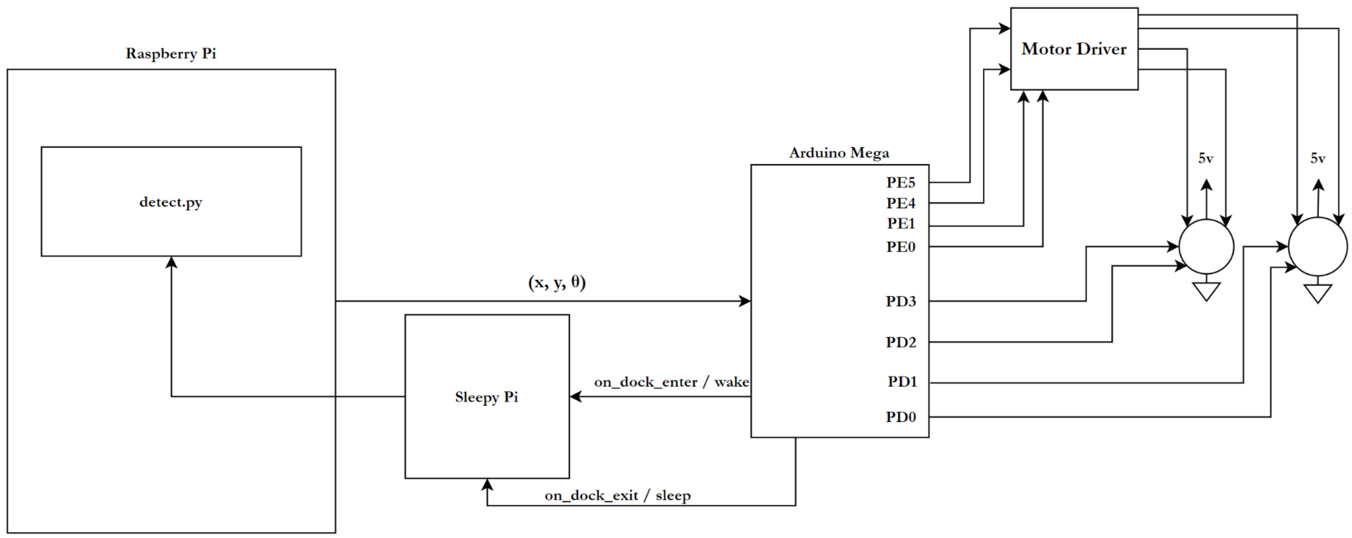


Figure 24 - Navigation top-level hardware

The navigation hardware features four PWM outputs and four interrupt inputs. The interrupt inputs will be used by the encoder library to deduce tick counts per change in time. The PWM outputs will be sent to the motor driver and are used to drive the motors. The Arduino Mega will feature logic outputs and inputs to manage the power on and off logic facilitated by the power regulation of the Sleepy Pi 2 to the Raspberry Pi 4 depending on the current state of the FSM. The logic output pin will facilitate turning the Raspberry Pi on in the "on_docking_enter" transition to the 'Docking' state and it will shut down the Raspberry Pi on the "on_docking_exit" transition upon completion of the docking protocol.

Navigation Software

The navigation sub-system shall be responsible for ensuring the Rover accurately gets to its desired pose: either the driveway or the docking station. Two odometry techniques are used for localization of the Rover: a computer vision based odometry technique and an encoder odometry technique. The odometry data is parsed with a localization class which outputs a localized pose of the Rover. This pose is used by a PID position controller which minimizes the error between desired and actual pose to achieve the desired position. The output of the PID controller is an angular velocity that must be further parsed into a suitable proportional value to output four PWM signals for both motors using Arduino output logic.

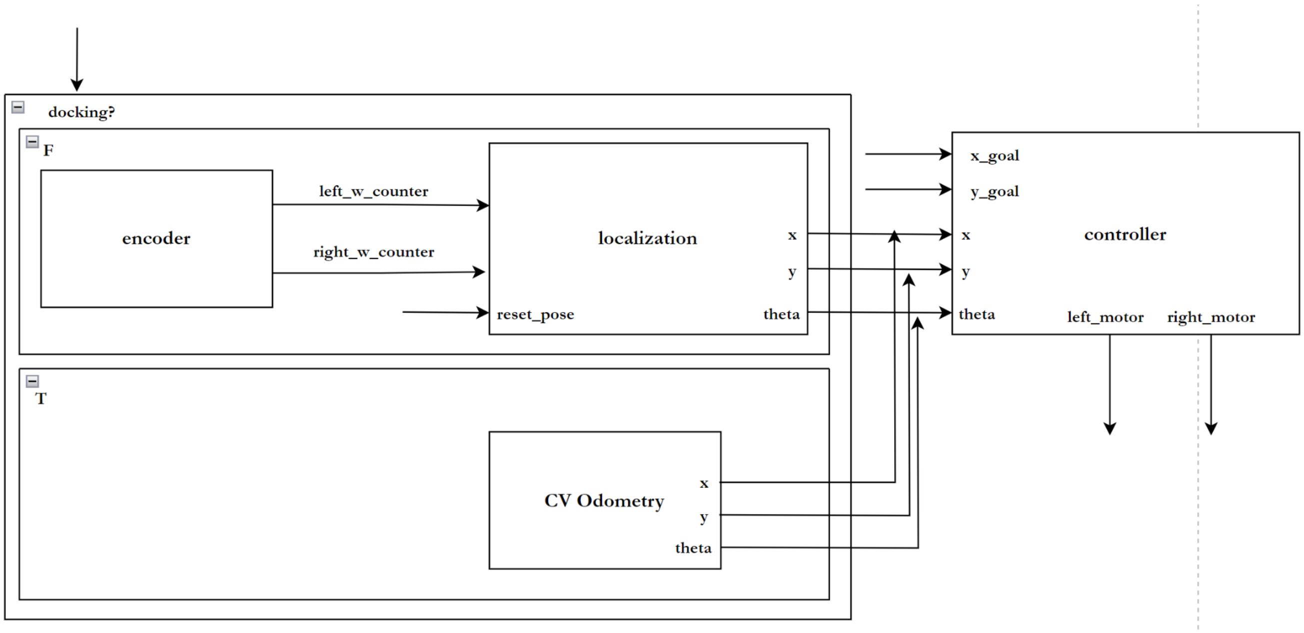


Figure 25 - Navigation top-level block diagram

Computer Vision Odometry

The computer vision odometry uses the openCV library. An ArUco marker is an artificially generated square marker featuring a broad black border and an internal binary matrix that defines its unique identifier. Before detection or pose estimation the Raspberry Pi Camera Module V2 must be calibrated properly. Following the calibration process, a camera matrix is obtained, which is a 3x3 matrix containing the focal distances and camera center coordinates (referred to as intrinsic parameters). Additionally, there are distortion coefficients, represented by a vector of 5 or more elements, capturing the distortions introduced by the camera. The tag will be detected and then pose estimation is performed. The pose estimation is implemented as follows: the position and orientation of the camera with respect to the marker are described by a 3D transformation, mapping from the marker's coordinate system to that of the camera. This transformation is defined by rotation and translation vectors. These transformations are facilitated by Lie algebra, specifically Rodrigues's rotation formula. The output of this formula is a rotation matrix given a rotation screw axis as input. The above functionality was implemented using a python script called 'detect.py'.

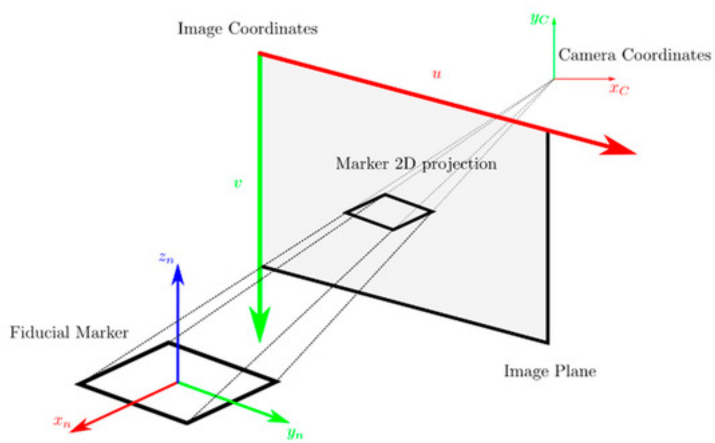


Figure 26 - Camera and Aruco frame representation w/ translation and rotation vectors.

<u>Camera Calibration</u>

In order for the openCV algorithms to work properly, it is necessary to ensure the camera is calibrated properly. This is a lengthy process and involves using a 2-D checkerboard and measuring the dimensions of one square in millimeters. Then an error minimization algorithm is computed to calibrate the camera using 112 different checkerboard images at different angles and distances from the camera. The error between what the camera sees and the actual dimensions are minimized and the output of the algorithm is a configuration of the camera that accurately estimated pose in world space in the form of a distortion matrix. A python script called 'calibrate.py' was created to perform the above functionality.

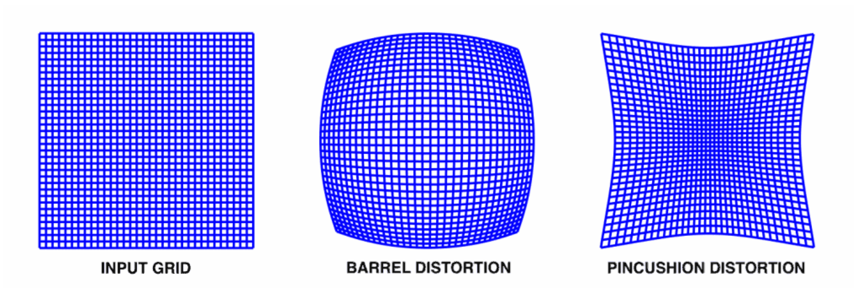


Figure 27a & 27b - Distortion examples

<u>Encoder Odometry</u>

The encoder odometry is a well established technique to localize a Rover. The output of the class 'encoder.cpp' are tick counts in one derivative time step. The 'localization.cpp' method will parse the tick counts into angular velocities in rpm, then convert those angular velocities to rad/s, and finally compute the linear and angular velocities of the Rover frame, that frame being halfway between the rear wheels. The position of the Rover is deduced from the velocities by simply integrating the linear and angular velocity equations. The derivation of the linear and angular velocity equations can be found below.

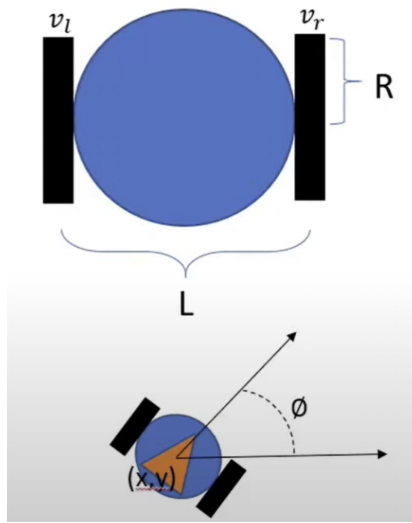


Figure 28 - Rover configuration parameters L (wheel spacing) and R (wheel radius)

Encoder Calculations

```
CONVERT TICK TO RPM

1 rpm - 40 ticks/min  <- 40 ticks per rev.

x rpm - tick count / min

x = (tick count / dt) * 1/40 [rpm] <- speed x in rpm

dt -> nano seconds    <- dt in ns
1 min = 60E9 ns

X = (tick count / dt) * (60E9 / 40)

X = tick count / dt * 1.5*E9 [rpm] <- x in rpm w/ dt in ns

SPEED FORMULAS

v(rad/s) = v(rpm) * pi/30   <- convert rpm to rad/s

Vrover = R/2 * (Vr + Vl) [m/s]  <- linear vel
Wrover = R/L * (Vr - Vl) [rad/s]  <- angular vel

source: Georgetown robotics
```

Figure 29- Encoder counts to speed calculations

Docking

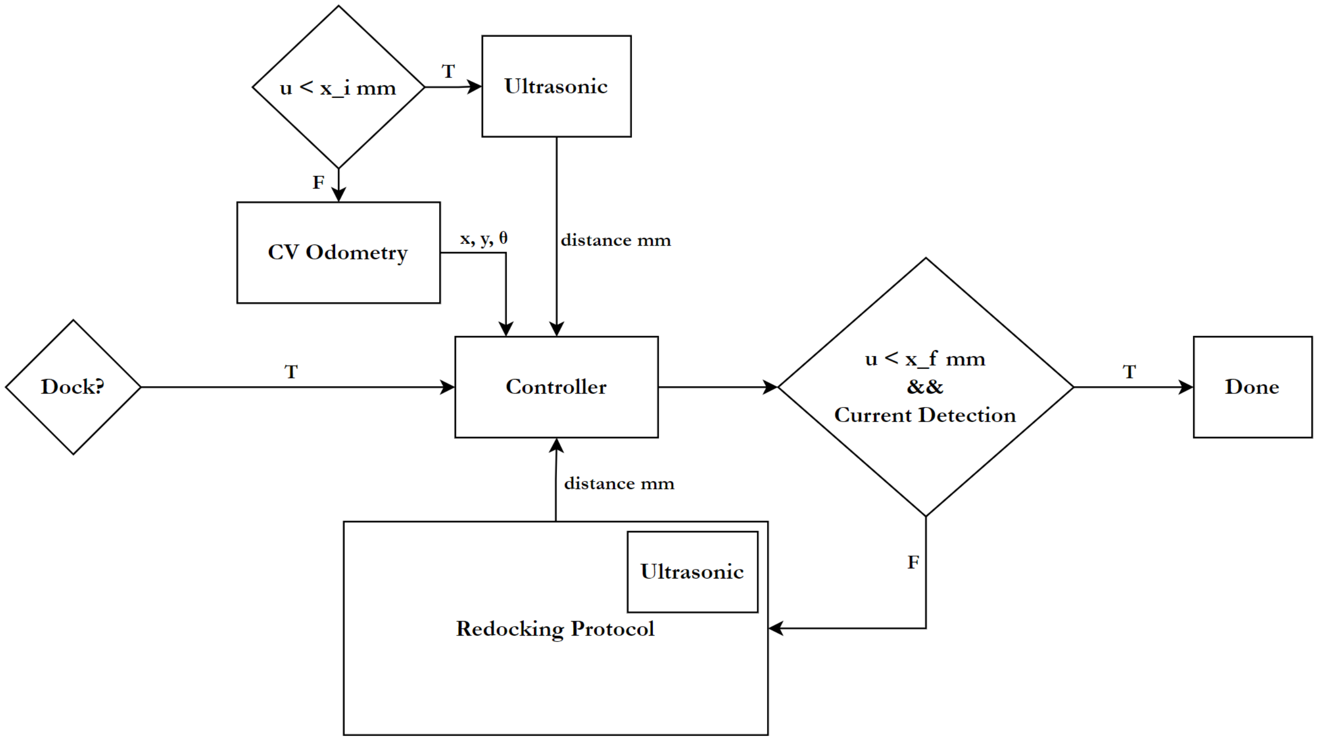


Figure 30 - Docking protocol

u: ultrasonic distance measurement

x_i: initial ultrasonic distance threshold to transfer control from CV to PID angular velocity control and ultrasonic linear velocity control

x_f: final ultrasonic distance threshold to initiate current sensing

The docking protocol in firmware features the CV odometry submodule as well as an ultrasonic submodule. The CV odometry will be outputting a localization for the controller, and the controller will attempt to reduce the error between the tag's pose and the localization of the Rover. Once the Rover reaches the top of the ramp and is within an x_i distance constraint, a PID controller will regulate the angular velocity of the Rover to ensure it drives straight and the linear velocity of the Rover will be proportional to the ultrasonic distance measurement. Once the Rover is within a final distance constraint x_f from the docking station wall, a current sensor input will wait to be read on a timeout that will terminate the procedure. If the final distance constraint is met but the Rover does not detect current within the timeout constraint, a realignment procedure will be conducted in which the Rover moves back Y millimeters and reattempts the docking procedure using the latter control, namely ultrasonic and PID angular velocity regulation to ensure straight motion.

Remote Sensing: NRF vs. RFM

During our testing and development of our initial proposal, the team discovered that the previous solution, the NRF2401+ PA + LNA would not be suitable for the project as the team could not produce a long range stable communication using them. The team decided to pivot for our final design to the RFM95 after testing the distance and ease of use. A team member was able to walk to Wawa from Acopian (roughly 0.3 of a mile away) and maintain connection.

Remote Control

One crucial component of the MARGE device is the remote controller. This device is a requirement of the system, as it allows for overrides that let unauthorized vehicles pass. It also has the functionality to change the system's mode and allows for cosmetic changes to the rover. The remote allows the user to control the MARGE device's functionality from a distance, which is a convenient feature for any operator.

Arduino Remote Control Code Library Note: The Arduino Nano's code after testing will work with the Radiohead Arduino library in order to establish communication between the RFMs.

In order to properly communicate with the Rover, the remote control needs to send a variety of signals to the device. These signals will be sent with the push of a button. **These buttons have LEDs embedded in them** that can be controlled externally via microcontroller, which allows for proper debugging of the remote. These buttons will send the following signals:

- no_push: The device is sent into the "normally open" operating mode.
- nc_push: The device is sent into the "normally closed" operating mode.
- sch_push: The device is sent into the "scheduled" operating mode.
- dr_push: The rover will be sent into the driveway as an override.
- ds_push: The rover will be sent to the docking station as an override.
- ct_push: The cone on the rover will be sent up or down depending on the previous state.
- lt_toggle: The LED lights on the rover will turn on or off depending on the previous state.

Two signals will be sent to the remote to control behavior. The "reset" signal is asserted when the remote is powered on. It will ensure that every output is off and all signals are set to "false" so that the device can behave properly on startup. The "sig_rec" signal will be asserted true when the transceiver on the rover acknowledges that a signal from the remote was received. This signal will trigger the IDLE state that is mentioned below.

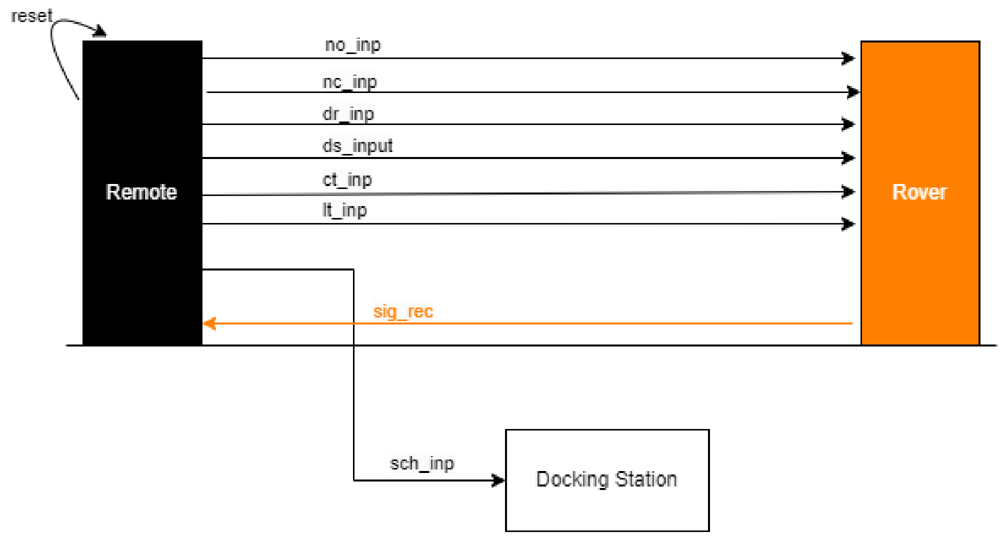


Figure 31 - Signal diagram for the remote control

The behavior of the remote's FSM consists of nine unique states. The device defaults to an IDLE state where all LEDs are off. When a button is pressed, the remote transitions to a state corresponding to the button's designated behavior. In these states, the LED in the corresponding button will turn on. For instance, in the NORMALLY_OPEN state, the button in the green "Normally Open" LED button will be illuminated. The device will then transition to the DELAY state where nothing happens until the rover receives the signal, processes it, and sends a "sig_rec" signal to the remote. At this point, the device transitions back to the IDLE state, and the illuminated LED will turn off. The code will be designed so that the remote can not send more than one signal at a time to remove potential errors.

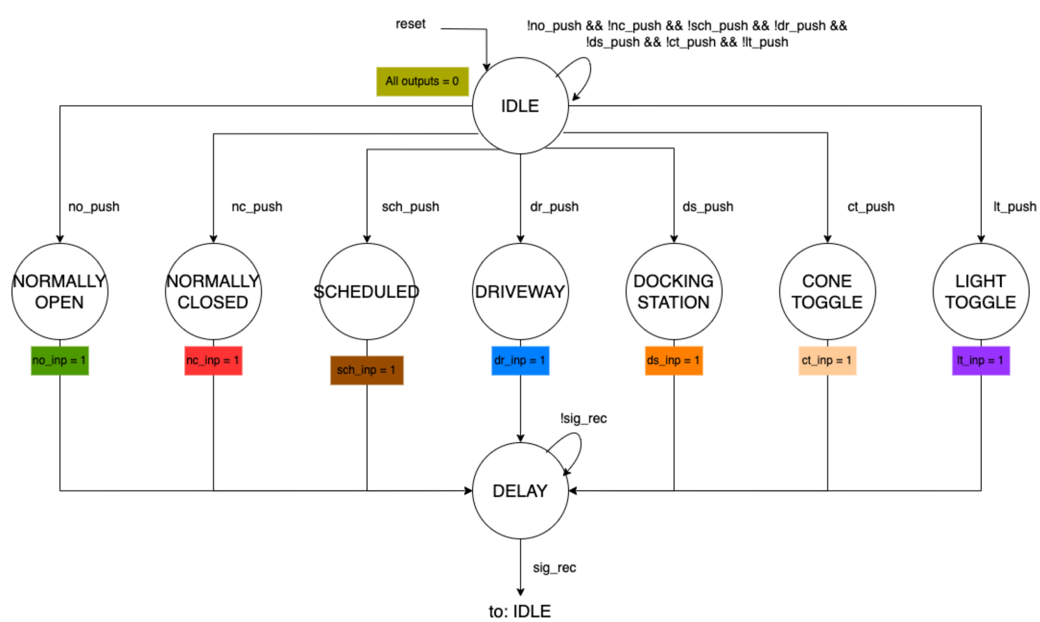


Figure 32 - State diagram for the remote control

FSM of Rover / Explaining Behavior

In order to control the behavior of the rover, a finite state machine (FSM) will be implemented on an Arduino Mega that takes input signals from the remote, the beacon in the authorized vehicle, and the docking station. The rover will transition between states depending on these inputs and the mode that is set by the remote or the GUI. Below is the list of variables that the FSM contains:

Inputs:

- Operating behavior:
  - car_detect - Signal that is sent when a car is detected from the forward-facing radar.
  - auth_car_detect - Signal from the beacon that informs the rover that a recognized vehicle has approached
  - leave_car_detect - Signal from the back-facing radar that detects if a vehicle is leaving the driveway.
  - in_driveway - Output from the navigation network that informs the rover that it is in the center of the driveway.
  - in_ds - Output from the navigation network that informs the rover that it is in the docking station.
  - reset - Signal sent when the device is powered on. It clears all outputs and ensures that the rover is in the IDLE state.
- Remote control inputs:
  - no_inp - State that triggers the "Normally Open" mode on the rover. This signal also can be sent from the GUI on the docking station.
  - nc_inp - State that triggers the "Normally Closed" mode on the rover. This signal can also be sent from the GUI on the docking station.
  - dr_inp - Input from the remote to override the mode settings and send the rover to the center of the driveway.
  - ds_inp - Input from the remote to override the mode settings and send the rover to the docking station.
  - lt_inp - Signal sent from the remote that turns the lights on the rover on or off. This is an input that affects the cosmetics of the rover and does not contribute to the state behavior.
  - ct_inp - Signal sent from the remote that allows the cone mechanism on the rover to move up or down. This is an input that affects the cosmetics of the rover and does not contribute to the state behavior.
- Charging behavior:
  - charge- Variable that comes from the battery detection sensors, detects the battery's percentage
  - ds_low_charge - Input that sends the rover back to the docking station (if not already) and into the IDLE mode if the docking station has low battery and can not charge the rover.

- Scheduled mode:
  - sched_nc - Signal that is asserted high when the device is in the "scheduled" mode and the time corresponds to "Normally Closed."
  - sched_no - Signal that is asserted high when the device is in the "scheduled" mode and the time corresponds to "Normally Open."

Internal Variables:

- threshold- Value that the battery's charge should not go past. If it is lower than this number, the rover must go to the IDLE state to charge. This threshold is currently set at 15% but may change as we develop the design.

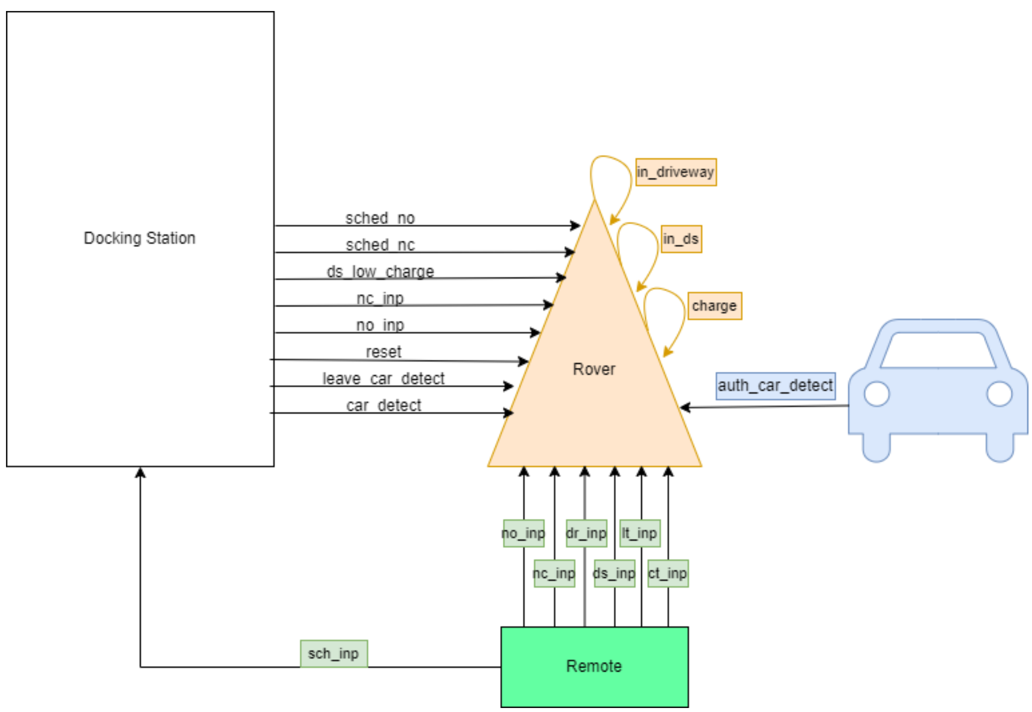


Figure 33- Signal diagram for the rover and the docking station

There are 11 states that are in the rover's FSM. The connections of the states are displayed in the state transition diagram below. The following list describes the detail of each state:

1. IDLE - This is the default state for the rover to charge. While in this state, the rover will remain in the charge bay within the docking station. No sensing will be done by the docking station and the rover will enter its low-power mode. The rover will enter the NO_IDLE state if an input signal "no_inp" is sent from either the remote or the GUI or if the scheduled "Normally Open" mode is enabled, assuming that the charge is greater than the defined threshold. The rover will enter the NC_MOVE state if an input signal "nc_inp" is sent from either the remote or the GUI or if the scheduled "Normally Closed" mode is enabled, assuming that the charge is greater than the defined threshold.
2. NO_IDLE - In this state, the rover remains in the docking station but the docking station's radar sensors are turned on. It will also periodically check the battery of the rover to ensure it is above the

threshold. If a car is detected and it is not authorized to enter or the “dr_inp” signal on the remote is pressed, the rover will transition to the NO_TO_DRIVEWAY state and begin the process of blocking the incoming vehicle. If the device transitions into the scheduled “Normally Closed” mode, the rover will transition to the NC_MOVE state. If the charge of the battery is less than the threshold value, the rover will transition back to the IDLE state to charge. If none of the above are true, the rover will stay in the NO_IDLE state.

3. NO_TO_DRIVEWAY - This state tracks the navigation of the rover to the center of the driveway. It will remain in this state until the navigation code confirms that the rover is in the center of the driveway. Upon completion, the rover will transition to the NO_DRIVEWAY state.
4. NO_DRIVEWAY - In this state, the rover sits in the center of the driveway until an unauthorized vehicle is no longer attempting entry. While in this state, the rover is running checks on its battery life to ensure that it is over the threshold. If an authorized vehicle is detected, the “ds_inp” signal is sent from the remote, or the back-facing radar recognizes a car that wants to exit the driveway, the rover will transition to the NO_TO_STATION mode to traverse back to the docking station. If the device enters its scheduled “Normally Closed” mode or if the “nc_inp” signal is sent from the remote or the GUI, the rover will transition to the NC_IDLE mode and remain in the center of the driveway. If the charge is lower than the threshold, the rover will transition to the DRIVEWAY_TO_CHARGE state to begin its navigation to the charger. If none of the above is true, the rover will remain in the center of the driveway and wait for a signal.
5. NO_TO_STATION - This state tracks the navigation of the rover to the docking station. It will remain in this state until the navigation code confirms that the rover is in the docking station. Upon completion, the rover will transition to the NO_IDLE state.
6. NC_MOVE - This state tracks the navigation of the rover to the center of the driveway. It will remain in this state until the navigation code confirms that the rover is in the center of the driveway. Upon completion, the rover will transition into the NC_IDLE state.
7. NC_IDLE - In this state, the rover will block the driveway whilst the docking station uses the radar sensors to look for vehicles. Any vehicle approaching the rover from the road will be forced to turn around as the cone stays in the center of the driveway so long as it is not authorized with the beacon. If a vehicle approaches the rover and sends a signal from the beacon, the rover receives a “ds_inp” signal from the remote, or a car is detected to leave the driveway from the other side, the rover will enter the NC_TO_STATION state to begin its journey back to the docking station. If the rover is signaled to enter the scheduled “Normally Open” mode or the “no_inp” signal is sent from the remote or the GUI, the rover will transition to the NO_TO_STATION state and return to the docking station. If the charge is lower than the specified threshold, the rover will transition to the DRIVEWAY_TO_CHARGE state to begin its navigation to the charger. Otherwise, the rover will remain in the NC_IDLE mode and block off the driveway.
8. NC_TO_STATION - This state tracks the navigation of the rover to the docking station. It will remain in this state until the navigation code confirms that the rover is in the docking station. Upon completion, the rover will transition to the NC_STATION state.

9. NC_STATION - In this state, the rover sits in the docking station until the authorized or exiting vehicle has passed. If no car is detected or the “dr_input” is selected on the remote, the device will transition to the NC_TO_DRIVEWAY state. If the rover receives the signal to transition to the scheduled “Normally Open” mode or the rover receives a “no_inp” signal from the remote or the GUI, the rover will transition to the NO_IDLE state. If the charge is lower than the specified threshold, the rover will transition to the IDLE mode to charge. If none of the above conditions are met, the rover will stay at the docking station until its next move is communicated.
10. NC_TO_DRIVEWAY - This state tracks the navigation of the rover to the center of the driveway. It will remain in this state until the navigation code confirms that the rover is in the docking station. Upon completion, the rover will transition into the NC_IDLE state.
11. DRIVEWAY_TO_CHARGE - In this state, the rover returns to the docking station with the sole intent to charge. It will remain in this state until the navigation code confirms that the rover is in the docking station. Upon completion, the rover will transition into the IDLE state.

The “sch_inp” input from the remote will toggle the device into scheduled mode. A program running on the docking station will emulate a real-time clock where the user can set times for the scheduled mode on the GUI. When the device enters the scheduled mode, it will toggle the “sched_nc” and “sched_no” inputs to the FSM pending on the current time. When the user sends the “sch_inp” input from the remote to the docking station, it will be parsed within this code and the proper schedule signal will be asserted to the FSM. The algorithm for determining the time will be created using an arduino library that will be determined by the firmware team at a later date.

Though not crucial to the device’s behavior, the user also has the option to toggle the lights and the cone on the rover with the remote control. In all non-moving states except for IDLE, a function will be created that will toggle the lights or the cone depending on the signal it receives.

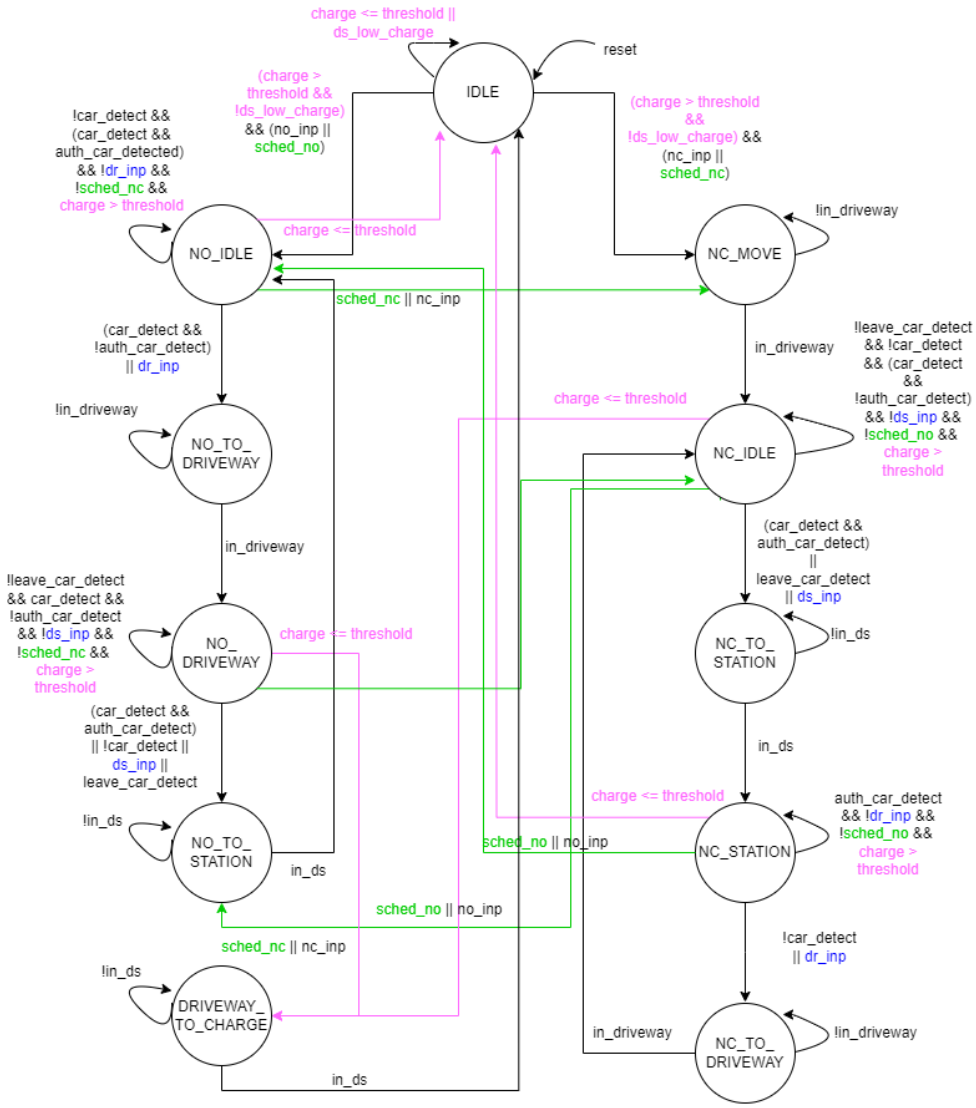


Figure 34 - State transition diagram for the rover

USER CONTROL

Control from Docking Station

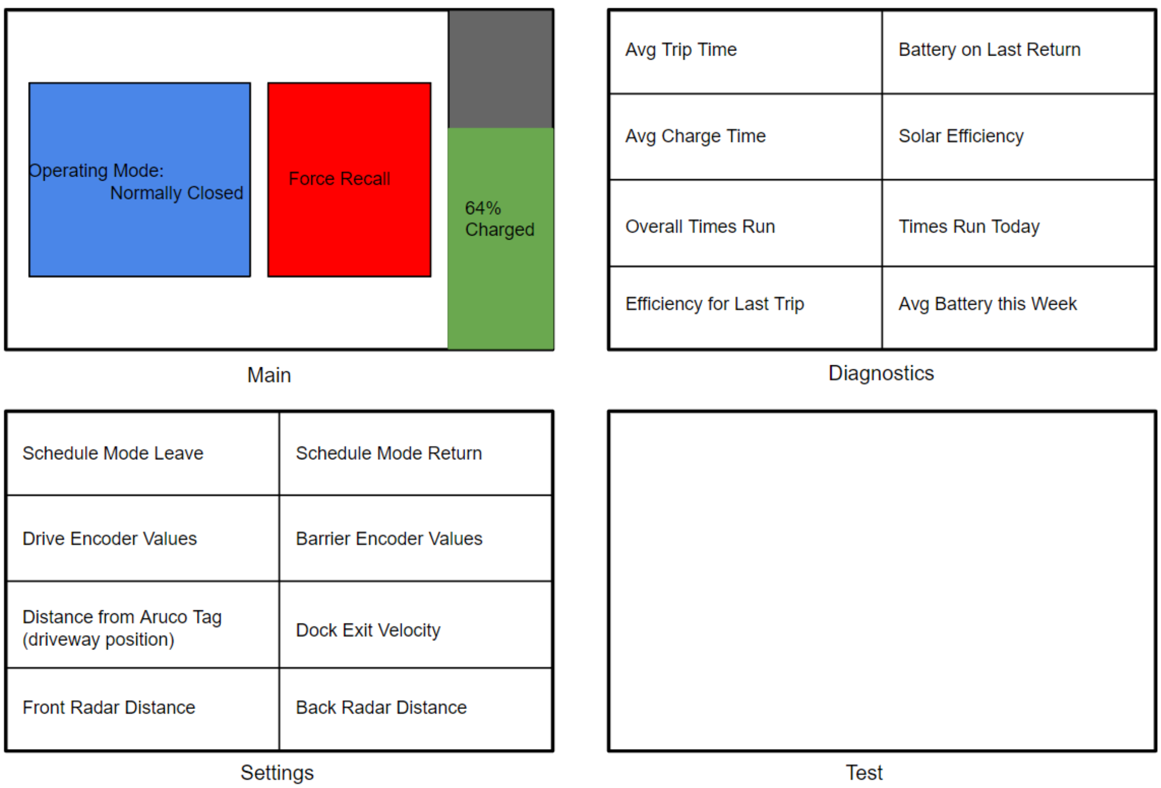


Figure 35 - LCD GUI Design

The docking station will have a mounted, non-touchscreen, LCD sunlight visible display. On the screen, a GUI will be displayed to allow a user to visualize and interact with data pertinent to the operation of the rover. Operation of the display will be possible through 4 buttons and a knob located on the docking station just below it. This will happen almost exclusively through the provided knob and fourth button. This button is used as a selector for users to interact with the functions and settings presented to them. Each piece of data on screen will have the ability to be interacted with. First, the user will turn the knob to highlight the area on screen they would like to interact with (by turning the knob, the user cycles through the options presented). Next the user presses select, opening a submenu unique to each screen. They will then use the knob to either alter values or select an option. Lastly, a user would press the select button again, finalizing their change and exiting the submenu.

Overall, the GUI will be divided into 4 individual tabs:

- **Main**: Whenever the display turns on, users will be presented with this screen. It shows the current operating mode of the rover, alongside a reading of the rover's current battery life and a button to manually recall the rover. There will be a mode selection submenu, as well as a confirm recall submenu, resulting from selecting their respective areas. This screen will be linked to **button one**.
- **Diagnostics**: Here, data will be shown on the day to day function of the rover. Each specific field can be selected to view a graph of values over time to demonstrate inconsistencies that may exist. All information mentioned on the screen is subject to be changed (especially when determining which unit of time to utilize). Upon selection of a field, users would be presented with a graph of the data over some stretch of time. They can use the knob to cycle through points on the graph even though they cannot alter it. This screen will be linked to **button two**.
- **Settings**: On this screen, there will be variable data that the user can interact with and tune to ensure the rover is functioning in a way that meets their specifications. When selected, the submenu that opens presents you with the current value of that variable. Users can twist the knob to select a relative value (within bounds set by Team Alpha) to fine tune it to their desires. This screen will be linked to **button three**.
- **Test**: Test mode is intended for developmental purposes, and will put the robot into a dummy mode where it is easily manipulatable. This screen will not be linked to a button, but will require some sequence to be pressed to activate - this mode is not intended for use by the end user. At present, it is difficult to know what might need to belong on the test screen, so it is acknowledged but not designed.

INTEGRATION & VALIDATION

Spring Workplan

This spring work plan will serve as the document that guides, from a top-level perspective, our integration efforts in the spring. Responsibilities in this plan will be specific and include task 'owners' and task completion dates. The owner of a task is the person(s) responsible for completing the task by the assigned date. Without significant prior advance warning for why a task deadline needs to be extended, and good reasoning behind the need for extension, the owner(s) will be held directly accountable for missing the deadline and appropriate action taken. For tasks with a date listed as 'N/a', or non-applicable, it is the expectation that this task is worked towards on a continual basis and its upkeep and care is the responsibility of the task owner. This document is not exhaustive, the tasks listed here will be general and focused from top-level. Additionally, not mentioned in this workplan are the non-engineering related role responsibilities of particular members. Each member on the team serving a non-engineering role is expected to continue their responsibilities in addition to their engineering development work. For engineering tasks, it is the responsibility of the task owner to determine the appropriate steps required to complete the task to the required quality standards. Any member of the team reserves the right to challenge the quality of a completed task, in which case the team must review the quality of work completed from which appropriate action can be taken. Additionally, the team as a whole and the team leader reserve the right to change the responsibilities of tasks and their dates of any owner(s), on the basis that there is valid justification and significant advance notice.

This document, in addition to the listed tasks, implies good faith and the understanding that every team member is equally responsible for the successful completion of this project. Each team member has mutual responsibility to contribute possibly beyond their listed responsibilities if deemed capable and if deemed necessary for the successful completion of the project. This document will take effect upon unanimous approval by all team members, and will be revised accordingly until such an agreement can take place.

Mechanical Design Team: (Devin Arnold & Samuel Ehgartner)

- Determine the hardware connector required for external interfacing of the docking station and rover, including all necessary wiring attachments such as crimp heads and add-ons for wire labeling and organization.
    - Owner: Devin
    - Deadline: End of Fall Semester
- Validate the Rover locomotion platform.
    - Owner: Samuel
    - Deadline: End of Fall Semester
- Validate the Docking station mechanical guiding system.
    - Owner: Samuel
    - Deadline: End of Fall Semester
- Validate adverse weather performance of the Rover and Docking station.
    - Owner: Sam and Devin
    - Deadline: March 1st, 2024
- Complete the Rover design and produce required part drawings for machine shop fabrication.
    - Owner: Devin
    - Deadline: End of Fall Semester
- Complete the Docking Station design and produce required part drawings for machine shop fabrication.
    - Owner: Sam
    - Deadline: End of Fall Semester
- Perform an ambient temperature test for data collection of Rover and Docking station internal temperatures as compared to external ambient temperature.
    - Owner: Devin
    - Deadline: April 12th, 2024
- Design and fabricate the remote control for both specifications.

- Owner: Samuel
- Deadline: February 16th, 2024 (Dependent on Firmware Team)
- Assist the Electrical Design Team in the physical wiring setup of all electrical and signal systems of both the rover and docking station.
    - Owner(s): Mechanical Design Team
    - Deadline: N/a

Electrical Design Team: (Corin Nosow, Drew Borek & Jack Horowitz)

- Assist the Mechanical Design Team in the physical wiring setup of all electrical and signal systems of both the rover and docking station.
    - Owner(s): Electrical Design Team
    - Deadline: N/a
- Setup and qualify commercial solar power generation and distribution components.
    - Owner: Jack
    - Deadline: Within 2 weeks after arrival, but no later than March 1st, 2024.
- Perform hardware qualification of the Aft Radar within reasonable time after its arrival.
    - Owner: Jack
    - Deadline: Within 1 week after arrival, but no later than February 12th, 2024.
- Complete the schematic and layout of the rover power distribution board.
    - Owner: Jack
    - Deadline: End of fall semester, out for manufacture
- Complete the schematic and layout of the docking station power distribution board.
    - Owner: Jack
    - Deadline: End of fall semester, out for manufacture
- Perform a thermal performance test of both the rover and docking station power distribution boards.
    - Owner: Jack
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Perform a functionality test of both the rover and docking station power distribution boards.
    - Owner: Jack
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Complete Team Alpha's portion of the drivetrain motor controller schematic and layout.
    - Owner: Jack
    - Deadline: End of fall semester, out for manufacture granted Team Beta can meet the deadline.
- Perform a thermal and functionality performance test of the drivetrain motor controller, producing results insightful to PWM frequency design decisions and other design factors.
    - Owner: Jack
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Complete the schematic and layout of the barrier actuation motor controller.
    - Owner: Drew

    - Deadline: End of fall semester, out for manufacture:
- Perform a thermal performance test for the barrier actuation motor controller.
    - Owner: Drew
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Perform a functionality test for the barrier actuation motor controller.
    - Owner: Drew
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Complete design and layout of RF communications board, designed around LoRa SX1276.
    - Owner: Drew
    - Deadline: End of fall semester, out for manufacture:
- Perform a functionality test of the RF communications board.
    - Owner: Drew
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Perform a functionality and quality assurance test for all system motors, determining a current vs. torque curve over the full range of torque/current (idle to stall).
    - Owner: Drew
    - Deadline: Within 3 weeks of the beginning of the spring semester.
- Assist in the development of the LCD hardware setup, and firmware development of, a fully designed and user-conscious GUI.
    - Owner: Drew
    - Deadline: N/a
- Complete schematic and layout of hall effect current sensor board.
    - Owner: Corin
    - Deadline: End of Fall Semester
- Perform a functionality, and thermal performance if necessary, test of the hall effect current sensor board.
    - Owner: Corin
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.
- Perform a functionality test of the ultrasonic sensor as a part of the validation of the docking procedure.
    - Owner: Corin
    - Deadline: March 8th, 2024
- Complete schematic and layout of LCD and button signal distribution board.
    - Owner: Corin
    - Deadline: End of Fall Semester
- Perform a functionality test of the LCD and button interface as a part of the validation of the signal distribution board.
    - Owner: Corin
    - Deadline: Within 2 weeks of the arrival of the board, but no later than March 1st, 2024.

Embedded Design Team: (Michala Dennis, Austin Pelayo & Alexander Labell)

- Perform a full navigation test utilizing an Aruco tag on a physical locomotive platform, or a simulation equivalent that evaluates performance in both typical and low light conditions.
    - Owner: Alexander
    - Deadline: End of semester
- Complete code for and qualify the sensor control and data acquisition firmware, including calculating docking station and rover battery state of charge (SOC).
    - Owner: Austin Pelayo
    - Deadline: Code Layout before Winter Break
- Complete code for and qualify the RF transceiver control firmware.
    - Owner: Austin Pelayo
    - Deadline: Code Completed before Winter Break
- Complete code for and qualify GUI control firmware.
    - Owner: Michala
    - Deadline: March 21st, 2024
- Complete code for and qualify all level FSM's at a simulation level.
    - Owner: Michala
    - Deadline: February 28th, 2024
- Complete code for and qualify the barrier actuation system firmware.
    - Owner: Austin Pelayo
    - Deadline: Laid out before Winter Break
- Design and complete a full testing mode that includes but is not limited to the following capabilities:
    - Simulate any system hardware input to see its applicable system output.
    - Simulate any system software input to see its applicable system output.
    - Perform a system-level diagnostic test to evaluate the operating state of system sensors, battery state of charge, and any other factors determined pertinent.
    - Owner: Embedded Design Team
    - Deadline: End of April 2024
- Complete code for and qualify a fault management system that can detect, and if applicable diagnose, hardware faults in the system. This is expected for system sensors and higher level modules, not IC level failures.
    - Owner: Alex
    - Deadline: Mid Feb
- Design a power management system that can optimize power consumption, specific to each operating mode, and protect the system batteries from exhaustion and subsequent system failure.
    - Owner: Embedded Design Team
    - Deadline: Spring Break 2024
- Perform unit testing in all above programs as applicable.
    - Owner: Embedded Design Team

    - Deadline: End of Spring 2024 semester
- Hardware task: Complete the schematic and layout of the remote controller hardware, for both the general remote control and the vehicle authentication specifications.
    - Owner: Austin
    - Deadline: 17/11/2024
- Hardware task: Perform a functionality test to validate the remote controller hardware, for both the general remote control and the vehicle authentication specifications.
    - Owner: Austin
    - Deadline: TBD - second semester

System-Level Validation Test Plan

Test 1: Allowing an authorized vehicle to pass in the Normally Open mode

- **Pass/fail criteria:** This test will pass if the rover stays in the docking station and lets an authorized vehicle pass while the MARGE system is operating in the Normally Open mode.

Test 2: Allowing an authorized vehicle to pass in the Normally Closed mode

- **Pass/fail criteria:** This test will pass if the rover moves from the driveway to the docking station and lets an authorized vehicle pass while the MARGE system is operating in the Normally Closed mode.

Test 3: Blocking an unauthorized vehicle to pass in the Normally Open mode

- **Pass/fail criteria:** This test will pass if the rover moves from the docking station to the center of the driveway and stays there to prevent an unauthorized vehicle from passing in the Normally Open mode..

Test 4: Blocking an unauthorized vehicle to pass in the Normally Closed mode

- **Pass/fail criteria:** This test will pass if the rover stays in the center of the driveway to prevent an unauthorized vehicle from passing in the Normally Closed mode.

Test 5: Rover enters its Idle (charging) mode from the driveway

- **Pass/fail criteria:** This test will pass if the rover navigates from the driveway to the charging pad in the docking station to charge.

Test 6: Navigating past an obstacle while returning to the docking station

- **Pass/fail criteria:** This test will pass if the rover is able to navigate around an obstacle in its path and return to the docking station.

Test 7: Remote switches the modes of the rover

- **Pass/fail criteria:** This test will pass if the user is able to switch between the Normally Open, Normally Closed, and Scheduled navigation modes on the remote.

Test 8: Charging the remote

- **Pass/fail criteria:** This test will pass if the remote is able to charge and maintain life with the USB-C charging port.

Test 9: Testing the range of the RCSD

- **Pass/fail criteria:** This test will pass if the remote is able to communicate with the rover and docking station from a minimum of 100 feet, the set remote-controlled sensing distance (RCSD).

Test 10: Validating the charging capabilities of the solar panels

- **Pass/fail criteria:** This test will pass if the solar panel is able to charge the docking station.

Test 11: MARGE system operates in inclement weather

- **Pass/fail criteria:** This test will pass if the MARGE subsystem can continue its expected operations in inclement weather (rain, snow, hail, fog, etc.)

Test 12: MARGE device success in various terrain conditions

- **Pass/fail criteria:** This test will pass if the MARGE subsystem can continue its expected operations in a variety of terrains (pavement, dirt, gravel, etc.)

Test 13: MARGE device success in daylight

- **Pass/fail criteria:** This test will pass if the MARGE subsystem can continue its expected operations in daylight.

Test 14: MARGE device success in night time

- **Pass/fail criteria:** This test will pass if the MARGE subsystem can continue its expected operations in the dark (at night).

Test 15: MARGE device longevity

- **Pass/fail criteria:** This test will pass if the MARGE subsystem can continue to operate as expected after a long period of time (a week without user interruption/stoppage)

Test 16: Scheduled mode validity checks

- **Pass/fail criteria:** This test will pass if the rover is able to switch between Normally Open and Normally closed modes while in the scheduled mode.

Test 17: User interface helpful for switching modes

- **Pass/fail criteria:** This test will pass if the user is able to interact with the GUI in a straightforward manner.

Test 18: Works for different sized vehicles

- **Pass/fail criteria:** This test will pass if the MARGE system works for vehicles such as motorcycles, compact cars, and golf carts.

Test 19: Moving over elevated objects

- **Pass/fail criteria:** This test will pass if the rover is able to move over curbs or other small objects and still operate normally.

APPENDIX

Appendix A: Relevant Codes & Standards

1. US Title 29 CFR Part 1910 Occupational Safety and Health Standards, Sub Part O, Machinery and Machine Guarding.
2. 2002/95/EC RoHS Directive for Hazardous Materials
3. 2002/96/EC WEEE Directive for Ecological Disposal
4. NFPA79/UL508 Standards for Power Wiring
5. EIA/TIA 568 Standards for Network Wiring
6. ANSI Z136.1-2000: *American National Standard for Safe Use of Lasers*
7. *[1]"Arduino Mega 2560 Rev3," Arduino Online Shop.* *https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us*