LAFAYETTE COLLEGE
ECE 491 - SENIOR DESIGN I
TEAM I - PRELIMINARY DESIGN PROPOSAL

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

TABLE OF CONTENTS

List of Figures 2
Introduction 3
Mechanical Design 3
Electrical Design 6
Embedded Control 18
User Control 26
Appendix 30

LIST OF FIGURES

Figure 1 - Docking Station Mechanical Design
Figure 2 - Rover Mechanical Design
Figures 3A, 3B, 3C & 3D - Barrier Mechanical Design
Figure 4 - Rover Top-Level Preliminary Design
Figure 5 - Docking Station Top-Level Preliminary Design
Figures 6A, 6B & 6C - Motor Current Consumption Analysis
Figure 7 - Motor Interface
Figure 8 - Drivetrain Top Level Design
Figure 9 - Pololu Motor Controller Diagram
Figure 10 - Aruco Fiducial Tag
Figure 11 - Computer Vision Localization with Aruco Tags
Figure 12 - Arduino Mega 2560 Rev3 Pinout
Figure 13 - Navigation Top Level Interface
Figure 14 - Docking Protocol
Figure 15 - Rover Top Level Finite State Machine
Figure 16 - Docking Station Top Level Finite State Machine
Figure 17 - Mode A: “Normally Closed”
Figure 18 - Mode B: “Normally Open”
Figure 19 - Mode C: “Scheduled”
Figure 20 - GUI with Button Layout
Figure 21 - Operator Remote-Override Controller Top Level Design

INTRODUCTION

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.

The Docking Station holds a primary battery that is charged by a solar panel positioned adjacent to the Docking Station. Additionally, the Rover contains its own battery to enable wireless operation from the Docking Station. The Rover charges itself when inside the Docking Station. The Rover will be a wheeled device that steers itself from the Docking Station to a position along the driveway, and vice versa it will navigate itself back to the Docking Station. This will be executed without the assistance of the operator.

MECHANICAL DESIGN

The Docking Station

We propose the Docking Station will be constructed using a metal shelf frame with machined outer panels for environmental protection and partitions within the structure. Based upon the sizes of the components being stored within, the Station will be approximately 30” long, 20” wide, and 20” tall. The Docking Station will be divided into three compartments. The lower level will be split into the garage for the MARGE Rover to park and recharge and the battery storage. There will be a small ramp at the base of the entrance as well as wheel channels to allow the Rover to more easily enter the garage. The opening for the Rover to enter will be 14” tall. At the rear of the garage, there will be spring-loaded male contact points for the rover to interface for charging. The Rover will have complimentary charge plates for the male contacts to touch. Behind the docking station’s male ports will be the battery housing. The power storage within the Station will be a 12 volt, 50 amp-hour battery. In addition, the battery housing will contain the necessary power management devices such as step down converters, charge controller, and required wiring. The battery will be secured to the frame of the docking station using brackets. The upper level of the Docking Station will consist mostly of the computational devices. This includes microcontrollers, necessary sensors, wiring, and other computation based electronics. On the exterior of the upper level, there will be a backlight display to allow the user to configure and evaluate the system’s functionality. The Docking Station’s power will be supplied by a 100W monocrystalline panel. The panel will be fixed at a permanent 34 degree tilt, but will be attached on a 360 degree swivel mount. This allows for the user to rotate the panel to a more optimal direction based upon their driveway's direction.

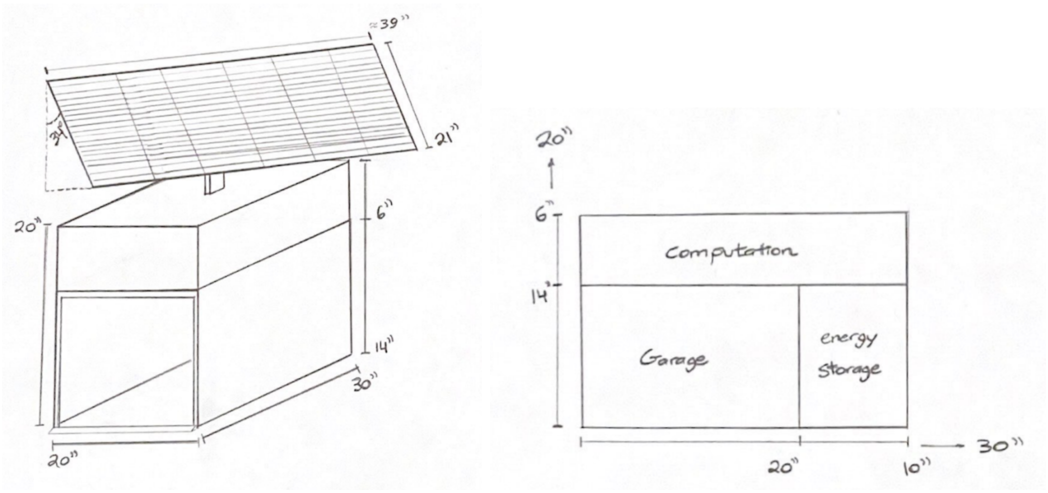


Figure 1 - Docking Station Mechanical Design

<u>The MARGE Rover</u>

The Rover will be responsible for traveling between the Docking Station and the midpoint of the driveway. The Rover body will have a rectangular prism shape of about 18" long, 15" wide, and 8" tall (without wheels) and make use of two primary wheels. The side of the chassis that faces the Docking Station will be chamfered and rounded to allow for an easier alignment with the Docking Station. The two primary wheels will use six inch diameter pneumatic tires and are each driven by the GR-EP-45ENC 45mm 12V 20W Planetary Gearmotor with a 50.9:1 gear ratio. This particular motor was chosen based on the torque and speed abilities, as well as a set design constraint of the Rover being about 15 pounds. The main wheels will be supplemented with two swivel wheels located on the front and back of the vehicle. The swivel wheels will not be driven by any motor. Steering of the Rover will be accomplished by applying differing drive commands to the motors to create rotational motion. This gives the Rover the ability to turn in place, increasing the overall maneuverability. The Rover's navigation system will use computer vision based on ArUco markers, which are small QR-code patterns. A camera will be mounted on the device's side that always faces the Docking Station, where a ArUco marker will be visible. Lights inside the Docking Station will illuminate the marker to allow for accurate navigation when there is little visibility. The navigation system will require microcontrollers to be onboard the Rover to process the image data and make calculations on where to move (see Embedded Control section). A battery is also required to power all of these components, so a 12 volt 10 amp-hour battery will be used.

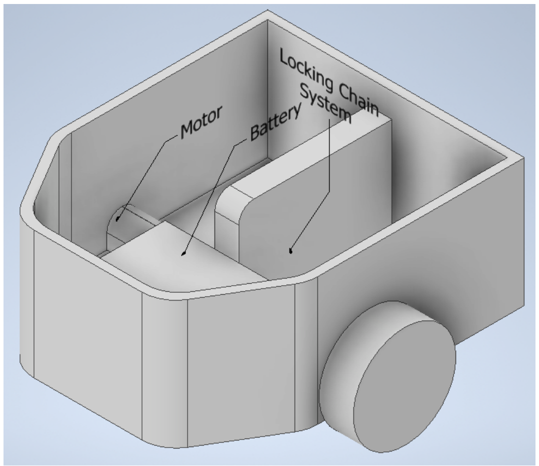
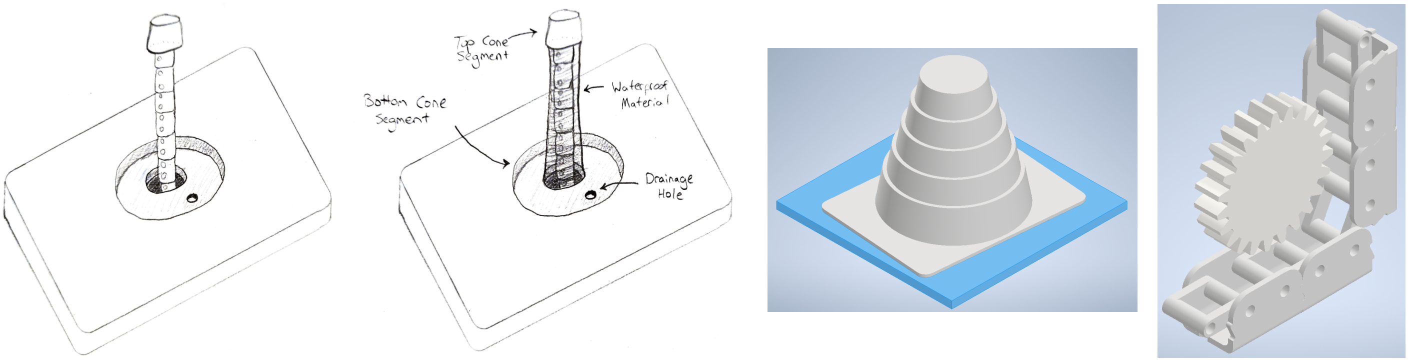


Figure 2 - Rover Mechanical Design

The Rover's visual cue for unauthorized vehicle situations will be in the form of a deployable traffic cone. The cone will be vertically extended when necessary to display a visual warning to 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 two feet high, in addition to the Rover's ride height. The cone itself will be made of segments, where the base of any given segment is larger than the top of the previous segment. When extending, the innermost segment is connected to the chain and lifts the next largest segment up. This continues until all segments are deployed and the chain reaches its maximum height. At this point the motor will lock and keep the cone upright. In addition, the Rover will use 4 LED lights positioned near the corners of the chassis to serve as further visual indication.

Weather

Making the Rover waterproof in inclement weather is difficult due to the cone segments having small gaps between them. To prevent rain water from affecting the components inside the chassis, 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.



Figures 3A, 3B, 3C & 3D - Barrier Mechanical Design

To combat hot weather, both the Docking Station and the Rover will have slotted vents. The Rover's vents will be placed on the side facing away from the Docking Station and the Docking Station's vents will be placed on the back side facing away from the driveway on the upper layer. The slotted vents will have an overhang to allow water to run off of them to prevent it from getting inside the body.

ELECTRICAL DESIGN

Top-Level Electrical Design:

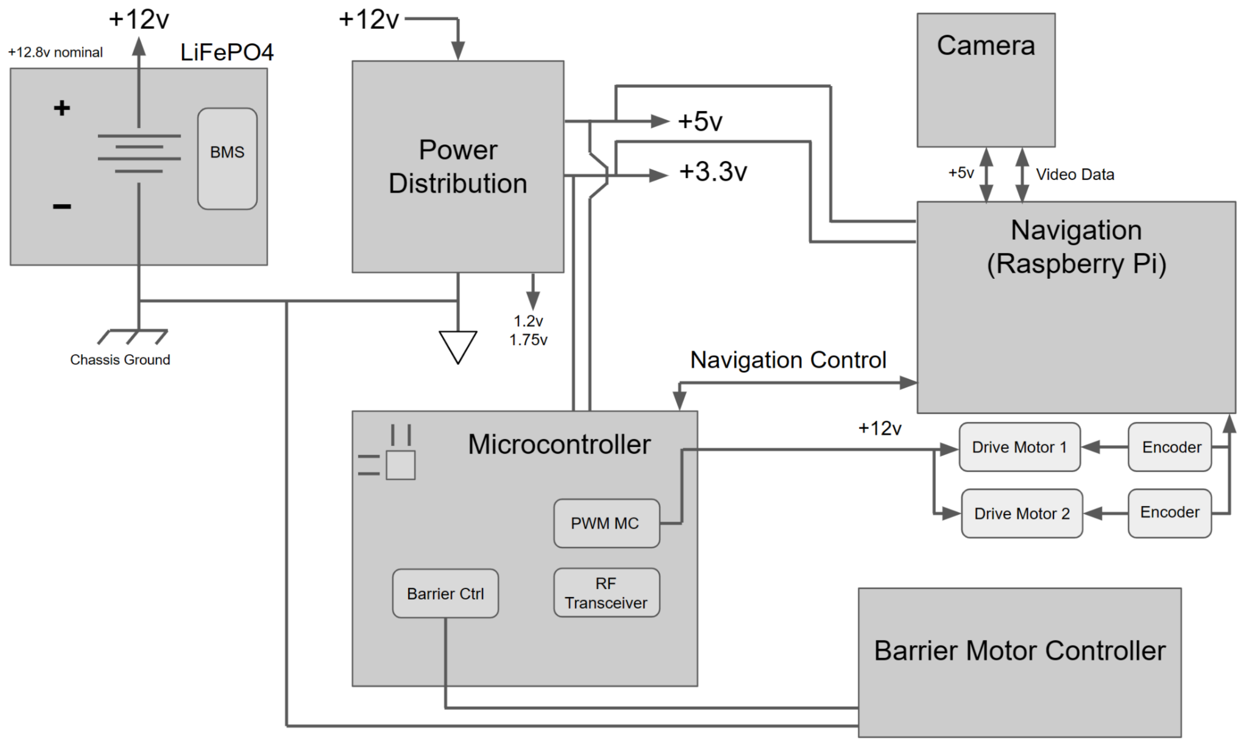


Figure 4 - Rover Top-Level Preliminary Design

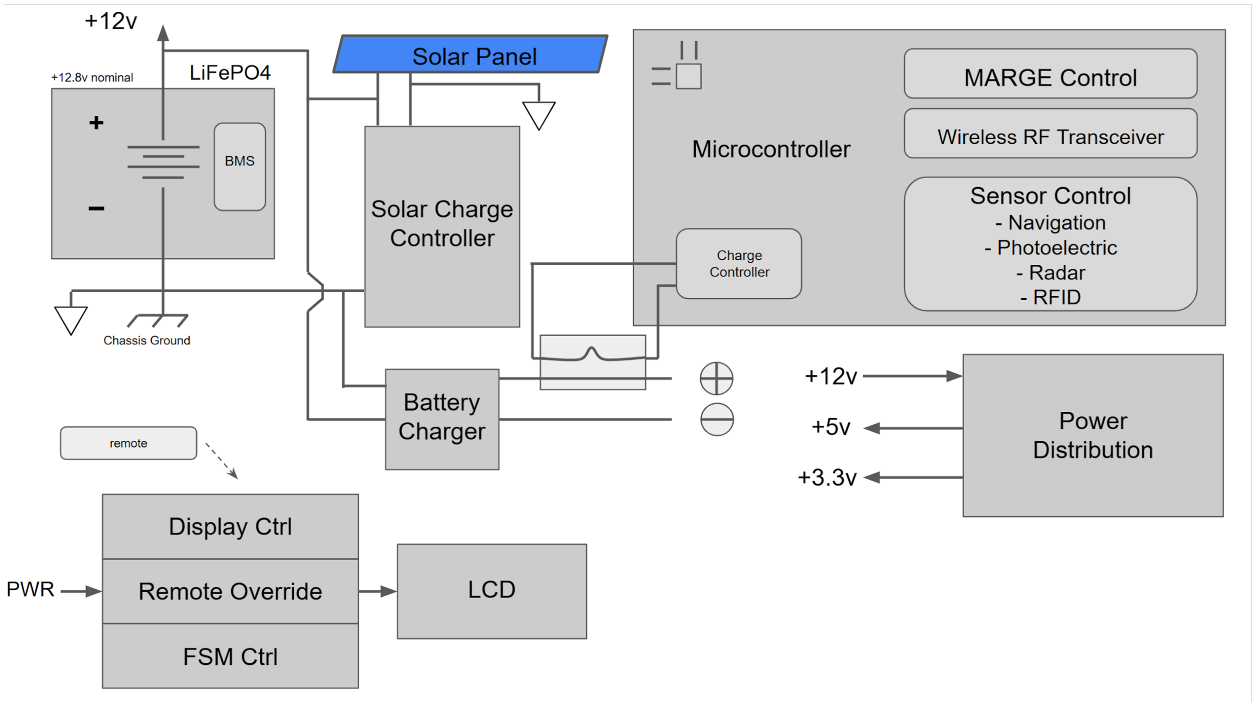


Figure 5 - Docking Station Top-Level Preliminary Design

<u>Batteries</u>

Both the Rover and the Docking Station will have their own respective system batteries. The Docking Station will charge its battery pack from the solar panel and solar charge controller. The Rover will charge its battery from the docking station battery.

The Rover battery voltage is proposed to be 12.V nominally. This decision involves several factors. Firstly, 12V is the automotive standard due to its reasonable power capabilities. A 12V battery is optimized for powering automotive level loads while not having to source too much current. In a car, this looks like 20-30A, while in our application that current draw is much lower. This leads into the second main reason, which is that our proposed motors run off 12V. Most DC brushless motors that provide the torque figures we require run off 9 or 12V, mostly 12V. It is imperative from an efficiency standpoint that the relatively large dumps of energy needed for driving the motors do not have to be stepped up or down through a regulator. In which case, we would suffer significant losses due to the efficiency of the converter, additionally purchasing a high power converter will add unnecessary cost to our design. Lastly, for powering the onboard sensors and various systems, stepping down voltage generally comes at higher losses when the difference between the input and output voltages increases. 12V provides a happy medium where we can directly drive our motors off the battery voltage, and reasonably step down the voltages through a power distribution board to our anticipated power needs, of which comprise of 5V and 3.3V loads at the moment. For calculating the Rover battery capacity, an analysis was performed based off of our current mechanical design, and basic assumptions and specs based on the RFPP.

Worst Case: 15 lbs Rover → 6.804 kg

Target Speed: 1 ft/sec → 0.305 m/s

Wheel Diameter: 6 in. → 0.1524 m

Differential Drive Steering System

$$F = ma \rightarrow (\overset{kg}{6.804})(\overset{m/s^2}{0.305}) \leftarrow \text{get to top speed in 1 sec}$$

$F = 2.075\ \text{N}$

Wheel Radii = 3in = 0.0762m → $\tau = F \times r = 0.158\ \text{N-m}$

$\tau = 15.83\ \text{kg-cm}$

*without friction

working estimate of rough torque needed to meet initial spec. → $\tau^* \simeq 16\ \text{kg-cm}$

Motor Selected:

~~24 kg-cm @ 156 rpm~~

50.9:1 Gear ratio

58kg-cm ; 59rpm

6 in diameter wheel: $6\pi = 18.85$ in Circumference

1ft/sec travel: $\xcancel{\frac{18.85\ \text{in}}{12\ \text{in}}} = \frac{12\ \text{in}}{18.85\ \text{in}} = 0.63662\ \text{rpm}$ (wheel speed)

$50.9 \times 0.63662 = 32.4\ \text{rpm}$

Continuous Current: 3.5A @ $12V_{DC}$

Maximum Current: 9.7A @ $12V_{DC}$

Target Spec:

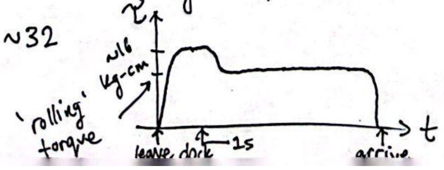
Travel distance Worst Case: 8ft each way (in/out of docking station)

I.C.& N.O. Mode: Travel only when neccessary

Minimum: 0 trips

Maximum: ~32

Rough Torque-over-travel:



Need to determine 'rolling' torque requirement:

Assuming Rover body is rectangular prism shape;

Moment of Inertia: $I = \frac{1}{12} m \left(L^2 + W^2 + h^2 \right) = 0.799 \, m^2 kg$

Assuming: 15 lbs rover,

Already at 1 ft/sec Velocity,

$L = W =$ ~~2ft~~ 1.5ft, $h =$ 1ft for body of rover,

Reasonable coefficient of friction; $\mu = 0.01$,

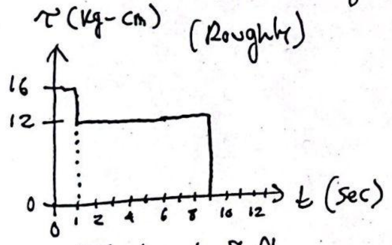
and wheel radii of 6 in;

Rolling resistance counter-force from friction;

$$N = m \cdot g = 66.52 N \quad , \quad F_{resist} = \mu N = 0.665 N$$

$$\tau = r \cdot F_{resist} = 0.1013 \, Nm = \boxed{10.13 \text{ kg-cm}}$$

→ Thus, the rover in this sub-optimal case will require ~16 kg-cm of torque to accelerate to 1ft/sec Velocity at a rate of 1ft/sec². Then, to maintain motion to the destination, ~ 10.13 kg-cm which we can round to 12 kg-cm to give further headroom in our design.



To travel 8 ft

Torque Request over Time of Travel

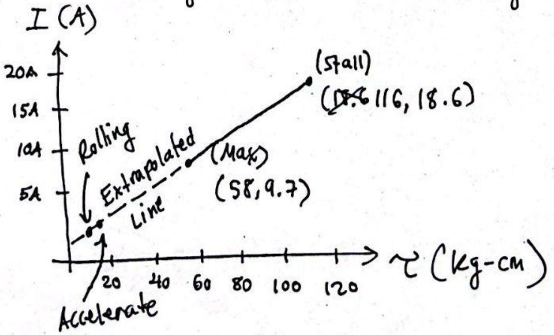
→ Assuming a roughly linear, proportional relationship between requested torque and current consumption; (continous current: 3.5A)

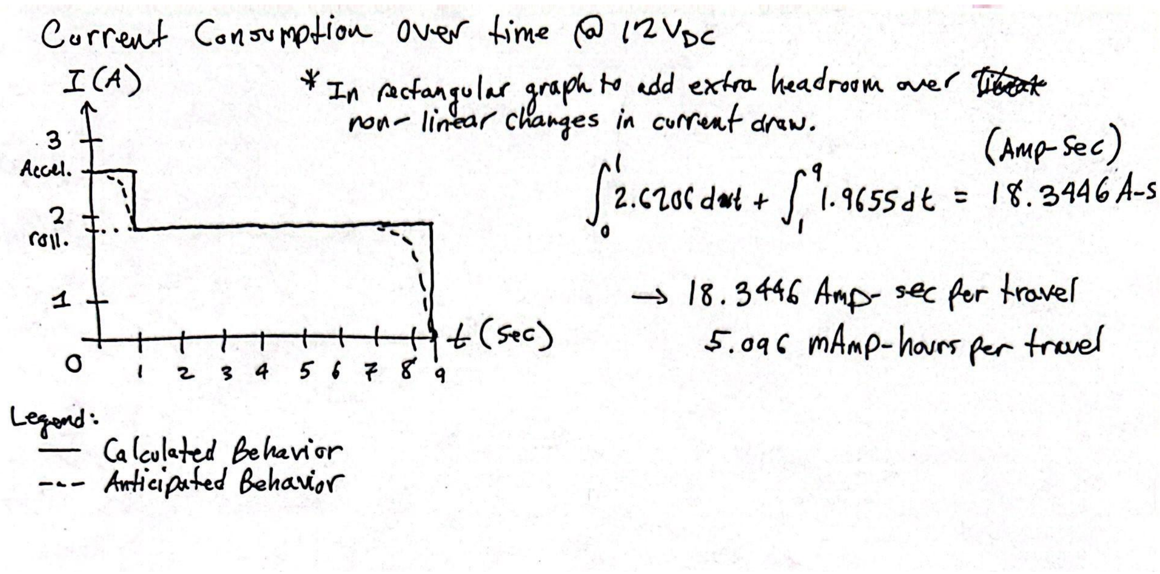
Motor Specs:	current:	Torque:
Stall:	18.6A	116 kg-cm
Max:	9.7 A	58 kg-cm
Accelerate:		~16 kg-cm
Rolling:		~12 kg-cm

$$\frac{1}{2}\left[\frac{18.6}{116} + \frac{9.7}{58}\right] = 0.16379 = \alpha$$

$$16 \cdot \alpha = 2.6206 \, A$$

$$12 \cdot \alpha = 1.9655 \, A$$





Figures 6A, 6B & 6C - Motor Current Consumption Analysis

Based off of this analysis, we can draw insights into the proposed battery capacity for this application, which for the purposes of this proposal will be 10Ah. This is in consideration of the fact that firstly, many dozens of trips may be performed in a day with only tens of seconds of charging time between runs. Secondly, a particular challenge of the normally closed mode is that the device may have to sit in idle, out in the pathway, for extended periods of time. This is while running the onboard systems, albeit in a low power consumption mode. Eco-Worthy, a company specializing in solar charging energy production and storage systems for camping, RV and outdoor applications, sells this spec battery for $50 with a built in BMS. Additionally, it has a 1C charge/discharge rate. This allows the ability to fast charge while at the docking station, and the ability to dump charge to the motors to produce high torque. The motors draw 3.5A continuously and 9.7A maximum, each. While we intend to not max out both motors at the same time, 10A should be maximally sufficient. Lastly, this size battery is roughly in the weight and volume targets of the device we are aiming to construct.

For the docking station battery, there are different considerations as compared to the Rover. Firstly, weight and size are less of a concern and take a backseat to overall capacity. The capability to run the Docking Station sensors and systems, and charge the Rover intermittently throughout a full night is paramount. Events such as a nighttime party could be considered the 'worst case' design scenario where many trips are required in a short period of time when sunlight is limited or not present. With this in mind, our proposal is a 12V nominal battery with a 50Ah capacity. This capacity is quite large, however this allows for plentiful storage for overnight, and extended periods of time where overcast conditions limit the solar charging capabilities of the Docking Station. The voltage is chosen for many of the same reasons as it was for the Rover, with the added benefit of simplifying charging due to the voltages being the same nominal figure. Eco-worthy sells a 12.8V, 50Ah LiFePO4 battery with a built-in BMS for $160. It is a deep-cycle battery which is beneficial for this kind of slow draw application where maintaining the nominal voltage for longer in the discharge cycle is important. Additionally, with a

40A/60A charge/discharge current limit the battery can provide enough energy to fast charge the Rover and easily be charged with a maximum charge voltage of 14.5V, which is right at the limit the same company's solar charge controller operates at.

In conclusion, for the batteries, two 12V nominal batteries, 10Ah and 50Ah for the Rover and Docking Station respectively, are our proposal. For both batteries we propose a Lithium-Iron-Phosphate chemistry. LiFePO4 is suitable due to its slightly higher nominal '12V' voltage, 12.8V, its ease of use for higher capacity systems and its common compatibility with solar charging accessories.

Solar Power

Solar capacity for charging is a consideration of power output and voltage. Considering our docking station capacity of 50Ah, a target time of full charge in bright sunlight would be a spec that would lead us to a solar power output spec. Being able to fully charge in a full day of sunlight, reasonably about eight hours, is important for recuperating energy discharged from the batteries overnight and preparing for the next night. This is all the while running the current power requirements. With 50Ah this puts us at 6.25A of charging current for eight hours. However, to comfortably allow the system to still power itself during the day with a net positive of energy gain, a spec closer to 8-10A would be more appropriate and is what we propose. Eco worthy also produces solar chargers and panels that are all vertically integrated to work together. The kit comes with a 100W monocrystalline panel feeding a 14.5V charge controller. This wattage provides us with ~7A at the charging voltage of 14.5V. The charge controller is of an MPPT type.

Battery Charging

Charging to the Rover battery will be done entirely through the Docking Station battery. We currently are not proposing any solar charging capability on the Rover itself. The charge controller we plan to use will boost the Docking Station battery to 14.5V and charge at a maximum of 8A. Additionally, the charge controller features a dial to change the rate between 2, 4 and 8A. Though the original intention was to use a single unit for the solar charge controller and the battery charge controller, after research and looking into the information available about these two products it was deemed best to keep the functionalities separated on two different units.

Energy Conservation

Energy conservation is a very important aspect of the design, particularly for the Rover. The normally closed mode defined in the RFPP calls for the Rover to operate for long periods of time with at best intermittent charging, if not no charge at all. Firstly, our design philosophy involves locating only necessary components for the functioning of the Rover on the device itself. All other functions are located on the Docking Station where a larger battery and power source are located. However, for the Rover we plan to design, in firmware and hardware, a low power mode in which the device can consume little power. Currently it is unclear the exact control of when this mode will be activated, but it will involve the Rover awaiting a move command, when in the normally closed mode, from the Docking Station where the vehicle sensors are located.

With regards to hardware on the Rover, the power distribution board will be installed with a power monitor and regulators with a shutdown pin. This is a common setup within manufacturer part families. The LTC2964 from Analog Devices is an example candidate for this role. Analog Devices produces linear and switching regulators that can be shutdown to draw only a miniscule quiescent current (order of microamps or less). This is an example of one hardware design approach for power distribution that can shut down non essential power supplies.

An early study of the Raspberry Pi power consumption seems to suggest that power consumption when returning from the driveway to the docking station will be approximately 800mA to 1000mA (5.1 W) when the Raspberry Pi is running the computer vision algorithms. The idle state of the Raspberry Pi consumes roughly 540mA or 2.7 W, this is unacceptable. Thus we propose an arduino controlled sleepy Pi power management board to be installed on the Raspberry Pi to reduce the idle power consumption to approximately 3mA and allow for fast reboots. This is a must considering the relatively small amount of time the Rover will be actively seeking the docking station using the CV.

Our visible barrier, described above in the Mechanical Design section, will be able to lock in place once fully extended. This will allow us to leave the barrier actuated and not consume extra power. Additionally, reflective tape will be utilized instead of LED lights to further reduce power consumption and provide better visibility at night and during a bright day.

<u>Motor System & Control</u>

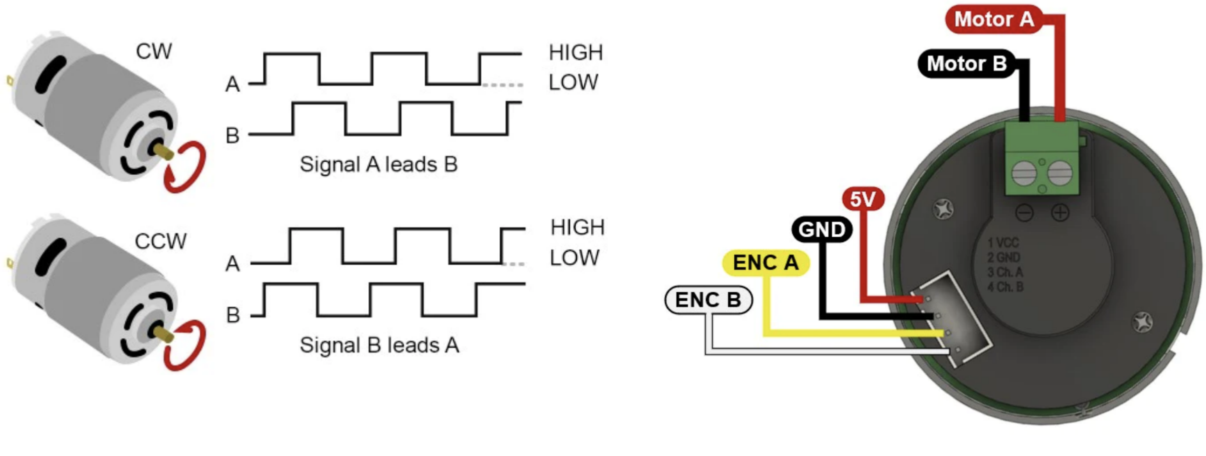


Figure 7 - Motor Interface

The Rover will be driven by two brushless DC high-torque motors. This will comprise of our differential drive system, controlling the steering of the vehicle by modulating the motor speed differentially between the wheels. Shown above is the interface of the motor itself. Motor A & B designate the power inputs driving the motor itself. The four connections below are for the built-in encoder, which runs on 5V and will return two PWM signals. The encoder is a hall-effect sensor based quadrature encoder which returns seven complete pulses per channel, per rotation of the motor. The frequency of the pulses designates the speed of the motor and the phase difference between the two PWM signals designates direction, with one leading and one lagging, as shown above.

For our motor controller, there are several functionalities required. Firstly, an H-Bridge circuit will control the direction of travel for each motor independently. Secondly, the capability to modulate the power delivered to the motor from the control of the onboard Arduino such that we can control velocity and acceleration. Lastly, there are many standard motor protections required such as overcurrent, overvoltage and reverse polarity. This is to protect not just the motor but also the supplying circuitry.

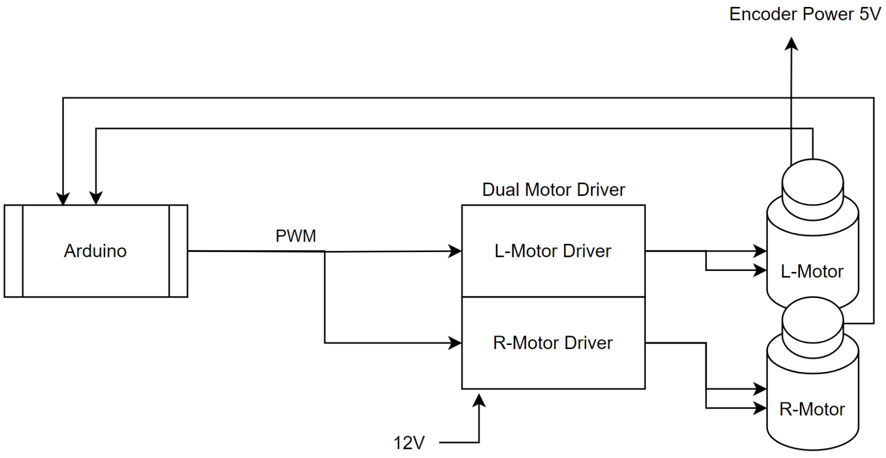
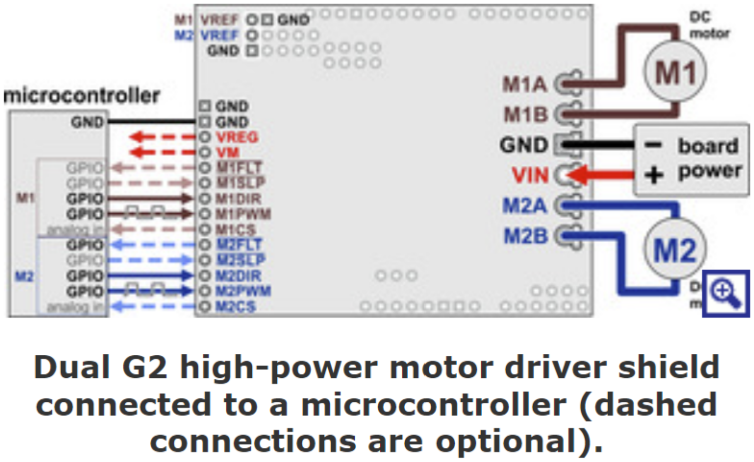


Figure 8 - Drivetrain Top Level Design

Above, a top level design of what the system would need to look like is depicted. PWM, in addition to being the method of reading encoder data, is also our chosen method for control of the motors due to its ease of use and wide support on Arduino systems. For our motor controller, we propose the Pololu Dual G2 High-Power Motor Driver 24v14. This controller features two independent H-bridge motor control circuits and itself is an Arduino shield. The controller has motor stall protection, undervoltage and short circuit protection. Additionally, it comes with six 100uF input capacitors for input stabilization and a current sensor which can be used to enforce user selectable current limits. Lastly, it comes with ample documentation, instruction and library support. This Arduino shield will be used as our motor control system and will be interfaced as shown in the figure below.



Dual G2 high-power motor driver shield connected to a microcontroller (dashed connections are optional).

Figure 9 - Pololu Motor Controller Diagram

In addition to our drivetrain motor system, motor control will need to be provided for the barrier actuation motor. This motor will be a much lower power model as actuating the light, 3D-printed barrier and chain assembly will not require much torque. Additionally, modulating speed is not necessary, a constant speed for actuation and the ability to reverse direction is all that is required. This will also be done with an H-bridge onboard a custom-built simple motor controller. This controller will feature the basic motor protections discussed in the drivetrain section. A PWM signal from the Arduino will drive a power mosfet, which will be specced to handle the current and power dissipation requirements. This will provide a rudimentary means for power control of which will then be connected to the H-bridge to control direction. A pair of diodes, a 1N4001 and zener for example, will be installed in parallel with the motor to provide basic overvoltage clamping and reverse polarity protection. A medium speed blow fuse will be installed in series with the motor current supply specced just below the continuous current limit of the motor to protect the motor and the motor control circuitry.

Vehicle Sensors

Our sensing apparatus for performing vehicle detection, authentication and exit involves two types of sensors. Firstly, for authentication of detected vehicles, we plan to use microwave frequency RFID. This involves a reader module onboard the Docking Station and then passive tags distributed to the operators of the authorized vehicles for installation on the windshield. RFID is a growing standard for smart parking lots and home security protection systems for automobiles. It is fast, secure and reliable. The particular set of units we are looking at are all designed with 24/7 outdoor operation in mind. An important drawback to note is that microwave frequency RFID readers are hard to find within a reasonable budget, as a result UHF RFID readers are also under consideration. Products being considered are ones that have common communication interfaces, examples being I2C, SPI, UART and CAN. Though CAN is considered less preferable for this project. Additionally, due to cost and range concerns, an RF receiver and broadcast module are under consideration. This would involve a single receiver onboard the Docking Station with a RF broadcast device given to each verified vehicle. This would likely be a custom built, battery powered solution like the remote. Due to these concerns above, the authentication system is still not fully finalized at this time.

Lastly, for our vehicle detection, we will be using a 24GHz radar array. The NATO defined K-band is the standard frequency range for automobile detection and can detect cars as far as 200m away due to their reflective material composition. There are several units under consideration at the moment however the proposal is to mount the radar array facing the exit of the driveway. A 'radome' style covering will have to be used in order to effectively protect the electronics, but not interfere heavily with the operation of the sensing. Though the manufacturing means to do this is currently unknown, an optimal radome covering would feature a material thickness some integer multiple of half of the wavelength of the radar wave. This is a well known practice to reduce the effect of the radome covering, as it makes it "invisible" to interference. Currently, the reverse detection profile (the rearward viewing) of the radar array is being considered as a candidate to detect an exiting vehicle from the driveway. However, in an effort to maintain a strong user's experience where vehicles traveling up to the MARGE do not need to stop for it to move out of the way, a dedicated sensor is the backup proposal. Currently, it is believed that sensor arrays will need to be purchased and then a radar controller developed in house. However, additional spending can overcome this problem by buying a fully integrated module.

Important to note for radar is that it has unparalleled performance in adverse weather conditions and during the night/day cycle. Additionally, radar sensors are very parallelizable. Most importantly however, is that we can determine the direction and velocity of heading of a target in addition to simply detecting its presence. We plan to use this to great effect, crucially to allow our sensing range to increase all the while reducing the chance of the system being inadvertently tripped by roadway traffic not in the driveway of concern.

As a direct response to the RFPP. We propose the following specifications for distances:

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.

The NO MSSB is rounded up from 22.35m which is the distance a car will travel in 5 seconds when at 10mph, which is the maximum approach speed we will design for. With the reasonable driveway width being approximately 5-6 feet, the Rover should reasonably be able to make it to the center of the road before the car's arrival. The extra distance provides ample headroom for a driver to notice the Rover. The Rover will be proposed to drive at 1ft/sec when traveling to/from the Docking Station in Normally Closed mode. This is to conserve energy in a less time dependent operation environment. However, for Normally Open mode, we propose to drive the Rover out at 2ft/sec to reduce the time required to intercept. For the Normally Closed mode, this minimum distance will require a vehicle to slow down if they are driving above 5 mph. The 10m spec is chosen such that the system can function in short driveways in at least one of the two main modes, and so that it can function easily for the user given that they drive slow (below 5 mph).

3. The Recognized Vehicle Sensing Distance (RVSD) target will be 30m.

This target is on the shorter side of the reasonable range for this property of our system. This is based on the MSSB for the normally open mode which is limited by our authentication range in the current RFID plan. However, if the switch to an RF broadcasting solution is deemed necessary then this range can be drastically improved. This is however at the cost of having to design our custom car broadcast devices and increase the cost of adding new vehicles to the user. Although, the total cost could be significantly reduced overall.

4. The Unrecognized Vehicle Sensing Distance (UVSD) will be 120m.

This is based on our radar sensing capabilities, minus a small factor due to an additional loss in range needed to classify a car as approaching or as orthogonal traffic. We believe we will be able to detect presence as far as 150-180 m from the MARGE.

Navigation Hardware

The necessary on Rover navigation hardware includes an Arduino Mega (STM32H7 MCU), wheel encoders built into the DC motors, a MPU6050 gyroscope, the Raspberry Pi Camera Module V2-8 Megapixel 1080p, a Raspberry Pi 4, 4GB, and a cooling tower for the Raspberry Pi.

The raspberry Pi will be mounted on the Rover chassis and will be powered by the onboard battery. The Raspberry Pi will be preprogrammed with openCV, which is the software library utilized to abstract the low level computer vision and image processing algorithms used for localization of the robot. A raspberry Pi camera module V2 will be mounted and configured on the Raspberry Pi and positioned to point at the docking station. The docking station will feature one ARUCO tag mounted to the front face above the charging port.

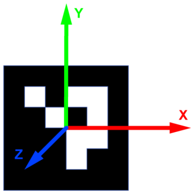


Figure 10 - Aruco Fiducial Tag

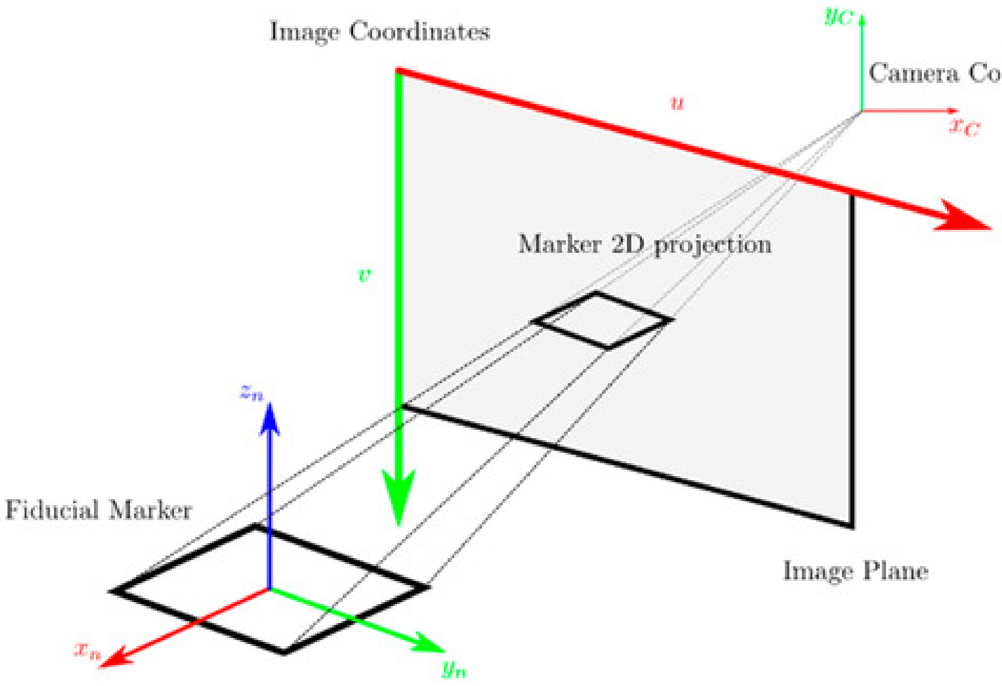


Figure 11 - Computer Vision Localization with ARUCO tags

<u>Docking Hardware</u>

The docking hardware shall include the Raspberry Pi camera module V2 and a single Aruco tag mounted to the charging dock. Additionally, we propose an additional external sensor. However, further research is necessary in analyzing how this control input functions in parallel with the CV algorithm and the pure pursuit algorithm.

To detect if charge is coming into the Rover when the charge prongs on the Rover have made contact with the Docking Station's contact points at the back of the garage, an ISL28022 precision digital power monitor is proposed for use in order to detect when a charging current is present. It is both a high-side and low-side digital current sensor and voltage monitor for the bus voltage that also stores power calculations to or from the load that has an on-board ADC and an I2C interface. It will require an external shunt resistor in order for the current measurements to be enabled.

Microcontroller Selection

The rover is intended to run off of an Arduino Mega 2560 Rev3 microcontroller. This selection was made due to the excess of ports on the board. The Mega 2560 boasts 54 digital IO ports (with 15 allowing PWM), 16 analog inputs, as well as 4 UART ports. The Mega operates at 5V with a recommended input between 7-12V. The Mega is 101.52mm by 53.3mm with a weight of 37g.

Any number of issues can arise during testing/construction, so having more ports than necessary should allow for many changes to be made without having to invest in/upgrade our microcontroller. Additionally, by selecting an Arduino board, access is gained to a wealth of libraries and documentation that other choices may not provide.

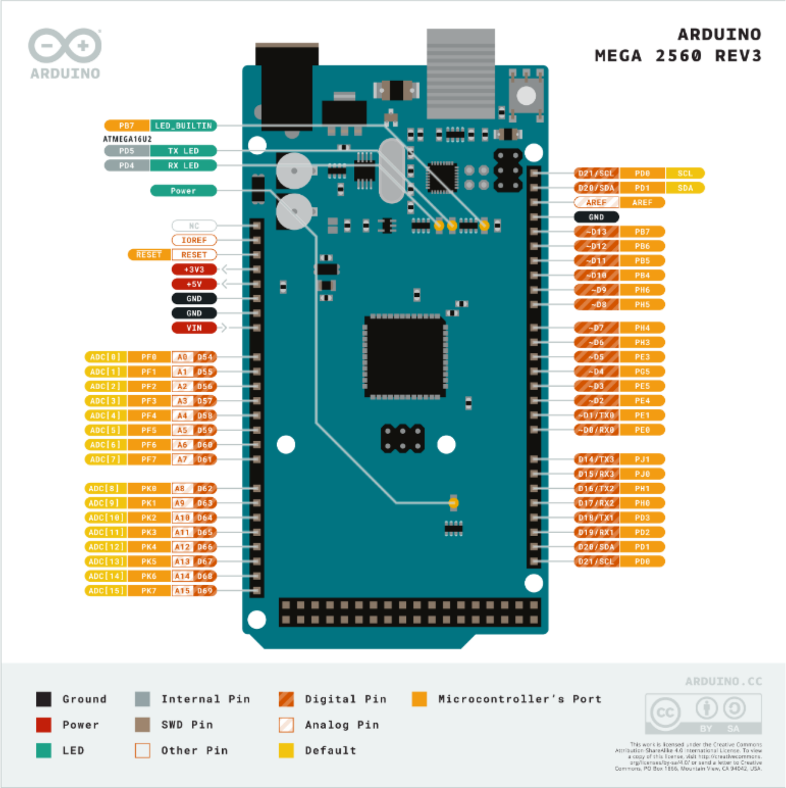


Figure 12 - Arduino Mega 2560 Rev3 Pinout

EMBEDDED CONTROL

Navigation System Control Introduction

The differential drive Rover will navigate to and from the Docking Station using two localization subsystems: computer vision image processing and odometry, called Nav-S1 and Nav-S2 respectively. The top level control law shall perform fault handling as well as navigation subsystem management. The navigation subsystem management shall be implemented to save power and computational resources. The control law will perform a periodic (~once a day) loss calculation on the error between the CV odometry and the encoder odometry, by running them in parallel. An algorithm will analyze the localization for each time stamp and determine if the error margin meets an acceptable error tolerance. If this is the case, the robot will use the simpler but less accurate Nav-S2 (encoder odometry) to get to the center of the driveway rather than the computationally heavy Nav-S1 (CV odometry), while maintaining protocol Nav-S1 to get back to the docking station due to the necessity of precise control. If the error tolerance is not met the algorithm will make the alternative classification: Nav-S2 (encoder odometry) is not viable due to slippage or other factors. These factors can include but are not limited to unequal wheel diameters, asymmetrical robot, misalignment, limited encoder resolution, uneven terrain, etc. In this case, Nav-S1 will be used to get out and back to the docking station. This methodology has the performance benefit in minimizing resources when they are unnecessary: encoder odometry as an alternative to vision odometry has the potential benefits of computationally faster, lower power constraints, and software simplicity.

Top Level Functionality with Fault Management System

The top level navigation controller will take a single distance input from the Docking Station to the center of the driveway based on the driveway's spatial configuration. Nav-S1 and/or Nav-S2 will be used to determine the robots approximate pose relative to the docking station based on feedback from the openCV odometry data or encoder odometry data. Based on this localization data, inverse kinematics will be performed to provide inputs to the motor velocities to fulfill the desired pose. The aforementioned top level probabilistic support system will be in place that will favor either of the navigation protocols to meet error and power constraints. The support system will sit above both navigation protocols and take three inputs: the localization data Nav-S1, the localization data of Nav-S2, and the robot's current goal trajectory: either "Docking" or "Driveway". Then it will use a weighting system to generate loss for each protocol; the algorithm will favor the protocol with minimal loss. This decision will be factored into the localization data given to the pure pursuit algorithm which computes the motor velocities. When docking the system will always choose Nav-S1 however when moving to the "Driveway" goal, Nav-S2 may be favored due to power, speed and processing constraints on Nav-S1. In the case that the localization data varies between Nav-S1 and Nav-S2 to a degree that one system appears probabilistically compromised, faults will be generated. A Nav-S1 fault will have a far higher priority than a Nav-S2 fault. If a critical Nav-S1 fault is encountered the robot will halt operation and perform a diagnostic. If it is determined that probabilistically Nav-S1 is failing due to highly sinuous CV localization data, a critical fault will be generated, and if un-handled, Nav-S2 will be entirely favored. Nav-S2 faults are less troublesome, and depending on the fault in question, the robot will resume normal operation but favor Nav-S1.

Nav-S1 Functionality

The first subsystem shall use ARUCO markers with the openCV computer vision library that processes images of AR tags to be used for fine-tuned localization of the robot. The library uses advanced image processing techniques to estimate the camera's pose given the images taken of the ARUCO tag. The process follows the following general steps: analyze square shapes that may be ARUCO tags, extract the marker bits, followed by pose estimation by calibrating the camera and then using a 3d transformation from the marker coordinate system to the camera coordinate system; specified by rotation and translation vectors.

One tag will be mounted on the charging station and the camera will be fixed in a known position on the robot that will be used to calculate the pose of the robot relative to the charging station. The pose of the robot factors both position and orientation. An alternative to the aforementioned method would be to mount the camera on the charging station and mount the tags on the robot. Thus forward kinematics rather than inverse kinematics would be used to estimate the robot's relative pose from the camera position at (0,0) within the configuration space.

Nav-S2 Functionality

Nav-S2 will compare the encoder localization to the goal configuration and compute the required motor velocities to create a trajectory to reach the goal configuration. The gyroscope will act as a support to the top level Nav-S1 controller to precisely maintain a straight course despite asymmetrical robot implementation, uneven terrain and other environmental factors.

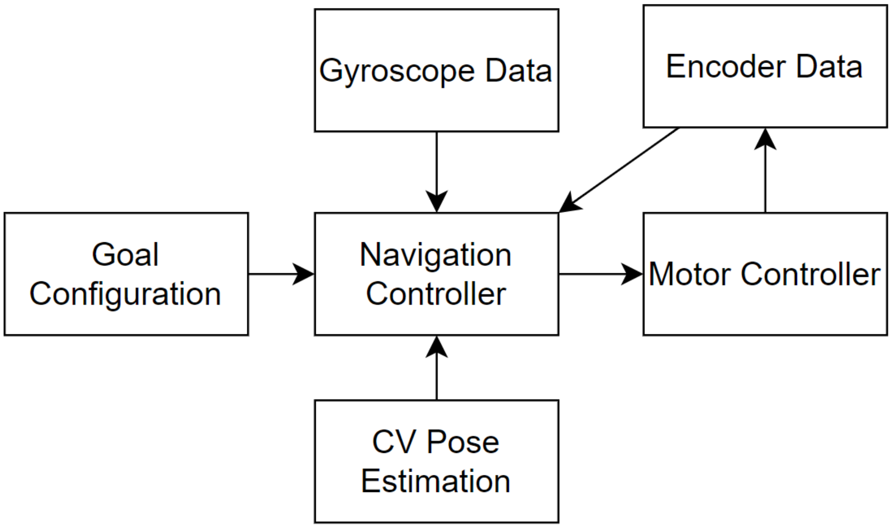


Figure 13 - Navigation Top Level Interface

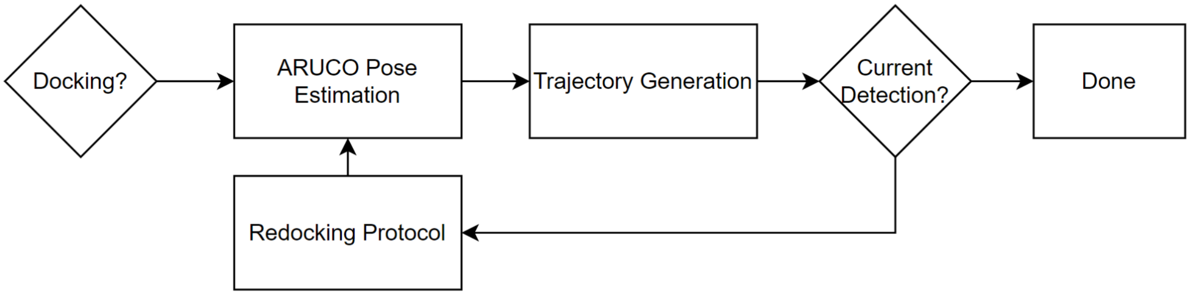


Figure 14 - Docking Protocol

Docking System

We propose a docking protocol that shall be initiated from top level inputs. The protocol shall use computer vision pose estimation coupled with the pure pursuit algorithm. The pure pursuit algorithm uses the visual This technique facilitates the high precision needed during the docking process. An active high current sensor shall be mounted on the robot that will output an active low to the arduino to indicate the presence of current. Thus, this detection shall facilitate the termination of the protocol (current detected) or alternatively a realignment procedure (no current detected).

Software Behavior

The "brain" of the Rover is designed using Finite State Machines (FSMs). In the final design, there will be an overarching top-level FSM that connects the behavior of each operating mode (Top Level Robot FSM and Top Level Garage FSM) . The FSM design that we have come up with for the garage and robot would add an additional three micro controllers to the design (without including one for the remote).

The first diagram below represents the top level FSM for the robot. The purpose of this FSM is to determine what mode of operation the robot should be in. The FSM starts with initialization and a startup state which is used to gain parameter values from the user along with initialization of any sensors. In order to switch modes of operation this FSM relies on a reset state that sends the robot back to the garage to switch between modes. Within these modes, will kick off their own FSMs which will be further discussed below.

It is worth noting that there is an additional FSM at the top level to constantly be receiving receiver values from the remote. Upon a change in mode (indicated by the remote or garage) this FSM on a microcontroller (arduino) will pick it up using the NRF2401 and will be communicated to the main microcontroller using a Uart connection to allow for more SPI connections and for ease.

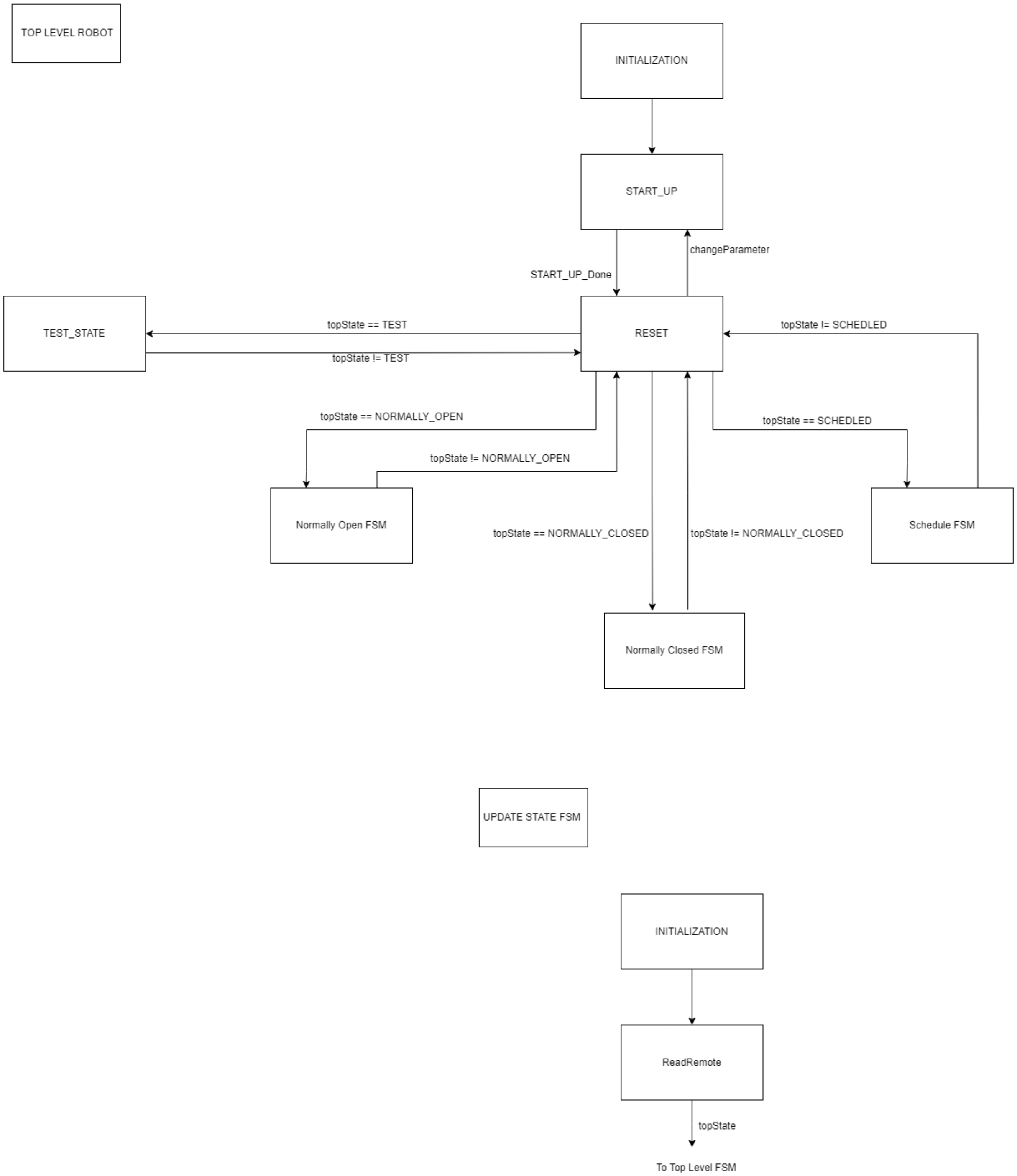


Figure 15 - Rover Top Level Finite State Machine

The garage for the robots FSM will be able to control parameters set up (ex middle of the road distance) along with monitoring and changing mode of operation. The monitoring will allow the garage's display to show values of the robot (ex percent of battery, or number of cars passed). The garage will also use a NRF2401 transceiver to send and receive signals from the robot.

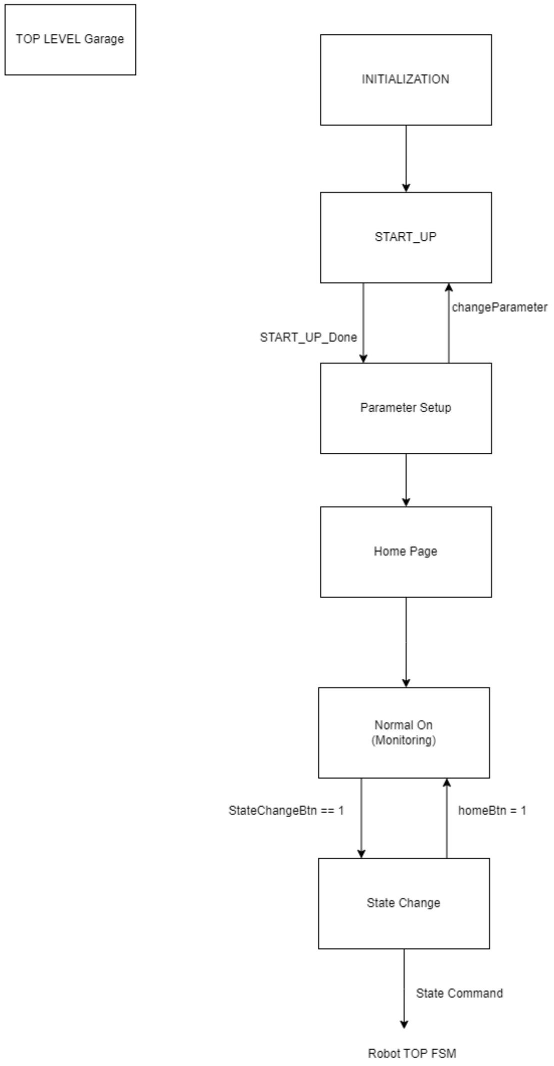


Figure 16 - Docking Station Top Level Finite State Machine

In Mode A (the "Normally Closed" mode), the MARGE system is initialized and then reports to its position in the middle of the driveway. The robot stays in this position until a car is detected by the navigation system. The Rover will determine if the vehicle is recognized, and it will move to the docking station if this is the case.

Otherwise, it will stay put. Once in the docking station, the Rover will wait until the approved car passes before returning to its position in the middle of the driveway.

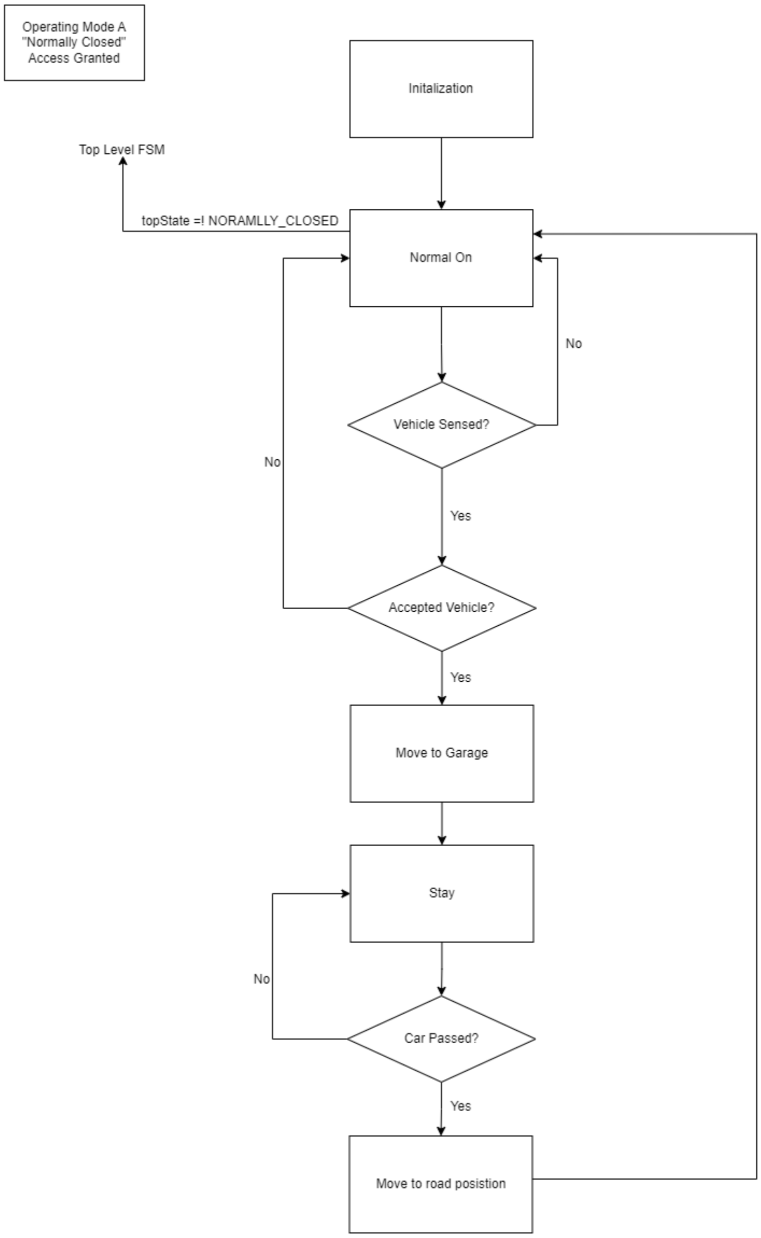


Figure 17 - Mode A: "Normally Closed"

In Mode B (the "Normally Open" mode), the robot is initialized but then stays in the docking station. While in the station, it will use its navigation sensors to see if a vehicle is coming. If a vehicle is approaching and it is unrecognized, the Rover will leave the docking station and move to the center of the driveway. Once the unrecognized car turns around or the user presses the “Override” button on the remote, the Rover will return back to its position at the docking station.

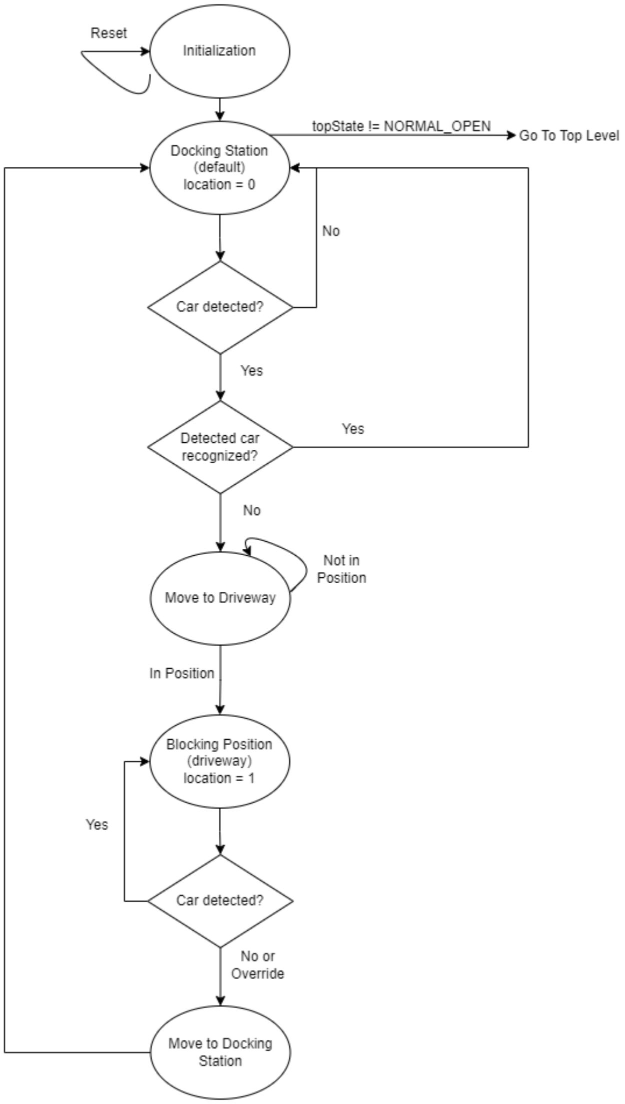


Figure 18- Mode B: “Normally Open”

In the Scheduled mode (operating mode C) the FSM will default to the garage in order to be used by the reset in the overarching FSMs reset state. After we know that the robot is in the garage it will be ready to be used in the scheduled state. The scheduled state relies on values given during the START_UP state in the main FSM in order to get an understanding of the scheduled bounds the robot will need to be out for. When this knowledge is required the robot in the scheduled mode will go out within the time bounds and return when not. This mode also would make use of an interrupt to leave which would send the robot back into the garage upon the signal being received in order to flawlessly transition to a different mode of operation.

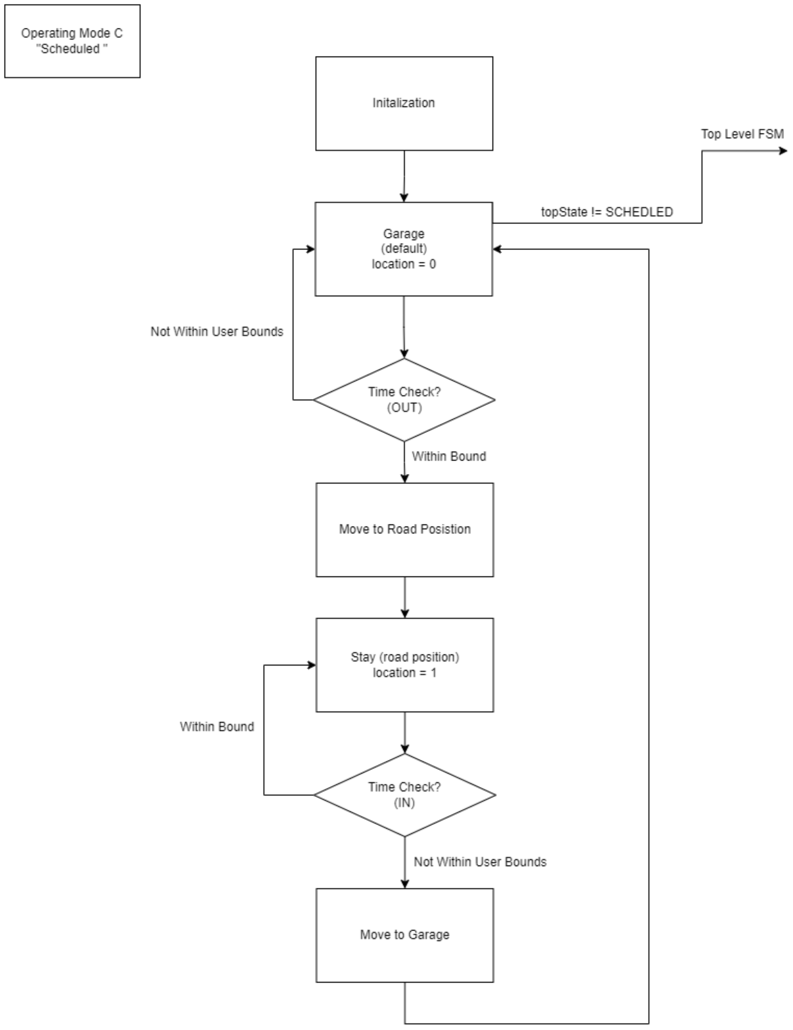


Figure 19 - Mode C: "Scheduled"

USER CONTROL

User Manual

The end user should receive 3 packages (or one very big, sectioned package). There should be the rover, the dock, and the solar panel, separate from each other.

1. Place the dock at some distance away from their driveway.
    a. Attach the solar panel to the top of the dock with included hardware.
2. Place the rover in the center of the driveway in line with (and fiducial facing) the dock.

3. Turn on the dock.
4. Turn on the rover.
5. On the dock, press the “calibrate” button.
    a. The GUI should show a progress bar with the expected time to finish.
    b. This should involve initial calculations from the Raspberry Pi navigation module to determine the exact position of the rover with reference to the dock.
    c. After initial calculations, the rover should drive to the dock and park.
        i. The rover will then begin encoder calibration where it drives to the selected point (and back to the dock) saving the encoder data and averaging it each time.
        ii. Once finished, the rover will remember this value and attempt to achieve it every time it leaves the dock.
    d. Once calibration is done (the rover will be in the dock already), some effect will happen (lights flicker, tone plays, etc.) and the GUI will show a “calibration finished screen”.
6. At this point, the user can leave, and the GUI will automatically shut off.

On-Dock GUI

The dock portion of the system will have an LCD screen used to return system data to the user and enter test mode. This screen will have an emphasis on being backlit and sunlight readable. It will be controlled by buttons mounted beneath it. Each informational panel will have an individual button that corresponds to it (there will be no nested panels).

Figure 20 - GUI with Button Layout

Buttons (in order) included will be:

1. **Power** - If off, this will turn on the screen and display a welcome message. If the screen is on, it will turn off instead.
2. **Selector** - This will come either in the form of buttons or a knob. This will be used to select specific values on the current screen.
3. **Input** - This will come either in the form of buttons or a knob. When looking at a value the user is able to interact with, they can use this to either increase or decrease it. There will be pre-programmed bounds that the user can not exceed to preserve the system.
4. **B1** - Opens the settings screen. On this screen, users will be able to see (and manipulate) important settings such as: calibrated encoder values, distance to driveway center, or scheduled mode times.
5. **B2** - Opens the diagnostics screen. On this screen, users can view data meaningful to the operation of the rover such as: current battery life, time since last dock, cars blocked today, or solar energy received on dock.
6. **B3** - Begins calibration mode. This button will begin the calibration routine, only if the rover is not currently in the dock. See the "User Manual" section for information on the operation of this mode.
7. **ADDITIONAL MODE** - By inputting some sequence of buttons (or holding one down) the user will enter development mode. While in this mode, the user will have full control over any manipulatable values. The end user should have zero reason to access this mode.

The MARGE system should limit end-user input to its minimum value. As such, this LCD screen should be treated mostly as a developer panel and documentation should avoid having an end-user interact with it for extended periods of time during setup.

After initial setup, the user will have to run a calibration routine by pressing a button on the dock. At present we would like to have this to be as simple as possible with only asking questions about the surrounding area such as the distance to the middle of the driveway. During this calibration, the user will be asked but not required to give values for the scheduled mode.

Remote Control

An important feature of the MARGE system is the remote control used to send commands to the Rover from a more distant location. This remote control will be created using a NRF24L01+ transceiver module that communicates at a 2.4 GHz frequency. On the remote itself, the transceiver (acting as a transmitter) will interface with an Arduino Uno and several buttons. The buttons, numbered in the design below, serve the following purposes:

POWER: Toggles the MARGE device on and off.
1 (NORMALLY OPEN): Switches the MARGE device to the Normally Open mode.

2 (NORMALLY CLOSED): Switches the MARGE device to the Normally Closed mode.
3 (SCHEDULED): Switches the MARGE device to its Scheduled mode.
4 (DRIVEWAY): Sends the Rover to the center of the driveway, regardless of which mode it is in.
5 (DOCKING STATION): Sends the Rover to the docking station, regardless of which mode it is in.
6 (CONE UP/DOWN): Toggles the contractible cone to go up/down.
7 (LIGHTS ON/OFF): Toggles LED lights on the Rover to turn on/off.

A second transceiver module (acting as a receiver) will be attached to the docking station for the remote to communicate with it. The transceiver pair operates at a 2.4GHz frequency, and without obstacles it can communicate at a distance of 1100 meters (approximately 3600 feet). The messages sent from the remote to the receiver will be dependent on the buttons pressed, and those signals will be crucial in controlling the elements of various FSMs within the MARGE subsystem. The mockup below shows the tentative layout of the buttons, the microcontroller, and the transceiver, but likely the electronic components will be hidden under a mechanically-designed casing.

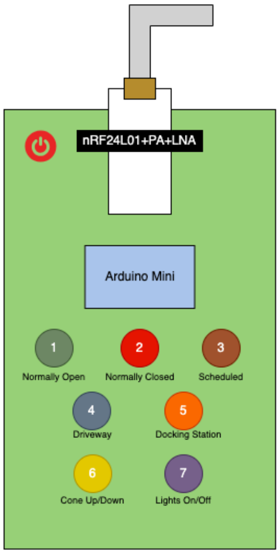


Figure 21- Operator Remote-Override Controller Top Level Design

Libraries
The libraries used include openCV, arduino encoder library, Raspberry Pi OS.

Arduino Library for Radio
Arduino Library for the nrf2401
SPI for Arduino

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*