Introduction to Engineering Design with Professional Development 1

Final Report for
Emergency Rescue Rover

Team: Rescue Rover
Section 3
Instructors: Glen Gross and Tathagata Bhaduri

Version 2.0
April 27, 2022

Prepared by:
Daniel Mandras (2023 / Electrical Engineering)
Jonas Kendra (2024 / Mechanical Engineering)
Andres Lin Huang (2024 / Biomedical Engineering)
Diego Fernandez (2024 / Materials Science and Engineering)
Alexis Hernandez (2024 / Aeronautical Engineering)
Ashton Ivey (2024 / Mechanical Engineering)

Executive Summary

The emergency rover was designed as a low-cost option for first responders in local townships. This includes police, firefighters, ambulance, and any individuals who are put in danger in their jobs. The rover was designed to be useful in a variety of environments, and capable of withstanding a variety of environmental hazards, whilst remaining easy to operate, and delivered at a low cost to consumers.

Namely, the rover is robust enough to withstand forces of nearly 3 kN with no more than a dent. It can maneuver over small obstacles, and it can withstand temperatures above 200° F for a period of time. The runtime of the rover also exceeds the projected value, running for close to two hours, which is almost double what the expected runtime of one hour was.

Despite the successes of the rover, there is more that can be done to expand upon the progress made by the team. Improvements can be made to reducing weight, improving internal components, and improving upon mobility and diversity of sensors.

Table of Contents

Revision History

Table 1 - Revisions

Version	Date	Name	Reason for Changes
0.0	04/07/2022	Version 1	Initial document.
1.0	04/13/2022	Version 2	Addition to each section
2.0	04/27/2022	Final Version	Revision

1 Introduction

This project was conceived as a low-cost alternative for local emergency response teams to alleviate the necessity to put first responders' lives at risk when surveying a fire or disaster situation.

Data acquisition is important across many different fields, but prohibitively expensive devices prevent organizations from gathering information that is crucial to their operation. Whether it's surveying an emergency situation, gauging atmospheric conditions after a natural disaster, etc.

When doing research into alternative products, it was clear that nothing was commercially available at a reasonable price point. This is largely due to the R&D costs associated with creating such a high-tech device. Many companies aim to sell a few, high end products. Namely, most products benchmarked were sold for tens of thousands of dollars, or thousands for a bare-bones chassis. Because of this, the team decided to create a low-cost alternative that is more accessible for the general populous.

Many of these data acquisition rovers have many characteristics in common, which the end user deems necessary to solve their problems. These include maneuverability, ease of use, efficacy of data acquisition, and dependability. Versatility is also a notable requirement, as many users in our target market prefer a tool that can be used in a variety of situations, rather than a highly specific tool for one situation.

The rescue rover tackles all of these problems. Treads and suspension allow for maneuvering over difficult terrain. Users can operate the rover effectively with just minutes of training, and data acquisition occurs every second. In terms of versatility, the rover's interchangeable components and sensors allow for utilization in just about any situation. Because of the rover's rugged design, it can withstand just about any force it is subjected to, which leads to an extremely reliable, effective product for end users.

2 Project Objectives & Scope

In scope:

- Create a single working prototype rover
- Perform calculations to justify components
- Test rover in variety of environments/ways to confirm efficacy

Out of scope:

- Scale up production to commercialize product
- Advertise and begin to market product

2.1 - Mission Statement

Produce a robust, reliable device that can be cheaply manufactured in order to serve its purpose for the emergency response community.

2.2 - Customer Requirements

Table 2 - Customer Requirements

Customer Requirement	Technical Specification	Target Value / Range of Values
Robust/Durable	Withstands high-force hits from outside (kN)	Withstands 1kN force to chassis
Easy to Deploy	Fast to deploy	Under 1 minute to deploy
Portable	Weight	30 lbs max
Simple to Operate	Operates on pre-existing hardware	Software functions on Windows, Mac
Won't Get Stuck	Traverses rubble, small debris, stairs	Goes up stairs, up to 9.5" rise, 9.5" run (OSHA max)
Withstands Field-Temperature	Survives in fire-like temperatures	200 Fahrenheit / 93 Celsius for ~3 minutes
Records/Provides Useful Data; Protected Data	Sensors transmit data live, save data periodically	Live feed w/ .5 sec delay, data saved each minute to cloud
Versatile In Utility	Modularity/sensors can be easily replaced	Min. 2x sensors, w/ ability to replace
Doesn't Hurt Users	Safety	0 casualties
Range of Area that System Covers	Range Radius	25m
Inexpensive from a college student's perspective	Dollars per person	$25-50
Consistent	Time before failure arrives	5 years

2.3 - Technical Specifications

Please see section 2.2, table 2 for Technical Specifications.

3 Assessment of Relevant Existing Technologies

Table 3 - Competitive Benchmarking

Competitive Product	Title / Description	Relation to this project
Foster-Miller Talon:	Military funded rover, used in reconaissance, and boasting the ability to utilize small arms; costs $60k+ per unit	This is a much more expensive version of what our rover aims to achieve, but this comes at the cost, of an extremely high cost per unit. While our unit can

Competitive Product	Title / Description	Relation to this project
		be retailed for roughly $500-1k, this product is a minimum of $60k per unit. Of course, the Talon is more robust than our project, but the price range makes it unusable for our target audience.
R6 Light Commercial Rover (Patent Pending)	Commercial grade rover; sold as rolling chassis, and meant to be coded by end user for their own needs.	At a price of $1900 per unit, for only a rolling chassis, this product has a high price point, for an unfinished, unusable product. At a significantly lower price–point, our users could do more as an out-of-the-box product. [9]
Jet Propulsion Laboratory robotic two-wheeled vehicle	Nasa's two-wheeled rover, operating with sensors within two barrel-type wheels	This product is not commercially available. It is very specifically made by NASA and few have been made, making it not a viable competitor, despite being a good benchmark for effectiveness

Table 4 - Patent Research for Related Technologies

Patent Number	Title / Description	Relation to this project
WO2007058868A2	Foster Miller Talon	Military-grade version of our rover. Sold for 60K per unit. Price is astronomically high, but boasts some useful features that we intended to emulate.
8496077	Robotic two-wheeled vehicle	Operated to look like a bike, sensors are stored inside both, drum-like wheels. While less

Patent Number	Title / Description	Relation to this project
		effective at maneuvering, this NASA patent is highly effective at data acquisition, something which we modeled our own rover after

4 Professional and Societal Considerations

Our team applied the engineering design process to produce solutions that meet the specified needs with consideration for the topics found in Table 4 - Engineering Solutions Impact.

Table 5 - Engineering Solutions Impact

Area of Impact	Impact	Description of Impact
Public Health and Safety	5	The rover is intended to improve the health and safety of first responders. Ideally, our product is directly intended to save the lives of responders. However, this may theoretically come at the cost of response time to fires, crime scenes, or medical response. This cost in response time, however, helps ensure the responder's safety, allowing them to be in the field longer, and preventing field-injury or death. Summed up, our product keeps first responders safe, despite a slight increase in response time. As a result, we believe our rover will have a net positive impact on everybody, but an especially positive impact on first responders.
Global	4	The rover is designed to be cheap and effective. However, as it requires a knowledge of computers/technology to operate, it will be less easily accepted/ used in third world countries. Additionally, the requirement to charge the rover may be prohibitive in these third-world countries. This means there is a positive effect in first world countries, and net-zero or slight negative for third-world, leading to a decision of a 4, for slight positive impact.
Cultural	4	As society becomes increasingly reliant on robotics to accomplish day-to-day tasks, there is often a concern of robotics taking jobs, at the cost of ease-of-life. Our robot shouldn't take any jobs, and if anything, will

Area of Impact	Impact	Description of Impact
		create new jobs for rover operators, in addition to other emergency response members. Additionally, the rover should create data/analytics on each encounter that can be reviewed/discussed after the fact to improve.
Societal	4	The product will hopefully change the mindset of first responders, from putting their lives on the line, to helping themselves in addition to others. The industry is known to be dangerous, but additional protective equipment may change the users' mindset in some ways, having less "heroes", and more knowledge, instead of having them dive into the unknown.
Environmental	2	The prototype, and final product will contain several motors, and a large battery. These may need to be replaced in time, and will need to be recycled properly. Depending on the battery type of our final product, (we are using lead acid for the prototype, which isn't exactly environmentally friendly) However, over the course of operation, there shouldn't be any major negative-effects of operation, so I'd give our environmental impact a net- 2.
Economic	5	The rover should be positive for the economy. For small cities/towns, they cannot afford expensive robotics that are already on the market. We would be creating a low-cost alternative that will force the market to be more competitive, and compete for more customers. This increased competition should be a net-positive for consumers, while also enabling those not previously in the market, to enter the market.

5 System Concept Development and Selection

One section in which alternative systems were generated was locomotion. We came up with three options for movement, each of which are shown below. The three concepts were corkscrews, treads, or simple wheels.

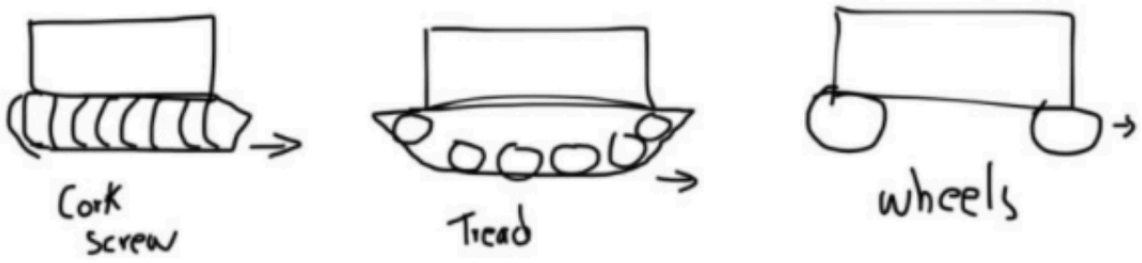


Figure 1, movement concepts

Table 6, concept selection matrix

Movement method	Corkscrew	Tread	Wheels
Ease of creation	3	2	1
Effectiveness at pitch	2	1	3
Speed	3	2	1
Simple gearing	3	1	1
Works in water?	1	3	3
Effectiveness in sand/rubble	1	2	3
Total	13	11	12

6 Subsystem Analysis and Design

Our project was divided into six subsystems: ergonomics, electronics, drivetrain/gearing, the frame, the black box, and communications. Each subsystem is explained in detail below, with many figures and supporting information.

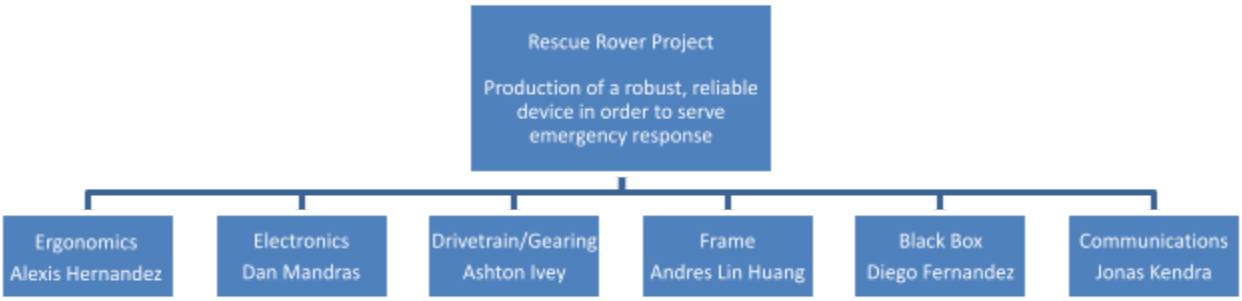


Figure 2, Hierarchical Subsystem Diagram

6.1 - Communications

The goal of the communication subsystem is to enable a user to operate the rover from a remote location, while gaining valuable insight into what is happening near the rover. This is done through the implementation of several environmental sensors, which record data on the rover. Simultaneously, the rover will send this information (video feed, live temperature readouts) to the user, who can interpret this information, and use it to make informed decisions about where to look next, and where to send their team. While the prototype implements temperature sensors and video feed, these sensors can be interchanged fairly easily to support the needs of any given customer. Theoretically, the user can implement anything from a humidity sensor, to an air quality sensor, and even potentially a geiger counter, with only minor changes to the base design.

The subsystem can be reduced to three, smaller projects. These include coding, sensor implementation, and hardware setup.

Hardware setup happened first. For the prototype, we used the Raspberry Pi 4b, with 8GB RAM, and 128 GB Micro SD, (Storage). These specs are significantly higher than what the customer would need on a real, commercial product, but the team utilized the Pi 4b, a device which it had at the time. A commercial product would still need a microcomputer, but could downgrade in the device's hardware, to 2-4GB ram, and as little as 8-16GB storage. However, if the user wished to save Video/data feed for each use, a larger storage may be necessary. This could be something that the user can change to meet their individual needs. While the Raspberry Pi with the higher specs costs roughly $120, a simpler version could cost around $50, leading to significant cost reduction.

After acquiring the hardware, it had to be set up for the project. Raspbian OS, a Linux-style system produced by Raspberry pi was the obvious choice, as it allows for flexibility on the team's end to change the system's boot settings as required. [2] Some notable changes that were required to get the project to work include enabling a steady-state IP, changing the systems GPIO (GPIO; General Purpose Input/Output), data-acquisition to work from a single pin, and enabling the pi to run headless; that is,

allowing the pi to graphically display the desktop without a monitor being attached. This last step was crucial to ensure that a remote computer could act as a controller, and “see” the information that was available on the pi, whilst inside the rover. Each of these steps were done in the device’s terminal, to prepare the device for use in the project.

The second of the three sub-subsystems was coding. This is something that can be easily uploaded to any subsequent device, and can be treated as a one-time, fixed cost in production. While it would be useful to continue to improve the code over time, and increase functionality, the base-model is fully usable.

Essentially, the code enables the pi to perform several operations at the same time, thanks to multithreading. Python was used for this project due to its simplicity of use, team knowledge of the language, and its ability to perform mathematical operations fairly quickly.

Multithreading was essential for the project due to limitations of the Raspberry Pi’s security. Live reading of key typing is difficult to implement because the system believes it to be a security threat. The only way to get live readouts is through making the user a “Sudo” (Superuser Do), which disables most device security. While this was an option, the team decided against it, as this makes the customer much more likely to destroy their computer. The team’s alternative solution was using a simple “input” command in python. While it is safer, and doesn’t require the use of “Sudo”, it comes at a cost in the efficiency of code. With the freeze on the given line, our solution was multithreading. The multithreading allows the program to switch which function is being executed, whenever there is a holdup; in this case, whenever the user isn’t actively providing an input, the temperature sensor code will run. The time lose through multithreading is small enough that it causes practically no change in response time of the rover, so this solution was accepted. [8]

The pseudocode for rover operation can be seen below, in figure (x)

The device starts by installing several different modules relevant to operation. This includes the time module, used to perform operations in time intervals, the GPIO module, to allow the rover to output power to different pins, the Threading and functools module to enable multithreading, and the CSV module to allow sensor data to be exported to a save file. The following operation is summarized in the flow chart, also shown below

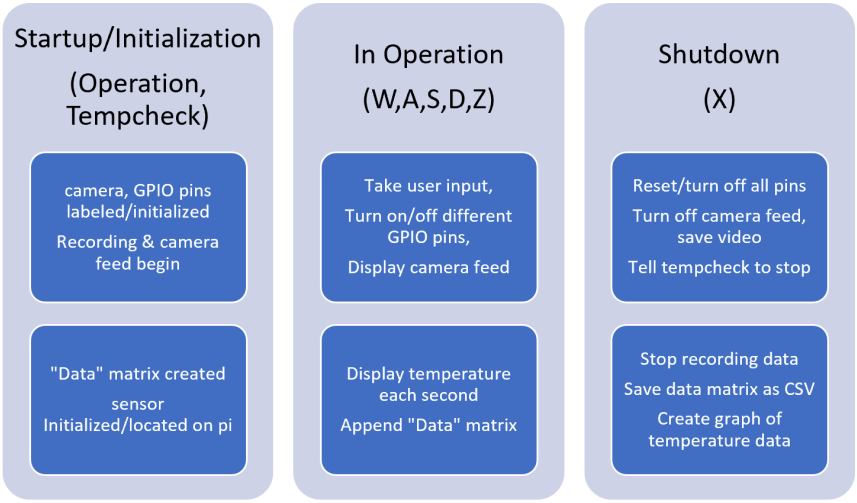


Figure 3, Code operation flow-chart

```
#######START OF PSEUDOCODE #######
Import threading>Thread
Import functools>partial
Import time>sleep
Import picamera
Import RP.GPIO>GPIO
Import gpiozero>CPUTemperature
Import os
Import glob
Import CSV
Import Numpy>np
Import matplotlib.pyplot>plt
##Set up sensor, reference files
Label camera
Initialize Data input as numpy array([[seconds,ambient
temp,CPU temp]])
Set GPIO mode, BCM
Setup GPIO for each motor
end=0 #Global variable to end separate functions
Def plot():
        “
        Plots data at end of operation
        “
        plt.plot(Seconds,AmbientTemp)
        Plt.label #label x,y, overall title

Def operate():
        “””
        This is the main command; initializes camera,
controls rover motors
        “””
        Start camera()
        i=0
        While i<1:
                x=input()
                If x=”w”
                        GPIO.turn_on(motor1(forward),motor2(forward))
                If x=”a”

                GPIO.turn_on(motor1(back),motor2(forward))
                ....#enable this for other possible
directions
                If x=”x”
                GPIO.cleanup()
                plot()
                Global end
                end=1 #this is used to control the
        tempcheck code
Def camerastart():
        print(“Starting feed”)
        camera.start_preview()
        camera.start_recording(“filename.h264”)

Def read_temp_raw():
        f=open(device_file, ‘r’)
        lines=f.readlines()
        f.close()
        return lines
Def read_temp():
        Try: #INCASE SENSOR FAILS
                lines=read_temp_raw
                While lines[0].strip()[-3:] != ‘YES’:
                        lines=read_temp_raw()
                Equals_pos - lines[1][equals_pos+2:]
                if equals_pos != -1:
                        Temp_string =
lines[1][equals_pos+2:]
                        Temp_c =
float(temp_string)/1000
        Except:
                temp_C=0
        Return temp_c
```

```
Def tempcheck(maxitter=10000):
        i=0
        cpu=CPUTemperature() #this is in celsius
regardless. User only needs to know it cant reach 100C
        Global end
        Global data
        While i<maxitter:
                ambient=read_temp*9/5_+32 #Converts
data to fahrenheit, optional for american
                If ambient==32:
                        print("ERROR; SENSOR
DISCONNECTED")
                Else:
                        print(i,"seconds","external
temp:", ambient, "F", CPU temp:", cpu, "C")

newdata=np.array([[i,ambient,cpu.temperature]])
                        data=np.append(data,newdat
                a,axis=0) #this updates the data each
                sec.
                If cpu.temperature>70:
                        print("CPU temperature
                critical!")
                If cpu.temperature>80:
                        print("SHUTTING DOWN")
                        np.savetxt(#SAVE FILE IN
                CSV NOTATION)
                        plot()
                sleep(5)
                os.system("sudo shutdown -h now")
                #This shuts down the pi, through
                terminal
        sleep(1)
        i+=1
        If end==1:
                np.savetxt(#Same format as earlier)
                i=maxitter
                plot()
Thread(target=partial (operate)).start()
Tempcheck() #these last two lines will run both 'operate' and
'tempcheck simultaneously, using //multithreading
##########END OF PSEUDOCODE ########
```

Figure 4, Pseudocode

The third sub-subsystem is the sensors. As mentioned earlier, there were changes that had to be made for the Pi operating system to allow the pi to recognize sensor inputs. Additionally, care must be used when interchanging the sensors, as the specific pins utilized must match what the operating system expects, as well as the rover control code. If done incorrectly, the entire code will fail, leading to a non-functioning rover. Luckily, the sensor utilized for our project, DS18B20 worked well because of its single-pin data transmission, and prewritten python modules and stackoverflow-style resources helped to interpret and express the data feed as it was generated. [1,3]

For customer use, switching out data sensors may require modification to the operating system, and code. To make the product commercially viable, the team would require at least one coder who works on making modifications to enable different sensors to be utilized. This would not be difficult to do, but is something that must be considered for a viable product.

Overall, the subsystem performs many processes, and acts as the brain, and nervous system for the rover. It enables a user to communicate with the rover, and the rover to communicate meaningful data to the user. To make the rover function even better, work can be done into more universal sensor implementation, and in creating a proprietary program to control the rover remotely, as opposed to the VNC (Virtual network connection) program currently used.

For the sake of the demo, Python code was run from Thonny, a processing software for python code. This was used for ease of writing, development, and easily

visible kernel outputs. Pictured below, in Figure 5, is the code whilst running. The camera feed is visible on the top right, and the live temperature readout is visible on the bottom left, in the shell. While we wouldn't want the actual code to be visible on the user end, this was satisfactory and usable for our prototype.

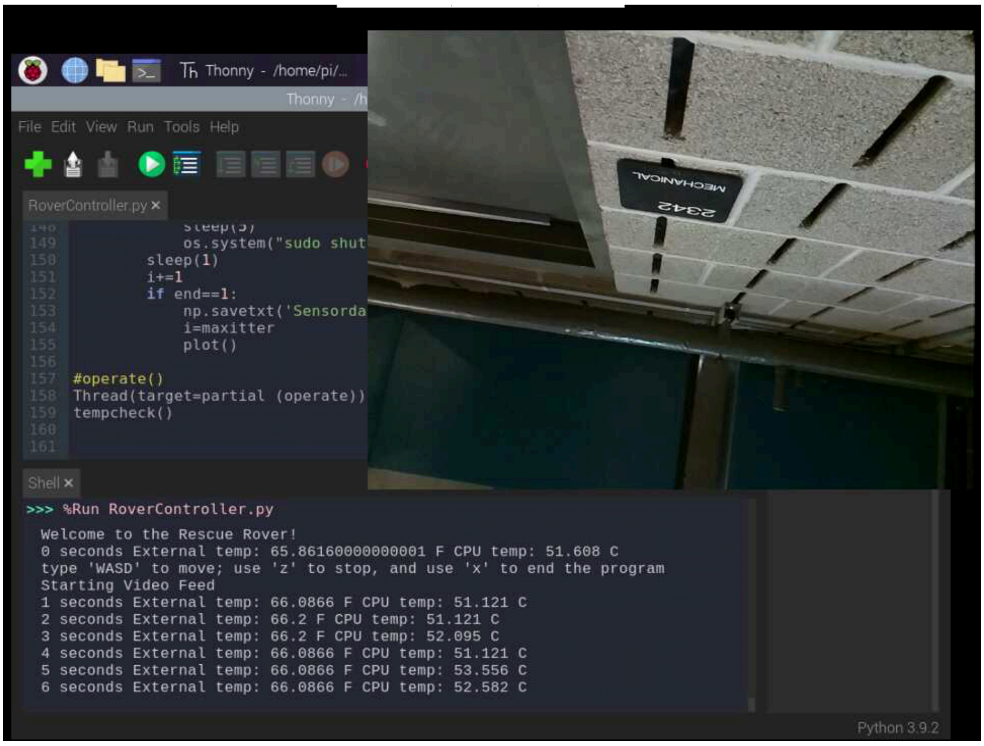


Figure 5, Python code on Raspberry Pi whilst in operation (Bottom left shell, Top right video)

After the code has finished operating, a graph will be generated, showing the user the temperature, plotted over time. After resolving some minor issues, with blank output, graphs like the one pictured below are automatically generated. [6] The data is also saved into a CSV format for ease of use/modification by the user afterwards. [5] This output can be changed for any relevant sensor, and the title/other relevant information can be used for any exploratory need. The plot below shows temperature readouts in the IED shop, for 20 seconds while being heated, and returning to ambient temperature over the next 30. It is worth noting that the change in temperature is not instant; this is simply a function of how reactive the sensor is. For more immediate readout, a more powerful sensor would be required. However, this isn't a major issue, as there is a clear spike whenever the change in temperature is rapid, as opposed to slow change with minor shifts. For the purpose of our rover, we would want to approach the true temperature rapidly, but the actual value isn't necessarily as important.

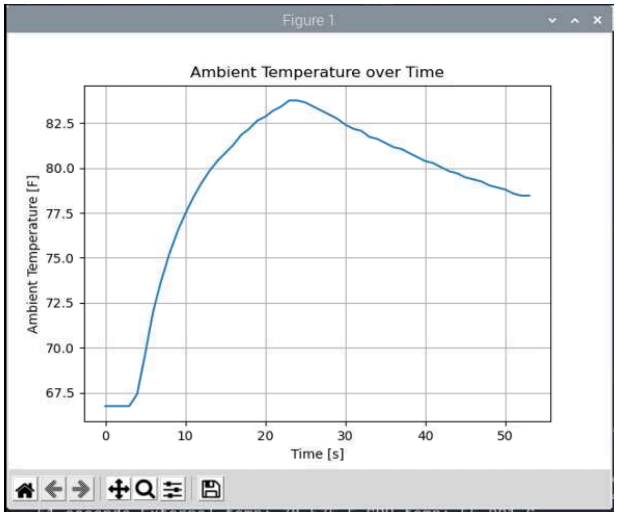


Figure 6, temperature over time plot

In terms of safety, there have been two major issues that the code combats. First of all, the rover is designed to operate, despite failure of some of its components. If the sensor is disconnected/fails in the middle of operation, the first version of the code simply errored out, leaving the user with a useless rover, sitting in a hostile environment. A simple fix to this was using a try/except statement to allow the rover to continue to operate after the component fails. [7] It will return a meaningful error statement to the user such that they will know where the error is occurring. The rover remains usable should this error occur. An example of this error in operation is shown below, in figure 7.

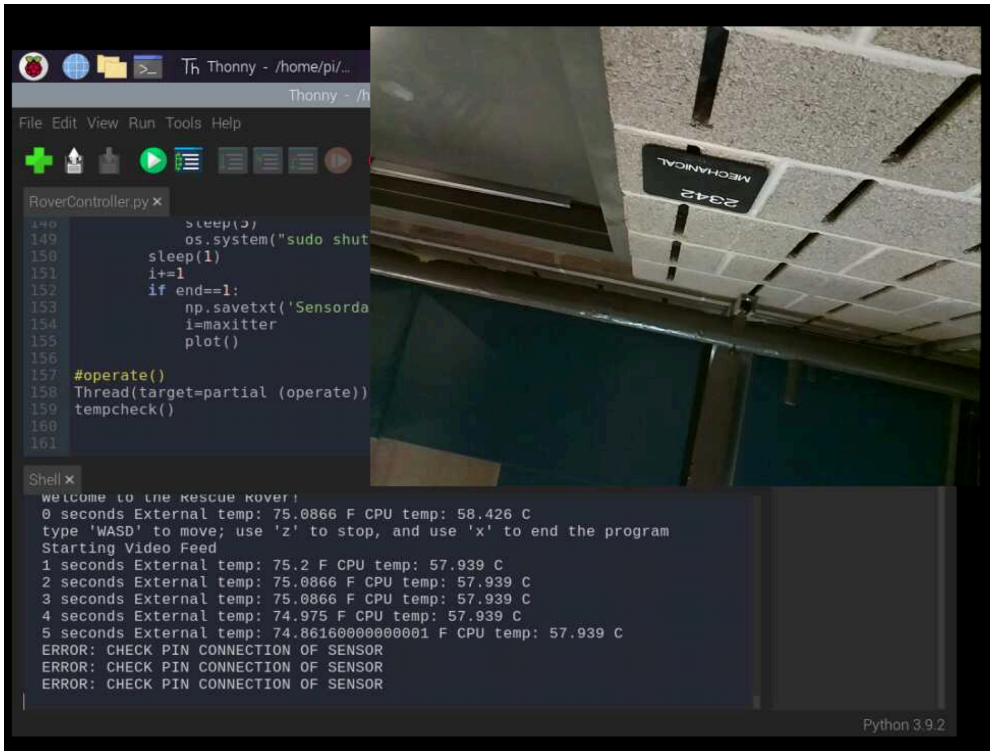


Figure 7, code in operation after sensor has been removed (Bottom Left)

A second implementation for safety of the rover is the ability for the internal computer to shut down when the environment is too harsh. In the case of this rover, we measured internal CPU temperature. For the safety and longevity of the rover, the raspberry pi inside cannot go over 100 celsius, lest the internal electronics fail. Thus, the internal temperature is monitored every second. Once it approaches a critical sensor, it will shut down to save the electronics for the next mission.

6.2 - Electronics

The main purpose of the electrical subsystem is to reliably provide power for the entire rover, including the Raspberry Pi, motor controller, motors, and any auxiliary sensors attached to the rover. This was accomplished through the use of a sealed lead-acid battery, which while rather heavy, provides an ample amount of current for an extended period of time, allowing for a longer runtime of the rover. In addition to the battery, this subsystem included a motor controller specifically designed for the Raspberry Pi, the Waveshare DC Motor Controller. This controller attaches directly to the GPIO pins of the Pi, allowing for a much more compact solution than previously envisioned. A charging circuit was not included in the prototype due to cost and space concerns, but could easily be implemented. Currently, to charge the battery, one must remove the battery and charge it on an external trickle charger.

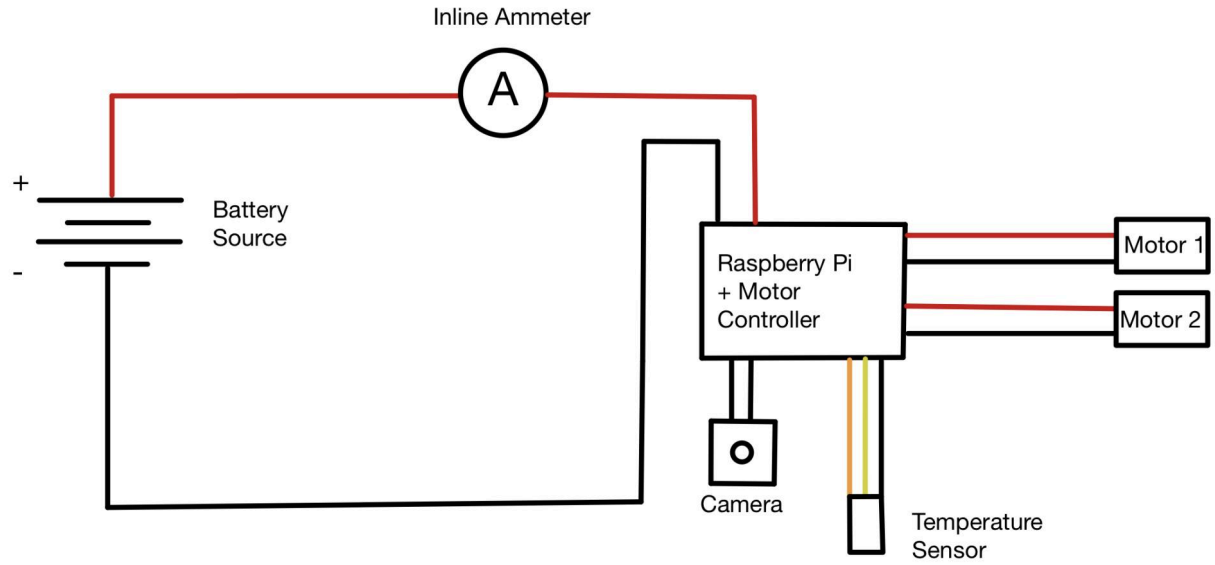


Figure 8, schematic of electrical subsystem

As shown in Figure 8 above, the circuit design of this rover is very straightforward and easy to implement or replace parts. There is a singular battery source, for testing purposes an inline ammeter to show current draw, and then the Raspberry Pi and motor controller. Connected to the motor controller are both motors, and then the temperature sensor through a pass-through on the controller to the Pi's GPIO pins. Lastly, the camera is connected to a dedicated header on the Pi, specifically for an external camera.

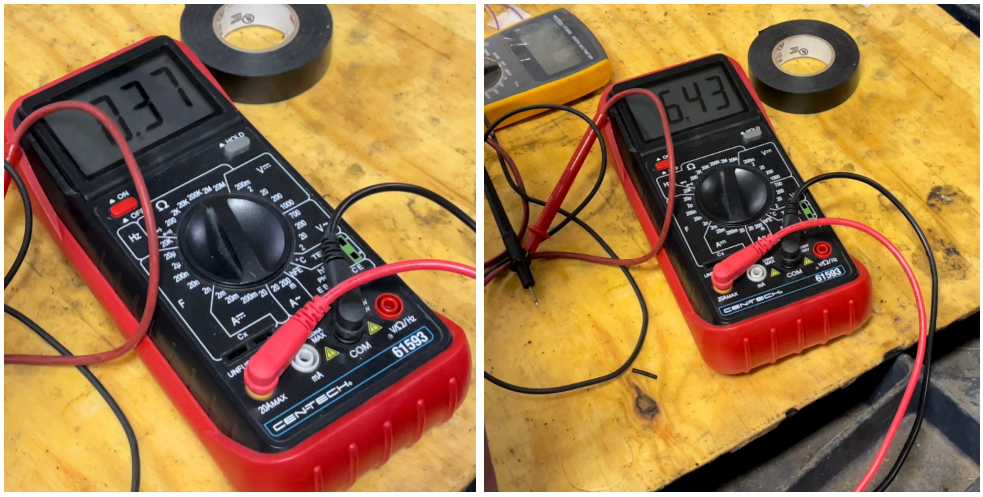


Figure 9, Instantaneous current readings.

For the motor control, there were many options available, ranging greatly in utility, ease of use and cost. The original plan was to use a bank of relays triggered by the Raspberry Pi in order to send the full voltage from the battery to each motor. In theory, this would have worked well, however, the Pi did not send enough voltage through its GPIO pins to trigger the relays, which needed 5 V, when the Pi can only send 16.3 mV. An alternate solution found was to purchase a premade motor controller specifically designed for the Pi, which would allow for a much simpler setup, and potentially even incremental speed control of each motor using PWM, instead of full on and full off. The first controller the team purchased was the “SB New Motorshield” off of Amazon. This controller utilized dual L293D H-Bridge controllers, each rated for 600mA sustained, and 1.2 A peak. At this point of the project, the team was not aware of the current draw of the motors, and was under the assumption it would be under 1.2 A per motor, which was the peak current that the controller could provide. Unfortunately, it was quickly discovered that each motor drew 2.6 A at full speed, and that it was necessary to run each motor at the full speed, due to them being geared for torque, not for maximum speed, as they were originally garage door motors. This controller also ran extremely hot, and with the microcontrollers facing the Pi’s processor, it resulted in unfavorable thermals, which led to it being returned. After that, another motor controller was found on Amazon, the “Waveshare DC Motor Controller”. This controller, running dual MC33886 H-Bridges, can provide 5 A per motor sustained, better thermals due to being a larger package and a simpler control scheme than the previous controller. This design allows for much more headroom, allowing for potentially much faster, power hungry motors to be installed, increasing the speed at which the rover can travel. During testing, this controller was found to drive the motors at a proper speed, and with upwards-facing microcontrollers, the thermals were much improved compared to the previous controller.

Up next was determining how to interface with the rover. Originally, the team discussed using an RC controller and remote to move the rover around, however that left us unable to view the video and data streams simultaneously. The next solution the team considered was running a low-end wireless router on the rover, and then creating a local subnetwork that a laptop would then connect to, allowing full remote access to the Raspberry Pi. The limitation with this was range and cost, as we were unable to find a suitable router that could host its own network for less than 50 dollars. After this, we went with our current solution, which for demonstration purposes was to use RPI's wireless network, rpi_wpa2. In a real-world solution, either a router or some form of close-range high-bandwidth network interface will be used to communicate between the Pi and the computer for the remote control.

Selecting the battery came after all other components of this subsystem were ready. There was a discussion amongst the team about which type of battery to use, ranging from a low-end car battery, a golf cart battery, and then various other types. Car and golf cart batteries ended up being much too expensive and heavy, resulting in the selection of a mid-sized sealed lead-acid battery (SLA), which had a high current and voltage output applicable for this rover, 12 Ah at 12 V. This battery offered an optimal tradeoff between weight and capacity, as it only weighs approximately 7 pounds, compared to car batteries, which can weigh up to ten times as much.

Wiring up the electrical subsystem was a simple task. The battery connects to the motor controller, which then distributes power to all of the other components, either through itself or the Raspberry Pi, in the case of the camera. The motors connect directly to the controller, with the red (positive) wires for each motor going to the odd-numbered terminals (M1 and M3) and the black (negative) wires going to the even-numbered terminals (M2 and M4). The temperature sensor connects to the pass-through GPIO pins, along with the cooling fan. Lastly, the battery hooks up to the two designated terminals for it, with a break in between the positive wire to allow the ammeter to plug in to show the instantaneous current draw. There is a power switch on the top of the motor controller which is used to turn the entire rover on and off, cutting current flow off from the battery completely.

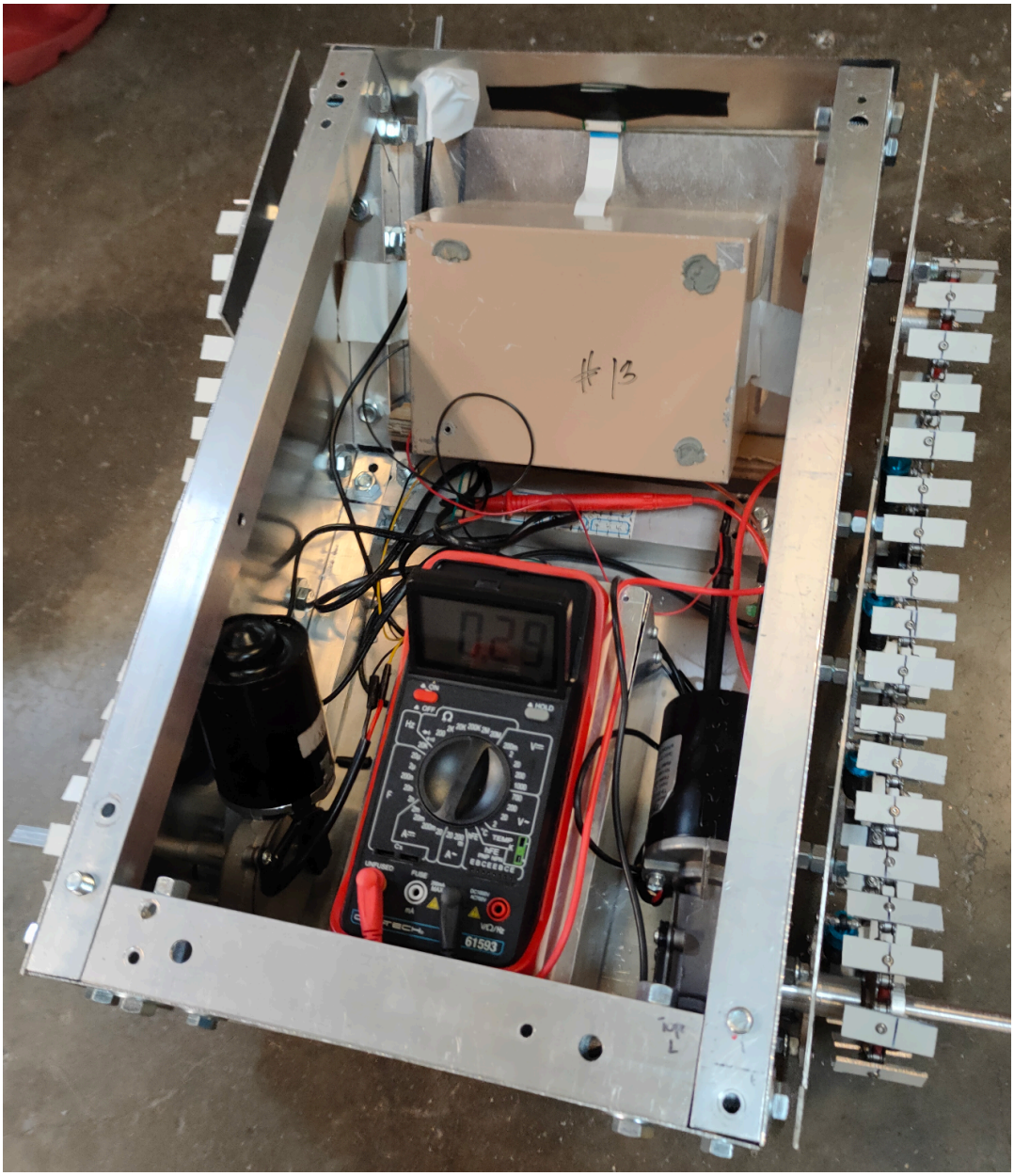


Figure 10, Assembled electrical subsystem.

In Figure 10 shown above, the completed rover can be seen, sans cover. Showcased in the center are the various components of the electrical subsystem. The motors are at the bottom left, bolted into the frame. In the center, below the multimeter, is the battery box, which encases the battery. The multimeter was installed for testing purposes, and would not be implemented in the final product. Above the multimeter / battery box, is the black box, which contains the Pi and motor controller. Above the black box is the camera, fixed to the rover with black electrical tape, and to the left of it the temperature sensor, attached with white duct tape.

6.3 - Black Box

6.3.1 - Design

The Black Box subsystem is responsible for protecting the valuable computer components of the rescue rover such as the Raspberry Pi. Should the Rover fail for any reason while inside a dangerous environment, the Black Box ensures that it will be salvageable and still operational. This is important so that data can be gathered and analyzed about its environment after failure. The video and temperature data recorded on the Raspberry Pi provides a picture for emergency responders and to analyze patterns for future situations. The box does not need to be easily accessible but it should allow for the Raspberry Pi to be inserted and removed as needed. It also should be able to withstand a significant impact from falling debris and such.

The customer requirements were gathered through various methods of research and also as a result of interviewing potential customers who are or were active members of EMS or Fire departments. After having spoken to John Sczesniak, a former volunteer Firefighter, a clearer picture was generated of the mechanical stress that the Black Box would need to be able to endure in a house fire environment. This was impactful in the design process and influential in the materials selection of the Black Box. Another potential customer who provided great insight was Nickolas Hernandez, who had experience as EMS and was a current firefighter in Naples, Fl. Nickolas provided invaluable information for the creation of the Black Box. Nickolas gave information regarding temperatures in which firefighters were allowed to venture into. This provided a basis for the temperatures in which the Black Box would be unable to be retrieved at and spend significant amounts of time in.

The data gathered was directly implemented into the Design of the Black Box. The extreme conditions it would need to undergo as well as the large amounts of force it would need to withstand led to the implementation of a multilayered design large enough to fit the Raspberry Pi only. The multilayered design leads to the Black Box having a large volume, however, this is appropriate for the conditions. The three layers which comprise the Black Box by design include a ¼ inch thick steel sheet as the innermost layer, a 1 inch thick sheet of Faced Insulation as the middle layer, and a 1/8th inch thick aluminum sheet as the exterior layer. The theoretical volume of the Black Box design is as follows:

2.5L, 3.8W, 1H (in Inches) 9.5 inches cubed

2.5x3.8 = 9.5 sq inches (2), 2.5x1 = 2.5 sq inches (2), 3.8x1 = 3.8 sq inches (2)
Total 25.3 sq inch

Initial layer of ¼ thick steel,

2.8L, 4.1 W, 1.25H (in Inches) 14.35 cubic inch

2.8x4.1 = 11.48 sq inch (2), 2.8x1.25 = 3.5 sq inch (2), 4.1x1.25 = 5.125 sq inch (2)

total= .4 square feet

40.21 sq inch

Middle layer of 1 inch thick insulation.

3.8L, 5.1W, 2.25H (in Inches) 43.6 cubic inches

3.8x5.1 = 19.38 sq inch (2), 3.8x2.25 = 8.6 sq inch (2), 5.1x2.25 =11.475 sq inch(2)

.64 square feet total

79 sq inch total

External layer of ⅛ inch thick Aluminum

Evident in the calculations above the Black Box is relatively large however, necessary for maximum protection of its components. Aluminum was chosen as the exterior layer for its lightweight property as well as its high Young's modulus for its density. Although compared to steel it has much higher thermal conductivity and lower Young's modulus, aluminum was the material chosen for the exterior layer because it is the largest layer, and choosing steel would have made the Box significantly heavier. Therefore, what aluminum lacks in thermal resistivity and strength, it makes up for in its lower density, making it most appropriate to be the largest layer in the Design. [25] An alternative considered was Beryllium which has a higher Young's Modulus, lower density, and lower thermal conductivity however it is much more expensive compared to aluminum.

The middle layer is essential for the thermal resistivity function of the Black Box design. The middle layer material does not need to be particularly rigid or have a high Young's Modulus, however, it is required to have a low thermal conductivity to best protect the inside components from overheating or even melting. Faced insulation was chosen for the middle layer solely for its low thermal conductivity value. It has a lower thermal conductivity than regular insulation as well as being much easier to install due to one side being faced. Insulation is also thick, the design accounts for a one inch thick layer of faced insulation which allows the thermal resistivity property of insulation to be most effective without jeopardizing the volume constraints placed upon the Design due to the frame. [24]

The innermost layer in the Design is most responsible for protection against serious impact while also needing to be resistant to thermal changes. For this reason steel was chosen due to its very high Young's Modulus and low thermal conductivity. An alternative considered for this layer was Tungsten, however, it is much more dense than steel which would jeopardize the weight requirement placed by the suspension. Steel is also much cheaper than Tungsten and is easier to manipulate without the ability to weld pieces together.

All three layers in the Design work in combination to protect the Black Box from serious impact and extreme heat. With no limitations the box would be welded together at the seams and would have a theoretical heat transfer equation which is evident in Figure 11.

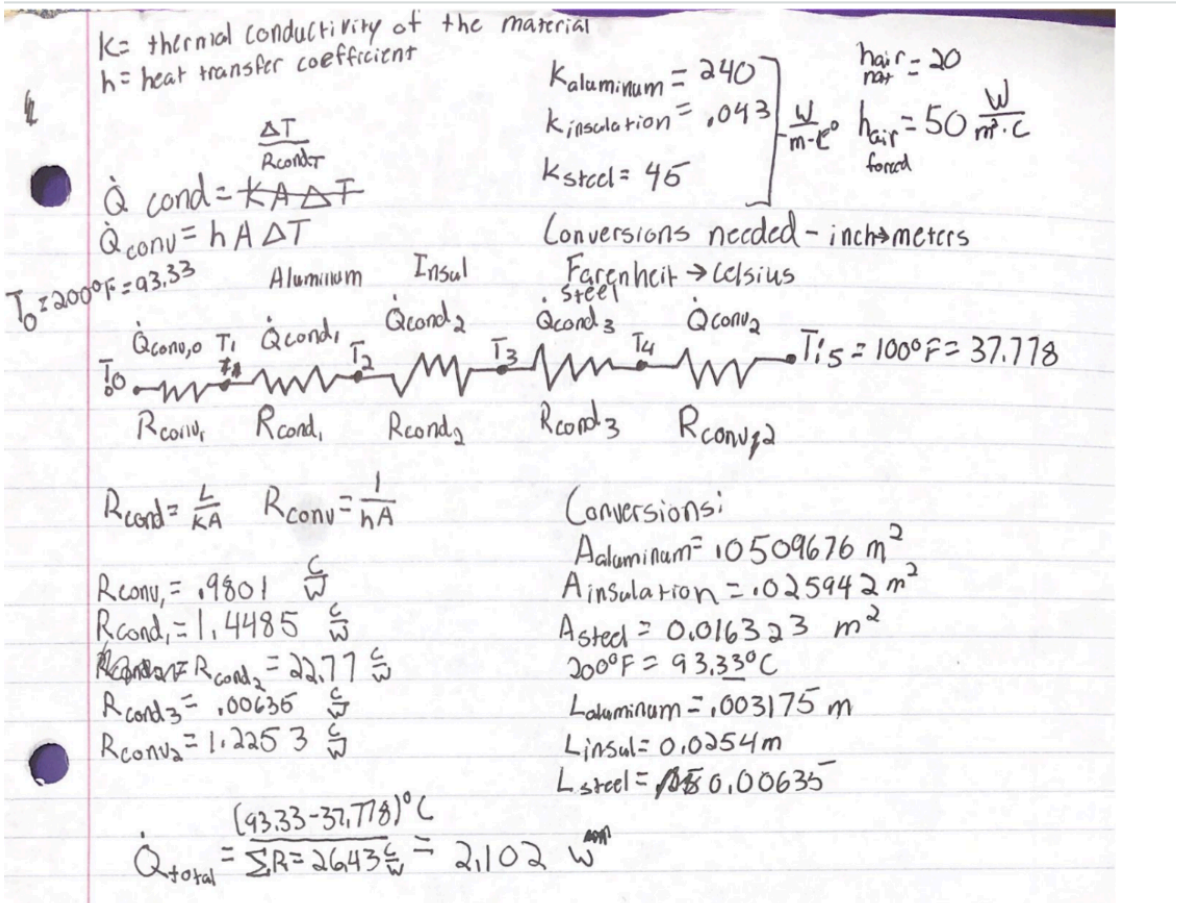


Figure 11, Mathematical simulation of heat transfer for the layers in the Black Box.

The mathematical simulation of heat transfer in Figure 11 is considering the heat entering is uniform and all design procedure has been followed exactly. The thermal conductivity values and heat transfer coefficients were gathered from online data. The heat transfer and heat resistivity equations are from knowledge gained in prior classes.

6.3.2 - Construction

Figure 11 depicts the final complete Black Box and all of its layers and attachments. Many limitations arose when constructing the box in reality due to having to use materials easily available to us. Various substitutes were made as a result of these changes, however the performance of the box was still more than adequate. The internal layer of steel was substituted with a pre-made aluminum box found in the IED shop. The aluminum box satisfied the structural necessities as the seams were folded into each other, providing stronger support and structural integrity. The increased thermal conductivity as a result of the aluminum box being substituted in for the less conductive steel, was counteracted with the addition of another layer at the base in which the Raspberry Pi would sit on to reduce the conduction on the side of the box directly touching the Pi. The wood layer would provide sufficient heat resistivity to replace what was lost in the substitution of steel and also provide an easy mechanism to strap down the Raspberry Pi providing protection in the event that the Box was to move violently. Rubber straps were attached to the wooden plank to secure the Raspberry Pi to the base of the Box as seen in Figure 11. [25]

The middle layer of the Box which was designed as insulated fiberglass was substituted for regular fiberglass insulation without the faced side. This made it harder to neatly wrap around the interior layer and also had a higher thermal conductivity. While the thermal conductivity was slightly higher as opposed to the faced insulation, the disparity was not significant enough to warrant a significant change in the Design, and therefore, was substituted without any further adjustment, evident in Figure 11 below.

The exterior layer was not substituted for anything and a sheet of aluminum was wrapped around the exterior. The sheet was cut into the calculated dimensions and was wrapped around the rest of the box. The box was sealed at the seams using Fireplace Caulk which was effective at holding the box together as well as resisting heat.

Figure 12, Depicts the fully constructed Black Box with the test subject prior to being placed in the oven at 200 degrees Fahrenheit. Notice the individual layers of aluminum, insulation, and another aluminum box as well as the corners sealed with fireplace caulk.

6.3.3 - Integration

Once the box was completed and passed its respective test, it was time to integrate it into the robot. This posed a new set of problems as the box needed to be significantly smaller. The adjustments made to the box were quite drastic in order for it to function properly within the frame of the robot as seen in Figure 13 below. The volume needed to be reduced drastically and needed to be done so that it could meet the customer specifications of maintaining a 100 degrees Fahrenheit internal temperature after 4 minutes in an external environment of 200 degrees Fahrenheit.

Figure 13, Depicts the process of adjusting the box to reduce volume as efficiently as possible

The exterior aluminum shell was removed and the middle layer of insulation was moved to the inside of the rover. The insulation was sealed to the interior of the shell with fireplace caulk and then another layer of reflective roll insulation for safety. A hinge was added as well to accommodate frequent changes and interactions with the Raspberry Pi. The final box integrated into the rover was still efficient enough to meet the customer specifications although it was much smaller.

6.4 - Drivetrain

The purpose of the drivetrain subsystem is to enable the rover to traverse over the ground so that it can navigate through its environment. This requires the drivetrain subsystem to convert the rotation of the motor output shafts into linear motion of the rover and support the rover as it moves over difficult terrain. Being that these two tasks are complex to execute, the drivetrain was divided into 5 primary sections and one minor section. The five primary sections of the drivetrain were the tracks, sprockets, guide wheels, shocks/springs and suspension linkages, and connections to the motors. The sixth minor section consists of the methods and

components used to integrate the primary parts together and then integrate the subsystem into the rover.

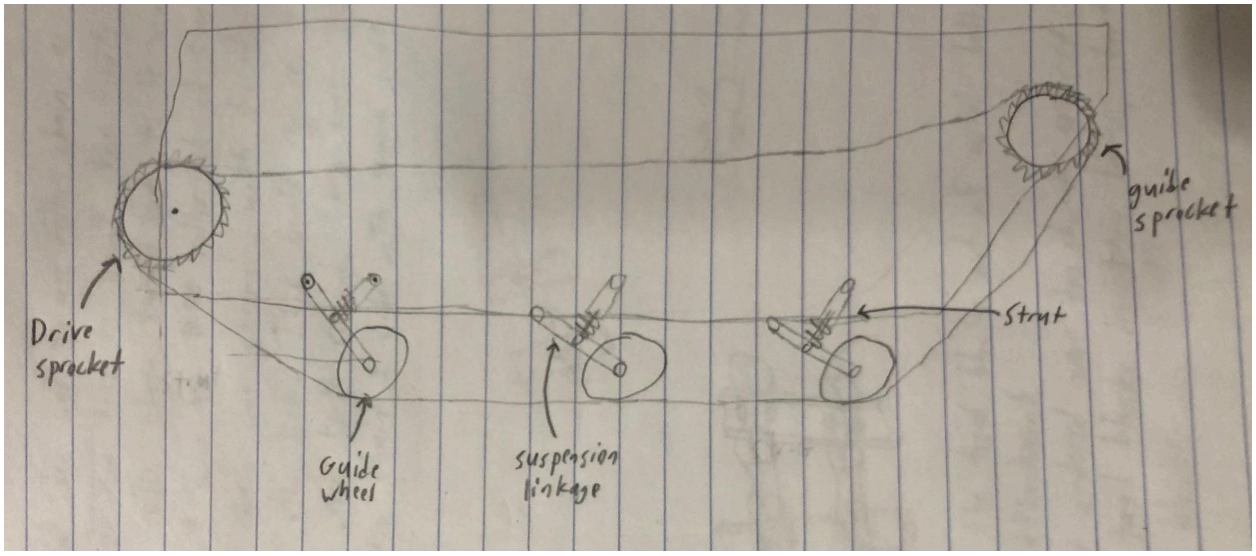


Figure 14, drivetrain subsystem concept sketch

The design for the drivetrain subsystem was developed using only one side of the rover to simplify the process, with the design of the entire assembly being replicated for the other side during actual construction. The remainder of the discussion of this subsystem, unless otherwise stated, will be described in terms of one side of the full assembly.

6.4.1 - Tracks

To start the design of the tracks, research was done to gather some ideas about what the general construction of the tracks would consist of. The easiest option would be to purchase tracks that were pre-assembled, but this proved to be too expensive to meet the cost requirements, both due to initial cost and having to purchase an entire set if one link broke[14]. The decision was made to construct the tracks rather than purchasing them. After some benchmarking previous designs and constructions of tracks similar in scale to the rover, a design utilizing a chain was used[15]. The initial plan was to use #40 attachment roller chain, as it already had holes and flanges to attach the tread blocks, but this was also too expensive[16]. After some more benchmarking for designs using regular bike chain, a preliminary design was chosen[17]. The design would use a single loop of chain with tread blocks that would be secured to every other link using a nut and bolt.

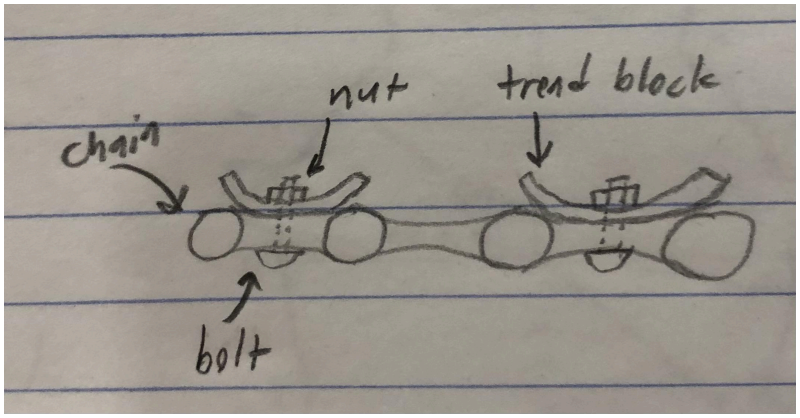


Figure 15, sketch of track design

Metal would be used for the tread blocks, as it would be more resistant to wear while in repeated contact with the ground as the rover moved

The tread blocks were constructed out of long sections of 2mm thick aluminum that were then cut into rectangles that were 40mm x 15mm. A ⅛” hole was drilled in the center of each tread block and an M3x14mm bolt and M3 nut were used to secure the tread block to the bike chain.

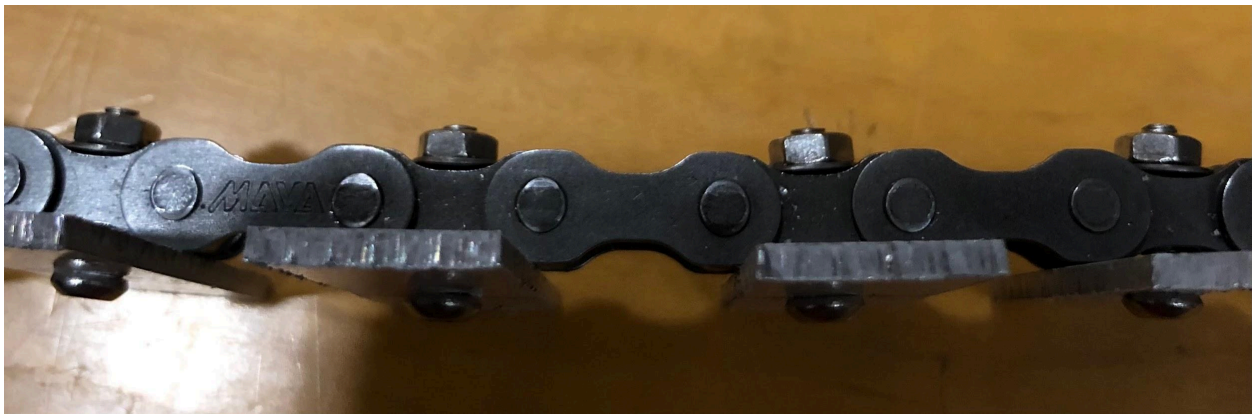


Figure 16, final assembled track side view

6.4.2 - Sprockets

Since bike chain was used for the tracks, the first idea was to purchase and use a sprocket from a bike, but this would create an issue with the tracks. The tread blocks are bolted onto every other chain link, so every other tooth of the sprocket would have to be removed to make room for the tread blocks. Because of this, it was decided to fabricate the sprockets instead of purchasing bike sprockets and making modifications. This would also allow the shape of the sprockets to be tailored to fit this specific application.

Two sprockets were needed, one drive sprocket and one guide sprocket (so a total of two of each would be needed for the full subsystem). The drive and guide sprockets would be identical except for the hole in the center for their respective shafts. Sketches were made to model the basic geometry of a sprocket

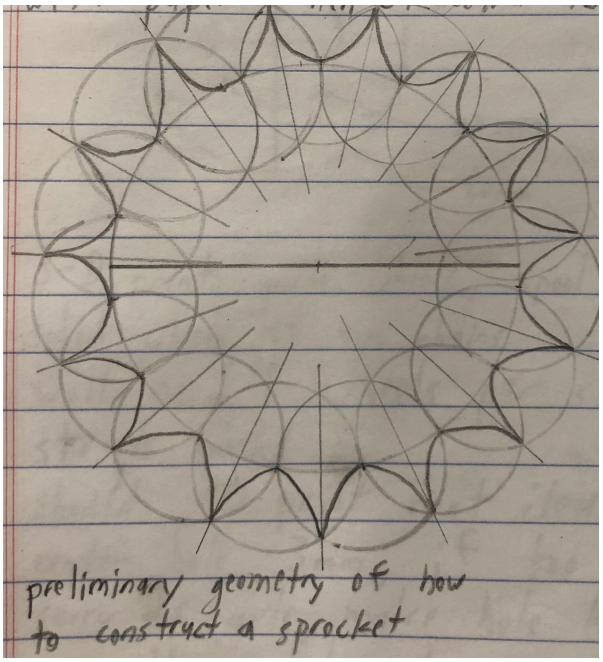


Figure 17, preliminary geometry to construct sprocket

and were then adapted to remove every other tooth and be the correct size to fit with the chain

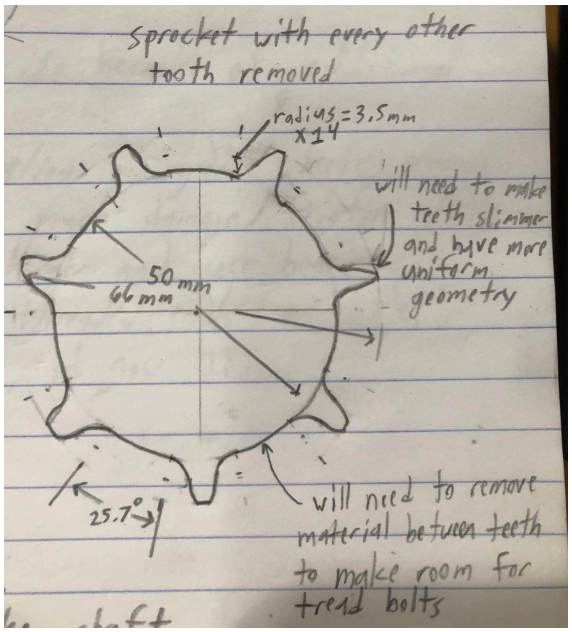


Figure 18, first sketch of sprocket design

This first to-scale sketch was traced out onto cardboard and cut out to determine the fitment with the chain. This sketch was then remade in Siemens NX and was lazer cut out of a 2.5mm sheet of acrylic. The acrylic sprocket was tested with the chain again and filed down to make a smooth fit. Cutouts were made in the sprocket to make room for the M3 nuts on the tracks so that the chain would sit flush against the sprocket. Once the cutouts were the correct size and

shape and the teeth had been filed down to allow the chain to fit around the sprocket, these changes were updated in the NX model. The model was then cut out of 2.5mm acrylic once more and was tested with the chain. Up until this point, all of the sprockets had not fit without some material being filed away, so the goal of this step was to model a sprocket that could be laser cut out of acrylic and would fit with the tracks without any modifications needed to the teeth. Laser cutting, test-fitting, and changes to the CAD drawing were repeated until the sprocket fit without any further changes necessary.

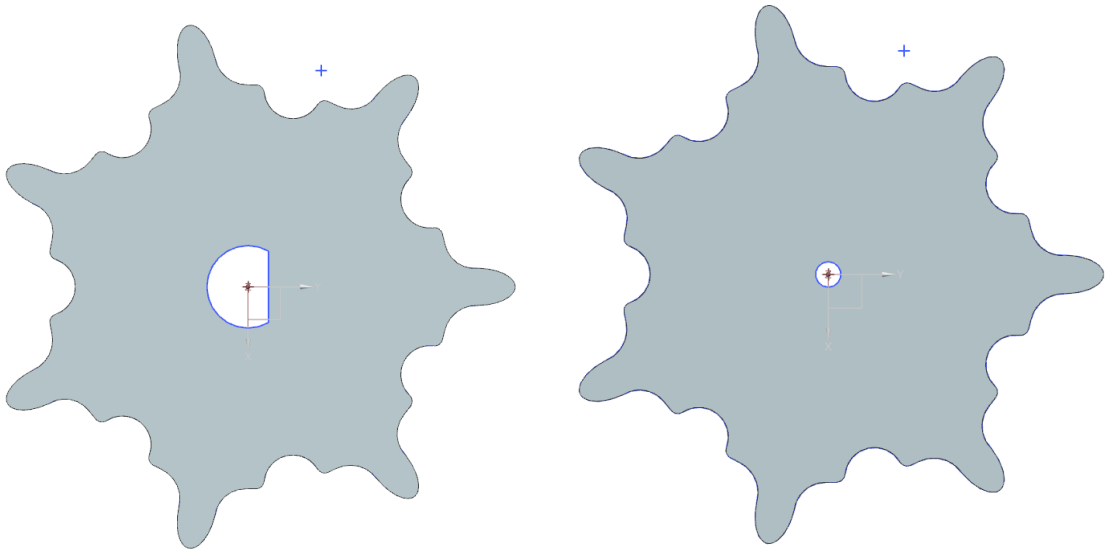


Figure 19, models used to make drive sprocket, left, and guide sprocket, right

A 4mm hole was added to the guide sprocket to allow for an axle to be utilized and a “D” shaped cutout was added to the drive sprocket to allow attachment to the driveshaft. These final designs were then laser cut out of 5mm thick acrylic. The only modifications that would be necessary would be to sand down the teeth of the sprockets, as the 5mm acrylic was too thick to fit in the gaps in the chain.

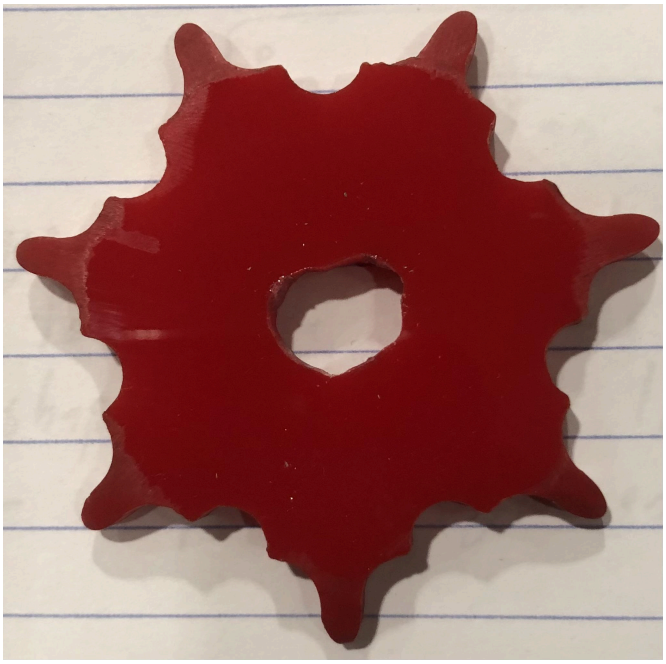


Figure 20, final drive sprocket

6.4.3 - Guide Wheels

The purpose of the guide wheels is to keep the tracks in line and supported in the space between the drive and guide sprockets. Four guide wheels would be used and their design was kept simple. The wheels would be similar in shape to a spool, being a cylinder with a channel made into the middle, as shown below

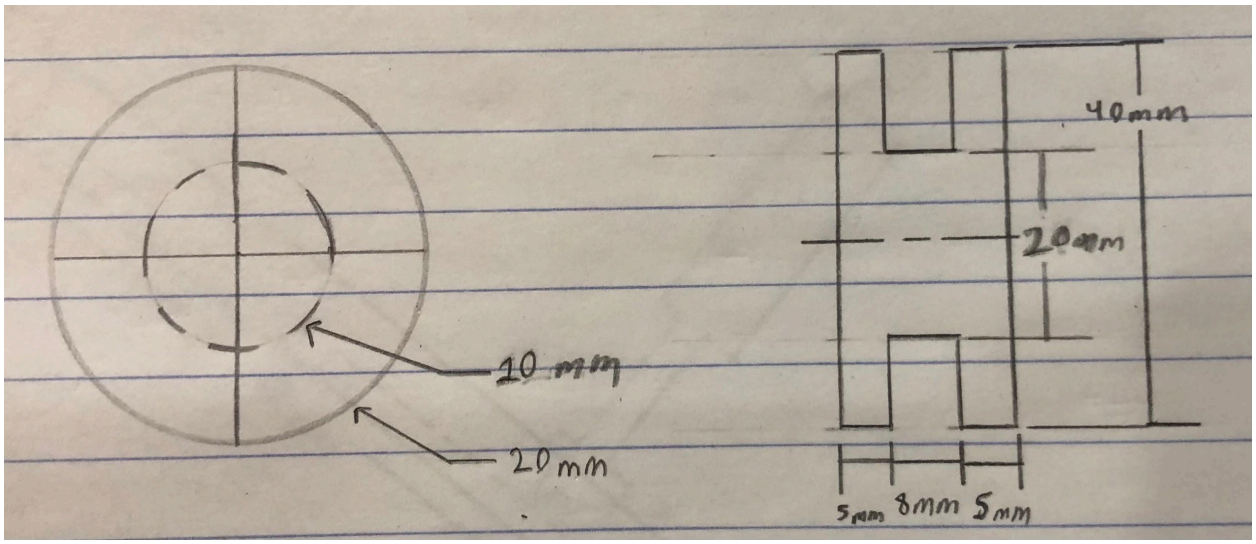


Figure 21, first sketch of guide wheels

The chain would ride inside the channel and the larger diameter edges of the wheel would roll across the inside of the tread blocks as the rover moved. A 4mm hole was added in the final geometry to allow an axle to run through each wheel.

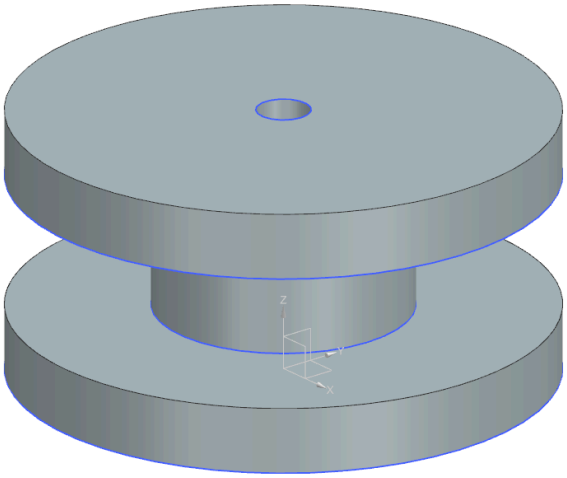


Figure 22, model of guide wheel

The wheels were 3-D printed out of PLA, starting with one wheel to verify that it would fit with the tracks, followed by the remaining wheels. After the first wheel was printed and tested, the channel needed to be widened from 8mm to 10mm, and this was the final dimension that was used.

6.4.4 - Shocks/Springs and Suspension Linkages

The purpose of the shocks and springs is to support the weight of the rover and provide the means for the tracks to conform to the surface that the rover is traversing over. Manufacturing springs and struts was decided to be too complex, considering the relatively low cost of purchasing RC car springs and struts[18]. While benchmarking for existing designs for the treads, information about existing suspension setups was also obtained[19]. A preliminary design was made that included a c-channel bar with the strut mounting inside of the channel

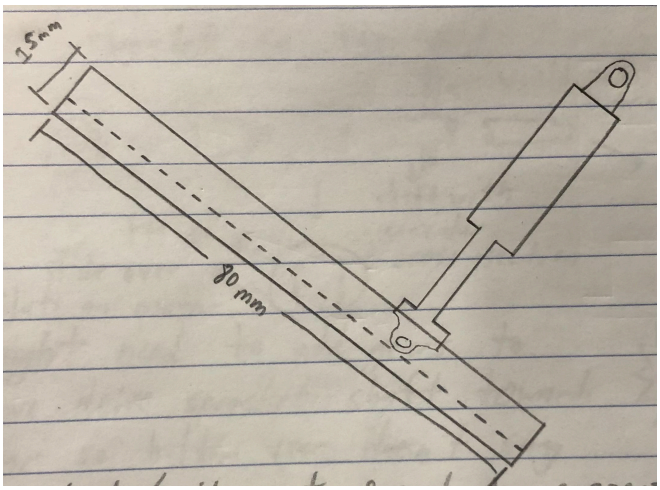


Figure 23, first strut and linkage sketch

The linkages would be 3-D printed out of PLA and the struts would be attached using M3 screws and nuts. The design would include 3 of these strut-linkage assemblies per side.

However, after the first set of struts arrived damaged, and the actual stiffness of the springs could be felt, it was decided to instead use 4 struts and linkages per side, and use larger struts and springs, to ensure that the weight of the rover could be comfortably supported. In addition, the design of the linkages was simplified to use a flat plate and spacers instead of a c-channel.

Figure 24, suspension linkage

A cutout the same size and shape of the side of the rover frame and 4 suspension linkages were cut out of polystyrene in order to test the setup for the suspension on the side of the rover. A 1 liter bottle of water was then used as a weight to test if the struts would be able to support the projected 30lb weight of the rover. During the test, each strut compressed 5mm and a ratio comparing the test weight and spring compression to the total weight and spring compression confirmed that struts would be able to support the proposed 30lb weight of the rover.

$$\frac{W}{D} = \frac{w}{d} \rightarrow \frac{W}{0.035m} = \frac{1kg}{0.005m} \rightarrow W = 7kg \approx 15.4lb * 2 = 30.8lb$$

Where:

W = total weight
w = test weight
D = total allowable spring compression
d = test spring compression

The new linkages were then laser cut out of 2.5mm thick acrylic and were attached, along with the struts, to the final mounting solution for all of the components of the subsystem. Once it was verified that the linkages were the correct geometry to be used in the subsystem, it was noticed that the linkages had to be attached somewhat loosely to their mounting plates to allow them to articulate freely. This meant that they were bending when more weight was applied to the subsystem. The decision was made to fabricate a new set of linkages out of ⅛” thick aluminum to improve rigidity of the linkages

Figure 25, final suspension linkage

and these aluminum linkages were used for the final design.

6.4.5 - Connections to the Motors

The final major section of this subsystem is the connection from the output shaft of the motor to the input hole on the drive sprocket. The motor shaft consisted of circular section with two flats on opposite sides with a threaded section at the end

Figure 26, motor output shaft

A preliminary sketch was made for the general driveshaft geometry, showing how it would interact with the motor output shaft, side of the rover, and drive sprocket.

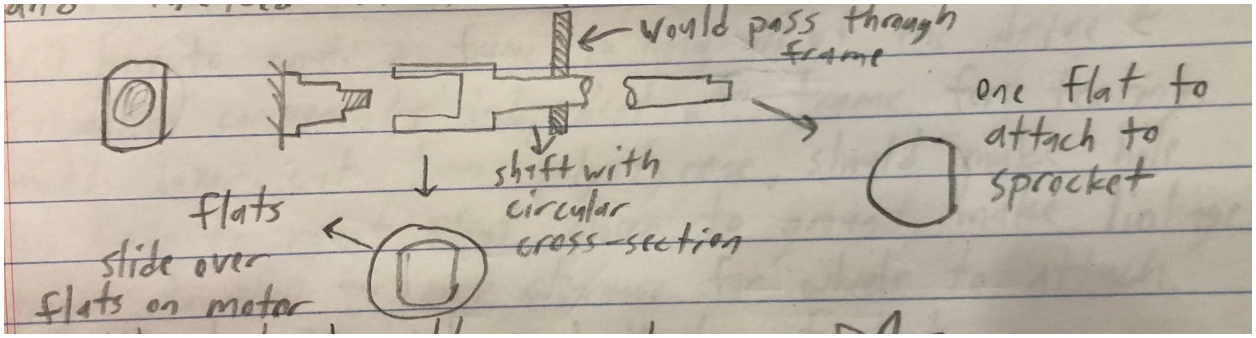


Figure 27, driveshaft concept sketch

The final design was similar to the sketch, with a few changes and additions being made. On the motor end of the driveshaft, the hole in the final design was circular with a hole drilled and tapped on the side for a set screw. The set screw would tighten against one of the flat surfaces on the motor shaft to lock the driveshaft in place. The drive sprocket end was also changed to have a “D” shaped cross section for its whole length to allow for the position of the drive sprocket to be adjusted.

Figure 28, final driveshaft

The driveshaft was milled out of a solid rod of aluminum and no adjustments or modifications were made for the final product.

6.4.6 - Integration of Components

In order to allow the drivetrain subsystem and frame subsystem to be worked on independently, the drivetrain components were mounted onto a separate plate which was then attached to the rest of the rover. The mounting plate was made out of a sheet of aluminum and largely mimics the shape of the side profile of the frame, aside from unneeded material being removed from the top. A basic model was made in NX to get an idea as to how the wheels, struts, and linkages would be attached.

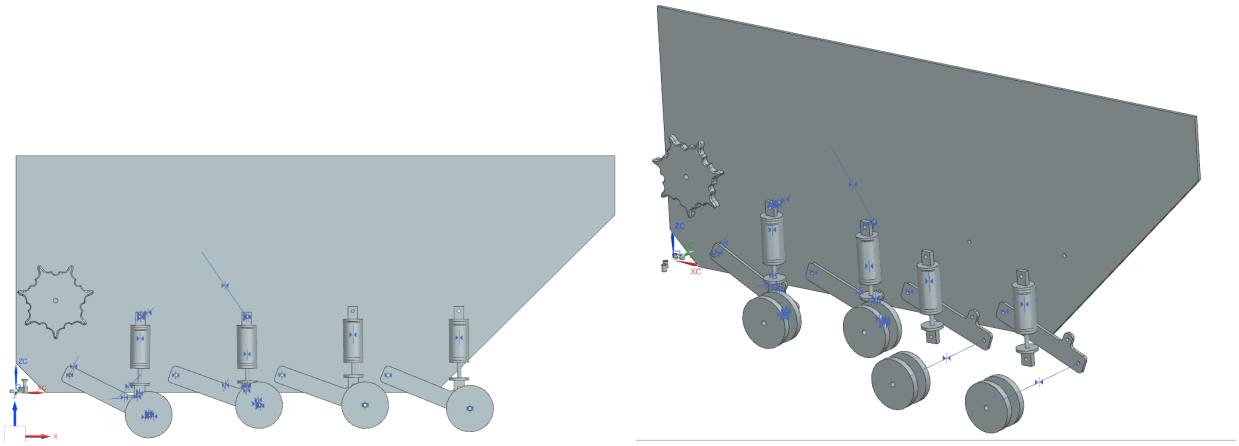


Figure 29, model of component integration

The wheels, struts, and linkages were attached to the mounting plate using M3 screws and nuts. The axles for the wheels and guide sprocket were made using a ⅛” steel rod that had M3x0.5 threads cut into each end and lock nuts securing each end. Spacers were made from the same aluminum stock as the driveshafts and had a ⅛” hole in the center to allow for the axle to pass through. A locking collar was made using the aluminum stock to hold the drive sprocket onto the driveshaft. Its design is relatively simple, consisting of a 10mm hole to slide onto the driveshaft and a hole drilled and tapped into the side to allow for a set screw to hold the collar in place.

Figure 30, locking collar

Holes were drilled into the mounting plate so that the corresponding screws on the frame could be used to secure the mounting plate to the rover.

6.5 - Frame

It is the compartment containing all the internal electronics and serves as the framework of what the rover will become. There are two main considerations that relate directly to its design which are the space required and the protection required.

Since the rover will be deployed into dangerous places, it must be able to withstand bumps or shocks and extreme temperatures. The frame should be able to protect the internal components reducing their chances of being damaged. In addition, the frame should have enough space so that the components can fit inside. [20]

These two factors, the protection and the space required contribute to the weight of the frame and the goal is to come up with an efficient design that can satisfy the consideration of minimizing the weight.

6.5.1 - Design Process

The design is based on a box where the components are stored, from this, it is modified to meet the requirements.

Dimensions were determined to be 20 in x 12 in x 8 in (length, wish and height respectfully). These numbers were considered according to the estimated size of the internal components based on their location and the estimated size of the drivetrain. The objective of the frame is that it adapts to the dimensions of the components and the drivetrain adapts to the dimensions of the frame, but due to manufacturing limitations, the frame also has to adapt to the drivetrain.

As shown in *Figure 31*, the first sketch, the basic concept of the frame stores the main internal components that are: the motor, the battery, the black box, the sensor, and the camera. It was not yet clear how the drivetrain was going to be, later it was determined that it would be attached to the sides of the frame.

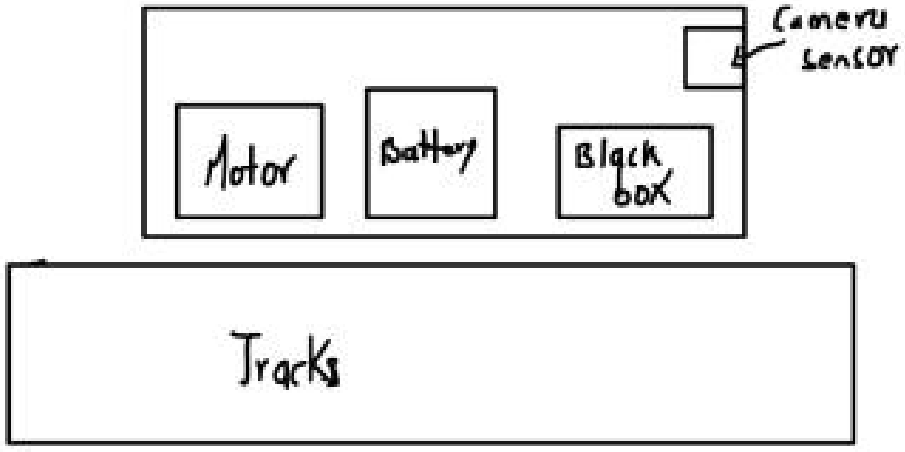


Figure 31, Sketch concept of the frame

One of the goals of the rover was to be able to climb stairs. For this reason, an angular cut was added to the front and a smaller one to the back as can be seen in

Figure 32. Due to limitations and manufacturing difficulties, this objective became secondary. Yet these changes persisted in the design.

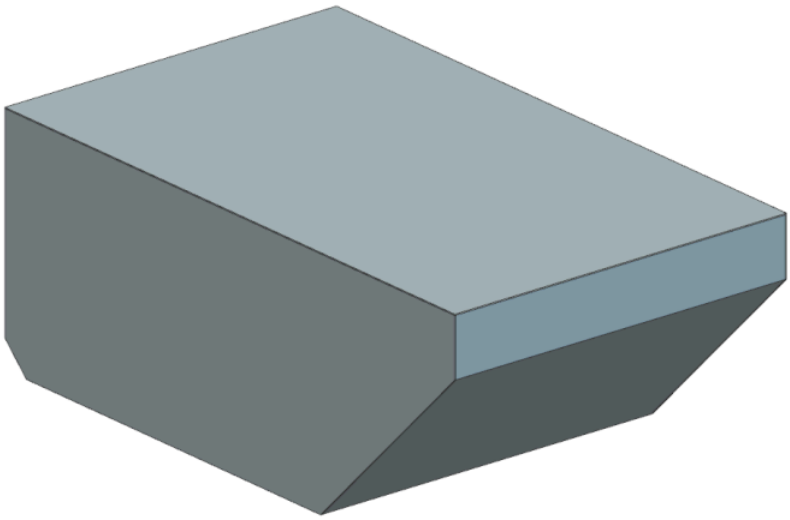


Figure 32, Faces of the rover, trimetric view

To reinforce the resistance of the rover and increase stability, metal bars were added to the edges of the frame as if it were its skeleton. As shown in *Figure 33* and *Figure 34*, the bars will help withstand impacts and provide additional stability.

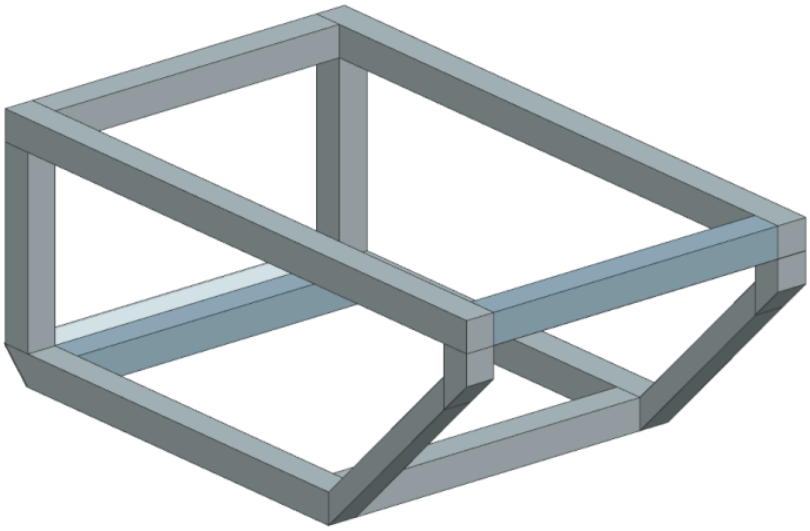


Figure 33, Skeleton of the rover, trimetric view

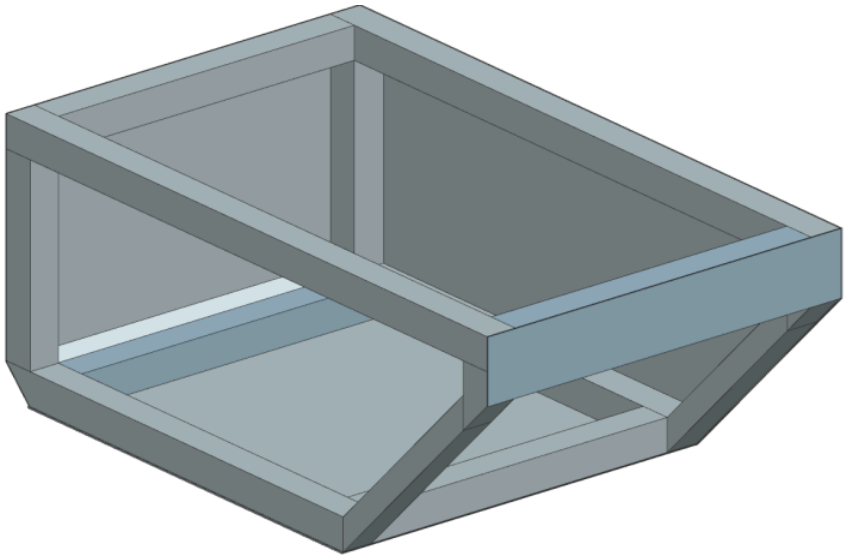


Figure 34, skeleton integrated with the faces, trimetric view (some faces were hided)

Holes were drilled on the sides and front of the frame to facilitate the integration of other subsystems such as the drivetrain with the motors and the camera and sensors.
Figure 35 shows the holes drilled in the side that will serve to attach the drivetrain.
Figure 36 shows the hole drilled in the front used for the camera

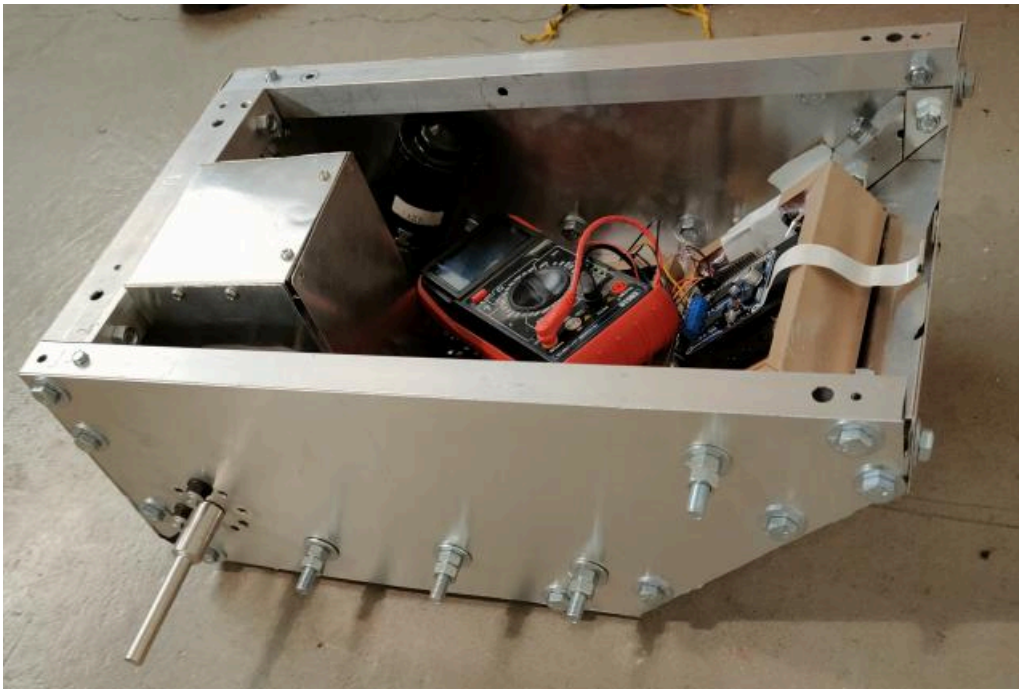


Figure 35, Frame with holes drilled on the side

Figure 36, Frame with holes drilled on the front

Figure 37 shows that the frame successfully was able to house all the components of the rover enabling its integration. Note that the ideal design would have additional reinforcements inside and it would be painted with thermal paint to be resistant to extreme temperatures.

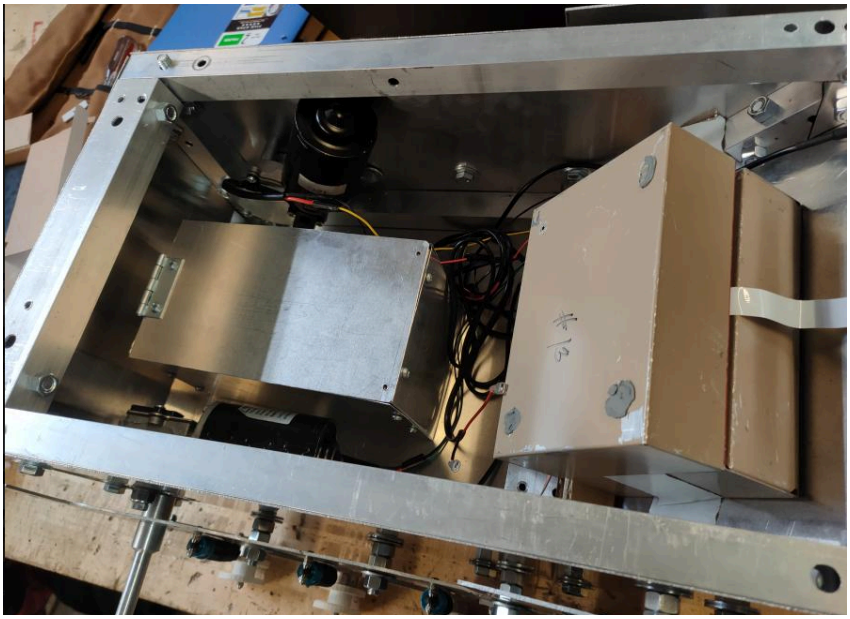
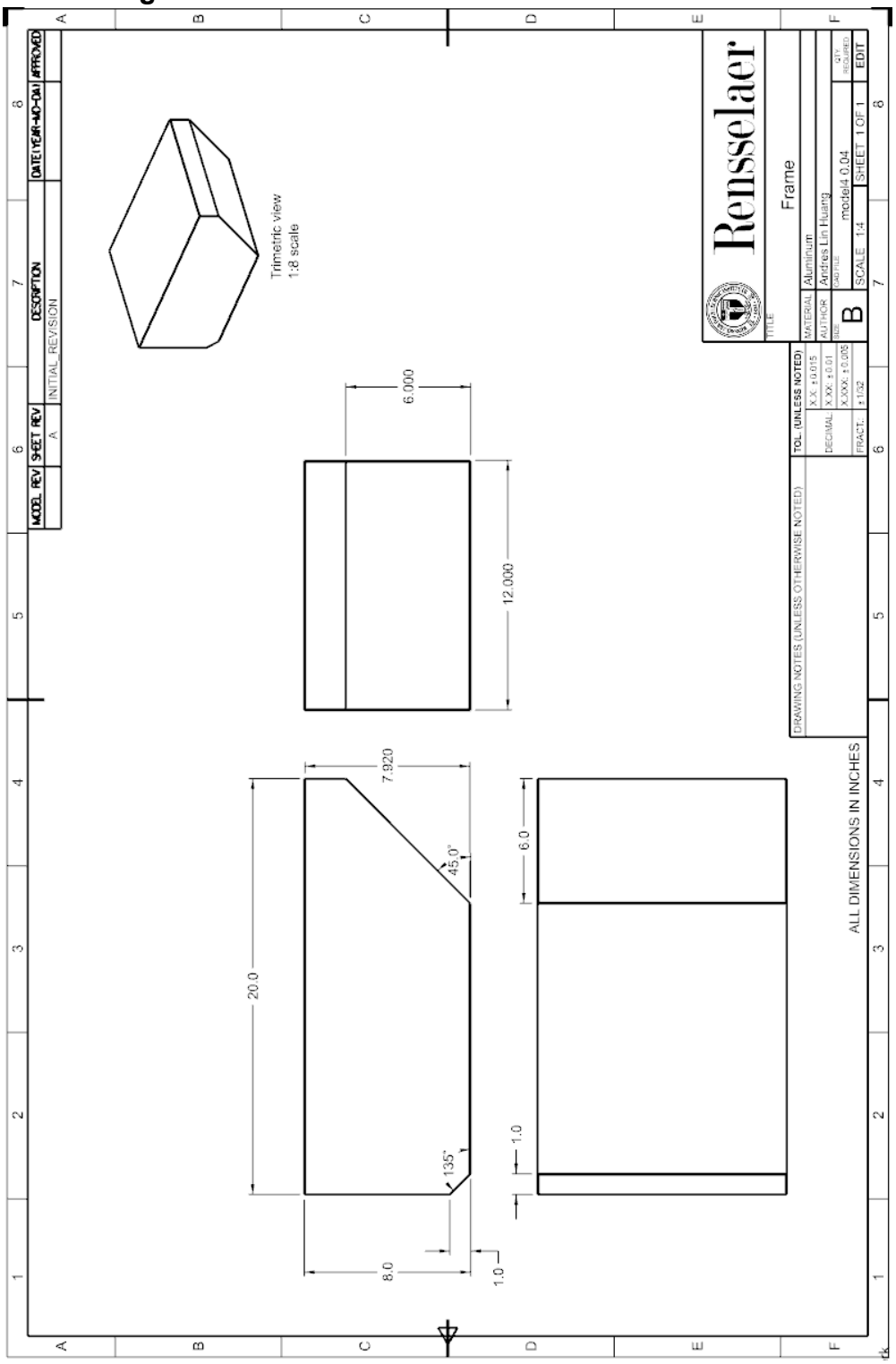
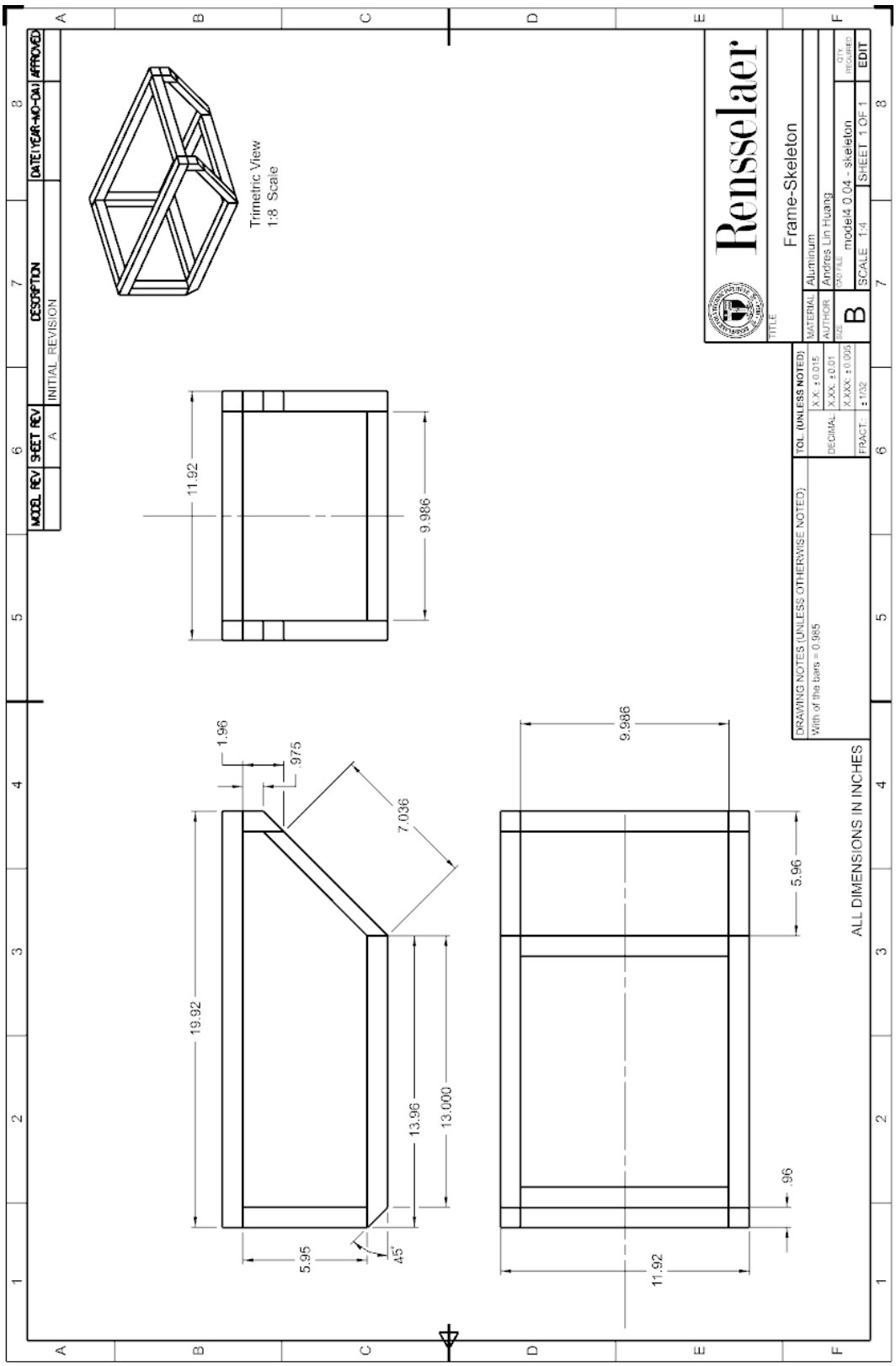


Figure 37, Integration of the frame with the other subsystems

6.5.2 Detail drawings of the frame



Detail drawing of the faces of the frame



Detail drawing of the skeleton of the frame

6.5.3 - Challenges

Precision of cuts and holes

Ideally the rover faces should be cut with high precision machines such as a laser cutter or a water jet cutter. Instead, a band saw was used which limited the accuracy of the cuts whose dimensions were manually measured.

The FIgure 38, shows the cut faces and although it is hard to see with naked eye, the dimensions and angles are not perfect.

Figure 38, Faces of the frame cut

Availability of materials

Ideally the frame would be built from strong, lightweight materials such as an alloy of aluminum and magnesium. This material, in addition to being light and strong, is resistant to corrosion. Due to budget constraints, the materials were gathered from scraps of different aluminum sheets, and the aluminum bars, also from scraps, were already being drilled. Due to these discrepancies, modifications of the main design were made to accommodate the materials available.

The figure 38 shows the bars already cut that were used. It can be seen that holes were drilled already and the width of the bars are not the same.

Figure 39, Bars cut

Inability to weld metal parts

The initial plan was to weld the aluminum parts, however this was not possible since it was not possible to weld in the IED shop and the welding kit in the engineering processes shop is not for aluminum. As an alternative to joining the metal parts, bolts and nuts were used. This was a big problem because the design had to be altered in order to make the holes. In addition, this represented extra time and costs for the project.

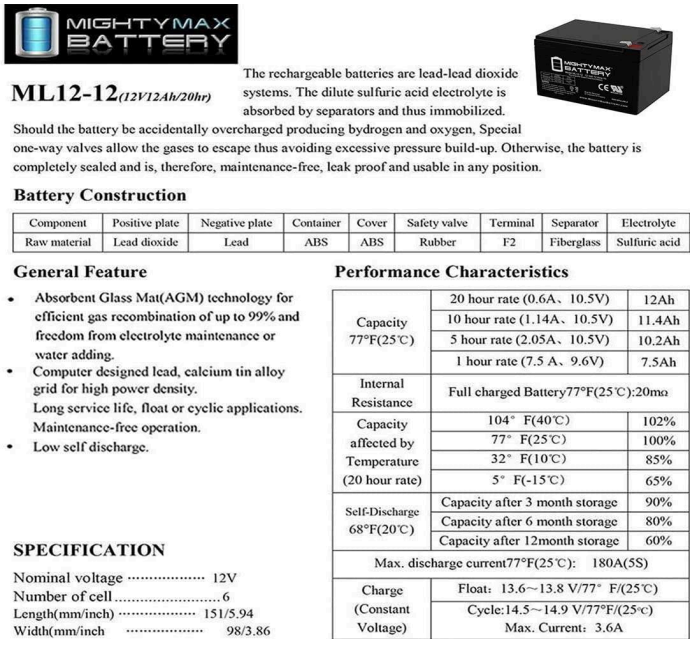
The figure 40 shows the frame being assembled using nuts, bolts and flat washers.

Figure 40, Frame with bolts and nuts being used

6.6 Ergonomics:

The main role of the ergonomics subsystem is to ensure that user's interactions are safe and efficient. This is accomplished by having a battery protection box and making sure the rover itself could be used with the utmost safety. This would improve the user's interface by increasing their comfort when working with the product and maximum safety.

The first component of the ergonomics subsystem is the battery protection box. The box is mainly intended to protect the battery from the excessive heat of a fire and to protect the user if there is a mishap with the battery. The box must also be easily accessible and the user should be able to open with ease, having zero concerns about a safety issue occurring when handling. While brainstorming the design, some specifications to consider is that sealed lead-acid batteries have a rated capacity based on the best operating temperature of 68-77°F and for the battery to operate over the range the expected service life will reduce by 50% for every 10°F constant increase above the recommended temperature. The battery may be discharged over a wide temperature range of -40°F to 140°F. Another piece of knowledge to consider is that lead-acid batteries release hydrogen gas that can be potentially explosive. The box then must be adequately ventilated to prohibit the build-up of hydrogen gas. In addition a user should have easy accessibility to the battery if it needs to be charged or if required to change the old battery for a new one.



MIGHTY MAX BATTERY

ML12-12 *(12V12Ah/20hr)*

The rechargeable batteries are lead-lead dioxide systems. The dilute sulfuric acid electrolyte is absorbed by separators and thus immobilized. Should the battery be accidentally overcharged producing bydrogen and oxygen, Special one-way valves allow the gases to escape thus avoiding excessive pressure build-up. Otherwise, the battery is completely sealed and is, therefore, maintenance-free, leak proof and usable in any position.

Battery Construction

Component	Positive plate	Negative plate	Container	Cover	Safety valve	Terminal	Separator	Electrolyte
Raw material	Lead dioxide	Lead	ABS	ABS	Rubber	F2	Fiberglass	Sulfuric acid

General Feature

- Absorbent Glass Mat(AGM) technology for efficient gas recombination of up to 99% and freedom from electrolyte maintenance or water adding.
- Computer designed lead, calcium tin alloy grid for high power density.
  Long service life, float or cyclic applications.
  Maintenance-free operation.
- Low self discharge.

Performance Characteristics

Capacity 77°F(25℃)	20 hour rate (0.6A、10.5V)	12Ah
	10 hour rate (1.14A、10.5V)	11.4Ah
	5 hour rate (2.05A、10.5V)	10.2Ah
	1 hour rate (7.5 A、9.6V)	7.5Ah
Internal Resistance	Full charged Battery77°F(25℃):20mΩ	
Capacity affected by Temperature (20 hour rate)	104° F(40℃)	102%
	77° F(25℃)	100%
	32° F(10℃)	85%
	5° F(-15℃)	65%
Self-Discharge 68°F(20℃)	Capacity after 3 month storage	90%
	Capacity after 6 month storage	80%
	Capacity after 12month storage	60%
Max. discharge current77°F(25℃): 180A(5S)		
Charge (Constant Voltage)	Float: 13.6~13.8 V/77° F/(25℃)	
	Cycle:14.5~14.9 V/77°F/(25℃) Max. Current: 3.6A	

SPECIFICATION

Nominal voltage ····· 12V
Number of cell ····· 6
Length(mm/inch) ····· 151/5.94
Width(mm/inch ····· 98/3.86

Figure 41, Battery used for rover information

For the beginning of the design process a list of suitable materials was made. This included the possibility of making the body of the frame out of aluminum. The choice of aluminum was simply the best for the box because it is lightweight yet durable and very malleable. Aluminum is also a good reflector of heat, and that with the lightweight makes it an ideal material for the box. In addition, since the box will withstand high temperature, insulation would be added for extra insurance of keeping the temperature within the box less than 140°F. The choice was between foam insulation and reflective insulation. Foam insulation proved to withstand up to 250°F and the lowest R-value to obtain would be 2.7 which calculates to a little more then 60% of heat will be stopped from the insulation. However the downfall would be that the smallest thickness the foam could be is 0.5 inches and for prolonged or repeated contact with the foam may cause irritation, requiring protective gloves to be worn . Meanwhile the thickness of the reflective insulation is 1/16 inches, allowing the box to be slightly smaller and being easier to place in the rover. The reflective insulation doesn't require any protection for the user if they come in contact with it. Reflective insulation consists of a double layer of polyethylene bubbles, sandwiched between two highly reflective aluminum sheets allowing the heat to be reflected off the sheets and the double bubble provides an effective thermal break. Based on how the insulation is applied the lowest R-value the insulation may have is about R-3.0 and could operate at 250°F. With the two options to decide, the reflective insulation was the best due to the fact it will be safer for the user to retrieve the battery when needed and would allow the box to fit into the compact room of the rover.

Leading to the design of the protection box. The basic idea of the box is to build a box that would house the battery and have a detachable top which may be removed or opened by unscrewing the screws that keep the top in place. There were two approaches for the design of the box. The first was to have the top be composed of two faces of the box so the user could have a wider opening for retrieving the battery. The second approach was to only have one face of the box open. The best fit for the design was the second approach. This is due to the fact that the box would be placed in a compact space so if the user must retrieve the battery swiftly having only one face would allow the user to keep the box within the rover and still be able to easily access the battery. With this in mind the design of the box continues to the formation of the main body of the box. The main body is intended to protect the battery and to do so the box must also be durable. Aluminum has a high malleability allowing it to be very capable of being shaped or bent. This may cause the final strength of the box to be less then the ideal strength but to minimize this possibility the main body would be made from one single piece of aluminum and would be bent to increase strength. In addition, to optimize the strength of the box, each corner will be strengthened with a steel square rod. Allowing the box to be structurally secured and each side to be secured tightly together with the help of the rods. Now with the top and bottom determined, to unit them

and make it easy for the user to open, the top will be attached to the bottom with a hinge and have two screws holding it down on the other end. Allowing the user to use a screwdriver to open the box and having only two screws to work with. Other than the structure of the box, the box must include ventilation. For ventilation, the best option is to add thin slits along one side of the box to allow air flow between the inside and outside but to also ensure minimal heat transfer between the outside space and inside.

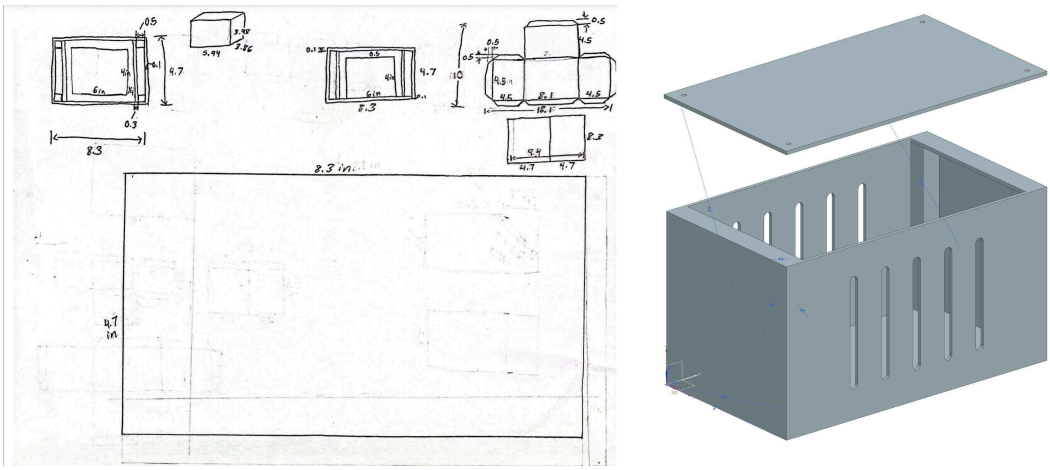


Figure 42, Hand sketch of original design for box (Left Side) and CAD design for updated version (Right Side)

When assembling the box, things to consider was the placement of the insulation and since it was a secondary boundary for temperature control the final product didn't have insulation all around the box and only in places where needed. The number of slits and size of them was also a challenge when building the box since the tools available weren't suitable for the expected outcome. The sizing of the slits weren't perfect but with the margin error they work as intended however the number of slits was smaller then expected so this decreased ventilation by almost half. Considering there should have been slits on two sides.

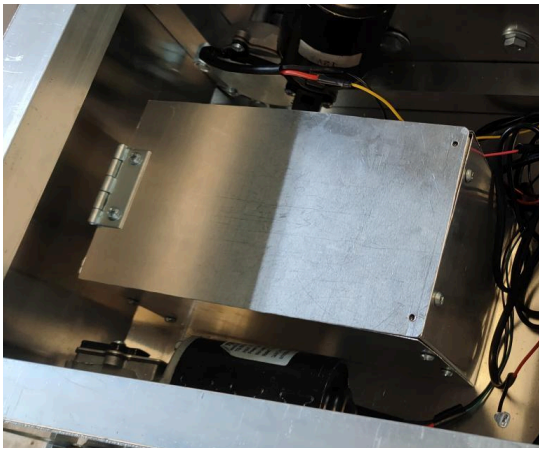


Figure 43, Final product of the battery protection box and is also shown implemented into the rover

For testing I constructed two different assessments so I could demonstrate the two main purposes of the components. The first assessment involved a demonstrator opening the box with a screwdriver. The purpose of this is to show how the box opens with ease and that a user could handle it without any safety issues. Which plays into the main goal of the subsystem of ensuring user's interactions to be safe and efficient. At the end they individually score it out of 5 for both of the categories of easily accessible and safety. The goal is to get an average of a 4 out of 5. The second assessment consisted in placing the battery protection box near an open flame and in an oven to test how much heat is resisted from the box. This would be shown by taking the surrounding temperatures and inside temperatures of the box. The purpose of this test is to show that the battery is protected from the heat it will withstand if placed in a house fire. The goal is for the inside temperature to not go over 100°F. The reason for 100°F is due to the battery having the maximum tolerance of 140°F and since the battery service life decreases each time the temperature increases from about 70°F. To minimize the decrease in life we don't want the battery to ever so slightly reach the maximum tolerance so having a good range in between the goal and actual maximum is the best course of action for the battery.

At the end of each assessment the goals for each were reached and the assessments were accomplished as planned. For the first one the goal as stated above was a score of 4 out of 5 but at the end of the demonstration the demonstrator scored it 5 out of 5. For the second assessment I placed the box on a stove top and with the fire reading of about 400°F the temperature inside stayed less than 100°F which demonstrated that the box will be able to withstand the high temperatures of 200°F, the overall group's goal temperature for the rover to withstand. In addition to being placed near an open flame, the box was placed in an oven with it being set to 250°F. To test what occurred inside the box a frozen strawberry was placed inside the box and as a control variable, one was placed outside on a tray for a timespan of 10 min. Once the

time was done the result of the test was that the strawberry inside the box was still cold to the touch and showed no signs of extensive heat affecting it. Meanwhile the strawberry on the tray open to the environment showed harm from the heat and the strawberry was melted and heated to a temperature that was over room temperature. Proving the box executes the purpose of it accurately and provides a good protection against extensive heat. At the end the subsystem accomplishes the requirements and goals.

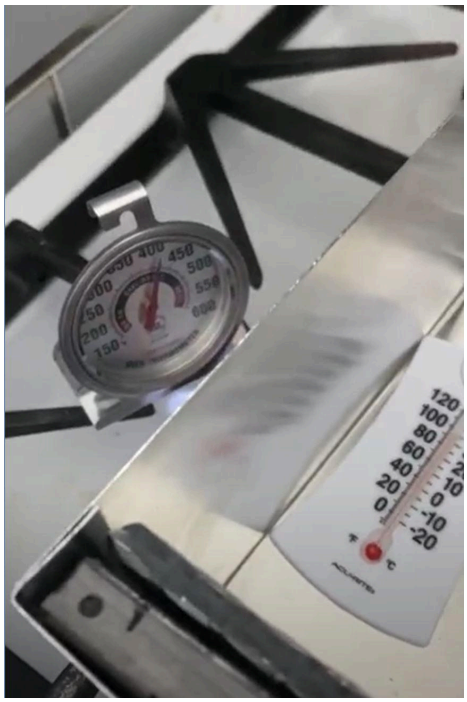


Figure 44, Demonstration of the first part of the second test

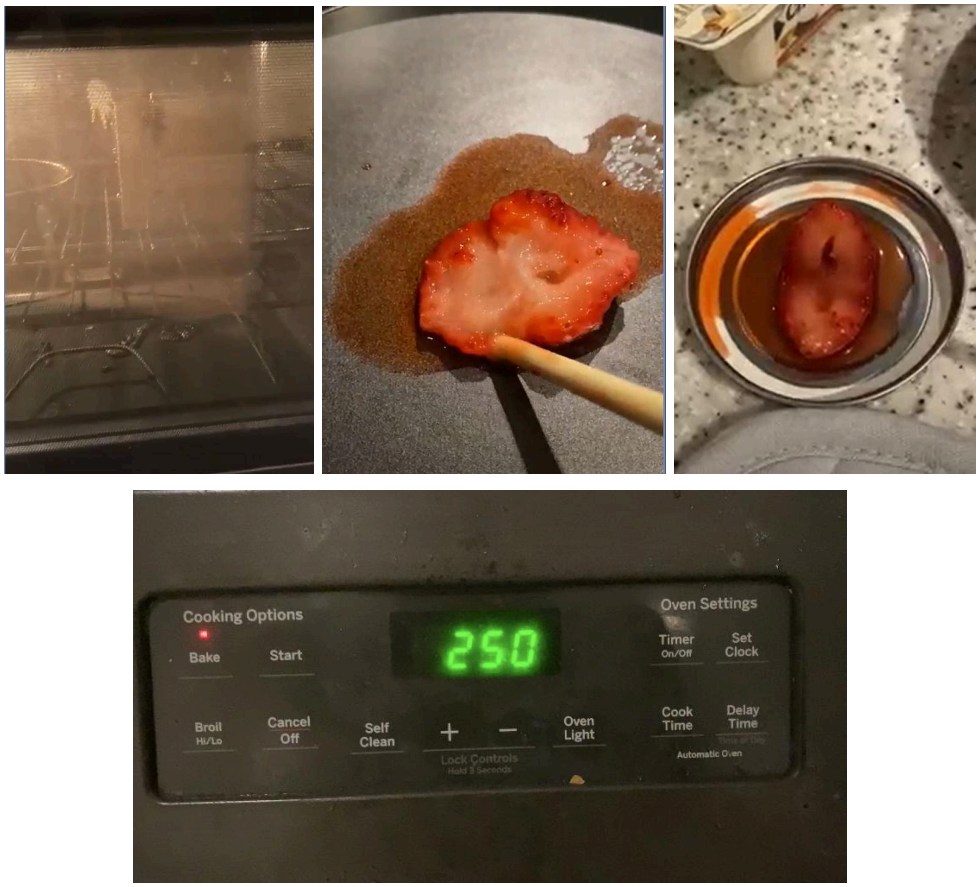


Figure 45, Demonstration of the second part of the second test

7 Results and Discussion

7.1 - Results

7.1.1 - Communications

Users can connect to the rover remotely in approximately sixty seconds. Response time on the rover is under one second, and data is transmitted practically instantly, to precision greater than 1 degree Fahrenheit. In the case of electronic failure, failsafes ensure the rover is fully operational in case of sensor disconnect, and ensures safety and longevity through shut-down failsafes.

7.1.2 - Electrical

An initial survey showed that customers would prefer at least 1 hour of battery life. After acquiring all of the components that would run off the battery, it was mathematically estimated that the entire system draw would be approximately 8.6 amps. The motors draw about 2.4-2.6 amps each at maximum speed , the Raspberry Pi and motor controller combined draw up to 3 amps peak, but normally sit around much less than that. The temperature sensor, camera and cooling fan all draw a negligible amount of current. Combined, the maximum theoretical current draw is 8.6 amps, and

the battery can supply 12 amp-hours on a full charge. Using the equation Runtime = Suppliable Current / Current Draw, 12A*h / 8.6 Amps = 1.395 hours, or 83 minutes. For the safety and longevity of the battery, the user would be advised to not run the rover for much longer than 1.25 hour, ensuring no permanent damage is done to the battery. This is the minimum calculated runtime, and in real-world testing the current draw from the Raspberry Pi is much lower, as it is not running at 100% load all the time. Real world peak current draw was measured to be approximately 6.3 A at full load, as was demonstrated during this subsystem's demonstration. This would result in a full-load runtime of 1.90 hours, or 114 minutes in real-world testing. Actual runtime will be longer than this figure, dependent on how often the rover is moving at full speed or if it is just collecting data. Shown in Figure 9 in section 6.2 are two separate current readings: the first, 0.37 A, is the Pi running the code at idle. The second reading, 6.43 A, shows the approximate current draw when the code and motors are running.

7.1.3 - Black Box

The black box met all requirements for heat retention,beyond the required 200 degree temperature, lasting beyond five minutes. Additionally, the black box successfully houses raspberry pi within, without leading to mechanical or electronic failure.

7.1.4 - Drivetrain

While the suspension would have been able to support the 30lb projected weight of the rover, it was not able to fully articulate under the actual 40-50lbs that the rover actually weighed. During testing, the rover was able to travel at an average speed of 1 m/s. The suspension was successfully able to articulate enough to allow the rover to traverse a 1 inch bump.

7.1.5 - Frame

For the frame, the customer requirement had its main focus on its strength or capacity to withstand a considerable force impact. Based on the customer interview the target specification was to withstand a 1 kN impact force.

The test was focused on testing the amount of impact force the frame can withstand from a falling object. A metal socket was dropped from a specific height. The test was done twice with a mass of the socket of approximately 1 kg and the height was 1 m and 1.5m.

The math of the test was based on Newtonian physics using the laws of motion and conservation of energy and is provided below. Figure 46 shows the scenario of the test. [21][22]

$$PE = mgh$$
$$KE = \frac{1}{2}mv^2$$
$$v = \sqrt{2gh}$$
$$W = PE$$
$$F = \frac{mgh}{d}$$

Where:
PE= Potential energy
KE= kinetic energy
m= mass
g= gravity force
h= height
v=velocity of impact
d= penetration distance or distance traveled after impact

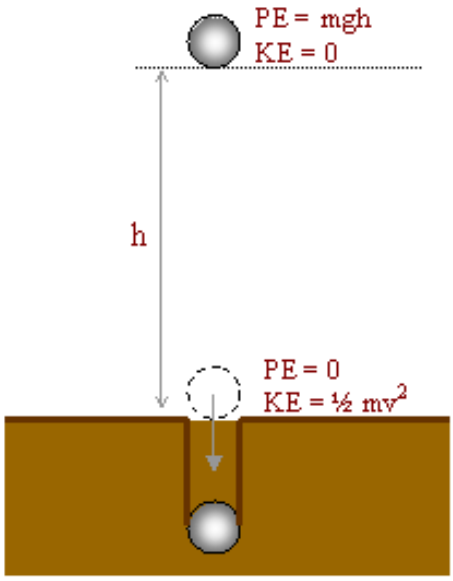


Figure 46, Representation of the math [23]

Using the actual values of the test, the impact force was computed to be 2940 N.

$$F = \frac{1 * 9.81 * 1.5}{0.005} = 2940\ N$$

The pictures shown below, Figure 47 and 48 show the frame with the metal socket used and the frame after the impact with the estimated distance after impact of 0.005 m.

Figure 47, frame and metal socket used in the testing

Figure 48, Impact site after the test

Also as shown in figure 48 above, the test was successful, the frame was able to withstand the impact with a really small bend. It can be concluded that the frame is strong and very solid and can serve its purpose.

7.1.6 - Ergonomics

7.2 - Significant Technical Accomplishments

Throughout all subsystems, we learned that many different programs were more useful than previously thought. For the communications subsystem, various Python libraries and Raspberry Pi utilities, such as the built-in Python programmer, were much more effective at compiling code and accessing different hardware devices than expected.

8 Conclusions

The team completed every goal it set out to achieve, except for some shortcomings in maneuverability, due to machining and assembly issues. Moving forward, optimizations should be done to reduce weight of the rover, as rover weight greatly exceeded original expectations.

However, most of the goals being met and exceeded is not enough to bring the rover to market. While the prototype exceeds expectations in testing for a specific scenario (Fires), much more could be done to improve versatility in other situations, such as waterproofing, more sensor implementation, and improved maneuverability.

Any team attempting to take the product to market would need to find better equipment to put inside the rover, reducing weight and increasing efficacy. Namely, improvements can be made to the battery to reduce weight, and to the motors, which are notably heavy, and have higher torque than required.

After making the above changes, the team should continue to rigorously test the rover, and make changes based on the shortcomings observed. Possibilities include increasing maneuverability over increasingly difficult obstacles, or improving upon code to allow more sophisticated sensors to be implemented. These goals would ultimately improve upon the rover's versatility and robustness.

All in all, the first prototype of the rover presented several technical challenges that were mostly met. However, future teams could easily find several areas to improve upon the prototype, if provided with a large enough budget/resources.

9 References

[1]I. State, "How to Build a Raspberry Pi Temperature Monitor," *Medium*, Aug. 14, 2020. https://medium.com/initial-state/how-to-build-a-raspberry-pi-temperature-monitor-8c2f70acaea9

[2]Raspberry Pi Foundation, "Raspberry Pi — Teach, Learn, and Make with Raspberry Pi," *Raspberry Pi*, 2019. https://www.raspberrypi.org/

[3]https://www.facebook.com/circuitbasic, "Raspberry Pi DS18B20 Temperature Sensor Tutorial," *Circuit Basics*, Mar. 25, 2016. https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/

[4]S. C. | R. Pi | 102, "How to Setup a Raspberry Pi Without a Monitor or Keyboard," *Circuit Basics*, Jan. 20, 2015. https://www.circuitbasics.com/raspberry-pi-basics-setup-without-monitor-keyboard-headless-mode/(accessed Apr. 25, 2022).

[5]"How to Write to CSV Files in Python," *Python Tutorial - Master Python Programming For Beginners from Scratch*. https://www.pythontutorial.net/python-basics/python-write-csv-file/

[6]"python - Matplotlib (pyplot) savefig outputs blank image," *Stack Overflow*. https://stackoverflow.com/questions/9012487/matplotlib-pyplot-savefig-outputs-blank-image(accessed Apr. 25, 2022).

[7]"Try, Except, else and Finally in Python," *GeeksforGeeks*, Aug. 22, 2020. https://www.geeksforgeeks.org/try-except-else-and-finally-in-python/(accessed Apr. 25, 2022).

[8]"Multithreading in Python | Set 1," *GeeksforGeeks*, Jul. 13, 2017. https://www.geeksforgeeks.org/multithreading-python-set-1/

[9]"R6 Light Commercial Rover," *AION ROBOTICS*. https://store.aionrobotics.com/products/r6-light-commercial-rover(accessed Apr. 27, 2022).

[10]Battery University, "Can the lead-acid battery compete in modern times?," *Battery University*, 17-Feb-2022. [Online]. Available: https://batteryuniversity.com/article/can-the-lead-acid-battery-compete-in-modern-times. [Accessed: 27-Apr-2022].

[11]"ISO-C1 Polyisocyanurate Safety Data Sheet," *Polyisocyanurate (Polyiso) Insulation and EPS (Expanded Polystyrene)*. [Online]. Available: https://www.dyplastproducts.com/msds-isoc1. [Accessed: 27-Apr-2022].

[12]"Reflective insulation: What it is & how it works," *Gold Star Insulation*, 29-Jul-2016. [Online]. Available: https://www.goldstarinsulation.com/blog/how-reflective-insulation-works. [Accessed: 27-Apr-2022].

[13]"RFOIL technical data - cnbhnp.com." [Online]. Available: http://www.cnbhnp.com/images/polycarb-sealing-insulation/rfoil-technical-data.pdf. [Accessed: 27-Apr-2022].

[14]"Amazon.com: SZDoit Smart Shock Absorption Robot Tank Car Chassis Kit with Suspension System for Arduino Raspberry Pi DIY STEAM Education Platform Tracked Vehicle (Black Frame + Silver Wheel) : Toys & Games," *www.amazon.com*. https://www.amazon.com/SZDoit-Absorption-Raspberry-Education-Platform/dp/B089475848/ref=pd_lpo_2?pd_rd_i=B089475848&th=1 (accessed Apr. 27, 2022).

[15]"Track Systems - 28 June 2008," *www.rctankcombat.com*. https://www.rctankcombat.com/articles/track-systems/ (accessed Apr. 28, 2022).

[16]"RC40-1R-K1-2PL-10FT #40 K1 Roller Chain - Attachments Every 2nd Pitch - ANSI Standard Roller Chain," *NitroChain.com*. https://www.nitrochain.com/40-k1-2p-attachment-roller-chain-10ft?gclid=Cj0KCQjw06OTBhC_

ARIsAAU1yOUibs3FxQR0E2k4ZE2g4d5ctn9DGjybVhi0aylkO3WB_bjJrZXJ_5gaAvw7EALw_wcB (accessed Apr. 27, 2022).
[17]amando96More, "How to Make Custom and Strong Tank Tracks for Very Cheap.," *Instructables*. https://www.instructables.com/How-to-make-custom-and-strong-tank-tracks-for-very/ (accessed Apr. 27, 2022).
[18]"Amazon.com: GDOOL Front Rear RC Shock Absorber, 4-Pack Aluminum 98mm Adjustable Assembled Suspension for 1:10 Redcat HSP 94166 94106 94107 94155 94170 Buggy Truck Crawler Upgraded Hop-up Parts (Black) : Toys & Games," *www.amazon.com*. https://www.amazon.com/GDOOL-Absorber-Adjustable-Assembled-Suspension/dp/B09164YLNT/ref=asc_df_B09164YLNT/?tag=hyprod-20&linkCode=df0&hvadid=533376926530&hvpos=&hvnetw=g&hvrand=4747609532682283213&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9004710&hvtargid=pla-1424394561564&th=1 (accessed Apr. 27, 2022).
[19]Thingiverse.com, "RC tank tracks by Evilman," *www.thingiverse.com*. https://www.thingiverse.com/thing:518021 (accessed Apr. 27, 2022).
[20]Z. Agustin et al., "Scholar Commons SAFER: Search and Find Emergency Rover Recommended Citation," 2016. Accessed: Apr. 28, 2022. [Online]. Available: https://scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1062&context=mech_senior (accessed Apr. 28, 2022).
[21]"Energy of falling object," Gsu.edu, 2020. http://hyperphysics.phy-astr.gsu.edu/hbase/flobi.html. (accessed Apr. 28, 2022).
[22]"How to Calculate the Force of a Falling Object," Sciencing, 2018. https://sciencing.com/calculate-force-falling-object-6454559.html. (accessed Apr. 28, 2022).
[23]"Solve problem related to impact force from falling object," www.livephysics.com. https://www.livephysics.com/tools/mechanics-tools/solve-problem-related-impact-force-falling-object/ (accessed Apr. 28, 2022).
[24]"Thermal Conductivity of Steel," Thermtest Inc., May 04, 2021. https://thermtest.com/thermal-conductivity-of-steel#:~:text=The%20thermal%20conductivity%20of%20steel
[25]The Editors of Encyclopedia Britannica, "Young's modulus," Encyclopædia Britannica. Jan. 10, 2019. [Online]. Available: https://www.britannica.com/science/Youngs-modulus

10 Appendix A: Selection of Team Project

The team discussed various different potential projects before deciding on the emergency rover. Ideas discussed included a variety of rovers, such as a grocery carrier rover, snow shoveling rover, a human exoskeleton support, and a concussion detector. While most of those projects would have been more than viable for this class conceptually, they often did not have enough subsystems present to include all of our teammates, nor were some of them feasible for college students to design. The emergency rover had enough subsystems, and we surmised that the rover would be able to be completed and functional in a timely manner. In order to effectively narrow down between the potential projects, the team met in a Discord meeting and compared the positives and negatives of each potential project.

Originally, potential projects included a snow removal rover, bionic arm, sensor helmet, and a 3d printer. Amongst these potential ideas, we eliminated several based on theoretical difficulty and safety. Namely, the sensor helmet was eliminated for simplicity, the 3d printer was eliminated for complexity, and the bionic arm was deemed unsafe to execute. The snow removal rover was deemed to be similar to our rescue rover, and simply had its objectives modified until the team arrived at the rescue rover.

An additional factor in choosing the rescue rover was due to the practicality and humanitarian aspect associated with it. Rather than being a “convenience” tool, the team’s goals were greater. We wanted to both improve our engineering skills in a manageable fashion, and also benefit local emergency response teams.

11 Appendix B: Customer Requirements and Technical Specifications

Table 7 - Customer Requirements and Technical Specifications

Customer Need	Requirement	Target Values	Actual Values
Robust/Durable	Withstands high-force hits from outside (kN)	Withstands 1 kN force to chassis	Upwards of 2.8 kN
Easy to deploy	Low weight/size, fast to deploy	Under 1 minute to deploy, 30lb max, handheld	Once given power, can be connected to in under 2 minutes
Simple to operate	Operates on pre-existing hardware	Software functions on windows, mac	Functions on windows, mac, linux
Won't get stuck	Traverses rubble, small debris, stairs	Goes over 1" bump without failure	Can go across one inch bumps
Withstands field-temperature	Survives in fire-like temperatures	200° Fahrenheit for 4 minutes	Can withstand external 200° F for 4+ minutes while maintaining an internal temperature of 100° F.
Records/provides useful data; Protected data	Sensors transmit data live, save data periodically	Live feed w/ .5 sec delay	Live video feed, roughly .25 sec delay
Versatile in utility	Modularity/sensors can be easily replaced	Min. 2x sensors, with ability to replace	Utilizes video, temperature sensor, and internal temp sensor. Replacement requires plug in to raspberry pi and simple terminal command/ small change reflected in code

12 Appendix C: Gantt Chart

WBS	Task	Lead	Start	End	Days	% Done	Work Days
1	**Due Dates**	All	Mon 2/14/22	Mon 4/25/22	71		51
1.1	Milestone 1 Memos	All	Mon 2/14/22	Wed 3/02/22	17	100%	13
1.2	Milestone 2 Forms	All	Thu 3/03/22	Mon 4/04/22	33	100%	23
1.3	Milestone 3 Reports/Presentation	All	Tue 4/05/22	Mon 4/25/22	21	100%	15
1.4	(Presentaion is given on 4/25)						
2	**Ergonomics / Safety**		Mon 2/14/22	Mon 4/11/22	57		41
2.1	Brainstorm consepts for safety implementation	Alexis	Mon 2/14/22	Fri 2/18/22	5	100%	5
2.2	Determine material needed	Alexis	Sat 2/19/22	Fri 2/25/22	7	100%	5
2.3	Design/CAD safety componets	Alexis	Sat 2/26/22	Wed 3/09/22	12	100%	8
2.4	Build black box for battery	Alexis	Thu 3/10/22	Wed 3/23/22	14	100%	10
2.5	Design easy transportation	Alexis	Thu 3/24/22	Sat 4/02/22	10	100%	7
2.6	Insure exposed contents are coved	Alexis	Sun 4/03/22	Mon 4/11/22	9	100%	6
3	**Programming/Communication**	Jonas	Mon 2/14/22	Fri 4/15/22	61		45
3.1	Buy & Test camera on Pi	Jonas	Mon 2/14/22	Sun 2/20/22	7	100%	5
3.2	Get pi camera view remotely from PC	Jonas	Mon 2/21/22	Sun 2/27/22	7	100%	5
3.3	Get wiring/control set up for movement	Jonas	Mon 2/28/22	Sun 3/13/22	14	100%	10
3.4	Confirm movement/RC	Jonas	Mon 3/14/22	Sun 3/27/22	14	100%	10
3.5	implement sensors	Jonas	Mon 3/28/22	Fri 4/08/22	12	100%	10
3.6	Prepare for presentations, documentation	Jonas	Sat 4/09/22	Fri 4/15/22	7	100%	5
4	**Electronics**	Dan	Mon 2/14/22	Mon 4/18/22	64		46
4.1	Create schematic	Dan	Mon 2/14/22	Wed 2/16/22	2	100%	3
4.2	Buy and test components	Dan	Wed 2/16/22	Sat 3/05/22	1	100%	13
4.3	Determine physical layout of hardware	Dan	Sat 2/26/22	Fri 3/25/22	3	100%	20
4.4	Begin assembly of components	Dan	Sat 2/26/22	Mon 4/04/22	1	100%	26
4.5	Complete component assembly	Dan	Tue 4/05/22	Fri 4/08/22	1	100%	4
4.6	Combine and test complete hardware	Dan	Sat 4/09/22	Fri 4/15/22	1	100%	5
4.7	Create compnent for presentation	Dan	Sat 4/16/22	Mon 4/18/22	1	100%	1
5	**Frame**	Andres	Mon 2/14/22	Sun 4/17/22	63		45
5.1	Have sketches of possible frames/ gather dimensions	Andres	Mon 2/14/22	Wed 2/23/22	10	100%	8
5.2	Math-physics related	Andres	Thu 2/24/22	Wed 3/02/22	7	100%	5
5.3	CAD	Andres	Fri 3/04/22	Thu 3/17/22	14	100%	8
5.4	Get materials/parts	Andres	Fri 3/04/22	Thu 3/17/22	14	100%	8
5.5	Building of the frame	Andres	Fri 3/18/22	Thu 4/07/22	21	100%	15
5.6	Assample the parts	Andres	Fri 4/08/22	Tue 4/12/22	5	100%	1
5.7	Testing	Andres	Wed 4/13/22	Sun 4/17/22	5	100%	1
6	**Black Box**		Mon 2/14/22	Fri 3/18/22	33		25
6.1	Materials Selection	Diego	Mon 2/14/22	Wed 2/16/22	3	100%	3
6.2	Acquiring the materials	Diego	Thu 2/17/22	Wed 2/23/22	7	100%	5
6.3	Measuring of all inside components	Diego	Thu 2/24/22	Sun 2/27/22	4	100%	2
6.4	Construction of Black Box	Diego	Mon 2/28/22	Wed 3/09/22	10	100%	8
6.5	Application into whole robot	Diego	Thu 3/10/22	Fri 3/18/22	9	100%	7
7	**Drivetrain/Gearing**		Mon 2/14/22	Sun 4/10/22	56		40
7.1	design motors/gearing/treads	Ashton	Mon 2/14/22	Sun 2/27/22	14	100%	10
7.2	design suspension	Ashton	Mon 2/28/22	Sun 3/13/22	14	100%	10
7.3	mock up attachment to frame	Ashton	Mon 3/14/22	Sun 3/20/22	7	100%	5
7.4	test subsystem with rest of rover	Ashton	Mon 3/21/22	Sun 3/27/22	7	100%	5
7.5	crash test	Ashton	Mon 3/28/22	Sun 4/10/22	14	100%	10

Rescue Rover Project Schedule

Go Pro

Project Start Date: 2/14/2022 (Monday)
Display Week: 1
Project Manager: Divided amongst team

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
14 Feb 2022	21 Feb 2022	28 Feb 2022	7 Mar 2022	14 Mar 2022	21 Mar 2022	28 Mar 2022	4 Apr 2022	11 Apr 2022	18 Apr 2022	25 Apr 2022

WBS	Task	Lead	Start	End	Days	% Done	Work Days
1	**Due Dates**	All	Mon 2/14/22	Mon 4/25/22	71		51
1.1	Milestone 1 Memos	All	Mon 2/14/22	Wed 3/02/22	17	100%	13
1.2	Milestone 2 Forms	All	Thu 3/03/22	Mon 4/04/22	33	100%	23
1.3	Milestone 3 Reports/Presentation	All	Tue 4/05/22	Mon 4/25/22	21	100%	15
1.4	(Presentaion is given on 4/25)						
2	**Ergonomics / Safety**		Mon 2/14/22	Mon 4/11/22	57		41
2.1	Brainstorm consepts for safety implementation	Alexis	Mon 2/14/22	Fri 2/18/22	5	100%	5
2.2	Determine material needed	Alexis	Sat 2/19/22	Fri 2/25/22	7	100%	5
2.3	Design/CAD safety componets	Alexis	Sat 2/26/22	Wed 3/09/22	12	100%	8
2.4	Build black box for battery	Alexis	Thu 3/10/22	Wed 3/23/22	14	100%	10
2.5	Design easy transportation	Alexis	Thu 3/24/22	Sat 4/02/22	10	100%	7
2.6	Insure exposed contents are coved	Alexis	Sun 4/03/22	Mon 4/11/22	9	100%	6
3	**Programming/Communication**	Jonas	Mon 2/14/22	Fri 4/15/22	61		45
3.1	Buy & Test camera on Pi	Jonas	Mon 2/14/22	Sun 2/20/22	7	100%	5
3.2	Get pi camera view remotely from PC	Jonas	Mon 2/21/22	Sun 2/27/22	7	100%	5
3.3	Get wiring/control set up for movement	Jonas	Mon 2/28/22	Sun 3/13/22	14	100%	10
3.4	Confirm movement/RC	Jonas	Mon 3/14/22	Sun 3/27/22	14	100%	10
3.5	implement sensors	Jonas	Mon 3/28/22	Fri 4/08/22	12	100%	10
3.6	Prepare for presentations, documentation	Jonas	Sat 4/09/22	Fri 4/15/22	7	100%	5
4	**Electronics**	Dan	Mon 2/14/22	Mon 4/18/22	64		46
4.1	Create schematic	Dan	Mon 2/14/22	Wed 2/16/22	2	100%	3
4.2	Buy and test components	Dan	Wed 2/16/22	Sat 3/05/22	1	100%	13
4.3	Determine physical layout of hardware	Dan	Sat 2/26/22	Fri 3/25/22	3	100%	20
4.4	Begin assembly of components	Dan	Sat 2/26/22	Mon 4/04/22	1	100%	26
4.5	Complete component assembly	Dan	Tue 4/05/22	Fri 4/08/22	1	100%	4
4.6	Combine and test complete hardware	Dan	Sat 4/09/22	Fri 4/15/22	1	100%	5
4.7	Create compnent for presentation	Dan	Sat 4/16/22	Mon 4/18/22	1	100%	1
5	**Frame**	Andres	Mon 2/14/22	Sun 4/17/22	63		45
5.1	Have sketches of possible frames/ gather dimensions	Andres	Mon 2/14/22	Wed 2/23/22	10	100%	8
5.2	Math-physics related	Andres	Thu 2/24/22	Wed 3/02/22	7	100%	5
5.3	CAD	Andres	Fri 3/04/22	Thu 3/17/22	14	100%	8
5.4	Get materials/parts	Andres	Fri 3/04/22	Thu 3/17/22	14	100%	8
5.5	Building of the frame	Andres	Fri 3/18/22	Thu 4/07/22	21	100%	15
5.6	Assample the parts	Andres	Fri 4/08/22	Tue 4/12/22	5	100%	1
5.7	Testing	Andres	Wed 4/13/22	Sun 4/17/22	5	100%	1
6	**Black Box**		Mon 2/14/22	Fri 3/18/22	33		25
6.1	Materials Selection	Diego	Mon 2/14/22	Wed 2/16/22	3	100%	3
6.2	Acquiring the materials	Diego	Thu 2/17/22	Wed 2/23/22	7	100%	5
6.3	Measuring of all inside components	Diego	Thu 2/24/22	Sun 2/27/22	4	100%	2
6.4	Construction of Black Box	Diego	Mon 2/28/22	Wed 3/09/22	10	100%	8
6.5	Application into whole robot	Diego	Thu 3/10/22	Fri 3/18/22	9	100%	7
7	**Drivetrain/Gearing**		Mon 2/14/22	Sun 4/10/22	56		40
7.1	design motors/gearing/treads	Ashton	Mon 2/14/22	Sun 2/27/22	14	100%	10
7.2	design suspension	Ashton	Mon 2/28/22	Sun 3/13/22	14	100%	10
7.3	mock up attachment to frame	Ashton	Mon 3/14/22	Sun 3/20/22	7	100%	5
7.4	test subsystem with rest of rover	Ashton	Mon 3/21/22	Sun 3/27/22	7	100%	5
7.5	crash test	Ashton	Mon 3/28/22	Sun 4/10/22	14	100%	10

13 Appendix D: Expense Report

Table 8 - Project Expenses

Item	Quantity	Unit Price	Subtotal	Shipping
Raspberry pi, 4b, 8gb ram w/ 128gb microSD card (w/ peripherals/kit)	1	$120	$120	$0
DS18B20 Temperature Sensor	1	$15	$15	$0
PiCamera	1	$15	$15	$0
Bike Chain	2	$8.99	$19.42	$0
Electric Motors	2	$80	$160	
Shock Absorbers (4 pack)	2	$15.98	$34.52	$0
Caulk Gun	1	$4.60	$4.60	$0
Fireplace Caulk	2	$7.54	$15.08	$0
⅛” steel rod (36”)	1	$3.84	$3.84	$0
Drivetrain Hardware	N/A	N/A	$47.86	$0
Forge membership	1	$15	$15	$0
Forge expenses	N/A	N/A	$4.00	
Total			$454.22	

14 Appendix E: Team Members and Their Contributions

14.1 - Daniel Mandras

Researched motor control methods and appropriate remote control systems. Worked to design and implement a power source, motor control system, and wiring necessary to allow the rover to properly function. Worked with other subsystems to ensure that the rover would be properly integrated. Made certain that the rover can run for an appropriate amount of time off of battery power, and that the battery can be easily changed or charged. Assisted other subsystems in the IED shop.

14.2 - Jonas Kendra

Researched Raspberry Pi coding and operation, wrote code to operate the rover, and helped organize group meetings. Worked to implement sensors to the rover, and convert sensor data into meaningful information for the user. Worked with other subsystems to ensure everything was integrated properly. Ensured user-experience in operation was optimal, and implemented methods of saving data. Fixed all bugs associated with code.

14.3 - Diego Fernandez

Researched thermal conductivity of materials to simulate appropriate heat transfer tests for the black box subsystem. Worked to design a layered box in which key computer components of the rover would be stored and protected from extreme conditions. Worked with other subsystems so that the box could be useful and integrated appropriately. Ensured that the volume of the box was not too large but still efficient enough that the inside components would remain functional through extreme temperatures. Assisted with other subsystems in the IED shop.

14.4 - Andres Lin Huang

Designed the frame of the rover and made a CAD model of it. Crafted the frame from aluminum sheets and bars using tools and machines from the IED shop. Ensure that the frame satisfies the customer requirements so it is strong enough to withstand impacts. Designed and conducted the test. Worked with other subsystems to come up with a design that can adapt to the space and integration needs required of the internal components. Assisted with sub-tasks from other subsystems.

14.5 - Alexis Hernandez

Researched thermal conductivity of materials to correctly choose material to withstand the high temperatures of a house fire for the battery protection box. Worked to design a box in which the battery for the rover would be stored and protected from extreme conditions. Worked with other subsystems so that the box integrated appropriately. Ensured the box accomplished its purpose in keeping the battery protected and was a sufficient size to fit into the rover so that a user may use the rover safely and efficiently. Assisted with other subsystems in the IED shop.

14.6 - Ashton Ivey

Researched tread and drivetrain designs. Made sketches and models for sprockets, guide wheels, tracks, driveshafts, and suspension components. Worked with frame and electronics subsystems to ensure that drivetrain would successfully integrate with those systems. Aided in testing code to ensure proper function. Helped other team members use machines in the IED and Processes shops.

15 Appendix F: Statement of Work

In our project, the team aims to produce a robust, reliable device that can be cheaply manufactured in order to serve its purpose for the emergency response community. Now, local emergency response units don't have the budget necessary to inspect an area before sending people in. As a situation where these responders are deployed may be dangerous, the team is trying to create a rover that ensures the responder's safety, by relaying critical information to the user, as well as allowing the user to survey the area risk free, without putting first responder lives in danger.

Team

Dan Mandras, Jonas Kendra, Andres Lin-Huang, Diego Fernandez, Alexis Hernandez, Ashton Ivey

Semester Objectives

1. Design and create a remote controlled rover that successfully transmits data back and forth between the user and rover.
2. Apply engineering knowledge to create a functioning product
3. Gather data from potential customers
4. Research and evaluate advanced solutions regarding remote control, data logging, networking, video compression in reference to the team's current constraints

Approach

Rapidly produce a robust, reliable device that can be cheaply manufactured in order to serve its purpose for the emergency response community.

Deliverables and Dates

Milestone 1: System Concept Reviews:
2/24/22
Milestone 2: Working Prototype/demonstration
4/18/22
Milestone 3: Design Presentation
4/25/22

16 Appendix G: Professional Development - Lessons Learned

- *K – Keep – these things worked and you would use/do them again*
  - *Gantt Chart*
  - *Weekly meetings*
  - *“Forming” phase/introductions*
  - *Discussing individual skills and abilities*
  - *Discussion about each subsystem’s progress on weekly basis*
- *P – Problem – these things did not work out well and might be avoided in the future*
  - *Division of subsystems: devote more work/time to suspension and drivetrain: Split them in two; be careful when defining subsystems*
  - *Enforce deadlines more strictly*
  - *Integrate subsystems frequently to prevent assembly issues*
- *T – Try – these things may be worth attempting in the future*
  - *Test subsystems at regular intervals, rather than end of semester*
  - *Use multiple communication methods*

17 Appendix H: Software / Technology Used

17.1 - Collaboration Among Team Members

- Discord
- Webex Teams
- Google Documents
- Google Slides
- Google Drive

17.2 - Subsystem Design

- Siemens NX
- VNC
- Goodnotes for iPad

17.3 - Programming

- Python
- Thonny
- Linux terminal
- Pi OS

17.4 - Subsystem Testing/Simulation/Emulation

- Siemens NX
- Python/Thonny

18 Appendix I: User Manual

Setup:

- Install the Raspbian OS to your Raspberry pi. Make sure it can connect to the internet.
- Install VNC software to the Raspberry pi. Make sure you connect your account to your laptop to operate it remotely.
- Install RoverController code for python from RescueRover website.
- Attach necessary sensors, and update code accordingly. For temperature sensor DS18B20, use pins 2,8,10 for Vcc, Data, and Ground. Connect red motor wires to M1 and M3, and black motor wires to M2 and M4 on the motor controller board

Operation:

- Plug Raspberry Pi into the rover; Make sure it is turned on, and battery is charged, in terminal. Be careful not to touch any live wires to avoid shock.
- Ensure Rover, laptop are connected to same wifi network for operation. It is advisable to set a default on the Raspberry pi to a source you plan on using consistently in operation
- Connect to rover/Raspberry pi through VNC
- Open RoverController code
- Click “Run”
- Watch the video feed on the top right of the screen to monitor rover, and view terminal in the bottom of screen to view sensor data as it is generated
- Follow instructions provided in the terminal: use “w” to move forward, “a” to go left, “d” to go right, and “s” to go in reverse. Additionally, “z” can be used to stop and “x” will stop operation and save data
- To analyze information after operation, look on the desktop to find CSV file, data plot, and video feed