ASME UNIVERSITY ROVER CHALLENGE: EXPO Presentation

Advised by **He Shen, PhD**

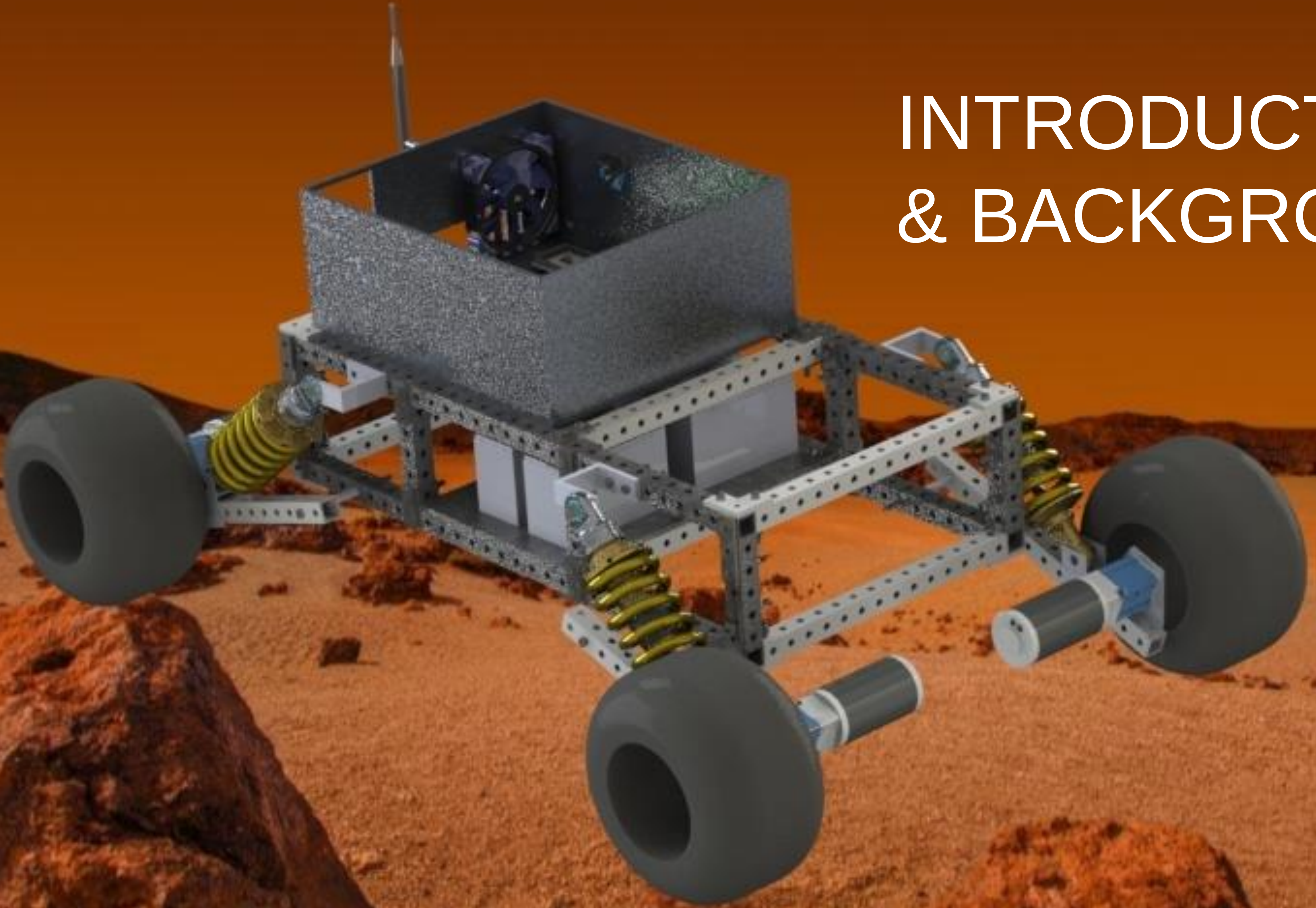
College of Engineering, Computer Science and Technology

California State University, Los Angeles

AGENDA

1. Introduction & Background
2. Technical Breakdown
3. Assessment & Summary
4. Next Steps

INTRODUCTION & BACKGROUND



TEAM MEMBERS

Group 7.1

Gabriel Zepeda
EE

Aldrin Pacquiao
ME

Frank Moreno
EE

Luis Monterroza
ME

Zack Garrett
EE

Group 7.2

Oscar Guzman
EE

Victor Gonzalez
ME

Pak Poon
ME

Zori Marfazelian
ME

Natalie Deo
EE

BACKGROUND

HISTORY

Mars exploration has piqued human interest over the past few decades.

WHY

- Habitability
- Existence of life
- Application

FUTURE

Continued exploration with hopes of human colonization.

Figure 1: Diagram of International Space Agency Mars Missions

UNIVERSITY ROVER CHALLENGE (URC)

Figure 2: URC hosted in June 2018

HISTORY

- Originated in 2006 to assist as a program to "build a rover that would be of use to early explorers on Mars."
- Held at the Mars Desert Research Station, located outside of Hanksville, Utah.
- Began as a program within the United States with just four teams participating in the first year. In 2018, 95 teams applied to participate in the challenge.
- The competition has become a part of the Rover Challenge Series which other competitions including the European Rover Challenge, Canadian International Rover Challenge and the Indian Rover Challenge.

MISSION & SCOPE

MISSION

The Mars Society's University Rover Challenge challenges students to build remotely operated rovers that can accomplish a variety of tasks that might one day assist astronauts working on the surface of Mars.

SCOPE: AUTONOMOUS NAVIGATION

Rovers shall be required to autonomously traverse to posts or between gates in this staged mission across easy and moderately difficult terrain.

CHALLENGE REQUIREMENTS

Table 1: Challenge Requirements

Design	Requirement	Engineering Specification	Justification
Size	1.2m x 1.2m	1.2m x 1.2m	Design Parameter
Weight Deployed	50kg	< 50kg	Design Parameter
Overall Weight	70kg	< 70kg	Design Parameter
Kill Switch	Kill Switch	Kill Switch	Design Parameter
Communication Distance	1km	1.2km	Design Parameter + Extra Buffer
Low-Band Frequency	900MHz 902-928Mhz	2.4GHz 2.4-2.483GHz	Design Parameter
High-Band Frequency	2.4GHz 2.4-2.483GHz	2.4GHz 2.4-2.483GHz	Design Parameter
Base Antenna Height	3m	3m	Design Parameter
Base Station Power	120V @ 60Hz	120V @ 60Hz	Design Parameter
Weather Conditions	100ºF, Dusty	Protect Vulnerable Equipment, Casing, Fans	Software & Hardware Proper Functionality
Battery Time	60 min	120 min	Mission Length + Extra Buffer

MISSION REQUIREMENTS

Table 2: Autonomous Mission Requirements

Design	Requirement	Engineering Specification	Justification
LED Indicator	Red: Autonomous Blue: Teleoperation Green (Flash): Leg Complete	Red: Autonomous Blue: Teleoperation Green (Flash): Leg Complete	Design Parameter
QR Tag	Find & Scan	Image Recognition Software	Mission Requirement
Locating	High Accuracy Location Sensor	GPS + Barometer + Gyroscope	Mission Requirement
Distance	2km	2.2km	Mission Requirement + Extra Buffer
Return Home	Return to previous gate if unable to complete leg	Return to Gate State Machine (Software)	Mission Requirement

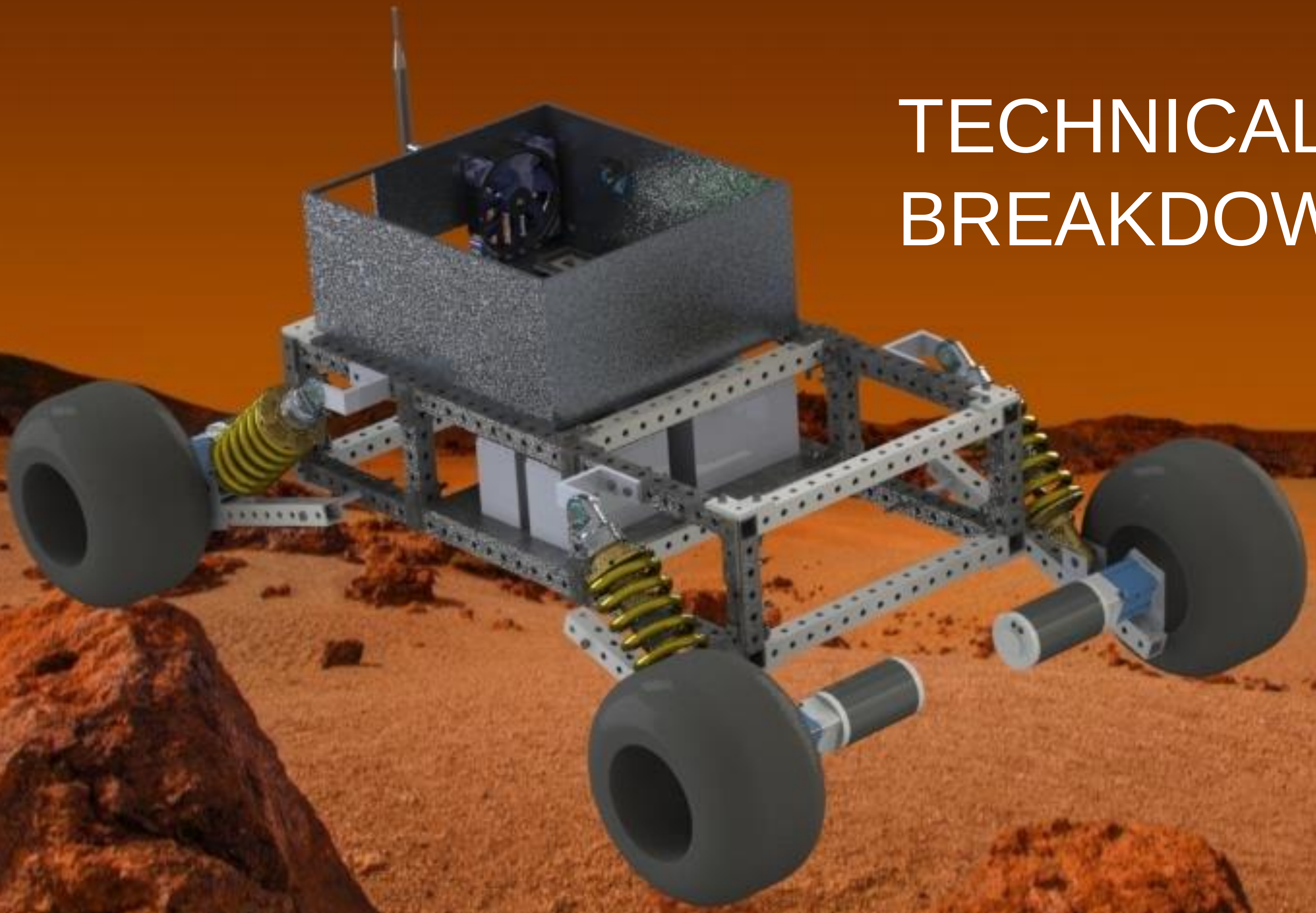
GANTT CHART OVERVIEW

Table 3: Gantt Chart Sections

Team: Subteam	Task	ASSIGNED TO	PROGRESS	START	END
Research		Team			
ME: General Design		Zori | Pak | Luis			
ME: Drive System		Victor | Luis			
EE: Power		Oscar			
EE: Communications		Gabe | Zack			
Software: Programming		Aldrin | Frank | Gabe | Nat | Pak			
Software: Image Recognition		Aldrin | Frank | Gabe | Nat | Pak			

URC

TECHNICAL BREAKDOWN



MECHANICAL ENGINEERING (ME) TEAM

Aldrin, Luis, Pak, Victor, Zori

MECHANICAL: GENERAL DESIGN

General Design → Drive System → Power → Comm. System → Programming → Image Recognition

ME GENERAL DESIGN GANTT CHART

Table 4: Gantt Chart for ME General Design Sub-team

ME: General Design		Zori | Pak | Luis			
	Frame and Suspension	Zori	100%	1/20/21	5/6/21
	Assembly	Team	35%	4/2/21	TBD
	Thermal Analysis	Luis	100%	3/12/21	3/19/21

FRAME & SUSPENSION

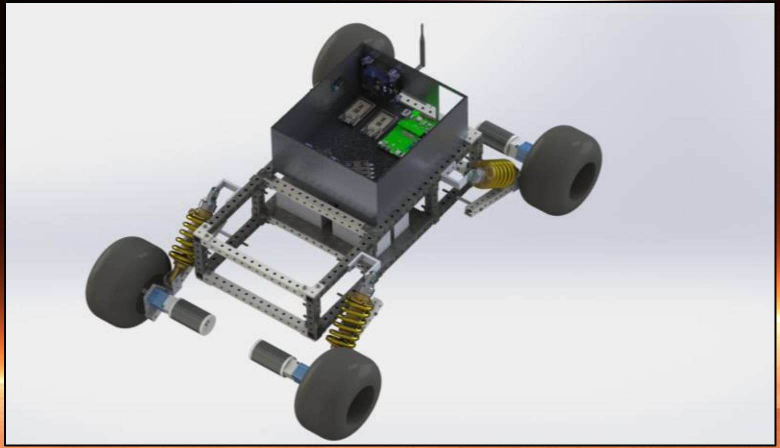


Figure 3:
Full Frame and Suspension Design Overview

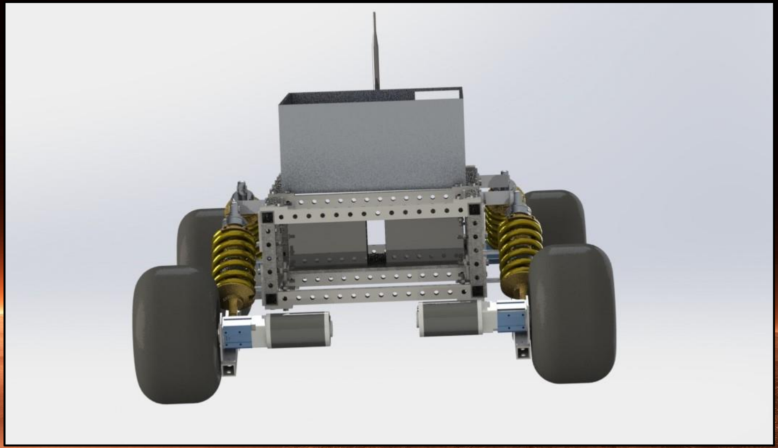


Figure 4:
Front Side view of the Frame and Suspension

Figure 5:
U-Bracket Strut Connection System

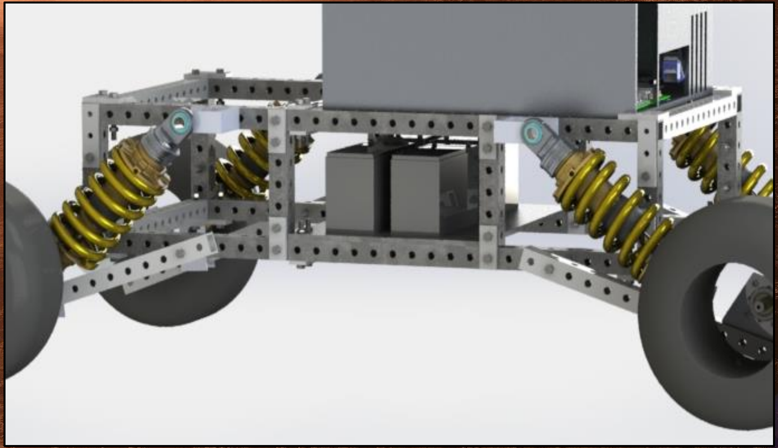


Figure 6:
Undercarriage Battery Tray Setup

THERMAL ANALYSIS (COMPONENT BOX)

Table 5: Thermal Power Dissipation

Component	# Units	Voltage (V)	Current (A)	Power (W)
USB Hub	1	12	3	36
Jetson	1	12	3	36
Arduino	1	5	1.5×10^{-3}	7.5×10^{-3}
IMU	1	3.6	2.7×10^{-3}	9.7×10^{-3}
GPS	1	3.3	25×10^{-3}	83×10^{-3}
Motor drivers	1	5	3	15
TOTAL				**87.84**

MECHANICAL: DRIVE SYSTEM

General Design | Drive System | Power | Comm. System | Programming | Image Recognition

ME DRIVE SYSTEM GANTT CHART

Table 6: Gantt Chart for ME Drive System Sub-team

Team: Subteam	Task	ASSIGNED TO	PROGRESS	START	END
ME: Drive System		Victor | Luis			
	Motor	Victor	100%	10/10/20	2/5/21
	Motor Driver	Victor	100%	10/22/20	2/5/21
	Wheels	Team	100%	10/22/20	2/5/21
	Encoder Sensor	Luis	100%	11/13/20	2/5/21

ME DRIVE SYSTEM CALCULATIONS

Force per wheel during acceleration w/Safety Factor (SF=2)

$$a = \frac{1.56\frac{m}{s}}{3\ s} = \boldsymbol{0.52\frac{m}{s^2}}$$

$$F = ma$$
$$= (50kg)\left(0.52\frac{m}{s^2}\right)$$
$$= \boldsymbol{26N}$$

$$F_{wheel} = \frac{26N}{4}x\ SF$$
$$= \boldsymbol{13\ Nm}$$

Torque per wheel for Acceleration

$$T_{wheel} = F_{wheel}\ x\ r_{wheel}$$
$$= (13\ Nm)(0.127m)$$

$$= \boldsymbol{1.651\ N}$$

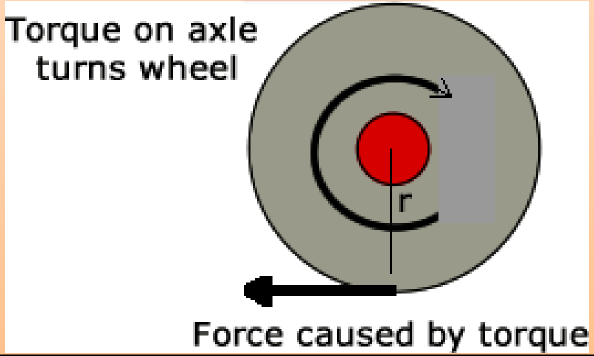


Figure 7: Depiction of torque

Max Torque Needed (SF=2)

$$F = (50\ kg)\left(9.81\frac{m}{s^2}\right)$$
$$F_{wheel} = \frac{490N}{4} = 122.5\ N$$
$$T_{max} = (122.5\ N)(0.127\ m)(SF)$$

$$= \boldsymbol{31.12\ Nm}$$

ME DRIVE SYSTEM COMPONENT LIST

Table 7: List of Drive System Components

Item Name	Product	Qty	Torque (Nm)	Voltage (V)	Power (W)	Current (A)
Motor	2.5 in CIM Motor	4	2.42	12	337 (Max)	2.7 free
Gear Box	CIM Sport Gear Box (20:1)	4	48.4	-	-	-
Motor Driver	HiLetgo Motor Driver	2	-	5.5-24	720 (Max)	12 (Continuous) 30 (Max)

ELECTRICAL ENGINEERING (EE) TEAM

Gabe, Zack, Oscar, Frank, Natalie

ELECTRICAL: POWER

General Design → Drive System → Power → Comm. System → Programming → Image Recognition

EE POWER GANTT CHART

Table 8: Gantt Chart for EE Power Sub-team

EE: Power		Oscar			
	List of Electrical Components	Oscar	100%	11/13/20	12/6/20
	Hardware Architecture Model	Oscar	100%	11/13/20	12/6/20
	Battery Selection	Oscar	100%	11/13/20	12/6/20
	Power Distribution Board	Oscar	100%	11/13/20	12/6/20
	Component Box Model	Oscar	85%	10/16/20	TBD
	Component Box Assembly	Oscar | Gabe	50%	3/30/21	TBD

EE COMPONENT LIST

Table 9: List of Electrical Components

Item Name	Brand/Product	#	Voltage (V)	Power (W)	Current (A)
Battery	Power King 12V/ 7Ah	4	12	TBD	7h
Computer	Jetson TX2	1	5.5-19.6	7.5/15	3
IMU	Adafruit 9DOF (BNO055)	1	VDD: 2.4-3.6	8.16×10^{-3}	2.72×10^{-3}
Microcontroller	Arduino Mega	2	7-12	4.5	< 0.5
GPS	Adafruit Ultimate GPS (ADA746)	1	3.3	66×10^{-3}	20×10^{-3} tracking 25×10^{-3} acquisition
Transceiver	TP-Link N300	1	9	5.4	0.6
Antenna	Tupavaco 12dBi Omni	2	Powered by Transceiver		
Camera	Camera HD 30fps	1	5	TBD	TBD
LiDAR	MakerFocus X2L	1	5	2	0.35-0.4
LEDs	Amazon-brand LEDs	5	7	0.146	20×10^{-3}

EE HARDWARE ARCHITECTURE

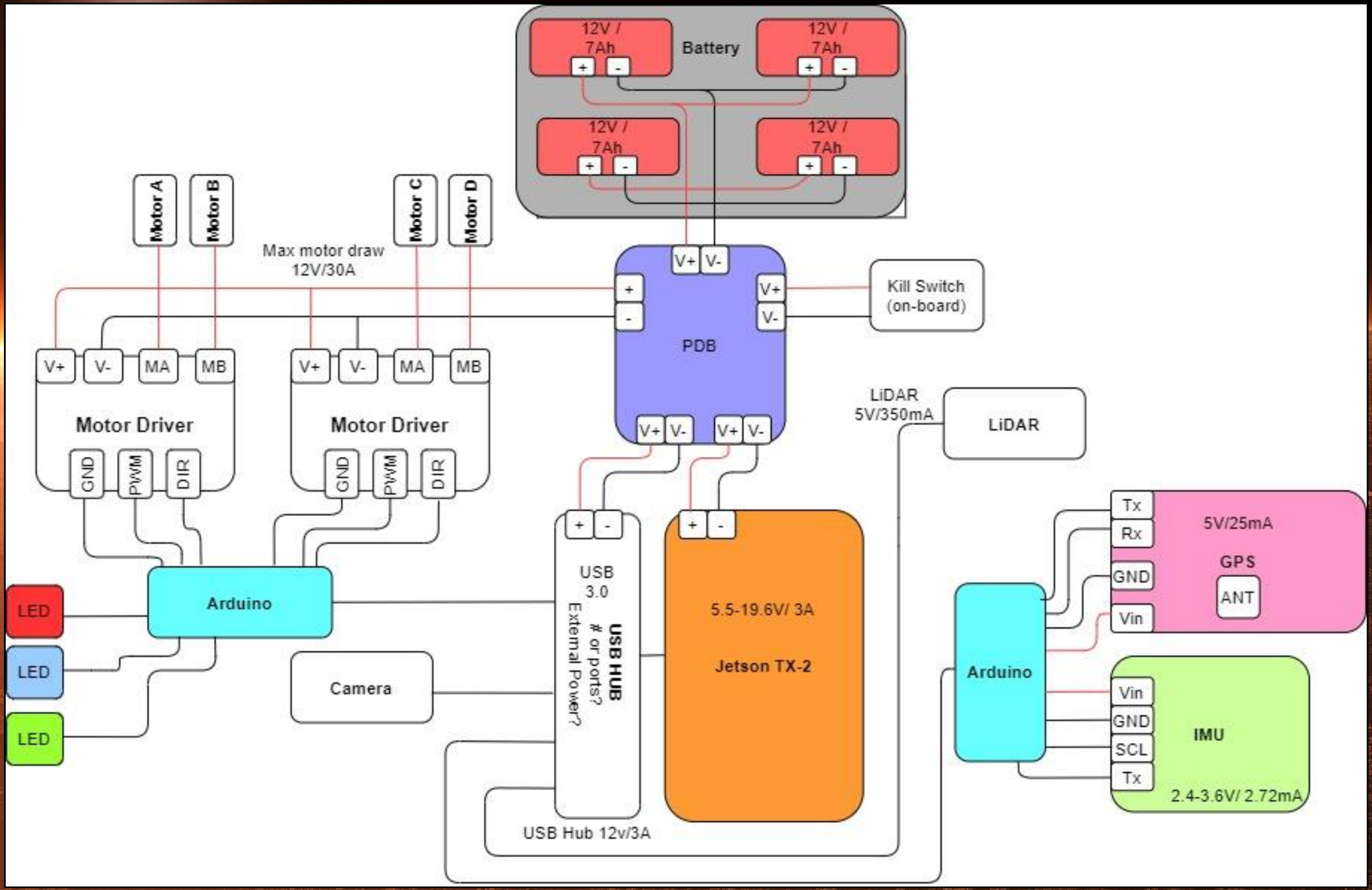


Figure 8: Rover hardware architecture

EE POWER COMPONENT BOX

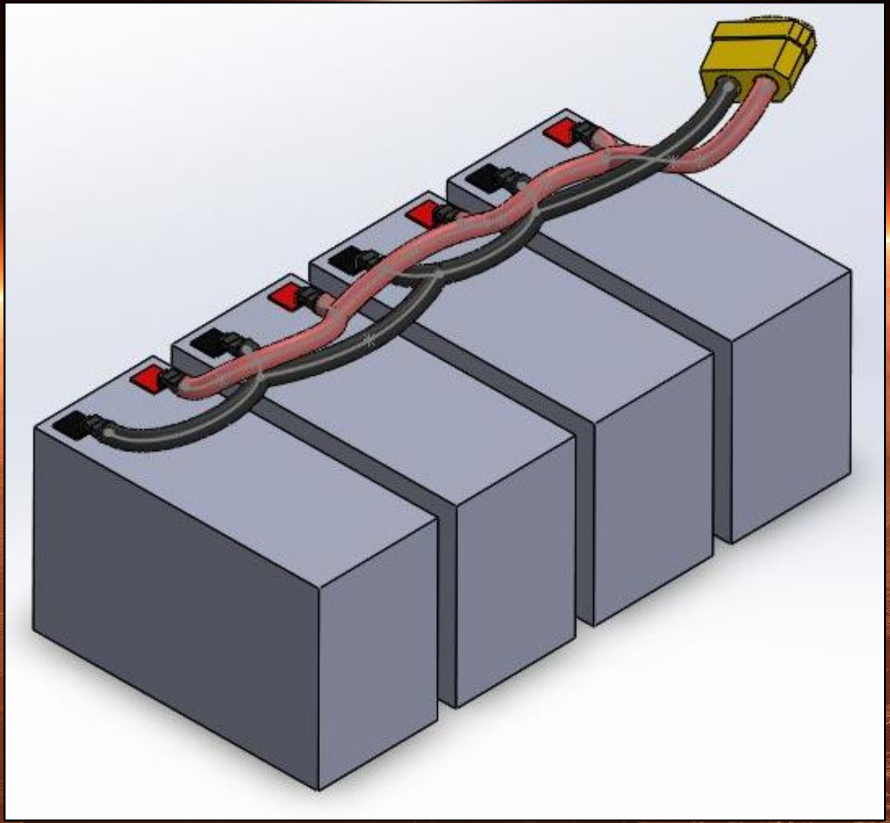


Figure 9: Batteries in parallel

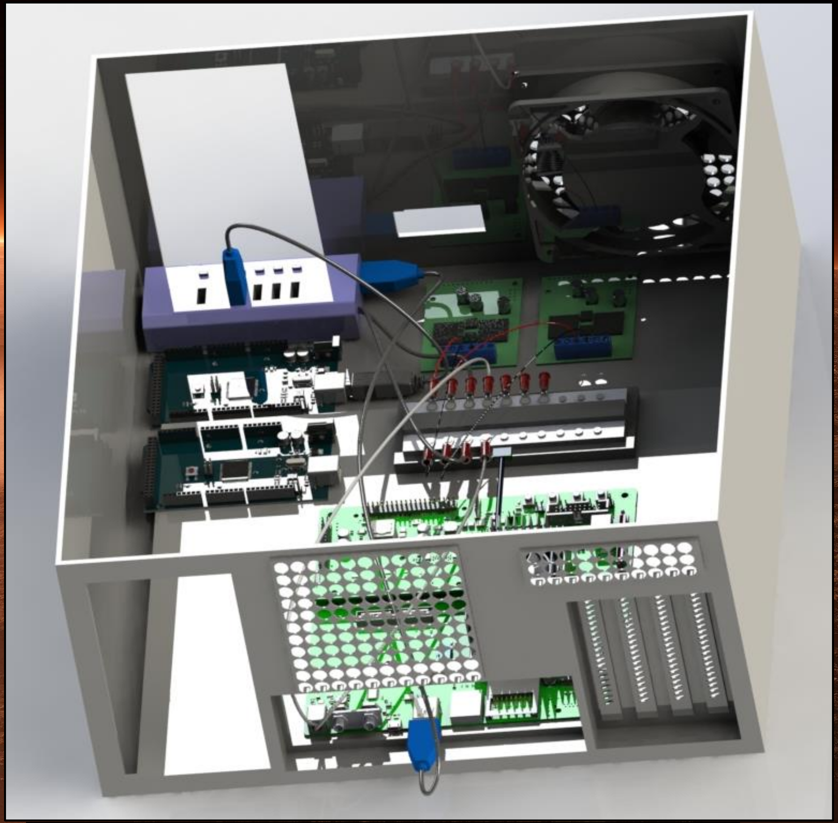


Figure 10: Component Box Model

ELECTRICAL: COMMUNICATION SYSTEM

General Design → Drive System → Power → Comm. System → Programming → Image Recognition

EE COMMUNICATION GANTT CHART

Table 10: Gantt Chart for EE Communication Sub-team

EE: Communications		Gabe | Zack			
	Communication Design	Gabe | Zack	100%	10/16/20	11/30/20
	Choose a design for wifi network	Gabe | Zack	100%	10/23/20	11/13/20
	Determine Antennas	Gabe | Zack	100%	10/23/20	1/8/21
	Access Point	Gabe | Zack	100%	10/23/20	2/5/21
	Test Equipment	Gabe | Zack	70%	4/9/21	TBD

COMMUNICATION NETWORK MODEL

Network Metrics

Access Point:

- POE compatible
- Frequency Compatibility
- Wireless Standard
- Network Speed

Figure 11: TP-Link TL-WA801ND

Omni-Directional Antenna

Figure 12: Omnidirectional Antenna

Antenna:

- Gain
- Operating Frequency
- Directivity

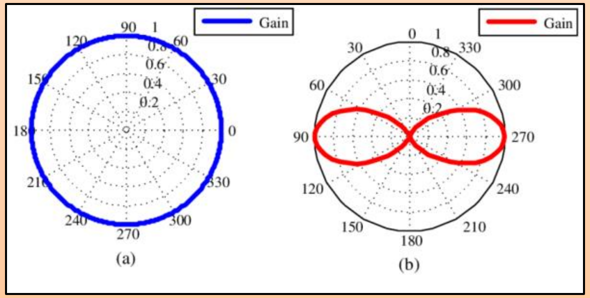


Figure 13: Omni-Directional Antenna lobe vertical view (left) and side view (right)

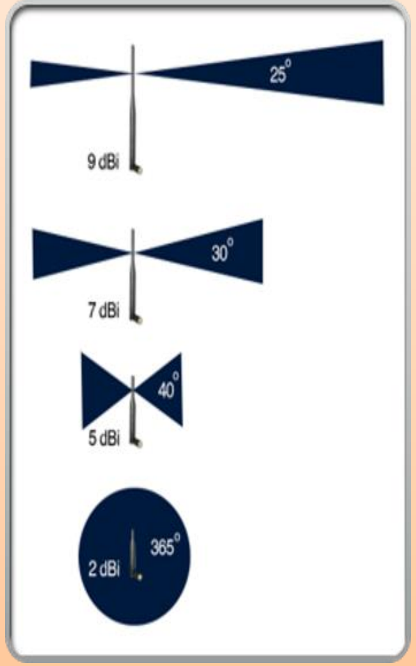


Figure 14: Omni-Directional Antenna gain difference

ANTENNAS

12 dB Antenna Gain

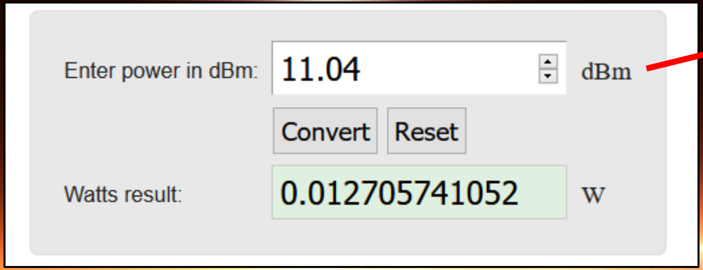


Figure 15. Antenna Power to dBm (decibel Milliwatts)

2.4 GHz operating Frequency

The signal generated by the access point must be at least **12.7 mW** to reach the specified distance of **1Km**.

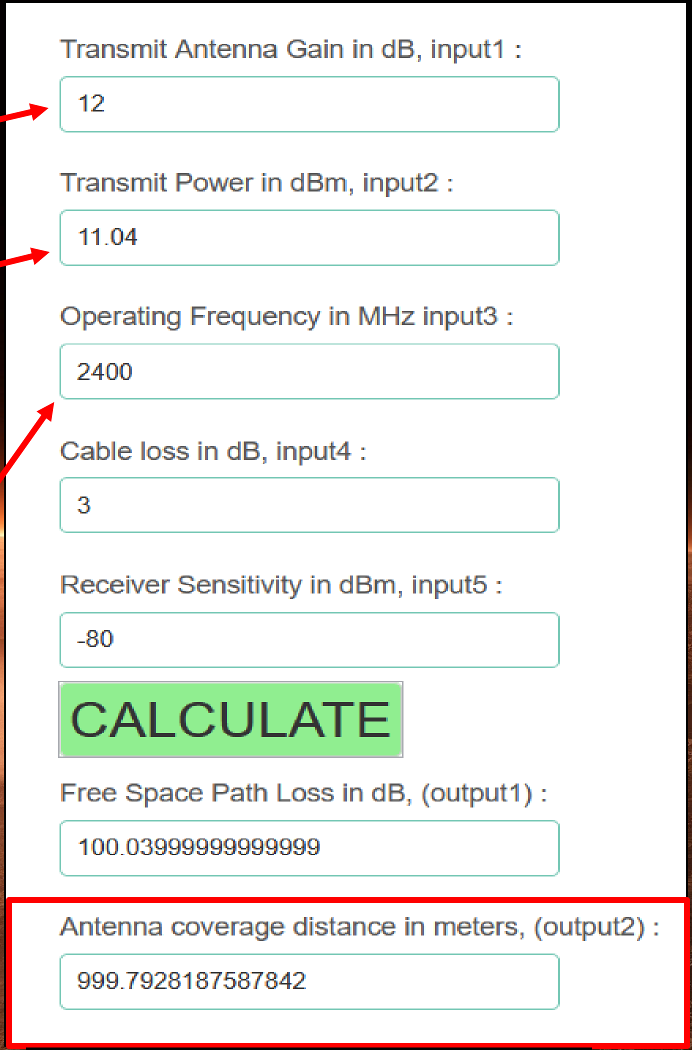


Figure 16. Antenna Range Calculator

Figure 6. Antenna Details

SOFTWARE (SW) TEAM

Aldrin, Frank, Gabe, Natalie, Zack, Pak

SOFTWARE: PROGRAMMING

General Design → Drive System → Power → Comm. System → **Programming** → Image Recognition

SW PROGRAMMING GANTT CHART

Table 11: Gantt Chart for Software General Sub-team

Software: Programming	Aldrin | Frank | Gabe | Nat | Pak			
Install Programs	SW Team	100%	10/5/20	10/12/20
ROS Workspace	Aldrin | Frank | Gabe | Nat | Pak	100%	10/12/20	10/17/20
Communication	Gabe	100%	TBD	TBD
State Machine Design	Aldrin | Frank	50%	10/22/20	TBD
Control Systems	All Teams	75%	11/6/20	TBD
Master-Slave Communication	Natalie | Gabe	100%	2/1/21	3/1/21
ROS to Arduino Integration	Aldrin	100%	10/22/20	12/30/20
Arduino to ROS Integration	Aldrin	100%	10/22/20	11/30/20
ROS Tutorials	SW Team	100%	10/5/20	10/12/20
Arduino Control File	Gabe | Aldrin	100%	3/19/21	4/2/21
Python Control File	Nat | Frank	100%	3/19/21	4/2/21
List of Sensor Components	Frank	100%	10/20/20	1/8/21

SW ARCHITECTURE

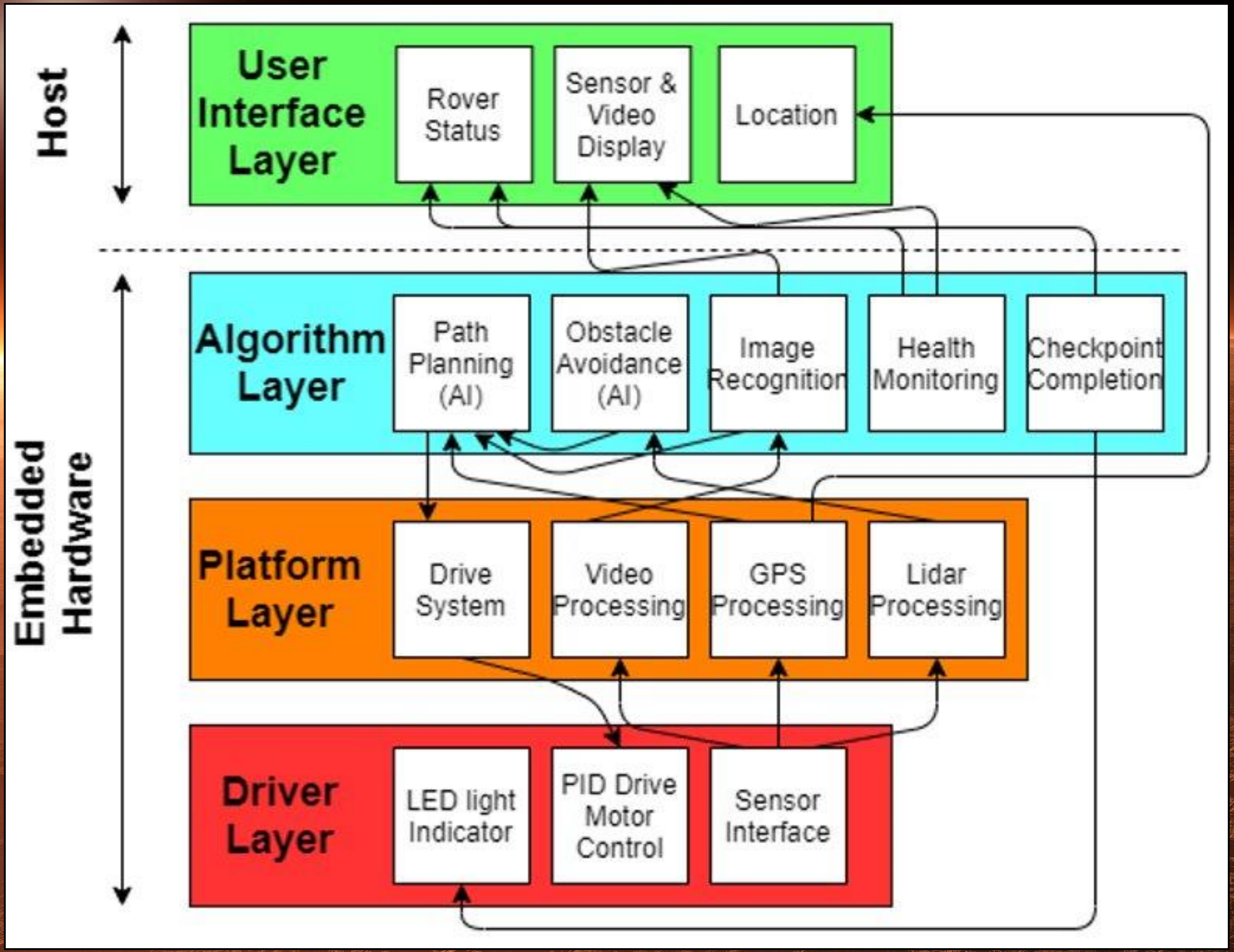


Figure 17: Software Architecture

SW PROGRAMMING STATE MACHINE

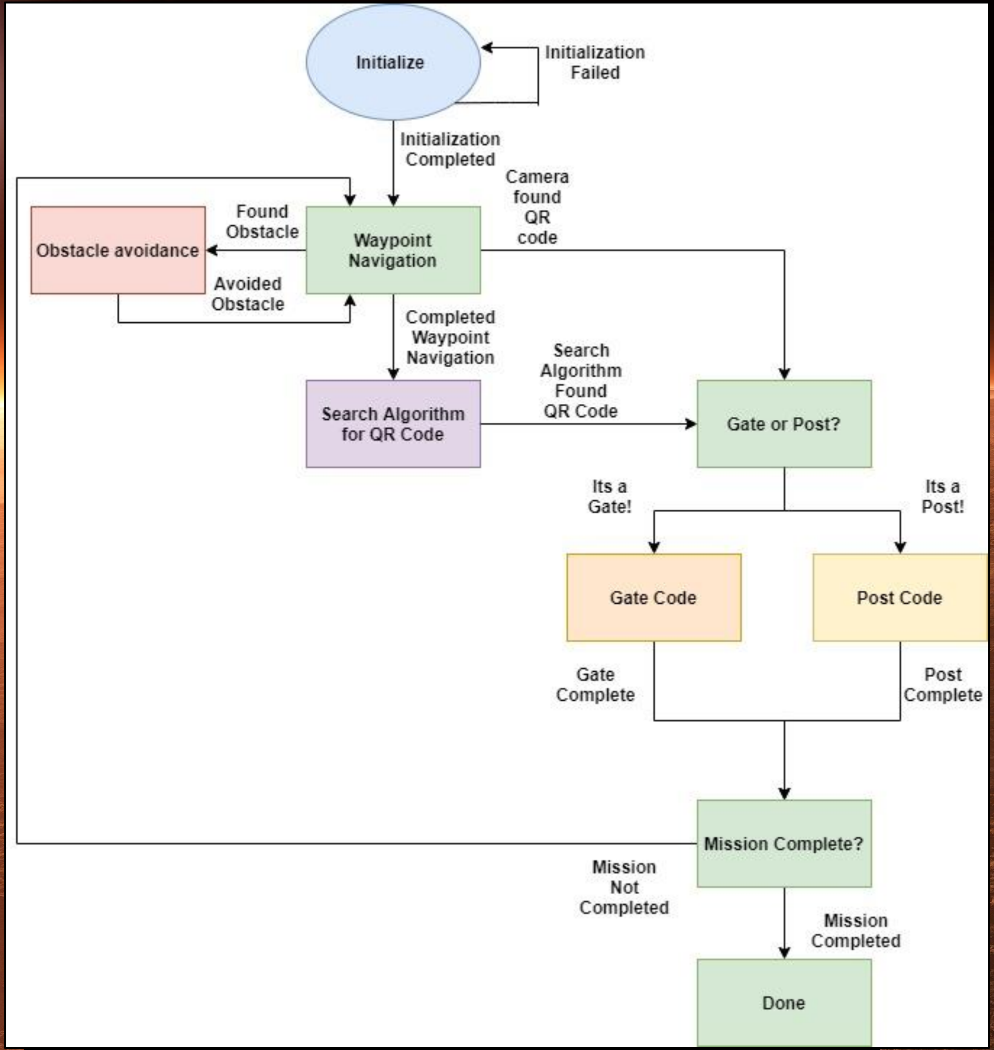


Figure 18: State Machine Block Diagram

COMPONENT COMMUNICATION HARDWARE

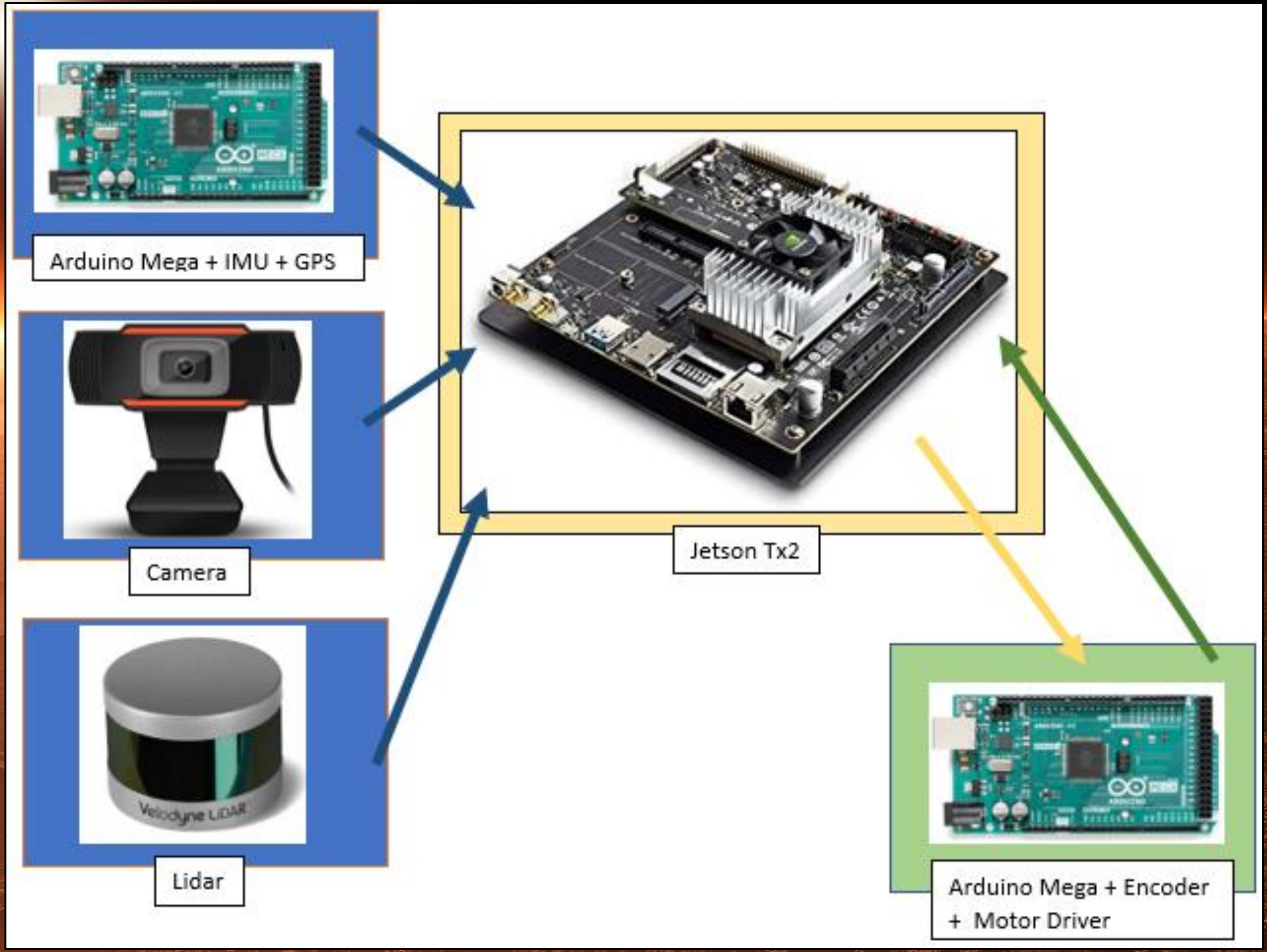


Figure 19: Component Communication Hardware

SW INFORMATION PATH

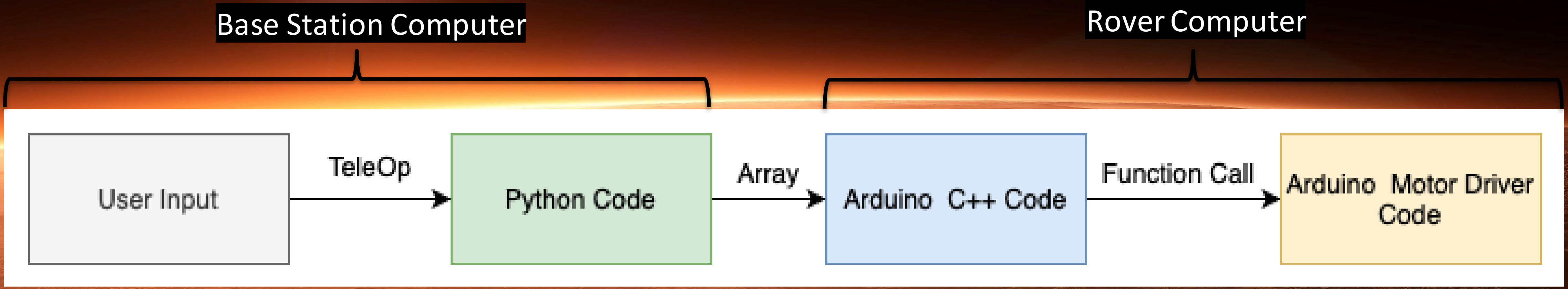
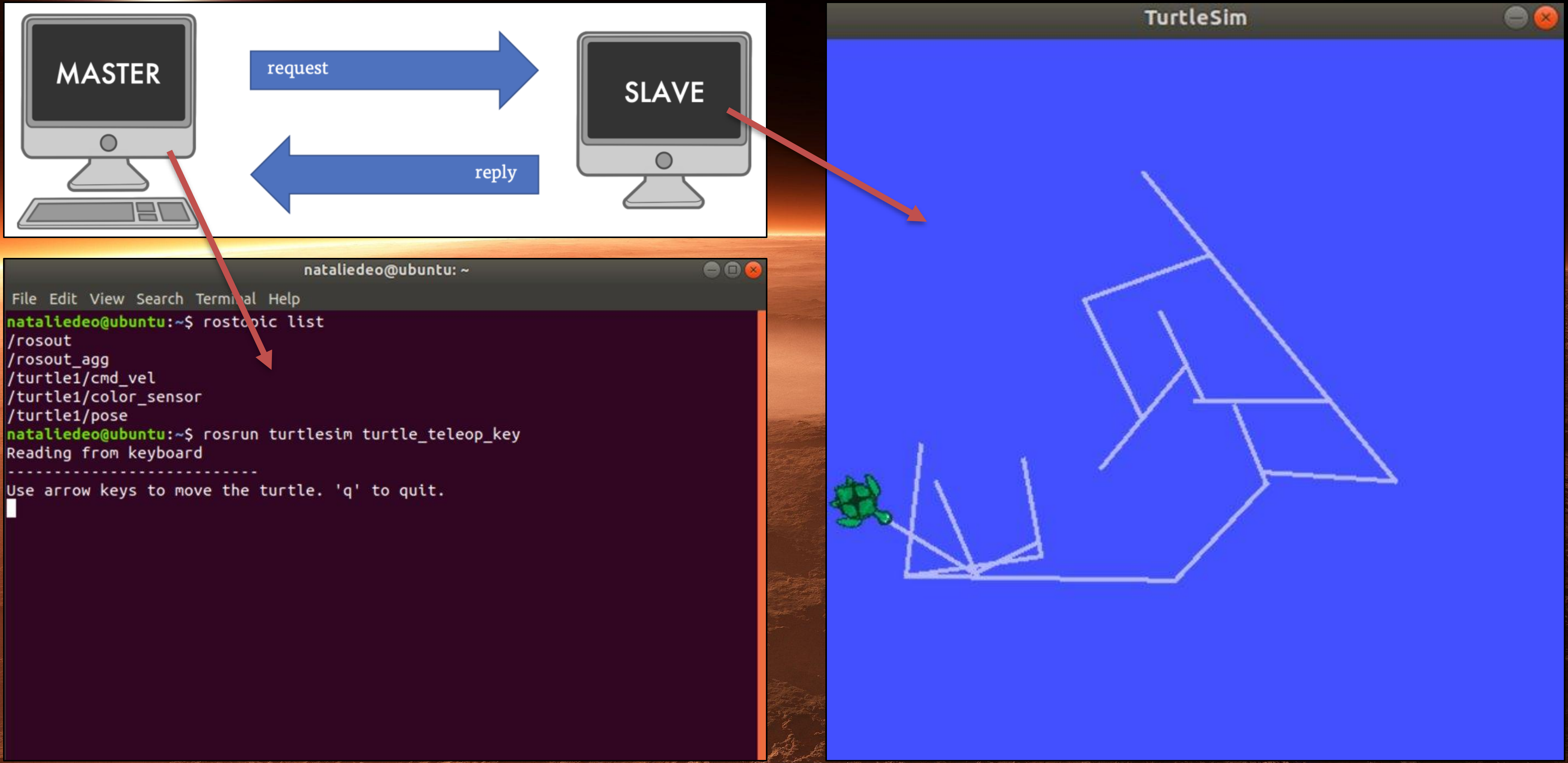


Figure 20: Information Path

MASTER-SLAVE COMMUNICATION

Figure 21: Master-Slave Communication



SOFTWARE: IMAGE RECOGNITION

General Design → Drive System → Power → Comm. System → Programming → Image Recognition

SW IMAGE RECOGNITION GANTT CHART

Table 12: Gantt Chart for Software Image Recognition Sub-team

Software: Image Recognition		Aldrin | Frank | Gabe | Nat | Pak			
	ComputerVision to ROS Integration	Aldrin | Pak	50%	11/7/20	TBD
	Kinect Camera Integration	Frank	0%	TBD	TBD
	Sending Camera Feed to Master	Nat | Gabe	100%	2/1/21	3/1/21
	YOLO V4	Aldrin | Pak	100%	10/20/20	1/8/21

YOLO V4

Best Model

- 50 images trained off 1 AR Tag

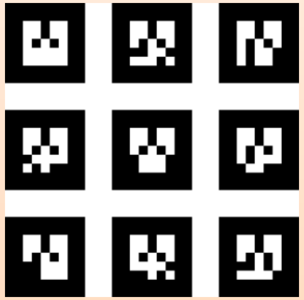


Figure 22: AR Tag

Model used Open CV & Python 3 to detect and output the location in pixels of the object.

OUTPUT: B&W when detected image is centered

AR tags are from URC rules and guidelines

```python
            lib_darknet_path='libdarknet.so',
            batch_size=1,
            gpu_id=0)

    while(True):
        ret_val, frame = cam.read()#cam.read() returns ret (0/1 if the camera is working) and img,
#the actual image of the camera in a numpy array
        display = frame #save the img as a variable for displaying later

        img = cv2_to_pil(frame) #convert the image to PIL so you can use it that way.
        img_arr = np.array(img.resize((d.network_width(), d.network_height()))) #resize PIL image to match cfg files for darknet weights

        detections = d.perform_detect(image_path_or_buf = img_arr, show_image=False) #returns values for detected objects
        height,width = display.shape[:2] #returns frame width and height
        # print(f'{detection.class_name.ljust(10)} | {detection.class_confidence * 100:.1f} % | {box}')

        center_x_arr = [] #resets array
        points = [] #resets array

        #finds bounding box locations and marks cv
        for detection in detections:
            box = detection.left_x, detection.top_y, detection.width, detection.height
            pt1,pt2,center_x,center_y = cv_box_values(width,height,box)
            center_x_arr.append(center_x)
            cv2.rectangle(display,pt1, pt2, [0,0, 255], 2)
            points.append([pt1,pt2])

        center_x_arr.sort()
        if center_x_arr != []:print((center_x_arr))
        for i in center_x_arr:
            if (i > int(width*3/8)) and (i < int(width*5/8)):
                #greyscale display
                display = cv2.cvtColor(display,cv2.COLOR_BGR2GRAY)
                break

        cv2.imshow('AR_Detector',display)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

```
[325, 504, 505]
[239, 325, 502, 503]
[241, 323, 376, 376, 502, 502]
[236, 238, 373, 374, 374, 501, 503, 505]
[236, 241, 248, 321, 372, 374, 376, 501, 502, 506]
[241, 248, 373, 375, 378, 504, 506]
[245, 249, 375, 376, 379, 504, 506, 508]
[248, 250, 380, 381, 382, 506, 509, 510]
[249, 252, 383, 508, 511]
[250, 253, 386, 509, 511]
[252, 327]
Segmentation fault (core dumped)
aldrin@aldrin:~/python-darknet-yolo-v4$
```

Video 1: Test1_Single AR Tag

ASSESSMENT & SUMMARY

COST AND WEIGHT

TOTAL ESTIMATED COST

$2,448.08

TOTAL ESTIMATED WEIGHT

~85 LBS.

CURRENT ROVER RENDER

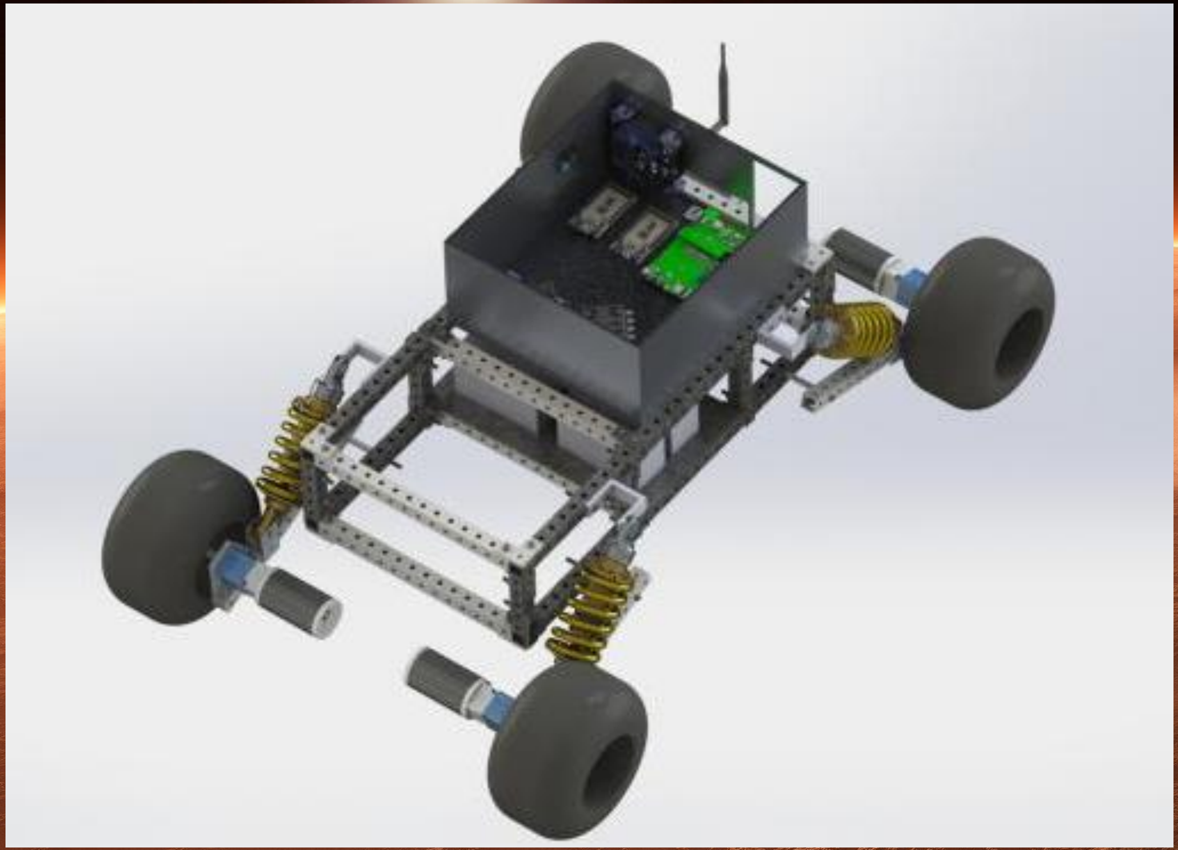


Figure 23: Isometric View of Rover

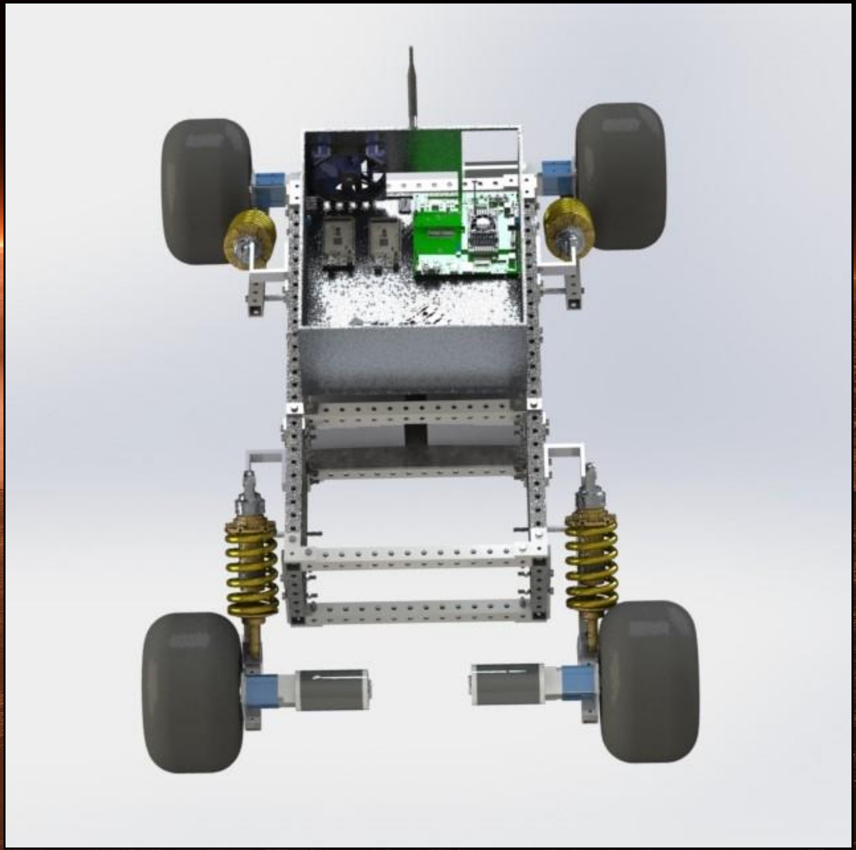


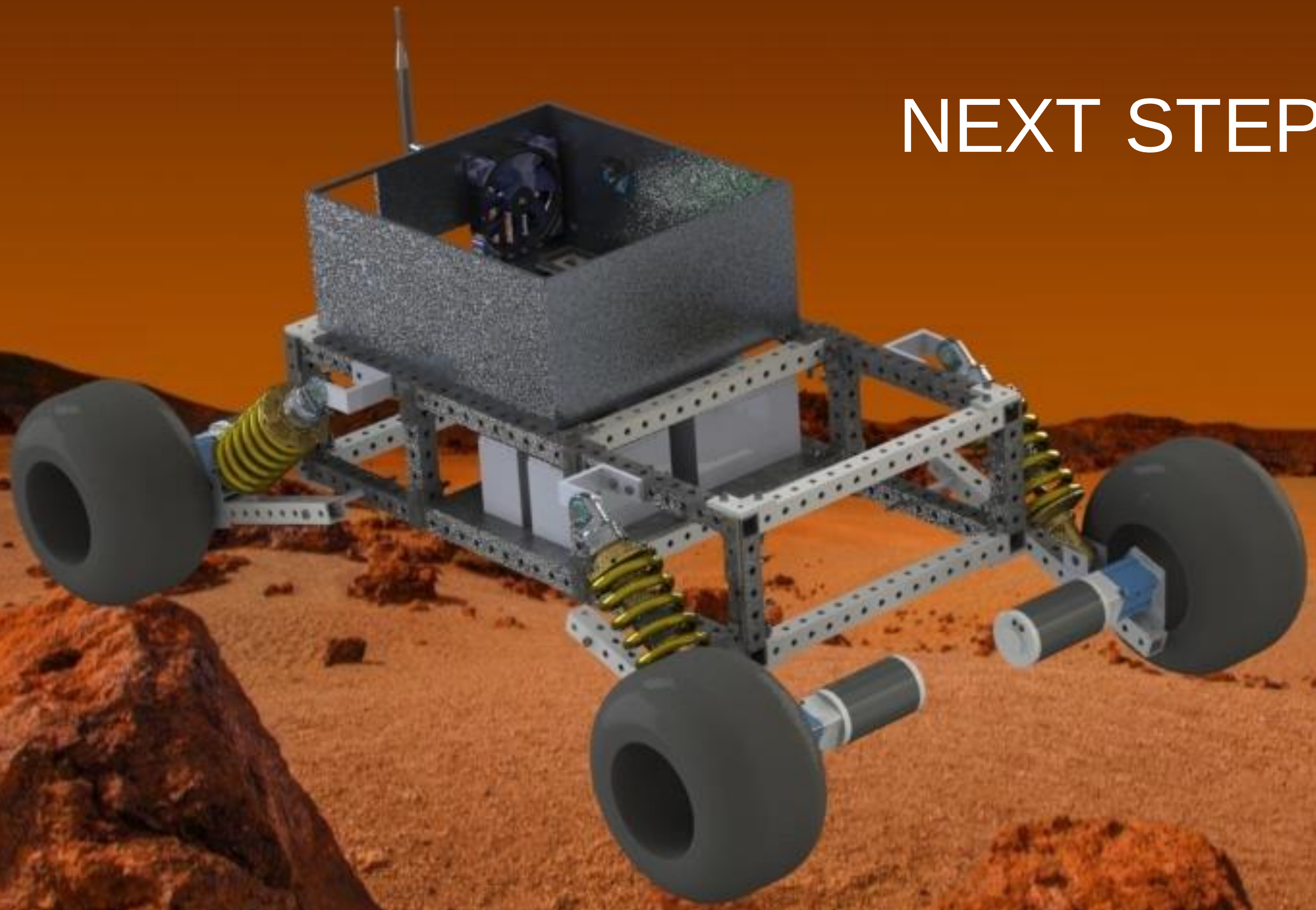
Figure 24: Top View of Rover

SUMMARY

- This project was prepared and organized to meet the standards of URC guidelines.
- The Mechanical Team completed its designs of the general design, suspension and frame, and drive systems.
- The Electrical Team formulated designs for the power and communication systems to allow for the rover to fulfill the mission.
- The Software Team tested and developed programming within ROS that will allow the rover to carry out the remote control and autonomous navigation missions.

URC

NEXT STEPS



PLAN FOR TRANSITION

All physical parts will be handed to Dr. He Shen to continue working with ASME and future senior design teams to further and finalize the design.

All software documentation and scripts are saved in GitHub with ownership transferred to Dr. He Shen.

QUESTIONS?

+ Huge thanks to Dr. He Shen & Dr. Mike Thorburn!