# 2026 Lunabotics Presentation

**2025-2026 Senior Projects team:**

-Justin Stahmer

-Braulio Caraveo

-Bryce Brumfield

-Francisco Garcia

-Noah Grove

-Juan Sanchez

-Alexander Canga

-Zoey Prevost

-Matthew Garcia

-Cruz Reyna

-Brandon Rodriguez

# Team Responsibilities

**Management**

- Justin Stahmer (Team Lead)

**Mechanical Group**

- Francisco Garcia
- Noah Grove
- Juan Sanchez
- Brandon Rodriguez

**Electrical Group**

- Matthew Garcia
- Cruz Reyna

**Software Group**

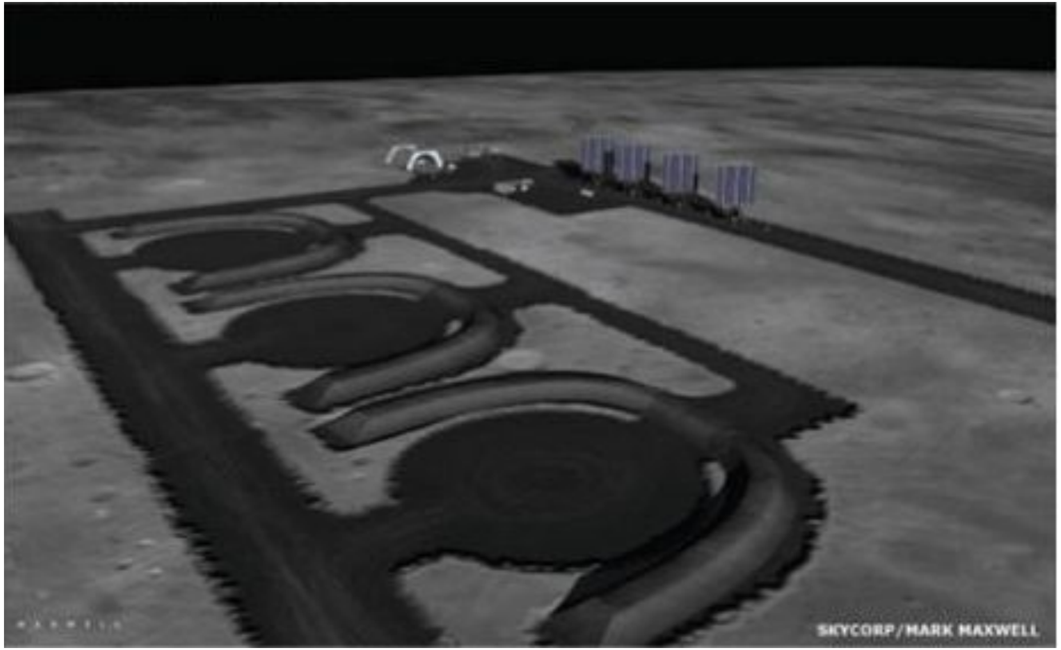
- Bryce Brumfield
- Braulio Caraveo
- Matthew Garcia
- Zoey Prevost
- Alexander Canga

# Management Overview

Justin Stahmer

# Introduction

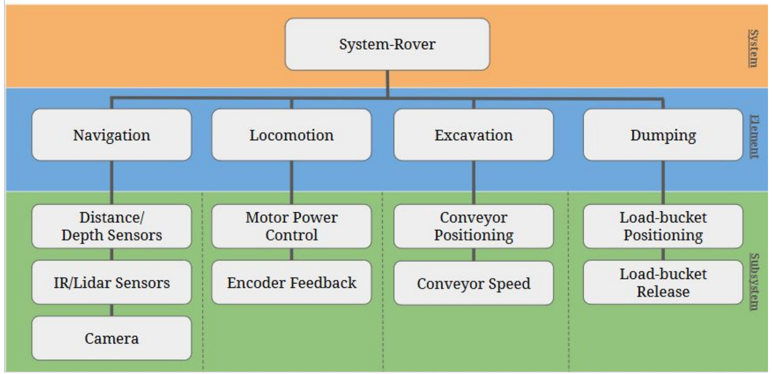
- Summary
- NASA Requirements
  - PMP- Project Management Plan
  - SRR - System Requirements Report
  - PP - Project Proposal
- Benefit to UHCL



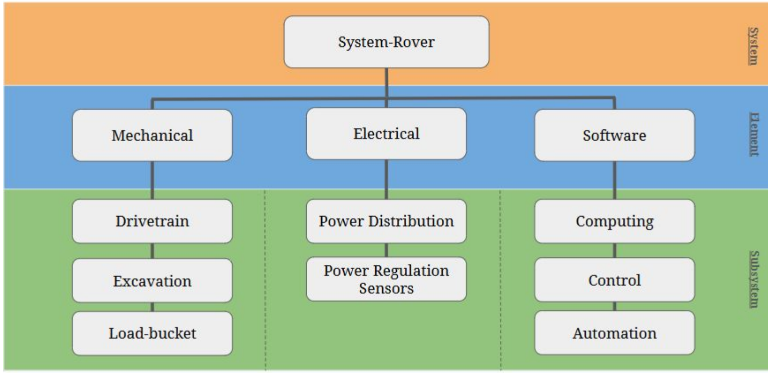
# Division of System Hierarchy

- Compliance with NASA Documentation for Project Development
- Useful for prioritizing objectives by team
- Allows for multiple aspects of the rober to be developed simultaneously

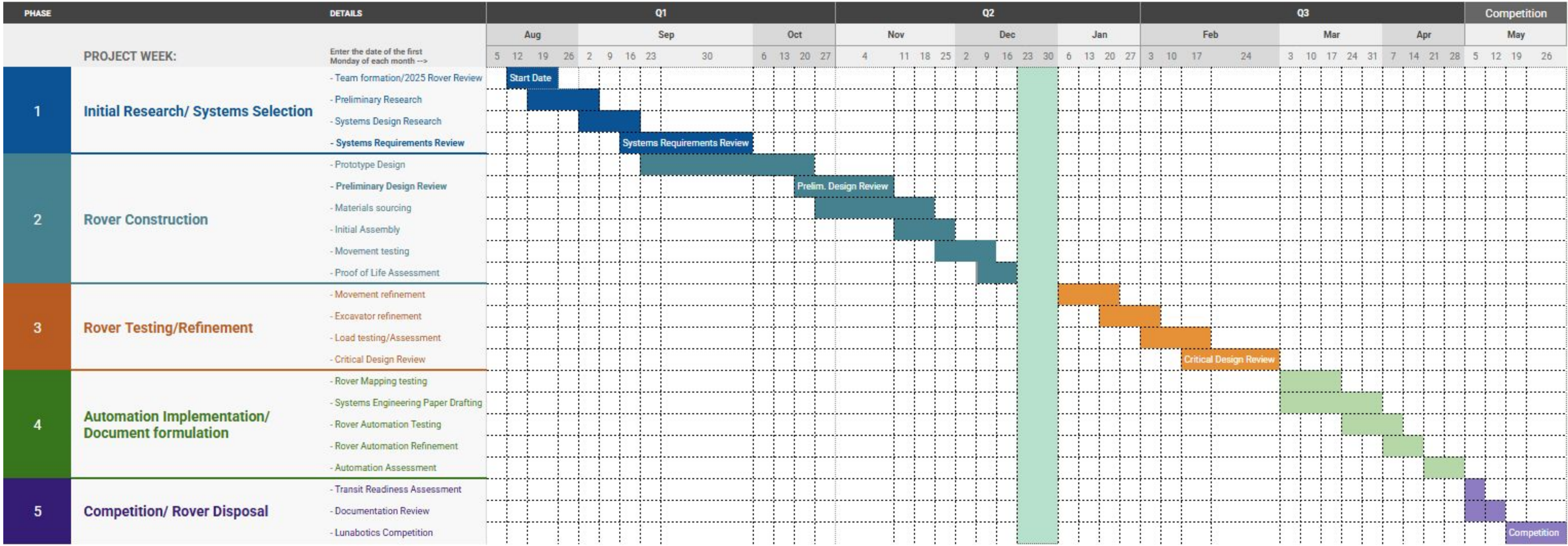
System Hierarchy (Internal/Software)




System Hierarchy (External/Hardware)



# Project Timeline



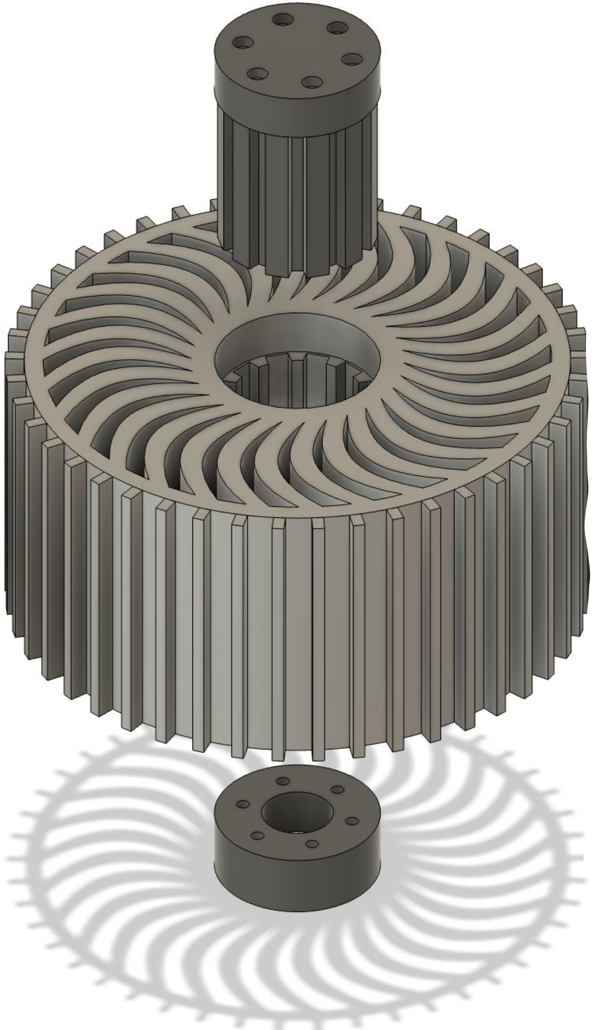
# Mechanical Design & Assembly Overview

Francisco Garcia

Noah Grove

Juan Sanchez

Brandon Rodriguez

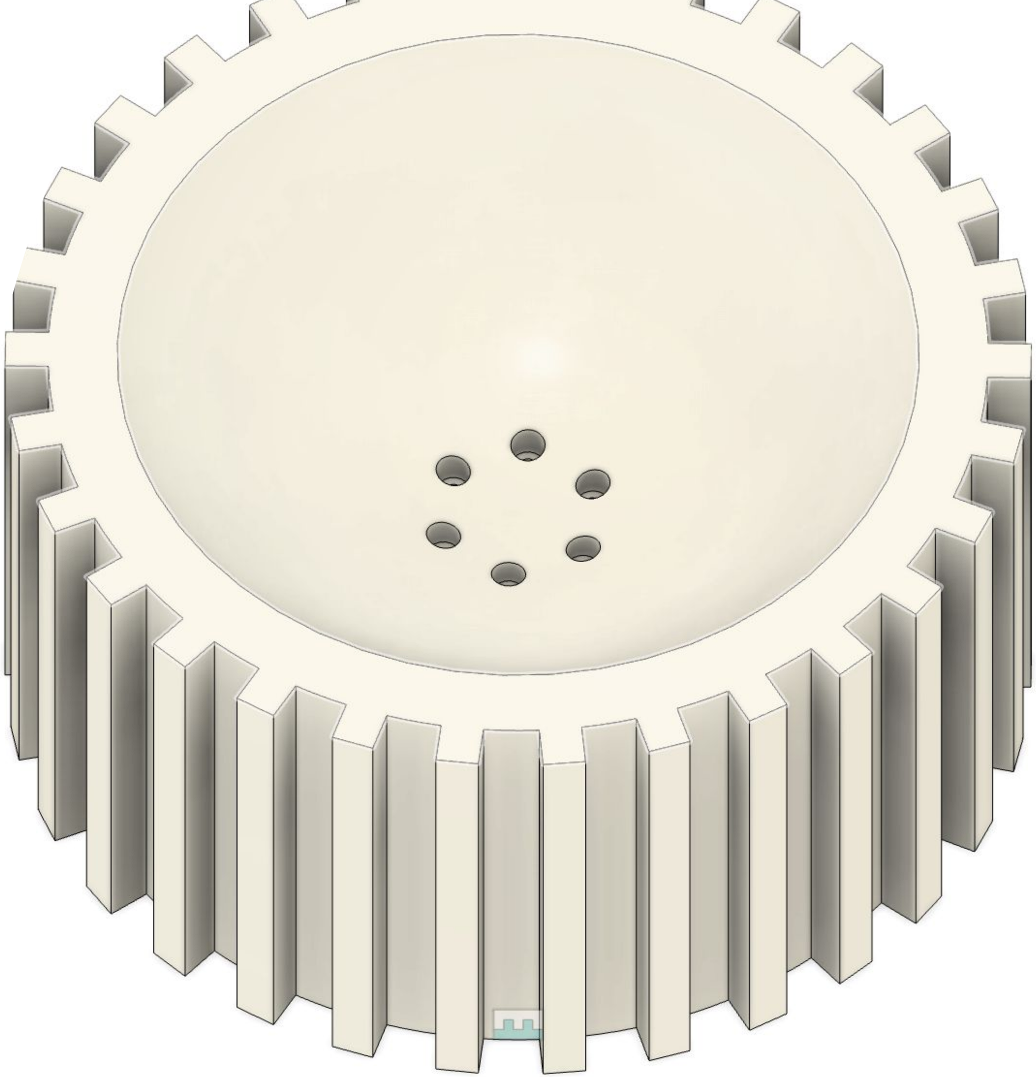


# Wheel Design

- Previous semester's prototype wheel.
- 250mm diameter
- Constructed from TPU and PETG

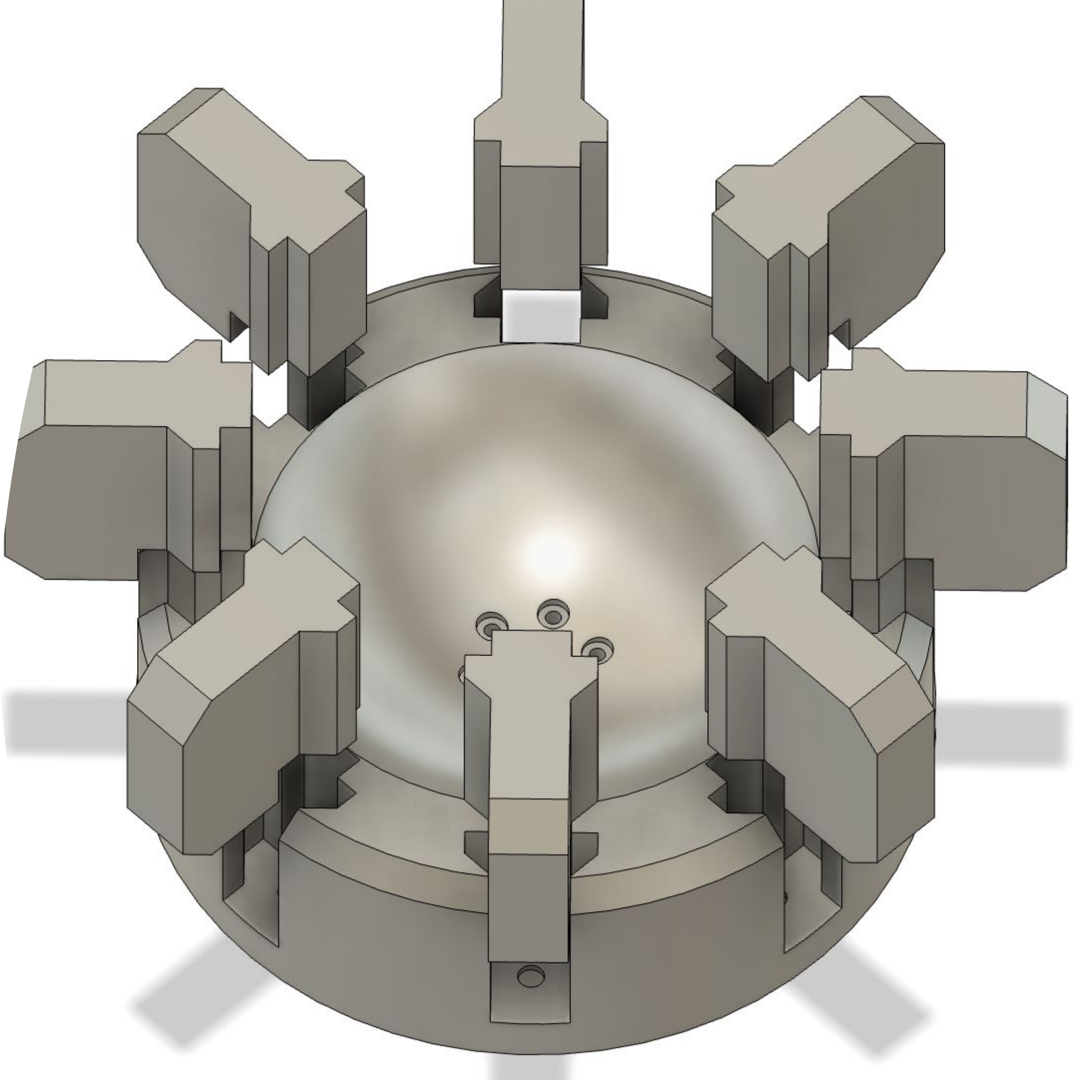
# Wheel Design

- 1st Prototype
- Constructed completely of PETG
- Larger Diameter – 300 mm in total



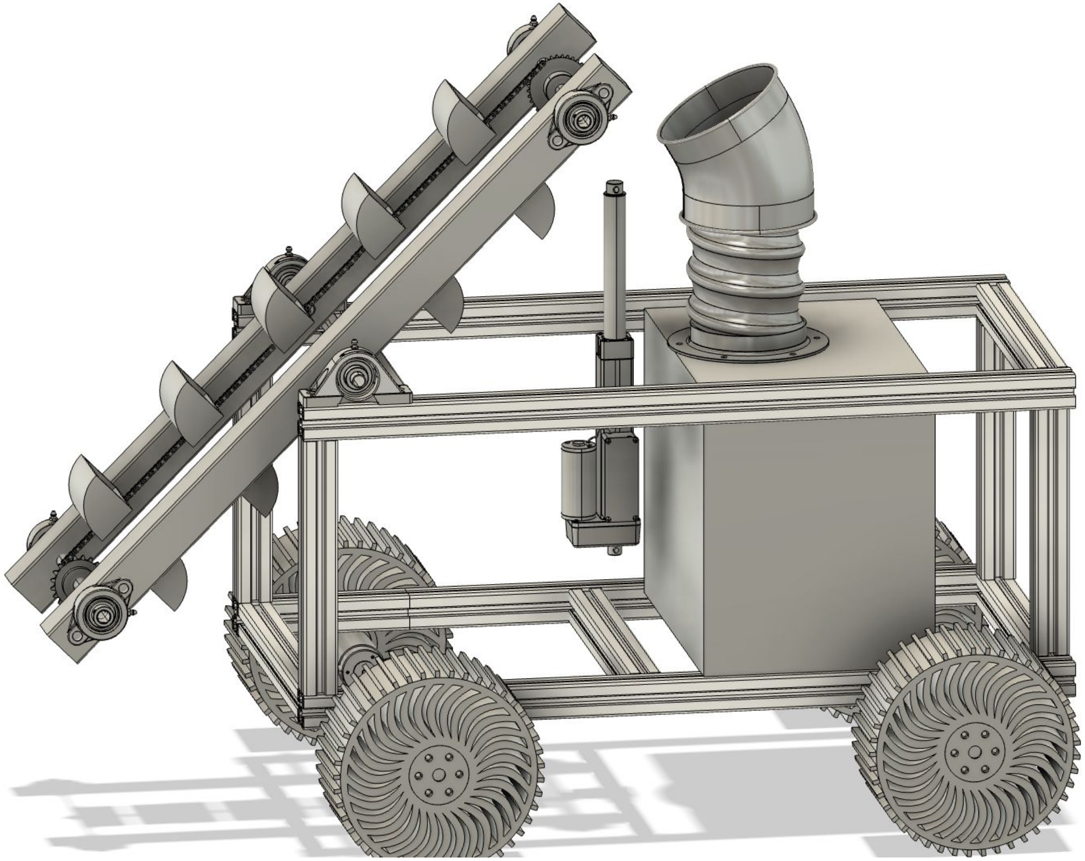
# Wheel Design

- Current wheel design
- Larger Diameter wheel hub
- Modular Fin design.
- Wheel hub diameter is 300mm
- Fins add 50mm each



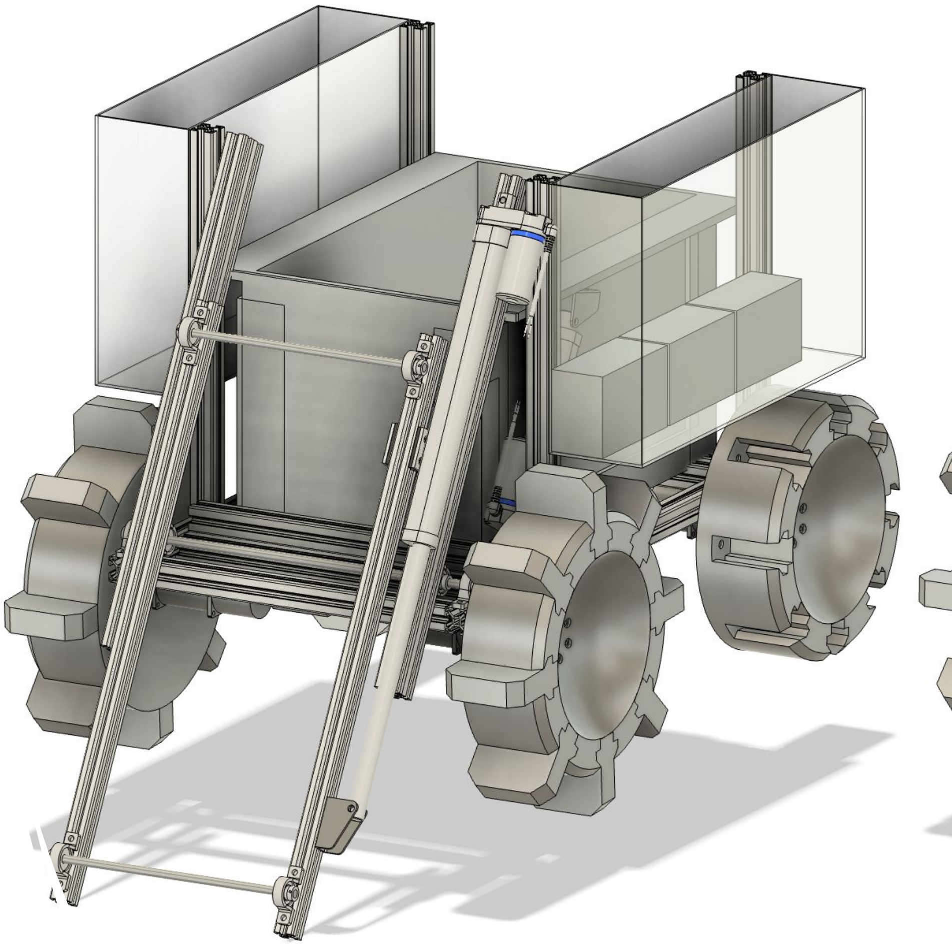
# Frame Design

- Previous Semester Frame design
- Frame is 1 meter x 480 mm x 380mm
- Uses 4040 aluminum Extrusions



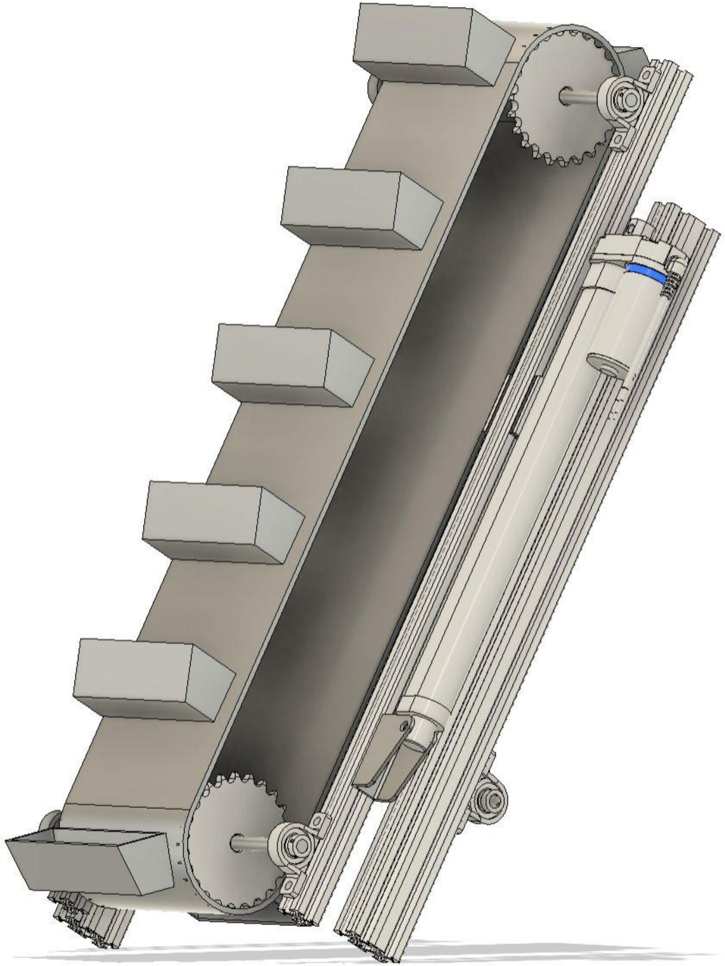
# Frame Design

- Current Frame design
- More square design
- Lighter
- Uses 3030 aluminum extrusions
- 700mm x 460mm x 510mm for just the frame
- Current Robot Dimensions are roughly 665 mm x 900mm x 725 mm



# Excavator Design

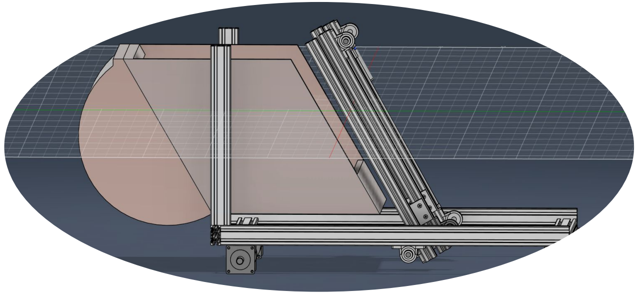
- Designed to move linearly, to dig for depth
- Common bucket ladder design
- Uses linear actuator and linear bearings to move
- Lighter

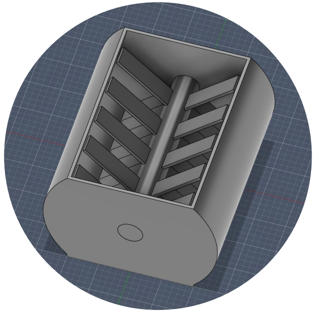


# Dumping Designs

- 3 Initial Designs
  - Dump truck
  - Trap door
  - Rotary door
- Dump truck design was chosen
- Changed from more traditional looking to a boxy shape
- 3D Printed Braces holding Acrylic walls and bottom

## Early Designs

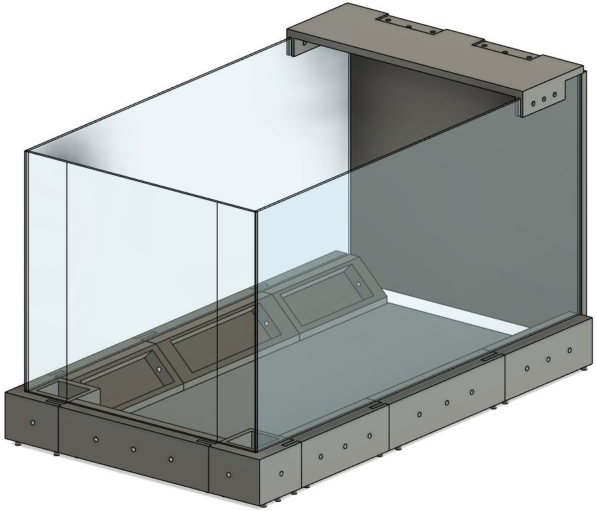




## Proposed Design

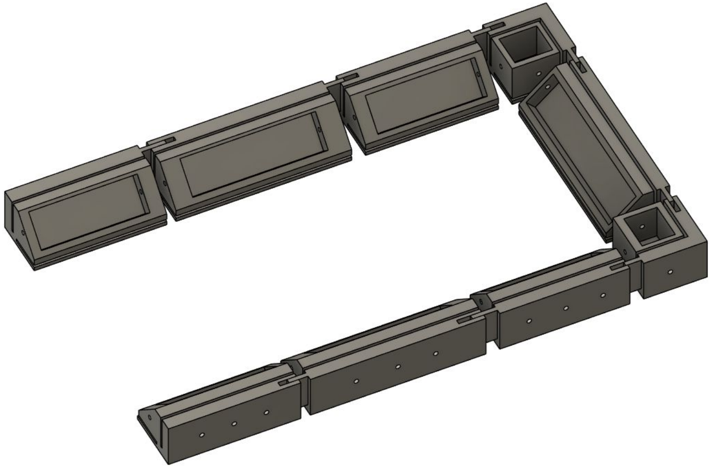
# Dumping Bucket

- Designed to maximise regolith carried for each trip
- Regolith will enter at the top, exit through the door
- Bucket will be tilted using a linear actuator at the front
- Future Plans: add shroud to limit spillage



# Bucket Frame

- Bucket consists of a 3D-printed frame holding Acrylic boards
- Braces custom fit to lock Acrylic panels in place
- Acrylic and PETG cut down on weight as opposed to metal
- Transparent bucket walls for monitoring material flow and testing.



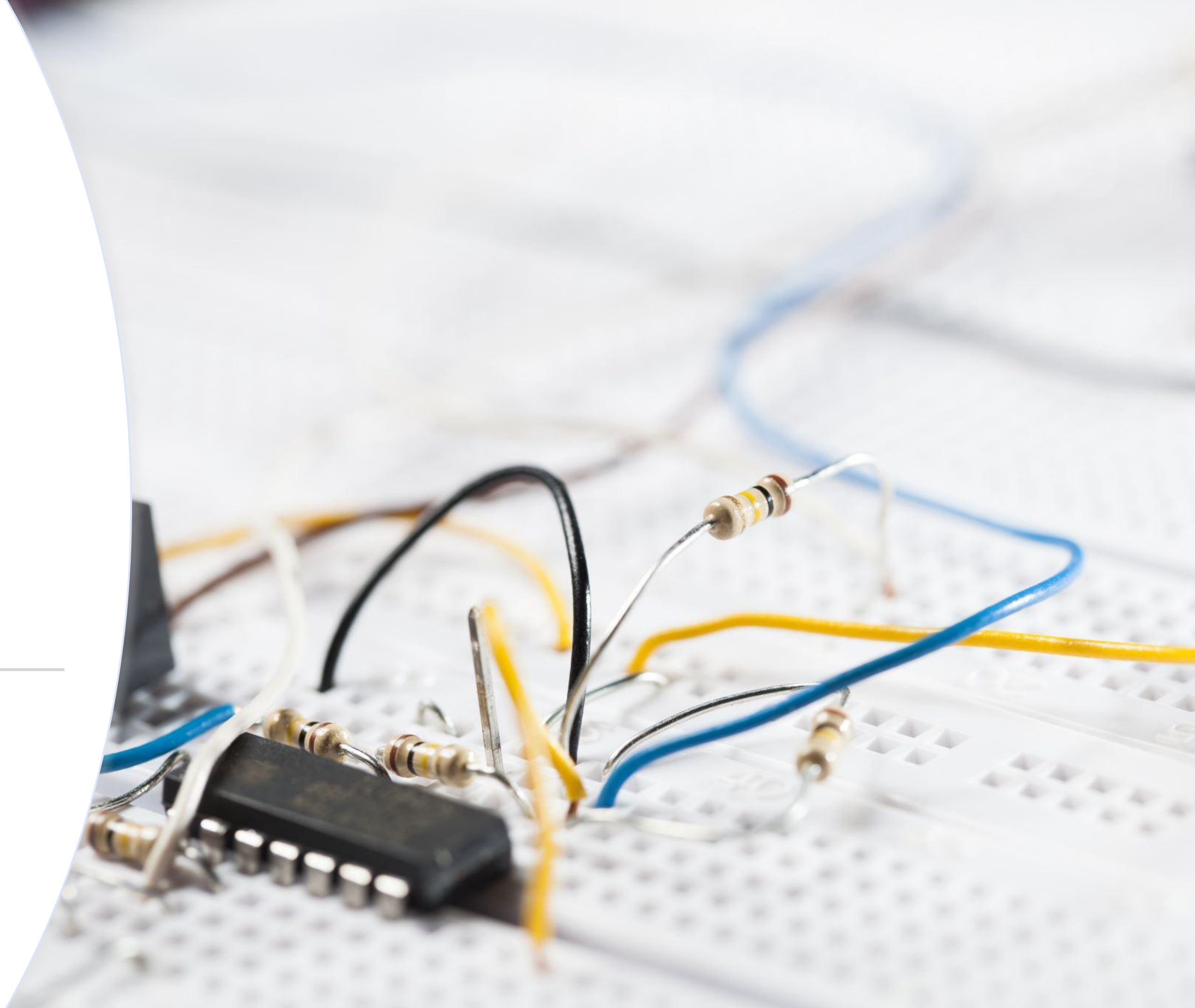
# Bucket Door

- Lightweight and efficient design
- 3D printed using PETG
- 2 Components, Top and Front
- Solenoid Latch keeps door locked
- Door Opens/Closes due to Bucket Tilt
- Next semester, plan to test Door Strength and Closing Mechanics

# Electrical and Power Systems Overview

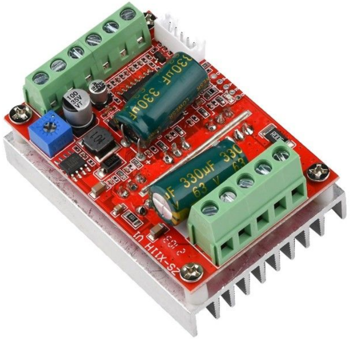
Matthew Garcia

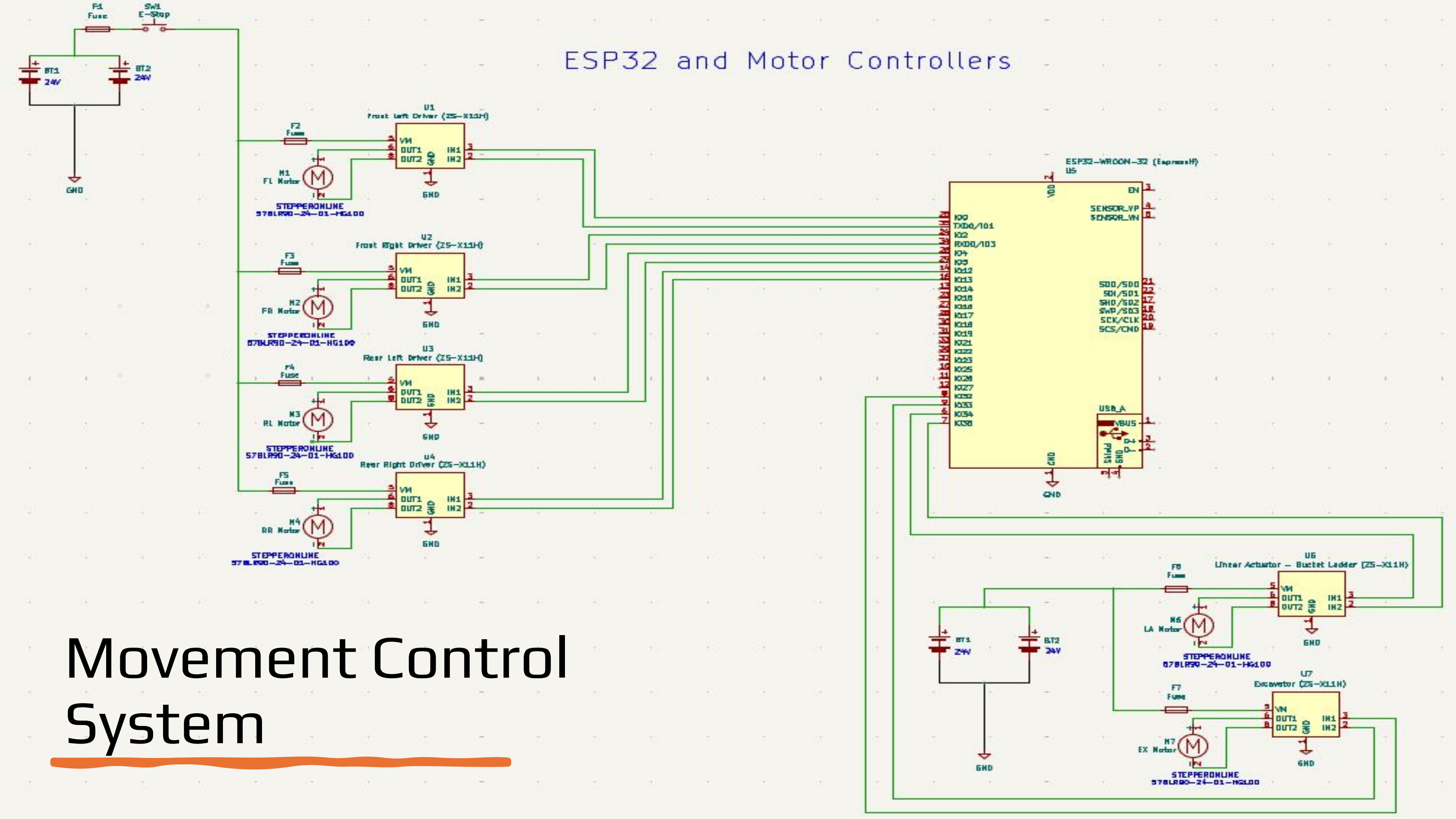
Cruz Reyna



# ESP32 Motor Controller & Power System

- 24V dual-battery system providing main drive power for all rover motors
- Each wheel motor is driven through a ZS-X11H motor controller
- Motor drivers for both linear actuators and excavator mechanisms
- Integrated fuses and grounding for system protection and power isolation

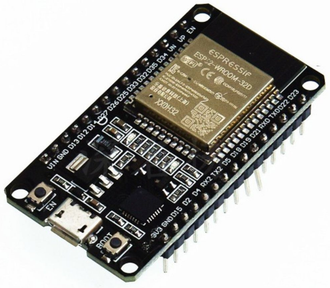



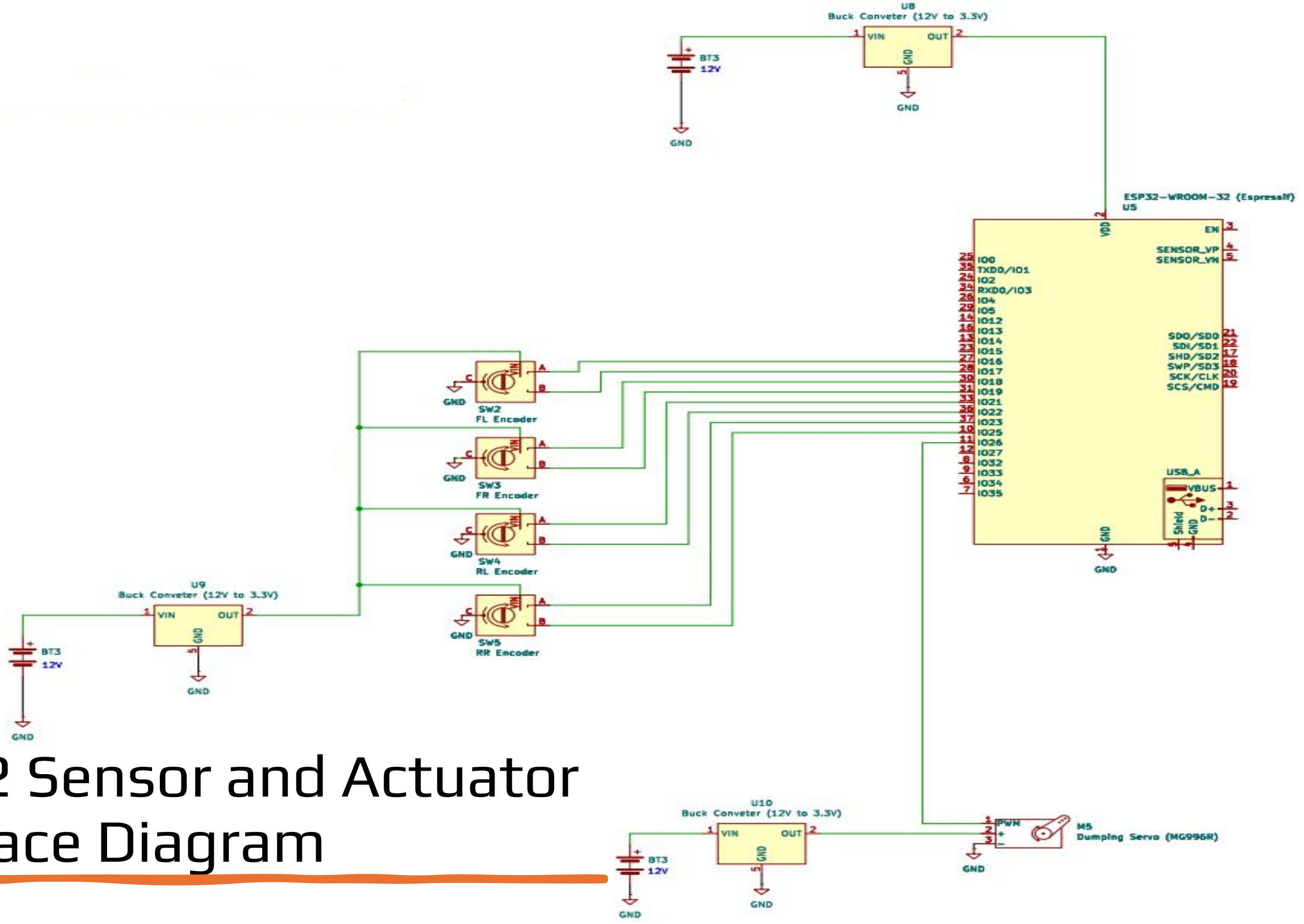



# Movement Control System

# Voltage Regulation and Sensor Interface

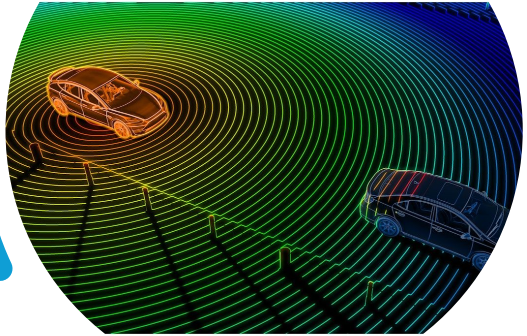
- Powered by a 12V battery system
- Buck converter to regulate 12V down to 3.3V to power the ESP32 control board
- AS5600 magnetic encoders provide accurate wheel position feedback
- Solenoid and linear actuator for dumping mechanisms

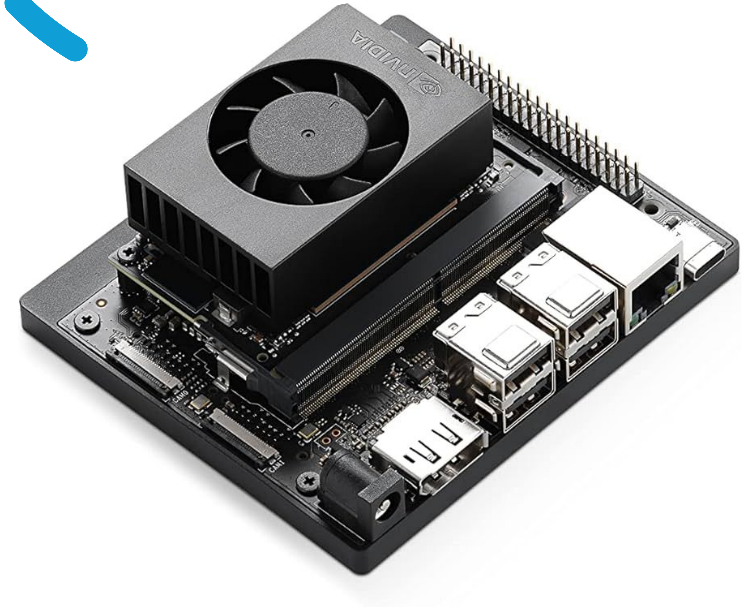




# ESP32 Sensor and Actuator Interface Diagram

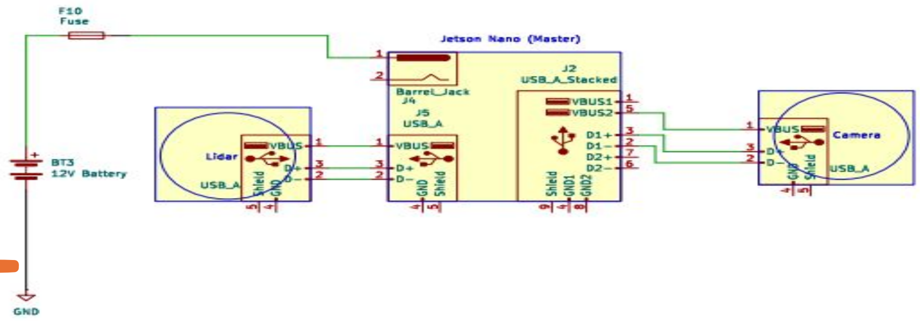
# Jetson Orin Nano Integration

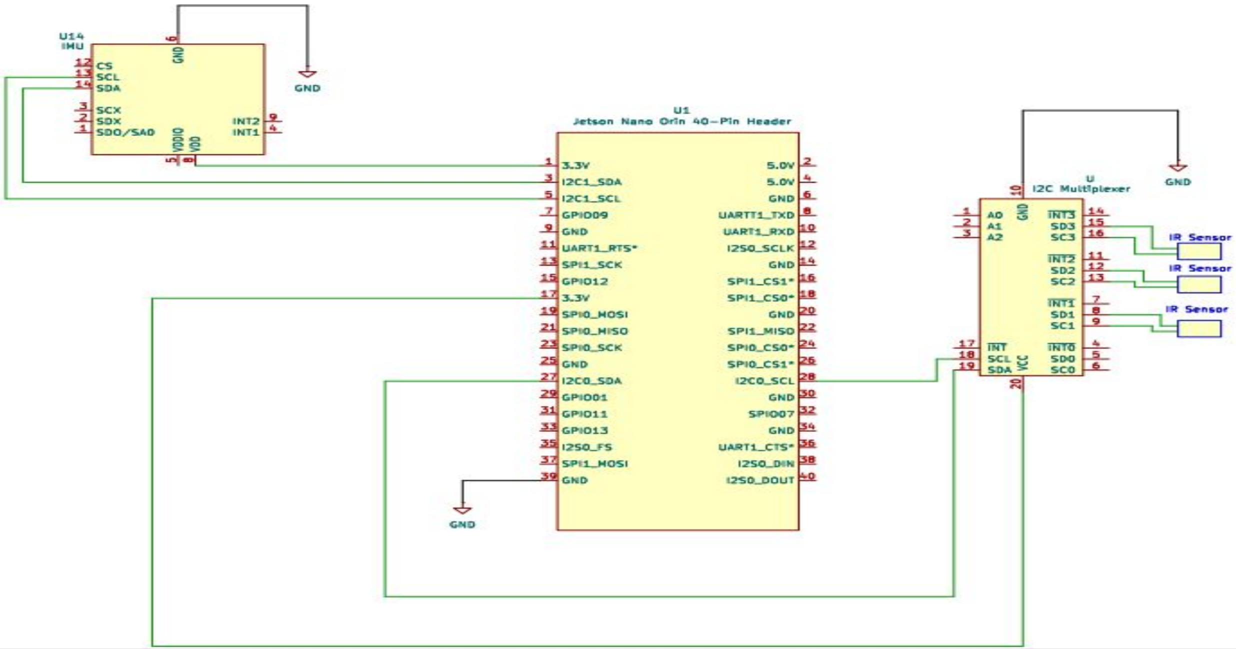


- Powered by a 12V battery supplying stable power for onboard computation and perception
- Serves as the master computer handling autonomy, perception, and high-level control
- Provides data to the ESP32 subsystem for low-level motor and actuator control
- LiDAR and 3D camera connected for real-time mapping and vision processing
- 40-pin GPIO interface for sensors such as the IMU and IR sensors

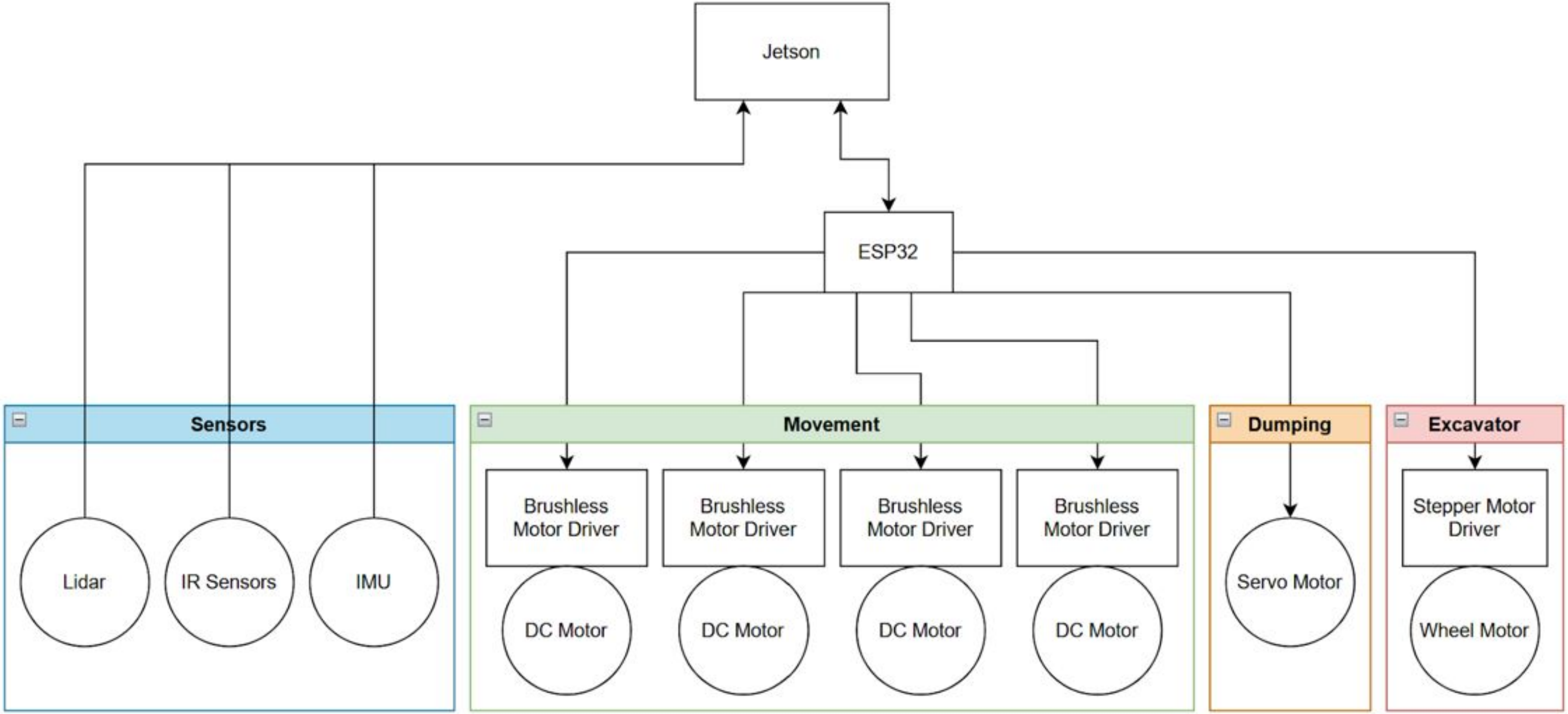


# Jetson Orin Nano and 40-Pin Header Connection



# Electrical System Flowchart



# Software Subsystems Overview


Bryce Brumfield

Braulio Caraveo

Matthew Garcia

Zoey Prevost

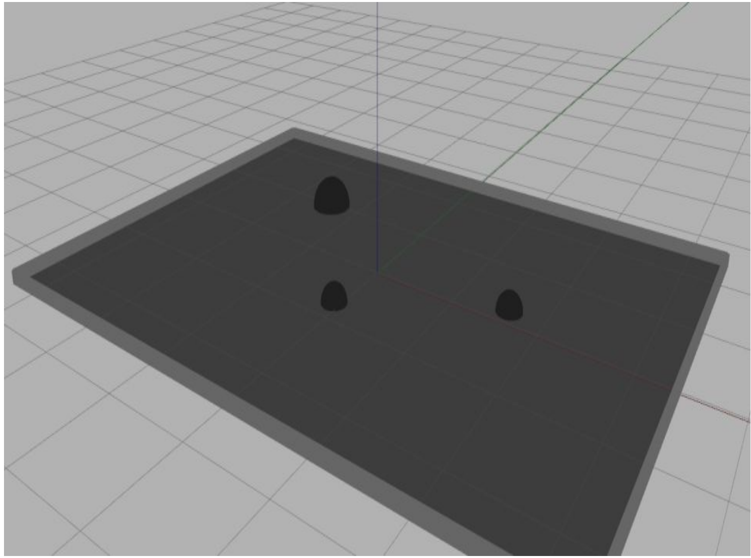
Alexander Canga


# Autonomous Navigation

- The autonomous navigation module enables the rover to travel through sensor data and dynamic path planning.
- Navigation integrates received data from the LiDAR, and infrared (IR) sensors to map surroundings in real time.
- This was implemented using ROS2, Nav2, SLAM, and MATLAB.

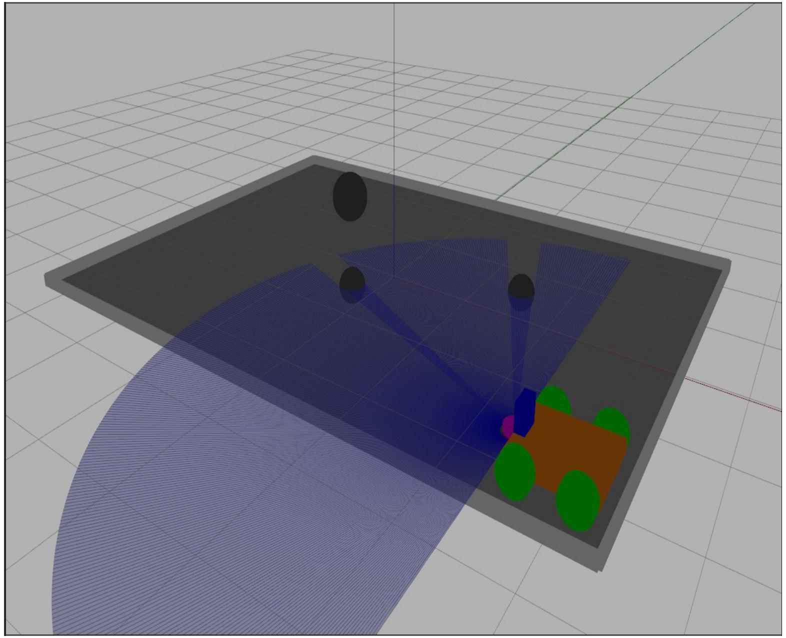
# Lunabotics Arena

- Lunabotics Arena was created as a dimensionally accurate representation of real competition environment we will operate rover in.
- It was initially designed in MATLAB but then converted to a usable SDF file for testing the rover within Gazebo.
- This arena is a world built based on real parameters and is used to test rover behavior in Gazebo with ROS2.

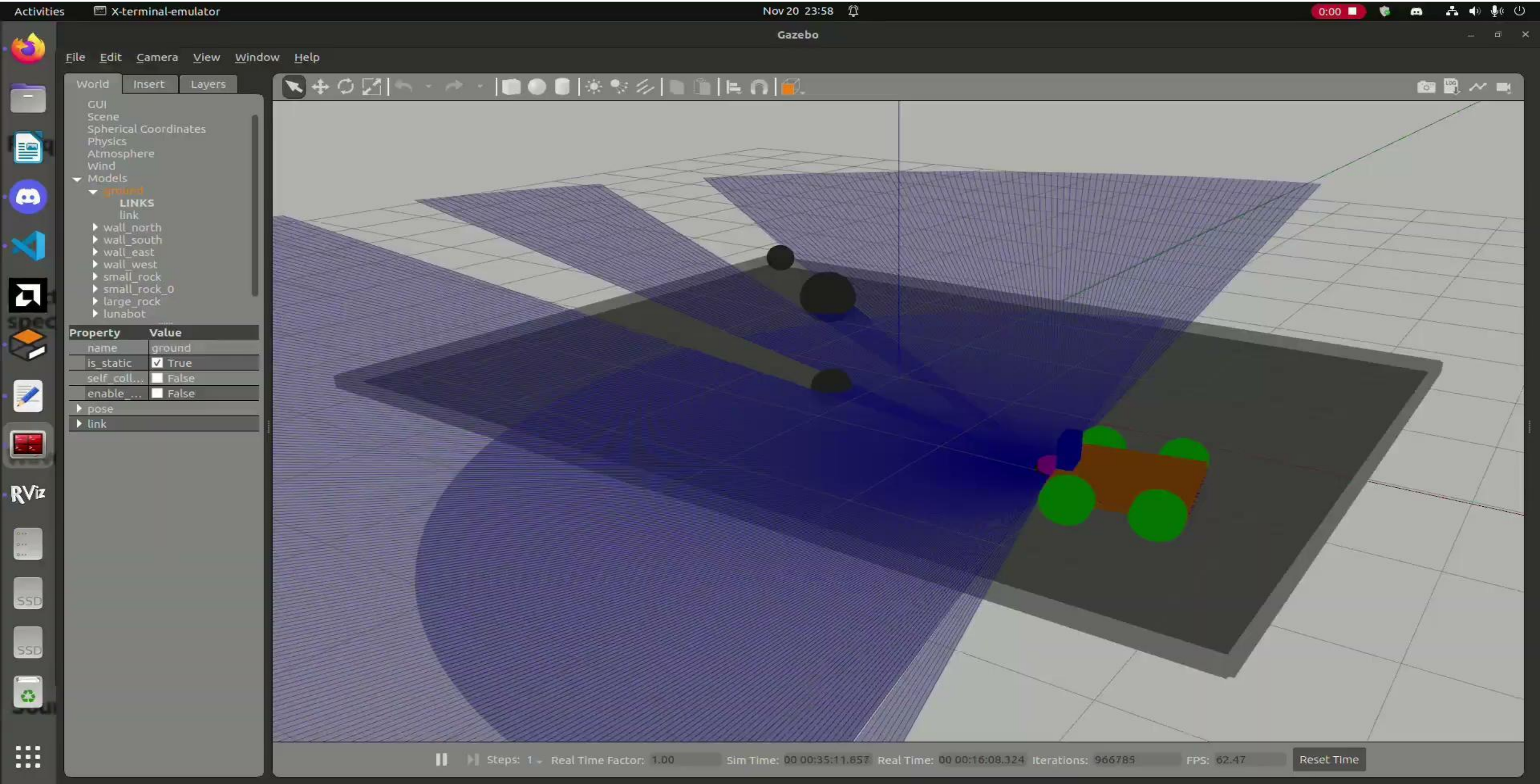


# Rover & Arena

- Autonomous navigation software will be developed along with goal markers inside of the arena to identify position identification.
- As shown in the picture, LiDAR functionality is implemented as well as object detection.
- Other detection that still needs to be implemented via IR sensor.
- Further improvements will be made on the 3D camera and general computer vision system.

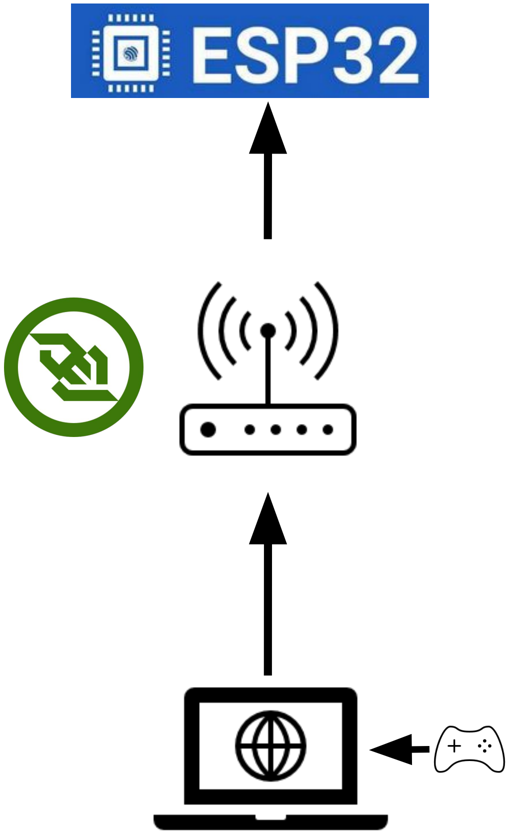


# Autonomous Navigation Demonstration



# Rover Control System

- We are using an ESP32 the WebSockets communications over local network.
- WebSockets enables control over wireless network.
- Computer converts controller input into motor PWM commands.
- Data was previously sent as simple words
- Method is flexible for controller mapping but undermines flexible speed control.

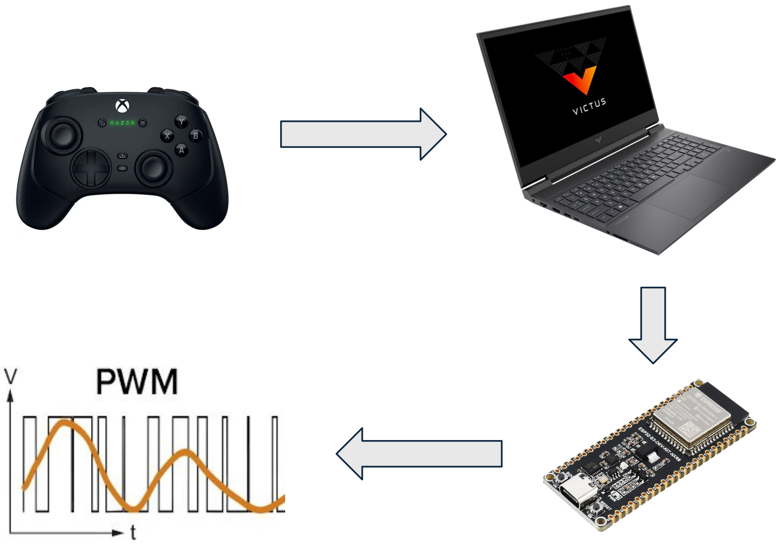


# Scalable Controller Input

- Allows for non-discrete velocity commands as well as proper in-place rotation.
- Using micro-Python for flexible programming and process streamlining.
- Limitations faced for excavation control, will require further changes and improvements
- Micro-ROS is the end goal once fully realized system has been achieved.

# Wireless Control Pipeline

- Implemented real-time rover control using WebSockets.
- Xbox controller input is converted into velocity commands on the laptop.
- Laptop sends commands wirelessly to the ESP32 over WebSocket.
- ESP32 converts velocity commands into PWM signals for smooth, responsive motor control.

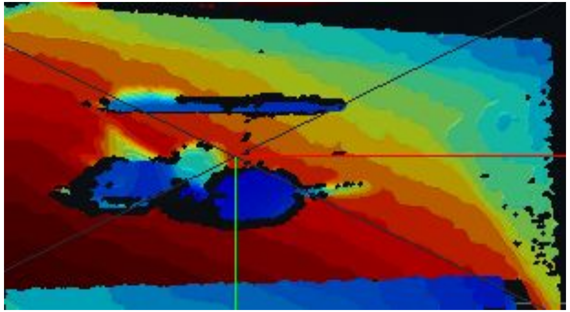


# Why ROS?

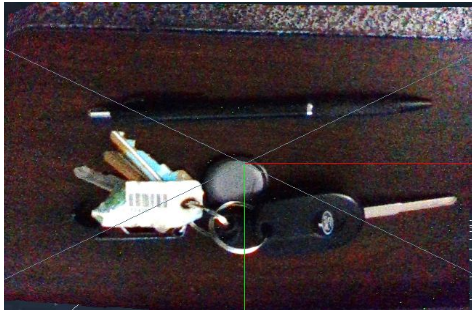
- The Robot Operating System, presently ROS2, is commonly used for robot software.
- Considering factors such as project timeframe, it is the best option.
- ROS2 is extensive, but its complexity would make hosting it onboard the rover impractical.
- Micro-ROS is designed for use in microcontrollers, able to communicate with a ROS2 instance which handles most of the computing.

# Computer Vision Overview

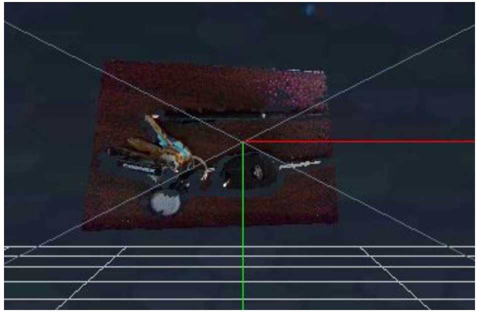
- Supports autonomous navigation and driver-assisting views
- Uses Intel RealSense D435 (depth + RGB) and two Logitech C270 webcams (RGB)
- Detects rocks, craters, and other obstacles
- Processes through OpenCV on the Jetson



Depth



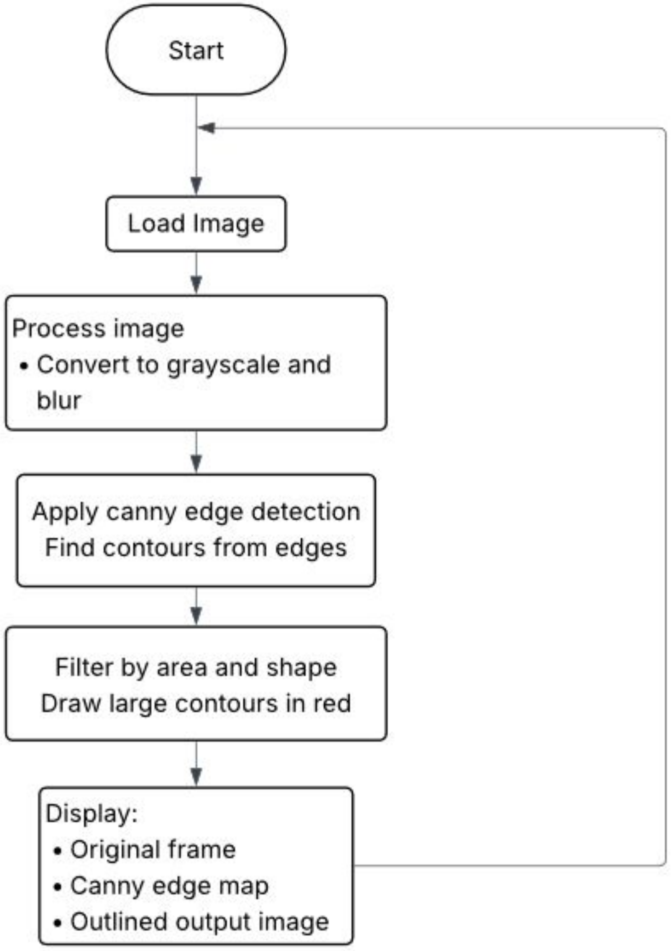
RGB



RGB AND Depth

# OpenCV

- Processes RGB feed from all cameras
- Outputs hazard detection and hazard visualization
- Used for autonomy and driver assistance



# OpenCV

**Input:** Photo of 2025 Lunabotics Arena

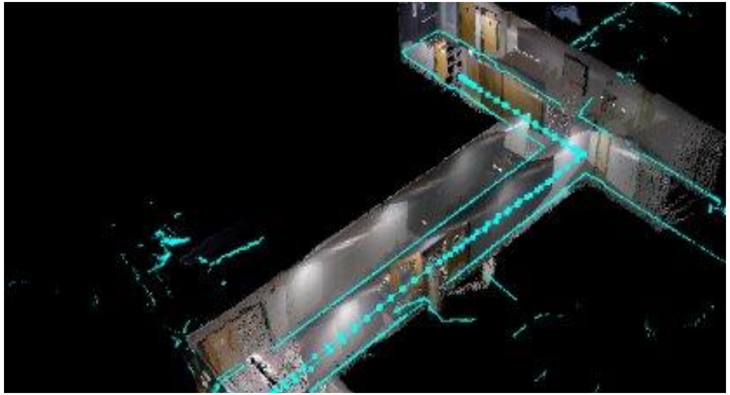
**Output:** All Edges Detected

**Output:** Edges Highlighted

# Other Image Processing

- **SLAM: Simultaneous Localization and Mapping**
- Uses RealSense depth data to build a 3D terrain map
- Tracks rover's real-time position on map
- Enables autonomous path planning and spatial awareness
- Implemented through ROS2

- **YOLOv8 Object Detection**
- Uses a pre trained AI model to recognize rocks, craters, and obstacles



**Example SLAM Map of Office**

"Create 2D map – Issue #1196," *IntelRealSense/realsense-ros*, GitHub, May 13 2020. [Online]. Available: https://github.com/IntelRealSense/realsense-ros/issues/1196. [Accessed: Nov. 07, 2025].

# Future Developments

- Implement Micro-ROS for communication between ESP32 and Jetson Orin.
- Develop velocity mapping from joystick inputs to linear and rotational motion as well as excavation control.
- Begin autonomous navigation logic using sensor data from LiDAR and IMU.
- Implement accurate camera stream for constant surveillance.