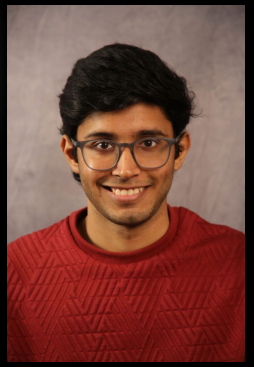
Lunar ROADSTER

(Robotic Operator for Autonomous Development of Surface Trails and Exploration Routes)

"Starting with a foothold on the Moon, we pave the way to the cosmos"

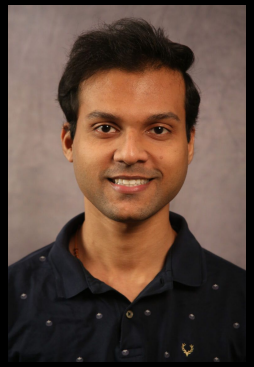
The Team

Ankit Aggarwal



Deepam Ameria

Bhaswanth Ayapilla

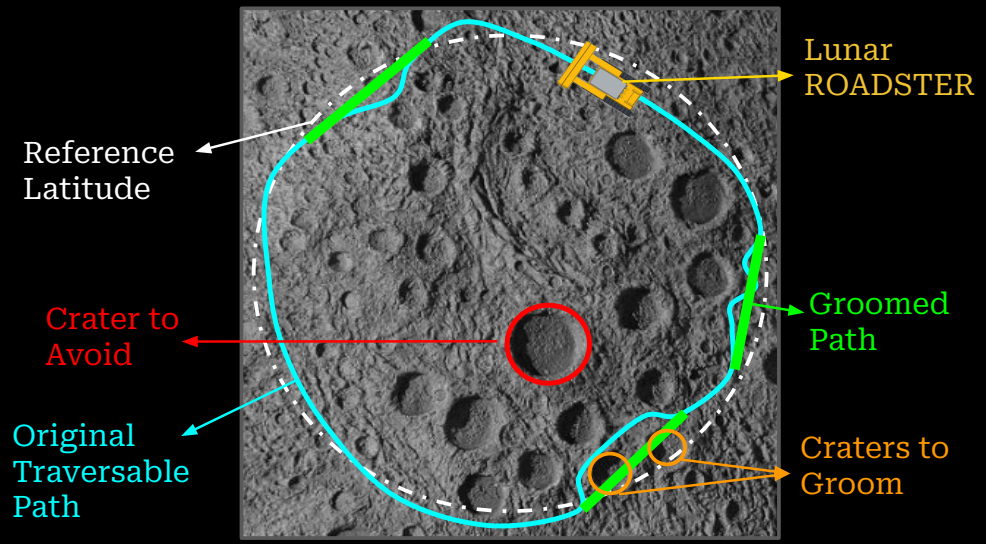


Simson D'Souza

Boxiang (William) Fu

Dr. William "Red" Whittaker

Project Overview: Lunar ROADSTER

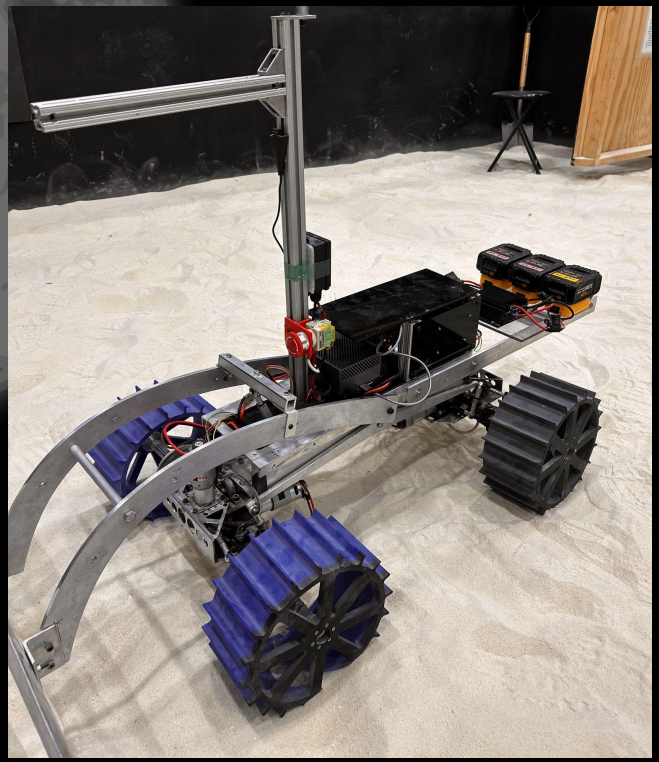


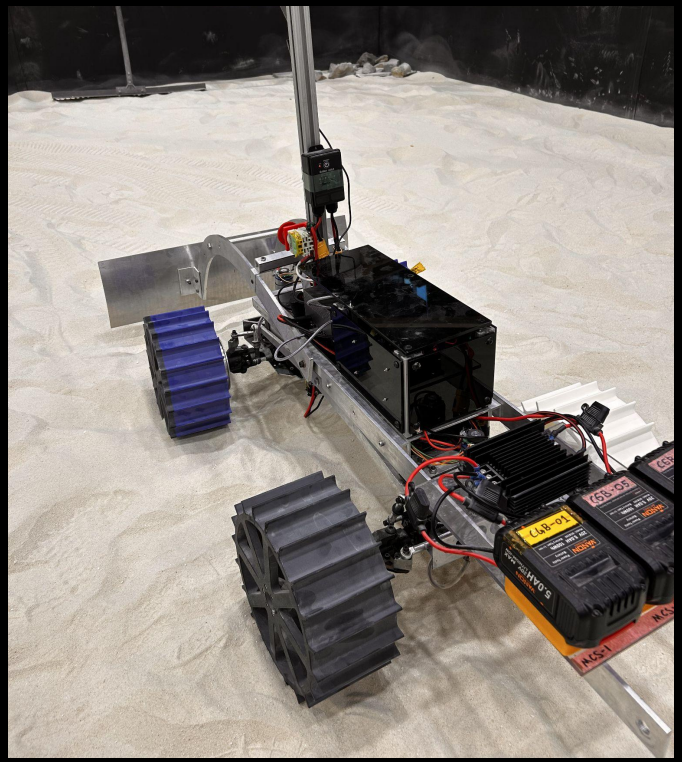
An autonomous moon-working rover capable of finding ideal exploration routes and creating traversable surface trails

Subsystem Status

Subsystem	Completion %	Future Work
Sensors	75%	Integrate New IMU and Fisheye Camera
Computations	60%	
1. Jetson and Docker	90%	Set Up New Sensor Drivers
2. Localization Unit	80%	Implement Skycam Localization
3. Transport Planner Unit	60%	Implement New Transport and Tool Planner
4. Navigation Planner Unit	40%	Generate Map and Implement Global and Local Navigation Planner
5. FSM Planner Unit	70%	Update FSM to Support New Modules
6. Validation Unit	10%	Detect Craters and Implement Validation module
External Infrastructure	90%	Implement New Total Station Localization Method
Mechanical	90%	Fabricate and Install Actuator Arch
Actuation	85%	Linear Actuator Upgrade/Tuning
Electrical Power	100%	None

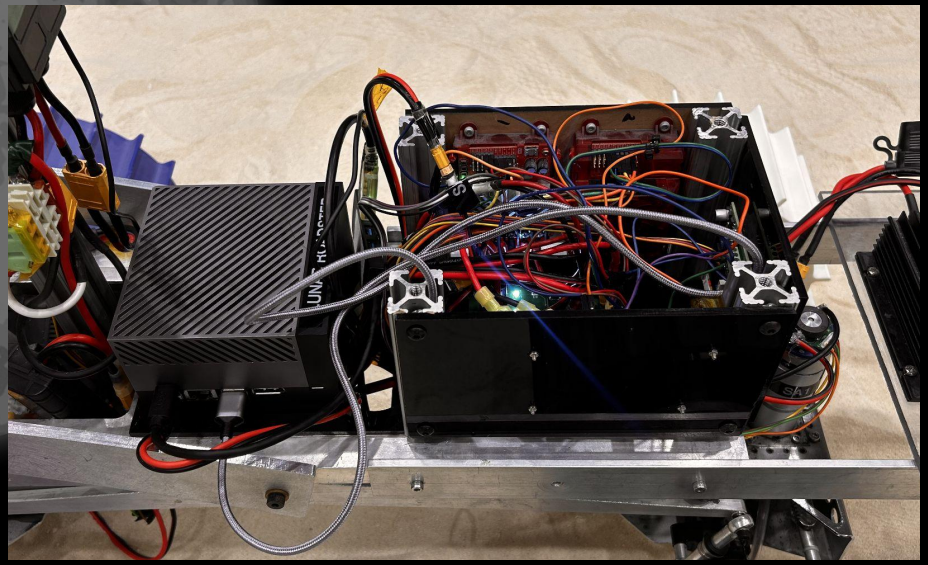
Hardware: Rover Hardware Maintenance

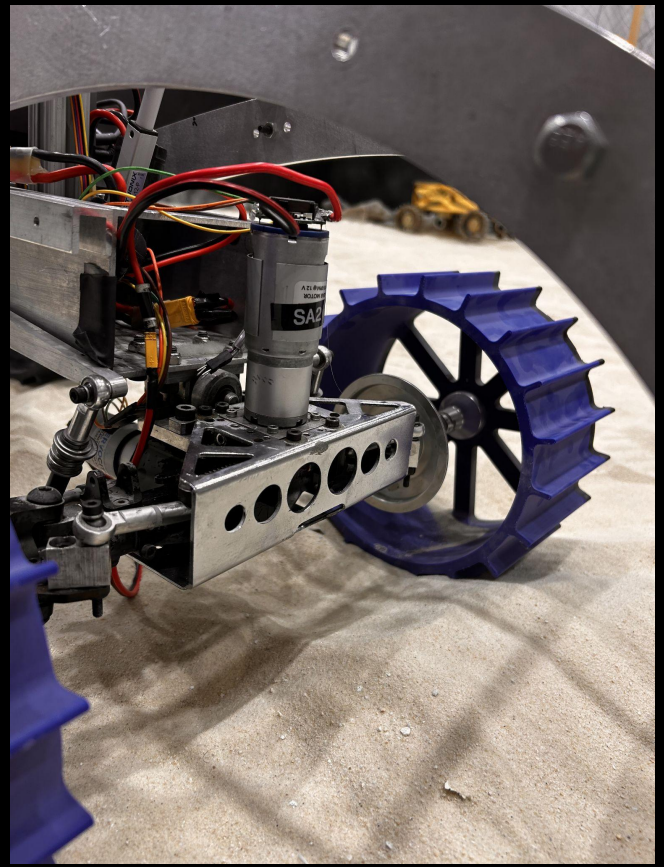




All components were checked for damages thoroughly and the rover was re-assembled

Hardware: Rover Wire Maintenance

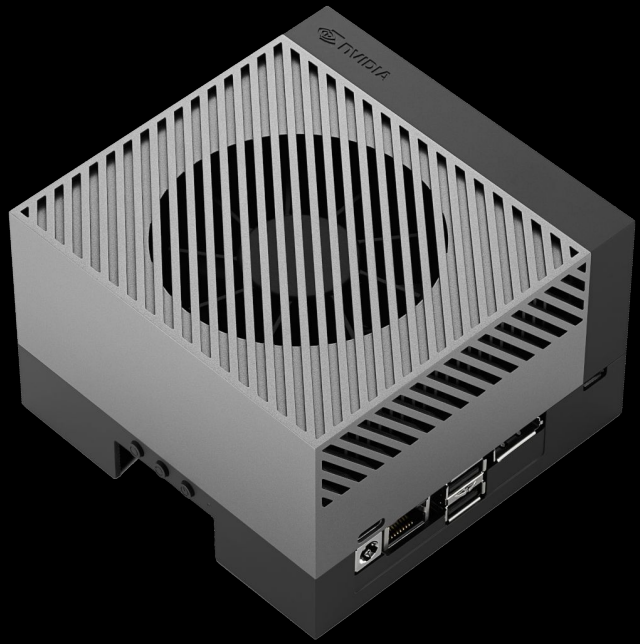




- Replaced all faulty and loose connections
- Re-soldered broken connections
- Cleaned up wire routing

Hardware: Upgrade Compute to Orin

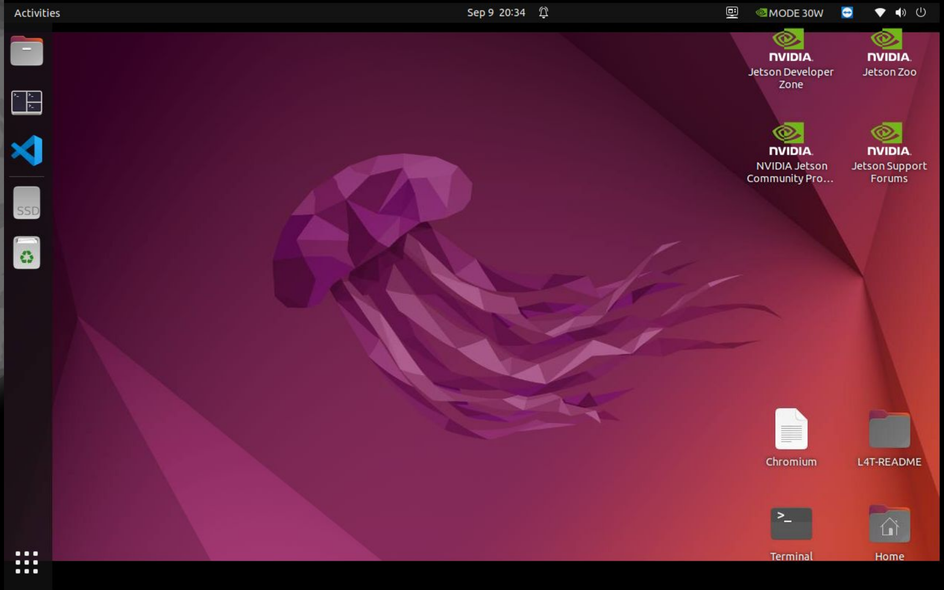
~4x CUDA cores
~1.7x faster CPU performance
~2x memory
~1.5x higher bandwidth

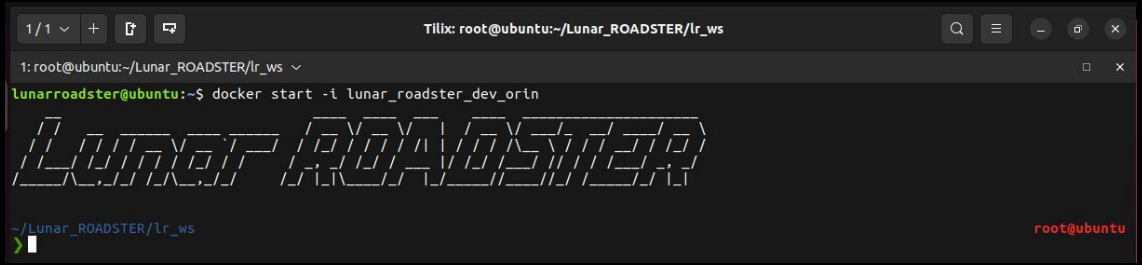


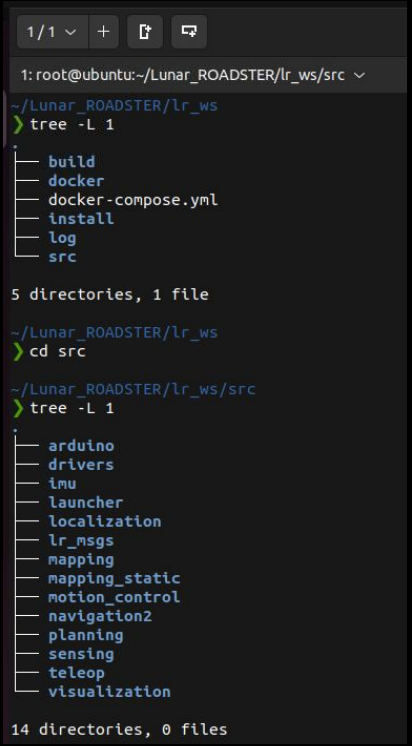
Jetson AGX Xavier

Jetson Orin

Software: Setup Docker on Orin







Software: Code Quality Compliance

- Standardized code namespace, header information, and package.xml metadata

```
/**
 * @file visualization.cpp
 * @brief Converts a global occupancy grid map into a GridMap representation for terrain visualization.
 *
 * This node listens to the global occupancy grid, converts it into a GridMap with elevation data,
 * and publishes the result for visualization in RViz or further processing. The elevation is assumed
 * to be encoded in the occupancy grid as integer centimeters (0–100). The GridMap frame and position
 * are computed from the OccupancyGrid metadata.
 *
 * This module is primarily intended for visualizing the full global terrain map, and does not compute
 * any statistical summaries like slope or RMSE.
 *
 * @version 1.0.0
 * @date 2025-07-07
 *
 * Maintainer: Boxiang (William) Fu
 * Team: Lunar ROADSTER
 * Team Members: Ankit Aggarwal, Deepam Ameria, Bhaswanth Ayapilla, Simson D'Souza, Boxiang (William) Fu
 *
 * Subscribers:
 * - /mapping/filtered_global_map: [nav_msgs::msg::OccupancyGrid] The full global occupancy grid with elevation information.
 *
 * Publishers:
 * - /grid_map: [grid_map_msgs::msg::GridMap] The generated elevation map in GridMap format.
 *
 * Services:
 * - None
 *
 * @credit Adapted from the Local Visualization node for use in full-terrain global visualization.
 */
```

```
<name>PACKAGE NAME</name>
<version>1.0.0</version>
<description>INTENDED USE for Lunar ROADSTER project</description>
<maintainer email="NAME@cs.cmu.edu">NAME</maintainer>
<license>Apache License 2.0</license>
```

```
/**
 * @file
 * @brief DESCRIPTION OF CODE
 *
 * @author AUTHOR NAME
 * @version 1.0.0
 * @date CURRENT DATE
 *
 * Maintainer: MAINTAINER NAME
 * Team: Lunar ROADSTER
 * Team Members: Ankit Aggarwal, Deepam Ameria, Bhaswanth Ayapilla, Simson
 *
 * Subscribers:
 * - /SUB: [MSG TYPE] Description
 *
 * Publishers:
 * - /PUB: [MSG TYPE] Description
 *
 * Services:
 * - /SRV: [SRV TYPE] Description
 *
 * @credit NAME IN PACKAGE XML, Team CraterGrader (if code was originally
 */
```

Software: Architecture Compliance

- Refactored ROS topics to conform to a Directed Acyclic Graph (DAG) architecture

We will be employing three different node types for our project:

1. **Source Node:**

   These are event sources with no input message from other nodes. Examples would be user inputs and sensor inputs.

2. **Functional Node:**

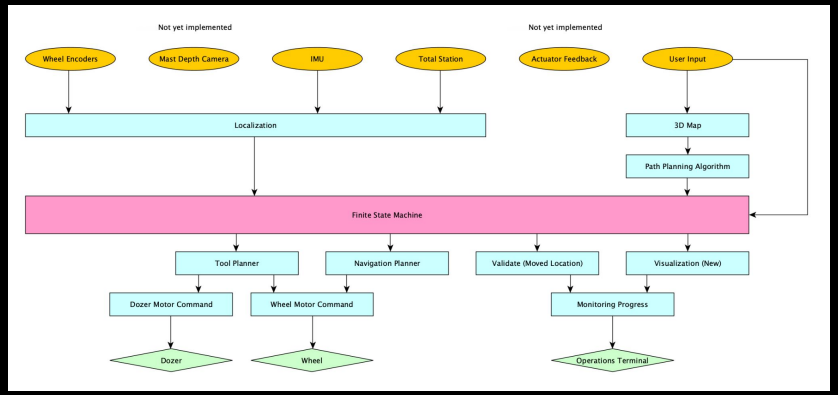
   These are algorithm operations that receive inputs from source nodes or other functional nodes and computes something to output. Examples would be our SLAM node or FSM Planner Node.

3. **Sink Node:**

   These are graph sinks that receive input messages from other nodes but do not output messages to be used for other nodes. Examples would be GUI output and motor commands (in a DAG architecture, motor command responses would need to be a source node, and be called back periodically on a timer, mimicking a pseudo-closed-loop. The reason for not implementing a full closed-loop is that if a callback packet is lost, the loop is broken and would cause compute problems. A pseudo-closed-loop would simply ignore this callback iteration, and compute on the next packet of received data).

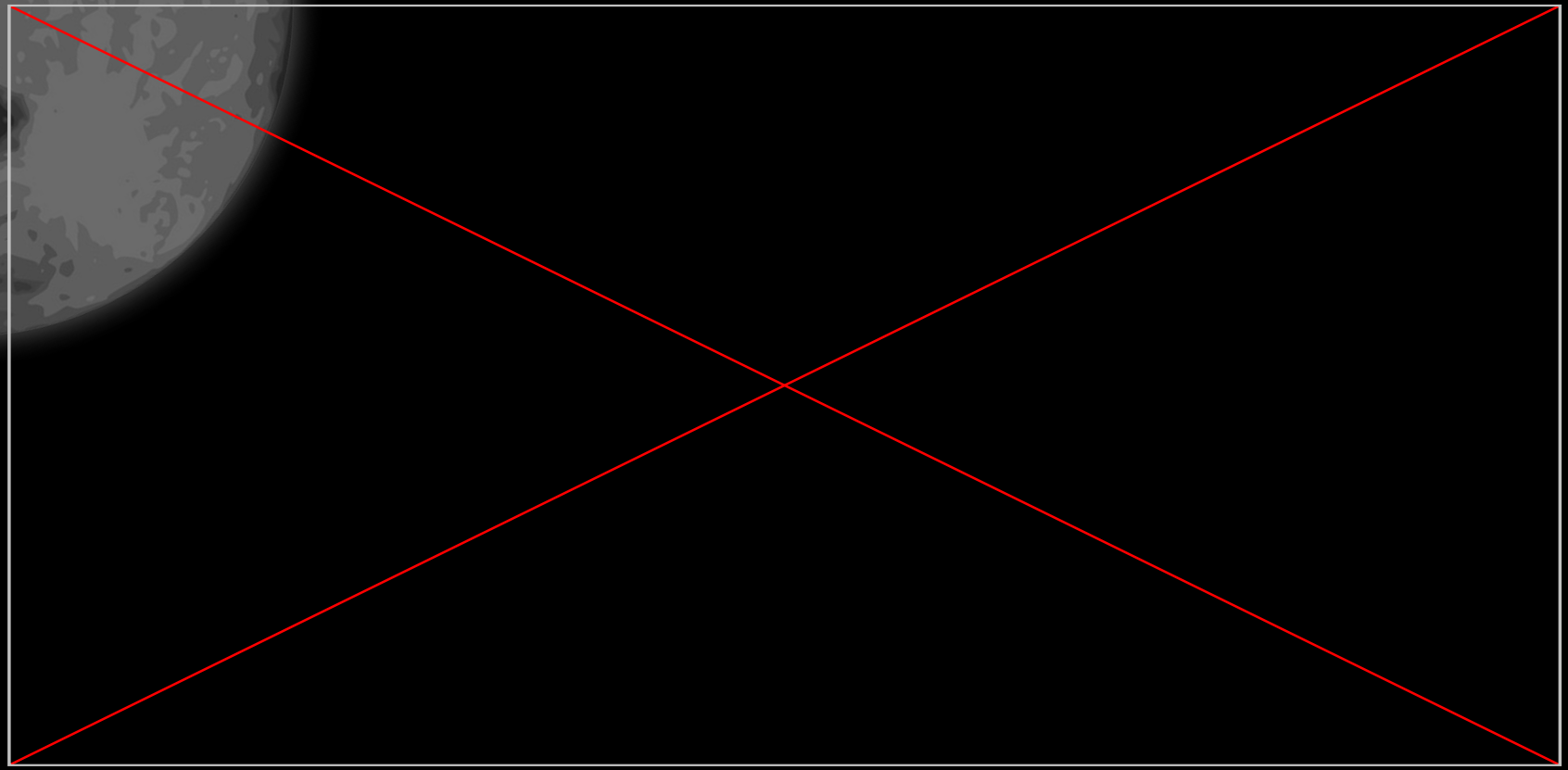
Definition:

A finite directed graph with no directed cycles. Arrows dictate the flow of information



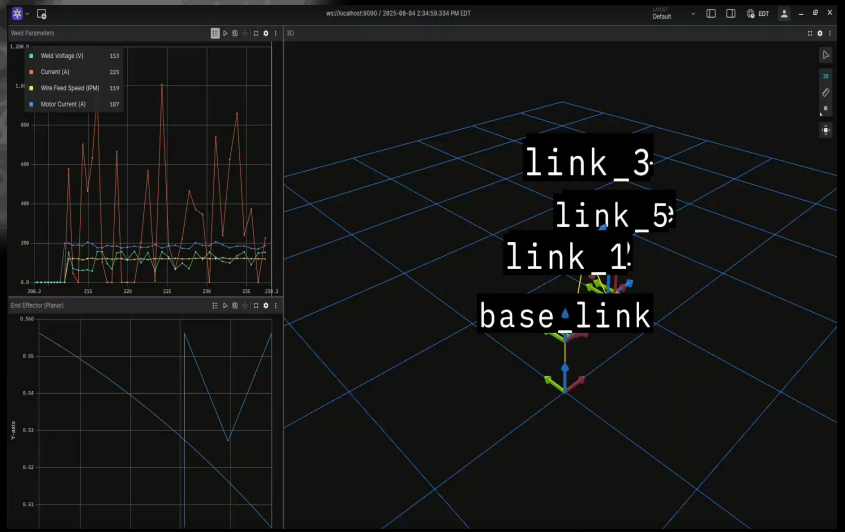
Software: ZED Camera Integration

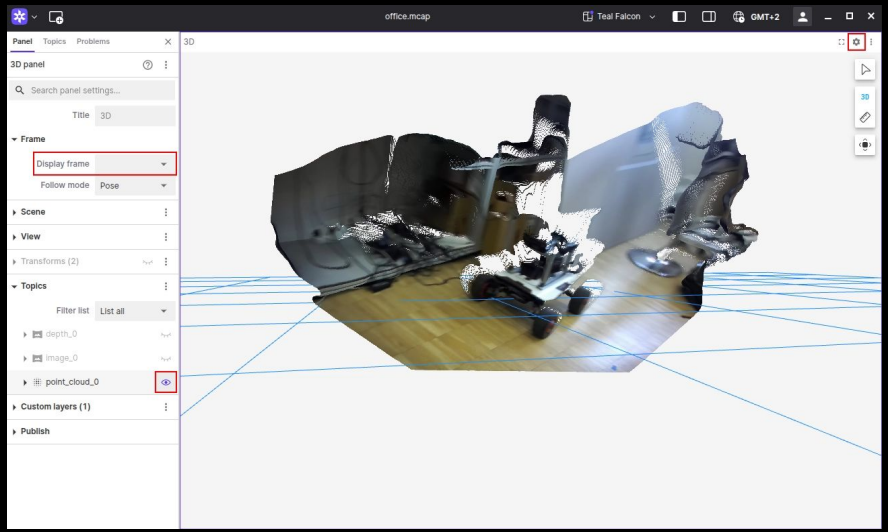
- ZED SDK and ROS2 Wrapper integrated inside Docker and tested successfully



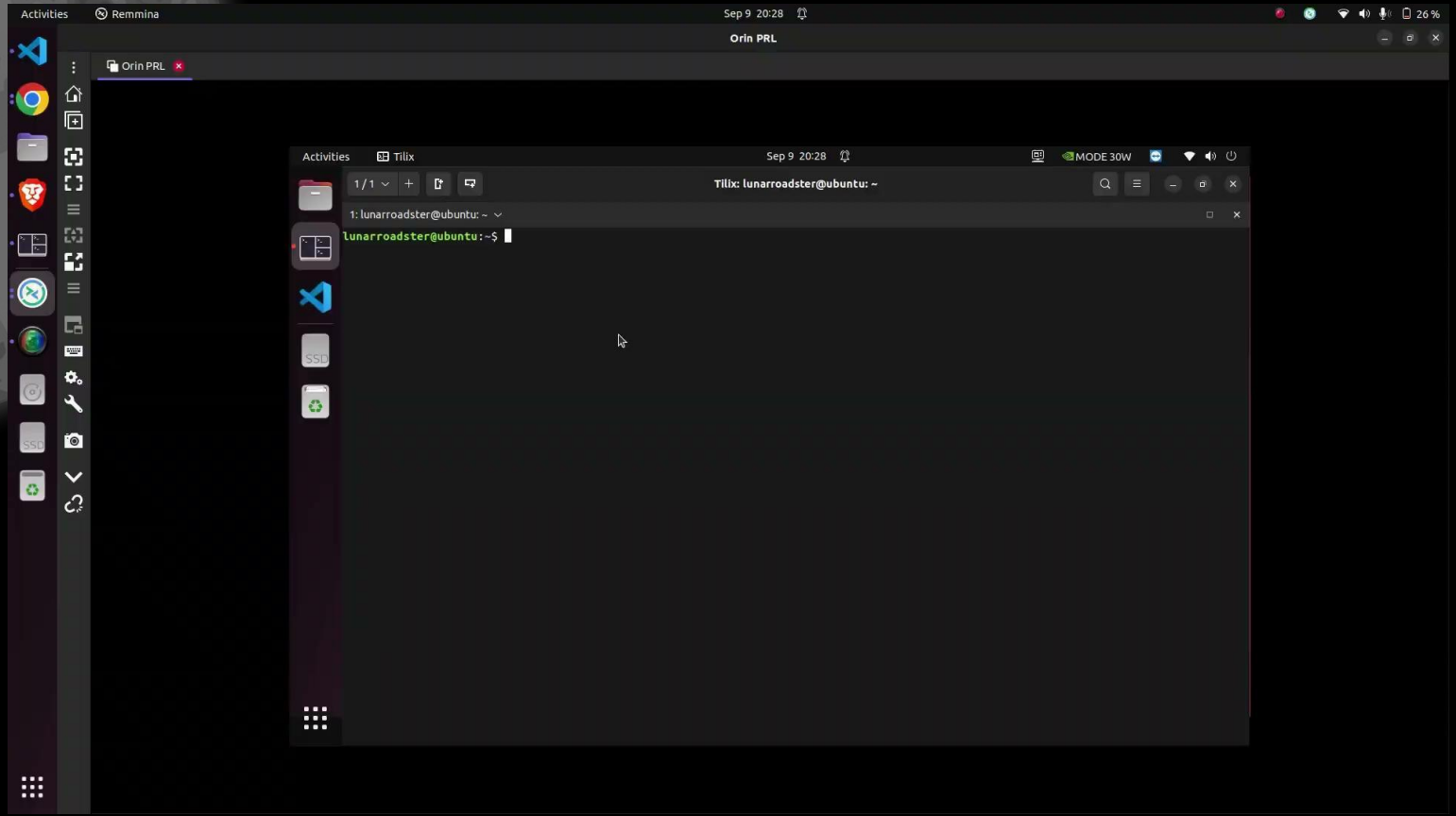
Software: GUI Setup

- Foxglove setup completed in the Docker.
- As system modules are developed the GUI dashboard will be updated.



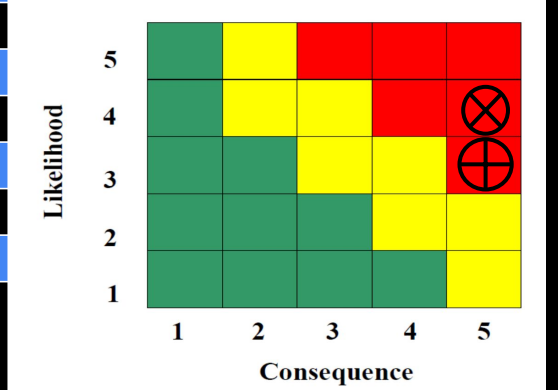


Software: Master Launcher



Risk Management

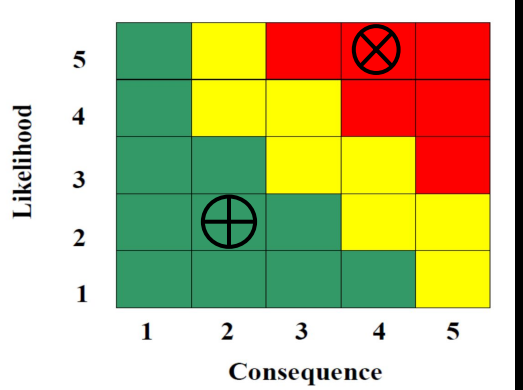
Risk ID	Risk Title	Risk Owner	Risk Type:	Logistics
R30	No spares available	Team		
Description		**Date Added**		
Discontinued model, spare parts unavailable		3/4/2025		
		Date Updated		
		8/30/2025		
Consequence				
The whole project falling through, or redo almost all subsystems on a different rover.				



Action/Milestone	Success Criteria	Date Planned	Date Implemented
Check out eBay and other similar platforms for spares	Successfully find exact spares on these platforms	3/6/2025	
Check out and stock similar parts if not same	Successfully find and stock similar parts	3/6/2025	
Find a twin rover that was used by a previous team on campus	Successfully find the twin rover and scavenge parts	3/6/2025	3/7/2025
Maintain all parts, especially mechanical parts	Successfully avoid future breakdowns and part failures	9/10/2025	

Risk Management

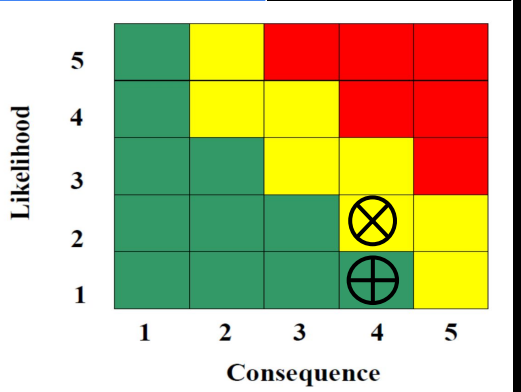
Risk ID	Risk Title	Risk Owner	Risk Type:	Technical
R34	Arduino requires reset before operation	Bhaswanth		
Description		**Date Added**		
Arduino needs to be manually reset each time before starting autonomy or switching between autonomy and teleoperation modes.		3/4/2025		
		Date Updated		
		4/10/2025		
Consequence				
Slows down setup time and impacts operational readiness, delaying mission start and mode transitions.				



Action/Milestone	Success Criteria	Date Planned	Date Implemented
Check USB port permissions and drivers issues on Jetson	Successfully establish consistent serial connection without reset	4/26/2025	
Verify that Arduino is connected via USB 3.0 instead of USB 2.0 port	Ensure stable high-speed communication	4/26/2025	
Check for ROS node frequency mismatches causing packet loss to Arduino	Match ROS publish/subscribe rates	4/26/2025	

Risk Management

Risk ID	Risk Title	Risk Owner	Risk Type:	Logistics
R36	PRL Moonyard Access	William		
Description		**Date Added**		
Securing Moonyard access for testing/demos will be restricted and challenging		8/29/2025		
		Date Updated		
		8/29/2025		
Consequence				
No testbed available for testing and/or SVD				



Action/Milestone	Success Criteria	Date Planned	Date Implemented
Devise and discuss a testing and demo plan with Prof. Red and Prof. David Wettergreen beforehand and reserve slots	Successfully meet and discuss the schedule of high priority projects	9/11/2025	
Complete Medical Evaluation to get unrestricted but controlled access	Successfully complete the Medical Evaluation and get unrestricted access to the Moonyard	9/5/2025	

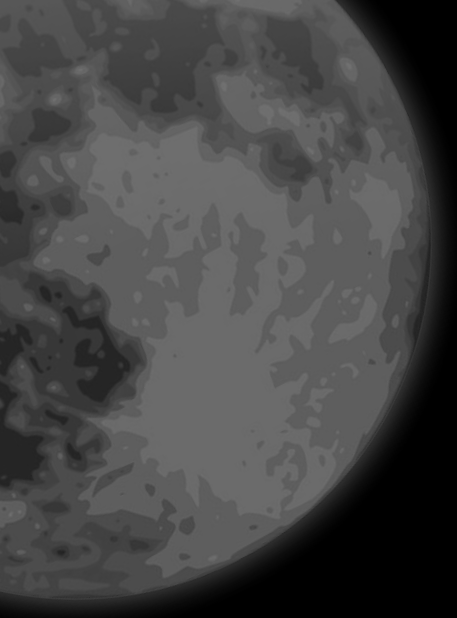
Issues Log

Issue ID	Date Initiated	Date Resolved	Participants	Description	Options	Resolution	Justification
I06	02/25/2025		**Ankit Aggarwal** **Deepam Ameria**	FRC Workshop Access	1. Request Tim 2. Ask John		
I11	08/29/2025		**Team**	PRL Moon Yard Access	1. Complete medical evaluation ASAP 2. Unit tests in LL67		
I12	08/29/2025		**Bhaswanth Ayapilla**	Localization frame shift after total station battery swap	1. Implement resection method using three known prism locations instead of orientate-to-line 2. Explore and test alternative localization methods (using SkyCam)		
I13	09/08/2025		**Team**	Compute unit (Orin & Jetson) unable to communicate with Arduino	1. Replace old Arduino with new one 2. Find code workarounds to force communication 3. Retrace wiring to make sure everything is wired correctly		

Future Work

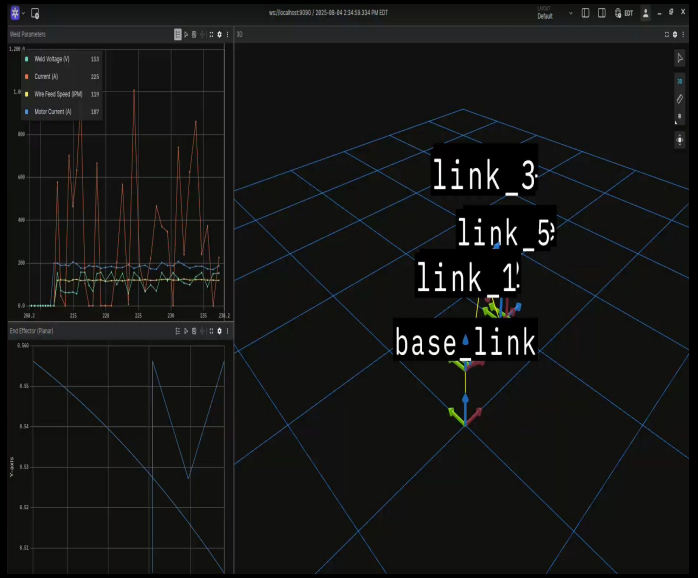
Before Progress Review 8, we hope to achieve:

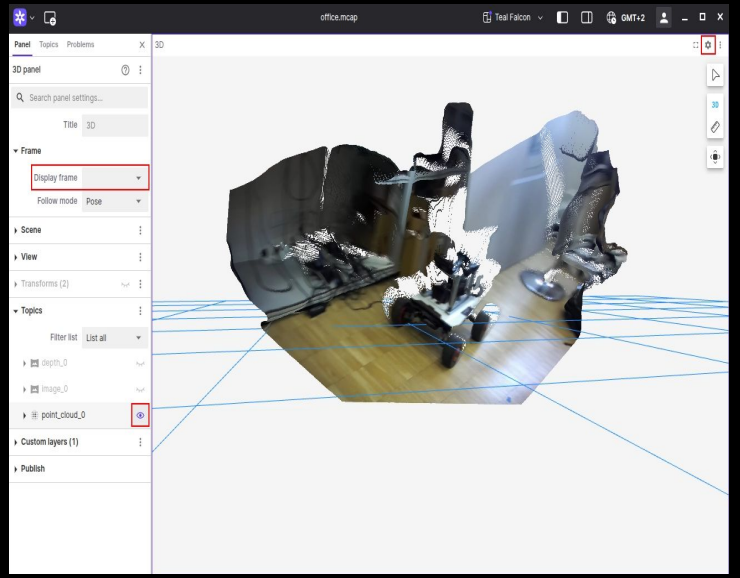
- Resolve Arduino connection & reset issue
- Implement new total station resection method
- Finalize validation stack code architecture
- Begin implementing global path planner
- Begin implementing global navigation controller
- Finish implementing selection of gradable craters from global map generated from FARO scanner



THANKS!

Team Lunar ROADSTER


Weld Parameters
Weld Voltage (V) 153
Current (A) 225
Wire Feed Speed (IPM) 339
Motor Current (A) 187
End Effector (Planar)
link_3
link_5
link_1
base_link


office.mcap
Teal Falcon
GMT+2
Panel
Topics
Problems
3D panel
Search panel settings...
Title
3D
Frame
Display frame
Follow mode
Pose
Scene
View
Transforms (2)
Topics
Filter list
List all
depth_0
image_0
point_cloud_0
Custom layers (1)
Publish
3D