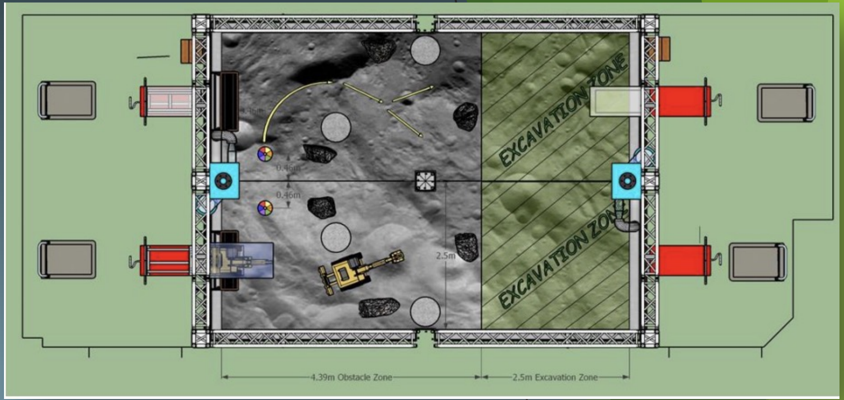
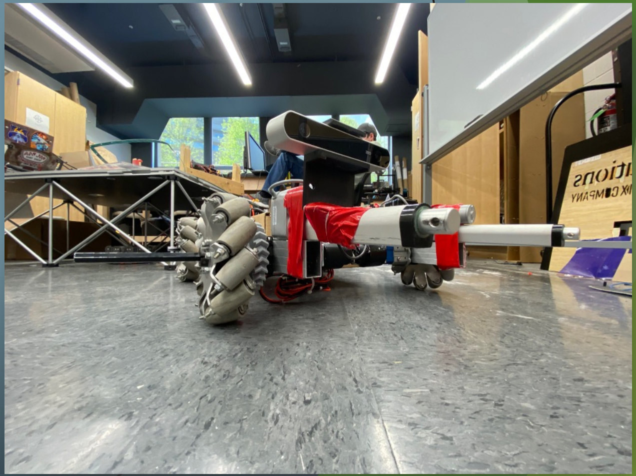
NASA Lunabotics Competition

Team 17: *Ahmed Moustafa, Rohit Kala, Justin Kilgo, Jackson Newman, and Jackson Burger*

Requirements

- 1.1m length x 0.6m width x 0.6m height.
- No more than 80kg.
- Can be controlled manually or autonomously.
- Must provide its own power.
- Cannot use any materials that would not work on worlds other than Earth.





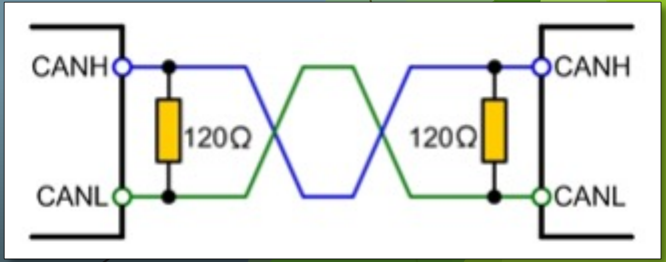
Key Concepts

Robot Operating System 2 (ROS 2):

- Framework that allows organization of data processing elements through a ROS Graph.
- Uses a Publisher-Subscriber policy for communication between Nodes – the individual executables within the graph

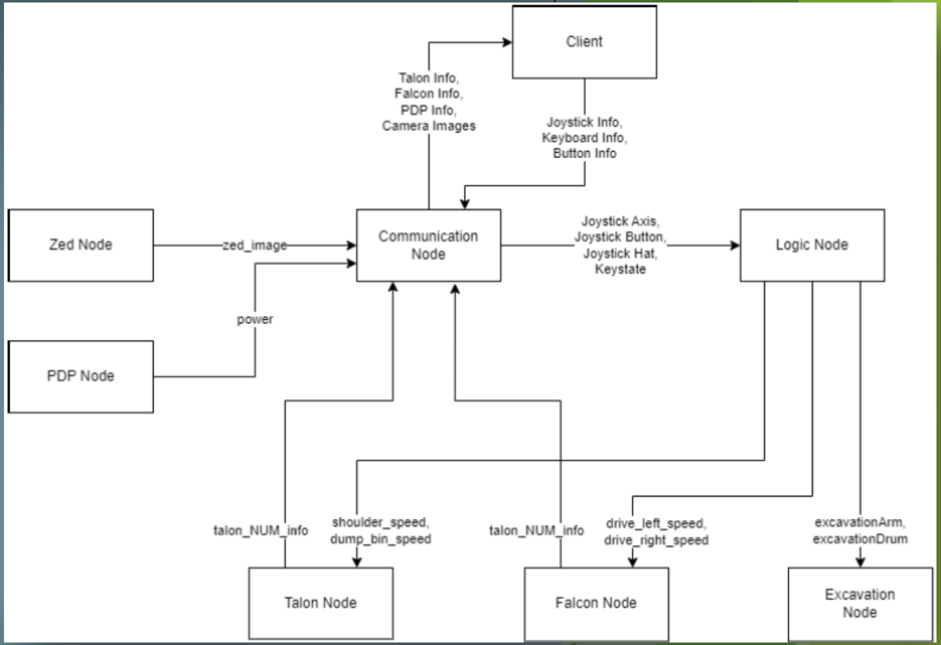
Controller Area Network (CAN bus):

- First developed in 1983 at Robert Bosch GmBH
- Robust message-based network designed to allow devices to communicate without a host computer



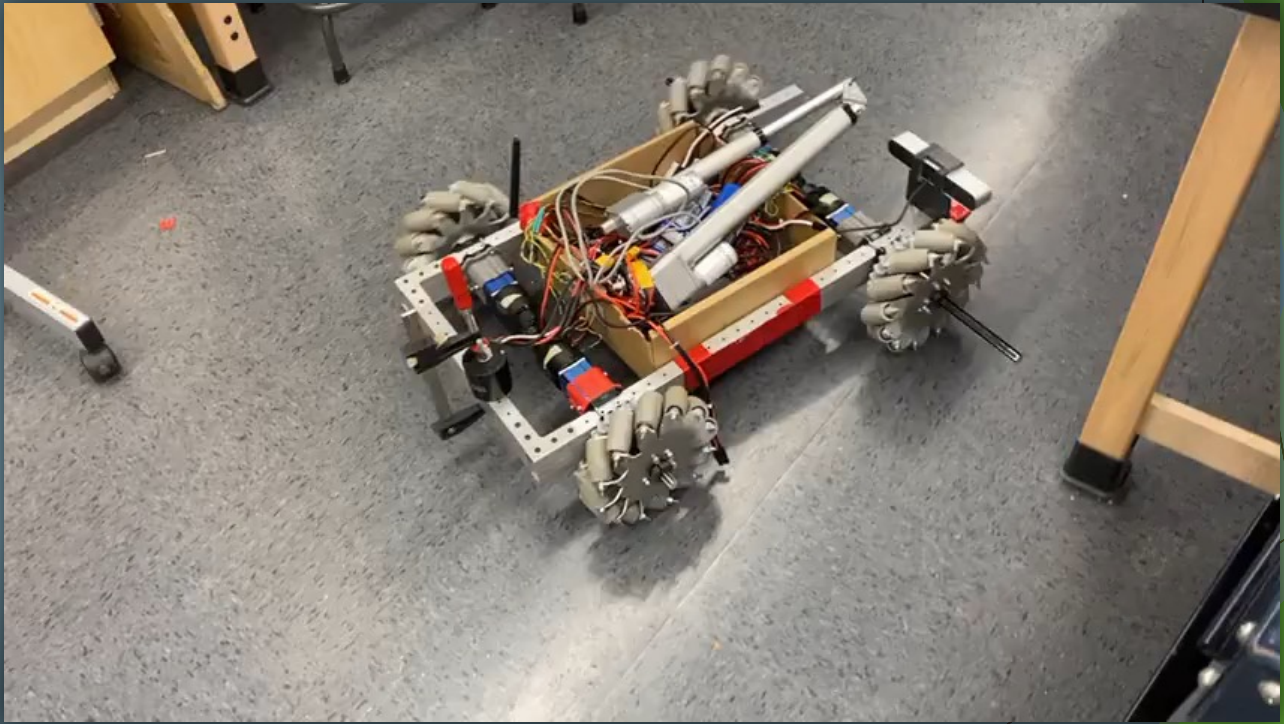
Manual Control

- Became our primary focus until physical robot is complete due to scoring priorities
- Client responsible for user input (*joystick*)
- Communication node facilitates the transmission of data throughout the ROS graph
  - Talon & Falcon nodes control motor speeds
  - PDP node measures device voltage and current
  - Logic node houses operation → calls other nodes
  - Excavation node controls belt and arm

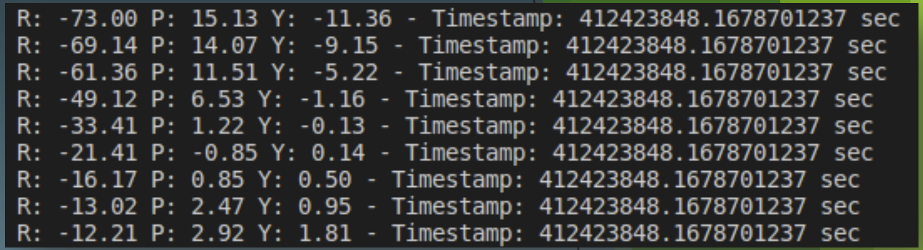


Node Diagram

Manual Control Demo

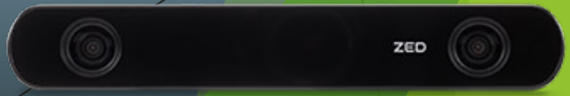


ZED Camera



```
R: -73.00 P: 15.13 Y: -11.36 - Timestamp: 412423848.1678701237 sec
R: -69.14 P: 14.07 Y: -9.15 - Timestamp: 412423848.1678701237 sec
R: -61.36 P: 11.51 Y: -5.22 - Timestamp: 412423848.1678701237 sec
R: -49.12 P: 6.53 Y: -1.16 - Timestamp: 412423848.1678701237 sec
R: -33.41 P: 1.22 Y: -0.13 - Timestamp: 412423848.1678701237 sec
R: -21.41 P: -0.85 Y: 0.14 - Timestamp: 412423848.1678701237 sec
R: -16.17 P: 0.85 Y: 0.50 - Timestamp: 412423848.1678701237 sec
R: -13.02 P: 2.47 Y: 0.95 - Timestamp: 412423848.1678701237 sec
R: -12.21 P: 2.92 Y: 1.81 - Timestamp: 412423848.1678701237 sec
```

- Camera can track its position on a point cloud map.
- Camera contains a gyroscope that measures the angular velocity of the robot.
- We utilize an OpenCV library to detect ArUco markers.
- We create a ROS2 Node to gather information from the ZED API.
- The Node also publishes this information to other nodes.

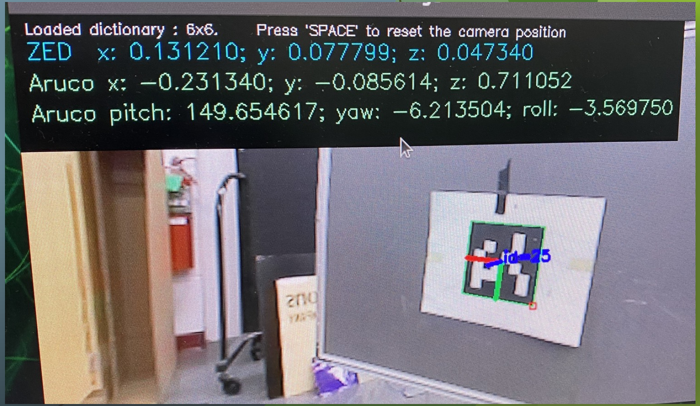


Robot Autonomy

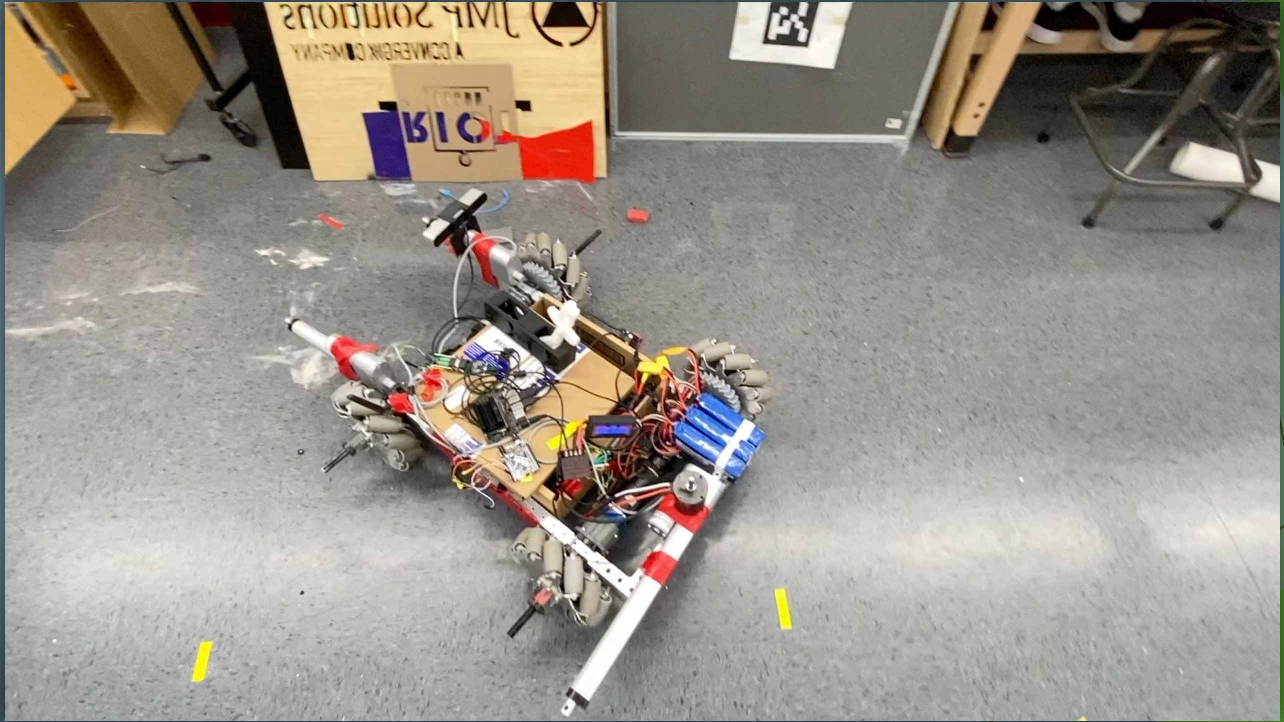
Initial Strategy:

1. Tracking the position of the robot.
2. Tracking the orientation of the robot.
3. Calculating angle between ArUco marker and robot.
4. Turning the robot a certain amount of degrees.
5. Driving the robot a specific number of feet.
6. Running the Excavation Macro.
7. Turn 180 degrees and return to the starting position.
8. Running the Dumping Macro.

```
if(robotState==LOCATE){
    changeSpeed(0.15,-0.15);
    if(position.arucoVisible==true){
        if (abs(position.aruco_roll) < 90.0) {
            left = 1;
        } else { // dot on top
            left = -1;
        }
        RCLCPP_INFO(this->node->get_logger(), "Left: %d", left);
        setDestAngle(position.yaw + 90.0);
        destination.x=-2;
        destination.z=1;
        changeSpeed(0,0);
        robotState=ALIGN;
    }
}
```



Autonomous Control Demo

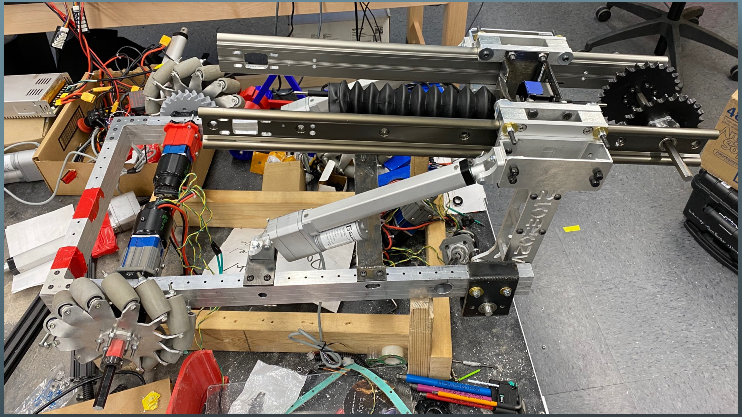


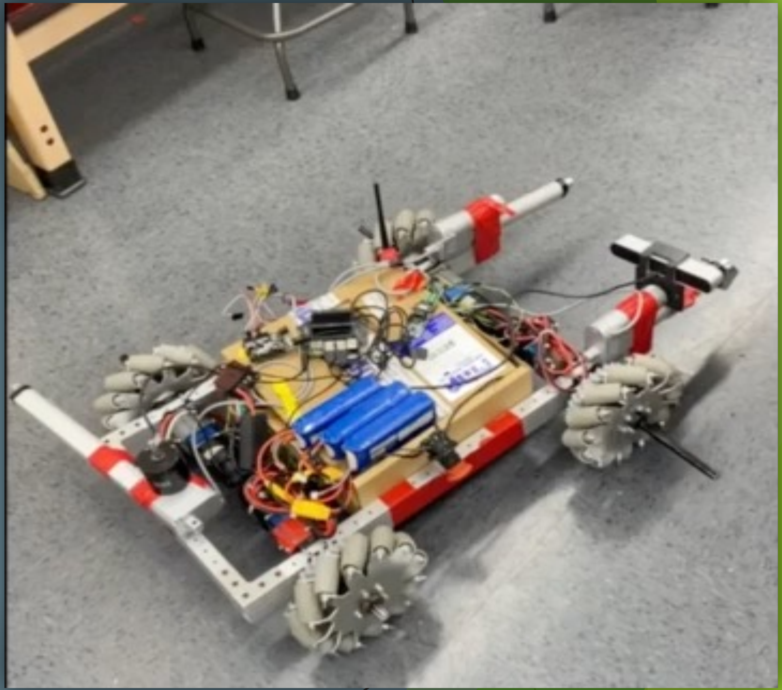
Task Schedule

Task	Dates
Investigate Zed2i camera and software to begin object detection	2/6 - 2/13
Implement node for camera to publish images	2/13 - 2/27
Implement node to receive images and track position	2/20 - 2/27
Communication with client	2/27 - 3/11
Test/Fix previous driving automation code	3/25 - 4/1
Implement autonomous functions to test expected operations of autonomy (turning, driving certain distance, etc.)	4/1 - 4/15
Handle ArUco marker orientation	4/8 - 4/15
* Implement algorithm that performs expected operations for the excavation tool	3/6 - 3/20
* Incorporate path planning and driving nodes to navigate the field	3/27 - 4/10
* Implement testing for autonomous and manual control	4/3 - 4/17

Challenges/Roadblocks

- Hardware was delivered late.
- Documentation on the software we were using was scarce.
- Hardware would fail unexpectedly.





Deliverables

- **Project Website**
  - Website contains project & team information
- **Project Proposal/Report**
  - Proposal explains overall ideas for the project and the Report explains the design decisions made
- **Autonomous Code**
  - We ultimately have programed a semi-autonomous robot to satisfy the Robotic Mining Requirements
- **Manual Control Code**
  - Alongside autonomy is the robot's manual joystick control functionality

GitHub

Thank You!