Definitions:
The robot consists of three “stages” on which components are attached. The stage farthest from the test rig is the “x stage”, on the ground and closest to the test rig is the “y stage” and the highest stage is the “z stage.” Any components placed on these stages (such as motors and gears) will have a corresponding prefix. The end effector is attached to the z stage.

The lead screws which move the robot are named in the same fashion. The “x screw” is parallel to the test rig and the ground. The “y screw” is placed normal to the test rig. The “z screw” is parallel to the test rig and normal to the ground.

Our robot uses normal force carriers and platforms to transmit the rotational force of the gears into translational movement. When the gear turns and pushes the normal force carrier, it pushes the platform which is a block attached to the stage with an unthreaded screw which moves the stage in the desired direction. These components have the same prefix as the lead screw on which they move.

Robot Status
<u>State of Mechanical</u>

- Design of all mechanical components completed
- All components are finalized as of now except the following:
    - The end effector has been assembled but not attached to the z stage.
    - The camera mount has been assembled but not attached to the z stage.
    - The top level of the robot the support rod holders attach to has been designed.
    - The support rod holders has been designed.
- Motors have been placed in their respective mounts in the finalized position.
- Gears have been laser cut from acrylic and fitted to motors and lead screws.
- Movement along the x and y screws has been tested and functions on the ¾”-10 screws.
- X platform guard has been made to prevent wires from interfering with the x gears.
- Partial transition from ¾”-10 to ¾”-6 screws
    - Note: due to supplier shortages, we were unable to use the lead screws specified in our design (¾”-6). Instead, we used ¾”-10 screws which we performed milestone 1 on
    - After the milestone, we began to switch to ¾”-6 screws but still needs to be finished and tested.

Next steps:
- Finish transitioning to ¾”-6 screws.
- Laser cut the top level and support rod holders so support rods can be put in.
- Cut the polyethylene sheet to size so it can be attached to the z stage and slide along the support rods
- Test movement in the z direction
- Purchase standoffs to mount PCBs to stages
- Create guards around other moving parts to prevent harm to humans or wires.

- Determine if weight (sand or gravel) is required to weigh down the sides of the robot during operation.

Electrical

- Modular stepper motor driver circuits soldered (with pin headers for detaching/attaching drivers)
- Wire harness implemented for carrying power/signals between stages
- Circuits calibrated, debugged, and tested

Next steps:

- Add fan for active cooling, as well as 12V power from buck convertor for fan
- Design and purchase PCBs from Shenzhen to replace soldered perfboards, to increase current capacity and reliability (in time for Milestone 3)
- Remove excess solder from previously-used stepper motor drivers
- Address issues outlined in “Known Electrical Issues”

State of Software

- Raspberry Pi Zero W set up
  - Flashed Raspberry Pi with Raspbian Stretch Lite
  - Set up Raspberry Pi for headless SSH connection over mobile hotspot WiFi, laptop hotspot WiFi, and home WiFi
  - Installed and updated ROS, OpenCV, and other necessary software onto Raspberry Pi
- Wrote Motor GPIO code to control 3 stepper motors, using commands received via ROS
- Wrote and tested a manual controller using PyGame and ROS to interface with Motor GPIO code
- Tested taking a picture with the Raspberry Pi’s attached camera and sending it to a laptop using SCP
- Performed an SSH ping test to find statistics on latency, upload rate, and download rate between laptop and Pi
- Camera set up and tested to obtain images
- OpenCV script written and tested to identify port indicators at a variety of angles
- Controls script written to derive speed and direction given CV output
- ROS elements integrated into CV and Controls code, testing required to validate
- Separate OpenCV and controls code, so that controls code can run at a different rate and make use of other sensor data as necessary (without need for multi-threading/multi-processing)

Next steps:

- Test OpenCV code for runtime on the Raspberry Pi and on a laptop
  - Decide to run OpenCV code on Pi or on laptop based on findings
  - If laptop is chosen, write shell script that sends images from the Pi to the laptop regularly via SCP, and find a way to have ROS messages move across devices
- Edit OpenCV code to read a continuous stream of pictures, and publish location data to other software elements via ROS
- Test software control loop with the rest of the robot in time for Milestone 2 - optimize for speed and precision in time for Milestone 3

Member Contributions

<u>Eddie:</u>

Mechanical Contributions:

- Created CAD of all components except gears and created an assembly of the robot
- Machined the platforms, adapters for gears to attach to the lead screws, laser cut the gears, laser cut the end effector at the Student Machine Shop
- Helped with construction of stages.
- Assembled the x screw end supports
- Assembled the robot with the help of Grace.
- Cut normal force carriers to size.
- Performed mechanical testing of the x and y stages
- Determined materials to be purchased at McMaster Carr and at Lowes
- Visited Lowes with Grace to pick up components (including a 6’ lead screw)
- Managed the change from ¾”-6 to ¾”-10 lead screws

Electrical Contributions:

- Soldered the wire harness
- Soldered the initial motor driver board without the headers
- Soldered the power supply board which houses the buck converter, the 5 V regulator, and the 3.3 V regulator.

Software Contributions:

- Helped select a suitable indicator for the computer vision software
- Helped determine which image processing tools are useful before indicator location is detected.

<u>Grace:</u>

Mechanical Contributions:

- Created CAD for Gears
- Constructed motor risers
- Helped with construction of stages
- Helped with construction for various other mechanical aspects (drilling, screws etc.)
- Constructed x-platform guard

Electrical Contributions:

- Soldered stepper motor driver circuit for one motor

Software Contributions:

- Designed port indicator setup and obtained images for testing purposes
- Implemented OpenCV to successfully detect port indicators from various different angles and orientations
- Created controls script informed by CV output
- Added ROS integrative elements to Controls and CV code

Rocco:

Mechanical Contributions:

- Helped design and construct end supports
- Helped make detailed design decisions such as:
  - Size and orientation of stages
  - Presence, thickness, and material for guiding rails
- Assembled robot to collect photo evidence

Electrical Contributions:

- Designed stepper motor driver circuits for 3 motors; soldered two of them onto perfboard
- Researched, tested, and calibrated buck converter and stepper motor drivers
- Selected and acquired interfacing components such as perfboards, wires, and pin headers based on requirements such as safety, current draw, and modularity
- Selected and acquired fan for active cooling
- Debugged circuits
- Helped design and implement wire harness

Software Contributions:

- Flashed Raspberry Pi with Raspbian Stretch Lite
- Set up Raspberry Pi for headless SSH connection over mobile hotspot WiFi, laptop hotspot WiFi, and home WiFi
- Installed and updated ROS, OpenCV, and other necessary software onto Raspberry Pi
- Wrote Motor GPIO code to control 3 stepper motors, using commands received via ROS
- Wrote and tested a manual controller using PyGame and ROS to interface with Motor GPIO code
- Tested taking a picture with the Raspberry Pi’s attached camera and sending it to a laptop using SCP
- Performed an SSH ping test to find statistics on latency, upload rate, and download rate between laptop and Pi