Pipetting Robot for Use in HIV Therapy Monitoring Tests in Western Cape, South Africa

2020 - 2021

By Selam Getaneh, Obumneme Osele, William Zeng, and Kevin Zhang

Executive Summary

In Western Cape, South Africa, scientists need a more efficient method for monitoring the efficacy of antiretroviral therapy (ART) in patients with HIV, as the total population receiving ART is increasing and the ART failure rate is decreasing. Sample pooling has been proven to be an efficient method for detecting the virological failure in patients receiving ART.

The team's main objective is to develop a pipetting robot capable of executing sample pooling so the end user can monitor ART success in patients with HIV. This quarter's team is continuing work on the pipetting robot's mechanical design so it can be capable of tracking, pipetting, mixing, and outputting plasma samples in order to make sample pooling a more feasible method to implement in labs monitoring ART.

After outlining our device's new workflow and updated target specifications, we have made significant strides in working towards a prototype. Adding onto the previous work in XY motion, we have completed a Z axis of motion. Our build is based on the Z-axis mechanism designed last year, but after manufacturing, we found weight issues with the old design. We addressed these difficulties with a combination of software and hardware fixes. Afterwards, we completed successful repeatability and accuracy testing for the new axis of motion.

One of the most drastic changes we have made is in the new design of the input tube rack, which is now a rotating input rack with complementary sensing and barcode scanning modules. Additionally, we also have completed the mounting of our dispersal mechanism. All modules other than the pipette tip disposal have been manufactured, calibrated them, and are functioning independently. We ran into issues at the very end with our controller mechanism, preventing full integration of the individual modules, so the main future direction will be to replace the controller and conduct more accuracy and speed testing.

Table of Contents

1. Review of Design Problem

The largest human immunodeficiency virus (HIV) epidemic in the world is located in South Africa, where 7.5 million people are living with HIV as of 2019. Although HIV prevalence is still highest in South Africa, the VF prevalence is decreasing while the total population receiving ART is increasing. Therefore, the testing efficiency of sample pooling is increasing, making sample pooling a cost-effective strategy used to monitor the efficacy of ART. Due to the extra time and risk of human error though, a feasible approach to sample pooling is not possible without a reliable and affordable mechanical solution.

Our goal in this quarter and next is to continue designing and building a pipetting robot that can automate all the tasks needed for sample pooling. These tasks include mobilizing the tubes and pipette, scanning tube barcodes, and aspirating and dispensing liquid. Our primary users are scientists studying HIV sample pooling in South Africa: Dr. Martin Nieuwoudt and Dr. Gert van Zyl. As we are a continuation project from last year, we are both using and modifying elements of the previous work. This quarter, we have successfully updated the floor plan design, completed the input carousel rack construction, implemented barcode scanning, and updated the controller mechanism of the pipetting robot.

1.1 Technical Assessment of Existing Solutions

After canvassing the features of existing solutions on the market, opportunities for improvement on the existing Capstone design presented itself as well as support for the retention of some attributes. The aim is to make adjustments to the existing design to promote its functionality and usability for our stakeholders. The robot is meant to perform both pooling and organizing procedures to improve the efficiency and throughput of sample pooling In this section, we intend to outline key market attributes found in existing pipetting robot solutions as well as the ways in which they are either already implemented in the existing Capstone project or how they will be potentially included.

Pipetting Arm

Across existing designs, we observed designs for pipetting robots that included an arm that was capable of planar motions to perform pipetting tasks as requested by the user. Pipetting arms were designed to move in the X-Y plane to defined locations and utilized an added Z-motion feature to approach labware of interest in the vertical direction. Some existing solutions accomplished establishing the labware of interest employing AI based-recognition while others relied on user input guided by a preset calibration. For the sake of simplicity and based on the lack of interest for a complex recognition system from our client, we will be implementing the latter of the two options for alignment with labware.

For some market systems, arms appear to use force based sensing to secure pipetting tips and an internal ejection mechanism to release pipette tips for disposal. Given that the current Capstone design works in a market available single channel manual pipette with its own ejection mechanism, we intend to design a ledge into the system that, when interacted with, activates the already in place ejection mechanism. Per the previous work completed on the project, the aspiration and dispensing of the fluid will be accomplished using the air cushion method. In this method, a piston capable of pushing and pulling fluid into the tip of the pipette is used in concert with an air cushion that separates the fluid from the piston. The reports from the previous team and our team conversation support that the accuracy of this method which is being used by our manual pipette add-on meets the specifications from our clients.

Deck Plate

During the operation of the pipetting robot, several different tasks need to be done in a particular sequence. For example, the process of adding one sample to the pool involves picking up a pipette tip, navigating the pipette to the input tube, picking some fluid up, scanning the input barcode, navigating to/depositing in the output, and disposing of the pipette tip. These ordered processes involve different labware and hence need to be organized into dedicated and efficient positions. This organization is where the deck plate comes in.

The deck plate effectively is an XY-plane with coordinates for the necessary stations for each of the tasks, and in nearly all the designs, the pipetting arm can identify these coordinates. The Opentrons OT-2 robot (base price at $5000) has a gridded deck plate, with metal bars forming slots for customizable labware to be slotted in[1]. There are similar components in competitor products such as the Eppendorf epMotion or the Sirius Automation MiniTasker which has a very functionally similar deck plate[2].

The modules that fit into the deck plates are sold separately and will likely significantly add on to the price of the product. The labware modules in all of the products that are offered are varied. Nearly all competitors have microwell plates, pipette tip holders, and test tube holders, but only the Sirius Automation MiniTasker has a barcode scanning functionality. [3]For our product, the customizability and range of labware modules is not critical as we only want to optimize one workflow to maximize throughput.

---

[1] Labworks, O. (n.d.). OT-2 Liquid Handler: Opentrons Lab Automation from $5,000. Retrieved from https://opentrons.com/ot-2/

[2] EpMotion® 5075 TMX. (n.d.). Retrieved from https://online-shop.eppendorf.us/US-en/Automated-Pipetting-44509/Liquid-Handling-Workstations-44510/epMotion5075TMX-PF-17832.html

[3] MiniTasker®. (2020, July 09). Retrieved from https://www.siriusautomation.com/products/automated-systems/minitasker/

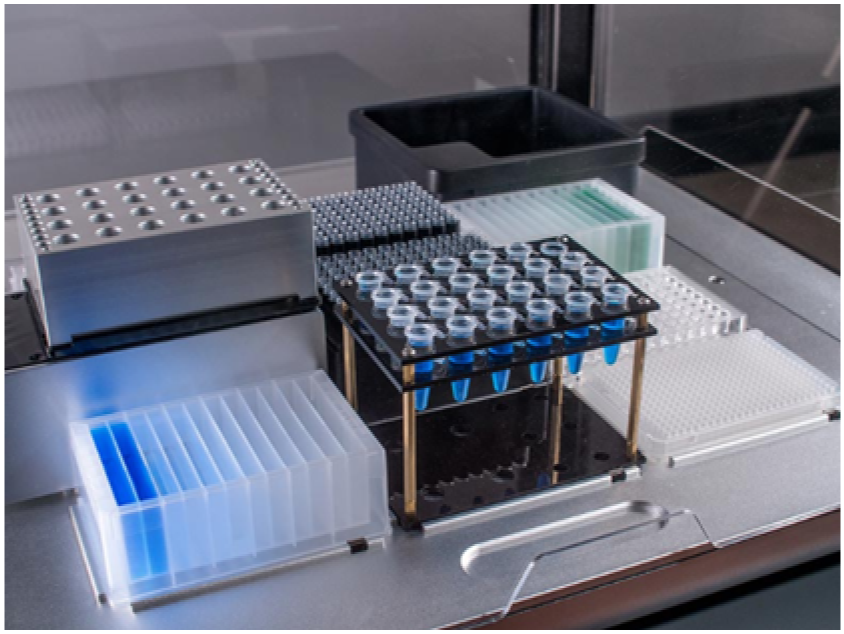


Figure 1: Opentrons Deck Plate

<u>*Barcode Scanning*</u>

In an effort to address the project deliverable of sorting samples, barcode scanning emerged as a feature on the market that proves promising. Currently on the market, there is the Sirius Automation which scans samples individually and pushes the input through a sample sorting module to keep track of samples belonging to certain pools. The design involves constructing a robotic arm that grabs samples and transports them to a 1D barcode scanning area in the system and then transports them back to their slot in the deck plate. We intend to retain the concept of a stationary 1D barcode scanning station but eliminate the necessity for a grabbing arm. This is further discussed in *Section 3*.

1.2 Workflow of Device

Our team agreed with some elements of last year's workflow, but we had to make adjustments in order to incorporate a rotating input rack (see later section 3). Along the circumference of the circular deck plate, there is one position where the input tube is both scanned and has its blood extracted. The tubes take turns in this position as the deck plate rotates, and during this time, the pipette deposits the blood in the output tube and the swaps tips.

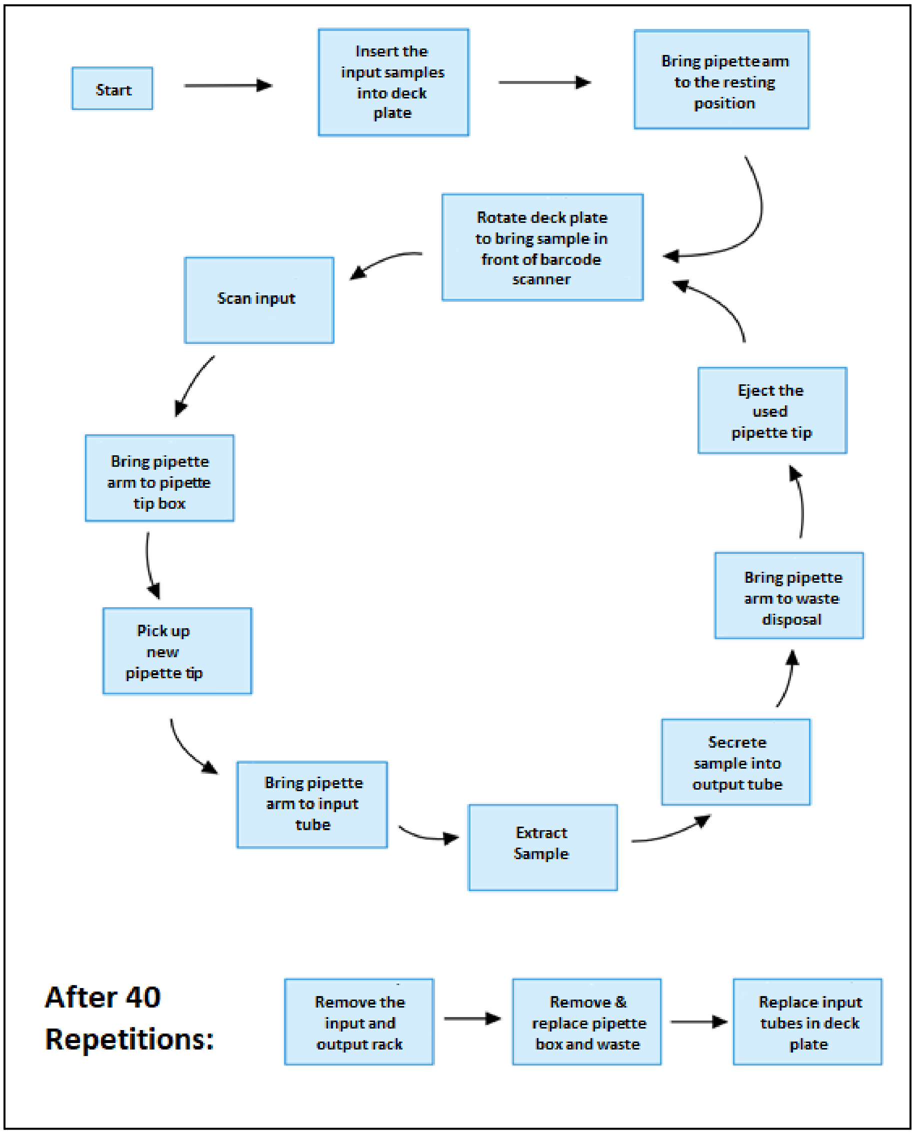


Figure 2: Workflow Diagram of Pipetting Robot

1.3 Overview of Requirements

After consulting Dr. Martin Nieuwoudt and Professor Mark Fisher, we have determined the requirements for our pipetting robot. The requirements are listed as follows:

- Accurate scanning of input test tubes with a 1-D barcode scanner
- Throughput rate of 300 samples per day (30 samples per hour)
- Secure pipetting avoidant of leaks over sample
- Disposal of pipette tips between each sampling
- Device size compatible to lab setting
- Cost below existing models ($1,000)
- Life requirement

Accurate Scanning Of Input Test Tubes

Accurate scanning of the 1D barcode located on the side of the test tube is paramount in keeping track of samples. Dr. Martin Nieuwoudt was unable to provide the exact orientation of the barcode sticker on the test tubes, so our machine would ideally be able to scan all angles of barcode alignment. Solving this issue will help mitigate human error in sample pooling.

Throughput Rate Of 300 Samples Per Day

Dr. van Zyl's lab collects up to 500 plasma samples per day. Our pipetting robot should achieve a similar processing speed, but it is more importantly that our robot does not fall behind the PCR testing speed that follows sample pooling, as this will slow down the HIV testing process as a whole. As an adjustment, we set the minimal throughput rate at 300 samples per 10 hour work day since the PCR testing speed at Dr. van Zyl's lab is slightly slower than sample collection.

Avoid Leaks Over Sample

To accurately test HIV in individuals, drippage from pipette tips into other samples must be kept at a minimum. Poor contamination control would only increase the sample pooling efficiency problem, and nullifies the purpose of our project design.

Disposal Of Pipette Tips Between Each Sampling

Because of the sheer amount of samples our machine needs to process, there needs to be pipette tips available for 1.5 to 2 times the daily throughput and the disposal of used tips between each sampling to avoid contamination. There will be four new pipette boxes at the designated station along with a waste container that is emptied out at the end of each day.

Compatibility To Lab Setting

The space that the robot uses must be smaller than the standard laboratory benchtop, which is 1.5 meters in length and 1 meters in width. The input of the robot should match the existing test tube rack in the lab. If not, further instruction that provides the input orientation should be made available to lab members. The interface of the robot should be user friendly and intuitive.

Cost Below Existing Models

The pipetting robot will likely expand beyond Dr. van Zyl's lab. The goal is to implement the device in labs all over South Africa, so the cost must be affordable. The reason why existing products are not implemented for sample pooling is that they are expensive and overqualified in terms of functionality. For our intents and purposes, we want to design a robot that costs at most $1000 after assembly and appropriate testing.

1.4 Updated Plan for Specifications Testing

We would like to test our requirements by the following metrics:

Table 1: Requirements and Specifications

Requirement	**Metric**	**Units**	**Marginal Spec**	**Ideal Spec**	**Rationale**
Accuracy of 1D Barcode Scanning	Percentage Error	Incorrect Scans/Total Scans	0	0	No errors in patient tracking
Throughput rate	Samples / work day	Samples / hr	90	120	Clients requested that our product be able to keep up with their PCR Machine
Accuracy of pipette Arm	Distance offset (dimensional tolerance)	millimeters	0.35	0.15	The diameter of each tip is ~7 mm, so offset must be less than half the radius of the tip
Accuracy of rotating deck plate	Angular offset	Degrees	0.772	0.1	Radians x radius of deck = circumference offset, circumference offset must be less then half the radius of tube; assuming radius of deck = 0.7m
Dimensions	Width x Length	Meter x meter	<1.5 x 1	< 1.2 x 1	Needs to be able to fit on a standard bench top

Cost	Money	USD	2000	1000	We want the device to be significantly under the market price but also as low as possible to be more accessible
Power Supply	Volts alternating current, cycles per second	VAC, Hz	220, 50	220, 50	Needs to be able to accommodate power outlets in South Africa
Cleanability	Compatible Cleaning Chemicals	N/A	High percentage ethanol, isopropyl alc.	High percentage ethanol, isopropyl alc.	Clients specified as what they use to sanitize
Product Lifetime	Number of Operational Years	Years	5	10	Based on client expectations and comparison with their other devices
Pipette Tips Available	Number of Pipette Tips that can be replenished at once	Number of Pipette Tips	300	600	We should have enough room to house enough fresh pipette tips for an entire workday, but ideally we could double that throughput's number

2. XYZ Motion

2.1 XY Motion Summary

The design team from last year made significant progress on the XY motion of the pipette. The prototype of their XY motion has been passed down to our team for further work on the Z-axis motion. From their XY motion testing, they were able to achieve 99.7% accuracy within the target specs listed for the pipette arm for individual axis movement.

We have been able to implement the controller (Universal G-Code Sender) to operate the prototype from the previous year; however, due to the limited number of axes available, we are considering changing the control mechanism. This will be discussed further in a later section.

In addition, because we wanted to ensure that our updated deck plate design (see later section) and barcode scanning station can fit comfortably within the XY motion frame already constructed, we fully dimensioned out in XY all the known components.

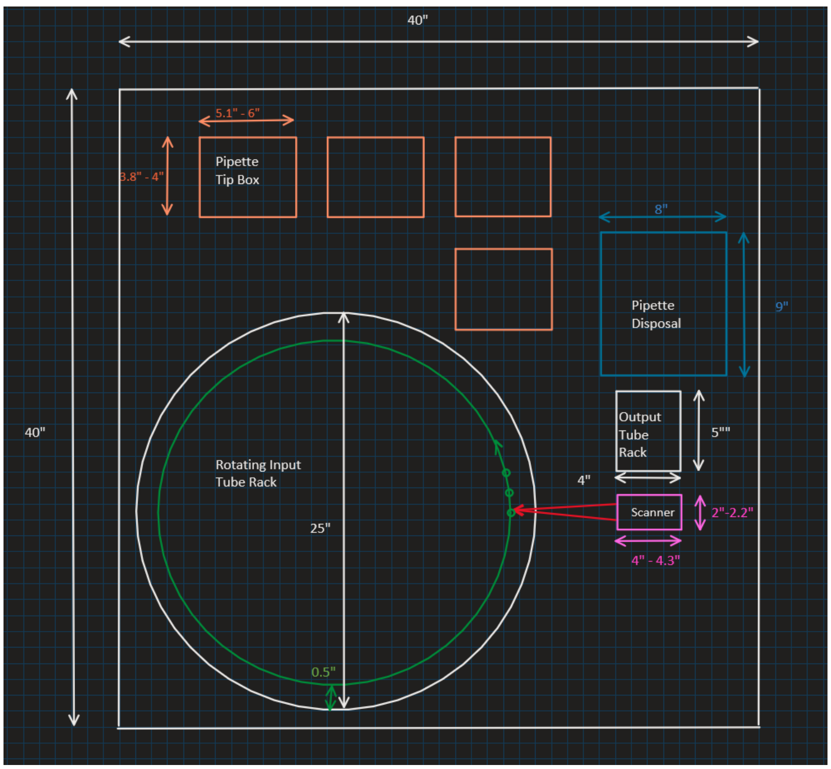


Figure 3: Dimensions of the New Deck Plate Layout

2.2 Z Motion Troubleshooting

While the previous team did not actually implement the Z motion, they did finish their CAD for it before they left. As a result, after our team determined the design to be viable, we have ordered the necessary parts and completed construction. Since the Z motion does not need to be as accurate as the XY motion, we believe the same specifications are more than necessary. Early on, we experienced some issues with the Z-axis motion due to the weight of the NEMA 23 motor. Last quarter, this was a major concern because the pipette module was unable to move upwards reliably against gravity even before adding on the pipette. GIFs of the state of all 3 axes by the end of fall quarter are shown at the end of this section. As you can see, even though the X and Y motion were moving comparably to what was reported by the previous team, the Z motion was unable to move upwards under the weight of its components against gravity. It turned out to be both a hardware and a software issue. Fortunately, we were able to solve this issue by adjusting the motor's feed rate for Z motion and replacing the previously burned out motor drivers. By reducing the feed rate, we were able to achieve a smaller torque which could allow the gantry plate to move vertically against gravity with the combined weight of the NEMA 23 motor and pipette module without overwhelming the driver.

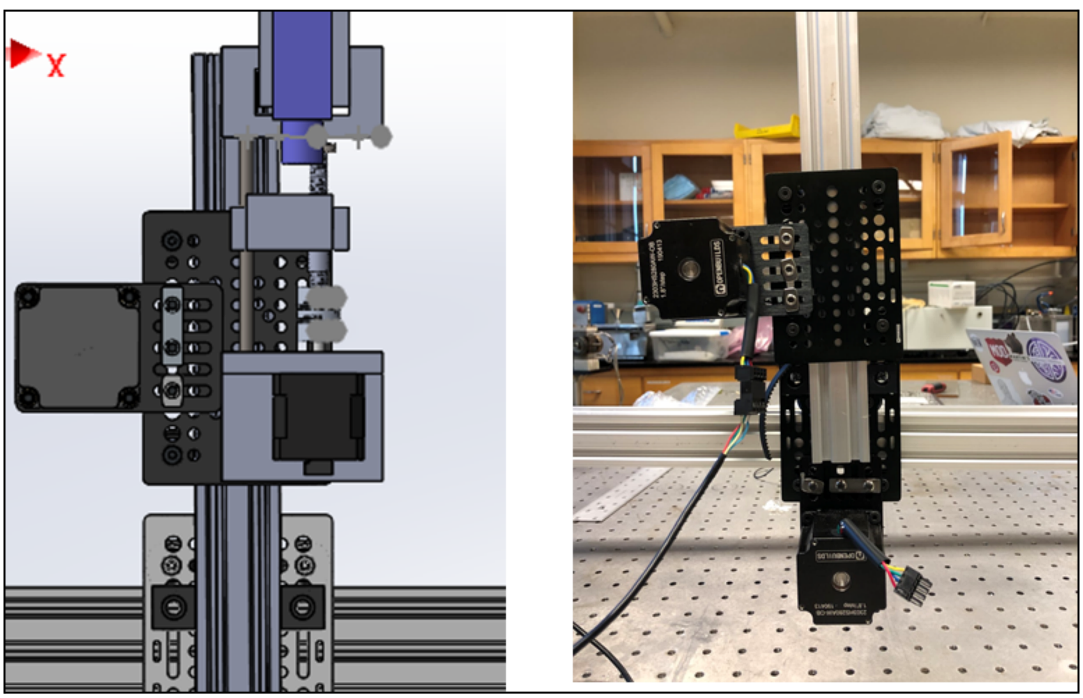


Figure 4: Z-Axis Motion without pulley belt (inverted view) from CAD to construction

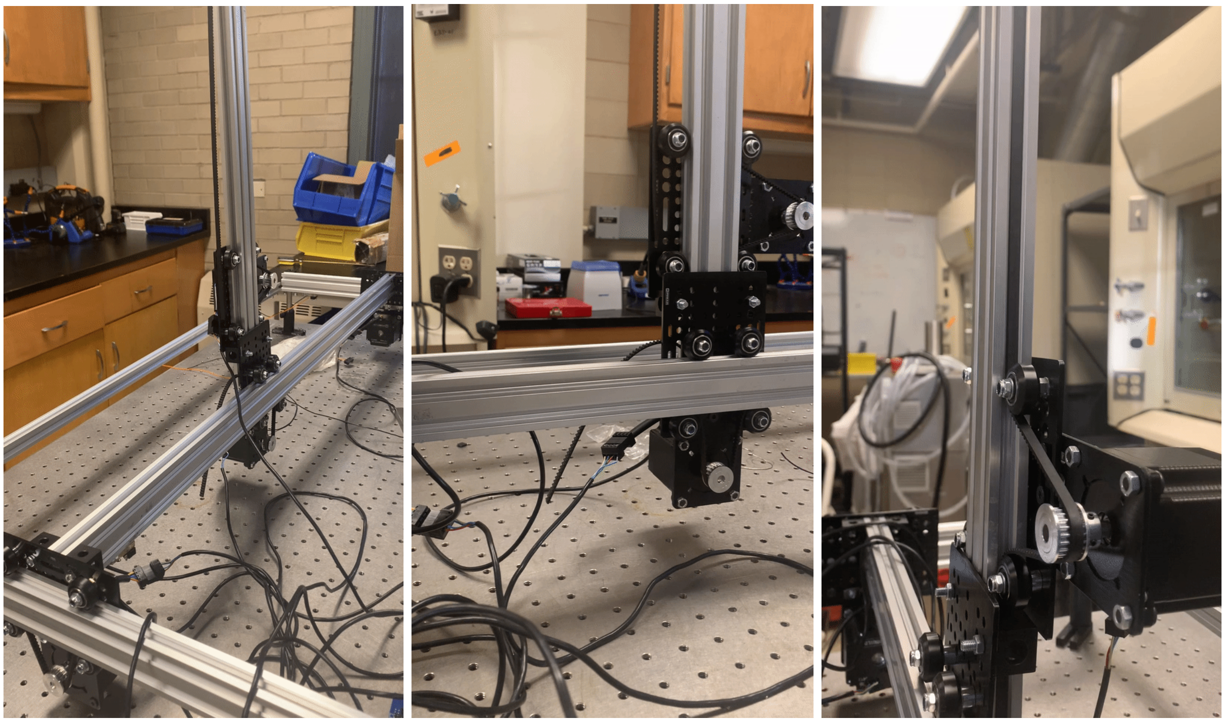


Figure 5: State of the movement for all 3 axes by end of fall quarter (see previous z-motion issue troubleshooted this quarter)

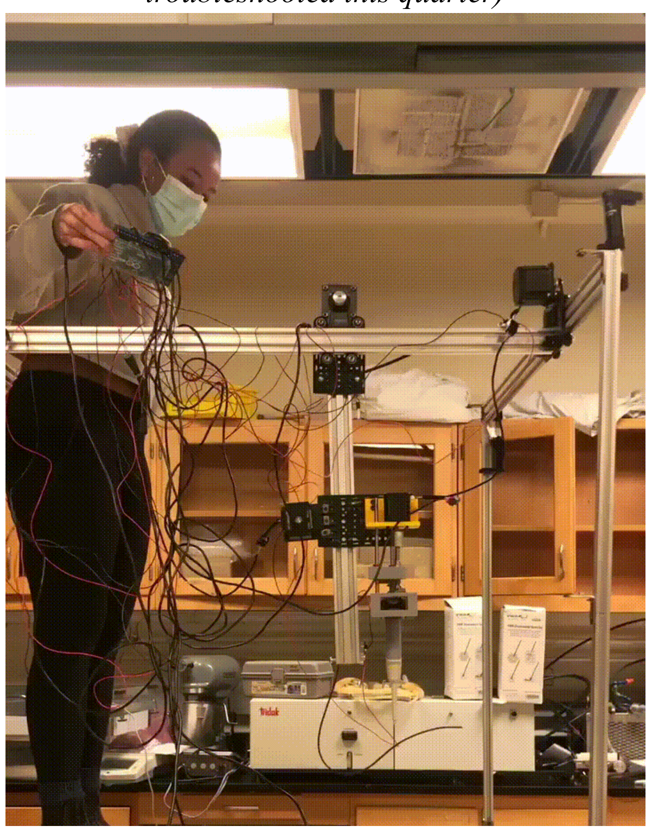


Figure 6: State of movement for z axis after troubleshooting this quarter

2.3 Limit Switch Installation and Homing Sequence

After completing movement in all 3 axes, the system needed a way to positionally calibrate itself everytime time it turned on. We did this by implementing mechanical limit switches at the boundary of each axis. Mounts were made using acrylic attachments designed for screw-in to the 8020. After implementation, we ran our first homing sequence. The bar hits the limit switch and then slowly backs away to the exact, first position where the limit switch is no longer triggered.

Figure 7: Limit switches using a) wired connections to GRBLduino b) acrylic mounts

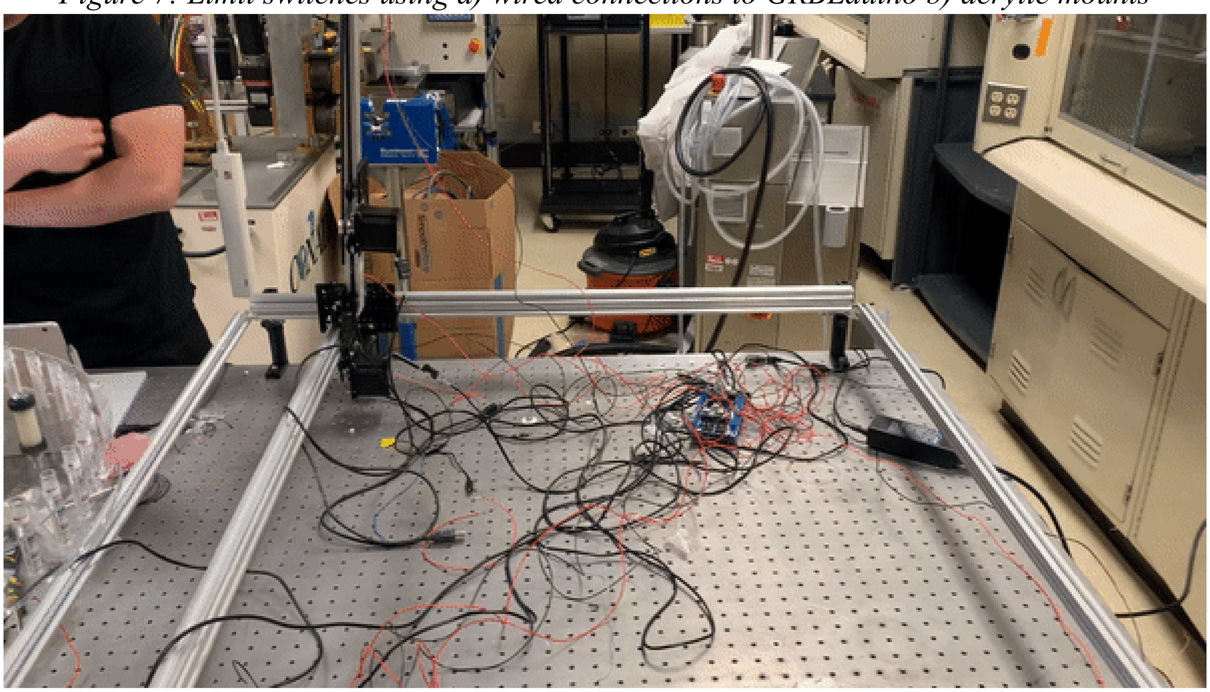


Figure 8: Calibration-homing sequence

2.4 Assembling the Frame

After setting up everything for the XYZ motion and attaching the pipetting module, we constructed a full 8020 frame and “flipped” our previous set-up we were using for troubleshooting purposes. Additionally, to anticipate possible troubleshooting, we did a full wiring reorganization and labelling. Below are 2 figures of our device set-up

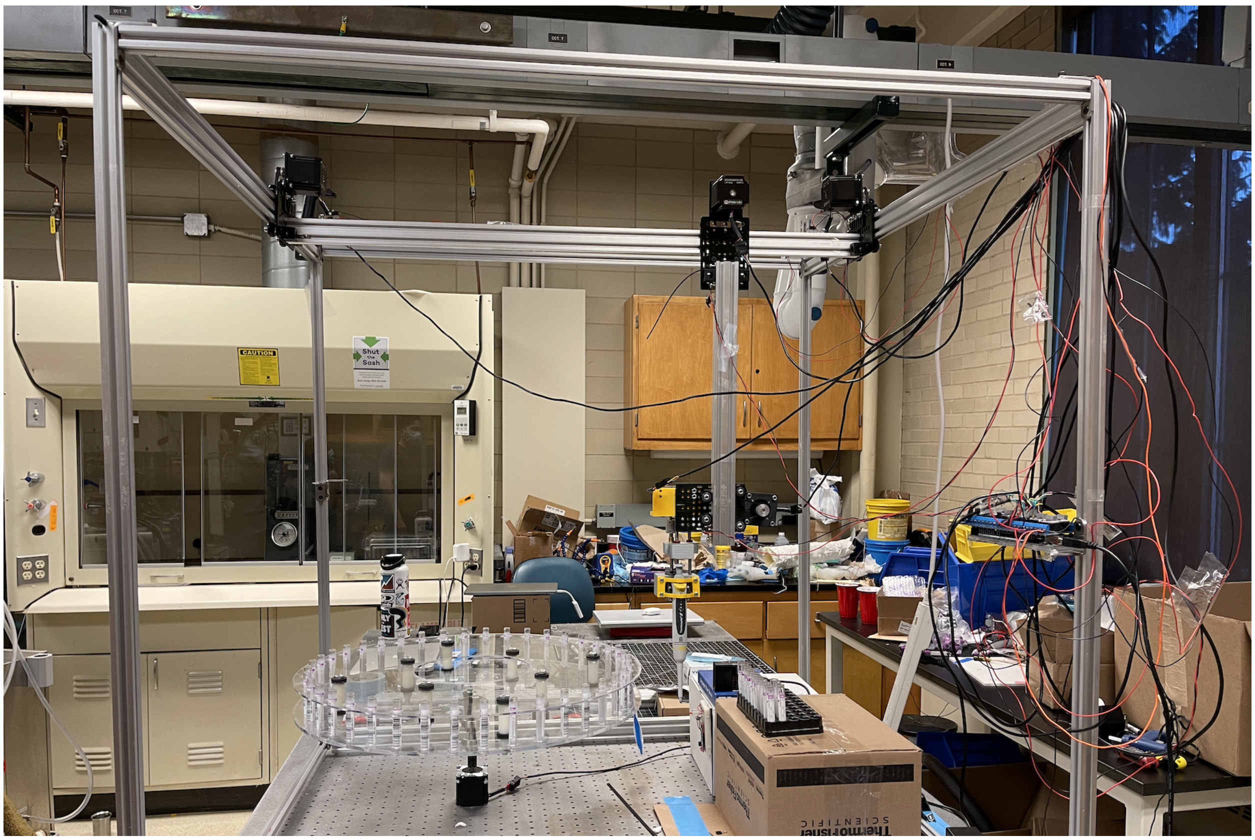


Figure 9: Side view of Full Robot Frame Assembly

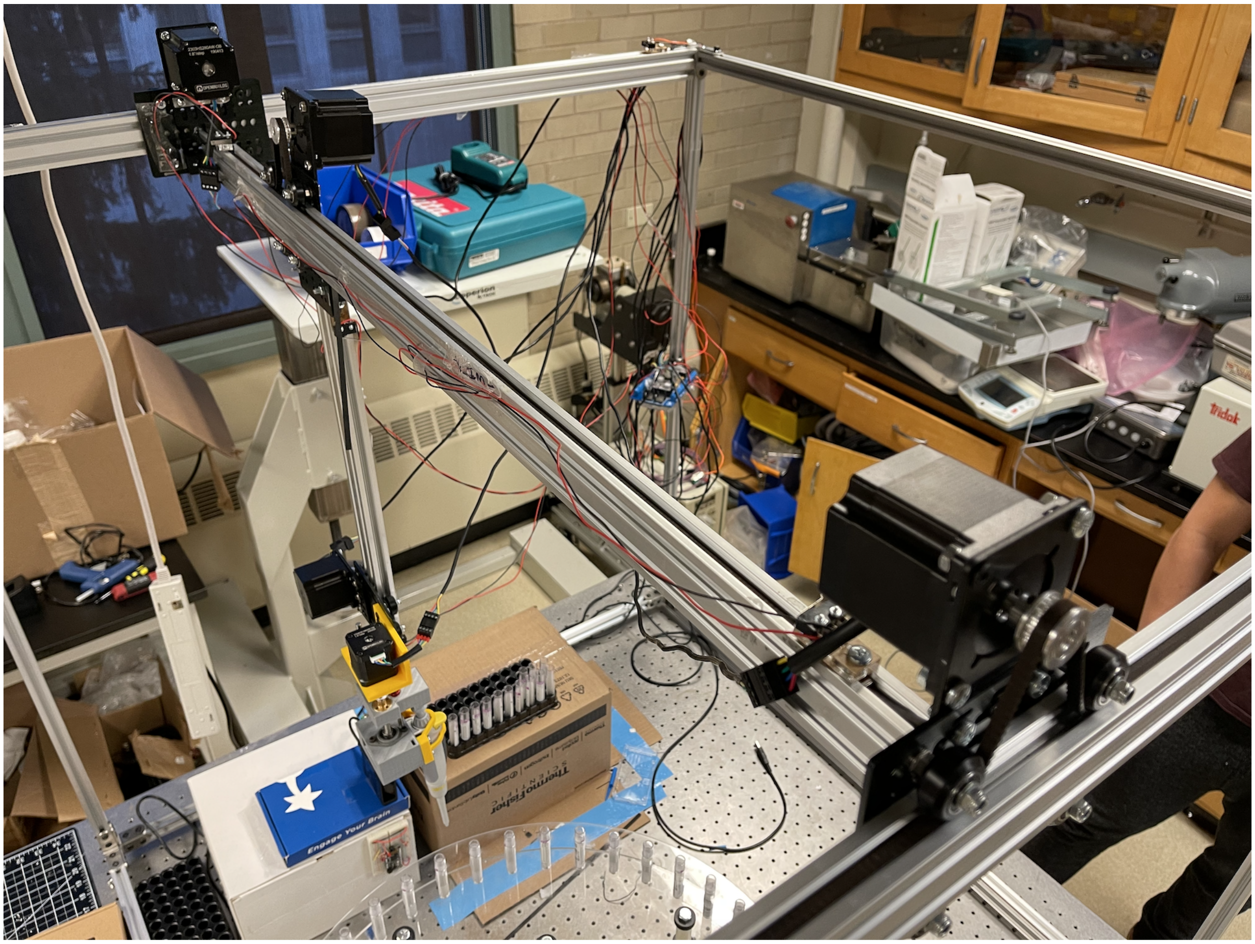


Figure 10: Top view of Full Robot Frame Assembly

3. Rotating Carousel Input Rack

3.1 Alternatives Analysis

The previous team proposed a “dual-functioning” pipetting and grabbing arm in their initial report (Design A), but we did not believe it was feasible given the way that they had constructed the pipetting arm. In addition, the design required having two separate stations and subsequently more distance travelled. Having reviewed their product, we came up with two possible alternative solutions that we believed were more viable given the previous work done on the project. Design B has a rotating “carousel” deck plate that rotates input tubes into one location where both scanning and pipetting occurs. Design C has a rotating scanner that scans tubes while the pipette navigates to each pipette individually. found that the previous team only worked on the pipetting function along with the xy-motion of the arm. We ultimately choose Design B based on our Pros/Cons analysis shown below.

Table 2: Pros/Cons for alternative models of the input rack design

	A. Grabbing Arm (from previous work flow)	**B. Rotating Circular deck plate**	**C. Central rotating barcode scanner**
Pros	- More space efficient - Tubes do not need to be changed as often - Allows for only liquid transfer in the work station	- Allows for more precise movements (only one degree of freedom through rotation) - Minimizes distance travelled and avoids pipette jogging over other samples reducing drip contamination even without a workstation - Eliminates need for a mobile barcode scanning station - Will be faster than a work-station model	- Tubes do not need to be changed as often - Eliminates need for extra barcode scanning station and saves further space by being mounted on the deck plate
Cons	- Difficult to implement 2 separate 3-D moving arms on rails in the same space - more travel distance of samples results in more chance of contamination - needs to have an additional bar code scanning station	- A circular plate with only tubes on the edges takes up more space than a gridded deck plate. Multiple rows can be considered, but this would complicate barcode scanning	- The pipette arm will have to travel to very specific locations along the circle that may be hard to implement using linear actuators - A circular plate with only tubes on the edges takes up more space than a gridded deck plate. Multiple rows can be considered, but this would complicate barcode scanning

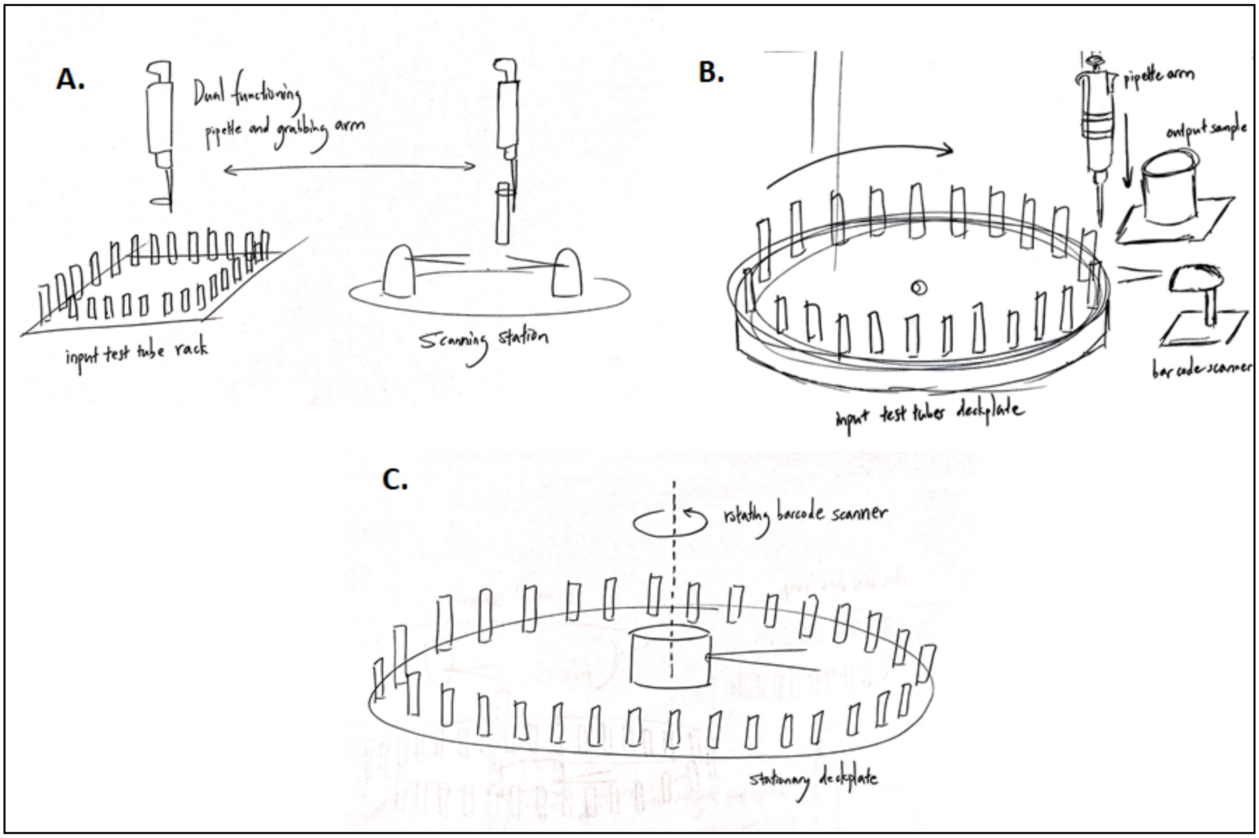


Figure 11: Sketches of Input Rack

After deciding on Design B, the next step in development was to choose the way that we will move and control the rotation of the bar code scanning. While originally we thought the design would proceed with simply a stepper motor, we realized that each stepper can only turn at multiples of 1.8 degrees. In order to reach even marginal specifications, the offset can only be by 0.772 degrees, so we came up with the following alternative solutions shown below in *Table 3*. Based on our analysis, we are leaning towards design choice C, the DC motor with the optical sensor, because this design both allows for adjustment for each tube and requires the least amount of raw materials.

Table 3: Pros/Cons for alternative ways to power input plate

	Pros	**Cons**
A. Continue Using Stepper Motor	• No sensors needed • Simpler design • Easier to replace input rack	• Input rack spacing must be 1.8 deg multiple (more precise) • Must calibrate often once constructed • Small deviations in the servo over time will compound
B. DC Motor and Hall Effect Sensor	• Compatible w/ multiple rack sizes • No need to manually calibrate • Adjusts angle for each movement	• Need to have a magnet under every single slot in rack • Difficult to replace input rack
C. DC Motor and Optical Sensor	• Compatible w/ multiple rack sizes • No need to manually calibrate • Adjusts angle for each movement	• Difficulty in sensor placement (will have to be in the same place as where we currently place barcode scanner) • Difficult to replace input rack

3.2 Carousel CAD Assembly

After deciding to move forward with the rotating carousel input rack design, the team developed a CAD assembly for the carousel by using SolidWorks software. Decisions for material selection were based on the carousel's intended use as well as a goal to minimize cost. *Figure 5* and *Figure 6* contain views of the assembly. Note that although the tubes within the assembly are capped, these tubes are to be placed manually in the carousel uncapped. Additionally, 1D barcodes are to be included vertically on each of the test tubes depicted on the figures below.

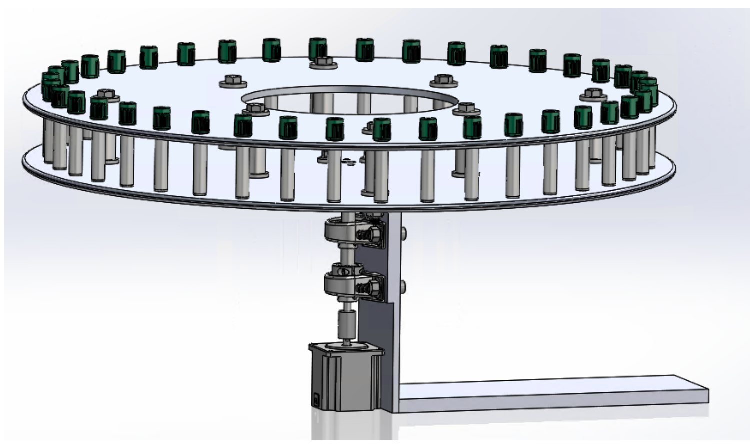


Figure 12: Carousel CAD assembly

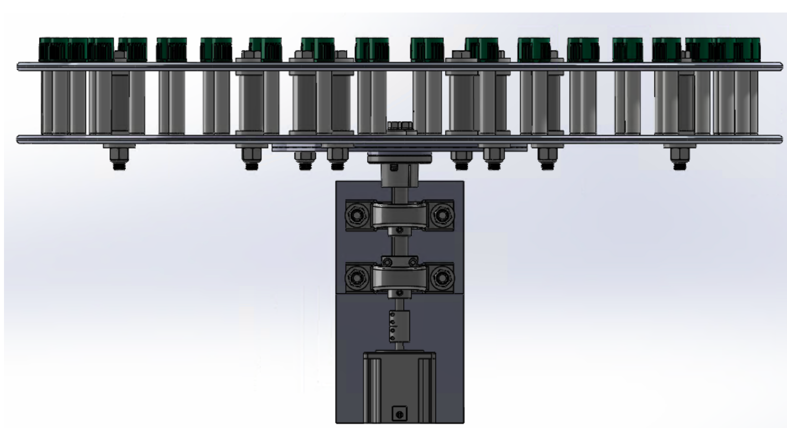
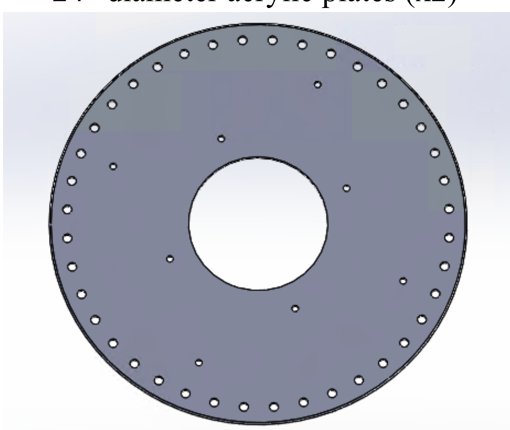
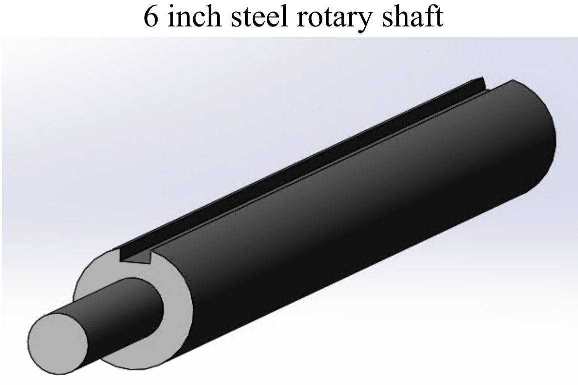
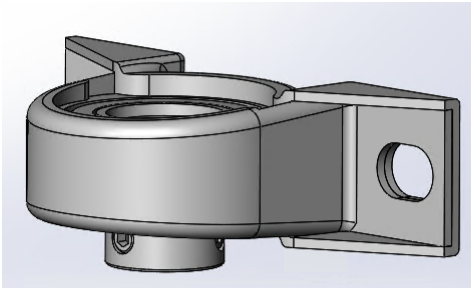
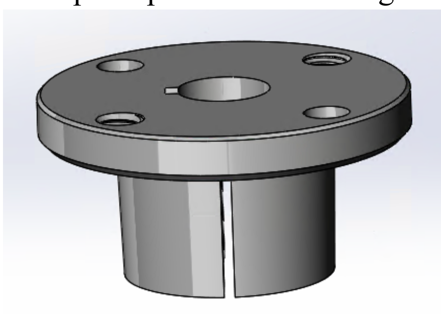
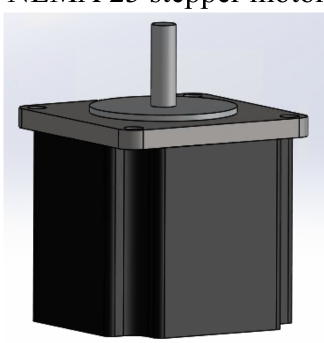
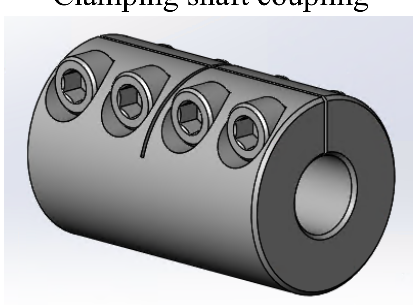


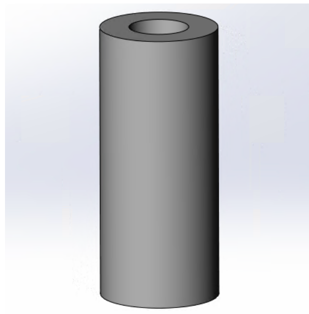
Figure 13: Front view of carousel CAD assembly

When formulating this design, we intended to minimize size as best we could in order to minimize weight and cost of materials while still maintaining the specifications previously mentioned in *Section 1.3.* Components were also selected with ease of manufacturing and assembly in mind. For example, laser cutting is a preferred manufacturing process due to its high accuracy and quick completion. For the time being, we intend for the carousel to be mounted on a steel L-bracket. This allows for the carousel to be readily tested independently. Below, *Table 3* lists the major components of carousel as well as their uses.

Table 4: Major components of input rack carousel

Major components	***Use***
24” diameter acrylic plates (x2) 	● Secures forty input sample tubes with clearance tube holes on the top plate and tighter tube holes on the bottom ● Top acrylic sheet is ⅛” thick and bottom acrylic sheet is ¼” thick in order to provide extra support to tubes ● Material is relatively lightweight and inexpensive
6 inch steel rotary shaft 	● Acts as the axis of rotation for the carousel ● ½” diameter shaft will be turned on the lathe to match the coupling component of the motor shaft of ¼” diameter
Mounted sealed steel ball bearing (x2)	● Provides rotational support for the rotary shaft

	● Contains external mounting block which houses the rotational bearings ● Inclusion of two bearings prevents shaft misalignment
Split tapered steel bushing 	● Connects the bottom acrylic plate to the steel rotary shaft with cap screws
NEMA 23 stepper motor 	● Sets the rotational motion of the carousel with a stainless steel shaft ● Allows for multiples of 1.8 degree step angle
Clamping shaft coupling 	● Links the NEMA 23 motor shaft to the rotary shaft using clamping screws ● Grips evenly around shaft, providing better holding power than a set screw coupling would
Nylon unthreaded spacers	● Allows for two inch gap between the top and bottom acrylic plates of the deck plate so an external barcode can access test tube barcodes

	• Resists abrasion better than metal spacers • Coincides with cushioning washers at interface with acrylic plates

Steel is the material of choice for most components due to its high strength and compatibility with the NEMA 23 motor's steel shaft. Cushioned washers are to be used on the interfaces of acrylic surfaces to dampen vibration from motion and resist wear from rubbing with metal surfaces. Additionally, nylon insert lock nuts are desirable to prevent bolt threads from coming loose with the motion of the carousel.

3.3 Mocking up Input Rack Barcode Scanning

Before we committed to building the deckplate, we constructed a mock-up of the rack using cardboard to get some more insight on the barcode scanning with the rotating test tubes.

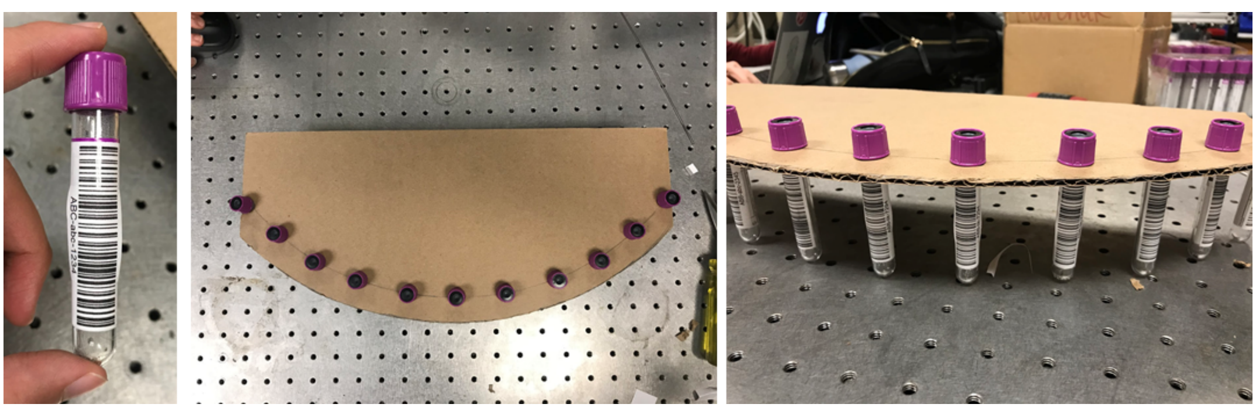


Figure 14: Tube with Barcode Label and Input Rack Mock-up

1D barcodes were computer generated and taped to sample test tubes. The barcode is 2 inches in length, and covers about one third of the test tube.

We placed the test tubes into the carousel mockup, which holds up to 11 test tubes (only a section of the carousel is made). The proportions of the mockup are made exactly to scale to our CAD (24 inch diameter, 1 inch buffer between the edge and the sample, 1.73 inch arc length between tubes). The carousel is then stabilized to the lab table, with the barcode scanner fixed at one position.

The purpose of this specification testing is to determine the optimal distance between the barcode scanner and the target test tube for scanning accuracy. At different distances (1.5 inches, 2 inches, 2.5 inches), the carousel is spun at a reasonable speed, and the number of barcode detections are recorded.

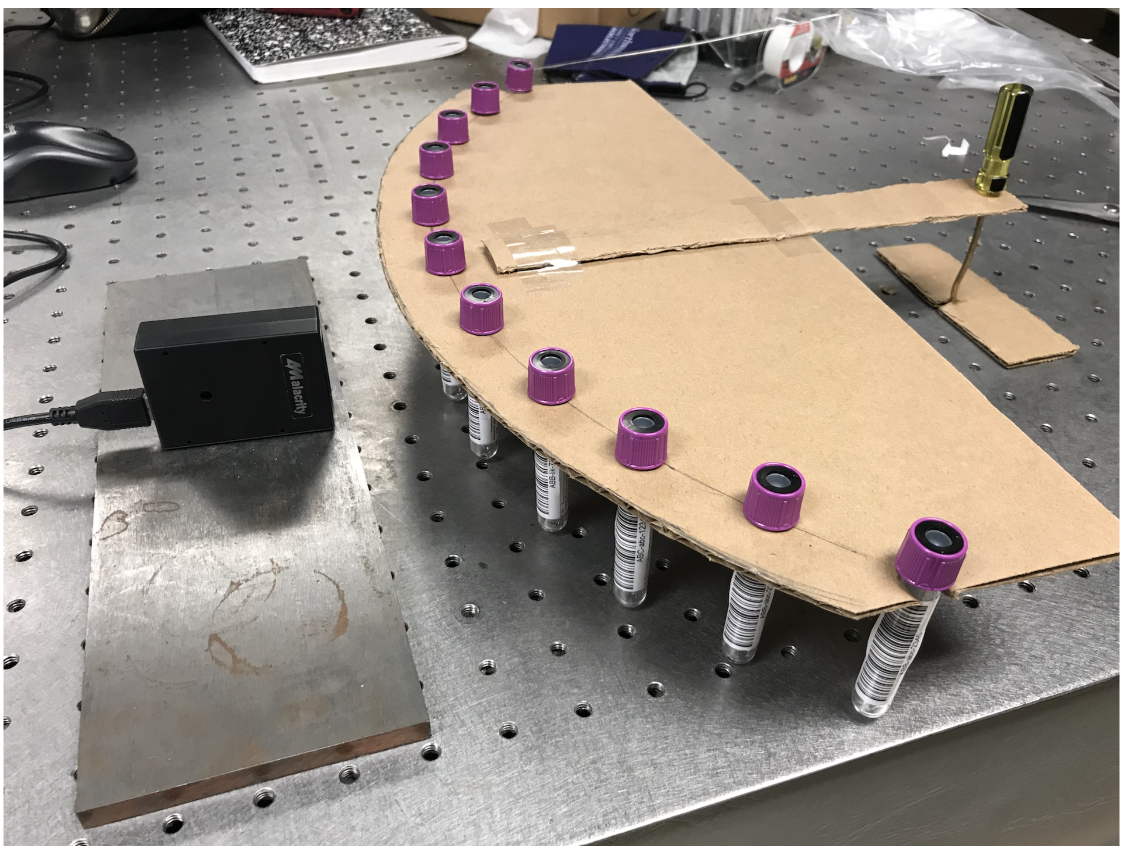


Figure 15: Scanning Test Set-up

We found that the optimal distance is **2 inches**, at which almost all the test tubes were scanned with ease.

Table 5: Scanning Test Data

	Distance												
Tubes	1in	1in	1in	1.5in	1.5in	1.5in	2in	2in	2in	2.5in	2.5in	2.5in	3in
T1	no	no	no	ABB-abc-2345	no	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	no	no	no	no
T2	no	no	no	no	no	no	ABC-abc-1234	ABC-abc-1234	ABC-abc-1234	no	ABC-abc-1234	ABC-abc-1234	no
T3	no	no	no	no	no	no	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	no	no
T4	no	no	no	no	no	no	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	ABB-lik-7294	no
T5	no	no	no	no	no	no	no	ABB-lik-7294	no	no	ABB-lik-7294	no	no
T6	no	no	no	ABB-abc-2345	no	no	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	no	no	ABB-abc-2345	no
T7	no	no	no	no	no	no	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	ABB-abc-2345	no
T8	no	no	no	no	no	no	ABC-abc-1234	ABC-abc-1234	ABC-abc-1234	no	no	ABC-abc-1234	no
T9	no	no	no	no	no	no	ABC-abc-1234	ABC-abc-1234	no	ABC-abc-1234	ABC-abc-1234	ABC-abc-1234	no

A future test we plan on doing is to find the allowable angular offset allowable for the tubes in each individual slot. If this allowable offset is insufficient, a future testing idea is having barcodes on both sides of the tube.

3.4 Manufacturing of Final Rotating Deck Plate Module

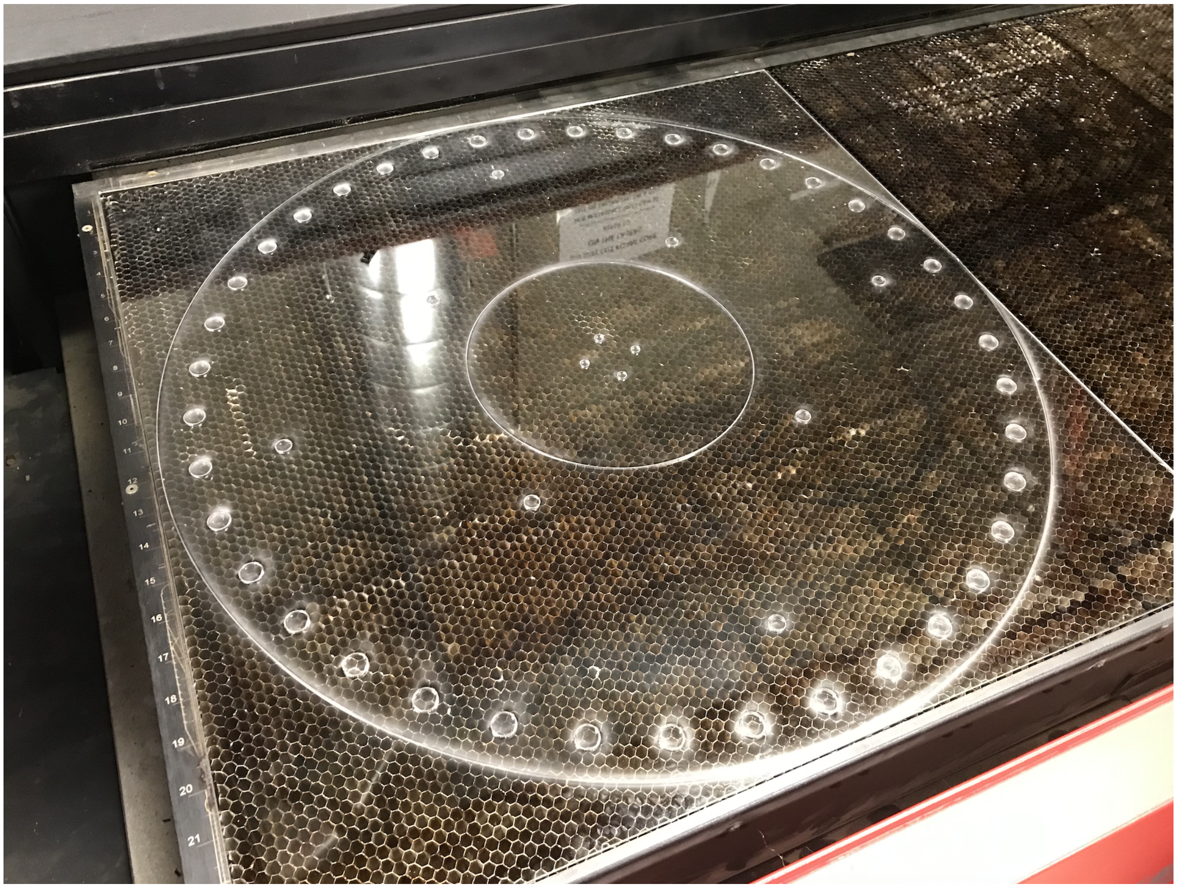


Figure 16: Laser Cutting Acrylic

The upper and lower deck plates are laser cut from acrylic sheets. To figure out the diameter of the test tube holders, we laser cut slots with different diameters (13mm, 12mm, 11mm) on sample acrylic sheets to test their fit with the test tubes. Based on the results, we settled on 13mm for the upper acrylic sheet, and 11.5mm for the lower.

After assembly of the deckplate, we see that the test tubes fit relatively well into the input slots. There is close to no shifting of the tubes upon carousel rotation, nor is there any apparent deflection of the acrylic deckplate given the minimal support at the center. We plan to design a better support at the connection point as well as a bottom supporting piece for the motor to strengthen carousel durability and lifetime.

Figure 17: Completing Assembly of Input Rack

3.5 Integration of Barcode Scanning with Prototype

To test the barcode scanning, we connected the carousel motor to the GRBLDuino board and simulated rotation at different step sizes and frame rates. We set up a temporary barcode scanner station at the same height as the sample test tubes, and placed the scanner 2 inches away from the test tubes as previously measured.

The completion of scanning at the speed shown below is 100% through 5 trials, which satisfies our target specification. We have yet to determine the ideal speed at which the carousel should rotate, which we will address in the upcoming quarter.

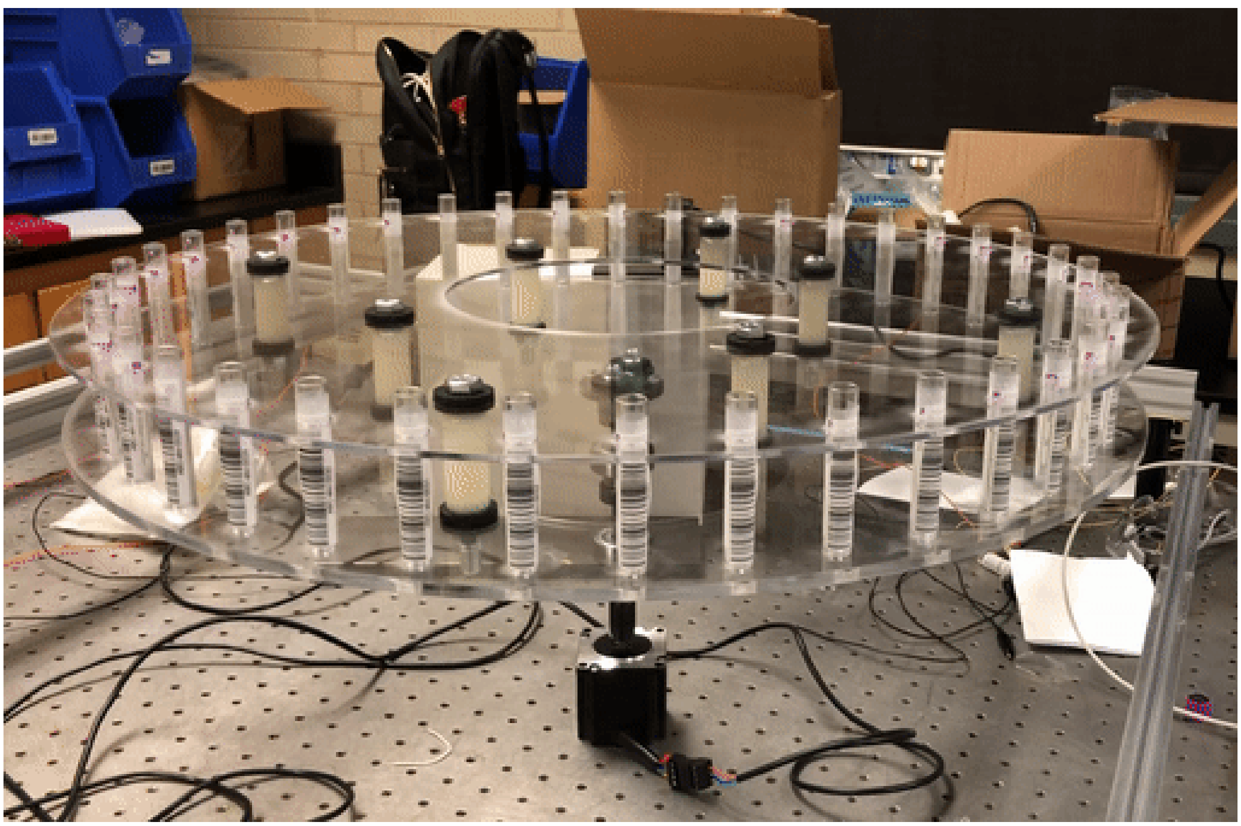


Figure 18: Reliable Automatic Barcode Scanning

3.6 Optical Limit Switch Development

After setting up our frame with mechanical limit switches, we moved on to brainstorming how to home the input carousel deck plate to a starting position. This capability offered the opportunity to simplify the G-Code to work in reference to an understood starting configuration. We figured having one tube position be our first position would best so we reviewed what type of switch system worked for our purposes. After assessing our needs for our design and consulting with Professor Nicholas Marchuk, we settled on a light based sensing system because it required the least complicated method to integrate with our system. We simply needed one light sensor and a method to communicate with the limit switch wiring for the GRBL Mega Shield. We experimented with an RGB Color sensor first but quickly realized it would require extra housing to avoid a room's natural lighting from interfering with it. Then we moved on to a Reflective IR sensor. This Reflective IR Sensor, sourced from Adafruit, works by pairing an IR LED and an IR phototransistor within its design. The IR LED bounces IR off objects to determine their reflectivity. Per the manufacturer's specifications, light colored material bounce the light allowing us to detect it as opposed to dark colored material that absorbs the IR light. In practice, we observed that the device was best effective at selectively detecting light colored objects at a distance of approximately 5 mm. For our purposes, we utilized an additional development board (Feather M4 Express) to output a logic High or Low voltage based on whether the sensor detects something. This output is meant to be wired to the B axis limit option on the GRBL Mega Shield. Figure XX below shows our breadboard setup that we used to test both the RGB Color Sensor and Reflective IR Sensor. We equipped the device with the below:

```
void setup() {
  pinMode(10, INPUT); // sensor pin
  pinMode(9, OUTPUT); // output pin
}

void loop() {
  int sensorVoltage = digitalRead(10); // sensor reads 0 when pressed
  if (sensorVoltage == 0) {
    digitalWrite(9, HIGH);
  }
  else {
    digitalWrite(9, LOW);
  }
}
```

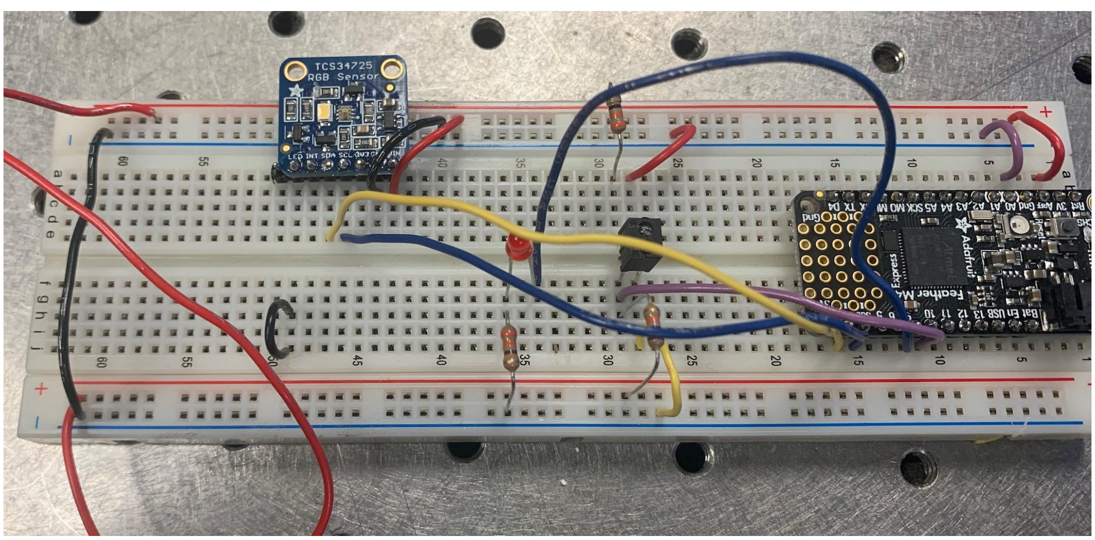


Figure 19: Breadboard setup for testing sensors

We positioned the sensor underneath the barcode scanner as seen below and mocked up a white tag underneath the starting position to see how the sensor works in practice.

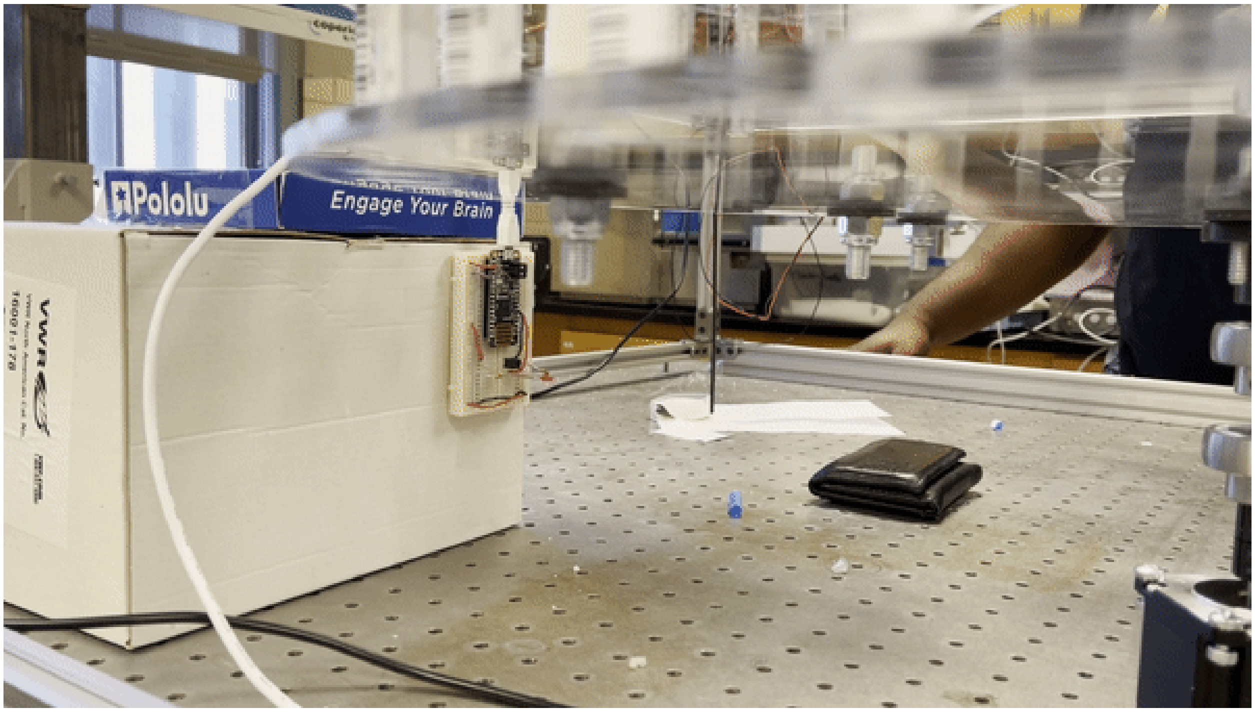


Figure 20: The sensor setup in the frame (see LED on board)

4. Controller Mechanism - GRBL/Arduino

4.1 Overview of Fall Troubleshooting History

We received from the previous team a GRBLDuino Mega Shield integrated with Arduino Mega2560 controller board that should have been equipped with compiled source code ("GRBL") that allowed us to send G Code to it from a user platform called Universal G Code Sender (UGS). However, when we tried to send G Code from the platform, we were met with an error that stated "Grbl has not finished booting". Referring to the troubleshooting guidelines provided by the developers of Universal G Code Sender, we double checked to make sure that common issues found when this error occurred were addressed. We checked the baud rate to see if it was 115200 as specified for the latest GRBL source code update. We made sure we were connected to the correct port on our laptop devices. We made sure all drivers required for the controller were installed.

The next step was to make sure GRBL is properly flashed on our controller. The authors of the source code indicated that this was already accounted for through a collaboration they established with Arduino. However, in the case that there were issues, they provided a .hex file that could be manually flashed. We attempted this and still were not able to get it working. As a last resort, we removed all previous UGS settings and reinstalled it. This still did not work.

We noticed that there were multiple versions of the grbl code available on Github and some were not designed for our controller board so we searched for alternative source codes on Github that would work specifically for our board. We found an open source code named "Grbl8cMega2560" that compiled and allowed us to send G Code to the board. We were able to get the stepper motors to turn in the XY plane but this source code would only allow us to go in a positive direction for both X and Y axis. It also limited us as it only enabled 3 axes out of the 6 available on the GRBLDuino Mega Shield.

From the datasheet provided by the manufacturer of the GRBL Shield, we found GRBL source code named "grbl-Mega-4axis" marketed as flashed and compiled for 6 axis. However, we ran into the same error as the original grbl source code above. In the next section (4.2), we discuss how we addressed this issue.

4.2 Rebuilding GRBL Shield

We ordered a new GRBL Shield and rebuilt the setup ourselves so we can account for any unseen errors we are not noticing right now with what we were given. We currently have access to 6 axes on the shield. The X axis utilizes 2 of them while the Y and Z axes utilize 1 respectively. The pipetting mechanism and the input rack also utilize 1 respectively.Due to our struggles with the Shield at the time, we intended to test the Input Rack motion and the encoder system we choose by applying to a separate Feather M4 Express controller board so as to not hinder progress. The code was to be done in the Arduino IDE making it easily transferable to the Arduino Mega2560 controller board once the XYZ motion becomes fully operational.

However, we were able to bypass this process because after deleting all the previous source codes and configuration files from the Arduino IDE library on the computer we were using, the manufacturer's source code ("grbl-Mega-4axis") compiled properly. We were then able to use the Setup Wizard in Universal GCode Sender to establish the platform as a mode of communication with the Shield via GCode. We were able to Clone Axis A to Axis X for better stability and instruct axes, X,Y,Z and A to perform specific tasks. At the end of Fall quarter, we could not communicate with Axes B and C and were unsure if this was because the GCode we were using to try and communicate with them was incorrect or if the Universal GCode Sender platform is not robust enough to delegate to six different axes. However, upon continuing with work at the beginning of Winter Quarter, we realized that the source code did allow for communication with B and C axes. After connecting to the Arduino Mega2560 development board in Universal G-Code Sender, a message containing the device settings is printed in the console as shown in Figure 21 below. The settings options printed out in the UGS console for this version of the grbl source code was simply not specified by the authors.

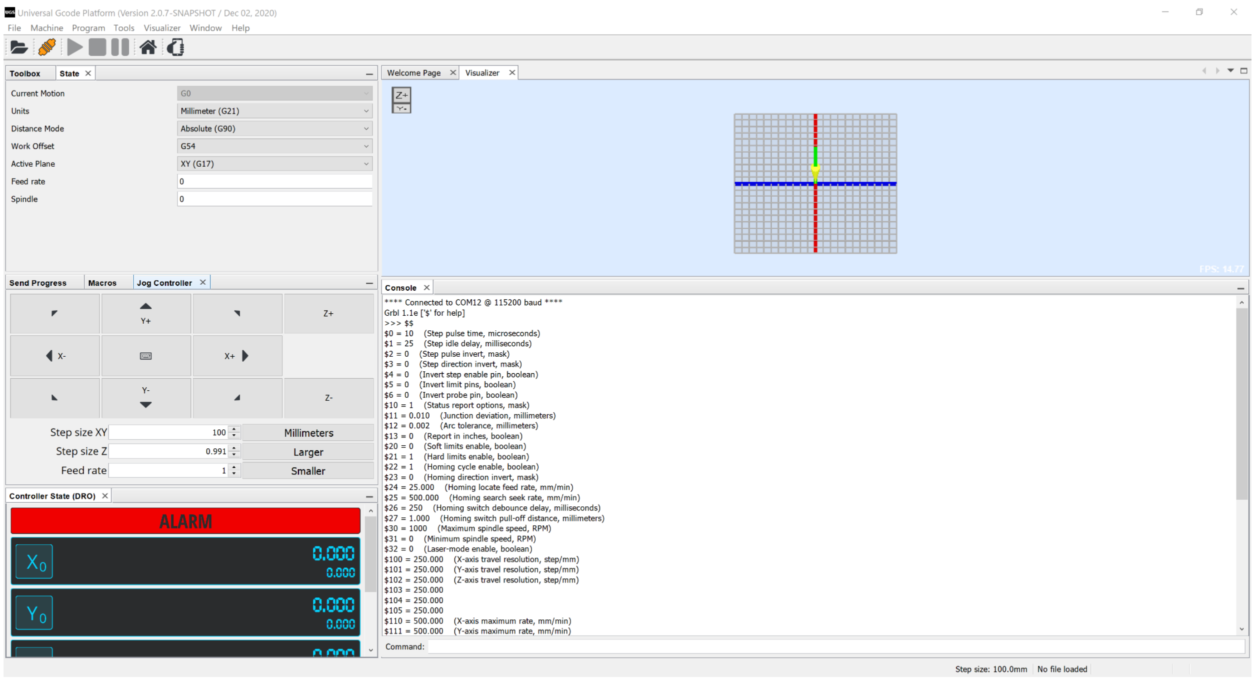


Figure 21: GRBL Software Device Settings Readout

4.3 Calibration of XYZ and Rotating Input Rack Step Size

In order for a commanded movement set in the UGS Platform to be matched with the hardware's movement in realtime, our team had to complete step calibration for each motor used in the robot. We completed this calibration process for the NEMA 23 motors which provide motion along the X, Y, and Z axes along with the NEMA 23 motor which allows for the input rack's rotational motion.

XYZ step calibration:

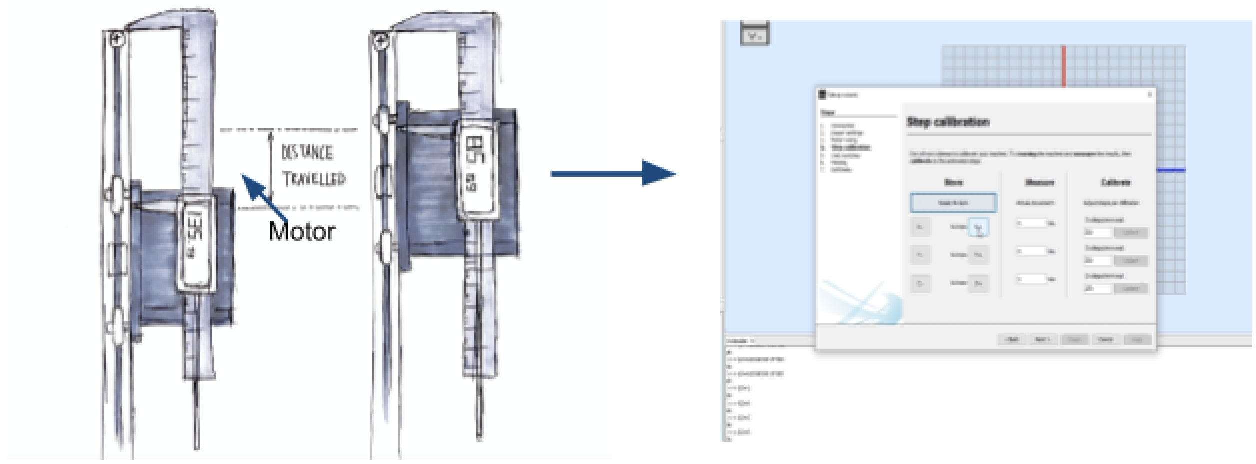


Figure 22: XYZ Calibration

By adjusting the step size for XYZ motion, we were able to complete calibration for XYZ motion. We did this by fixing the step size along each specific axis using the UGS Platform's Startup Wizard. First, we selected some arbitrary initial step size for one axis. Then, we commanded the motor to move one step in a direction along the same axis and measured the resulting displacement. This was repeated for each axis. The following numbers were used for calibration:

Input rack step calibration:

In a somewhat similar manner, we were able to calibrate the input rack motion so that the commanded movements in the UGS Platform matched with hardware movements in real time. The deckplate was calibrated so that with each step, a new test tube will be scanned by the automatic, hands free barcode scanner at a location which is deemed to be the target position. From this position, a patient sample can be extracted using the pipette module.

The following video displays this mechanism after the input rack was calibrated. Each audible beep gives feedback to the user that a new test tube is scanned after moving to the target position.

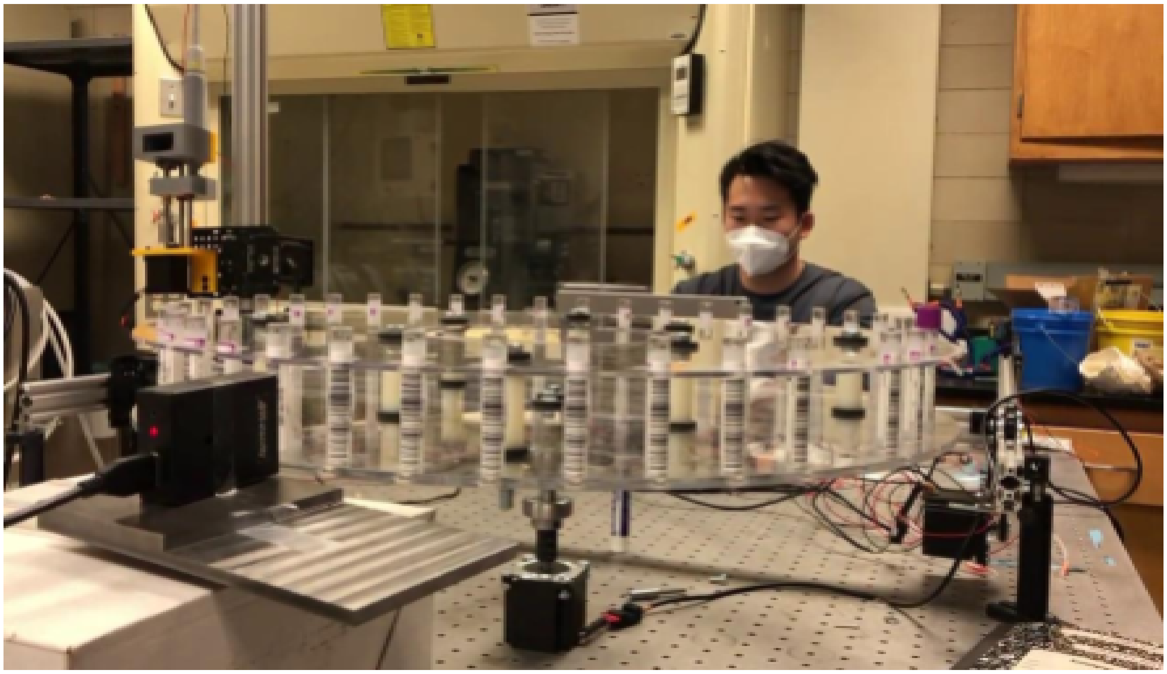


Figure 23: Deckplate calibration (Double click and press play to view video)

5. Testing

5.1 Completed Z-motion Testing

5.1.1 Z-Motor Positional Accuracy Testing

Having assembled and calibrated our pipetting robot, we want to test if the pipetting arm is able to travel accurately to the target test tubes to conduct sample pooling. Because XY motion testing had already been completed by the previous team, we wanted to focus on the Z motion testing.

We changed the positional accuracy protocol from the previous team to accommodate for testing larger distances since the error measured will be more applicable to the actual functioning of the machine. The specifications to be tested for positional accuracy can be visualized with Figure 24 below:

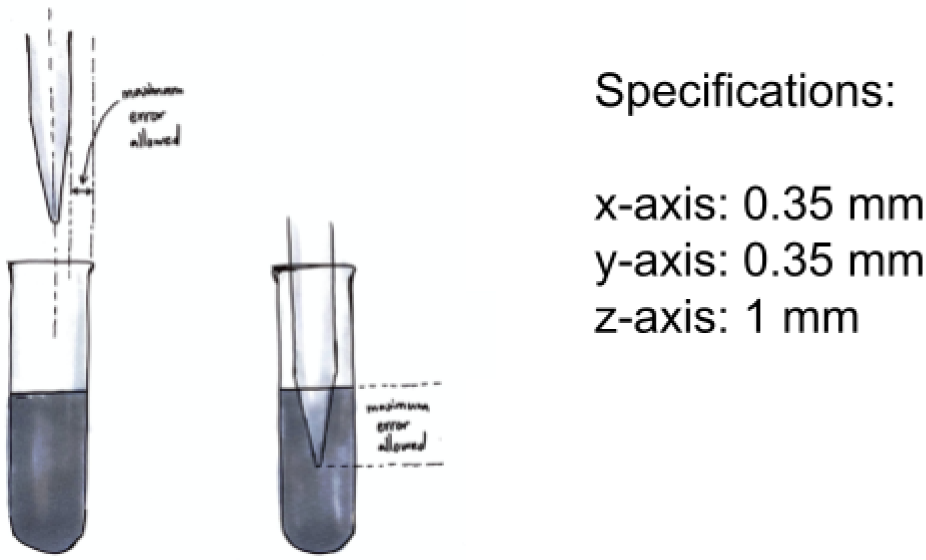


Figure 24: Visualization of Specifications for Testing

For the x and y axis, the margin of error is the space between the pipette tip and the edge of the test tube. Having to also consider the accuracy needed to locate and pick up new pipette tips, the xy specification comes out to 0.35 mm. For the z axis, the margin of error correlates with the length of the sample tube. If we aim to insert the pipette to the center of the blood sample (assuming the test tube is half filled), the z specification then comes out to 1mm if we air on the conservative side.

We used digital calipers instead of the dial indicator to measure positional accuracy since it can measure large distances. This also allows us to test positional error at any Z coordinate instead of just (0,0,0), so we changed the experimental set up accordingly.

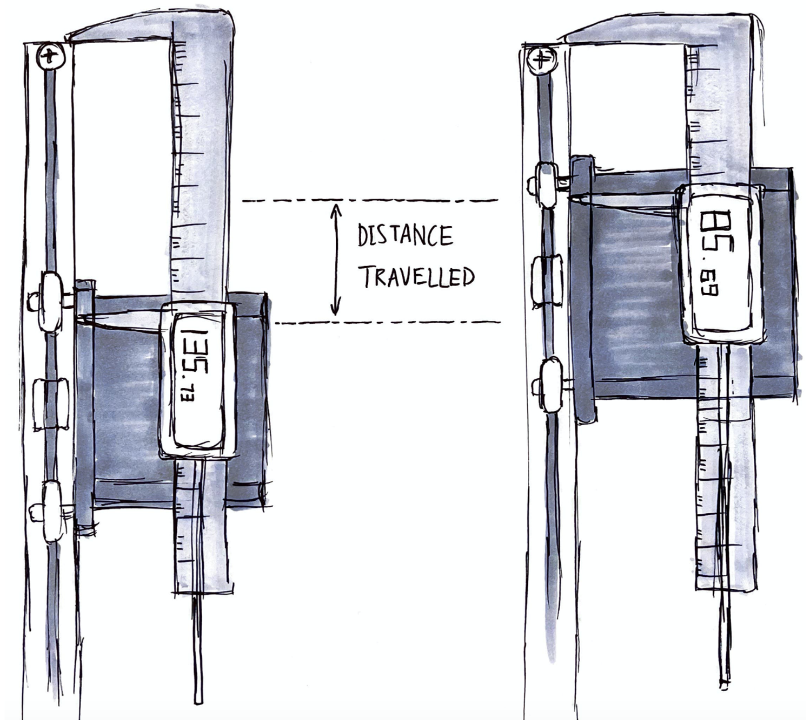


Figure 25: Z-positional Testing Set-up

For z-positional testing, we measured the displacement error of the z motor as it travelled to two coordinates, one 50 mm away from the home position and another 25 mm away. We set the home position at coordinate G175 since we had to use a digital caliper to take our measurements, and this allowed us to measure the distance between the z motor and the end of the z-rail to infer positioning. Because it is difficult to determine the exact positioning of the motor along the z-axis (with the motor basically hanging upright), we used the set distance between the end of the z-rail and the motor at the different positions as the standard to measure positional accuracy. G175 equates to a distance of 135.4 mm. G200 equates to a distance of 110.4 mm. G225 equates to a distance of 85.4 mm.

26 trials are conducted for each coordinate (G200 and G225), and the positional error for each trial is calculated as the following:

G175 → G200 (50 mm travelled):

Trial	initial distance (mm)	final distance (mm)	distance traveled (mm)	positional error (mm)
1	135.67	110.42	25.25	0.25
2	135.42	110.4	25.02	0.02
3	135.47	110.47	25	0
4	135.58	110.58	25	0
5	135.52	110.42	25.1	0.1
6	135.4	110.48	24.92	-0.08
7	135.41	110.56	24.85	-0.15
8	135.45	110.49	24.96	-0.04
9	135.39	110.63	24.76	-0.24
10	135.4	110.5	24.9	-0.1
11	135.46	110.56	24.9	-0.1
12	135.54	110.72	24.82	-0.18
13	135.7	110.51	25.19	0.19
14	135.32	110.54	24.78	-0.22
15	135.4	110.55	24.85	-0.15
16	135.41	110.56	24.85	-0.15
17	135.39	110.59	24.8	-0.2
18	135.42	110.54	24.88	-0.12
19	135.46	110.5	24.96	-0.04
20	135.41	110.58	24.83	-0.17
21	135.44	110.5	24.94	-0.06
22	135.44	110.54	24.9	-0.1
23	135.38	110.43	24.95	-0.05
24	135.41	110.44	24.97	-0.03
25	135.42	110.54	24.88	-0.12
26	135.42	110.46	24.96	-0.04

G175 → G225 (50 mm travelled):

Trial	initial distance (mm)	final distance (mm)	distance traveled (mm)	positional error (mm)
1	135.38	85.48	49.9	-0.1
2	135.61	85.49	50.12	0.12

3	135.39	85.38	50.01	0.01
4	135.73	85.69	50.04	0.04
5	135.41	85.46	49.95	-0.05
6	135.4	85.28	50.12	0.12
7	135.38	85.48	49.9	-0.1
8	135.49	85.43	50.06	0.06
9	135.36	85.42	49.94	-0.06
10	135.33	85.46	49.87	-0.13
11	135.43	85.46	49.97	-0.03
12	135.48	85.47	50.01	0.01
13	135.37	85.39	49.98	-0.02
14	135.48	85.37	50.11	0.11
15	135.34	85.42	49.92	-0.08
16	135.39	85.41	49.98	-0.02
17	135.38	85.36	50.02	0.02
18	135.35	85.36	49.99	-0.01
19	135.28	85.54	49.74	-0.26
20	135.37	85.35	50.02	0.02
21	135.37	85.36	50.01	0.01
22	135.44	85.35	50.09	0.09
23	135.3	85.34	49.96	-0.04
24	135.38	85.28	50.1	0.1
25	135.43	85.38	50.05	0.05
26	135.33	85.35	49.98	-0.02

The positional errors are presented in a box-and-whiskers plot:

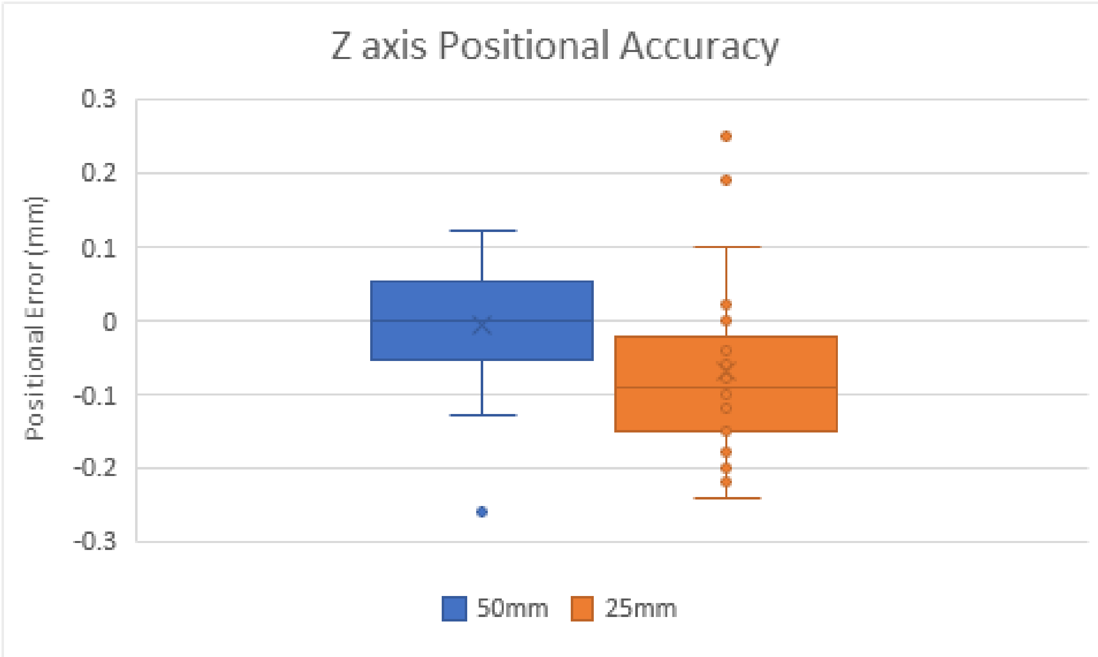


Figure 26: Z-axis Positional Accuracy

We see that the displacement errors mostly fall within 0.1 to 0.2 mm from the target position for both coordinates tested. Because our specification for the z-axis (1 mm) is significant greater than 3 times the standard deviation of both sets of data (0.085 mm and 0.114 mm, respectively), this suggest that only 0.1% of the time would a positional error occur that falls outside our marginal specification and significantly impact the sample pooling process. From here, we conclude that the z-motor positional accuracy satisfies our requirement.

5.1.2 - Z-Motor Repeatability Testing

Next, we utilized the homing data to G175 to analyze the repeatability of the z-motion along the z-axis. We normalized the homing distance between the end of the z-rail and the z-motor to 135.4 mm and plotted the positional error against the consecutive trials conducted.

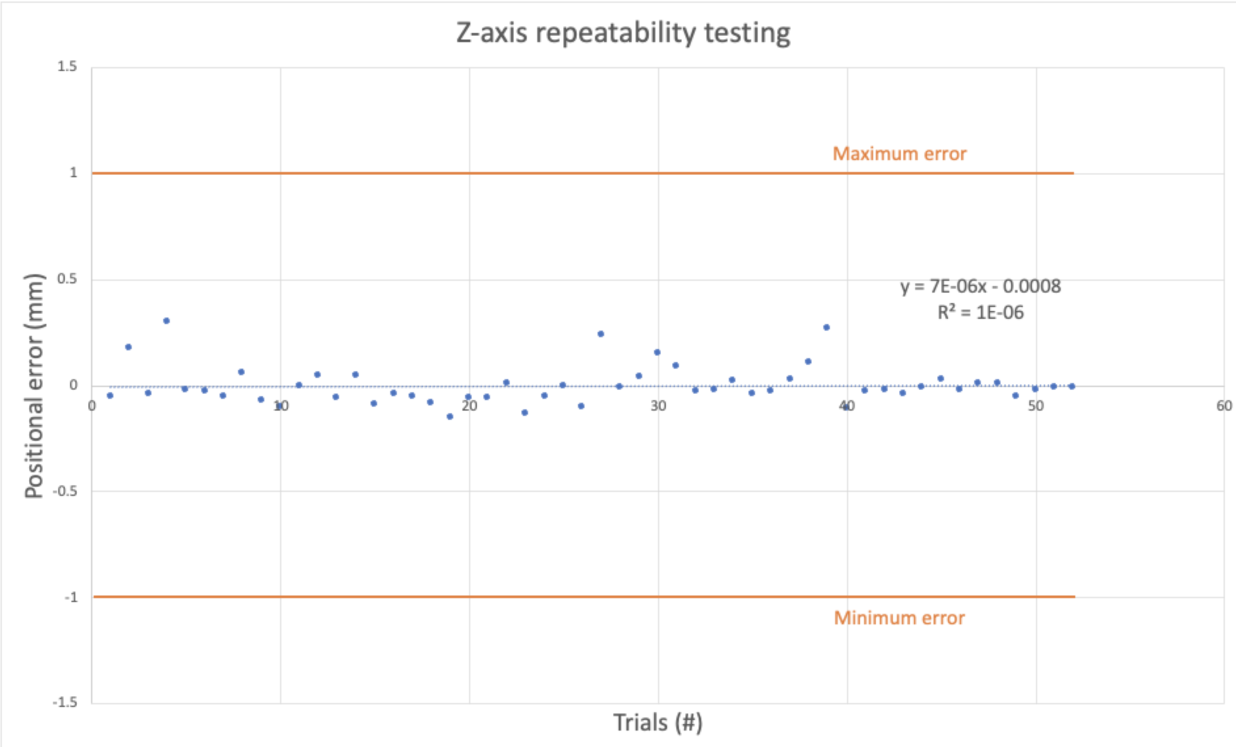


Figure 27: Z-axis Repeatability Testing

We see here that after 52 consecutive trials of homing the motor to coordinate G175, there is no significant drift of the homing position even though each individual trial has its own small positional error. The slope of the best line of fit is 0, which suggests that these errors do not accumulate over time. This means we should be able to consistently pool all the samples from the entire deckplate without any recalibration, which saves time and improves our throughput rate.

5.2 Planned Testing: Input Rack Accuracy, Repeatability, and Calibration

Before our testing plans were interrupted by the issues we were facing with our controller mechanism, we planned to complete input rack accuracy and repeatability testing. These tests can be conducted only after the next team is able to troubleshoot through those issues with the GRBLduino.

5.2.1 - Deckplate Positional accuracy

This test would be used to prove the accuracy of the rotational deckplate component of our pipetting robot system. If successful, this would show that the smallest margin which the rotational deckplate can accurately move is 98% of the time. Fifty trials should be conducted for each of three experiments that measures the margin of error at different lengths of rotation.

In order for the robot to successfully complete pooling of all 40 samples on the deckplate (one full rotation), the accumulated margin of error (distance between experimental and actual deckplate positioning along its edge) needs to fall within half of the width of the sample test tube, assuming that the pipette arm starts sampling at the center of test tube #1.

The diameter of the test tube is 12.50 mm. Therefore, the positional error of one deckplate rotation needs to be less than 6.25 mm, and the margin of error allowed for the movement between adjacent tubes is $\frac{6.25\ mm}{40\ tubes}$= 0.15625 mm along the edge of the deckplate. This error in radians comes out to $\frac{0.15625mm}{279.4mm}$= 0.00055923407 rad, since the radian measure of a central angle θ is defined as the ratio of subtended arc length to the radius of the circle.

To quantify the actual positional error in rotation, the deckplate should be commanded to rotate at different angles. For each commanded rotation, the deviance between the target scanning position and the actual position of the target test tube should be measured. The shortest distance (d) between the two points is measured using a digital caliper, from which the error in radians can be calculated using the relationship shown in *Equation 1*.

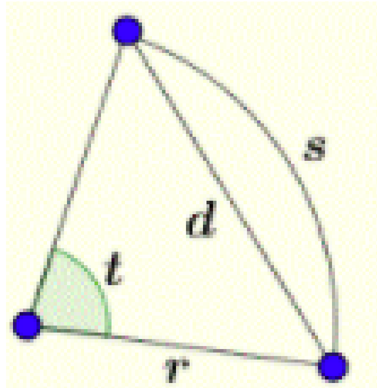


Figure 28: Illustration to supplement relationship between arc length (s), radius (r), chord length (d), and angle (t). Note that t is in radians.

$$d = 2 * r * sin(\frac{t}{2})$$

Equation 1

d is the measured distance between the target scanning position and the center mark of the target test tube. *r* is the radius of the deckplate (measured as distance between the center of the deckplate and the center of the target test tube). *t* is the rotational error in radians.

5.2.2 - Deckplate Repeatability Testing

This test would be used to prove the repeatability of the rotational deckplate component of our pipetting robot system. If successful, this should show that the rotational deckplate can repeat sample pooling through 600 consecutive trials with a tolerable margin of error. Measure the error in radians and calibrate the deckplate back to starting position (either manually or with the calibration program if written). Repeat the trial 600 times to prove consistency.

5.2.3 - Deckplate Calibration Testing

The optic sensor is to be used as the starting position of the deckplate (Test tube 1). Optic detection with this sensor can also be used to calibrate the rotational deckplate.

The calibration test should be composed of trials at different initial deckplate positions. A program should be written which will run, telling the deckplate to rotate until the starting position is reached via optic detection. The end position should be recorded, and error should be measured as the distance between the target starting position (directly across the 1D barcode scanner) and the actual position of test tube 1. Error in radians can be calculated similarly to that of rotational accuracy testing.

5.3 Planned Testing: Speed Testing of Hypothetical Throughput

Below is the framework by which our speed testing will be performed.

Purpose of Test

The test will be used to ensure that the speed of the movement is sufficient to move through our workflow. If successful, the test will show that we can move the pipetting arm through the entire workflow within 40 seconds, which translates to a throughput rate of 90 samples per hour.

Materials Needed

Timer

XYZ axes with pipette or weight attached

Methodology

After writing out the different 3-D locations of the workflow in Gcode. We will send the command to our controller. At the same time, we will begin timing. Timing will stop after the controller has moved back to its original position. This test will be in 40 trials initially, but more testing will be done, time permitting.

Data Collection

Data will be collected on an excel sheet labelled [date]_xyzspeedtests. The results will then be fit to a normal distribution and have its confidence intervals returned.

6. Failure Mode and Effects Analysis

In the following table, we address the potential failures that could occur in each component of our pipetting robot: the motor shield, the rotating deckplate, and the pipetting arm. We identify all that could potentially go wrong, the relative effect of each failure, and corrective actions that can be taken to reduce severity or failure frequency.

Motor Shield:

Failure Mode and Effects Analysis									Sheet No. ___1_ of					
() System (X) Subsystem () Component	Description: The motor shield is the microcontroller system that integrates the various motion axes within the product				Comments:									
1	2	3	4	5	6	7	8	9	10	11	Action Results			
Function	Failure Mode	Effect(s) of Failure	Mechanism(s)/ Cause(s) of Failure	Current Controls	S	O	D	RPN	Recommended Corrective Actions	Actions Taken	S	O	D	RPN
Sends command to motors	Communication with motors is disrupted	Motor failure or reversal												
	1) Overheating	motor driver or motor being heat damaged causing no movement or reversed movement	current is exceeding the regulatory capacity of the chip	prevention by adding cooling component	4	3	1	12	adjust potentiometer, implement fan, heat sink	heat sink	3	2	1	6
	2) Wire loosening	motor being unresponsive or reversed movement	One of the motor directional wires or current driving wires is not connected	prevention by better organizing wiring	4	1	1	4	consolidate the wiring using heat-shrink tubing and fasteners					

Limit switch for XYZ + B movement	Motors don't stop when at the limits of the machine	damage to the motor, the pipetting arm, and possible contamination of the sample in transit; components could fly off if everything is attached to the gantry plate												
	1) Mechanical issue with limit switch	damage to the motor, the pipetting arm, and possible contamination of the sample in transit	some of the older limit switches are not as responsive to touch (take slightly longer to click)	detect mechanism of failure mode and take corrective action	5	1	1	5	have spare limit switches on hand and inspect limit switches before turning device on or perhaps narrow down issue with current switch and find reliable limit switch					
	2) Wire loosening	damage to the motor, the pipetting arm, and possible contamination of the sample in transit	limit switches are all located at the periphery, so the wiring stretches from all over to converge at the shield	prevention by better organizing wiring	5	1	1	5	fasten down the wires to run tightly along the 8020 columns of the product					

The function of the microcontroller and motor shield is to centralize control over the various motors needed in the device. Failure modes related to motor control, especially those in the XYZ axes have particularly severe consequences. For example, the burning out of a motor driver can disable one of the motors, resulting in a misplacement of sample or contamination. Failure modes related to physical limits of the system, in other words the limit switches, even more severe because a motor trying to past the constraints of the device may cause injury to technicians operating the device.

We are most concerned about this heating issue as we have had to replace motor drivers multiple times already due to heat damage; however, at the same time some of the axes require that amount of power in order to operate under the weight. We are looking into a cooling system that includes both heatsinks and a fan. Wiring issues also came up as a mechanical way in which our controller system may break down, but this week reorganized all the wiring so we believe that this failure mode is already significantly reduced.

Rotating Deckplate:

Failure Mode and Effects Analysis									Sheet No. ___1_ of ______					
() System (X) Subsystem () Component	Description: Rotating deckplate				Comments: Device rotates patient plasma samples so that an individual sample can be scanned and collected by the pipetting arm at one location									
1	2	3	4	5	6	7	8	9	10	11	Action Results			
Function	Failure Mode	Effect(s) of Failure	Mechanism(s)/Cause(s) of Failure	Current Controls	S	O	D	RPN	Recommended Corrective Actions	Actions Taken	S	O	D	RPN
Moves patient plasma sample tubes to a location where sample can be scanned and extracted	Patient sample included in wrong pool (or in no pool)	Inefficient/incomplete sample pooling												
	1) Poor initial tracking	Patient sample not included in pool	Not all tubes are properly scanned	Prototype testing	5	1	5	25	Test spin to ensure that all tubes are scanned with a counter feature in Gcode or error message if not enough tubes are found ; multiple tests, comparison better detectibity					
	2) Drift	Patient sample included in wrong pool	Inertia error from high speed and weight of carousel	Speed testing for drift prevention	5	1	5	25	Redesign code so that a time stamp is included					
Carries test tubes with barcode facing out	Patient sample tube barcode unreadable	Patient sample not trackable												

	1) Test tube barcode not facing outward	Barcode scanner can't reach barcode	Technician doesn't place patient plasma sample tube in deckplate with barcode facing outward	Prevention by communicating to technician that the barcodes on the test tubes should be facing outward	2	1	1	2	Communicate clearly with technician perhaps in the form of a user guide					
	2) Test tube barcode not vertically straight	Barcode scanner can't scan barcode	Barcode isn't labeled consistently straight	Prevention by communicating to technician that barcodes should be labeled properly	3	1	1	3	Redesign Gcode or user interface to clearly communicate to user that all test tubes must be properly labeled					

Function of the deckplate is to hold all the samples and move each tube accurately and consistently into position as the pipetting arm moves. Most of the failure modes will be evaluated as we move forward with prototype testing. Fine tuning the step size will minimize the likelihood of skipping a tube or the barcode scanner not registering a tube. Other failure modes are focused on errors by the technicians operating our device such as placement of the tube or labelling the tube. There is a limited amount that we can do to prevent these failures, but it is possible to write code that can detect when an error has occurred and alert the users. Perhaps in future iterations, we will be able to provide a larger-scale redesign that can help avoid these issues in the first place such as multiple barcode scanners.

Pipetting Arm:

() System (X) Subsystem () Component	Description: Pipetting arm													
1	2	3	4	5	6	7	8	9	10	11	Action Results			

Function	Failure Mode	Effect(s) of Failure	Mechanism (s)/Cause(s) of Failure	Current Controls	S	O	D	RPN	Recommended Corrective Actions	Actions Taken	S	O	D	RPN
Extract/deposit blood samples into in/output tubes	Failure to extract/deposit sample when aligned	Blood sample skipped over. Unintentional mixing of samples. Not enough sample for subsequent PCR testing	Pipette arm/motor malfunction											
	1) the pipette motor does not rotate the proper amount to extract/deposit blood samples	Inproper amount of sample extracted/deposited	Propagation in pipette motor movement error. The pipette motor becomes stuck	Pipette motor testing in sample extraction	3	3	4	36	Verify if redesign with a detection system is necessary					
	2) The plugger of the pipette becomes "stuck". Independent of pipette motor movement		wear from overuse. spring malfunction	Pipette repeatability testing done independently from robot	5	2	4	40	Ensure that a brand new pipette is used in robot					
Avoid contamination between sampling	Mixing of input blood samples	Inaccuracy in subsequent PCR testing	Pipette arm/motor malfunction											
	1) Dripping of liquid onto exposed tubes during pipette arm movement	Input tubes will contain splashes of other blood samples	Not enough liquid pushed out by the pipette motor during sample deposition	Optimized size of sample extraction to avoid unnecessary movement of liquid	5	1	5	25	Incorporate shield that catches dripping. Make sure pipette arm does not move above untargeted sampe tubes					
	2) Failure in ejecting used pipette tips between sampling	Same tip will be used throughout sample pooling	The force at which tips are loaded onto the pipette is too great, requiring additional force from the pipette motor for tip ejection	Pipette tip loading/ejection testing	5	2	4	40	TBD					

The main function of the pipetting arm is to extract blood samples from the input test tubes and deposit samples in the output test tubes. One mode of failure is if the pipette is unable to extract or deposit blood samples once the arm is aligned with the test tube. The consequence of this failure includes skipping over individual blood samples throughout the sample pooling process, or samples mixed or insufficiently loaded into the output tube, making the subsequent PCR testing “inaccurate”. The pipette arm may malfunction if the pipette arm motor does not rotate the proper amount to extract or deposit the samples. This can happen through the propagation of error in pipette motor rotation. Another way the pipette arm may malfunction is if the plugger becomes “stuck” from overuse. Currently, our controls include pipette repeatability testing with and without the pipette motor to see if the proper amount of blood samples can be extracted and deposited repeatedly. One corrective action we can take is to make sure that the pipette used in our robot is swapped out every month to avoid the wear on the plugger.

Another major function of the pipette arm is to avoid contamination between sampling. The mode of failure would be the accidental mixing of input samples before sample pooling, which can arise from 1) the dripping of blood samples as the pipette arm travels above the test tubes and 2) the failure in ejecting used tips between sampling. The dripping can be solved by implementing a dripping shield as the pipette arm travels, or we can just design the pipette arm movement so that it never travels above other samples as it moves between target input and output tubes. The failure in tip ejection can be controlled by tip loading/ejection testing, where we find the optimal force applied by the z motion motor that allows the tips to be properly secured and ejected when needed.

7. Future Directions

7.1 Troubleshooting Issues at end of Quarter

At the end of the quarter, our testing and completion of our remaining pipette tip module was delayed due to our computers not being able to connect to our Arduino. As a result, we stopped working on everything else to try and identify the issue. We found pretty quickly that one of the power wires coming loose had caused a short on our original Arduino board.

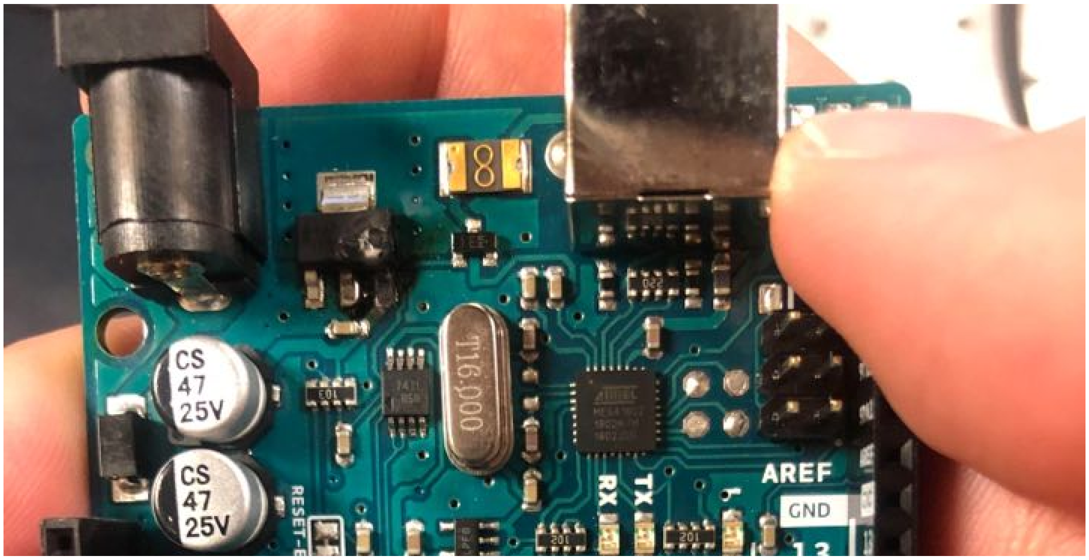


Figure 29: A short on our Arduino

After we had replaced our damaged components, we were able to communicate with the Arduino but there was a new issue where the Arduino board could not communicate the motor shield to execute G-Code. While consulting with our mechatronics coach, we performed the following troubleshooting.

- Reflashed GRBL onto the Arduino
- Checked all capacitors to make sure the polarities were correct
- Checked that the power supply was providing power to the motor shield
- Adjusted current limit on motor drivers
- Tried turning on one motor at the time in the case that the power supply was not providing enough current to drive all the motors
- Resoldered all connections to ensure that there are not shorts occurring on the board
- Replaced all motor drivers again in the case that they burned out

Due to exhausting the list of things that the issue could be. The following are the possibilities for what could be happening

- o There is a production defect either with the new motor shield or with the new Arduino board
- o We unknowingly through handling, transport, or soldering damaged part of the circuit board or shield

We believe that if we had bought the pre-assembled motor shield instead and had more time, we would have been able to deliver a complete proof-of-concept demonstration.

7.2 Summary of Next Steps

To build on the project, we have a couple suggestions on the next steps the next team can take. First, the new failure mode in which the controller cannot communicate with the motor shield needs to be addressed. We consulted with only Professor Marchuk but they are other experts on campus that can be consulted as well like Professors Matt Elwin, Kevin Lynch and Randy Freeman who could shed some light on where the disconnect is coming from. A preassembled Mega shield could also be ordered. There is a chance that the new shield we assembled ourselves could have bugs in it that could be addressed by ordering it preassembled. A different supplier for the motor shield could also be considered. We have been using Eccentric Workshop as our source but there could be other options that are more reliable. Another concept could be reorganizing the design to use two smaller Arduino development boards connected to two different motor shields capable of accessing 3 axes. The main setback here would be coordinating the G-Code commands but we imagine having one motor shield for X,Y and A axes then another for controlling the Z, B and C axes. A benefit here is that there are several more supplier options for 3 axes motor shields to try different setups.

Aside from this, there is a need to brainstorm a more robust cooling mechanism for the electronics. We had to regularly replace motor driver carriers due to overheating so devising a way to cool the device could save costs and provide more longevity. There could also be an investment in higher quality driver carriers. To improve aesthetics and promote organization, a neater cable routing system needs to be devised. Lastly, we were unable to add the pipette tip station so that work still needs to be done.