Palm Print: Handheld 3D Printer

A Major Qualifying Project

submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfilment of the requirements for the

Degree of Bachelor of Science

Submitted By:

Zackary Fischer, Mechanical Engineering

Tommy Lee, Mechanical Engineering

Liam Sullivan, Mechanical Engineering

Advised By:

Joe Stabile, Mechanical Engineering

Table of Contents:

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2019\3d_printer

1.0 Introduction **3**

2.0 Background **3**

2.1 3D Printing 3

2.2 Types of FDM Printers 4

2.3 Components of a 3D Printer 6

2.3.1 Extruders 6

2.4 Handheld/Do-It-Yourself Printers 15

3.0 Methodology and Iterations **17**

3.1 Iterations and Designs 17

3.2 Final Design 35

3.2.1 CAD 35

3.2.2 Assembly 37

3.2.3 Software 38

4.0 Analysis **47**

4.1 ANSYS 47

4.2 Base Analysis 47

4.2.1 Z Arm FEA 47

4.2.2 Z Arm FEA Older Models 52

4.3 Shim and Rods Analysis 53

4.3.2 Shim and Rods Older Versions 59

4.4 Movement Analysis 61

4.4.1 X Movement Archive 61

4.4.2 X Y Movement Archive 63

5.0 Conclusion and Recommendations **63**

6.0 Appendix **64**

7.0 References **66**

Abstract

3D printing is considered to be one of the next big things in the manufacturing world. Also known as additive manufacturing (AM), it allows for products to be designed with geometry which was not achievable by traditional manufacturing methods. AM is done layer by layer using many different materials including metal, plastics and nylon. AM is useful for many different sectors of the world, ranging from aerospace to medical fields. A portable 3D printer would allow for on-site creation of 3D printed parts. There are also learning opportunities. If students had their own personal 3D printers, they could rapidly prototype their designs anywhere. In this report, we look at some potential designs which allow FDM AM to be brought wherever it is needed.

1.0 Introduction

3D printing is a process that has been around for over 25 years. There are several different designs and styles for 3D printers, iterated by many companies across the world. Also known as additive manufacturing, it is used in schools, workplaces, homes, and many other places. New applications for 3D printers arise seemingly almost every day. These applications can range from the fields of medicine and dental, to automotive, to food, to fashion, and much more. Our group is focused on a new approach to 3D printing, the Palm Print. The goal of this project is to design and manufacture a 3D printer small enough to fit in the palm of one's hand, allowing for quick and easy access to prints from any location. For this project, we studied previous 3D printing models, analysing the best aspects of each, and using those concepts for our own design.

2.0 Background

2.1 3D Printing

3D printing, also known as additive manufacturing (AM) or rapid prototyping (RP), is a growing technology that is becoming essential in almost all areas of engineering. AM was originally brought to the public by Charles Hull, an Engineering Physics major from The University of Colorado, in the 1980s (Gross et al. 2014). Seeing the lengthy and costly process of prototyping, Hull set out to create an innovative new technology for industrial fabrication. As this technology developed, it became a way to shorten manufacturing time, prototype more frequently, and cut costs of testing products enormously. Before this technology, parts were created by removing useless material from a full block of stock. This technique is referred to as subtractive manufacturing. Additive manufacturing takes a different approach and creates a part by printing a part layer by layer out of nothing ultimately creating no waste when prototyping. 3D printing rapidly grew more popular within the following years and now you can find a 3D printer at almost every industrial business and even most college universities.

The process of 3D printing begins with a computer aided design (CAD). This is any 3D modeling program, such as AutoCAD or Solidworks, that allows you to digitally create a 3D object. The CAD design is then transferred to a stereolithography or .STL file. The STL file describes the surface geometry of the object, but does not include color or texture. This data allows communication between the 3D modeling program and the 3D printer. Hull introduced the .STL file in 1986 (Gross et al. 2014). The 3D printer then interprets the .STL file and creates a G-file via a slicer program. The G-file divides the 3D file into a sequence of 2D cross-sections and the printer will print those layer by layer to create the 3D product. The 3D printer then heats up and gets ready to print whatever material the user has chosen. Fused Deposition Modeling (FDM) is currently the most popular type of additive manufacturing that uses this process.

2.2 Types of FDM Printers

There are currently three main types of 3D printers that utilize FDM. There are Cartesian, Delta and Polar type printers. FDM's main principle is the extrusion of molten thermoplastics and all three of these printers use this. Although they all use the same principle, they differ in how they utilize it and that causes different print qualities and print times depending on the size and detail of the print.

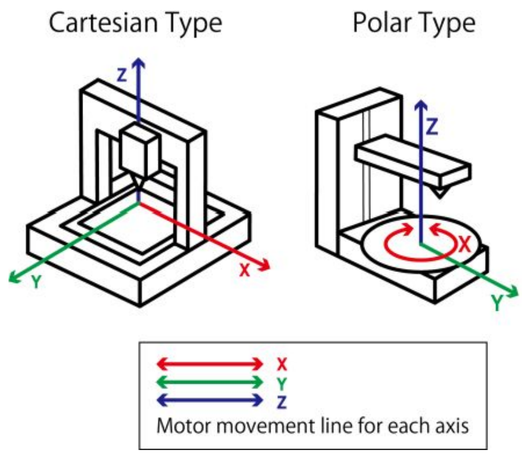


Figure 1: Types of FDM Printers

Cartesian style printers are the most common printers used in the market right now. They utilize the cartesian coordinate system to orient the print bed and the print head. Motors allow the print bed and print head to move in the X, Y, or Z direction. Many printers are created for the print bed to move in the Z direction and the print head to move in the X and Y direction. This can differ between printers but all move in a simple X, Y, and Z motion. The Delta printer has a circular base with the printhead suspended from the top. This printer also works on the cartesian coordinate system, but not as simply. The base of this printer never moves. The printhead is suspended by three metal arms in a triangular formation. All three of these arms move the printhead in the X, Y, and Z directions. The Polar style printers use the polar coordinate system. This system is similar to cartesian but the coordinate describes points on a circular grid rather than a square. As seen in Figure 1, all of the styles of FDM 3D printers have a different and unique look and style. Each of these printers has its own benefits and drawbacks. Cartesian style printers are slower while producing a better quality print. Delta style printers sacrifice the quality of the print for the speed. The Polar printers only use two motors which conserves energy usage.

2.3 Components of a 3D Printer

This section will go over all of the electrical and mechanical components that are necessary in a 3D printer.

2.3.1 Extruders

Extruders are the components of a 3D printer which control the feeding and processing of the plastic filament. The two main components of an extruder are the driver and the hot end. The driver, also referred to as the cold end, feeds the filament to the hot end through a gear system. The hot end then heats up the filament until it is hot enough to be printed. The cold end is hooked up to a stepper motor that drives the filament between two hobbed gears or a gear and a guide pulley.

There are two different types of extruders, direct drive extruders and bowden extruders. The cold end and hot end are all connected in one big assembly for the direct drive extruder. This arrangement minimizes the distance the filament has to travel and can allow for a more reliable 3D print with flexible filament. This does not mean this is the only way to print flexible material, but it is the most reliable since the travel from the pinch and the hot end is smaller. Direct drives on the other hand, are bulkier and heavier because it is one assembly. Also, the whole assembly must move on the gantry since it contains the hot end where the material is being printed from. For the bowden style extruder, the cold end is not connected to the hot end. Instead, the cold end feeds the filament through a smooth tube, usually Teflon, to the hot end. This can sometimes cause problems with flexible material getting stuck between the pinch and the hot end in the tube.

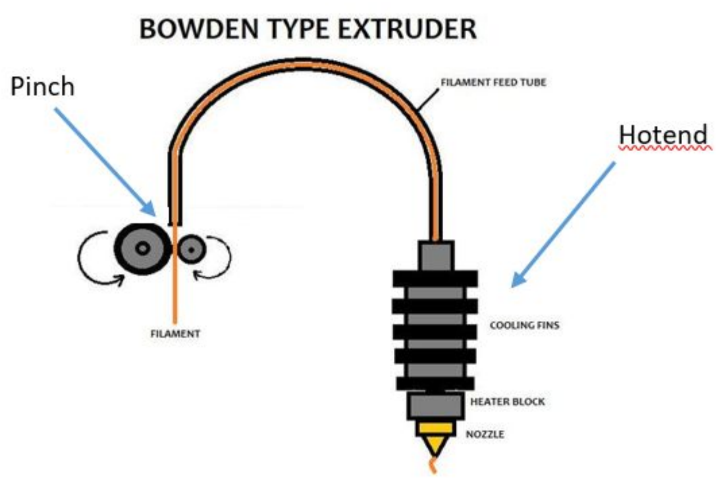


Figure 2: Pinch and hot end diagram

It also is a much lighter style. If you have a smaller printer and the gantry, or other mechanism holding the hot end, cannot withstand as much stress this less bulky style might be better. The hot end will be the only thing that needs to move during printing and you can have the cold end attached somewhere at the base or another stronger structure on the printer.

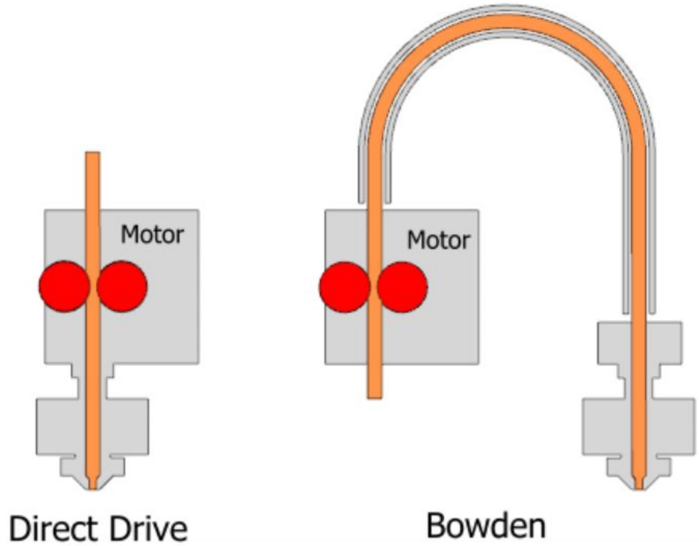


Figure 3: Types of Extruders

2.3.2 Stepper Motors

Stepper motors are DC motors that move in discrete steps.They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time (Earl, 2017). These motors are controlled by a computer, making them very precise. They’re best for applications of speed control, positioning, and low speed torque.

Stepper motors contain an iron shaft surrounded by an electromagnetic stator. Both the rotor and stator have poles. When the stator is powered the rotor moves to align itself with the stator or rotor moves to have a minimum gap with the stator. Thus to run a stepper motor the stators are powered in a sequence. Hence we provide pulses to the motor driver that will create this sequencing (Ferrah,Younes, 2018).

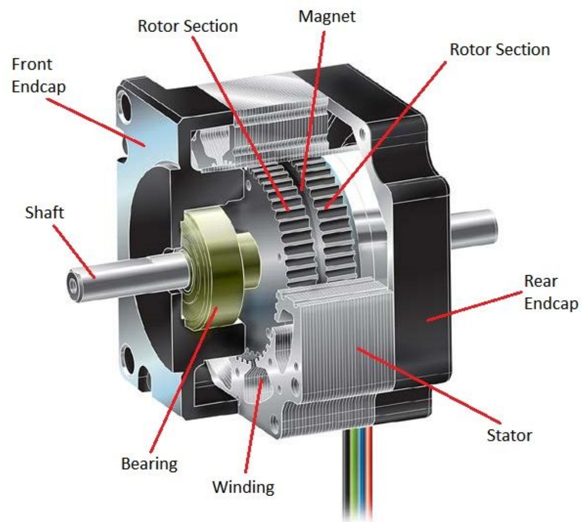


Figure 4: Stepper Motor Set-up

There are two types of stepper motors: bipolar and unipolar. Unipolar motors have two windings that each have a center tap. The taps come out of the wire either as separate wires or come together and form one wire. This is why unipolar motors have either five or six wires. The center tap wires(s) run through a power supply and the ends of the coils are alternatively grounded (Jones, 2004). Bipolar stepper motors also have two windings but only four wires. This is because unlike the unipolar motors, these do not have a center tap. The advantage to not having center taps is that current runs through an entire winding at a time instead of just half of the winding. As a result, bipolar motors produce more torque than unipolar motors of the same size (Jones, 2004). Bipolar motors are the motors that are most commonly used for 3D printers, although you can hack unipolar motors to be bipolar and they work the same.

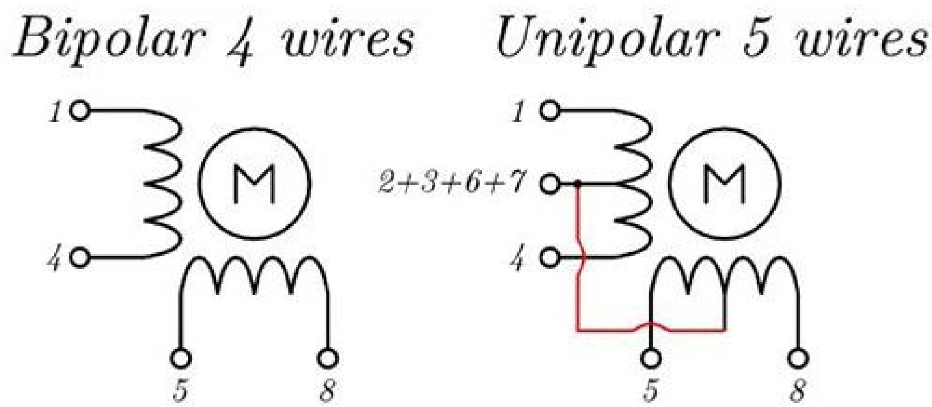


Figure 5: Unipolar Vs. Bipolar Wiring

You can find stepper motors with different gear ratios, torque, shaft size, diameter, and wiring. Choosing the correct stepper motors for your 3D printer depends on the torque, precision, speed, and size you need.

The motor we used for our X and Y axis is the 28BYJ-48 stepper motor. This is a 5 wire unipolar motor that runs on 5V. It has 32 steps per revolution and a gear ratio of 64:1. With as 1/64 reduction gear set, that actually means the motor has 2038 steps per revolution. A five wire unipolar motor is not compatible with most of the Motherboards and Drivers that are used for 3D printers. In our case it is not compatible with the rest of the hardware we use. Because of this we had to hack the motor, which will be discussed more in depth in the methodology section. We chose to use this motor because it is much cheaper than the other unipolar motors that we were choosing between.

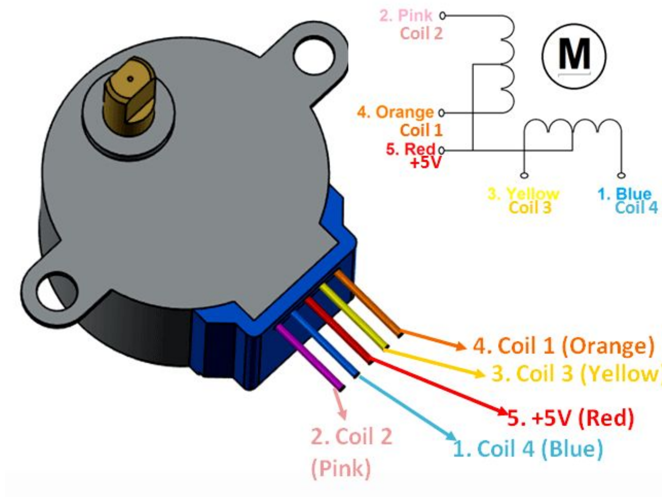


Figure 6: 28BYJ-48 Motor

2.3.3 Endstops

All axes of a 3D printer need a datum to reference their movements. This is also known as a home position or an endstop. These endstops also help the machine protect itself by not moving past its intended range of motion. Endstops can either be mechanical, magnetic, or optical depending on the application. 3D printers most commonly use mechanical endstops, such as a switch, but some also use optical endstops.

Mechanical endstops are most common because they are much less complicated and cheaper than an optical option. They are simply positioned on each axis where the printer reaches its motion limit and triggers the machine when reached. Little force is needed to trigger the mechanical switches, ensuring that the machine will indicate it is at its limit of motion. However, these switches will have a cycle life and may have to be maintained over time depending on how many times you use your printer.

Figure 7: Mechanical Endstop

2.3.4 Control Boards

The control board of a 3D printer is one of the most essential components. They handle the G-code, temperature regulation, and moving the motors. It is the brains of the machine and without this nothing else would work. The control board has two different components, the microcontroller and the circuit board. Many times these are combined into one board, other times they come separately and are attached to one another. There are many different kinds of control boards, but the most popular one that many printers use is the Ramps 1.4 board.

The Ramps 1.4 board is used with the Arduino Mega 2560 board. The Arduino is a microcontroller and the Ramps can be attached to it as a shield set up. The board distributes power to all parts of the 3D printer. It will control the motors, extruder, thermistors, the end stops, and the heated bed if your 3D printer has one. Attached to the Ramps 1.4 board you need motor driver boards that will drive the motors. You will need one per motor and place them in their proper position ensuring the pins are in the correct spots. We used A4899 driver boards.

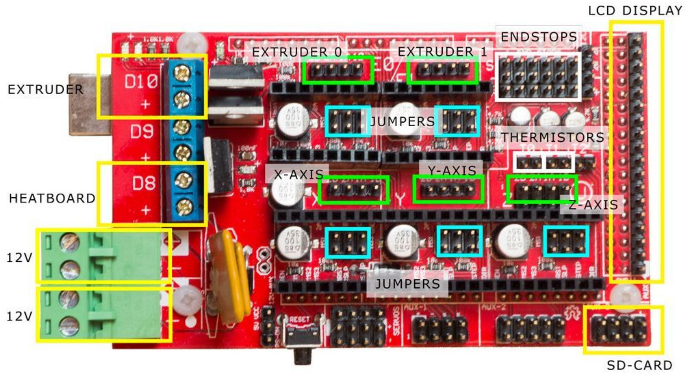


Figure 8: Ramps 1.4 Set-up

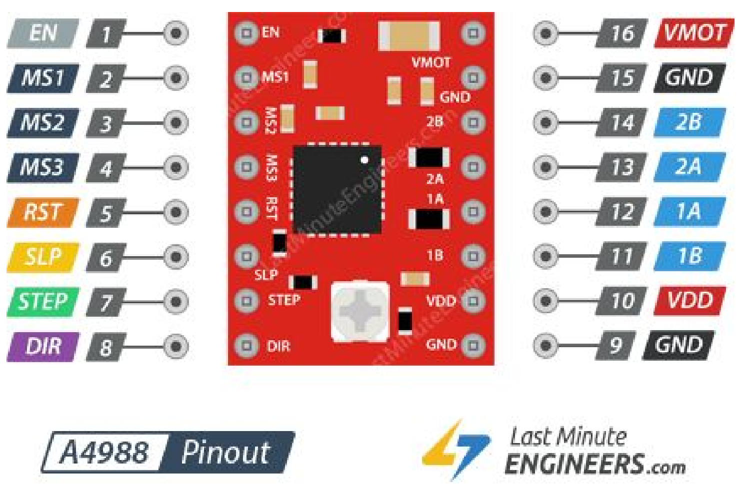


Figure 9: A4899 Driver Board

2.3.5 Firmware

The firmware of a 3D printer is the bridge between the hardware and the software of the machine. As the software sends G-code commands to the printer, the firmware interprets the commands into a recognizable language for the machine. The G-code commands, created by a slicer program based on the CAD design, are uploaded to the control boards from the printer through the firmware. The firmware needs to be configured specifically for each individual 3D printer. The configuration of the firmware is dependent on many variables such as what motors you have, how many extruder, how big your printer is, if you are using a belt and pulley system or a lead screw, etc. There are many different firmwares available for use of a 3D printer. However, since we had chosen to use a RAMPS 1.4 there are only a few that are compatible. Table 2.1 will show you the compatible Firmwares.

Firmware:	**Notable Features:**
Klipper	● High precision stepper movement ● High step rates ● Multiple microcontrollers supported on single printer ● Supports Cartesian, Delta, and CoreXY
Sprinter	● SD card reader ● Stepper extruder ● Extruder speed control ● Movement speed control ● Constant or exponential acceleration ● Heated build platforms
Teacup	● Moves steppers smoothly ● Start-stop ramping ● SD card reader ● Unlimited number of heaters
Marlin	● Multi-platform ● Configurable for a wide variety of

	machines ● Interpret based temperature and movement ● High stepping rate ● SD card support ● LCDs ● Endstop support ● Autotemp

Table 2.1 Compatible Firmwares (RepRap. List of Firmware. 14 March 2017. 23 April 2017)

The most popular firmware for 3D printers is Marlin because of its extensive support, reliability, and precision. Based on our research we decided to go with Marlin firmware. Since our 3D printer is to be small and handheld, we wanted a precise firmware that could handle small movements. We also found it had the most support articles on how to use it.

2.4 Handheld/Do-It-Yourself Printers

There are many “Do-it-yourself” mini 3D printers out there online. Many of them had components that inspired our final design. All of the printers we came across in our research had their own unique qualities to them. Many used old parts from DVD players or CD players for their X and Y motion. One that our group came across early on in our research was called the “EWaste 3D printer.” This printer uses all old parts from CD/DVD drives. This made us really strive for a low cost printer because people are able to create these from recycled parts.

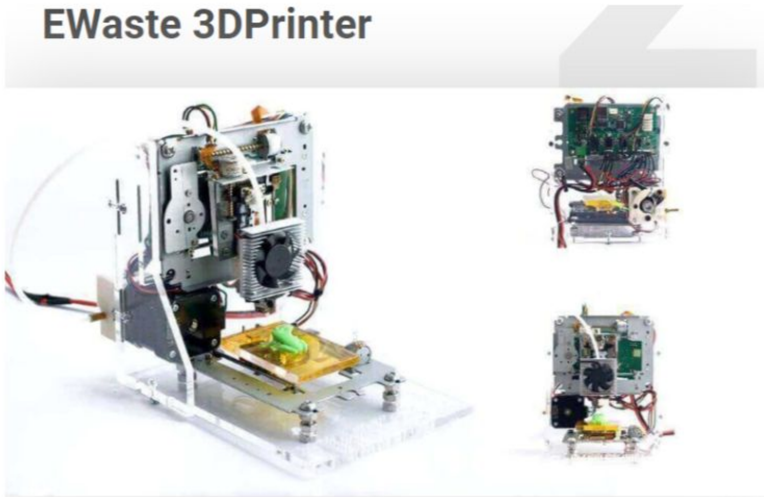


Figure 10: EWaste 3D Printer

The most influential DIY 3D printer was one that a man made on his Youtube Channel. This one also used CD/DVD drives and other household items.Many 3D printers have the Hotend moving in the Z and the X direction and the bed only moving in the Y. This DIY printer has the base moving in both the X and Y and the Hotend only moving up and down in the Z direction. The hot end is attached to a linear actuator on an arm that arches over the center of the printer. This was a big influence on how we designed or final product.

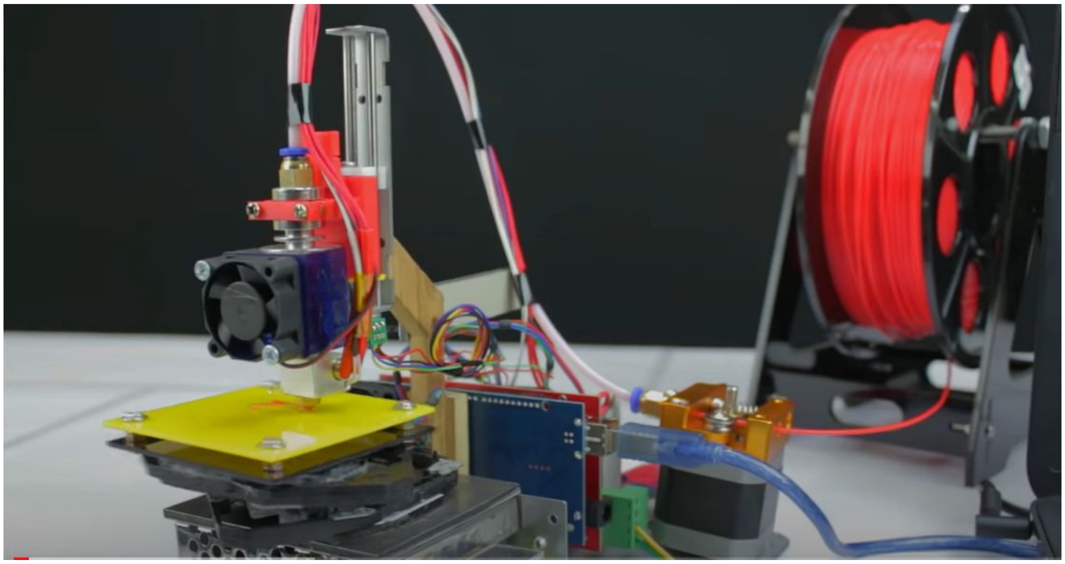


Figure 11: DIY Mini 3D Printer from Youtube

3.0 Methodology and Iterations

3.1 Iterations and Designs

Throughout our work on this project, our team made many changes to our design through our various iterations. Our first design was focused on getting X and Y movement. Our team used a previous project derived from a core X-Y. We scaled the assembly down and 3D printed our parts. Once our parts were printed, we realized which aspects would not perform the way we wanted them to.

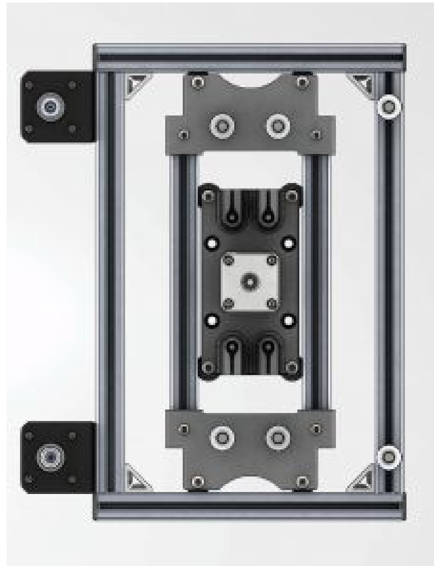


Figure 12: X-Y Core

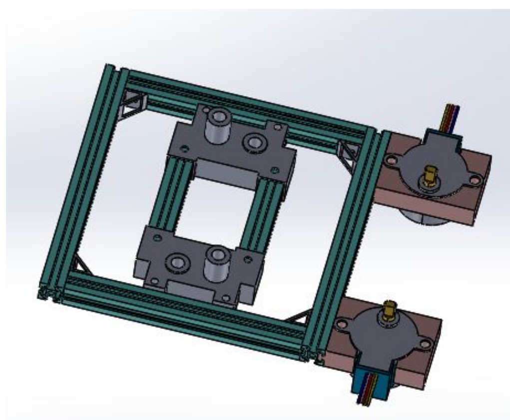


Figure 13: Solidworks Model

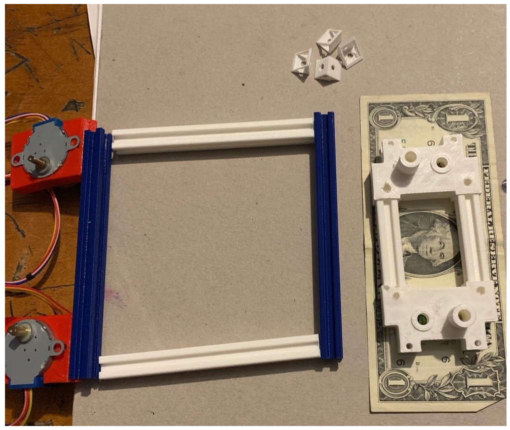


Figure 14: First Design next to $1 for size reference

Some of the main takeaways we were able to get from our first venture into hardware space was that PLA parts do not slide well on other 3D PLA parts. Our team went back to the drawing board and looked at other 3D printers for ideas. Our team continued to have the print bed move in both the X and Y direction. Our new design utilized two PLA 3D printed sides with steel rods connecting them. A middle block moved along those two rods which carried another stepper motor and moved the print bed in the other direction. This iteration also introduced sliders with a preload on the rods. This allowed the sliders to continue to move when the rods were not perfectly parallel.

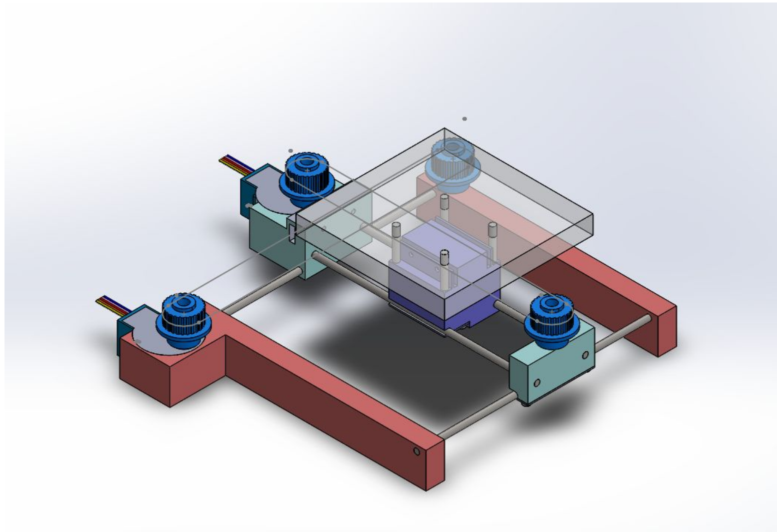


Figure 15: Next iteration utilizing preloaded sliders

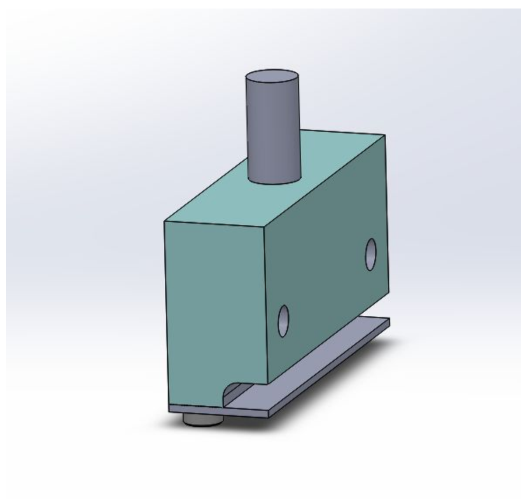


Figure 16: Slider with preloaded shim

There were a few issues with this iteration. The one one being the overall stability of the product. The friction from the sliders caused the PLA sides to move. The entire assembly was unstable and did not allow smooth movement to occur. We used the same parts in our next iteration, but included a base which was one part in hopes to stabilize the assembly.

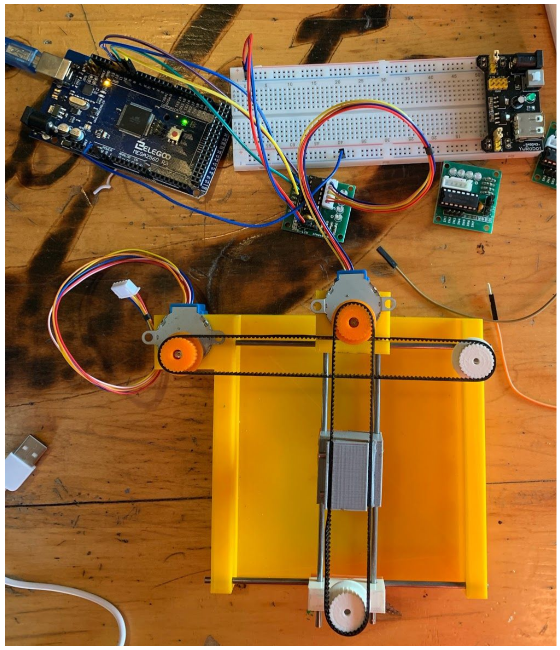


Figure 17: Design with a stable base

The stability allowed us to get movement in both directions however the Y movement still had issues. This was due to the fact that there were 3 separate parts which were being moved by a motor at the end. We consolidated those three parts into one part and moved the motor to apply to force on a centered location on the side of the base.

We attempted to improve our Y movement by pinching the belt in between the rods as well as making the Y slider one piece instead of multiple parts, which were used in our previous designs. This yielded good results.

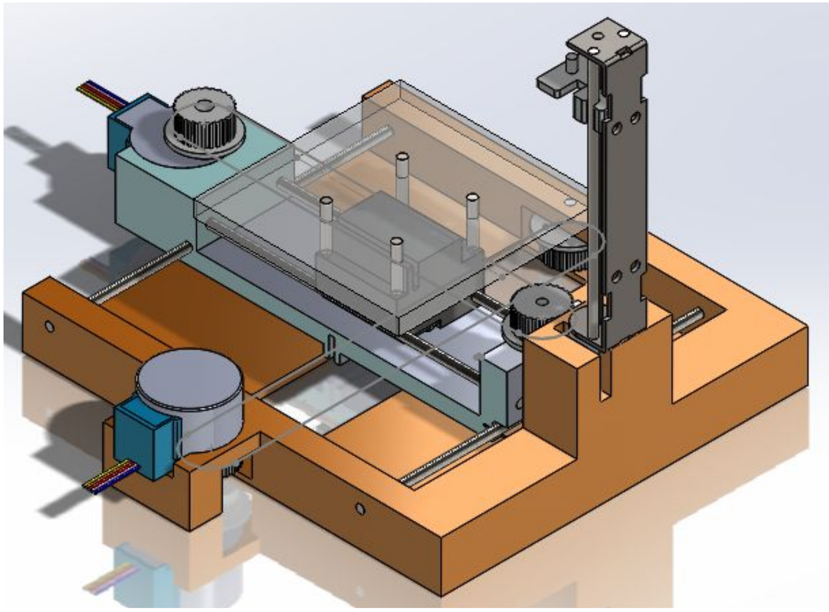


Figure 18: Design with Y slider as one piece

Our next iteration focused on locking the rods in the base to stop movement. When the sliders were moving, the rods would get pulled out of place. To stop this from happening, we substituted the circular holes in the base with V grooves. The rods were placed in the V groove and screws with oversized washers were fastened next to the rods. The oversized washer applied a downward force to the rod which locked it into the V groove and stopped any unwanted movement.

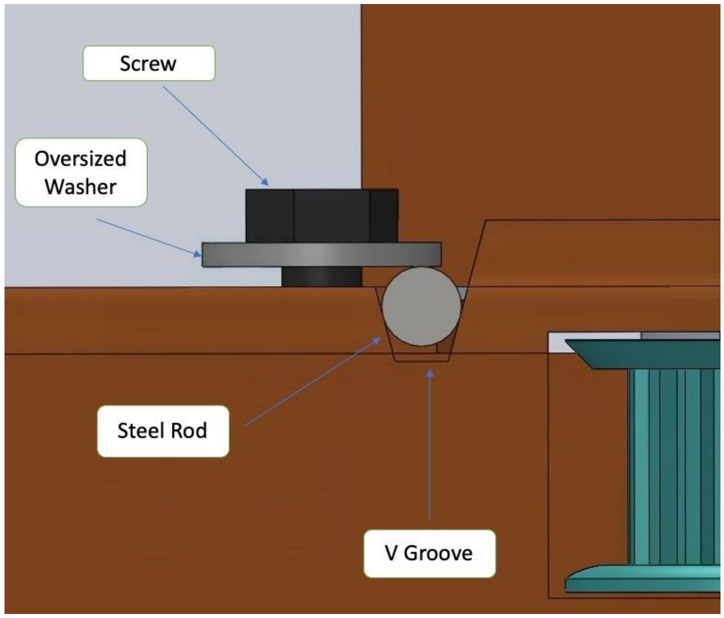


Figure 19: Oversized Washer with V groove

The rods which the Y slider used were also moved closer together to reduce the moment created from the forces of friction. This design also implemented sliders with 3D printed PLAcantilever shims. This eliminated the need to add shim stock to the part, decreasing our manufacturing time. We also began to look into shrinking our design with this iteration. We changed our timing belts and pulleys from 1/4th inch to 1/8th of an inch. This allowed us to use smaller pulleys and shrink the overall size. Unnecessary material was also removed from the base and sliders.

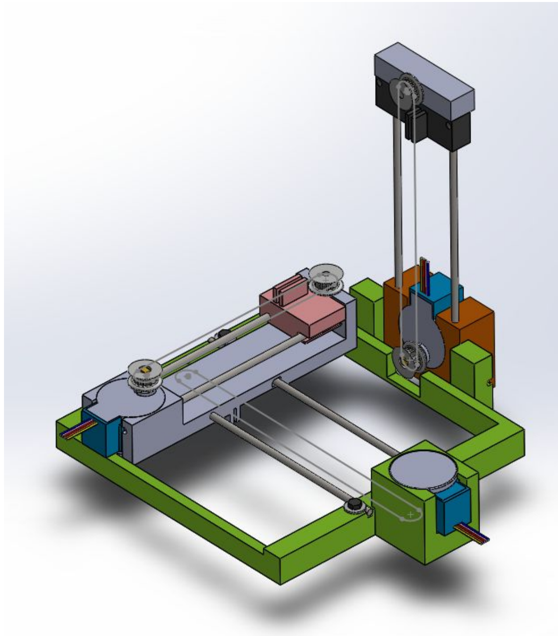


Figure 20: Design with V grooves and closer rods

The next changes we made focused on our sliders. Both of the sliders were not as stable as we hoped. Small forces on either of the sliders would make them wobble, which would lead to a decrease in quality of our prints.To alleviate this issue, we swapped the circular holes which guide the rods out and implemented a V groove paired with a cantilever shim.

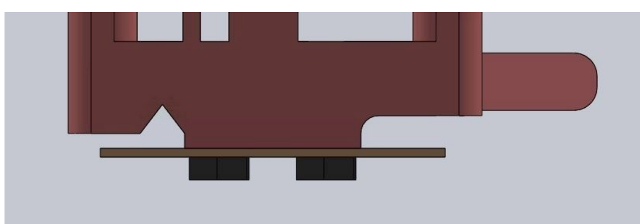


Figure 21: Grooves on Sliders

We used ANSYS to find the preload the shin stock would exert on the rods to find the force of the shim stock on the rod. We experimented with different preloads on the cantilever shims to find the ideal force to eliminate unwanted movement but also allow for easy movement along the rods.

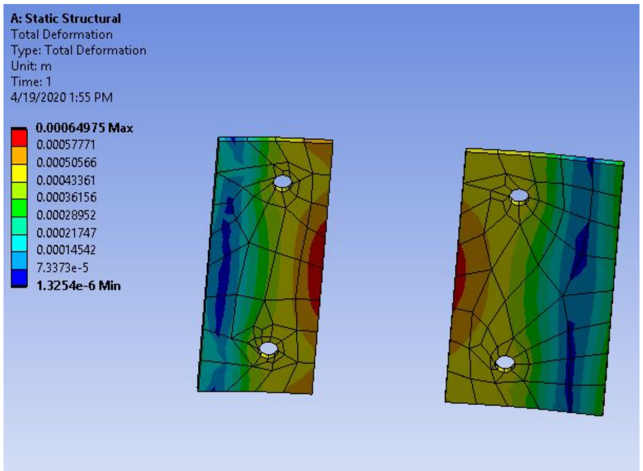


Figure 22: Analysis on Shims

The team also ran into some issues with pinching the 1/8th inch timing belt. Our previous method used with the 1/4th inch timing belt was driving a pin through the belt which secured it to the slider.

The 1/8th inch timing belt did not work well with this method and ripped when moved with the motor. To fix this, our team altered our belt pinch and used smaller pins.

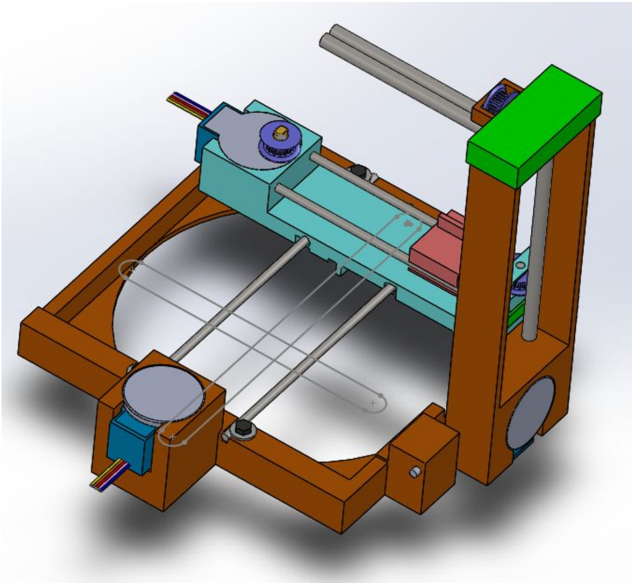


Figure 23: Design with V Grooves implemented on sliders

The new belt pinching method worked well, but we had issues tensioning the 1/8th inch timing belts. This caused issues with the accuracy of our movements as the pulleys would sometimes “miss a step”. To alleviate the lack of tension on the belts, we ordered torsion springs which are attached to the belt to increase tension.

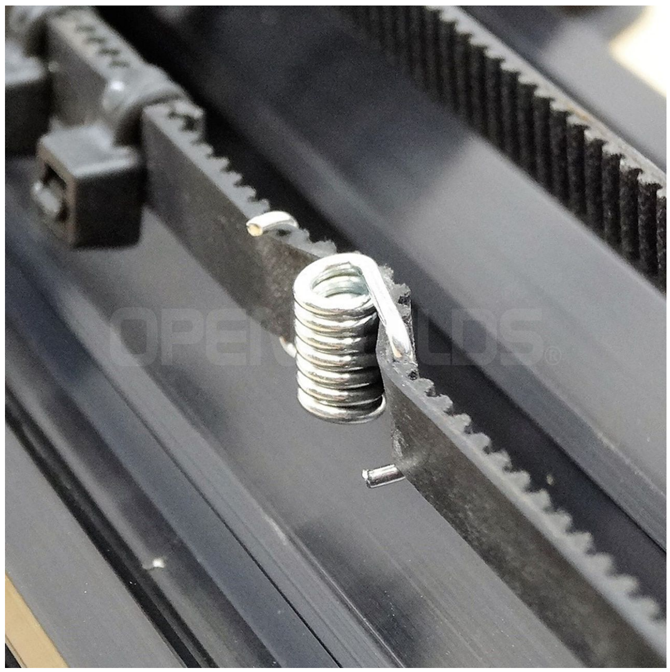


Figure 24: Example of a Torsion Spring used to Tension a Timing Belt

The torsion springs increased the tension of the belts, but it was not enough. We continued experiencing problems with the 1/8th inch timing belts and went back to a design with 1/4th inch timing belts. Once we reverted back, our X and Y movement was smooth and accurate.

While the X and Y movement was being improved, other members of the team worked on our Z movement. The Z direction moves the hot end of the extruder up and down from the print bed. Our first iteration of the Z direction was a simple motor coupled to a threaded rod. It had a guide rod next to it which kept it on track. This idea was inspired by other 3D printers such as the Prusa.

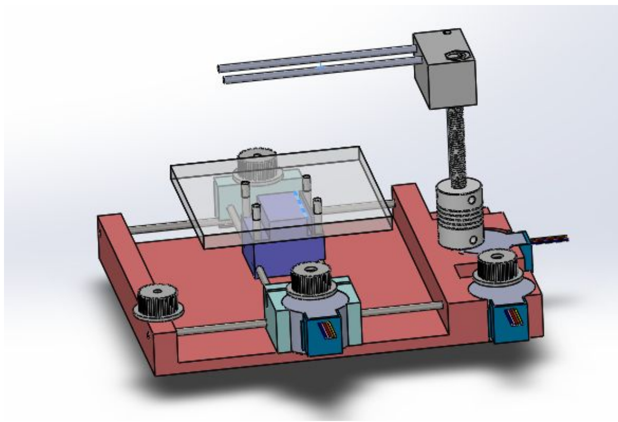


Figure 25: First Iteration of Z direction

While searching for mini stepper motors online, we came across a linear actuator with a very small stepper motor attached at the bottom. This was our inspiration for the second idea of the Z direction. The second design had two of the linear actuators on either side of the base of the assembly. If this design was continued it would have a cross arm attached to both actuators with the hot end in the middle.

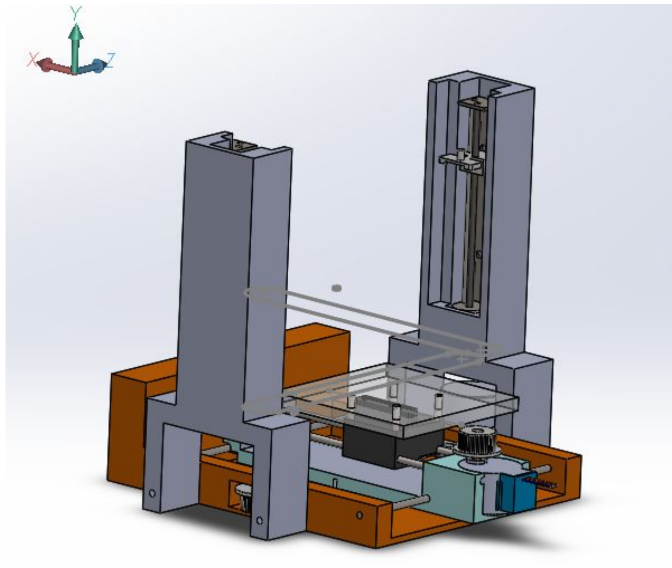


Figure 26: Second Iteration of Z direction

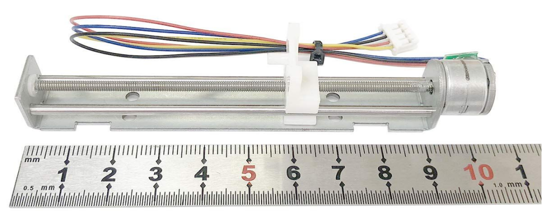


Figure 27: Linear actuator from Amazon

We then decided we wanted to try to use the pulley and belt method we used for the X and Y directions since it was working for us. We created a design with the same fundamentals as our other axis which is shown below. This design also allowed for the Z axis to collapse, making it easier to transport.

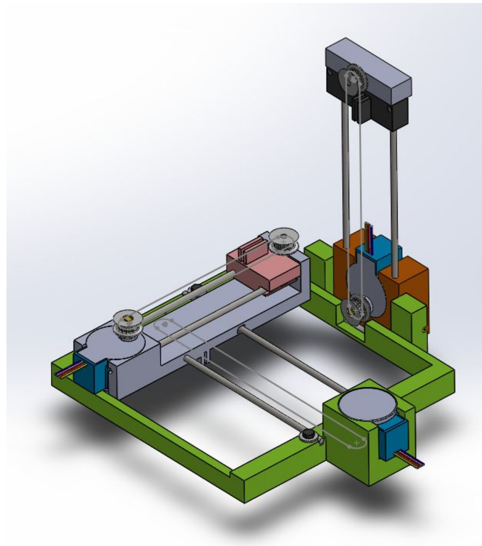


Figure 28: Z Direction with pulley method

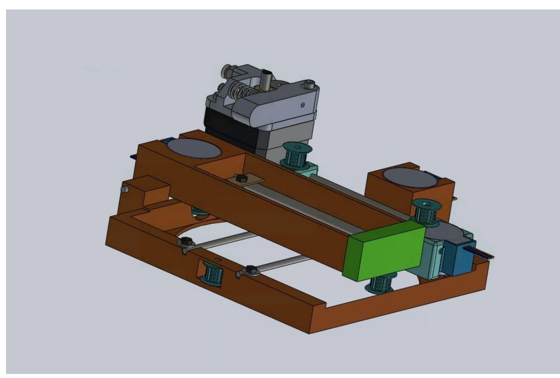


Figure 29: Collapsibility of Z direction

Our team liked this design and the concept of the collapsibility. We also wanted to take more concepts from the XY direction and apply them to the Z direction. We decided to make the Top and bottom bases of the Z direction one piece in order to make it more stable. We also added a cap on the end of the Z direction to stop the rods from moving. This iteration also utilizes 5mm steel rods for the Z direction instead of 3mm steel rods which are used in the X and Y direction to reduce wanted bending.

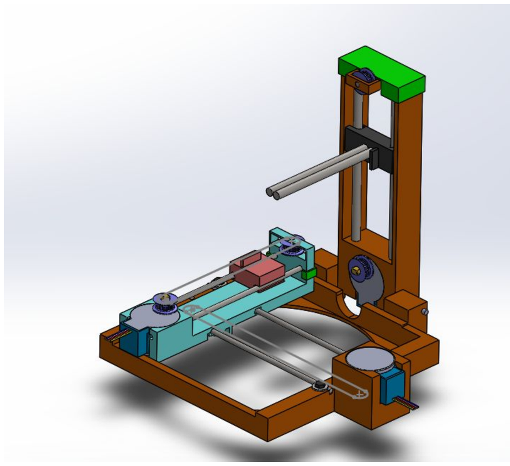


Figure 30: Z direction as one piece

Afters some testing, our team decided that the timing belt option was not as promising as our threaded rod option for the Z axis. Our team made our first model with the threaded rod assembly purchased from vender . The threaded rod assembly was held on with a 3D printed arm which was screwed into the base.

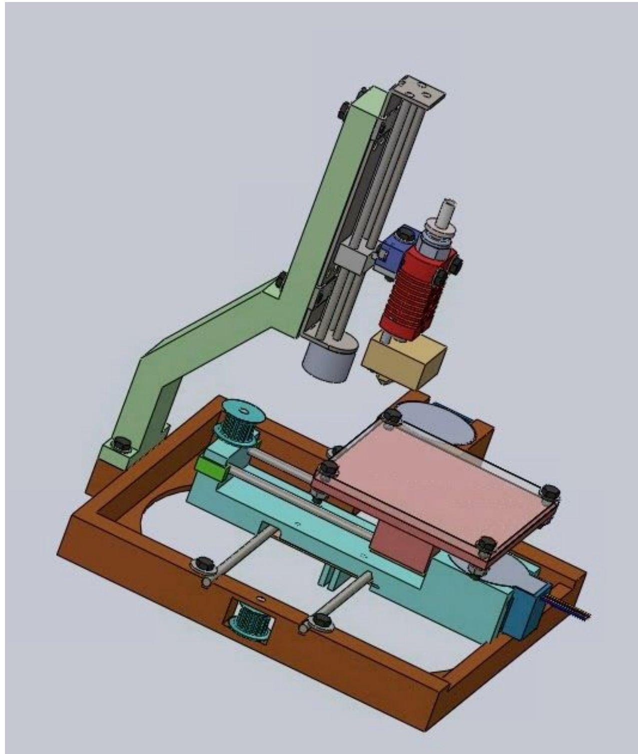


Figure 31: CAD model of Threaded Rod Option

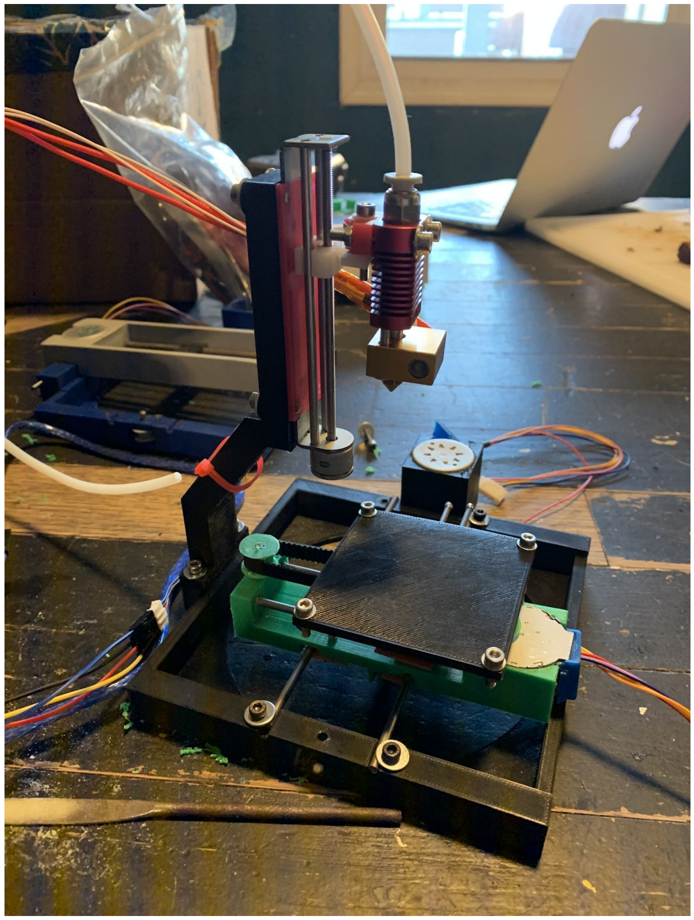


Figure 32: Threaded Rod Option

We were able to achieve good movement in all three axises with this model. Moving forward, we wanted to increase the portability of the printer and improve the stability of the hot end. To improve the portability, we created a removable Z arm which utilized a snap fit. This allowed for easy disassembly when transporting the printer. We performed various analyses to ensure the fit would support the weight of the hot end, but also be easy to remove and reattach.

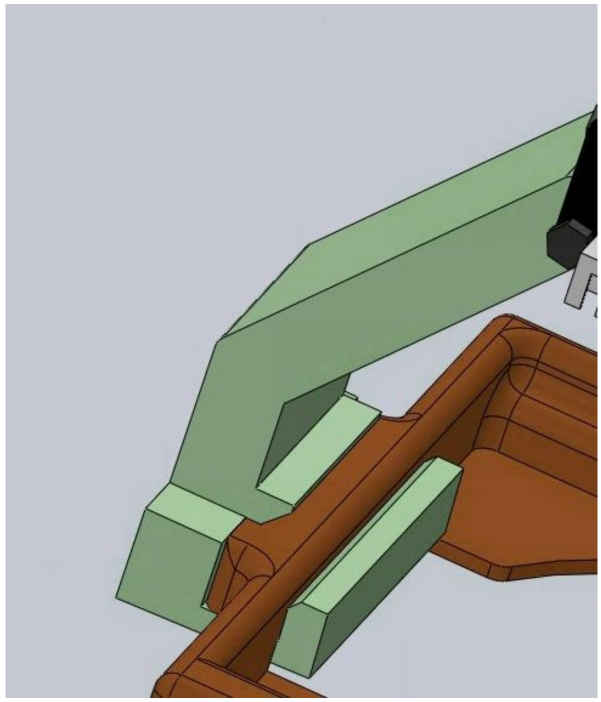


Figure 33: Snap fit Z arm

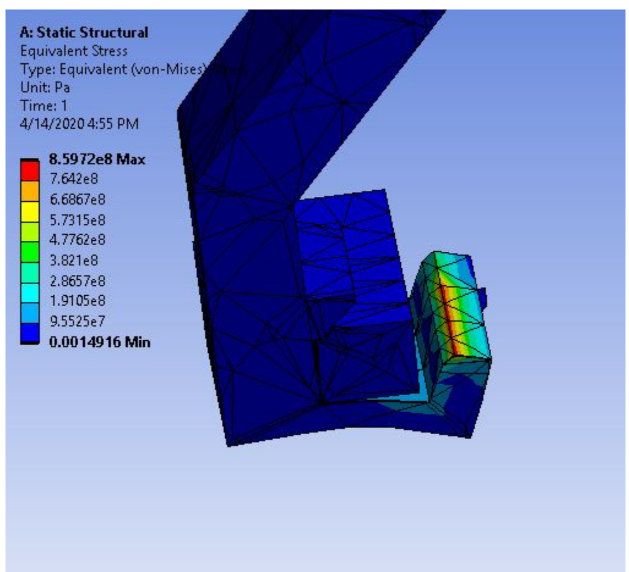


Figure 34: Ansys analysis on Z arm

Our hot end was still not as stable as we would like it to be. We created new fixtures which attached the hot end to the threaded rod assembly and reduced unwanted movement. The sub assembly consisted of three main parts. Two of them connected the hot end to the threaded rod assembly. The third part glided along the guide rods of the threaded rod assembly, stabilizing the hot end.

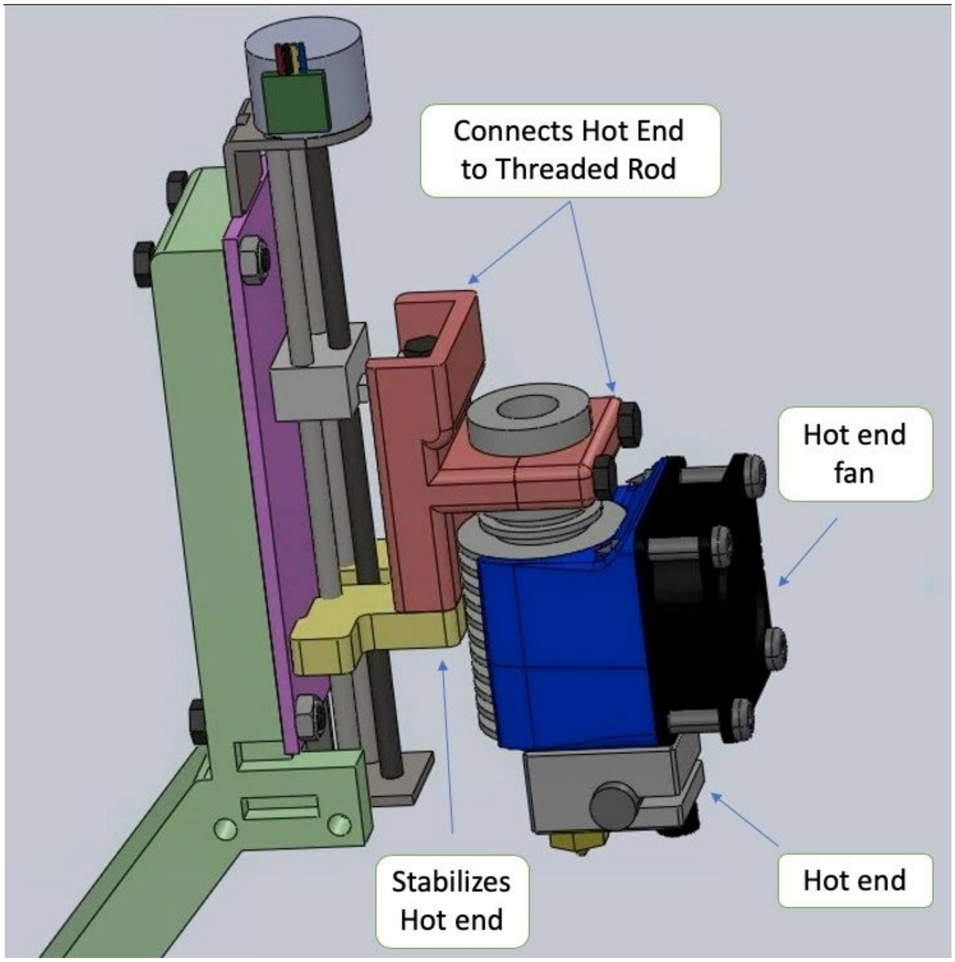


Figure 35: Hot end fixture

3.2 Final Design

3.2.1 CAD

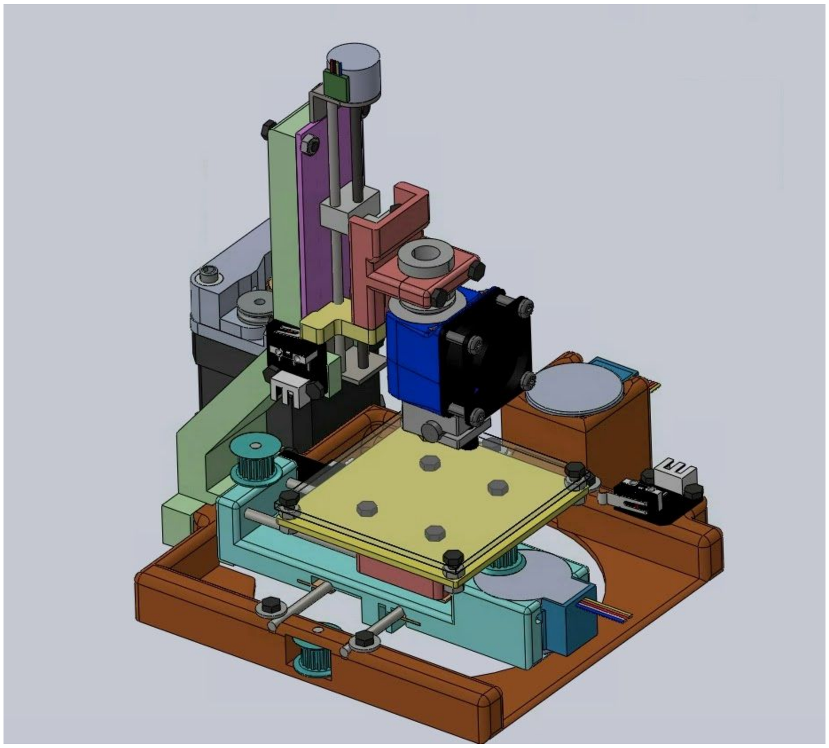


Figure 36: Final CAD Model

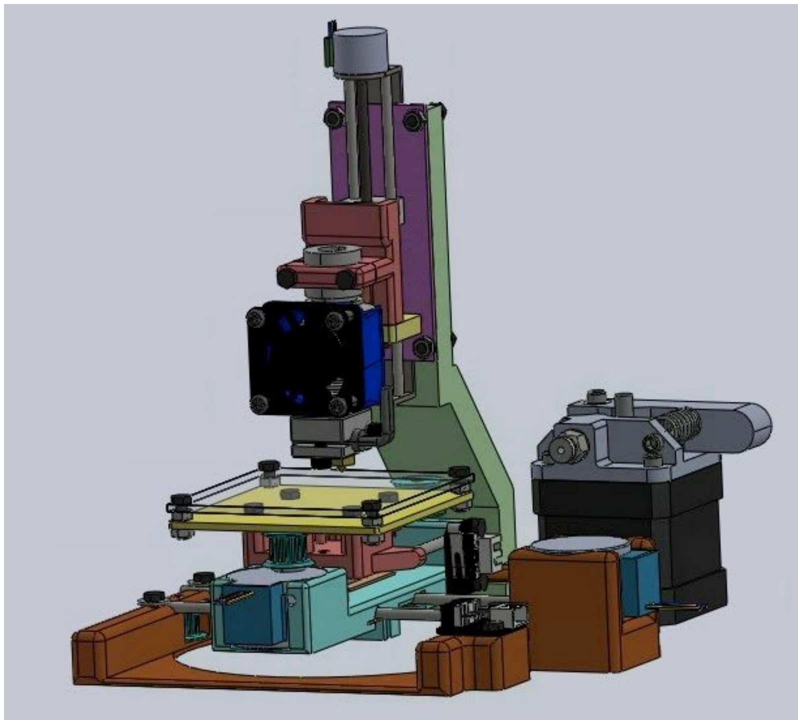


Figure 37: Final CAD Model

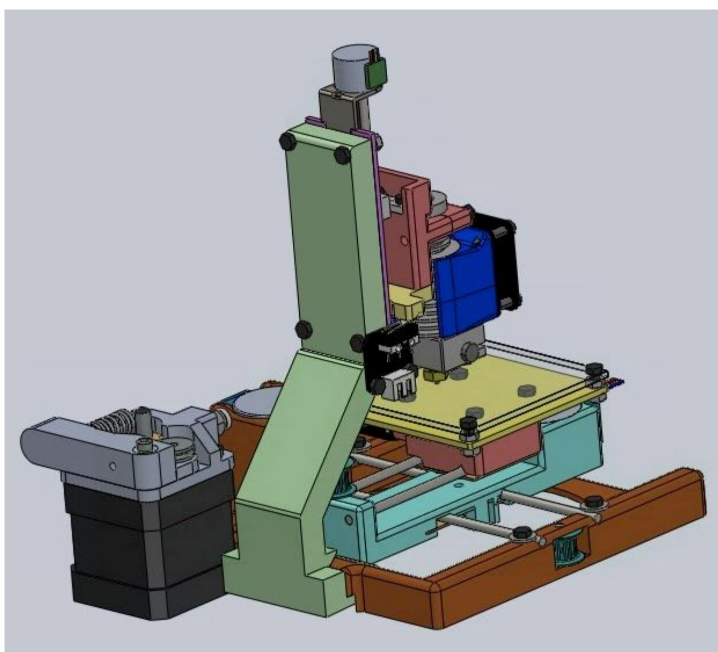


Figure 38: Final CAD Model

3.2.2 Assembly

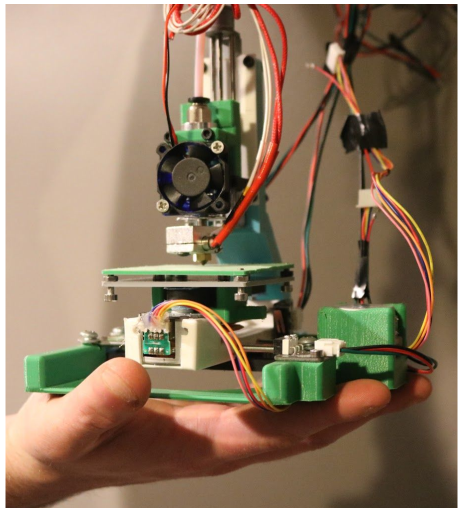


Figure 39: Final Assembly

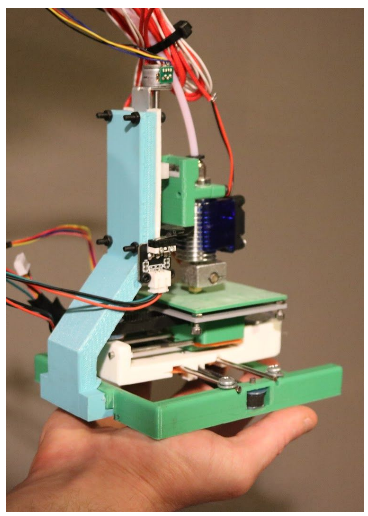


Figure 40: Final Assembly

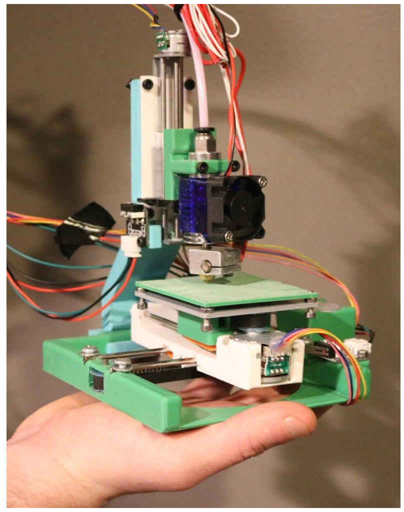


Figure 41: Final Assembly

3.2.3 Software

There are many different ways you can code a 3D printer. Our printer was run by an Arduino 2560 therefore we used the Arduino platform with Marlin Firmware to code. We also used Pronterface to control the printer. Pronterface translates your Marlin Code and then runs the G code you feed it to control the printer. You can also manually control the X, Y, Z movements, temperature, and extruder. Every 3D printer needs to be coded uniquely in Marlin based on many factors of the printer. Marlin Firmware has a generic configuration code and you need to go through it and change the lines and commands that are unique to your printer. Below are the most important lines that we changed in the configuration for our printer.

Steps

1. **Line 113: Baudrate**

In this line we set our Baudrate which is the communication speed of our printer. We googled our main board's data sheet in order to determine what this number should be. Our Baudrate was 115200.

```
 *
 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 */
#define BAUDRATE 115200
```

2. **Line 121: Defining MOTHERBOARD**

This is where you tell the firmware what main board you are using. You need to change the "code" that aligns with the board you are using for your 3D printer. To find this find the little dropdown arrow at the far right of the tabs and click on boards.h For example, we use a Ramps 1.4 board and our code is "RAMPS_14_EFB."

```
// Choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_14_EFB
#endif
```

3. **Line 136: Defining Extruders**

Here we identified the number of extruders/hot ends our printer has. Our printer only uses 1 extruder/hot end.

```
// This defines the number of extruders
// :[1, 2, 3, 4, 5, 6, 7, 8]
#define EXTRUDERS 1
```

4. **Line 270: Defining Temperature Sensor**

In this line you need to tell the firmware what type of thermistor your hot end uses and how many it has. Depending on the thermistor you are using they have "codes" you will need to enter.

```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_5 0
#define TEMP_SENSOR_6 0
#define TEMP_SENSOR_7 0
#define TEMP_SENSOR_BED 0
#define TEMP_SENSOR_PROBE 0
#define TEMP_SENSOR_CHAMBER 0
```

You can find the codes for each thermistor type in the lines directly above the line you enter it in. Our thermistor was a 100K and the code that lined up with this was "1."

```
 *     0 : not used
 *     1 : 100k thermistor - best choice for EPCOS 100k (4.7k pullup)
 *   331 : (3.3V scaled thermistor 1 table for MEGA)
 *   332 : (3.3V scaled thermistor 1 table for DUE)
 *     2 : 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)
 *     3 : Mendel-parts thermistor (4.7k pullup)
 *     4 : 10k thermistor !! do not use it for a hotend. It gives bad resolution at high temp. !!
 *     5 : 100K thermistor - ATC Semitec 104GT-2/104NT-4-R025H42G (Used in ParCan & J-Head) (4.7k pullup)
```

5. **Endstop Inverting**

This section addresses the end stops, also known as the limit switches. You should only change this section if there are homing issues with your printer.

```
// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
#define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.
//#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
//#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
//#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
```

Toggle the true and false entry if one of your axes is going in the opposite direction when you are trying to home it. Most printers only have min endstops, like ours, therefore, you should only toggle the min endstop lines for the axises you are having problems with. We ended up having to invert all of our axises to correct our homing.

6. **Defining Axis Steps per Unit**

The steps per unit value is the amount of steps the motors will make in order to move that certain axis one unit. In this case one unit is equal to one millimeter. To figure out your steps per unit you can use an online calculator or calculate it yourself. To calculate this in the X and Y direction which utilizes a belt and pulley system, you will need to know the steps per revolution for your motor, the circumference of your pulleys, and the microstepping you are using. The equation you will then use is: (Steps per revolution / microsteps / circumference of pulley). Our initial steps per units for the X and Y axises was 655.36. Our motors had 2048 steps per revolution with 1/16 microstepping and 50 mm circumference for our pulleys. Once we got this value we had to run the axis at this number a certain distance. We tried to run the axises 10 mm and measured that it only traveled 7mm. We then recalculate with this equation: (10mm * 655.36)/7mm. This got us our final steps per unit of 936.22 for the X and Y axises. For the Z axis we used a lead screw and a different motor. Finding the steps per unit for a lead screw has a different equation. The equation is (Steps per revolution / pitch).

```
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *                                      X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT   { 936.22, 936.22, 615.38, 500 }
```

7. Inverting Directions

In this section you can control the direction the stepper motors turn. After testing the motors change the true or false input for any axis that is moving in the wrong direction.

```
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false
#define INVERT_Y_DIR false
#define INVERT_Z_DIR false
```

8. Travel Limits

These lines tell the printer where each axis is after homing. It also tells the machine how far each of the axises can travel from the homing position. To measure these dimensions perform the home command. When the carriage is fully homed measure how far the hot end nozzle is from the print bed and how far the X and Y axis can travel before crashing into the opposite side. Below is an example diagram of the measurements you should be taking and the inputs into the firmware based on the measurements.

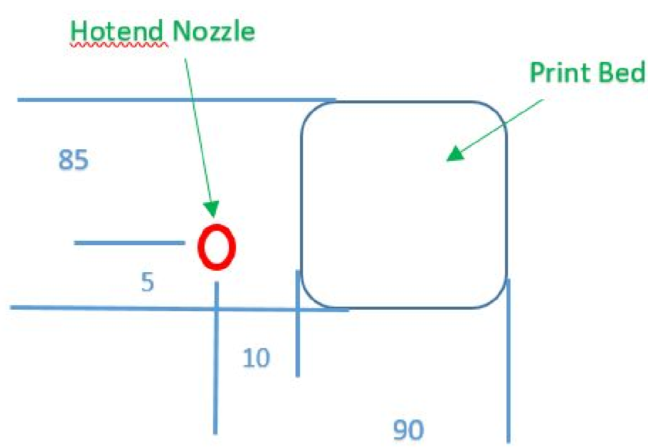


Figure 42: Position of hotend after homing to determine travel limits

```
// Travel limits (mm) after homing, corresponding to endstop positions.
#define X_MIN_POS -10
#define Y_MIN_POS 5
#define Z_MIN_POS 0
#define X_MAX_POS 90
#define Y_MAX_POS 80
#define Z_MAX_POS 75
```

Here is our travel limits we calculated for Palm Print. We created our dimensions so that the hot end nozzle is at the edge of the print area in the X and Y direction after homing. Because of this we can use the print bed dimensions for the travel limits.

```
// The size of the print bed
#define X_BED_SIZE 60
#define Y_BED_SIZE 40

// Travel limits (mm) after homing, corresponding to endstop positions.
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS X_BED_SIZE
#define Y_MAX_POS Y_BED_SIZE
#define Z_MAX_POS 75
```

Once all of this is set up you can begin to run test prints. You can then adjust the speeds of the motors, feed rate of the filament, and the temperatures to continue to improve your prints. Pronterface will allow you to control the motors manually and allow you to upload G-code prints. Below is a picture of the main page of Pronterface and all the controls it has.

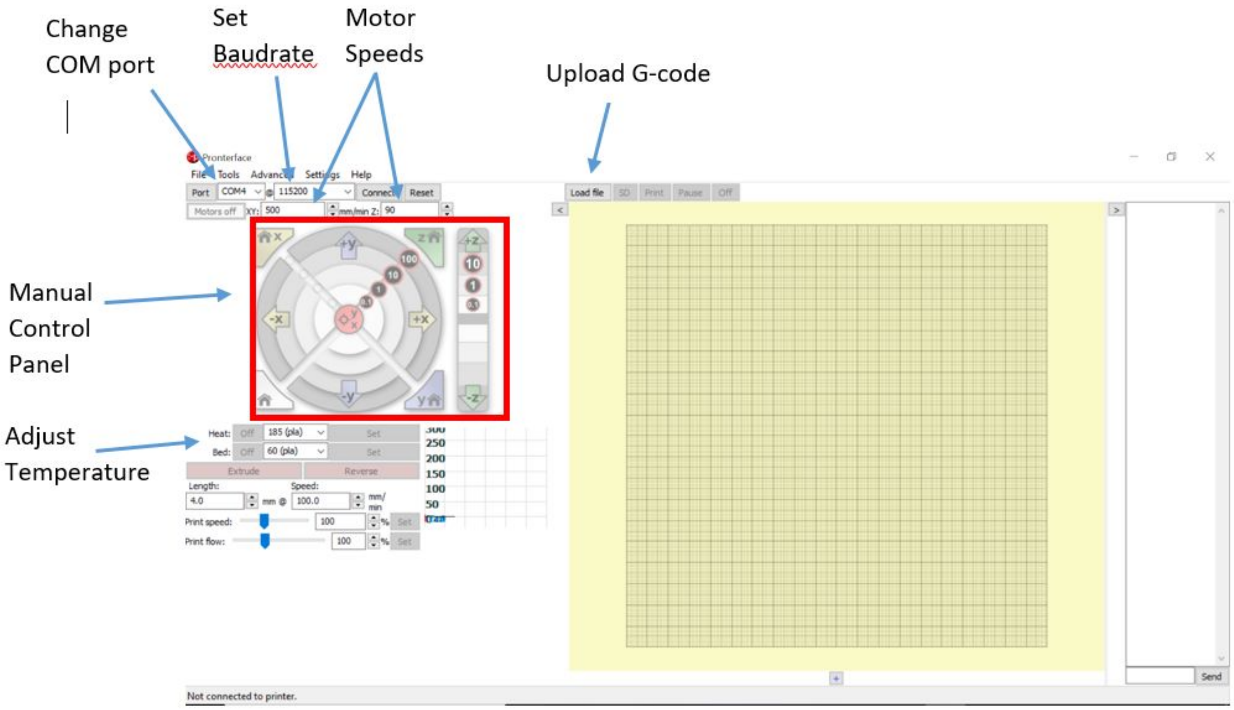


Figure 43: Pronterface main page

You need to input the correct Baudrate and COM port in order for your printer to be able to communicate correctly with Pronterface. As you can see, there are many things you can control from this frame. Speed of the motors, temperature, feedrate of the filament can all be adjusted here. You can also manually control the printer with the control panel in the top left. To upload G-code to the printer click load file above the yellow grid.

3.3 Hacking the Stepper Motor

The stepper motor we used for the X and Y axis movements is the 28BYJ-48 motor. This motor is a 5 wire unipolar motor. We chose this motor because of its affordability. However, because it is a 5 wire unipolar motor it is not compatible with the Ramps 1.4 board that we use. Because of this we had to hack the motor and make it work as a bipolar motor. In order to do this we needed to reconfigure the wire setup of the motor and remove one of the wires. Below is the original configuration of the wires of the motor.

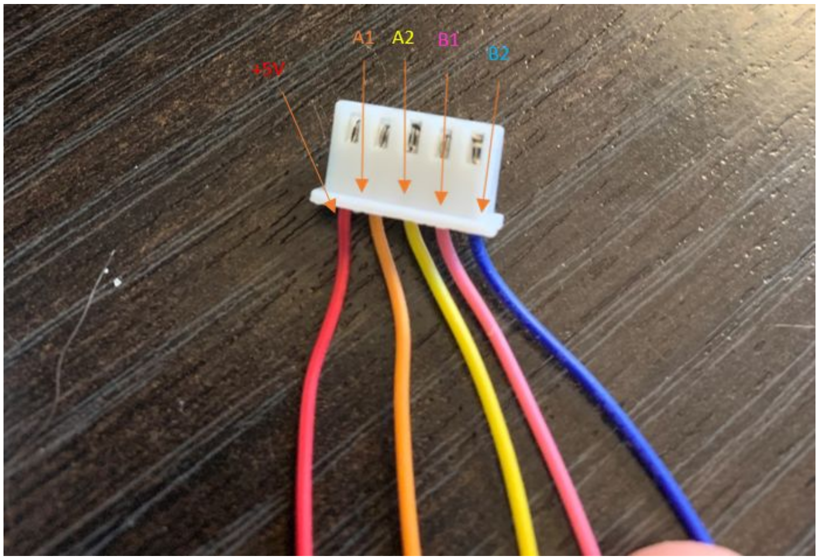


Figure 44: Original Configuration of 28BYJ-48 wires

The red +5V wire is the main difference between the unipolar and bipolar motors.The first thing we need to do is break the connection of the red wire on the circuit board of the motor.

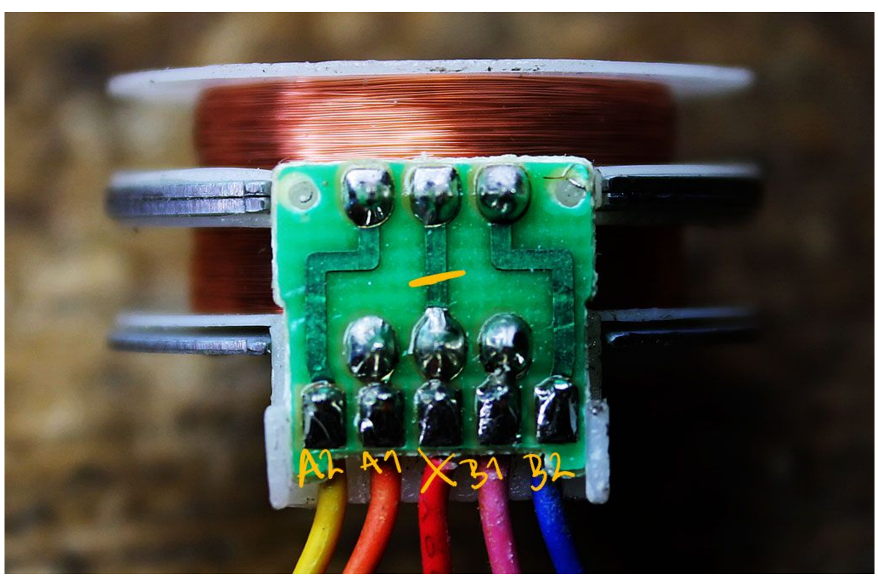


Figure 45: Cutting the Connection of +5V wire

To do this simply take a knife or a sharp object and cut the connection of the middle channel. After that you can desolder the red wire and take it out all together. Once you have done that we still need to reconfigure the A1, A2, B1, and B2 wires. You do not desolder them from the circuit board and rearrange them there, you need to rearrange them in the wiring housing where you will be plugging the motor into the board. Because the board has a certain pin set up, you need to arrange the wires in accordance to the board and the motor driver. Below is the new arrangement of the wires in the housing that will allow the 28BYJ-48 to run as a bipolar motor with the Ramps 1.4 board and A4899 motor driver.

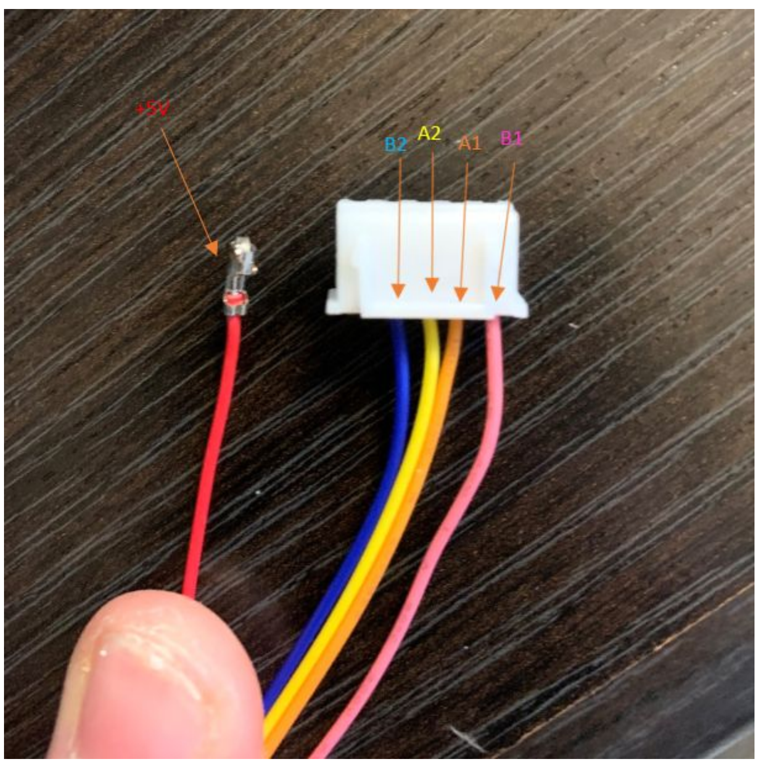


Figure 46: Hacked Wiring Arrangement of 28BYJ-48 Motor

You can either desolder and remove the red wire all together or just remove it from the housing.

4.0 Analysis

4.1 ANSYS

ANSYS Workbench is a platform for providing a comprehensive, integrated, and analytical simulation of a previously designed model. It has many uses, including static structural analysis, which is what we used in our project. After each iteration, we would perform structural analysis on all relevant parts of the design. This program allowed us to test Total Deformation, and Equivalent Strain and Stress of any aspect of the design we felt was necessary to test. Using this program, we were able to see any design flaws, and fix them before actually printing the design, and discovering it would fail in the assembly process. In other words, this program saved our group very valuable time, and it was very instructive to our group. Our ANSYS calculations focused on a few specific aspects of our design. These were movement, shim and rods, and base analysis.

4.2 Base Analysis

This section contains all analyses for the Z arm structure.

4.2.1 Z Arm FEA

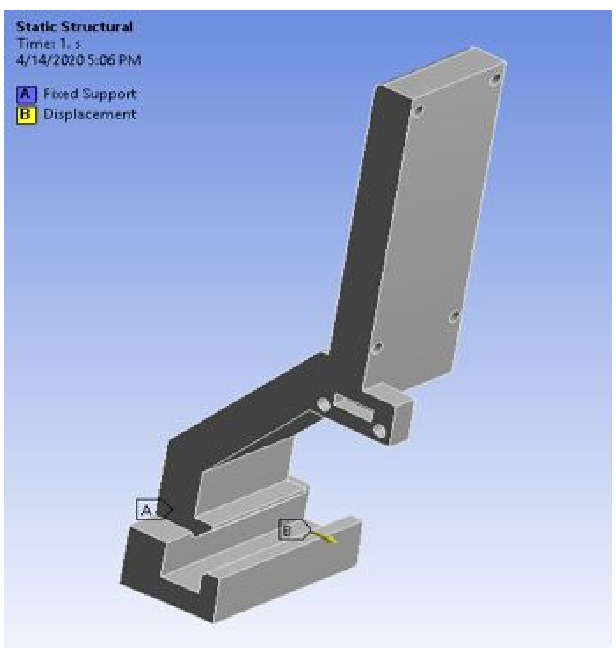
Located in:

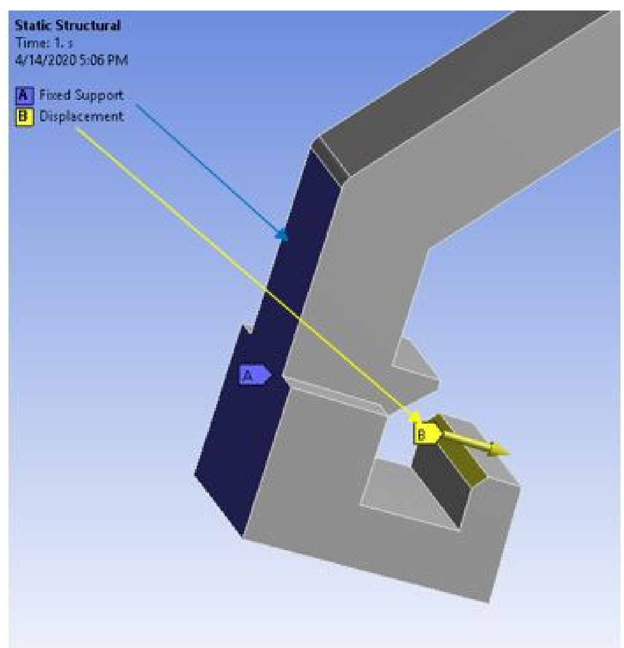
\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 2 Analysis\Base Analysis\Z Arm Analysis\Z Arm Archive

This analysis was designed to test if the hook section of the Z arm would be able to withstand the force needed to assemble it around the base.

Here, the back vertical side of the Z arm is given a fixed support, to mimic a person holding it while assembling the printer. The tip of the hook is given an outward displacement to mimic the bending it must do to fit around the base.

The analysis proved that a slightly stronger material than PLA would be best, as PLA would wear down over time and eventually snap. We decided to print the Z arm using Nylon in the future so that it will be more durable.





Figures 47 and 48: Applied Forces on Z arm ANSYS

The following two images are visuals for the base and Z arm once they are assembled. They are included here for visual clarification. The first is a full view, the second image is a cross sectional view.

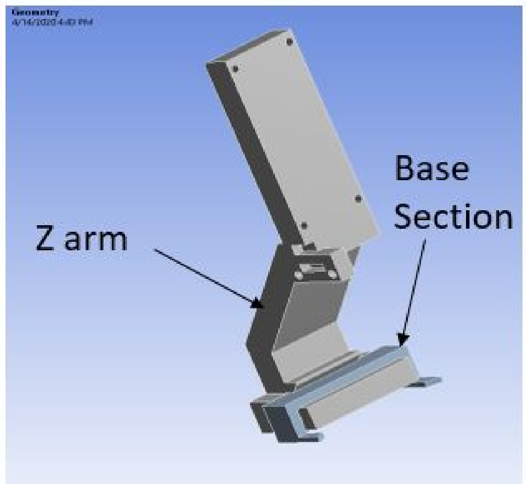


Figure 49: ANSYS Design for Base and Z arm

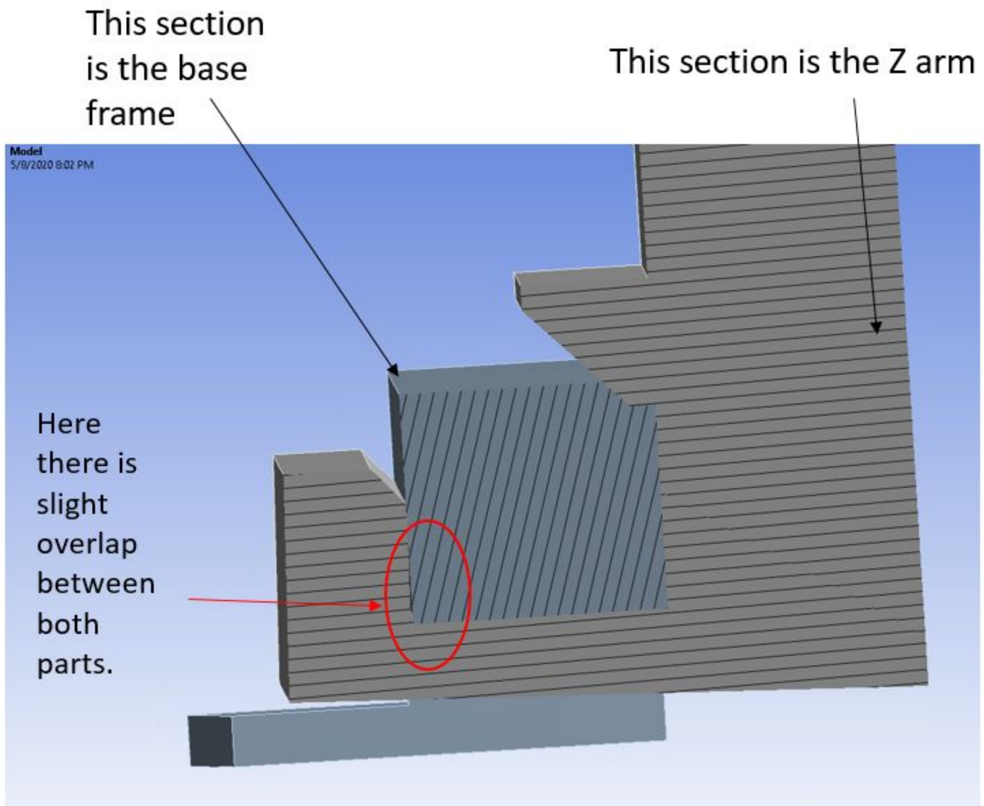


Figure 50: Assembled Base and Z arm section view

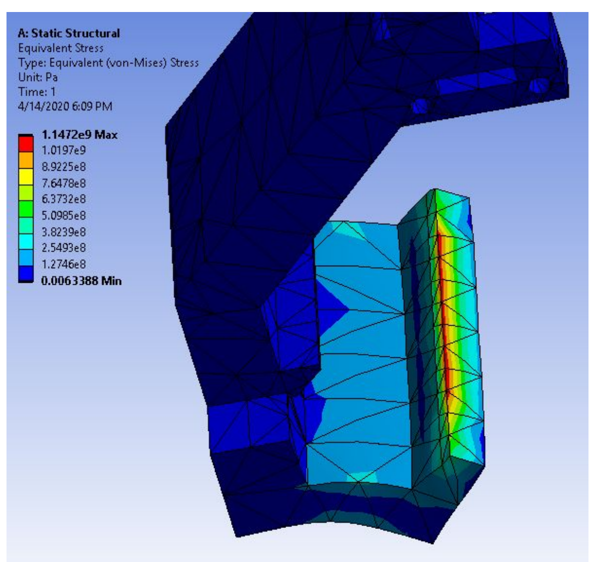


Figure 51: Equivalent Stress

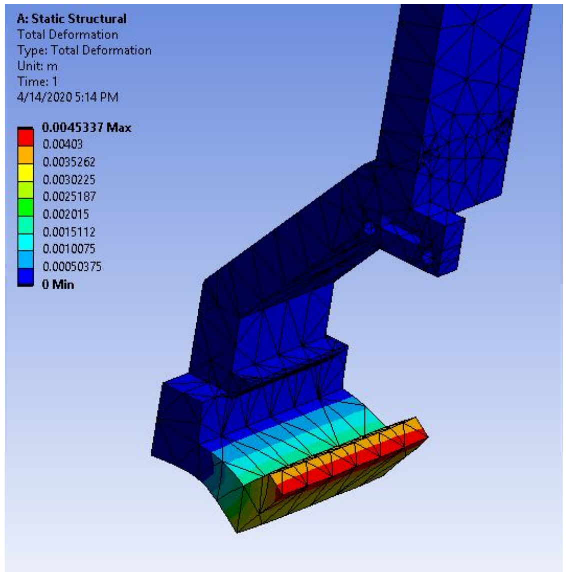


Figure 52: Total Deformation

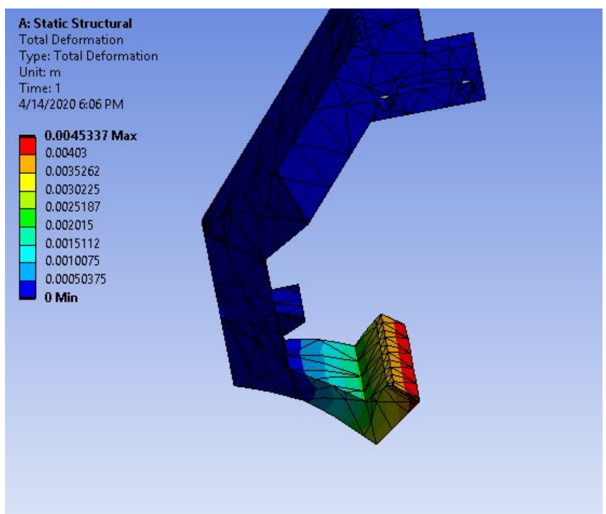


Figure 53: Total Deformation section view

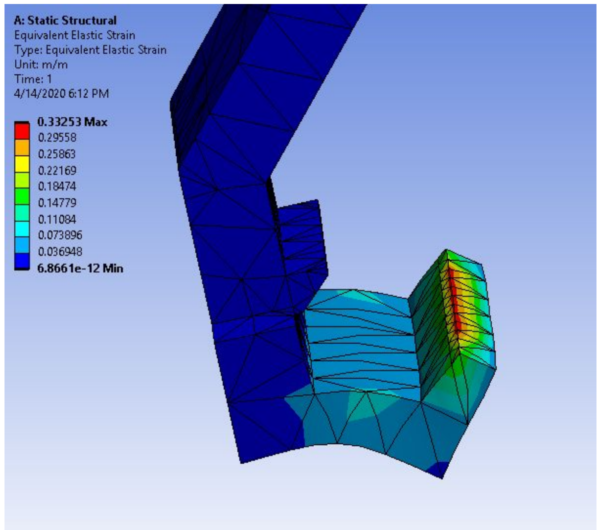


Figure 54: Equivalent Elastic Strain

4.2.2 Z Arm FEA Older Models

Located in:

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 2 Analysis\Base Analysis\Base and Z Arm\Base and Z arm Archive

There are previous designs for the FEA of the Z arm, which included a section piece of the base frame. The design was preassembled. We added fixed supports to the base and the back of the Z arm, and placed a displacement on the hook of the Z arm. This design proved to be confusing and difficult to execute, and the design was aborted. The final FEA did not include the section of the base frame.

4.3 Shim and Rods Analysis

Located in:

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 2 Analysis\Rod and Shim Analysis\! NEW Shim and Rod\Shim and Rod Archive Final

The objective of this design is to test how much friction there will be between the shim and the rods. We also used this analysis to find the force reaction of the shims pushing back against the rods. We were able to use this to calculate the drag force. Using the drag force, we were able to determine how much torque is required from each motor. Our shim is made out of PETG plastic. This has a tensile yield strength of 23 to 58 mpa. The rods are made of stainless steel. Below is the setup for Rod and Shim FEA, as well as pictures of the full Y slider assembly for reference.

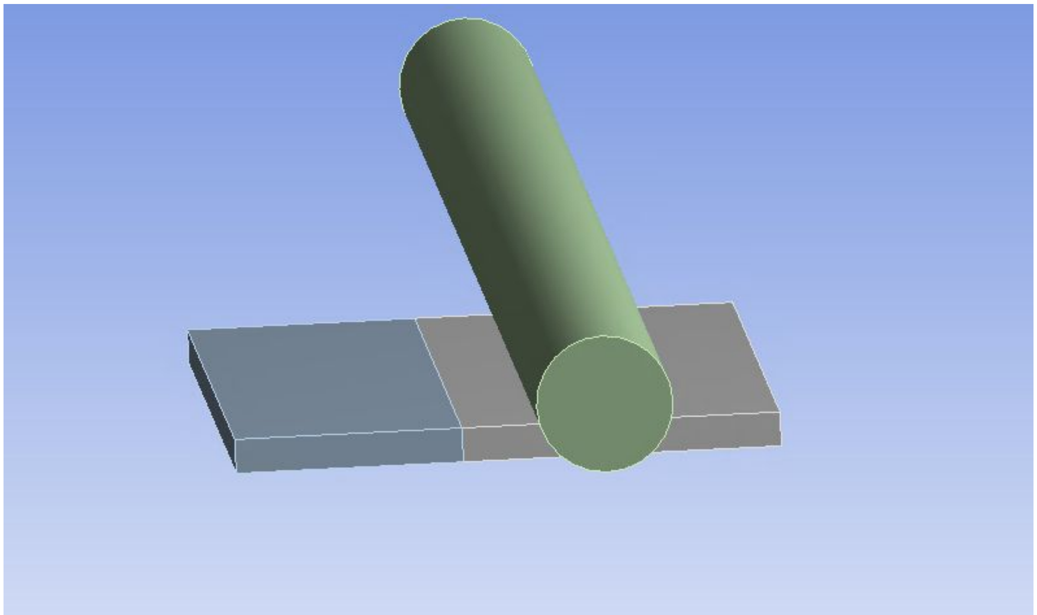


Figure 55: Geometry of Shim and Rod FEA

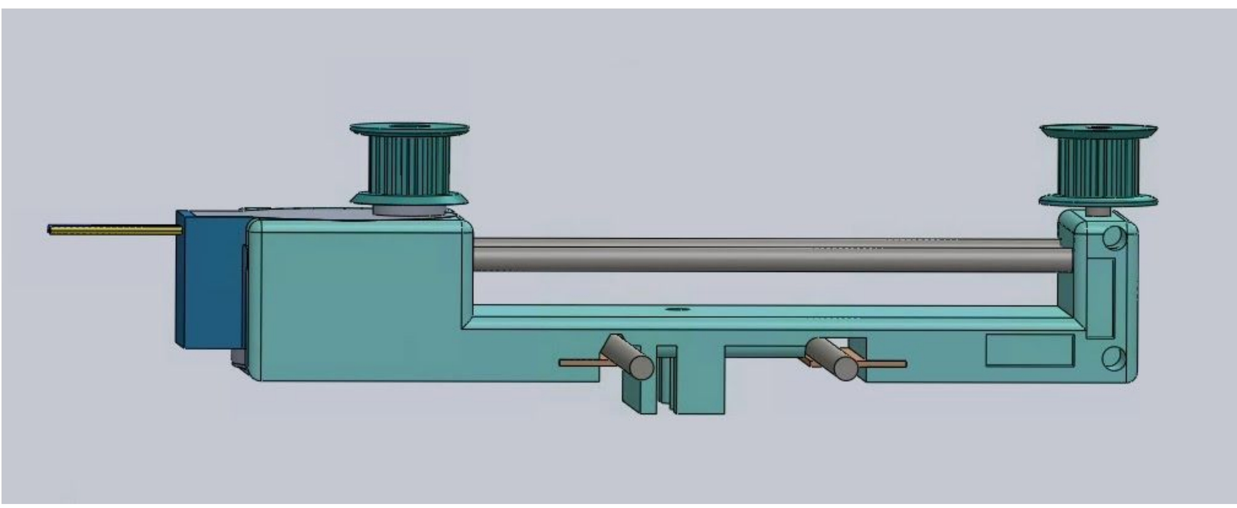


Figure 56: Full Y Slider View Shim and Rod FEA

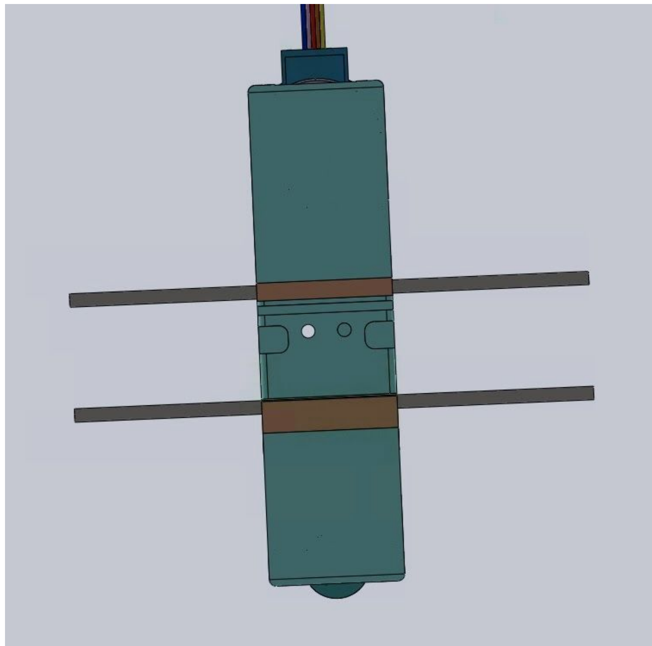


Figure 57: Full Y Slider Bottom View Shim and Rod FEA

For this design, we split the shim into two parts. Shown below, the smaller end of the shim represents the part that is inserted into the slot opening in the Y slider. It was given a fixed support to mimic how it would be held in place. The larger end represents the shim that is outside of the Y slider. The rod sits on the larger section of the shim, and is given a downward displacement of 0.79 mm into the shim to mimic the force of the shim and rod pushing against each other. This allowed us to find the reaction force as well as total deformation, and equivalent stress and strain.

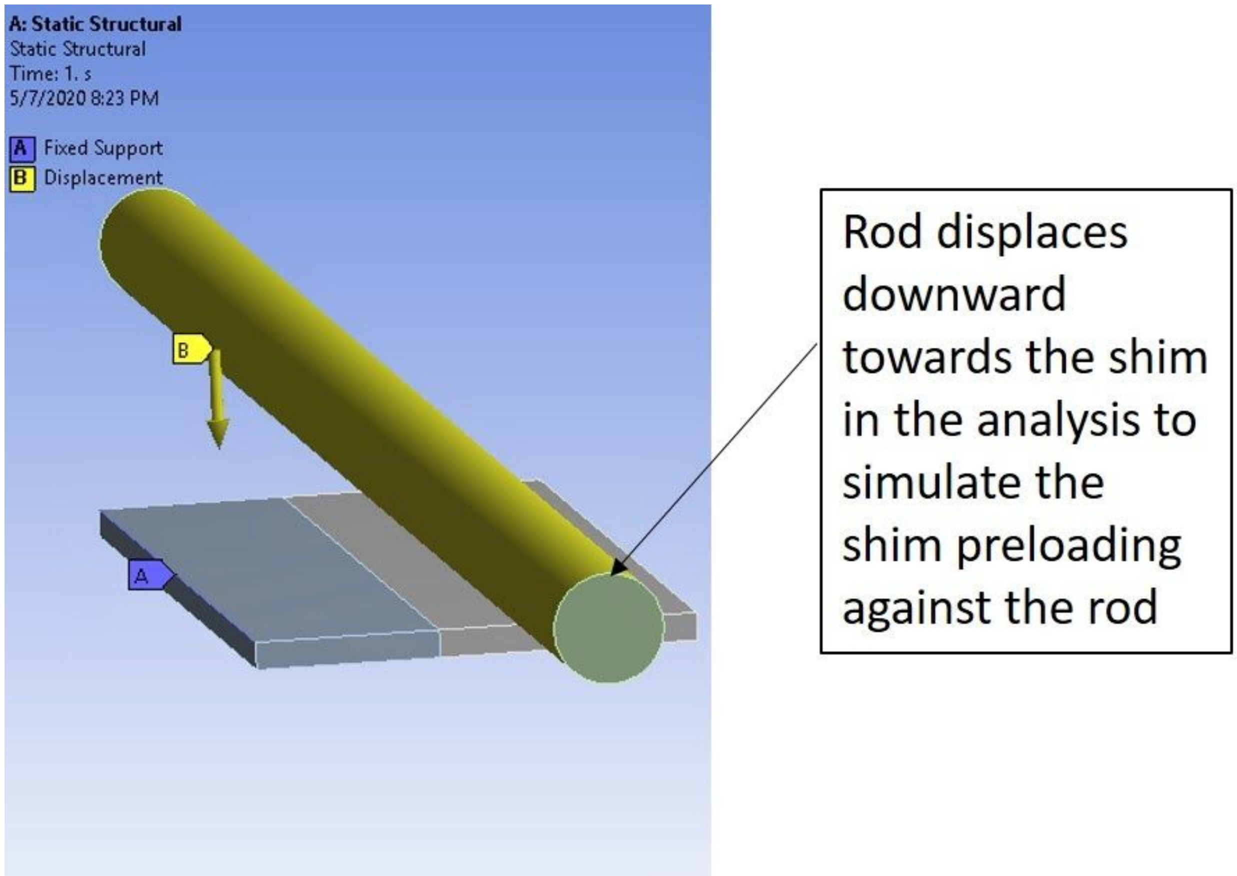


Figure 58: Setup for Rod and Shim Analysis

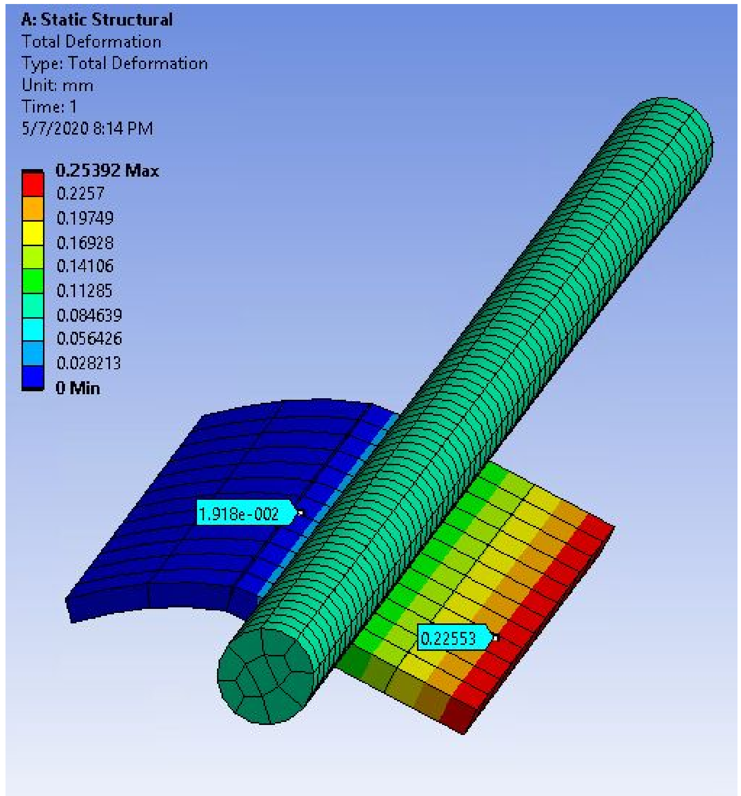


Figure 59: Total Deformation of Rod and Shim

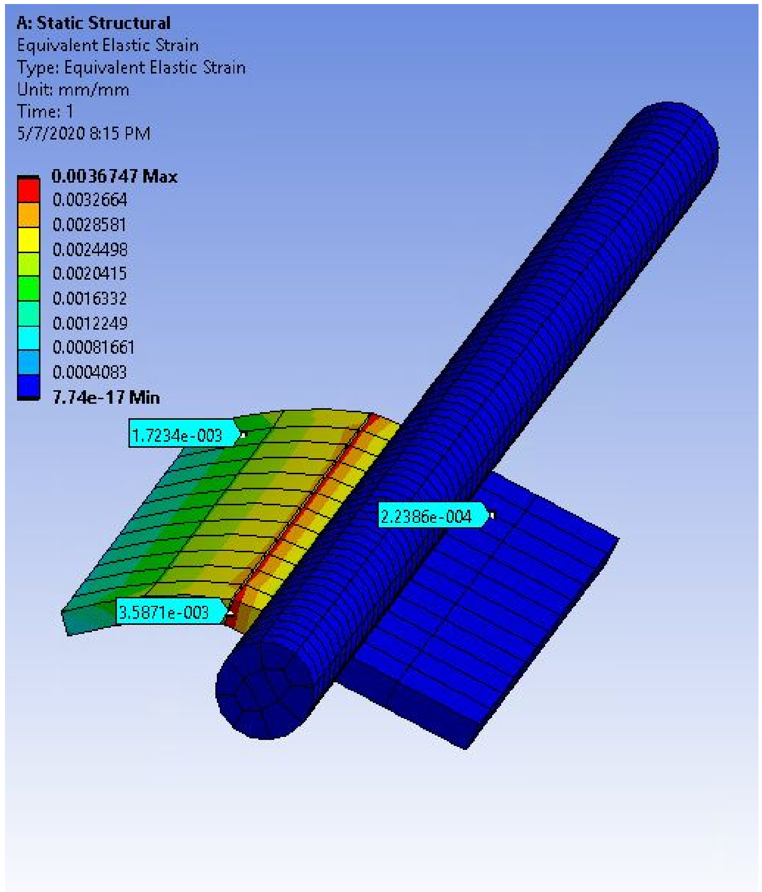


Figure 60: Equivalent Elastic Strain for Rod and Shim

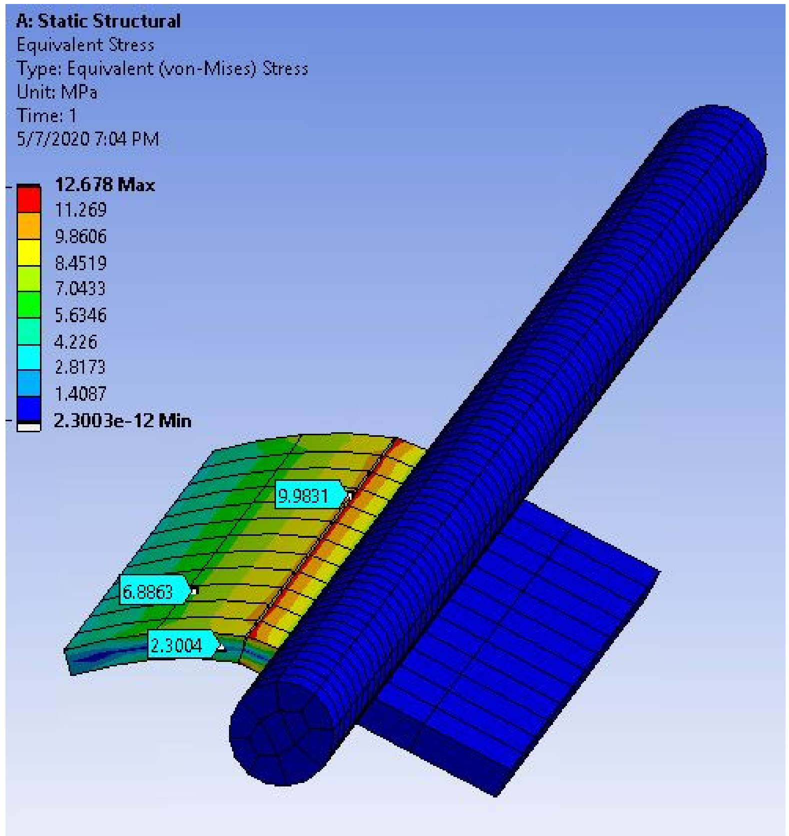


Figure 61: Equivalent Stress

4.3.2 Shim and Rods Older Versions

We performed various FEAs for this design before reaching our ideal setup. Our previous iterations used screw holes to attach the shim to the bottom of the sliders. After careful examination, we decided to use slot insertion instead. Below is a picture of the older model for comparison with our newest model.

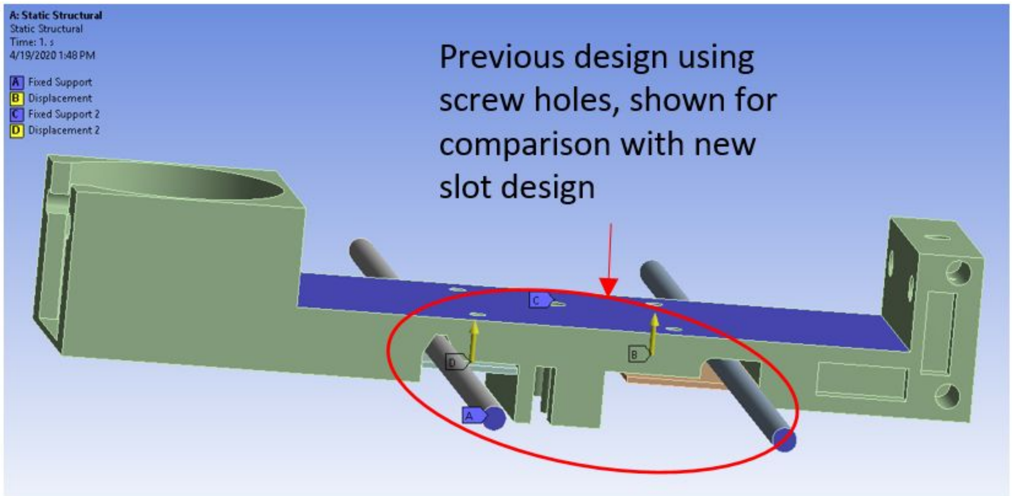


Figure 62: Design of Shim and Rod Analysis

Older versions are located in:

- \\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 2 Analysis\Rod and Shim Analysis\Base Rod and Shim Archived 2
- \\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 1 Basic Movement\X Y Movement\Shim Base and Rod ANSYS Archived
- \\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 2 Analysis\Y Slider Shim Rods\FEA Y Slider Archive

4.4 Movement Analysis

This section contains all analyses of movement of the print bed. It contains X, Y, and Z movement analysis. It also contains older analyses for shim and rod movement.

4.4.1 X Movement Archive

Located in:

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 1 Basic Movement\X Movement

This contains analysis for the print bed moving in the X direction. Here we placed fixed supports on the Y slider, and placed a displacement on the print base to move in the X direction.

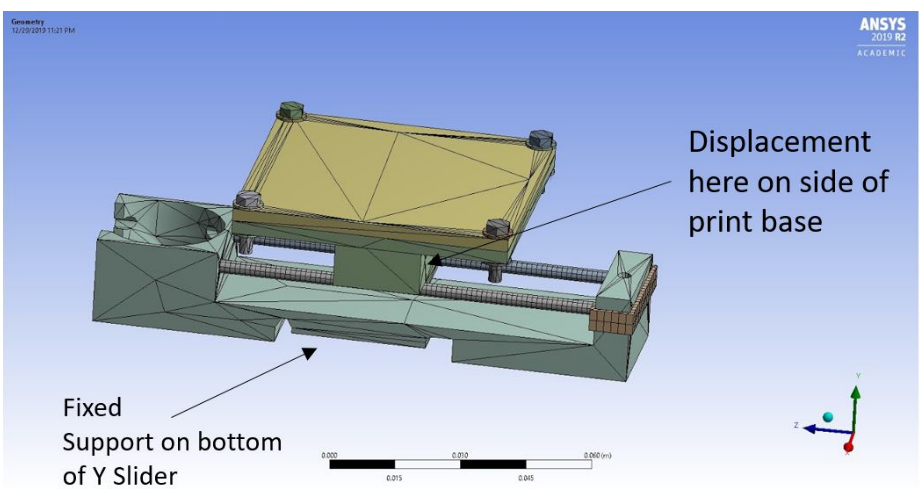


Figure 63: Design of X Movement

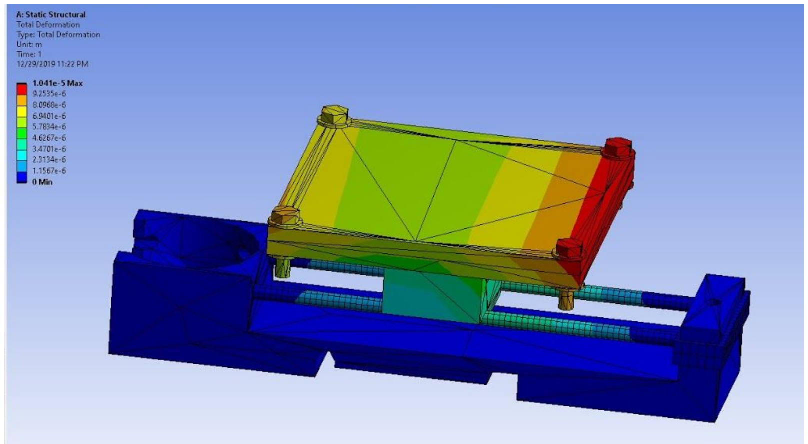


Figure 64: Total Deformation

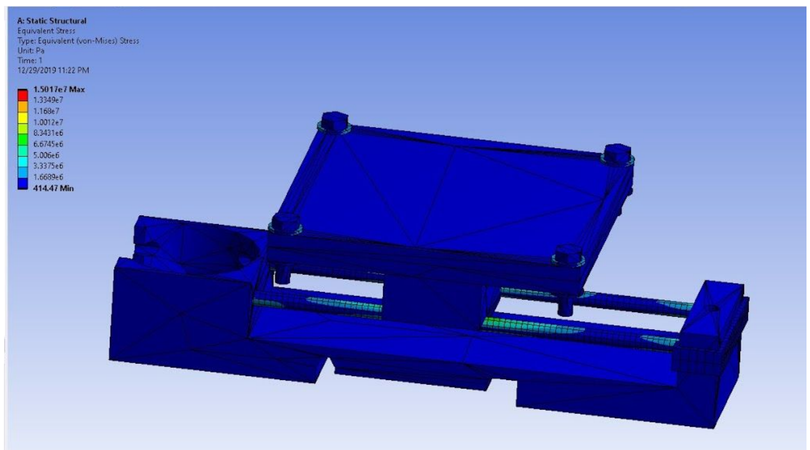


Figure 65: Equivalent Stress

Our results here show us that there will be no problems with movements in the X direction. We included the print bed in the analysis for comparison with the rods. For total deformation, the print bed experiences the highest values. In comparison with the print bed, the rod experiences a negligible amount of deformation, proving that the rods will have no problem handling their task of supporting the print bed. This analysis also proves that the bed will not experience any twisting or bending. Our print bed will be fully capable of completing all needed tasks.

4.4.2 X Y Movement Archive

Located in:

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\ZF_Training\ANSYS\Round 1 Basic Movement\X Y Movement\Base and X Y sliders ANSYS Archived

5.0 Conclusion and Recommendations

By the end of our project we had a working prototype of our handheld 3D printer. This prototype was small enough to fit in the palm of your hand and also had a detachable Z arm to collapse it even further. Our team believes our product would cost around $75 to purchase materials and manufacture, making it very cheap in terms of 3D printing. We were able to run a couple of prints with the printer before we were sent home for the remainder of the year due to the pandemic. Due to that, we were not able to prototype any further and could not make some of the changes we think would be beneficial.

For future students working on this project our team has a few recommendations. To begin, we believe the next stages in prototyping the snap on Z arm should be printed in Nylon instead of PLA. Nylon is more flexible for a snap fit and can better handle that cycled stress. We would also recommend trying to implement the ⅛ inch size timing belts instead of the ¼ inch size. This will allow for smaller pulleys also, ultimately shrinking the size of the printer even more.

6.0 Appendix

Map to CAD Files

Final Model we have assembled

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\2020 Final Model.zip

Final model with updates

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\Palm Print

Smaller Extruder Design

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\28byj48 extruder

Current Extruder

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\Creality Extruder

Hot Ends

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\Hot End

Linear Actuator

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\Linear Actuator

Old Iterations

\\storage.wpi.edu\academics\projects\mqp_small_speaker_systems\2020\Palm_Print\Current_CAD_Files\Old Iterations

7.0 References

-, 3DAddict, et al. “Marlin 1.1 Beginner Guide for 3D Printer Firmware.” 3DAddict, 27 Apr. 2019, 3daddict.com/beginner-guide-marlin-printer-firmware/.

“28BYJ-48 Bipolar Mod.” Ardufocus, ardufocus.com/howto/28byj-48-bipolar-hw-mod/.

“3D Printer Extruder – The Ultimate Guide.” All3DP, 13 Apr. 2020, all3dp.com/1/3d-printer-extruder-nozzle-guide/.

“Arduino MEGA Shield - RAMPS.” DomoticX Knowledge Center, 10 May 2017, domoticx.com/arduino-mega-shield-ramps/.

A. Savini and G. G. Savini, "A short history of 3D printing, a technological revolution just started," *2015 ICOHTEC/IEEE International History of High-Technologies and their Socio-Cultural Contexts Conference (HISTELCON)*, Tel-Aviv, 2015, pp. 1-8, doi: 10.1109/HISTELCON.2015.7307314.

Bethany C. Gross, Jayda L. Erkal, Sarah Y. Lockwood, Chengpeng Chen, and Dana M. Spence Analytical Chemistry 2014 *86* (7), 3240-3253 DOI: 10.1021/ac403397r

Boichut, Philippe. “Direct Drive, Bowden, Remote Motor, the Differences...” SpiderBot by Qualup, 3 Feb. 2019, www.spiderbot.eu/direct-drive-bowden-remote-motor-the-differences/?lang=en.

By. “Direct Drive Extruder VS Bowden Type Extruder-Fundamental Guide On Extruder.” Sovol3d, Sovol3d, 24 May 2019, sovol3d.com/blogs/news/direct-drive-vs-bowden-extruder.

Das, Arnab Kumar. “Crazy Engineer's Drawing Robot Arduino GRBL CoreXY Drawbot.” Hackster.io, 26 Oct. 2019, www.hackster.io/arnabdasbwn/crazy-engineer-s-drawing-robot-arduino-grbl-corexy-drawbot-fb5269.

Earl, Bill. “All About Stepper Motors.” Adafruit Learning System, learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor.

Jones, Douglas W. “Stepper Motor Fundamentals .” BristolWatch, Microchip Technology Inc., 26 Jan. 2004, www.bristolwatch.com/pdf/stepper.pdf.

“Knowledge Base: The Introductory Guide on FDM 3D Printing.” Creality 3D Printers, creality.com/info/knowledge-basethe-introductory-guide-on-fdm-3d-printing-i00168i1.html

“Mechanical Endstop.” RepRap, 27 Dec. 2014, reprap.org/wiki/Mechanical_Endstop.

“Prusa i3 and Variants.” RepRap Forums :: Prusa i3 and Variants, reprap.org/forum/read.php?406%2C785531.

“Stepper Motor Construction.” ISL Products International, www.islproducts.com/designnote/stepper-motor-fundamentals/stepper-motor-construction/.

Younes, Dima, et al. “Design of Control System for 3D Printer Based On DSP and FPGA.” Design of Control System for 3D Printer Based On DSP and FPGA - Volume 6, No. 1, June 2018 - Journal of Automation and Control Engineering (JOACE), 2018, www.joace.org/index.php?m=content&c=index&a=show&catid=67&id=405.