Palm Print: Portable 3D Printer

This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on the web without editorial or peer review.

Submitted By:

Scott Rementer: Mechanical Engineering

Perla Wehbe: Electrical Engineering

Brian Wilkinson: Mechanical Engineering

Advised By:

Joe Stabile: Mechanical Engineering

Table of Figures:

Abstract:

Additive manufacturing (AM) has revolutionized the manufacturing world. It allows for the manufacturing of complex parts that would not normally be possible using traditional manufacturing methods. By using additive manufacturing instead of subtractive manufacturing, a tremendous amount of waste is saved since the part is built up layer by layer to the exact geometry instead of cutting away a piece of material larger than the finished part. Through these advantages, AM has been adopted by many industries such as the aerospace and medical fields.

Since AM is a very cost effective and efficient way of manufacturing, it has become one of the most popular types of rapid prototyping. Rapid prototyping is very important during education because it allows students to work hands-on with their designs and learn more about designing for manufacturing. 3D printers can be very costly to purchase and are very difficult to move, after its initial set up. A cost effective and compact 3D printer would allow students more flexibility by allowing them to manufacture parts almost anywhere. In this project, we look into a cost effective and compact 3D printer that can easily fit inside a student's backpack.

Background:

3D Printing:

3D printing is an additive manufacturing process in which a computer designed model can be printed into a 3 dimensional object. A 3D printer builds up the object layer by layer. For a 3D printer to do this, the computer designed model must be put into a slicer program to break it down into layers. In this program, the user can also adjust printer settings such as speed, infill, material type, support material, and much more. The slicer program then generates a G Code that can be uploaded to the printer to allow it to understand how to print the model. [9]

Components of a 3D Printer:

Print Bed:

The print bed is the surface that the printer will print the object on. The print bed is usually a piece of glass but plastic/rubber print beds are also common. Some materials work better with a certain type of print bed so that is also something to keep in mind. Print beds can be heated or non-heated. The benefit of a heated print bed is that it helps prevent the object from warping during the print because of thermal contraction. A heated bed also allows the printer to be able to print with a larger variety of materials whereas a non-heated bed limits the materials that can be used. It is very important that the bed is level or else you will run into problems when printing. A print bed can either be physically leveled or the printer itself can use a program to account for any variations in the levelness. The print bed must also be removable so that it can be cleaned and to also help with removing printed objects. [1]

Extruder:

The extruder, or the printing head, is the part that is responsible for melting and dispensing the filament material. The cold end of the extruder pulls in the filament from the spool, and the hot end melts the filaments and dispenses it. The filament is dispensed through a nozzle on the end of the extruder. Nozzles are interchangeable and come in a variety of different sizes. A smaller nozzle is great for fine details, but they result in a longer printing time. A large nozzle is great for objects that do not require fine detail and will result in a faster print. The nozzle being used can have a big effect on the quality of your print. It is also common for nozzles to get clogged, which is one of many issues that can be experienced when trying to print something. [1]

Types of Motion:

There are two types of motion controls that 3D printers can utilize. The first is Cartesian. Cartesian printers move linearly in each the X, Y, and Z axes. These printers tend to be a cube like shape. The second type of printer is a delta printer. These printers also move in the X, Y, and Z axes but they do not move linearly in each direction like a Cartesian printer. Delta printers move with triangulation and tend to be a cylinder shape. [7]

Methods of Motion (Linear Actuators):

Each axis of a 3D printer can be moved using either a threaded rod and leadscrew system or a belt and pulley system. Most printers use a threaded rod and leadscrew for the Z axis, and belts and pulleys for the X and Y axis. A threaded rod and leadscrew works by rotating the threaded rod, forcing the leadscrew to thread up or down the rod based on the direction of rotation. A belt and pulley system works by having pulleys at either end of the axis, with the belt attached to the print bed or extruder, depending on the design of the printer. When a pulley is rotated, it causes the belt to spin, moving either the print bed or extruder. Each axis, whether using a threaded rod and leadscrew or a belt and pulley system, will utilize a rail system for added support and to ensure true linear motion. [7]

One type of guide that can be used is ball guides. These guides use ball bearings to ride on a smooth rod, providing very smooth motion. This type of guide can handle very heavy loads and has the highest stiffness out of the three types of guides. One of the drawbacks to this type of guide is that they are usually very expensive and can be loud when moving at a high velocity. The next type of guide is wheel guides. The guides have a center rail, usually with guides that the wheels fit into. The wheels are mounted onto the moving component and hug the rail. This type of guide operates with very low friction and still maintains a high stiffness. One of the drawbacks to this type of guide is that they can be easily damaged by shock loads. The third type of guide is a slide guide. This type of guide uses smooth metal rods that a cart can slide back and forth on. These guides are very quiet and can handle high shock loads very well. These guides can only handle lighter loads and slower movement. [7]

Electrical Components:

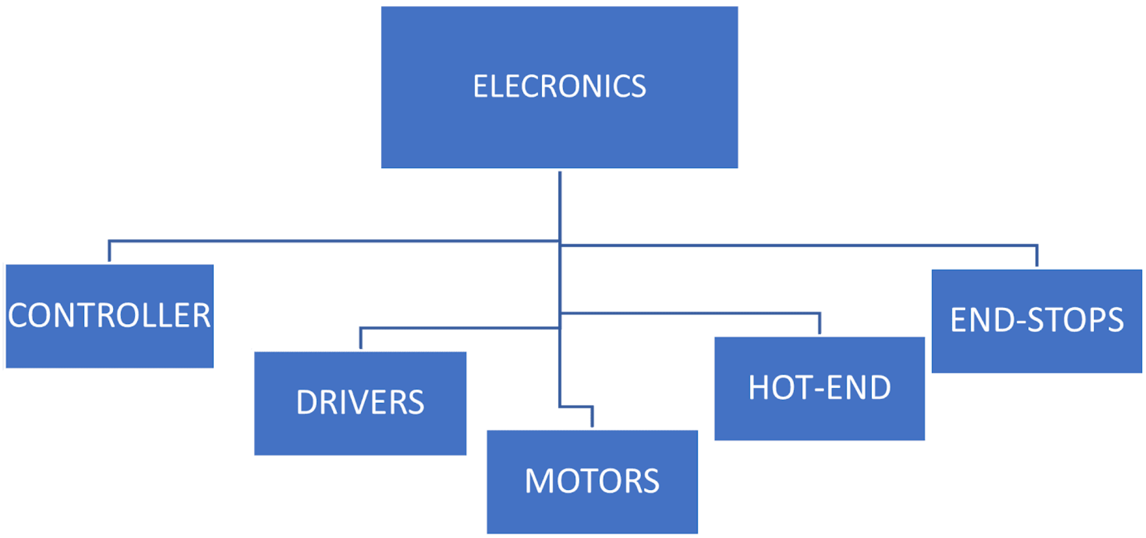


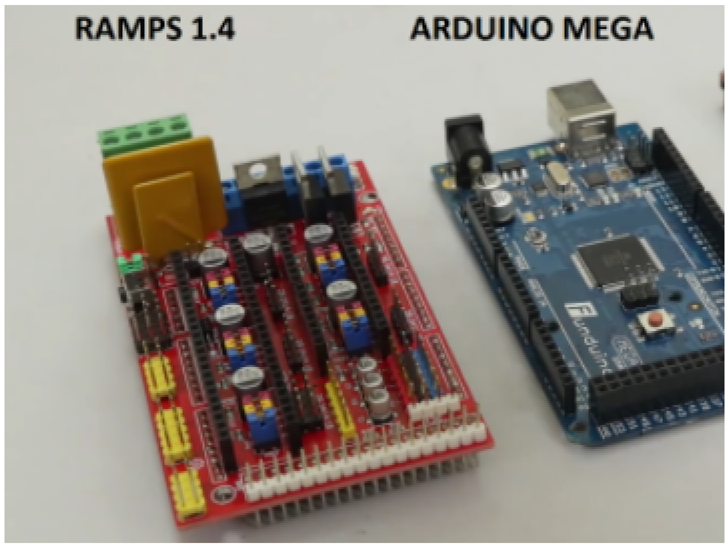
Figure 1: Electrical component diagram

Controller boards:

The controller boards are the brain within our 3D printer. The controller board takes the inputs (G-code) and yields to the rest of the printer's mechatronics what to do with it. That interpretation is done by the "firmware" introduced within the controller board. We will talk about firmware later in the report.Previously, the project was accomplished using the Mega 2560 and RAMPS combination.

Arduino Mega 2560 and RAMPS1.4:

The arduino Mega 2560 handles all of the computation and interfaces with the RAMPS board that does power handling and IO for each of the components. This combination requires the addition of stepper drivers to attach to the RAMPS. The Mega 2560 uses an 8 bit processor and the combination permits the user to attach a second extruder, an SD reader port and an LCD screen. It allows the user to tweak the rotation speeds in the stepper motors. It ensures the synchronisation of the timing of the coils within the motors. It controls the stepper motors, the end-stops and monitors and modifies the temperature of the hot-end of the extruder.



https://www.creativitybuzz.org/diy-3d-printer/#Step_1_3D_Printer_Parts

Figure 2: Arduino and ramps boards

Sanguinololu Board:

Another board that can be used and is based on the Arduino boards is the Sanguinololu board. It processes G-code instructions and can control Polulu stepper motor drivers. Its function is similar to the RAMPS. However, the Sanguinololu has all the user needs on one board. The RAMPS require the Ardiuno microcontroller. Plus, the Sanguinololu does not connect to a second extruder or an LSD screen and we cannot add an SD reader port.

Stepper Motors and Drivers:

The stepper motors are responsible for the movement of the arms, via the use of timing belts and pulleys. We are using bipolar NEMA stepper motors. Bipolar motors are used for high performance. NEMA motors are capable of producing a very high resolution and are very suitable for applications that require high precision.

The stepper motor drivers feed the motor with power pulses. The power pulses enable the motors to move through measured steps. NEMA motors have twice the step accuracy of a typical motor and Bipolar motors are used for high performance. It is possible to control the stepper motor through micro-stepping. The shaft of the motor will move in smaller steps ;1 step, ½ step, ¼ step, ⅛ step ,and 1/16 step.

To calculate the X and Y steps, we need to know the motor step angle, the belt pitch, the pulley tooth count, the driver microstepping and the belt presets. If a stepper motor has a 1.8 degrees step size (200 step per revolution), it represents the amount the shaft rotates in one step. To achieve intermediate steps, the current supply levels need to be modified. Therefore, it is important to use a potentiometer on the stepper driver to determine the amount of current sent through the motor.

Software:

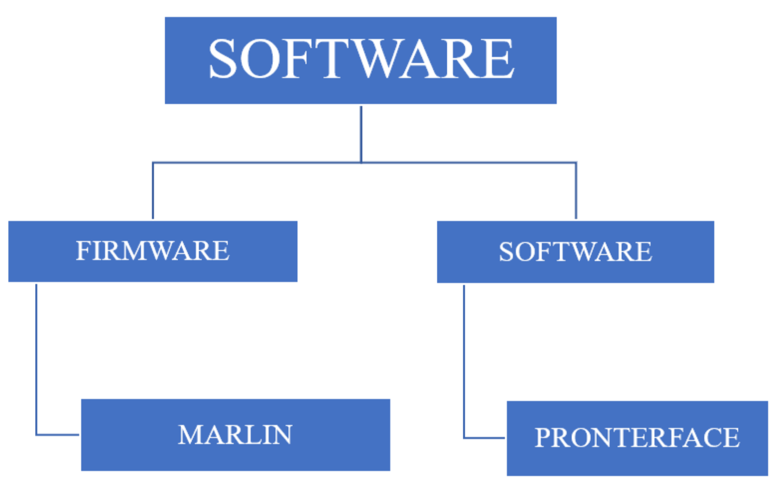


Figure 3: Software diagram

MARLIN:

The firmware is capable of making the printer move agreeing to the G-code commands. The firmware will be able to calculate the distances each axis is allowed to move in order to reach the required position. To configure the firmware, we downloaded and installed the arduino IDE. We also downloaded and extracted the Marlin firmware. We edited the Marlin firmware and uploaded it to the RAMPS. By modifying the X and Y steps configuration with Marlin Firmware, we can get an X and Y movement.

PRONTERFACE:

Pronterface is a simple Graphical User Interface software that permits the user to input G-code instructions to the controller. This information is sent to the controller, via the firmware which will control the movement of the printer.

Existing/Similar Solutions:

TOME 3D Printer:

Figure 4: TOME 3D printer

The TOME 3D Printer is a 3D printer that was developed to be completely portable. The overall dimensions of the TOME are 4” x 8” x 11” and has a print volume of 5” x 5” x 5”. The printer is battery powered and has the capability to print for 4-6 hours on one charge. The printer is designed to print PLA and has a heated print bed. The TOME is also WiFi enabled which allows users to request prints from one another. The model shown above is designed for field work and is contained within a rugged case. The inventors hope to offer different versions of the printer including one with a leather case for normal everyday use. For field use, the printer can be charged using solar or wind power. The inventors hope to be able to retail the TOME for less than $1500. [8]

PocketMaker:

Figure 5: PocketMaker 3D printer

The PocketMaker was designed to be very compact and portable. This printer is very lightweight, weighing only 1.87 pounds. It has a print volume of 3.15” x 3.15” x 3.15”. The printer is designed for either ABS or PLA filament but can be used with other materials too. The PocketMaker can also be controlled with a smartphone using their app, eliminating the need for a computer. The PocketMaker is also very affordable. It retails for only $99. [4]

Pocket3DPrinter:

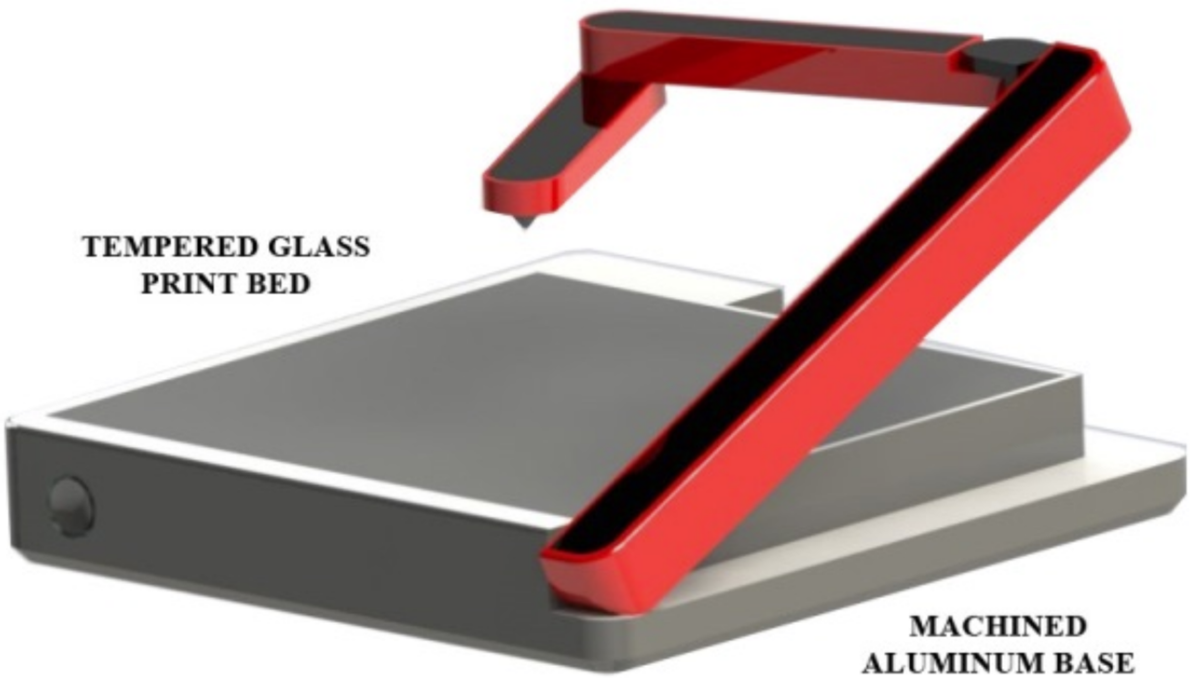


Figure 6: Pocket3DPrinter

The Pocket3DPrinter was designed to be very portable. This printer, when folded, is not much bigger than a standard iPad. It is constructed using machined aluminum and tempered glass. This printer uses photosensitive resin instead of standard molten plastic. There is a UV LED on the tip of the printing arm that cures the resin as soon as it is dispensed. This eliminates the need to allow the printer to cool down before transporting it. The printer comes in two sizes. The smaller of the two being 1” x 6” x 7” when folded, and has a print volume of 6” x 6” x 5”. The larger of the two is 1” 8” x 9” when folded, and has a print volume of 8” x 8” x 7”. The printer is battery power and has the ability to print for up to 3 hours. The smaller version of the printer costs $299 and the larger version costs $449. [5]

Design Brainstorms:

Cartesian Design:

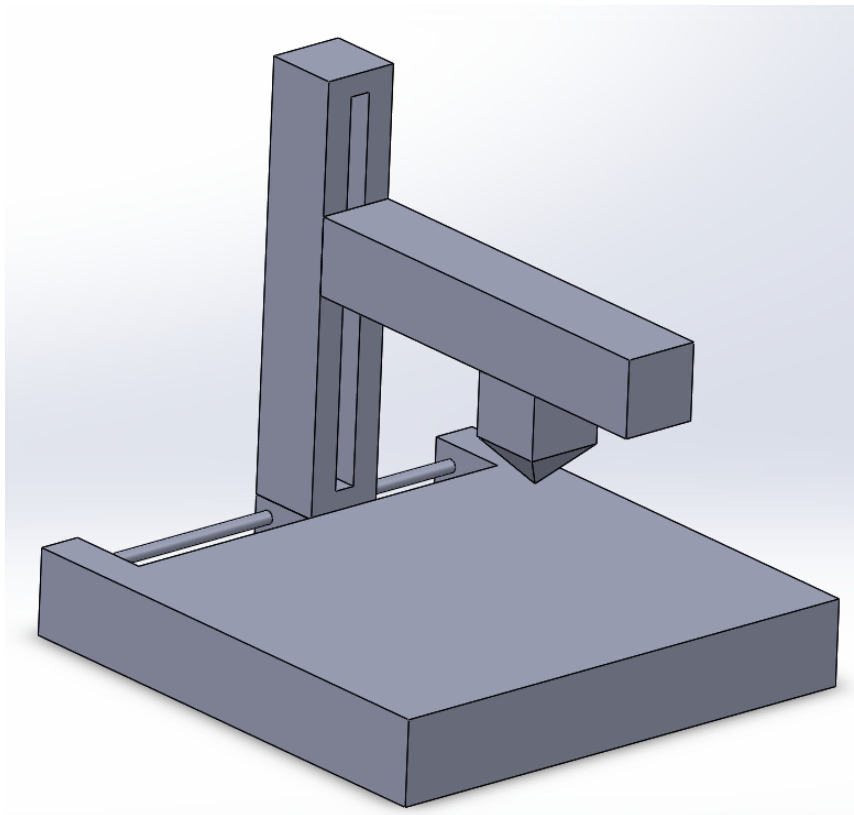


Figure 7: First cartesian design iteration

The first design that we came up with was a Cartesian style printer. This design had a fixed print bed and the print head moved in all 3 axes. Each axe moved independent of each other and would be controlled by their own stepper motor. Each axe would also be moved using a threaded rod and leadscrew. There would be polished guide rods within each of the arms as well as the back side of the base for added support. The design is 6” x 6’ x 6”. For portability, the vertical arm would move to a “home” position at one end of its axis, then the arm would be able to fold over the print bed.

Core XY Design:

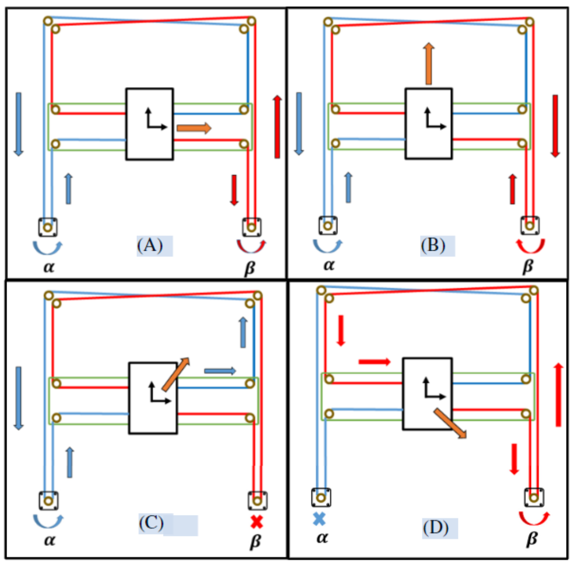


Figure 8: Core XY diagram

Another potential design that we explored was using a Core XY mechanism in the vertical plane to control the z axis and the x axis. A Core XY mechanism is made up of two motors, two open ended timing belts and eight pulley points to control the movement of the print head. Instead of each axis of motion being independently controlled by one motor, two motors work together to accomplish the desired direction of motion. As for the basic movements of this mechanism, using one motor you would pull the printhead at a 45 degree direction away from the driving motor. When each motor operates in a different direction, the print head will either travel up or down. When each motor operates in the same direction, the print head will either move directly right or left. Combining these three forms of motion allows the Core XY mechanism to accomplish precise movement in two axises of movement. For our original iteration using this mechanism we planned on having a fixed print bed while the printhead is controlled by Core XY and the third axis of motion is also directly attached to the Core XY mechanism. Eventually we found this version to be difficult to realize so this design was no longer worked on. [2]

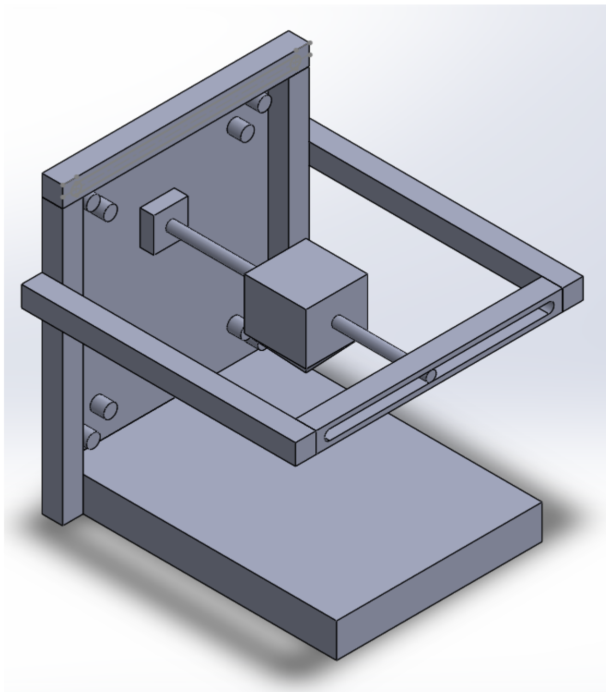


Figure 9: First core XY design iteration

Linear Actuator Design:

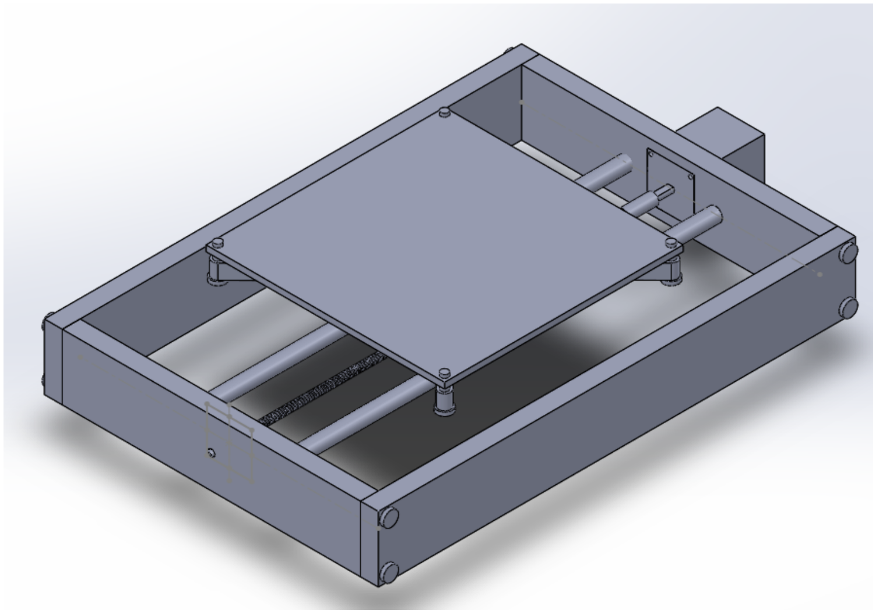


Figure 10: Base linear actuator (two guide rods)

Once our team explored different options for our 3D printer, we were able to decide on a base linear actuator that would be able to be easily adapted to different final designs. The linear actuator consists of a 7in x 5in base frame with a 4in x 4in print bed that moves in one axis. This motion will be controlled by a screw drive driven by a NEMA 8 stepper motor that would be directly attached to a 3mm threaded rod using a shaft coupling. This design originally had two 6mm steel guide rods for stability but this was recently changed to one guide rod to save room in the printer base. The frame will initially be made out of PLA to simplify the manufacturing process while the remainder of joints in the frame will be held together by 1/8 inch diameter screws. The print bed itself was originally designed to be manually levelled by spring loaded screws attached to the print bed and the carriage. Currently we are also exploring the option of a fixed print bed that is leveled digitally using software.

Stepper Driver Board Design:

To create a stepper driver board, we used the EAGLE software. EAGLE is electronic design automation (EDA) software where it is possible to add components, create schematics, and rout PCBs. After studying the software, we established a guide to administer the process of building a stepper driver using EAGLE.

Download the EAGLE software:

Autodesk offers Eagle Software free of charge. It is a limited version which includes two schematic sheets, two signal layers, and an $80cm^2$ board area. Use the link below for a free download.

https://www.autodesk.com/products/eagle/free-download?plc=F360&term=1-YEAR&support=ADVANCED&quantity=1

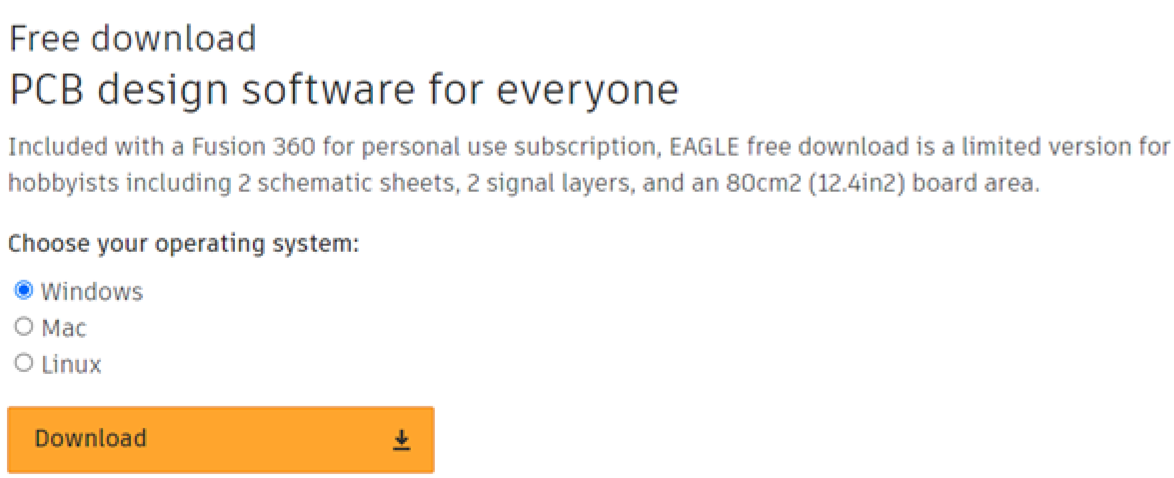


Figure 11: Process for EAGLE software download

Choose the proper operating system, then press the download button. Once downloaded, setup EAGLE by agreeing to the license agreement. Then, select the preferred destination location.

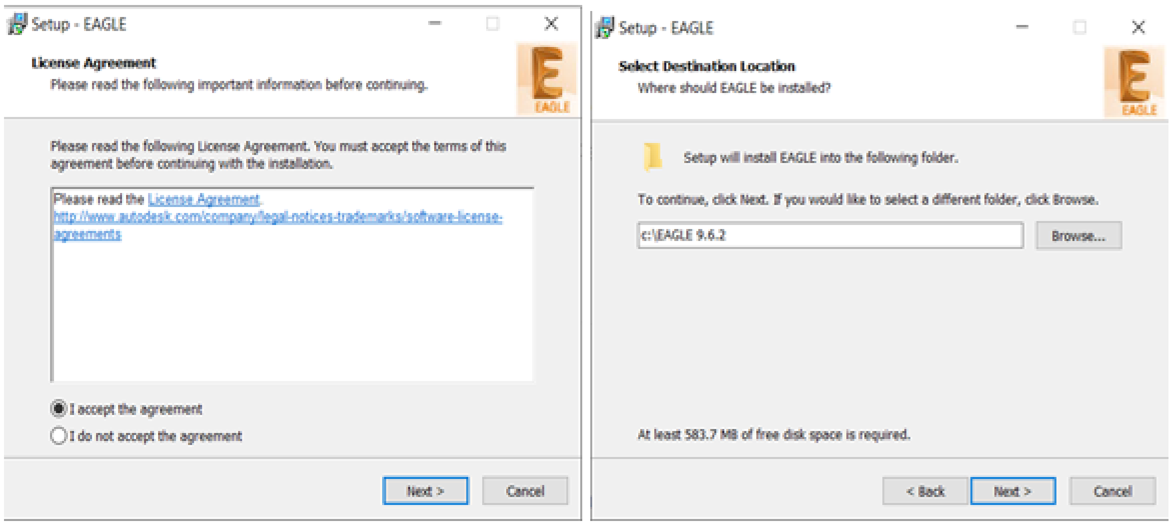


Figure 12: The process for EAGLE software installation

Choose the proper IC to Drive a Bipolar Stepper Motor:

The first step is to choose the correct stepper driver IC. In the previous project, A4988 driver was used for the RAMPS. However, we decided to use a similar IC stepper driver with larger leads pointing outwards, the DRV8825. The DRV8825, has two H-bridge drivers and a six microstep resolution. It is designed to drive a bipolar stepper motor. It features adjustable current limiting, over-current and over-temperature protection. Use the link to access the datasheet.

https://www.ti.com/lit/ds/symlink/drv8825.pdf

The first page of the datasheet provides the features, applications, description, and simplified schematic of the IC interpreted in the previous section. In the Description section, note the Device Information table

PART NUMBER	PACKAGE	BODY SIZE (NOM)
DRV8825	HTSSOP (28)	9.70 mm × 6.40 mm

Figure 13: DRV8825 package information

From the table, the package version for the DRV8825 is HTSSOP 28. HTSSOP stands for Thermal Enhanced Thin Shrink Small Outline Package. The number 28 stands for termination count (28 leads).

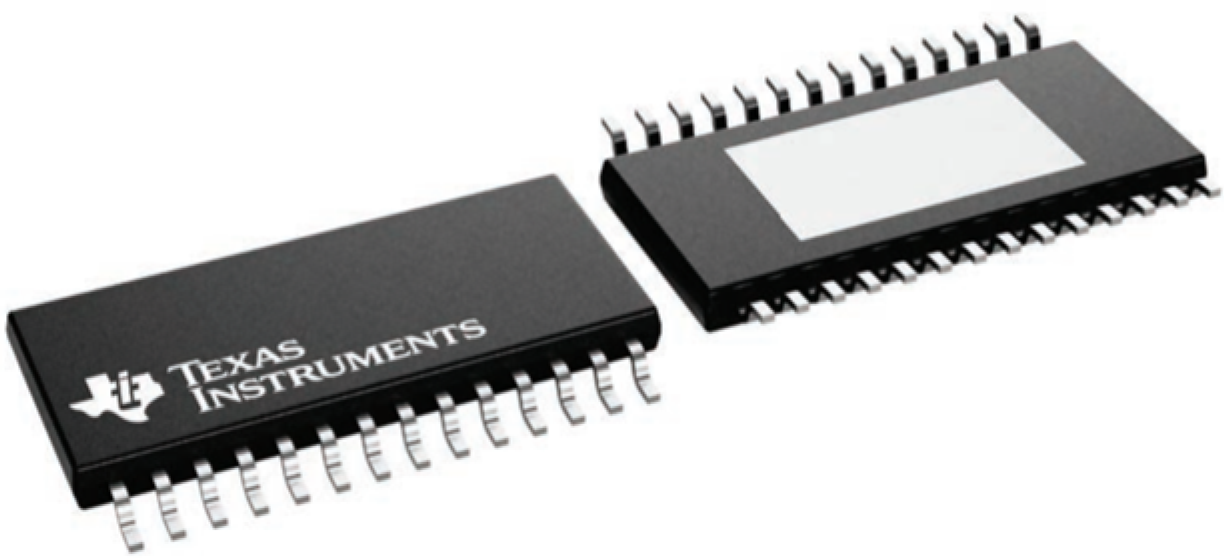
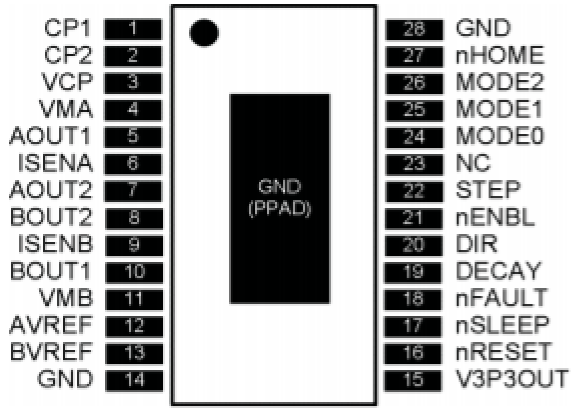


Figure 14: Representation of the package family TSSOP

Section 6 in the datasheet, shows the pin Configuration and Functions.

6 Pin Configuration and Functions



Pin Functions

PIN		I/O (1)	DESCRIPTION	EXTERNAL COMPONENTS OR CONNECTIONS
NAME	NO.			
POWER AND GROUND				
CP1	1	I/O	Charge pump flying capacitor	Connect a 0.01-μF 50-V capacitor between CP1 and CP2.
CP2	2	I/O	Charge pump flying capacitor	
GND	14, 28	—	Device ground	
VCP	3	I/O	High-side gate drive voltage	Connect a 0.1-μF 16-V ceramic capacitor and a 1-MΩ resistor to VM.
VMA	4	—	Bridge A power supply	Connect to motor supply (8.2 to 45 V). Both pins must be connected to the same supply, bypassed with a 0.1-μF capacitor to GND, and connected to appropriate bulk capacitance.
VMB	11	—	Bridge B power supply	
V3P3OUT	15	O	3.3-V regulator output	Bypass to GND with a 0.47-μF 6.3-V ceramic capacitor. Can be used to supply VREF.
CONTROL				
AVREF	12	I	Bridge A current set reference input	Reference voltage for winding current set. Normally AVREF and BVREF are connected to the same voltage. Can be connected to V3P3OUT.
BVREF	13	I	Bridge B current set reference input	
DECAY	19	I	Decay mode	Low = slow decay, open = mixed decay, high = fast decay. Internal pulldown and pullup.
DIR	20	I	Direction input	Level sets the direction of stepping. Internal pulldown.
MODE0	24	I	Microstep mode 0	MODE0 through MODE2 set the step mode - full, 1/2, 1/4, 1/8/ 1/16, or 1/32 step. Internal pulldown.
MODE1	25	I	Microstep mode 1	
MODE2	26	I	Microstep mode 2	
NC	23	—	No connect	Leave this pin unconnected.
nENBL	21	I	Enable input	Logic high to disable device outputs and indexer operation, logic low to enable. Internal pulldown.
nRESET	16	I	Reset input	Active-low reset input initializes the indexer logic and disables the H-bridge outputs. Internal pulldown.
nSLEEP	17	I	Sleep mode input	Logic high to enable device, logic low to enter low-power sleep mode. Internal pulldown.
STEP	22	I	Step input	Rising edge causes the indexer to move one step. Internal pulldown.
STATUS				
nFAULT	18	OD	Fault	Logic low when in fault condition (overtemp, overcurrent)

(1) Directions: I = input, O = output, OD = open-drain output, IO = input/output

Figure 15: Pin configurations and functions

In the table, the column “External Components or Connections”, assesses the components and their connections between the IC pins.

List of Components from the datasheet:

- 0.01-µF 50-V capacitor
- 0.1-µF 16-V ceramic capacitor
- 1-MΩ resistor
- Appropriate bulk capacitance
- 0.1-µF capacitor
- 0.47-µF 6.3-V ceramic capacitor
- Two Current sense resistors

Selecting the appropriate current sense resistors:

Holding torque of a stepper motor measures how much rotating force is recommended to force a stationary stepper motor shaft out of position. Holding torque is related to a motor’s torque constant and the current applied to the stator windings. In stepper motors, a fixed-frequency current regulation (current chopping) is applied to set the appropriate current through the two windings for smoother operation. If the current attains the current chopping threshold, the H-bridge disables the current until the beginning of the next PWM cycle. The chopping current is established by a comparator that will compare the voltage across a current sense resistor connected to the xISEN pins.

Chopping current is calculated as follows:

$I_{chop} = V_{xREF}/(5*R_{Isense})$

V_{xREF} is the input voltage from the xVREF pin.

Using 0.1Ω current sense resistor, the formula becomes the following:

$V_{xREF} = I_{MAX}*0.5$

From the DRV8825 datasheet, AVREF and BVREF are bridge A and B current set reference inputs. V3P3OUT is 3.3V regulator output. The voltage reference VREF is determined depending on the current rating of the stepper motor. Therefore, AVREF and BVREF pins are connected to a trimmer potentiometer which is also connected to the 3.3V regulator output and ground. Subsequently, the value of the trimmer potentiometer is modified to produce the VREF necessary to set the equivalent current rating.

Selecting the appropriate bulk capacitance:

A bulk capacitance is required and must be sized accordingly to the application constraints. The datasheet usually provides the value of the bulk capacitance. However, system level testing is required to identify the appropriate value due to its dependence on a variety of factors. More information on the bulk capacitance shown in section 10.1 in the DRV8825 datasheet.

The DRV8825 has a low-ESR ceramic capacitor, which makes it vulnerable to voltage spikes. It is required to use at least a 47μF capacitor across the motor power supply pins. It is common for controlling the NEMA stepper motors.

Search and add Libraries to EAGLE:

EAGLE libraries include symbols, footprints, and 3D models of electronic components. EAGLE includes various libraries. However, some components might not be found directly in the software.They must be downloaded and added into the Library Manager.

- To find the DRV8825, a google drive file was shared on the Texas Instruments Website. Download the following link to gain access to the library https://docs.google.com/file/d/0B_7eNPf3uK72ZlNsemQ4NkxfMlU/edit

- The following link provides many sparkfun libraries. However, only the resistors and capacitors libraries were used. https://github.com/sparkfun/SparkFun-Eagle-Libraries

To add the libraries, click on the library tab in the EAGLE software, then click on Open Library Manager.

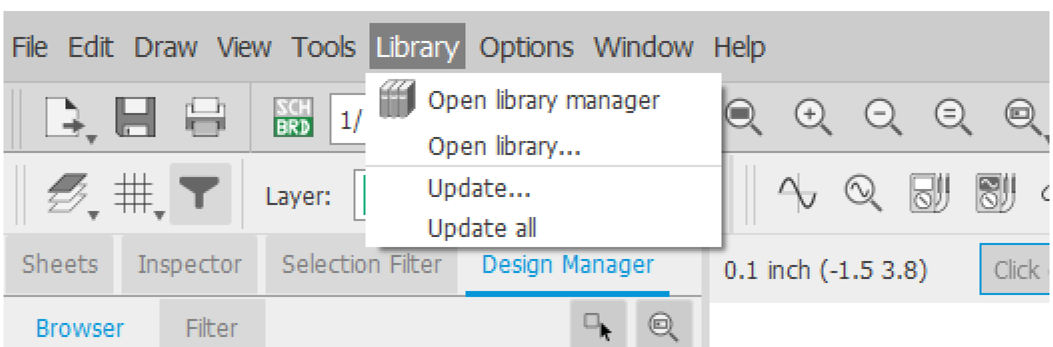


Figure 16: Open library manager

Click on the “Available” tab, then click browse. Add the libraries downloaded.

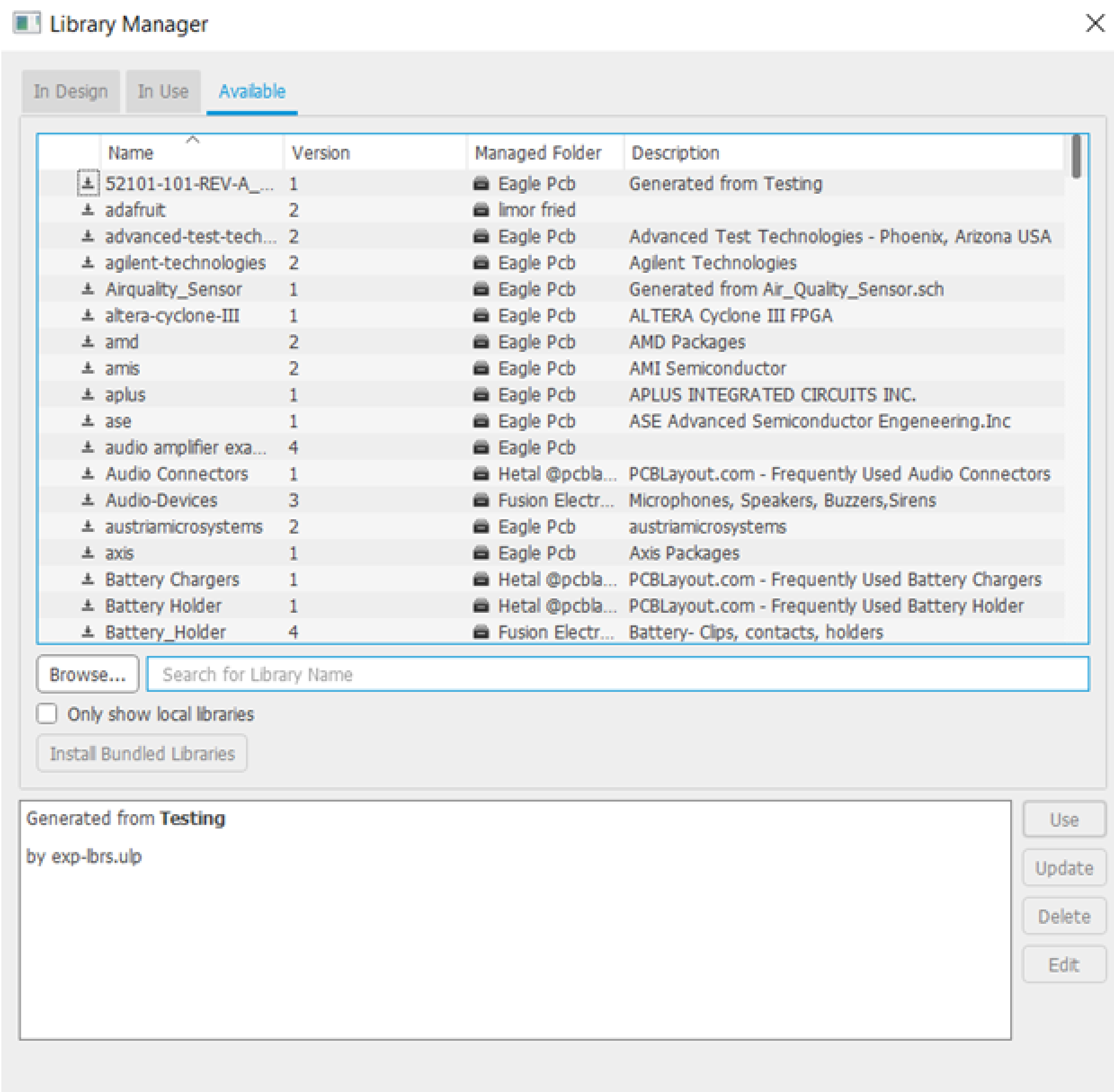


Figure 17: Library manager

To use the libraries added, search their name in the "Available" tab where it states, "Search for Library Name". Once found, press "Use". Also, search for the following libraries that are already in the Library Manager "Available" tab and click again on "Use" for each library:

- Terminal Blocks (Hetal @pcblayout)

– Resistor_Capacitor (Hetal @pcblayout)

– Potentiometer_Trimmers (Hetal @pcblayout)

– Headers (Hetal @pcblayout)

Build the schematic:

In EAGLE, press File,New, Schematic.After the libraries have been included, press on “Add Part”  to add Components.

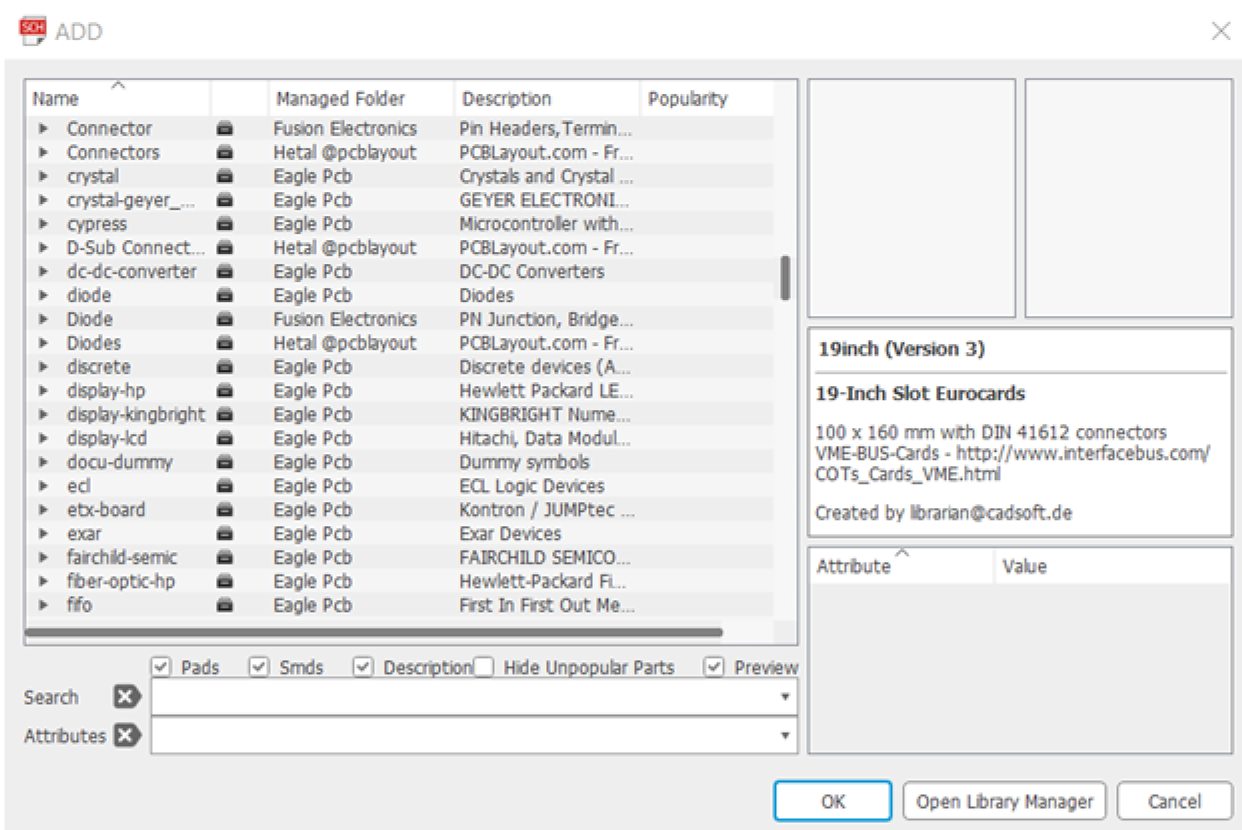


Figure 18: Add components

When searching for components, the value of the component is not essential. The most important, is for the component to provide a footprint. For example, in the figure below the 2.2uF capacitor does not show a footprint on the top right box.

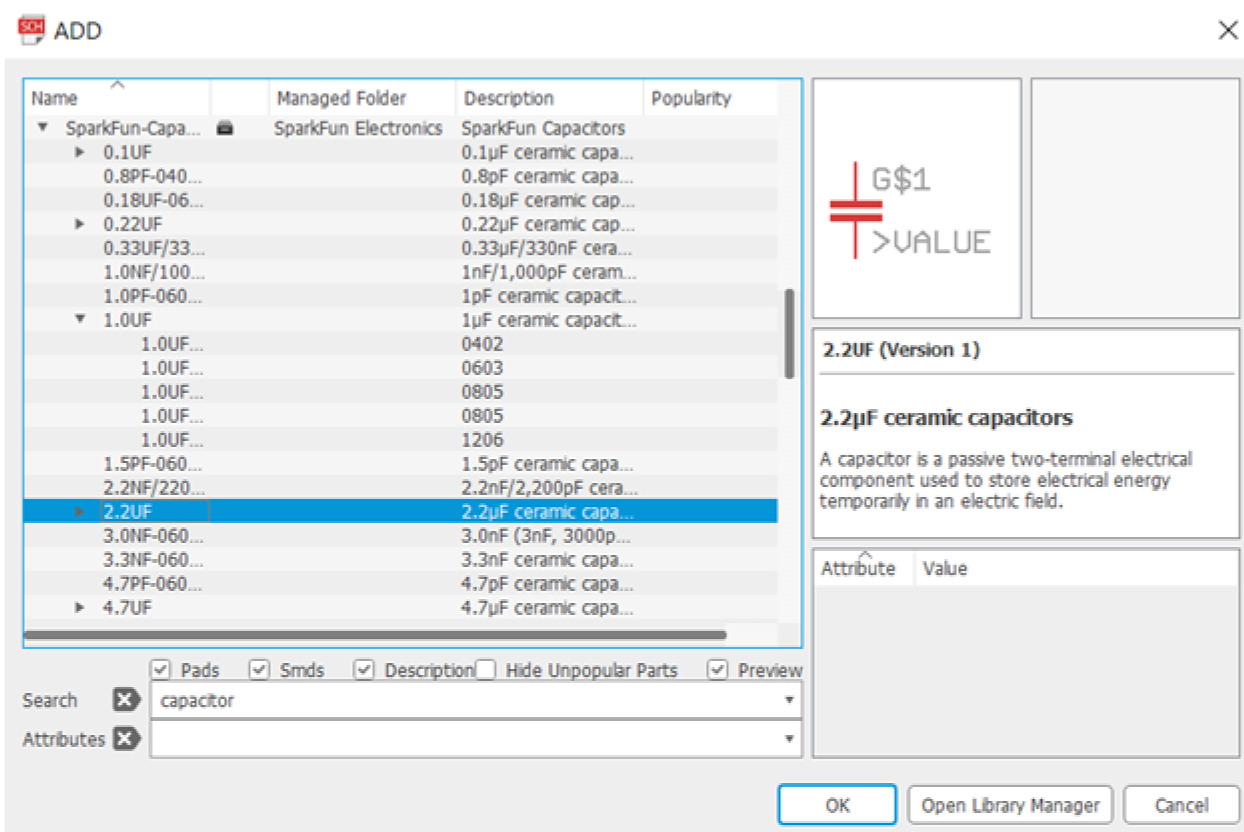


Figure 19: Add components

However, the 1uF capacitor in the figure below provides a footprint.

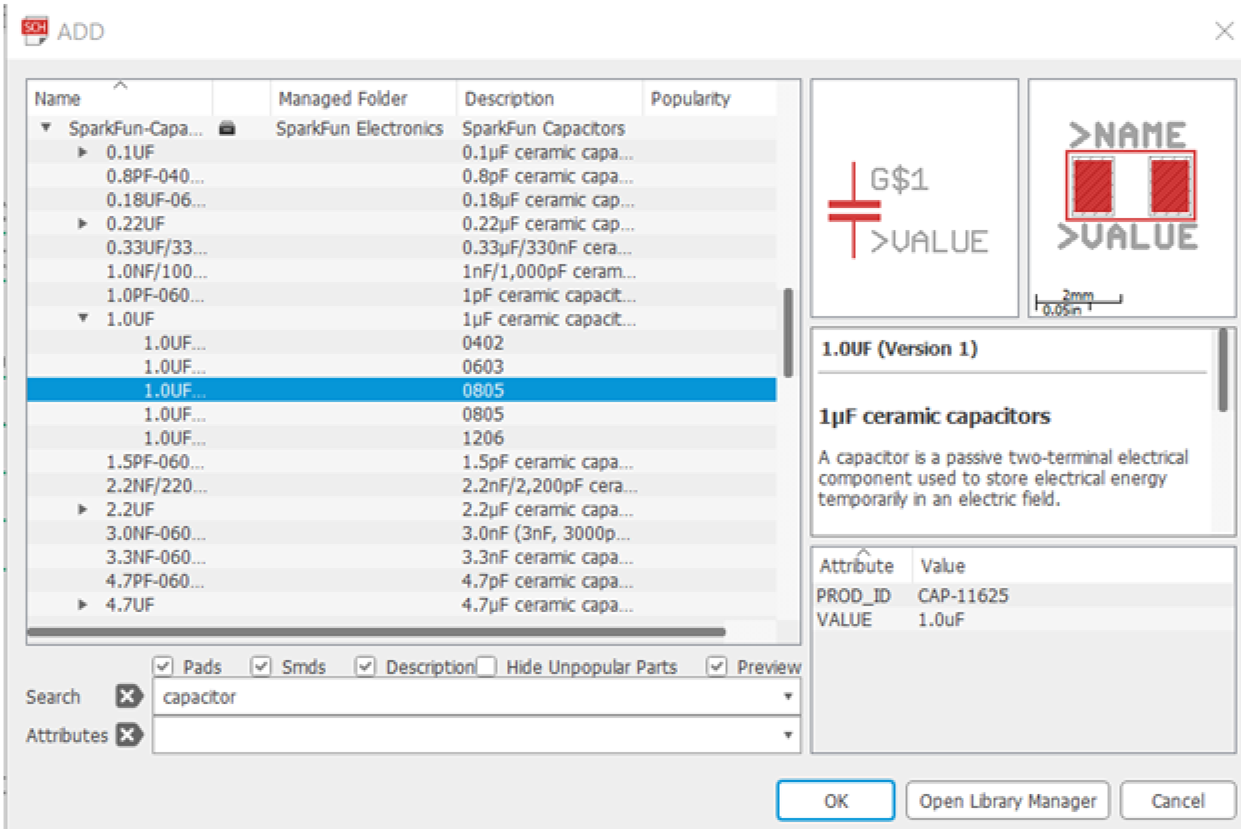


Figure 20: Add components

The footprint will be added when constructing the PCB layout and will also determine the package number of the component that will be ordered from Digi-Key Electronics to be soldered onto the printed circuit board. To obtain more knowledge of the component utilized in the schematic, right-click on one of the parts presented in the Design Manager. A list will show different aspects of the component to be accessed.

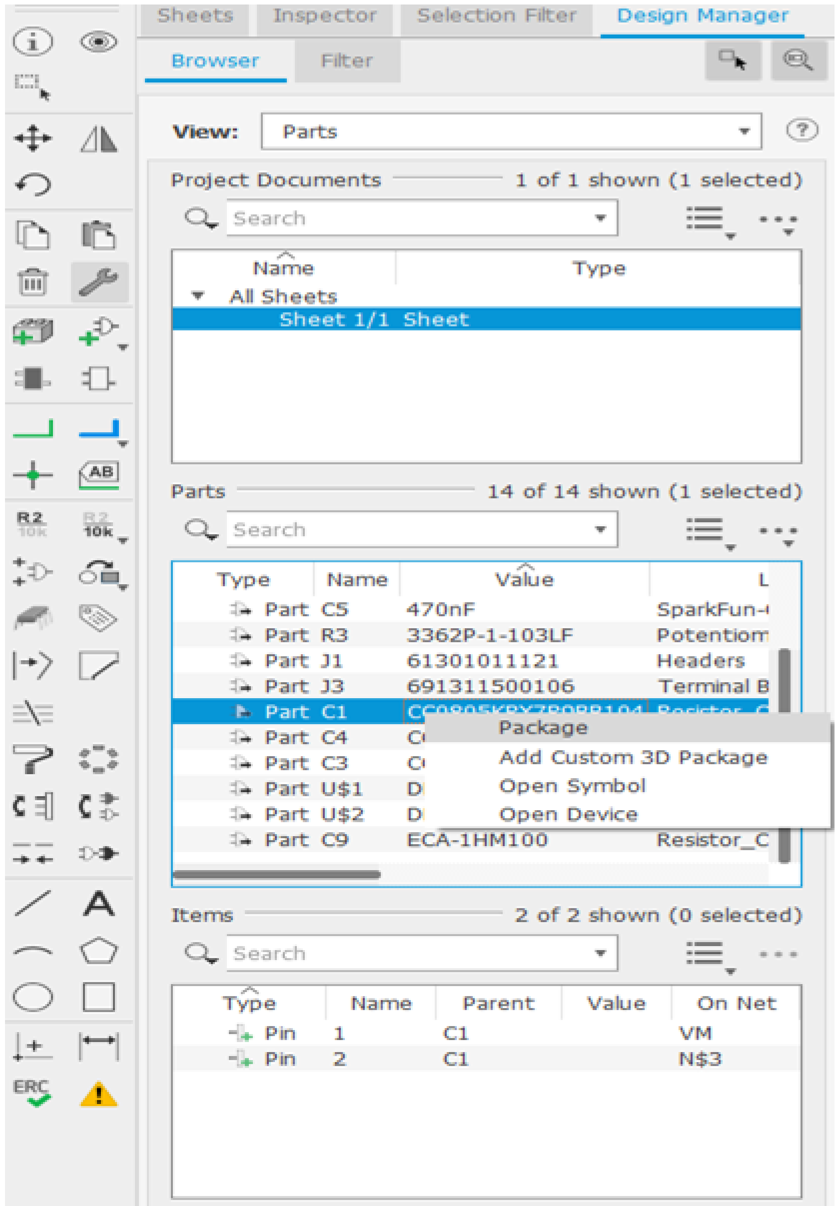


Figure 21: Design manager

“Open Device” contains Digikey part number (if available) and package number. If part number is available, simply copy and paste it into Digi-Key electronics website.

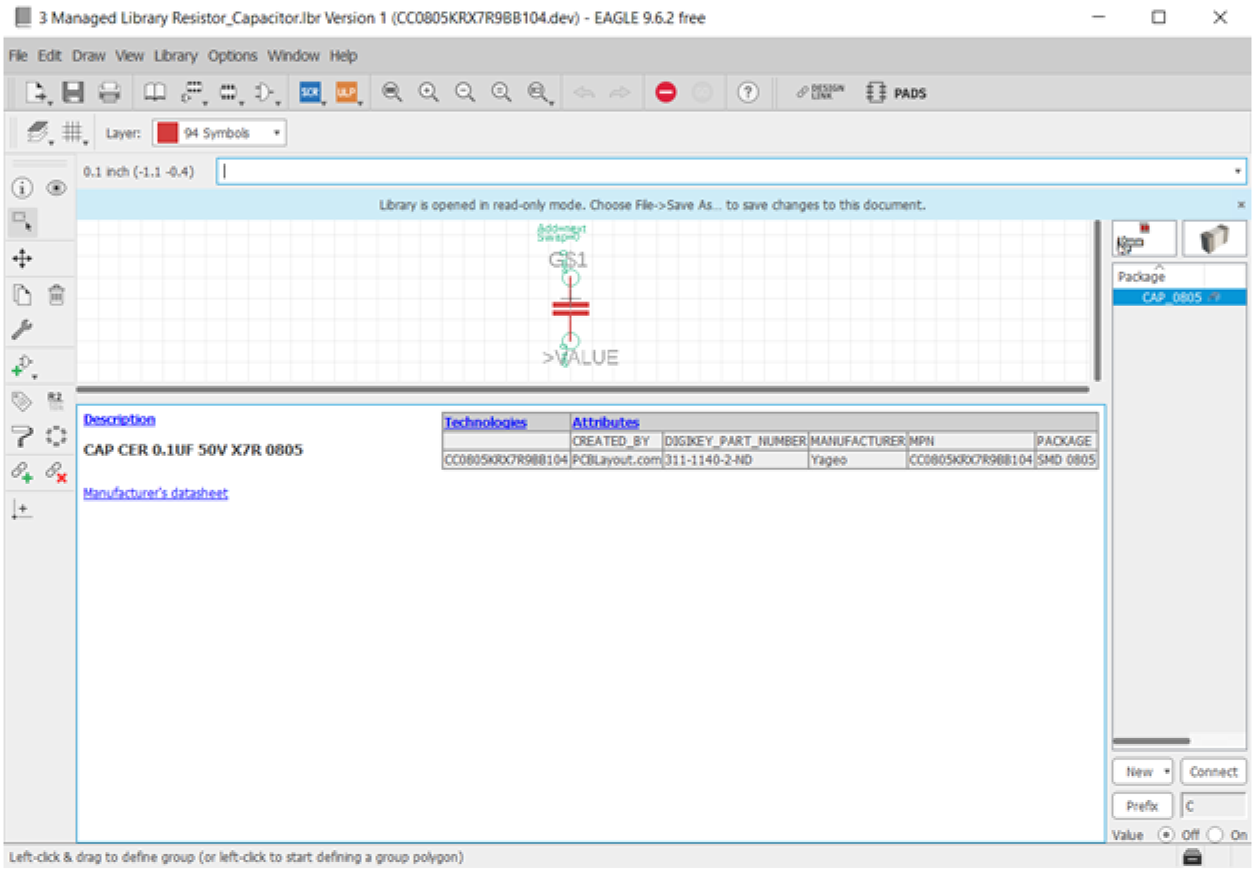


Figure 22: Open device tab

Press on "Net"  to connect the components. Press on "Name" to assign labels to connections like GND or VM …

By following the datasheet Pin Configurations and Functions, start Connecting the Components.

1) Add the DRV8825 to the schematic.

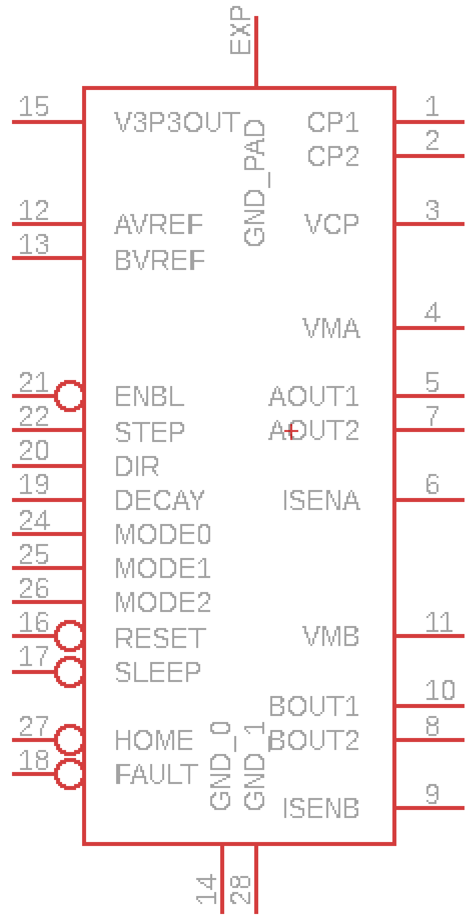


Figure 23: DRV8825 footprint

2) Pin 1 and 2: Connect a 0.01-µF 50-V capacitor between CP1 and CP2

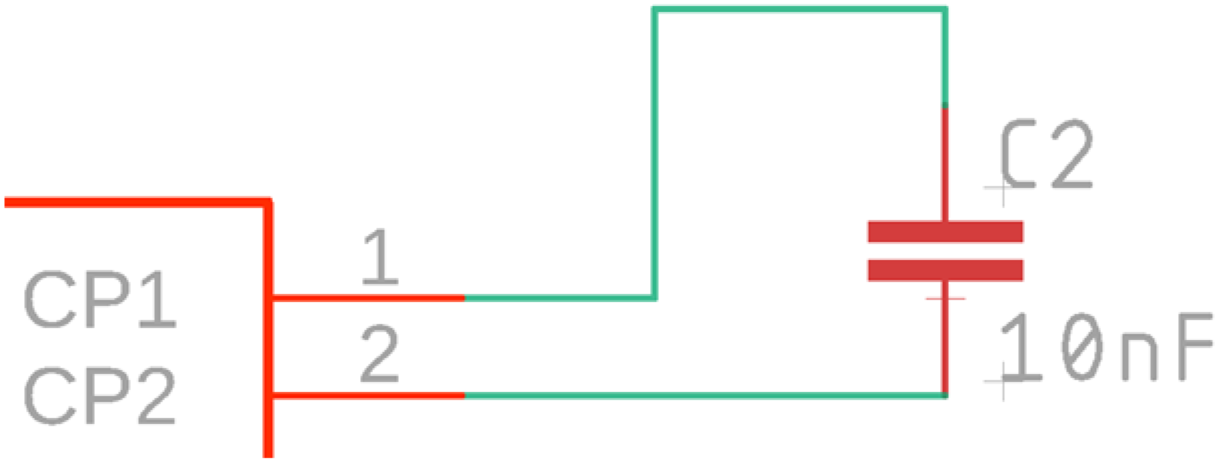


Figure 24: Pin1, 2 connections

3) Pin3: Connect a 0.1-µF 16-V ceramic capacitor and a 1-MΩ resistor to VM.

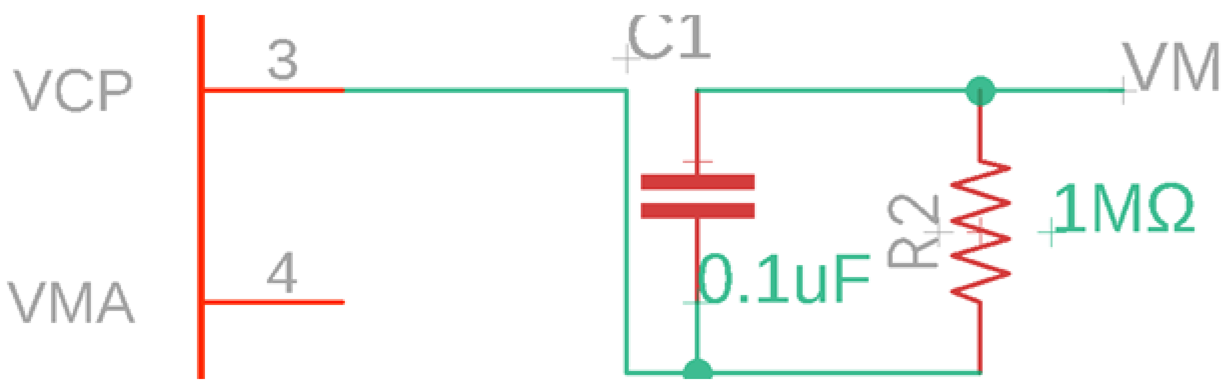


Figure 25: Pin3 connections

4) Pin4 and 11: Connect to motor supply (8.2 to 45 V). Both pins must be connected to the same supply, bypassed with a 0.1-µF capacitor to GND.

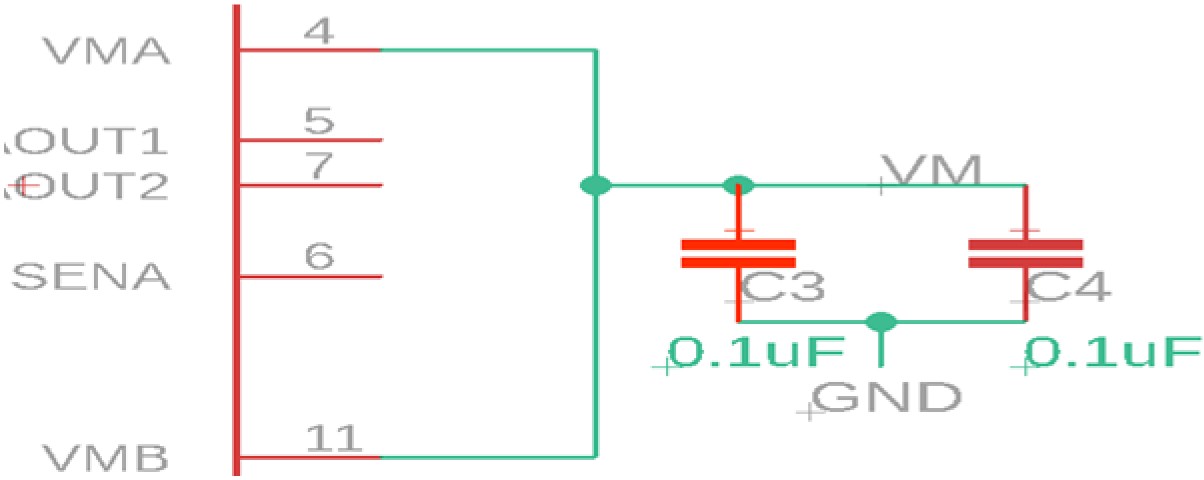


Figure 26: Pin4,11 connections

5) Pin15: Bypass to GND with a 0.47-µF 6.3-V ceramic capacitor.

Figure 27: Pin15 connections

6) Pin12 and 13: connect to a 10kΩ trimmer potentiometer.

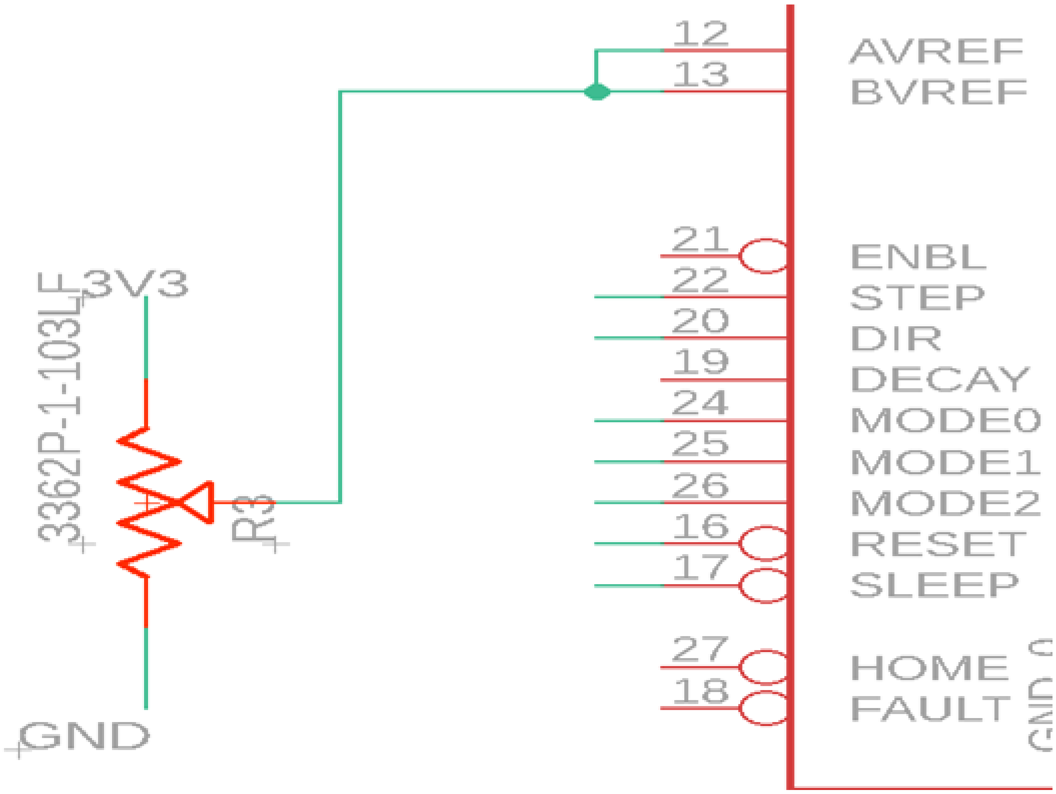


Figure 28: Pin12,13 connections

7) Pin6 and 9: connect to current sense resistor for Bridge A and B.

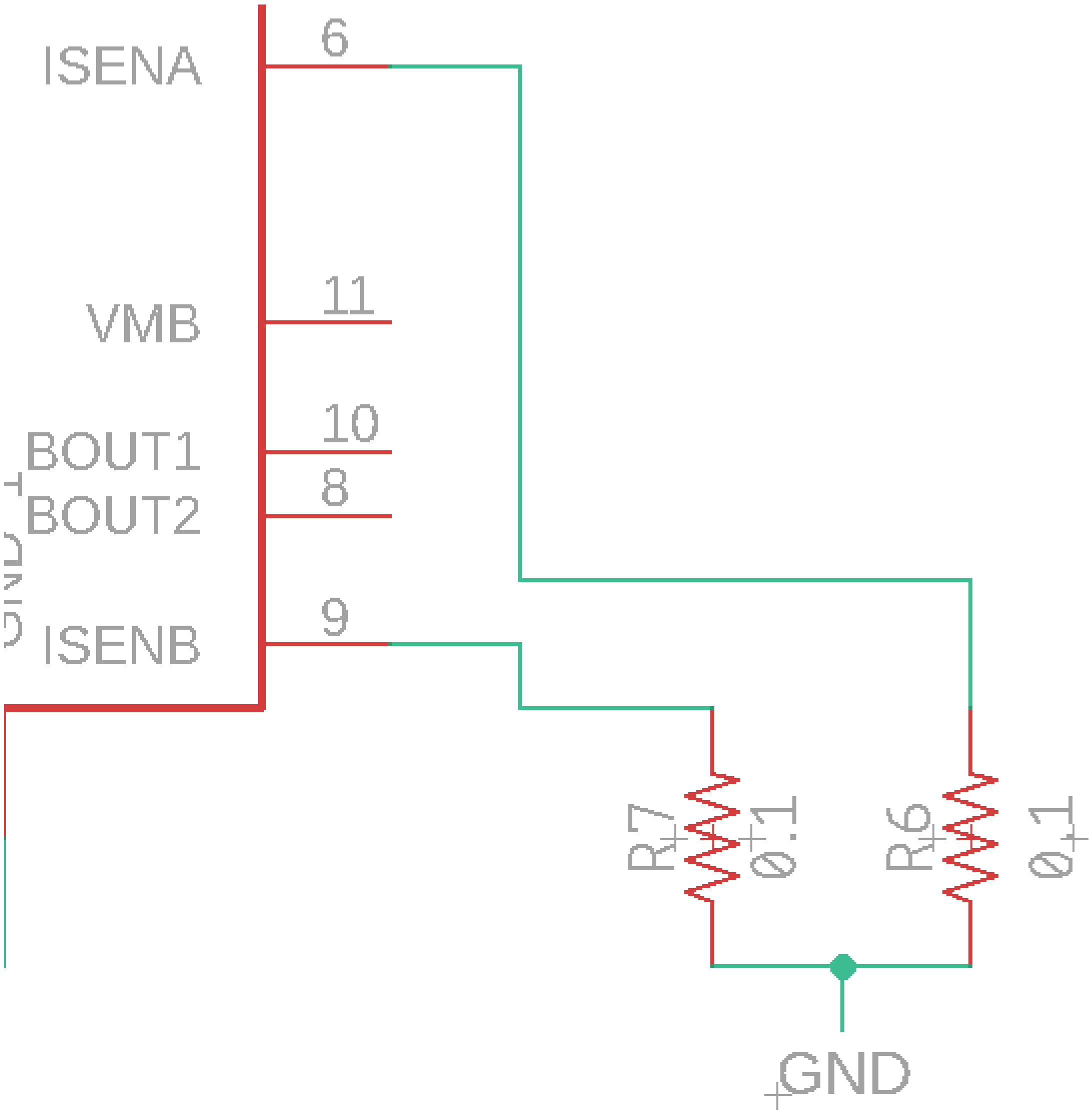


Figure 29: Pin6,9 connections

8) Pin 14, 28, exp: connect to ground

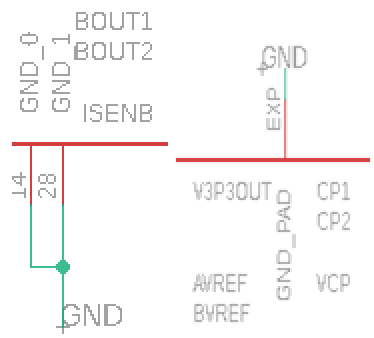


Figure 30: Pin 14,28,EXP connections

9)

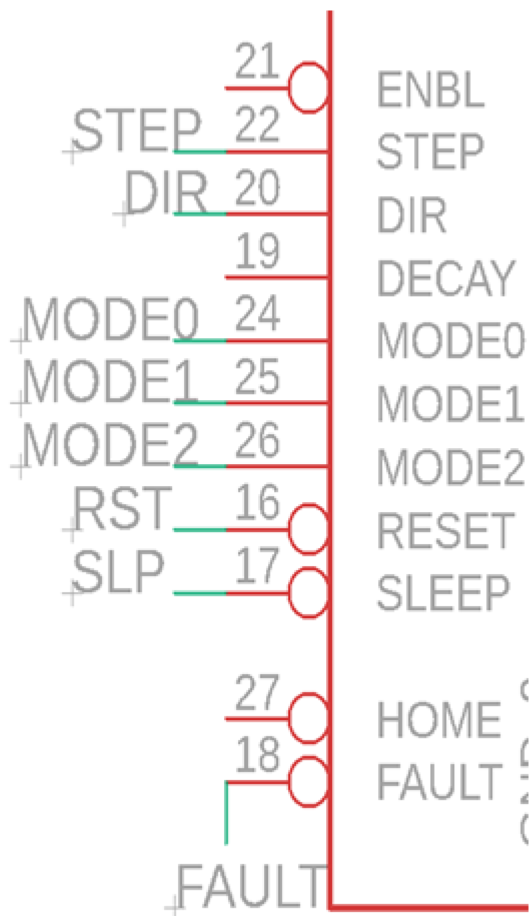


Figure 31: Pins connections

Pin 21: Logic high to disable device outputs and indexer operation, logic low to enable. Internal pulldown. This pin can be left disconnected, its default state is to enable the driver.

Pin 22: rising edge causes the indexer to move on step. Internal pulldown.

Pin 20: level sets the direction of stepping. Internal pulldown.

Pin 19: decay mode. Left disconnected. It is not used for this application.

Pin 24, 25, 26: MODE0 through MODE2 set the step mode – full, ½, ¼, 1/8, 1/16, or 1/32 step. Internal pulldown.

Pin 16: Active-low reset input initializes the indexer logic and disables the H-bridge output. Internal pulldown. RESET pin must be high to enable the driver.

Pin 17: logic high to enable device, logic low to enter low-power sleep mode. Internal pulldown. SLEEP pin must be high to enable the driver.

Pin 27: logic low when at home state of step table (left disconnected for this application).

Pin 18: logic low when in fault condition (overtemp, overcurrent)

The pins are connected to a connection header ,vertical, 10 positions.

10)

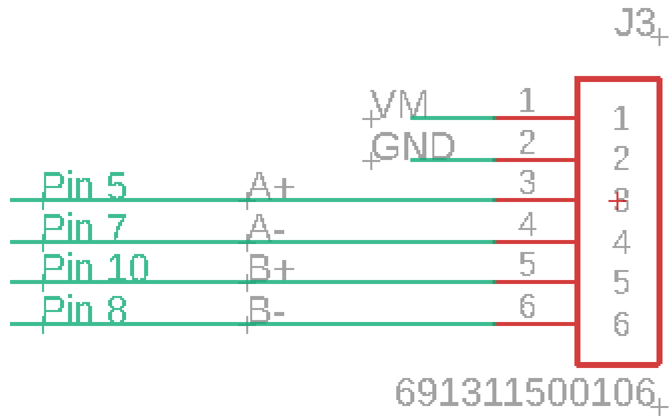


Figure 32: Stepper motor connections

Pin 5,7: connect to bipolar stepper motor winding A. The positive current is AOUT1- AOUT2.

Pin 10,8: connect to bipolar stepper motor winding B. The positive current is BOUT1-BOUT2.

The pins are connected to a terminal block header, vertical, 6 positions.

The completed schematic is shown in figure below:

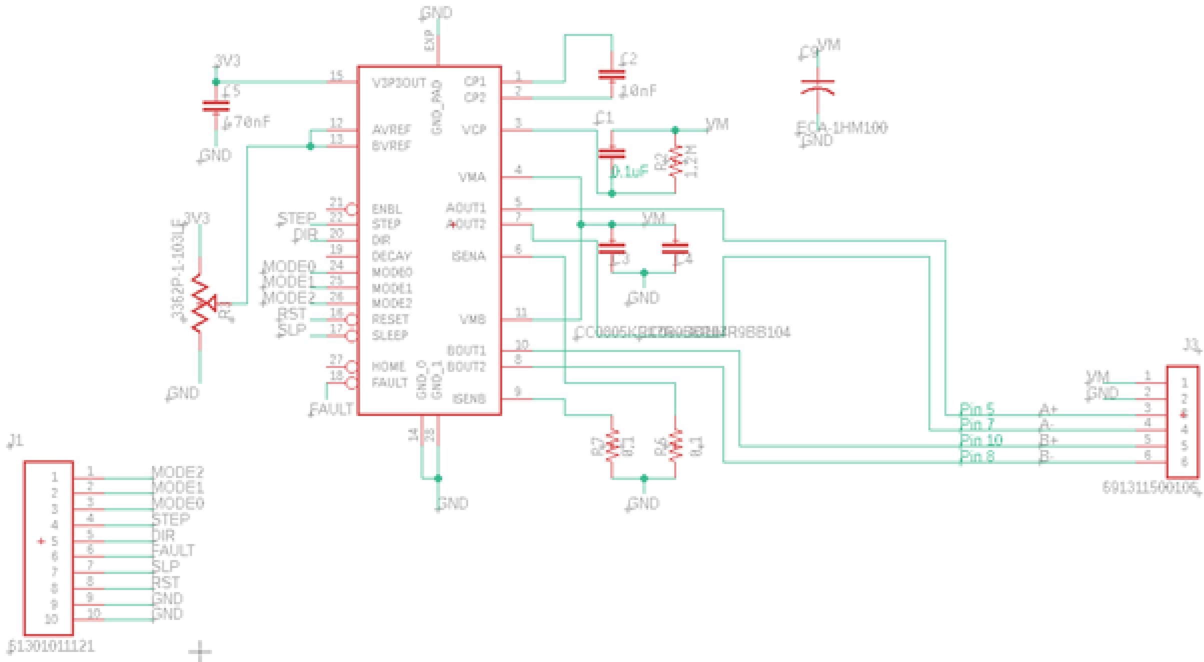


Figure 33: Overall schematic

After the schematic is finished, click on Tools, ERC (Electrical Rule Check) function, to check for warnings. Some warnings can be approved if intentional.

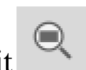
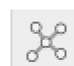
Build the Board Layout:

Generate the board by clicking on 

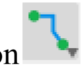
at the top left. The components and connections will be shown on the left side of the origin. Place the following components as close as possible to their corresponding pins:

- The capacitor between VMA and VMB pins with a thick trace.

- The capacitor between CPL and CPH pins.
- The capacitor between VMA and VCP pins.
- The capacitor to bypass V3P3 to ground.

Move the components to the board and set them up neatly around the origin  . In general, it is best to place the components like how they appear in the schematic.Once the components are placed, move the sides of the board closer to the components leaving the side where the origin is, untouched. Press Zoom to Fit  to get a better view. Change the grid size  if a finer control over placing the components is needed. Press Edit, Net classes to change the width to 12mil, Drill to 20mil and Clearance to 10mil as standard sizes. Press ratsnest  to recompute airwires after components have been moved.

Grounding Top and Bottom Layer:

To make the board a double sided PCB with two ground layers, set the Layer to Top, press the polygon  , press on the origin sign and follow the sides of the board back to the starting point. Press again to be able to name it. The red line will show now as red dots. Name the connection ground “GND”. Repeat the same action to the bottom layer also naming it “GND”. No routing is needed for Ground connections since the top and bottom layers are grounded. Notice the airwires connection to Ground disappeared. A signal can transition from one layer to another using via, which is a conducting hole, like a jumper. Vias are used when there is no longer space to make a connection on a certain layer. A transition to another layer is possible to accomplish the connection. Start routing by pressing on  , which will turn airwires into a trace. The routing should be organized. Different options are available when routing like selecting the width, layer, and type of bend from the menu bar.

Press Tools, DRC (Design Rule Check) and click Check to check for errors or warnings in the design (presence of airwires or wire stubs…) If an airwire warning shows, it is stating that a trace is blocking the path of another. Try to modify the routing for a better path. Also, use vias to move from one layer to another. Note that grounding the Top and Bottom layer will increase the risk of errors when soldering the board specially for beginners. If there was excess solder, it can lead to potential short circuits.

Routing Ground Connections:

An alternative option is to route the ground connections the same way the other routing was made, precise and neat. This will not cause major complications while soldering the board.

Organize the Board:

Move the names of the components. From the layer menu  at the top left corner, double-click on the following names to change their color to white. “tPlace”, “bPlace”, “tNames” and “bNames”.

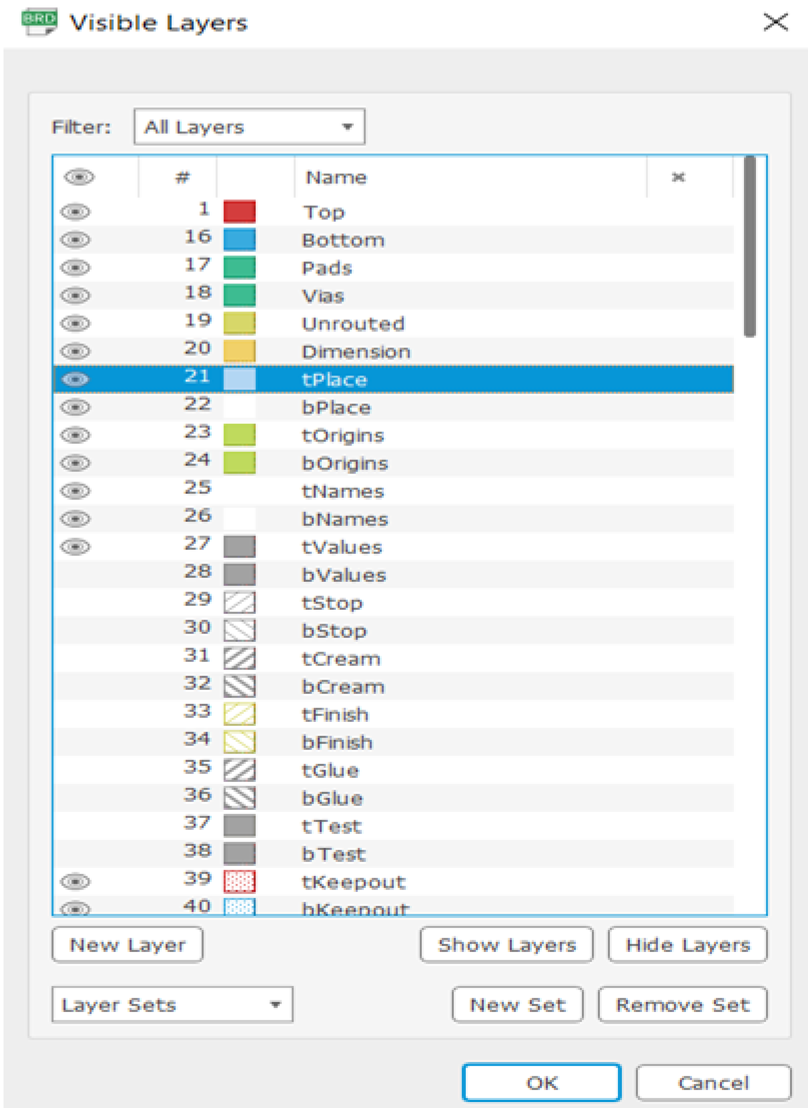


Figure 34: Visible layers

From the layer menu, change the color of "tDocu" and "bDocu" to a bold color like yellow for example.

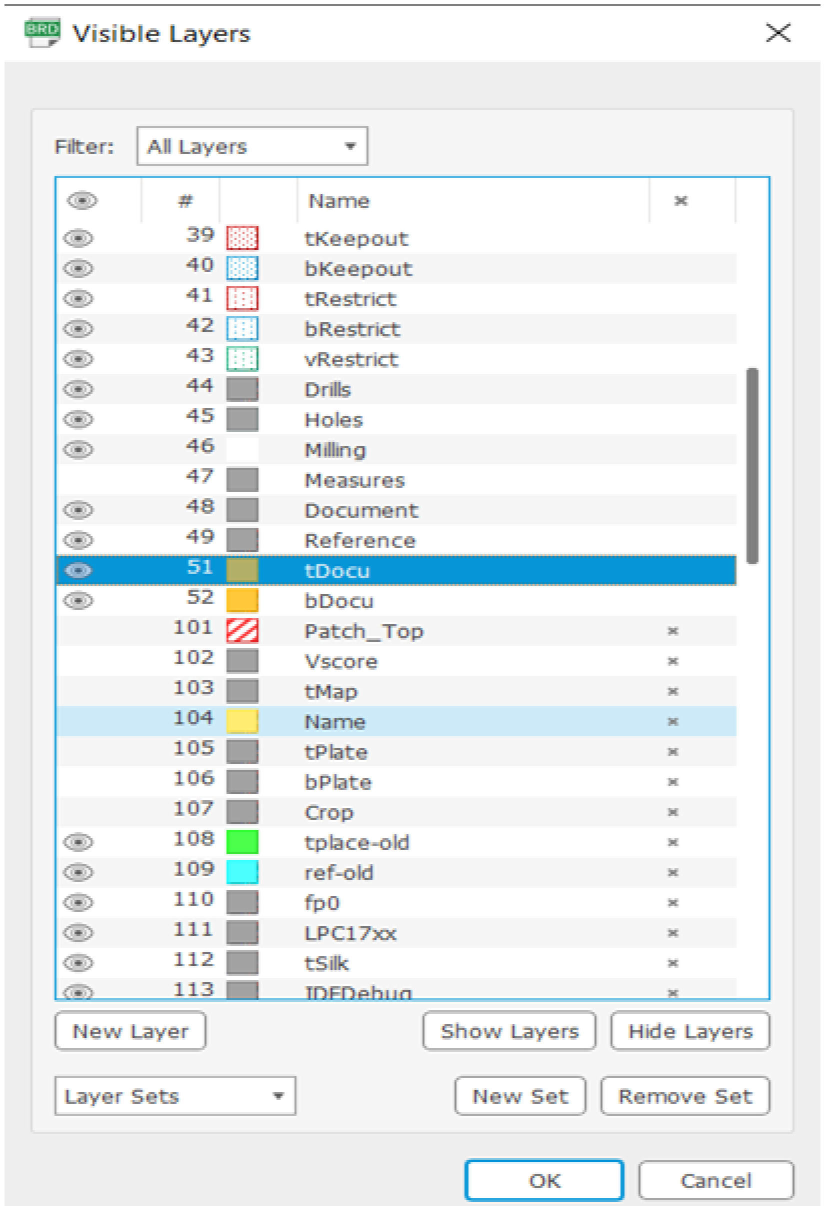


Figure 35: Visible layers

Give the board a name. Click on  and type the name of the board and choose tPlace for layer to place the text. The font can be changed to a vector. Place the text on the board.

The completed PCB layout is shown below:

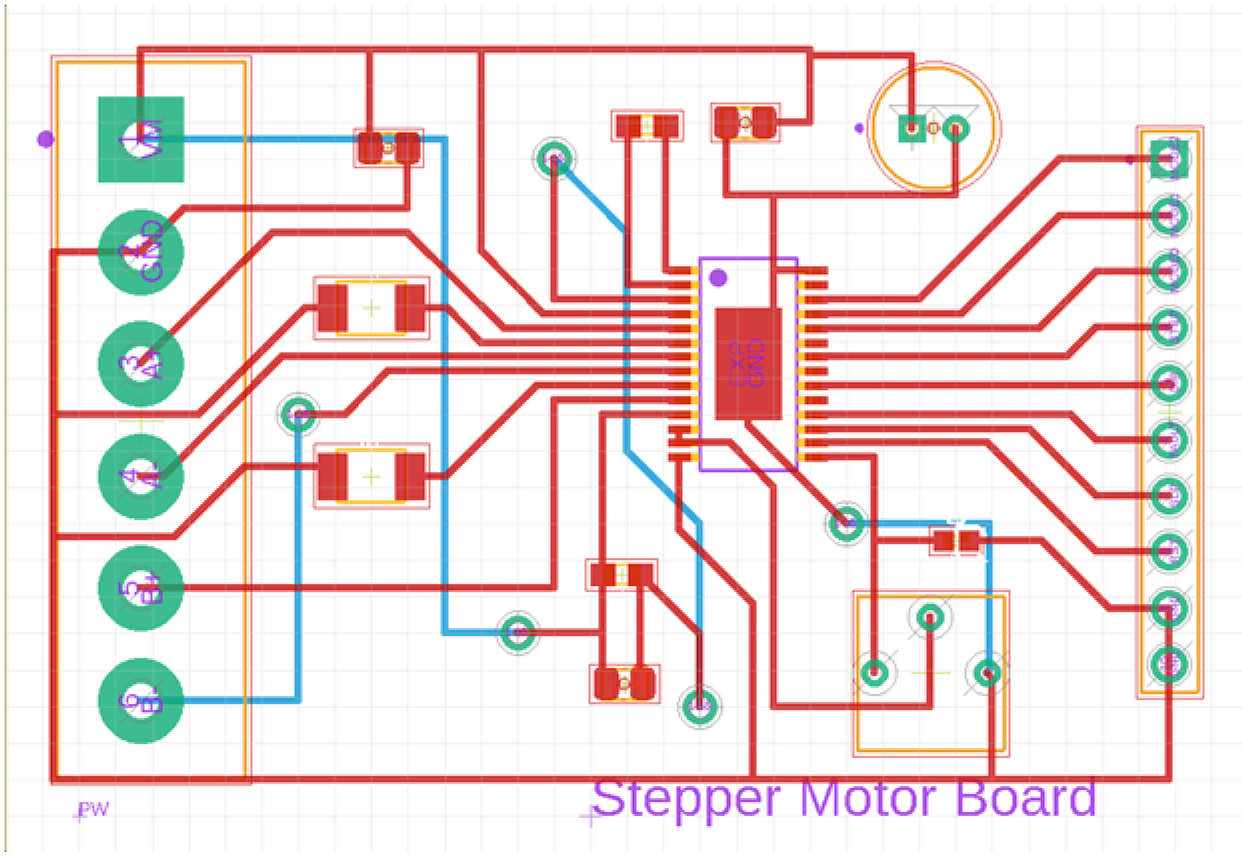


Figure 36: Completed PCB layout

Generate Gerber Files from EAGLE:

When the design is finished, the last step before the fabrication is to generate Gerber files. EAGLE includes a computer-aided manufacturing (CAM) processor that will allow it to load a CAM file and generate the specific files for the design. Click on the CAM processor at the top  . The PCB device at Worcester Polytechnic Institute accepts Output type Gerber X2. The CAM job selected is template_2_layer.cam which is the default CAM job provided by the software.

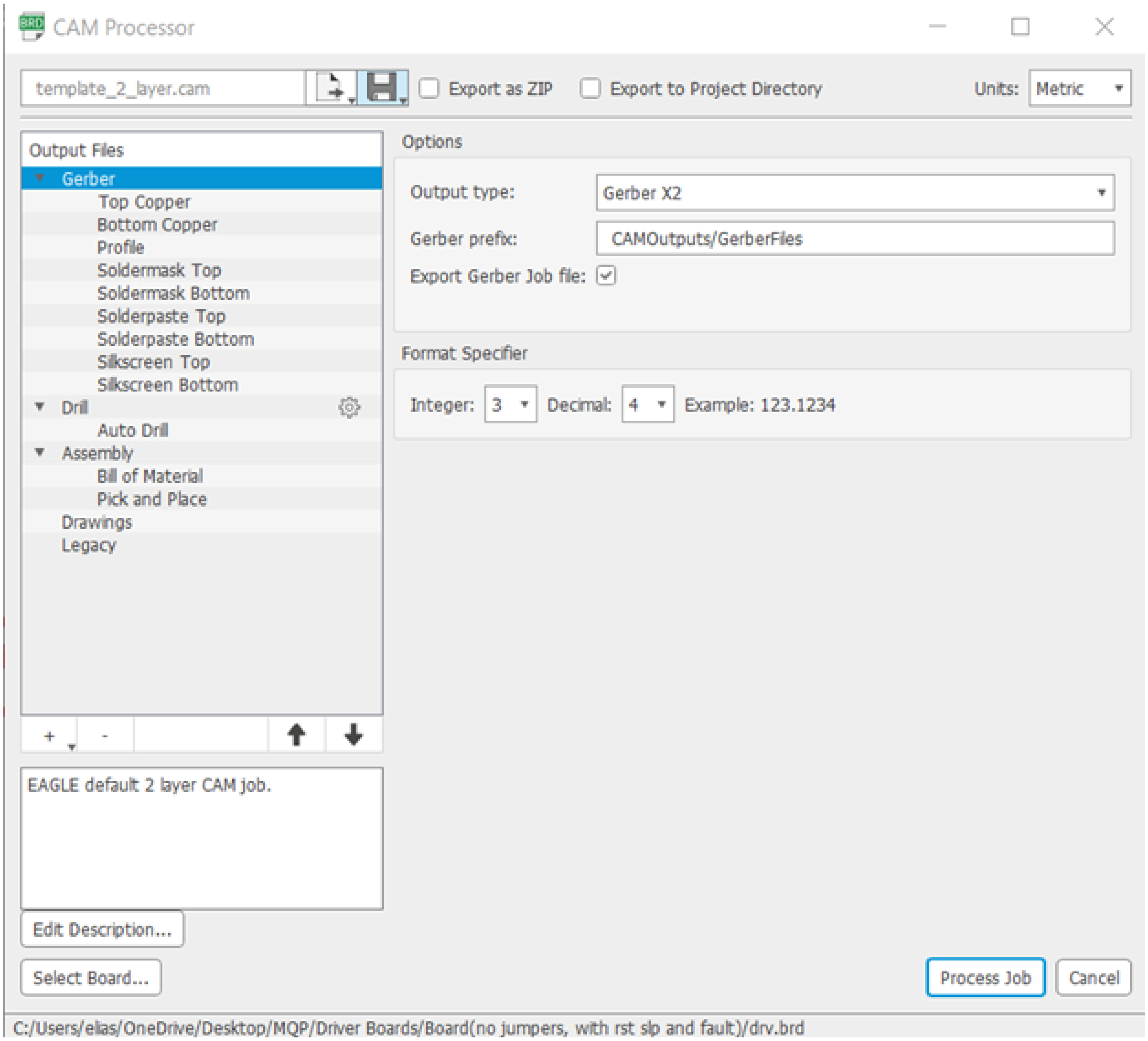


Figure 37: CAM processor

To the left, the output files are shown which will provide all the combination of layers to a device. Click on "Process Job" and the file should be saved in the EAGLE file for the design as "CAMOutputs". The file is then exported as a ZIP file and sent to be fabricated.

Fabricated Stepper Driver Boards:

Grounded top and bottom layers:

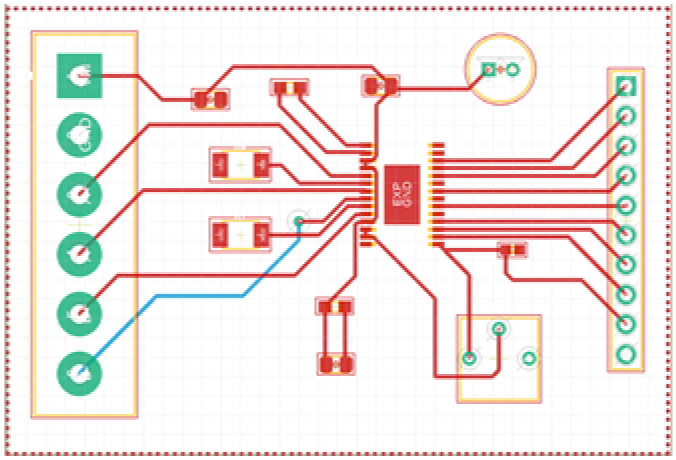


Figure 38: The first design with top and bottom ground layers

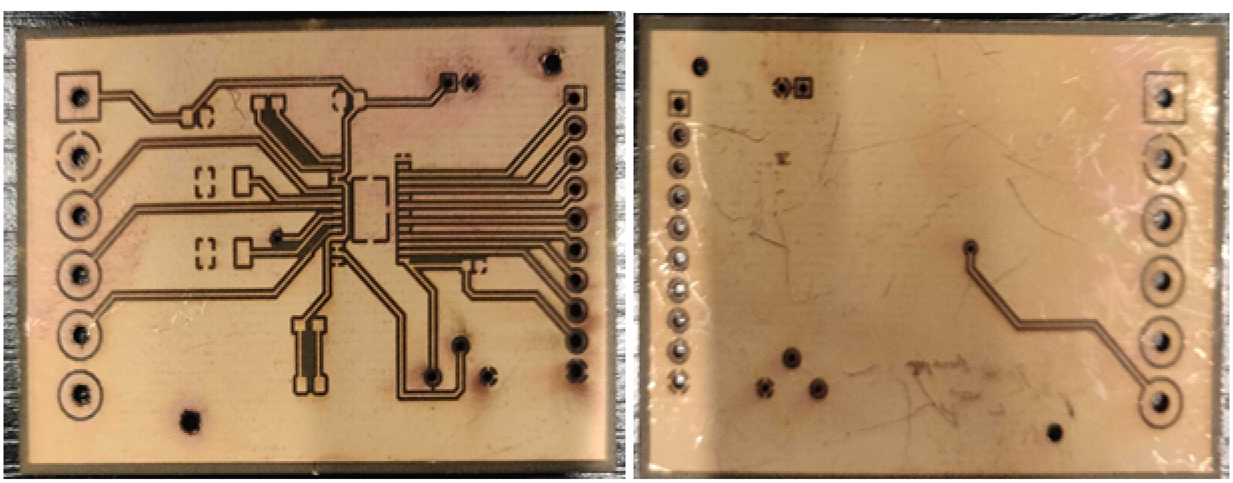


Figure 39: The first PCB design with top and bottom ground layers

Routed Ground Connections:

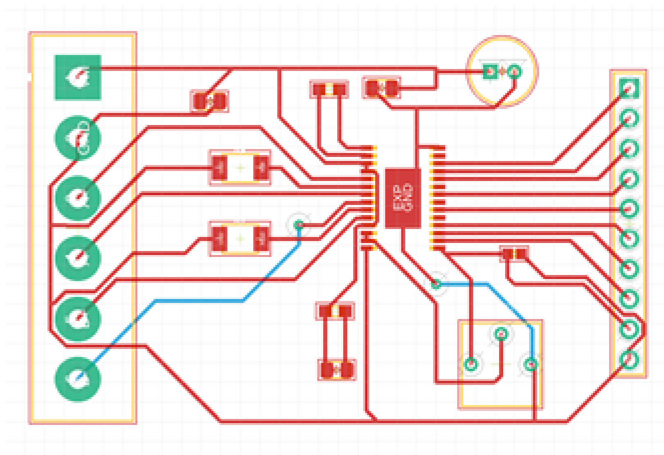


Figure 40: The second design with routed ground connections

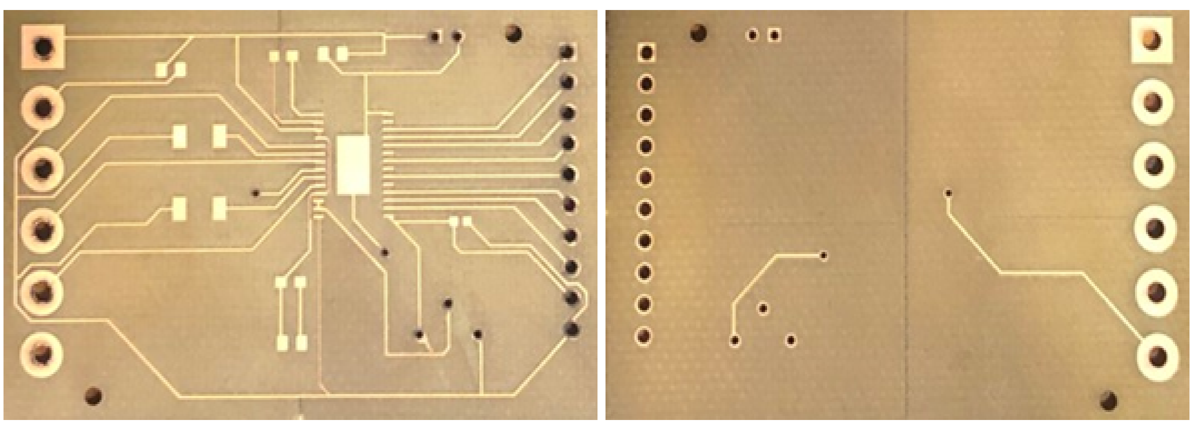


Figure 41: The second PCB design with routed ground connections

We concluded that the second device is more efficient with less percentage of risks when soldering.

How Stepper Motors work:

Stepper motors are DC motors. They have multiple coils arranged in groups called “phases”. When a high pulse is emitted, one phase energizes and attracts the cogged wheel which will drive the motor one step, without any position sensor for feedback. By energizing each phase in sequence, the motor will rotate one step at a time. The direction of the motor is determined by the sequence of the pulses. The speed of the motor is determined by the frequency of the pulses. The motor will run depending on the number of pulses. Stepper motors are used in many precision motion control applications due to their very precise positioning and speed control.

Microstepping:

Microstepping is when it is possible to move the motor shaft into positions between steps. For example, if we apply current to both coils equally, the motor shaft will lock halfway between the two coils. The same principle includes 1/4 steps, 1/8 steps, 1/16 steps, and 1/32 steps to allow extremely precise positioning.

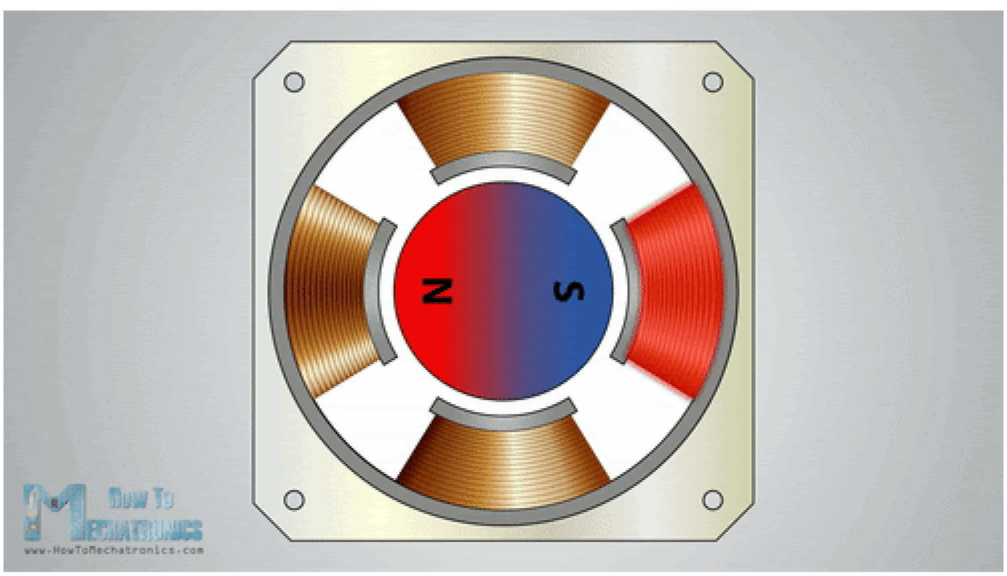


Figure 42: Motor shaft moving in discrete steps

https://thumbs.gfycat.com/CompetentWarmheartedBangeltiger-size_restricted.gif

Bipolar Stepper motor:

Bipolar stepper motors are a type of stepper motor with a single winding per phase and no center tap. Each stator coil winding has two terminals. The magnetic field on each stator must change for the stator shaft to rotate. A two-phase bipolar stepper motor will have four leads. There is no natural reversal of the current direction. Therefore, the use of a driver IC with an internal H bridge circuit will reverse the polarity of stator poles. The IC is capable of handling the high current drawn by the stepper motor. The H bridge is necessary to handle the spike when the coil current changes direction. Both are needed to protect the Microcontroller.

Testing the Stepper Driver Board:

The bipolar stepper motor we used is NEMA 17, part number 17HS16-2004S1. Its electrical specification is shown below.

SPECIFICATION / CONNECTION	BIPOLAR
VOTAGE(VOC)	2.20
AMPS/PHASE	2.00
RESISTANCE/PHASE(Ohms)@25°C	1.10±10%
INDUCTANCE/PHASE(mH)@1KHz	2.60±20%
HOLDING TORQUE(Nm)[lb-in]	0.45[3.98]
STEP ANGLE(°)	1.80
STEP ACCURACY(NON-ACCUM)	±5.00%
ROTOR INERTIA(g-cm²)	54.00
WEIGHT(Kg)[lb]	0.30[0.66]
TEMPERATURE RISE:MAX.80°C（MOTOR STANDSTILL;FOR 2PHASE ENERGIZED）	
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]	
INSULATION RESISTANCE 100 Mohm（UNDER NORMAL TEMPERATURE AND HUMIDITY）	
INSULATION CLASS B 130°C[266°F]	
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE)	
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)	

Figure 43: Electrical specification from the 17HS16-2004S1 datasheet

https://www.oyostepper.com/images/upload/File/17HS16-2004S1.pdf

The stepper motor is with a 1.8 degree step. It has four wires and 1m cable and 2.54mm pitch connector. Each phase draws 2.0A with holding torque 45Ncm.

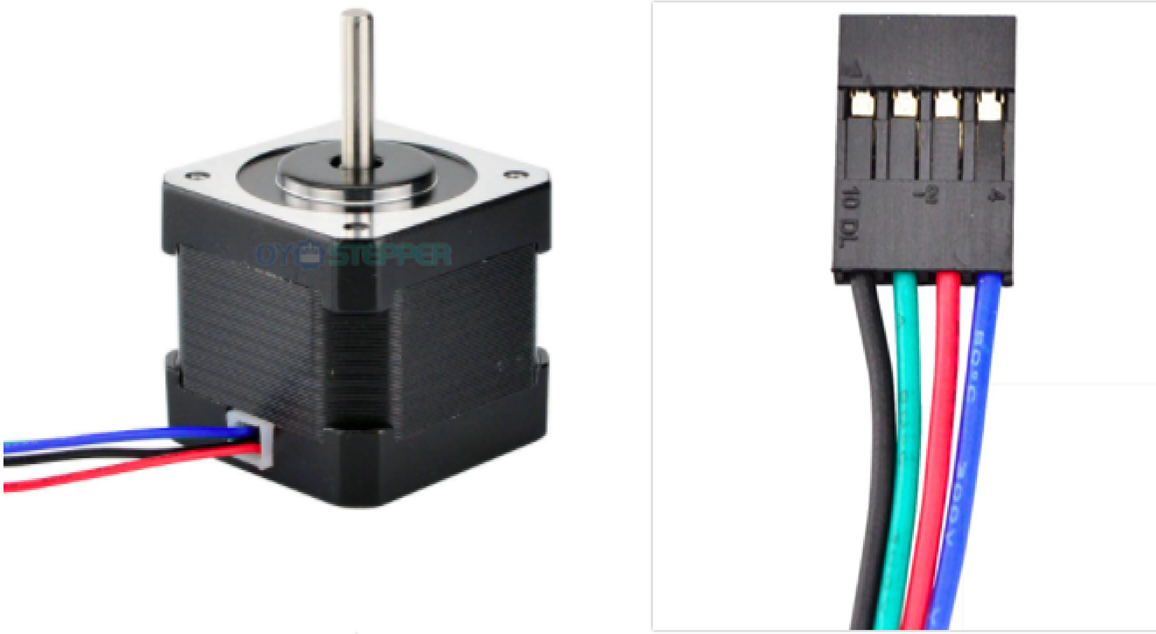


Figure 44: NEMA 17 (17HS16-2004S1)

https://www.oyostepper.com/goods-45-Nema-Size-17-Stepper-Motor-Bipolar-45Ncm-64ozin-2A-42x40mm-4-Wires-w-1m-Cable-Connector.html

Pin1	Pin2	Pin3	Pin4
A+	A-	B+	B-
Black	Green	Red	Blue

Figure 45: NEMA 17 (17HS16-2004S1) pinouts

The microcontroller that we used is Arduino Mega 2560.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Figure 46: Arduino Mega 2560's electrical specification

https://store.arduino.cc/usa/mega-2560-r3

The stepper motor driver we created has a maximum current rating of 2.5A per coil. The current sense resistors used further limit the maximum current to 2.2A. It is designed to draw heat out of the IC. If we were to supply more than 1.5A, there would be a heat sink. The stepper motor driver Specification is shown below.

Minimum operating voltage	8.2V
Maximum operating voltage	45V
Continuous current per phase	1.5A
Maximum current per phase	2.2A
Minimum logic voltage	2.5V
Maximum logic voltage	5.25V
Microstep resolution	Full,1/2,1/4,1/8,1/16 and 1/32
Reverse voltage protection	No

Figure 47: Stepper motor driver electrical specification

Analysis of the first fabricated Stepper Driver Board:

Once the stepper motor driver board was fabricated, we soldered the components needed.

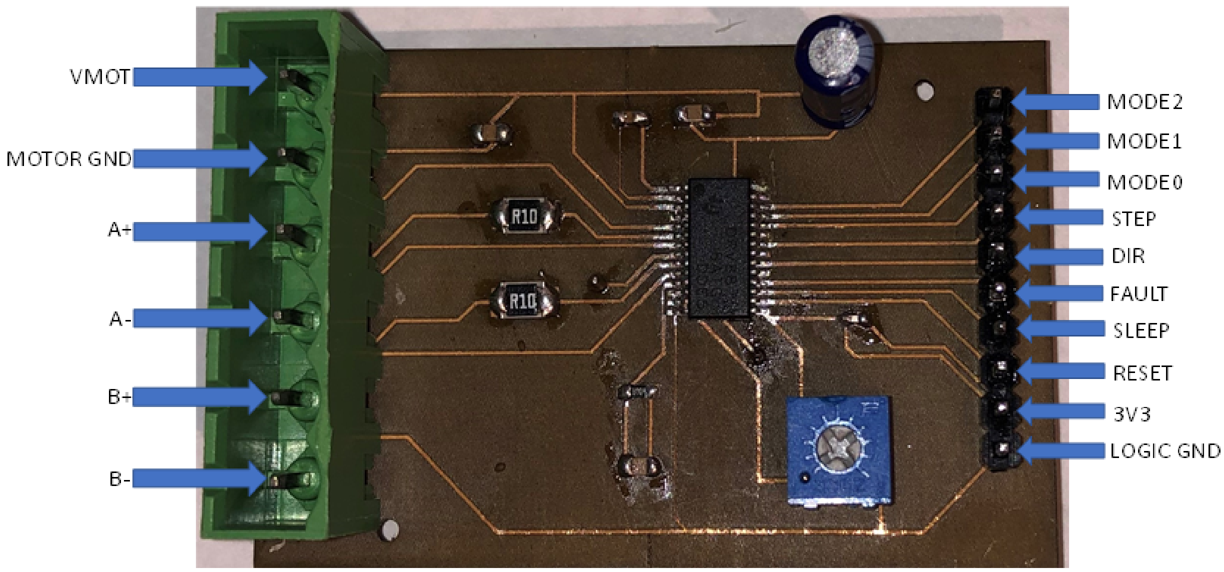


Figure 48: Stepper driver pinouts

We connected the motor power supply (12V 30A) to VMOT and motor GND. Following the pin connection of the motor coil previously stated, we connected A+, A-, B+, and B-.

MODE2, MODE1, and MODE0 are left disconnected at first, so the driver operates in full-step mode. The three selector inputs have internal 100kohm pull-down resistors. When the three modes are LOW, the microstep resolution is a full step.

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Figure 49: DRV8825 step mode

We connected the STEP pin to digital pin 3 and DIR pin to digital pin 2. We left the FAULT pin disconnected. The FAULT pin drives low whenever the H-bridge FETs are disabled as the result of over-current protection or thermal shutdown. We connected the RESET and SLEEP pin to the 5V of the controller.

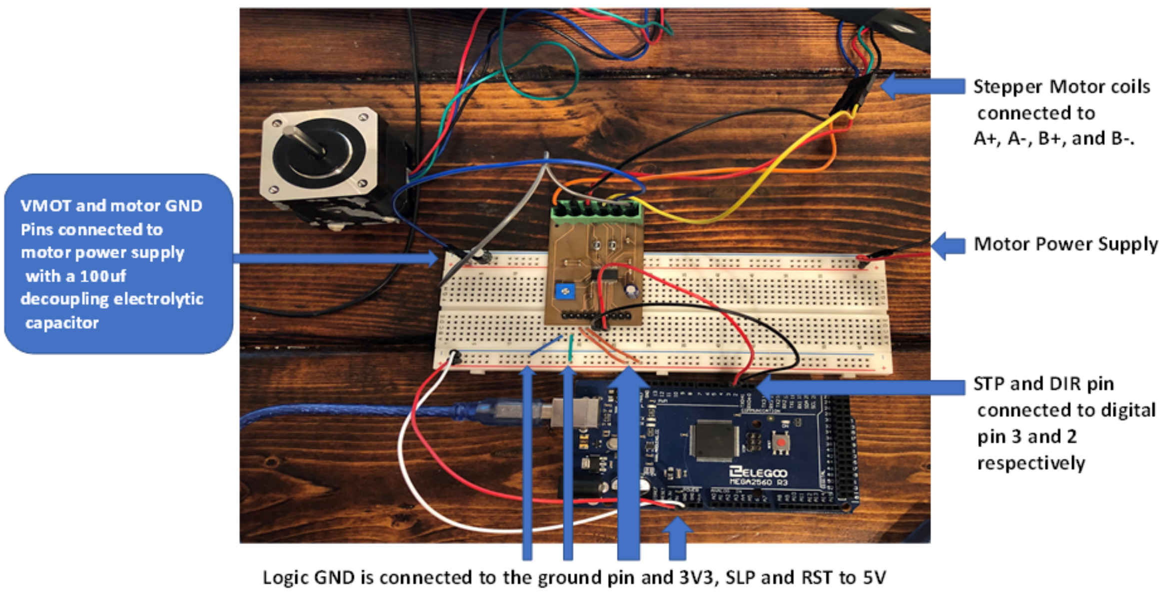


Figure 50: Circuit assembly

Analysis:

According to the IC datasheet, SLEEP pin is active low input. When this pin is LOW, the driver is in sleep mode. Therefore, it was set to HIGH. Also, the RESET pin is active low input. When LOW, all STEP inputs are ignored, until it is pulled to HIGH. Therefore, it was set to HIGH. ENABLE pin is active low input. The drive is enabled when pulled LOW. It is LOW by default, so it was left disconnected. On the board, we have included a pin for the 3V3 output connection which was connected to the microcontroller's 5V. The potentiometer is used to set the current limit. The first step to set the current limit is determining the amount of current we want to use to run the motor which indicates finding the rated current per phase of the motor. (the maximum amount of current that the motor was designed for). To get the full torque from the motor, the driver’s continuous current rating should be higher than the rating of the motor. The driver’s continuous current rating is 1.5A less than the rating of the NEMA 17 motor used is 2.00A. Therefore, we are limited to the driver. We can set the current limit lower than this if we want the system to run cooler and take less power avoiding any problems. We cannot set the current limit higher than the rating of your motor. That would damage the motor. The current limit equation found previously is Current Limit = VREF x 2. In our case, the current limit is equal to 1.5A. Consequently, VREF will be equal to 0.75V. First, the stepper motor is disconnected. We turn the power supply on. Note that disconnecting or connecting the stepper motor while it is powered could damage it or the driver. Knowing the calculated VREF, we can check what is the VREF voltage from the potentiometer. We can change that voltage to match the voltage required to get our desired current limit, using the multimeter. We turn the power off and we connect the stepper motor. We Load the program onto the controller and reconnect the power. The motor should step. To verify that setting VREF gave a correct current limit value, we disconnected the power again. We then reconnected the multimeter in series with one coil of the motor. It is difficult to measure the current through the stepper motor’s coils while it is stepping. The current is constantly changing. Therefore, we change to a program that will not move the motor. The amount of current flowing will be steady. However, the current on the multimeter will not read the number expected. This is because we have the driver set to operate in full-step mode. In full step mode, the driver only energizes both coils at about 70% of the current limit. The number will also slightly vary due to factors like averaging and tolerance of the components and meter.

Results:

When measuring the voltage across many components on the circuit board. The values conducted were fallacious. Plus, when modifying the value of the potentiometer. The voltage value remained. Various errors were established. Pin 15 (V3P3OUT) can be used to supply VREF. VREF is pin 12 and 13 connected to the potentiometer. It should not have a logic supply pin as the IC gets its power from the internal 3V3 voltage regulator. Therefore, 3V3 should not be used to power the project.

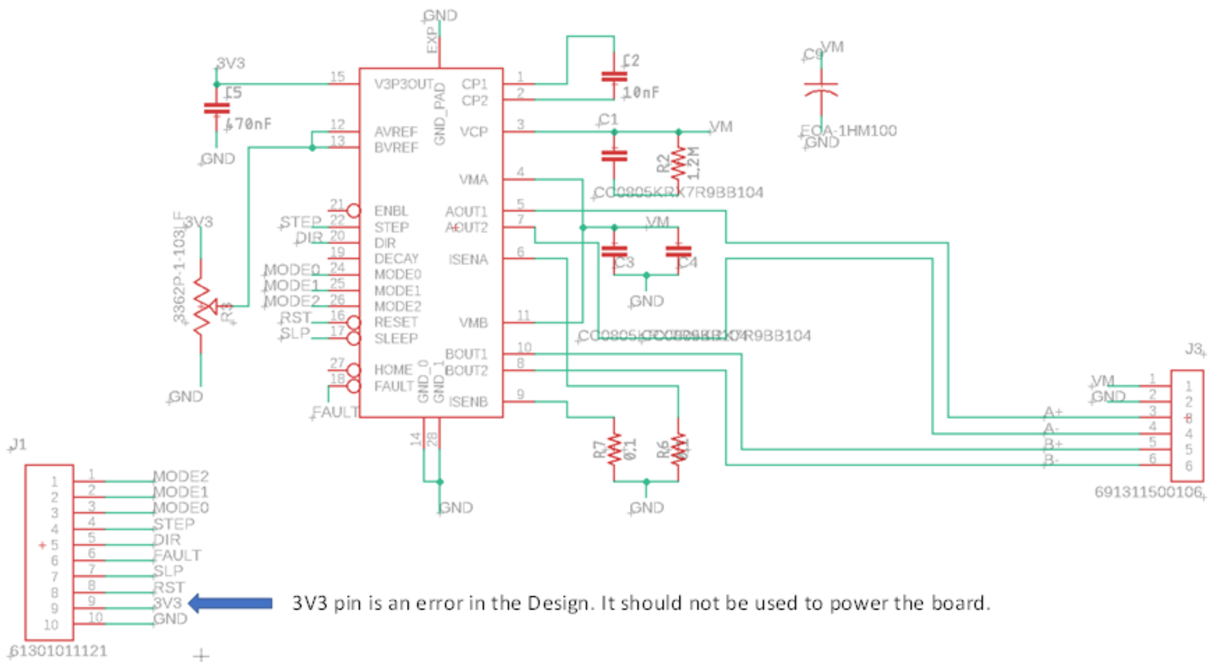


Figure 51: Schematic includes 3V3 pinout

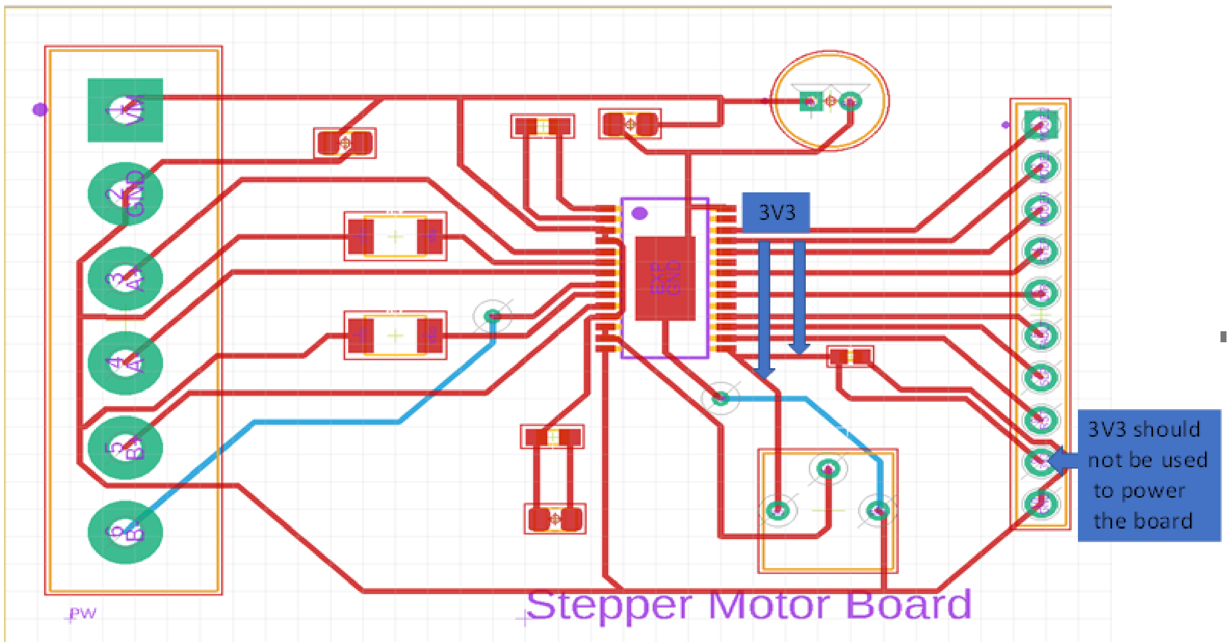


Figure 52: PCB layout includes 3V3 routing error

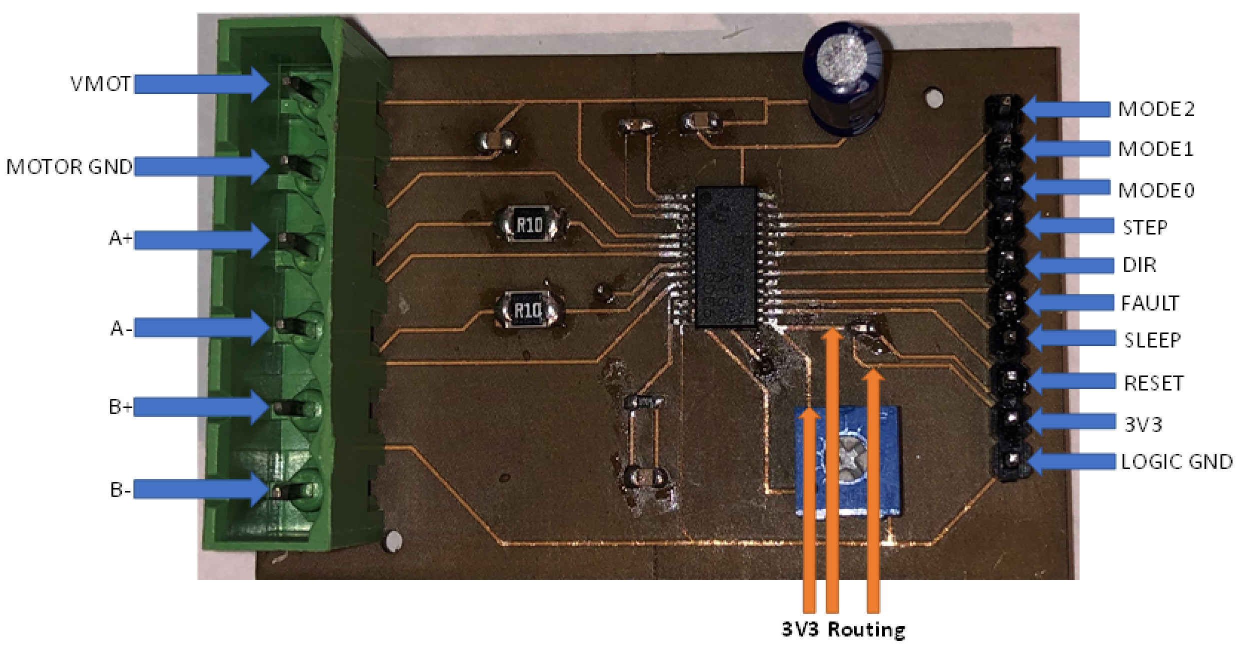


Figure 53: Circuit board shows copper traces of 3V3

To power the circuit, FAULT is connected to the SLEEP pin through a 10kohm resistor that acts as a FAULT pull-up whenever SLEEP is externally held high. No external pull-up is necessary on the FAULT pin. A 1.5Kohm protection resistor is in series with the FAULT pin which makes it safe to connect this pin directly to a logic voltage supply.

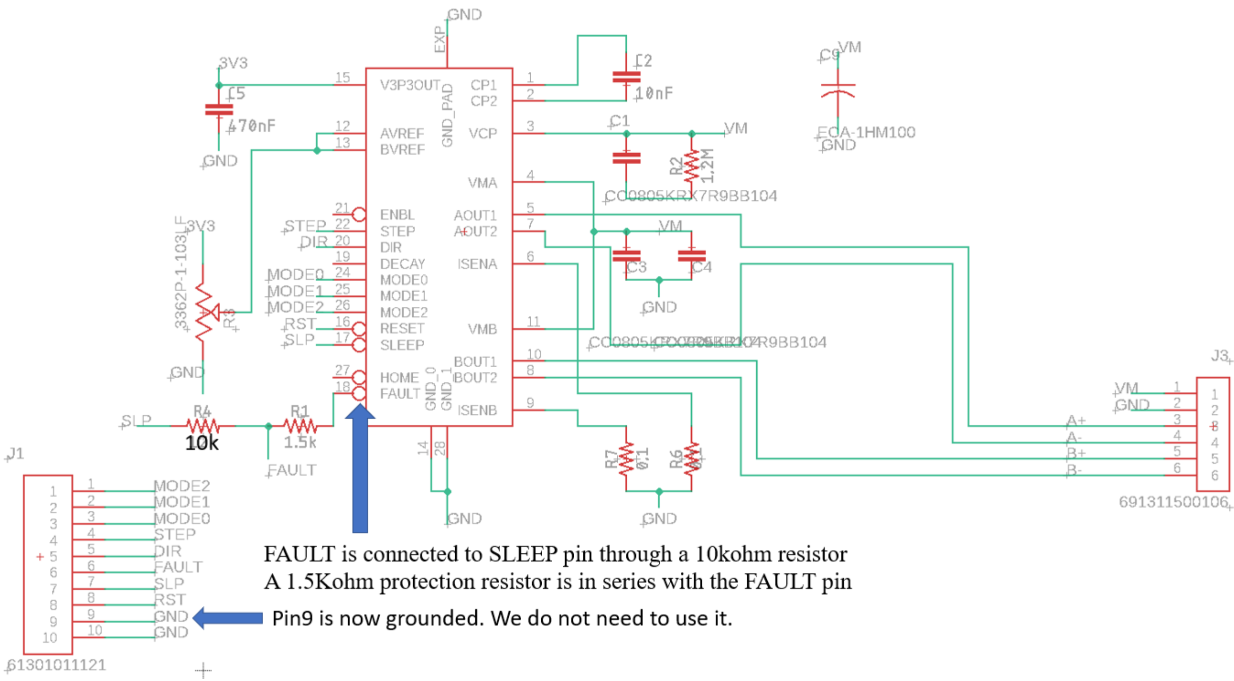


Figure 54: Schematic modified

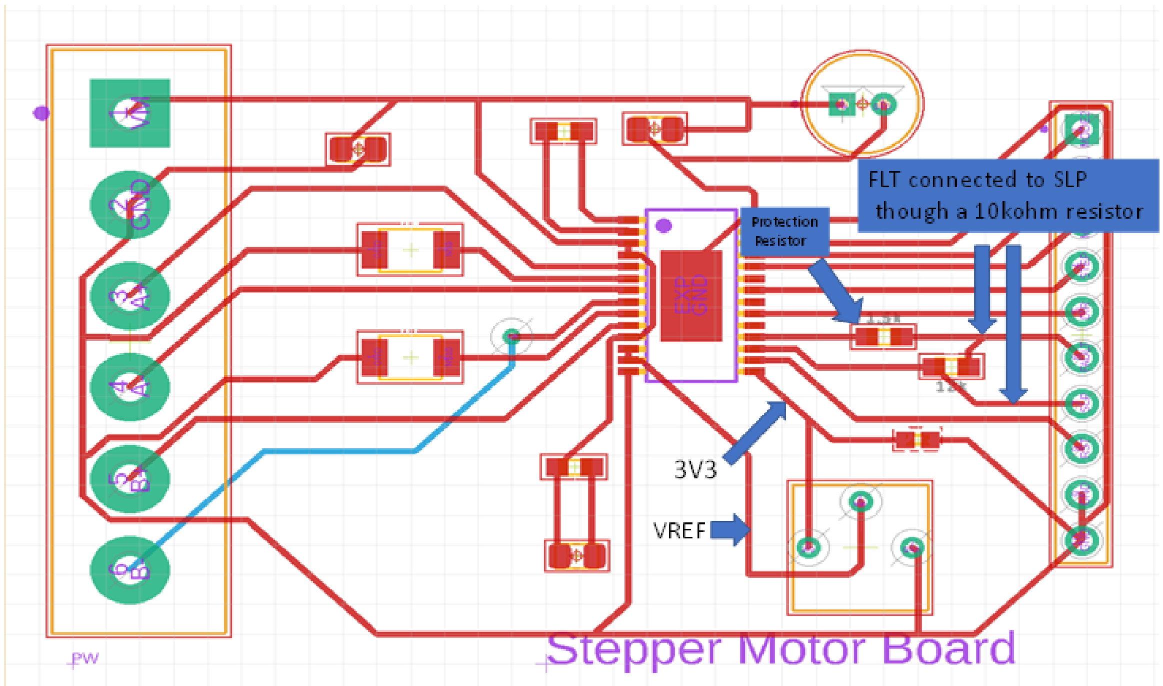


Figure 55: PCB layout modified

In the following step, another driver board was fabricated. However, the 10kohm and the 1.5kohm resistors were externally connected on a breadboard.

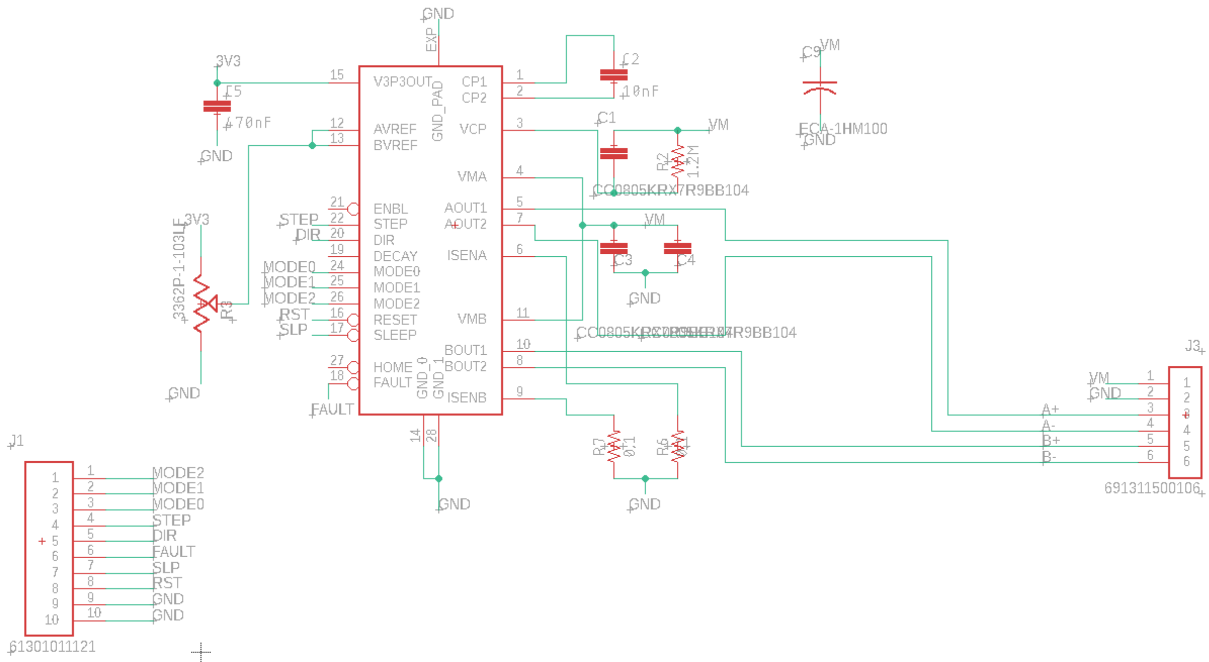


Figure 56: Schematic of a functioning stepper driver board

After pursuing the earlier steps identified and powering the board, we successfully managed to run the stepper motor. The figure below shows the measurements taken using a multimeter across different components on the board. The modes pins are set to LOW for full step operation. The voltage reference on the potentiometer is set to 0.75V accordingly.

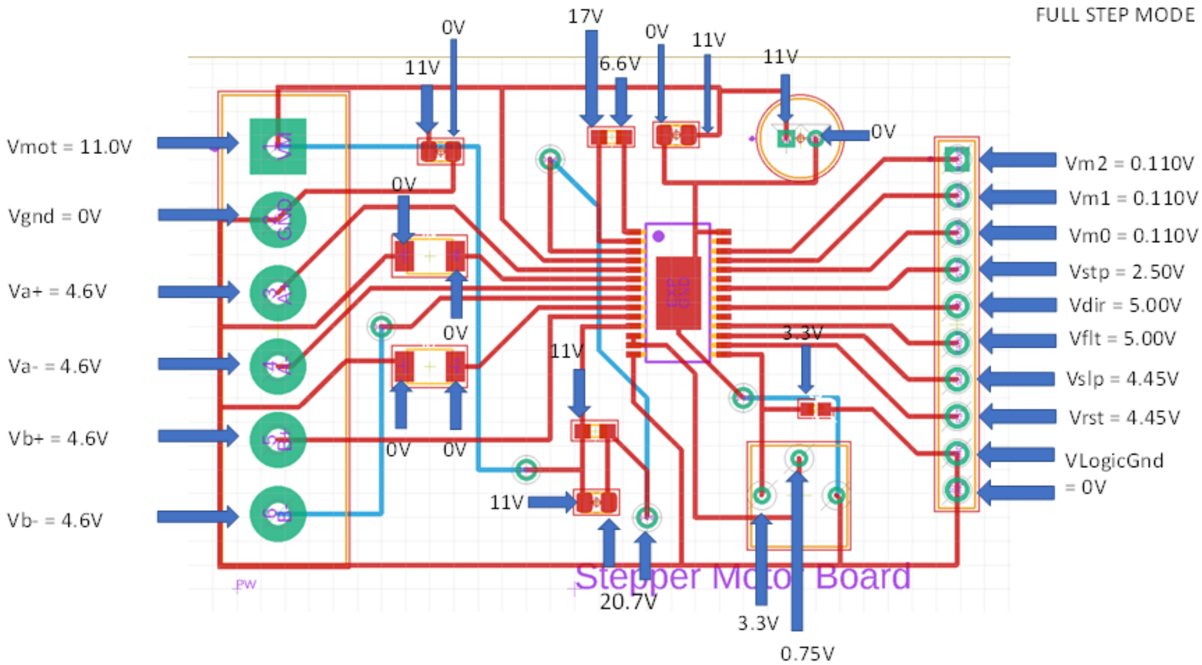


Figure 57: Multimeter measurements of the working driver board

Testing the STEP pin:

Arduino Code:

```
const int dirPin = 2;
const int stepPin = 3;
const int stepsPerRevolution = 200; // Motor steps per rotation

void setup()
{
  // Declare pins as Outputs
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}
void loop()
{
  // Set motor direction clockwise
  digitalWrite(dirPin, HIGH);

  // Spin motor  1 revolution slowly
  for(int x = 0; x <stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2000);
  } //this will cause a pulse to be emitted from the step pin and that will drive the step input on the drv8825


  delay(1000); // Wait a second
```

Figure 58: Arduino code to spin the motor clockwise

We started our loop by spinning the motor clockwise. We wanted to spin the motor one rotation at a relatively slow speed. We used a for loop to go from zero to the steps per revolution (=200) and that will go one revolution and will increment one by one. For every increment, we set the step pin high, delay it by 2000µs, then send the step pin low and delay by 2000µs.

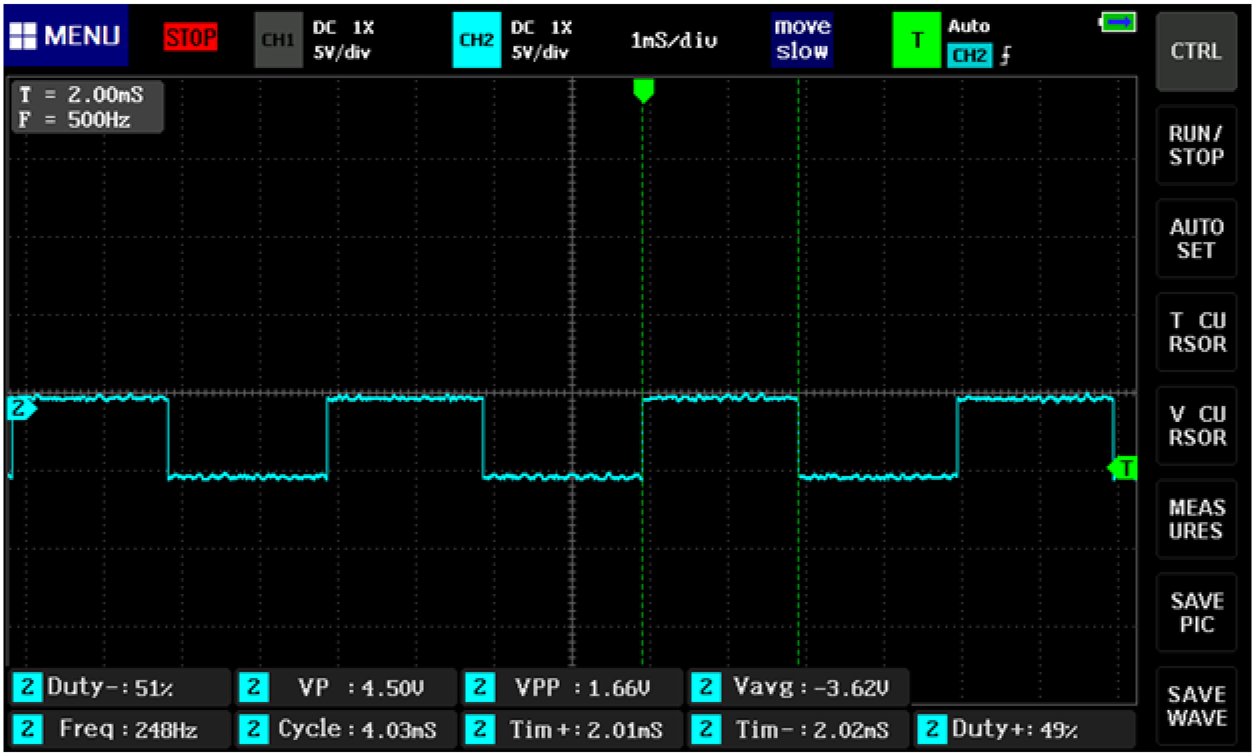


Figure 59: Oscilloscope reading of STEP pin

We noticed the pulse emitted by the step pin turning high for 2ms then turning low for 2ms.

V peak to peak = 5V

V Avg = 2.5V

We are using a Mini Digital Storage Oscilloscope. The readings on the oscilloscope are inaccurate due to the noises and the length of time it takes to register the correct values.

We then set the motor to rotate counterclockwise. We intended to spin the motor two rotations faster. Therefore, for every increment, we set the step pin high, delay it by 1000µs, then send the step pin low and delay by 1000µs.

Arduino Code:

```
  // Set motor direction counterclockwise
  digitalWrite(dirPin, LOW);

  // Spin motor 2 rotations quickly
  for(int x = 0; x < stepsPerRevolution*2; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
  delay(1000); // Wait a second

}
```

Figure 60: Arduino code to spin the motor counterclockwise

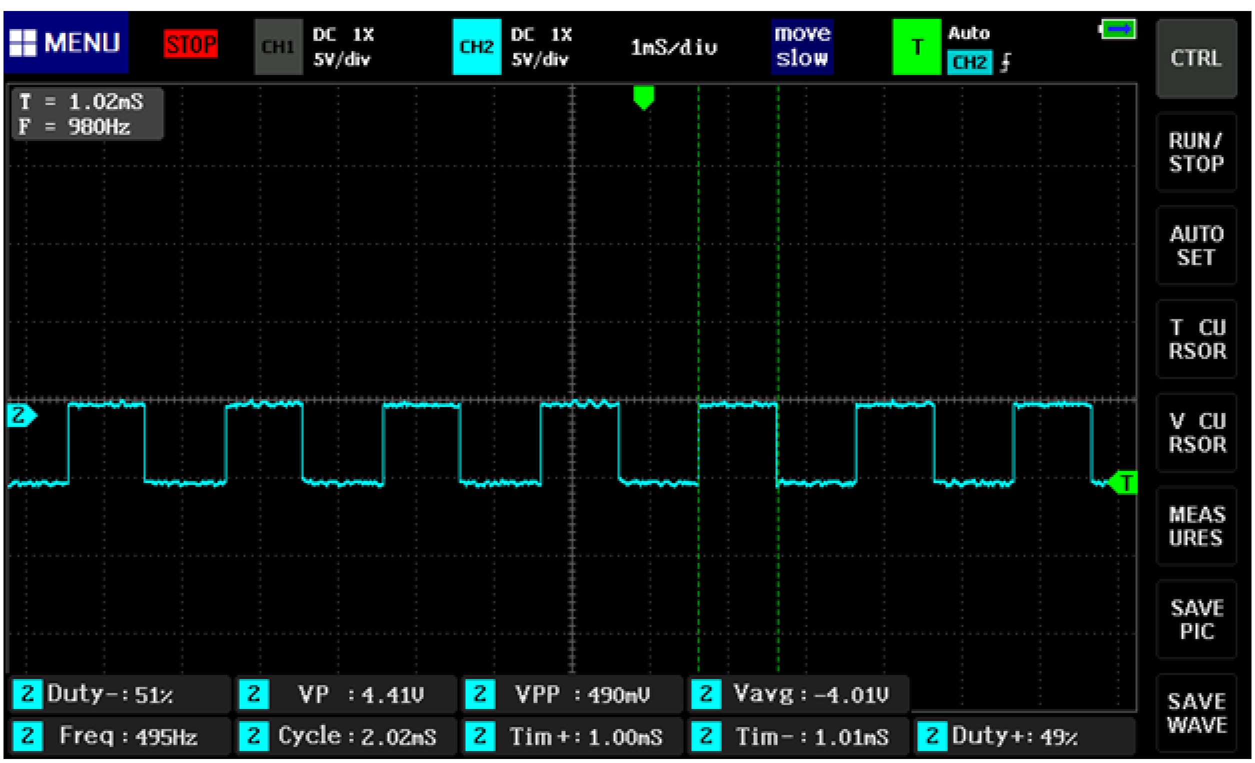


Figure 61: Oscilloscope reading of STEP pin

We see the pulse emitted by the step pin turning high for 1ms then turning low for 1ms.

V peak to peak = 5V

V Avg = 2.5V

Testing the DIRECTION pin:

Implanting the previous Arduino code and running the motor both clockwise and counterclockwise, we should be able to witness a HIGH voltage when the motor is stepping in the clockwise direction and LOW voltage when the motor is stepping in the counterclockwise direction.

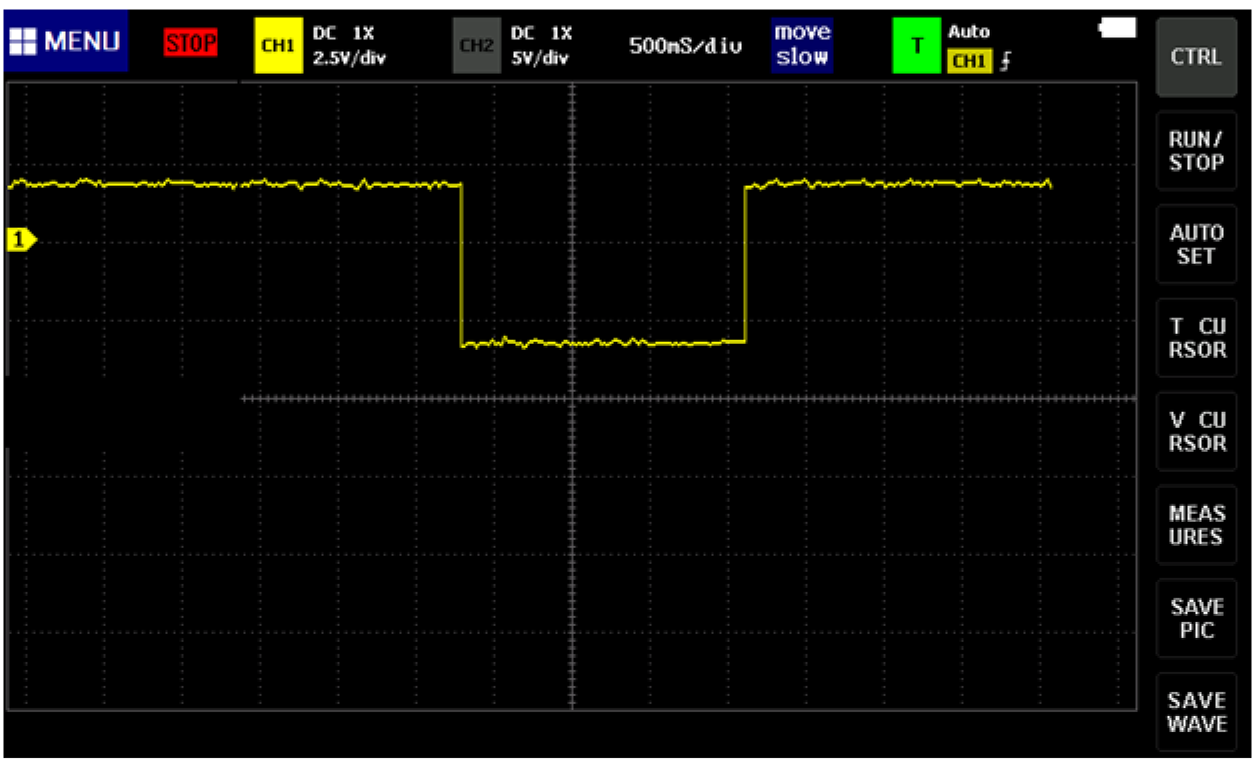


Figure 62: Oscilloscope reading of DIRECTION pin

Testing the current across the coils:

We are stepping the motor in Full Step mode clockwise. We are expecting to witness a lot of noise since the motor is not stepping in a smooth tiny increment. The A-A' terminal and B-B' terminals in the bipolar stepper motor need its polarity reversed.

The pulse waveforms for each coil are shown in the oscilloscope readings below.

Coil A+:

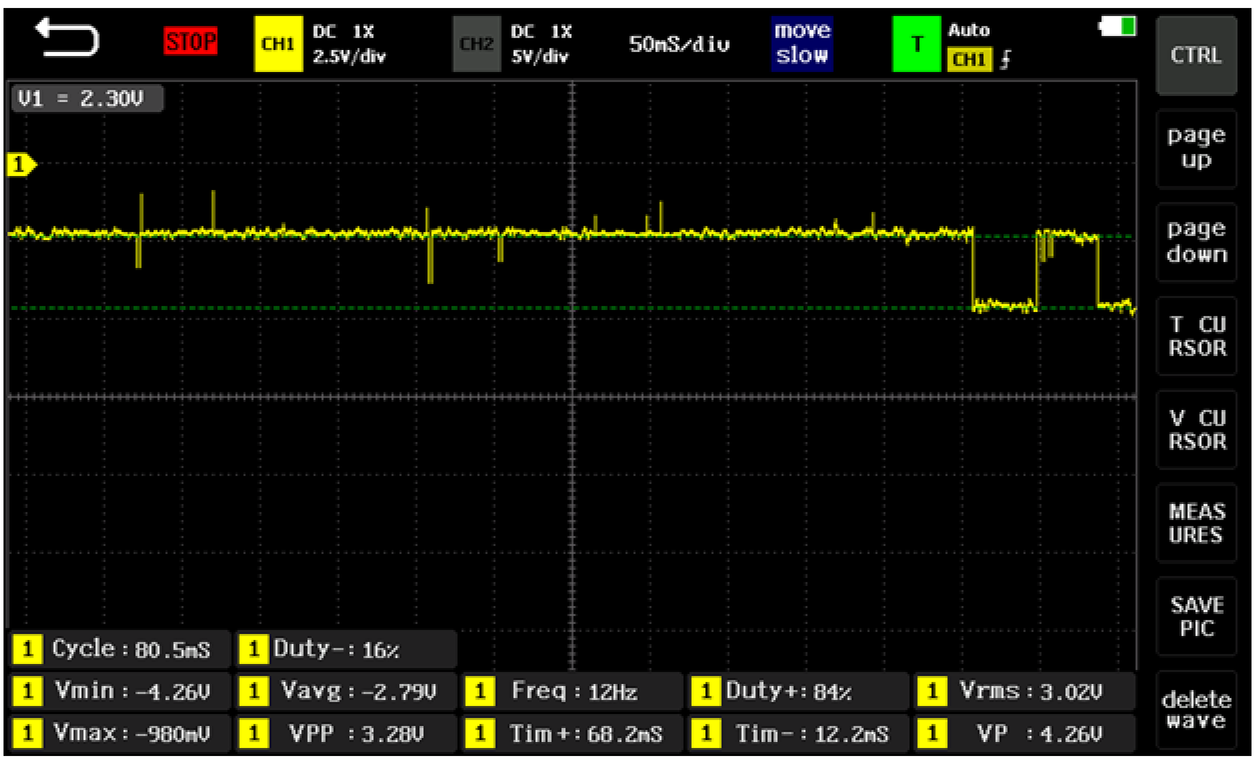


Figure 63: Oscilloscope reading of Coil A+

Coil A+, when the motor is at rest, shows a HIGH pulse.

Coil B+:

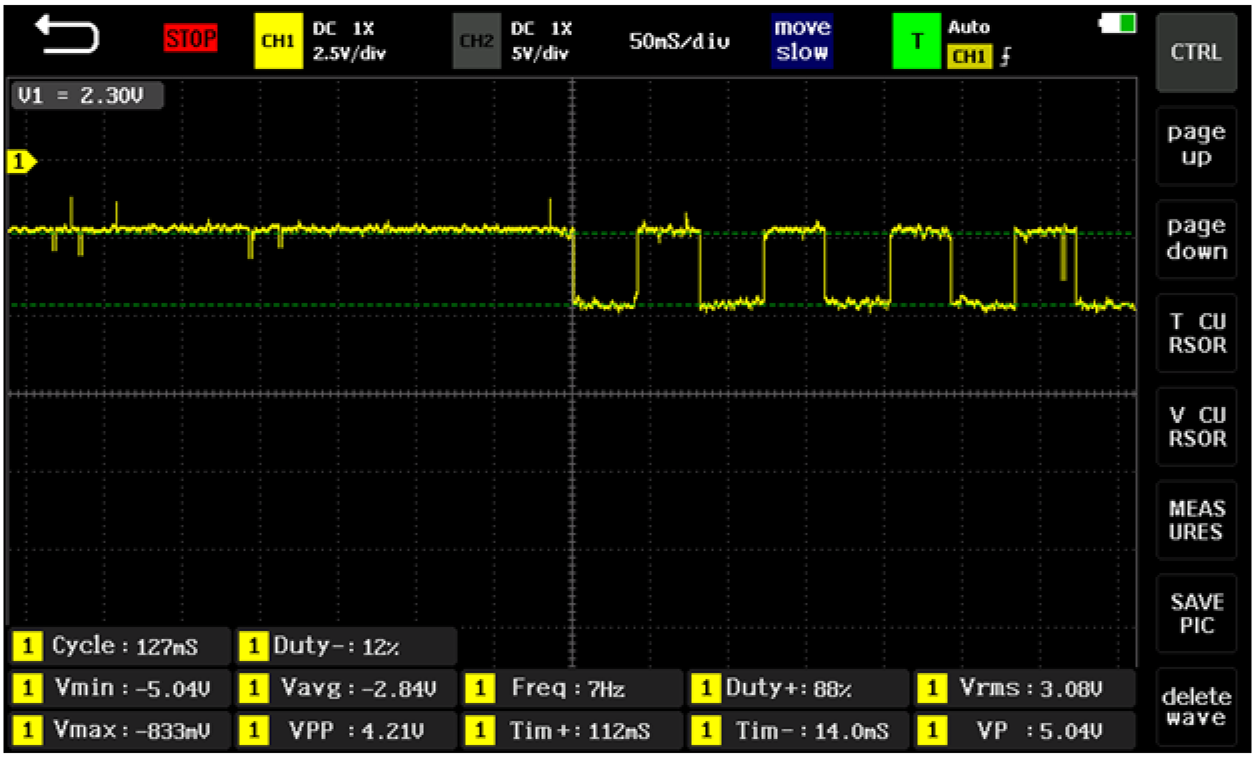


Figure 64: Oscilloscope reading of Coil B+

Coil B+, when the motor is at rest, shows a HIGH pulse.

Coil A-:

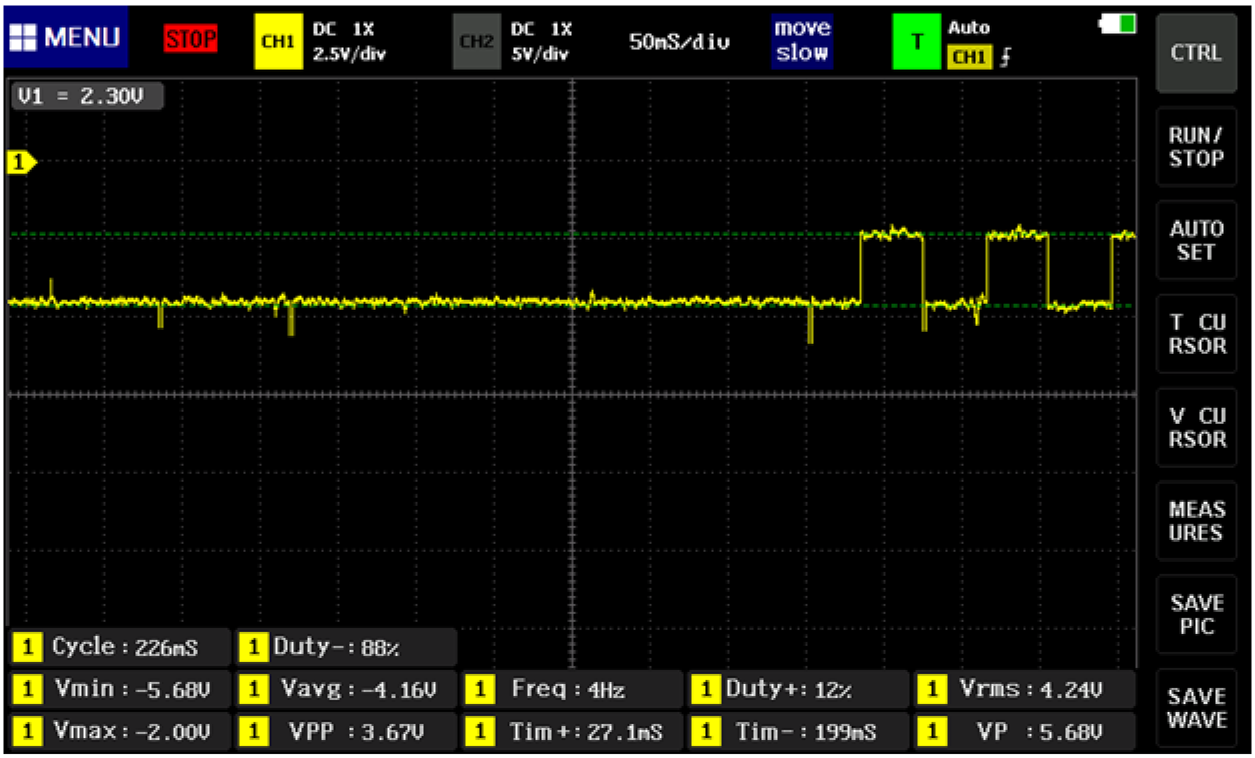


Figure 65: Oscilloscope reading of Coil A-

Coil A-, when the motor is at rest, shows a LOW pulse.

Coil B-:

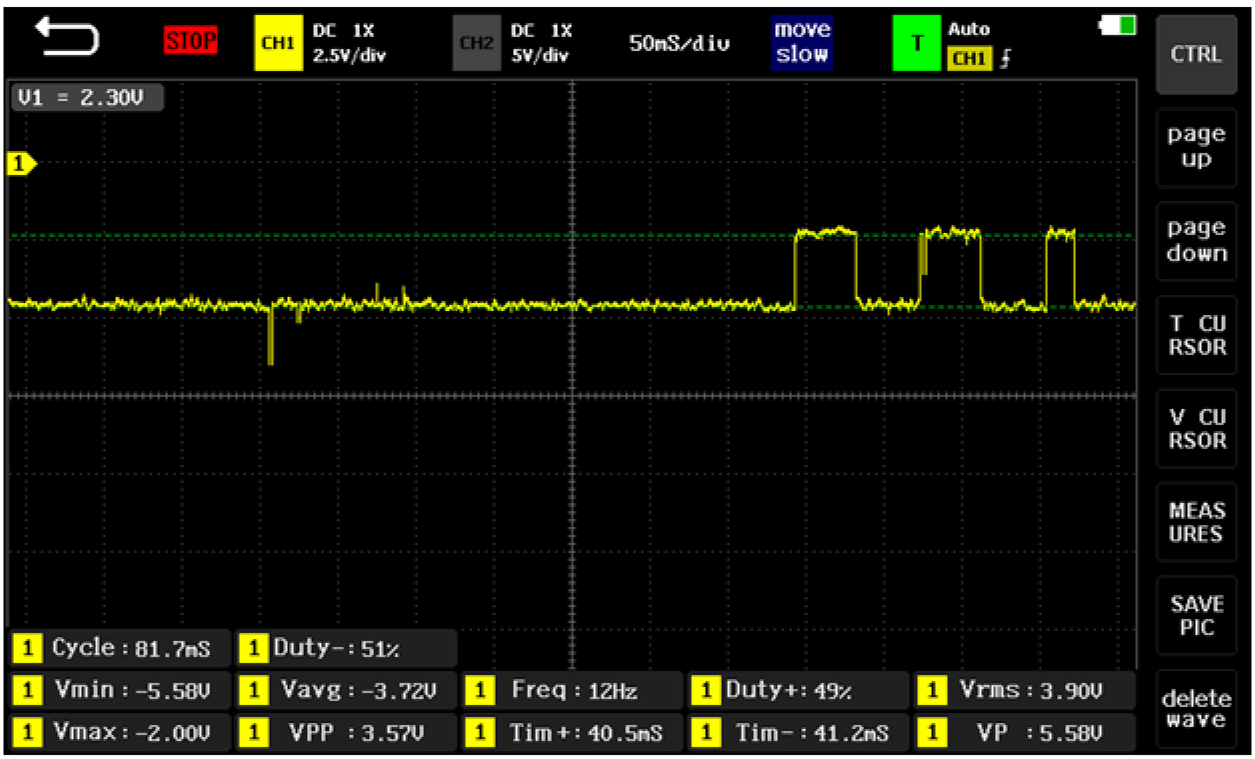


Figure 66: Oscilloscope reading of Coil B-

Coil B-, when the motor is at rest, shows a LOW pulse.

Controller Board:

Ramps 1.4 provides the interface between the Arduino Mega and the electronic devices on the 3D printer. In this section, we will construct and deliver a controller board similar to the Ramps 1.4, which is the brain within our 3D printer. The board controls four stepper motors and monitor and alters the temperature of the hot-end of the extruder. As well as that, it is responsible for the end-stops which are important for finding the home position for the axes.

The controller board will acquire four stepper drivers designated for the X, Y, and Z axes and the extruder, three endstops circuits, one heater circuit, one thermistor circuit and one fan pinout.

Stepper Drivers:

Schematic:

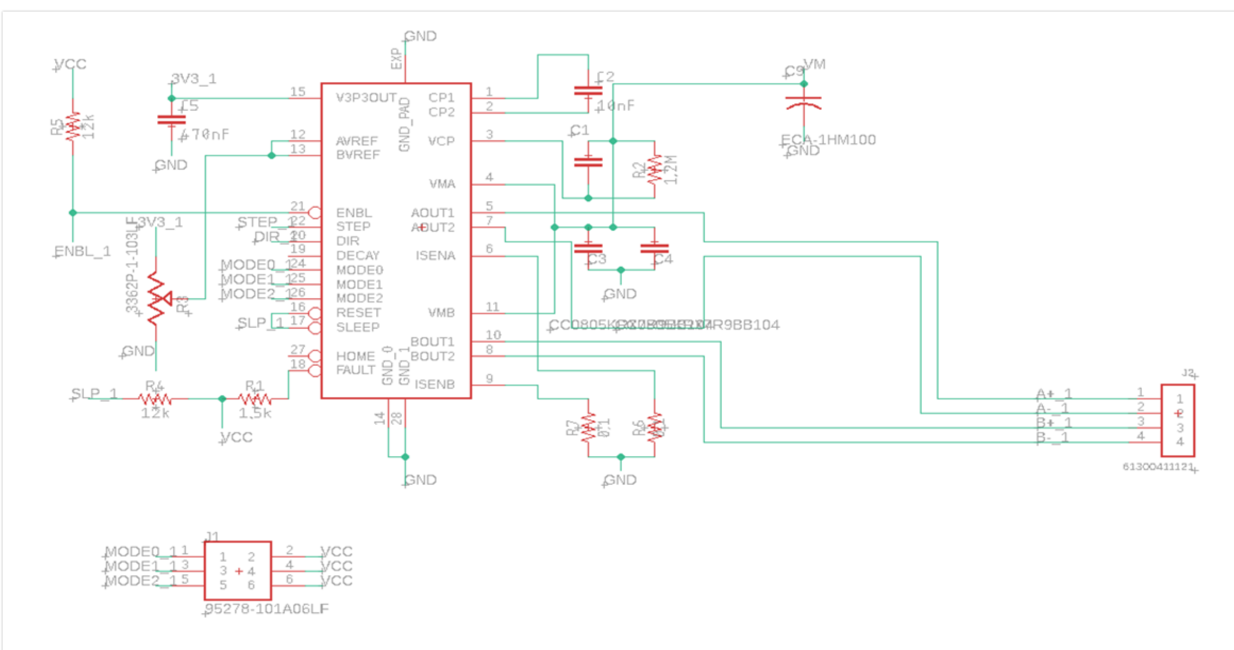


Figure 67: Stepper driver schematic

The circuit above displays one stepper driver. We included four circuits for four stepper motors. The digital pins connected to the microcontroller are step, direction, enable and VCC for each driver. For setting the stepping mode, a set of electrically conductive pins are mounted to the PCB in such a way that pins are standing in a vertical direction to the PCB. They are arranged in groups called jumper blocks. These contact points can be closed by a jumper box. We will have for each driver three jumper blocks (MODE0, MODE1, MODE2).

Figure 68: Electrically conductive pins

Figure 69: Jumper box

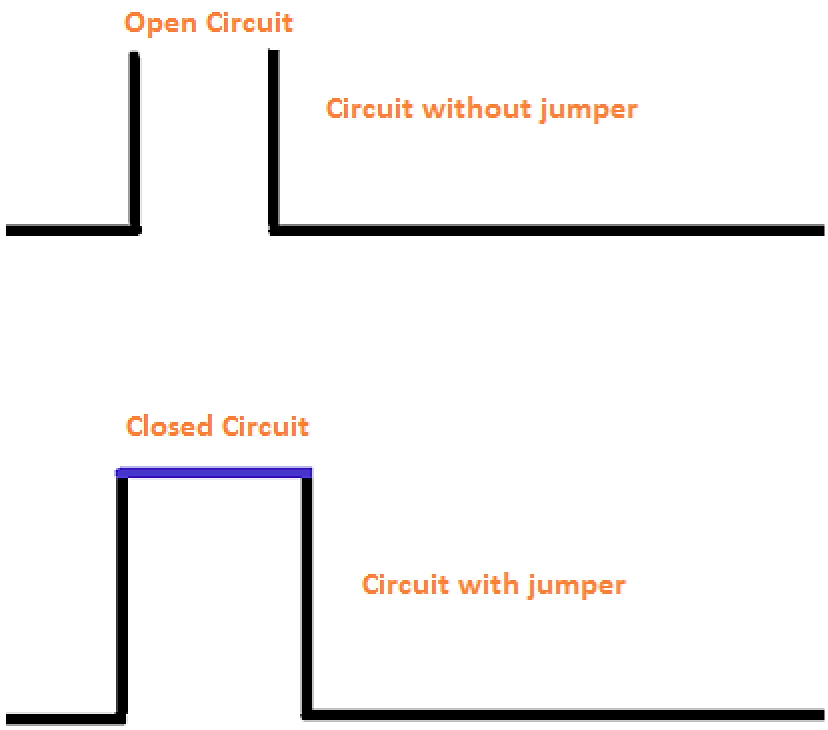


Figure 70: Electronic jumpers explained

https://www.physics-and-radio-electronics.com/blog/electronic-jumpers/

We have used a connector header and connected one side to the three modes and the other side to HIGH (VCC).

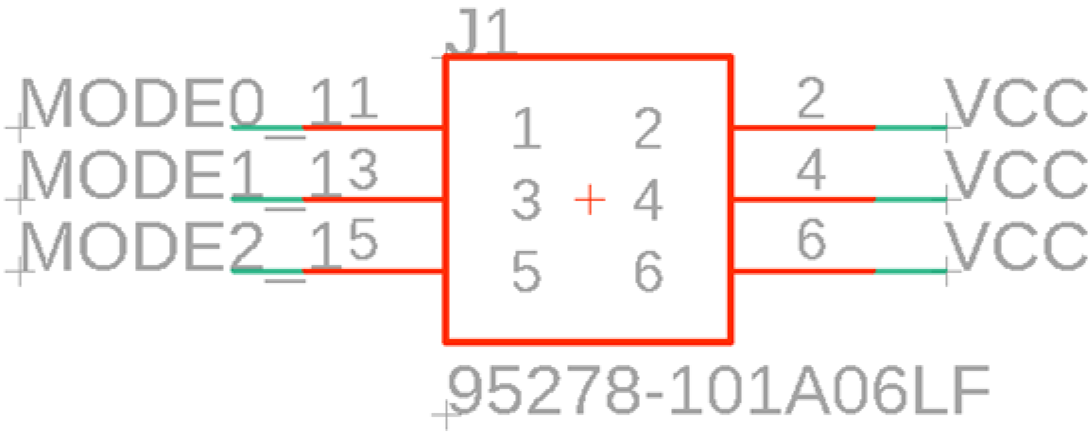


Figure 71: Schematic of the connector header

Mechanical Endstops:

A simple mechanical switch positioned to trigger when an axis reaches the end of its motion.

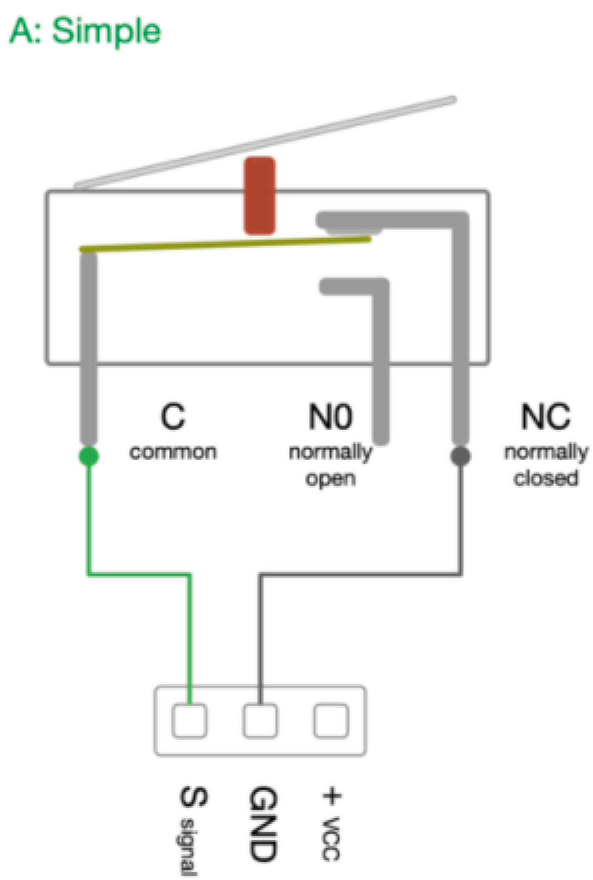


Figure 72: Simple endstop design

https://reprap.com/wiki/Mechanical_Endstop

Common (C) connects to the input normally close (NC) when the lever is not pressed. Common (C) connects to the input normally open (NO) when the lever is pressed.

Pin	Function
VCC (+)	Pin to supply +5V on

SIG(S)	Signal pin. It will output high(+%) if it is triggered, or low if it is clear.
GND(-)	The Ground pin

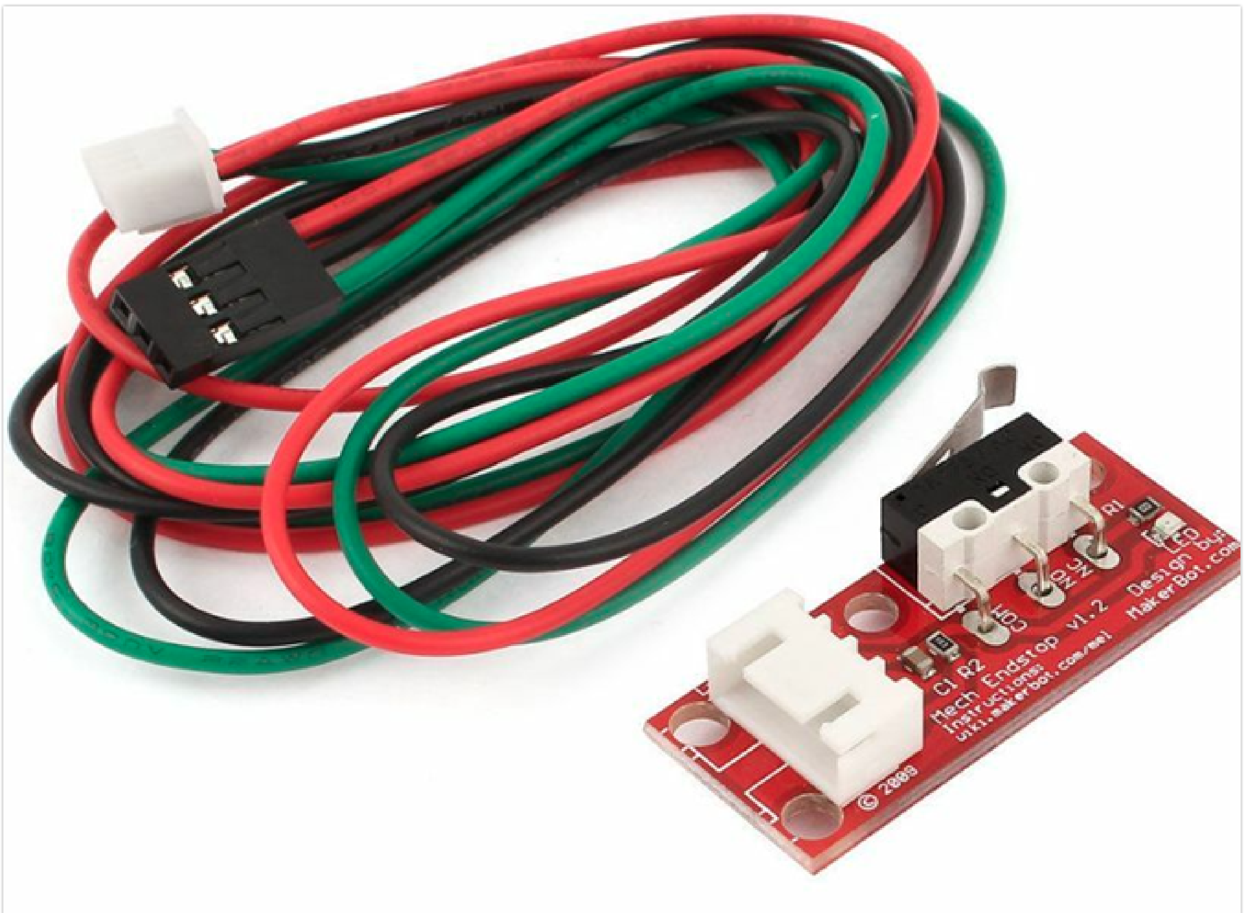


Figure 73: Mechanical endstop

https://www.compassdhmprojects.it/switch-finecorsa/501-finecorsa-meccanici-reprap-mechanical-endstops-3d-printer-endstop.html

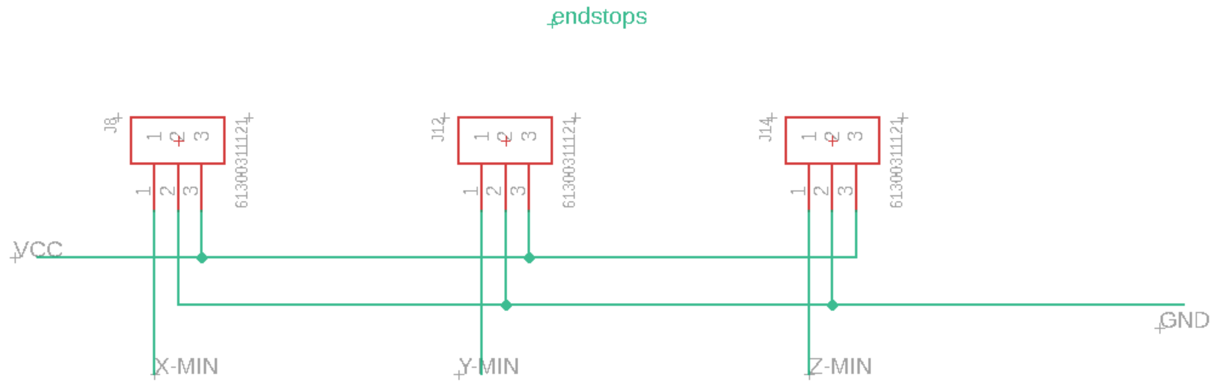


Figure 74: Endstops schematic

For the axis X,Y, and Z, we have three endstops to trigger when an axis reaches X Min, Y Min, and Z Min. X-MIN, Y-MIN, Z-MIN pins are connected to the Arduino's digital pins.

Hot End:

The hot end is the part of the extruder that heats and melts the material. It has two chambers, one melt chamber and one cold end chamber. The melt chamber is heated to melt the filament to a proper level. The cold end chamber prevents the softening and swelling of the filament. A fan is attached to the cold end to prevent the heat from affecting the filament before it reaches the melt chamber. The heater block holds the heater cartridge and thermistor. The thermistor monitors the temperature of the heater block and sends its values to the control board. The heater cartridge is responsible for heating up the hotend.

Heater Cartridge:

To build the circuit, we used an N-channel enhancement type MOSFET, STP55NF06L. It is a logic level mosfet. It is normally off when the gate-source voltage is 0V, so a voltage must be applied to the gate for current to flow through the drain-source channel. That is, the arduino output pin must be set to HIGH to provide power.

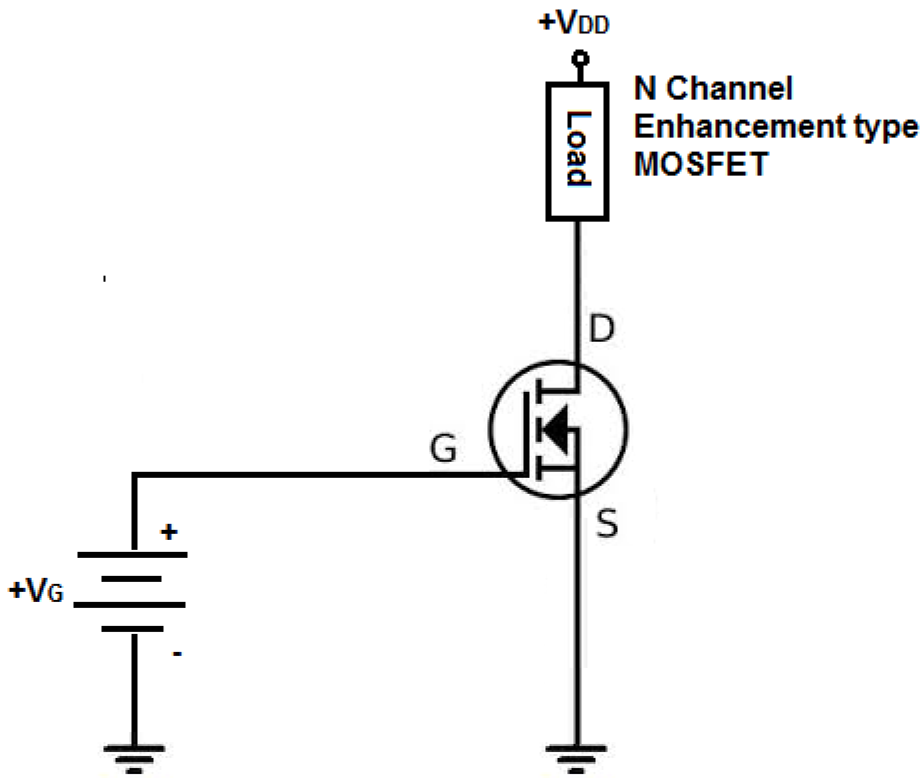


Figure 75: N-Channel enhancement type MOSFET

http://www.learningaboutelectronics.com/Articles/N-Channel-MOSFETs

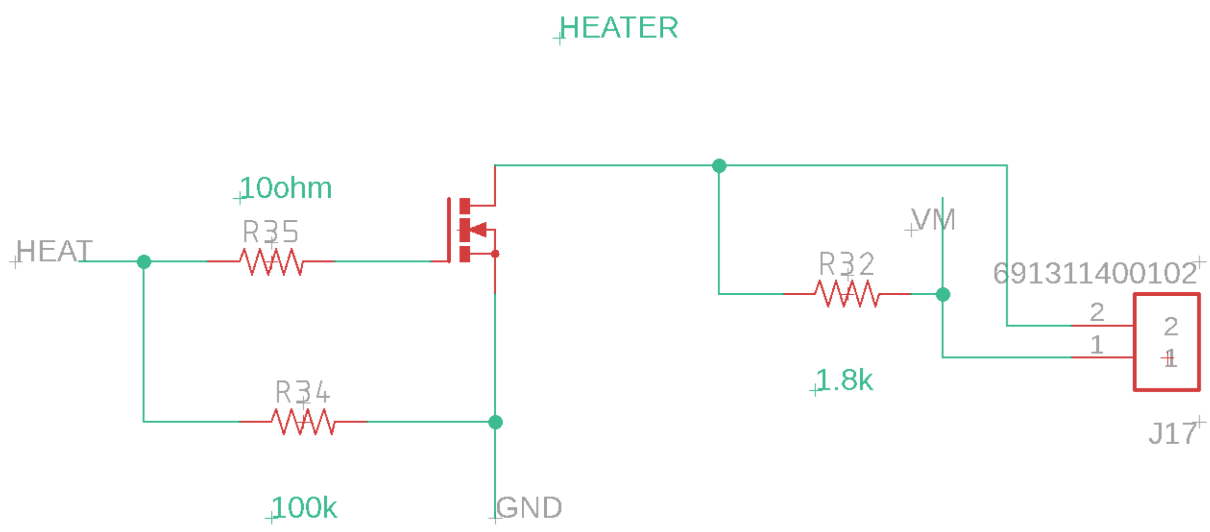


Figure 76: Heater cartridge schematic

The gate is highly capacitive and can draw a big instantaneous current. Therefore, resistors have been placed with the pin "HEAT" connected to the Arduino's digital pin. The drain is connected to the voltage supply and the source is connected to ground. When the gate voltage is 0V, no current flows through. When the positive voltage is applied, voltage is generated between the drain and source.

Thermistor:

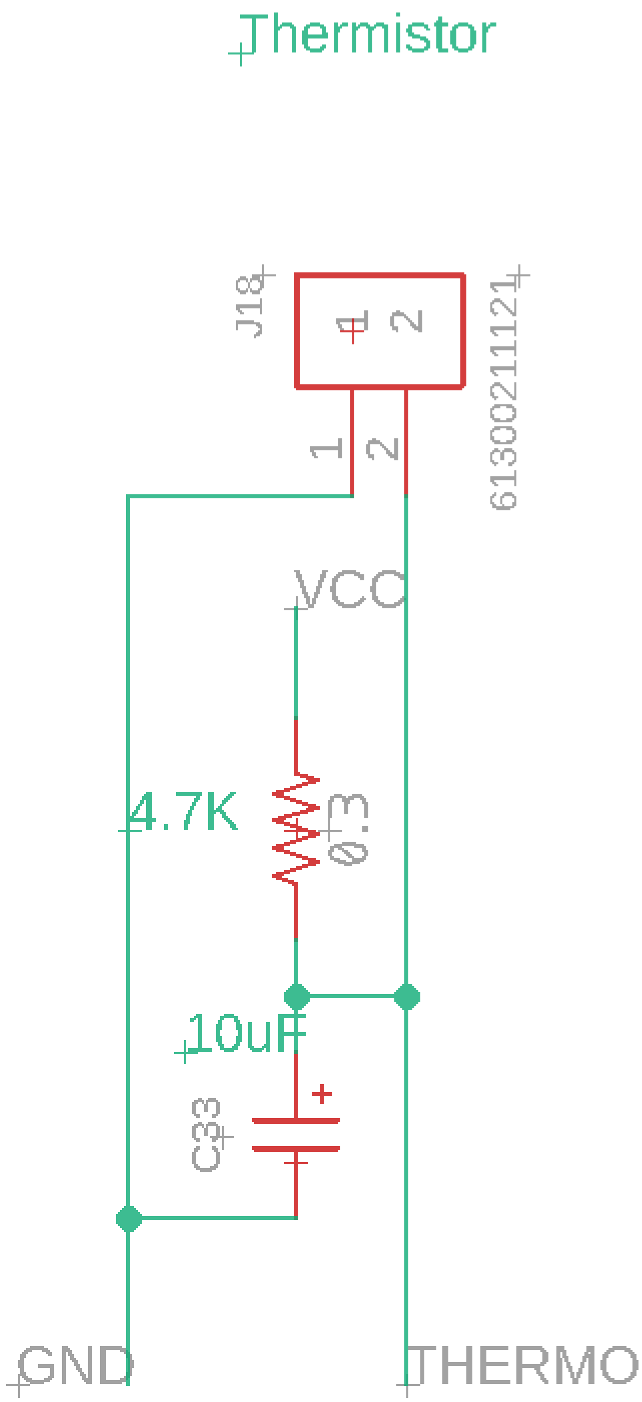


Figure 77: Thermistor schematic

The resistor is used to limit the voltage coming from VCC (5V). The capacitor is used to lower some of the noise to get a stable reading. "THERMO" pin is connected to the Arduino's analog pin.

Fan:

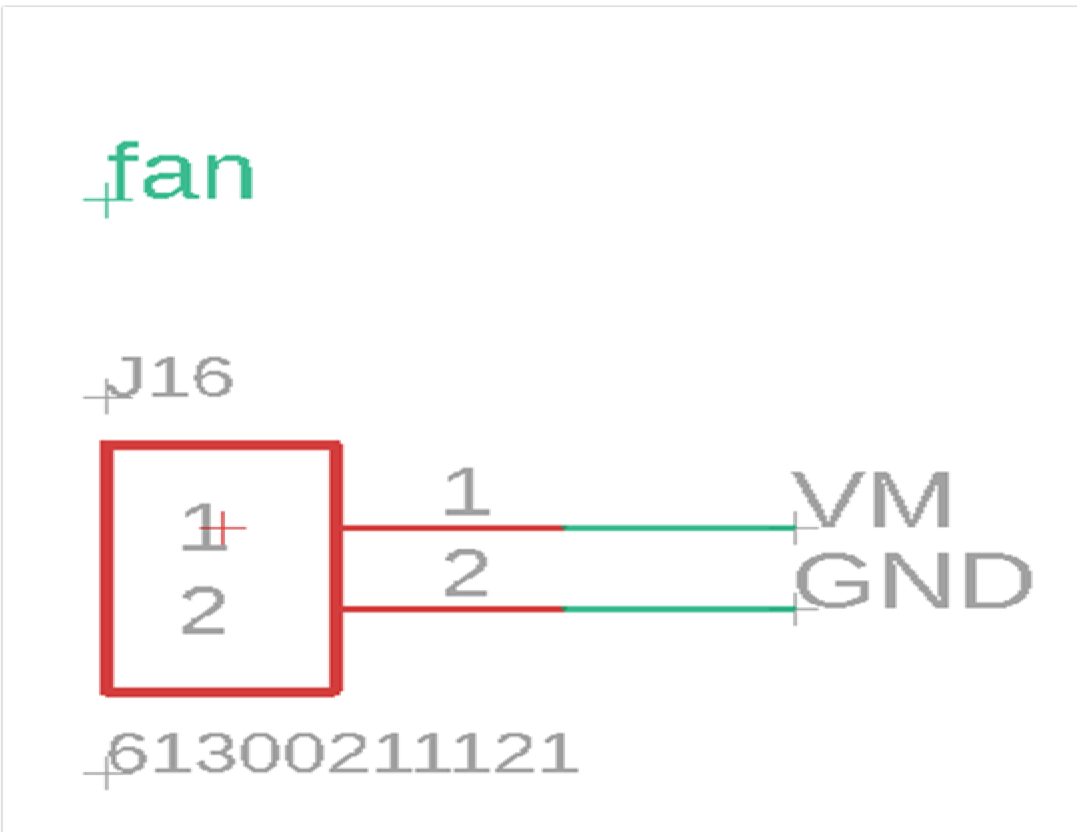


Figure 78: Fan schematic

The fan is driven with the power supply voltage.

Powering the Controller Board:

The factors to keep into consideration when applying the voltage necessary to the control board are the Arduino Mega maximum input voltage, filtering capacitor maximum voltages, and PTC fuse maximum voltages.

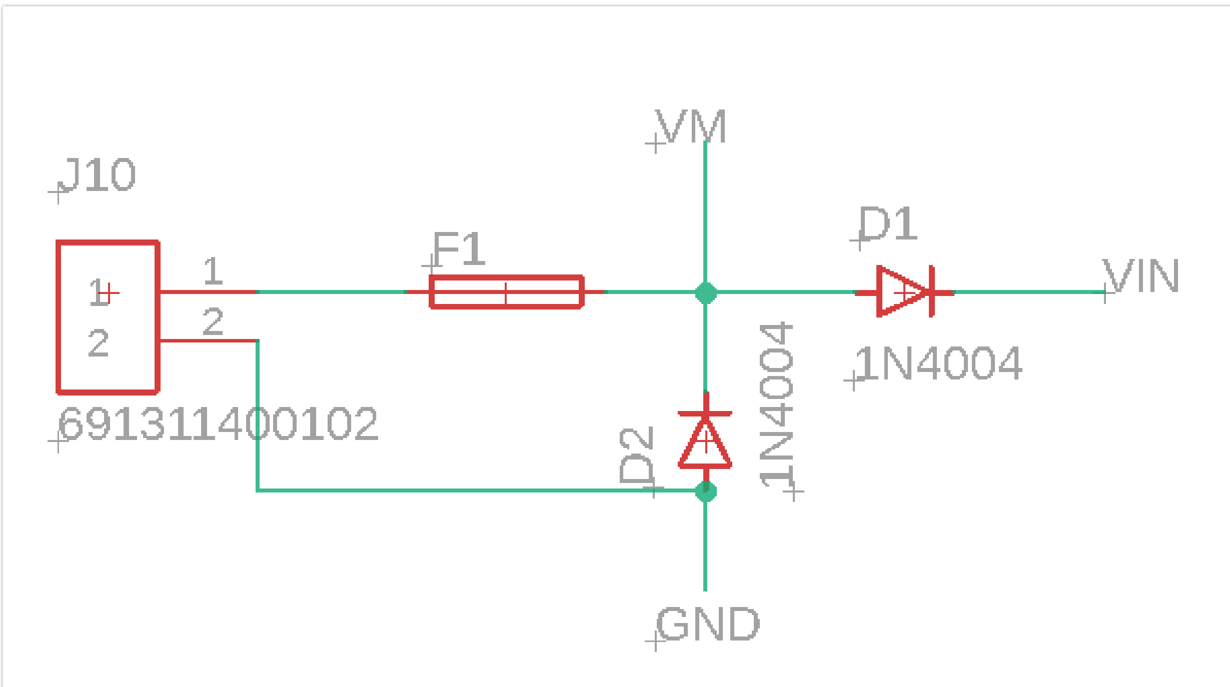


Figure 79: Circuit to power the controller board

F1: MF-R500 (5A) fuse rated to 30V.

D1 and D2: 1N4004 (400V) general purpose rectifiers diode with the maximum current carrying capacity 1A and they withstand peaks up to 30A.

The 1N4004 diode D1 connects the board to the Arduino Mega which has a recommended maximum input voltage of 12V (VIN). if the diode is not present, the arduino should be connected through the USB connector or through a separate 5V line. When choosing the value of the electrolytic capacitors for protection, the safe measure is to have their voltage double the rate maximum input voltage.

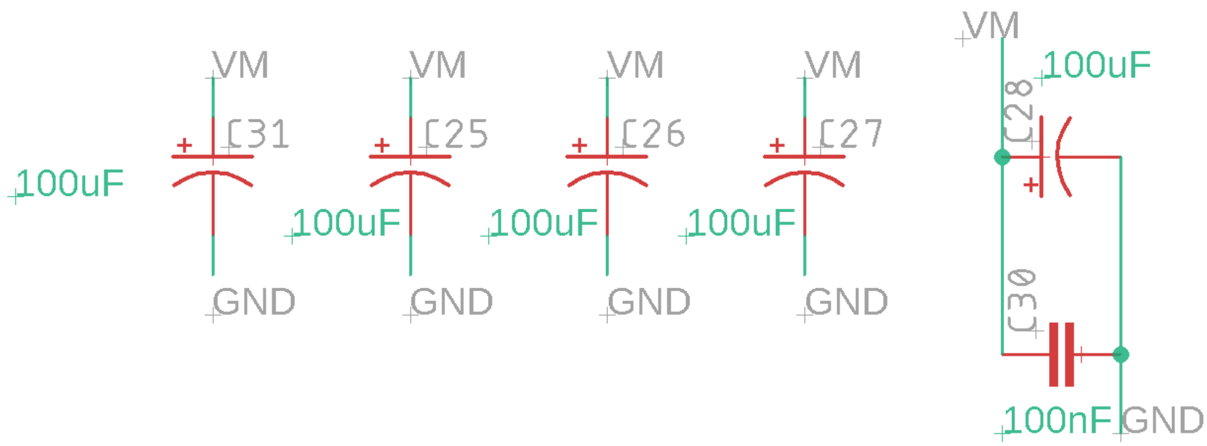


Figure 80: Electrolytic capacitors

Final Schematic and PCB layout:

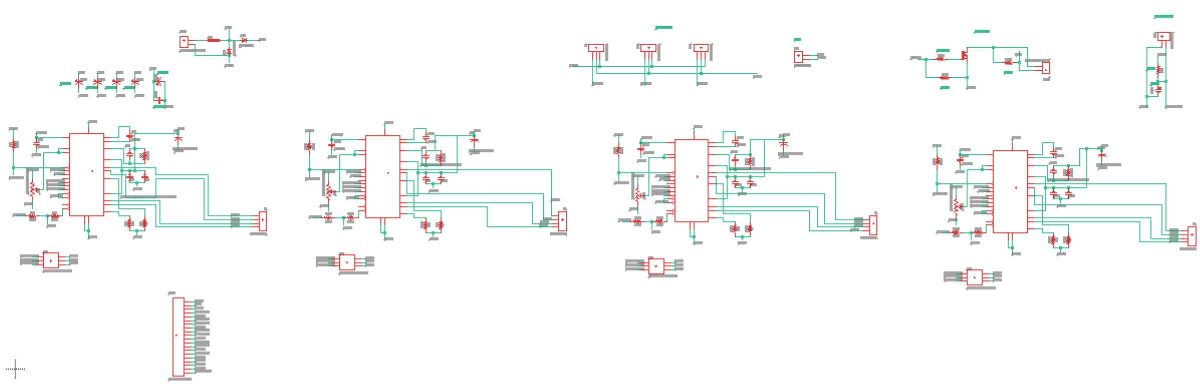


Figure 81: Full schematic of the controller board

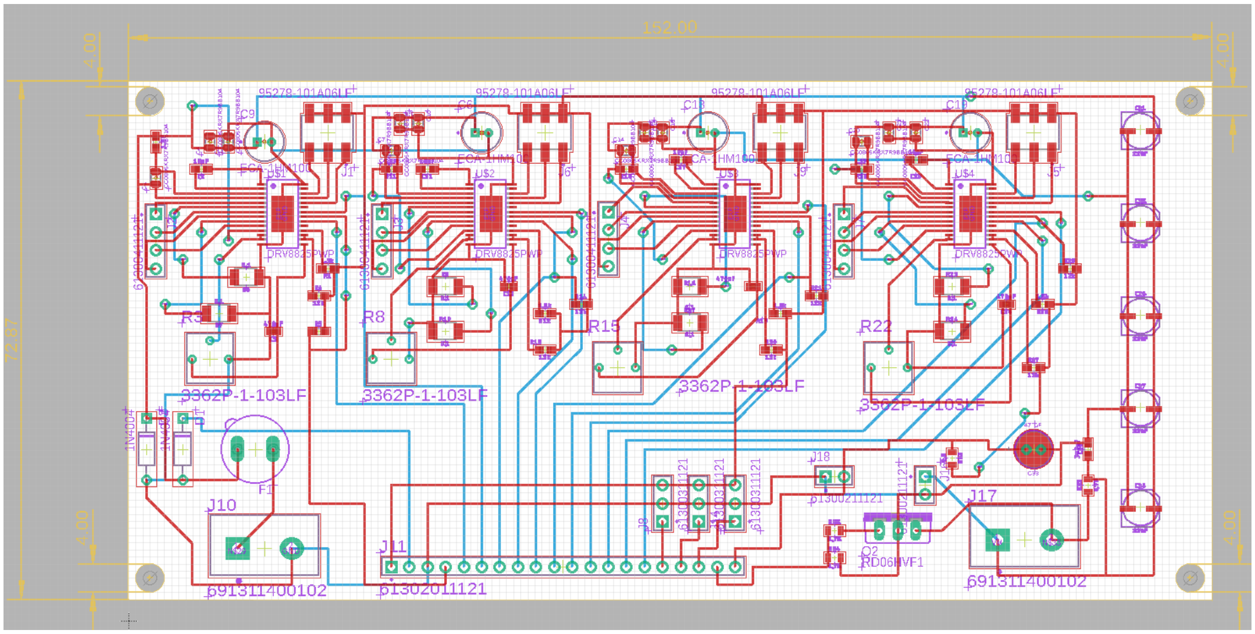


Figure 82: PCB layout of controller board

The board dimension is 152x72.82mm.

The board was fabricated and soldered.

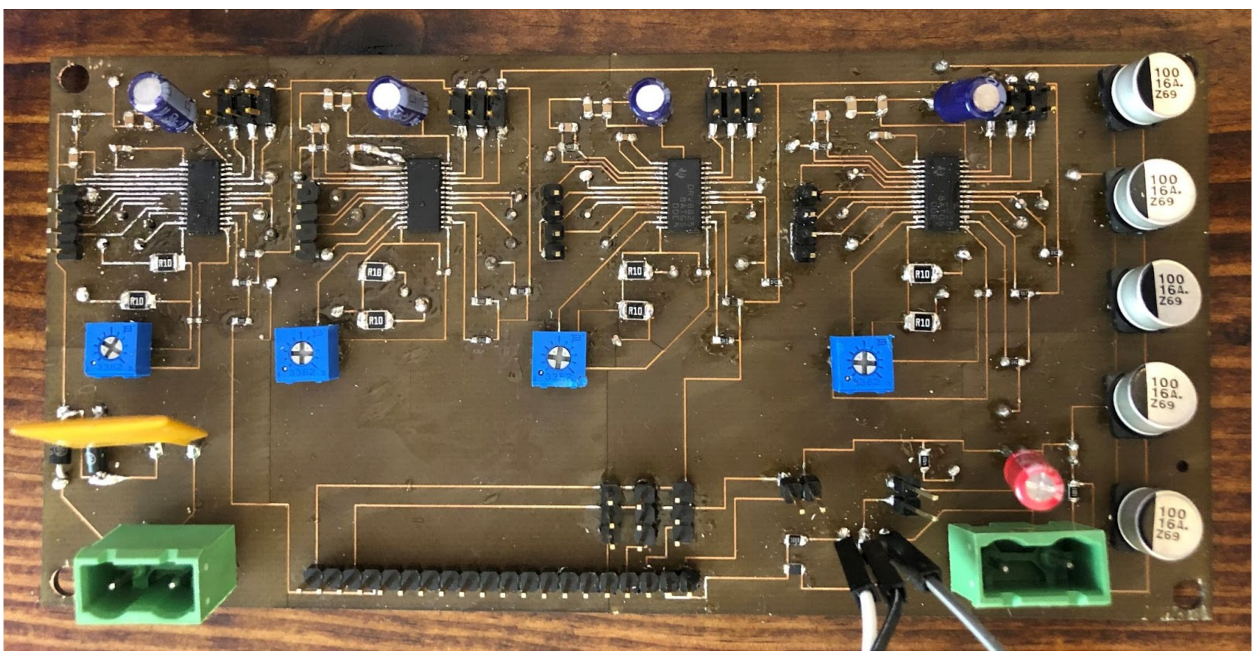


Figure 83: Controller board

Initial CAD Designs:

Linear Actuator:

The first step to completing our first printer assembly was to provide a structurally sound base for our print bed. Our two guide rod design was sufficient but due to our desire to save space we removed one guide rod. This was accomplished by removing the one guide rod and offsetting the threaded rod to the spot where the removed guide rod was previously. To allow linear motion in the Y axis, an open ended box was created on the print bed where the threaded rod passes through. By placing a nut in that box we are able to fix the position of the nut causing the print bed to move forward and backward rather than rotating. The other side of the print bed is fixed by using a small piece of shim stock to act as a cantilever beam on the guide rod.

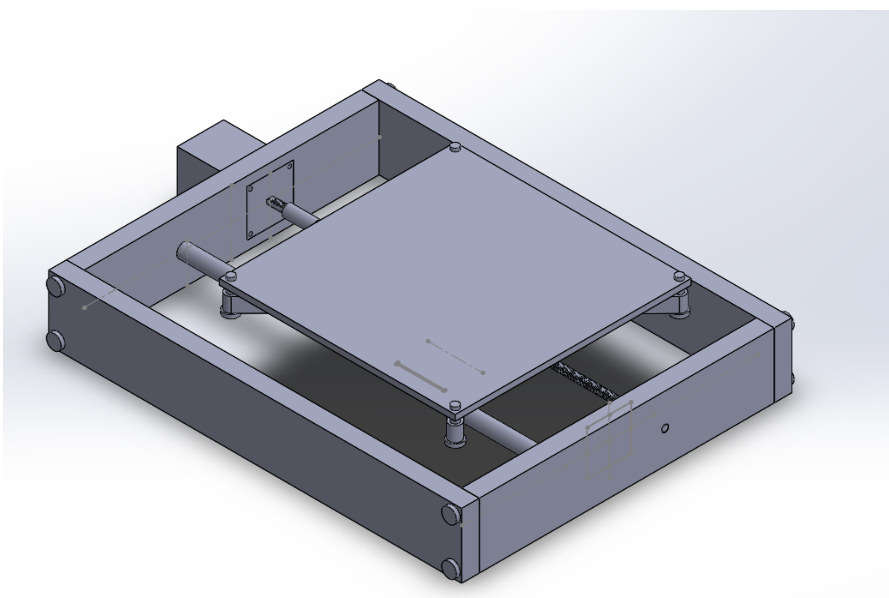


Figure 84: Linear actuator (one guide rod)

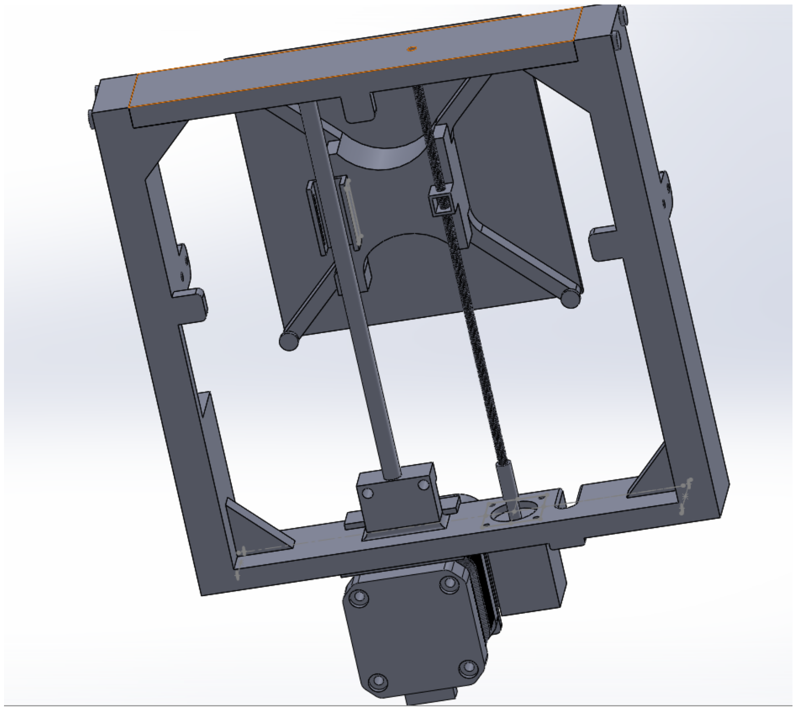


Figure 85: Bottom view of linear actuator

Core XY:

Once we had our design for our base, we had to find a way to cover two axes of motion in one plane, to accomplish this we decided to go forward with our core xy idea from our initial brainstorming. The first thing we realized as we decided to go forward with this design was that our initial mockup was unsuitable to go forward with, this was due to our mechanical limitations as well as a need for a deeper understanding of the mechanism. The first change we made to implement the core xy mechanism into our design was to move it to the center so it split the linear actuator in half longways. To achieve motion within the Z axis for this design we implemented rails into the sides of the core xy mechanism to allow the center printhead assembly to be pulled up and down. The sides of this assembly are fitted with bearings that ride within the rails that have been implemented. To accomplish motion in the X axis we fitted two parallel guide rads that go across the X axis. This is where the print head assembly is fitted to a pla cart that rides along the two guide rods.

After the method of motion in our X and Z axes were established, we had to design a way to drive this motion without causing too much bulk. We opted for two Nema 8 stepper motors to drive our motion within the core xy mechanism. Our initial idea to mount these motors was to fix them within the base and use a pulley system to transfer motion into our desired orientation. This ended up becoming far too complicated as well as adding an unneeded amount of complexity to our design.

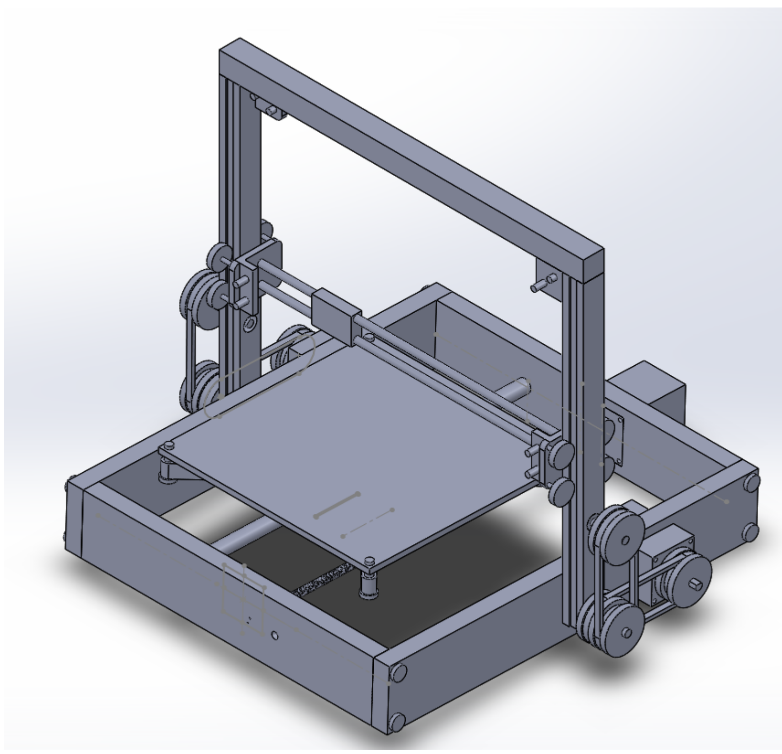


Figure 86: First complete CAD assembly

Once we decided not to go with our idea of the pulley system we had to go back to the drawing board to figure out a way to mount our Core XY motors. Our next solution was to mount the two motors within the same plane as the core xy to maintain the simplicity in the mechanism. This was achieved by designing motor mounts that grabbed the sides of the Core

XY assembly. While the overall idea was sound due to the size of our assembly, creating separate motor mounts to attach to our assembly proved to risk our structural integrity as well as compromise our accuracy due to human error.

Other changes were also added within this iteration of our CAD design, at this point in our project we were able to start ordering our parts needed for our design. This caused us to take another look at the sizes of our parts relative to each other. We realized that we had to rethink the sizes of our parts due to the limitations of creating a design on such a small scale. The major change we had to make was to enlarge the components that hold the pulleys for the Core XY. Due to this change, we also had to widen our assembly to maintain our desired print space for our 3D printer.

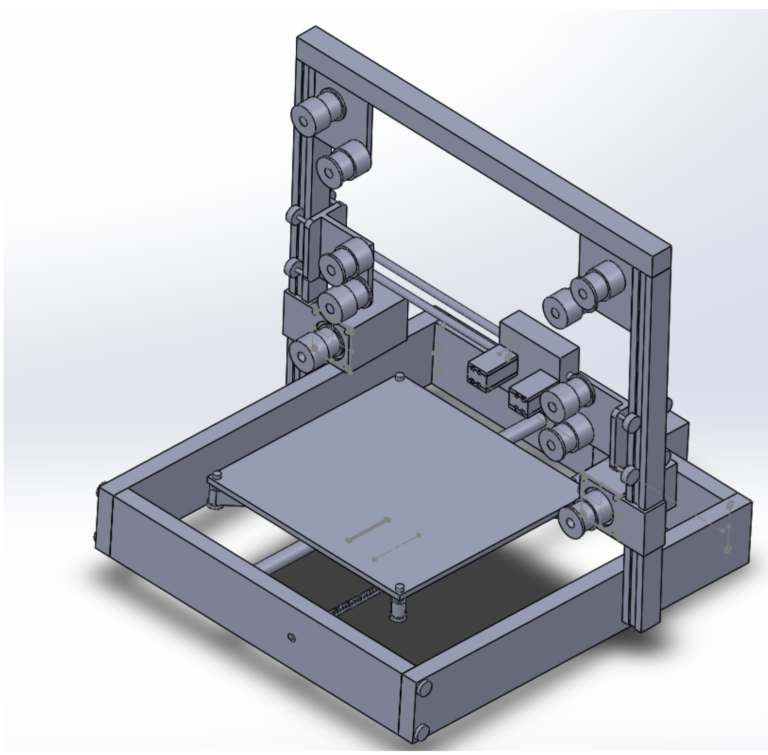


Figure 87: Second complete CAD assembly

Simplifying the Assembly:

After trying to manufacture the previous iteration of our design, we realized we had to simplify our core xy assembly to eliminate error that is caused by the size of our design. The main change that was made to accomplish this was to take our fixed components of our core xy and integrate them into the frame so that they are all one part. This was accomplished integrating three components into one part. The first thing that was integrating was the frame itself. Previously, the core xy frame was going to be three parts fastened together at the ends, this was changed to be one whole frame that required no further assembly after printing the part. The next component integrated was the Nema 8 motor brackets. Instead of having separate brackets that grabbed the side of the frame, we integrated the brackets directly into the frame. The final component integrated was the pulley mount that was placed at each corner of the core xy. Originally this component served two purposes, the first was two mount the pulleys needed to complete the mechanism, they also served a secondary purpose as a way to fasten the different parts of the frame together. Since the secondary purpose was eliminated from our assembly, we were able to integrate the component directly into the part.

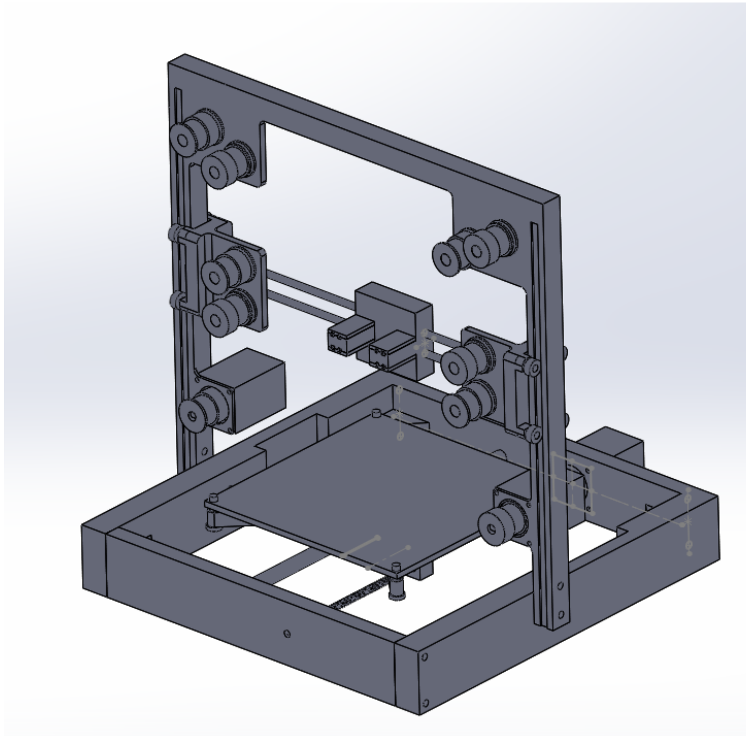


Figure 88: Simplified CAD design

While this design in theory accomplished our needs for the 3D printer design, we encountered various problems when it came to manufacturing our design. The first problem we encountered was that we overestimated the strength of PLA for our design. After initial manufacturing, we found that our motor mounts for our 3D printer were too flimsy to meet our needs. This caused us to thicken our motor mounts as well as widen the mounts to allow more tolerance for the holes made to fasten the motors with screws. Another change made to our assembly was to add gussets to various corners on the assembly where we say that structural assembly was a concern.

Another major change made to our design was made due to our previous decision to implement various components into one part. When printing the Core XY frame, the shape of the rails required us to use support material to complete the print, but due to the size of our design the rails were too small to create support material. This flaw caused the rails that were face down to not print, instead that chole side was a flat face. In turn, we had to figure out a way to print this part without changing its orientation as well as not using support material. Our solution to this problem was to change the rail to a v cut that narrowed down to the correct width at the desired height. Due to the steep slope of the v cut, support material was not needed to close the gap created by the cut.

The final change made to our assembly was the clamps used to grab our open ended belts at the center. As seen in previous figures, we had four separate clamps to grab each end of the open ended belt, these proved to be too small for practical use. These were changed to become two separate plates that served the same purpose. Each plate has a hole in the center that allows them to be fastened using only one screw and nut.

First Prototype:

3D Printing Parts:

Our 3D Printer design consists almost entirely of 3D printed parts. To manufacture these parts, the 3D printing lab within the makerspace was used. There are two types of 3D printers available to use in the 3D printing lab which are the Ultimaker 3 and the Lultzbot TAZ6 printers. While the Ultimaker 3 has more precise tolerancing compared to the other printer, the TAZ was able to successfully print our smallest parts without too much difficulty.

To gain access to these printers to use one has to become a basic or advanced user by passing the online quiz for 3D printing through foisie makerspace. Once someone becomes a basic or advanced user you are able to access 3DPrinter OS where you can upload your files to be printed. The process of taking a solidworks file through 3DPrinter OS to be successfully manufactured is not too difficult. First, once the intended solidworks file is complete it has to be saved as an stl. due to that type of file being compatible with most if not all 3D printing softwares. Once that is done you are able to upload the stl. to the software where you can configure its orientation. For most files, 3DPrinterOS offers an optimal rotation option that would put your part in the most efficient orientation for the 3D printer. While this option is good enough for most parts being manufactured it may not always work. This is due to the fact that support material has to be used to make certain geometries that cannot be held up on their own during the printing process and if not careful this support material may not be able to be removed and can ruin your part. This can be avoided in a number of ways, the first way is to be mindful of small openings in your part where support material can become solid throughout and permanently meld to your part. Once these areas are accounted for you can orientate your part so those areas would not need support material throughout the manufacturing process.

The second way to avoid this issue would be to modify the geometry of your part to get rid of the need for support material. The general rule to follow for this is that most angles steeper the 45 degrees would not need support material to manufacture. While this is a clever way around some problems, it has to be made sure that these modifications to your part do not compromise the overall function and structure of the design. Once the desired orientation is set the next step is to slice the part within the software. This consists of the various manufacturing options you can set for the printer to create your part. In most cases the only option that should be modified to your needs would be the infill option. This option allows you to choose the density of your part to either create a smaller part or to save on time or printing materials. It was found that the smallest parts of the printer only needed a 30% infill to maintain structural integrity so it can be assumed that most parts would not need to exceed that infill value. There are various other options to modify such as layer height, type of print bed adhesion, print material used and the temperature of operation. In most scenarios, these other options would not be needed so once your desired options are set you can send your part virtually to the printer to be manufactured.

Considerations Before Manufacturing:

The majority of the major parts of our first prototype were 3D printed. This allowed us to produce parts very quickly. The quick turnaround time for parts allowed us to make changes, when needed, very easily without impacting our schedule too significantly. Since we were 3D printing the majority of our parts, we had to make a few changes to our designs to accommodate using PLA and needed support material as discussed previously.

One of the biggest issues that we ran into with 3D printing our parts was with the Core XY frame. The frame had to be printed laying down. This meant that the channels in the side rails that were on the bottom needed support material when they were printed. After the frame was fully printed, we were unable to remove this support material. The reason for this was that the channel was not very deep. It was only as deep as a single layer of PLA. This meant that when a layer of support material was added, it essentially just filled in the channel with a full layer of PLA. We were unable to cut this out and decided to try using a dremel to try and grind away the layer of PLA. We were able to use a Dremel to remove some PLA but it was not very precise and we knew that this would be an issue. The channels need to be perfectly smooth and even to allow for the Core XY assembly to move properly. We were able to solve this issue by changing the shape of the channel to eliminate the need for support material. This change was discussed more previously.

The rest of our parts that could not be 3D printed were purchased off of Amazon. These parts included: motors, polished steel rods, threaded rods, polished steel pins, bearings, belts, pulleys, motor couplings, and screws. The polished steel rods, threaded rods, and belts could not be ordered to the exact length that we needed so we had to cut them to the correct size. All other parts were considered when making our CAD design. This meant they all fit without any alterations being needed.

Manufacturing First Prototype:

Once we had all of our parts 3D printed parts printed and all of the stuff that we ordered, it was time to put it together. We started by assembling the linear actuator. The motor fit perfectly into the side of the linear actuator frame. The holes for the motor mounting screws were slightly misaligned but we were able to drill these holes out and use washers to make up for the slightly larger holes. The next step was to attach the threaded rod to the motor using the shaft coupling. Both shafts fit perfectly into the coupling. The next step was to thread on the print bed assembly. A 3mm nut was press fit into the print bed assembly to allow it to thread onto the threaded rod properly and to move smoothly. The next step was to insert the polished steel rod and attach the fourth side of the linear actuator frame. Both the threaded rod and polished steel rod fit perfectly into the indents on the fourth side and they were perfectly lined up. This meant both rods were perfectly parallel and the print bed was able to move very smoothly.

The Core XY assembly was put together next. We started by inserting the two polished rods into one of the side rail carts, then sliding on the center cart, and then putting the second side rail cart on the other side of the polished rods. This assembly was then fixed onto the Core XY frame with bearings that slide into the channels in the frame. The next step was to install this assembly onto the linear actuator. This was done using four screws for now. In the future the Core XY frame will be mounted onto two rotating pins with two locking pins. The next step was to install the two motors. The motor mounting screw holes were missing on the motor brackets but we were able to just drill these holes. The next step was to install all of the bearings for the pulleys. Some of the holes needed to be drilled out to make them perfectly circular but other than that all of the bearings press fitted perfectly into the brackets. Next was to install the pulleys. Half of the pulleys were mounted with the teeth closests to the frame and the other half were mounted with the teeth away from the frame. This was to allow the belts to cross each other without rubbing or colliding. All of the pulleys fit great except for one of the ones installed on a motor. The problem we ran into was because we flipped the pulley, the motor shaft was too short to reach the end of the pulley where the set screws were located. We have not corrected this problem yet but we plan to drill and tap two holes in the teeth of the pulley for the set screws. The final step was to install the two belts. This was also relatively easy, ensuring to run the belt on the correct orientation around the proper pulleys.

Once everything was assembled, we started moving the center cart side to side and up and down to make sure everything moved properly. The movement was not as smooth or as easy as we would have liked it to be but this was expected for our first prototype. This prototype allowed us to see that our vertical core xy design was possible and that we were on the right tract. We were also able to identify the changes that we will need to make to refine our design.

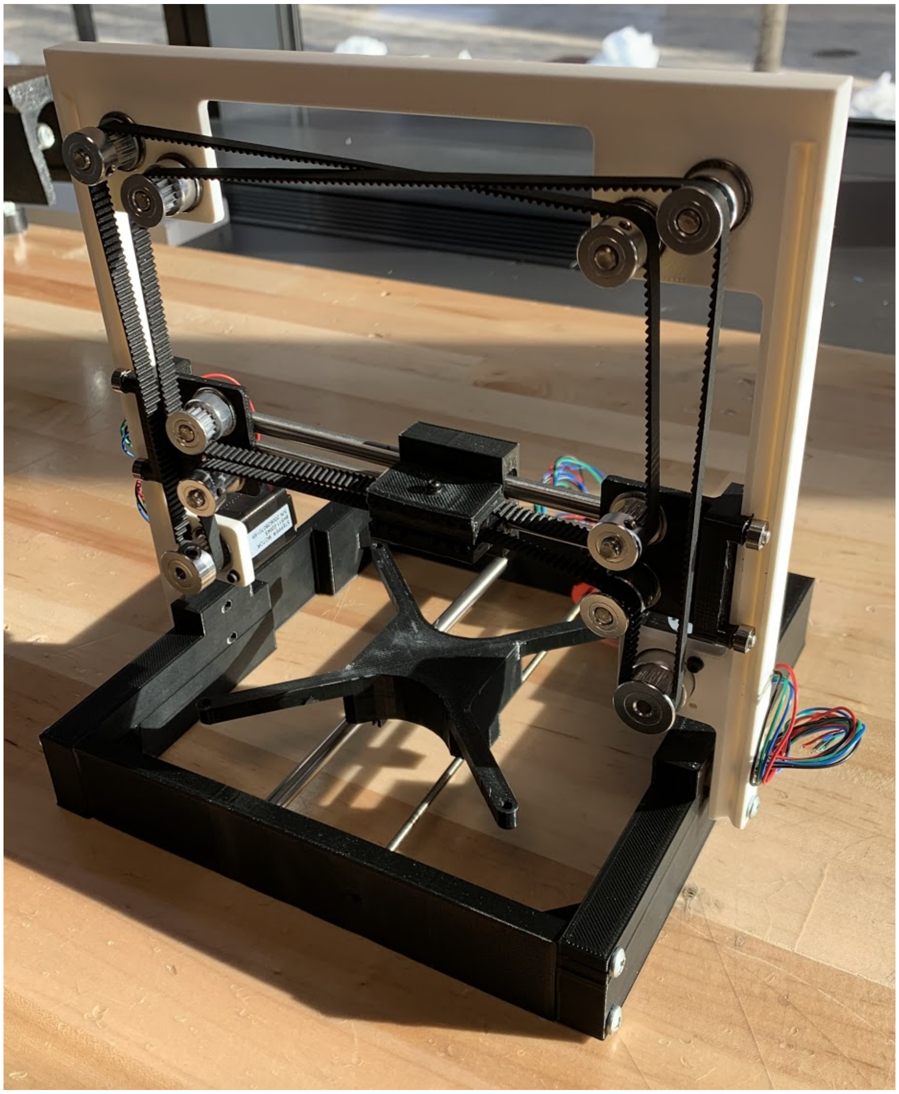


Figure 89: First assembled prototype

Improvements Needed From First Prototype:

The biggest issue that we found with our prototype is that our Core XY movement needs to be stiffened up. When we were moving the center cart by hand, it was very easy to move the two side carts and make them not parallel with each other anymore. This binded everything up. We also noticed that the pulley brackets flexed a lot when we moved it. The other issue that we noticed was that the side cart bearings did not ride in the side frame grooves like we had hoped.

The side carts themselves hugged the rail, keeping them in place. This meant that instead of riding on the bearings, the extrusions on the carts were just sliding on the edge of the frame.

To fix the issue with sturdiness, we first plan to add gussets to the bearing brackets. This will eliminate the flex that we were experiencing in the brackets. We also plan to add bosses to the brackets where the two polished rods are inserted. This will hopefully keep the two rods from being able to shift up and down, keeping the two side carts parallel with each other. We also made sure that these holes are as precise as we can make them to eliminate any slop.

Another problem we encountered assembling our first prototype was due to the tolerancing of our 3D printed parts. Once assembled, we found the carts that grabbed the sides of the Core XY frame were not riding on their bearing but were riding on the extrusions created to hold the pins for the bearing. Our initial solution to remedy this problem is to acquire larger bearings to be used for this component of the design. This allows us to maintain the extrusions for the pins so we don't have to sacrifice structural integrity. By enlarging the bearings used for these carts, we have to modify these parts to accompany this change. The first change is to add gussets to the brackets as stated above. The change we have to make also is to make these carts wider to ammpony to larger outer diameter of the bearings we are going to use.

Second/Final Prototype:

Improvements Made From First Prototype:

After determining many areas that needed to be improved on our first prototype, we immediately started on these improvements for our second prototype. This first issue that we focussed on was improving the stability of the entire design. We accomplished this by adding gussets to the Core XY motor and pulley mounts, as well as thickening the mounts. We also added gussets to the four corners of the linear actuator. We did not have an issue with stability within the linear actuator, but we felt that there was no harm in adding them. We also added strength to the linear actuator by shortening the depth of the cutout for the motors when folded. We noticed that the cutouts did not need to go the full height of the linear actuator in order to have enough clearance for the motors, so we were able to make the bottom of the cutouts the full thickness of the linear actuator. We also improved the stability of the Core XY guide rods by creating extrusions on the side carts to have the rods seated in 10 mm of material instead of 2 mm of material.

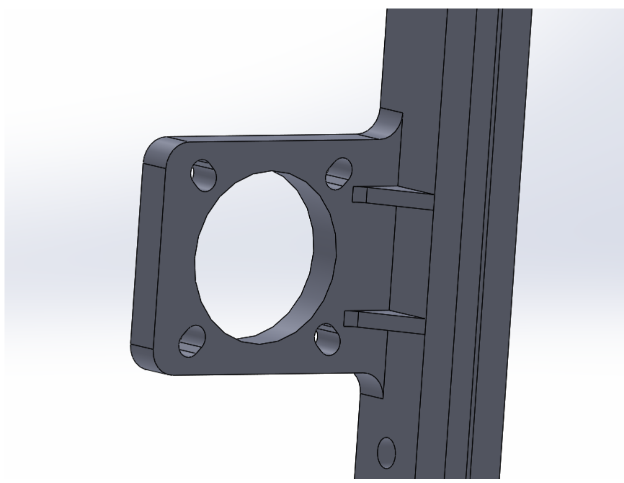


Figure 90: Improved motor mounts

The next issue that we focused on was the issue that we ran into with the motion of the Core XY carts in the Core XY frame rails. The issue was that the extrusions on the Core XY carts were riding on the sides of the rail instead of the bearings riding in the grooves of the rails. We fixed this by widening the carts slightly to give more clearance between the extrusions and the rail. We also had to make the grooves in the rail shallower to ensure that the bearings would still be able to contact them.

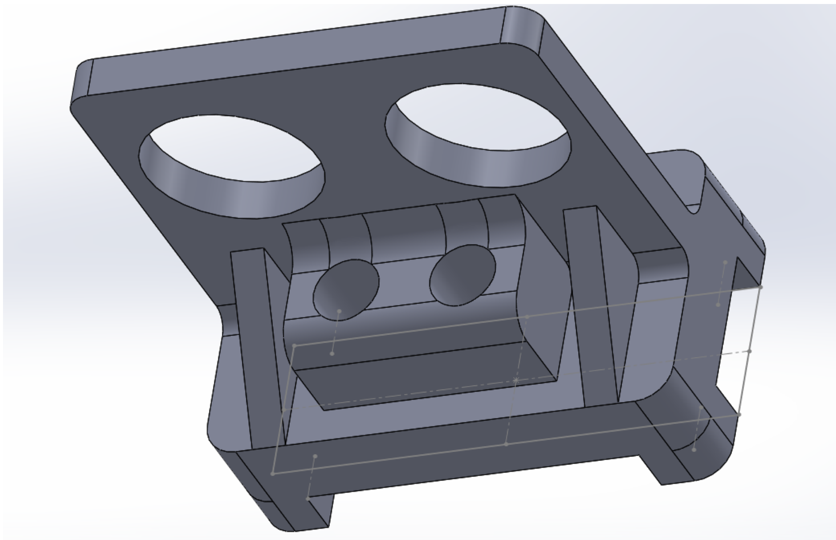


Figure 91: Improved bracket design

Once all of the issues were fixed, it was time to incorporate items that were not included in the first prototype. These items included: end stop mounts, filament extruder mount, and the filament feeder mount. Our design required one end stop on each axis, totalling three. The end stops had holes in them, so we decided that they could easily be mounted with screws. The only issue was that the end stops were not perfectly flat on the back side. The pins from each component extend out of the back of the board to allow them to be soldered. This meant that we couldn’t mount the end stops on a flat surface, but instead had to make extrusions to hold the end stops off of the surfaces. We mounted one end stop on the back of one of the sets of top pulleys on the Core XY frame, for the Z axis, another on one of the Core XY side carts for the Y axis, and one above the threaded rod on the linear actuator for the X axis. This required modification of these parts as well as the parts that will be contacting the end stops to ensure full contact at the desired position.

Mounting the filament extruder was also very simple. After doing research on the best methods of mourning the extruder, we found the best method to be simply clamping the top part. This was a very simple addition to make. However, we did have to modify the linear actuator frame to allow for clearance when the printer was folded. We had to create a cut out for the filament tube so that it would not get pinched or interfere with the folding of the printer. Next, we had to mount the filament feeder. The feeder assembly and motor was much bigger than we had hoped. We decided to mount the feeder assembly on the outside of the linear actuator, next to the linear actuator motor. We created a mount that sandwiched between the feeder and the feeder motor and then hooked over the edge of the linear actuator. Since the feeder assembly was much larger than we anticipated, it created an issue when it came to folding the printer. We had to increase the height of the Core XY assembly by 25 mm to allow enough clearance when the printer was folded.

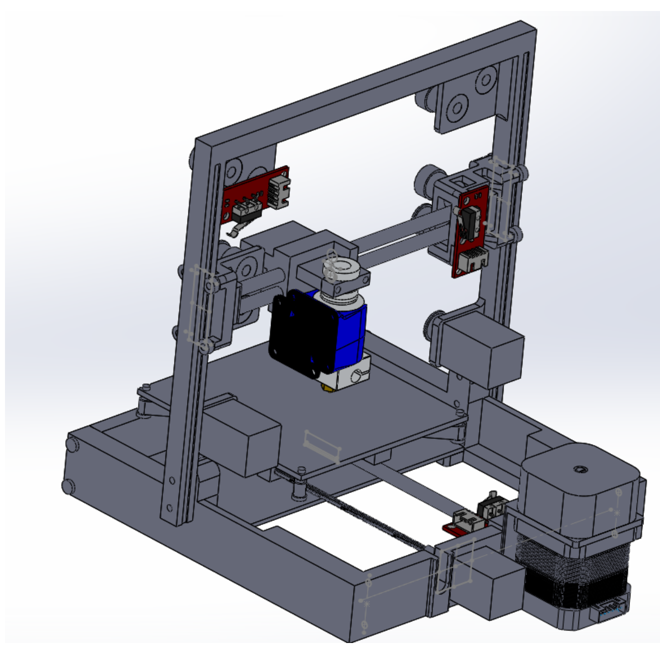


Figure 92: Improved printer design

Static Structural Analysis

Due to the small nature of certain parts in our design, a static structural analysis of those parts were made using ansys. This was due to the fact that there were certain stresses being applied to these parts that might have compromised their integrity. This was done for two parts, the first was the Core XY brackets that are held within the grooves on the Core XY frame. The second part that was subjected to a static structural analysis was the center bracket that held the belt tensioner.

The Core XY brackets are one of the smallest parts within our design. These brackets are held to the groove by bearings that act as wheels that are connected to pins within the bracket. They are able to maintain constant contact with the grooves due to the width of the brackets themselves being slightly smaller than the width of the grooves. The stress enacted on the part was calculated by finding the angular deformation of the part due to the differences in widths. Once that was calculated, the angular deformation was enacted on the part within ansys and by using the material properties of PLA the enacted stress on the part was found. The stress on these two brackets came out to be 24.496 MPa, this fell within the acceptable parameters so it was concluded that the part was able to withstand the strain enacted on it.

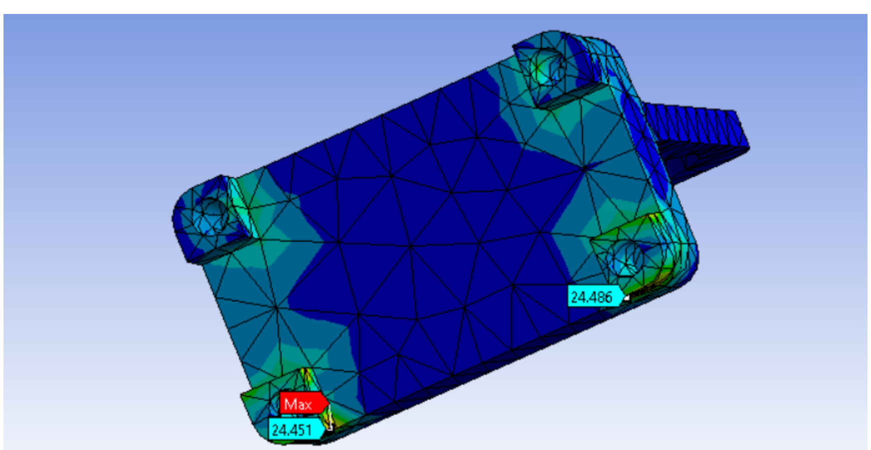


Figure 93: Core XY bracket analysis

The second part to undergo static structural analysis was the Center bracket. The stress that the belt tensioner portion of the bracket was simulated to make sure the essential operations of the printer were not too much strain on the part. This analysis was completed by calculating the force enacted on the belts due to the torque of each NEMA 8 motor during operation. That force was then placed on the center bracket in ansys where the tension would be most extreme on the part. It was found that the stress due to the tension of the bults was 2.98 MPa which was more than acceptable to maintain structural integrity within the part.

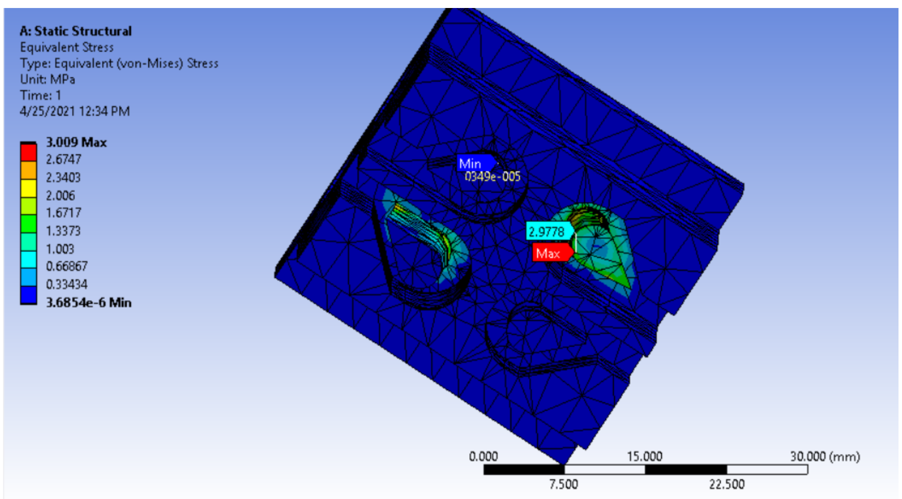


Figure 94: Center bracket analysis

Manufacturing Second Prototype:

Once all of the issues from the first prototype were fixed, and the additional changes were made, it was time to start manufacturing the second prototype. We were able to reuse all of the purchased parts such as the motors, pulleys, bearings, polished rods, etc from the first prototype. We ended up needing to reprint all of the 3D printed components because of all of the changes made. The second prototype was assembled exactly the same way as the first prototype. The linear actuator was assembled first, then the Core XY assembly was added, then the filament extruder and feeder were mounted. We were also able to successfully tap the teeth of the pulley that we had an issue with securing in the first prototype because of the pulley orientation and the motor shaft not being long enough.

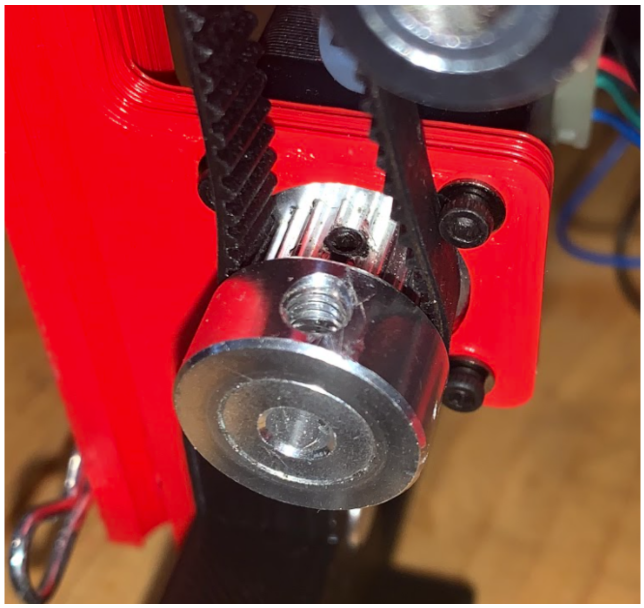


Figure 95: Successfully tapped pulley

We were very happy with the second prototype. With the changes made, the stability of the entire assembly was exactly what we were hoping for. The Core XY also moved very smoothly now that it was riding on the bearings instead of the cart extrusions. We were also able to install the filament extruder and feeder, as well as the end stops without any issues.

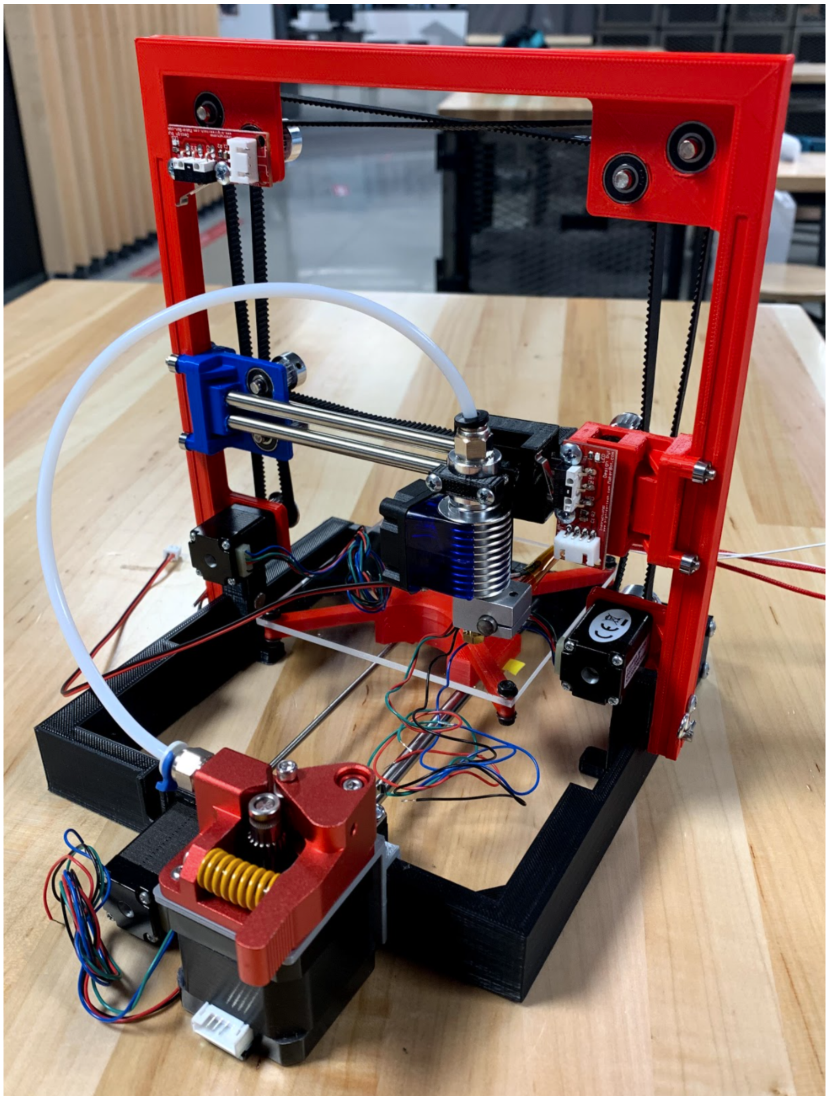


Figure 96: Second/final prototype in operating position

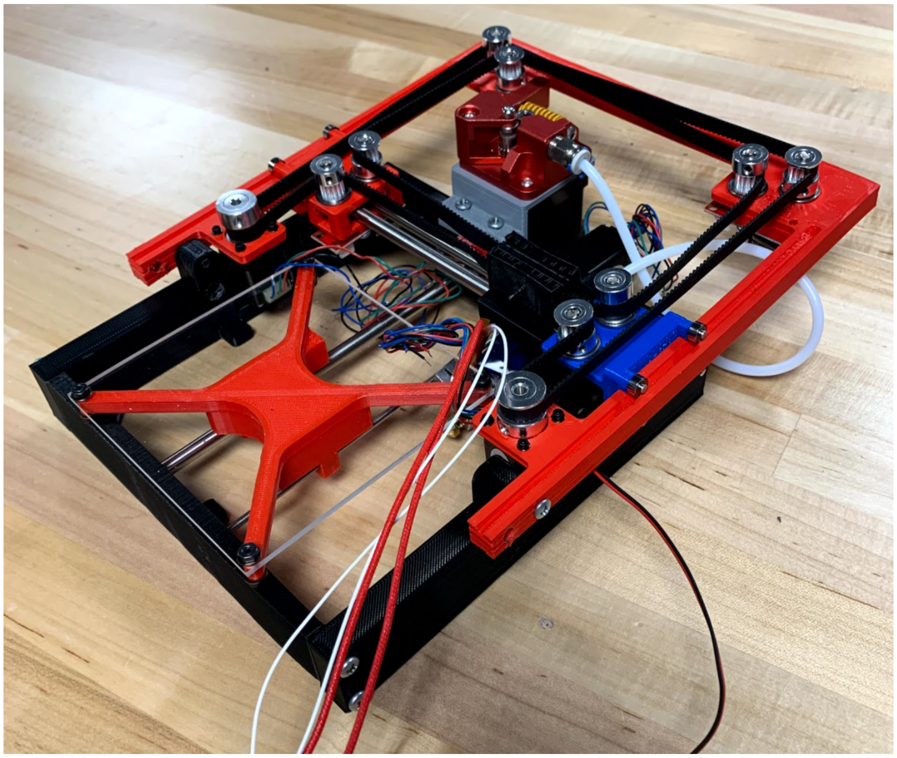


Figure 97: Second/final prototype in folded position

Future Considerations:

Manufacturing:

The first change that we would implement in the future is to 3D print our components using a material other than PLA. We found that our printed components were very brittle and this meant that we had to make a lot of changes in our design to account for this. The change in material would also improve the durability of the product because it would be more resilient to small impacts that it would most likely encounter when it is being transported. To make this change, we would recommend using a material such as ABS plastic for manufacturing parts. Also, while it may be slightly more expensive, using a dual extruder to combine materials for parts may be another good option in adding structural integrity. We would also recommend using a higher end 3D printer to print the components. We used a few different types of 3D printers over the course of this year and we found that the printer itself had a significant impact on the quality of the printed part. When we first started manufacturing parts, we were using a printer that was owned by one of our friends because it was readily available and because we had limited access to campus due to the pandemic. We had a lot of issues with our parts using this printer. It was very common for parts to be warped or for some areas to be misformed, as well as some parts would not be able to fully print without failing. Once we were able to access campus 3D printers, we had a significantly easier time printing parts and the quality was much better. We would also recommend looking into using 3D printers that have the capability to use dissolvable support material. For some of our parts, we had to change the design because once it was printed, we were unable to remove the support material without damaging the part. As for the soldering process, we worked with an inexpensive soldering system which led to inaccuracy and imprecision specifically while soldering the IC's. Also, we printed the circuit boards with a laser-cut machine provided by the university. We have encountered problems with copper traces detaching easily while soldering. It is best to print the circuit boards at a PCB manufacturing company for high quality circuit boards.

Components:

For our prototypes, we used a lot of purchased components such as bearings, pulleys, belts, motors, and electronics. Since our product is significantly smaller than a standard 3D printer, it was difficult to find components that were small enough for our design. The biggest issue that we ran into was with the feeder assembly. All of the feeder assemblies that we found available online utilized a Nema 17 stepper motor. For a standard 3D printer, this is a very common motor and is the appropriate size. However, in our design, this assembly is very bulky. After weighing the feeder assembly, we found that it was only about 50 grams less than the rest of the printer assembly. This is very significant especially since our goal is to produce a product that is compact and lightweight so that it can be easily transported in a standard backpack. We would recommend designing a feeder assembly that can utilize a much smaller stepper motor to reduce the bulkiness of this component.

References:

1. Anderson, T. (n.d.). *Anatomy of a 3D Printer: How Does a 3D Printer Work?* MatterHackers. Retrieved May 5, 2021, from /articles/anatomy-of-a-3d-printer
2. Bakker, B. (n.d). *How to Control a Stepper Motor with DRV8825 Driver and Arduino.* Retrieved May5, 2021, from https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/
3. *CoreXY | Cartesian Motion Platform*. (n.d.). Retrieved May 5, 2021, from https://corexy.com/theory.html
4. Corrigan, R. (2013). *Rostock Mini 3D Printer.* Retrieved May 5, 2021, from https://robot-kit.ru/wa-data/public/blog/download/3D_Printer_Delta_Rostock_mini_Final_Year_Report.pdf
5. Hopkins, T. (n.d). *Stepper Motor Driving.* AN235 Application Note. Retrieved May 5, 2021, from https://www.st.com/resource/en/application_note/cd00003774-stepper-motor-driving-stmicroelectronics.pdf
6. *How to build a DIY 3D Printer step by step at Home*. (n.d.). Retrieved May 5, 2021, from https://www.creativitybuzz.org/diy-3d-printer/#Step_1_3D_Printer_Parts
7. Liu, S., Liu, R., & Zhao, C. (n.d.). *Ramps 1.4 User's Manual*. Retrieved May5, 2021, from https://img.banggood.com/file/products/20141223204523SKU138520%20Ramps%201.4%20users%20manual.pdf
8. Mechanical endstop. (n.d.). Retrieved May 05, 2021, from https://reprap.com/wiki/Mechanical_Endstop

9. *Only $99 on Indiegogo Now: The PocketMaker Fits in Your Hand, 3D Prints Like a Pro - 3DPrint.com | The Voice of 3D Printing / Additive Manufacturing*. (n.d.). Retrieved May 5, 2021, from https://3dprint.com/155059/indiegogo-pocketmaker-3d-printer/
10. Peterson, Z. (2021). *What Size Decoupling Capacitor Should I Use for My Digital ICS?* Retrieved May 05, 2021, from https://resources.altium.com/p/what-size-decoupling-capacitor-should-i-use-my-digital-ics
11. *Pocket3DPrinter, Unique, Foldable, Pocket Size 3D Printer, Launches on Indiegogo for $299 - 3DPrint.com | The Voice of 3D Printing / Additive Manufacturing*. (n.d.). Retrieved May 5, 2021, from https://3dprint.com/18240/pocket3dprinter/
12. *Pololu - drv8825 stepper motor DRIVER Carrier, high current.* (n.d.). Retrieved May 5, 2021, from https://www.pololu.com/product/2133
13. RAMPS 1.4. (n.d.). Retrieved May 05, 2021, from https://reprap.org/wiki/RAMPS_1.4#Power_Supply.2C_Details_and_Considerations
14. Rice, R. (2018). *Stepper Motor Torque Basics.* Control Engineering. Retrieved May 5, 2021, from https://www.controleng.com/articles/stepper-motor-torque-basics/
15. Richter, A. (n.d.). *The Anatomy of a 3D Printer: Controller Boards*. MatterHackers. Retrieved May 5, 2021, from /articles/3d-printer-anatomy-controller-boards
16. *ROSENBERG, N. (2019). DESIGNING 3D PRINTERS: Essential knowledge. INDEPENDENTLY PUBLISHED.*
17. RS, G. (2015). *Stepper Motors and Drives, What is Full Step, Half Step and Microstepping?* DesignSpark. Retrieved May 5, 2021, from

https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping

18. Shet, J. (2019). *DRV8825 Stepper Motor Driver With Arduino Tutorial.* Iknowvations. Retrieved May 5, 2021, from https://iknowvations.in/arduino/drv8825-stepper-motor-driver-with-arduino-tutorial/
19. *The TOME, First Ever Portable, Self-contained 3D Printer - Coming soon for less than $1500 - 3DPrint.com | The Voice of 3D Printing / Additive Manufacturing*. (n.d.). Retrieved May 5, 2021, from https://3dprint.com/7462/tome-portable-3d-printer/
20. *What is 3D printing? How does a 3D printer work? Learn 3D printing*. (n.d.). 3D Printing. Retrieved May 5, 2021, from https://3dprinting.com/what-is-3d-printing/
21. *What is a current sense resistor? Different types, specifications, selection, and application guide for current sensor resistors*. (n.d.). Retrieved May 5, 2021, from https://components101.com/articles/what-is-current-sense-resistors-types-specifications-and-selection