UCSD Design/Build/Fly Foamcutter Manual

Yuting Huang
ythuang96@gmail.com

Updated:
December 6, 2018

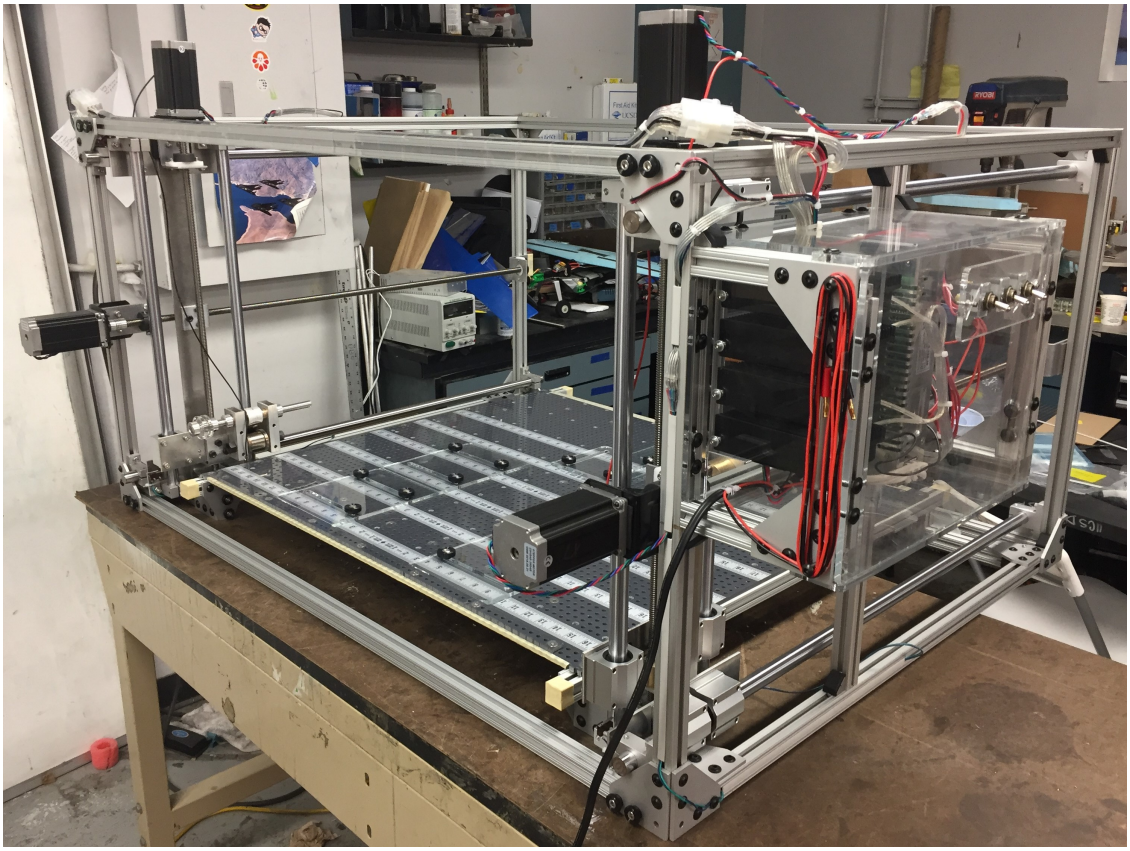
Preface

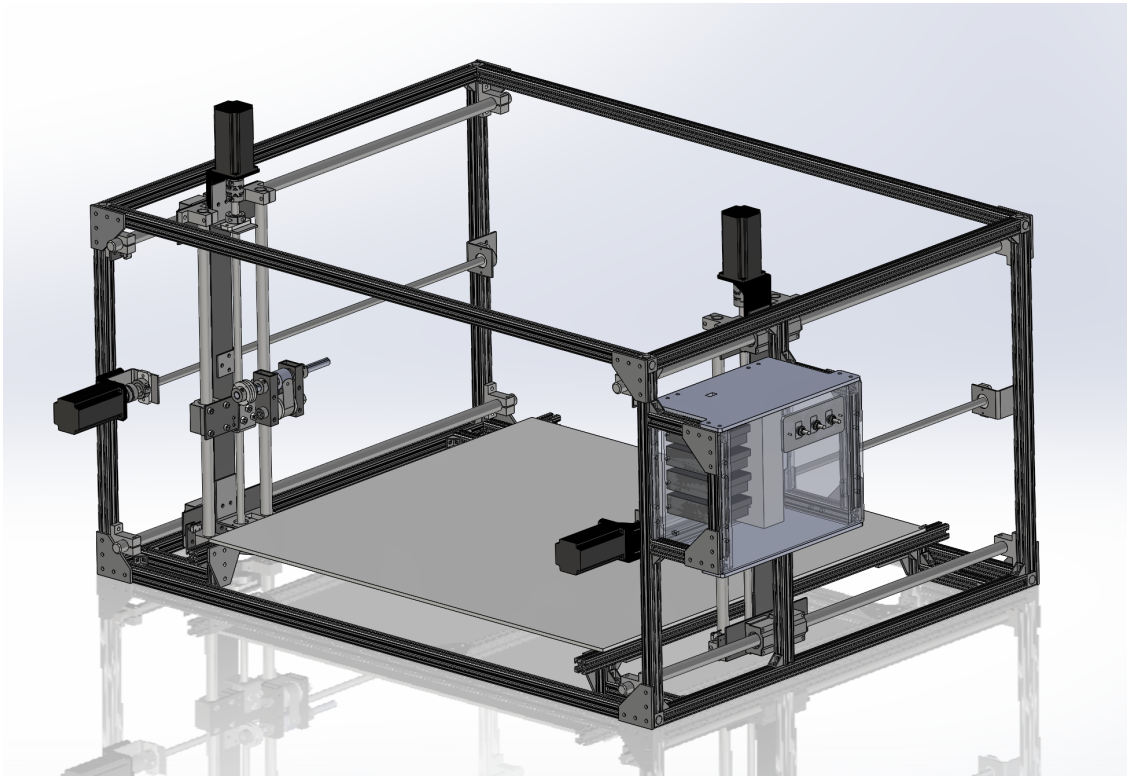
Before operating the foamcutter, please read this manual carefully to avoid injuries and damages to the equipment. All content in the text boxes are extremely important and should be given extra attention.

This foamcutter project was completed with funds from University of California San Diego Design/Build/Fly, and is intended to make the manufacturing process of composite planes easier.

All SolidWorks files, C code, MatLab scripts and operation manual are available for download at the author's Github page: https://github.com/ythuang96/FoamCutter. Please email the author at ythuang96@gmail.com to report any bug in code or any improvement suggestions.

This project was an improvement on the foamcutter design by Derek Ung and David Cruz. Special thanks to Dr. Mark Anderson, UCSD MAE department for support of the project. And thanks to Geo Lopez for his help during the project.





Contents

Chapter 1

Preparations

1 Laptop Setup

The foamcutter is controlled by a Raspberry Pi 3 B+. In order to communicate with the Pi, a Windows laptop has to install **PuTTY** (or similar) to enable secure shell (SSH) connection to the Pi, and **WinSCP** (or similar) for graphical file management on the Pi. A Linux laptop have SSH capability built in, therefore, no software required.

> Note:
>
> This setup process is only required once per laptop. If already performed, skip this section and continue to section 4.

1.1 PuTTY Setup

Please follow the following steps for install and setup of PuTTY:

1. Download PuTTY at https://www.putty.org/ then install.
2. Launch PuTTY and a window similar to fig 1.1 below should show up:

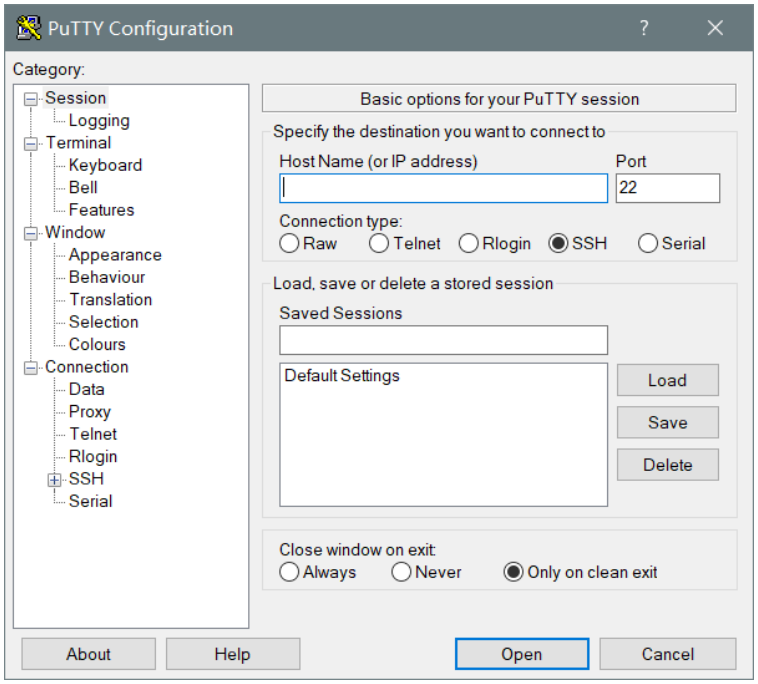


Figure 1.1: PuTTY Launch Window

3. As shown in fig 1.2 below, in the “Host Name” field, enter “foamcutter.local", then make sure Connection type is SSH. Next, in “Saved Session field”, enter “foamcutter", and lastly, click save.

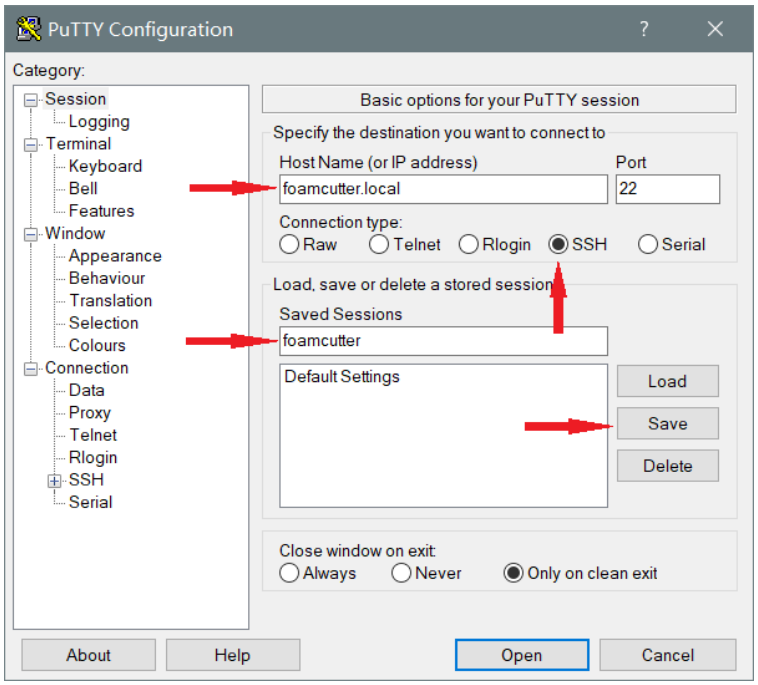


Figure 1.2: PuTTY Setup

4. The session with name “foamcutter" should show up in Saved Session as shown in

fig 1.3 below, and the PuTTY setup is completed.

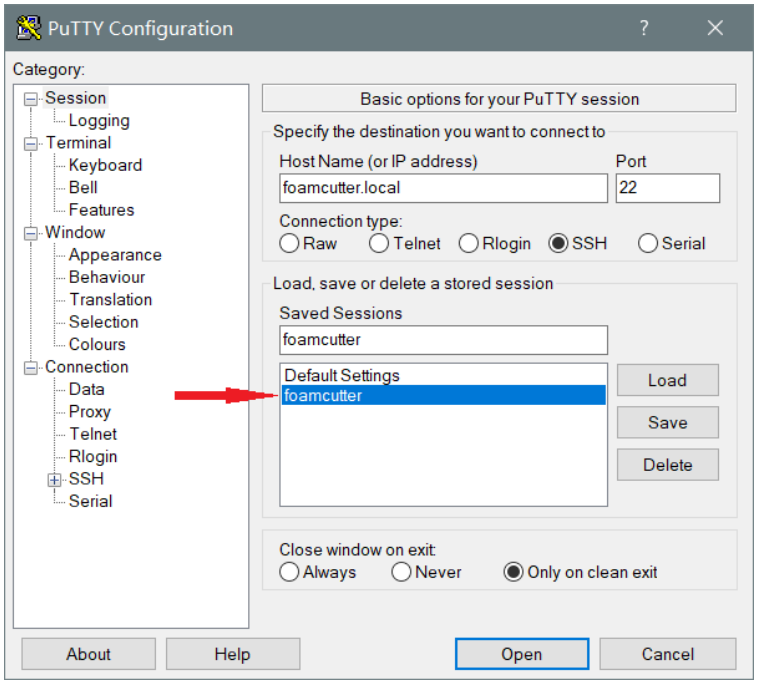


Figure 1.3: Saved PuTTY Session

5. Note on using PuTTY: to copy text from PuTTY, simply select the text, do not press "Control + C". To paste text to PuTTY, simply right click mouse, do not press "Control + V".

1.2 WinSCP Setup

Please follow the following steps for install and setup of WinSCP:

1. Download WinSCP at https://winscp.net/eng/download.php then install.
2. Launch WinSCP and a window similar to fig 1.4 below should show up:

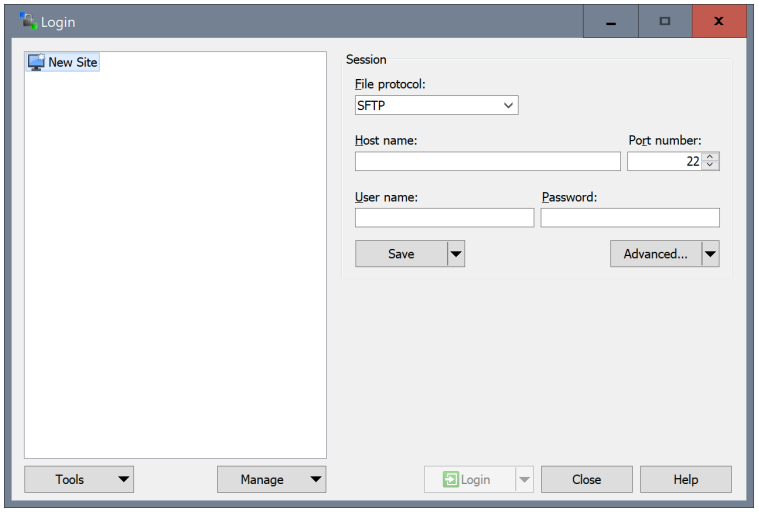


Figure 1.4: WinSCP Launch Window

3. As shown in fig 1.5 below, make sure the protocol is “SFTP", then enter “foamcutter.local" for the host name, “pi" for the user name, and “ucsdaiaadbf" for the password.

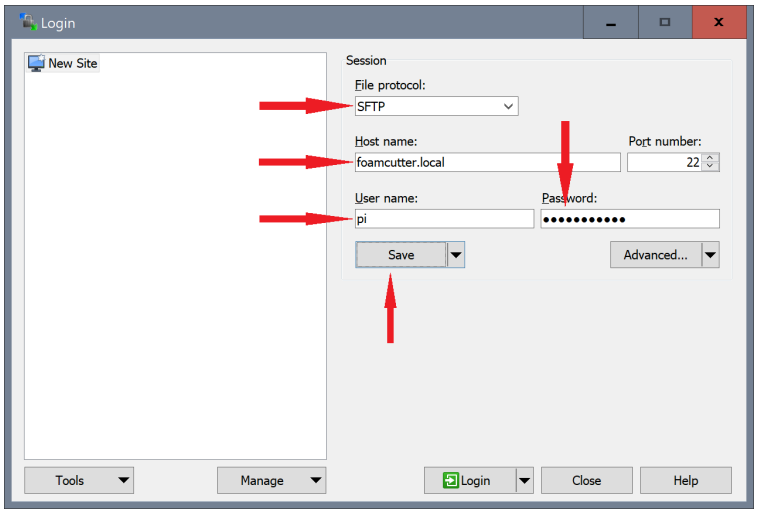


Figure 1.5: WinSCP Setup

4. Click save, and a window shown in fig 1.6 below should pop up. Both “Save password" and “Create desktop shortcut" are recommended. Click “OK" and the setup for WinSCP is completed.

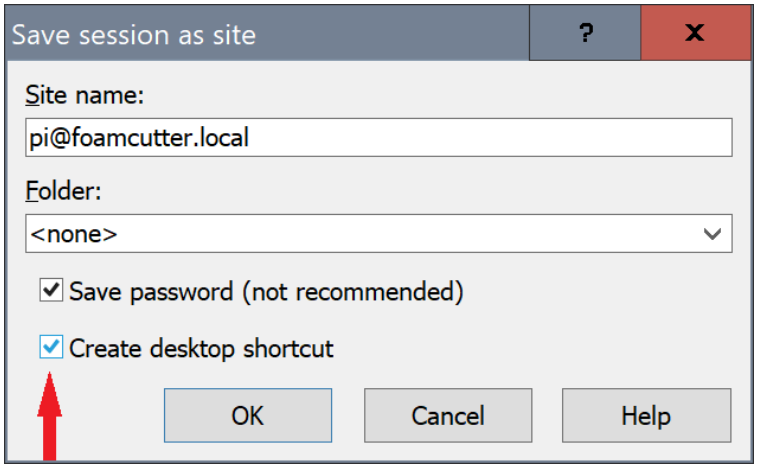


Figure 1.6: WinSCP Setup continued

2 Assemble the Foamcutter

The entire foamcutter as shown in fig 1.7 below consists of two side frames each having two axis of motion, four 4-feet long connection rods, and a work panel for securing the foam.

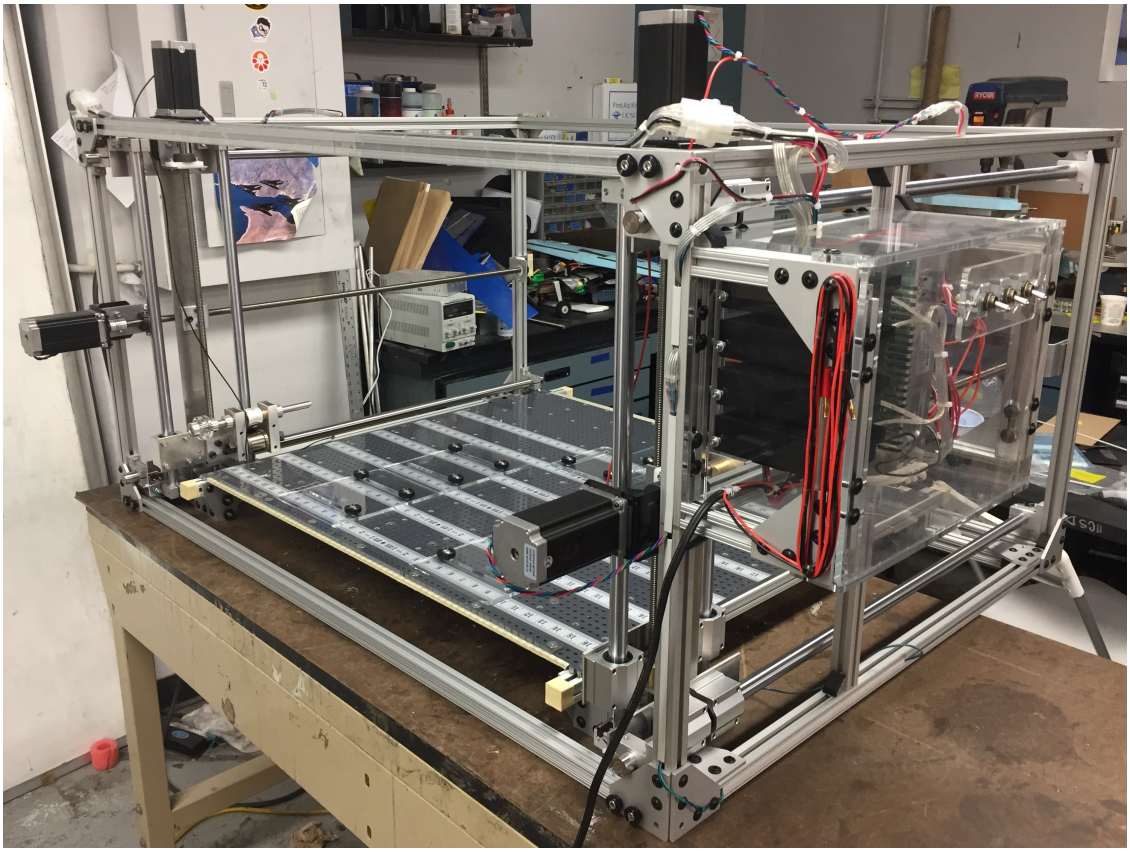


Figure 1.7: Foamcutter Overview

Caution:

- Side frames are extremely heavy, lift with caution;
- Avoid bumping head onto the connection rods when leaning into and out of the frame.

Do:

- Always make sure the two side frames are connected with the connection rods;
- Always push the two side frames firmly against each other before tightening the thumb screws;
- Remember to plug in the two Molex connectors.

Don't:

- Never let the side frames stand on their own;
- Never over tighten the thumb screws on the work panel;
- Never unscrew screws that are not thumb screws unless performing repairs.

2.1 Connect the two Side Frames

The two side frames are connected with 4 connection rods, use the 4ft long connection rods for performing cutting and use the 2ft long connection rods (not purchased yet) for storage of the foamcutter. Each of the 4 connection rods are secured with two thumb screws at each end shown in fig 1.8, loosen the thumb screws to remove the connection rod.

Figure 1.8: Thumb Screws for Connection Rods

During the installation of the connection rods, do not let the side frames stand on their own. The side frames are capable of standing on their own, but are not designed to do so: deformation will occur and degrade the accuracy. When installing the connection rods, push the two side frames firmly against each other to make sure there are no gaps between the side frames and the connection rods. Then tighten the thumb screws. An Allen key is recommended to further tighten the thumb screws.

2.2 Connect two Molex Connector

There are two Molex connectors located at the two top corners of the foamcutter as shown in fig 1.9 below. Make sure to connect them before operating the foamcutter. The Molex connectors are foolproof, **do not force it when unable to plug in**.

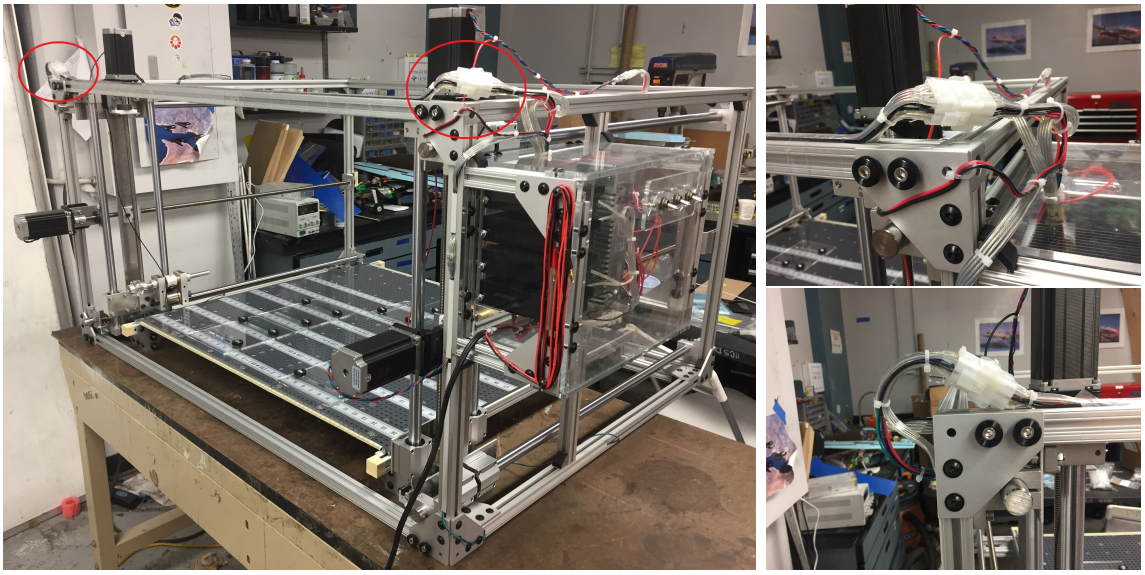


Figure 1.9: Molex Connectors

2.3 Install the Work Panel

The work panel is connected to the two side frames at the four corners of the panel. Each corner is secured with two thumb screws as shown in fig 1.10 below.

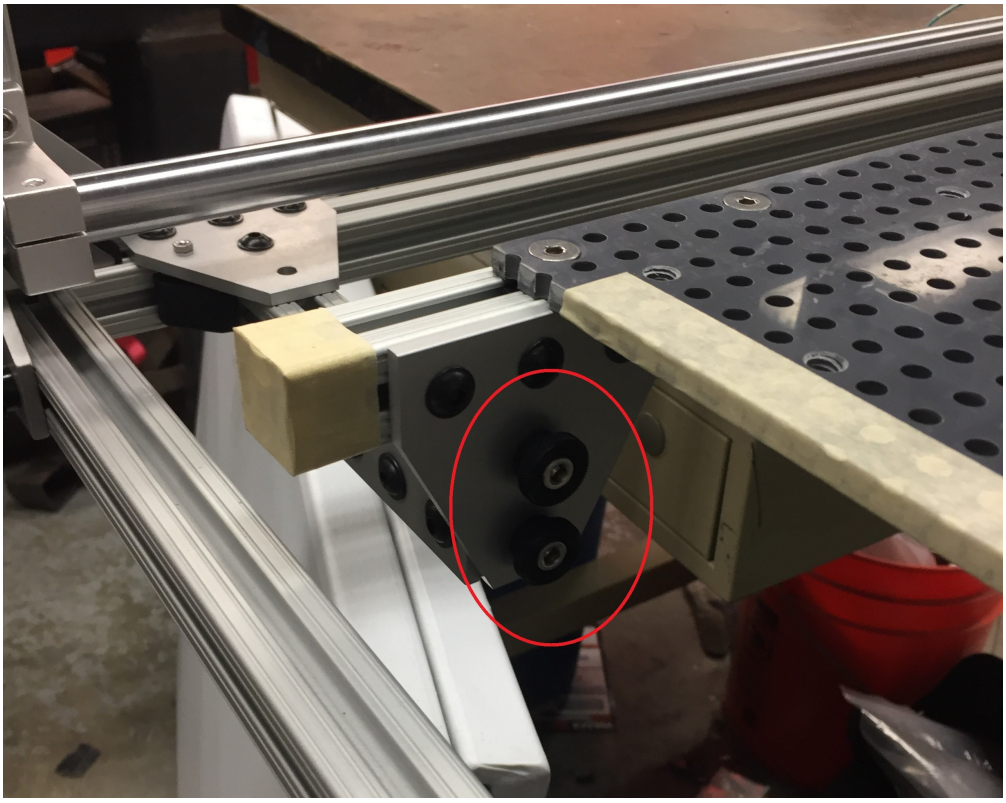


Figure 1.10: Work Panel Thumb Screws

The work panel should be oriented so that the guiding block, which is a long piece of PVC screwed to the work panel (pointed by the red arrows in the fig 1.11 below) should be at the closer side relative to the two horizontal motors (marked by the red circle in the fig 1.11 below).

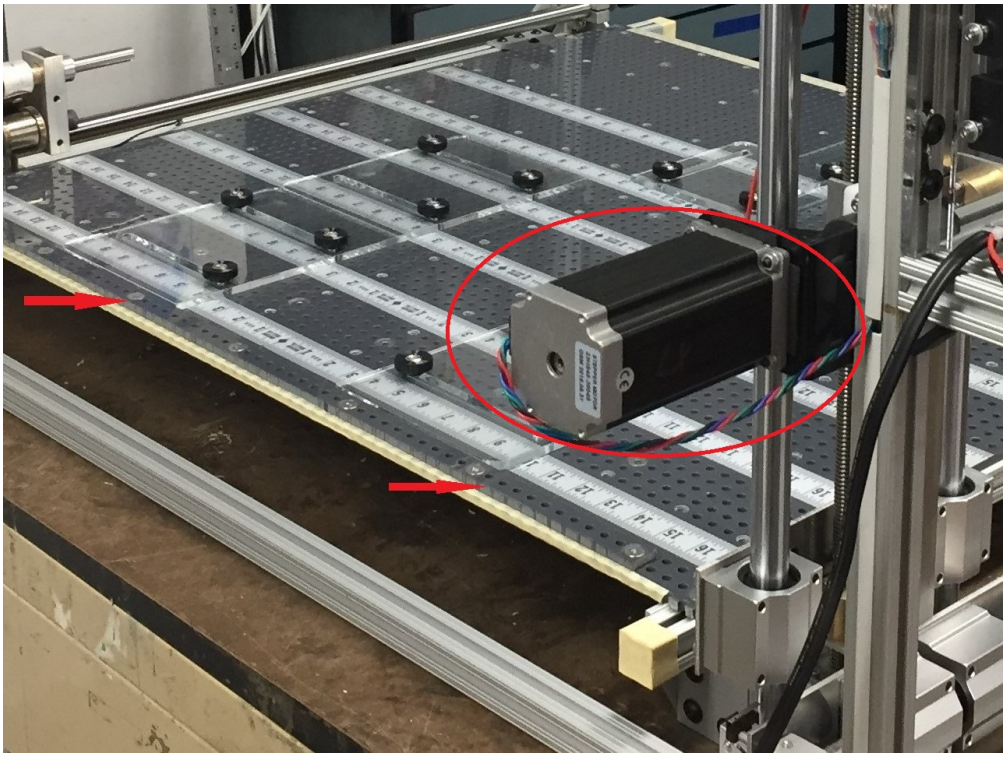
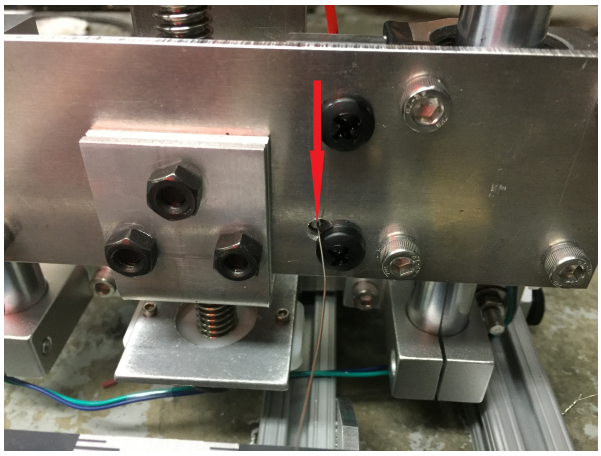


Figure 1.11: Work Panel Guiding Block Orientation

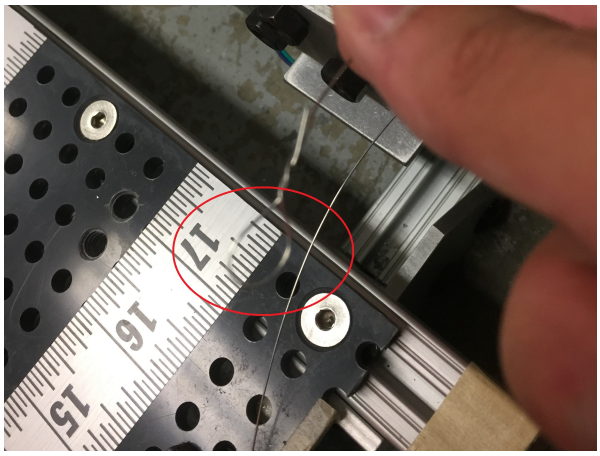
To install the work panel, first slide the work panel into the 4 supporting rods at the 4 corners with the correct orientation. Make sure all 4 corners sit securely in the supporting rods, then tighten all 8 thumb screws.

2.4 Install the Wire

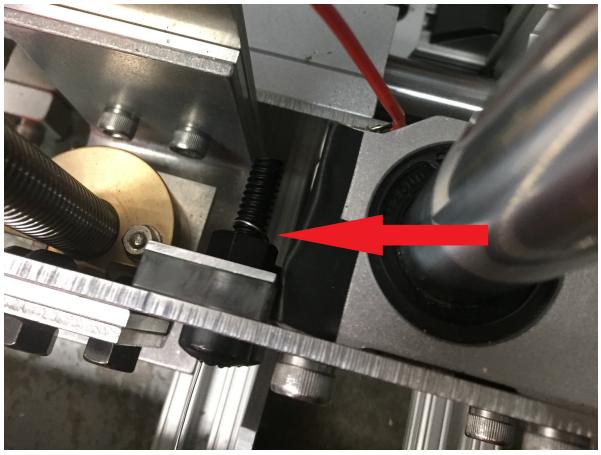
The wire is held to tension of approximately 10 lbs with a constant force spring. To install the wire, follow the steps shown below.



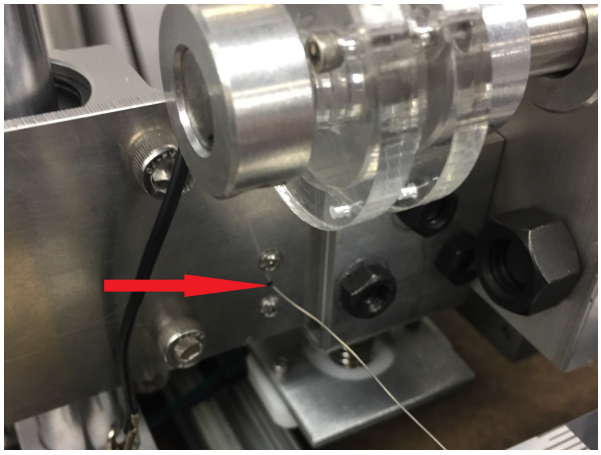
(a) Step 1



(b) Step 2



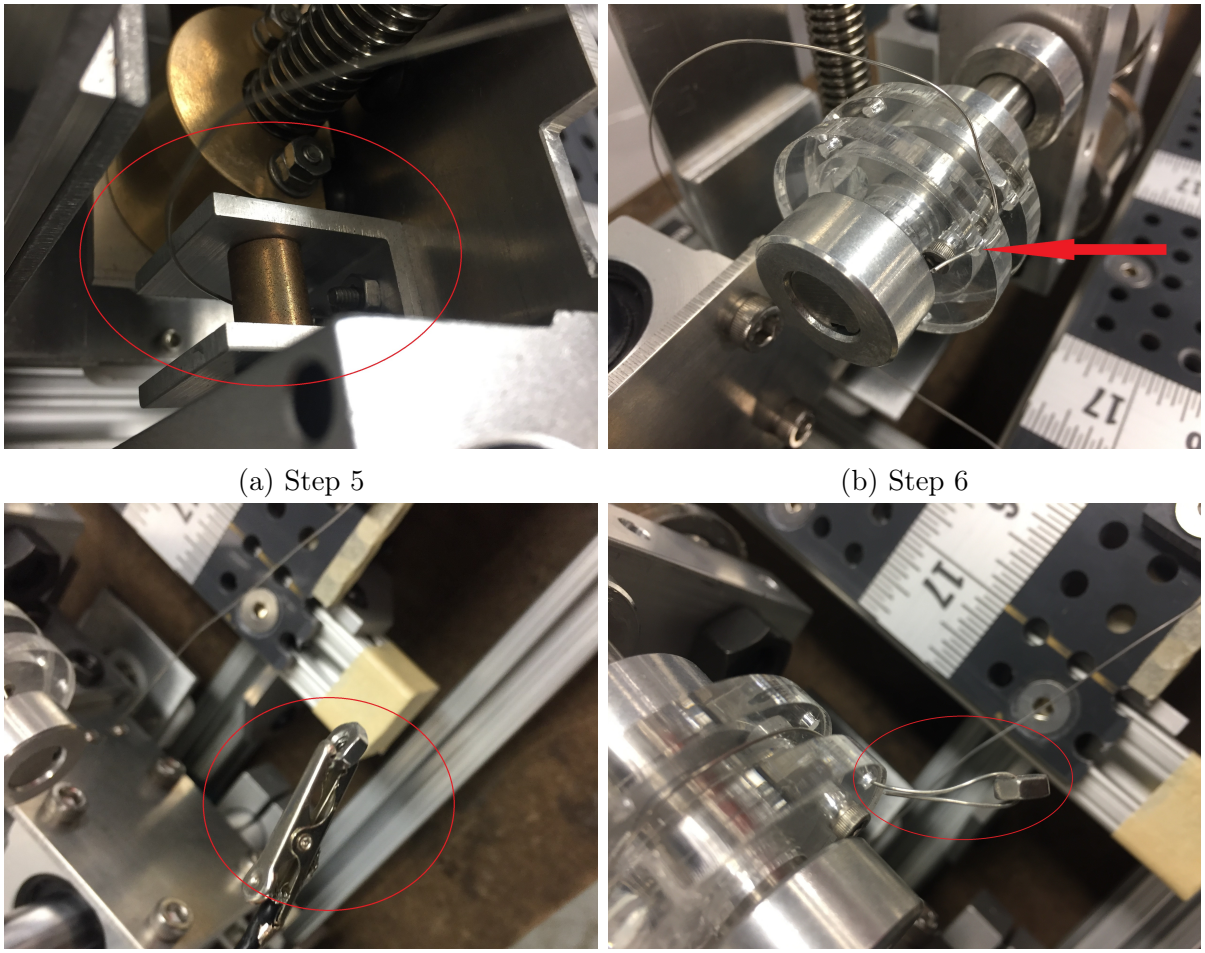
(c) Step 3



(d) Step 4

Figure 1.12: Steps 1-4

1. Feed the wire through the small guiding hole on the side without the pulley.
2. Make a hoop at the end of the wire.
3. Put the hoop around the Nylon bolt and straighten the wire across the entire span of the foamcutter.
4. Cut the wire at about 3ft longer than the span of the foamcutter. Then feed the end of the wire through the small guiding hole on the side with the pulley.



(a) Step 5

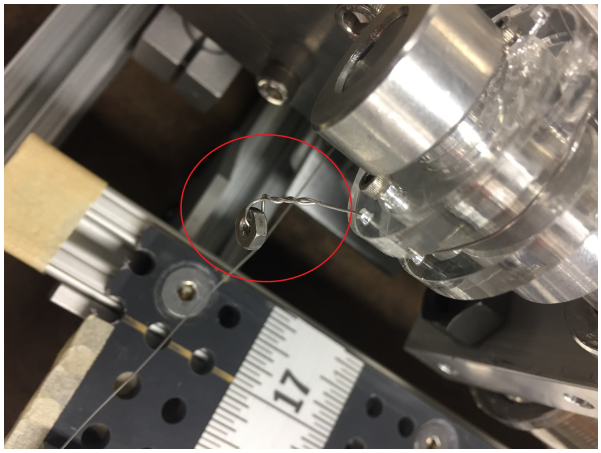
(b) Step 6

(c) Step 7

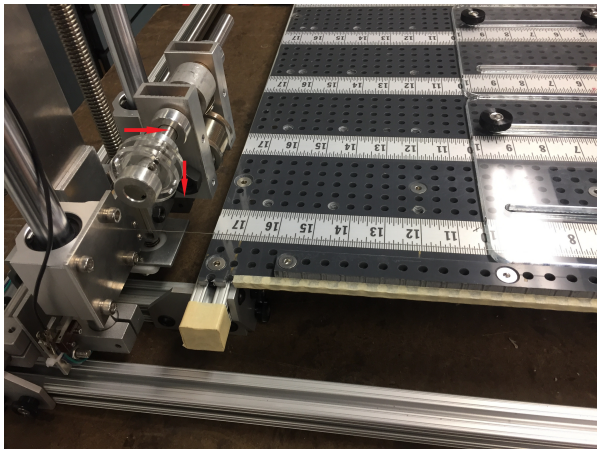
(d) Step 8

Figure 1.13: Steps 5-8

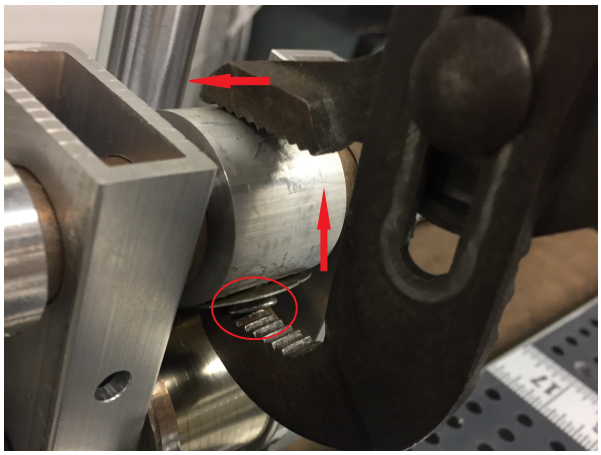
5. On the back side of the pulley, feed the wire around the copper sleeve bearing.
6. Feed the end of the wire through one of the small mounting holes on the acrylic pulley.
7. Remove the nut stored on the alligator clip.
8. Wrap the wire around the nut.



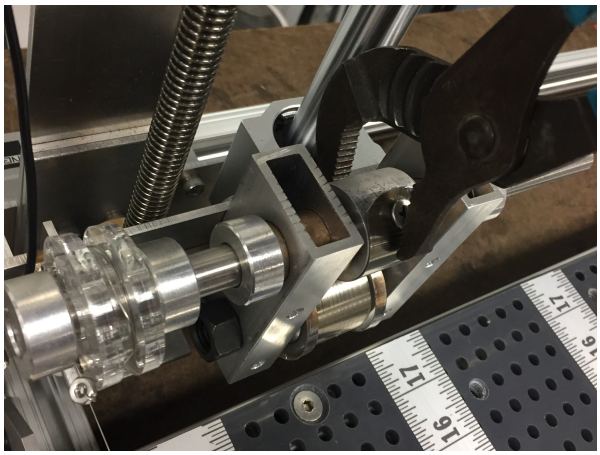
(a) Step 9



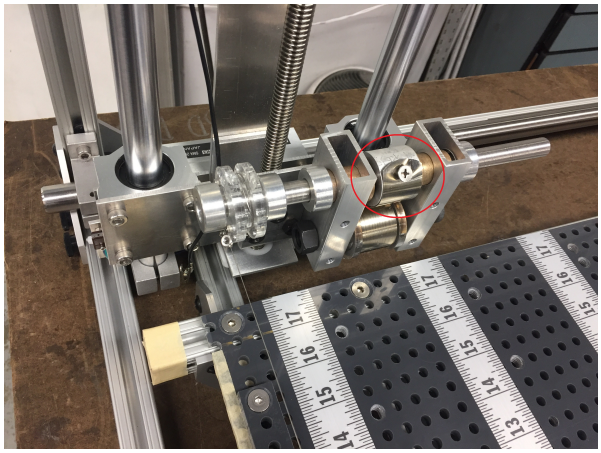
(b) Step 10



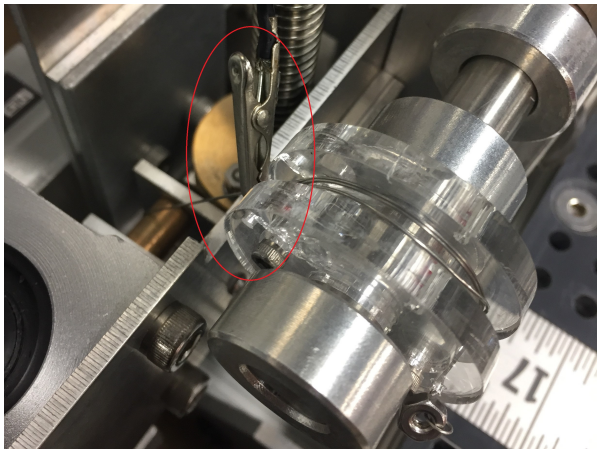
(c) Step 11



(d) Step 12



(e) Step 13



(f) Step 14

Figure 1.14: Steps 9-12

9. Twist the wire around the nut.
10. Rotate the acrylic pulley clockwise to tighten the wire.
11. Hold the acrylic pulley by hand. Use a wrench to further tighten the spring by holding

onto the mounting screw of the spring and rotate counterclockwise.

12. Further tighten the spring with the wrench.
13. Tighten the spring to the approximate position shown in the figure.
14. Clip the alligator click next to the pulley. On the other end, clip the other alligator clip on the back side of the guiding hole, and **make sure the alligator clip does not touch anything other than the hot wire.**

3 Foamcutter Software Setup

3.1 Connect to the Pi

1. Perform the Laptop setup guide in Chapter 1 Section 1 if not already completed.
2. Plug in the power cord to the foamcutter to power on the Pi.
3. Connect to the WiFi hotspot named **"foamcutter"** with password **"ucsdaiaadbf"** or connect to the ethernet port to the ethernet port of the laptop with a ethernet cable.
4. SSH into the Pi by opening PuTTY and double click on the saved session named **"foamcutter"**. Enter the user name **"pi"** then password **"ucsdaiaadbf"**.
5. Open WinSCP, a window similar to fig 1.15 below should show up. The left hand side are folders and files on your computer, while the right hand side is files on the Pi. Make sure the right hand side is in the folder **"/home/pi"**, if not, navigate to it. To create new folder on the Pi, use the **"new directory"** button. To upload files from your computer to the Pi, simply drag the file/folder from the left half to the right half.

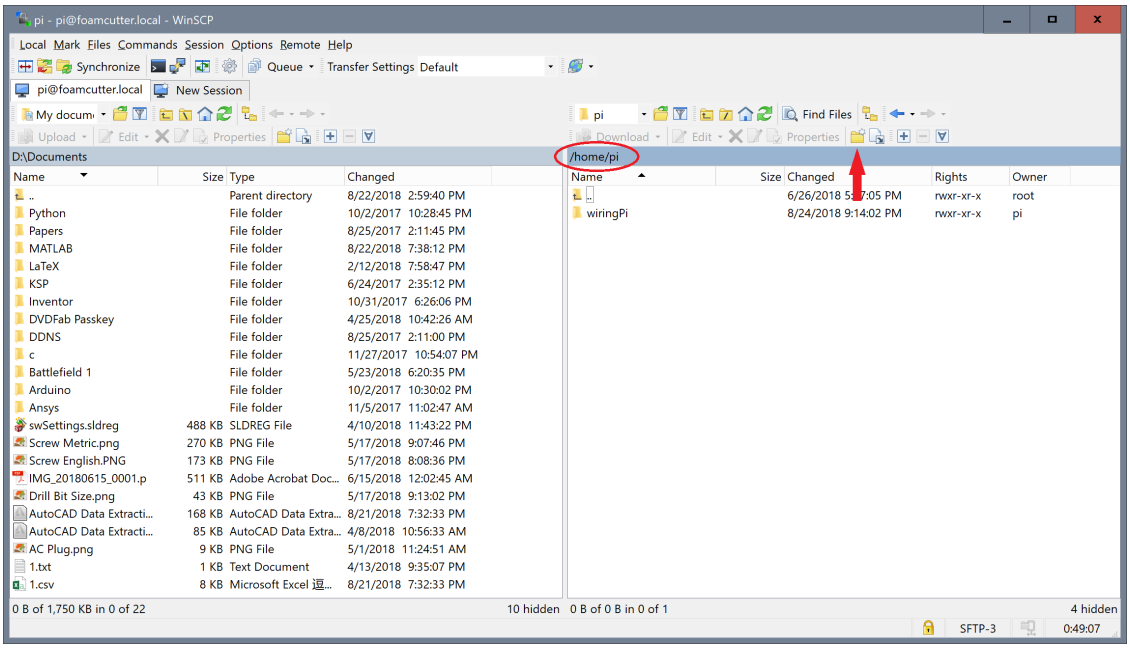


Figure 1.15: WinSCP Window

3.2 Calibrate Wire Origin

The 4 axes of the foamcutter each have one limit switch installed. Upon starting of the program, the foamcutter moves all axes towards the negative direction until hitting the limit switches. Then the foamcutter moves the 4 axes each a distance specified in the setting file to reach the Origin. **This calibration only needs to be performed once in a while to improve accuracy.** Please follow the steps to calibrate the wire origin:

1. Download the **"User Package.zip"** from https://github.com/ythuang96/FoamCutter, and unzip it.
2. Create a new folder on the Pi called **"foamcutter"** in the directory of **"/home/pi"**.
3. Upload the files named **"foamcutter.c"**, **"foamcutter_setup.h"** and **"Makefile"** from the **"User Package"** just downloaded into the **"foamcutter"** folder just created on the Pi.
4. Open PuTTY and run the following commands line by line:
   - **cd**
   - **cd foamcutter**
   - **make foamcutter**
   - **sudo ./foamcutter**

   This will launch the foamcutter program.
5. When the program reads **"I see the system is not homed yet, please press ENTER to home the system:"** press the Enter key. The foamcutter will move all 4 axes towards the negative direction until touching the limit switch. Then all 4 axes will move towards the positive direction to the origin.
6. Place a Aluminum block on the work panel against the work panel guiding block, and observe from top down as shown in fig 1.16 below. Both the far left and far right of the wire should sit right against the aluminum block, measure the amount that needs to be adjusted in millimeters.

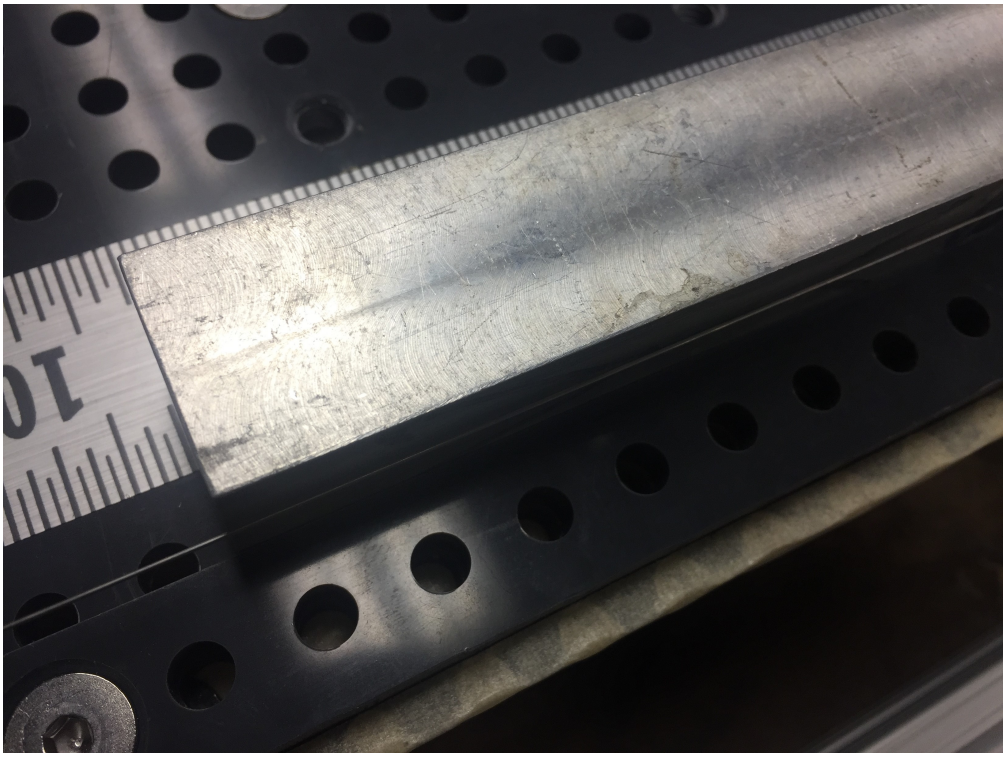


Figure 1.16: Calibrate Wire Origin in the Horizontal Direction

7. Again, place a Aluminum block on the work panel against the work panel guiding block, but observe horizontally as shown in fig 1.17 below. Both the far left and the far right of the wire should sit at the same height. The height could be arbitrary, but should be higher than the thumb screws on the work panel. Again, measure the amount that needs to be adjusted in millimeters.

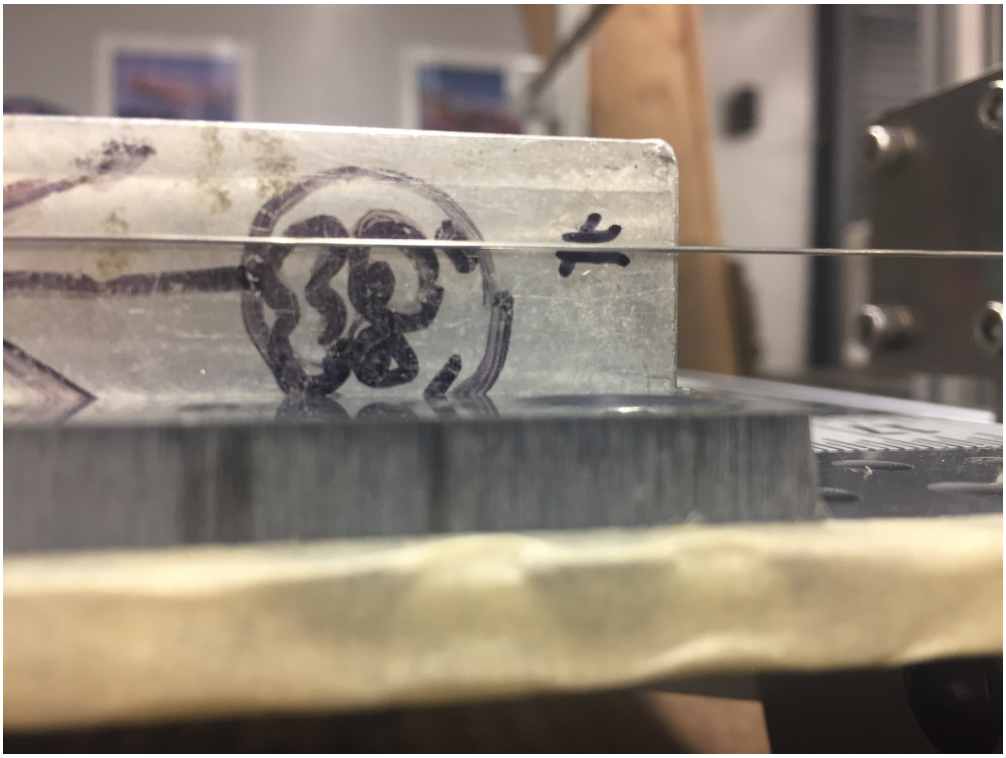


Figure 1.17: Calibrate Wire Origin in the Vertical Direction

8. Open WinSCP, navigate to **"/home/pi/foamcutter"** on the Pi (the right hand side of the WinSCP window), and double click on the file **"foamcutter_setup.h"** to edit it.
9. Find the 5 lines that read as the following:
   // **position of limit switch relative to cutter origin**
   #define LIM2ORIGIN_LX -4100
   #define LIM2ORIGIN_RX -3550
   #define LIM2ORIGIN_LY -2500
   #define LIM2ORIGIN_RY -2400
   and adjust the four numbers: if wish to move an axis 1mm further away (or closer) from the limit switch, subtract (or add) 157 from the number corresponding to the axis. **All 4 numbers must be negative integers, no decimal, and no equations (please calculated the number yourself and replace the original number).**
10. Save the file then repeat the process from step 4 to adjust the 4 numbers until the wire sits at the desired location. **Then save the 4 numbers on your computer.**

3.3 Install the Foamcutter Software

1. Perform the Wire Origin Calibration guide in Chapter 1 Section 3.2 if not already completed.
2. Open PuTTY and run the following commands line by line:
   - **cd**

- **cd foamcutter**
- **sudo make install foamcutter**

3. Delete the folder **"/home/pi/foamcutter"** on the Pi.
4. You can now use **sudo foamcutter** to run the foamcutter program.

3.4 Update the Foamcutter Software

1. If a new version is available, download the **"User Package.zip"** from https://github.com/ythuang96/FoamCutter, and unzip it.
2. Create a new folder on the Pi called **"foamcutter"** in the directory of **"/home/pi"**.
3. Upload the files named **"foamcutter.c"**, **"foamcutter_setup.h"** and **"Makefile"** from the **"User Package"** just downloaded into the **"foamcutter"** folder just created on the Pi.
4. - If wish to perform Wire Origin Calibration, proceed to Chapter 1 Section 3.2 step 4 to step 10. Then continue to the steps in Chapter 1 Section 3.3.
   - If wish to use the previously calibrated parameters, open WinSCP, edit the file **"/home/pi/foamcutter_setup.h"**, find the 5 lines that read as the following:
     // position of limit switch relative to cutter origin
     #define LIM2ORIGIN_LX -4100
     #define LIM2ORIGIN_RX -3550
     #define LIM2ORIGIN_LY -2500
     #define LIM2ORIGIN_RY -2400
     and change the 4 numbers to the previously calibrated parameters. Then proceed to Chapter 1 Section 3.3 step 2 to step 4.

4 G-Code Generation

4.1 G-Code for Wing

1. Download the needed airfoil coordinates at http://m-selig.ae.illinois.edu/ads/coord_database.html#S
2. Place the needed airfoil coordinate files in the same folder as the MatLab code **"DBF_foamcutter_wing.m"** which is provided in the **"User Package.zip"** from https://github.com/ythuang96/FoamCutter.
3. Enter the parameters into the MatLab script and run the script. A figure tracing the the motion of the two ends of the wire should show up, and the G-code should be saved in the same folder.

Note:

When cutting extremely tampered parts, figure similar to the fig 1.18 below might show up during the G-code generation process:

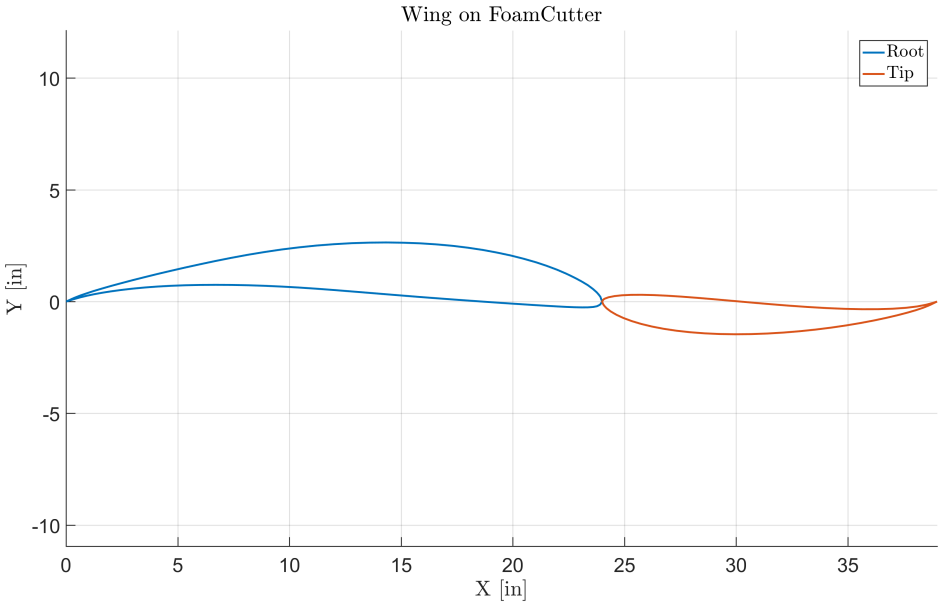


Figure 1.18: Example Figure During G-code Generation for a Wing

The parameters used for this figure is shown below:

```
%% Enter Parameters Below
% airfoil section file for root
root_filename = 'AH-79-100.dat';
% root chord length [in]
root_chord = 6;

% airfoil section file for tip
tip_filename = 'E216.dat';
% tip chord length [in]
tip_chord = 3;
% root chord has to be greater or equal to tip chord

% +1 for right wing, -1 for left wing
right_wing = 1;

% semi-span [in]
semi_span = 3;

% leading edge sweep [deg]
LE_sweep = 0;

% twist [deg]
twist = 0.0;

% g-code output file name
g_filename = 'HTail';

% width of CNC cutter [in]
cutter_width = 39;
% scale the cord length to accomodaate for broken trailing edge
% recommend using 1.2, then cut trailing edge with a blade to desired
% length.
scale_factor = 1.2;
```

Figure 1.19: Example Parameters During G-code Generation for a Wing

Note that the root and tip cord length differed a lot, and the span is significantly smaller than the cutter width, therefore, this is an extremely tapered cut. As a result, the inverted airfoil shown in fig 1.18 above is expected. As explained in fig 1.20 below, for extremely tapered cuts, the two ends of the wire could move in opposite directions, resulting in the inverted airfoil shown in fig 1.18.

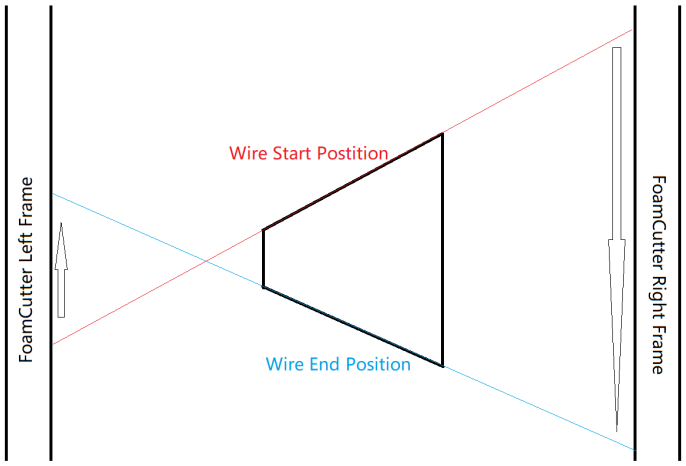


Figure 1.20: Sketch of an Extremely Tapered Cut

4.2 G-Code for General Shapes

Key Points:

- Make sure to increase AutoCAD precision;
- Make sure to scale the drawing by 25.4 in AutoCAD if the SolidWorks file is in inches;
- Make sure to run "Explode" command in AutoCAD;
- Make sure there are no overlapping lines;
- Make sure to convert spline to polyline in AutoCAD and explode again.

1. Export the projection of the top or side (or both) profiles from SolidWorks into .dxf files, the steps are demonstrated with the payload bay shown in fig 1.21 below.

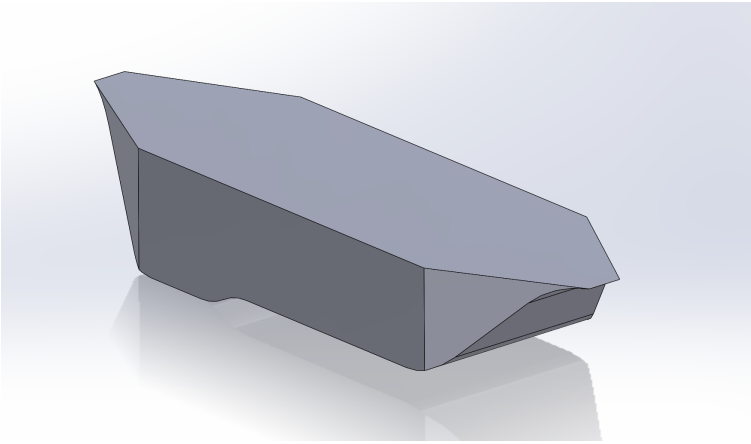


Figure 1.21: Sample Payload Bay

- The projection of the side profile can be obtained by slicing the part vertically in the center: go to "Insert → Features → Split" as shown in fig 1.22 below:

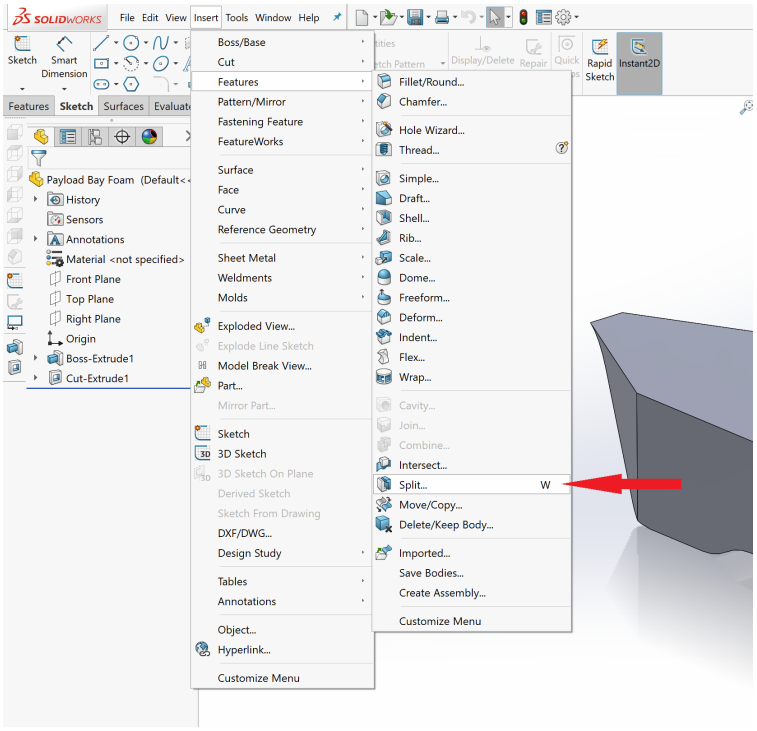


Figure 1.22: SolidWorks Split Command

- In the pop up window, as shown in fig 1.23 below, in "Trim Tools" selected the desired plane (the front plane in this case) then click "Cut Body". In the "Resulting Bodies" box, select one of the resulting bodies. Lastly, make sure the "Consume cut bodies" check box is selected, and click the top left green check mark. This will delete the selected half of the part (the brown half in fig 1.23).

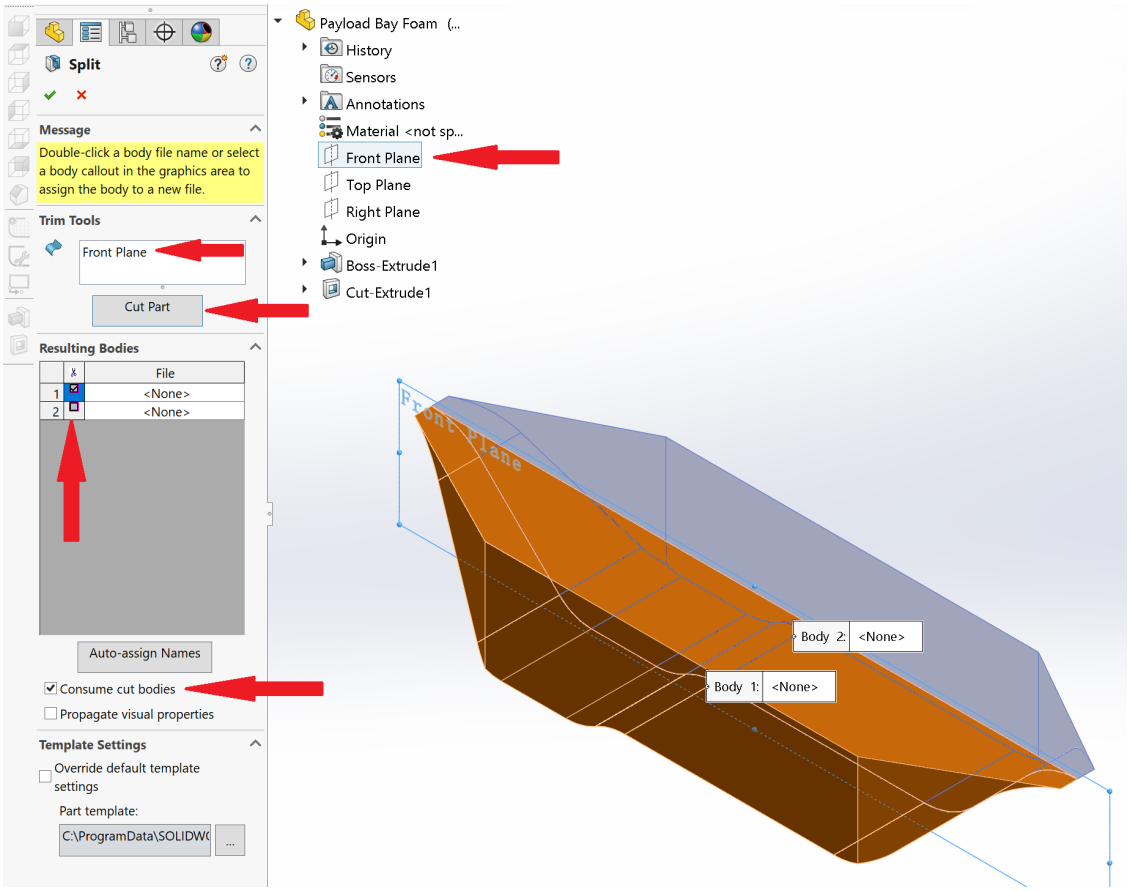


Figure 1.23: SolidWorks Split Menu

- Right click on the cut plane and select "Export to DXF/DEG" to export the projection of the side profile as shown in fig 1.24 below:

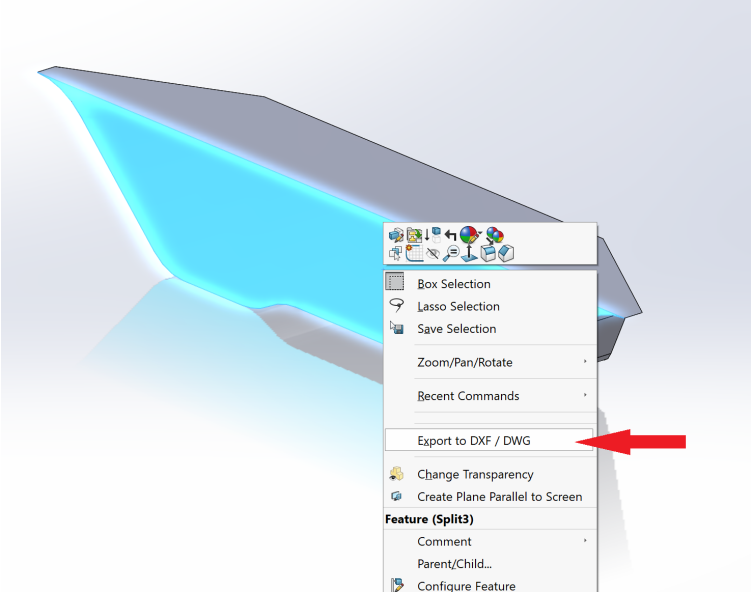


Figure 1.24: Export Side Profile

- The top plane of the part is the projection from above, therefore, skip the split command and directly right click on the plane and export:

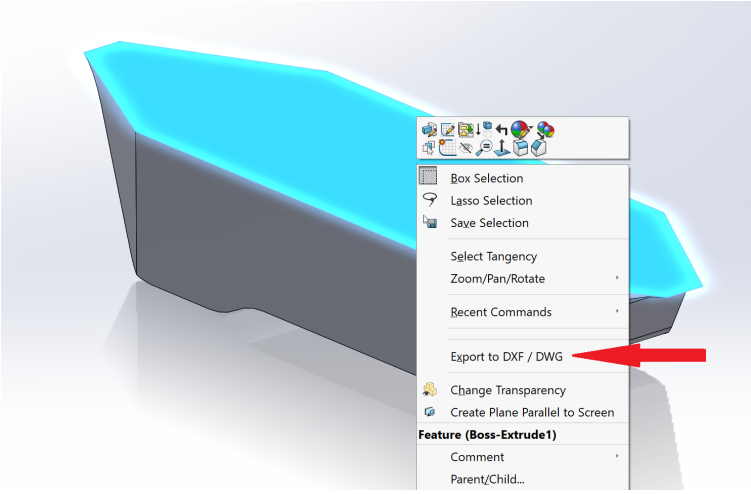


Figure 1.25: Export Top Profile

2. Open AutoCAD, increase the precision by clicking the top left red AutoCAD sign, select "Drawing Utilities → Units" and increase the precision to 4 decimal points, as shown in fig 1.26 below.

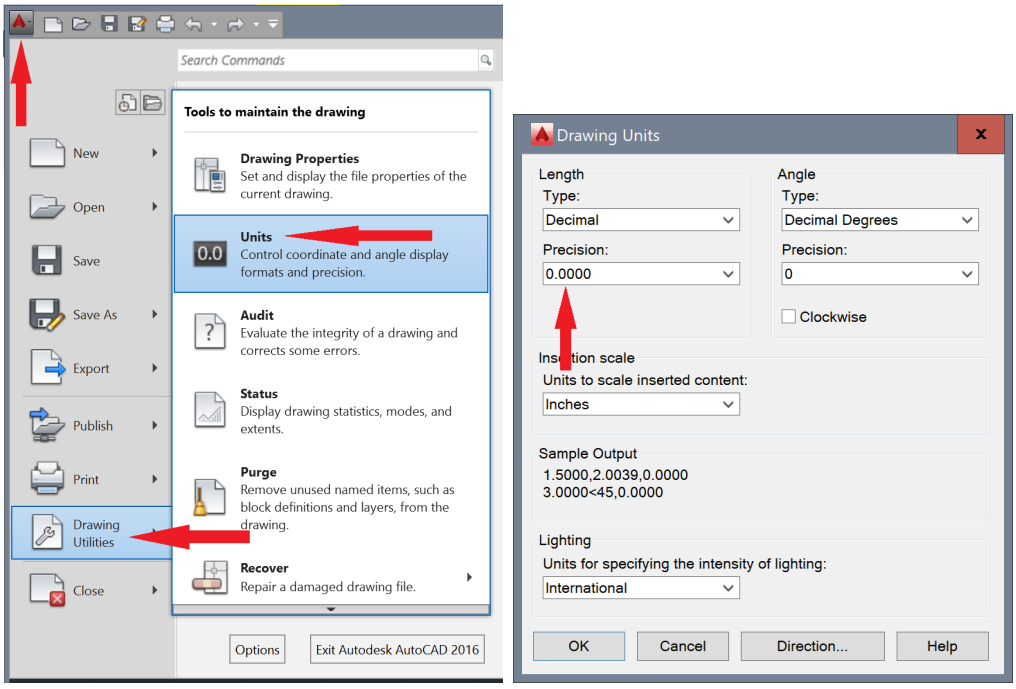


Figure 1.26: Increase AutoCAD Precision

3. Import the dxf files from SolidWorks.
4. Scale the drawing by 25.4 if the SolidWorks file is in inches.
5. Select all objects and run "Explode" command.
6. Select all Spline (if any), right click and select "Spline → Convert to Polyline" as shown in fig 1.27 below. When ask to enter accuracy, enter 1 and press enter.

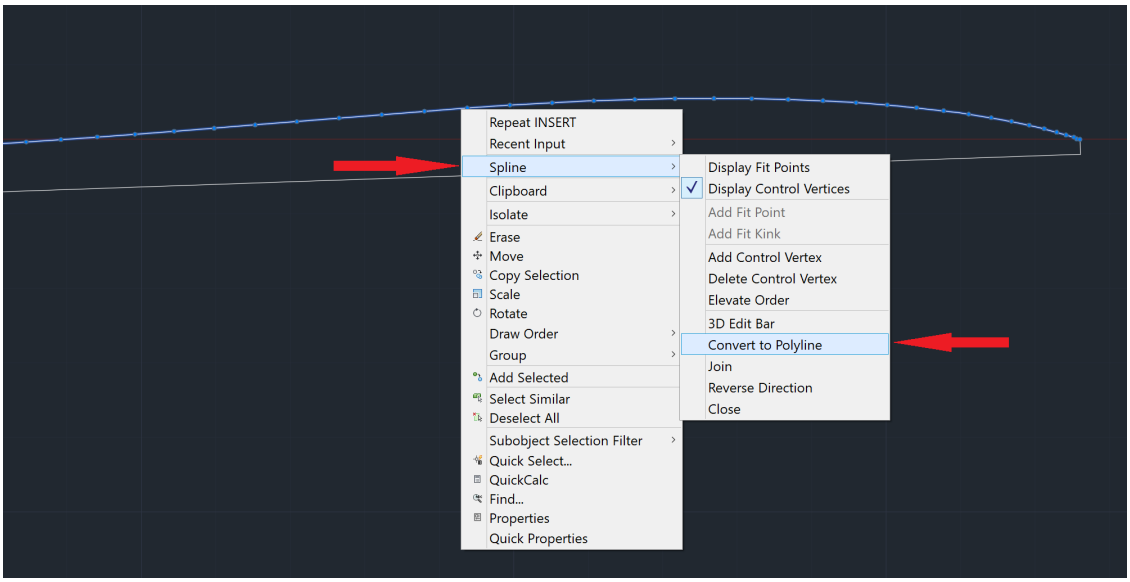


Figure 1.27: Convert Spline

7. Explode again (explode the polylines converted from spline).

8. Rotate and delete overlapping lines if necessary.
9. Select all lines and run "eattext" command, a window similar to fig 1.28 should pop up, select "Edit an existing data extraction" then click the browse button.

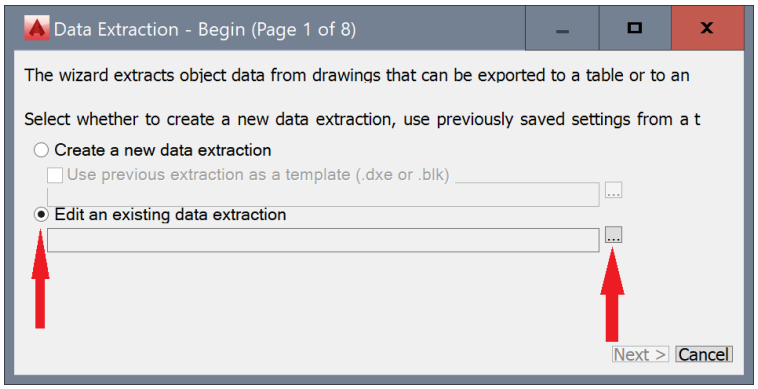


Figure 1.28: Eattext Page 1

10. Browse for the extraction templates, select the template with or without arc depending on the drawing. (The templates are included in the **"User Package.zip"** from https://github.com/ythuang96/FoamCutter)

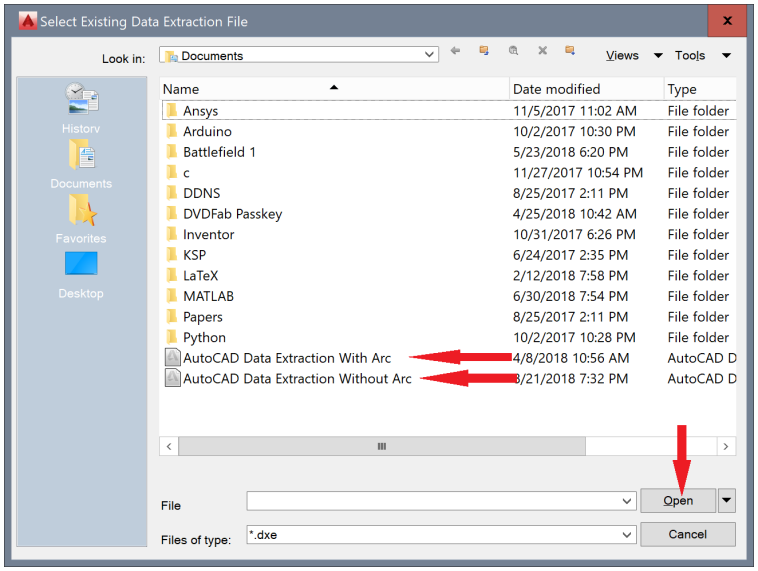


Figure 1.29: Browse for Data Extraction Templates

11. On page 2, check "include current drawing", select the one that is not the current drawing (if any) and click remove.

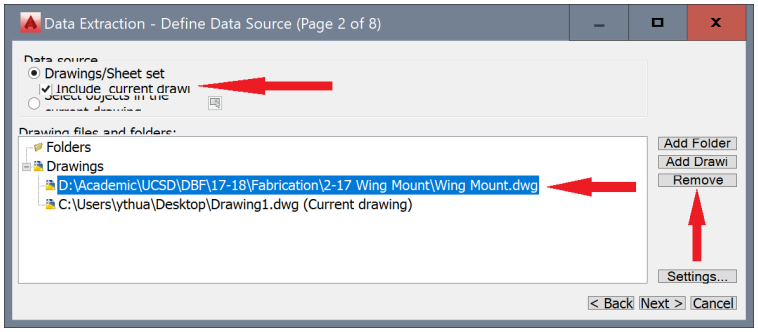


Figure 1.30: Eattext Page 2 Step 1

12. Then select the current drawing and click "Next".

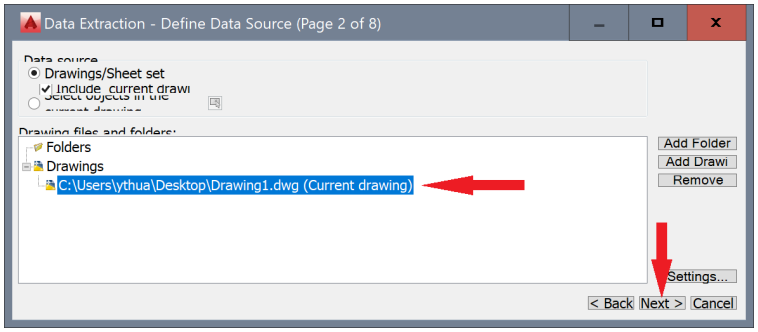


Figure 1.31: Eattext Page 2 Step 2

13. On page 3, make sure the drawing only has lines, or only has lines and arcs, depending on the template used. If not, cancel the process, fix the drawing, and redo the "eattext" command.

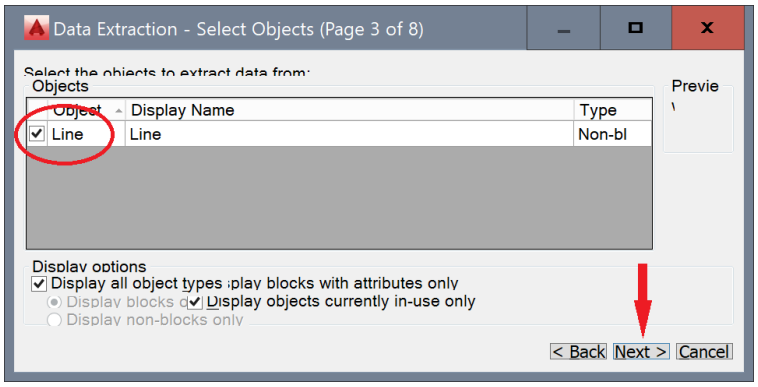


Figure 1.32: Eattext Page 3

14. Continue clicking "Next" until reaching page 6, name the data point file and save to the desired location. Click "Next" then "Finish".

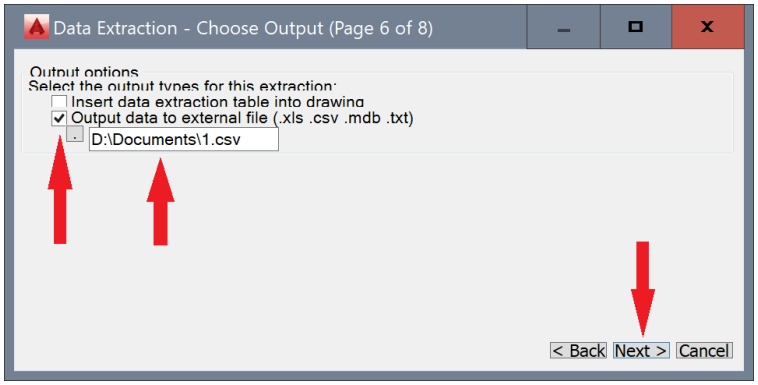


Figure 1.33: Eattext Page 6

15. Put the .csv file from AutoCAD into the same folder as the MatLAB script **"DBF_foamcutter_general_shape.m"** (provided in the **"User Package.zip"** from https://github.com/ythuang96/FoamCutter) and run the MatLAB code. Follow the prompts, and the G-code should be ready.

Chapter 2

Operate the FoamCutter

1 Switches

The foamcutter has 3 switches installed as shown in fig 2.1 below.

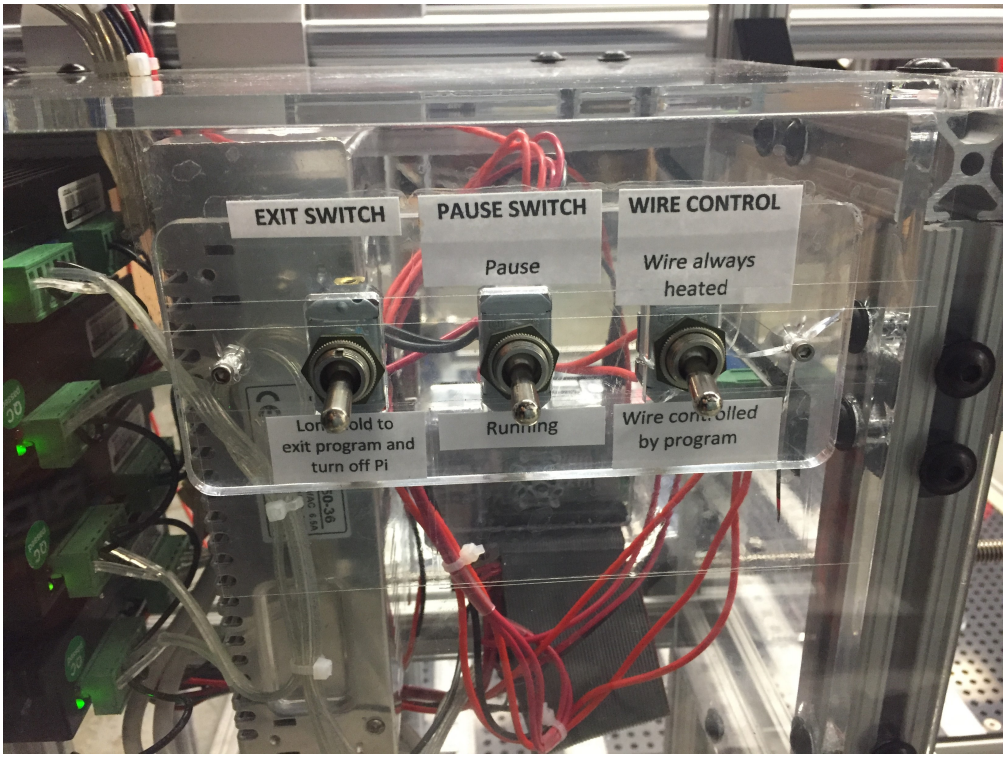


Figure 2.1: Three Switches of the Foamcutter

- Left most switch: **Exit switch**
  Only operational when the foamcutter software is running.
  Hold the switch for 4 seconds will stop all motor movement, exit the foamcutter software and shutdown the Raspberry Pi.

- Middle switch: **Pause switch**
  Only operational when the foamcutter software is running. Toggle the switch up to pause the motor movements.
  Toggle the switch down to resume the motor movements.
- Right most switch: **Wire Control switch**
  Operational regardless of foamcutter software.
  Toggle the switch up will enable the wire heating when the wire power supply is turned on and connected.
  Toggle the switch down will enable the foamcutter software to control the wire heating. The software automatically turns on the wire heating when starting the cut and turns off the wire heating when cut ends. **This switch should generally be left at the down position for safety.**

2 Start a Cut

Please follow the steps below to operate the foamcutter:

1. Prepare the G-code following guide from Chapter 1 Section 4.
2. Assemble the foamcutter following guide from Chapter 1 Section 2.
3. Check that the Wire Control switch (right most switch) is toggled to the down position.
4. Plug in the power cord.
5. Plug in the power supply for the hot wire but do not turn it on.
6. On your laptop, connect to the WiFi network named **"foamcutter"**, with password **"ucsdaiaadbf** or connect to the ethernet port to the ethernet port of the laptop with a ethernet cable.
7. Open PuTTY and connect to the Pi with user name **"pi"** and password **"ucsdaiaadbf"**.
8. Open WinSCP and connect to the Pi. Upload the G-code file to the Pi folder **"/home/pi"**.
9. In PuTTY, run: **sudo foamcutter**
10. The software will first check the Pause switch status and will prompt to toggle the switch down if needed.
11. The software will then ask to perform the homing process. Double check that the Molex connectors are connected, then press enter to perform homing.
12. The software will then enter the main menu, follow the prompts to select desired options to perform cuts or move the wire.

3 Secure the Foam

The foam can be secured with acrylic plates with thumb screws on the three sides. Follow the steps below to secure the foam on the work panel:

1. Measure the width of the foam that need to be cut.

2. Set one acrylic plate half of the width of the foam so that the foam will be centered on the work panel. Make sure both ruler reads the same so that the acrylic plate is in the vertical direction as shown in fig 2.2 below.

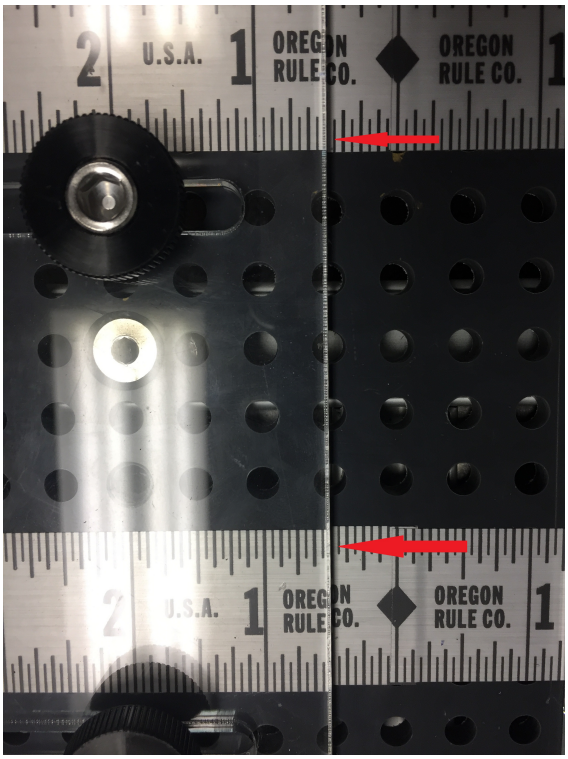


Figure 2.2: Secure the Foam Step 2

3. Secure the first plate with 3 thumb screws. **Do not over tighten the thumbscrews, or the threads on the work panel might be destroyed.**
4. Place the foam on the work panel with one side firmly against the guiding block and another side firmly against the first acrylic plate.
5. Place a second acrylic plate firmly against the foam and secure with 3 thumb screws as shown in fig 2.3 below.

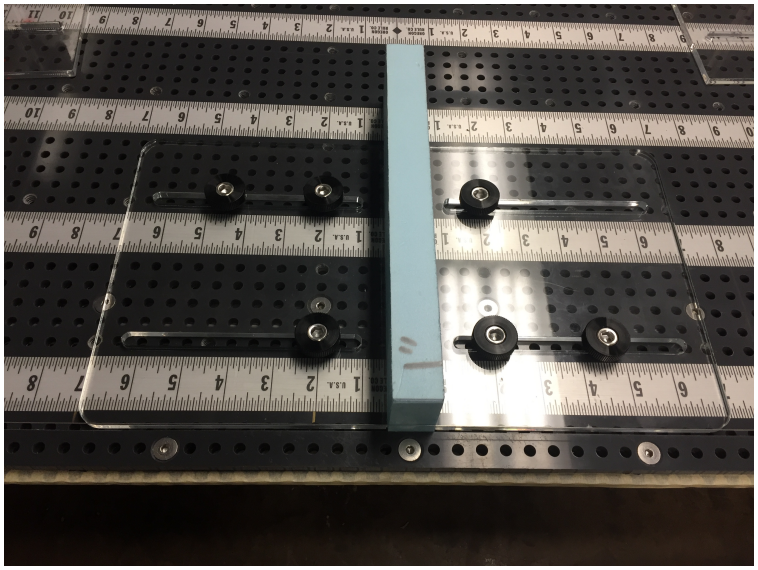


Figure 2.3: Secure the Foam Step 5

6. Secure the back of the foam with another acrylic plate and secure with 3 thumb screws as shown in fig 2.4 below.

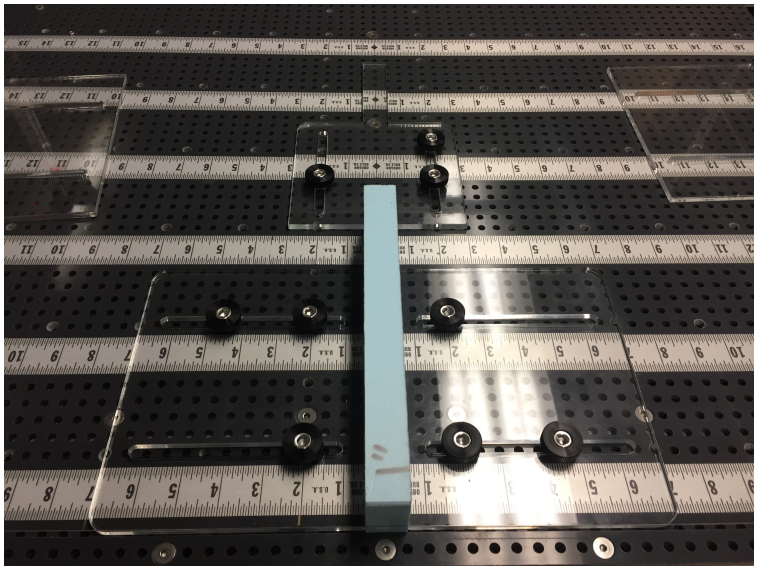


Figure 2.4: Secure the Foam Step 6

7. Place a small piece of aluminum block on top of the foam to prevent vertical movement as shown in fig 2.5 below.

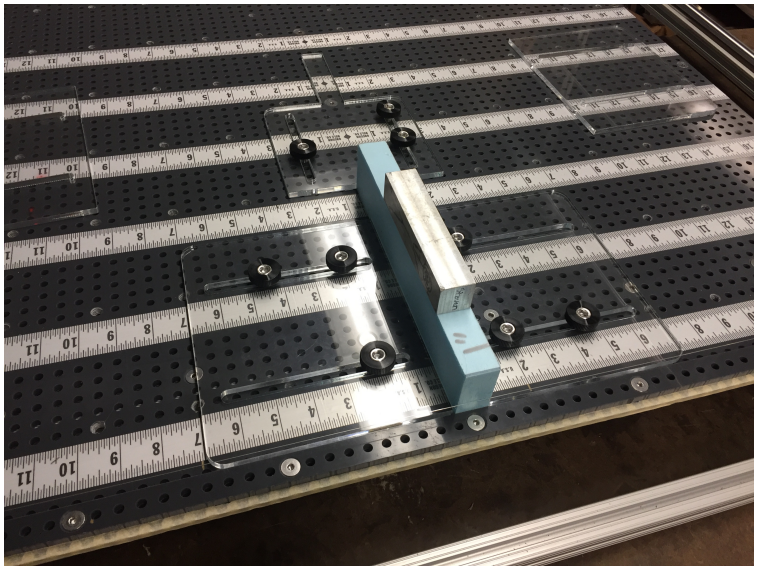


Figure 2.5: Secure the Foam Step 7

8. To secure very small pieces of foam, one of the acrylic plates can be flipped as shown in fig 2.6 below.

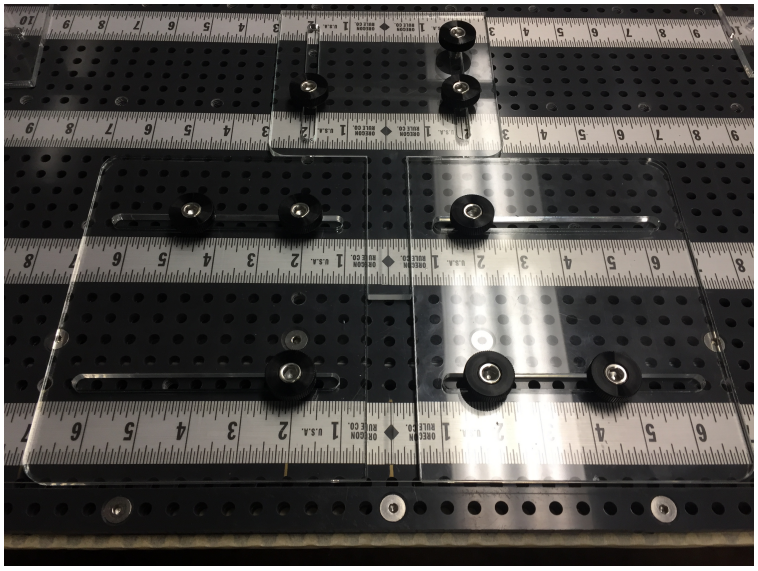


Figure 2.6: Secure the Foam Step 6

4 End a Cut

To the cutting operation half way through a G-code:

- In the PuTTY session, press **"Control + C"**. This will stop the cut and exit the program. It will also ask if desired to shutdown the Pi immediately.

- Or hold the Exit switch for more than 3 seconds. This will stop the cut and exit the program, but will also immediately shutdown the Pi.

To exit the program after the cut is finished, apart form the two options listed above, the user can also select the menu options to return to the main menu, and select the third option to exit the program. It will also ask if desired to shutdown the Pi immediately.

5 Cleaning Up

After the Pi is properly turned off by:

- Press Enter when seeing **"If you would like to shutdown the Pi now. Please press ENTER. Otherwise, press 'n' then press ENTER:"** when exiting the foamcutter software.
- Or run **"sudo shutdown now"** in PuTTY.

the green LED should flash a couple of times indicating the operating system is reading/writing to the SD card. After the green LED turns off, unplug the power cord and wrap it around the acrylic electronics box. Also remember to turn off the power supply for the hot wire and wrap the power cables for the hot wire around the two screws.

Chapter 3

Raspberry Pi Setup

Note:
All steps in this chapter are already performed on the currently used Raspberry Pi, and do not need to be performed again unless the Raspberry Pi or the SD card is damaged and needs replacement.

1 Install Raspbian on the Raspberry Pi

Raspbian is one of the recommended operating system for the Raspberry Pi. The following steps are used to install Raspbian and setup the Pi.

1. Perform the Laptop setup guide in Chapter 1 Section 1 if not already completed.
2. Download the Raspbian Sketch Lite version at: https://www.raspberrypi.org/downloads/raspbian/ The Lite version does not have desktop support, but should be sufficient for the foamcutter.
3. Download and install Etcher at https://etcher.io/.
4. Plug in the SD card into a laptop and run Etcher.
5. In Etcher, select the raspbian image just downloaded, then select the SD card. Lastly, click flash.
6. Wait for flash to complete, the SD card should now have two partition with one named "boot".
7. Enable SSH on the Pi by creating a file name "ssh" **without any extension** in the partition named "boot" on the SD card.
8. Unplug the SD card from the laptop and plug into the Pi. Connect a Ethernet cable from the Pi to same router your laptop is currently connected to.
9. Connect power to the Pi.
10. SSH into the Pi by opening up PuTTY and create a new session with host name: **"raspberrypi.local"**. Note on using PuTTY: to copy text from PuTTY, simply select the text, do not press "Control + C". To paste text to PuTTY, simply right click mouse,

do not press "Control + V".

11. Enter the user name **"pi"** then password **"raspberry"**.
12. Change the log in password: enter **"passwd"** on the command line and press Enter. Enter the current password **"raspberry"**, then enter the new password **"ucsdaiaadbf"** twice.
13. Change the name of the Pi by entering **"sudo raspi-config"**, select **"Network Options → Hostname"**, then enter the new hostname: **"foamcutter"**, then press Enter.
14. Select "Finish", then press Enter. When asked to reboot, select "Yes".

2 Install WiringPi on the Raspberry Pi

WiringPi is a GPIO Interface library written for the Raspberry Pi. A complete guide can be found at: http://wiringpi.com/. To install on the pi, follow the complete guide at: http://wiringpi.com/download-and-install/ or follow the steps below:

1. Power on the Pi and open PuTTY on your laptop.
2. Double click on the saved PuTTY session named "foamcutter" as shown in fig 3.1 below.

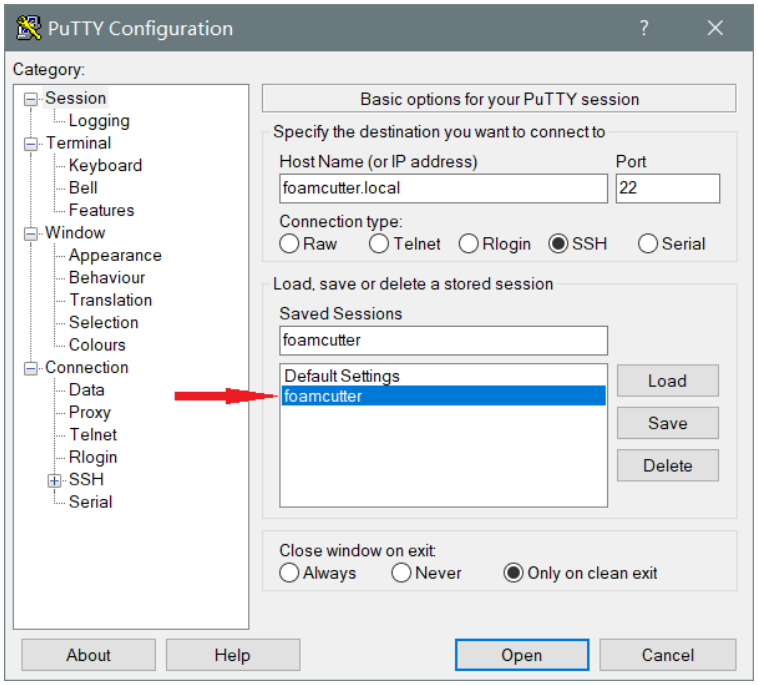


Figure 3.1: SSH to the Raspberry Pi

3. Enter user name **"pi"** and password **"ucsdaiaadbf"**.
4. Update the operating system using the following two command:
   - **sudo apt-get update**
   - **sudo apt-get upgrade**
5. Install GIT using:

- **sudo apt-get install git-core**

6. Install WiringPi using:
   - **cd**
   - **git clone git://git.drogon.net/wiringPi**
   - **cd /wiringPi**
   - **./build**
7. Completed when seeing:

   All Done.

   NOTE: To compile programs with wiringPi, you need to add:
   -lwiringPi
   to your compile line(s) To use the Gertboard, MaxDetect, etc.
   code (the devLib), you need to also add:
   -lwiringPiDev
   to your compile line(s).

3 Making the Pi into a Wireless Hotspot

Do not perform the following procedure. This method seems to have unsolved connection problems and is not recommended currently. Connection through an Ethernet cable is recommended.

Making the Pi into a Wireless Hotspot will enable operation of the foamcutter even in areas without a wireless network. The following steps are adapted from: https://elinux.org/RPI-Wireless-Hotspot

1. SSH into the Pi via PuTTY.
2. Run: **sudo apt-get install hostapd udhcpd**
3. Run: **sudo nano /etc/udhcpd.conf** This will open a text editor, use keyboard instead of mouse to navigate.
   - Find the line starting with **interface** and change to **interface wlan0**
   - Find the line starting with **opt router** and change to **opt router 192.168.42.1**
   - Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.
4. Run: **sudo nano /etc/default/udhcpd**
   - Add # to the front of the line **DHCPD_ENABLED="no"**
   - Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.
5. Run: **sudo ifconfig wlan0 192.168.42.1**
6. Run: **sudo nano /etc/network/interfaces**
   - Add the following lines to the bottom of the file if the line **iface wlan0 inet dhcp** is not present, otherwise replace it:
     iface wlan0 inet static
     address 192.168.42.1

netmask 255.255.255.0
- Add # to the beginning of each of the following lines if present:
  allow-hotplug wlan0
  wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
  iface default inet manual
- Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.

7. Run: **sudo nano /etc/hostapd/hostapd.conf**
   - Add the following lines to the bottom of the file:
     interface=wlan0
     driver=nl80211
     ssid=foamcutter
     hw_mode=g
     channel=6
     macaddr_acl=0
     auth_algs=1
     ignore_broadcast_ssid=0
     wpa=2
     wpa_passphrase=ucsdaiaadbf
     wpa_key_mgmt=WPA-PSK
     rsn_pairwise=CCMP

     channel=1
     ieee80211n=1
     wmm_enabled=1
     ht_capab=[HT40+][SHORT-GI-20][DSSS_CCK-40]
   - Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.
8. Run: **sudo nano /etc/default/hostapd**
   - Change the line from **#DAEMON_CONF=""** to **DAEMON_CONF="/etc/hostapd/hostapd.conf"**
   - Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.
9. Run: **sudo nano /etc/default/hostapd**
   - Add **net.ipv4.ip_forward=1** to the bottom of the file.
   - Press **Control** + **O** then **Enter** then **Control** + **X** to save and close file.
10. Run: **sudo update-rc.d hostapd enable**
11. Run: **sudo update-rc.d udhcpd enable**
12. Run: **sudo reboot**

Appendices

Appendix A

C Code for the Foamcutter Program

1 foamcutter.c

The C code for the foamcutter program, **foamcutter.c** is attached below. Verison 1.0.0, lasted updated: 6/30/2018.

```
/*********************************************************************
* FoamCutter
* This code is written for Design/Build/Fly CNC foamcutter.
* Written by Yuting Huang (ythuang96@gmail.com).
* Please report any bug to my email address.
*
* Last update: 6/30/2018
*
* Current Version: V 1.0.0
*********************************************************************/
#define VERSION_A 1
#define VERSION_B 0
#define VERSION_C 0
#include "foamcutter_setup.h"

typedef enum state_t{
    HOMED, GCODE, EXITING
} state_t;
typedef struct position_t{
    int32_t LX, LY, RX, RY;
} position_t;
typedef struct speed_t{
    float LX, LY, RX, RY;
} speed_t;
typedef struct coord_t{
    float LX_old, LY_old, RX_old, RY_old;
    float LX, LY, RX, RY;
} coord_t;
typedef struct coord_lim_t{
    float LX_max, LY_max, RX_max, RY_max;
    float LX_min, LY_min, RX_min, RY_min;
} coord_lim_t;

/*********************************************************************
************************ GLOBAL VARIABLES ***************************
*********************************************************************/
state_t state_;
position_t target_position_;
position_t current_position_;
position_t reached_position_;
position_t stop_;
speed_t set_speed_;
coord_t coord_;
coord_lim_t coord_lim_;

float coord_offset_x_;
float coord_offset_y_;
int gcode_menu_option_;
int ETA_;
struct timespec start_time_;
FILE *ptr_file_;

int state_STOP_ = 0;

// Threads
pthread_t LX_thread;
pthread_t LY_thread;
pthread_t RX_thread;
pthread_t RY_thread;
pthread_t printing_thread;
pthread_t cut_manager;
pthread_t switch_thread;
struct sched_param params_motor_thread;
struct sched_param params_print_thread;
struct sched_param params_cut_manager;
struct sched_param params_switch_thread;

/*********************************************************************
********************** FUNCTION DECLARATIONS ************************
*********************************************************************/
// THREADS
void* LX_thread_func(void* ptr);
void* LY_thread_func(void* ptr);
void* RX_thread_func(void* ptr);
void* RY_thread_func(void* ptr);
void* cut_manager_func(void* ptr);
void* print_func(void* ptr);
void* switch_thread_func(void* ptr);

// SYSTEM FUNCTIONS
void initialize_pin();
void home();
int loadtext(char* filename);
int check_cord(char* str);
int allreached();
void stop_all();
float cut_length_func();
```

```
void drive(int pin_pul, int pin_dir, float speed, int32_t delta_pulse, int* ptr_current, int* ptr_stop, int polarity );
void cut_gcode(char* filename);
void moveto(float x, float y);

// MENU FUNCTIONS
void main_menu();
void gcode_menu();
void move_menu();
int menu(int numb_of_options);
int menu_enter();
int menu_yes();
int menu_enter_one(float* output, char* string);
int menu_enter_two(float* output1, float* output2, char* string);

// OTHER FUNCTIONS
void nsleep(uint64_t ns);
int file_filter(const struct dirent *entry);
void removespace(char* str);
void SigHandler(int dummy);
float max(float a, float b);
float min(float a, float b);
void print_time(int sec);
int str2f(char* str, float* output);


/*****************************************************************
***************************** MAIN *******************************
*****************************************************************/

int main(){
    // Setup GPIO pins
    if (wiringPiSetupGpio () == -1) {
        printf("Initialization failed. Most likely you are not root\n");
        printf("Please remember to use 'sudo foamcutter'.\n");
        return 1 ;
    }
    initialize_pin(); digitalWrite(PIN_RELAY, LOW);
    // Setup signal handler for CTRL+C
    signal(SIGINT, SigHandler);

    // Start motor threads
    params_motor_thread.sched_priority = 90;
    pthread_setschedparam(LX_thread, SCHED_FIFO, &params_motor_thread);
    pthread_create(&LX_thread, NULL, LX_thread_func, (void*) NULL);
    pthread_setschedparam(LY_thread, SCHED_FIFO, &params_motor_thread);
    pthread_create(&LY_thread, NULL, LY_thread_func, (void*) NULL);
    pthread_setschedparam(RX_thread, SCHED_FIFO, &params_motor_thread);
    pthread_create(&RX_thread, NULL, RX_thread_func, (void*) NULL);
    pthread_setschedparam(RY_thread, SCHED_FIFO, &params_motor_thread);
    pthread_create(&RY_thread, NULL, RY_thread_func, (void*) NULL);

    stop_.LX = stop_.LY = stop_.RX = stop_.RY = 0;

    // Print Header
    printf("\n");
    printf("+-----------------------------------------------------------------+\n");
    printf("| DBF Foamcutter Program by Yuting Huang                          |\n");
    printf("| Current Version is V%d.%d.%d                                      |\n" \
        , VERSION_A, VERSION_B, VERSION_C);
    printf("| Contact me at ythuang96@gmail.com to report bugs                |\n");
    printf("|                                                                 |\n");
    printf("|-----------------------------------------------------------------|\n");
    printf("|                    Brief User Instructions                      |\n");
    printf("| At any time in the program:                                     |\n");
    printf("| 1. Press CTRL+C to exit the prgram                              |\n");
    printf("| 2. Long press EXIT button exit the prgram and shutdown the Pi   |\n");
    printf("| 3. Toggle PAUSE button to pause/resume all motor momevments     |\n");
    printf("+-----------------------------------------------------------------+\n\n");

    // Check Pause Switch
    int counter1 = 0;
    if (state_ != EXITING) {
        for (int i = 1; i<= 21; i++) {
            if (digitalRead(PIN_PAUSE)) counter1 ++;
            nsleep(500000);
        }
        if (counter1 > 10) printf("Please toggle the PAUSE switch to resume\n\n");
    }
    while (state_ != EXITING && counter1 > 10) {
        counter1 = 0;
        for (int i = 1; i<= 21; i++) {
            if (digitalRead(PIN_PAUSE)) counter1 ++;
            nsleep(500000);
        }
    }

    // Start Switch Thread
```

```
    params_switch_thread.sched_priority = 50;
    pthread_setschedparam(switch_thread, SCHED_FIFO, &params_switch_thread);
    pthread_create(&switch_thread, NULL, switch_thread_func, (void*) NULL);

    // Home the system
    if (state_ != EXITING) home();

    while (state_ != EXITING) main_menu();

    // stop all motors
    stop_all();
    // end swtich thread
    pthread_join(switch_thread, NULL);
    // end all motor threads
    pthread_join(LX_thread, NULL); pthread_join(LY_thread, NULL);
    pthread_join(RX_thread, NULL); pthread_join(RY_thread, NULL);
    // print exit messages
    printf("EXIT successful, Thank you for using the FoamCutter program.\n");
    if (state_STOP_) {
        printf("\nShuting down ... ...\n");
        system("shutdown -P now");
        return 0;
    }
    printf("\nIf you would like to shutdown the Pi now. Please press ENTER.\n");
    printf("Otherwise, press 'n' then press ENTER:    "); fflush(stdout);

/***************** SHUTDOWN MENU OPTIONS *************************/
    fd_set input_set; struct timeval timeout;
    timeout.tv_sec = 10; timeout.tv_usec = 0;

    // Listening for input stream for any activity
    FD_ZERO(&input_set); FD_SET(0, &input_set);
    while (!select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 10;
        FD_ZERO(&input_set ); FD_SET(0, &input_set);
    }

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // determine length of input
    int i; for(i=0; input_option[i]!='\0'; i++); i --;

    if (i == 1 && (input_option[0] == 'n' || input_option[0] == 'N')) {
        // if chose not to shutdown
        printf("\nOk, Please remember to use 'sudo shutdown now'\n");
        printf("to shutdown the Pi before unpluging the power.\n\n");
    }
    else { // chose to shutdown
        printf("\nShuting down ... ...\n");
        system("shutdown -P now");
    }

    return 0;
}

/*********************************************************************
***************************** THREADS ********************************
*********************************************************************/
void* LX_thread_func(void* ptr){
    while (state_ != EXITING) {
        if (set_speed_.LX == 0 || reached_position_.LX == 1 || stop_.LX == 1) {
            if (current_position_.LX == target_position_.LX){
                reached_position_.LX = 1;
            }
            digitalWrite(PIN_LX_PUL, LOW);
            nsleep(1000000);
        }
        else {
            drive(PIN_LX_PUL, PIN_LX_DIR, set_speed_.LX, \
                target_position_.LX - current_position_.LX, \
                &(current_position_.LX), &(stop_.LX), POLARITY_LX);
            if (!stop_.LX) reached_position_.LX = 1;
        }
    }
    return NULL;
}

void* LY_thread_func(void* ptr){
    while (state_ != EXITING) {
        if (set_speed_.LY == 0 || reached_position_.LY == 1 || stop_.LY == 1) {
            if (current_position_.LY == target_position_.LY){
                reached_position_.LY = 1;
            }
            digitalWrite(PIN_LY_PUL, LOW);
            nsleep(1000000);
        }
        else {
```

```
            drive(PIN_LY_PUL, PIN_LY_DIR, set_speed_.LY, \
                target_position_.LY - current_position_.LY, \
                &(current_position_.LY), &(stop_.LY), POLARITY_LY);
            if (!stop_.LY) reached_position_.LY = 1;
        }
    }
    return NULL;
}

void* RX_thread_func(void* ptr){
    while (state_ != EXITING) {
        if (set_speed_.RX == 0 || reached_position_.RX == 1 || stop_.RX == 1) {
            if (current_position_.RX == target_position_.RX){
                reached_position_.RX = 1;
            }
            digitalWrite(PIN_RX_PUL, LOW);
            nsleep(1000000);
        }
        else {
            drive(PIN_RX_PUL, PIN_RX_DIR, set_speed_.RX, \
                target_position_.RX - current_position_.RX, \
                &(current_position_.RX), &(stop_.RX), POLARITY_RX);
            if (!stop_.RX) reached_position_.RX = 1;
        }
    }
    return NULL;
}

void* RY_thread_func(void* ptr){
    while (state_ != EXITING) {
        if (set_speed_.RY == 0 || reached_position_.RY == 1 || stop_.RY == 1) {
            if (current_position_.RY == target_position_.RY){
                reached_position_.RY = 1;
            }
            digitalWrite(PIN_RY_PUL, LOW);
            nsleep(1000000);
        }
        else {
            drive(PIN_RY_PUL, PIN_RY_DIR, set_speed_.RY, \
                target_position_.RY - current_position_.RY, \
                &(current_position_.RY), &(stop_.RY), POLARITY_RY);
            if (!stop_.RY) reached_position_.RY = 1;
        }
    }
    return NULL;
}

void* cut_manager_func(void* ptr){
    char buf[500];
    while(state_ == GCODE && fgets(buf,500, ptr_file_)!=NULL){
        while (state_ == GCODE && !allreached()) {
            nsleep(10000000);
        } // wait till last coord is reached
        removespace(buf); // remove spaces
        if (!strncmp(buf,"G4P",3)) { // check if is a pause statement
            float temp; str2f(buf+3, &temp);
            nsleep((uint64_t)(floor(temp*1.0E9)));
        }
        else if (!strncmp(buf,"G1",2)) { // if a cut statement
            check_cord(buf); // read coordinates and update global coord_
            // Update new target position
            target_position_.LX = (int32_t)floor((coord_.LX + coord_offset_x_) * MM2PULSE);
            target_position_.LY = (int32_t)floor((coord_.LY + coord_offset_y_) * MM2PULSE);
            target_position_.RX = (int32_t)floor((coord_.RX + coord_offset_x_) * MM2PULSE);
            target_position_.RY = (int32_t)floor((coord_.RY + coord_offset_y_) * MM2PULSE);
            // Calculate time to move to the next coord
            float dL = sqrt( pow(target_position_.LX - current_position_.LX,2.0) \
                           + pow(target_position_.LY - current_position_.LY,2.0));
            float dR = sqrt( pow(target_position_.RX - current_position_.RX,2.0) \
                           + pow(target_position_.RY - current_position_.RY,2.0));
            float time = (dL + dR)/2.0/FEEDRATE/MM2PULSE;
            // Set speed for all 4 axis
            set_speed_.LX = (target_position_.LX - current_position_.LX)/time/MM2PULSE;
            set_speed_.LY = (target_position_.LY - current_position_.LY)/time/MM2PULSE;
            set_speed_.RX = (target_position_.RX - current_position_.RX)/time/MM2PULSE;
            set_speed_.RY = (target_position_.RY - current_position_.RY)/time/MM2PULSE;
            if (state_ == GCODE) {
                // Start the cut by setting reached_position_ to 0
                reached_position_.LX = reached_position_.LY = 0;
                reached_position_.RX = reached_position_.RY = 0;
                // wait till new coord is reached
                nsleep((uint64_t)(floor(time*1.0E9)));
            }
        } // end while --- read line by line
    }
    // if the state_ is still GCODE, but the while loop ended;
    // means the end of file is reached, and therefore cut is complete
```

```
    if (state_ == GCODE) { state_ = HOMED;}
    return NULL;
}

void* print_func(void* ptr){
    struct timespec current_time;
    int elapsed_time = 0;
    int remain_time;
    while(state_ == GCODE){
        clock_gettime( CLOCK_REALTIME, &current_time);
        elapsed_time = current_time.tv_sec - start_time_.tv_sec;
        remain_time = ETA_ - elapsed_time;
        printf("\r");
        printf("%7.3f/%8.3f|%7.3f/%8.3f|%7.3f/%8.3f|%7.3f/%8.3f|    ", \
            current_position_.LX/MM2PULSE*MM2IN , current_position_.LX/MM2PULSE, \
            current_position_.LY/MM2PULSE*MM2IN , current_position_.LY/MM2PULSE, \
            current_position_.RX/MM2PULSE*MM2IN , current_position_.RX/MM2PULSE, \
            current_position_.RY/MM2PULSE*MM2IN , current_position_.RY/MM2PULSE);
        print_time(elapsed_time); printf("    |   ");
        print_time(remain_time);  printf("   |");
        fflush(stdout);
        nsleep(1000000000); // run at 1 Hz
    }
    return NULL;
}

void* switch_thread_func(void* ptr){
    int state_P = 0;
    int counter_P, counter_S;
    int counter_S2 = 0;
    while (state_ != EXITING) {
        counter_P = counter_S = 0;
        for (int i = 1; i<= 41; i++) {
            if (digitalRead(PIN_PAUSE)) counter_P ++;
            if (digitalRead(PIN_STOP )) counter_S ++;
            nsleep(500000);
        }

        if (counter_P > 25 && !state_P) {
            stop_all();
            state_P = 1;
        }
        else if (counter_P <= 15 && state_P && state_ == GCODE) {
            state_P = 0;
            stop_.LX = stop_.LY = stop_.RX = stop_.RY = 0;
            digitalWrite(PIN_RELAY, HIGH);
        }
        else if (counter_P <= 15 && state_P && state_ == HOMED) {
            state_P = 0;
            stop_.LX = stop_.LY = stop_.RX = stop_.RY = 0;
        }

        if (counter_S > 25) counter_S2 ++;
        else counter_S2 = 0;

        if (counter_S2 == 100){
            state_ = EXITING; stop_all(); state_STOP_ = 1;
            printf("\n\n************************** EXITING PROGRAM **************************\n");
        }
    }
    return NULL;
}

/*********************************************************************
********************** SYSTEM FUNCTIONS **********************
*********************************************************************/

void initialize_pin(){
    // Setup limit switch pins
    pinMode(PIN_LX_LIM,INPUT); pullUpDnControl(PIN_LX_LIM,PUD_DOWN);
    pinMode(PIN_LY_LIM,INPUT); pullUpDnControl(PIN_LY_LIM,PUD_DOWN);
    pinMode(PIN_RX_LIM,INPUT); pullUpDnControl(PIN_RX_LIM,PUD_DOWN);
    pinMode(PIN_RY_LIM,INPUT); pullUpDnControl(PIN_RY_LIM,PUD_DOWN);
    // Setup motor dive pins
    pinMode(PIN_LX_DIR,OUTPUT); pinMode(PIN_LX_PUL,OUTPUT);
    pinMode(PIN_LY_DIR,OUTPUT); pinMode(PIN_LY_PUL,OUTPUT);
    pinMode(PIN_RX_DIR,OUTPUT); pinMode(PIN_RX_PUL,OUTPUT);
    pinMode(PIN_RY_DIR,OUTPUT); pinMode(PIN_RY_PUL,OUTPUT);
    // Setup switches pins
    pinMode(PIN_PAUSE,INPUT); pullUpDnControl(PIN_PAUSE,PUD_DOWN);
    pinMode(PIN_STOP ,INPUT); pullUpDnControl(PIN_STOP ,PUD_DOWN);
    // Setup relay control pin
    pinMode(PIN_RELAY,OUTPUT);
    return;
}

void home(){
```

```
    // Print Header
    printf("I see the system is not homed yet, please press ENTER to home the system:    ");
    fflush(stdout);
    if (menu_enter() == -2) return; // if EXITING state, end function
    printf("Homing ...   "); fflush(stdout);

    // Home the system
    current_position_.LX = current_position_.LY = current_position_.RX = current_position_.RY = 0;
    reached_position_.LX = reached_position_.LY = reached_position_.RX = reached_position_.RY = 0;
    target_position_.LX  = target_position_.LY  = target_position_.RX  = target_position_.RY  = -640000;

    // Home X axis
    set_speed_.LY = set_speed_.RY = 0.0;
    set_speed_.LX = set_speed_.RX = -1.0;

    int counter1, counter2;
    int state1 = 0; int state2 = 0;

    while (state_ != EXITING && (!state1 || !state2)) {
        counter1 = counter2 = 0;
        for (int i = 1; i<= 41; i++) {
            if (digitalRead(PIN_LX_LIM)) counter1 ++;
            if (digitalRead(PIN_RX_LIM)) counter2 ++;
            nsleep(20000);
        }
        if (counter1 > 30 && !state1) {
            state1 = 1;
            stop_.LX = 1;
            set_speed_.LX = 0.0;
            current_position_.LX = LIM2ORIGIN_LX;
            reached_position_.LX = 1;
        }
        if (counter2 > 30 && !state2) {
            state2 = 1;
            stop_.RX = 1;
            set_speed_.RX = 0.0;
            current_position_.RX = LIM2ORIGIN_RX;
            reached_position_.RX = 1;
        }
    }

    if (state_ == EXITING) return;
    nsleep(200000000);
    target_position_.LX = LIM2ORIGIN_LX + 1000;
    target_position_.RX = LIM2ORIGIN_RX + 1000;
    set_speed_.LX       = set_speed_.RX       = +FEEDRATE;
    stop_.LX = stop_.RX = 0;
    reached_position_.LX = reached_position_.RX = 0;
    while (state_ != EXITING && (!reached_position_.LX || !reached_position_.RX)) { nsleep(1000000);}

    // Home Y axis
    if (state_ == EXITING) return;
    nsleep(500000000);
    set_speed_.LX = set_speed_.RX = 0.0;
    set_speed_.LY = set_speed_.RY = -1.0;

    counter1 = counter2 = state1 = state2 = 0;
    while (state_ != EXITING && (!state1 || !state2)) {
        counter1 = counter2 = 0;
        for (int i = 1; i<= 41; i++) {
            if (digitalRead(PIN_LY_LIM)) counter1 ++;
            if (digitalRead(PIN_RY_LIM)) counter2 ++;
            nsleep(20000);
        }
        if (counter1 > 30 && !state1) {
            state1 = 1;
            stop_.LY = 1;
            set_speed_.LY = 0.0;
            current_position_.LY = LIM2ORIGIN_LY;
            reached_position_.LY = 1;
        }
        if (counter2 > 30 && !state2) {
            state2 = 1;
            stop_.RY = 1;
            set_speed_.RY = 0.0;
            current_position_.RY = LIM2ORIGIN_RY;
            reached_position_.RY = 1;
        }
    }
    if (state_ == EXITING) return;
    nsleep(500000000);

    // Move to orgin
    if (state_ == EXITING) return;
    printf("All limits reached, Moving to Origin ...   "); fflush(stdout);

    target_position_.LX = target_position_.LY = target_position_.RX = target_position_.RY = 0;
```

```
    set_speed_.LX       = set_speed_.LY       = set_speed_.RX       = set_speed_.RY       = +FEEDRATE;
    stop_.LX = stop_.LY = stop_.RX = stop_.RY = 0;

    reached_position_.LY = reached_position_.RY = reached_position_.LX = reached_position_.RX = 0;
    while (state_ != EXITING && !allreached()) { nsleep(1000000);}

    if (state_ == EXITING) return;
    state_ = HOMED;
    printf("Homing Complete! \n\n");

    return;
}

int loadtext(char* filename){
    float pause_time = 0.0;
    float cut_length = 0.0;
    int asym_cut = 0;
    int error = 0; // set error to 1 will return to main menu
    float span;
    float coord_x_min;
    float coord_y_min;
    float width;
    float height;

    coord_lim_.LX_max = coord_lim_.LX_min = 0.0;
    coord_lim_.LY_max = coord_lim_.LY_min = 0.0;
    coord_lim_.RX_max = coord_lim_.RX_min = 0.0;
    coord_lim_.RY_max = coord_lim_.RY_min = 0.0;

    coord_.LX_old = coord_.LX = 0.0;
    coord_.LY_old = coord_.LY = 0.0;
    coord_.RX_old = coord_.RX = 0.0;
    coord_.RY_old = coord_.RY = 0.0;

    coord_offset_x_ = coord_offset_y_ = 0.0;
/************************** READ FILE ******************************/
    FILE *ptr_file; char buf[500];
    ptr_file =fopen(filename, "r");

    int line_numb = 1;
    while (fgets(buf,500, ptr_file)!=NULL){ // get line by line
        removespace(buf); // remove spaces
        if (!strncmp(buf,"G4P",3)) { // check if is a pause statement
            float temp;
            if (str2f(buf+3, &temp)) {pause_time += temp;}
            else {
                printf("G-code error at line %d. Returning to Main Menu.\n\n",line_numb);
                error = 1;
                return 1;
            }
        }
        else if (!strncmp(buf,"G1",2)) { // if a cut statement
            if (check_cord(buf)) { // check if statement is valid
                cut_length += cut_length_func();
                if (fabs(coord_.LX - coord_.RX) + fabs(coord_.LY - coord_.RY) > 0.0001 ) {
                    asym_cut ++;
                }
                // update the max/min coordinates
                coord_lim_.LX_max = max(coord_lim_.LX_max, coord_.LX);
                coord_lim_.LX_min = min(coord_lim_.LX_min, coord_.LX);
                coord_lim_.LY_max = max(coord_lim_.LY_max, coord_.LY);
                coord_lim_.LY_min = min(coord_lim_.LY_min, coord_.LY);
                coord_lim_.RX_max = max(coord_lim_.RX_max, coord_.RX);
                coord_lim_.RX_min = min(coord_lim_.RX_min, coord_.RX);
                coord_lim_.RY_max = max(coord_lim_.RY_max, coord_.RY);
                coord_lim_.RY_min = min(coord_lim_.RY_min, coord_.RY);
            }
            // if not a valid line, print error message and stop reading
            else {
                printf("G-code error at line %d. Returning to Main Menu.\n\n",line_numb);
                error = 1;
                return 1;
            }
        }
        line_numb ++;
    } // end while --- read line by line
    fclose(ptr_file); // read complete

    if (!error && state_ != EXITING) { // if no reading error occured
        coord_x_min = min(coord_lim_.LX_min,coord_lim_.RX_min);
        coord_y_min = min(coord_lim_.LY_min,coord_lim_.RY_min);
    } // end if --- check error
/*************** FIRST: check if offset is needed ******************/
    // check x
    if (coord_x_min < 0 && state_ != EXITING && !error){
        printf("I see you have a min x coordinate of %7.3f in (%8.3f mm)\n", \
            coord_x_min*MM2IN, coord_x_min);
```

```
        printf("A negative value is not allowed\n");
        printf("You can use the minimum offset, or enter one yourself\n");
        printf("Would you like to use the minimum offset for x?\n");
        if (!menu_yes()){
            while(state_ != EXITING) {
                int temp = menu_enter_one(&coord_offset_x_,"Please enter the x offset");
                if      (temp == -1) {printf("Invalid input, please enter again.\n\n");}
                else if (temp ==  1) {
                    if (coord_offset_x_ < - coord_x_min) {
                        printf("Insufficient x offset, please enter a bigger x offset\n\n");
                    }
                    else break;
                }
            }
        }
        else { coord_offset_x_ = - coord_x_min;}
    }
    // check y
    if (coord_y_min < 0 && state_ != EXITING && !error){
        printf("I see you have a min y coordinate of %7.3f in (%8.3f mm)\n", \
            coord_y_min*MM2IN, coord_y_min);
        printf("A negative value is not allowed\n");
        printf("You can use the minimum offset, or enter one yourself\n");
        printf("If you are cutting part of a 3-piece wing\n");
        printf("I would recommend enter the same offset for all 3 pieces.\n");
        printf("It will make the vaccum bagging easier\n\n");
        printf("Would you like to use the minimum offset for y?\n");
        if (!menu_yes()){
            while(state_ != EXITING) {
                int temp = menu_enter_one(&coord_offset_y_,"Please enter the y offset");
                if      (temp == -1) {printf("Invalid input, please enter again.\n\n");}
                else if (temp ==  1) {
                    if (coord_offset_y_ < - coord_y_min) {
                        printf("Insufficient y offset, please enter a bigger y offset\n\n");
                    }
                    else break;
                }
            }
        }
        else { coord_offset_y_ = - coord_y_min;}
    }
    coord_y_min += coord_offset_x_;
    coord_y_min += coord_offset_y_;
    coord_lim_.LX_max += coord_offset_x_; coord_lim_.LX_min += coord_offset_x_;
    coord_lim_.LY_max += coord_offset_y_; coord_lim_.LY_min += coord_offset_y_;
    coord_lim_.RX_max += coord_offset_x_; coord_lim_.RX_min += coord_offset_x_;
    coord_lim_.RY_max += coord_offset_y_; coord_lim_.RY_min += coord_offset_y_;

    if (coord_lim_.LX_max > X_MAX || coord_lim_.RX_max > X_MAX) {
        printf("X axis out of bound, maximum X distance is 29 inches\n");
        printf("Returning to G-code Menu\n\n");
        error = 1; gcode_menu_option_ = -1;
    }
    if (coord_lim_.LY_max > Y_MAX || coord_lim_.RY_max > Y_MAX) {
        printf("Y axis out of bound, maximum Y distance is 16 inches\n");
        printf("Returning to G-code Menu\n\n");
        error = 1; gcode_menu_option_ = -1;
    }
/************** SECOND: determine and check foamsize ****************/
    if(!asym_cut && state_ != EXITING && !error){ // if not an asymetric cut
        printf("I see this is a symmetric cut\n");
        printf("The minimum require foam size is:\n");
        width  = coord_lim_.LX_max;
        height = coord_lim_.LY_max;
        printf("Width     (x-direction): %7.3f in (%8.3f mm)\n", width*MM2IN, width);
        printf("Thickness (y-direction): %7.3f in (%8.3f mm)\n", height*MM2IN, height);
        printf("Please leave some extra space.\n");
    }
    else if (asym_cut && state_ != EXITING && !error){ // if an asymetric cut
        printf("I see this is an asymmetric cut\n");
        printf("The minimum require foam size depends on the span of the cut.\n");
        printf("Please enter the span size of the cut.\n");
        while(state_ != EXITING && menu_enter_one(&span,"Please enter the span size") == -1) {
                printf("Invalid input, please enter again.\n\n");
        }
        width  = min(coord_lim_.LX_max,coord_lim_.RX_max) + \
            fabs(coord_lim_.LX_max-coord_lim_.RX_max)*(CUTTERWIDTH + span)/2.0/CUTTERWIDTH;
        height = min(coord_lim_.LY_max,coord_lim_.RY_max) + \
            fabs(coord_lim_.LY_max-coord_lim_.RY_max)*(CUTTERWIDTH + span)/2.0/CUTTERWIDTH;
        if (state_ != EXITING){
            printf("Width     (x-direction): %7.3f in (%8.3f mm)\n", width*MM2IN, width);
            printf("Thickness (y-direction): %7.3f in (%8.3f mm)\n", height*MM2IN, height);
            printf("Please leave some extra space.\n");
        }
    }
    if(state_ != EXITING && !error){
        printf("Does this look correct and matches your foam size?\n"); fflush(stdout);
```

```
        if (!menu_yes()){
            printf("Looks like there's something wrong. Returning to G-code Menu.\n\n");
            error = 1; gcode_menu_option_ = -1;
        }
    }
/****************** THIRD: review cut settings ********************/
    if (state_ != EXITING && !error){
        printf("*************************** CUT SETTINGS ****************************\n");
        printf("G-code file: %s.\n", filename);
        if (asym_cut) {
            printf("Asymetric Cut of span %7.3f in (%8.3f mm)\n", span*MM2IN, span);
        }
        else {printf("Symmetric Cut\n");}
        printf("Minimum Foam Size:\n");
        printf("Width     (x-direction): %7.3f in (%8.3f mm)\n", width*MM2IN, width);
        printf("Thickness (y-direction): %7.3f in (%8.3f mm)\n", height*MM2IN, height);
        if (coord_offset_x_) {
            printf("x offset of %7.3f in (%8.3f mm)", coord_offset_x_*MM2IN, coord_offset_x_);
        }
        else {printf("No x offset");}
        printf(" and ");
        if (coord_offset_y_) {
            printf("y offset of %7.3f in (%8.3f mm)", coord_offset_y_*MM2IN, coord_offset_y_);
        }
        else {printf("No y offset");}
        printf("\n");
        printf("Estimate total time of cut: ");
        ETA_ = round(pause_time + cut_length/FEEDRATE );
        print_time(ETA_); printf("\n");

        if (asym_cut) {printf("Please make sure the foam is centered.\n");}
        printf("\nWould you like to start the cut?\n");

        if (!menu_yes()){
            printf("OK, setting incorrect. Returning to G-code Menu.\n\n");
            error = 1; gcode_menu_option_ = -1;
        }
    }
    return error;
}

int check_cord(char* str){
    coord_.LX_old = coord_.LX; coord_.LY_old = coord_.LY;
    coord_.RX_old = coord_.RX; coord_.RY_old = coord_.RY;
    char* ptr_X = strchr(str, 'X');
    char* ptr_Y = strchr(str, 'Y');
    char* ptr_Z = strchr(str, 'Z');
    char* ptr_A = strchr(str, 'A');
    if (ptr_X && ptr_Y && ptr_Z && ptr_A &&  \
        (ptr_X < ptr_Y) && (ptr_Y < ptr_Z) &&(ptr_Z < ptr_A) ){
        char temp1[20], temp2[20], temp3[20], temp4[20];
        float tempf1, tempf2, tempf3, tempf4;
        strncpy(temp1,ptr_X+1,ptr_Y-ptr_X-1); temp1[(ptr_Y-ptr_X-1)] = '\0';
        strncpy(temp2,ptr_Y+1,ptr_Z-ptr_Y-1); temp2[(ptr_Z-ptr_Y-1)] = '\0';
        strncpy(temp3,ptr_Z+1,ptr_A-ptr_Z-1); temp3[(ptr_A-ptr_Z-1)] = '\0';
        strncpy(temp4,ptr_A+1,10           );

        if(str2f(temp1, &tempf1) && str2f(temp2, &tempf2) && \
            str2f(temp3, &tempf3) && str2f(temp4, &tempf4)){
            coord_.LX = tempf1; coord_.LY = tempf2;
            coord_.RX = tempf3; coord_.RY = tempf4;
            return 1;
        }
        else return 0;
    }
    else return 0;
}

int allreached() {
    return reached_position_.LX * reached_position_.LY * reached_position_.RX * reached_position_.RY;
}

void stop_all() {
    stop_.LX = stop_.LY = stop_.RX = stop_.RY = 1;
    digitalWrite(PIN_RELAY,LOW);
    return;
}

float cut_length_func(){
    float L_length = sqrt(pow((coord_.LX - coord_.LX_old),2.0) + pow((coord_.LY - coord_.LY_old),2.0));
    float R_length = sqrt(pow((coord_.RX - coord_.RX_old),2.0) + pow((coord_.RY - coord_.RY_old),2.0));
    return (L_length + R_length)/2.0;
}

void drive(int pin_pul, int pin_dir, float speed, int32_t delta_pulse, int* ptr_current, int* ptr_stop, int polarity ){
    int inc;
    if      (speed*polarity > 0) { digitalWrite(pin_dir, HIGH); inc =  1*polarity;}
```

```
    else if (speed*polarity < 0) { digitalWrite(pin_dir, LOW ); inc = -1*polarity;}
    // the time between pulses, calculated from speed
    // 4000 is the sleep time in the loop;
    // 100000 is the approximate code execution time;
    uint64_t sleep_time = floor(1000000000.0/MM2PULSE/fabs(speed)) - 4000 - 100000;
    nsleep(5000); // ensure dir pin leads by at least 5 microsec
    for (int i = 0; i < abs(delta_pulse); i++) { // send out desired number of pulses
        digitalWrite(pin_pul, HIGH);
        nsleep(4000); // ensure pulse width of at least 2 microsec
        digitalWrite(pin_pul, LOW );
        *ptr_current += inc;
        if (*ptr_stop) break; // stop the motor if stop is a 1;
        nsleep(sleep_time);
    }
    return;
}

void cut_gcode(char* filename){
    state_ = GCODE;
    if (state_ == EXITING) return;
    moveto(-5.0,0.0);
    while (state_ != EXITING && !allreached()){ nsleep(1000000);}
    if (state_ == EXITING) return;
    printf("Please connect and turn on the power supply for the hot wire\n");
    printf("Please make sure the voltage is approximately 10V, and press ENTER to continue:    ");
    fflush(stdout);
    if (menu_enter() == -2) return;
    digitalWrite(PIN_RELAY,HIGH);
    printf("Please now adjust the power supply to the desired current.\n");
    printf("Recommend 2.1 to 2.3 Amps, depending on the cut span.\n");
    printf("Use higher current for wider cuts.\n");
    printf("Increase current if wire bows significantly.\n");
    printf("Press ENTER to start cutting:    ");
    fflush(stdout);
    if (menu_enter() == -2) return;
    printf("Heating wire ... ..."); fflush(stdout);
    if (state_ != EXITING) nsleep(5000000000);
    if (state_ == EXITING) return;
    printf(" Cut Starting\n");
    moveto(0.0,0.0);
    while (state_ != EXITING && !allreached()) nsleep(1000000);
    if (state_ == EXITING) return;

    coord_.LX_old = coord_.LX = 0.0;
    coord_.LY_old = coord_.LY = 0.0;
    coord_.RX_old = coord_.RX = 0.0;
    coord_.RY_old = coord_.RY = 0.0;
    clock_gettime( CLOCK_REALTIME, &start_time_);

    ptr_file_ =fopen(filename, "r");
    printf("       LX       |       LY       |       RX       |       RY       |          TIME          |\n");
    printf("   in  /   mm   |   in  /   mm   |   in  /   mm   |   in  /   mm   |  Elapsed  | Remaining |\n");

    params_print_thread.sched_priority = 40;
    params_cut_manager.sched_priority = 99;
    pthread_setschedparam(printing_thread, SCHED_FIFO, &params_print_thread);
    pthread_create(&printing_thread, NULL, print_func, (void*) NULL);
    pthread_setschedparam(cut_manager, SCHED_FIFO, &params_cut_manager);
    pthread_create(&cut_manager, NULL, cut_manager_func, (void*) NULL);

    while (state_ == GCODE) nsleep(100000);

    fclose(ptr_file_); // read complete

    if (state_ != EXITING) moveto(-5.0,0.0);
    while (state_ != EXITING && !allreached()) nsleep(1000000);
    digitalWrite(PIN_RELAY,LOW);
    if (state_ != EXITING) moveto( 0.0,0.0);
    while (state_ != EXITING && !allreached()) nsleep(1000000);
    if (state_ != EXITING) printf("\nCut Complete, Returning to Main Menu.\n\n");

    pthread_join(cut_manager, NULL);
    pthread_join(printing_thread, NULL);
    return;
}

void moveto(float x, float y){
    target_position_.LX = target_position_.RX = (int32_t)floor(x * MM2PULSE);
    target_position_.LY = target_position_.RY = (int32_t)floor(y * MM2PULSE);
    // Set speed for all 4 axis
    if      (target_position_.LX  > current_position_.LX) set_speed_.LX = +FEEDRATE;
    else if (target_position_.LX  < current_position_.LX) set_speed_.LX = -FEEDRATE;
    else if (target_position_.LX == current_position_.LX) set_speed_.LX = 0.0;

    if      (target_position_.LY  > current_position_.LY) set_speed_.LY = +FEEDRATE;
    else if (target_position_.LY  < current_position_.LY) set_speed_.LY = -FEEDRATE;
    else if (target_position_.LY == current_position_.LY) set_speed_.LY = 0.0;
```

```
871
872     if      (target_position_.RX  > current_position_.RX) set_speed_.RX = +FEEDRATE;
873     else if (target_position_.RX  < current_position_.RX) set_speed_.RX = -FEEDRATE;
874     else if (target_position_.RX == current_position_.RX) set_speed_.RX = 0.0;
875
876     if      (target_position_.RY  > current_position_.RY) set_speed_.RY = +FEEDRATE;
877     else if (target_position_.RY  < current_position_.RY) set_speed_.RY = -FEEDRATE;
878     else if (target_position_.RY == current_position_.RY) set_speed_.RY = 0.0;
879
880     if (state_ != EXITING) {
881         // Start the cut by setting reached_position_ to 0
882         reached_position_.LX = reached_position_.LY = 0;
883         reached_position_.RX = reached_position_.RY = 0;
884     }
885     return;
886 }
887
888 /*****************************************************************
889 ************************* MENU FUNCTIONS ************************
890 *****************************************************************/
891
892 void main_menu() {
893     printf("****************************** MAIN MENU ******************************\n");
894     printf("Please choose from the following options:\n");
895     printf("a. Load and cut from G-Code;\n");
896     printf("b. Move wire to specified location;\n");
897     printf("c. Exit Program.\n");
898     printf("Please enter the corresponding letter and press ENTER key:    ");
899     fflush(stdout);
900
901     switch (menu(3)) {
902         case  0: {gcode_menu();                                  break; }
903         case  1: {move_menu();                                   break; }
904         case  2: {state_ = EXITING;                              break; }
905         case -1: {printf("Invalid option. Let's try again.\n\n"); break; }
906         case -2: {                                               break; }
907     }
908     return;
909 }
910
911 void gcode_menu() {
912     if (state_ != EXITING && (current_position_.LX || \
913         current_position_.LY || current_position_.RX || current_position_.RY)){
914         printf("I see the wire is not at origin. Plesse press ENTER to move wire to origin:    ");
915         fflush(stdout);
916         if (menu_enter() != -2) {
917             moveto(0.0,0.0);
918             while (state_ != EXITING && !allreached()){ nsleep(1000000);}
919             printf("Origin Reached\n\n");
920         }
921     }
922
923     gcode_menu_option_ = -1;
924     int n = 10;
925     while (state_ != EXITING && gcode_menu_option_ == -1){
926         printf("****************************** GCODE MENU ******************************\n");
927         // keep looping when menu selection is invalid
928         struct dirent **namelist;
929         n = scandir("/home/pi/", &namelist, file_filter, alphasort);
930         // scan for files with .txt extension
931         if (n == 0) {
932             printf("I do not see any txt files in '/home/pi/' directory\n");
933             printf("Please put the desired gcode file in '/home/pi/' directory.\n");
934             printf("Returning to Main Menu.\n\n");
935             break;
936         }
937         else if (n > 9){
938             printf("Too many txt files in '/home/pi/' directory\n");
939             printf("Please clean it up.\n");
940             printf("Returning to Main Menu.\n\n");
941             break;
942         }
943         else{
944             printf("I see there are %d txt files listed below:\n\n",n);
945             int i = 0;
946             while (i++ < n){
947                 printf("%d:   %s\n", i, namelist[i-1]->d_name);
948             }
949             printf("0:   None of the above, or Return to Main Menu\n\n");
950             printf("Please select one by entering the corresponding number then press ENTER:    ");
951             fflush(stdout);
952             gcode_menu_option_ = menu(n+1);
953
954             if (gcode_menu_option_ >= 1) {
955                 printf("You selected:  '%s'   is that correct?\n",namelist[gcode_menu_option_-1]->d_name);
956                 switch (menu_yes()){
957                 case 1:
```

```
                    if (state_ != EXITING && !loadtext(namelist[gcode_menu_option_-1]->d_name) ) {
                        if (state_ != EXITING) cut_gcode(namelist[gcode_menu_option_-1]->d_name);
                        return;
                    }
                    break;
                case 0:
                    printf("OK, Let try again\n\n");
                    gcode_menu_option_ = -1;
                    break;
                }
            }
            else if (gcode_menu_option_ == 0 ) {
                printf("Please put the desired gcode file in the working directory.\n");
                printf("Returning to Main Menu.\n\n");
                break;
            }
            else if (gcode_menu_option_ == -1 ) { // Invalid input
                printf("Invalid Input, let's try again.\n\n");
            }
            else if (gcode_menu_option_ == -2 ) { // EXITING state
                break;
            } // end of if --- menu input check
        } // end of if --- file number check
    } // end of while
    return;
} // end of gcode_menu

void move_menu(){
    float x,y;
    while (state_ != EXITING){
        float current_x = (current_position_.LX + current_position_.RX)/2.0/MM2PULSE;
        float current_y = (current_position_.LY + current_position_.RY)/2.0/MM2PULSE;
        printf("****************************** MOVE MENU ******************************\n");
        printf("The current wire position is (x,y) = (%.3f,%.3f) in = (%.3f,%.3f) mm\n", \
            current_x*MM2IN,current_y*MM2IN,current_x,current_y);
        printf("Please choose from the following options:\n");
        printf("a. Move wire to a specific location relative to origin (Absolute Location);\n");
        printf("b. Move wire to a specific location relative to current position (Increment);\n");
        printf("c. Move wire to origin;\n");
        printf("d. Return to Main Menu;\n");
        printf("Please enter the corresponding letter and press ENTER key:    ");
        fflush(stdout);
        switch (menu(4)) {
            case  0: { // Move to a specific location
                while(state_ != EXITING && \
                    (menu_enter_two(&x,&y,"Please enter the destination x and y coordinates RELATIVE TO ORIGIN") == -1 \
                        || x < 0 || y <0)) {
                    printf("Invalid input, please enter again. Please note that negative destination is not allowed.\n\n");
                }
                if (state_ != EXITING) {
                    printf("|      |       Current      |      Increment     |     Destination    |\n");
                    printf("| Inch |  %7.3f,  %7.3f |  %7.3f,  %7.3f |  %7.3f,  %7.3f |\n", \
                        current_x*MM2IN, current_y*MM2IN, (x-current_x)*MM2IN, (y-current_y)*MM2IN, x*MM2IN, y*MM2IN);
                    printf("|  MM  | %8.3f, %8.3f | %8.3f, %8.3f | %8.3f, %8.3f |\n", \
                        current_x, current_y, (x-current_x), (y-current_y), x, y);
                    printf("Continue?\n");
                    switch (menu_yes()){
                    case 1:
                        if (state_ != EXITING ) {
                            printf("Moving ... ..."); fflush(stdout);
                            moveto(x,y);
                            while (state_ != EXITING && !allreached()){ nsleep(1000000);}
                            if (state_ != EXITING) printf(" Destination Reached\n\n");
                        }
                        break;
                    case 0:
                        printf("OK, Let try again\n\n");
                        break;
                    }
                }
                break; } // end case --- move to a specific location

            case  1: { // Move to a specific location
                while(state_ != EXITING && \
                    (menu_enter_two(&x,&y,"Please enter the destination x and y coordinates RELATIVE TO CURRENT POSITION") == -1 \
                        || x+current_x < 0 || y+current_y <0)) {
                    printf("Invalid input, please enter again. Please note that negative destination is not allowed.\n\n");
                }
                if (state_ != EXITING) {
                    printf("|      |       Current      |      Increment     |     Destination    |\n");
                    printf("| Inch |  %7.3f,  %7.3f |  %7.3f,  %7.3f |  %7.3f,  %7.3f |\n", \
                        current_x*MM2IN, current_y*MM2IN, x*MM2IN, y*MM2IN, (x+current_x)*MM2IN, (y+current_y)*MM2IN);
                    printf("|  MM  | %8.3f, %8.3f | %8.3f, %8.3f | %8.3f, %8.3f |\n", \
                        current_x, current_y, x, y, (x+current_x), (y+current_y));
                    printf("Continue?\n");
                    switch (menu_yes()){
                    case 1:
```

```
                        if (state_ != EXITING ) {
                            printf("Moving ... ..."); fflush(stdout);
                            moveto(x+current_x,y+current_y);
                            while (state_ != EXITING && !allreached()){ nsleep(1000000);}
                            if (state_ != EXITING) printf(" Destination Reached\n\n");
                        }
                        break;
                    case 0:
                        printf("OK, Let try again\n\n");
                        break;
                    }
                }
                break; } // end case --- move to a specific location

            case  2: { //case --- move to origin
                if (!current_position_.LX && !current_position_.LY && \
                    !current_position_.RX && !current_position_.RY) {
                    printf("Already at Origin\n\n");
                }
                else{
                    printf("Move to Orignin. Continue?\n");
                    switch (menu_yes()){
                    case 1:
                        printf("Moving ... ..."); fflush(stdout);
                        moveto(0.0,0.0);
                        while (state_ != EXITING && !allreached()){ nsleep(1000000);}
                        if (state_ != EXITING) printf(" Origin Reached\n\n");
                        break;
                    case 0:
                        printf("OK, Let try again\n\n");
                        break;
                    }
                }
                break; } // end case --- move to origin
            case -1: {printf("Invalid option. Let's try again.\n\n"); break; }
            default: {return;                                        break; }
        } // end switch
    } // end while
    return;
}

int menu(int numb_of_options){
    // pass in number of menu options, maximum of 10 options
    // return 0 to (numb_of_options - 1) if input is within the range
    // return -1 for invalid input
    // return -2 when state_ is exiting

    if (state_ ==  EXITING) return -2;
    fd_set          input_set;
    struct timeval  timeout;
    timeout.tv_sec = 10;
    timeout.tv_usec = 0;

    // Listening for input stream for any activity
    // If there
    FD_ZERO(&input_set );
    FD_SET(0, &input_set);
    while (state_ !=  EXITING && !select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 1;
        FD_ZERO(&input_set );
        FD_SET(0, &input_set);
    }

    if (state_ ==  EXITING) return -2;

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // determine length of input
    int i; for(i=0; input_option[i]!='\0'; i++); i --;
    int result;

    if (i == 1) {
        if     (input_option[0] >= 48 && input_option[0] <= 57) {
            result = input_option[0]-48;
        }
        else if(input_option[0] >= 65 && input_option[0] <= 74) {
            result = input_option[0]-65;
        }
        else if(input_option[0] >= 97 && input_option[0] <= 106) {
            result = input_option[0]-97;
        }
        else result = -1;
    }
    else result = -1;
    if (result >= numb_of_options) result = -1;

    printf("\n");
```

```
    return result;
}

int menu_enter(){
    // return 1 if anthing is entered, including ENTER key
    // return 0 if m is entered
    // return -2 when state_ is exiting
    if (state_ ==  EXITING) return -2;
    fd_set          input_set;
    struct timeval  timeout;
    timeout.tv_sec = 10;
    timeout.tv_usec = 0;

    // Listening for input stream for any activity
    FD_ZERO(&input_set );
    FD_SET(0, &input_set);
    while (state_ !=  EXITING && !select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 1;
        FD_ZERO(&input_set );
        FD_SET(0, &input_set);
    }

    if (state_ ==  EXITING) return -2;

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // determine length of input
    int i; for(i=0; input_option[i]!='\0'; i++); i --;

    if (i == 1 && (input_option[0] == 'm' || input_option[0] == 'M')) {
        printf("\n"); return 0; // return 0 if input is 'm' or 'M'
    }
    else {printf("\n"); return 1; }
}

int menu_yes(){
    // return 1 if anthing is entered, including ENTER key
    // return 0 if n is entered
    // return -2 when state_ is exiting
    if (state_ ==  EXITING) return -2;
    printf("Press ENTER for YES, or 'n' then ENTER for NO:    "); fflush(stdout);
    fd_set          input_set;
    struct timeval  timeout;
    timeout.tv_sec = 10;
    timeout.tv_usec = 0;

    // Listening for input stream for any activity
    FD_ZERO(&input_set );
    FD_SET(0, &input_set);
    while (state_ !=  EXITING && !select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 1;
        FD_ZERO(&input_set );
        FD_SET(0, &input_set);
    }

    if (state_ ==  EXITING) return -2;

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // determine length of input
    int i; for(i=0; input_option[i]!='\0'; i++); i --;

    if (i == 1 && (input_option[0] == 'n' || input_option[0] == 'N')) {
        printf("\n"); return 0; // return 0 if input is 'm' or 'M'
    }
    else {printf("\n"); return 1;}
}

int menu_enter_one(float* output, char* string){
    float scale = 1.0;
    if (state_ ==  EXITING) return -2;
    printf("Would you like to enter the coordinate in inches?\n");
    switch (menu_yes()) {
        case 1: scale = 25.4; printf("%s in inches then press ENTER:    ", string); break;
        case 0: scale =  1.0; printf("%s in millimeters then press ENTER:    ", string); break;
    }
    fflush(stdout);
    if (state_ ==  EXITING) return -2;


    fd_set          input_set;
    struct timeval  timeout;
    timeout.tv_sec = 10;
    timeout.tv_usec = 0;

    // Listening for input stream for any activity
    // If there
```

```
    FD_ZERO(&input_set );
    FD_SET(0, &input_set);
    while (state_ !=  EXITING && !select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 1;
        FD_ZERO(&input_set );
        FD_SET(0, &input_set);
    }

    if (state_ ==  EXITING) return -2;

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // remove white spaces
    removespace(input_option);
    float temp;
    if (str2f(input_option, &temp)) {
        *output = temp*scale;
        printf("\n"); return 1;
    }
    else {printf("\n"); return -1;}
}

int menu_enter_two(float* output1, float* output2, char* string){
    float scale = 1.0;
    if (state_ ==  EXITING) return -2;
    printf("Would you like to enter the coordinate in inches?\n");
    switch (menu_yes()) {
        case 1:
        scale = 25.4;
        printf("%s in INCHES \nseparated with comma then press ENTER:    ", string);
        break;
        case 0:
        scale =  1.0;
        printf("%s in MM \nseparated with comma then press ENTER:    ", string);
        break;
    }
    fflush(stdout);
    if (state_ ==  EXITING) return -2;

    fd_set          input_set;
    struct timeval  timeout;
    timeout.tv_sec = 10;
    timeout.tv_usec = 0;

    // Listening for input stream for any activity
    // If there
    FD_ZERO(&input_set );
    FD_SET(0, &input_set);
    while (state_ !=  EXITING && !select(1, &input_set, NULL, NULL, &timeout)) {
        timeout.tv_sec = 1;
        FD_ZERO(&input_set );
        FD_SET(0, &input_set);
    }
    if (state_ ==  EXITING) return -2;

    // get input
    char input_option[256]; fgets(input_option,256,stdin);
    // remove white spaces
    removespace(input_option);
    // get the two
    char* ptr = strchr(input_option, ',');
    if (ptr == NULL) {return -1;}
    char temp1[20]; strncpy(temp1,input_option,ptr-input_option);
    temp1[(ptr-input_option)] = '\0';
    char temp2[20]; strncpy(temp2,ptr+1,20);
    float temp1f, temp2f;
    if (str2f(temp1, &temp1f) && str2f(temp2, &temp2f)) {
        *output1 = temp1f*scale;
        *output2 = temp2f*scale;
        printf("\n"); return 1;
    }
    else {return -1;}
}

/*****************************************************************
************************* OTHER FUNCTIONS ************************
*****************************************************************/

// Sleep for nanoseconds
void nsleep(uint64_t ns){
    struct timespec req,rem;
    req.tv_sec = ns/1000000000;
    req.tv_nsec = ns%1000000000;
    // loop untill nanosleep sets an error or finishes successfully
    errno=0; // reset errno to avoid false detection
    while(nanosleep(&req, &rem) && errno==EINTR){
        req.tv_sec = rem.tv_sec;
```

```
        req.tv_nsec = rem.tv_nsec;
    }
    return;
}

// Filter out file without .txt extension
int file_filter(const struct dirent *entry){
    return !strcmp(entry->d_name + strlen(entry->d_name) -4, ".txt");
}

// Remove space in a string
void removespace(char* str) {
    int i,j=0;
    for(i=0;str[i]!='\0';i++) {
        if(str[i]!=' ' && str[i] != 10 && str[i] != 13)
            str[j++]=str[i];
    }
    str[j]='\0';
    return;
}

// Signal Handler for CTRL+C
void SigHandler(int dummy) {
    state_ = EXITING; stop_all();
    printf("\n\n************************** EXITING PROGRAM **************************\n");
    return;
}

float max(float a, float b) {
    if(a >= b) return a;
    else return b;
}

float min(float a, float b) {
    if(a >= b) return b;
    else return a;
}

void print_time(int sec){
    if (sec <=  0) printf("00:00");
    else printf("%02d:%02d", (int)floor(sec /60.0),sec % 60);
    return;
}

int str2f(char* str, float* output){
    removespace(str);
    float out = 0;
    int dec_location;
    int dec_numb = 0;
    int i; for(i=0; str[i] !='\0'; i++);
    int length = i;
    int j =1;
    for(int i = length-1; i >= 0; i--) {
        if (str[i]>=48 && str[i]<=57){out = out+j*(str[i]-48); j = j*10;}
        else if (str[i] == 46) {dec_numb ++; dec_location = i;}
        else if (i == 0 && str[i] == 43) {out = +out;}
        else if (i == 0 && str[i] == 45) {out = -out;}
        else return 0;
    }
    if (dec_numb == 0) {out = out;}
    else if (dec_numb == 1) {out = out/pow(10.0,length-1-dec_location);}
    else {return 0;}
    *output = out;
    return 1;
}
```

2 foamcutter_setup.h

The setting file, **foamcutter_setup.h** is attached below.

```c
/*******************************************************************************
* foamcutter_setup.h
*******************************************************************************/

#ifndef FOAMCUTTER_SETUP
#define FOAMCUTTER_SETUP

#include <pthread.h>
#include <signal.h>
#include <unistd.h>
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <dirent.h>
#include <time.h>
#include <errno.h>
#include <math.h>
#include <sys/resource.h>
#include <sys/types.h>

// Cutter Width
#define CUTTERWIDTH (36.0*IN2MM) // inches

// Unit Conversion
#define IN2MM 25.4
#define MM2IN (1.0/25.4)
#define IN2REV 10.0 // Lead screw 10 TPI
#define REV2PULSE 400.0 // depend on the switch settings on the motor drive
#define MM2PULSE ( IN2REV / IN2MM * REV2PULSE) // 157.48 PULSE = 1 mm

// Settings
#define FEEDRATE 0.8 // unit: mm/s
#define FEEDRATE_PUL (FEEDRATE*MM2PULSE)

// position of limit switch relative to cutter origin
#define LIM2ORIGIN_LX -4100
#define LIM2ORIGIN_RX -3550
#define LIM2ORIGIN_LY -2500
#define LIM2ORIGIN_RY -2400

// Motor Drive Pins
#define PIN_LX_DIR 14
#define PIN_LX_PUL 15
#define PIN_LY_DIR 18
#define PIN_LY_PUL 23
#define PIN_RX_DIR 24
#define PIN_RX_PUL 25
#define PIN_RY_DIR 8
#define PIN_RY_PUL 7

// Limit Switches
#define PIN_LX_LIM 6
#define PIN_LY_LIM 13
#define PIN_RX_LIM 19
#define PIN_RY_LIM 26

// Relay
#define PIN_RELAY 12

// Buttons
#define PIN_PAUSE 17
#define PIN_STOP 27

// Motor Polarity
// set to +1 or -1
// reverse the setting if the wire moves away from the limit switch during homing.
#define POLARITY_LX +1
#define POLARITY_LY -1
#define POLARITY_RX +1
#define POLARITY_RY -1

// MAX cut area
#define X_MAX (29.0*IN2MM)  // 29 in
#define Y_MAX (16.0*IN2MM)  // 16 in

#endif  //FOAMCUTTER_SETUP
```

Appendix B

MatLab Code for G-code Generation

1 DBF_foamcutter_general_shape.m

The MatLab code for generating G-code for general shape, **DBF_foamcutter_general_shape.m** is attached below. Last updated: 8/26/2018.

```
%% DBF Foamcutter for Genearl Shapes
% This code is written by Yuting Huang (ythuang96@gmail.com);
% Please report all bug to the author's email address.
% Last updated: 8/26/2018

% This is written for DBF foamcutting, to generate G-code from general shape
% AutoCAD drawings.

%% User Manual
% 1. Export lines and arcs form AutoCAD, save as csv file.
% 2. Copy the csv file to the same folder as this MatLab code.
% 3. Run Code and done!
% Press CRT+C at anytime to terminate code.

%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
clear all; close all; clc;
tolerance = 0.0002;
accuracy = 2; % length in mm of segments when breaking arc
%% Determine Units
% GUI stuff
UIControl_FontSize_bak = get(0, 'DefaultUIControlFontSize');
set(0, 'DefaultUIControlFontSize', 30);
unit = menu('Is Drawing in millimeters?','Yes','No');
if unit == 1; % if drawing is in mm, continue generation of G-code
    %% Check the current folder for csv files
    D = dir('*.csv');
    if ~length(D) % if no .csv file exsist print error message
        fprintf('I could not find any file with .csv');
        fprintf(' estension in the current folder.\n');
        fprintf('Please move the .csv file created by AutoCAD ');
        fprintf('''eattext'' command into the current ');
        fprintf('working folder and try again.\n');
    else
        % Create a menu to select csv files in the current folder
        string = ['file = menu(''I detected ' num2str(length(D)) ...
            ' csv files list below, please select one'',' ];
        for i = 1:length(D); string = [string '''' D(i).name ''','']; end
        string = [string '''None of the above'');'];
        eval(string);
        if file <= length(D);
            filename = D(file).name(1:end-4); clear  string;
            if file <= length(D) && file;
                %%  Inport File
                inport = csvread([filename  '.csv'],1,2);
                [m,n] = size(inport);
                % make changes if there are only lines
                if n ==4; inport = [zeros(m,5) , inport]; end
                %% Eliminate 0 length lines
                k = 1;
                for i = 1:size(inport,1);
                    if any(inport(i,6:9)~= [0 0 0 0]) && ...
```

```
                    all(inport(i,6:7) == inport(i,8:9));
                m = m-1;
            else temp(k,:) = inport(i,:); k = k+1;
            end
        end
        inport = temp; clear temp k i D file;
        %% Seperate Arc With line
        n_arc = 0; n_line = 0;
        for i = 1:m;
            if all(inport(i,6:9) == [0 0 0 0]);
                n_arc = n_arc + 1; arc(n_arc,:) = inport(i,1:5);
            else n_line = n_line + 1;
                line(n_line,:) = inport(i,6:9);
            end
        end
        %% Break Arcs into lines
        alllines = line;
        for i = 1:n_arc;
            n_segment = ceil(2*pi*arc(i,3)*arc(i,5)/360/accuracy);
            dtheta = arc(i,5)/n_segment;
            arcpoints = zeros(n_segment+1,2);
            for j = 1:n_segment+1;  % break arc into points
                theta = arc(i,4) + (j-1)*dtheta;
                arcpoints(j,:) = arc(i,1:2) + ...
                    arc(i,3).*[cosd(theta),sind(theta)];
            end
            % chage the start and end point so that the arc join
            % the lines
            for j = 1:n_line;
                if abs(arcpoints(1,:) - line(j,1:2)) <= 0.01;
                    arcpoints(1,:) = line(j,1:2);
                elseif abs(arcpoints(1,:) - line(j,3:4)) <= 0.01;
                    arcpoints(1,:) = line(j,3:4);
                end
                if abs(arcpoints(end,:) - line(j,1:2)) <= 0.01;
                    arcpoints(end,:) = line(j,1:2);
                elseif abs(arcpoints(end,:) - line(j,3:4)) <= 0.01;
                    arcpoints(end,:) = line(j,3:4);
                end
            end
            % put all lines with arc points together
            alllines = [alllines ; arcpoints(1:end-1,:) , ...
                arcpoints(2:end,:)];
        end
        %% Sort the lines in order
        sort(1,:) = alllines(1,:);
        alllines(1,:) = [];
        for i = 2:size(alllines,1)+1;
            compare = sort(i-1,3:4);
            [n2,~] = size(alllines);
            for j = 1:n2;
                if all(abs(compare - alllines(j,1:2)) <= tolerance);
                    sort(i,:) = alllines(j,:);
                    alllines(j,:) = []; check = 1;  break;
```

```
109                         elseif all(abs(compare - alllines(j,3:4)) <= tolerance);
110                             sort(i,1:2) = alllines(j,3:4);
111                             sort(i,3:4) = alllines(j,1:2);
112                             alllines(j,:) = []; check = 1;  break;
113                         end
114                     end
115                     if ~check
116                         % cannot find the another line that connects with
117                         % the previous
118                         fprintf('There is an open countour.\n');
119                         fprintf('This is most likely caused by an ');
120                         fprintf('extra line underneath a long line.\n');
121                         fprintf('Please check your drawing.\n');
122                         return;
123                     end
124                     check = 0;
125                 end
126                 sort2 = [sort(:,1:2); sort(end,3:4)];
127                 %% Shift to positive
128                 min_x = min(sort2(:,1)); min_y = min(sort2(:,2));
129                 sort2(:,1) = sort2(:,1) - min_x;
130                 sort2(:,2) = sort2(:,2) - min_y;
131                 max_x = max(sort2(:,1)); max_y = max(sort2(:,2));
132                 %% Plot Curve
133                 figure(1); set(1,'position',[0 0 1920 1080]); hold on;
134                 plotx = sort2(:,1); ploty = sort2(:,2); plot(plotx,ploty );
135                 title('Drawing Unit mm','fontsize',30);
136                 axis equal;
137                 %% Plot number
138                 [n,~] = size(sort2);
139                 j = 1; index = [];
140                 for i = 1:n-1;
141                     if sort2(i,1) == 0 || sort2(i,1) == max_x  ...
142                         || sort2(i,2) == 0 || sort2(i,2) == max_y;
143                         text(plotx(i),ploty(i),sprintf('%d',j),'fontsize',20);
144                         j = j+1; index = [index, i];
145                     end
146                 end
147                 hold off;
148                 %% Determine Start Point
149                 start = index(input('Which point would you like to start?    '));
150                 sort3 = [sort2(start:end-1,:) ; sort2(1:start-1,:);  ...
151                     sort2(start,:)];
152                 %% Cut Direction Reverse if chosen to
153                 direction = menu(['The Current Cut Direction is shown in', ...
154                     'the Figure with Increasing Number,', ...
155                     'Reverse Cut Direction?','NO','YES']);
156                 if direction == 2; final = rot90(sort3',1);
157                 elseif direction == 1; final = sort3;  end
158                 %% Final Plot
159                 clf; hold on;
160                 finalx = final(:,1); finaly = final(:,2);
161                 plot(finalx,finaly);
162                 title('Final shape on Foam Cutter, Drawing Unit mm', ...
```

```
                    'fontsize',30);
                j = 1;
                for i = 1:n
                    if final(i,1) == 0 || final(i,1) == max_x  ...
                        || final(i,2) == 0 || final(i,2) == max_y;
                        text(finalx(i),finaly(i),sprintf( '%d',j),'fontsize',20);
                        j = j+1;
                        if j == 4; break; end
                    end
                end
                axis equal; hold off;
                set(0, 'DefaultUIControlFontSize', UIControl_FontSize_bak);
                %% Generate G-code
                fidw = fopen([filename  '.txt'],'wt');
                fprintf(fidw,'G21\n'); fprintf(fidw,'M49\n');
                fprintf(fidw,'F80\n'); fprintf(fidw,'S80\n');
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,0,0,0,0);
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ,...
                    finalx(1),finaly(1),finalx(1),finaly(1));
                for i = 2:length(finalx)-1
                    fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                        ,finalx(i),finaly(i),finalx(i),finaly(i));

                    % Calculate Length
                    length = sqrt((finalx(i)-finalx(i-1))^2 +  ...
                    (finaly(i)-finaly(i-1))^2);
                    if length >= 100;
                        fprintf(fidw,sprintf( 'G4 P%d\n',floor(length/100) ) );
                    end
                    % Add 1 sec pause per 100 mm cut for long cuts
                end
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,finalx(end),finaly(end),finalx(end),finaly(end));
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,0,0,0,0);
                fprintf(fidw,'M2');
                fclose(fidw);
                disp(['The G-Code is saved as  ''' filename ...
                    '.txt''  in this folder. ']);
                set(gcf,'PaperUnits','inches','PaperPosition',[0 0 16 9]);
                print(filename,'-dpng','-r240');
            else % if 'none of the above selectee'
                fprintf('Please move your desired file to the current' );
                fprintf(' folder and try again.\n');
            end
        else % if no file selected
            fprintf('Please move your desired file to the current' );
            fprintf(' folder and try again.\n');
        end % end file selection check
    end  % end file exsistence check
else % if drawing not in mm, print error message
    fprintf('Please go back to AutoCAD and use the ''Scale''' );
    fprintf(' command to scale the drawing by 25.4. \n' );
```

```
217     fprintf('Inches do not provide high enough accuracy\n');
218 end % end 'unit' check
```

2 DBF_foamcutter_wing.m

The MatLab code for generating G-code for wing, **DBF_foamcutter_wing.m** is attached below. Last updated: 8/23/2018.

```
%% DBF Foamcutter for Wing
% This code is modified by Yuting Huang (ythuang96@gmail.com) based on
% Dr.Anderson's code written on Scilab.
% Please report all bug to the author's email address.
% Last updated: 8/23/2018

% This is written for DBF foamcutting, to generate G-code from wing
% prameters.
clear all; close all; clc;

%% Enter Parameters Below
% airfoil section file for root
root_filename = 'AH-79-100.dat';
% root chord length [in]
root_chord = 6;

% airfoil section file for tip
tip_filename = 'E216.dat';
% tip chord length [in]
tip_chord = 3;
% root chord has to be greater or equal to tip chord

% +1 for right wing, -1 for left wing
right_wing = 1;

% semi-span [in]
semi_span = 3;

% leading edge sweep [deg]
LE_sweep = 0;

% twist [deg]
twist = 0.0;

% g-code output file name
g_filename = 'HTail';

% width of CNC cutter [in]
cutter_width = 39;
% scale the cord length to accomodaate for broken trailing edge
% recommend using 1.2, then cut trailing edge with a blade to desired
% length.
scale_factor = 1.2;

%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% Open File
fid = fopen(root_filename);
temp = textscan(fid,'%f %f','headerLines', 1);
fclose(fid);
root_pts = [temp{1}, temp{2}];

```

```
fid = fopen(tip_filename);
temp = textscan(fid,'%f %f','headerLines', 1);
fclose(fid);
tip_pts = [temp{1}, temp{2}];
clear temp fid;

%% Deconstruct Airfoil into Upper and Lower Parts
root_size = size(root_pts,1);
root_turn_point = find(root_pts(:,2)<0,1) - 1;
root_upper_x = root_pts(1:root_turn_point, 1);
root_upper_y = root_pts(1:root_turn_point, 2);
root_lower_x = root_pts(root_turn_point:root_size, 1);
root_lower_y = root_pts(root_turn_point:root_size, 2);

tip_size = size(tip_pts,1);
tip_turn_point = find(tip_pts(:,2)<0,1) - 1;
tip_upper_x = tip_pts(1:tip_turn_point, 1);
tip_upper_y = tip_pts(1:tip_turn_point, 2);
tip_lower_x = tip_pts(tip_turn_point:tip_size, 1);
tip_lower_y = tip_pts(tip_turn_point:tip_size, 2);

%% Interpolate
n = 301;
root_upper_yp = interp1(root_upper_x,root_upper_y,linspace(1,0,n)');
root_upper_xp = linspace(1,0,n)';
root_lower_yp = interp1(root_lower_x,root_lower_y,linspace(0,1,n)');
root_lower_xp = linspace(0,1,n)';

tip_upper_yp = interp1(tip_upper_x,tip_upper_y,linspace(1,0,n)');
tip_upper_xp = linspace(1,0,n)';
tip_lower_yp = interp1(tip_lower_x,tip_lower_y,linspace(0,1,n)');
tip_lower_xp = linspace(0,1,n)';

%% Scale to Chord Length
% chord length on machine
root_chord_ext = root_chord + ...
    0.5*(root_chord - tip_chord)*(cutter_width - semi_span)/semi_span;
tip_chord_ext = root_chord - ...
    0.5*(root_chord - tip_chord)*(cutter_width + semi_span)/semi_span;

root_upper_xp = root_chord_ext*root_upper_xp;
root_upper_yp = root_chord_ext*root_upper_yp;

root_lower_xp = root_chord_ext*root_lower_xp;
root_lower_yp = root_chord_ext*root_lower_yp;

tip_upper_xp = tip_chord_ext*tip_upper_xp;
tip_upper_yp = tip_chord_ext*tip_upper_yp;

tip_lower_xp = tip_chord_ext*tip_lower_xp;
tip_lower_yp = tip_chord_ext*tip_lower_yp;

%% Rotate Tip by Twist Angle
c4 = tip_chord/4;
```

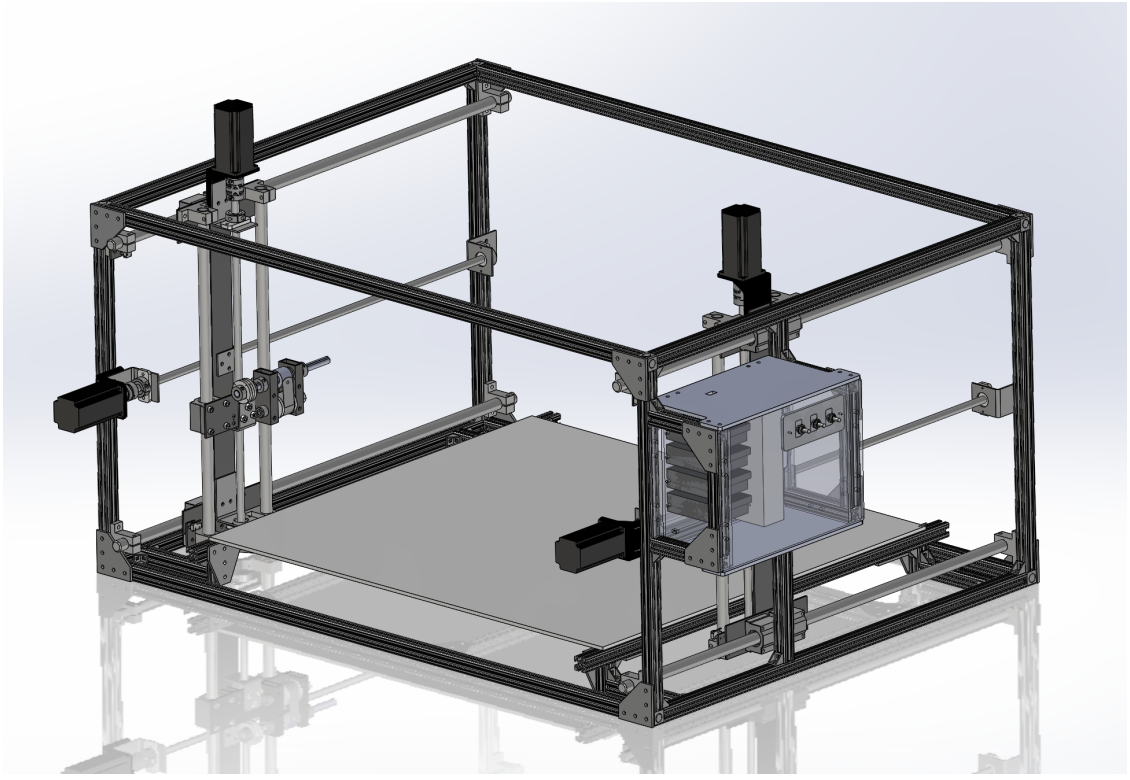
```
109 twist_rad = (pi/180)*twist;
110
111 tip_upper_xpr = tip_upper_xp*cos(twist_rad)  ...
112     + tip_upper_yp*sin(twist_rad) + c4*(1.0-cos(twist_rad));
113 tip_upper_ypr = -tip_upper_xp*sin(twist_rad)  ...
114     + tip_upper_yp*cos(twist_rad) + c4*sin(twist_rad);
115
116 tip_lower_xpr = tip_lower_xp*cos(twist_rad)  ...
117     + tip_lower_yp*sin(twist_rad) + c4*(1.0-cos(twist_rad));
118 tip_lower_ypr = -tip_lower_xp*sin(twist_rad)  ...
119     + tip_lower_yp*cos(twist_rad) + c4*sin(twist_rad);
120
121 %% Use Sweep Angle to Shift Tip Relative to Root
122 sweep_shift = cutter_width*tan(LE_sweep*pi/180);
123
124 tip_upper_xpr = tip_upper_xpr + sweep_shift;
125 tip_lower_xpr = tip_lower_xpr + sweep_shift;
126
127 %% Swap x-axis to Start Cut on Trailing Edge
128 root_upper_xp = root_chord_ext - root_upper_xp;
129 root_lower_xp = root_chord_ext - root_lower_xp;
130 tip_upper_xpr = root_chord_ext - tip_upper_xpr;
131 tip_lower_xpr = root_chord_ext - tip_lower_xpr;
132
133 %% Combine Upper and Lower Surfaces
134 root_x = [root_upper_xp; root_lower_xp];
135 root_y = [root_upper_yp; root_lower_yp];
136
137 tip_x = [tip_upper_xpr; tip_lower_xpr];
138 tip_y = [tip_upper_ypr; tip_lower_ypr];
139
140 %% Make a Plot of Root and Tip
141 set(0,'defaultlinelinewidth',2)
142 set(0,'defaultaxeslinewidth',1)
143 set(0,'defaultaxesfontsize',20)
144
145 figure(1); set(1,'position',[0 0 1920 1080]); hold on;
146 plot(root_x,root_y);
147 plot(tip_x,tip_y);
148 legend1 = legend('Root','Tip');
149 set(legend1,'interpreter','latex'); set(legend1,'fontsize',18);
150 title('Wing on FoamCutter','interpreter','latex','fontsize',25);
151 xlabel('X [in]','interpreter','latex','fontsize',25);
152 ylabel('Y [in]','interpreter','latex','fontsize',25);
153 axis equal; grid on;
154
155 %% Write G-code File
156 root_x = scale_factor*root_x*25.4;
157 root_y = scale_factor*root_y*25.4;
158 tip_x = scale_factor*tip_x*25.4;
159 tip_y = scale_factor*tip_y*25.4;
160
161 fidw = fopen([g_filename '.txt'],'wt');
162
```

```
163 fprintf(fidw,'G21\n'); fprintf(fidw,'M48\n');
164 fprintf(fidw,'F80\n'); fprintf(fidw,'S80\n');
165
166 fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n',0,0,0,0);
167
168 if right_wing > 0
169     fprintf(fidw, 'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n', ...
170         root_x(1),root_y(1),tip_x(1),tip_y(1));
171     fprintf(fidw, 'G4 P5\n');
172     for i=length(root_x):-1:1
173         fprintf(fidw, 'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n', ...
174             root_x(i),root_y(i),tip_x(i),tip_y(i));
175     end
176 else
177     fprintf(fidw, 'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n', ...
178         tip_x(1),tip_y(1),root_x(1),root_y(1));
179     fprintf(fidw, 'G4 P5\n');
180     for i=length(root_x):-1:1
181         fprintf(fidw, 'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n', ...
182             tip_x(i),tip_y(i),root_x(i),root_y(i));
183     end
184 end
185
186 fprintf(fidw, 'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n',0,0,0,0);
187 fprintf(fidw,'M2');
188
189 fclose(fidw);
190
191
```

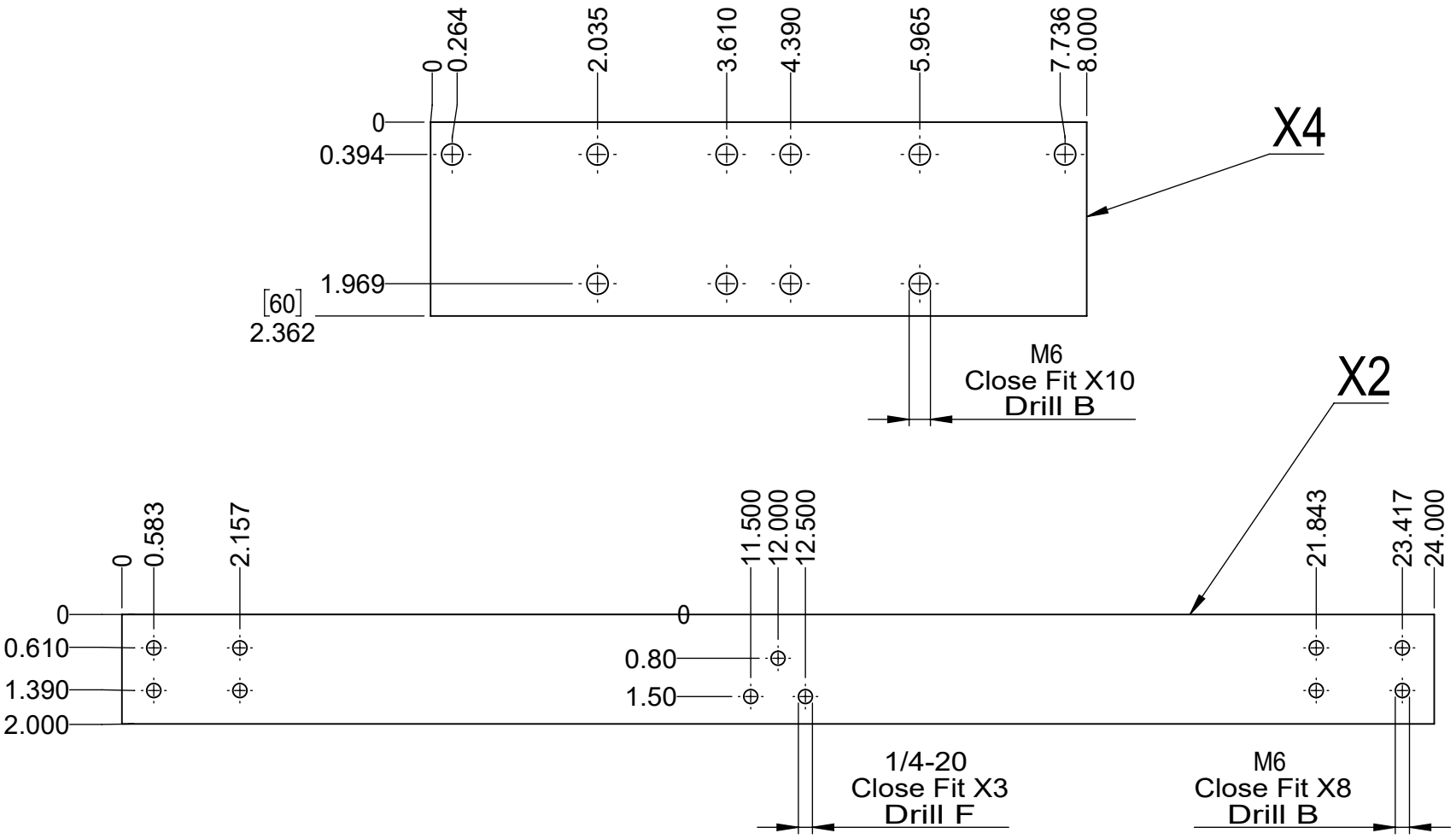
Appendix C

Drawings

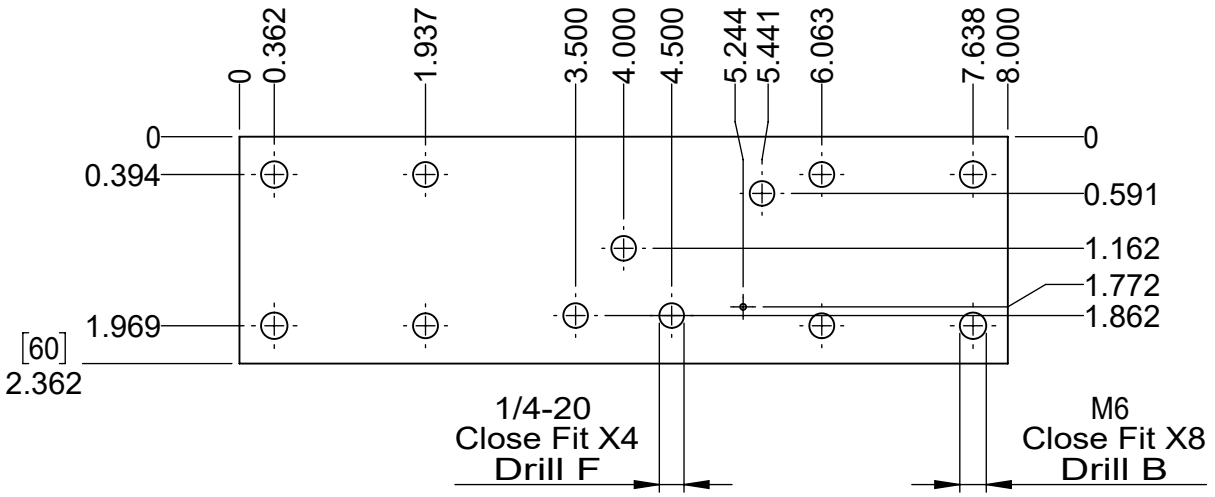
The complete SolidWorks drawings can be found at https://github.com/ythuang96/FoamCutter. The following is the complete assembly in SolidWorks:

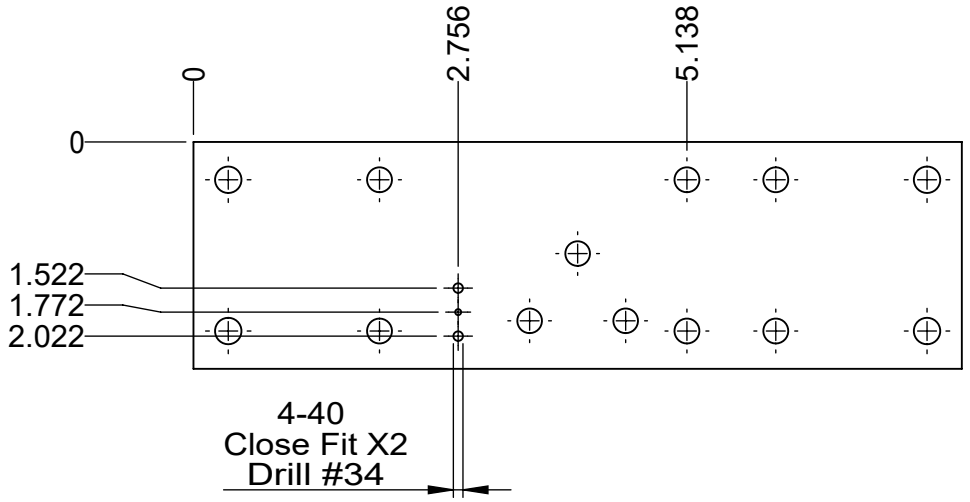


Drawings attached below are all aluminum pieces that required machining. But the drawings are not drawn in compliance with engineering standards and therefore should only be used as a reference when machining yourself.



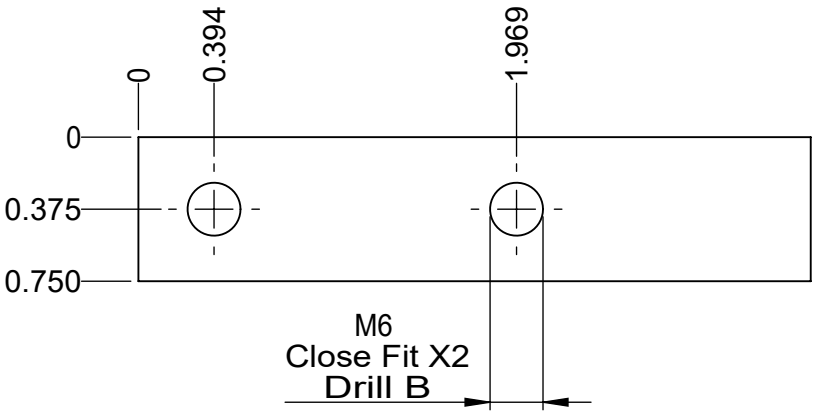
TITLE: Al Sheet 1 Al Sheet 2		
SIZE A	DWG. NO. 1	REV
SCALE: 1:2	WEIGHT:	SHEET 1 OF 11

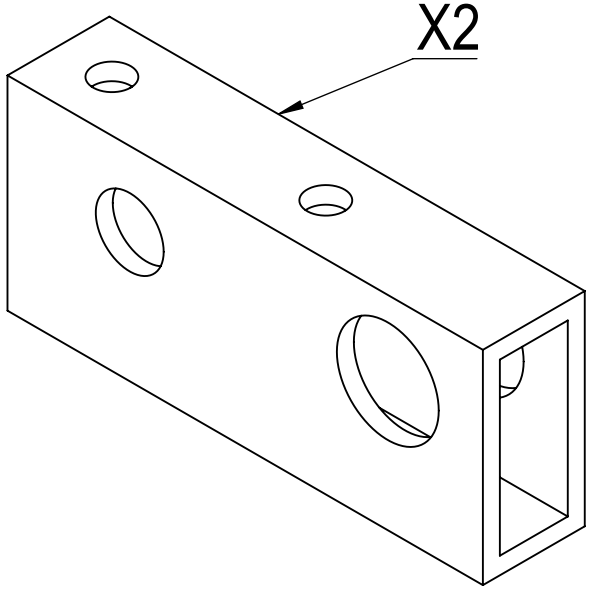


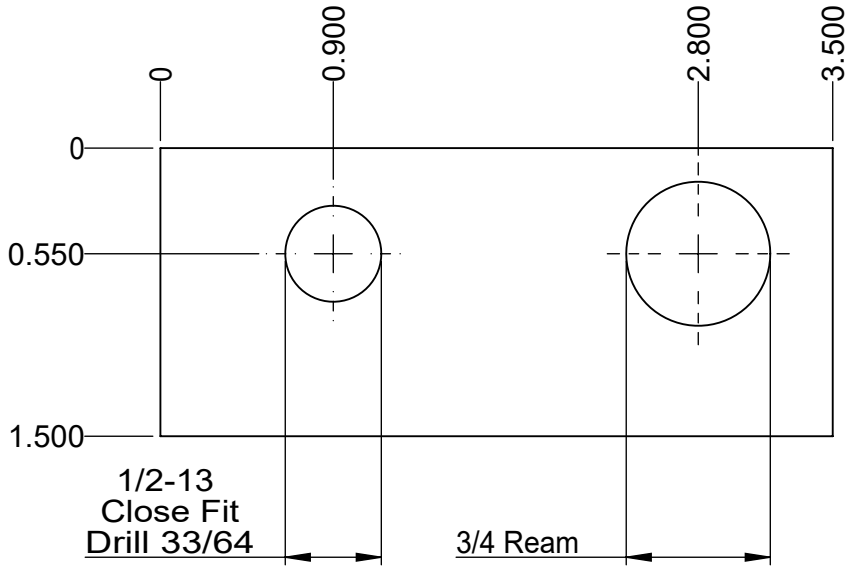


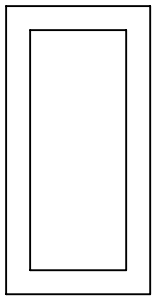
76

TITLE: Al Sheet 3-1 Al Sheet 3-2		
SIZE A	DWG. NO. 1	REV
SCALE: 1:2	WEIGHT:	SHEET 2 OF 11

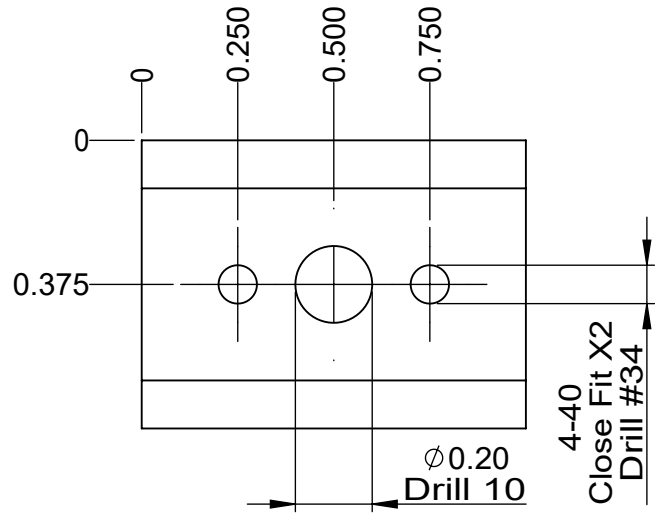


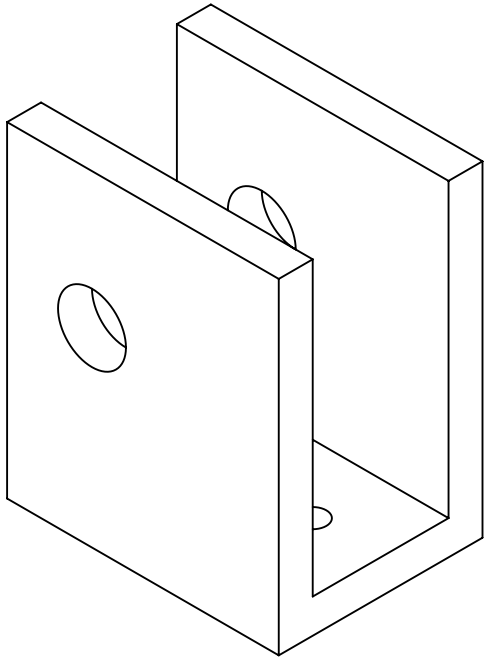


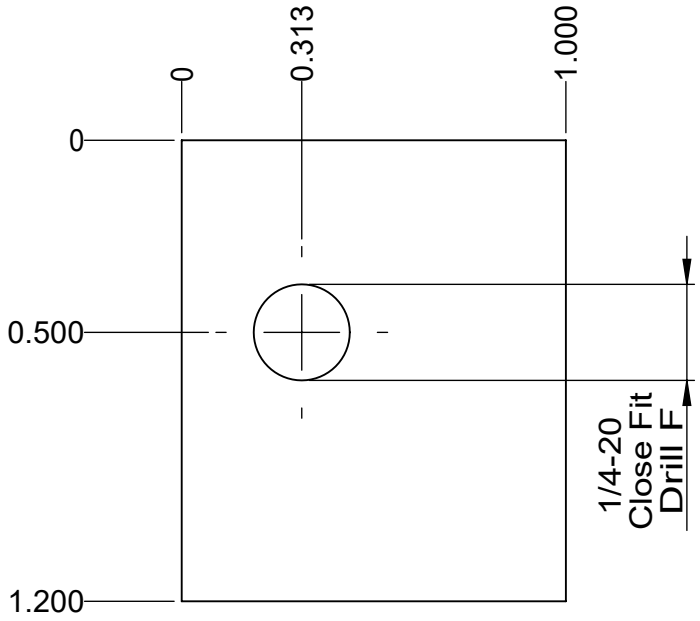


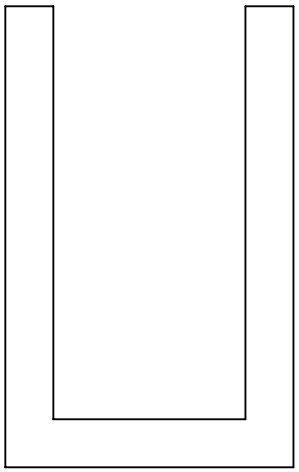


TITLE: Rec Tube 1		
SIZE A	DWG. NO. 1	REV
SCALE: 1:2	WEIGHT:	SHEET 3 OF 11

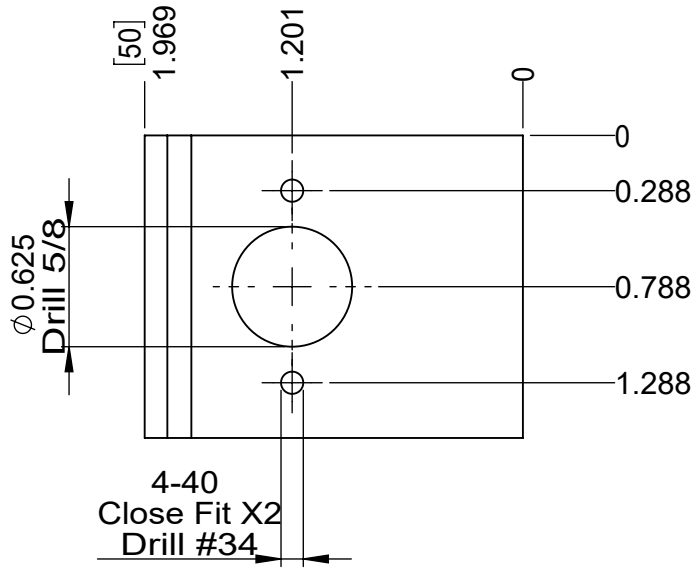


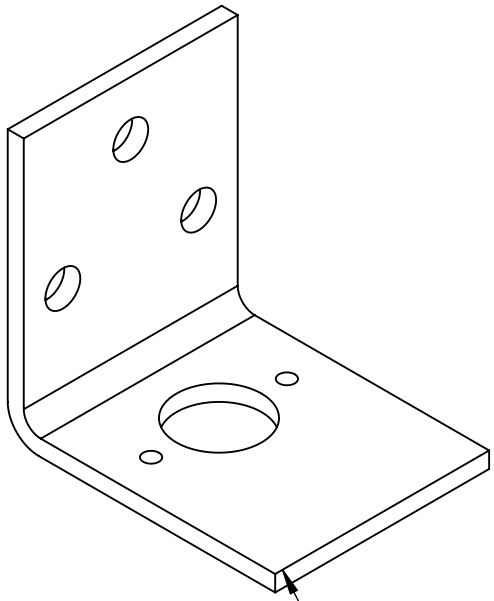


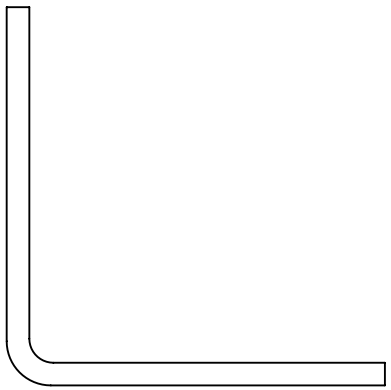


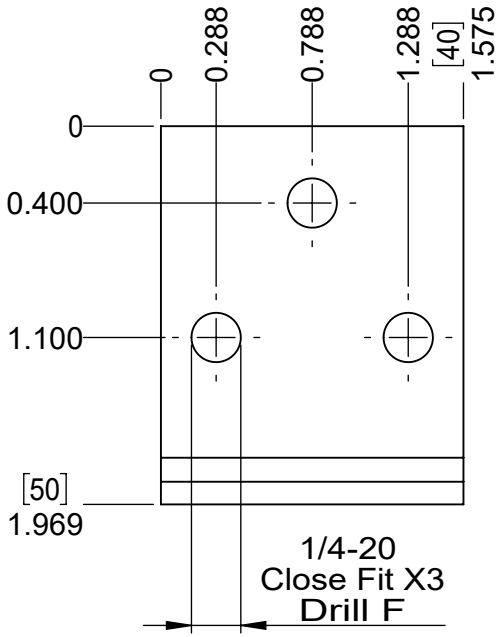


TITLE: Rec Tube 2		
SIZE A	DWG. NO. 1	REV
SCALE: 2:1	WEIGHT:	SHEET 4 OF 11









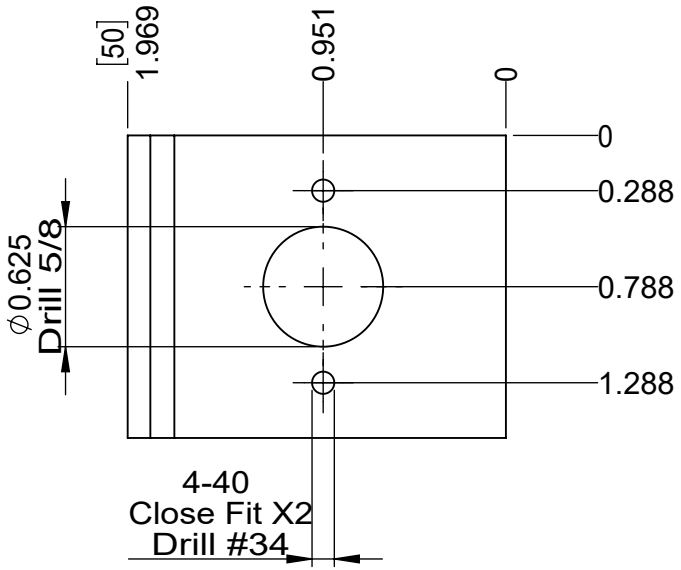
X2

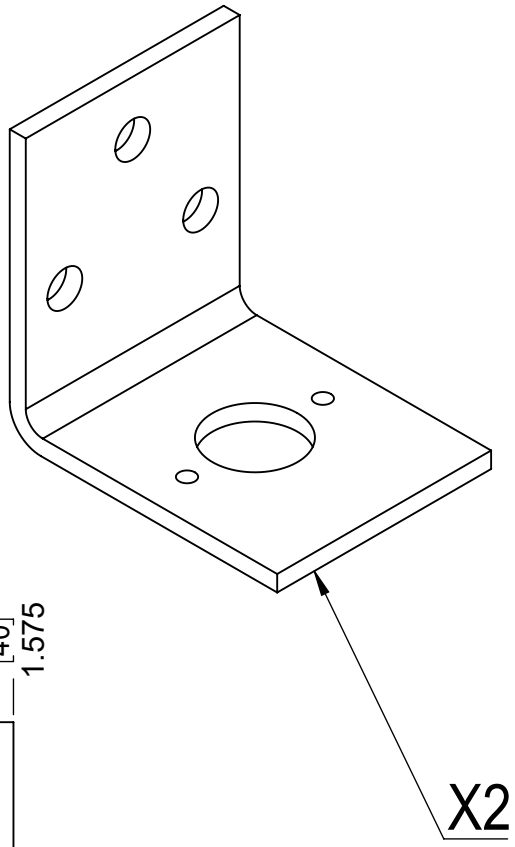
TITLE:

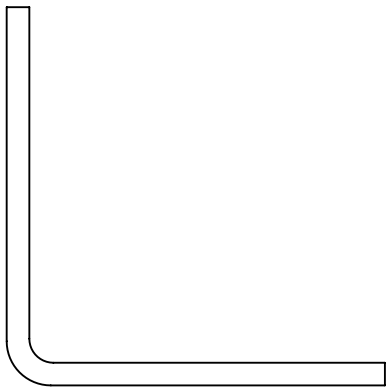
L Bracket 50-50-40 Nut Vertical

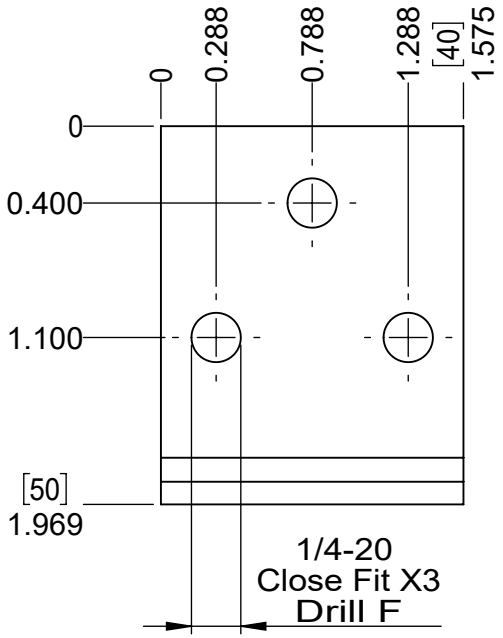
SIZE	DWG. NO.	REV
A	1	

SCALE: 1:1	WEIGHT:	SHEET 5 OF 11

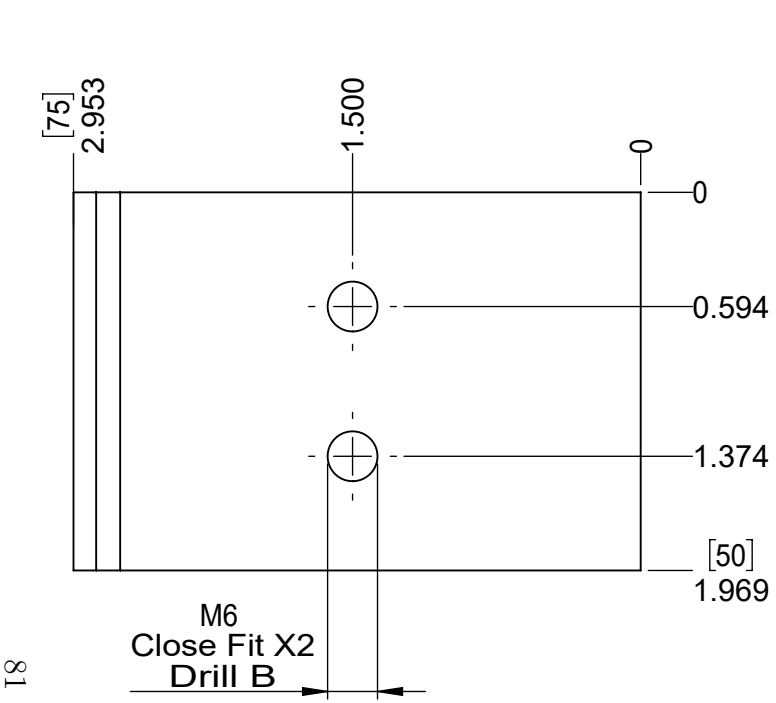


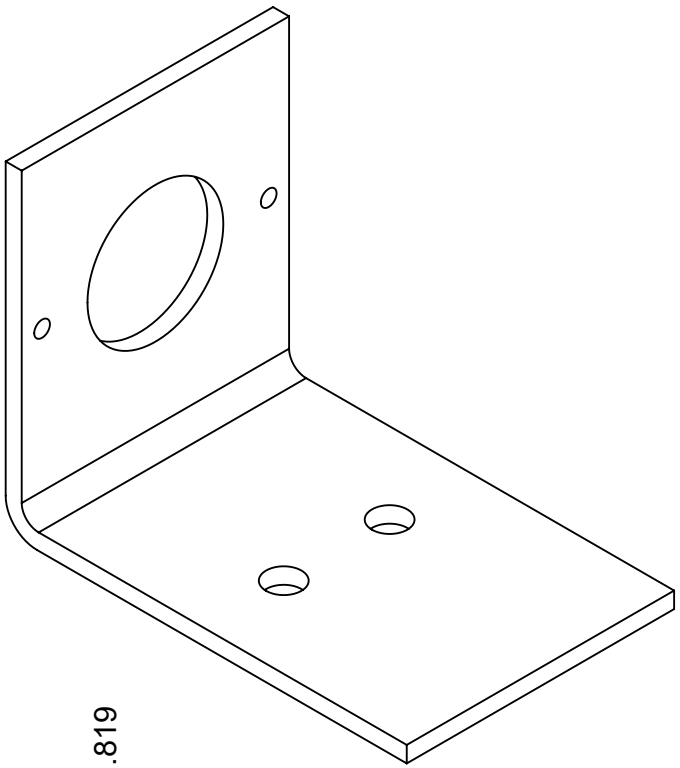


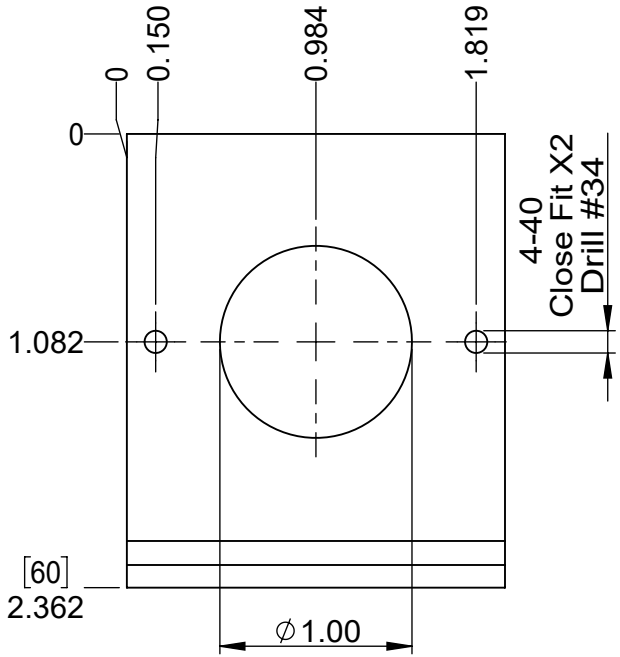




TITLE: L Bracket 50-50-40 Nut Horizontal		
SIZE A	DWG. NO. 1	REV
SCALE: 1:1	WEIGHT:	SHEET 6 OF 11





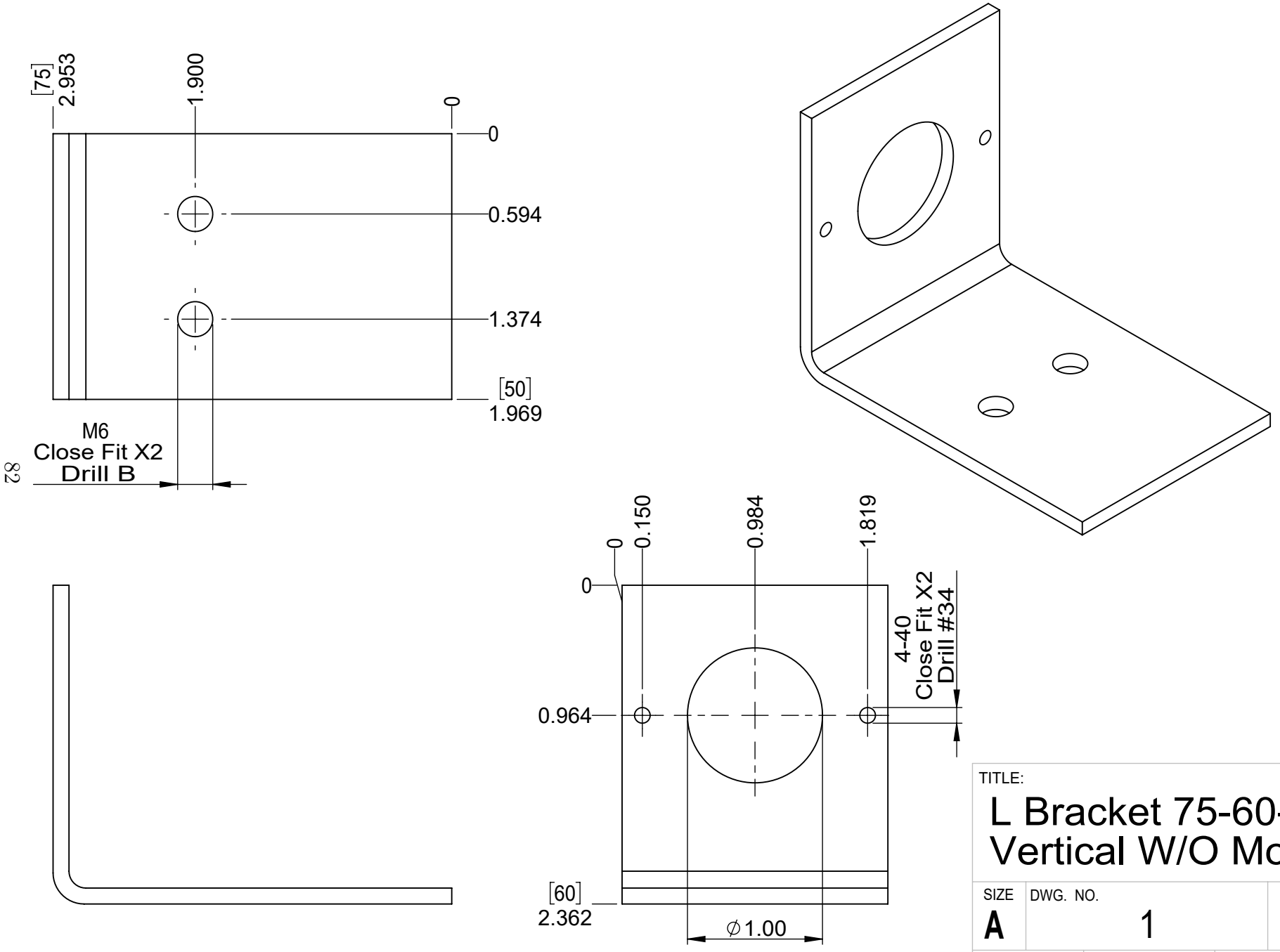


TITLE:

L Bracket 75-60-50 Vertical Motor

SIZE	DWG. NO.	REV
A	1	

SCALE: 1:1	WEIGHT:	SHEET 7 OF 11

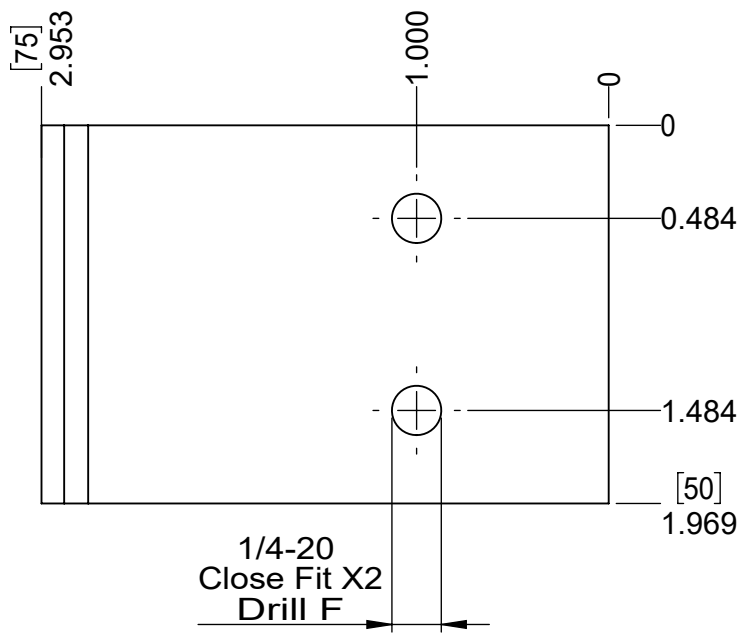


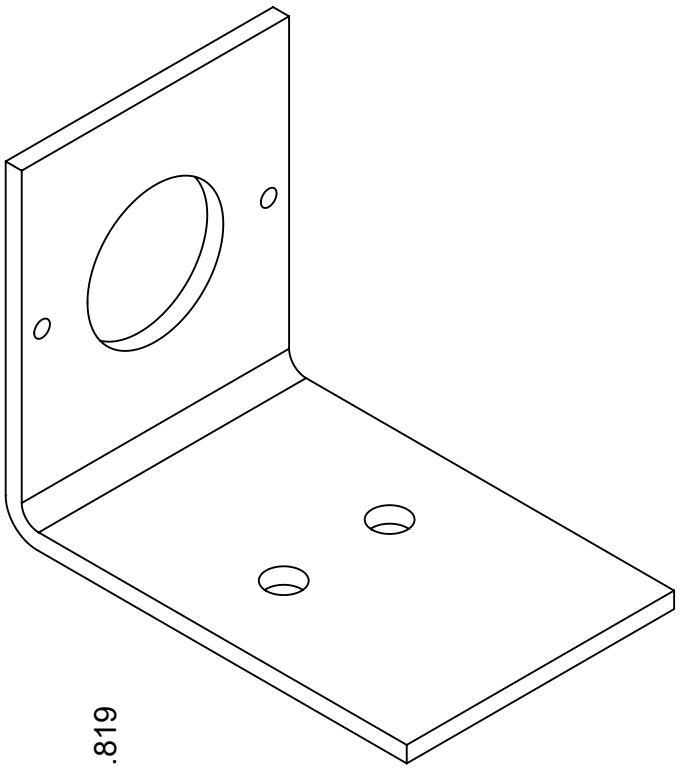
TITLE:

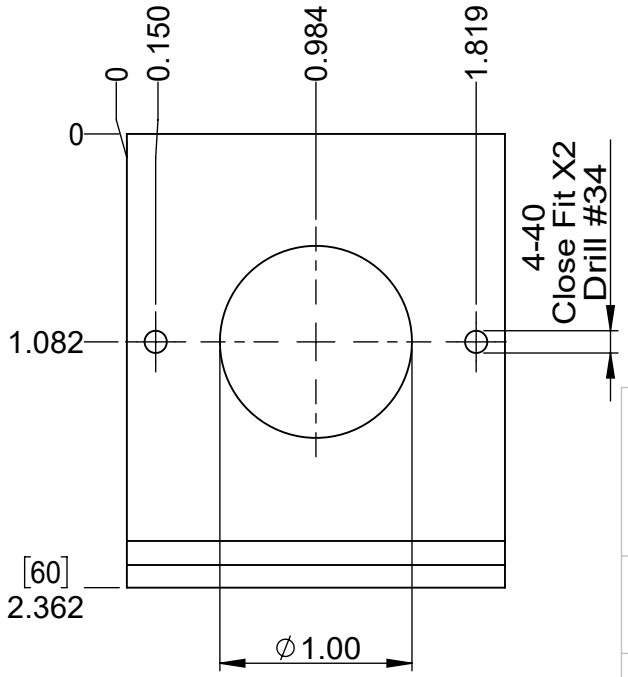
L Bracket 75-60-50 Vertical W/O Motor

SIZE	DWG. NO.	REV
A	1	

SCALE: 1:1 | WEIGHT: | SHEET 8 OF 11





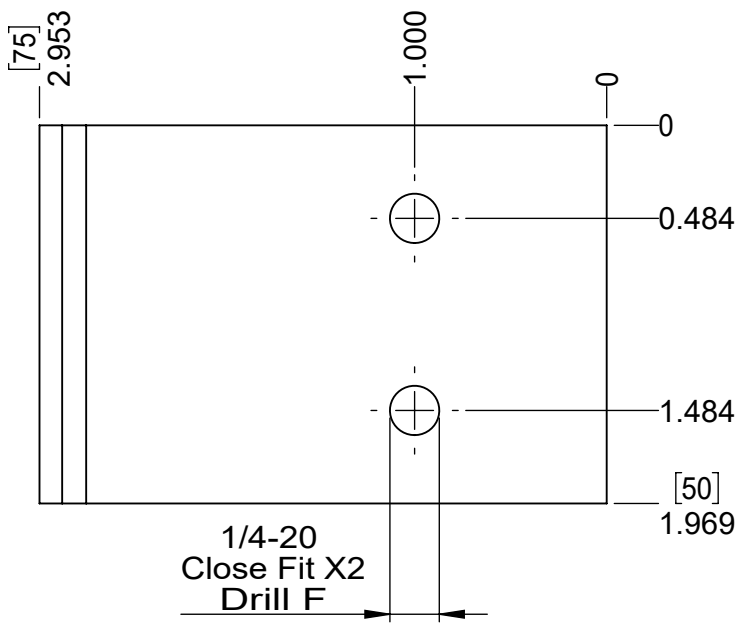


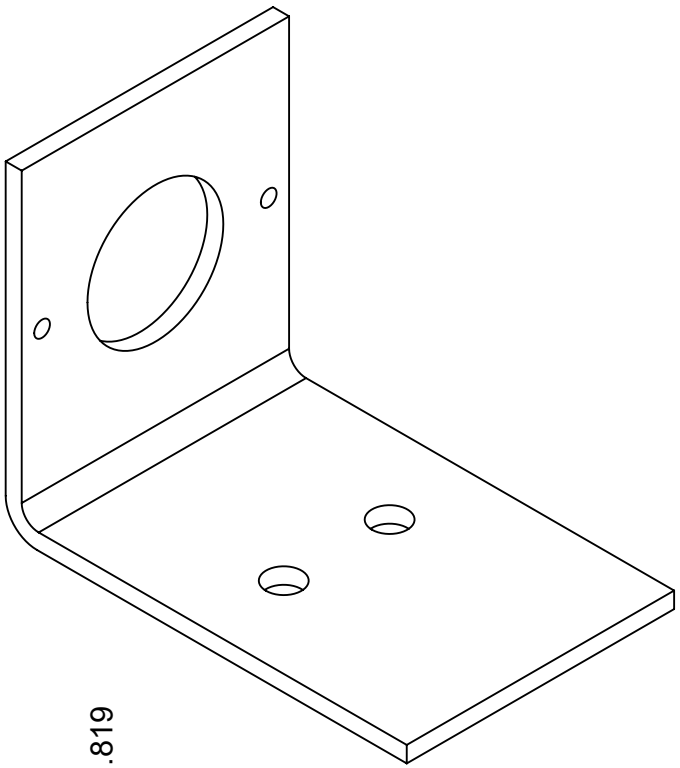
TITLE:

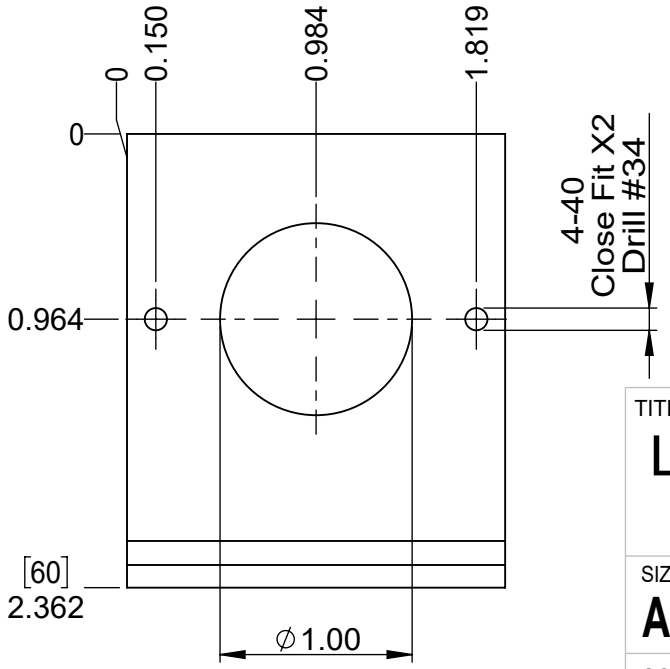
L Bracket 75-60-50 Horizontal Motor

SIZE	DWG. NO.	REV
A	1	

SCALE: 1:1 | WEIGHT: | SHEET 9 OF 11





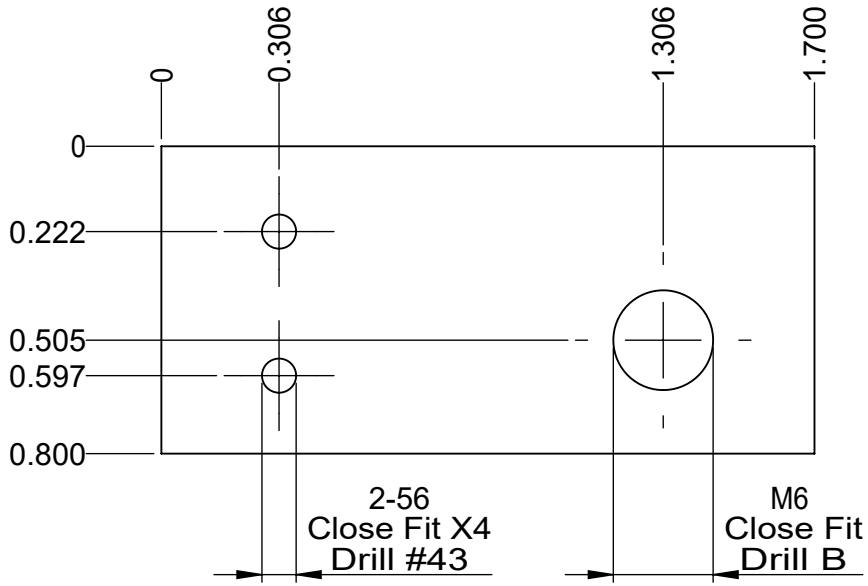


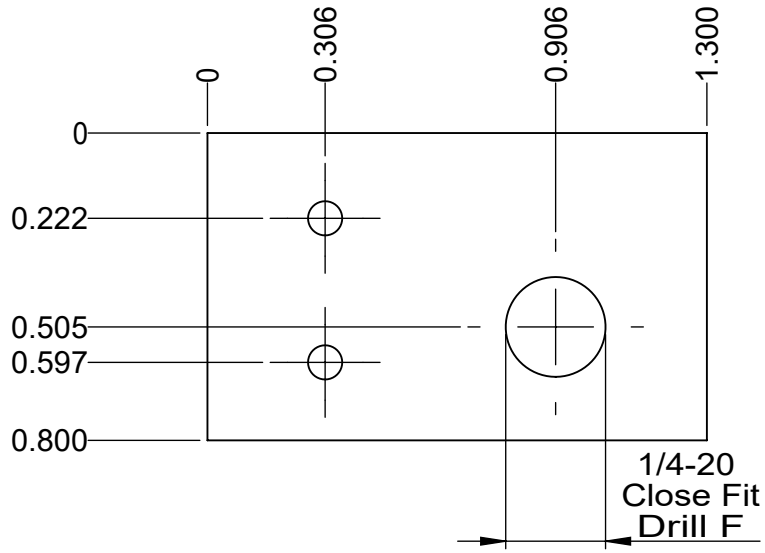
TITLE:

L Bracket 75-60-50 Horizontal W/O Motor

SIZE	DWG. NO.	REV
A	1	

SCALE: 1:1	WEIGHT:	SHEET 10 OF 11





TITLE:

Limit Switch Plate

SIZE	DWG. NO.	REV
A	1	

SCALE: 2:1	WEIGHT:	SHEET 11 OF 11

Appendix D

Part List

The parts used for this project are listed below.

Category	Name	Spec			Screw Size	Link	Unit Price	Amount for Final Desgin	Comment
T-Slot	Rail	1''	4ft			https://www.mcmaster.com/#47065t101/=1cbr8ec	$12.31	4	
			3ft				$10.57	10	X2 cut to 34'', X2 use for cutting shorter ones
			2ft				$7.79	5	
			1ft				$5.84	2	
	Corner Brace					https://www.mcmaster.com/#47065t216/=1cbr8mf	$4.96	38	
	Corner Surface Bracket					https://www.mcmaster.com/#47065t267/=1ct8pah	$7.62	8	
	Tee Surface Bracket					https://www.mcmaster.com/#47065t278/=1ct8qy1	$7.95	10	
	3 Way Outside Corner					https://www.mcmaster.com/#47065t244/=1cbra50	$9.86	8	
	Fasten Plate	Single			1/4''-20-1/2''	https://www.mcmaster.com/#47065t142/=1cbr7ot	$2.30	10	
	Fasten Plate	Dual	1'' screw distance		1/4''-20-1/2''	https://www.mcmaster.com/#47065t147/=1cbr7xq	$4.29	10	
	Dust Cover					https://www.mcmaster.com/#47065t93/=1cbr627	$4.15	3	
Aluminum	L Bracket	75 * 60 * 50 mm				https://www.mcmaster.com/#2313n46/=1cby7fm	$13.60	8	
	L Bracket	50 * 50 * 40 mm				https://www.mcmaster.com/#2313n43/=1cttttv	$8.39	4	
	Al Sheet	2 * 24 * 1/8''				https://www.mcmaster.com/#89015k231/=1ctty5m	$10.46	2	
	Al Sheet	8 * 8 * 1/8''				https://www.mcmaster.com/#89015k239/=1ctu0cb	$13.07	3	
	Al Rod	1 1/4'' OD 1'' length				https://www.mcmaster.com/#1610t12/=1d2s65g	$2.96	1	
Electronics	Manual Switch	Toggle - Monentary		Exit Switch	1/2''	https://www.mcmaster.com/#7343k29/=1ct8u34	$9.14	1	
		Toggle - Maintained		Pause and Wire switch	1/2''	https://www.mcmaster.com/#7343k184/=1ct8uci	$4.34	2	
	Limit Switch				2-56	https://www.mcmaster.com/#7193k3/=1cazszw	$7.17	4	
	Standoff				2-56	https://www.mcmaster.com/#90268a203/=1ccj61x	$1.66	10	
	Flat cable Signal	26 awg	4 cable			https://www.mcmaster.com/#9634t203/=1ctvn73	$2.08	10	
	Flat cable Power	18 awg	4 cable			https://www.mcmaster.com/#9634t603/=1ct11kf	$3.78	10	
Panel	Ruler	L to R				https://www.mcmaster.com/#1910a42/=1ct0b2b	$2.31	5	
		R to L				https://www.mcmaster.com/#1910a22/=1ct0aul	$2.31	5	
	Perforated Sheet	PVC 1/4'' thick	Hole D 0.25''	Hole Spacing 0.5''		https://www.mcmaster.com/#92985t52/=1ct8ij0	$92.78	1	
Wire Retractor	Constant-Force Spring	10.60 lbs	Width 1''	ID 1.2'', OD 1.52''		https://www.mcmaster.com/#9293k12/=1ctcaxk	$10.83	1	
	Sleeve Bearing		Length 1/2''	ID 1/4'', OD 3/8''		https://www.mcmaster.com/#6391k136/=1cucsi1	$0.67	1	
	Sleeve Bearing		Length 1 1/2''	ID 1/2'', OD 3/4''		https://www.mcmaster.com/#6391k413/=1ctcfzf	$4.71	1	
	Sleeve Bearing		Length 3/8''	ID 1/2'', OD 3/4''		https://www.mcmaster.com/#6391k521/=1ctsrf0	$1.50	0	No longer required in final design
	Flanged Sleeve Bearing		Length 1''	ID 1/2'', OD 3/4''		https://www.mcmaster.com/#6338k424/=1ctsg38	$2.28	2	
	One-way Bearing		Length 1/2''	ID 1/2'', OD 3/4''		https://www.mcmaster.com/#2489k24/=1ctsnot	$11.17	1	
	Shaft		Length 9''	D 1/2''		https://www.mcmaster.com/#7398k6/=1cuchny	$19.40	1	
	Shaft Collar					https://www.mcmaster.com/#9946k15/=1ctt7z4	$2.77	4	
	Rectangular Tube		Length 12''	1.5*0.75''		https://www.mcmaster.com/#6546k54/=1cts37f	$9.12	1	

Category	Name	Spec			Screw Size	Link	Unit Price	Amount for Final Desgin	Comment
Screw	Button Head Screw	2-56 1/4''				https://www.mcmaster.com/#92196a077/=1cu00p8	$5.97	1	
		4-40 0.5''				https://www.mcmaster.com/#92949a110/=1ctzdks	$3.38	1	
		4-40 1 3/8''				https://www.mcmaster.com/#92949a813/=1ctzdgv	$5.19	1	
		1/4-20 0.5''				https://www.mcmaster.com/#97763a263/=1ctznzk	$8.35	1	
		1/4-20 1''				https://www.mcmaster.com/#97763a267/=1ctze0m	$6.72	1	
	Socket Head Screw	1/2-13 3.5''				https://www.mcmaster.com/#90044a167/=1ctzj8m	$5.05	1	
		M6 12mm				https://www.mcmaster.com/#91292a134/=1ctzrd6	$7.44	1	
		M6 20mm				https://www.mcmaster.com/#91292a137/=1ctzorm	$4.44	1	
		M6 30mm				https://www.mcmaster.com/#91292a139/=1ctzfq8	$9.41	0	No longer required in final design
	Thumb Screw	1/4-20 1/2''				https://www.mcmaster.com/#94567a510/=1d2s6uf	$2.65	24	
Nut	4-40 Nut					https://www.mcmaster.com/#91841a005/=1ctzaex	$2.91	1	
	1/4-20 Nut					https://www.mcmaster.com/#97149a100/=1ctzgtl	$7.24	1	
	1/2-13 Nut					https://www.mcmaster.com/#97149a250/=1ctzjtn	$6.51	1	
	M6 Nut					https://www.mcmaster.com/#91828a251/=1cu0gf7	$8.73	1	
Shaft	Shaft	20mm	1000mm			https://www.mcmaster.com/#6459k62/=1cagjsi	$95.82	4	X2 cut from previous foamcutter
		20mm	600mm			https://www.mcmaster.com/#6459k59/=1cagjxv	$57.48	4	X4 cut from previous foamcutter
	Shaft Support			Mounting Hole 45mm	1* M5 2*M7	https://www.mcmaster.com/#61815k36/=1cag4oi	$25.19	16	X8 from previous foamcutter
Transmission	Linear Bearings			40 mm spacing	4* M6	https://www.mcmaster.com/#9338t54/=1cag46d	$68.20	12	X8 from previous foamcutter
	Shaft Coupler	10mm to 1/2 inch				https://www.mcmaster.com/#2464k34/=1cbgsmi	$68.64	4	X4 from previous foamcutter
	Lead screw	1/2''-10	3 ft	304 Stainless		https://www.mcmaster.com/#98980a132/=1cbizut	$36.59	4	
	Nut		OD 1.47''	thinckness 1''		https://www.mcmaster.com/#1343k134/=1cbj0f6	$36.81	4	
	Collar		OD 1.3125''	thinckness 0.5''		https://www.mcmaster.com/#6698k13/=1cbj08y	$18.35	8	X4 from previous foamcutter
	Bearing		OD 1.125''	thinckness 0.3125''		https://www.mcmaster.com/#60355k704/=1cbrell	$9.08	0	No longer required in final design
	Shim					https://www.mcmaster.com/#91182a703/=1cbrlk2	$11.59	1	
	Acrylic	24'' * 48'' *1/4''				https://www.mcmaster.com/#8589k84/=1cy43x1	$55.76	1	
	Acrylic	6'' * 12'' * 1/8''				https://www.mcmaster.com/#8560k275/=1cyln3f	$5.37	1	No longer required in final design
	Delrin	6'' * 6'' * 1/4''				https://www.mcmaster.com/#8573k121/=1d2s6lb	$9.43	1	

Appendix E

Purchase History

The purchase history of this project are listed below.

Category	Name	Spec		Unit Price	Amount for Final Desgin	Order1 Amount	$1,235.68	Order2 Amount	$1,256.92	Order3 Amount	$148.63	Order4 Amount	$53.70
T-Slot	Rail	1''	4ft	$12.31	4	4	$49.24	----	----	----	----	----	----
			3ft	$10.57	10	9	$95.13	1	$10.57	----	----	----	----
			2ft	$7.79	5	5	$38.95	1	$7.79	----	----	----	----
			1ft	$5.84	2	----	----	----	----	2	$11.68	----	----
	Corner Brace			$4.96	38	32	$158.72	6	$29.76	----	----	----	----
	Corner Surface Bracket			$7.62	8	8	$60.96	4	$30.48	8	$60.96	----	----
	Tee Surface Bracket			$7.95	10	10	$79.50	----	----	----	----	2	$15.90
	3 Way Outside Corner			$9.86	8	8	$78.88	----	----	----	----	----	----
	Fasten Plate	Single		$2.30	10	10	$23.00	----	----	----	----	----	----
	Fasten Plate	Dual	1'' screw distance	$4.29	10	10	$42.90	----	----	----	----	----	----
	Dust Cover			$4.15	3	3	$12.45	----	----	----	----	----	----
Aluminum	L Bracket	75 * 60 * 50 mm		$13.60	8	8	$108.80	----	----	----	----	----	----
	L Bracket	50 * 50 * 40 mm		$8.39	4	4	$33.56	----	----	----	----	----	----
	Al Sheet	2 * 24 * 1/8''		$10.46	2	2	$20.92	----	----	----	----	----	----
	Al Sheet	8 * 8 * 1/8''		$13.07	3	3	$39.21	----	----	----	----	----	----
	Al Rod	1 1/4'' OD 1'' length		$2.96	1	----	----	----	----	1	$2.96	----	----
Electronics	Manual Switch	Toggle - Monentary		$9.14	1	1	$9.14	----	----	----	----	----	----
		Toggle - Maintained		$4.34	2	2	$8.68	----	----	----	----	----	----
	Limit Switch			$7.17	4	4	$28.68	----	----	----	----	----	----
	Standoff			$1.66	10	10	$16.60	----	----	----	----	----	----
	Flat cable Signal	26 awg	4 cable	$2.08	10	10	$20.80	----	----	----	----	----	----
	Flat cable Power	18 awg	4 cable	$3.78	10	10	$37.80	----	----	----	----	10	$37.80
Panel	Ruler	L to R		$2.31	5	5	$11.55	----	----	----	----	----	----
		R to L		$2.31	5	5	$11.55	----	----	----	----	----	----
	Perforated Sheet	PVC 1/4'' thick	Hole D 0.25''	$92.78	1	1	$92.78	----	----	----	----	----	----
Wire Retractor	Constant-Force Spring	10.60 lbs	Width 1''	$10.83	1	1	$10.83	----	----	----	----	----	----
	Sleeve Bearing		Length 1/2''	$0.67	1	1	$0.67	----	----	----	----	----	----
	Sleeve Bearing		Length 1 1/2''	$4.71	1	1	$4.71	----	----	----	----	----	----
	Sleeve Bearing		Length 3/8''	$1.50	0	2	$3.00	----	----	----	----	----	----
	Flanged Sleeve Bearing		Length 1''	$2.28	2	2	$4.56	----	----	----	----	----	----
	One-way Bearing		Length 1/2''	$11.17	1	1	$11.17	----	----	----	----	----	----
	Shaft		Length 9''	$19.40	1	1	$19.40	----	----	----	----	----	----
	Shaft Collar			$2.77	4	4	$11.08	----	----	----	----	----	----
	Rectangular Tube		Length 12''	$9.12	1	1	$9.12	----	----	----	----	----	----

Category	Name	Spec		Unit Price	Amount for Final Desgin	Order1 Amount	$1,235.68	Order2 Amount	$1,256.92	Order3 Amount	$148.63	Order4 Amount	$53.70
Screw	Button Head Screw	2-56 1/4''		$5.97	1	1	$5.97	----	----	----	----	----	----
		4-40 0.5''		$3.38	1	1	$3.38	----	----	----	----	----	----
		4-40 1 3/8''		$5.19	1	1	$5.19	----	----	----	----	----	----
		1/4-20 0.5''		$8.35	1	1	$8.35	----	----	----	----	----	----
		1/4-20 1''		$6.72	1	1	$6.72	----	----	----	----	----	----
	Socket Head Screw	1/2-13 3.5''		$5.05	1	1	$5.05	----	----	----	----	----	----
		M6 12mm		$7.44	1	1	$7.44	----	----	----	----	----	----
		M6 20mm		$4.44	1	1	$4.44	----	----	----	----	----	----
		M6 30mm		$9.41	0	1	$9.41	----	----	----	----	----	----
	Thumb Screw	1/4-20 1/2''		$2.65	24	----	----	----	----	24	$63.60	----	----
Nut	4-40 Nut			$2.91	1	1	$2.91	----	----	----	----	----	----
	1/4-20 Nut			$7.24	1	1	$7.24	----	----	----	----	----	----
	1/2-13 Nut			$6.51	1	1	$6.51	----	----	----	----	----	----
	M6 Nut			$8.73	1	1	$8.73	----	----	----	----	----	----
Shaft	Shaft	20mm	1000mm	$95.82	4	----	----	2	$191.64	----	----	----	----
		20mm	600mm	$57.48	4	----	----	0	$0.00	----	----	----	----
	Shaft Support			$25.19	16	----	----	8	$201.52	----	----	----	----
Transmission	Linear Bearings			$68.20	12	----	----	4	$272.80	----	----	----	----
	Shaft Coupler	10mm to 1/2 inch		$68.64	4	----	----	0	$0.00	----	----	----	----
	Lead screw	1/2''-10	3 ft	$36.59	4	----	----	4	$146.36	----	----	----	----
	Nut		OD 1.47''	$36.81	4	----	----	4	$147.24	----	----	----	----
	Collar		OD 1.3125''	$18.35	8	----	----	4	$73.40	----	----	----	----
	Bearing		OD 1.125''	$9.08	0	----	----	8	$72.64	----	----	----	----
	Shim			$11.59	1	----	----	1	$11.59	----	----	----	----
	Acrylic	24'' * 48'' *1/4''		$55.76	1	----	----	1	$55.76	----	----	----	----
	Acrylic	6'' * 12'' * 1/8''		$5.37	1	----	----	1	$5.37	----	----	----	----
	Delrin	6'' * 6'' * 1/4''		$9.43	1	----	----	----	----	1	$9.43	----	----