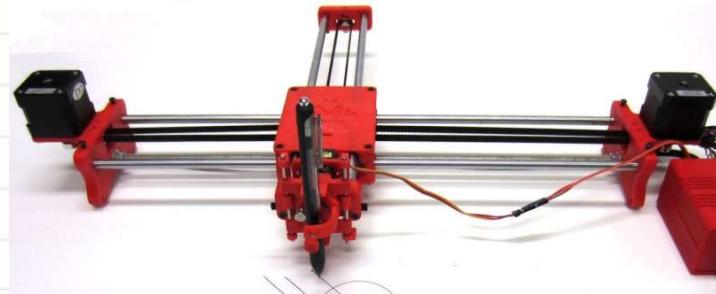


Drawing Robot

Assembly & User Guide

Version 3.0



By: Henry Arnold
Jonathan K

Drawing Robot

- Assembly
- Software
- Example Projects



Drawing Robot **ASSEMBLY**

Getting Started...

- The Drawing Robot takes about two weekends to build. It takes approximately 20 hours of 3D printing time to make the parts. Your construction time may vary depending on your experience with 3D printing, GRBL, Arduino, etc.
- Cost to build the Drawing Robot is between \$75 and \$100 depending on where you buy your parts and whether you already own some of the parts such as the Arduino.
- The resulting Drawing Robot is a very capable machine which is similar in form and function to the commercially available AxiDraw Robot.



Parts List

Components

(2) Nema 17 Stepper Motors <https://www.amazon.com/Stepper-Motor-Bipolar-64oz-Printer/dp/B00PNEQI7W>

(2) Linear Rod M8 x 450mm, X Axis <https://www.amazon.com/dp/B07DPHDMDT>

(2) Linear Rod M8 x 350mm, Y Axis <https://www.amazon.com/dp/B07JKTLFD7>

(Note: If longer Linear Rods were purchased, you may cut them to length)

(2) Linear Rod 3mm, Z Axis (from old CDROM)

(1) Threaded Rod M8 x 480mm (3/16" x 20" threaded rod is fine for non-metric countries like USA)

<https://www.amazon.com/dp/B002CTZTSU>

(8) LM8UU Bearings <https://www.amazon.com/uxcell-Bushing-Linear-Motion-Double/dp/B00X9H22SO>

(1) Servo Sg90 <https://www.amazon.com/Uperten-Micro-Helicopter-Remote-Control/dp/B07KVJ84FS>

(2) Spring (from ball point pen)

(2) GT2 Pulley, 16 teeth <https://www.amazon.com/Anet-Timing-Pulley-Aluminum-Printer/dp/B07D294B2T>

(5) Bearing 624zz <https://www.amazon.com/uxcell-Bearing-4x13x5mm-Shielded-Bearings/dp/B07PLC6GY3>

Parts List

Components

(1) 2000mm GT2 belt <https://www.amazon.com/Mercury-Meters-timing-Rostock-GT2-6mm/dp/B071K8HYB4>

(1) Arduino Uno (kit with CNC and 4988)

<https://www.amazon.com/kuman-Shield-Expansion-Stepper-Driver/dp/B06XHKSVTG>

(1) Arduino CNC Shield

(2) A4988 Stepper Drivers

(6) Jumpers (The kit above is missing these jumpers)

<https://www.amazon.com/ZYAMY-2-54mm-Standard-Circuit-Connection/dp/B077957RN7>

(1) 12V 2A Power Supply

<https://www.amazon.com/Adapter-100-240V-Transformers-Switching-Adaptor/dp/B019Q3U72M>

(2-4) Limit switches (optional)

<https://www.amazon.com/URBESTAC-Momentary-Hinge-Roller-Switches/dp/B00MFRMFS6>

Parts List

Hardware

Nuts

- **(7)** M3-0.5
- **(5)** M4-0.7
- **(4)** 5/16in-18

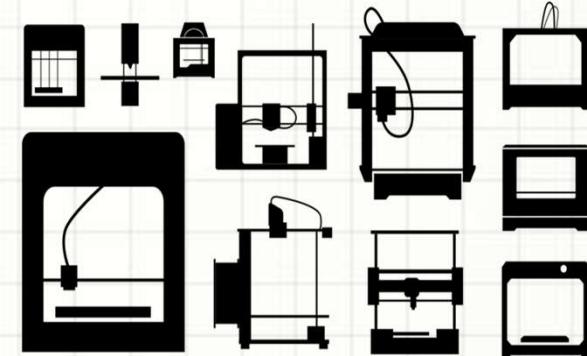
Screws

- **(13)** Phillips M3-0.5 x 16mm
- **(4)** Phillips M3-0.5 x 6mm
- **(5)** Phillips M4-0.7x 35mm
- **(1)** Hex M3-0.5 x 20mm

Washers

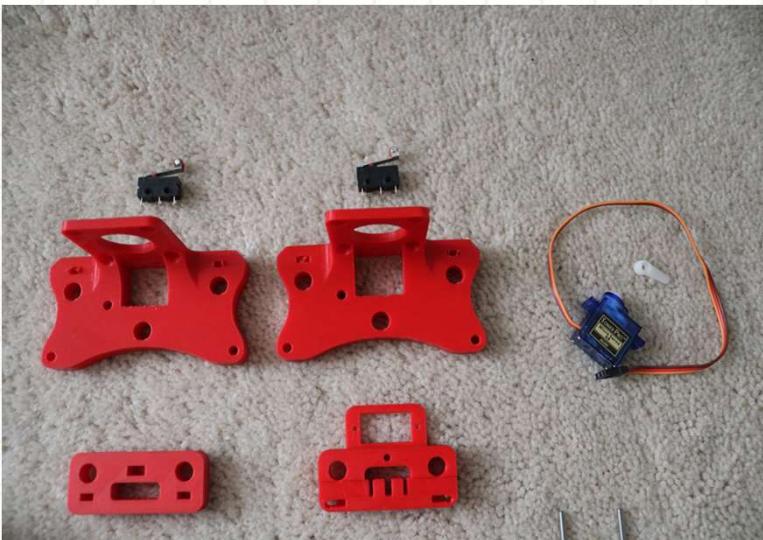
- **(4)** 5/16in washer
- **(4)** M3 washers

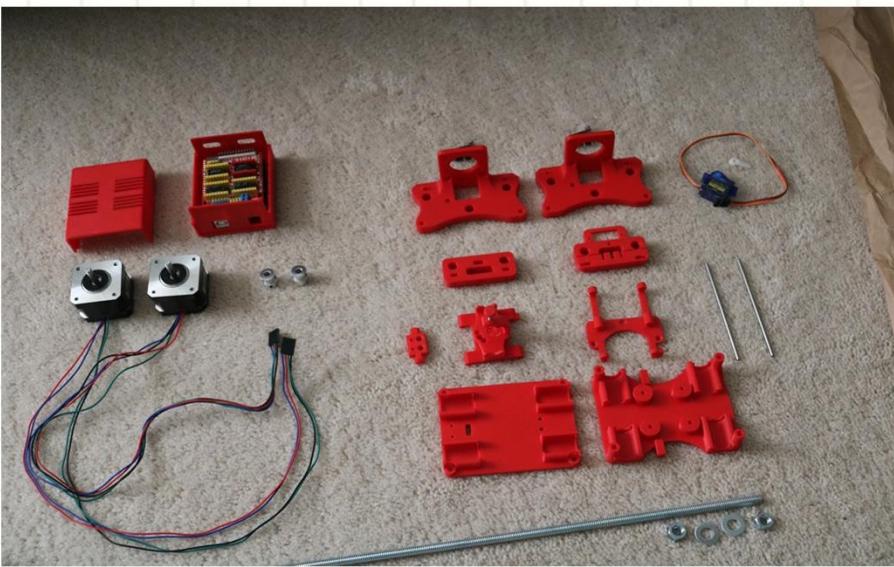
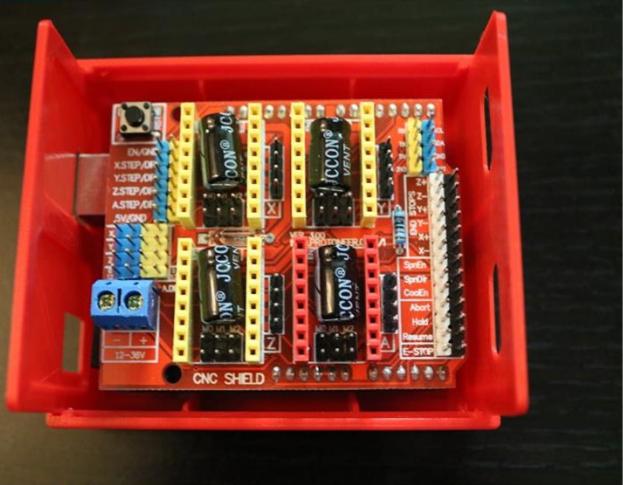
3D Printing



- To start this project download the files from Thingiverse:
<https://www.thingiverse.com/thing:2349232>
- Open the 3D models in your slicer(Cura, Sli3er, Simplify 3D, etc.)
- I recommend using Cura as I have heard that people have had issues with some of the models using Sli3er
- Input the settings that you will be using on the parts
 - Use 75% infill on all the parts (An infill of 70 - 100% will work as well)
 - Printed all the parts with 0.10 – 0.20 mm layer height
 - Printed with Hatchbox Red PLA
 - Use supports on the Penholder, Slider, X_Support_L and the X_Support_R
(Remove the supports from these parts once they are printed)
- The longest part took around 9hrs and the shortest took 30 minutes to print

3D Printed Parts





Cut your Linear Rods to Length

- If you purchased rods that were cut to length, then skip this step
- Use a measuring tape and sharpie to mark the spots where the rods need to be cut
 - Use a vise to hold the rods in place when you cut them
 - Remember that you need **(2)** 350mm and **(2)** 450mm long linear rods
 - On the threaded rod, mark your cutting point at 470mm
- I used an angle grinder to cut the rods (Be sure to wear safety glasses)
- Use a bench grinder to smooth the ends that were cut(chamfer)





Assemble the X-Axis (Linear/Threaded Rods)

- Take the **(2)** 450mm linear rods and insert them into either x-support part
 - Use may need to use a round file to smooth out the holes that you insert them in
 - Also, you can use a rubber mallet to help insert the rods
- Now take the threaded rod and insert it in the hole below. Feed a 5/16in washer and 5/16in nut on both sides of the x-support part



Assemble the X-Axis (Bearings)

- Now you want to push the LM8UU bearings into their place on the top and bottom clamshell
 - The top and bottom clamshell take **(4)** bearings each
- Take **(4)** 624zz bearings and push them through the 3D printed idler pulleys. Leave the 5th bearing for later when you assemble the Y-axis



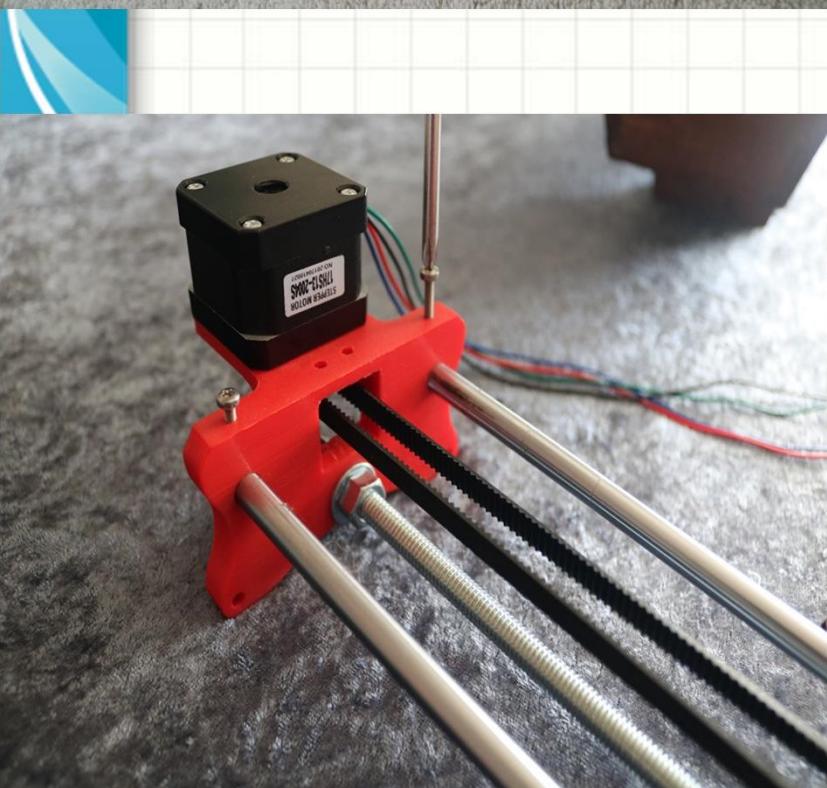
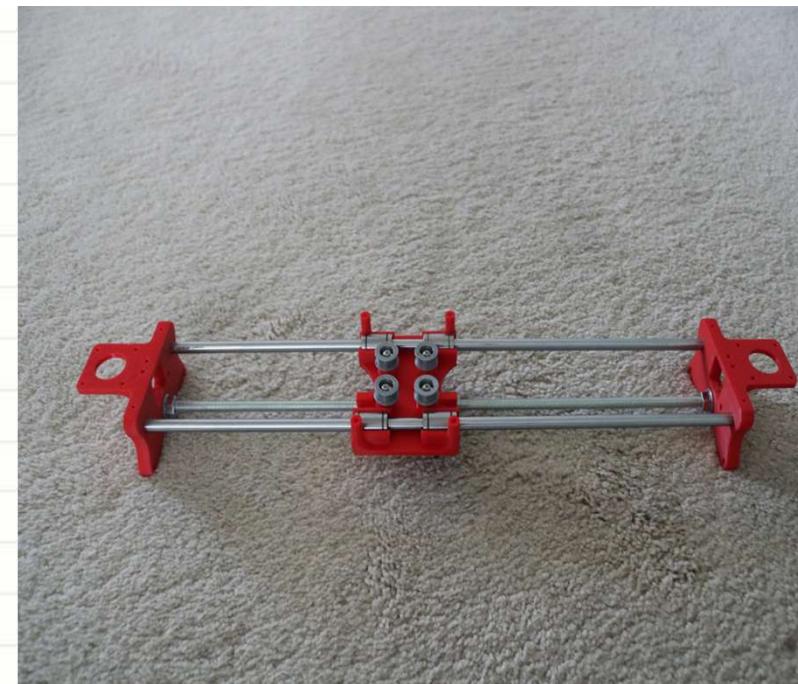
Assemble the X-Axis (Carriage)

- Get **(4)** M3-0.5 x 20mm screws, **(4)** M3 nuts, **(4)** M3 washers and **(4)** 624zz bearings with the idler pulleys installed
- Take one screw and feed a washer through it, the washer will rest on the bearing. The nut will be at the bottom of the carriage, which will secure the bearing in place



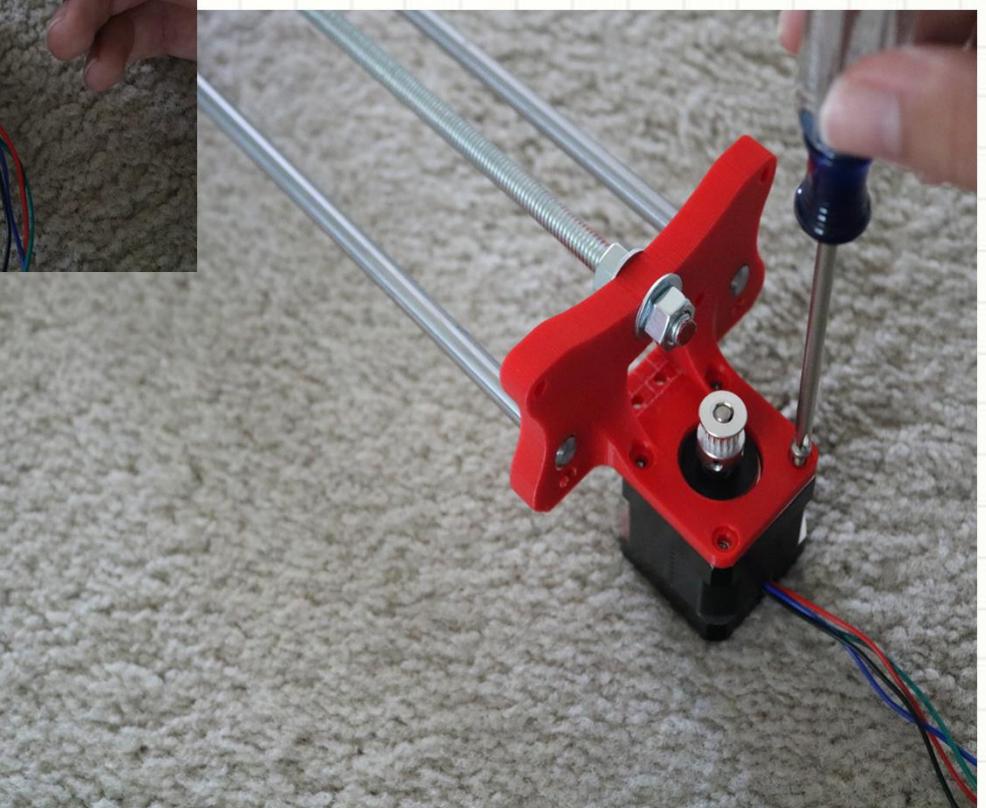
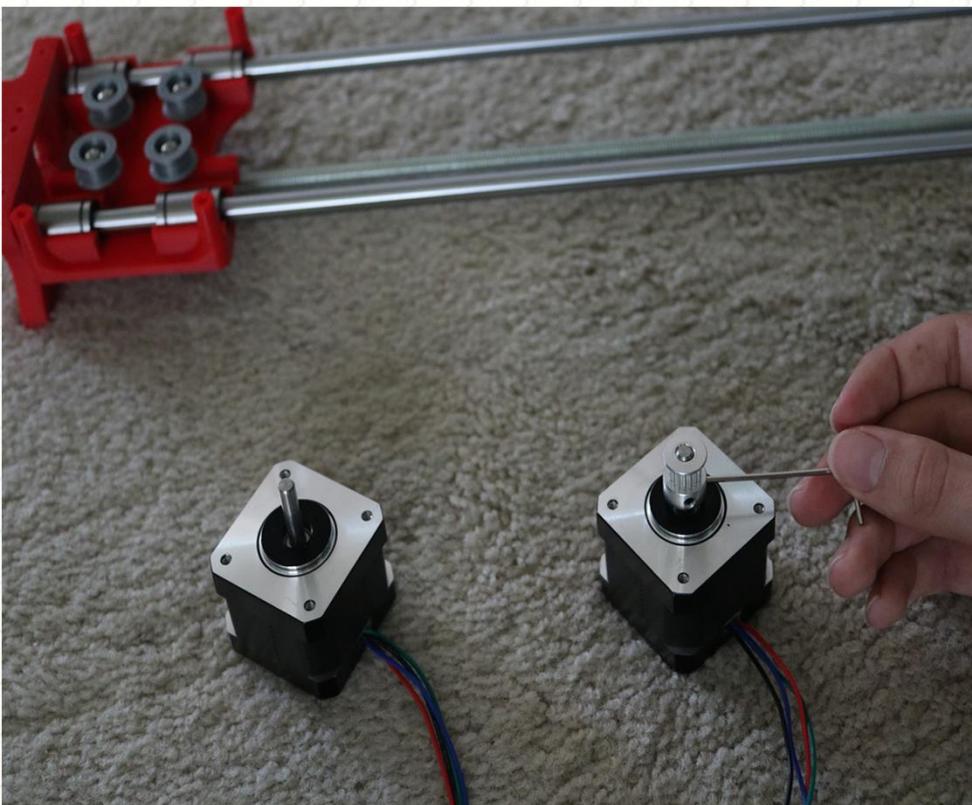
Assemble the X-Axis (X-Support)

- Slide the clamshell through the 450mm(X-axis) linear rods
- Use a rubber mallet again to attach the last X-support on the linear rods
 - Make sure that the rods stick out equally on both sides
 - Slide the other end of the threaded rod through the hole on the X-support
- Put on the last set of nuts and washers to hold the X-support in place
- Now that the X-axis is complete, you can use **(2)** Phillips M3-0.5 x 16mm screws per X-support to help keep the linear rods from sliding



Assemble the X-Axis (Stepper Motors)

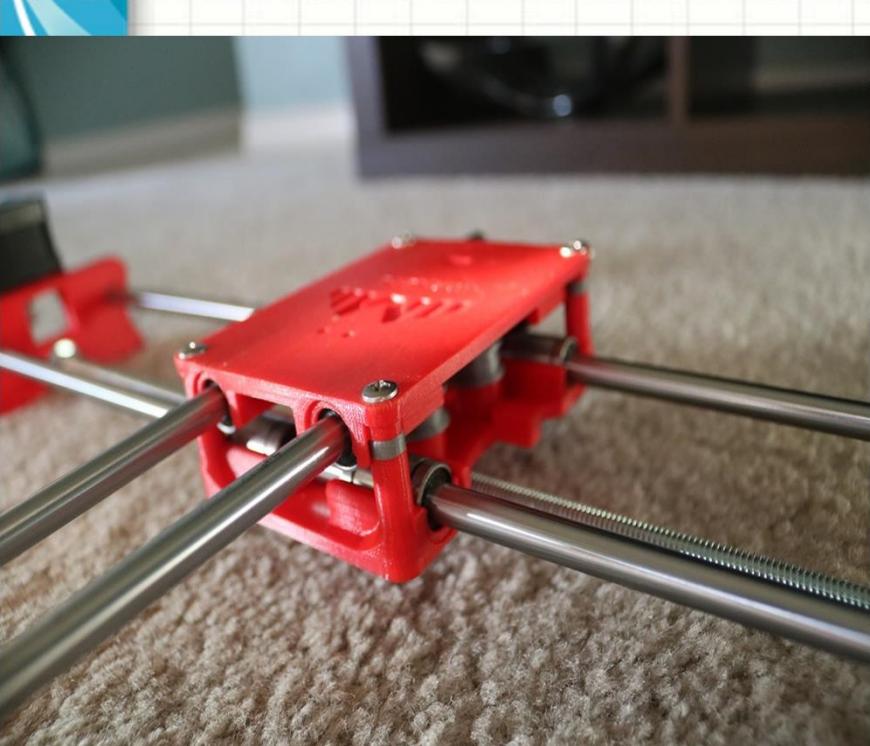
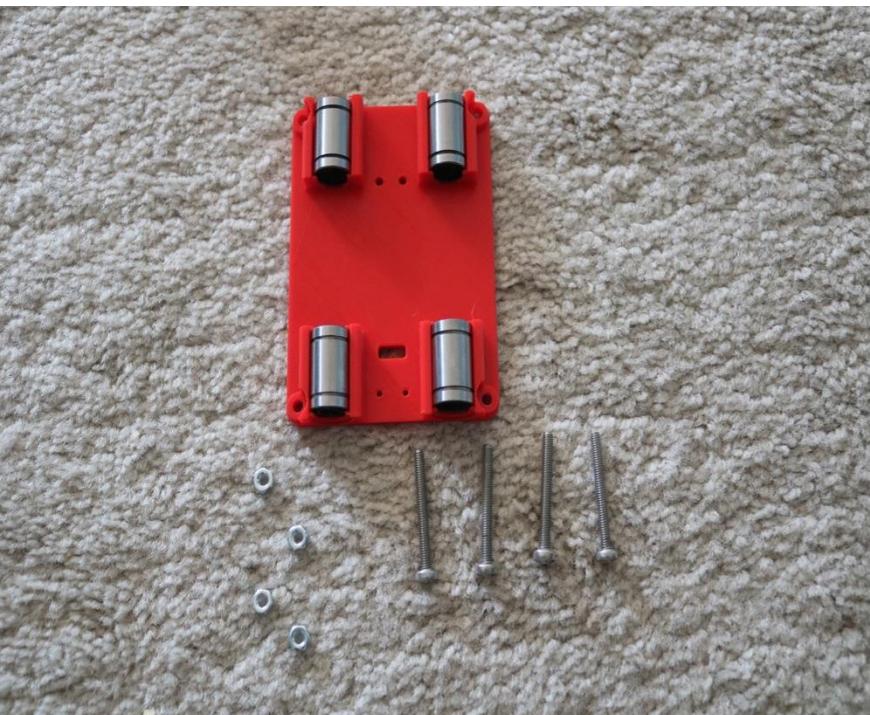
- Use an appropriate sized allen wrench to attach the 16 teeth pulleys on the stepper motor shafts
- Flipping the entire chassis around will make it easier to attach the stepper motors
- Use **(8)** M3-0.5 x 6mm screws and a Phillips screwdriver to attach the **(2)** stepper motors



Assemble the Y-Axis (Clamshell)

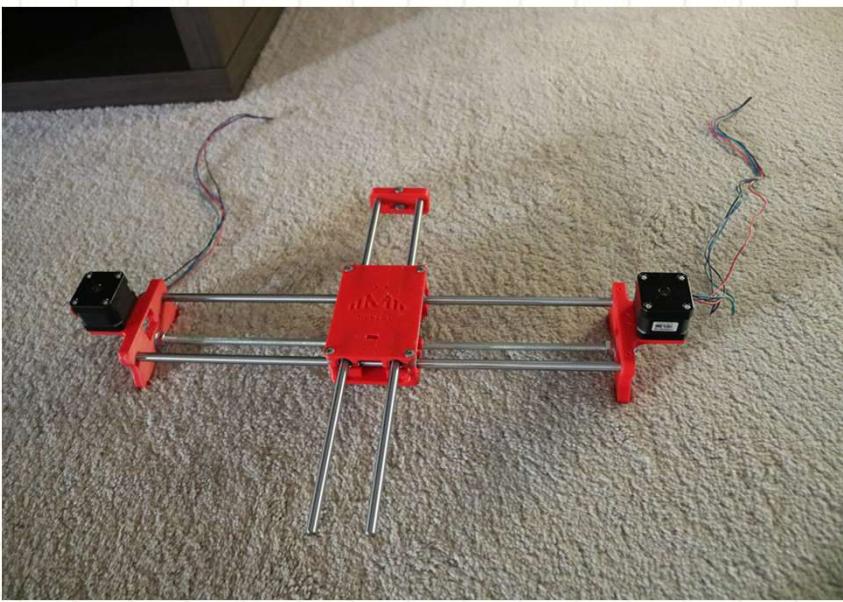
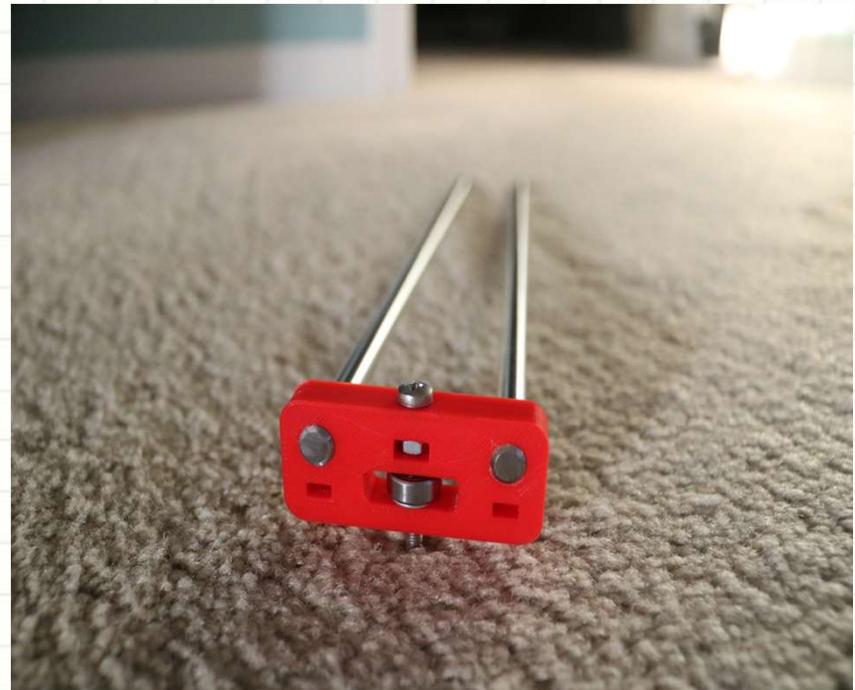
Optional if you have problems keeping belt on bearings

- Grab **(4)** M4-0.5 x 35mm screws and **(4)** M4 nuts
- Make sure that you have the **(4)** idler pulleys and the **(4)** washers printed:
<https://www.thingiverse.com/thing:2424284>
- Insert the washers in between the two clamshells, with a screw in between
- Screw the top and bottom clamshells together



Assemble the Y-Axis (Y-Back/Front)

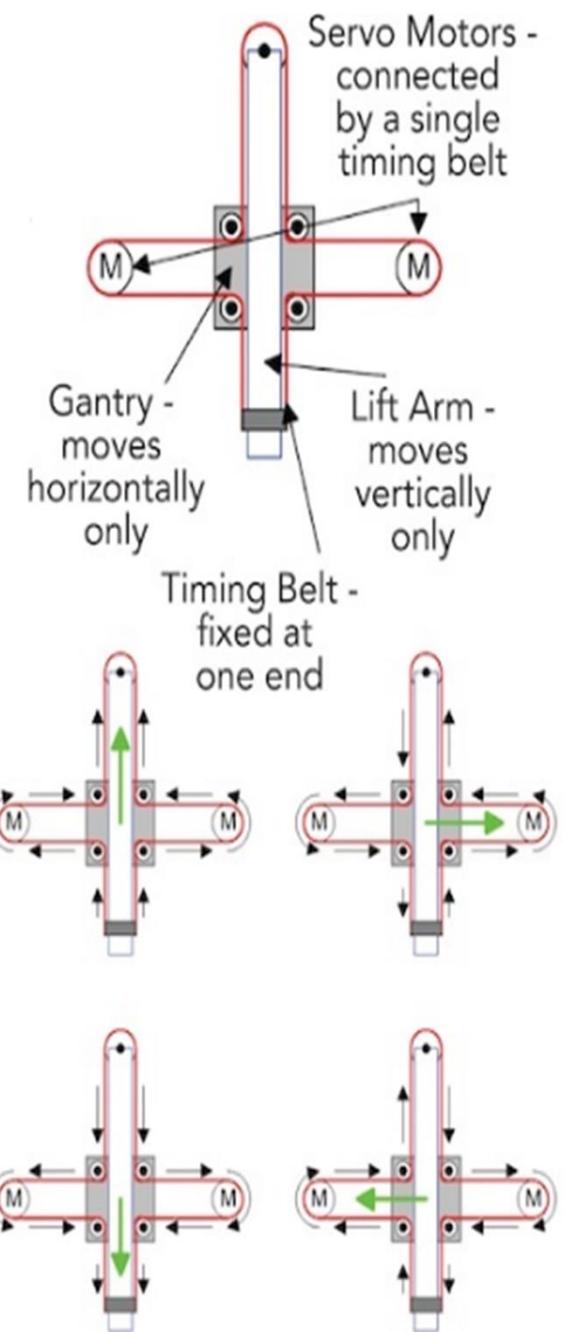
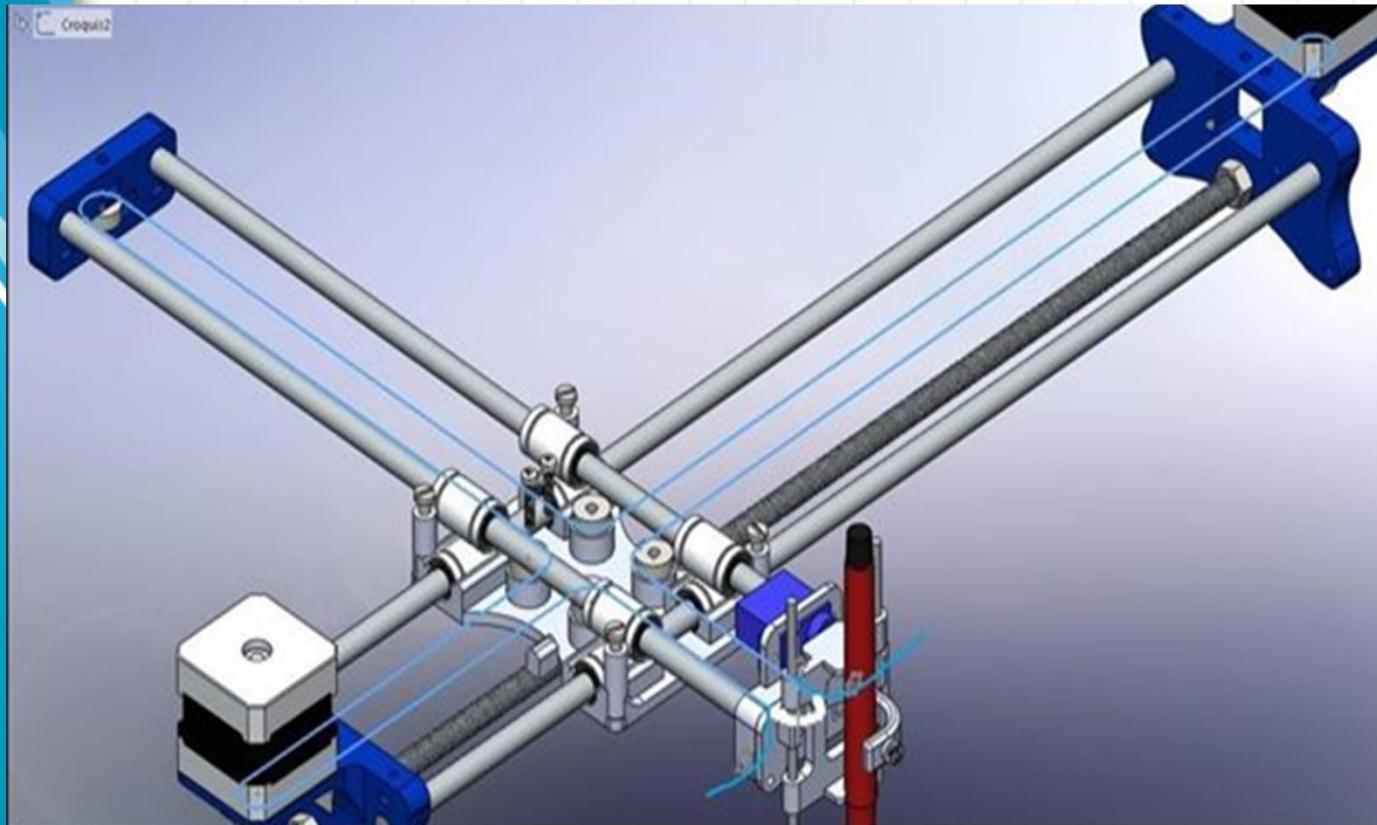
- Y-Back:
 - Take the **(2)** 350mm linear rods and insert them the Y-back piece by using a rubber mallet
 - Get **(1)** M4-0.5 x 35 screw, **(1)** M4 nut and the 5th 624zz bearing
 - Get **(2)** M3-0.5 x 16 screws to secure the linear rods
 - Slide in the bearing when inserting the screw through the Y-back piece
- Y-Front:
 - Slide the the linear rods/Y-back piece through the LM8UU bearings and attach the Y-front piece using a rubber mallet

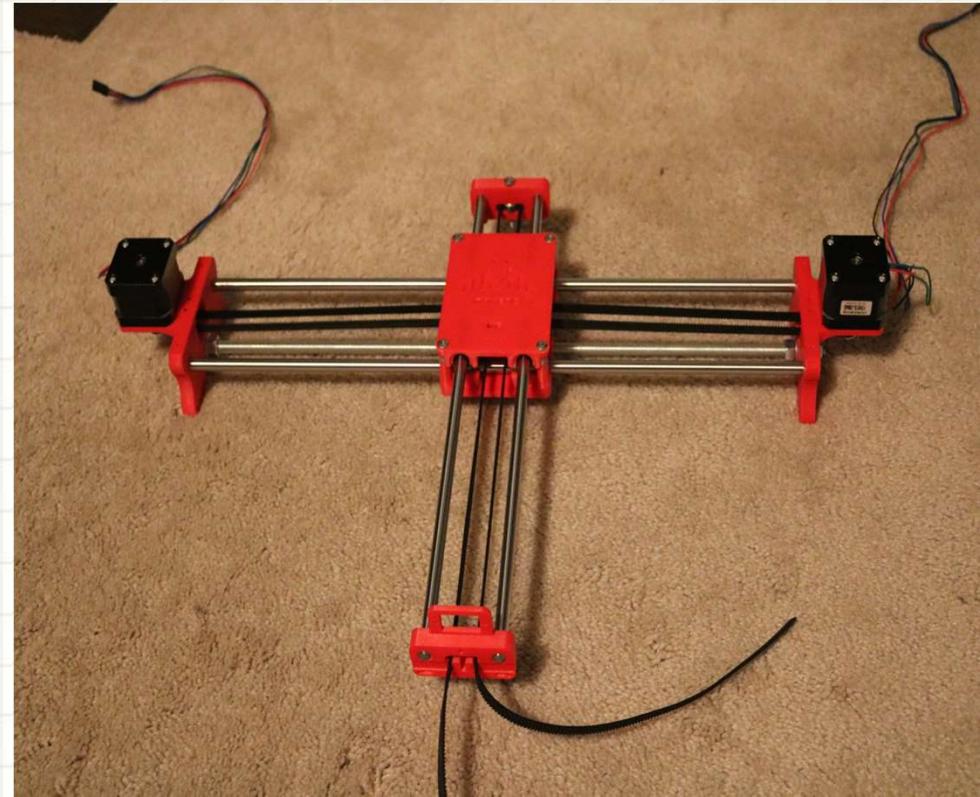
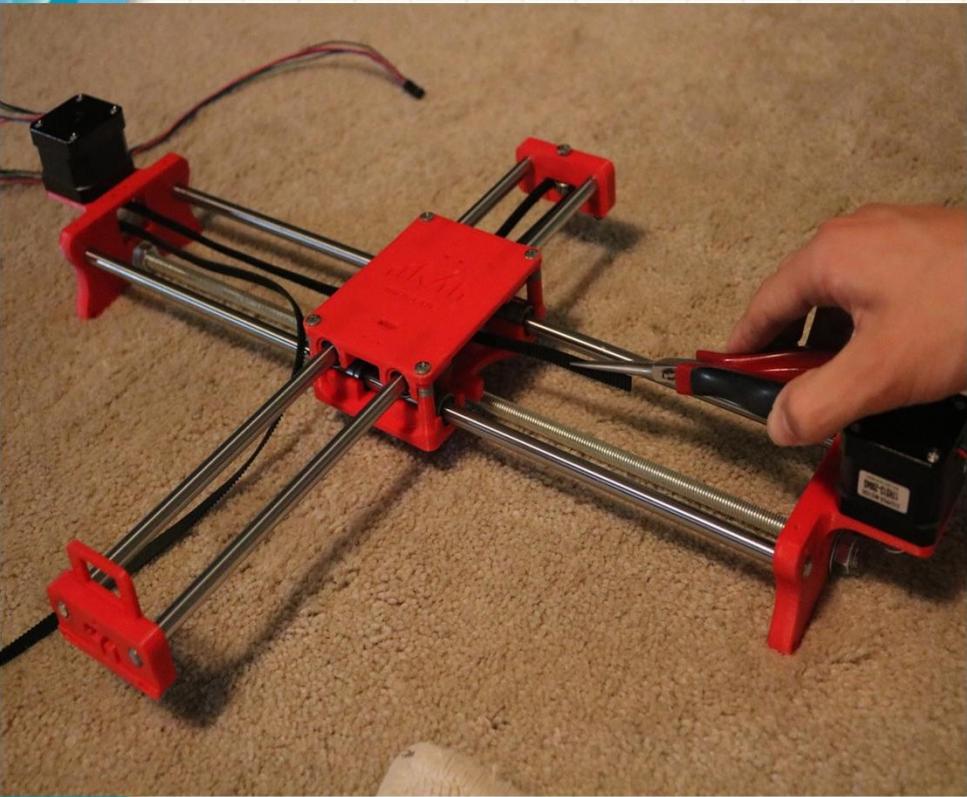


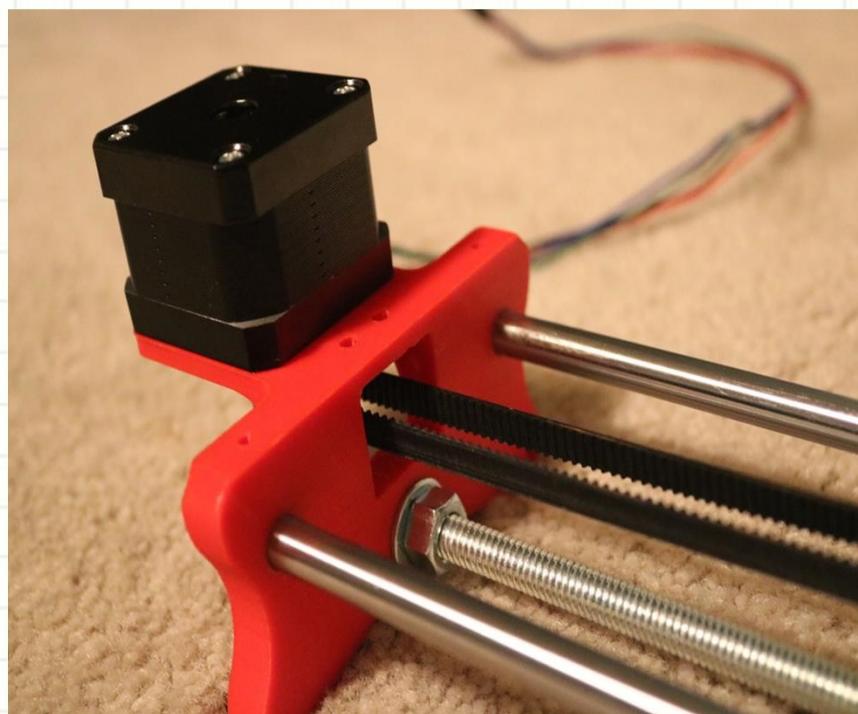
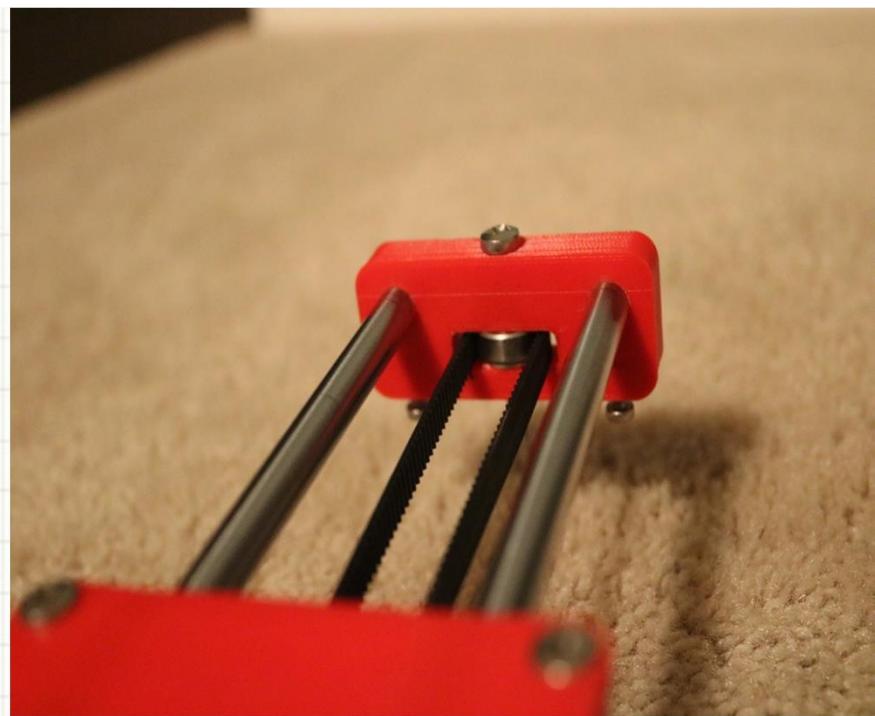
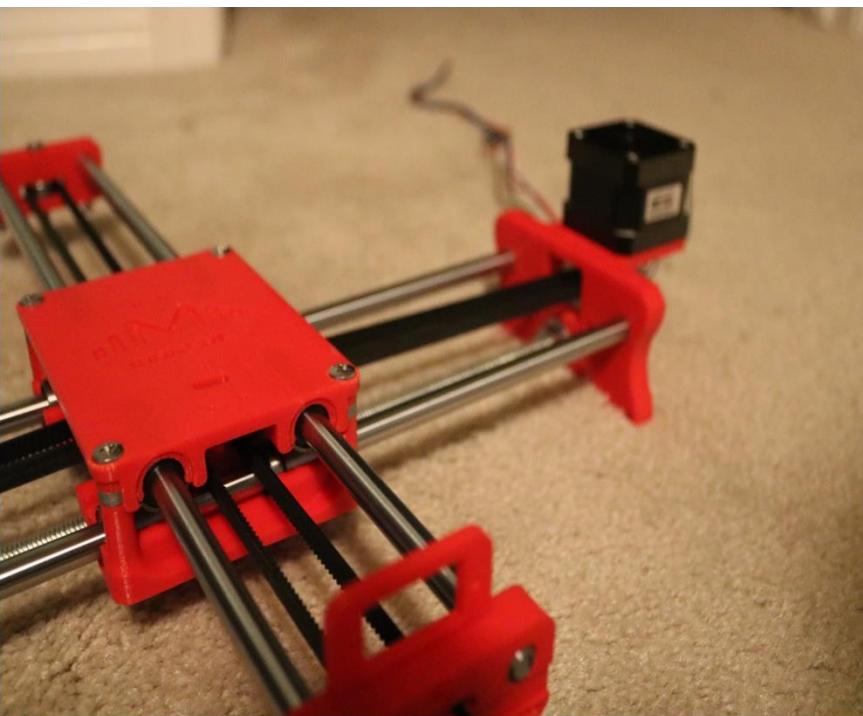
Assemble the X-Y Axis (Belt)

- Use a pair of needle nose pliers to help guide the GT2 belt more easily through the clamshell
- Take the two ends of the belt and slide them through the “teeth” on the Base Slider
- The belt should be tight and not loose
- Note that once the GT2 belt is on, it is normal for the clamshell not to move easily

Belt Diagram

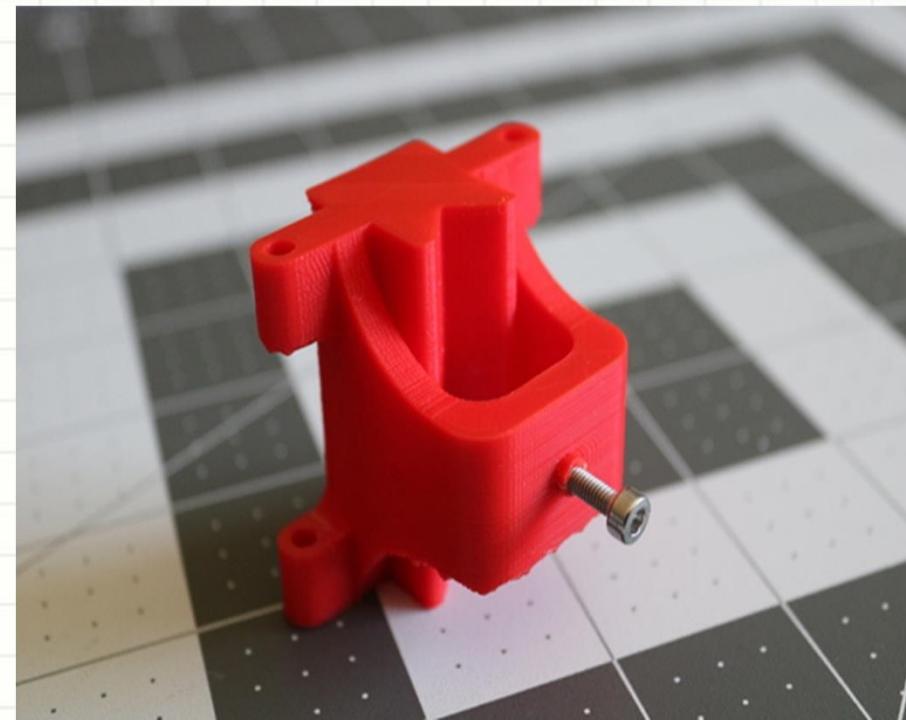
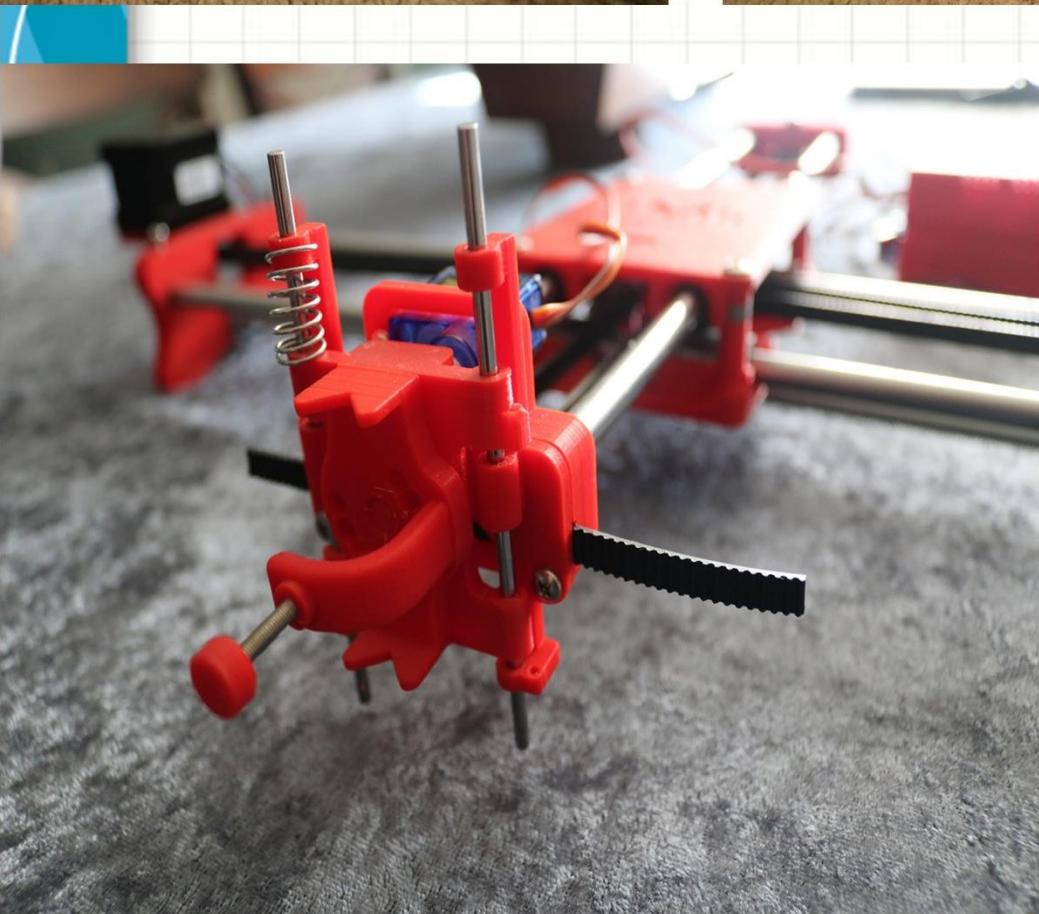
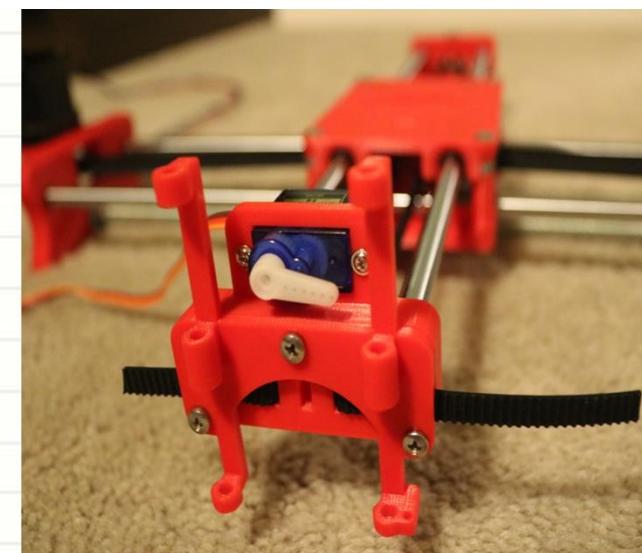
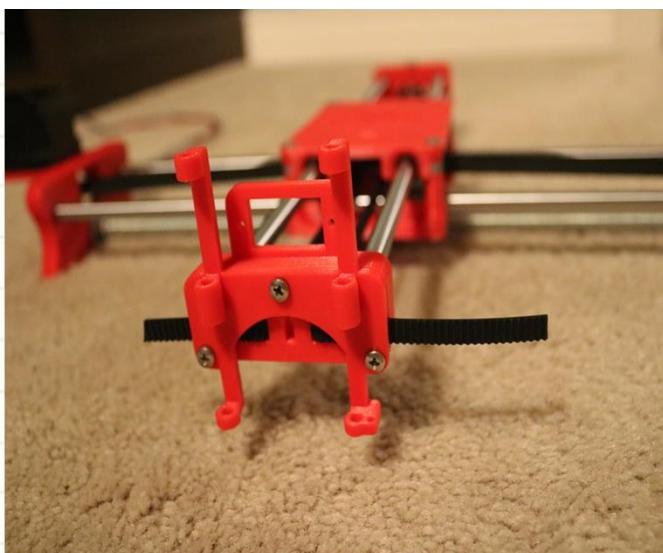
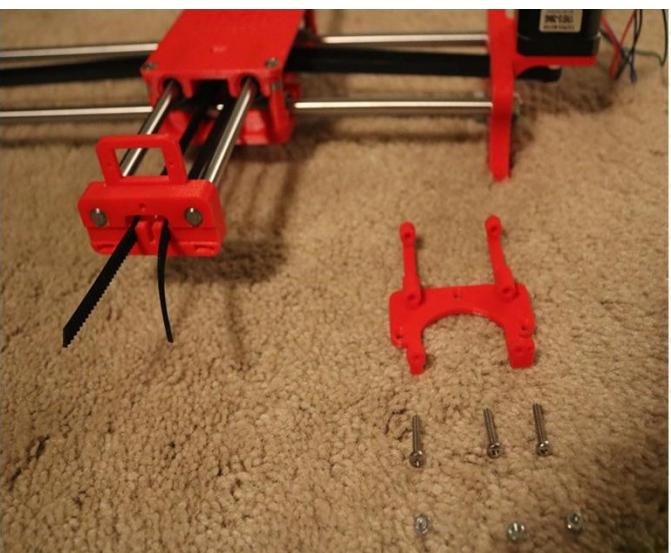






Assemble the Z-Axis

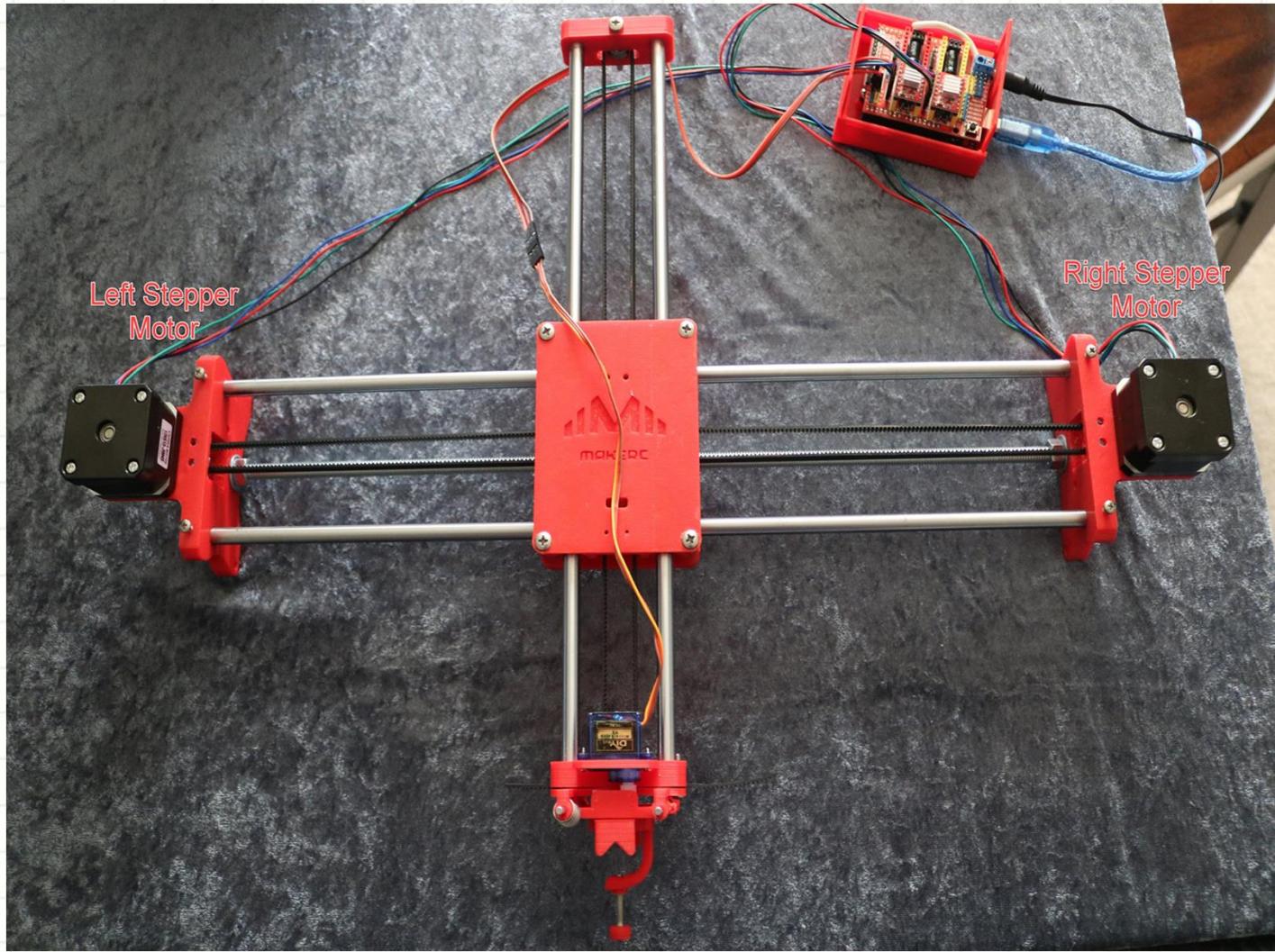
- Get **(2)** 3mm linear rods and the following 3D printed parts:
 - Slider
 - Pen Holder
 - Base Slide
 - 3MM Metric Thumb Screw
- Get **(1)** Hex M3-0.5 x 20mm screw and the Metric Thumb Screw and push them together. Use superglue to keep it together.
- Get **(3)** M3-0.5 x 16mm screws which you will use to secure the Base Slide to the Y-Front part. You may need to use **(3)** M3-0.5 nuts in order to hold it in place
- Push the Slider and Pen Holder together to make one piece
- Now take that new part and the **(2)** 3mm linear rods and slide the rods through the holes. Place a small spring in between the two parts so there is a little bit of pressure to lift the Slider. You may need to cut the spring a bit until there is an adequate amount of pressure on the slider.
- You may also 3D print the improved pen holder which I designed: <https://www.thingiverse.com/thing:2782375>



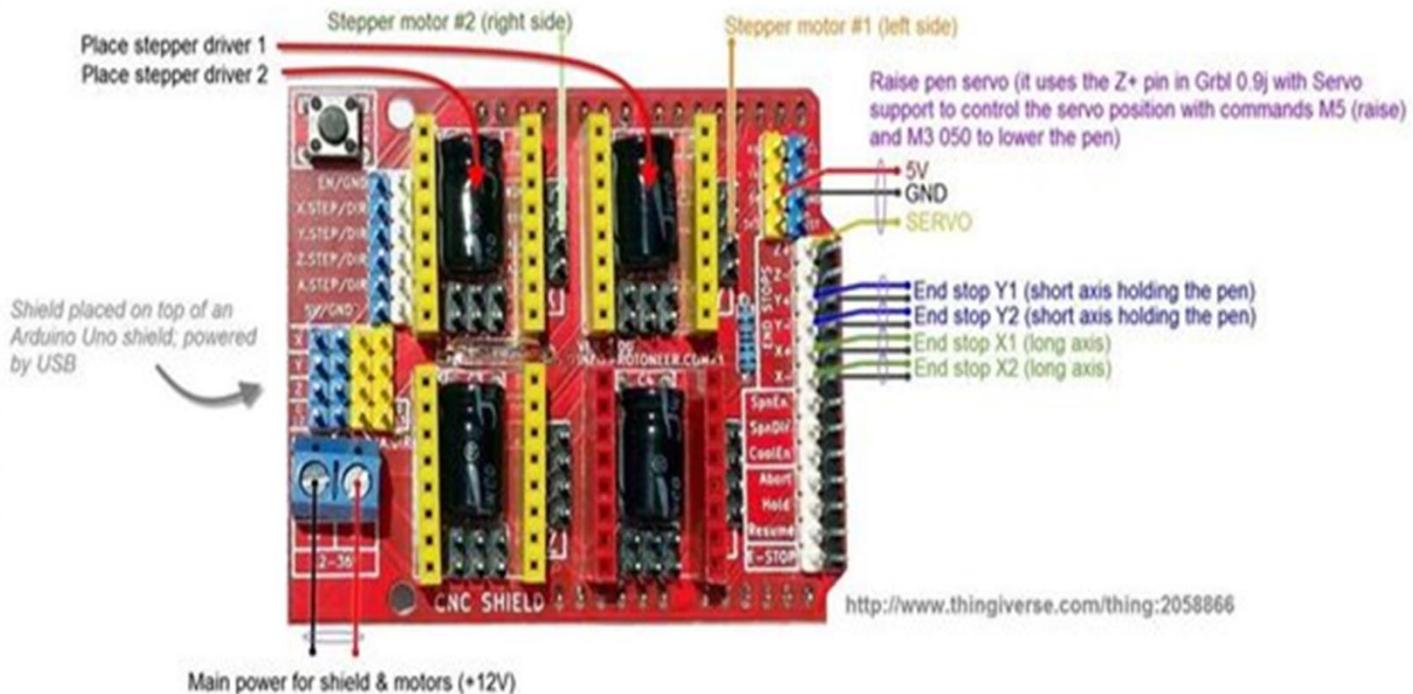
CNC Shield Setups

Connecting the CNC Shield

- There one of two ways to power the CNC shield:
 - Solder an 18 gauge wire as shown on the following slide which powers the CNC shield from the Arduino
 - Power the CNC shield through its power connector.
- Install 3 jumpers on the CNC shield for each stepper driver
- Connect Stepper Drivers:
 - For A4988 Stepper Drivers , connect the stepper driver with the screw facing away from the pushbutton as seen in the photos. Once they are plugged in you can test if they work by moving the robot around using the Rabbit GCode Sender. If your robot moves very slowly or does not move at all, double check all your wiring or use a small screwdriver and adjust the screw on the stepper driver. Note: you may or may not need to adjust the screw but if you do, adjust it **very** slightly
- Wire the servo as shown in the photo in the following slides

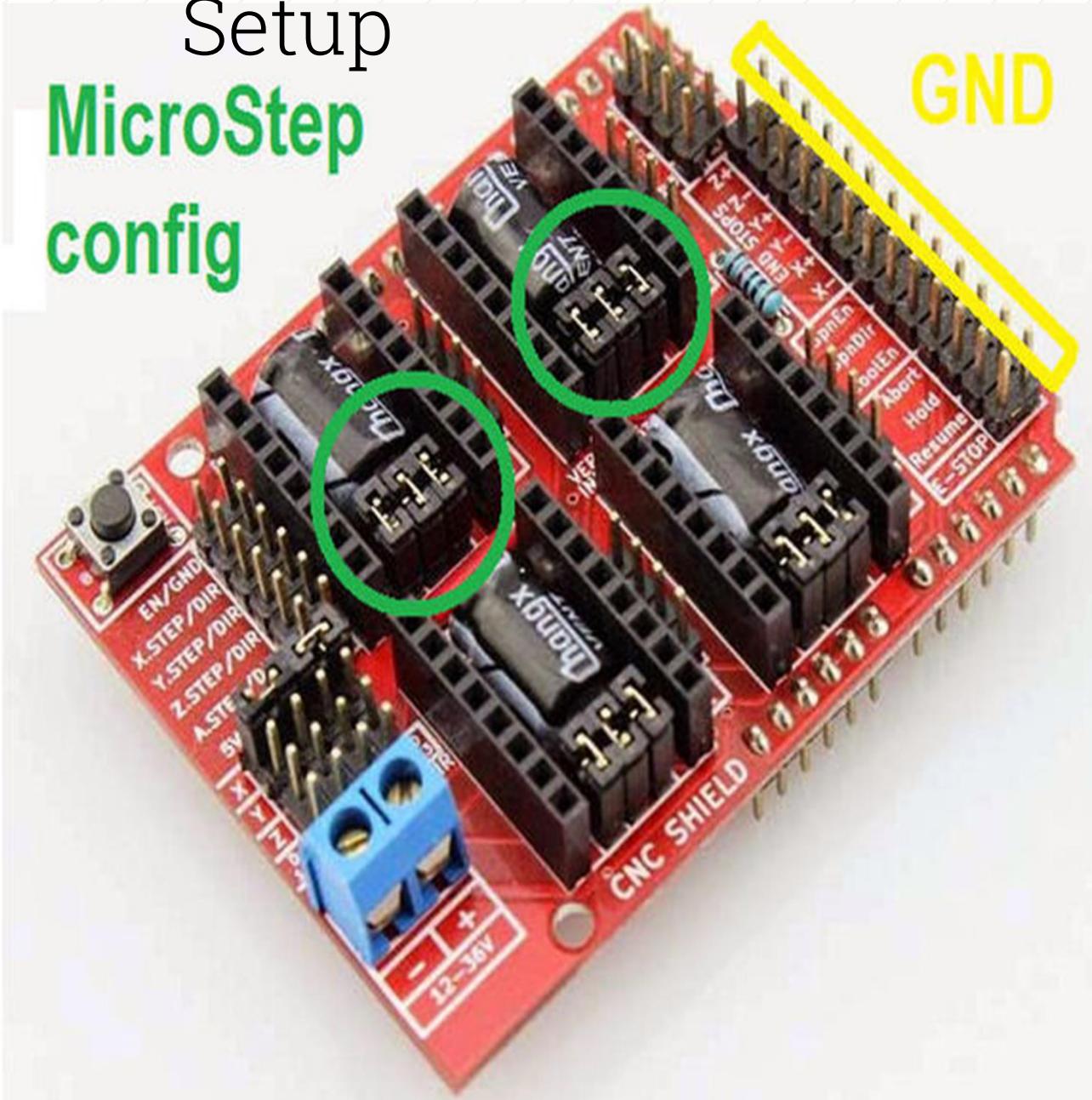


Wiring Diagram

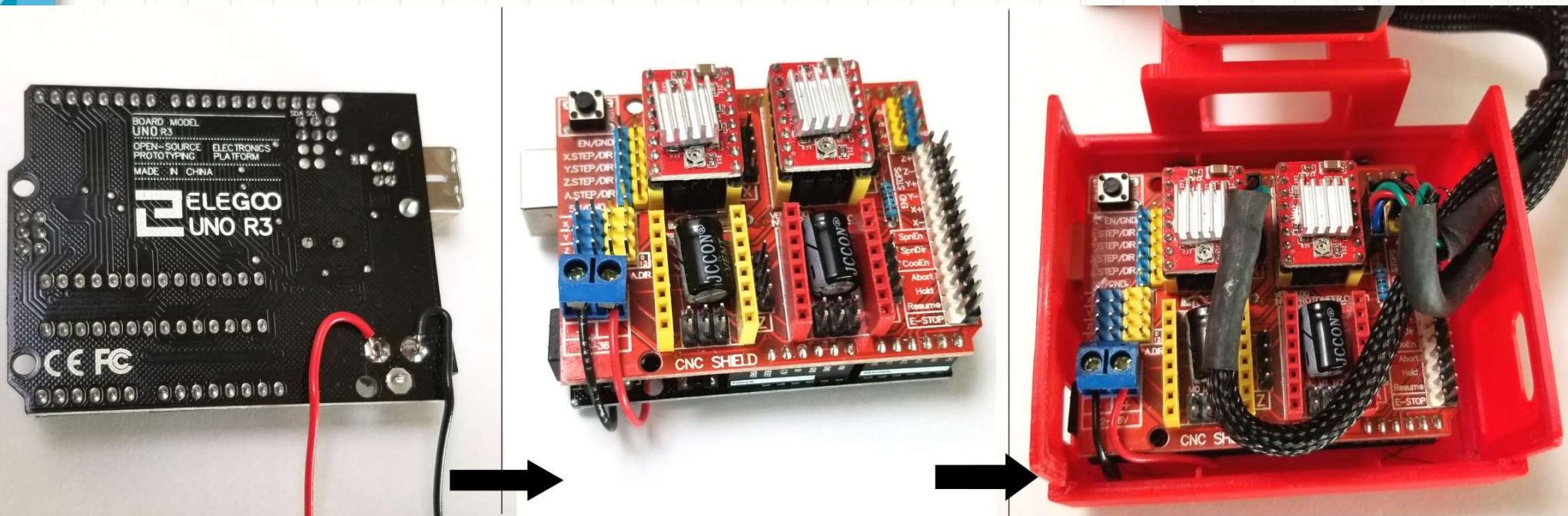


Stepper Driver Setup

MicroStep
config



CNC Power



Connecting Power From Arduino to CNC Shield

Setting the Stepper Drive Current

Setting the stepper current drive for the A4988 is unfortunately not consistent across manufacturers. If you want to estimate, I would try 0.1V per 200ma of drive current. 0.5V would mean 1A of drive current which is more than enough. If your actual ratio is 0.1V per 100ma of drive current, you will end up with 0.5A of drive current which should also work. If you want to be sure, you will have to ask the company that made your driver or check the board yourself.

The drive current does not govern the speed. Around 700ma of driver current is just about perfect and increasing the current does not make the robot go faster. It does generate more heat on your driver.

Install GRBL on Arduino

- GRBL first needs to be installed onto the Arduino Uno in order for you to send Gcode commands with the Rabbit Gcode Sender
- Download the GRBL firmware here:
<https://github.com/robottini/grbl-servo>
- You can follow the guide on how to install GRBL here:
<https://github.com/gnea/grbl/wiki/Compiling-Grbl>
 - Before compiling the arduino sketch, locate the config.h file and replace it with the new one from the thing files

Install Extension and Set Parameters

- GRBL first needs to be installed onto the Arduino Uno in order for you to send Gcode commands with the Rabbit Gcode Sender
- Download the GRBL firmware here: [Link](#)
- You can follow the guide on how to install GRBL here: [Link](#)
 - Before compiling the arduino sketch, locate the config.h file and replace it with the new one from the thing files

Install Extension and Set Parameters

- GRBL first needs to be installed onto the Arduino Uno in order for you to send Gcode commands with the Rabbit Gcode Sender
- Download the GRBL firmware here: [Link](#)
- You can follow the guide on how to install GRBL here: [Link](#)
 - Before compiling the arduino sketch, locate the config.h file and replace it with the new one from the thing files

GRBL Parameters

- You will need to change the GRBL parameters before you start drawing with your machine
- Enter these values as seen to the right in the command box
- For example, type in: \$121 = 500.000

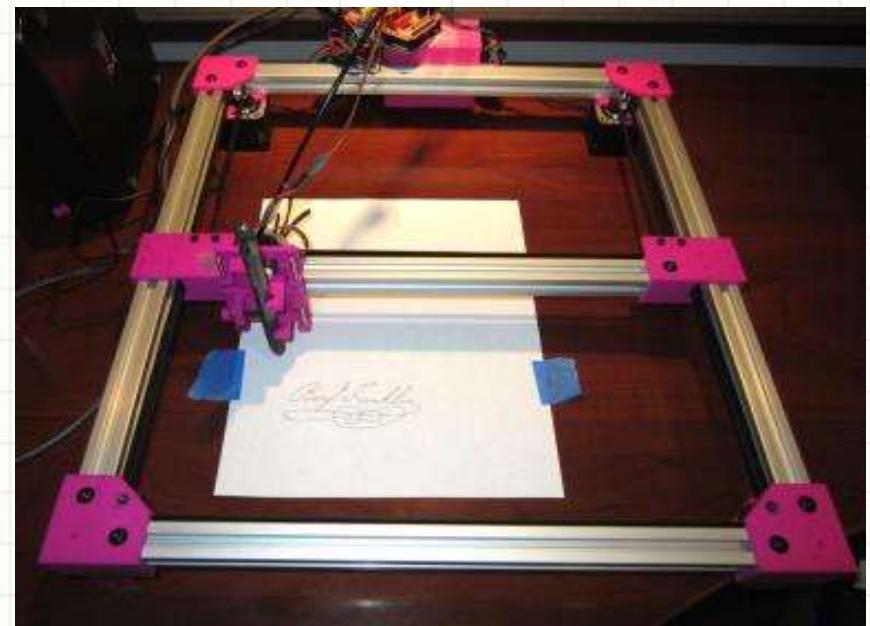
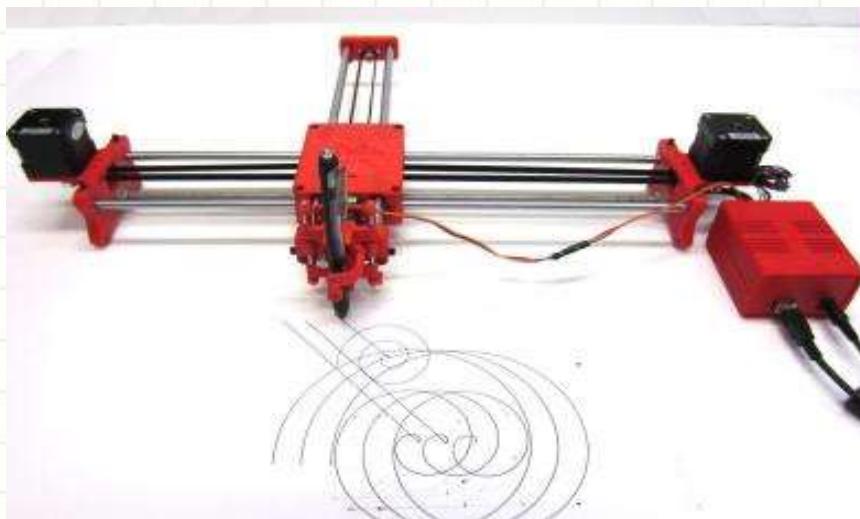
ID	Value	Description
\$0	10	(step pulse, usec)
\$1	25	(step idle delay, msec)
\$2	0	(step port invert mask:00000000)
\$3	0	(dir port invert mask:00000000)
\$4	0	(step enable invert, bool)
\$5	0	(limit pins invert, bool)
\$6	0	(probe pin invert, bool)
\$10	3	(status report mask:00000011)
\$11	0.010	(junction deviation, mm)
\$12	0.002	(arc tolerance, mm)
\$13	0	(report inches, bool)
\$20	0	(soft limits, bool)
\$21	1	(hard limits, bool)
\$22	0	(homing cycle, bool)
\$23	0	(homing dir invert mask:00000000)
\$24	25.000	(homing feed, mm/min)
\$25	500.000	(homing seek, mm/min)
\$26	250	(homing debounce, msec)
\$27	1.000	(homing pull-off, mm)
\$100	250.000	(x, step/mm)
\$101	250.000	(y, step/mm)
\$102	250.000	(z, step/mm)
\$110	2000.000	(x max rate, mm/min)
\$111	2000.000	(y max rate, mm/min)
\$112	2000.000	(z max rate, mm/min)
\$120	500.000	(x accel, mm/sec^2)
\$121	500.000	(y accel, mm/sec^2)
\$122	500.000	(z accel, mm/sec^2)
\$130	200.000	(x max travel, mm)
\$131	200.000	(y max travel, mm)
\$132	200.000	(z max travel, mm)



DRAWING ROBOT SOFTWARE INSTALL

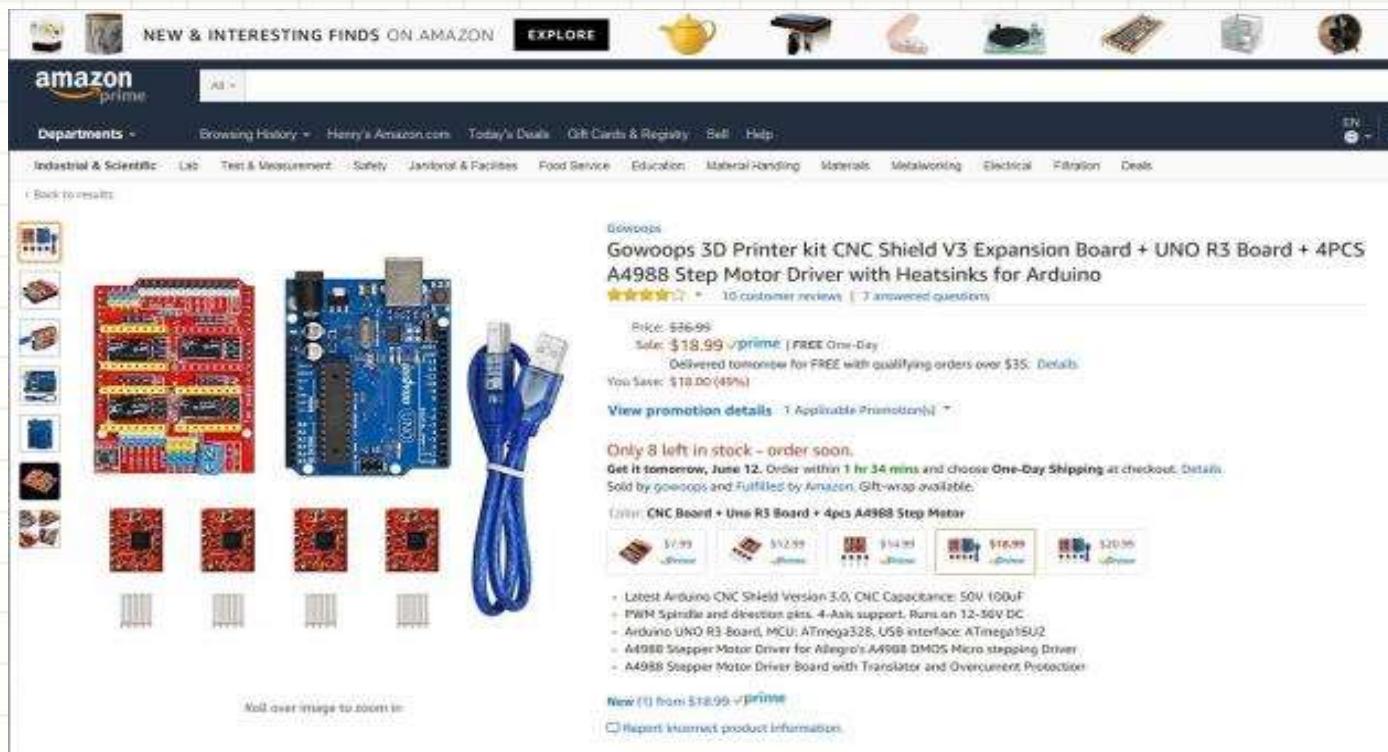
Drawing Robot Software

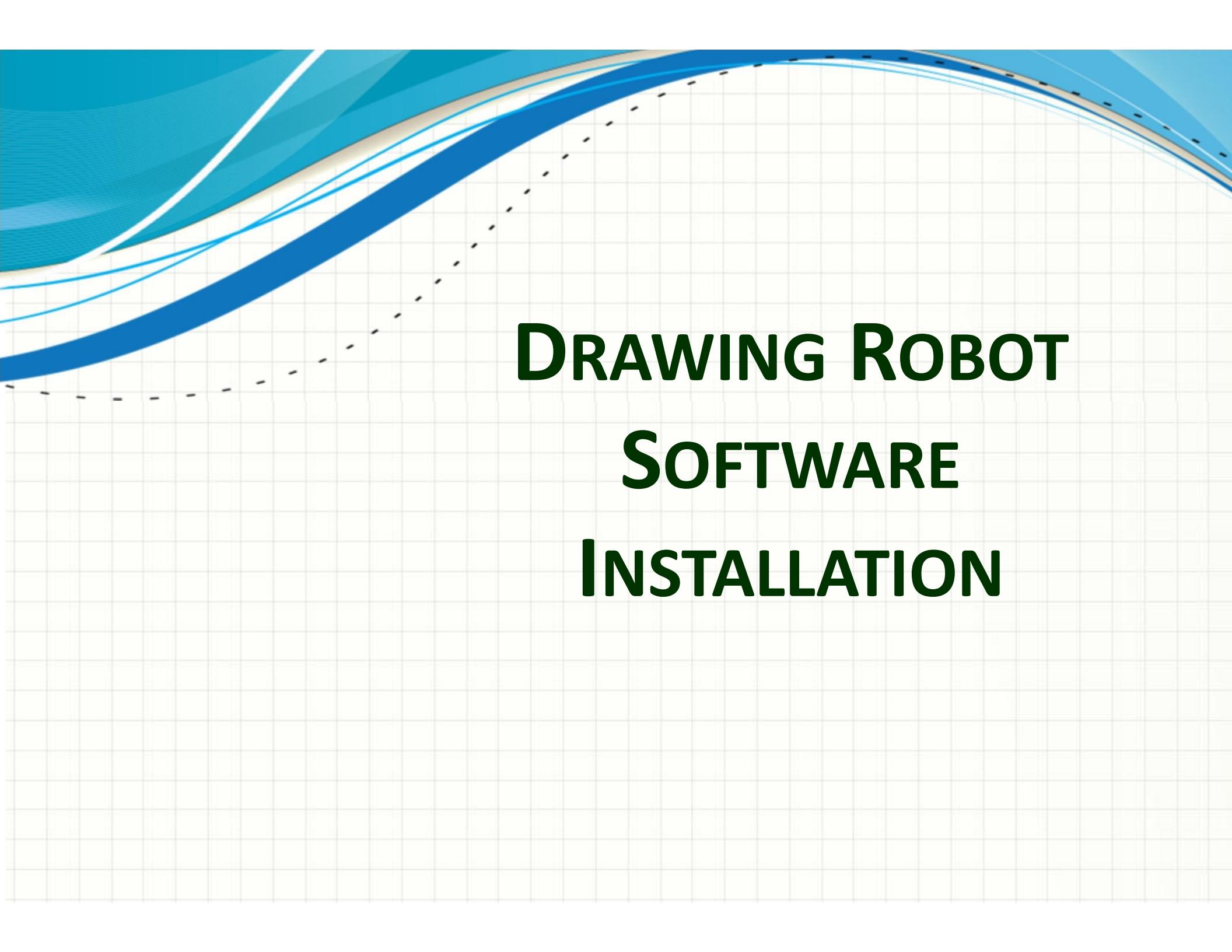
If you are viewing this document, you probably just finished assembling your Drawing Robot. In order to use your Drawing Robot you will need to install software on your computer. This document explains how to install the software, how to use your Drawing Robot , and how to do basic operations with Inkscape. These instructions will work with both the open beam style Drawing Robot and the 2020 DrawBot closed frame style.



Drawing Robot Software

Both of these drawing robots use an Arduino as the computer and an extra module called a CNC Shield. This computer should have already been programmed with the firmware and configured with parameters. If it hasn't, you need to refer back to the original instructions for building either of these drawing robots.





DRAWING ROBOT SOFTWARE INSTALLATION

Drawing Robot Software

You will need to install three pieces of software to make your DrawBot work.

- **Inkscape** – Graphics creation program that can create and also import graphics from other sources and then prepare the output file needed for the DrawBot to draw.
- **Inkscape MI GRBL Extension** – This software knows how to convert the vector graphics file into G-Code which the DrawBot uses to draw the graphic.



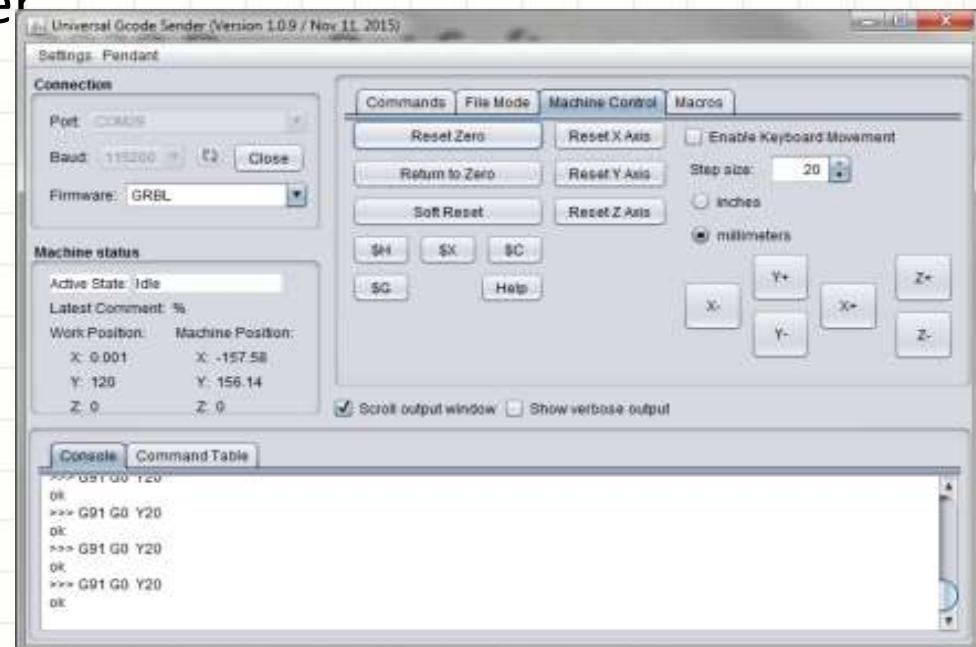
Inkscape



Drawing Robot Software

Once Inkscape and the Inkscape MI GRBL Extension have created the G-Code file needed to draw the graphic, you need a program to send the G-Code to the DrawBot through a USB port. This program is called the Universal G-Code Sender

- **Universal G-Code Sender** – This program also acts as a front panel to the DrawBot. It gives you control over the DrawBot motions, allows you to reset the DrawBot, and it allows you to send G-Code files to the DrawBot.



NOTE: Your computer must have a Java Runtime Environment installed in order to run the UGS. Do a Google search on how to add the JRE.

Why so many Software Pieces?

Here's the deal:

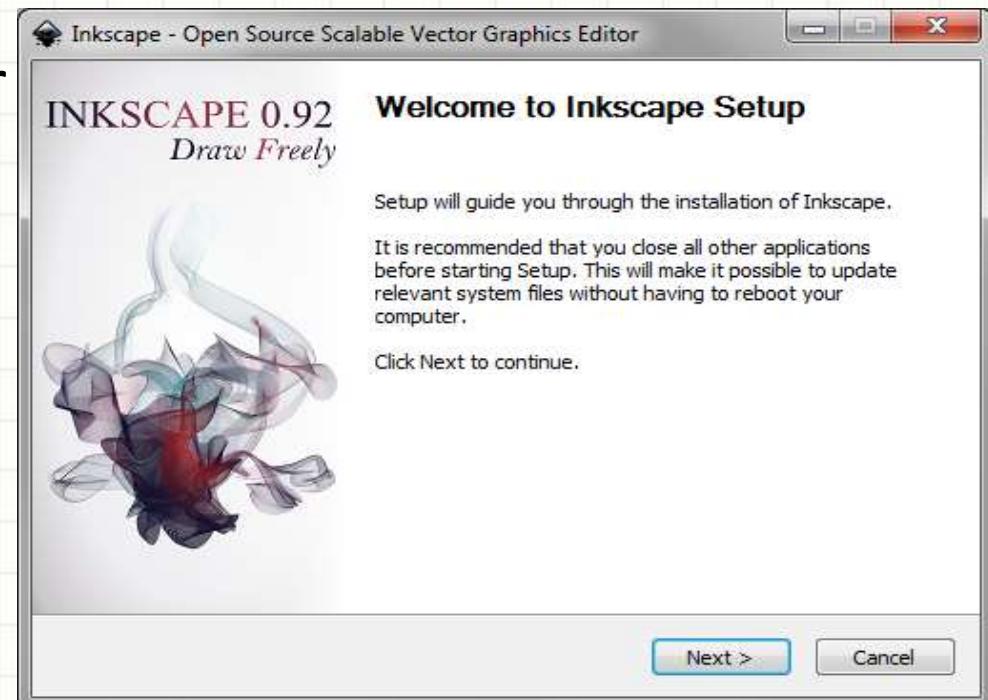
- Inkscape is a drawing program much like Photoshop or GIMP. It can create graphic objects. It has the capability for programmers to add additional features such as converting drawings into special formats such as G-Code. These features are called extensions.
- The MI Extension is a special piece of code that Inkscape uses to convert a drawing to G-Code.
- G-Code is a special format of drawing file that includes the actual motions need to guide the DrawBot into drawing your picture. Most of the format contains move instructions for the X and Y axis and Z axis commands to raise and lower the pen.
- Once you have a G-Code file, you need a way to get the DrawBot ready and to then send the G-Code file through the USB port to the DrawBot. The Universal G-Code Sender program takes care of that. It also allows you to manually move the different axis of the robot. Remember that the UGS needs a Java Run time environment in order to run



Installing Inkscape

Install Inkscape version 0.92. Other versions may work but we know that the G-Code extension works with this version.

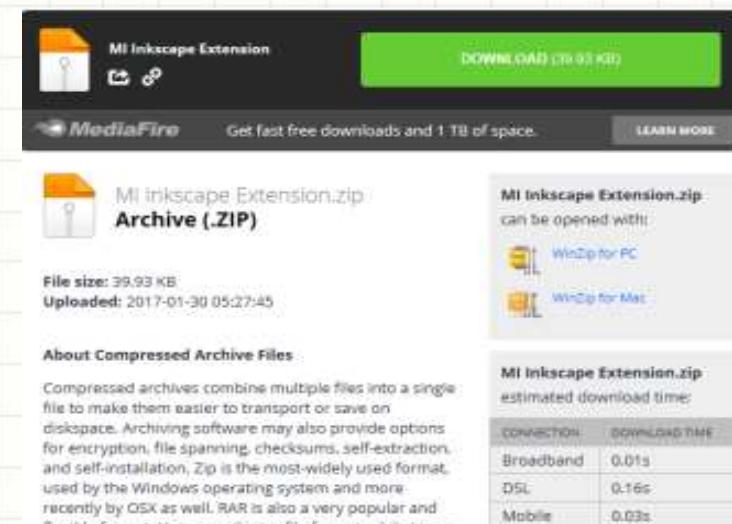
- Download the latest version of Inkscape for your computer here:
<https://inkscape.org/en/>
- You may also use the copy of Inkscape found on the flash drive that was provided to you.
- Follow the directions after executing the installation script.



Install the Inkscape MI GRBL Extension

Install the Inkscape MI GRBL Extension. You must copy the MI Extension directory contents to the
c:\\Program Files\\Inkscape\\share\\extensions directory.

- Download the Inkscape MI Extension from here:
<http://www.mediafire.com/file/ae0wquqornzc3o2/MI+Inkscape+Extension.zip>
- You may also use the copy of MI GRBL Extension from the flash drive that was provided to you.
- Copy the MI Extension files shown below to c:\\Program Files\\Inkscape\\share\\extensions



Documents library

MI Inkscape Extension

Name	Date modified	Type	Size
□ dxf_input.inx	1/9/2015 8:11 AM	INX File	2 KB
□ dxf_input.py	1/9/2015 8:11 AM	PY File	25 KB
□ servo.inx	1/29/2017 1:59 AM	INX File	2 KB
□ servo.py	5/10/2016 1:18 PM	PY File	143 KB

Install the Universal G-Code Sender

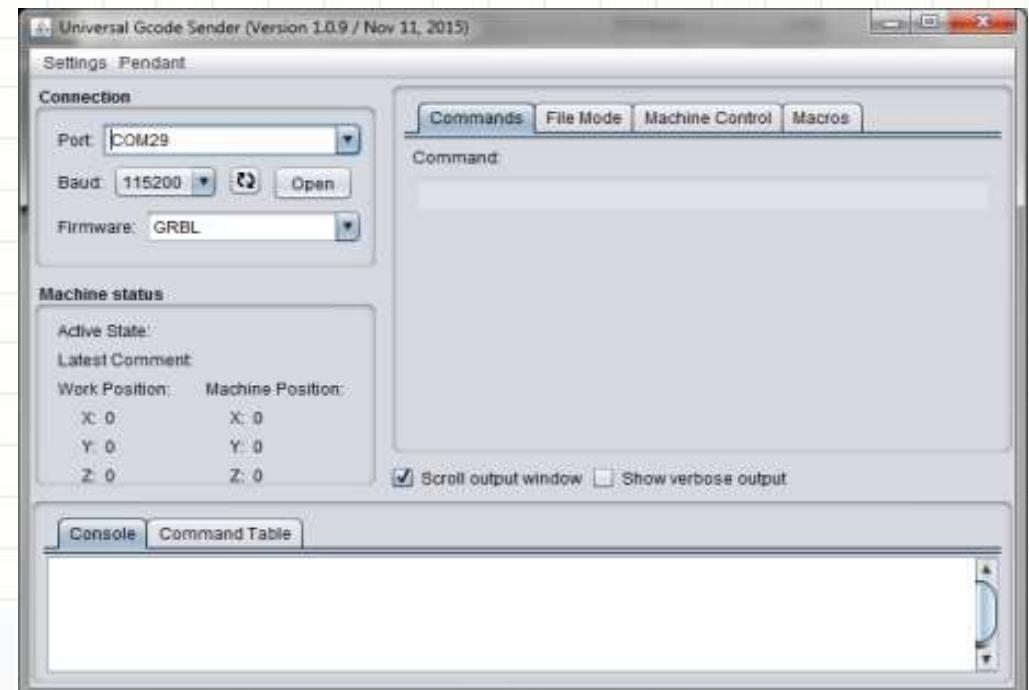
Make a directory for your copy of the Universal G-Code Sender UGS

- Get the Universal G-Code Sender from here:
https://winder.github.io/ugs_ website/download/
- Copy the files from the flash drive or unzip them into your new directory.
- To execute the UGS, execute

Documents library

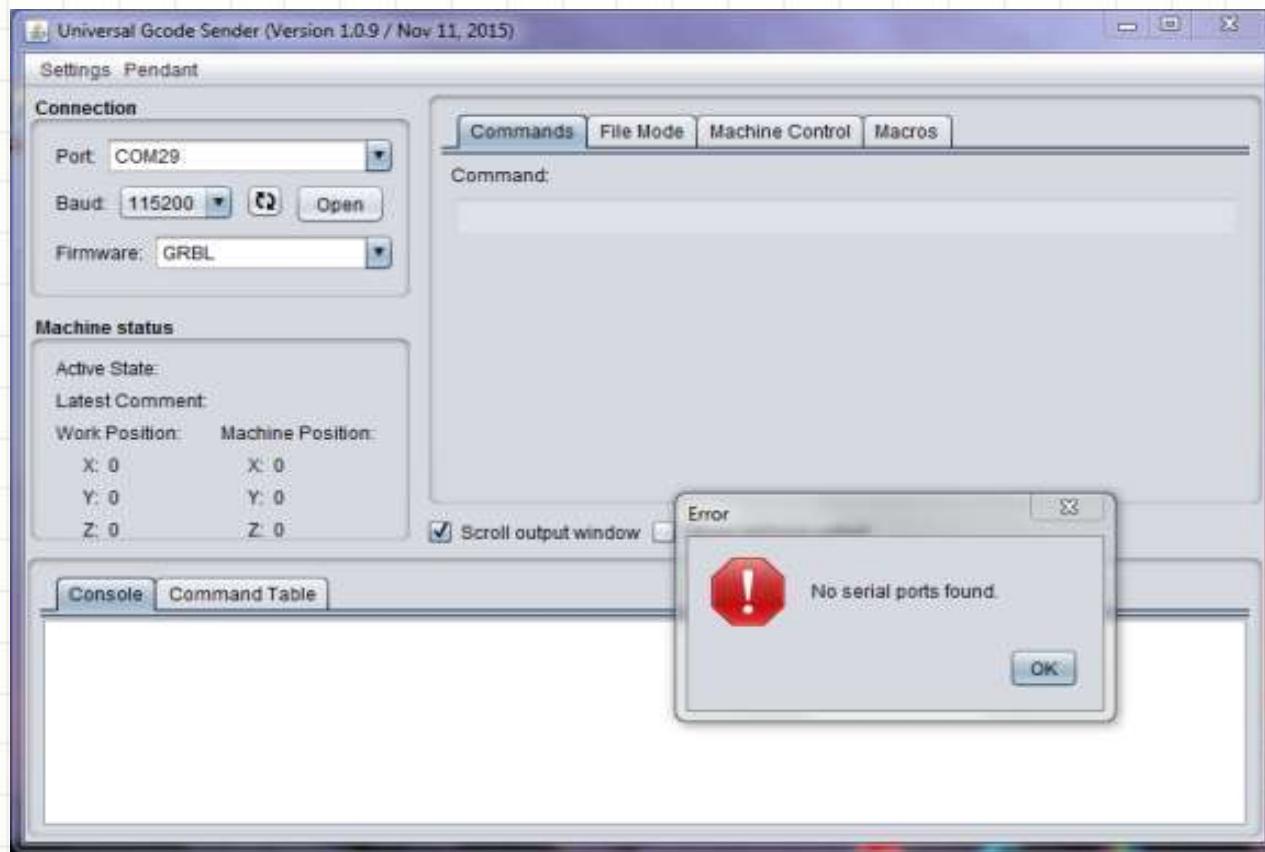
UniversalGcodeSender

Name	Date modified	Type	Size
README.txt	10/30/2015 4:41 PM	Text Docu...	1 KB
start.sh	10/30/2015 4:41 PM	SH File	1 KB
start-windows.bat	10/30/2015 4:41 PM	Windows ...	1 KB
UniversalGcodeSender.jar	11/11/2015 9:21 AM	Executable...	10,526 KB
UniversalGcodeSender-v1.0.9.zip	5/26/2017 1:24 PM	WinZip File	9,843 KB



Install the Universal G-Code Sender

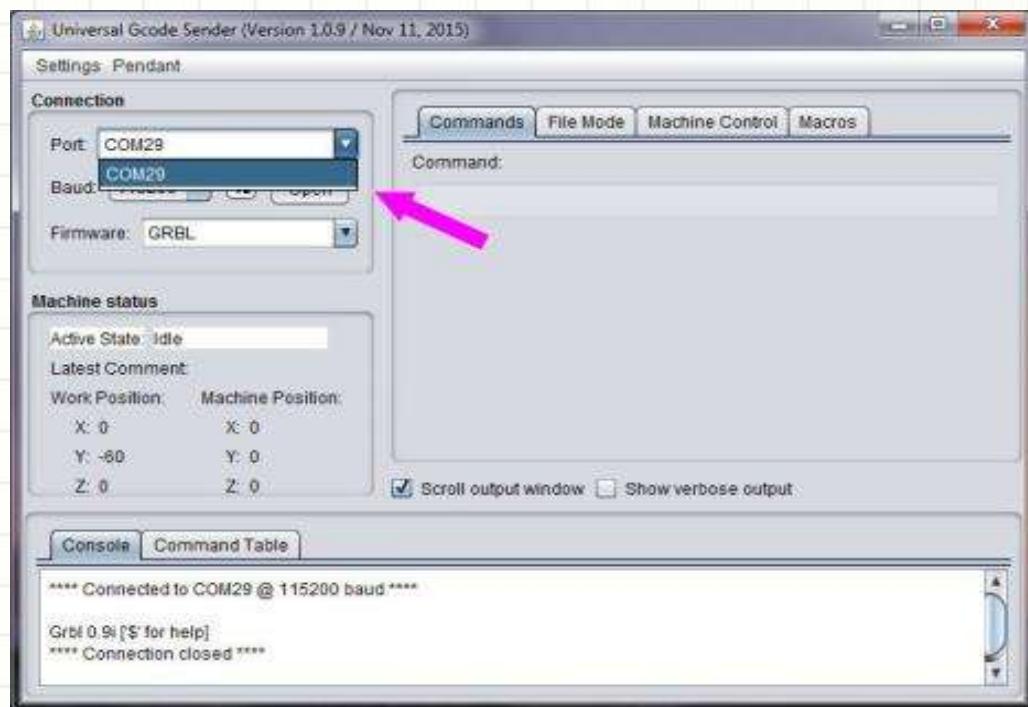
If you start the UGS and it does not find your DrawBot, you will get an error message as shown here. You will need to install a USB driver to make the DrawBot visible to your USB port. The next slide explains how. If the UGS won't start, you probably need to install a Java JRE.

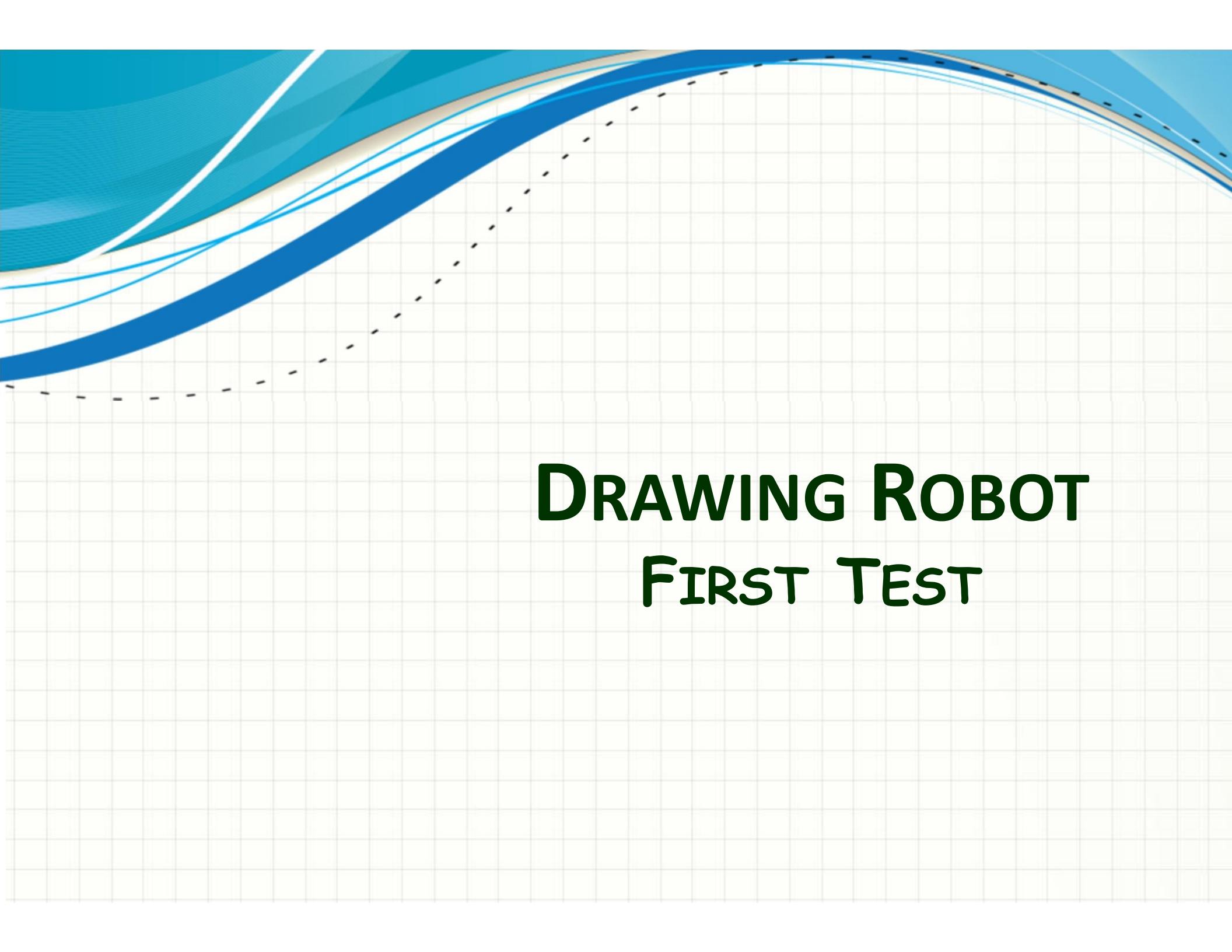


If Your Computer doesn't see the DrawBot

If your computer does not recognize your Drawing Robot, you may need to manually install the USB drivers. This is similar to when you install a printer or a mouse. If you did not get an error message, skip these instructions.

- Follow the USB driver installation instructions here:
<https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/windows---quick-and-easy>
Reboot your computer and try plugging the DrawBot into a USB port.
- When you start the Universal G-Code Sender, it should report that it found the serial port by listing the COMXX port. It probably won't be 29 as shown. It can be an number between 02 and 32.





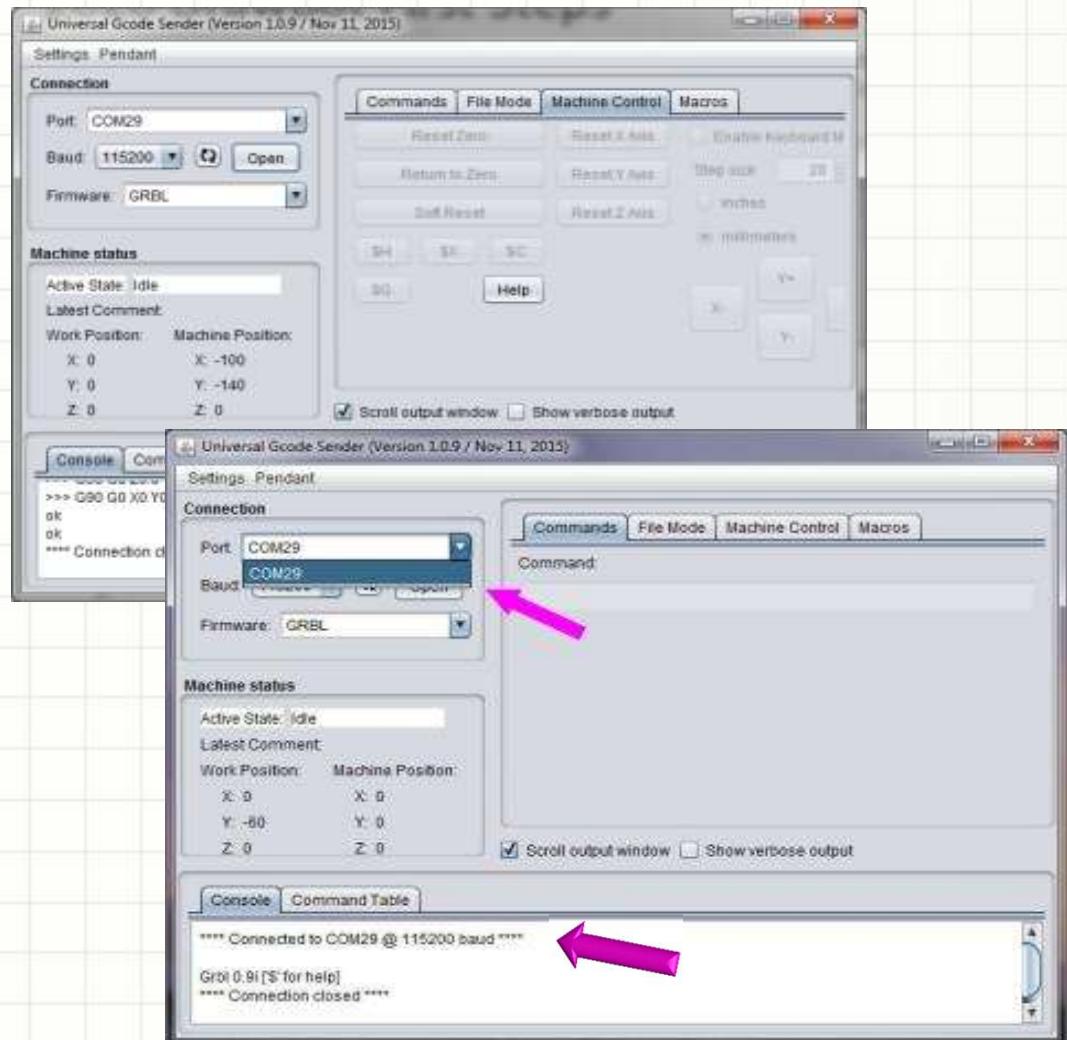
DRAWING ROBOT

FIRST TEST

Drawing Robot First Test – Step 1

Before we create custom drawings and convert them to G-Code, we will test the DrawBot by sending an existing G-Code file using the Universal G-Code Sender (UGS). This is a good first step to testing a DrawBot.

- Start the UGS by executing the “*start-windows.bat*” file in your Universal G-Code Sender directory.
- The UGS should start up and show an available COM port.
- Make sure the COMXX port is selected and then press “Open”
- You should see a message in the Console stating that UGS was able to connect to COMXX like shown in the lower screen shot.



Drawing Robot First Test – Step 2

If you were able to open the connection to your DrawBot, you should be able to test the motion. Make sure the DrawBot pen carriage is somewhere in the middle of the frame. You may gently move it with your hand.

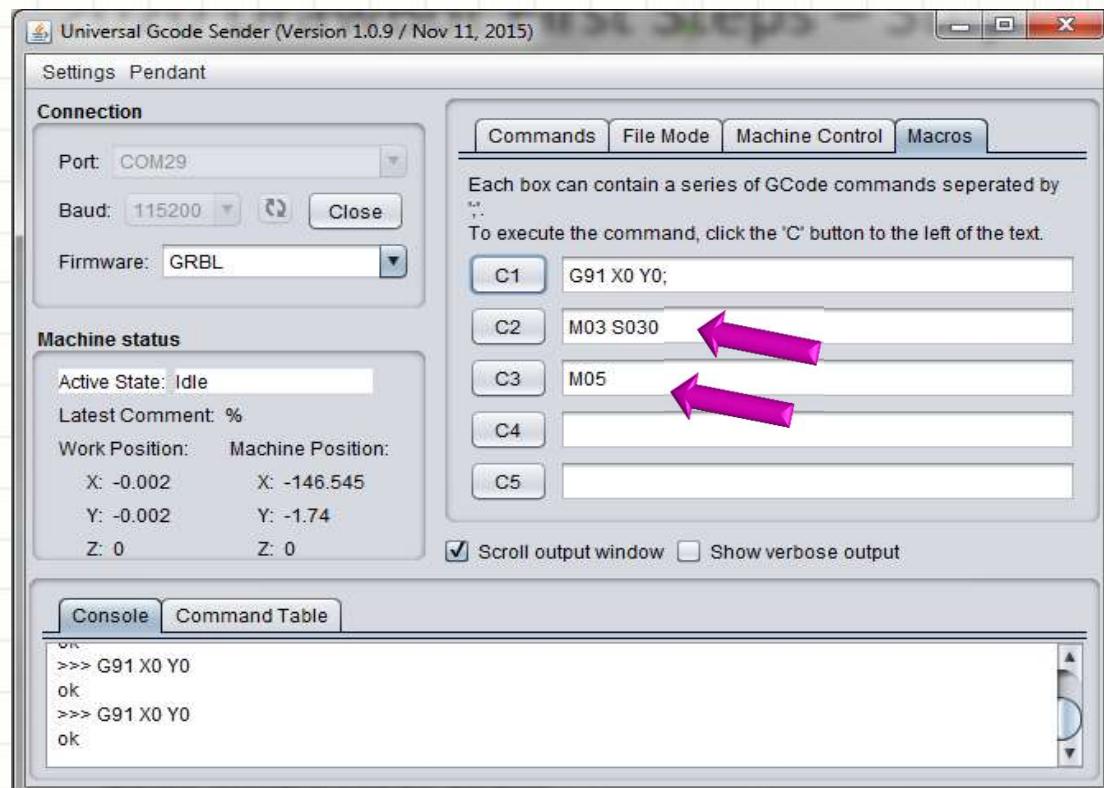
- Select the Machine Control Tab. The buttons should be enabled.
- Change the “Step size” to either 10 or 20.
- Try pressing either +Y or +X. The carriage should move by 20 mm in the direction you select for each push of the button.
- Play around with moving the carriage being careful not to go too far once the carriage is near the end of its travel. If you go too far, the motors will buzz.



Drawing Robot First Test – Step 3

In order to move the pen up and down in the Z direction, you will need to enter what are called macros. Macros are like a nickname for a set of instructions. We want to create macros for Z up and Z down of the pen

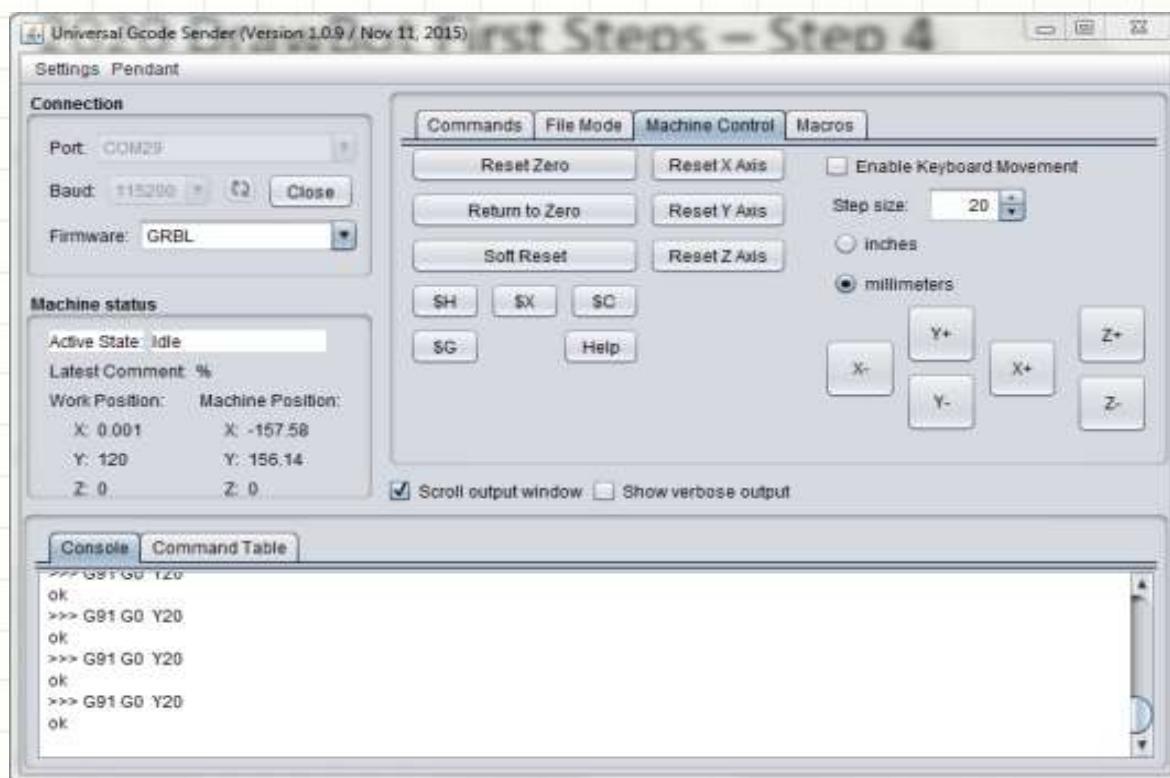
- Select the Macros Tab.
- Add the C2 Macro
M03 S030 which is Z down
- Add the C3 Macro
M05 which is Z up
- Test these Macros by
pressing the C2 and C3
buttons. Your pen should go
down and then back up.
- These macros only need to
be entered one time. The
UGS will store them for
future use.



Drawing Robot First Test – Step 4

You are now ready to test your DrawBot making its first drawing.

- Load a pen by placing it in the holder and tighten the set screw. Make sure that with the Macro command C2 (down) that the pen is on the paper and with the macro C3 (up) that the pen is off the paper.
- Move the carriage to your 0,0 (X,Y) corner using the motion commands.
- Select “Reset Zero” to set this as the origin.
- Select the “File Mode” Tab

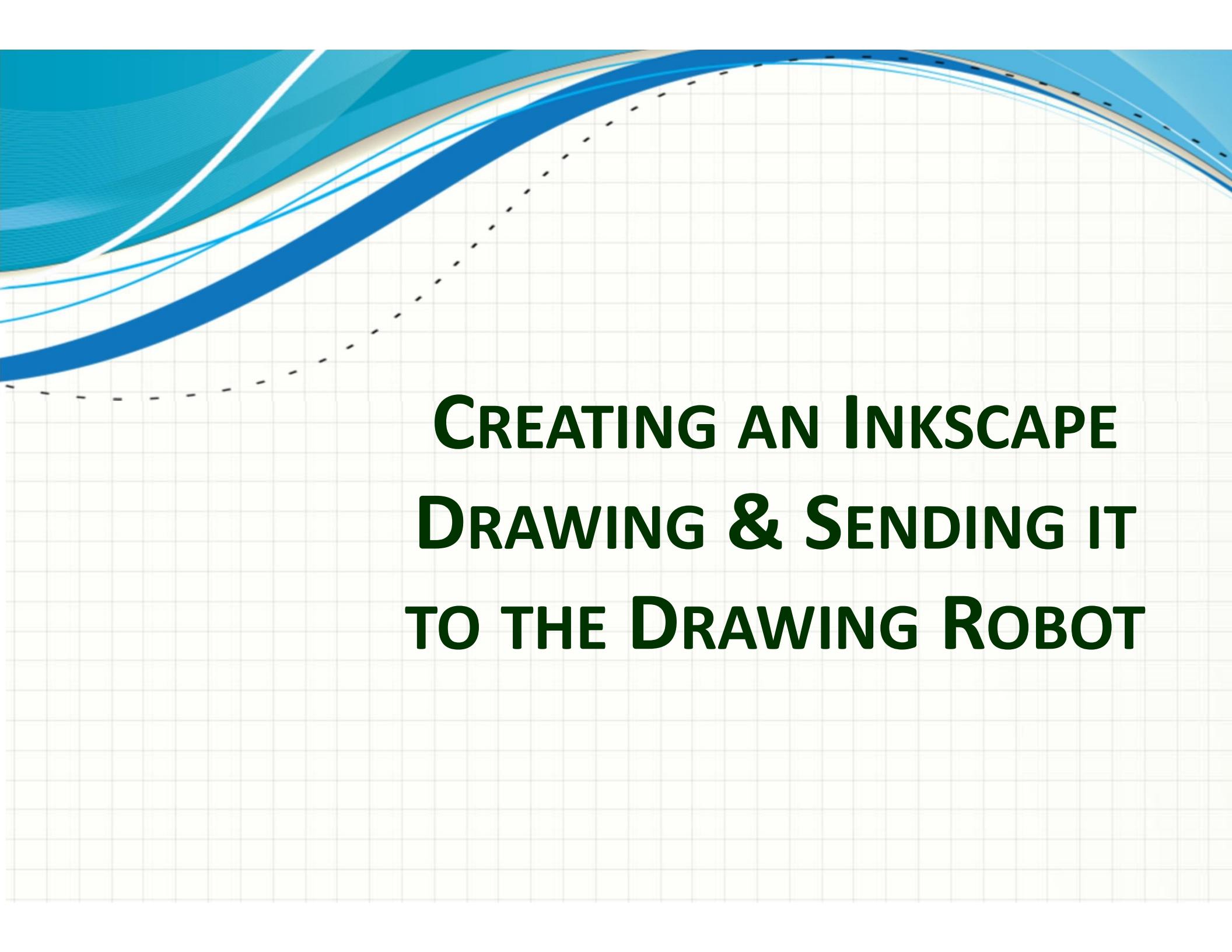


Drawing Robot First Test – Step 5

Load and draw your first file.

- From the File Mode Tab select “Browse”. A pop up will allow you to navigate to a sample file. Your flash drive will have a file called “BenFranklin_v1.1.gcode”. Select that file and “Open”.
- Select “Send” and your DrawBot should start drawing the file.
- If anything goes wrong, select “Cancel”.
- Otherwise you should get a plot like Ben Franklin’s signature.



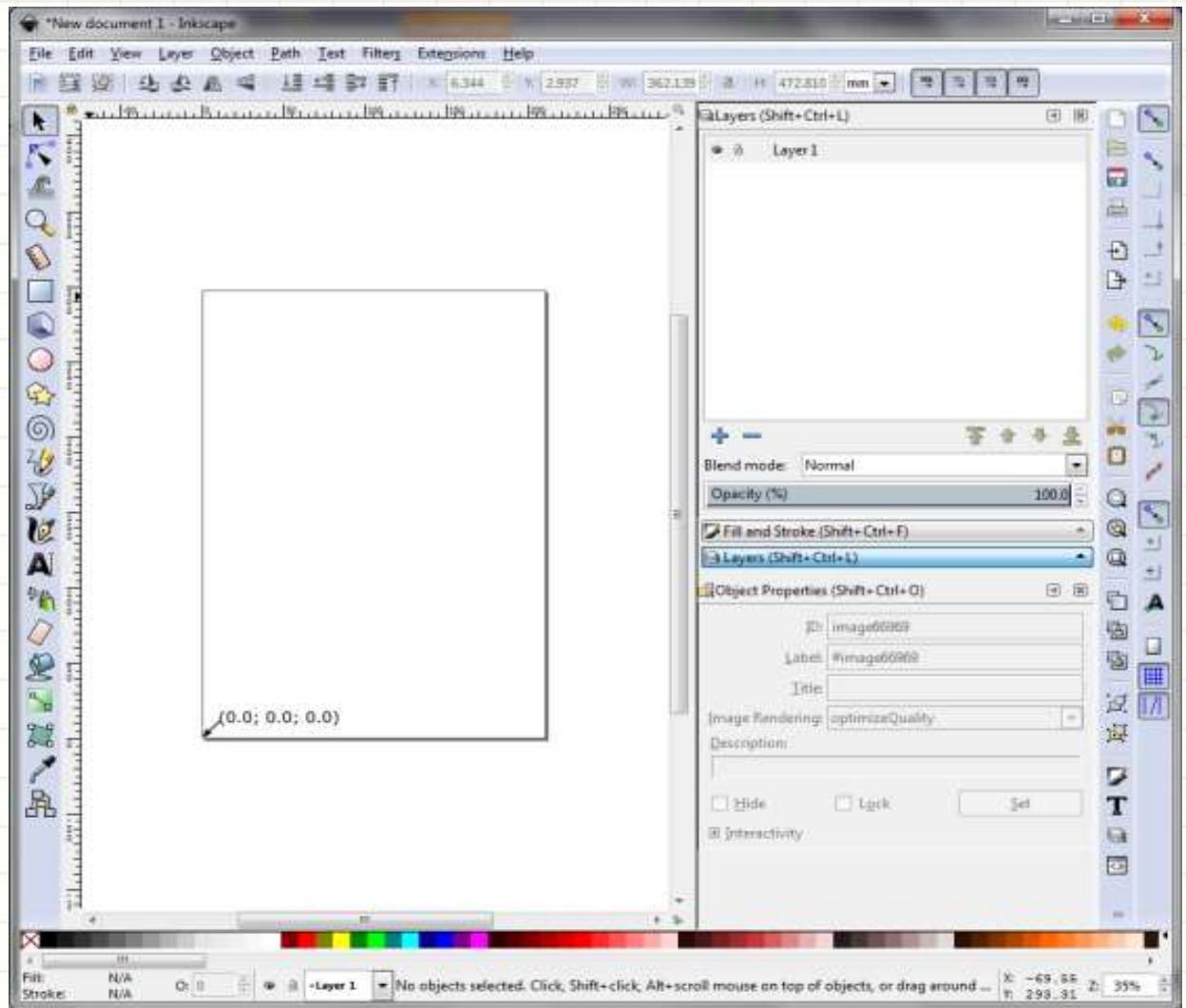


CREATING AN INKSCAPE DRAWING & SENDING IT TO THE DRAWING ROBOT

Using Inkscape – Step 1

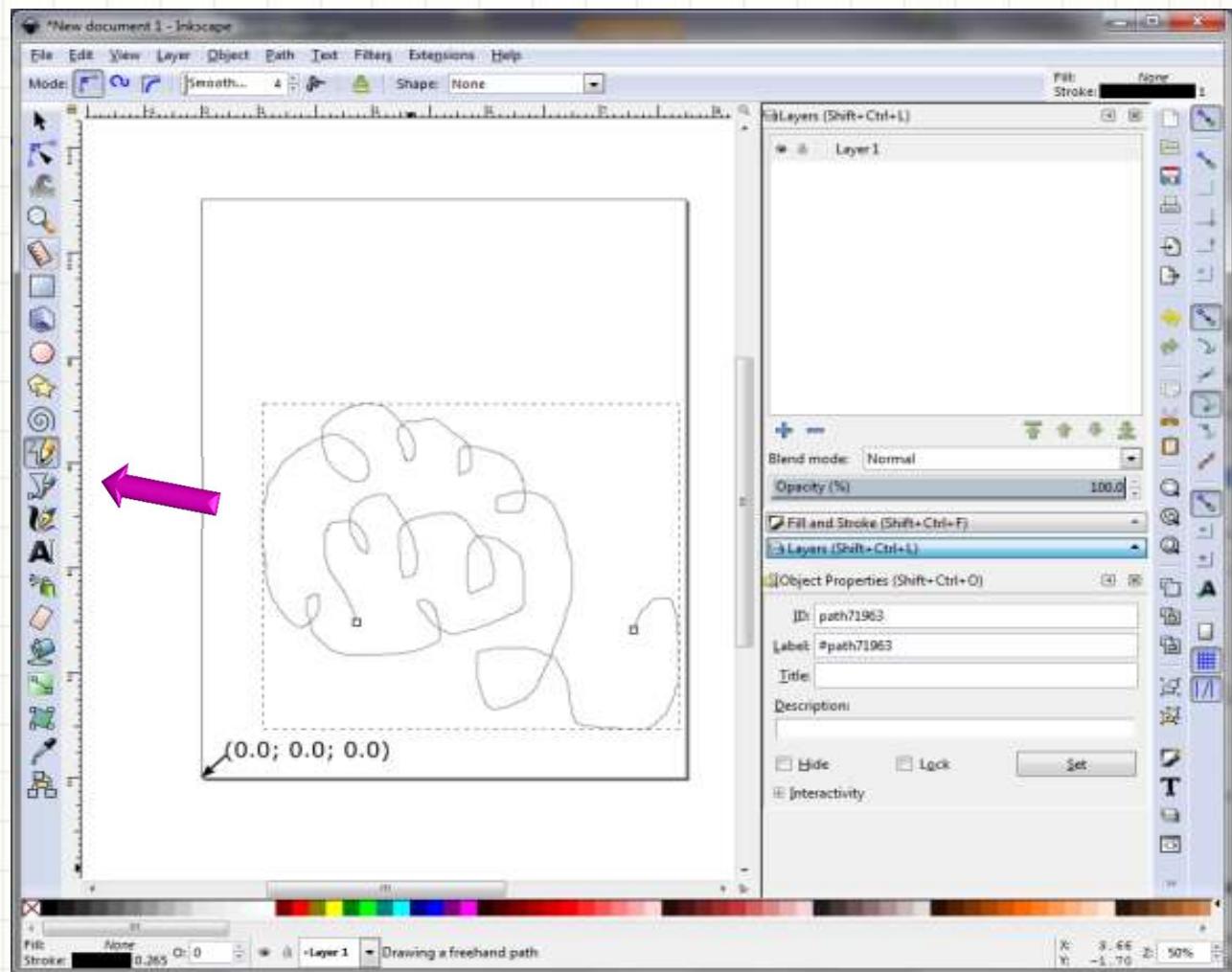
You will now create a simple drawing in Inkscape, generate G-Code, and then send it to the DrawBot.

- Open Inkscape.
It usually takes a little while for it to open and be ready for use.



Using Inkscape – Step 2

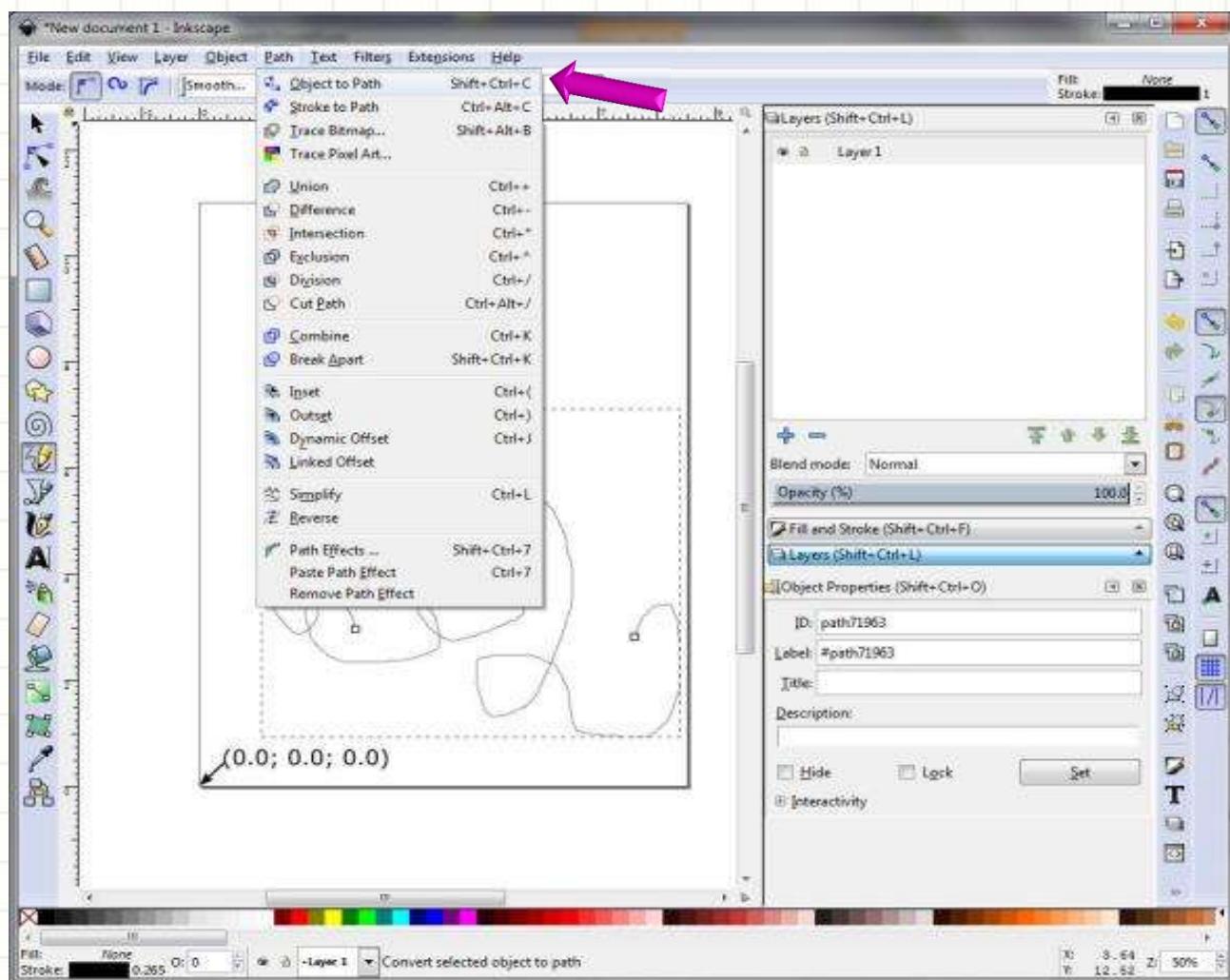
- Select the “Draw Freehand” tool which looks like a pencil. You may use other drawing tools if you like.
- Drawing anything you like such as a squiggle or something more if you are artistic.
- Make sure your drawing is selected by using the selection arrow and drawing a box around it.



Using Inkscape – Step 3

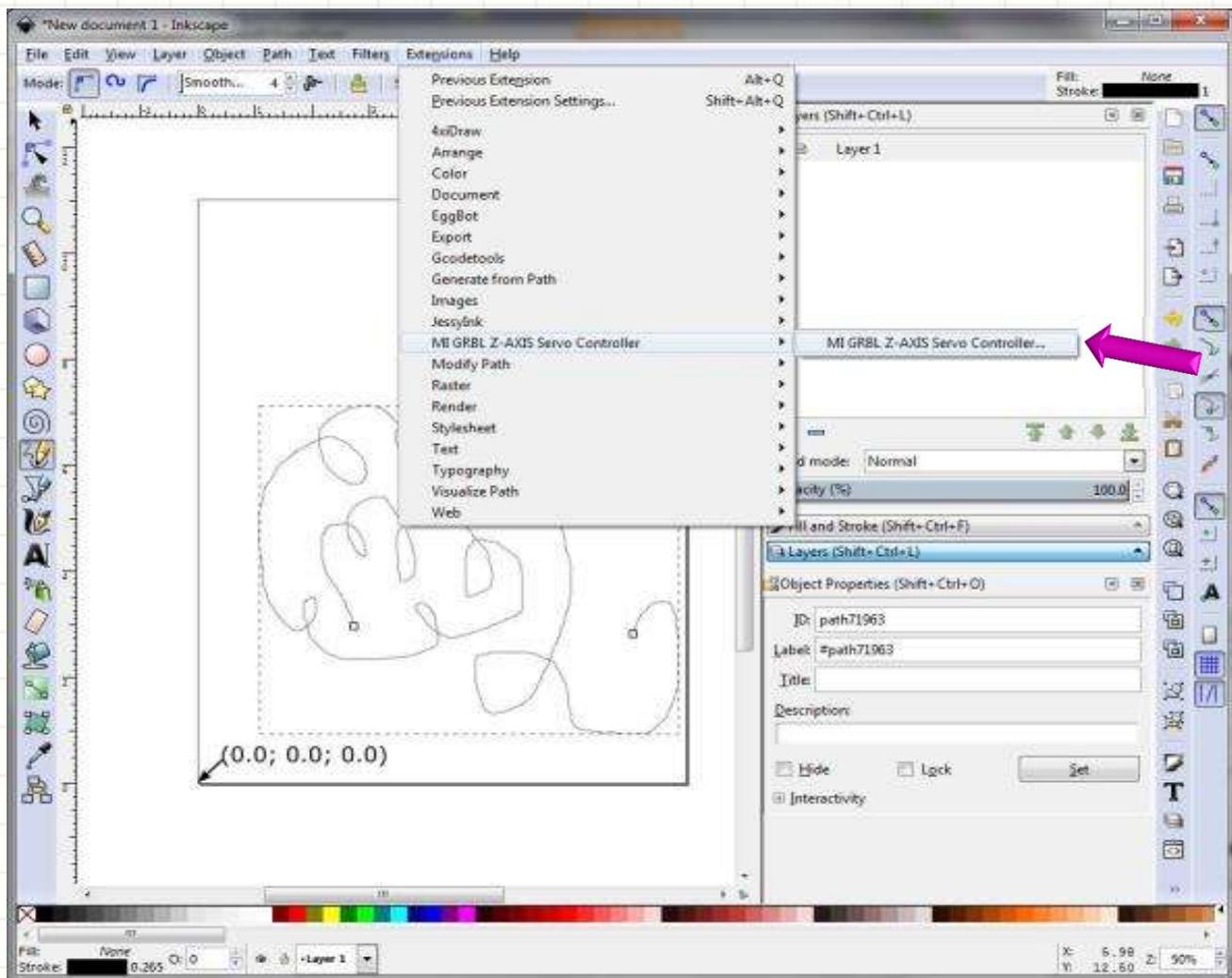
- Select the “Path” pull down and then “Object to Path”

This allows Inkscape to interpret the drawing as line paths



Using Inkscape – Step 4

- Select the pull down “Extensions” and then “MI GRBL Z Axis Servo Controller” and “MI GRBL Z Axis Servo Controller”



Using Inkscape – Step 5

The extension pop up will appear. It is very important that you fill in the fields as shown.

Servo Up M3

Servo Down M5

X axis Speed 3000

Y axis Speed 3000

Angle for Servo 30

Delay 0.1

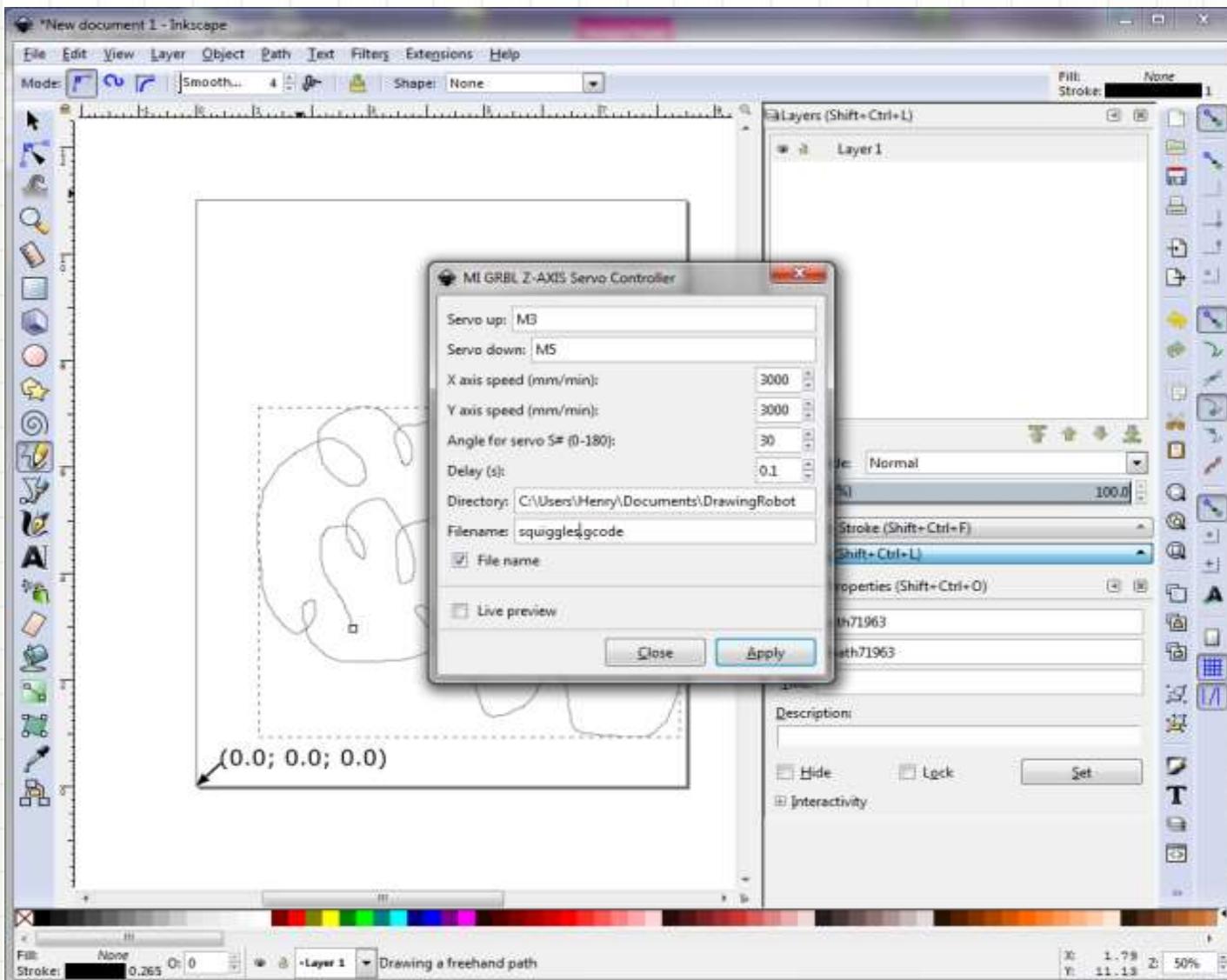
Directory
 <your choice>

Filename
 <your choice>

Filename Box checked

**These fields will be saved
for future use.**

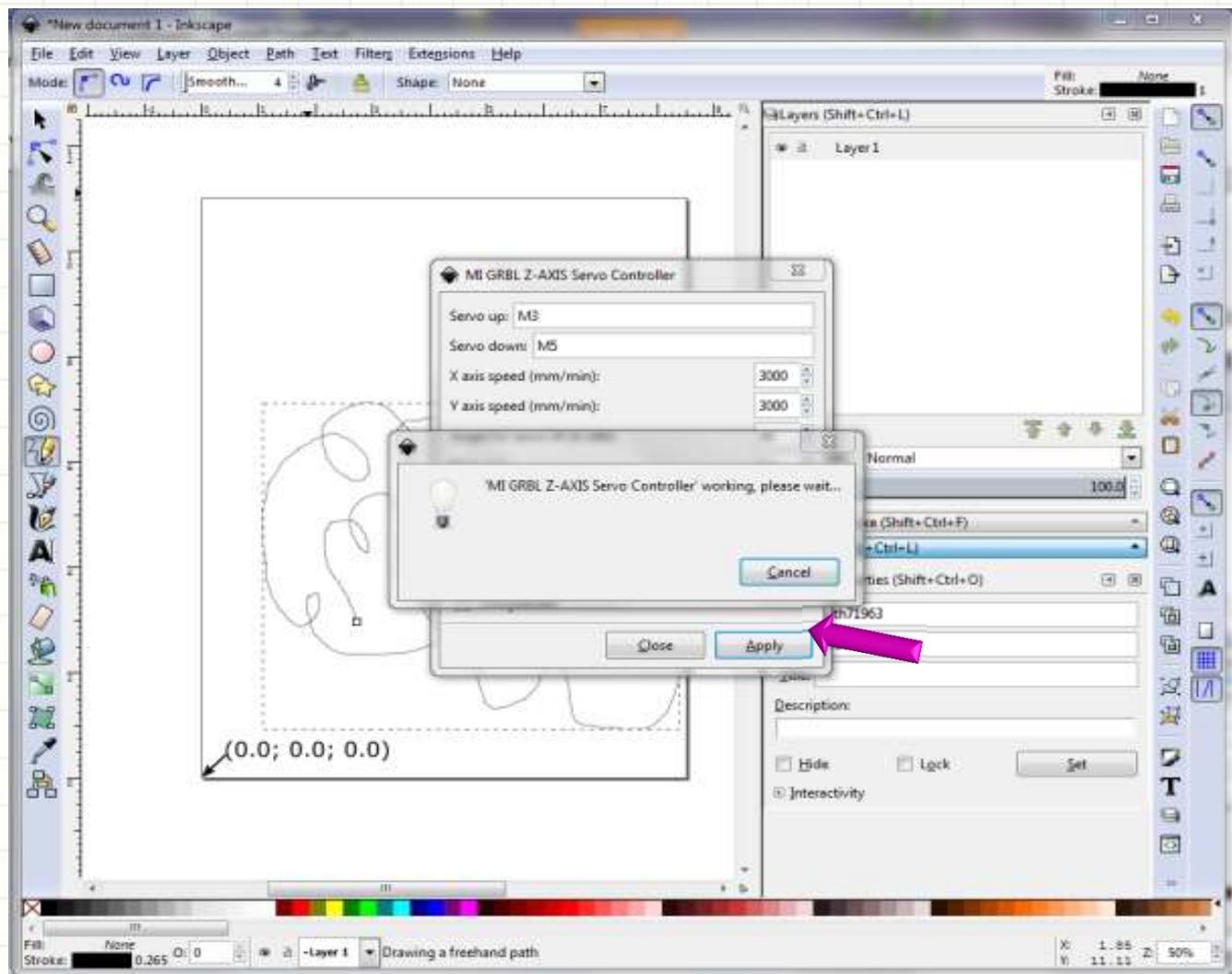
Select “Apply”



Using Inkscape – Step 6

You should see a “Working, please wait” pop up like the one shown while the Extension processes your drawing. When the Pop up disappears, your G-Code will be in the directory that you selected.

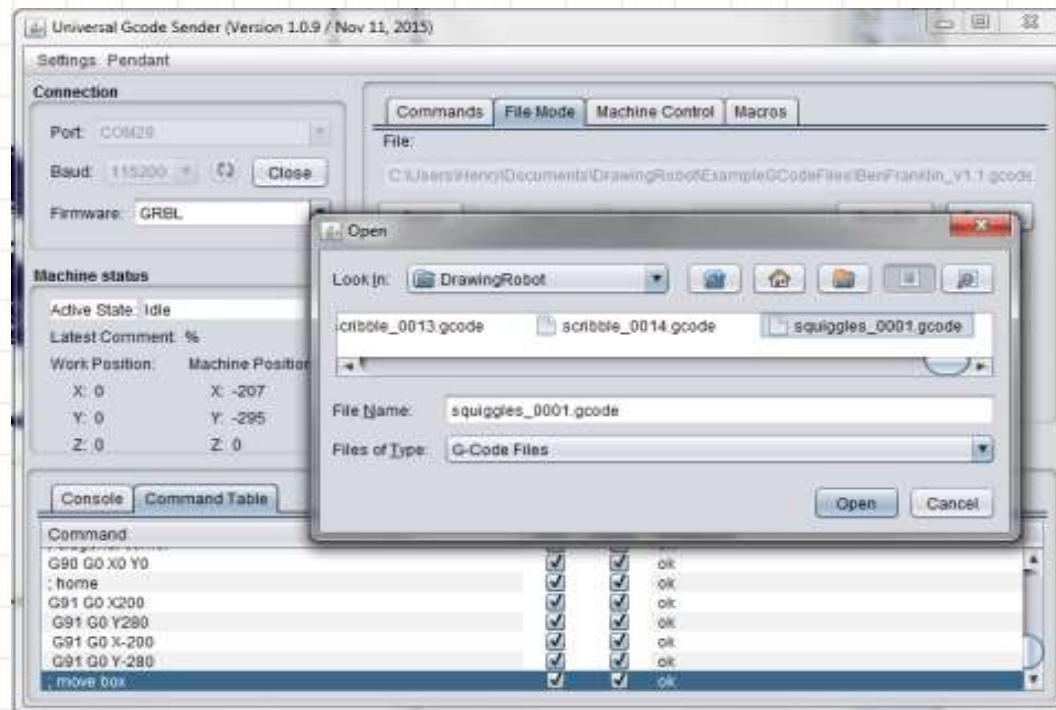
Select “Close”
After the Cancel
Box disappears

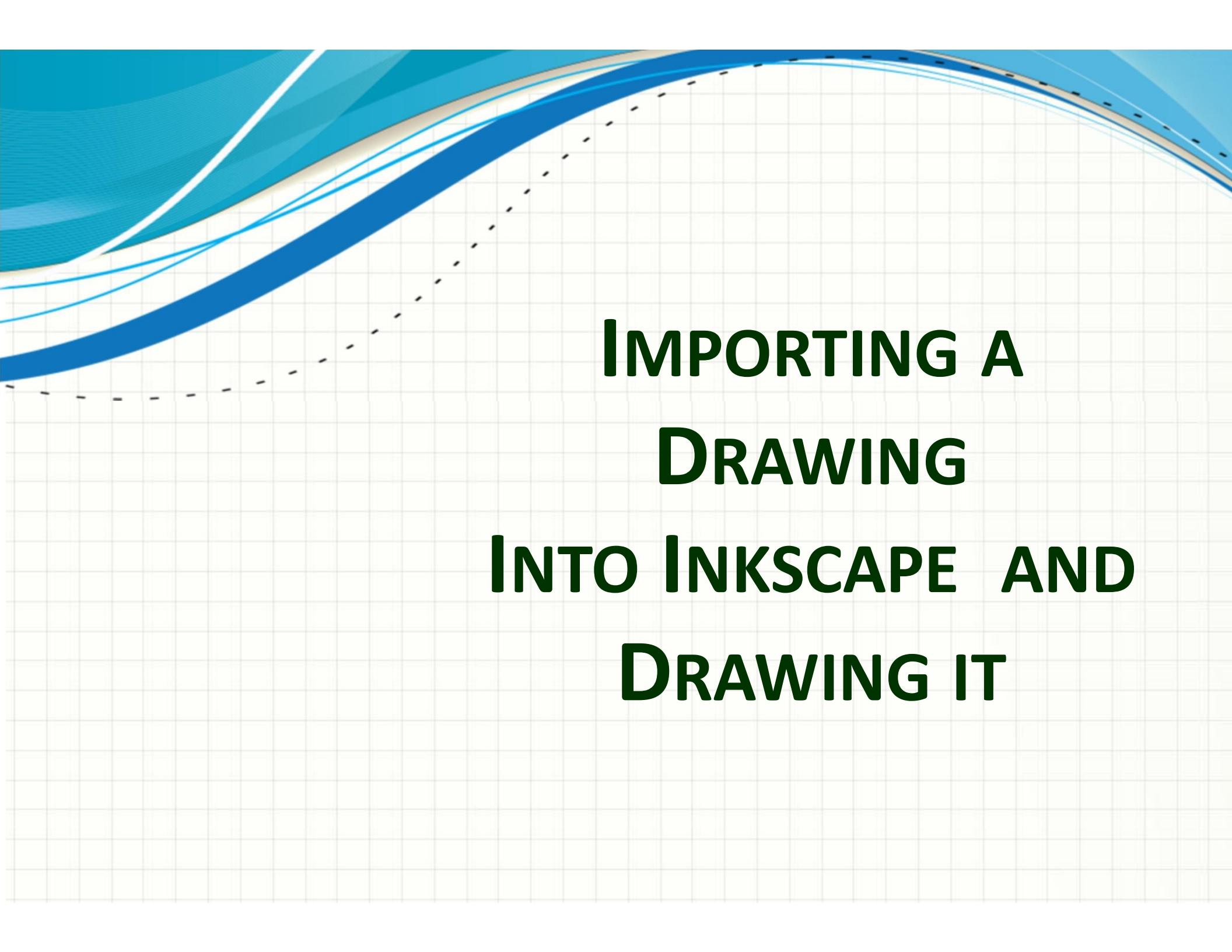


Using Inkscape – Step 7

Now load your file using the Universal G-Code Sender (UGS).

- From the “File Mode” Tab select “Browse”. A pop up will allow you to navigate to your file. Our example filename was “squiggles” and Inkscape added a _0001.gcode to the name. The file is squiggles_0001.gcode. It will appear in the directory you selected.
- Make sure you have your DrawBot set at the zero location and that the pen is ready.
- Select “Send” and your DrawBot should start drawing the file.
- If anything goes wrong, select “Cancel”.
- Otherwise you should get a plot that looks like what you drew.





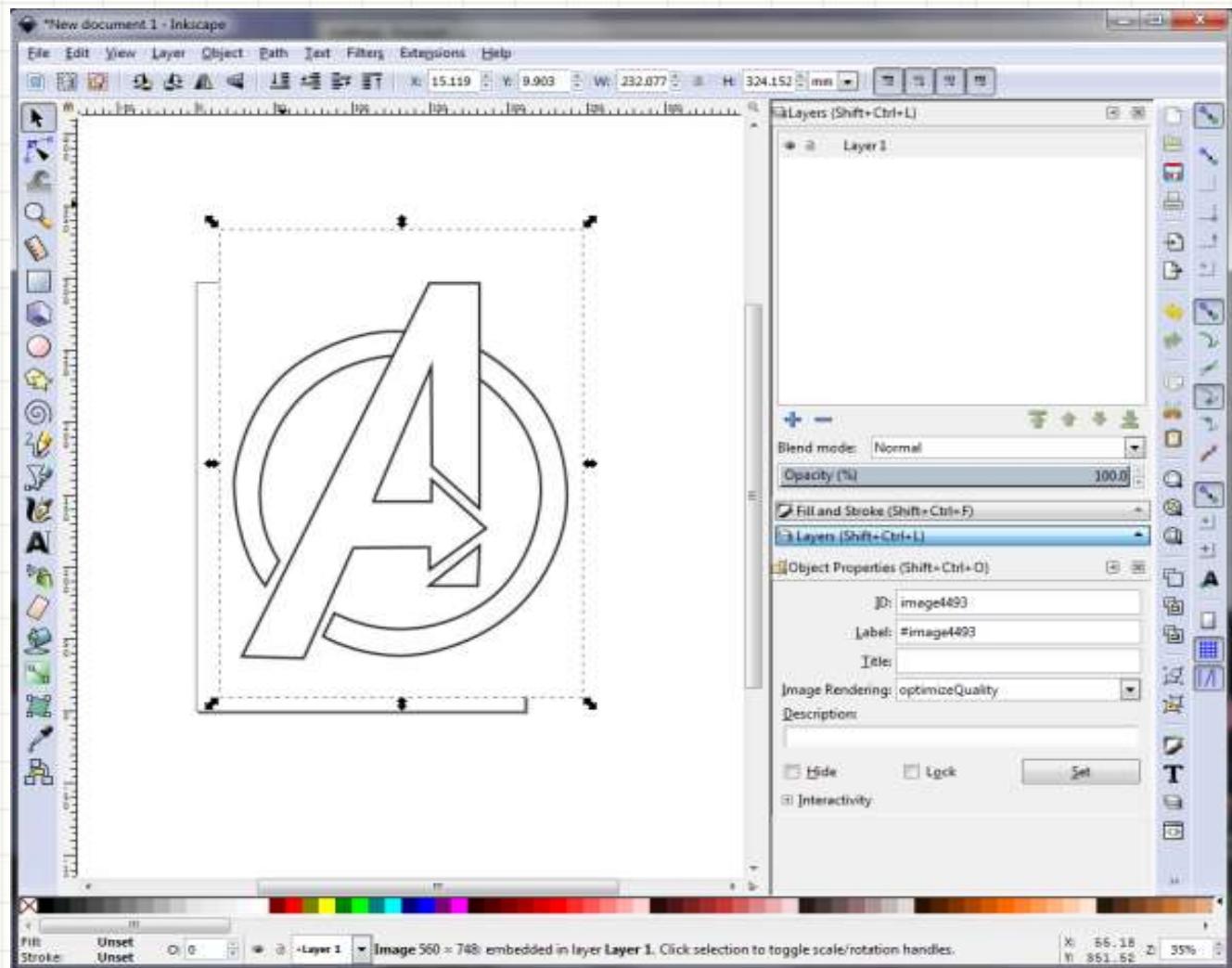
IMPORTING A DRAWING INTO INKSCAPE AND DRAWING IT

Using Inkscape with Imported Graphics –

Step 1

Sometimes you may want to plot something that already exists as a JPG or GIF graphic. The following slides explain how to import a graphic into Inkscape and prepare it to be drawn by your DrawBot.

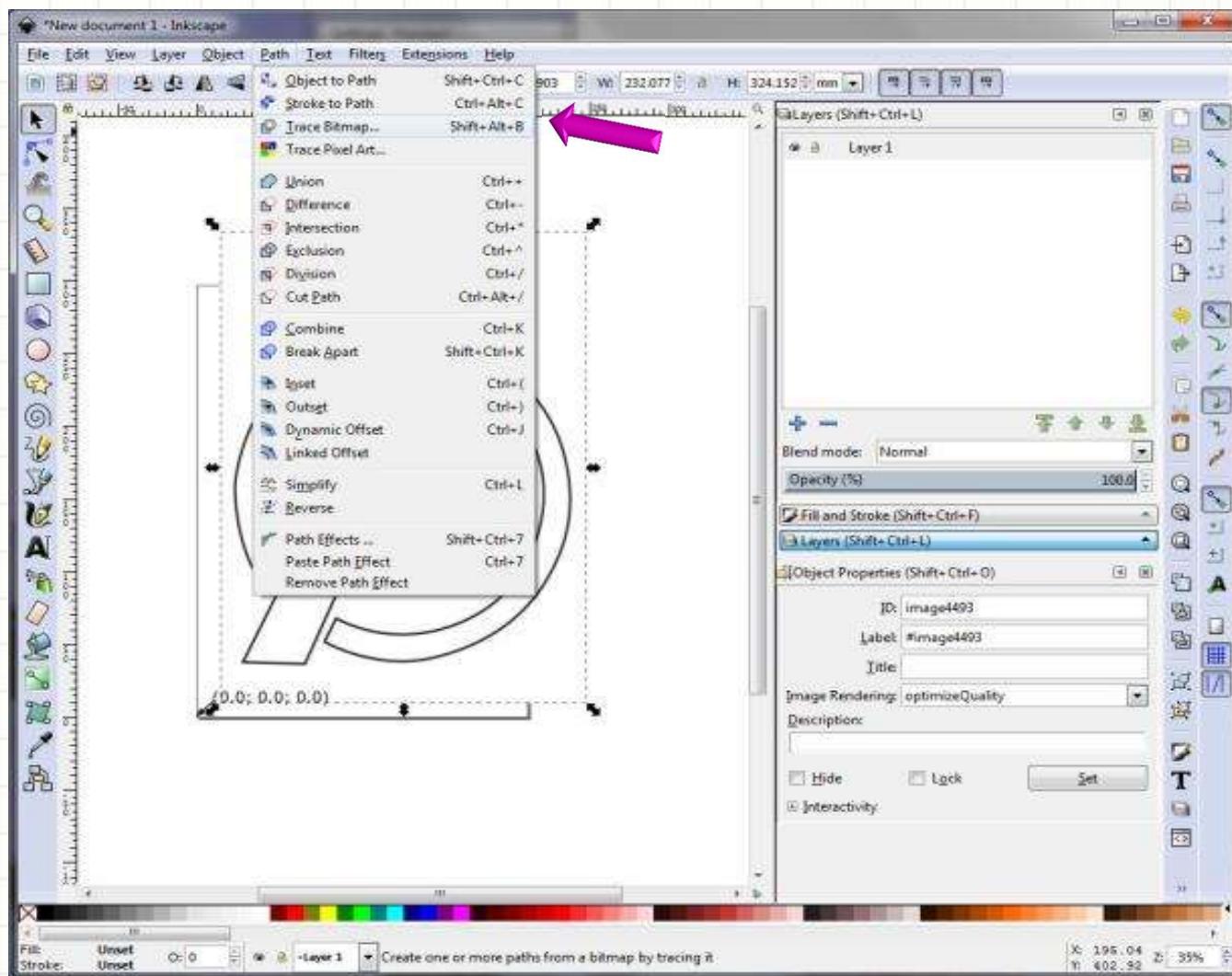
- Open Inkscape.
- Select “File” and then “Import”
- Select the graphic you would like to import.
- Size and move the graphic to the size and location you want.



Using Inkscape with Imported Graphics – Step 2

The current drawing is a bitmap. We need Inkscape to be able to find the edges of the graphic to create lines to draw

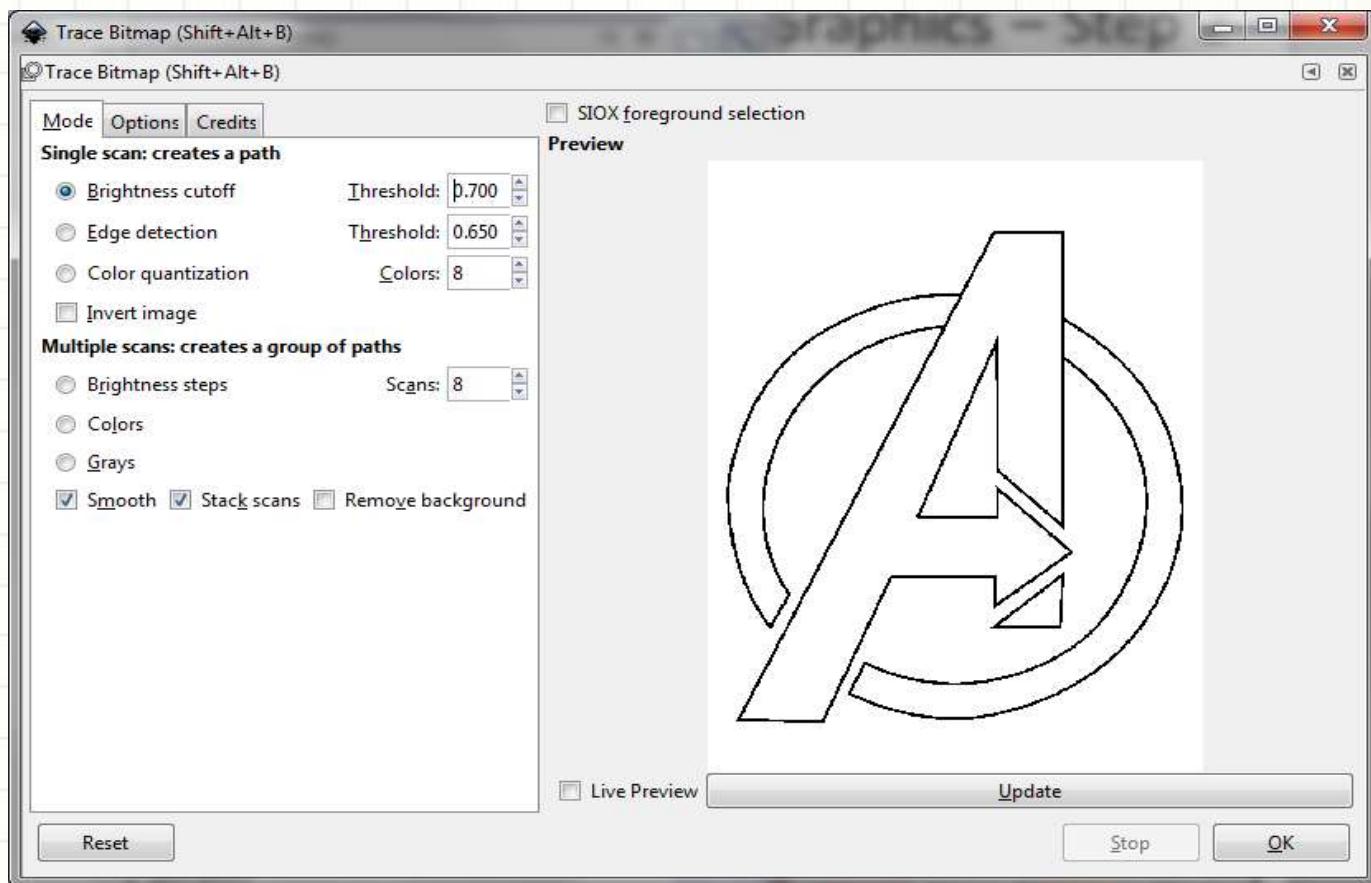
- Select the pull down “Path” and then select “Trace Bitmap”
- A popup will appear which is explained in the next slide.



Using Inkscape with Imported Graphics – Step 3

The “Trace Bitmap” feature allows Inkscape to find the lines in your graphic. I use “Brightness Cutoff” but “Edge Detection” also works. Play with the thresholds until you get a result that looks sharp and has well defined lines.

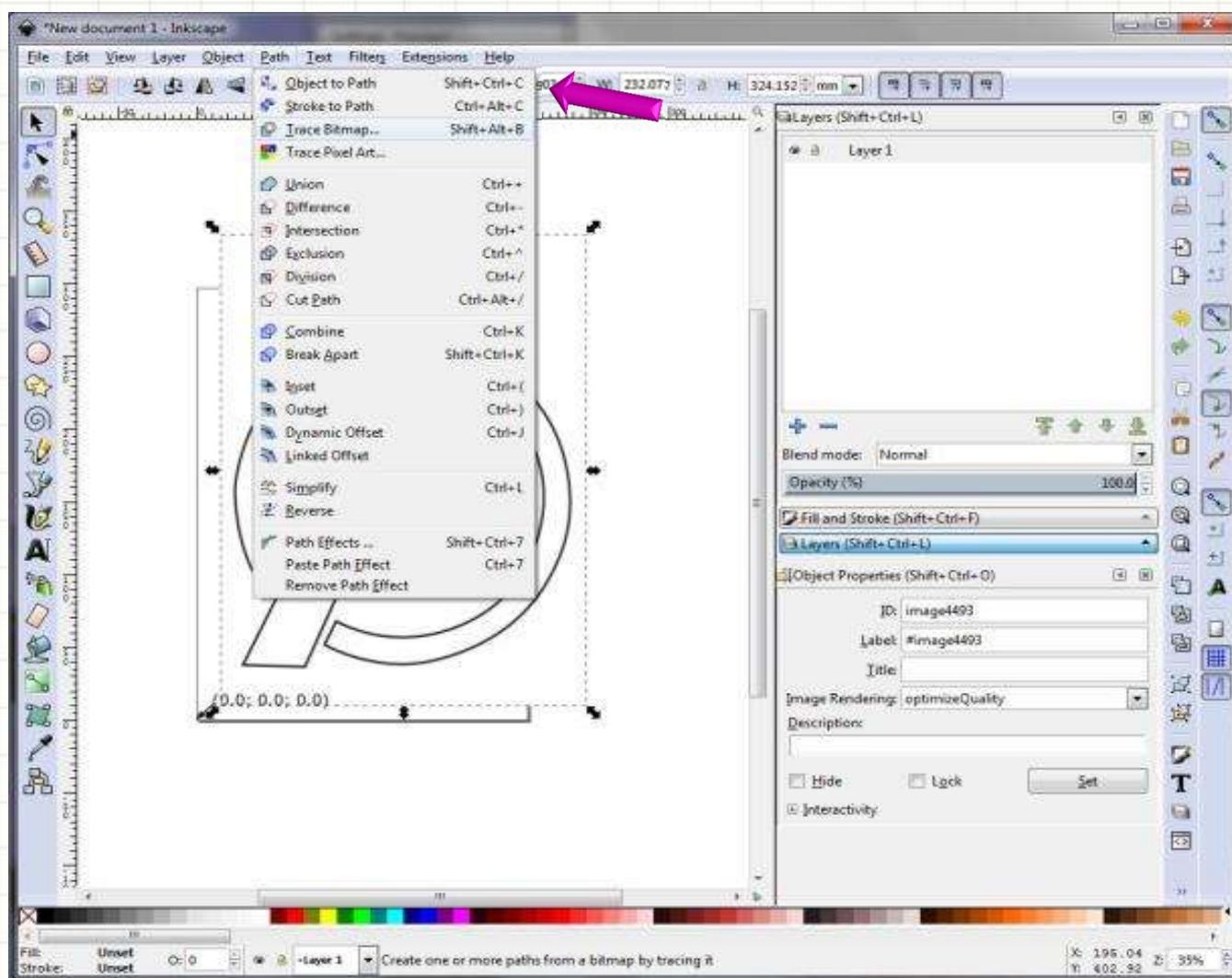
- To see the result as you work, either press “Update” or click the “Live Preview” box
- Once you are satisfied, click OK and then the upper right X to close the window.



Using Inkscape with Imported Graphics – Step 4

Inkscape now knows where the edges are in your graphic. We now want Inkscape to select line paths for drawing our graphic.

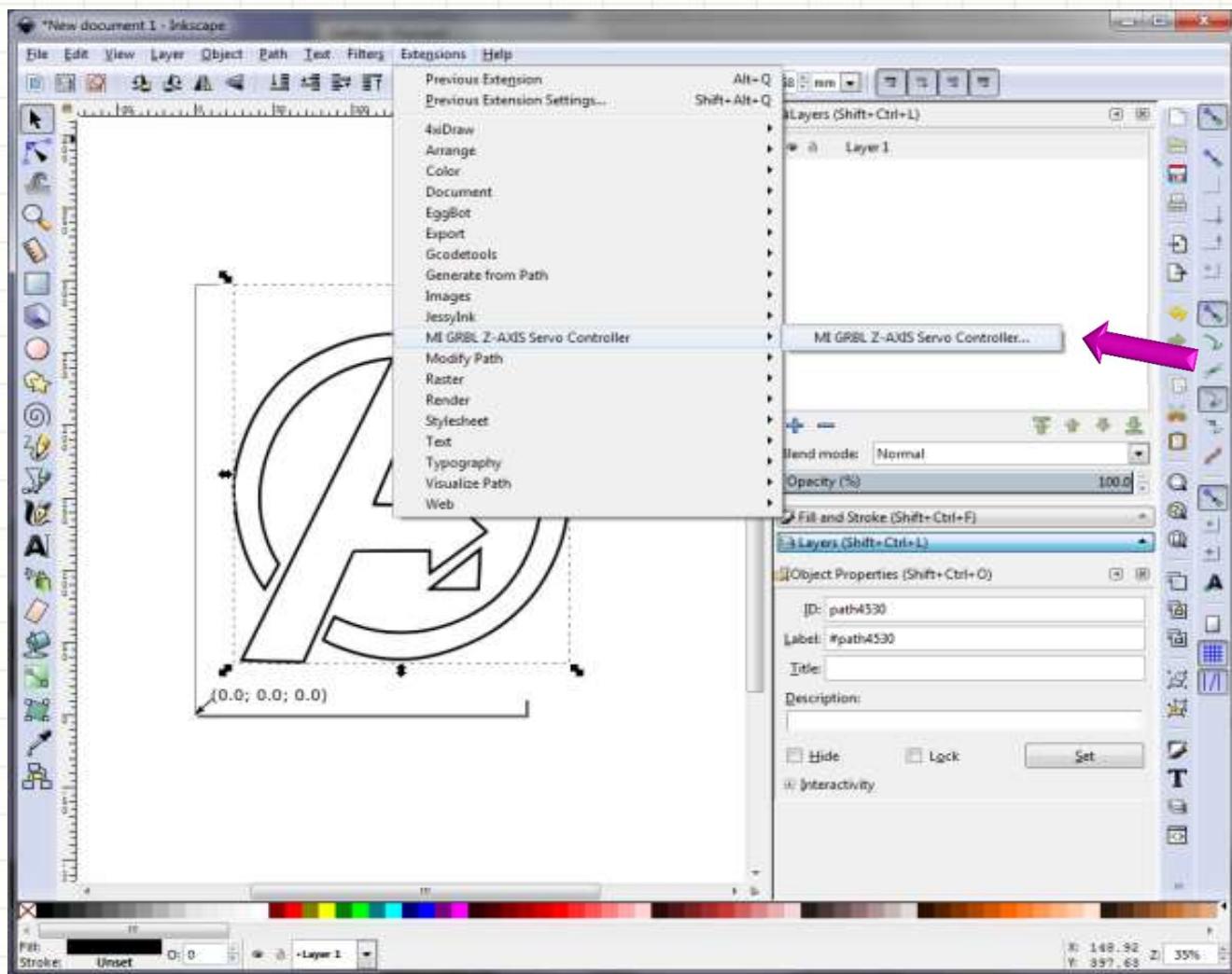
- Select the pull down “Path” and then select “Object to Path”
- There is no pop up on this feature. The action happens behind the scenes.



Using Inkscape with Imported Graphics – Step 5

We now have a drawing where defined edges were turned into line paths we want to draw. We now need to turn the line paths into G-Code which the DrawBot can use to draw the graphic.

- Select “Extensions” and then “MI GRBL...” and “MI GRBL Z Axis Servo Controller”

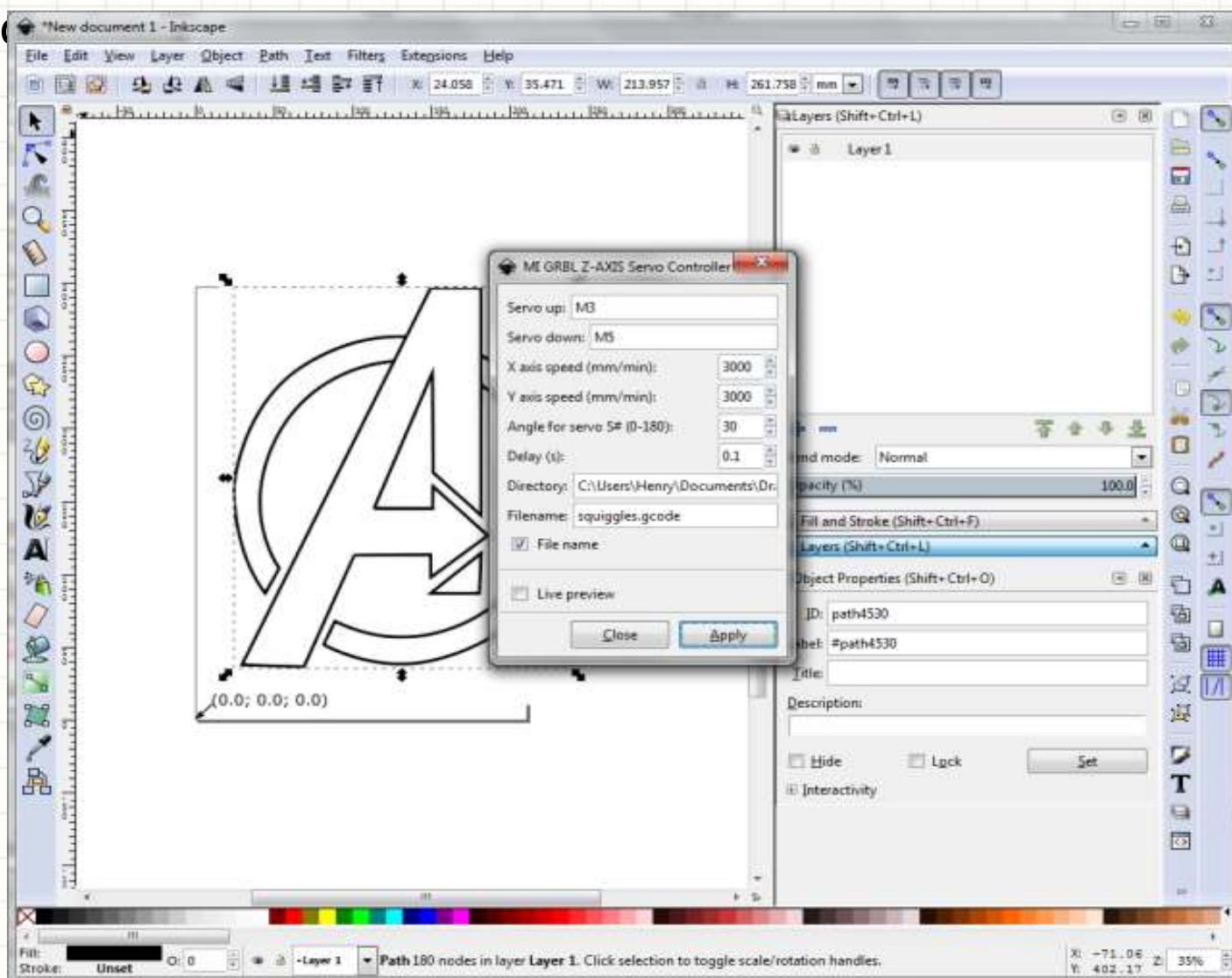


Using Inkscape with Imported Graphics – Step 6

The MI extension pop up will appear. It is very important that the fields are as shown.

Servo Up M3
Servo Down M5
X axis Speed 3000
Y axis Speed 3000
Angle for Servo 30
Delay 0.1
Directory <you choose>
File name checked

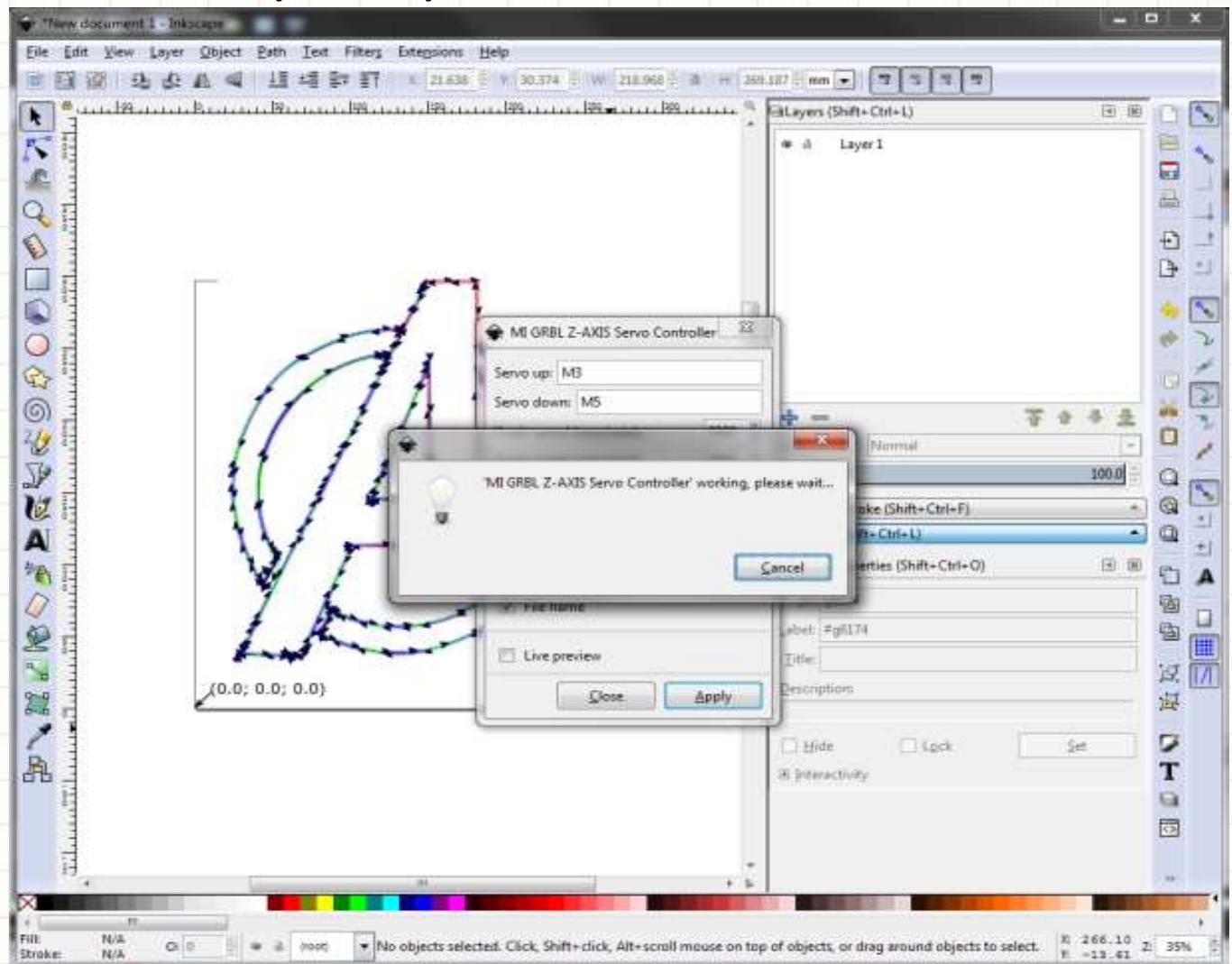
**These fields will be saved for future uses.
Select “Apply”**



Using Inkscape with Imported Graphics – Step 7

You should see a pop up like the one shown while the Extension processes your drawing. When the Pop up disappears, your G-Code will be in the directory that you selected.

Select “Close”
After the Cancel
Box disappears

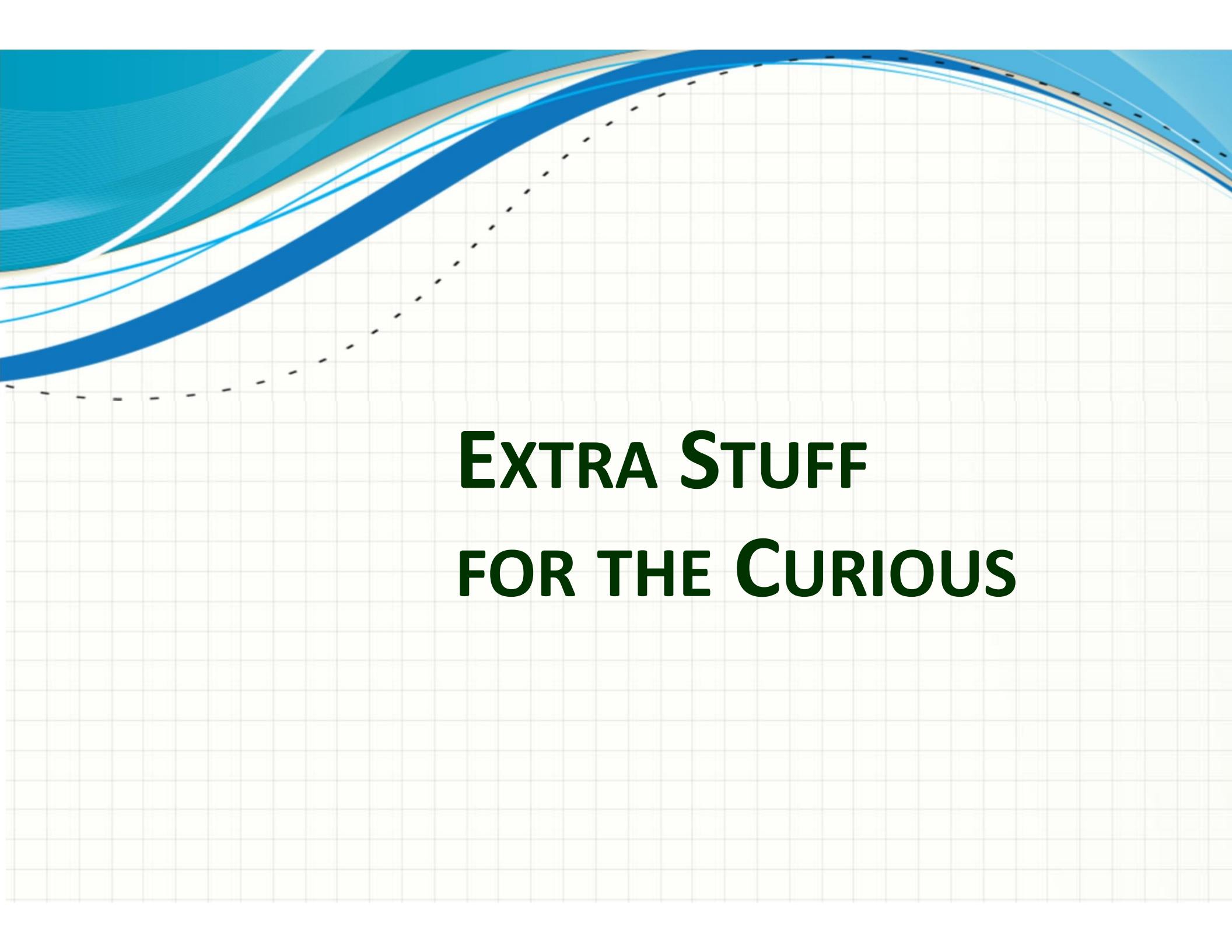


Using Inkscape with Imported Graphics – Step 8

Now load your file using the Universal G-Code Sender (UGS). If you don't remember how to do that, refer back to the previous example.

- Make sure you have your DrawBot set at the zero location and that the pen is ready.
- Select “Send”
- You should get a plot that looks like the imported graphic.





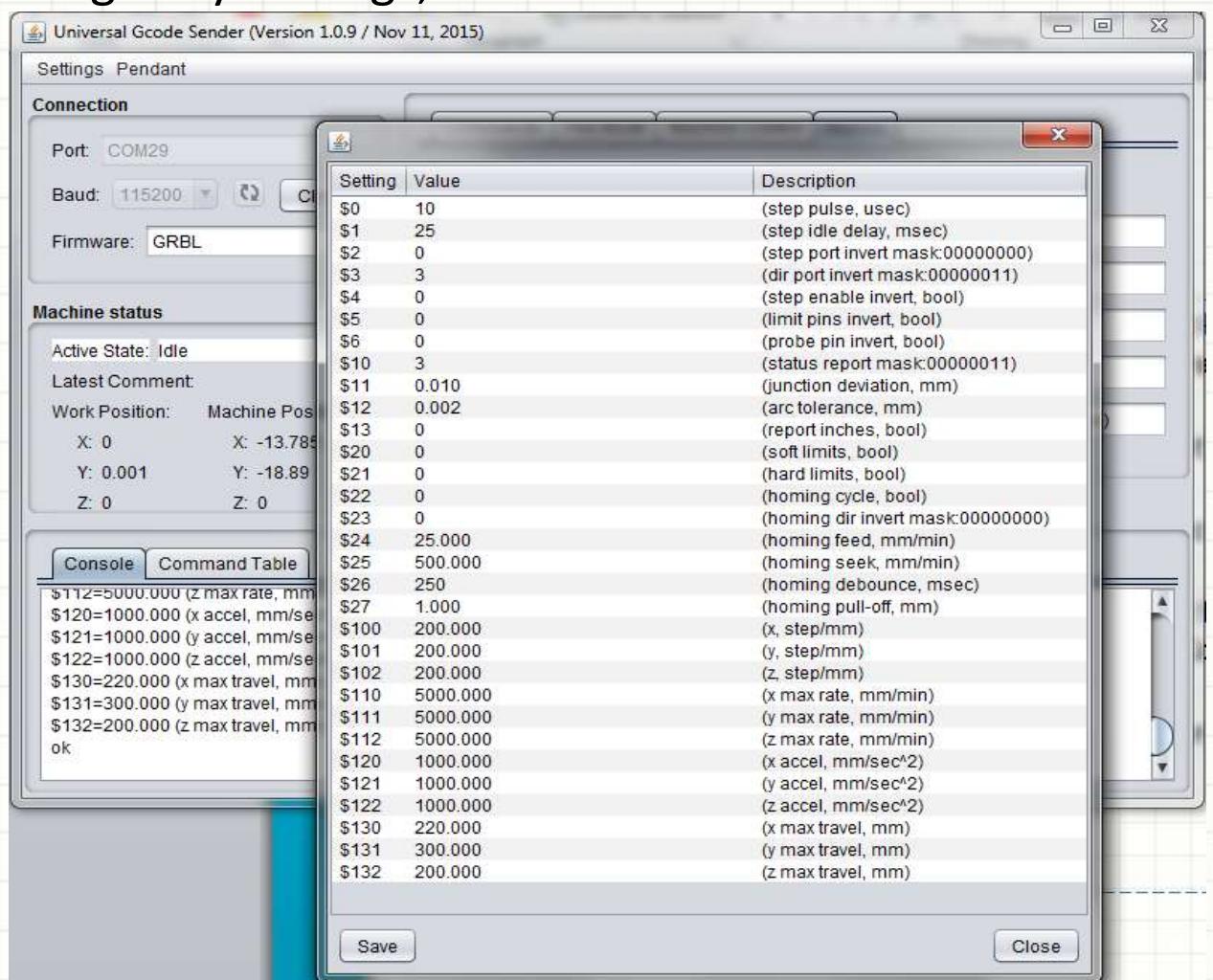
EXTRA STUFF FOR THE CURIOUS

Drawing Robot Tips

- Be sure to use the “Reset to Zero” to let the DrawBot know where you want your origin to be. Erratic behavior can be caused by the DrawBot starting from an unknown origin.
- If you have problems, don’t hesitate to try “Soft Reset” from the UGS or a hard reset which is to power the DrawBot off then back on.
- Make sure your pen touches the paper on the C2 macro and that it lifts clearly off the paper on a C3 macro.
- The UGS “Console” shows the commands being used for each motion. You can learn a little about G-Code commands by looking at the “Console” or “Command Table” windows.
- If you have a tricky problem with a G-Code file, try turning “Verbose” mode on by checking that box.
- If you are having strange problems, check your GRBL parameters as explained on the next slide.

Drawing Robot GRBL Parameters

If you are having problems with your DrawBot, check the GRBL parameters by selecting “Settings”, “Firmware Settings”, and “GRBL”. Check your settings against the ones shown. If you change any settings, select “Save” before “Close”.



Software for the Drawing Robot

- Inkscape
<https://inkscape.org/en/>
- Inkscape MI Extension for creating G-Code
<http://www.mediafire.com/file/ae0wquqornzc3o2/MI+Inkscape+Extension.zip>
- Universal G-Code Sender Ver 1.0.9
https://winder.github.io/ugs_website/download/
- USB Drivers if needed
<https://www.arduino.cc/en/Guide/DriverInstallation>

For those that want to tinker with the DrawBot firmware

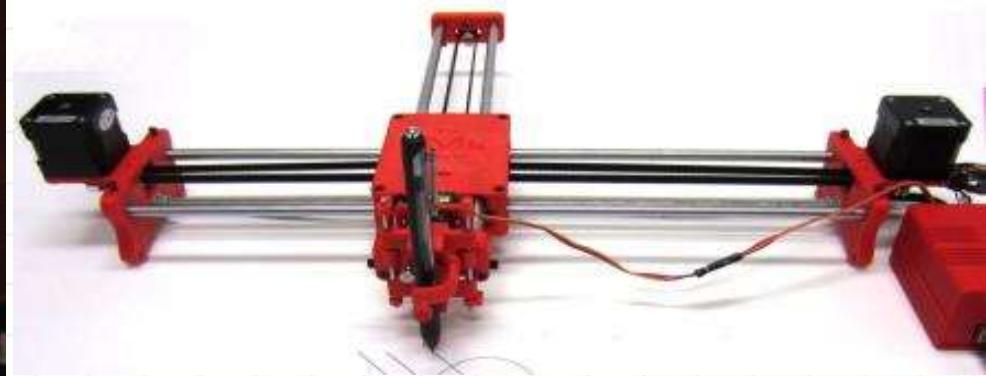
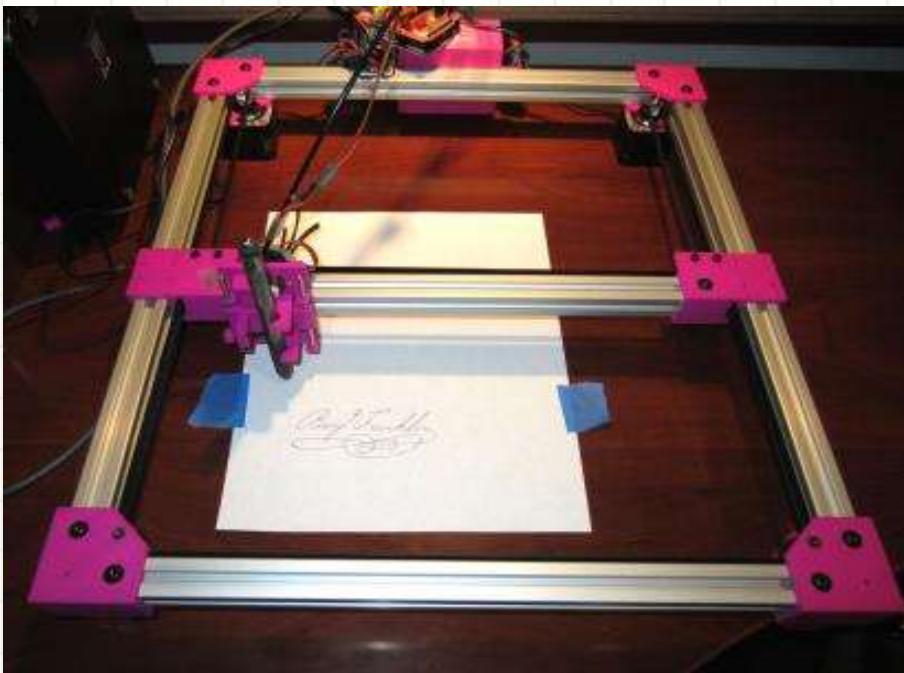
- GRBL – Arduino firmware that makes the DrawBot go.
<https://github.com/robottini/grbl-servo>
- Arduino Development Software
<https://www.arduino.cc/en/Guide/HomePage>

Drawing Robot Thingiverse Pages

Everything you need to build a drawing robot from scratch.

2020 DrawBot

<https://www.thingiverse.com/thing:2374238>



Drawing Robot
<https://www.thingiverse.com/thing:2349232>