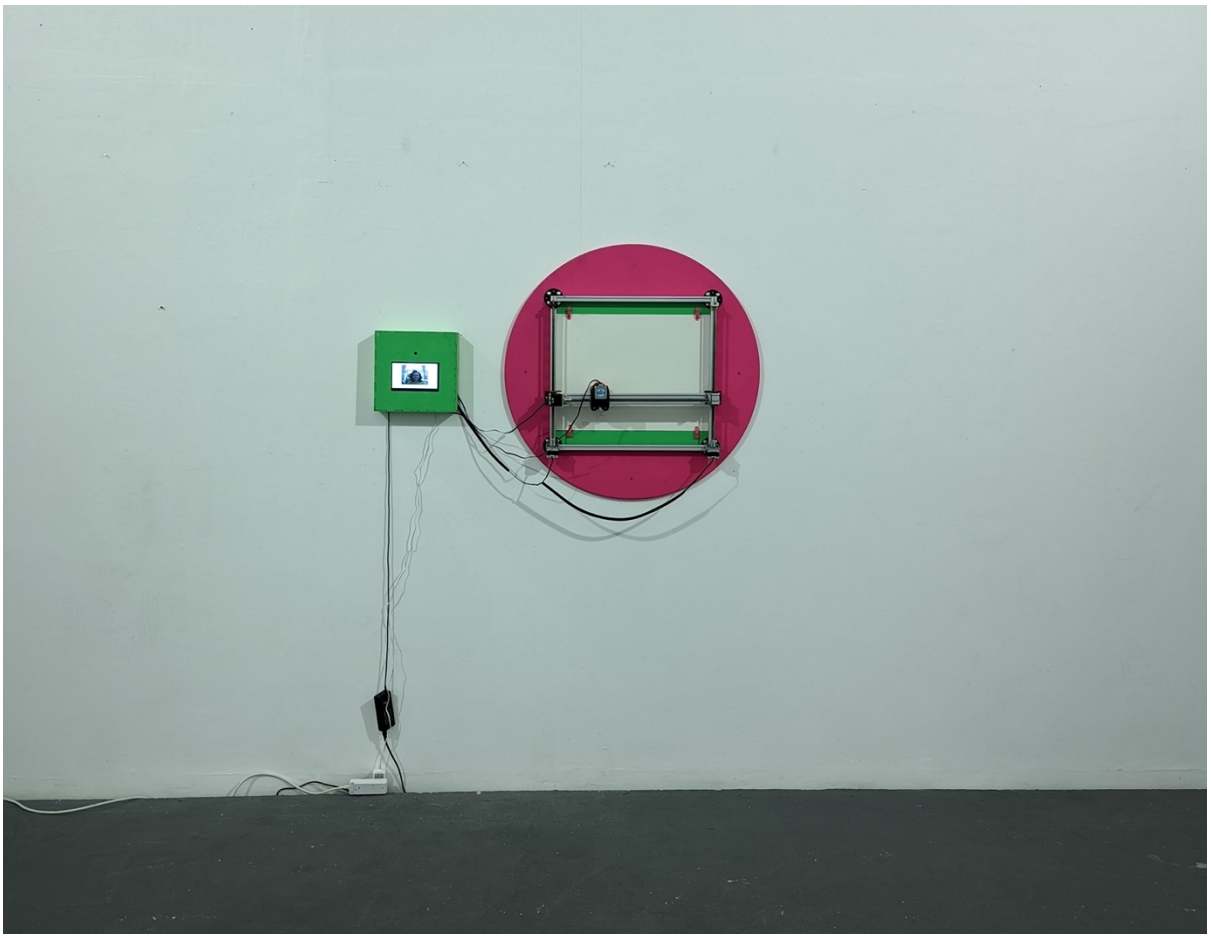


Machine Drawing: Space and Time

Robin Leverton

https://vimeo.com/693756592

Introduction

Space and Time is an installation which utilises a custom-made pen-plotter to continuously draw the space it faces. The work continues my interests in systemic collapse and de-centralised responsibility, whereby through intended interaction with the system, the participant is partially responsible for its ultimate failure. In respect to *Machine Drawing: Space and Time* this is represented by the continuous redrawing over the same page; A space's architecture may be neatly re-constructed, but its living occupants create a dynamic subject that changes with each frame taken by the machine's camera. As such, over the exhibited time, the drawing will lose structure as details begin to confuse, with the eventual case being a single mass where human activity has occurred and no individual pose being discernible.

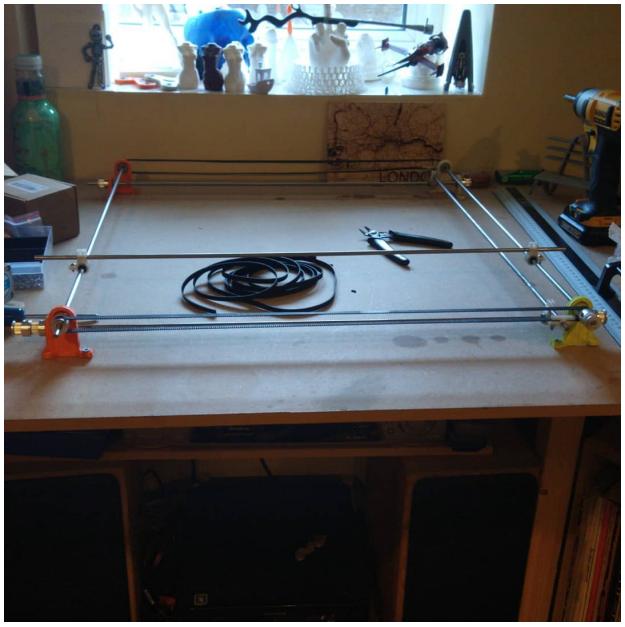
Inspiration

Inspiration for this method of work comes from the Auto-Destructive Art movement and particularly the work of Jean Tinguely and performances such as *Homage to New York, 1960*.(Tinguely 1960). The Auto-Destructive Art (ADA) movement was a follow-on from Dadaism which grew out of the First World War. ADA artists sought to engage with the aesthetic of industrialised life and mechanised brutality and the introduction and proliferation of nuclear weapons. The movement has always been political, and any form of creation that exhibits its own destruction can be seen as metaphor to any number of crises facing the modern world. Unsustainable carbonisation, the use of fossil fuels and global warming, the creation and waste of plastics, deforestation, the concentration of wealth and widening inequality are all limbs of the Uroboros that is neoliberal capitalism. But there is a further point, pertinent to the dumb machines of Tinguely's work and the 'smart' computation within the work I present: That being a critique of production, intelligence and labour. The machine may be now able to see and make decisions, but it ultimately is executing a task with no moral judgement of its outcome, be that the wiping out of a beautiful drawing, its own mechanical destruction or even the destruction caused by autonomous drone strike.

The Build

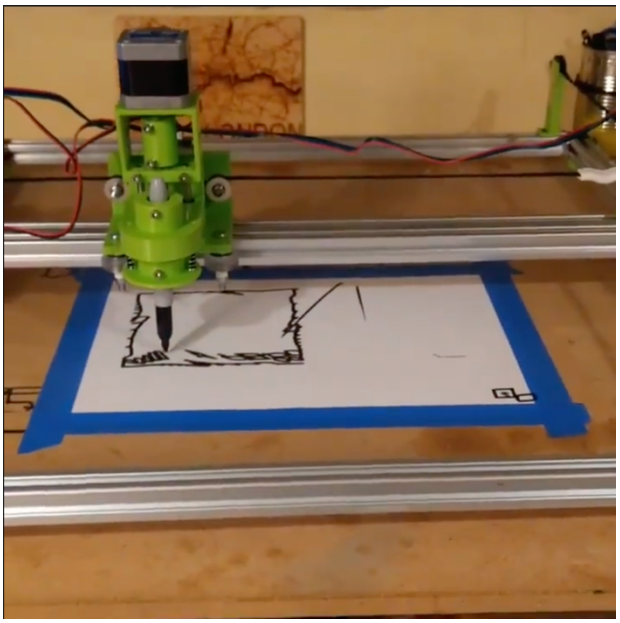
About 3 years ago, I had the idea that I wanted to build a drawing machine, an XY plotter as they're called. For the first two years I mulled it around in my head, thinking what I wanted to use it for, upon realising my idea of continuous drawing over itself, I built the first version. I at first looked at the possibility of using a servo to lift the arm, but I wanted to have more force pushing the pen into the paper in order to have the ability to run it on a wall.

I knew that the frame should be square to act as a picture frame while the drawing was being produced. The first design used 6mm steel rod for this frame, but there was too much flex for the heavy pen holder, so this was soon replaced by aluminium extrusion. The X and Y here are driven by belts and the Z axis is a 5mm screw.

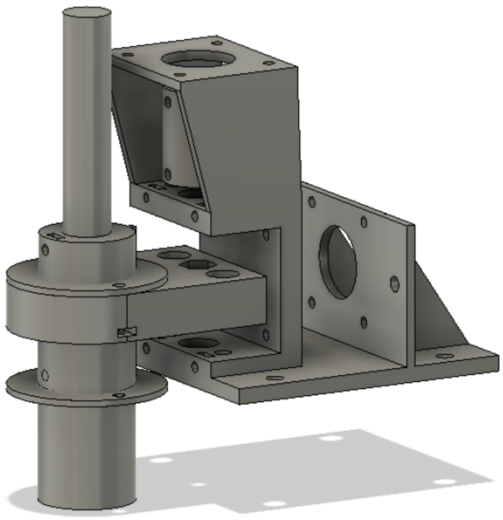


The first design attempt

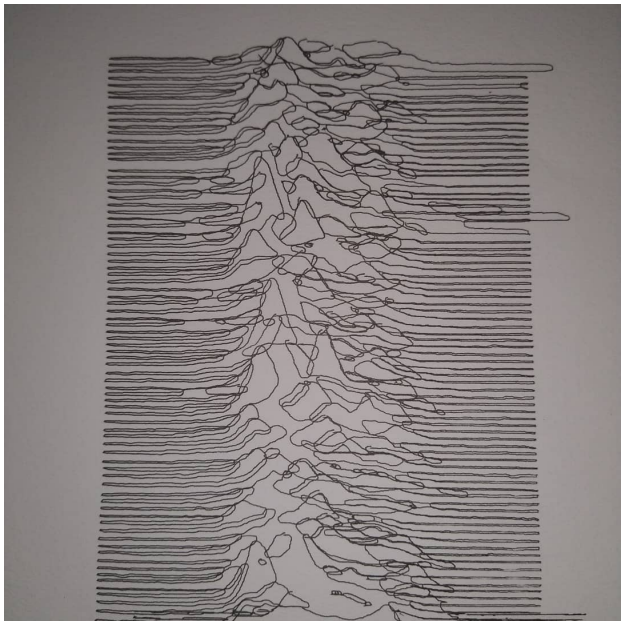
The first drawing

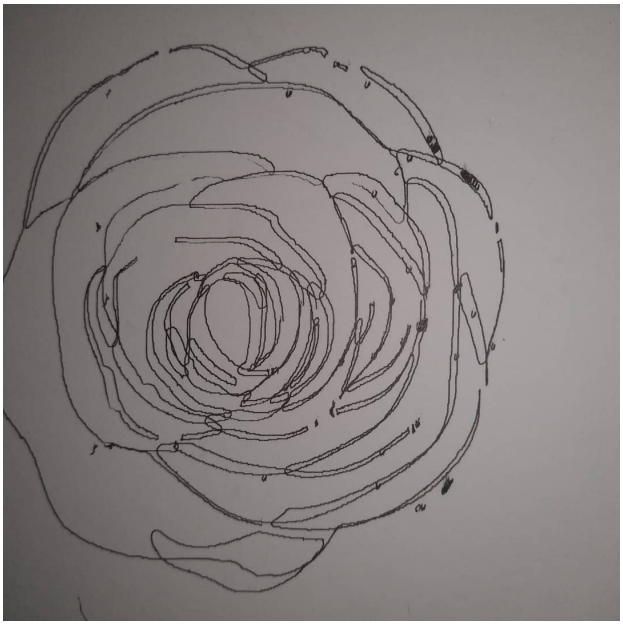


V2.0



V2.0 Pen holder / X carriage

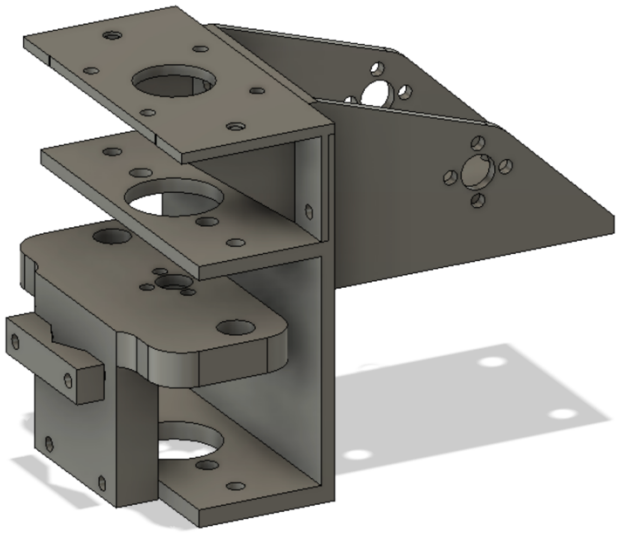




V2.0 Test Drawing Unknown Pleasures album cover

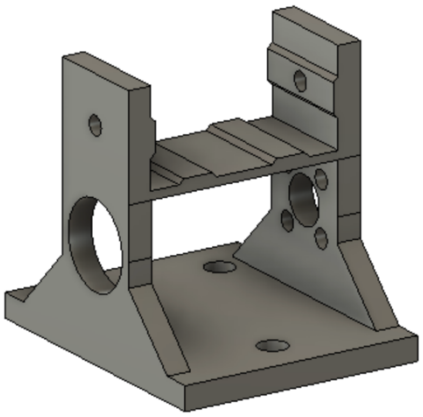
Although I was getting decent drawing results, they were liable to wabbling and inaccuracies, which I wasn't happy with. I also had trouble getting good tension on the belts, so I decided to change them for lead screws and replace the 5mm sawn-off screw with a proper lead screw for the Z axis.

This quest for accuracy, reduced wobble, and incorporation of lead screws entailed plenty of iterations of redesigns to all aspects of the machine. To keep the Y axis in line, I decided to steal the twin stepper motors from a 3D printer that run in parallel.

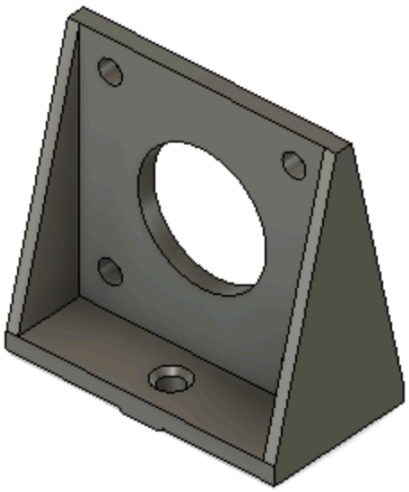


Graveyard of 3D printed parts

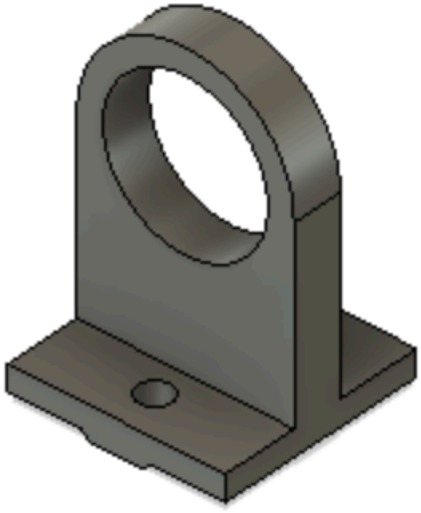
V3.0 X and Z carriage Fusion 360 Model



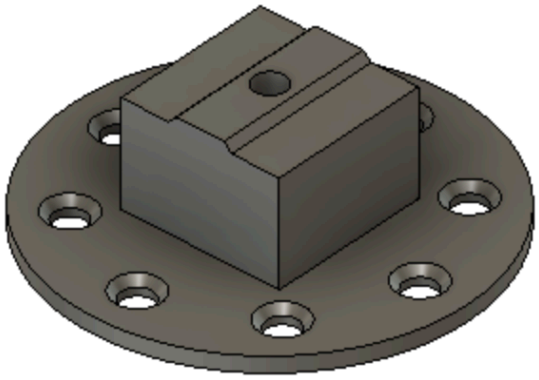
V3.0 Y carriage Fusion 360 Model



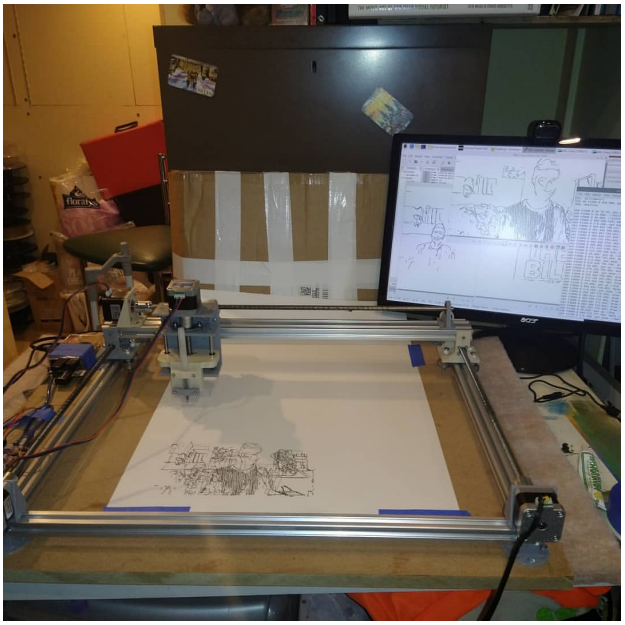
V3.0 Y Stepper Motor Holder Fusion 360 Model



V3.0 Y Lead Screw Bearing Holder Fusion 360 Model

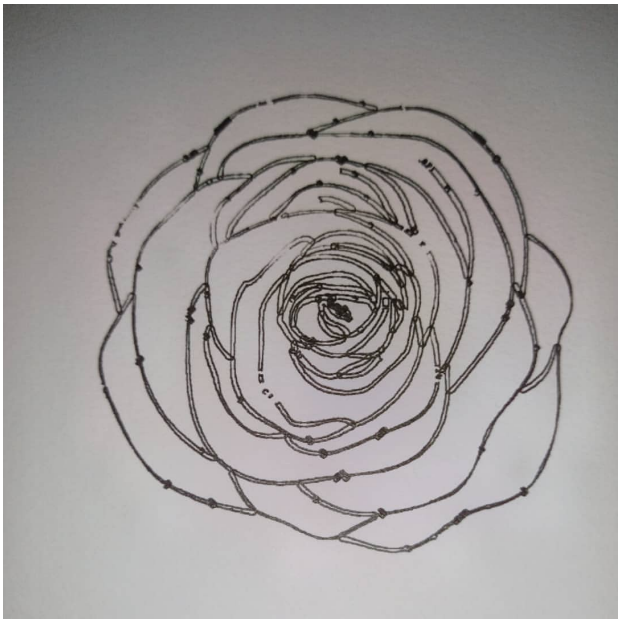


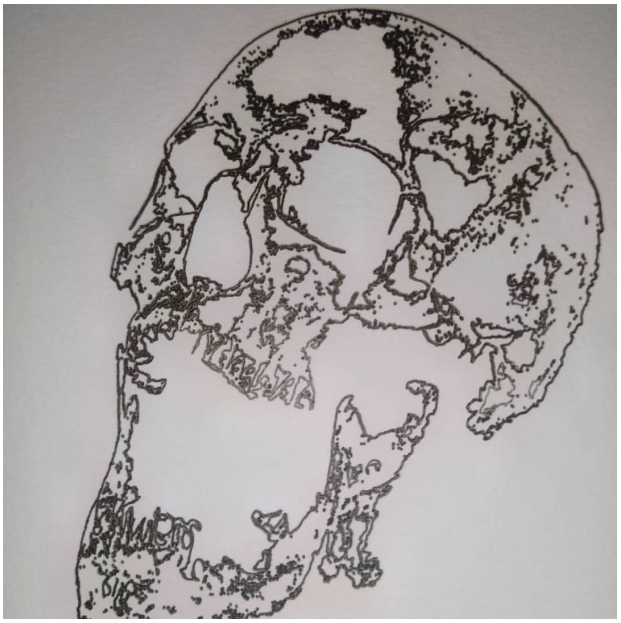
V3.0 Foot Fusion 360 Model



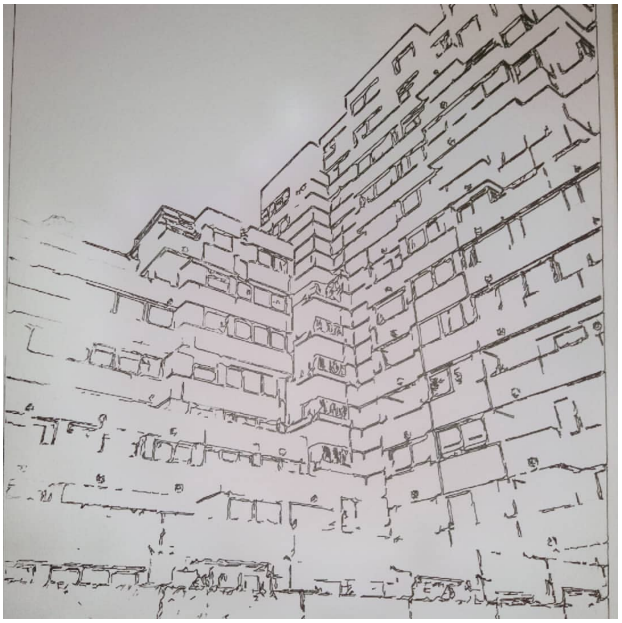
V3.0 Machine

Once this was all together, I was able to get great drawings, though it couldn't run very fast and was incredibly loud due to the A4988 stepper motor drivers running on the grbl shield attached to the Arduino. It was also at this point where I started running the system off a Raspberry Pi to be a fully self-contained unit.



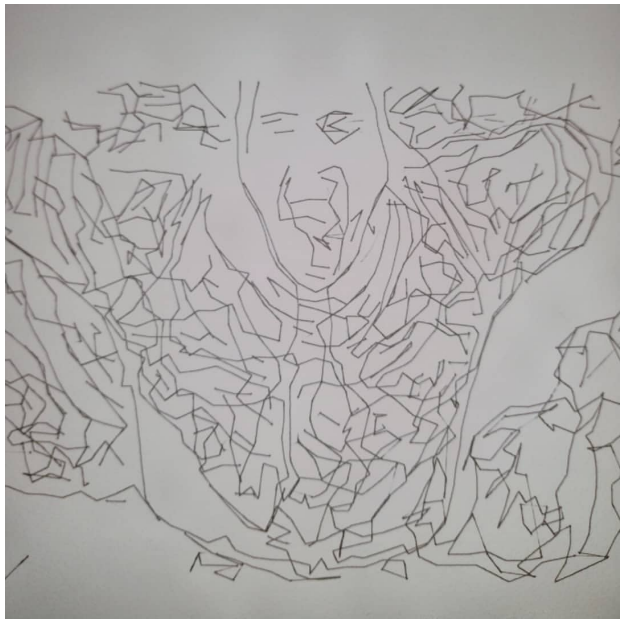


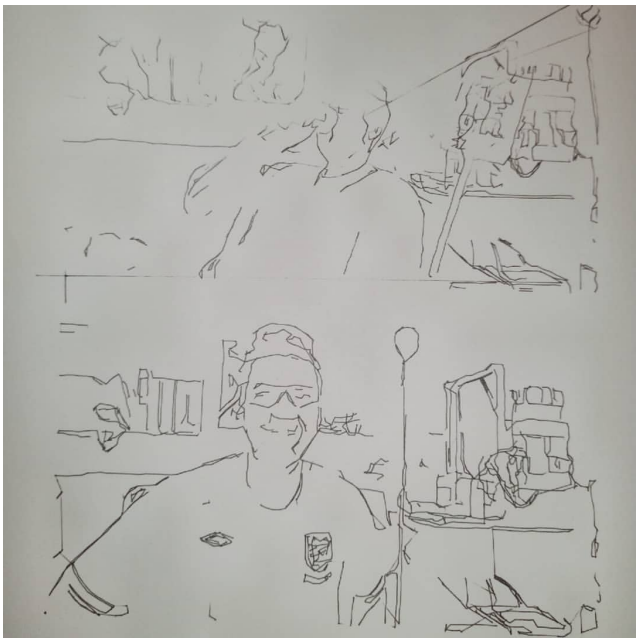
V3.0 Flower and skull test drawings



At this point my focus switched to the software and I was able to patch together several python scripts found on github to achieve what I wanted: A workflow that took a photo using a webcam, turned it into a line drawing (via OpenCV into SVG) and then turn that into GCODE.

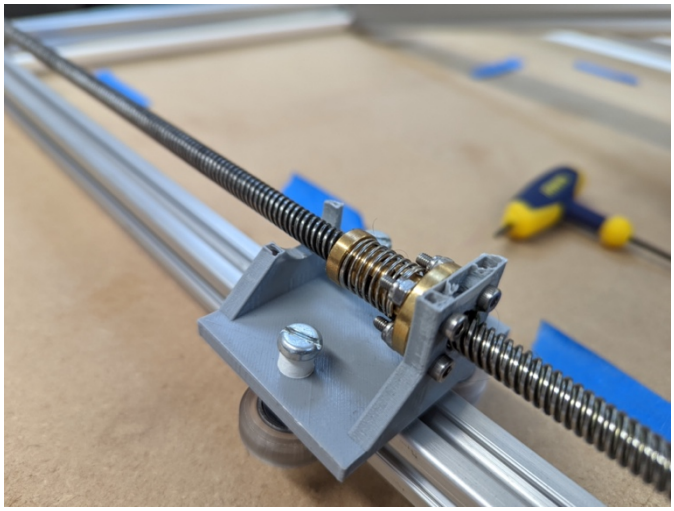
My initial efforts with the webcam I had were underwhelming (see right), so I invested in the Raspberry Pi PiCamera module which created much better results (see below)

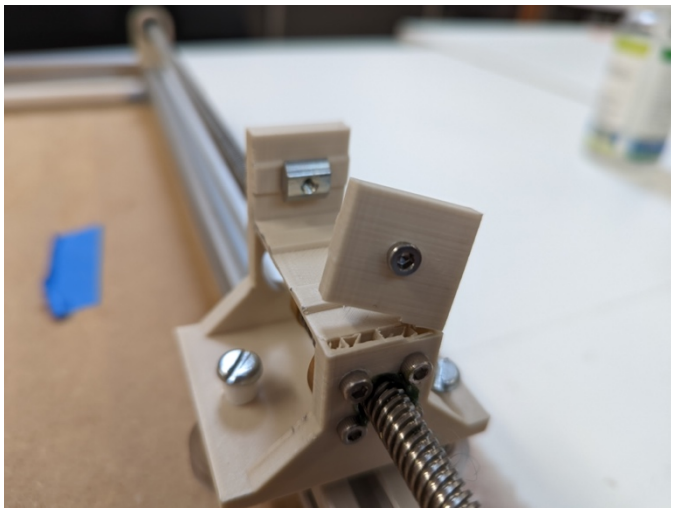


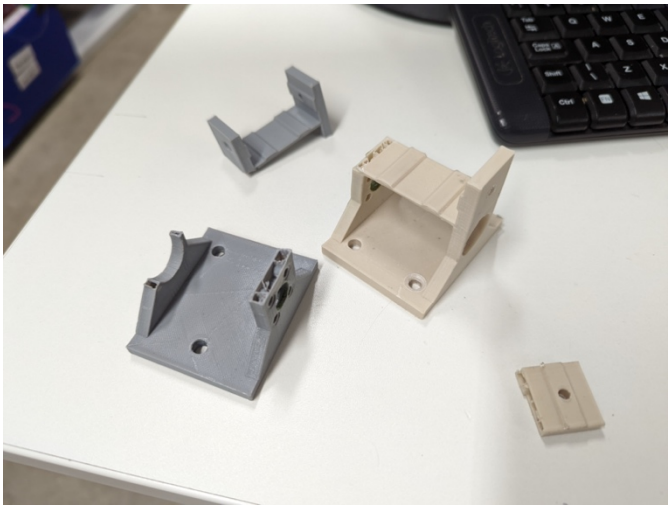


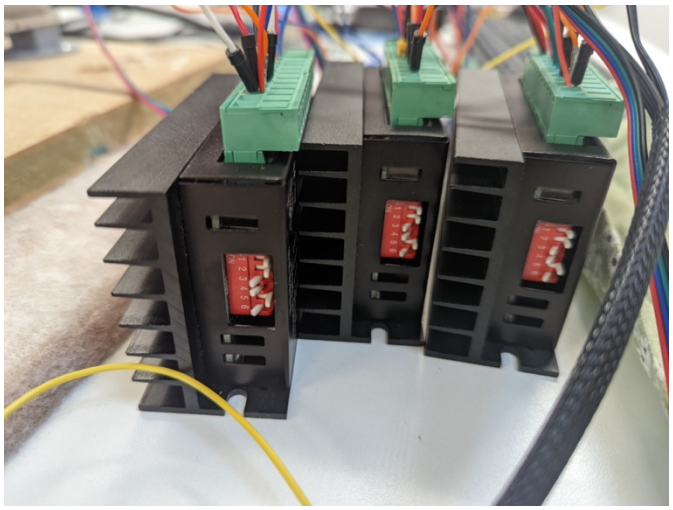
This is where V3.0 took me, with the repeated drawing being trialled here with 3 drawings. The system was still very loud and slow and the code inefficient and patchworked. It is after this point I began working on it at Goldsmiths.

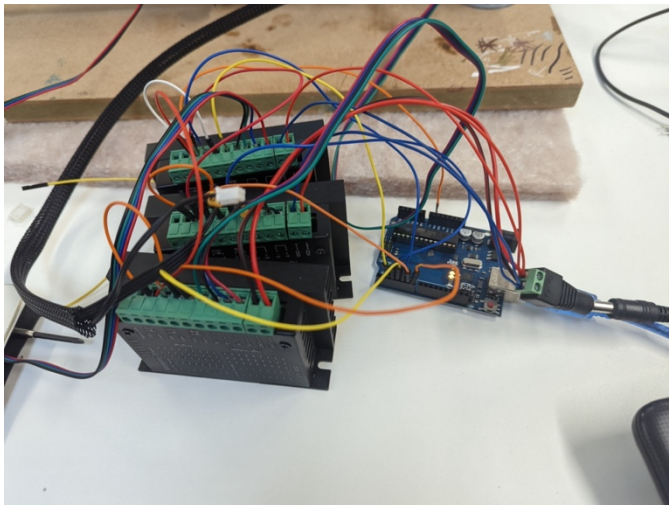
I brought the plotter in to the Hatchlab and inevitably some pieces broke, namely both Y carriages and one of the bearing holders, but all easily re-printable.





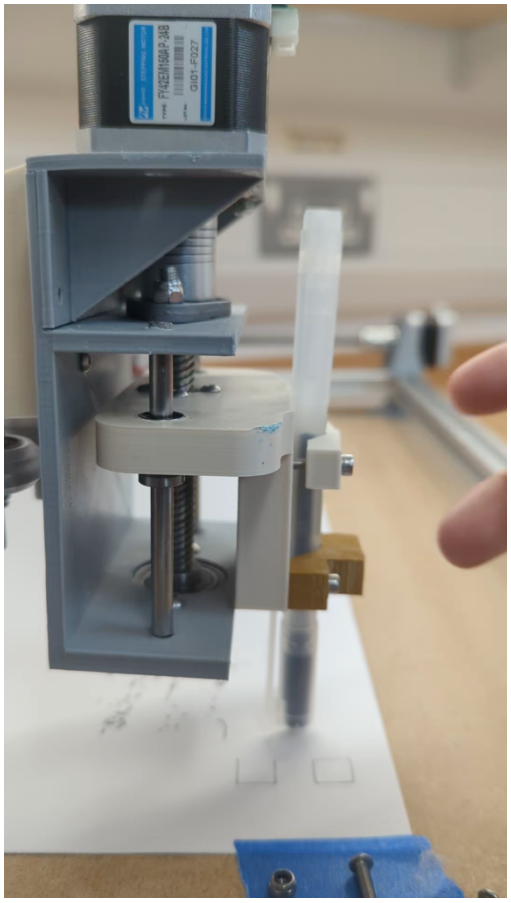


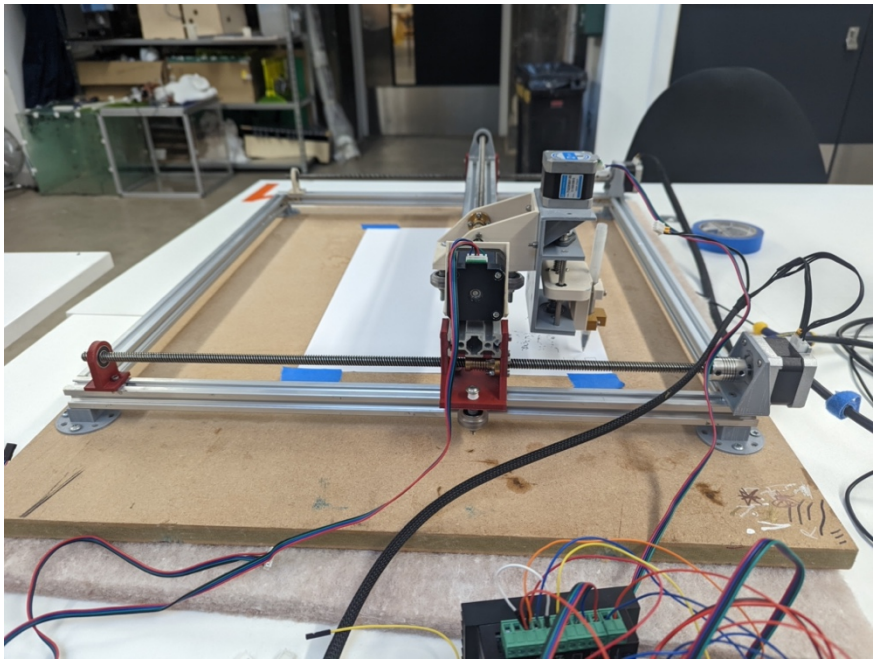




Switching from the GRBL shield to TB6600 stepper motor drivers made the whole system much quieter

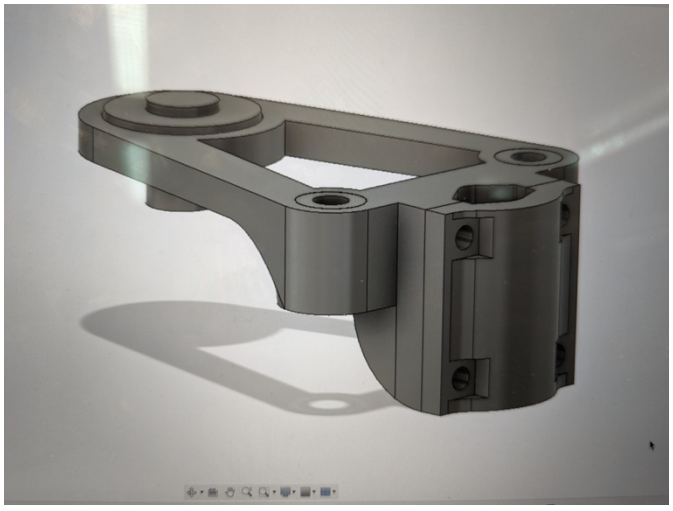
I then set about rewriting all the code in python, using Picamera to talk to the camera module, using OpenCV for Canny Edge Detection and then contour detection. This outputs an array of arrays which can then be used to generate a text file that is saved as a .gcode file, which in turn is passed line by line over the serial connection to the arduino.

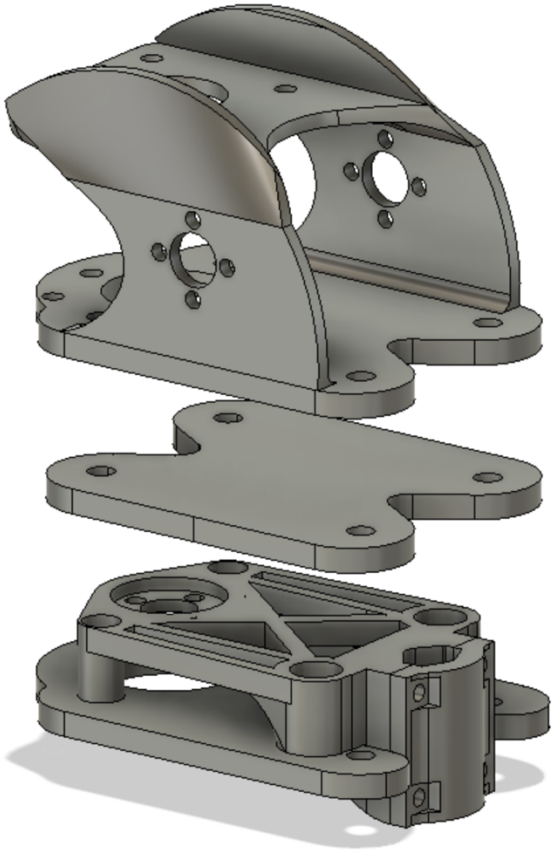


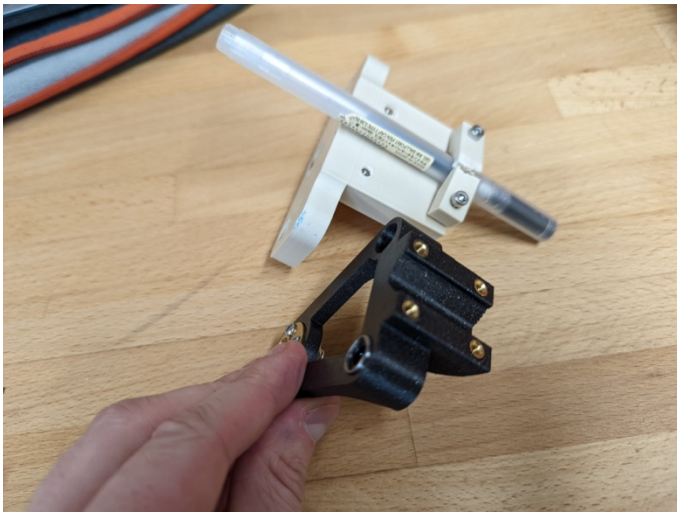


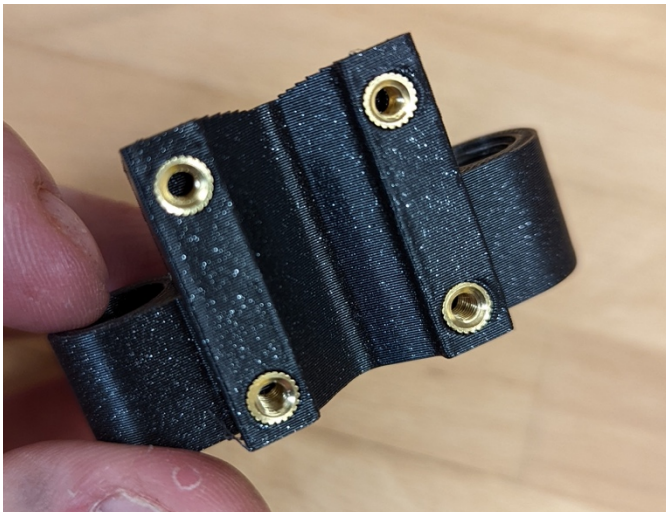
V3.0 suffered from a large deflection at the pen due to the large moment from the lead screw (point of rotation), the weight and the flex of the plastic.

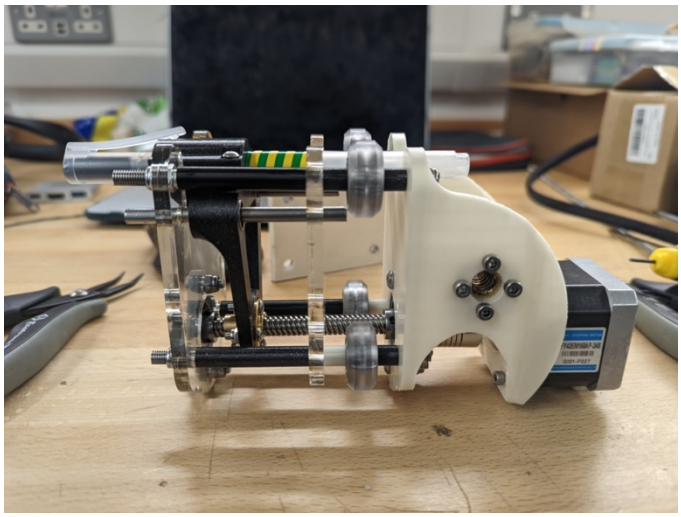
In order to remedy this, I re-designed a much tighter packaged and more balanced carriage.

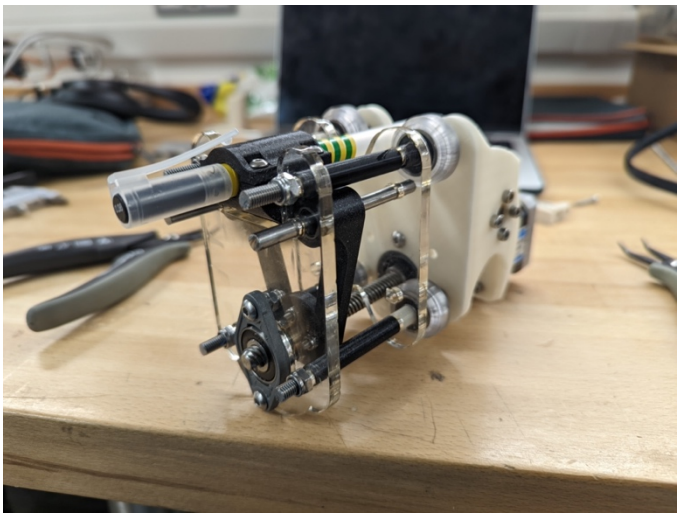






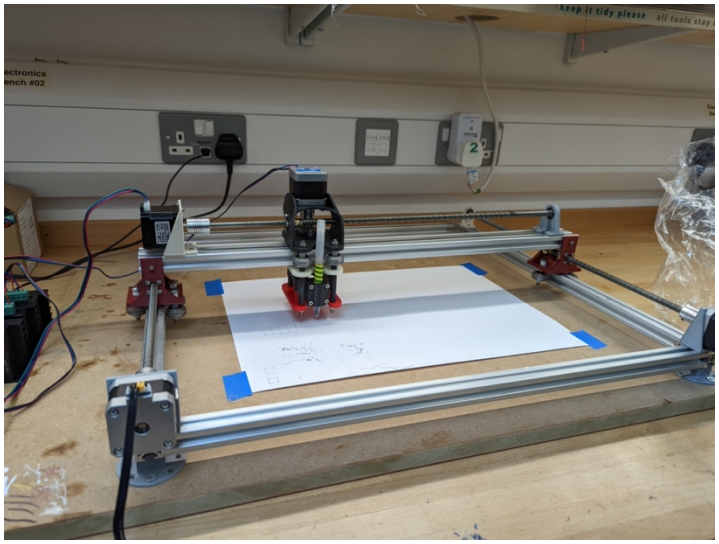


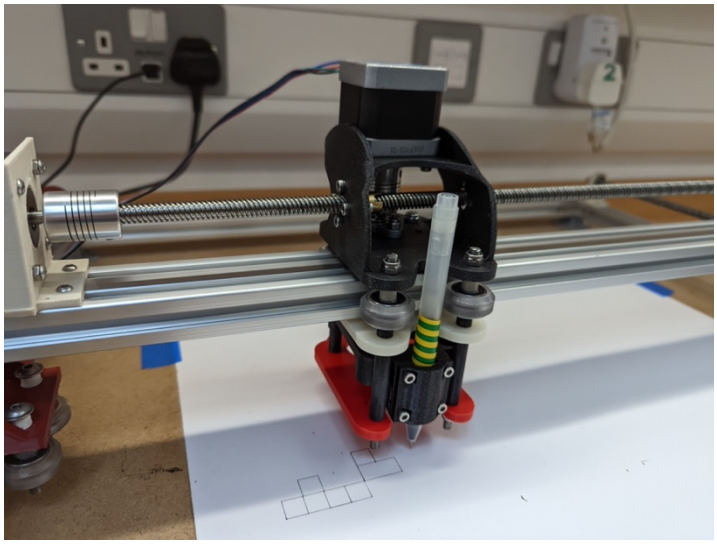


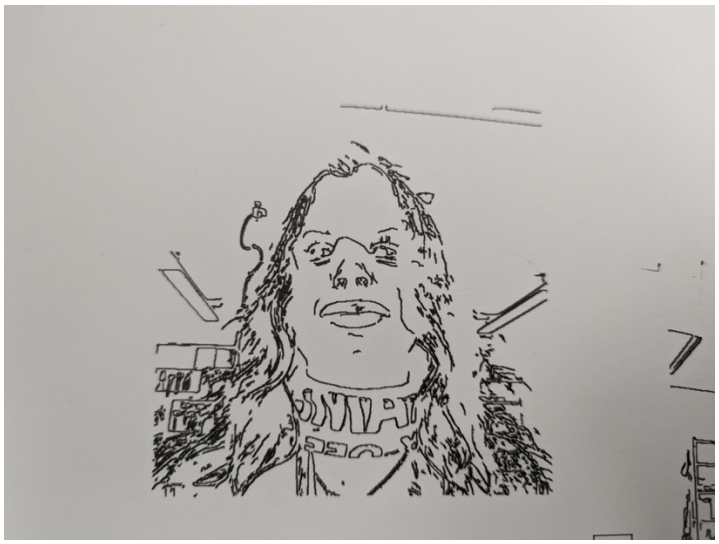


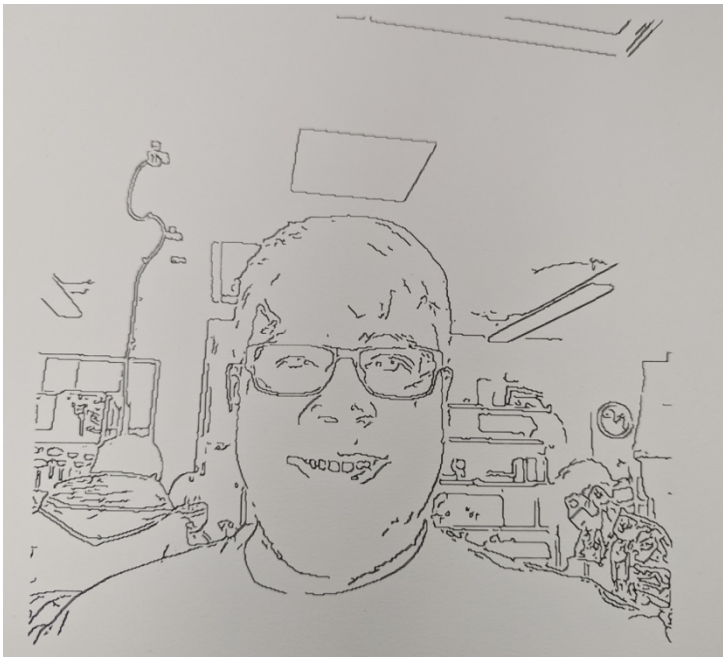
V4.0 had an initial hiccup of being slightly too large, which resulted in a couple of reprints to get the gap between wheels straddling the 2040 extrusion correct to 0.1 mm.

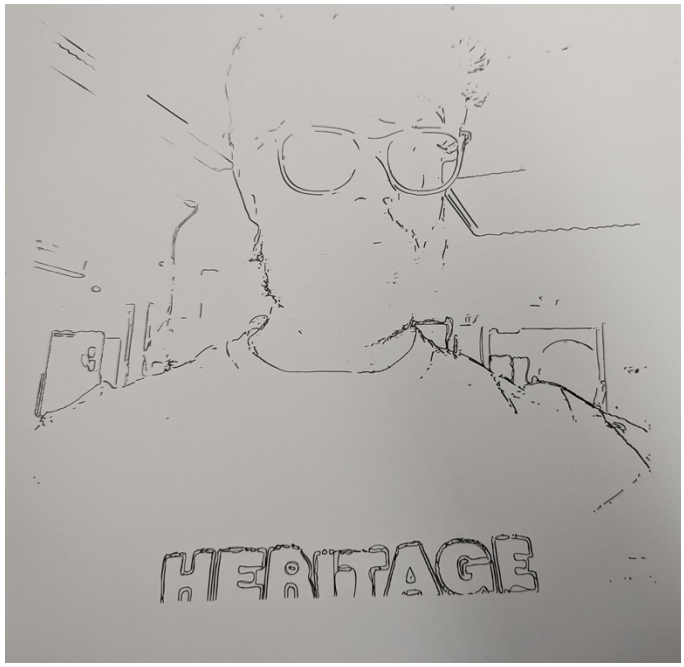
Once this had been dealt with I was able to start increasing the moving and drawing speeds greatly, bringing the drawing time for a frame to a reasonable ~30 mins.



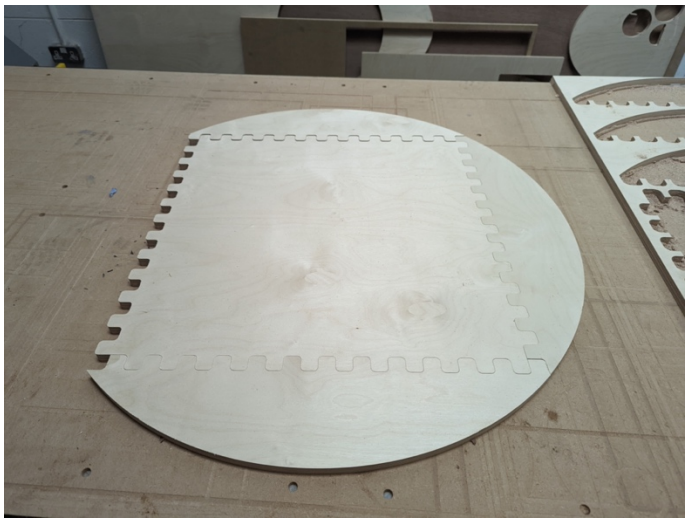


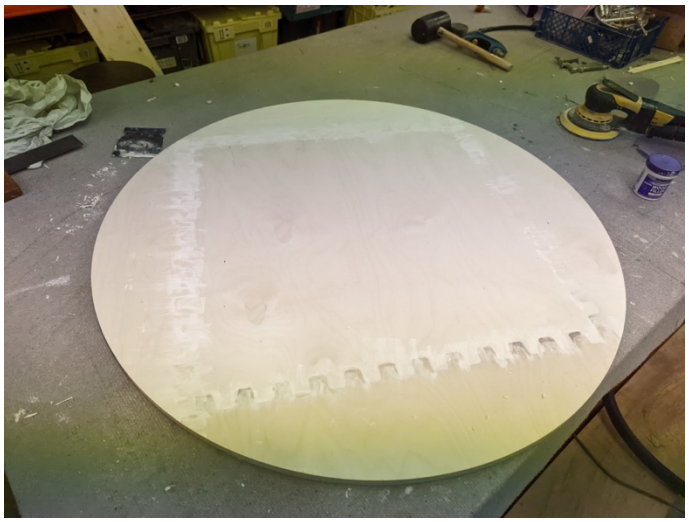


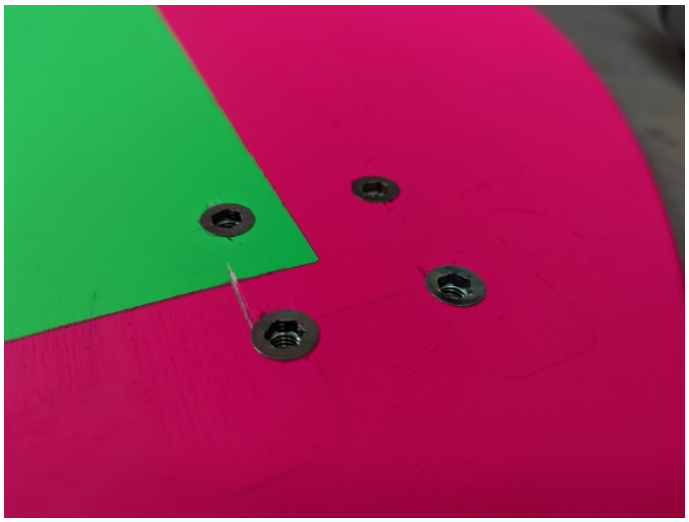


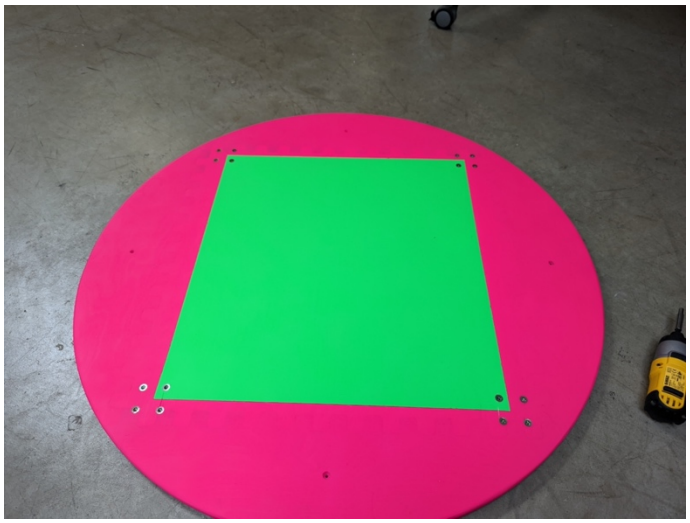


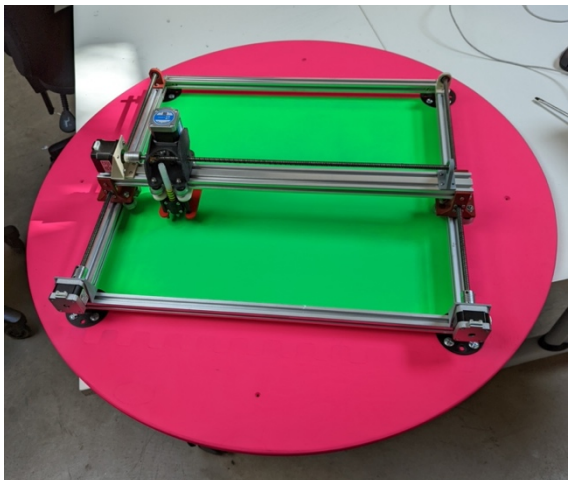
After bringing the drawing speed and accuracy up to an acceptable level, I began to work on improving the software to give a smoother line rather than the more digital vertical or horizontal stepped lines I had been producing. Using Cubic interpolation, I was able to output smooth line drawings (bottom left).





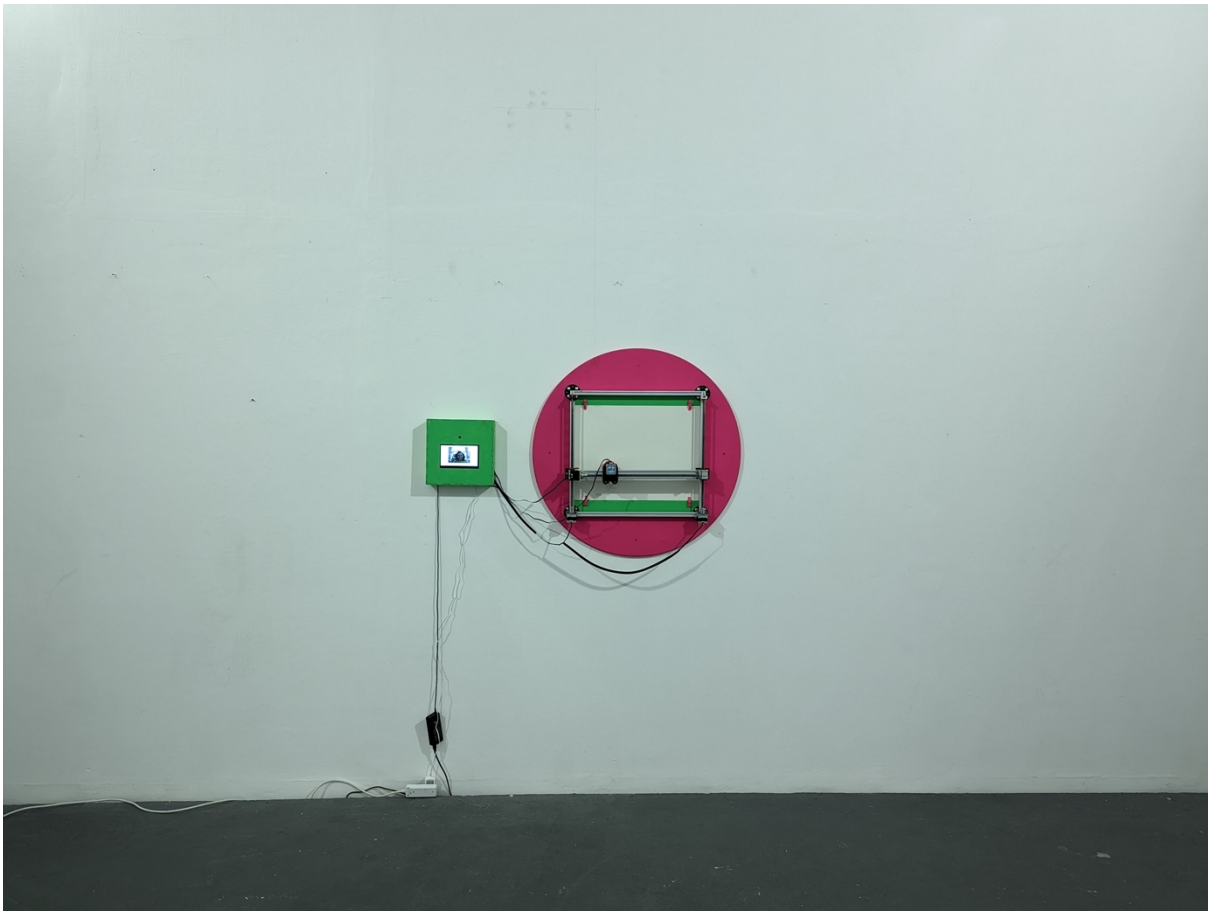


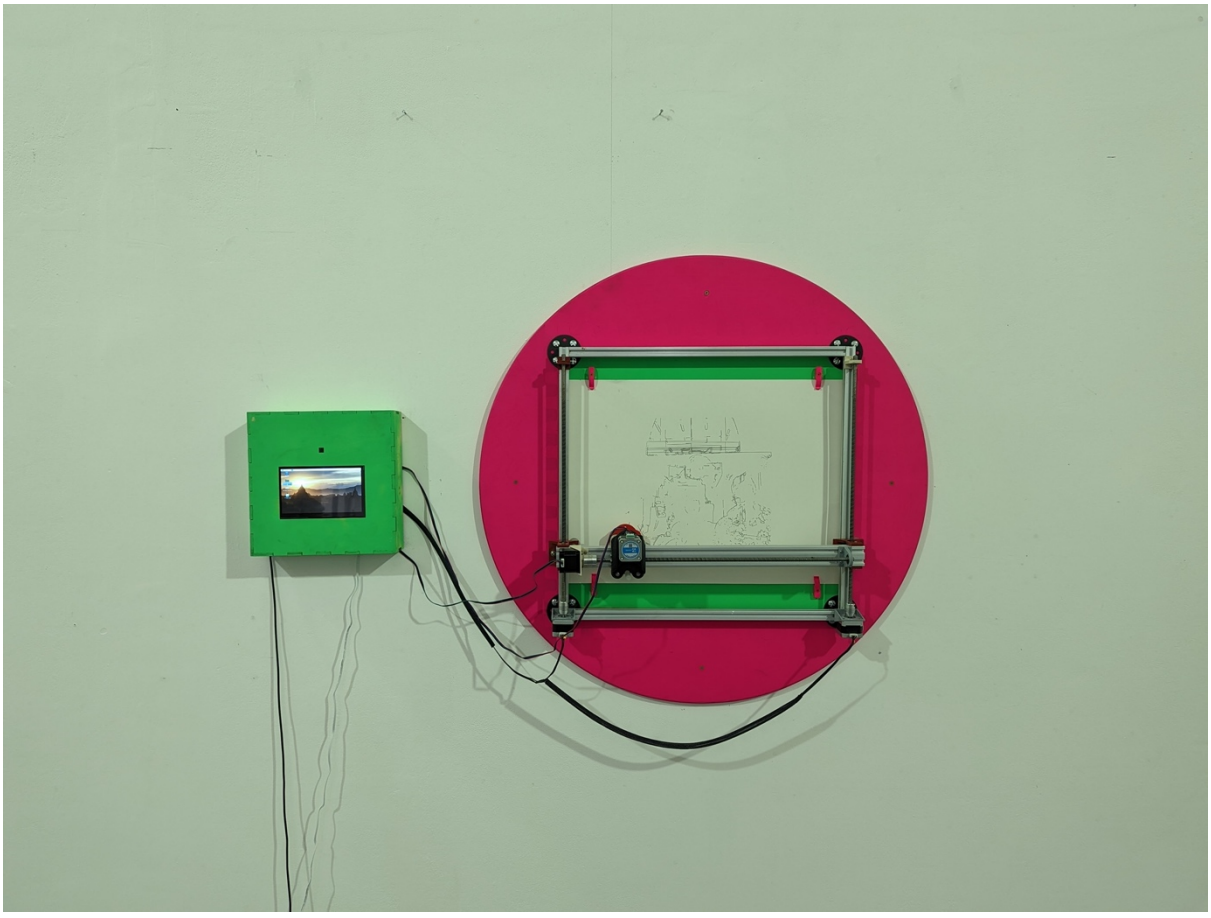


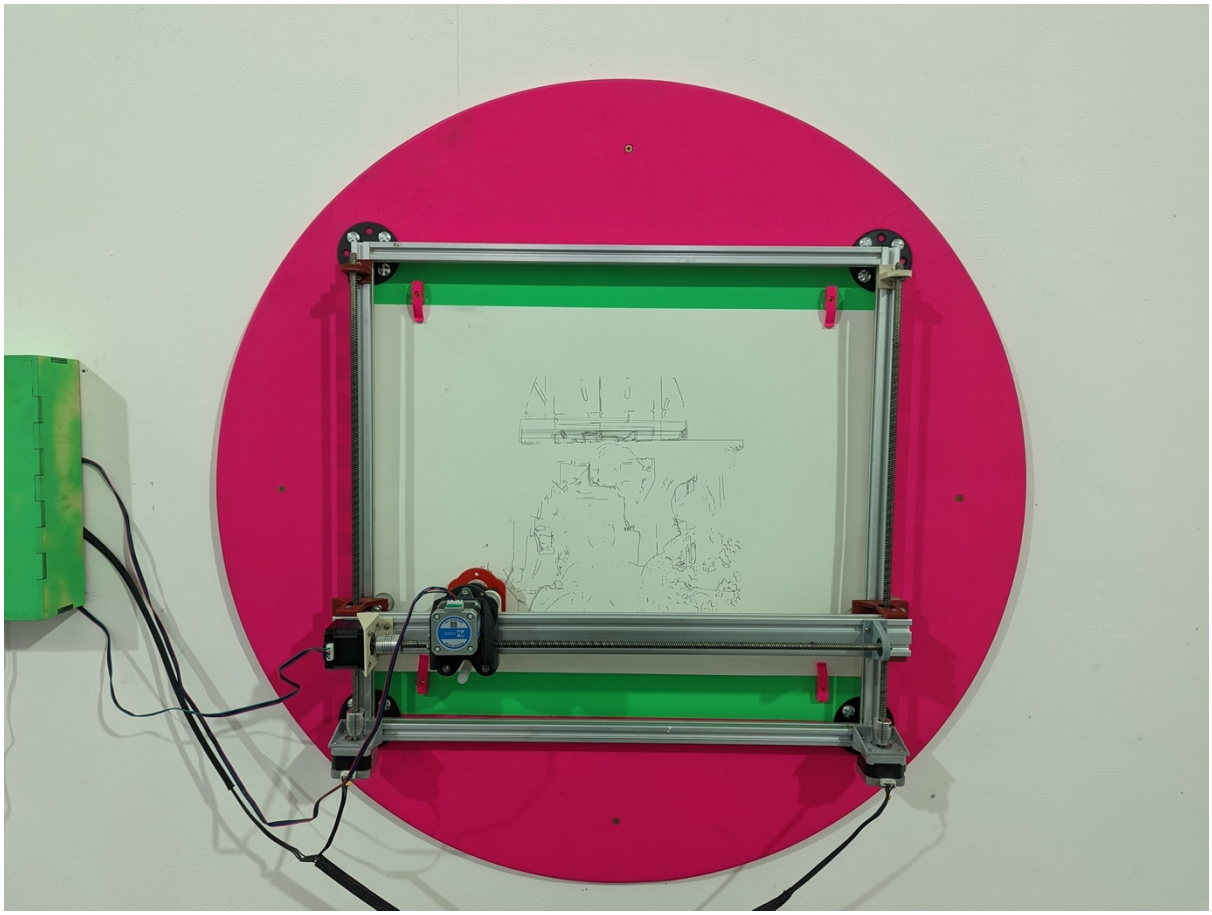


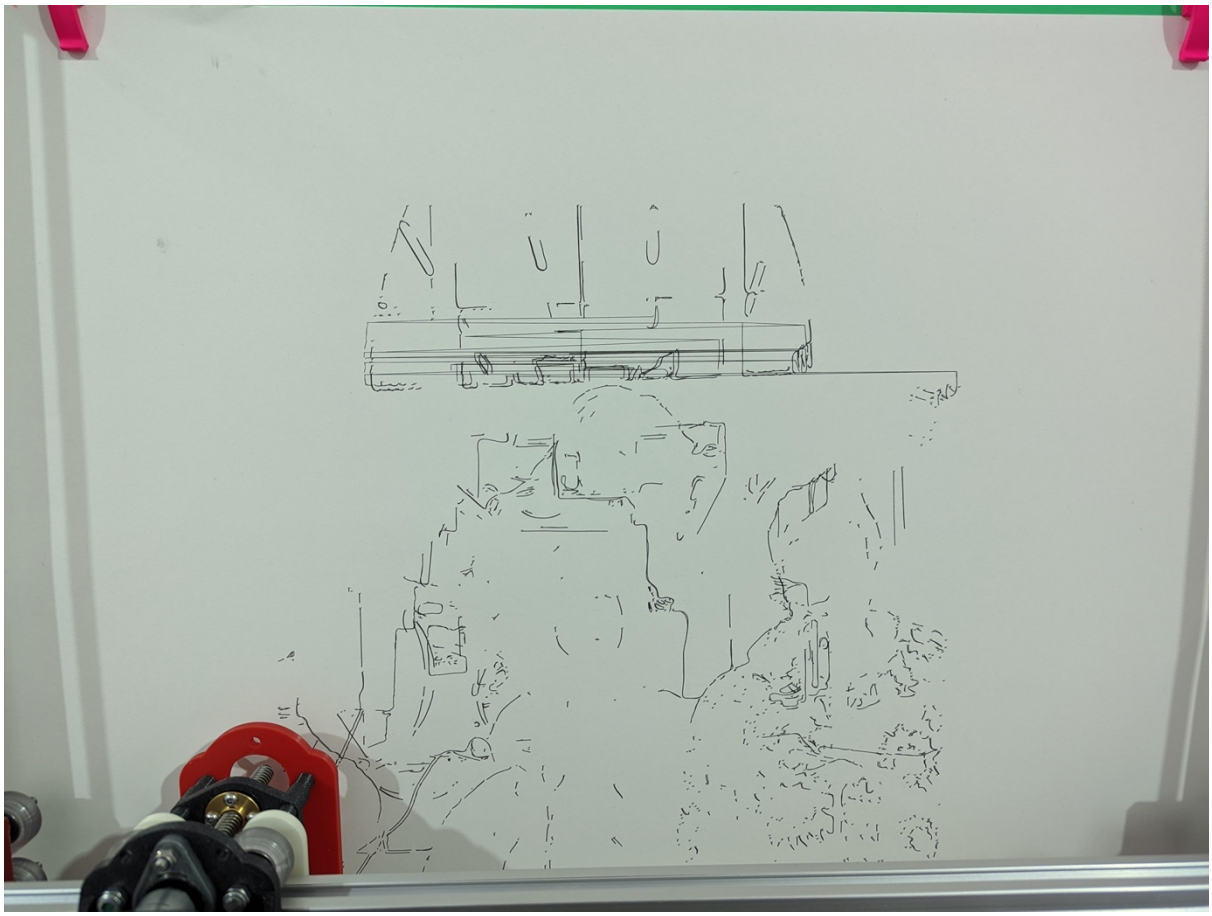
The base was made of 18mm Plywood cut from offcuts on a cnc, so I used a puzzle outline to fit them together with 0 tolerance before gluing and painting. The high contrast neon pink and green are for a striking effect to create a sculptural element on the wall not simply a functional one.

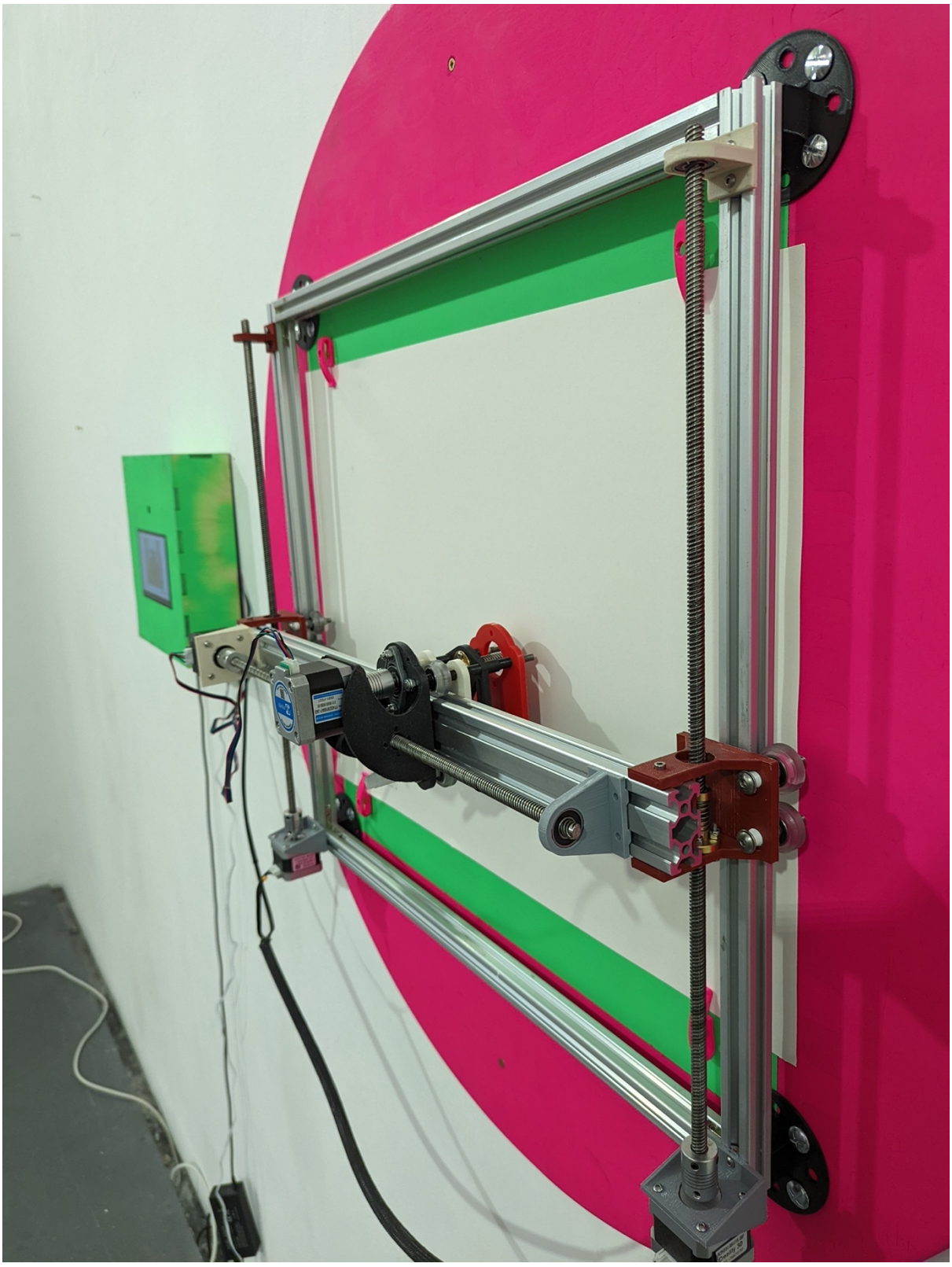
The plotter is held to the board using threaded inserts and there are 4 countersunk holes for attaching via screws to the wooden shell of an exhibition space.











And here is the final piece installed on a wall and ready to draw!

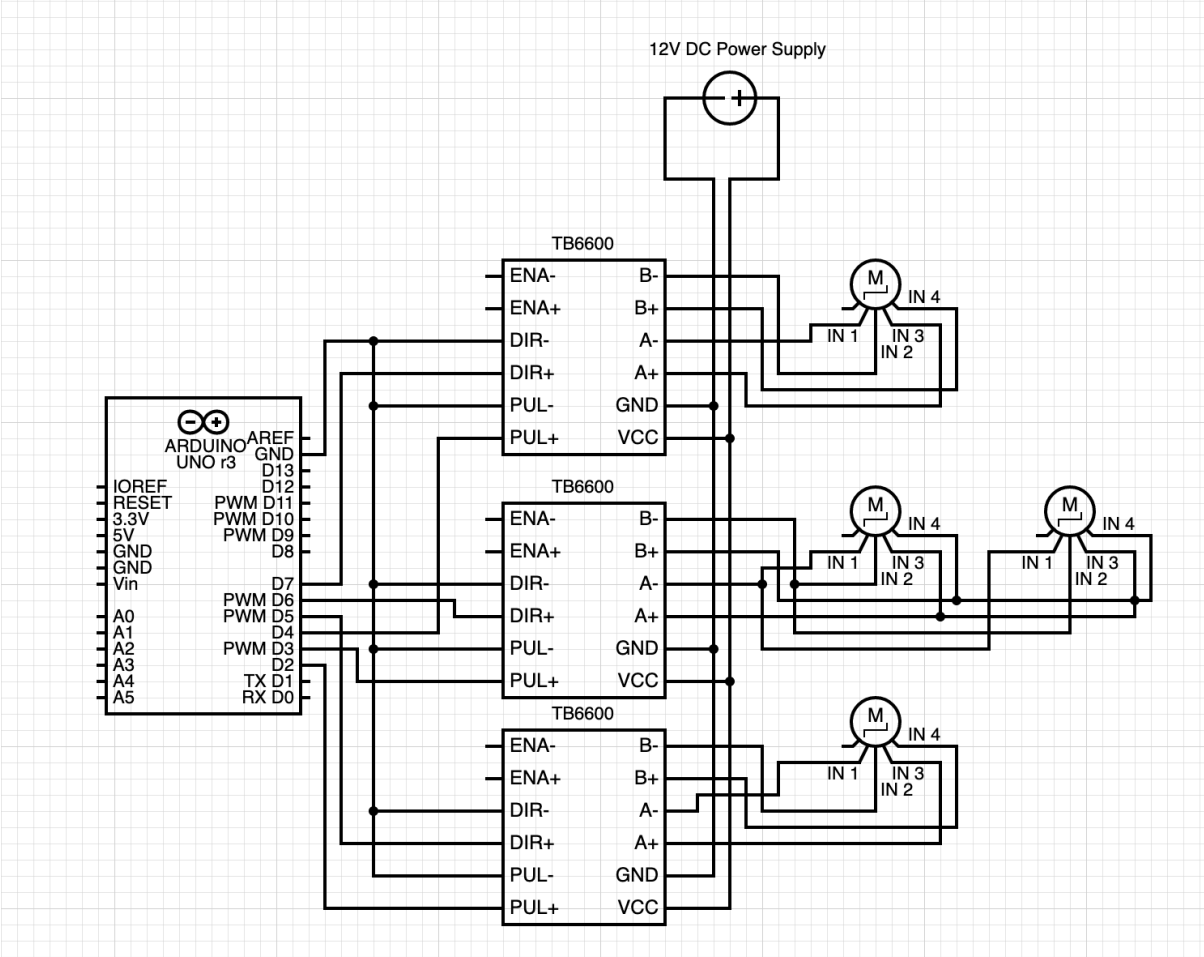
A python script uses the PiCamera module to take a photo, which is then passed through the Canny Edge Detector in OpenCV. The thresholds for this detector are adjustable in the debug mode of the GUI. This image is then passed through the Contour detector which returns an array of contours which in turn are arrays of points. Each contour is passed through a cubic interpolation method to smooth out lines, producing a new longer array of points which is used to write a gcode file that is finally passed, line by line, over serial to the Arduino which runs the GRBL gcode interpreter firmware.

A basic GUI written with PyQt5 displays working buttons to use debug mode or run the program, and displays the current image being drawn as it does so.

List of parts:

- Birch Plywood Base
- Laser Cut Plywood Housing
- 2020 V-Slot Aluminium Extrusion 600mm X 4
- 2040 V-Slot Aluminium Extrusion 600mm X 1
- 8mm Lead Screw 500mm X 3
- 8mm Lead Screw 100mm X 1
- Lead Screw Coupler X 4
- Bearing X 3
- VSlot Wheels X10
- Nema 17 Stepper motor X 4 (Model Numbers: Two Trees 17HS4491S, Creality 3D JK2HS34-0844YA-06F X 2, FOYO FY42EM150AP-24B)
- TB6600 Stepper Motor Driver X 3
- Elegoo Arduino Uno X 1
- Raspberry Pi 3B+
- Raspberry Pi Picamera Module
- Elecrow X000WHY0DD 7 inch Touchscreen Monitor X 1
- 12V 60000mA AC/DC adapter
- Various Cables, screws, nuts and fixings
- 3D Printed X/Z Carriage X 1, Y Carriage X 2, Bearing Holder X 3, Stepper Motor Holder X 3, Feet X 4
- Pen

Circuit Diagram



References

"Auto-Destructive Art – Art Term | Tate". 2021. *Tate*. https://www.tate.org.uk/art/art-terms/a/auto-destructive-art.

Tinguely, Jean. 1960. Homage To New York. Kinetic Sculpture. New York: Museum of Modern Art.