

generate svg for pen plotters using python

//virtual cc fest 2022

./outline_&_resources

outline & resources

session outline

first: this short slide presentation (on plotters and session software)

task 1: some basic svg concepts

task 2: generating svg files with py5

-- *bonus content / stretch goals* --

task 3: generating svg files with blender

task 4: some codeless svg generation (uji, inkscape l-systems)

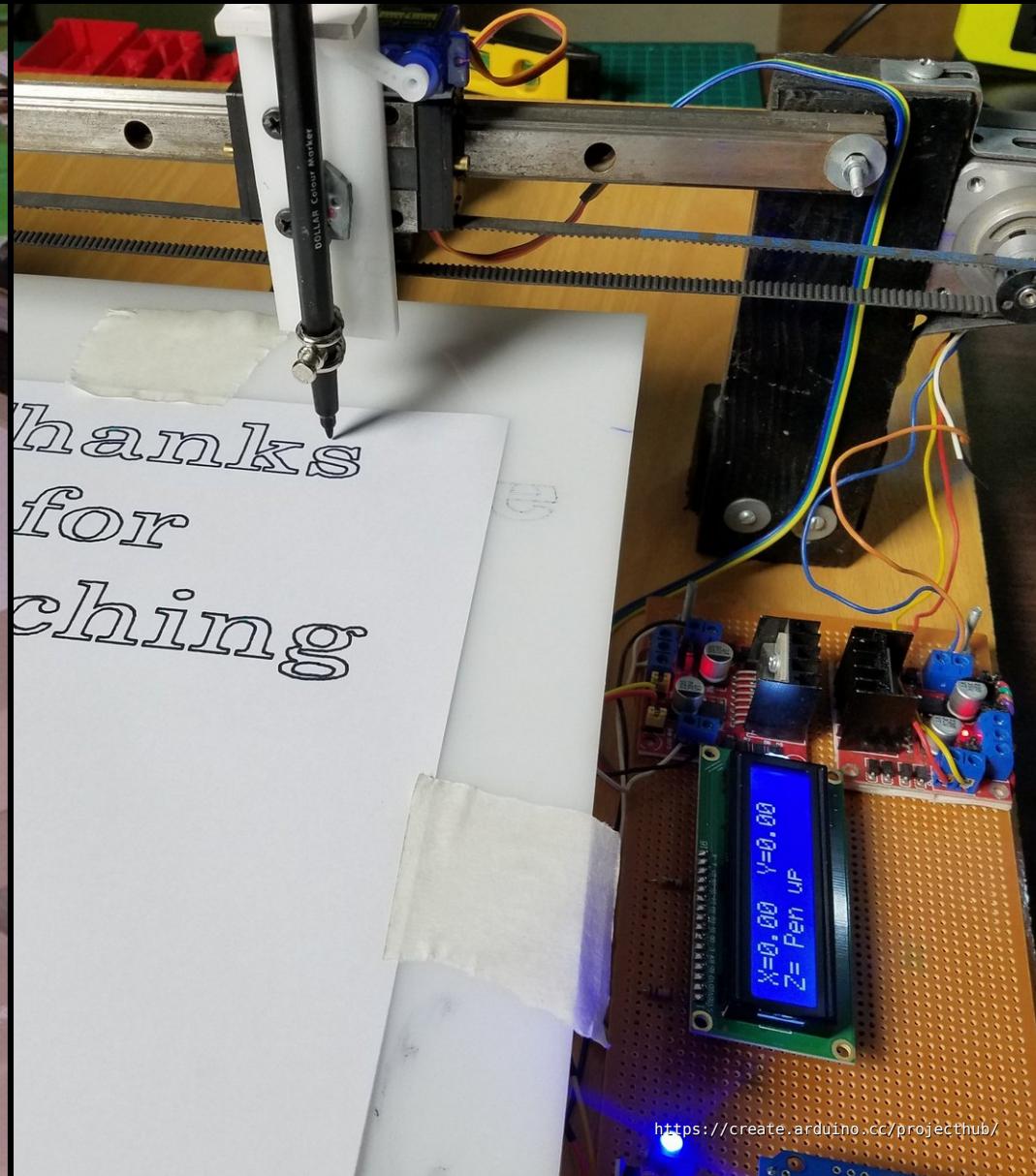
<https://github.com/tabreturn/cc-fest-plotter>

- * session recording (uploaded later)
- * task files
- * py5 resources
- * useful plotter utilities
- * inspiration (artworks)
- * other useful resources

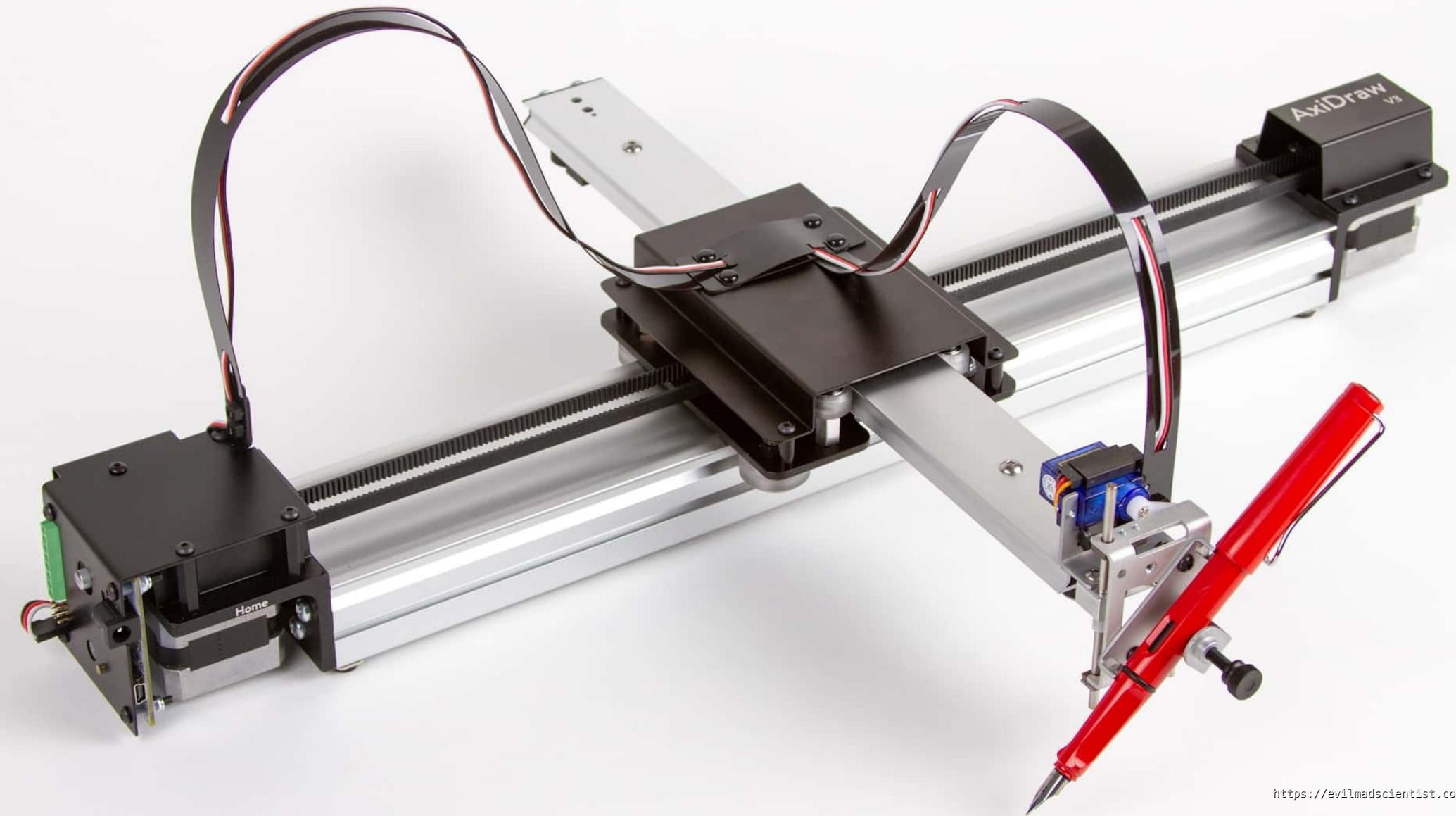
`./plotters`

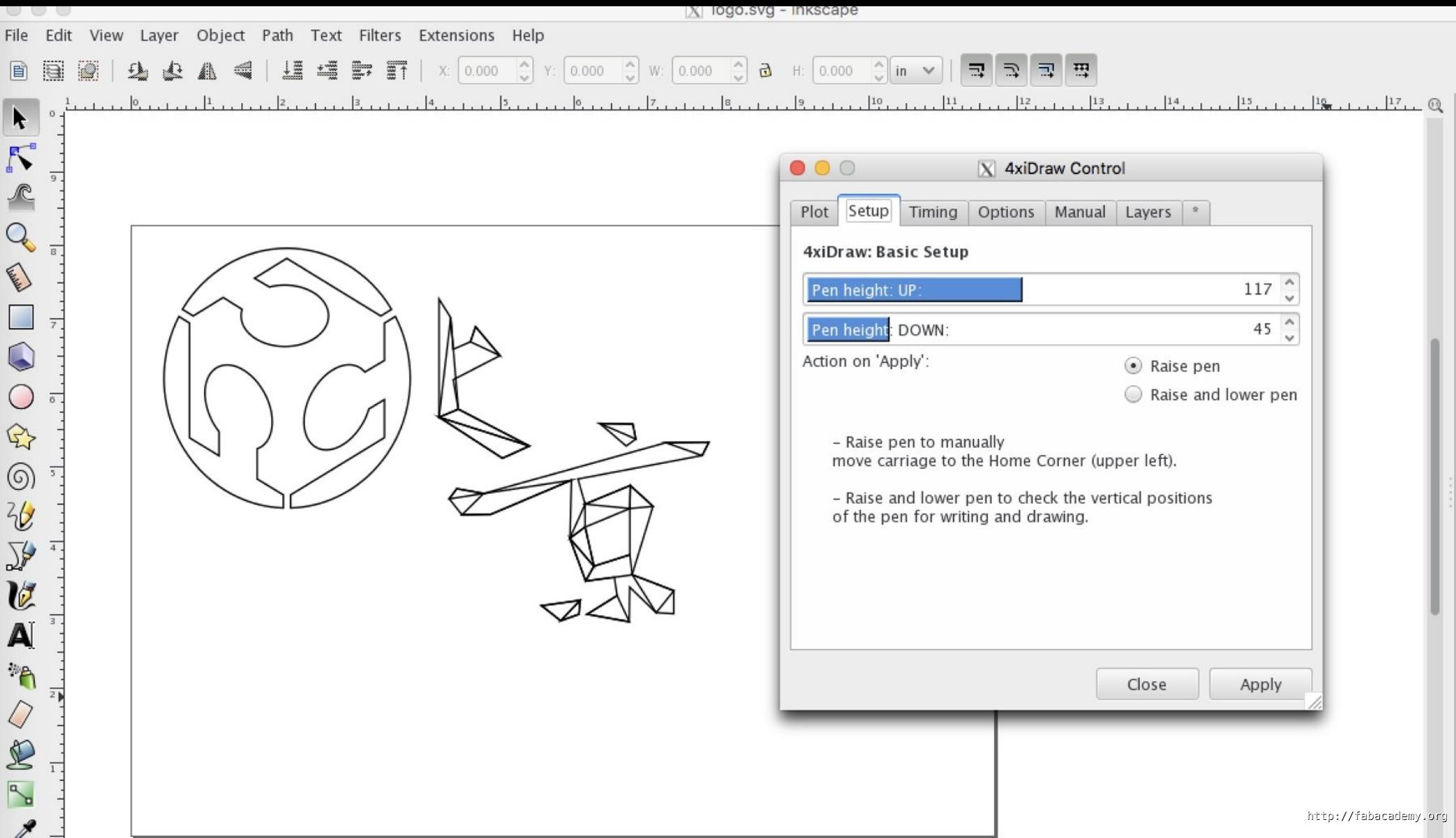
`plotters`

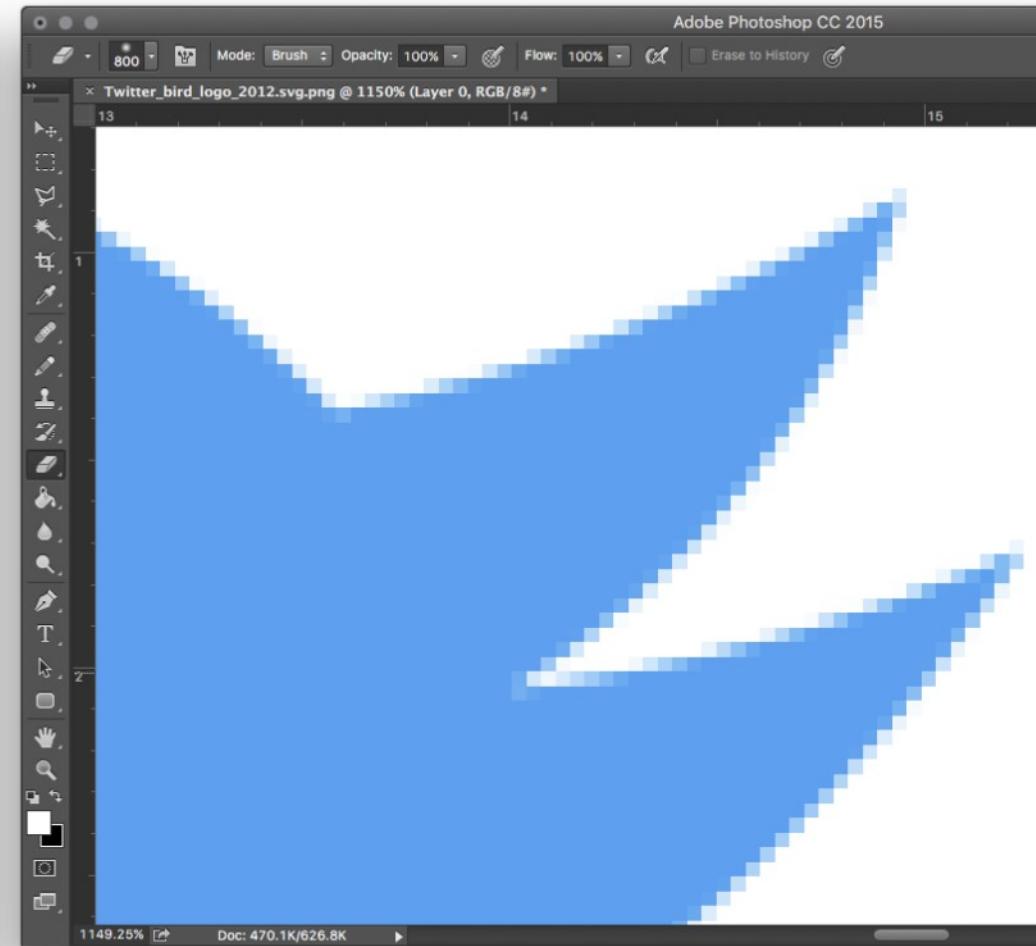
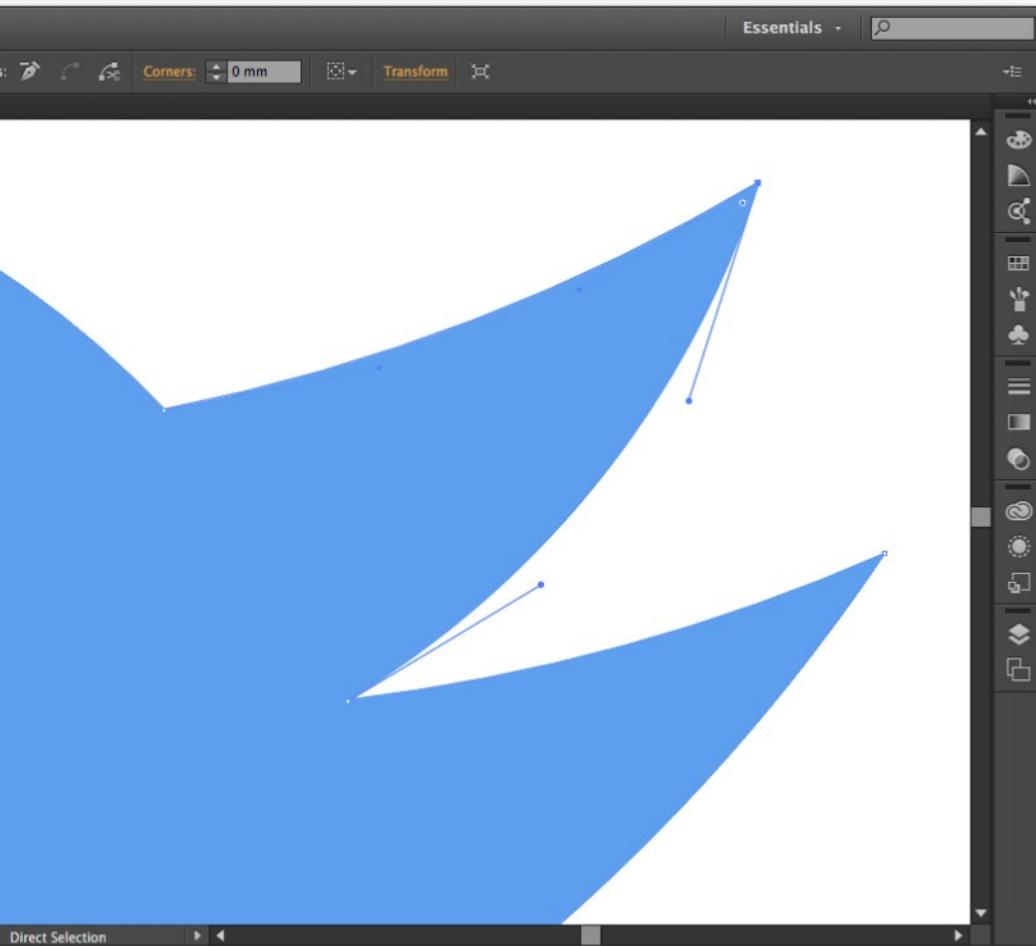
./plotters/examples



./plotters/setup/axidraw





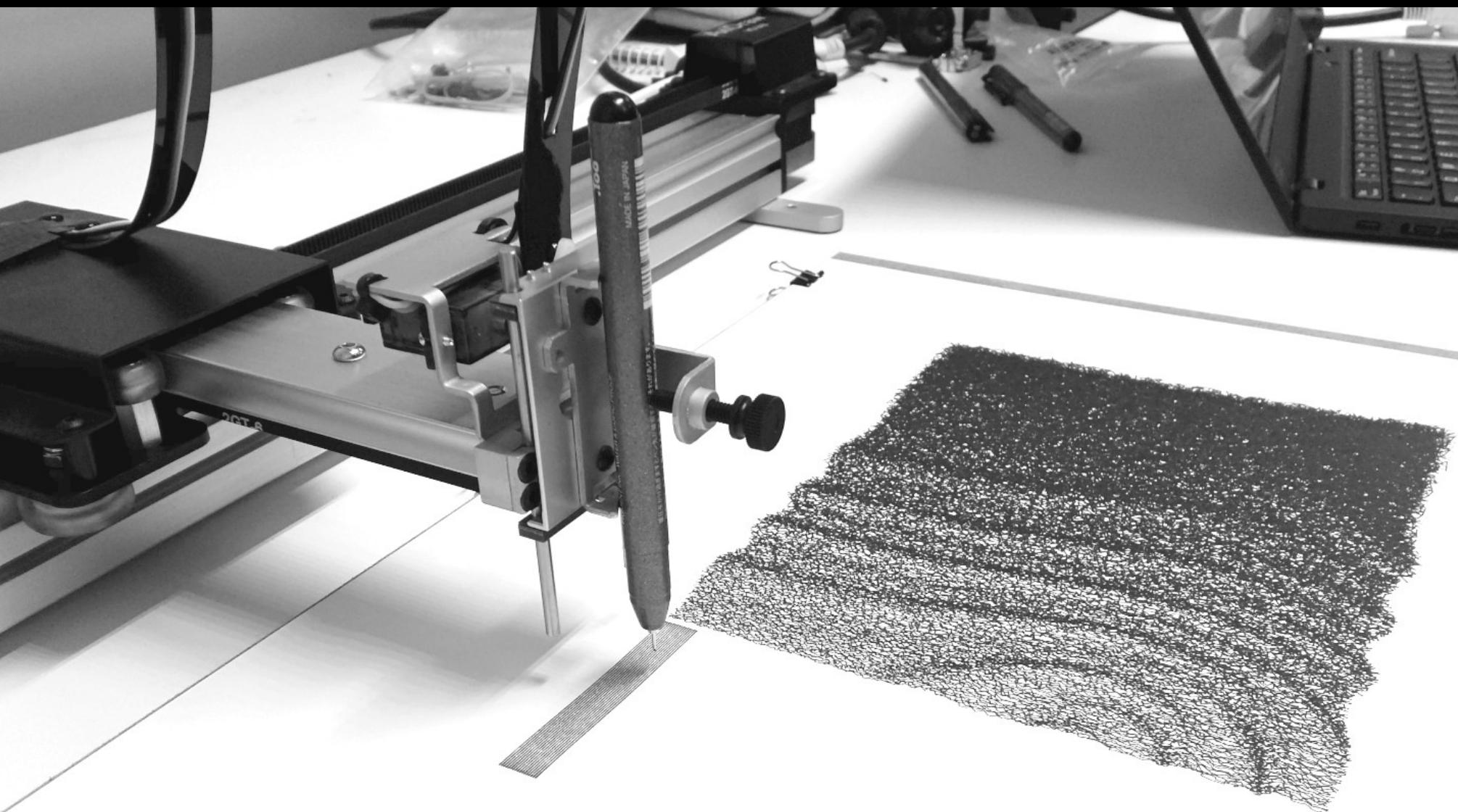


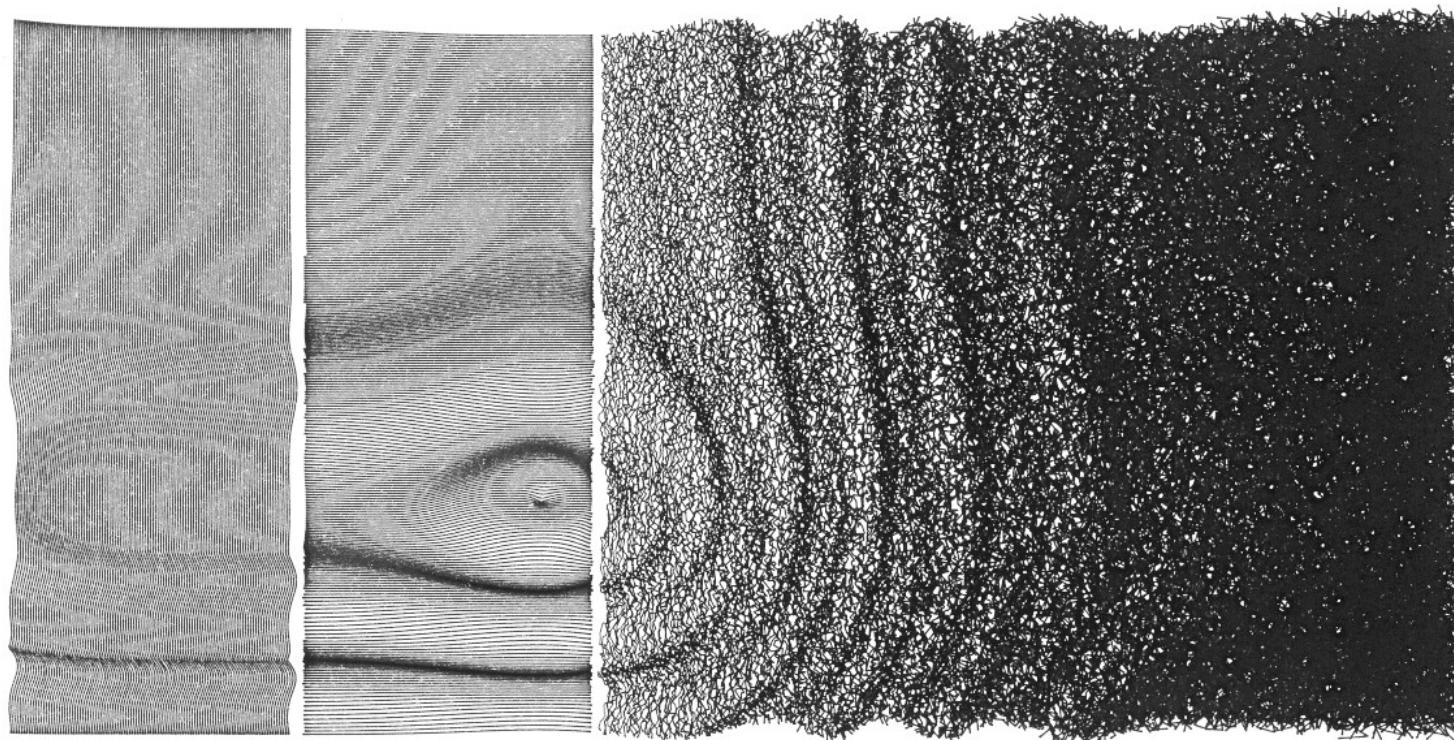
```
<rect x="10" y="10" width="100" height="100" fill="blue" />
```

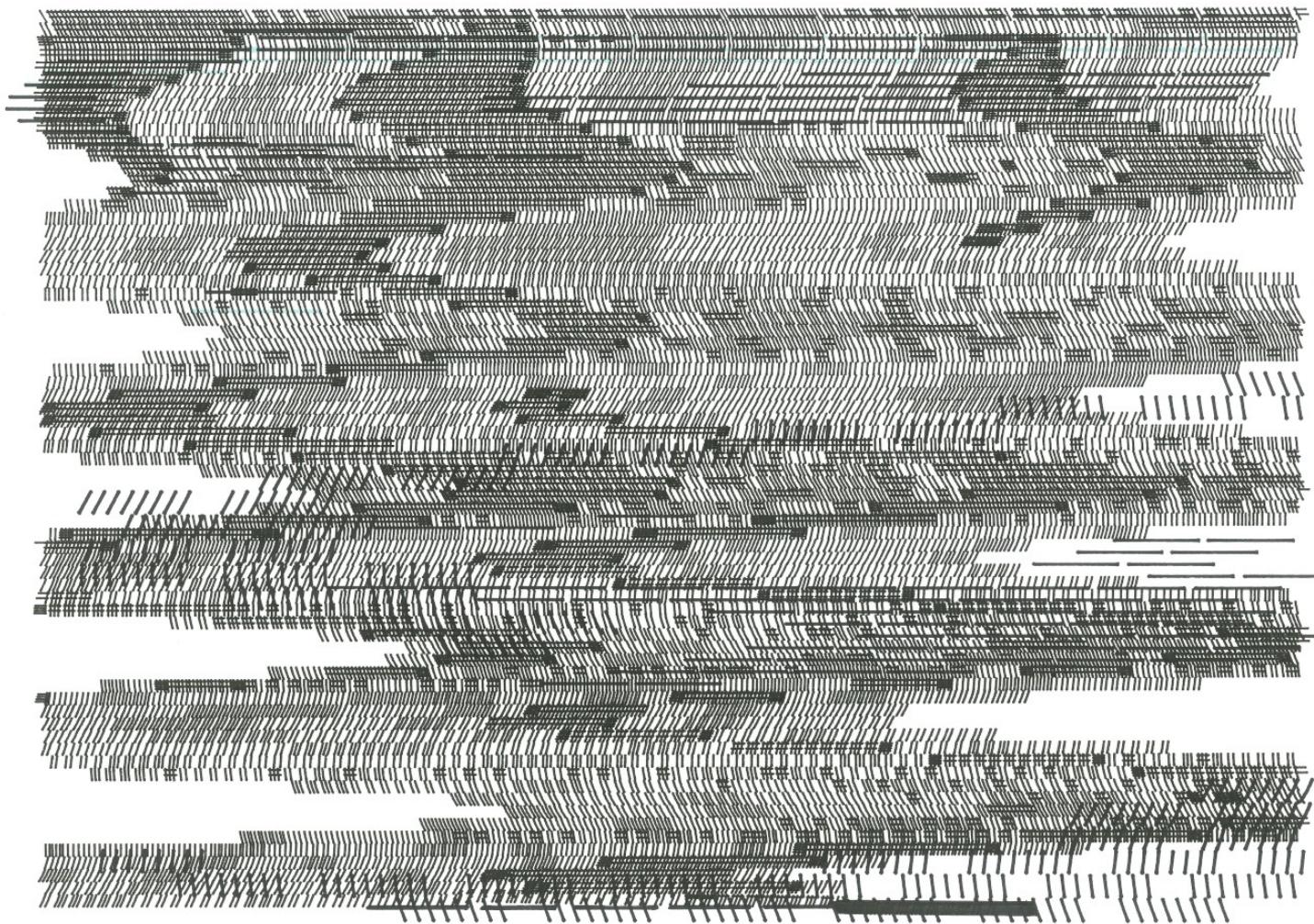
but more on svg later ...

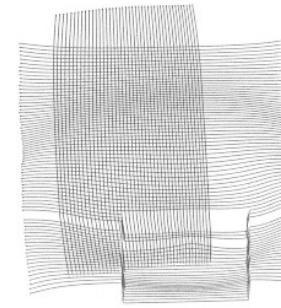
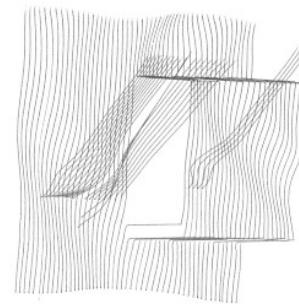
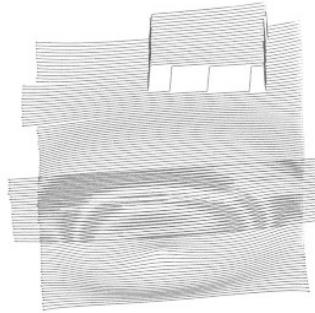
./plotter_art

plotter art

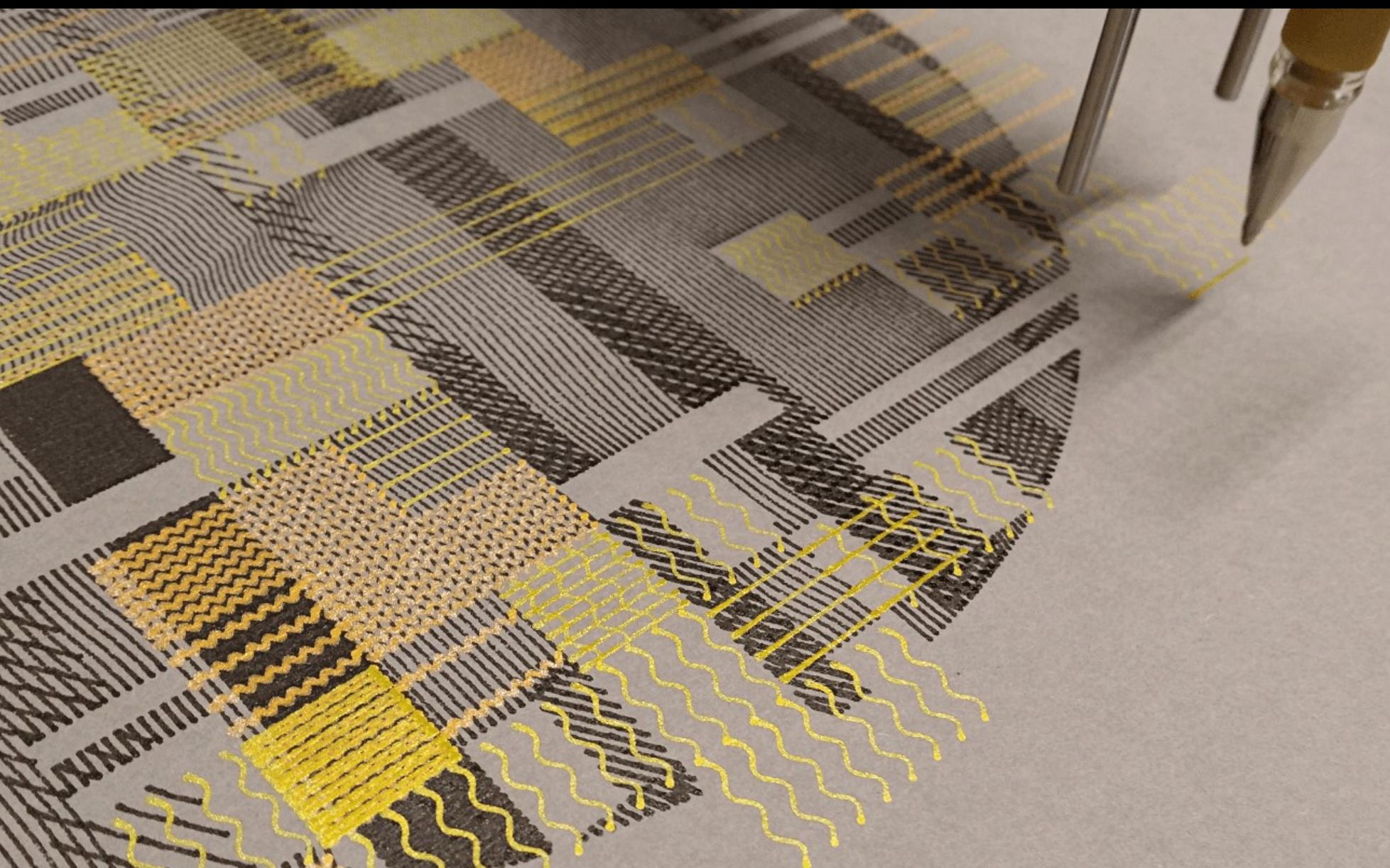




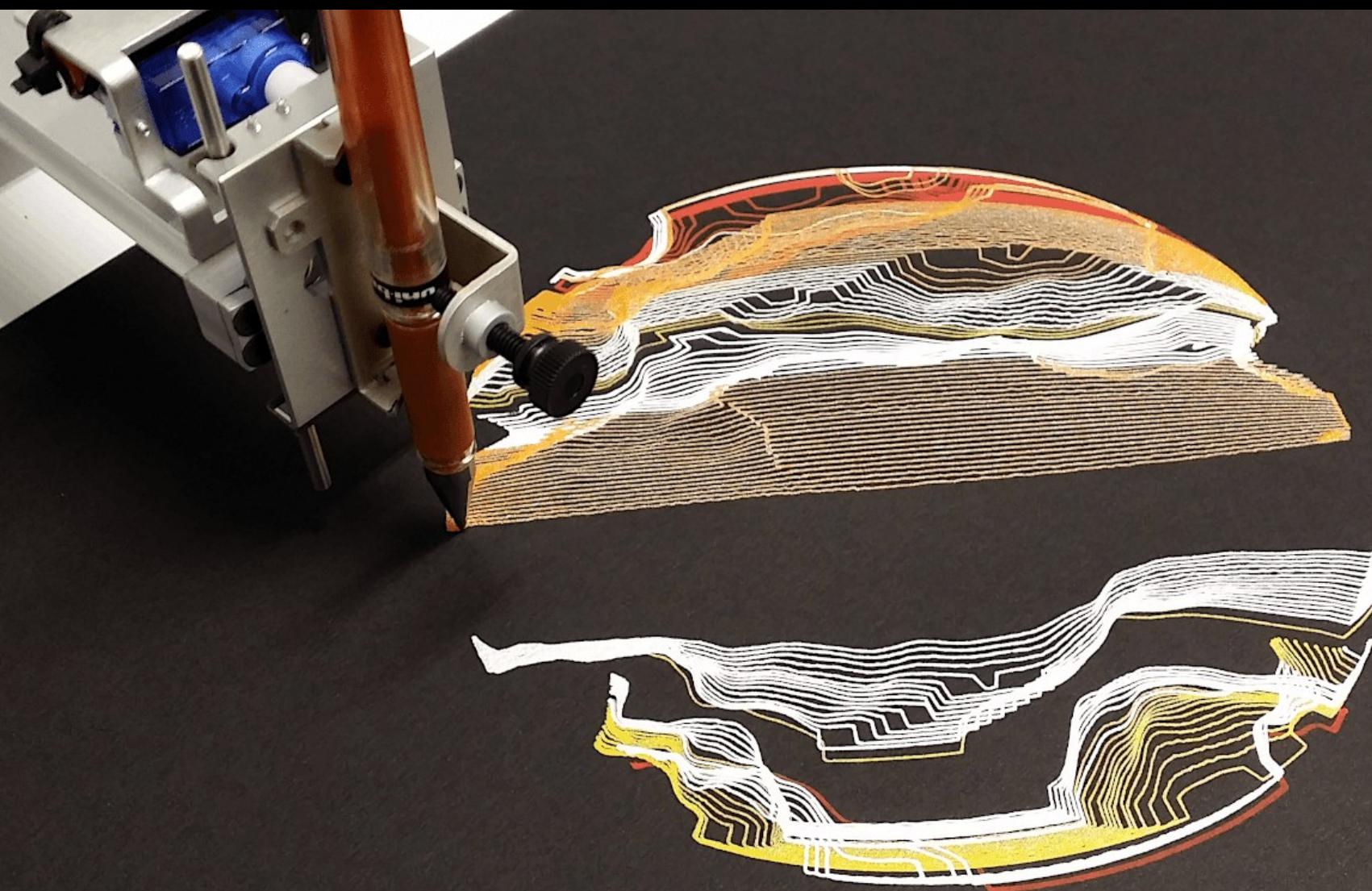




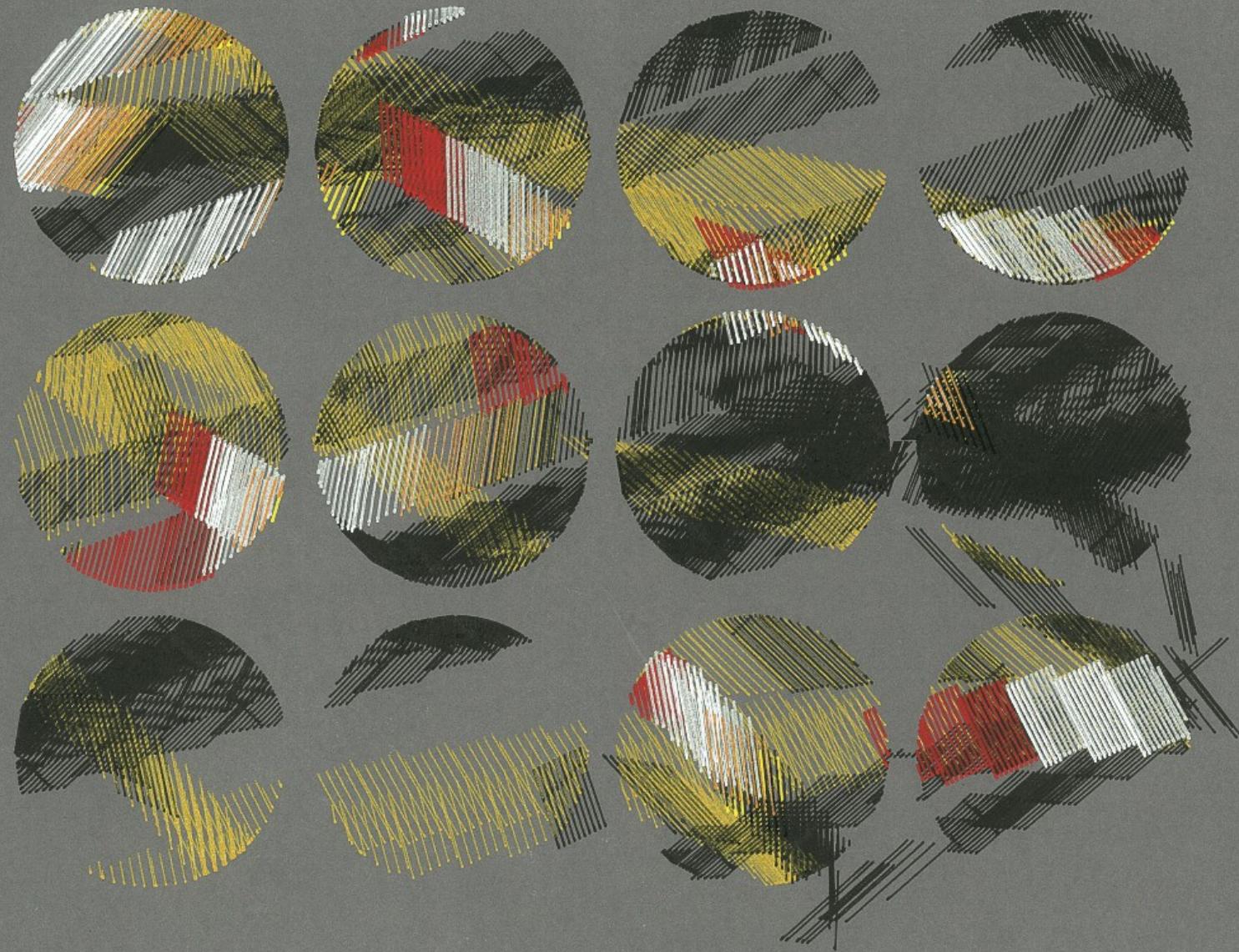


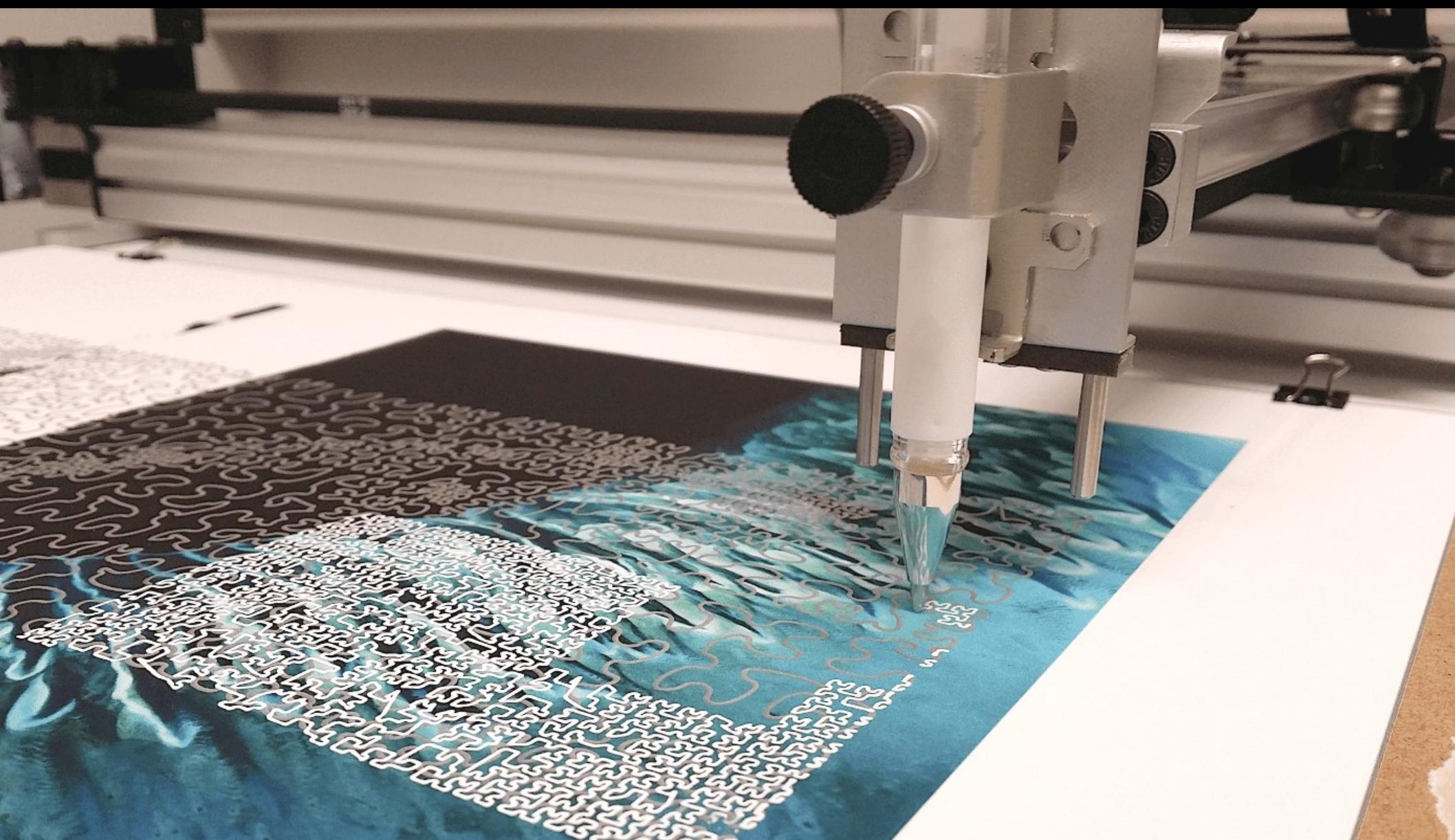


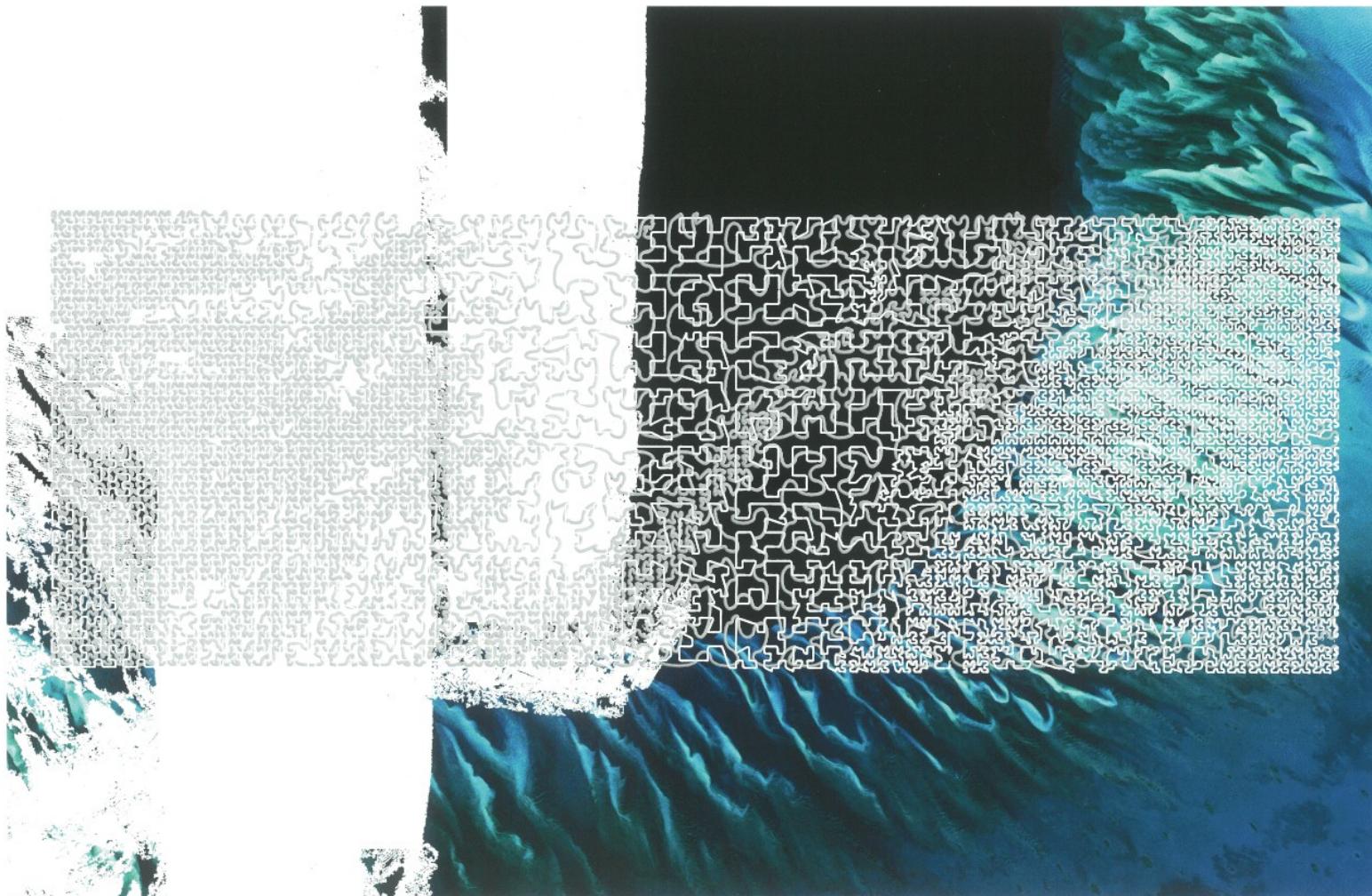
./plotter_art/multiple_inks

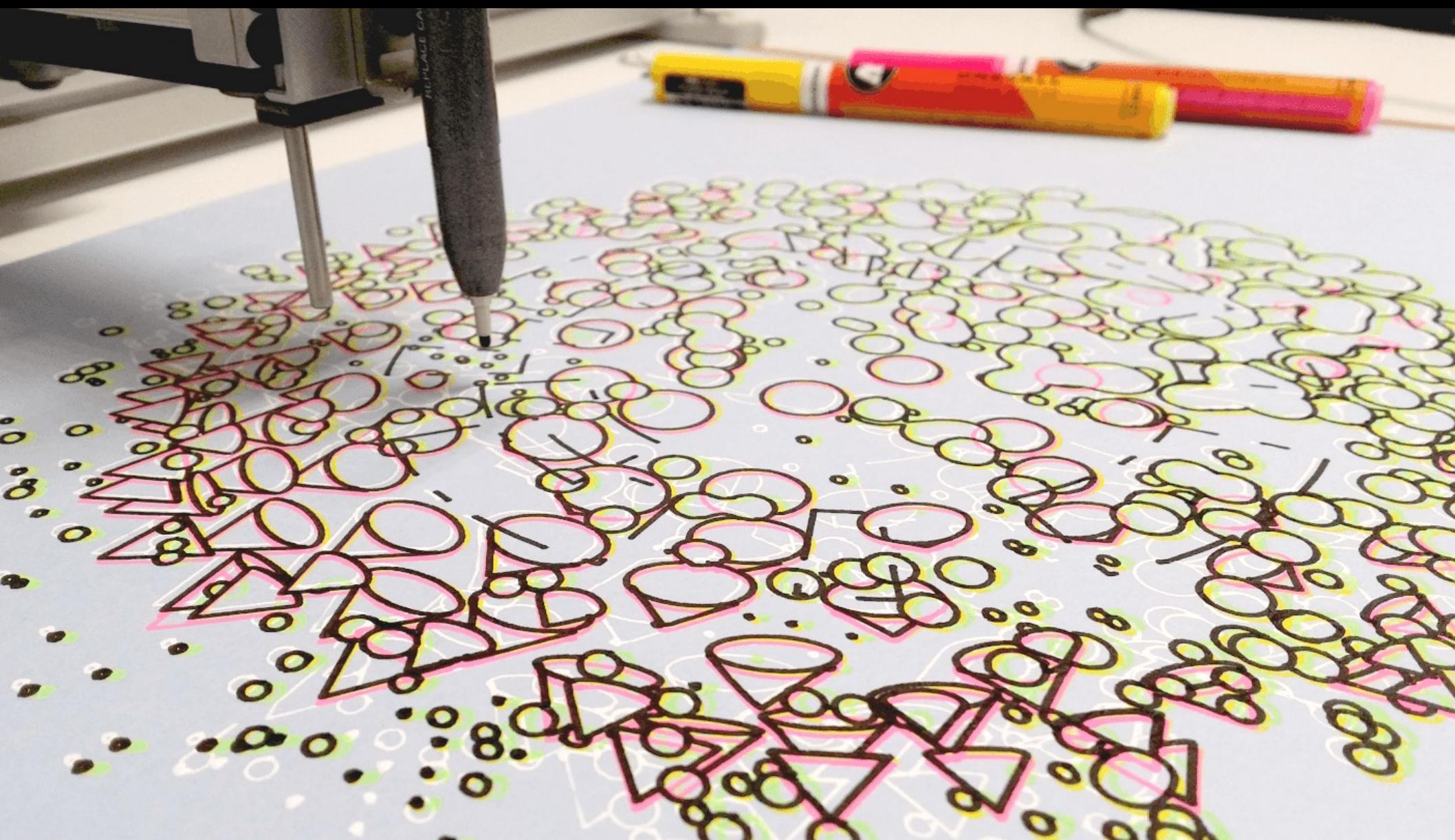


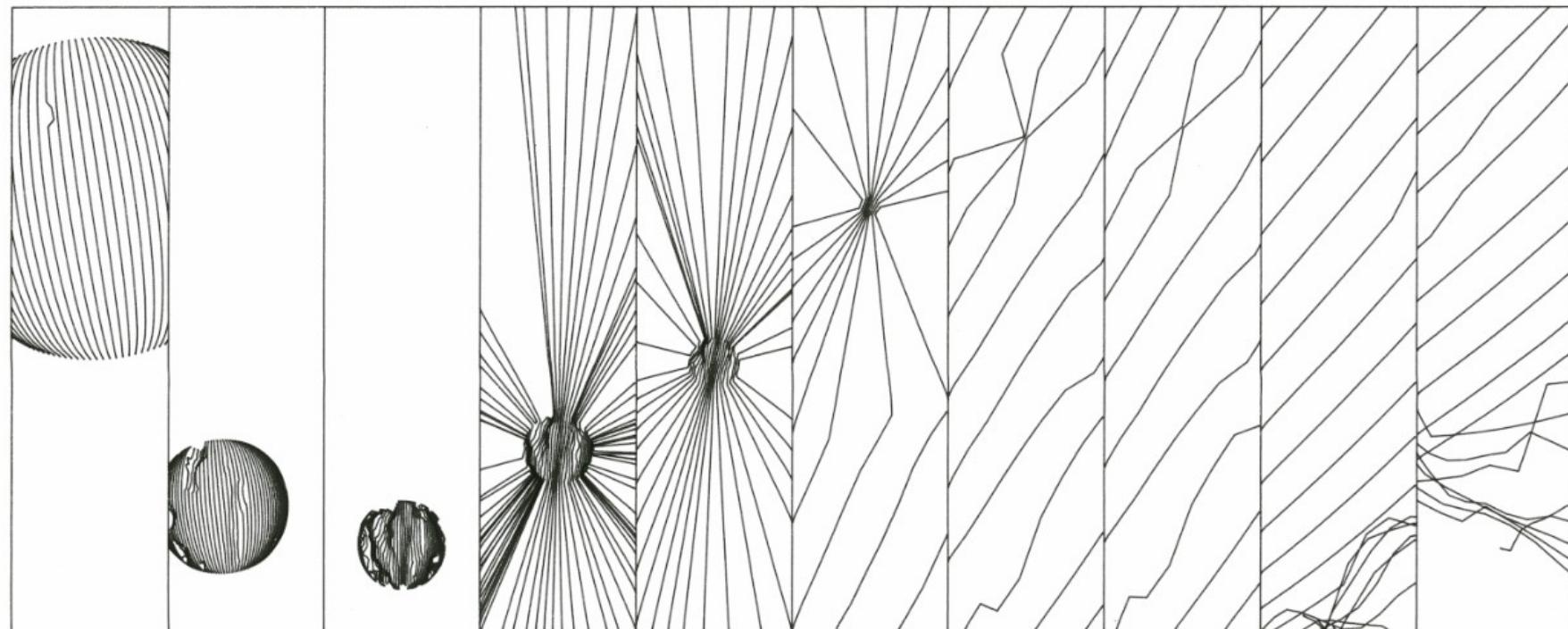












`./plotter_art/inspiration`

for more inspiration, see the github resources page

(links to artists and websites showcasing plotter art)

./py5

py5

"py5 is a new version of processing for python 3.8+. it makes the java processing jars available to the cpython interpreter using jpyre. it can do just about everything processing can do, except with python instead of java code."

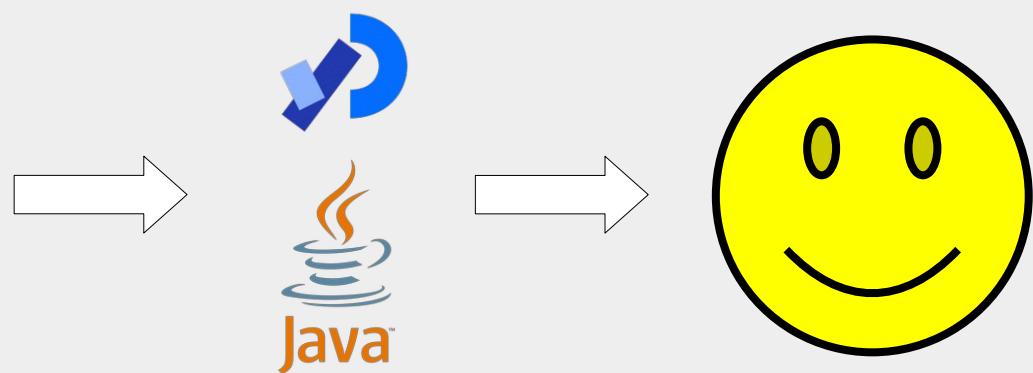
"processing.py (processing python mode) is a jython version of processing, combining the same processing core libraries that py5 utilizes with jython, a java implementation of python. processing.py is the spiritual ancestor to and inspiration for py5."



A screenshot of the Processing IDE window titled "sketch_220123d | Processing 4.0b3". The main area displays the following Java code:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
processing java code
```

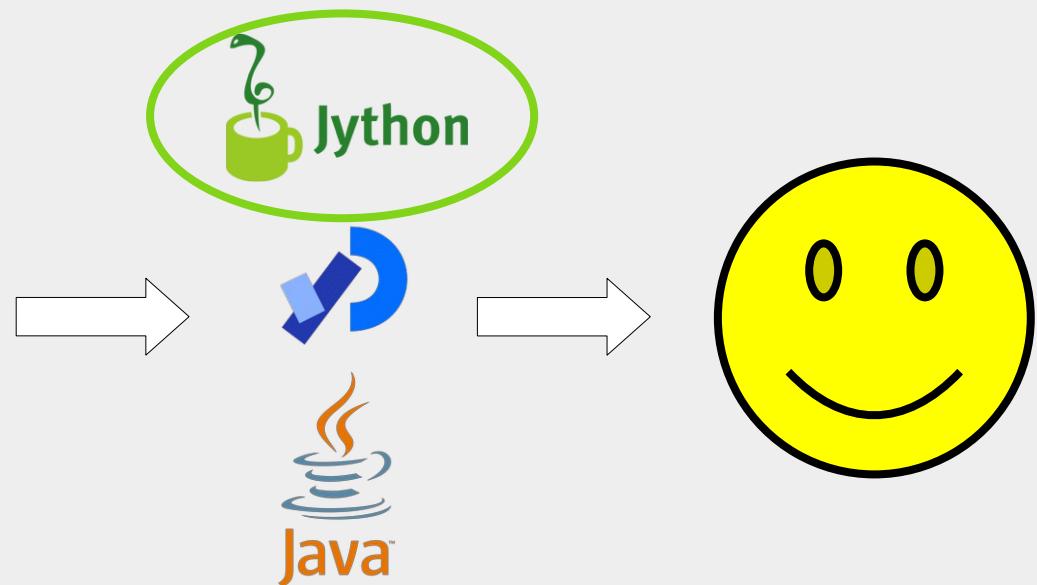
The code is preceded by a series of line numbers from 1 to 17. Below the code area, there is a dark grey workspace. At the bottom of the IDE, there are two tabs: "Console" and "Errors".

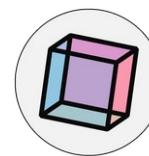


A screenshot of the Processing Python IDE. The main window title is "sketch_220123c | Processing 4.0b3". The code editor contains the following Python code:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
```

processing python code

The code editor has a light blue background and a dark blue header bar. At the bottom, there is a "Console" tab. The status bar at the bottom shows the path "sketch_220123c" and the version "Processing 4.0b3".



py5

sketch_210819a.py - VSCode

File Edit Selection View Go Run Terminal Help

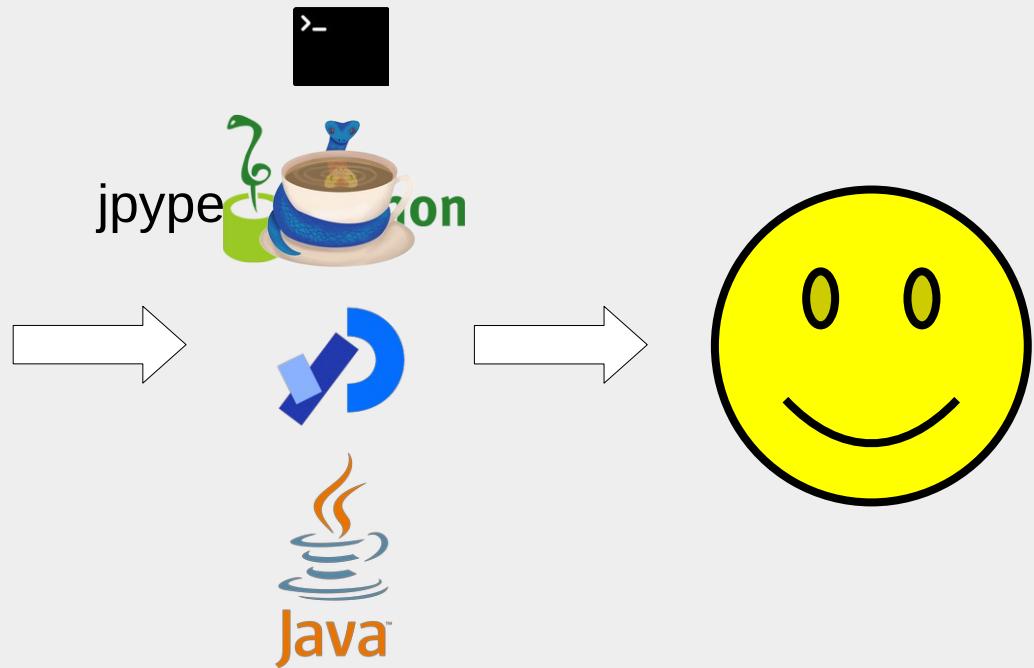
sketch_210819a.py X

home > nuc > Desktop > sketch_210819a.py

1 |

python code

Ln 1, Col 1 Spaces: 4 UTF-8 LF Python ⚡



./py5/thonny_plug-in

The screenshot shows the Thonny Python IDE interface. The main window title is "Thonny - /home/nuc/Desktop/demo.py @ 6:35". The menu bar includes File, Edit, View, Run, Tools, Help, and py5. A toolbar below the menu has icons for file operations, run, and stop. An open editor tab is titled "demo.py" with the following code:

```
1 def setup():
2     size(300, 200)
3     rect_mode(CENTER)
4
5 def draw():
6     rect(mouse_x, mouse_y, 10, 10)
```

Below the editor is a "Shell" tab containing a Python 3.9.5 session transcript:

```
Python 3.9.5 (bundled)
>>> %cd /home/nuc/Desktop
>>> %Run /home/nuc/.local/lib/python3.9/site-packages/py5_tools/tools/run_sketch.py /home/nuc/Desktop/demo.py
>>> %cd /home/nuc/Desktop
>>> %Run /home/nuc/.local/lib/python3.9/site-packages/py5_tools/tools/run_sketch.py /home/nuc/Desktop/demo.py
```

A separate window titled "Sketch" is visible in the background, showing a blank white canvas with a small black square at the center, representing the current state of the Py5 sketch.



JupyterLab — Mozilla Firefox

File Edit View Run Kernel Tabs Settings Help

https://hub.gke2.mybinder.org/user/hx2a-py5examples-m380kyfm/lab/tree/examples/introduction to py5bot.ipynb

Introduction to py5bot.ipynb

py5bot

This is py5bot. A simple and easy to use programming environment for teaching the very basics of Python programming and creative coding with py5.

Each cell in this notebook can contain a series of py5 drawing commands. The drawing commands will be executed to create a static image that will be embedded in the notebook.

The main design goal is to provide a simple programming environment that is suitable for beginners. Individual programming concepts can be explained in isolation from more complicated Python concepts like functions or modules.

When hosted on Jupyter Hub and paired with Jupyter Lab's "Show Contextual Help" feature, py5bot can become an easy to use programming environment for educators to teach Python to beginners.

[launch binder](#)

Below is a simple example.

```
[1]: size(200, 200)
background(255, 255, 0)
rect(50, 50, 100, 100)
```

[1]:

A yellow square with a white rectangle centered inside it, representing the output of the provided code cell.

https://mybinder.org/v2/gh/hx2A/py5examples/HEAD?urlpath=lab

./end

end