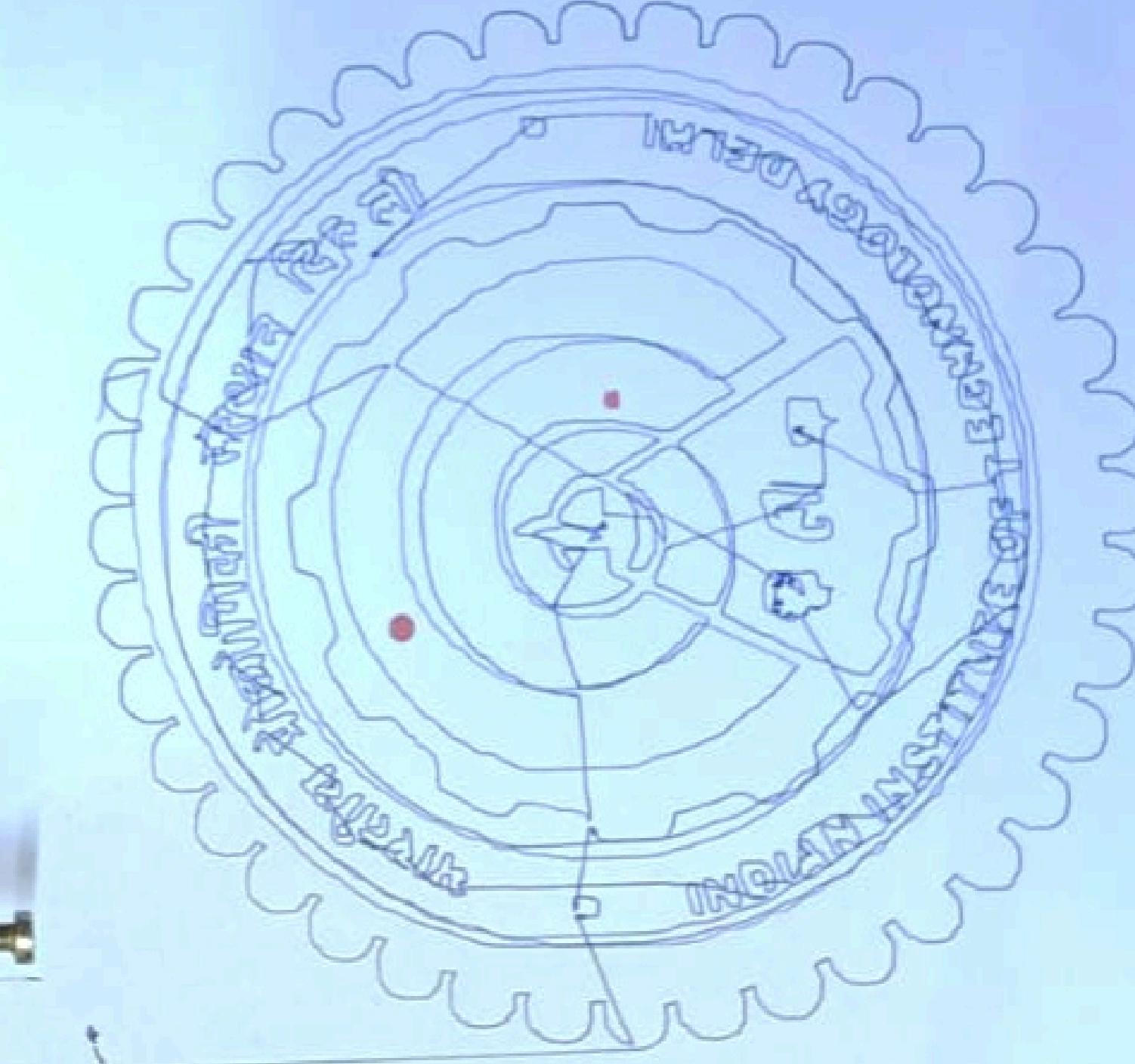
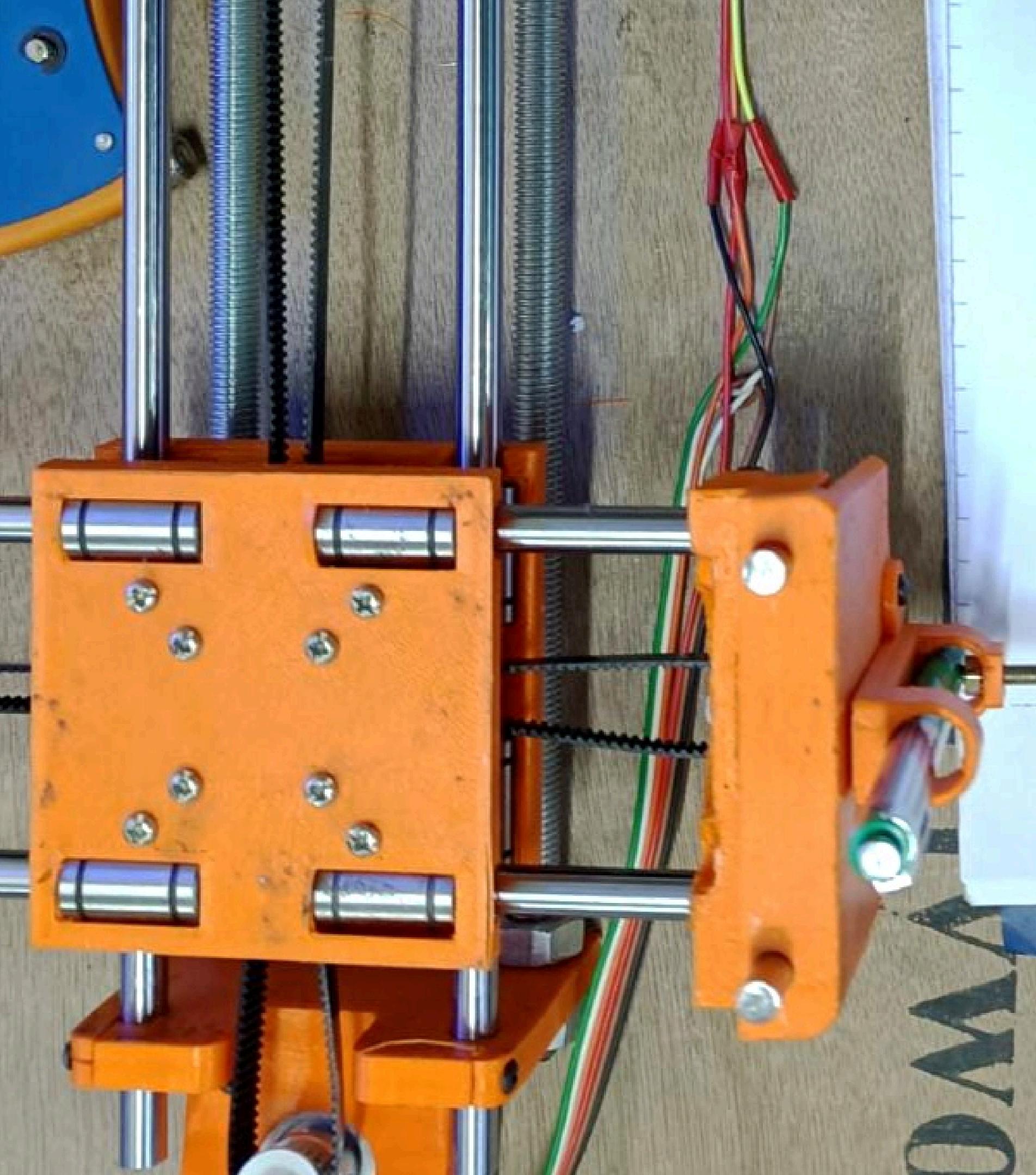


CNC 2-D Plotter



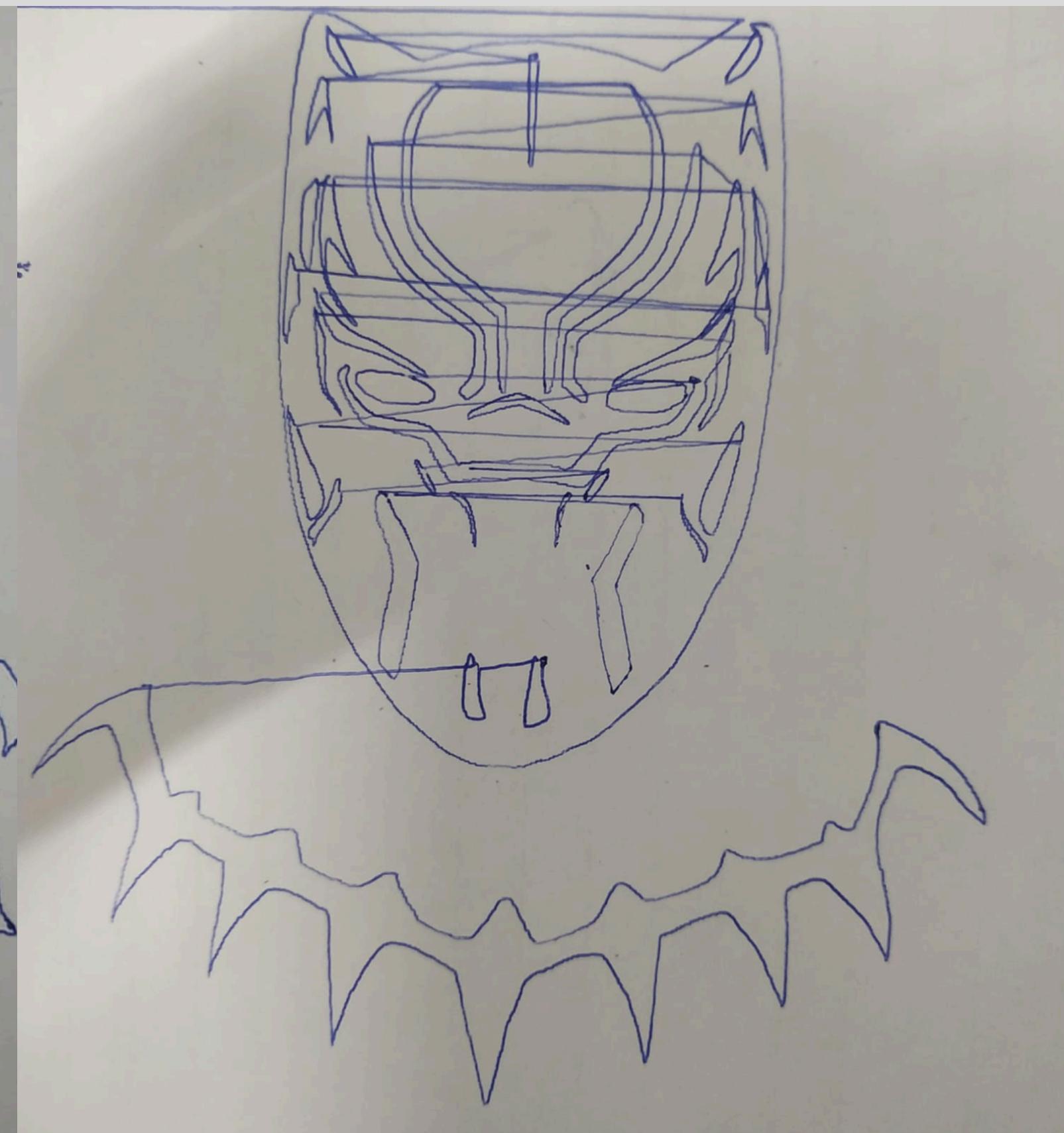
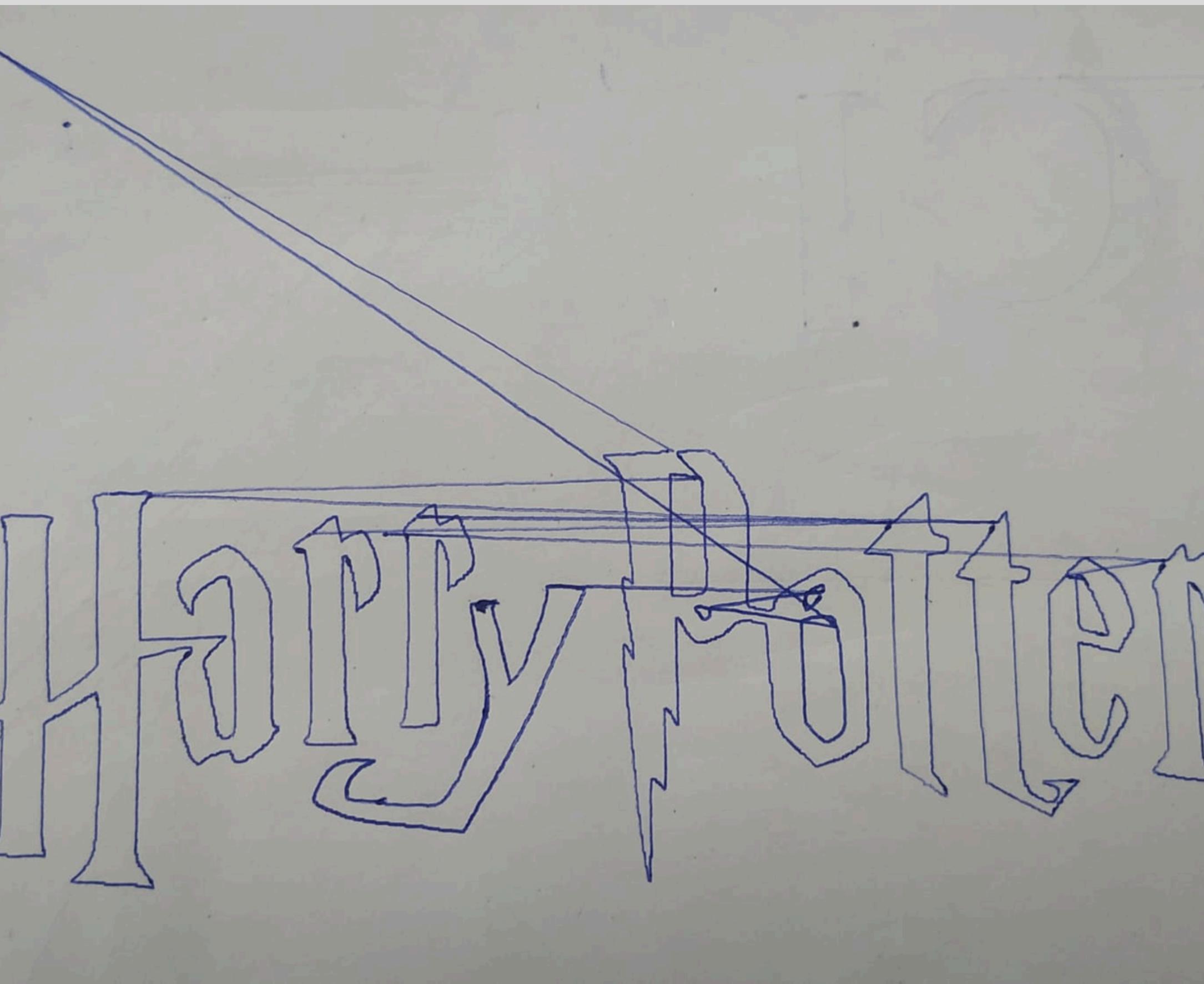
Team Members:

**Shyam
Juhi
Vibhor
Shefali
Priti
Arpit**



DOWN

Some of our Plots





Components used

Linear Shafts 8mm*400mm - 2pcs

Linear Shafts 8mm*320mm - 2pcs

Linear Shaft 6mm*200mm - 1pcs

LM6UU Bearings - 2pcs

LM8UU Bearings - 8 pcs

16 Teeth 5mm Bore Timing Pulleys - 2 pcs

16 Teeth 3mm Bore Idler Pulleys - 1 pcs

No Teeth 16 3mm Bore Idler Pulleys - 4 pcs

GT2 Belt - 2m

Nema 17 Stepper Motors - 2 pcs

Arduino + CNC Shield + A4988 Drivers - 1 pcs

28BYJ-48 12V Stepper Motor - 1 pcs

M10*400mm Threaded Rods - 2 pcs

M10 Nuts - 8 pcs

M3*8mm Bolts - 11 pcs

M3*20 Bolts - 10 pcs

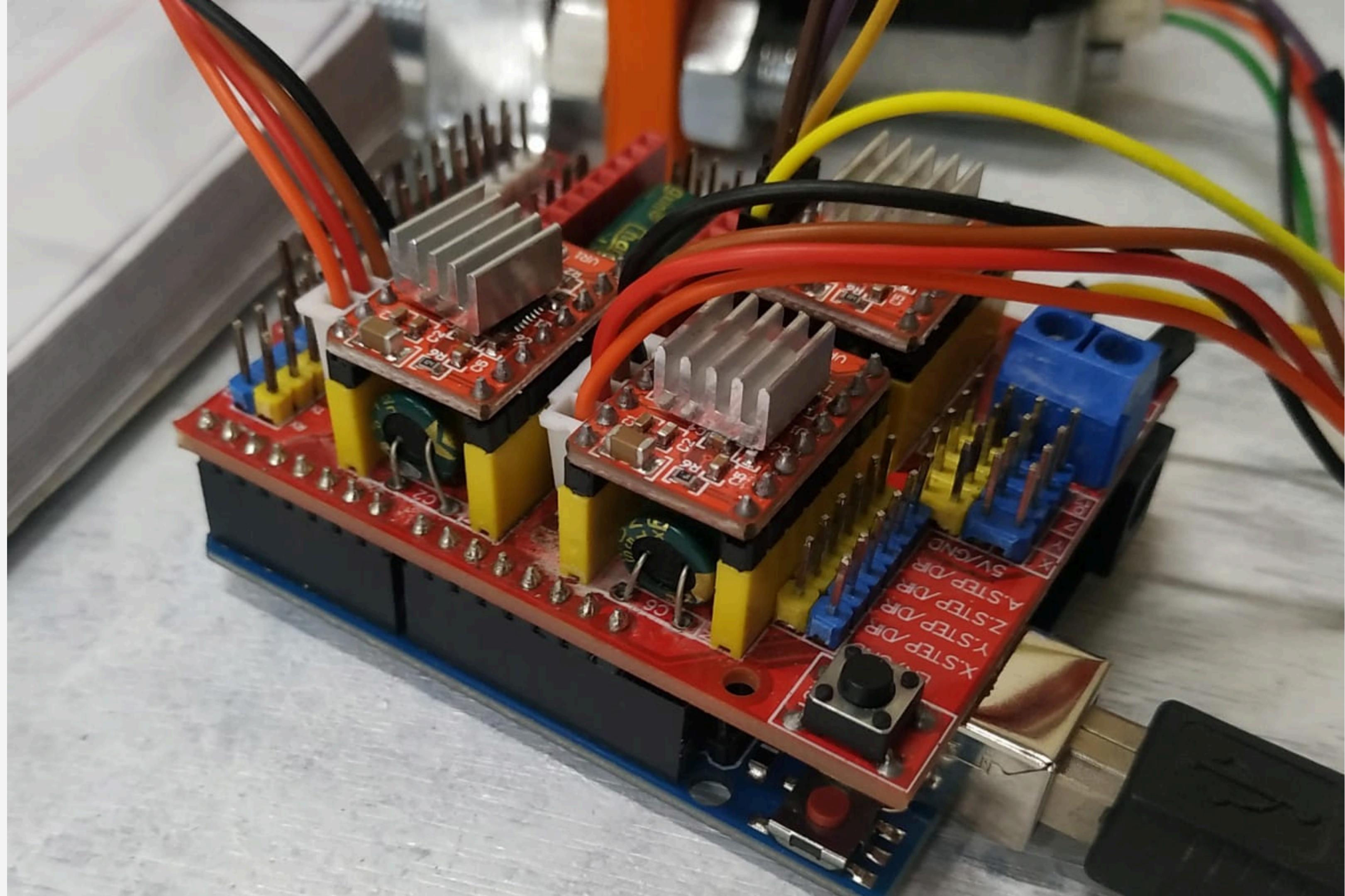
M3*30 Bolts - 8 pcs

M3 Nuts - 22 pcs

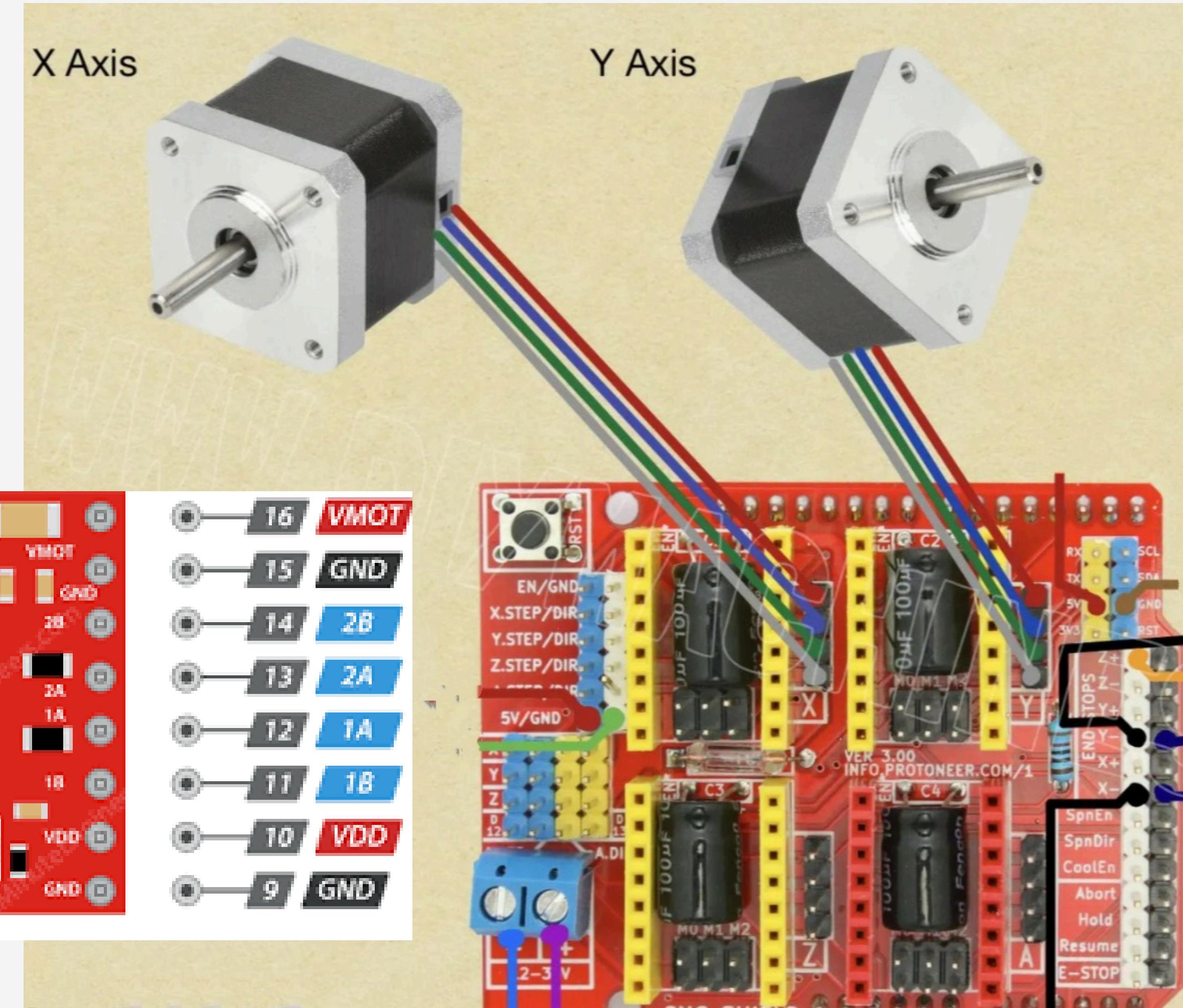
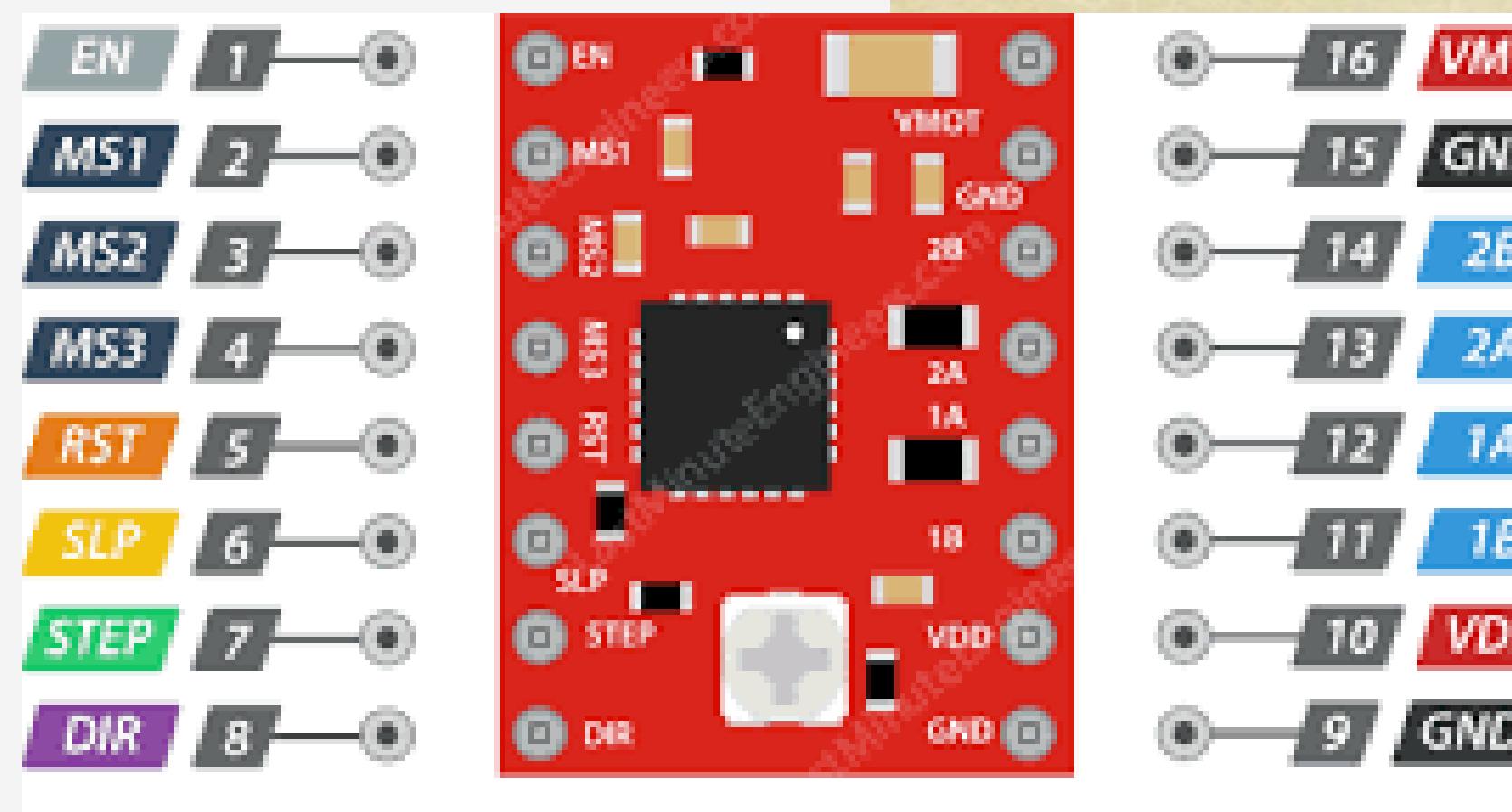


CIRCUIT CONNECTIONS





Reference circuit



Universal G code Sender

A full featured gcode platform used for interfacing with advanced CNC controllers like GRBL , TinyG, g2core and Smoothieware. Universal Gcode Sender is a self-contained Java application which includes all external dependencies and can be used on most computers running Windows, MacOSX or Linux.

Universal G Code Sender

The screenshot shows the Universal G Code Sender software interface. The top menu bar includes FILE, EDIT, MACHINE, PROGRAM, TOOLS, VISUALIZER, WINDOW, and HELP. The toolbar contains icons for file operations, firmware selection (GRBL), port selection (COM3), baud rate (115200), and various control functions like play/pause, stop, and lock.

The main window features several tabs: Controller State (DRO), Welcome Page, black-avengers-png-logo-2.gcode (selected), and Visualizer.

The Controller State (DRO) tab displays digital readouts for X0, Y0, and Z0 axes, all showing 0.000. It also shows FEED RATE 0, SPINDLE 0, and an ALARM button.

The Source tab shows the G-code script:

```
1 (Scribbled version of C:\Users\HP\AppData\Local\Temp\ink_ext_XXXXXX.scr
2 ( unicorn.py --tab="plotter_setup" --pen-up-angle=50 --pen-down-angle=150
3 G21 (metric ftw)
4 G90 (absolute mode)
5 G92 X0.00 Y0.00 Z0.00 (you are here)
6
7 M300 S30 (pen down)
8 G4 P150 (wait 150ms)
9 M300 S50 (pen up)
10 G4 P150 (wait 150ms)
11 M18 (disengage drives)
12 M01 (Was registration test successful?)
13 M17 (engage drives if YES, and continue)
14
15 (Polyline consisting of 2 segments.)
16 G1 X-68.21 Y-117.17 F3500.00
17 M300 S30.00 (pen down)
18 G4 P150 (wait 150ms)
19 G1 X-84.71 Y-116.79 F3500.00
```

The Visualizer tab displays a 3D coordinate system with X+, Y-, and Z+ axes. A grid is shown, and a yellow marker indicates the current position of the plotter head at approximately (X: -84.71, Y: -116.79, Z: 236.42 mm). A red horizontal line segment is also visible.

The bottom left shows the Jog Control panel with buttons for X-, Y+, X+, Y-, Z+, and Z-. Step size controls (Step size XY: 10, Step size Z: 1, Step size ABC: 1) and feed rate controls (Feed rate: 5,000) are also present.

The bottom right shows the FPS: 9.2 indicator.

Further Improvements to be made

- Loose connections have to be removed.
- Belt tension is not enough
- We have to adjust the pen holder for Z- axis to be functional.
- Plane of drawing is not perfectly horizontal.

References

We have programmed CNC 2D plotters using grbl (downloaded for github). We have used inkscape to generate G-code files of the images we wish to plot/draw. We have used Universal G-Code Sender to upload the G-code files and direct our plotter to draw it.

We have assembled our CNC 2D Plotter using the following resources.

We had to 3D print various parts required for this machine.

Youtube Link >>>> <https://youtu.be/qanuOqAqHyo>

Resource link (for parts & reference) >>>> <http://www.thingiverse.com/thing:1514145>

Thank you