

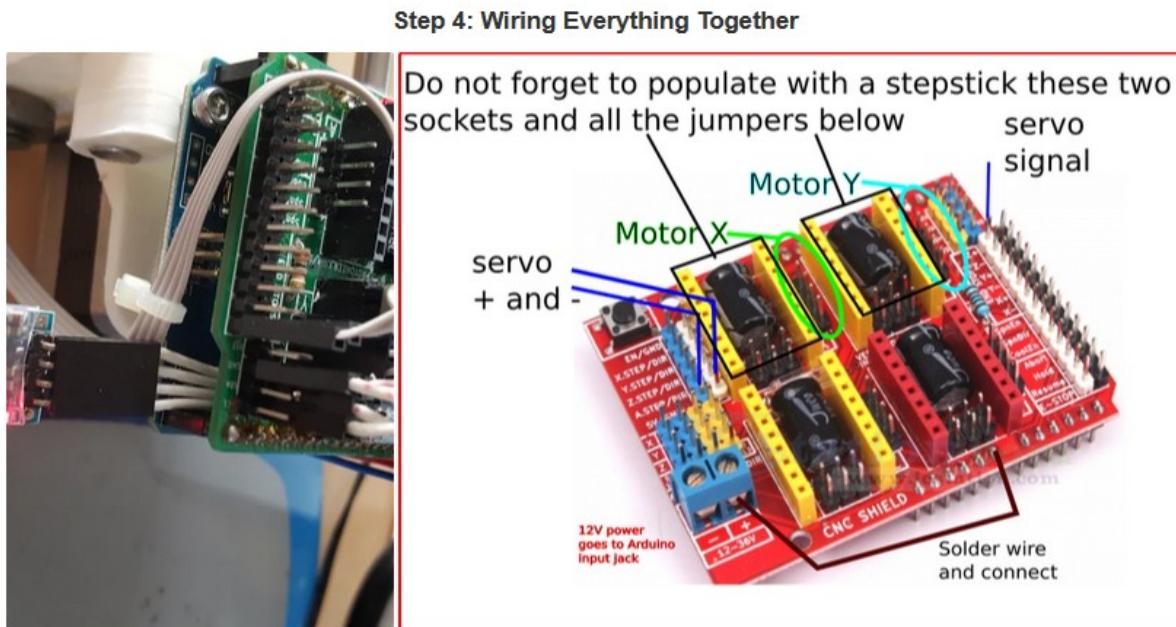
# General Information and Connections

## Table des matières

|  |    |
|--|----|
| 1 Control & connections.....                     | 2  |
| 1.1 CNC Shield Guide.....                        | 3  |
| 1.1.1 Stepper Motor Step Resolution Jumpers..... | 4  |
| 1.2 Servo.....                                   | 5  |
| 1.2.1 Servo program features.....                | 5  |
| 1.3 Stepper Motor.....                           | 6  |
| 1.3.1 Stepper Motor connections.....             | 7  |
| 2 Limit Switches.....                            | 8  |
| 2.1 Connections.....                             | 8  |
| 2.2 Filtering.....                               | 10 |
| 3 Push buttons.....                              | 11 |
| 3.1 Connections.....                             | 11 |
| 3.2 Actions.....                                 | 13 |

# 1 Control & connections

File : <https://www.instructables.com/4xiDraw/>



Before inserting the CNCShield over the Arduino you want to do [this trick](#), that will allow to power everything from the Arduino power jack. Failing to do this connection from Vin to + header on CNCShield most likely will make your servo not to work properly.

On top of Arduino you insert the CNCShield board and on top of it, two of the Pololu StepStick stepper driver boards. But before inserting these two boards for axis X and Y, make sure you put three jumpers in the headers (that will later be obstructed by the Pololu carrier boards).

Two four-wire cables come from the stepper motors.

Each stepper motor goes to X and Y axis four pin headers on the CNCShield.

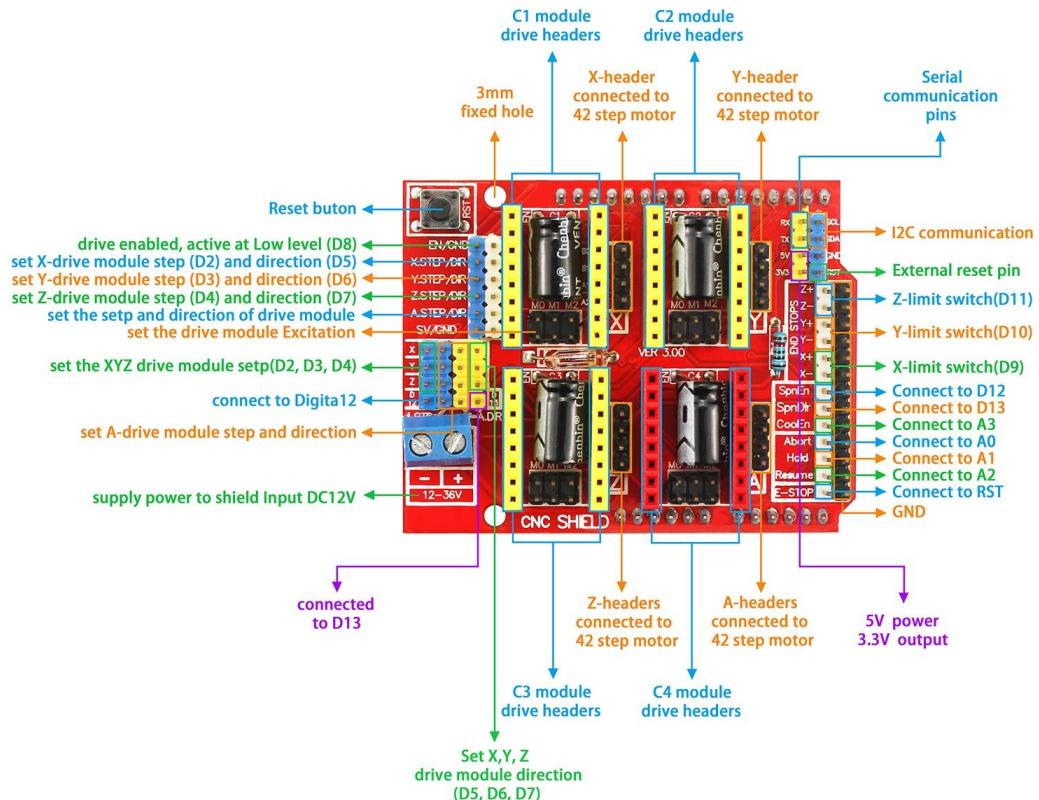
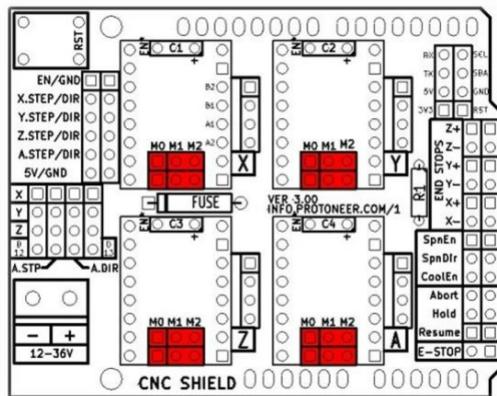
*There is an optional improvement: make the plotter wireless by adding a Bluetooth module, but I would only do this once everything else is up and running.*

## 1.1 CNC Shield Guide

File : CNC-Shield-Guide-v1.0.pdf

Below is a general outline process for connection of the various components:

- Taking normal static electricity precautions, insert the CNC Shield into the Arduino Uno making sure the correct pins of the CNC Shield are inserted into the correct UNO headers.
- Decide on the micro stepper setting for your application and place the jumpers as required



### 1.1.1 Stepper Motor Step Resolution Jumpers

In the table below High indicates that a Jumper is insert and Low indicates that no jumper is inserted.

[DRV8825](#) Stepper Driver configuration:

| M0   | M1   | M2   | Microstep Resolution |
|------|------|------|----------------------|
| Low  | Low  | Low  | Full step            |
| High | Low  | Low  | Half step            |
| Low  | High | Low  | 1/4 step             |
| High | High | Low  | 1/8 step             |
| Low  | Low  | High | 1/16 step            |
| High | Low  | High | 1/32 step            |
| Low  | High | High | 1/32 step            |
| High | High | High | 1/32 step            |

## 1.2 Servo

It is used for moving the pencil up or down.

A three-wire cable will be coming from the servo.

Servo cable has to go to (red) +5V, (black) GND and signal (white or brown) to *Digital pin 11*. Servo cable is too short, so a 250 mm extension cable will be needed.

Use the commands M03 Sxxx (xxx between 0 and 255) to rotate the servo between 0-180.

The command M05 turns the servo to zero degrees.

### 1.2.1 Servo program features

In the program, Servo is controlled by Spindle named functions.

The actual movement of the Servo, up or down depending on the G-Code, is made in file:

*spindle\_control.c*

*spindle\_control.h*

```
define RC_SERVO_SHORT 15 // Timer ticks for 0.6ms pulse duration (9 for 0.6ms)
define RC_SERVO_LONG 32 // Timer ticks for 2.5 ms pulse duration (39 for 2.5ms)
```

For having the servo working from 0 --> 180 degrees change RC\_SERVO\_SHORT and put 9, RC\_SERVO\_LONG and put 39

For inverted movement, uncomment the line

```
##define RC_SERVO_INVERT 1 // Uncomment to invert servo direction
```

## 1.3 Stepper Motor

### Stepper Motor NEMA17

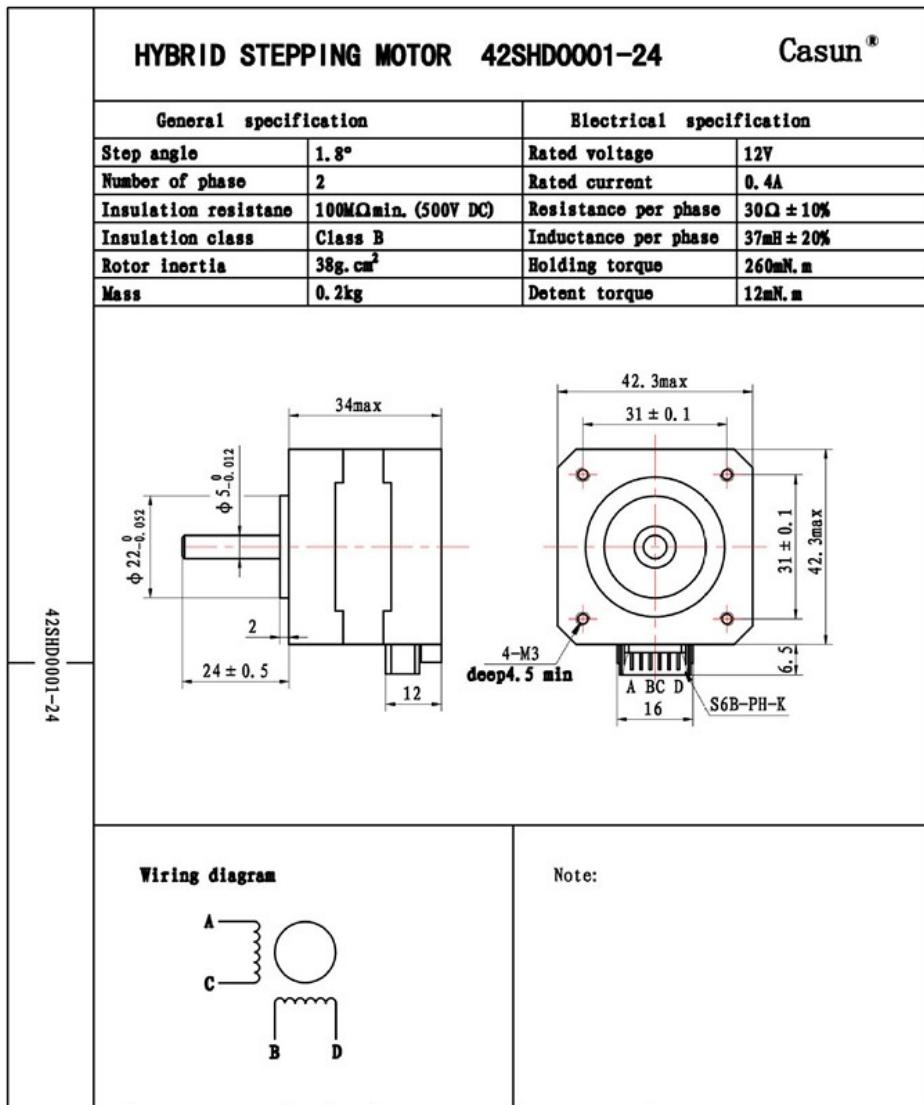
File : NEMA17-schneider.pdf

### Wiring and Connections

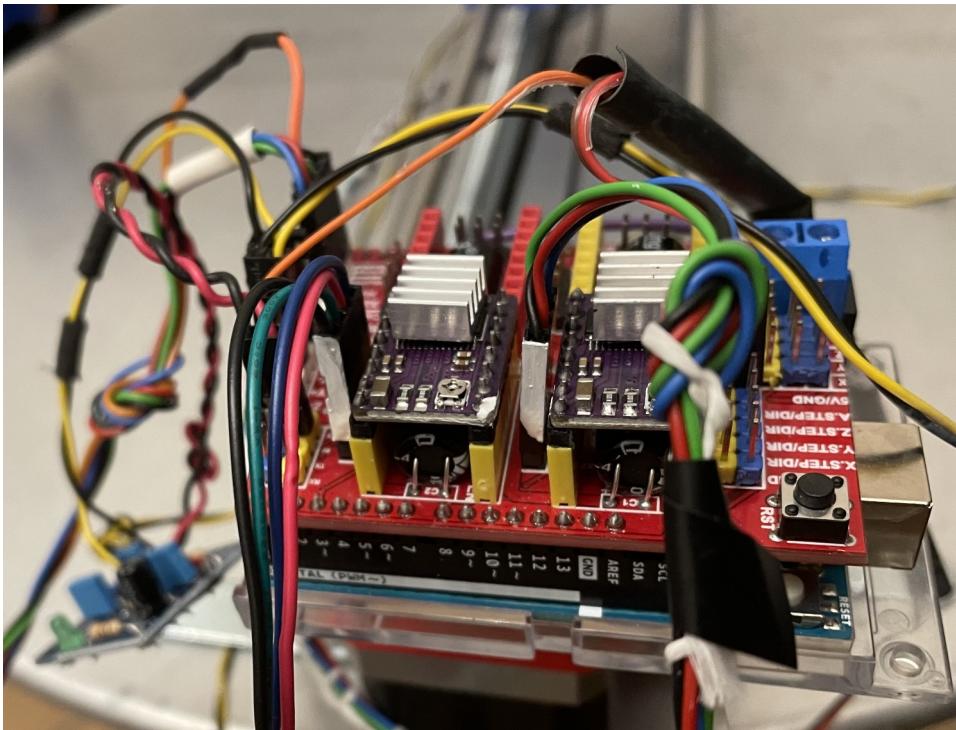
| Signals and wire colors |       |
|-------------------------|-------|
| Phase A                 | Red   |
| Phase /A                | Blue  |
| Phase B                 | Green |
| Phase /B                | Black |

### Stepper Motor 42SHD0001

File : 42SHD0001.jpeg



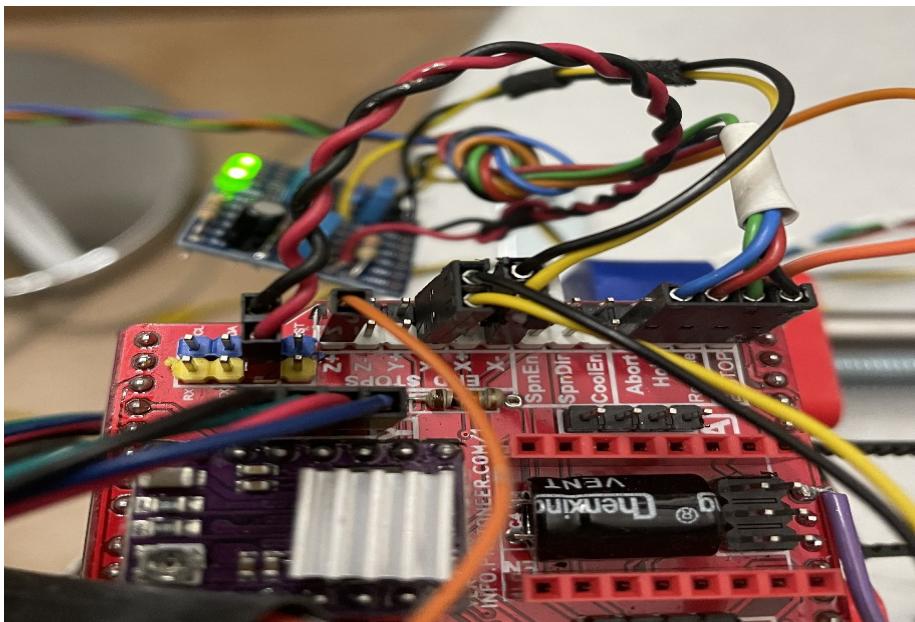
### 1.3.1 Stepper Motor connections



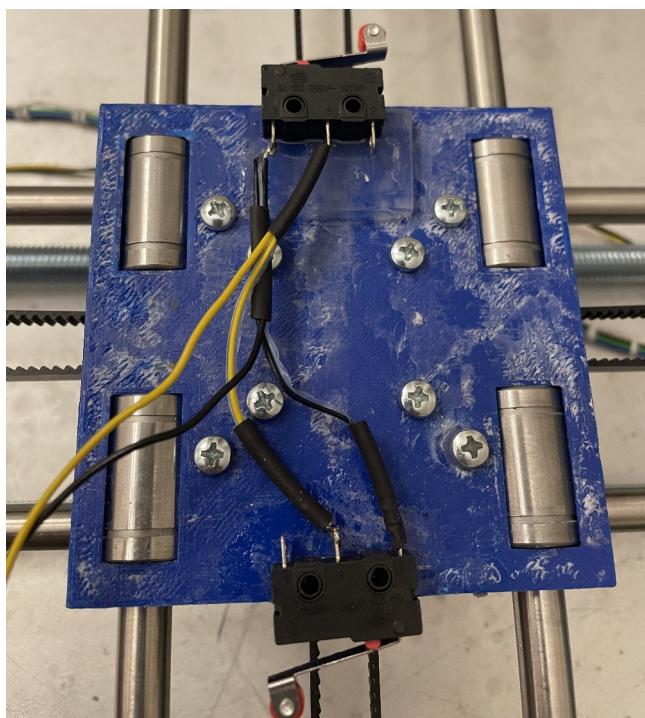
## 2 Limit Switches

### 2.1 Connections

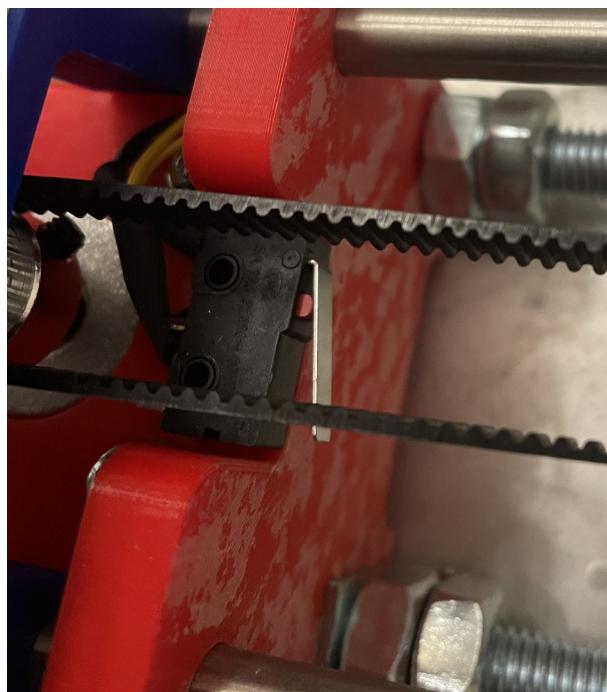
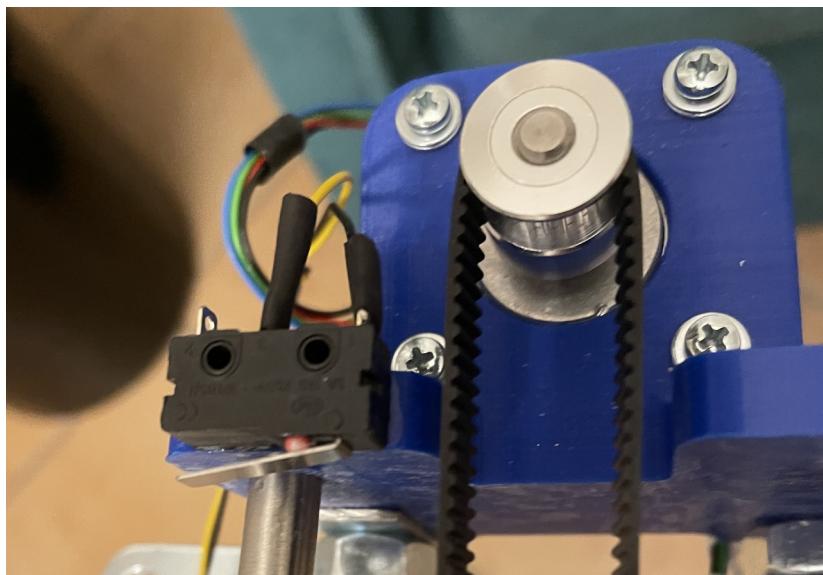
Limits switches Stops generating an Alarm when movement exceeds limits.



Y axis limits

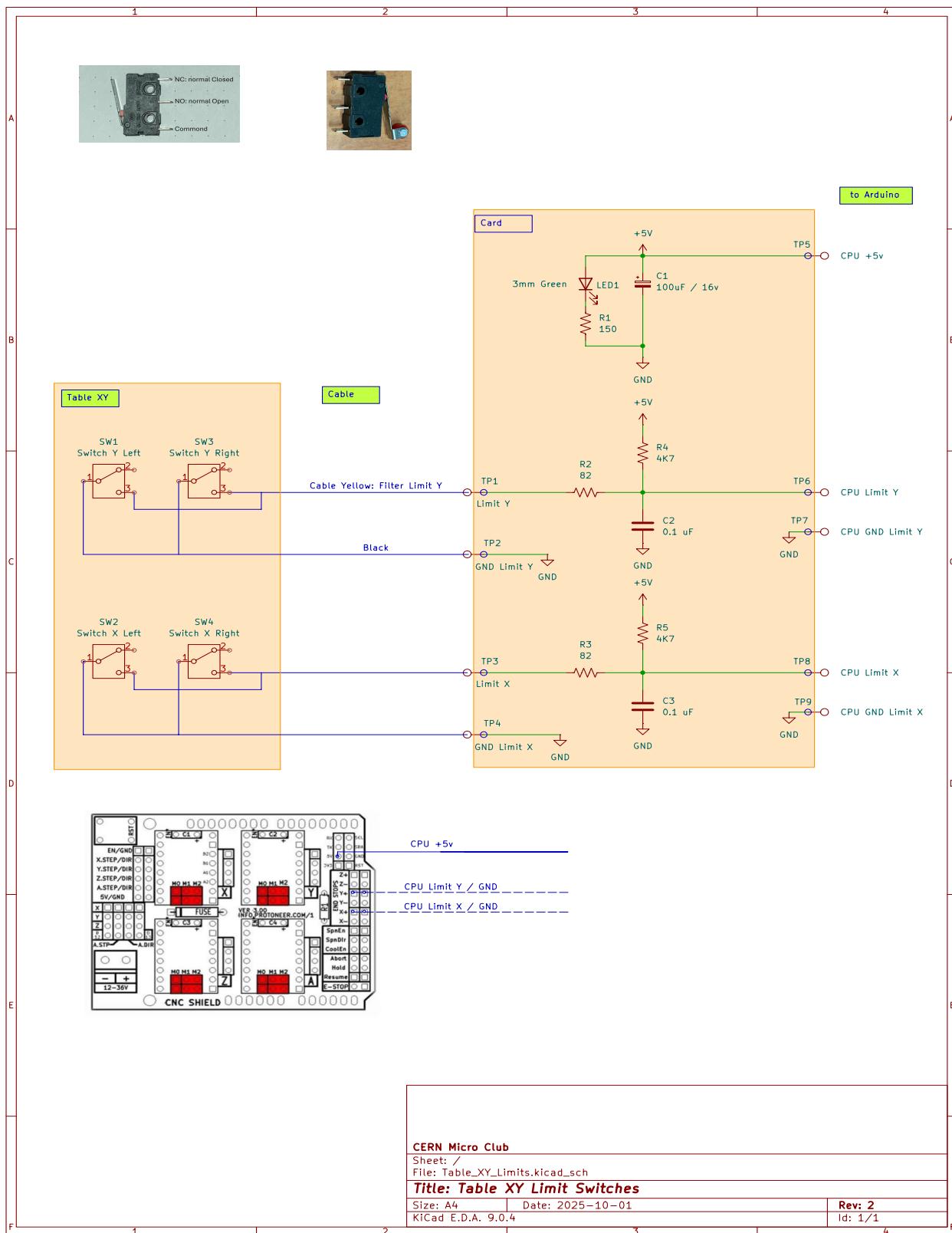


X axis limits



## 2.2 Filtering

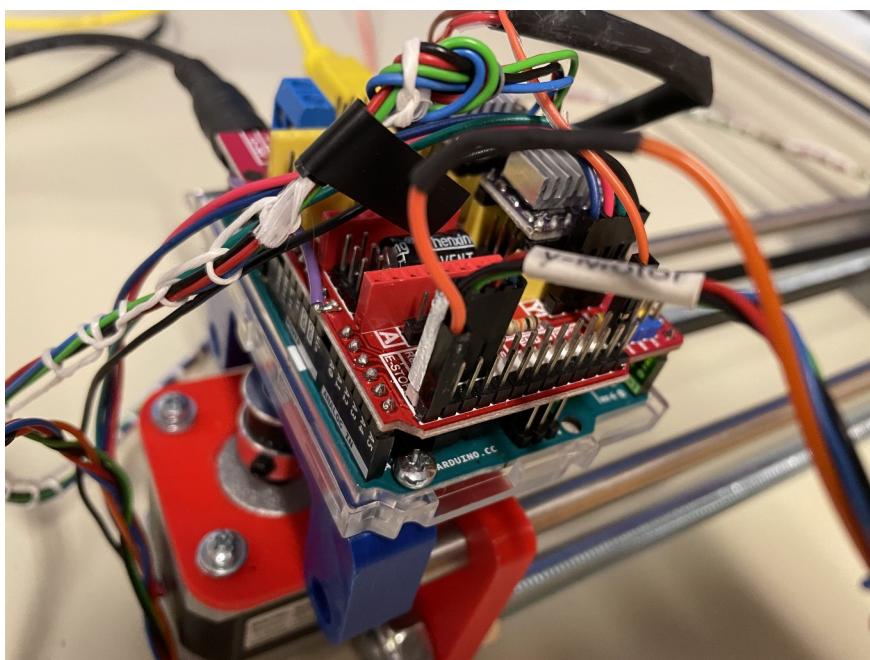
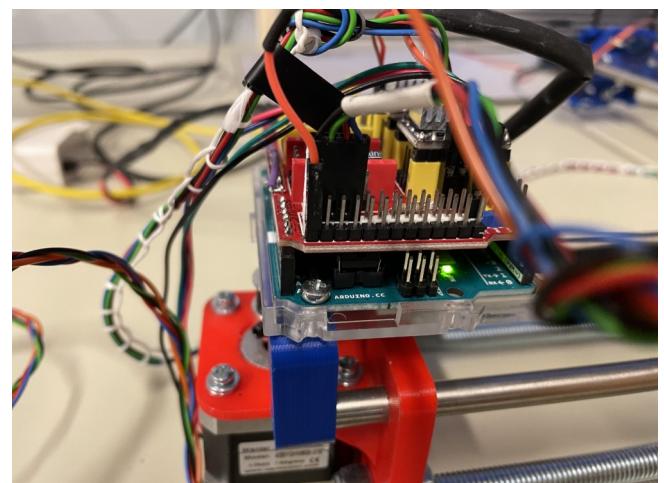
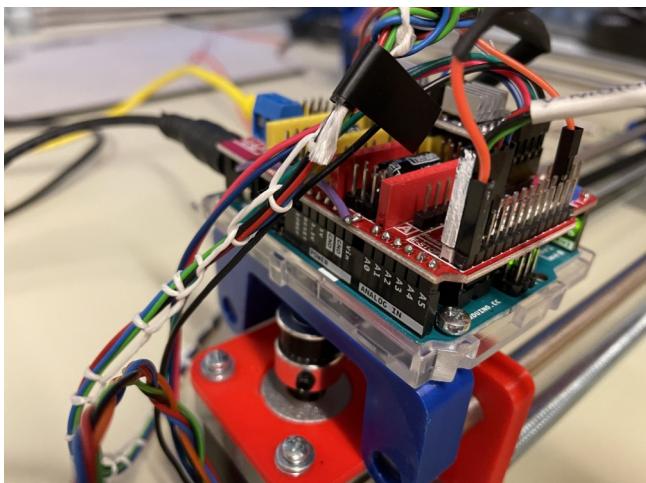
Inputs are filtered before entering the Arduino for eliminating any spurious noise.

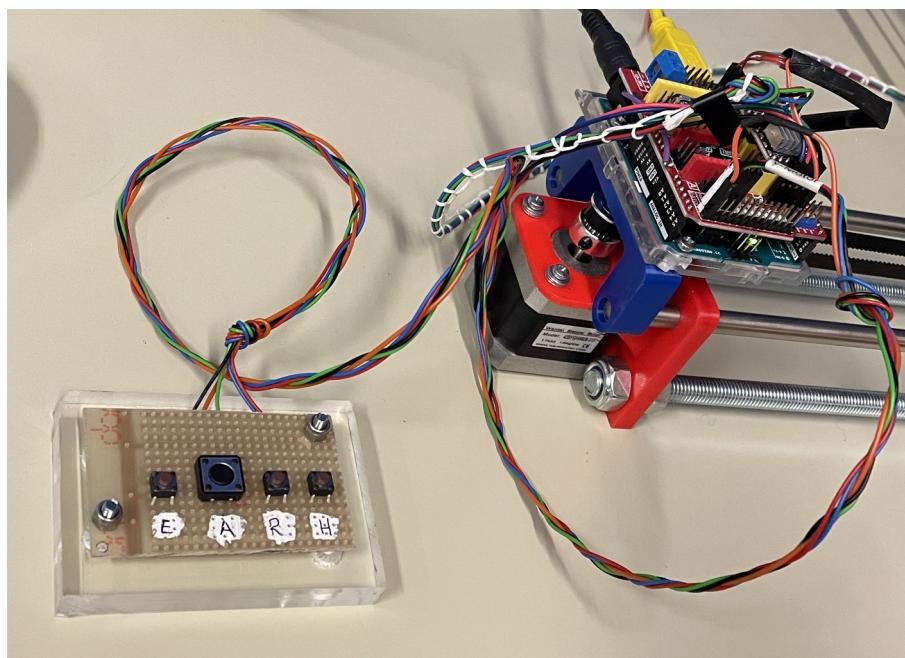
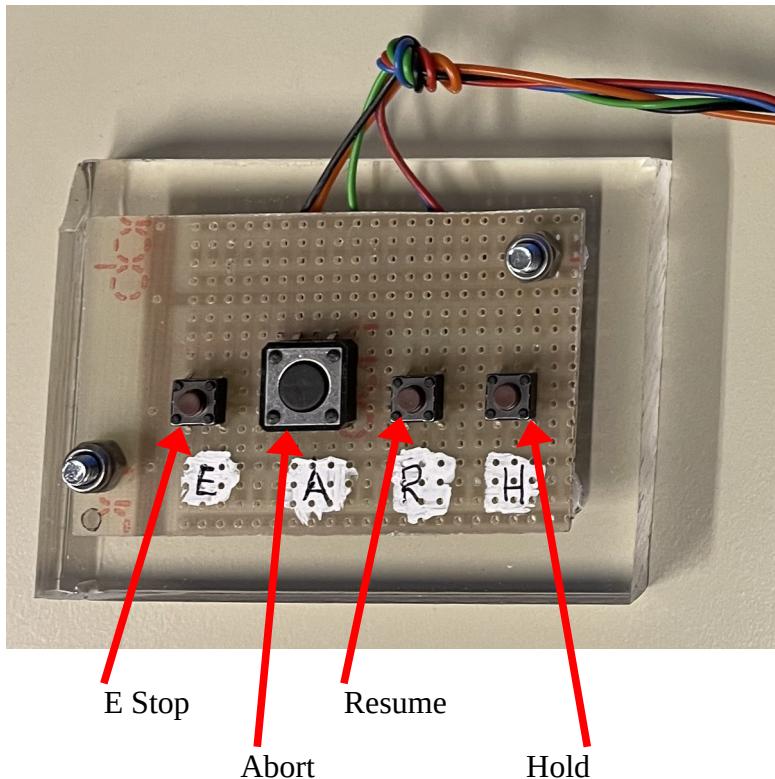


## 3 Push buttons

### 3.1 Connections

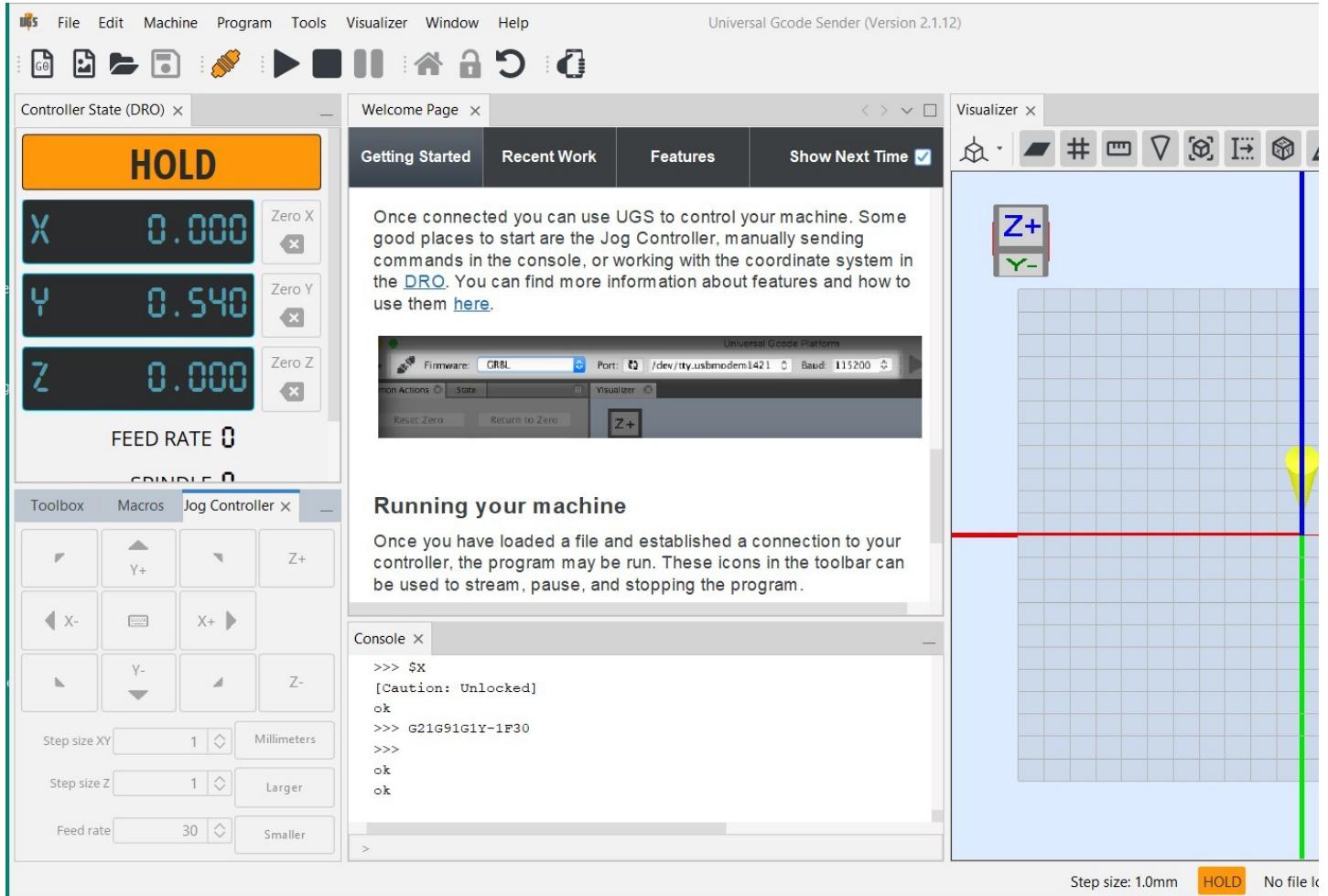
| Button        | Card Name | Cable colour                  | Description                    |
|---------------|-----------|-------------------------------|--------------------------------|
| <i>E Stop</i> | E         | Black<br>Connector white mark | In parallel with Arduino Reset |
| <i>Resume</i> | R         | Green                         | Resume printing (after Hold)   |
| <i>Hold</i>   | H         | Red                           | Hold printing                  |
| <i>Abort</i>  | A         | Blue                          | Abort present printing         |
| <i>GND</i>    |           | Orange                        |                                |





## 3.2 Actions

When *HOLD* is pressed, process continues after *RESUME* is pressed.



When **ABORT** is pressed, enter command **\$X** in *Console TX* for unlocking and sending another G File.

**\$H** is for Homing.

