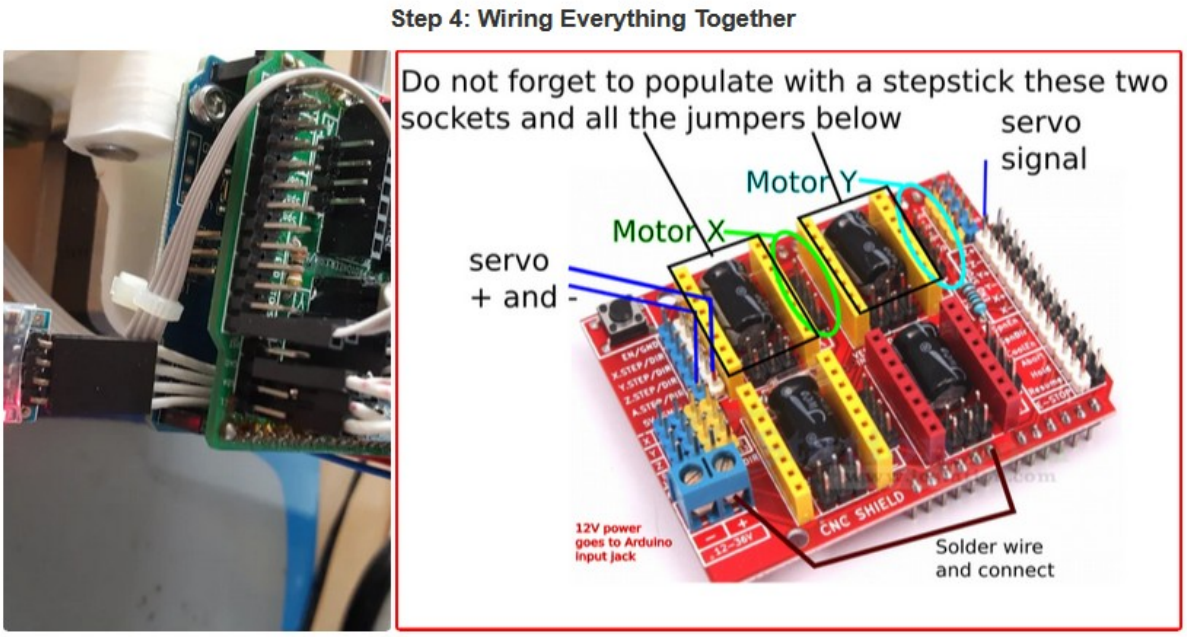
General Information and Connections

Table des matières

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.

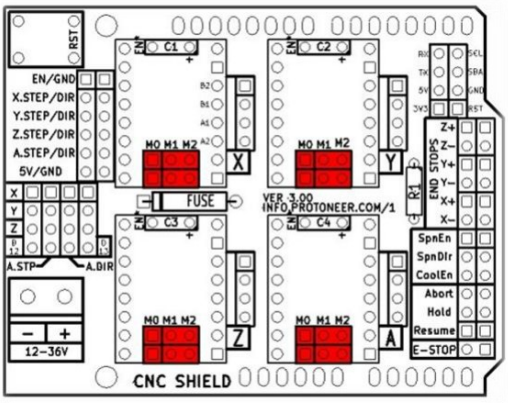
There is an optional improvement: make the plotter wireless by adding a Bluetooth module, but I would only do this once everything else is up and running.

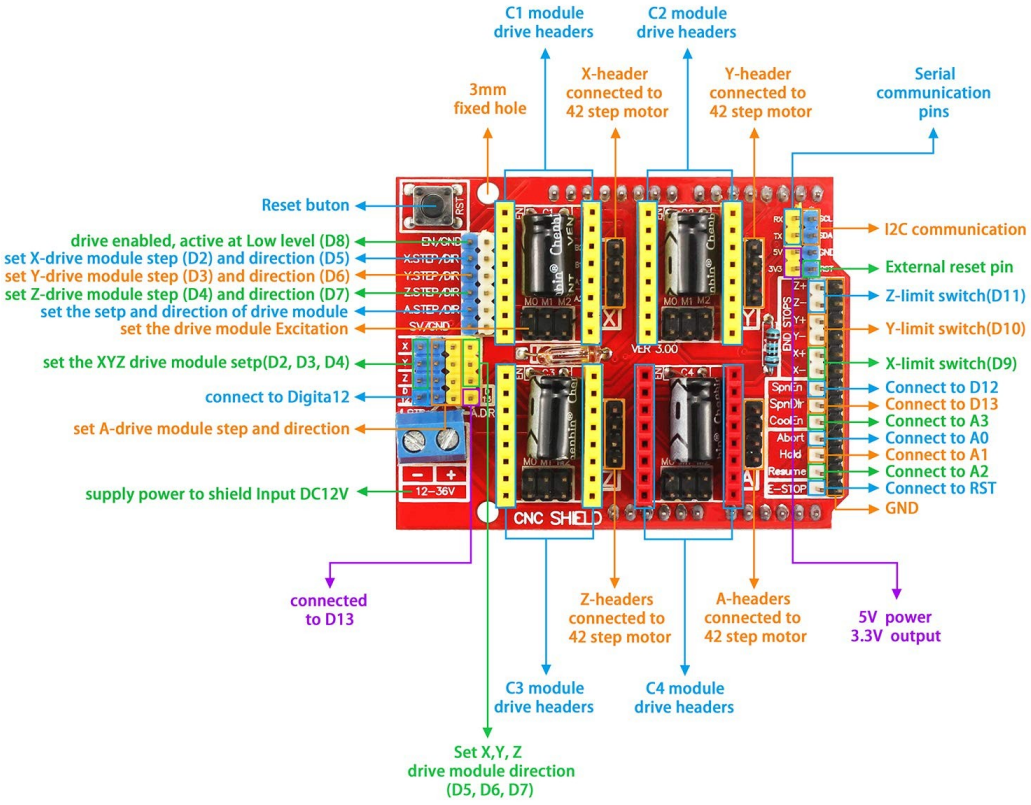
1.1 CNC Shield Guide

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*.

Commands

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

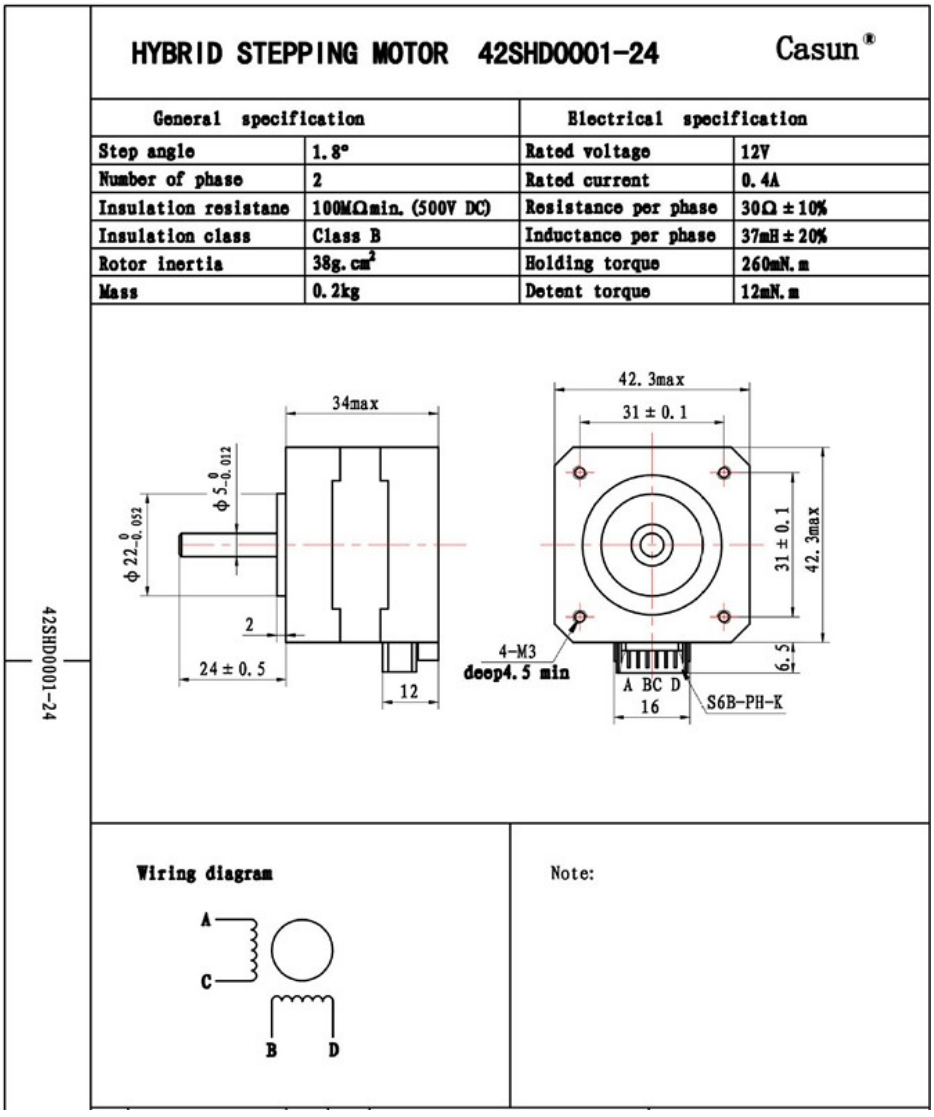
File : NEMA17-schneider.pdf

Wiring and Connections

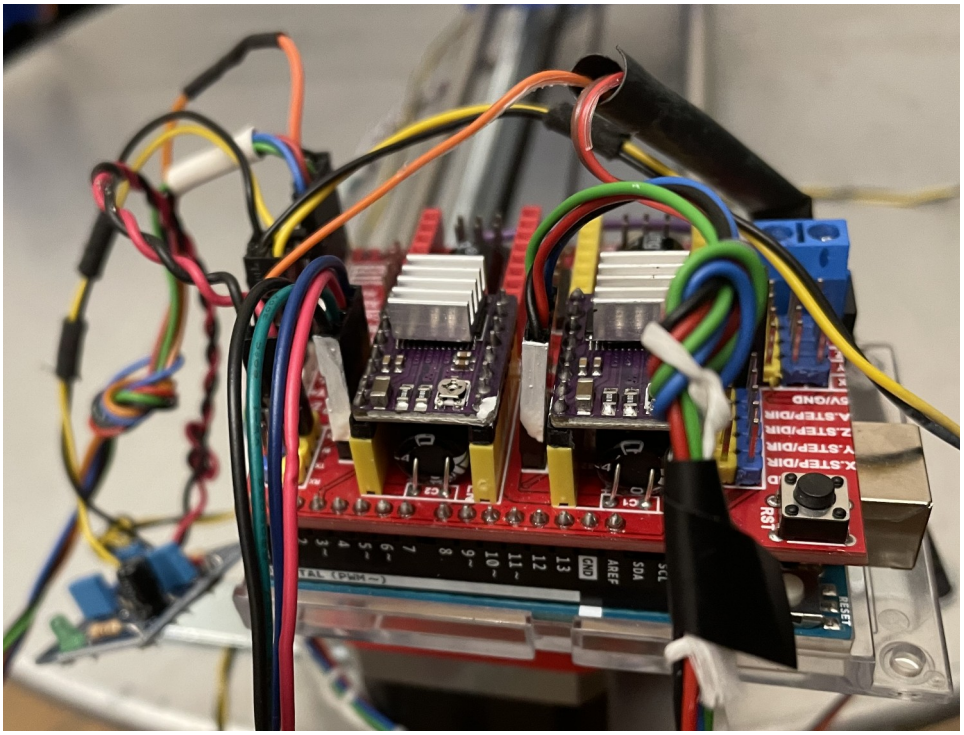
Signals and wire colors	
Phase A	Red
Phase /A	Blue
Phase B	Green
Phase /B	Black

Stepper Motor 42SHD0001

File : 42SHD0001.jpeg



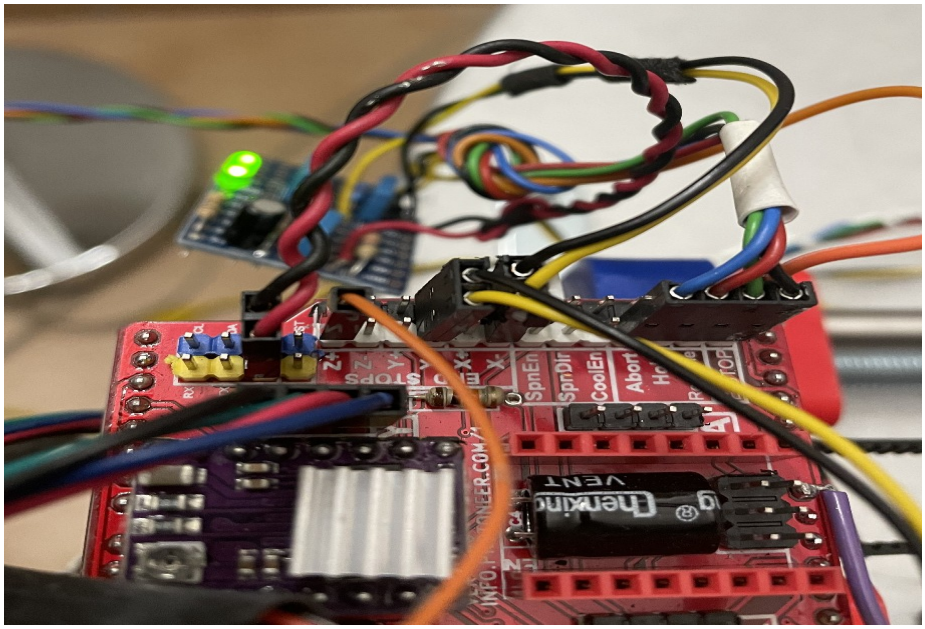
1.3.1 Stepper Motor connections



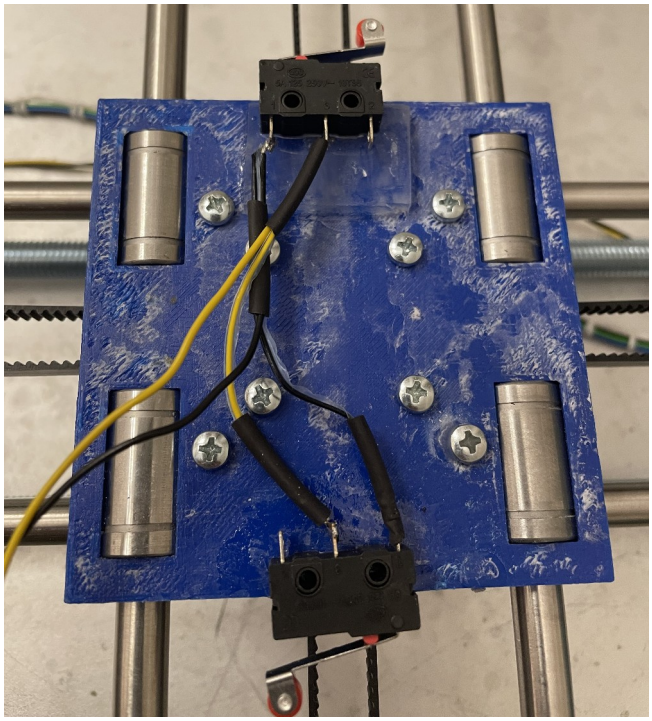
2 Limit Switches

2.1 Connections

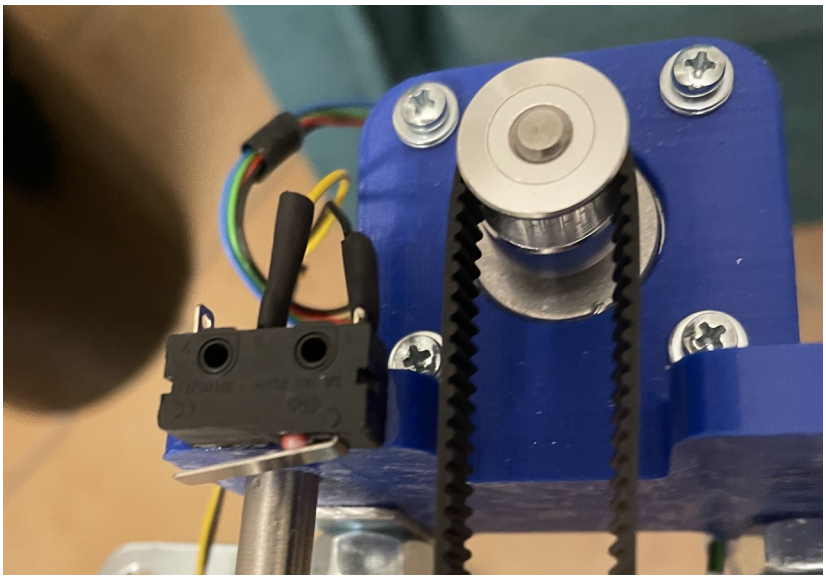
Limits switches Stops generating an Alarm when movement exceeds limits.

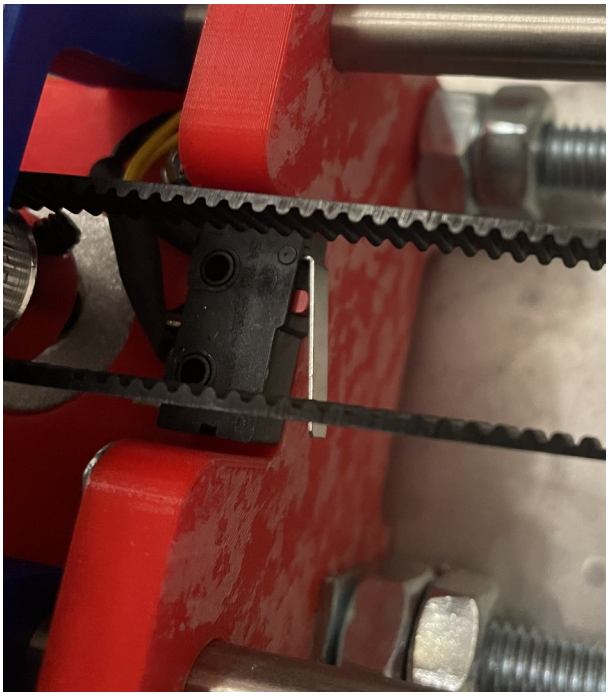


Y axis limits



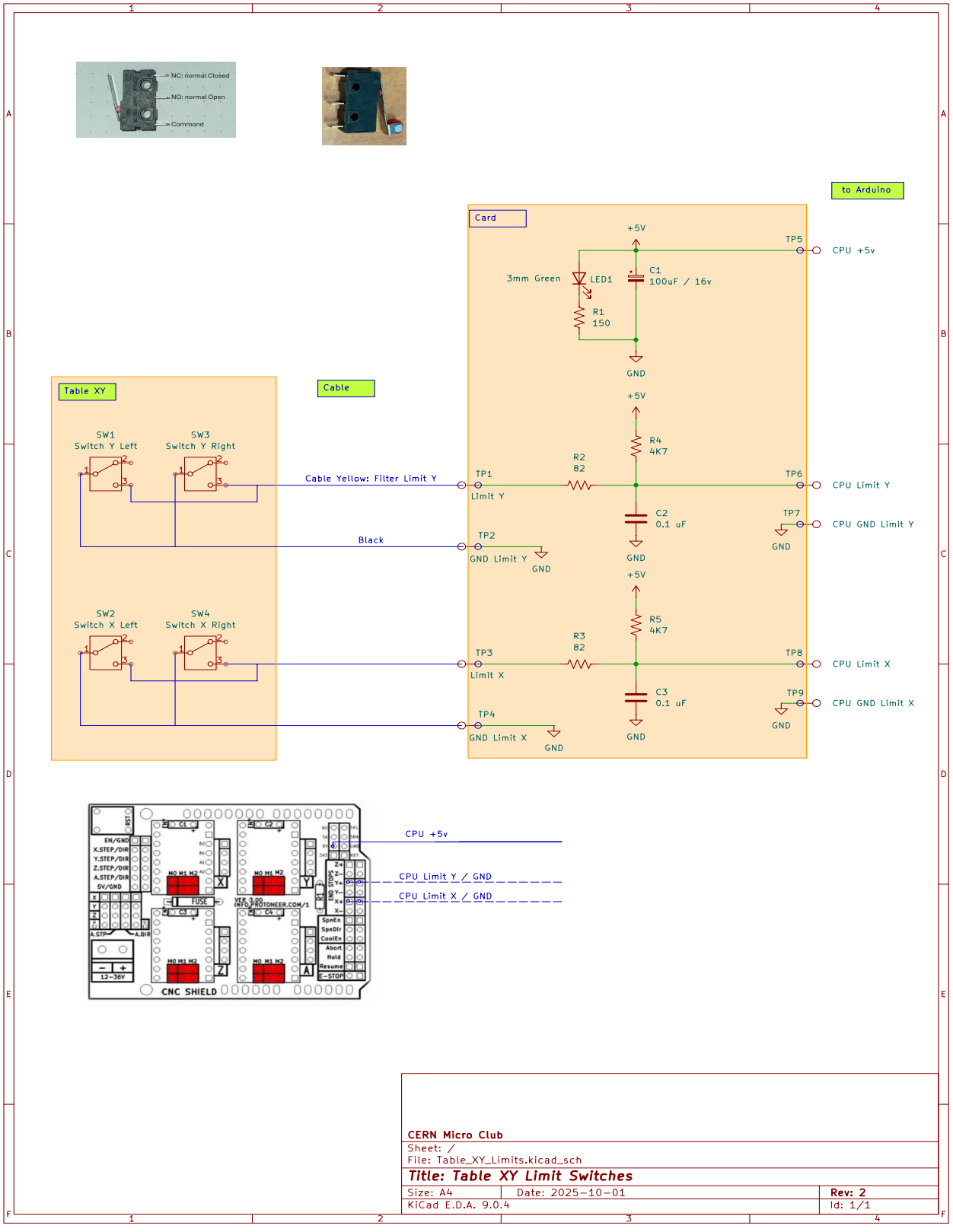
X axis limits





2.2 Filtering

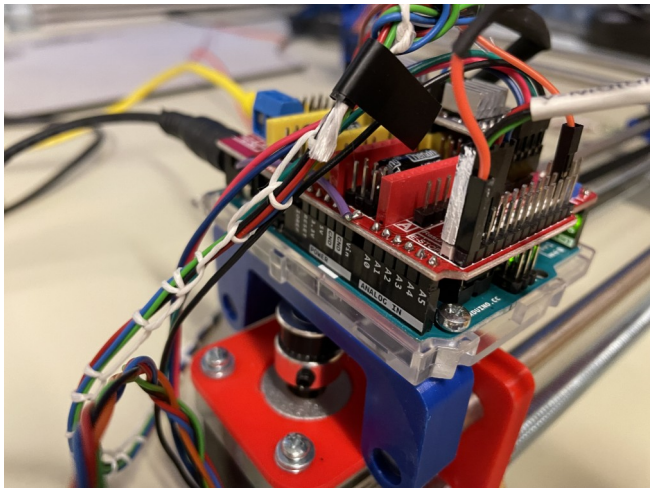
Inputs are filtered before entering the Arduino for eliminating any spurious noise.

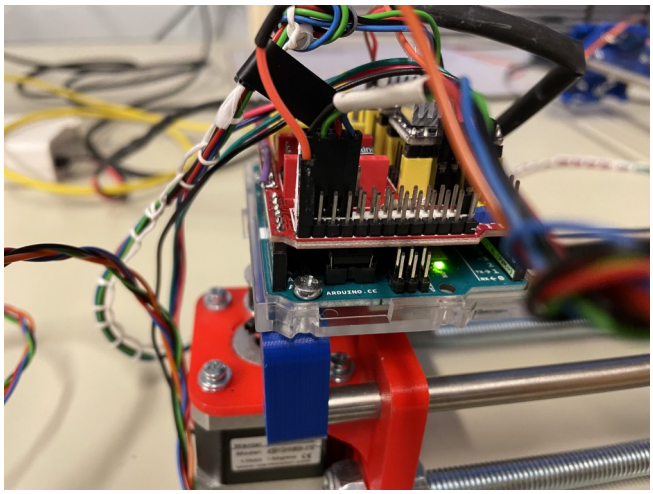


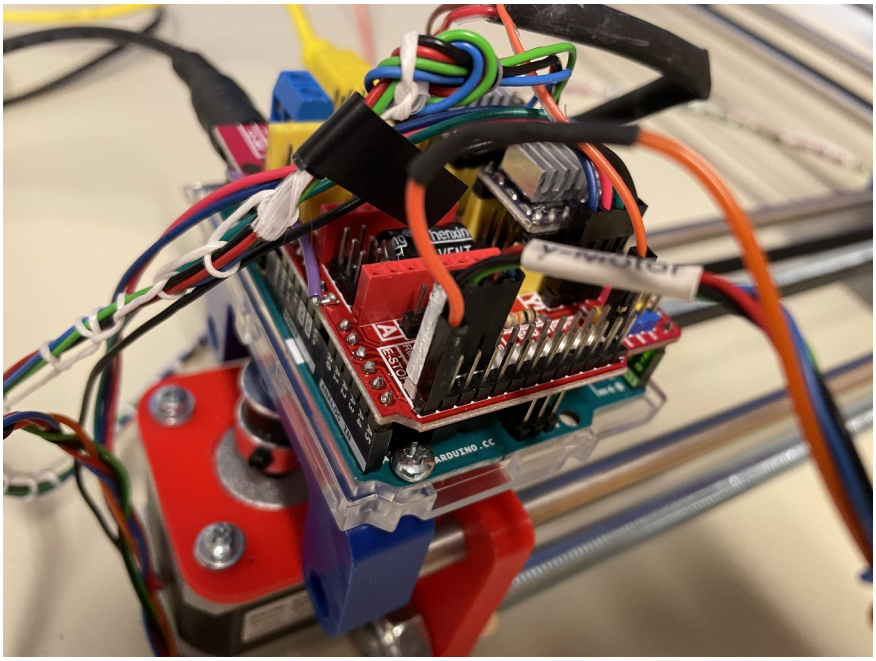
3 Push buttons

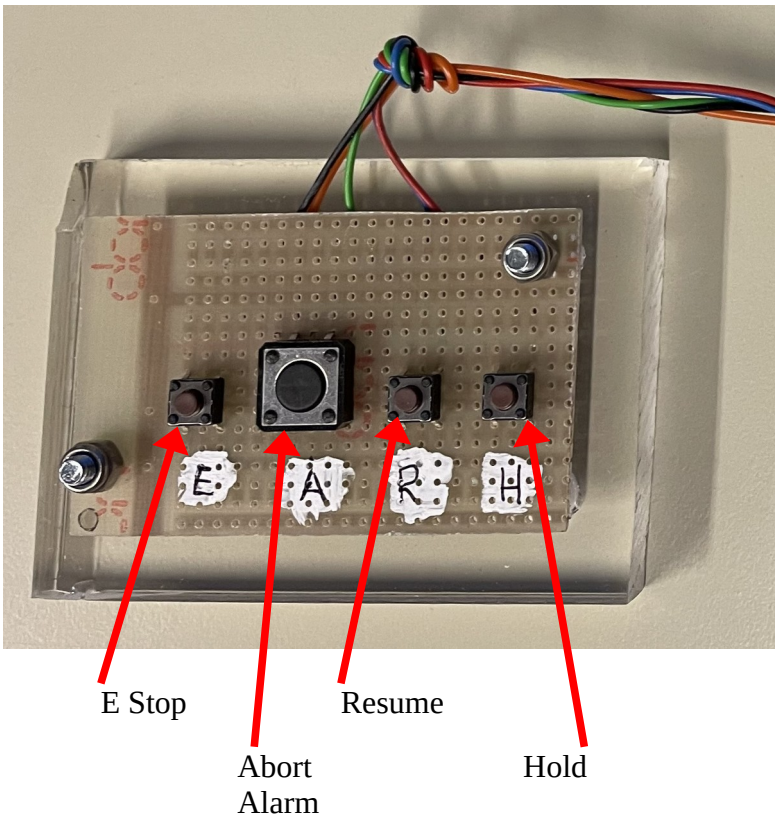
3.1 Connections

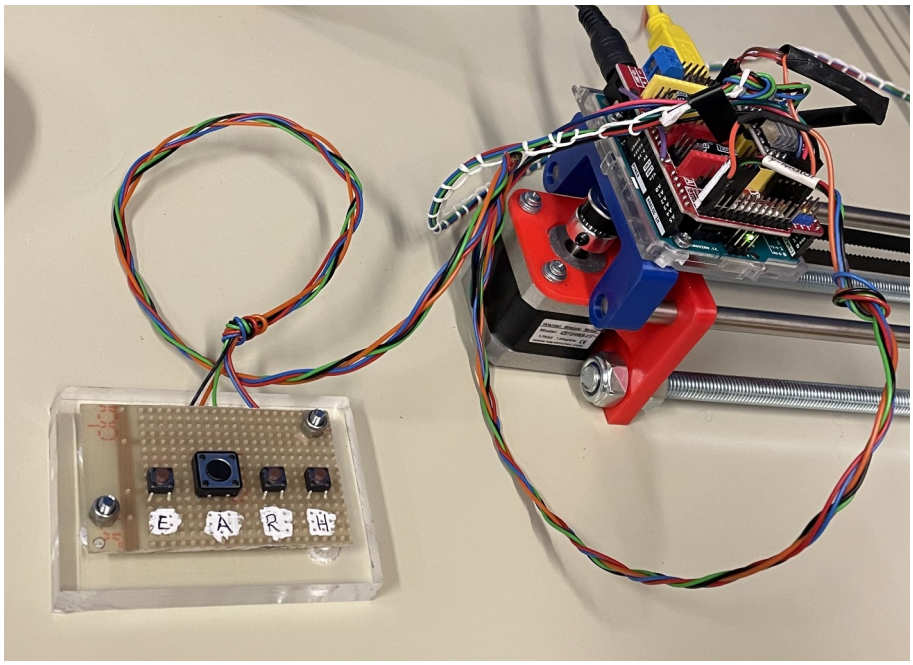
Button	Card Name	Cable colour	Description
E Stop	E	Black Connector white mark	In parallel with Arduino Reset
Resume	R	Green	Resume printing (after Hold)
Hold	H	Red	Hold printing
Abort	A	Blue	Abort present printing
GND		Orange	





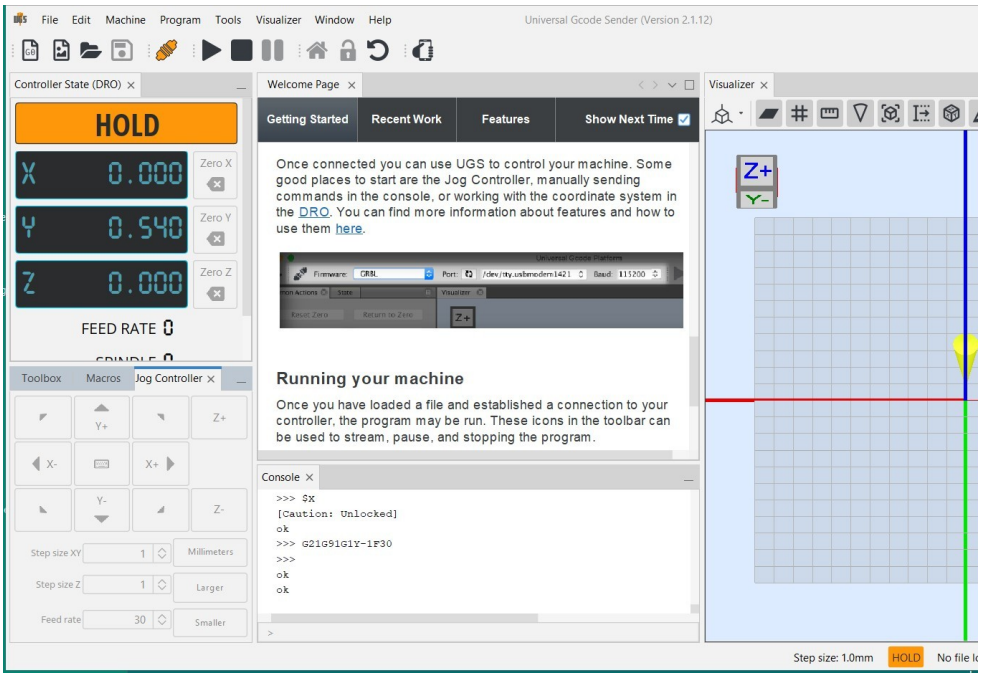



E
A
R
H
E Stop
Abort
Alarm
Resume
Hold


E
A
R
H

3.2 Actions

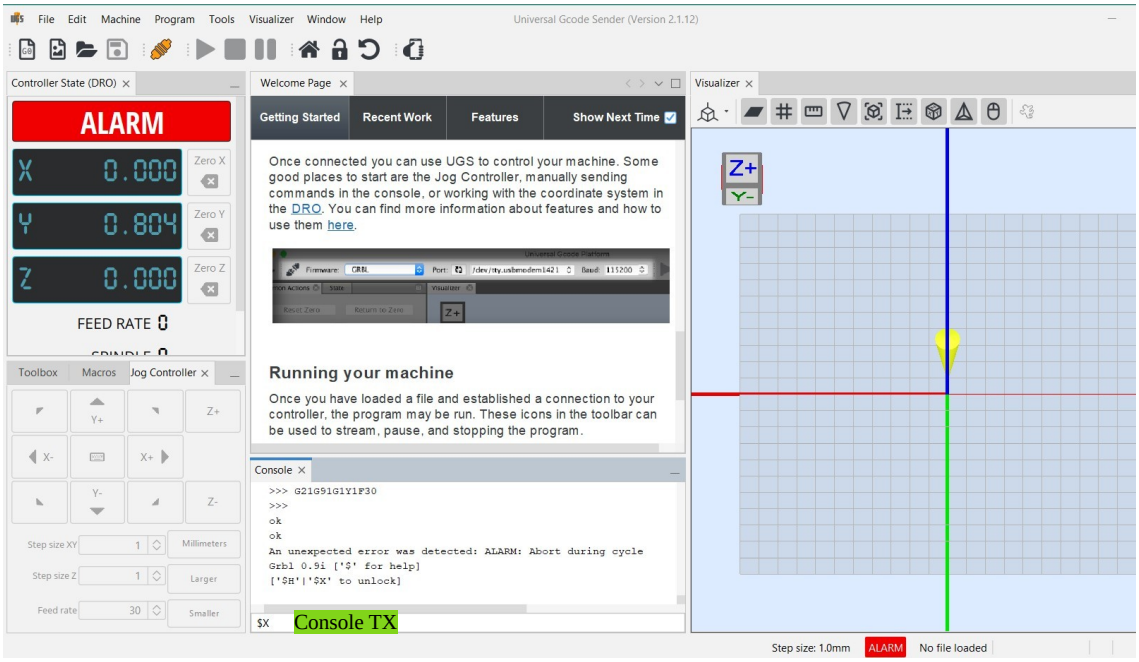
When *HOLD* is pressed, process continues after *RESUME* is pressed.



When *ABORT* is *pressed:*

- In *UGcode-Sender toolbox* click *Unlock*
  or
- *enter* command *$X in Console TX*
- Then, status passes to *IDLE*

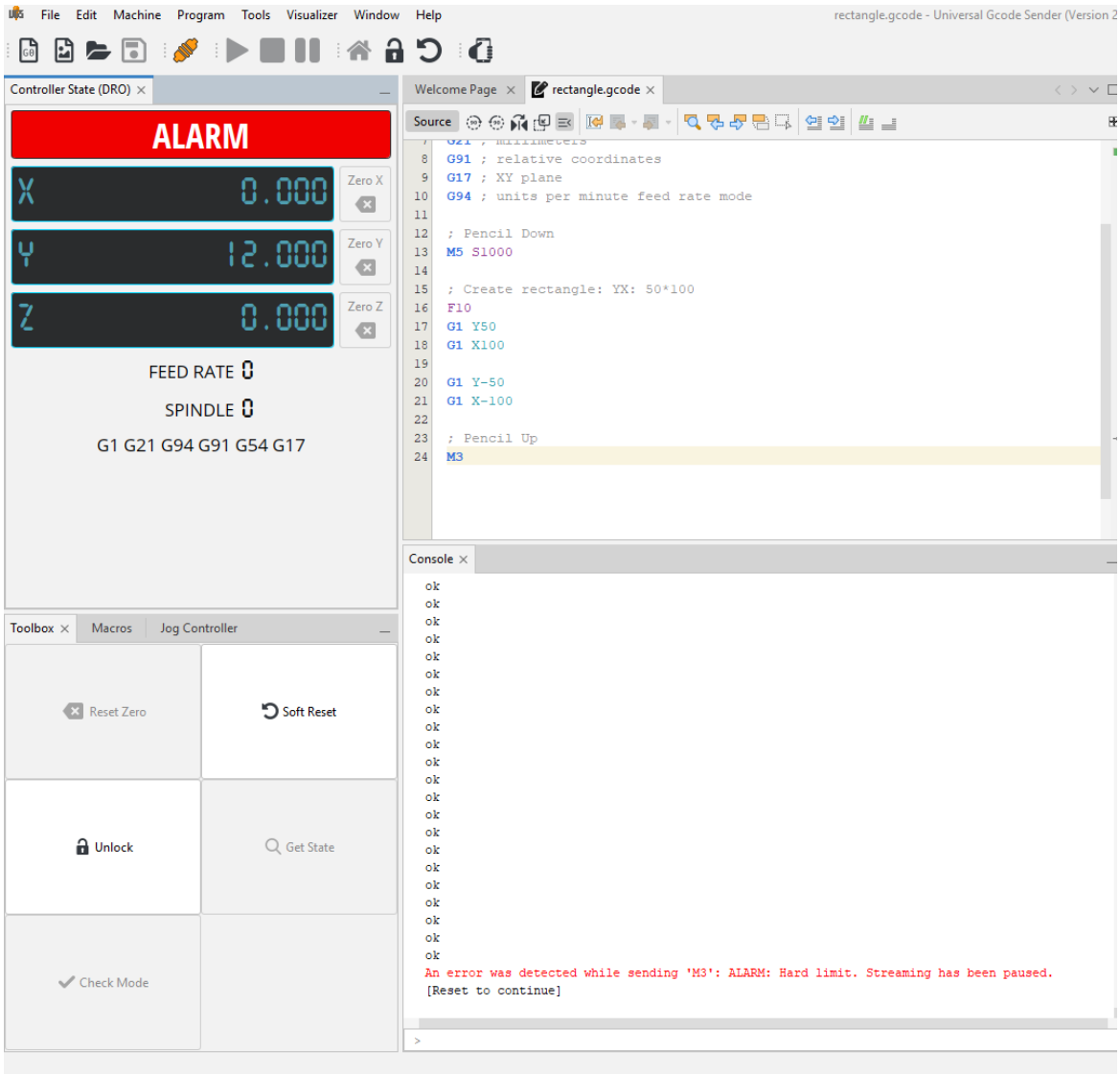
Note: $H is for Homing



4 Detection of Limit Switches

When *Limit Switch* are detected:

- *Press E Stop* to reset and Reload the program
  or
- In *UGcode-Sender toolbox* click *Soft Reset*, then, for unlocking and reloading
  - In *UGcode-Sender toolbox* click *Unlock*
    or
  - *enter* command *$X in Console TX*
- Then, status passes to *IDLE*



5 Ugcode-sender

When process is abnormally stopped, *Controller State* coordinates */ Visualizer* indications stays unchanged, therefore they must be manually reset with *Zero [X/Y/Z]* buttons