Things Read to setup your Barduino + CNC shield

—> Grbl_Esp32. ino + all libraries from this repo https://gitlab.com/fablabbcn-projects/cnc-machines/six-pack-cnc/-/tree/master/firmware-grblesp32/firmware-3AxisBrushedMotor/Grbl_Esp32

—> Documentation for the motor driver: https://www.pololu.com/product/2132 https://www.pololu.com/product/2133

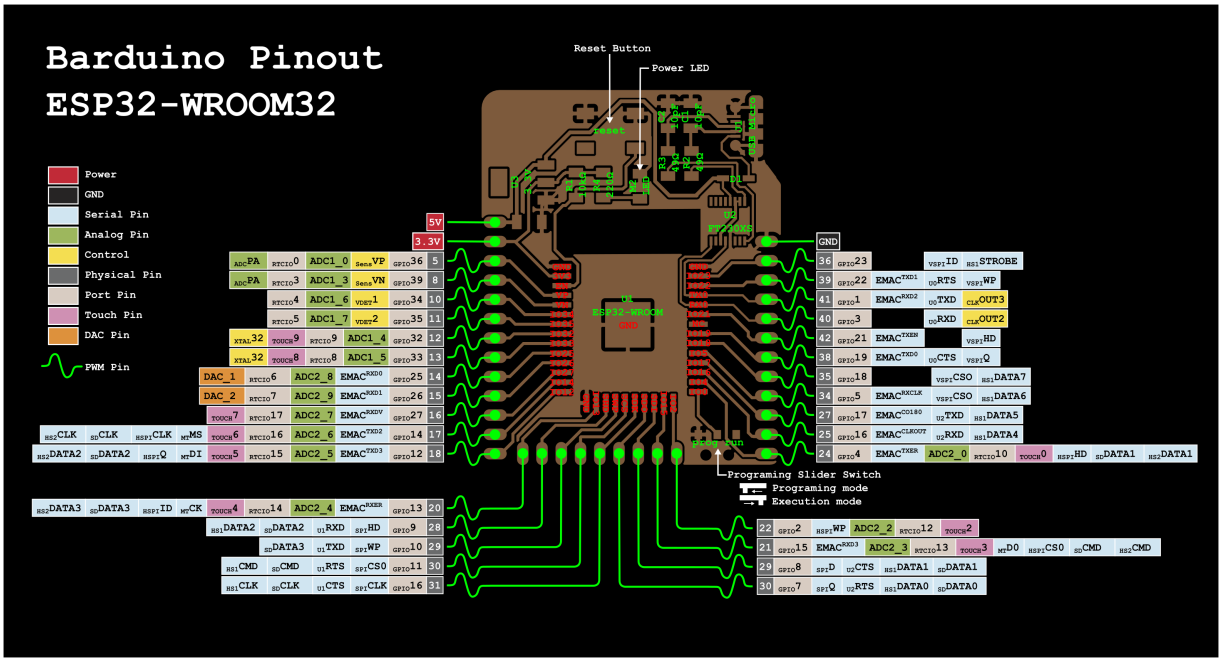
NEMA 11x 3 + NEMA 17 x3

NEMA14 Size: 35 mm square × 36 mm, not including the shaft (NEMA 14)

Barduino Info

https://gitlab.com/fablabbcn-projects/electronics/barduino

Barduino 2.2 —> https://gitlab.com/fablabbcn-projects/electronics/barduino/-/tree/master/barduino-microusb-2.2
Interface FTDI-USB* Fabbable Yes CNC Shield Compatible Yes



Libraries

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
https://dl.espressif.com/dl/package_esp32_index.json

Shield - original repo

https://github.com/bdring/Grbl_Esp32/wiki/Development-Roadmap

ESP32- Web UI
https://github.com/luc-github/ESP3D-WEBUI

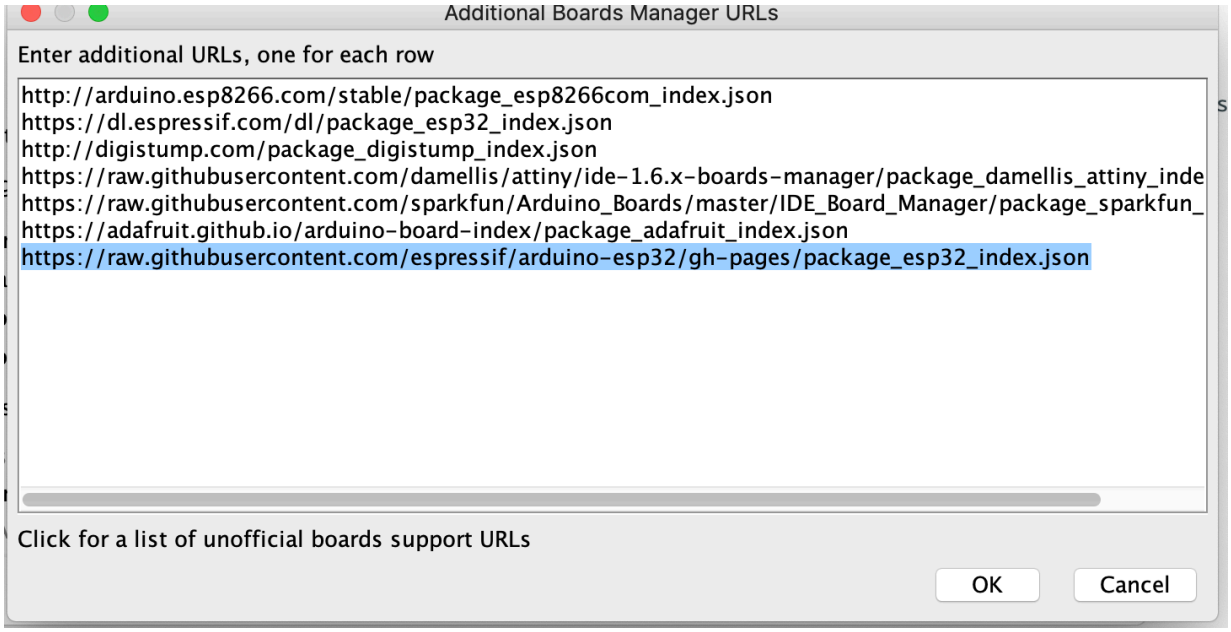
_____

Things to do to Use the CNC shield + Barduino

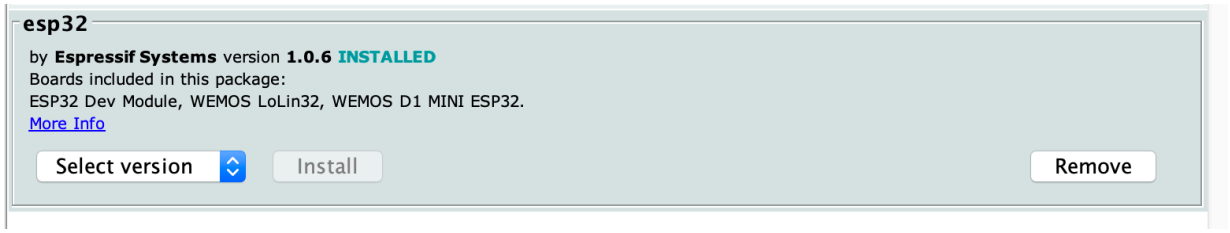
1- Download or Clone this repo —>https://gitlab.com/fablabbcn-projects/cnc-machines/six-pack-cnc/-/tree/master/firmware-grblesp32/firmware-3AxisBrushedMotor/Grbl_Esp32
2- Copy this two libraries from the libraries folder on the Firmware libraries folder to your Arduino Library folder. Mine is on Documents/Arduino/Libraries.

- ESP32SSDP
- arduinoWebSockets

3- Install Board Manader for ESP32 —> Arduino —> Preferences
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



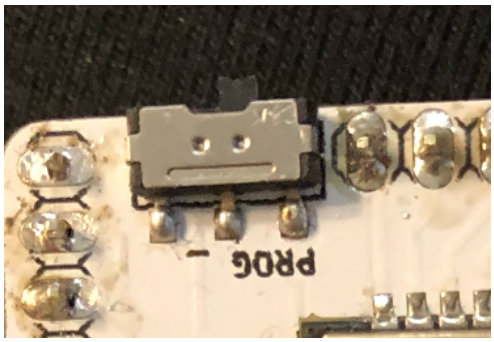
4 - Install ESP32 Espressif on Board Manager



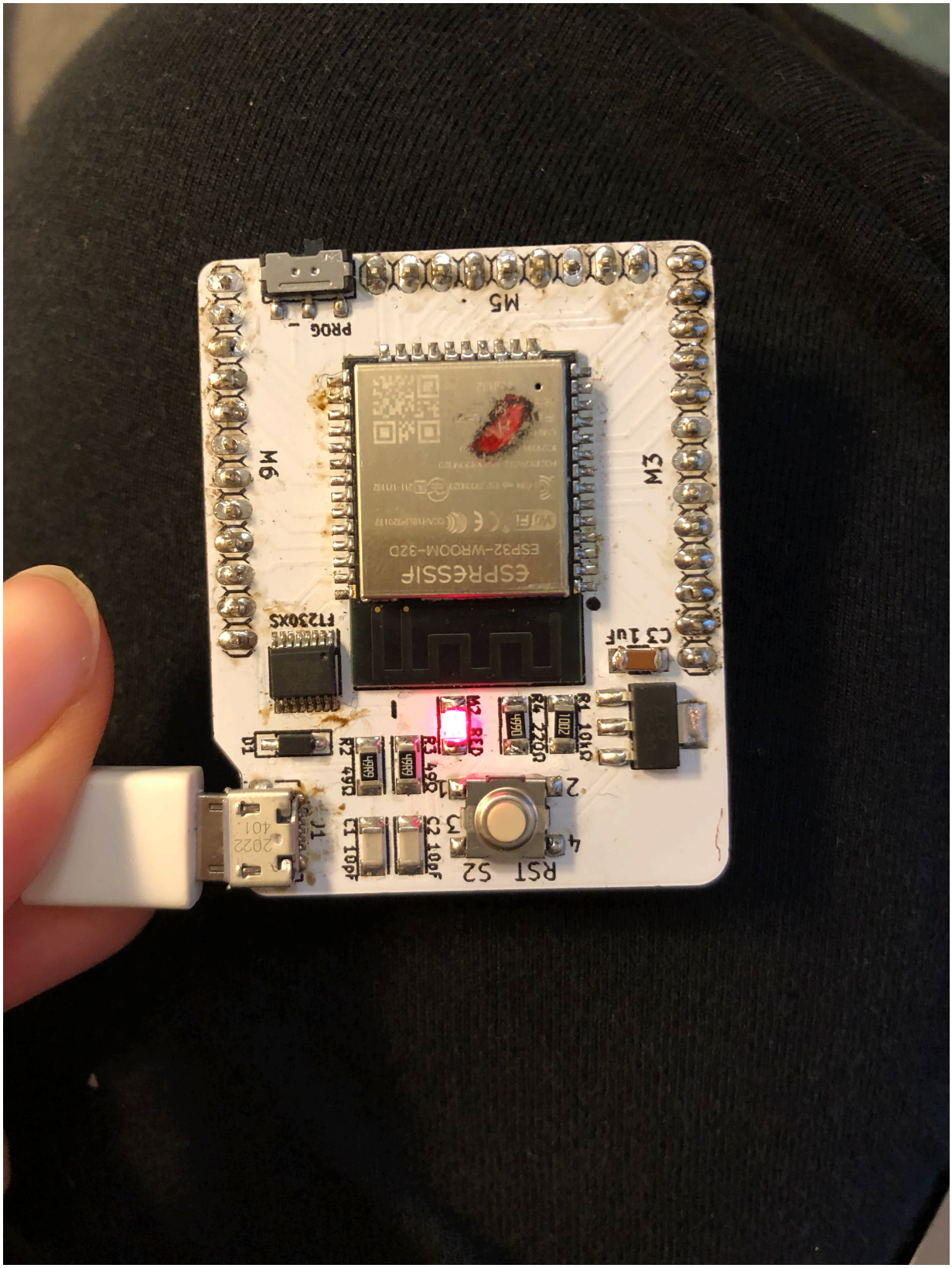
5- Reading installations instructions from here:

https://fabacademy.org/2020/labs/barcelona/students/david-prieto/projects/AssemblingSPML/#assembling-a-six-pack-milling-machine

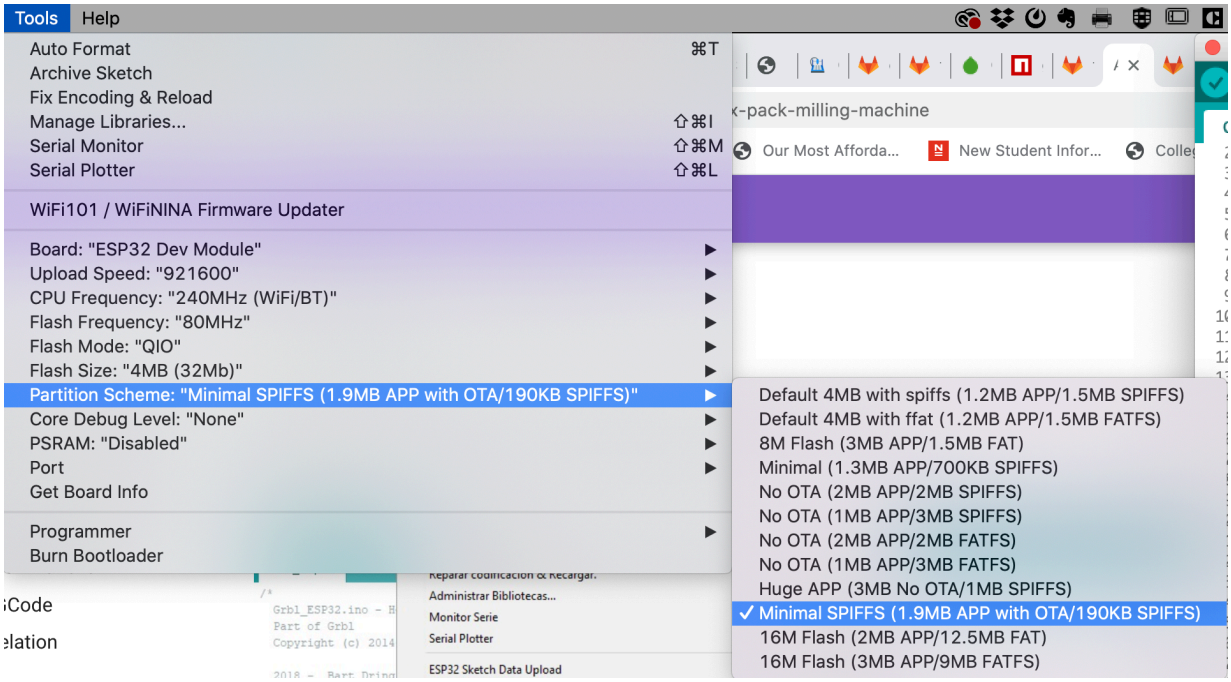
Before 6 - Make sure that before you connect the board the switch is on programming mode:



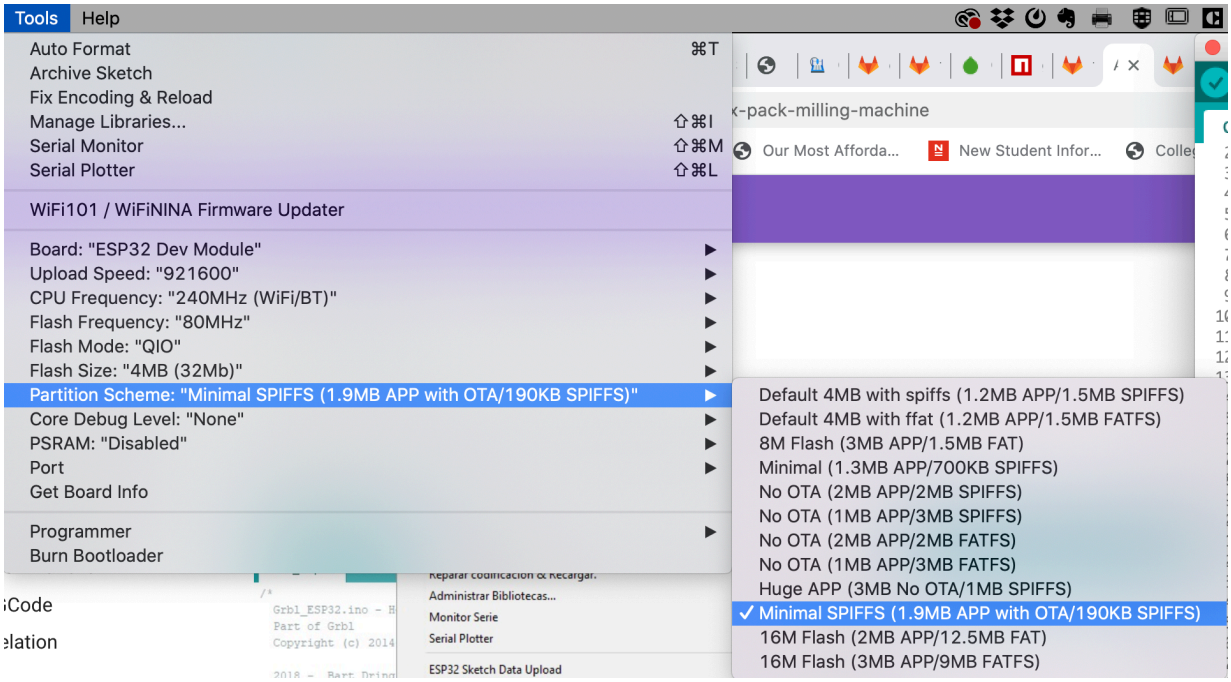
6- Original Grbl documentation says to remove the barduino board from the shield to program the firmware to prevent any possible damage to the shield. Connect the board:



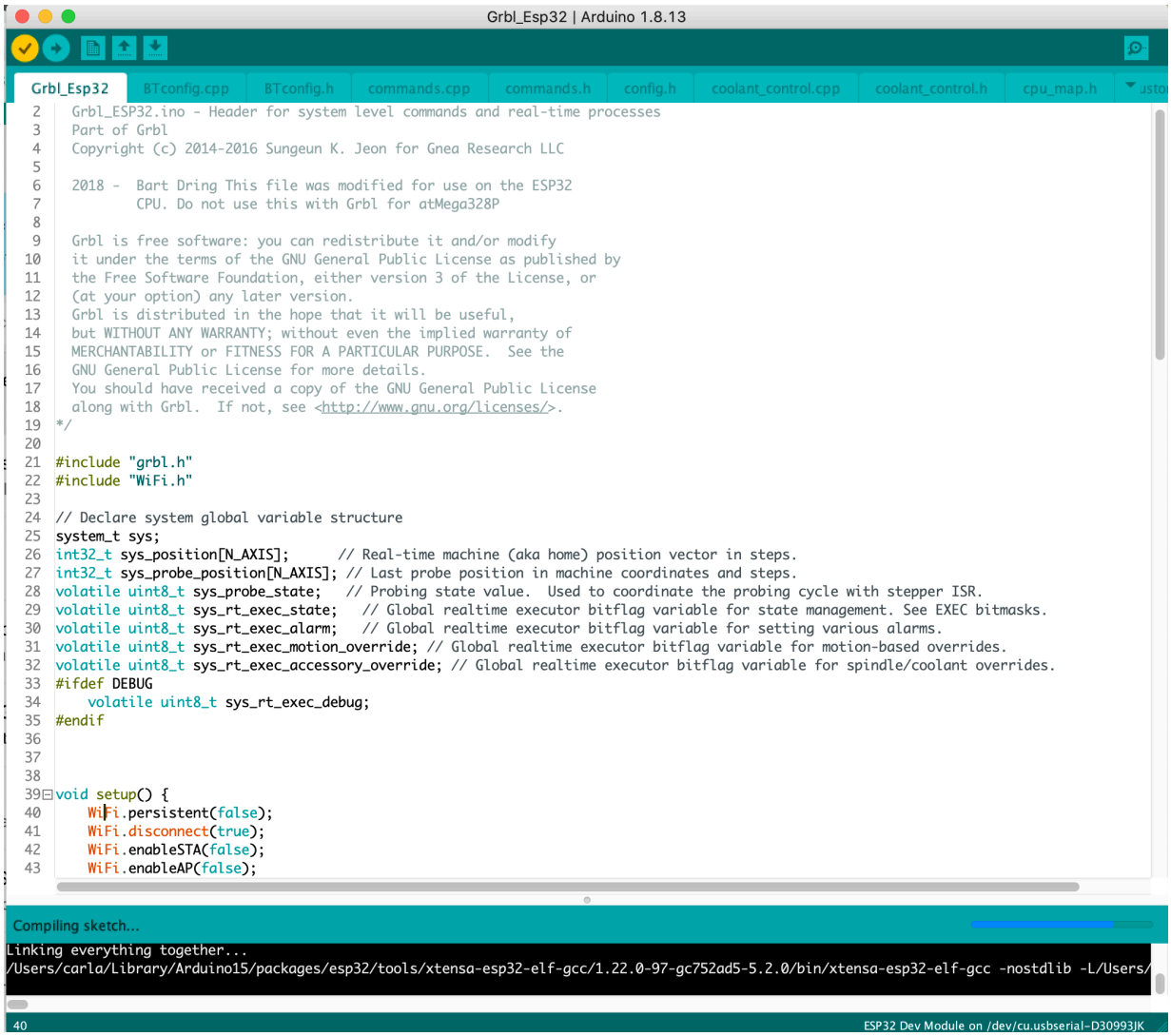
7- select board. ESP32 - dev Module



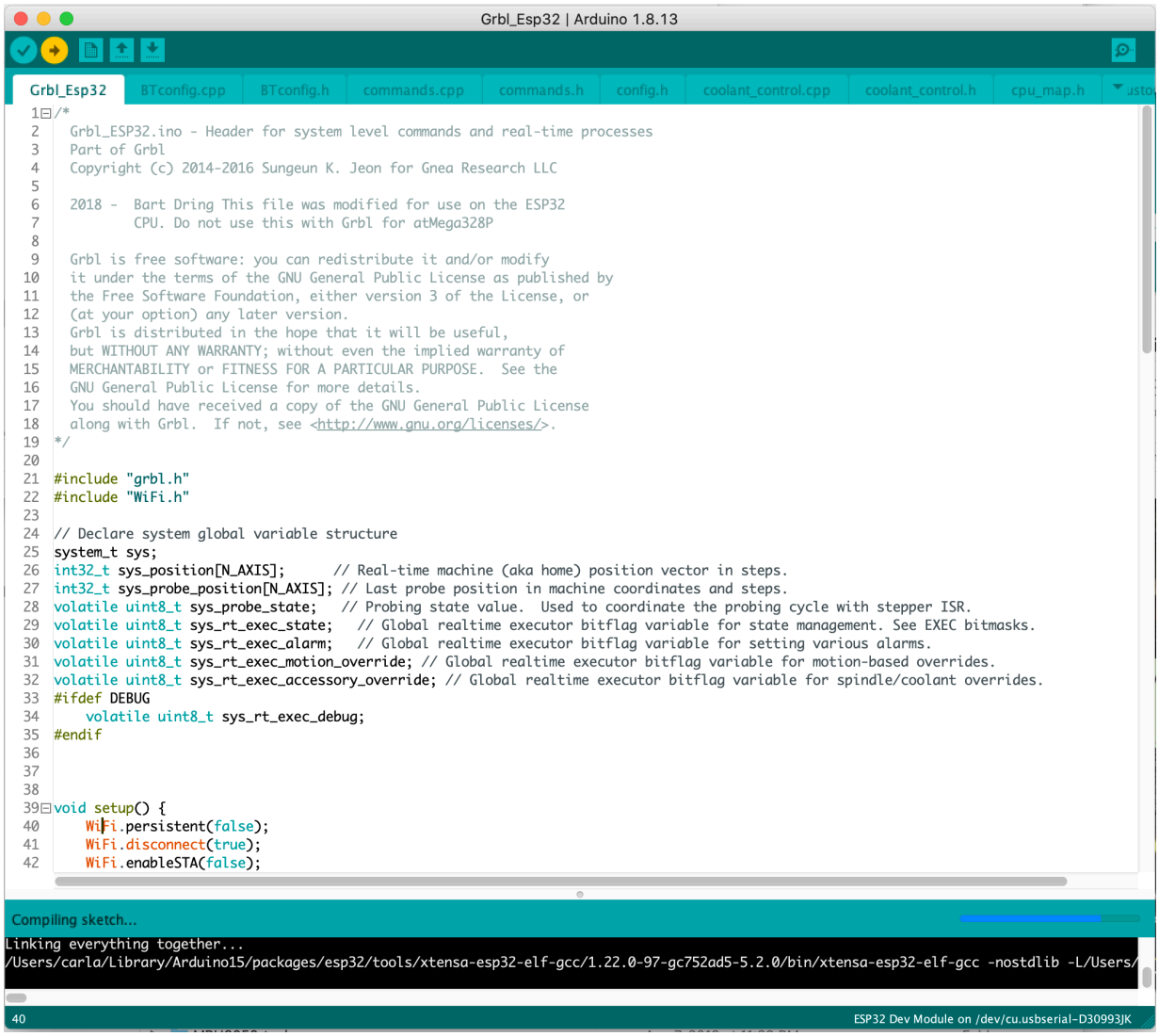
8- change Partition scheme to Minimal SPIFFS



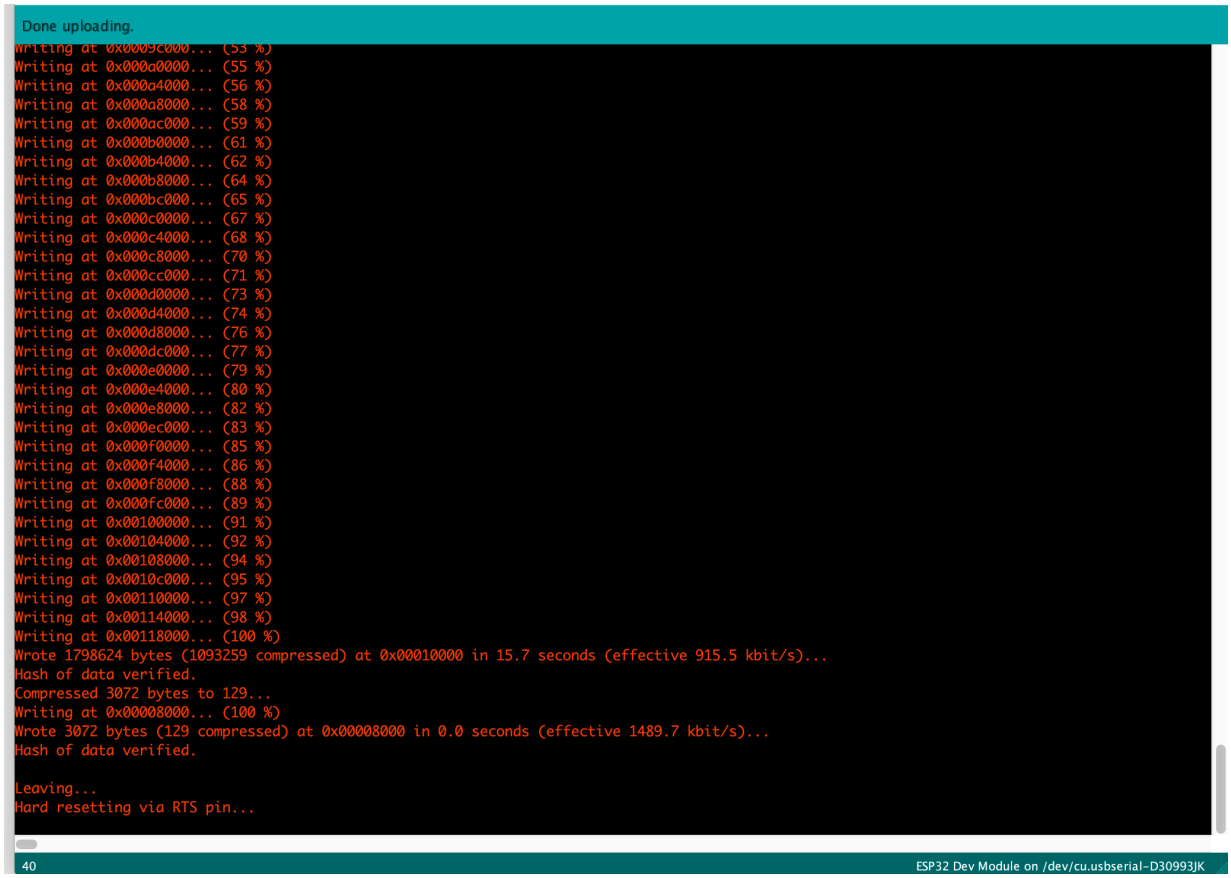
9- Compile Firmware Arduino File - it will take a while depending on the machine.



10 - Upload the code

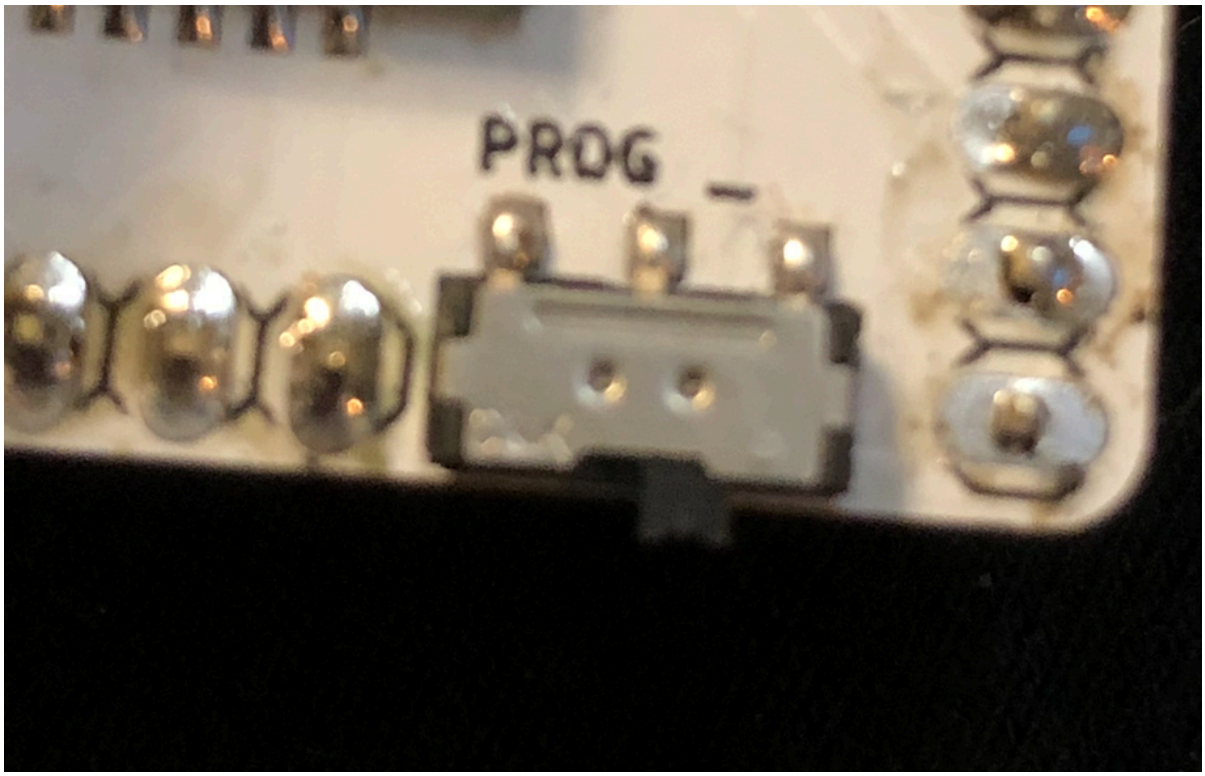

Grbl_Esp32 | Arduino 1.8.13
Grbl_Esp32
BTconfig.cpp
BTconfig.h
commands.cpp
commands.h
config.h
coolant_control.cpp
coolant_control.h
cpu_map.h
/*
Grbl_ESP32.ino - Header for system level commands and real-time processes
Part of Grbl
Copyright (c) 2014-2016 Sungeun K. Jeon for Gnea Research LLC
2018 - Bart Dring This file was modified for use on the ESP32
CPU. Do not use this with Grbl for atMega328P
Grbl is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
Grbl is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
You should have received a copy of the GNU General Public License
along with Grbl. If not, see <http://www.gnu.org/licenses/>.
*/
#include "grbl.h"
#include "WiFi.h"
// Declare system global variable structure
system_t sys;
int32_t sys_position[N_AXIS]; // Real-time machine (aka home) position vector in steps.
int32_t sys_probe_position[N_AXIS]; // Last probe position in machine coordinates and steps.
volatile uint8_t sys_probe_state; // Probing state value. Used to coordinate the probing cycle with stepper ISR.
volatile uint8_t sys_rt_exec_state; // Global realtime executor bitflag variable for state management. See EXEC bitmasks.
volatile uint8_t sys_rt_exec_alarm; // Global realtime executor bitflag variable for setting various alarms.
volatile uint8_t sys_rt_exec_motion_override; // Global realtime executor bitflag variable for motion-based overrides.
volatile uint8_t sys_rt_exec_accessory_override; // Global realtime executor bitflag variable for spindle/coolant overrides.
#ifdef DEBUG
volatile uint8_t sys_rt_exec_debug;
#endif
void setup() {
WiFi.persistent(false);
WiFi.disconnect(true);
WiFi.enableSTA(false);
Compiling sketch...
Linking everything together...
/Users/carla/Library/Arduino15/packages/esp32/tools/xtensa-esp32-elf-gcc/1.22.0-97-gc752ad5-5.2.0/bin/xtensa-esp32-elf-gcc -nostdlib -L/Users/
40
ESP32 Dev Module on /dev/cu.usbserial-D30993JK

DONE!!!!



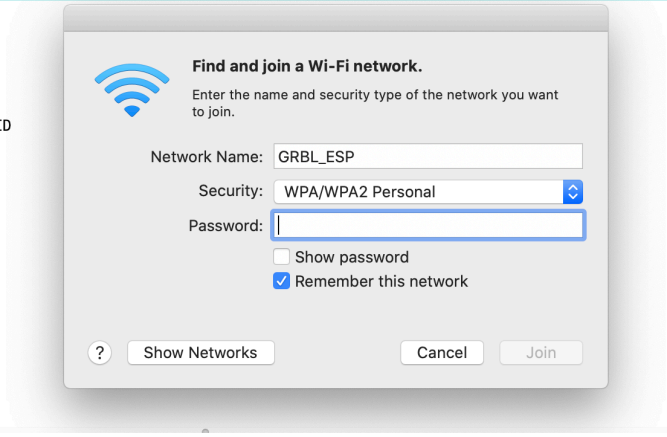
11 - Disconnect the board.

12- Change the switch to Execution Mode

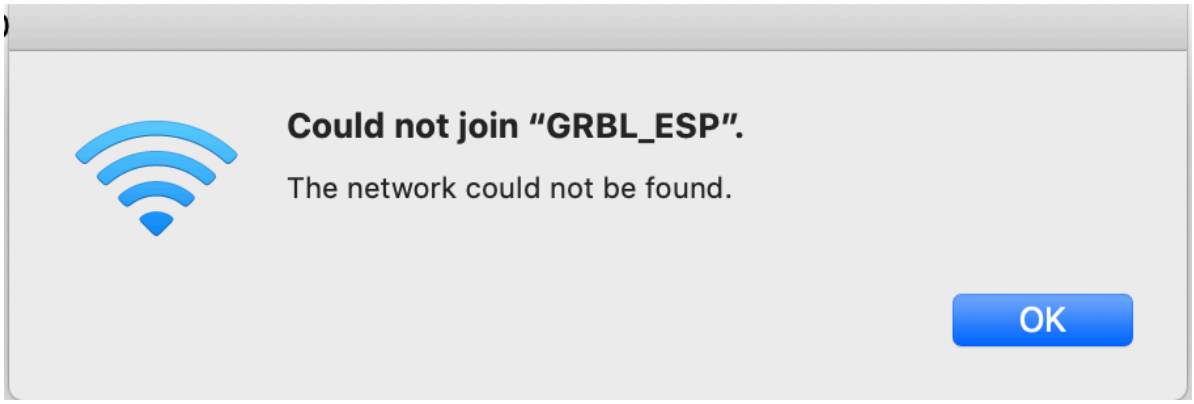


13 - Check for the ESP32 wifi connection. Select Wificonfig.h file to see the Wifi connection details or check screenshot here:

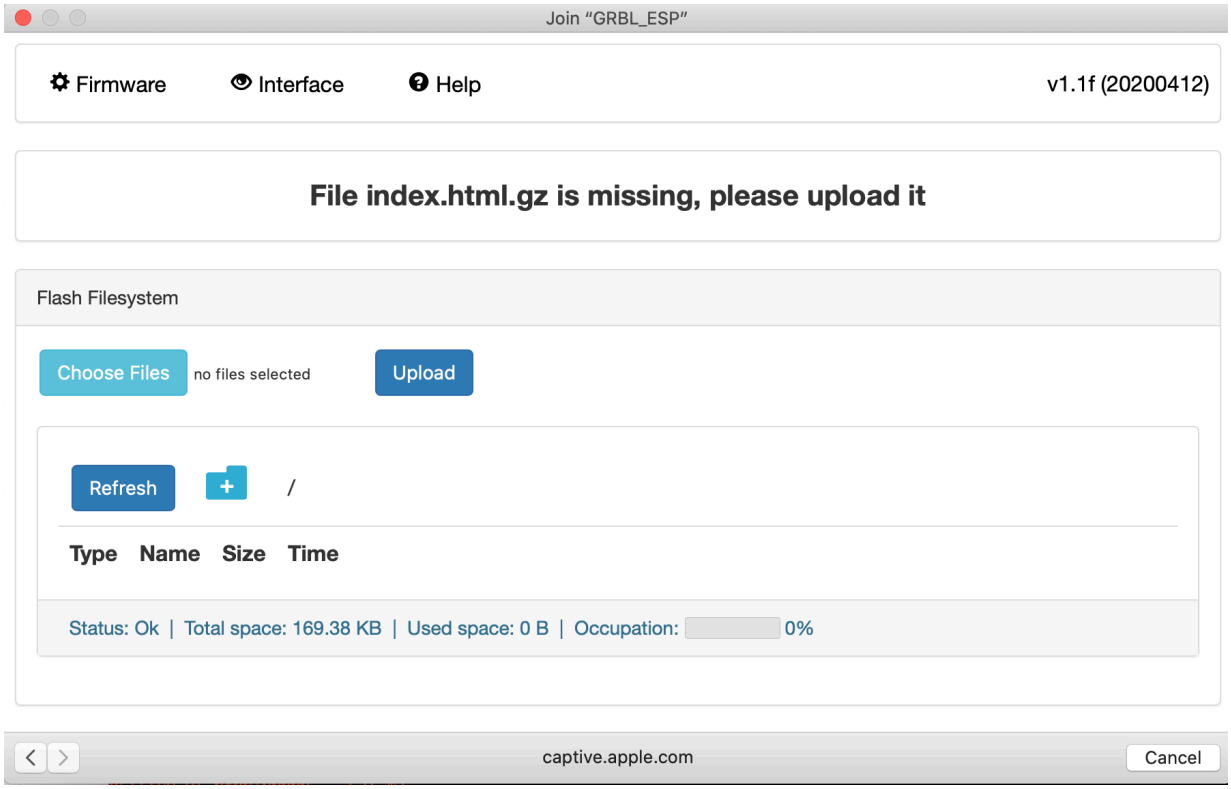
```
#define ESP_SAVE_ONLY 0
#define ESP_APPLY_NOW 1

//defaults values
#define DEFAULT_HOSTNAME "grblesp"
#ifdef CONNECT_TO_SSID
    #define DEFAULT_STA_SSID CONNECT_TO_SSID
    #define DEFAULT_STA_PWD SSID_PASSWORD
#else //!CONNECT_TO_SSID
    #define DEFAULT_STA_SSID "GRBL_ESP"
    #define DEFAULT_STA_PWD "12345678"
#endif //CONNECT_TO_SSID
#define DEFAULT_STA_IP "0.0.0.0"
#define DEFAULT_STA_GW "0.0.0.0"
#define DEFAULT_STA_MK "0.0.0.0"
#define DEFAULT_AP_SSID "GRBL_ESP"
#define DEFAULT_AP_PWD "12345678"
#define DEFAULT_AP_IP "192.168.0.1"
#define DEFAULT_AP_MK "255.255.255.0"
#define DEFAULT_AP_CHANNEL 1
#define DEFAULT_WEBSERVER_PORT 80
#define DEFAULT_HTTP_STATE 1
```



If you try to do it on programming mode, it won't work:

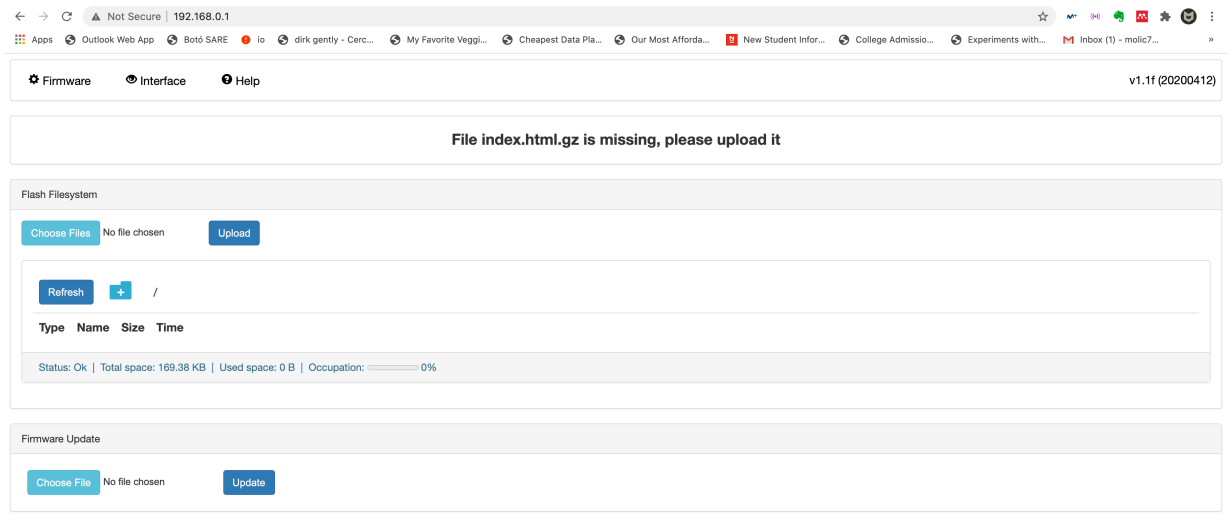


14 — This is the page that pops-up on MacOS (also probably on Windows, too). It will show you the main website of the board. As it’s not a fully functional browser, close it.
Using your browser of preference, visit 192.168.0.1. Make sure you are still connected to the wireless network GRBL_ESP.

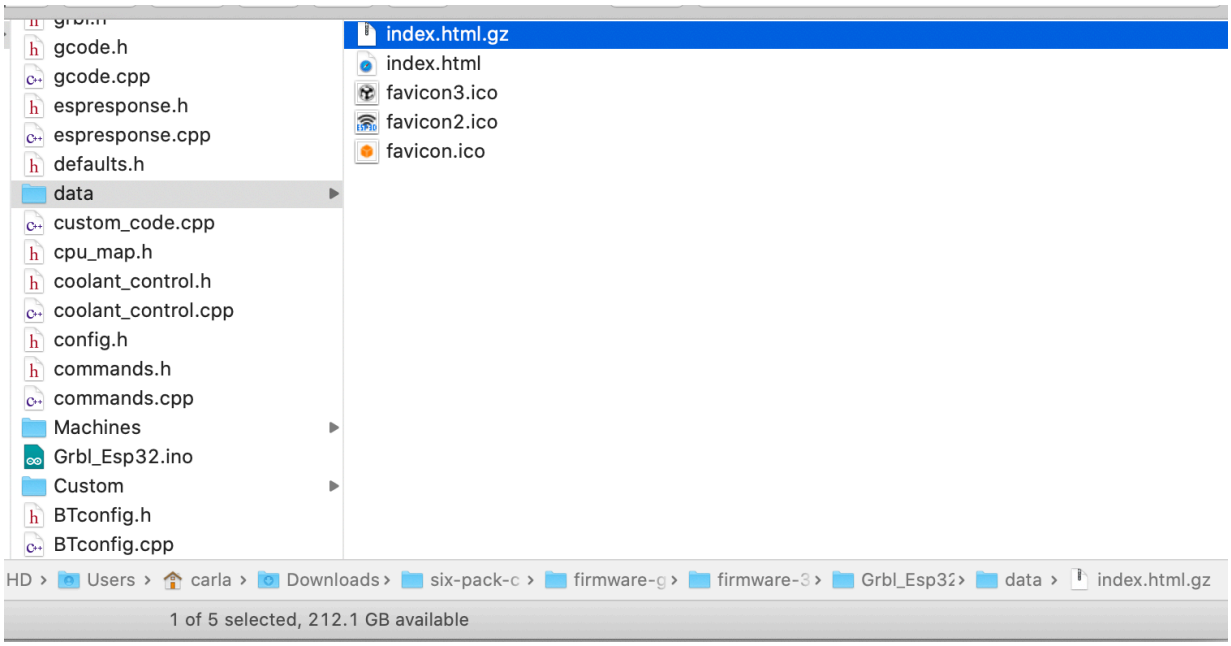


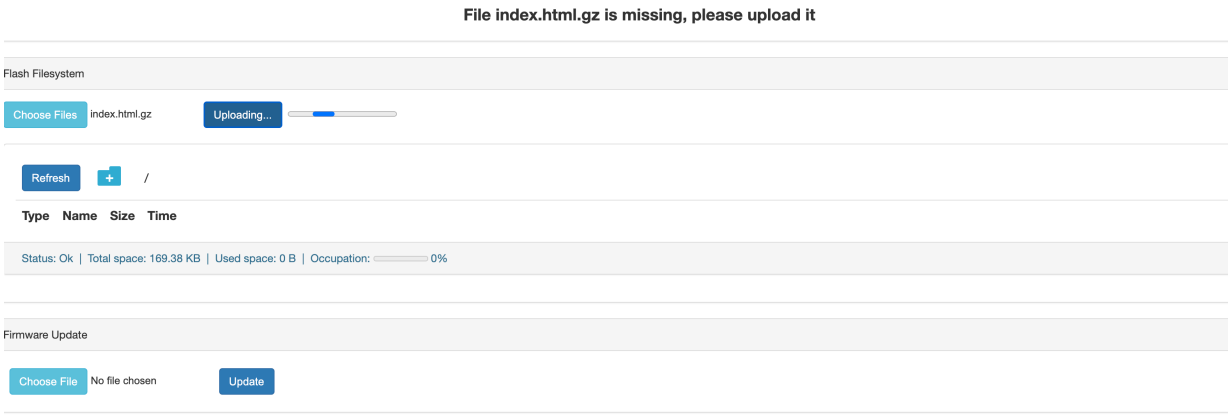
15- We need to upload the index.html.gz file from the data folder that was on the firmware files. Thanks to Tue and David from 2020 Barcelona Fabacademy who had an awesome documentation: http://academany.fabcloud.io/fabacademy/2020/

labs/barcelona/barcelona-machines/light-machine/

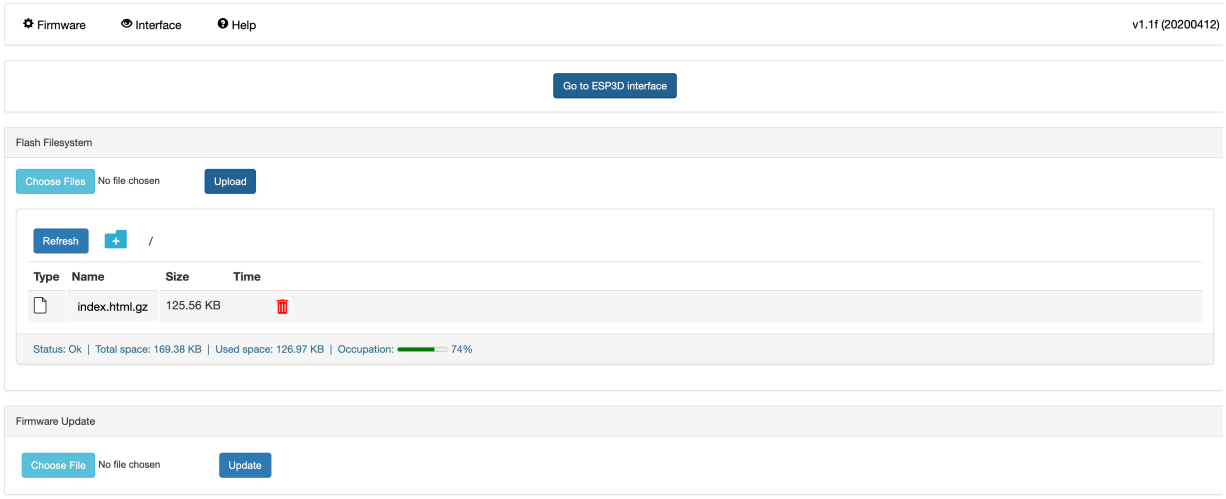


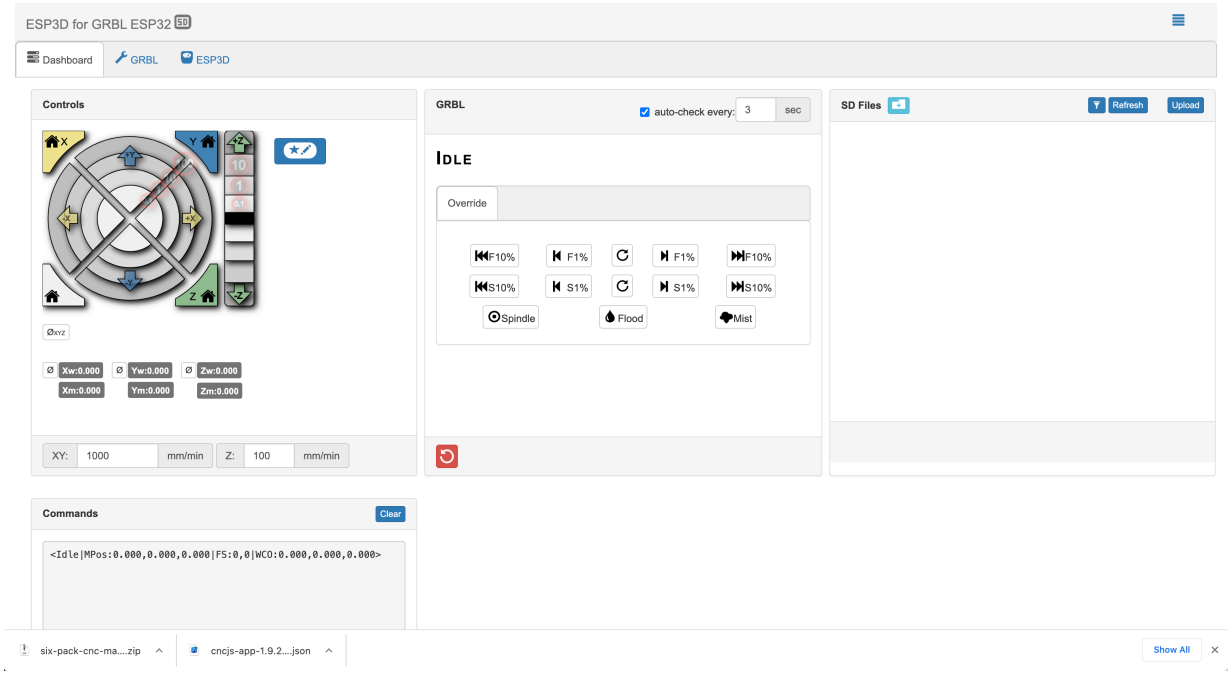
After finding the file —> Go to 192.168.0.1



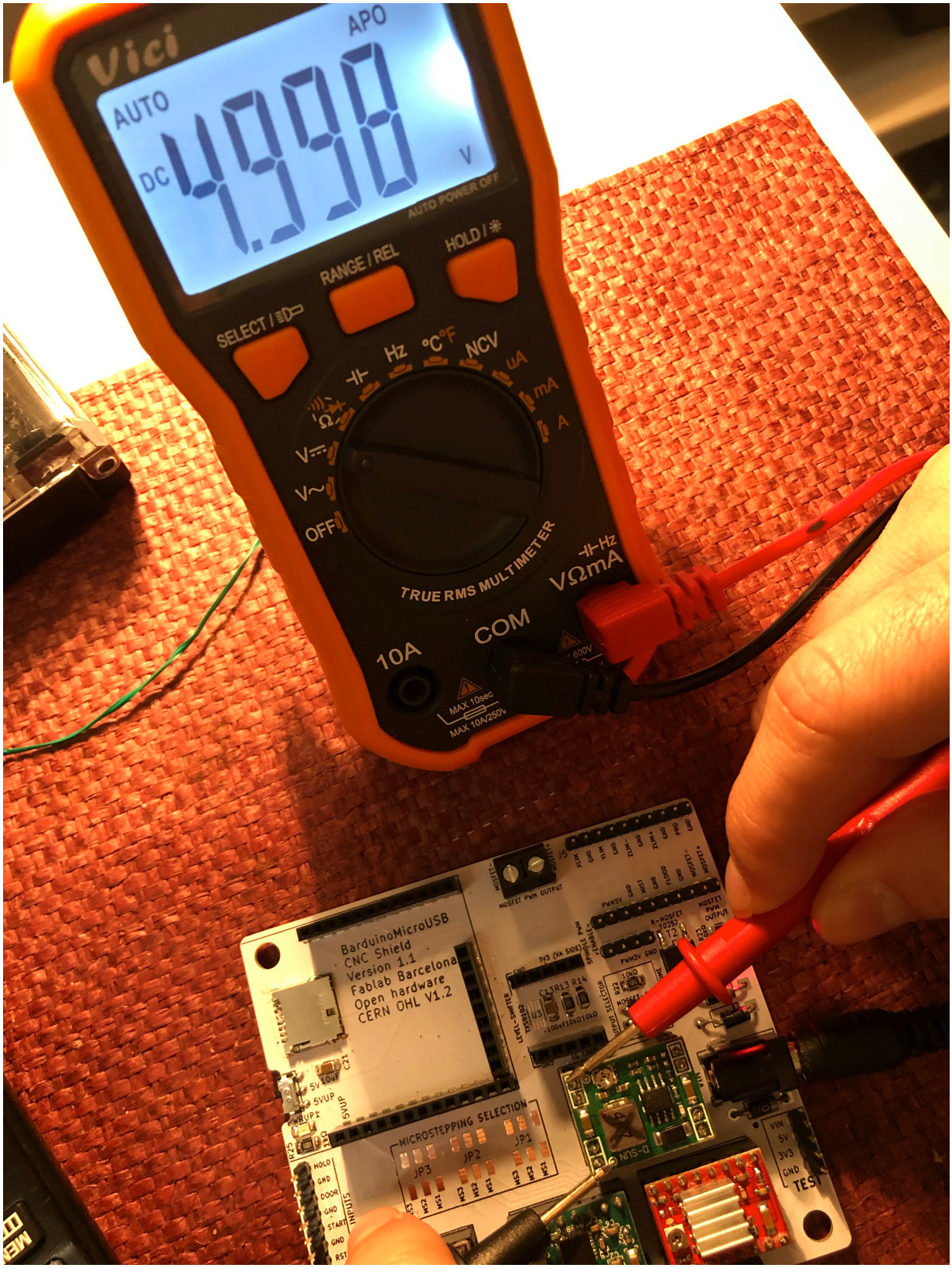


It will say Uploaded 100% and a new button will appear “Go to ESP3D interface”. Click on it and it will take you to the motor control interface.

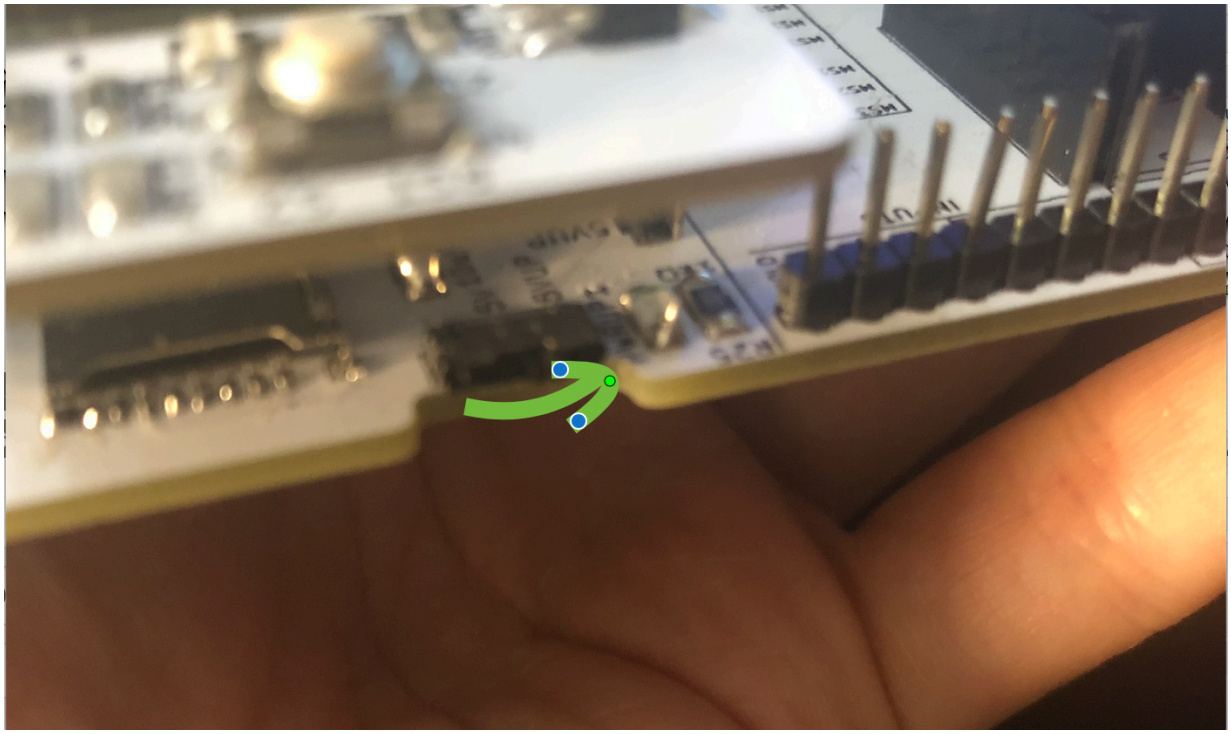




16 — Check the Voltage of the Voltage regulator.



17 - Now you can connect the Barduino to the driver board. Move the switch under the board to 5VUP before mounting the Barduino to the header pins.



18 - WARNING - Time to setup the current limit for each motor driver —> Watch this video and come back

https://www.youtube.com/watch?v=89BHS9hfSUk

As you'll see you need to check the specs for your motor, meaning NEMA 11, 14, & 17 might have differents specs. If we setup each motor driver according to its motor they won't be interchangeable (unless it's the same model) or the motor could get damaged

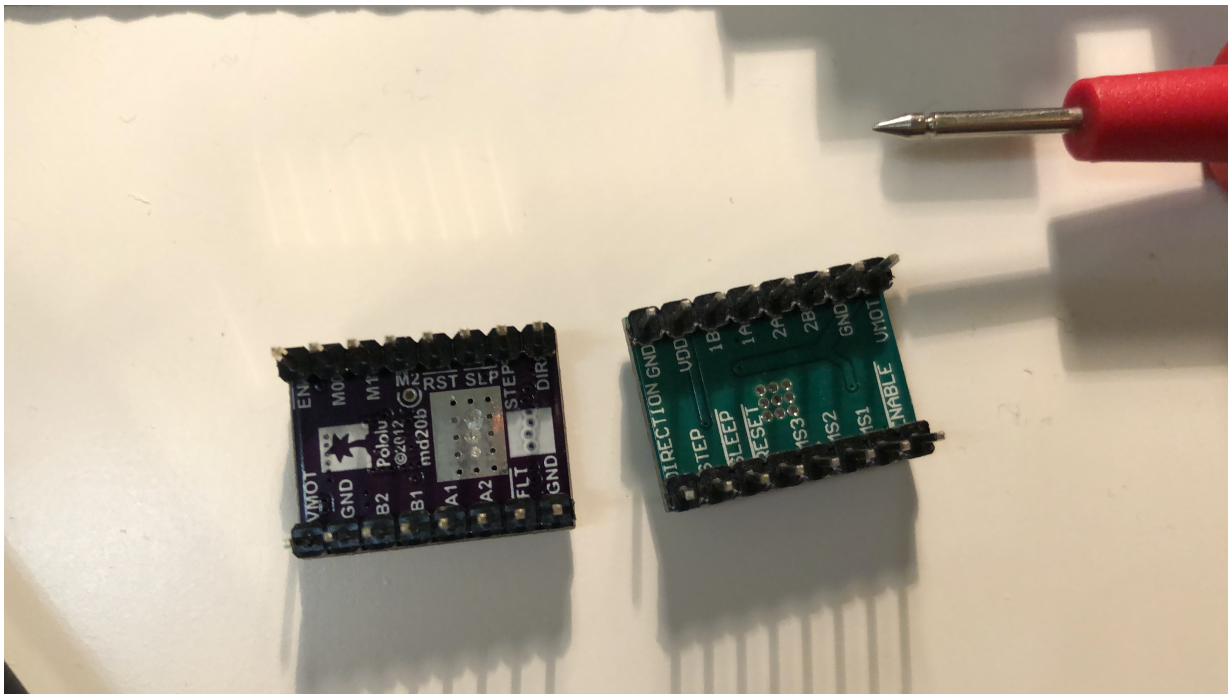
16A Purple one - Polulu driver DRV8825

The other two drivers look like knockouts of the other polulu models. I couldn't find the specs

	DRV8825	DRV8834	DRV8880	MP6500		TB67S279FTG	TB67S249FTG	STSPIN820	STSPIN220
	8.2 V	2.5 V	6.5 V	4.5 V		10 V	10 V	7 V	1.8 V
	45 V	10.8 V	45 V	35 V		47 V	47 V	45 V	10 V
A	1.5 A	1.5 A	1 A	1.5 A		1.1 A	1.6 A	0.9 A	1.1 A
	2.2 A	2 A	1.6 A	2.5 A	2 A	2 A	4.5 A	1.5 A	1.3 A
	1/32	1/32	1/16	1/8		1/32	1/32	1/256	1/256
	4	4	4	4		4	4	4	4
	high max voltage, high current	low-voltage operation, high current	AutoTune, digital current reduction, high max voltage	high current	digital current control, high current	Auto Gain Control, ADMD, high max voltage	Auto Gain Control, ADMD, high max voltage, high current	128 and 256 microsteps, high max voltage	64, 128, and 256 microsteps, low-voltage operation
49	$8.95	$5.95	$6.95	$5.95	$5.95	$7.75	$9.95	$7.75	$5.95

additional cooling.

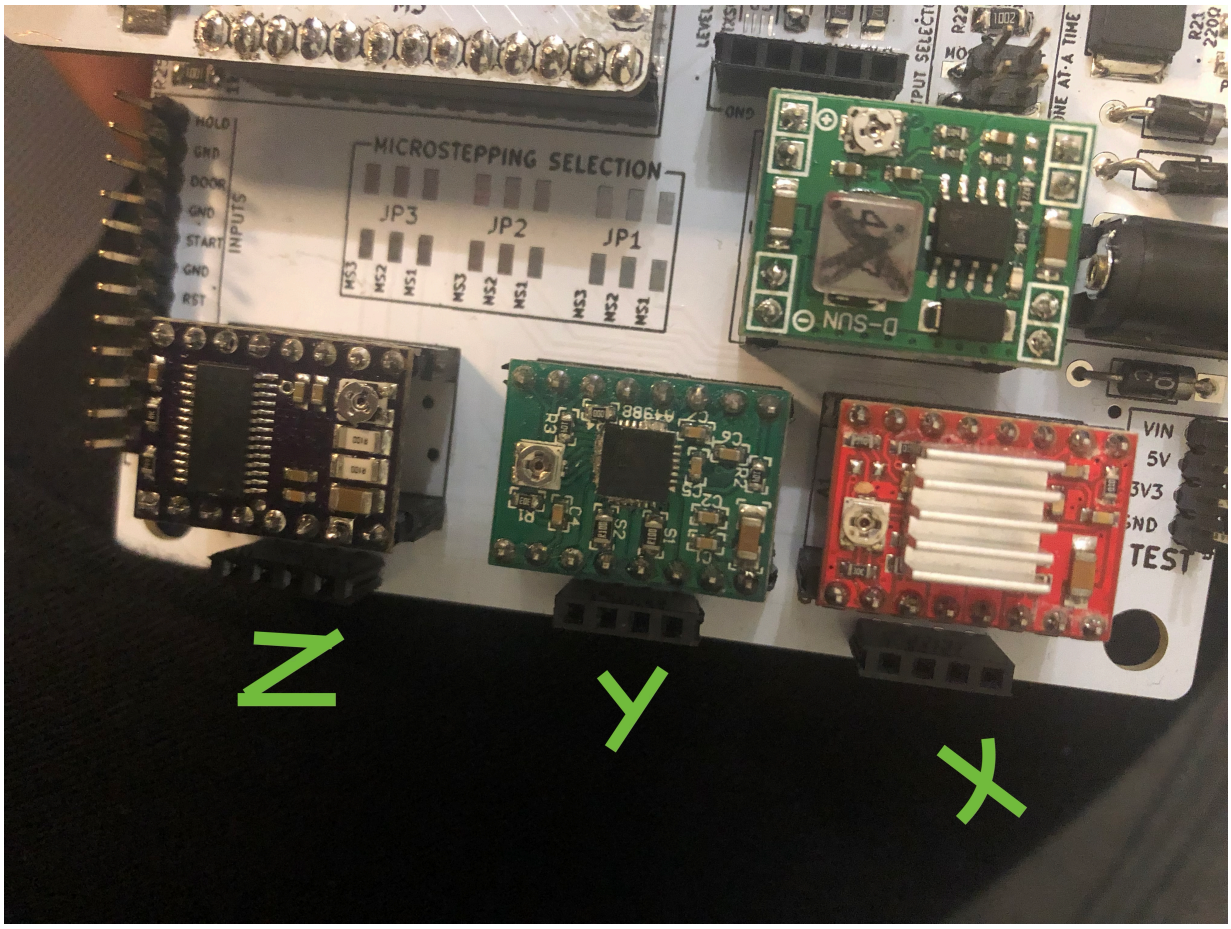
Not all have the same pinout orientation. —> Check for GND to match the shield .



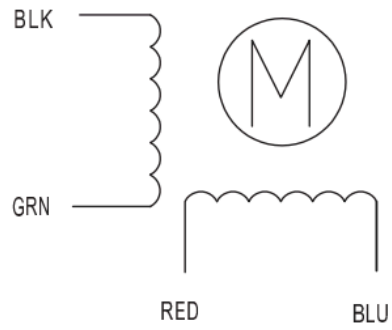
For the NEMA 14- SY35ST36 —> The Voltage measured from the screw or VREF to GND needs to be around 500mV.

https://www.pololu.com/file/0J689/SY35ST36-1004A.pdf

19- Time to connect the motor- Make sure that the power supply is desconnected.



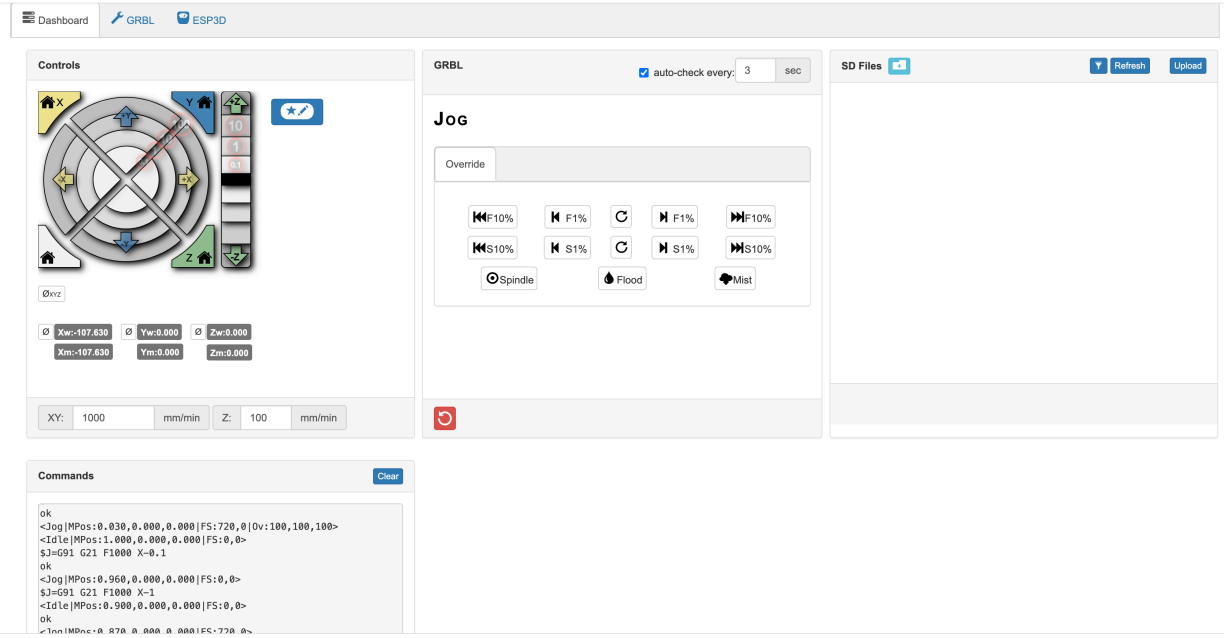
- Wiring Diagram:



- Dimensions:

I connected the wires according the wiring diagram on the motor’s datasheet —> Black, green, red and blue.

20- Time to connect to the GRBL-ESP http://192.168.0.1/



When sending the commands the motors moves. Not super consistently, as it seems that wiring is a bit flickery. Wiring should be checked, everything else seems to be working.