SKR 1.3 & TMC2130 on a Creality Ender-3

A few month ago, I ordered a [Creality Ender 3](https://www.creality3donline.com/c/ender-3_0400" \t "_blank) with the intention to tweak it to be a better printer than the [Prusa MK3](https://www.prusa3d.com/original-prusa-i3-mk3/" \t "_blank) I have access at the maker space at my work.

One of the upgrade I wanted to do, was the silent stepper driver from [Trinamic](http://trinamic.com/" \t "_blank) after hearing about them in YouTube videos from [Thomas Salanderer](https://www.youtube.com/channel/UCb8Rde3uRL1ohROUVg46h1A) and [Teaching Tech](https://www.youtube.com/channel/UCbgBDBrwsikmtoLqtpc59Bw). This requires a new card and Teaching Tech did a very good guide on [how to install the MGS Gen L on a Creality Ender 3](https://www.youtube.com/watch?v=LNdMYgwez8Y). Around the same time, [Creality released a version of their board with TMC2208s](https://www.creality3donline.com/creality3d-new-upgrade-silent-114-mainboard-for-ender-3-pro-ender-5-customized-und-non-standard-matching_p0147.html" \t "_blank). Those two options are easy to install but the Creality board is way too limited (especially very small memory) and both of those board are still using 8-bit processor (card derived from Arduinos).

A 32-bit card for the same price is the [BigTreeTech SKR 1.3](https://www.biqu.equipment/products/bigtreetech-skr-v1-3-smoothieboard-32-bit-motherboard-open-source-tft3-5-2004lcd-12864lcd-display-support-bltouch-upgrade-skr-v1-1-leveling-for-3d-printer" \t "_blank). It is compatible with the MKS Gen L but also include all the wiring for SPI and UART mode in the card, no need for fancy cabling to add those modes. I ordered mine from AliExpress with 4 TMC2130. I went for the TMC2130 over the TMC2208 because I want to play with setting crash detection on my Ender 3. If I weren’t not a coder and wanting to play with the stallguard feature, I would have gone for TMC2208, which gives better torque in silent mode. **[EDIT]** If I were to order it now, I would ordered with **TMC2209** with get the best of both TMC2208 and TMC2130.

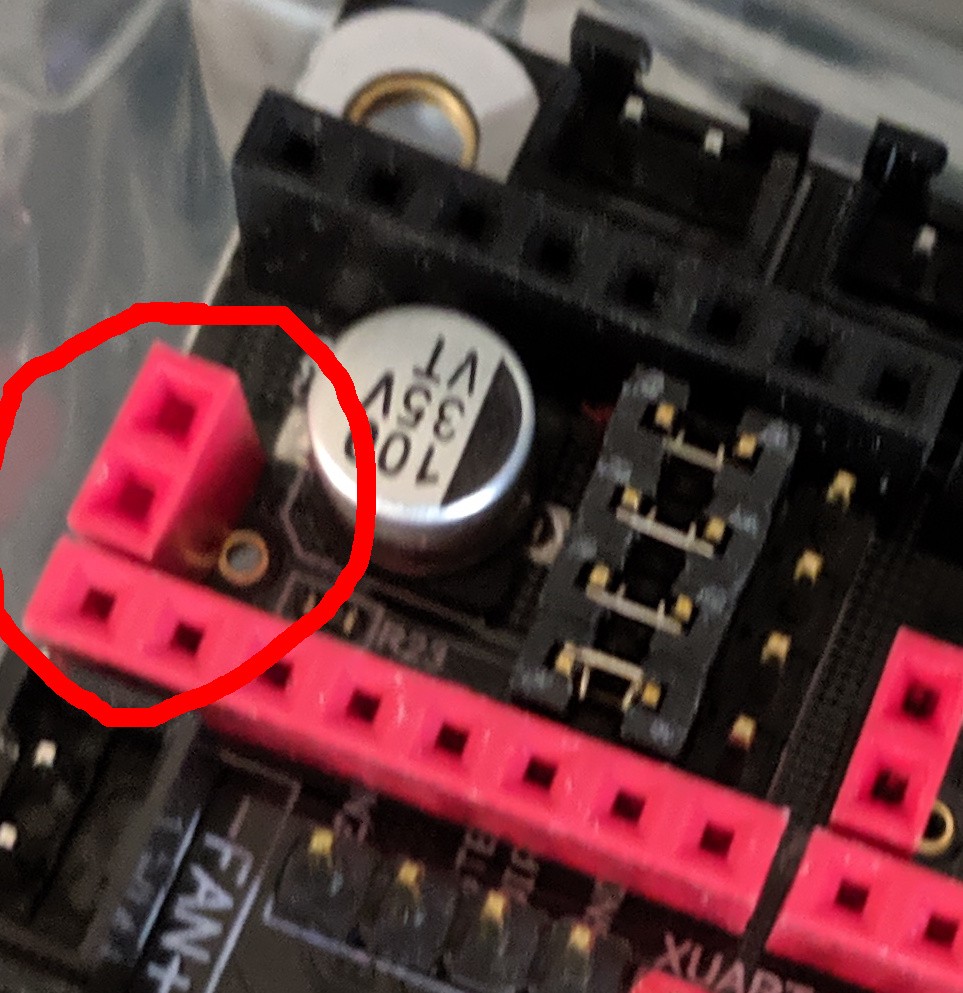
The only drawback of this card? Its lack of documentation! So here is my experience.

**Preparing the SKR 1.3 for TMC2130**

The hardest part on the SKR1.3 is to figure out what is the wiring. There is the [official video by BigTreeTech](https://www.youtube.com/watch?v=oaXfXkPYHpw) but it is really fast and does contains much details. [The documentation on Github is also hard to follow](https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/SKR%20V1.3-Instructions.pdf).

The main issue that troubled me was how to wire the TMC2130. I got the version 2 of the BigTreeTech TMC2130 breakout board which [does not contains the diagnostic pin soldered](https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/issues/3) (as far as I understand the version 3 does have them). So I was puzzled when I saw that board:

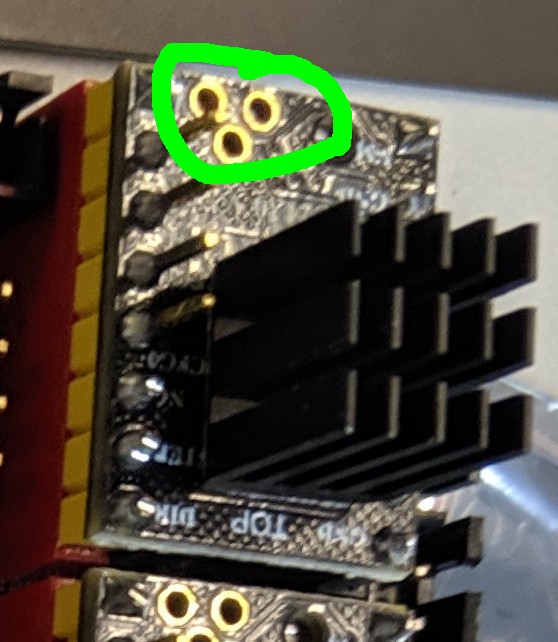
Image for post



SKR 1.3 Stepper Driver breakout with extra pin for diagnostic

There is no corresponding pin on the TMC2130 breakout board:

Image for post



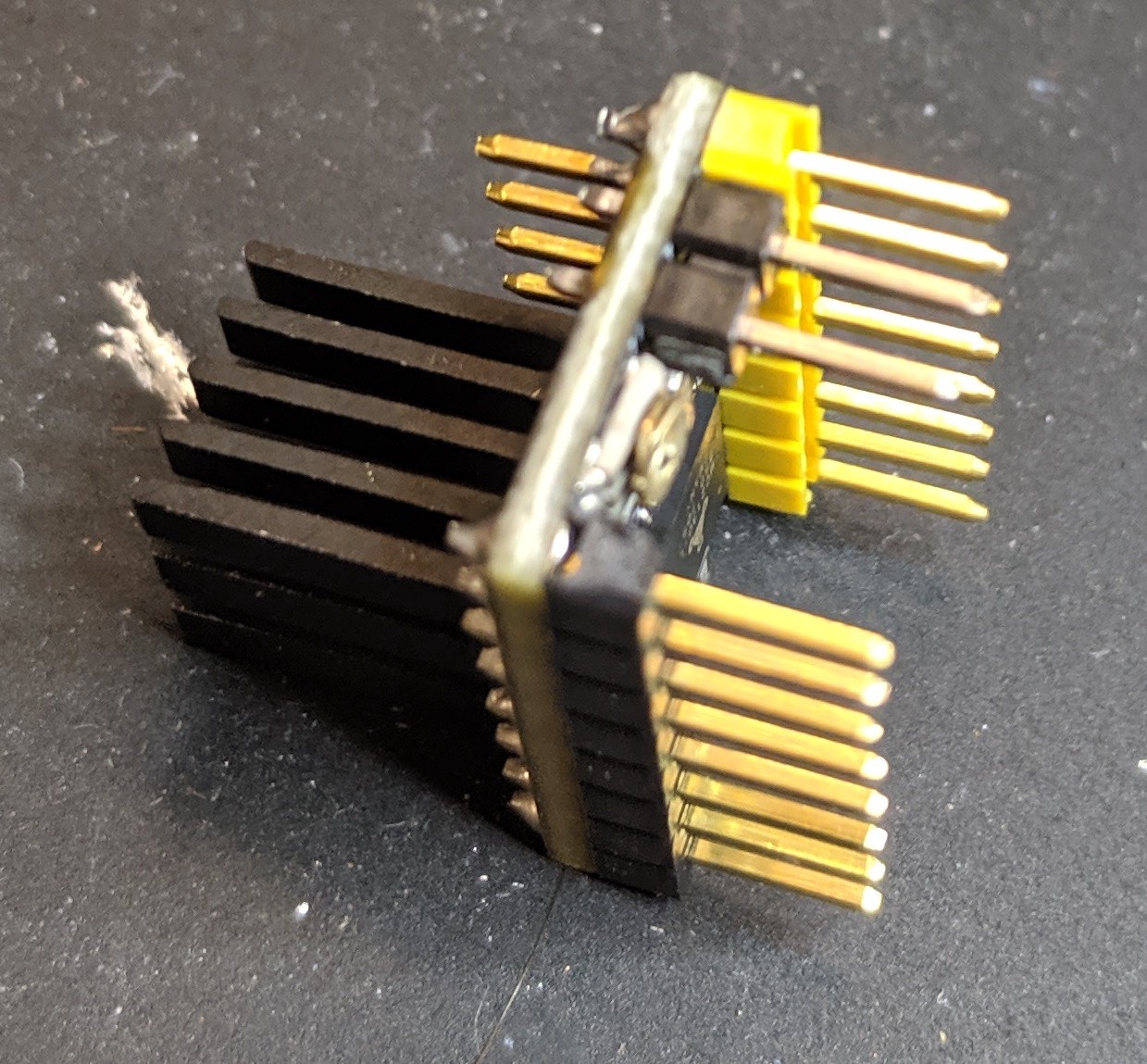
TMC2130 breakout v2. without diagnostic pin.

After a few search on the internet, including [a bug where the answer from BigTreeTech employee is pretty cryptic](https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/issues/3) and the [pinout of the original breakout board from Waterrott](https://learn.watterott.com/silentstepstick/pinconfig/), I deduced that the pin can be safely connected:

1. The innermost pin of the breakout is the diag1 pin and in SKR is connected to a jumper to connect it to the correct endstop (a bit what is [shown in that picture provided by BigTreeTech,](https://user-images.githubusercontent.com/25599056/56651290-51df0a80-66bb-11e9-8ae2-9dab84496c25.png) except that that picture shows the incorrect pin).
2. The second pin is either diag0 or V-REF depending on the breakout board and in the SKR board it is not connected anywhere. From the test I made, my stepper driver have the V-REF on that pin.

So it is safe to connect both pin even if you are not using sensorless homing since another jumper can be disconnected. Hence **I recommend to always solder a pin to diag1**. You can also connect the two pins, that is what I did:

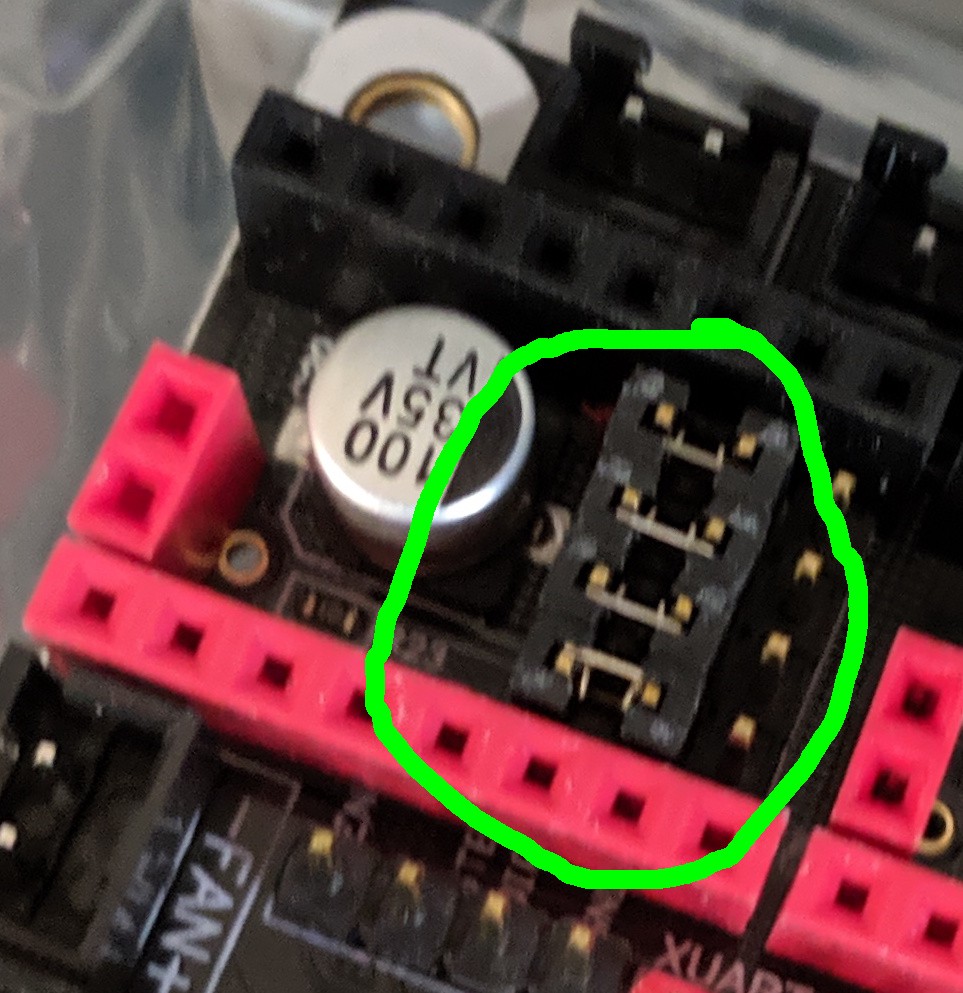
Image for post



The TMC2130 stepper driver breakout from BigTreeTech v2 with pin soldered to diag1 and vref.

My board came with the correct jumper for the SPI mode for the TMC2130, so no need to change that for me:

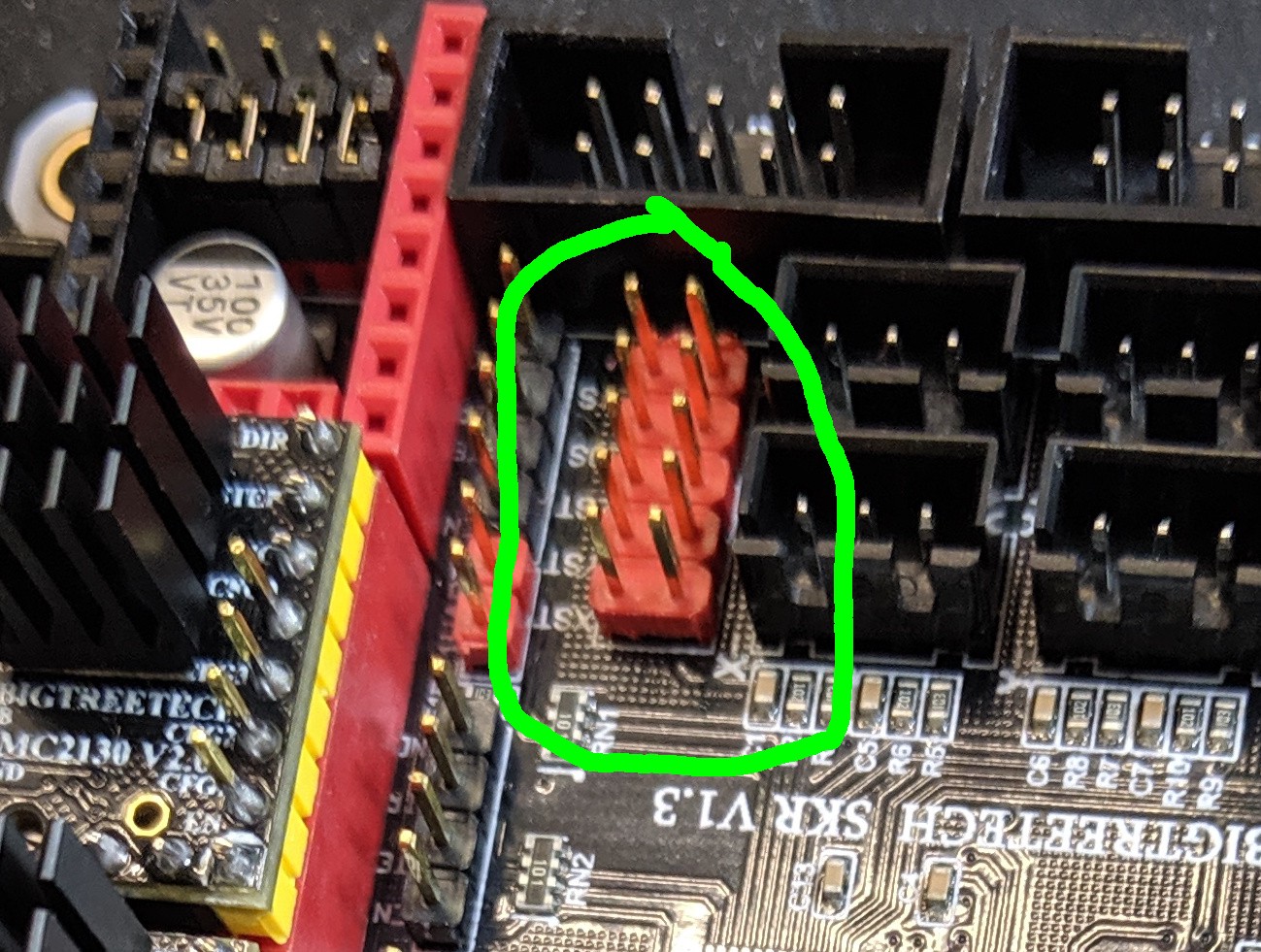
Image for post



All the jumper on the stepper driver must be connected this way to enable SPI mode for the TMC2130.

Once I was there, only one step remained. I wanted to first do a replacement of the board while still using the endstop and the BlTouch. So I needed to not connect the diagnostic pin to the endstop. Happily, the SKR board is made for it, just remove XST, YST, ZST, E0S and E1S jumper:

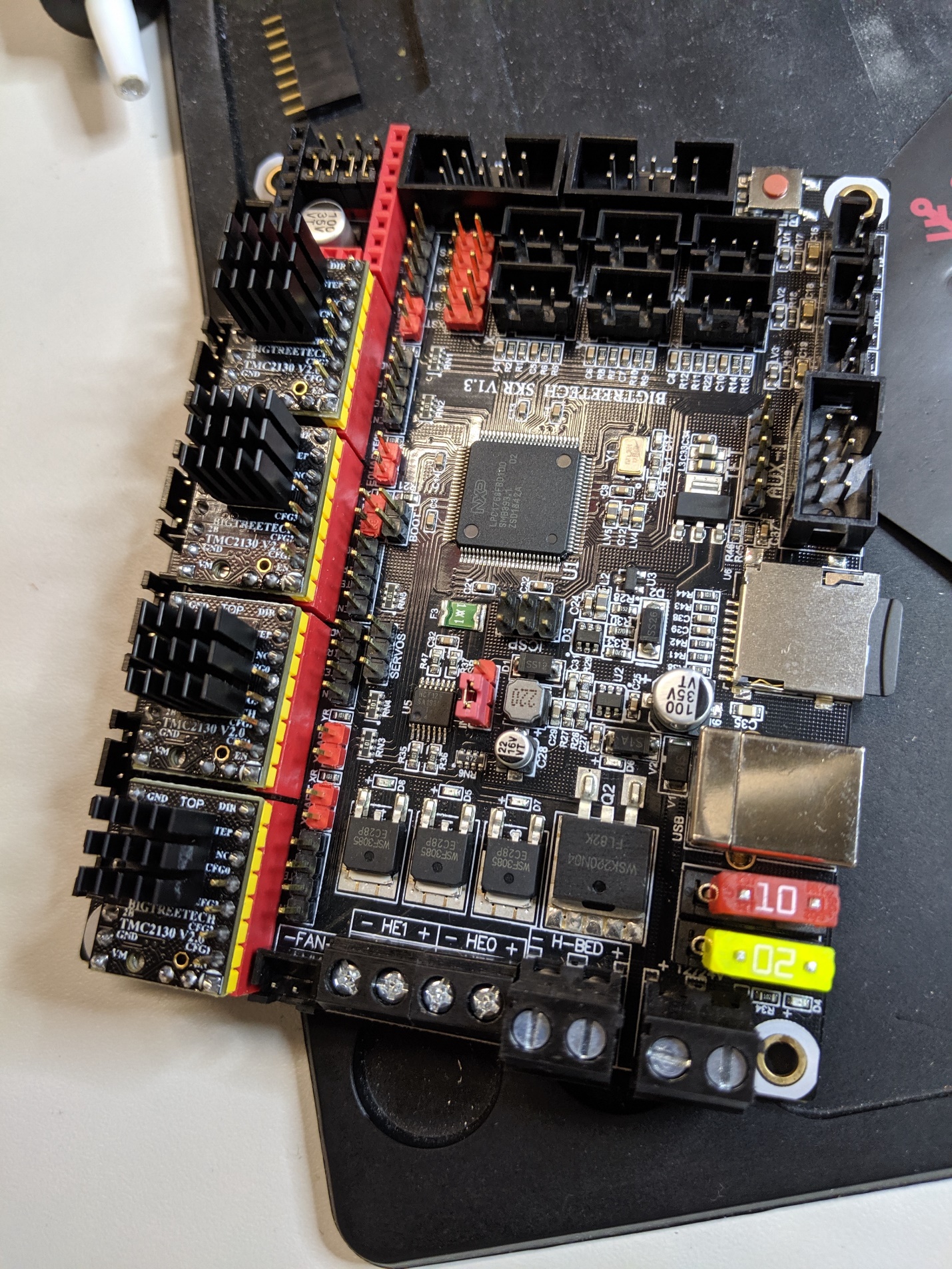
Image for post



Remove all the pin for disabling sensorless homing.

The final configured board with the TMC2130 drivers on:

Image for post



At this point I did a last check, by following [Teaching Tech advice](https://www.youtube.com/watch?v=OUadiW5QLBE), I powered the board using a 24V power supply (I had a stabilized power supply, if you don’t have one, you can use the Ender3 power supply) and verified the VREF on all stepper driver. **It is fine to put them around 1V (+- 0.2V).** The actual current setting is done in the firmware. Mine already configured correctly (most probably because I ordered them directly to BigTreeTech with the board).

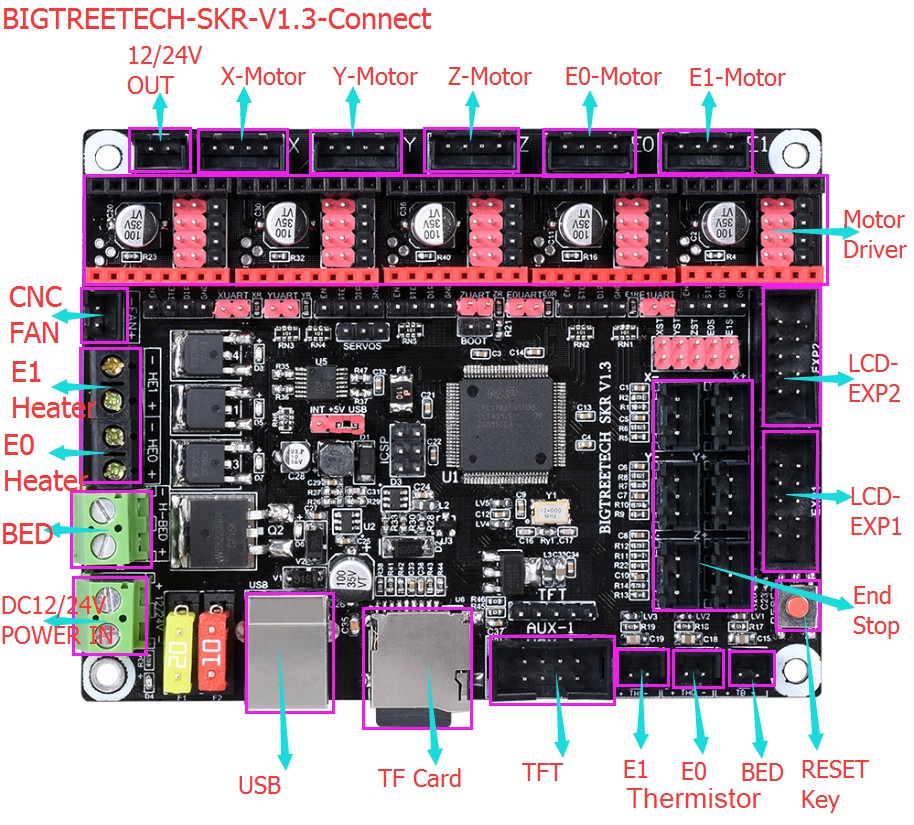
***Note:****this config can be done at any moment, so it can wait until the card is installed. It was more convenient for me to use the available power supply and to do it without the trouble of everything..*

**Connecting the SKR 1.3 to the Ender-3**

The next step is to do the connection. There was no big surprise since the connection is the same as what [Teaching Tech explained on a MKS Gen L](https://www.youtube.com/watch?v=LNdMYgwez8Y) and the SKR 1.3 wiring diagram is pretty clear.

To remember which cable is what, I added label to each cable from the Ender 3 that did not had a label on it.

Image for post

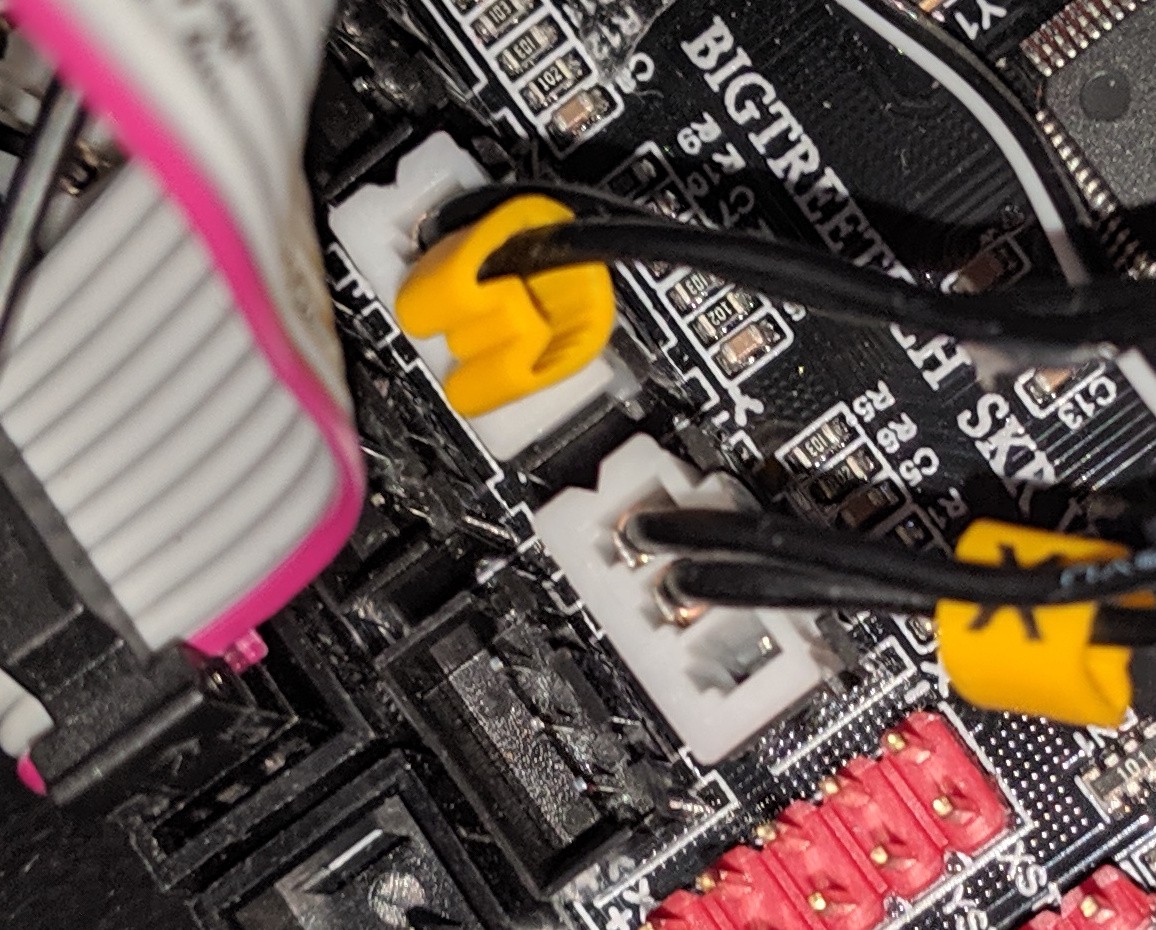


Official Wiring Diagram for the SKR1.3 [[Source](https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/Wiring%20diagram/SKR%20V1.3%20Wiring%20diagram.jpg)]

Most connection are pretty straightforward, the few exception are:

* The **hot-end fan goes in E1 heater connector**. A configuration in Marlin 2 will trigger the fan when the hot-end goes over 50C.
* The **board cover fan goes in the 24V out**. This will be trigger when the board is powered. ***TMC2130 must be cooled with both a heatsink and a fan!***
* The LCD screen goes to LCD-EXP1 and stays connected on the LCD side on the EXP3. However, this resumed the beeper that was cut thanks to the PIN27 board for the BlTouch. I simply kept the PIN27 board without the Bl-Touch on it to silence the beeper.
* For the endstop, I had extra JST connectors so I switch to use 3-pin version. [Teaching Tech proposes another technique to do it by removing one guide of the 2 pin connector so it can fit](https://www.youtube.com/watch?v=LNdMYgwez8Y).

Image for post



3-pin JST connectors for the endstop.

* The Bl-Touch endstop goes in the Z-min endstop (with the same technique as for the other endstop, for now). Contrary to the endstop, the Bl-Touch cabling direction is important, I had to connect the white wire on the exterior of the JST connector and the black on the middle. The Bl-Touch servo driver goes to the *SERVOS*pins:

Image for post



BlTouch servo cable goes to the SERVOS pin with the yellow wire toward the endstop and LCD connectors. (exact color might change depending on the BlTouch packaging).

**Configuring Marlin 2.0 for the SKR1.3 and TMC2130 on the Ender-3**

SKR 1.3 needs Marlin 2.0 (support for 32 bit boards). As I already had a configuration of Marlin [bugfix-1.1.x](https://github.com/MarlinFirmware/Marlin/tree/bugfix-1.1.x) for my Ender 3. I started by switching to [bugfix-2.0.x](https://github.com/MarlinFirmware/Marlin/tree/bugfix-2.0.x). At the moment there is very few reason to switch to Marlin 2.0 unless to support new hardware but as feature won’t get added to 1.1.x anymore, it will probably change in the near future (e.g. now it has faster probing with BlTouch).

Switching to bugfix-2.0.x was simply a matter of [copying the Ender-3 configuration files](https://github.com/damienmg/Marlin/commit/b3960af4c7932d8c047c160279fea602f89f8803), [redoing my tweaks](https://github.com/damienmg/Marlin/commit/b852c44970c0b1e8de7de8aea74ff6213f4286e5) and [setting up the BlTouch](https://github.com/damienmg/Marlin/commit/96a08031c8bed3e33239e03f2d33dd5794ef14de) (see [Teaching Tech guide](https://www.youtube.com/watch?v=sUlqrSq6LeY) again). After that my Ender-3 was working with the original Melzi board and Marlin 2.0.

From this config, setting up the SKR 1.3 in Marlin was pretty [straightforward without sensorless homing](https://github.com/damienmg/Marlin/commit/462464410650430c318c1426e6fb22b7b0aceab3).

In Configuration.h:

* Set the serial port (first on USB, second the onboard):

#define SERIAL\_PORT -1  
#define SERIAL\_PORT\_2 0

* Replace the MOTHERBOARD:

#define MOTHERBOARD BOARD\_BIGTREE\_SKR\_V1\_3

* Reverse the Bl-Touch endstop logic:

#define Z\_MIN\_ENDSTOP\_INVERTING true

* Set the driver type to TMC213*0:*

*#define X\_DRIVER\_TYPE TMC2130  
#define Y\_DRIVER\_TYPE TMC2130  
#define Z\_DRIVER\_TYPE TMC2130  
#define E0\_DRIVER\_TYPE TMC2130*

* Remove the #define SERVO0\_PIN 27, the SKR 1.3 configuration file already set it with the new pin for it.
* Remove the #define SLIM\_LCD\_MENUS, we have enough space in the SKR 1.3 for the firmware.

In Configuration\_adv.h:

* To activate the hot-end fan when the temperature rise above 50C, set E0\_AUTO\_FAN\_PIN to the second hot-end pin (P2\_04): #define E0\_AUTO\_FAN\_PIN P2\_04. **Without that, the hot-end fan will not turn on**.
* Uncomment #define LPC\_SD\_ONBOARDand #define USB\_SD\_ONBOARD, so the SD card reader can be used for printing and can be accessed by USB (hence being able to flash by USB).
* Adapt a bit the current to use the one proposed by [Teaching Tech](https://www.youtube.com/watch?v=OUadiW5QLBE):

#define X\_CURRENT 760  
#define Y\_CURRENT 760  
#define Z\_CURRENT 760  
#define E0\_CURRENT 900

* Configure the TMC to use SPI mode, and some debug information:

#define TMC\_USE\_SW\_SPI  
#define MONITOR\_DRIVER\_STATUS  
#define TMC\_DEBUG

* Finally I activated #define HYBRID\_THRESHOLD to get fast speed without overheating (going back to spreadcycle when going fast is better for torque but noiser).

The last change is the fact that we must build this software no longer with Arduino IDE but with PlatformIO. One can install the command-line PlatformIO software or preferably use the[PlatformIO IDE](https://platformio.org/platformio-ide) (I used VSCode which is the IDE I always use at home). Once this is installed, I had to tell PlatformIO that the board I use is based on a LPC1768, or it will fail on compilation. It is done by settings env\_default in the platformio.ini file:

env\_default = LPC1768

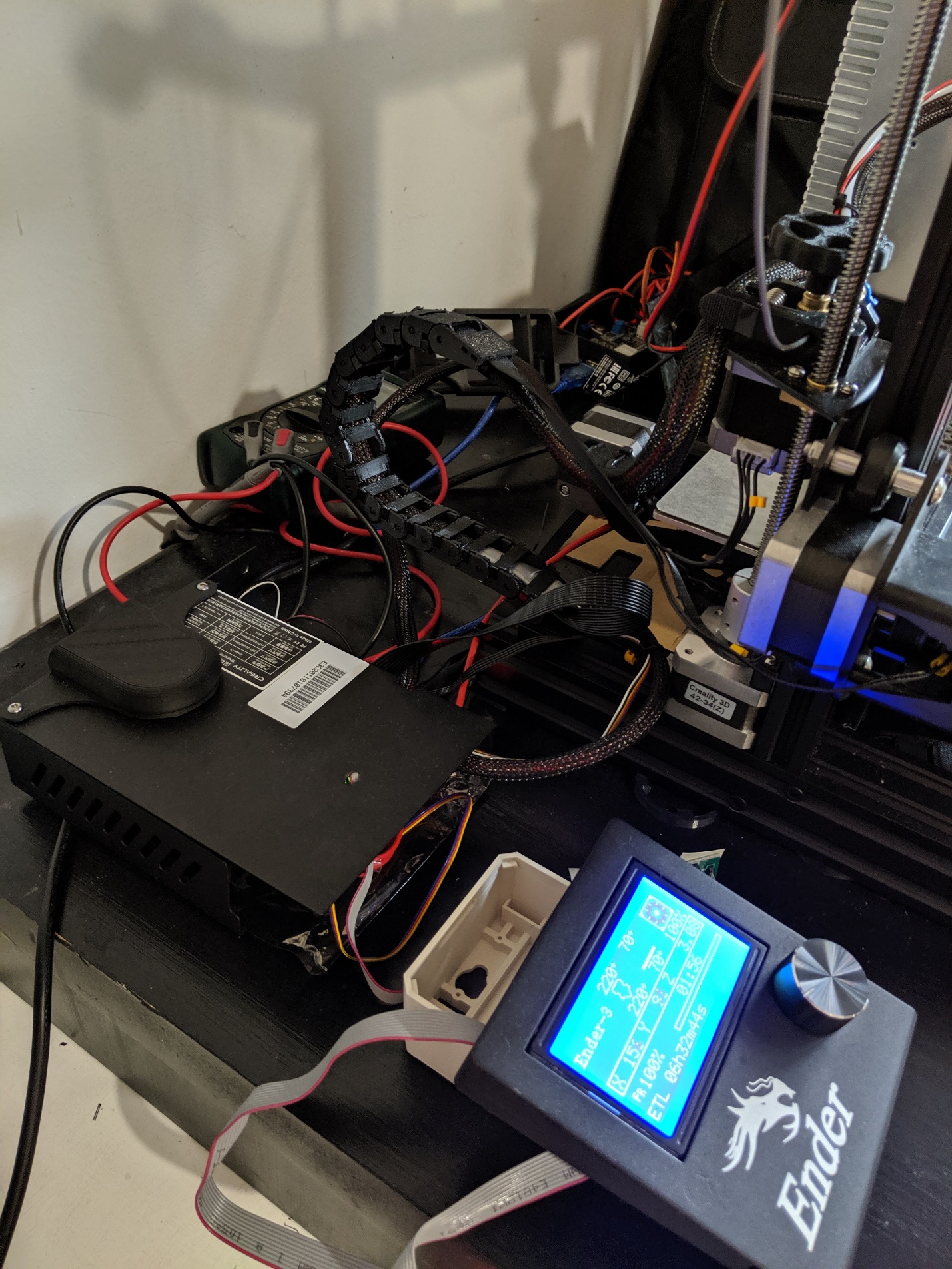
***All those change can be seen directly on the***[***commit diff on GitHub!***](https://github.com/damienmg/Marlin/commit/462464410650430c318c1426e6fb22b7b0aceab3)

Finally I just need to hit build, and copy the file under .pioenvs/LPC1768/firmware.bin to firmware.bin at the root of the SD-Card. Plugin the SD-card in the board and powering the system. The SKR1.3 will read the firmware.bin file and move it to firmware.cur upon success. Apparently, one can push the reset button on the board to force a firmware update but I never had to do it.

I am now doing the update through [Octoprint’s FirmwareUpdater](https://github.com/OctoPrint/OctoPrint-FirmwareUpdater" \t "_blank) plugin (full instruction on the webpage), I recommend it.

Now my Ender-3 is working with very silent motor, but the setup is a bit messy:

Image for post



My messy setup using the Ender-3 box for holding the fan until I build a better one.

The end? No :)

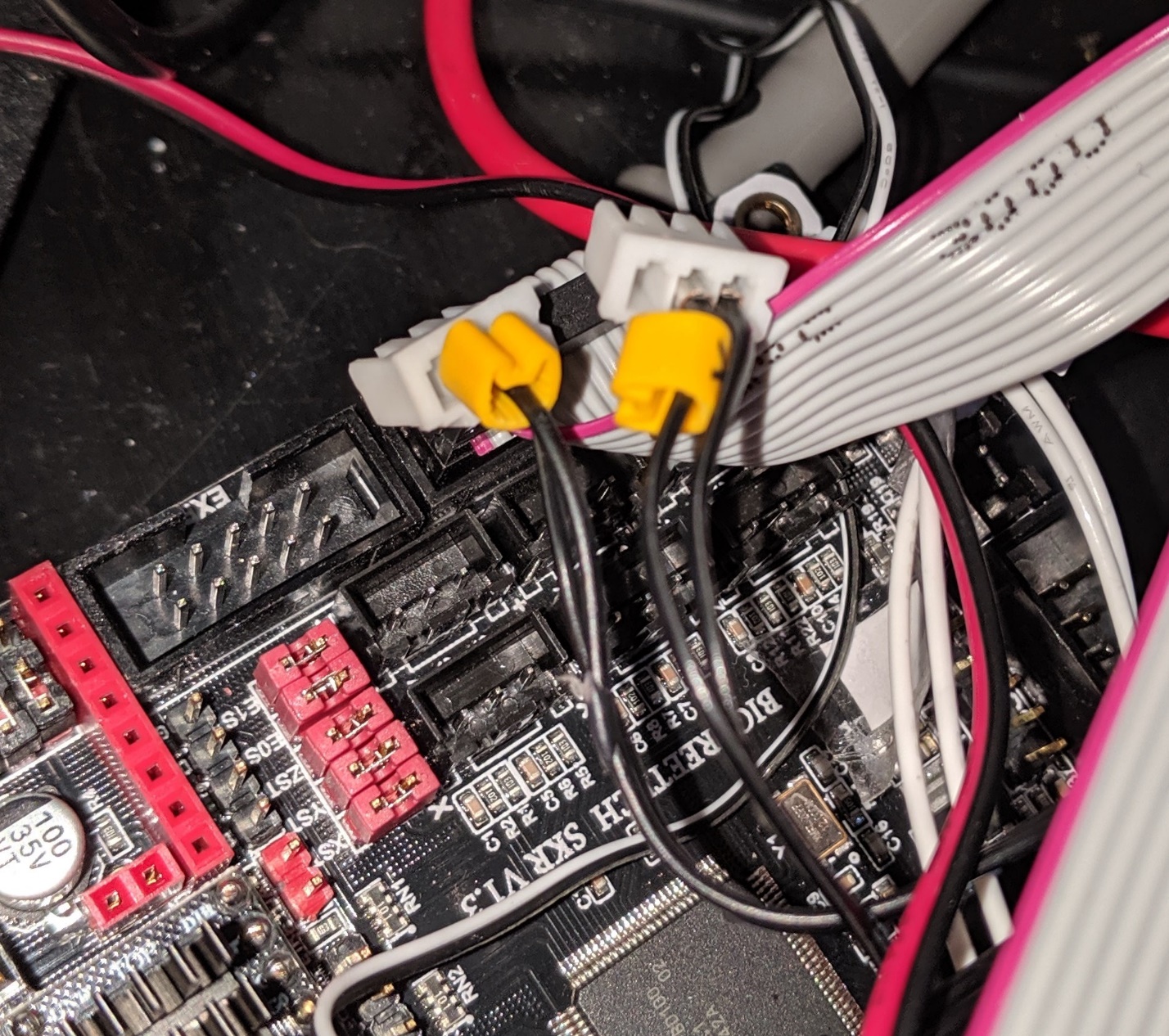
**Sensorless homing**

Remember I said I wanted to play with crash detection, so I need to wire the diagnostic pin and activate sensorless homing!

Activating sensorless homing is pretty straightforward but, if you want to get stallguard information on the z axis, it requires to plug the Bl-Touch in the Z max pin instead of the Z min. To make this in the configuration, I simply inverted the pin in the Marlin/src/pins/pins\_BIGTREE\_SKR\_V1.3.h file: [fe9ade931](https://github.com/damienmg/Marlin/commit/fe9ade931890c8dd7288cea8f5f7338d5251942c).

Once done, I can plug the Bl-Touch in Z-Min, unplug the two other endstop and add the XST, YST, ZST, E0S jumper (E1S does not matter):

Image for post



**Sensorless homing:** disconnect X and Y endstops, BlTouch on Z-max and all \*ST jumpers on.

Finally, I activated [sensorless homing in the firmware](https://github.com/damienmg/Marlin/commit/fe9ade931890c8dd7288cea8f5f7338d5251942c" \t "_blank):

* Invert the endstops in Configuration.h:

#define X\_MIN\_ENDSTOP\_INVERTING true  
#define Y\_MIN\_ENDSTOP\_INVERTING true

* Activate sensorless homing:

#define SENSORLESS\_HOMING

* Set the stall sensitivity to 10, which I observed to be good (this can be live tuned with the M914 gcode):

#define X\_STALL\_SENSITIVITY 10  
#define Y\_STALL\_SENSITIVITY 10

Once uploading the firmware, I had to somehow force store those new value for the sensitivity in the EEPROM so I used Octoprint’s terminal to send the corresponding command:

M914 X10 Y10 ; Set X and Y stall sensitivity to 10  
M500 ; Save to eeprom

Another way to do it would be to force the reset of the EEPROM from the firmware, but this reset also the Z offset of my BlTouch:

M502 ; Load factory default  
M500 ; Save to eeprom

And this is the current state of my Ender-3. When I will have a bit more time, I’ll dive into doing crash detection.

**Some more…**

* I’ll keep on **posting my Marlin configuration on Github**, so you can follow the [dmg-ender3-skr13](https://github.com/damienmg/Marlin/tree/dmg-ender3-skr13) branch if you are interested.
* **Higher microsteps:** I tried to switch to higher microsteps, the TMC2130 driver are able to go up to 256 native microsteps but I observed [funny behavior](https://github.com/MarlinFirmware/Marlin/issues/14114). According to Marlin authors, it is recommended to not go too high also because it needs to send an event per microsteps. From my observation, going up is making the motor a bit less noiser in spreadcycle (still silent in stealthchop).
* **Support:**BigTreeTech support is really low, and like Creality, they don’t really understand the OpenSource movement and just do dump of files. They hardly engage with the community. Happily, people in the [Facebook group](https://www.facebook.com/groups/505736576548648/) are really helping, thanks a lot to all the folks on this group!
* **More resources:**A resource I have not linked so far is [Chris Riley’s SKR 1.3 video](https://www.youtube.com/watch?v=duNHOPlh2Pg). It was released after I installed my own card but I definitely recommend taking a look at it.