

Stepper Driver Configurations for SKR PRO V1.1 Board



by

@GadgetAngel



Based on Work by @rfulling

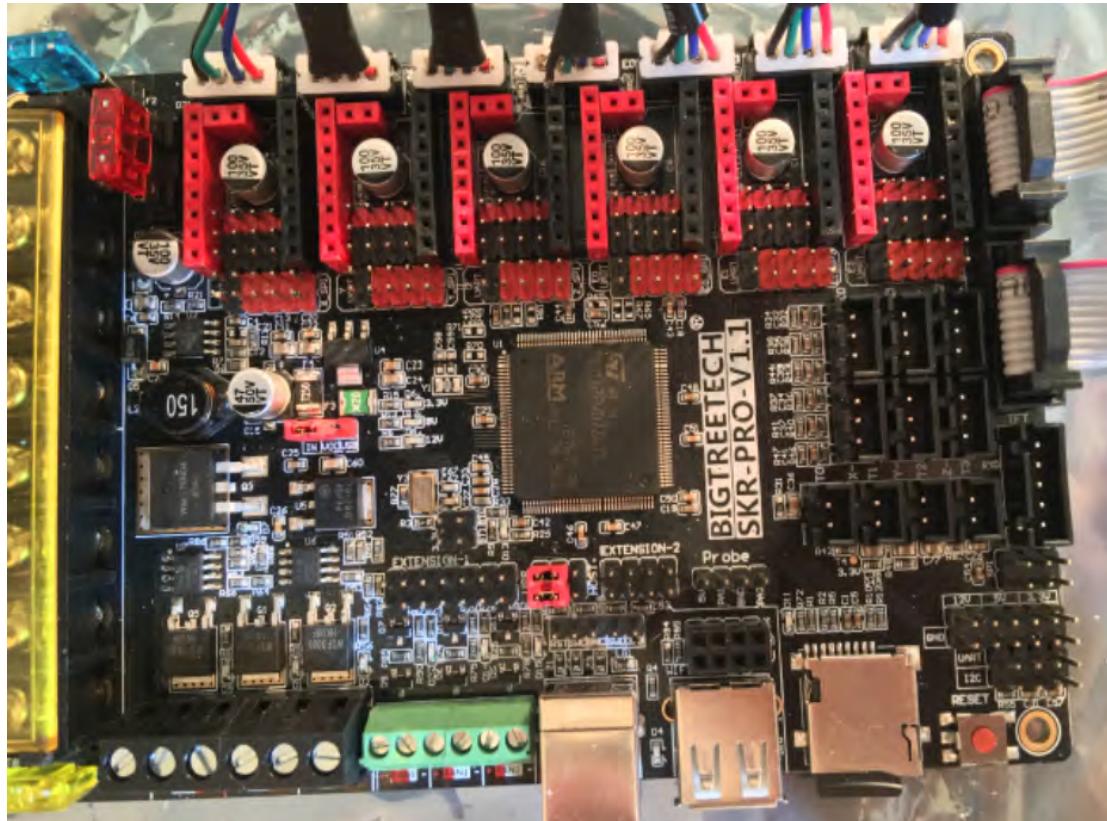


Table of Contents

1.	A4988.....	3 - 5
2	The (latest release of) Marlin Setup for A4988 Drivers.....	6 - 9
3.	DRV8825.....	10 - 13
4.	The (latest release of) Marlin Setup for DRV8825 Drivers.....	14- 18
5.	BIQU LV8729.....	19 - 22
6.	The (latest release of) Marlin Setup for BIQU LV8729 Drivers.....	23 - 27
7.	FYSETC LV8729.....	28 - 31
8.	The (latest release of) Marlin Setup for FYSETC LV8729 Drivers.....	32 - 36
9.	Lerdge LV8729.....	37- 40
10.	The (latest release of) Marlin Setup for Lerdge LV8729 Drivers.....	41 - 45
11.	FYSETC ST820.....	46 - 49
12.	The (latest release of) Marlin Setup for FYSETC ST820 Drivers.....	50 - 56
13.	BIQU ST820.....	57 - 60
14.	The (latest release of) Marlin Setup for BIQU ST820 Drivers.....	61 - 67
15.	POLOLU ST820.....	68 - 71
16.	The (latest release of) Marlin Setup for POLOLU ST820 Drivers.....	72 - 78
17.	POLOLU MP6500.....	79 - 81
18.	The (latest release of) Marlin Setup for POLOLU MP6500 Drivers.....	82 - 86
19.	POLOLU TB67S249FTG.....	87 - 90
20.	The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers.....	91 - 95
21.	BIQU MKS TMC2100.....	96 - 99
22.	The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers.....	100 - 104

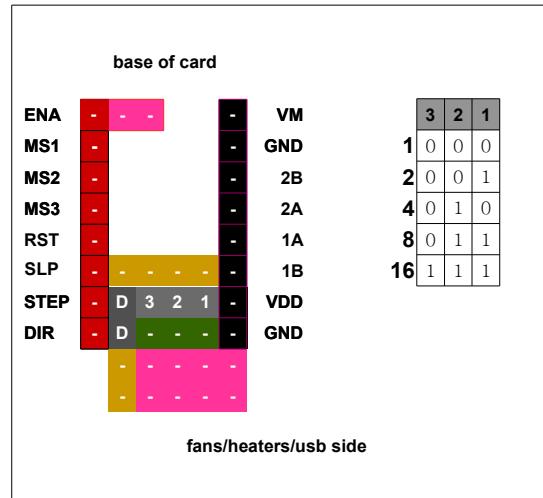
Table of Contents

23. BIQU TMC2130 Stand Alone Mode.....	105- 108
24. The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode	109 - 113
25. BIQU TMC2130 SPI Mode.....	114 - 115
26. The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode	116 - 128
27. BIQU TMC2208 V3.0 Stand Alone Mode.....	129 - 131
28. The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode	132- 136
29. BIQU TMC2208 V3.0 One Time Programming (OTP) Mode.....	137 - 139
30. The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode	140 - 144
31. BIQU TMC2208 V3.0 UART Mode.....	145 - 146
32. The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode	147 - 156
33. FYSETC TMC2208 V1.2 Stand Alone Mode.....	157 - 159
34. The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode	160 - 164
35. FYSETC TMC2208 V1.2 One Time Programming (OTP) Mode.....	165 - 167
36. The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode.	168 - 172
37. FYSETC TMC2208 V1.2 UART Mode.....	173 - 174
38. The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode	175 - 184
39. BIQU TMC2209 V1.2 Stand Alone Mode for StealthChop.....	185 - 187
40. The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop	188 - 192
41. BIQU TMC2209 V1.2 Stand Alone Mode for SpreadCycle.....	193 - 195
42. The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle	196 - 200
43. BIQU TMC2209 V1.2 UART Mode.....	201 - 202
44. The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode	203 - 215

Table of Contents

45. BIQU TMC5160 V1.2 SPI Mode.....	216 - 217
46. The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode	218 - 231
47. Appendix - How to adjust the Vref on a Stepper Motor Driver board using the Potentiometer.....	232 - 233
48. Appendix B - For the TMC drivers whats the difference between Stand alone Mode and ("UART" or "SPI ") mode?.....	234 - 234
49. Appendix B - How To Calculate Vref For Stepper Motor Divers.....	234 - 234
50. Appendix C - The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers.....	235 - 243

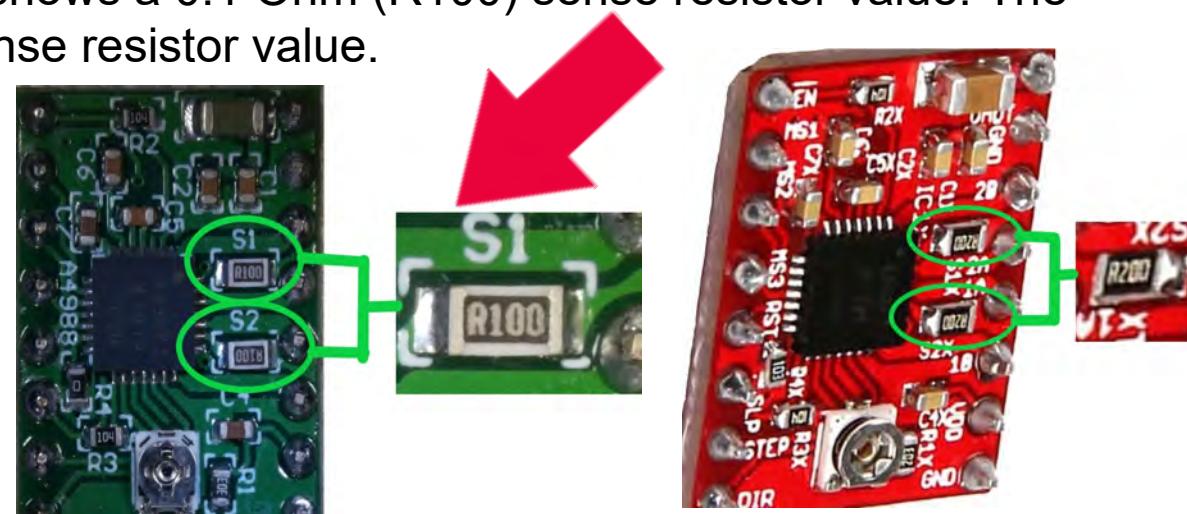
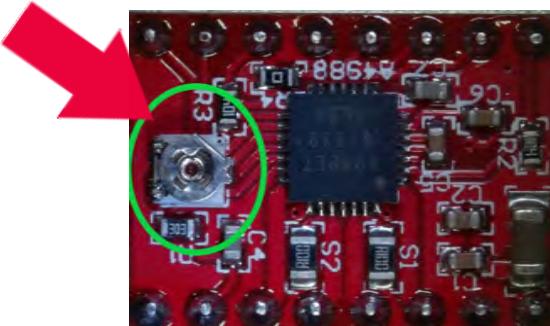
A4988



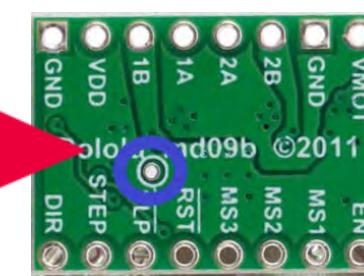
Driver Chip	MS3	MS2	MS1	Subdivision	Excitation Mode
A4988 Maximum 16 subdivision 35V 2A	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	2W1-2 Phase
	H	H	H	1/16	4W1-2 Phase
Driving current calculation formula $Rs=0.1\Omega$	$I_{max} = V_{ref} / (8 * Rs)$			$V_{REF} = 8 \cdot I_{MAX} \cdot R_S$	
	Note: Use 90% of the calculated Vref when tuning the stepper driver board				

Note: Not all driver boards for the A4988 use the same current sense resistor (Rs), check your driver board for the value of the Rs resistor by examining the board, as shown in the **GREEN** box below. The GREEN PCB shows a 0.1 Ohm (R100) sense resistor value. The RED PCB shows a 0.2 Ohm (R200) sense resistor value.

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

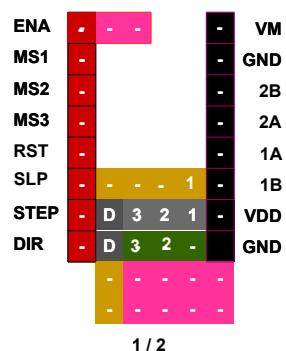
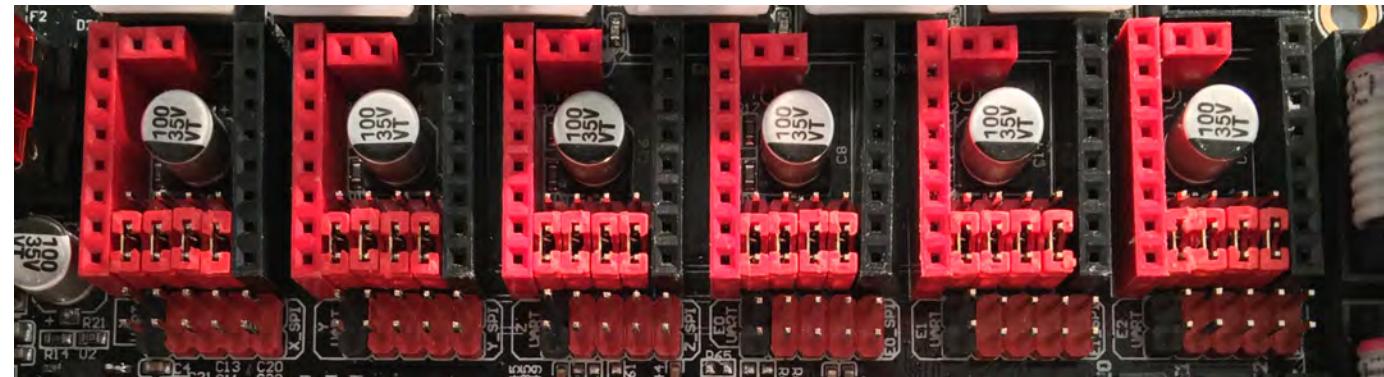
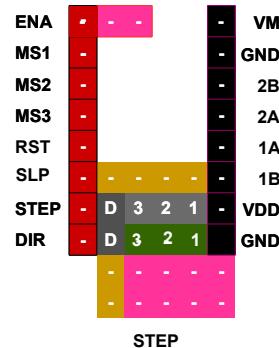


Note: "Vref Test point" location is on the bottom of the driver board, as shown in **BLUE**

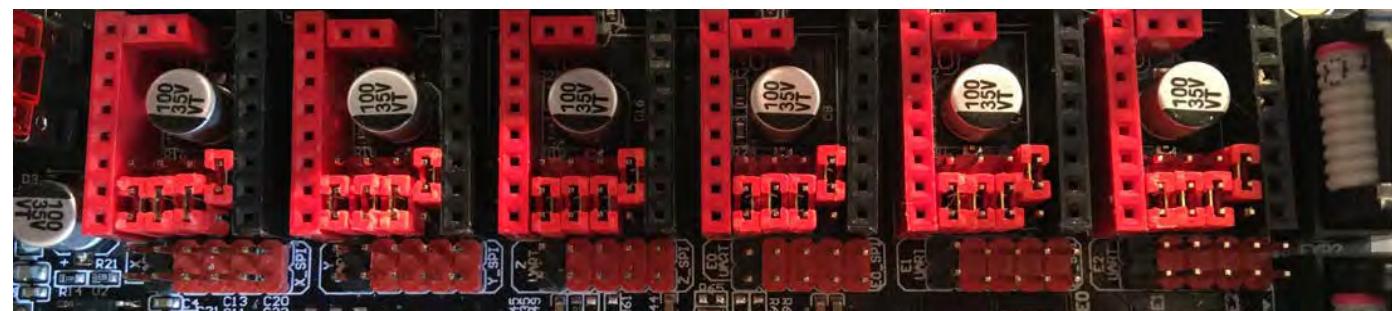


A4988

Note: The "D" jumper MUST be SET!

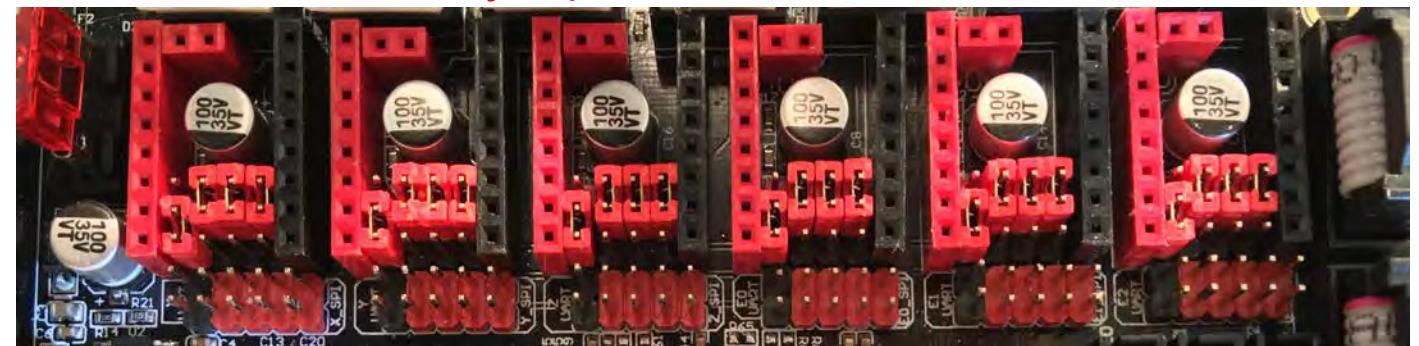
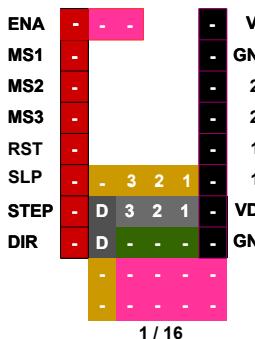
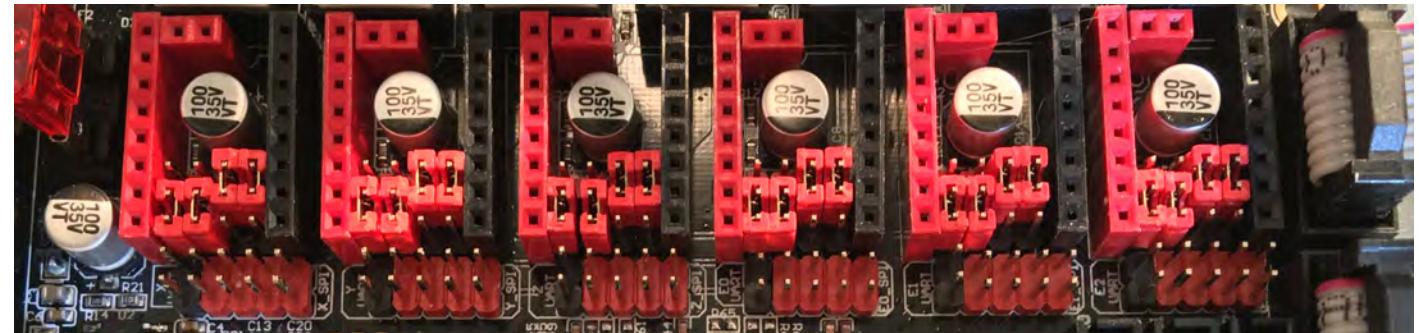
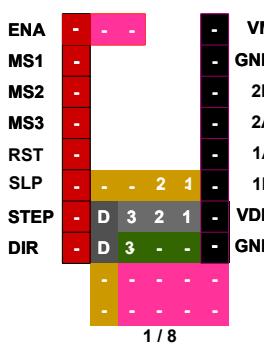
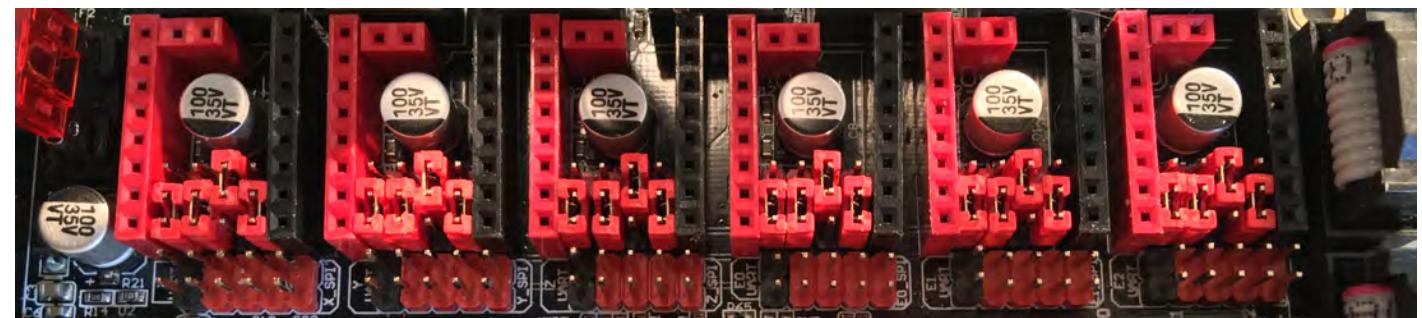
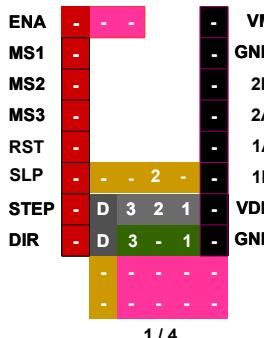


Note: The "D" jumper MUST be SET!



A4988

Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for A4988 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for A4988 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using A4988 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use A4988 drivers. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

```

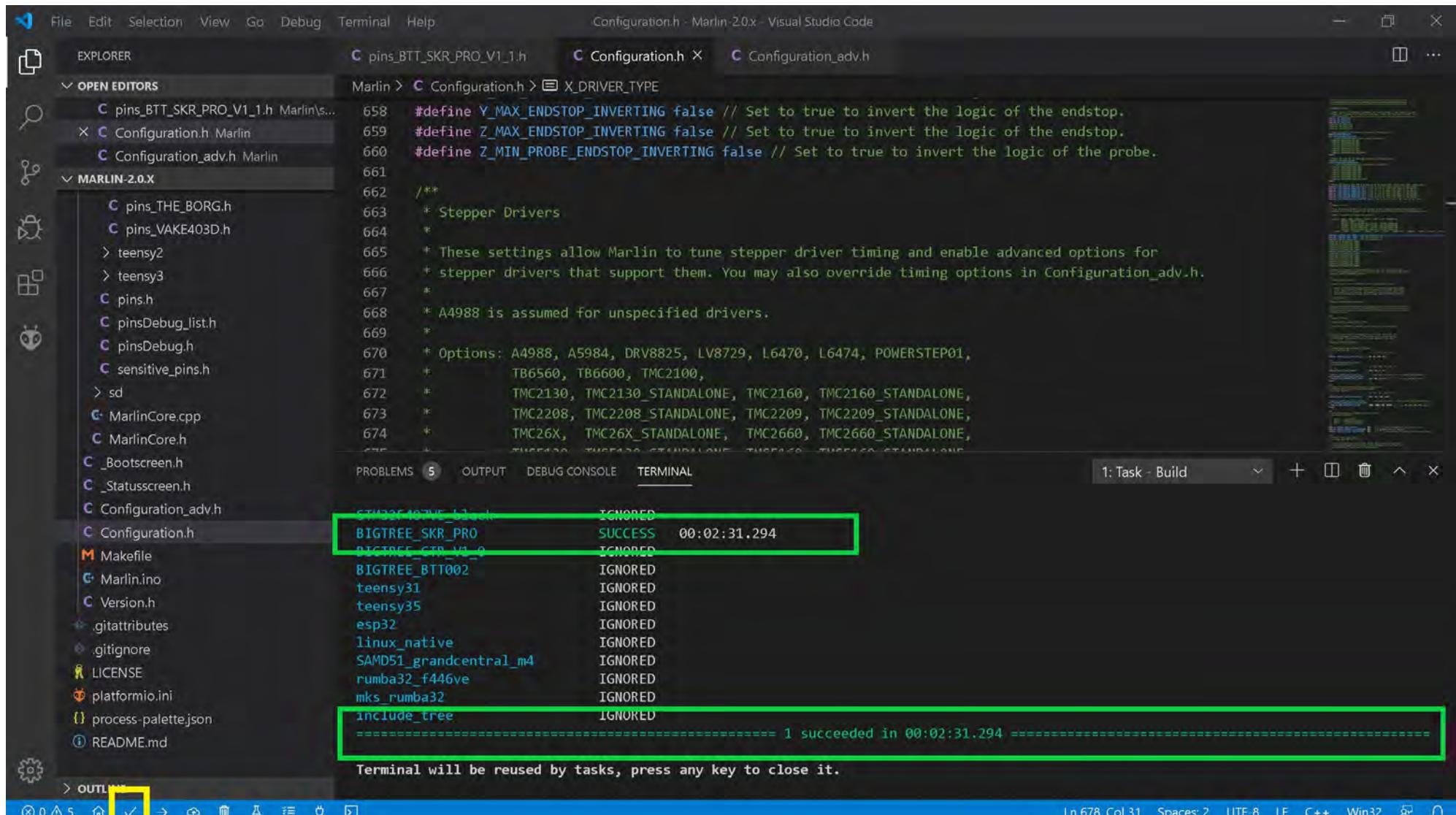
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h Marlin
Marlin > Configuration_adv.h Marlin
Marlin > pins_BTT_SKR_PRO_V1_1.h Marlin\...
MARLIN-2.0.X
C lcdprint.h
C thermistornames.h
C ultralcd.cpp
C ultralcd.h
> libs
> module
> pins
> sd
C MarlinCore.cpp
C MarlinCore.h
C Bootscreen.h
C Statusscreen.h
C Configuration_adv.h
C Configuration.h
M Makefile
C Marlin.ino
C Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686, Col 54 Spaces: 2 UTF-8 LF C++ Win32
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC256, TMC256_STANDALONE, TMC2560, TMC2560_STANDALONE
675 * TMC2568, TMC2568_STANDALONE, TMC25680, TMC25680_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
677 */
#define X_DRIVER_TYPE A4988 //JTM was commented out
#define Y_DRIVER_TYPE A4988 //JTM was commented out
#define Z_DRIVER_TYPE A4988 //JTM was commented out
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE A4988 //JTM was commented out
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988
694 // Enable this feature if all enabled endstop pins are interrupt-capable.
695

```

- Go to the next page.

The (latest release of) Marlin Setup for A4988 Drivers

- The end of Marlin setup for A4988 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled sucessfully you will see the following messages, as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the following details:

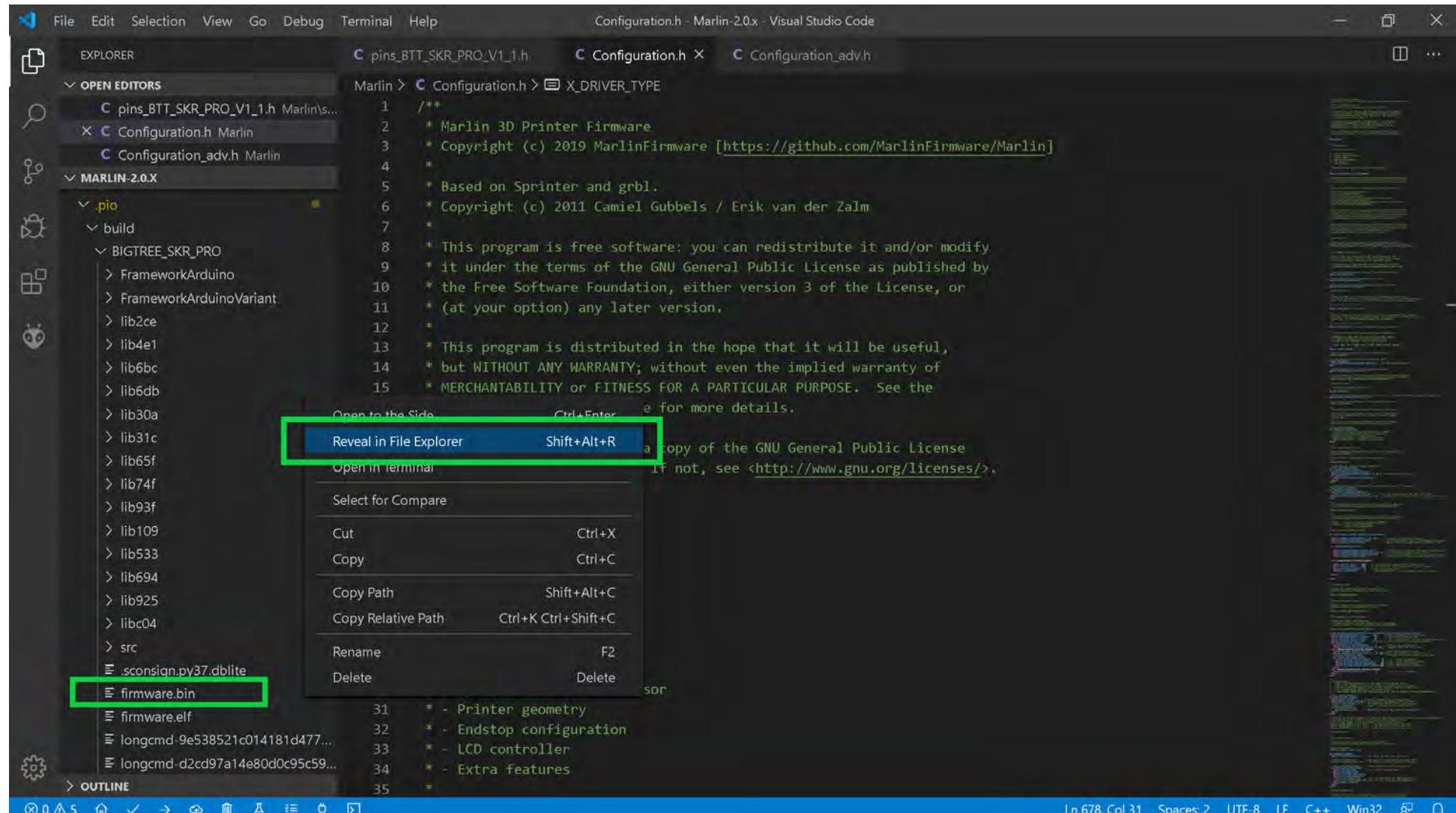
- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):**
 - OPEN EDITORS: pins_BTT_SKR_PRO_V1_1.h, Configuration.h (Marlin), Configuration_adv.h (Marlin)
 - MARLIN-2.0.X: pins_THE_BORG.h, pins_VAKE403D.h, teensy2, teensy3, pins.h, pinsDebug.list.h, pinsDebug.h, sensitive_pins.h, sd, MarlinCore.cpp, MarlinCore.h, Bootscreen.h, Statusscreen.h, Configuration_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md
- Terminal Tab:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected).
- Terminal Output (highlighted in green):**

```
CTB320_IOTHE_BORG
BIGTREE_SKR_PRO           IGNORED
BIGTREE_CTR_V1_0           IGNORED
BIGTREE_BTT002             IGNORED
teensy31                   IGNORED
teensy35                   IGNORED
esp32                      IGNORED
linux_native                IGNORED
SAMD51_grandcentral_m4    IGNORED
rumba32_f446ve              IGNORED
mks_rumba32                 IGNORED
include_tree                 IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```
- Bottom Status Bar:** OUTLINE (highlighted with a yellow box), 0 4 5, a checkmark icon, Terminal will be reused by tasks, press any key to close it, Ln 678, Col 31, Spaces: 2, UTF-8, LF, C++, Win32, etc.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for A4988 Drivers

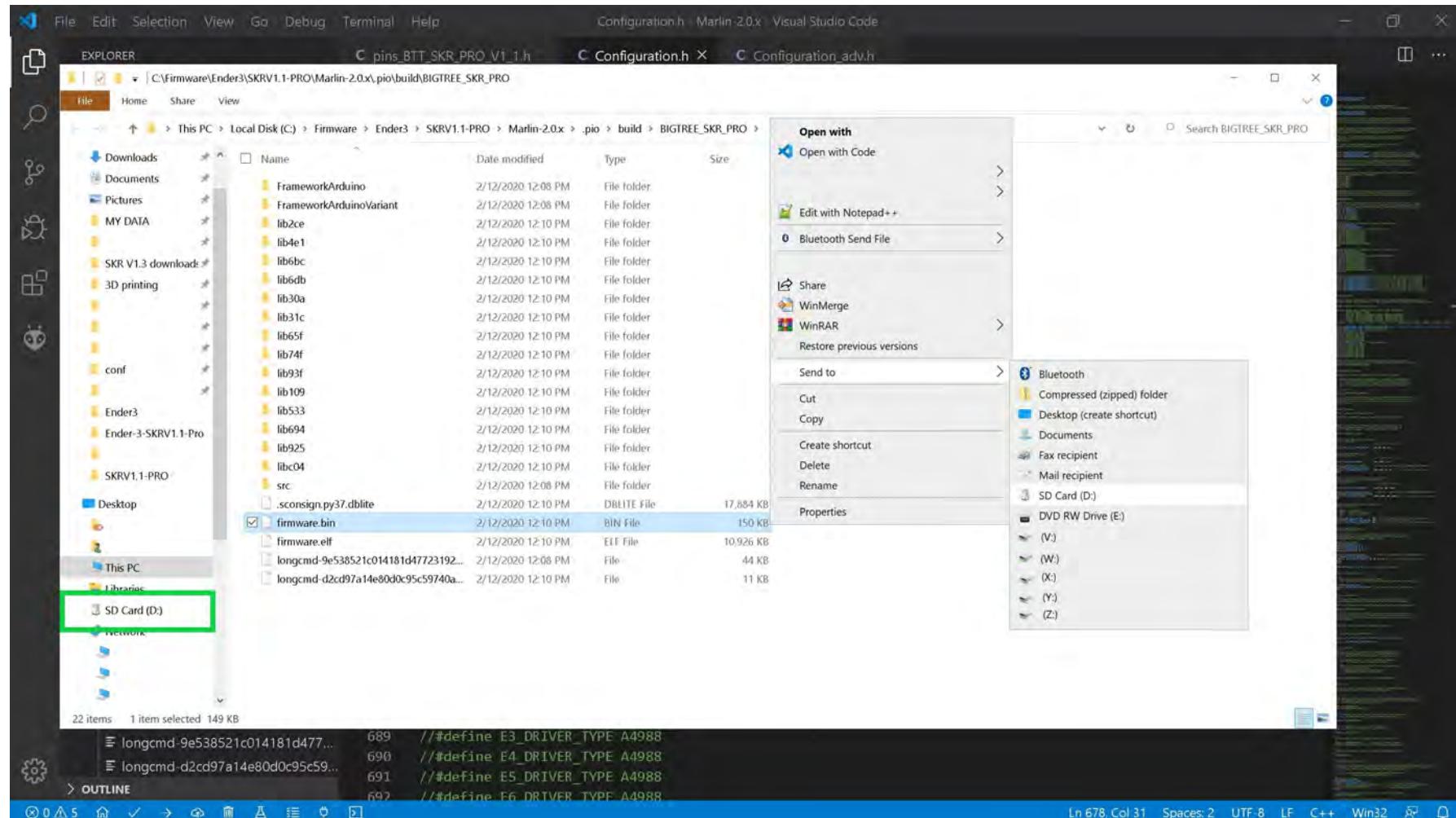
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file explorer window.



- Go to the next page.

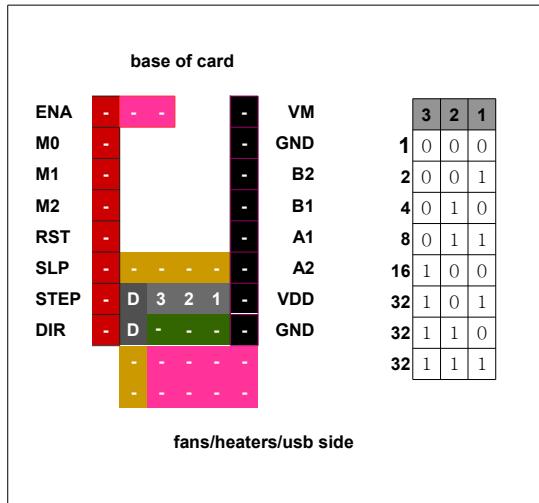
The (latest release of) Marlin Setup for A4988 Drivers

- Ensure your micro-SD card, that came with your SKR PRO V1.1 board, is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

DRV8825

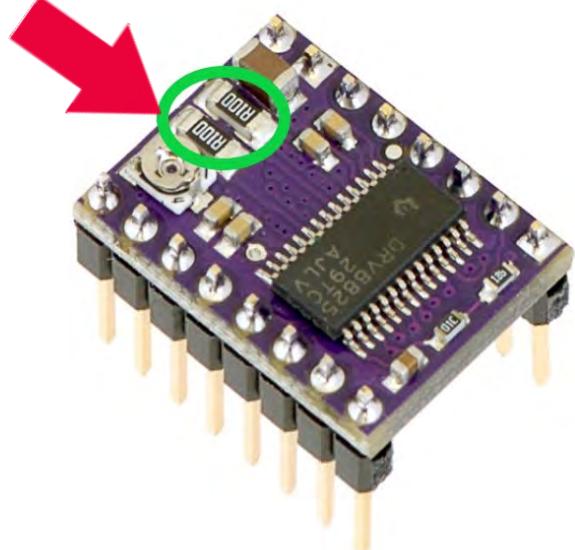


Driver Chip	MODE2	MODE1	MODE0	Subdivision	Excitation Mode
DRV8825 Maximum 32 subdivision 8.2V-45V 2.5A at 24V T=25°C	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	
	H	L	L	1/16	
	H	L	H	1/32	Three Different Ways to Set 1/32 Steps with DRV8825 Stepper Driver
	H	H	L	1/32	
	H	H	H	1/32	
Driving current calculation formula $R_s = 0.1\Omega$		$I_{MAX} = \frac{V_{REF}}{5 \cdot R_s}$			$V_{REF} = \frac{I_{MAX}}{5 \cdot R_s}$
Note: Use 90% of the calculated Vref when tuning the stepper driver board					

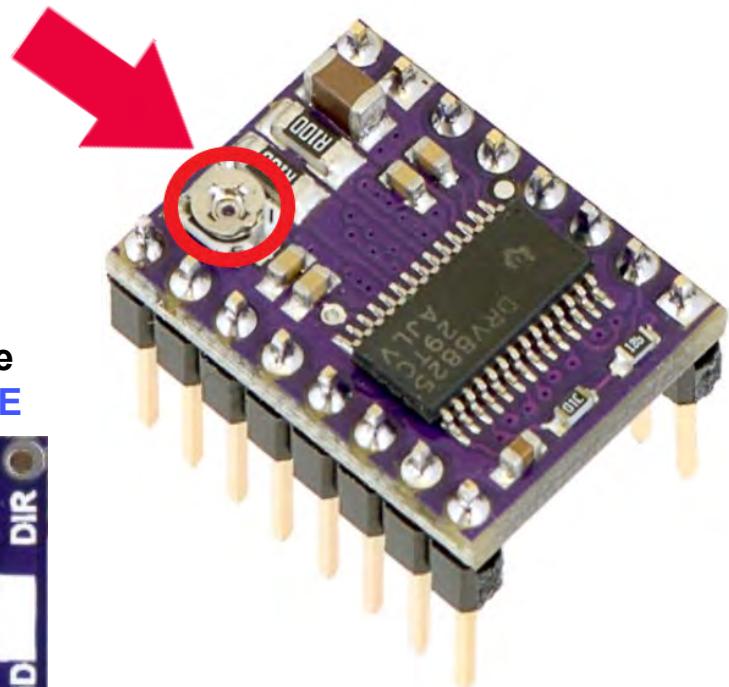
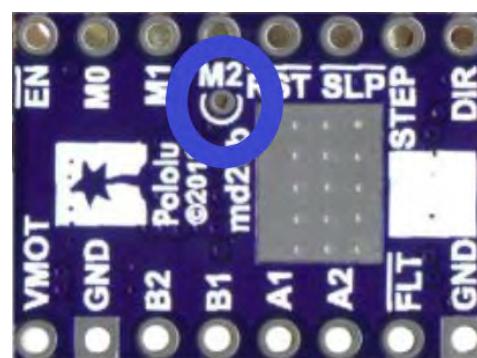
NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to adjust your Vref.

[See the appendix](#) for instructions on how to set the Vref on a driver board.

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.



Note: "Vref Test point"
location is on the bottom of the driver board, as shown in BLUE



DRV8825

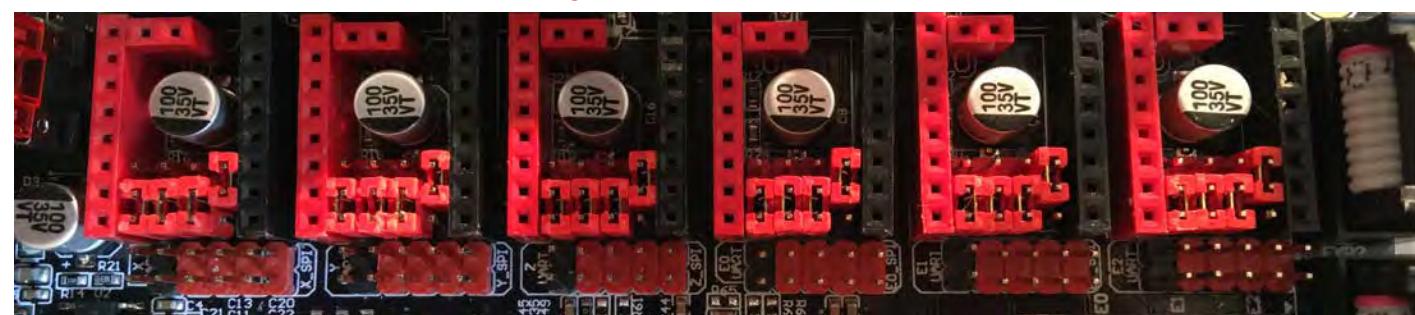
Note: The "D" jumper MUST be SET!

ENA	-	-	-	-	-	-	-	VM
M0	-	-	-	-	-	-	-	GND
M1	-	-	-	-	-	-	-	B2
M2	-	-	-	-	-	-	-	B1
RST	-	-	-	-	-	-	-	A1
SLP	-	-	-	-	-	-	-	A2
STEP	-	D	3	2	1	-	-	VDD
DIR	-	D	3	2	1	-	-	GND
STEP								



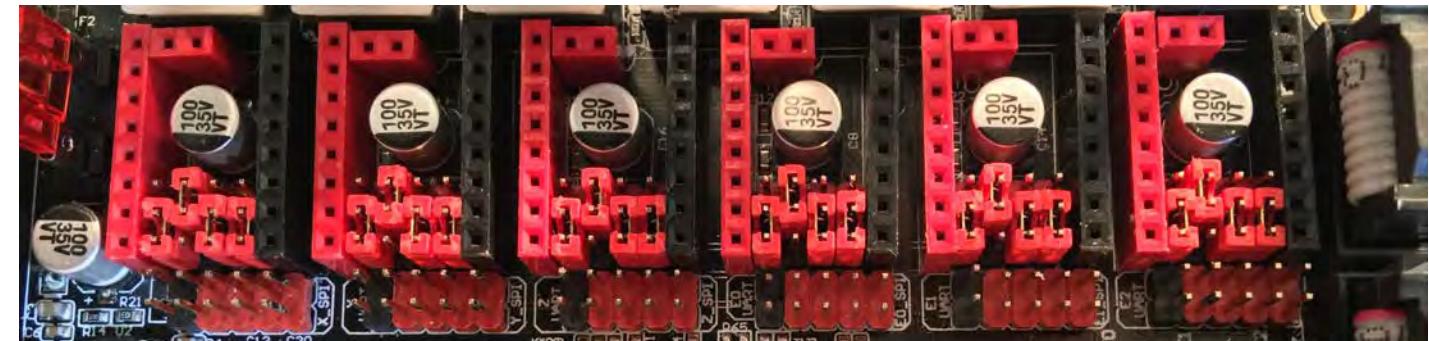
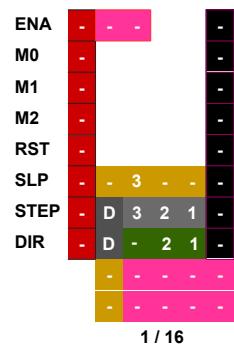
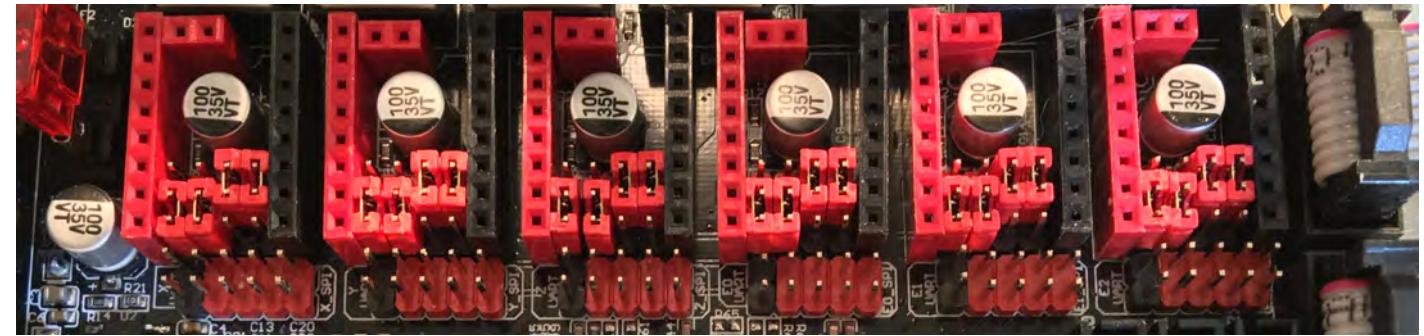
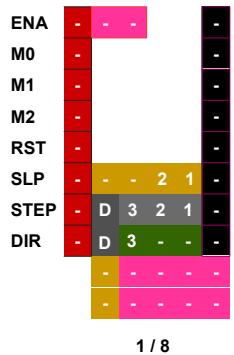
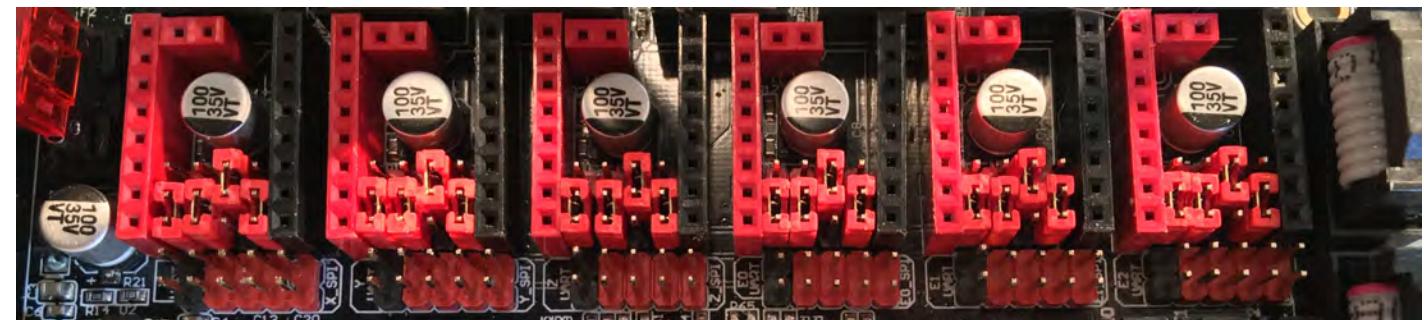
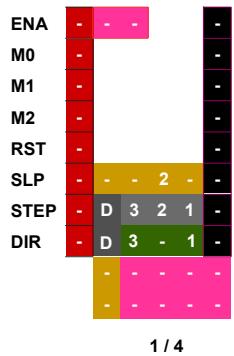
ENA	-	-	-	-	-	-	-	VM
M0	-	-	-	-	-	-	-	GND
M1	-	-	-	-	-	-	-	B2
M2	-	-	-	-	-	-	-	B1
RST	-	-	-	-	-	-	-	A1
SLP	-	-	-	-	-	1	-	A2
STEP	-	D	3	2	1	-	-	VDD
DIR	-	D	3	2	-	-	-	GND
1 / 2								

Note: The "D" jumper MUST be SET!



DRV8825

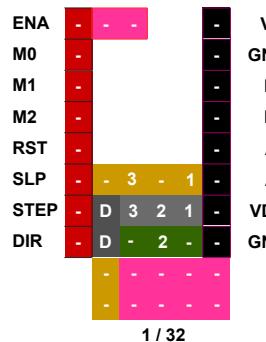
Note: The "D" jumper MUST be SET!



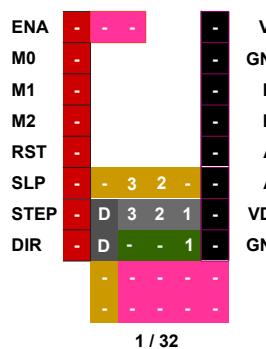
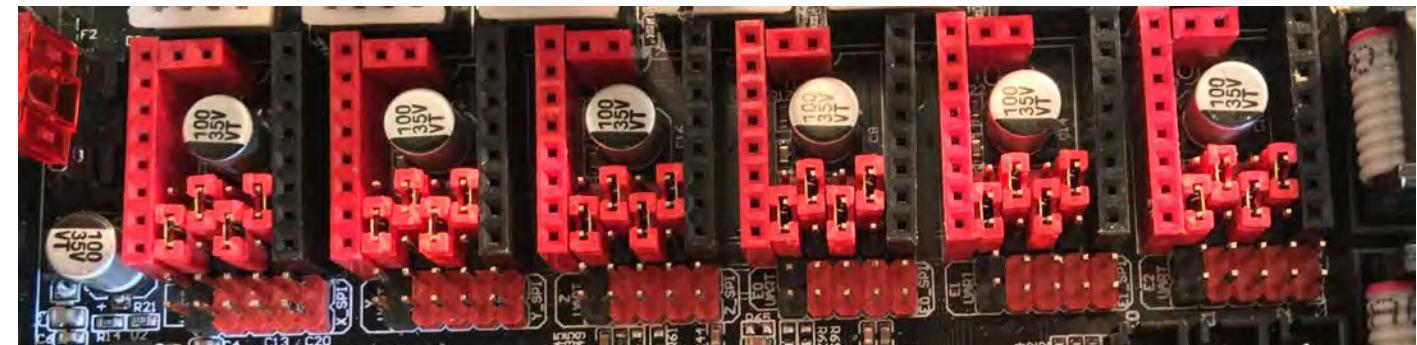
Note: The "D" jumper MUST be SET!

DRV8825

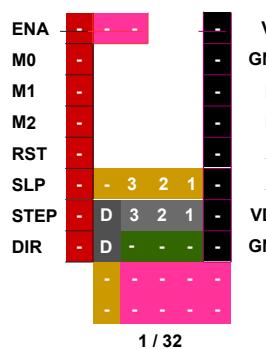
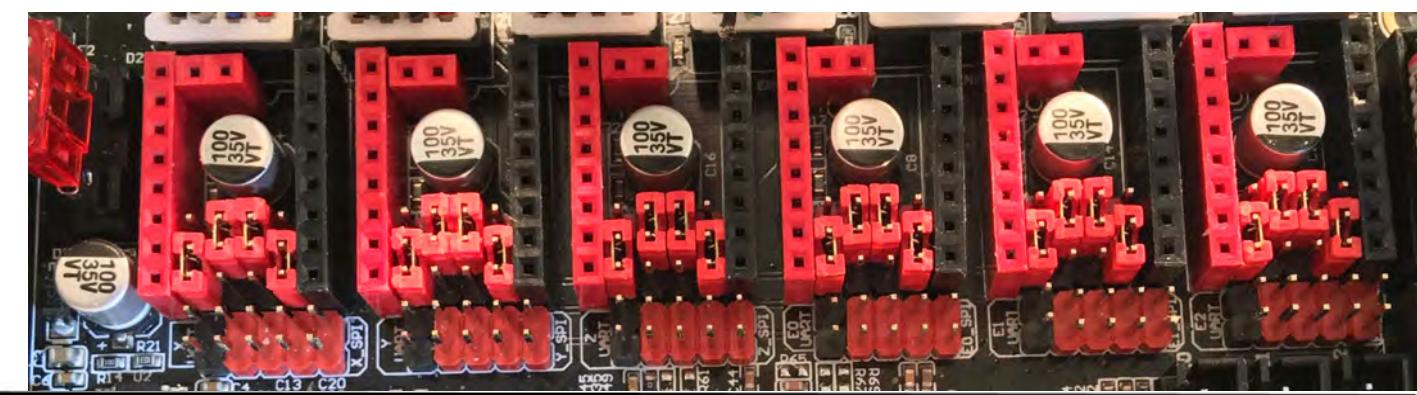
Note: All THREE of these settings will work for 1/32, choose your preference!!



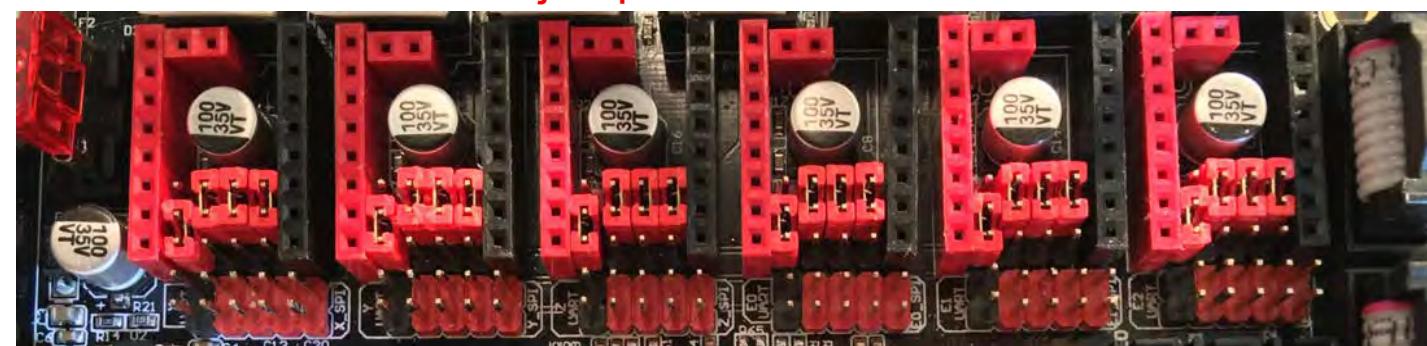
Note: The "D" jumper MUST be SET!



Note: The "D" jumper MUST be SET!



Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for DRV8825 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for DRV8825 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using DRV8825 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use DRV8825 drivers. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

```

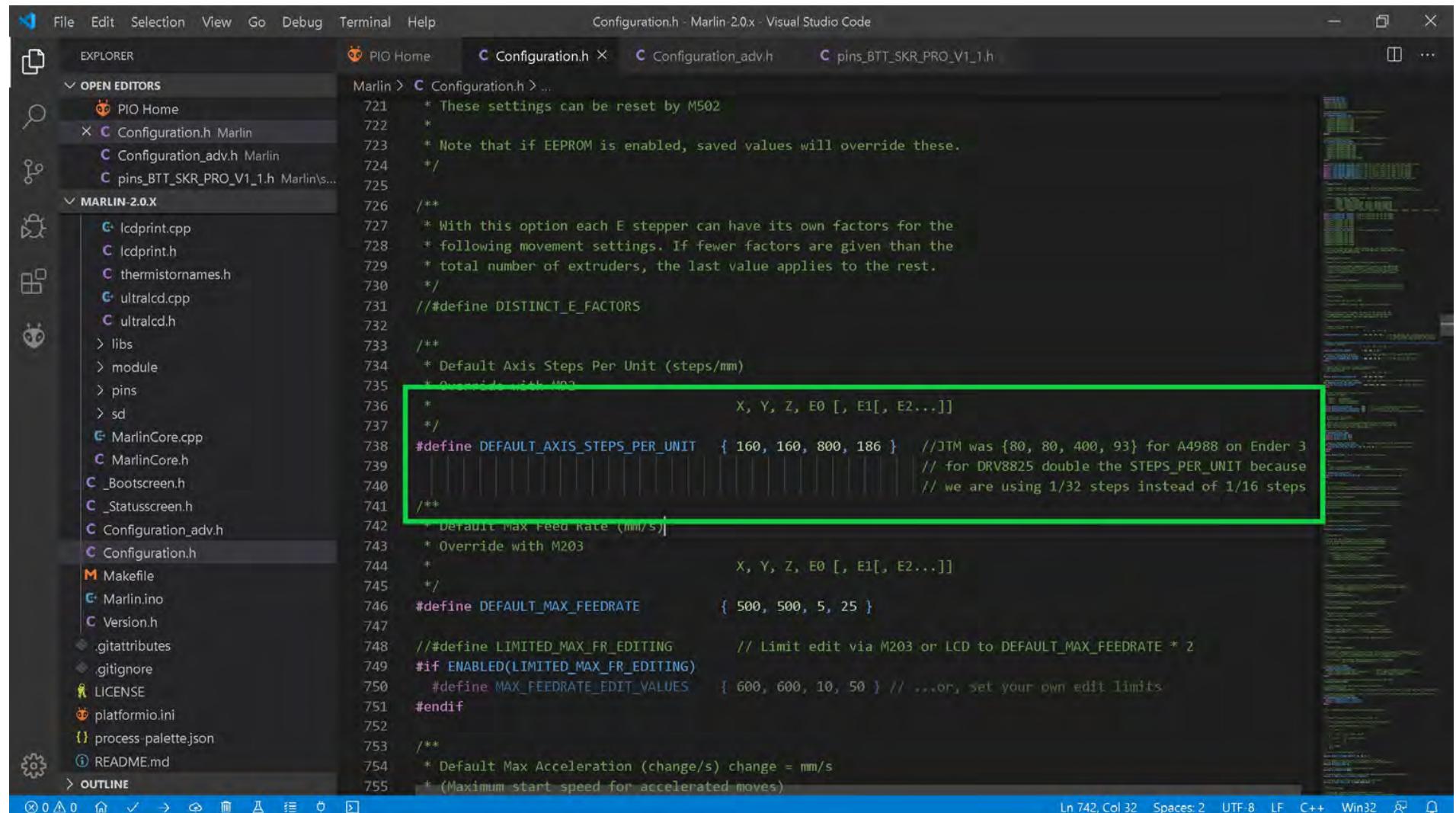
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * TMC2130_S
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130'
677 */
678 #define X_DRIVER_TYPE DRV8825 //JTM was commented out having a value of A4988
679 #define Y_DRIVER_TYPE DRV8825 //JTM was commented out having a value of A4988
680 #define Z_DRIVER_TYPE DRV8825 //JTM was commented out having a value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE DRV8825 //JTM was commented out having a value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

- We are changing from A4988 stepper motor drivers on the Ender 3 to DRV8825 stepper motor drivers for each axis and the extruder stepper motor driver. We will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```

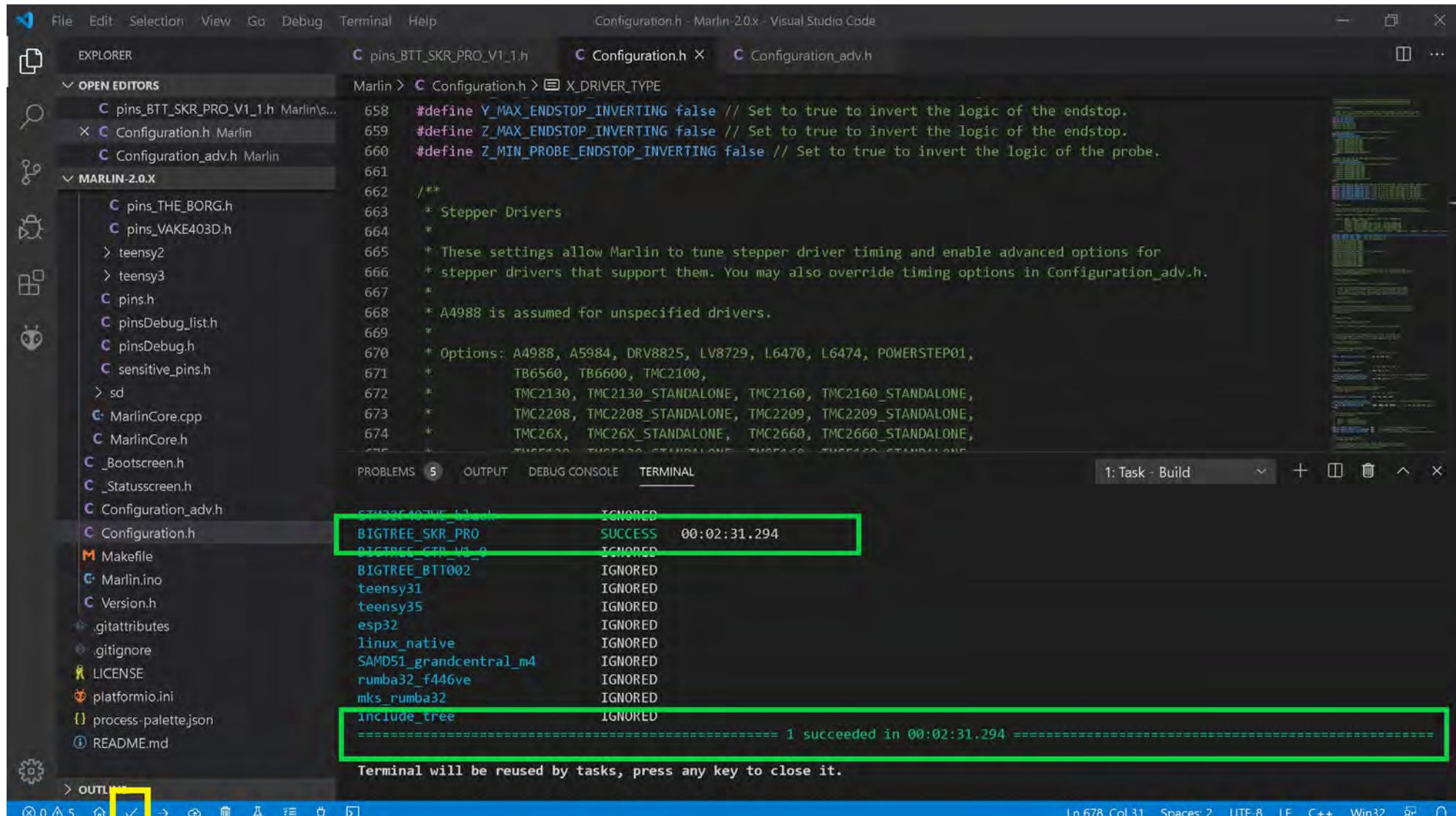
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h ...
PIO Home
Configuration.h Marlin
Configuration_adv.h Marlin
pins_BTT_SKR_PRO_V1_1.h Marlin...
MARLIN-2.0.X
Lcdprint.cpp
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
721 * These settings can be reset by M502
722 *
723 * Note that if EEPROM is enabled, saved values will override these.
724 */
725 /**
726 * With this option each E stepper can have its own factors for the
727 * following movement settings. If fewer factors are given than the
728 * total number of extruders, the last value applies to the rest.
729 */
730 // #define DISTINCT_E_FACTORS
731 /**
732 * Default Axis Steps Per Unit (steps/mm)
733 */
734 * X, Y, Z, E0 [, E1[, E2...]]
735 */
736 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // JTM was {80, 80, 400, 93} for A4988 on Ender 3
737 // for DRV8825 double the STEPS_PER_UNIT because
738 // we are using 1/32 steps instead of 1/16 steps
739 /**
740 * Default Max Feed Rate (mm/s)
741 */
742 * Override with M203
743 * X, Y, Z, E0 [, E1[, E2...]]
744 */
745 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
746 /**
747 * Limited Max Feed Rate Editing
748 */
749 #if ENABLED(LIMITED_MAX_FR_EDITING)
750 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
751 #endif
752 /**
753 * Default Max Acceleration (change/s) change = mm/s
754 */
755 * (Maximum start speed for accelerated moves)
Ln 742, Col 32 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

- The end of Marlin setup for DRV8825 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h

MARLIN-2.0.X pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins\pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h

```
Marlin > Configuration.h > X_DRIVER_TYPE
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**
663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build + ☐ ^ ×

STM32F407VC-DISCOVERY	IGNORED
BIGTREE_SKR_PRO	SUCCESS 00:02:31.294
BIGTREE_SKR_V3_0	IGNORED
BIGTREE_BTT002	IGNORED
teensy31	IGNORED
teensy35	IGNORED
esp32	IGNORED
linux_native	IGNORED
SAMD51_grandcentral_m4	IGNORED
rumba32_f446ve	IGNORED
mks_rumba32	IGNORED
include_tree	IGNORED

===== 1 succeeded in 00:02:31.294 =====

Terminal will be reused by tasks, press any key to close it.

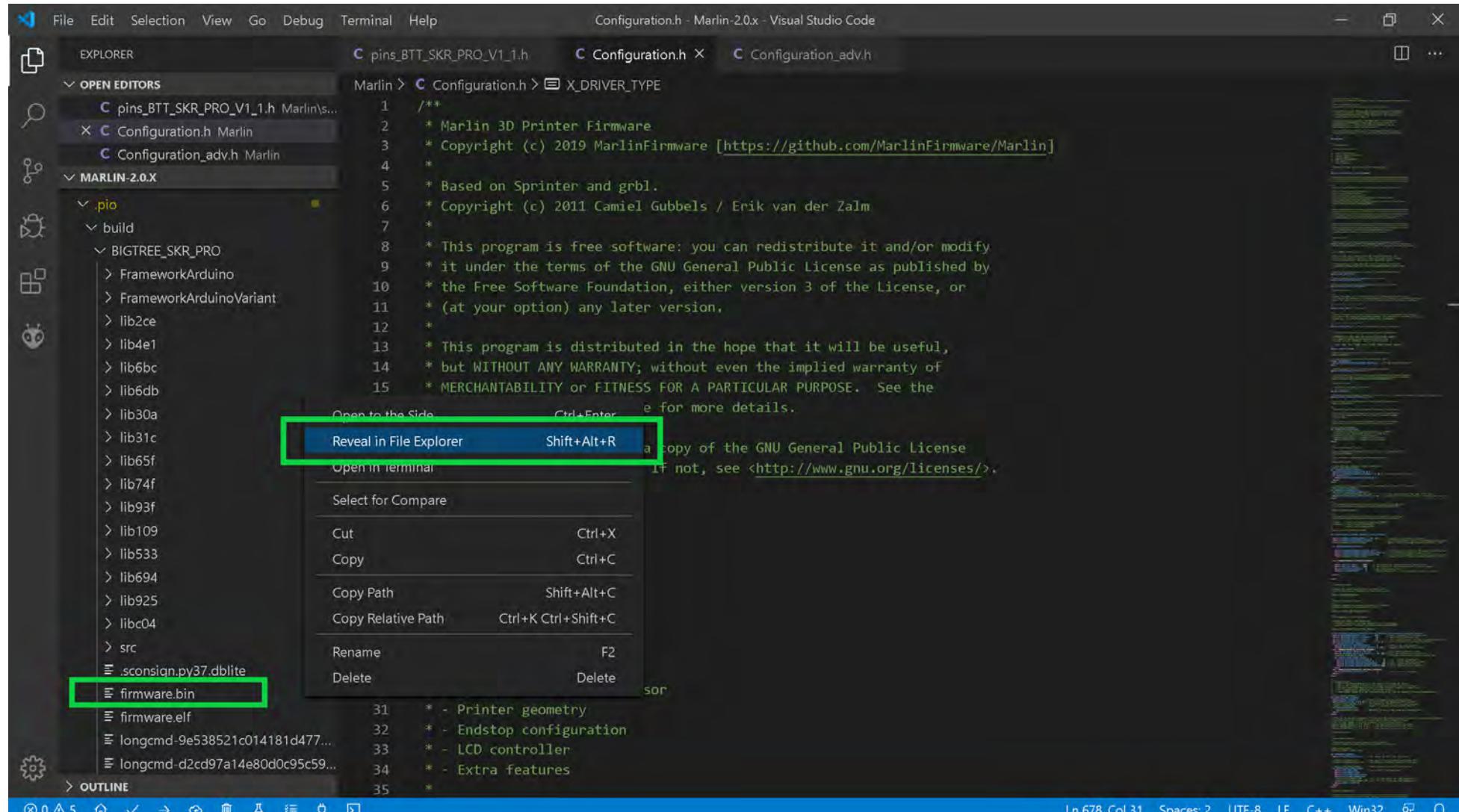
OUTLINE 

Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32 ⚙️

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

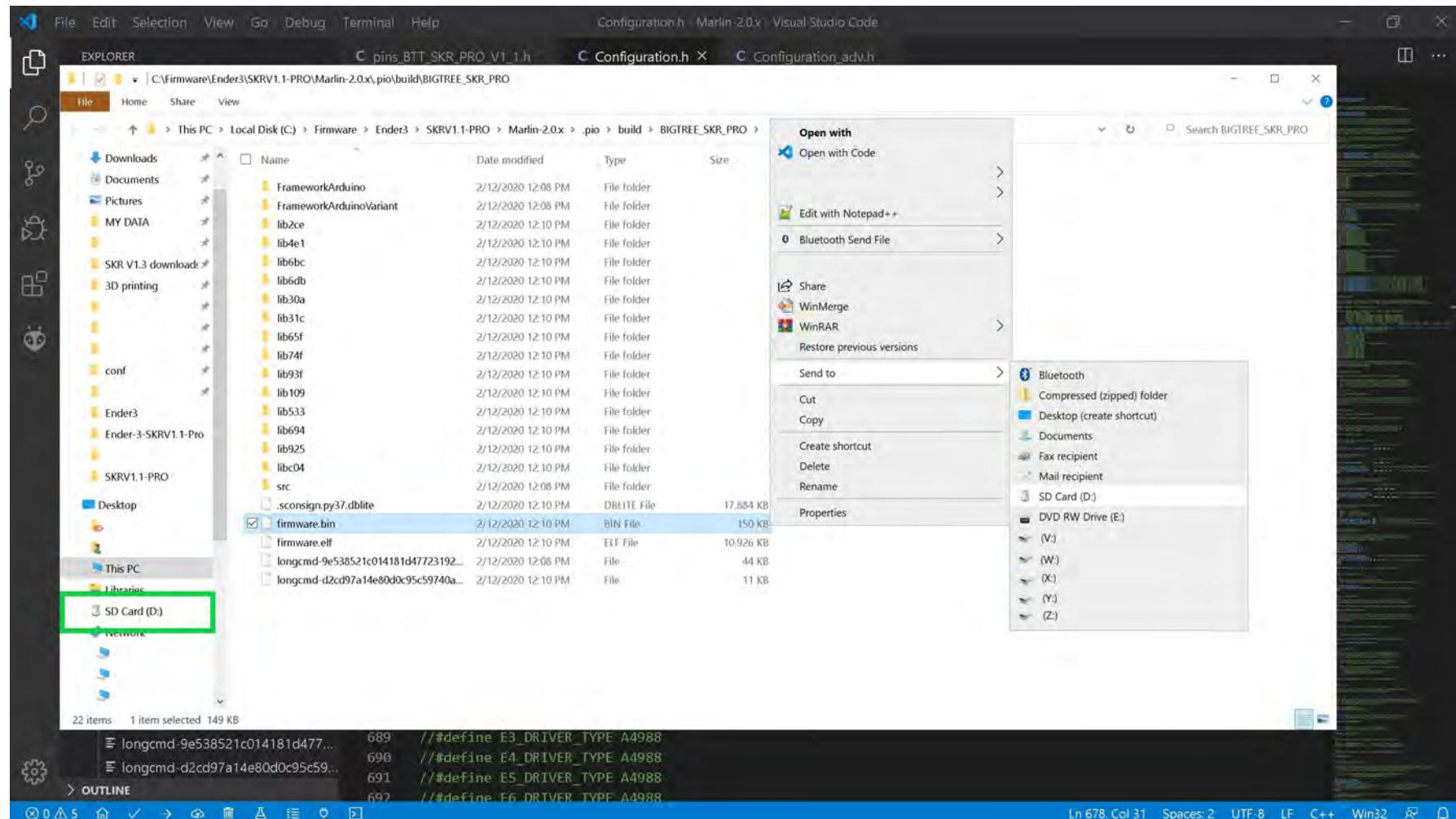
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



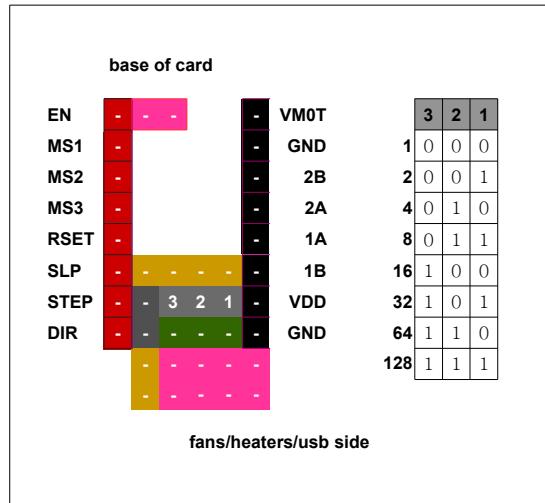
- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

- Ensure your micro-SD card, that came with your SKR PRO V1.1 board, is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

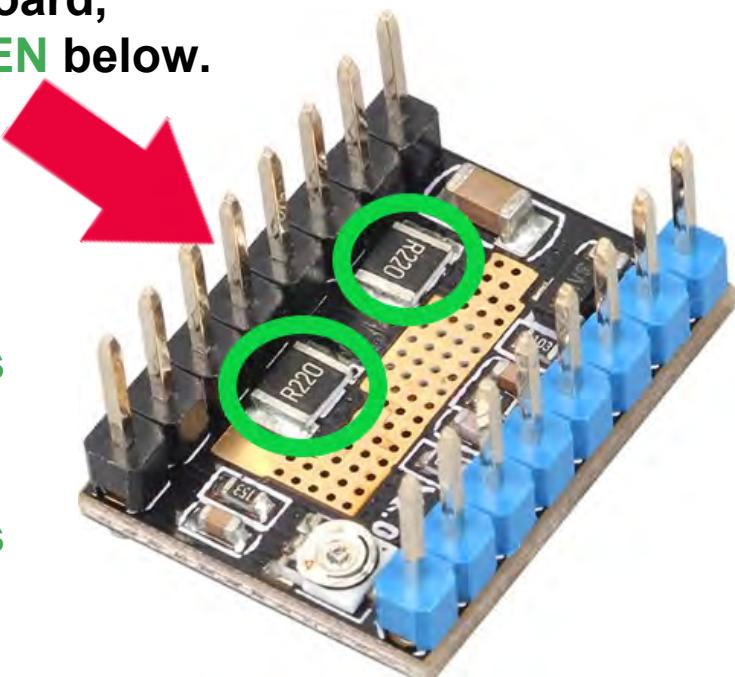
BIQU LV8729

Driver Chip	MD3	MD2	MD1	Subdivision	Excitation Mode
LV8729 Maximum 128 subdivision 36V 1.5A	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	2W1-2 Phase
	H	L	L	1/16	4W1-2 Phase
	H	L	H	1/32	8W1-2 Phase
	H	H	L	1/64	16W1-2 Phase
	H	H	H	1/128	32W1-2 Phase
Driving current calculation formula		$I_{MAX} = V_{REF} \cdot 2$			$V_{REF} = (I_{MAX}) / 2$
$R_s = 0.1\Omega$		$I_{MAX} = \frac{V_{REF}}{1.1}$			$V_{REF} = I_{MAX} * 1.1$

Note: Use 90% of the calculated Vref when tuning the stepper driver board

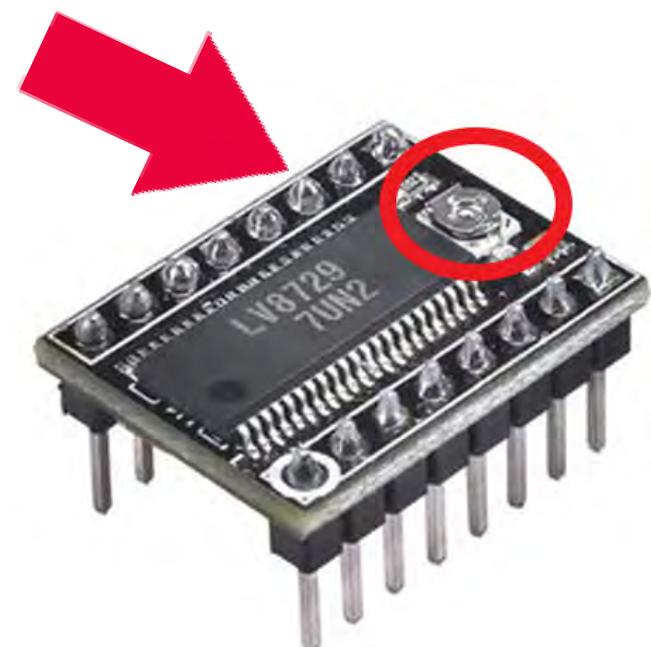
Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.



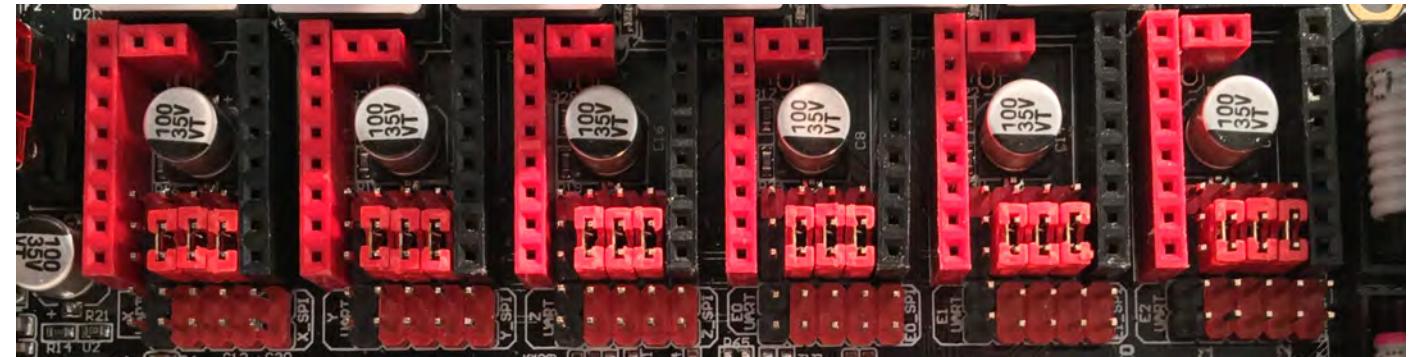
$Rs = R220$ is 0.22 Ohms

$Rs = R100$ is 0.1 Ohms

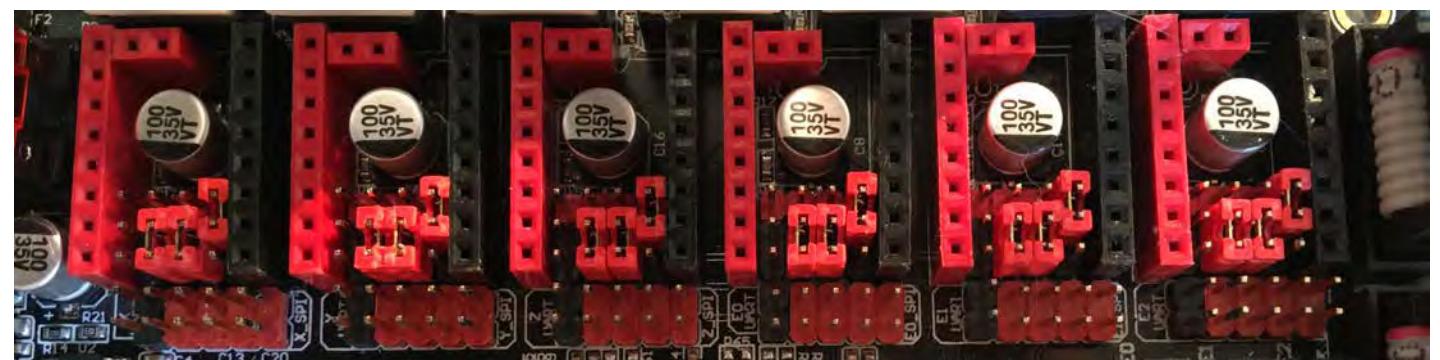


BIQU LV8729

EN	-	-	-	VMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	-	-	2A	
RSET	-	-	-	1A	
SLP	-	-	-	1B	
STEP	-	3	2	1	VDD
DIR	-	3	2	1	GND
				STEP	



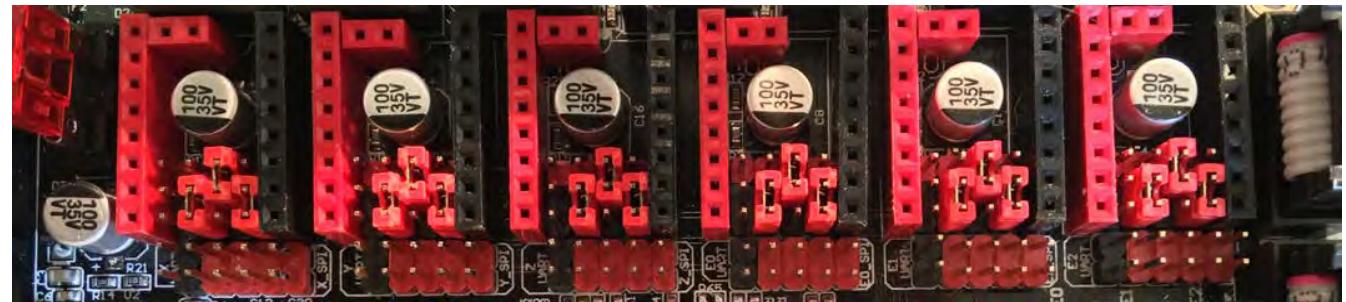
EN	-	-	-	VMOT		
MS1	-	-	-	GND		
MS2	-	-	-	2B		
MS3	-	-	-	2A		
RSET	-	-	-	1A		
SLP	-	-	1	-	1B	
STEP	-	3	2	1	-	VDD
DIR	-	2	2	-	-	GND
					1 / 2	



BIQU LV8729

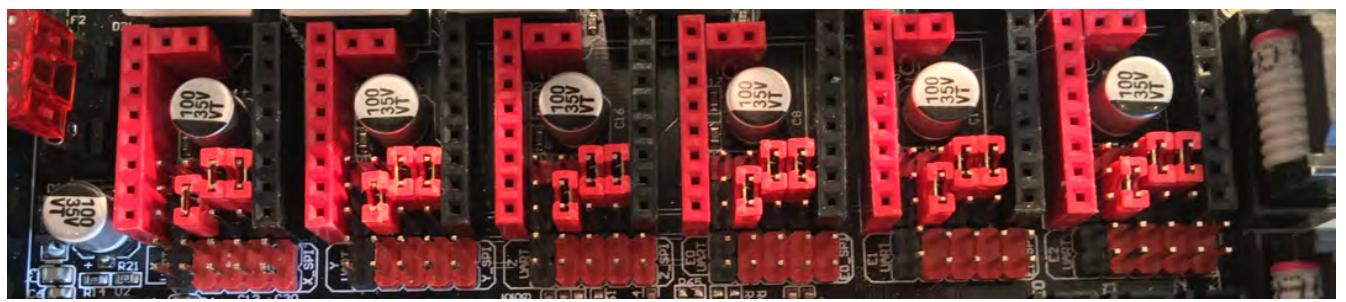
EN	-	-	-	-	VMOT
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RSET	-	-	-	-	1A
SLP	-	-	2	-	1B
STEP	-	3	2	1	-
DIR	-	-	3	1	-

1 / 4



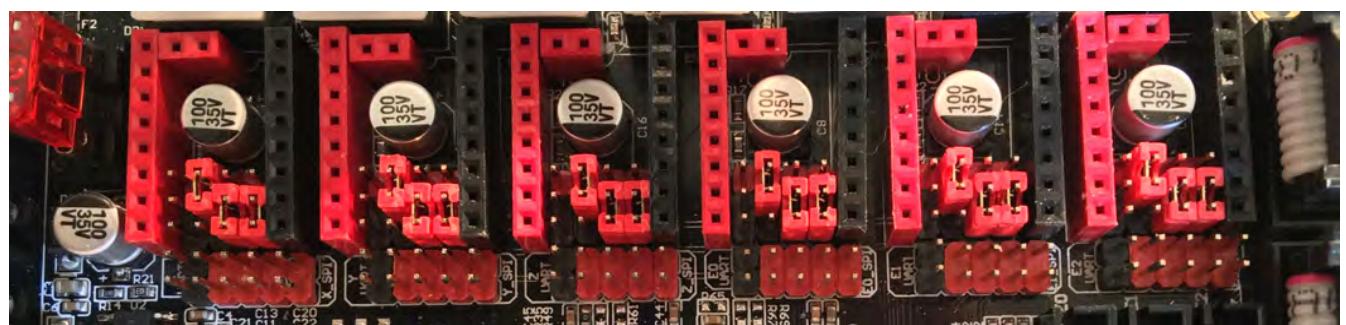
EN	-	-	-	-	VMOT
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RSET	-	-	-	-	1A
SLP	-	-	2	2	1B
STEP	-	3	2	1	-
DIR	-	-	3	-	-

1 / 8



EN	-	-	-	-	VMOT
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RSET	-	-	-	-	1A
SLP	-	3	-	-	1B
STEP	-	3	2	1	-
DIR	-	-	2	1	-

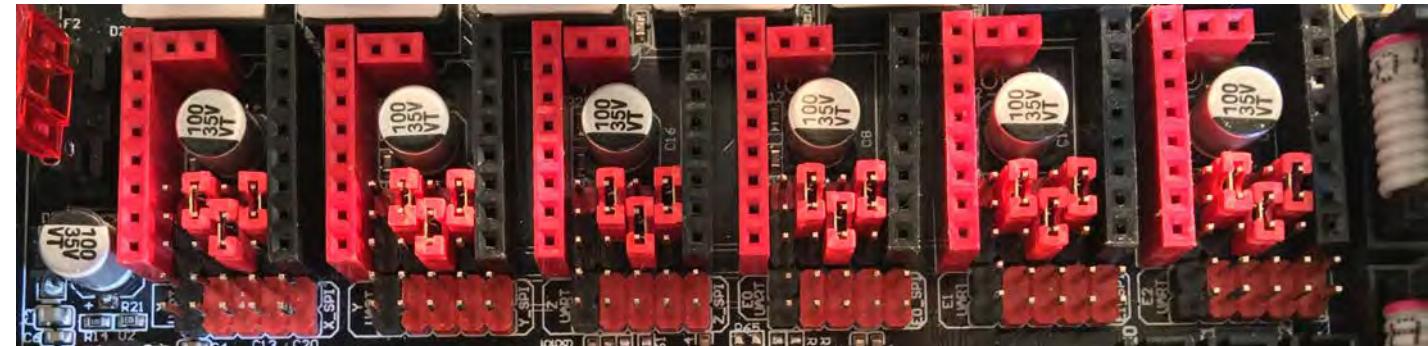
1 / 16



BIQU LV8729

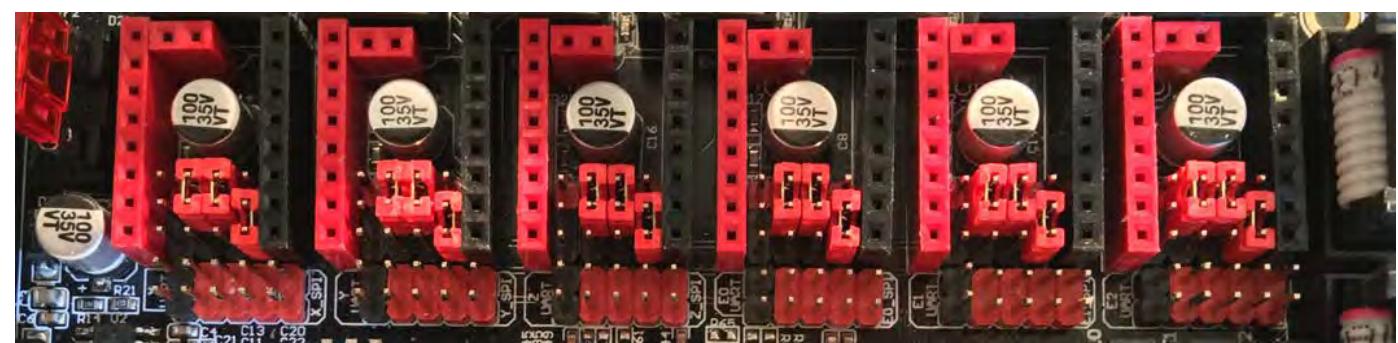
EN	-	-	-	-	VMOT	-
MS1	-	-	-	-	GND	-
MS2	-	-	-	-	2B	-
MS3	-	-	-	-	2A	-
RSET	-	-	-	-	1A	-
SLP	-	3	1	-	1B	-
STEP	-	3	2	1	VDD	-
	-	-	2	-	GND	-

1 / 32



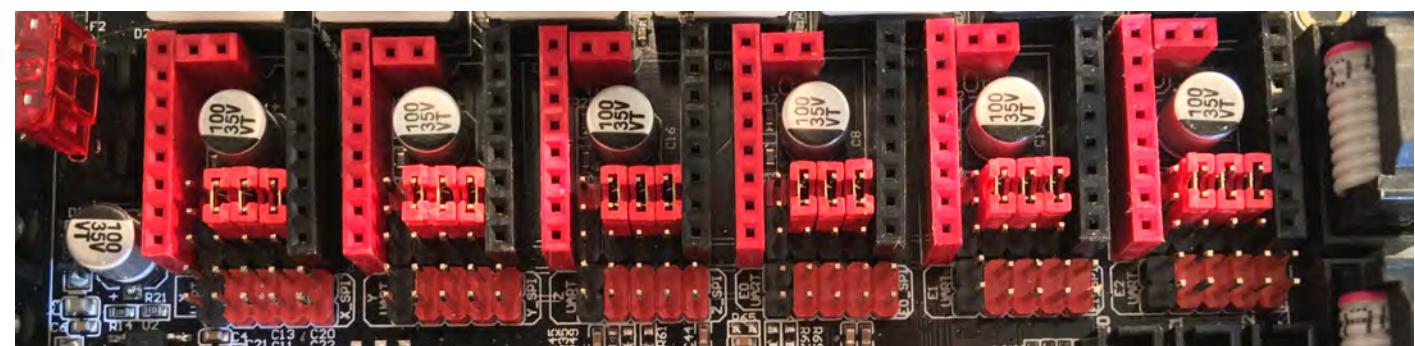
EN	-	-	-	-	VMOT	-
MS1	-	-	-	-	GND	-
MS2	-	-	-	-	2B	-
MS3	-	-	-	-	2A	-
RSET	-	-	-	-	1A	-
SLP	-	3	2	-	1B	-
STEP	-	3	2	1	VDD	-
DIR	-	-	1	-	GND	-

1 / 64



EN	-	-	-	-	VMOT	-
MS1	-	-	-	-	GND	-
MS2	-	-	-	-	2B	-
MS3	-	-	-	-	2A	-
RSET	-	-	-	-	1A	-
SLP	-	3	2	1	1B	-
STEP	-	3	2	1	VDD	-
DIR	-	-	1	-	GND	-

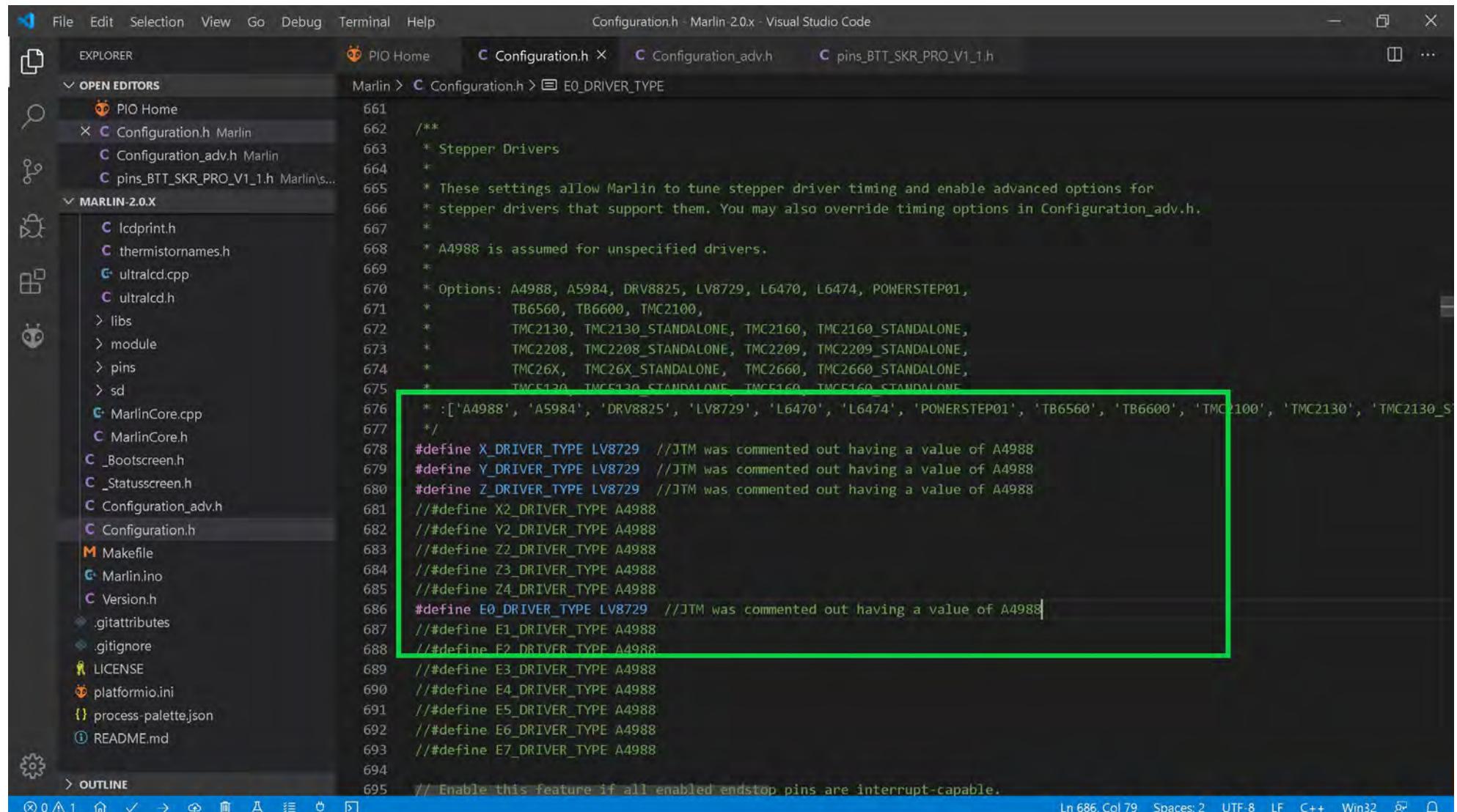
1 / 128



The (latest release of) Marlin Setup for BIQU LV8729 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU LV8729 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using BIQU LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU LV8729 drivers. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 */
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160',
677 */
678 #define X_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
679 #define Y_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
680 #define Z_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

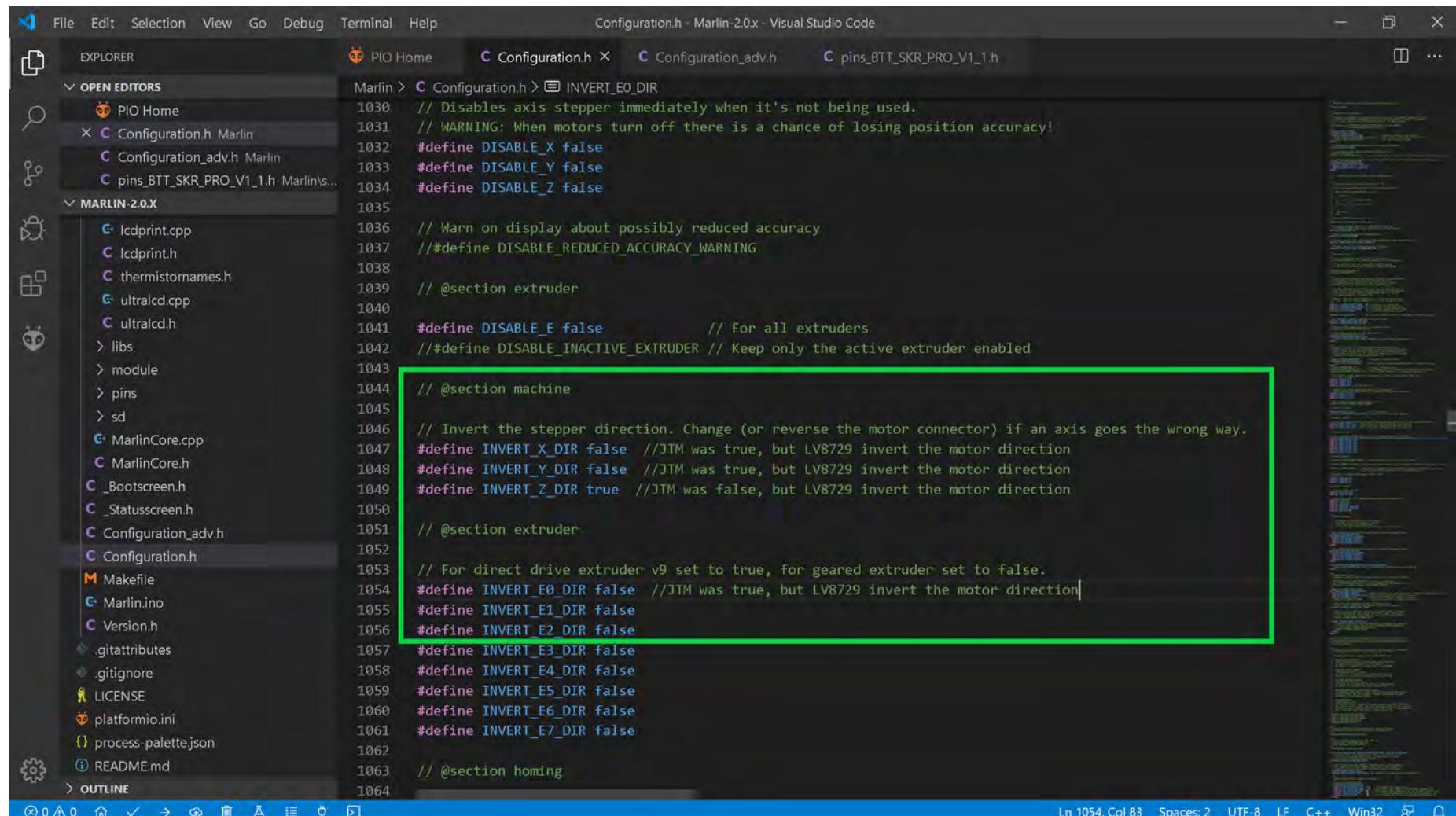
```

Ln 686, Col 79 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU LV8729 Drivers

- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as show in the GREEN box below



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a dark theme. On the left, the 'EXPLORER' sidebar shows various Marlin source files and configuration files. The main code area displays the following C++ code:

```

Marlin > Configuration.h > INVERT_E0_DIR
1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 // #define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false           // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false      // JTM was true, but LV8729 invert the motor direction
1048 #define INVERT_Y_DIR false      // JTM was true, but LV8729 invert the motor direction
1049 #define INVERT_Z_DIR true       // JTM was false, but LV8729 invert the motor direction
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false     // JTM was true, but LV8729 invert the motor direction
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

```

A green rectangular box highlights the section of code from line 1046 to line 1061, which defines the INVERT_Ex_DIR macros for axes E0 through E7. The code indicates that for the LV8729 driver, the values should be swapped compared to the JTM driver.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU LV8729 Drivers

- The end of Marlin setup for BIQU LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `pins_BTT_SKR_PRO_V1_1.h`, `Configuration.h`, and `Configuration_adv.h`.
- Terminal:** Displays the command-line output of the compilation process.
- Bottom Toolbar:** Contains various icons, including a yellow box highlighting the "Build" icon (a checkmark).
- Terminal Output:**

```
Configuration.h - Marlin-2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins\pins_BTT_SKR_PRO_V1_1.h
Configuration.h Marlin Configuration.h X
Configuration_adv.h Marlin Configuration_adv.h X

MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + □ ×
STM32F407VC-DJ.vcxproj: IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_0 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
1 succeeded in 00:02:31.294

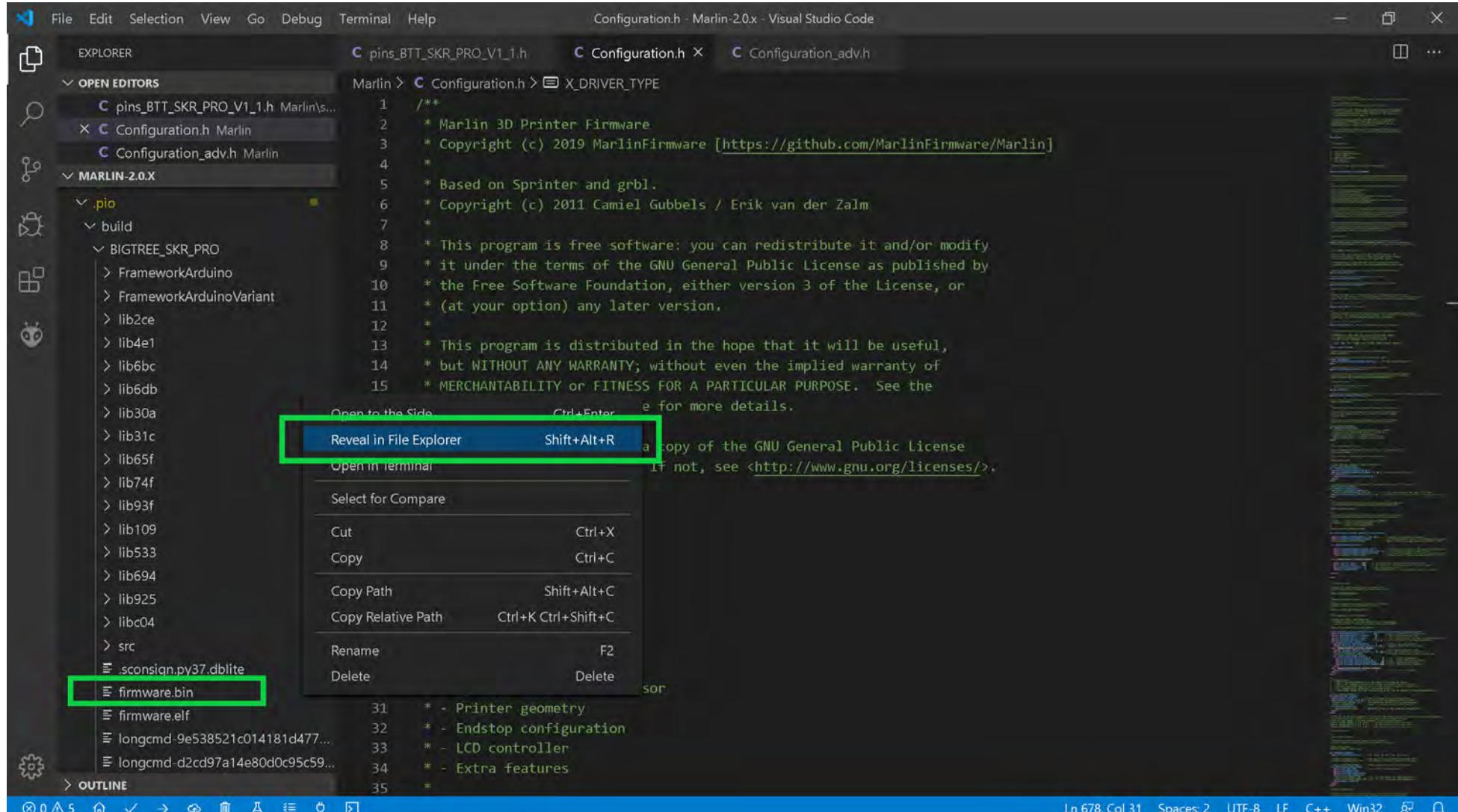
Terminal will be reused by tasks, press any key to close it.

Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32 ⚙️
```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU LV8729 Drivers

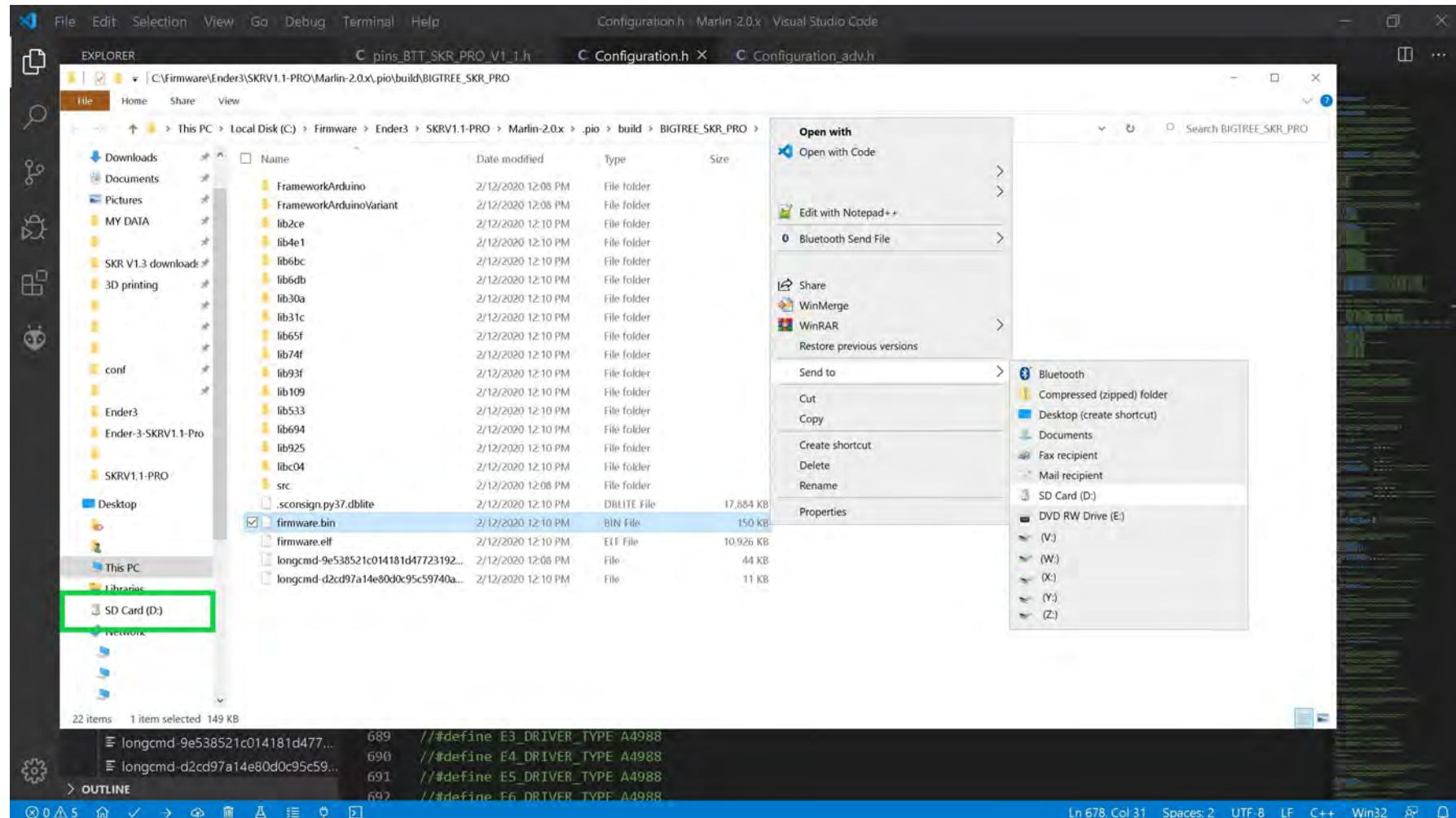
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file explorer window.



- Go to the next page.

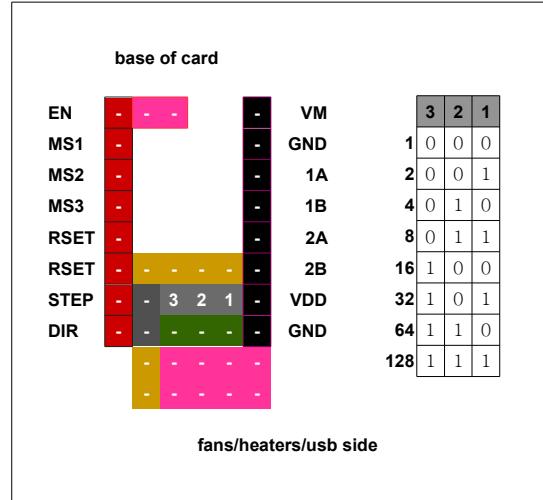
The (latest release of) Marlin Setup for BIQU LV8729 Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



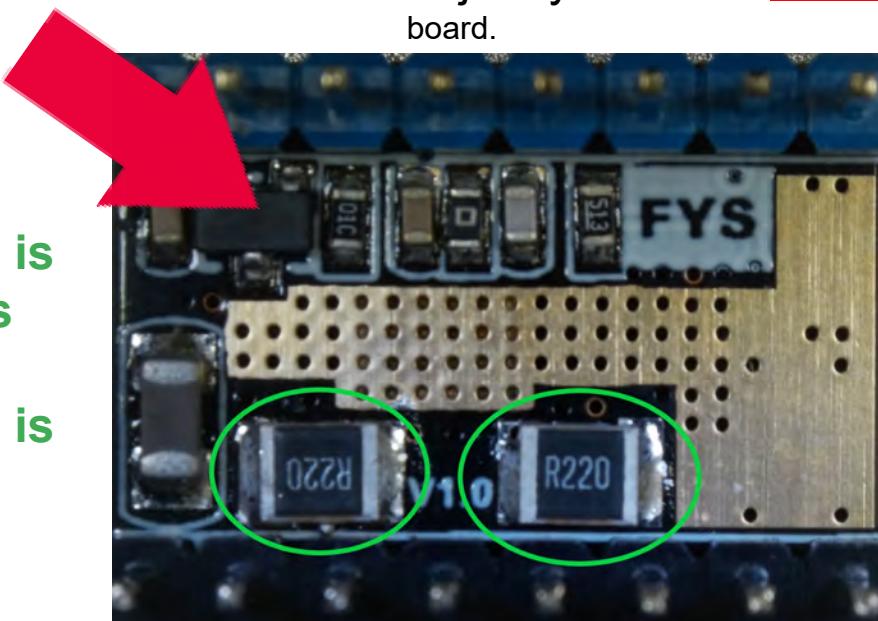
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

FYSETC LV8729



Driver Chip	MD3	MD2	MD1	Subdivision	Excitation Mode
LV8729 Maximum 128 subdivision 36V 1.5A	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	2W1-2 Phase
	H	L	L	1/16	4W1-2 Phase
	H	L	H	1/32	8W1-2 Phase
	H	H	L	1/64	16W1-2 Phase
	H	H	H	1/128	32W1-2 Phase
Driving current calculation formula	$I_{MAX} = V_{REF} \cdot 2$	$V_{REF} = (I_{MAX}) / 2$			
$Rs = 0.1\Omega$	$I_{MAX} = \frac{V_{REF}}{1.1}$	$V_{REF} = I_{MAX} * 1.1$			

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

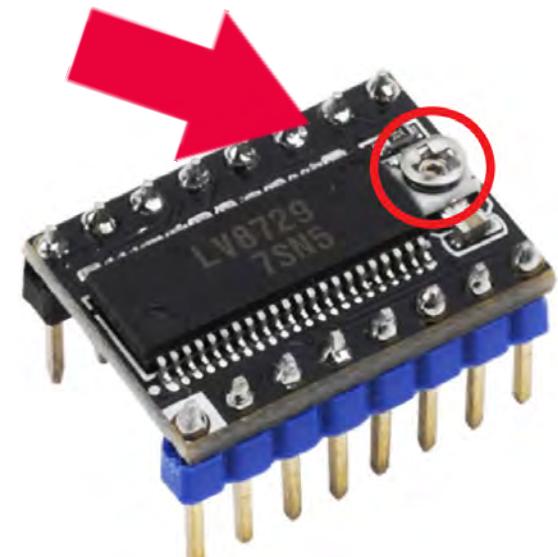


$Rs = R220$ is 0.22 Ohms

$Rs = R100$ is 0.1 Ohms

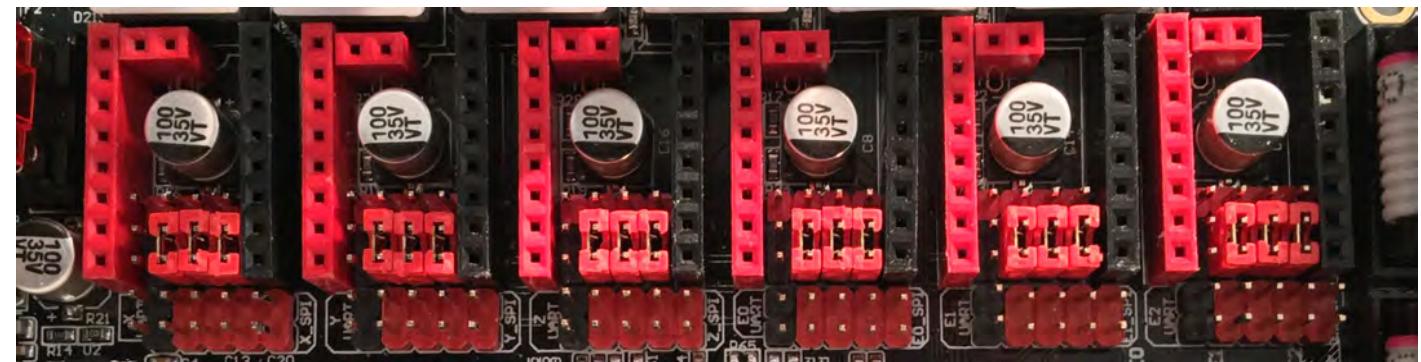
Note: Use 90% of the calculated Vref when tuning the stepper driver board

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

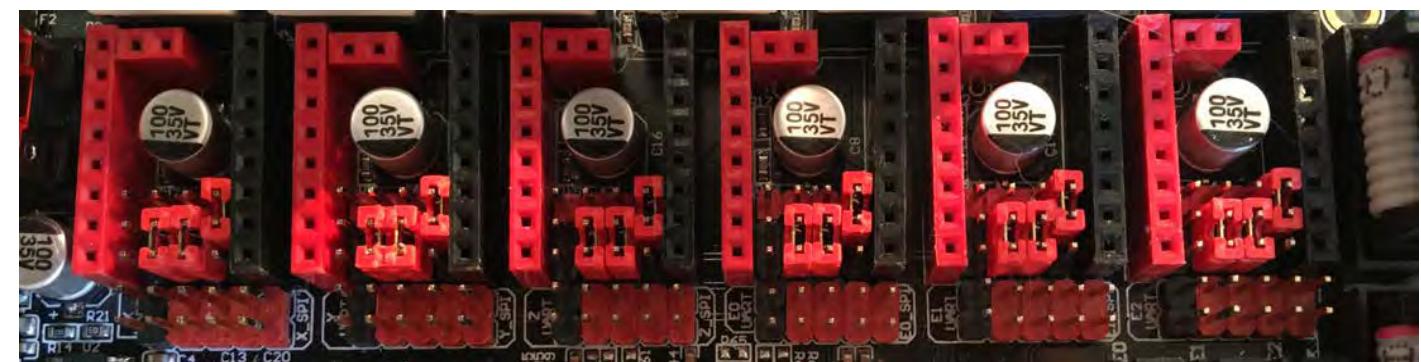


FYSETC LV8729

EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1A
MS3	-	-	-	-	1B
RSET	-	-	-	-	2A
RSET	-	-	-	-	2B
STEP	-	-	3 2 1	-	VDD
DIR	-	-	3 2 1	-	GND
STEP					



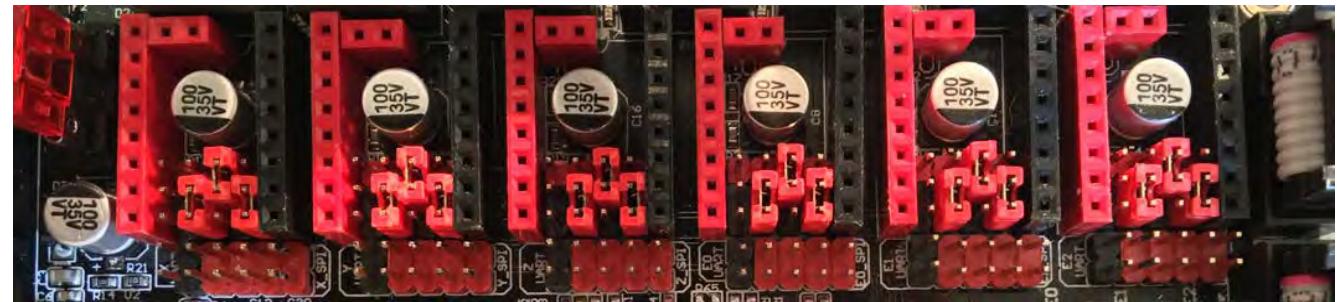
EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1A
MS3	-	-	-	-	1B
RSET	-	-	-	-	2A
RSET	-	-	-	1	2B
STEP	-	-	3 2 1	-	VDD
DIR	-	-	3 2	-	GND
1 / 2					



FYSETC LV8729

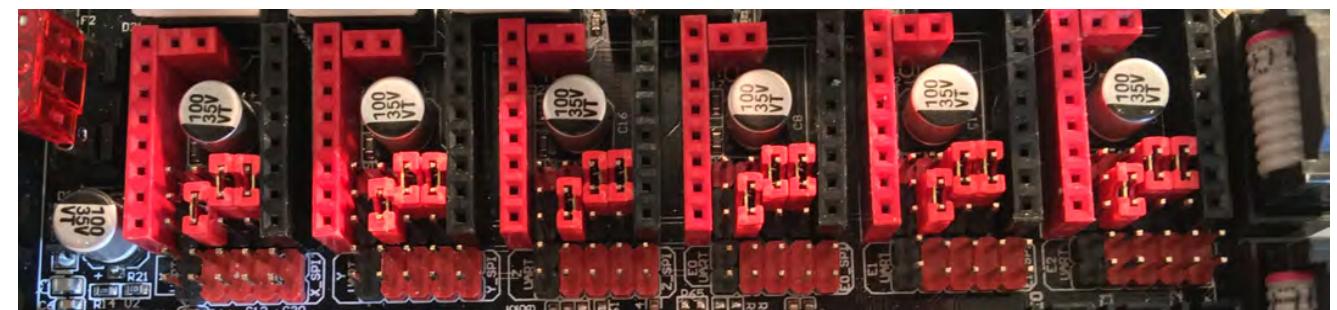
EN	-	-	-	-
MS1	-	-	-	-
MS2	-	-	-	-
MS3	-	-	-	-
RSET	-	-	-	-
RSET	-	-	2	-
STEP	-	3	2	1
DIR	-	3	2	1

1 / 4



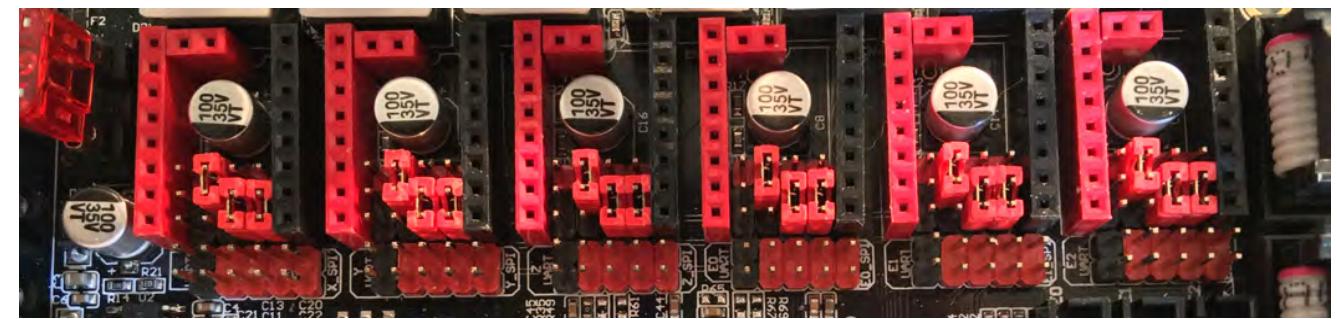
EN	-	-	-	-
MS1	-	-	-	-
MS2	-	-	-	-
MS3	-	-	-	-
RSET	-	-	-	-
RSET	-	2	1	-
STEP	-	3	2	1
DIR	-	3	2	1

1 / 8

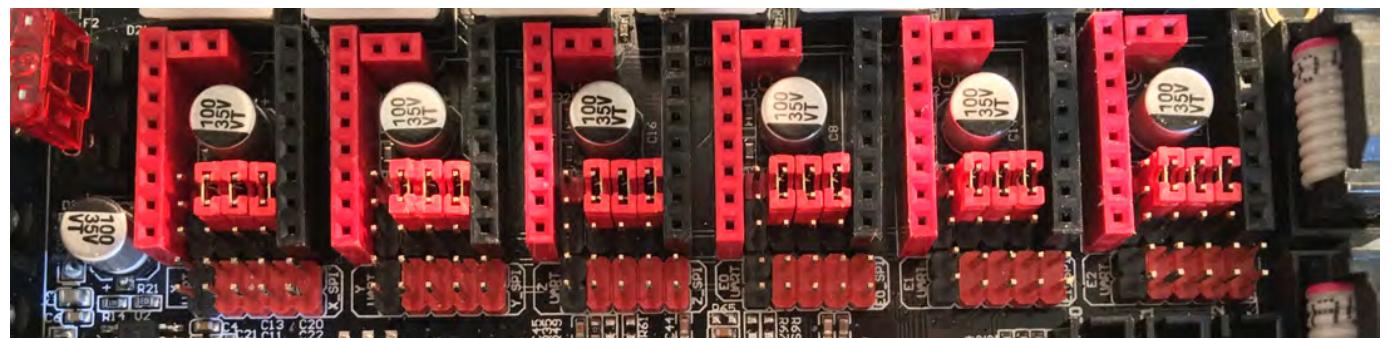
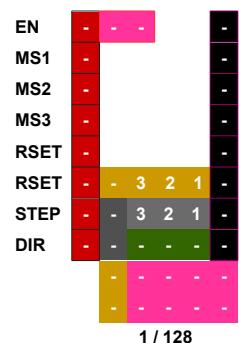
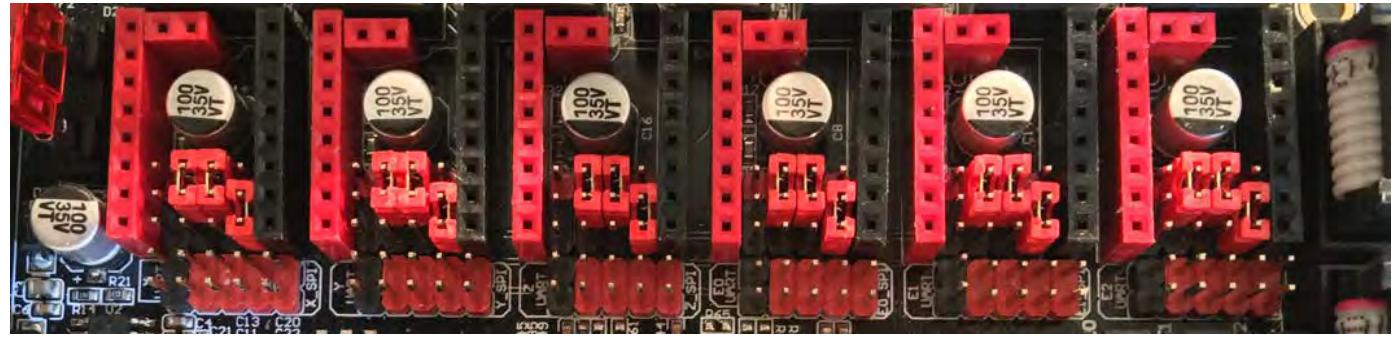
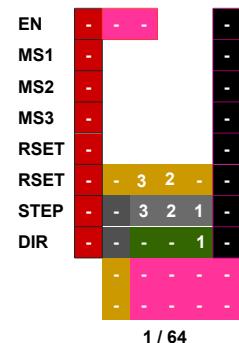
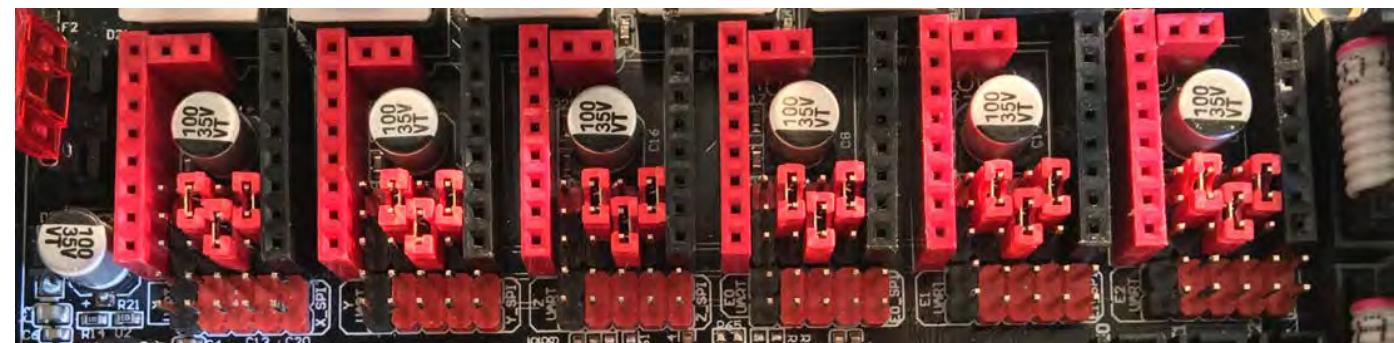
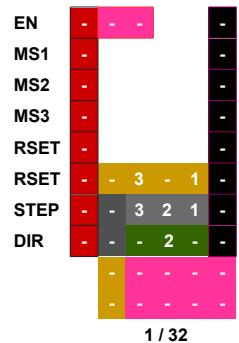


EN	-	-	-	-
MS1	-	-	-	-
MS2	-	-	-	-
MS3	-	-	-	-
RSET	-	-	-	-
RSET	-	3	-	-
STEP	-	3	2	1
DIR	-	3	2	1

1 / 16



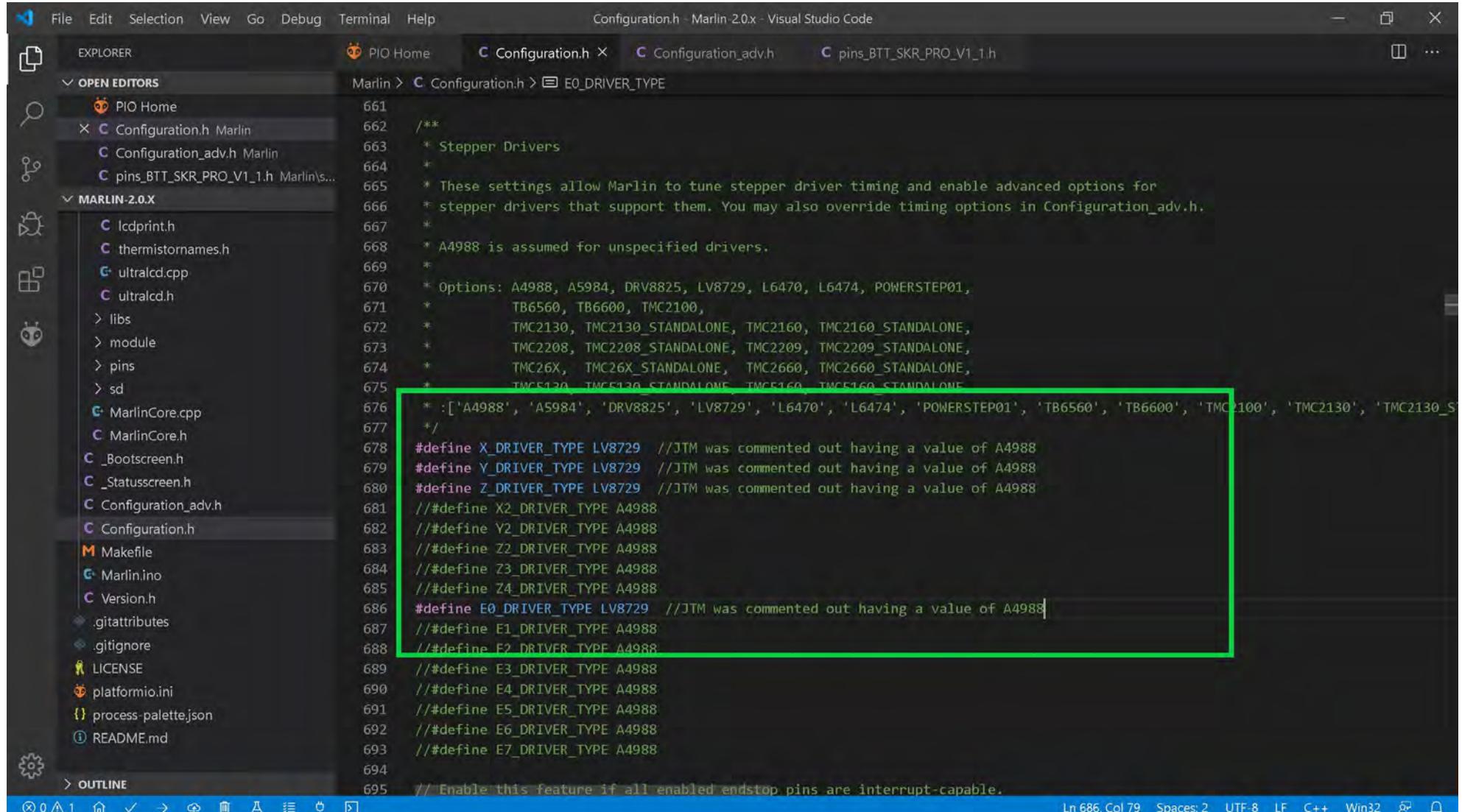
FYSETC LV8729



The (latest release of) Marlin Setup for FYSETC LV8729 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for FYSETC LV8729 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using FYSETC LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC LV8729 drivers. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

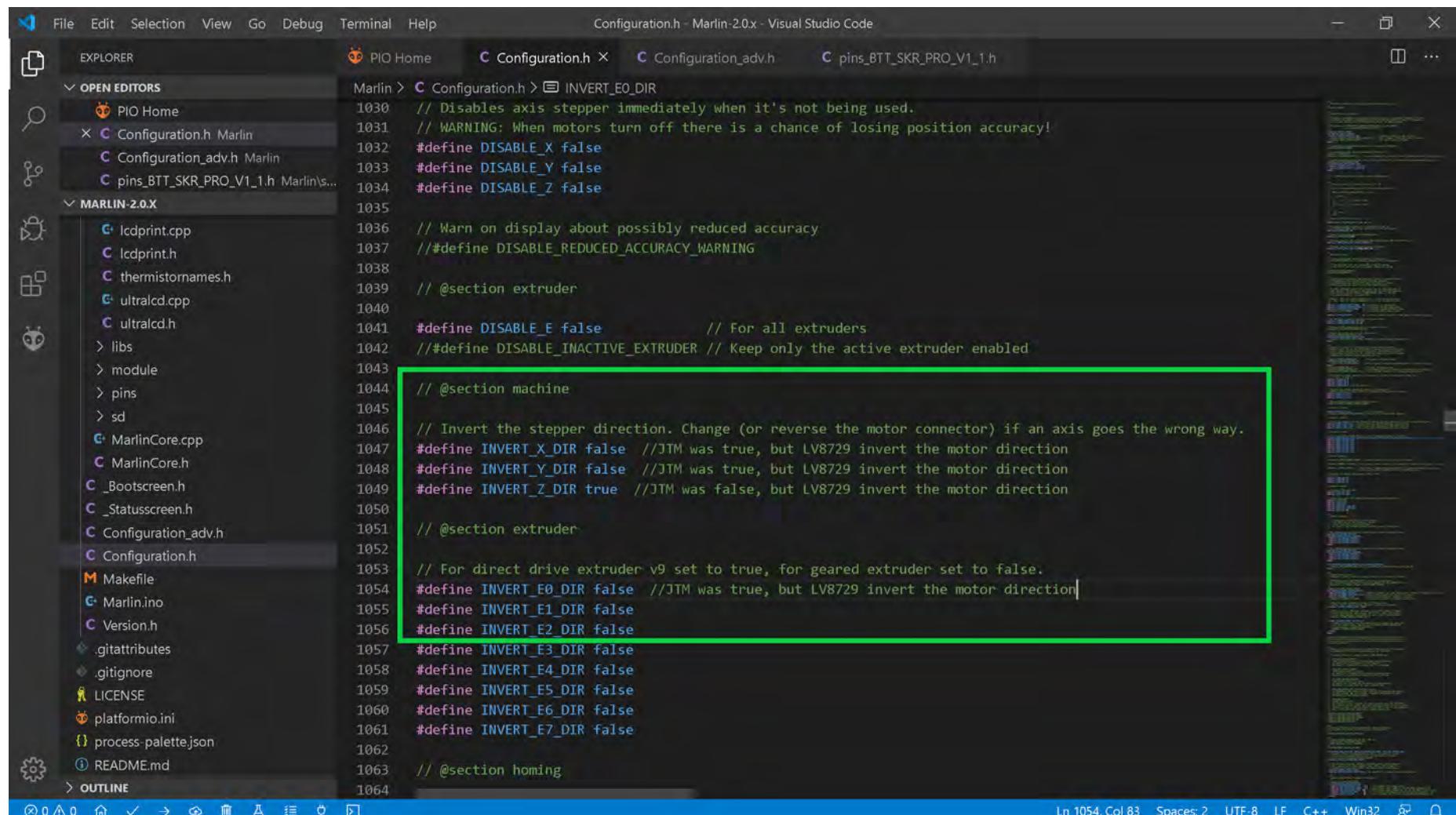
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home
Configuration.h Marlin
Configuration_adv.h Marlin
pins_BTT_SKR_PRO_V1_1.h Marlin\...
MARLIN-2.0.X
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686 Col 79 Spaces: 2 UTF-8 LF C++ Win32
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_ST
676 */
#define X_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
#define Y_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
#define Z_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define Z3_DRIVER_TYPE A4988
#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
// Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC LV8729 Drivers

- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as show in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 // #define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false          // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false      // JTM was true, but LV8729 invert the motor direction
1048 #define INVERT_Y_DIR false      // JTM was true, but LV8729 invert the motor direction
1049 #define INVERT_Z_DIR true       // JTM was false, but LV8729 invert the motor direction
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false     // JTM was true, but LV8729 invert the motor direction
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

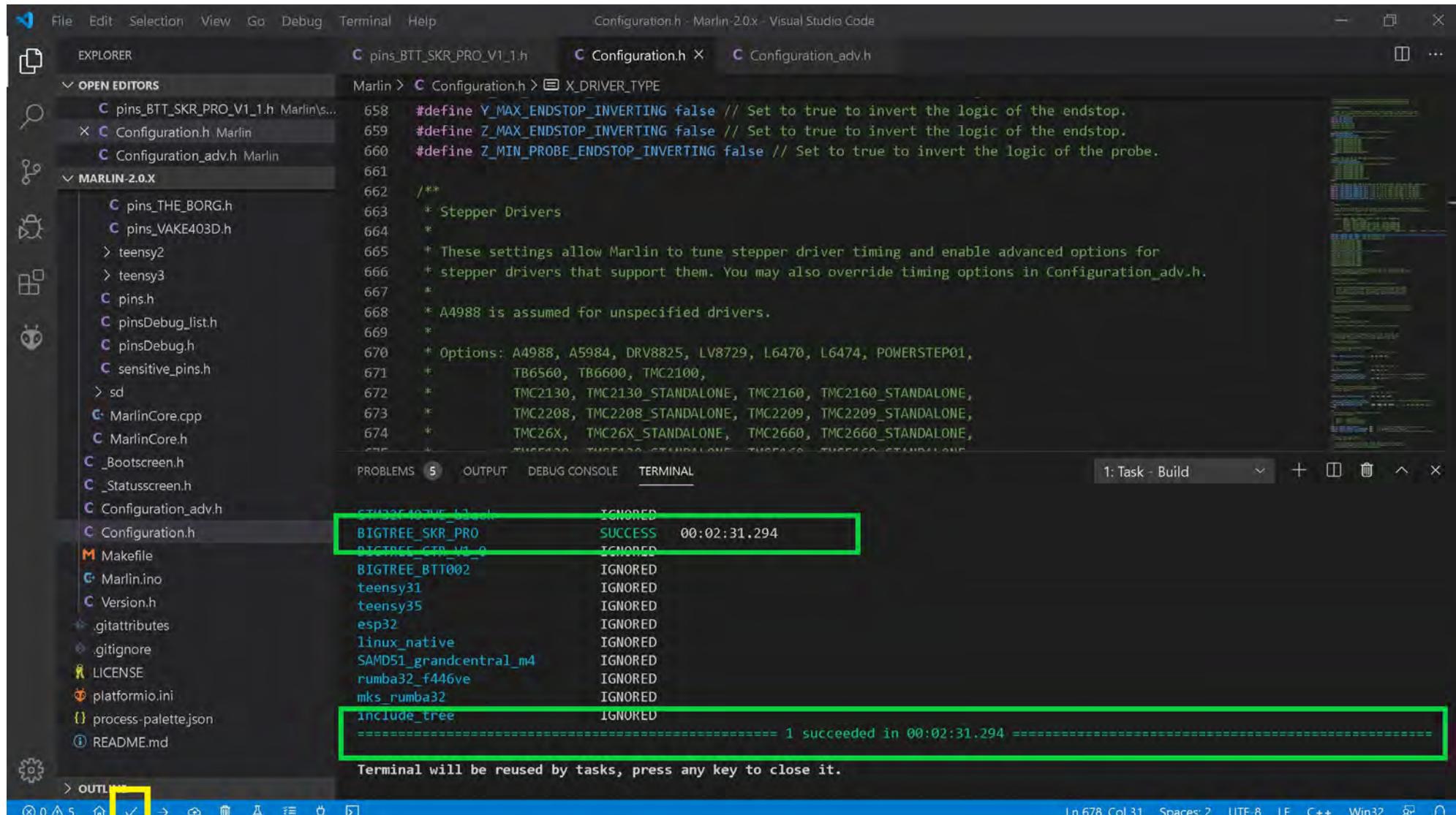
```

Ln 1054, Col 83 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC LV8729 Drivers

- The end of Marlin setup for FYSETC LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

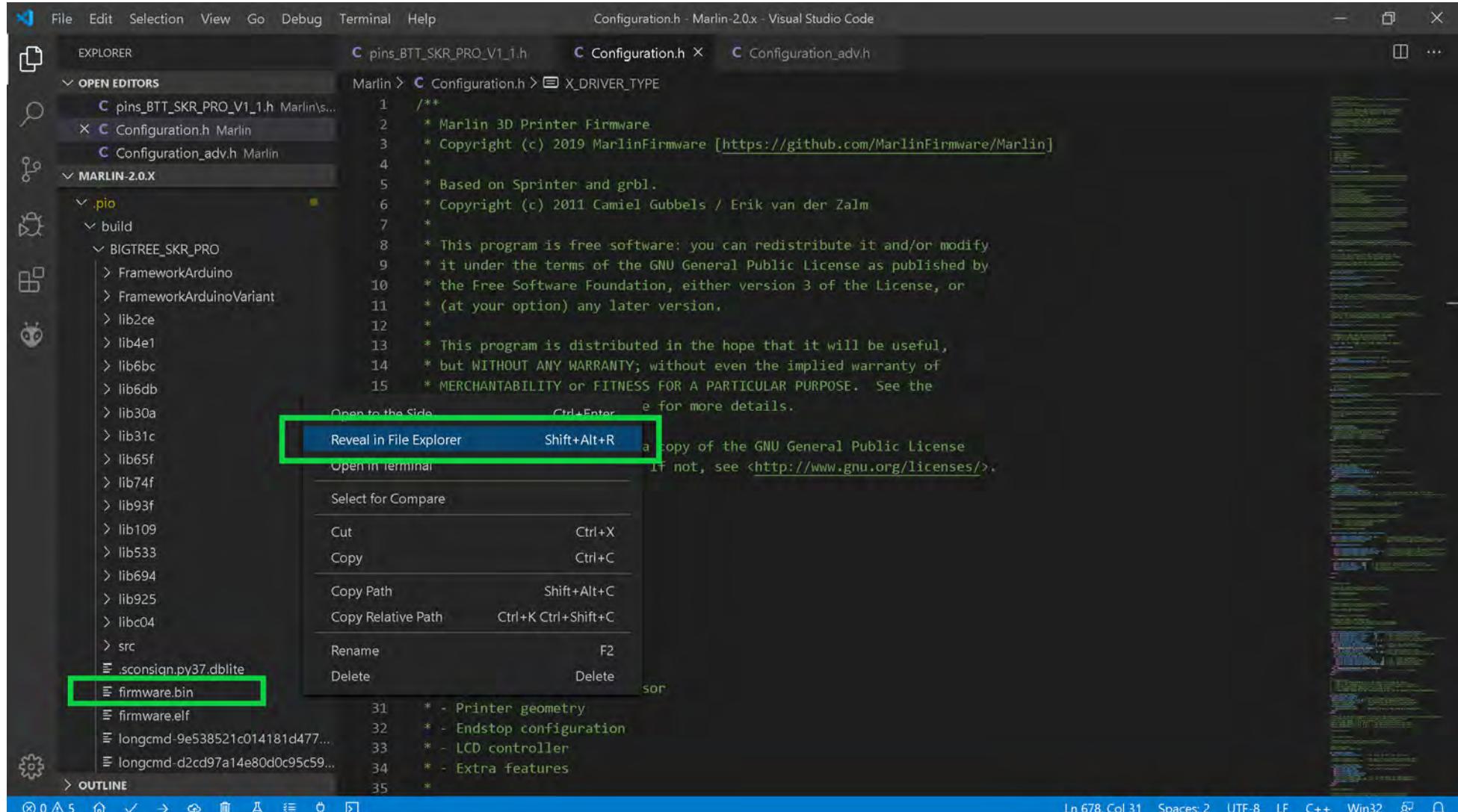
File Edit Selection View Go Debug Terminal Help
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins\pins_BTT_SKR_PRO_V1_1.h
Configuration.h Marlin Configuration.h X
Configuration_adv.h Marlin Configuration_adv.h X
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
STM32F407VC-D3.cu IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V3_0 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
1 succeeded in 00:02:31.294
Terminal will be reused by tasks, press any key to close it.
Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32 ⚙

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for FYSETC LV8729 Drivers

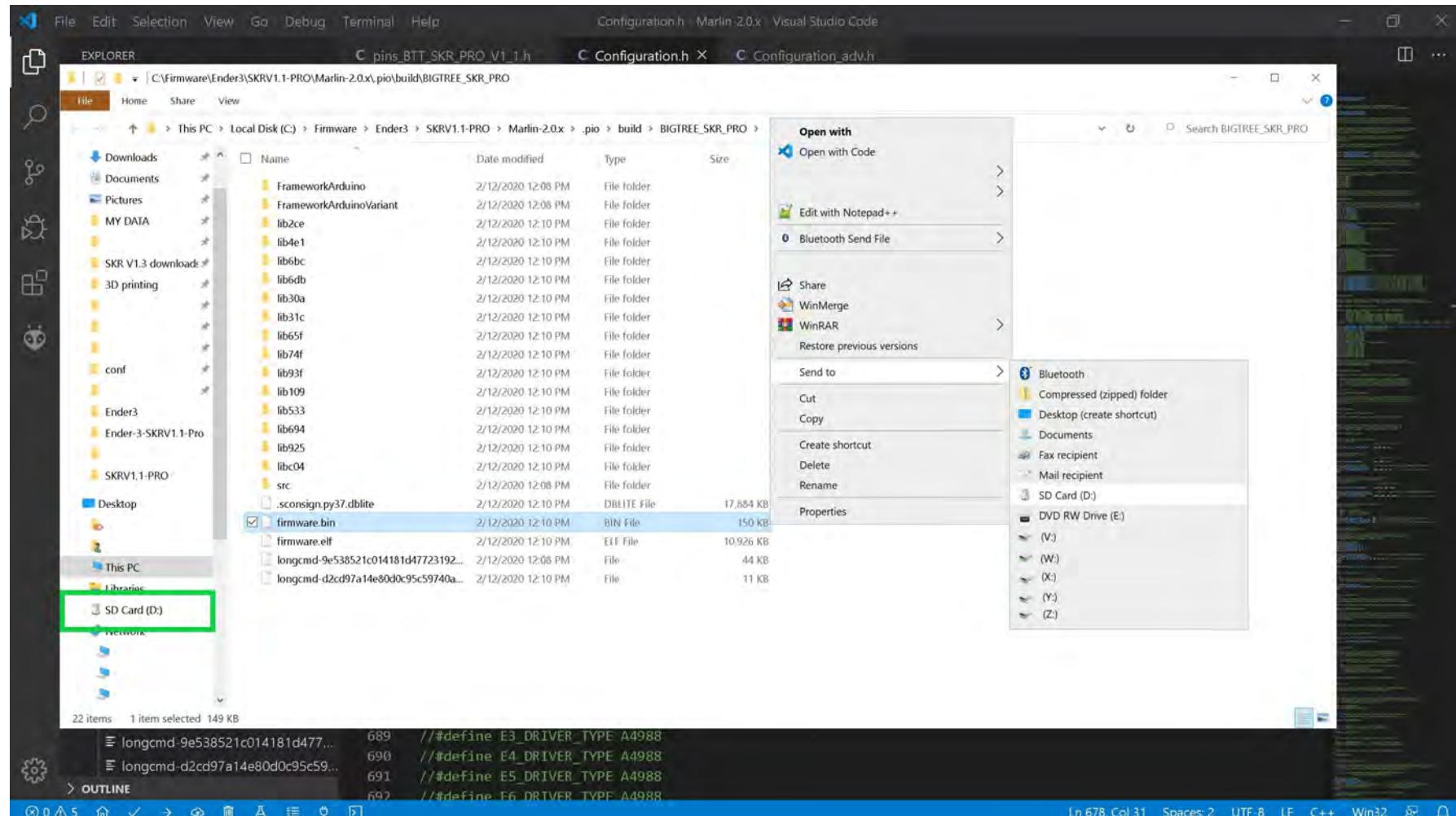
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



- Go to the next page.

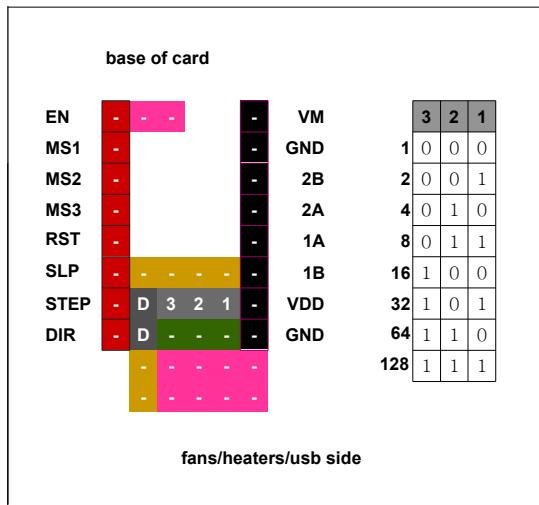
The (latest release of) Marlin Setup for FYSETC LV8729 Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

Lerdge LV8729

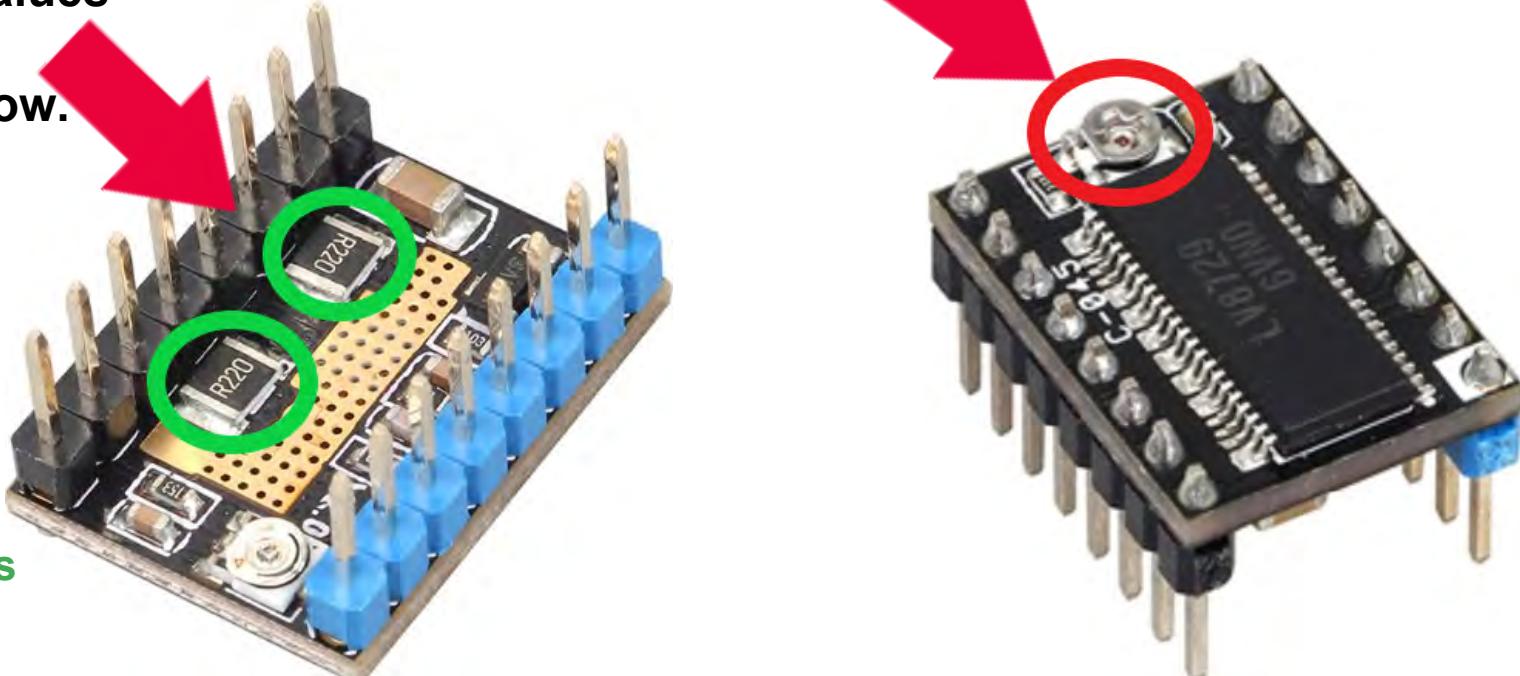


Driver Chip	MD3	MD2	MD1	Subdivision	Excitation Mode
LV8729 Maximum 128 subdivision 36V 1.5A	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	2W1-2 Phase
	H	L	L	1/16	4W1-2 Phase
	H	L	H	1/32	8W1-2 Phase
	H	H	L	1/64	16W1-2 Phase
	H	H	H	1/128	32W1-2 Phase
Driving current calculation formula		$I_{MAX} = V_{REF} \cdot 2$			$V_{REF} = (I_{MAX}) / 2$
$Rs = 0.1\Omega$		$I_{MAX} = \frac{V_{REF}}{1.1}$			$V_{REF} = I_{MAX} * 1.1$

Note: Use 90% of the calculated Vref when tuning the stepper driver board

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

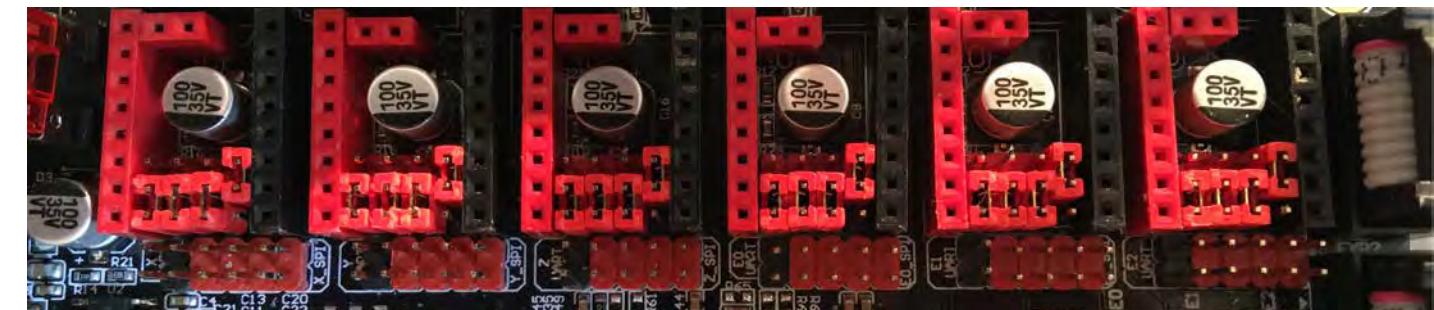
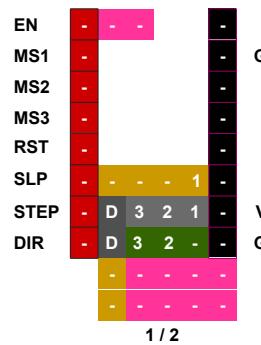
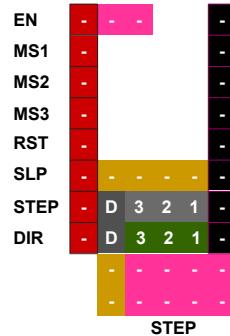


$Rs = R220$ is 0.22 Ohms

$Rs = R100$ is 0.1 Ohms

Ledge LV8729

Note: The "D" jumper MUST be SET!



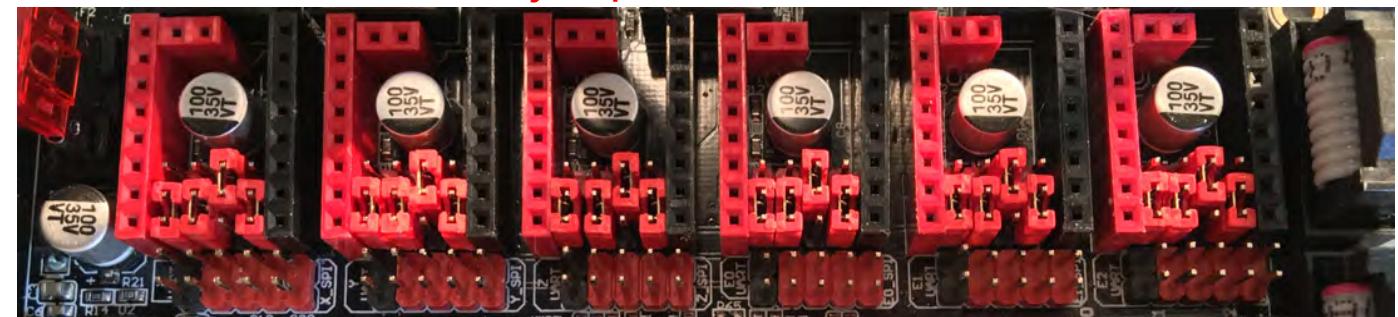
Note: The "D" jumper MUST be SET!

Lerdge LV8729

EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RST	-	-	-	-	1A
SLP	-	-	2	-	1B
STEP	-	D	3	2	1
DIR	-	D	3	1	-
					VDD
					GND

1 / 4

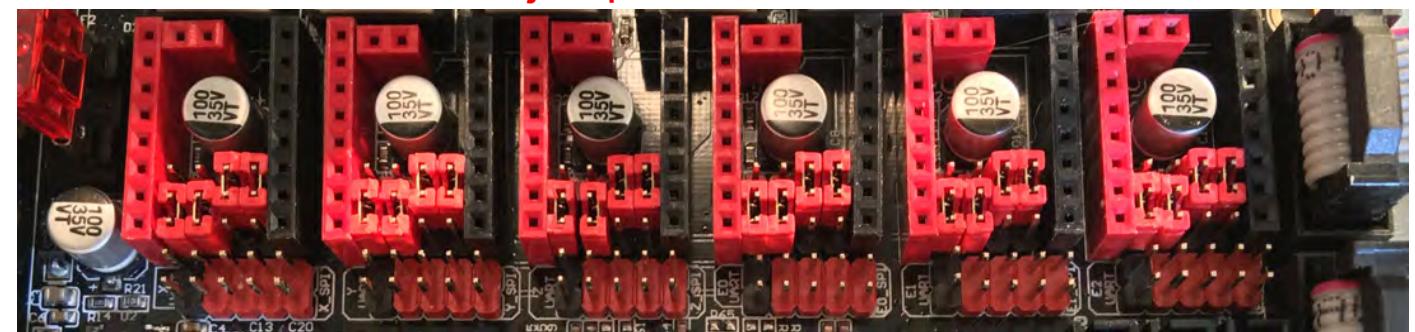
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RST	-	-	-	-	1A
SLP	-	-	2	1	1B
STEP	-	D	3	2	1
DIR	-	D	3	-	-
					VDD
					GND

1 / 8

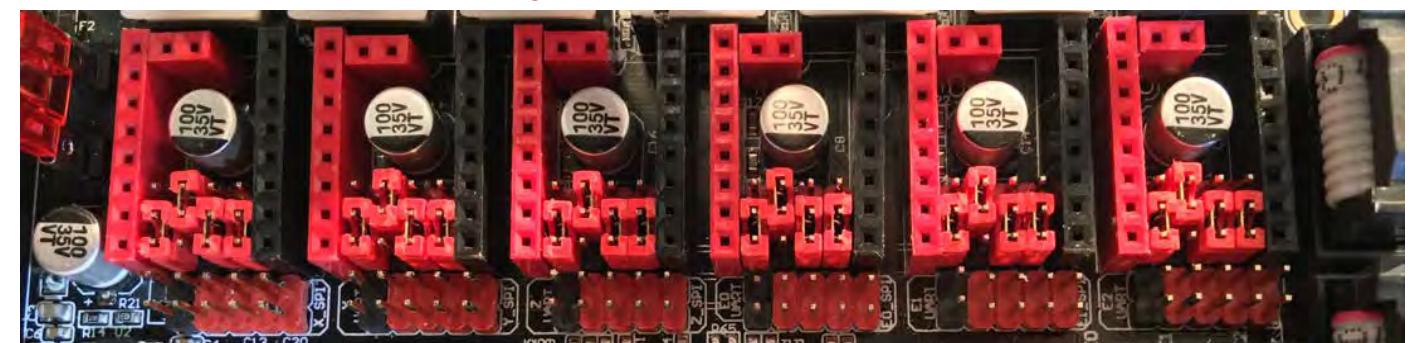
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	2B
MS3	-	-	-	-	2A
RST	-	-	-	-	1A
SLP	-	-	3	-	1B
STEP	-	D	3	2	1
DIR	-	D	2	1	-
					VDD
					GND

1 / 16

Note: The "D" jumper MUST be SET!

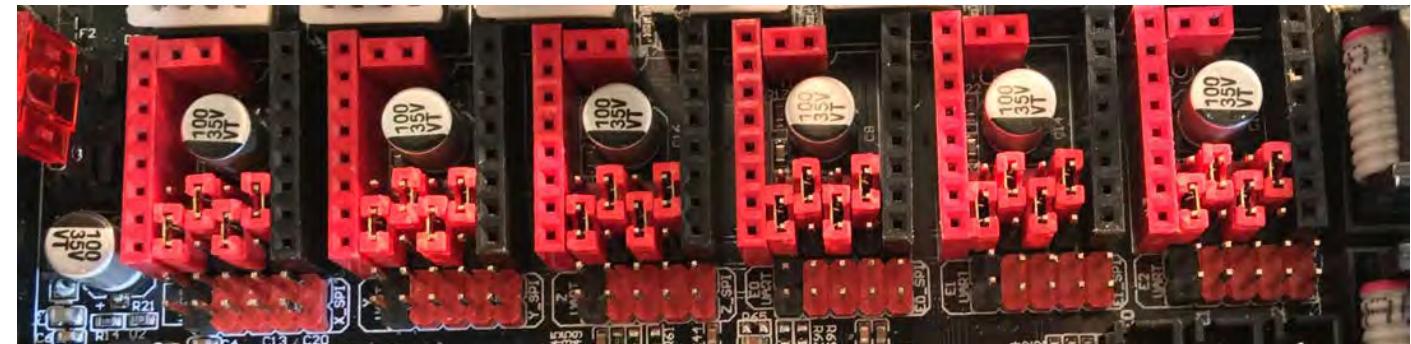


Lerdge LV8729

EN	-	-	-	-	VM		
MS1	-	-	-	-	GND		
MS2	-	-	-	-	2B		
MS3	-	-	-	-	2A		
RST	-	-	-	-	1A		
SLP	-	3	1	-	1B		
STEP	-	D	3	2	1	-	VDD
DIR	-	D	-	2	-	-	GND

1 / 32

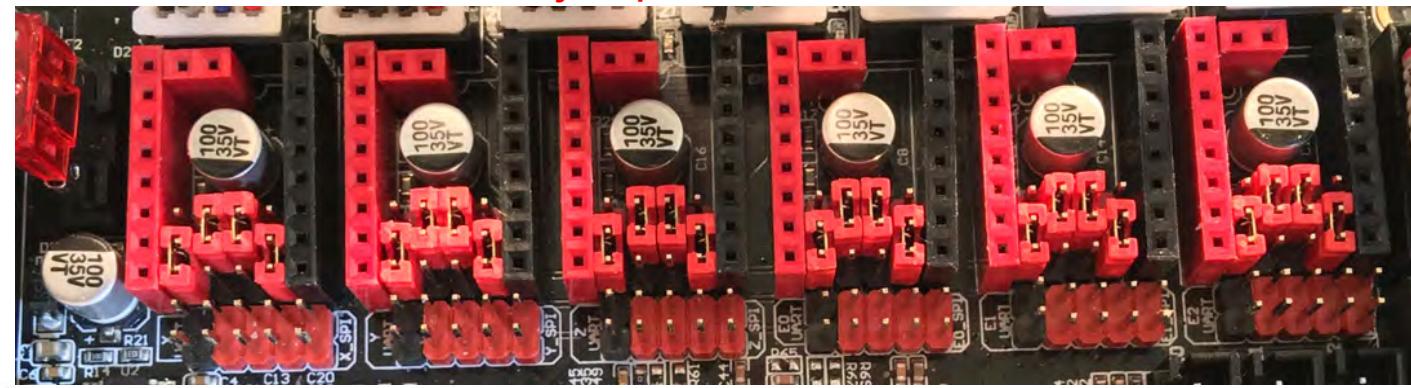
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM		
MS1	-	-	-	-	GND		
MS2	-	-	-	-	2B		
MS3	-	-	-	-	2A		
RST	-	-	-	-	1A		
SLP	-	3	2	-	1B		
STEP	-	D	3	2	1	-	VDD
DIR	-	D	-	2	-	-	GND

1 / 64

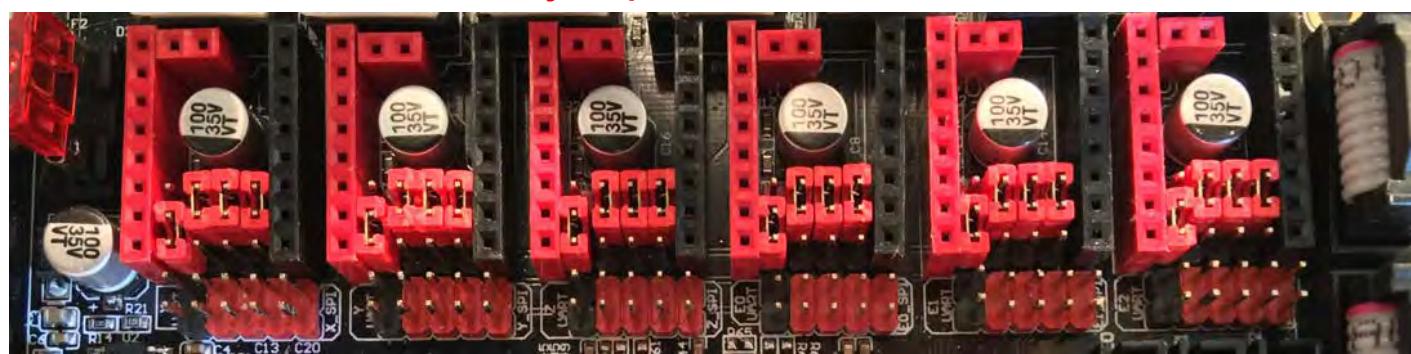
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM		
MS1	-	-	-	-	GND		
MS2	-	-	-	-	2B		
MS3	-	-	-	-	2A		
RST	-	-	-	-	1A		
SLP	-	3	2	1	1B		
STEP	-	D	3	2	1	-	VDD
DIR	-	D	-	2	-	-	GND

1 / 128

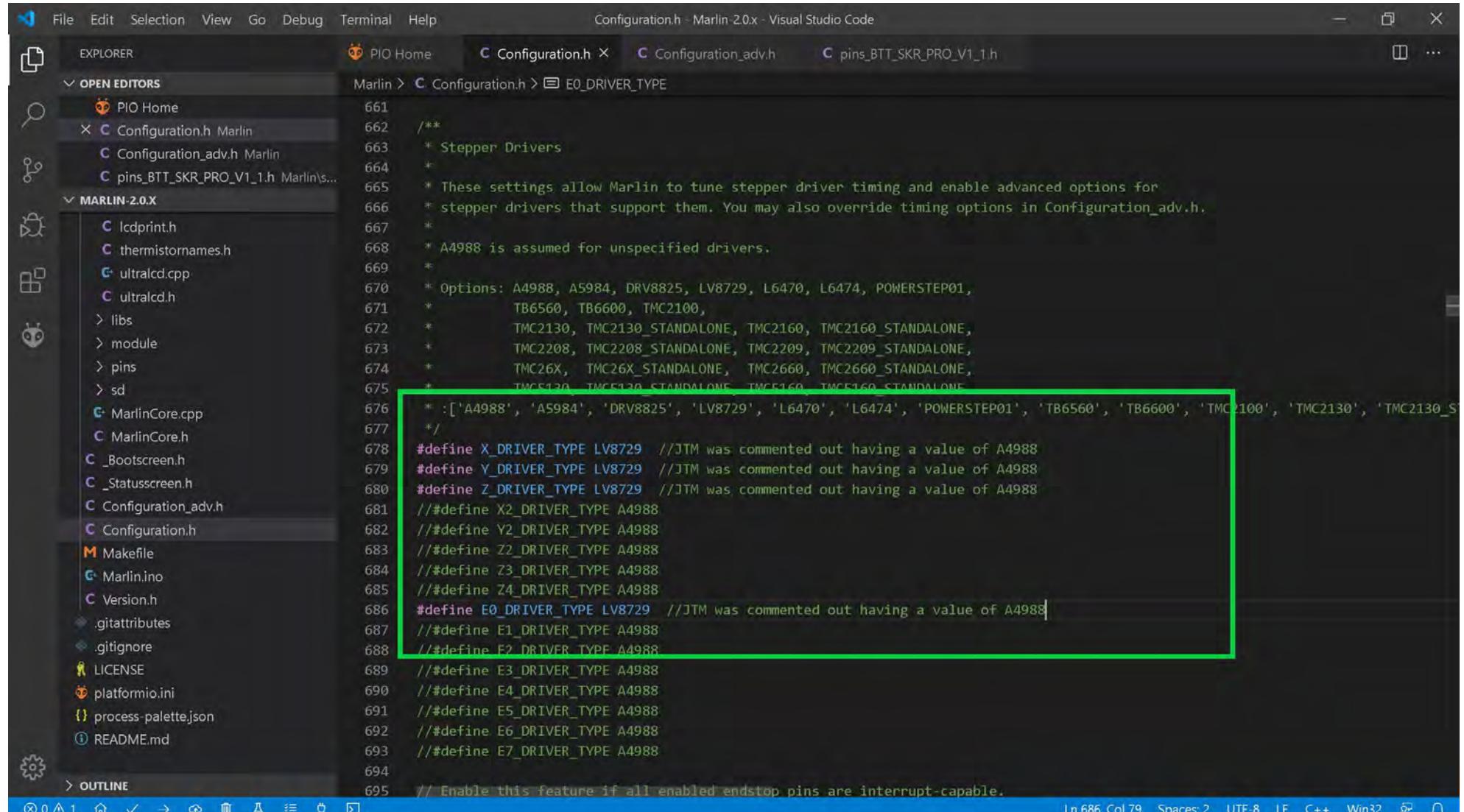
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for Lerdge LV8729 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for Lerdge LV8729 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using Lerdge LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use Lerdge LV8729 drivers. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following driver type definitions:

```

662 //*
663 * Stepper Drivers
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 */
668 * A4988 is assumed for unspecified drivers.
669 */
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 *           TB6560, TB6600, TMC2100,
672 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 *           TMC5120, TMC5120_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S'
677 */
678 #define X_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
679 #define Y_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
680 #define Z_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE LV8729 //JTM was commented out having a value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694 // Enable this feature if all enabled endstop pins are interrupt-capable.

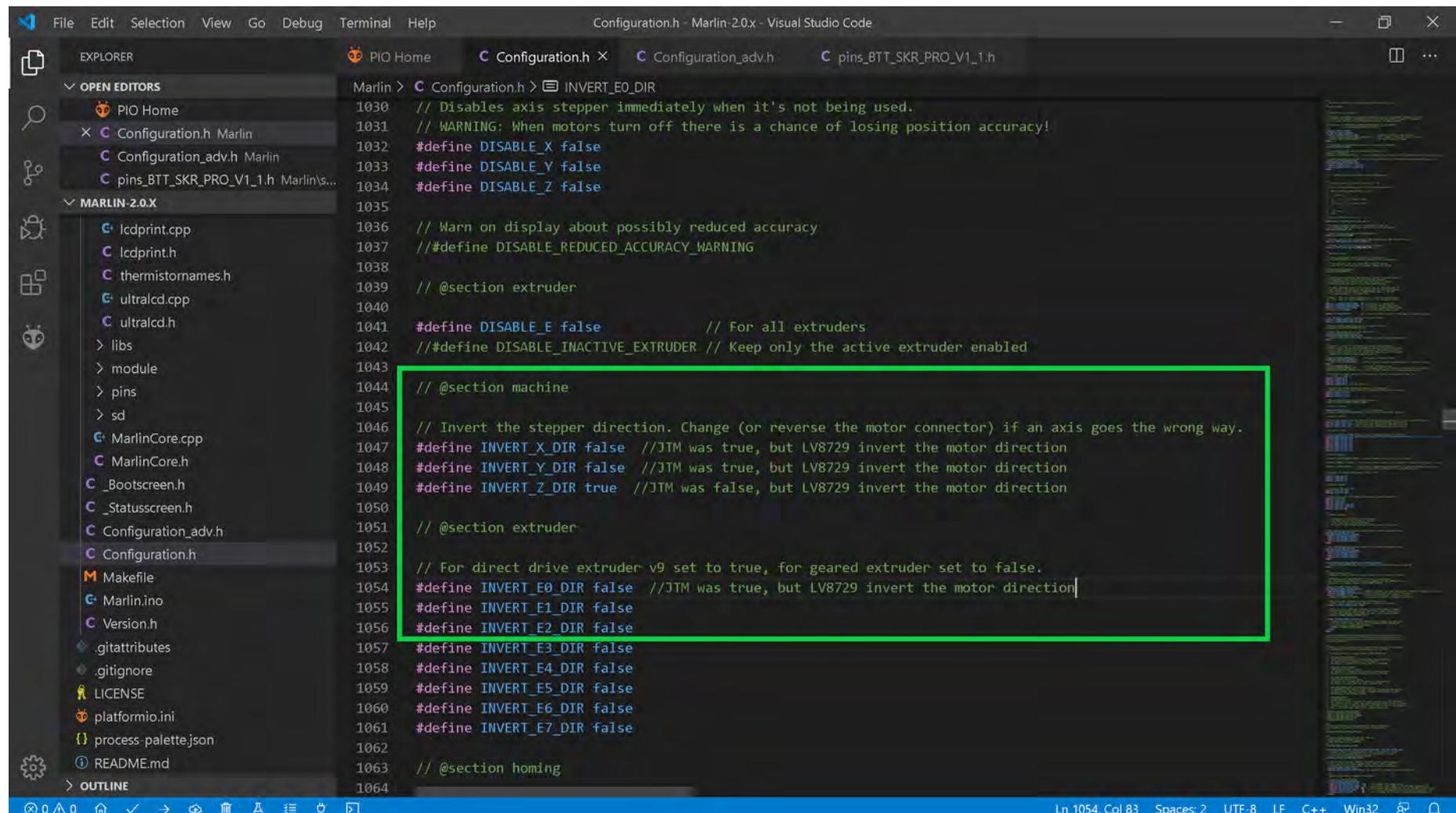
```

A green rectangular box highlights the driver type definitions for X, Y, Z, E0, E1, and E2 axes (lines 678-688). The rest of the code is shown in grey, indicating it is not selected.

- Go to the next page.

The (latest release of) Marlin Setup for Lerdge LV8729 Drivers

- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as show in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 // #define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false          // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false      // JTM was true, but LV8729 invert the motor direction
1048 #define INVERT_Y_DIR false      // JTM was true, but LV8729 invert the motor direction
1049 #define INVERT_Z_DIR true       // JTM was false, but LV8729 invert the motor direction
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false     // JTM was true, but LV8729 invert the motor direction
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

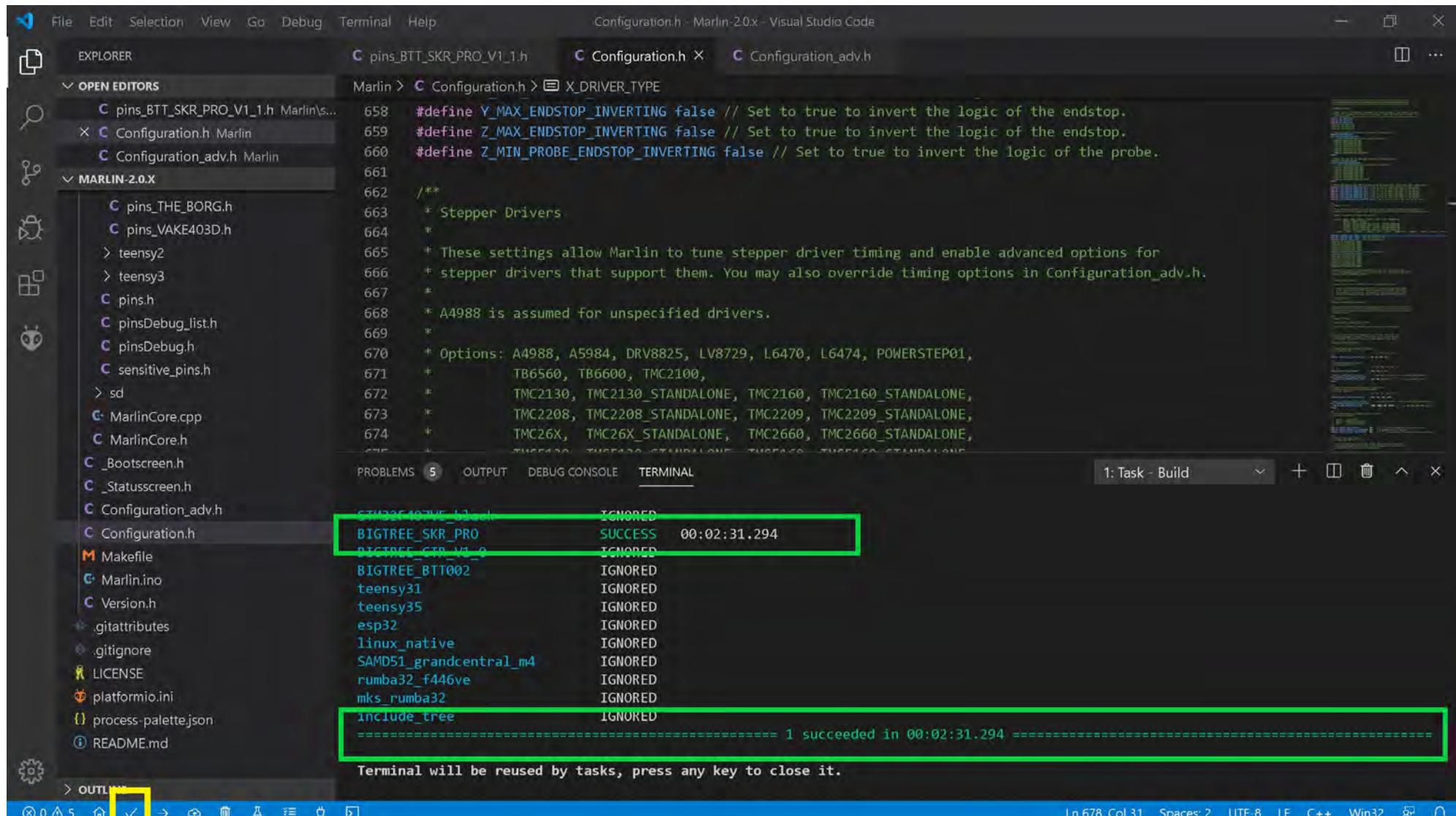
```

Ln 1054, Col 83 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for Lerdge LV8729 Drivers

- The end of Marlin setup for Lerdge LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the Marlin repository open. The terminal tab at the bottom displays the build log:

```

STM32F407VC-D3.c: IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_0: IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

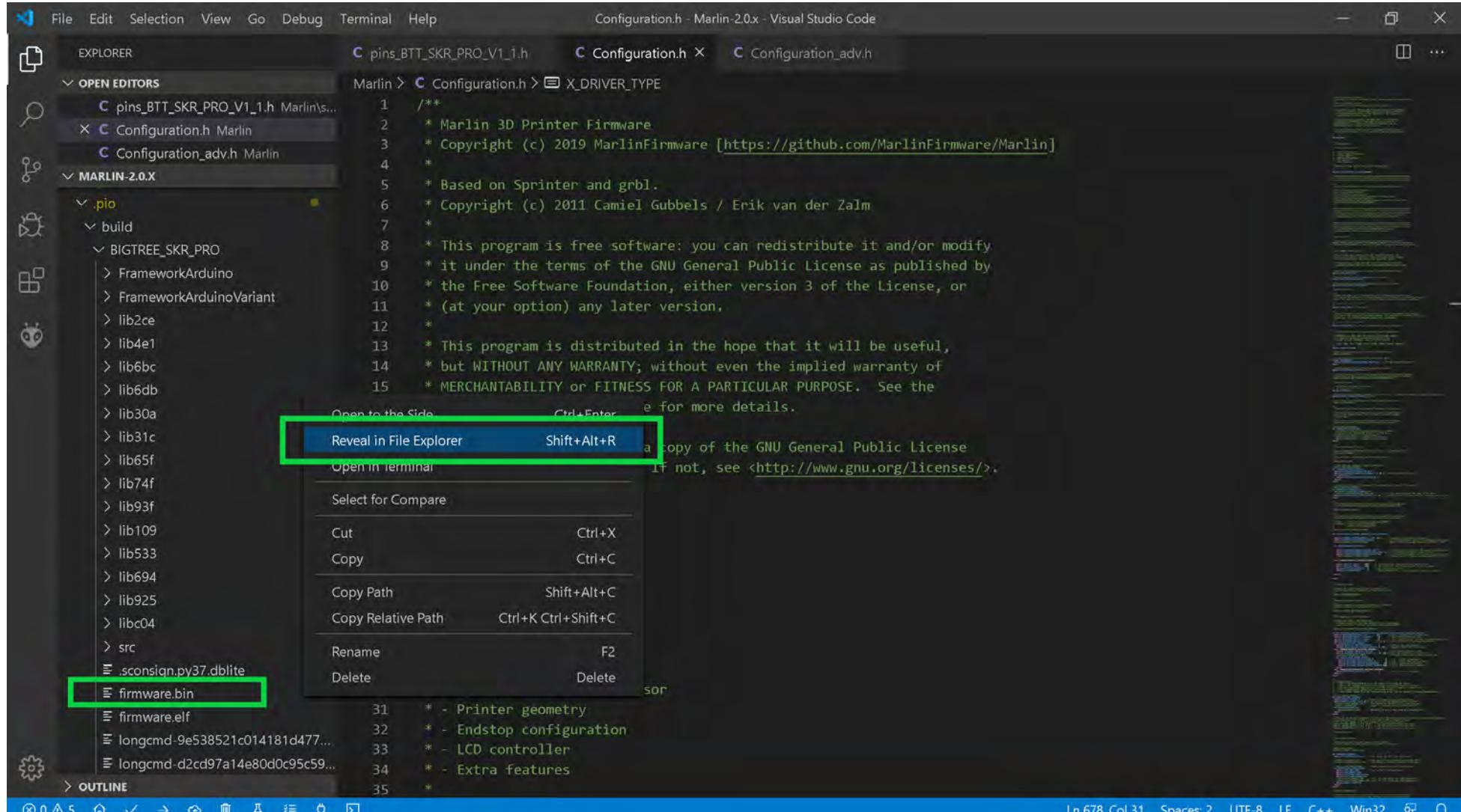
```

A yellow box highlights the checkmark icon in the terminal toolbar, indicating the build was successful. A green box highlights the terminal output showing the build results.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for Lerdge LV8729 Drivers

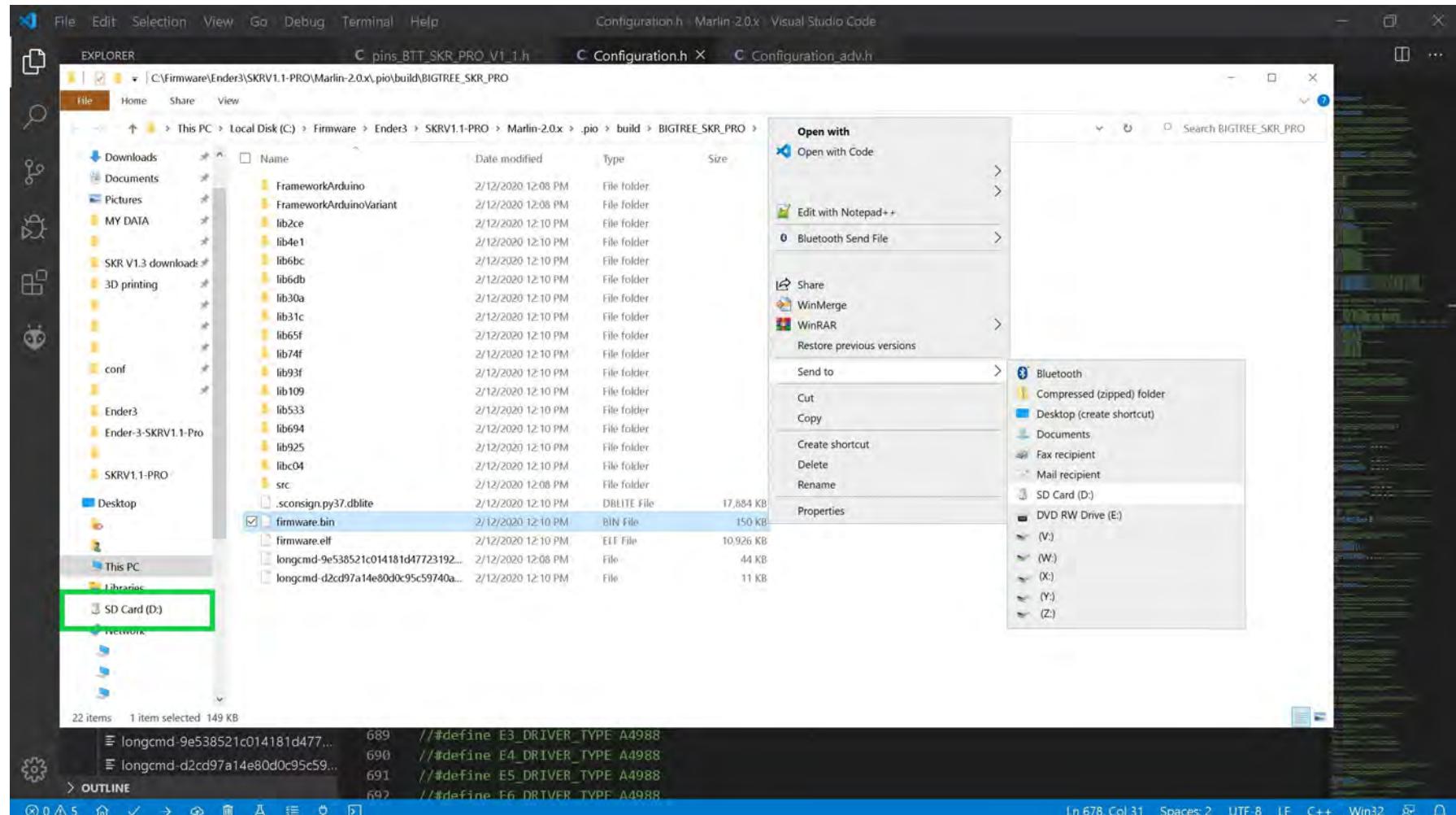
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

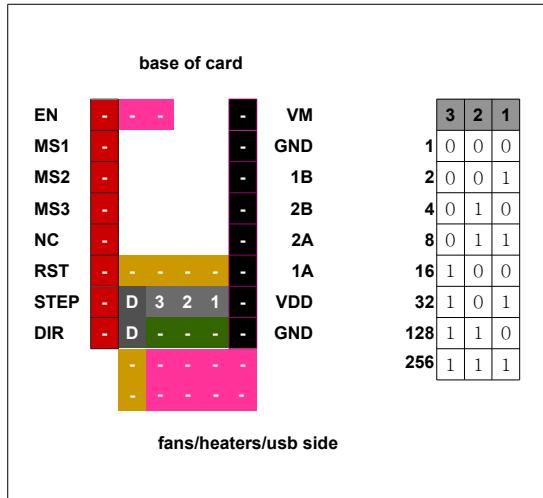
The (latest release of) Marlin Setup for Lerdge LV8729 Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

FYSETC ST820



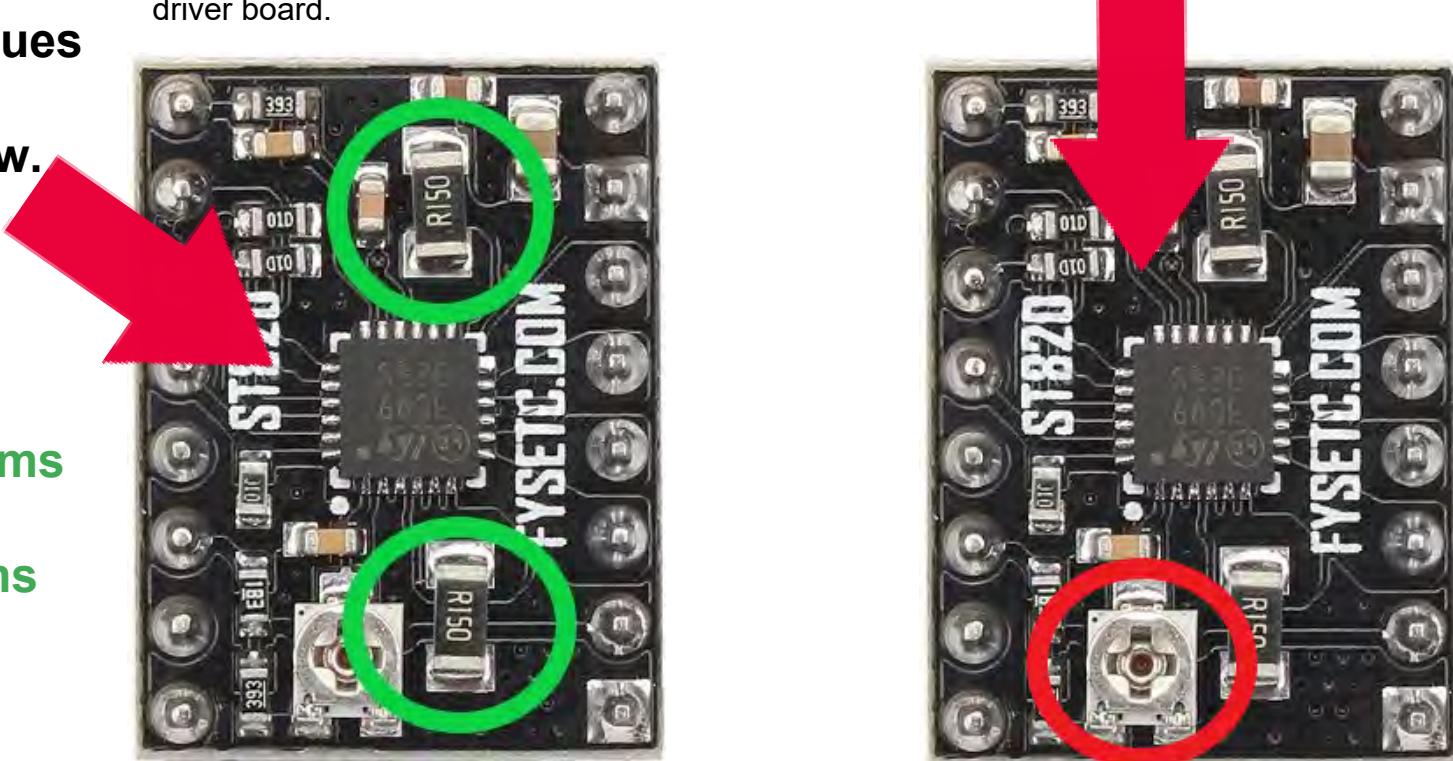
Driver Chip	MS3	MS2	MS1	Subdivision
ST820 Maximum 256 subdivision 45V 1.5A	L	L	L	Full Step
	L	L	H	1/2
	L	H	L	1/4
	L	H	H	1/8
	H	L	L	1/16
	H	L	H	1/32
	H	H	L	1/128
	H	H	H	1/256
Driving current calculation formula $R_s = 0.15\Omega$		$I_{MAX} = \frac{V_{REF} - V_{DD}}{5 * R_s}$		$V_{REF} = 5 * I_{MAX} * \frac{R_s}{V_{DD}}$
Note: Use 90% of the calculated Vref when tuning the stepper driver board				

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

$Rs = R_{150}$ is 0.15 Ohms

$Rs = R_{200}$ is 0.2 Ohms

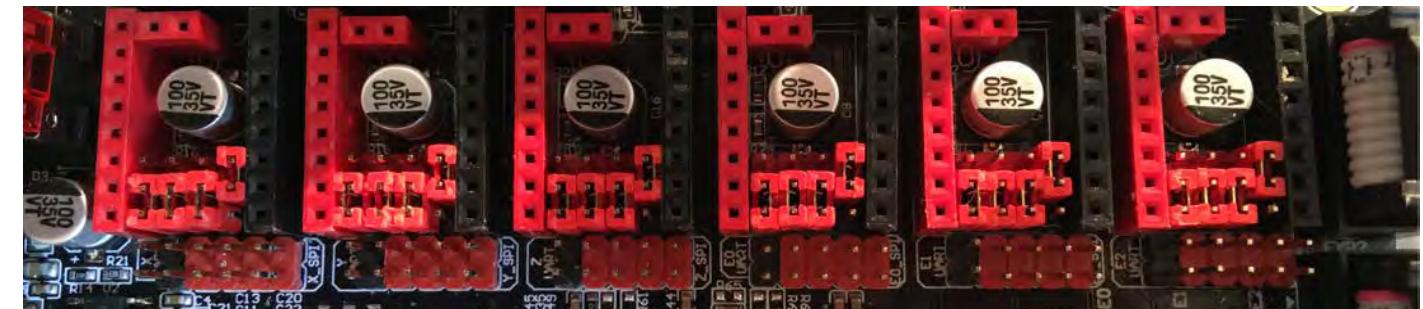
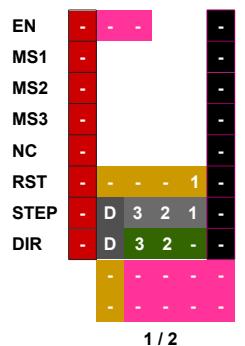
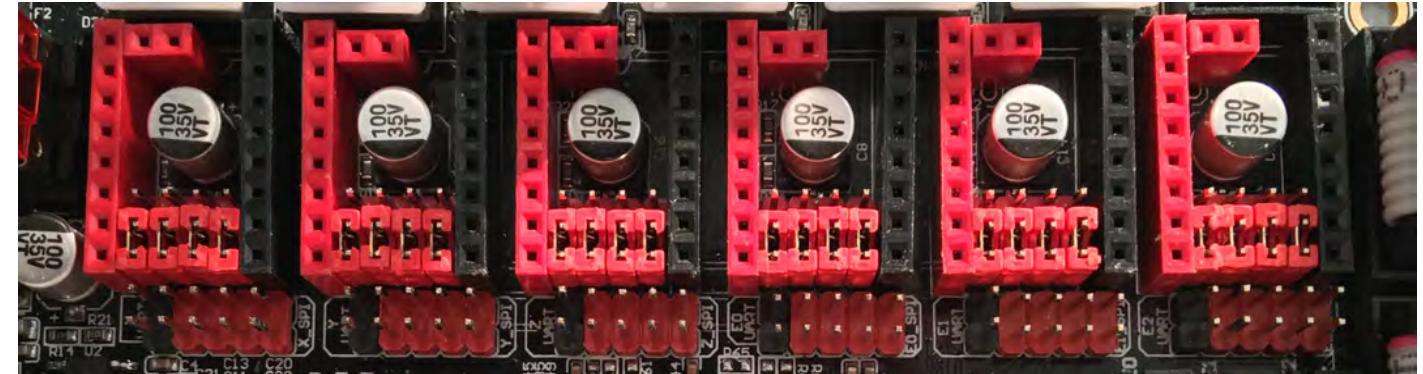
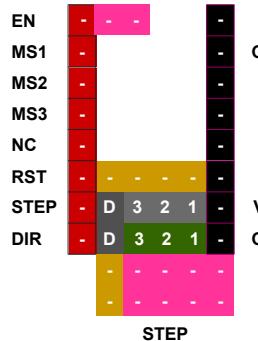
NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.



FYSETC ST820

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also: This driver requires constant cooling the moment any motor is used, or it will switch on and off.

Note: The "D" jumper MUST be SET!



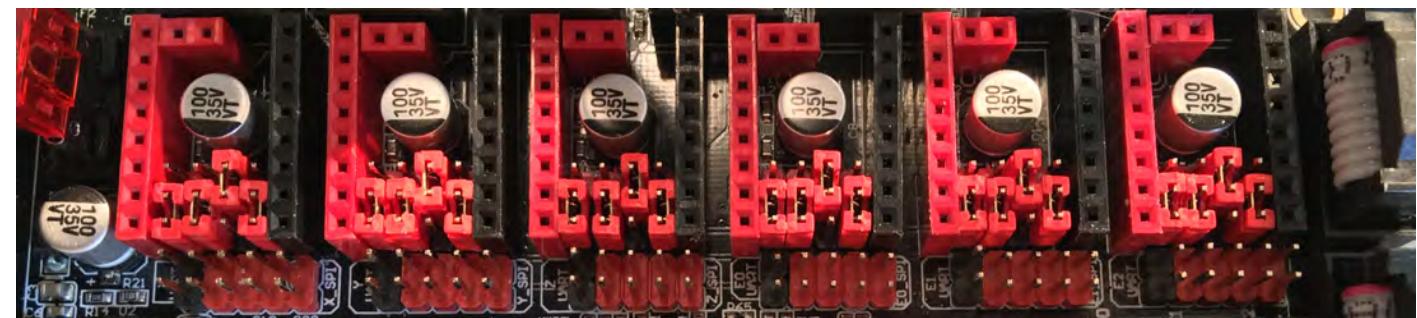
Note: The "D" jumper MUST be SET!

FYSETC ST820

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also: This driver requires constant cooling the moment any motor is used, or it will switch on and off.

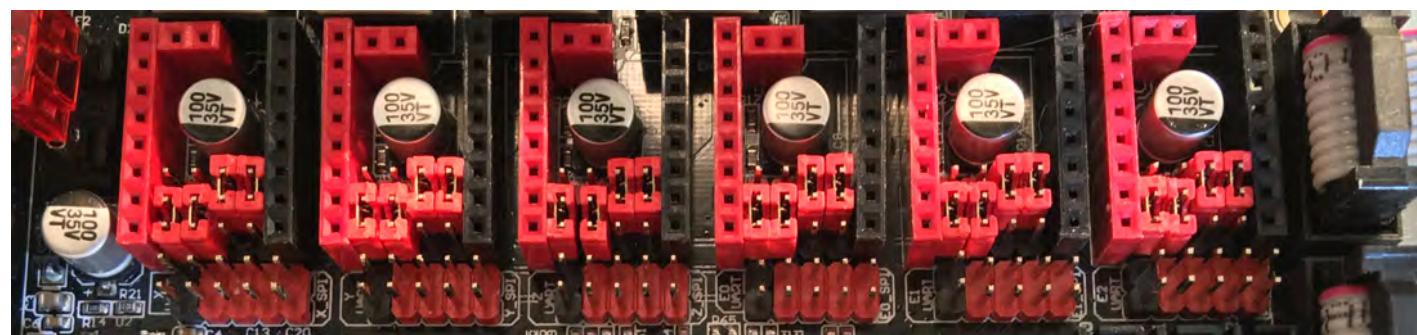
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	2	-
STEP	-	D	3 2 1	-
DIR	-	D	3 - 1	GND
1 / 4				

Note: The "D" jumper MUST be SET!



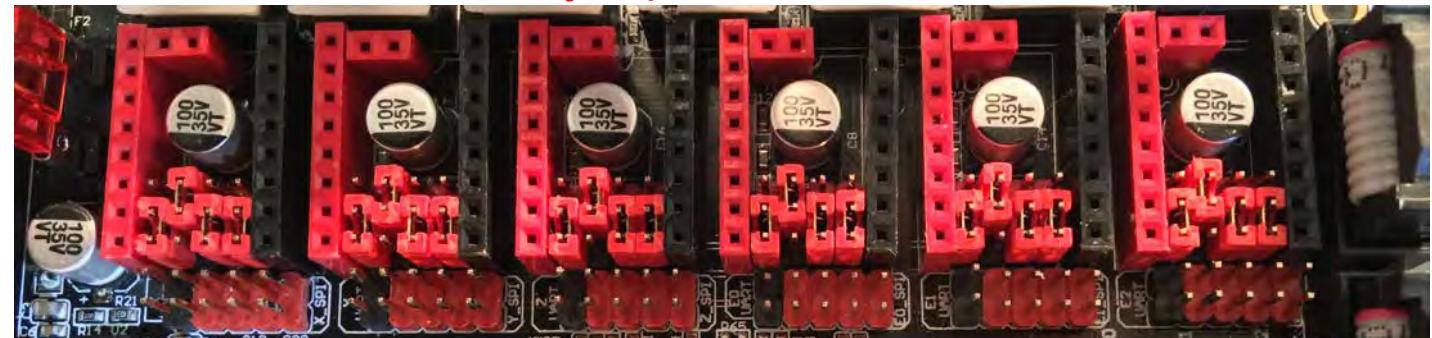
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	2 1	-
STEP	-	D	3 2 1	-
DIR	-	D	3 - -	GND
1 / 8				

Note: The "D" jumper MUST be SET!



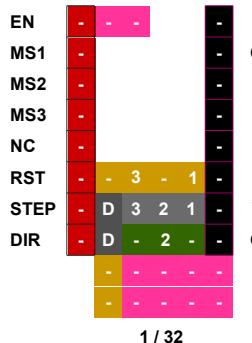
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	3	-
STEP	-	D	3 2 1	-
DIR	-	D	- 2 1	GND
1 / 16				

Note: The "D" jumper MUST be SET!

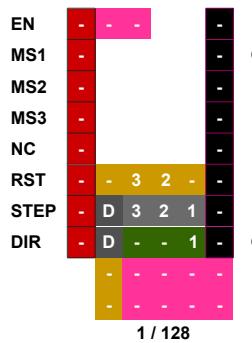
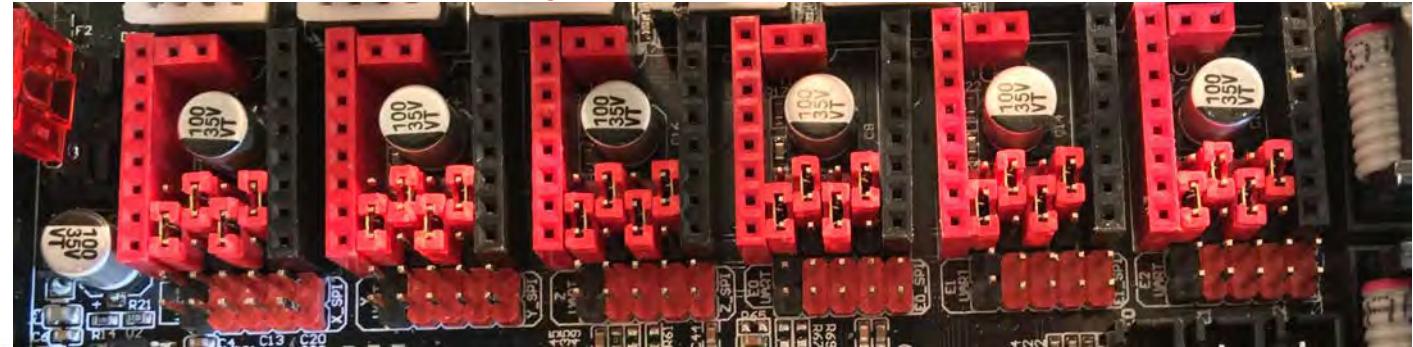


FYSETC ST820

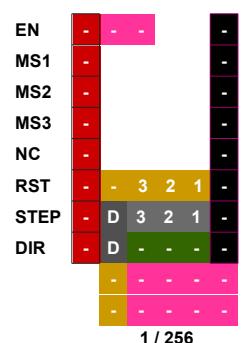
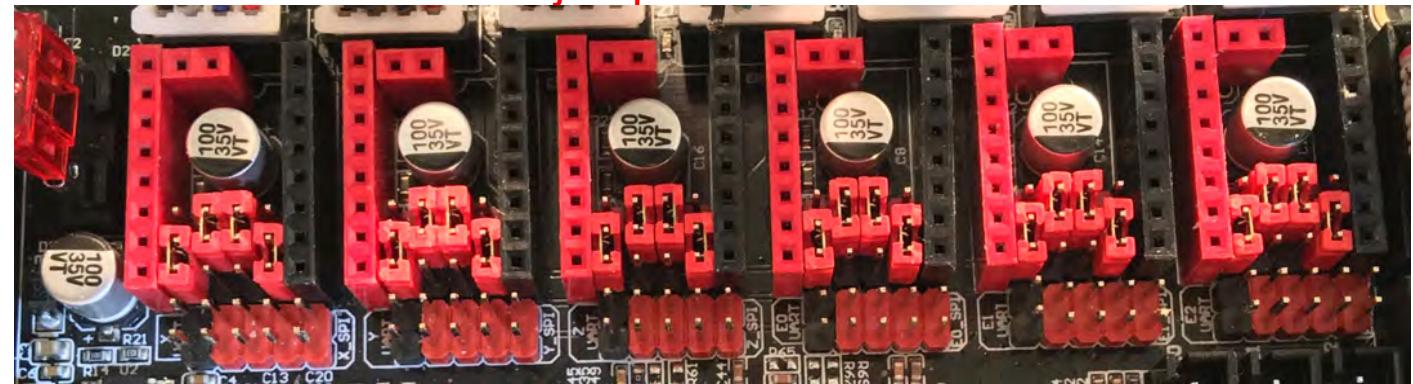
Important: This driver has special requirements in the Configuration and Configuration_adv.h Also:This driver requires constant cooling the moment any motor is used, or it will switch on and off.



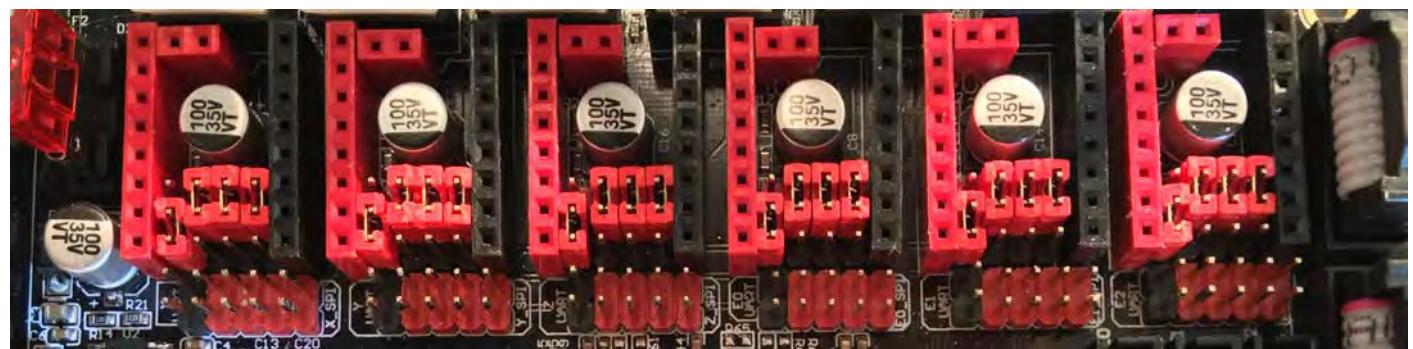
Note: The "D" jumper MUST be SET!



Note: The "D" jumper MUST be SET!



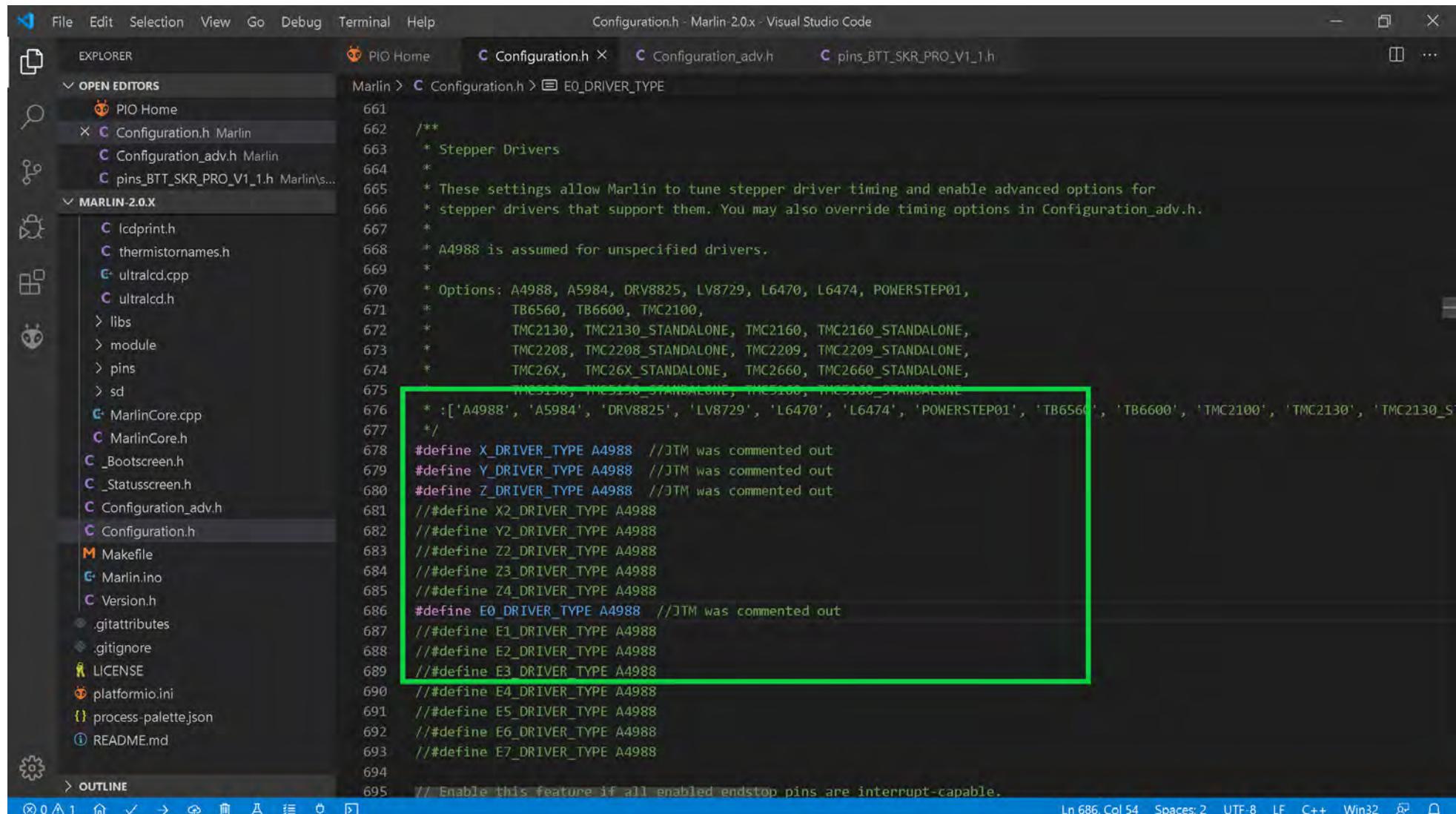
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for FYSETC ST820 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for FYSETC ST820 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using FYSETC ST820 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC ST820 drivers. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The **ST820** is a drop in replacement for the **A4988**. Since Marlin does not have an option for **ST820** we will use the **A4988** option.



```

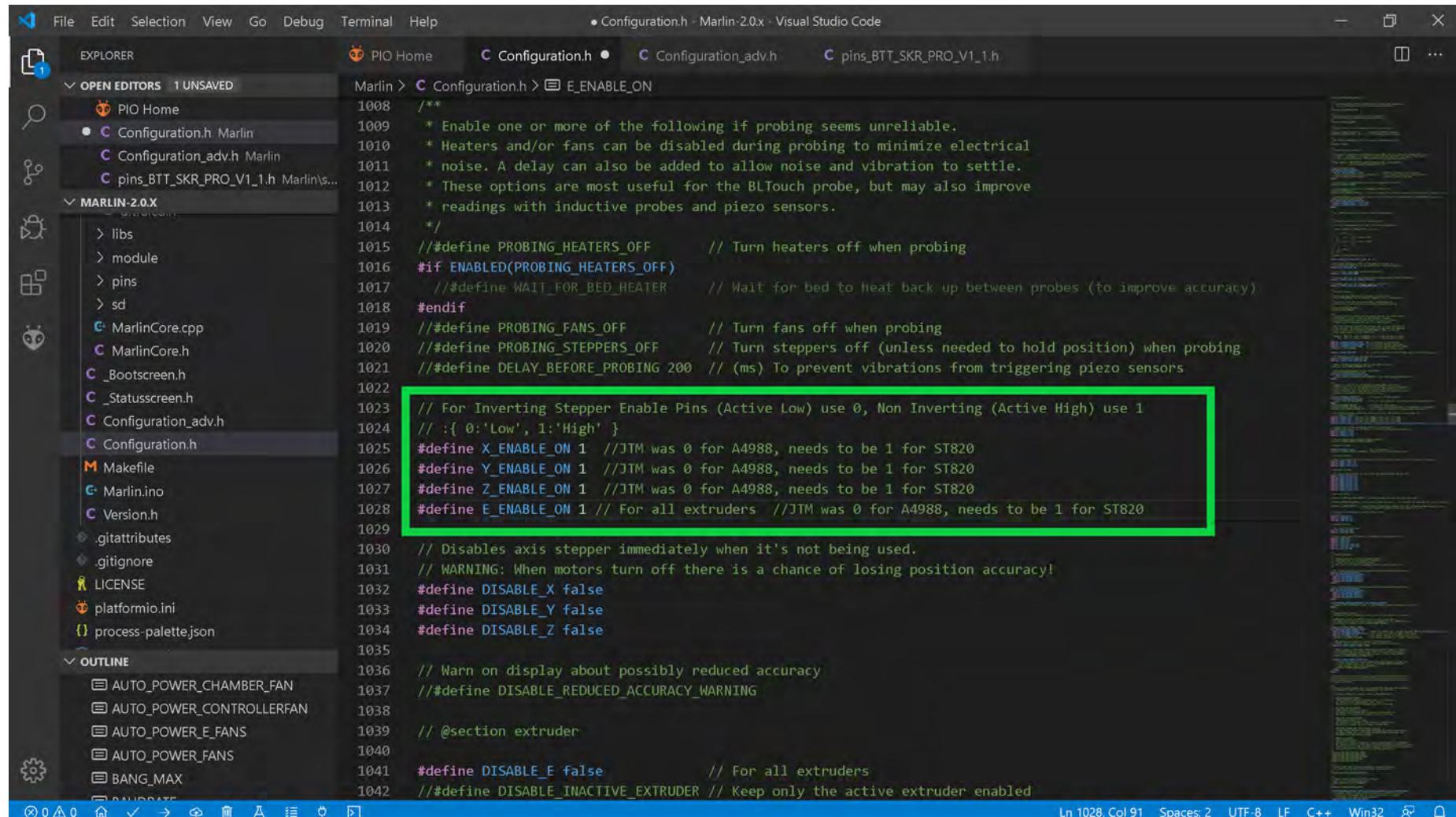
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > EO_DRIVER_TYPE
661
662 /**
663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 *           TB6560, TB6600, TMC2100,
672 *           TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 *           TMC316, TMC316_STANDALONE, TMC3160, TMC3160_STANDALONE
676 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160'
677 */
#define X_DRIVER_TYPE A4988 //JTM was commented out
#define Y_DRIVER_TYPE A4988 //JTM was commented out
#define Z_DRIVER_TYPE A4988 //JTM was commented out
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define EO_DRIVER_TYPE A4988 //JTM was commented out
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC ST820 Drivers

- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x configuration header. A green rectangular box highlights the following code snippet:

```

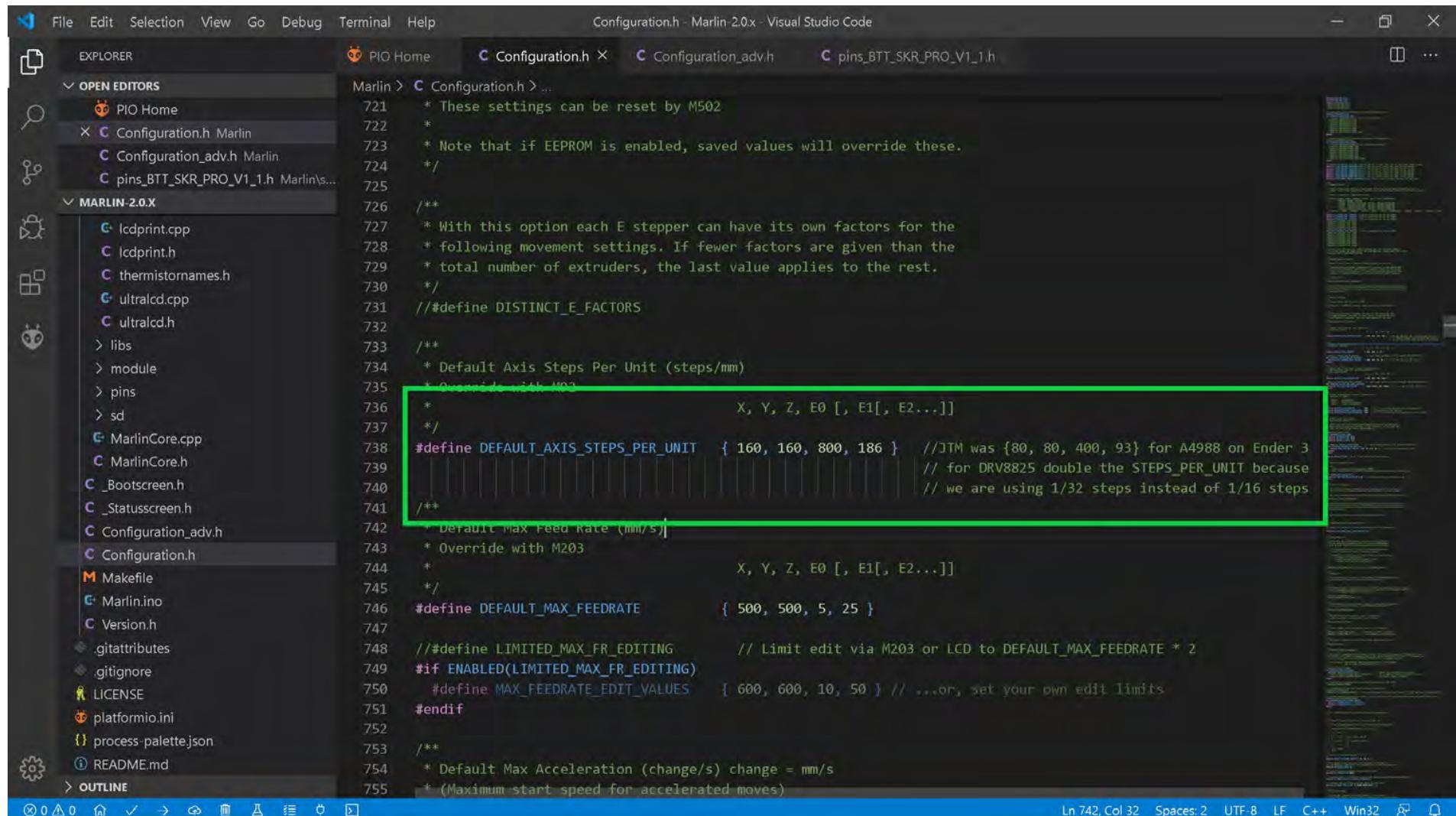
1024 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1025 // :{ 0:'Low', 1:'High' }
1026 #define X_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1027 #define Y_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1028 #define Z_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1029 #define E_ENABLE_ON 1 // For all extruders //JTM was 0 for A4988, needs to be 1 for ST820

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC ST820 Drivers

- We are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC ST820 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16.** So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Marlin Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration.h ...

```

721 * These settings can be reset by M502
722 *
723 * Note that if EEPROM is enabled, saved values will override these.
724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per Unit (steps/mm)
735 * 0 = X, 1 = Y, 2 = Z, 3 = E0 [ , E1[ , E2... ] ]
736 */
737 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // JTM was {80, 80, 400, 93} for A4988 on Ender 3
738 // for DRV8825 double the STEPS_PER_UNIT because
739 // we are using 1/32 steps instead of 1/16 steps
740
741 /**
742 * Default Max Feed Rate (mm/s)
743 * Override with M203
744 * 0 = X, 1 = Y, 2 = Z, 3 = E0 [ , E1[ , E2... ] ]
745 */
746 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
747
748 // #define LIMITED_MAX_FR_EDITING
749 #if ENABLED(LIMITED_MAX_FR_EDITING)
750 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
751 #endif
752
753 /**
754 * Default Max Acceleration (change/s) change = mm/s
755 * (Maximum start speed for accelerated moves)

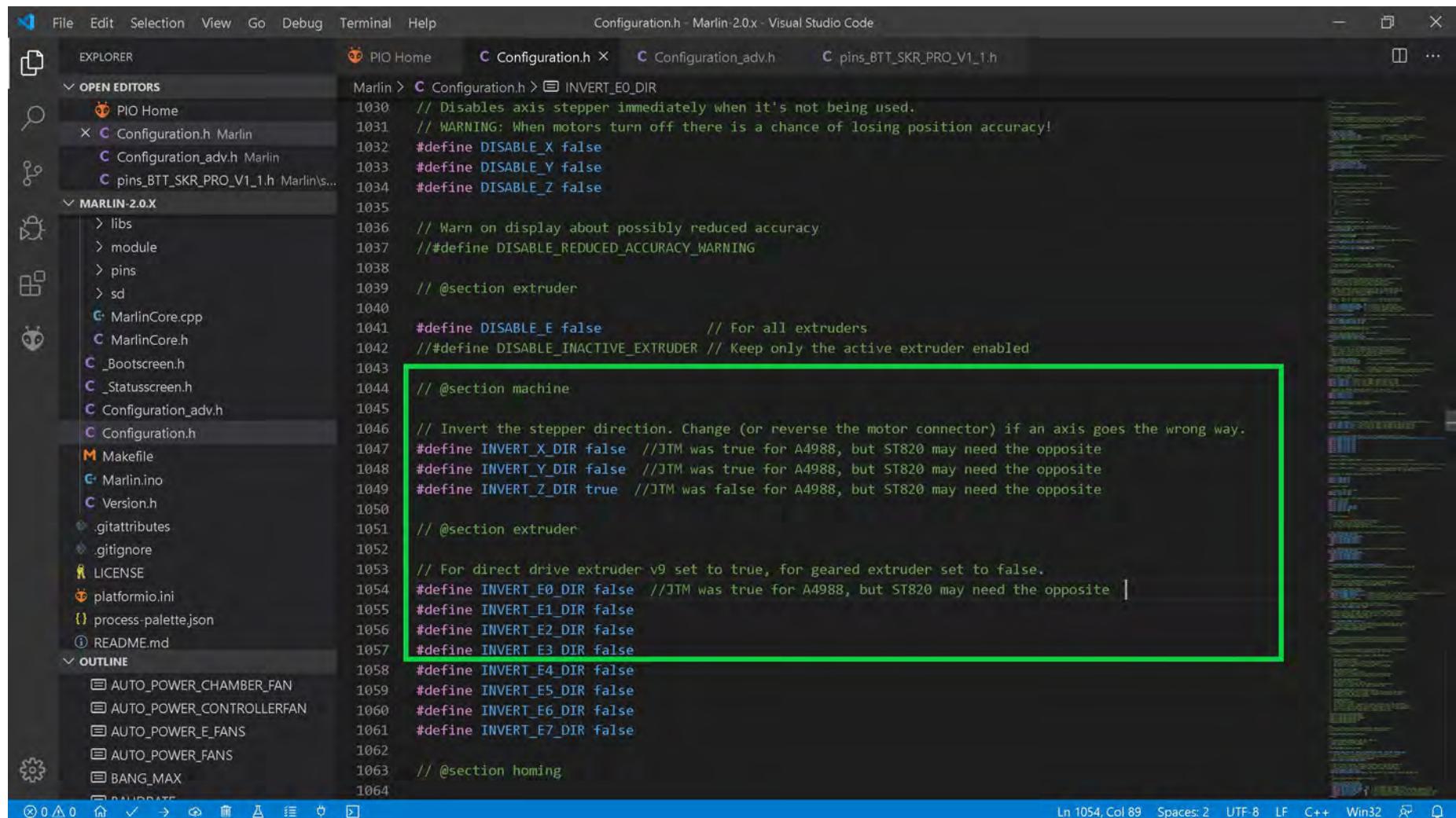
```

Ln 742, Col 32 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC ST820 Drivers

- **Optional Step:** I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") https://reprap.org/wiki/Stepper_wiring. Other people prefer to change the motor direction in the Marlin firmware. **So if you want or need to change the motor direction in Marlin**, then if the axis' setting you will be using the ST820 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 //#define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false          // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false      // JTM was true for A4988, but ST820 may need the opposite
1048 #define INVERT_Y_DIR false      // JTM was true for A4988, but ST820 may need the opposite
1049 #define INVERT_Z_DIR true       // JTM was false for A4988, but ST820 may need the opposite
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false    // JTM was true for A4988, but ST820 may need the opposite
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

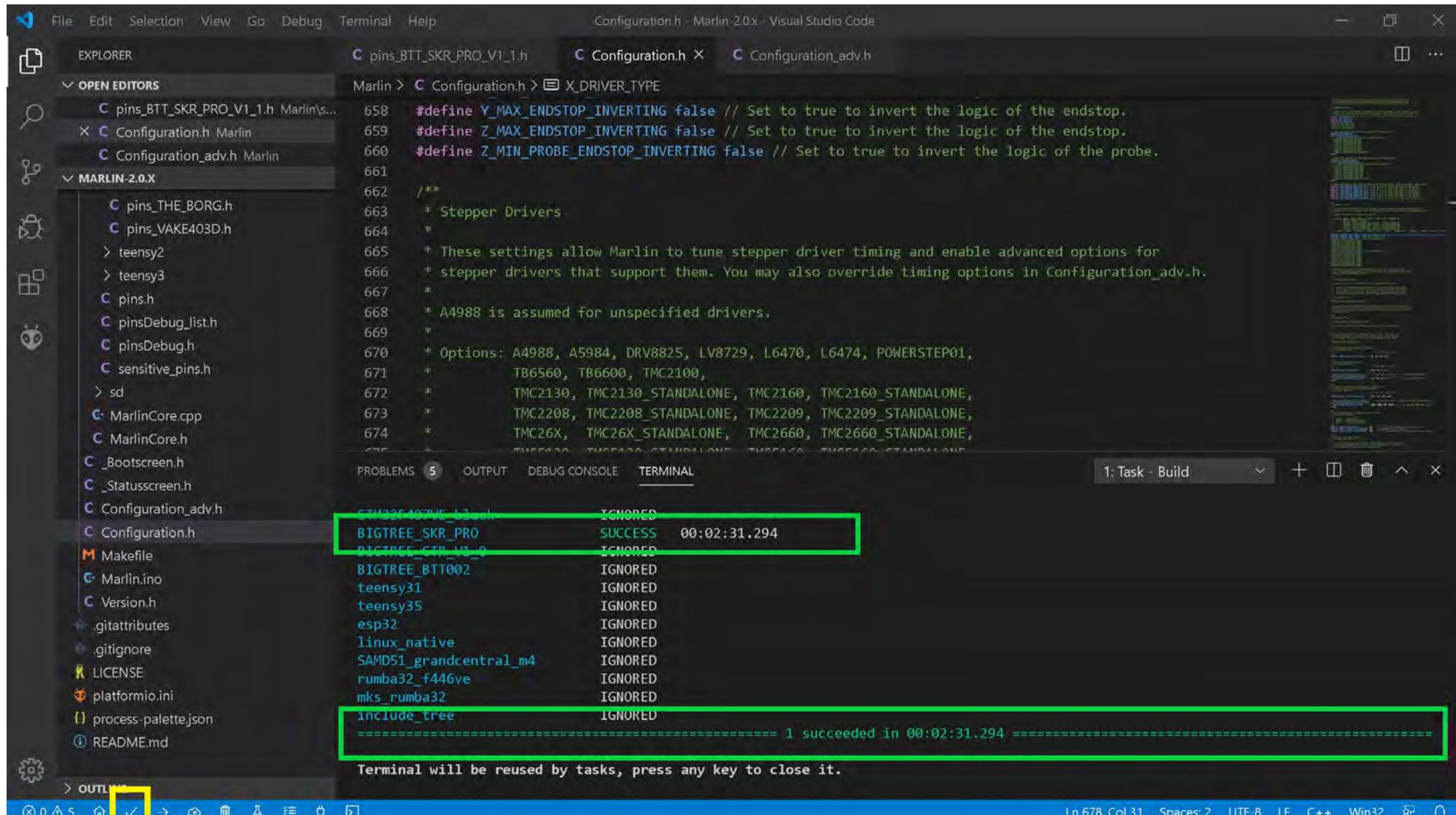
```

Ln 1054, Col 89 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC ST820 Drivers

- The end of Marlin setup for FYSETC ST820 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h

MARLIN-2.0.X pins_BTT_SKR_PRO_V1_1.h Marlin\s... Configuration.h X Configuration_adv.h

```
Marlin > Configuration.h > X_DRIVER_TYPE
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**
663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build +

STM32F407VC-DI	IGNORED
BIGTREE_SKR_PRO	SUCCESS 00:02:31.294
BIGTREE_SKR_V3_0	IGNORED
BIGTREE_BTT002	IGNORED
teensy31	IGNORED
teensy35	IGNORED
esp32	IGNORED
linux_native	IGNORED
SAMD51_grandcentral_m4	IGNORED
rumba32_f446ve	IGNORED
mks_rumba32	IGNORED
include_tree	IGNORED

===== 1 succeeded in 00:02:31.294 =====

Terminal will be reused by tasks, press any key to close it.

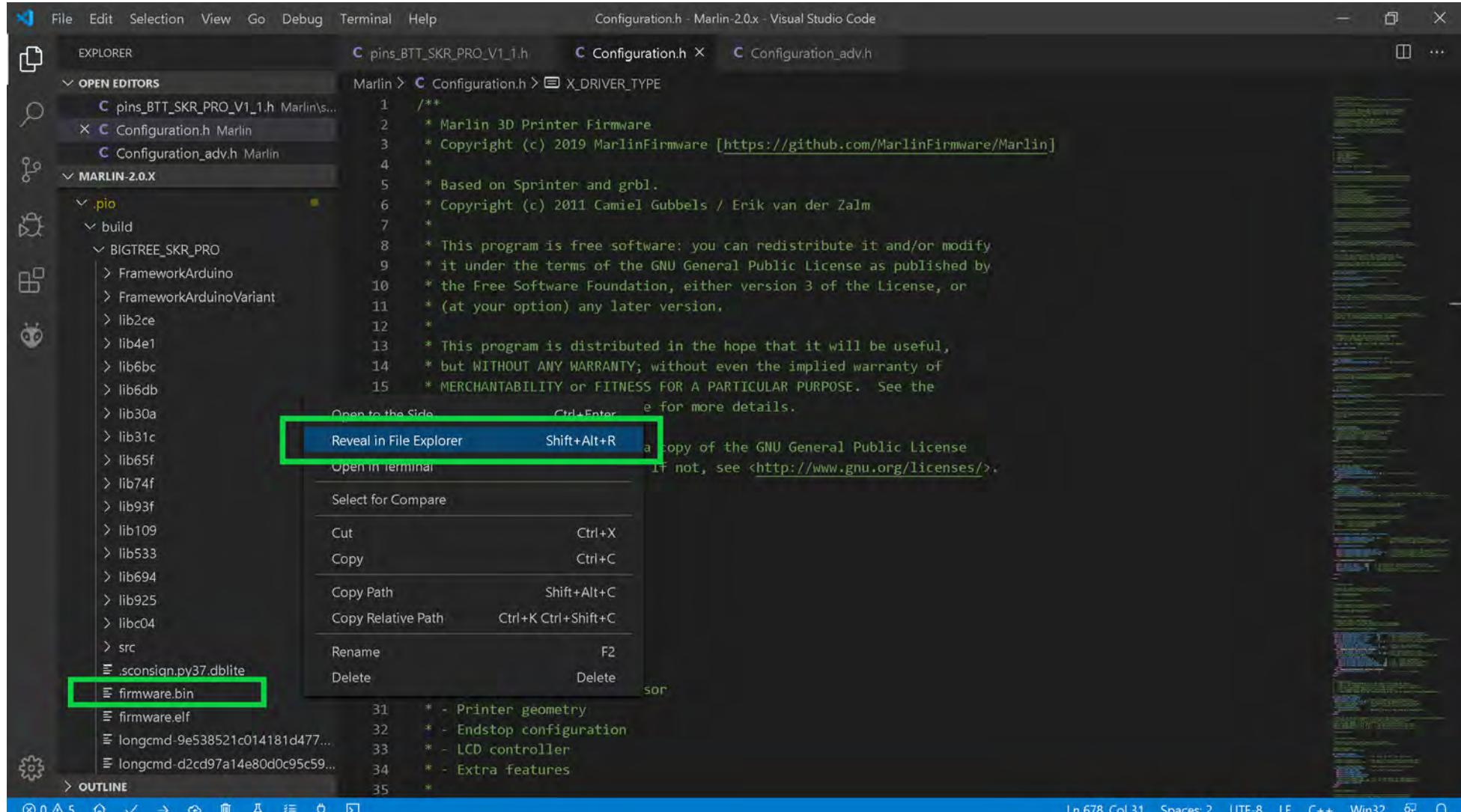
OUTLINE 

Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for FYSETC ST820 Drivers

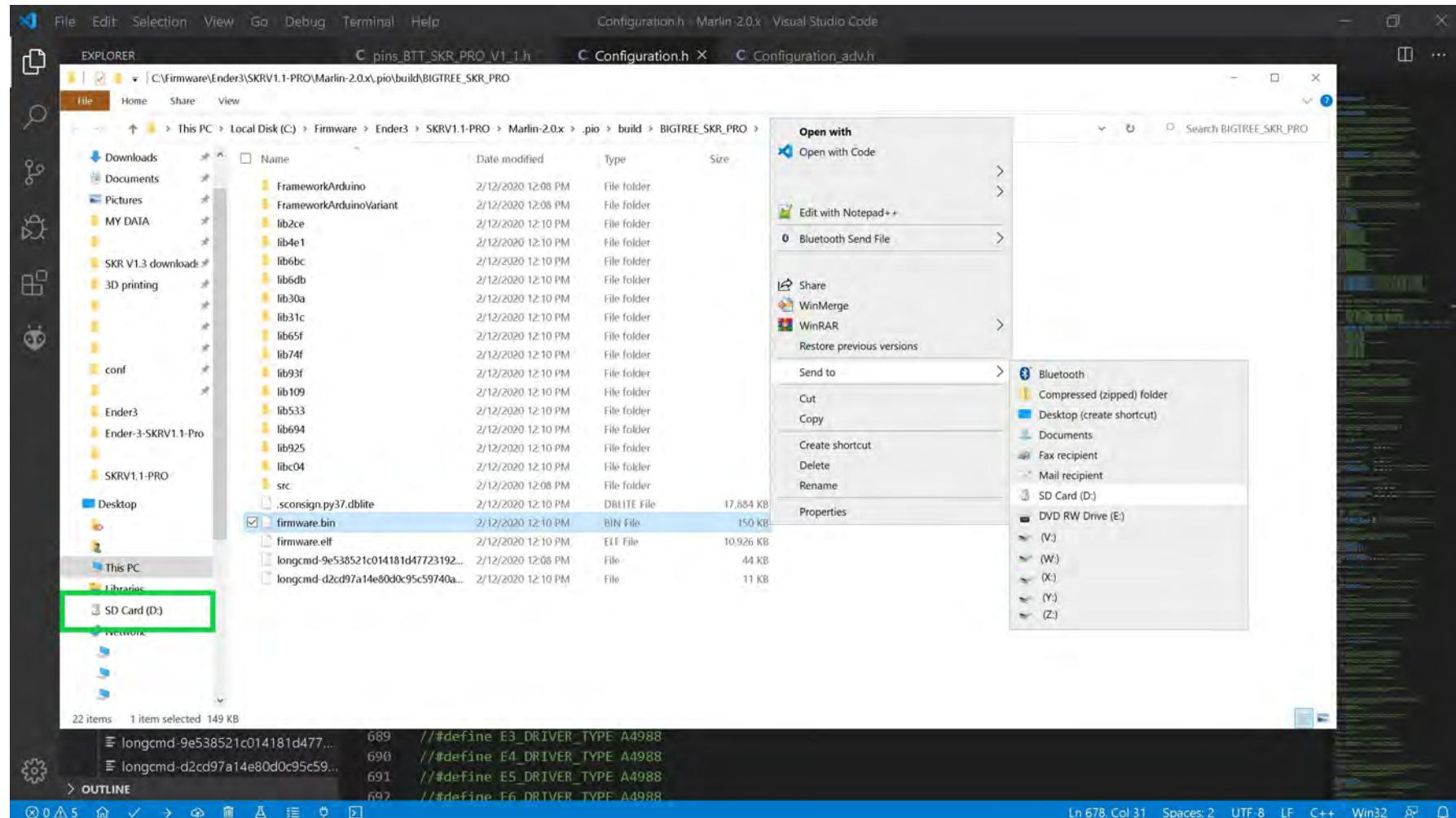
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

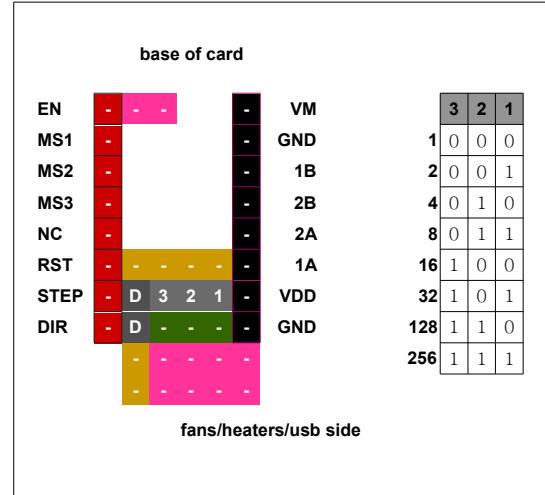
The (latest release of) Marlin Setup for FYSETC ST820 Drivers

- Ensure your micro-SD card, that came with your SKR PRO V1.1 board, is in a micro-to-SD-card adapter and plugged into yourWindow's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

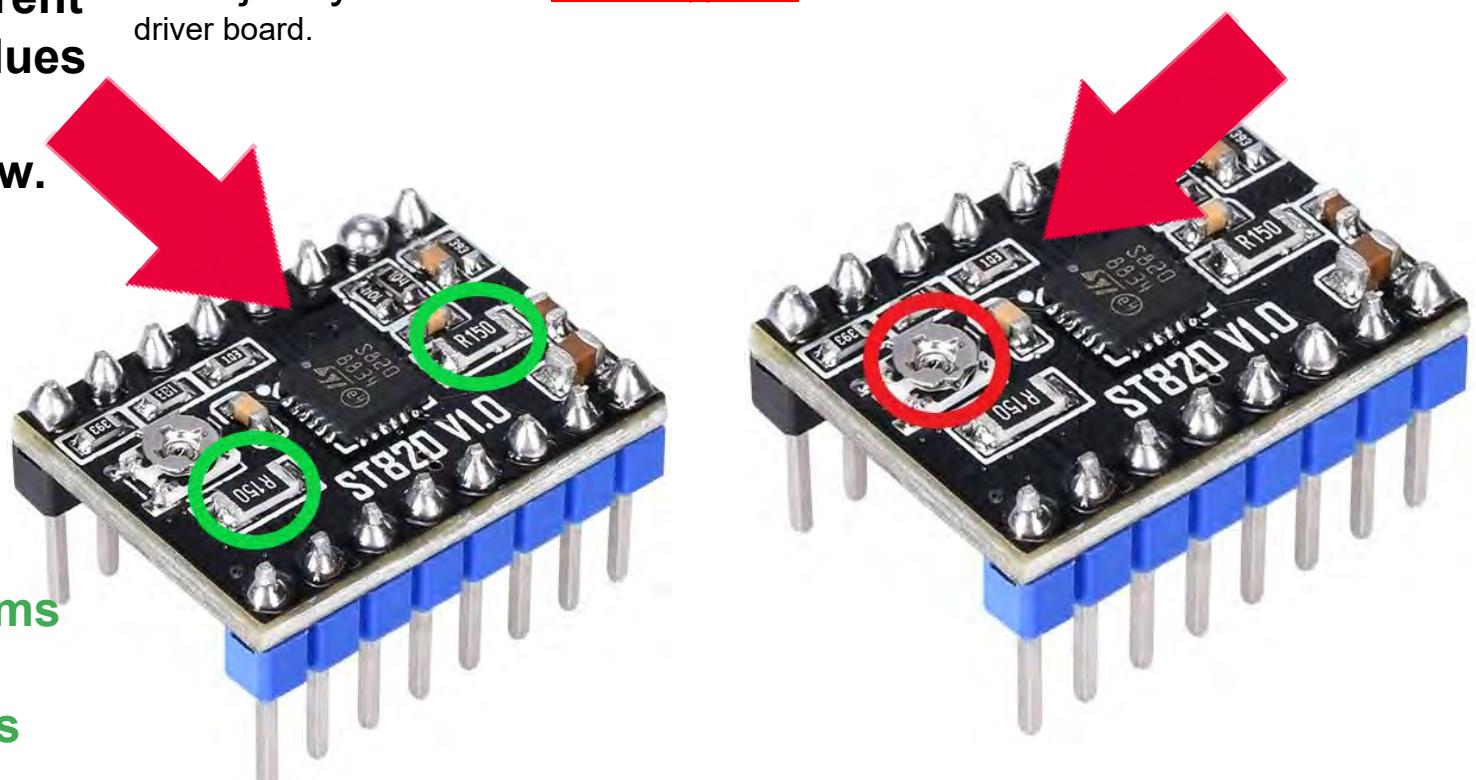
BIQU ST820



Driver Chip	MS3	MS2	MS1	Subdivision
ST820 Maximum 256 subdivision 45V 1.5A	L	L	L	Full Step
	L	L	H	1/2
	L	H	L	1/4
	L	H	H	1/8
	H	L	L	1/16
	H	L	H	1/32
	H	H	L	1/128
	H	H	H	1/256
Driving current calculation formula $R_s = 0.15\Omega$		$I_{MAX} = \frac{V_{REF} - V_{DD}}{5 * R_s}$		$V_{REF} = 5 * I_{MAX} * \frac{R_s}{V_{DD}}$
Note: Use 90% of the calculated Vref when tuning the stepper driver board				

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

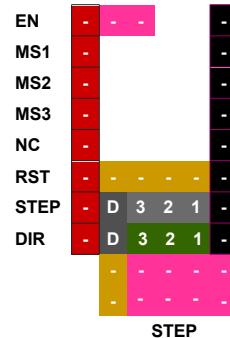


Rs = R150 is 0.15 Ohms

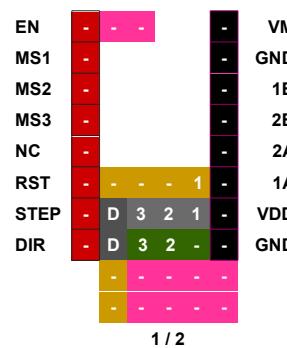
Rs = R200 is 0.2Ohms

Biqu ST820

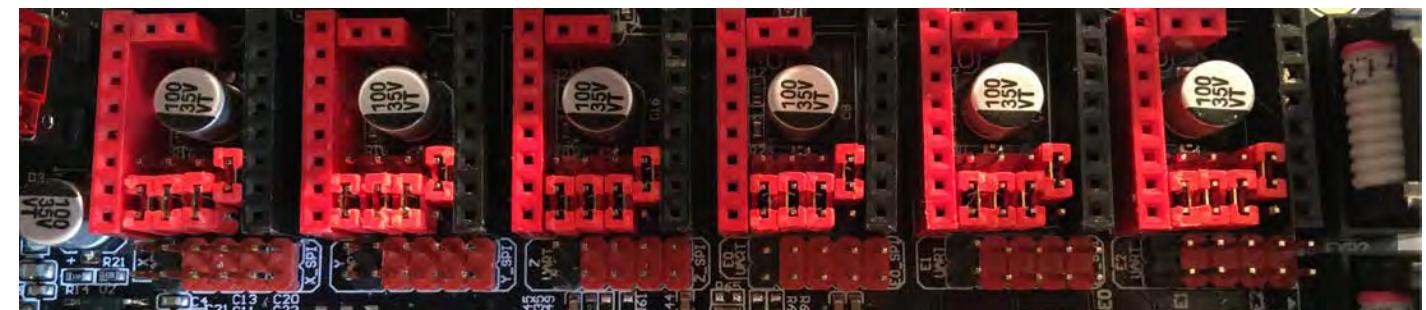
Important: This driver has special requirements in the Configuration and Configuration_adv.h Also: This driver requires constant cooling the moment any motor is used, or it will switch on and off.



Note: The "D" jumper MUST be SET!



Note: The "D" jumper MUST be SET!

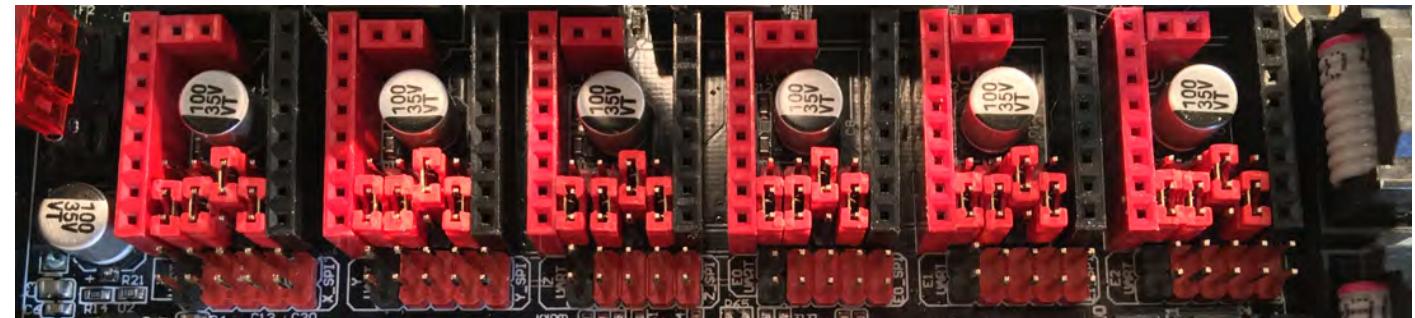


BIQU ST820

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also:This driver requires constant cooling the moment any motor is used, or it will switch on and off.

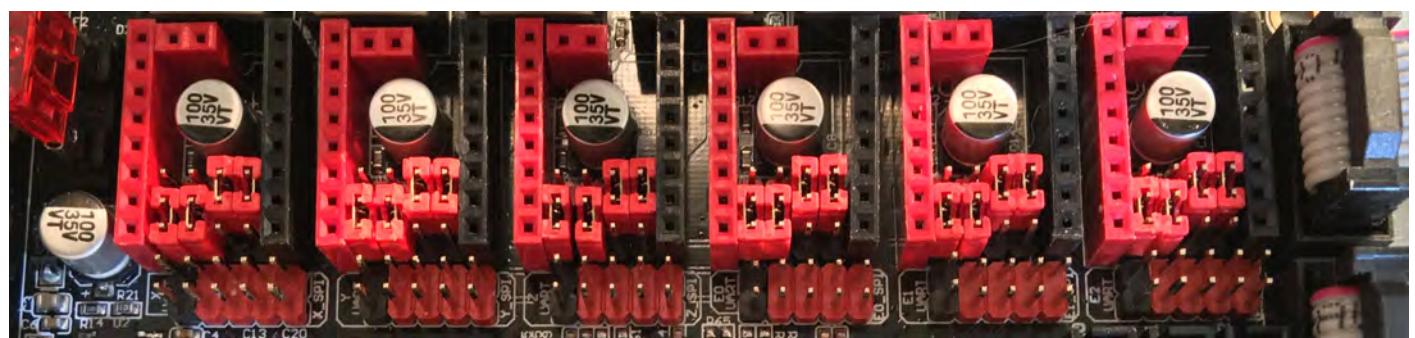
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	2	1A
STEP	-	D	3	VDD
DIR	-	D	3	GND

Note: The "D" jumper MUST be SET!



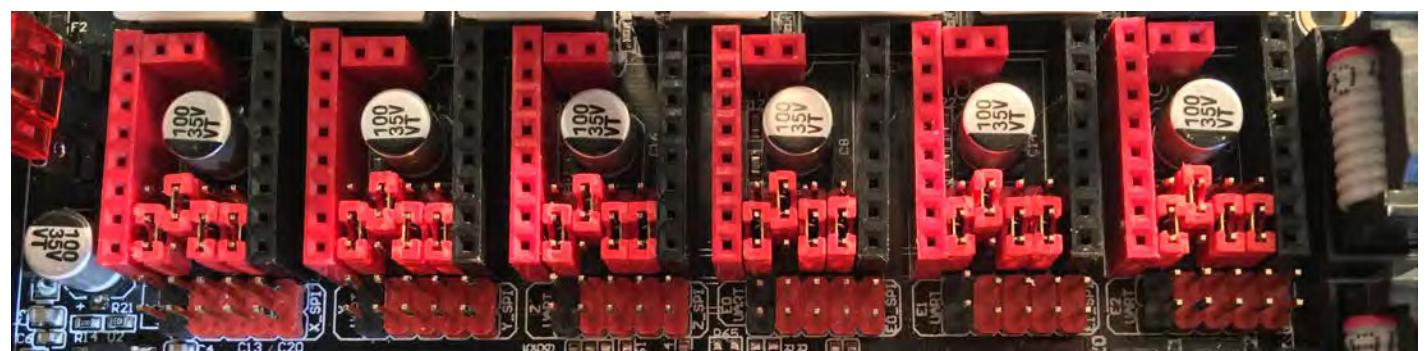
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	2	1A
STEP	-	D	3	VDD
DIR	-	D	3	GND

Note: The "D" jumper MUST be SET!



EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1B
MS3	-	-	-	2B
NC	-	-	-	2A
RST	-	-	3	1A
STEP	-	D	3	VDD
DIR	-	D	2	GND

Note: The "D" jumper MUST be SET!



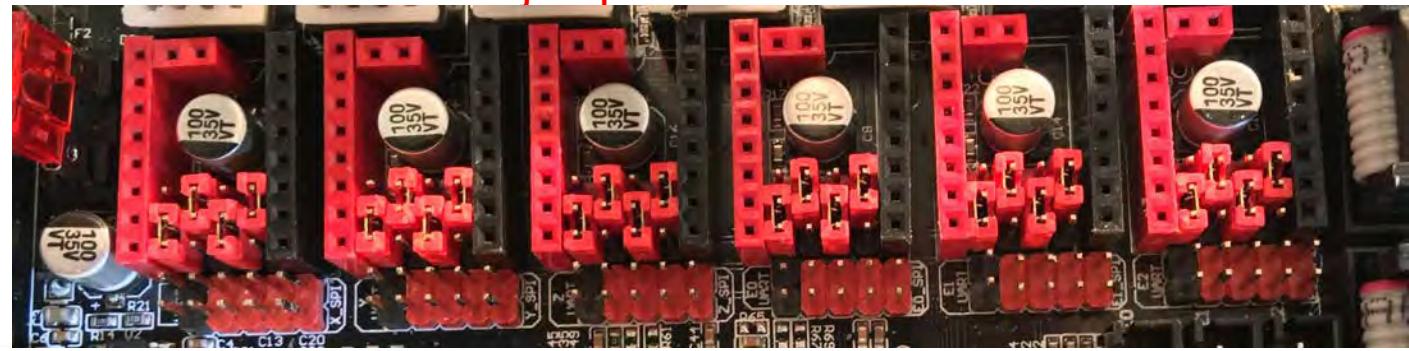
BIQU ST820

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also:This driver requires constant cooling the moment any motor is used, or it will switch on and off.

EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1B
MS3	-	-	-	-	2B
NC	-	-	-	-	2A
RST	-	-	3	1	1A
STEP	-	D	3	2	VDD
DIR	-	D	-	2	GND

1 / 32

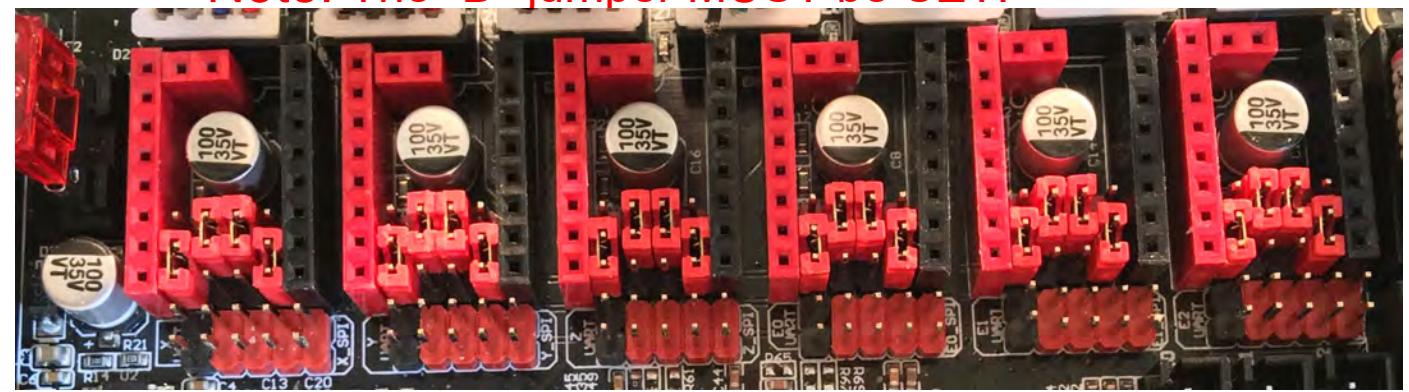
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1B
MS3	-	-	-	-	2B
NC	-	-	-	-	2A
RST	-	-	3	2	1A
STEP	-	D	3	2	VDD
DIR	-	D	-	1	GND

1 / 128

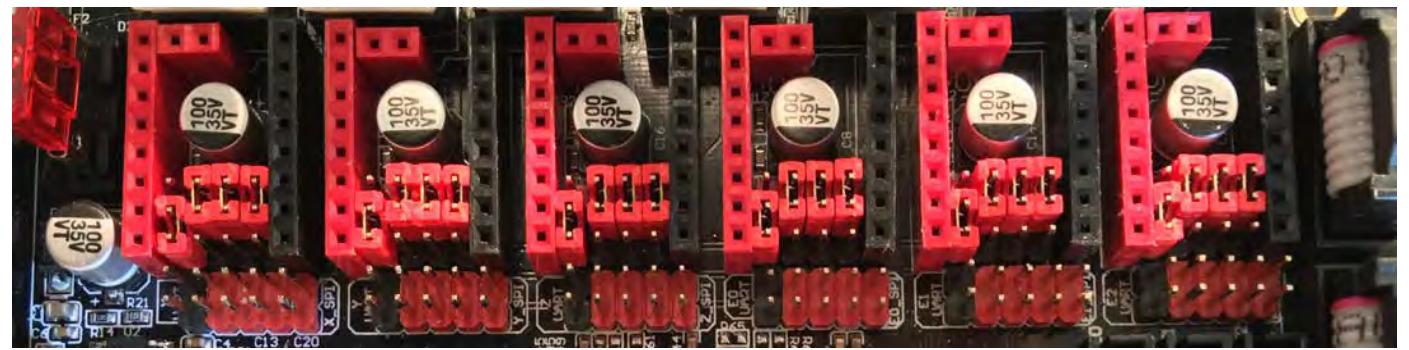
Note: The "D" jumper MUST be SET!



EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1B
MS3	-	-	-	-	2B
NC	-	-	-	-	2A
RST	-	-	3	2	1A
STEP	-	D	3	2	VDD
DIR	-	D	-	1	GND

1 / 256

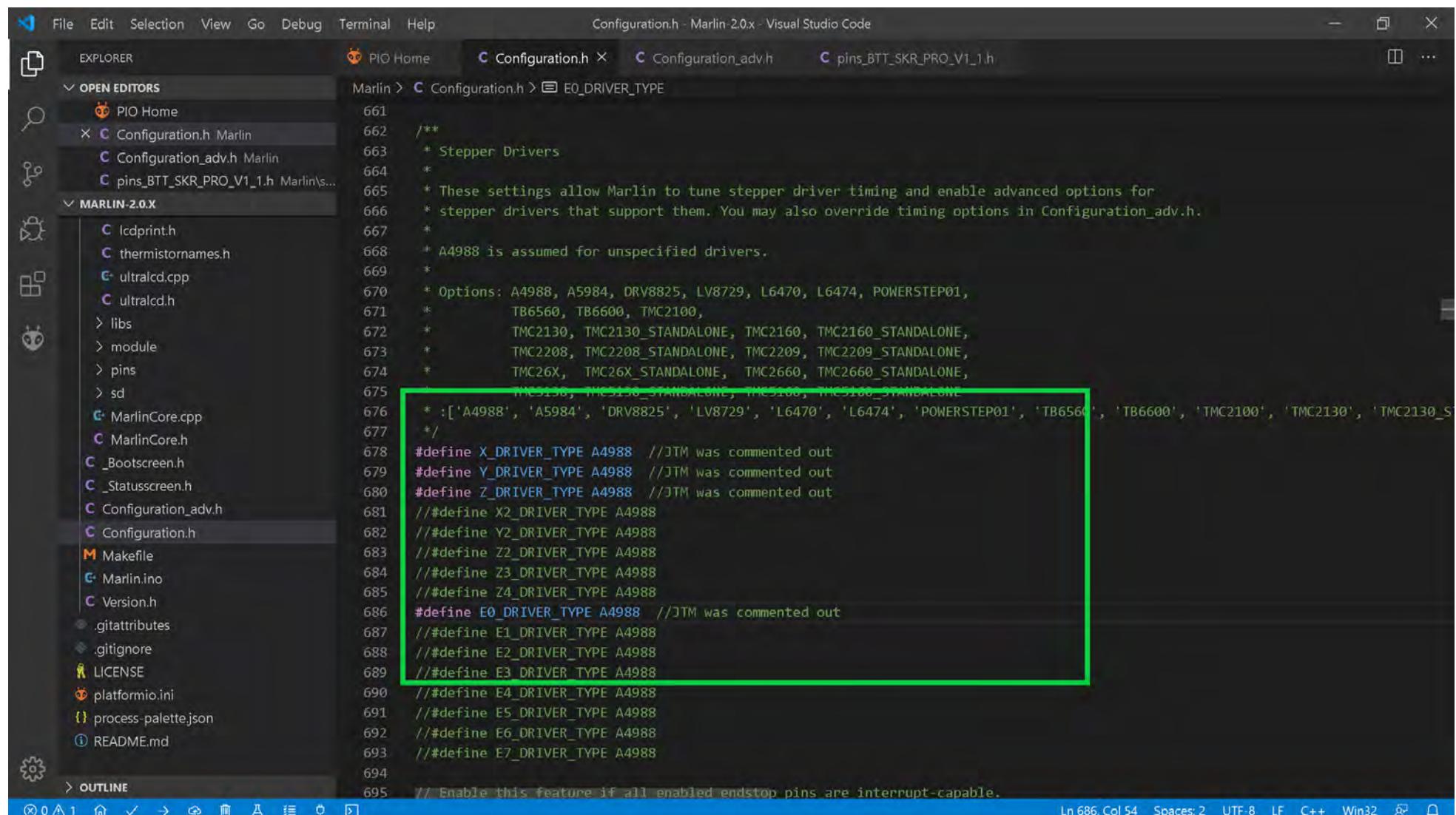
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for BIQU ST820 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU ST820 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using BIQU ST820 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU ST820 drivers. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The **ST820 is a drop in replacement for the A4988**. Since Marlin does not have an option for ST820 we will use the A4988 option.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):**
 - OPEN EDITORS: PIO Home, Configuration.h Marlin (highlighted), Configuration_adv.h Marlin, pins_BTT_SKR_PRO_V1_1.h Marlin\...
 - MARLIN-2.0.X: lcdprint.h, thermistornames.h, ultralcd.cpp, ultralcd.h, libs, module, pins, sd, MarlinCore.cpp, MarlinCore.h, _Bootscreen.h, _Statusscreen.h, Configuration_adv.h (highlighted), Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md.
- Code Editor:** The Configuration.h file is open. A green box highlights the driver configuration section starting at line 676. The code shows various driver definitions for X, Y, Z, E0, and E1-E7 axes, with the A4988 driver selected for all axes. The entire highlighted section is as follows:

```

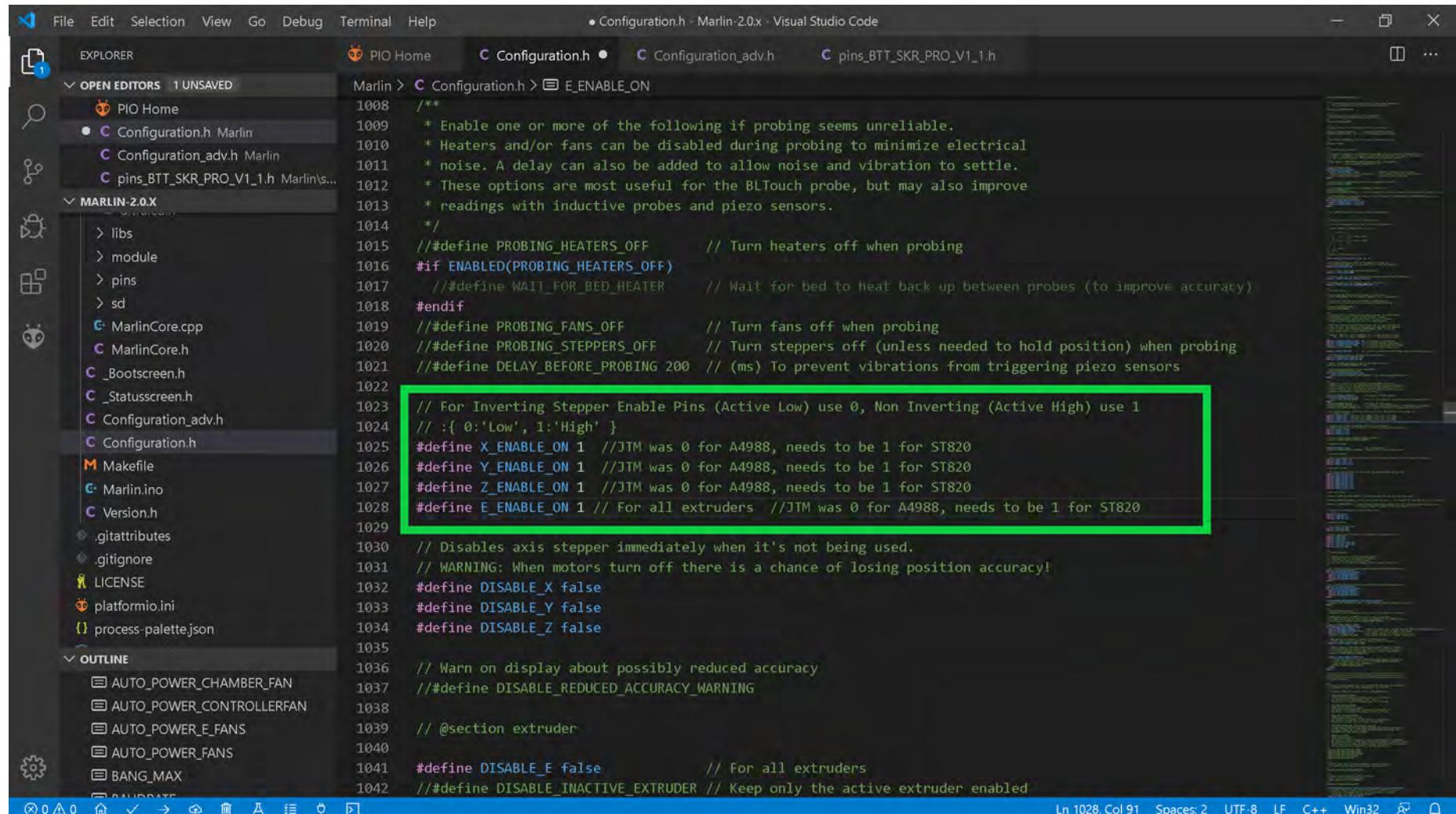
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01',
677 *      'TB6560', 'TB6600', 'TMC2100',
678 *      'TMC2130', 'TMC2130_STANDALONE', 'TMC2160', 'TMC2160_STANDALONE',
679 *      'TMC2208', 'TMC2208_STANDALONE', 'TMC2209', 'TMC2209_STANDALONE',
680 *      'TMC26X', 'TMC26X_STANDALONE', 'TMC2660', 'TMC2660_STANDALONE',
681 *      'TMC5130', 'TMC5130_STANDALONE', 'TMC5160', 'TMC5160_STANDALONE'
682 #define X_DRIVER_TYPE A4988 //JTM was commented out
683 #define Y_DRIVER_TYPE A4988 //JTM was commented out
684 #define Z_DRIVER_TYPE A4988 //JTM was commented out
685 #define Z2_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE A4988 //JTM was commented out
687 #define E1_DRIVER_TYPE A4988
688 #define E2_DRIVER_TYPE A4988
689 #define E3_DRIVER_TYPE A4988
690 #define E4_DRIVER_TYPE A4988
691 #define E5_DRIVER_TYPE A4988
692 #define E6_DRIVER_TYPE A4988
693 #define E7_DRIVER_TYPE A4988
694 // Enable this feature if all enabled endstop pins are interrupt-capable.

```
- Bottom Status Bar:** Ln 686, Col 54, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU ST820 Drivers

- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the Marlin repository open. The left sidebar shows the project structure under 'EXPLORER' with files like Configuration.h, Configuration_adv.h, and pins_BTT_SKR_PRO_V1_1.h. The main editor area displays the Configuration.h file. A green box highlights the following code block:

```

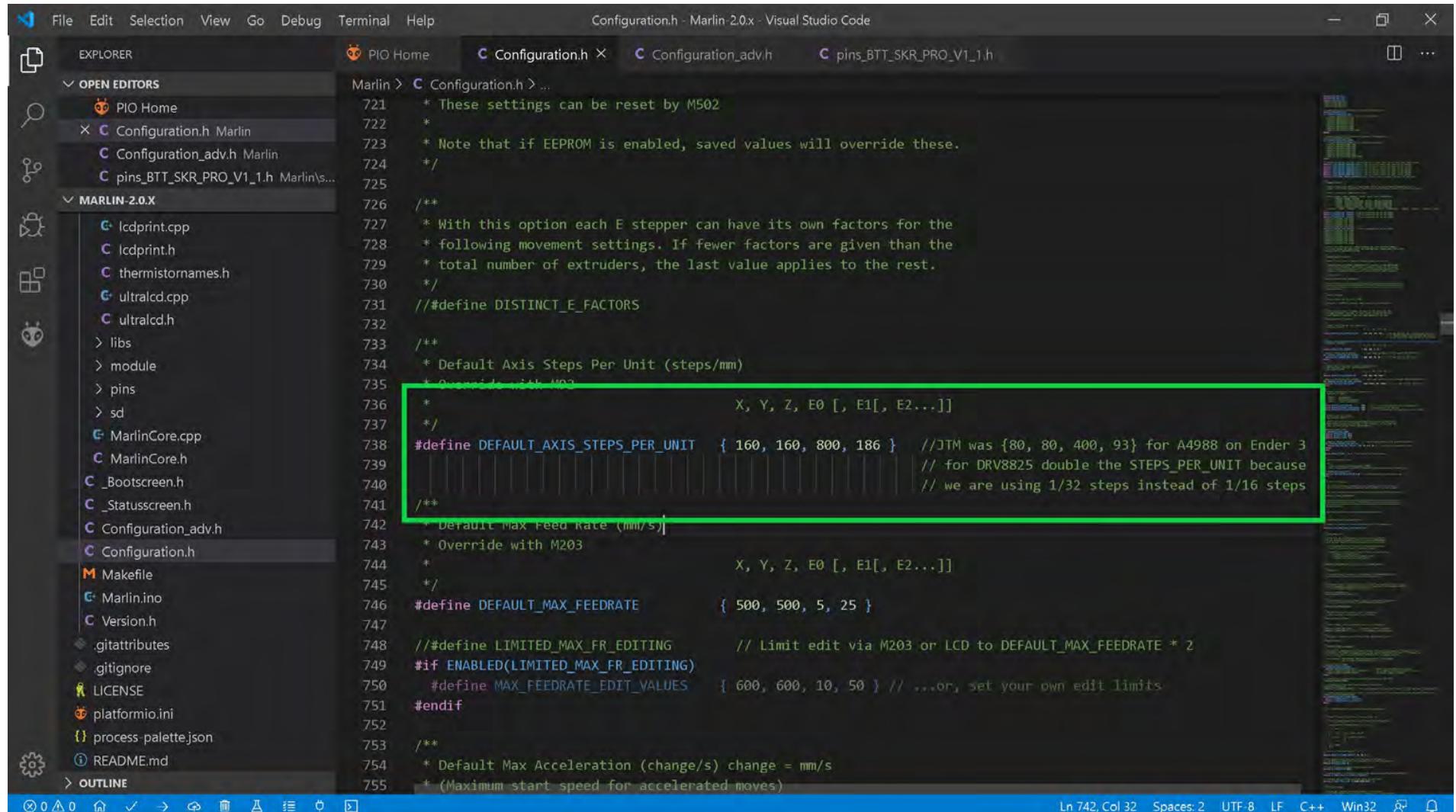
1008 /**
1009 * Enable one or more of the following if probing seems unreliable.
1010 * Heaters and/or fans can be disabled during probing to minimize electrical
1011 * noise. A delay can also be added to allow noise and vibration to settle.
1012 * These options are most useful for the BLTouch probe, but may also improve
1013 * readings with inductive probes and piezo sensors.
1014 */
1015 //#define PROBING_HEATERS_OFF // Turn heaters off when probing
1016 #if ENABLED(PROBING_HEATERS_OFF)
1017     //#define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1018 #endif
1019 //#define PROBING_FANS_OFF // Turn fans off when probing
1020 //#define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1021 //#define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1022
1023 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1024 // :{ 0:'Low', 1:'High' }
1025 #define X_ENABLE_ON 1 // JTM was 0 for A4988, needs to be 1 for ST820
1026 #define Y_ENABLE_ON 1 // JTM was 0 for A4988, needs to be 1 for ST820
1027 #define Z_ENABLE_ON 1 // JTM was 0 for A4988, needs to be 1 for ST820
1028 #define E_ENABLE_ON 1 // For all extruders // JTM was 0 for A4988, needs to be 1 for ST820
1029
1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 //#define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false // For all extruders
1042 //#define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU ST820 Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to BIQU ST820 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```

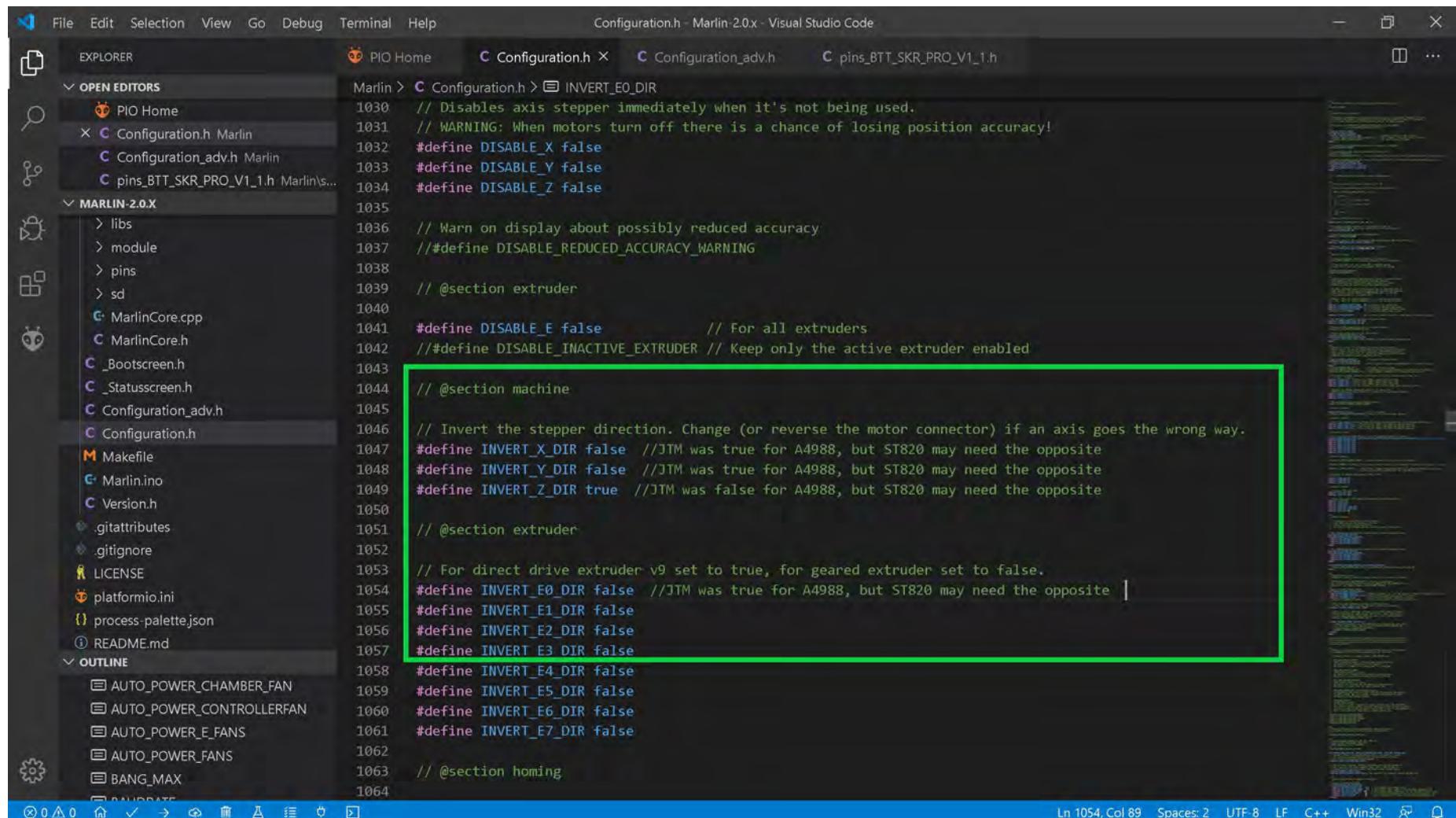
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h ...
PIO Home Configuration.h Marlin Configuration_adv.h Marlin pins_BTT_SKR_PRO_V1_1.h Marlin...
MARLIN-2.0.X
Lcdprint.cpp
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
721 * These settings can be reset by M502
722 *
723 * Note that if EEPROM is enabled, saved values will override these.
724 */
725 /**
726 * With this option each E stepper can have its own factors for the
727 * following movement settings. If fewer factors are given than the
728 * total number of extruders, the last value applies to the rest.
729 */
730 // #define DISTINCT_E_FACTORS
731 /**
732 * Default Axis Steps Per Unit (steps/mm)
733 */
734 * X, Y, Z, E0 [, E1[, E2...]]
735 */
736 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // JTM was {80, 80, 400, 93} for A4988 on Ender 3
737 // for DRV8825 double the STEPS_PER_UNIT because
738 // we are using 1/32 steps instead of 1/16 steps
739 /**
740 * Default Max Feed Rate (mm/s)
741 */
742 * Override with M203
743 * X, Y, Z, E0 [, E1[, E2...]]
744 */
745 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
746 /**
747 * #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
748 */
749 #if ENABLED(LIMITED_MAX_FR_EDITING)
750 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
751 #endif
752 /**
753 * Default Max Acceleration (change/s) change = mm/s
754 */
755 * (Maximum start speed for accelerated moves)
Ln 742, Col 32 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU ST820 Drivers

- **Optional Step:** I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide, the below information, in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") https://reprap.org/wiki/Stepper_wiring. Other people prefer to change the motor direction in the Marlin firmware. **So if you want or need to change the motor direction in Marlin**, then if the axis' setting you will be using the ST820 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet of C++ code:

```

Marlin > C Configuration.h > INVERT_E0_DIR
1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 //#define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false           // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false // JTM was true for A4988, but ST820 may need the opposite
1048 #define INVERT_Y_DIR false // JTM was true for A4988, but ST820 may need the opposite
1049 #define INVERT_Z_DIR true  // JTM was false for A4988, but ST820 may need the opposite
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false // JTM was true for A4988, but ST820 may need the opposite
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

```

A green rectangular box highlights the entire code block from line 1047 to line 1064, which contains the definitions for `INVERT_X_DIR` through `INVERT_E7_DIR`.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU ST820 Drivers

- The end of Marlin setup for BIQU ST820 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

The screenshot shows the Visual Studio Code interface with the Marlin repository open. The left sidebar shows the file structure, including files like `pins_BTT_SKR_PRO_V1_1.h`, `Configuration.h`, and `MarlinCore.cpp`. The main editor area displays code for stepper driver timing options. The bottom right terminal window shows the build process:

```

STM32F407VC-D3.c: IGNORED
BIGTREE_SKR_PRO: SUCCESS 00:02:31.294
BIGTREE_STP_13_0: IGNORED
BIGTREE_BT002: IGNORED
teensy31: IGNORED
teensy35: IGNORED
esp32: IGNORED
linux_native: IGNORED
SAMD51_grandcentral_m4: IGNORED
rumba32_f446ve: IGNORED
mks_rumba32: IGNORED
include_tree: IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

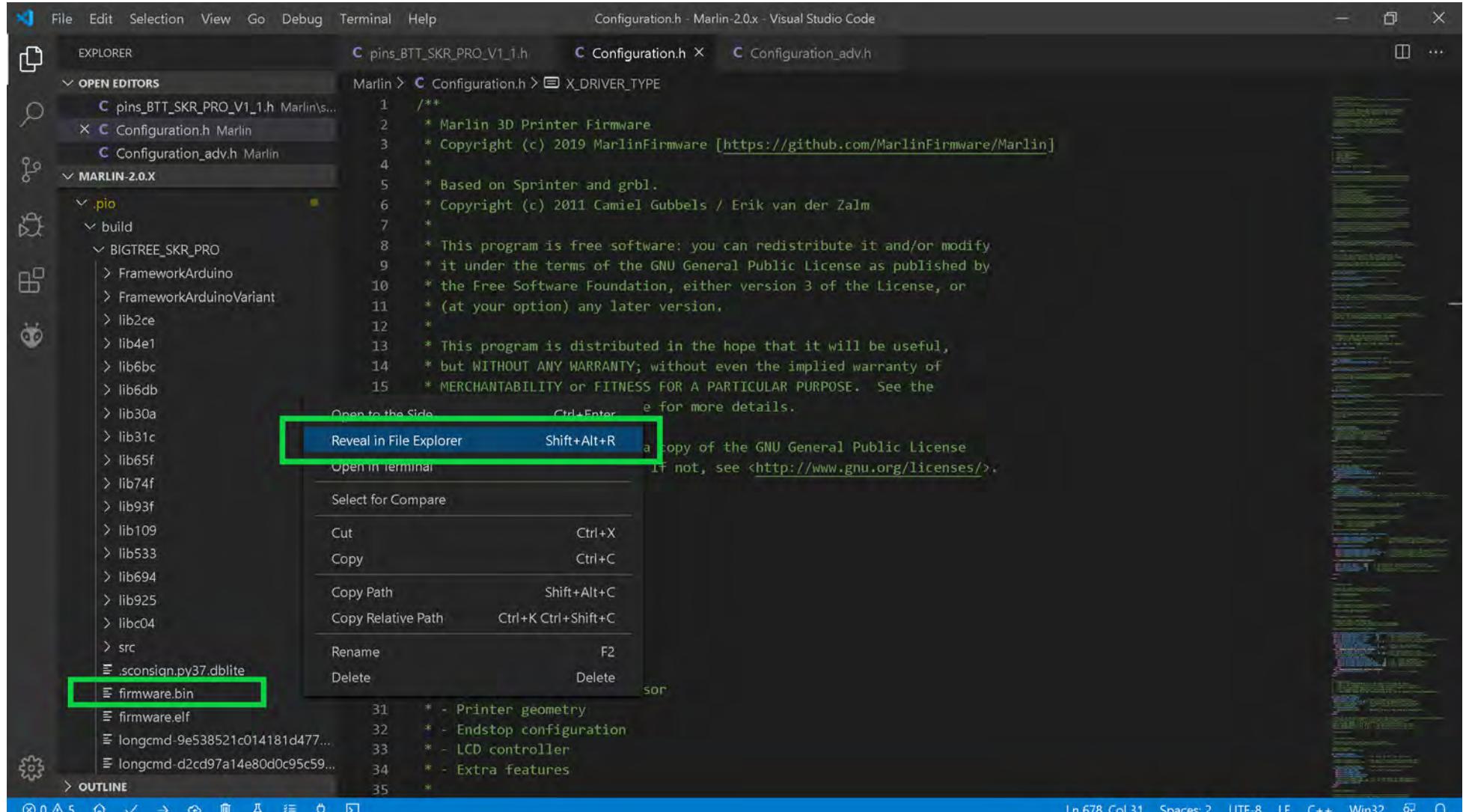
```

A yellow box highlights the checkmark icon in the terminal toolbar, and a green box highlights the successful compilation message in the terminal output.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU ST820 Drivers

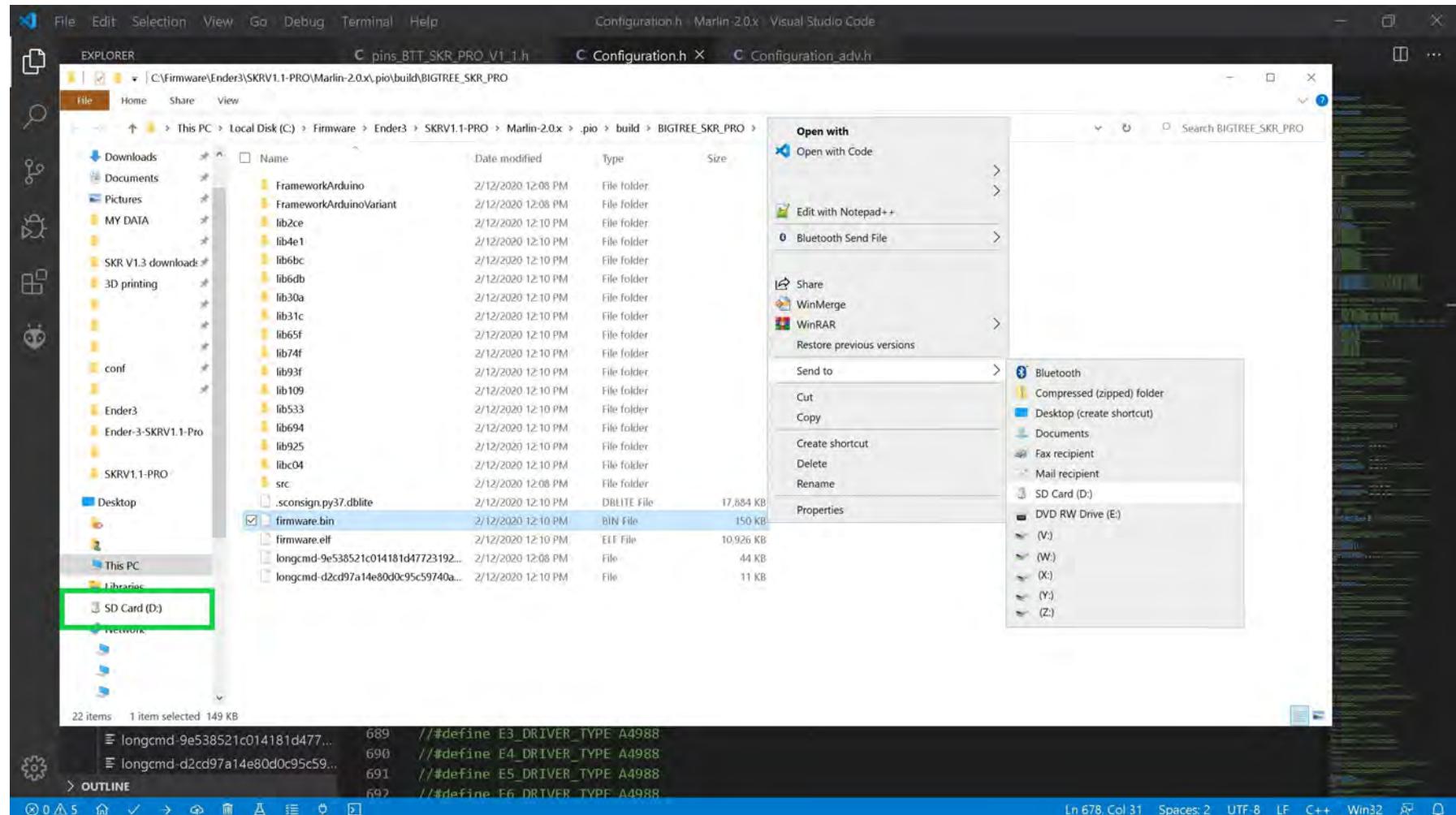
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



- Go to the next page.

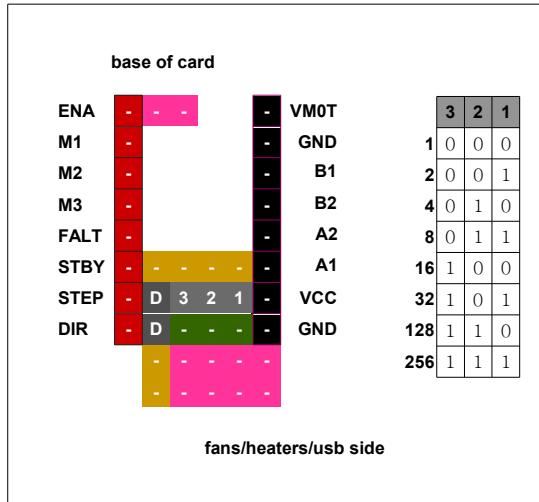
The (latest release of) Marlin Setup for BIQU ST820 Drivers

- Ensure your micro-SD card, that came with your SKR PRO V1.1 board, is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

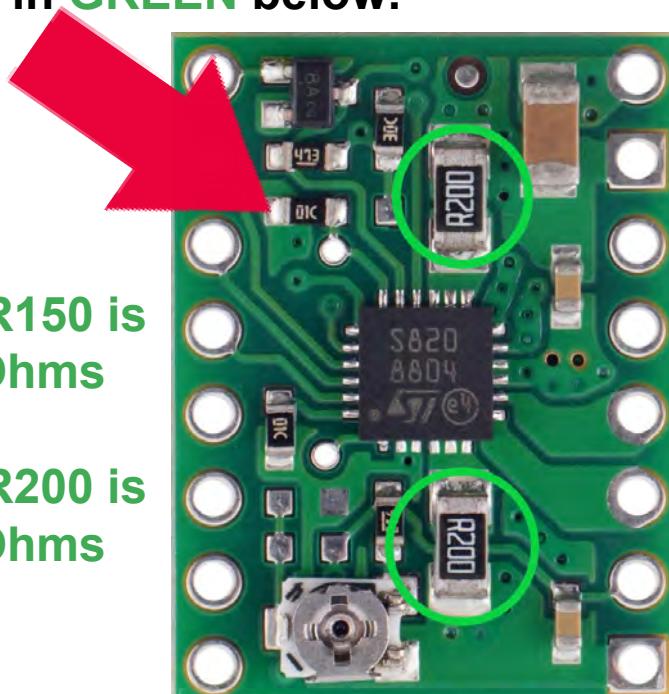
POLOLU ST820 (STSPIN820)



Driver Chip	MS3	MS2	MS1	Subdivision
Pololu ST820 Maximum 256 subdivision 45V 0.9 A	L	L	L	Full Step
	L	L	H	1/2
	L	H	L	1/4
	L	H	H	1/8
	H	L	L	1/16
	H	L	H	1/32
	H	H	L	1/128
	H	H	H	1/256
Driving current calculation formula $R_s = 0.2\Omega$	$I_{MAX} = V_{REF} * 5$		$V_{ref} = I_{MAX} / 5$	
	Note: Use 90% of the calculated Vref when tuning the stepper driver board			

Note: Check your current sense resistor (Rs) values on the driver board, shown in GREEN below.

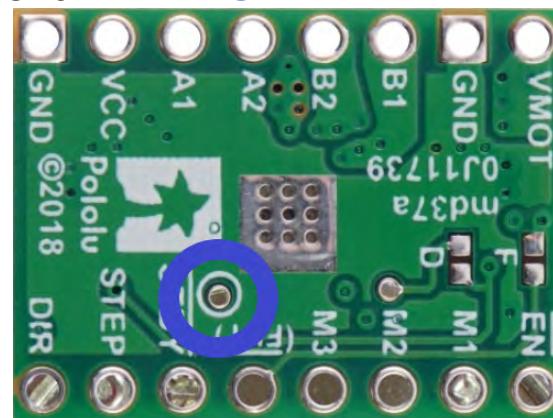
NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.



$Rs = R150$ is 0.15 Ohms

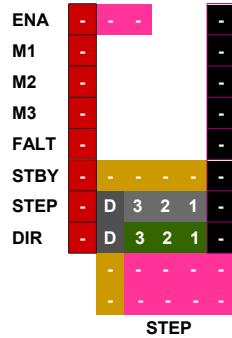
$Rs = R200$ is 0.20 Ohms

Note: "Vref Test point" location is on the bottom of the driver board, as shown in BLUE

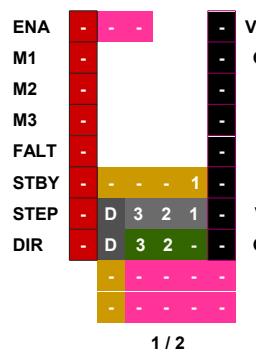
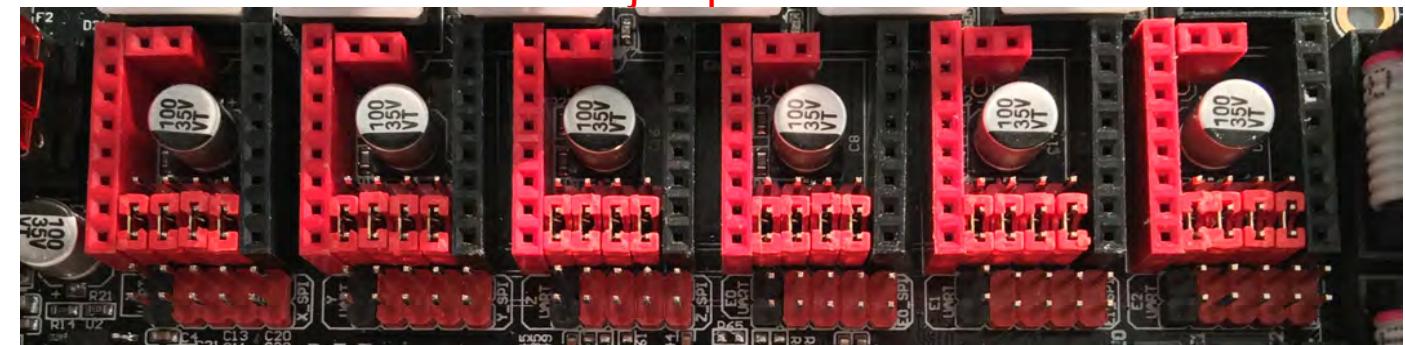


POLOLU ST820 (STSPIN820)

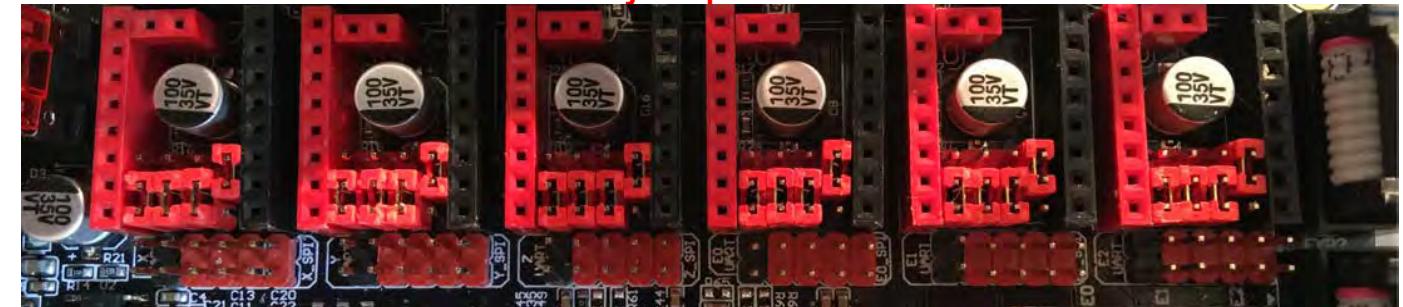
Important: This driver has special requirements in the Configuration and Configuration_adv.h Also: This driver requires constant cooling the moment any motor is used, or it will switch on and off.



Note: The "D" jumper MUST be SET!



Note: The "D" jumper MUST be SET!

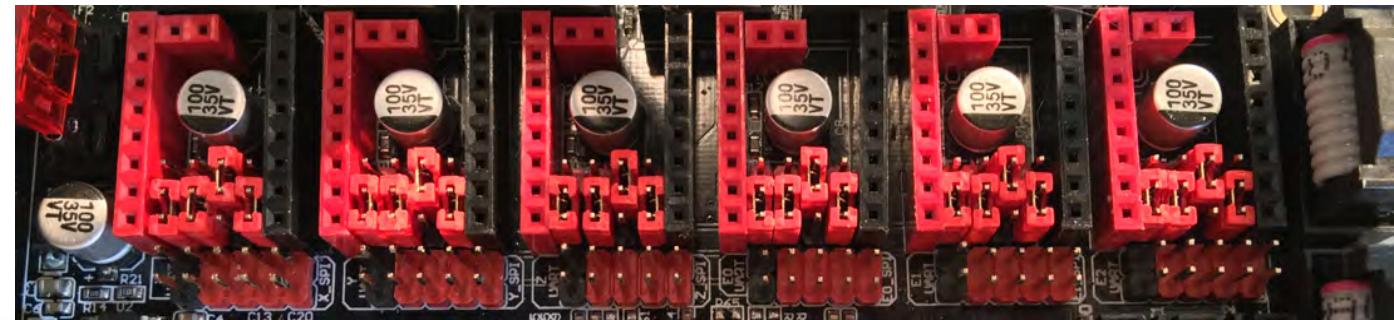


POLOLU ST820 (STSPIN820)

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also:This driver requires constant cooling the moment any motor is used, or it will switch on and off.

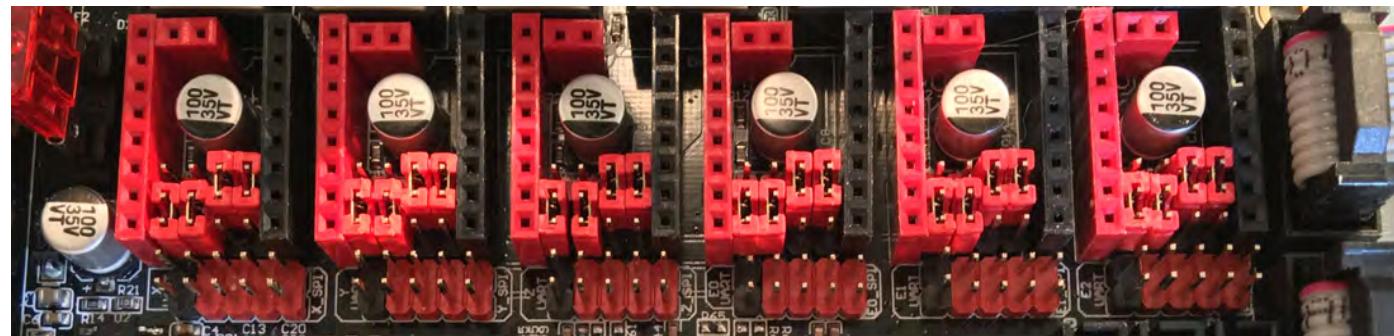
ENA	[]	[]	-	VMOT
M1	-	-	-	GND
M2	-	-	-	B1
M3	-	-	-	B2
FALT	-	-	-	A2
STBY	-	-	2	-
STEP	-	D	3	2
DIR	-	D	3	-
			1	1
			4	

Note: The "D" jumper MUST be SET!



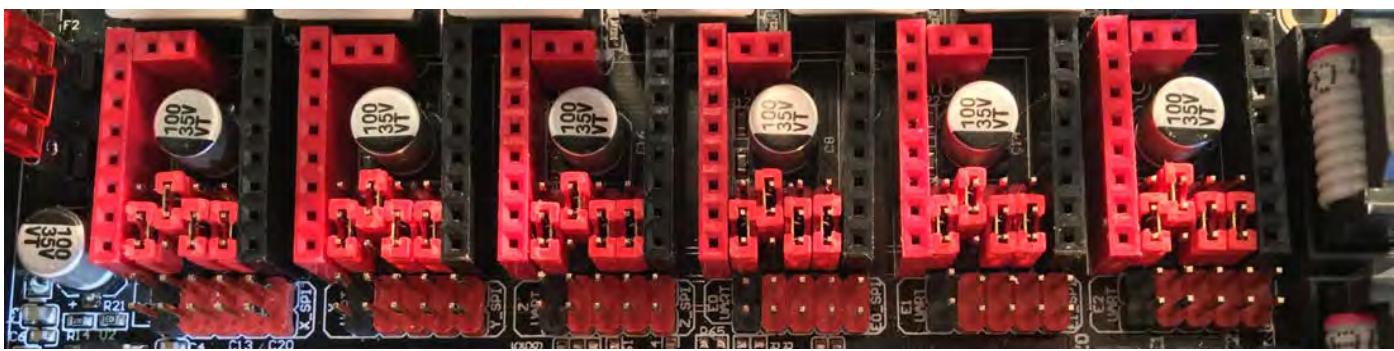
ENA	[]	[]	-	VMOT
M1	-	-	-	GND
M2	-	-	-	B1
M3	-	-	-	B2
FALT	-	-	-	A2
STBY	-	-	2	-
STEP	-	D	3	2
DIR	-	D	3	-
			1	1
			8	

Note: The "D" jumper MUST be SET!



ENA	[]	[]	-	VMOT
M1	-	-	-	GND
M2	-	-	-	B1
M3	-	-	-	B2
FALT	-	-	-	A2
STBY	-	-	3	-
STEP	-	D	3	2
DIR	-	D	2	1
			1	1
			16	

Note: The "D" jumper MUST be SET!



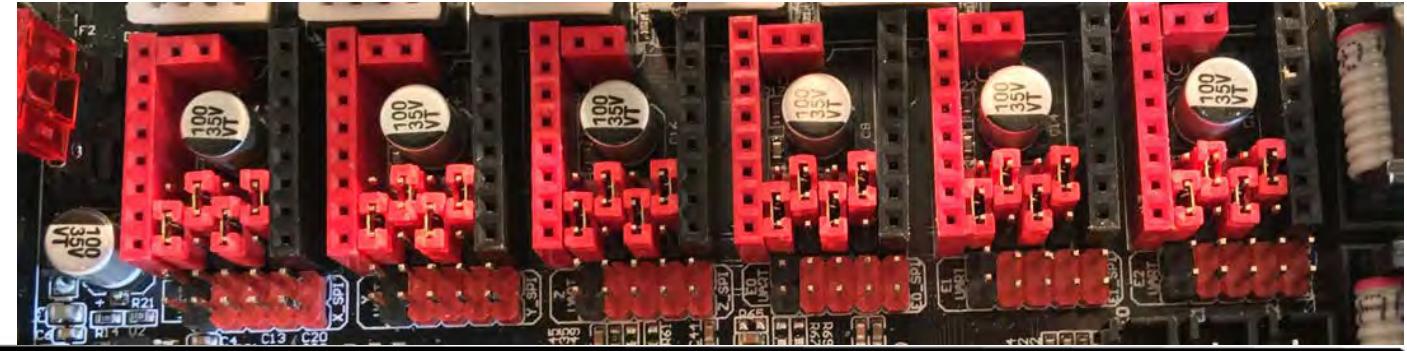
POLOLU ST820 (STSPIN820)

Important: This driver has special requirements in the Configuration and Configuration_adv.h Also:This driver requires constant cooling the moment any motor is used, or it will switch on and off.

ENA	-	-	-	-	VMOT
M1	-	-	-	-	GND
M2	-	-	-	-	B1
M3	-	-	-	-	B2
FALT	-	-	-	-	A2
STBY	-	3	1	-	A1
STEP	D	3	2	1	VCC
DIR	D	-	2	-	GND

1 / 32

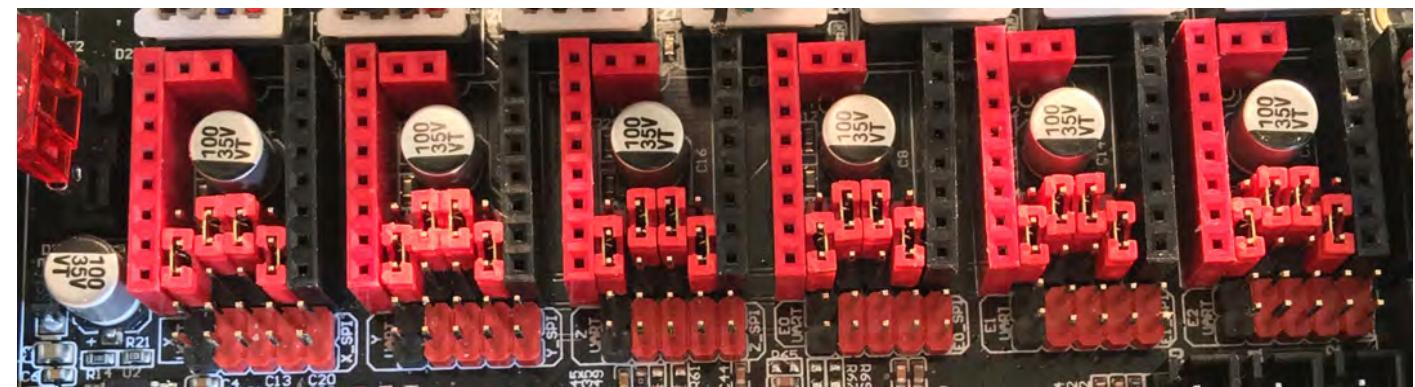
Note: The "D" jumper MUST be SET!



ENA	-	-	-	-	VMOT
M1	-	-	-	-	GND
M2	-	-	-	-	B1
M3	-	-	-	-	B2
FALT	-	-	-	-	A2
STBY	-	3	2	-	A1
STEP	D	3	2	1	VCC
DIR	D	-	1	-	GND

1 / 128

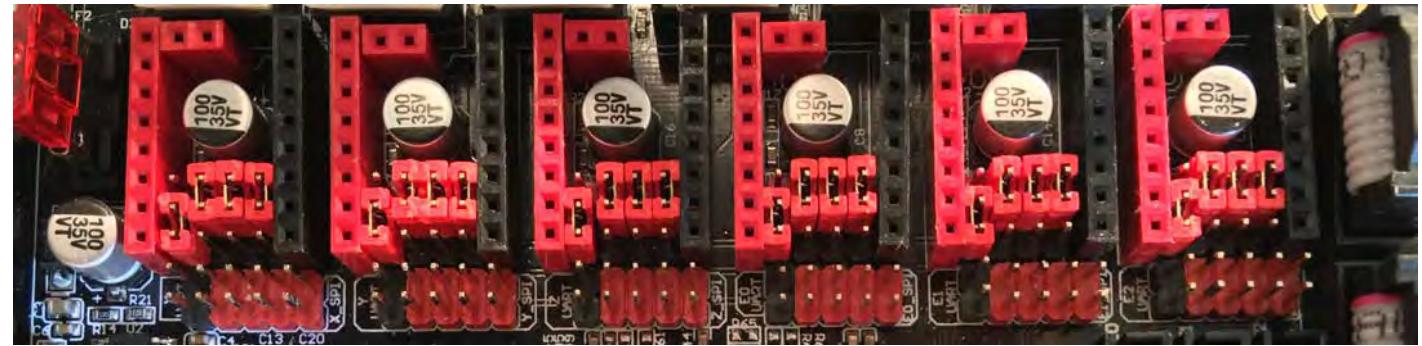
Note: The "D" jumper MUST be SET!



ENA	-	-	-	-	VMOT
M1	-	-	-	-	GND
M2	-	-	-	-	B1
M3	-	-	-	-	B2
FALT	-	-	-	-	A2
STBY	-	3	2	1	A1
STEP	D	3	2	1	VCC
DIR	D	-	-	-	GND

1 / 256

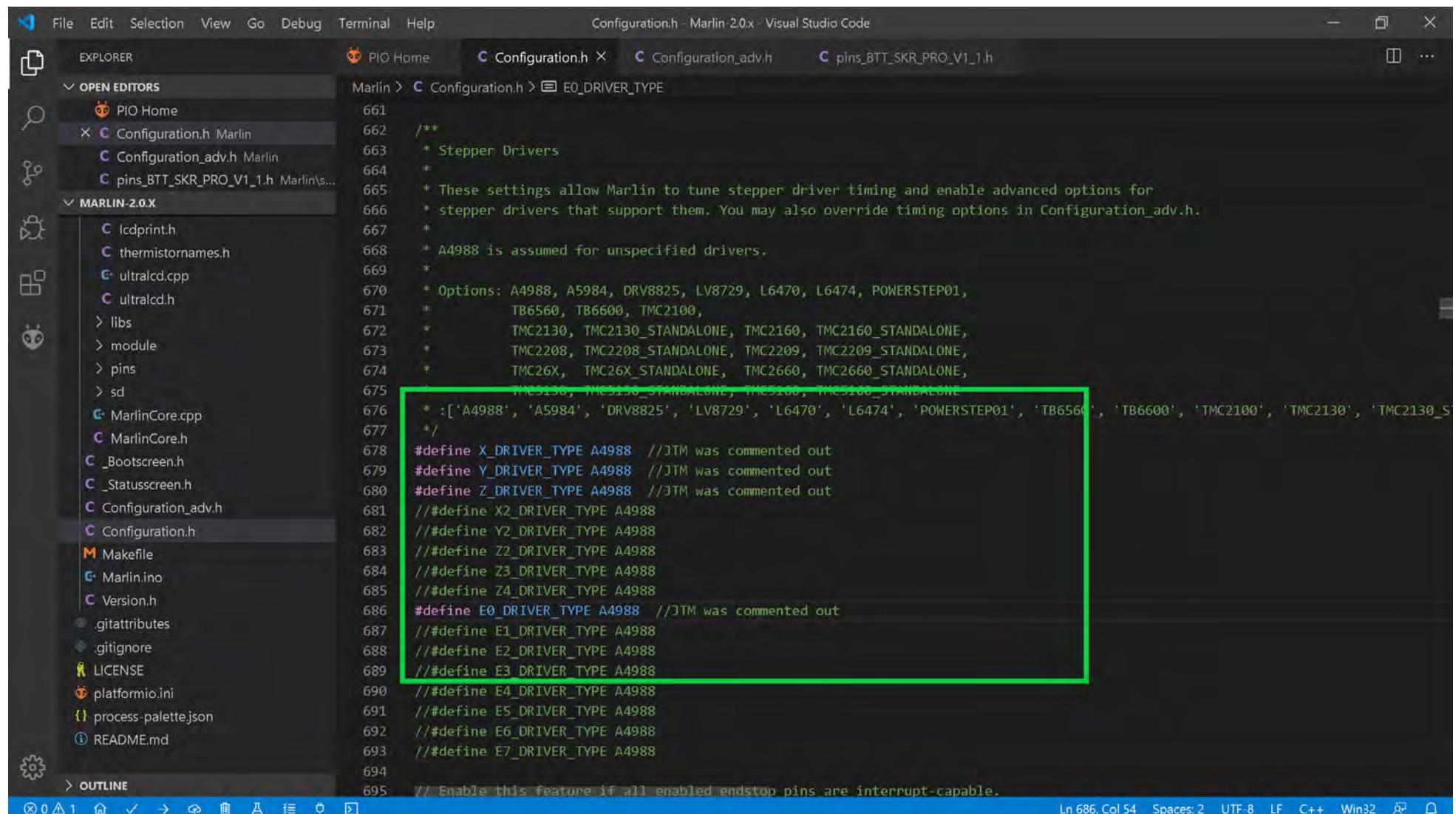
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for POLOLU ST820 (STSPIN820) stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using POLOLU ST820 (STSPIN820) drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU ST820 (STSPIN820) drivers. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The ST820 is a drop in replacement for the A4988. Since Marlin does not have an option for ST820 we will use the A4988 option.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following relevant section:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 *           TB6560, TB6600, TMC2100,
671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 *           TMC5168, TMC5168_STANDALONE, TMC5186, TMC5186_STANDALONE
675 *
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
677 */
678 #define X_DRIVER_TYPE A4988 //JTM was commented out
679 #define Y_DRIVER_TYPE A4988 //JTM was commented out
680 #define Z_DRIVER_TYPE A4988 //JTM was commented out
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE A4988 //JTM was commented out
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

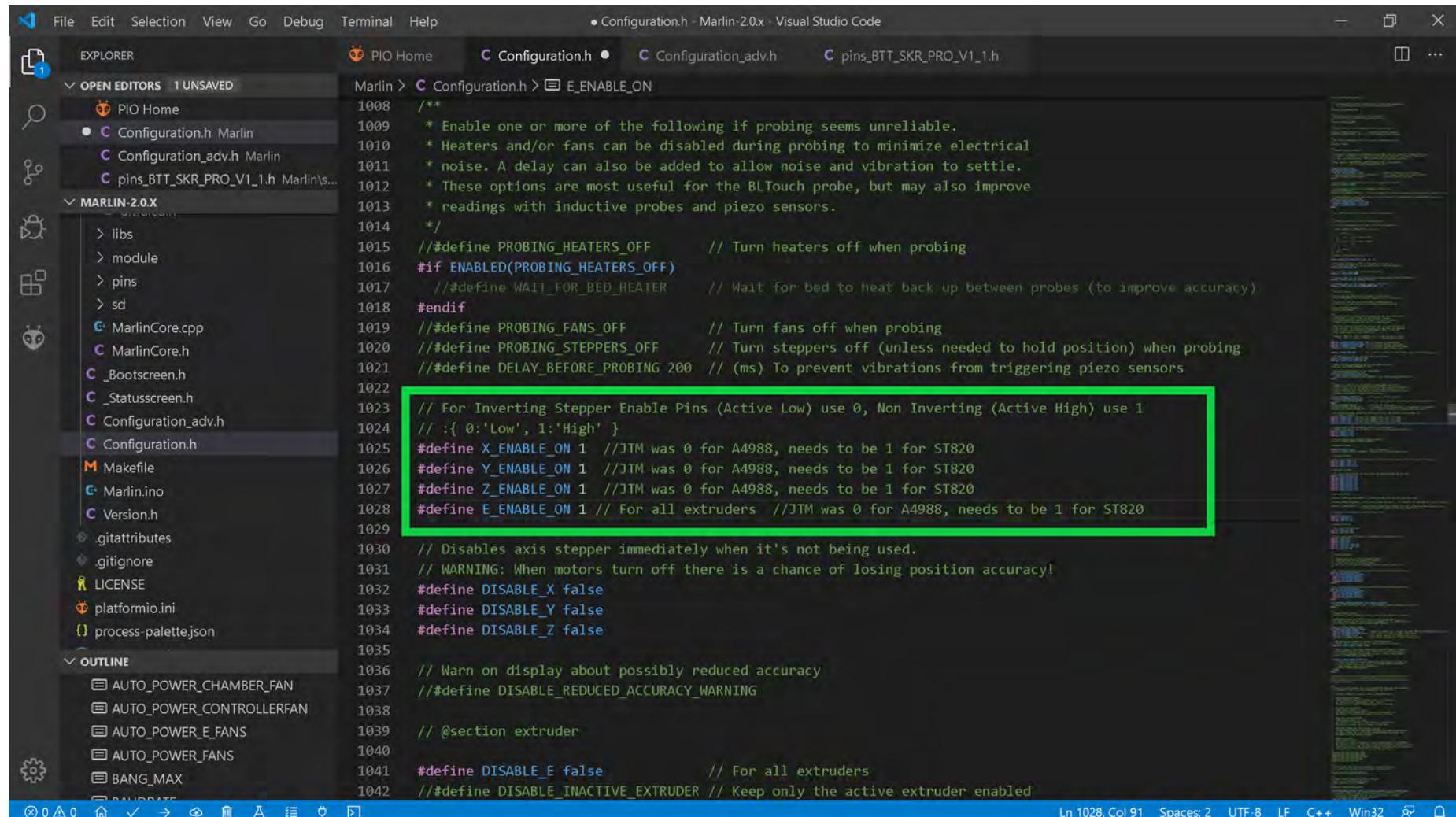
```

A green rectangular box highlights the driver selection lines (lines 678-693), specifically the `#define` statements for X, Y, Z, and E0 axes. These lines originally specified ST820 (e.g., `ST820_DRIVER_TYPE`) but have been replaced with A4988 (e.g., `A4988`). The original comments (//JTM was commented out) remain in place.

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a dark theme. On the left, the 'EXPLORER' sidebar shows various files and folders related to the Marlin firmware. The main code area displays the 'Configuration.h' file, specifically the section for defining enable pins. A green rectangular box highlights the following code block:

```

1024 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1025 // :{ 0:'Low', 1:'High' }
1026 #define X_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1027 #define Y_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1028 #define Z_ENABLE_ON 1 //JTM was 0 for A4988, needs to be 1 for ST820
1029 #define E_ENABLE_ON 1 // For all extruders //JTM was 0 for A4988, needs to be 1 for ST820

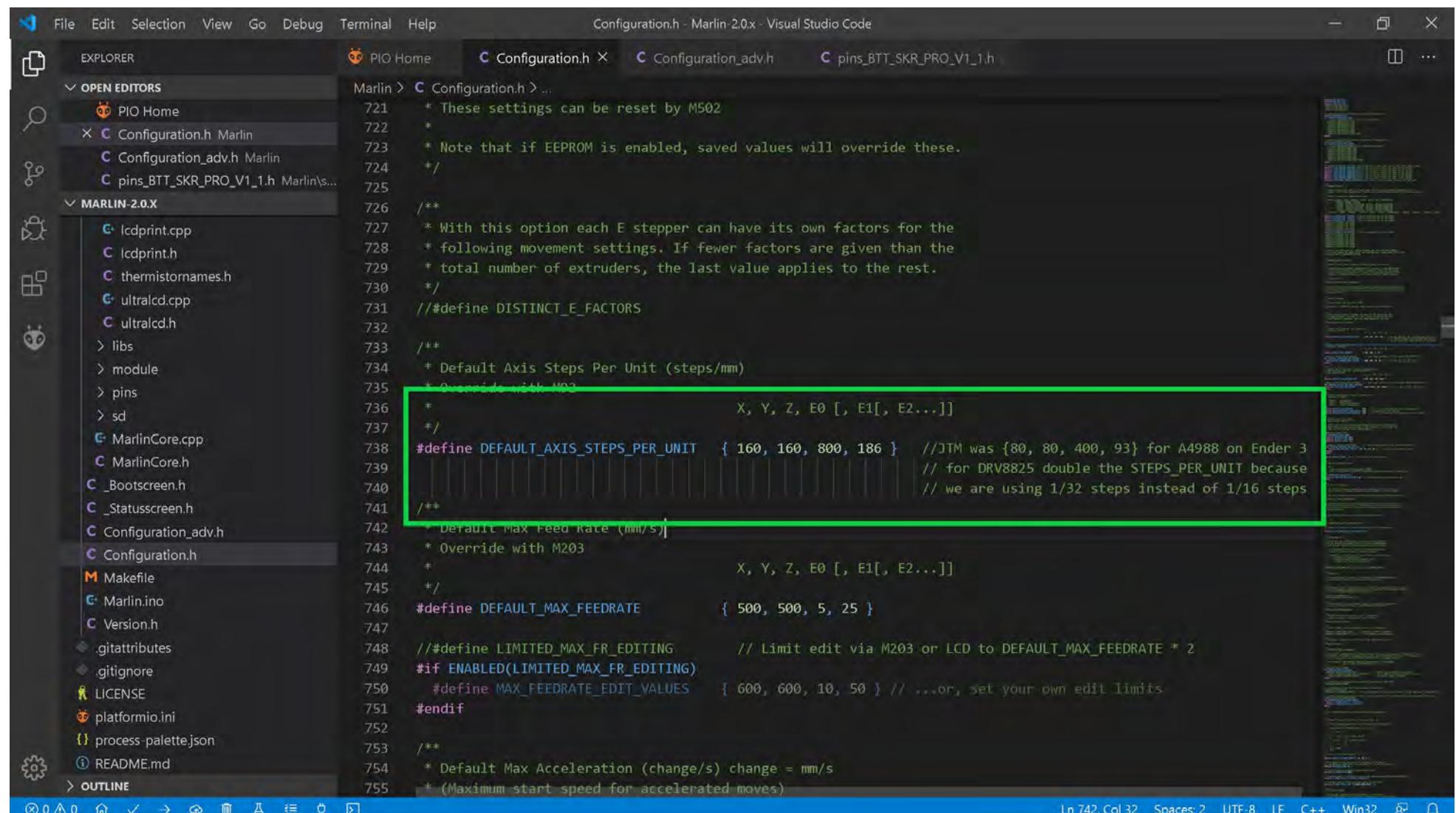
```

This code sets the enable pins for the X, Y, Z, and E axes to active high (1). The rest of the file contains comments and other configuration parameters for the Marlin firmware.

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to POLOLU ST820 (STSPIN820) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```

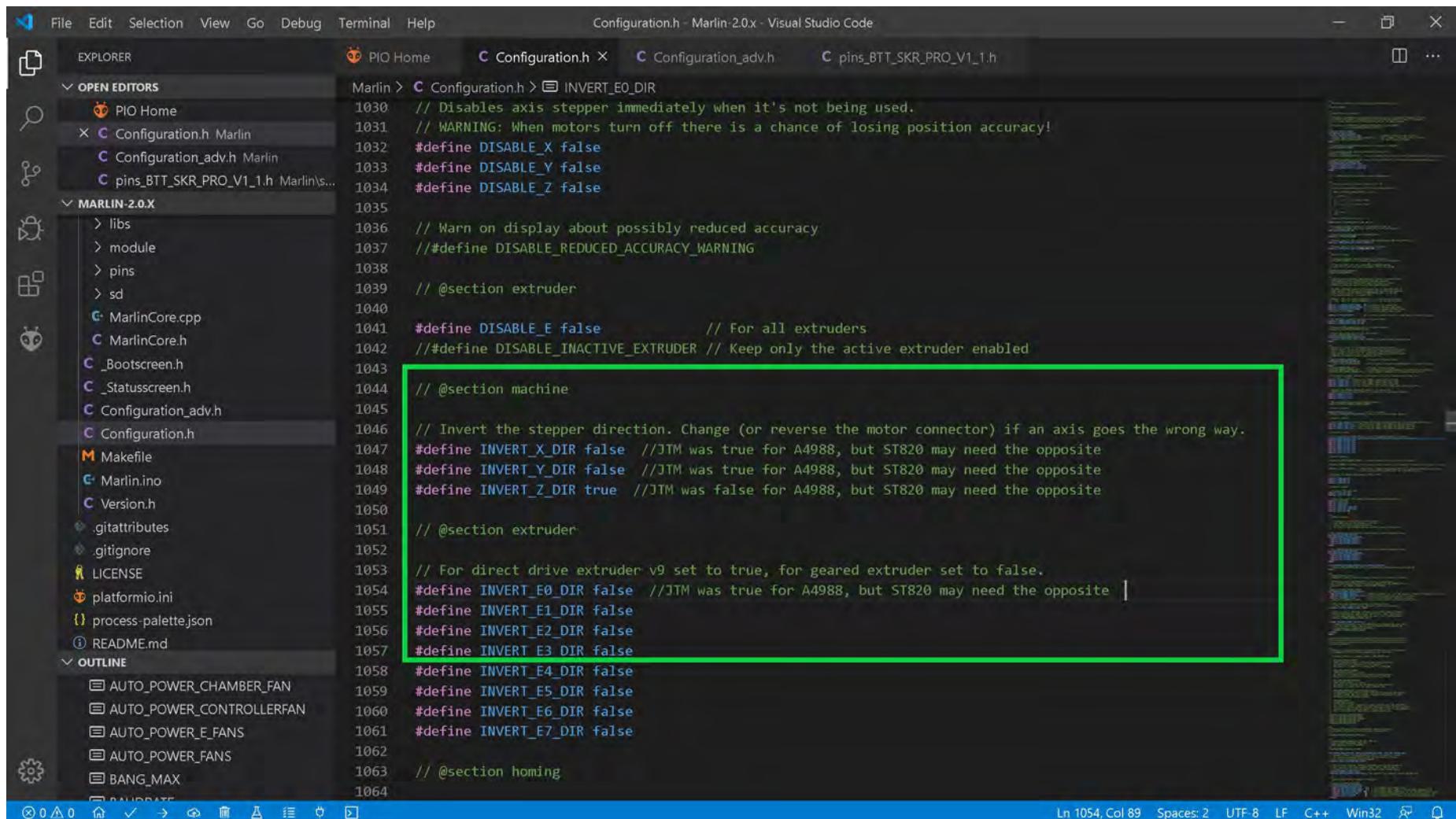
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h ...
    721 * These settings can be reset by M502
    722 *
    723 * Note that if EEPROM is enabled, saved values will override these.
    724 */
    725 /**
    726 * With this option each E stepper can have its own factors for the
    727 * following movement settings. If fewer factors are given than the
    728 * total number of extruders, the last value applies to the rest.
    729 */
    730 //#define DISTINCT_E_FACTORS
    731 /**
    732 * Default Axis Steps Per Unit (steps/mm)
    733 * Override with M203
    734 * X, Y, Z, E0 [, E1[, E2...]]
    735 */
    736 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
    737 // for DRV8825 double the STEPS_PER_UNIT because
    738 // we are using 1/32 steps instead of 1/16 steps
    739 /**
    740 * Default Max Feed Rate (mm/s)
    741 * Override with M203
    742 * X, Y, Z, E0 [, E1[, E2...]]
    743 */
    744 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
    745 /**
    746 #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
    747 #if ENABLED(LIMITED_MAX_FR_EDITING)
    748 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
    749 #endif
    750 /**
    751 * Default Max Acceleration (change/s) change = mm/s
    752 */
    753 /**
    754 * Default Max Start Speed (mm/s) change = mm/s
    755 */
Ln 742, Col 32 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- **Optional Step:** I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") https://reprap.org/wiki/Stepper_wiring. Other people prefer to change the motor direction in the Marlin firmware. **So if you want or need to change the motor direction in Marlin**, then if the axis' setting you will be using the ST820 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration.h > INVERT_E0_DIR
1030 // Disables axis stepper immediately when it's not being used.
1031 // WARNING: When motors turn off there is a chance of losing position accuracy!
1032 #define DISABLE_X false
1033 #define DISABLE_Y false
1034 #define DISABLE_Z false
1035
1036 // Warn on display about possibly reduced accuracy
1037 //#define DISABLE_REDUCED_ACCURACY_WARNING
1038
1039 // @section extruder
1040
1041 #define DISABLE_E false          // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 #define INVERT_X_DIR false // JTM was true for A4988, but ST820 may need the opposite
1048 #define INVERT_Y_DIR false // JTM was true for A4988, but ST820 may need the opposite
1049 #define INVERT_Z_DIR true // JTM was false for A4988, but ST820 may need the opposite
1050
1051 // @section extruder
1052
1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
1054 #define INVERT_E0_DIR false // JTM was true for A4988, but ST820 may need the opposite
1055 #define INVERT_E1_DIR false
1056 #define INVERT_E2_DIR false
1057 #define INVERT_E3_DIR false
1058 #define INVERT_E4_DIR false
1059 #define INVERT_E5_DIR false
1060 #define INVERT_E6_DIR false
1061 #define INVERT_E7_DIR false
1062
1063 // @section homing
1064

```

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- The end of Marlin setup for POLOLU ST820 (STSPIN820) drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editor:** Displays the content of Configuration.h, specifically the X_DRIVER_TYPE section.
- Terminal:** Shows the build log with the following output:

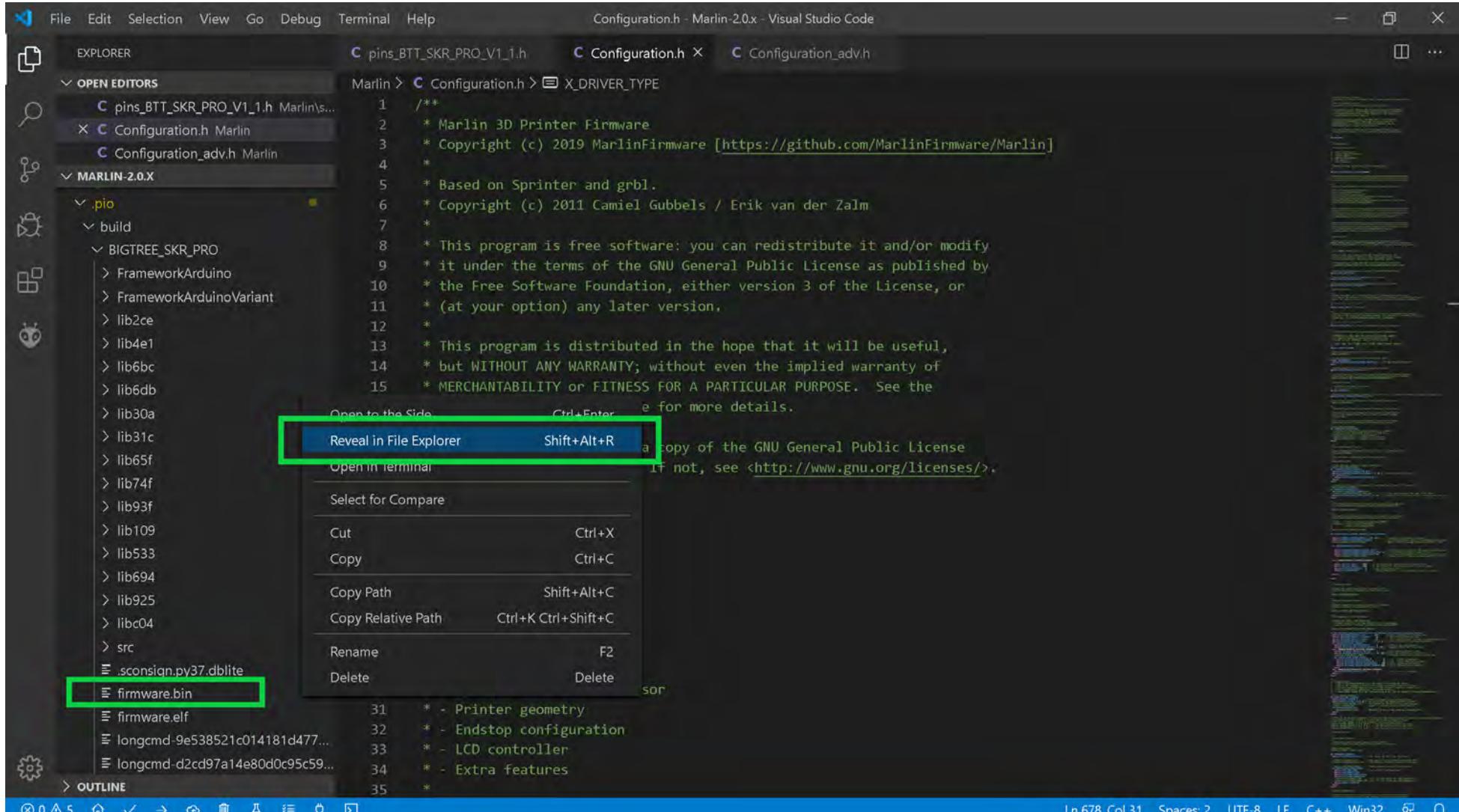

```

STH325102VE_52_01: BIGTREE_SKR_PRO SUCCESS 00:02:31.294
STH325102VE_52_01: BIGTREE_SKR_V1_0 IGNORED
STH325102VE_52_01: BIGTREE_BTT002 IGNORED
STH325102VE_52_01: teensy31 IGNORED
STH325102VE_52_01: teensy35 IGNORED
STH325102VE_52_01: esp32 IGNORED
STH325102VE_52_01: linux_native IGNORED
STH325102VE_52_01: SAMD51_grandcentral_m4 IGNORED
STH325102VE_52_01: rumba32_f446ve IGNORED
STH325102VE_52_01: mks_rumba32 IGNORED
STH325102VE_52_01: include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
      
```
- Status Bar:** Shows file count (0), line (Ln 678), column (Col 31), spaces (Spaces: 2), and file type (C++).

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

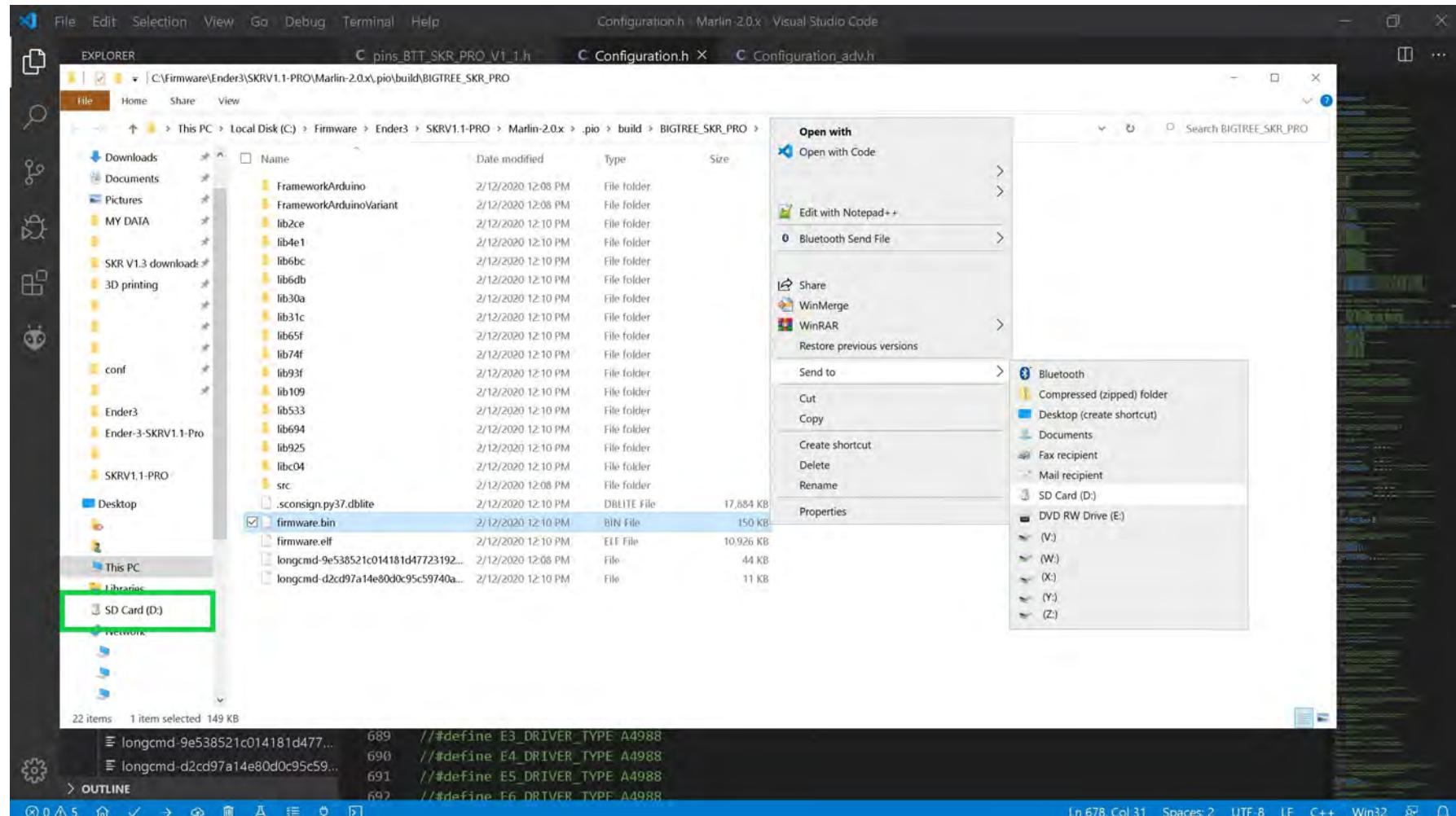
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

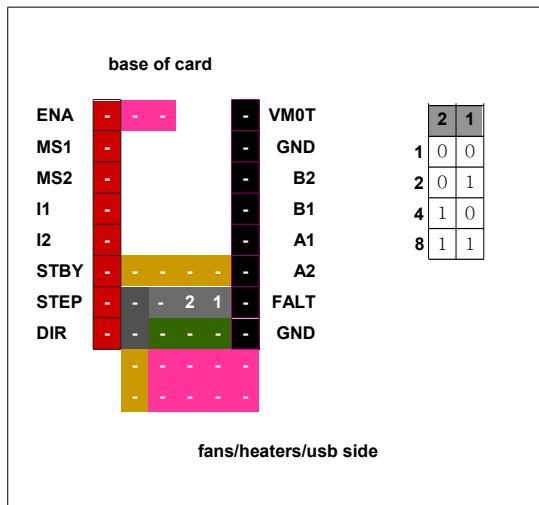
The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

POLOLU MP6500

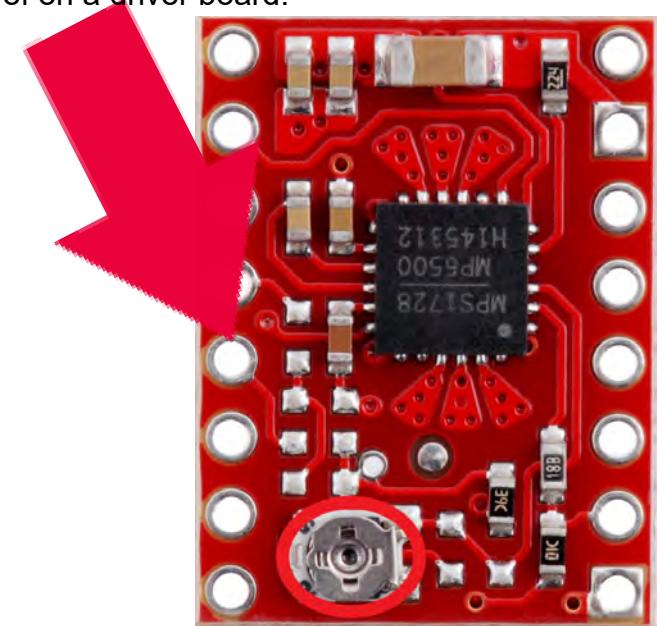
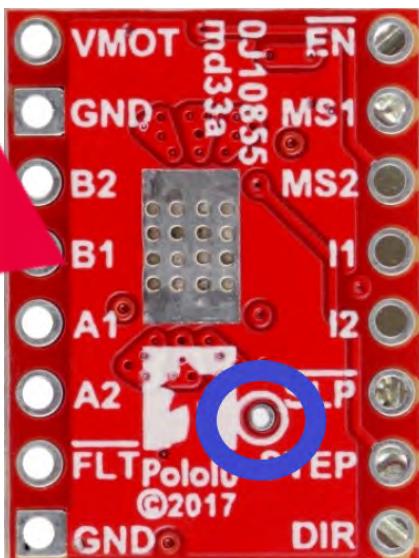


Driver Chip	MS2	MS1	Microstep Resolution
MP6500	Low	Low	Full step
Maximum 16 subdivision	Low	High	Half (1/2) step
4.5 V to 35 V 1.5 A	High	Low	Quarter (1/4) step
	High	High	Eighth (1/8) step
Driving current calculation formula	$I_{MAX} = V_{ref} * 3.5$		
	$V_{ref} = \frac{I_{MAX}}{3.5}$		

Note: Use 90% of the calculated Vref when tuning the stepper driver board

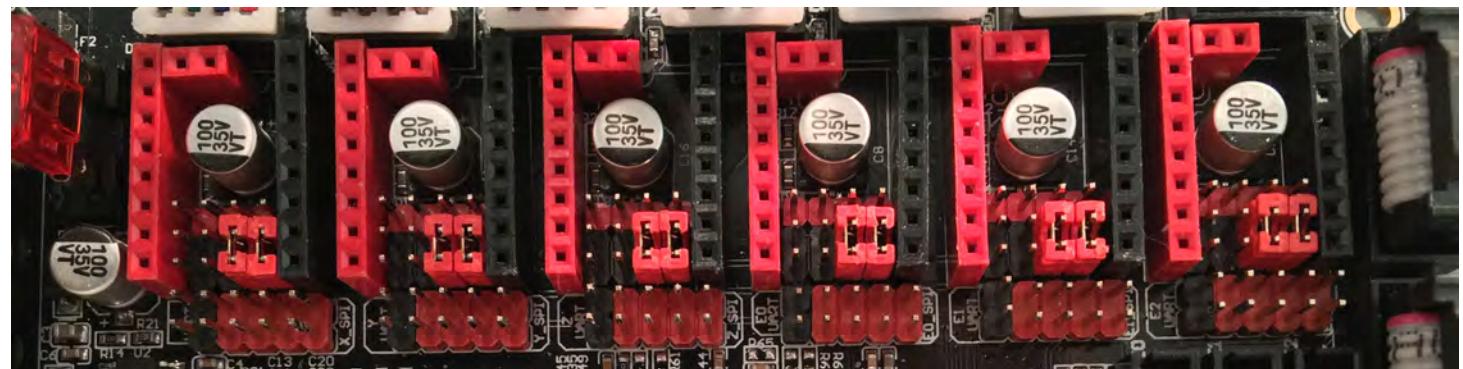
Note: "Vref Test point" location is on the bottom of the driver board, as shown in **BLUE**

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to adjust your Vref [See the appendix](#) for instructions on how to set the Vref on a driver board.



POLOLU MP6500

ENA	[]	VMOT	-
MS1	[]	GND	-
MS2	[]	B2	-
I1	[]	B1	-
I2	[]	A1	-
STBY	[]	A2	-
STEP	[]	FALT	-
DIR	[]	GND	-
	[]	STEP	-



ENA	[]	VMOT	-
MS1	[]	GND	-
MS2	[]	B2	-
I1	[]	B1	-
I2	[]	A1	-
STBY	[]	A2	-
STEP	[]	FALT	-
DIR	[]	GND	-
	[]	STEP	-

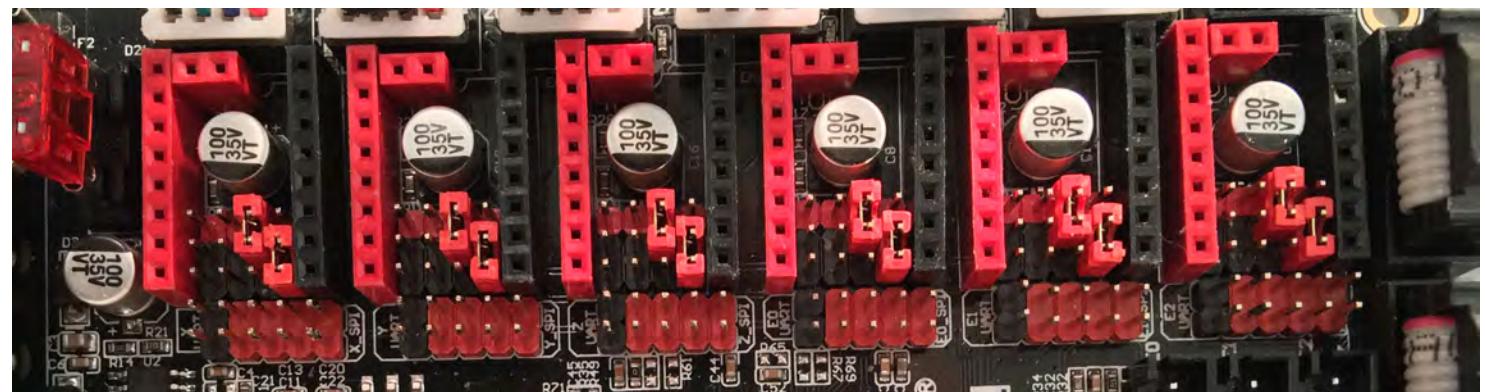


1 / 2

POLOLU MP6500

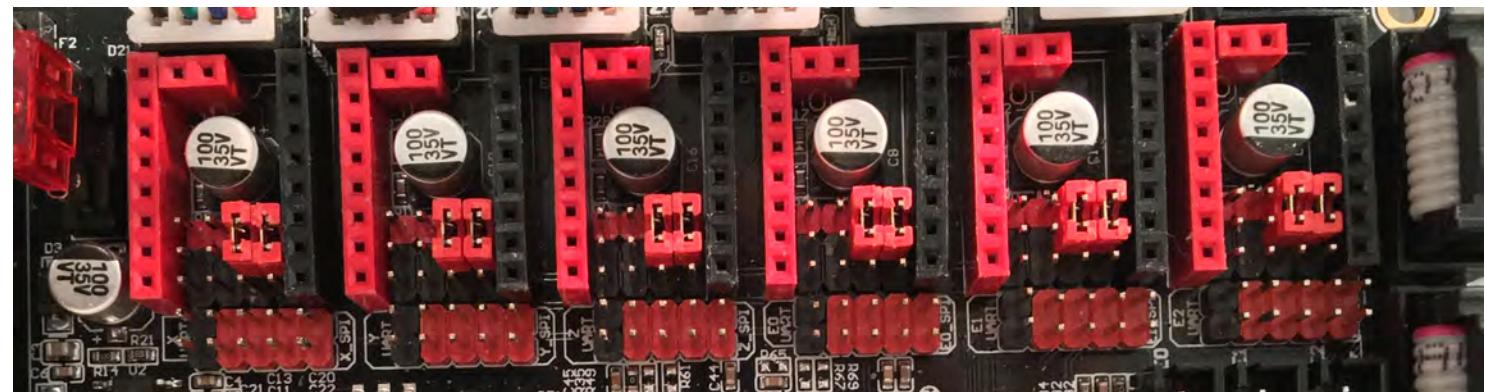
ENA	-	-	-	VMOT	-	-
MS1	-	-	-	GND	-	-
MS2	-	-	-	B2	-	-
I1	-	-	-	B1	-	-
I2	-	-	-	A1	-	-
STBY	-	-	2	A2	-	-
STEP	-	-	2	FALT	-	-
DIR	-	-	1	GND	-	-

1 / 4



ENA	-	-	-	VMOT	-	-
MS1	-	-	-	GND	-	-
MS2	-	-	-	B2	-	-
I1	-	-	-	B1	-	-
I2	-	-	-	A1	-	-
STBY	-	-	2	A2	-	-
STEP	-	-	2	FALT	-	-
DIR	-	-	1	GND	-	-

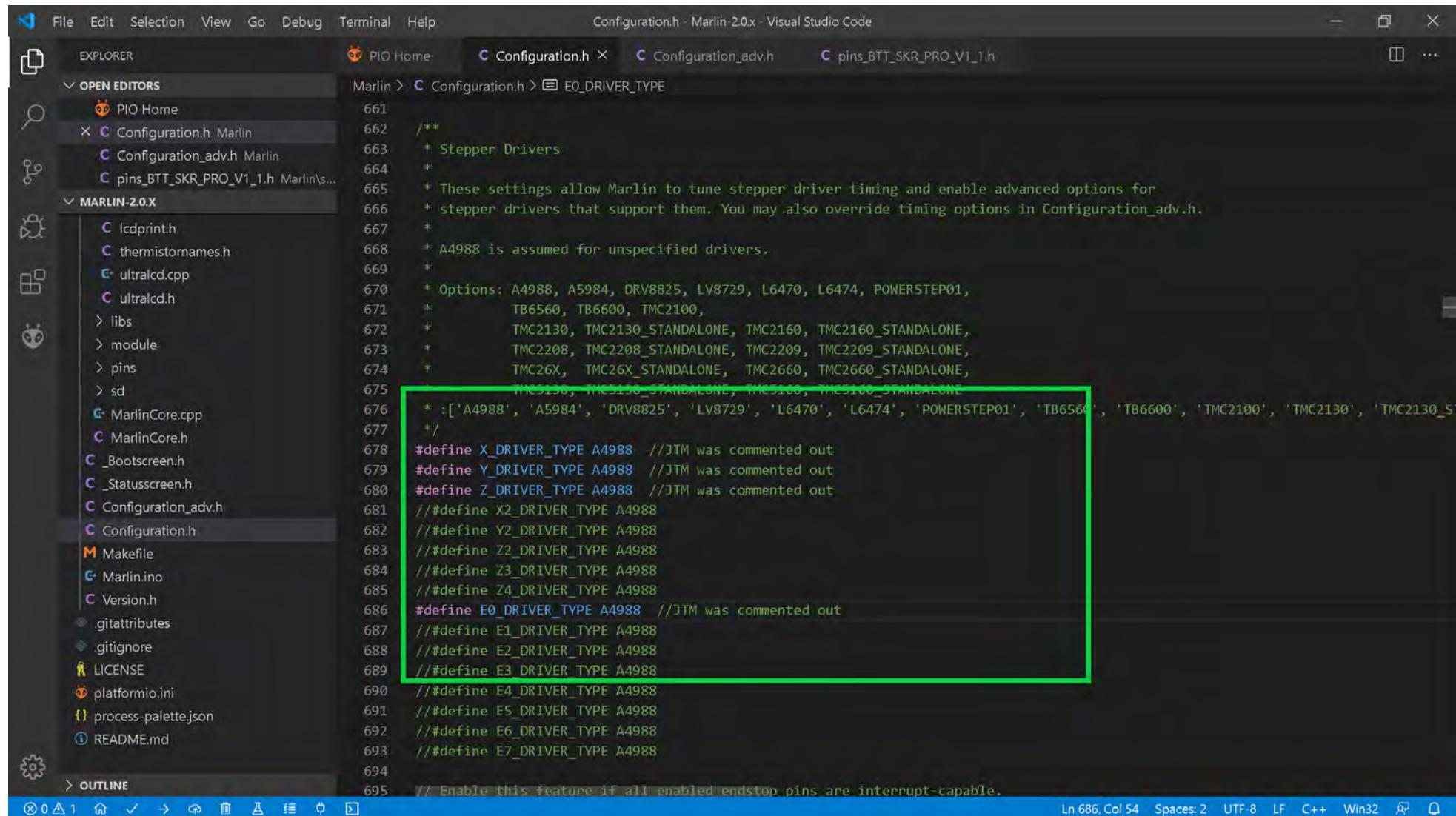
1 / 8



The (latest release of) Marlin Setup for POLOLU MP6500 Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for POLOLU MP6500 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using POLOLU MP6500 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU MP6500 drivers. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The ST820 is a drop in replacement for the A4988. Since Marlin does not have an option for POLOLU MP6500 we will use the A4988 as the driver type.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following relevant sections:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC3180, TMC3180_STANDALONE, TMC3180, TMC3180_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
677 */
678 #define X_DRIVER_TYPE A4988 //JTM was commented out
679 #define Y_DRIVER_TYPE A4988 //JTM was commented out
680 #define Z_DRIVER_TYPE A4988 //JTM was commented out
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE A4988 //JTM was commented out
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

A green rectangular box highlights the driver type definitions for the X, Y, Z, and E0 axes, specifically the lines starting with `#define`.

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU MP6500 Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for POLOLU MP6500 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/8 stepping. So we are cutting our STEPS in half. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16.** So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {40, 40, 200, 46.5}, as seen in the **GREEN** box below.

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h ...
PIO Home 721 * These settings can be reset by M502
Configuration.h M... 722 *
Configuration_adv... 723 * Note that if EEPROM is enabled, saved values will override these.
pins_BTT_SKR_PR... 724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732 /**
733 * Default Axis Steps Per Unit (steps/mm)
734 * Override with M92
735 */
736 /**
737 * Y, Y, Z, E0 [, E1[, E2...]]
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 40, 40, 200, 46.5 } // JTM was {80, 80, 400, 93} for A4988 on Ender 3
739 // The MP6500 step maximum is 1/8 step which is half of 1/16 step
740 /**
741 * Default Max Feed Rate (mm/s)
742 * Override with M203
743 */
744 * X, Y, Z, E0 [, E1[, E2...]]
745 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
746
747 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
748 #if ENABLED(LIMITED_MAX_FR_EDITING)
749 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
750#endif
751 /**
752 * Default Max Acceleration (change/s) change = mm/s
753 * (Maximum start speed for accelerated moves)
754 * Override with M201
755 */

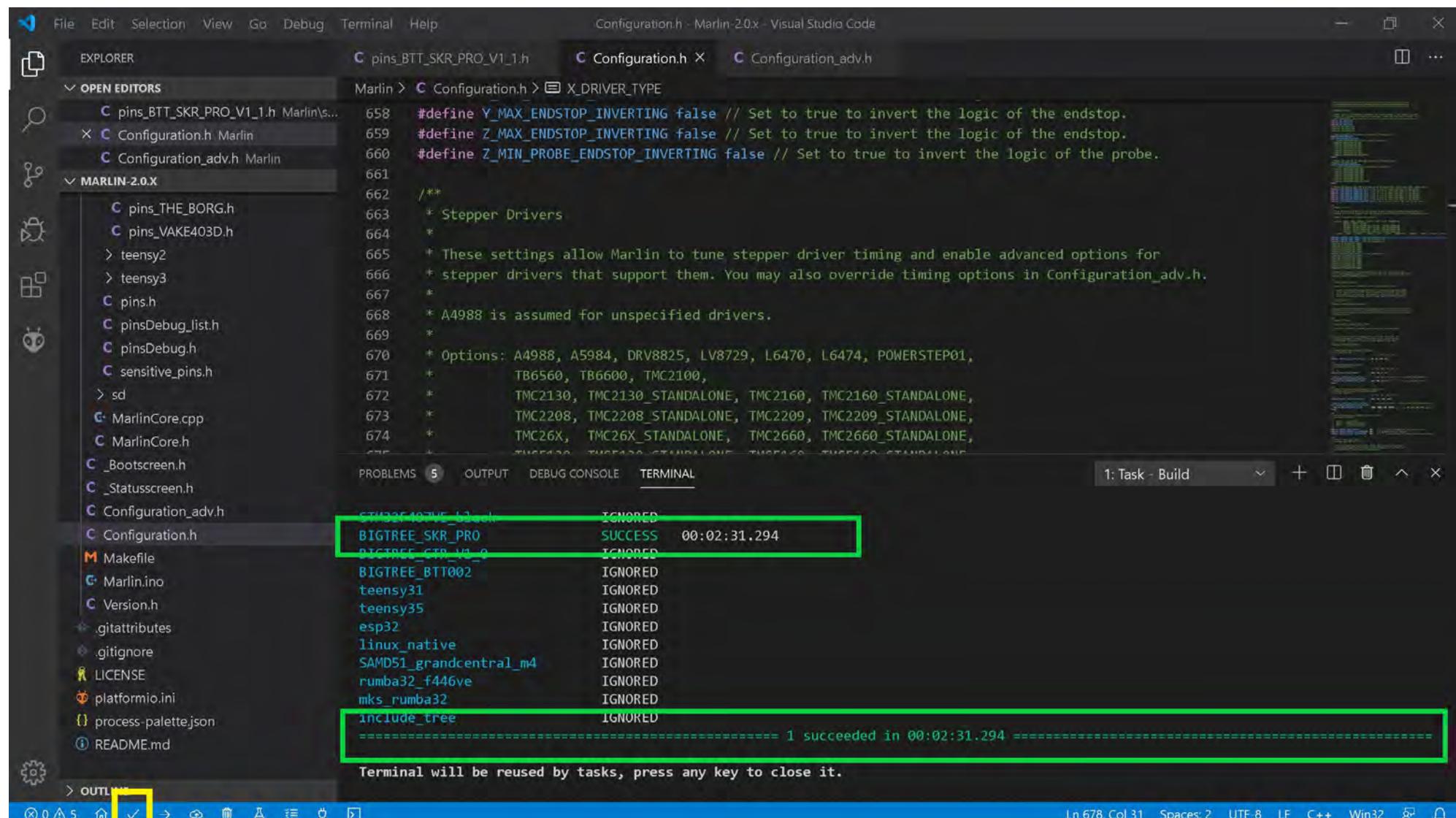
```

Ln 739, Col 126 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU MP6500 Drivers

- The end of Marlin setup for POLOLU MP6500 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



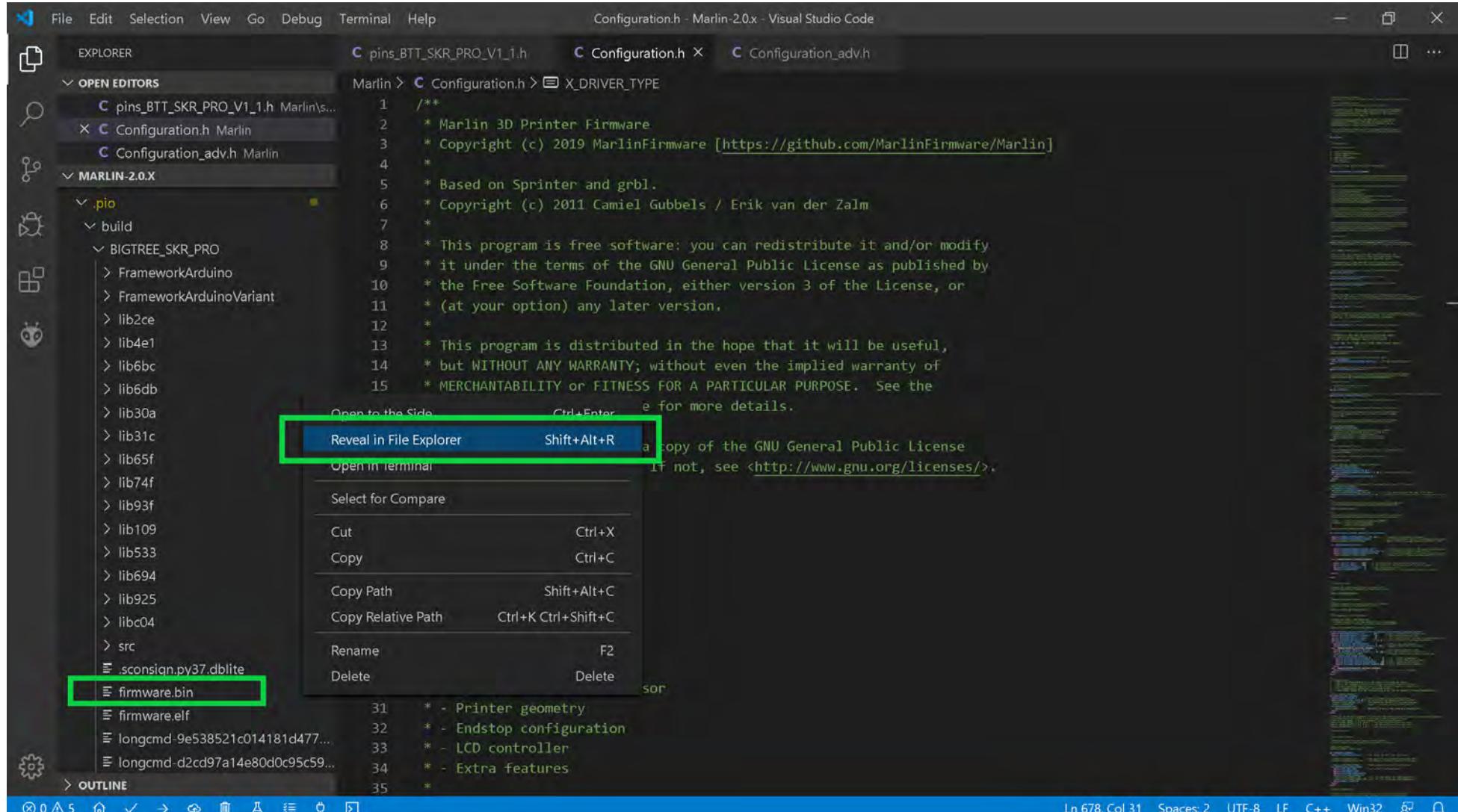
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and various Marlin core files.
- Editor:** Displays the Configuration.h file with code related to endstop inversion and probe settings.
- Terminal:** Shows the build log for the BIGTREETECH SKR Pro board. The log includes:
 - Build time: 00:02:31.294
 - Targets: BIGTREETECH_SKR_Pro, BIGTREETECH_SKR_V1_0, BIGTREE_BTT002, teensy31, teensy35, esp32, linux_native, SAMD51_grandcentral_m4, rumba32_f446ve, mks_rumba32, include_tree.
 - Status: All targets are marked as IGNORED.
 - Summary: 1 succeeded in 00:02:31.294
- Bottom Status Bar:** Shows file count (0), line (Ln 678), column (Col 31), spaces (Spaces: 2), and file type (C++).

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for POLOLU MP6500 Drivers

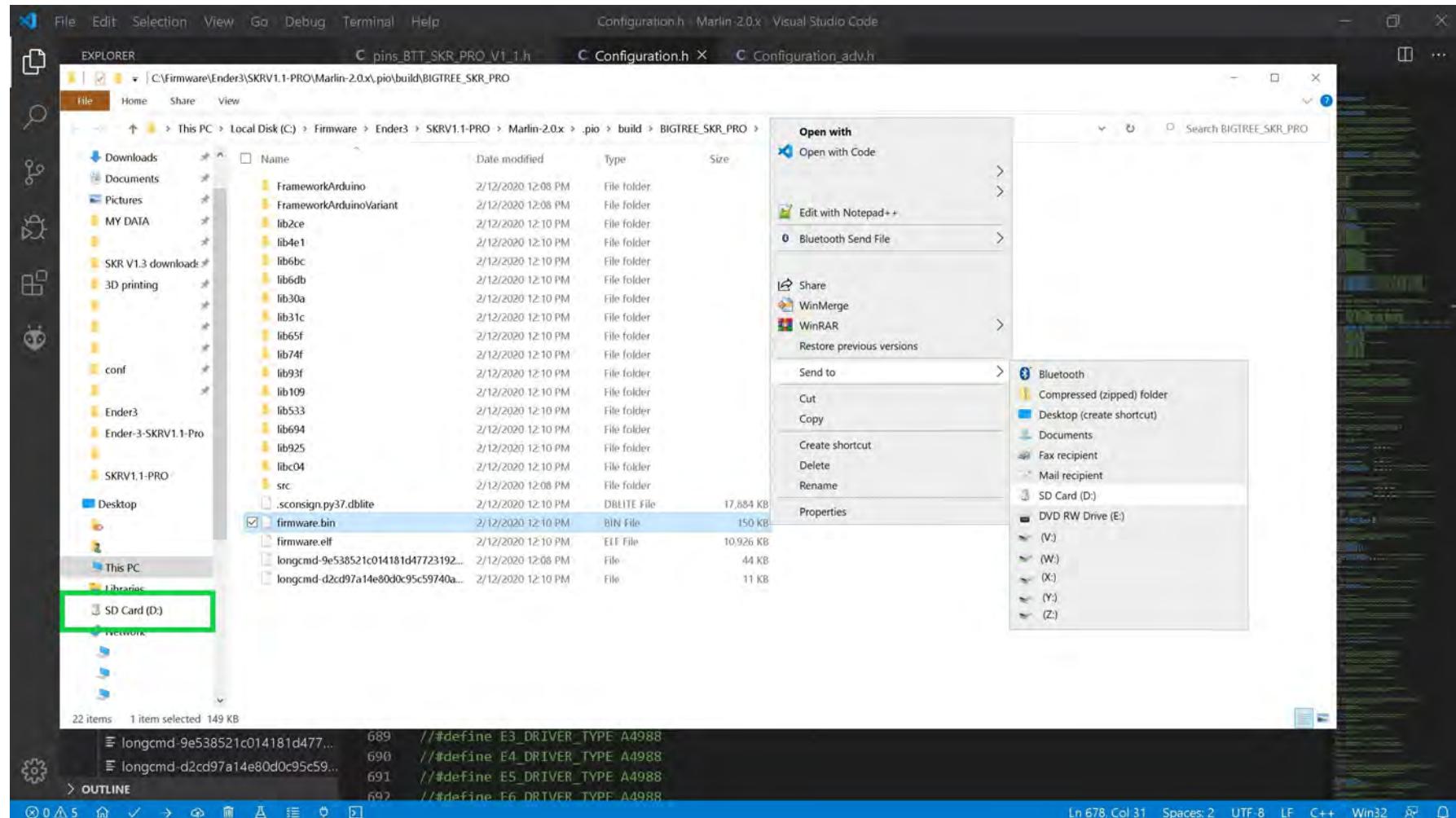
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

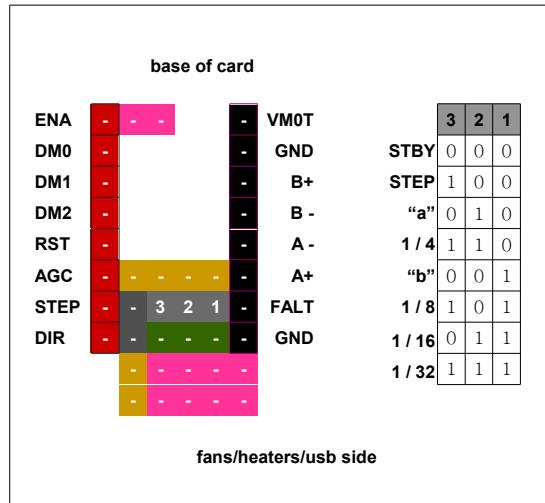
The (latest release of) Marlin Setup for POLOLU MP6500 Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

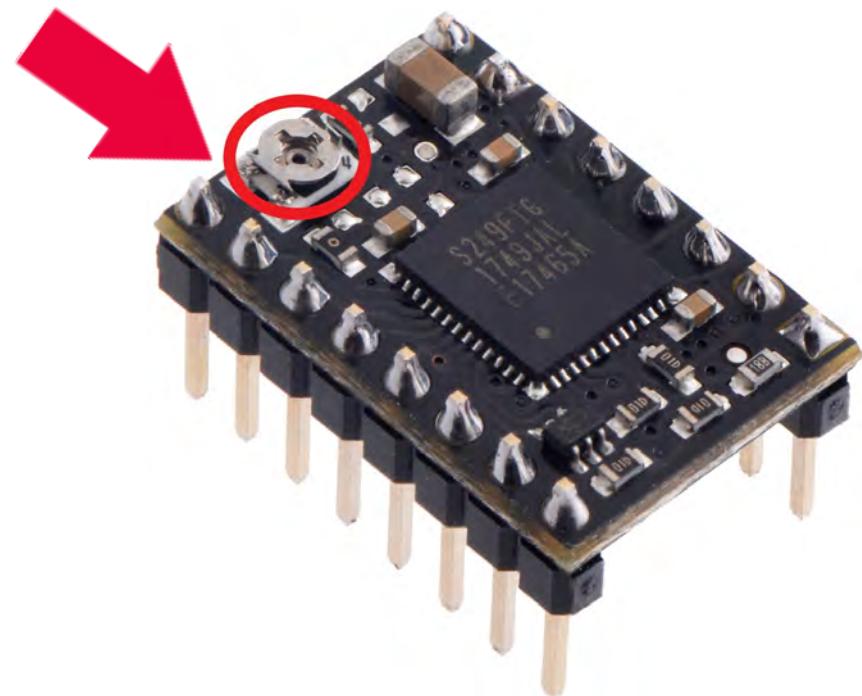
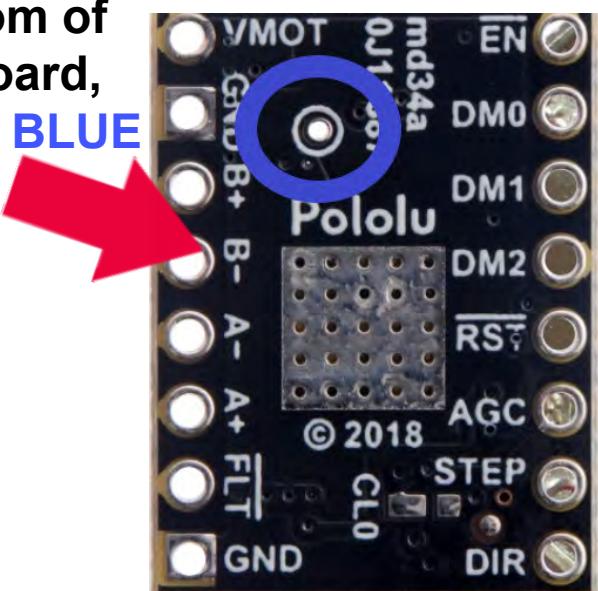
POLOLU TB67S249FTG



Driver Chip	D MODE2	D MODE1	D MODE0	Microstep Resolution
TB67S249FTG	Low	Low	Low	Standby mode (outputs disabled)
	High	Low	Low	Full step
Maximum 32 subdivision	Low	High	Low	Non-circular half step ("a")
10 V to 47 V 1.6 A	High	High	Low	1/4 step
	Low	Low	High	Circular half step ("b")
	High	Low	High	1/8 step
	Low	High	High	1/16 step
	High	High	High	1/32 step
Driving current calculation formula	$I_{MAX} = V_{REF} * 1.25$		$V_{REF} = \frac{I_{MAX}}{1.25}$	
	Note: Use 90% of the calculated Vref when tuning the stepper driver board			

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to adjust your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

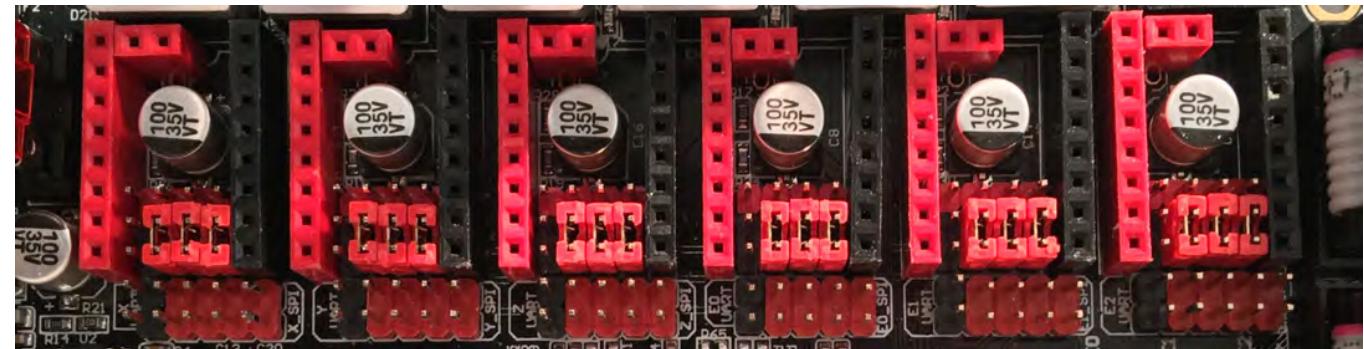
Note: "Vref Test point" location is on the Bottom of the driver board, as shown in **BLUE**



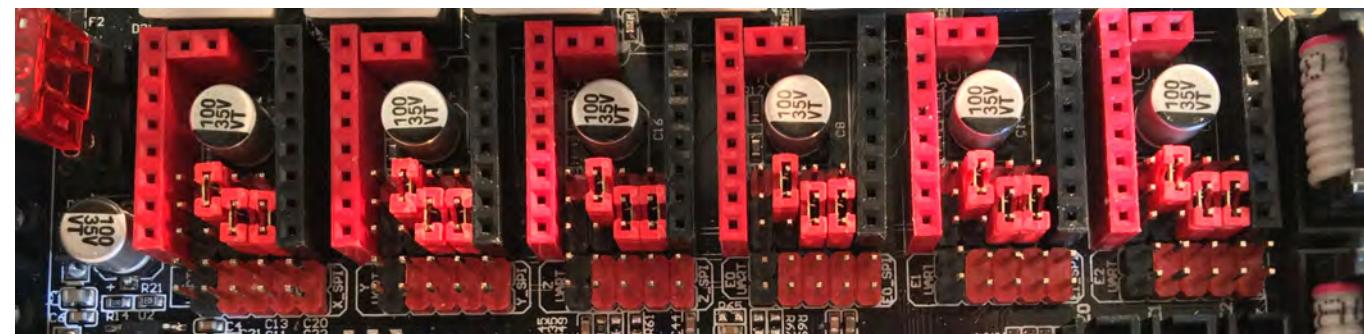
POLOLU TB67S249FTG

Note: “a” and “b” referred to a non-circular, and circular half step. Please understand that I do not know what this means and did not see a description about this in the vendor data. If you click on the title of this page the datasheet is available.

ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B -
RST	-	-	-	A -
AGC	-	-	-	A +
STEP	-	3 2 1	-	FALT
DIR	-	3 2 1	-	GND
Stand By				



ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B -
RST	-	-	-	A -
AGC	-	3	-	A +
STEP	-	3 2 1	-	FALT
DIR	-	-	2 1	GND
STEP				

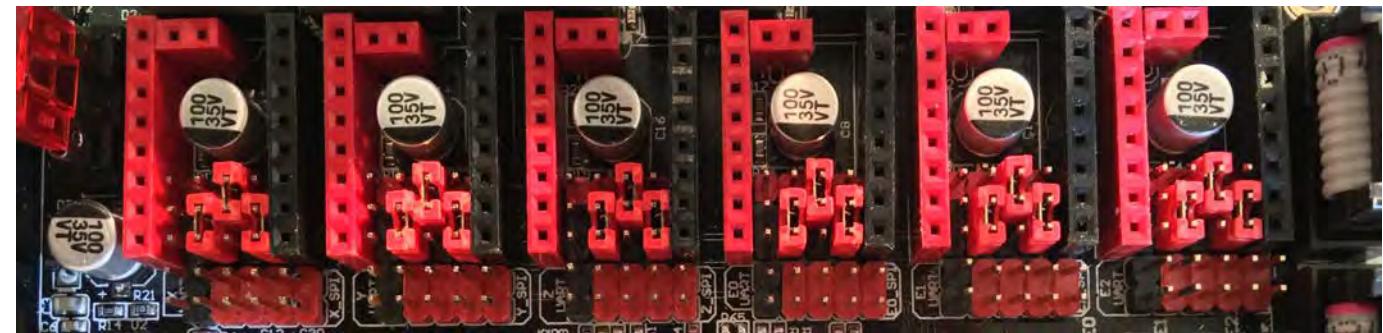


POLOLU TB67S249FTG

Note: “a” and “b” referred to a non-circular, and circular half step. Please understand that I do not know what this means and did not see a description about this in the vendor data. If you click on the title of this page the datasheet is available.

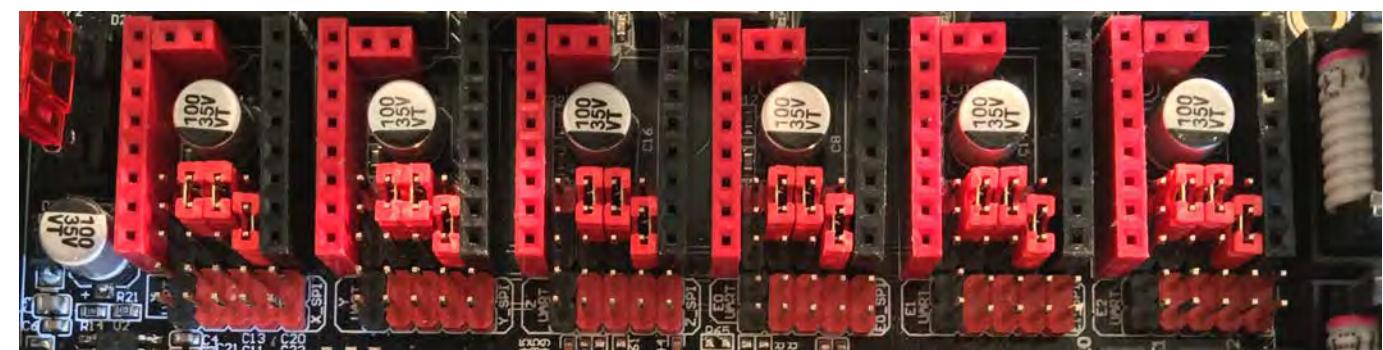
ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	-	2	A+
STEP	-	3	2	FALT
DIR	-	3	-	GND

1/2 on “a”



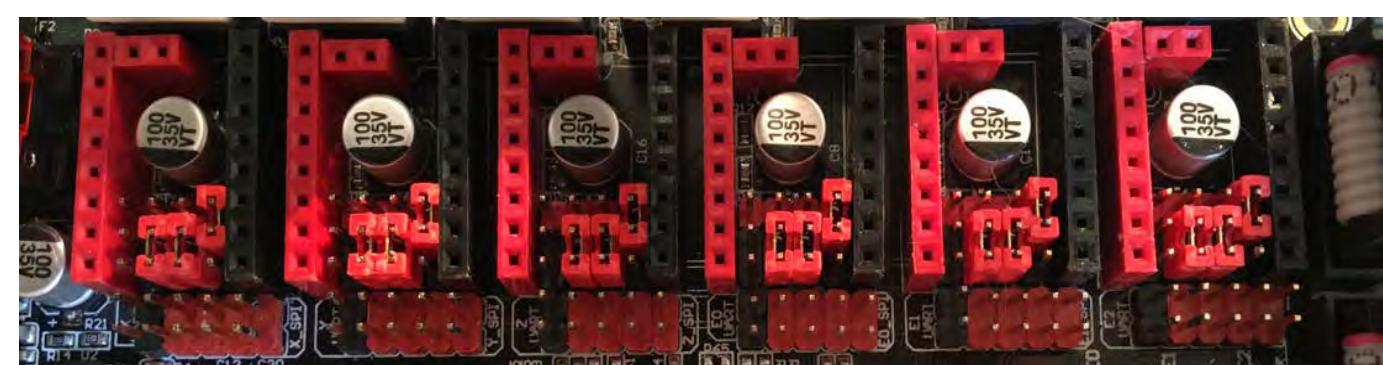
ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	-	3	A+
STEP	-	3	2	FALT
DIR	-	-	-	GND

1 / 4



ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	-	-	A+
STEP	-	-	3	FALT
DIR	-	-	2	GND

1/2 on “b”

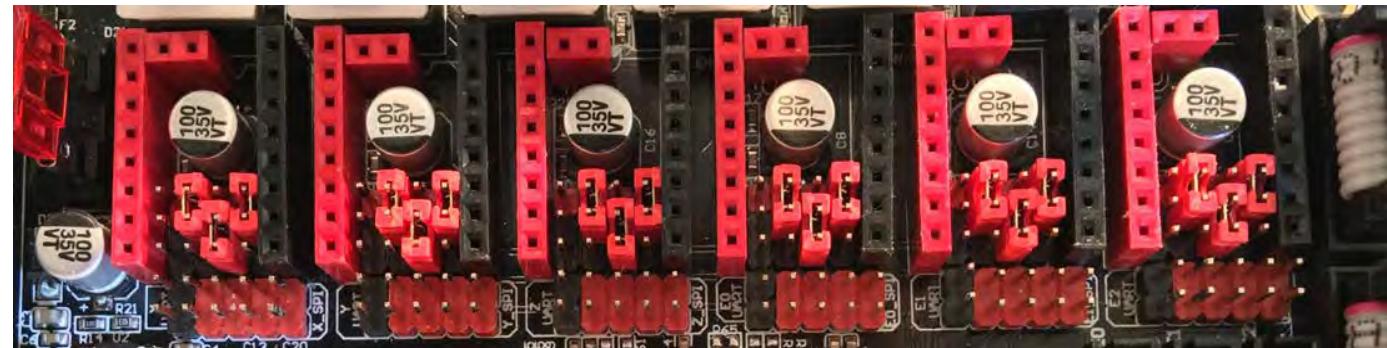


POLOLU TB67S249FTG

Note: “a” and “b” referred to a non-circular, and circular half step. Please understand that I do not know what this means and did not see a description about this in the vendor data. If you click on the title of this page the datasheet is available.

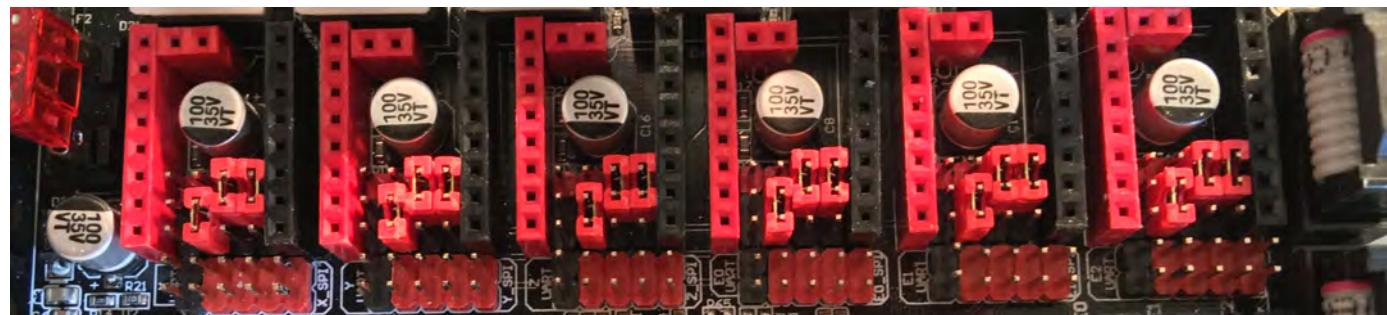
ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	3	1	A+
STEP	-	3	2	FALT
DIR	-	-	2	GND

1 / 8



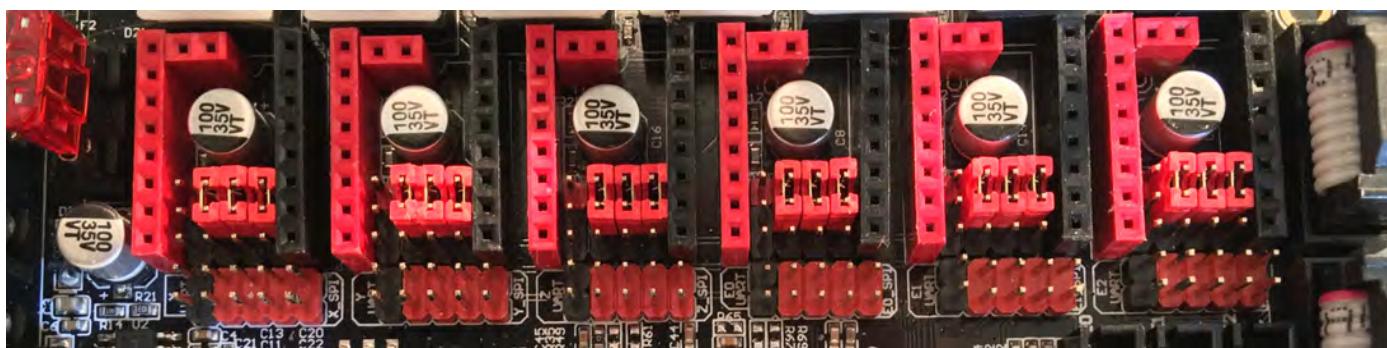
ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	2	1	A+
STEP	-	3	2	FALT
DIR	-	3	-	GND

1 / 16



ENA	-	-	-	VMOT
DM0	-	-	-	GND
DM1	-	-	-	B+
DM2	-	-	-	B-
RST	-	-	-	A-
AGC	-	3	2	A+
STEP	-	3	2	FALT
DIR	-	-	2	GND

1 / 32



The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

NOTE: Go to Appendix C then come back here for the changes to Marlin for POLOLU TB67S249FTG stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using POLOLU TB67S249FTG drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU TB67S249FTG drivers. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The **POLOLU TB67S249FTG** is a drop in replacement for the A4988. Since Marlin does not have an option for POLOLU TB67S249FTG we will use the A4988 as the driver type.

```

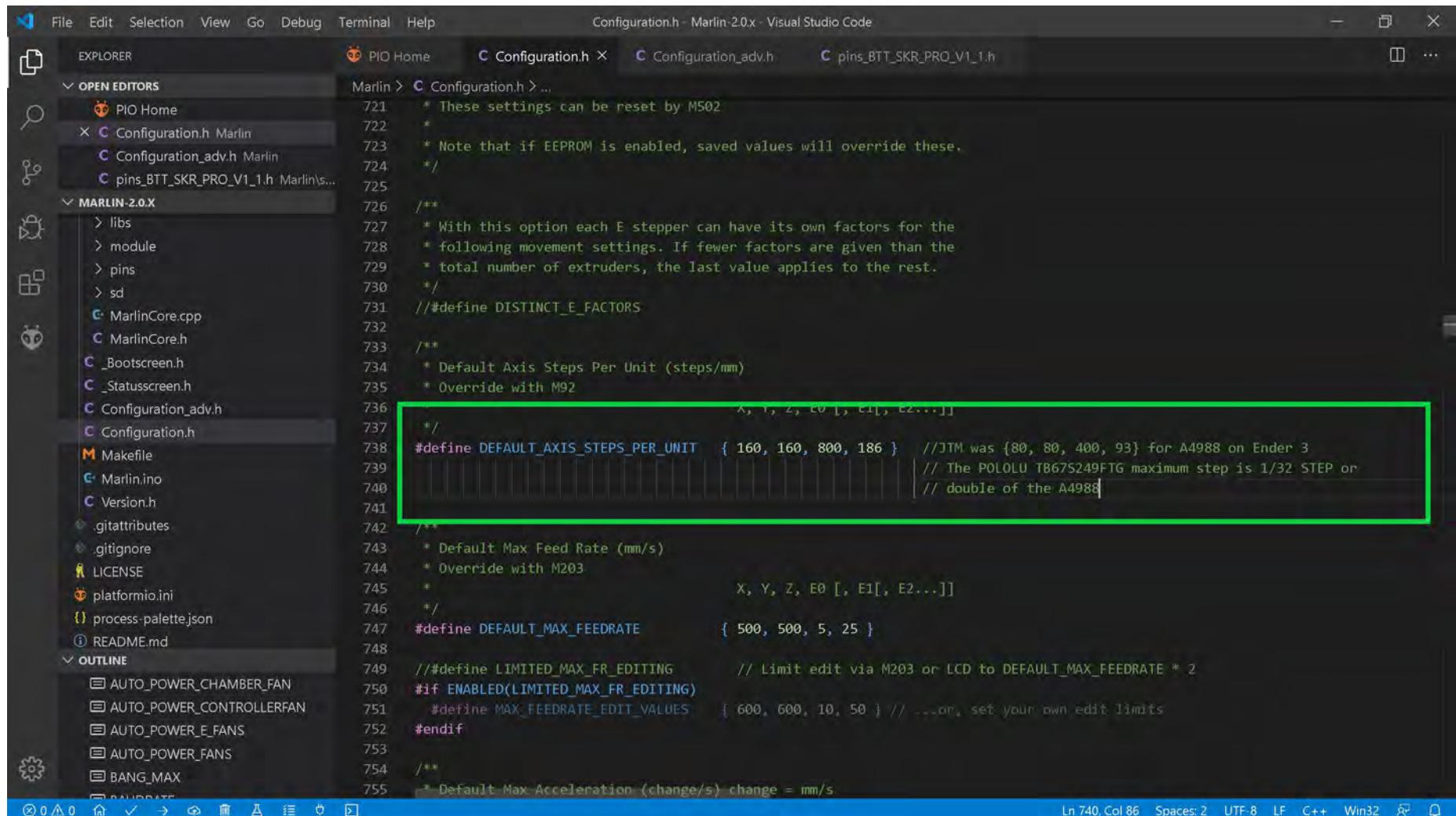
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home
Configuration.h Marlin
Configuration_adv.h Marlin
pins_BTT_SKR_PRO_V1_1.h Marlin's...
MARLIN-2.0.X
lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
0 1 2 3 4 5 6 7 8 9
Ln 686, Col 54 Spaces: 2 UTF-8 LF C++ Win32
 661 /**
 662 * Stepper Drivers
 663 *
 664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 666 *
 667 * A4988 is assumed for unspecified drivers.
 668 *
 669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 670 *           TB6560, TB6600, TMC2100,
 671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 674 *           TMC5168, TMC5168_STANDALONE, TMC5186, TMC5186_STANDALONE
 675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208',
 676 *      'TMC26X', 'TMC5168', 'TMC5186']
 677 */
#define X_DRIVER_TYPE A4988 //JTM was commented out
#define Y_DRIVER_TYPE A4988 //JTM was commented out
#define Z_DRIVER_TYPE A4988 //JTM was commented out
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE A4988 //JTM was commented out
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
 695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for POLOLU TB67S249FTG stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, **we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h ...
721 * These settings can be reset by M502
722 *
723 * Note that if EEPROM is enabled, saved values will override these.
724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732 /**
733 * Default Axis Steps Per Unit (steps/mm)
734 * Override with M92
735 * X, Y, Z, E0 [, E1[, E2...]]
736 */
737 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // JTM was {80, 80, 400, 93} for A4988 on Ender 3
738 // The POLOLU TB67S249FTG maximum step is 1/32 STEP or
739 // double of the A4988
740
741 /**
742 * Default Max Feed Rate (mm/s)
743 * Override with M203
744 * X, Y, Z, E0 [, E1[, E2...]]
745 */
746 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
747
748 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
749 #if ENABLED(LIMITED_MAX_FR_EDITING)
750 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
751 #endif
752
753 /**
754 * Default Max Acceleration (change/s) change = mm/s
755

```

Ln 740, Col 86 Spaces:2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

- The end of Marlin setup for POLOLU TB67S249FTG drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages ,as shown in the **GREEN** boxes below.

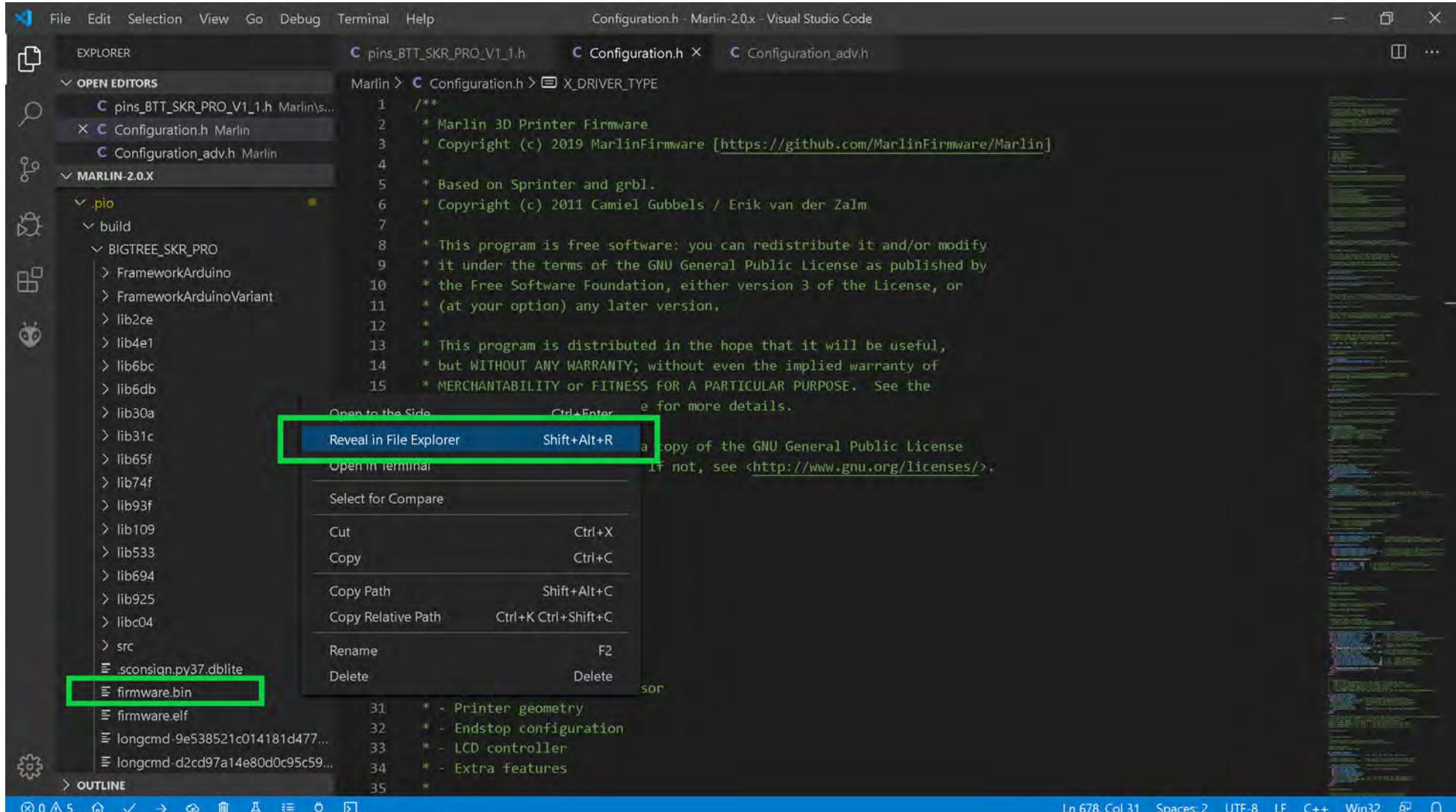
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editor:** The main editor window displays the content of Configuration.h, specifically the X_DRIVER_TYPE section.
- Terminal:** The bottom-right panel shows the build log. A green box highlights the first few lines of the log, which include the compilation of "BIGTREE_SKR_PRO" and other driver configurations. The log concludes with a message indicating 1 succeeded in 00:02:31.294.
- Status Bar:** Shows file counts (0 0 5), a checked checkbox icon, and file paths like "C:\Users\...\Marlin\src\main.cpp".
- Bottom Bar:** Includes tabs for File, Edit, Selection, View, Go, Debug, Terminal, Help, and a status bar with "Ln 678, Col 31, Spaces: 2, UTF-8, LF, C++, Win32".

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

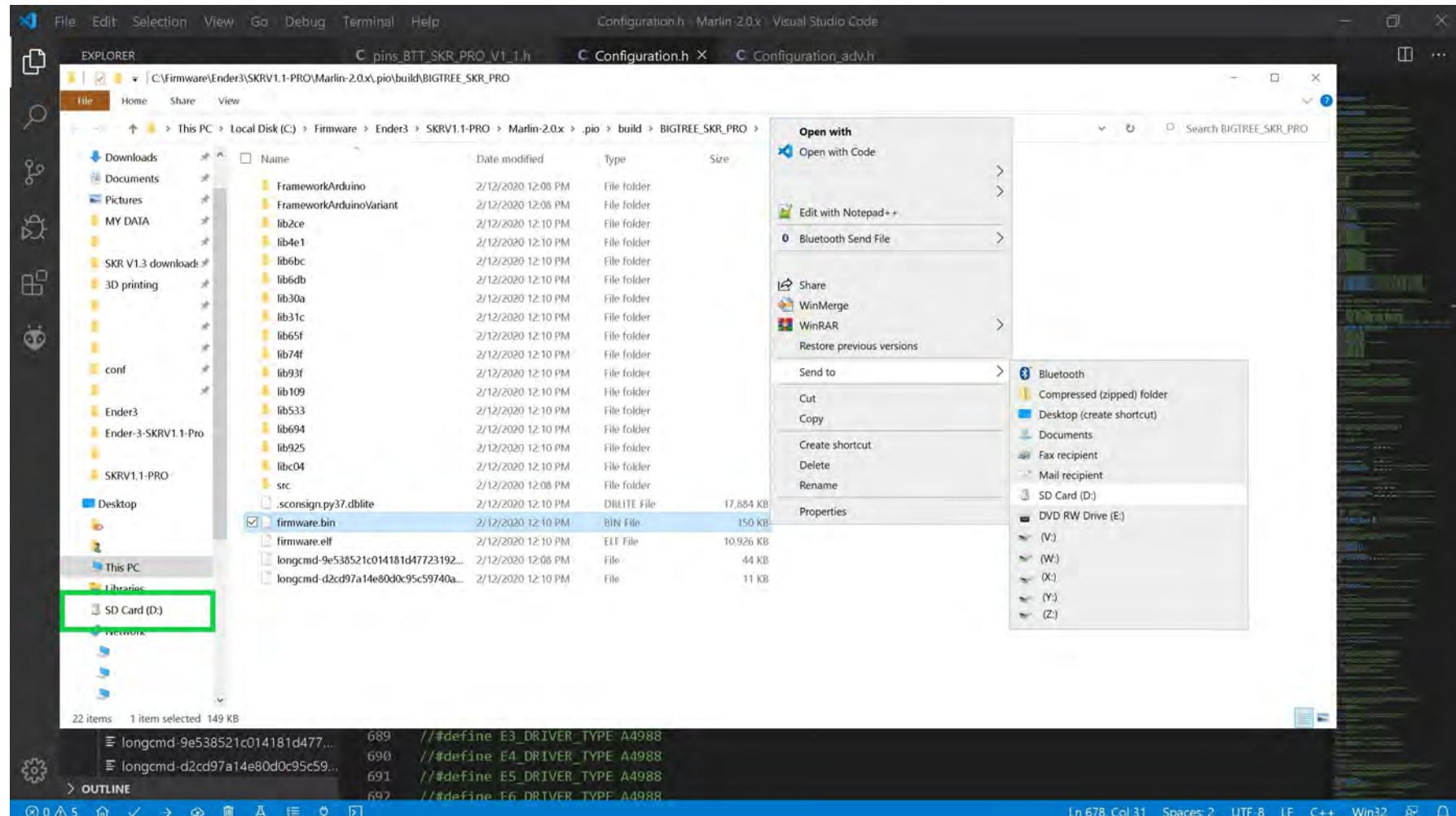
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



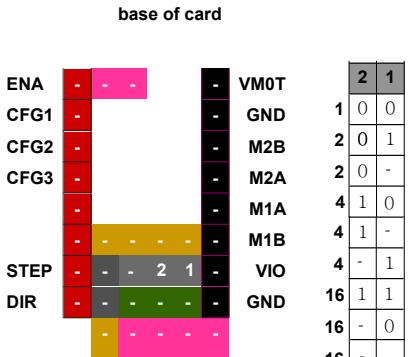
- Go to the next page.

The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU MKS TMC2100**Stand Alone Mode****Stand Alone Mode**

fans/heaters/usb side

Driver Chip	CFG2	CFG1	Steps	Interpolation	Mode
TMC2100 Standalone Mode Maximum subdivision 16 5.5V - 46V DC 1.4A	GND	GND	1 (full step)	none	spreadCycle
	GND	VIO	2 (half-step)	none	spreadCycle
	GND	open	2 (half-step)	256 µ-steps	spreadCycle
	VIO	GND	4 (quarter-step)	none	spreadCycle
	VIO	open	4 (quarter-step)	256 µ-steps	spreadCycle
	open	VIO	4 (quarter-step)	256 µ-steps	stealthChop
	VIO	VIO	16 µ-steps	none	spreadCycle
	open	GND	16 µ-steps	256 µ-steps	spreadCycle
	open	open	16 µ-steps	256 µ-steps	stealthChop

Driving current calculation formula

$$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$$

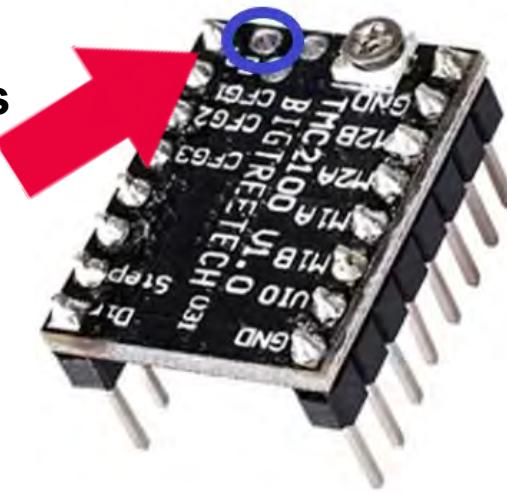
$$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$$

Note: Use 90% of the calculated Vref when tuning the stepper driver board

Note: Use the potentiometer (POT) on the top of the board (or use the board's "Vref Test point") to set your Vref. [See the appendix](#) for instructions on how to set the Vref on a driver board.

To learn more, please watch this [Youtube video done by Teaching Tech](#) and check out this link on the [TMC2100 Driver](#)

Note: "Vref Test point" location is on the top of the driver board, as shown in **BLUE**



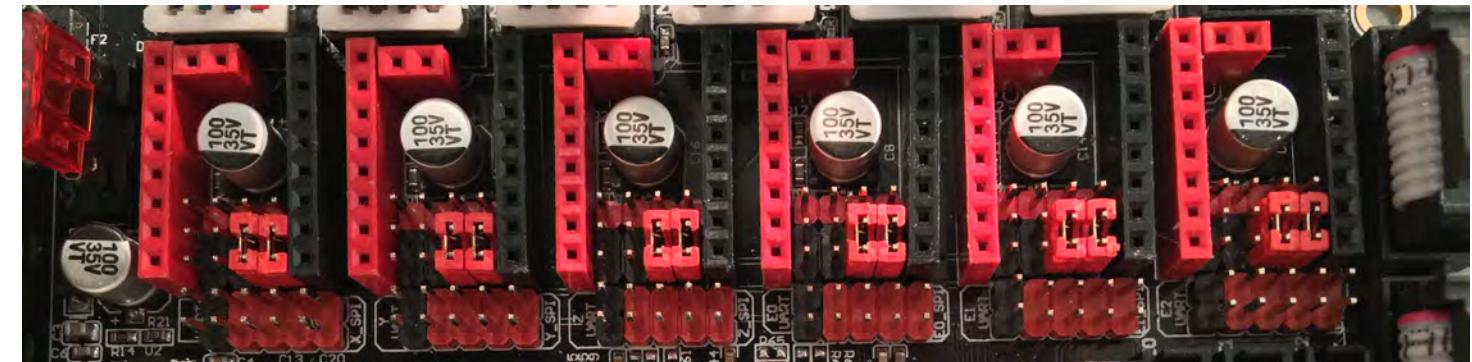
BIQU MKS TMC2100**Stand Alone Mode****Stand Alone Mode**

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
STEP	-	-	2 1	M1A
DIR	-	-	2 1	M1B
				VIO
				GND

STEP

Interpolation: none

SpreadCycle

**Stand Alone Mode**

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
STEP	-	-	1	M1A
DIR	-	-	2 1	M1B
				VIO
				GND

1/2

Interpolation: none

SpreadCycle

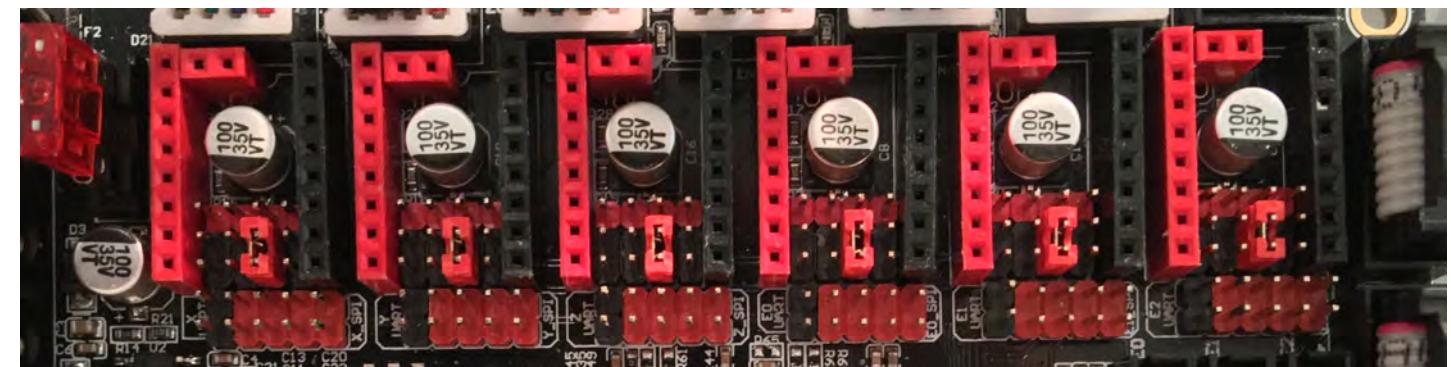
**Stand Alone Mode**

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
STEP	-	-	2	M1A
DIR	-	-	2	M1B
				VIO
				GND

1/2

Interpolation: 1/256

SpreadCycle

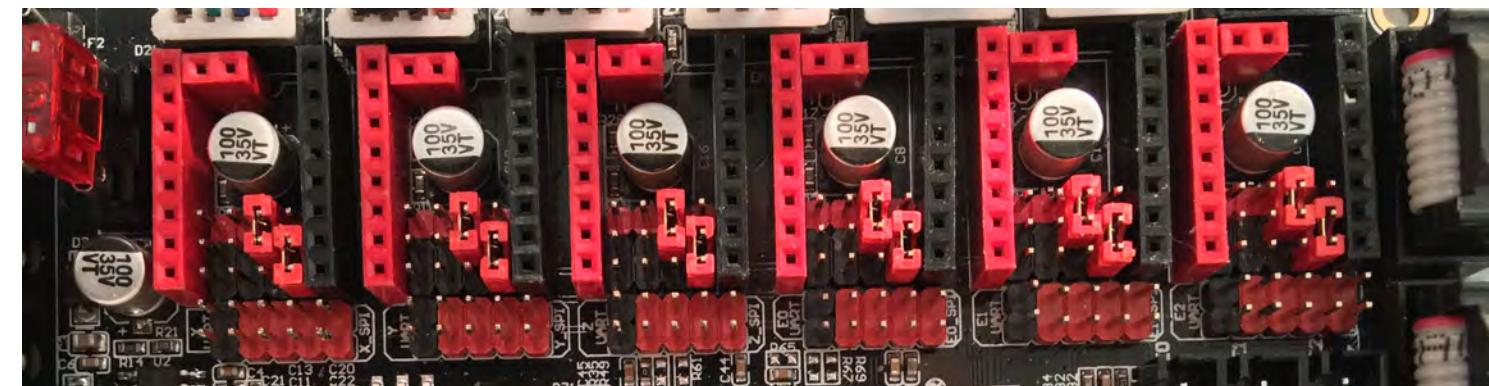


NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

BIQU MKS TMC2100**Stand Alone Mode****Stand Alone Mode**

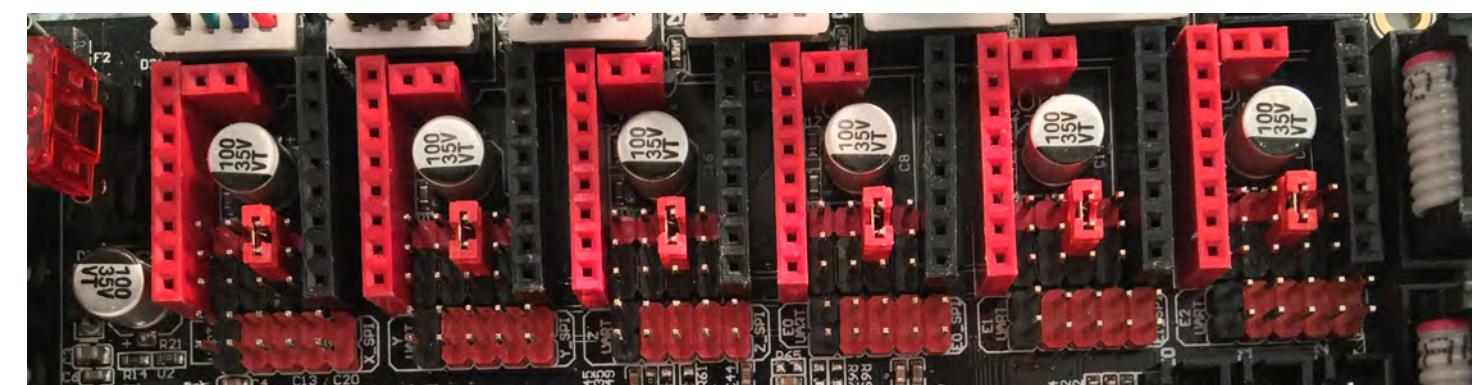
ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	2	M1A
STEP	-	-	2	M1B
DIR	-	-	1	VIO
	-	-	1	GND

1/4
Interpolation: none
SpreadCycle

**Stand Alone Mode**

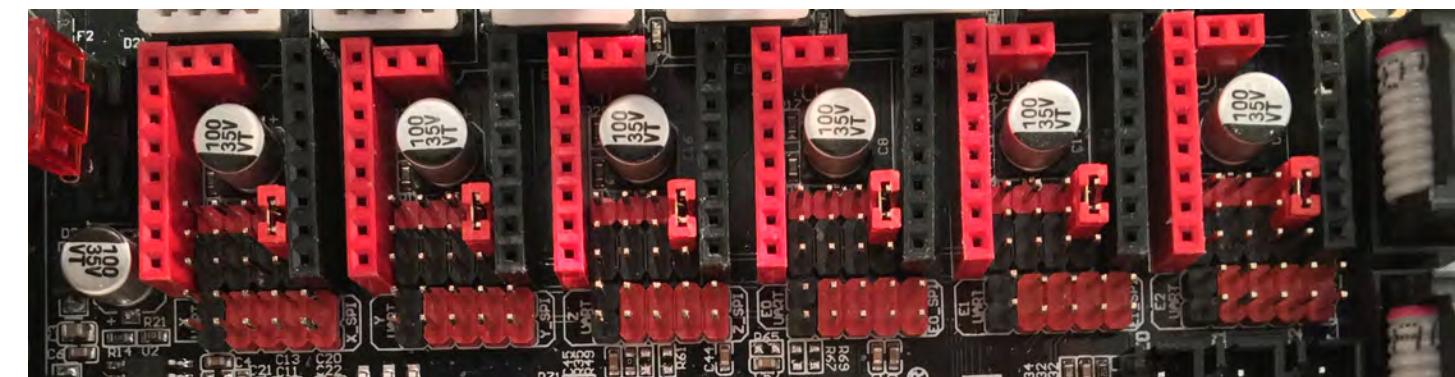
ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	2	M1A
STEP	-	-	2	M1B
DIR	-	-	-	VIO
	-	-	1	GND

1/4
Interpolation: 1/256
SpreadCycle

**Stand Alone Mode**

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	1	M1A
STEP	-	-	1	M1B
DIR	-	-	-	VIO
	-	-	1	GND

1/4
Interpolation: 1/256
StealthChop



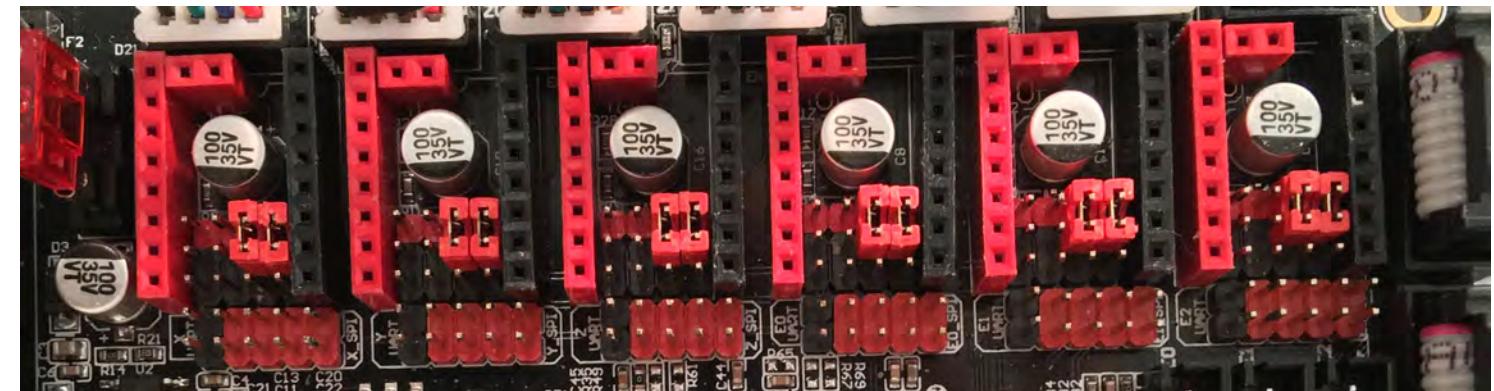
NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

BIQU MKS TMC2100

Stand Alone Mode

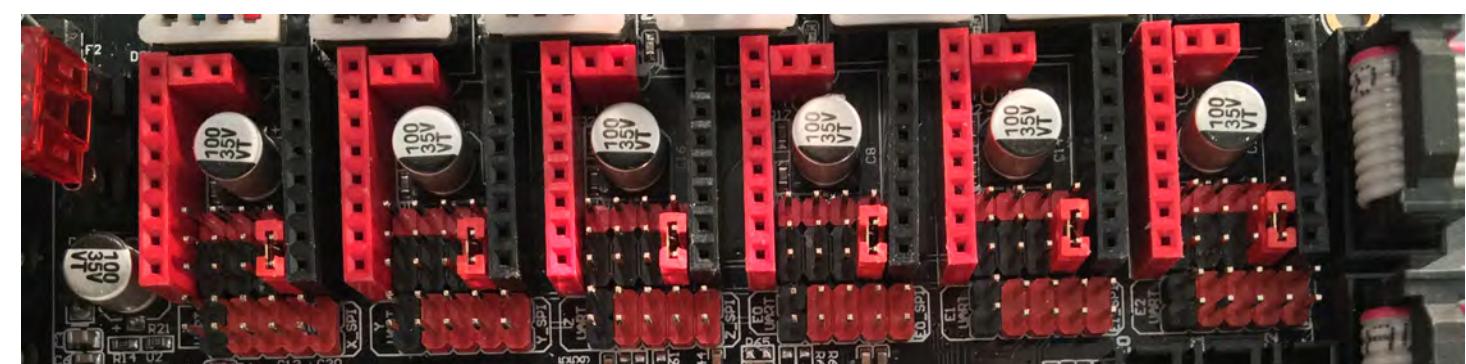
Stand Alone Mode

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	2 1	M1A
STEP	-	-	2 1	M1B
DIR	-	-	-	VIO
	-	-	-	GND
1/16				
Interpolation: none				
SpreadCycle				



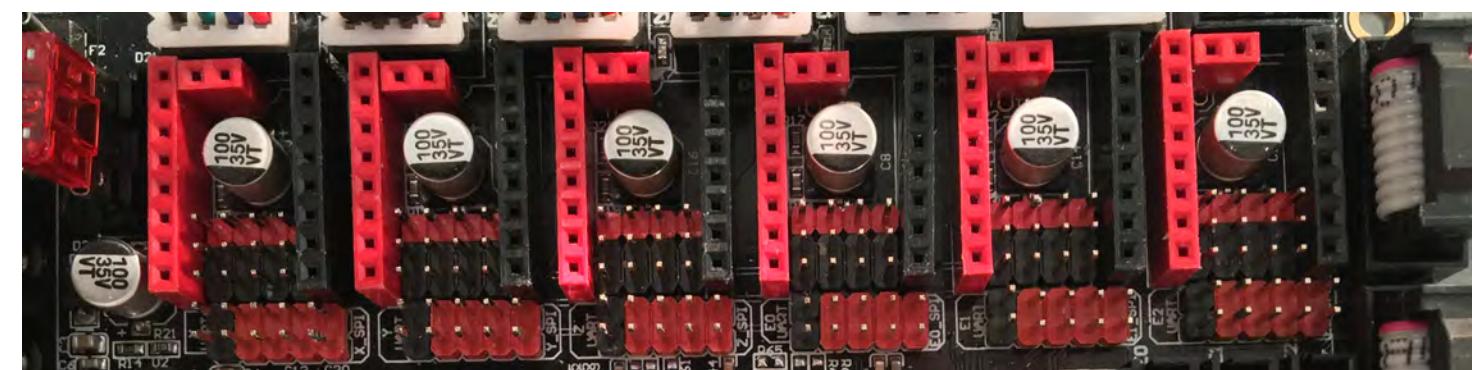
Stand Alone Mode

ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	2 1	M1A
STEP	-	-	1	M1B
DIR	-	-	1	VIO
	-	-	-	GND
1/16				
Interpolation: 1/256				
SpreadCycle				



Stand Alone Mode

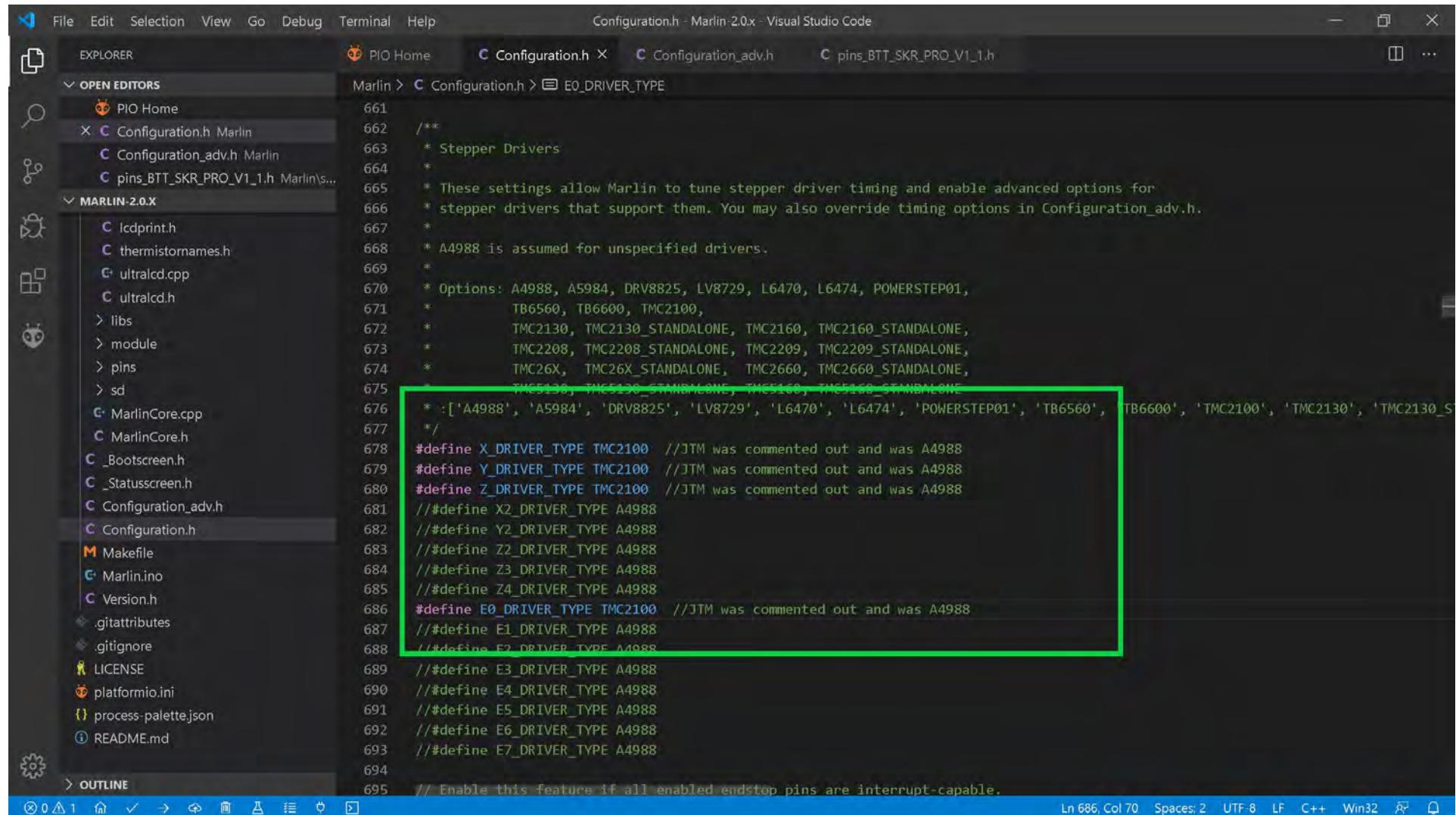
ENA	-	-	-	VM0T
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
	-	-	2 1	M1A
STEP	-	-	2 1	M1B
DIR	-	-	-	VIO
	-	-	-	GND
1/16				
Interpolation: 1/256				
StealthChop				



The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers in Stand Alone Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU MKS TMC2100 stepper motor drivers in stand alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2100 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2100 drivers in stand alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following driver type definitions:

```

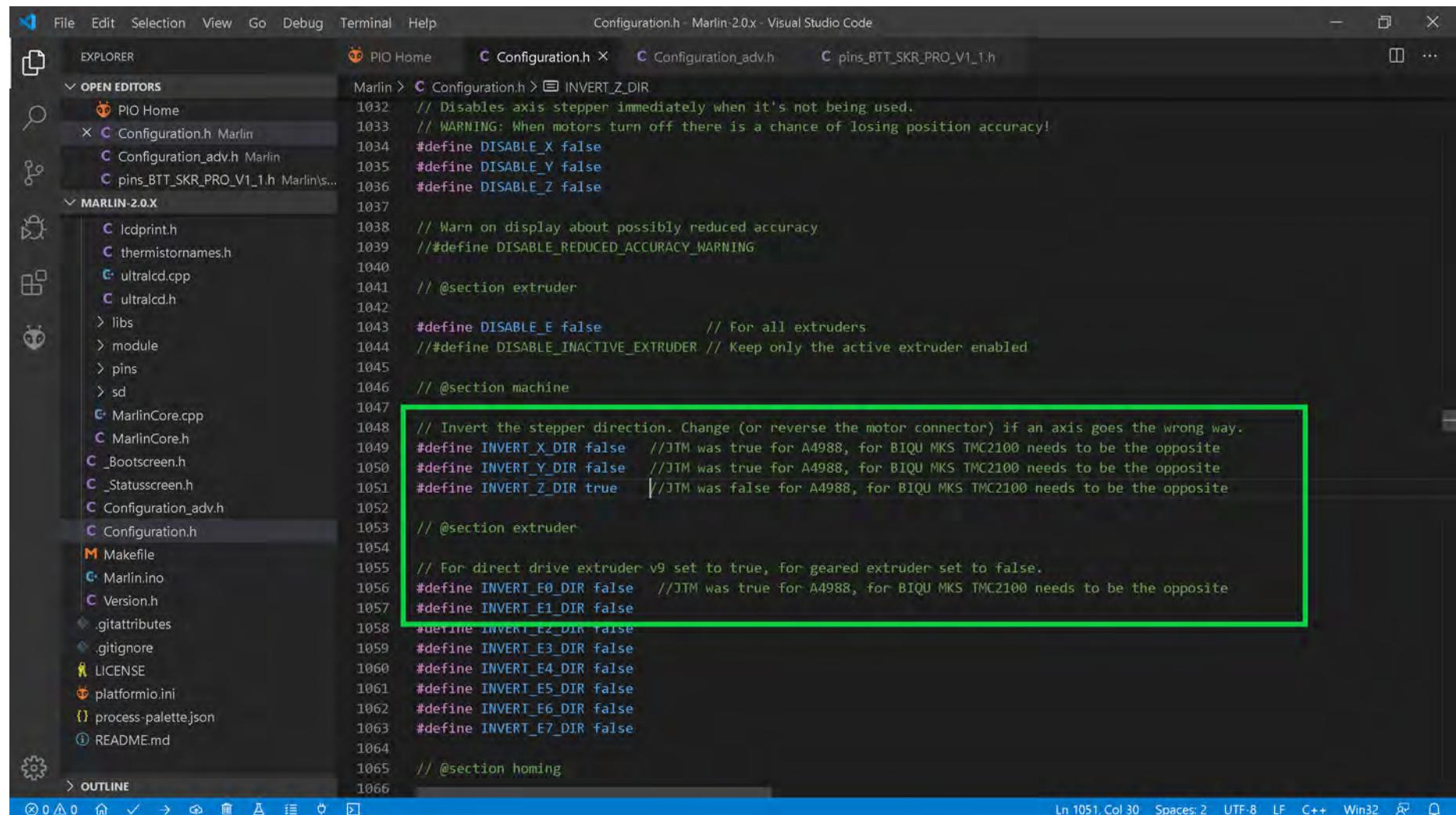
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
676 */
# define X_DRIVER_TYPE TMC2100 //JTM was commented out and was A4988
# define Y_DRIVER_TYPE TMC2100 //JTM was commented out and was A4988
# define Z_DRIVER_TYPE TMC2100 //JTM was commented out and was A4988
// #define X2_DRIVER_TYPE A4988
// #define Y2_DRIVER_TYPE A4988
// #define Z2_DRIVER_TYPE A4988
// #define Z3_DRIVER_TYPE A4988
// #define Z4_DRIVER_TYPE A4988
# define E0_DRIVER_TYPE TMC2100 //JTM was commented out and was A4988
// #define E1_DRIVER_TYPE A4988
// #define E2_DRIVER_TYPE A4988
// #define E3_DRIVER_TYPE A4988
// #define E4_DRIVER_TYPE A4988
// #define E5_DRIVER_TYPE A4988
// #define E6_DRIVER_TYPE A4988
// #define E7_DRIVER_TYPE A4988
694 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers in Stand Alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2100 drivers, I must invert the stepper motor direction because the TMC2100 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2100 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as show in the GREEN box below



Configuration.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Debug Terminal Help
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin Configuration.h INVERT_Z_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for BIQU MKS TMC2100 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for BIQU MKS TMC2100 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for BIQU MKS TMC2100 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // JTM was true for A4988, for BIQU MKS TMC2100 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

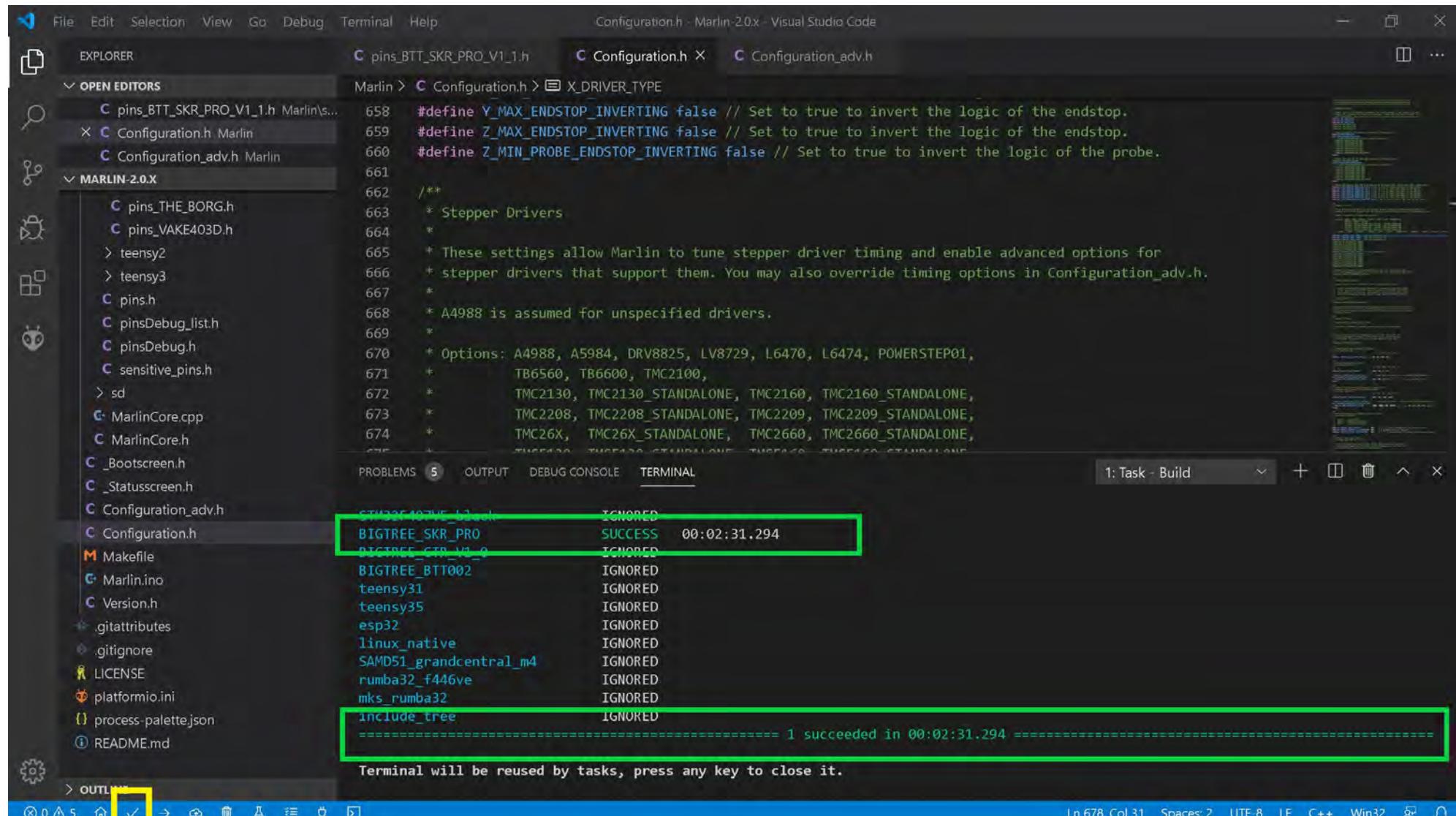
```

Ln 1051, Col 30 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers in Stand Alone Mode

- The end of Marlin setup for BIQU MKS TMC2100 drivers in stand alone mode . Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages ,as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the following details:

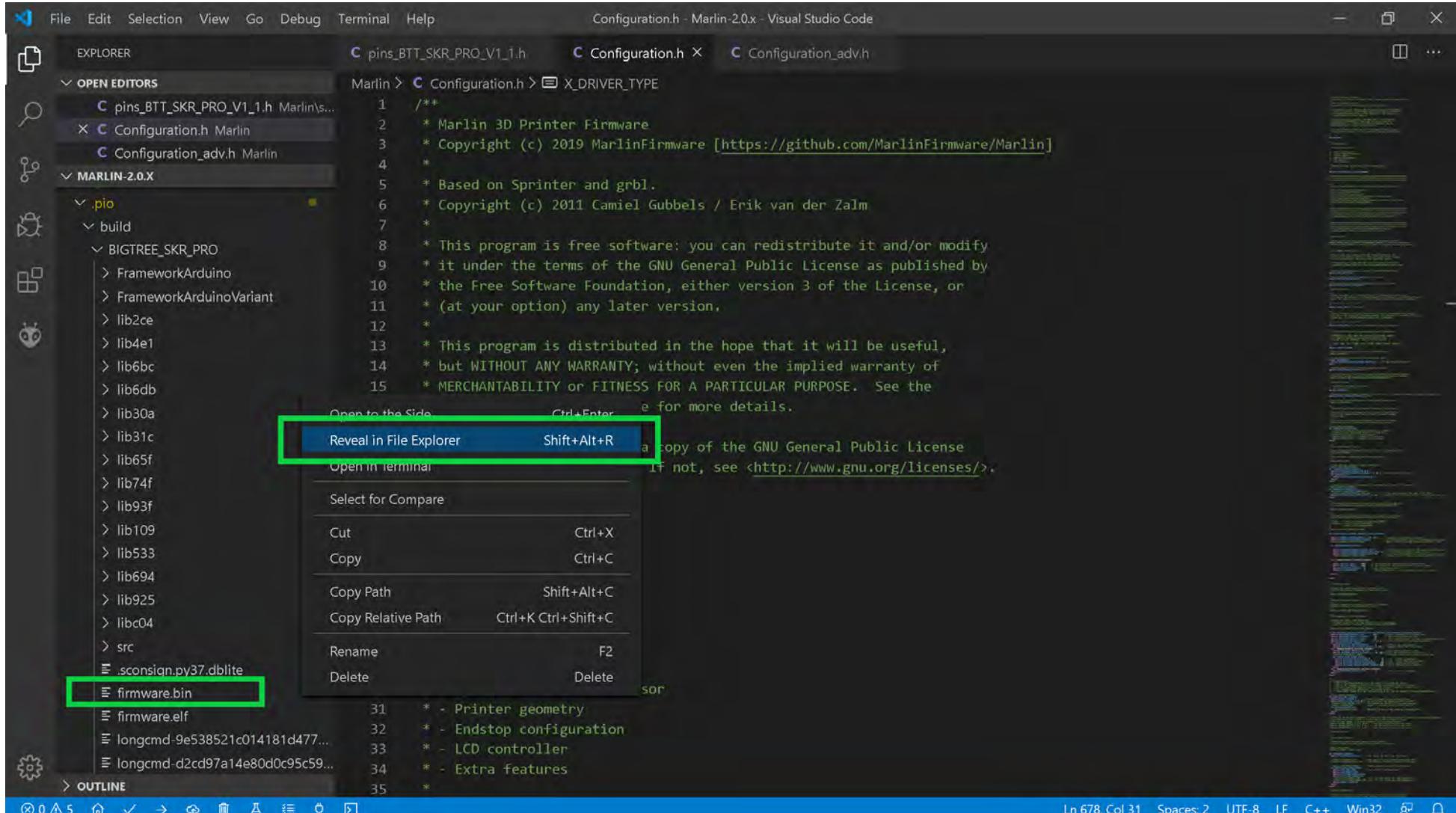
- File Explorer:** Shows the project structure with files like `pins_BTT_SKR_PRO_V1_1.h`, `Configuration.h`, and `Configuration_adv.h`.
- Editor:** Displays the `Configuration.h` file content, specifically the section for stepper drivers and endstop inversion.
- Terminal:** Shows the build log output. A yellow box highlights the first two lines of the log, which are "BIGTREE_SKR_PRO" and "BIGTREE_SKR_V1_0". A green box highlights the entire log output, which includes:


```
STM32F407VE_I2C.o: SUCCESS 00:02:31.294
BIGTREE_SKR_PRO
BIGTREE_SKR_V1_0: IGNORED
BIGTREE_BTT002: IGNORED
teensy31: IGNORED
teensy35: IGNORED
esp32: IGNORED
linux_native: IGNORED
SAMD51_grandcentral_m4: IGNORED
rumba32_f446ve: IGNORED
mks_rumba32: IGNORED
include_tree: IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```
- Status Bar:** Shows the terminal message "Terminal will be reused by tasks, press any key to close it." and the status bar with "Ln 678, Col 31" and other build-related information.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers in Stand Alone Mode

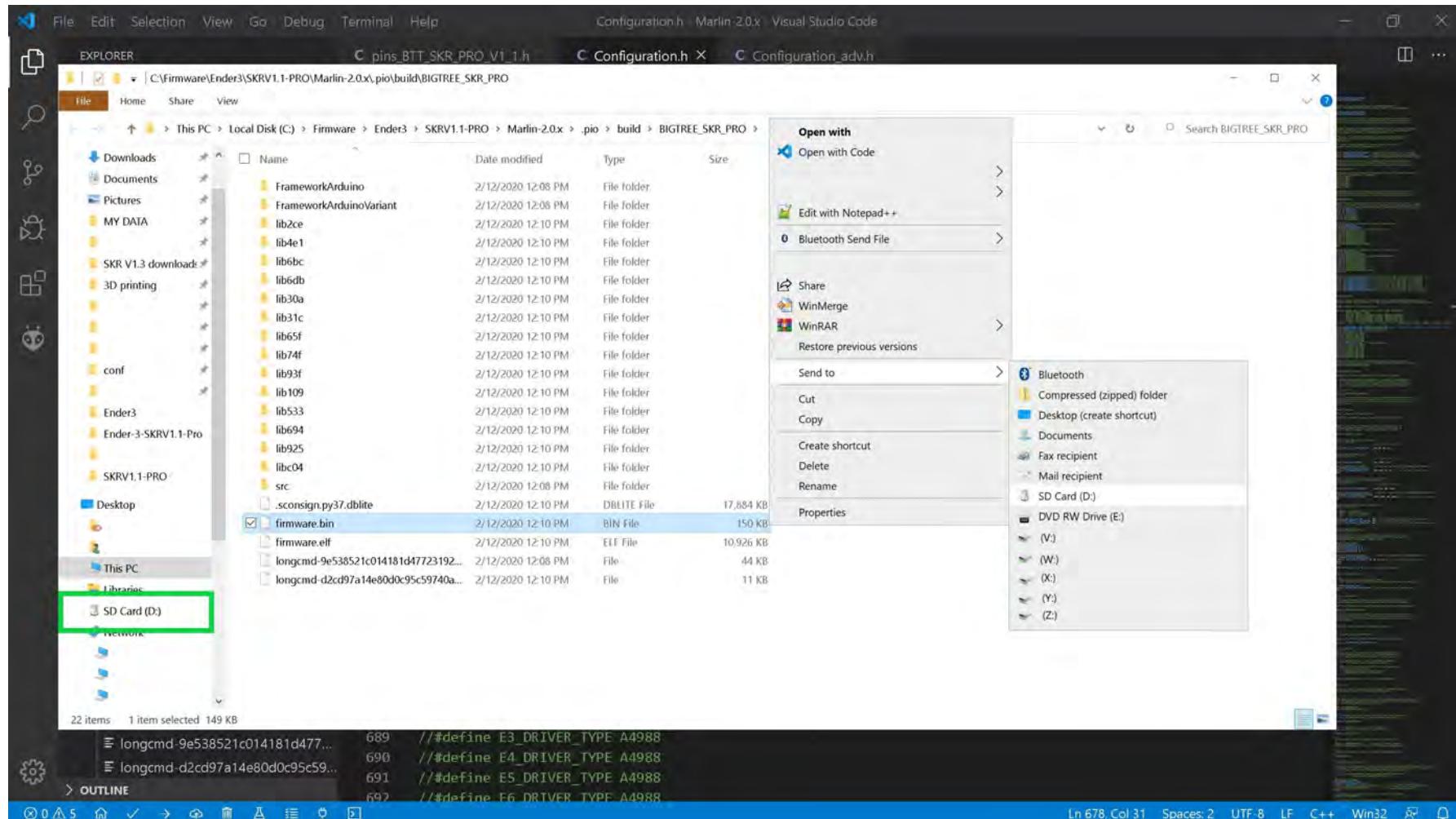
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



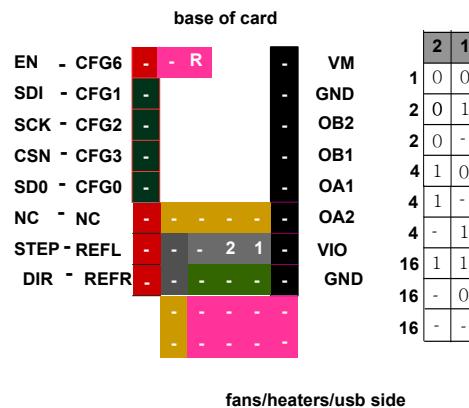
- Go to the next page.

The (latest release of) Marlin Setup for BIQU MKS TMC2100 Drivers in Stand Alone Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2130**Stand Alone Mode****Stand Alone Mode****Driver Chip****TMC2130****Standalone Mode**

Solder the SPI jumper on the bottom of the Driver Board to the adjacent pad to obtain Standalone Mode

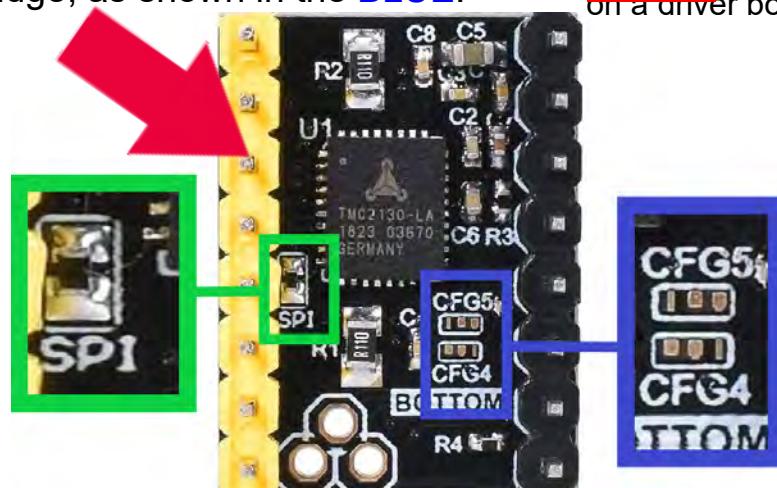
5.5V - 46V DC 1.4A
Maximum subdivision 16

Driving current calculation formula**CFG2****CFG1****Steps****Interpolation****Mode**

GND	GND	1	none	spreadCycle
GND	VIO	1/2	none	spreadCycle
GND	open	1/2	1/256	spreadCycle
VIO	GND	1/4	none	spreadCycle
VIO	open	1/4	1/256	spreadCycle
open	VIO	1/4	1/256	stealthChop
VIO	VIO	1/16	none	spreadCycle
open	GND	1/16	1/256	spreadCycle
open	open	1/16	1/256	stealthChop

To place the BIQU TMC2130 into Standalone Mode:

Solder the SPI Jumper together, on the bottom of the driver board to the adjacent pad, as shown in the **GREEN** box below. Ensure that at CFG4 location and CFG5 location that those pads are NOT soldered together to form a bridge, as shown in the **BLUE**.

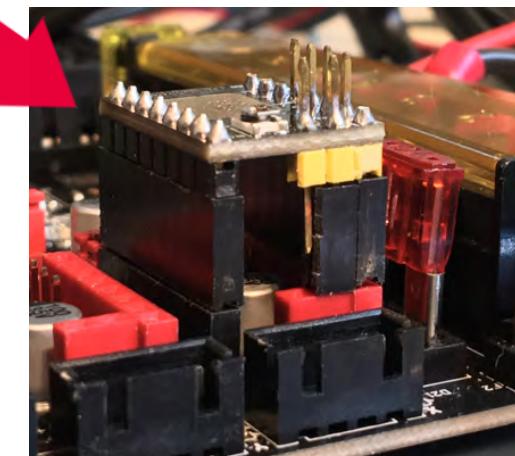


NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in **RED**; OR

use the "Vref Test point" location on the top of the driver board, as shown in **BLUE**. [See the appendix](#) for instructions on how to set the Vref on a driver board



Note: When the stall-guard function is **not used**, the stall-guard pin of the TMC2130 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board

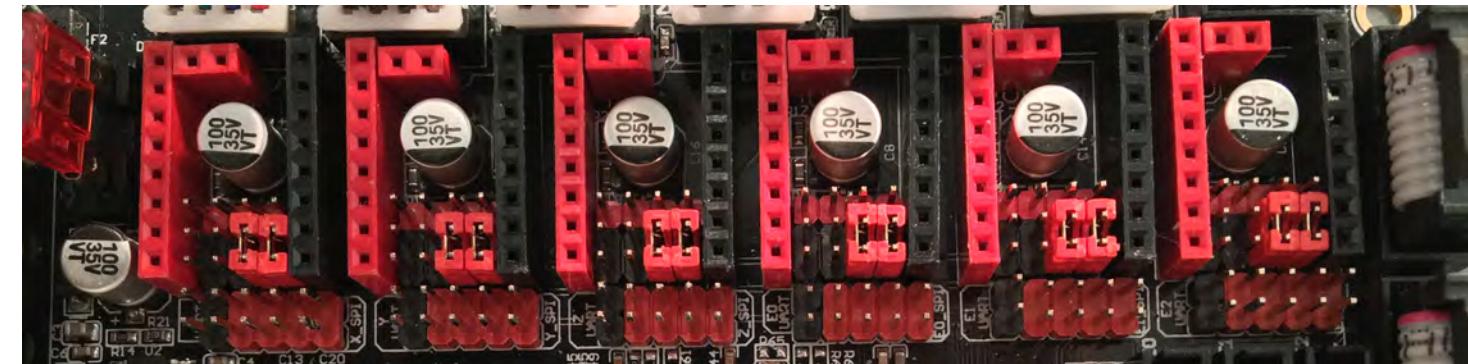


BIQU TMC2130

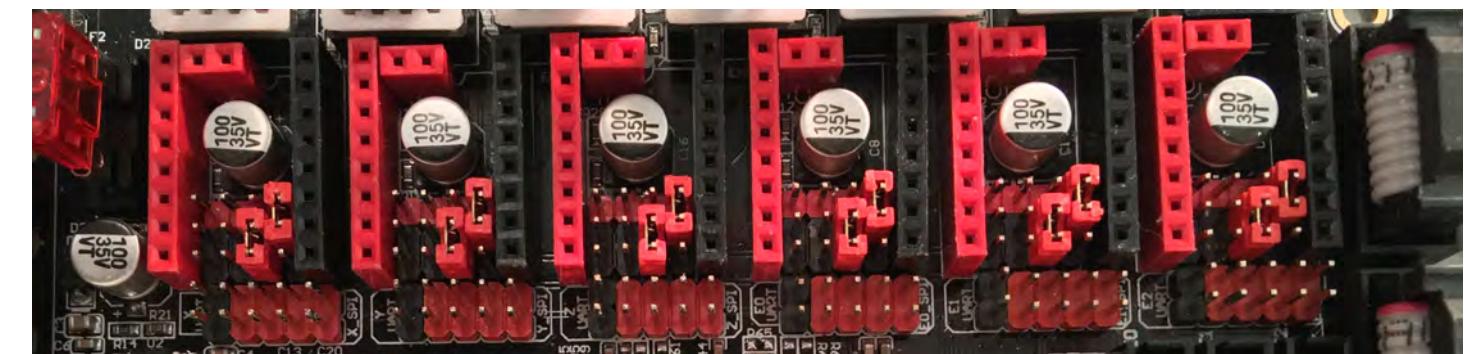
Stand Alone Mode

Stand Alone Mode**Stand Alone Mode**

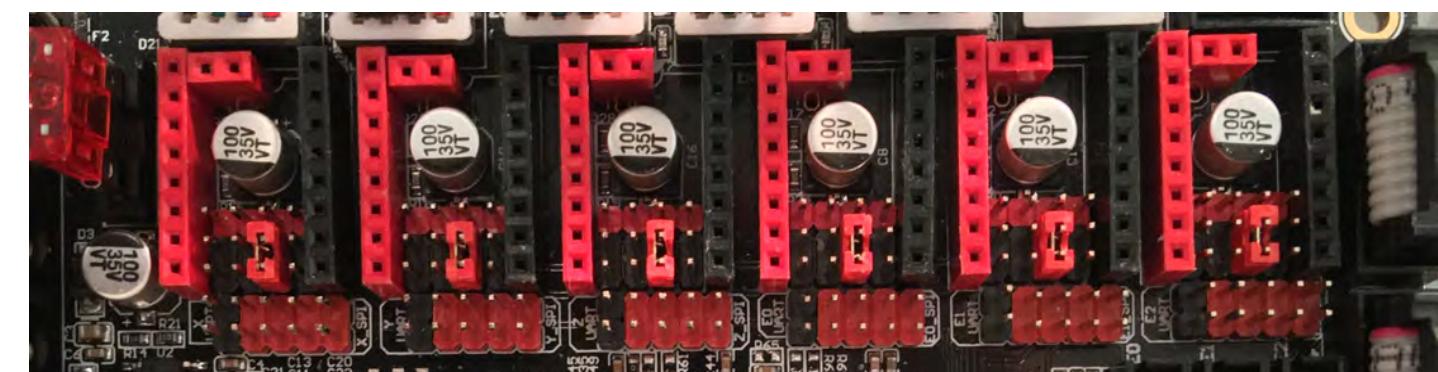
EN - CFG6	-	R	-	VM
SDI - CFG1	-		-	GND
SCK - CFG2	-		-	OB2
CSN - CFG3	-		-	OB1
SD0 - CFG0	-		-	OA1
NC - NC	-	-	-	OA2
STEP - REFL	-	-	2 1	VIO
DIR - REFR	Red	Green	2 1	GND
STEP				
Interpolation: none				
SpreadCycle				

**Stand Alone Mode**

EN - CFG6	-	R	-	VM
SDI - CFG1	-		-	GND
SCK - CFG2	-		-	OB2
CSN - CFG3	-		-	OB1
SD0 - CFG0	-		-	OA1
NC - NC	-	-	1	OA2
STEP - REFL	-	-	2 1	VIO
DIR - REFR	Yellow	Green	2	GND
1/2				
Interpolation: none				
SpreadCycle				

**Stand Alone Mode**

EN - CFG6	-	R	-	VM
SDI - CFG1	-		-	GND
SCK - CFG2	-		-	OB2
CSN - CFG3	-		-	OB1
SD0 - CFG0	-		-	OA1
NC - NC	-	-	-	OA2
STEP - REFL	-	-	2	VIO
DIR - REFR	Red	Green	2	GND
1/2				
Interpolation: 1/256				
SpreadCycle				



NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

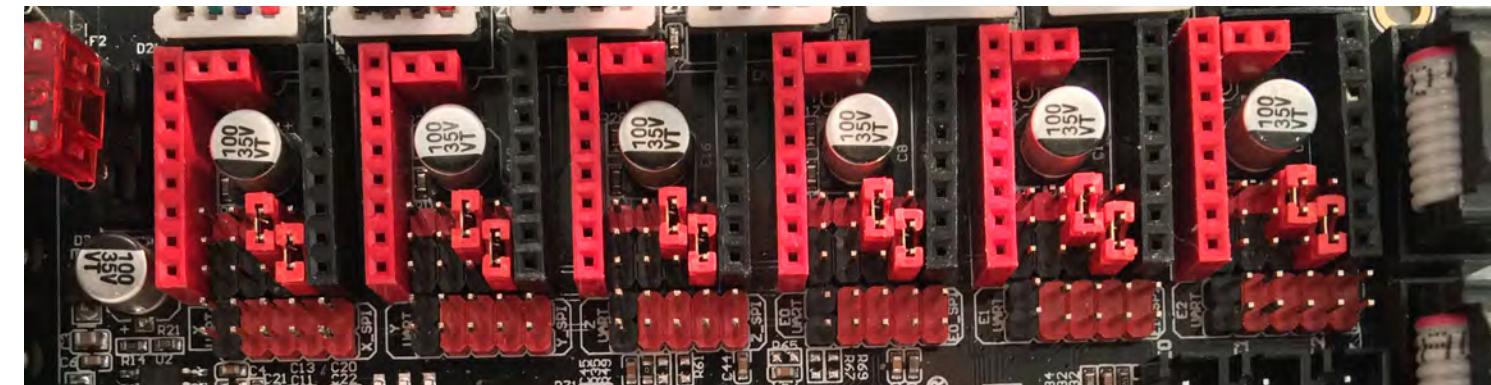
BIQU TMC2130

Stand Alone Mode

Stand Alone Mode**Stand Alone Mode**

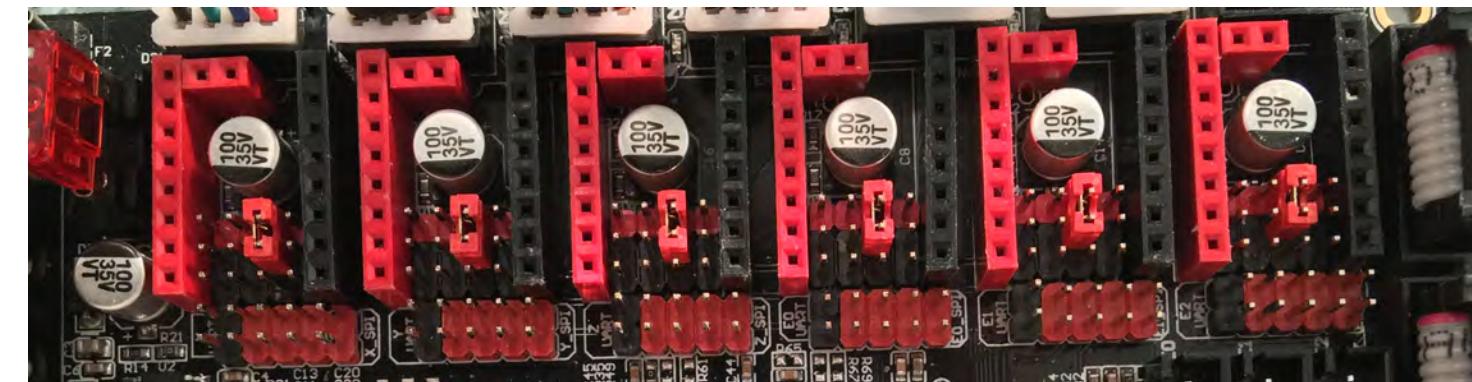
EN - CFG6	-	R	-	VM
SDI - CFG1	-	-	-	GND
SCK - CFG2	-	-	-	OB2
CSN - CFG3	-	-	-	OB1
SD0 - CFG0	-	-	-	OA1
NC - NC	-	2	-	OA2
STEP - REFL	-	2	1	VIO
DIR - REFR	-	4	-	GND

1/4
Interpolation: none
SpreadCycle

**Stand Alone Mode**

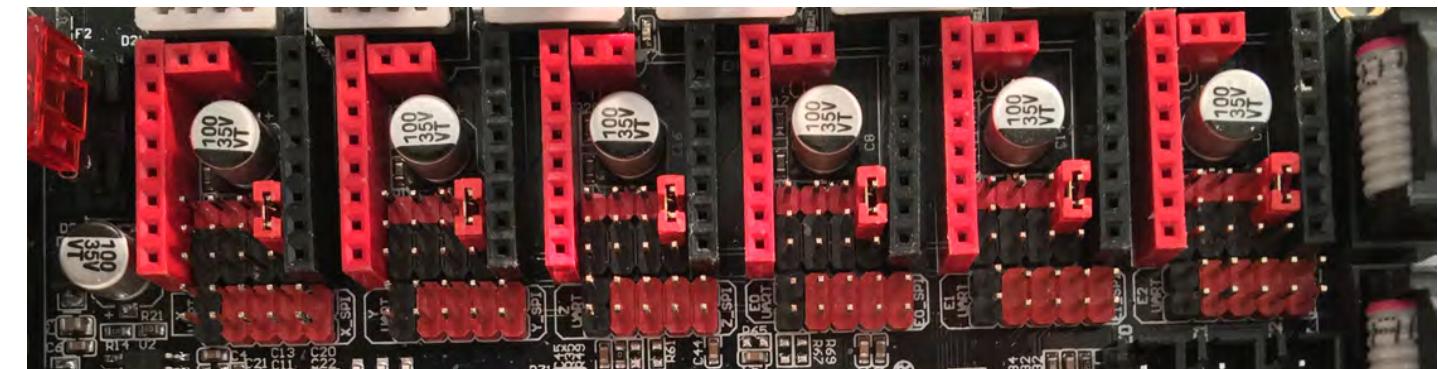
EN - CFG6	-	R	-	VM
SDI - CFG1	-	-	-	GND
SCK - CFG2	-	-	-	OB2
CSN - CFG3	-	-	-	OB1
SD0 - CFG0	-	-	-	OA1
NC - NC	-	2	-	OA2
STEP - REFL	-	2	-	VIO
DIR - REFR	-	-	-	GND

1/4
Interpolation: 1/256
SpreadCycle

**Stand Alone Mode**

EN - CFG6	-	R	-	VM
SDI - CFG1	-	-	-	GND
SCK - CFG2	-	-	-	OB2
CSN - CFG3	-	-	-	OB1
SD0 - CFG0	-	-	-	OA1
NC - NC	-	-	1	OA2
STEP - REFL	-	-	1	VIO
DIR - REFR	-	-	-	GND

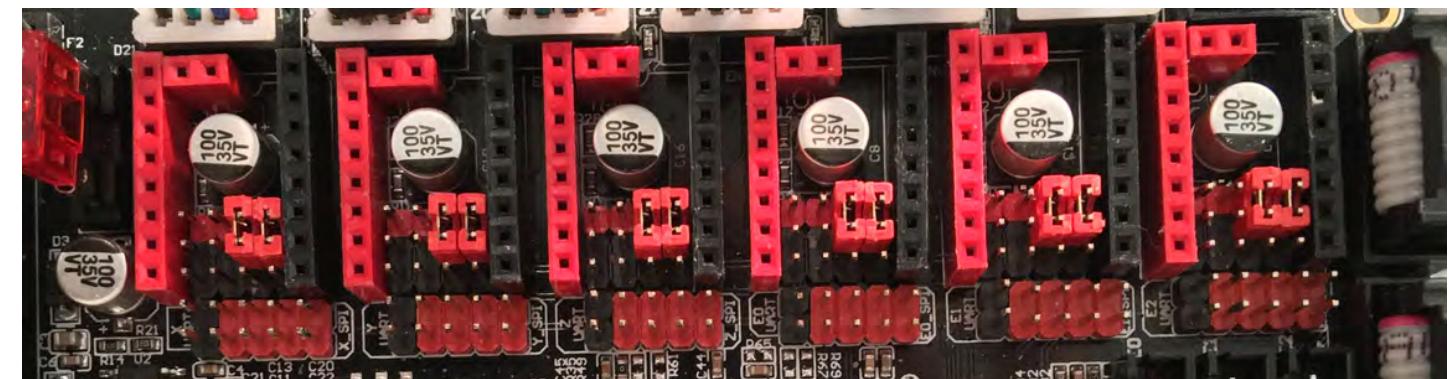
1/4
Interpolation: 1/256
StealthChop



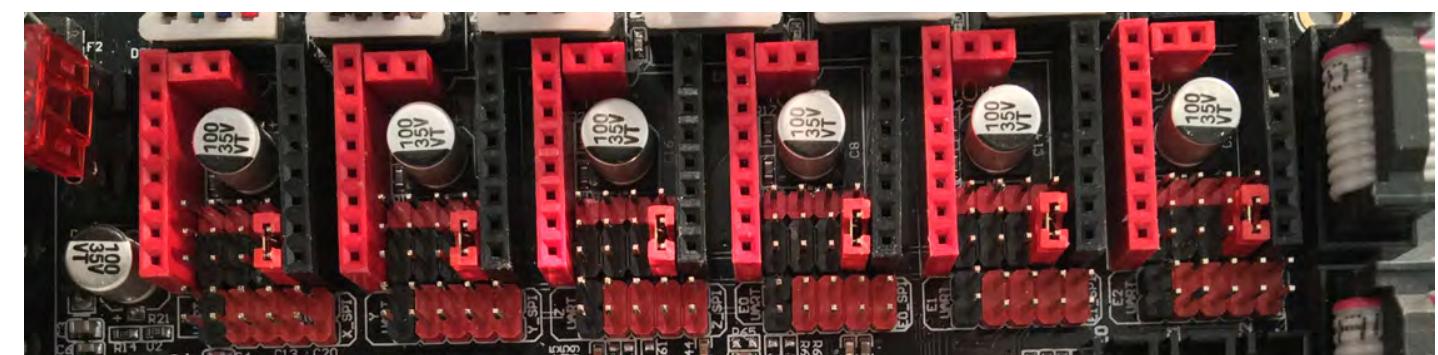
NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

BIQU TMC2130**Stand Alone Mode****Stand Alone Mode**

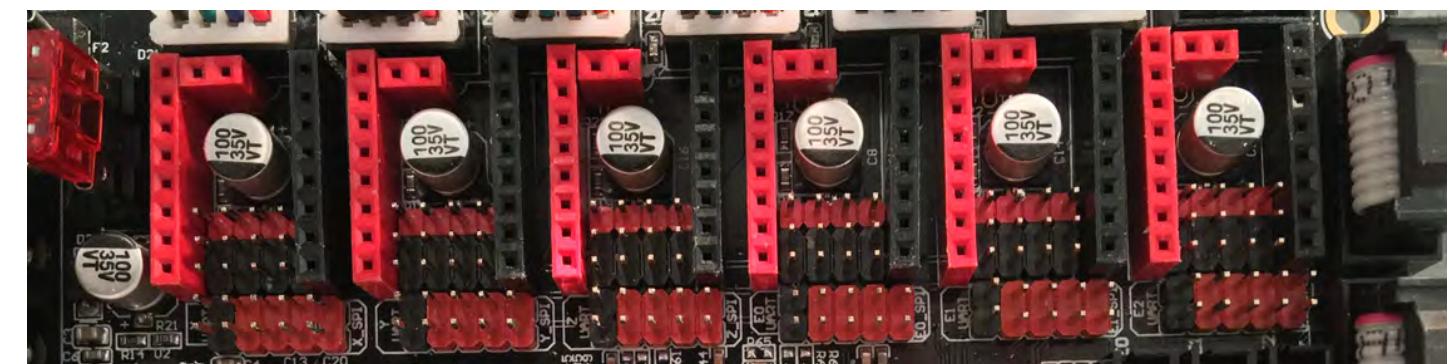
Stand Alone Mode	
EN - CFG6	- R
SDI - CFG1	-
SCK - CFG2	-
CSN - CFG3	-
SD0 - CFG0	-
NC - NC	- 2 2 -
STEP - REFL	- 2 1 -
DIR - REFR	- - -
1/16	
Interpolation: none	
SpreadCycle	

**Stand Alone Mode**

Stand Alone Mode	
EN - CFG6	- R
SDI - CFG1	-
SCK - CFG2	-
CSN - CFG3	-
SD0 - CFG0	-
NC - NC	- - -
STEP - REFL	- - 1 -
DIR - REFR	- - -
1/16	
Interpolation: 1/256	
SpreadCycle	

**Stand Alone Mode**

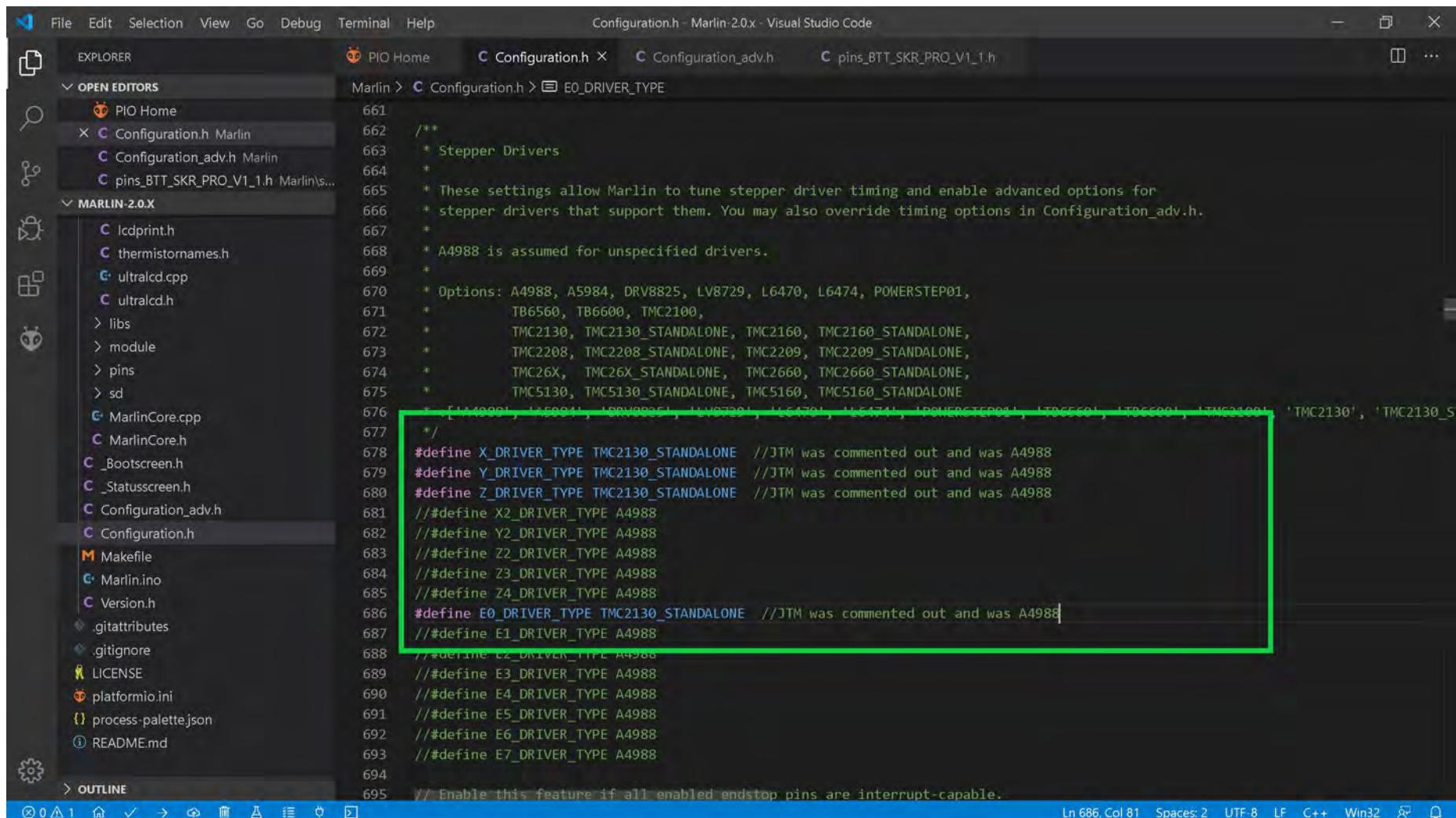
Stand Alone Mode	
EN - CFG6	- R
SDI - CFG1	-
SCK - CFG2	-
CSN - CFG3	-
SD0 - CFG0	-
NC - NC	- - -
STEP - REFL	- - -
DIR - REFR	- - -
1/16	
Interpolation: 1/256	
StealthChop	



The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2130 stepper motor drivers in stand alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2130 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in stand alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code defines driver types for various axes and extruders using preprocessor directives. A green box highlights the section where driver types are defined, specifically the `#define` statements for `X_DRIVER_TYPE`, `Y_DRIVER_TYPE`, `Z_DRIVER_TYPE`, `E0_DRIVER_TYPE`, and `E1_DRIVER_TYPE`. These lines were previously commented out with double slashes (//).

```

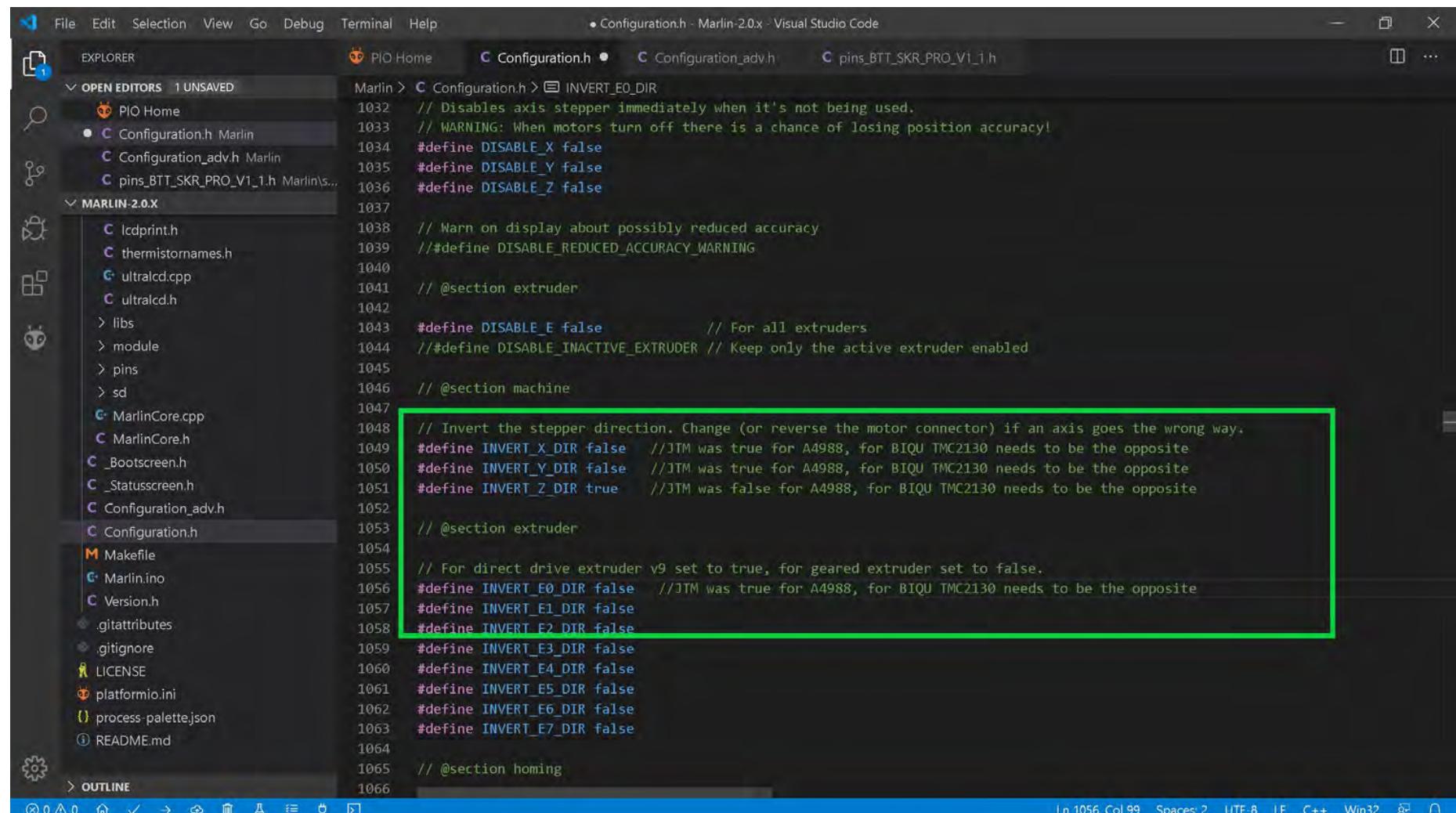
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * [A4988], [A5984], [DRV8825], [LV8729], [L6470], [L6474], [POWERSTEP01], [TB6560], [TB6600], [TMC2100], [TMC2130], [TMC2160]
677 */
678 #define X_DRIVER_TYPE TMC2130_STANDALONE //JTM was commented out and was A4988
679 #define Y_DRIVER_TYPE TMC2130_STANDALONE //JTM was commented out and was A4988
680 #define Z_DRIVER_TYPE TMC2130_STANDALONE //JTM was commented out and was A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2130_STANDALONE //JTM was commented out and was A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as show in the GREEN box below



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a dark theme. The file contains C++ code for Marlin 2.0.x. A green rectangular box highlights a specific section of the code:

```

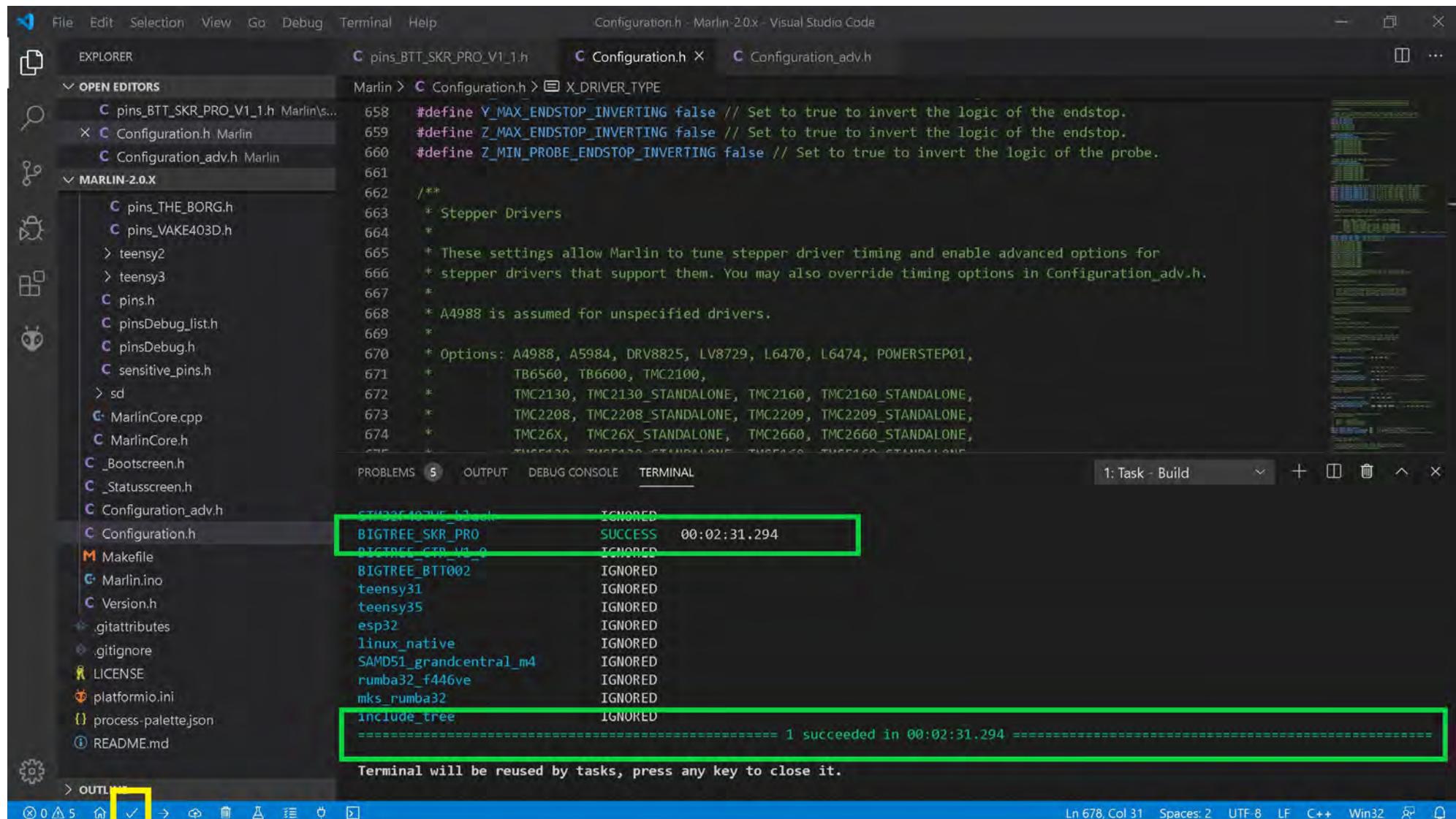
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for BIQU TMC2130 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false    // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode

- The end of Marlin setup for BIQU TMC2130 drivers in stand alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins\pins_BTT_SKR_PRO_V1_1.h
Configuration.h Marlin\src\configuration\configuration.h
Configuration_adv.h Marlin\src\configuration\configuration_adv.h
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
Task: Task - Build
 1 succeeded in 00:02:31.294
Terminal will be reused by tasks, press any key to close it.

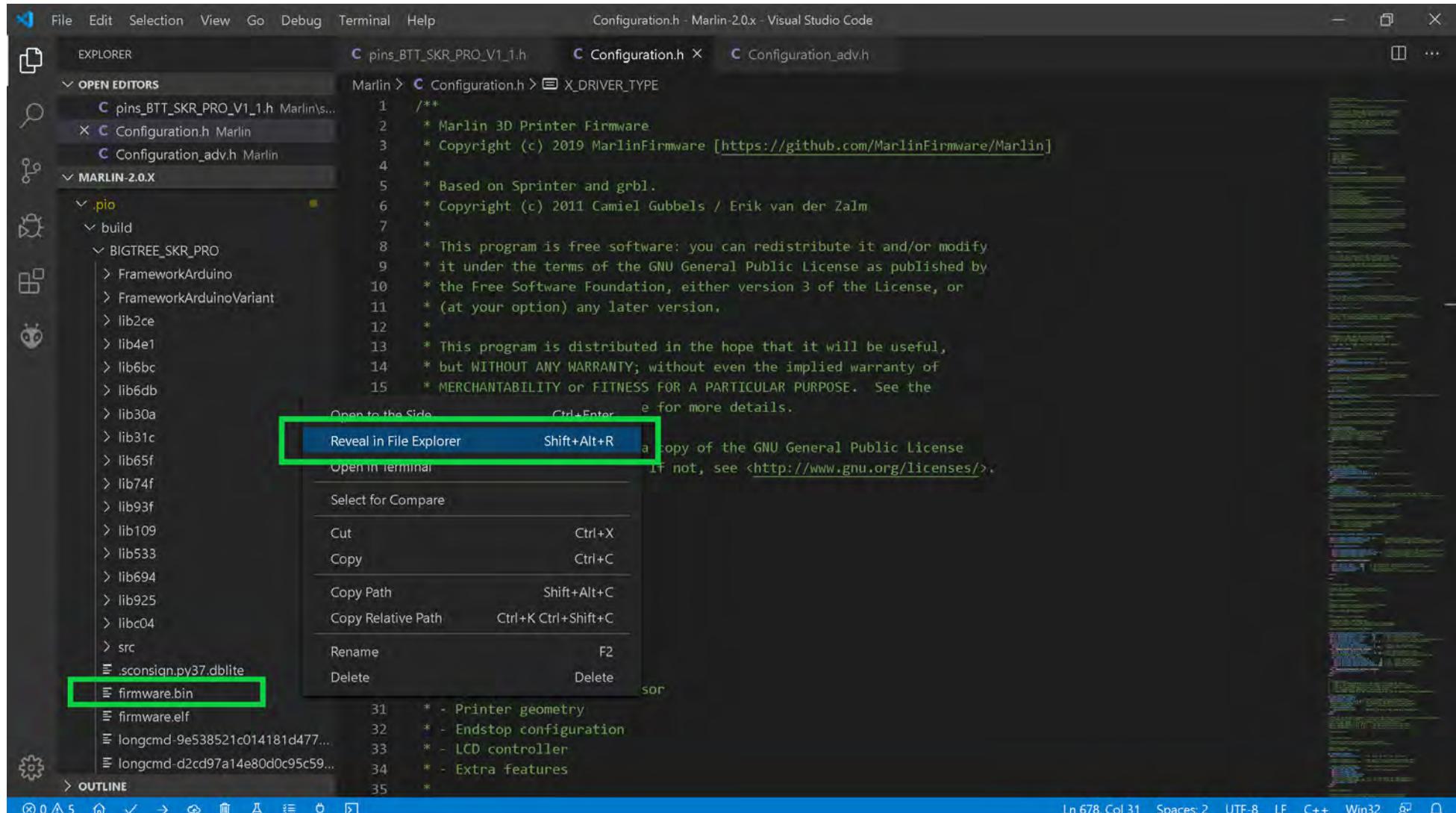
```

The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab selected. The terminal window displays the build process for Marlin 2.0.x. A yellow box highlights the terminal output, and a green box highlights the success message at the bottom.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode

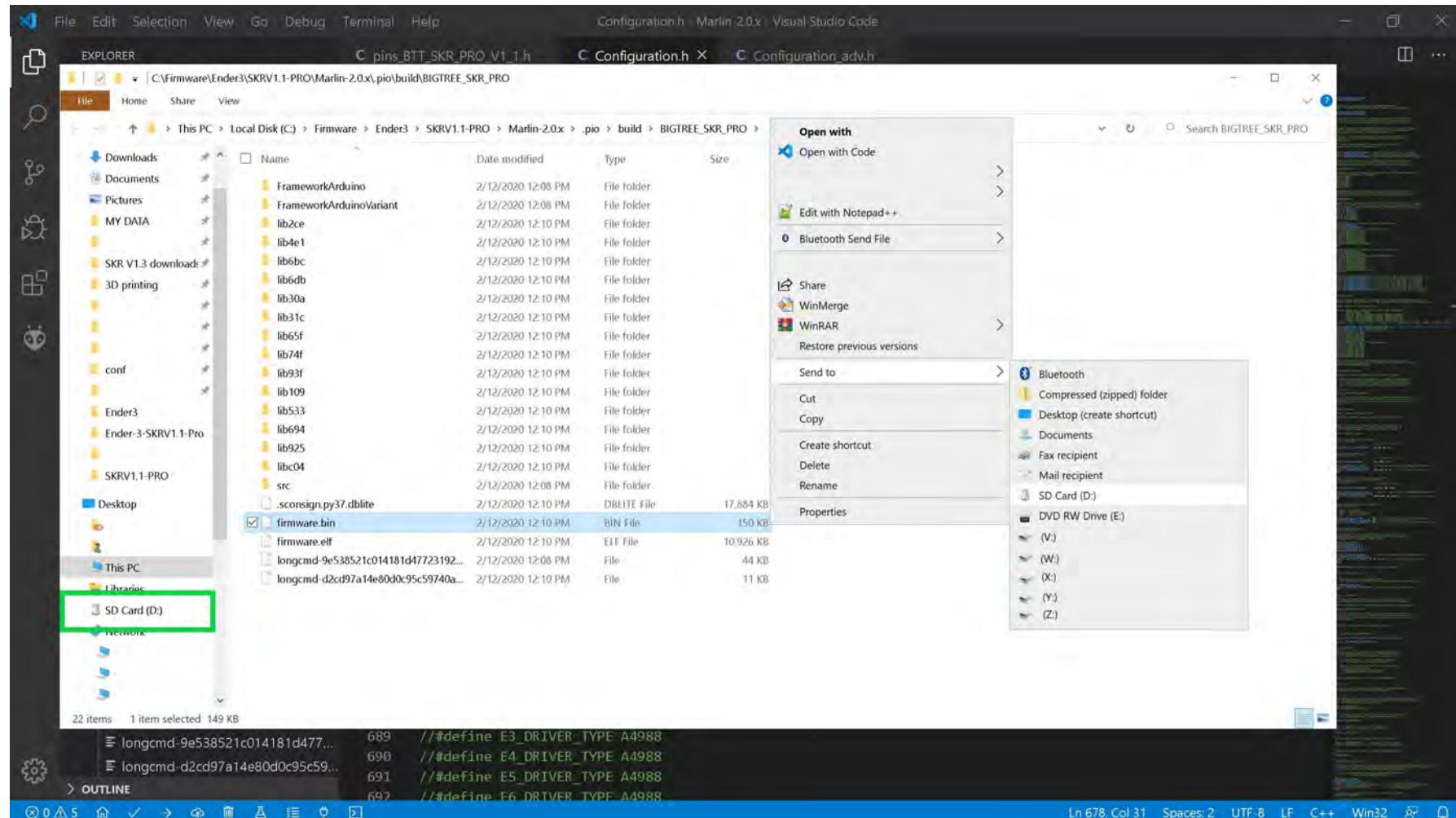
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in Stand Alone Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



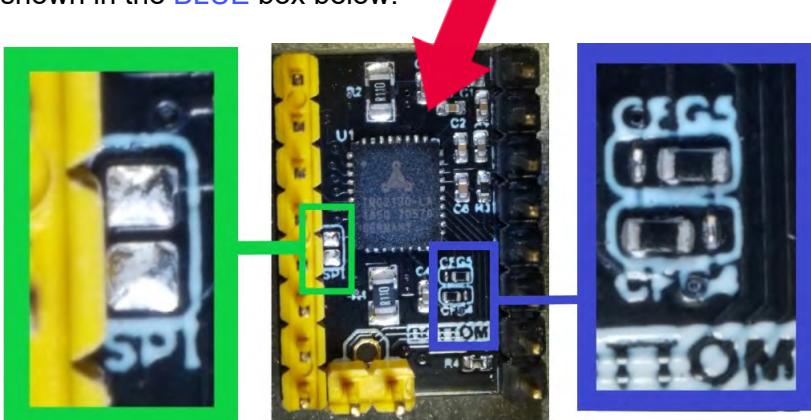
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2130SPI Mode**SPI****R - Stall Detection**

base of card		VM
EN - CFG6	- R	GND
SDI - CFG1	-	OB2
SCK - CFG2	-	OB1
CSN - CFG3	-	OA1
SD0 - CFG0	-	OA2
NC - NC	-	VIO
STEP - REFL	-	GND
DIR - REFR	-	
		SPI M
		SPI M
fans/heaters/usb side		

To place the BIQU TMC2130 into SPI Mode:

The SPI Jumper MUST have a gap between the two SPI pads, on the bottom of the driver board, as shown in the GREEN box below. Ensure that at CFG4 location and CFG5 location that the correct two pads are soldered together to form a bridge, as shown in the BLUE box below.

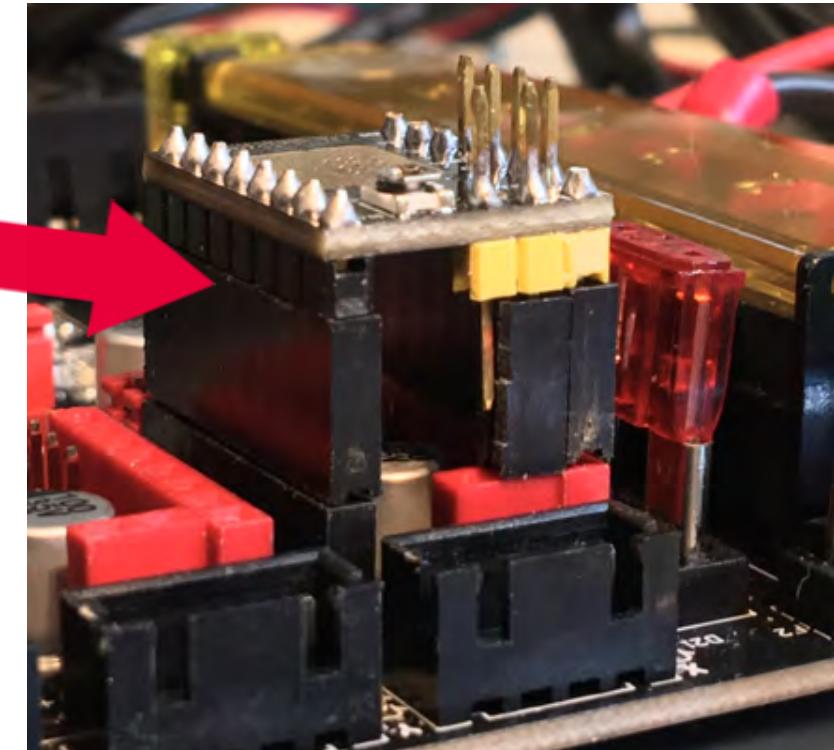


Note: You can use 50% to 90% of the calculated Irms ($I_{max}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

Steps are set inside of your Firmware

Driver Chip	TMC2130 SPI Mode SPI jumper (two pads) on the bottom of the Driver board MUST BE OPEN to obtain SPI Mode Maximum subdivision 256 5.5v - 46V DC 1.4A	$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$	$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$
-------------	--	---	---

Note: When the stall-guard function is **not used**, the stall-guard pin of the TMC2130 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board

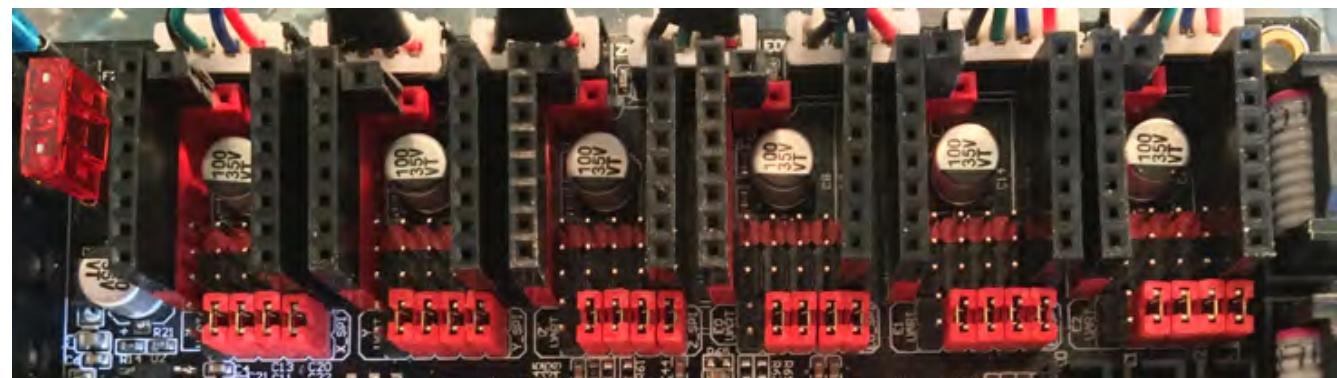


BIQU TMC2130**SPI Mode****SPI**

R - Stall Detection

SPI	
EN - CFG6	- R
SDI - CFG1	- GND
SCK - CFG2	- OB2
CSN - CFG3	- OB1
SD0 - CFG0	- OA1
NC - NC	- OA2
STEP - REFL	- VIO
DIR - REFR	- GND
S P I M	
S P I M	

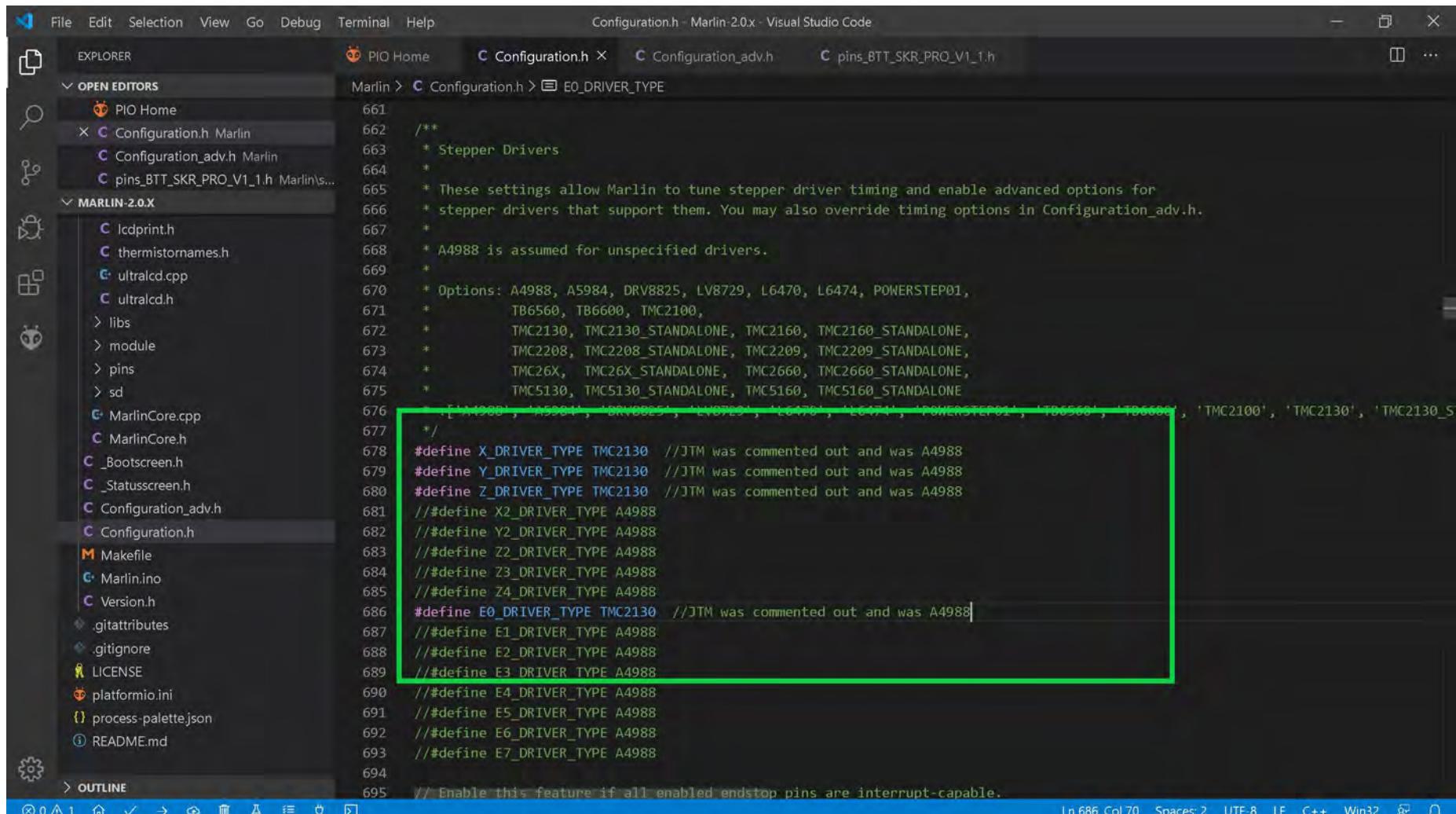
Note: Set JUMPERS "S","P","I","M" on the board!!



The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2130 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2130 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in SPI mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 *           TB6560, TB6600, TMC2100,
671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 *           TMC5160_STANDALONE, TMC5160_STANDALONE
676 *           TMC5160_STANDALONE, TMC5160_STANDALONE
677 */
#define X_DRIVER_TYPE TMC2130 //JTM was commented out and was A4988
#define Y_DRIVER_TYPE TMC2130 //JTM was commented out and was A4988
#define Z_DRIVER_TYPE TMC2130 //JTM was commented out and was A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2130 //JTM was commented out and was A4988
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

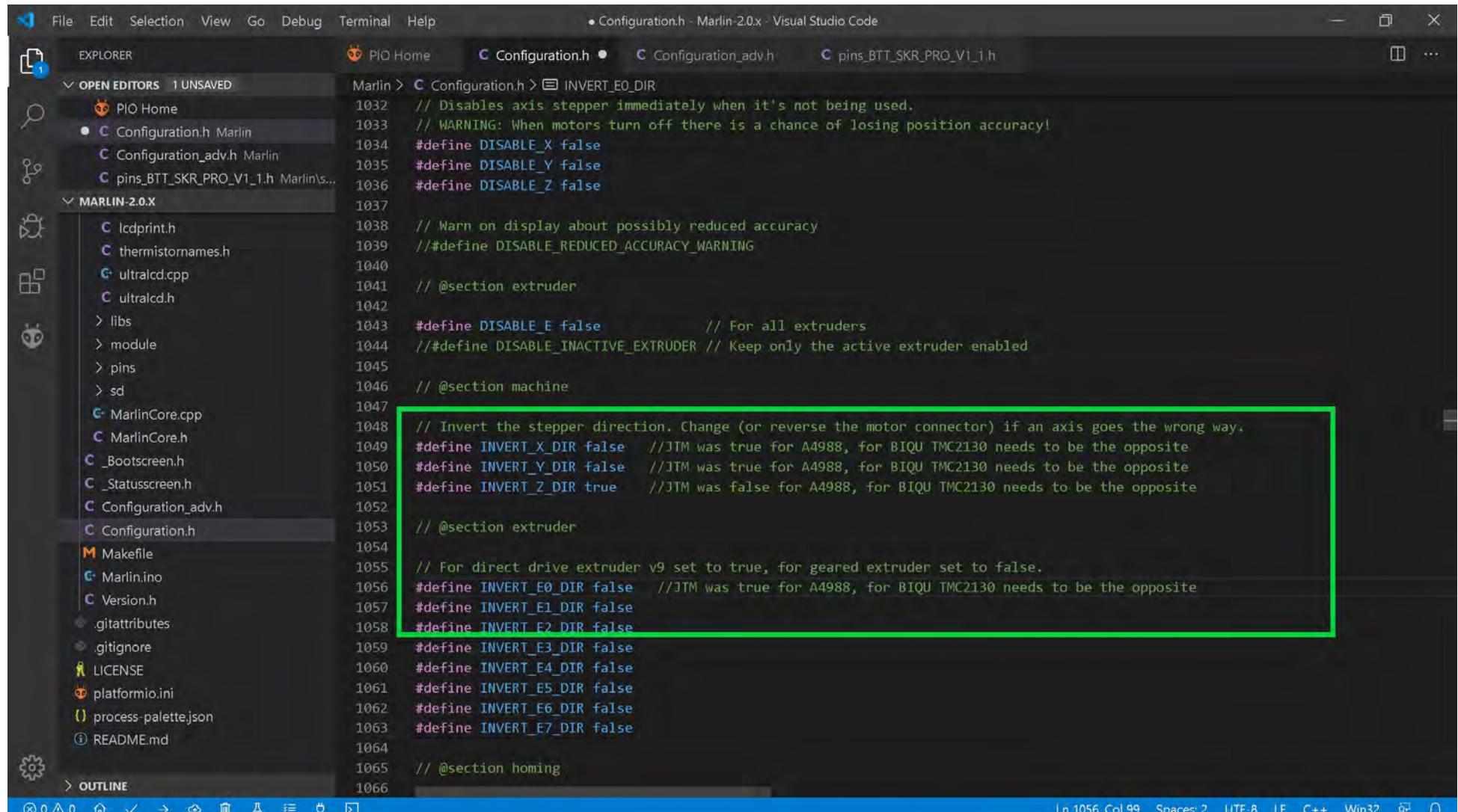
```

A green rectangular box highlights the driver type definitions for X, Y, Z, and E0 axes, specifically the lines starting with `#define`. These lines were previously commented out with double slashes (//).

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

MARLIN-2.0.X

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false

1037 // Warn on display about possibly reduced accuracy
1038 // #define DISABLE_REDUCED_ACCURACY_WARNING
1039
1040 // @section extruder
1041
1042 // #define DISABLE_E false          // For all extruders
1043 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1044
1045 // @section machine
1046
1047 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1048 #define INVERT_X_DIR false        // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1049 #define INVERT_Y_DIR false        // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1050 #define INVERT_Z_DIR true         // JTM was false for A4988, for BIQU TMC2130 needs to be the opposite
1051
1052 // @section extruder
1053
1054 // For direct drive extruder v9 set to true, for geared extruder set to false.
1055 #define INVERT_E0_DIR false       // JTM was true for A4988, for BIQU TMC2130 needs to be the opposite
1056 #define INVERT_E1_DIR false
1057 #define INVERT_E2_DIR false
1058
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

Ln 1056, Col 99 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- next you want to set your Vref in the Marlin firmware for each axis that has the TMC2130 driver, as seen in the GREEN box below. I changed the "X_CURRENT" to be the calculated Vref for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated Vref for my Y-Axis, which is 760mV on the Ender 3.

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration_adv.h > Y_MICROSTEPS
1994 * TMCSstepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCSstepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003     #define X_CURRENT          760      // (mA) RMS current. Multiply by 1.414 for peak current. //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2004     #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005     #define X_MICROSTEPS       16       // 0..256 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2006     #define X_RSENSE            0.11
2007     #define X_CHAIN_POS         -1      // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011     #define X2_CURRENT         800
2012     #define X2_CURRENT_HOME   X2_CURRENT
2013     #define X2_MICROSTEPS      16
2014     #define X2_RSENSE           0.11
2015     #define X2_CHAIN_POS        -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019     #define Y_CURRENT          760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2020     #define Y_CURRENT_HOME    Y_CURRENT
2021     #define Y_MICROSTEPS       16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2022     #define Y_RSENSE            0.11
2023     #define Y_CHAIN_POS         -1
2024 #endif
2025
2026 #if AXIS_IS_TMC(Y2)
2027     #define Y2_CURRENT         800
2028     #define Y2_CURRENT_HOME   Y2_CURRENT

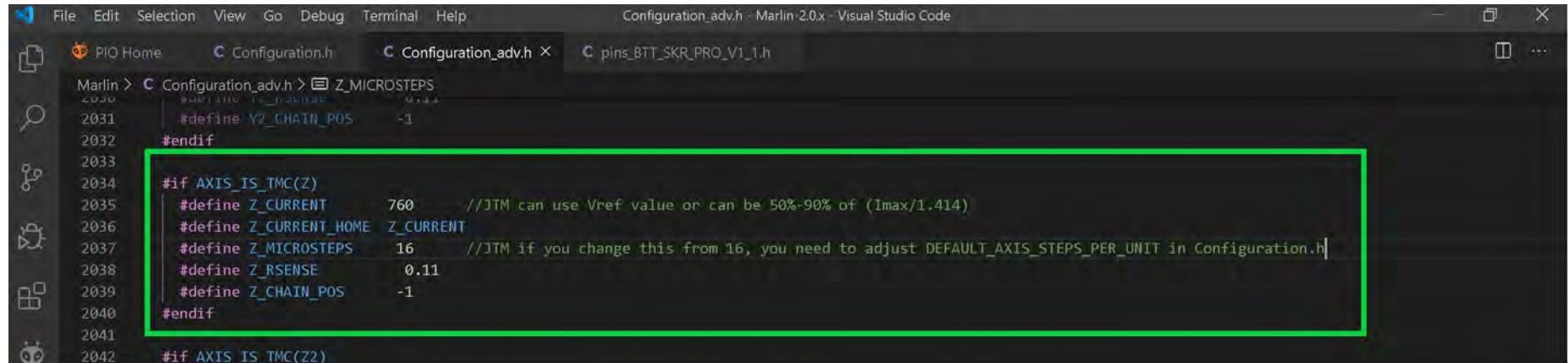
```

Ln 2021, Col 137 Spaces:2 UTF-8 LF C++ Win32 ⌂ 1

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

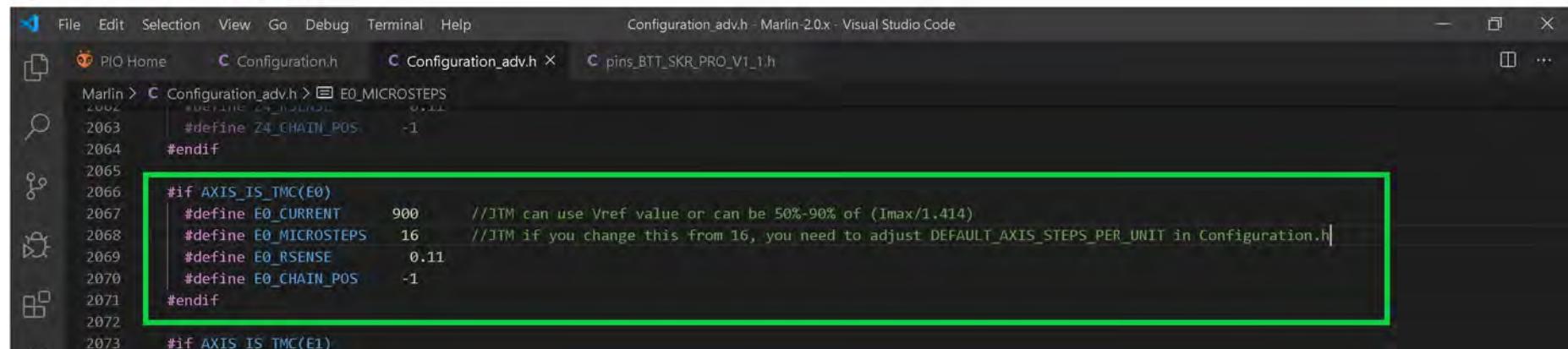
- Now, I am setting the Vref for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated Vref for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated Vref for my Extruder, which is 900mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > Z_MICROSTEPS
2030
2031     #define Z2_CHAIN_POS -1
2032
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT    760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2038         #define Z_RSENSE   0.11
2039         #define Z_CHAIN_POS -1
2040     #endif
2041
2042     #if AXIS_IS_TMC(Z2)

```

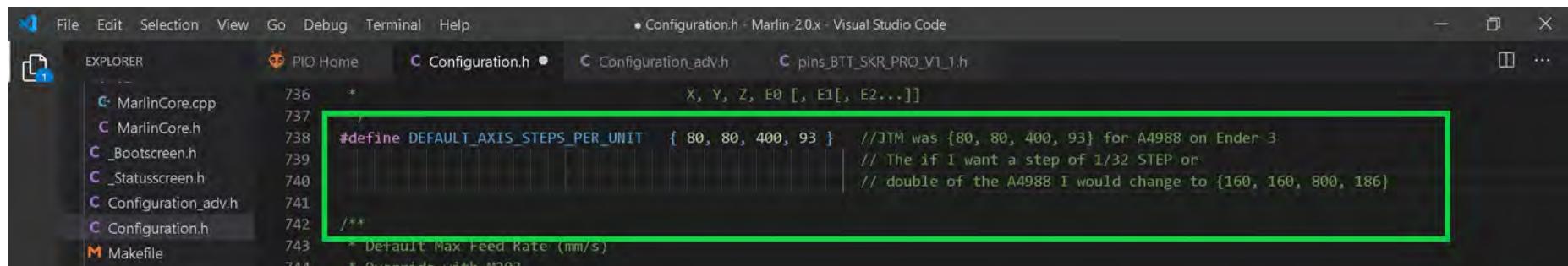


```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > E0_MICROSTEPS
2062
2063     #define Z4_CHAIN_POS -1
2064
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT    900      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2068         #define E0_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2069         #define E0_RSENSE   0.11
2070         #define E0_CHAIN_POS -1
2071     #endif
2072
2073     #if AXIS_IS_TMC(E1)

```

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



```

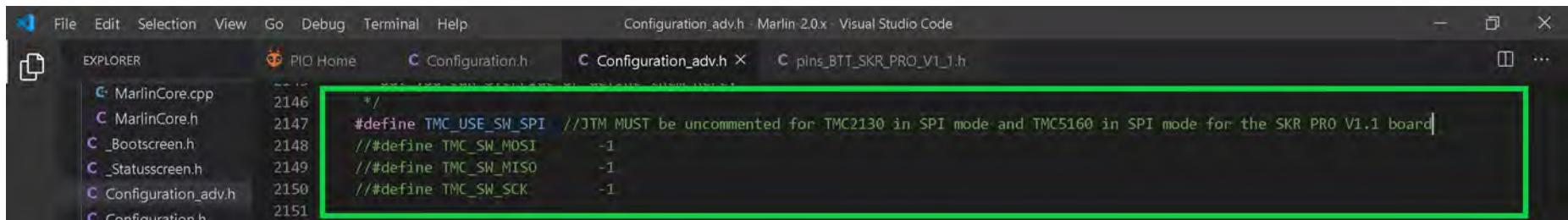
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
736     *
737     /
738     #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
739                                         // The if I want a step of 1/32 STEP or
740                                         // double of the A4988 I would change to {160, 160, 800, 186}
741
742     /**
743     * Default Max Feed Rate (mm/s)
744     * Override with M203

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- We need to uncomment out the "TMC_USE_SW_SPI" because the SKR PRO V1.1 pins file depends on this variable to define it's SPI pins



```

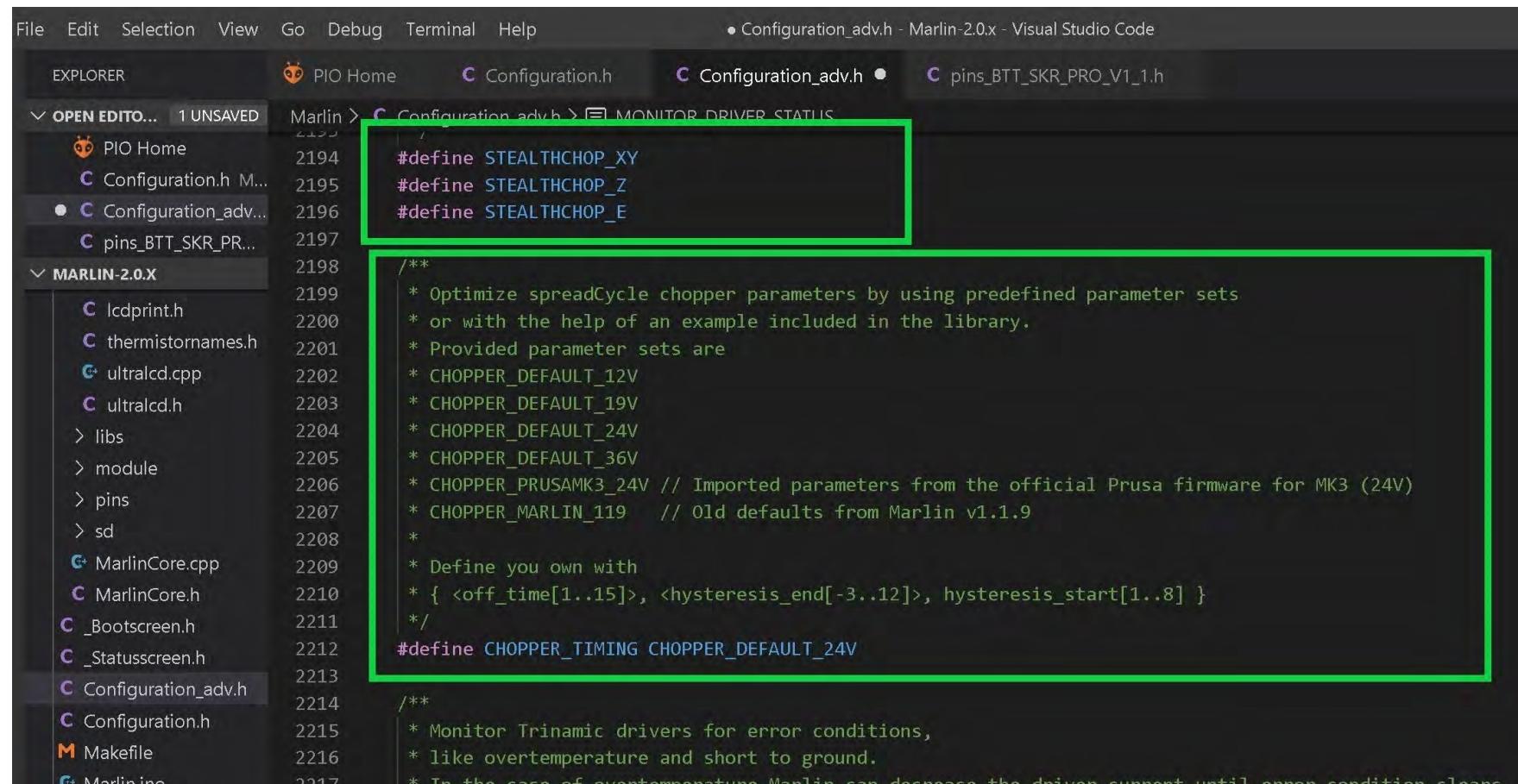
File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home C Configuration.h C Configuration_adv.h C pins_BTT_SKR_PRO_V1_1.h

2146 */
2147 #define TMC_USE_SW_SPI //JTM MUST be uncommented for TMC2130 in SPI mode and TMC5160 in SPI mode for the SKR PRO V1.1 board
2148 //#define TMC_SW_MOSI -1
2149 //#define TMC_SW_MISO -1
2150 //#define TMC_SW_SCK -1
2151

```

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I **want stealthChop enabled** so I want to make sure the the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if the proper "CHOPPER_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER_TIMING" is correct.



```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home C Configuration.h C Configuration_adv.h C pins_BTT_SKR_PRO_V1_1.h

OPEN EDITO... 1 UNSAVED Marlin > C Configuration_adv.h > MONITOR_DRIVER_STATUS
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217

#define STEALTHCHOP_XY
#define STEALTHCHOP_Z
#define STEALTHCHOP_E

/**
 * Optimize spreadCycle chopper parameters by using predefined parameter sets
 * or with the help of an example included in the library.
 * Provided parameter sets are
 * CHOPPER_DEFAULT_12V
 * CHOPPER_DEFAULT_19V
 * CHOPPER_DEFAULT_24V
 * CHOPPER_DEFAULT_36V
 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
 *
 * Define your own with
 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
 */
#define CHOPPER_TIMING CHOPPER_DEFAULT_24V

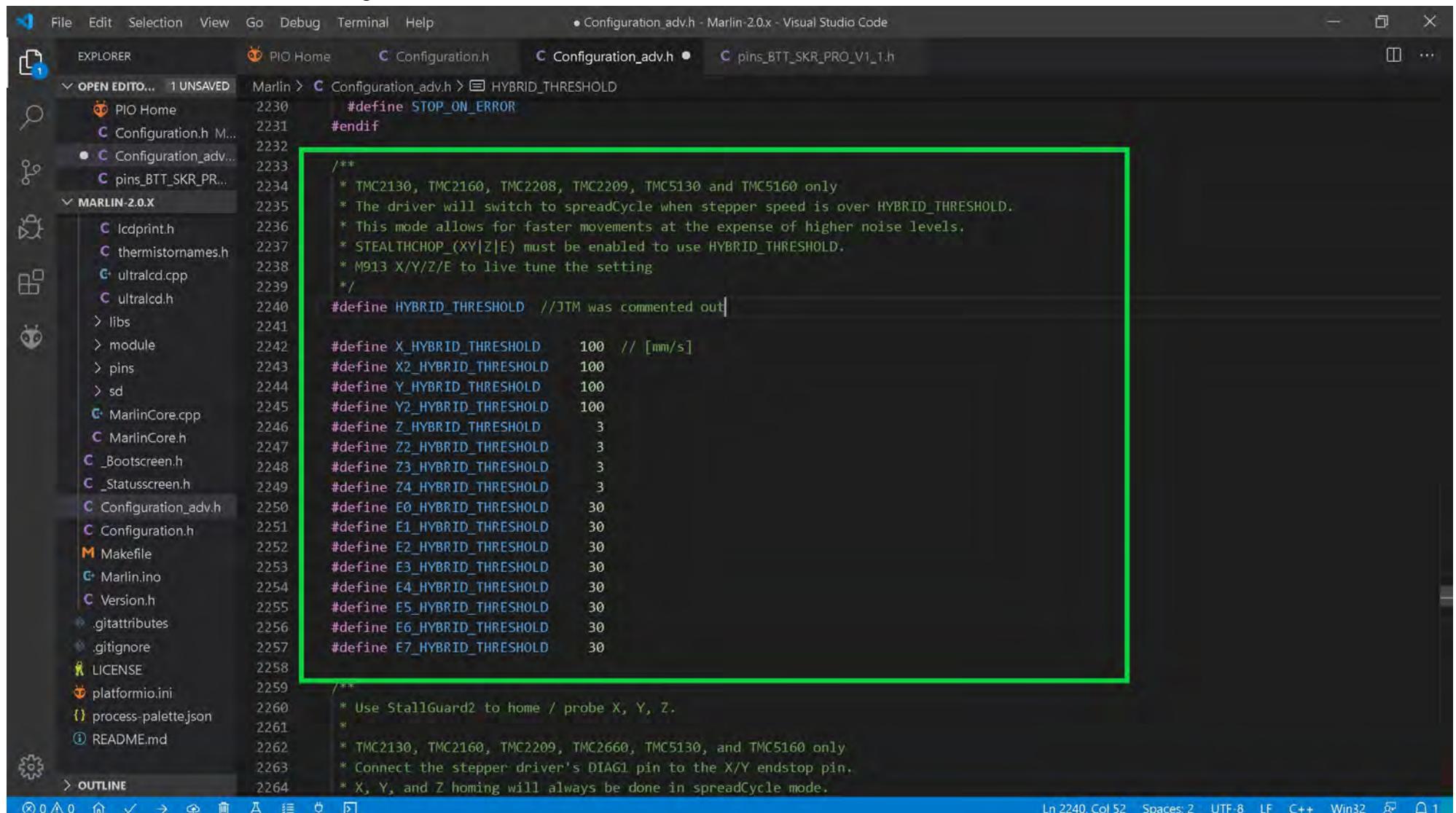
/**
 * Monitor Trinamic drivers for error conditions,
 * like overtemperature and short to ground.
 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- Now you either enable "HYBRID_THRESHOLD" or disable it. By default it is disabled. "HYBRID_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays the Marlin firmware configuration. A green rectangular box highlights the `HYBRID_THRESHOLD` section. The code in this section is as follows:

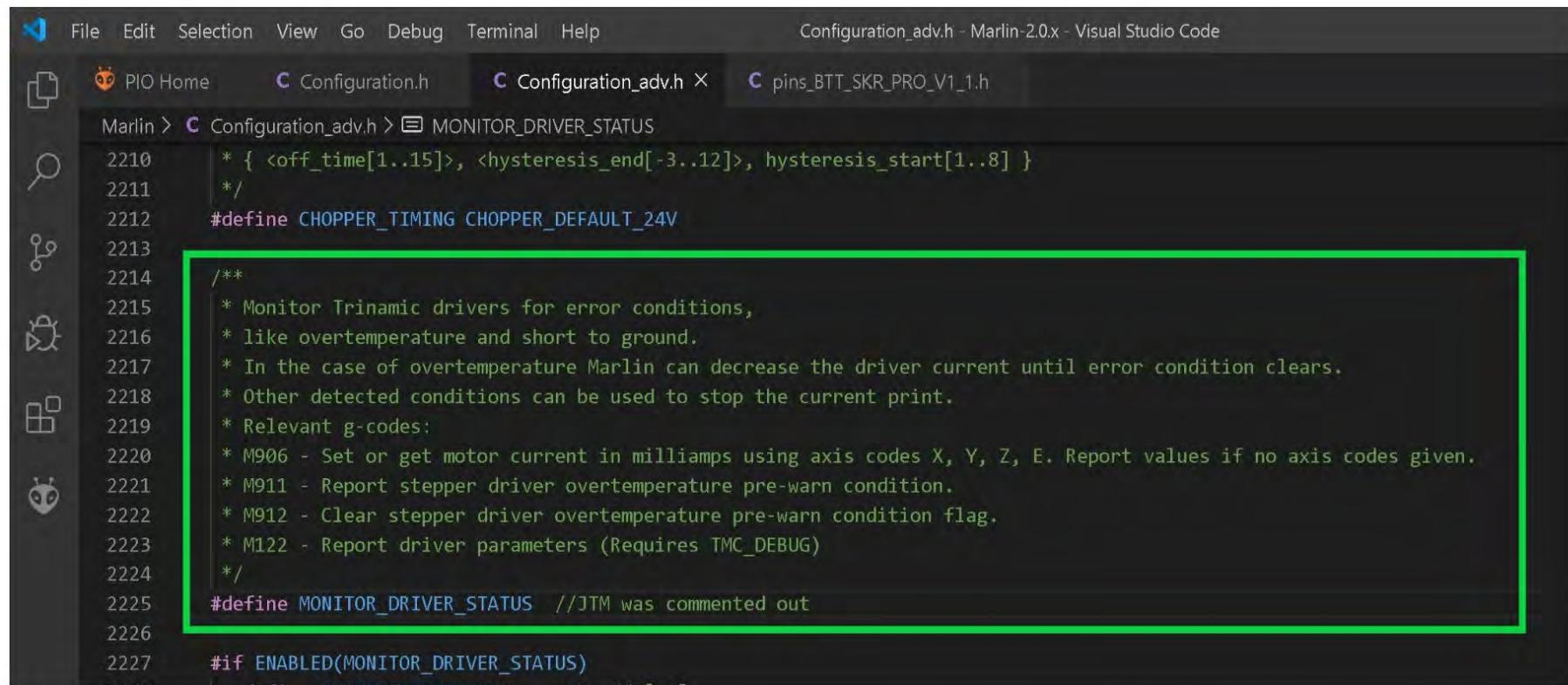
```

2230     #define STOP_ON_ERROR
2231     #endif
2232
2233 /**
2234 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
2235 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
2236 * This mode allows for faster movements at the expense of higher noise levels.
2237 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
2238 * M913 X/Y/Z/E to live tune the setting
2239 */
2240 #define HYBRID_THRESHOLD //JTM was commented out
2241
2242 #define X_HYBRID_THRESHOLD    100 // [mm/s]
2243 #define X2_HYBRID_THRESHOLD   100
2244 #define Y_HYBRID_THRESHOLD    100
2245 #define Y2_HYBRID_THRESHOLD   100
2246 #define Z_HYBRID_THRESHOLD    3
2247 #define Z2_HYBRID_THRESHOLD   3
2248 #define Z3_HYBRID_THRESHOLD   3
2249 #define Z4_HYBRID_THRESHOLD   3
2250 #define E0_HYBRID_THRESHOLD   30
2251 #define E1_HYBRID_THRESHOLD   30
2252 #define E2_HYBRID_THRESHOLD   30
2253 #define E3_HYBRID_THRESHOLD   30
2254 #define E4_HYBRID_THRESHOLD   30
2255 #define E5_HYBRID_THRESHOLD   30
2256 #define E6_HYBRID_THRESHOLD   30
2257 #define E7_HYBRID_THRESHOLD   30
2258
2259 /**
2260 * Use StallGuard2 to home / probe X, Y, Z.
2261 *
2262 * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
2263 * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
2264 * X, Y, and Z homing will always be done in spreadCycle mode.
2265 */

```

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line](#).
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

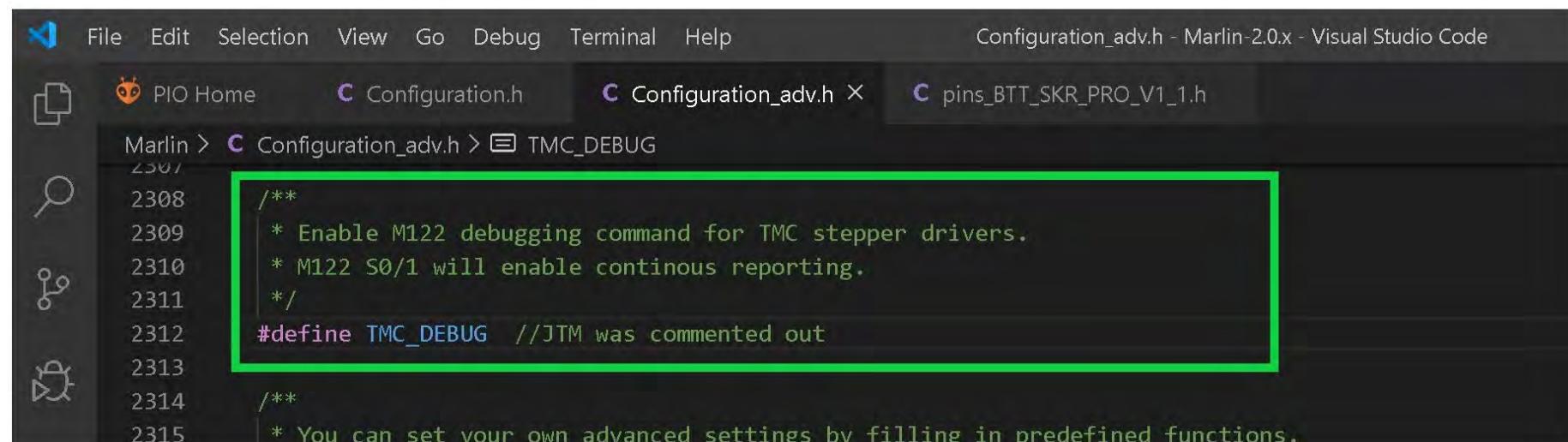
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

2210     * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211     */
2212     #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //JTM was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > TMC_DEBUG

```

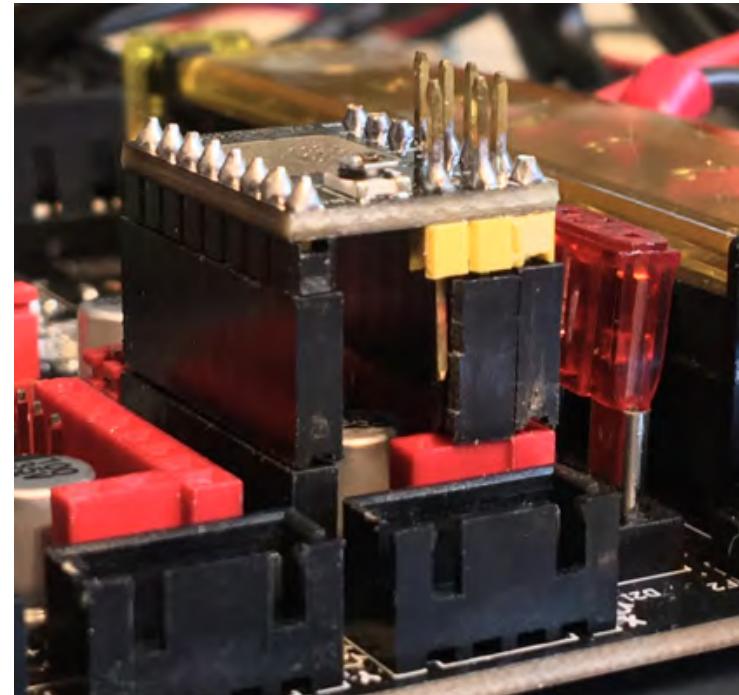
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //JTM was commented out
2313
2314 /**
2315 * You can set your own advanced settings by filling in predefined functions.

```

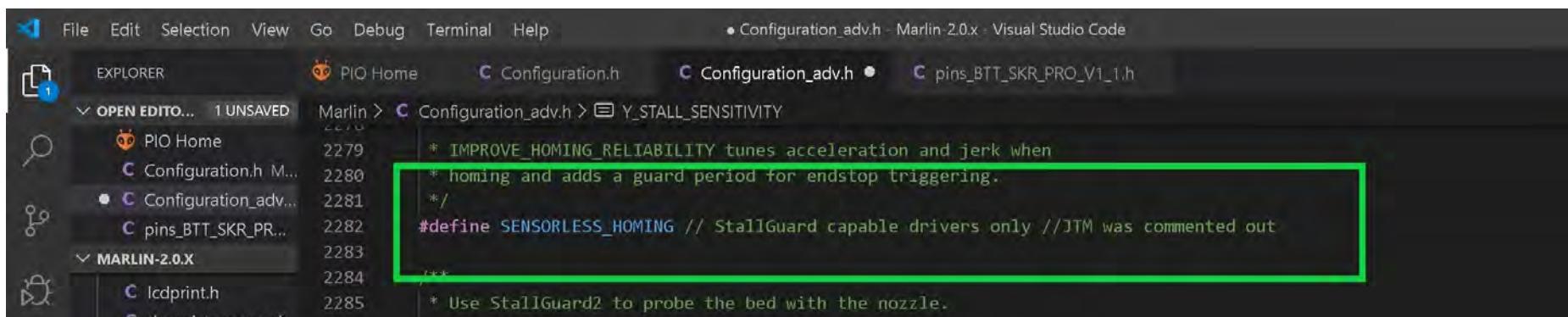
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- This next section covers sensor-less homing which is available for the TMC2130 in SPI mode. I want to enable it so I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG pin on the TMC2130 driver has to be plugged into the SKR PRO V1.1 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG pin on the Z axis TMC2130 is NOT connected to the board. I plan to plug my Z axis TMC2130 as seen in the picture below



- Sensorless homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"



```

File Edit Selection View Go Debug Terminal Help
• Configuration_adv.h - Marlin-2.0.x : Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h • pins_BTT_SKR_PRO_V1_1.h
✓ OPEN EDITOR... 1 UNSAVED Marlin > Configuration_adv.h > Y_STALL_SENSITIVITY
    PIO Home Configuration.h M...
    Configuration_adv.h • pins_BTT_SKR_PR...
    MARLIN-2.0.X Lcdprint.h thermistor.h
    Configuration_adv.h
    #define SENSORLESS_HOMING // StallGuard capable drivers only //JTM was commented out
    /**
    * Use StallGuard2 to probe the bed with the nozzle.
    */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- Next we set a "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X_STALL_SENSITIVITY" and the "Y_STALL_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z_STALL_SENSITIVITY".

```
#if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
#define X_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
#define Y_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//#define Z_STALL_SENSITIVITY 8
//#define SPI_ENDSTOPS // TMC2130 only
//#define IMPROVE_HOMING_RELIABILITY
#endif
```

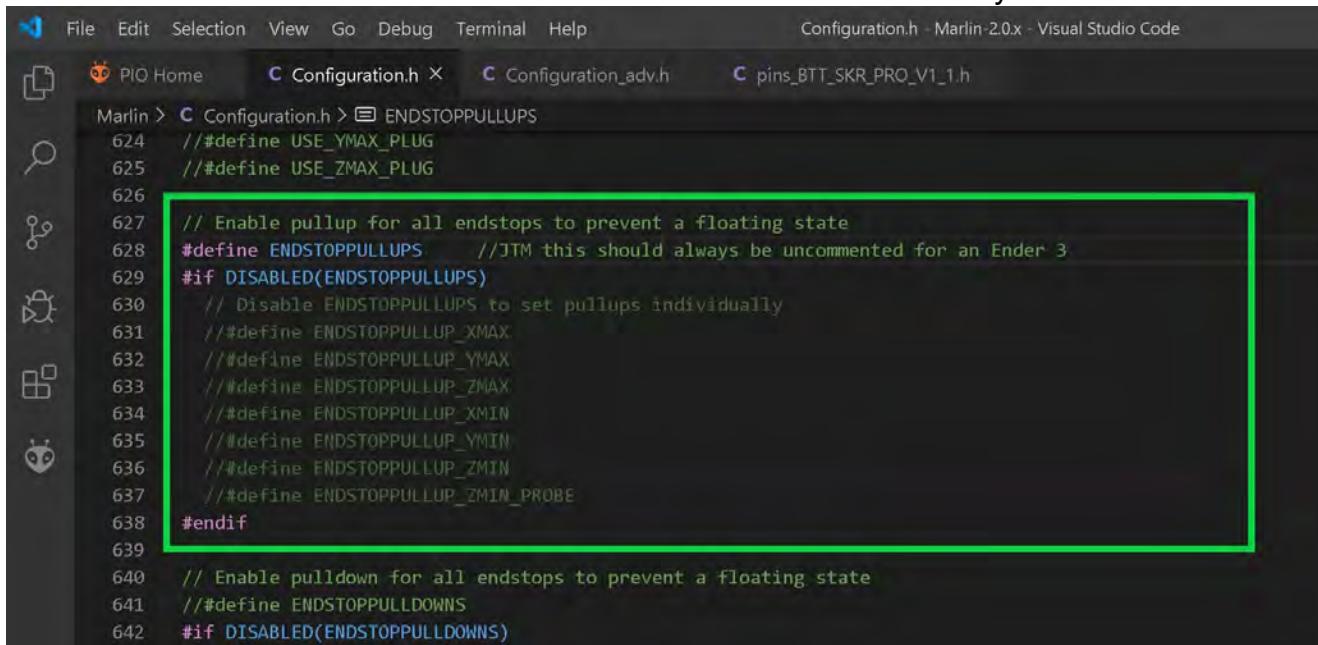
- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.

```
// @section homing
#define X_HOME_BUMP_MM 0 //JTM was 5
#define Y_HOME_BUMP_MM 0 //JTM was 5
#define Z_HOME_BUMP_MM 2
#define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
#define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
#define HOME_Y_BEFORE_X
#define CODEPENDENT_XY_HOMING
#endif
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default.



```

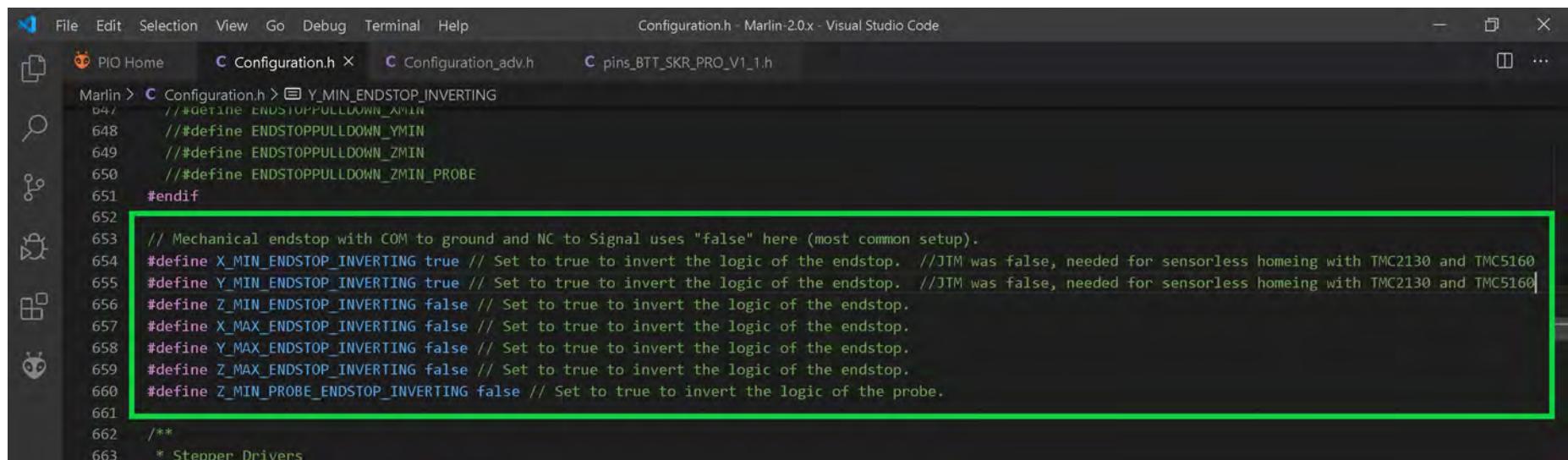
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin 2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration.h > ENDSTOPPULLUPS
624 // #define USE_YMAX_PLUG
625 // #define USE_ZMAX_PLUG
626
627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS // JTM this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 // #define ENDSTOPPULLUP_XMAX
632 // #define ENDSTOPPULLUP_YMAX
633 // #define ENDSTOPPULLUP_ZMAX
634 // #define ENDSTOPPULLUP_XMIN
635 // #define ENDSTOPPULLUP_YMIN
636 // #define ENDSTOPPULLUP_ZMIN
637 // #define ENDSTOPPULLUP_ZMIN_PROBE
638#endif
639
640 // Enable pulldown for all endstops to prevent a floating state
641 // #define ENDSTOPPULLDOWNS
642 #if DISABLED(ENDSTOPPULLDOWNS)

```

- Next to allow sensor-less homing to work we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z_MIN_ENDSTOP_INVERTING" set to false.



```

File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin 2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

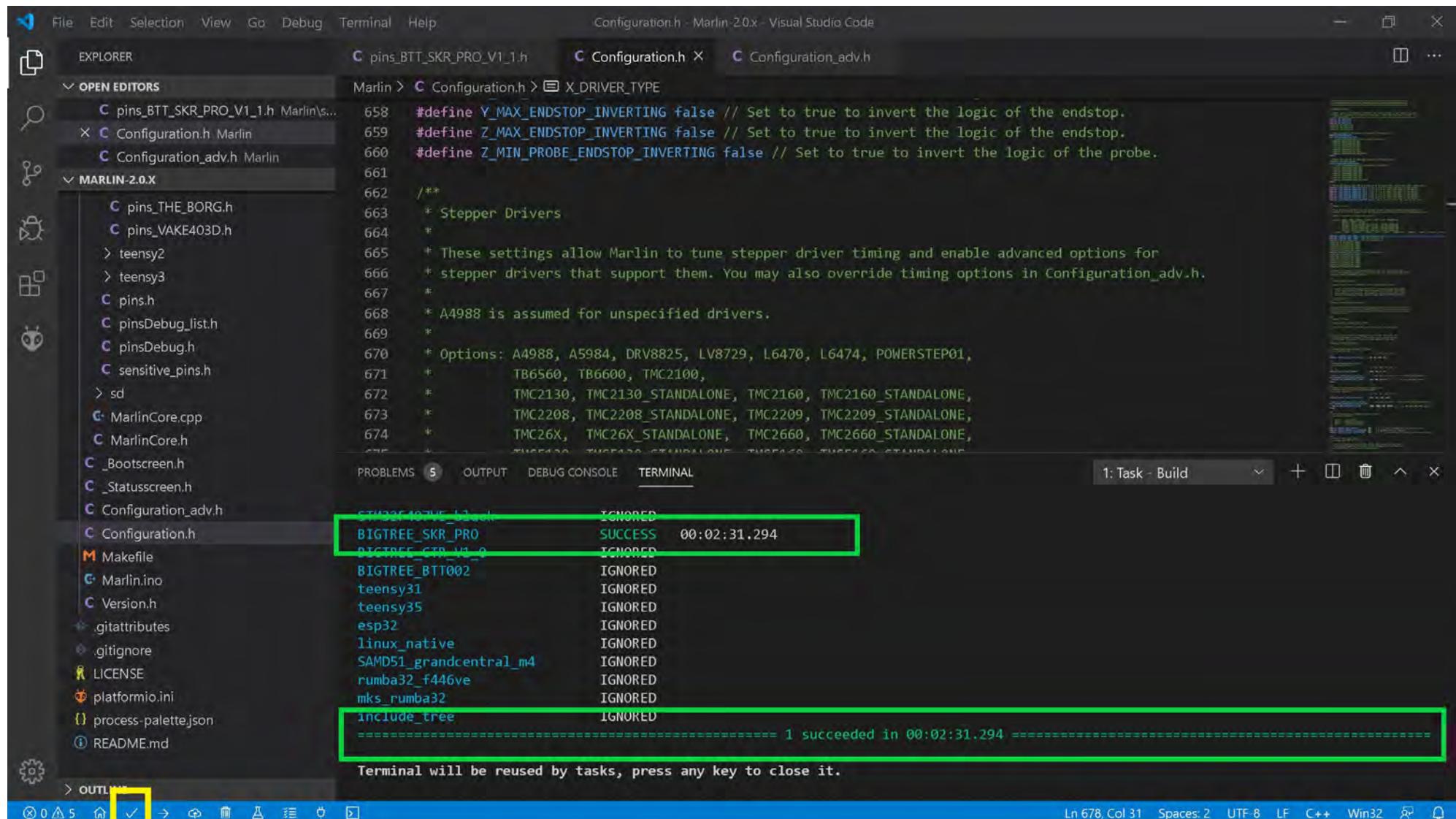
Marlin > Configuration.h > Y_MIN_ENDSTOP_INVERTING
647 // #define ENDSTOPPULLDOWN_XMIN
648 // #define ENDSTOPPULLDOWN_YMIN
649 // #define ENDSTOPPULLDOWN_ZMIN
650 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
651#endif
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654#define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop. // JTM was false, needed for sensorless homeing with TMC2130 and TMC5160
655#define Y_MIN_ENSTOP_INVERTING true // Set to true to invert the logic of the endstop. // JTM was false, needed for sensorless homeing with TMC2130 and TMC5160
656#define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658#define Y_MAX_ENSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**
663 * Stepper Drivers

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- The end of Marlin setup for BIQU TMC2130 drivers in SPI mode . Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages ,as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and various Marlin core files.
- Editor:** Displays the Configuration.h file with code related to stepper drivers and endstops.
- Terminal:** Shows the build process for different boards:

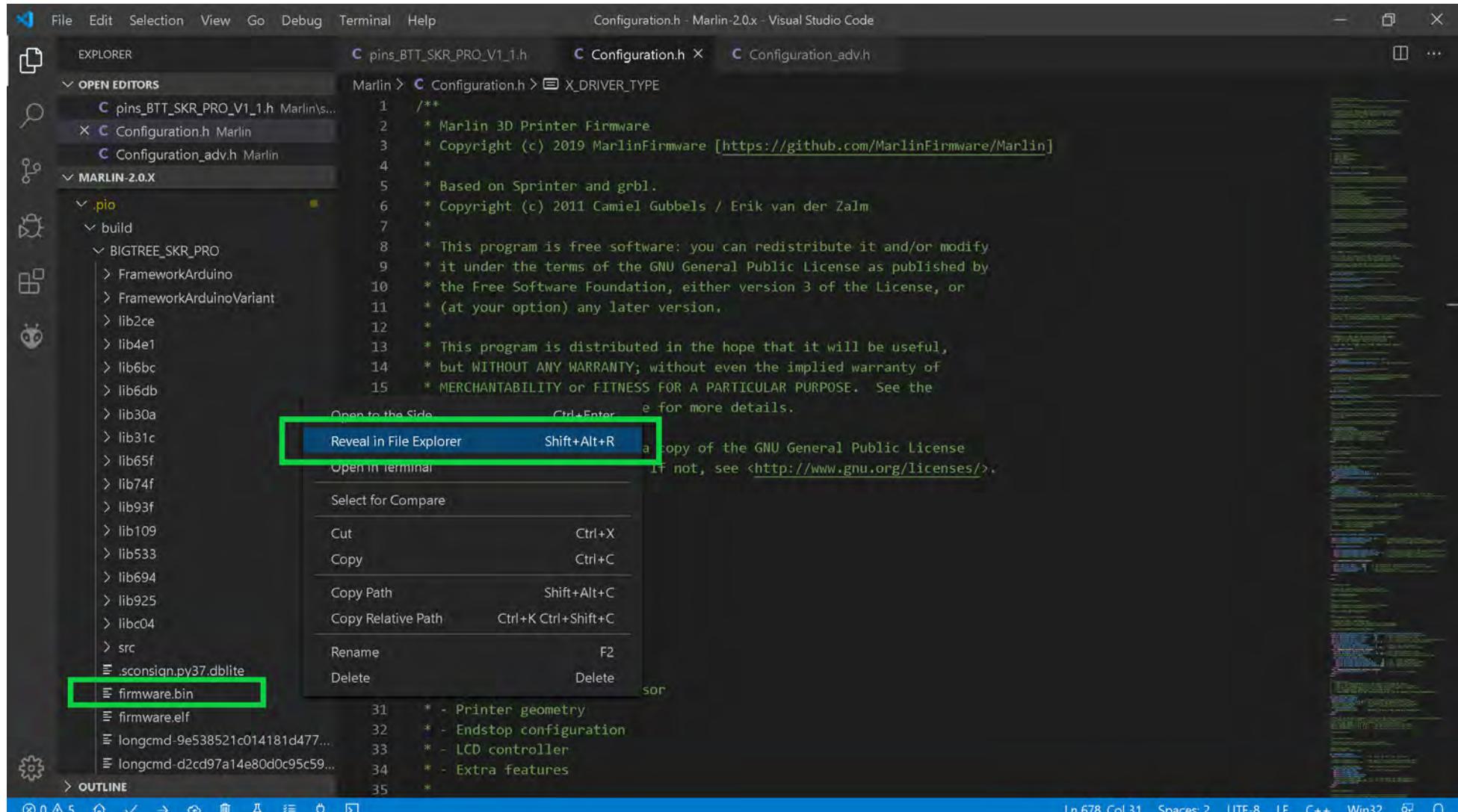
Board	Status	Time
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
BIGTREE_BTT002	IGNORED	
teensy31	IGNORED	
teensy35	IGNORED	
esp32	IGNORED	
linux_native	IGNORED	
SAMD51_grandcentral_m4	IGNORED	
rumba32_f446ve	IGNORED	
mks_rumba32	IGNORED	
include_tree	IGNORED	

 The terminal also displays the message: "===== 1 succeeded in 00:02:31.294 =====".
- Bottom Status Bar:** Shows the terminal will be reused, line count (Ln 678), column count (Col 31), and other build-related information.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

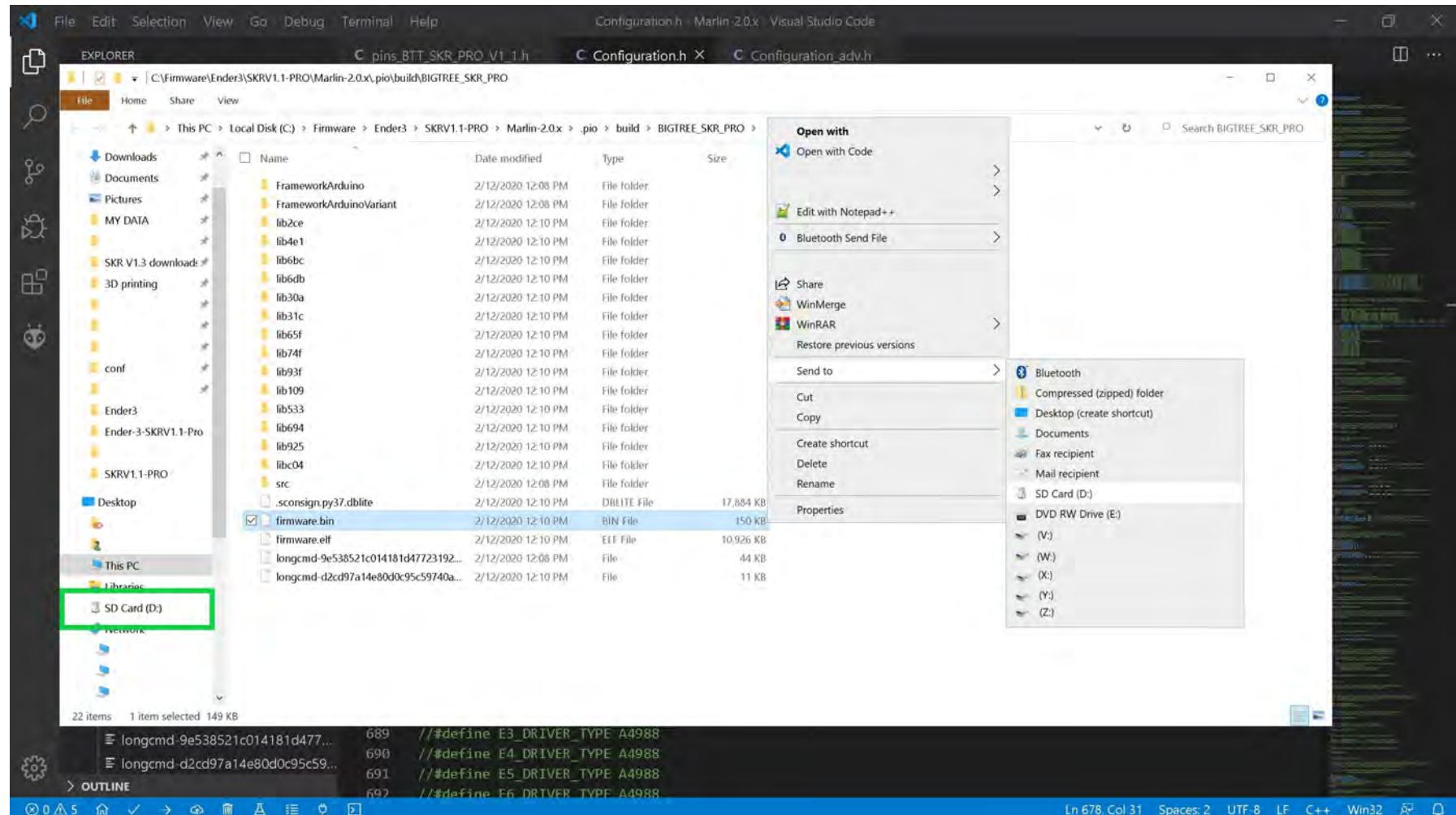
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



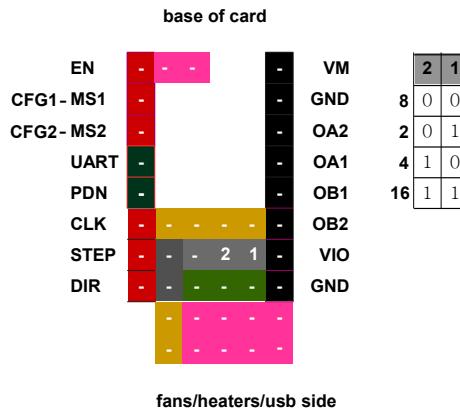
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



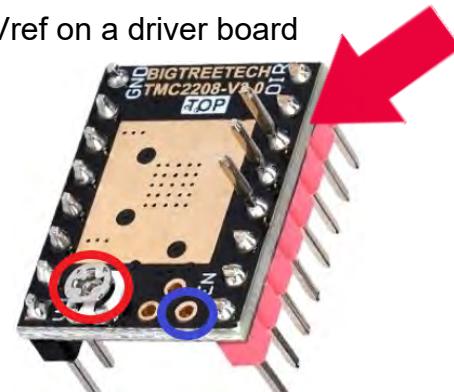
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2208 V3.0**Stand Alone Mode****Stand Alone Mode**

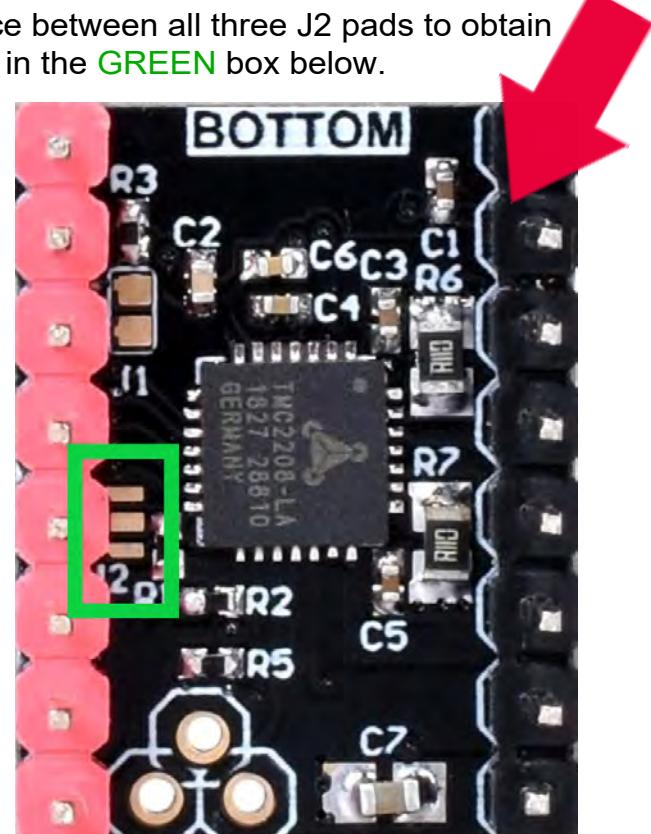
Driver Chip	CFG2/MS2	CFG1/MS1	Steps	Interpolation	Mode
TMC2208	GND	GND	1/8	1/256	stealthChop
Standalone Mode	GND	VIO	1/2	1/256	stealthChop
5.5v - 35v DC	VIO	GND	1/4	1/256	stealthChop
1.7A	VIO	VIO	1/16	1/256	stealthChop
Maximum subdivision 16					
Driving current calculation formula	$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$				
	Note: Use 90% of the calculated Vref when tuning the stepper driver board				
	$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$				

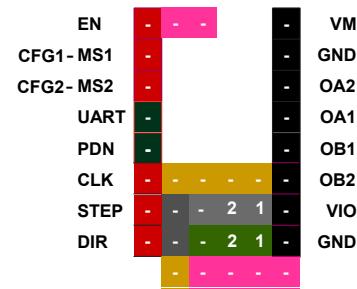
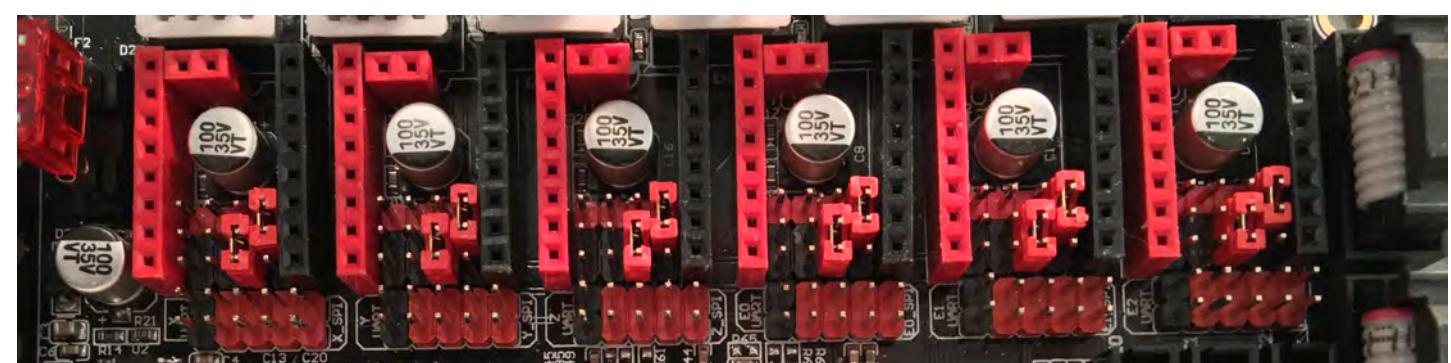
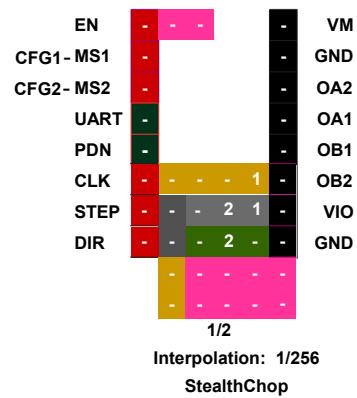
Note: MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in RED; OR use the "Vref Test point" location on the top of the driver board, as shown in BLUE . [See the appendix](#) for instructions on how to set the Vref on a driver board



Note: The three pads (J2), on the bottom of the BIQU TMC2208 V3.0 driver boards, **MUST NOT be** connected. Again, a gap MUST be in place between all three J2 pads to obtain Standalone Mode for the TMC2208 V3.0, as seen in the GREEN box below.



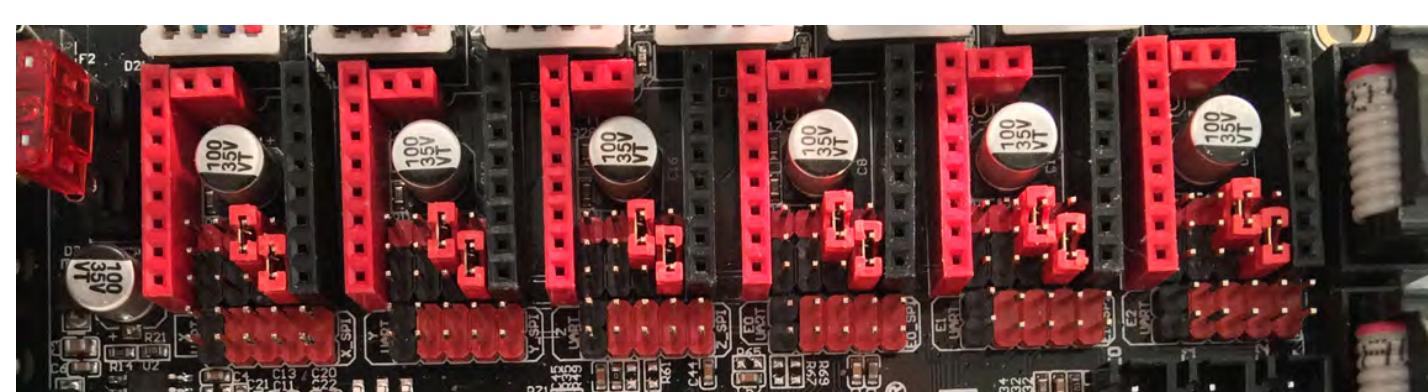
Stand Alone Mode**Stand Alone Mode****Stand Alone Mode**

BIQU TMC2208 V3.0

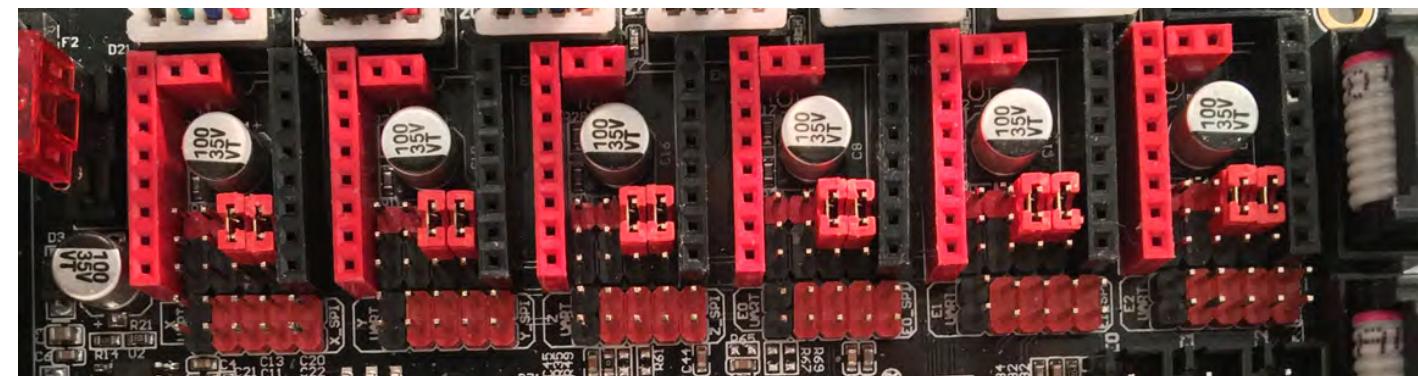
Stand Alone Mode

Stand Alone Mode**Stand Alone Mode**

EN	-	-	VM
CFG1- MS1	-	-	GND
CFG2- MS2	-	-	OA2
UART	-	-	OA1
PDN	-	-	OB1
CLK	-	2	OB2
STEP	-	2	VIO
DIR	-	1	GND
1/4			
Interpolation: 1/256			
StealthChop			

**Stand Alone Mode**

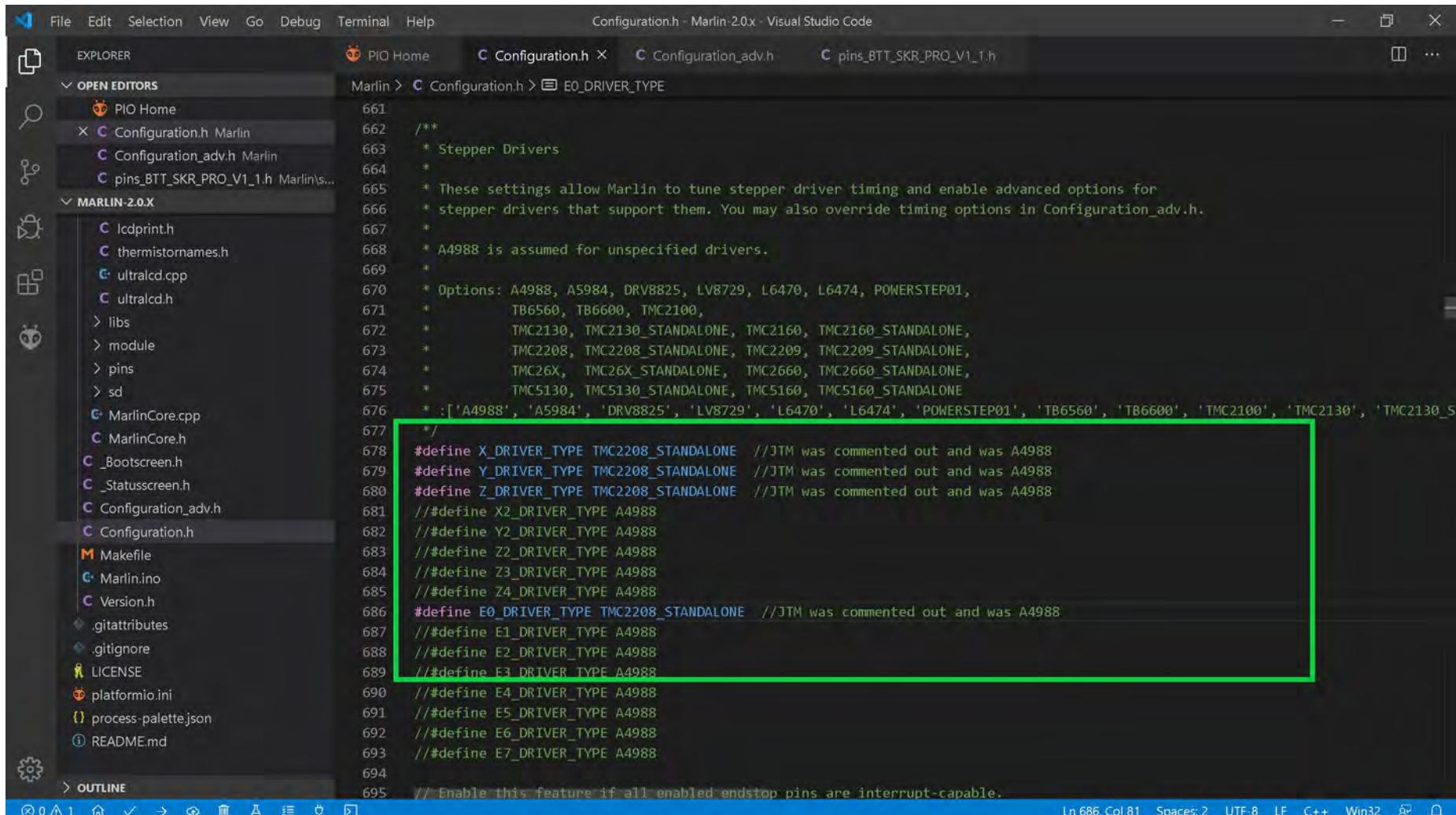
EN	-	-	VM
CFG1- MS1	-	-	GND
CFG2- MS2	-	-	OA2
UART	-	-	OA1
PDN	-	-	OB1
CLK	-	2	OB2
STEP	-	2	VIO
DIR	-	1	GND
1/16			
Interpolation: 1/256			
StealthChop			



The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2208 stepper motor drivers in stand alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand alone mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



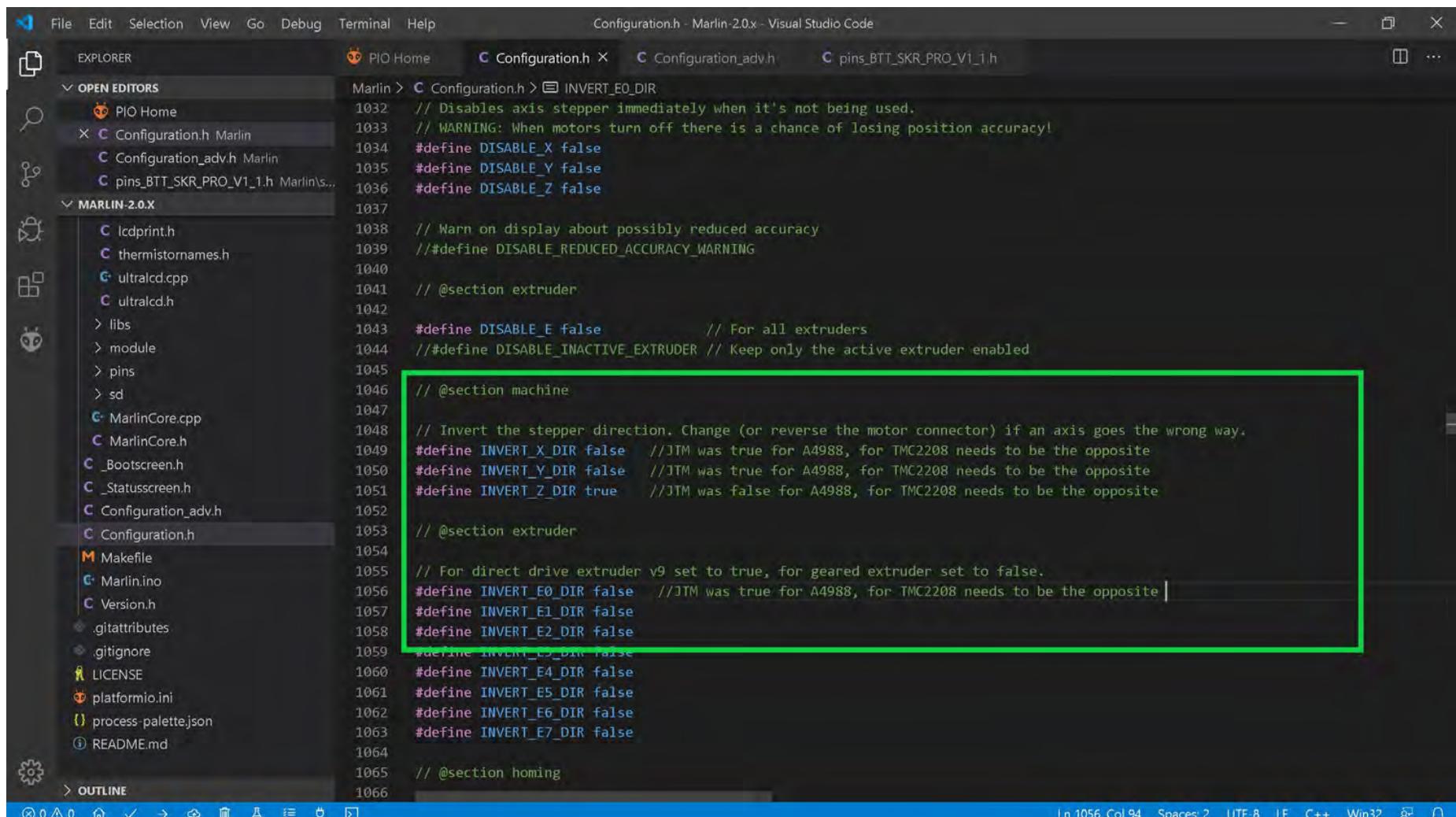
```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
676 */
677 #define X_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
678 #define Y_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
679 #define Z_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 #define E0_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693
694 // Enable this feature if all enabled endstop pins are interrupt-capable.
695 
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037 // Warn on display about possibly reduced accuracy
1038 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2208 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false    // JTM was true for A4988, for TMC2208 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

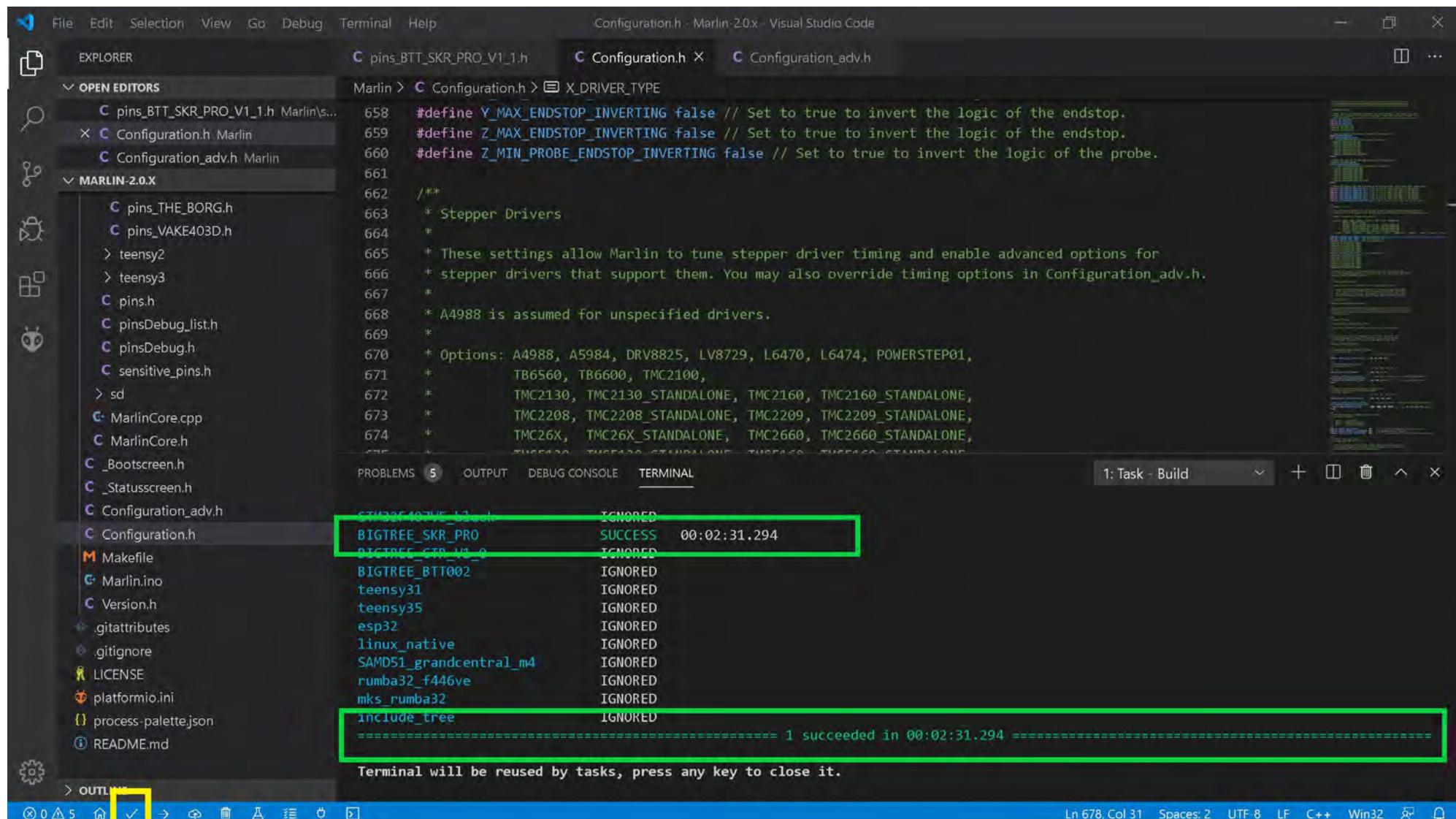
```

Ln 1056, Col 94 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in stand alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and MarlinCore.cpp.
- Terminal:** Displays the build log for the "BIGTREE_SKR_PRO" target, which completed successfully in 00:02:31.294.
- Build Status:** A yellow box highlights the checkmark icon in the bottom-left corner of the terminal tab, indicating a successful build.
- Build Log:**

```

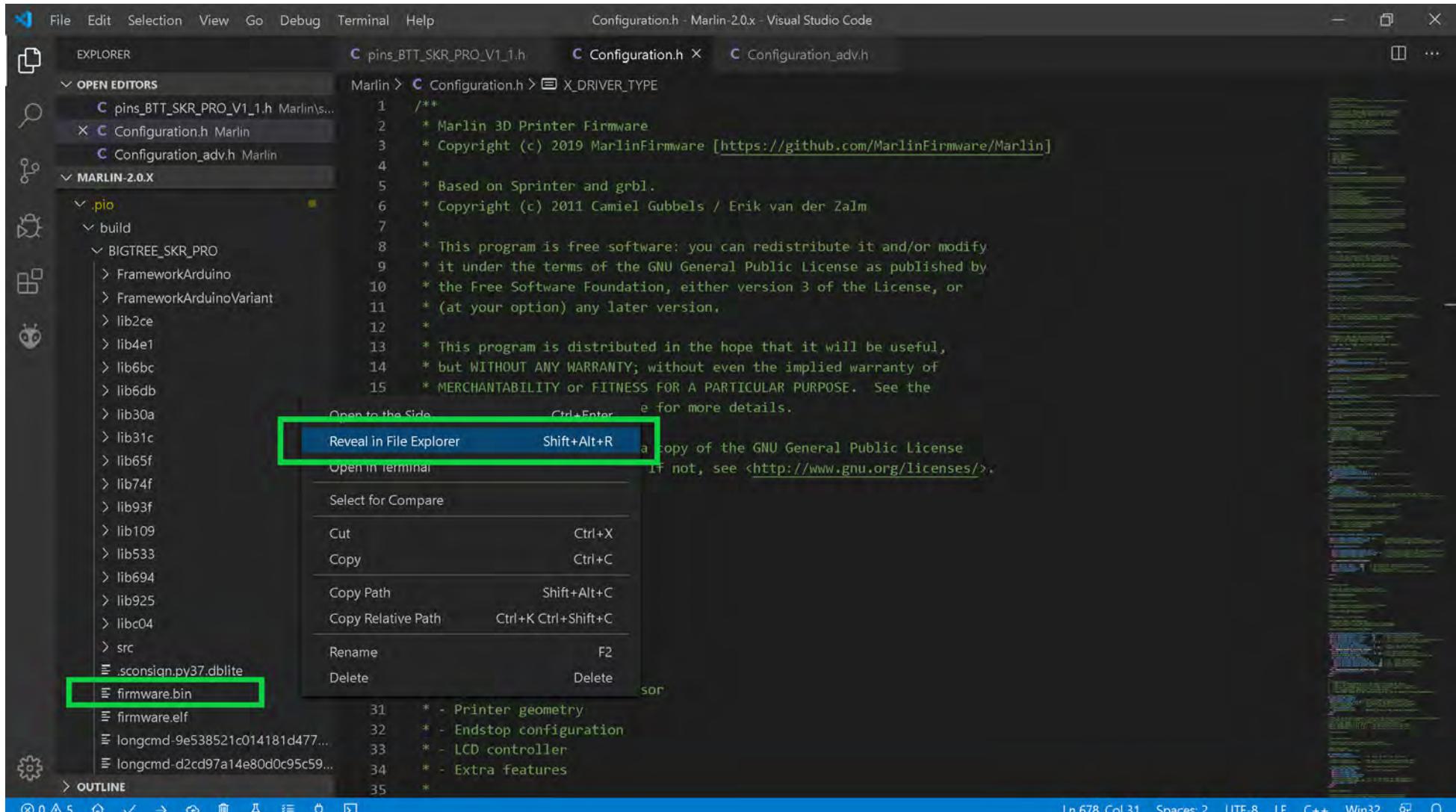
BIGTREE_SKR_PRO          IGNORED
BIGTREE_SKR_V1_2          IGNORED
BIGTREE_BT002             IGNORED
teensy31                  IGNORED
teensy35                  IGNORED
esp32                      IGNORED
linux_native               IGNORED
SAMD51_grandcentral_m4   IGNORED
rumba32_f446ve             IGNORED
mks_rumba32                IGNORED
include_tree                IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode

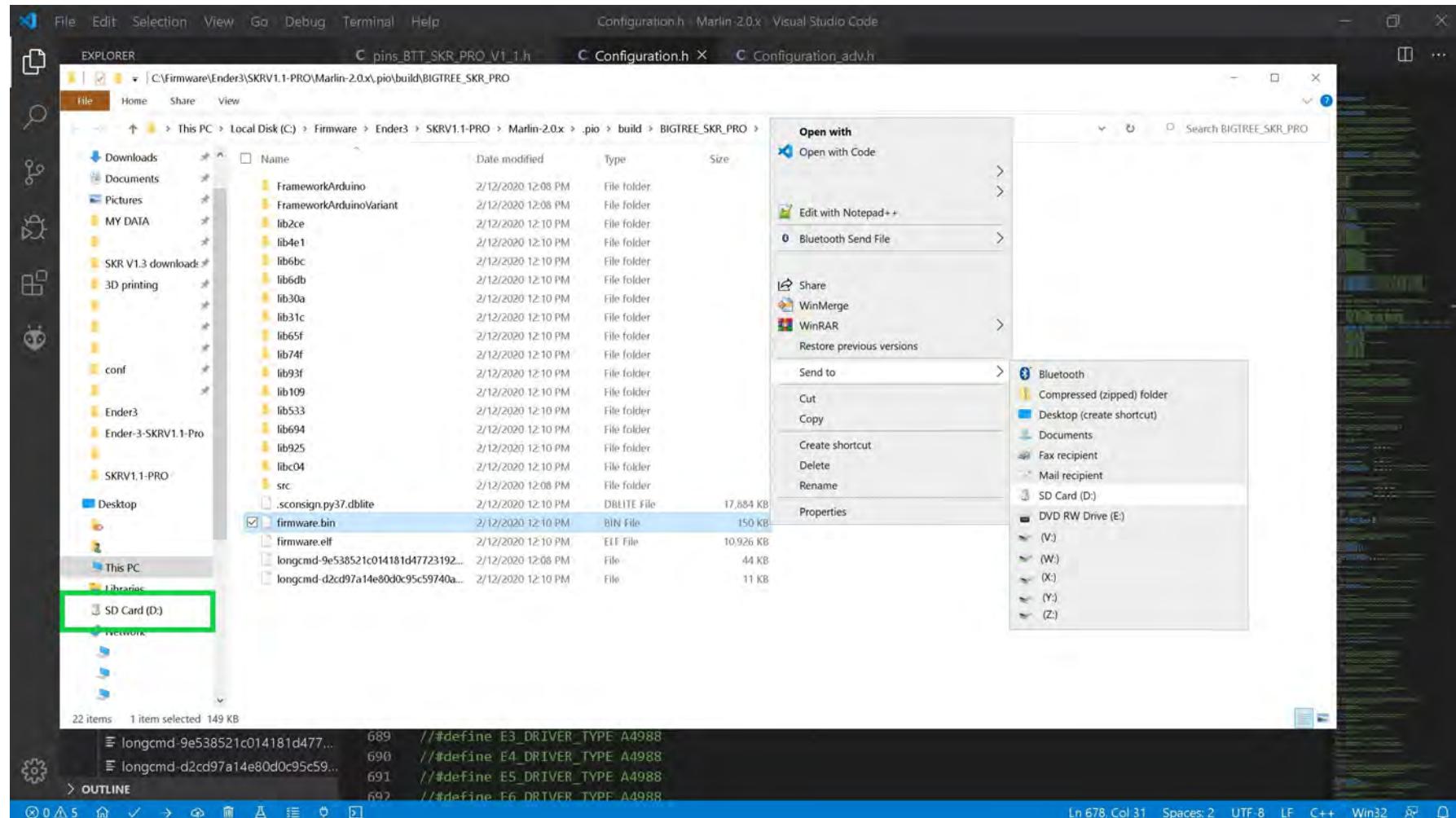
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



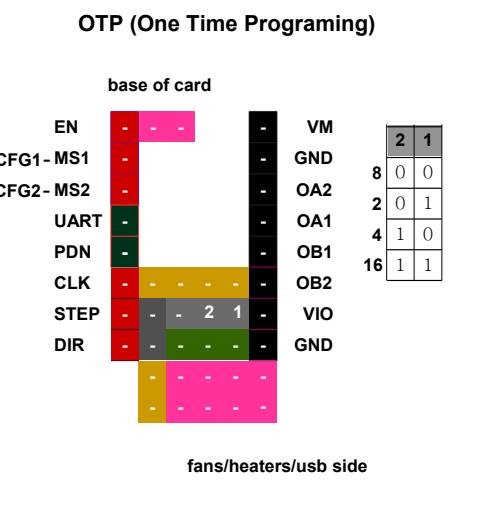
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand Alone Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2208 V3.0One Time Programming (OTP) Mode

Driver Chip	CFG2/MS2	CFG1/MS1	Steps	Interpolation	Mode
TMC2208 OTP Mode	GND	GND	1/8	1/256	spreadCycle
5.5v - 35V DC 1.7A	GND	VIO	1/2	1/256	spreadCycle
Maximum subdivision 16	VIO	GND	1/4	1/256	spreadCycle
	VIO	VIO	1/16	1/256	spreadCycle
Driving current calculation formula	$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$				
	Note: Use 90% of the calculated Vref when tuning the stepper driver board				
	$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$				

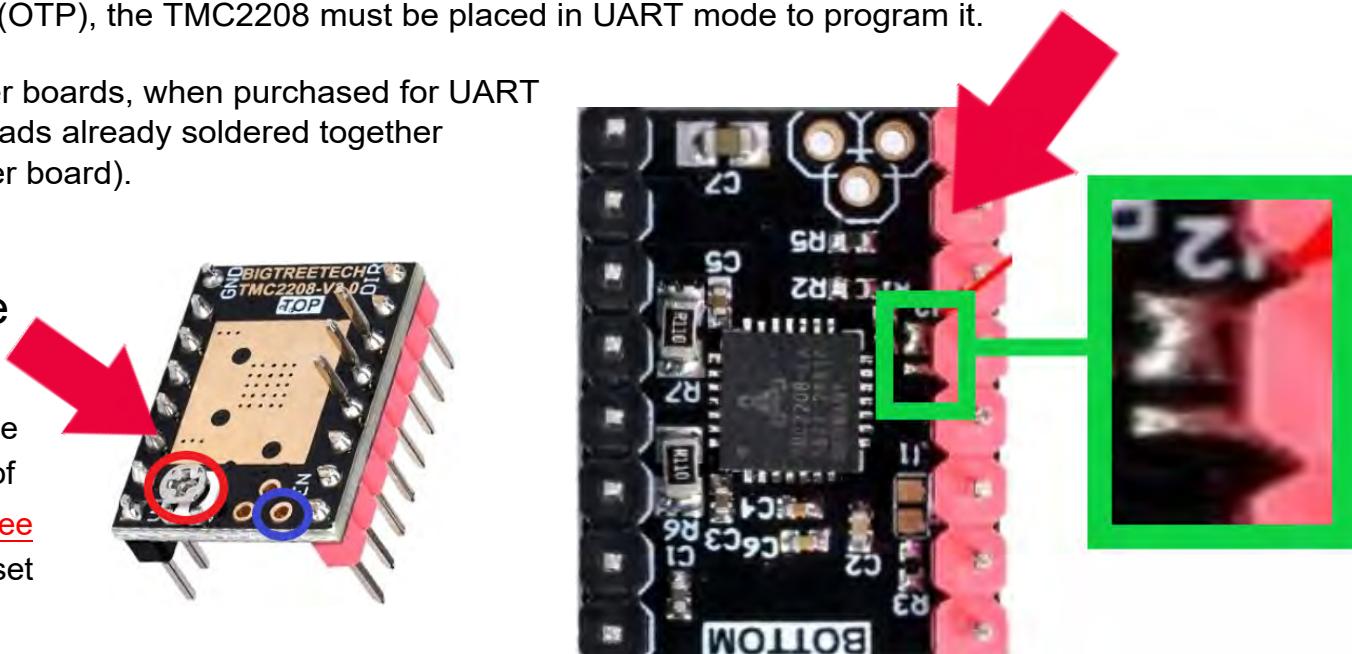
NOTE: Stand Alone Mode by default uses StealthChop, if you want SpreadCycle, you MUST use OTP mode. See TMC220x Configurator for One-Time-Programming Information: [TMC220x Configurator](#).

Important: To place BIQU TMC2208 V3.0 into OTP Mode, at J2,

you must solder the top two pads together OR solder the bottom two pads together on the bottom of the driver board. As an example, the picture below shows the TOP two pads soldered together, as shown in the GREEN box, so that OTP mode can be obtained. To do One-Time-Programming (OTP), the TMC2208 must be placed in UART mode to program it.

MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in RED; OR use the "Vref Test point" location on the top of the driver board, as shown in BLUE. [See the appendix](#) for instructions on how to set the Vref on a driver board



One Time Programming (OTP) Mode

OTP (One Time Programing)

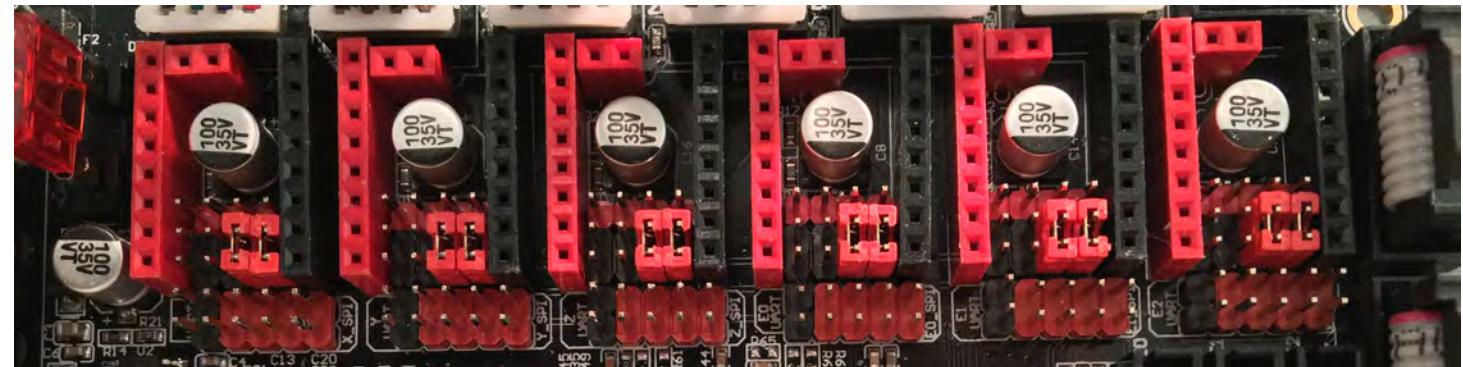
OTP (One Time Programing)

EN	-	-	-	VM	-	-
CFG1-MS1	-	-	-	GND	-	-
CFG2-MS2	-	-	-	OA2	-	-
UART	-	-	-	OA1	-	-
PDN	-	-	-	OB1	-	-
CLK	-	-	-	OB2	-	-
STEP	-	-	2 1	VIO	-	-
DIR	-	-	2 1	GND	-	-

1/8

Interpolation: 1/256

SpreadCycle



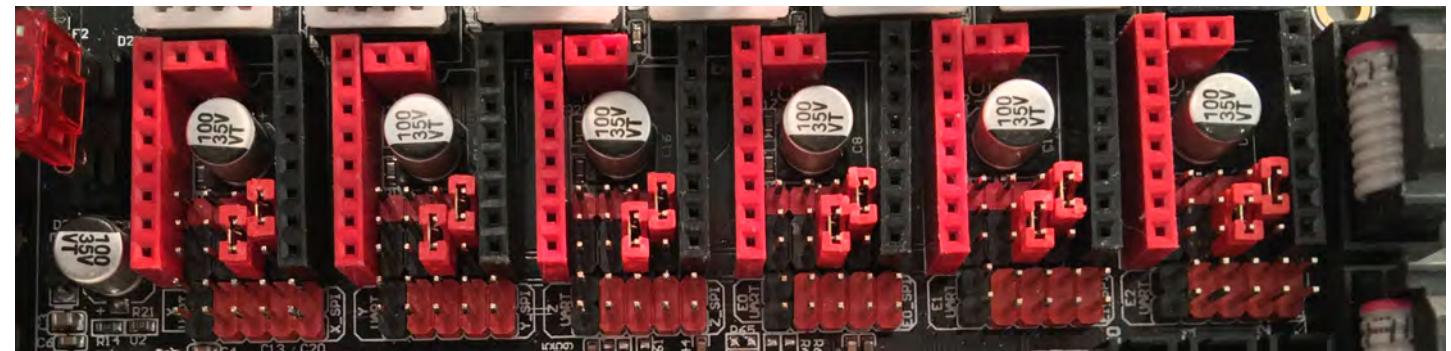
OTP (One Time Programming)

EN	-	-	-	VM	-	-
CFG1-MS1	-	-	-	GND	-	-
CFG2-MS2	-	-	-	OA2	-	-
UART	-	-	-	OA1	-	-
PDN	-	-	-	OB1	-	-
CLK	-	-	1	OB2	-	-
STEP	-	-	2 1	VIO	-	-
DIR	-	-	2	GND	-	-

1/2

Interpolation: 1/256

SpreadCycle



One Time Programming (OTP) Mode

OTP (One Time Programming)

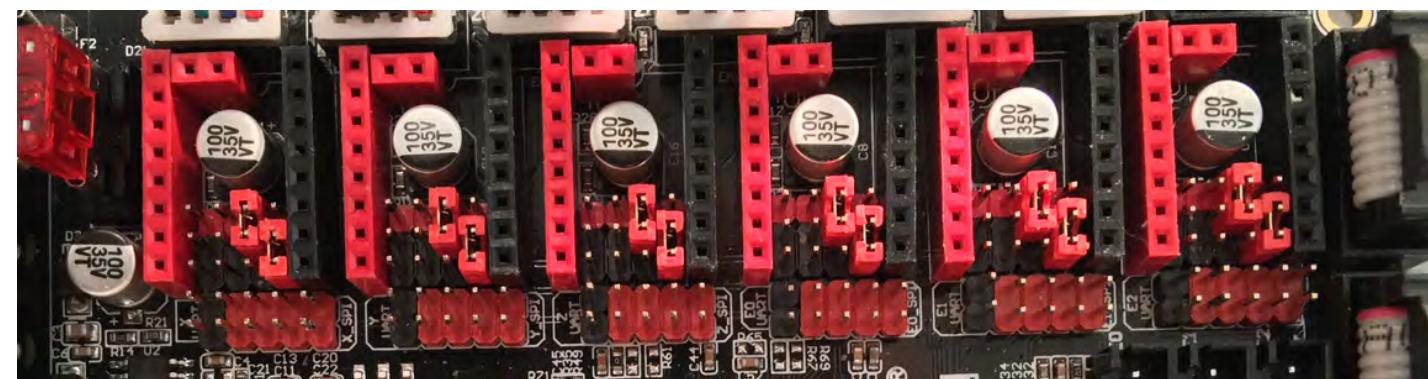
OTP (One Time Programming)

EN	-	-	-	VM	-	-
CFG1-MS1	-	-	-	GND	-	-
CFG2-MS2	-	-	-	OA2	-	-
UART	-	-	-	OA1	-	-
PDN	-	-	-	OB1	-	-
CLK	-	2	-	OB2	-	-
STEP	-	2	1	VIO	-	-
DIR	-	-	1	GND	-	-

1/4

Interpolation: 1/256

SpreadCycle



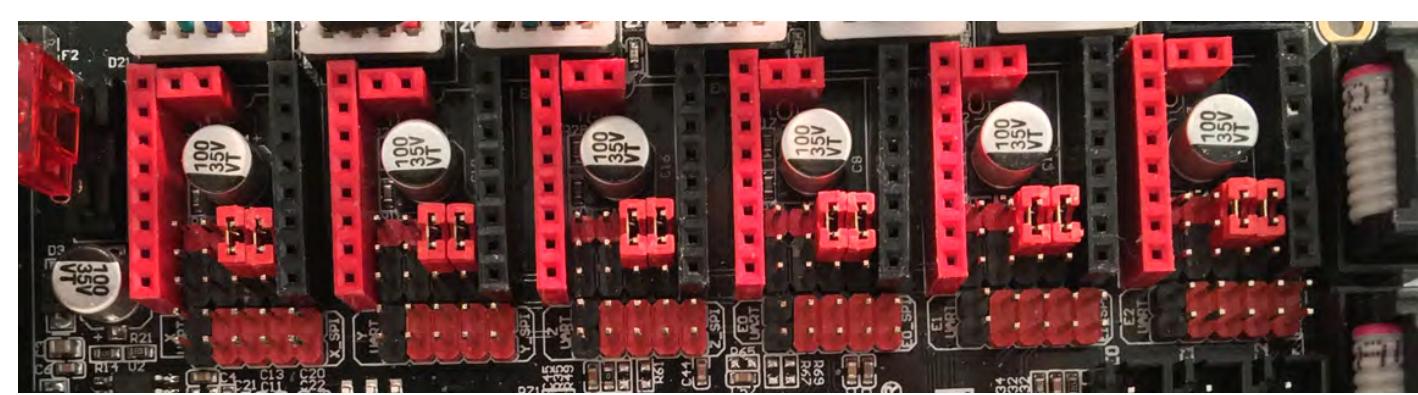
OTP (One Time Programming)

EN	-	-	-	VM	-	-
CFG1-MS1	-	-	-	GND	-	-
CFG2-MS2	-	-	-	OA2	-	-
UART	-	-	-	OA1	-	-
PDN	-	-	-	OB1	-	-
CLK	-	2	1	OB2	-	-
STEP	-	2	1	VIO	-	-
DIR	-	-	-	GND	-	-

1/16

Interpolation: 1/256

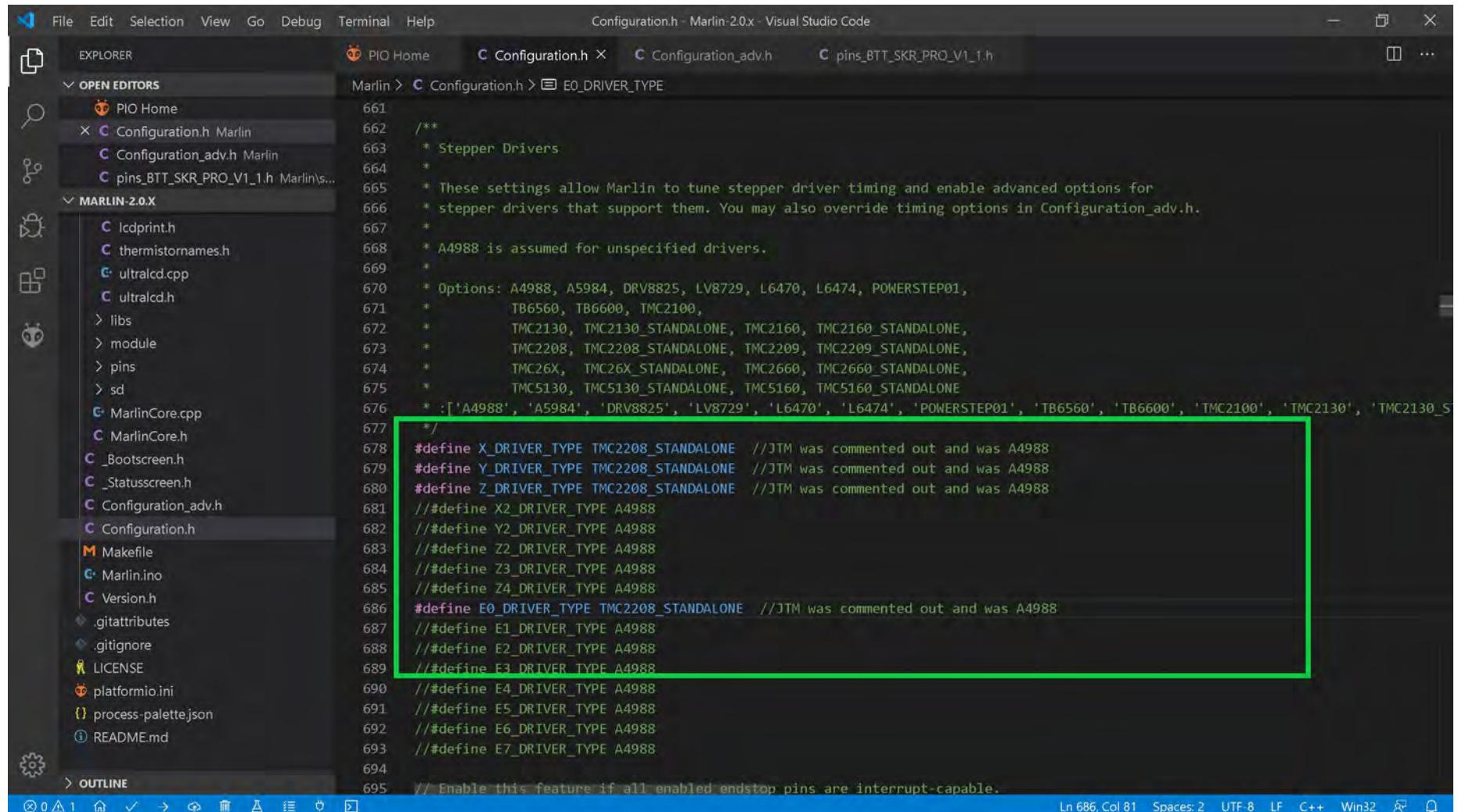
SpreadCycle



The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2208 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



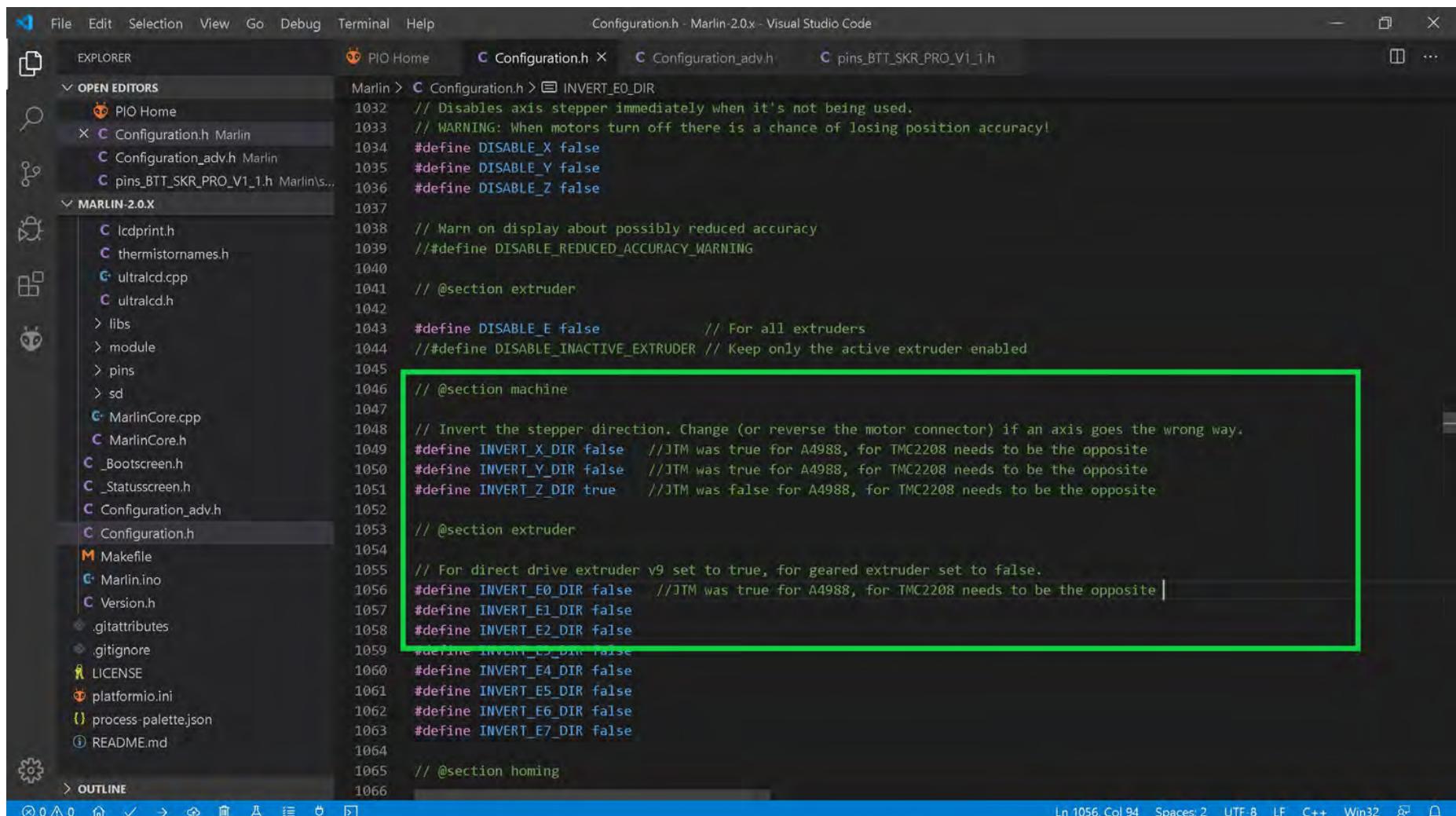
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration.h - Marlin 2.0.x - Visual Studio Code
- Explorer:**
 - OPEN EDITORS: PIO Home, Configuration.h (Marlin), Configuration_adv.h, pins_BTT_SKR_PRO_V1_1.h
 - MARLIN-2.0.X: Icdprint.h, thermistornames.h, ultralcd.cpp, ultralcd.h, libs, module, pins, sd, MarlinCore.cpp, MarlinCore.h, Bootscreen.h, Statusscreen.h, Configuration_adv.h, Configuration.h (highlighted), Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md
- Code Editor:** The Configuration.h file is open, showing code related to stepper drivers. A green rectangular box highlights the section from line 678 to 694, which defines driver types for X, Y, Z, E0, E1, E2, E3, E4, E5, E6, and E7 axes. The lines are annotated with comments indicating they were previously commented out (e.g., //JTM was commented out and was A4988).
- Status Bar:** Ln 686, Col 81, Spaces: 2, UTF-8, LF, C++, Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2208 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false    // JTM was true for A4988, for TMC2208 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

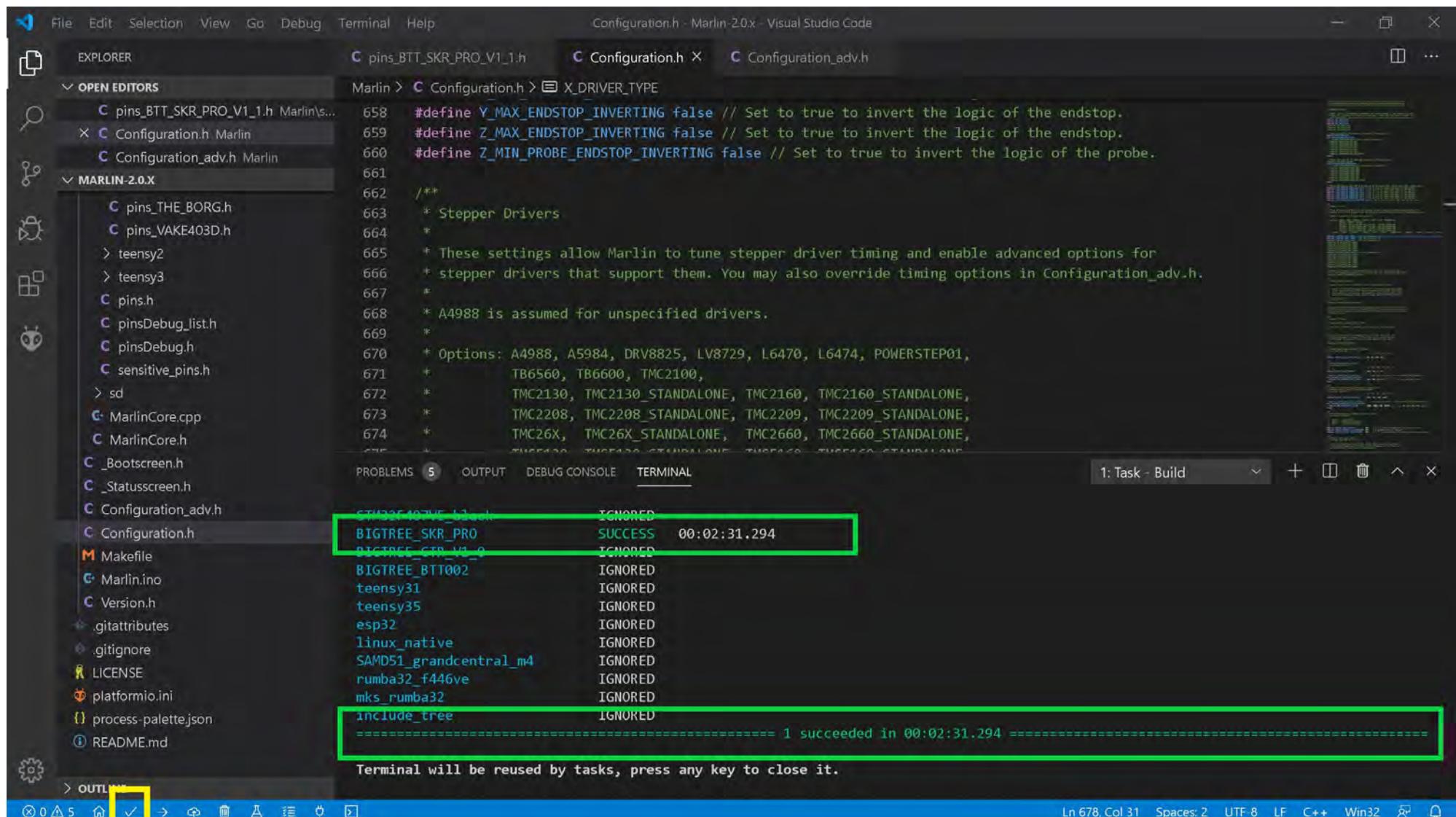
Ln 1056, Col 94 Spaces: 2 UTF-8 LF C++ Win32

OUTLINE

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

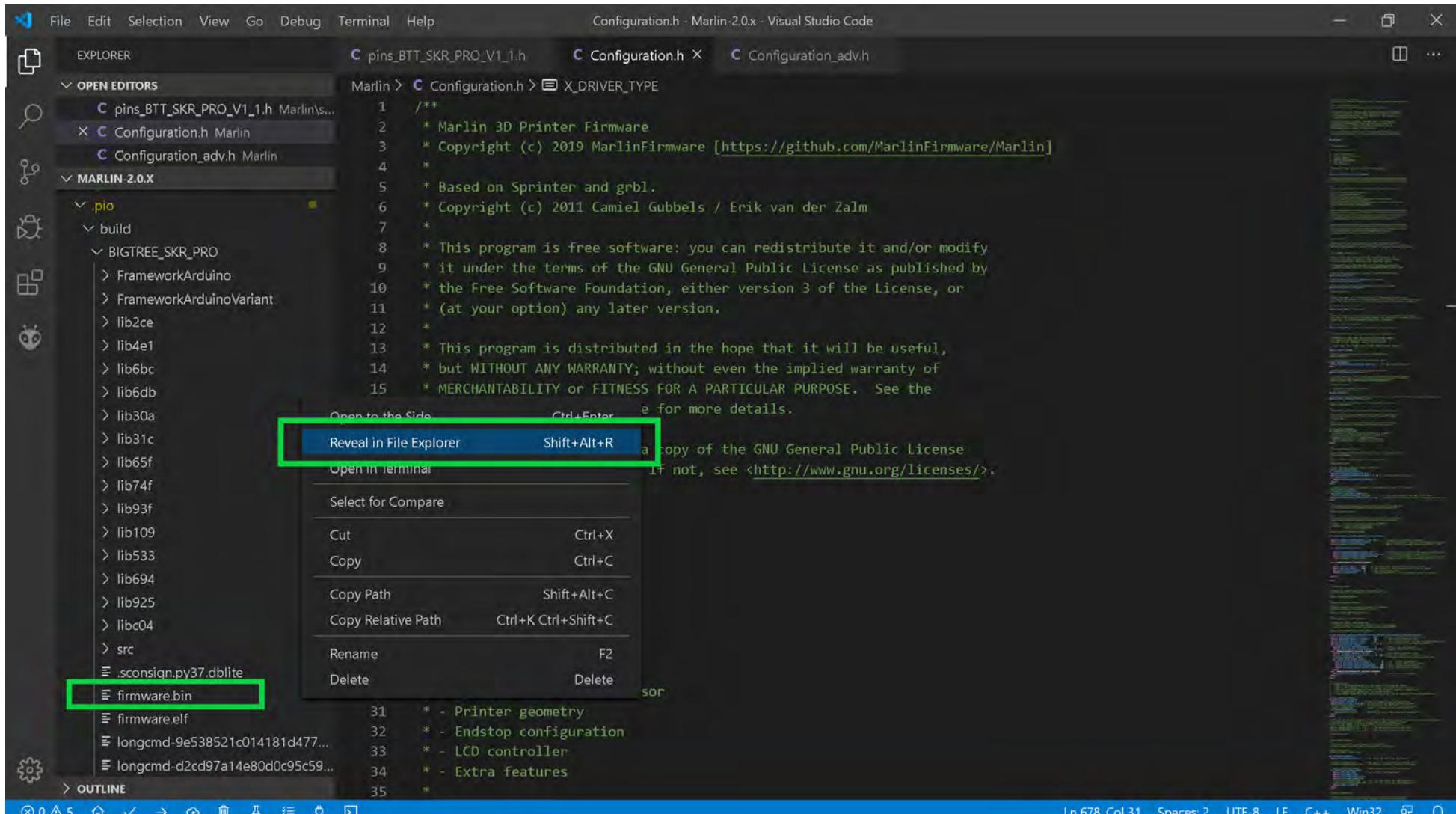
Configuration.h - Marlin-2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h X Configuration_adv.h
pins_BTT_SKR_PRO_V1_1.h Marlin\...
Configuration.h Marlin
Configuration_adv.h Marlin
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + - x
CTREE_BTT_V1_0 BIGTREE_SKR_PRO
BIGTREE_BTT_V1_0 IGNORED SUCCESS 00:02:31.294
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
Terminal will be reused by tasks, press any key to close it.

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

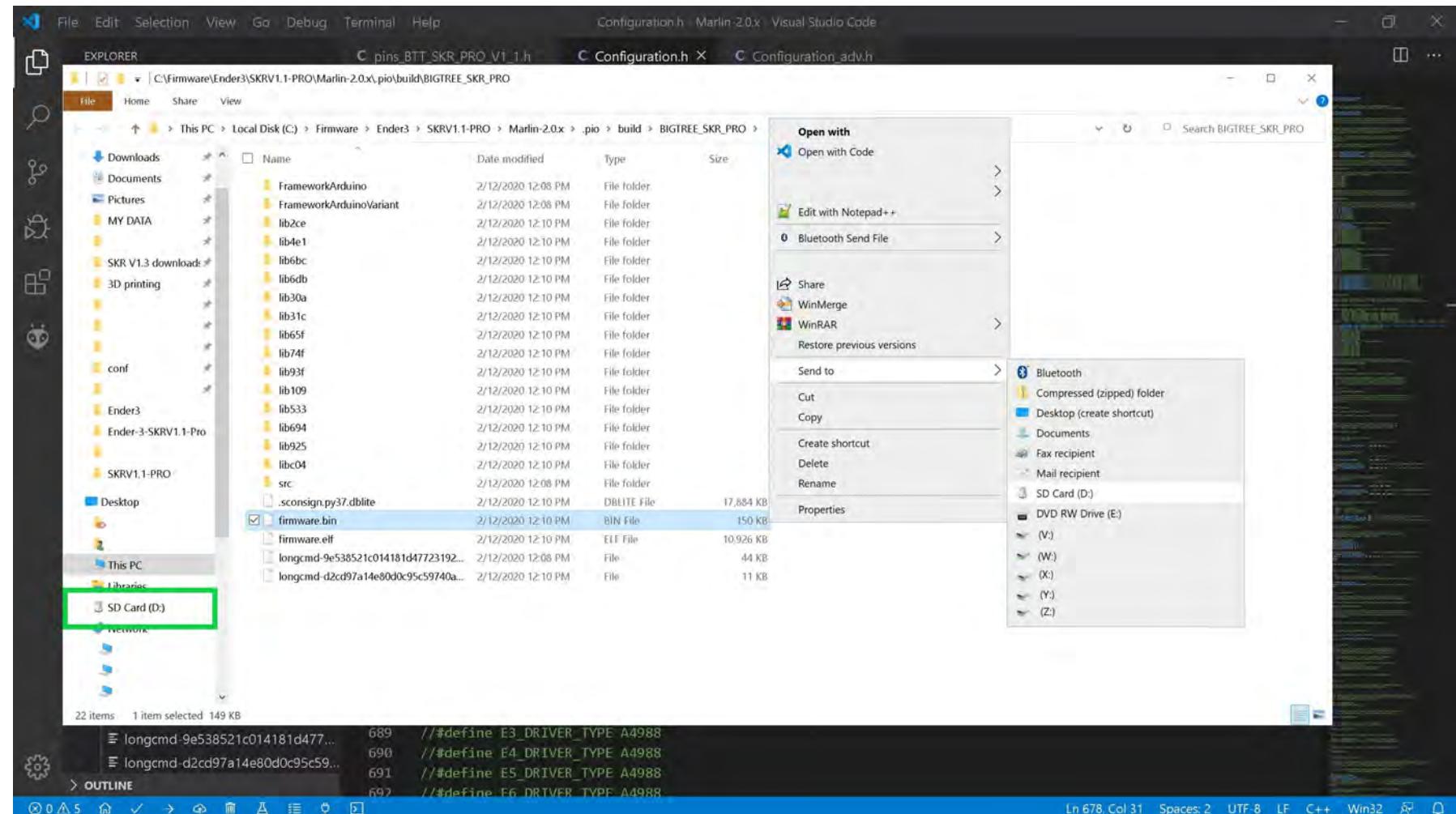
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



- Go to the next page.

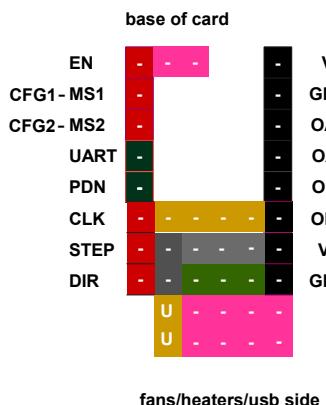
The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

UART



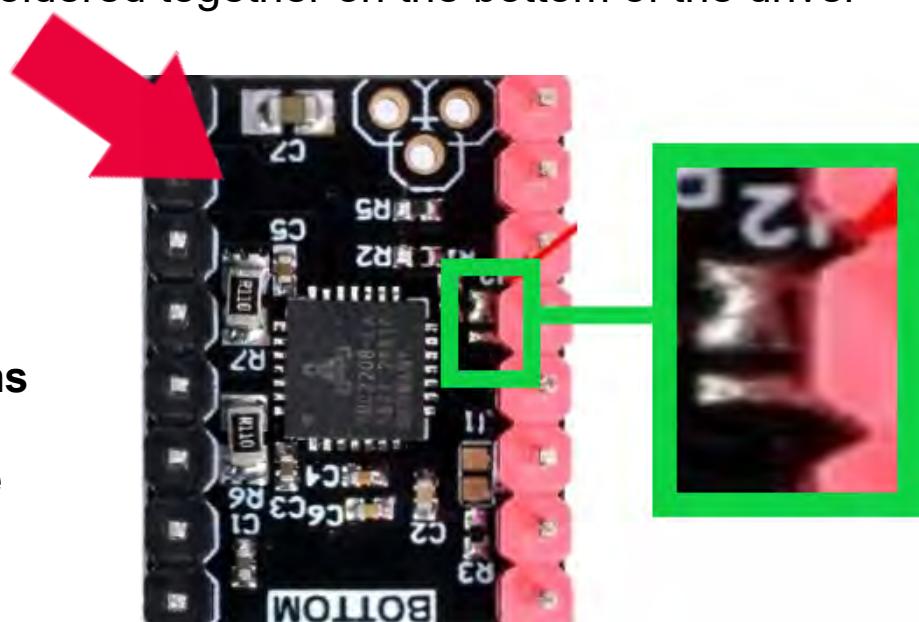
Driver Chip	
TMC2208	UART mode
5.5v - 35v DC	1.7A
Maximum subdivision	256
Driving current calculation formula	$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$
	$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$

Steps are set inside of your Firmware

Important: To ensure that the Biqu TMC2208 V3.0 is in UART Mode, check to see if two adjacent J2 pads are soldered together on the bottom of the driver board. An example is shown in the picture below.

MOST Biqu TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

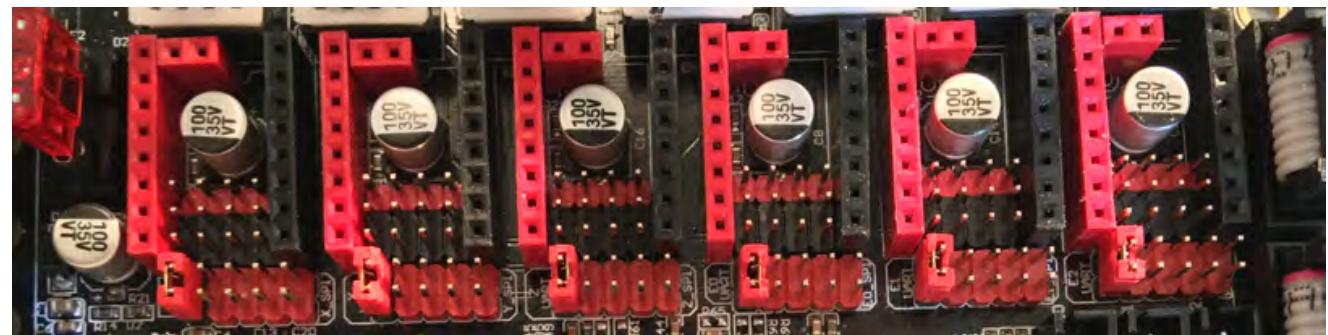
Note: You can use 50% to 90% of the calculated Irms ($I_{max}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.



UART

UART	
EN	-
CFG1 - MS1	-
CFG2 - MS2	-
UART	-
PDN	-
CLK	-
STEP	-
DIR	-
U	VM
U	GND
	OA2
	OA1
	OB1
	OB2
	VIO
	GND

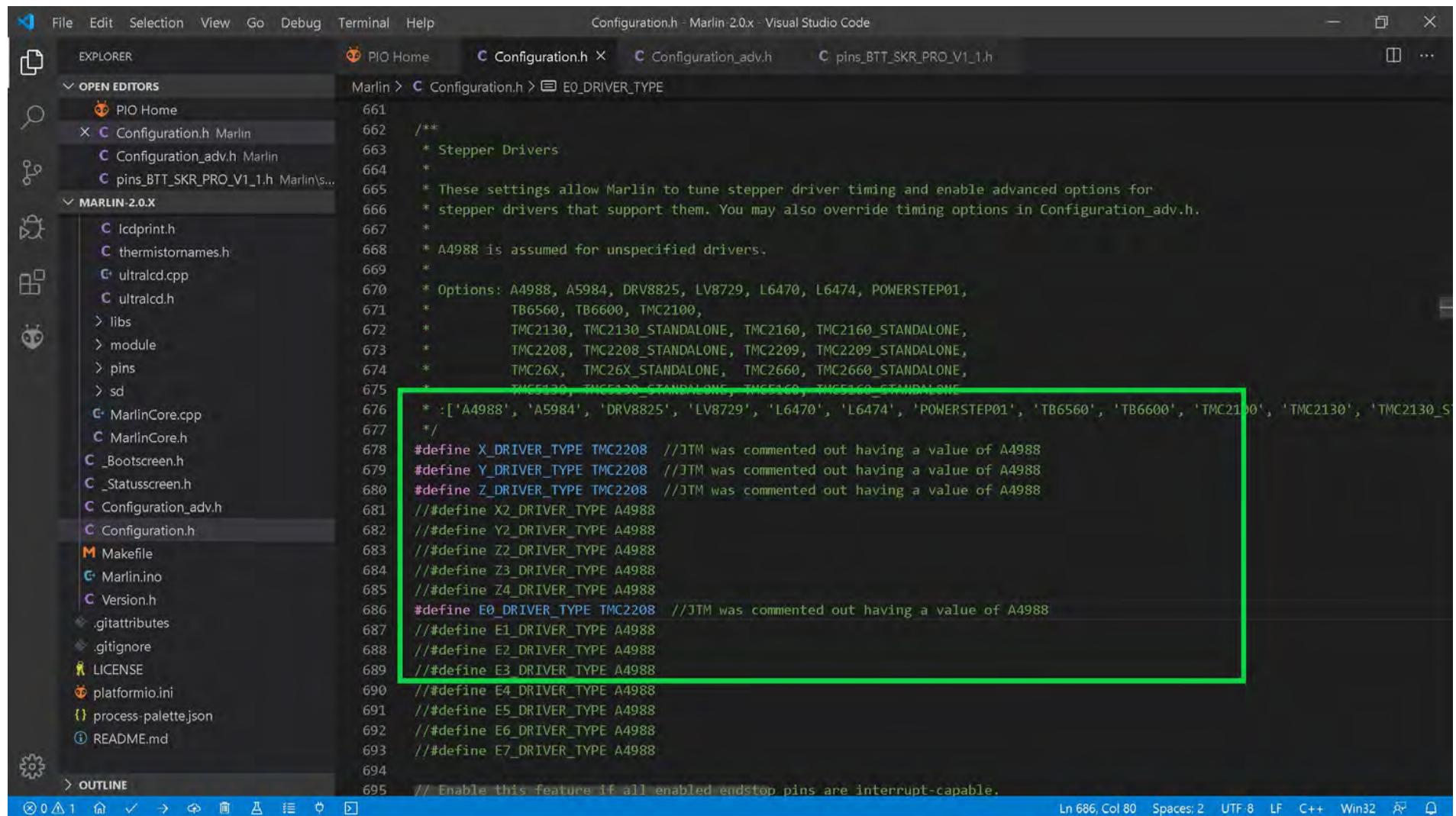
Note: Set Jumper "U" for UART MODE!!



The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2208 V3.0 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

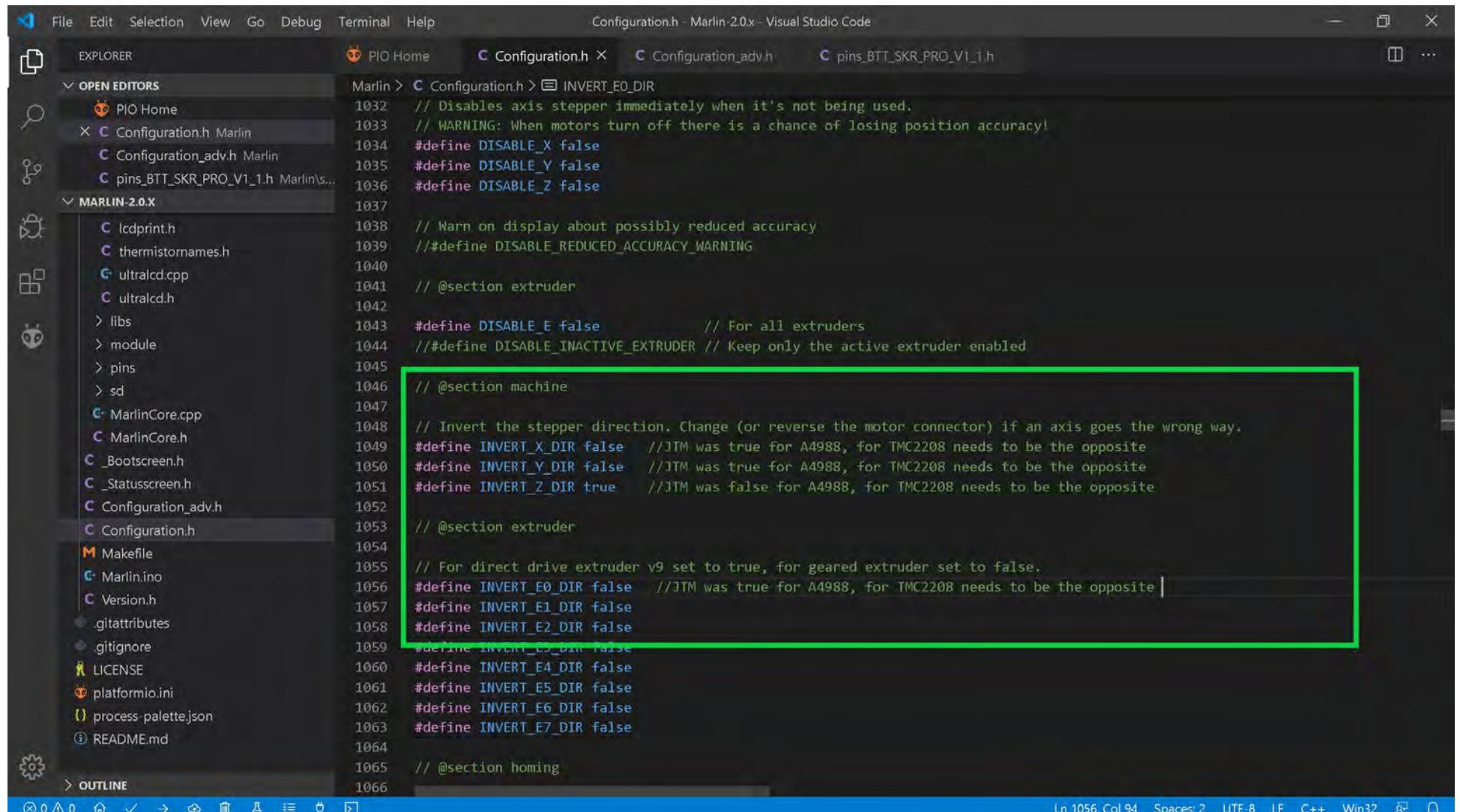
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS
Marlin > Configuration.h > E0_DRIVER_TYPE
661
662 //**
663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160_S
677 */
#define X_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define Y_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define Z_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define Z3_DRIVER_TYPE A4988
#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2208 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // JTM was true for A4988, for TMC2208 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

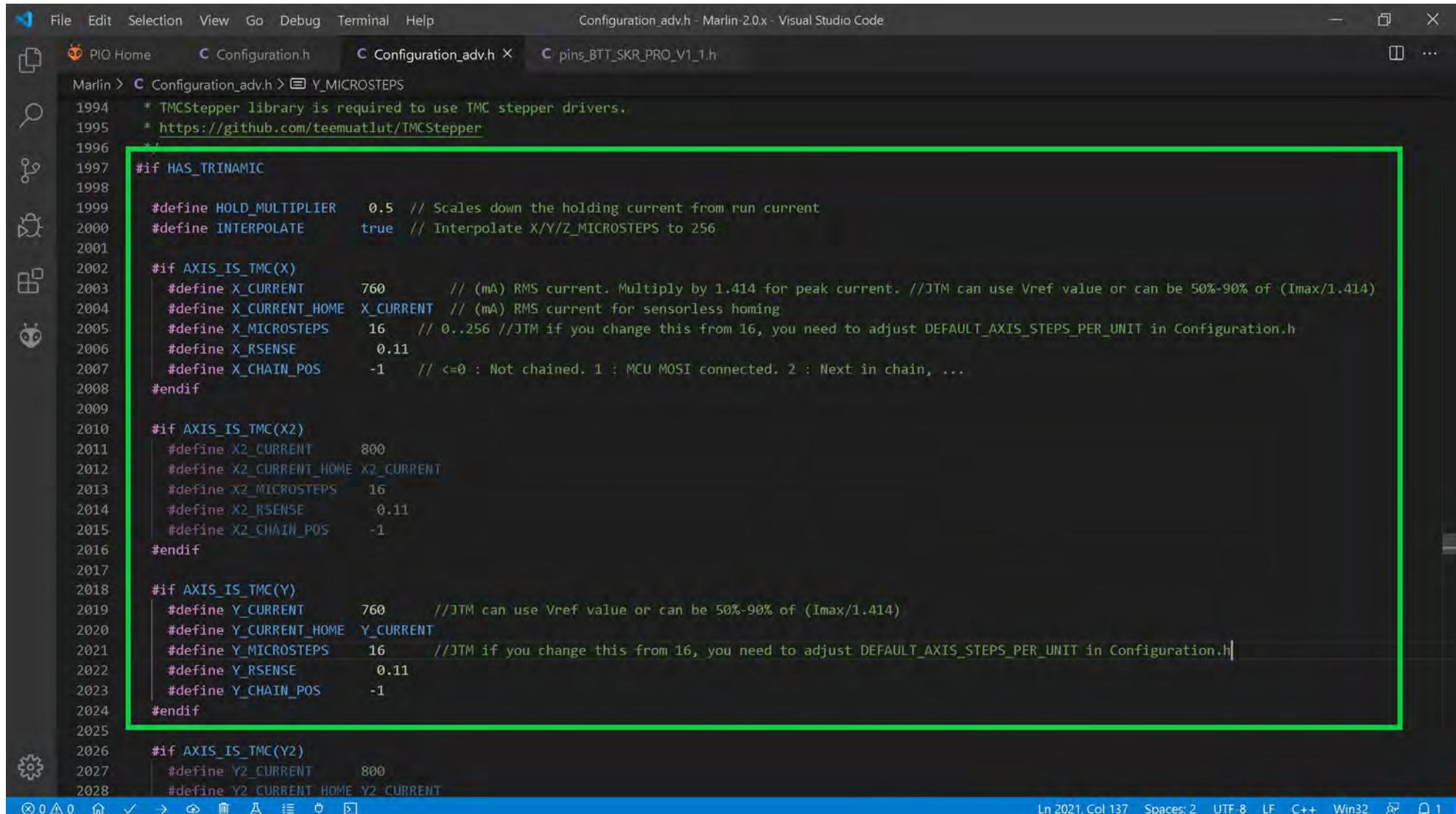
```

Ln 1056 Col 94 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- next you want to set your Vref in the Marlin firmware for each axis that has the TMC2208 driver, as seen in the GREEN box below. I changed the "X_CURRENT" to be the calculated Vref for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated Vref for my Y-Axis, which is 760mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration_adv.h > Y_MICROSTEPS
1994 * TMCSStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCSStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999     #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000     #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002     #if AXIS_IS_TMC(X)
2003         #define X_CURRENT          760 // (mA) RMS current. Multiply by 1.414 for peak current. //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2004         #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005         #define X_MICROSTEPS      16 // 0..256 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2006         #define X_RSENSE           0.11
2007         #define X_CHAIN_POS        -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008     #endif
2009
2010     #if AXIS_IS_TMC(X2)
2011         #define X2_CURRENT         800
2012         #define X2_CURRENT_HOME   X2_CURRENT
2013         #define X2_MICROSTEPS     16
2014         #define X2_RSENSE          0.11
2015         #define X2_CHAIN_POS       -1
2016     #endif
2017
2018     #if AXIS_IS_TMC(Y)
2019         #define Y_CURRENT          760 //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2020         #define Y_CURRENT_HOME    Y_CURRENT
2021         #define Y_MICROSTEPS      16 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2022         #define Y_RSENSE           0.11
2023         #define Y_CHAIN_POS        -1
2024     #endif
2025
2026     #if AXIS_IS_TMC(Y2)
2027         #define Y2_CURRENT         800
2028         #define Y2_CURRENT_HOME   Y2_CURRENT

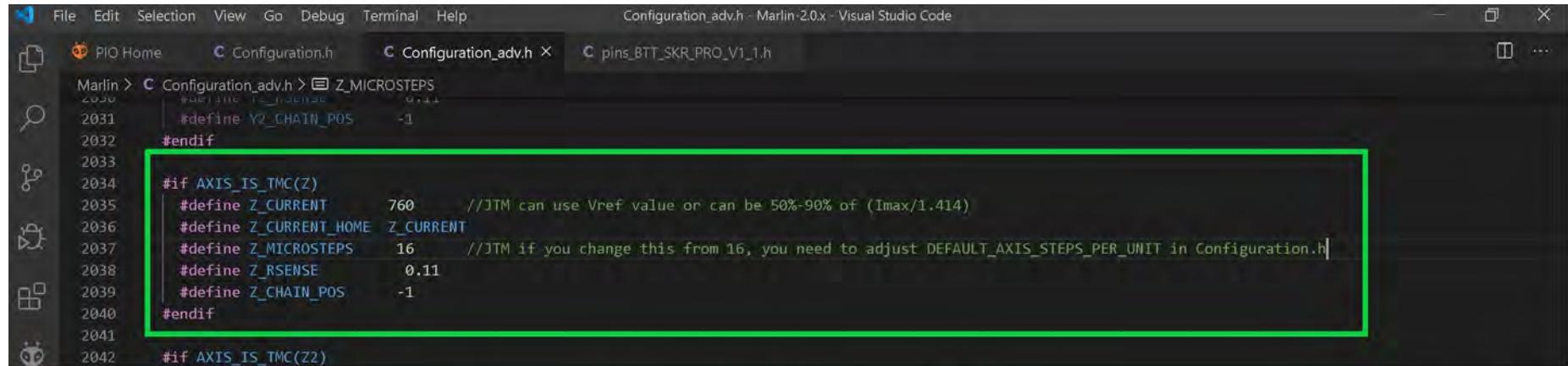
```

Ln 2021, Col 137 Spaces:2 UTF-8 LF C++ Win32 ⌂ 1

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

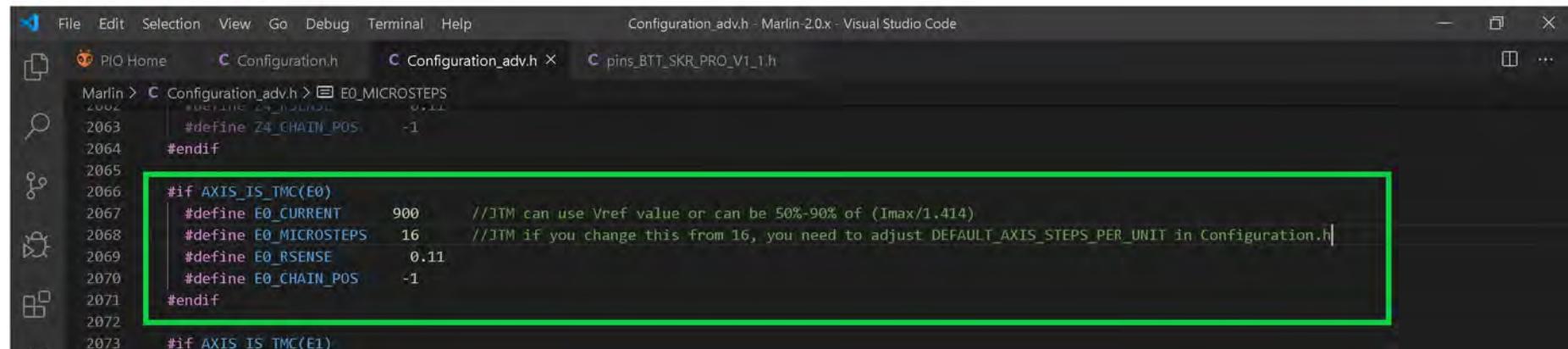
- Now, I am setting the Vref for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated Vref for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated Vref for my Extruder, which is 900mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > Z_MICROSTEPS
2030
2031     #define Z2_CHAIN_POS -1
2032
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT    760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2038         #define Z_RSENSE   0.11
2039         #define Z_CHAIN_POS -1
2040     #endif
2041
2042     #if AXIS_IS_TMC(Z2)

```

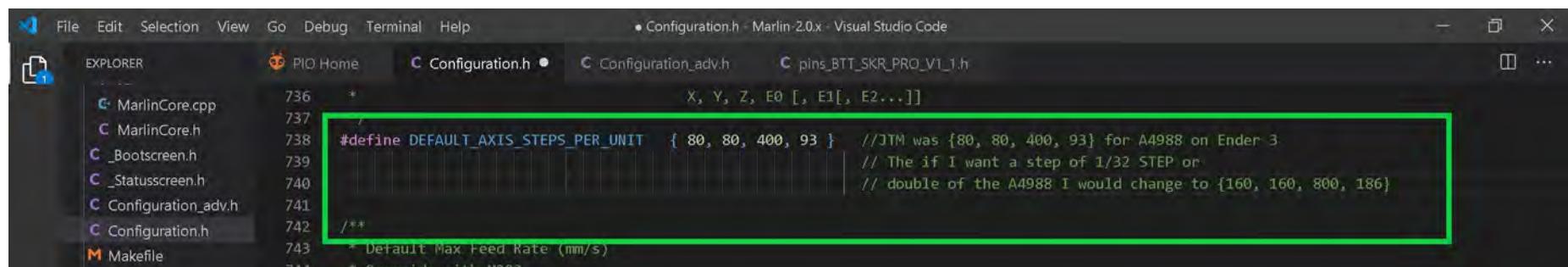


```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > E0_MICROSTEPS
2062
2063     #define Z4_CHAIN_POS -1
2064
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT    900      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2068         #define E0_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2069         #define E0_RSENSE   0.11
2070         #define E0_CHAIN_POS -1
2071     #endif
2072
2073     #if AXIS_IS_TMC(E1)

```

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER
MarlinCore.cpp 736     *
MarlinCore.h 737     /
Bootscreen.h 738     #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
_Statusscreen.h 739     // The if I want a step of 1/32 STEP or
Configuration_adv.h 740     // double of the A4988 I would change to {160, 160, 800, 186}
Configuration.h 741     /**
Makefile 742     /**
743     * Default Max Feed Rate (mm/s)
744     * Override with M203

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if the proper "CHOPPER_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER_TIMING" is correct.

```

File Edit Selection View Go Debug Terminal Help • Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

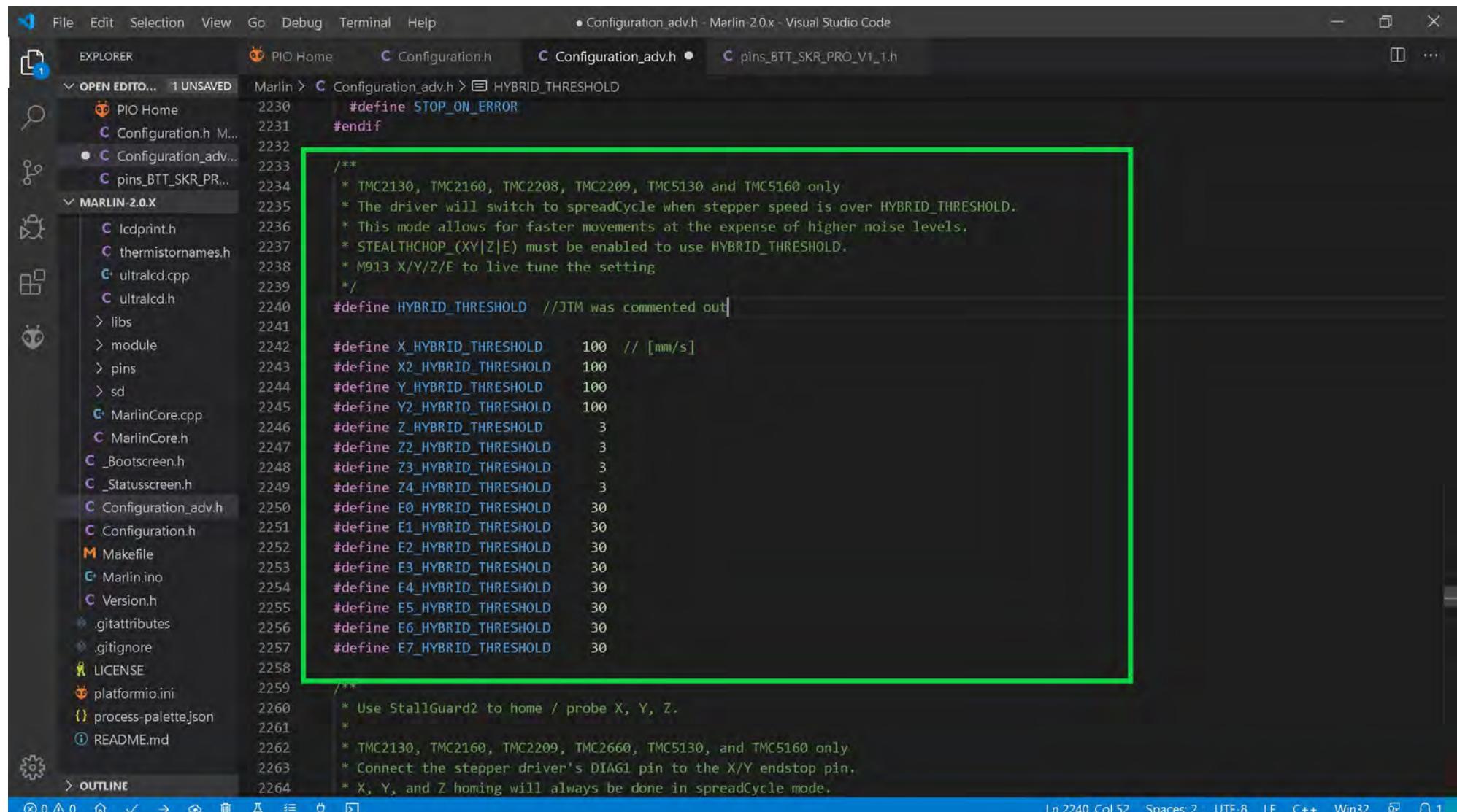
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITOR 1 UNSAVED Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
PIO Home 2194
Configuration.h M... 2195
● Configuration_adv... 2196
pins_BTT_SKR_PR... 2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213 /**
2214 * Monitor Trinamic drivers for error conditions,
2215 * like overtemperature and short to ground.
2216 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2217

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays the Marlin firmware configuration. A green box highlights the `HYBRID_THRESHOLD` section. The code in this section is as follows:

```

/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //JTM was commented out

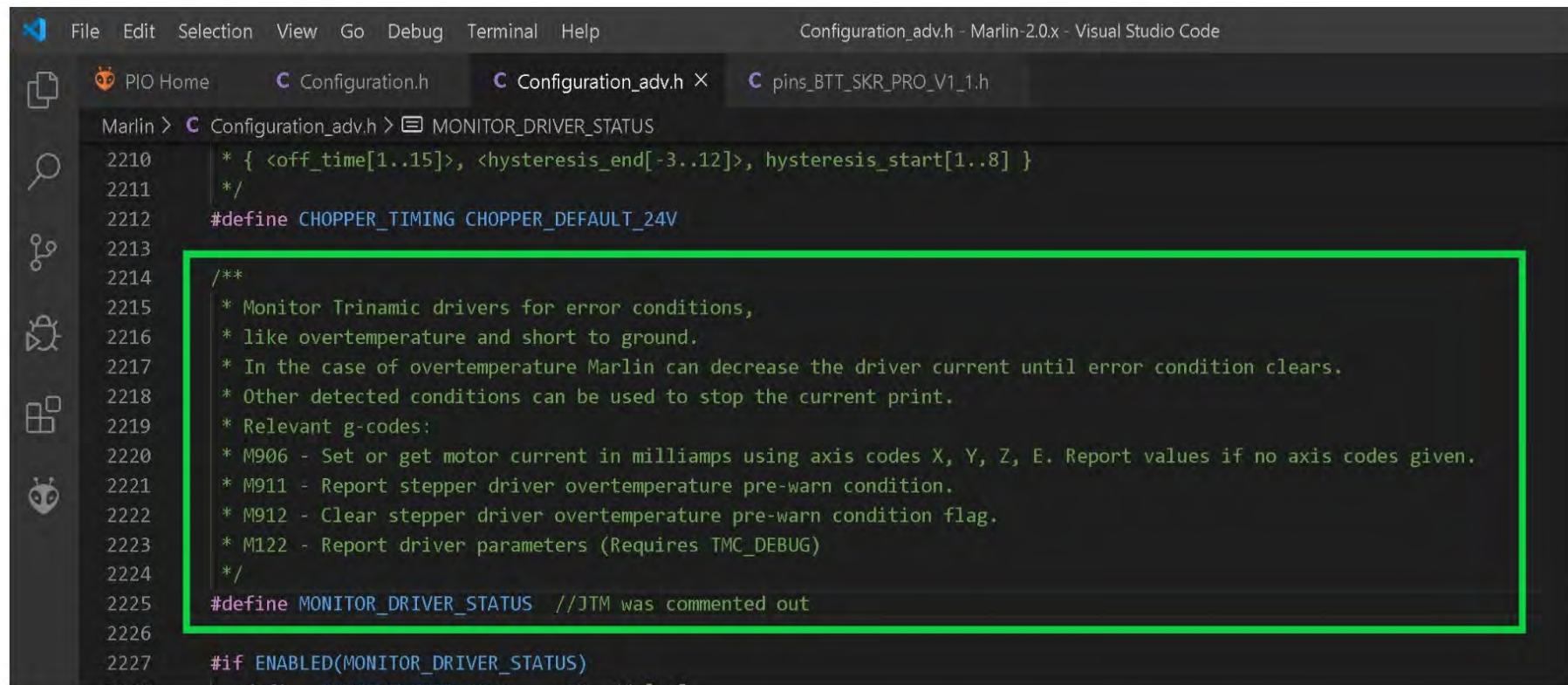
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

/*
 * Use StallGuard2 to home / probe X, Y, Z.
 *
 * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
 * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
 * X, Y, and Z homing will always be done in spreadCycle mode.
 */

```

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in Marlin's documentation located [on-line](#).
- Go to the next page.

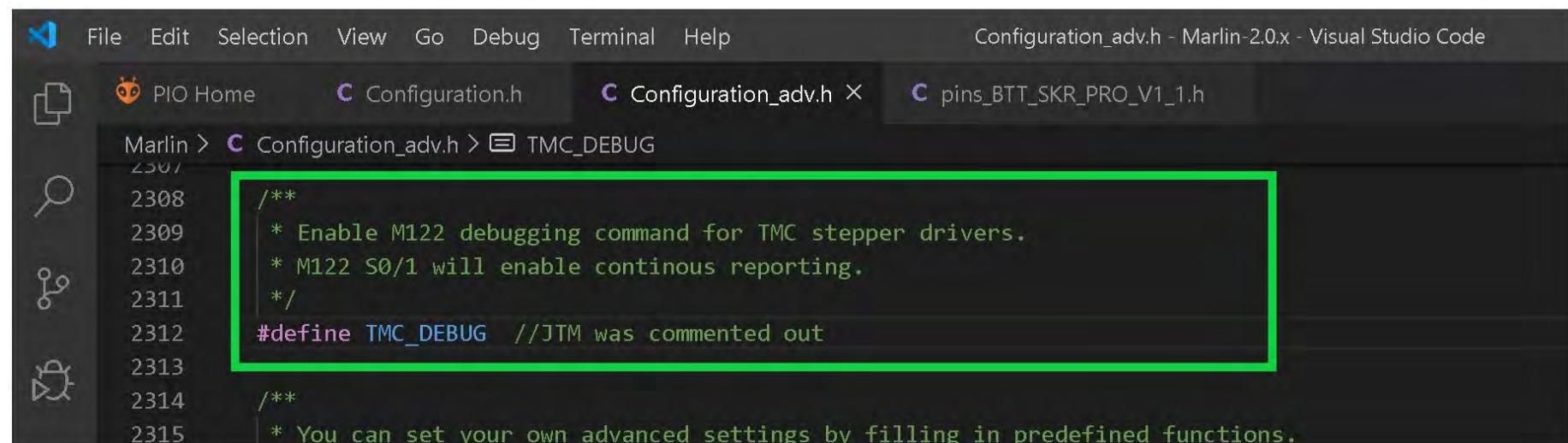
The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

```
Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
2210     * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211     */
2212     #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //JTM was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)
```



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

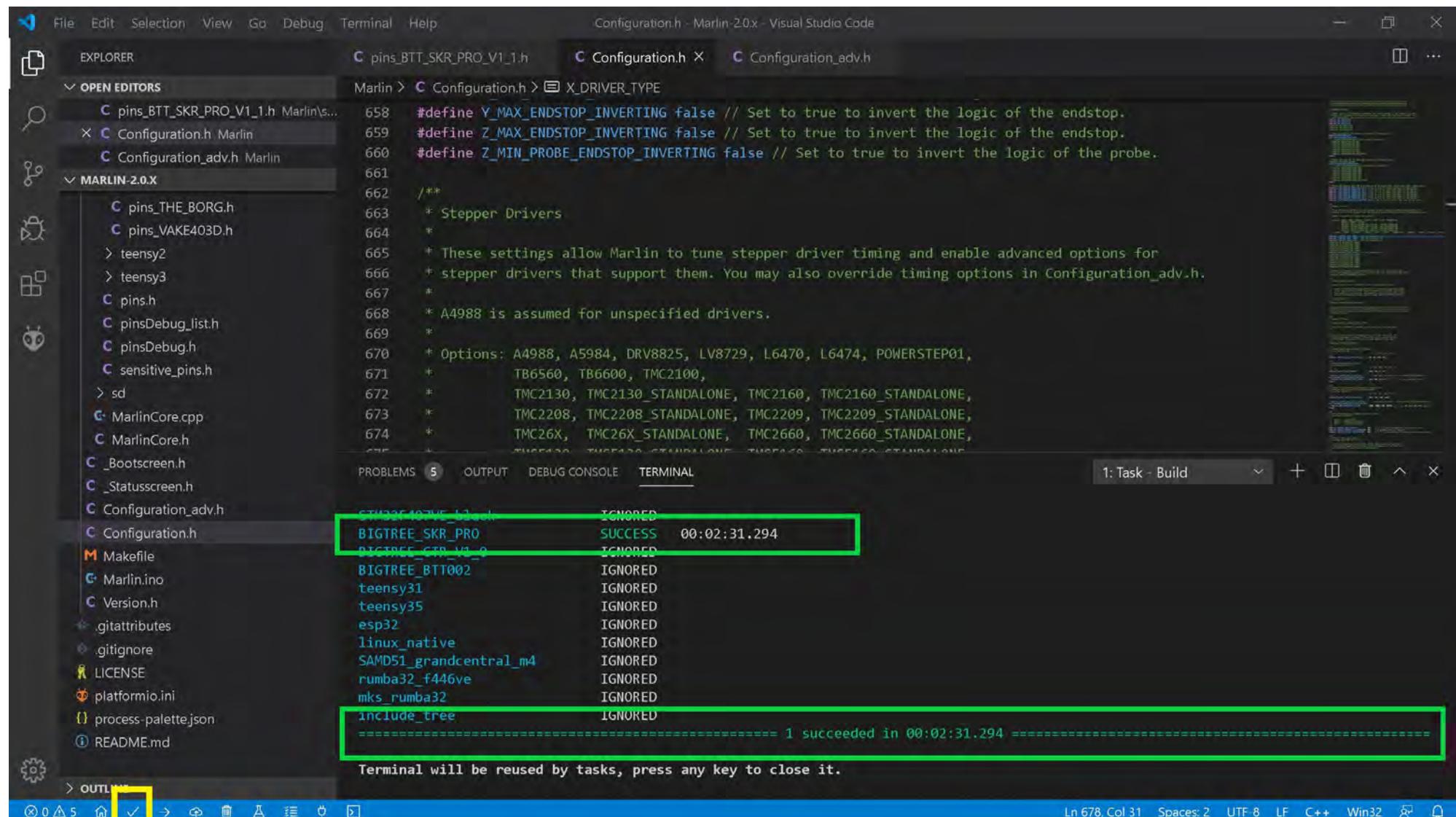
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

```
Marlin > Configuration_adv.h > TMC_DEBUG
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //JTM was commented out
2313
2314 /**
2315 * You can set your own advanced settings by filling in predefined functions.
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

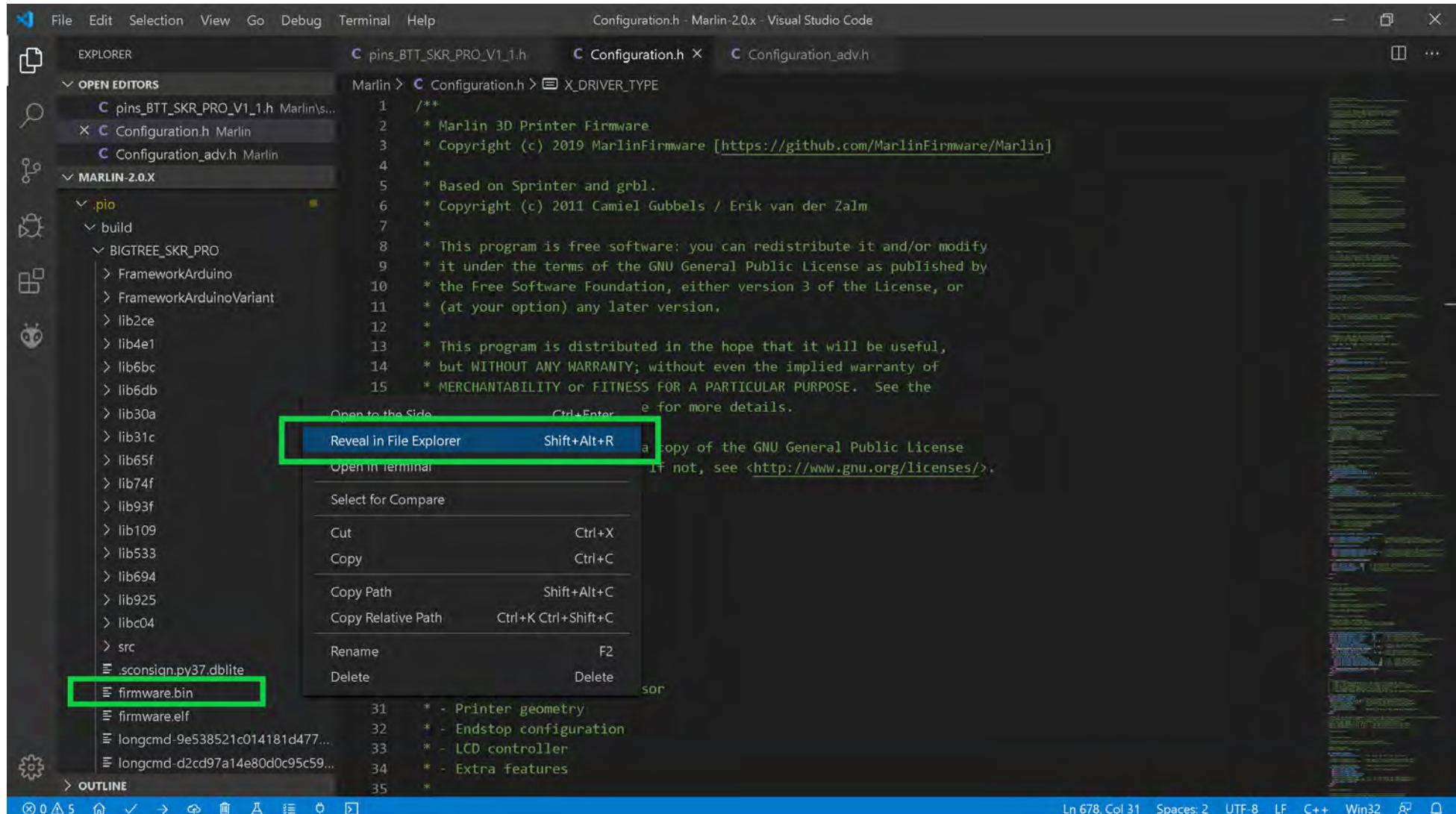
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin's...
X Configuration.h Marlin
C Configuration_adv.h Marlin
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palettejson
README.md
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + - x
BIGTREETECH_SKR_PRO SUCCESS 00:02:31.294
BIGTREETECH_SKR_V1_2 IGNORED
BIGTREETECH_BT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
===== 1 succeeded in 00:02:31.294 =====
Terminal will be reused by tasks, press any key to close it.
Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

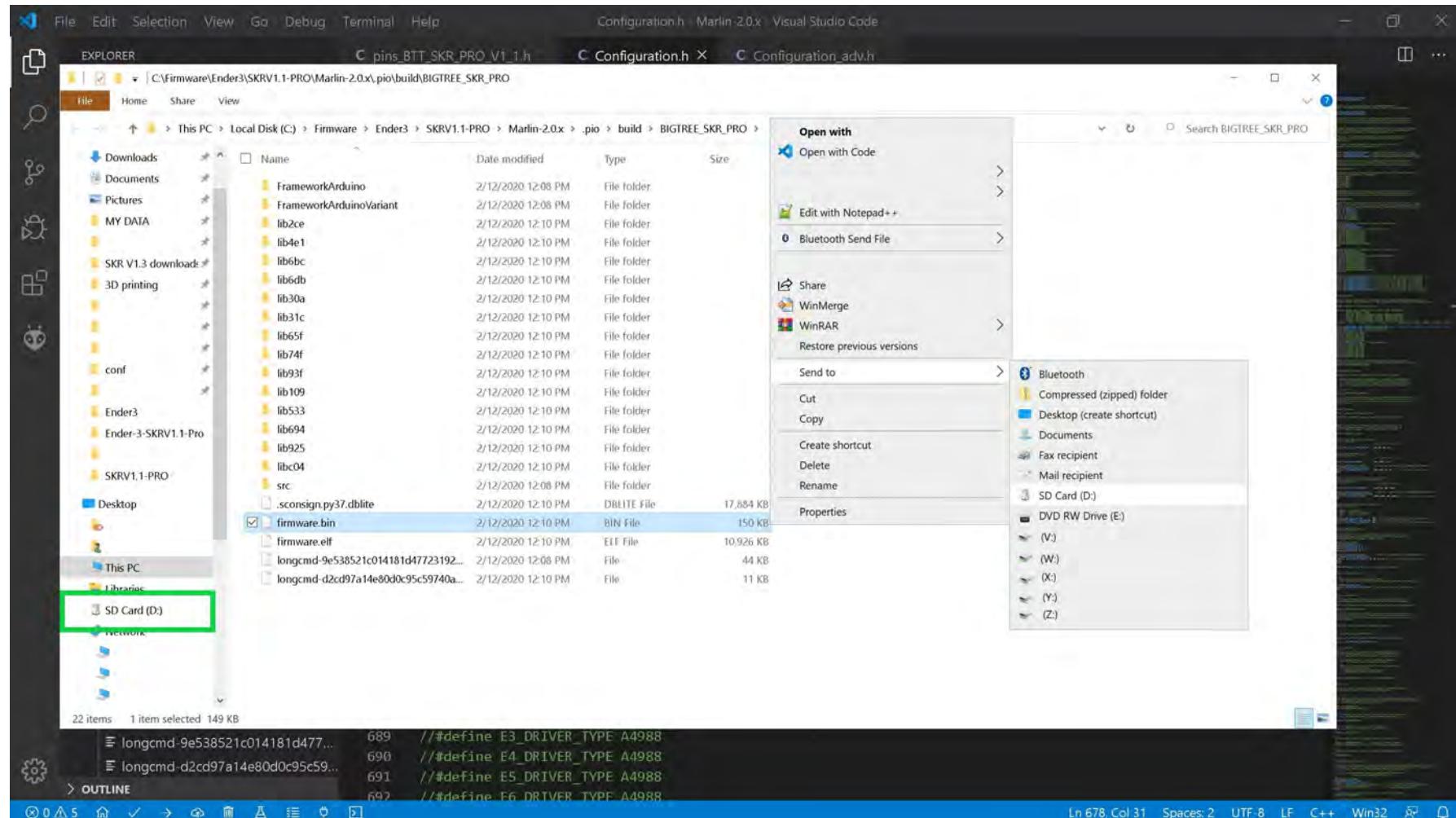
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".

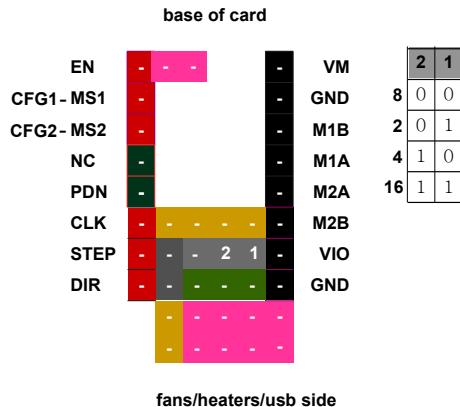


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

FYSETC TMC2208 V1.2

Stand Alone Mode

Stand Alone Mode



Driver Chip	CFG2/MS2	CFG1/MS1	Steps	Interpolation	Mode
FYSETC TMC2208	GND	GND	1/8	1/256	stealthChop
	GND	VIO	1/2	1/256	stealthChop
	VIO	GND	1/4	1/256	stealthChop
	VIO	VIO	1/16	1/256	stealthChop

FYSETC TMC2208
Standalone Mode
5.5v - 36v DC 2.0A
Maximum subdivision 16

Driving current calculation formula

$$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$$

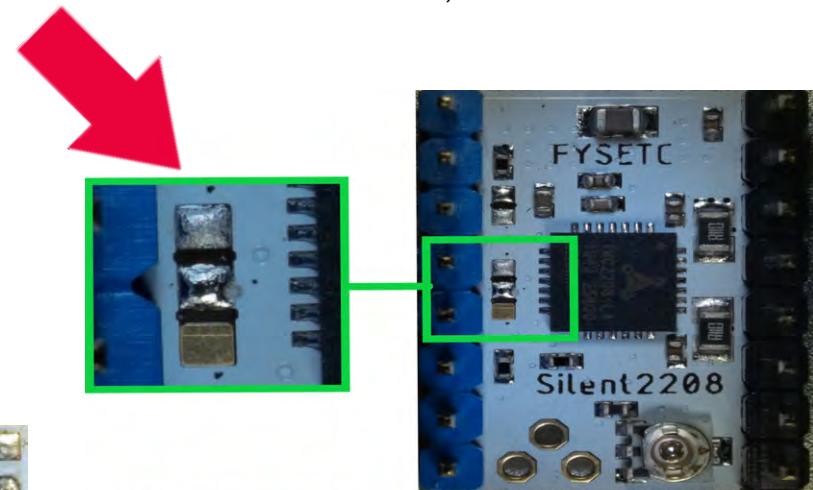
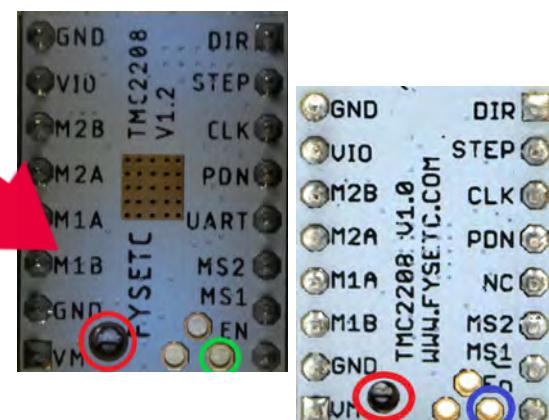
Note: Use 90% of the calculated Vref when tuning the stepper driver board

$$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$$

Note: To determine if your driver board is in UART mode, check the bottom of the driver board for three pads, located in the same position as shown in the picture below. There could be two or three pads located at this position (let's call it the "J2" position)! To have the driver board **in STANDALONE MODE**, **ALL the pads at "J2" MUST NOT be connected**. Again, a gap MUST be in place between all (three or two) "J2" pads to obtain Standalone Mode for the FYSETC TMC2208 V1.X, as seen in the **GREEN** box below.

MOST FYSETC TMC2208 board are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. So **please check your boards to ensure they are in the correct mode you desire!!**

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in **RED**; OR use the "Vref Test point" location on the top of the driver board, as shown in **BLUE**. See the appendix for instructions on how to set the Vref on a driver board

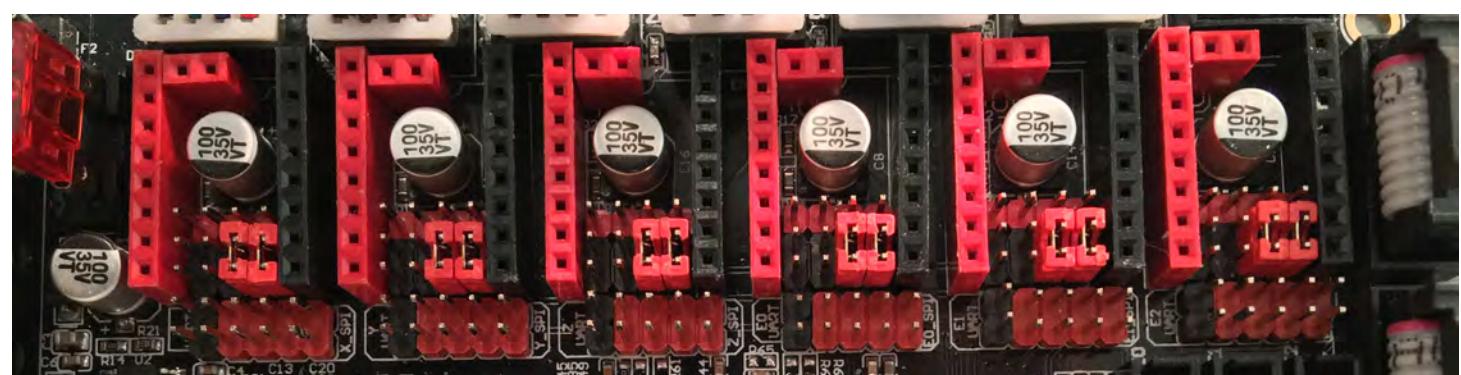


FYSETC TMC2208 V1.2 for Standalone Mode

FYSETC TMC2208 V1.2**Stand Alone Mode****Stand Alone Mode****Stand Alone Mode**

EN	-	-	VM	-
CFG1 - MS1	-	-	GND	-
CFG2 - MS2	-	-	M1B	-
NC	-	-	M1A	-
PDN	-	-	M2A	-
CLK	-	-	M2B	-
STEP	-	-	VIO	-
DIR	-	2 1	GND	-

1/8

Interpolation: 1/256
StealthChop**Stand Alone Mode**

EN	-	-	VM	-
CFG1 - MS1	-	-	GND	-
CFG2 - MS2	-	-	M1B	-
NC	-	-	M1A	-
PDN	-	-	M2A	-
CLK	-	1	M2B	-
STEP	-	2 1	VIO	-
DIR	-	2	GND	-

1/2

Interpolation: 1/256
StealthChop

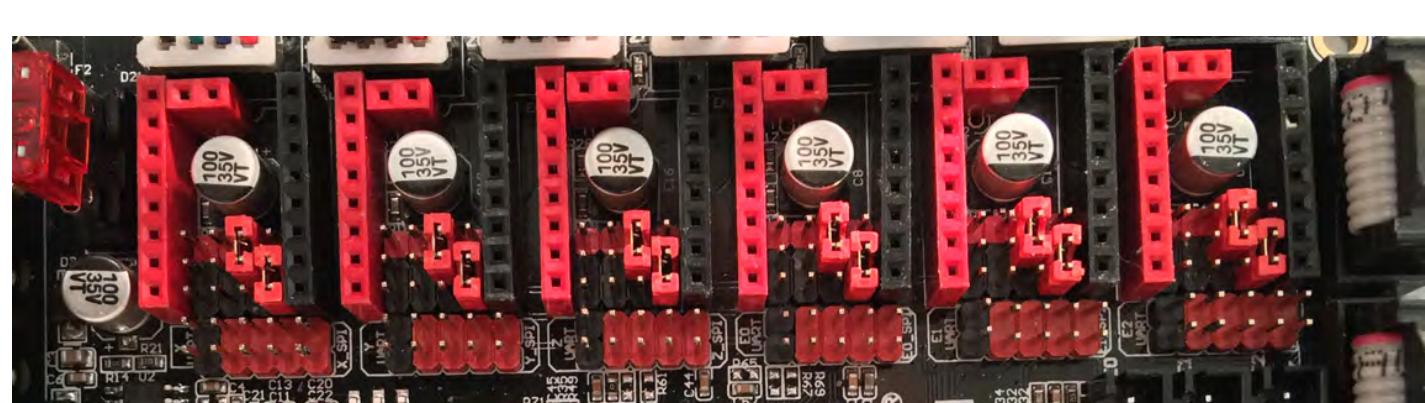
FYSETC TMC2208 V1.2

Stand Alone Mode

Stand Alone Mode

Stand Alone Mode	
EN	- - -
CFG1 - MS1	- - -
CFG2 - MS2	- - -
NC	- - -
PDN	- - -
CLK	- 2 -
STEP	- 2 1 -
DIR	- - 1 -

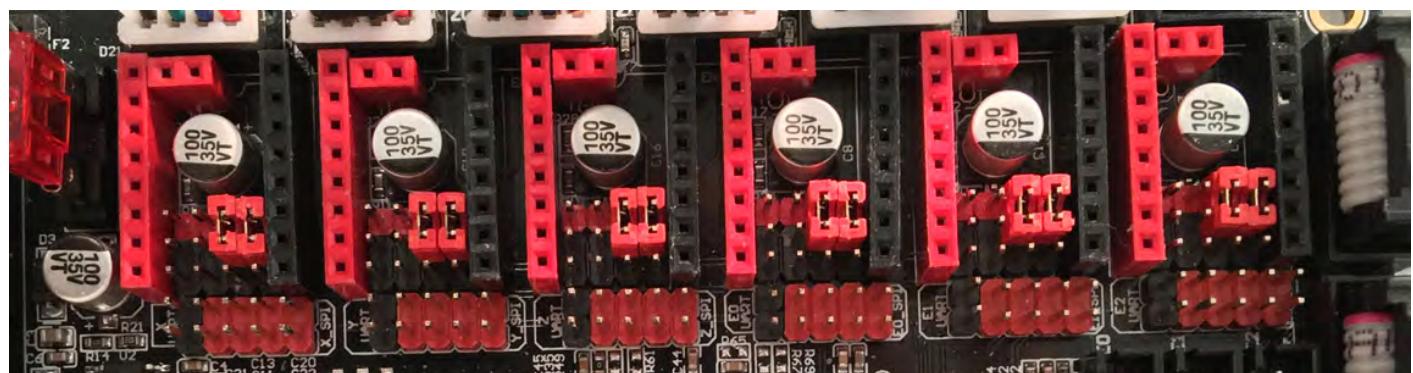
1/4

Interpolation: 1/256
StealthChop

Stand Alone Mode

Stand Alone Mode	
EN	- - -
CFG1 - MS1	- - -
CFG2 - MS2	- - -
NC	- - -
PDN	- - -
CLK	- 2 2
STEP	- 2 1 -
DIR	- - - -

1/16

Interpolation: 1/256
StealthChop

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for FYSETC TMC2208 V1.2 stepper motor drivers in stand alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

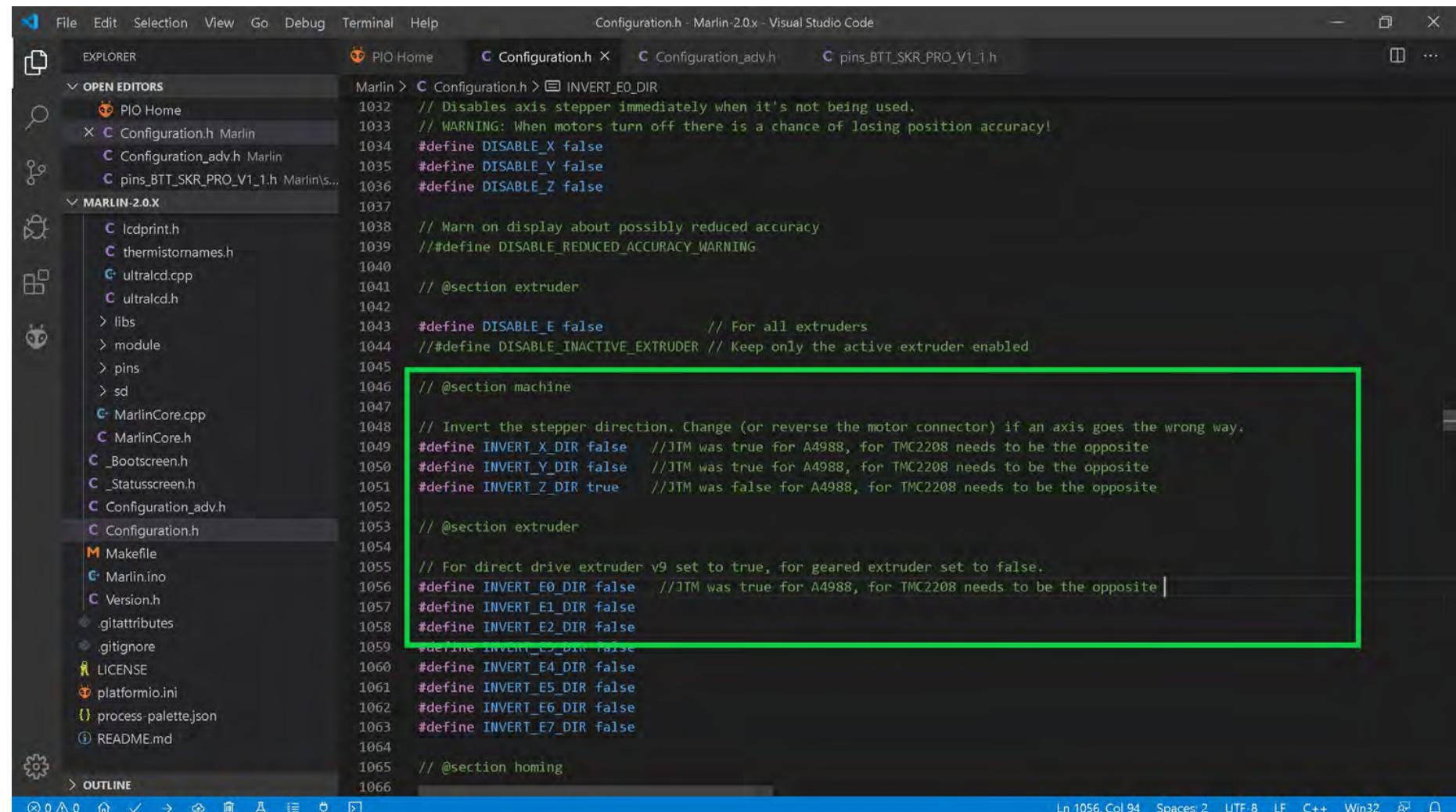
```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS
Marlin > Configuration.h > E0_DRIVER_TYPE
  661 /**
  662 * Stepper Drivers
  663 *
  664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  666 *
  667 * A4988 is assumed for unspecified drivers.
  668 *
  669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
  670 *           TB6560, TB6600, TMC2100,
  671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
  672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
  673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
  674 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
  675 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160_S
  676 */
  677 #define X_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
  678 #define Y_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
  679 #define Z_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
  680 //#define X2_DRIVER_TYPE A4988
  681 //#define Y2_DRIVER_TYPE A4988
  682 //#define Z2_DRIVER_TYPE A4988
  683 //#define Z3_DRIVER_TYPE A4988
  684 //#define Z4_DRIVER_TYPE A4988
  685 #define E0_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
  686 //#define E1_DRIVER_TYPE A4988
  687 //#define E2_DRIVER_TYPE A4988
  688 //#define E3_DRIVER_TYPE A4988
  689 //#define E4_DRIVER_TYPE A4988
  690 //#define E5_DRIVER_TYPE A4988
  691 //#define E6_DRIVER_TYPE A4988
  692 //#define E7_DRIVER_TYPE A4988
  693 // Enable this feature if all enabled endstop pins are interrupt-capable.
  694
  695 // NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.
  
```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037 // Warn on display about possibly reduced accuracy
1038 // #define DISABLE_REDUCED_ACCURACY_WARNING
1039 // @section extruder
1040
1041 #define DISABLE_E false // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043 // @section machine
1044 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1045 #define INVERT_X_DIR false // JTM was true for A4988, for TMC2208 needs to be the opposite
1046 #define INVERT_Y_DIR false // JTM was true for A4988, for TMC2208 needs to be the opposite
1047 #define INVERT_Z_DIR true // JTM was false for A4988, for TMC2208 needs to be the opposite
1048 // @section extruder
1049 // For direct drive extruder v9 set to true, for geared extruder set to false.
1050 #define INVERT_E0_DIR false // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_E1_DIR false
1052 #define INVERT_E2_DIR false
1053 #define INVERT_E3_DIR false
1054 #define INVERT_E4_DIR false
1055 #define INVERT_E5_DIR false
1056 #define INVERT_E6_DIR false
1057 #define INVERT_E7_DIR false
1058 // @section homing
1059
1060
1061
1062
1063
1064
1065 // @section homing
1066

```

Ln 1056, Col 94 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in stand alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

```

Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\...
Configuration.h Marlin
Configuration_adv.h Marlin
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
1 succeeded in 00:02:31.294 =====
Terminal will be reused by tasks, press any key to close it.
Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32

```

Terminal Output:

```

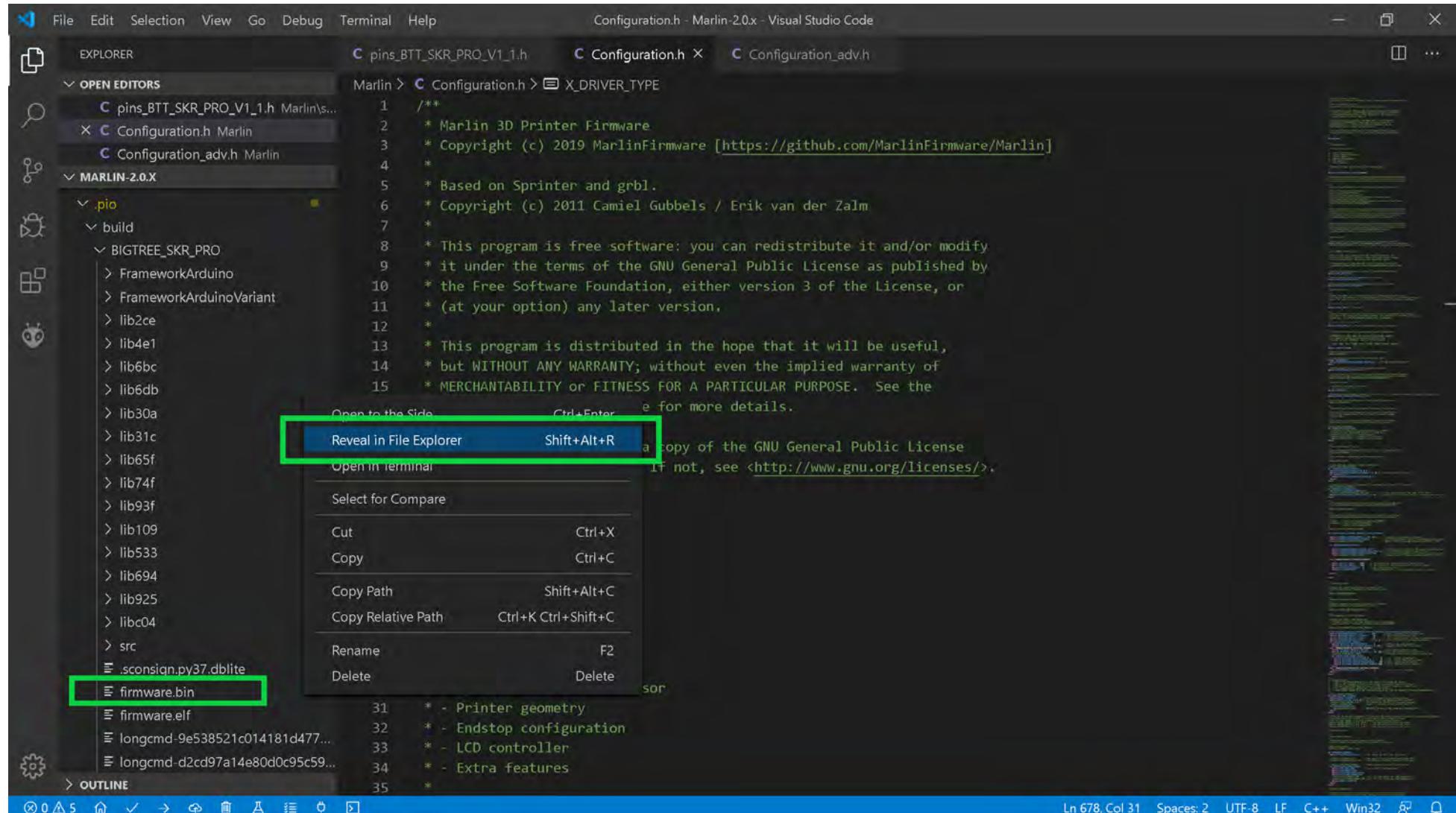
BIGTREETECH_SKR_PRO SUCCESS 00:02:31.294
BIGTREETECH_SKR_V1_2 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
===== 1 succeeded in 00:02:31.294 =====

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode

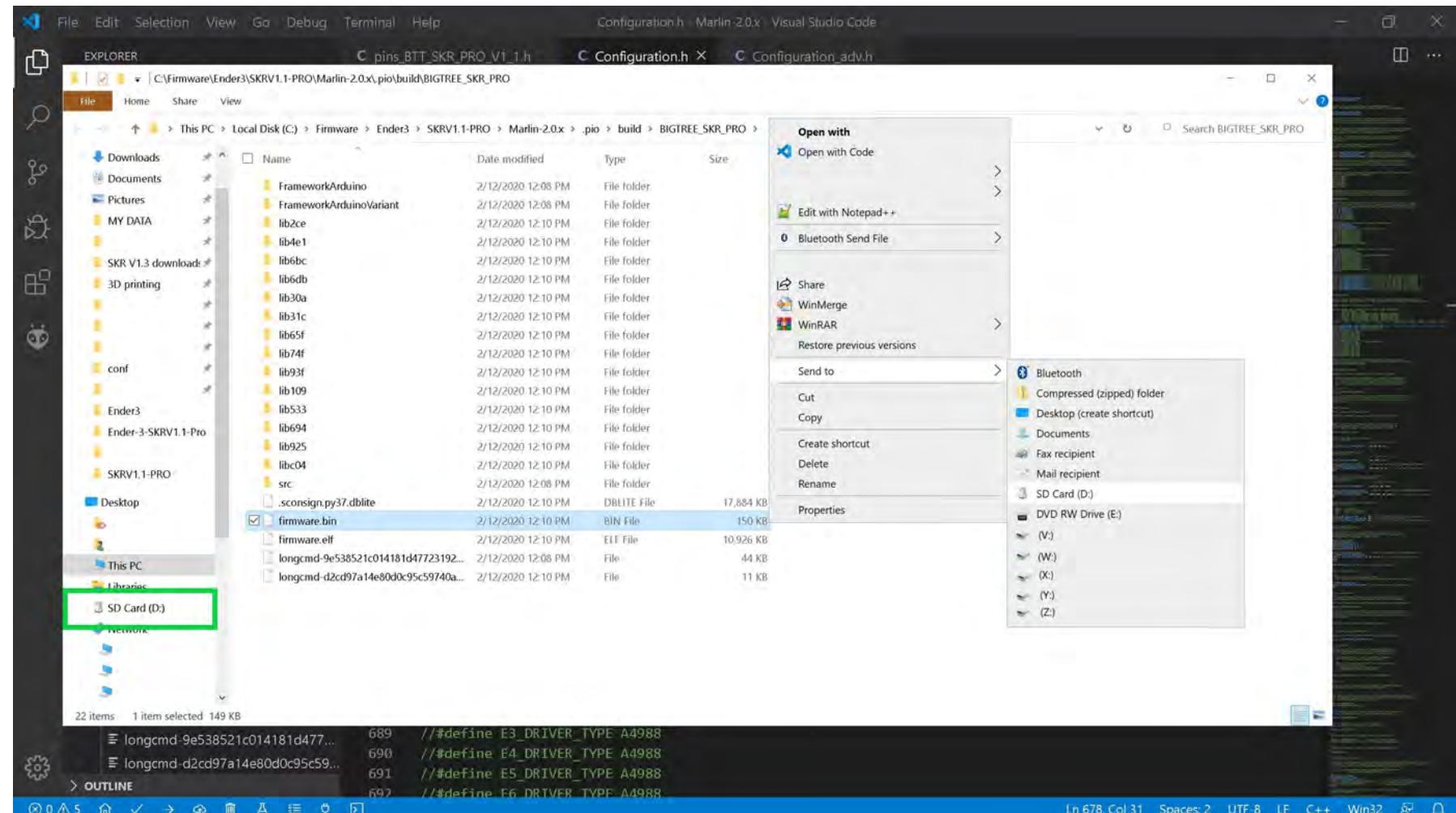
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand Alone Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".

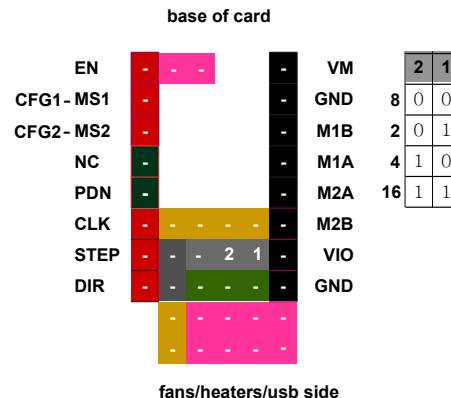


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

OTP (One Time Programming)



Driver Chip	CFG2/MS2	CFG1/MS1	Steps	Interpolation	Mode
FYSETC TMC2208	GND	GND	1/8	1/256	spreadCycle
	GND	VIO	1/2	1/256	spreadCycle
	VIO	GND	1/4	1/256	spreadCycle
	VIO	VIO	1/16	1/256	spreadCycle

OPT Mode
5.5v - 36v DC 2.0A
Maximum subdivision 16

Driving current calculation formula

$$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$$

Note: Use 90% of the calculated Vref when tuning the stepper driver board

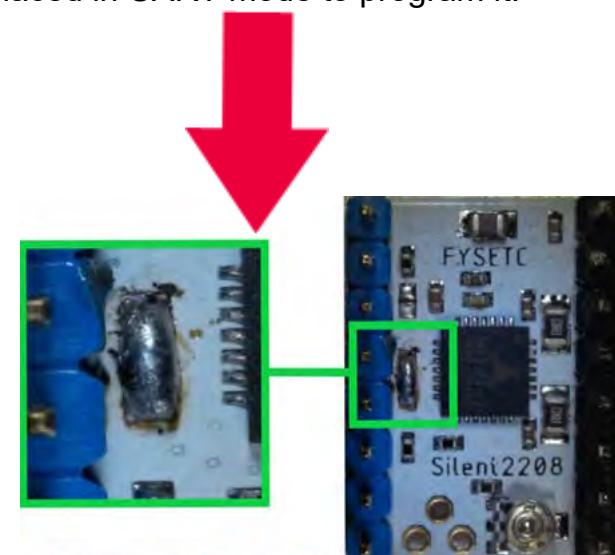
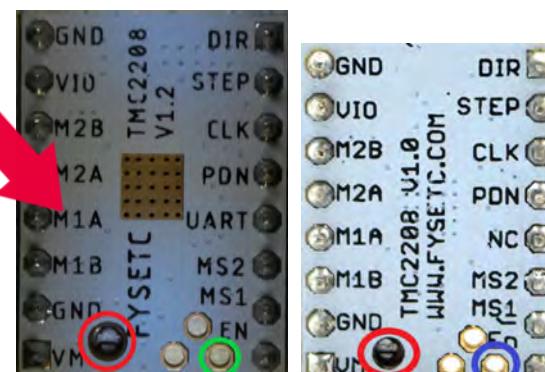
$$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$$

NOTE: Stand Alone Mode by default uses StealthChop, if you want SpreadCycle, you MUST use OTP mode. Here are the directions for running the TMC220x Configurator: <https://wiki.fysetc.com/TMC2208/#to-run-the-program>. See TMC220x Configurator for One-Time-Programming Information: [TMC220x Configurator](#).

Important: To place FYSETC TMC2208 V1.0 or V1.2 into OTP Mode you must solder all adjacent pads together on the bottom of the driver board. As an example, the picture (V1.2) below shows all the pads soldered together, as shown in GREEN. To do One-Time-Programming (OTP), the TMC2208 must be placed in UART mode to program it.

MOST FYSETC TMC2208 board are sold as V1.2 driver boards. The V1.2 driver board might be setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. So please check your boards to ensure they are in the correct mode you desire! To be able to program the Chip for OTP mode, the FYSETC TMC2208 board MUST be set in UART (all three pads bridged together)!

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in RED; OR use the "Vref Test point" location on the top of the driver board, as shown in BLUE. [See the appendix](#) for instructions on how to set the Vref on a driver board



FYSETC TMC2208 V1.2 in UART Mode

FYSETC TMC2208 V1.2

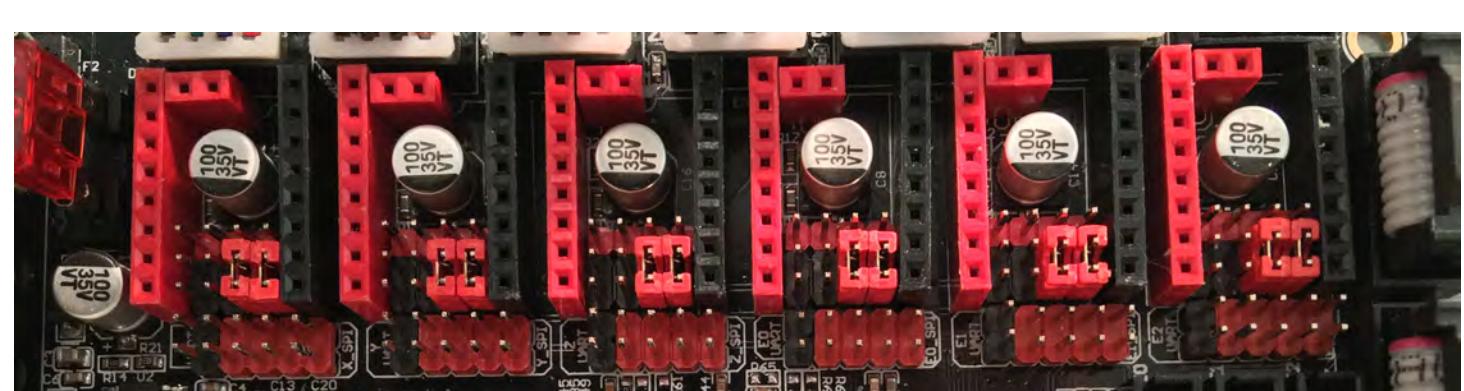
One Time Programming (OTP) Mode

OTP (One Time Programming)

OTP (One Time Programming)

EN		VM	
CFG1-MS1	-	-	-
CFG2-MS2	-	-	-
NC	-	-	-
PDN	-	-	-
CLK	-	-	-
STEP	-	2 1	-
DIR	-	2 1	-

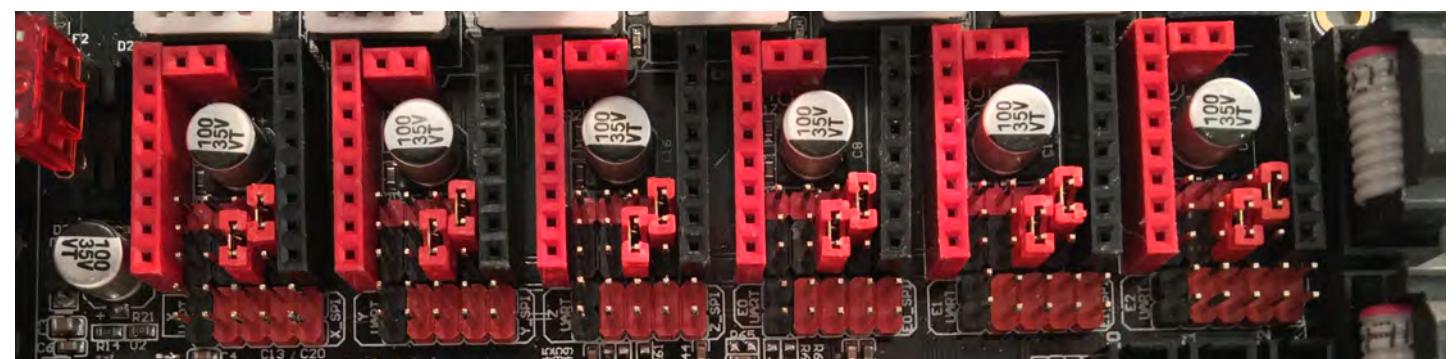
1/8
Interpolation: 1/256
SpreadCycle



OTP (One Time Programming)

EN		VM	
CFG1-MS1	-	-	-
CFG2-MS2	-	-	-
NC	-	-	-
PDN	-	-	-
CLK	-	1	-
STEP	-	2 1	-
DIR	-	2	-

1/2
Interpolation: 1/256
SpreadCycle

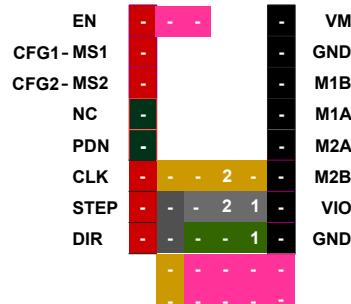


FYSETC TMC2208 V1.2

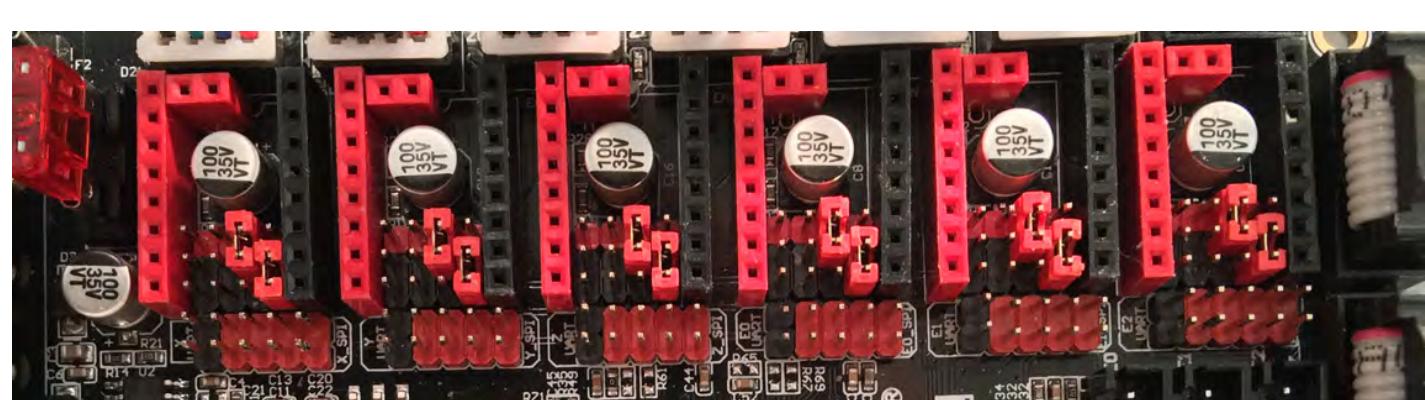
One Time Programming (OTP) Mode

OTP (One Time Programming)

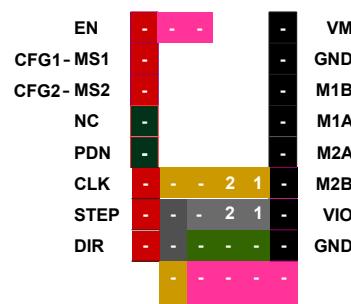
OTP (One Time Programming)



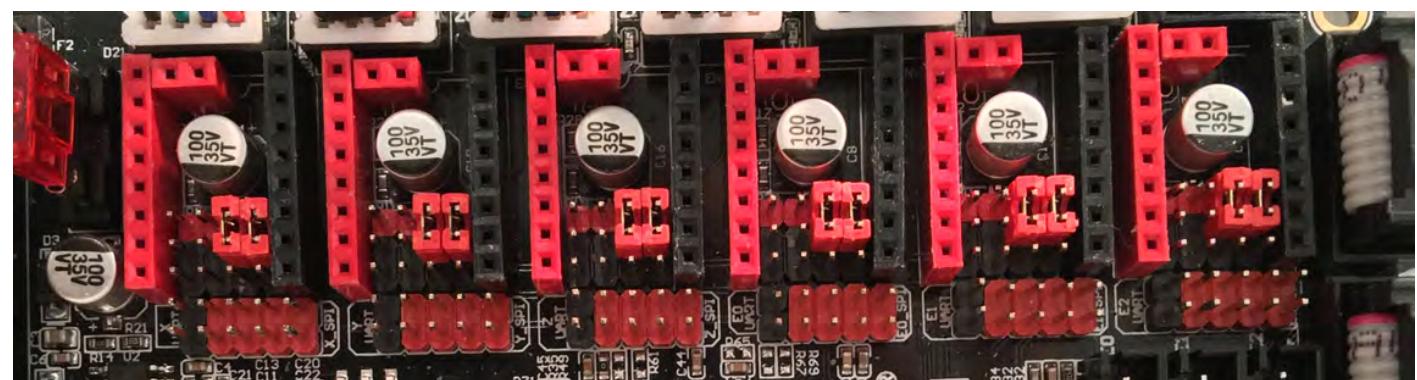
1/4
Interpolation: 1/256
SpreadCycle



OTP (One Time Programming)



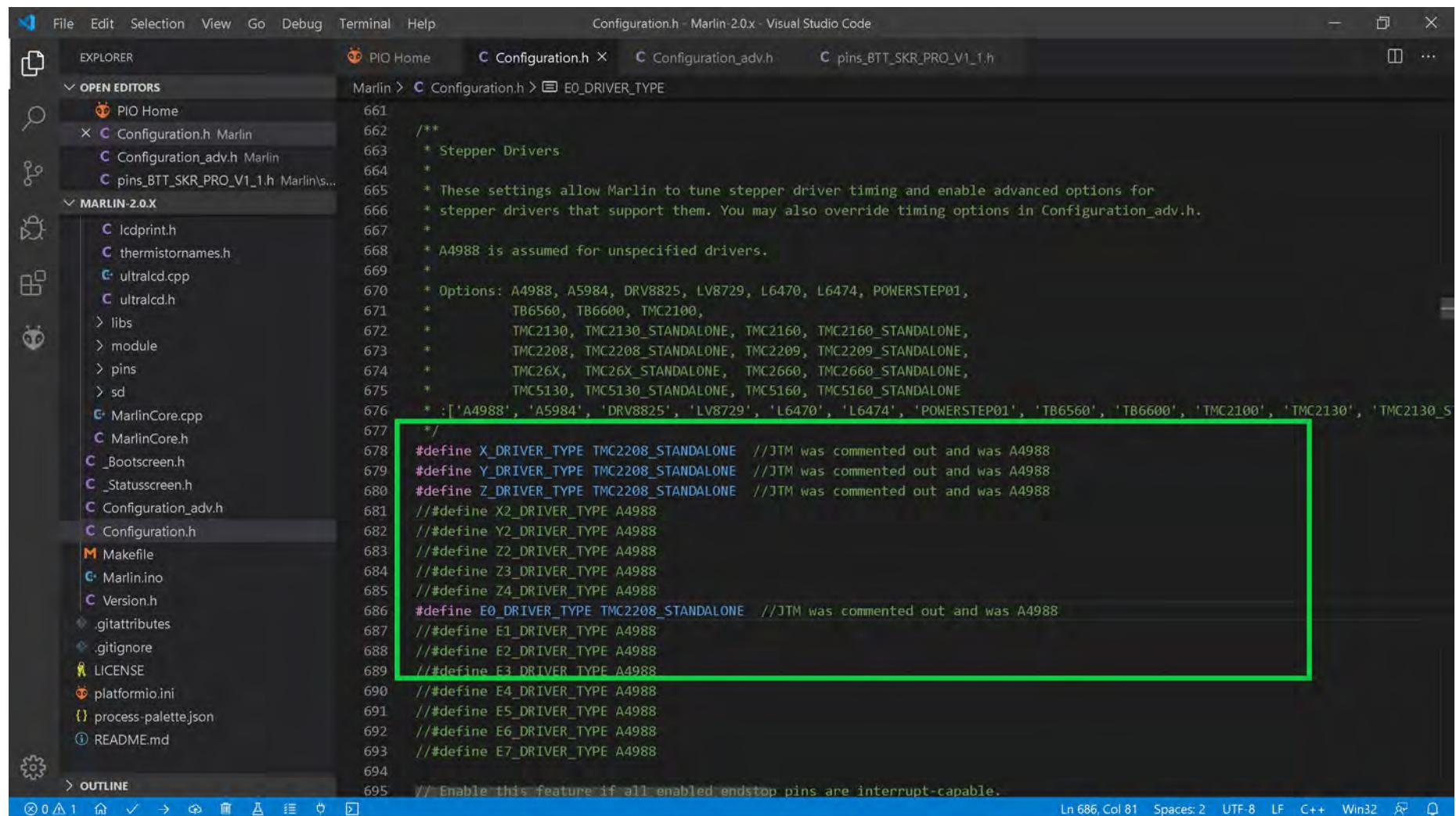
1/16
Interpolation: 1/256
SpreadCycle



The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for FYSETC TMC2208 V1.2 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):**
 - OPEN EDITORS: PIO Home, Configuration.h Marlin (selected), Configuration_adv.h, pins_BTT_SKR_PRO_V1_1.h.
 - MARLIN-2.0.X: Icdprint.h, thermistornames.h, ultralcd.cpp, ultralcd.h, libs, module, pins, sd, MarlinCore.cpp, MarlinCore.h, _Bootscreen.h, _Statusscreen.h, Configuration_adv.h, Configuration.h (selected), Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md.
- Center Area:** The code editor displays Configuration.h with the following content (lines 661-695):

```

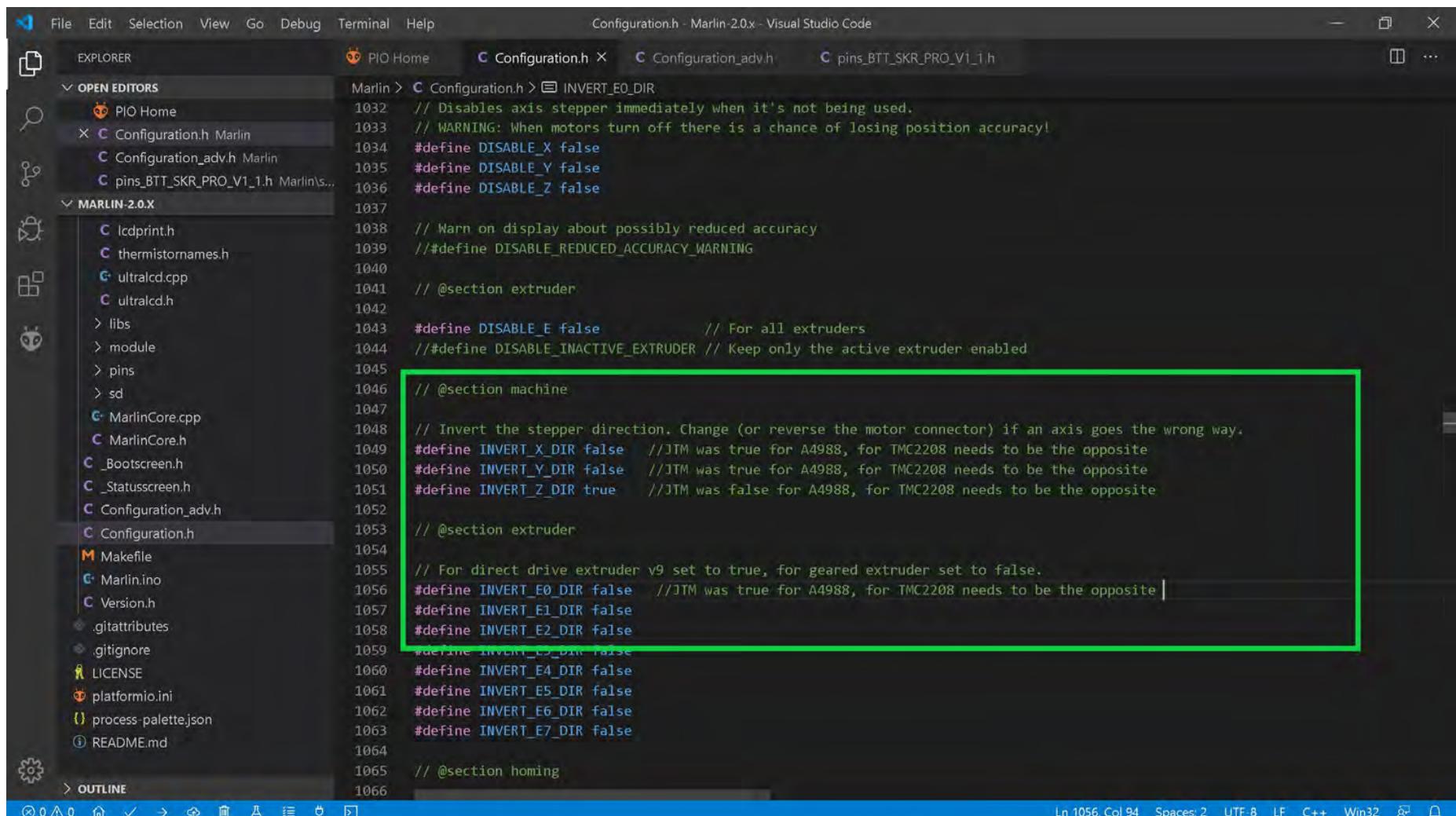
661  /**
662   * Stepper Drivers
663   *
664   * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665   * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666   *
667   * A4988 is assumed for unspecified drivers.
668   *
669   * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670   *           TB6560, TB6600, TMC2100,
671   *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672   *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673   *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674   *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675   *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208_STANDALONE']
676   */
677
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
681 // #define X2_DRIVER_TYPE A4988
682 // #define Y2_DRIVER_TYPE A4988
683 // #define Z2_DRIVER_TYPE A4988
684 // #define Z3_DRIVER_TYPE A4988
685 // #define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2208_STANDALONE //JTM was commented out and was A4988
687 // #define E1_DRIVER_TYPE A4988
688 // #define E2_DRIVER_TYPE A4988
689 // #define E3_DRIVER_TYPE A4988
690 // #define E4_DRIVER_TYPE A4988
691 // #define E5_DRIVER_TYPE A4988
692 // #define E6_DRIVER_TYPE A4988
693 // #define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```
- Bottom Status Bar:** Ln 686, Col 81, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2208 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // JTM was true for A4988, for TMC2208 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

Ln 1056, Col 94 Spaces: 2 UTF-8 LF C++ Win32

OUTLINE

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

The screenshot shows the Visual Studio Code interface with the Marlin repository open. The terminal tab displays the build log:

```

[ 100%] BIGTREE_SKR_PRO          IGNORED
[ 100%] BIGTREE_SKR_V1_0         IGNORED
BIGTREE_BTT002                  IGNORED
teensy31                         IGNORED
teensy35                         IGNORED
esp32                            IGNORED
linux_native                      IGNORED
SAMD51_grandcentral_m4           IGNORED
rumba32_f446ve                    IGNORED
mks_rumba32                       IGNORED
include_tree                       IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

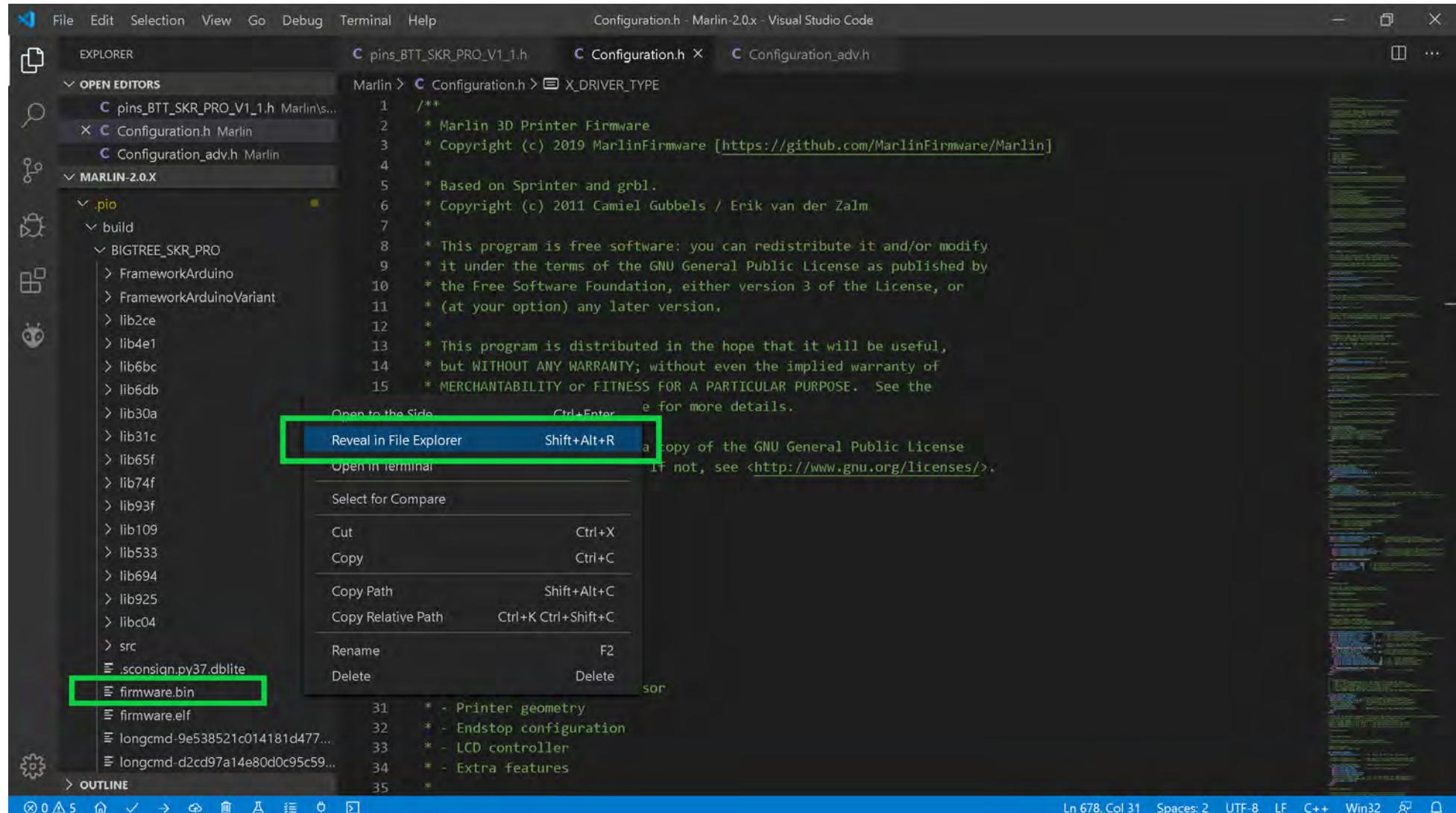
```

A yellow box highlights the checkmark icon in the bottom-left corner of the terminal window, indicating the build was successful. A green box highlights the terminal output showing the build completed successfully in 00:02:31.294.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

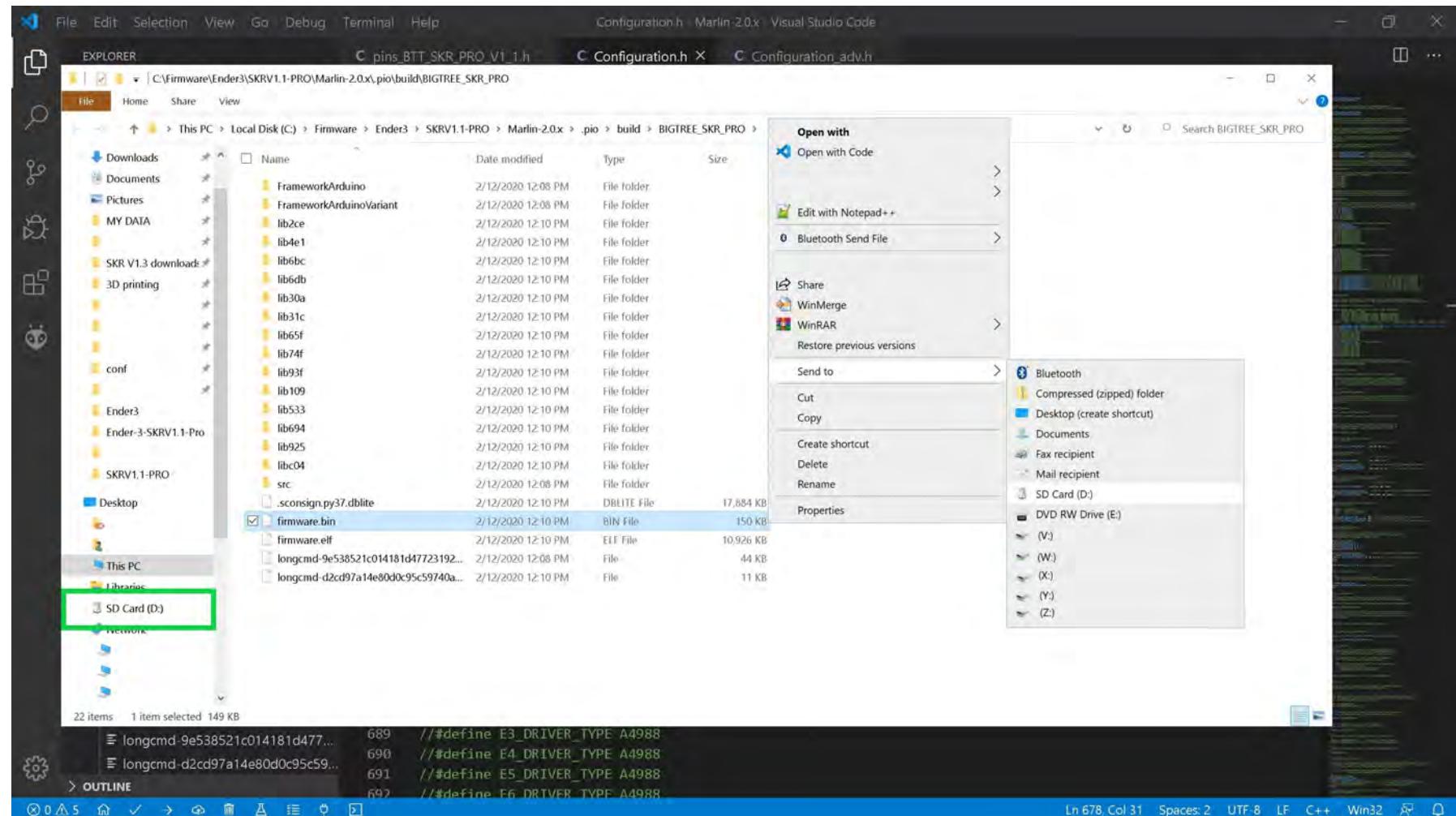
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows's machine open a file explorer window.



- Go to the next page.

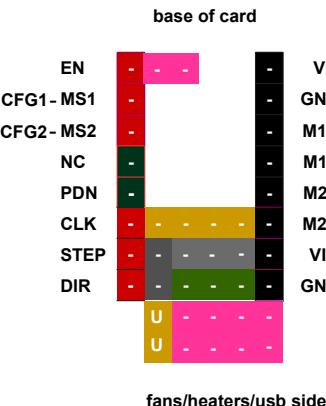
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

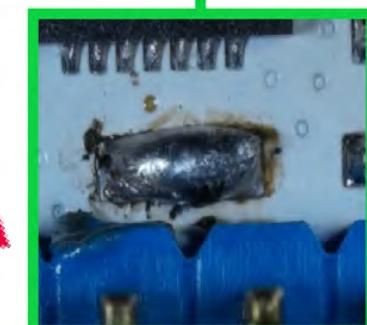
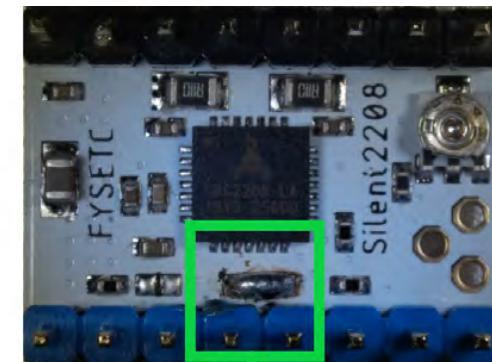
UART



MOST FYSETC TMC2208 board are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. **So please check your boards to ensure they are in the correct mode you desire!**

Driver Chip		UART Mode
FYSETC	TMC2208	UART mode
5.5v - 36v DC	2.0A	
Maximum subdivision	256	
Driving current calculation formula		
$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$		
$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$		

Important: To ensure that the FYSETC TMC2208 V1.0 or V1.2 is in UART Mode, check to see if all the adjacent pads are soldered together on the bottom of the driver board, as shown in **GREEN** box below.



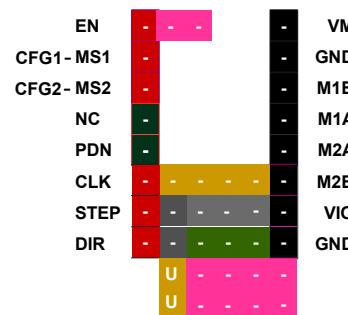
Note: You can use 50% to 90% of the calculated Irms ($I_{max}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

FYSETC TMC2208 V1.2 in UART Mode

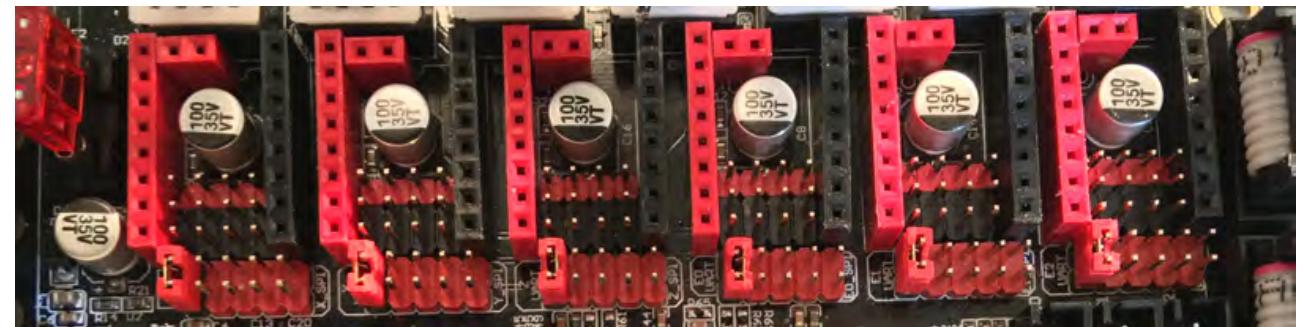
FYSETC TMC2208 V1.2

UART Mode

UART



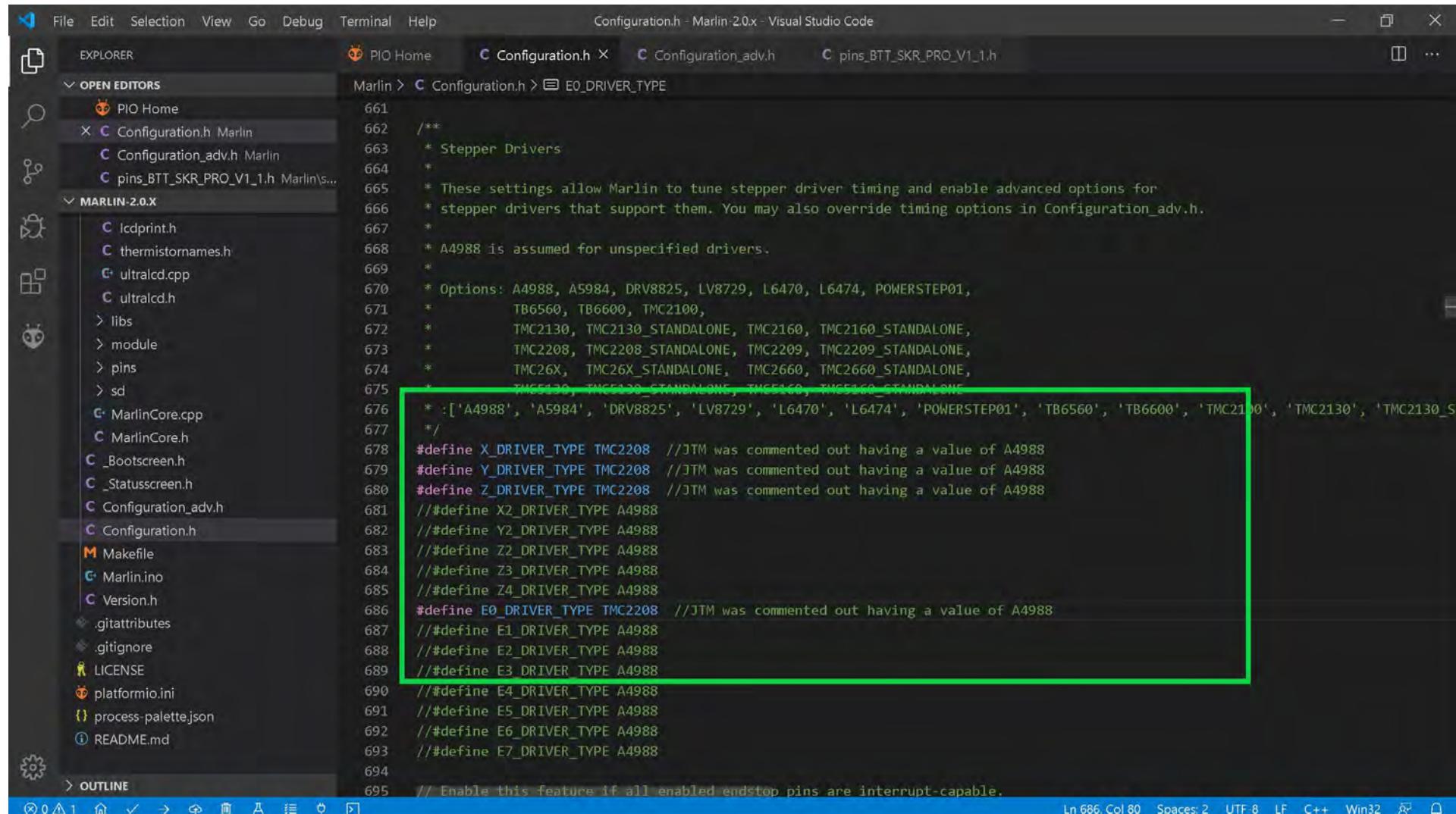
Note: Set Jumper "U" for UART MODE!!



The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for FYSETC TMC2208 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

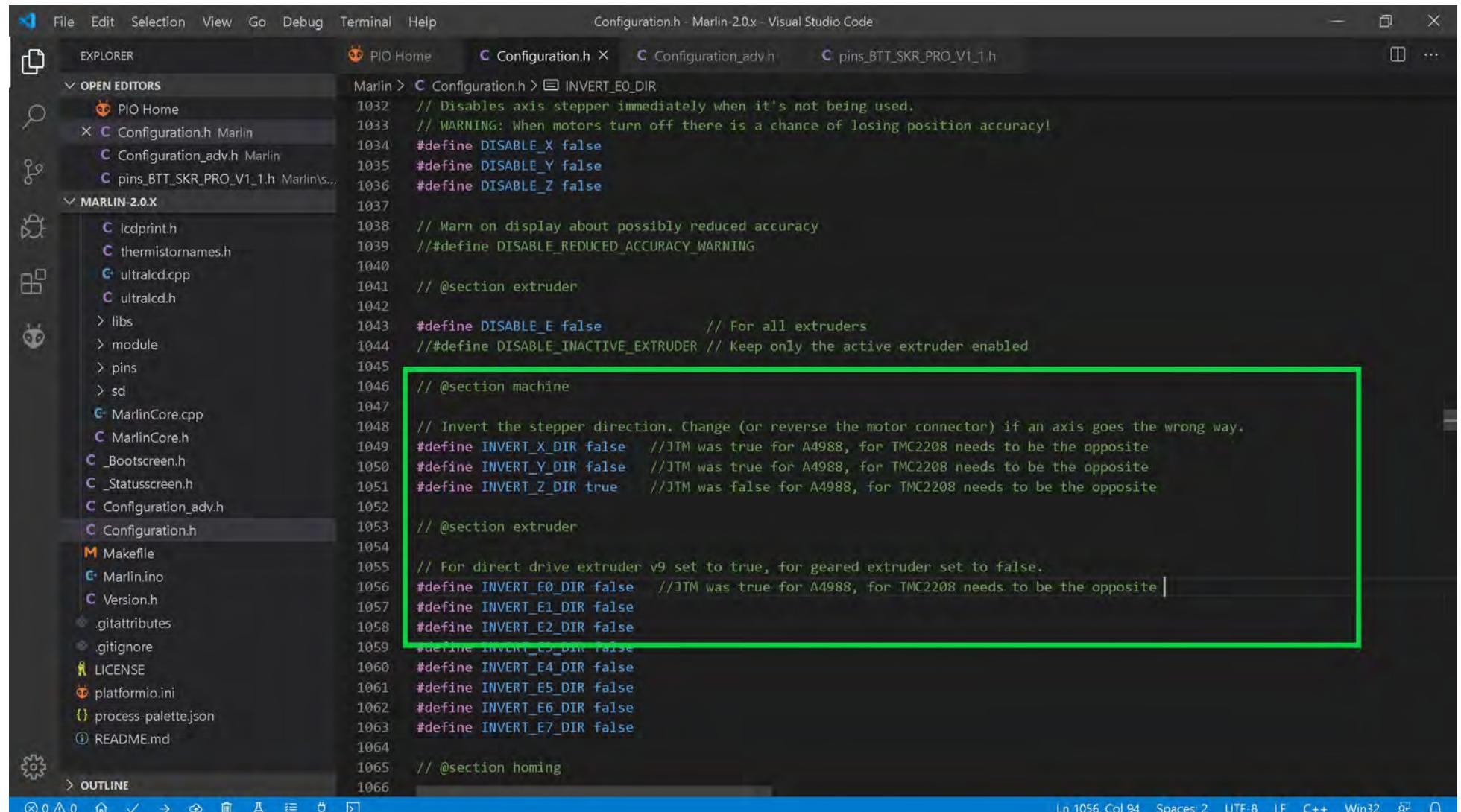
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home
Configuration.h Marlin
Configuration_adv.h Marlin
pins_BTT_SKR_PRO_V1_1.h Marlin\src\main\...
MARLIN-2.0.X
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686, Col 80 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 🔍
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5160, TMC5160_STANDALONE, TMC160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208'
677 */
#define X_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define Y_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define Z_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define Z3_DRIVER_TYPE A4988
#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208 //JTM was commented out having a value of A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

MARLIN-2.0.X

```

Marlin > C Configuration.h > INVERT_E0_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 //##define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 //##define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2208 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2208 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false    // JTM was true for A4988, for TMC2208 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

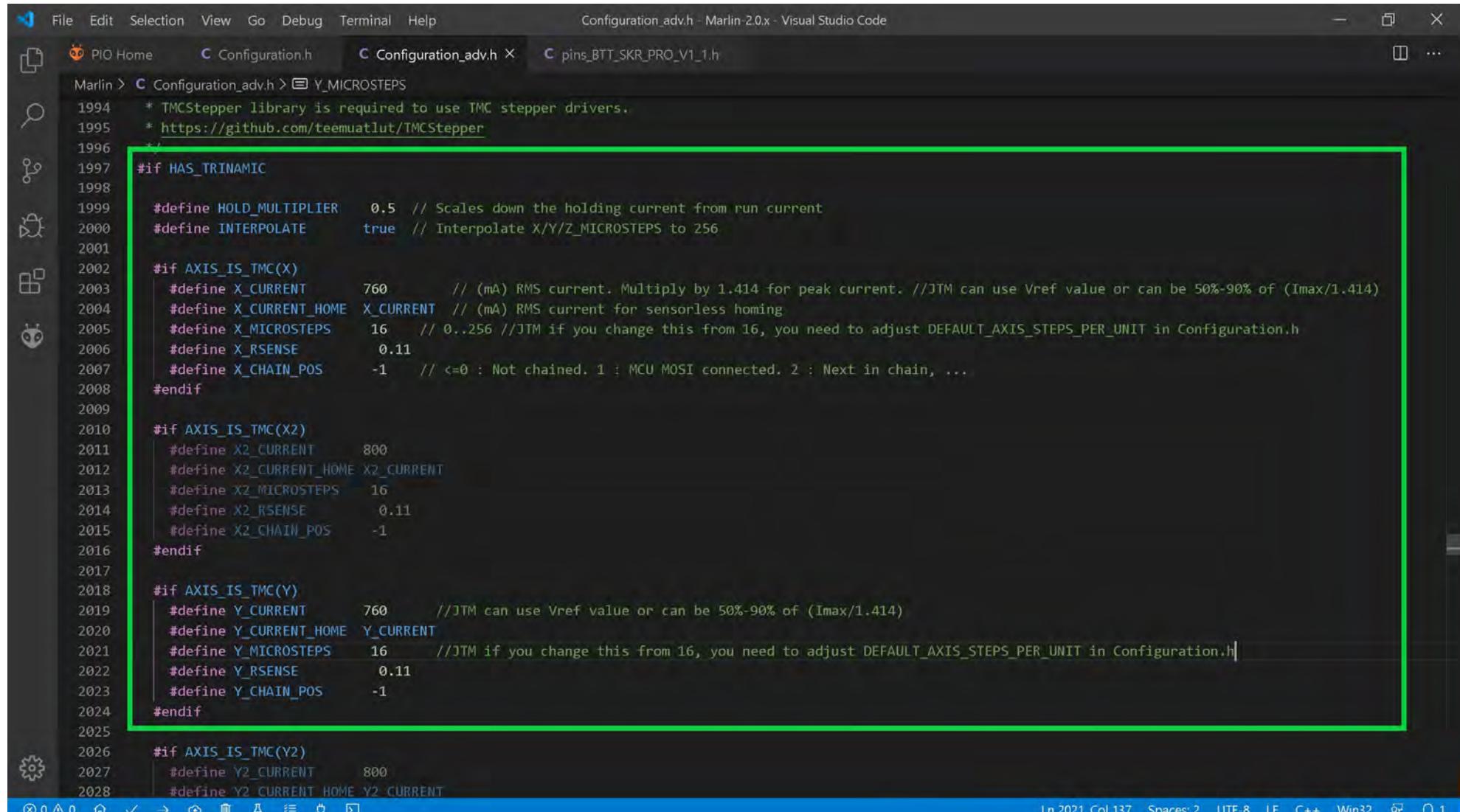
```

Ln 1056, Col 94 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- next you want to set your Vref in the Marlin firmware for each axis that has the TMC2208 driver. As seen in the GREEN box below. I changed the "X_CURRENT" to be the calculated Vref for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated Vref for my Y-Axis, which is 760mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration_adv.h > Y_MICROSTEPS
1994 * TMCSstepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCSstepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999     #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000     #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002     #if AXIS_IS_TMC(X)
2003         #define X_CURRENT          760 // (mA) RMS current. Multiply by 1.414 for peak current. //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2004         #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005         #define X_MICROSTEPS       16 // 0..256 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2006         #define X_RSENSE            0.11
2007         #define X_CHAIN_POS         -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008     #endif
2009
2010     #if AXIS_IS_TMC(X2)
2011         #define X2_CURRENT         800
2012         #define X2_CURRENT_HOME   X2_CURRENT
2013         #define X2_MICROSTEPS      16
2014         #define X2_RSENSE           0.11
2015         #define X2_CHAIN_POS        -1
2016     #endif
2017
2018     #if AXIS_IS_TMC(Y)
2019         #define Y_CURRENT          760 //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2020         #define Y_CURRENT_HOME    Y_CURRENT
2021         #define Y_MICROSTEPS       16 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2022         #define Y_RSENSE            0.11
2023         #define Y_CHAIN_POS         -1
2024     #endif
2025
2026     #if AXIS_IS_TMC(Y2)
2027         #define Y2_CURRENT         800
2028         #define Y2_CURRENT_HOME   Y2_CURRENT

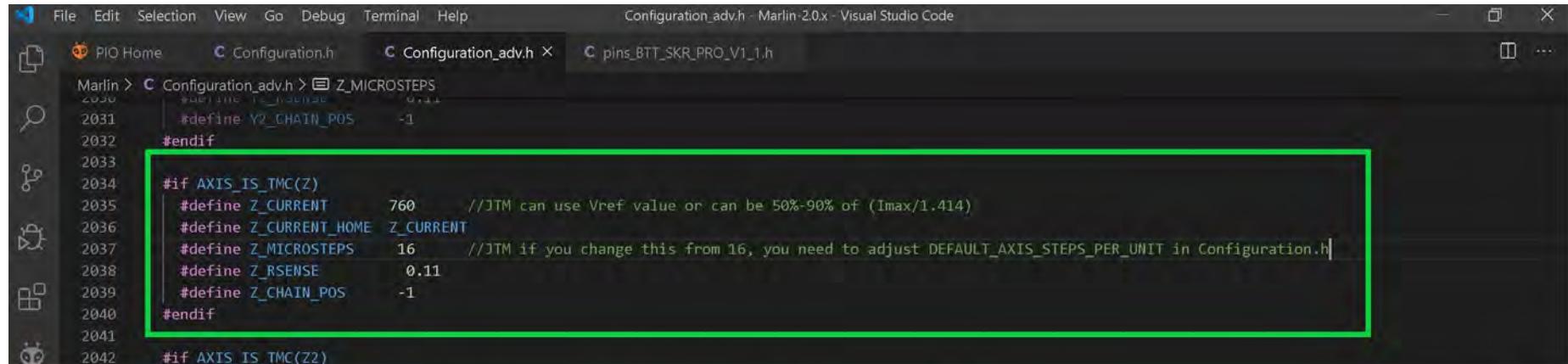
```

Ln 2021, Col 137 Spaces:2 UTF-8 LF C++ Win32  

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

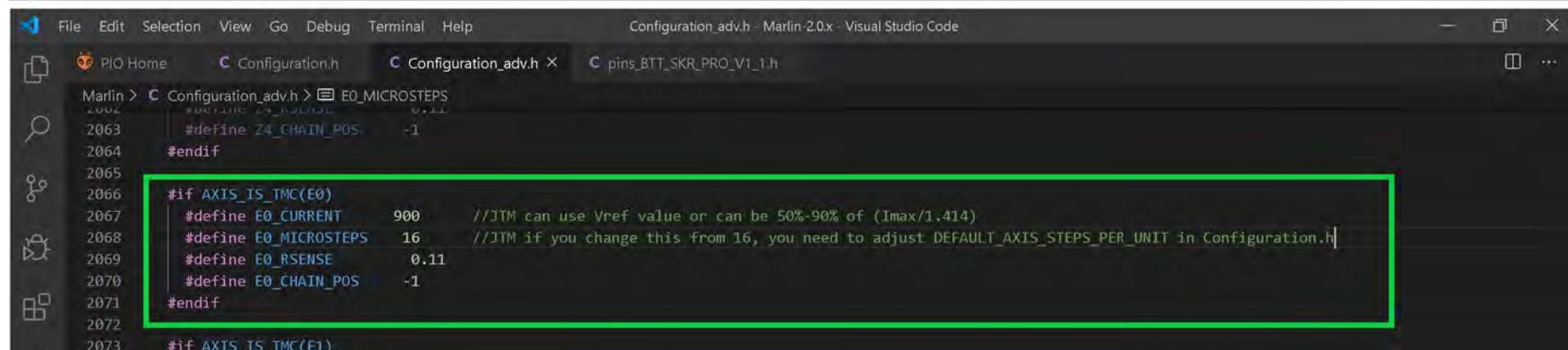
- Now, I am setting the Vref for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated Vref for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated Vref for my Extruder, which is 900mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > Z_MICROSTEPS
2030
2031     #define Z2_CHAIN_POS -1
2032
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT    760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2038         #define Z_RSENSE   0.11
2039         #define Z_CHAIN_POS -1
2040     #endif
2041
2042     #if AXIS_IS_TMC(Z2)

```

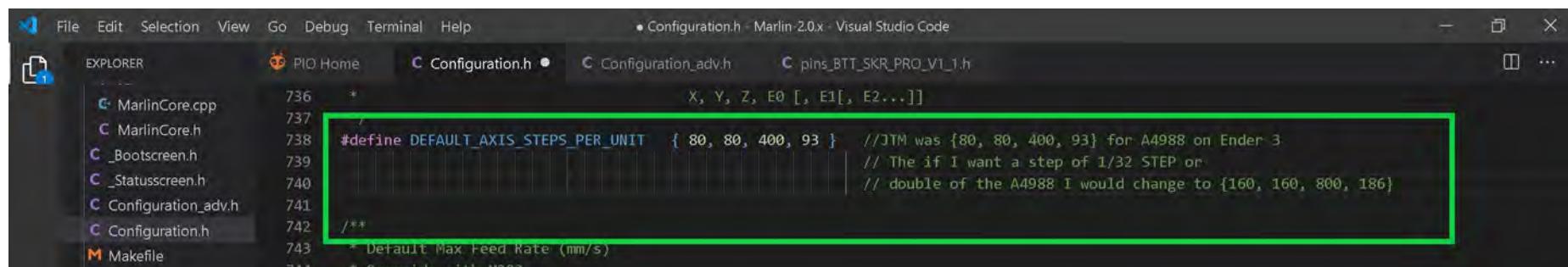


```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > E0_MICROSTEPS
2062
2063     #define Z4_CHAIN_POS -1
2064
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT    900      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2068         #define E0_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2069         #define E0_RSENSE   0.11
2070         #define E0_CHAIN_POS -1
2071     #endif
2072
2073     #if AXIS_IS_TMC(E1)

```

- If you changed the "MICROSTEPS" for any of the Axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



```

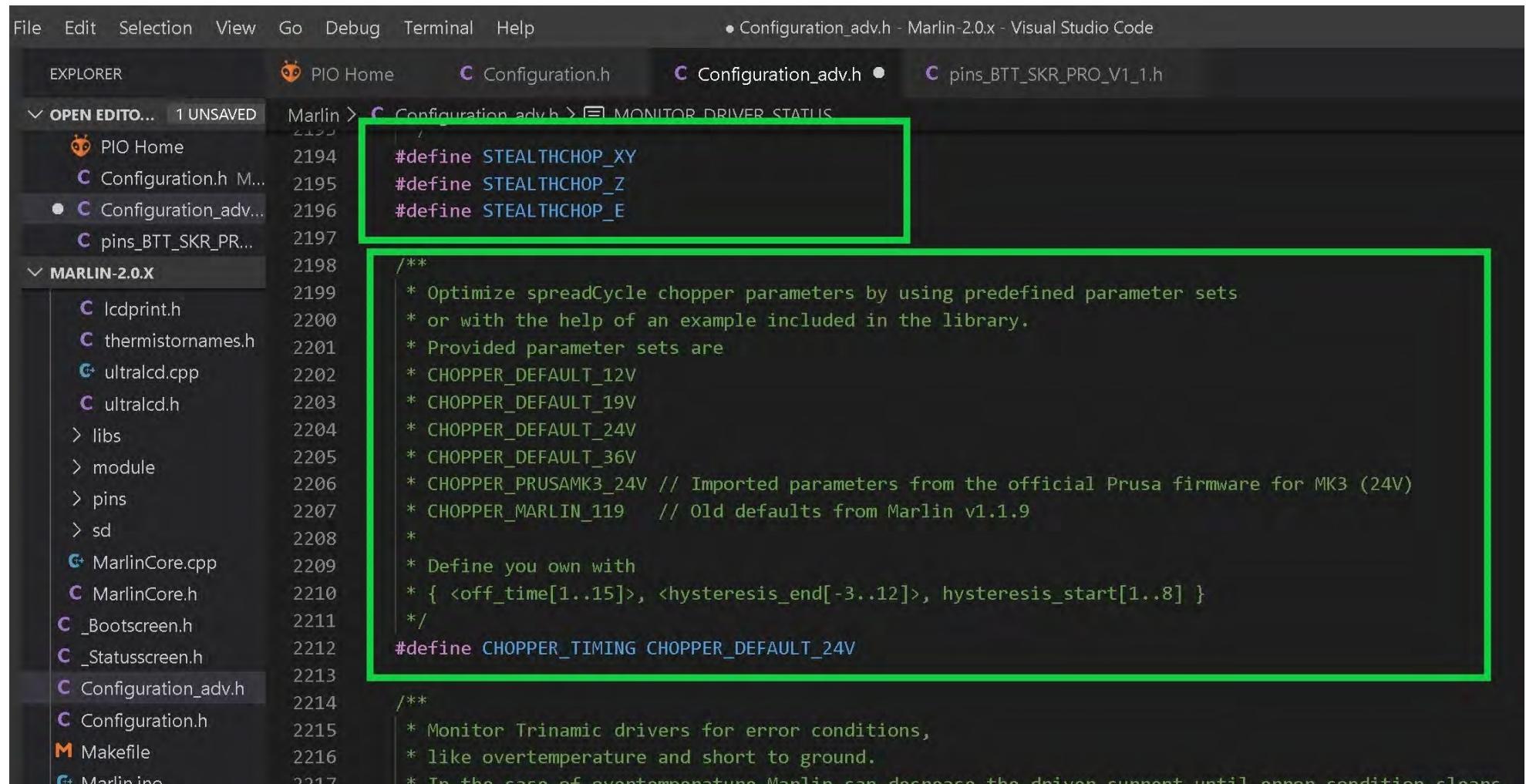
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER
MarlinCore.cpp 736     *
MarlinCore.h 737     /
Bootscreen.h 738     #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
_Statusscreen.h 739     // The if I want a step of 1/32 STEP or
Configuration_adv.h 740     // double of the A4988 I would change to {160, 160, 800, 186}
Configuration.h 741     /**
Makefile 742     */
743     * Default Max Feed Rate (mm/s)
744     * Override with M203

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if the proper "CHOPPER_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER_TIMING" is correct.

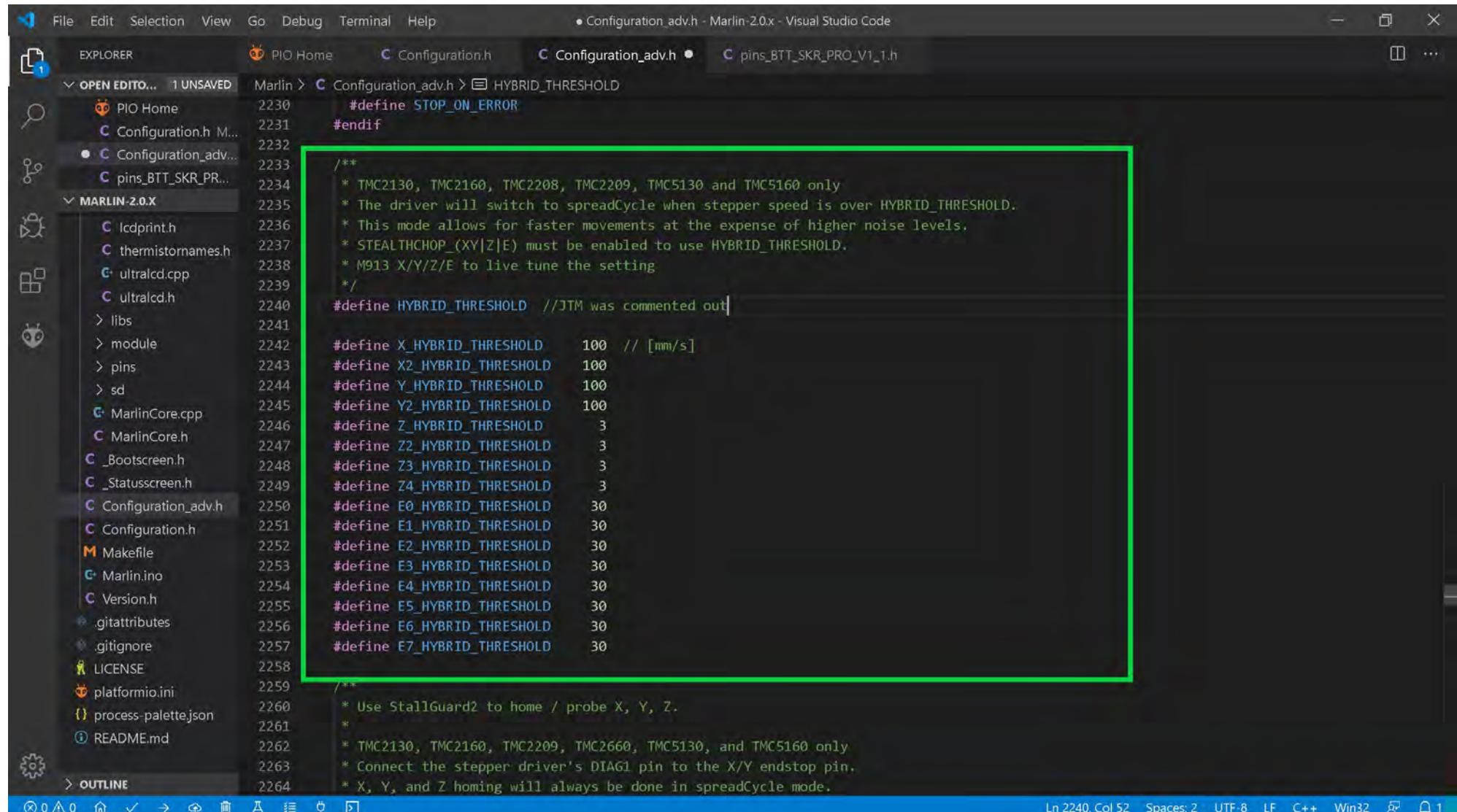


```
File Edit Selection View Go Debug Terminal Help • Configuration_adv.h - Marlin-2.0.x - Visual Studio Code  
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h  
OPEN EDITOR 1 UNSAVED Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS  
PIO Home 2194 #define STEALTHCHOP_XY  
Configuration.h M... 2195 #define STEALTHCHOP_Z  
● Configuration_adv... 2196 #define STEALTHCHOP_E  
pins_BTT_SKR_PR... 2197  
MARLIN-2.0.X 2198 /**  
 * Optimize spreadCycle chopper parameters by using predefined parameter sets  
 * or with the help of an example included in the library.  
 * Provided parameter sets are  
 * CHOPPER_DEFAULT_12V  
 * CHOPPER_DEFAULT_19V  
 * CHOPPER_DEFAULT_24V  
 * CHOPPER_DEFAULT_36V  
 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)  
 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9  
 *  
 * Define your own with  
 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }  
 */  
#define CHOPPER_TIMING CHOPPER_DEFAULT_24V  
2199 /**  
 * Monitor Trinamic drivers for error conditions,  
 * like overtemperature and short to ground.  
 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.  
2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217
```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Now you either enable "HYBRID_THRESHOLD" or disable it. By default it is disabled. "HYBRID_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays the Marlin firmware configuration. A green rectangular selection highlights the `HYBRID_THRESHOLD` section. The code within this section is as follows:

```

/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //JTM was commented out

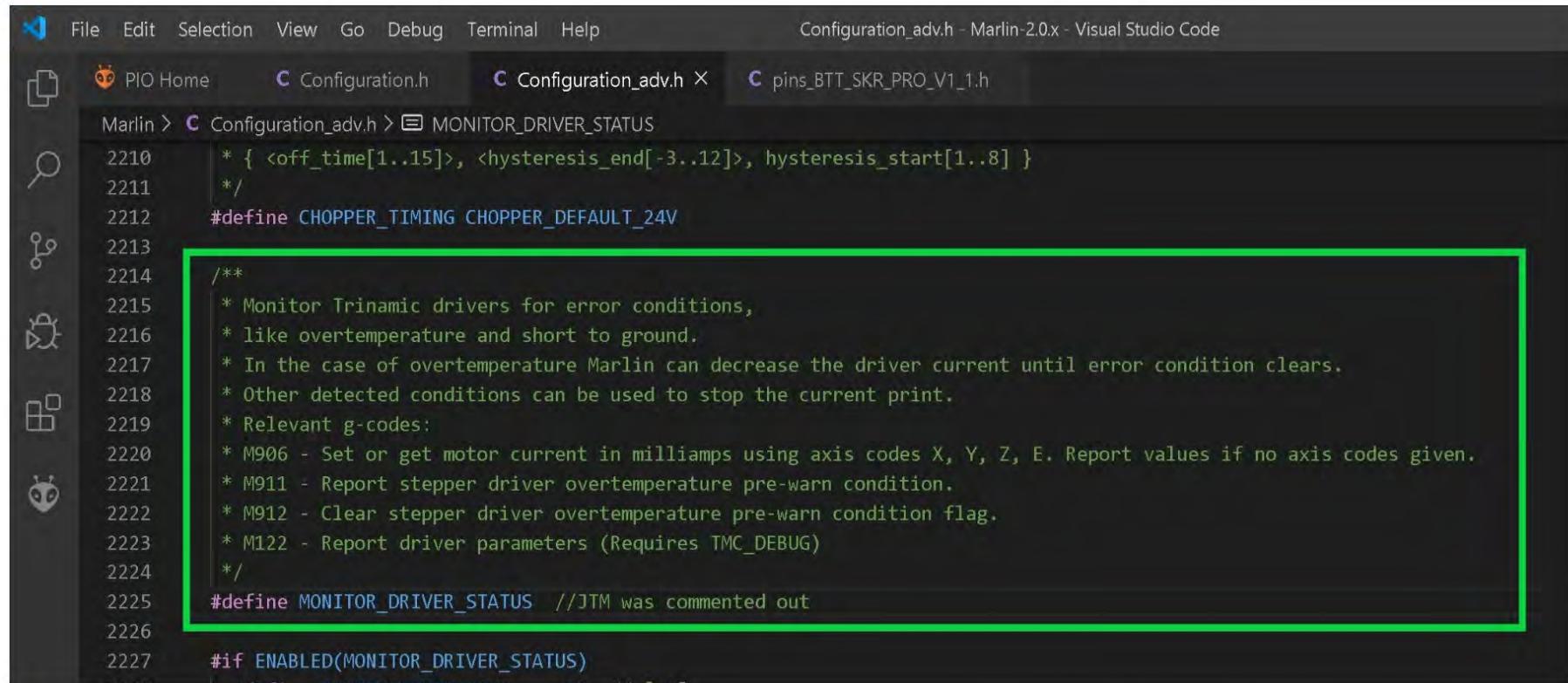
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

/*
 * Use StallGuard2 to home / probe X, Y, Z.
 *
 * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
 * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
 * X, Y, and Z homing will always be done in spreadCycle mode.
 */

```

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in Marlin's documentation located [on-line](#).
- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

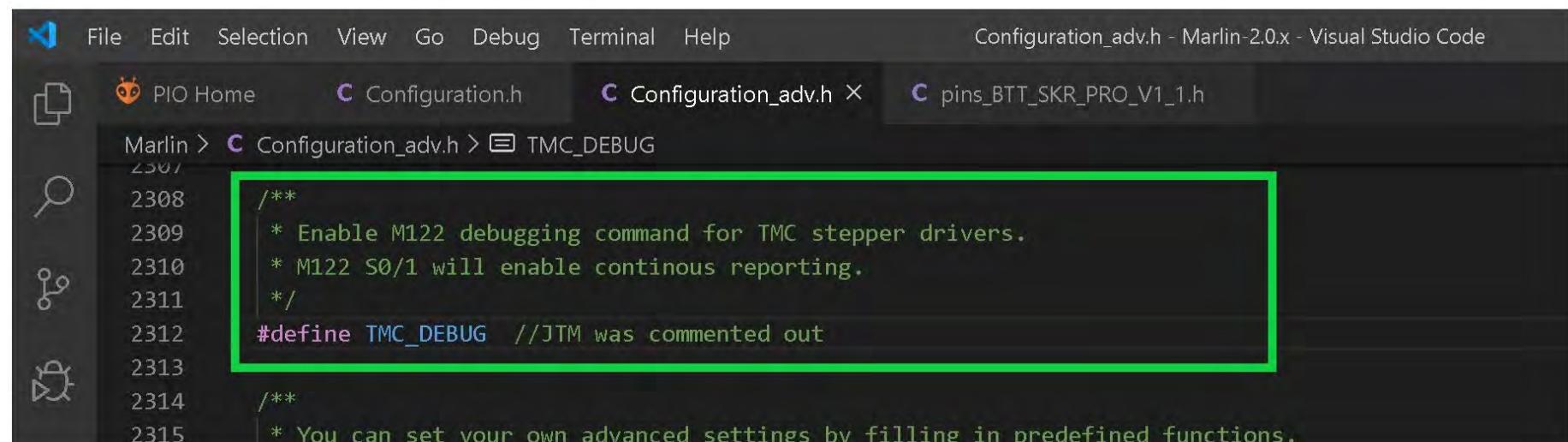
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

2210     * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211     */
2212     #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //JTM was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > TMC_DEBUG

```

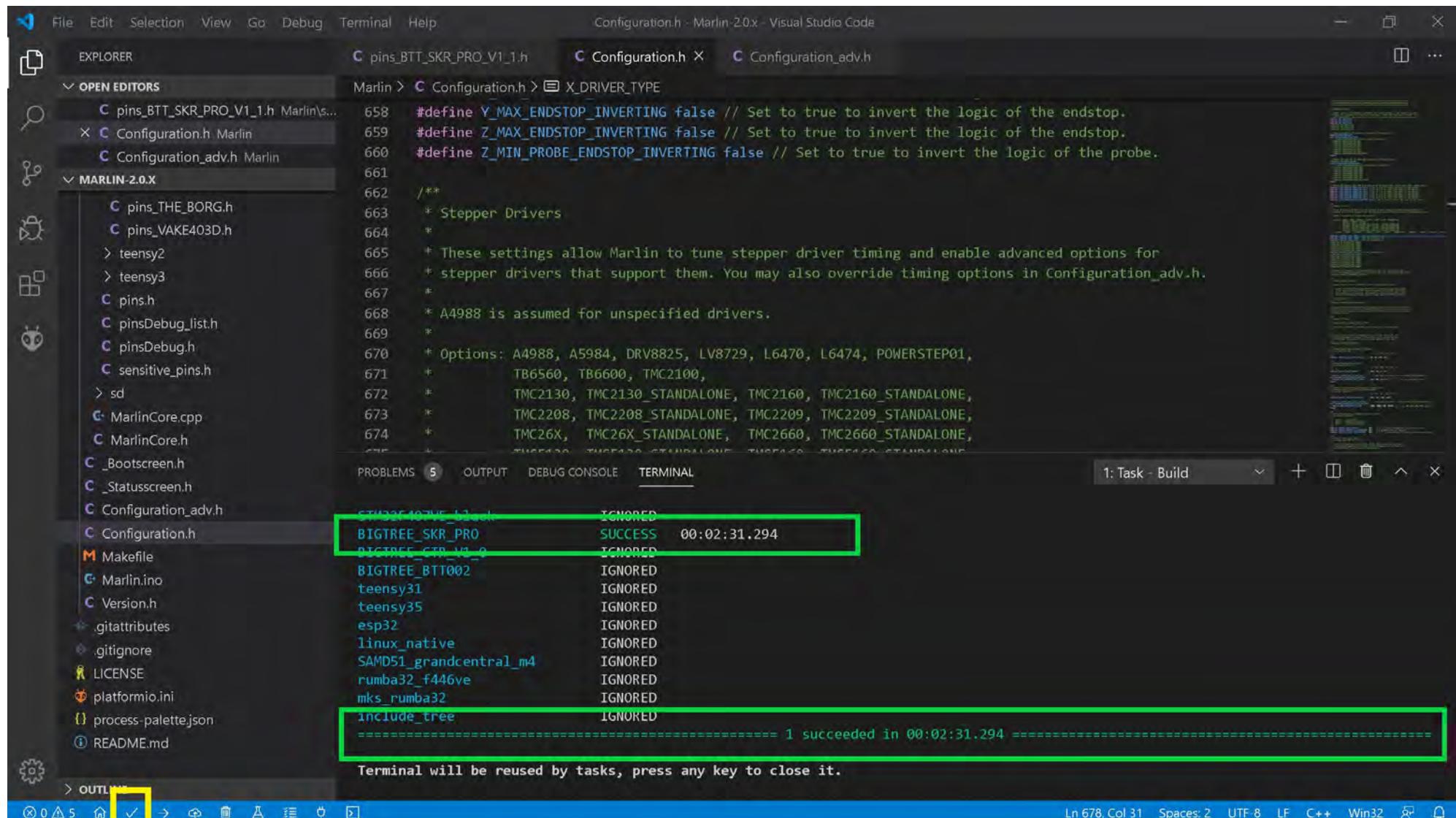
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //JTM was commented out
2313
2314 /**
2315 * You can set your own advanced settings by filling in predefined functions.

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin's...
X Configuration.h Marlin
C Configuration_adv.h Marlin
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
> teensy2
> teensy3
C pins.h
C pinsDebug_list.h
C pinsDebug.h
C sensitive_pins.h
> sd
C MarlinCore.cpp
C MarlinCore.h
C Bootscreen.h
C Statusscreen.h
C Configuration_adv.h
C Configuration.h
M Makefile
C Marlin.ino
C Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palettejson
README.md

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + - x
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_2 IGNORED
BIGTREE_BT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
===== 1 succeeded in 00:02:31.294 =====

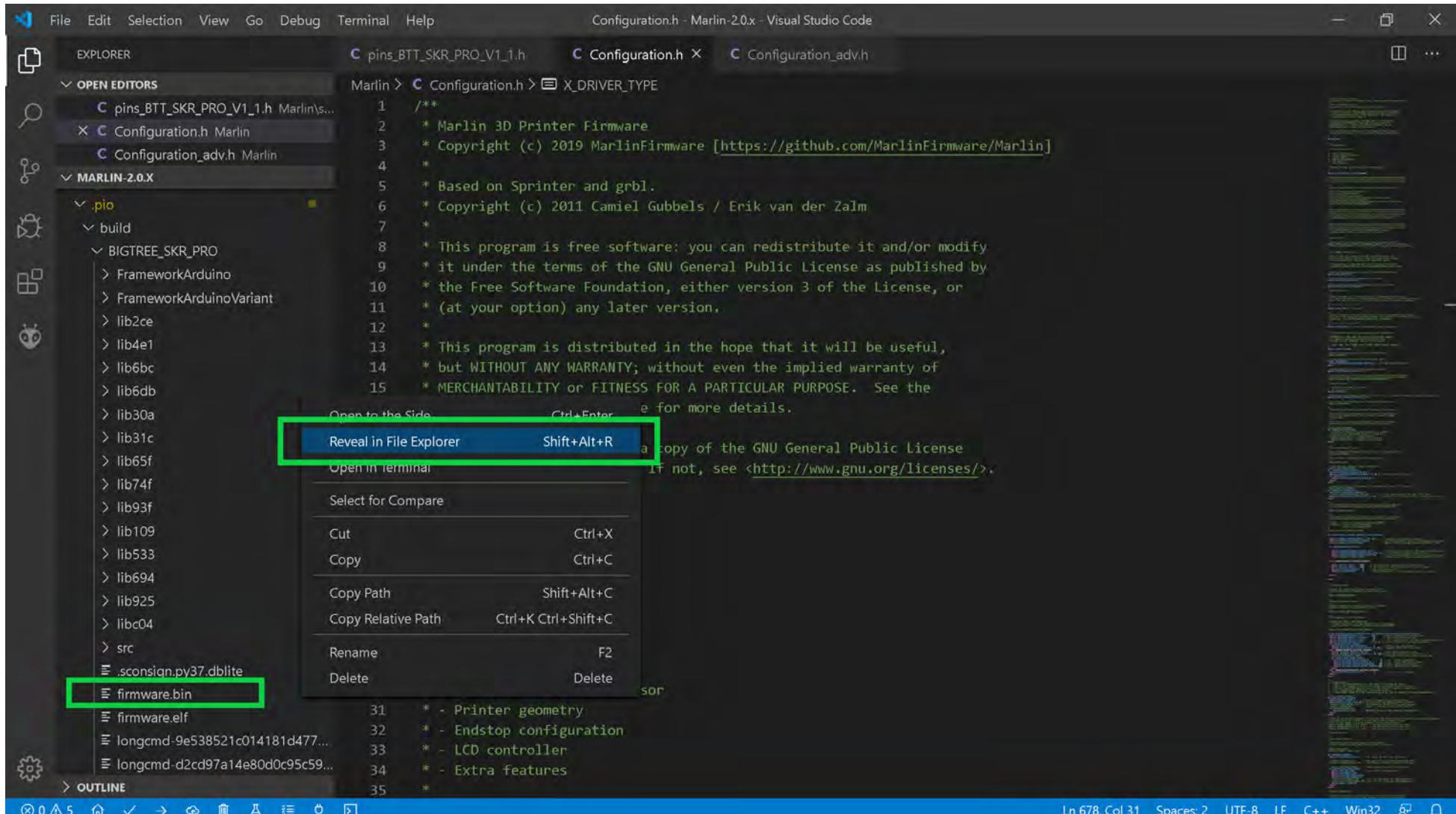
Terminal will be reused by tasks, press any key to close it.

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

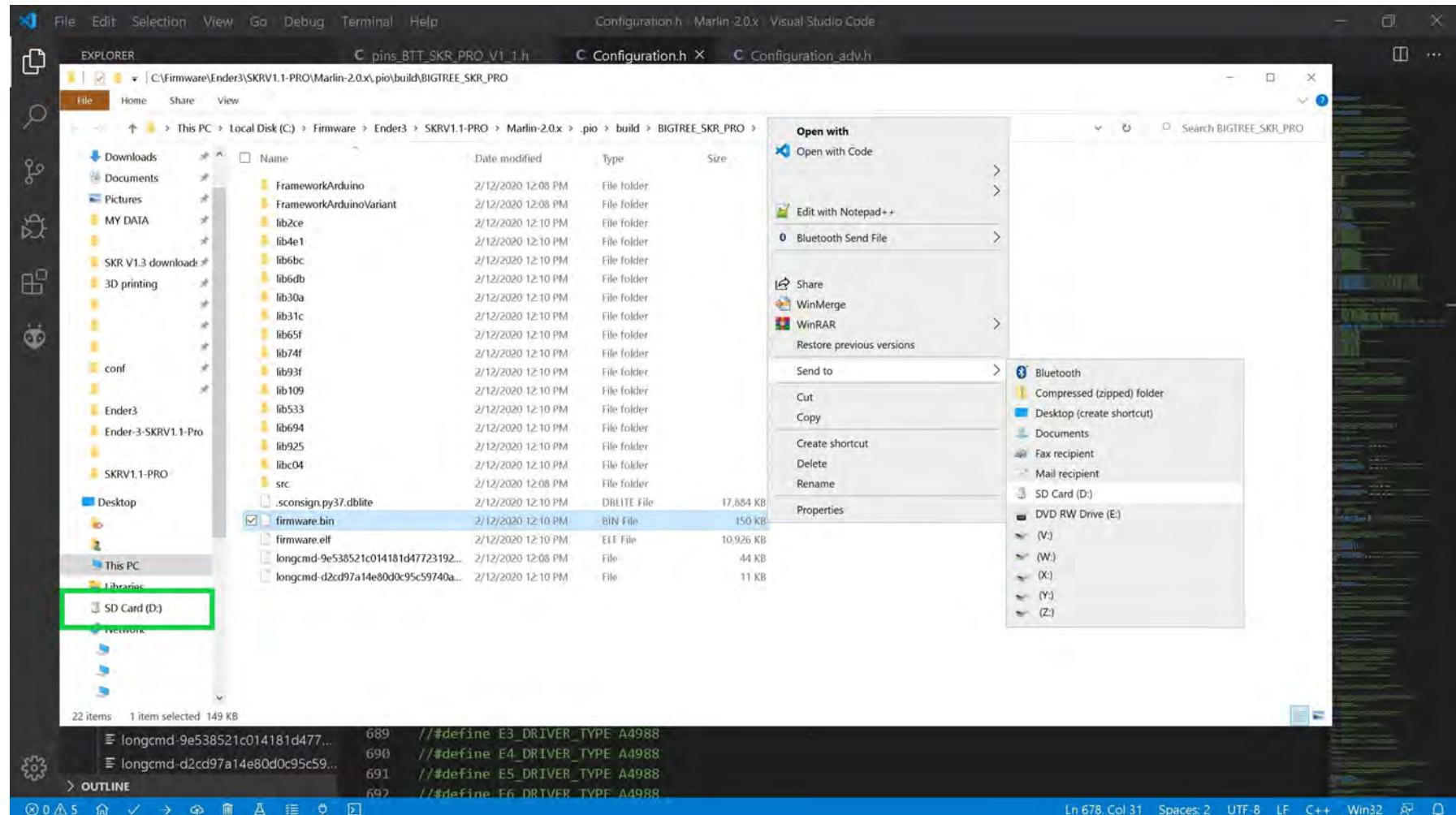
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

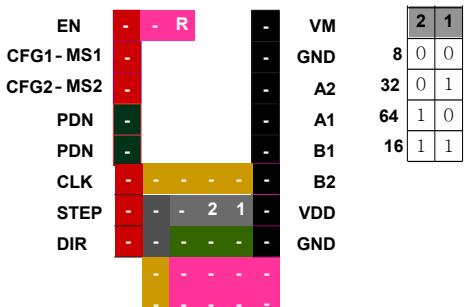
- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2209 V1.2**Stand Alone Mode for StealthChop****Stand Alone Mode - StealthChop**

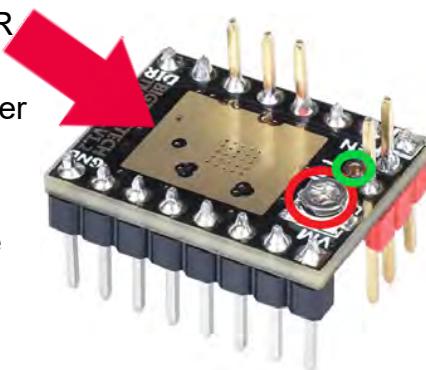
base of card



fans/heaters/usb side

MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in RED; OR use the "Vref Test point" location on the top of the driver board, as shown in BLUE . See the appendix for instructions on how to set the Vref on a driver board



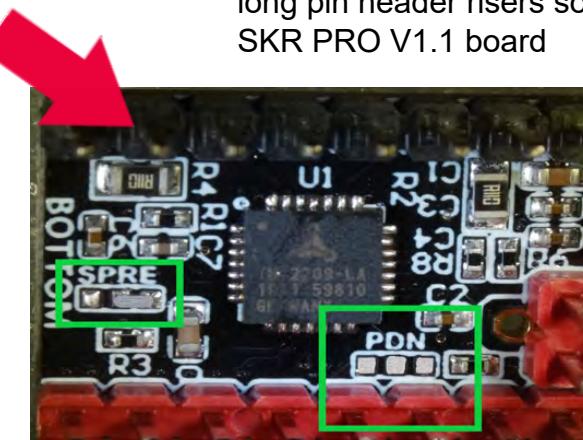
Note: Use 90% of the calculated Vref when tuning the stepper driver board

$$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$$

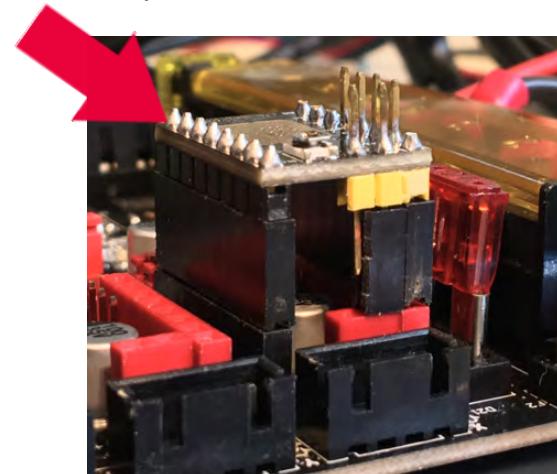
NOTE: The SPRE jumper is located on the bottom of the driver board. In Standalone Mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board.

Note: To switch to Standalone, none of the PDN pads should be bridged.

The picture below shows Standalone Mode with StealthChop.



Note: When the stall-guard function is **not used**, the stall-guard pin of the TMC2209 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board



BIQU TMC2209 V1.2

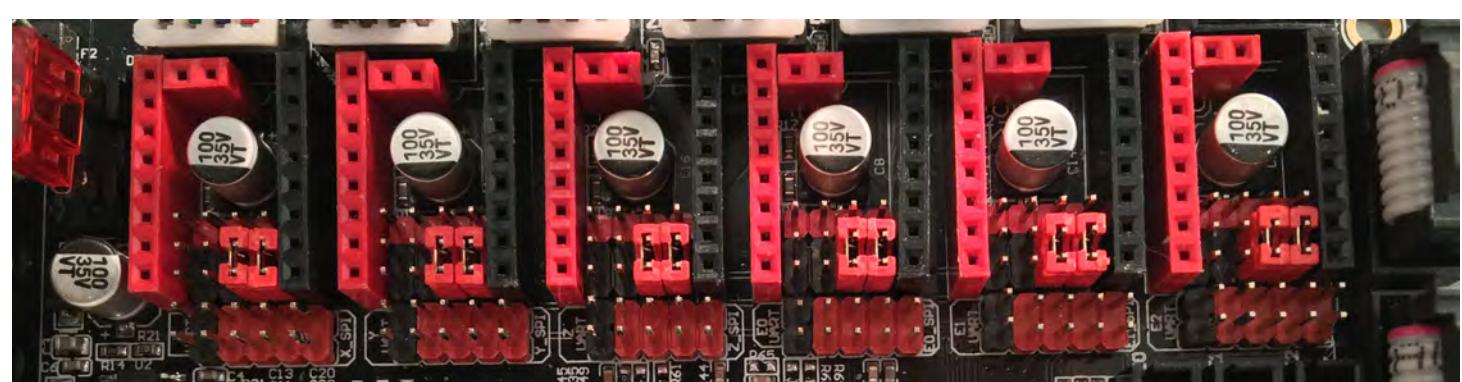
Stand Alone Mode for StealthChop

Stand Alone Mode - StealthChop

Stand Alone Mode - StealthChop

EN	-	R	-	VM
CFG1-MS1	-	-	-	GND
CFG2-MS2	-	-	-	A2
PDN	-	-	-	A1
PDN	-	-	-	B1
CLK	-	-	-	B2
STEP	-	-	2 1	VDD
DIR	-	-	2 1	GND

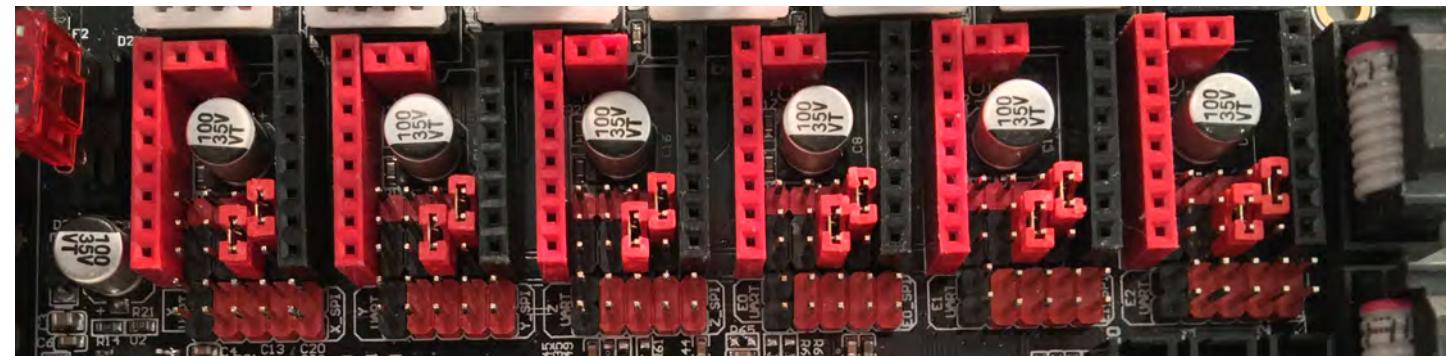
1/8
Interpolation: 1/256
StealthChop



Stand Alone Mode - StealthChop

EN	-	R	-	VM
CFG1-MS1	-	-	-	GND
CFG2-MS2	-	-	-	A2
PDN	-	-	-	A1
PDN	-	-	-	B1
CLK	-	-	1	B2
STEP	-	-	2 1	VDD
DIR	-	-	2	GND

1/32
Interpolation: 1/256
StealthChop



BIQU TMC2209 V1.2

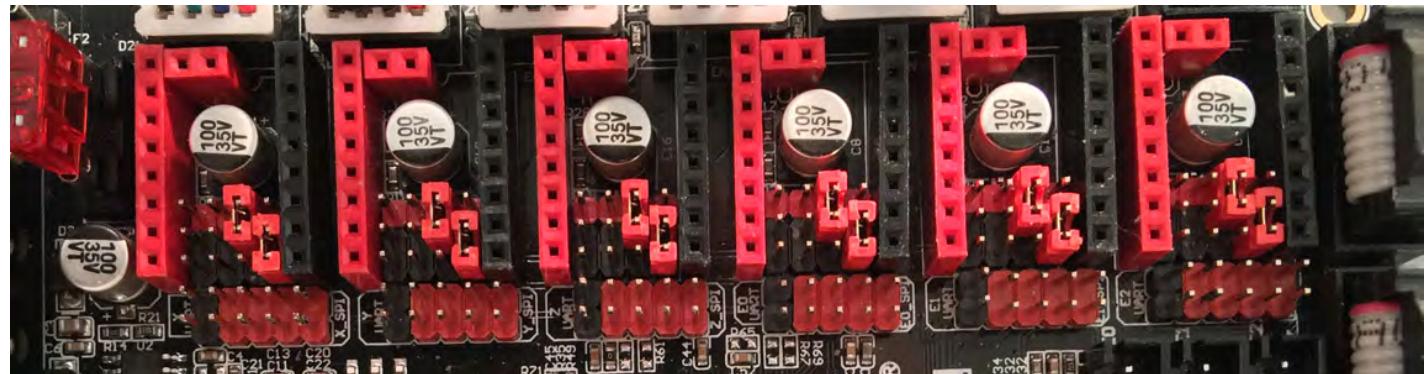
Stand Alone Mode for StealthChop

Stand Alone Mode - StealthChop

Stand Alone Mode - StealthChop

EN	-	-	R	-	VM
CFG1-MS1	-	-	-	-	GND
CFG2-MS2	-	-	-	-	A2
PDN	-	-	-	-	A1
PDN	-	-	-	-	B1
CLK	-	-	-	2	B2
STEP	-	-	-	2	VDD
DIR	-	-	-	1	GND

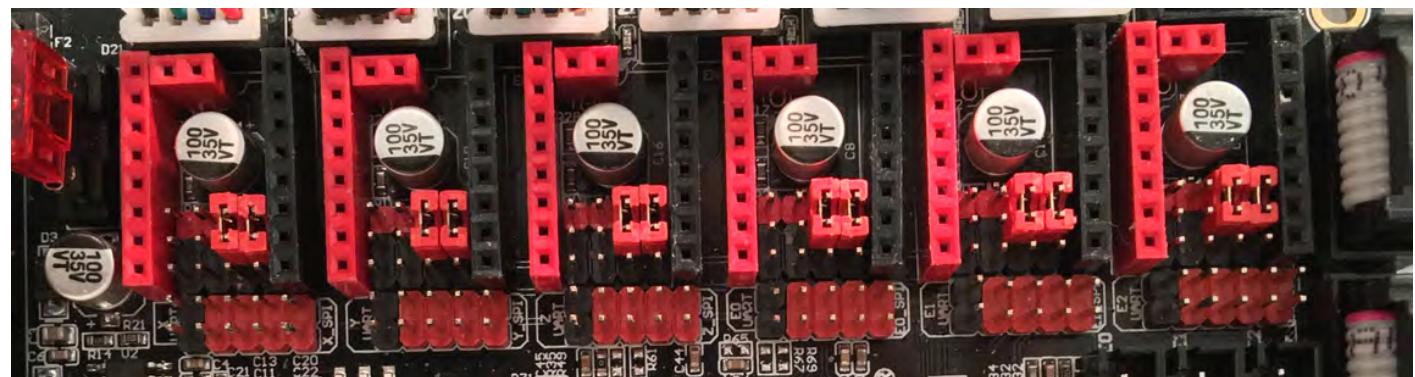
1/64
Interpolation: 1/256
StealthChop



Stand Alone Mode - StealthChop

EN	-	-	R	-	VM
CFG1-MS1	-	-	-	-	GND
CFG2-MS2	-	-	-	-	A2
PDN	-	-	-	-	A1
PDN	-	-	-	-	B1
CLK	-	-	-	2	B2
STEP	-	-	-	2	VDD
DIR	-	-	-	1	GND

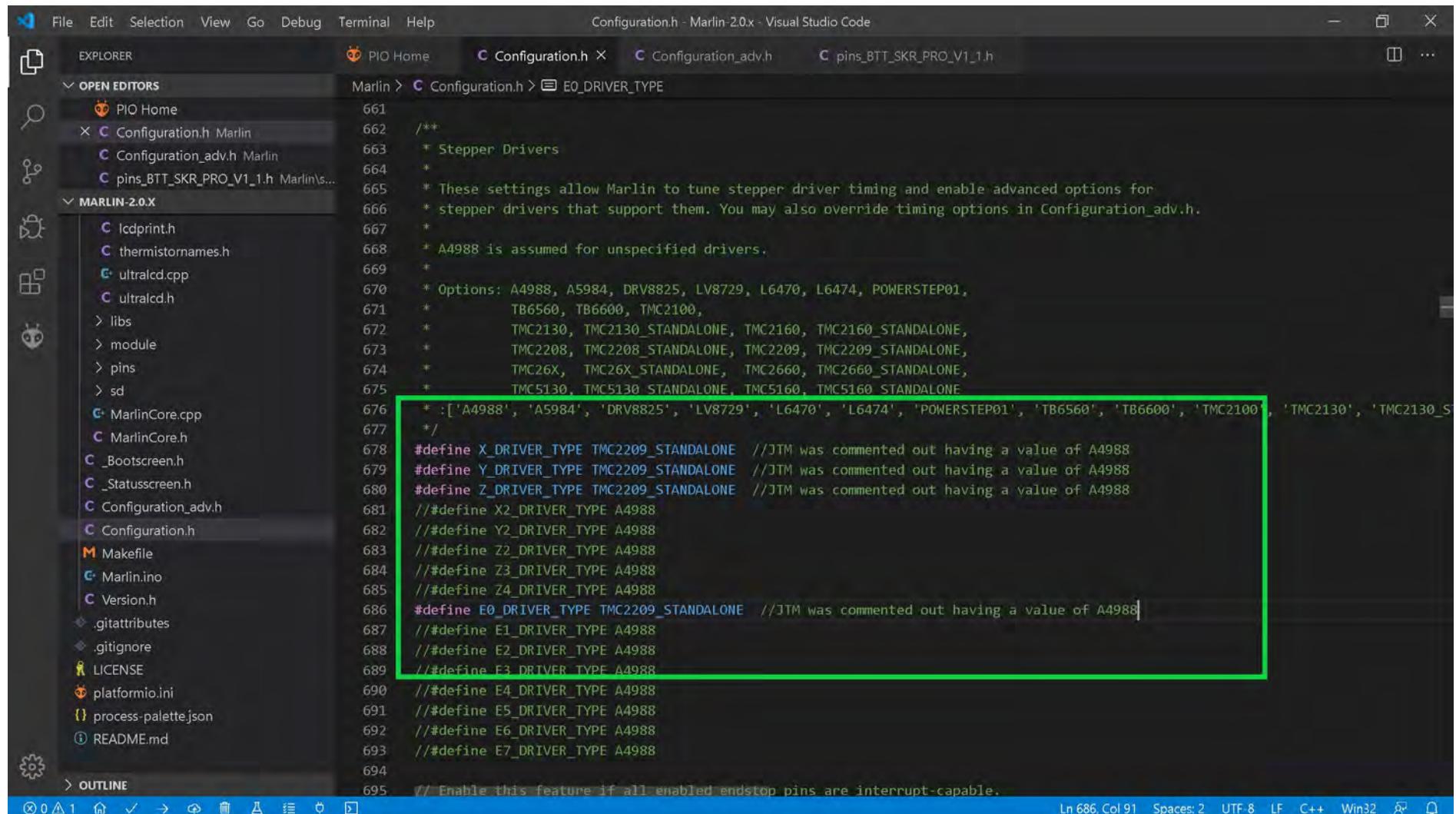
1/16
Interpolation: 1/256
StealthChop



The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2209 V1.2 stepper motor drivers in stand alone mode for stealthChop.

- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h Marlin
Marlin > Configuration_adv.h Marlin
Marlin > pins_BTT_SKR_PRO_V1_1.h Marlin...
MARLIN-2.0.X
LCDprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686, Col 91 Spaces: 2 UTF-8 LF C++ Win32 ⚡
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 */
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
677 */
#define X_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
#define Y_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
#define Z_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below

```

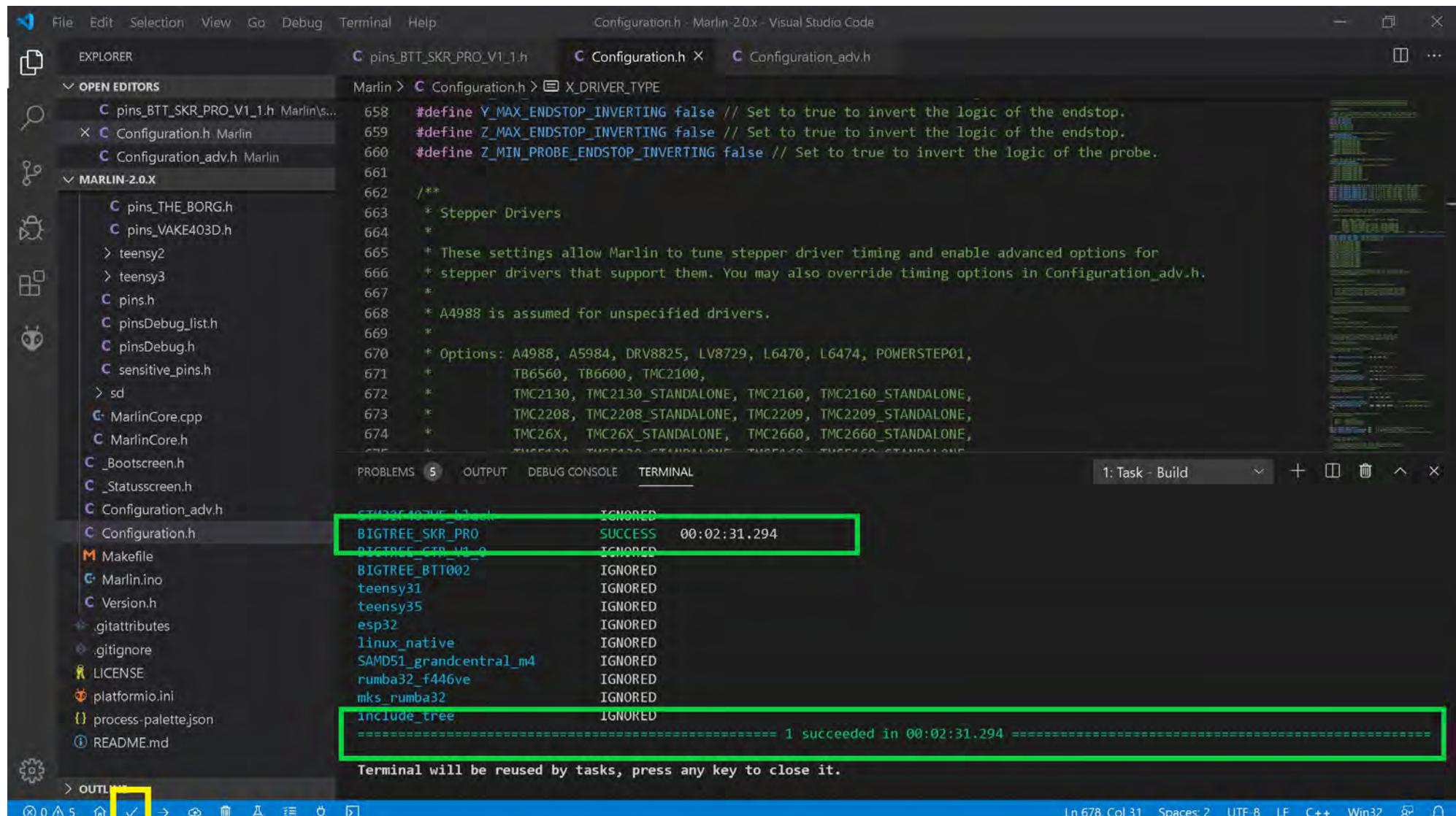
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // JTM was true for A4988, for TMC2209 needs to be the opposite
1050 #define INVERT_Y_DIR false // JTM was true for A4988, for TMC2209 needs to be the opposite
1051 #define INVERT_Z_DIR true // JTM was false for A4988, for TMC2209 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // JTM was true for A4988, for TMC2209 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop

- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand alone mode for stealthChop. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\...
Configuration.h Marlin
Configuration_adv.h Marlin
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

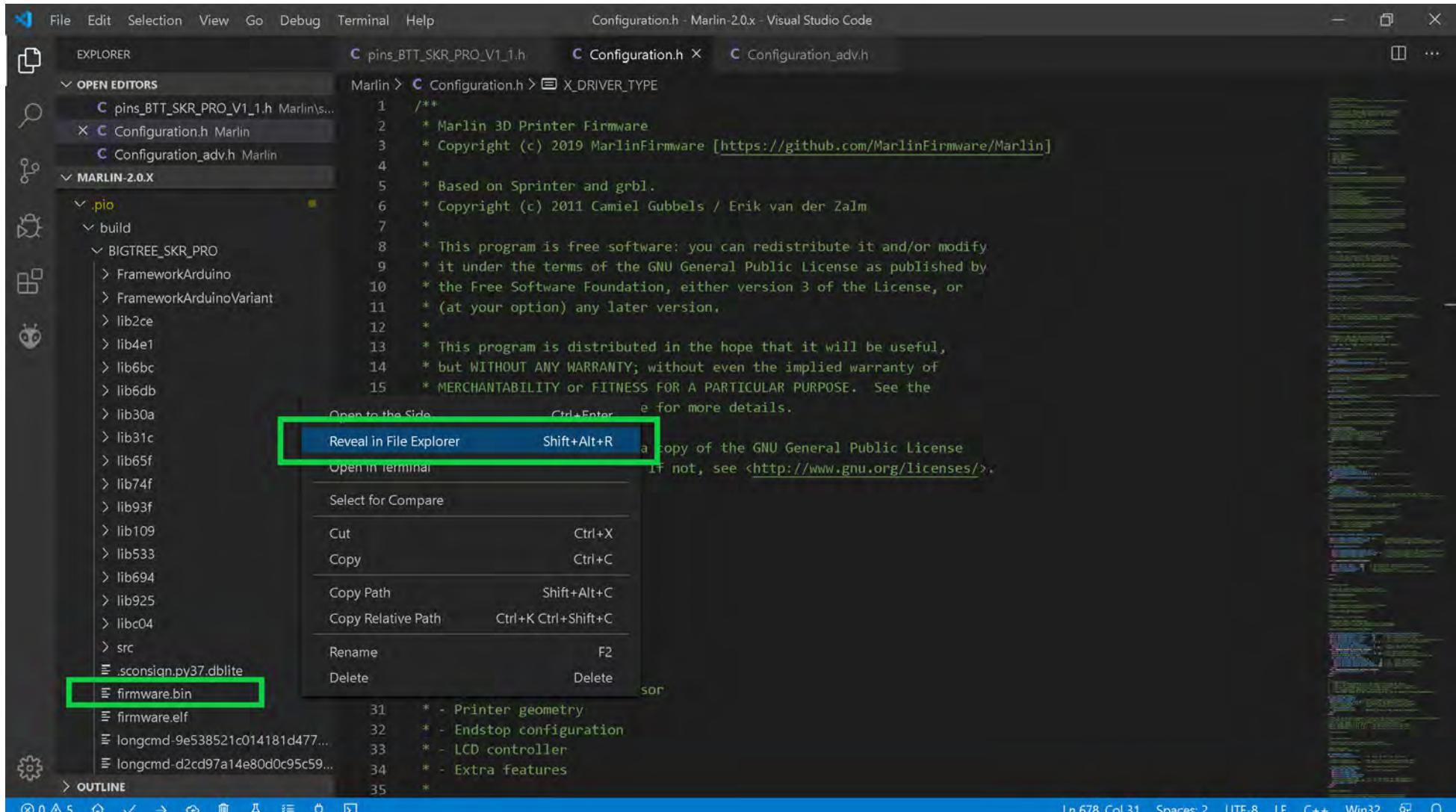
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
1 succeeded in 00:02:31.294
include_tree IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_0 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
===== 1 succeeded in 00:02:31.294 =====
Terminal will be reused by tasks, press any key to close it.

```

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop

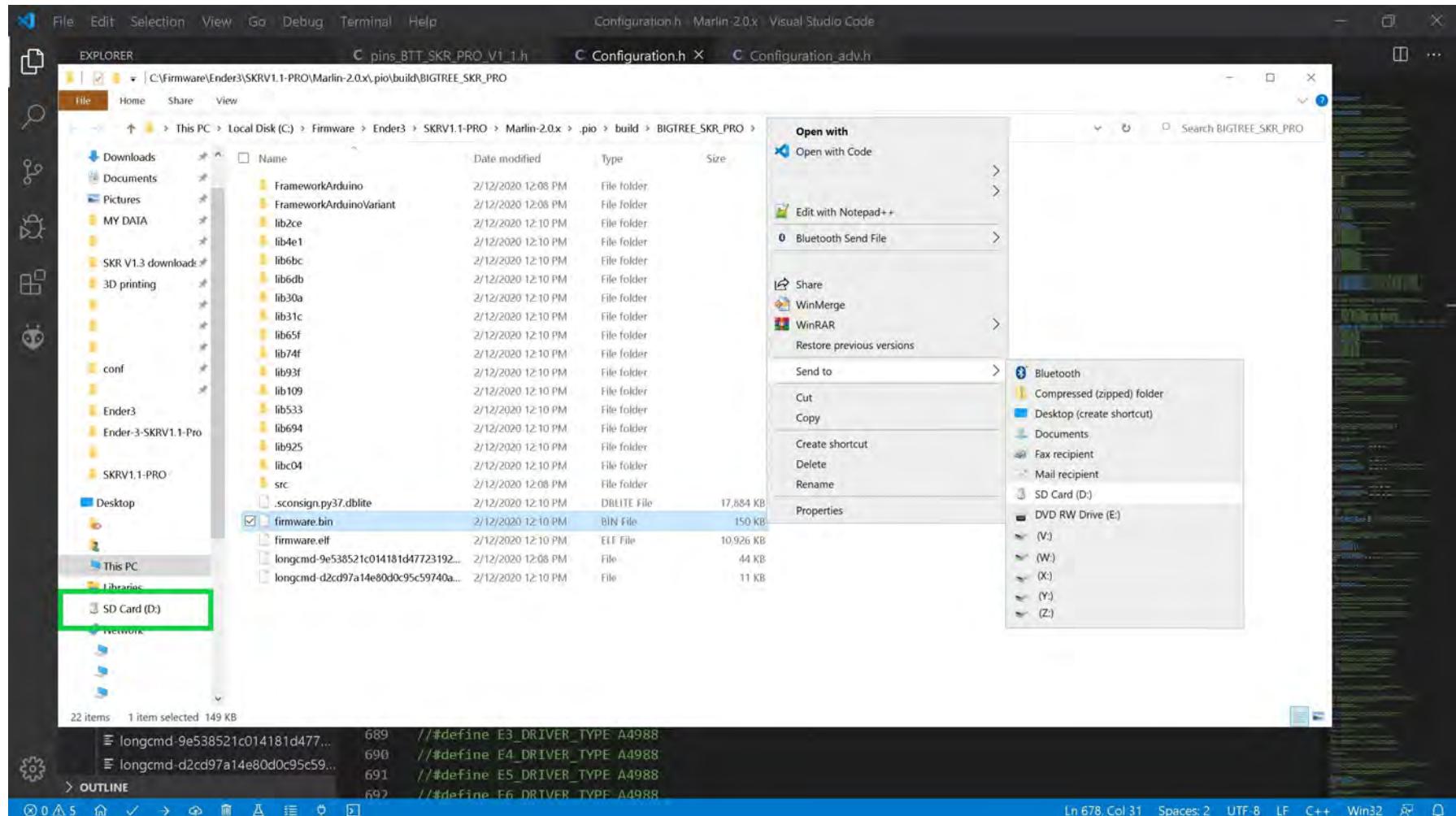
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for StealthChop

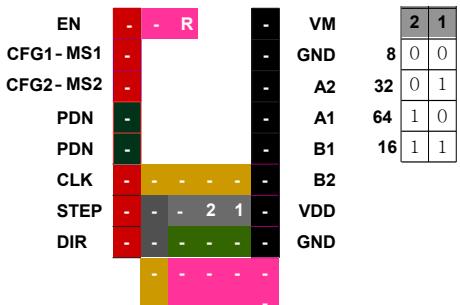
- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC2209 V1.2**Stand Alone Mode for SpreadCycle****Stand Alone Mode - SpreadCycle**

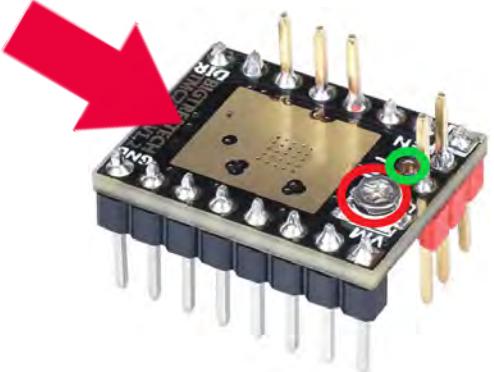
base of card



fans/heaters/usb side

MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your Vref, as shown in **RED**; OR use the "Vref Test point" location on the top of the driver board, as shown in **BLUE**. [See the appendix](#) for instructions on how to set the Vref on a driver board



Driver Chip	SPRE	CFG2/MS2	CFG1/MS1	Steps	Interpolation	Mode
TMC2209	GND	GND	GND	1/8	1/256	stealthChop
Standalone Mode	GND	GND	VIO	1/32	1/256	stealthChop
Maximum subdivision	GND	VIO	GND	1/64	1/256	stealthChop
5.5v - 28V DC	GND	VIO	VIO	1/16	1/256	stealthChop
2.8A	VIO	GND	GND	1/8	1/256	spreadCycle
	VIO	GND	VIO	1/32	1/256	spreadCycle
	VIO	VIO	GND	1/64	1/256	spreadCycle
	VIO	VIO	VIO	1/16	1/256	spreadCycle

Driving current calculation formula

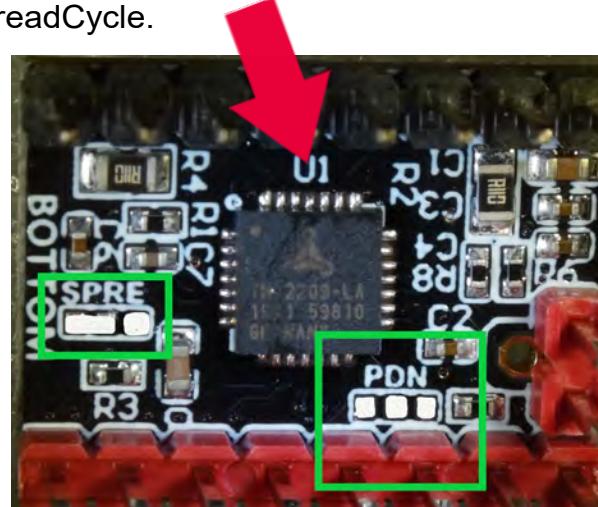
$$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$$

Note: Use 90% of the calculated Vref when tuning the stepper driver board

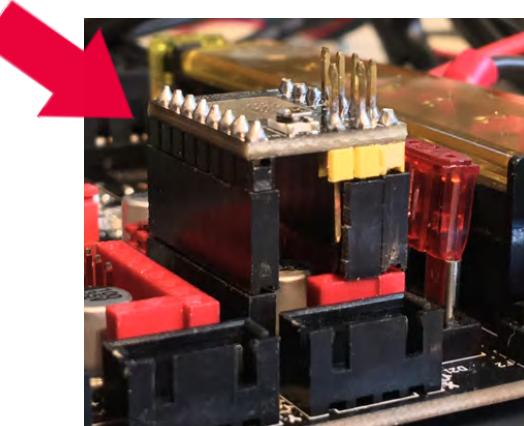
$$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$$

NOTE: The SPRE jumper is located on the bottom of the driver board. In Standalone Mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board.

Note: To switch to Standalone, none of the PDN pads should be bridged.
The picture below shows Standalone Mode with SpreadCycle.



Note: When the stall-guard function is **not used**, the stall-guard pin of the TMC2209 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board

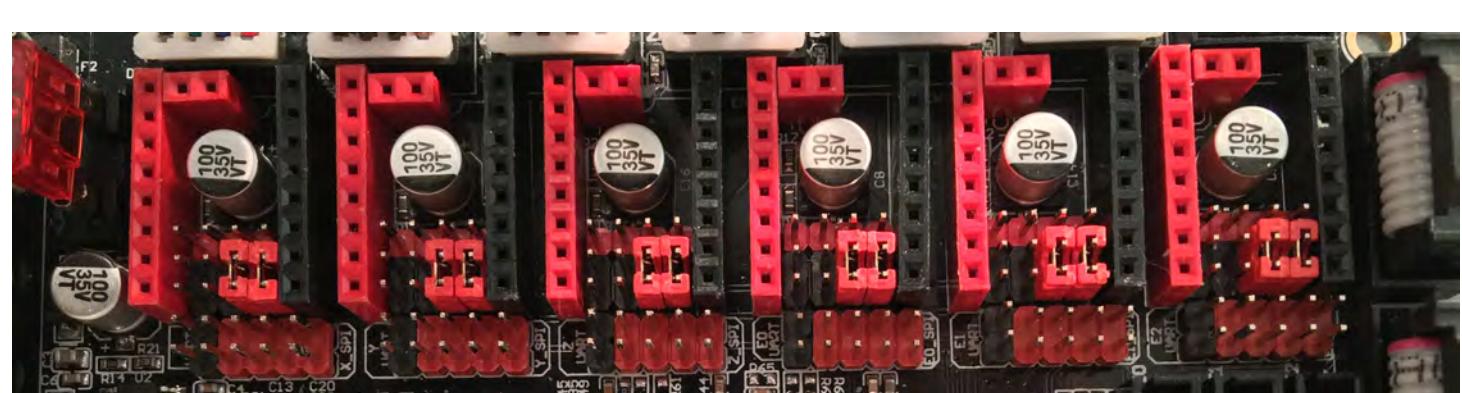


Stand Alone Mode - SpreadCycle

Stand Alone Mode - SpreadCycle

EN	-	R	-	VM
CFG1-MS1	-	-	-	GND
CFG2-MS2	-	-	-	A2
PDN	-	-	-	A1
PDN	-	-	-	B1
CLK	-	-	-	B2
STEP	-	-	2 1	VDD
DIR	-	-	2 1	GND

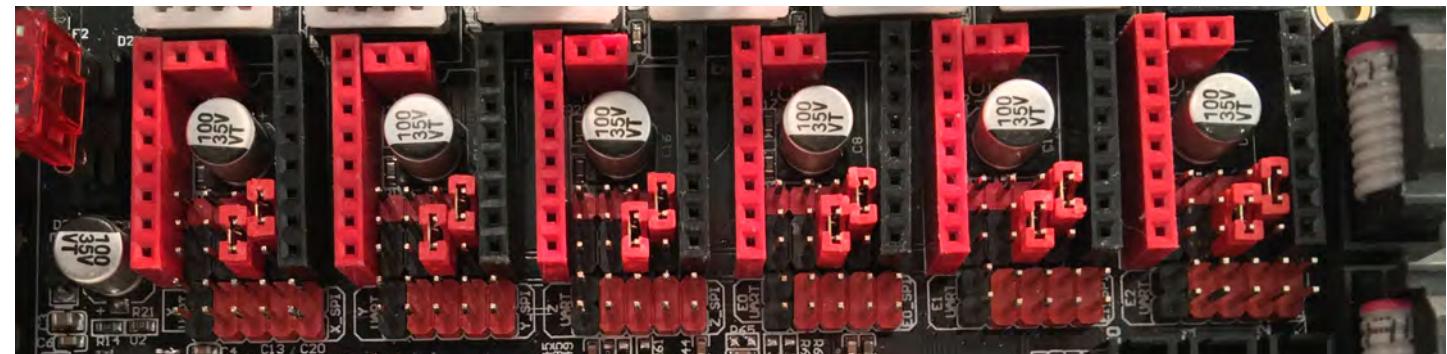
1/8
Interpolation: 1/256
SpreadCycle



Stand Alone Mode - SpreadCycle

EN	-	R	-	VM
CFG1-MS1	-	-	-	GND
CFG2-MS2	-	-	-	A2
PDN	-	-	-	A1
PDN	-	-	-	B1
CLK	-	-	1	B2
STEP	-	-	2 1	VDD
DIR	-	-	2	GND

1/32
Interpolation: 1/256
SpreadCycle



BIQU TMC2209 V1.2

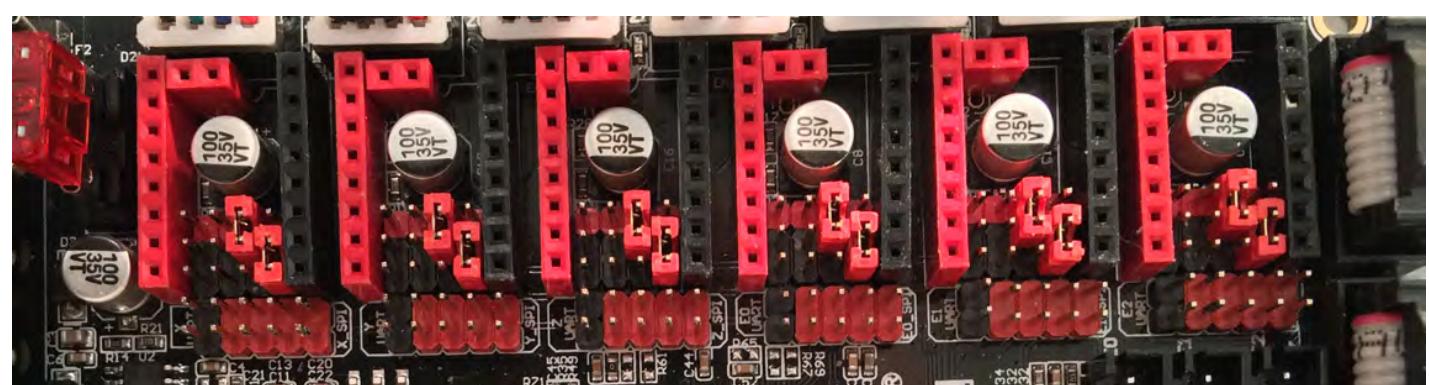
Stand Alone Mode for SpreadCycle

Stand Alone Mode - SpreadCycle

Stand Alone Mode - SpreadCycle

EN	-	R	-	VM
CFG1- MS1	-		-	GND
CFG2- MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	2	-	B2
STEP	-	2 1	-	VDD
DIR	-	-	1	GND

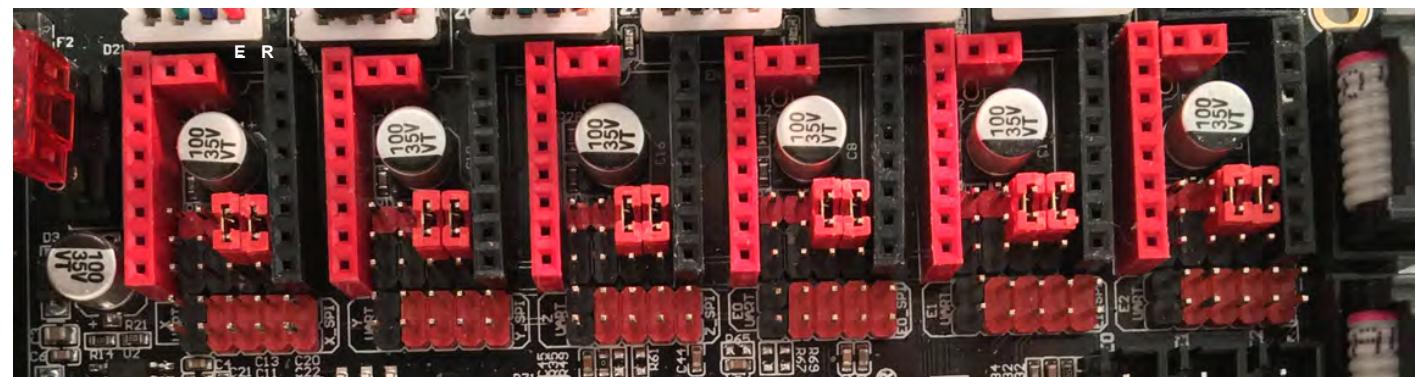
1/64
Interpolation: 1/256
SpreadCycle



Stand Alone Mode - SpreadCycle

EN	-	R	-	VM
CFG1- MS1	-		-	GND
CFG2- MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	2 1	-	B2
STEP	-	2 1	-	VDD
DIR	-	-	-	GND

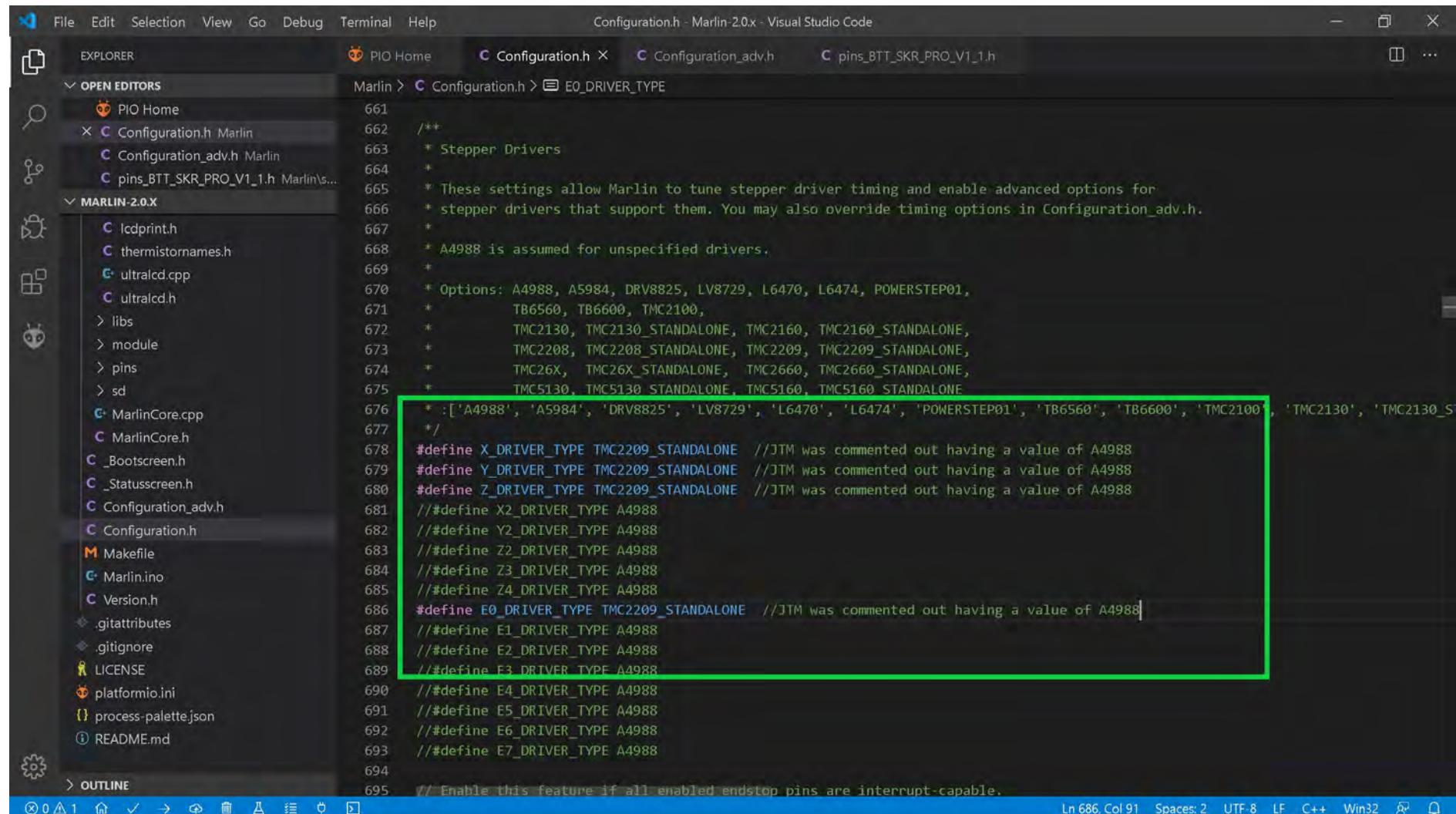
1/16
Interpolation: 1/256
SpreadCycle



The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2209 V1.2 stepper motor drivers in stand alone mode for spreadCycle.

- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand alone mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 *           TB6560, TB6600, TMC2100,
671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 *
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
677 */
678 #define X_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
679 #define Y_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
680 #define Z_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2209_STANDALONE //JTM was commented out having a value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694 */
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

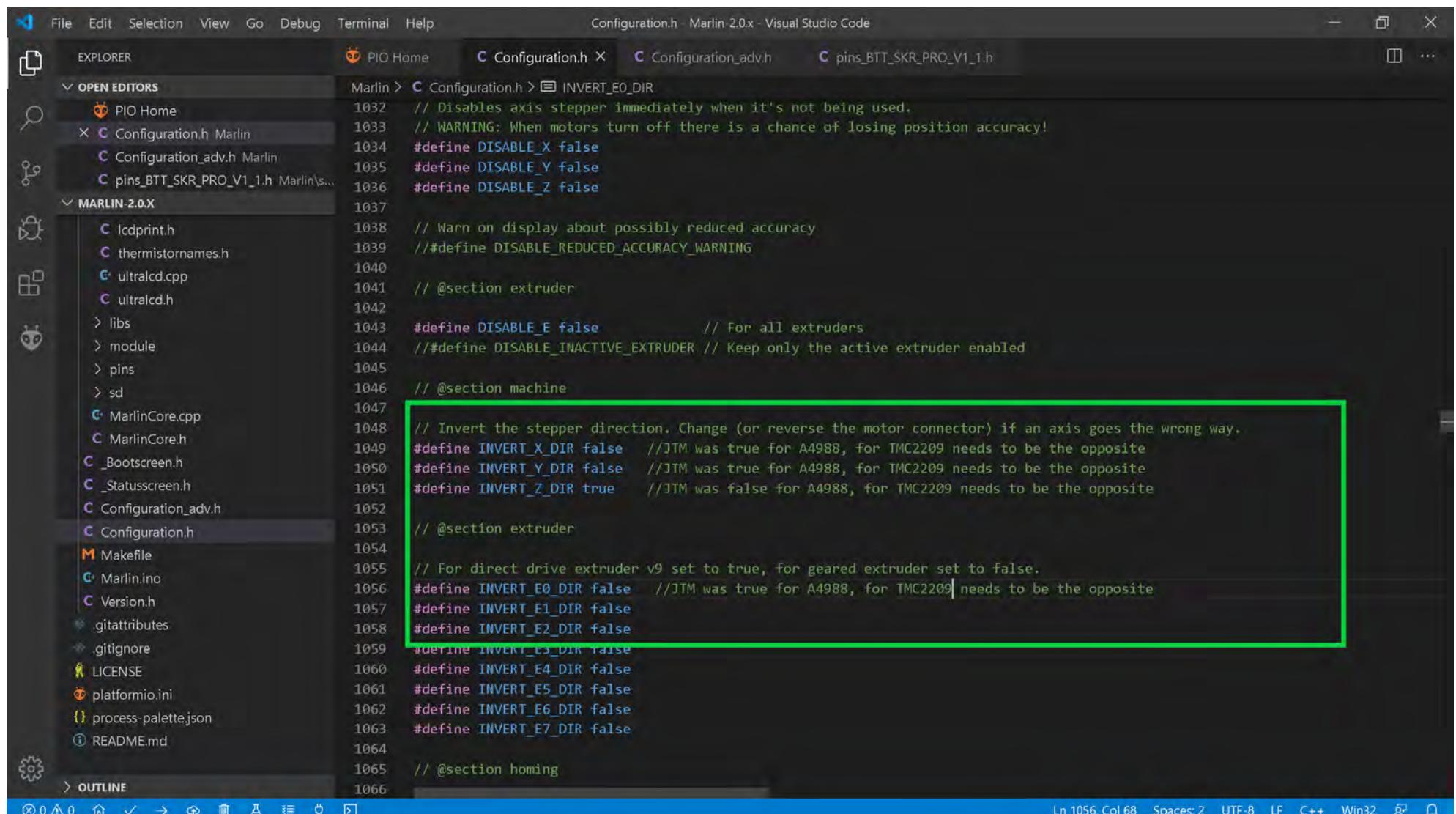
```

A green rectangular box highlights the driver type definitions (lines 678-693), specifically the uncommented line `#define E0_DRIVER_TYPE TMC2209_STANDALONE`. The status bar at the bottom right shows the line number as 686, column 91.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin > Configuration.h > INVERT_E0_DIR

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC2209 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC2209 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC2209 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // JTM was true for A4988, for TMC2209 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

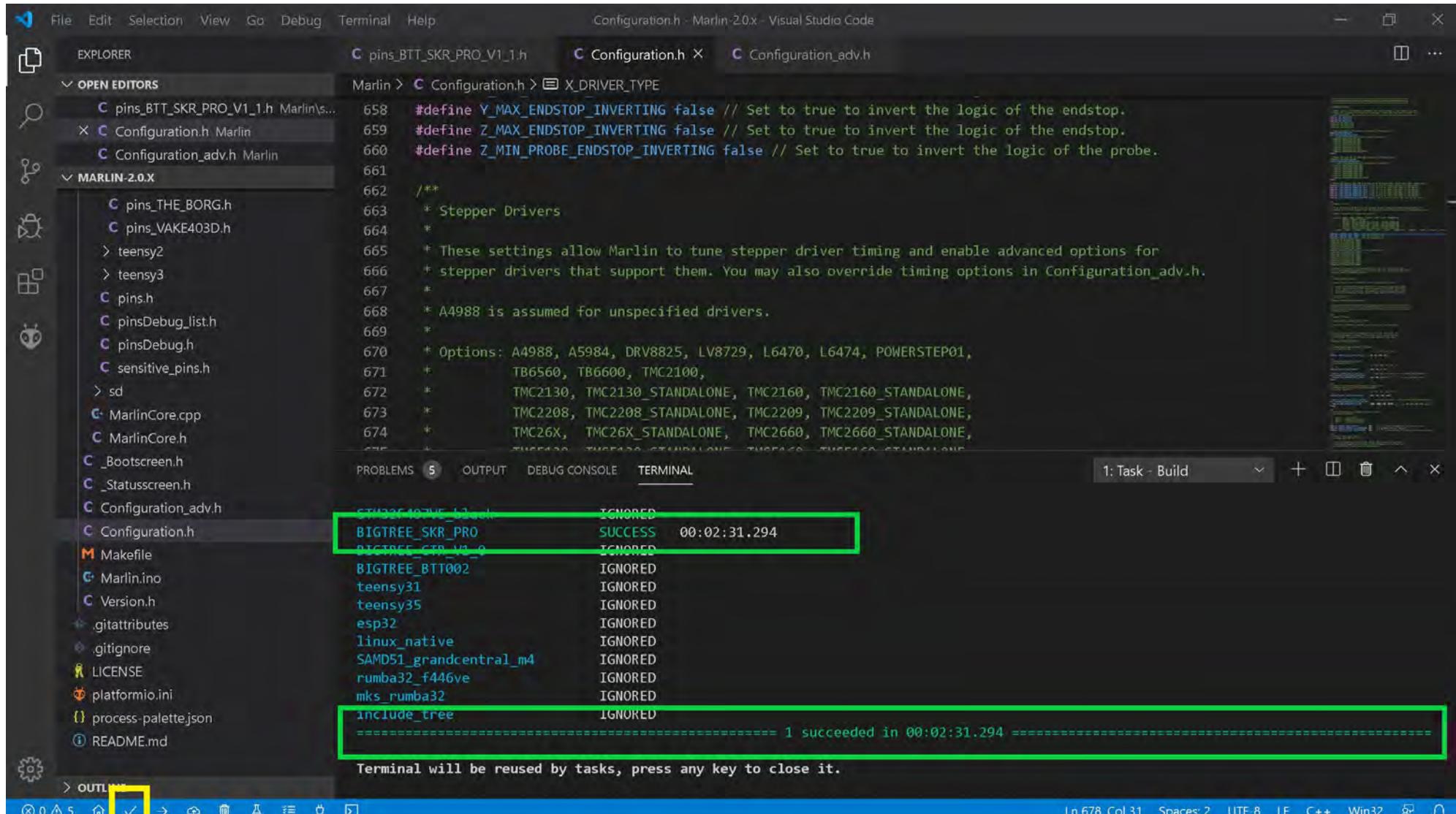
OUTLINE

Ln 1056, Col 68 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle

- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand alone mode for spreadCycle. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



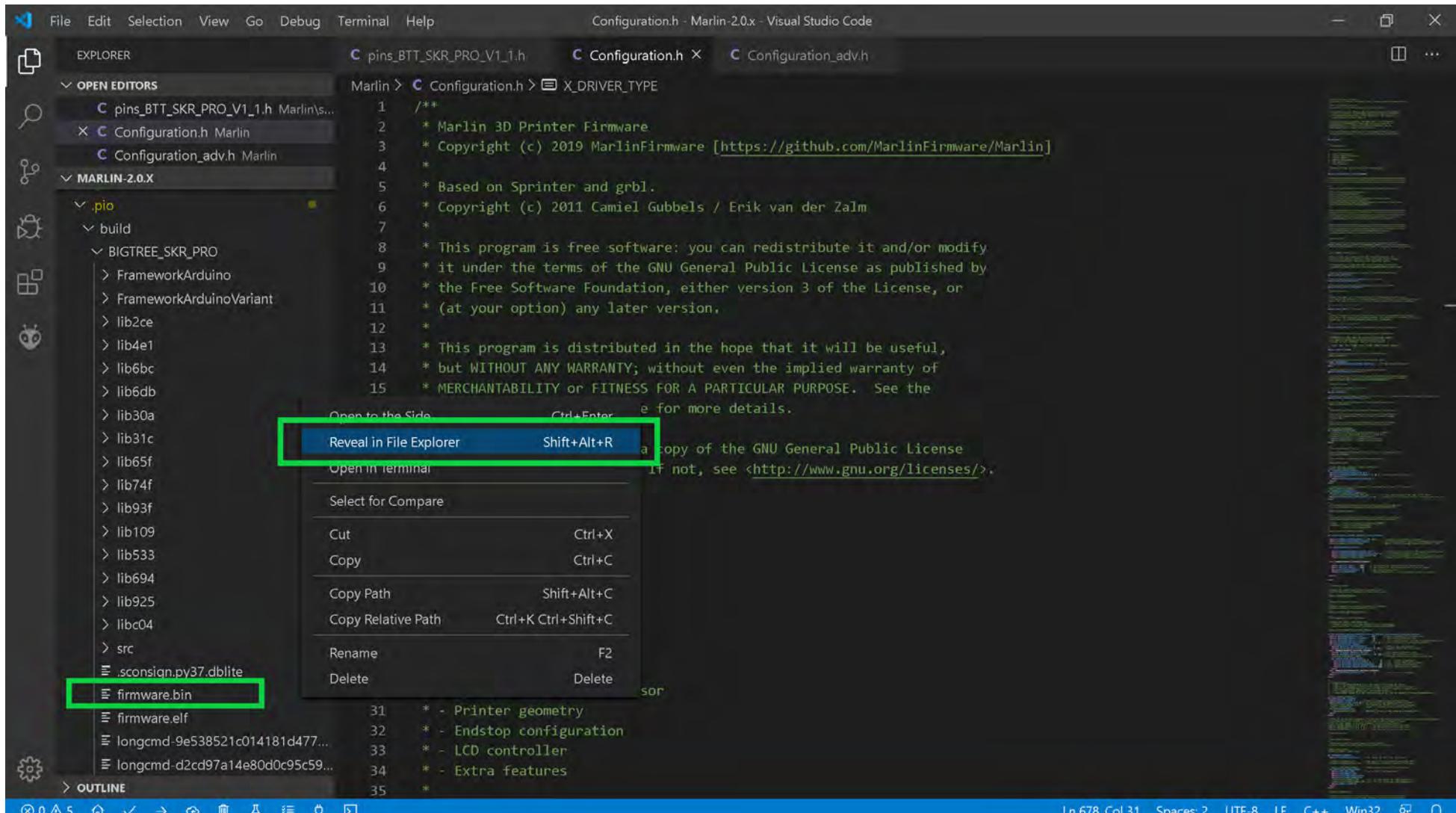
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `pins_BTT_SKR_PRO_V1_1.h`, `Configuration.h`, and `Configuration_adv.h`.
- Terminal:** Displays the build logs for the `STM32F103VE_52.cu` file. The logs show the compilation of `BIGTREE_SKR_PRO` and `BIGTREE_CTP_V1_0` with a success message of `00:02:31.294`. Other files listed include `BIGTREE_BT1002`, `teensy31`, `teensy35`, `esp32`, `linux_native`, `SAMD51_grandcentral_m4`, `rumba32_f446ve`, `mks_rumba32`, and `include_tree`. The terminal concludes with `===== 1 succeeded in 00:02:31.294 =====`.
- Bottom Status Bar:** Shows build statistics: 0 errors, 0 warnings, 5 problems, and a yellow checkmark icon indicating the build was successful.
- Bottom Right:** Shows build metrics: Ln 678, Col 31, Spaces: 2, UTF-8, LF, C++, Win32.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle

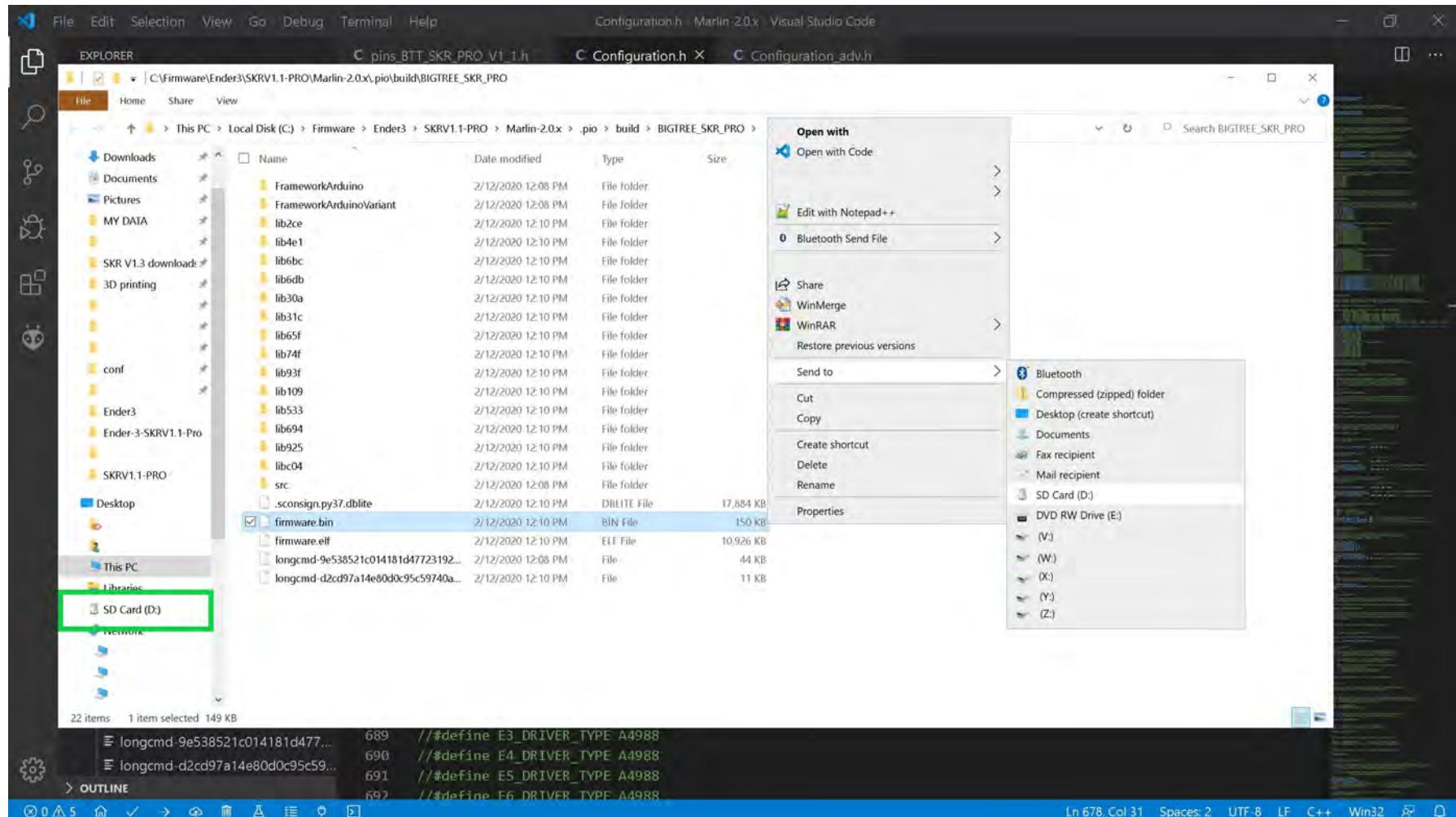
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand Alone Mode for SpreadCycle

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".

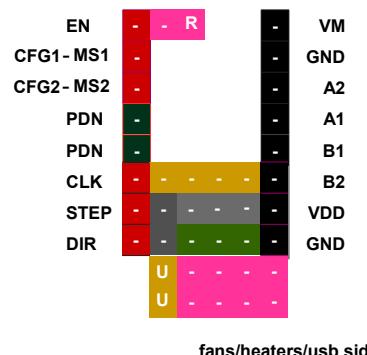


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

UART

R - Stall Detection

base of card



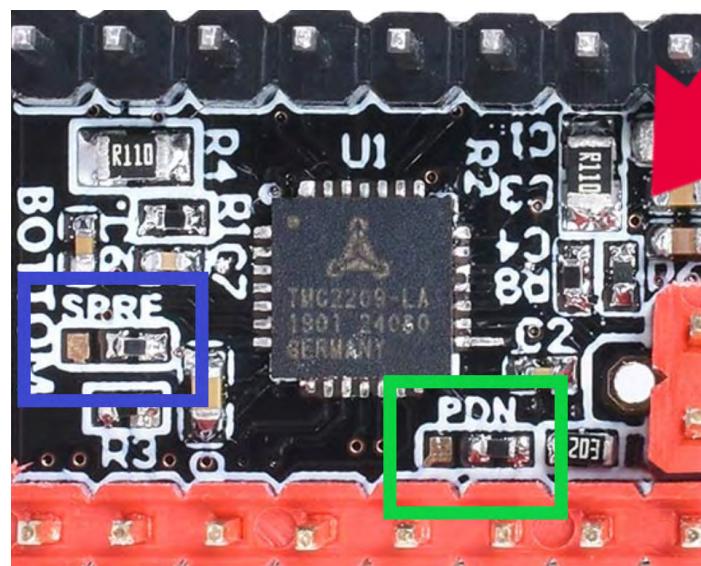
Driver Chip	
TMC2209	
UART mode	
Maximum subdivision 256	
5.5v - 28V DC	
2.8A	
Driving current calculation formula	

Steps are set inside of your Firmware

$$I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$$

$$V_{ref} = (\frac{I_{MAX}}{\sqrt{2}} * 2.5) / 1.77$$

Note: To ensure your TMC2209 is in UART mode, look on the bottom of the driver board for the PDN pads. Two of the three pads should be bridged together. If a bridge exists then the device is in **UART Mode**, as seen in the **GREEN box**.

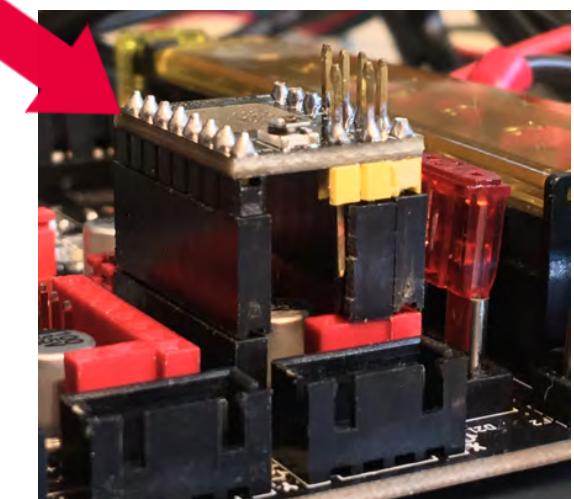


The **BLUE** box shows the device has StealthChop .

Note: You can use 50% to 90% of the calculated Irms (Imax/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

Note: When the stall-guard function is **not used**, the stall-guard pin of the TMC2209 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board

MOST BIQU
TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.



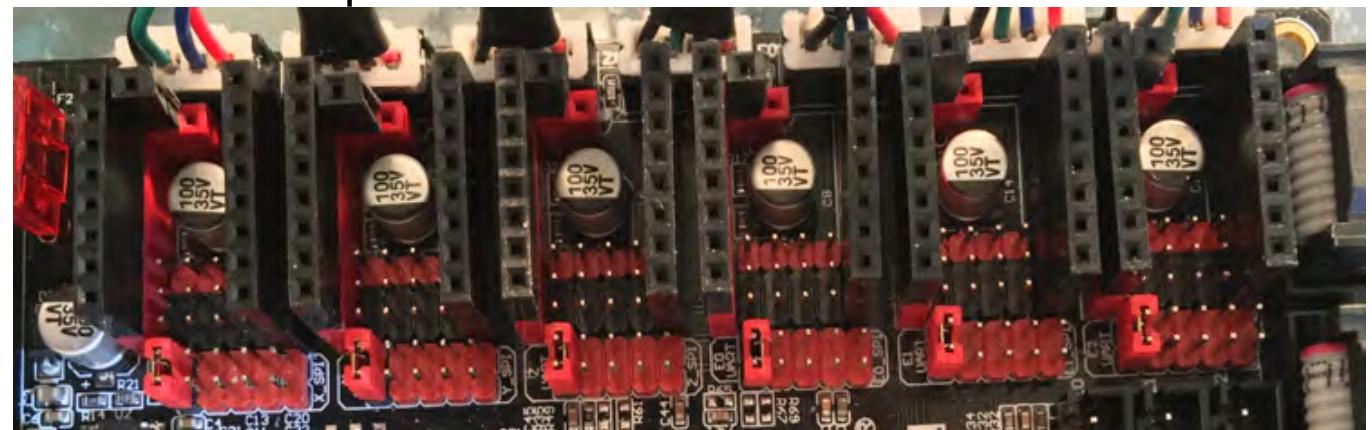
UART

R - Stall Detection

UART Mode

Note: Set Jumper "U" for UART MODE!!

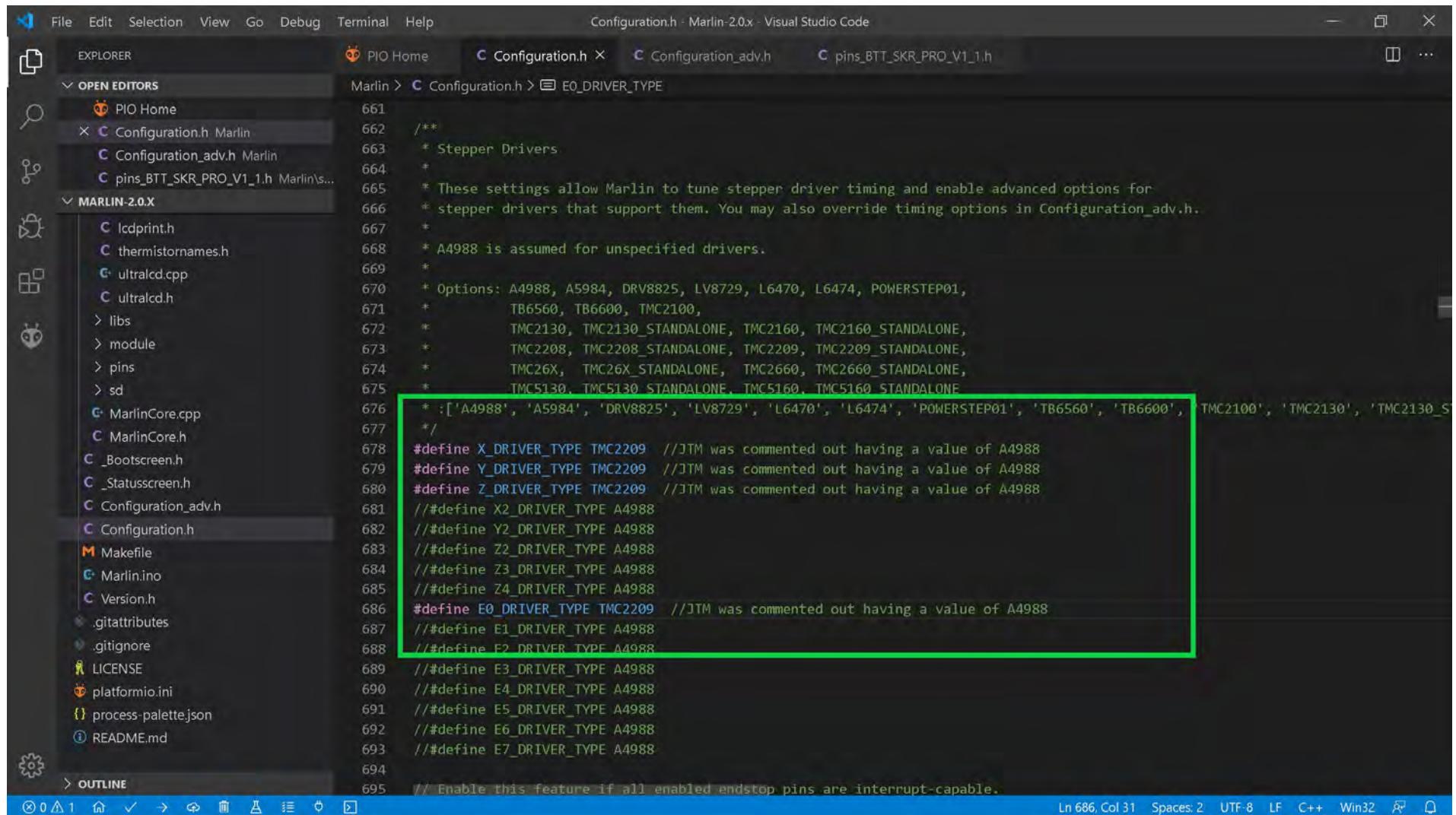
UART		
EN	R	-
CFG1-MS1	-	VM
CFG2-MS2	-	GND
PDN	-	A2
PDN	-	A1
CLK	-	B1
STEP	-	B2
DIR	-	VDD
	U	GND
	U	-



The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC2209 V1.2 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in UART mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following driver type definitions:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 *           TB6560, TB6600, TMC2100,
672 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_S
677 */
678 #define X_DRIVER_TYPE TMC2209 //JTM was commented out having a value of A4988
679 #define Y_DRIVER_TYPE TMC2209 //JTM was commented out having a value of A4988
680 #define Z_DRIVER_TYPE TMC2209 //JTM was commented out having a value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2209 //JTM was commented out having a value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

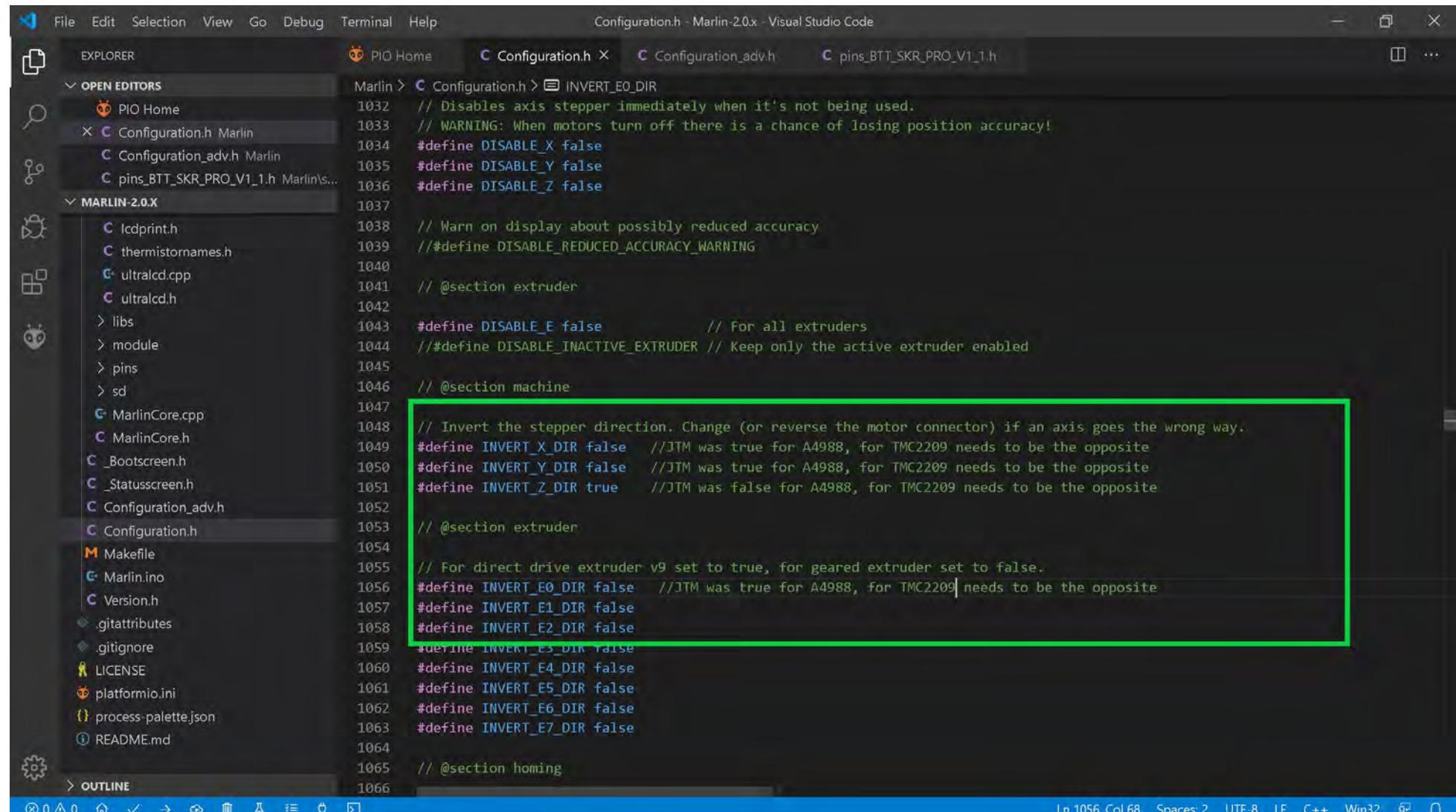
```

A green rectangular box highlights the driver type definitions for X, Y, Z, and E0 axes, specifically the lines starting with '#define' and the subsequent driver type names.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITORS Marlin Configuration.h INVERT_E0_DIR

```

1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false    // JTM was true for A4988, for TMC2209 needs to be the opposite
1050 #define INVERT_Y_DIR false    // JTM was true for A4988, for TMC2209 needs to be the opposite
1051 #define INVERT_Z_DIR true     // JTM was false for A4988, for TMC2209 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false   // JTM was true for A4988, for TMC2209 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

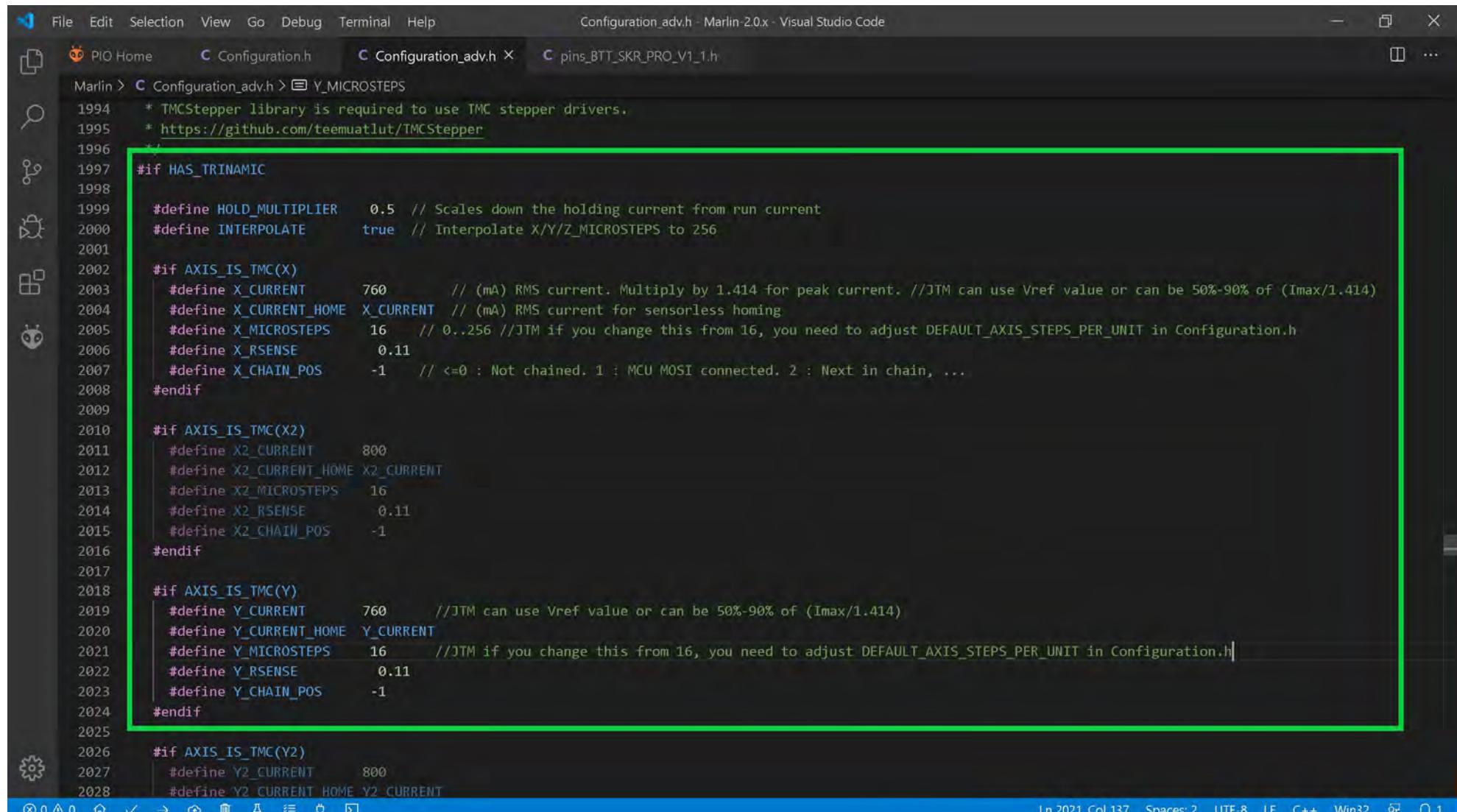
```

Ln 1056, Col 68 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- next you want to set your Vref in the Marlin firmware for each axis that has the TMC2130 driver, as seen in the GREEN box below. I changed the "X_CURRENT" to be the calculated Vref for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated Vref for my Y-Axis, which is 760mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration_adv.h > Y_MICROSTEPS
1994 * TMCSstepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCSstepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999     #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000     #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002     #if AXIS_IS_TMC(X)
2003         #define X_CURRENT          760 // (mA) RMS current. Multiply by 1.414 for peak current. //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2004         #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005         #define X_MICROSTEPS      16 // 0..256 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2006         #define X_RSENSE           0.11
2007         #define X_CHAIN_POS        -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008     #endif
2009
2010     #if AXIS_IS_TMC(X2)
2011         #define X2_CURRENT         800
2012         #define X2_CURRENT_HOME   X2_CURRENT
2013         #define X2_MICROSTEPS     16
2014         #define X2_RSENSE          0.11
2015         #define X2_CHAIN_POS       -1
2016     #endif
2017
2018     #if AXIS_IS_TMC(Y)
2019         #define Y_CURRENT          760 //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2020         #define Y_CURRENT_HOME    Y_CURRENT
2021         #define Y_MICROSTEPS      16 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2022         #define Y_RSENSE           0.11
2023         #define Y_CHAIN_POS        -1
2024     #endif
2025
2026     #if AXIS_IS_TMC(Y2)
2027         #define Y2_CURRENT         800
2028         #define Y2_CURRENT_HOME   Y2_CURRENT

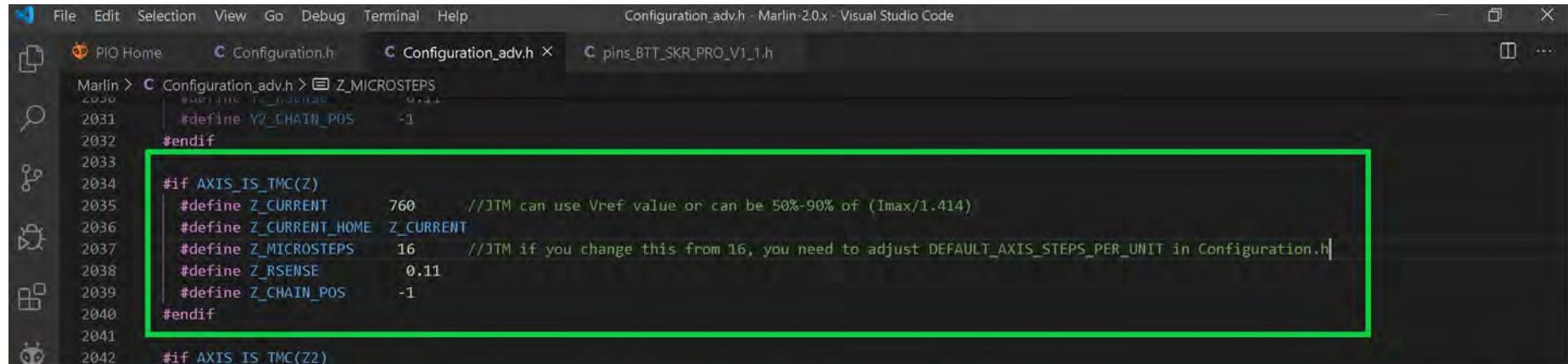
```

Ln 2021, Col 137 Spaces:2 UTF-8 LF C++ Win32 ⌂ 1

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

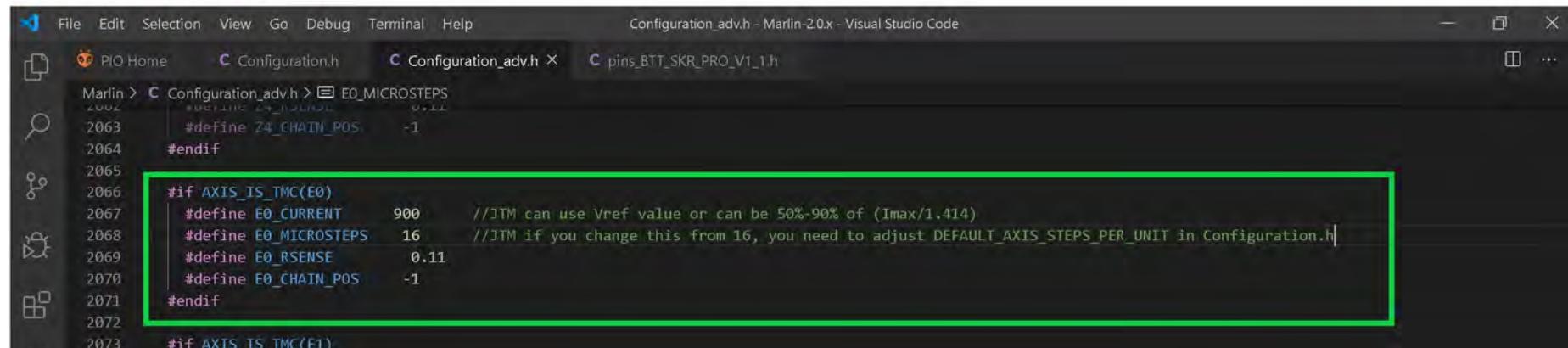
- Now, I am setting the Vref for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated Vref for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated Vref for my Extruder, which is 900mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > Z_MICROSTEPS
2030
2031     #define Z2_CHAIN_POS -1
2032
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT    760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2038         #define Z_RSENSE   0.11
2039         #define Z_CHAIN_POS -1
2040     #endif
2041
2042     #if AXIS_IS_TMC(Z2)

```

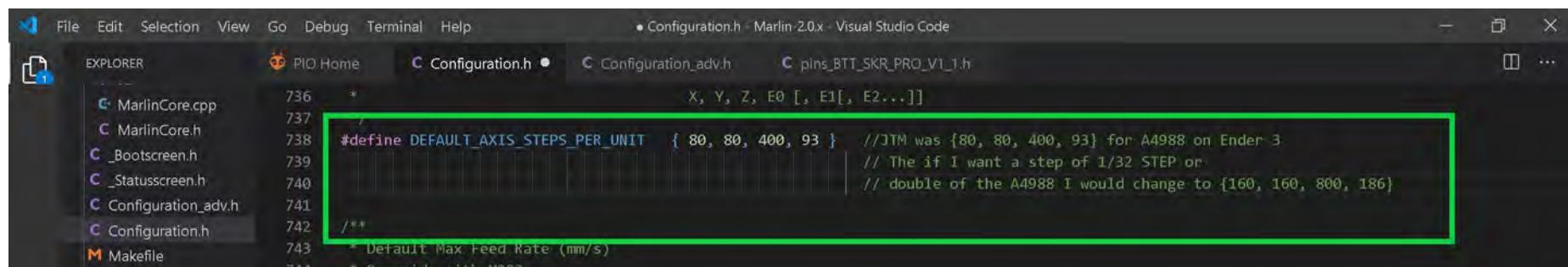


```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > E0_MICROSTEPS
2062
2063     #define Z4_CHAIN_POS -1
2064
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT    900      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2068         #define E0_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2069         #define E0_RSENSE   0.11
2070         #define E0_CHAIN_POS -1
2071     #endif
2072
2073     #if AXIS_IS_TMC(E1)

```

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



```

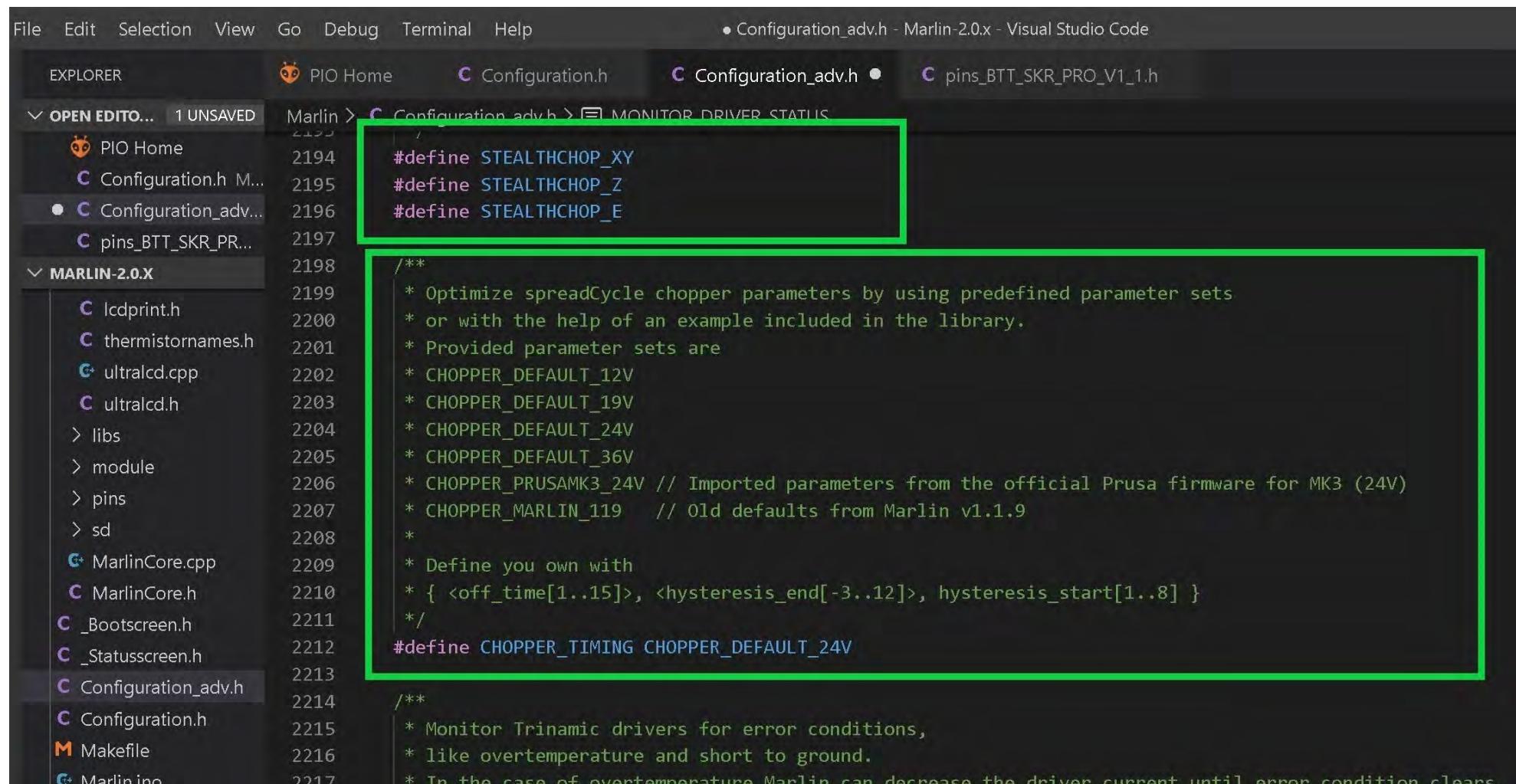
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
736     *
737     /
738     #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
739                                         // The if I want a step of 1/32 STEP or
740                                         // double of the A4988 I would change to {160, 160, 800, 186}
741
742     /**
743     * Default Max Feed Rate (mm/s)
744     * Override with M203

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if the proper "CHOPPER_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER_TIMING" is correct.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
- Explorer:**
 - OPEN EDITOR... 1 UNSAVED
 - PIO Home
 - Configuration.h M...
 - Configuration_adv...
 - pins_BTT_SKR_PR...
- Editor:** The Configuration_adv.h file is open, showing the following code block highlighted with a green rectangle:

```

#define STEALTHCHOP_XY
#define STEALTHCHOP_Z
#define STEALTHCHOP_E

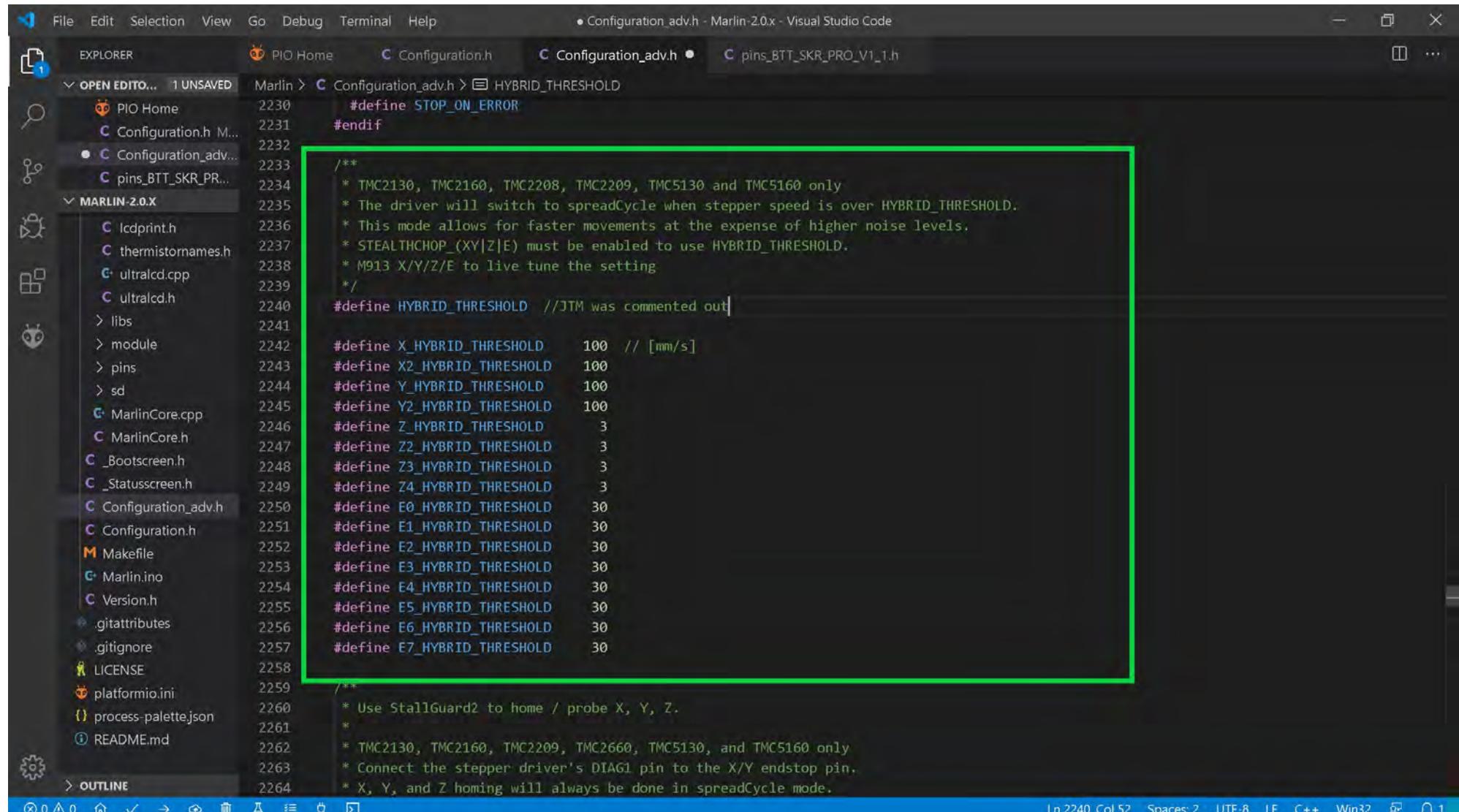
/**
 * Optimize spreadCycle chopper parameters by using predefined parameter sets
 * or with the help of an example included in the library.
 * Provided parameter sets are
 * CHOPPER_DEFAULT_12V
 * CHOPPER_DEFAULT_19V
 * CHOPPER_DEFAULT_24V
 * CHOPPER_DEFAULT_36V
 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
 *
 * Define your own with
 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
 */
#define CHOPPER_TIMING CHOPPER_DEFAULT_24V

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays the Marlin firmware configuration. A green box highlights the `HYBRID_THRESHOLD` section. The code in this section is as follows:

```

/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //JTM was commented out

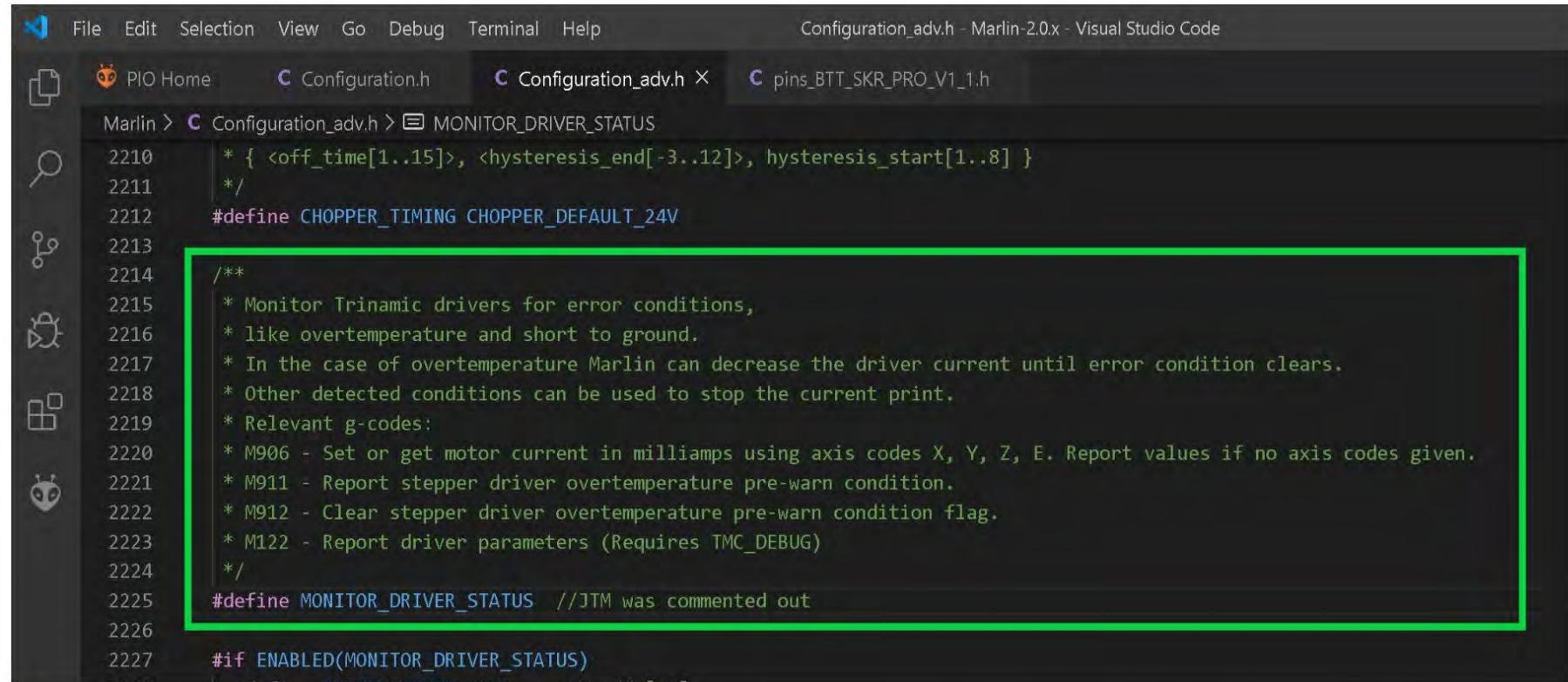
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

/*
 * Use StallGuard2 to home / probe X, Y, Z.
 *
 * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
 * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
 * X, Y, and Z homing will always be done in spreadCycle mode.
 */

```

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in Marlin's documentation located [on-line](#).
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

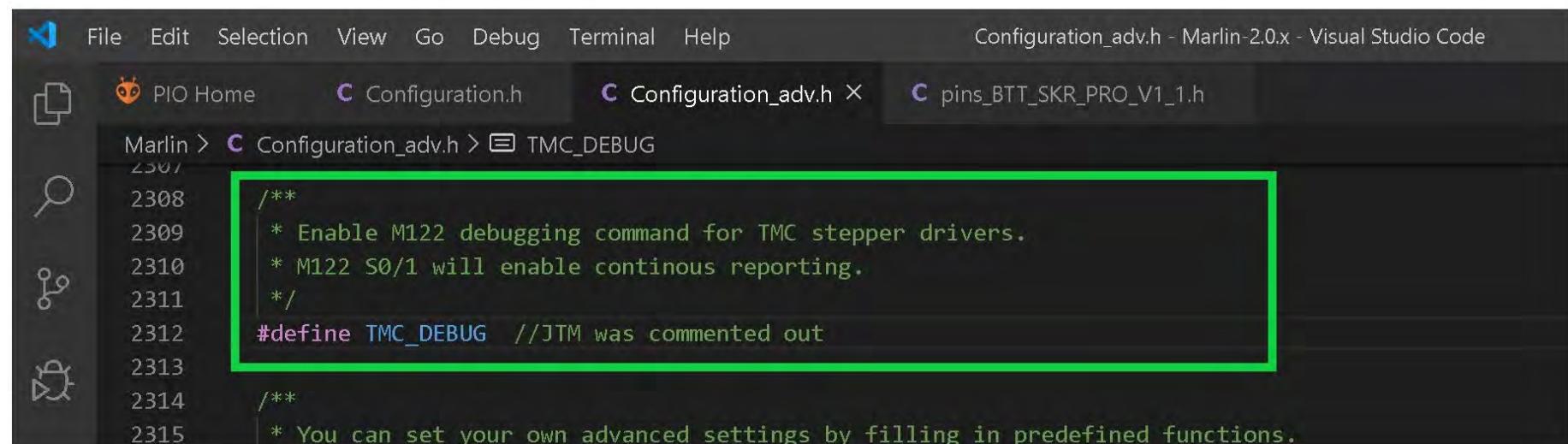
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

2210     * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211     */
2212     #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //JTM was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > TMC_DEBUG

```

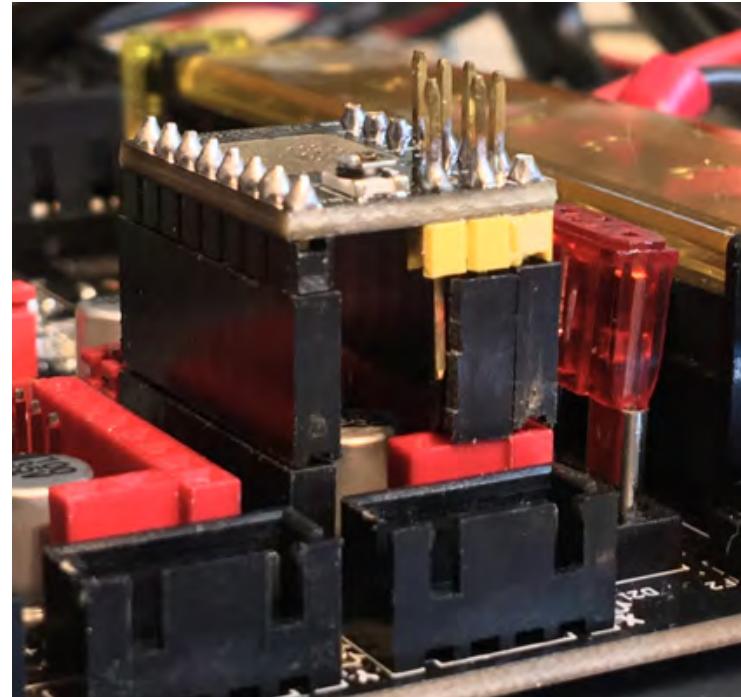
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //JTM was commented out
2313
2314 /**
2315 * You can set your own advanced settings by filling in predefined functions.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- This next section covers sensor-less homing which is available for the TMC2209 in UART mode. I want to enable it so I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG pin on the TMC2209 driver has to be plugged into the SKR PRO V1.1 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG pin on the Z axis' TMC2209 is NOT connected to the board. I plan to plug my Z axis TMC2209 into my SKR PRO V1.1 board by using long stackable header pin risers, as seen in the picture below.



- sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x : Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

OPEN EDITOR... 1 UNSAVED Marlin > Configuration_adv.h > Y_STALL_SENSITIVITY

```

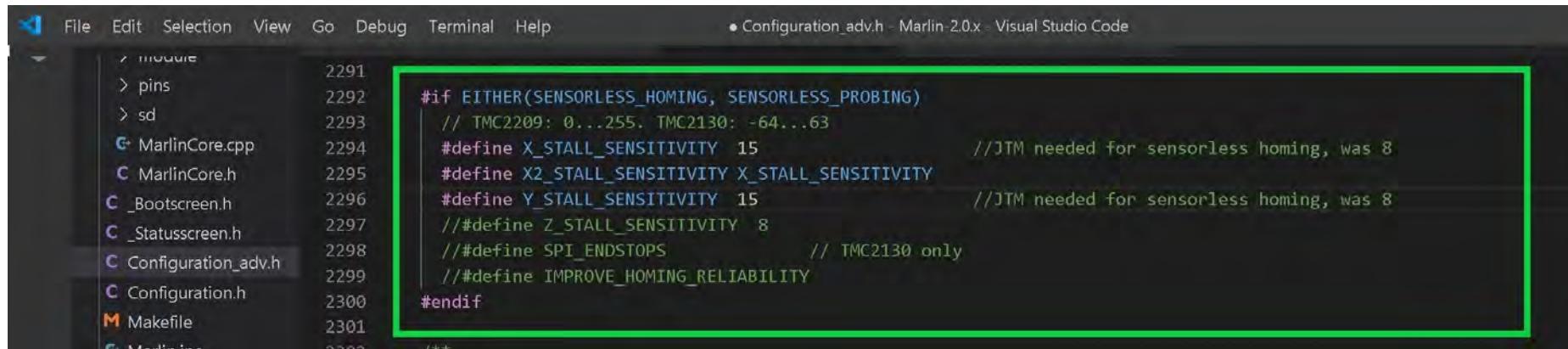
2279 * IMPROVE_HOMING_RELIABILITY tunes acceleration and jerk when
2280 * homing and adds a guard period for endstop triggering.
2281 */
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //JTM was commented out
2283 /**
2284 * Use StallGuard2 to probe the bed with the nozzle.
2285

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Next we set a "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X_STALL_SENSITIVITY" and the "Y_STALL_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z_STALL_SENSITIVITY".



```

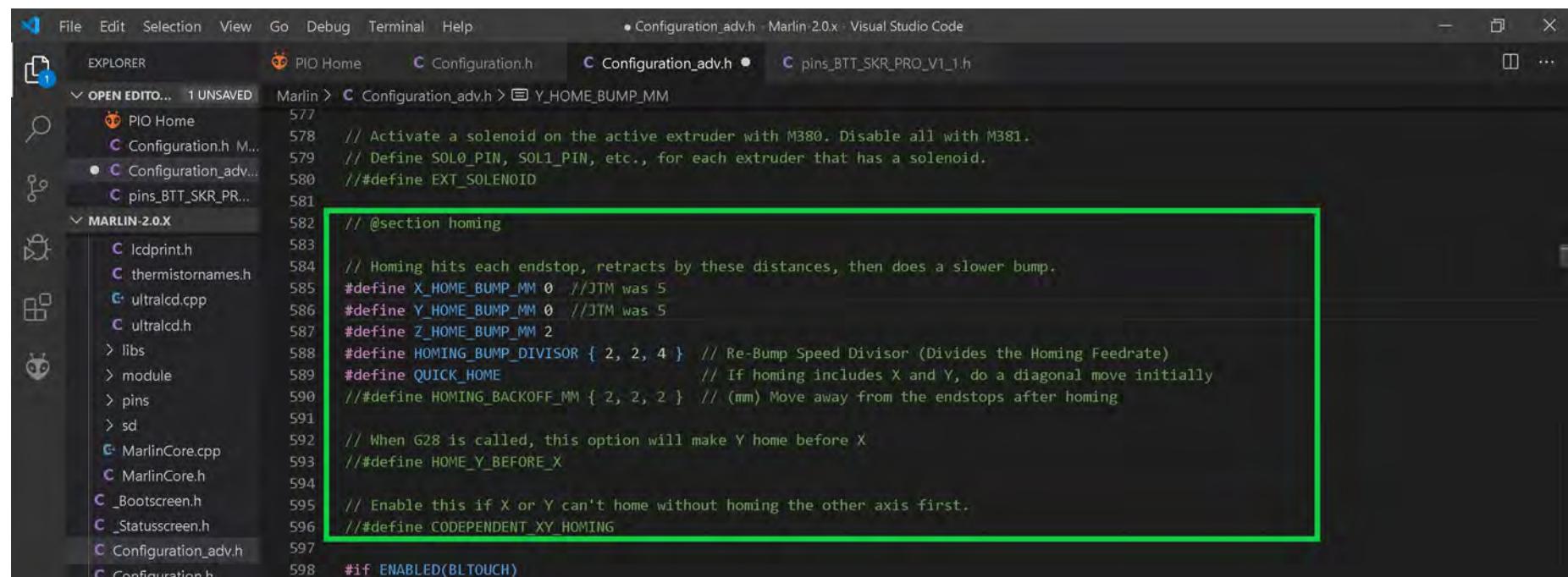
File Edit Selection View Go Debug Terminal Help • Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

MODULES
pins.h 2291
sd.h 2292
MarlinCore.cpp 2293
MarlinCore.h 2294
Bootscreen.h 2295
Statusscreen.h 2296
Configuration_adv.h 2297
Configuration.h 2298
Makefile 2299
Mainscreen.h 2300
pins_BTT_SKR_PRO_V1_1.h 2301
pins_BTT_SKR_PRO_V1_1.h 2302
***/ 2303

#if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
// TMC2209: 0...255. TMC2130: -64...63
#define X_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
#define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
#define Y_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//#define Z_STALL_SENSITIVITY 8
//#define SPI_ENDSTOPS // TMC2130 only
//#define IMPROVE_HOMING_RELIABILITY
#endif

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.



```

File Edit Selection View Go Debug Terminal Help • Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

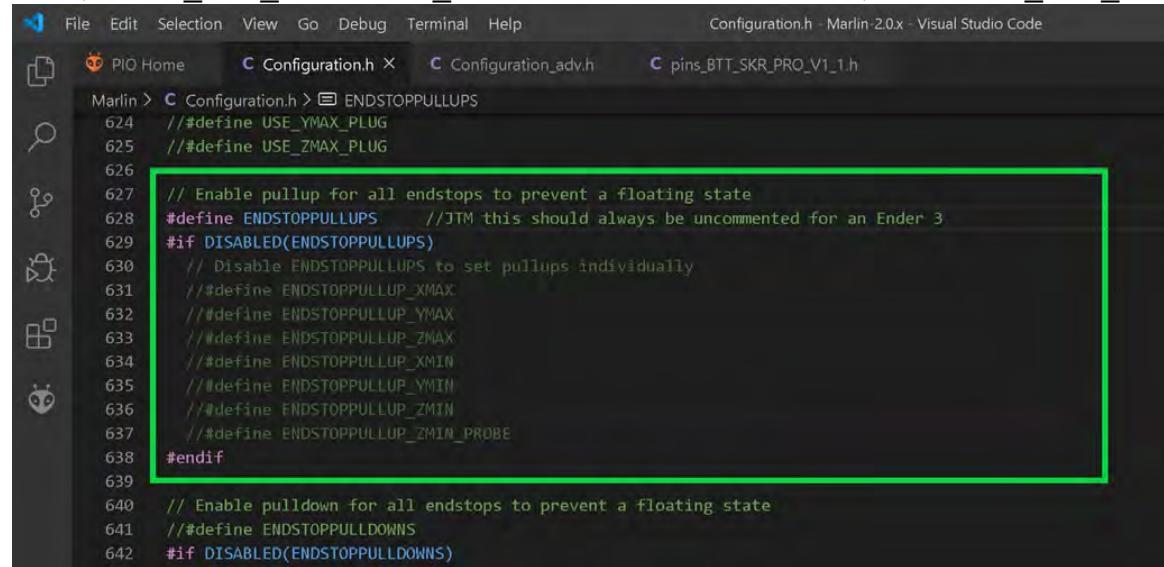
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITOR 1 UNSAVED Marlin > Configuration_adv.h > Y_HOME_BUMP_MM
577
578 // Activate a solenoid on the active extruder with M380. Disable all with M381.
579 // Define SOLO_PIN, SOL1_PIN, etc., for each extruder that has a solenoid.
580 //#define EXT_SOLENOID
581
582 // @section homing
583
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //JTM was 5
586 #define Y_HOME_BUMP_MM 0 //JTM was 5
587 #define Z_HOME_BUMP_MM 2
588 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
589 #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
590 //#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
591
592 // When G28 is called, this option will make Y home before X
593 //#define HOME_Y_BEFORE_X
594
595 // Enable this if X or Y can't home without homing the other axis first.
596 //#define CODEPENDENT_XY_HOMING
597
598 #if ENABLED(BLTOUCH)

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

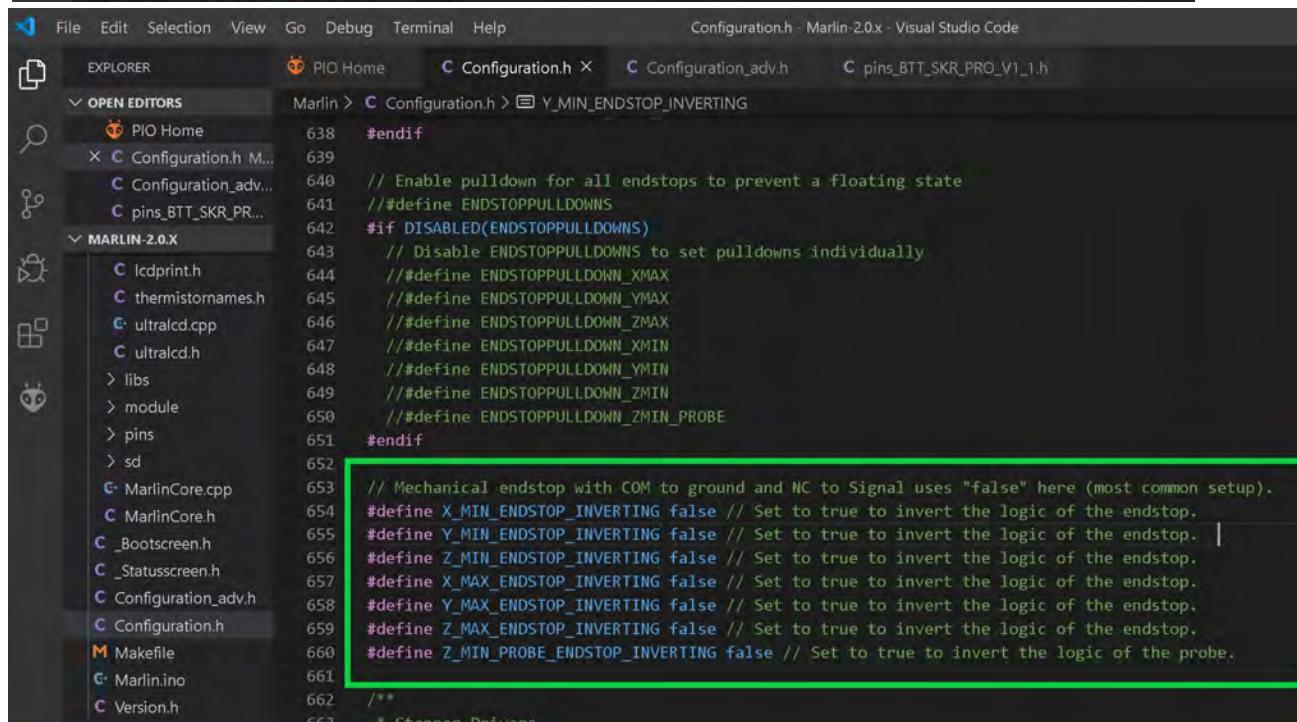
- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default. I also want to check to see how our MIN_ENDSTOP_INVERTINGs are set the right way. For an Ender 3 using TMC2209 drivers the "X_MIN_ENDSTOP_INVERTING" should be false, the "Y_MIN_ENDSTOP_INVERTING" should be false, and the "Z_MIN_ENDSTOP_INVERTING" should be false.



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration.h > ENDSTOPPULLUPS
624 // #define USE_YMAX_PLUG
625 // #define USE_ZMAX_PLUG
626
627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS // JTM this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 // #define ENDSTOPPULLUP_XMAX
632 // #define ENDSTOPPULLUP_YMAX
633 // #define ENDSTOPPULLUP_ZMAX
634 // #define ENDSTOPPULLUP_XMIN
635 // #define ENDSTOPPULLUP_YMIN
636 // #define ENDSTOPPULLUP_ZMIN
637 // #define ENDSTOPPULLUP_ZMIN_PROBE
638#endif
639
640 // Enable pulldown for all endstops to prevent a floating state
641 // #define ENDSTOPPULLDOWNS
642 #if DISABLED(ENDSTOPPULLDOWNS)

```



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration.h > Y_MIN_ENDSTOP_INVERTING
638#endif
639
640 // Enable pulldown for all endstops to prevent a floating state
641 // #define ENDSTOPPULLDOWNS
642 #if DISABLED(ENDSTOPPULLDOWNS)
643 // Disable ENDSTOPPULLDOWNS to set pulldowns individually
644 // #define ENDSTOPPULLDOWN_XMAX
645 // #define ENDSTOPPULLDOWN_YMAX
646 // #define ENDSTOPPULLDOWN_ZMAX
647 // #define ENDSTOPPULLDOWN_XMIN
648 // #define ENDSTOPPULLDOWN_YMIN
649 // #define ENDSTOPPULLDOWN_ZMIN
650 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
651#endif
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
655 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**
663 * Stepper Drivers

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- The end of Marlin setup for BIQU TMC2209 drivers in UART mode . Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

A screenshot of the Visual Studio Code interface showing the build process for Marlin firmware. The terminal window displays the compilation results:

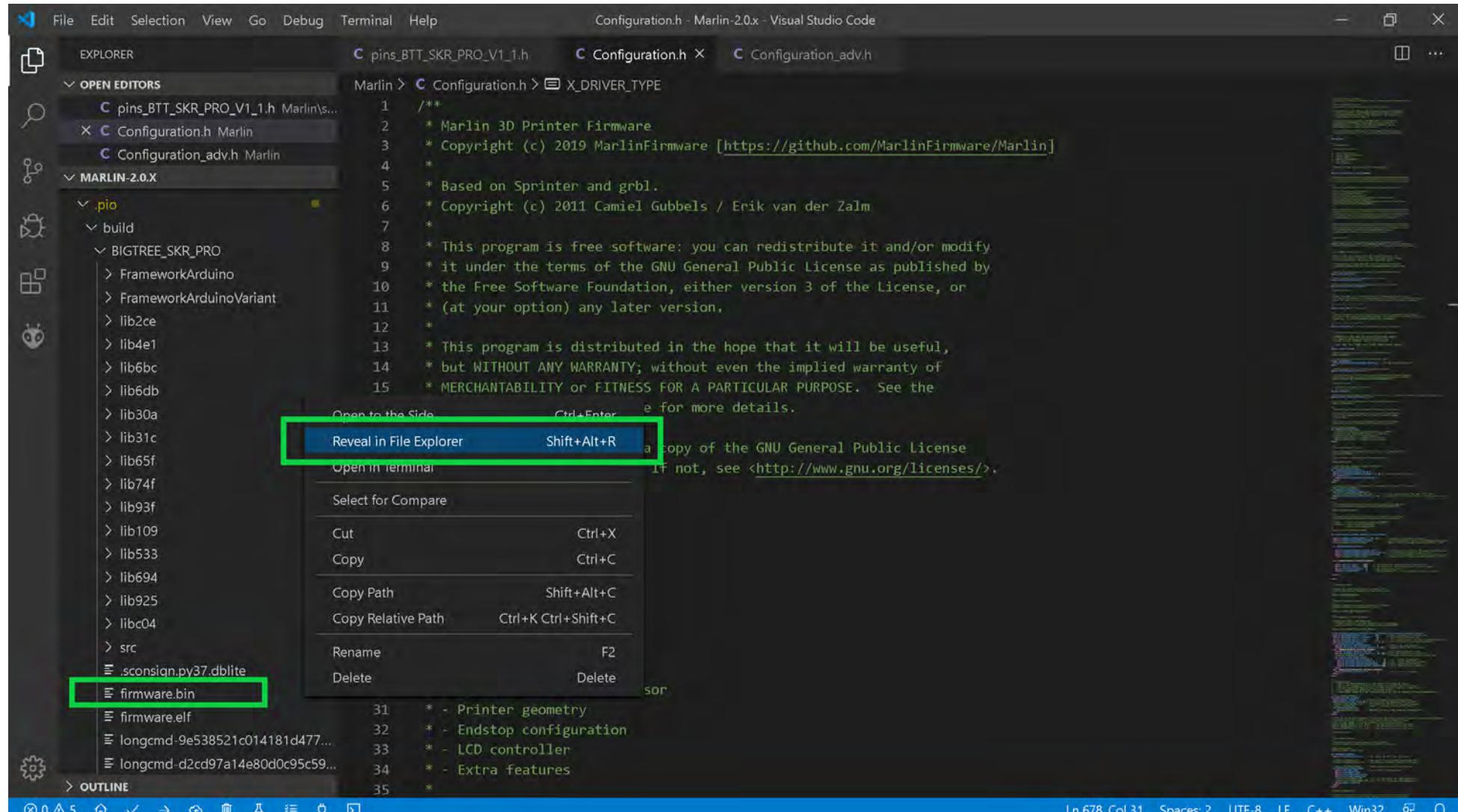
```
STH32E102VE_52_v1.1 BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1.0 IGNORED
BIGTREE_BT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```

The terminal message indicates 1 successful compilation in 00:02:31.294. The entire terminal output is highlighted with a green border.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

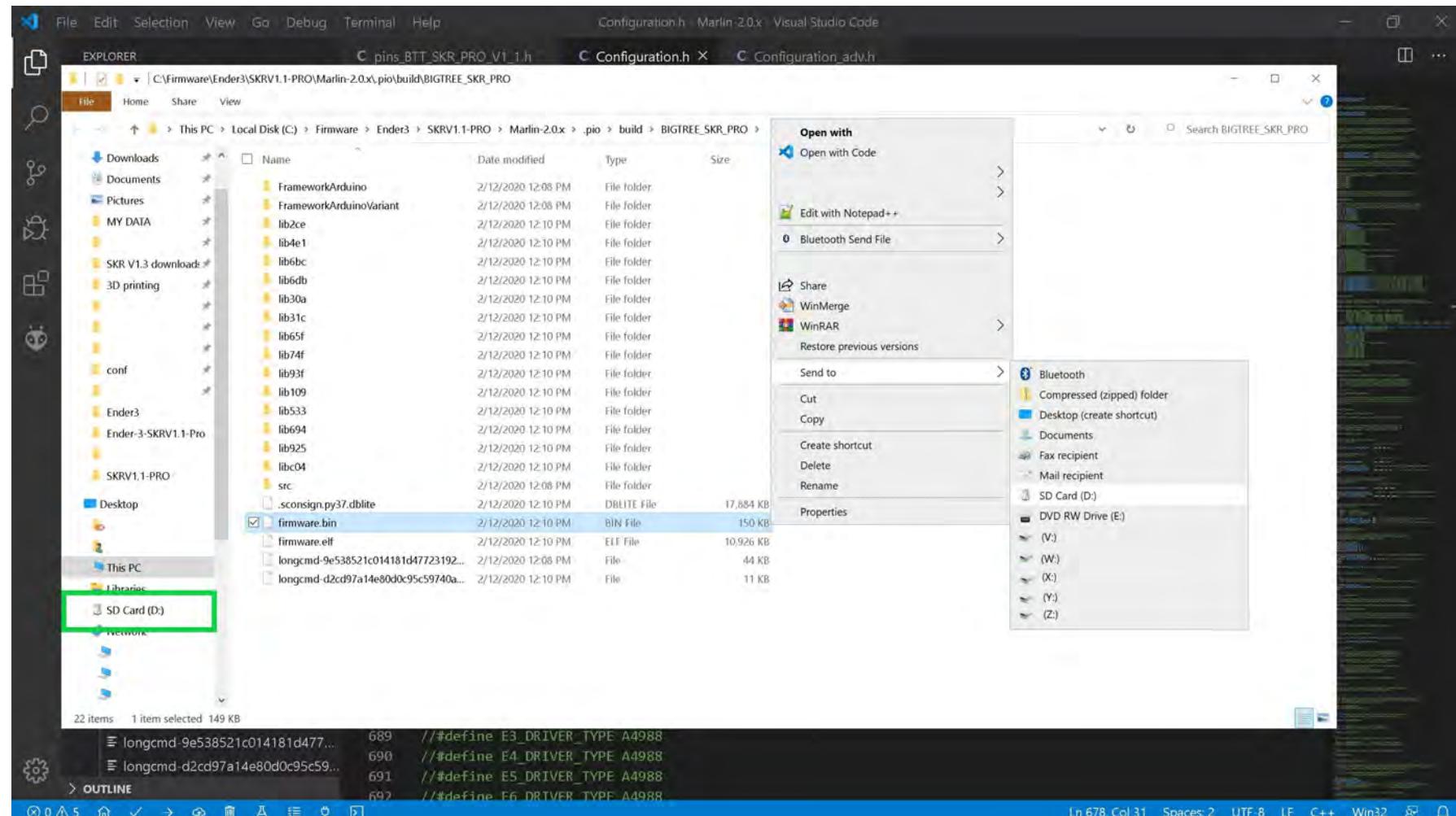
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



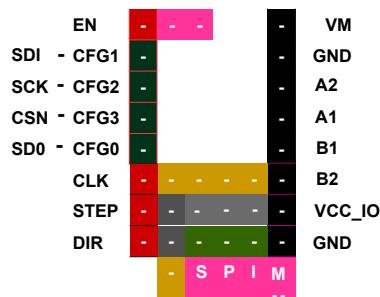
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

BIQU TMC5160 V1.2

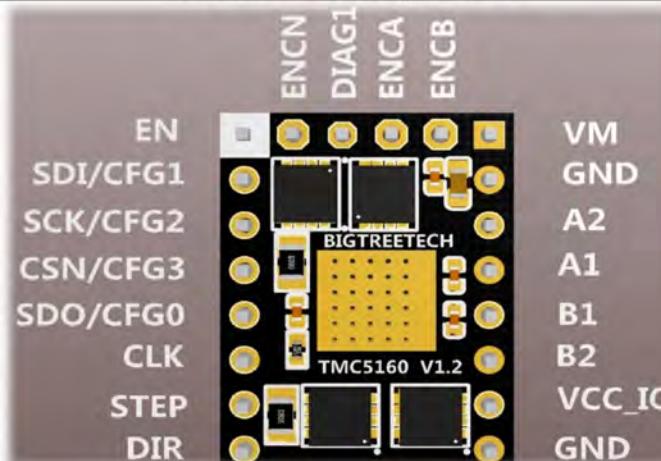
SPI Mode

SPI

base of card

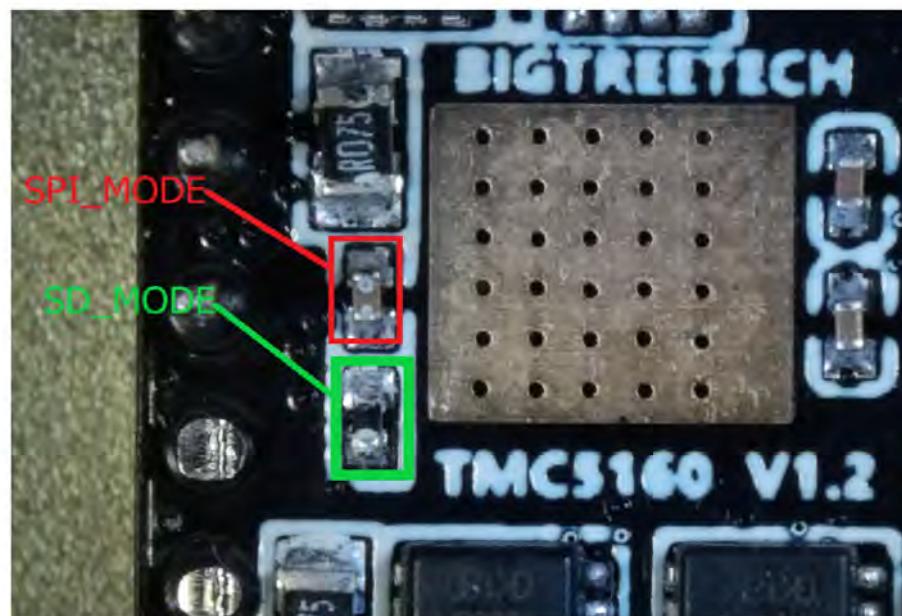


fans/heaters/usb side

IMPORTANT: The default configuration for the TMC5160 is SPI MODE!!**Pin Description**

NOTE: By default the TMC5160 is in SPI Mode, which means BOTH SPI_MODE and SD_MODE are jumpered to VCC_IO pin

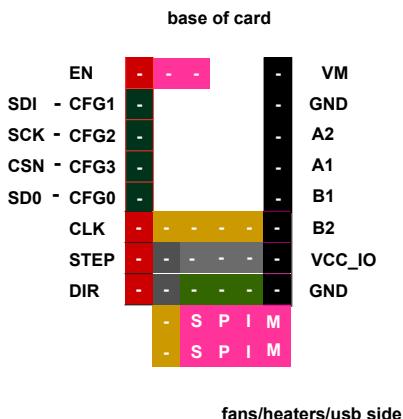
Note: To place TMC5160 into UART Mode, you MUST desolder one end on both SPI_MODE and SD_MODE jumper and then solder that end to GND pin.



Note: To place TMC5160 into SPI and Motion Controller Mode, you must desolder one end of SD_MODE jumper and then solder that end to GND pin on the Driver Board.

NOTE: By default the TMC5160 is in SPI Mode, which means BOTH SPI_MODE and SD_MODE are jumpered to VCC_IO pin

Important: Marlin supports TMC5160 in SPI MODE ONLY

BIQU TMC5160 V1.2SPI Mode**SPI**

TMC5160

SPI Mode

10v - 35v DC

4.2A

Maximum subdivision 256

**Steps are set inside
of your Firmware**

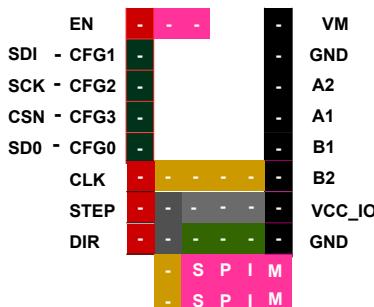
SD_MODE	SPI_MODE	Mode
GND	GND	UART enabled
GND	VIO	SPI and Motion Controller enabled
VIO	GND	Standalone
VIO	VIO	SPI enabled

Driving current calculation formula $I_{MAX} = (((V_{ref} * 1.77) / 2.5) * \sqrt{2})$

$$V_{ref} = (\frac{I_{MAX} * 2.5}{\sqrt{2}}) / 1.77$$

Note: The TMC5160 V1.2 by default comes in SPI mode, which means that SD_MODE and SPI_MODE are already jumpered to VIO (VCC_IO) on the driver board. The BIQU TMC5160 does NOT come with a POT or "Vref Test point" location because the Imax is set inside of the Firmware.

Note: You can use 50% to 90% of the calculated Irms (Imax/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

SPI

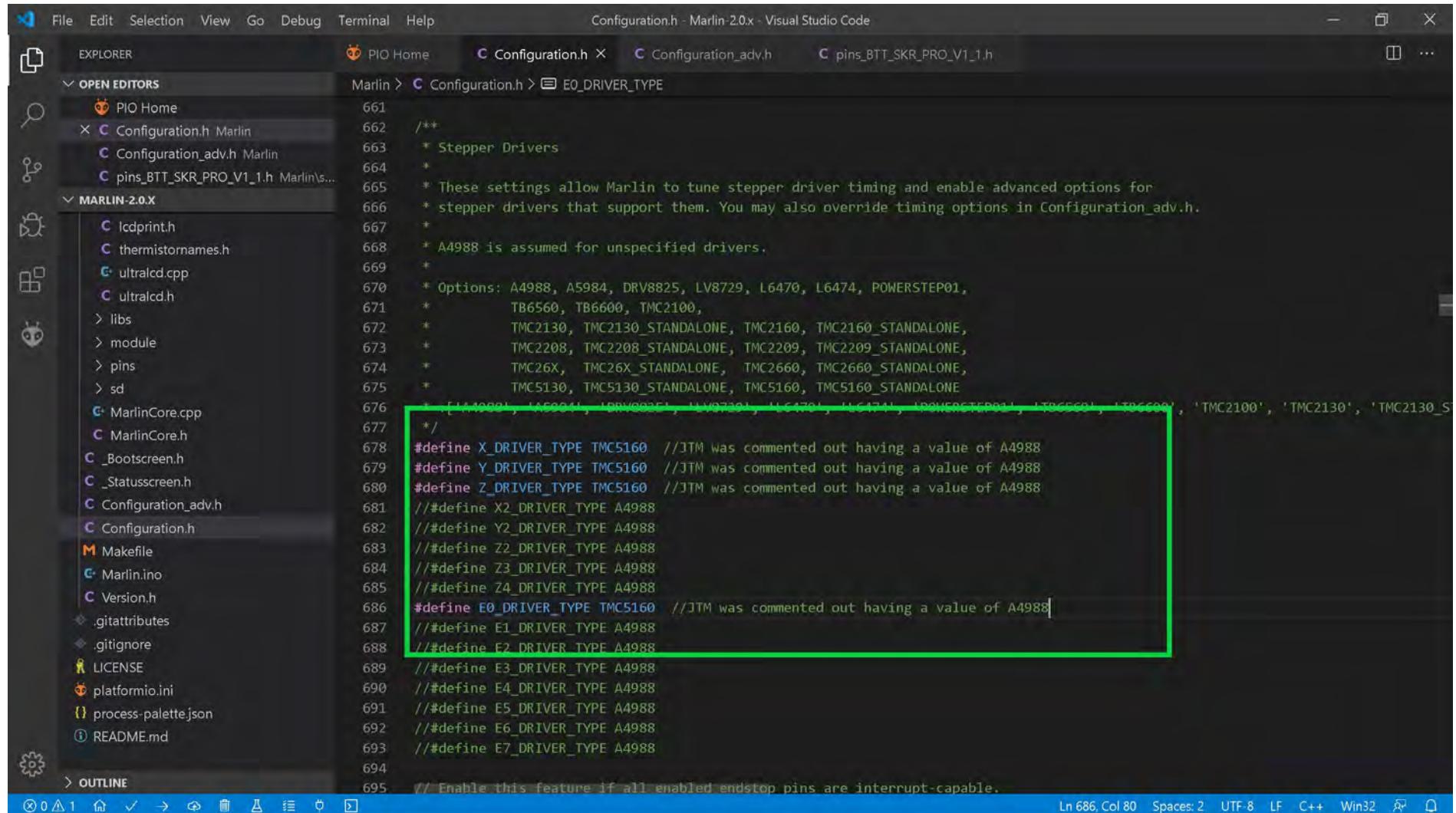
Note: Set JUMPERS "S","P","I","M" on the board!!



The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

NOTE: Go to Appendix C then come back here for the changes to Marlin for BIQU TMC5160 V1.2 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC5160 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5160 drivers in SPI mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



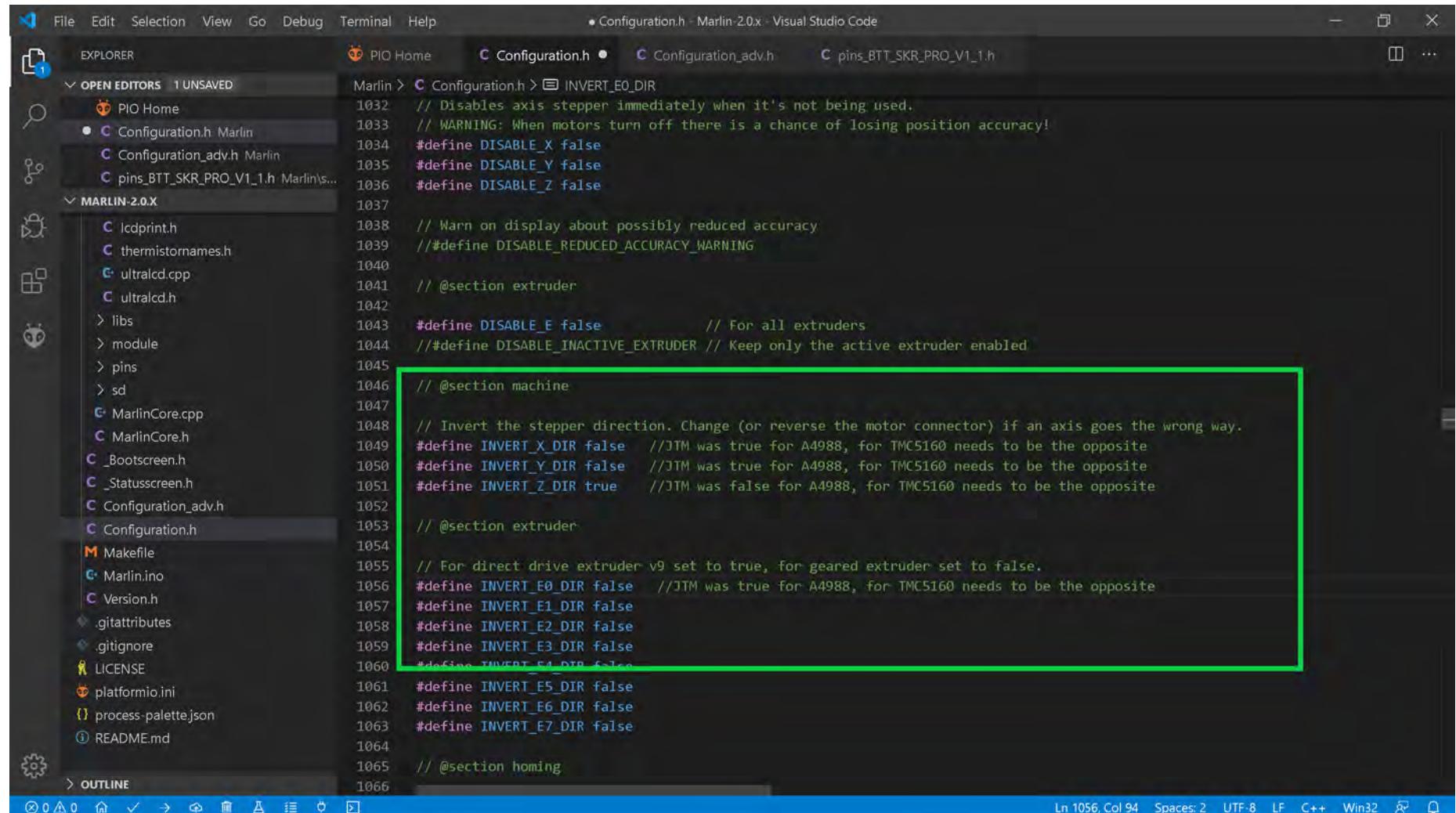
```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home
Configuration.h Marlin
Configuration_adv.h Marlin
pins_BTT_SKR_PRO_V1_1.h Marlin\src\MARLIN-2.0.X
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
libs
module
pins
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686, Col 80 Spaces: 2 UTF-8 LF C++ Win32
 661 /**
 662 * Stepper Drivers
 663 *
 664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 666 *
 667 *
 668 * A4988 is assumed for unspecified drivers.
 669 *
 670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 671 * TB6560, TB6600, TMC2100,
 672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 676 * [A4988], [A5984], [DRV8825], [LV8729], [L6470], [L6474], [POWERSTEP01], [TB6560], [TB6600], [TMC2100],
 677 * [TMC2130], [TMC2130_STANDALONE], [TMC2160], [TMC2160_STANDALONE],
 678 * [TMC2208], [TMC2208_STANDALONE], [TMC2209], [TMC2209_STANDALONE],
 679 * [TMC26X], [TMC26X_STANDALONE], [TMC2660], [TMC2660_STANDALONE],
 680 * [TMC5130], [TMC5130_STANDALONE], [TMC5160], [TMC5160_STANDALONE]
 681 */
#define X_DRIVER_TYPE TMC5160 //JTM was commented out having a value of A4988
#define Y_DRIVER_TYPE TMC5160 //JTM was commented out having a value of A4988
#define Z_DRIVER_TYPE TMC5160 //JTM was commented out having a value of A4988
#define E0_DRIVER_TYPE TMC5160 //JTM was commented out having a value of A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
// Enable this feature if all enabled endstop pins are interrupt-capable.
  
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC5160 drivers, I must invert the stepper motor direction because the TMC5160 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC5160 driver on was "true" change it to "false", as shown in the GREEN box below. If the setting was "false", now set it to "true", as shown in the GREEN box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following relevant section:

```

Marlin > C Configuration.h > INVERT_E0_DIR
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // JTM was true for A4988, for TMC5160 needs to be the opposite
1050 #define INVERT_Y_DIR false      // JTM was true for A4988, for TMC5160 needs to be the opposite
1051 #define INVERT_Z_DIR true       // JTM was false for A4988, for TMC5160 needs to be the opposite
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // JTM was true for A4988, for TMC5160 needs to be the opposite
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 // #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066

```

A green rectangular box highlights the following code block, which controls the inversion of stepper directions:

```

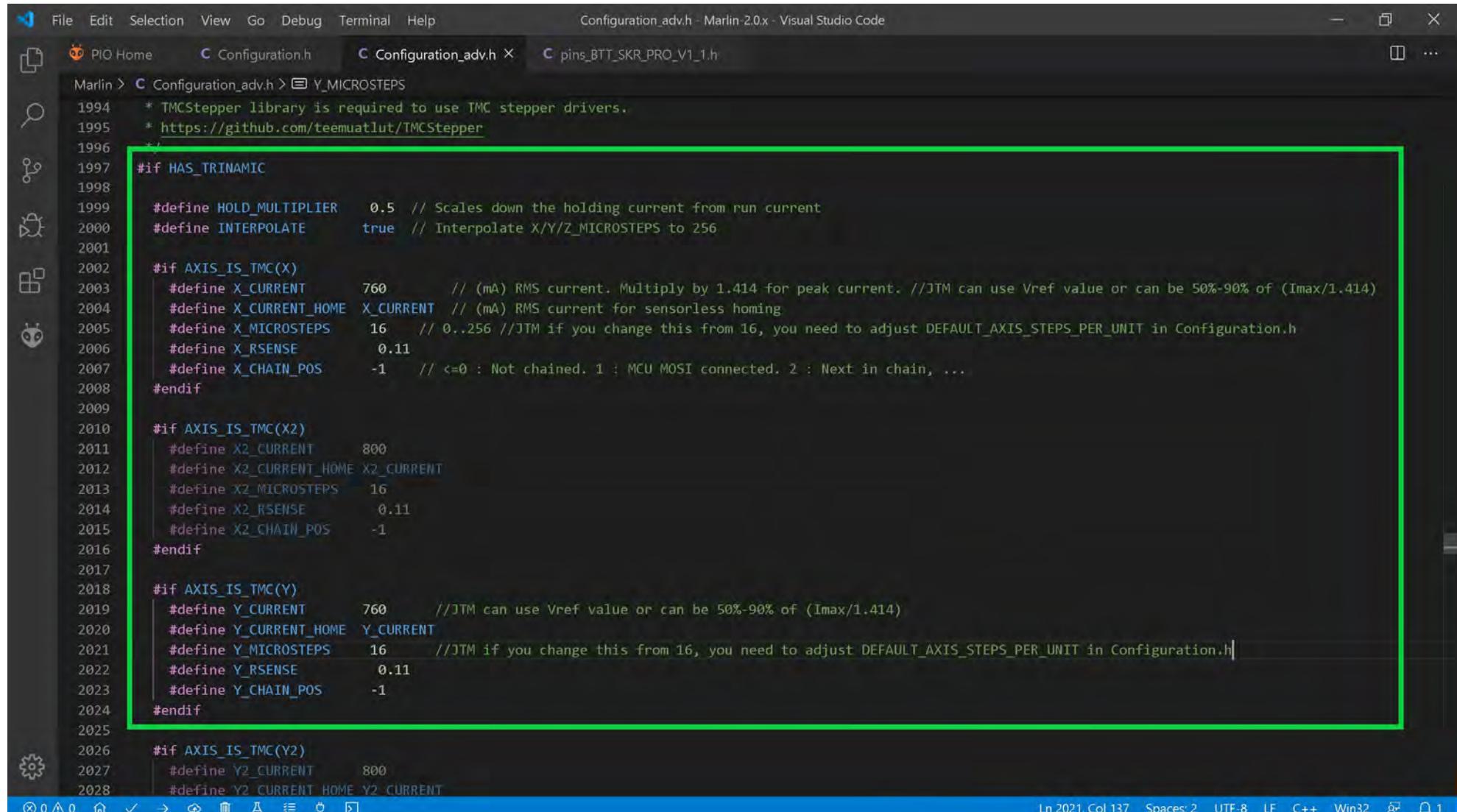
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false      // JTM was true for A4988, for TMC5160 needs to be the opposite
#define INVERT_Y_DIR false      // JTM was true for A4988, for TMC5160 needs to be the opposite
#define INVERT_Z_DIR true       // JTM was false for A4988, for TMC5160 needs to be the opposite

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- next you want to set your Vref in the Marlin firmware for each axis that has the TMC2130 driver, as seen in the GREEN box below. I changed the "X_CURRENT" to be the calculated Vref for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated Vref for my Y-Axis, which is 760mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
Marlin > Configuration_adv.h > Y_MICROSTEPS
1994 * TMCSstepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCSstepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999     #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000     #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002     #if AXIS_IS_TMC(X)
2003         #define X_CURRENT          760 // (mA) RMS current. Multiply by 1.414 for peak current. //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2004         #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005         #define X_MICROSTEPS       16 // 0..256 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2006         #define X_RSENSE            0.11
2007         #define X_CHAIN_POS         -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008     #endif
2009
2010     #if AXIS_IS_TMC(X2)
2011         #define X2_CURRENT         800
2012         #define X2_CURRENT_HOME   X2_CURRENT
2013         #define X2_MICROSTEPS      16
2014         #define X2_RSENSE           0.11
2015         #define X2_CHAIN_POS        -1
2016     #endif
2017
2018     #if AXIS_IS_TMC(Y)
2019         #define Y_CURRENT          760 //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2020         #define Y_CURRENT_HOME    Y_CURRENT
2021         #define Y_MICROSTEPS       16 //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2022         #define Y_RSENSE            0.11
2023         #define Y_CHAIN_POS         -1
2024     #endif
2025
2026     #if AXIS_IS_TMC(Y2)
2027         #define Y2_CURRENT         800
2028         #define Y2_CURRENT_HOME   Y2_CURRENT

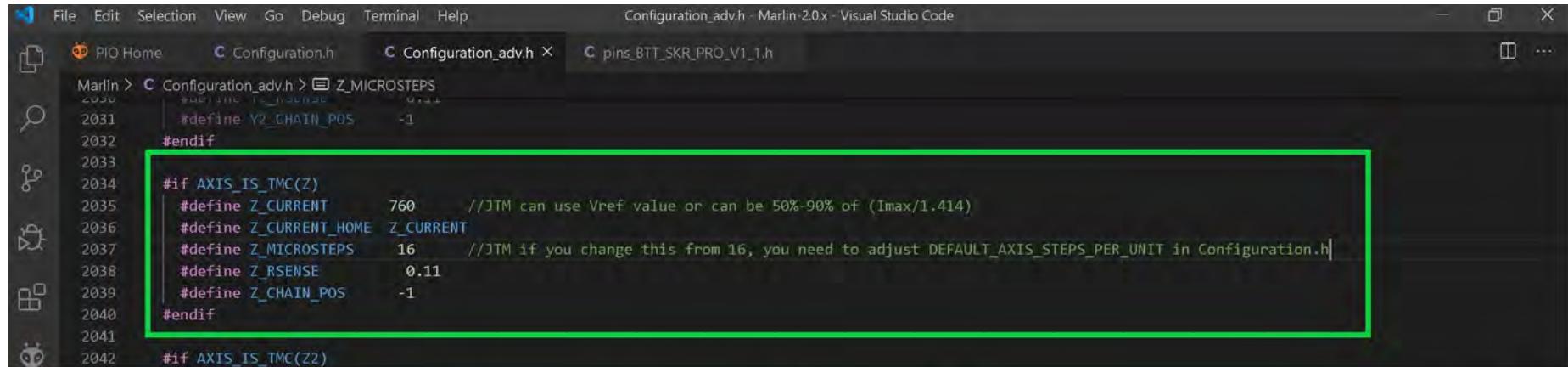
```

Ln 2021, Col 137 Spaces:2 UTF-8 LF C++ Win32  

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

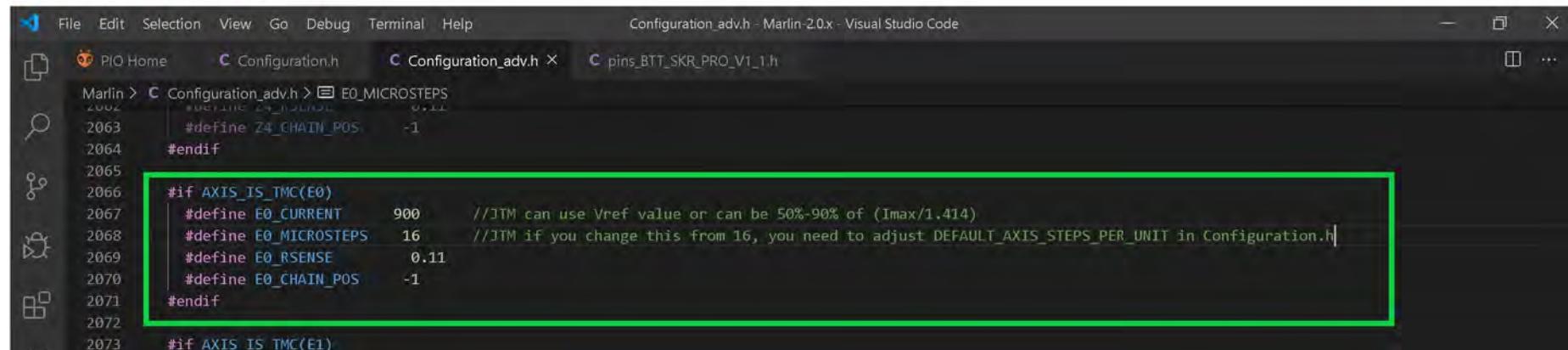
- Now, I am setting the Vref for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated Vref for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated Vref for my Extruder, which is 900mV on the Ender 3.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > Z_MICROSTEPS
2030
2031     #define Z2_CHAIN_POS -1
2032
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT    760      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2038         #define Z_RSENSE   0.11
2039         #define Z_CHAIN_POS -1
2040     #endif
2041
2042     #if AXIS_IS_TMC(Z2)

```

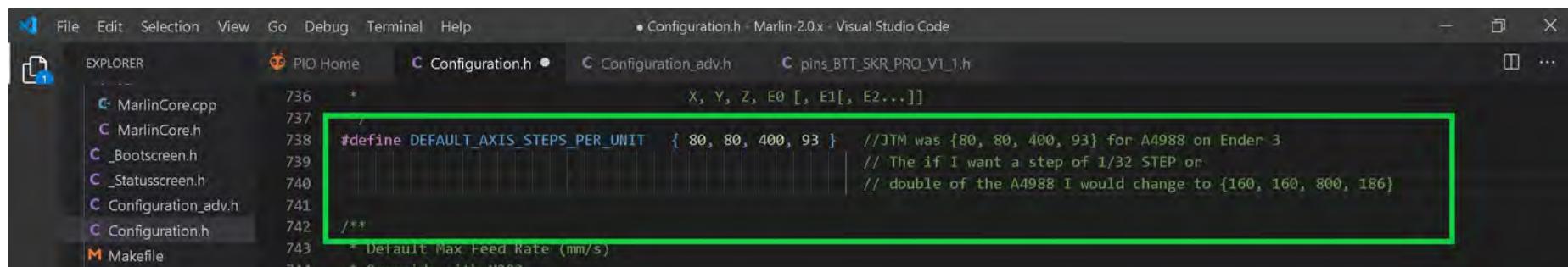


```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > E0_MICROSTEPS
2062
2063     #define Z4_CHAIN_POS -1
2064
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT    900      //JTM can use Vref value or can be 50%-90% of (Imax/1.414)
2068         #define E0_MICROSTEPS 16       //JTM if you change this from 16, you need to adjust DEFAULT_AXIS_STEPS_PER_UNIT in Configuration.h
2069         #define E0_RSENSE   0.11
2070         #define E0_CHAIN_POS -1
2071     #endif
2072
2073     #if AXIS_IS_TMC(E1)

```

- If you changed the "MICROSTEPS" for any of the Axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



```

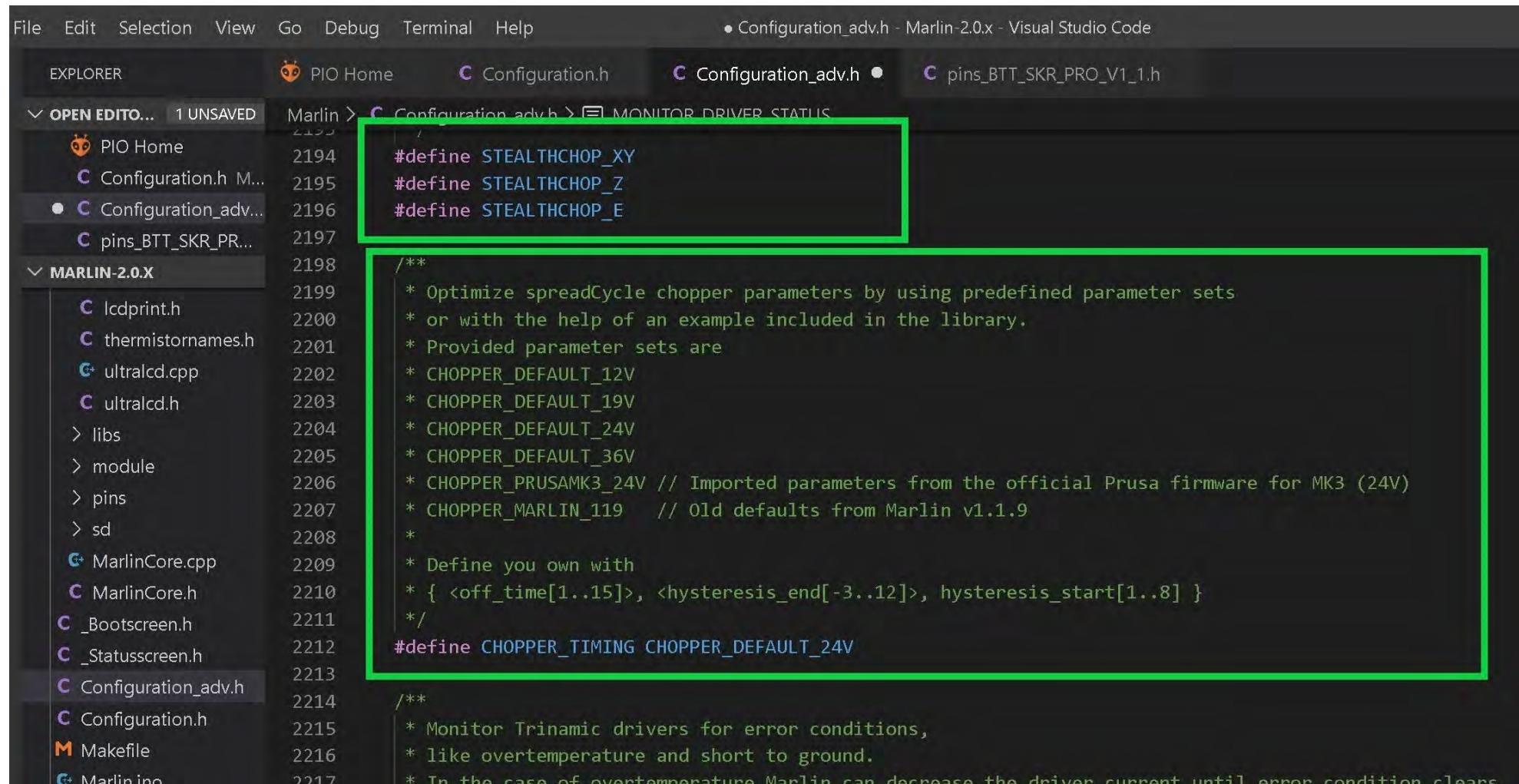
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
736     *
737     /
738     #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //JTM was {80, 80, 400, 93} for A4988 on Ender 3
739                                         // The if I want a step of 1/32 STEP or
740                                         // double of the A4988 I would change to {160, 160, 800, 186}
741
742     /**
743     * Default Max Feed Rate (mm/s)
744     * Override with M203

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if the proper "CHOPPER_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER_TIMING" is correct.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
- Explorer:**
 - OPEN EDITOR... 1 UNSAVED
 - PIO Home
 - Configuration.h M...
 - Configuration_adv...
 - C pins_BTT_SKR_PR...
- Editor:** The main editor window displays the `Configuration_adv.h` file. A green box highlights the following code block:

```
#define STEALTHCHOP_XY
#define STEALTHCHOP_Z
#define STEALTHCHOP_E
```
- Editor Content (Continued):**

```
/*
 * Optimize spreadCycle chopper parameters by using predefined parameter sets
 * or with the help of an example included in the library.
 * Provided parameter sets are
 * CHOPPER_DEFAULT_12V
 * CHOPPER_DEFAULT_19V
 * CHOPPER_DEFAULT_24V
 * CHOPPER_DEFAULT_36V
 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
 *
 * Define your own with
 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
 */
#define CHOPPER_TIMING CHOPPER_DEFAULT_24V
```
- Bottom Editor:**

```
/*
 * Monitor Trinamic drivers for error conditions,
 * like overtemperature and short to ground.
 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.

```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITOR... 1 UNSAVED Marlin > C Configuration_adv.h > HYBRID_THRESHOLD
PIO Home Configuration.h M...
C Configuration_adv...
C pins_BTT_SKR_PR...
MARLIN-2.0.X
C Lcdprint.h
C thermistornames.h
C ultralcd.cpp
C ultralcd.h
> libs
> module
> pins
> sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
Ln 2240, Col 52 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 1
File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h
OPEN EDITOR... 1 UNSAVED Marlin > C Configuration_adv.h > HYBRID_THRESHOLD
PIO Home Configuration.h M...
C Configuration_adv...
C pins_BTT_SKR_PR...
MARLIN-2.0.X
C Lcdprint.h
C thermistornames.h
C ultralcd.cpp
C ultralcd.h
> libs
> module
> pins
> sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
Ln 2240, Col 52 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 1


```

/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //JTM was commented out

#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

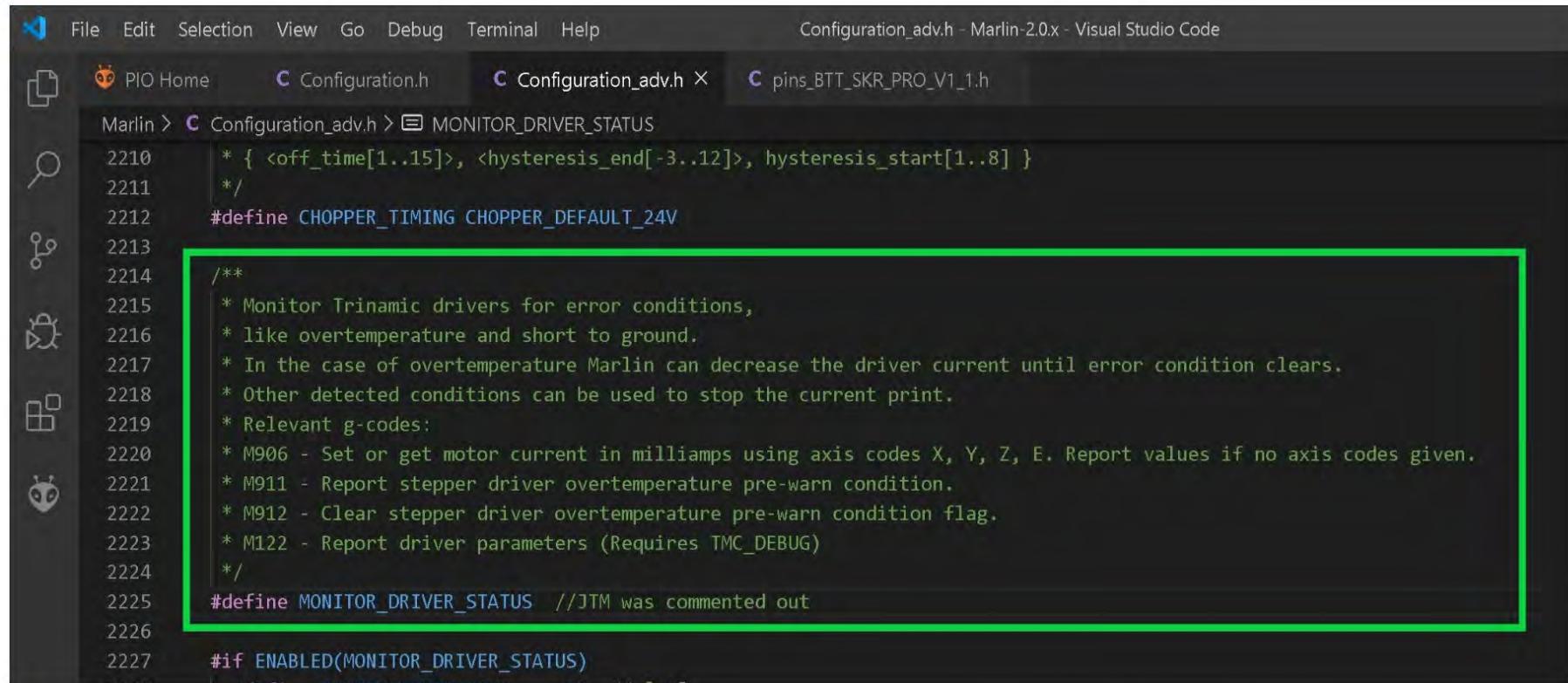
/*
 * Use StallGuard2 to home / probe X, Y, Z.
 *
 * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
 * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
 * X, Y, and Z homing will always be done in spreadCycle mode.

```


```

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in Marlin's documentation located [on-line](#).
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

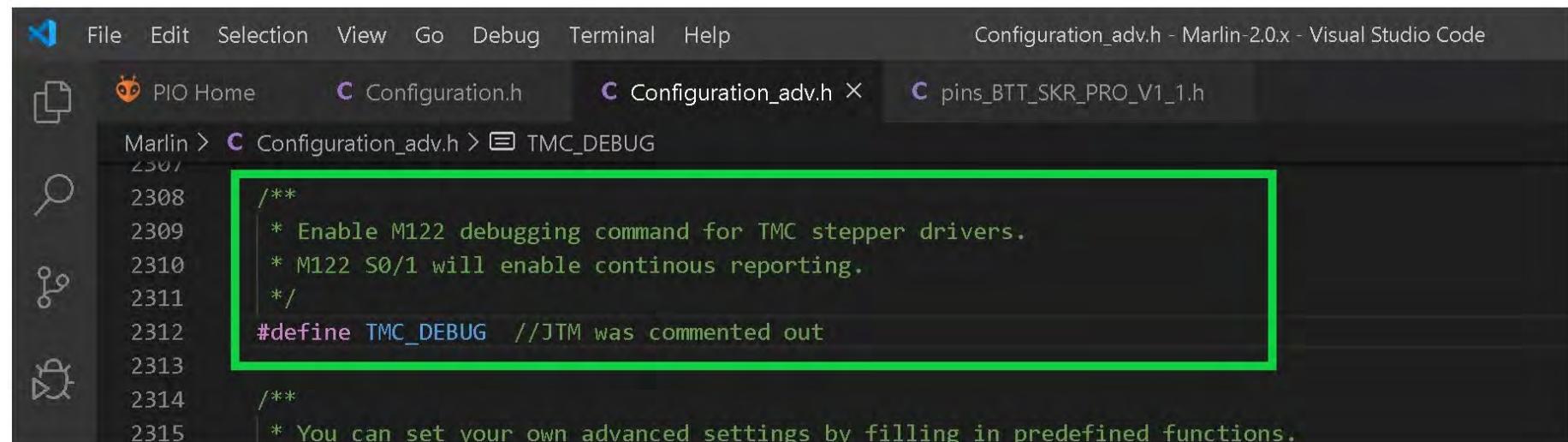
PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

2210     * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211     */
2212     #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //JTM was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

PIO Home Configuration.h Configuration_adv.h pins_BTT_SKR_PRO_V1_1.h

Marlin > Configuration_adv.h > TMC_DEBUG

```

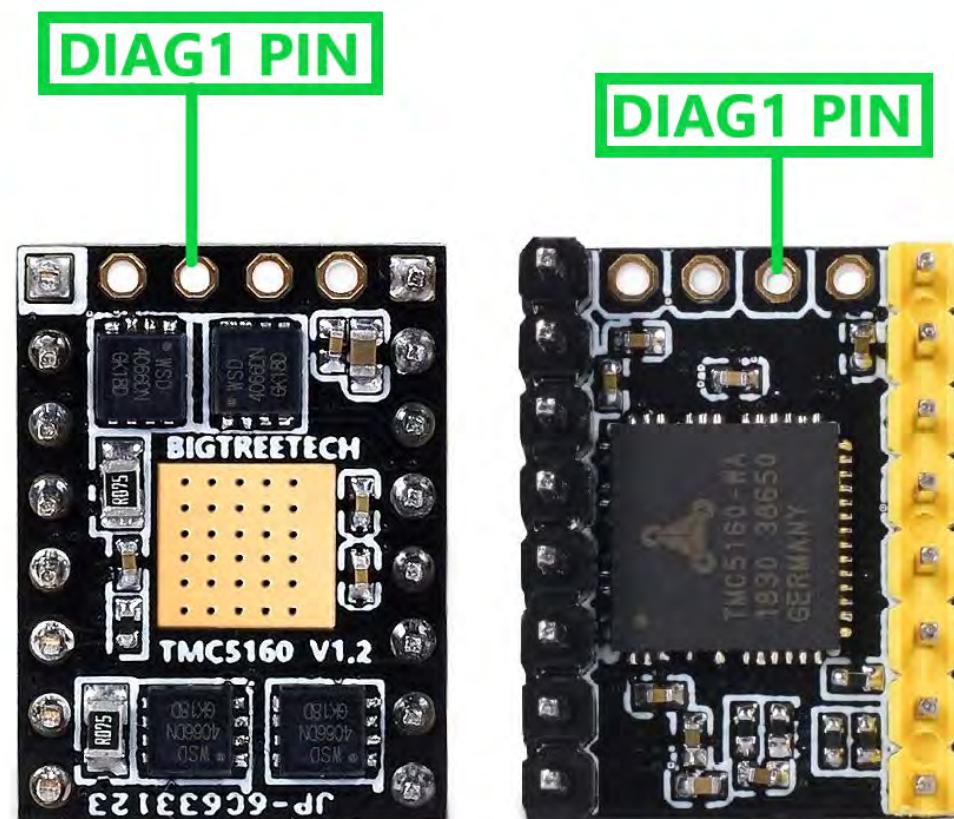
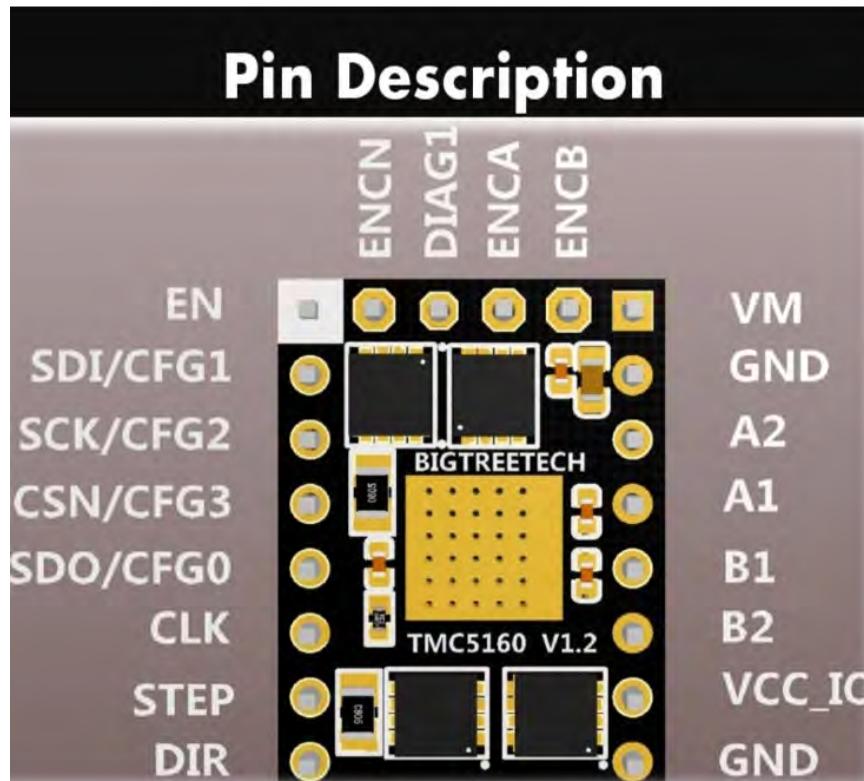
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //JTM was commented out
2313
2314 /**
2315 * You can set your own advanced settings by filling in predefined functions.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

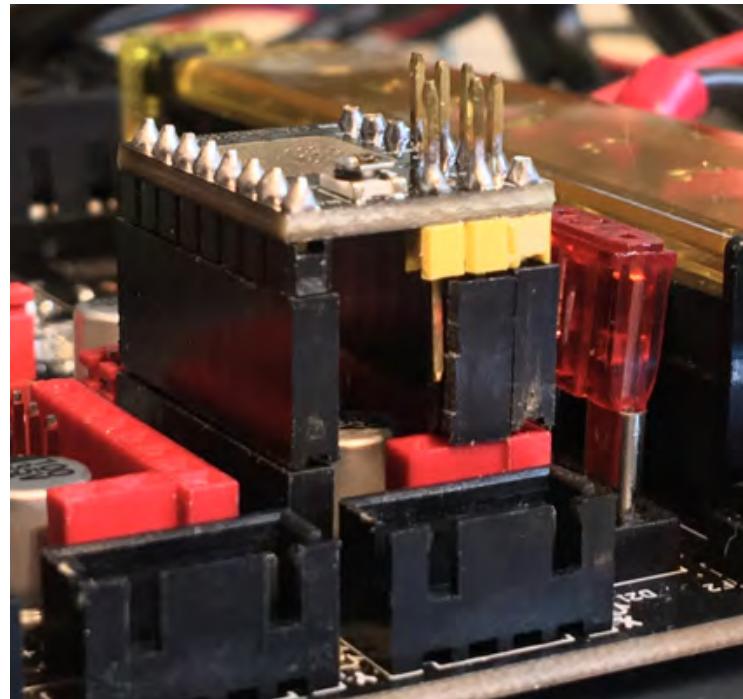
- This next section covers sensor-less homing which is available for the TMC5160 in SPI mode. I want to enable it BUT for the TMC5160 I first have to solder on the DIAG1 pin onto each TMC5160 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG1 pin for two TMC5160 drivers. Here is a picture of the TMC5160 V1.2 pin-out.



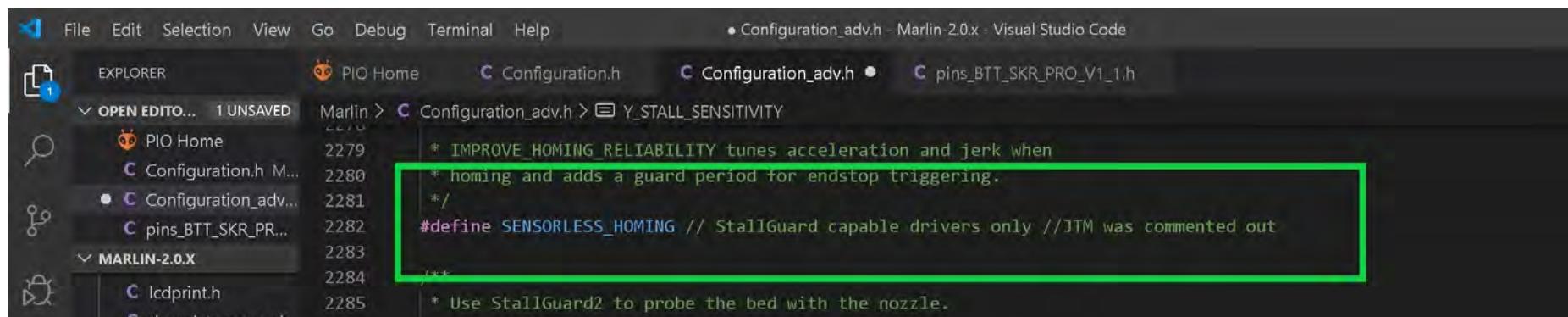
- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so that when I plug in the TMC5160 into the SKR PRO V1.1 board the DIAG1 pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC5160 driver has to be plugged into the SKR PRO V1.1 board. Since I am **not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC516 is NOT connected to the board.** I plan to plug my Z axis' TMC5160 by using long stackable header pin risers, as seen in the picture below.



- Sensor-less homing is commented out by default. So I remove the two leading "://" to un-comment "SENSORLESS_HOMING"



```

File Edit Selection View Go Debug Terminal Help
• Configuration_adv.h - Marlin-2.0.x : Visual Studio Code

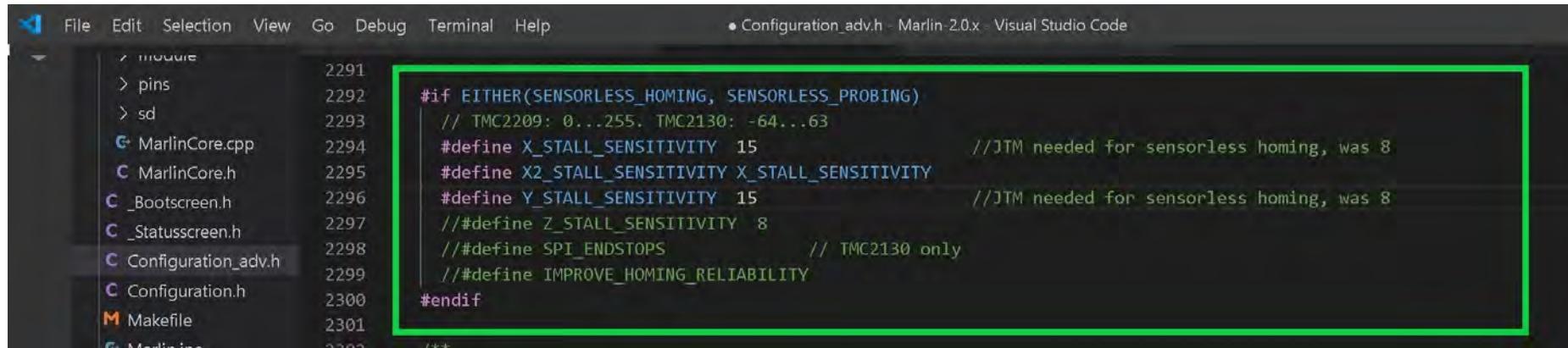
EXPLORER PIO Home Configuration.h Configuration_adv.h • pins_BTT_SKR_PRO_V1_1.h
✓ OPEN EDITOR... 1 UNSAVED Marlin > Configuration_adv.h > Y_STALL_SENSITIVITY
    PIO Home Configuration.h M...
    Configuration_adv.h • pins_BTT_SKR_PR...
    MARLIN-2.0.X Lcdprint.h thermistor.h
    Configuration_adv.h
    #define SENSORLESS_HOMING // StallGuard capable drivers only //JTM was commented out
    /**
    * Use StallGuard2 to probe the bed with the nozzle.
    */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Next we set a "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X_STALL_SENSITIVITY" and the "Y_STALL_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z_STALL_SENSITIVITY".



```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

// Configuration_adv.h
// Marlin Configuration File
// http://www.reprap.org

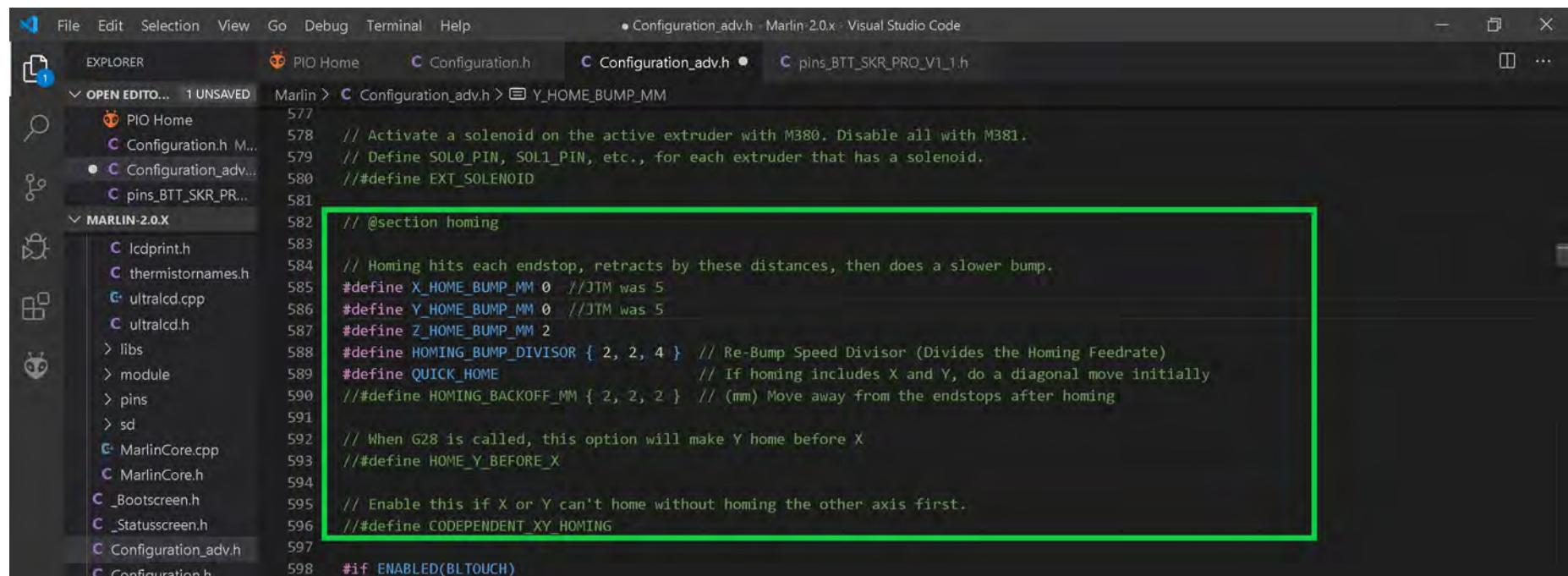
// This file contains all the configuration options.
// For a general overview, please refer to Configuration_adv.h.in

// When enabled, the printer will attempt to homing using sensorless homing
// If the homing fails, it will fall back to normal homing
// #define USE_SENSORLESS_HOMING

// If either Sensorless Homing or Sensorless Probing is enabled, then
// the following defines are used
// #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
//   #define X_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//   #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
//   #define Y_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//   // #define Z_STALL_SENSITIVITY 8
//   // #define SPI_ENDSTOPS // TMC2130 only
//   // #define IMPROVE_HOMING_RELIABILITY
// #endif

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.



```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

// Configuration_adv.h
// Marlin Configuration File
// http://www.reprap.org

// This file contains all the configuration options.
// For a general overview, please refer to Configuration_adv.h.in

// When enabled, the printer will attempt to homing using sensorless homing
// If the homing fails, it will fall back to normal homing
// #define USE_SENSORLESS_HOMING

// If either Sensorless Homing or Sensorless Probing is enabled, then
// the following defines are used
// #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
//   #define X_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//   #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
//   #define Y_STALL_SENSITIVITY 15 //JTM needed for sensorless homing, was 8
//   // #define Z_STALL_SENSITIVITY 8
//   // #define SPI_ENDSTOPS // TMC2130 only
//   // #define IMPROVE_HOMING_RELIABILITY
// #endif

// Home bump settings
// Define the home bump distance for each axis
// #define X_HOME_BUMP_MM 0 //JTM was 5
// #define Y_HOME_BUMP_MM 0 //JTM was 5
// #define Z_HOME_BUMP_MM 2

// Define the homing bump divisor
// #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
// Define the quick home move
// #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
// Define the homing backoff distance
// #define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing

// When G28 is called, this option will make Y home before X
// #define HOME_Y_BEFORE_X

// Enable this if X or Y can't home without homing the other axis first.
// #define CODEPENDENT_XY_HOMING

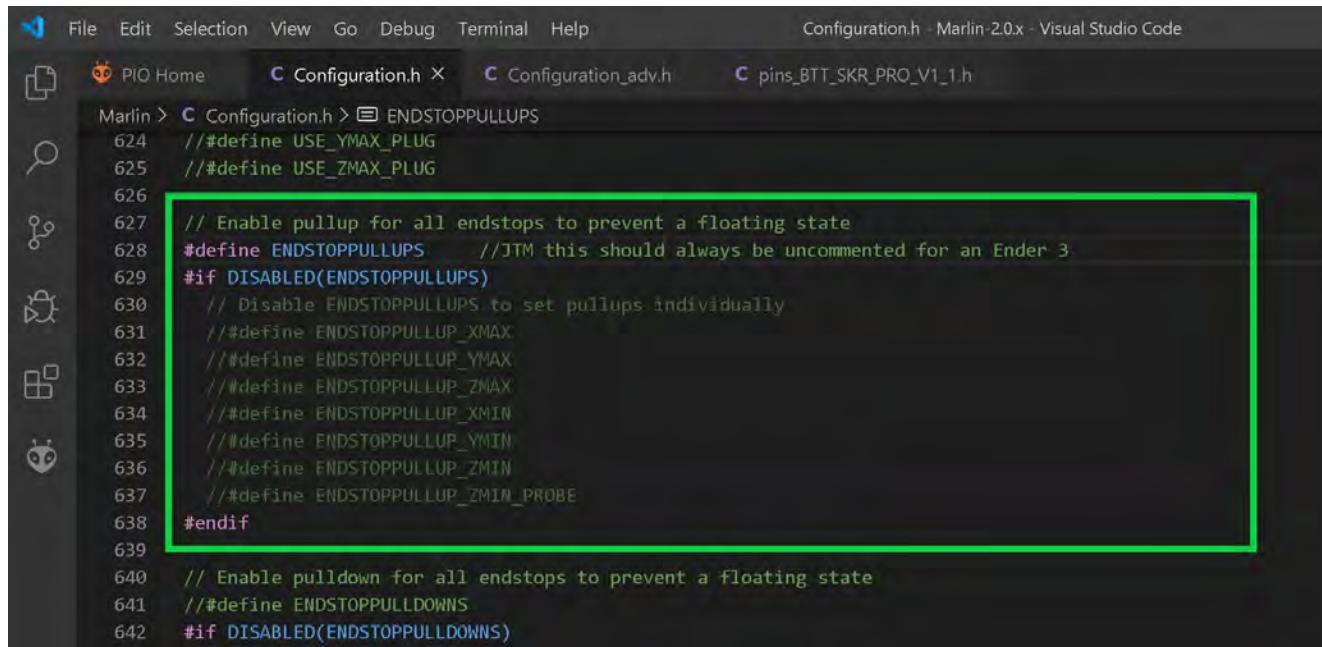
// If BLTOUCH is enabled, then enable it here
// #if ENABLED(BLTOUCH)

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default.



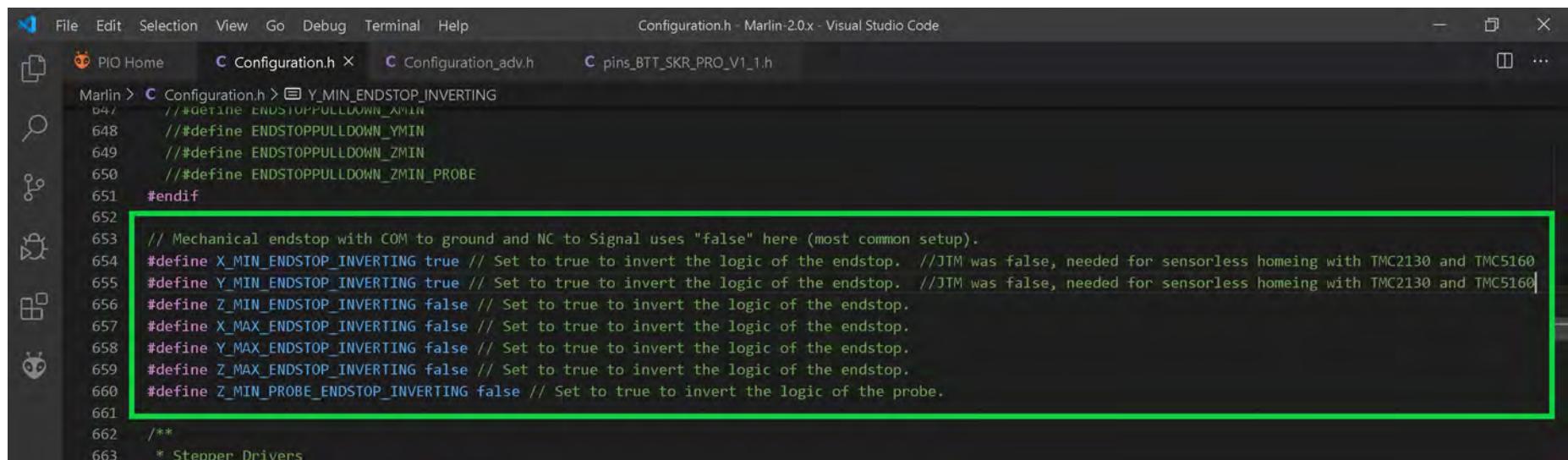
```

File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code

Marlin > C Configuration.h > ENDSTOPPULLUPS
624 // #define USE_YMAX_PLUG
625 // #define USE_ZMAX_PLUG
626
627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS // JTM this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 // #define ENDSTOPPULLUP_XMAX
632 // #define ENDSTOPPULLUP_YMAX
633 // #define ENDSTOPPULLUP_ZMAX
634 // #define ENDSTOPPULLUP_XMIN
635 // #define ENDSTOPPULLUP_YMIN
636 // #define ENDSTOPPULLUP_ZMIN
637 // #define ENDSTOPPULLUP_ZMIN_PROBE
638#endif
639
640 // Enable pulldown for all endstops to prevent a floating state
641 // #define ENDSTOPPULLDOWNS
642 #if DISABLED(ENDSTOPPULLDOWNS)

```

- Next to allow sensor-less homing to work we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENDSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z_MIN_ENDSTOP_INVERTING" set to false.



```

File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code

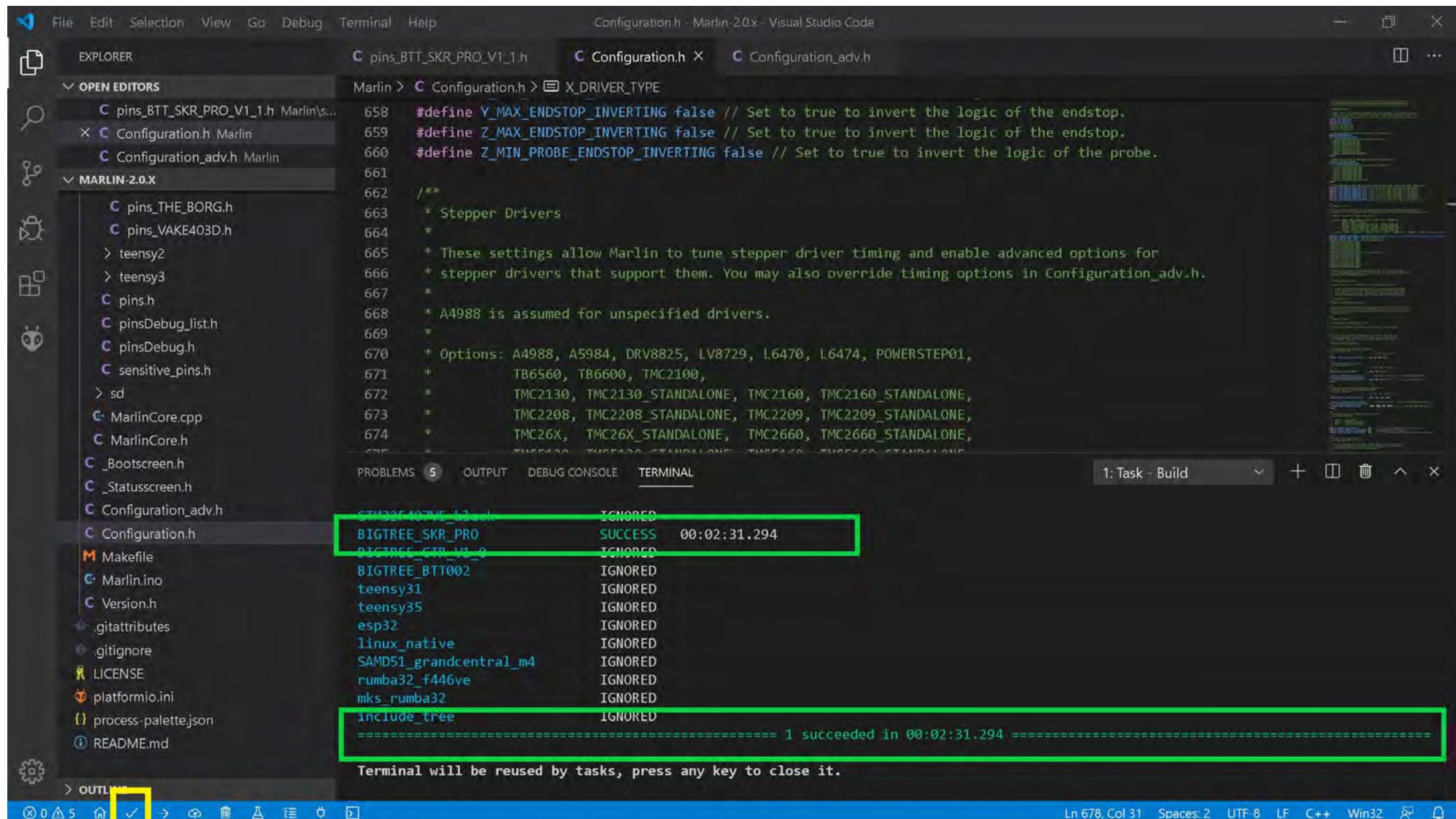
Marlin > C Configuration.h > Y_MIN_ENDSTOP_INVERTING
647 // #define ENDSTOPPULLDOWN_XMIN
648 // #define ENDSTOPPULLDOWN_YMIN
649 // #define ENDSTOPPULLDOWN_ZMIN
650 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
651#endif
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654#define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop. // JTM was false, needed for sensorless homeing with TMC2130 and TMC5160
655#define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop. // JTM was false, needed for sensorless homeing with TMC2130 and TMC5160
656#define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**
663 * Stepper Drivers

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- The end of Marlin setup for BIQU TMC5160 drivers in SPI mode . Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



```

Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins\pins_BTT_SKR_PRO_V1_1.h
Configuration.h Marlin\src\configuration\configuration.h
Configuration_adv.h Marlin\src\configuration\configuration_adv.h
MARLIN-2.0.X
pins_THE_BORG.h
pins_VAKE403D.h
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
Bootscreen.h
Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
[ 100% ] 1 succeeded in 00:02:31.294
Terminal will be reused by tasks, press any key to close it.

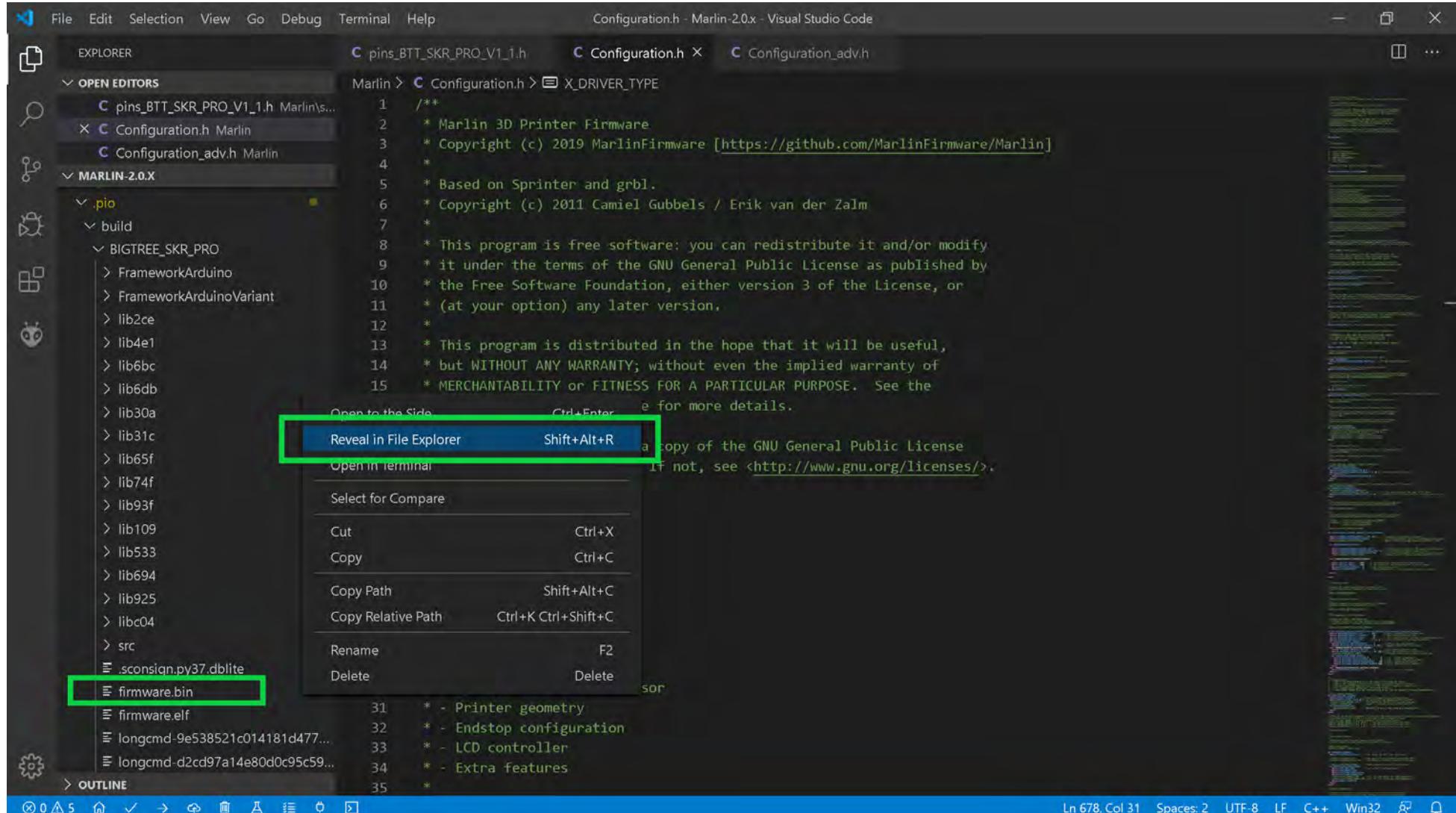
```

The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The terminal tab is active, displaying the build process. A yellow box highlights the terminal output, and a green box highlights the success message at the end of the build log.

- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro-SD card then place the micro-SD card into the micro-SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin" file, and learn how to transfer it to your micro-SD card go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

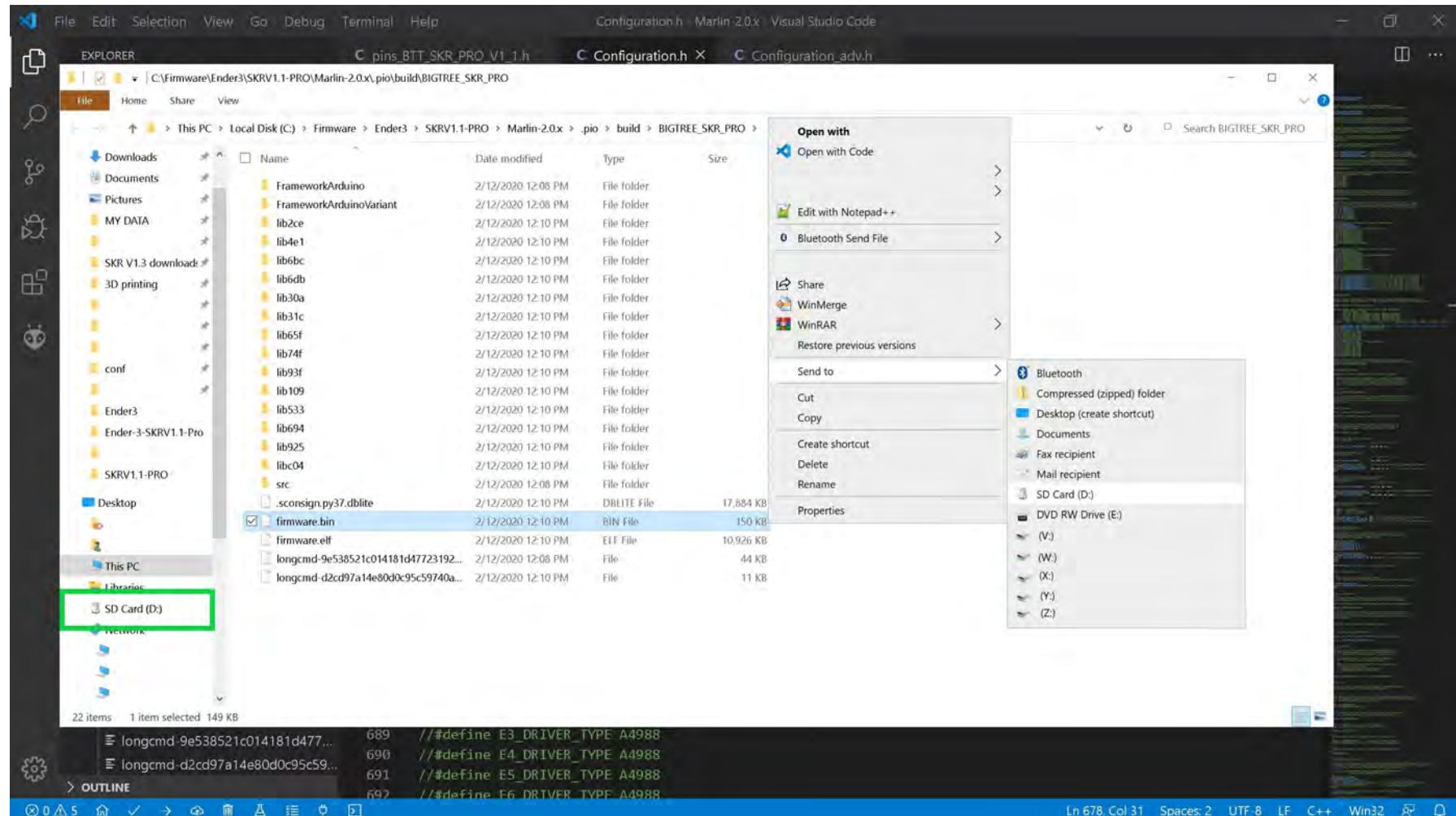
- the firmware.bin file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Ensure your micro-SD card that came with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
 - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro-SD card out of the adapter and place it into the micro-SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the "FIRMWARE.CUR" file will be uploaded to the board and renamed to "firmware.bin" on the micro-SD card.

How to adjust the Vref on a Stepper Motor Driver board using the Potentiometer^{1,2}

Modern 3D printers usually use NEMA17 motors. The first piece of information you will need is the "Rated Current" of your NEMA17 motor. So, write down your motor's part number and pay attention to the Step Angle, Holding Torque, Rated Current, Voltage, and Inductance.

Use the Step Angle to work out your printer's "DEFAULT_AXIS_STEPS_PER_UNIT" for your firmware with:

1. <http://www.prusaprinters.org/calculator/>

How to Tune Stepper Motor Drivers²

1. Turn power off your printer, unplug the stepper motor cables, turn power back on your printer and tune the stepper motor drivers that are already plugged into the SKR PRO V1.1 board.
2. When done, turn power off the printer, plug in the stepper motor cables, turn power back on your printer and test motor movement.

NOTE: Don't tune stepper motor drivers with the motors plugged in, if you accidentally set current too high you can fry the motor or the stepper motor driver.

NOTE: Don't plug or unplug stepper motors with the power on the printer (i.e. power on the SKR PRO V1.1 board)

Measure DC voltage between the stepper motor driver's trimpot (POT) or "Vref Test point" and your PSU's (12VDC/24VDC) ground. The ground at the PSU connector to the SKR PRO board is fine to use. Look up the correct current for your motor part number. If you have motors with no part number, assume they have a max of 1.00 amps (Imax) to be safe. Look up the proper formula for your stepper motor drivers (as show in this document), and find the voltage (but ONLY use 90% of the calculated Vref) which corresponds with the current you want to set. Use a ceramic screw driver to adjust the POT. A ceramic screw driver is non conductive and if you slip while making the adjustment to the POT you could short circuit the stepper motor driver board (i.e. KILL the driver board) in the process.

Time Saver tip, but more dangerous: Get slip-on alligator clips for your multimeter. Clamp ground to a 12VDC/24VDC ground (PSU Ground) wire and clamp positive to your plastic handle screwdriver. This way you'll measure the voltage as you adjust ("live adjustment") and don't need three hands.

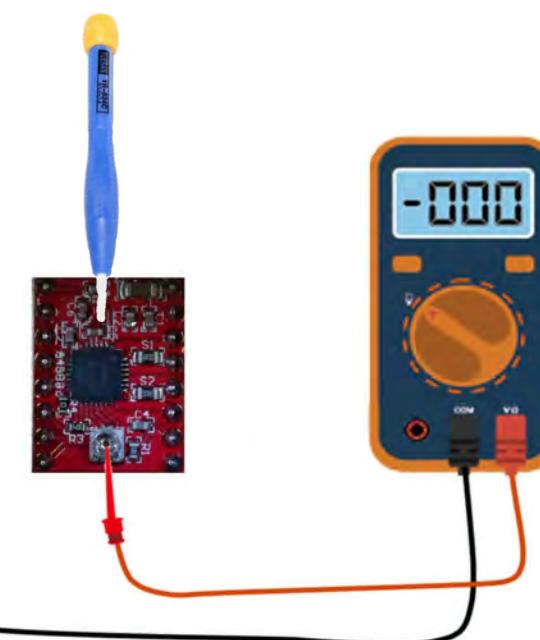
Note: See the next page for a diagram of the setup.

¹ from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers> and

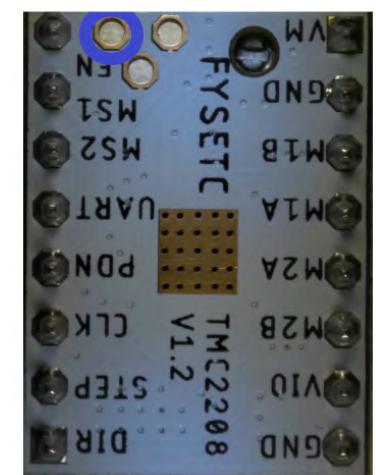
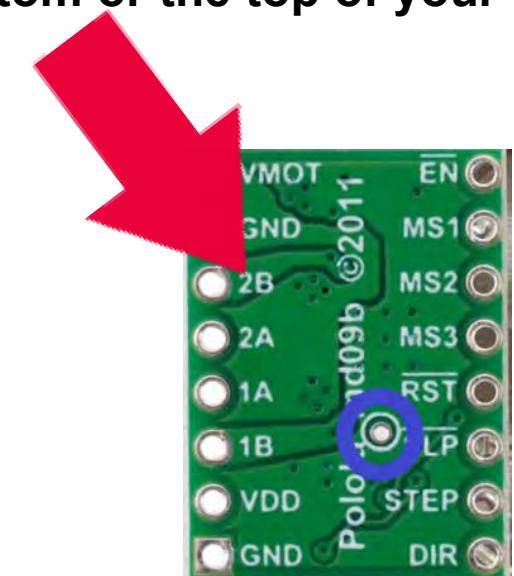
² from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

How to adjust the Vref on a Stepper Motor Driver board using the Potentiometer^{1,2}

Note: A ceramic screw driver is non conductive. If you use a plastic handle screw driver with alligator clips to your multimeter to make a "live adjustment" of Vref (and you slip while making the adjustment to the POT), you could short circuit the stepper motor driver board (i.e. KILL the board) in the process.



Note: Some stepper motor driver boards have a "Vref Test point" location, shown in **BLUE**. Check the bottom or the top of your board for a location.



¹ from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers>

² from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

APPENDIX B

For the TMC drivers what's the difference between Stand alone Mode and ("UART" or "SPI ") mode?

All the TMC driver chips **EXCEPT TMC5160** have a stand alone mode. **Stand Alone mode** allows you to just drop the driver into your motherboard to replace your A4988 driver. The **OTP mode** is similar, but you use some software to **PERMANENTLY** change the driver's mode of operation. The **UART or SPI modes** allow you to **dynamically** change the driver in your firmware.

How To Calculate Vref For Stepper Motor Drivers

My machine is an Ender 3, the X,Y,Z stepper motor "Rated Current" is 0.84 Amps, while E (extruder) stepper motor "Rated Current" is 1 Amps.

We use the Vref formula (found on the first page of each different stepper motor driver section of this document) and do the Vref calculation.

Remember this Vref calculation is just a suggested starting point. If your stepper motors are running **too hot** you will need to adjust the Vref **downward**. If your stepper motors are **skipping steps** when printing then you will need to adjust your Vref **upwards**. Our goal is to find a low enough Vref where our stepper motors are cool enough with out the printer missing any steps. For this example I will use the TMC2208 stepper motor drivers.

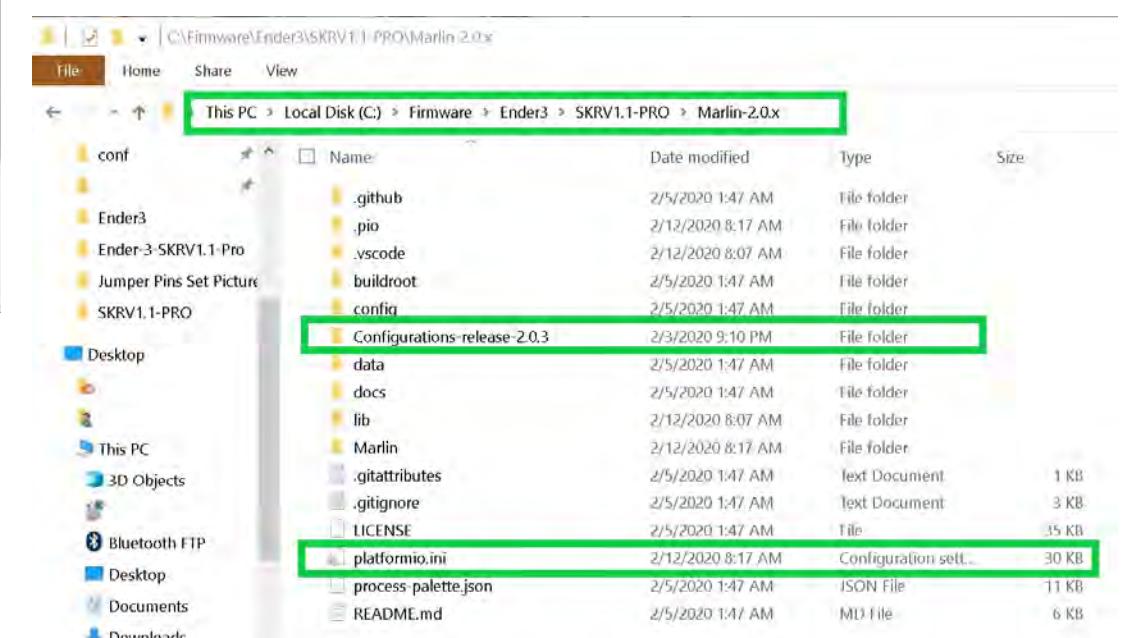
1. So, TMC2208 Vref formula is $Vref=((I_{max}/1.414)*2.5)/1.77$ OR $Vref=("Rated\ Current"/1.414)*2.5)/1.77$.
2. I take each of my Axis' "Rated Current" and plug it into that equation to get X-Axis Vref is equal to $((0.84/1.414)*2.5)/1.77)=0.8391$ volts.
3. Now, take 90% of that for a starting point Vref value of $((0.8391*0.90)=0.75519)$ 0.755 volts or 755mV or X-Axis Vref = 0.755V. Since X,Y and Z stepper motors have the same "Rated Current" we now have the Vref for X, Y and Z stepper motor drivers. Their value is 0.755 volts.
4. For E (extruder) Vref the equation is $((1.0/1.414)*2.5/1.77))=0.9988$ volts. Now, take 90% of that for a starting point Vref value of $((0.9988*0.90)=0.8990)$ 0.900 volts or 900mV. We now have the Vref for E (extruder stepper motor driver) which is 0.900 volts.
5. We use the our multimeter and turn the POT on the top of the stepper motor driver until we see the wanted Vref voltage displayed.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

Please follow this guide to install Visual Studio Code with Platformio onto your computer. The link to the instruction are located at: https://marlinfw.org/docs/basics/install_platformio.html

This example will use the Creality Ender 3 printer. Select the appropriate default configuration files for your specific printer!

- Download the **latest release of Marlin** from here: <https://marlinfw.org/meta/download/>
- Unzip the latest release of Marlin onto your hard drive
- Also, download the latest release of the Marlin Configuration files and then unzip the Marlin Configuration files so they reside in the same subdirectory as the Marlin files, see the pictures below for how I organize my file structure for Marlin



The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

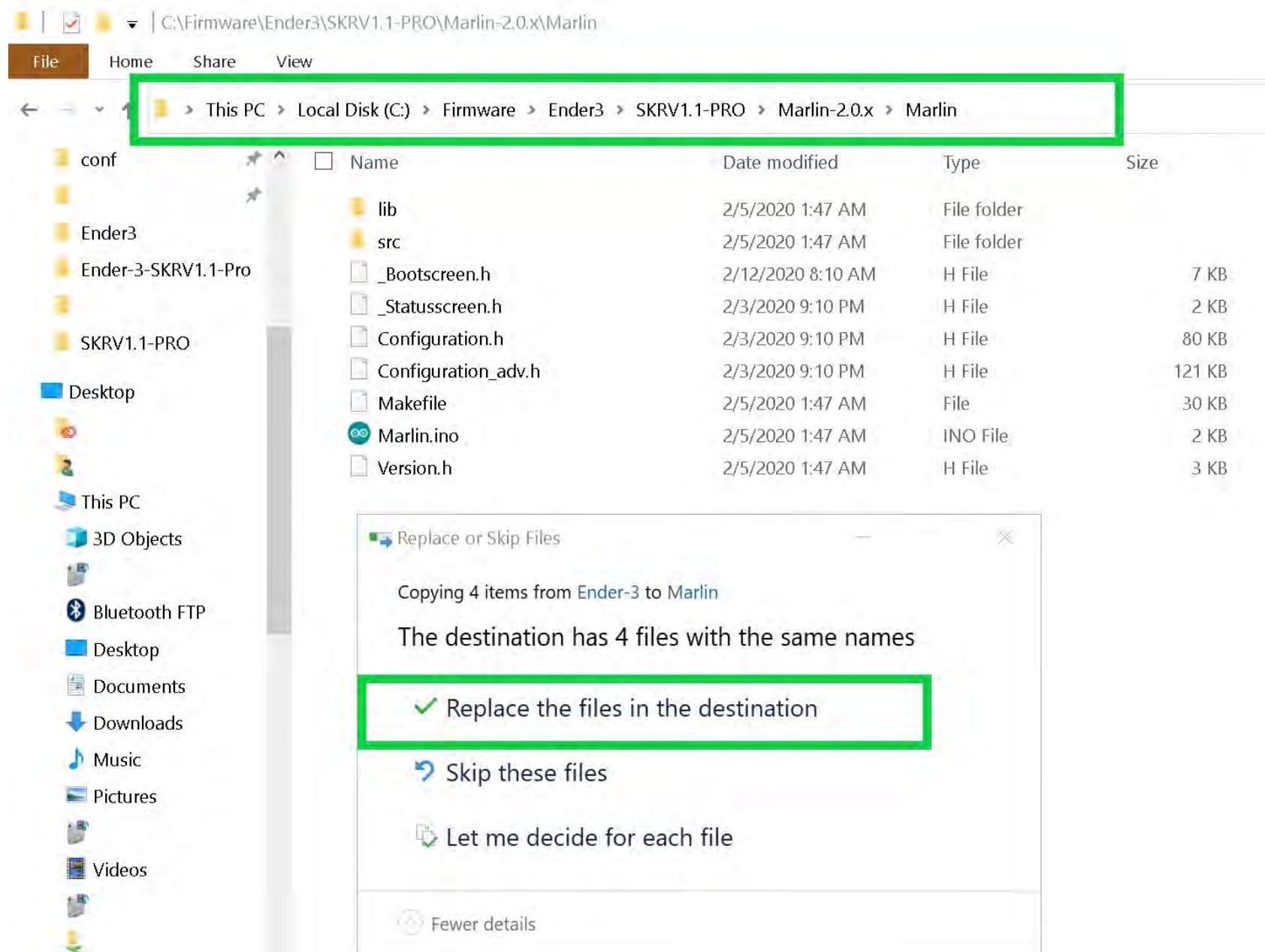
NOTE: This example will use the Creality Ender 3 printer and using the latest release of Marlin firmware which is version 2.0.3. Select the appropriate default configuration files for your specific printer!

- Copy the below default configuration files (for me Ender 3 files, for you it could be another printer) to the directory where Marlin's Configuration.h and Configuration_adv.h reside. See picture below for which files you will copy. See the next page to see where to place the files.



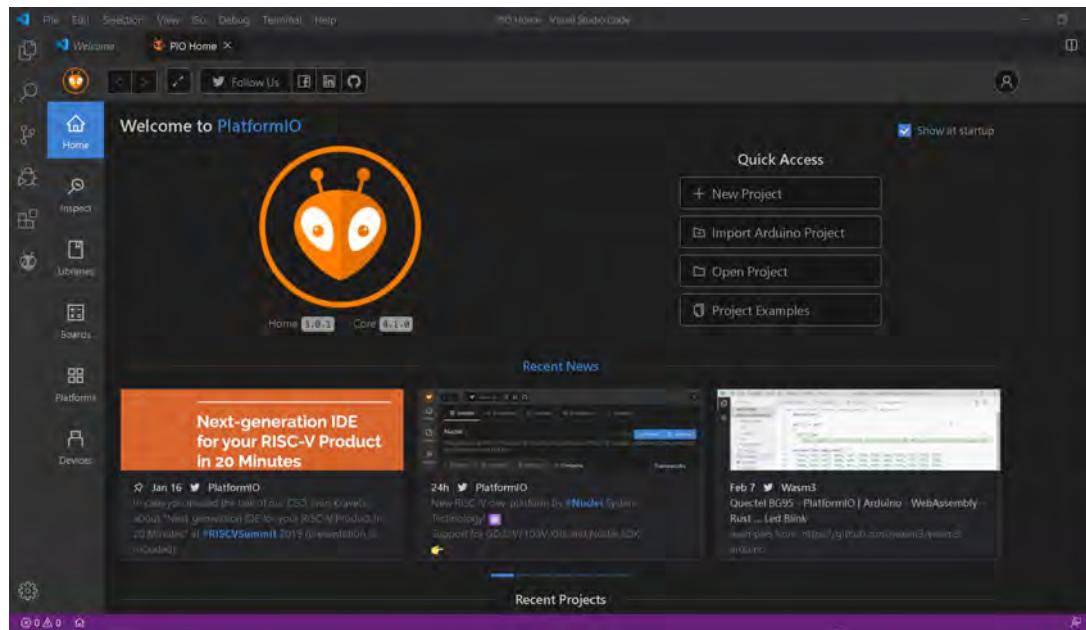
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Copy TO THIS directory so the above files reside in the same directory as Configuration.h and Configuration_adv.h. When prompted allow the overwrite of the files!

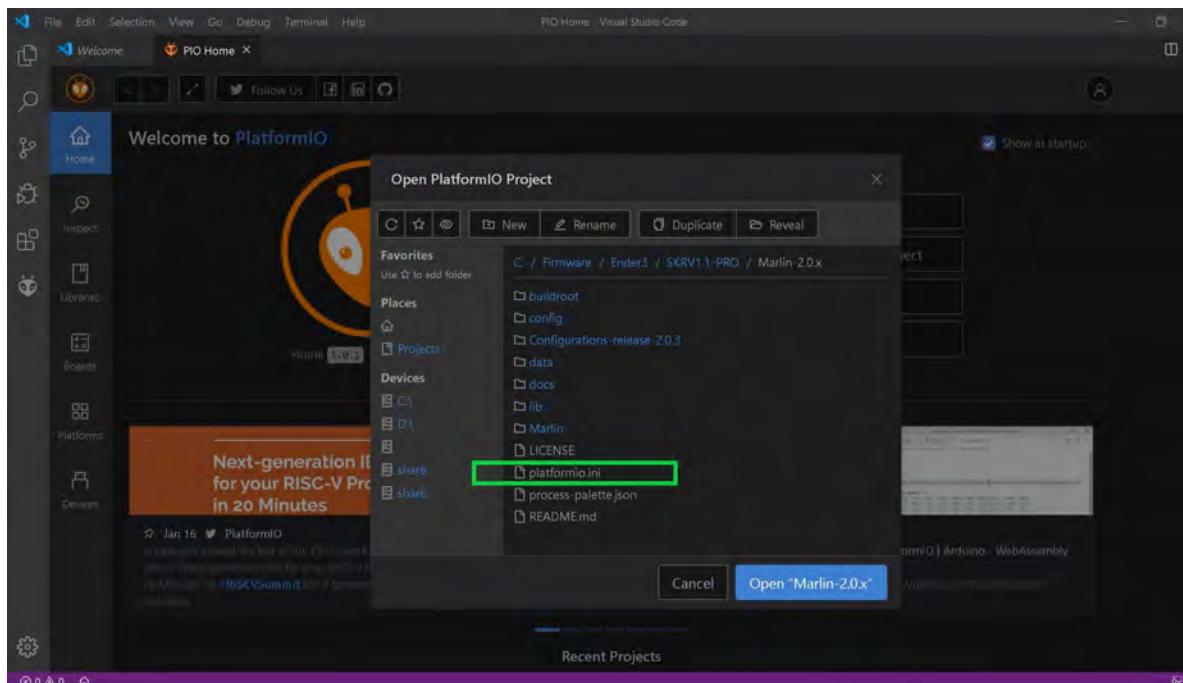


The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Open VS code (see picture below) and then select "Open Project".

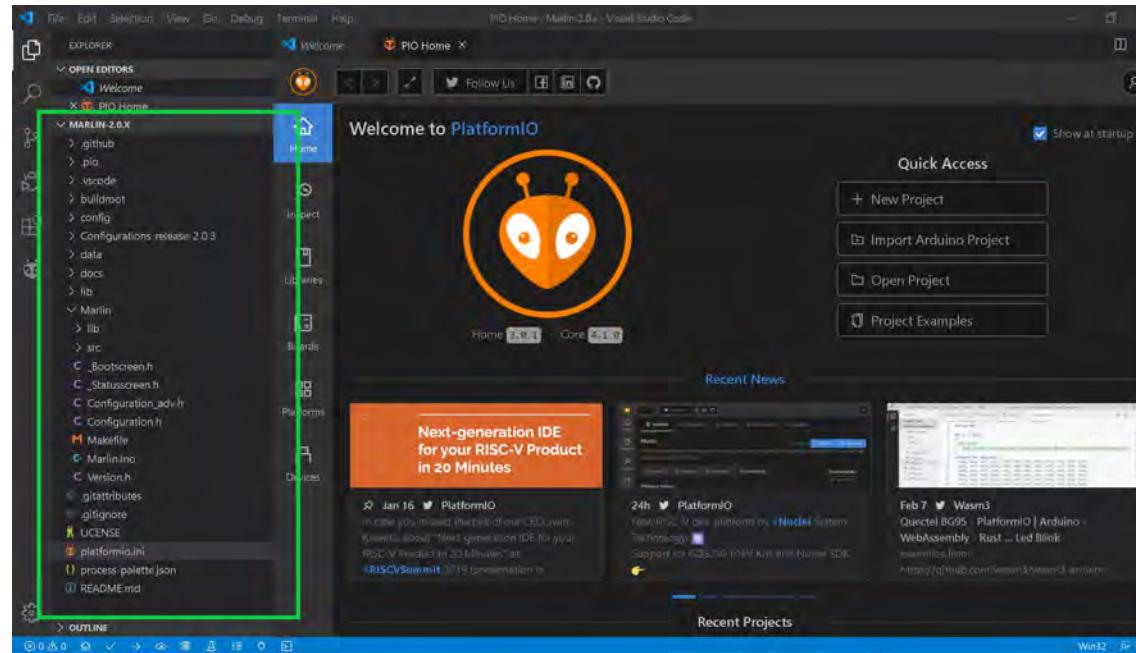


- Goto the directory where the platformio.ini file resides and open that folder (see picture below)

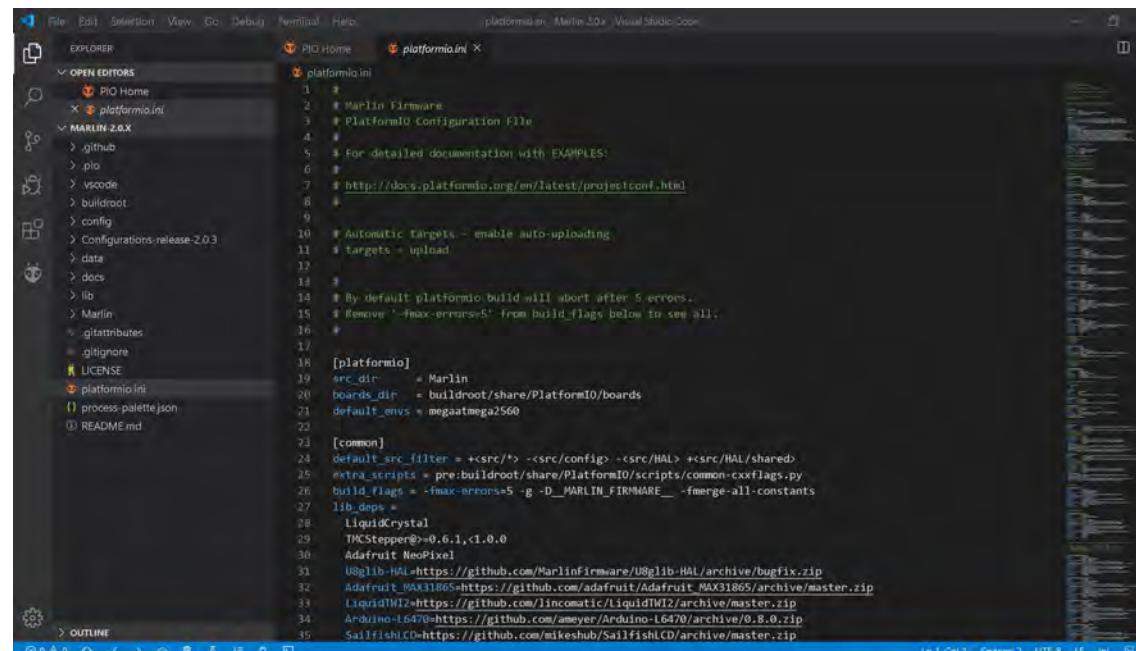


The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



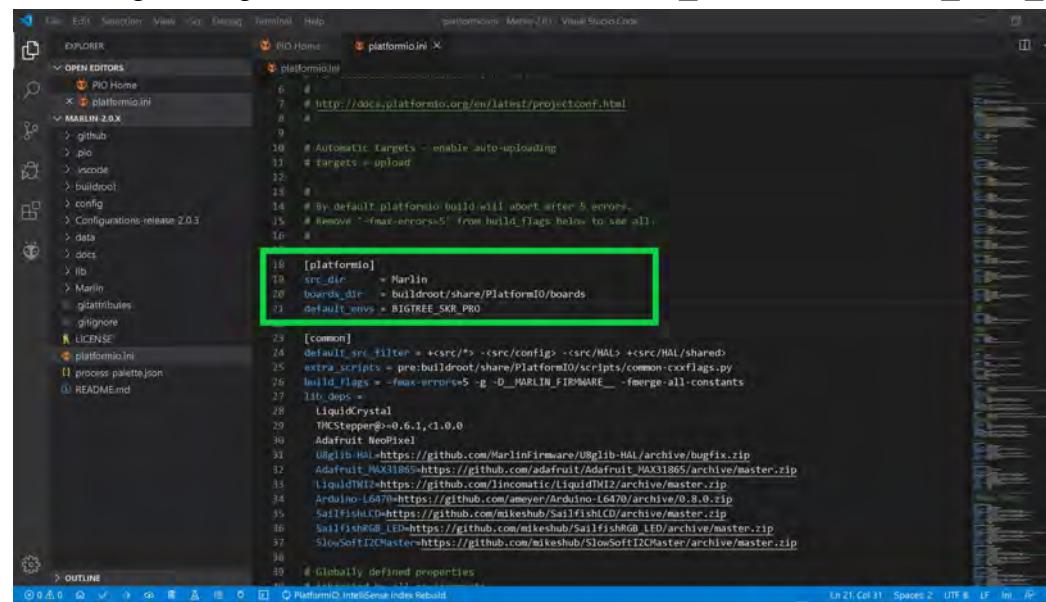
- What you will see when "platformio.ini" is opened up in the VS code editor window



NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Edit the "default_envs = megaatmega2560" line and make "default_envs = BIGTREE_SKR_PRO", as shown below in GREEN



```

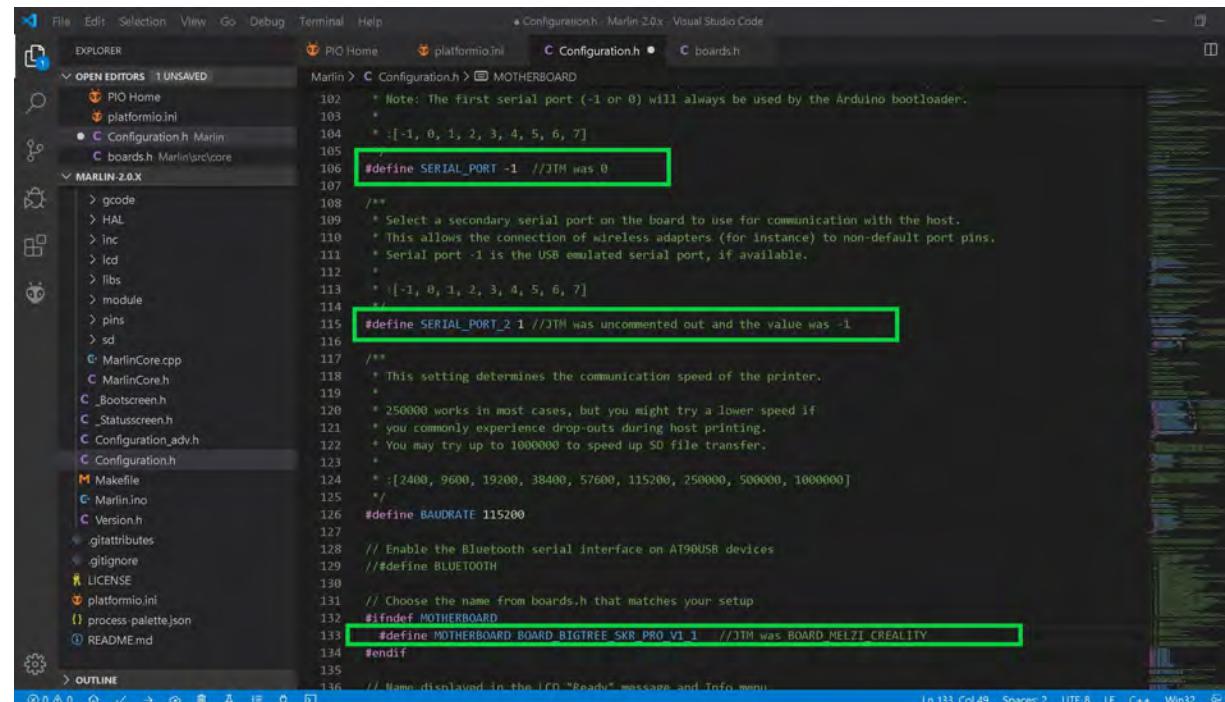
platformio.ini - Marlin 2.0.0 - Visual Studio Code

[platformio]
src_dir = Marlin
boards_dir = buildroot/share/PlatformIO/boards
default_envs = BIGTREE_SKR_PRO

[common]
default_src_filter = +/* -<src/config> -<src/HAL> +<src/HAL/shared>
extra_scripts = pre/buildroot/share/PlatformIO/scripts/common-cxxflags.py
build_flags = -fmax-errors=5 -g -D__MARLIN_FIRMWARE__ -fmerge-all-constants
lib_deps =
  LiquidCrystal
  TMCStepper@~0.6.1,<1.0.0
  Adafruit_NeoPixel
  U8glib-HAL-https://github.com/MarlinFirmware/U8glib-HAL/archive/bugfix.zip
  U8glib-MX31865-https://github.com/adafruit/U8glib-MX31865/archive/master.zip
  LiquidI2C-https://github.com/jlcomatic/LiquidI2C/archive/master.zip
  Arduino-L6470-https://github.com/ameyer/Arduino-L6470/archive/0.8.0.zip
  SailfishLCD-https://github.com/mikeshub/SailfishLCD/archive/master.zip
  SailfishRGB_LED-https://github.com/mikeshub/SailfishRGB_LED/archive/master.zip
  SlowSoftI2CMaster-https://github.com/mikeshub/SlowSoftI2CMaster/archive/master.zip
# Globally defined properties

```

- Go to the Configuration.h file and change the following three items, as seen in the 3 GREEN boxes below.



```

Configuration.h - Marlin 2.0.x - Visual Studio Code

[Configuration.h]
#define SERIAL_PORT -1 //JTM was 0
#define SERIAL_PORT_2 1 //JTM was uncommented out and the value was -1
#define MOTHERBOARD BOARD BIGTREE_SKR_PRO V1_1 //JTM was BOARD_MELZI_CREALITY

```

NOTICE: pin labels are relative to driver chip carrier board, not Controller. Double check compatibility before use. Numbered pairs denote required jumpers.

Page 240 of 243

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- time to learn where the SKR PRO V1.1 board's pins file is located: look at the left side and find the Marlin-2.0.X/Marlin/src/pins/stm32 subdirectory, as seen in the pictures below. Open the "pins_BTT_SKR_PRO_V1_1.h" file by double clicking on it.

The screenshot shows two instances of Visual Studio Code. The left instance displays the file tree (EXPLORER) of the Marlin repository. The right instance shows the content of the file `pins_BTT_SKR_PRO_V1_1.h`.

File Tree (Left):

```

    File  Edit  Selection  View  Go  Debug  Terminal  Help
    └── EXPLORER
        └── OPEN EDITORS
            └── MARLIN-2.0.X
                ├── github
                ├── pio
                ├── vscode
                ├── buildroot
                ├── config
                ├── Configurations-release-2.0.3
                ├── data
                ├── docs
                ├── lib
                └── Marlin
                    ├── lib
                    └── src
                        ├── core
                        ├── feature
                        ├── gcode
                        ├── HAL
                        ├── inc
                        ├── lcd
                        ├── libs
                        ├── module
                        └── pins
                            ├── esp32
                            ├── linux
                            ├── lpc1768
                            ├── lpc1769
                            ├── mega
                            ├── rambo
                            ├── ramps
                            ├── sam
                            ├── sanguino
                            └── stm32
    > OUTLINE

```

Code Editor (Right):

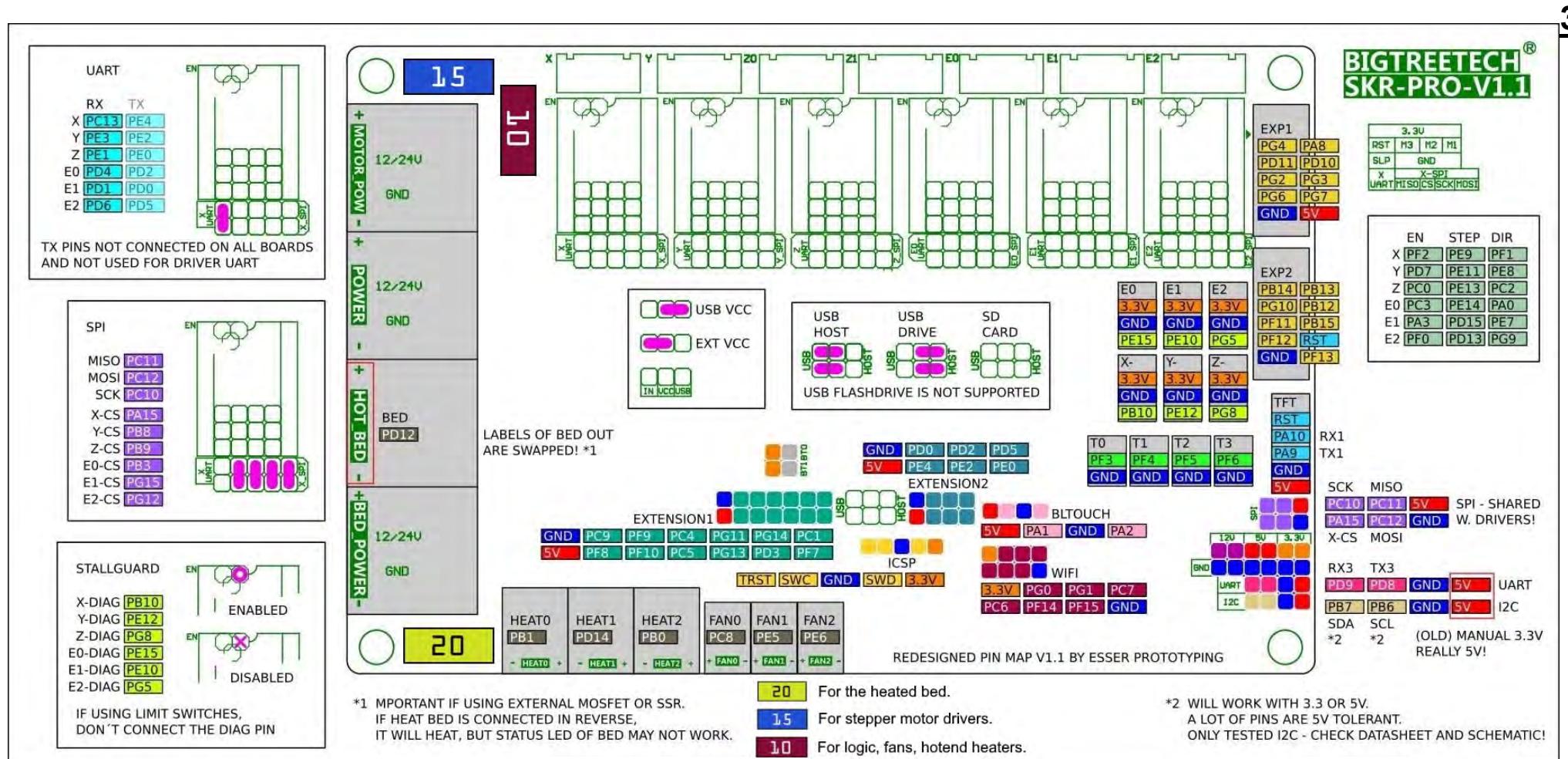
```

    File  edit  Selection  view  Go  Debug  Terminal  Help
    pins_BTT_SKR_PRO_V1_1.h ×
    Marlin > src > pins > stm32 > C_pins_BTT_SKR_PRO_V1_1.h ...
    161 // 
    162 // 
    163 // Temperature Sensors
    164 // 
    165 #define TEMP_0_PIN      PF4 // T1 <-> E0
    166 #define TEMP_1_PIN      PF5 // T2 <-> E1
    167 #define TEMP_2_PIN      PF6 // T3 <-> E2
    168 #define TEMP_BED_PIN    PF3 // T0 <-> Bed
    169 // 
    170 // 
    171 // Heaters / Fans
    172 // 
    173 #define HEATER_0_PIN    PB1 // Heater0
    174 #define HEATER_1_PIN    PD14 // Heater1
    175 #define HEATER_2_PIN    PB0 // Heater1
    176 #define HEATER_BED_PIN  PD12 // Hotbed
    C:\Firmware\Ender3(SKRV1.1)-PRO\Marlin-2.0.X\Marlin\src\pins\stm32
    C_pins_BEAST.h
    C_pins_BLACK_STM32F407VE.h
    C_pins_BTT_BT002_V1_0.h
    C_pins_BTT_GTR_V1_0.h
    C_pins_BTT_SKR_E3_DIP.h
    C_pins_BTT_SKR_MINI_E3_V1_0.h
    C_pins_BTT_SKR_MINI_E3_V1_2.h
    C_pins_BTT_SKR_MINI_E3.h
    C_pins_BTT_SKR_MINI_V1_1.h
    C_pins_BTT_SKR_PRO_V1_1.h
    C_pins_CHITU3D.h
    C_pins_FLY407ZG.h
    C_pins_FYSETC_AIO_I2.h
    C_pins_FYSETC_CHEETAH_V12.h
    C_pins_FYSETC_CHEETAH.h
    180 // 
    181 // 
    182 // Misc. Functions
    183 // 
    184 #define SDSS          PB12
    185 // 
    186 //** 
    187 * 
    188 *      NC | + - | GND
    189 *      RESET | + - | PF12(SD_DETECT)
    190 *      (MOSI)PB15 | + - | PF11(BTN_EN2)
    191 *      (SD_SS)PB12 | + - | PG10(BTN_EN1)
    192 *      (SCK)PB13 | + - | PB14(MISO)
    193 // 
    194 EXP2
    195 // 
    196 *      5V | + - | GND
    197 *      (LCD_D7) PG7 | + - | PG6 (LCD_D6)
    198 *      (LCD_D5) PG3 | + - | PG2 (LCD_D4)
    199 *      (LCD_RS) PD10 | + - | PD11 (LCD_EN)
    200 *      (BTN_ENC) PAB | + - | PG4 (BEEPER)
    EXP1

```

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- We want to set the fan on the hot end to automatically turn on when the hot end starts to get hot. In the pins_BTT_SKR_PRO_V1_1.h file you will find all the pins that are defined for the board. They should all conform to the SKR PRO V1.1 Pin diagram shown below.

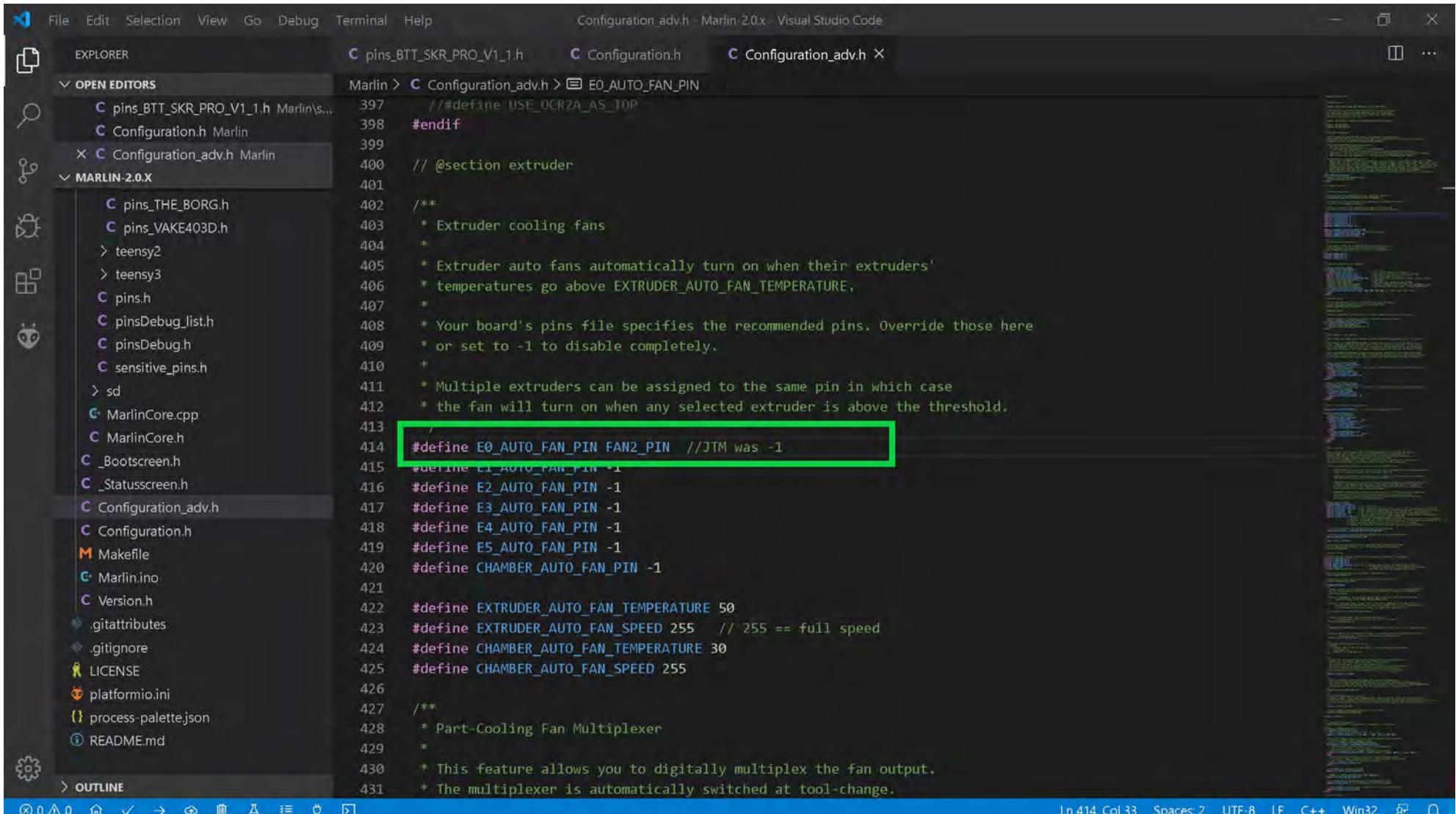


- We want to copy and paste the Marlin name or the actual pin number of where you hooked up the fan for your hot end and use that to set "E0_AUTO_FAN_PIN" in the Configuration_adv.h file. I am connecting FAN1 port to my electronics case fan. I am connecting FAN0 port to my part or print cooling fan and connecting FAN2 port to my dual 5015 hot end cooling fans. But we are only interested in how to set the hot end cooling fan up. So, in this example I will use FAN2 port for my hot end cooling fan. In "pins_BTT_SKR_PRO_V1_1.h" we see FAN2 port is defined as PE6. So you can choose to copy "PE6" or "FAN2_PIN".

3 Pin Diagram is done by Thomas White

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- To set the hot end fan, I will use FAN2_PIN and set "E0_AUTO_FAN_PIN" in the Configuration_adv.h file to FAN2_PIN, as seen in the picture below



File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code

EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h

OPEN EDITORS pins_BTT_SKR_PRO_V1_1.h Marlin\... Configuration.h Marlin Configuration_adv.h Marlin MARLIN-2.0.X pins_THE_BORG.h pins_VAKE403D.h > teensy2 > teensy3 pins.h pinsDebug.list.h pinsDebug.h sensitive_pins.h > sd MarlinCore.cpp MarlinCore.h _Bootscreen.h _Statusscreen.h Configuration_adv.h Configuration.h Makefile Marlin.ino Version.h .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

```

397 // #define USE_OCR2A_AS_TOP
398 #endif
399
400 // @section extruder
401
402 /**
403 * Extruder cooling fans
404 *
405 * Extruder auto fans automatically turn on when their extruders'
406 * temperatures go above EXTRUDER_AUTO_FAN_TEMPERATURE.
407 *
408 * Your board's pins file specifies the recommended pins. Override those here
409 * or set to -1 to disable completely.
410 *
411 * Multiple extruders can be assigned to the same pin in which case
412 * the fan will turn on when any selected extruder is above the threshold.
413 */
414 #define E0_AUTO_FAN_PIN FAN2_PIN //JTM was -1
415 #define E1_AUTO_FAN_PIN -1
416 #define E2_AUTO_FAN_PIN -1
417 #define E3_AUTO_FAN_PIN -1
418 #define E4_AUTO_FAN_PIN -1
419 #define E5_AUTO_FAN_PIN -1
420 #define CHAMBER_AUTO_FAN_PIN -1
421
422 #define EXTRUDER_AUTO_FAN_TEMPERATURE 50
423 #define EXTRUDER_AUTO_FAN_SPEED 255 // 255 == full speed
424 #define CHAMBER_AUTO_FAN_TEMPERATURE 30
425 #define CHAMBER_AUTO_FAN_SPEED 255
426
427 /**
428 * Part-Cooling Fan Multiplexer
429 *
430 * This feature allows you to digitally multiplex the fan output.
431 * The multiplexer is automatically switched at tool-change.

```

Ln 414, Col 33 Spaces: 2 UTF-8 LF C++ Win32

- To see more Marlin setup for the latest release, [please refer to the stepper motor driver section of this document for the stepper motor driver of your choice.](#)