

Stepper Driver Configurations for SKR PRO V1.1 Board



By

@GadgetAngel



Based on Work by

@rfulling

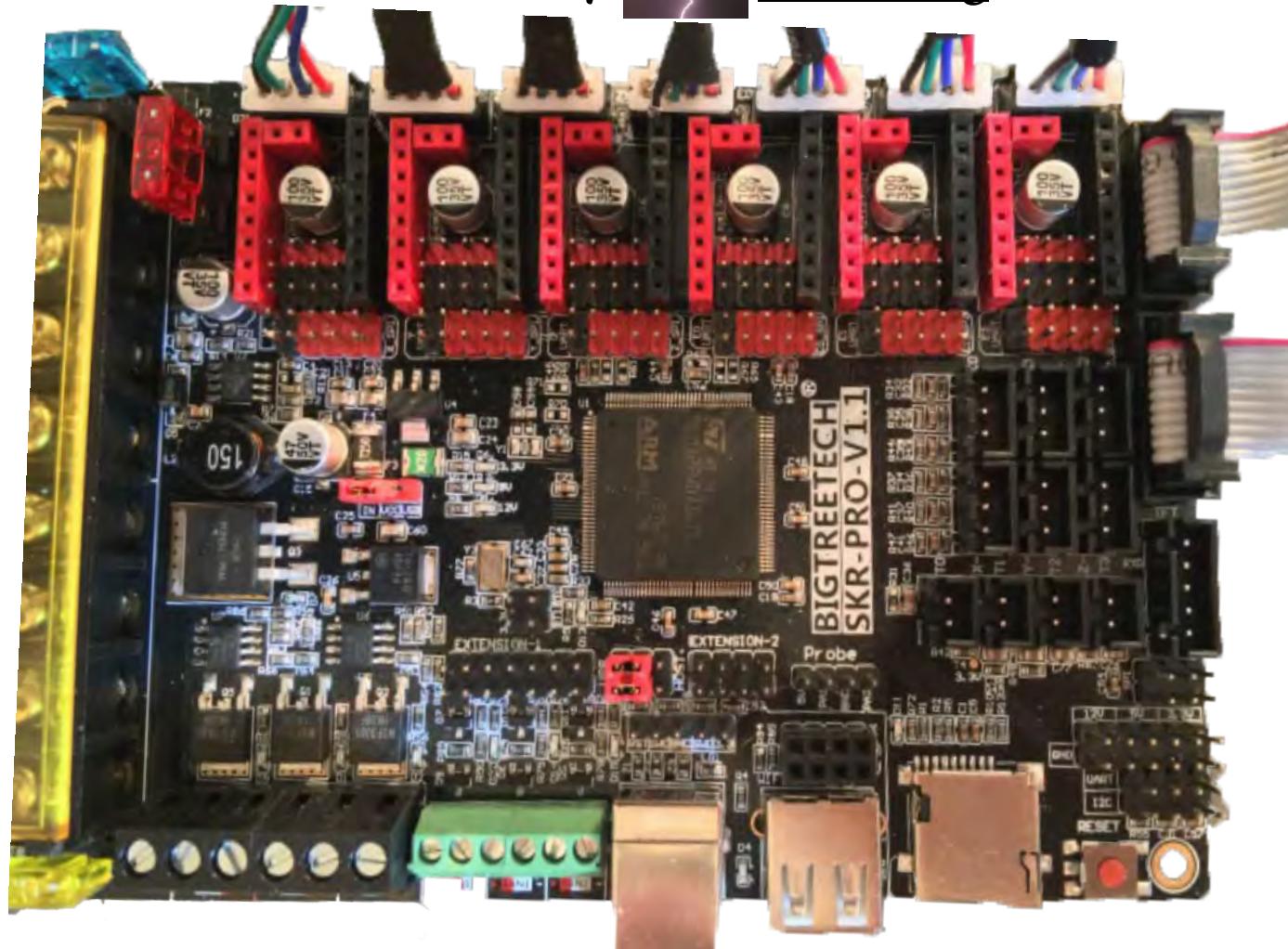


Table of Contents

POLOLU A4988	1
The (latest release of) Marlin Setup for POLOLU A4988 Drivers	5
BIQU A4988	9
The (latest release of) Marlin Setup for BIQUE A4988 Drivers	13
DRV8825	17
The (latest release of) Marlin Setup for DRV8825 Drivers	22
BIQU LV8729	27
The (latest release of) Marlin Setup for BIQUE LV8729 Drivers	32
FYSETC LV8729	38
The (latest release of) Marlin Setup for FYSETC LV8729 Drivers	43
LERDGE LV8729	49
The (latest release of) Marlin Setup for LERDGE LV8729 Drivers	54
MKS LV8729	60
The (latest release of) Marlin Setup for MKS LV8729 Drivers	65
FYSETC S6128 V1.1	71
The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers	76
FYSETC ST820	82
The (latest release of) Marlin Setup for FYSETC ST820 Drivers	87
BIQU ST820	94
The (latest release of) Marlin Setup for BIQUE ST820 Drivers	99
POLOLU ST820 (STSPIN820)	106
The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers	111
POLOLU MP6500	118
The (latest release of) Marlin Setup for POLOLU MP6500 Drivers	122

Table of Contents

POLOLU TB67S249FTG.....	128
The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers.....	133
BIQU S109	139
The (latest release of) Marlin Setup for BIQUE S109 Drivers	144
FYSETC S109	150
The (latest release of) Marlin Setup for FYSETC S109 Drivers	155
BIQU TMC2100 Stand-alone Mode.....	161
The (latest release of) Marlin Setup for BIQUE TMC2100 Drivers in Stand-alone Mode	166
MKS TMC2100 Stand-alone Mode.....	172
The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode	177
BIQU TMC2130 Stand-alone Mode.....	183
The (latest release of) Marlin Setup for BIQUE TMC2130 Drivers in Stand-alone Mode	188
BIQU TMC2130 SPI Mode	194
The (latest release of) Marlin Setup for BIQUE TMC2130 Drivers in SPI Mode.....	197
BIQU TMC2208 V3.0 Stand-alone Mode	211
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in Stand-alone Mode	215
BIQU TMC2208 V3.0 One Time Programming (OTP) Mode.....	221
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode ...	225
BIQU TMC2208 V3.0 UART Mode.....	231
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in UART Mode	234
FYSETC TMC2208 V1.2 Stand-alone Mode	245
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode.....	249
FYSETC TMC2208 V1.2 One Time Programming (OTP) Mode.....	255
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode...	259

Table of Contents

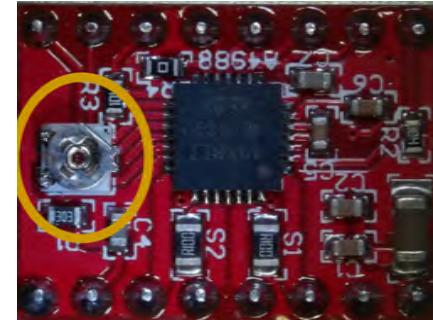
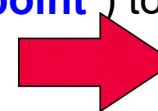
FYSETC TMC2208 V1.2 UART Mode.....	265
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode	268
BIQU TMC2225 V1.0 Stand-alone Mode	279
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode	283
BIQU TMC2225 V1.0 One Time Programming (OTP) Mode	289
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode ...	293
BIQU TMC2225 V1.0 UART Mode.....	299
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode	302
BIQU TMC2209 V1.2 Stand-alone Mode for StealthChop.....	313
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop	317
BIQU TMC2209 V1.2 Stand-alone Mode for SpreadCycle	323
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle	327
BIQU TMC2209 V1.2 UART Mode.....	333
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode	336
BIQU TMC5160 V1.2 SPI Mode.....	350
The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode.....	353
BIQU TMC5161 V1.0 SPI Mode.....	367
The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode.....	370
APPENDIX A.....	384
How to adjust the Vref on a Stepper Motor Driver board using the Potentiometer.....	384
APPENDIX B.....	386
For the TMC drivers what's the difference between stand-alone mode and ("UART" or "SPI ") modes?.....	386
How to Calculate Vref for Non-TMC Stepper Motor Divers	386
How to Calculate Vref for TMC Stepper Motor Drivers	387

Table of Contents

Driving Current Calculation Formulas for TMC Stepper Motor Drivers	388
#1 TMC2100 with $R_s=0.110\Omega$ (110mΩ)	388
#2 TMC2130 with $R_s=0.110\Omega$ (110mΩ)	388
#3 TMC2208 with $R_s=0.110\Omega$ (110mΩ) for Stand-alone Mode.....	388
#4 TMC2208 with $R_s=0.110\Omega$ (110mΩ) for UART Mode	389
#5 TMC2209 with $R_s=0.110\Omega$ (110mΩ) for Stand-alone Mode.....	389
#6 TMC2209 with $R_s=0.110\Omega$ (110mΩ) for UART Mode	390
#7 TMC2225 with $R_s=0.150\Omega$ (150mΩ) for UART Mode	390
#8 TMC5160 with $R_s=0.075\Omega$ (75mΩ) for SPI Mode	390
#9 TMC5161 with $R_s=0.062\Omega$ (62mΩ) for SPI Mode	391
#10 TMC2225 with $R_s=0.150\Omega$ (150mΩ) for Stand-alone Mode.....	391
APPENDIX C	392
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers.....	392
SKR PRO V1.1 Color PINS Diagram.....	405
SKR PRO V1.1 Color Wiring Diagram.....	406

POLOLU A4988

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's "**V_{ref} Test point**") to adjust your V_{ref}.



Note: "**V_{ref} Test point**" location is on the bottom of the driver board.



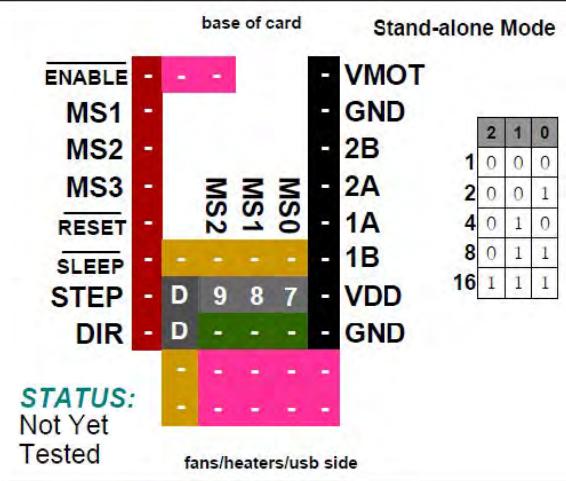
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

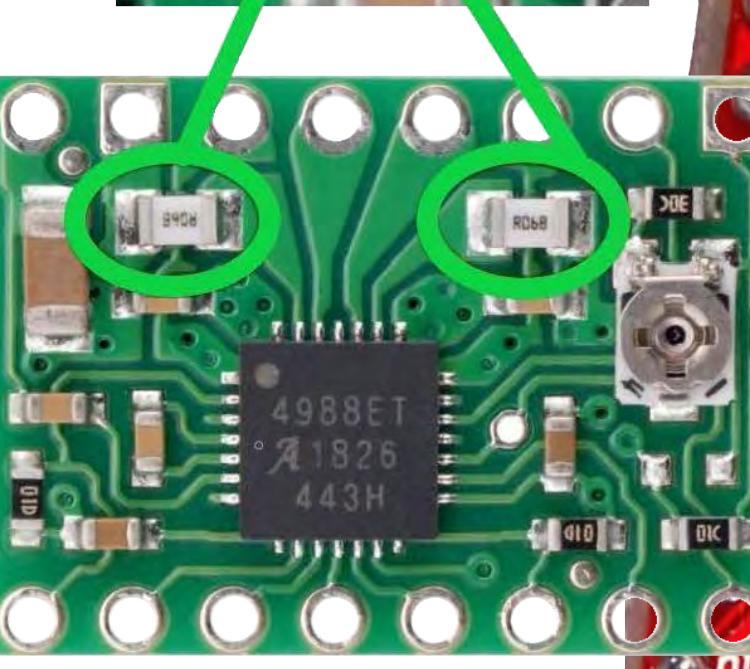
Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	2 Phase
	Low	Low	High	Half step	1-2 Phase
	Low	High	Low	Quarter step	W1-2 Phase
	Low	High	High	Eighth step	2W1-2 Phase
	High	High	High	Sixteenth step	4W1-2 Phase
Driving Current Calculation Formula R_s (Typical Sense Resistor) = 0.1Ω	$I_{MAX} = V_{ref} / (8 * R_s)$			$V_{ref} = 8 * I_{MAX} * R_s$	

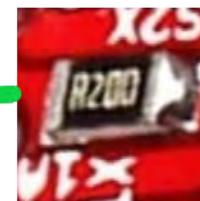
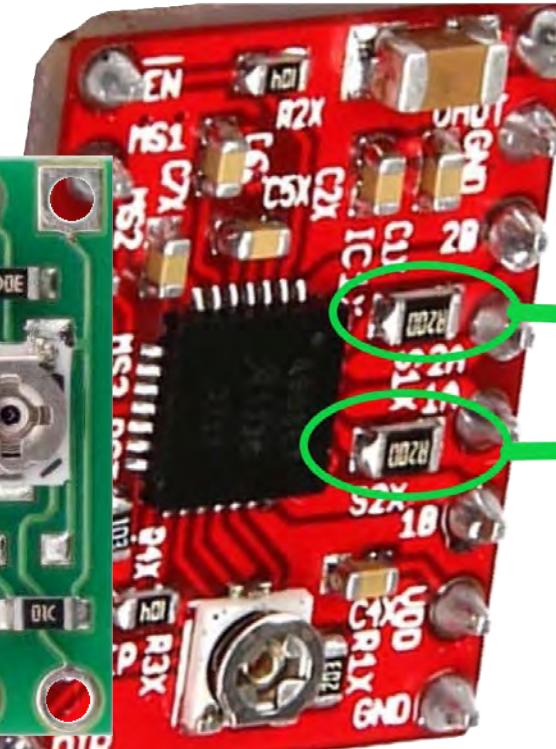
POLOLU A4988



Note: Not all driver boards for the A4988 use the same current sense resistors (R_s); check your driver board for the value of the (R_s) resistors by examining the board, as shown in **GREEN** below. The **GREEN PCB** shows a 0.1 Ohm (R100) sense resistor value. The **RED PCB** shows a 0.2 Ohms (R200) sense resistor value. Sense resistors (R_s) can appear in the following values, (these are just a few values): R050=0.05 Ohms; R068=0.068 Ohms; R100=0.1 Ohms; R200 = 0.2 Ohms.



R_s = R050 is 0.05 Ohms
 R_s = R068 is 0.068 Ohms
 R_s = R100 is 0.1 Ohms
 R_s = R150 is 0.15 Ohms
 R_s = R200 is 0.2 Ohms
 R_s = R220 is 0.22 Ohms



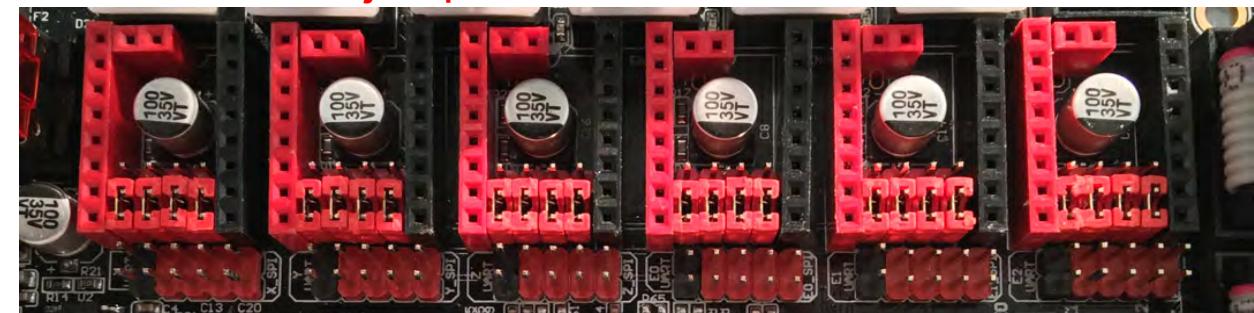
POLOLU A4988

Stand-alone Mode

STEP

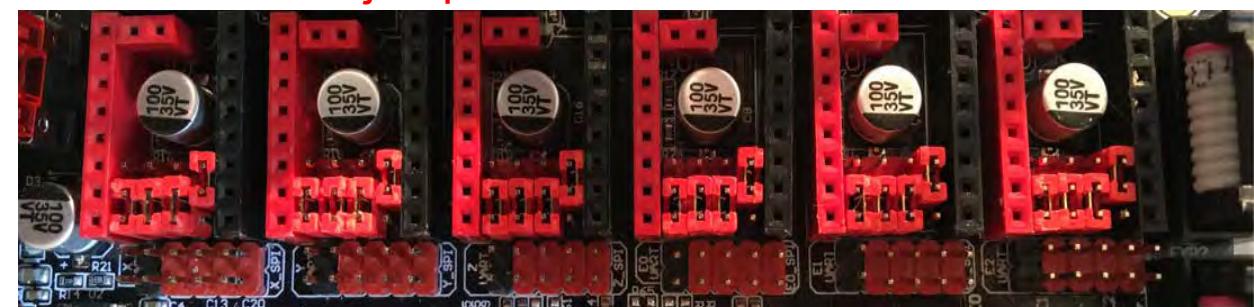
ENABLE	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	2B
MS3	-	-	-	2A
RESET	-	MS2	MS1	MS0
SLEEP	-	-	-	-
STEP	D	9	8	7
DIR	D	9	8	7
	-	-	-	-
	-	-	-	-

Note: The "D" jumper MUST be SET!

**1 / 2**

ENABLE	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	2B
MS3	-	-	-	2A
RESET	-	MS2	MS1	MS0
SLEEP	-	-	-	-
STEP	D	9	8	7
DIR	D	9	8	-
	-	-	-	-
	-	-	-	-

Note: The "D" jumper MUST be SET!



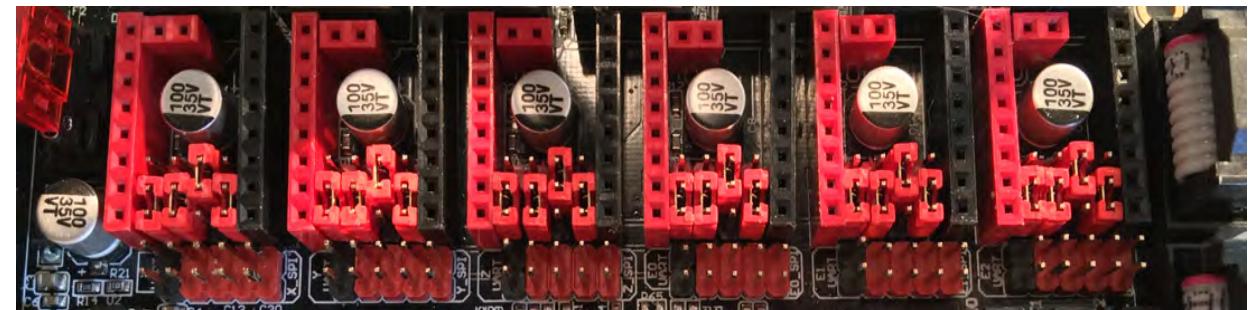
POLOLU A4988

Stand-alone Mode

ENABLE	- -	VMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	8	1B
STEP	D 9 8 7	VDD
DIR	D 9 - 7	GND
	- - - -	- - - -

1 / 4

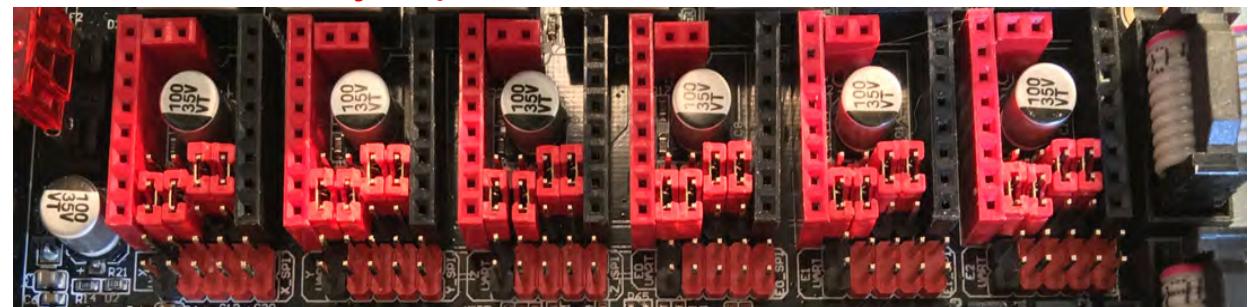
Note: The "D" jumper MUST be SET!



ENABLE	- -	VMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	8 7	1B
STEP	D 9 8 7	VDD
DIR	D 9 - -	GND
	- - - -	- - - -

1 / 8

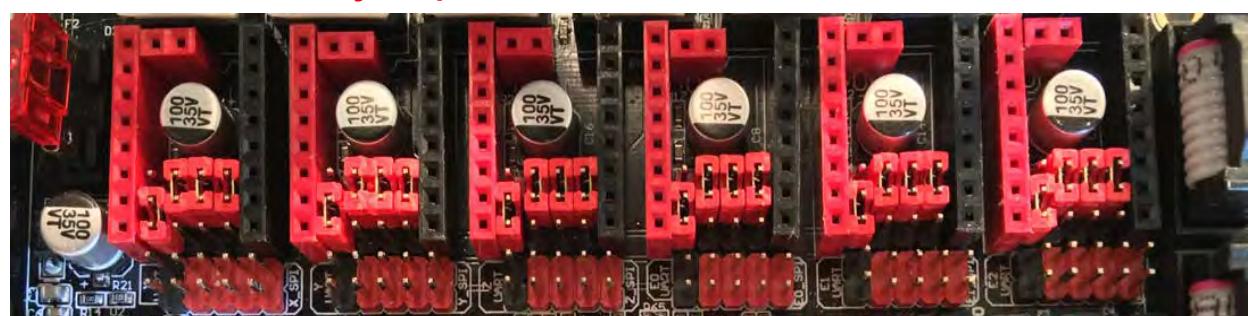
Note: The "D" jumper MUST be SET!



ENABLE	- -	VMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 8 7	1B
STEP	D 9 8 7	VDD
DIR	D - - -	GND
	- - - -	- - - -

1 / 16

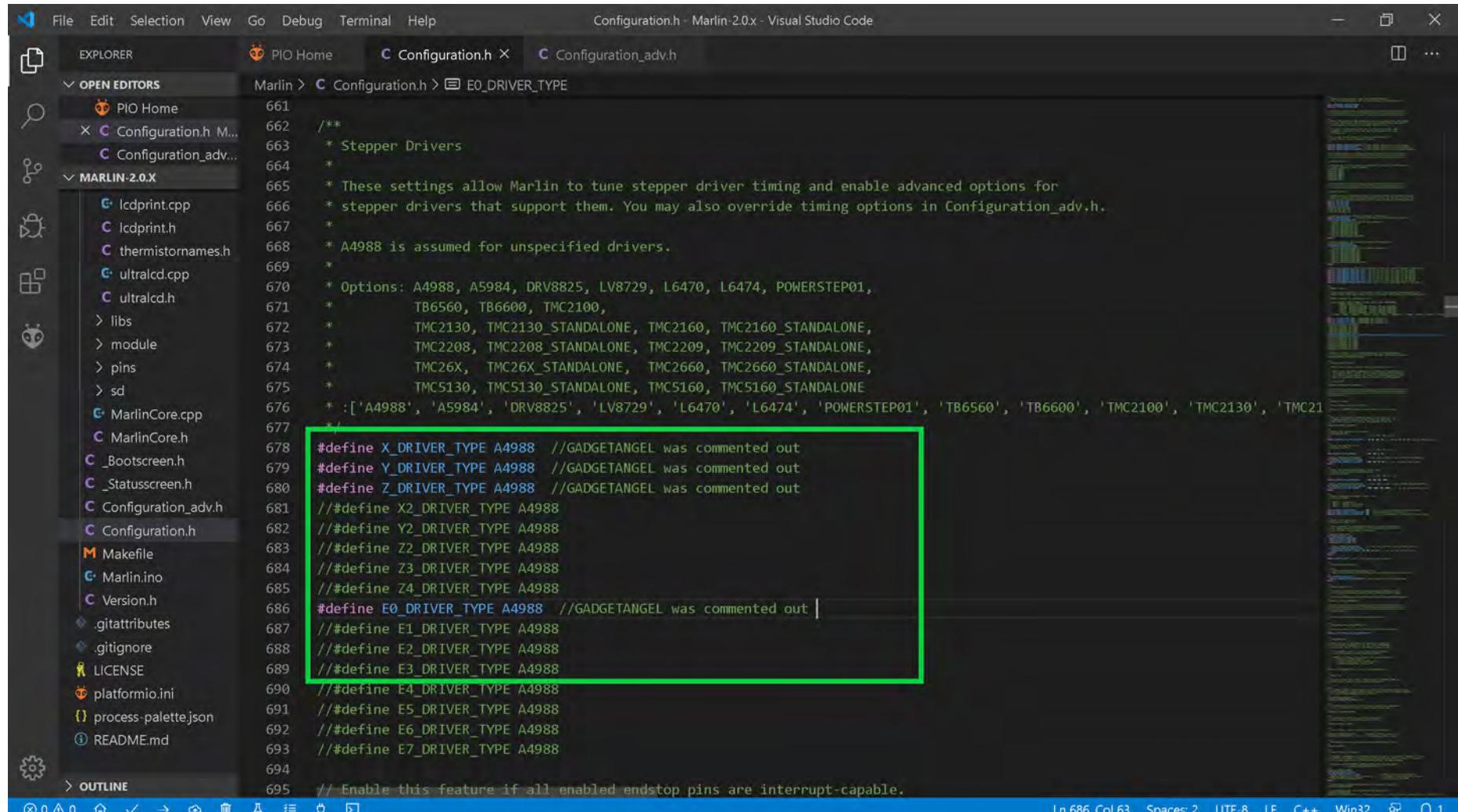
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for POLOLU A4988 Drivers

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

- Change the stepper motor drivers so that Marlin knows you are using POLOLU 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 POLOLU 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 Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
677 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
678 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
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 A4988 //GADGETANGEL was commented out
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 //##define E8_DRIVER_TYPE A4988
694 // Enable this feature if all enabled endstop pins are interrupt-capable.
695
Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 1

```

- Go to the next page.

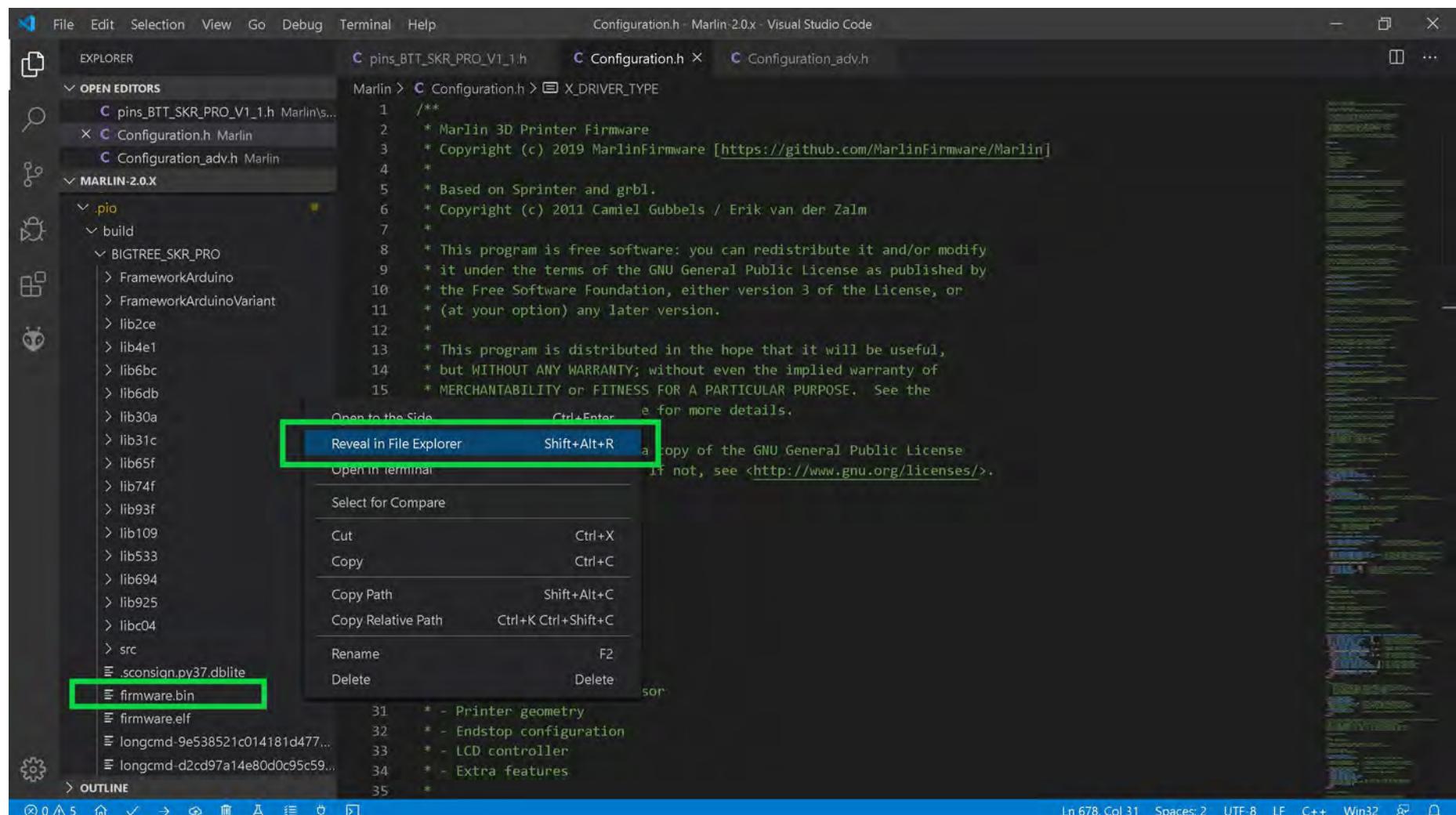
The (latest release of) Marlin Setup for POLOLU A4988 Drivers

- The end of Marlin setup for POLOLU A4988 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.

- 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 A4988 Drivers

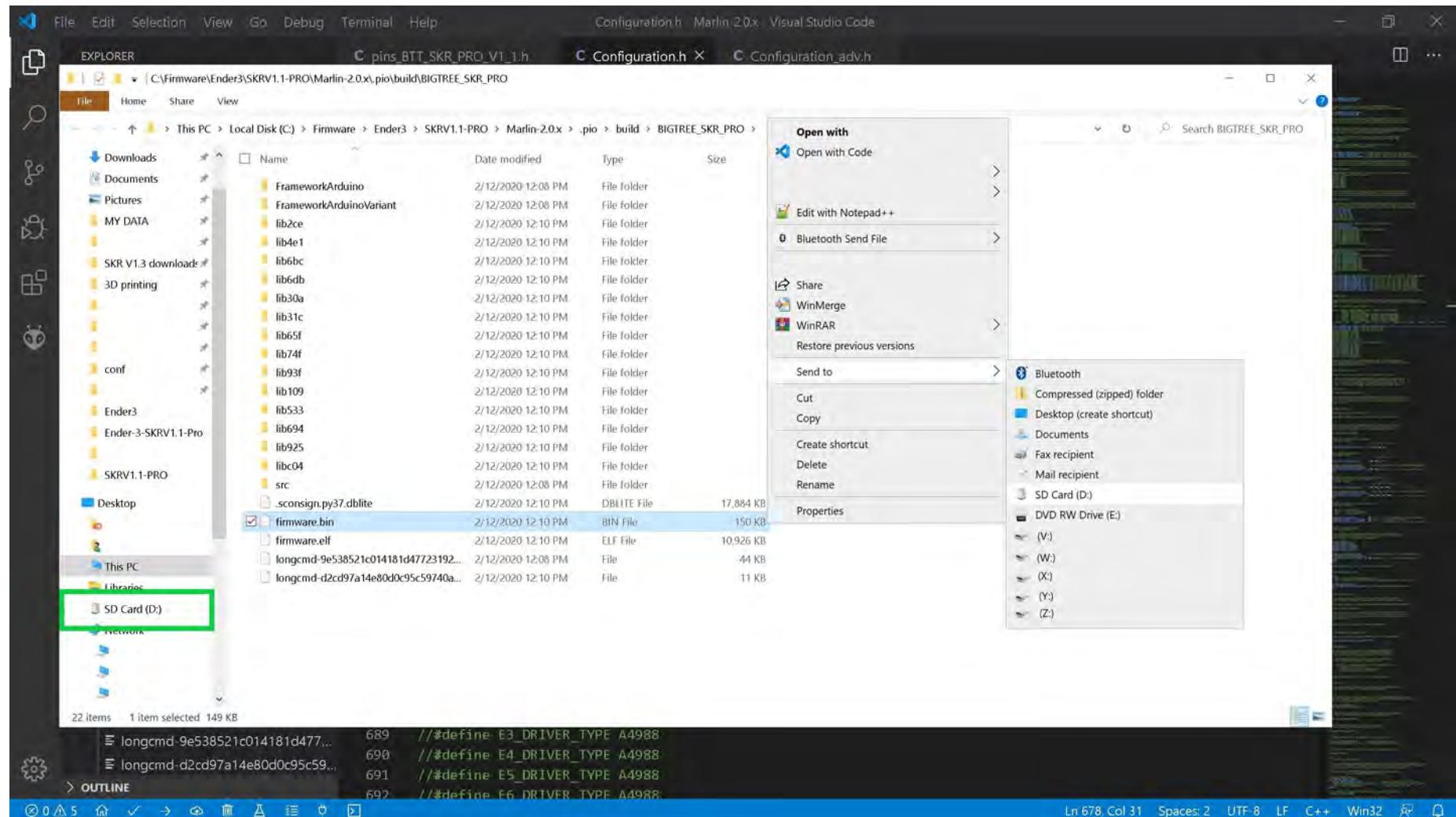
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file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



- Go to the next page.

The (latest release of) Marlin Setup for POLOLU A4988 Drivers

- Ensure your micro SD card, which 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".

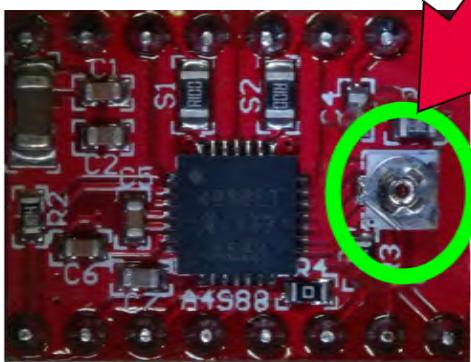


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

BIQU A4988

base of card		Stand-alone Mode	
ENABLE	- - -	VMOT	GND
MS1	-	2B	2 1 0
MS2	-	2A	1 0 0 0
MS3	-	1A	2 0 0 1
RESET	M2	4	4 0 1 0
SLEEP	M1 2	8	8 0 1 1
STEP	M0 9 8 7	1B	16 1 1 1
DIRECTION	D 9 8 7	VDD	GND
STATUS:	Not Yet Tested		
	fans/heaters/usb side		

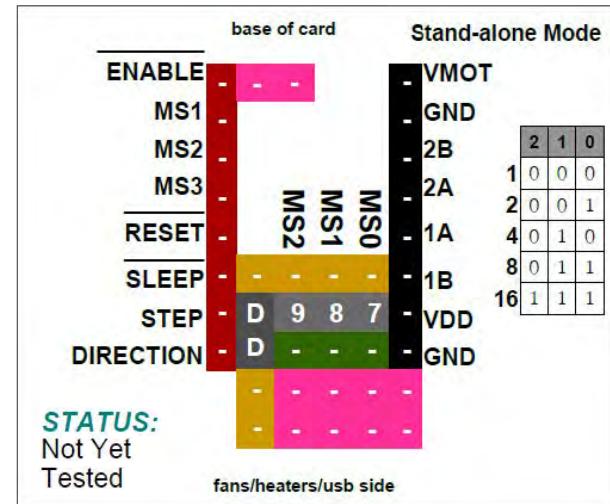
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



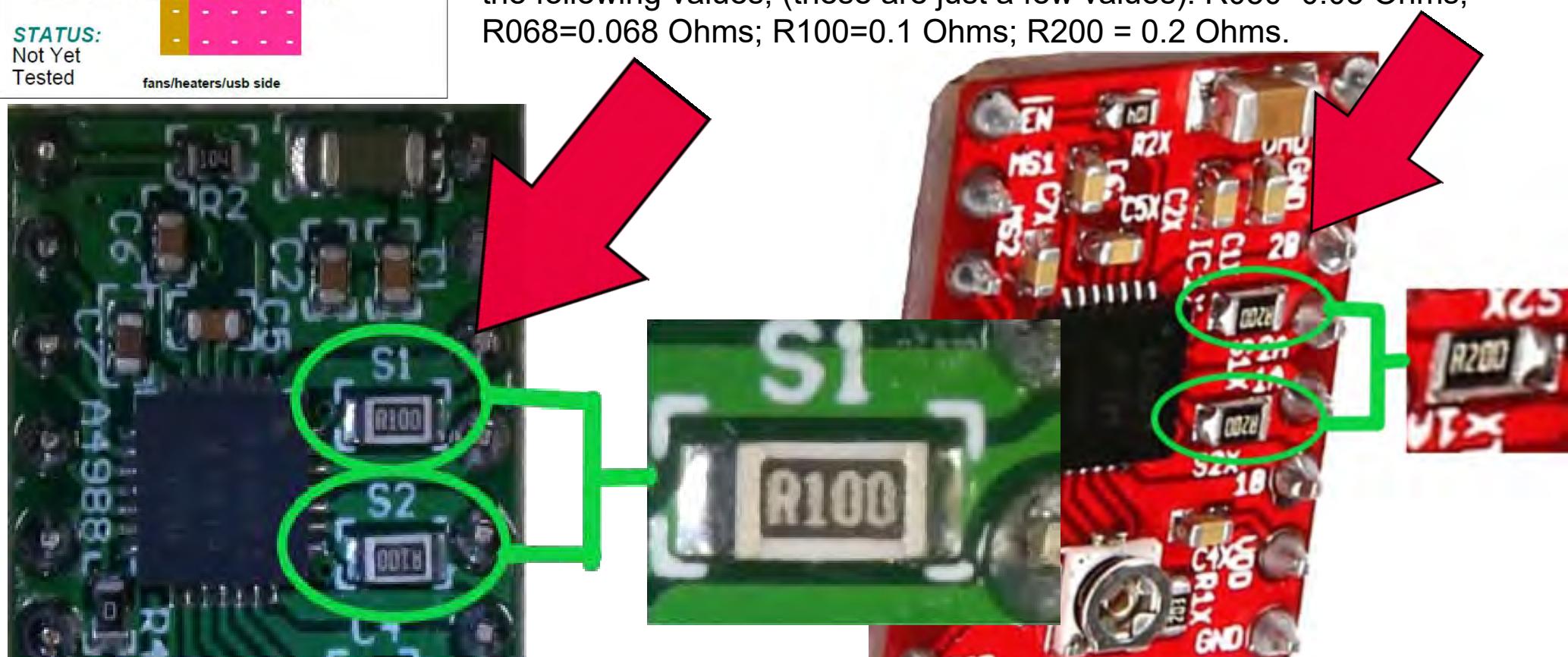
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
 BIQU® A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	
	Low	Low	High	Half step	2 Phase
	Low	High	Low	Quarter step	1-2 Phase
	Low	High	High	Eighth step	W1-2 Phase
	High	High	High	Sixteenth step	2W1-2 Phase
					4W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} = V_{ref} / (8 * R_s)$			$V_{ref} = 8 * I_{MAX} * R_s$	
R_s (Typical Sense Resistor) = 0.1Ω					

BIQU A4988

Note: Not all driver boards for the A4988 use the same current sense resistors (R_s); check your driver board for the value of the (R_s) resistors by examining the board, as shown in **GREEN** below. The **GREEN PCB** shows a 0.1 Ohm (R100) sense resistor value. The **RED PCB** shows a 0.2 Ohms (R200) sense resistor value. Sense resistors (R_s) can appear in the following values, (these are just a few values): R050=0.05 Ohms; R068=0.068 Ohms; R100=0.1 Ohms; R200 = 0.2 Ohms.

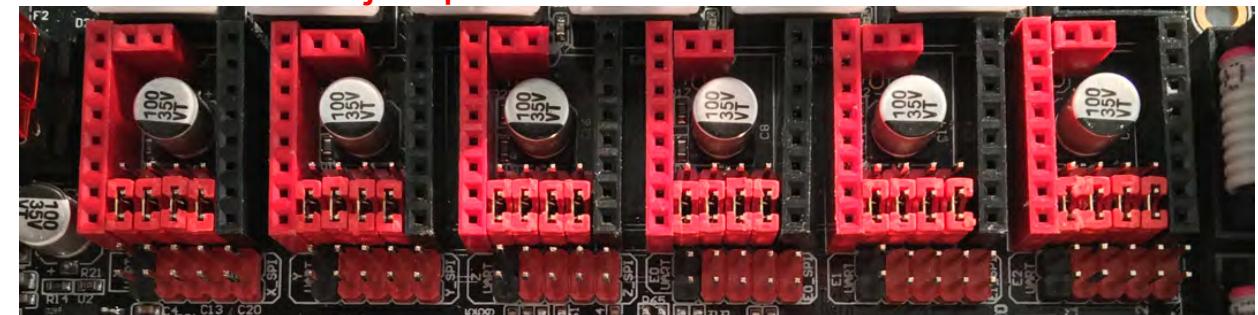


$R_s = R050$ is 0.05 Ohms; $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms; $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms; $R_s = R220$ is 0.22 Ohms

BIQU A4988**Stand-alone Mode****STEP**

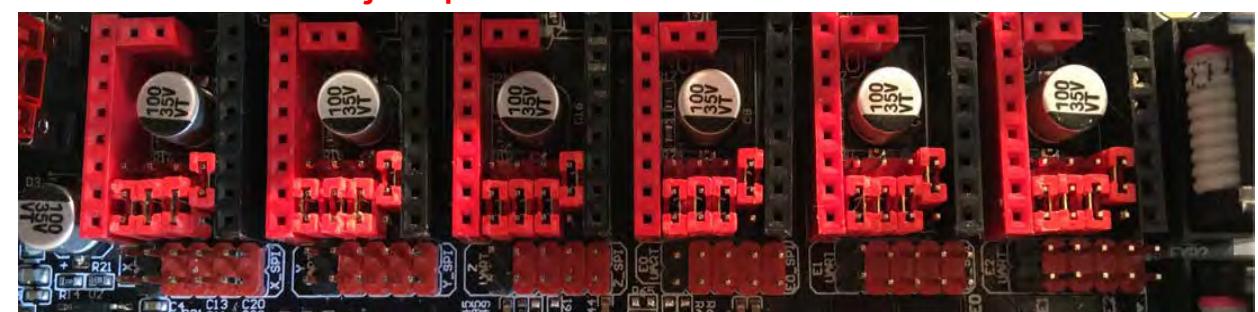
ENABLE	[]		VMOT
MS1			GND
MS2			2B
MS3			2A
RESET	[]	MS2 MS1 MS0	1A
SLEEP	[]		1B
STEP	D 9 8 7		VDD
DIRECTION	D 9 8 7		GND

Note: The "D" jumper MUST be SET!

**1 / 2**

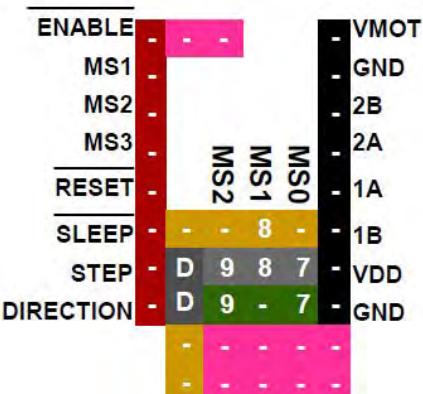
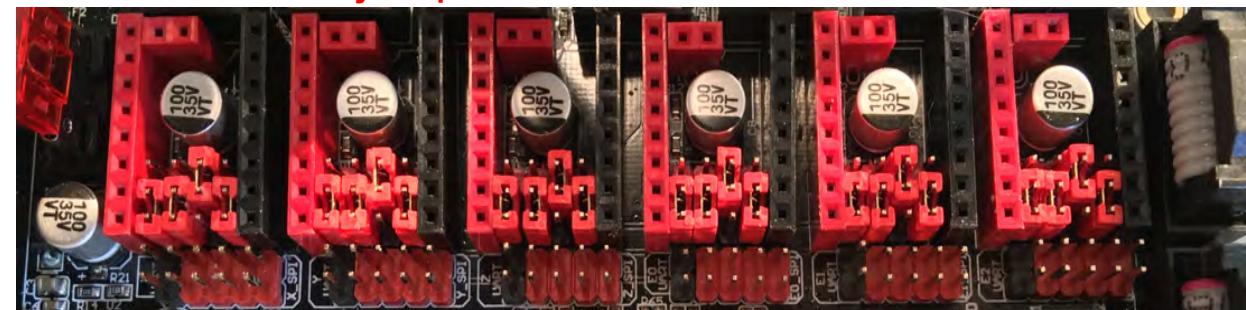
ENABLE	[]		VMOT
MS1			GND
MS2			2B
MS3			2A
RESET	[]	MS2 MS1 MS0	1A
SLEEP	[]		1B
STEP	D 9 8 7		VDD
DIRECTION	D 9 8 []		GND

Note: The "D" jumper MUST be SET!

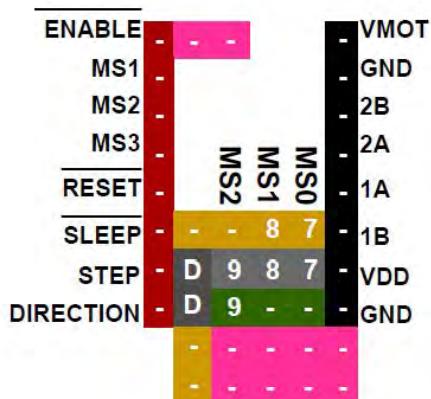
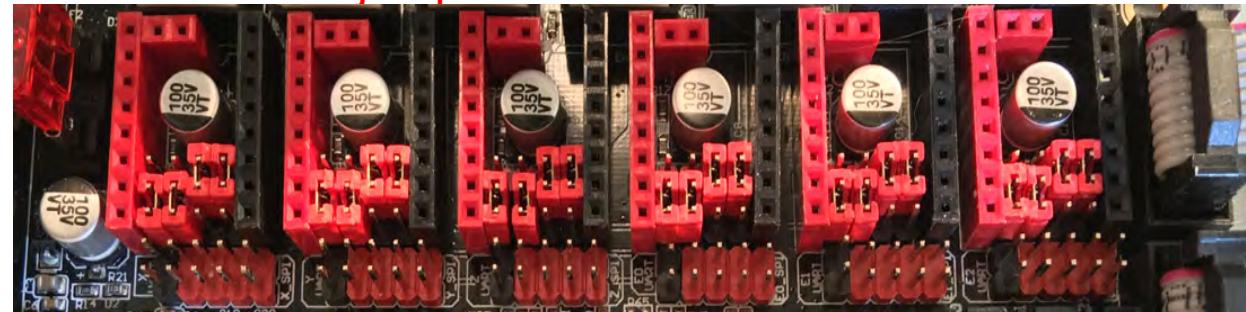


BIQU A4988**Stand-alone Mode**

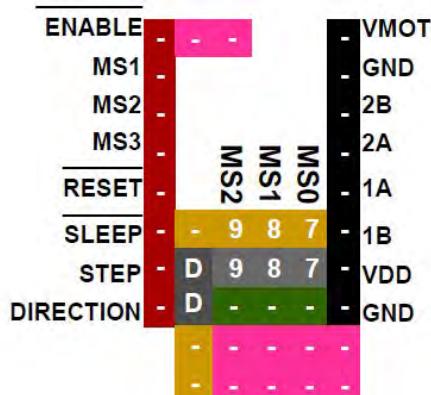
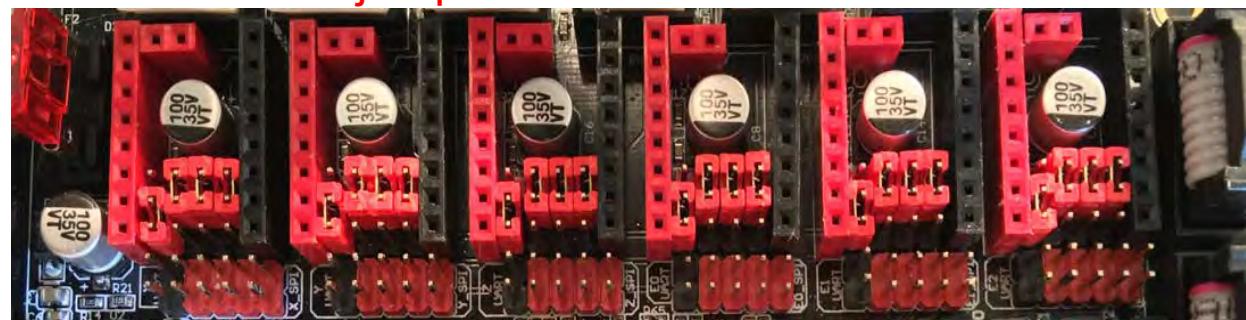
1 / 4

**Note: The "D" jumper MUST be SET!**

1 / 8

**Note: The "D" jumper MUST be SET!**

1 / 16

**Note: The "D" jumper MUST be SET!**

The (latest release of) Marlin Setup for BIQU A4988 Drivers

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

- Change the stepper motor drivers so that Marlin knows you are using BIQU 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 BIQU 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 Configuration_adv.h
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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
678 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
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
686 #define E0_DRIVER_TYPE A4988 //GADGETANGEL 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
687
688
689
690
691
692
693
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1

- Go to the next page.

The (latest release of) Marlin Setup for BIQU A4988 Drivers

- The end of Marlin setup for BIQU A4988 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 C pins_BTT_SKR_PRO_V1_1.h C Configuration.h X Configuration_adv.h

OPEN EDITORS Marlin > C Configuration.h > X_X.DRIVER_TYPE

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

C pins_THE_BORG.h C pins_VAKE403D.h > teensy2 > teensy3 C pins.h C pinsDebug.list.h C pinsDebug.h C sensitive_pins.h > sd E 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 Y LICENSE P platformio.ini I process-palette.json I README.md

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

BIGTREETECH_SKR_PRO	SUCCESS	00:02:31.294
BIGTREETECH_BTT002	IGNORED	
teensy31	IGNORED	
teensy35	IGNORED	
esp32	IGNORED	
linux_native	IGNORED	
SAMD51_grandcentral_m4	IGNORED	
rumba32_f446ve	IGNORED	
mks_rumba32	IGNORED	
include_tree	IGNORED	

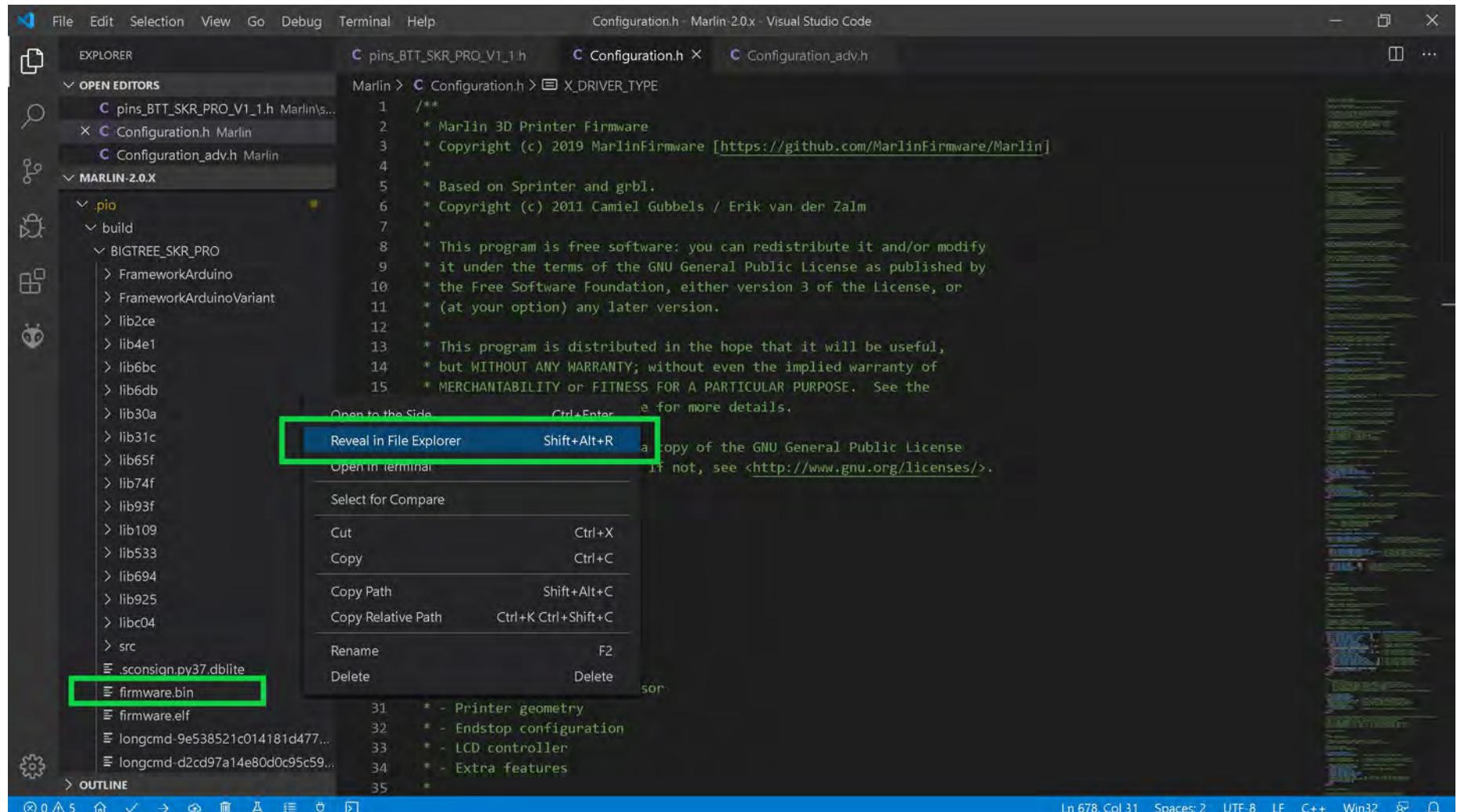
===== 1 succeeded in 00:02:31.294 =====

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 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 Windows machine open a file explorer window.

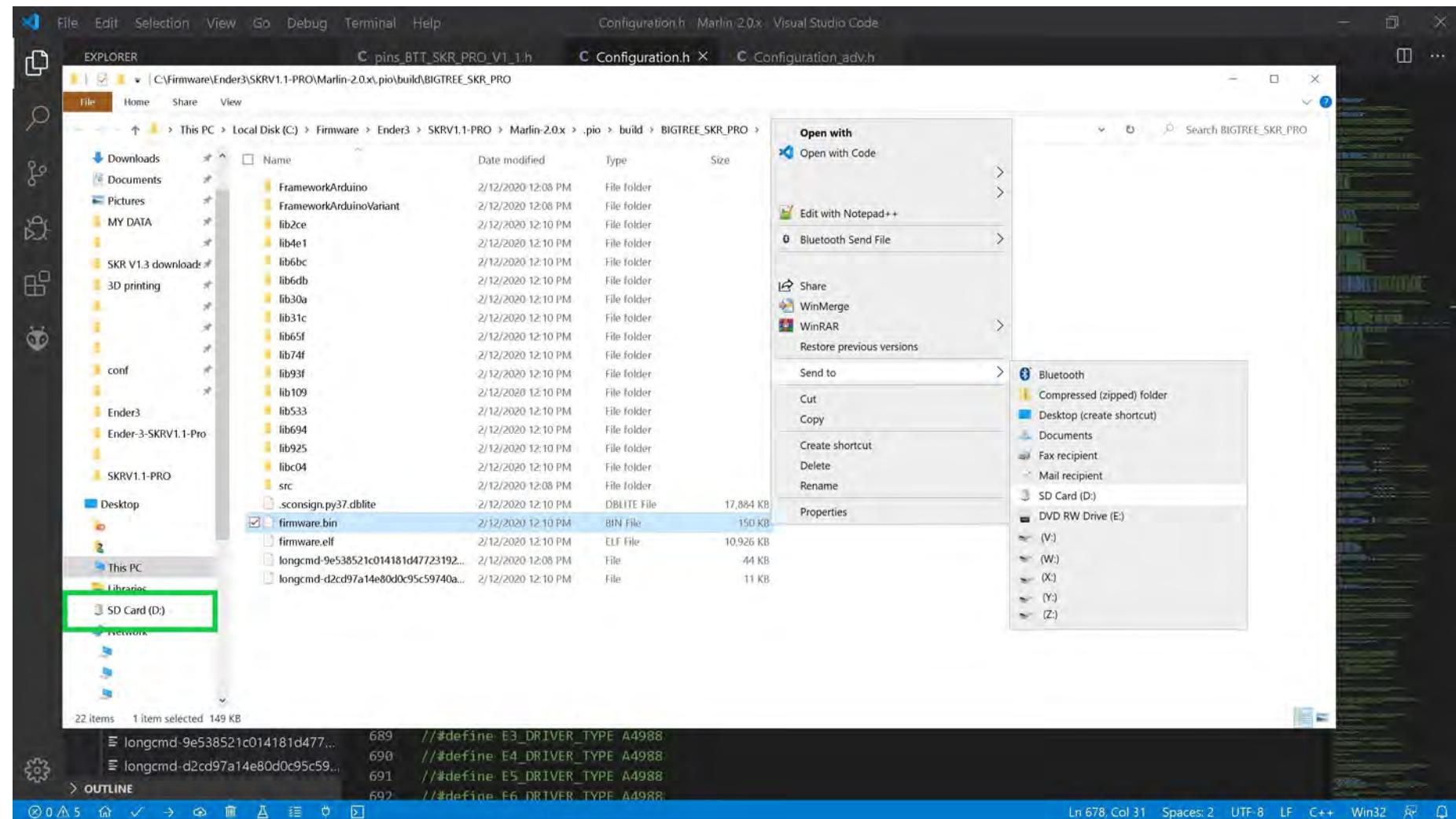


- Go to the next page.

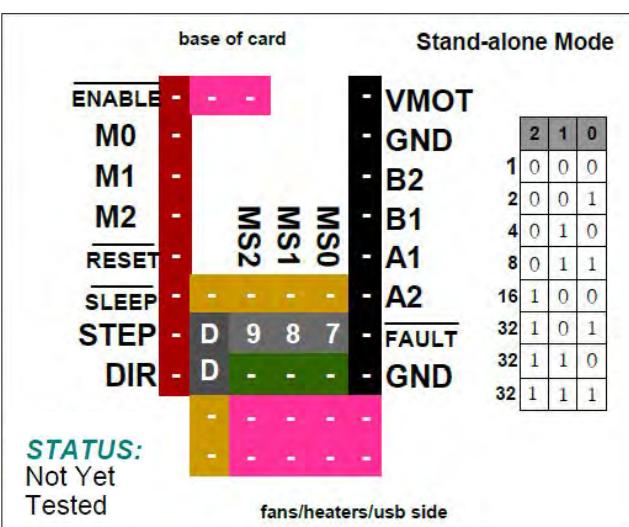
The (latest release of) Marlin Setup for BIQU A4988 Drivers

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, is in a micro-to-SD-card adapter and plugged into your Windows' 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".



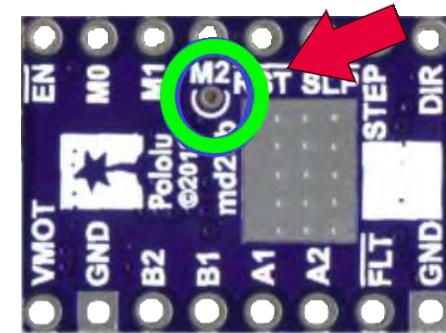
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

DRV8825

NOTE: Use the potentiometer (POT) on the top of the board (or the board's "V_{ref} Test point") to adjust your V_{ref}. See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

Note: "V_{ref} Test point" location is on the bottom of the driver board, as shown in GREEN



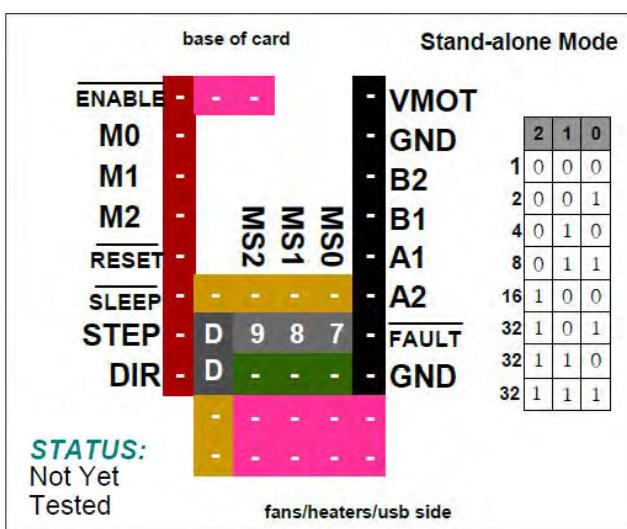
Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu	Low	Low	Low	Full step	2 Phase
DRV8825	Low	Low	High	Half step	1-2 Phase
Maximum 32 Subdivision	Low	High	Low	1/4 step	W1-2 Phase
45V DC	Low	High	High	1/8 step	2W1-2 Phase
2.2A (peak)	High	Low	Low	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	High	High	Low	1/32 step	8W1-2 Phase
	High	High	High	1/32 step	8W1-2 Phase

Driving Current Calculation Formula

$$R_S \text{ (Typical Sense Resistor)} = 0.1\Omega$$

$$I_{MAX} = \frac{V_{ref}}{5 * R_S}$$

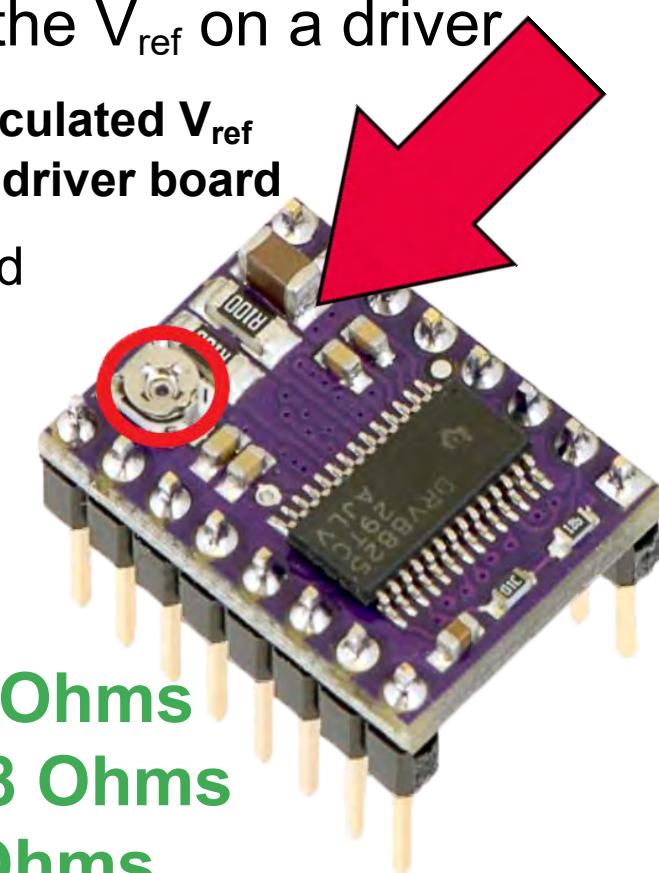
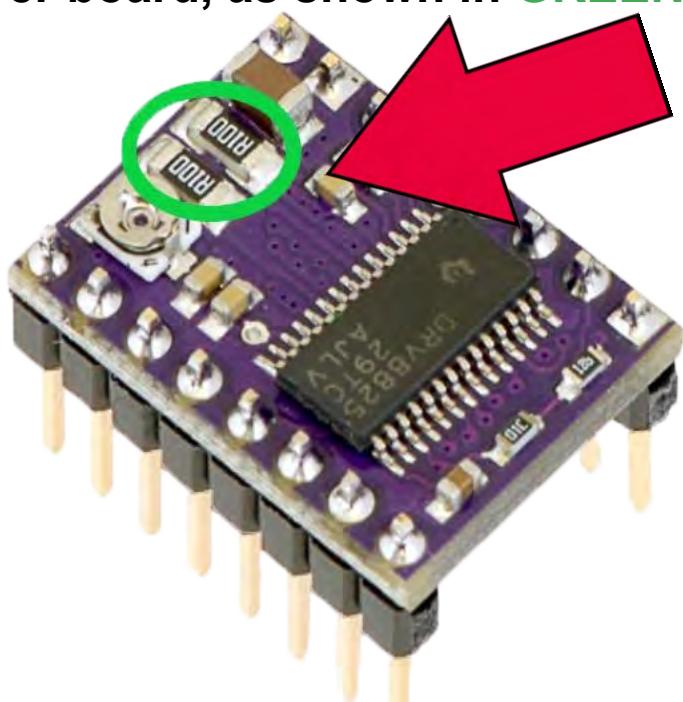
$$V_{ref} = 5 * I_{MAX} * R_S$$

DRV8825

NOTE: Use the potentiometer (POT) on the top of the board (or the board's "V_{ref} Test point") to adjust your V_{ref}. See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board. **Note:** Use 90% of the calculated V_{ref} when tuning the stepper driver board

Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN below.



R_s = R050 is 0.05 Ohms

R_s = R068 is 0.068 Ohms

R_s = R100 is 0.1 Ohms

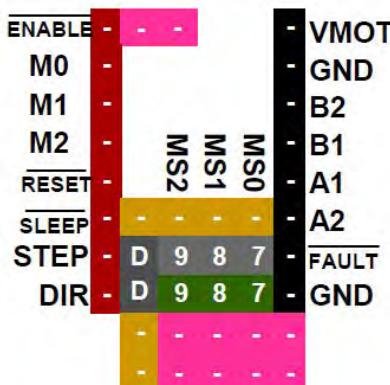
R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

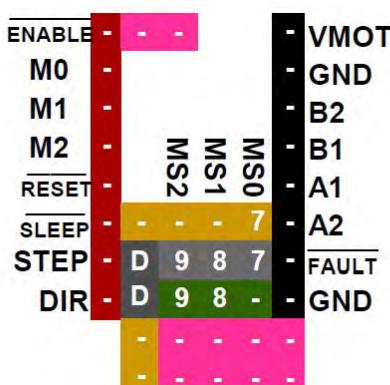
R_s = R220 is 0.22 Ohms

DRV8825

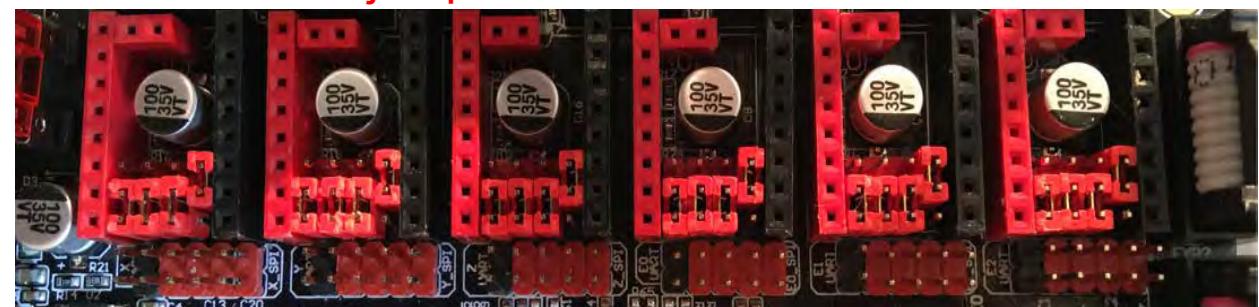
Stand-alone Mode



Note: The "D" jumper MUST be SET!



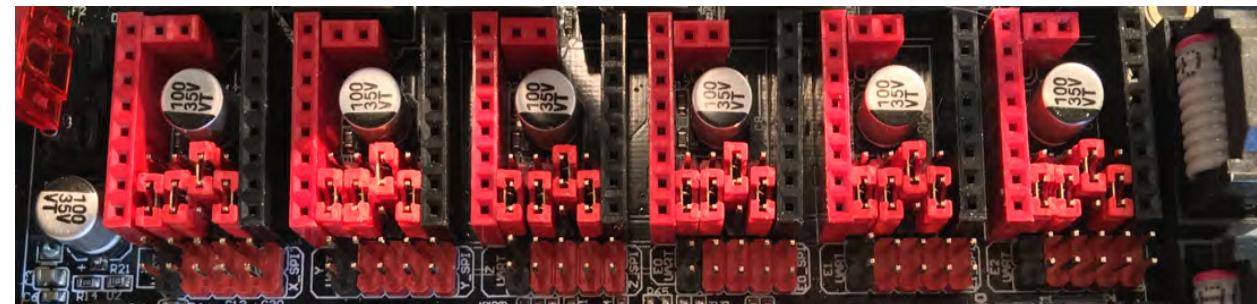
Note: The "D" jumper MUST be SET!



DRV8825**Stand-alone Mode**

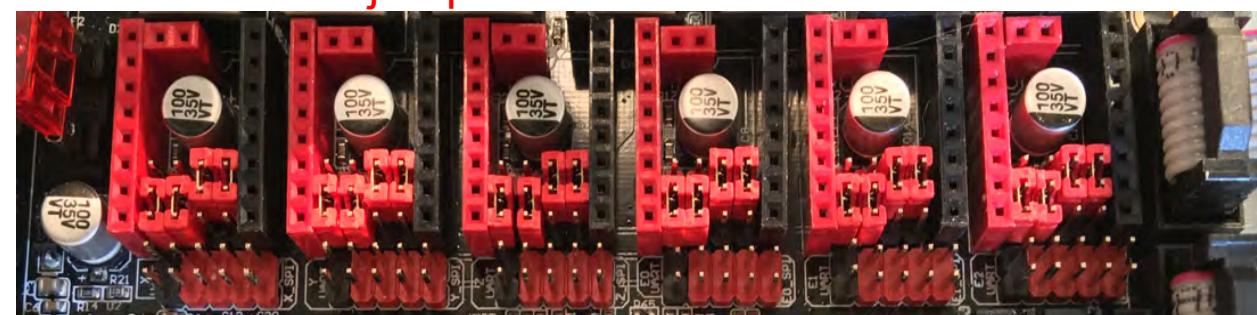
ENABLE	-	-	-	VMOT	
M0	-	-	-	GND	
M1	-	-	-	B2	
M2	-	-	-	B1	
RESET	-	MS2	MS1	MS0	A1
SLEEP	-	-	8	-	A2
STEP	D	9	8	7	FAULT
DIR	D	9	-	7	GND

1 / 4

Note: The "D" jumper MUST be SET!

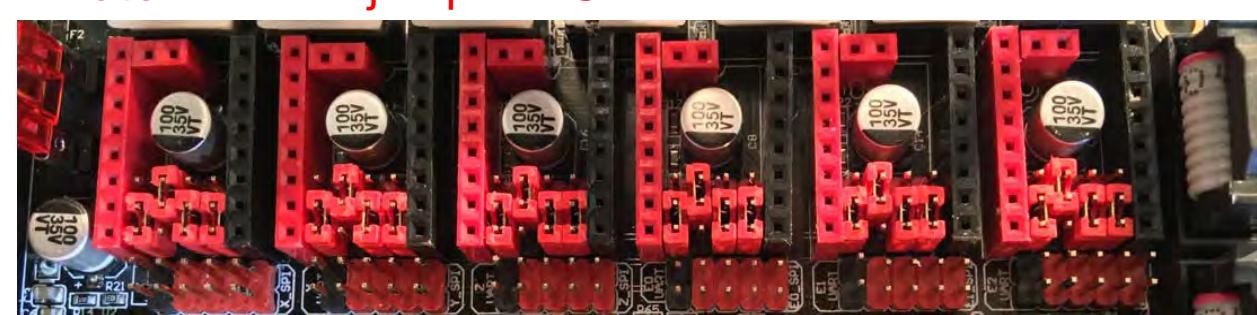
ENABLE	-	-	-	VMOT	
M0	-	-	-	GND	
M1	-	-	-	B2	
M2	-	-	-	B1	
RESET	-	MS2	MS1	MS0	A1
SLEEP	-	-	8	7	A2
STEP	D	9	8	7	FAULT
DIR	D	9	-	7	GND

1 / 8

Note: The "D" jumper MUST be SET!

ENABLE	-	-	-	VMOT	
M0	-	-	-	GND	
M1	-	-	-	B2	
M2	-	-	-	B1	
RESET	-	MS2	MS1	MS0	A1
SLEEP	-	9	-	-	A2
STEP	D	9	8	7	FAULT
DIR	D	-	8	7	GND

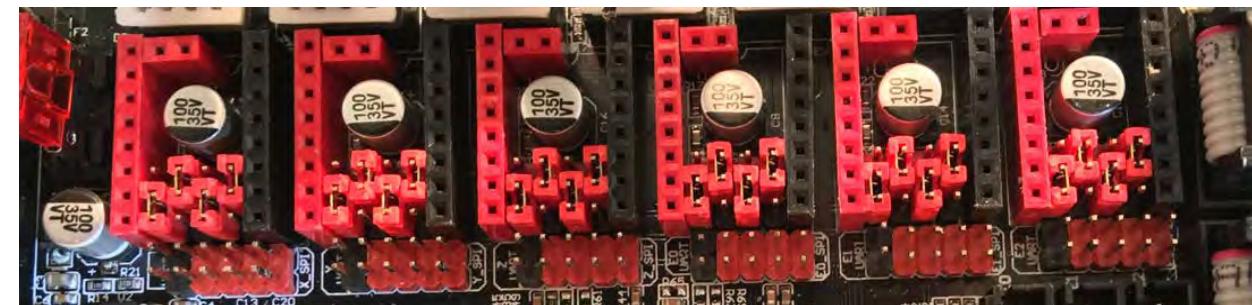
1 / 16

Note: The "D" jumper MUST be SET!

DRV8825**Stand-alone Mode**

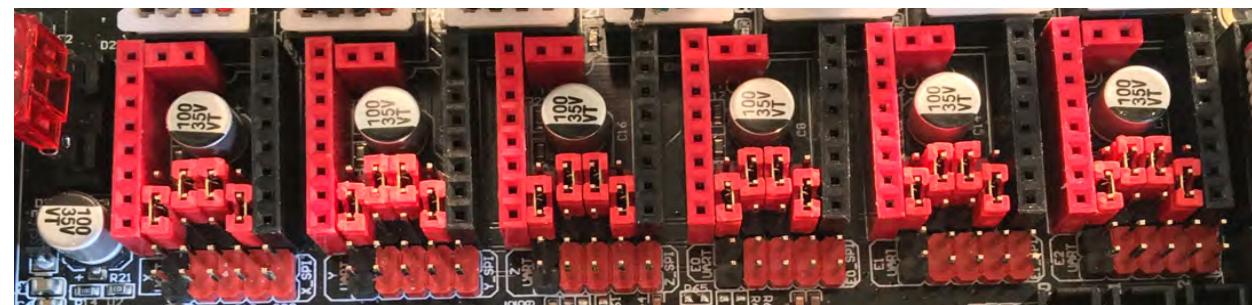
ENABLE	-	-	VMOT
M0	-	-	GND
M1	-	-	B2
M2	-	-	B1
RESET	-	MS2 MS1 MS0	A1
SLEEP	-	9 8 7	A2
STEP	D	9 8 7	FAULT
DIR	D	8	GND

1 / 32

Note: All THREE of these settings will work for 1/32, choose your preference!!**Note:** The "D" jumper MUST be SET!

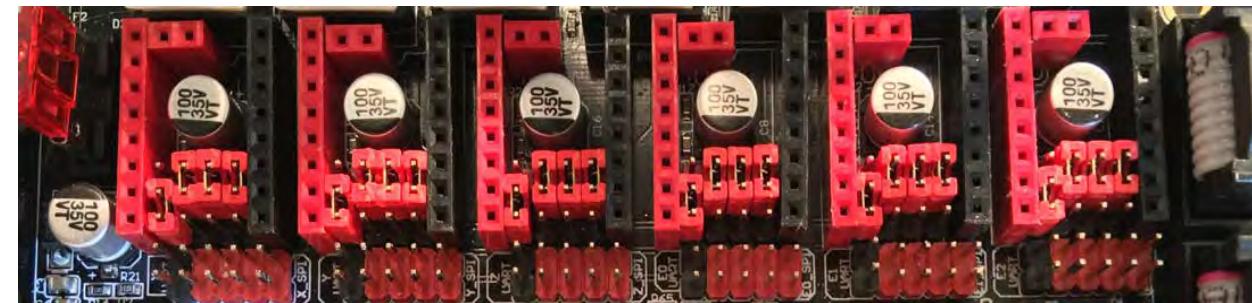
ENABLE	-	-	VMOT
M0	-	-	GND
M1	-	-	B2
M2	-	-	B1
RESET	-	MS2 MS1 MS0	A1
SLEEP	-	9 8 -	A2
STEP	D	9 8 7	FAULT
DIR	D	- 7	GND

1 / 32

Note: The "D" jumper MUST be SET!

ENABLE	-	-	VMOT
M0	-	-	GND
M1	-	-	B2
M2	-	-	B1
RESET	-	MS2 MS1 MS0	A1
SLEEP	-	9 8 7	A2
STEP	D	9 8 7	FAULT
DIR	D	- -	GND

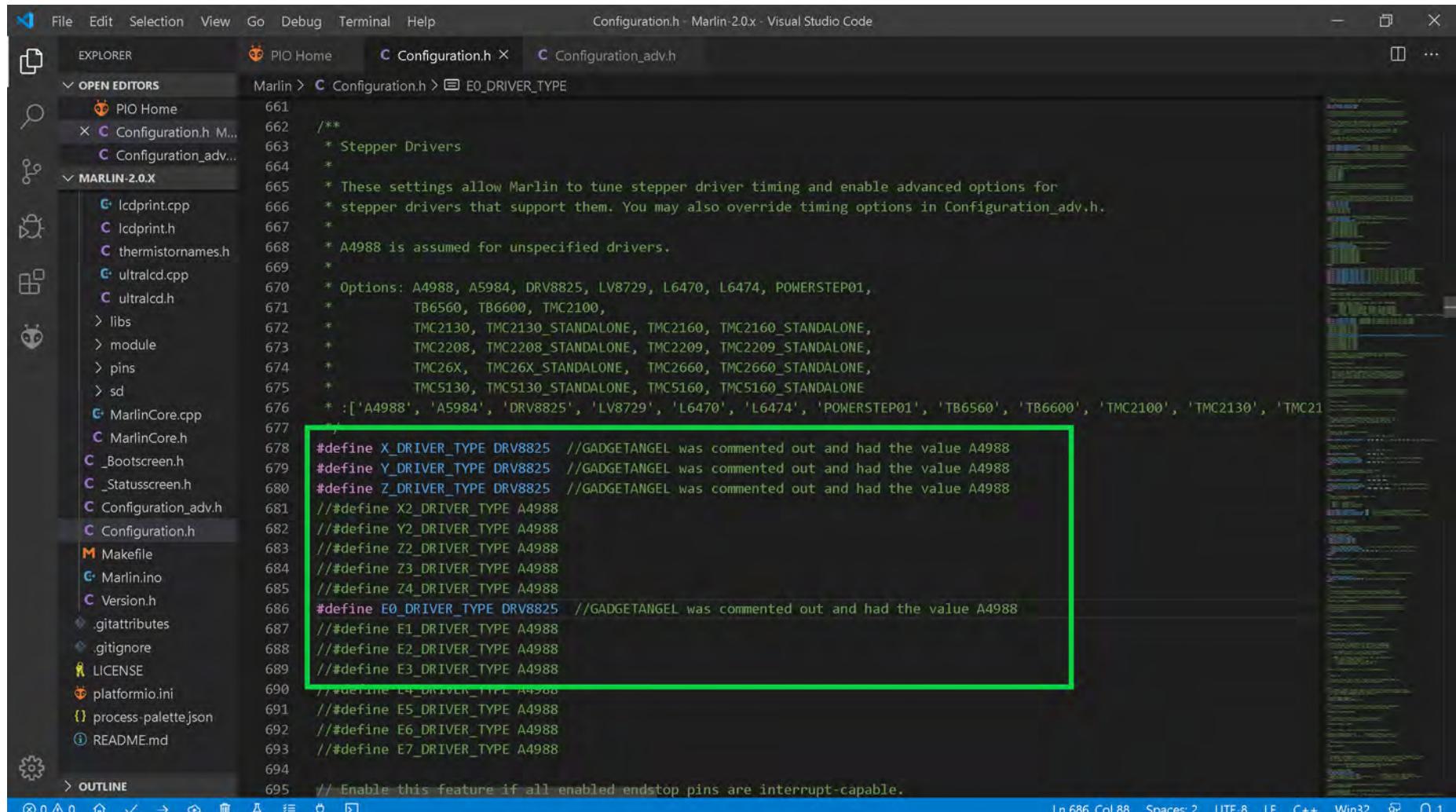
1 / 32

Note: The "D" jumper MUST be SET!

The (latest release of) Marlin Setup for DRV8825 Drivers

NOTE: Go to Appendix C, and 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 ("//").



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

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Editor Title:** Configuration.h - Marlin-2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):**
 - OPEN EDITORS:** PIO Home, Configuration.h (highlighted), Configuration_adv.h
 - 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 (highlighted), Configuration.h (highlighted), Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md
- Editor Content:** The code editor displays the `Configuration.h` file with line numbers 661 to 695. A green rectangular box highlights the driver configuration section starting at line 678. The highlighted code is as follows:

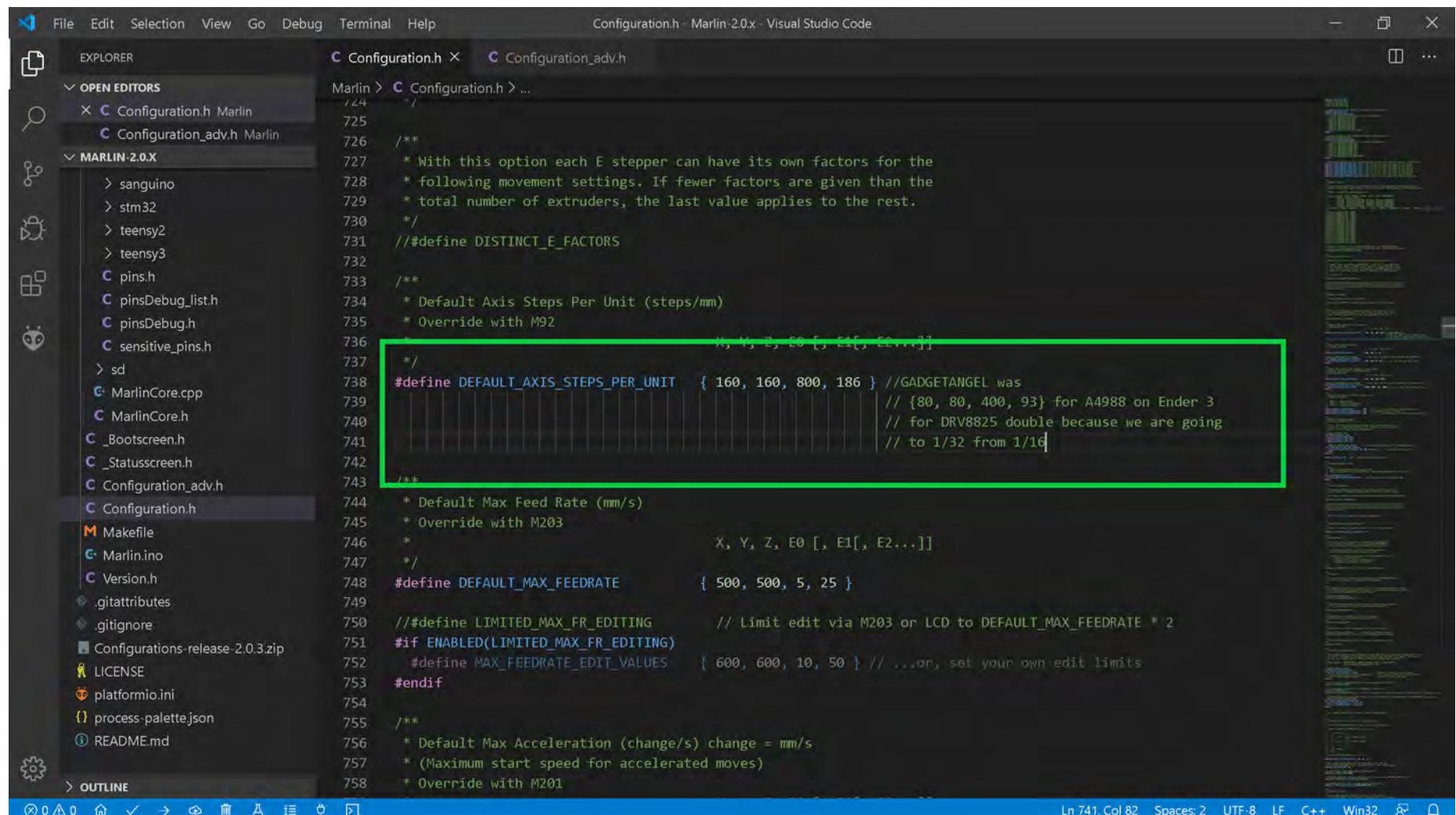
```

#define X_DRIVER_TYPE DRV8825 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE DRV8825 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE DRV8825 //GADGETANGEL was commented out and had the value 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 DRV8825 //GADGETANGEL was commented out and had the value 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
    
```
- Bottom Status Bar:** Ln 686, Col 88, Spaces: 2, UTF-8, LF, C++, Win32, Q 1

- 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.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the Marlin configuration files. A green rectangular box highlights the following line of code:

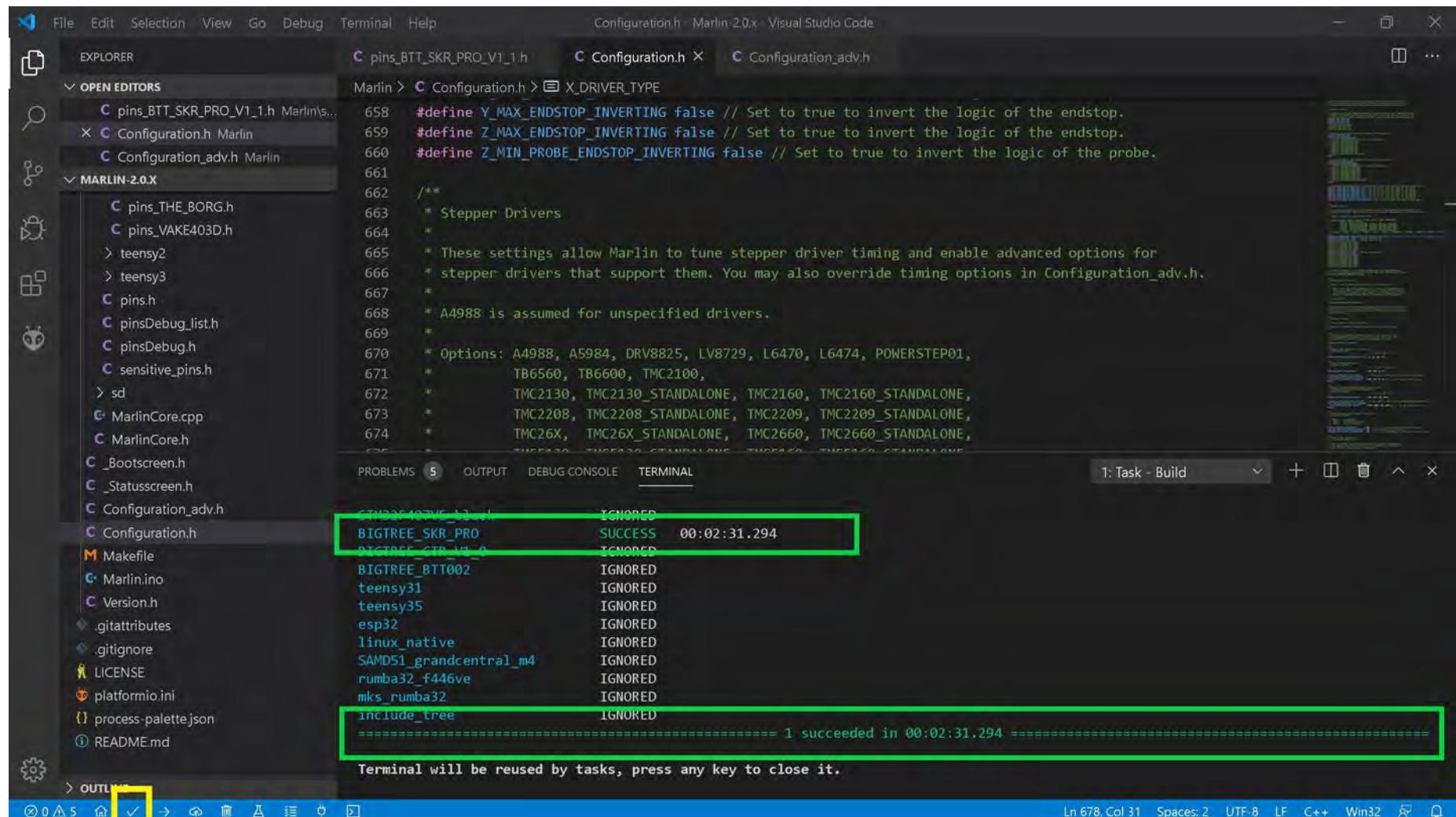
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// for DRV8825 double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current file is `Configuration.h - Marlin 2.0.x - Visual Studio Code`, line 741, column 82, with 2 spaces, in UTF-8 encoding, C++ mode, and Win32 terminal.

- 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.



The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The terminal tab displays the build process:

```

CTM2019TVE_13.sil  IGNORED
BIGTREETECH_SKR_PRO  SUCCESS  00:02:31.294
BIGTREETECH_SKR_V1  IGNORED
BIGTREETECH_BTT002  IGNORED
teensy31  IGNORED
teensy35  IGNORED
esp32  IGNORED
linux_native  IGNORED
SAMDS1_grandcentral_m4  IGNORED
rumba32_f446ve  IGNORED
mks_rumba32  IGNORED
include_tree  IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

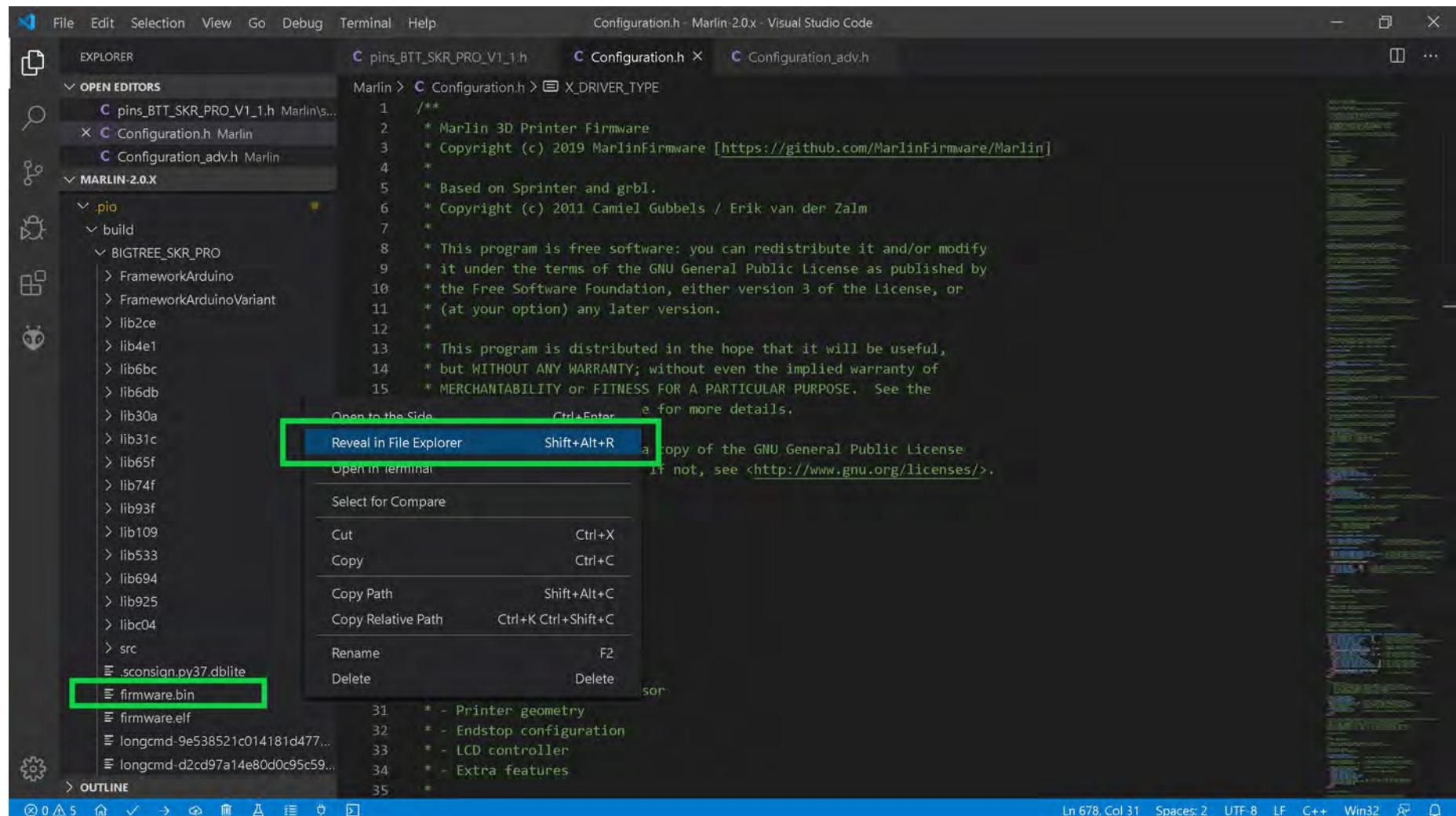
```

A yellow box highlights the checkmark icon in the bottom-left corner of the terminal tab bar, 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 DRV8825 Drivers

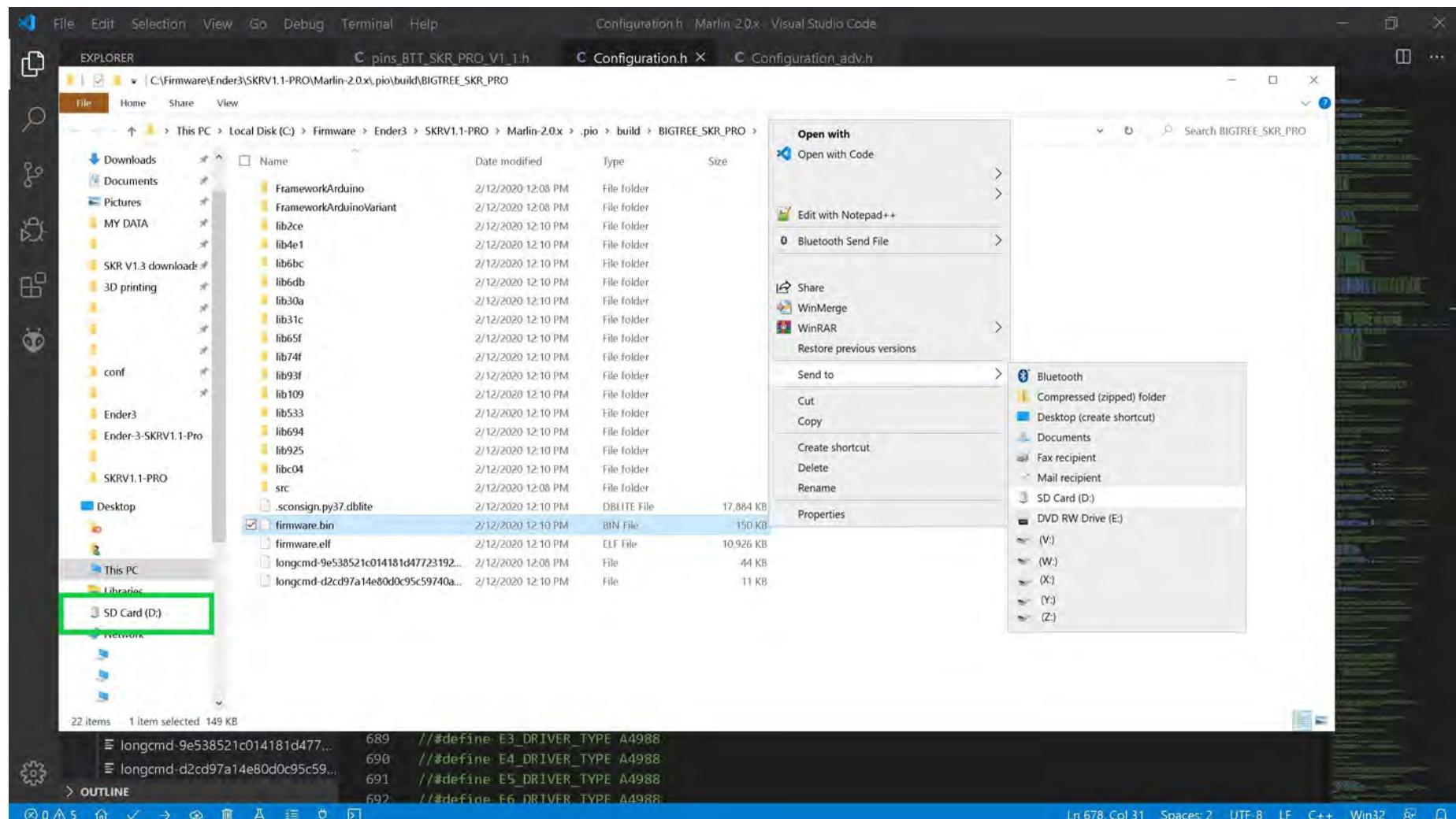
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



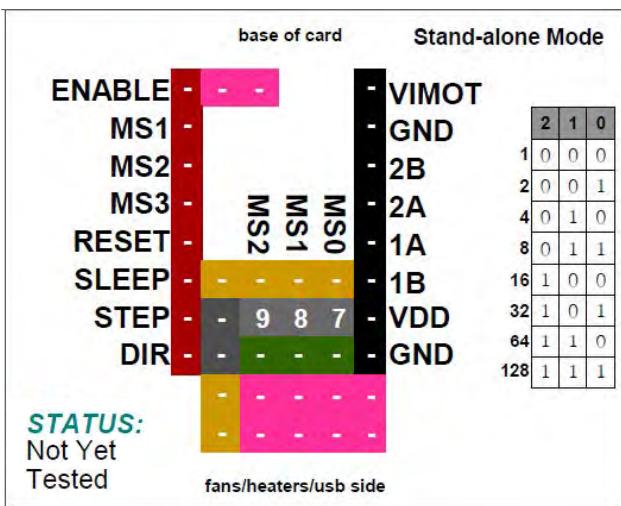
- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

- Ensure your micro SD card, which 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".



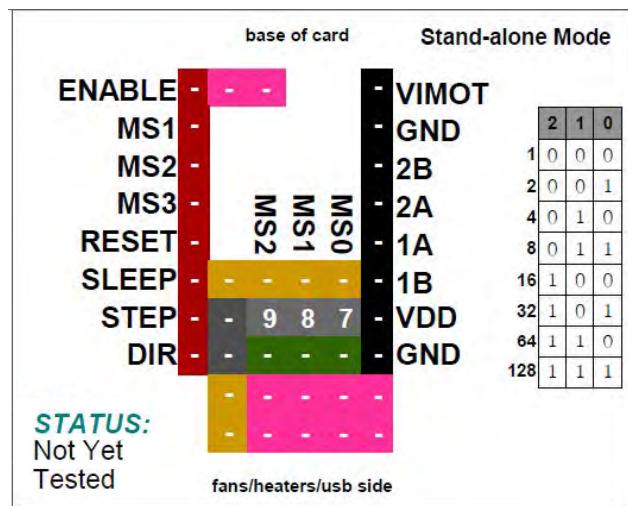
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



BIQU LV8729

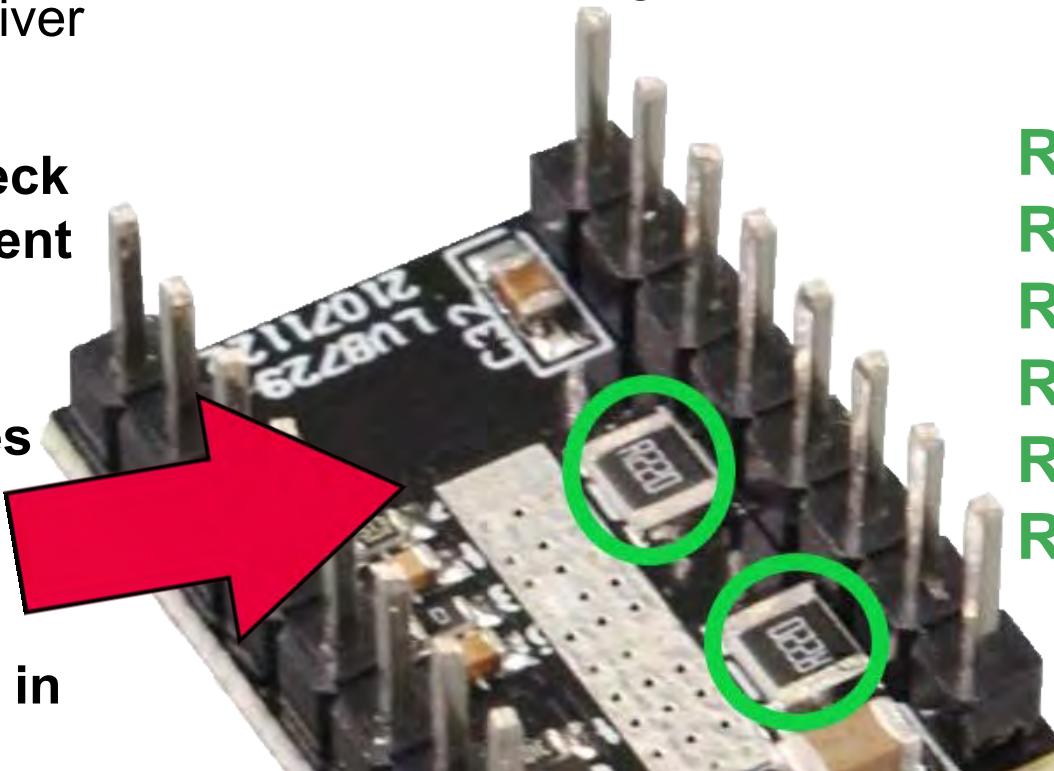
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
BIQU® LV8729 Maximum 128 Subdivision 36V DC 1.5A (peak)	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	High	High	Low	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S (Typical Sense Resistor)=0.22Ω	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$			$V_{ref} = 5 * I_{MAX} * R_S$	



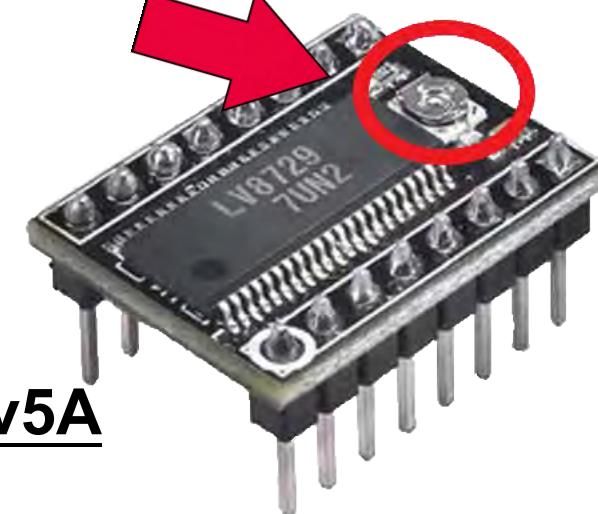
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



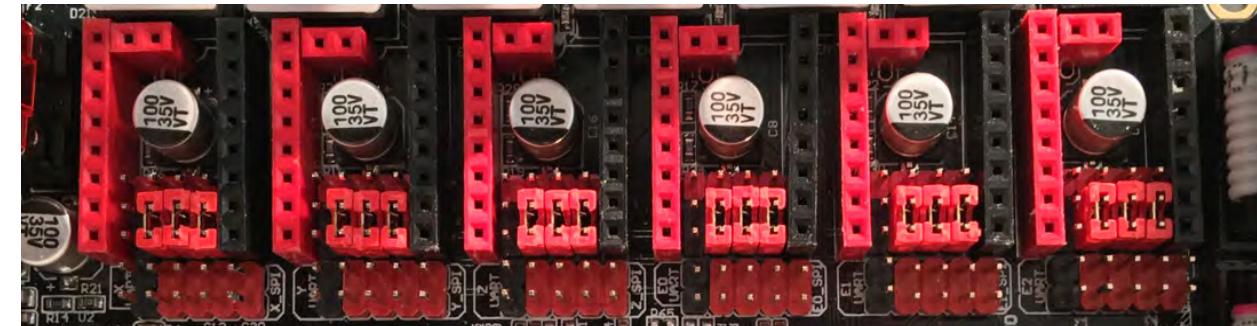
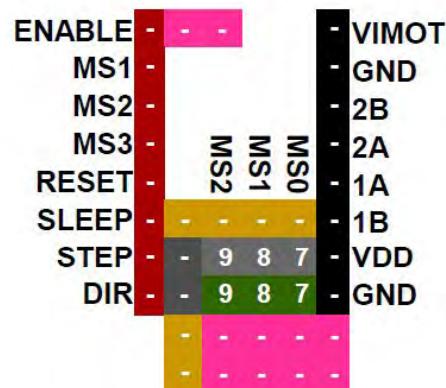
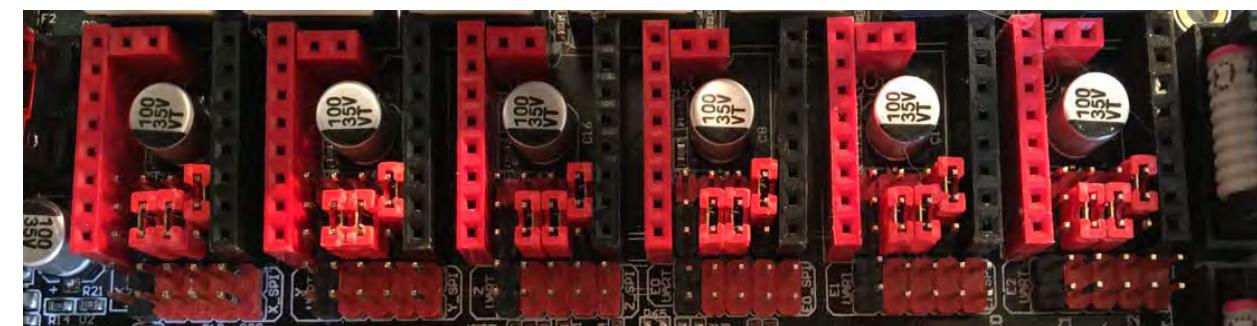
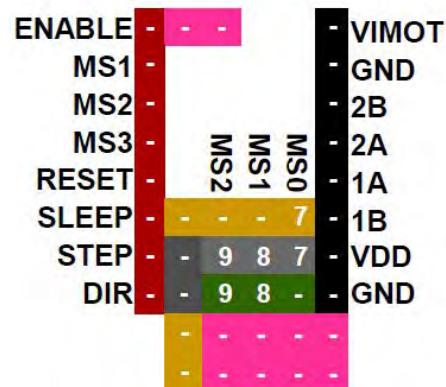
BIQU LV8729

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

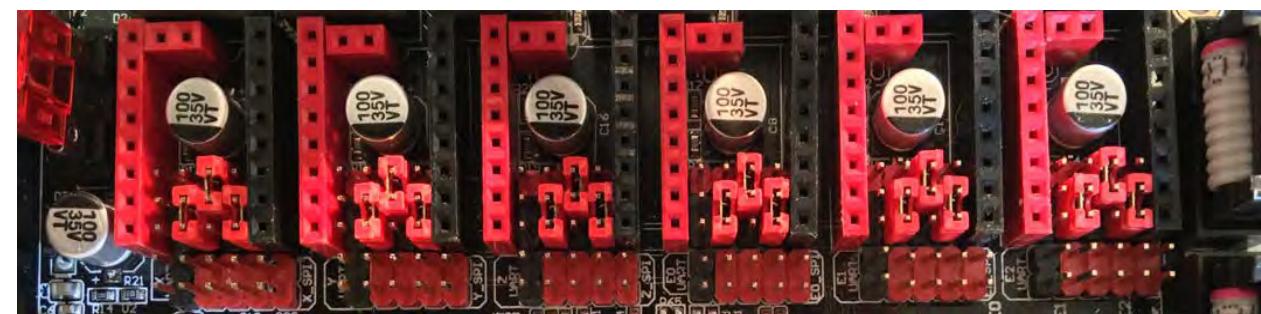
$R_s = R050$ is 0.05 Ohms
 $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms
 $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms
 $R_s = R220$ is 0.22 Ohms

BIQU LV8729**Stand-alone Mode****STEP****1 / 2**

BIQU LV8729**Stand-alone Mode**

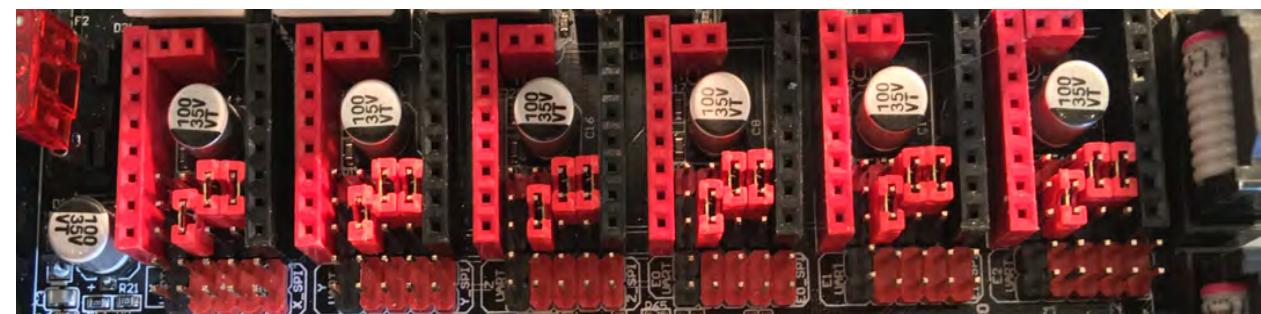
1 / 4

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	8	1B
STEP	9 8 7	VDD
DIR	9 - 7	GND



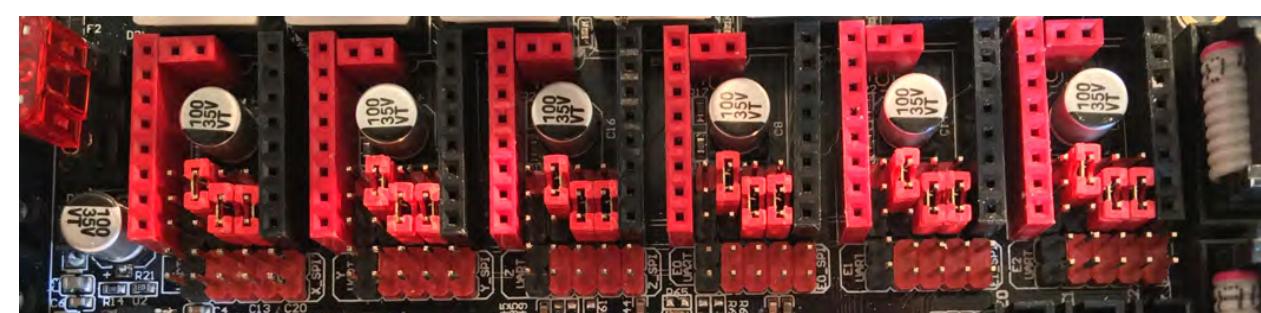
1 / 8

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	8 7	1B
STEP	9 8 7	VDD
DIR	9 -	GND



1 / 16

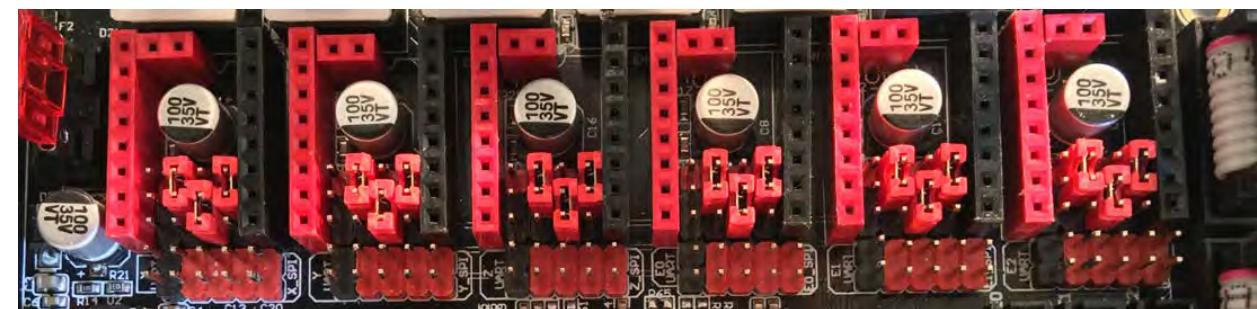
ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	9 - -	1B
STEP	9 8 7	VDD
DIR	- 8 7	GND



BIQU LV8729**Stand-alone Mode**

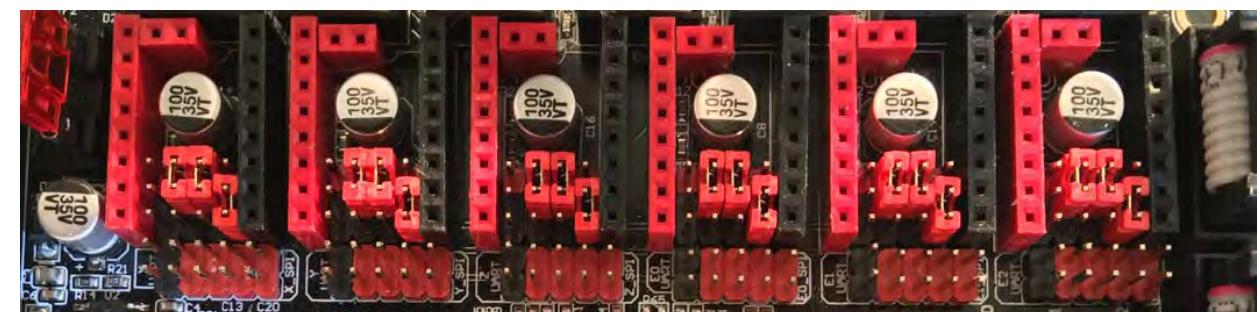
1 / 32

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 7	1B
STEP	9 8 7	VDD
DIR	- 8 -	GND



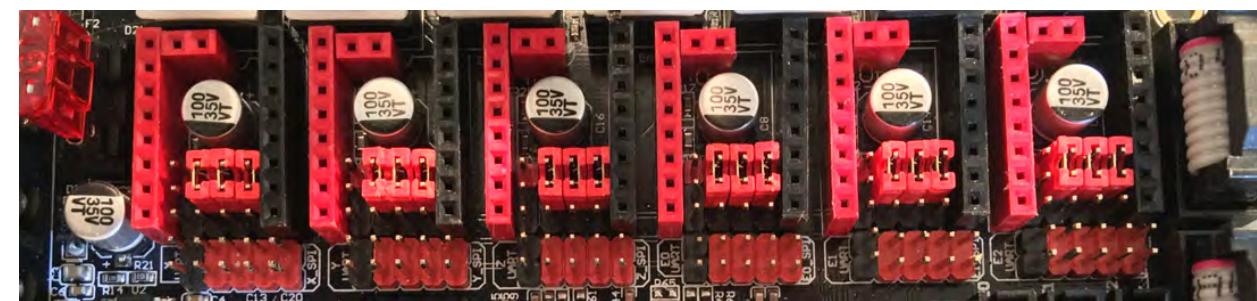
1 / 64

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 8 -	1B
STEP	9 8 7	VDD
DIR	- - 7	GND



1 / 128

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 8 7	1B
STEP	9 8 7	VDD
DIR	- - -	GND



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

NOTE: Go to Appendix C, and 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 ("//").

```

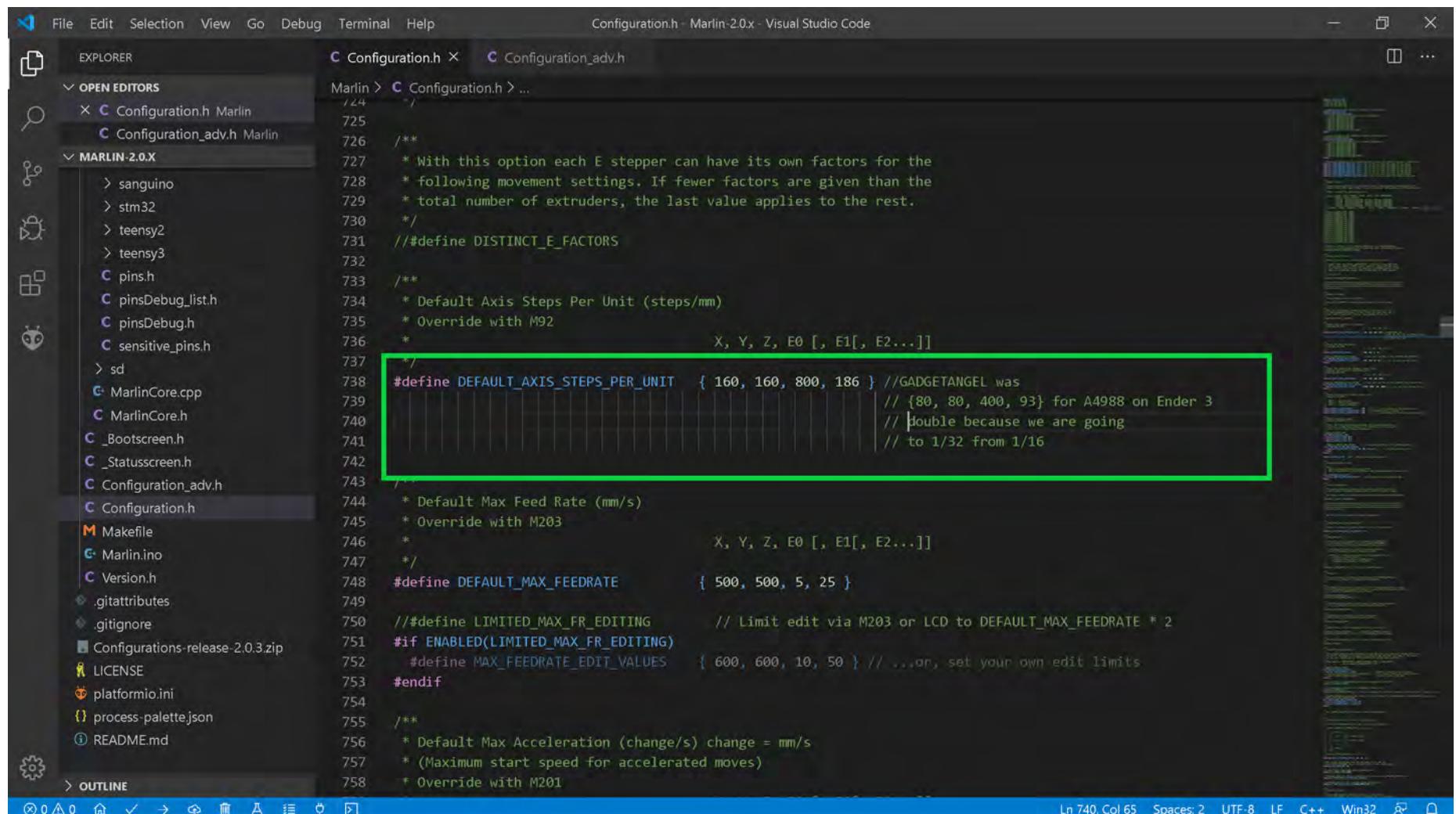
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
MARLIN-2.0.X
Lcdprint.cpp
Lcdprint.h
thermistornames.h
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
Ln 686, Col 87 Spaces: 2 UTF-8 LF C++ Win32 1
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', 'TMC2
676 */
677 #define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value 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 LV8729 //GADGETANGEL was commented out and had the value 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.

```

- Go to the next page.

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

- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 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.



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 line of code:

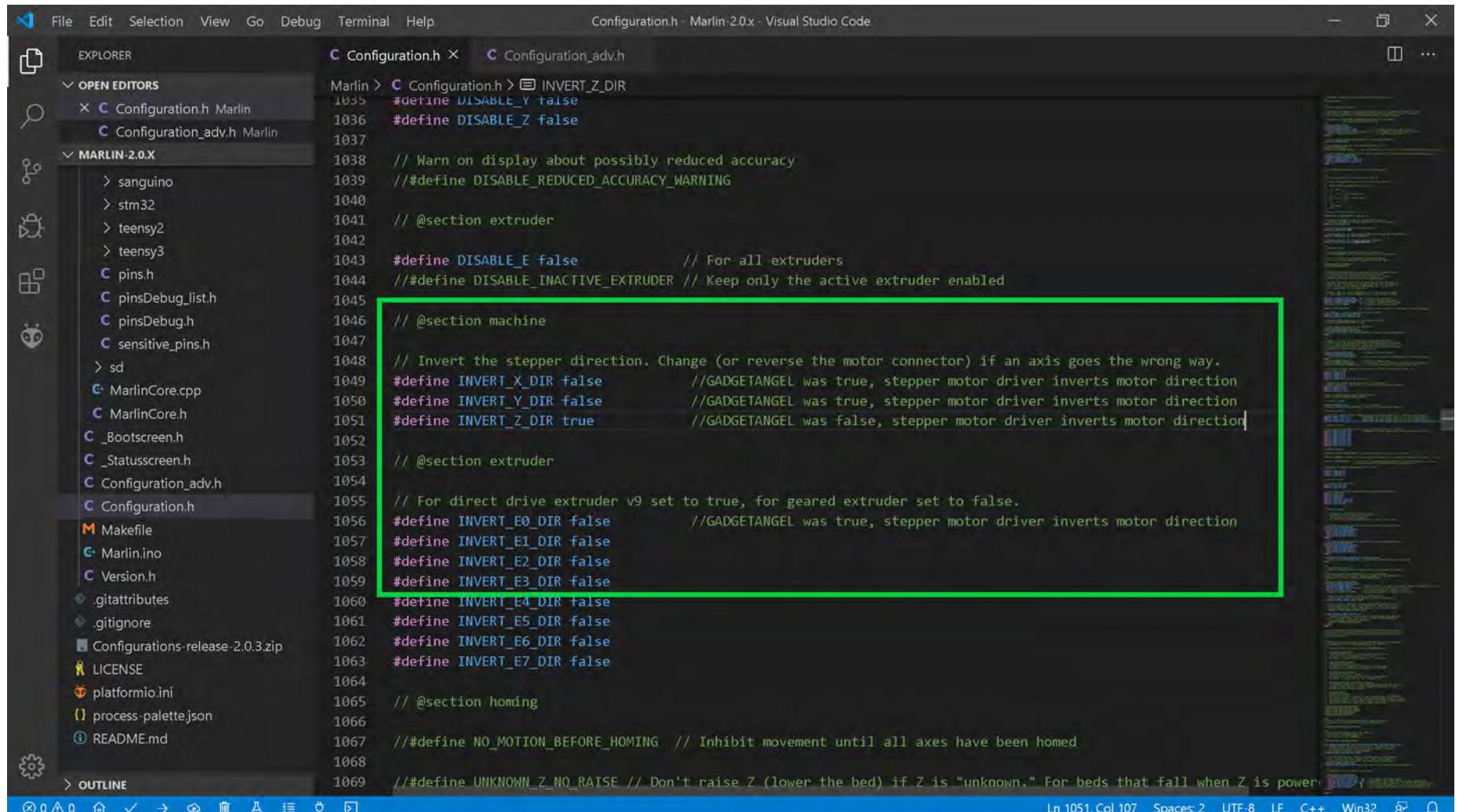
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current line (Ln 740), column (Col 65), and other settings like spaces (Spaces: 2), encoding (UTF-8), line endings (LF), and file type (C++). The left sidebar shows the project structure with various Marlin source files and configuration files.

- 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



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

EXPLORER OPEN EDITORS Configuration.h Configuration_adv.h

```

Marlin > Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060
1061 #define INVERT_E4_DIR false
1062 #define INVERT_E5_DIR false
1063 #define INVERT_E6_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered on

```

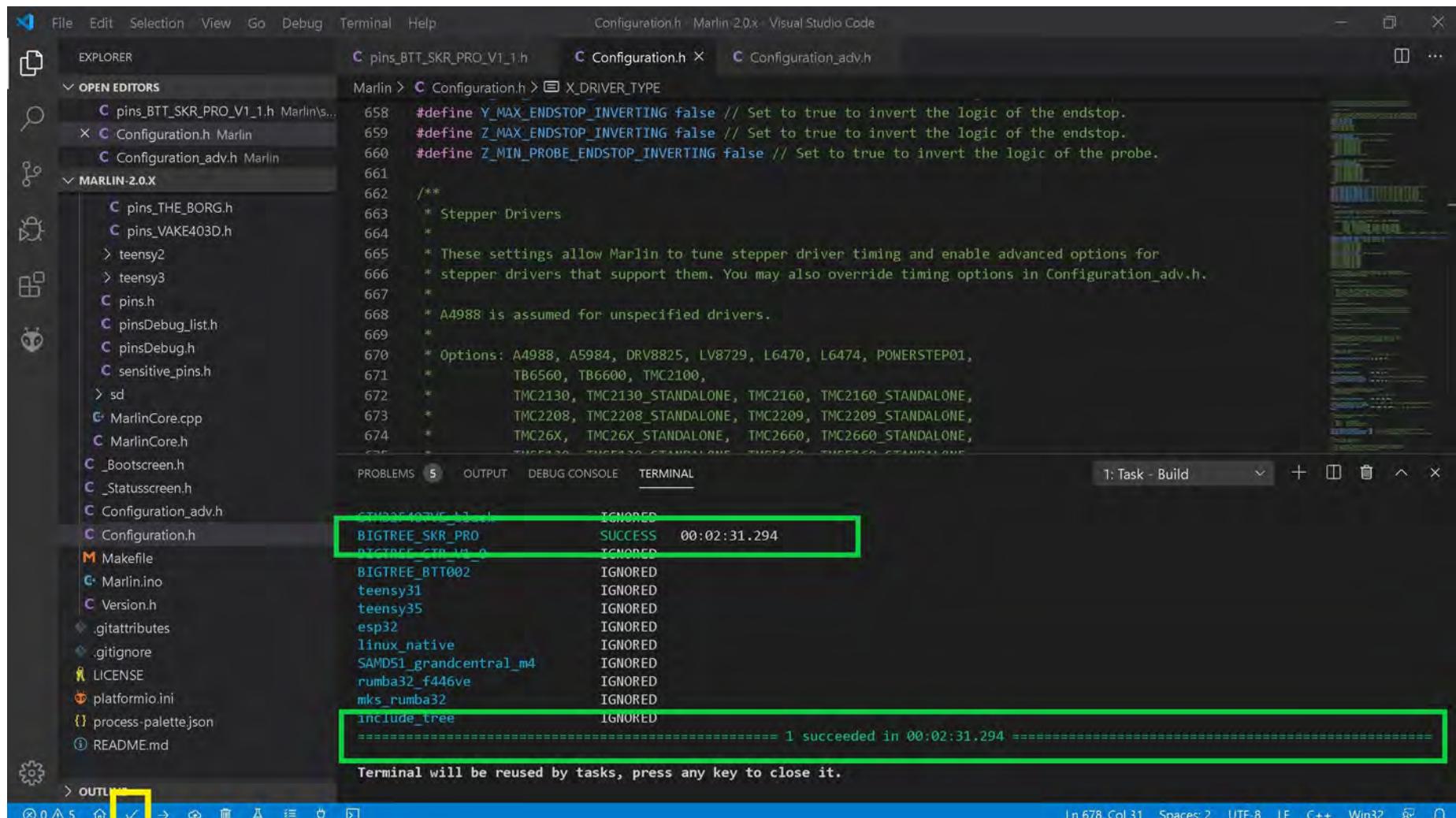
OUTLINE

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- 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:

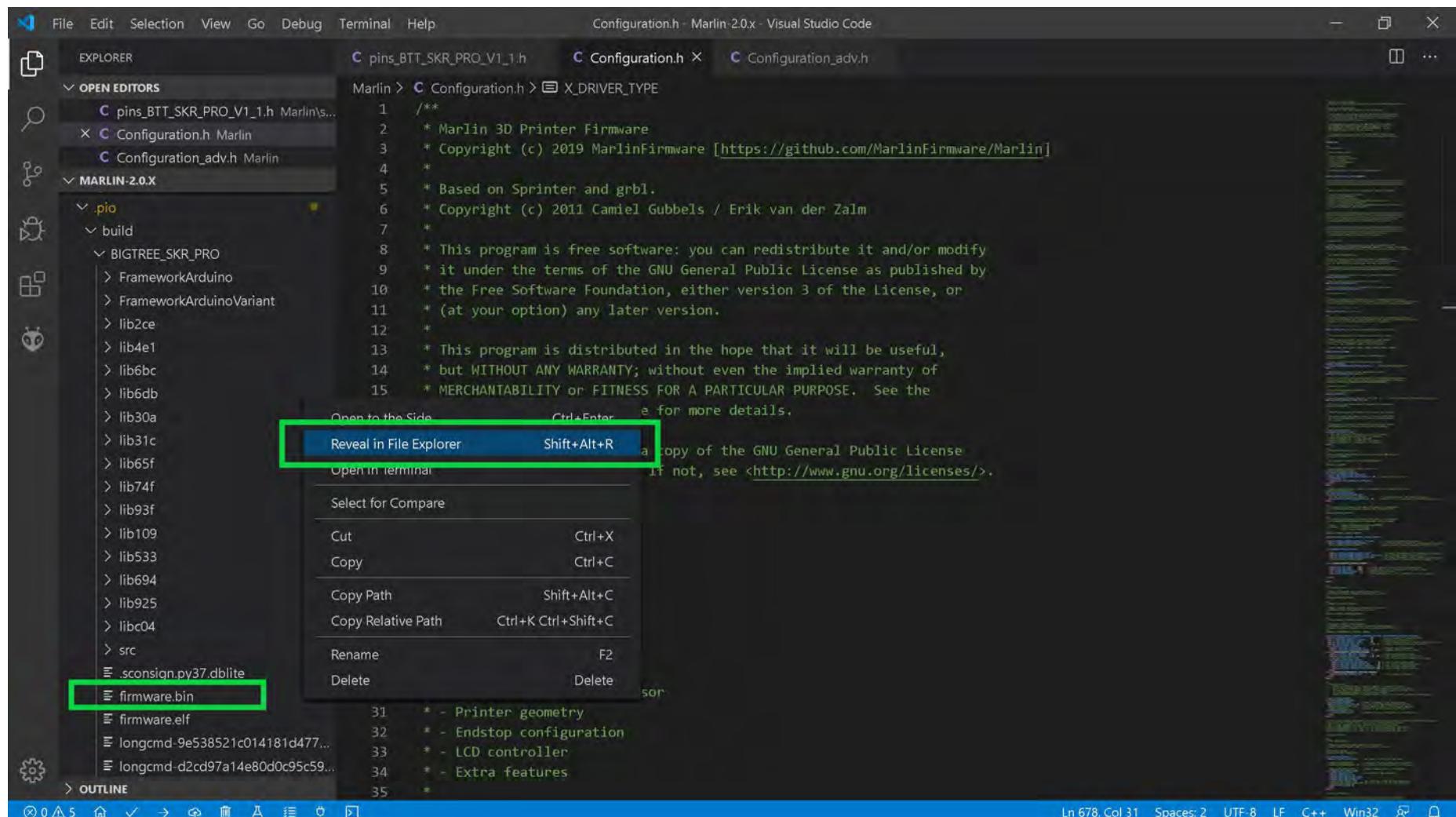
- File Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editors:** Three files are open: Configuration.h, Configuration_adv.h, and pins_BTT_SKR_PRO_V1_1.h.
- Terminal:** The terminal tab is active, showing the build log for the "BIGTREE_SKR_PRO" target. The log includes the following output:


```
STM32F407VE-L1-I BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_CTP_V5.0 IGNORED
BIGTREE_BT1002 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:** Shows the terminal will be reused by tasks, and the status bar indicates: 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

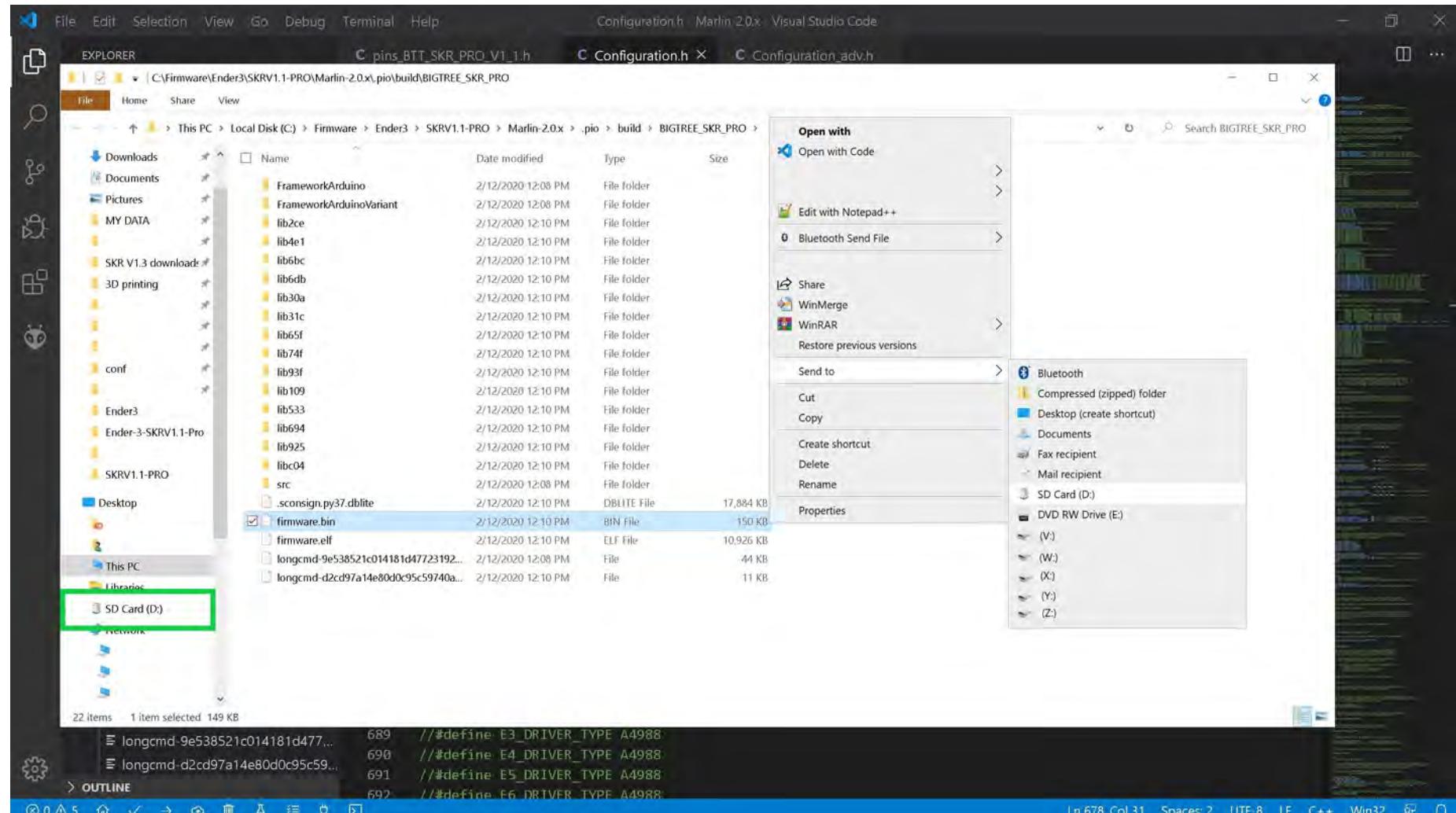
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

	base of card	Stand-alone Mode
EN		VM
MS1		GND
MS2		1A
MS3		1B
RESET	MS2	2A
RESET	MS1	2B
STEP	MS0	VDD
DIR	9 8 7	GND
STATUS:	Not Yet Tested	
	fans/heaters/usb side	

FYSETC LV8729

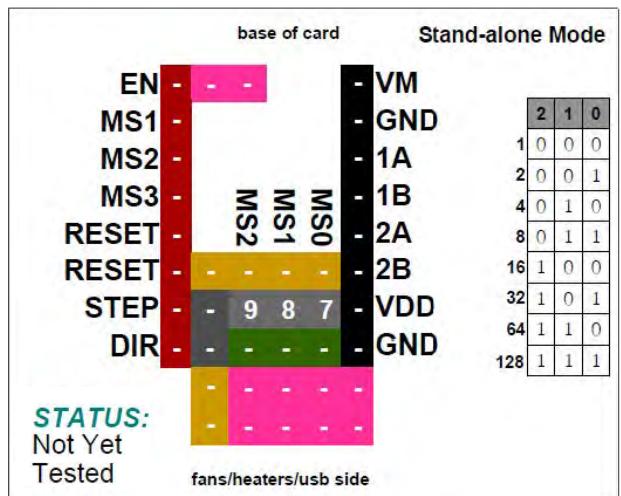
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
FYSETC LV8729 Maximum 128 Subdivision 36V DC 1.5A (peak)	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	High	High	Low	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula
R_S (Typical Sense Resistor)=0.22Ω

$$I_{MAX} = \frac{V_{ref}}{5 * R_S}$$

$$V_{ref} = 5 * I_{MAX} * R_S$$

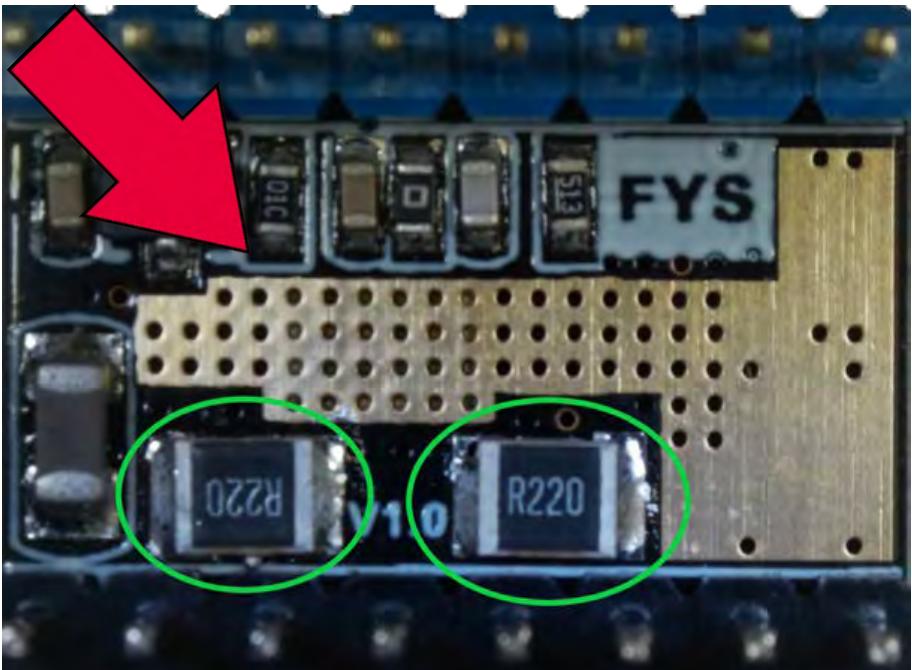


FYSETC LV8729

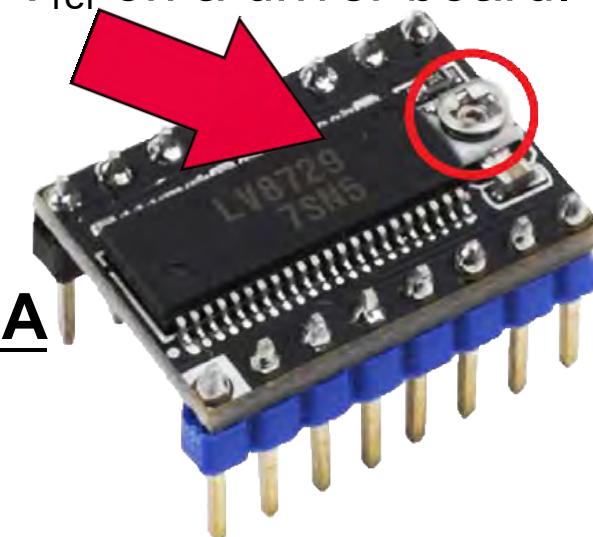
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in **GREEN**



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>



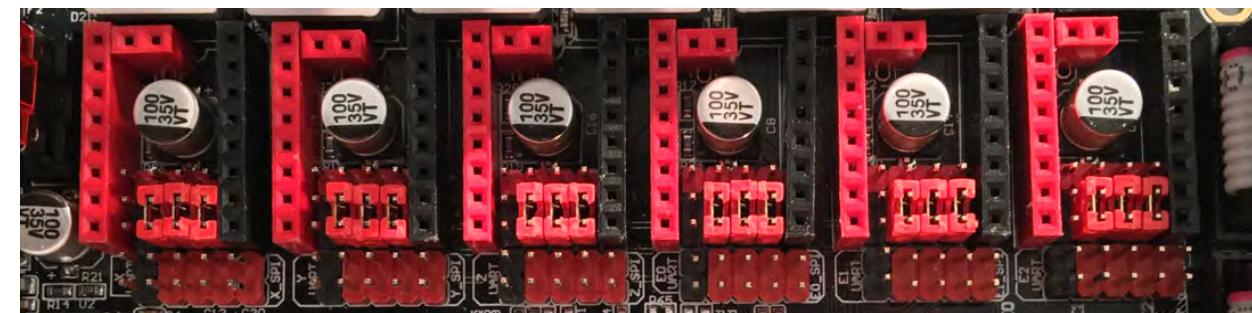
$R_s = R050$ is 0.05 Ohms;
 $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms;
 $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms;
 $R_s = R220$ is 0.22 Ohms

FYSETC LV8729

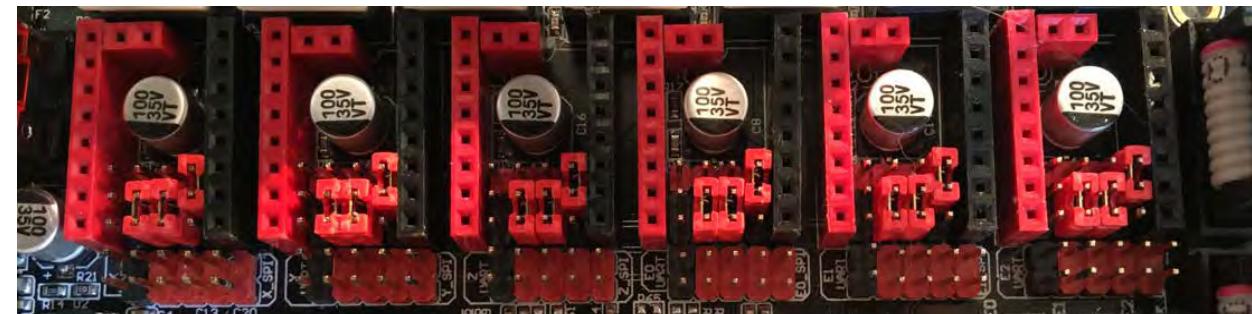
Stand-alone Mode

STEP

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	2A
RESET	-	-	-	2B
STEP	-	9	8	7
DIR	-	9	8	7
				VDD
				GND

**1 / 2**

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	2A
RESET	-	-	-	2B
STEP	-	9	8	7
DIR	-	9	8	-
				VDD
				GND

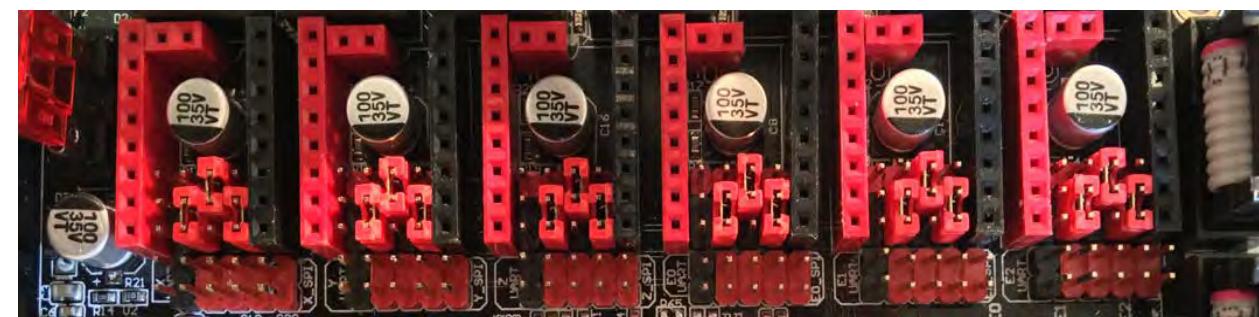


FYSETC LV8729

Stand-alone Mode

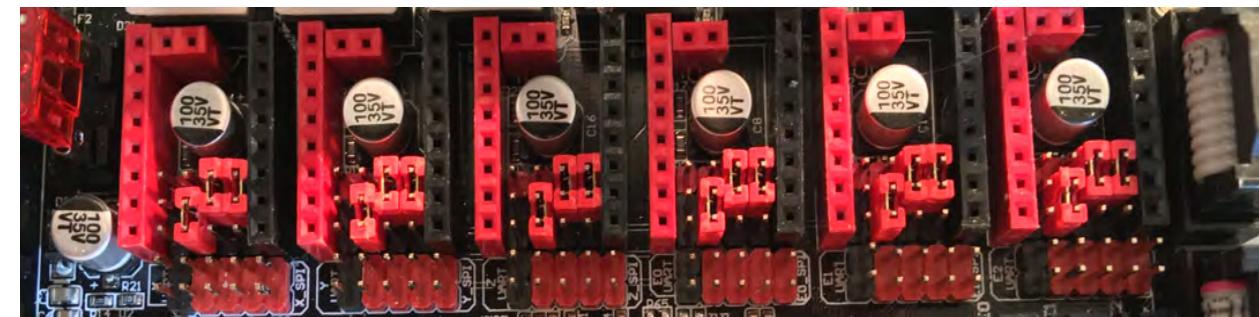
1 / 4

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	2A
RESET	-	8	-	2B
STEP	-	9	8	VDD
DIR	-	9	7	GND



1 / 8

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	2A
RESET	-	8	7	2B
STEP	-	9	8	VDD
DIR	-	9	-	GND



1 / 16

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	2A
RESET	-	9	-	2B
STEP	-	9	8	VDD
DIR	-	-	8	GND

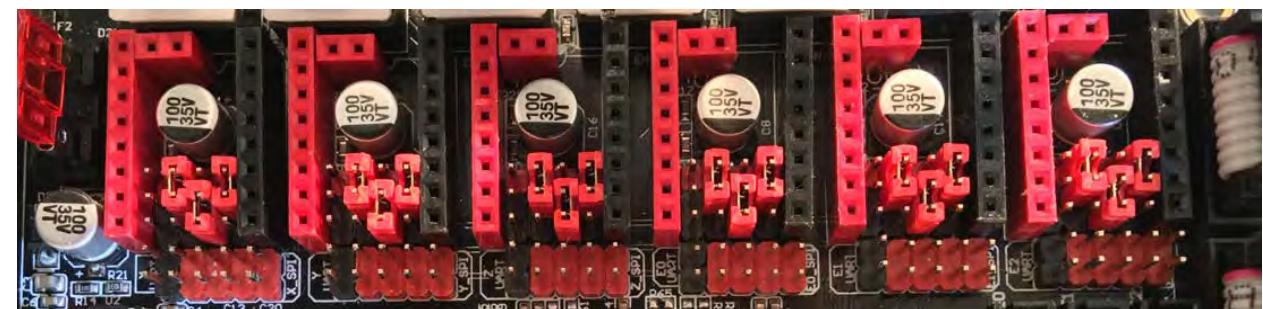


FYSETC LV8729

Stand-alone Mode

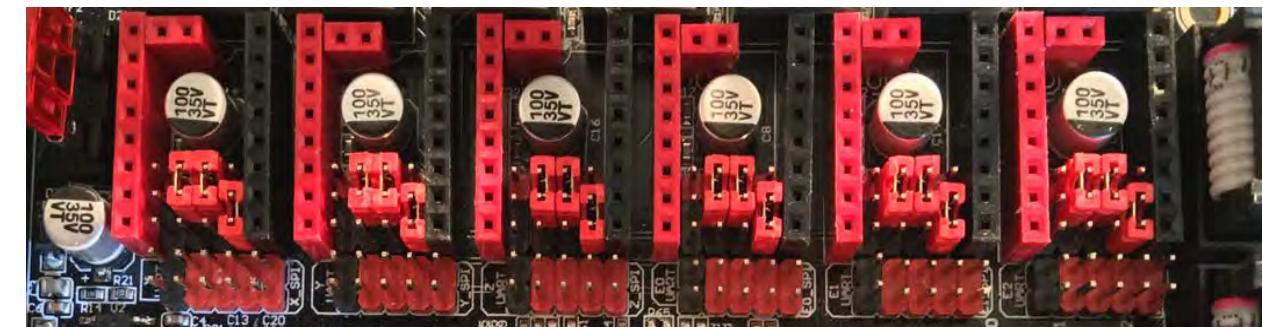
1 / 32

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
RESET	-	9	7	2A
STEP	-	9	8	7
DIR	-	-	8	GND



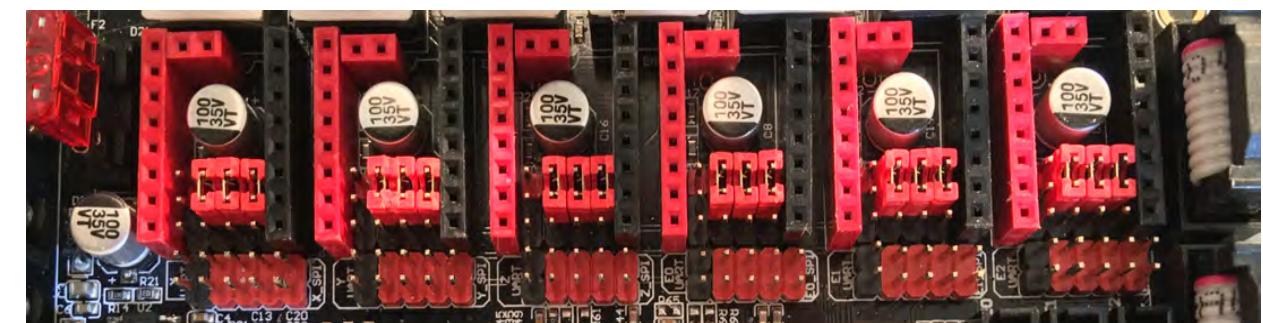
1 / 64

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
RESET	-	9	8	2A
STEP	-	9	8	7
DIR	-	-	7	GND



1 / 128

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
RESET	-	9	8	7
STEP	-	9	8	7
DIR	-	-	-	GND



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

NOTE: Go to Appendix C, and 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
Marlin > Configuration.h > E0_DRIVER_TYPE

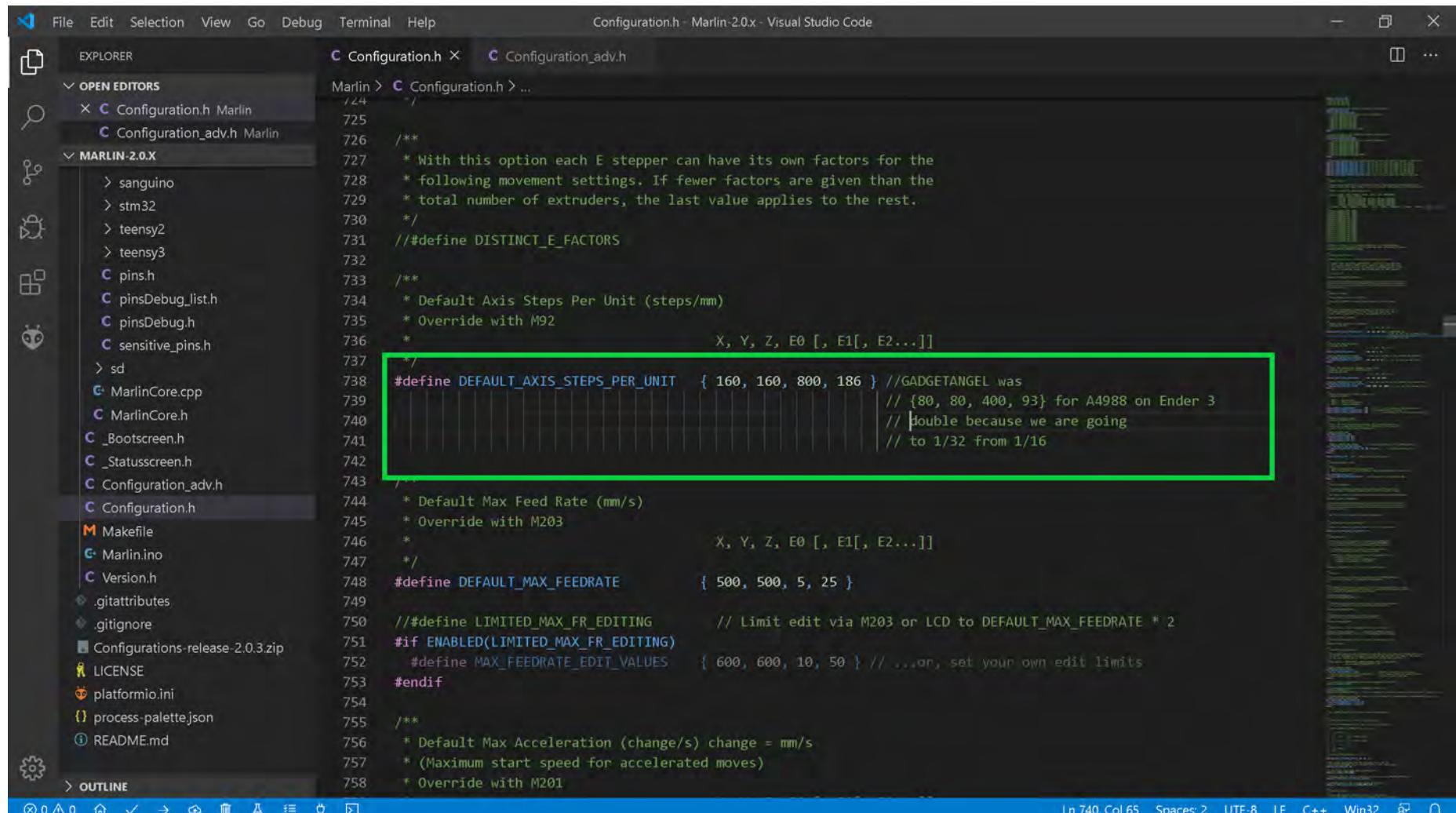
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: 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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
677
#define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value 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
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
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 LV8729 Drivers

- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 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.



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 line of code:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- 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

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

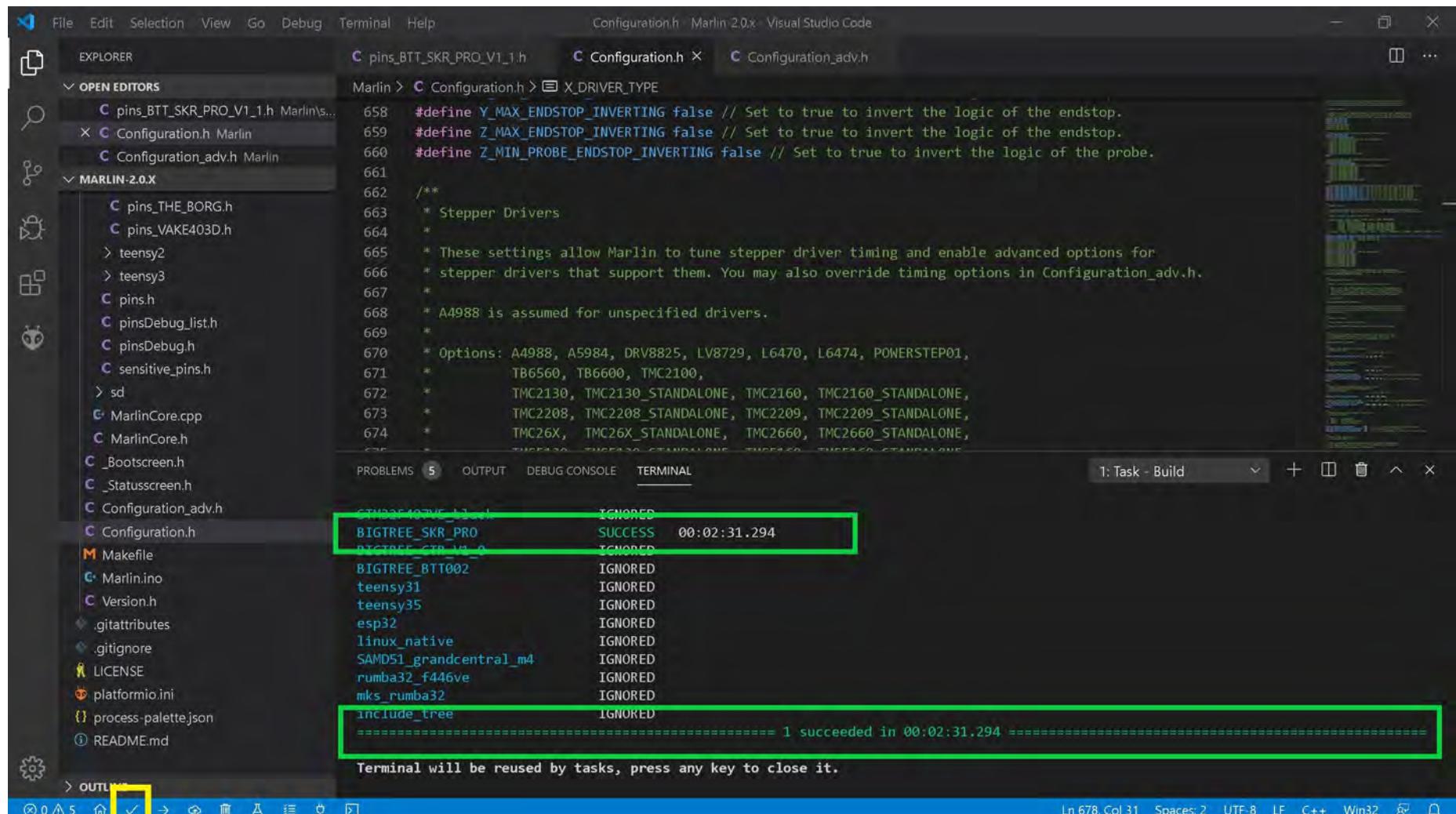
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):** Shows the project structure under MARLIN-2.0.X, including files like Configuration.h, Configuration_adv.h, pins.h, and various MarlinCore files.
- Central Area:** The code editor displays Configuration.h with the following content:

```
Marlin > Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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    // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```
- Right Sidebar:** Shows a vertical stack of code snippets or preview windows.

- 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
Configuration.h X Configuration_adv.h

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 + ×
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_PRO IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMDS1_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.

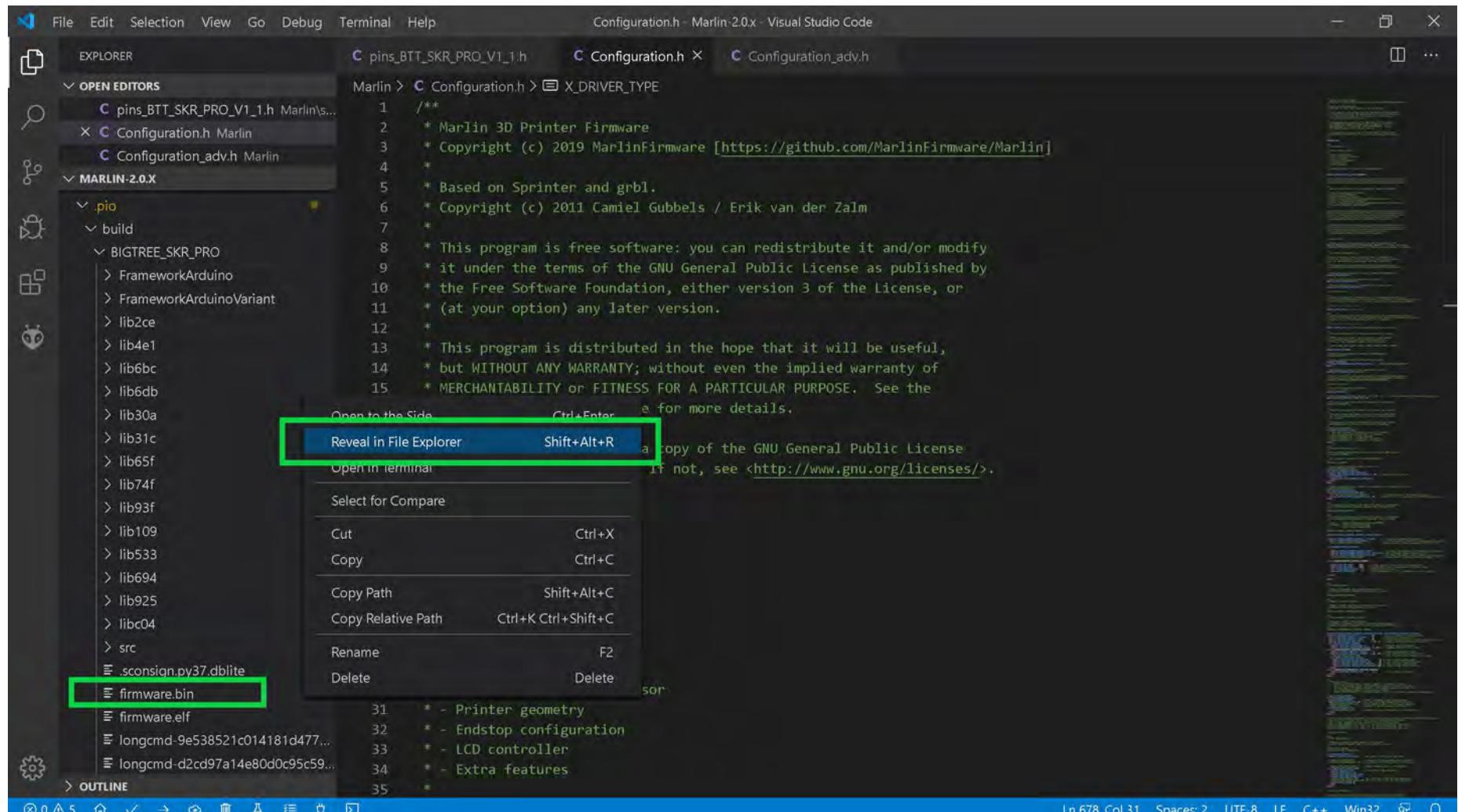
```

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

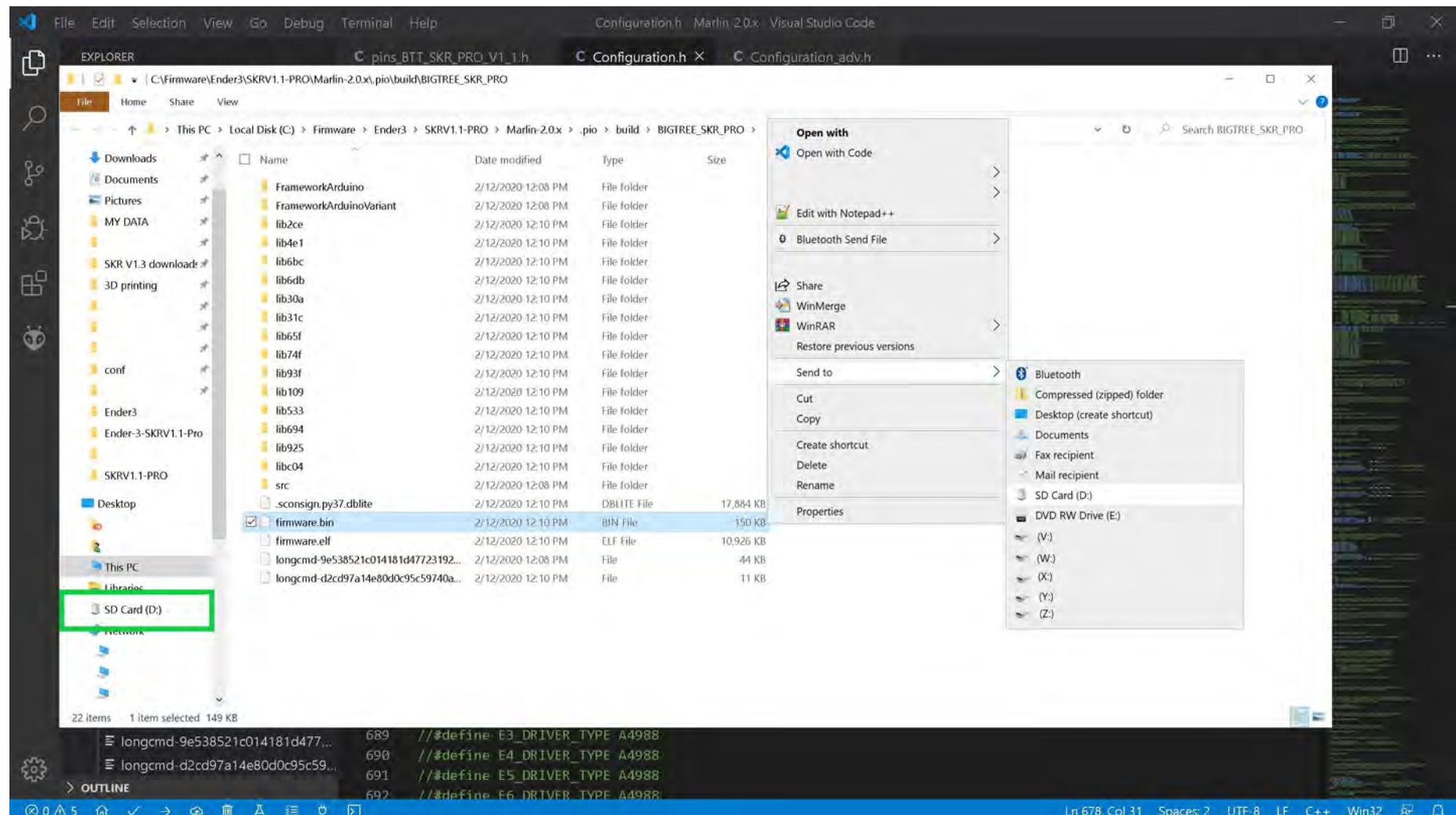
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



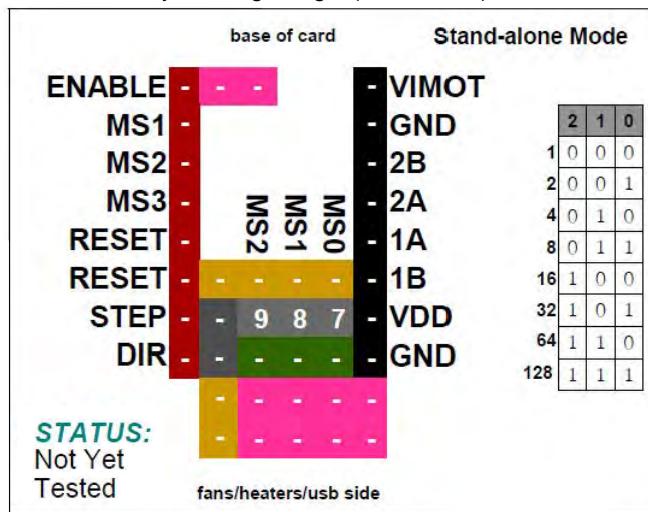
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

LERDGE LV8729

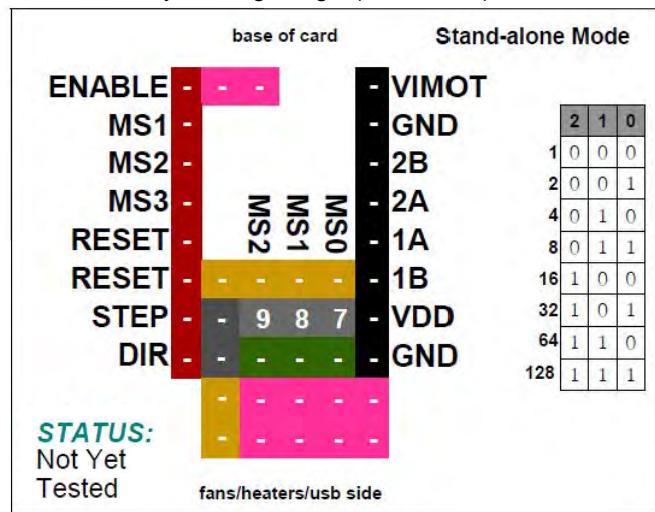
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
LV8729 Maximum 128 Subdivision 36V DC 1.5A (peak)	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	High	High	Low	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula
R_S (Typical Sense Resistor) = 0.22Ω

$$I_{MAX} = \frac{V_{ref}}{5 * R_S}$$

$$V_{ref} = 5 * I_{MAX} * R_S$$

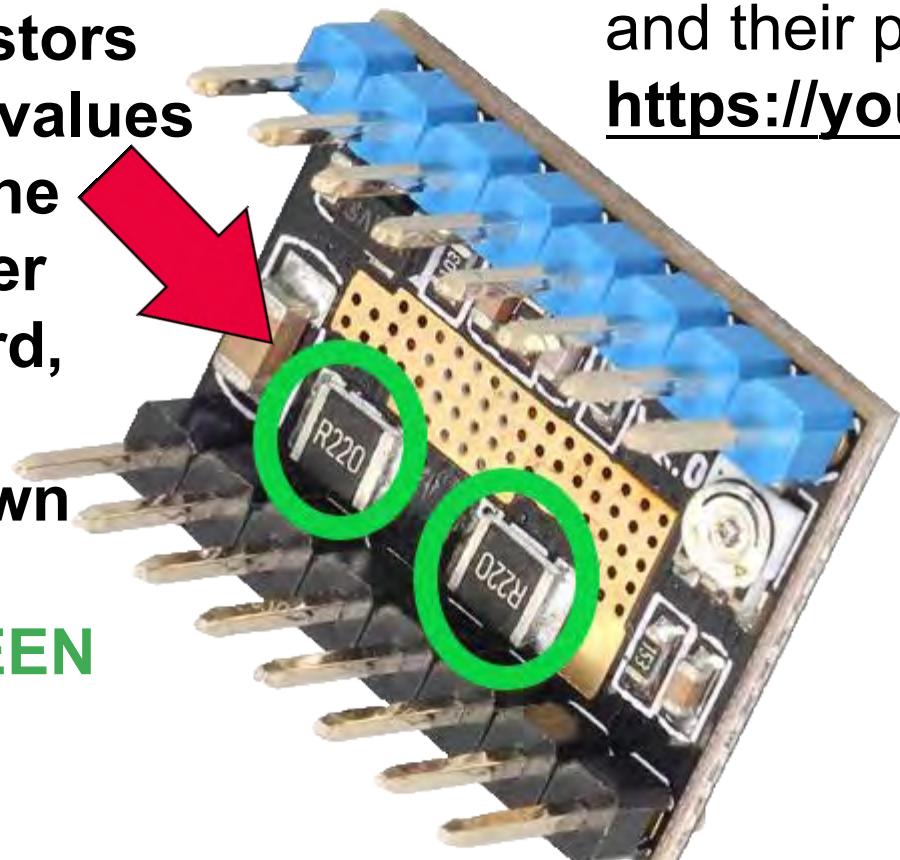


LERDGE LV8729

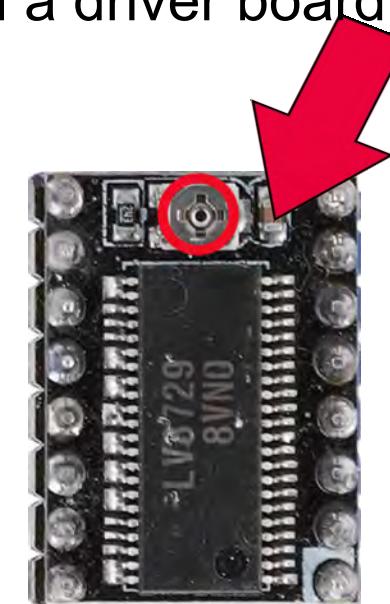
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>



$R_s = R050$ is 0.05 Ohms

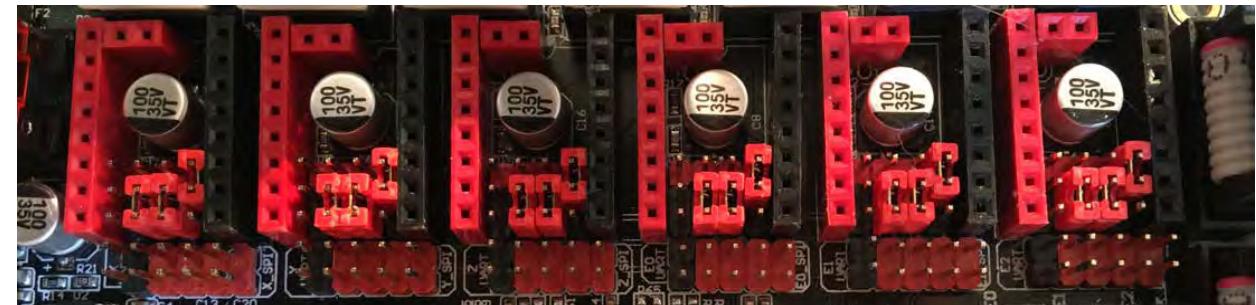
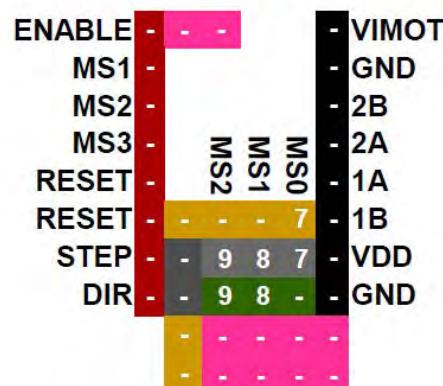
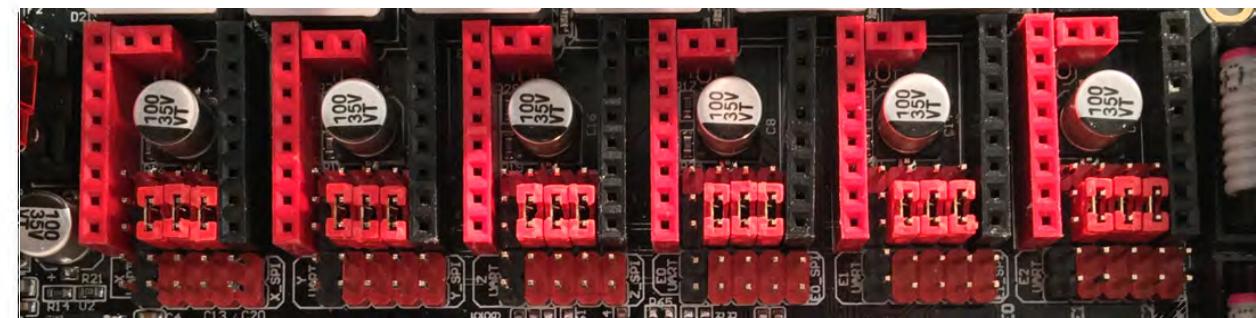
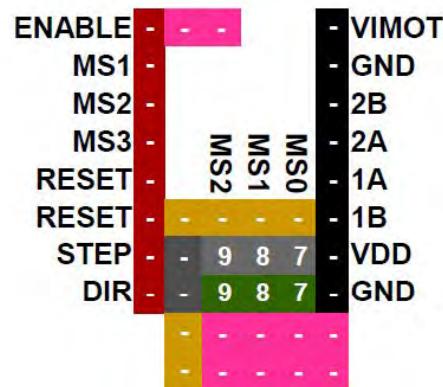
$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

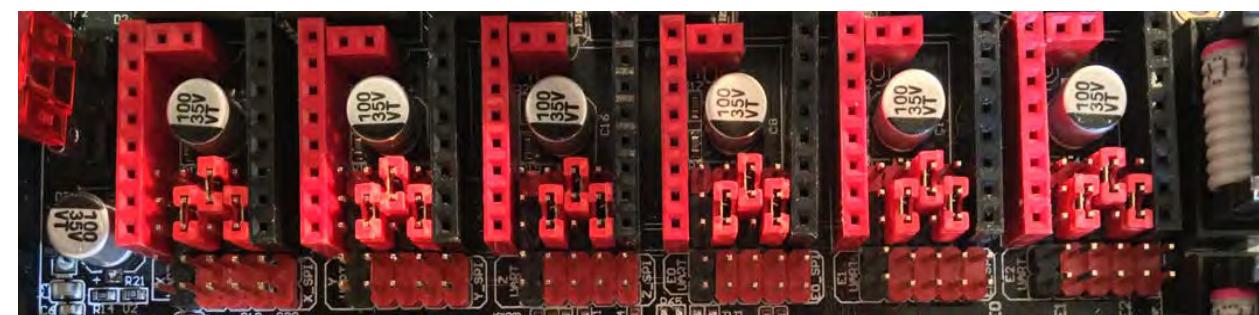
$R_s = R220$ is 0.22 Ohms

LERDGE LV8729**Stand-alone Mode**

LERDGE LV8729**Stand-alone Mode**

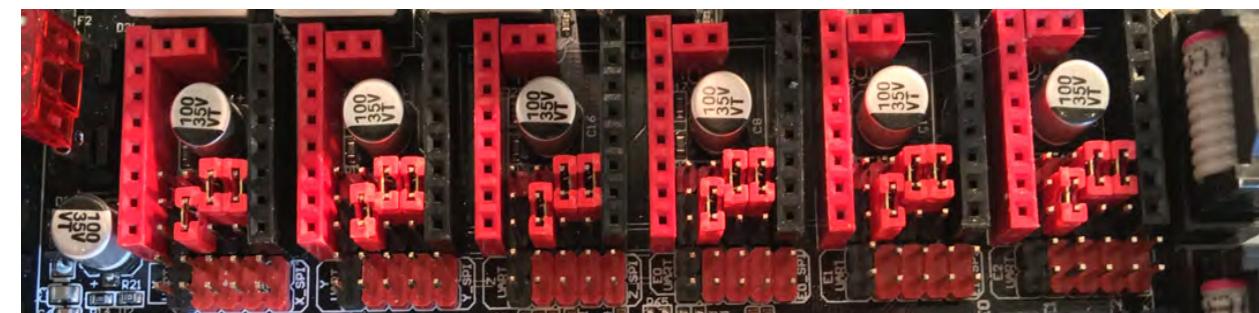
1 / 4

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
RESET	8	1B
STEP	9 8 7	VDD
DIR	9 - 7	GND



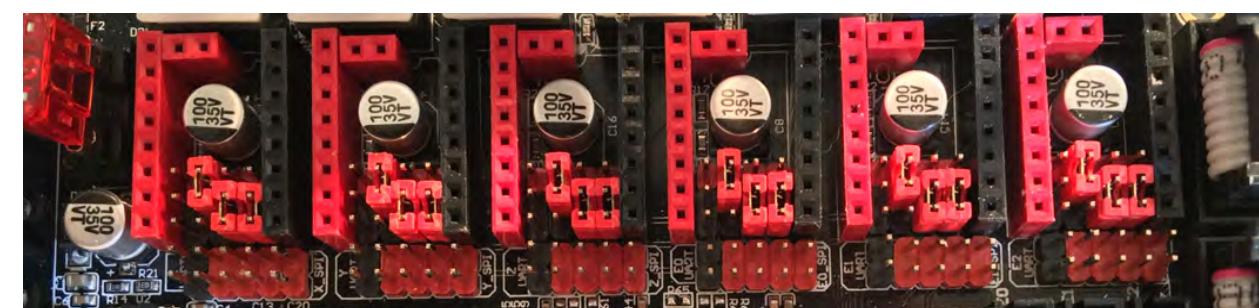
1 / 8

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
RESET	8 7	1B
STEP	9 8 7	VDD
DIR	9 - -	GND



1 / 16

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
RESET	9 - -	1B
STEP	9 8 7	VDD
DIR	8 7	GND



LERDGE LV8729**Stand-alone Mode**

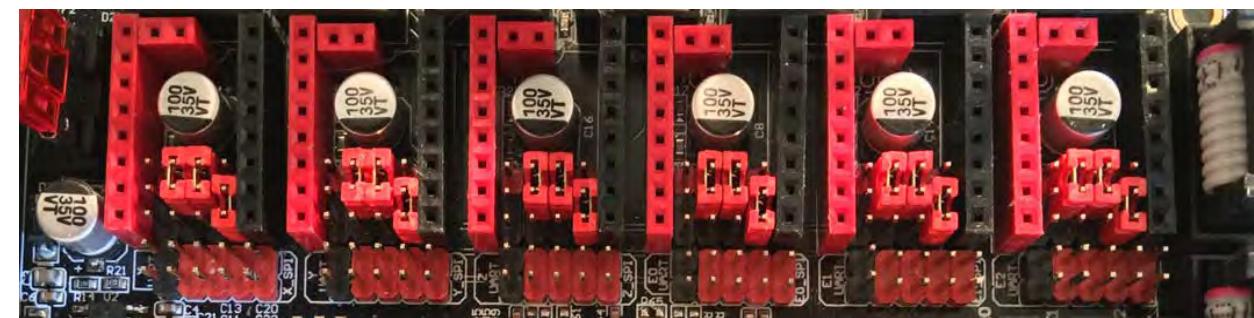
1 / 32

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	MS1
RESET	9	7
STEP	9	8
DIR	8	-
	-	-



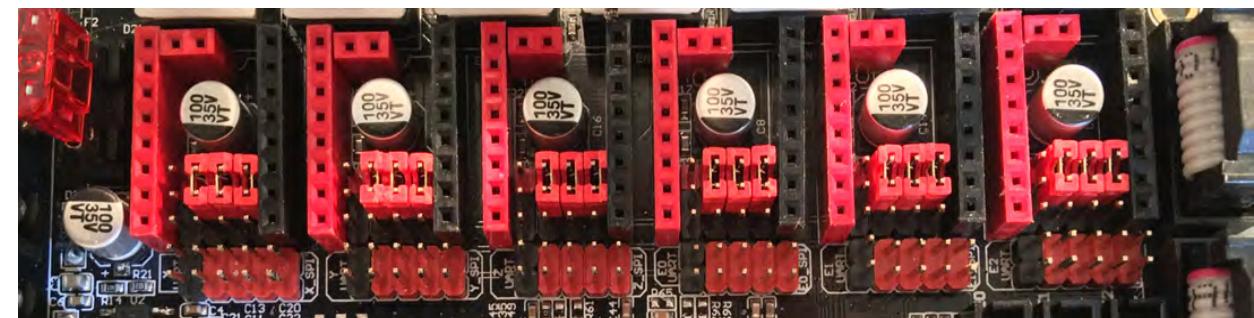
1 / 64

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	MS1
RESET	9	8
STEP	9	8
DIR	7	GND
	-	-



1 / 128

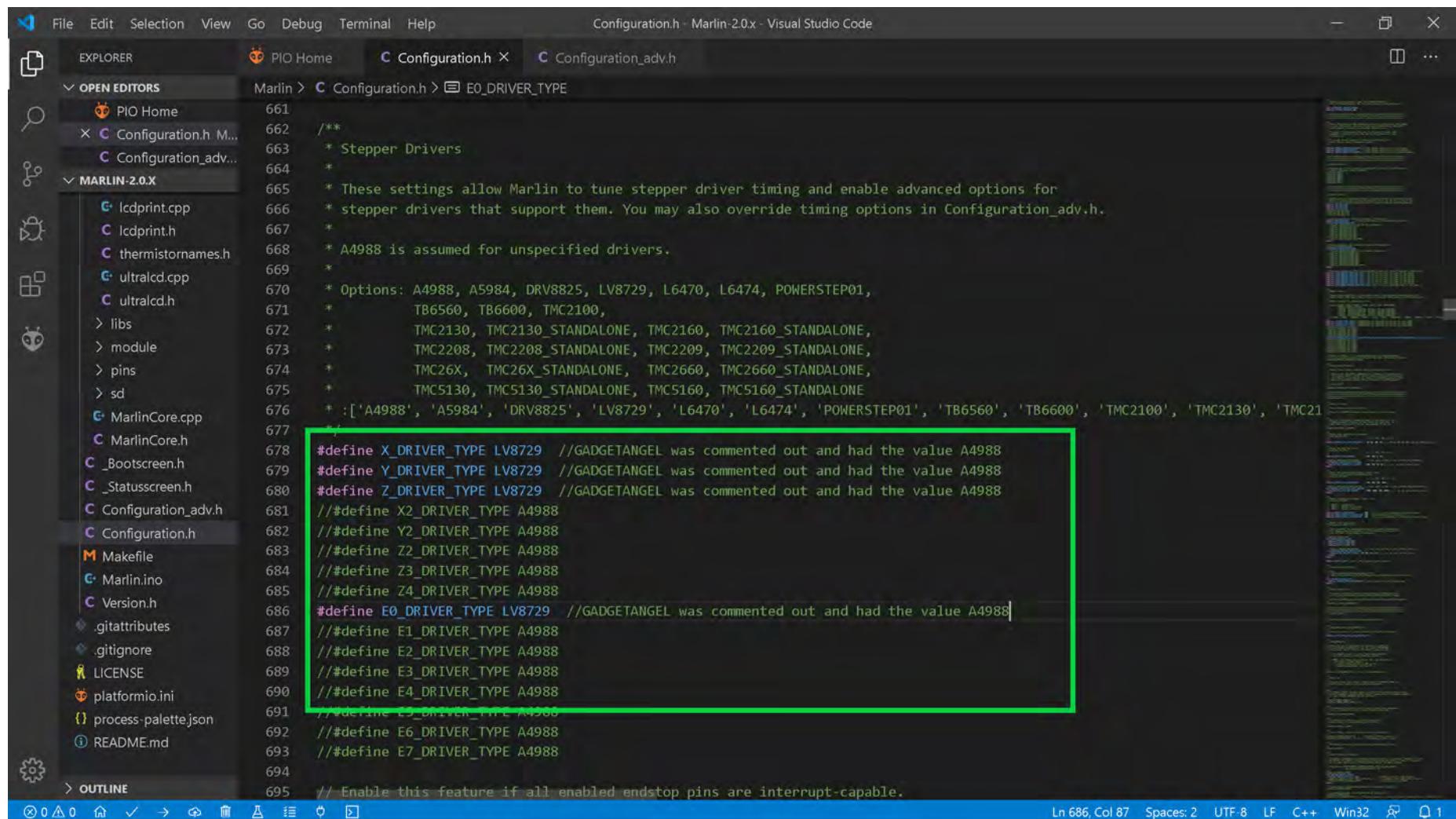
ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	MS1
RESET	9	8
STEP	9	8
DIR	7	GND
	-	-



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

NOTE: Go to Appendix C, and 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 ("//").



```

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
Marlin > Configuration.h > EO_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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677
#define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out and had the value 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
#define EO_DRIVER_TYPE A4988 //GADGETANGEL was commented out and had the value 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.

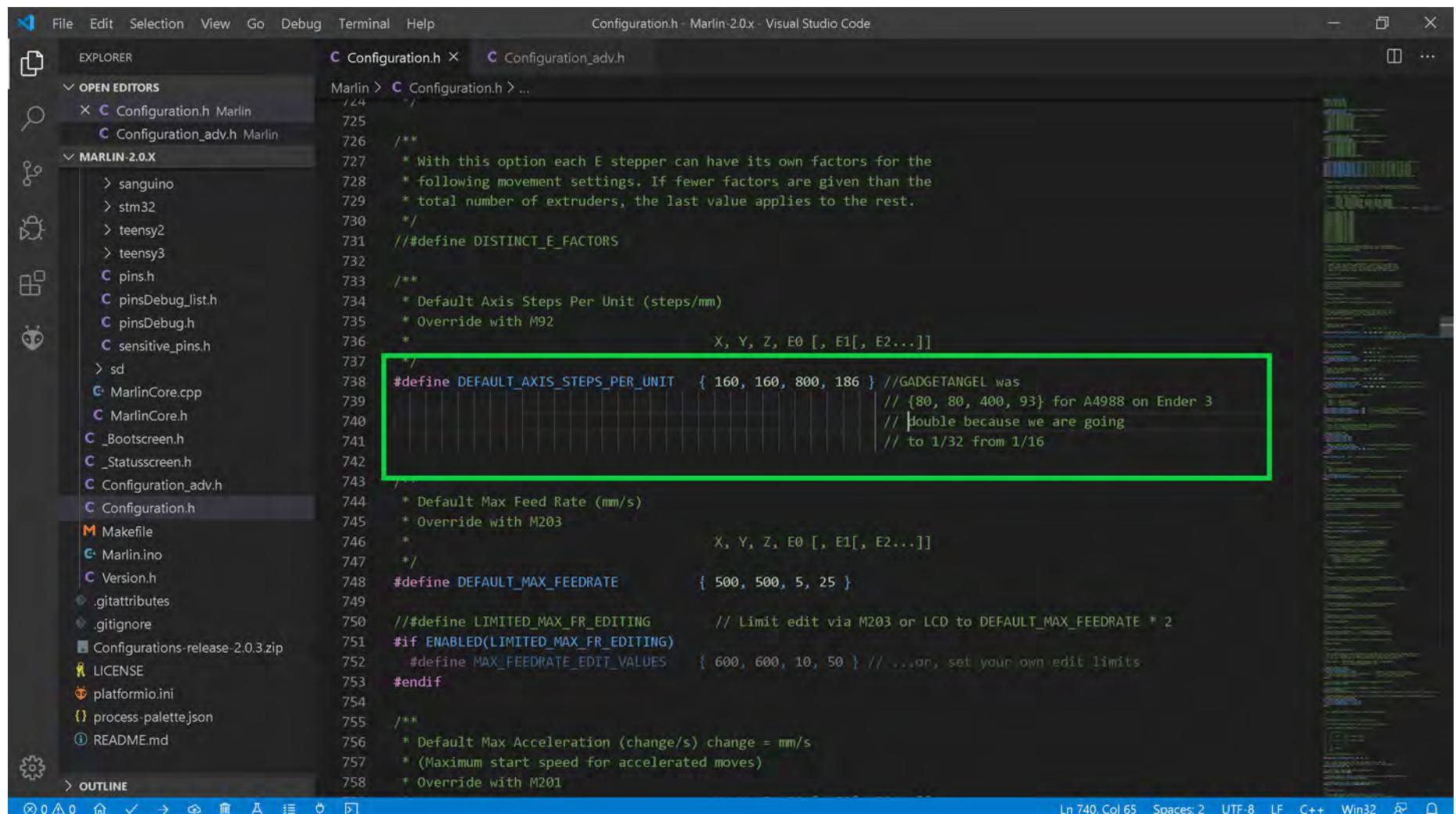
```

Ln 686, Col 87 Spaces: 2 UTF 8 LF C++ Win32 1

- Go to the next page.

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

- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 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.



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

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current line (Ln 740), column (Col 65), and other settings like spaces (Spaces: 2), encoding (UTF-8), line endings (LF), and file type (C++). The left sidebar shows the project structure with various Marlin source files and configuration files.

- 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

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):** Shows the project structure under MARLIN-2.0.X, including Configuration.h, Configuration_adv.h, and various pins and core files.
- Right Sidebar:** Shows a vertical stack of code snippets or preview windows.
- Code Editor:** The main area displays the Configuration.h file with the following content:

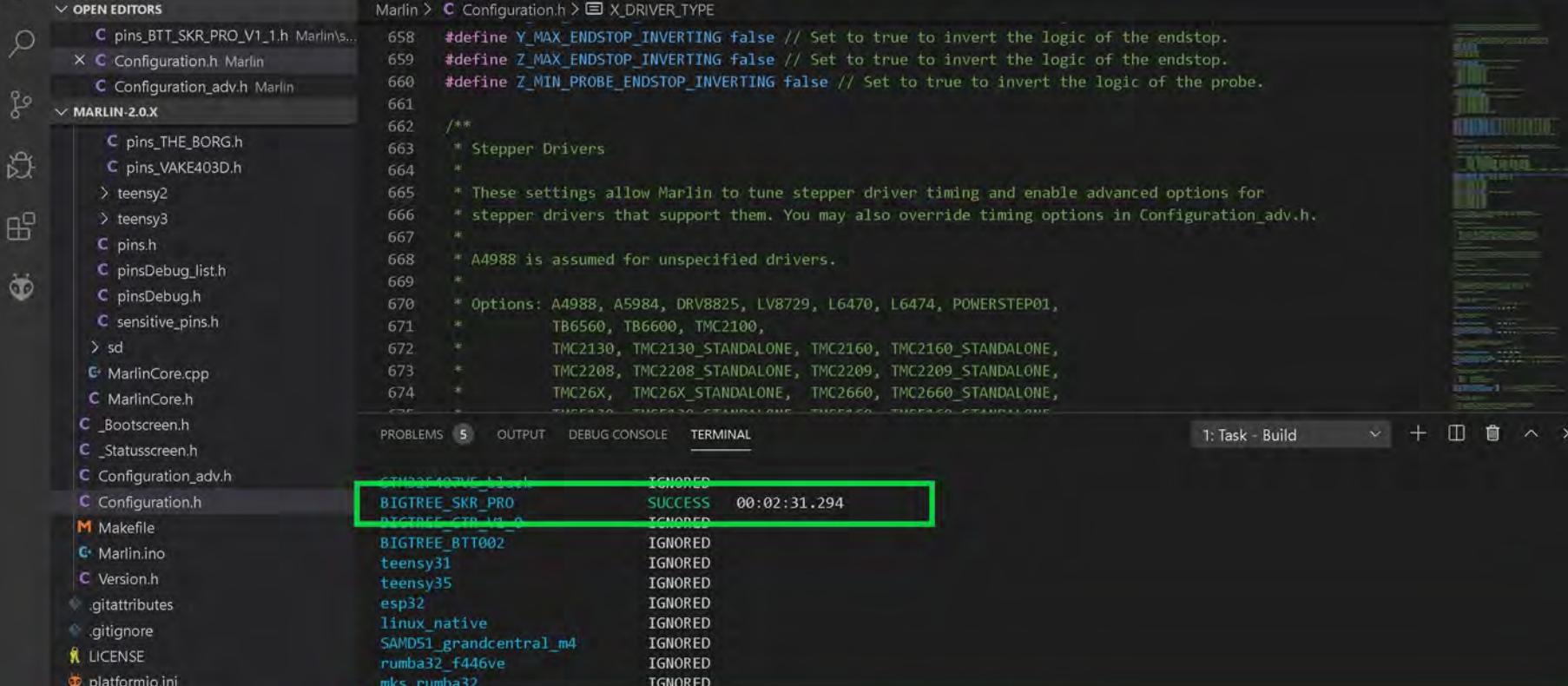
```
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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    // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```

A green rectangular box highlights the section of code from line 1049 to line 1069, which defines the INVERT_Z_DIR macro and other extruder-related macros.

- 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 a Visual Studio Code interface with the following details:

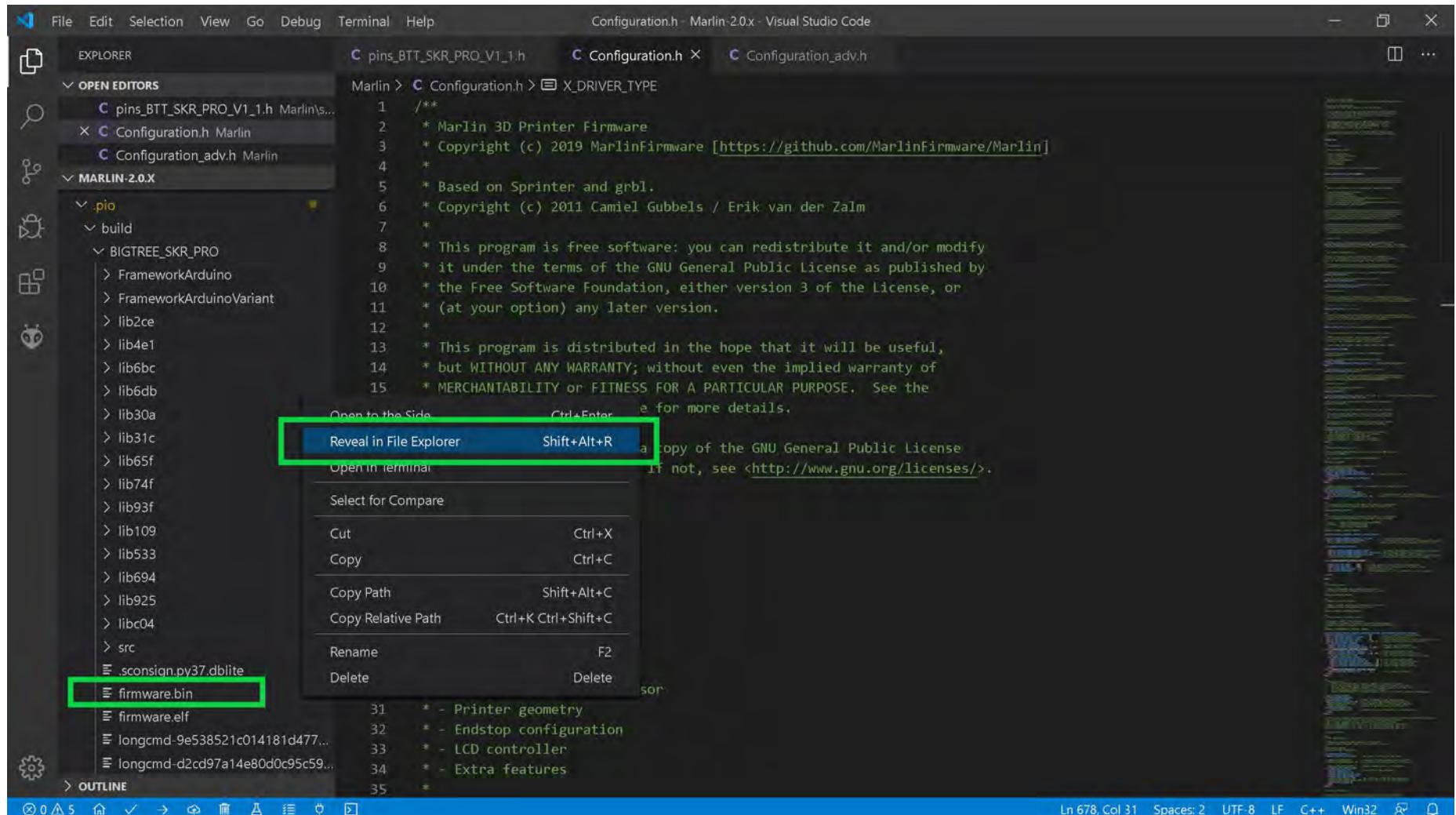
- File Explorer:** On the left, it lists files and folders under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editor:** The main editor area displays code from "pins_BTT_SKR_PRO_V1_1.h" and "Configuration.h".
- Terminal:** A terminal window at the bottom shows the output of a build command. The output is:

```
=====  
include_tree IGNORED  
===== 1 succeeded in 00:02:31.294 =====
```
- Bottom Status Bar:** Shows file statistics: "In 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 LERDGE LV8729 Drivers

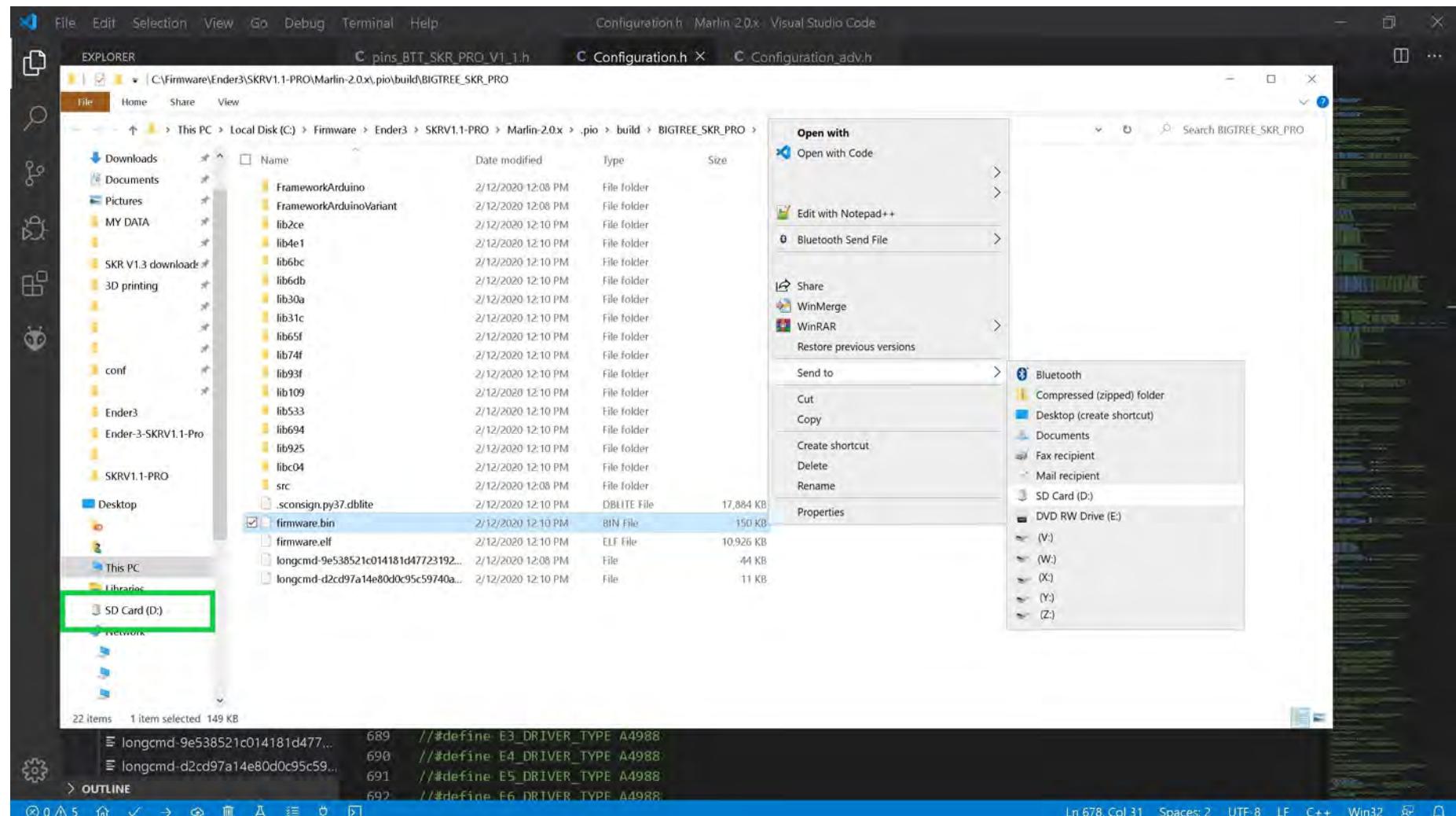
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



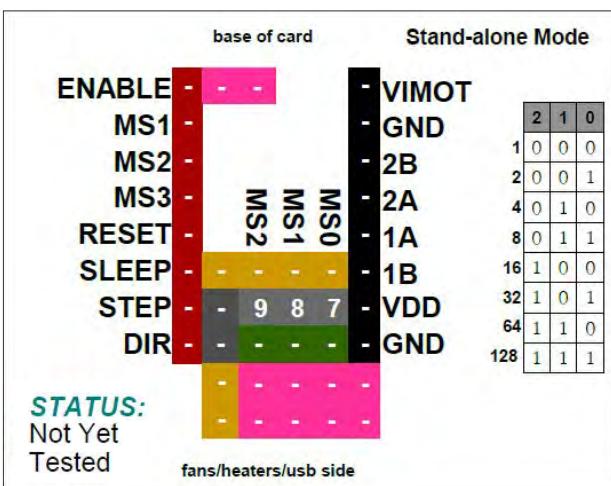
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

MKS LV8729

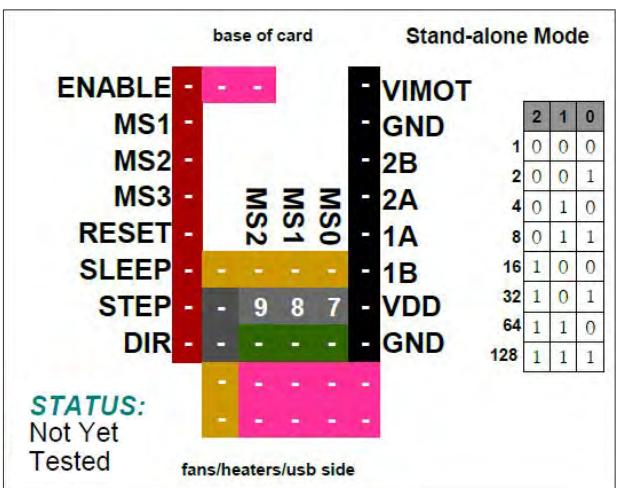
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
Makerbase LV8729 Maximum 128 Subdivision 36V DC 1.5A (peak)	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	High	High	Low	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula
 R_S (Typical Sense Resistor) = 0.22Ω

$$I_{MAX} = \frac{V_{ref}}{5 * R_S}$$

$$V_{ref} = 5 * I_{MAX} * R_S$$



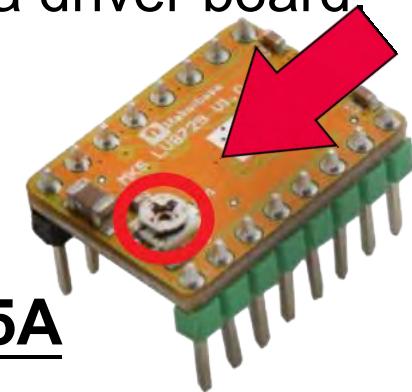
Note: Check your current sense resistor (R_s) value on the driver board, as shown in GREEN

MKS LV8729

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [**Appendix A**](#) for instructions on how to set the V_{ref} on a driver board.

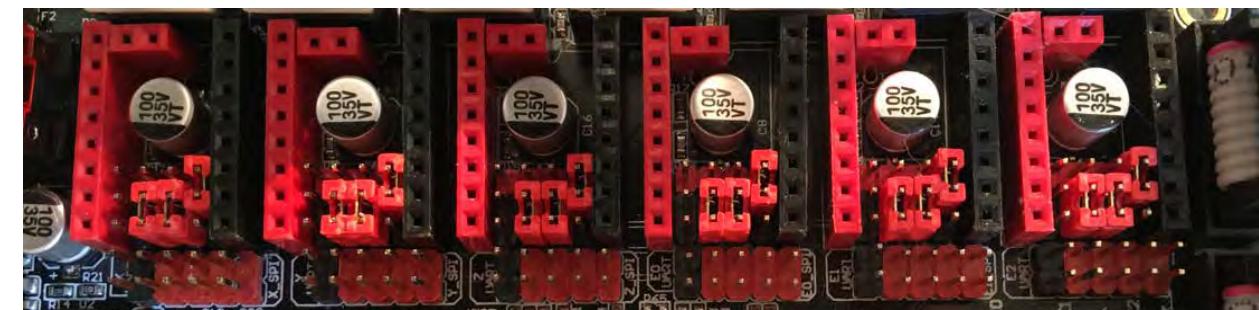
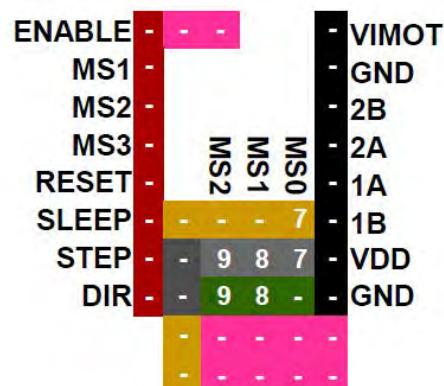
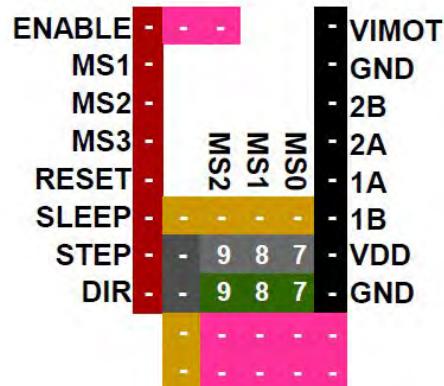
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>



- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

MKS LV8729

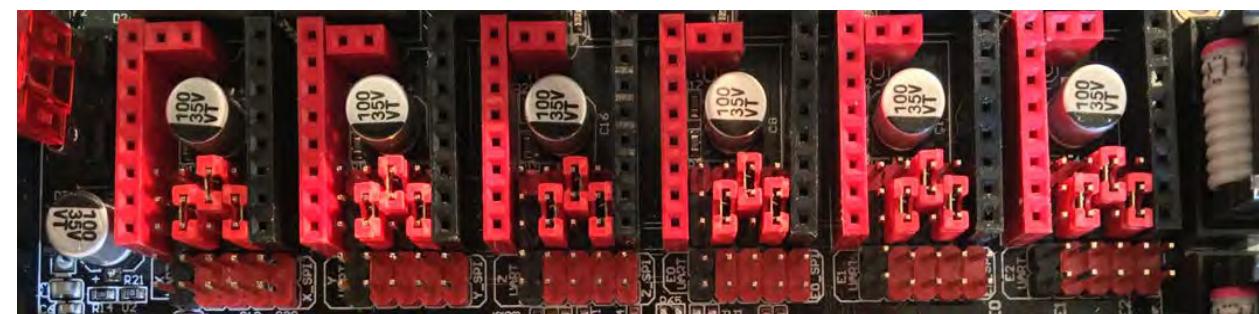
Stand-alone Mode



MKS LV8729**Stand-alone Mode**

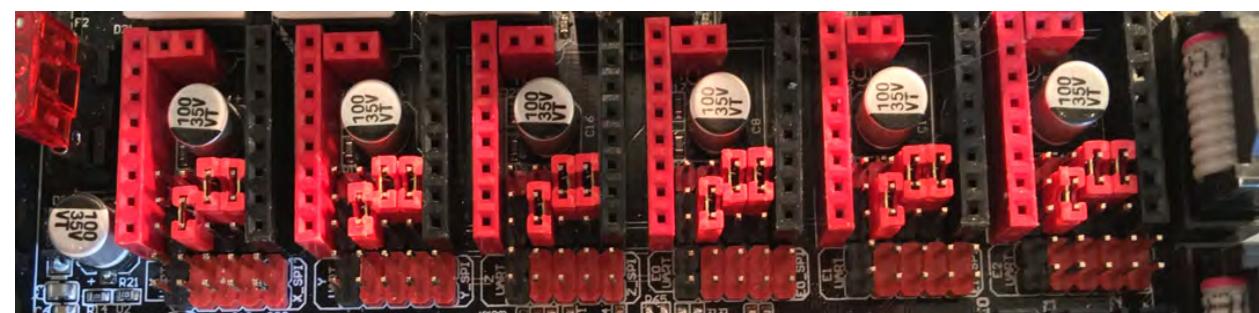
1 / 4

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	8	1B
STEP	9 8 7	VDD
DIR	9 - 7	GND



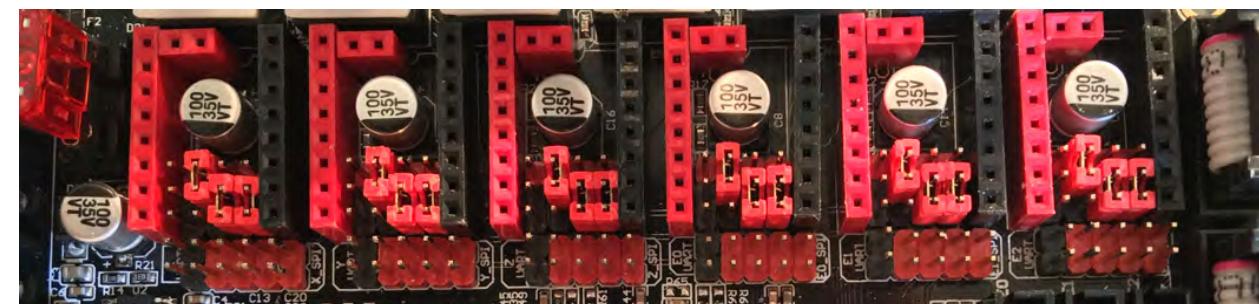
1 / 8

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	8 7	1B
STEP	9 8 7	VDD
DIR	9 - -	GND



1 / 16

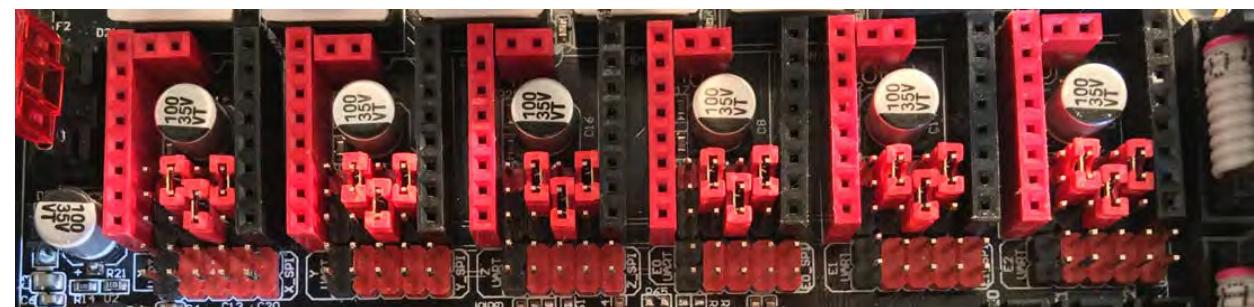
ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	MS2	2A
RESET	MS1	1A
SLEEP	9 - -	1B
STEP	9 8 7	VDD
DIR	8 7	GND



MKS LV8729**Stand-alone Mode**

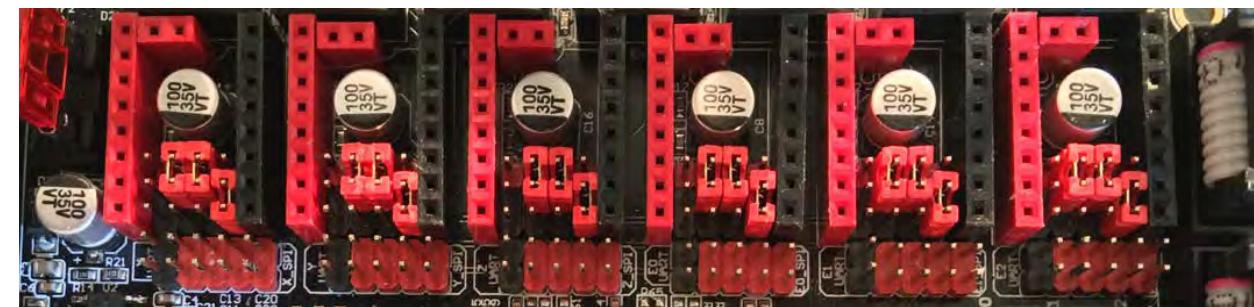
1 / 32

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 7	1B
STEP	9 8 7	VDD
DIR	8	GND



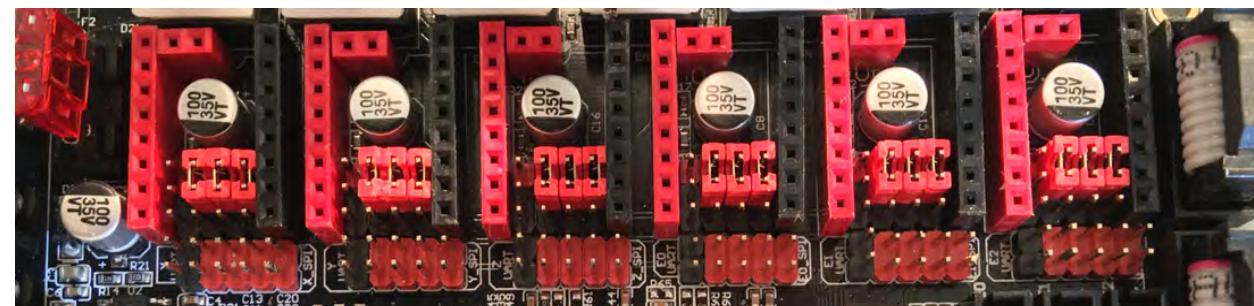
1 / 64

ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 8	1B
STEP	9 8 7	VDD
DIR	7	GND



1 / 128

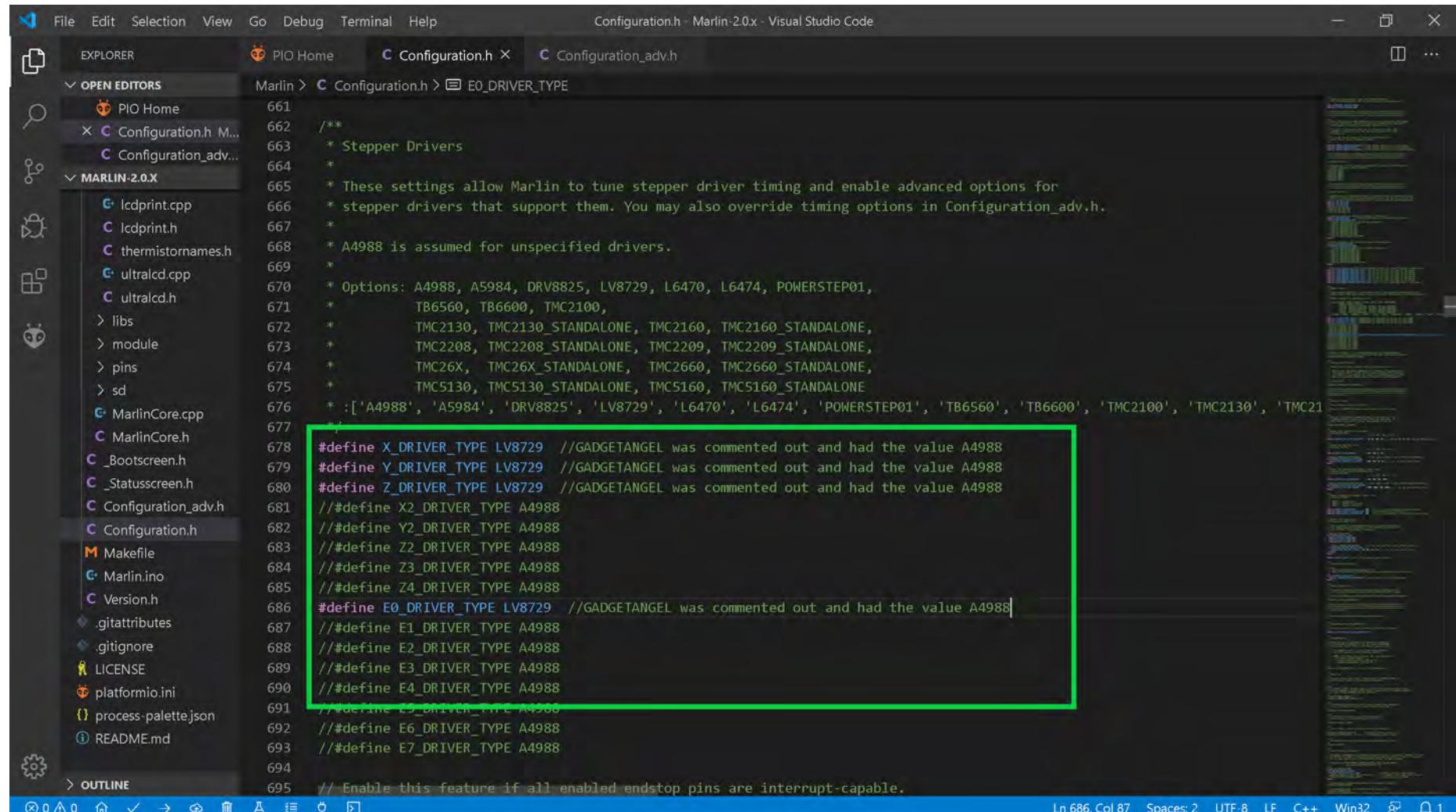
ENABLE	- -	VIMOT
MS1	-	GND
MS2	-	2B
MS3	-	2A
RESET	MS2	1A
SLEEP	9 8 7	1B
STEP	9 8 7	VDD
DIR	- - -	GND



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

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

- Change the stepper motor drivers so that Marlin knows you are using MKS 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 MKS 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 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  *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160',
676  *           'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
677  */
678 #define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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.

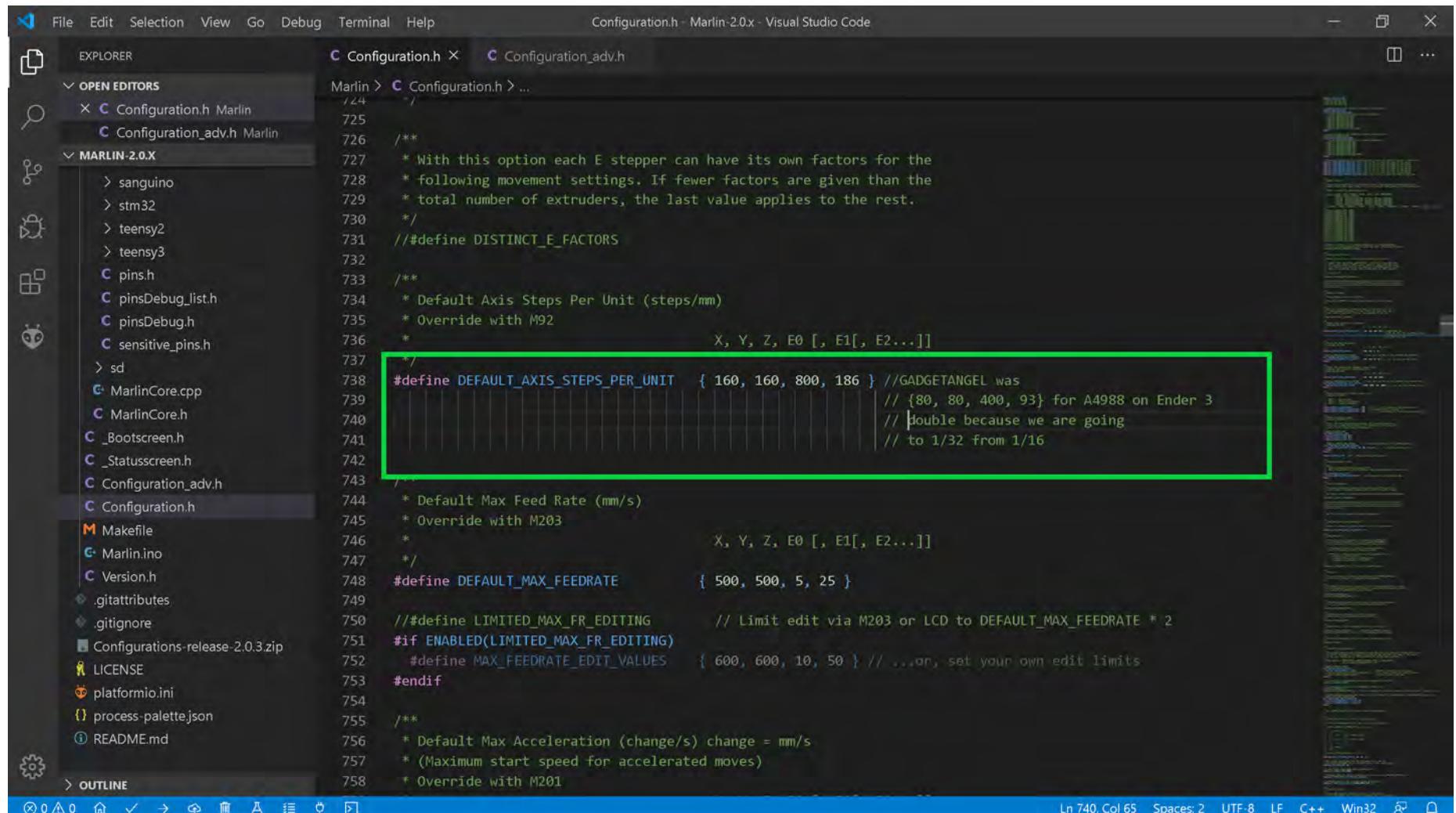
```

A green rectangular box highlights the driver type definitions for axes X, Y, Z, and E0, 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 MKS LV8729 Drivers

- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 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.



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

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current line (Ln 740), column (Col 65), and other settings like spaces (Spaces: 2), encoding (UTF-8), line endings (LF), and file type (C++). The left sidebar shows the project structure with various Marlin source files and configuration files like `Configuration.h` and `Configuration_adv.h`.

- Go to the next page.

The (latest release of) Marlin Setup for MKS 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-2.0x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h Configuration_adv.h

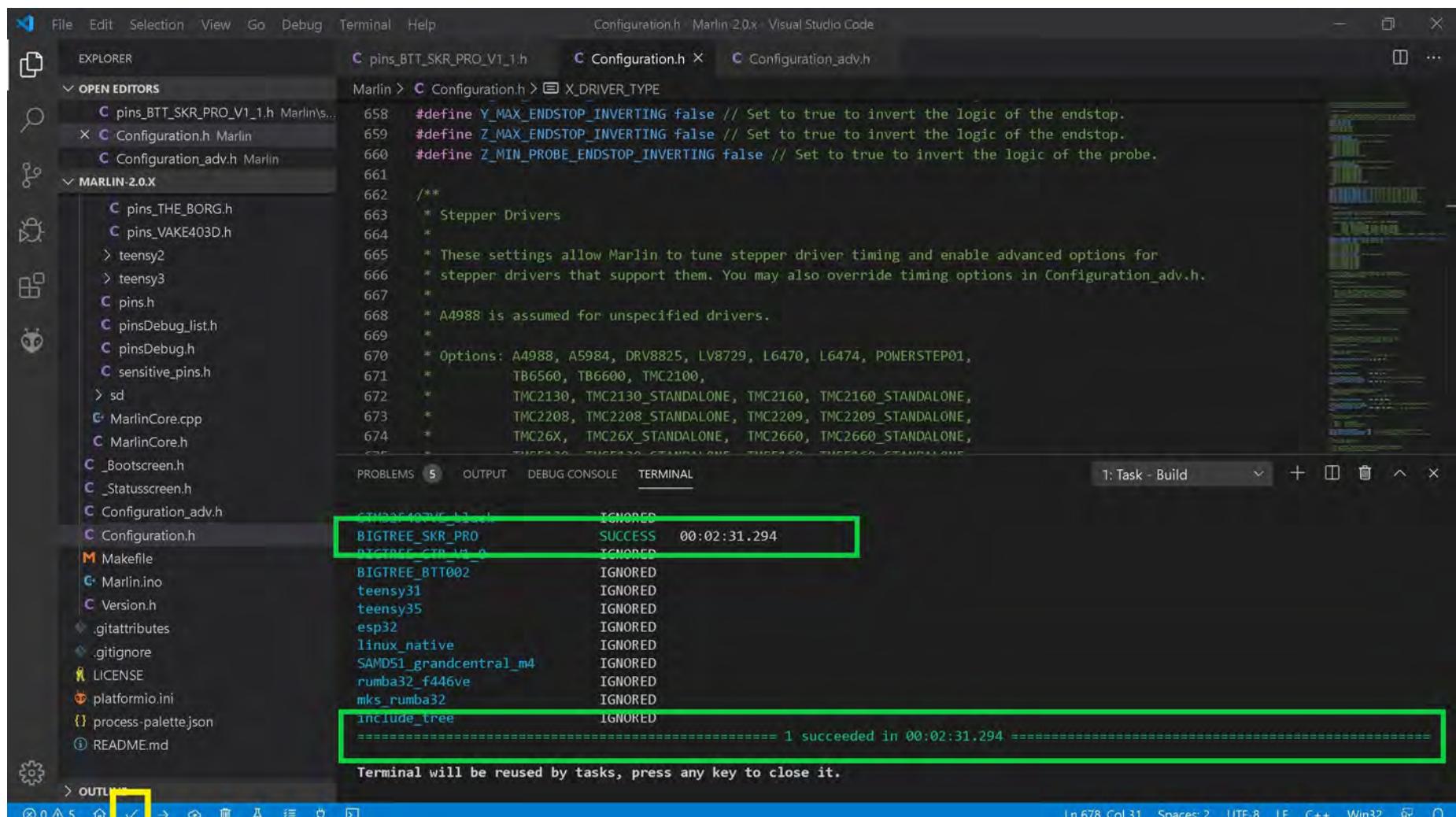
Marlin > Configuration.h > INVERT_Z_DIR

```
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```

- Go to the next page.

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

- The end of Marlin setup for MKS 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.



```

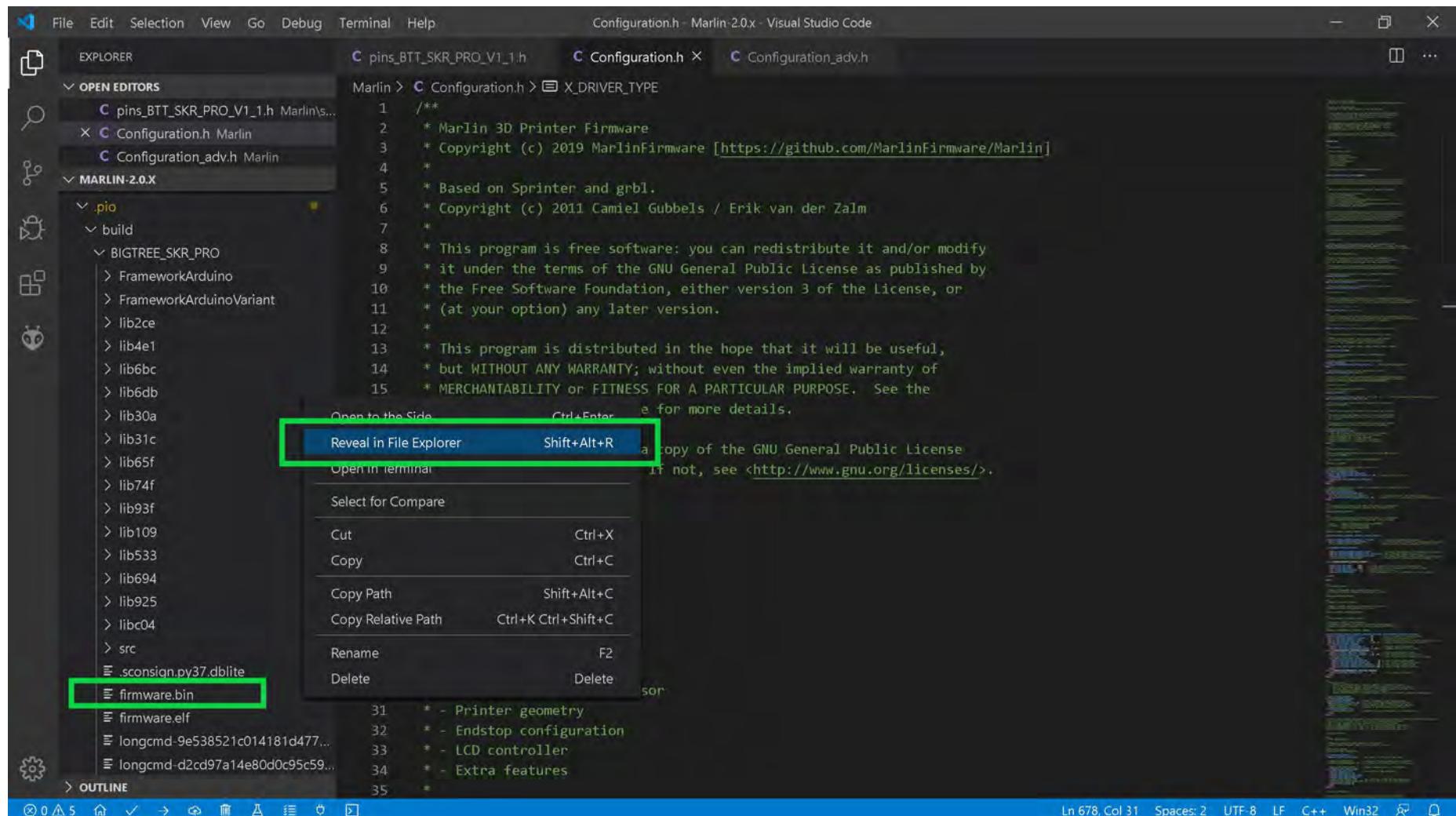
Configuration.h Marlin 2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h Marlin 2.0.x - Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\...
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 + - x
CTM2208_TMC_12_v1 IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_CTP_12_v0 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 MKS LV8729 Drivers

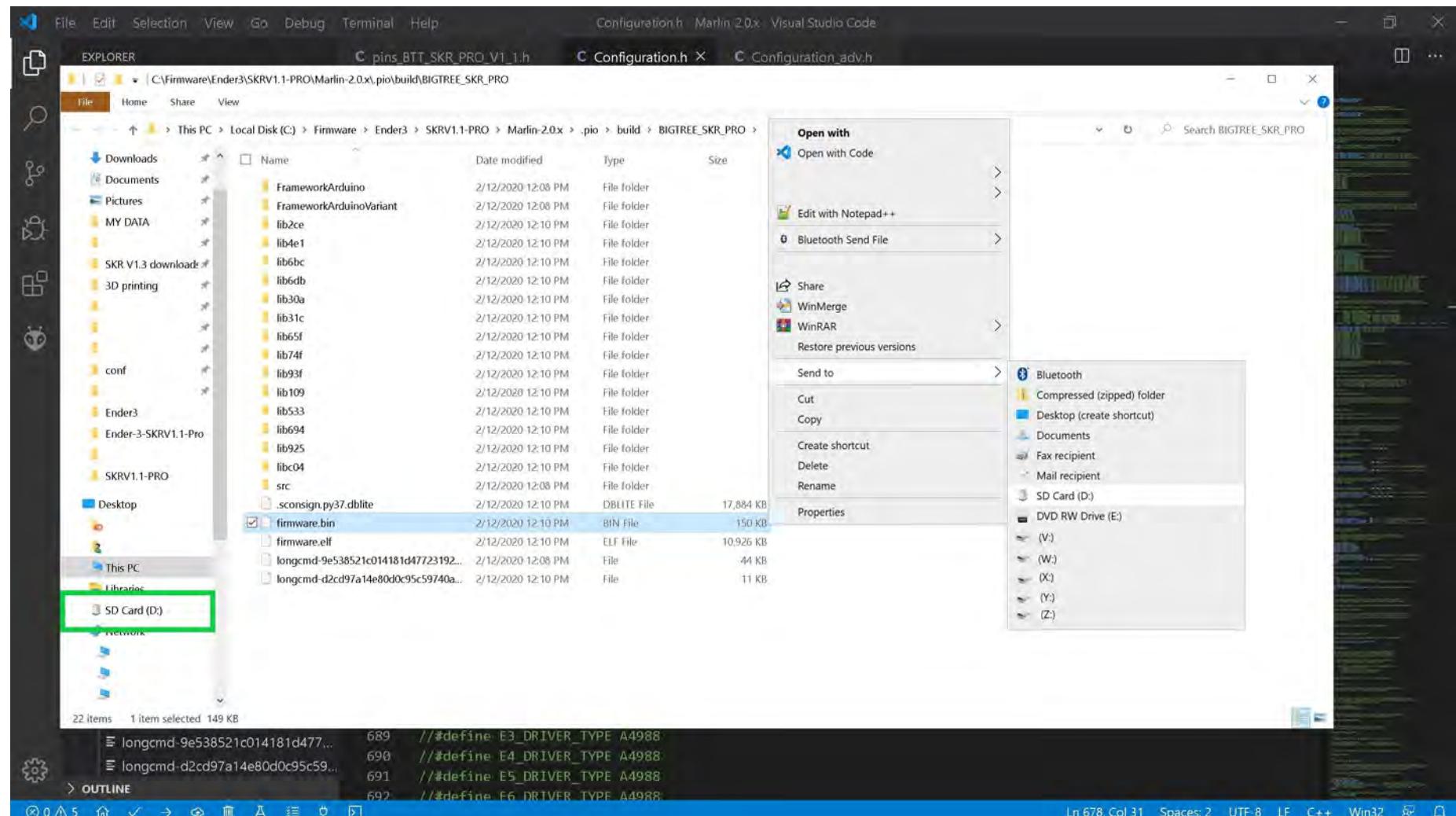
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



- Go to the next page.

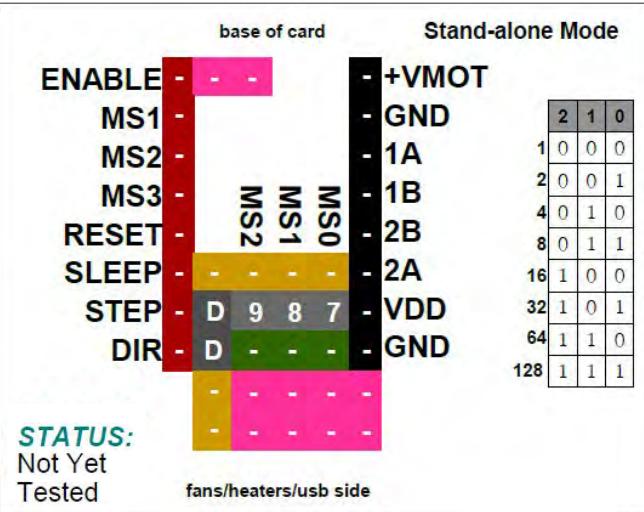
The (latest release of) Marlin Setup for MKS LV8729 Drivers

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

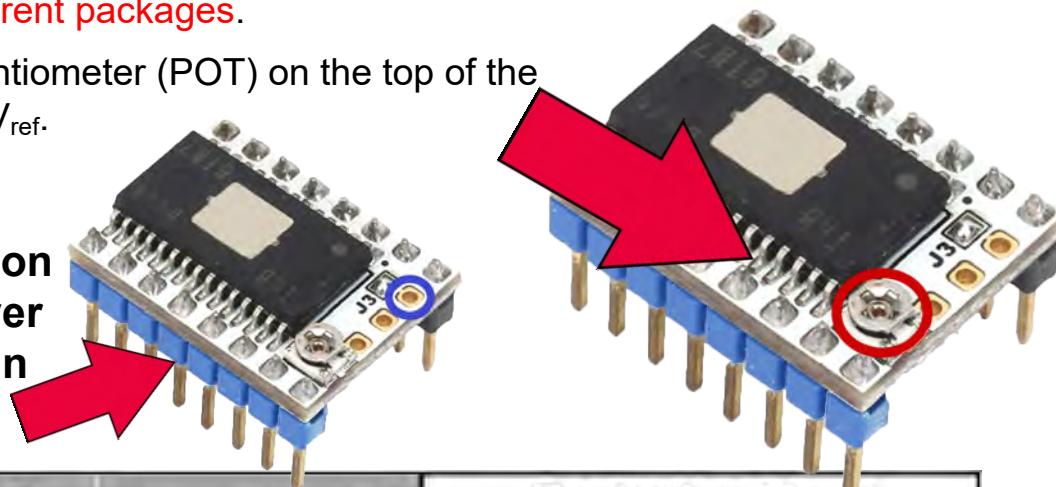
FYSETC S6128 V1.1



FYSETC S6128 V1.1 states: "SureStepr SD6128 is a stepper driver board based on the THB6128 chip"; my understanding is that the **THB6128** and the **LV8729** are **the same chip in different packages**.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

Note: " V_{ref} Test point" location is on the top of the driver board, as shown in **BLUE**

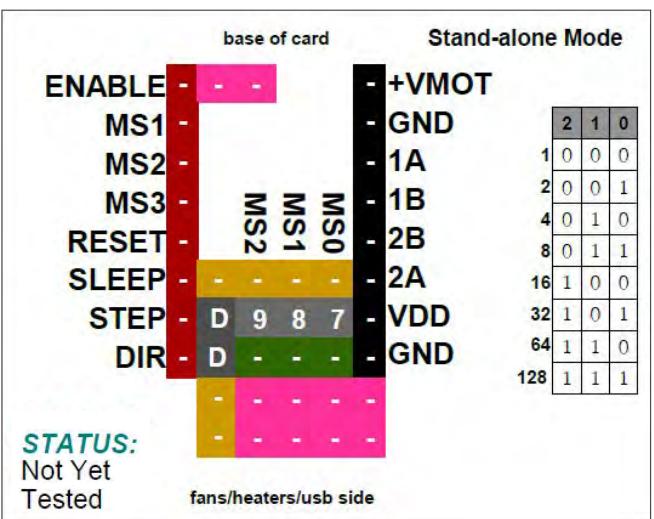


Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
FYSETC SureStepr SD6128	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	High	High	Low	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

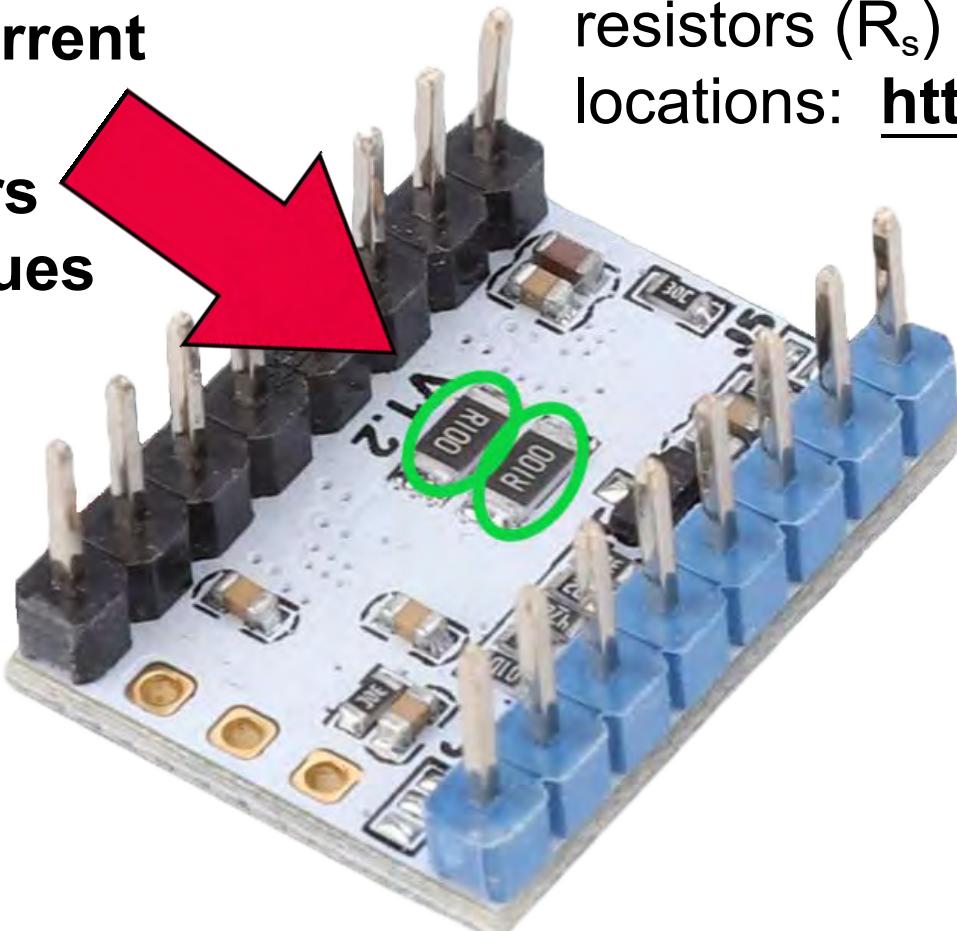
Driving Current Calculation Formula
R_S (Typical Sense Resistor) = 0.1Ω

$$I_{MAX} = \frac{V_{ref}}{5 * R_S}$$

$$V_{ref} = 5 * I_{MAX} * R_S$$

FYSETC S6128 V1.1

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>

$R_s = R050$ is 0.05 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

FYSETC S6128 V1.1

Stand-alone Mode

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.

ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	MS2	MS1	MS0
SLEEP	-	-	-
STEP	D 9	8	7
DIR	D 9	8	7
	-	-	VDD
	-	-	GND

STEP

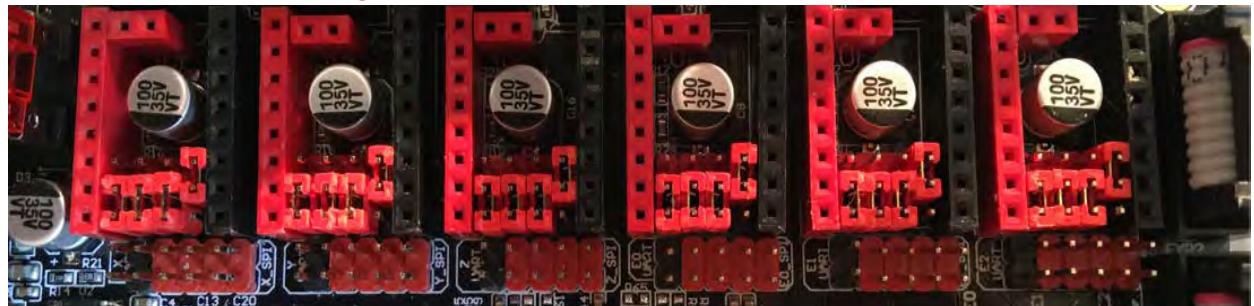
Note: The "D" jumper MUST be SET!



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	MS2	MS1	MS0
SLEEP	-	-	7
STEP	D 9	8	7
DIR	D 9	8	-
	-	-	VDD
	-	-	GND

1 / 2

Note: The "D" jumper MUST be SET!



FYSETC S6128 V1.1

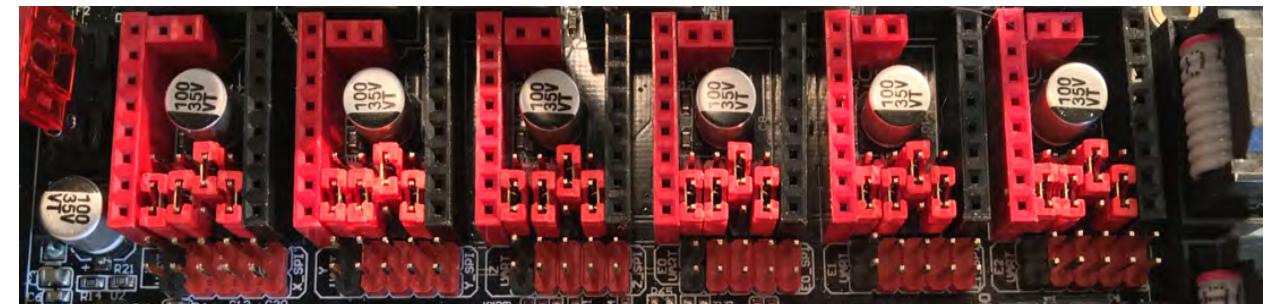
Stand-alone Mode

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.

ENABLE	- -	+VMOT
MS1	-	GND
MS2	-	1A
MS3	MS2	1B
RESET	MS1	1B
SLEEP	MS0	2B
STEP	8	2A
DIR	D 9 8 7	VDD
	D 9 - 7	GND
	- - -	-

1 / 4

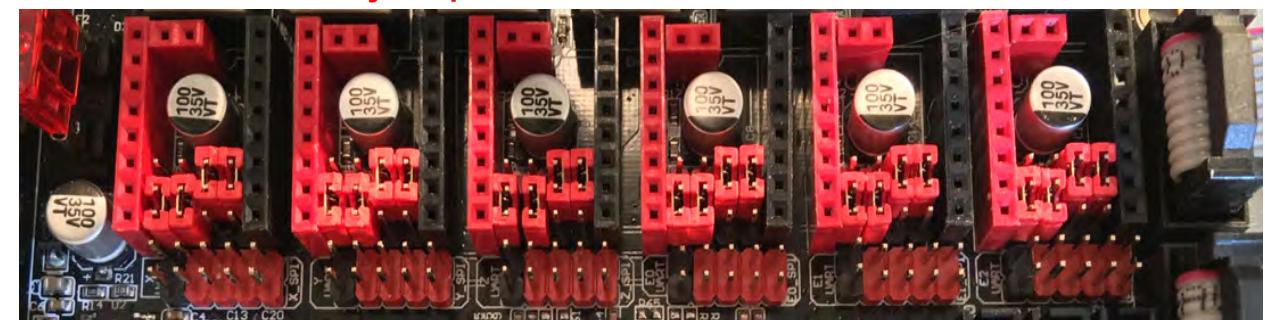
Note: The "D" jumper MUST be SET!



ENABLE	- -	+VMOT
MS1	-	GND
MS2	-	1A
MS3	MS2	1B
RESET	MS1	1B
SLEEP	MS0	2B
STEP	8	2A
DIR	D 9 8 7	VDD
	D 9 - -	GND
	- - -	-

1 / 8

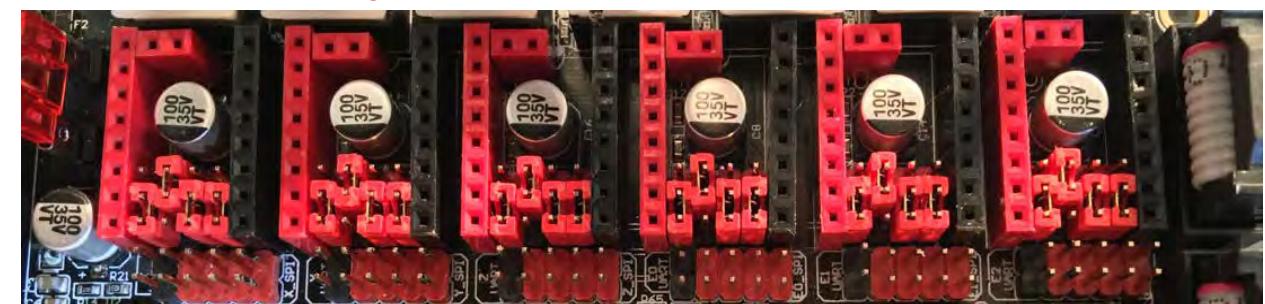
Note: The "D" jumper MUST be SET!



ENABLE	- -	+VMOT
MS1	-	GND
MS2	-	1A
MS3	MS2	1B
RESET	MS1	1B
SLEEP	MS0	2B
STEP	8	2A
DIR	D 9 8 7	VDD
	D 9 - 7	GND
	- - -	-

1 / 16

Note: The "D" jumper MUST be SET!



FYSETC S6128 V1.1

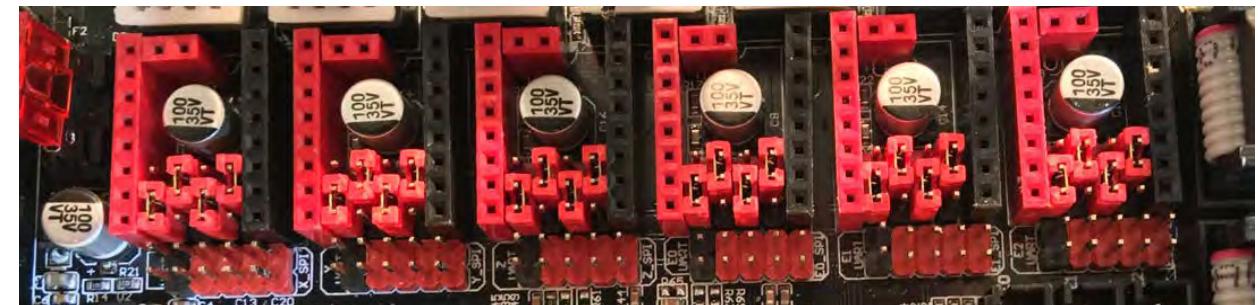
Stand-alone Mode

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.

1 / 32

ENABLE	-	-	-	+VMOT
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
SLEEP	-	9	7	2A
STEP	D	9	8	7
DIR	D	-	8	-
	-	-	-	GND
	-	-	-	-

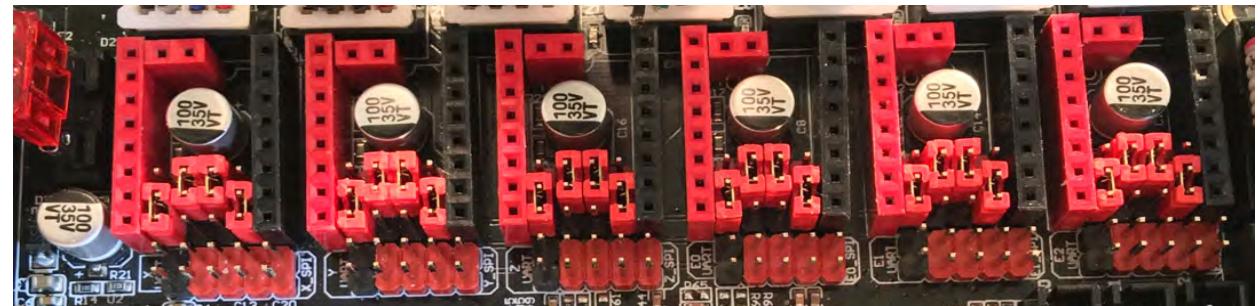
Note: The "D" jumper MUST be SET!



1 / 64

ENABLE	-	-	-	+VMOT
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
SLEEP	-	9	8	-
STEP	D	9	8	7
DIR	D	-	-	7
	-	-	-	-
	-	-	-	-

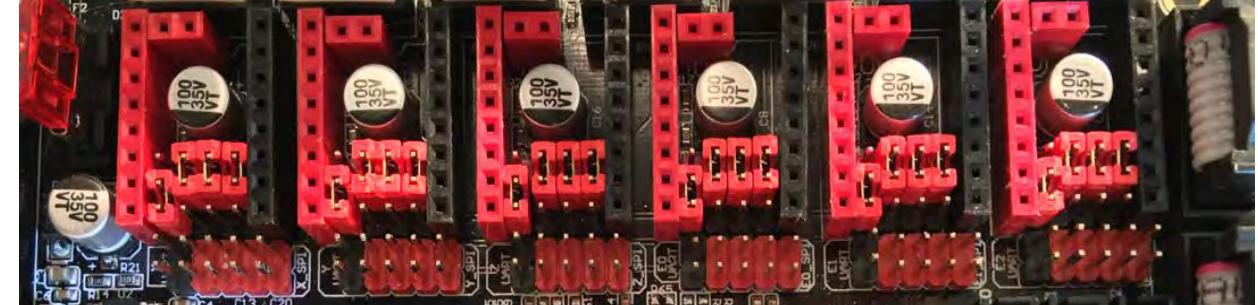
Note: The "D" jumper MUST be SET!



1 / 128

ENABLE	-	-	-	+VMOT
MS1	-	-	-	GND
MS2	-	-	-	1A
MS3	-	-	-	1B
RESET	-	MS2	MS1	MS0
SLEEP	-	9	8	7
STEP	D	9	8	7
DIR	D	-	-	-
	-	-	-	-
	-	-	-	-

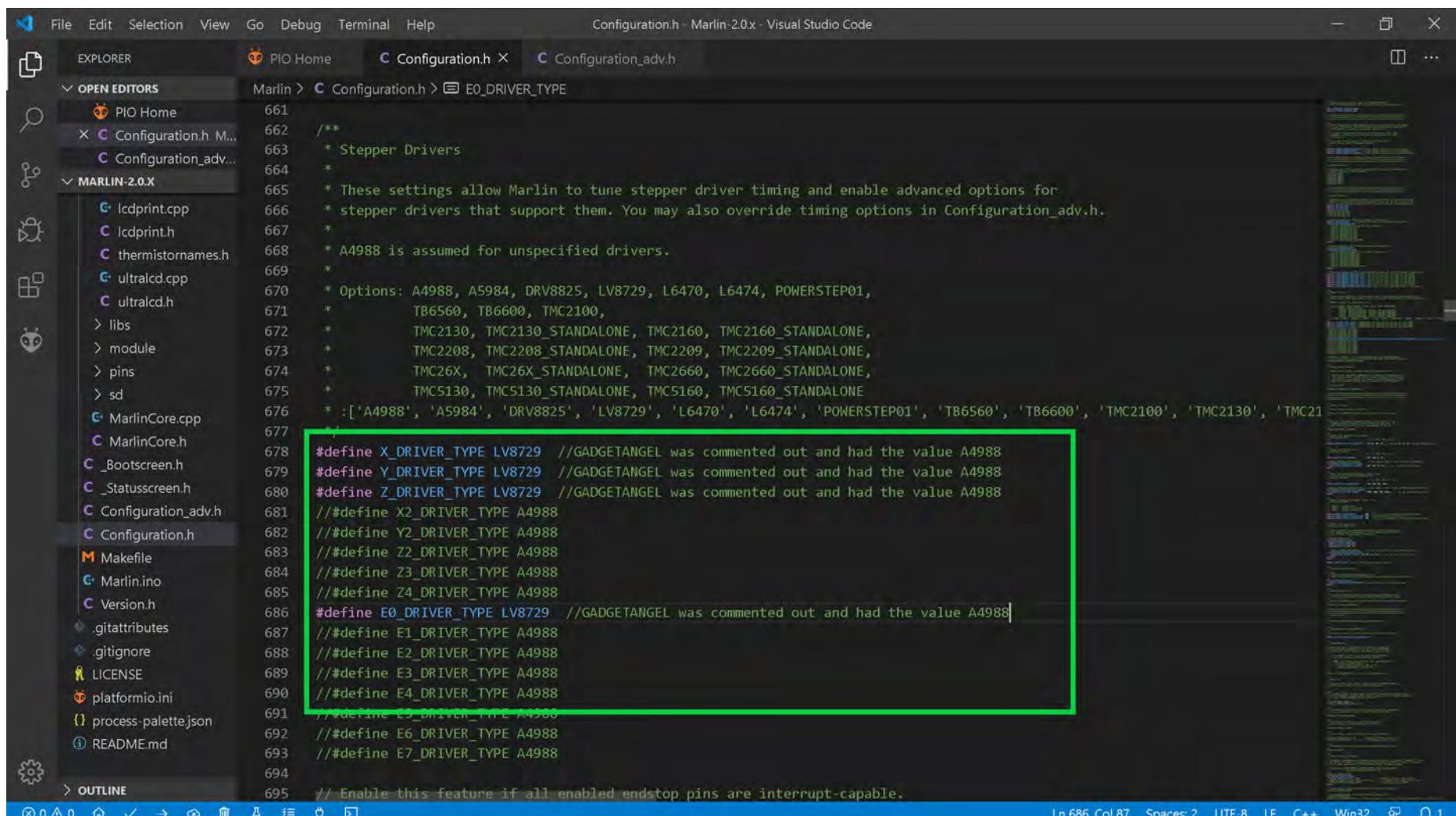
Note: The "D" jumper MUST be SET!



The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for FYSETC S6128 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using FYSETC S6128 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 S6128 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 S6128 is a drop in replacement for the LV8729. Since Marlin does not have an option for S6128 we will use the LV8729 option.



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

```

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 * T86560, 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', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
677
# define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
# define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
# define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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.

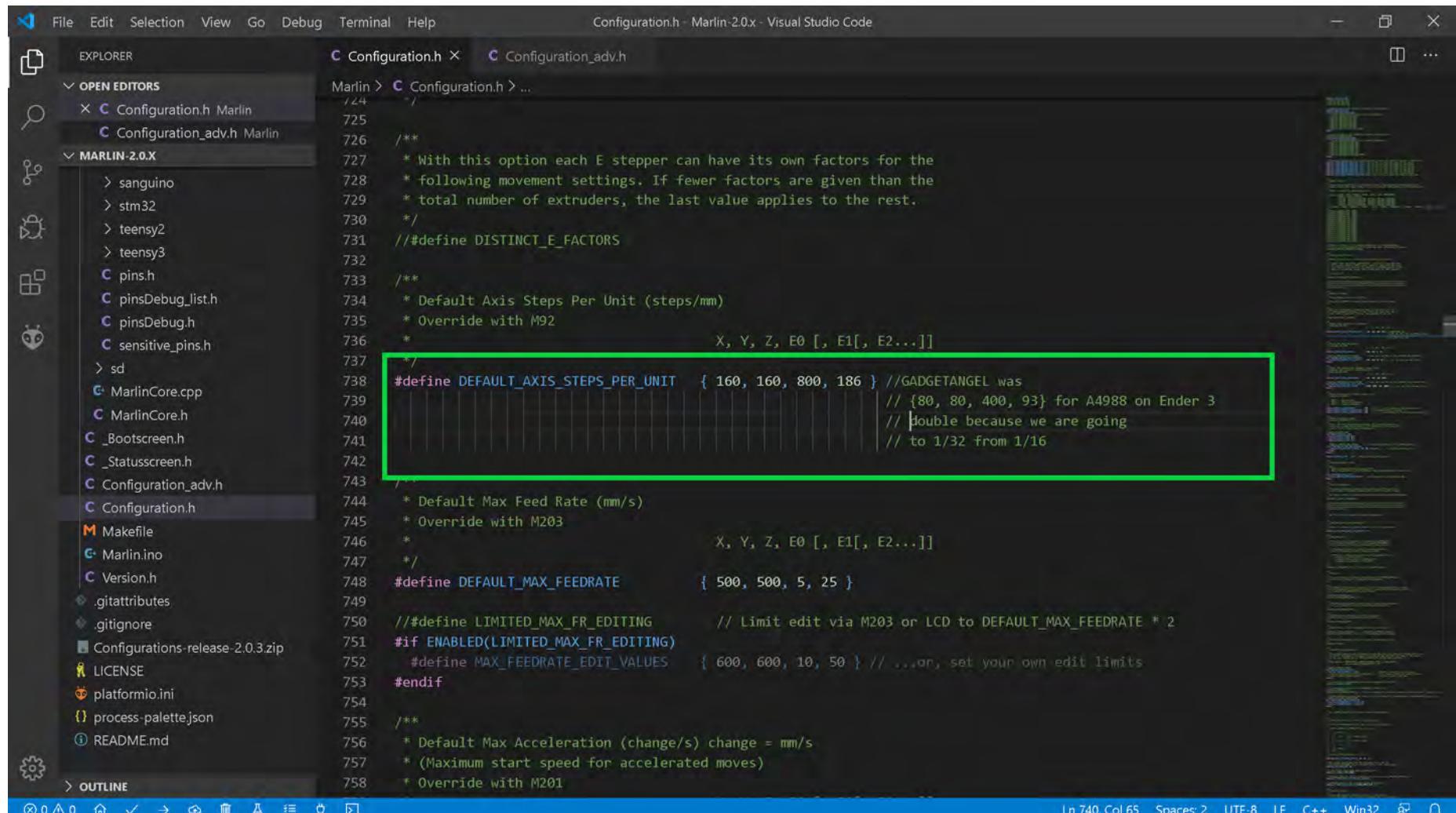
```

A green rectangular box highlights the line `# define E0_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988`. This indicates that the driver type for axis 0 has been changed from the default A4988 to the FYSETC S6128 driver type LV8729. The rest of the driver configuration lines for other axes (X, Y, Z, E1-E7) are also shown, with their comments removed.

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers

- We are changing from A4988 stepper motor drivers on the Ender 3 to S6128 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.



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

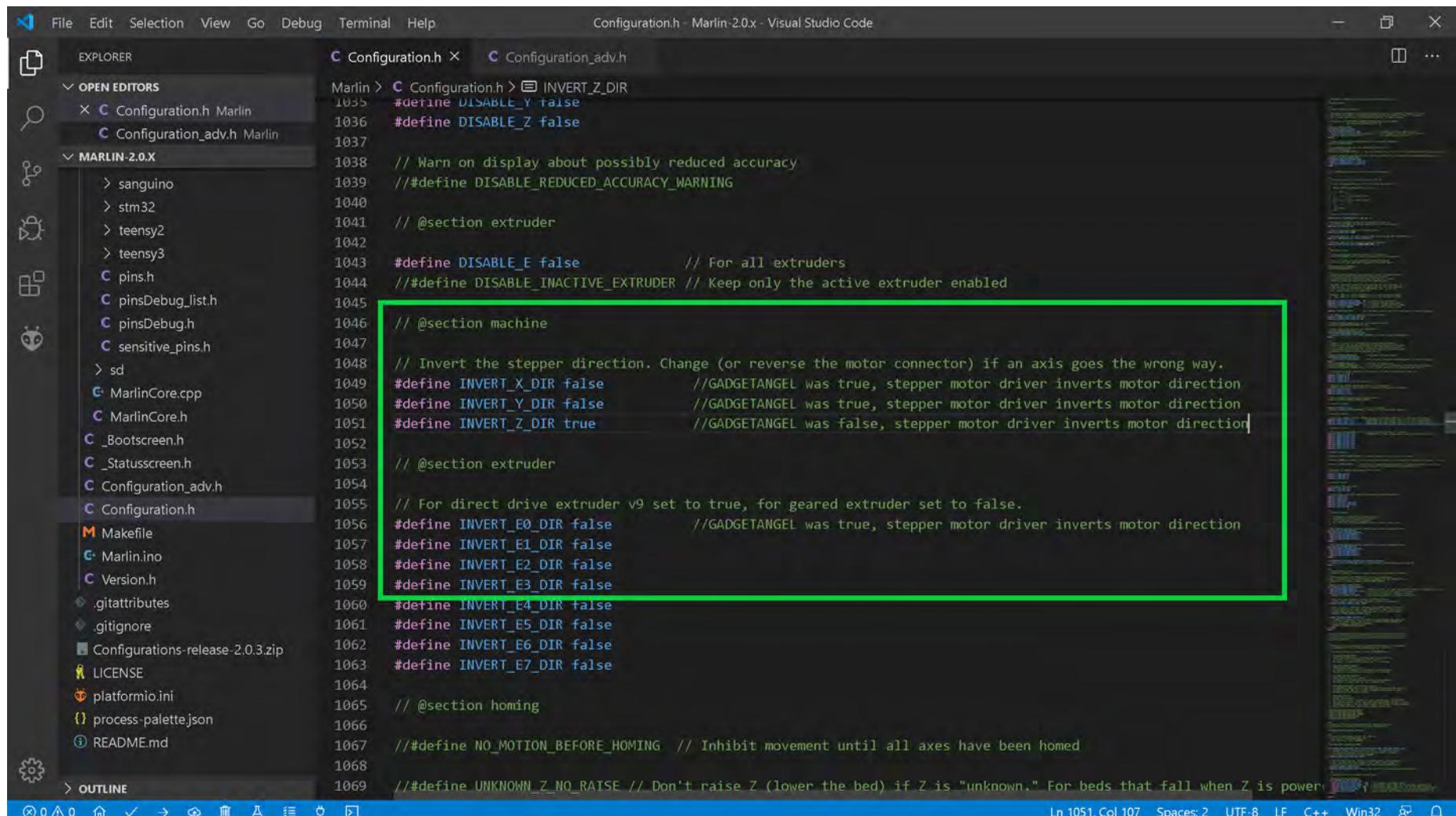
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right indicates the current file is `Configuration.h - Marlin-2.0.x - Visual Studio Code`, with line 738, column 65, spaces: 2, encoding: UTF-8, line feed: LF, character set: C++, and window size: Win32.

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers

- Since the A4988 driver is what my Ender 3 used, but, now I want to use S6128 drivers, I must invert the stepper motor direction because the S6128 or 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 S6128 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  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
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌂

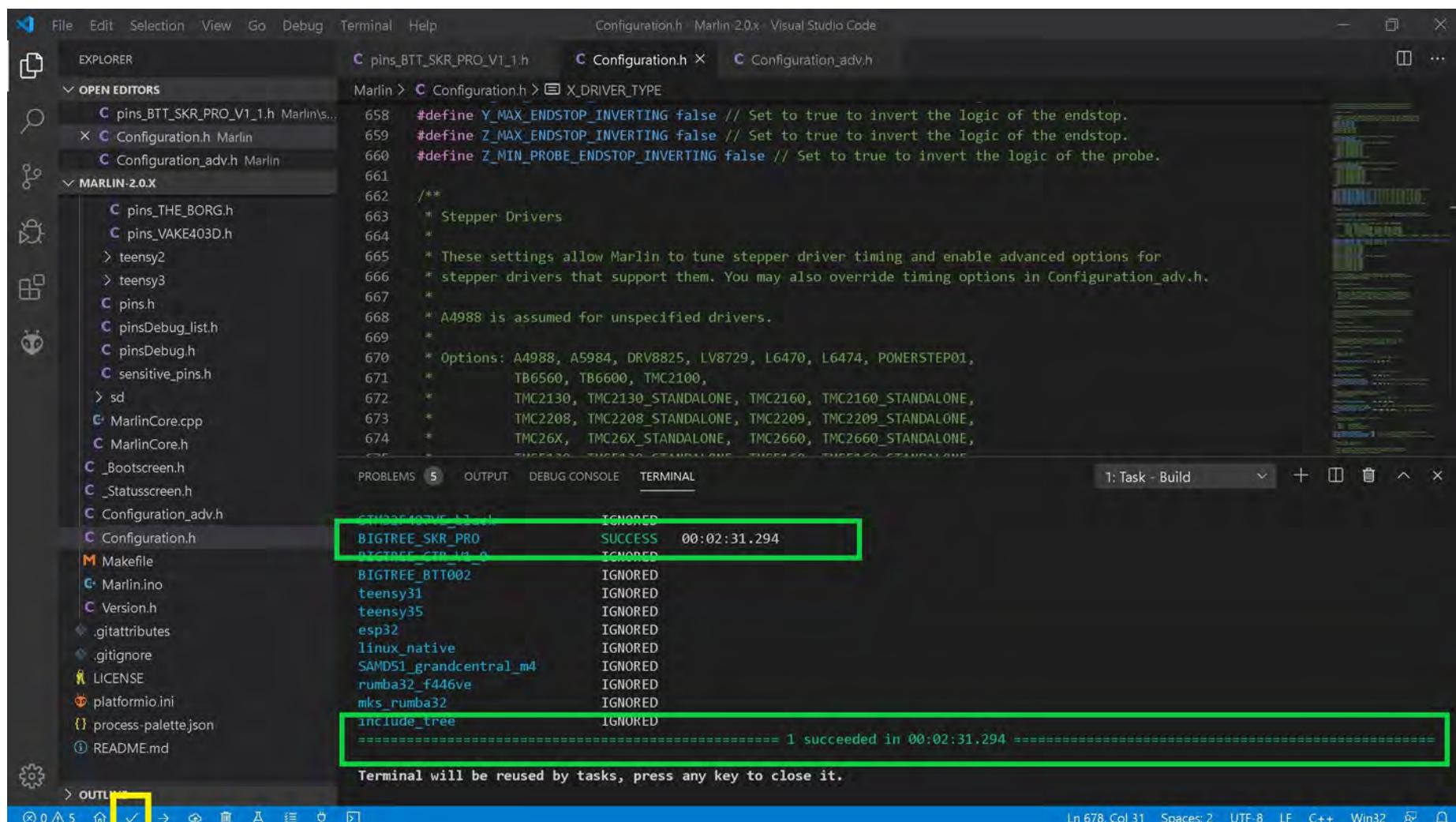
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // 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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is power

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers

- The end of Marlin setup for FYSETC S6128 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 2.0.x repository open. The left sidebar shows the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X'. The main editor area displays Configuration.h with code related to driver types and stepper drivers. Below the editor is a 'PROBLEMS' panel showing build status for various boards. The terminal at the bottom shows a successful build process:

```

CTM220F_TMC_12_V1_0: IGNORED
BIGTREE_SKR_PRO: SUCCESS 00:02:31.294
BIGTREE_CTP_12_V0: 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 =====

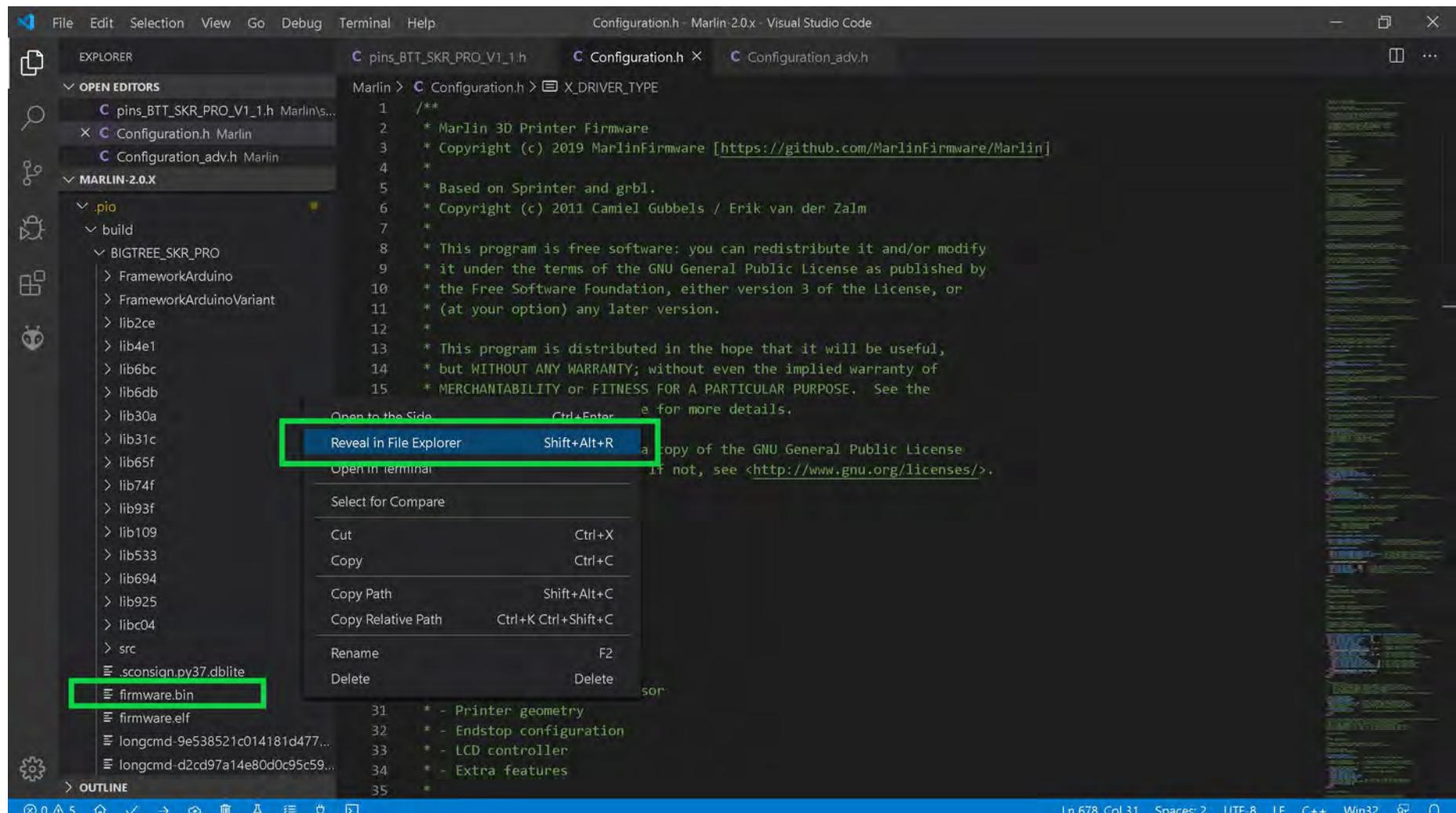
```

The terminal message 'Terminal will be reused by tasks, press any key to close it.' is visible 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 FYSETC S6128 V1.1 Drivers

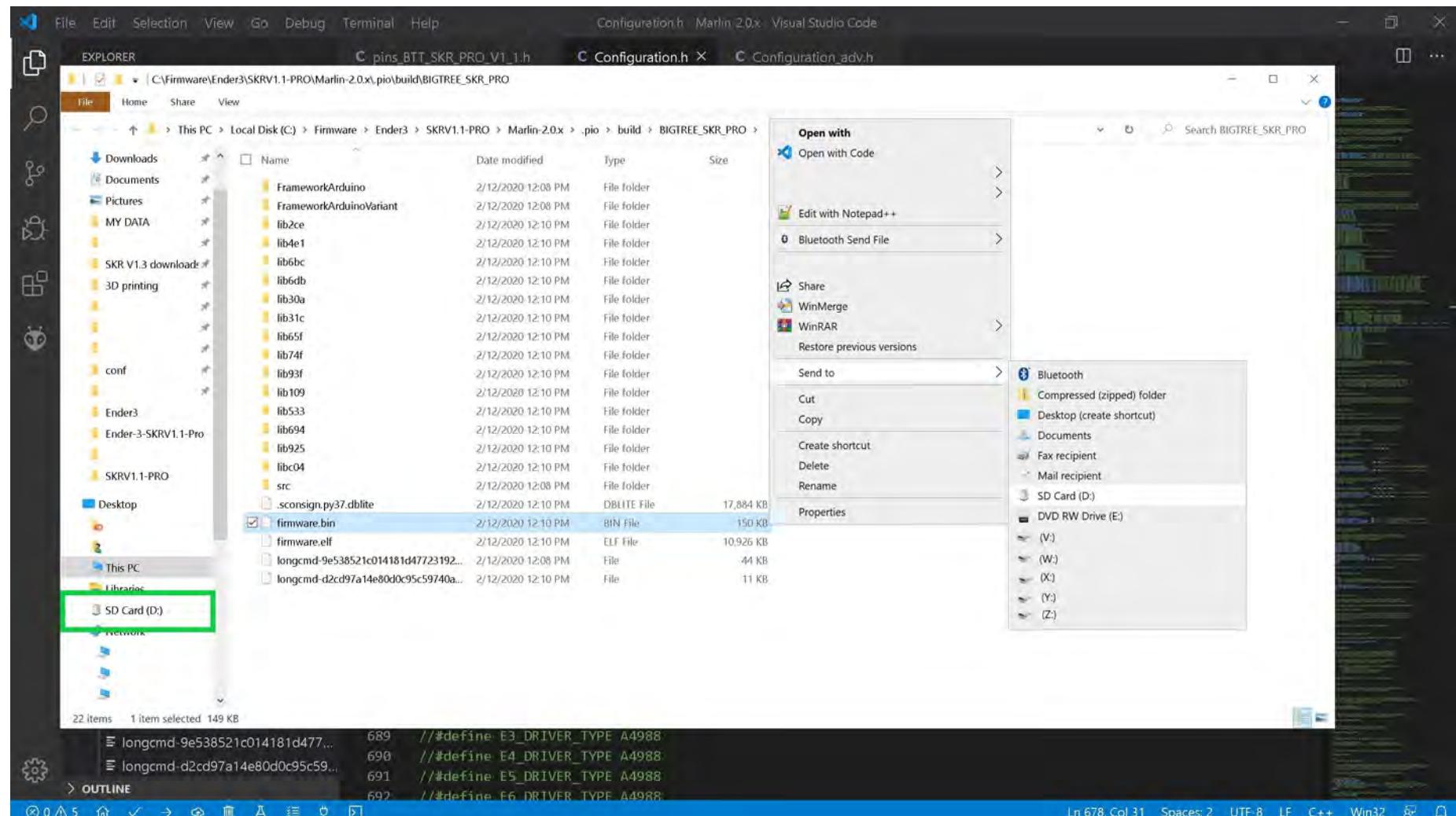
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer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



- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



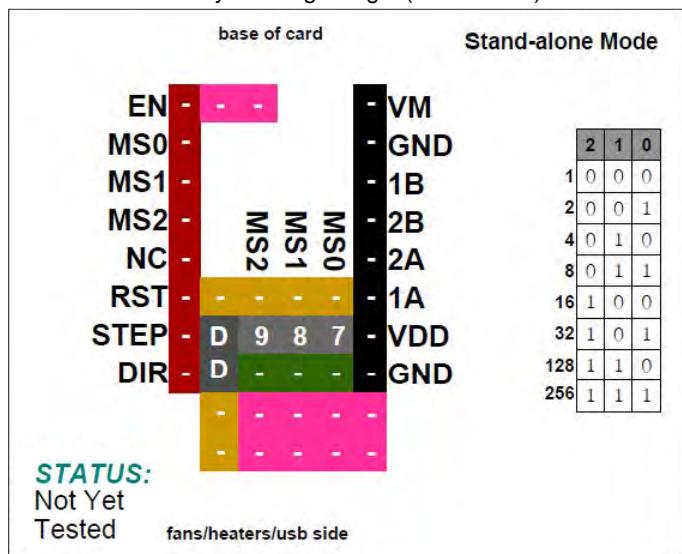
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

base of card			Stand-alone Mode		
EN	-	-	VM	-	GND
MS0	-	-	GND	-	1B
MS1	-	-	1B	-	2B
MS2	MS2	MS1	2B	-	2A
NC	-	-	2A	-	1A
RST	-	-	1A	-	VDD
STEP	D	9 8 7	VDD	-	GND
DIR	D	-	GND	-	-
STATUS:					
Not Yet Tested			fans/heaters/usb side		

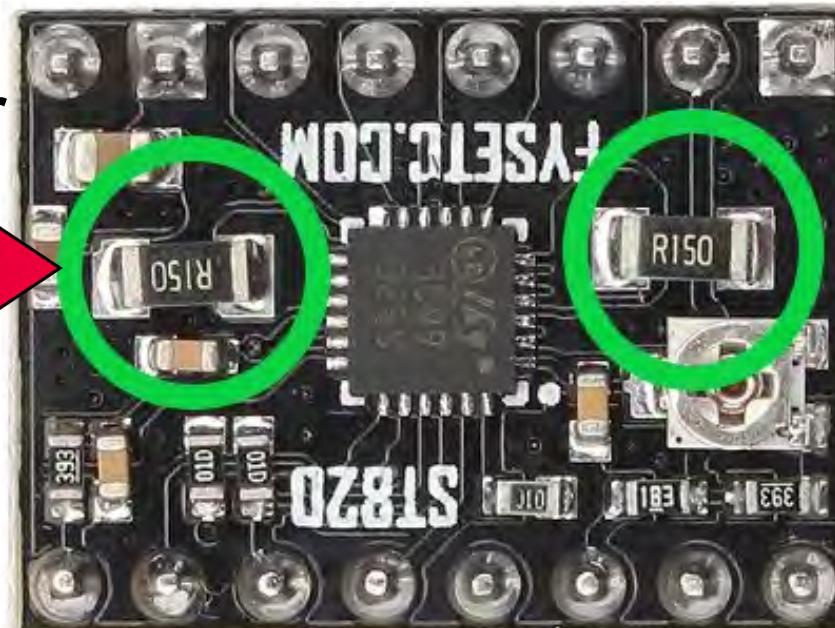
FYSETC ST820

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
FYSETC ST820	Low	Low	Low	Full step	2 Phase
Maximum 256 Subdivision	Low	Low	High	Half step	1-2 Phase
45V DC 1.5A (peak)	Low	High	Low	1/4 step	W1-2 Phase
	Low	High	High	1/8 step	2W1-2 Phase
	High	Low	Low	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	High	High	Low	1/128 step	16W1-2 Phase
	High	High	High	1/256 step	32W1-2 Phase
Driving Current Calculation Formula $V_{DD} = 3.3V \text{ or } 5V \text{ DC}$ $R_S (\text{Typical Sense Resistor}) = 0.15\Omega$	$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{R_S}$			$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * R_S$	



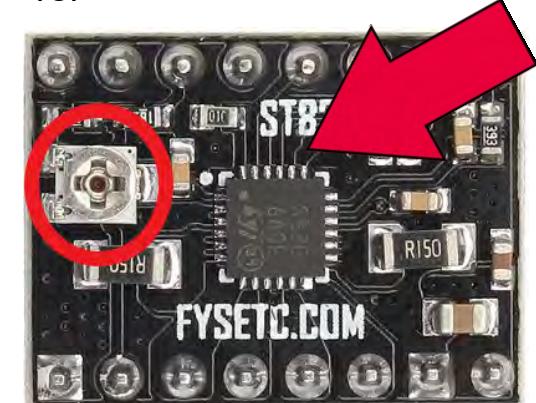
Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



FYSETC ST820

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



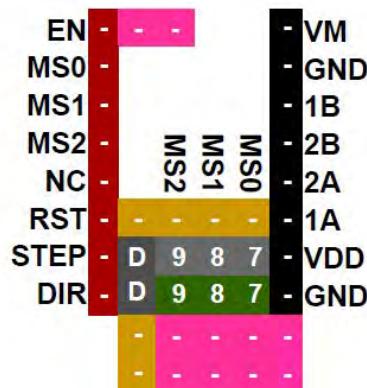
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

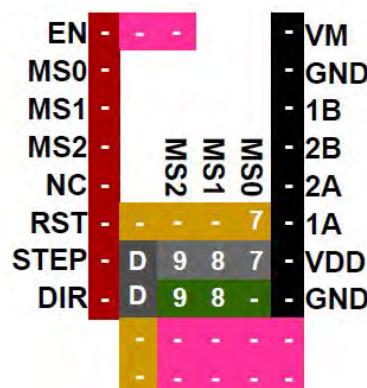
FYSETC ST820

Stand-alone Mode

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.



STEP



1 / 2

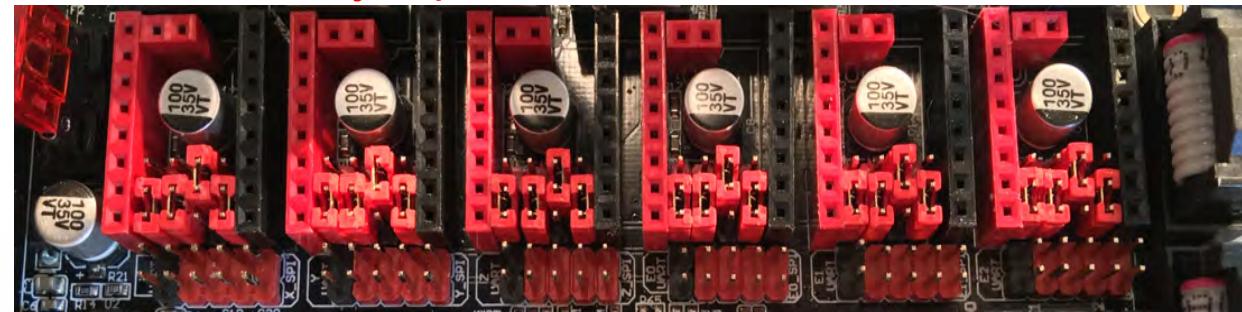
FYSETC ST820

Stand-alone Mode

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
MS0		GND
MS1		1B
MS2		2B
NC	MS2 MS1 MS0	2A
RST	- 8 -	1A
STEP	D 9 8 7	VDD
DIR	D 9 - 7	GND

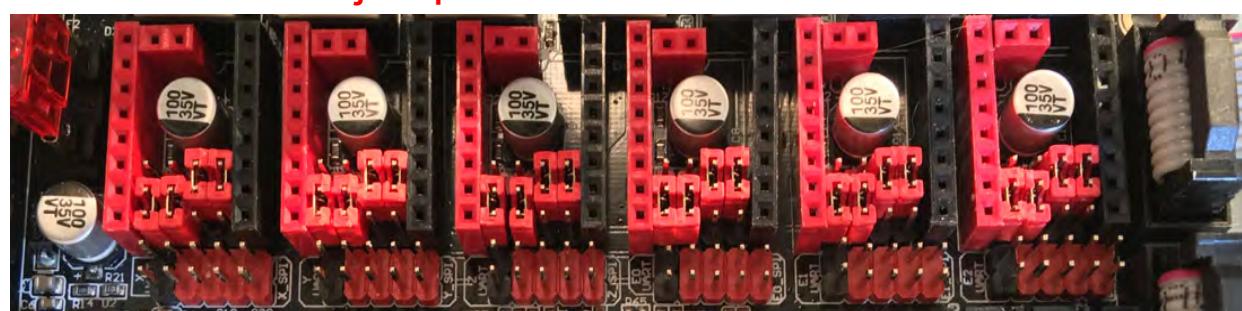
Note: The "D" jumper MUST be SET!



1 / 4

EN	--	VM
MS0		GND
MS1		1B
MS2		2B
NC	MS2 MS1 MS0	2A
RST	- 8 7 -	1A
STEP	D 9 8 7	VDD
DIR	D 9 - -	GND

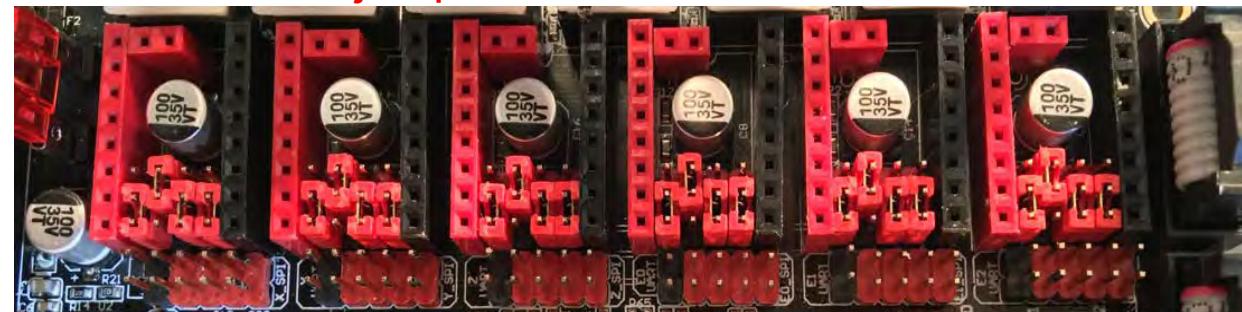
Note: The "D" jumper MUST be SET!



1 / 8

EN	--	VM
MS0		GND
MS1		1B
MS2		2B
NC	MS2 MS1 MS0	2A
RST	- 9 - -	1A
STEP	D 9 8 7	VDD
DIR	D 8 7 -	GND

Note: The "D" jumper MUST be SET!



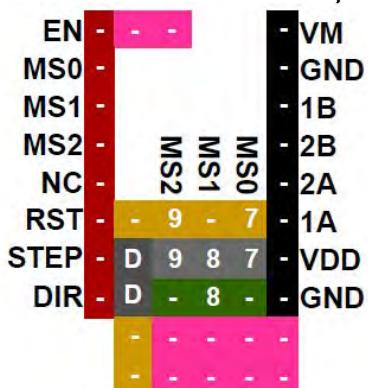
1 / 16

FYSETC ST820

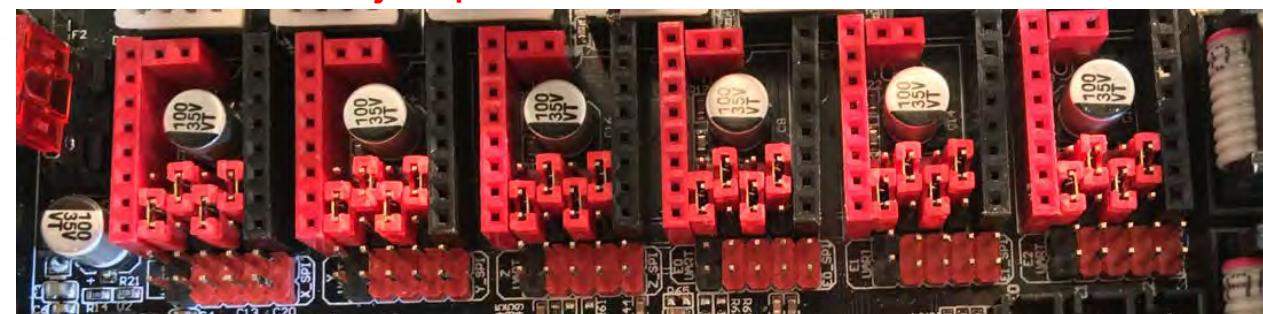
Stand-alone Mode

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.

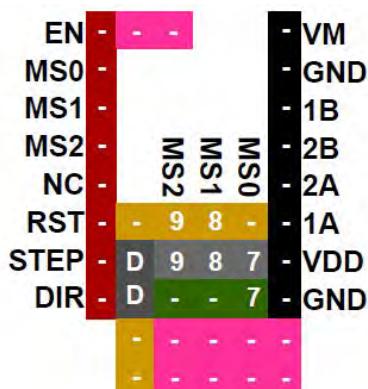
1 / 32



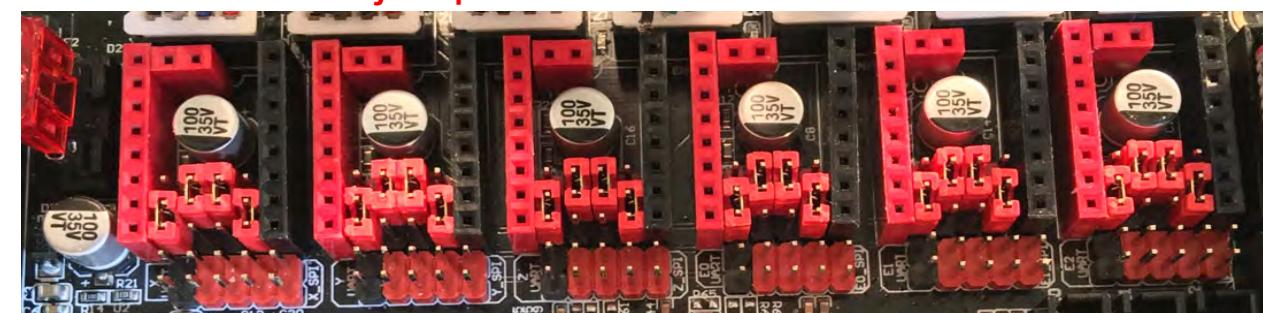
Note: The "D" jumper MUST be SET!



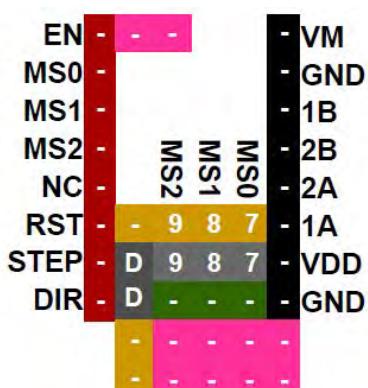
1 / 128



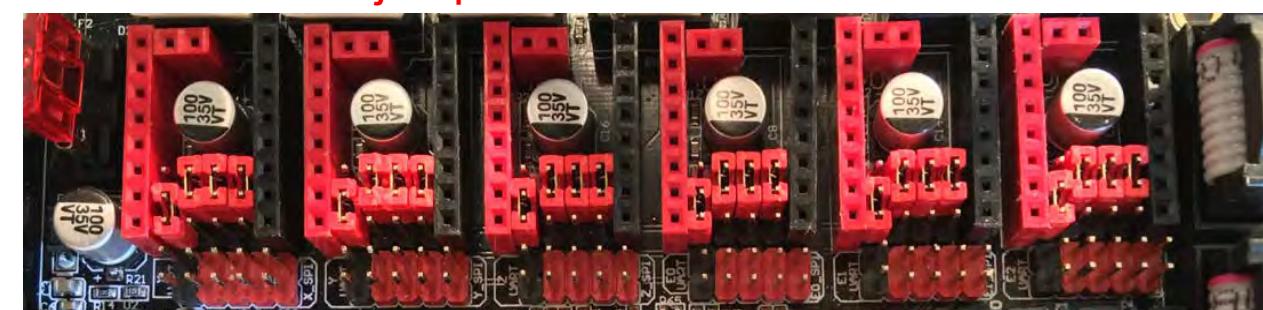
Note: The "D" jumper MUST be SET!



1 / 256



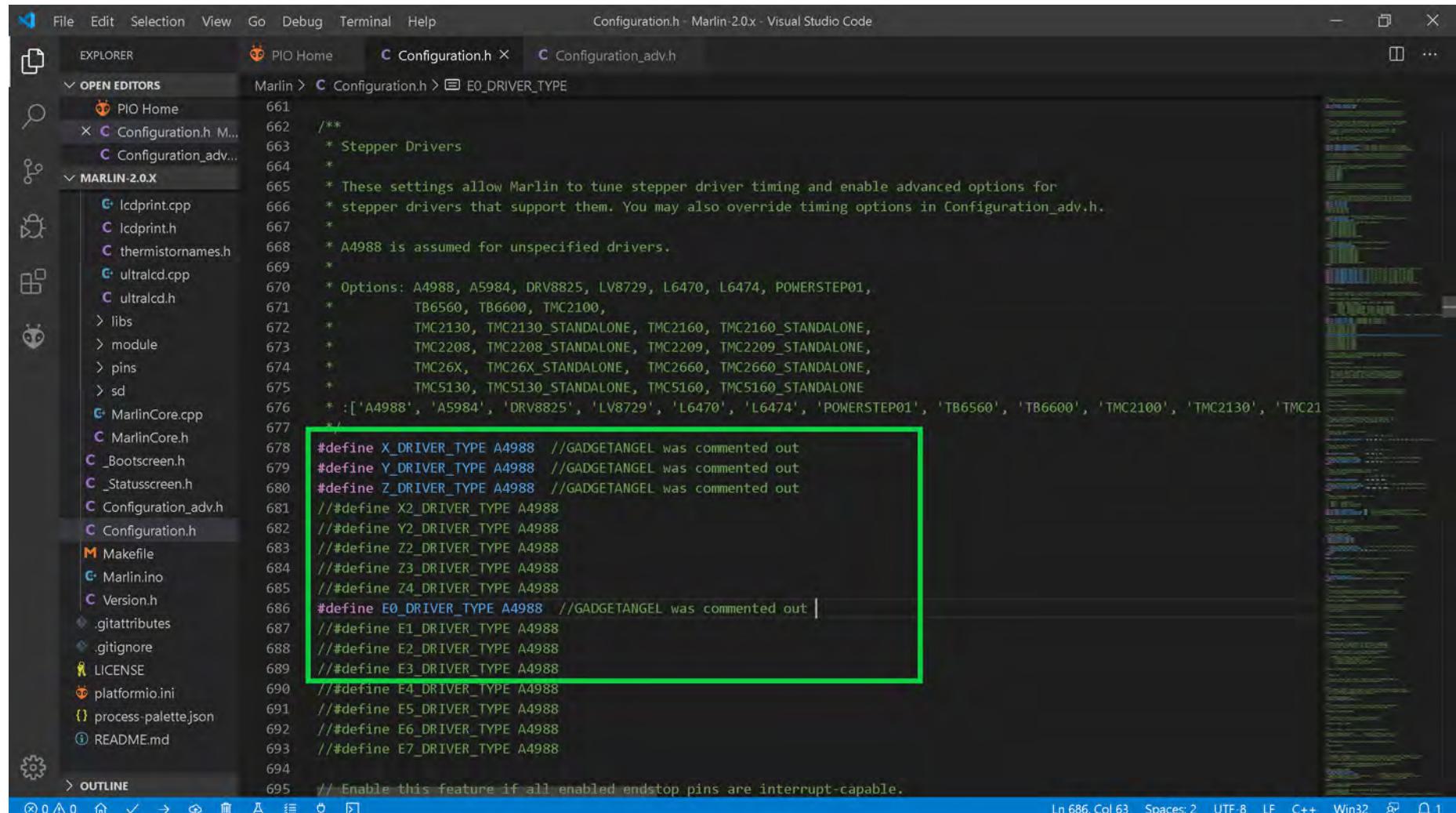
Note: The "D" jumper MUST be SET!



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

NOTE: Go to Appendix C, and 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 Configuration_adv.h
Marlin > Configuration.h > EO_DRIVER_TYPE

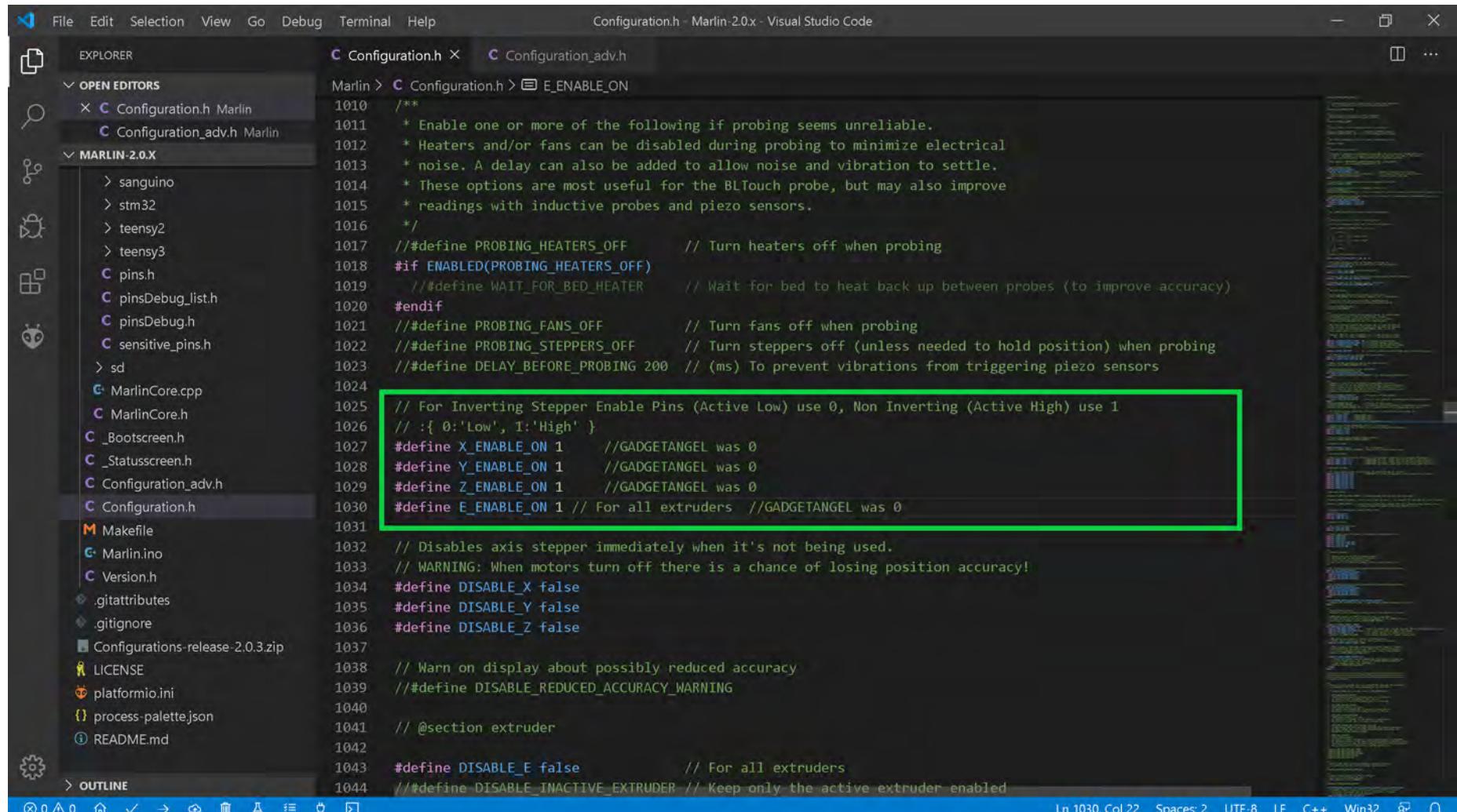
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: 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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
677 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
678 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
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 A4988 //GADGETANGEL was commented out |
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.

```

- 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.



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

```

1010 /**
1011 * Enable one or more of the following if probing seems unreliable.
1012 * Heaters and/or fans can be disabled during probing to minimize electrical
1013 * noise. A delay can also be added to allow noise and vibration to settle.
1014 * These options are most useful for the BLTouch probe, but may also improve
1015 * readings with inductive probes and piezo sensors.
1016 */
1017 // #define PROBING_HEATERS_OFF // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019 // #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 // #define PROBING_FANS_OFF // Turn fans off when probing
1022 // #define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1023 // #define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // :{ 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1 // GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1 // GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1 // GADGETANGEL was 0
1030 #define E_ENABLE_ON 1 // For all extruders // GADGETANGEL was 0
1031
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

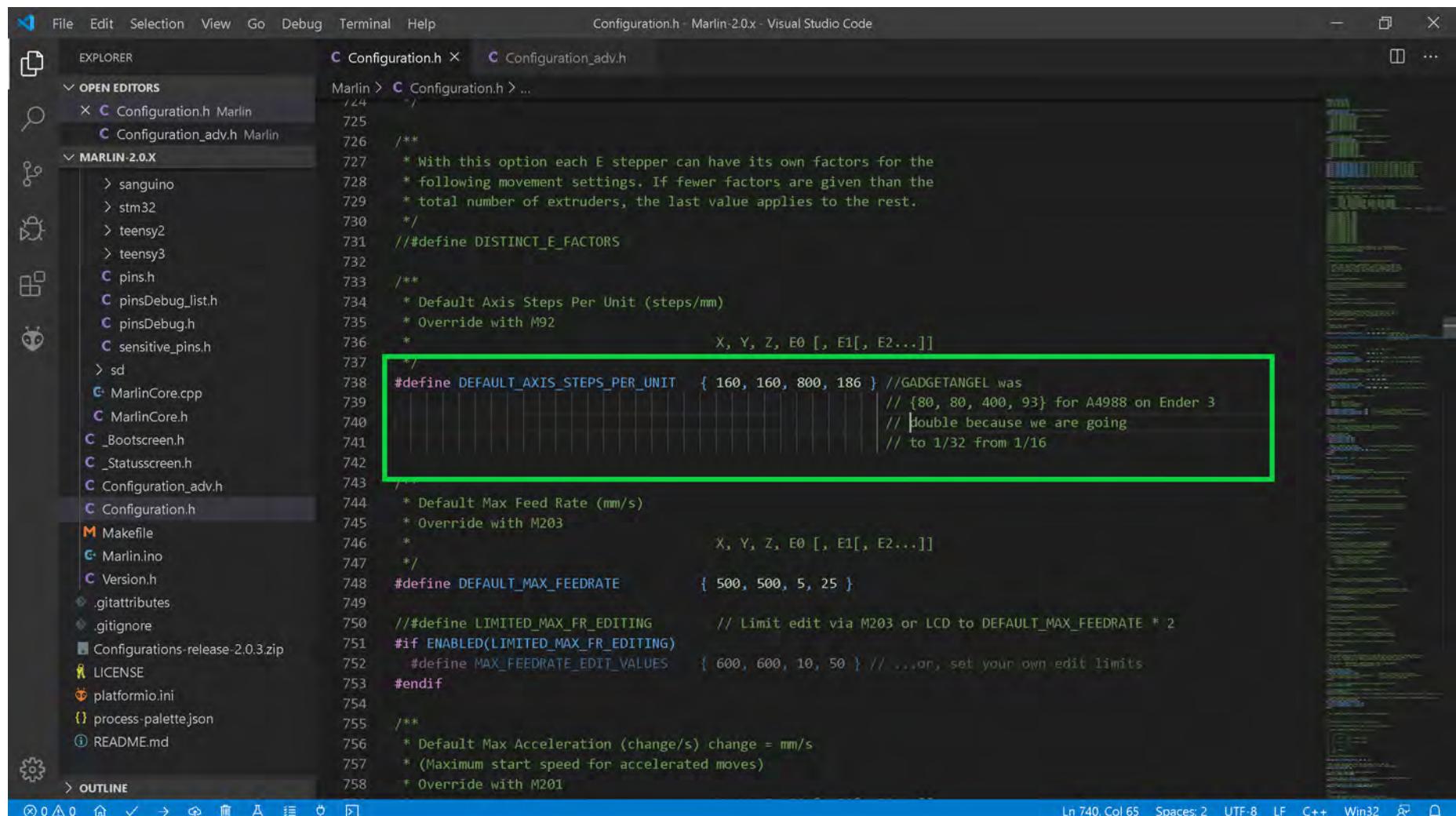
```

Ln 1030, Col 22 Spaces: 2 UTF-8 LF C++ Win32

- 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.



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 line of code:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates: Ln 740, Col 65, 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 OPEN EDITORS Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```

- 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 OPEN EDITORS MARLIN-2.0.X

C pins_BTT_SKR_PRO_V1_1.h C Configuration.h X Configuration_adv.h

Marlin > C 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 OUTPUT DEBUG CONSOLE TERMINAL

1: Task - Build

CTH2540TVE_11_1	IGNORED	
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
BTCTREE_CTC_12_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	

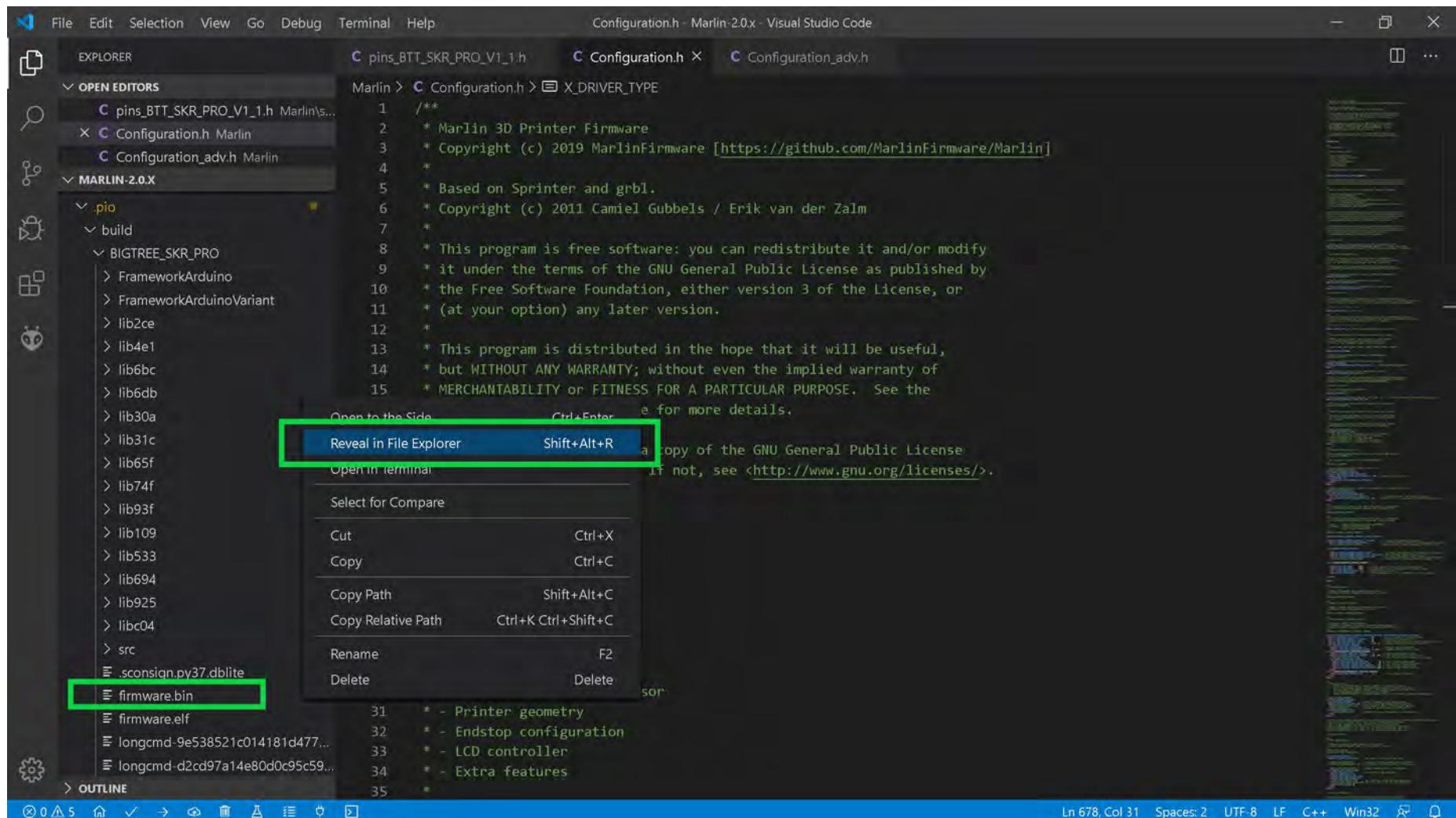
```
===== 1 succeeded in 00:02:31.294 =====
```

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 ST820 Drivers

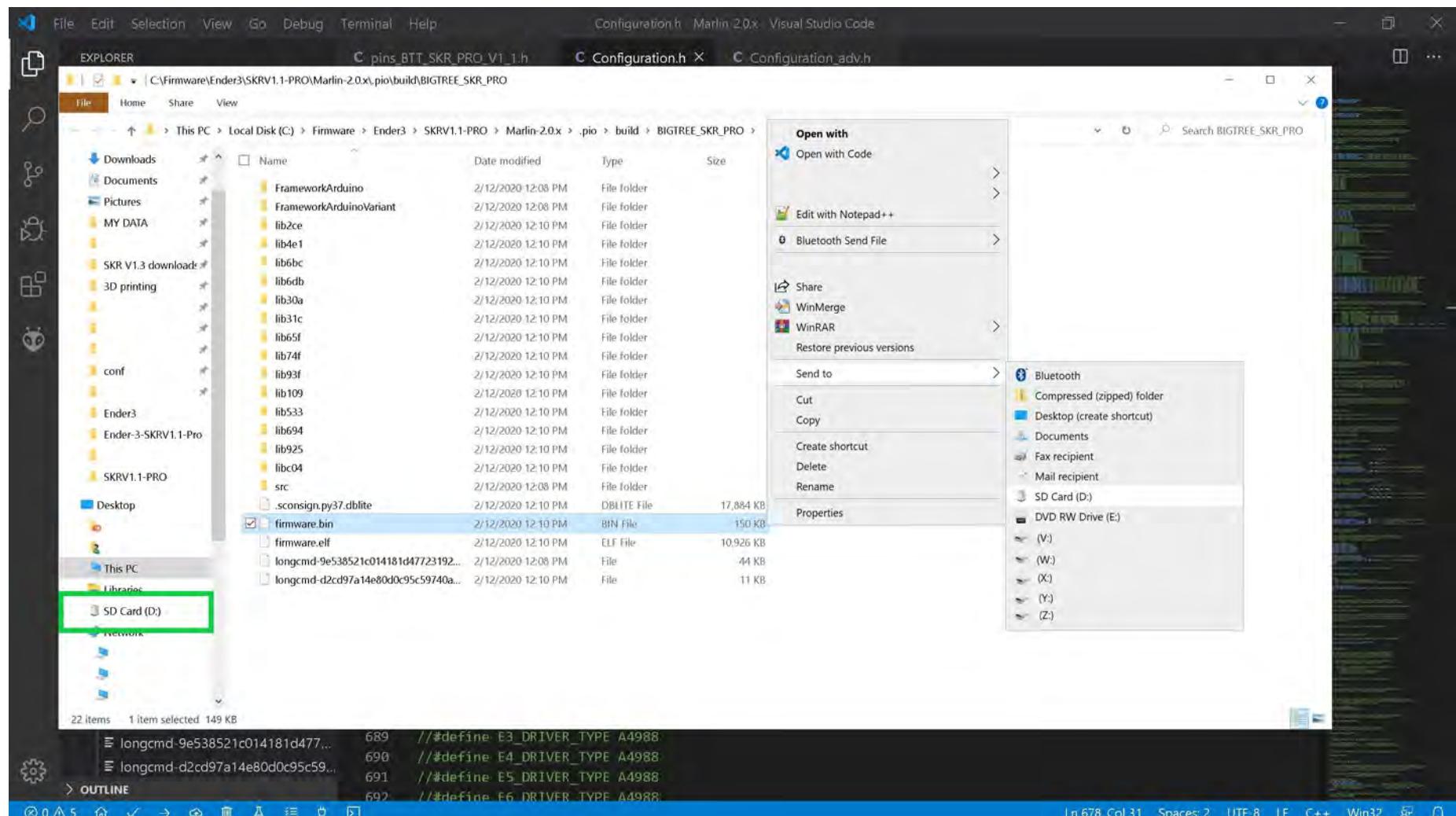
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



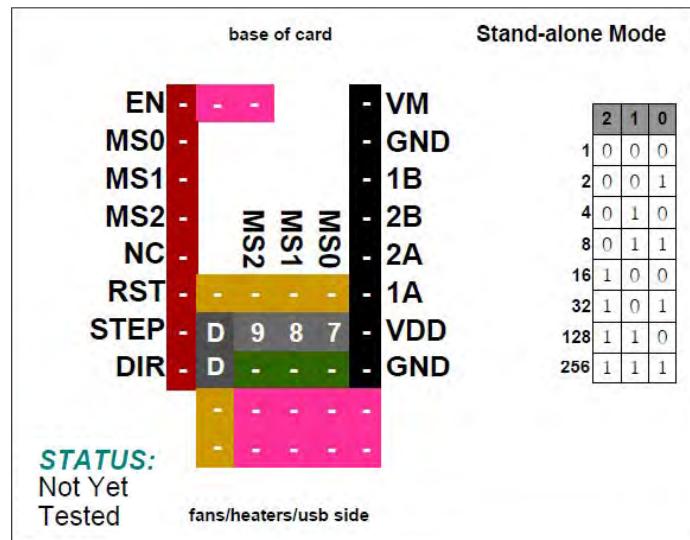
- Go to the next page.

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

- Ensure your micro SD card, which 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".

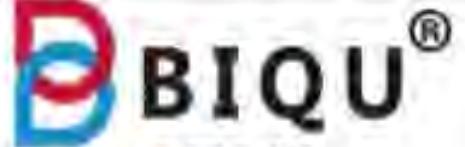


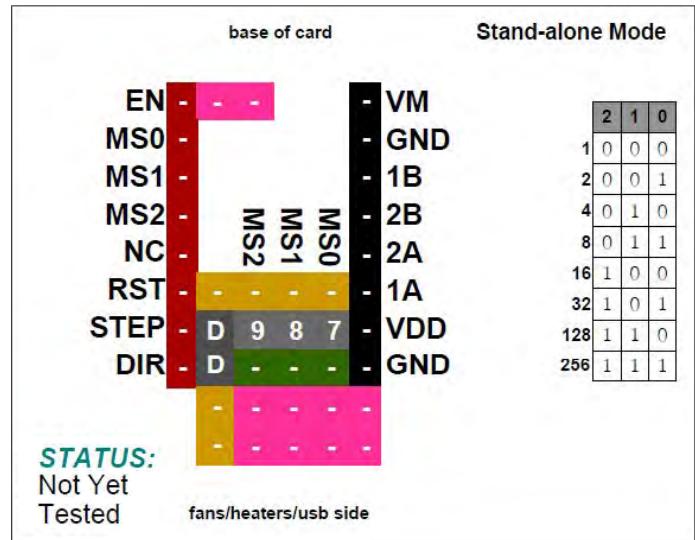
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



BIQU ST820

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
	Low	Low	Low	Full step	2 Phase
 BIQU® ST820	Low	Low	High	Half step	1-2 Phase
Maximum 256 Subdivision	Low	High	Low	1/4 step	W1-2 Phase
45V DC	Low	High	High	1/8 step	2W1-2 Phase
1.5A (peak)	High	Low	Low	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	High	High	Low	1/128 step	16W1-2 Phase
	High	High	High	1/256 step	32W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{R_S}$		$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * R_S$		
$V_{DD} = 3.3\text{ V or }5\text{ V DC}$					
R_S (Typical Sense Resistor) = 0.15Ω					

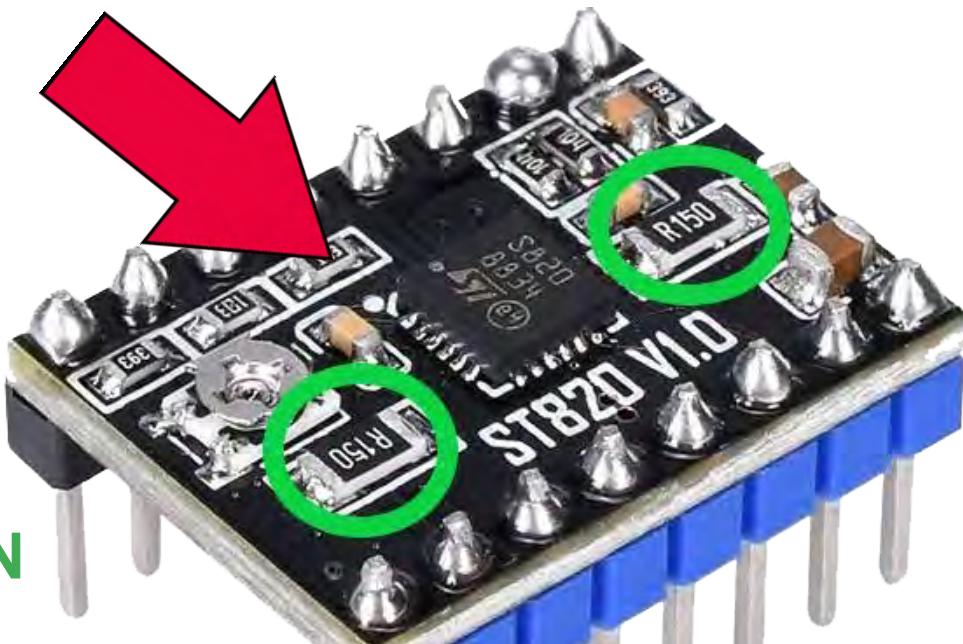


BIQU ST820

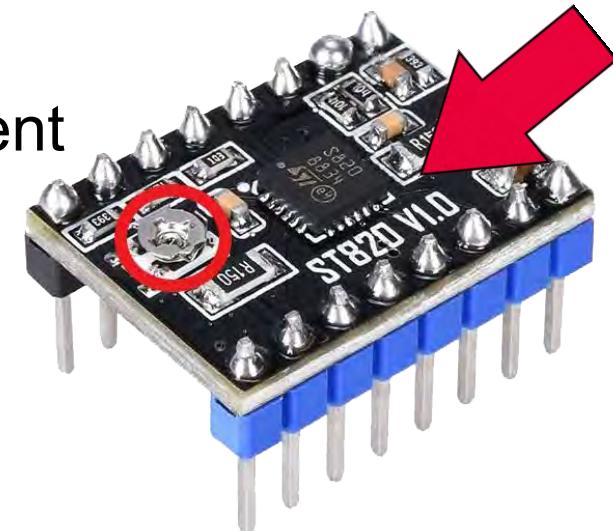
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



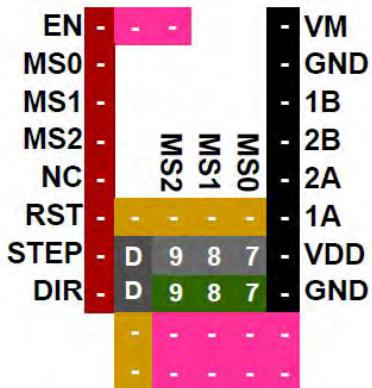
Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>



- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

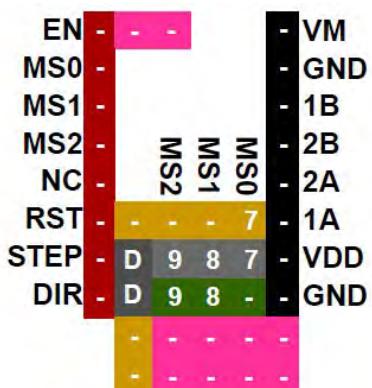
BIQU ST820**Stand-alone Mode**

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.



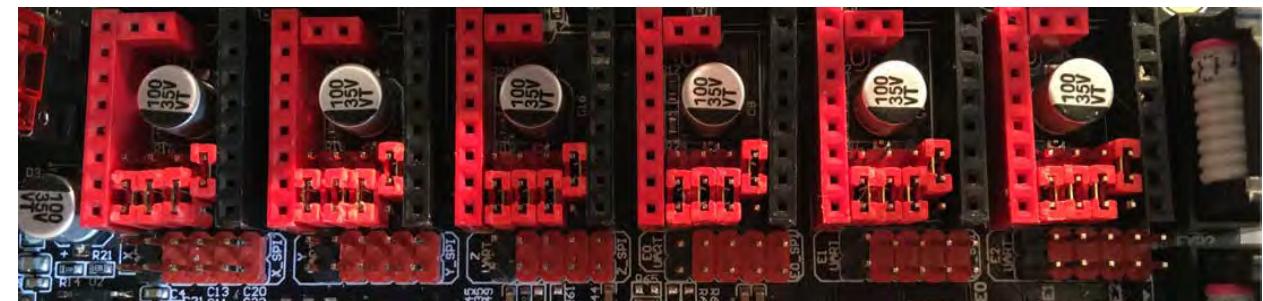
STEP

Note: The "D" jumper MUST be SET!



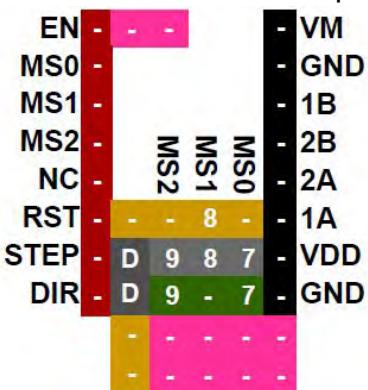
1 / 2

Note: The "D" jumper MUST be SET!



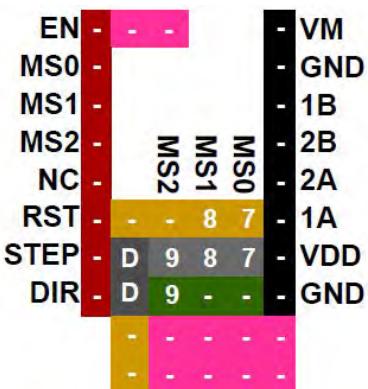
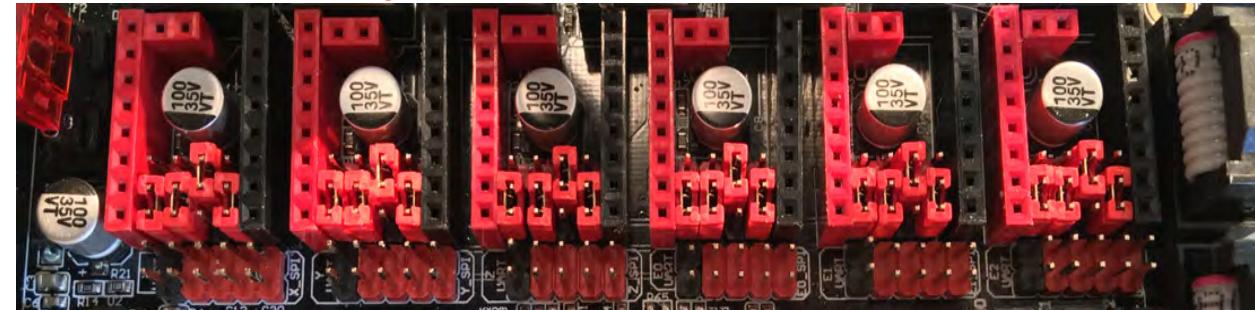
BIQU ST820**Stand-alone Mode**

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.



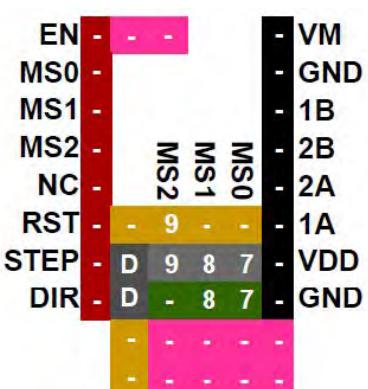
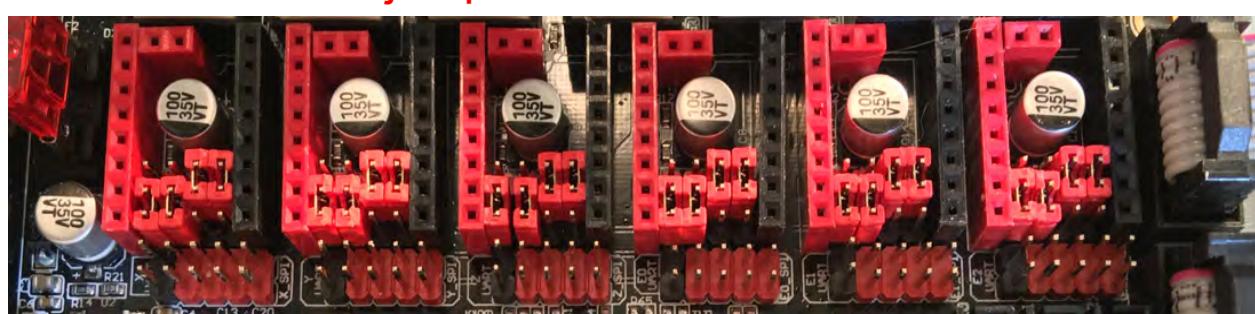
1 / 4

Note: The "D" jumper MUST be SET!



1 / 8

Note: The "D" jumper MUST be SET!



1 / 16

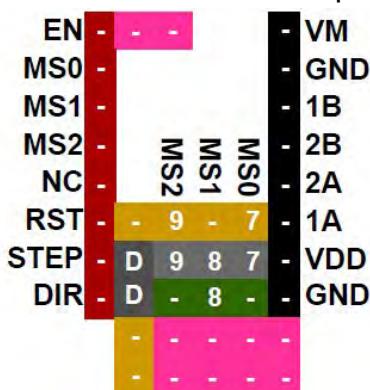
Note: The "D" jumper MUST be SET!



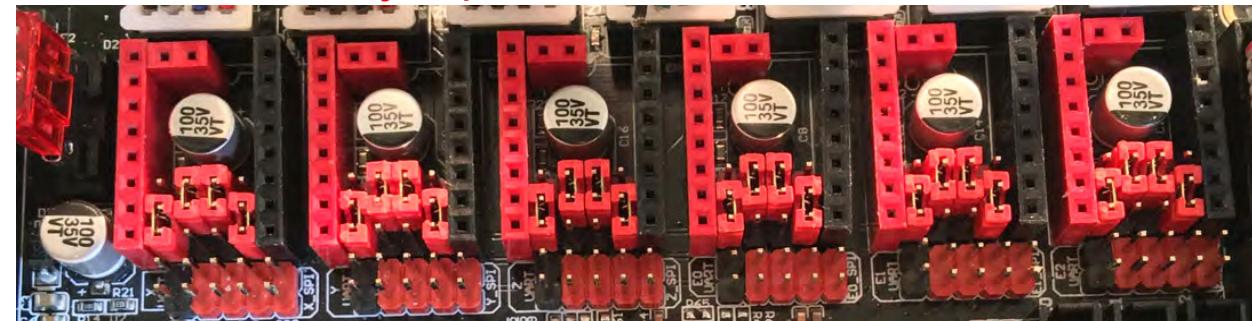
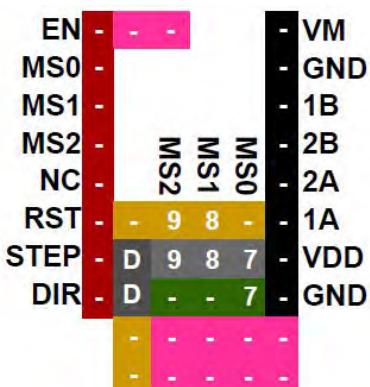
BIQU ST820**Stand-alone Mode**

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.

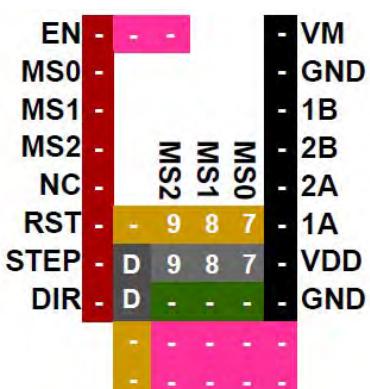
1 / 32



1 / 128



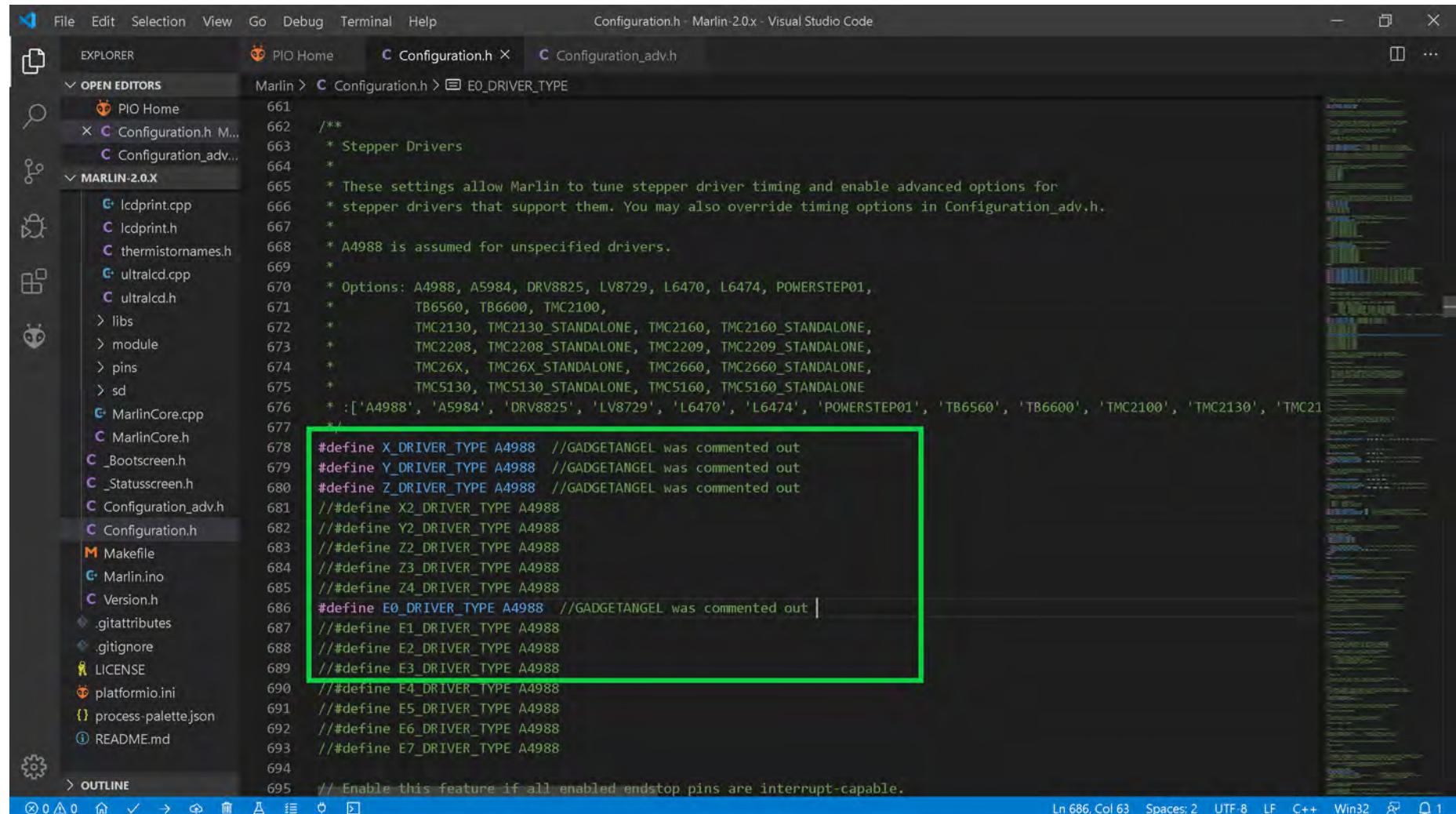
1 / 256



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

NOTE: Go to Appendix C, and 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 file `Configuration.h` open. The code editor displays the following snippet of C++ code:

```

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',
676 *           'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160'],
677 */
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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 section of code from line 678 to line 686, which defines the driver type for each axis (X, Y, Z, and E0). The line `#define E0_DRIVER_TYPE A4988` is preceded by a vertical bar character, indicating it was previously commented out.

- 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.

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

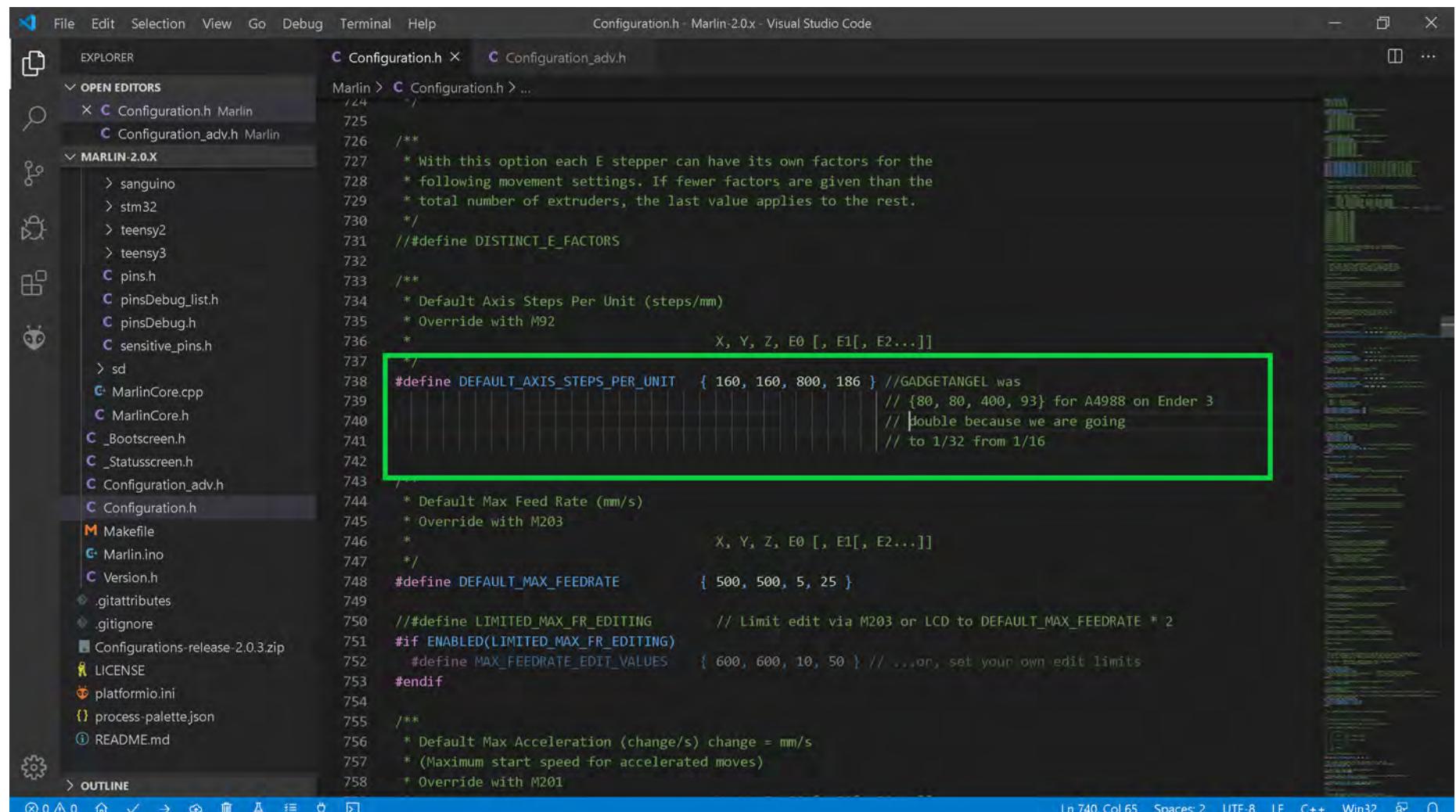
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  C pins.h
  C pinsDebug_list.h
  C pinsDebug.h
  C sensitive_pins.h
  > sd
  C MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C _Statusscreen.h
  C Configuration_adv.h
  C Configuration.h
  M Makefile
  G Marlin.ino
  C Version.h
  .gitattributes
  .gitignore
  Configurations-release-2.0.3.zip
  L LICENSE
  platformio.ini
  process-palette.json
  README.md
  > OUTLINE

Marlin > Configuration.h > E_ENABLE_ON
1010  /**
1011   * Enable one or more of the following if probing seems unreliable.
1012   * Heaters and/or fans can be disabled during probing to minimize electrical
1013   * noise. A delay can also be added to allow noise and vibration to settle.
1014   * These options are most useful for the BLTouch probe, but may also improve
1015   * readings with inductive probes and piezo sensors.
1016   */
1017 // #define PROBING_HEATERS_OFF      // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019   // #define WAIT_FOR_BED_HEATER    // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 // #define PROBING_FANS_OFF        // Turn fans off when probing
1022 // #define PROBING_STEPPERS_OFF    // Turn steppers off (unless needed to hold position) when probing
1023 // #define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // :{ 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1           // GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1           // GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1           // GADGETANGEL was 0
1030 #define E_ENABLE_ON 1           // For all extruders // GADGETANGEL was 0
1031
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
```

- 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.



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

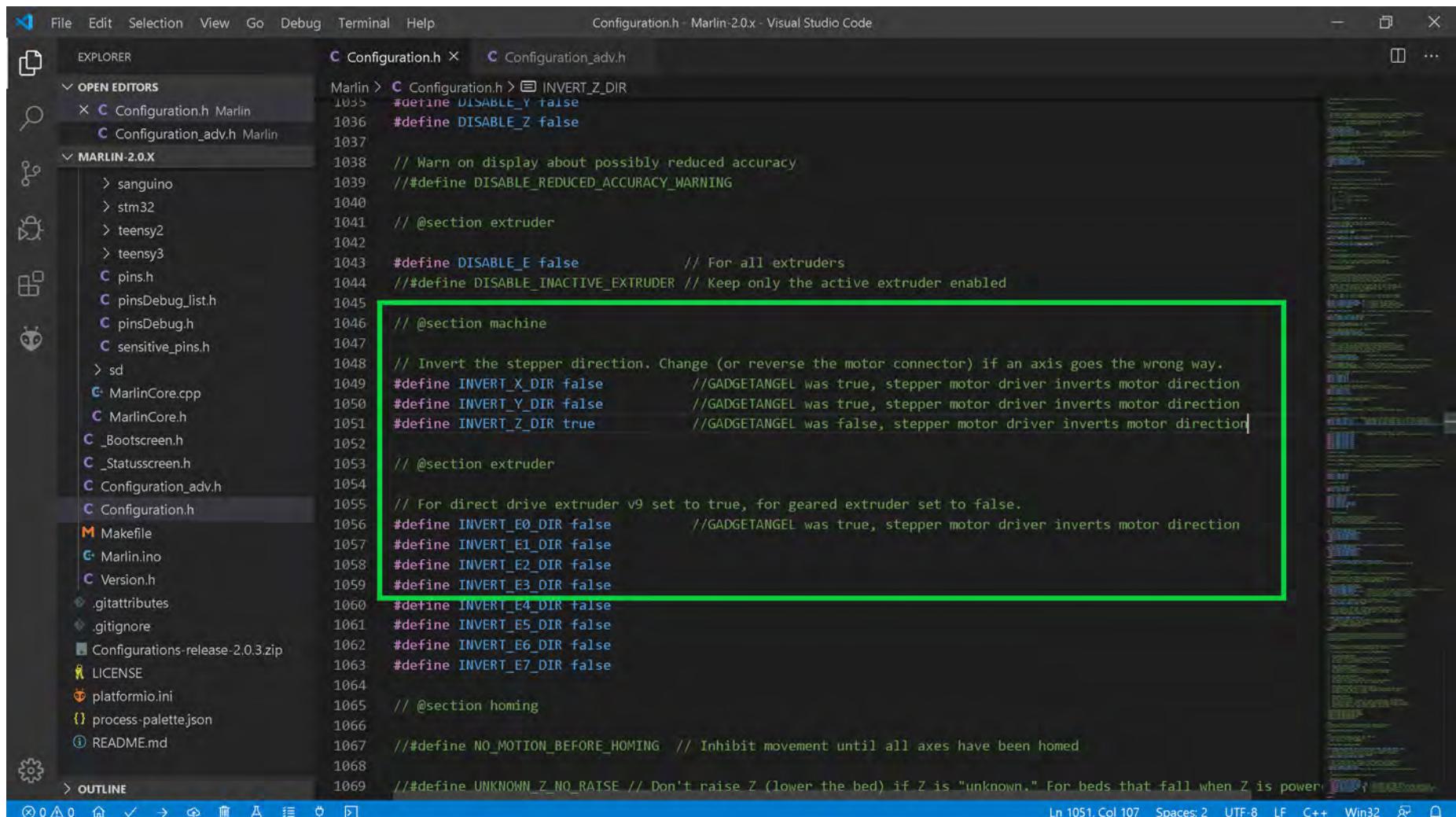
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right indicates: Ln 740, Col 65, 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



```

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

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
Marlin > Configuration.h Marlin
Marlin > Configuration_adv.h Marlin
MARLIN-2.0.X
> sanguino
> stm32
> 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
Configurations-release-2.0.3.zip
LICENSE
platformio.ini
process-palettejson
README.md
OUTLINE
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 🔍

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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- 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 following details:

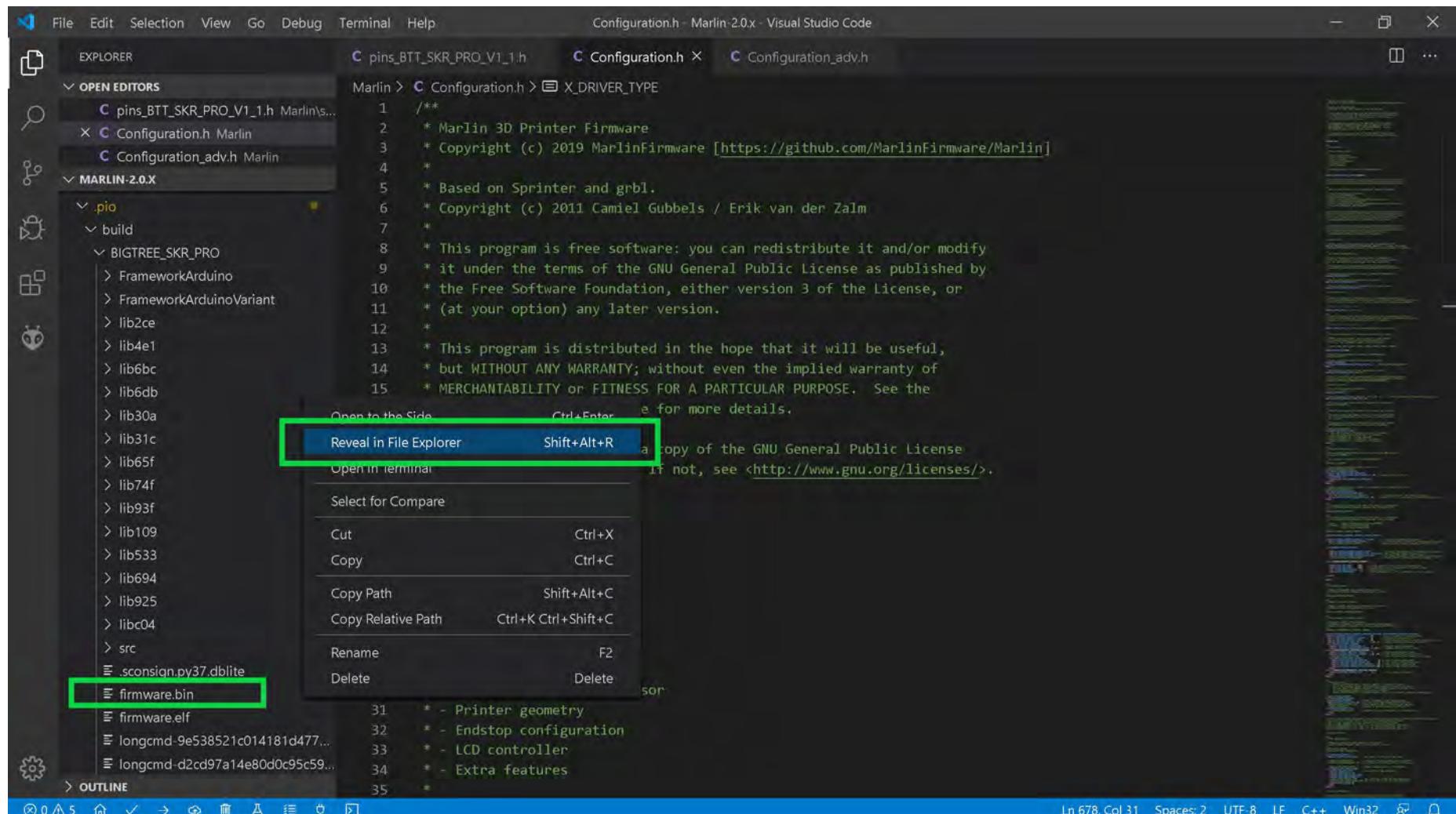
- File Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editor:** Displays the content of Configuration.h, specifically the X_DRIVER_TYPE section.
- Terminal:** Shows the build output. A yellow box highlights the "COMPILE" button in the toolbar. A green box highlights the terminal output, which includes the following text:


```
CTM20E403DVE_1.1.1          IGNORED
BIGTREE_SKR_PRO                SUCCESS 00:02:31.294
BIGTREE_ST8_1.0                 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 =====
```
- Output:** Shows the terminal message: "Terminal will be reused by tasks, press any key to close it."
- Status Bar:** Shows line count (Ln 678), column count (Col 31), spaces (Spaces: 2), encoding (UTF-8), line separator (LF), language (C++), and operating system (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 ST820 Drivers

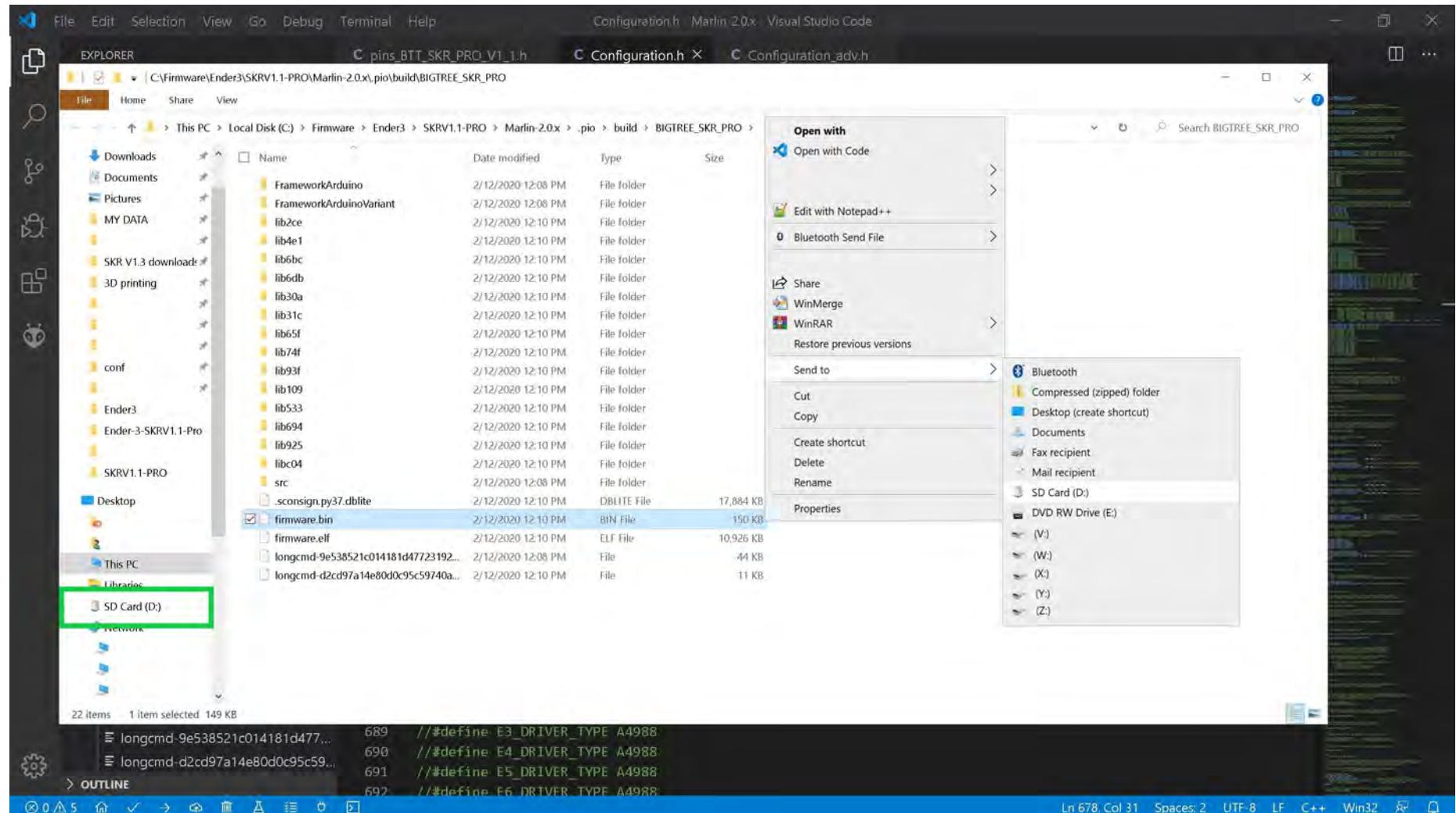
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



- Go to the next page.

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

- Ensure your micro SD card, which 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".



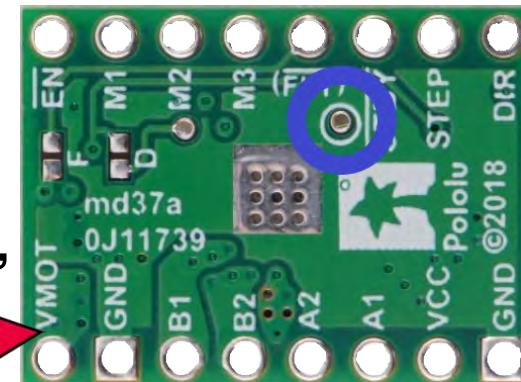
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

	base of card	Stand-alone Mode
ENABLE	- - -	VMOT
MODE1	-	GND
MODE2	-	B1
MODE3	-	B2
FAULT	- - -	A2
STBY	- - -	A1
STEP	D 9 8 7	VCC
DIR	D - -	GND
STATUS:	Not Yet Tested	
	fans/heaters/usb side	

POLOLU ST820 (STSPIN820)

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

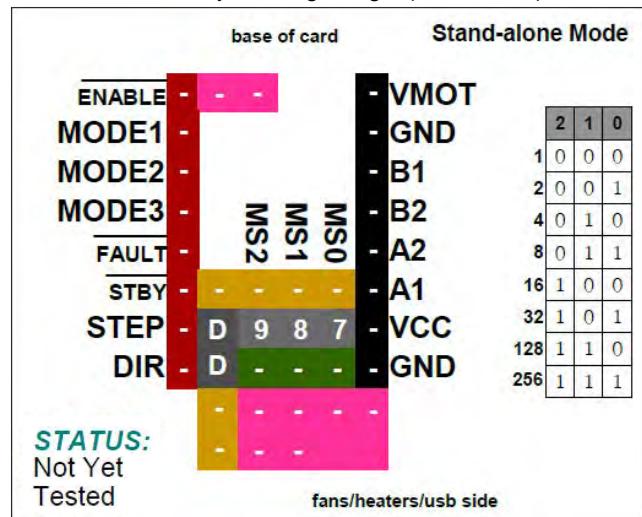
Note: "V_{ref} Test point" location is on the bottom of the driver board, as shown in BLUE



Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu ST820	Low	Low	Low	Full step	2 Phase
Maximum 256 Subdivision	Low	Low	High	Half step	1-2 Phase
45V DC	Low	High	Low	1/4 step	W1-2 Phase
1.5A (peak)	Low	High	High	1/8 step	2W1-2 Phase
Driving Current Calculation Formula	High	Low	Low	1/16 step	4W1-2 Phase
R_S (Typical Sense Resistor)= 0.2Ω	High	Low	High	1/32 step	8W1-2 Phase
	High	High	Low	1/128 step	16W1-2 Phase
	High	High	High	1/256 step	32W1-2 Phase

$$I_{MAX} = V_{ref} * \frac{1}{R_S}$$

$$V_{ref} = I_{MAX} * R_S$$

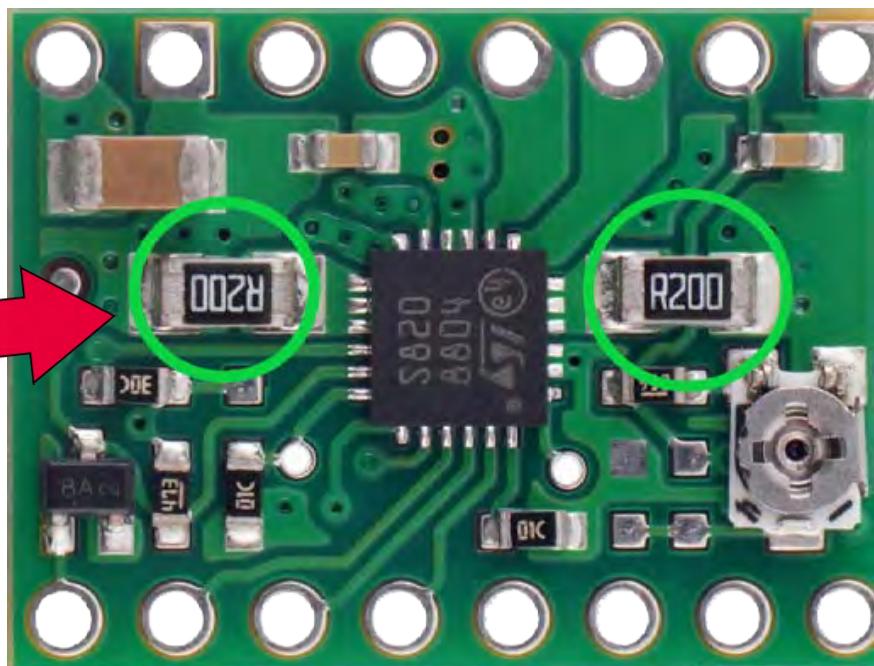


POLOLU ST820 (STSPIN820)

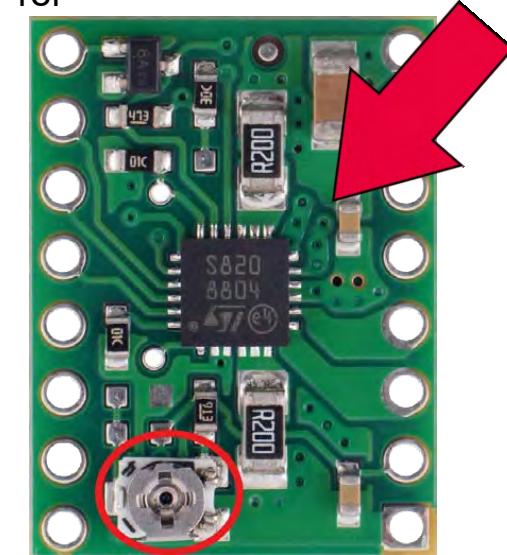
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



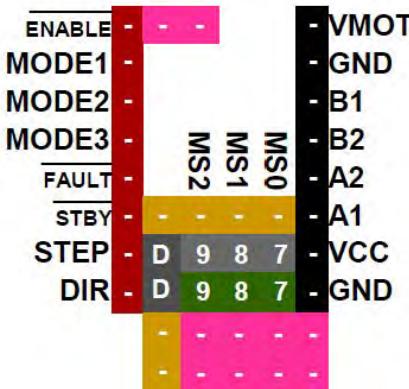
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>



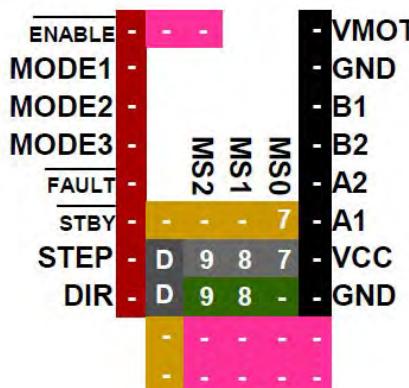
- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

Stand-alone Mode

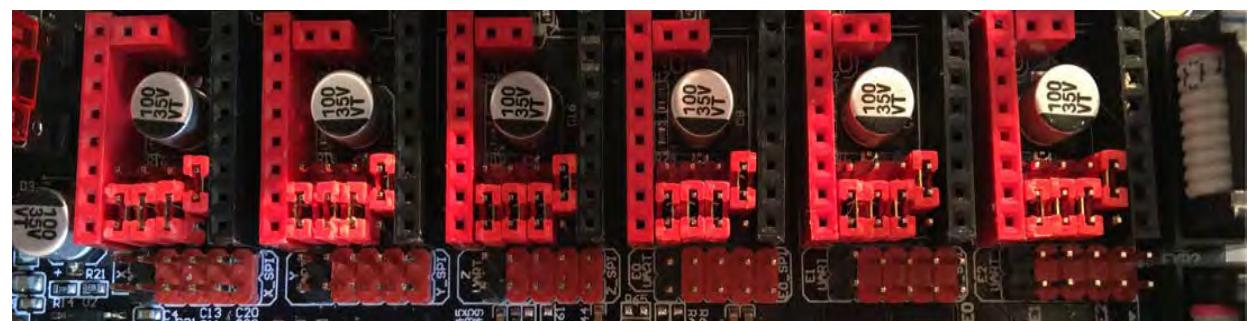
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

**STEP**

Note: The "D" jumper MUST be SET!

**1 / 2**

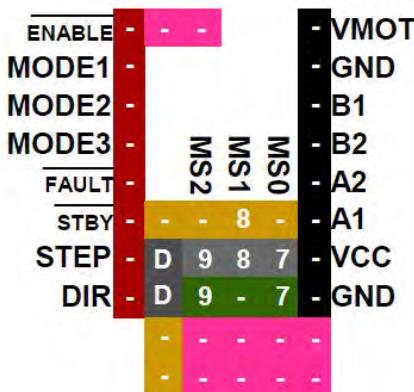
Note: The "D" jumper MUST be SET!



POLOLU ST820 (STSPIN820)

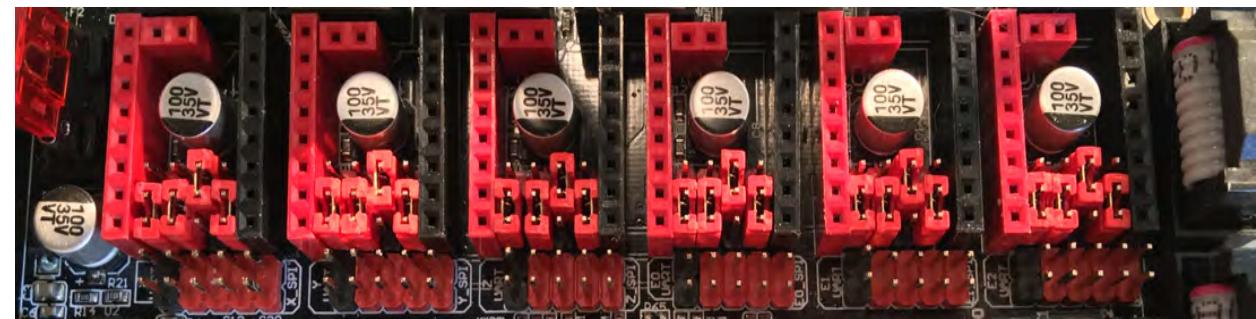
Stand-alone Mode

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.

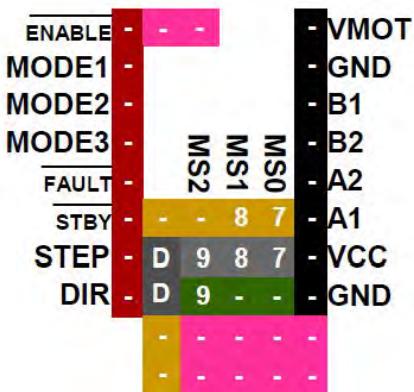


1 / 4

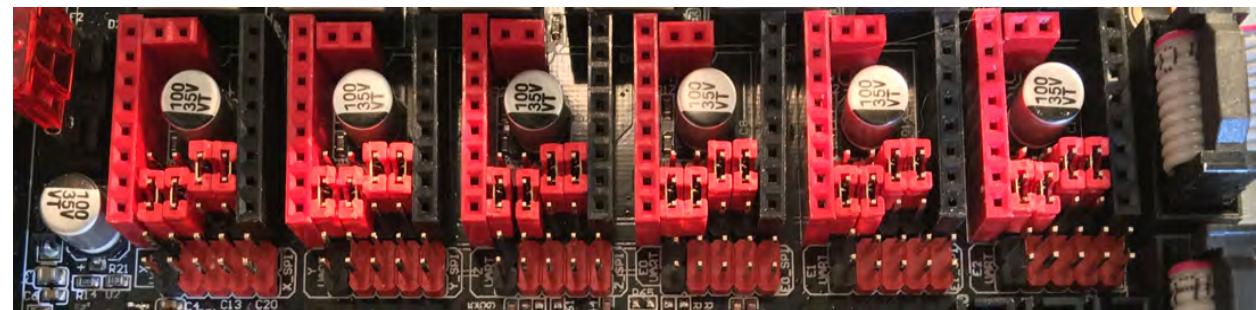
Note: The "D" jumper MUST be SET!



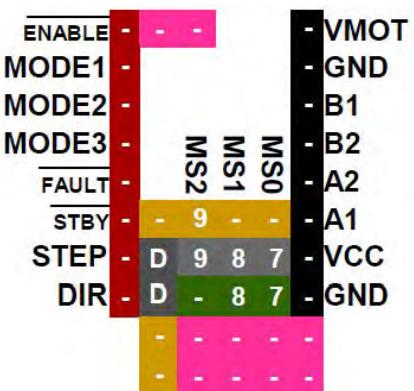
1 / 8



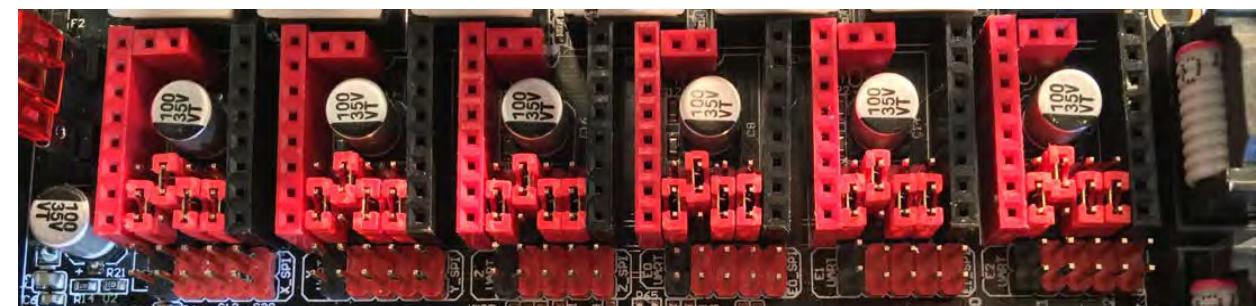
Note: The "D" jumper MUST be SET!



1 / 16



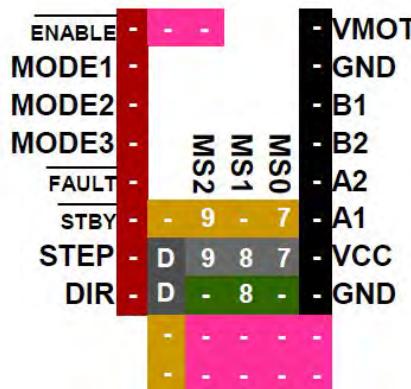
Note: The "D" jumper MUST be SET!



POLOLU ST820 (STSPIN820)

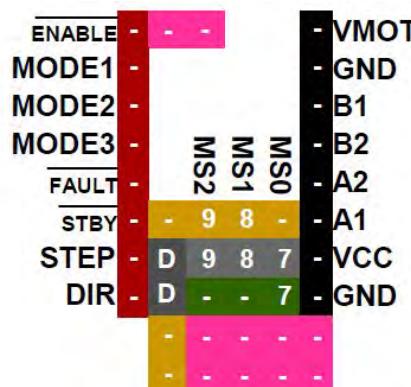
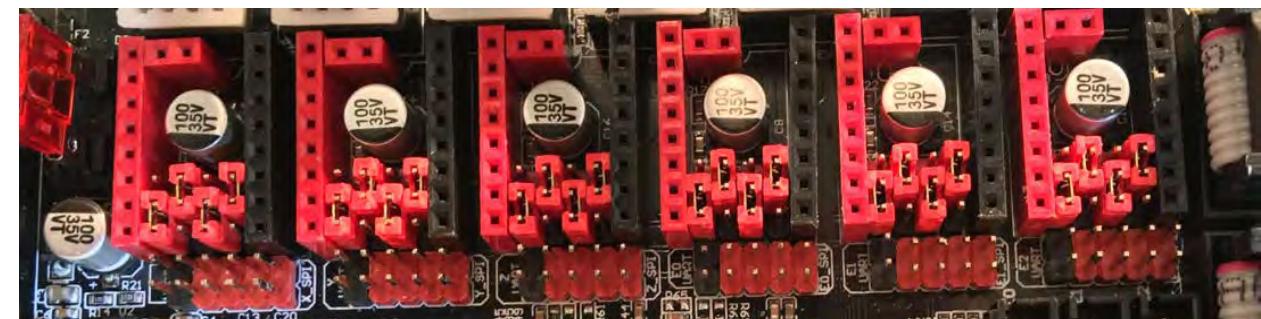
Stand-alone Mode

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.



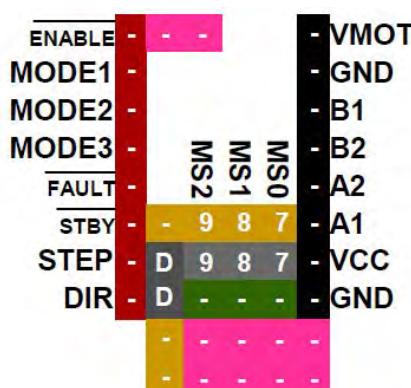
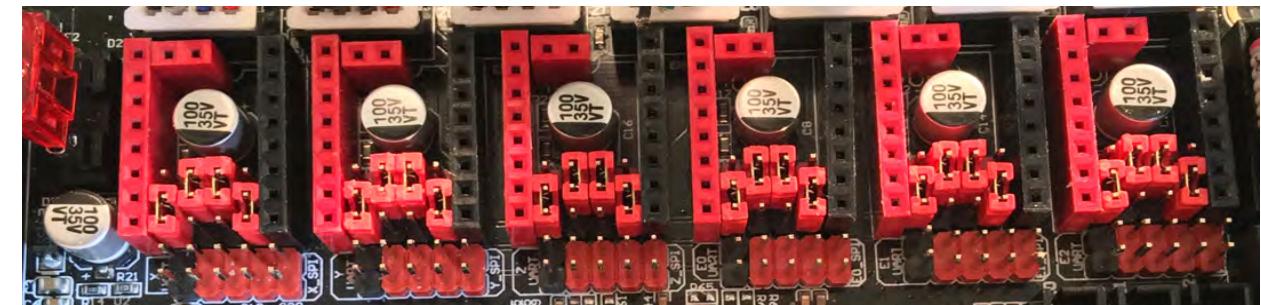
1 / 32

Note: The "D" jumper MUST be SET!



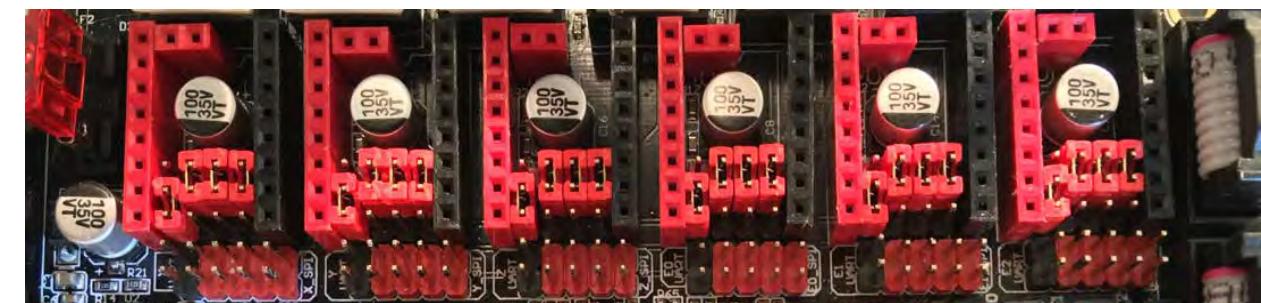
1 / 128

Note: The "D" jumper MUST be SET!



1 / 256

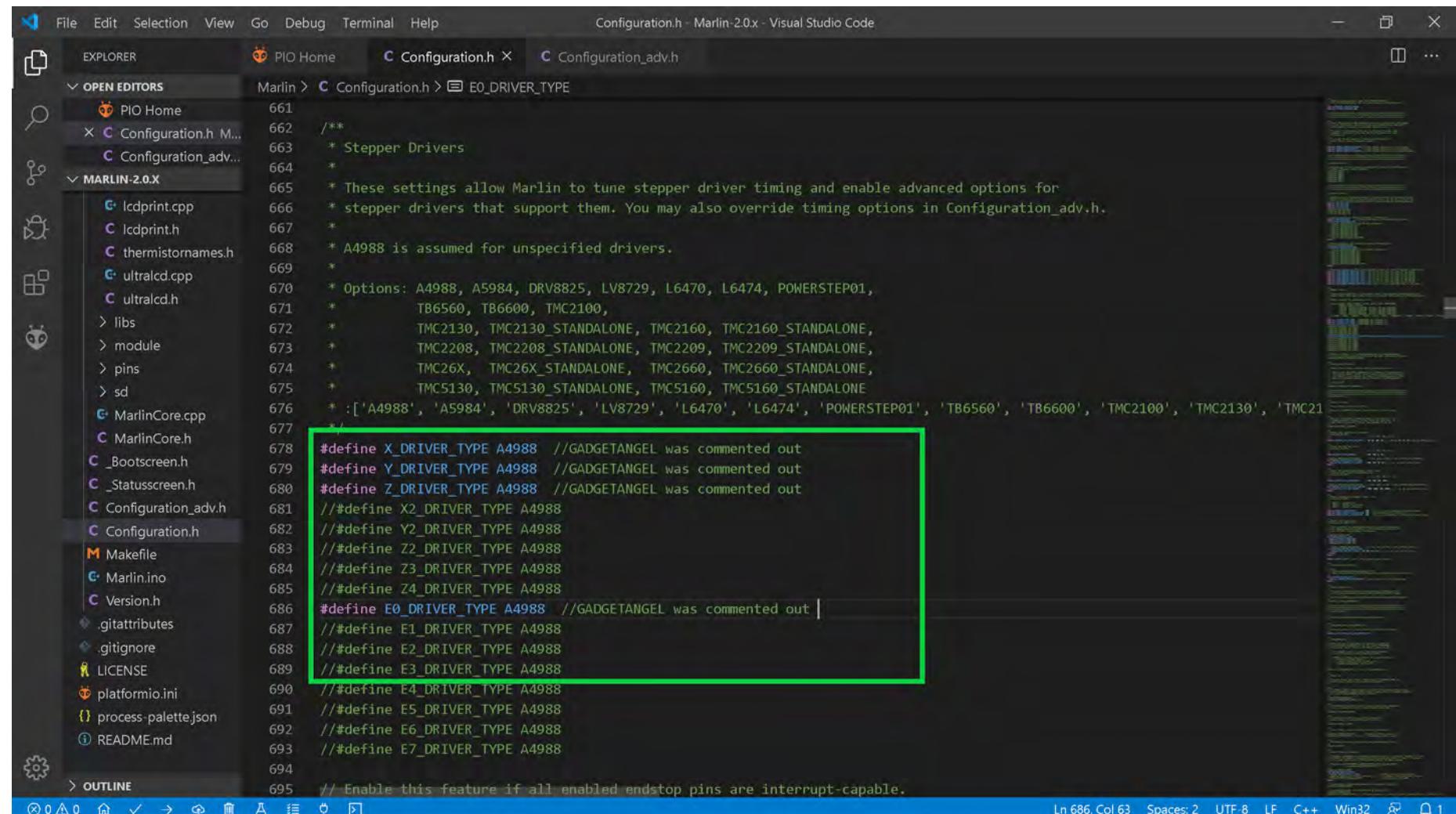
Note: The "D" jumper MUST be SET!



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

NOTE: Go to Appendix C, and 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.



```

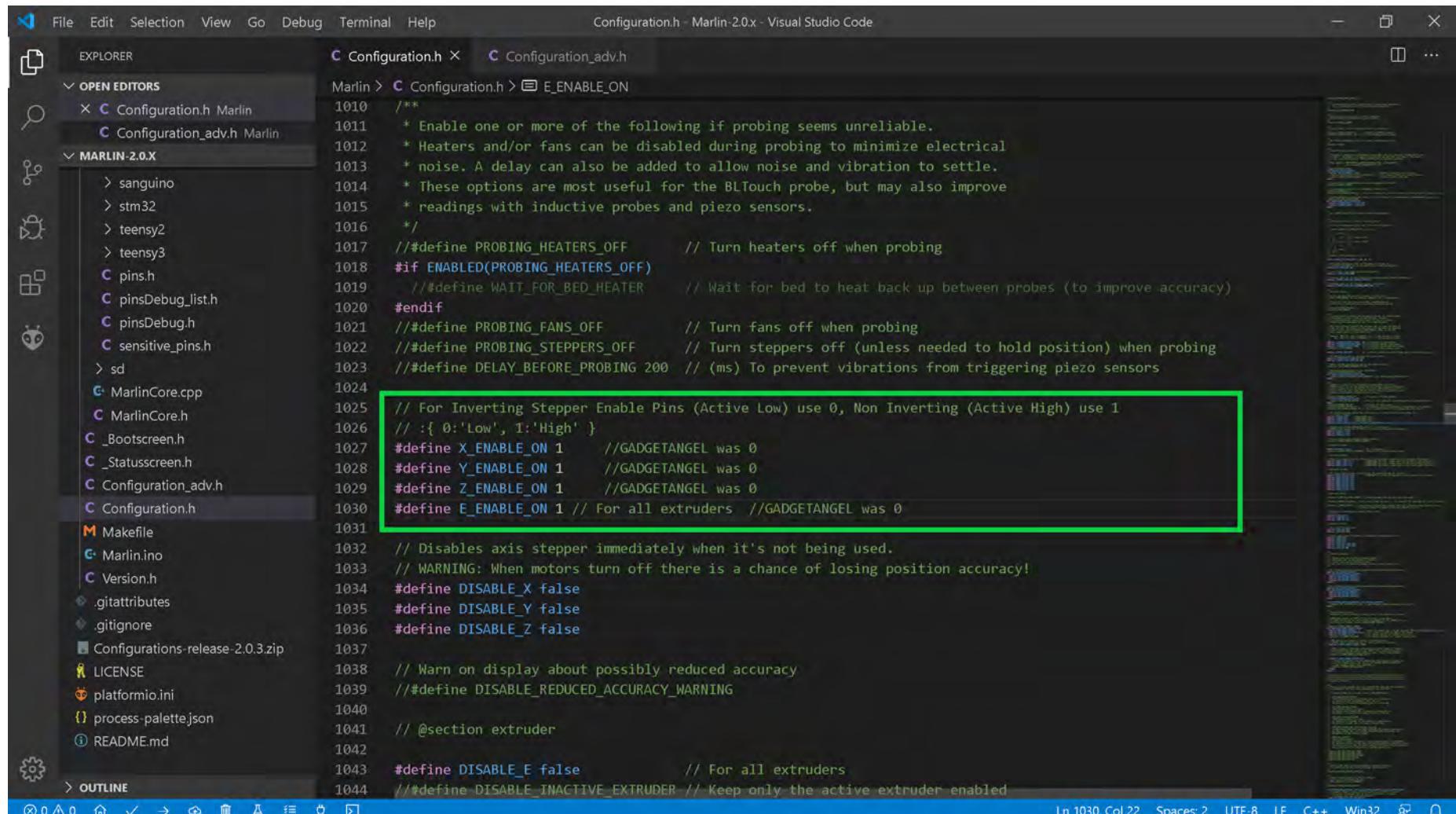
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
Marlin-2.0.X
Icdprint.cpp 661 /**
C Icdprint.h 662 */
C thermistornames.h 663 * Stepper Drivers
C ultralcd.cpp 664 *
C ultralcd.h 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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
677 */
#define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
#define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
#define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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 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.



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

```

File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
Marlin > Configuration.h > E_ENABLE_ON
1010 /**
1011 * Enable one or more of the following if probing seems unreliable.
1012 * Heaters and/or fans can be disabled during probing to minimize electrical
1013 * noise. A delay can also be added to allow noise and vibration to settle.
1014 * These options are most useful for the BLTouch probe, but may also improve
1015 * readings with inductive probes and piezo sensors.
1016 */
1017 //#define PROBING_HEATERS_OFF // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019   // #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 //#define PROBING_FANS_OFF // Turn fans off when probing
1022 //#define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1023 //#define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // :{ 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1 //GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1 //GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1 //GADGETANGEL was 0
1030 #define E_ENABLE_ON 1 // For all extruders //GADGETANGEL was 0
1031
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

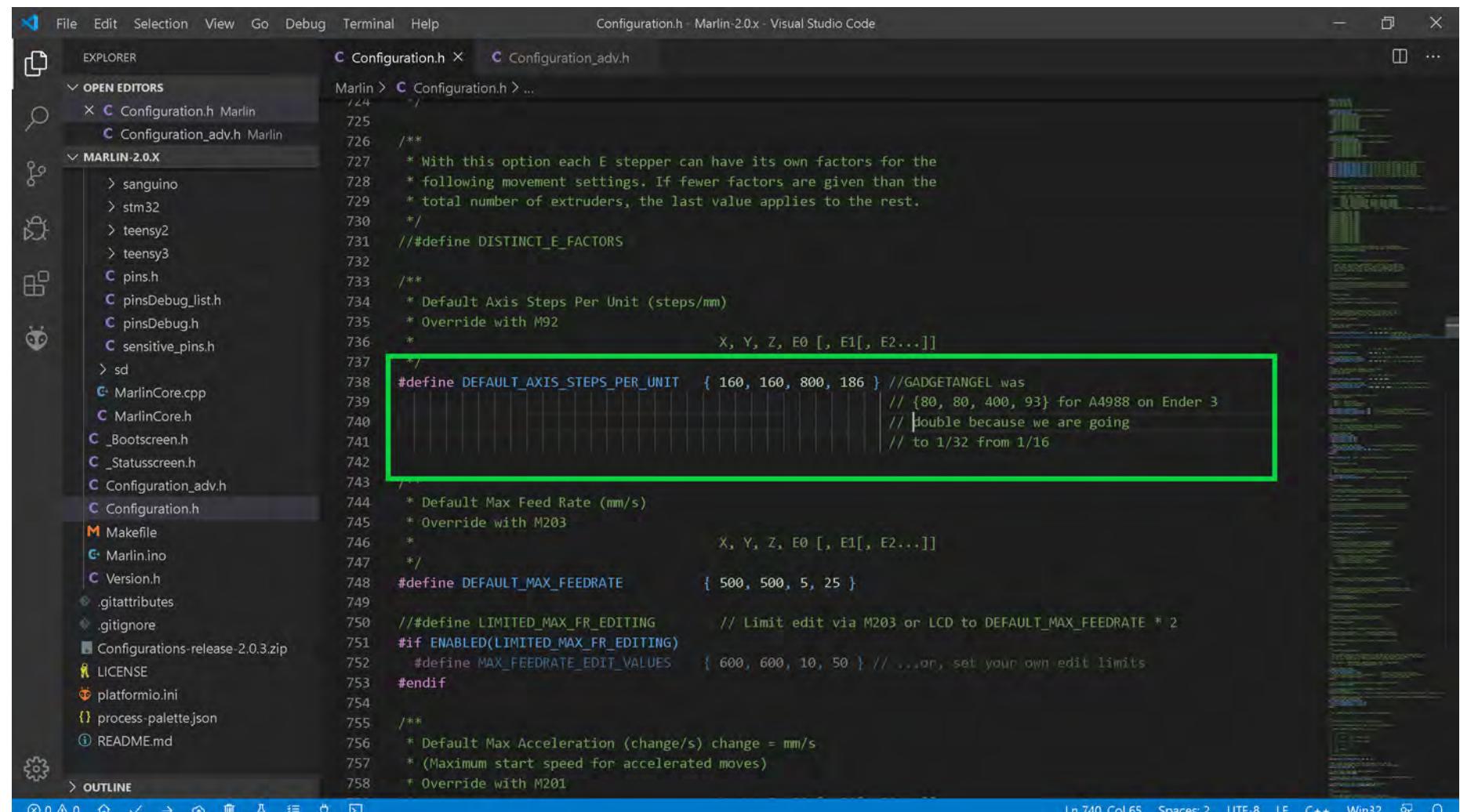
```

Ln 1030, Col 22 Spaces: 2 UTF-8 LF C++ Win32

- 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.



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 line:

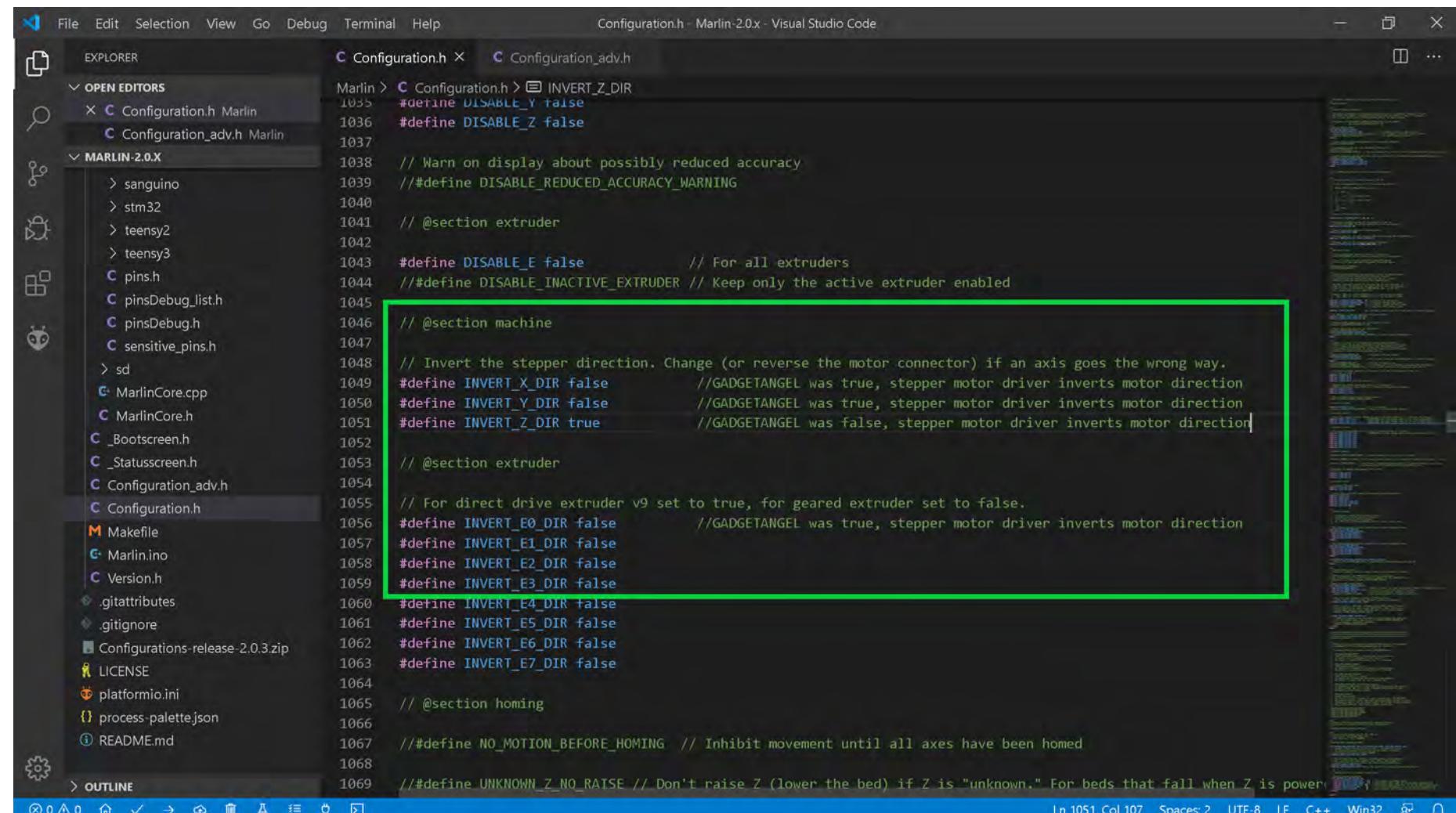
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

The 'EXPLORER' sidebar on the left shows the project structure, including files like 'MarlinCore.cpp', 'MarlinCore.h', and 'Configuration.h'. The status bar at the bottom indicates the current line (Ln 740), column (Col 65), and other settings.

- 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



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

```

File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 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
  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        //GADGETANGEL was true, stepper motor driver inverts motor direction
  1048 #define INVERT_Y_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
  1049 #define INVERT_Z_DIR true         //GADGETANGEL was false, stepper motor driver inverts motor direction
  1050
  // @section extruder
  1051
  1052 // For direct drive extruder v9 set to true, for geared extruder set to false.
  1053
  1054 //##define INVERT_E0_DIR false    //GADGETANGEL was true, stepper motor driver inverts 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
  // @section homing
  1063
  1064 //##define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1065
  1066 //##define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  1067
  1068
  1069

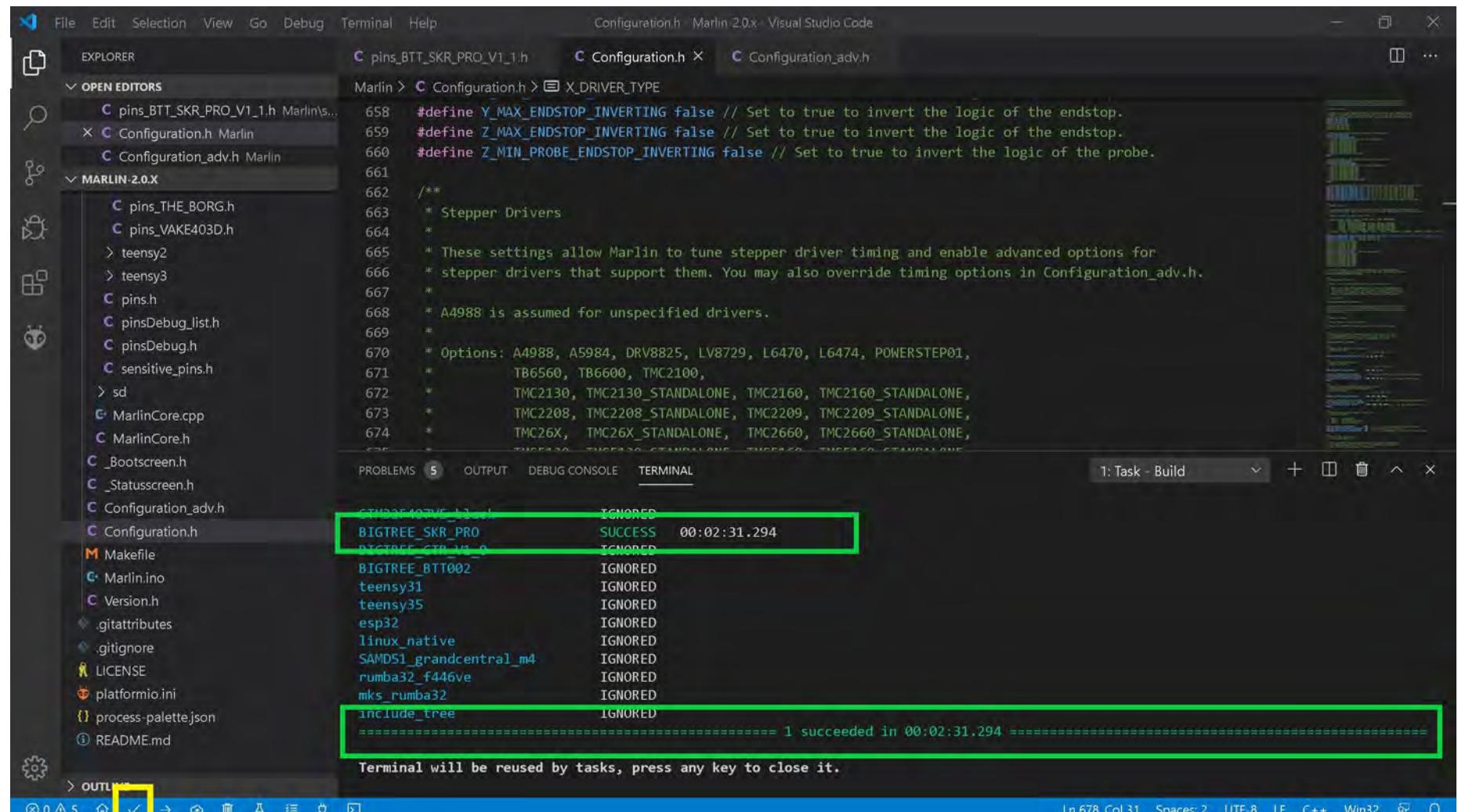
```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

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

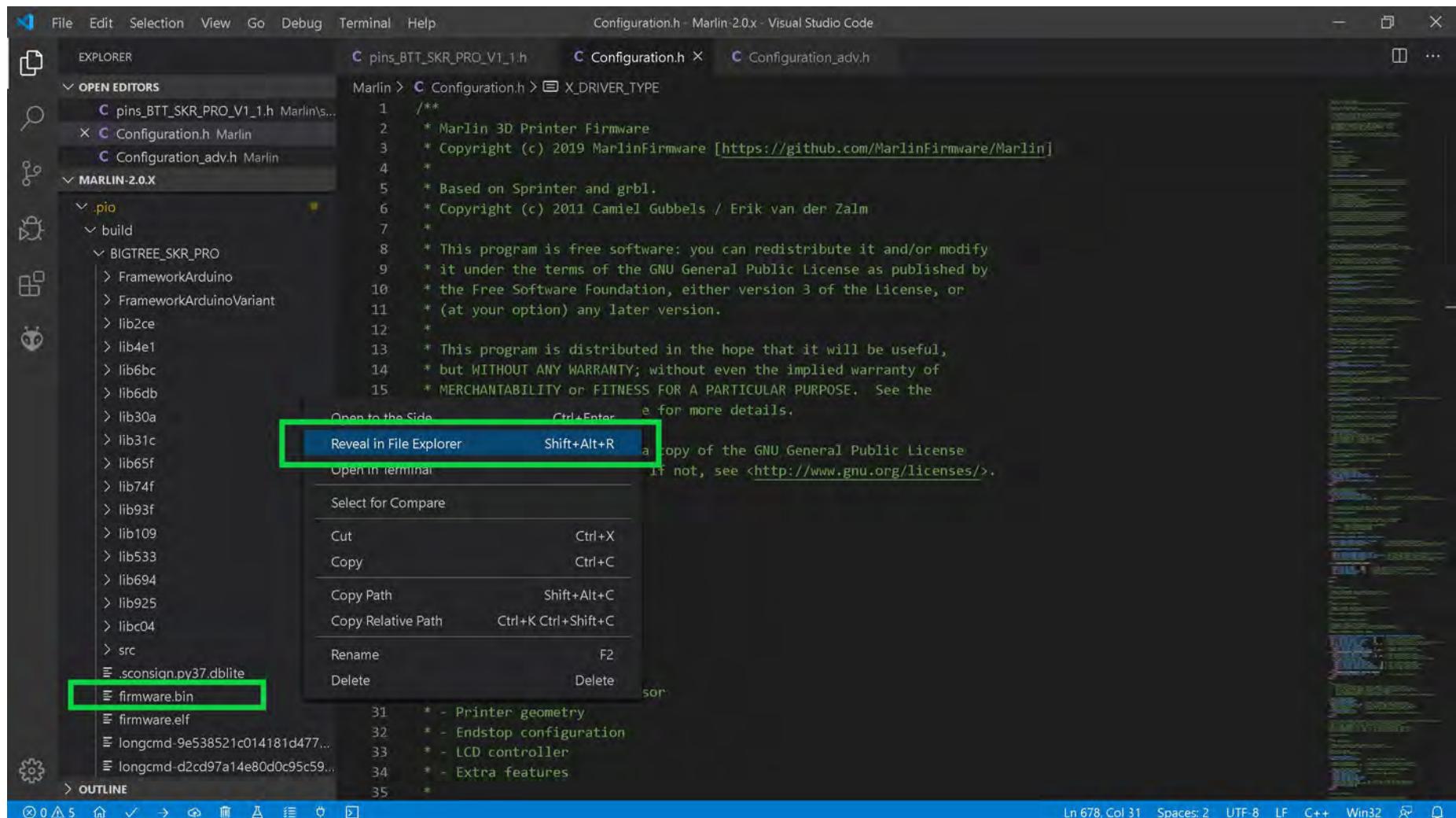
- The end of Marlin setup for POLOLU ST820 (STSPIN820) drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- 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

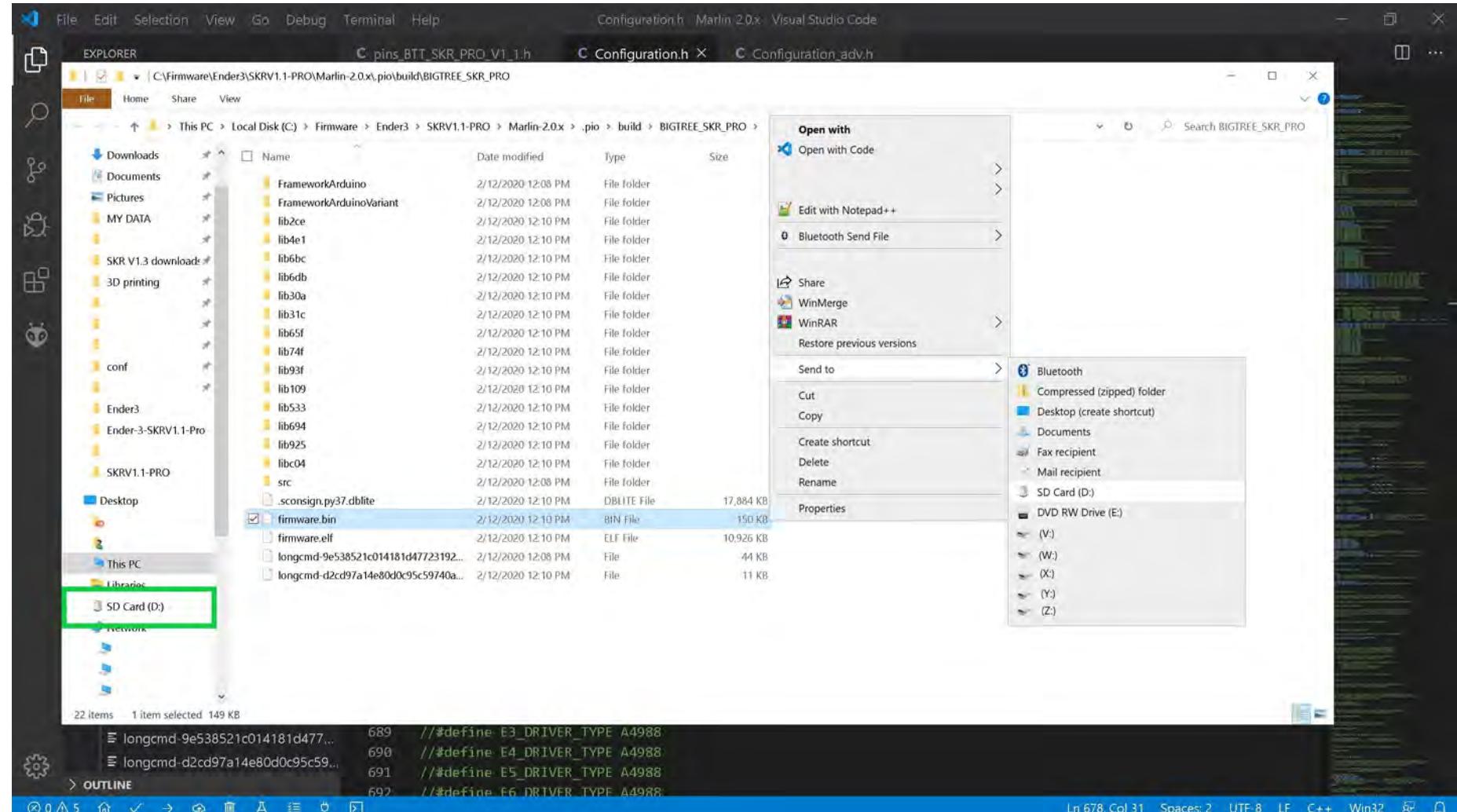
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



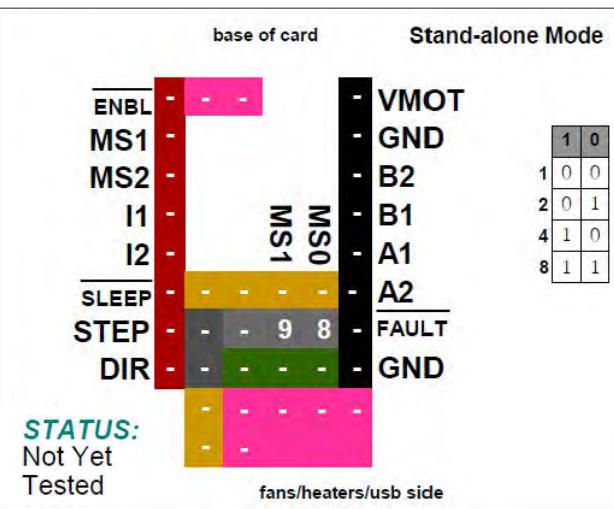
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".

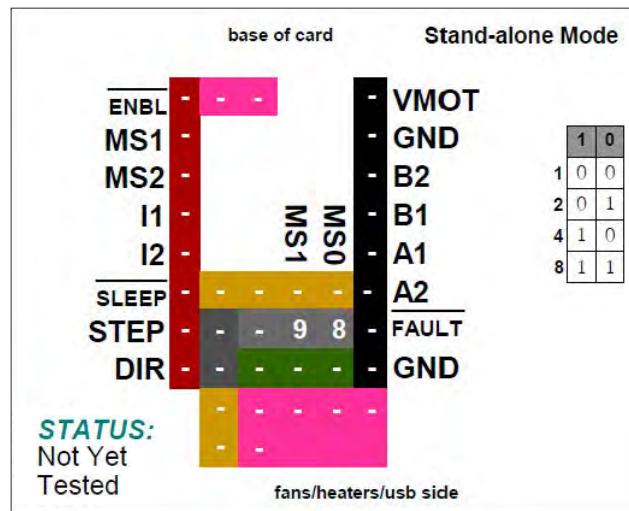


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

POLOLU MP6500

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu MP6500 Maximum 8 Subdivision 35V DC 2.5A (peak)	Low	Low	Full step	2 Phase
	Low	High	Half (1/2) step	1-2 Phase
	High	Low	Quarter (1/4) step	W1-2 Phase
	High	High	Eighth (1/8) step	2W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} = V_{ref} * 3.5$		$V_{ref} = \frac{I_{MAX}}{3.5}$	

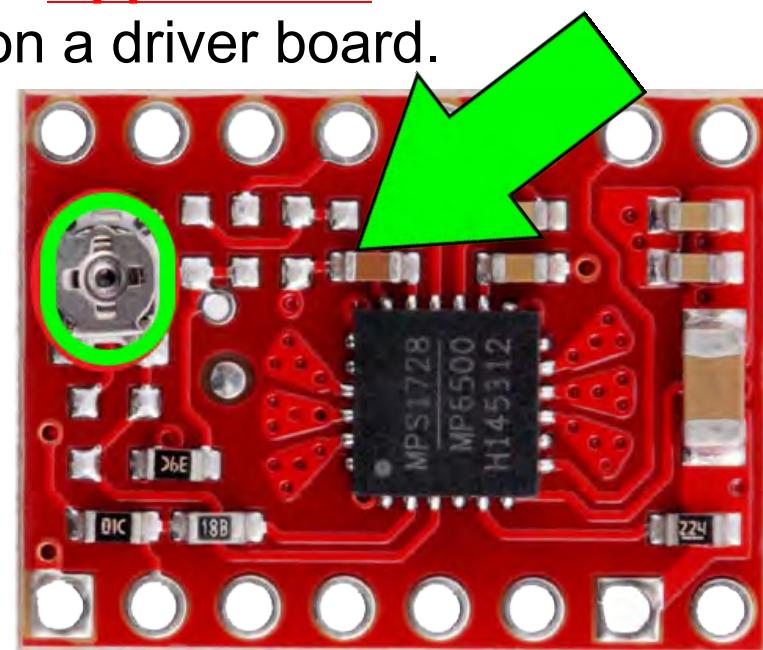
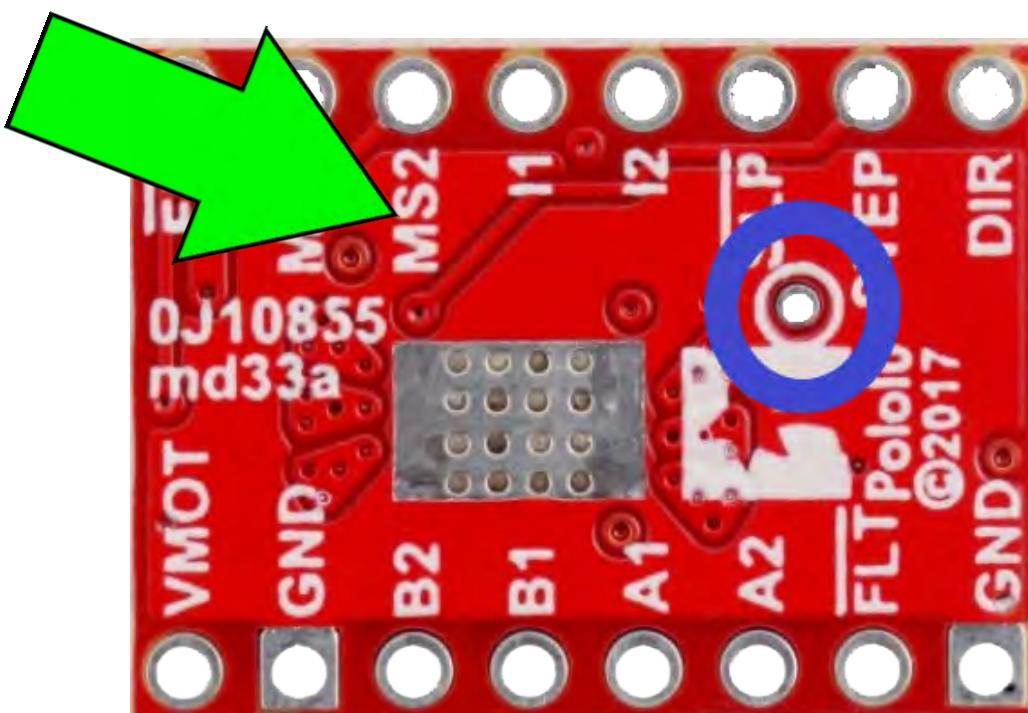


POLOLU MP6500

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

Note: " V_{ref} Test point" location is on the bottom of the driver board, as shown in BLUE



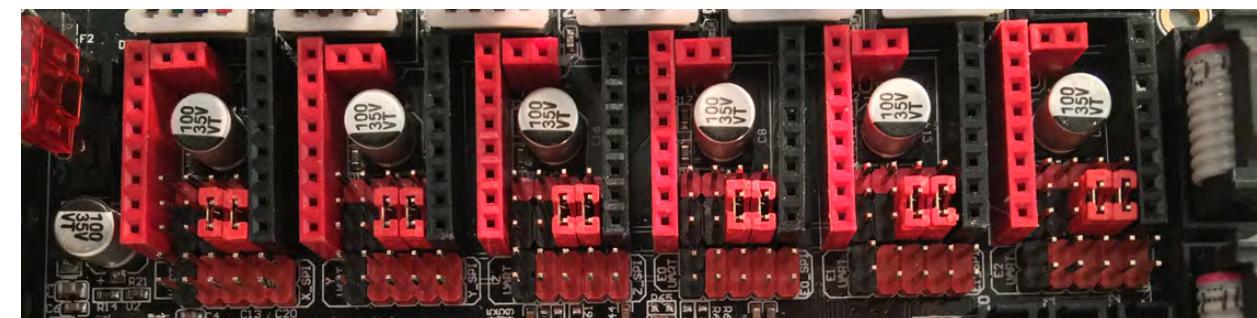
Note: MP6500 driver board does not use external current sense resistors (R_s).

POLOLU MP6500

Stand-alone Mode

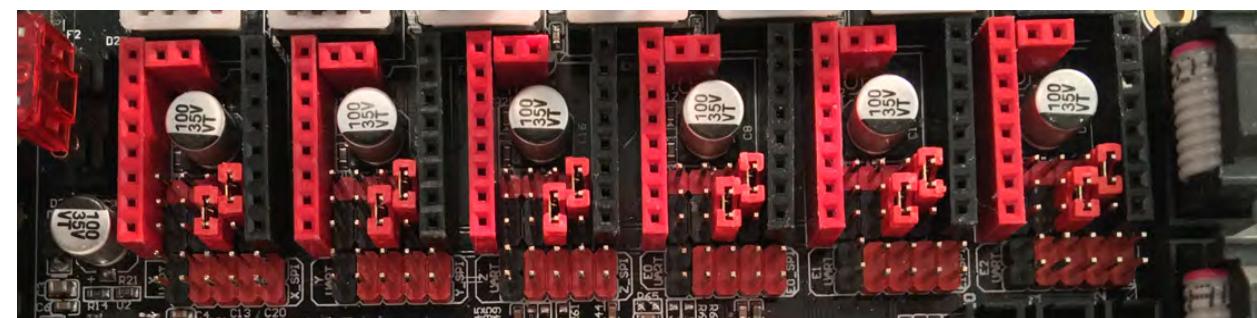
STEP

ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	-	-	B1
I2	-	-	-	A1
SLEEP	-	-	-	A2
STEP	-	-	9 8	FAULT
DIR	-	-	9 8	GND
	-	-	-	-



1 / 2

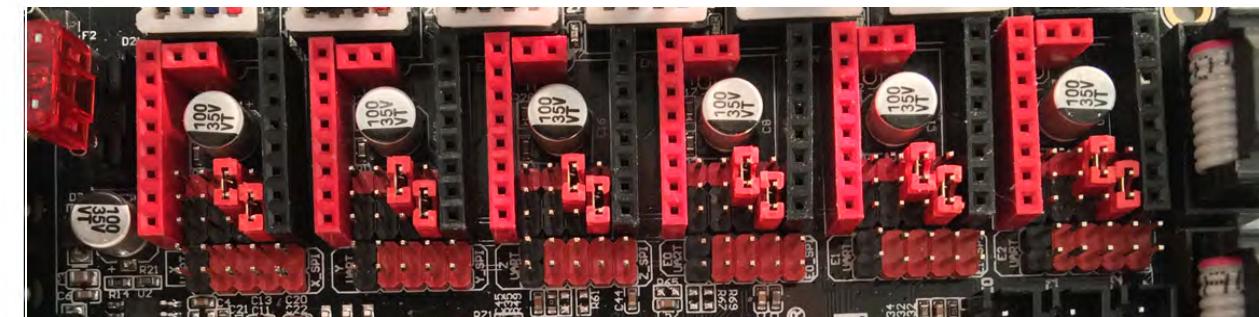
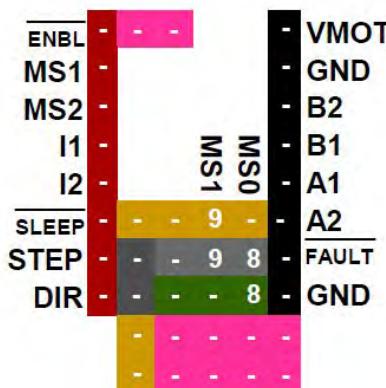
ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	-	-	B1
I2	-	-	-	A1
SLEEP	-	-	-	A2
STEP	-	-	9 8	FAULT
DIR	-	-	9 -	GND
	-	-	-	-



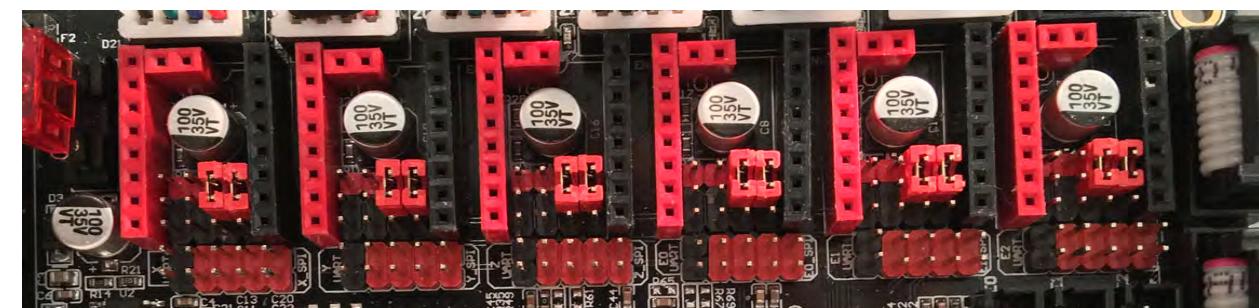
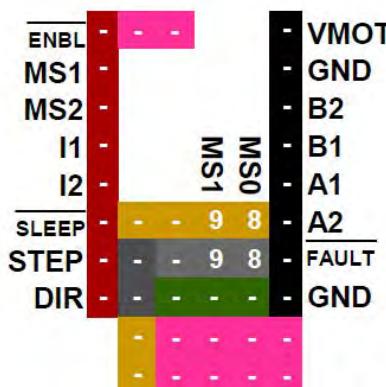
POLOLU MP6500

Stand-alone Mode

1 / 4



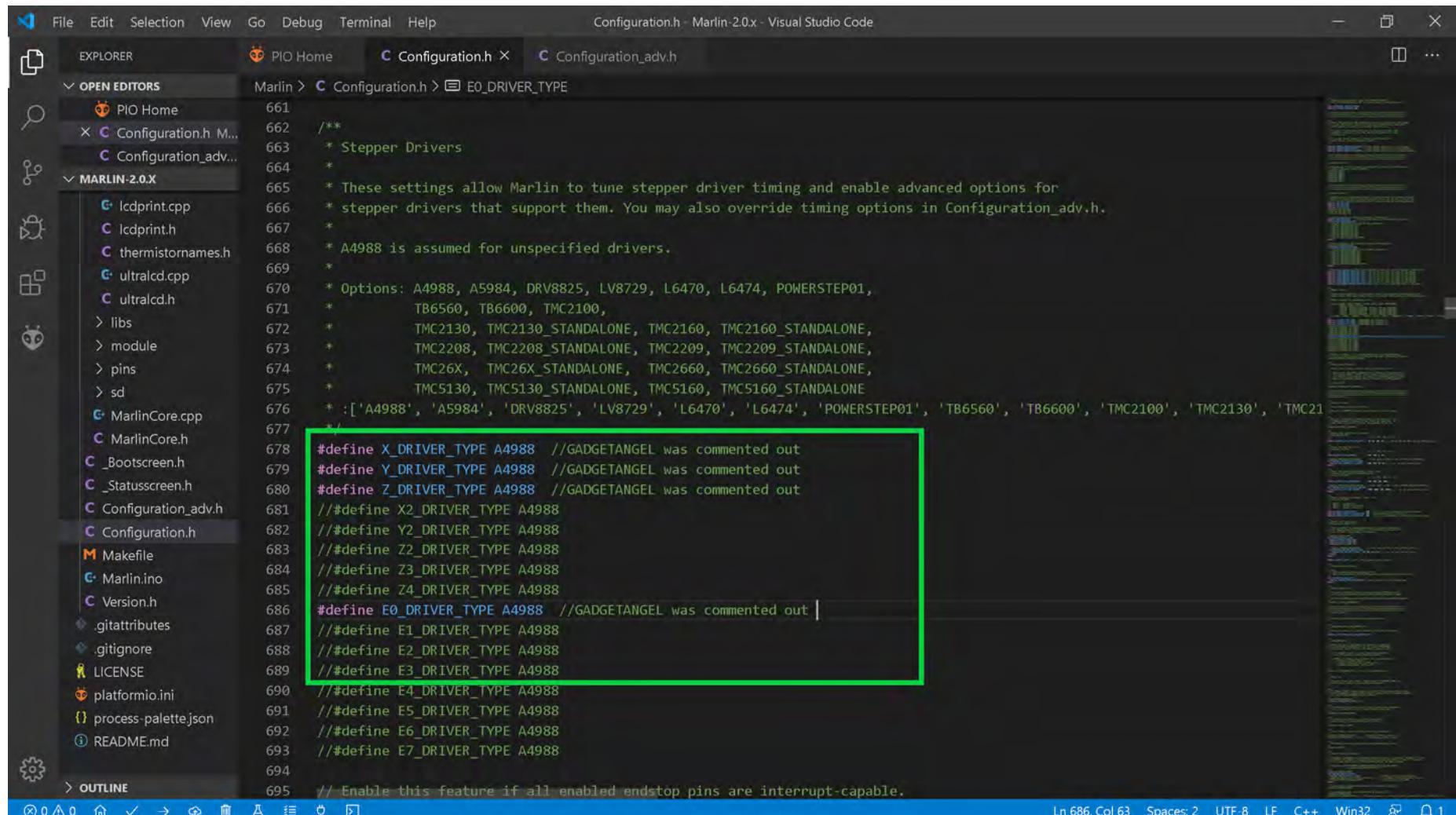
1 / 8



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

NOTE: Go to Appendix C, and 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 **POLOLU MP6500 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 following details:

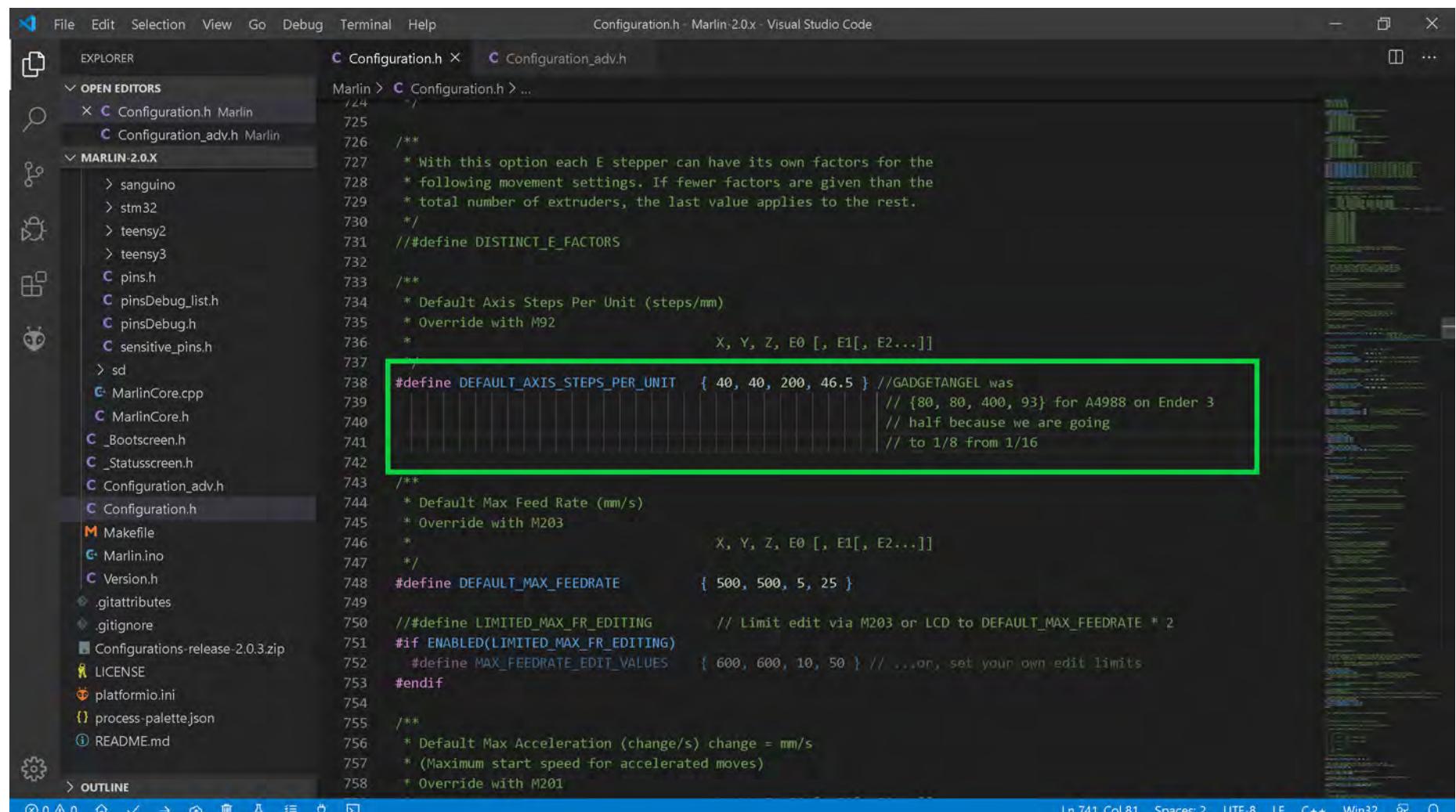
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0x - Visual Studio Code.
- Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Editor:** Displays the `Configuration.h` file content. A green box highlights the following code block:


```
#define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
#define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
#define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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
```
- Status Bar:** Ln 686, Col 63, Spaces: 2, UTF-8, LF, C++, Win32, 1.

- 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.



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 line:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 40, 40, 200, 46.5 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// half because we are going
// to 1/8 from 1/16
```

The left sidebar shows the project structure under 'MARLIN-2.0.X' and the right sidebar shows the 'PROBLEMS' and 'WORLD' panes.

- Go to the next page.

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

- **Optional Step:** I cannot find information on the POLOLU MP6500 driver's impact on motor direction. 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 MP6500 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 Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- 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.

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

EXPLORER C pins_BTT_SKR_PRO_V1_1.h C Configuration.h X Configuration_adv.h

OPEN EDITORS Marlin > C Configuration.h > X DRIVER_TYPE

MARLIN-2.0.X

C pins_BTT_SKR_PRO_V1_1.h Marlin's... C Configuration.h Marlin C Configuration_adv.h Marlin

C pins_THE_BORG.h
C 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-palette.json
README.md

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

BIGTREETECH_BTT002 SUCCESS 00:02:31.294

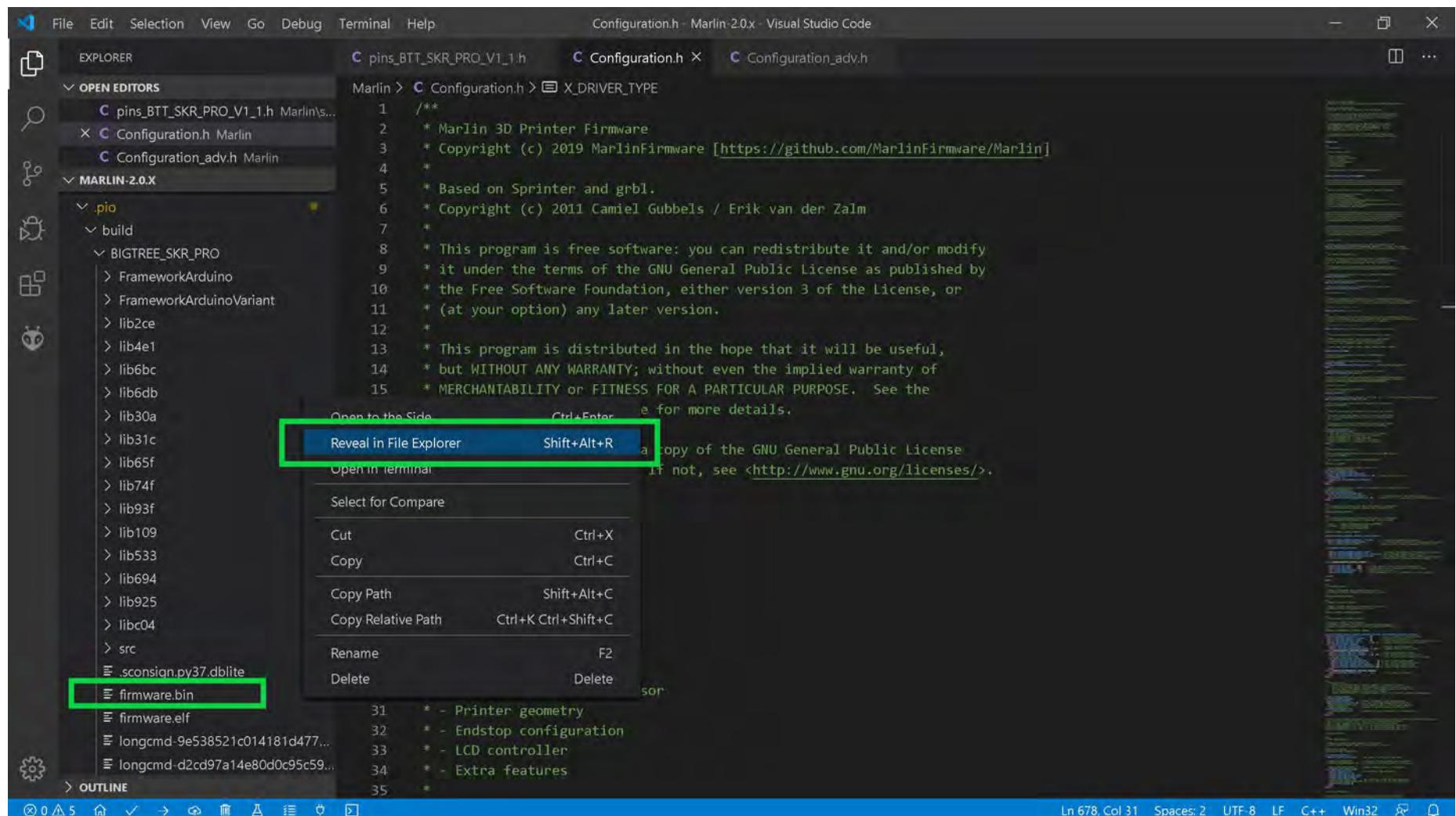
=====
1 succeeded in 00:02:31.294 =====

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 POLOLU MP6500 Drivers

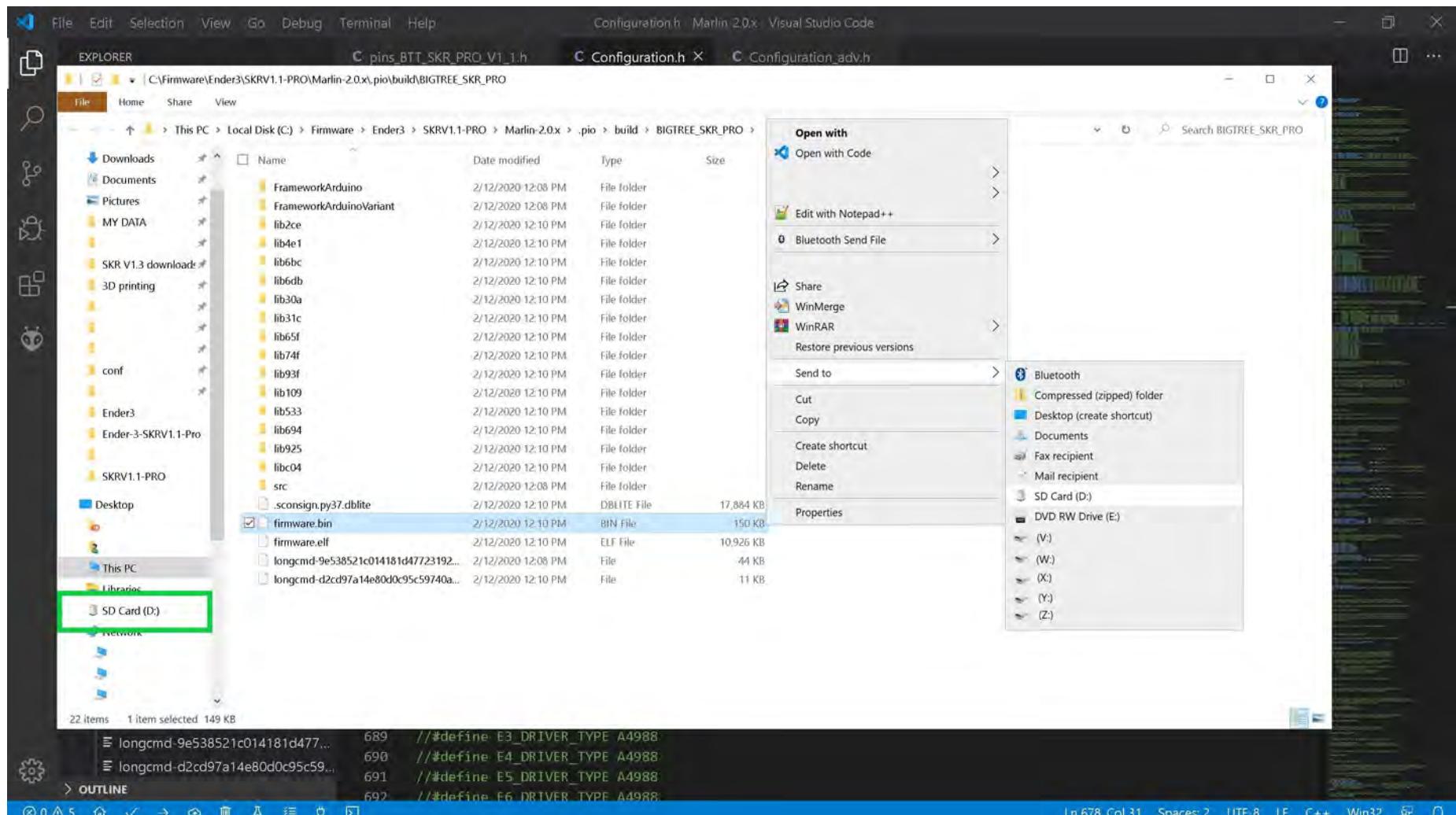
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



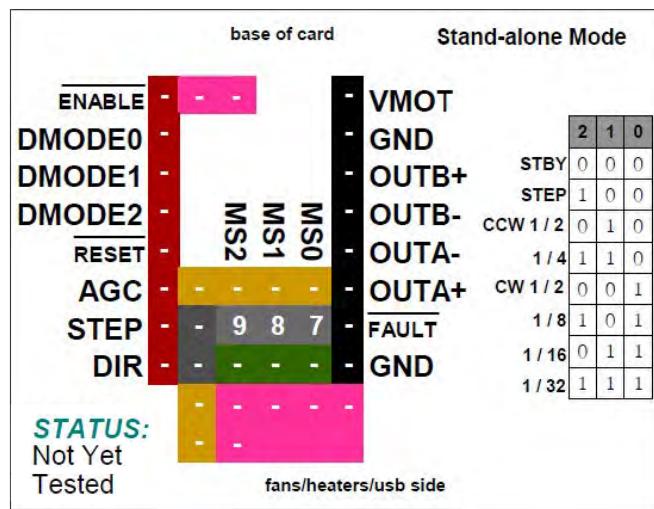
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



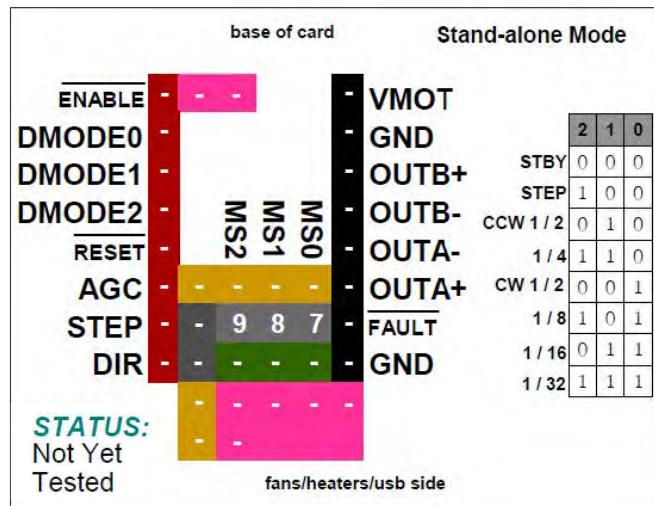
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



POLOLU TB67S249FTG

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution
Pololu TB67S249FTG	Low	Low	Low	Standby mode (outputs disabled)
Maximum 32 Subdivision	High	Low	Low	Full step
47V DC 4.5A (peak)	Low	High	Low	Non-circular half step ("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	$I_{MAX} = V_{ref} * 1.25$			$V_{ref} = \frac{I_{MAX}}{1.25}$

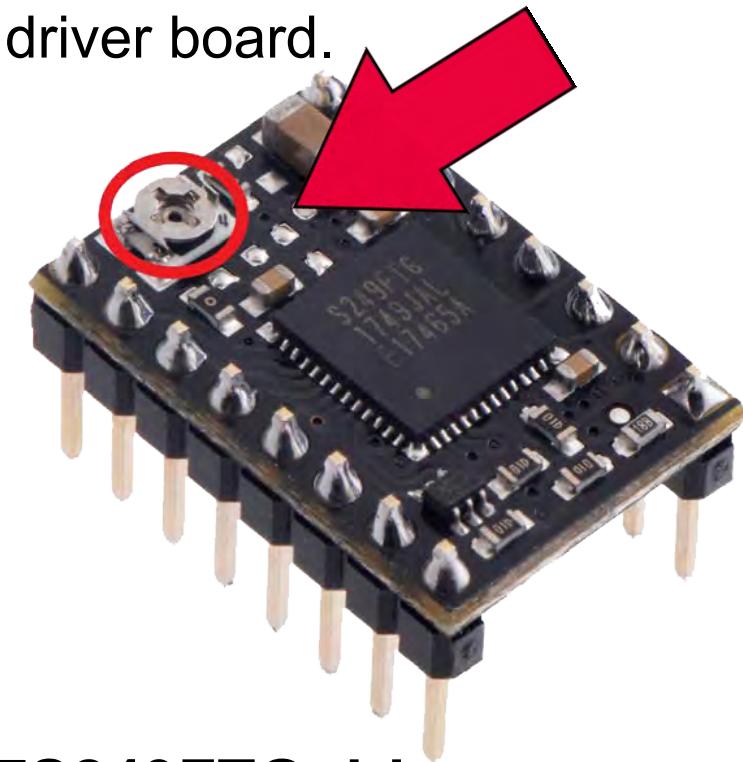
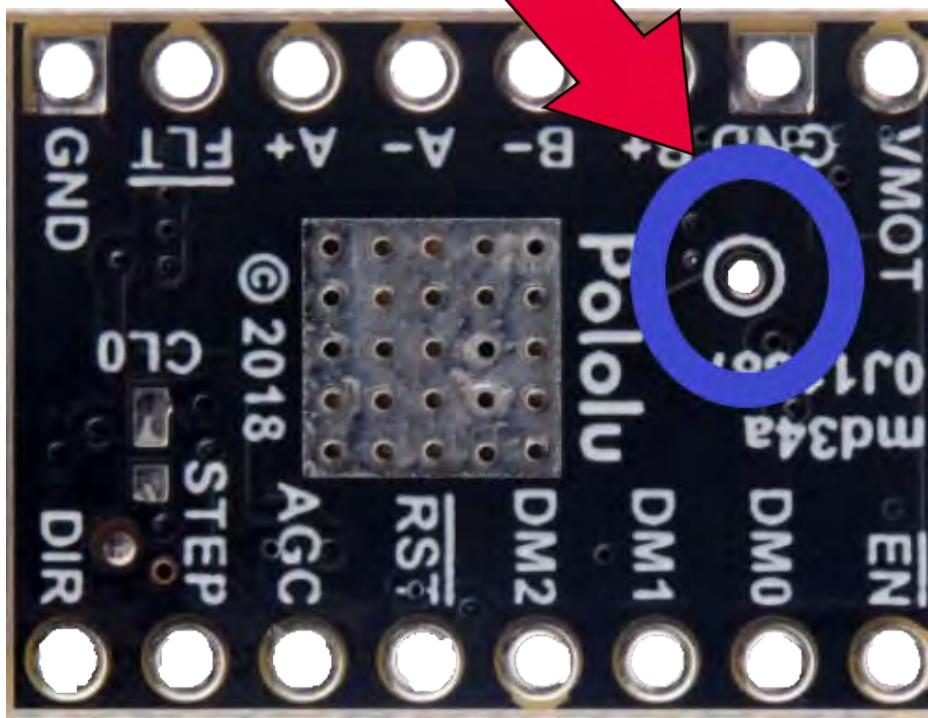


POLOLU TB67S249FTG

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

Note: "["V_{ref} Test point"](#)" location is on the Bottom of the driver board, as shown in **BLUE**



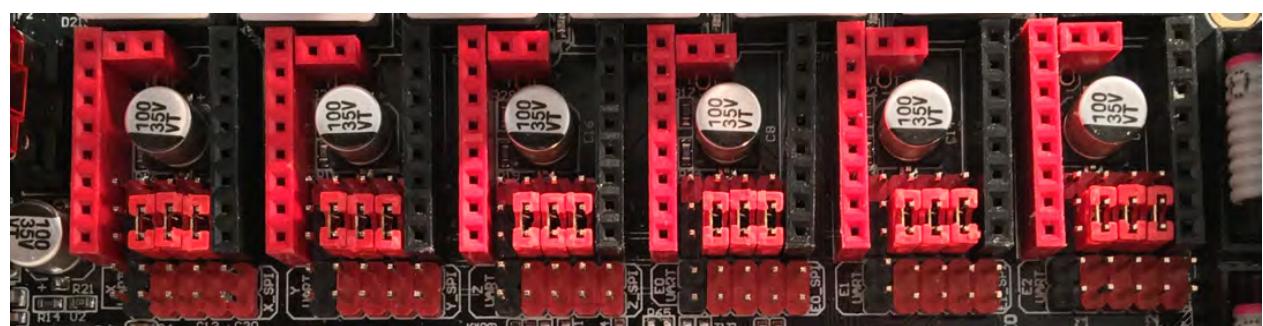
Note: TB67S249FTG driver board does not use external current sense resistors (R_s).

POLOLU TB67S249FTG

Stand-alone Mode

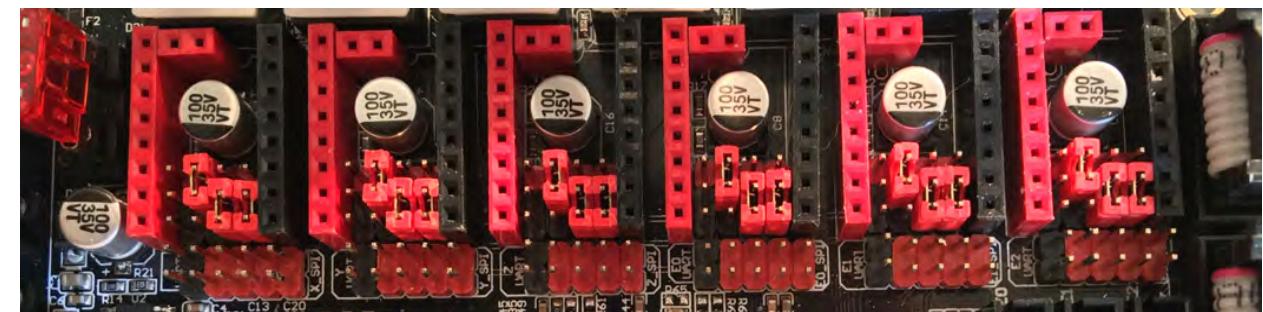
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

ENABLE				VMOT
DMODE0				GND
DMODE1				OUTB+
DMODE2		MS2	MS1	OUTB-
RESET		MS0		OUTA-
AGC				OUTA+
STEP	9	8	7	FAULT
DIR	9	8	7	GND



Stand By

ENABLE				VMOT
DMODE0				GND
DMODE1				OUTB+
DMODE2		MS2	MS1	OUTB-
RESET		MS0		OUTA-
AGC	9			OUTA+
STEP	9	8	7	FAULT
DIR		8	7	GND



STEP

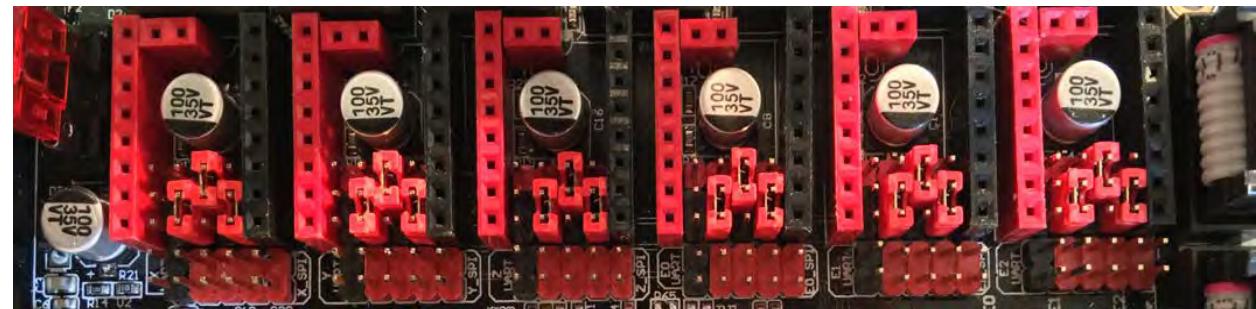
POLOLU TB67S249FTG

Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

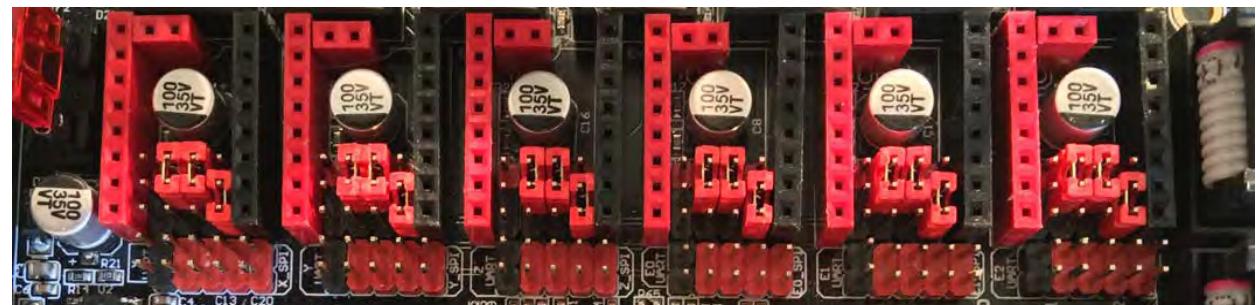
ENABLE	-	-	-	VMOT	
D MODE0	-	-	-	GND	
D MODE1	-	-	-	OUTB+	
D MODE2	-	MS2	MS1	MS0	OUTB-
RESET	-	-	-	-	OUTA-
AGC	-	-	8	-	OUTA+
STEP	-	9	8	7	FAULT
DIR	-	9	-	7	GND

CCW 1 / 2



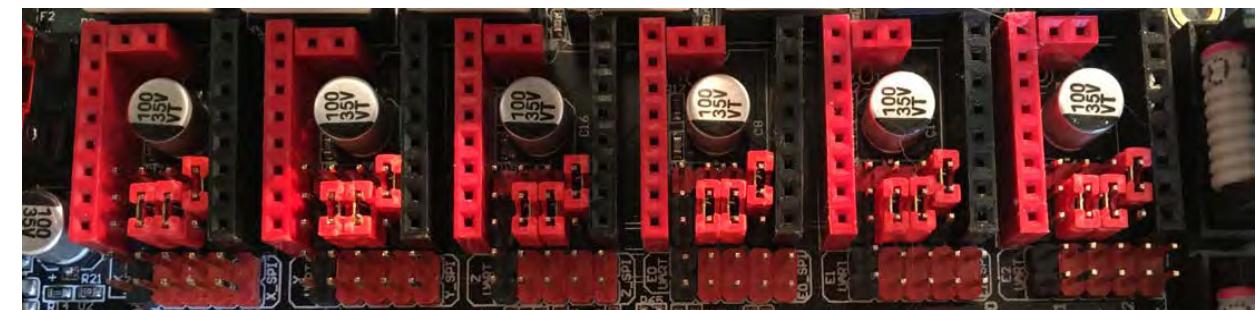
ENABLE	-	-	-	VMOT	
D MODE0	-	-	-	GND	
D MODE1	-	-	-	OUTB+	
D MODE2	-	MS2	MS1	MS0	OUTB-
RESET	-	-	-	-	OUTA-
AGC	-	-	9	8	OUTA+
STEP	-	9	8	7	FAULT
DIR	-	-	-	7	GND

1 / 4



ENABLE	-	-	-	VMOT	
D MODE0	-	-	-	GND	
D MODE1	-	-	-	OUTB+	
D MODE2	-	MS2	MS1	MS0	OUTB-
RESET	-	-	-	-	OUTA-
AGC	-	-	-	7	OUTA+
STEP	-	9	8	7	FAULT
DIR	-	9	8	-	GND

CW 1 / 2



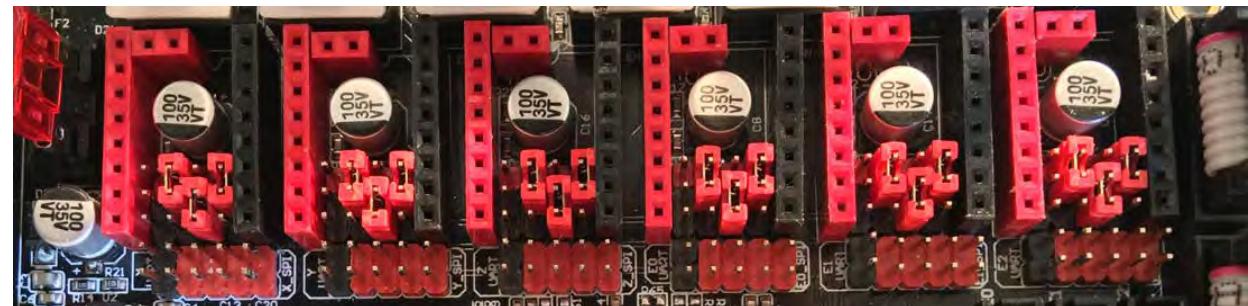
POLOLU TB67S249FTG

Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

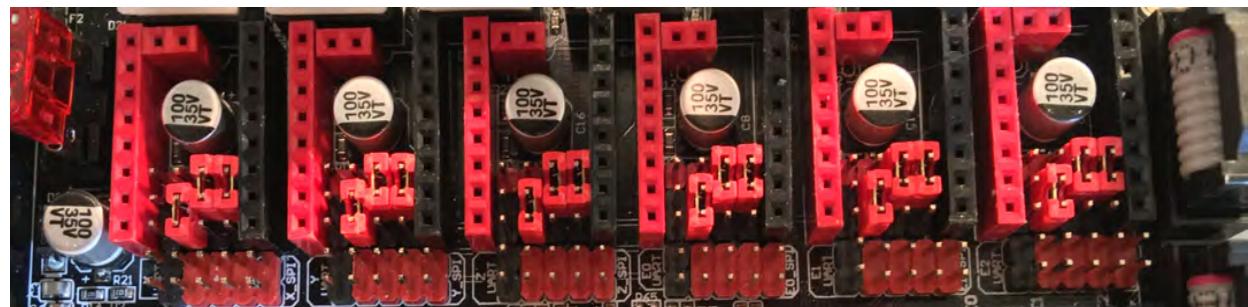
ENABLE	-	-	-	VMOT
DMODE0	-	-	-	GND
DMODE1	-	-	-	OUTB+
DMODE2	-	MS2 MS1 MS0	-	OUTB-
RESET	-	-	-	OUTA-
AGC	-	9 - 7	-	OUTA+
STEP	-	9 8 7	-	FAULT
DIR	-	-	8	-
				GND

1 / 8



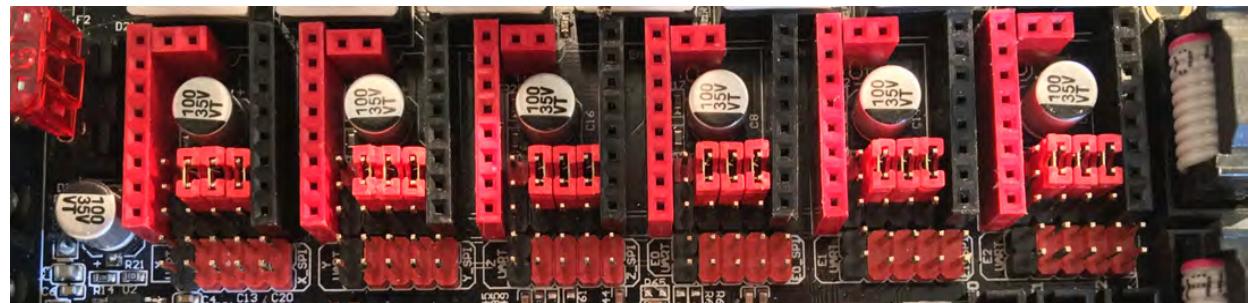
ENABLE	-	-	-	VMOT
DMODE0	-	-	-	GND
DMODE1	-	-	-	OUTB+
DMODE2	-	MS2 MS1 MS0	-	OUTB-
RESET	-	-	-	OUTA-
AGC	-	-	8 7	OUTA+
STEP	-	9 8 7	-	FAULT
DIR	-	9	-	-
				GND

1 / 16



ENABLE	-	-	-	VMOT
DMODE0	-	-	-	GND
DMODE1	-	-	-	OUTB+
DMODE2	-	MS2 MS1 MS0	-	OUTB-
RESET	-	-	-	OUTA-
AGC	-	-	9 8 7	OUTA+
STEP	-	9 8 7	-	FAULT
DIR	-	-	-	GND

1 / 32



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

NOTE: Go to Appendix C, and 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.

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

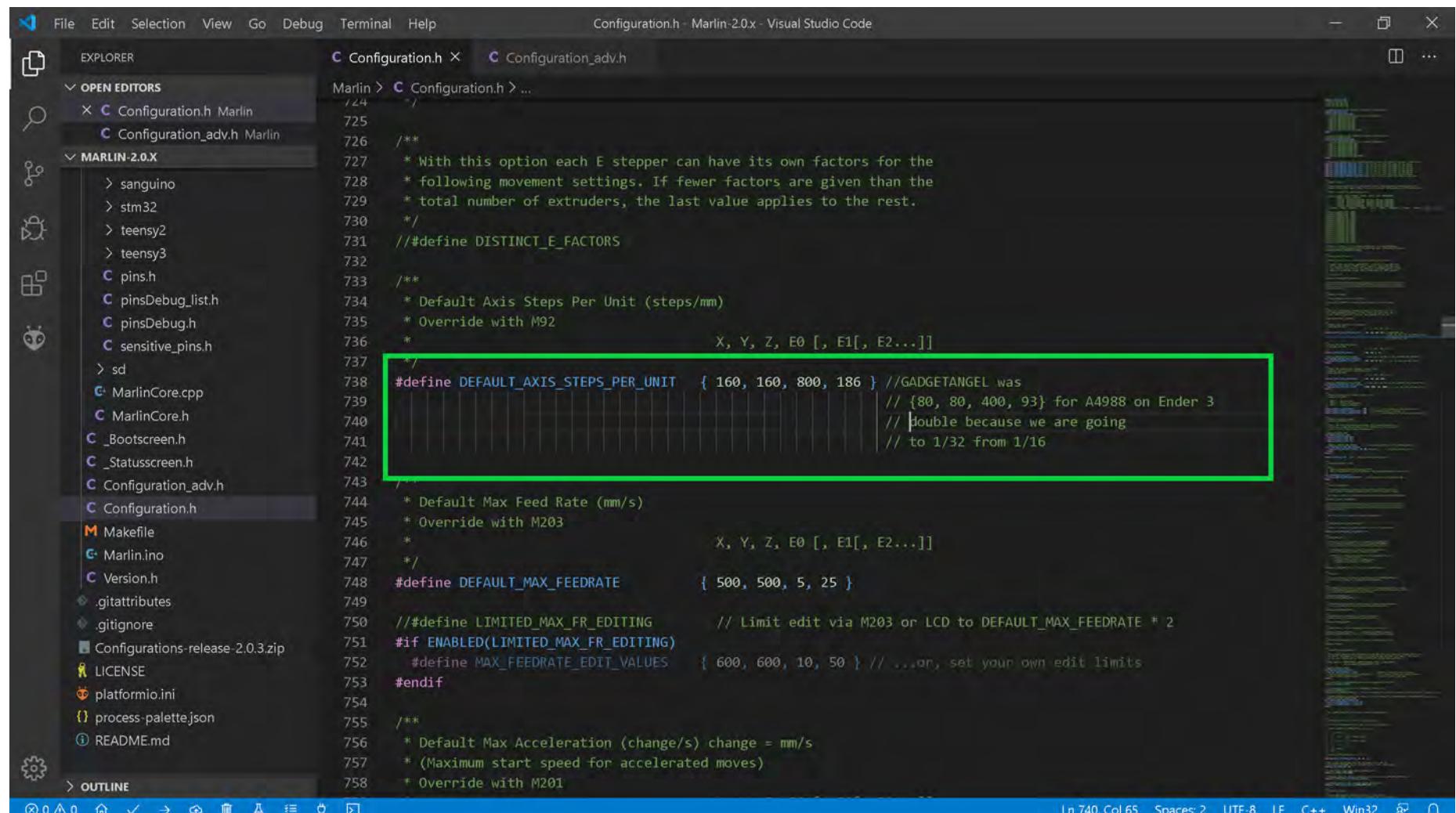
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Sidebar:** Explorer, Search, Find, Open Editors, MARLIN-2.0.X, and a list of files including MarlinCore.cpp, MarlinCore.h, _Bootscreen.h, _Statusscreen.h, Configuration_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, and README.md.
- Editor Area:** The main code editor displays Configuration.h with the following content:

```
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', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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.
```
- Right Panel:** Shows the Command Palette, Taskbar, and a preview of the current file.

- 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 OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin MARLIN-2.0.X sanguino stm32 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 Configurations-release-2.0.3.zip LICENSE platformio.ini process-palette.json README.md OUTLINE

```

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 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // double because we are going
741 // to 1/32 from 1/16
742 /**
743 * Default Max Feed Rate (mm/s)
744 * Override with M203
745 *
746 * X, Y, Z, E0 [, E1[, E2...]]
747 */
748 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
749 /**
750 * #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
751 #if ENABLED(LIMITED_MAX_FR_EDITING)
752 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...on, set your own edit limits
753 #endif
754 /**
755 * Default Max Acceleration (change/s) change = mm/s
756 * (Maximum start speed for accelerated moves)
757 * Override with M201

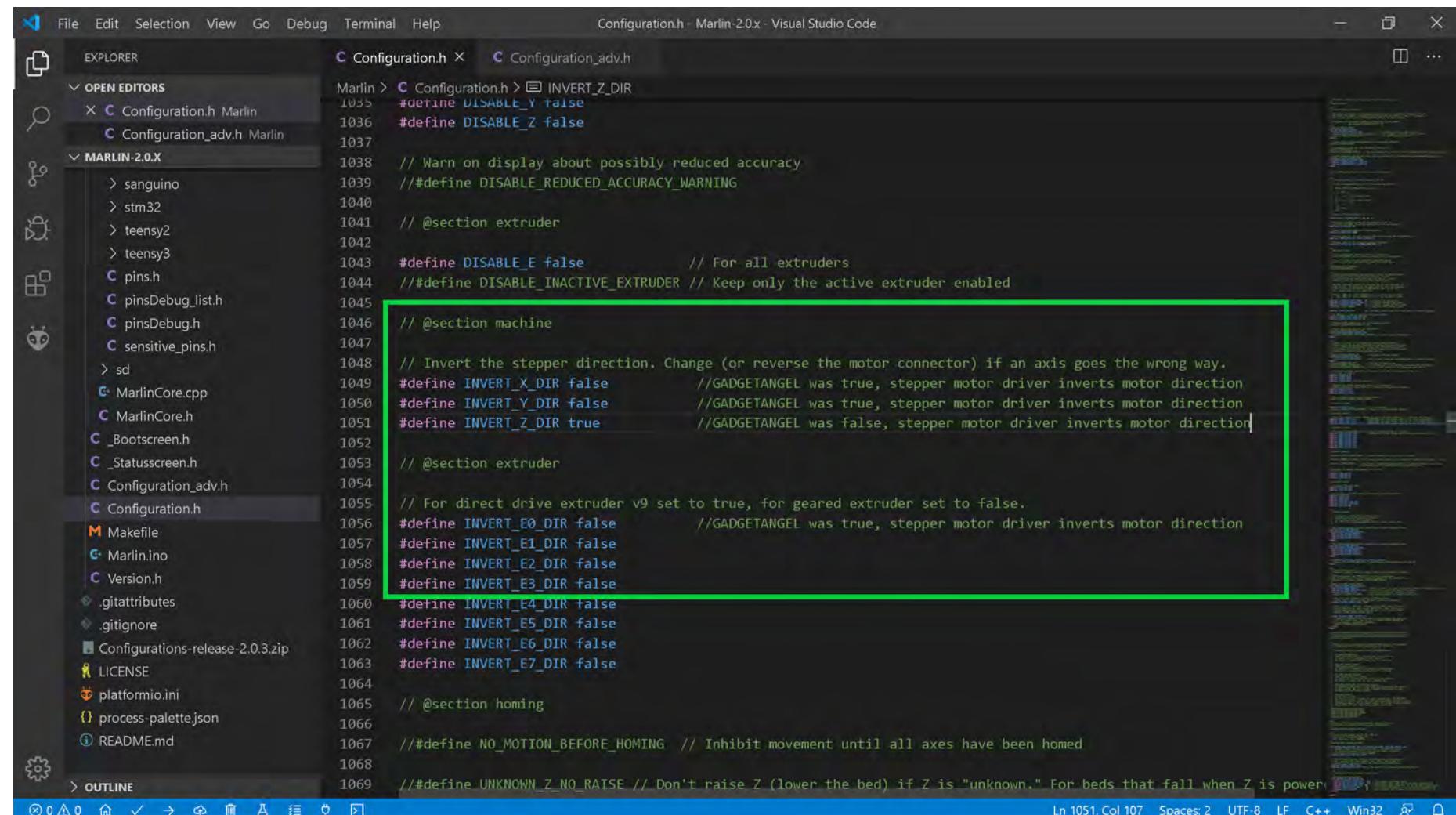
```

Ln 740, Col 65 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

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

- **Optional Step:** I cannot find information on the POLOLU TB67S249FTG driver's impact on motor direction. 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 TB67S249FTG 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 Marlin 2.0.x configuration header. A green rectangular box highlights a specific section of the code:

```

Marlin > Configuration.h > INVERT_Z_DIR
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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

The code in the green box defines the `INVERT_X_DIR`, `INVERT_Y_DIR`, and `INVERT_Z_DIR` macros. The `INVERT_Z_DIR` macro is currently set to `true`, which corresponds to the POLOLU TB67S249FTG driver's behavior where the driver inverts the motor direction.

- 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 Marlin 2.0.x repository open. The left sidebar displays the file structure under 'MARLIN-2.0.X'. The main editor area shows code for stepper driver timing, with lines 658-662 defining endstop inversion logic. The bottom right shows a terminal window with a green border, displaying a successful build log:

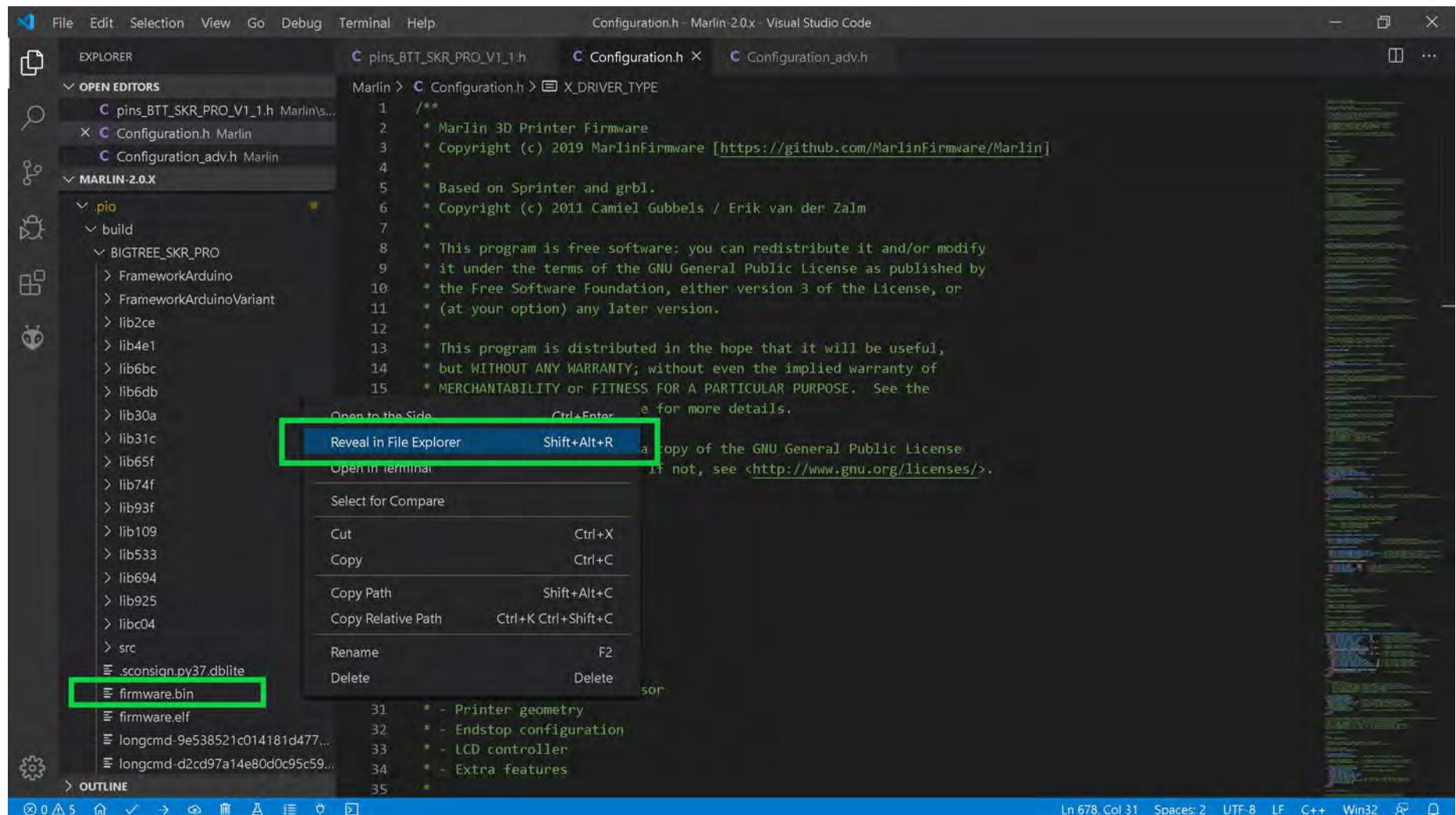
```
1: Task - Build
=====
[1/1] C:\Users\...\.platformio\build\bigtree_ctp_v1_0\BIGTREE_CTP_V1_0.ino
    BIGTREE_SKR_PRO SUCCESS 00:02:31.294
    BIGTREE_CTP_V1_0 IGNORED
    BIGTREE_BTT002 IGNORED
    teensy31 IGNORED
    teensy35 IGNORED
    esp32 IGNORED
    linux_native IGNORED
    SAMD51_grandcentral_m4 IGNORED
    rumba32_f446ve IGNORED
    mks_rumba32 IGNORED
    include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```

The terminal also contains the message: "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 POLOLU TB67S249FTG Drivers

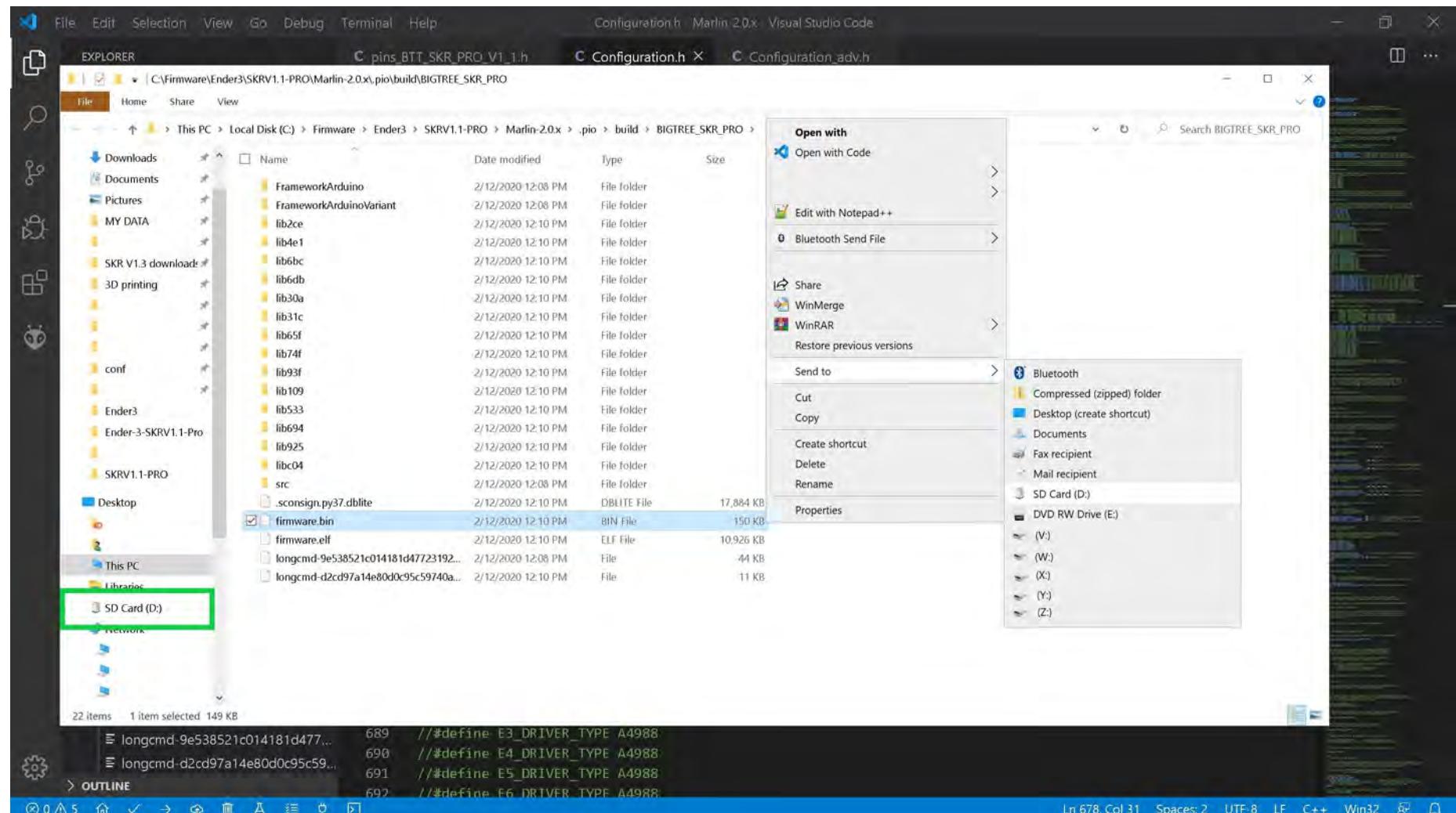
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

BIQU S109

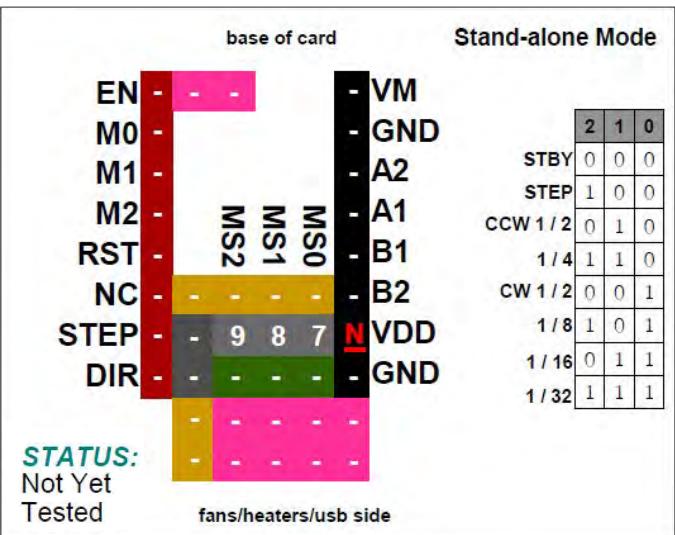
base of card		Stand-alone Mode
EN	-	VM
M0	-	GND
M1	-	A2
M2	-	A1
RST	MS2	B1
NC	MS1	B2
STEP	MS0	VDD
DIR	-	GND
STATUS:		
Not Yet Tested		
fans/heaters/usb side		

STBY	2	1	0
STEP	0	0	0
CCW 1/2	1	0	0
1/4	0	1	0
CW 1/2	1	1	0
1/8	0	0	1
1/16	1	0	1
1/32	0	1	1
	1	1	1

Note: V_{DD} is an Output (N), *on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!!*

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution
 S109	Low	Low	Low	Standby mode (outputs disabled)
	High	Low	Low	Full step
	Low	High	Low	Non-circular half step ("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 $V_{DD} = 5 \text{ V DC}$	$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{(5 * R_S)}$			$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * (5 * R_S)$
R_S (Typical Sense Resistor) = 0.1Ω				

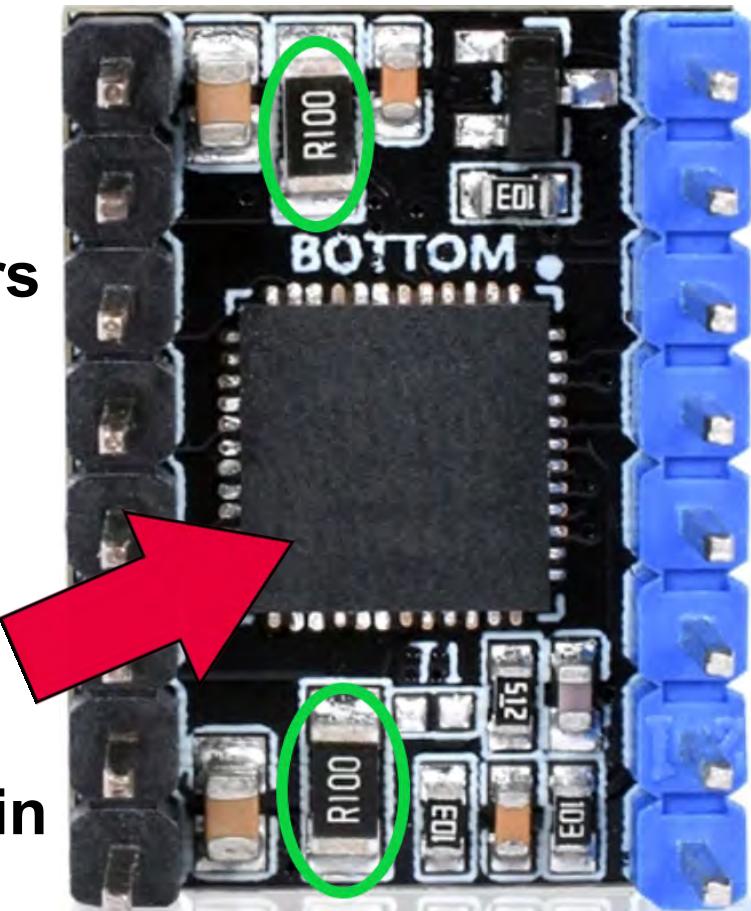


BIQU S109

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board. V_{DD} is an Output (**N**), **on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!**

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

BIQU S109

Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	-	-	A1
RST	-	MS2	MS1	MS0
NC	-	-	-	-
STEP	-	9	8	7
DIR	-	9	8	7

Stand By

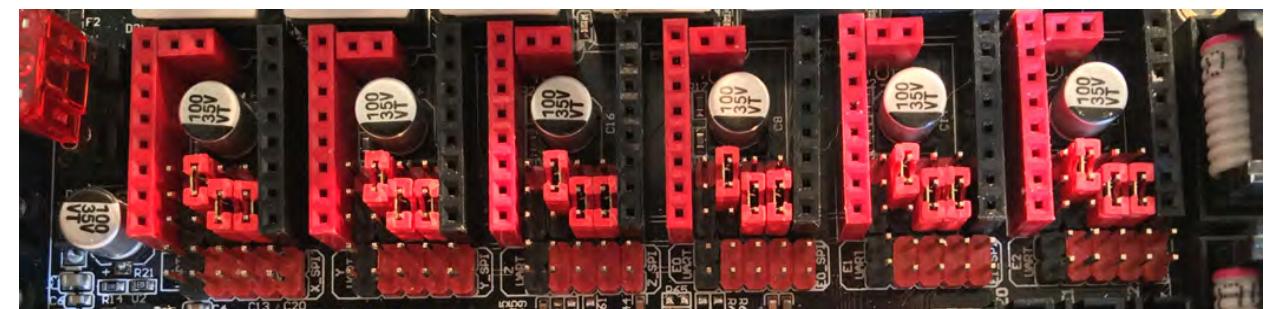
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	-	-	A1
RST	-	MS2	MS1	MS0
NC	-	9	-	-
STEP	-	9	8	7
DIR	-	-	8	7

STEP

Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



BQU S109

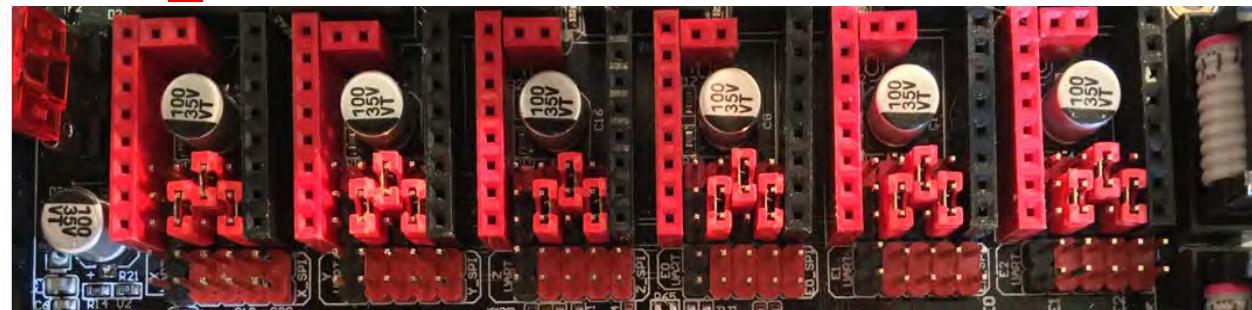
Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	-	-	A1
RST	-	MS2	MS1	MS0
NC	-	-	8	-
STEP	-	9	8	7
DIR	-	9	-	7
	-	-	-	GND

CCW 1 / 2

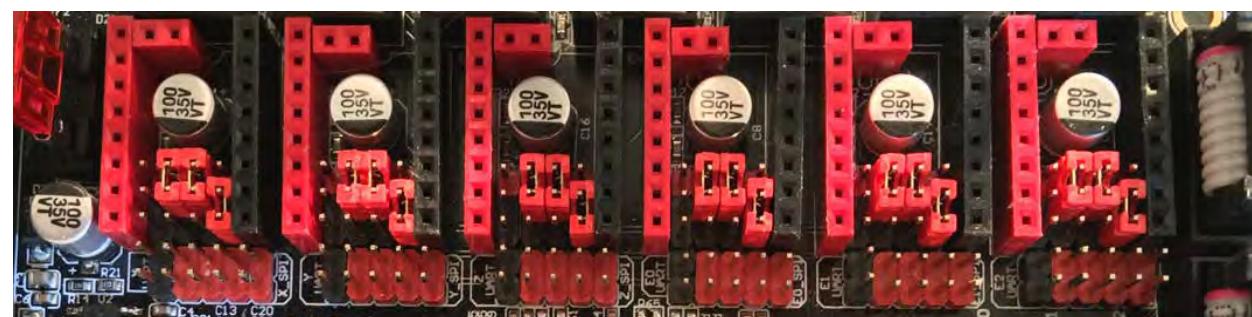
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	MS0
RST	-	-	-	-
NC	-	9	8	-
STEP	-	9	8	7
DIR	-	-	-	7
	-	-	-	GND

1 / 4

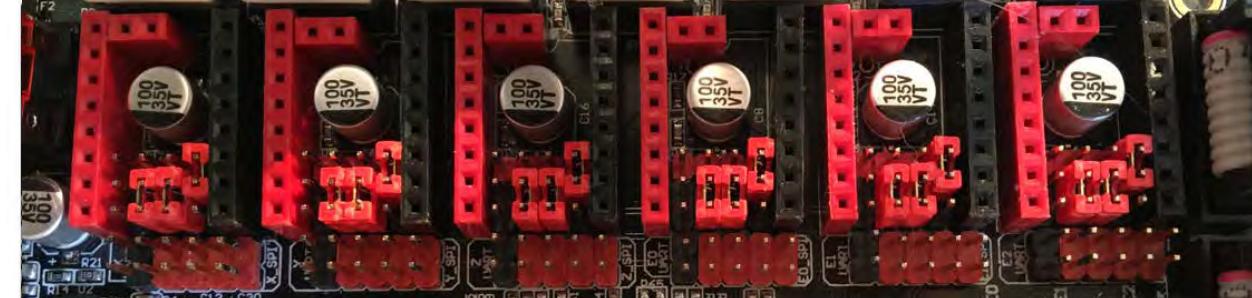
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	MS0
RST	-	-	-	-
NC	-	-	7	-
STEP	-	9	8	7
DIR	-	9	8	-
	-	-	-	GND

CW 1 / 2

Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



BQU S109

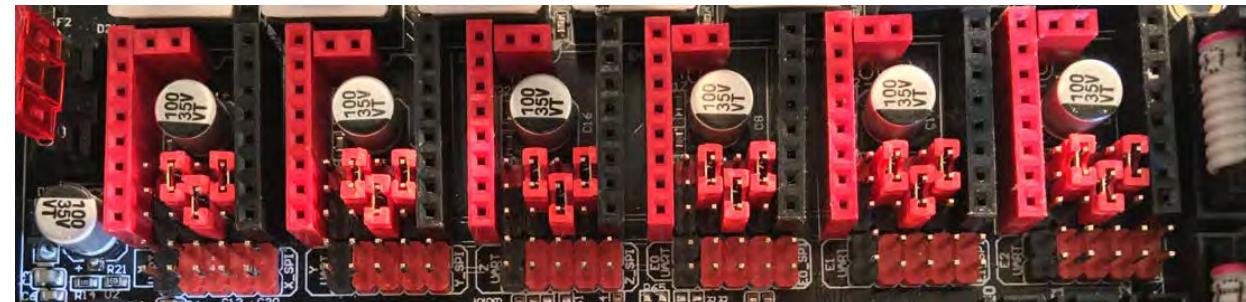
Stand-alone Mode

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	-	-	A1
RST	-	MS2	MS1	MS0
NC	-	9	7	8
STEP	-	9	8	7
DIR	-	8	-	GND

1 / 8

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

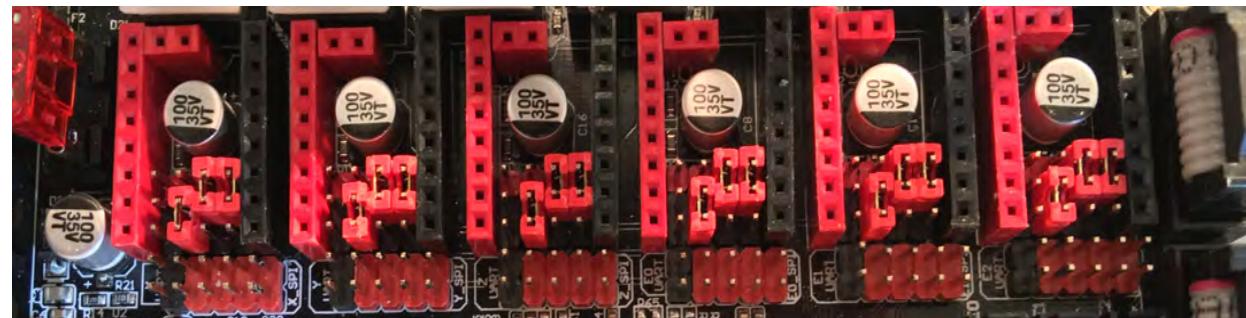
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	MS0
RST	-	8	7	9
NC	-	8	7	9
STEP	-	9	8	7
DIR	-	9	-	GND

1 / 16

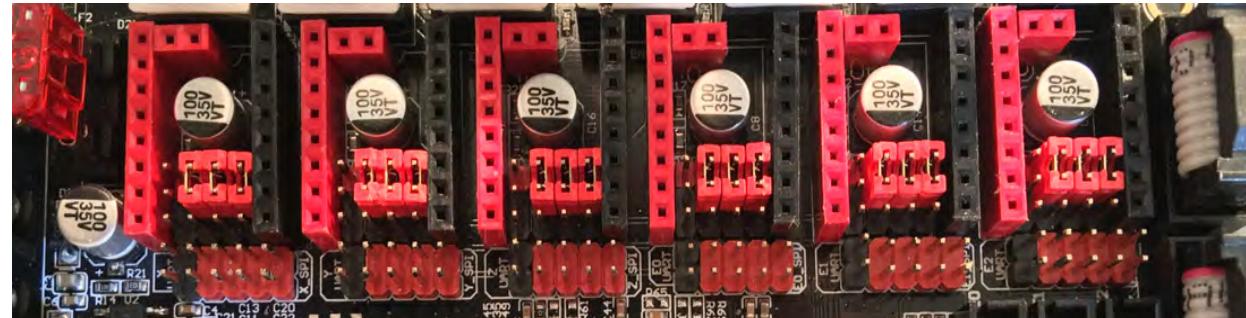
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	MS0
RST	-	9	8	7
NC	-	9	8	7
STEP	-	9	8	7
DIR	-	9	-	GND

1 / 32

Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



The (latest release of) Marlin Setup for BIQU S109 Drivers

NOTE: [Go to Appendix C](#), and then come back here for the changes to Marlin for BIQU S109 stepper motor drivers.

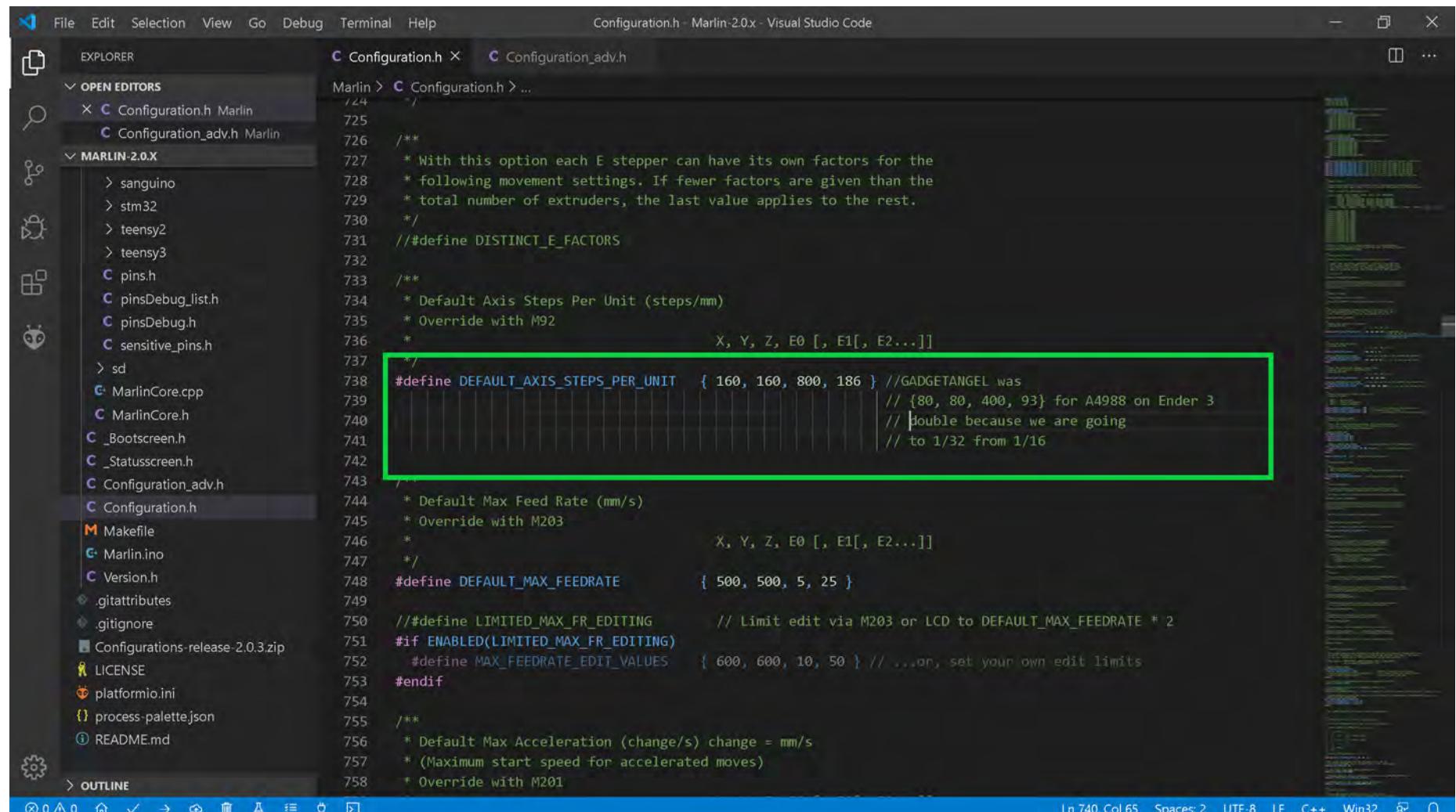
- Change the stepper motor drivers so that Marlin knows you are using BIQU S109 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU S109 drivers. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
 - The BIQU S109 is a drop in replacement for the A4988. Since Marlin does not have an option for BIQU S109 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 X Configuration_adv.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', 'TMC2208', 'TMC2209', 'TMC2660', 'TMC5130', 'TMC5160']  
677 /*  
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out  
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out  
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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.
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU S109 Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for BIQU S109 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.



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 line of code:

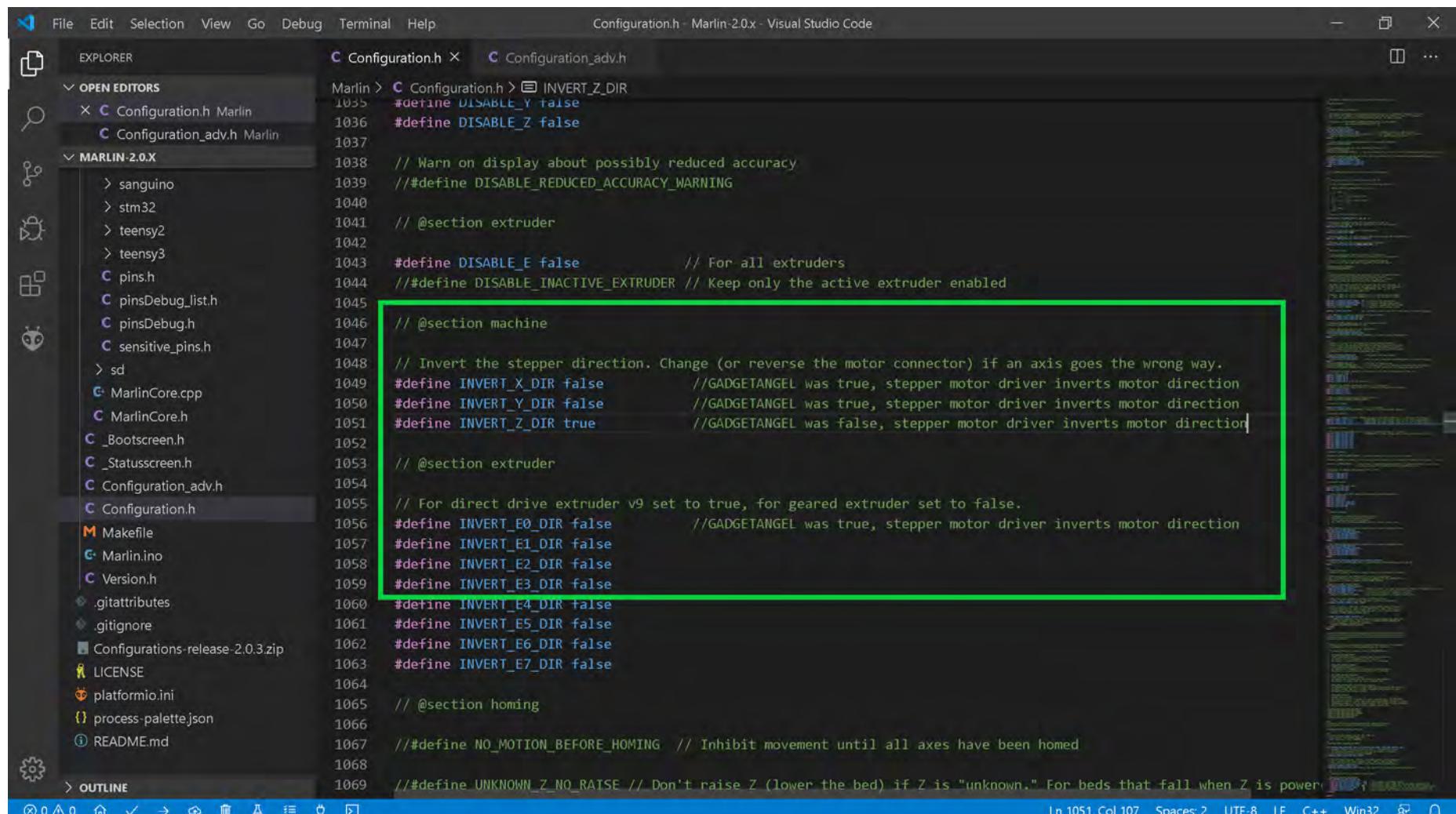
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU S109 Drivers

- **Optional Step:** I cannot find information on the BIQU S109 driver's impact on motor direction. 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 S109 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



```

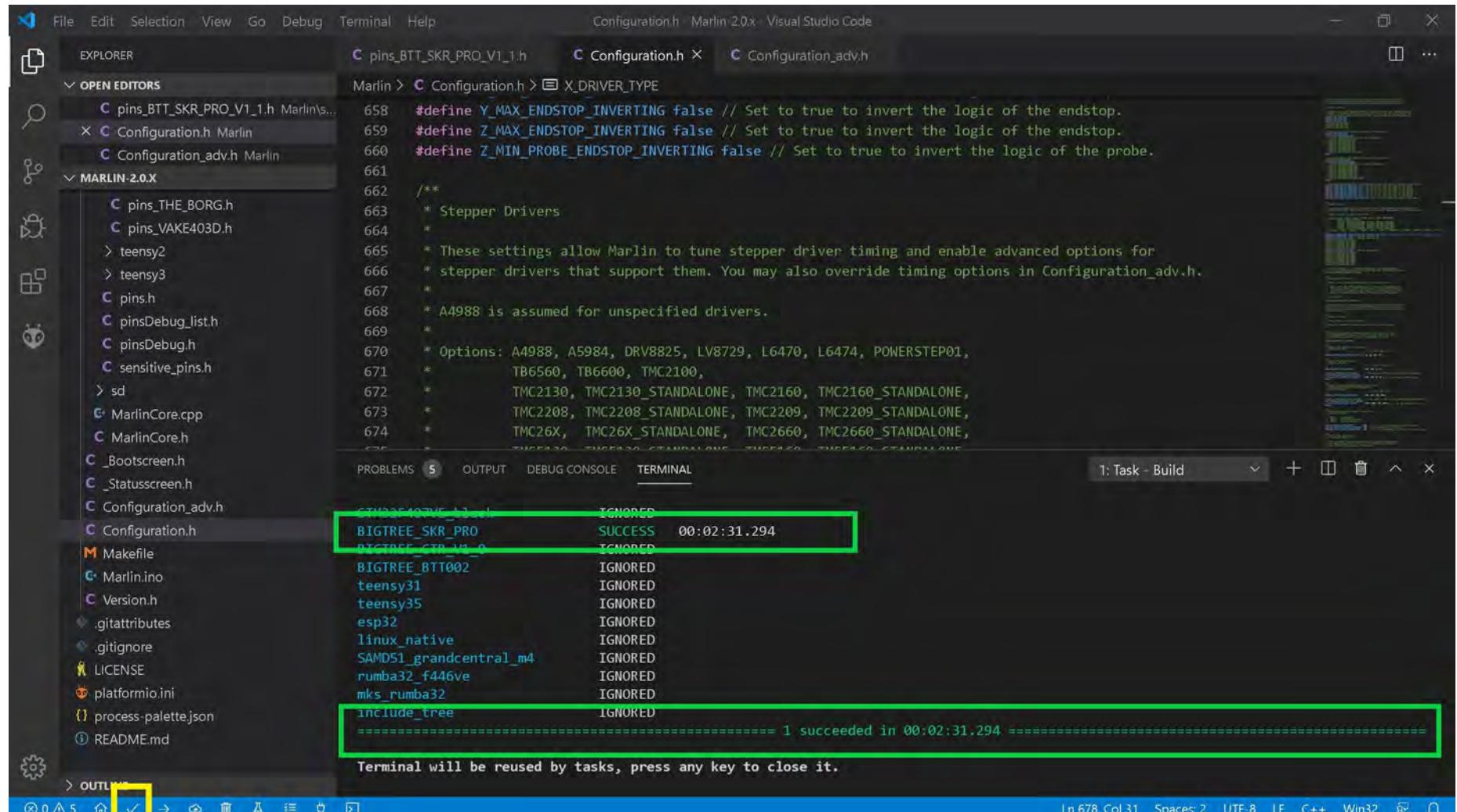
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 1036 #define DISABLE_Z 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      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  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     // GADGETANGEL was true, stepper motor driver inverts motor direction
  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
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU S109 Drivers

- The end of Marlin setup for BIQU S109 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

OPEN EDITORS

- C pins_BTT_SKR_PRO_V1_1.h
- C Configuration.h Marlin
- C Configuration_adv.h Marlin

MARLIN-2.0.X

- C pins_THE_BORG.h
- C pins_VAKE403D.h
- > teensy2
- > teensy3
- C pins.h
- C pinsDebug_list.h
- C pinsDebug.h
- C sensitive_pins.h
- > sd
- G MarlinCore.cpp
- C MarlinCore.h
- C _Bootscreen.h
- C _Statusscreen.h
- C Configuration_adv.h
- C Configuration.h
- M Makefile
- G Marlin.ino
- C Version.h
- .gitattributes
- .gitignore
- L LICENSE
- platformio.ini
- process-palette.json
- R README.md

pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h

```

568 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
569 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
570 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
571 /**
572 * Stepper Drivers
573 */
574 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
575 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
576 *
577 * A4988 is assumed for unspecified drivers.
578 */
579 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
580 * TB6560, TB6600, TMC2100,
581 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
582 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
583 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
584 * TMC2225, TMC2225_STANDALONE, TMC2610, TMC2610_STANDALONE

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

STM32F407VE-L1	IGNORED
BIGTREE_SKR_PRO	SUCCESS 00:02:31.294
BIGTREE_CTC_M4	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 =====

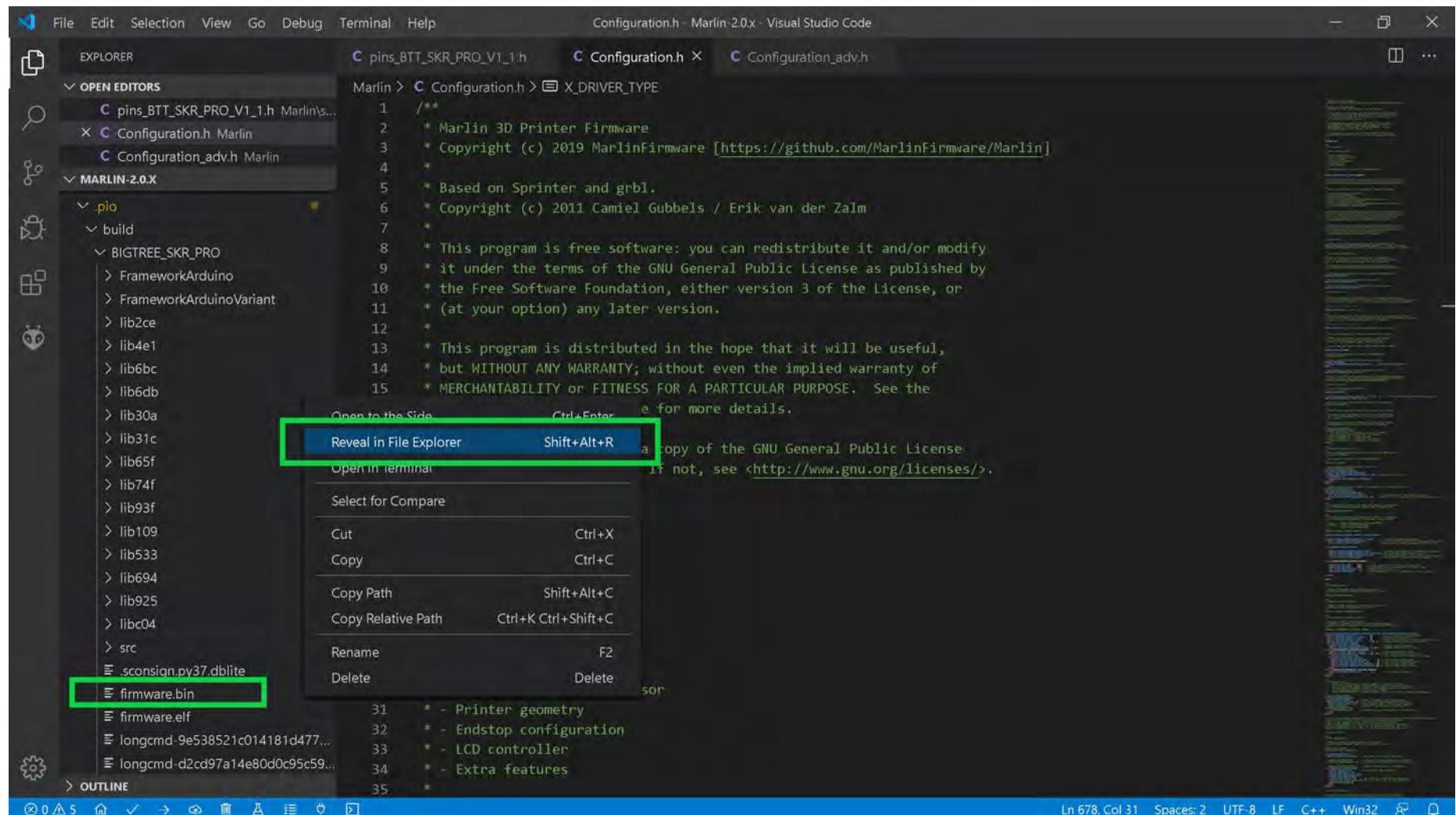
Terminal will be reused by tasks, press any key to close it.

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 S109 Drivers

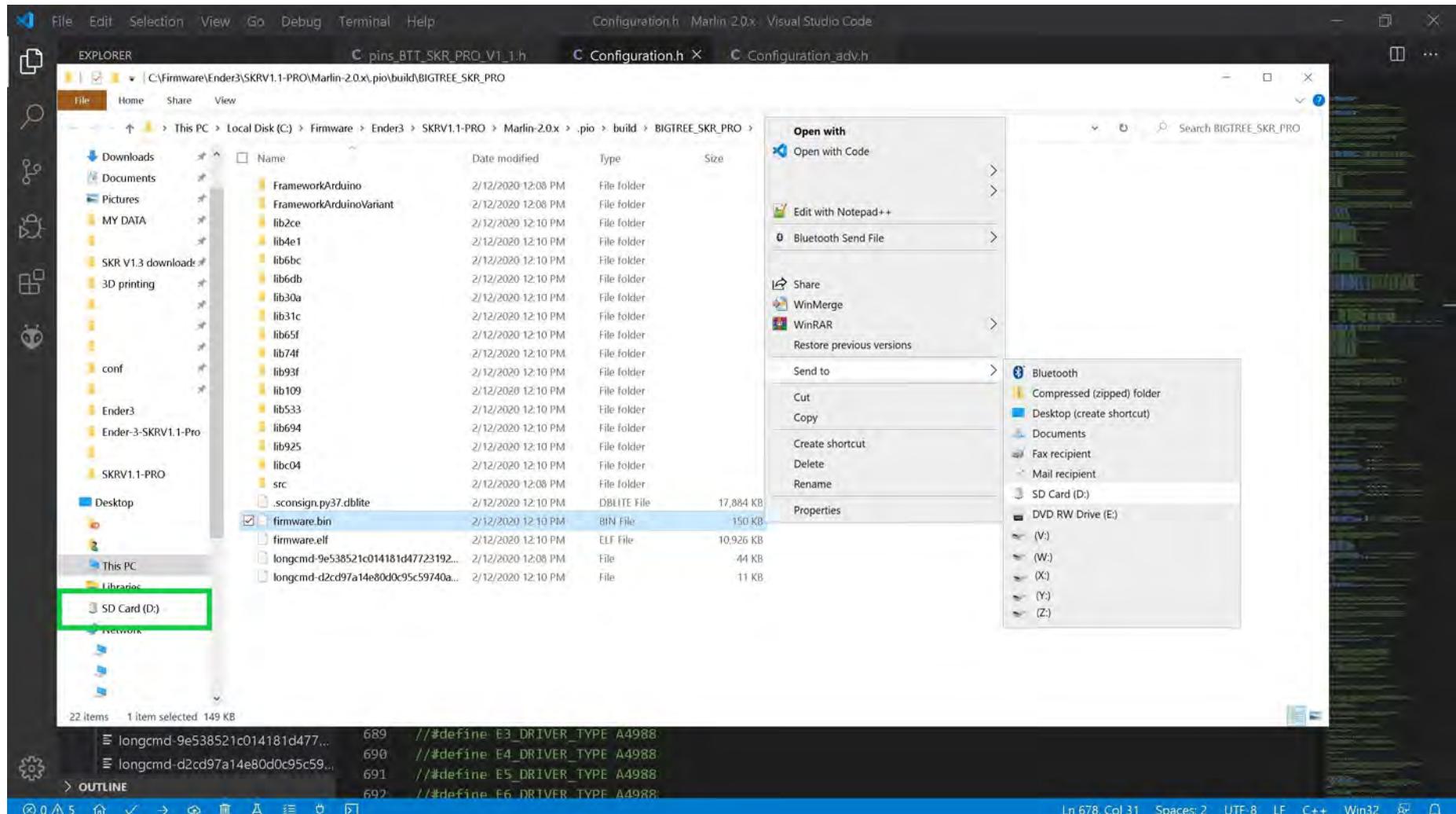
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



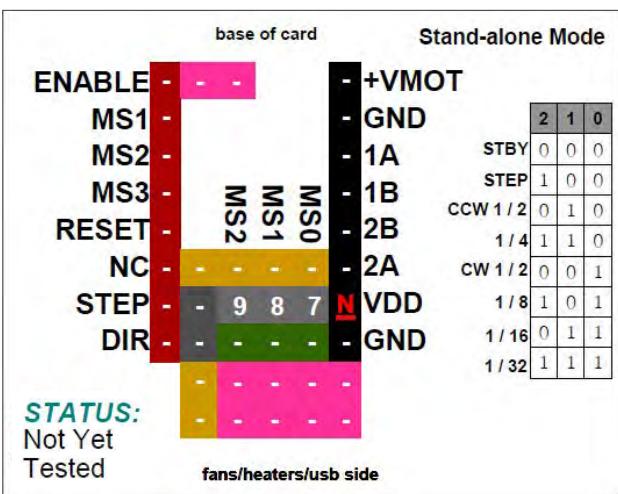
- Go to the next page.

The (latest release of) Marlin Setup for BIQU S109 Drivers

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

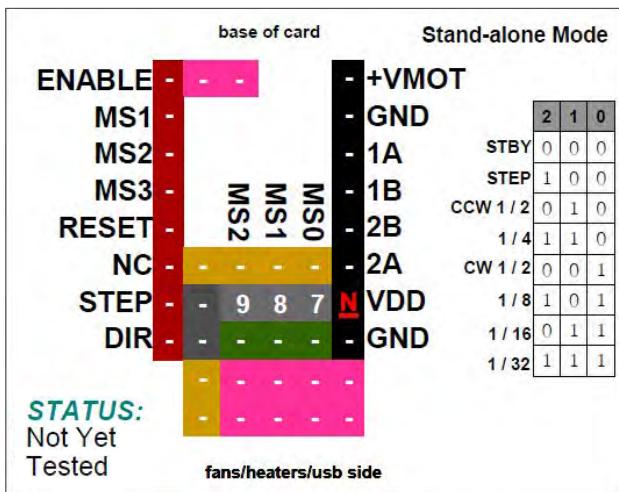


FYSETC S109

Note: V_{DD} is an Output (**N**), *on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!!*

Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

Driver Chip	MS2	MS1	MS0	Microstep Resolution
FYSETC	Low	Low	Low	Standby mode (outputs disabled)
S109	High	Low	Low	Full step
Maximum 32 Subdivision	Low	High	Low	Non-circular half step ("a")
50V DC 4A (peak)	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 $V_{DD} = 5 \text{ V DC}$	$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{(5 * R_S)}$		$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * (5 * R_S)$	
R_S (Typical Sense Resistor) = 0.1Ω				

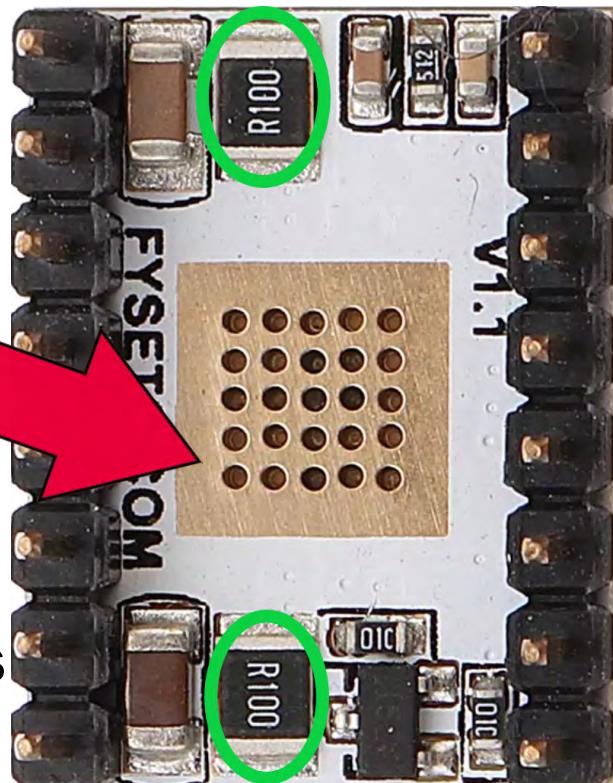


FYSETC S109

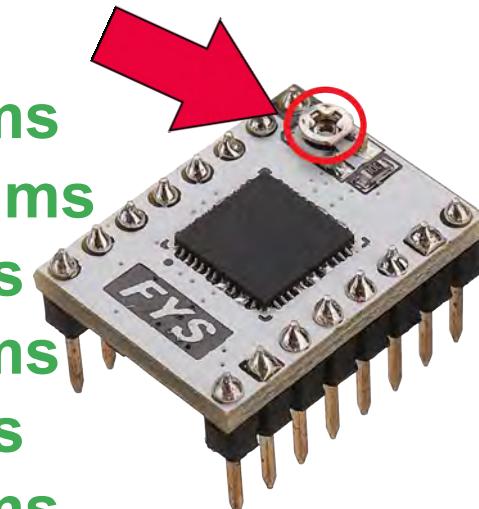
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board. V_{DD} is an Output (N), on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

FYSETC S109

Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	MS1	1B
RESET	-	MS2	2B
NC	-	-	2A
STEP	-	9 8 7	N VDD
DIR	-	9 8 7	- GND
	-	-	-

Stand By

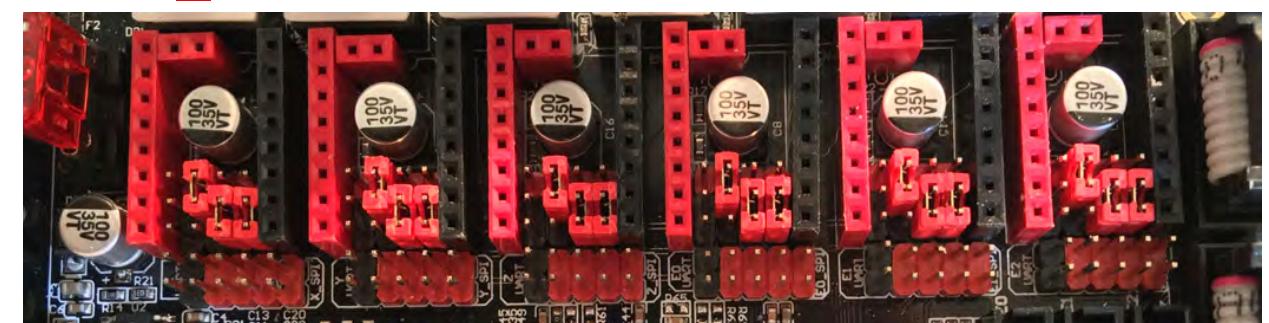
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	MS1	1B
RESET	-	MS2	2B
NC	-	9	- 2A
STEP	-	9 8 7	N VDD
DIR	-	8 7	- GND
	-	-	-

STEP

Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



FYSETC S109

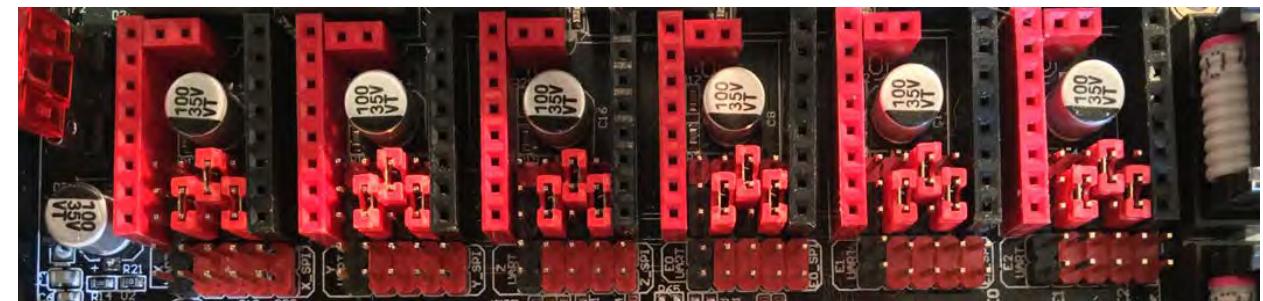
Stand-alone Mode

Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	MS2	MS1	MS0
NC	-	8	-
STEP	-	9	8 7 N
DIR	-	9	7 GND
	-	-	-

CCW 1 / 2

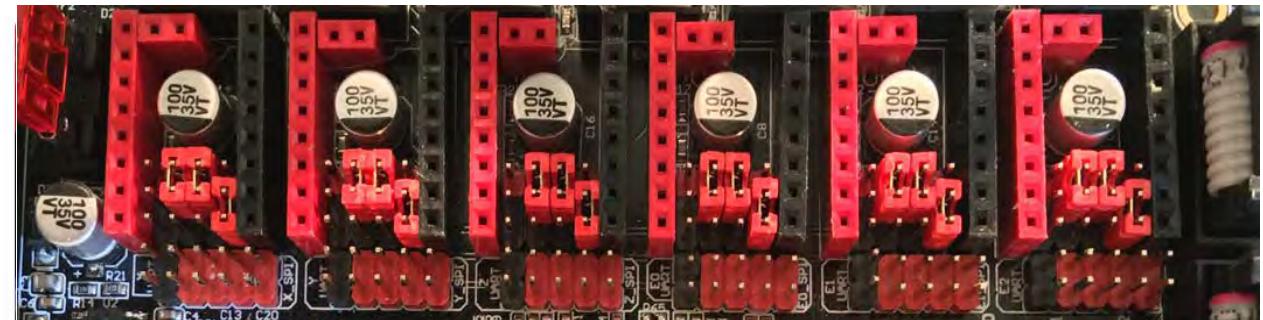
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	MS2	MS1	MS0
NC	-	9	8 -
STEP	-	9	8 7 N
DIR	-	-	7 GND
	-	-	-

1 / 4

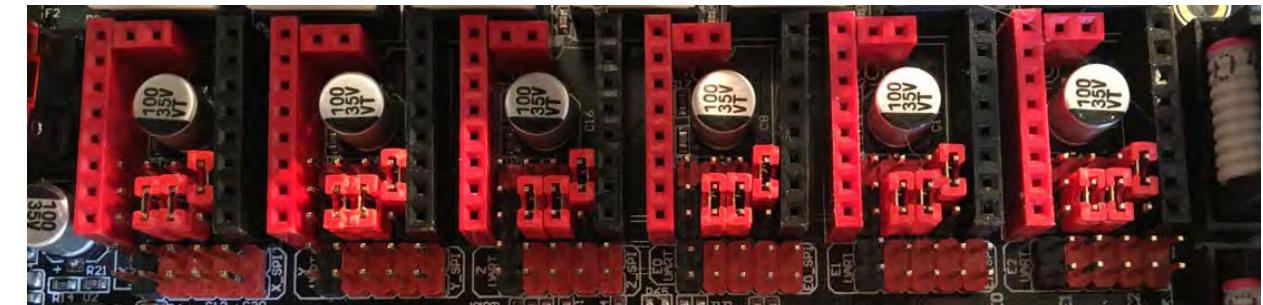
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	MS2	MS1	MS0
NC	-	-	7 -
STEP	-	9	8 7 N
DIR	-	9	8 -
	-	-	-

CW 1 / 2

Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



FYSETC S109

Stand-alone Mode

ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	-	MS2 MS1 MS0	2B
NC	-	9 8 7	2A
STEP	-	9 8 7	VDD
DIR	-	8	GND

1 / 8

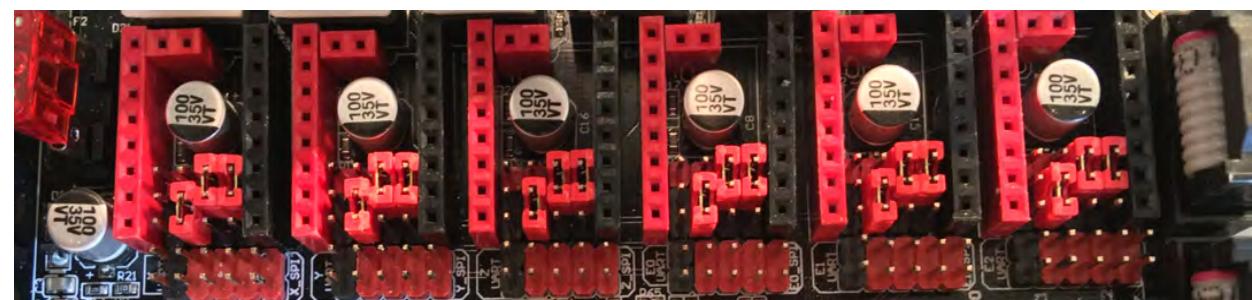
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	-	MS2 MS1 MS0	2B
NC	-	8 7	2A
STEP	-	9 8 7	VDD
DIR	-	9	GND

1 / 16

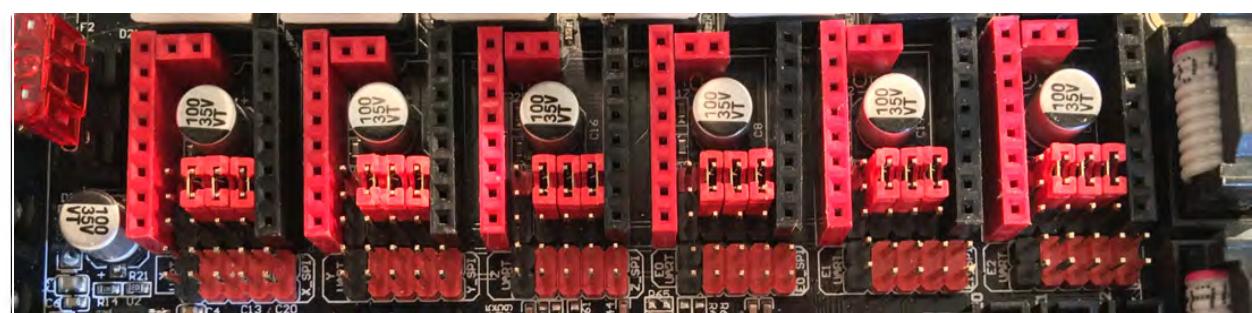
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



ENABLE	-	-	+VMOT
MS1	-	-	GND
MS2	-	-	1A
MS3	-	-	1B
RESET	-	MS2 MS1 MS0	2B
NC	-	9 8 7	2A
STEP	-	9 8 7	VDD
DIR	-	8	GND

1 / 32

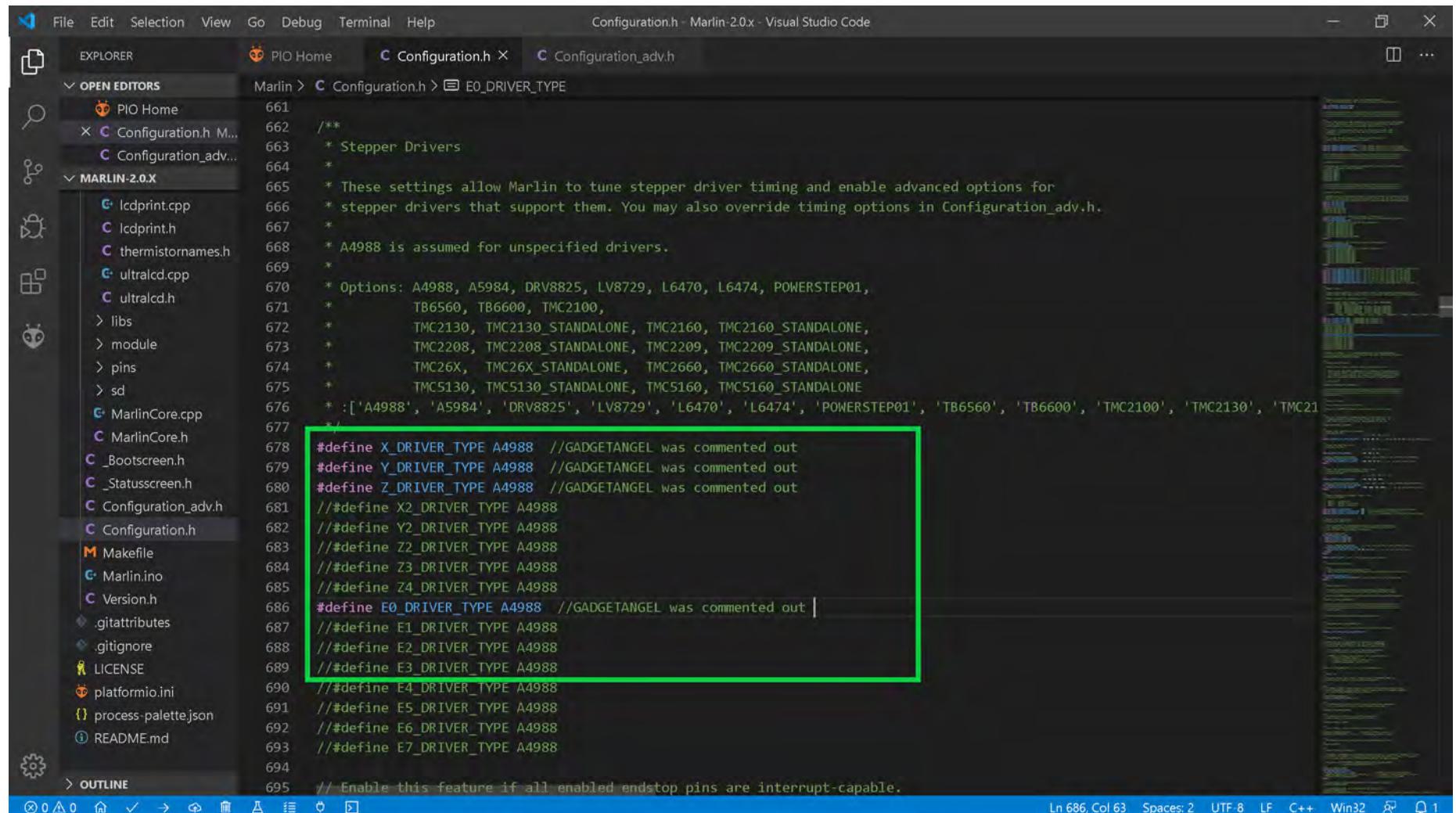
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



The (latest release of) Marlin Setup for FYSETC S109 Drivers

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for FYSETC S109 stepper motor drivers.

- Change the stepper motor drivers so that Marlin knows you are using FYSETC S109 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 S109 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 FYSETC S109 is a drop in replacement for the A4988. Since Marlin does not have an option for FYSETC S109 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 snippet from the `E0_DRIVER_TYPE` section:

```

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 /**
671 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
672 * TB6560, TB6600, TMC2100,
673 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
674 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
675 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
676 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
677 */
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL 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 //GADGETANGEL 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.

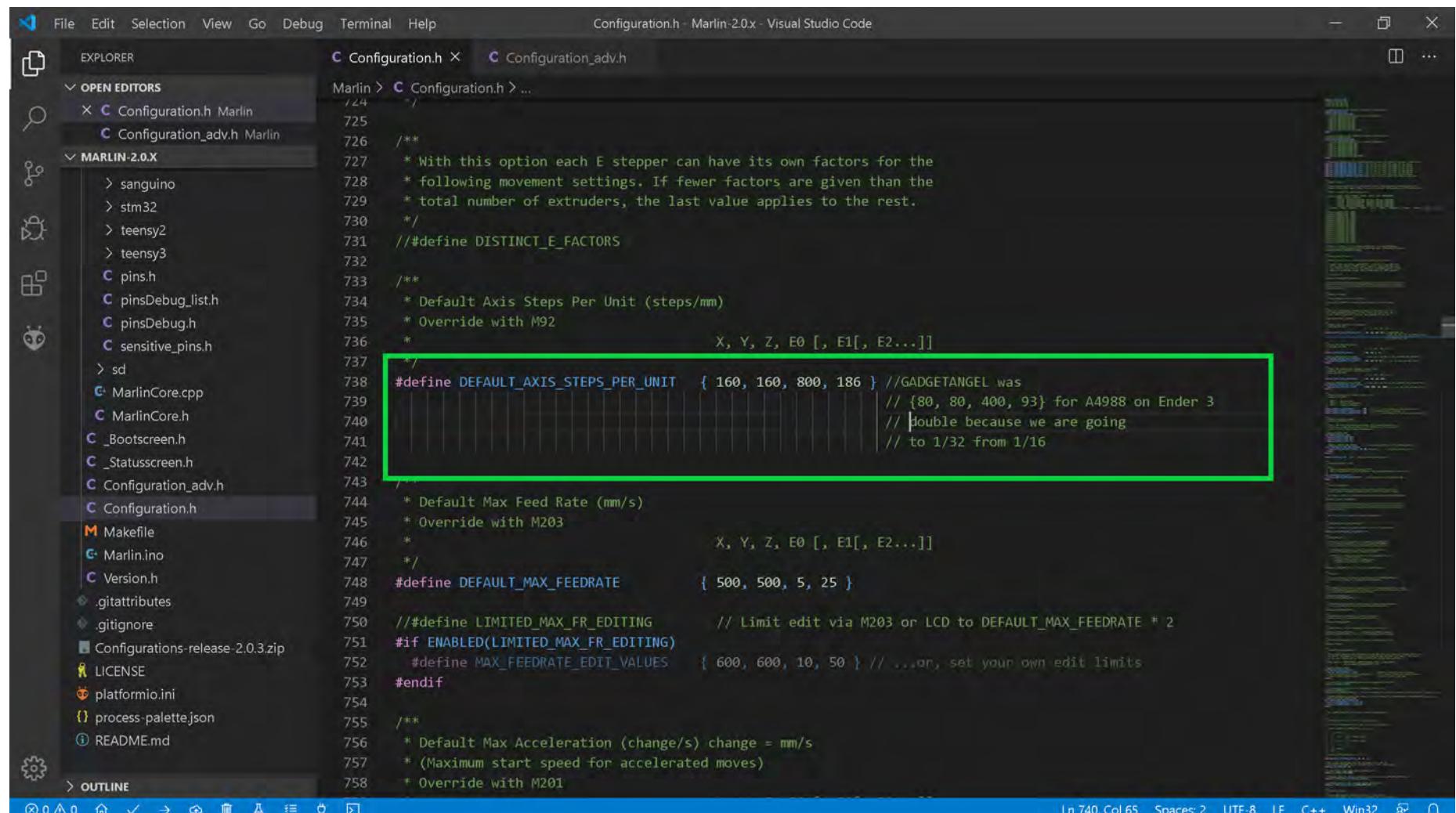
```

A green rectangular box highlights the lines from 678 to 694, which define the driver types for the X, Y, Z, and E0 axes. The lines are preceded by double slashes (//), indicating they are currently commented out.

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S109 Drivers

- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for FYSETC S109 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.



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 line of code:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S109 Drivers

- **Optional Step:** I cannot find information on the FYSETC S109 driver's impact on motor direction. 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 S109 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 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      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  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     // GADGETANGEL was true, stepper motor driver inverts motor direction
  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
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S109 Drivers

- The end of Marlin setup for FYSETC S109 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 OPEN EDITORS MARLIN-2.0.X

C pins_BTT_SKR_PRO_V1_1.h C Configuration.h X Configuration_adv.h

Marlin > C 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, PONERSTEP01,
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,

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build

STM32F103VC_L1.h	IGNORED
BIGTREETREE_SKR_PRO	SUCCESS 00:02:31.294
BIGTREETREE_SKR_V1_0	IGNORED
BIGTREETREE_BTT002	IGNORED
teensy31	IGNORED
teensy35	IGNORED
esp32	IGNORED
linux_native	IGNORED
SAMDS11_grandcentral_m4	IGNORED
rumba32_f446ve	IGNORED
mks_rumba32	IGNORED
include_tree	IGNORED

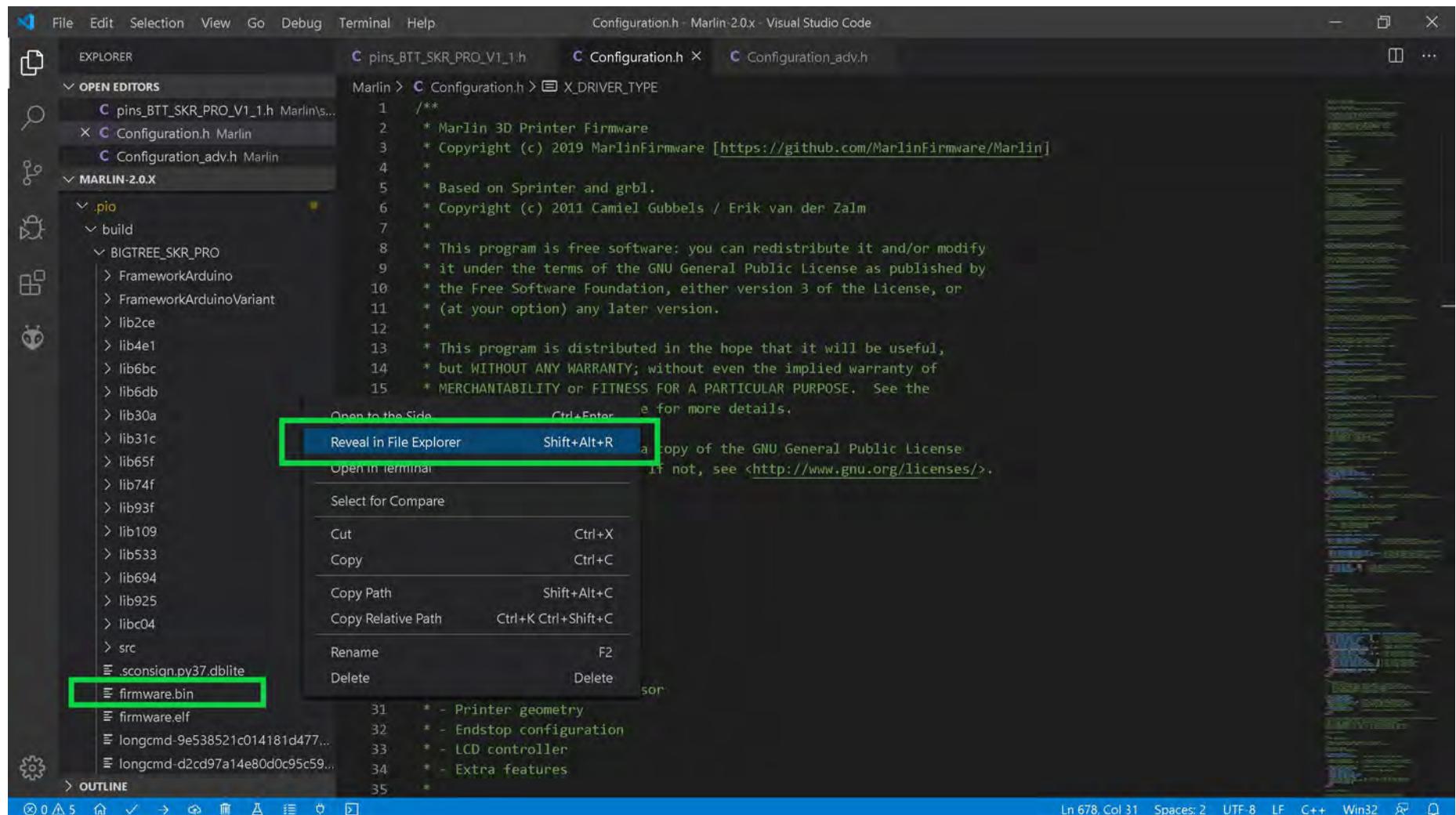
===== 1 succeeded in 00:02:31.294 =====

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 S109 Drivers

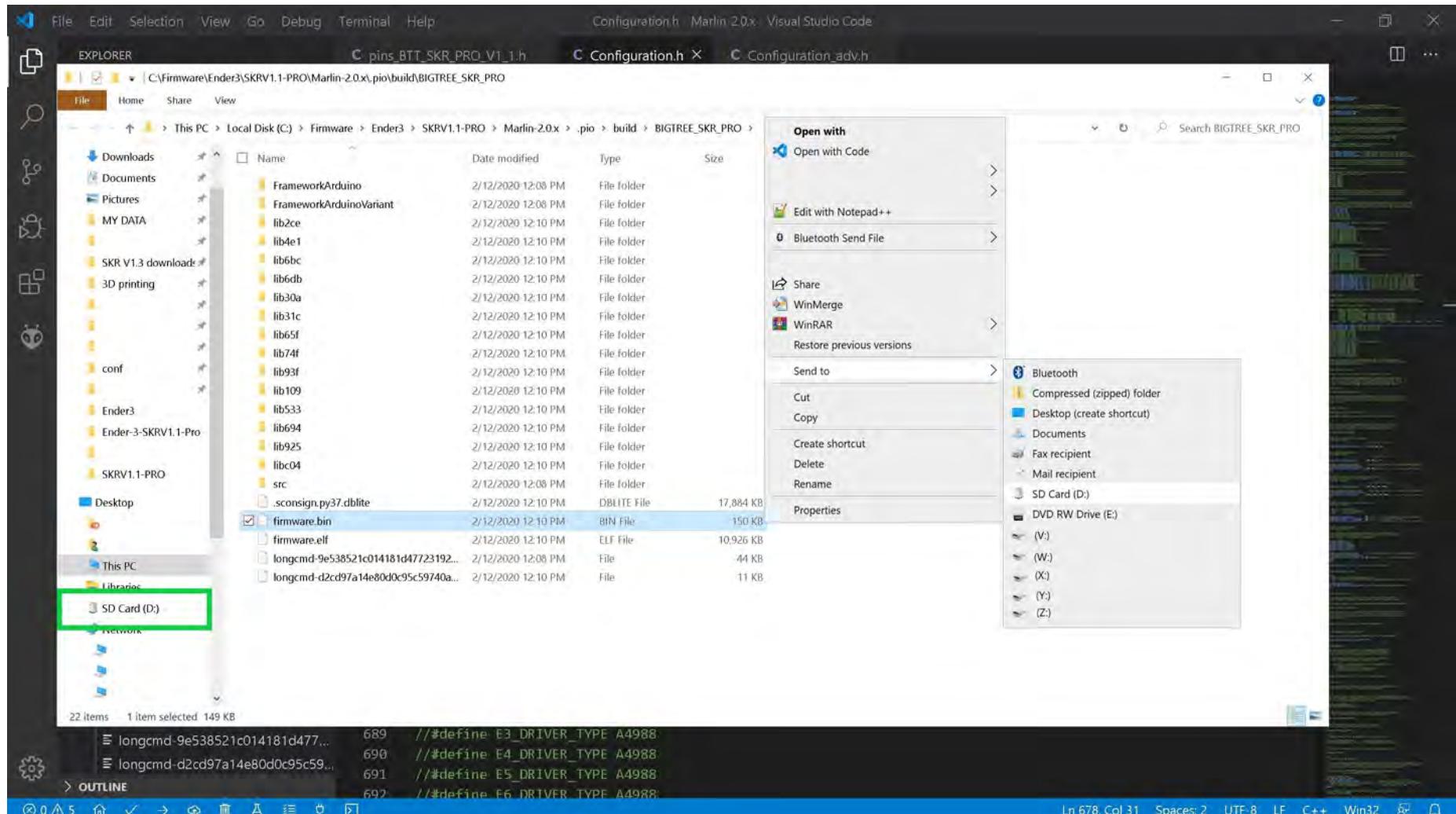
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer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



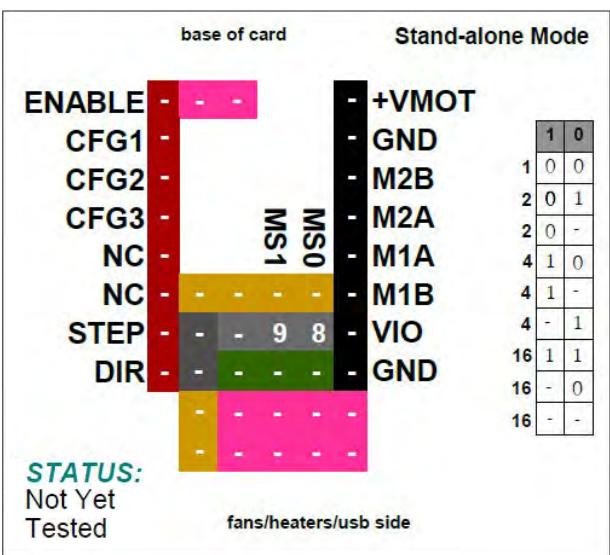
- Go to the next page.

The (latest release of) Marlin Setup for FYSETC S109 Drivers

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



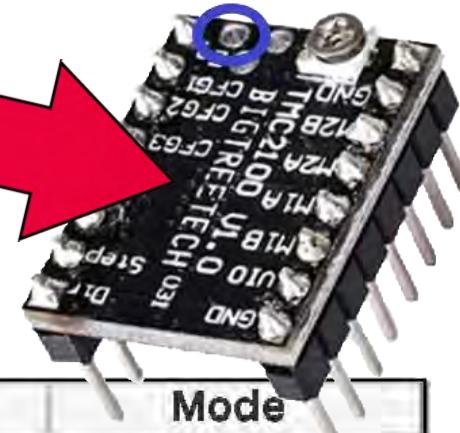
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



BIQU TMC2100

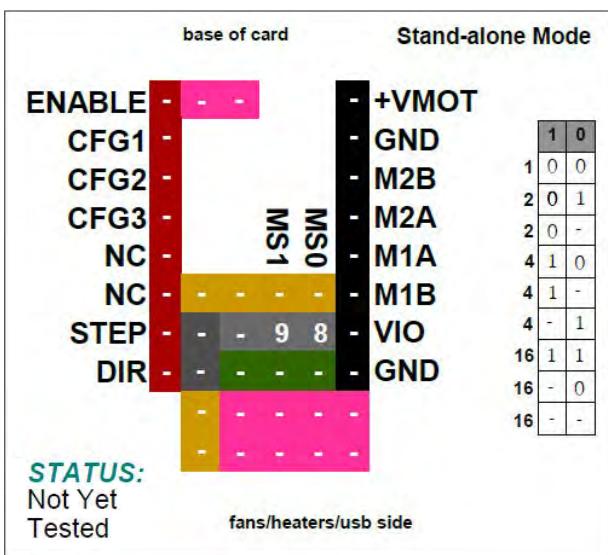
Stand-alone Mode

NOTE: Use the potentiometer (POT) on the top of the board or use the "**V_{ref} Test point**" location, as shown in **BLUE**, to adjust your driver board's V_{ref}



Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2100 Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	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

Driving Current Calculation Formula	I _{MAX} =V _{ref}	V _{ref} =I _{MAX}
R _S (Typical Sense Resistor)=0.11Ω	See Appendix B #1. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	See Appendix B #1. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$



BIQU TMC2100

Stand-alone Mode

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's " V_{ref} Test point" to set your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} 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: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN

- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

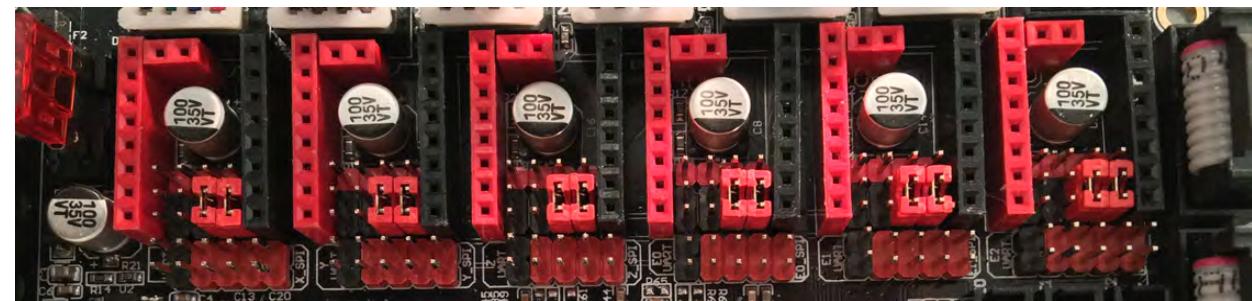
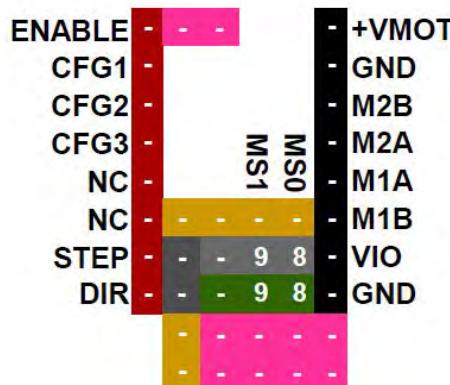
BIQU TMC2100

Stand-alone Mode

Stand-alone Mode

STEPInterpolation: **none**

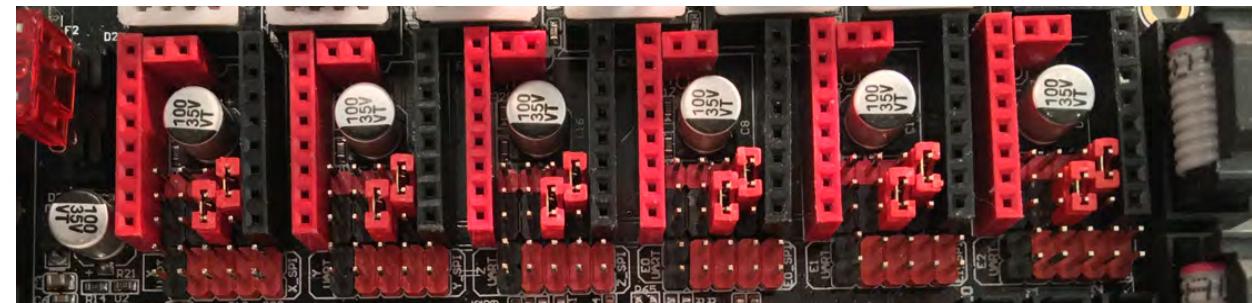
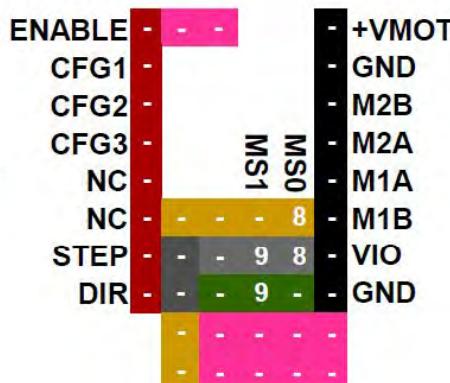
SpreadCycle



Stand-alone Mode

1 / 2Interpolation: **none**

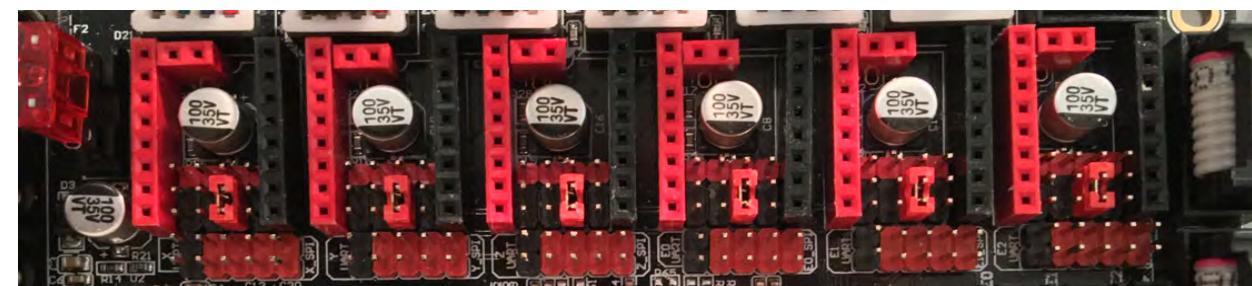
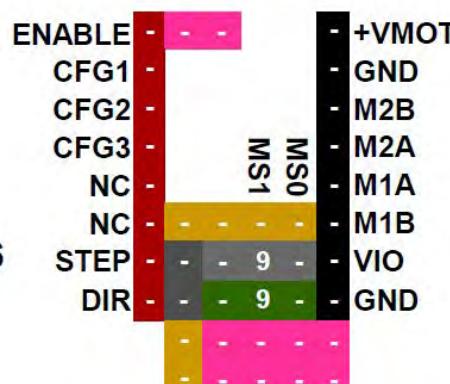
SpreadCycle



Stand-alone Mode

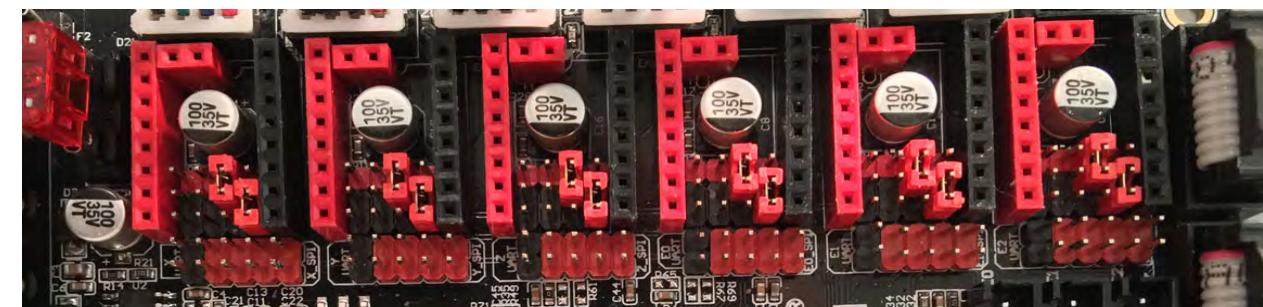
1 / 2Interpolation: **1/256**

SpreadCycle

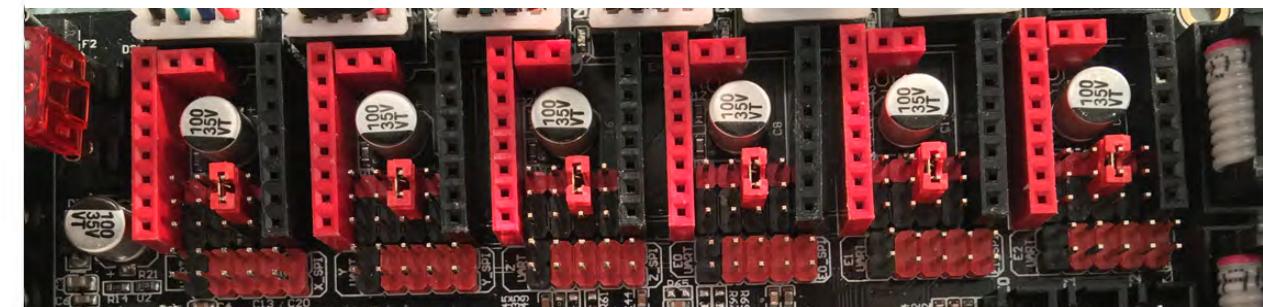


BIQU TMC2100**Stand-alone Mode**

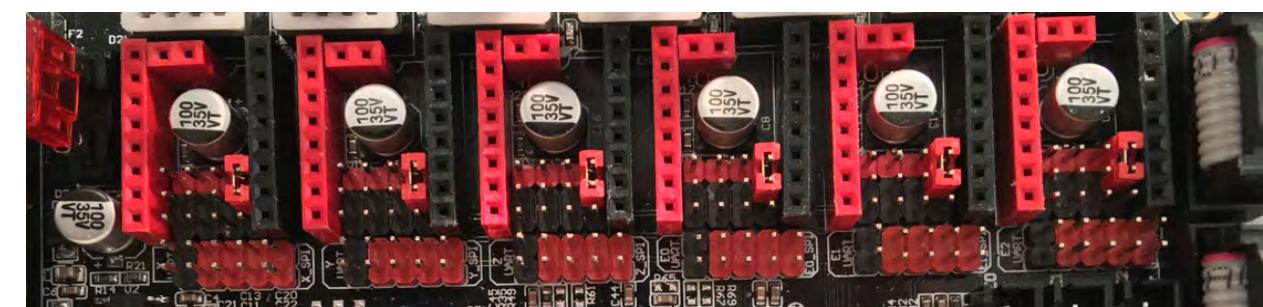
Stand-alone Mode	ENABLE	-	-	+VMOT
1 / 4	CFG1	-	-	GND
Interpolation: none	CFG2	-	-	M2B
SpreadCycle	CFG3	-	-	M2A
	NC	MS1	MS0	M1A
	NC	-	9	M1B
	STEP	-	9 8	VIO
	DIR	-	8	GND



Stand-alone Mode	ENABLE	-	-	+VMOT
1 / 4	CFG1	-	-	GND
Interpolation: 1/256	CFG2	-	-	M2B
SpreadCycle	CFG3	-	-	M2A
	NC	MS1	MS0	M1A
	NC	-	9	M1B
	STEP	-	9	VIO
	DIR	-	-	GND



Stand-alone Mode	ENABLE	-	-	+VMOT
1 / 4	CFG1	-	-	GND
Interpolation: 1/256	CFG2	-	-	M2B
StealthChop	CFG3	-	-	M2A
	NC	MS1	MS0	M1A
	NC	-	8	M1B
	STEP	-	8	VIO
	DIR	-	-	GND



BIQU TMC2100

Stand-alone Mode

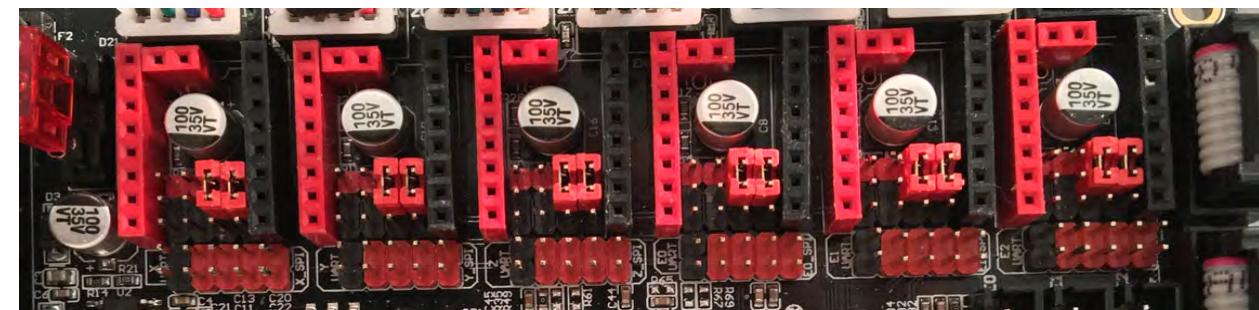
Stand-alone Mode

1 / 16

Interpolation: none

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	MS1	MS0	M1A
NC	-	9 8	M1B
STEP	-	9 8	VIO
DIR	-	-	GND



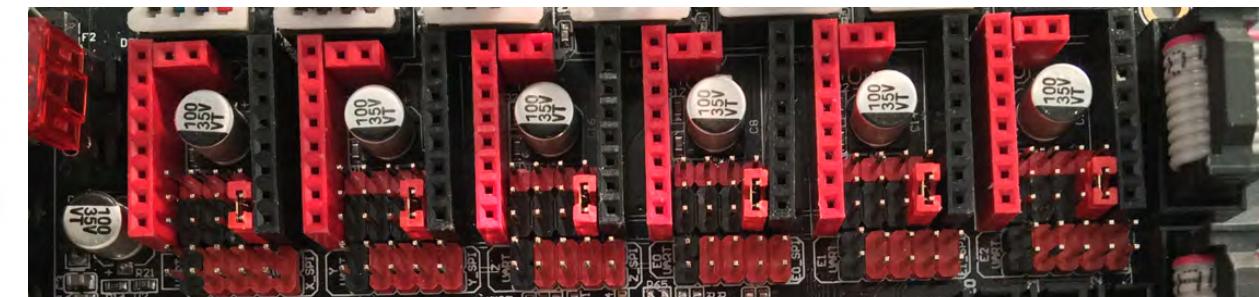
Stand-alone Mode

1 / 16

Interpolation: 1/256

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	MS1	MS0	M1A
NC	-	-	M1B
STEP	-	-	VIO
DIR	-	8	GND



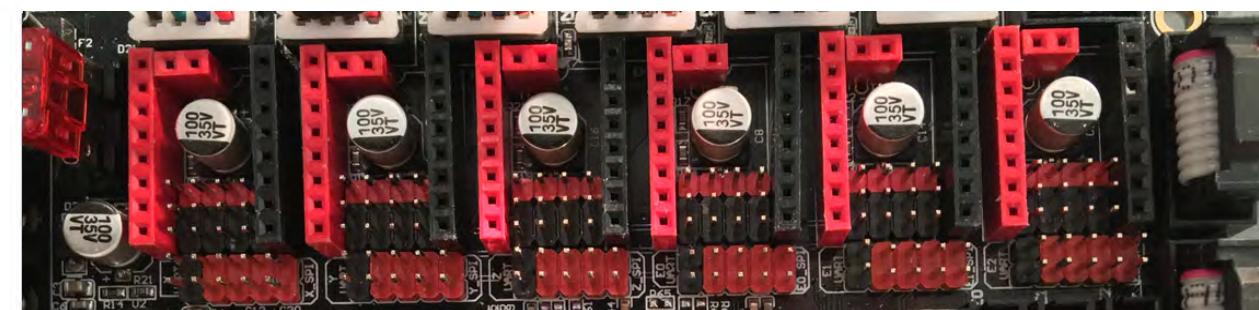
Stand-alone Mode

1 / 16

Interpolation: 1/256

StealthChop

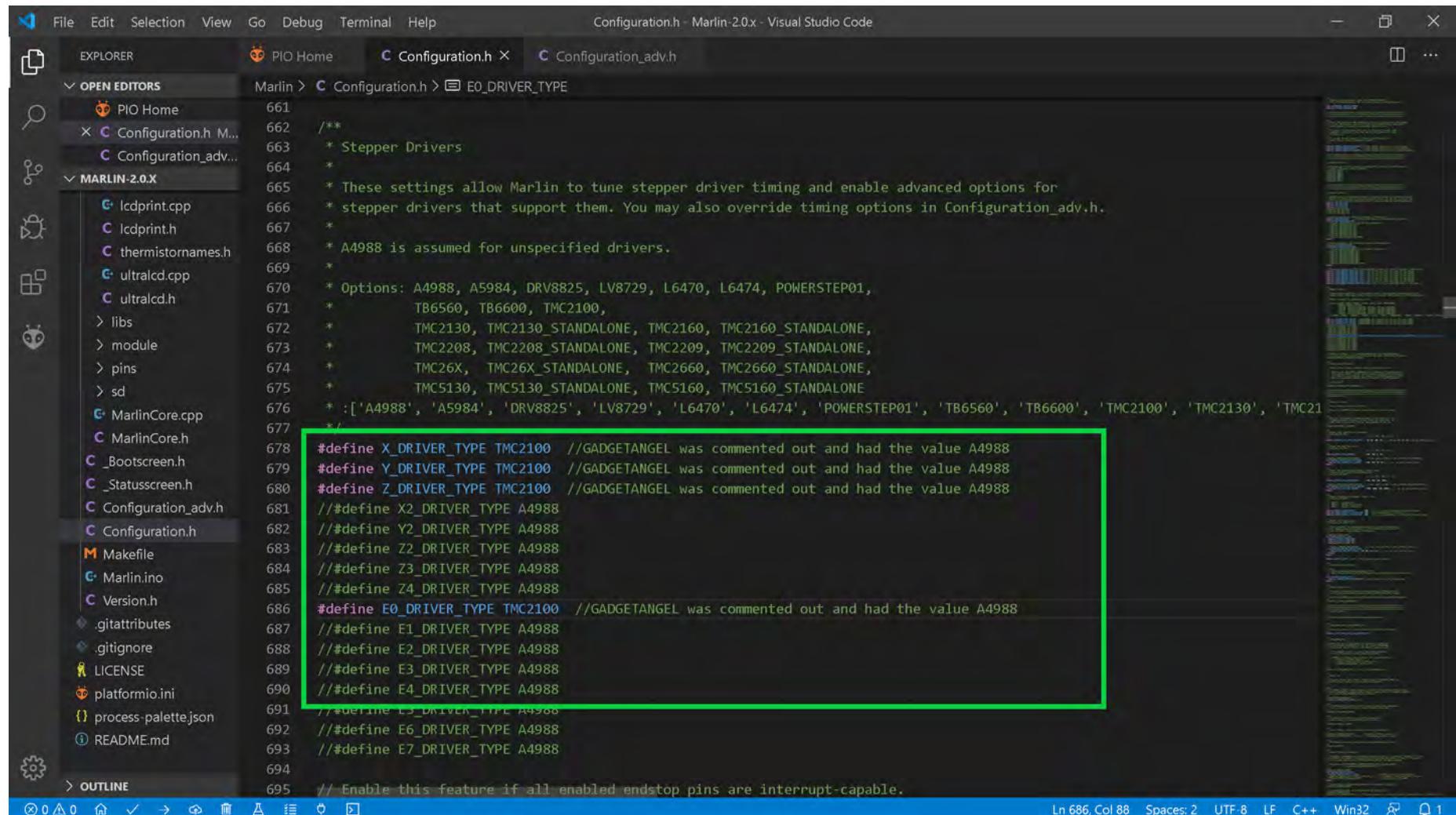
ENABLE	-	-	+VMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	MS1	MS0	M1A
NC	-	-	M1B
STEP	-	-	VIO
DIR	-	-	GND



The (latest release of) Marlin Setup for BIQU TMC2100 Drivers in Stand-alone Mode

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

- Change the stepper motor drivers so that Marlin knows you are using BIQU 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 BIQU 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:

```

#define X_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988

```

A green rectangular box highlights these four lines. The rest of the code in the file is mostly commented out with double slashes (//).

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2100 Drivers in Stand-alone Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to for BIQU TMC2100 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, 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 OPEN EDITORS MARLIN-2.0.X

```

Configuration.h Configuration_adv.h
Marlin > Configuration.h > DEFAULT_AXIS_STEPS_PER_UNIT
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 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps

742 /**
743 * Default Max Feed Rate (mm/s)
744 * Override with M203
745 *
746 * X, Y, Z, E0 [, E1[, E2...]]
747 */
748 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

749 /**
750 * Limited Max Feed Rate (mm/s)
751 * Override with M203
752 */
753 #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
754 #if ENABLED(LIMITED_MAX_FR_EDITING)
755 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
756#endif

757 /**
758 * Default Max Acceleration (change/s) change = mm/s
759 * (Maximum start speed for accelerated moves)
760 */
761 // Override with M201

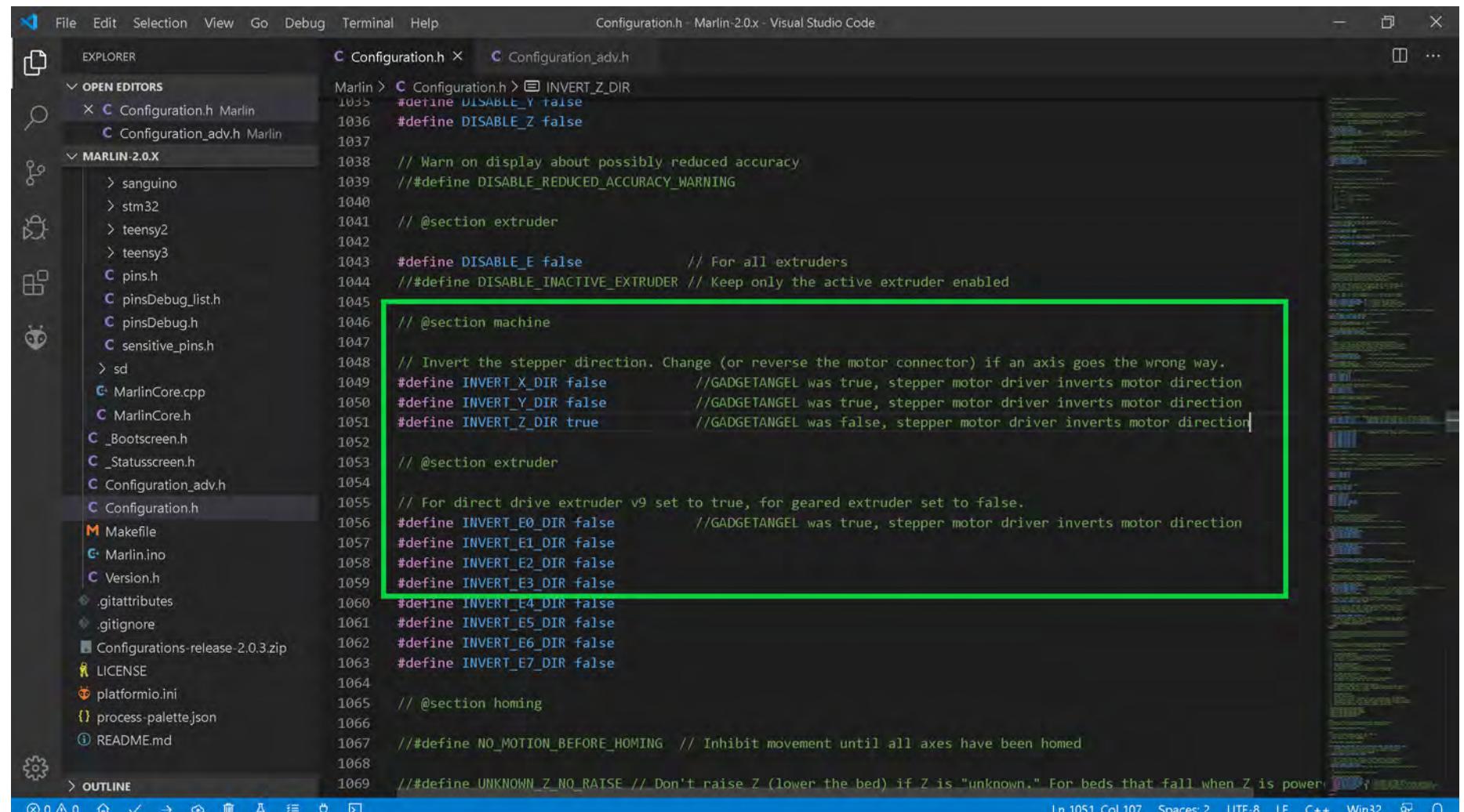
```

Ln 738, Col 62 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU 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



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 files and folders. The main code area displays the following configuration snippet:

```

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        //GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         //GADGETANGEL was false, stepper motor driver inverts motor direction
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       //GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 //##define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 //##define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

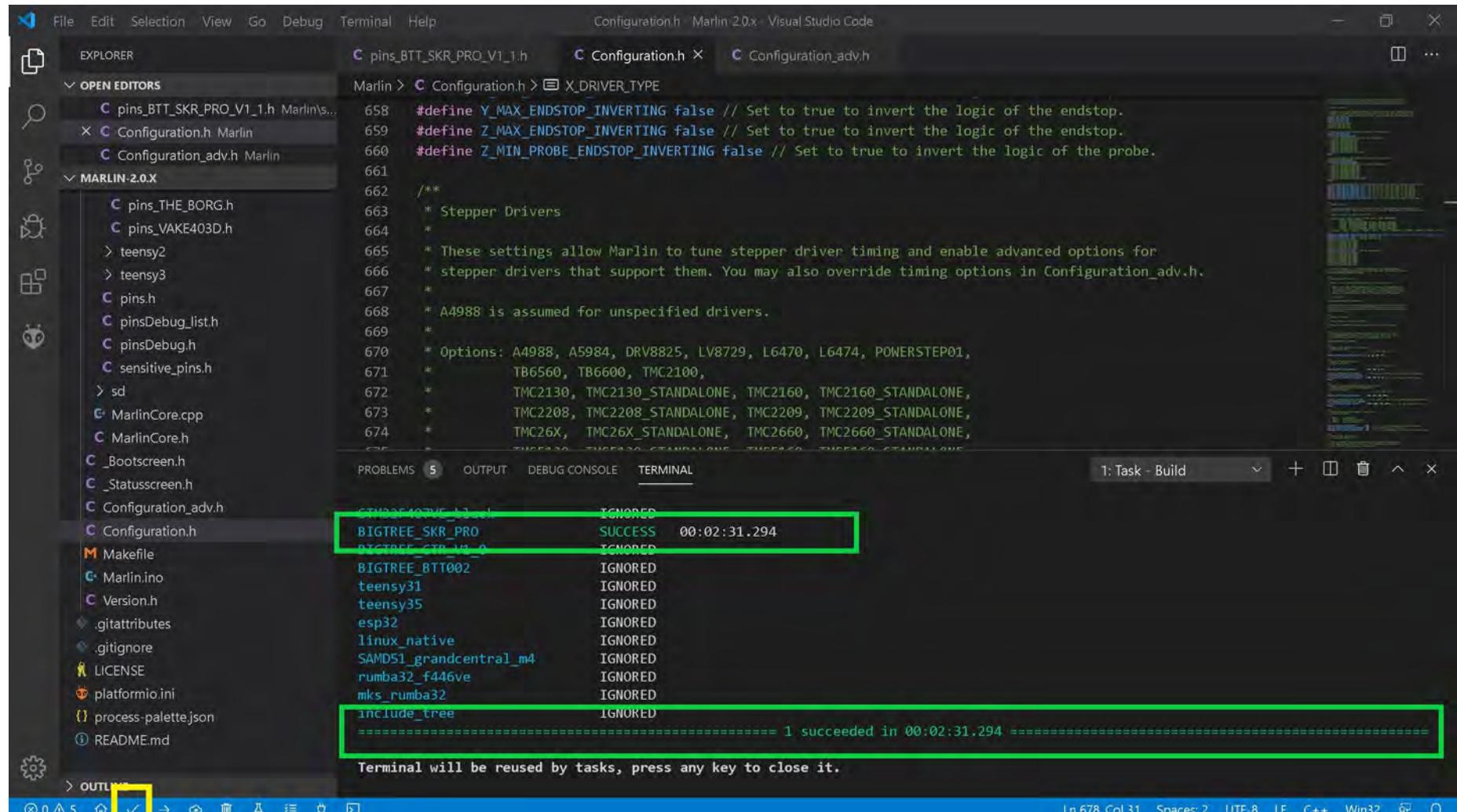
```

A green rectangular box highlights the section of code from line 1048 to line 1059, which defines the inverting of stepper directions for extruders. This is the part of the code that needs to be modified based on the user's requirements.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2100 Drivers in Stand-alone Mode

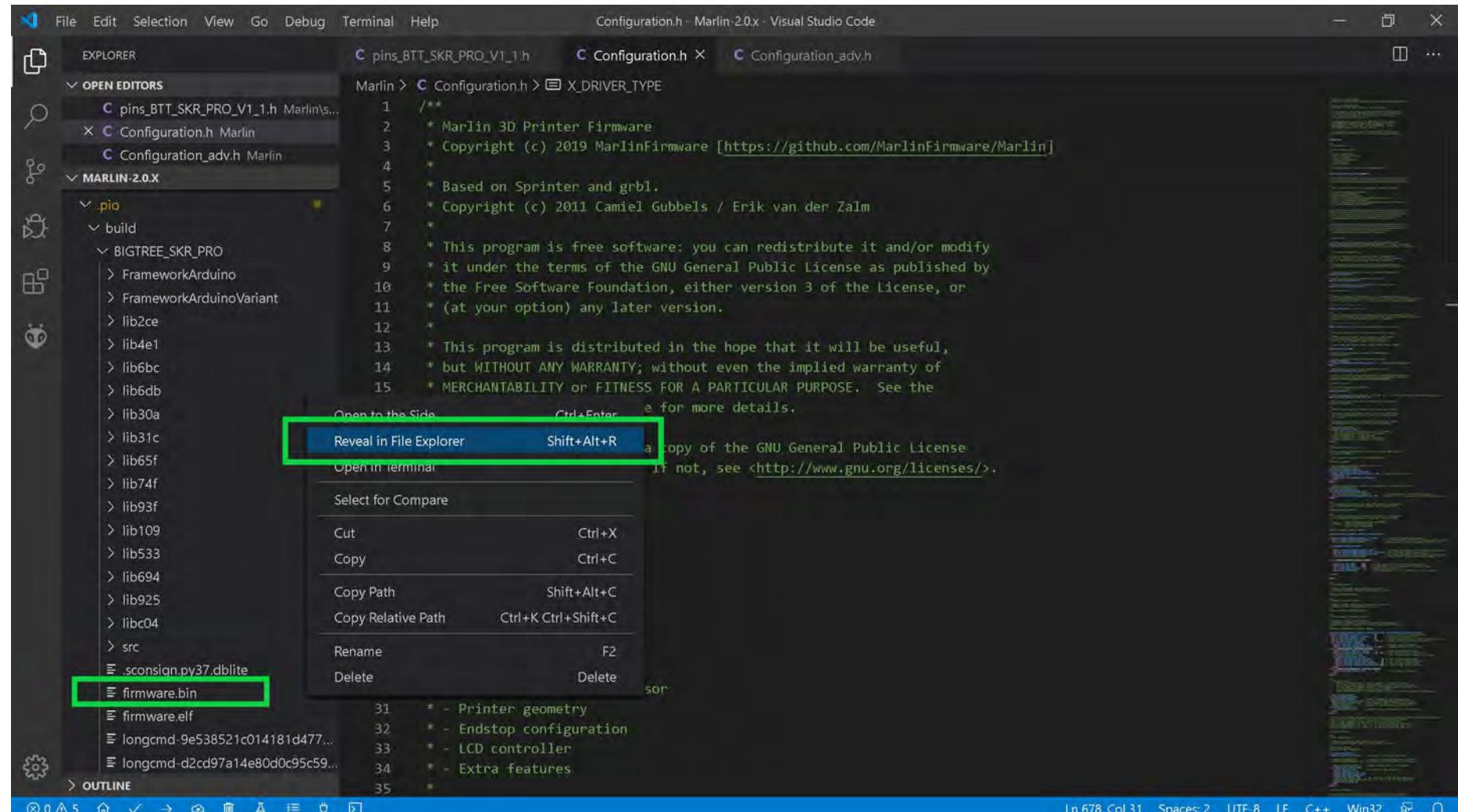
- The end of Marlin setup for BIQU TMC2100 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- 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 TMC2100 Drivers in Stand-alone Mode

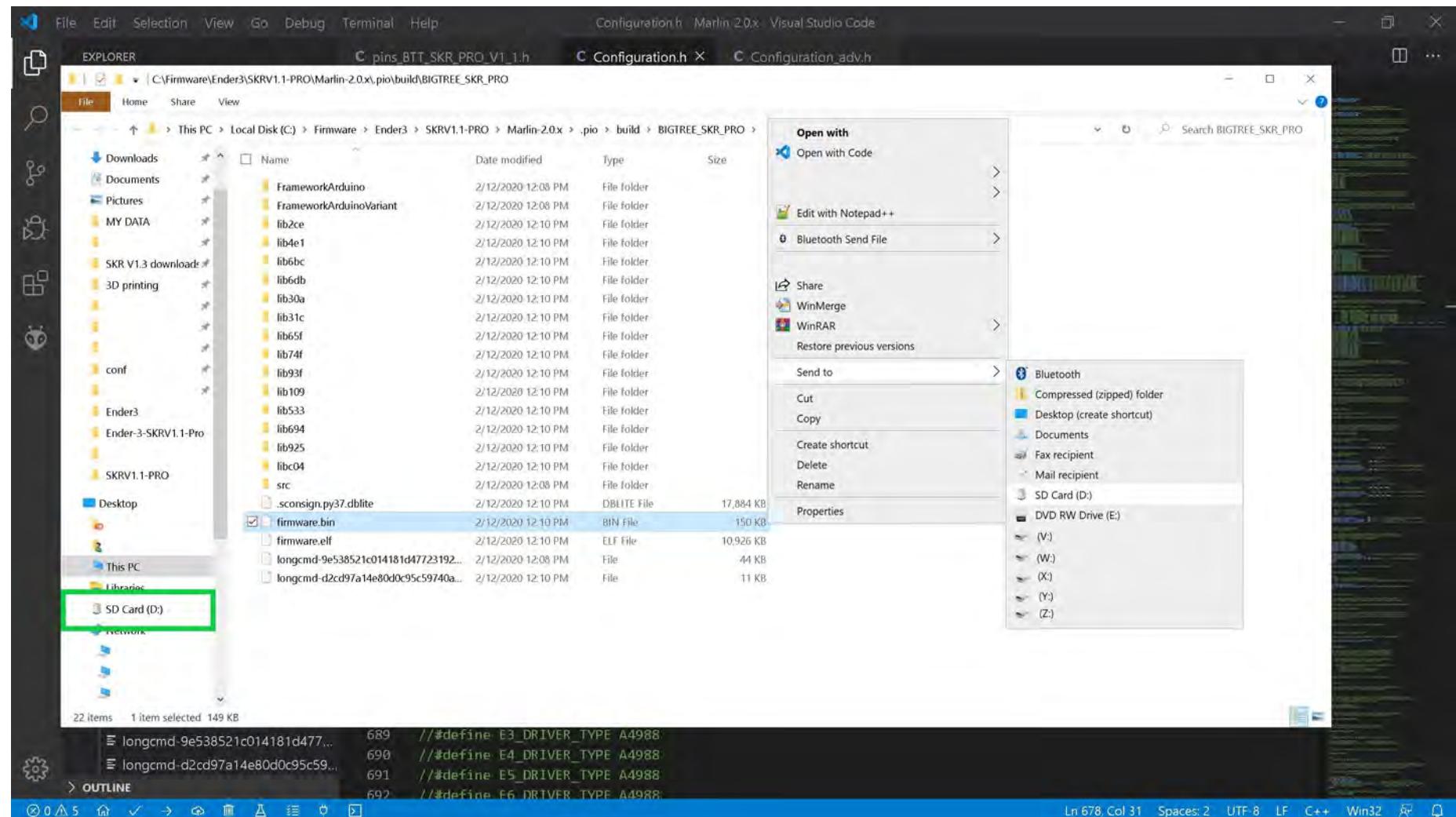
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



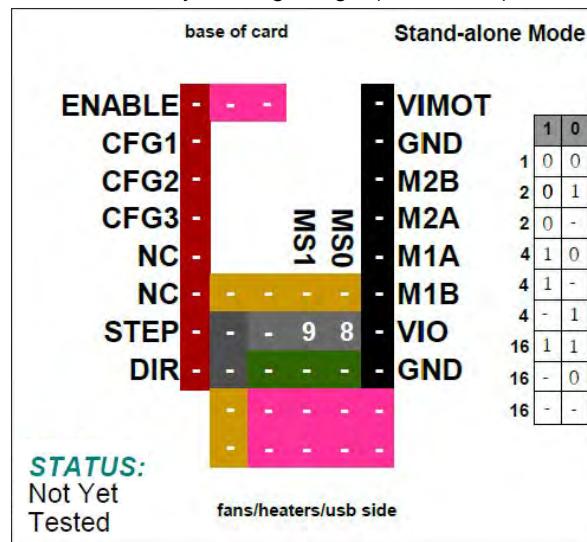
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2100 Drivers in Stand-alone Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



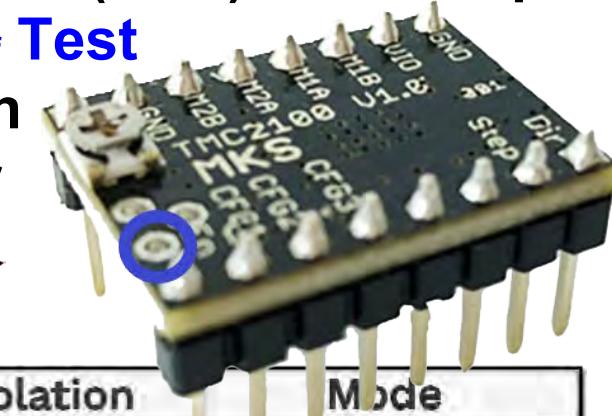
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



MKS TMC2100

Stand-alone Mode

NOTE: Use the potentiometer (POT) on the top of the board or use the "V_{ref} Test point" location, as shown in BLUE, to adjust your driver board's V_{ref}



Driver Chip	MS1	MS0	Steps	Interpolation	Mode
Makerbase TMC2100 Stand Alone 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

Driving Current Calculation Formula
 R_S (Typical Sense Resistor) = 0.11Ω

$$I_{MAX} = V_{ref}$$

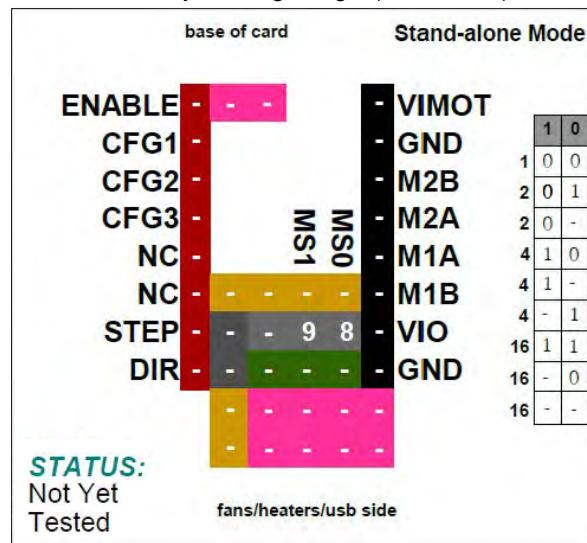
See Appendix B #1. Use 50% to 90% as shown below:

$$I_{MAX} = I_{MAX} * 0.90$$

$$V_{ref} = I_{MAX}$$

See Appendix B #1. Use 50% to 90% as shown below:

$$V_{ref} = V_{ref} * 0.90$$



MKS TMC2100

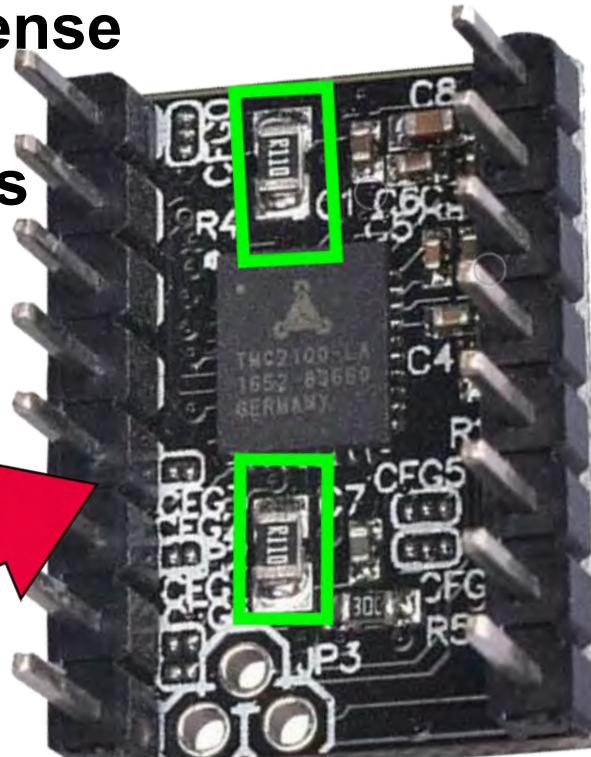
Stand-alone Mode

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board (or use the board's " V_{ref} Test point" to set your V_{ref} . See [**Appendix A**](#) for instructions on how to set the V_{ref} 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: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



$R_s = R050$ is 0.05 Ohms

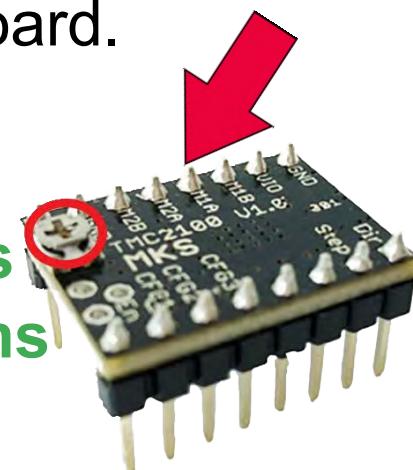
$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

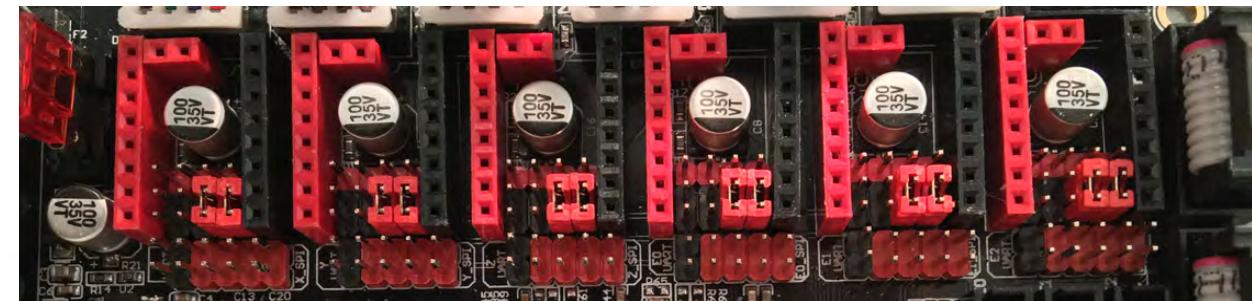
$R_s = R220$ is 0.22 Ohms



Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

MKS TMC2100**Stand-alone Mode**

Stand-alone Mode	ENABLE	-	-	VIMOT
	CFG1	-	-	-
	CFG2	-	-	GND
	CFG3	-	-	M2B
	NC	-	MS1	M2A
	NC	-	MS0	M1A
Interpolation: none	STEP	-	9 8	M1B
SpreadCycle	DIR	-	9 8	VIO
		-	-	GND

Stand-alone Mode

Stand-alone Mode

STEPInterpolation: **none**

SpreadCycle

Stand-alone Mode	ENABLE	-	-	VIMOT
	CFG1	-	-	-
	CFG2	-	-	GND
	CFG3	-	-	M2B
	NC	-	MS1	M2A
	NC	-	MS0	M1A
Interpolation: none	STEP	-	9 8	M1B
SpreadCycle	DIR	-	9 -	VIO
		-	-	GND

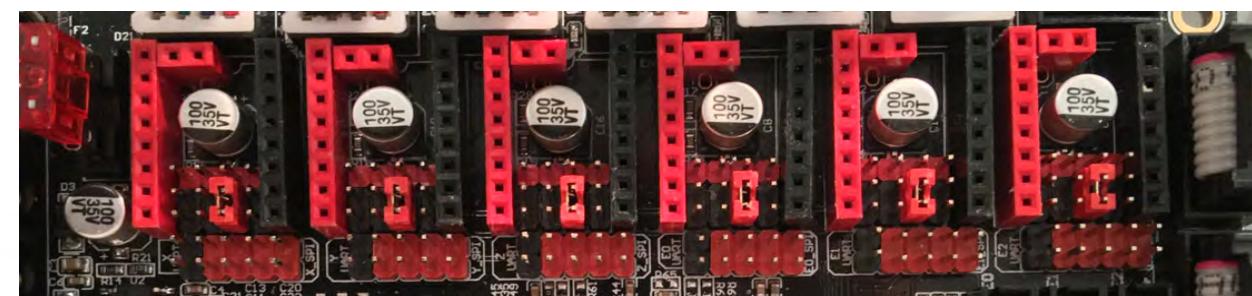


Stand-alone Mode

1 / 2Interpolation: **none**

SpreadCycle

Stand-alone Mode	ENABLE	-	-	VIMOT
	CFG1	-	-	-
	CFG2	-	-	GND
	CFG3	-	-	M2B
	NC	-	MS1	M2A
	NC	-	MS0	M1A
Interpolation: 1/256	STEP	-	9 -	M1B
SpreadCycle	DIR	-	9 -	VIO
		-	-	GND



MKS TMC2100**Stand-alone Mode****Stand-alone Mode**

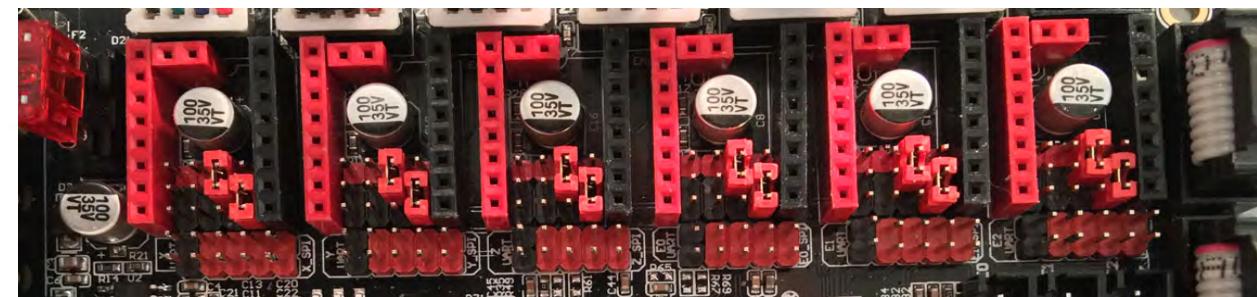
Stand-alone Mode

1 / 4

Interpolation: **none**

SpreadCycle

	ENABLE		VIMOT
Stand-alone Mode	CFG1		GND
	CFG2		M2B
	CFG3	MS0 MS1	M2A
	NC		M1A
	NC	9	M1B
	STEP	9 8	VIO
	DIR	8	GND



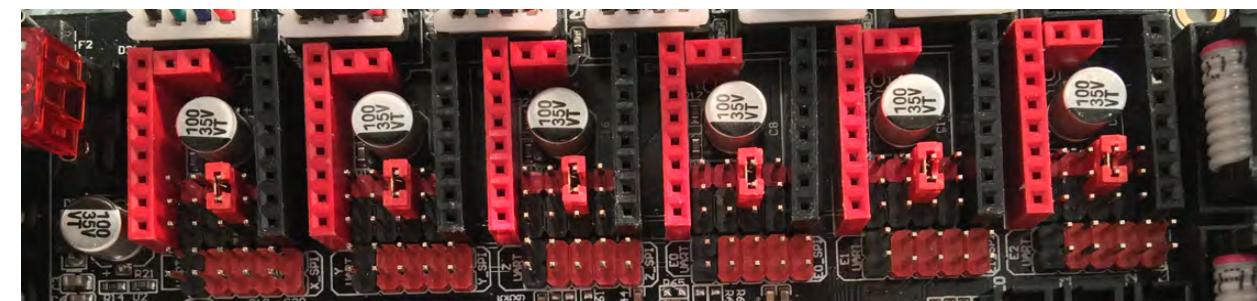
Stand-alone Mode

1 / 4

Interpolation: **1/256**

SpreadCycle

	ENABLE		VIMOT
Stand-alone Mode	CFG1		GND
	CFG2		M2B
	CFG3	MS0 MS1	M2A
	NC		M1A
	NC	9	M1B
	STEP	9	VIO
	DIR		GND



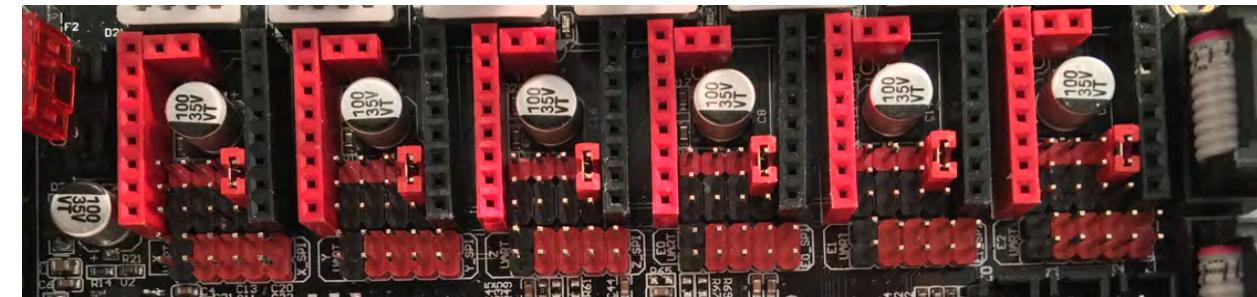
Stand-alone Mode

1 / 4

Interpolation: **1/256**

StealthChop

	ENABLE		VIMOT
Stand-alone Mode	CFG1		GND
	CFG2		M2B
	CFG3	MS0 MS1	M2A
	NC		M1A
	NC	8	M1B
	STEP	8	VIO
	DIR		GND



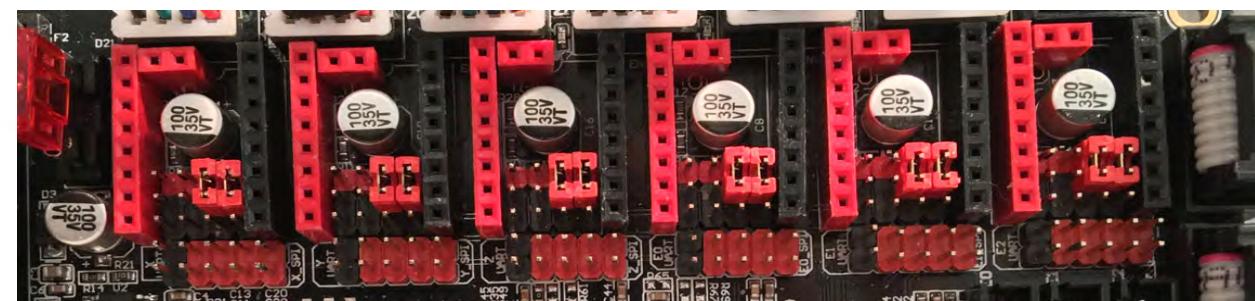
MKS TMC2100**Stand-alone Mode**

Stand-alone Mode

1 / 16Interpolation: **none**

SpreadCycle

ENABLE	[]	[]	-	VIMOT
CFG1	[]	[]	-	GND
CFG2	[]	[]	-	M2B
CFG3	[]	[]	-	M2A
NC	[]	MS1	MS0	M1A
NC	[]	9	8	M1B
STEP	[]	9	8	VIO
DIR	[]	[]	[]	GND

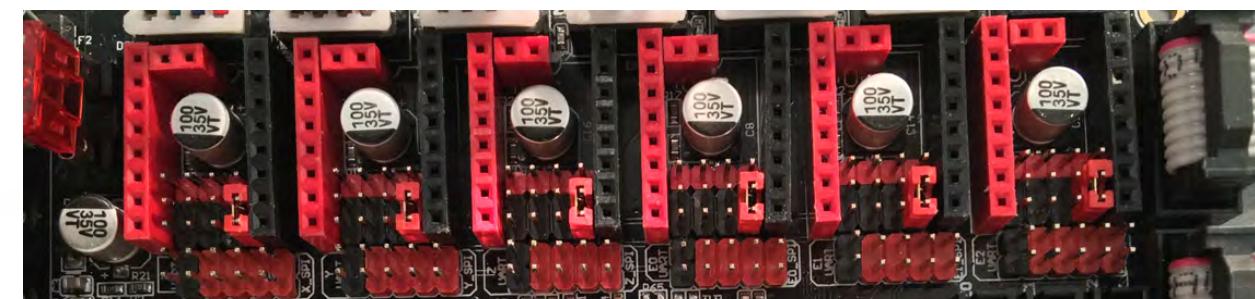


Stand-alone Mode

1 / 16Interpolation: **1/256**

SpreadCycle

ENABLE	[]	[]	-	VIMOT
CFG1	[]	[]	-	GND
CFG2	[]	[]	-	M2B
CFG3	[]	[]	-	M2A
NC	[]	MS1	MS0	M1A
NC	[]	[]	[]	M1B
STEP	[]	[]	8	VIO
DIR	[]	[]	8	GND

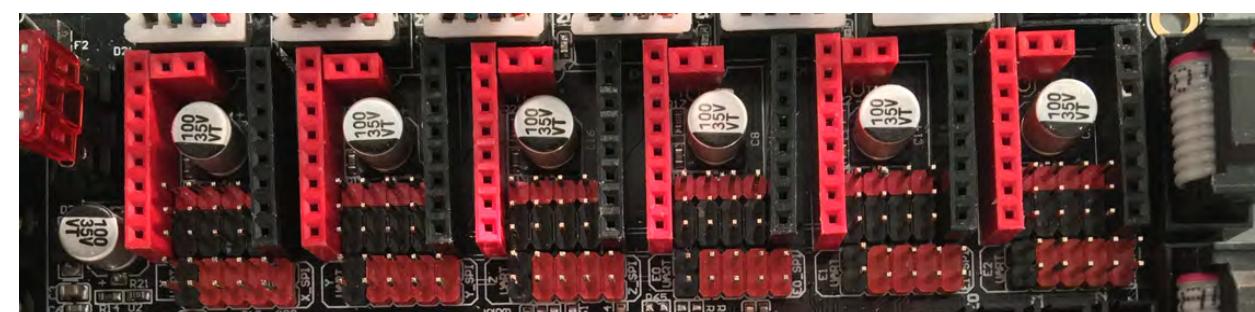


Stand-alone Mode

1 / 16Interpolation: **1/256**

StealthChop

ENABLE	[]	[]	-	VIMOT
CFG1	[]	[]	-	GND
CFG2	[]	[]	-	M2B
CFG3	[]	[]	-	M2A
NC	[]	MS1	MS0	M1A
NC	[]	[]	[]	M1B
STEP	[]	[]	[]	VIO
DIR	[]	[]	[]	GND

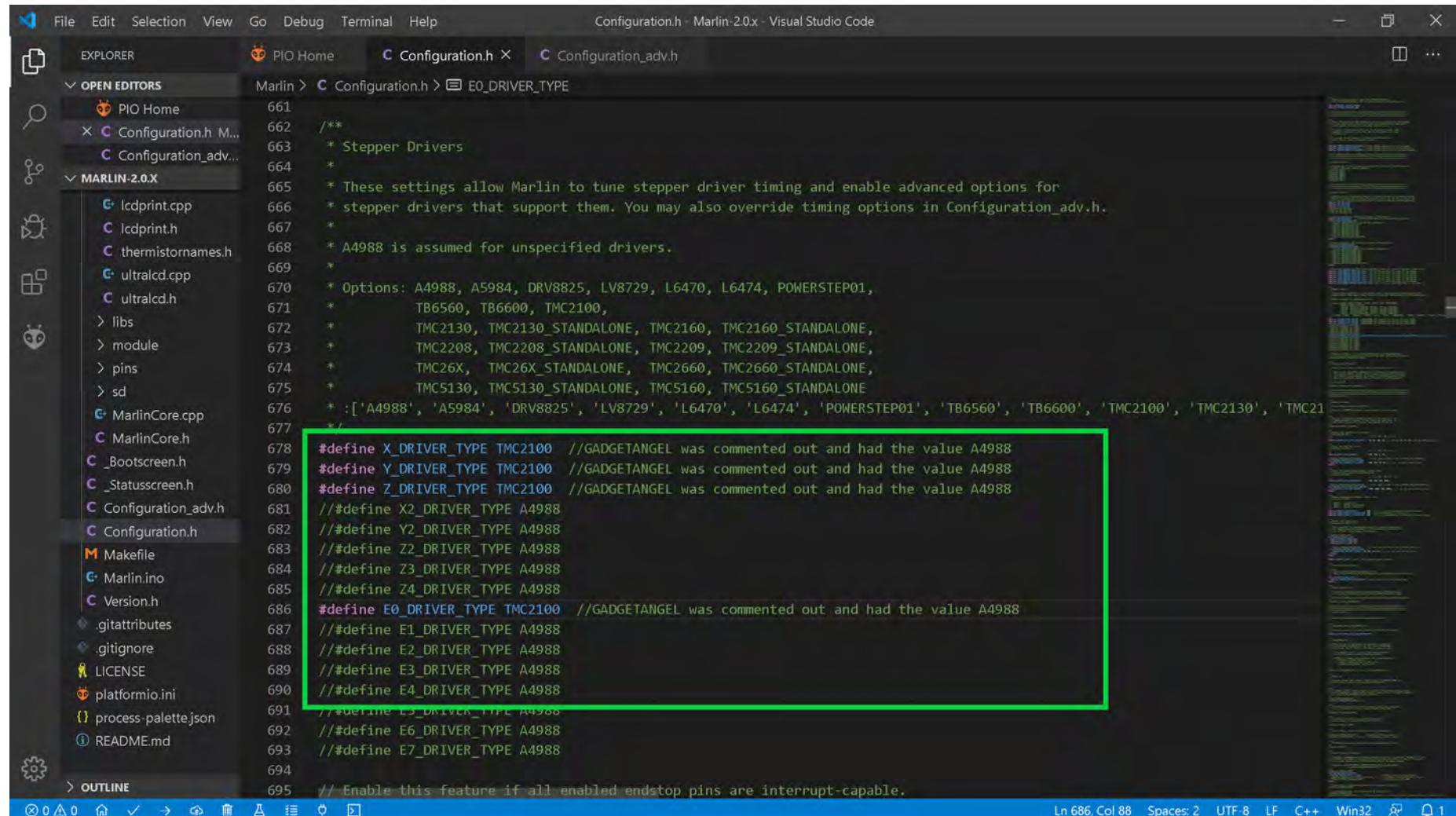


The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for 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 ("//").



```

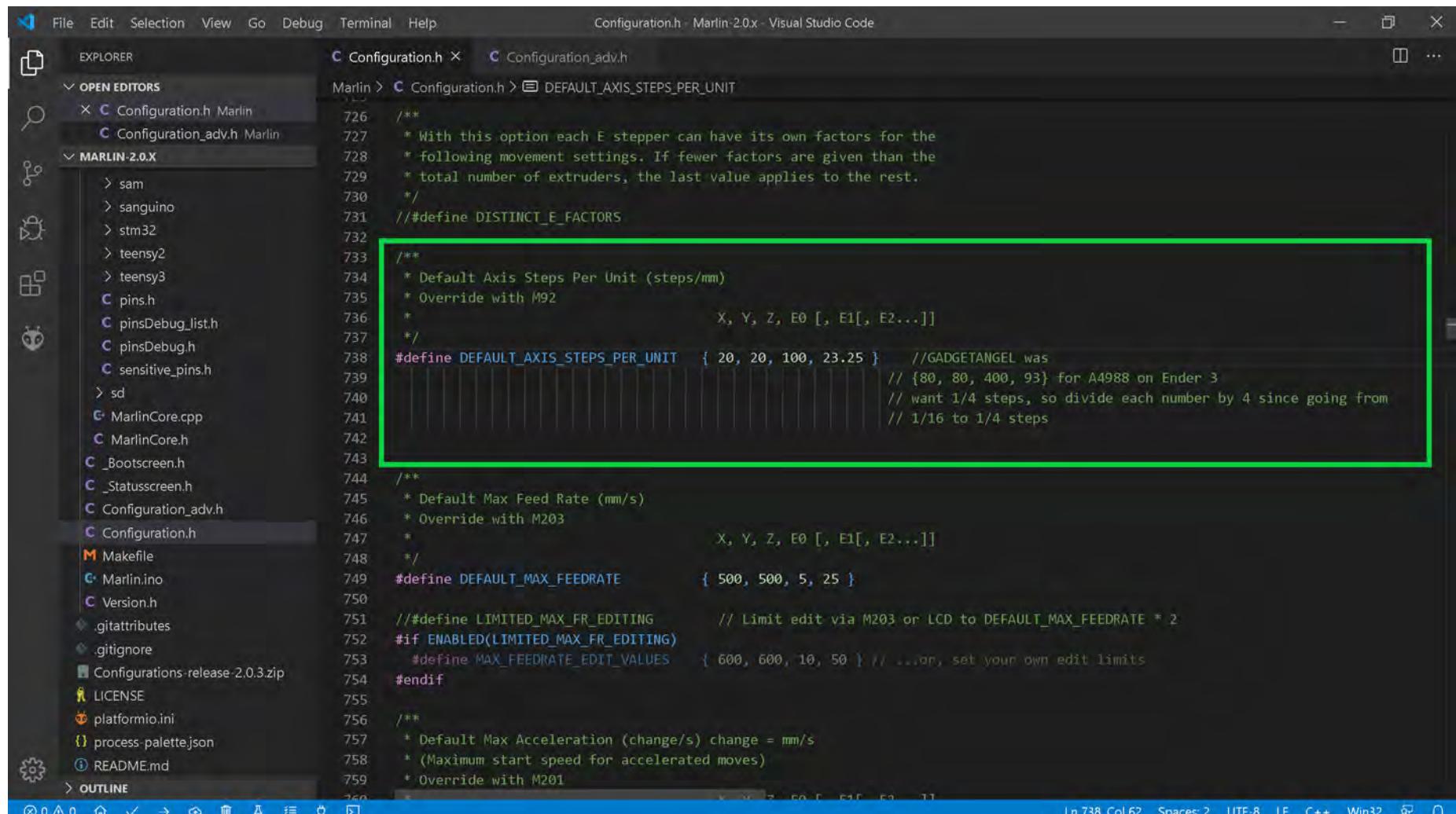
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
  PIO Home
  Configuration.h ...
  Configuration_adv...
MARLIN-2.0.X
  LCDprint.cpp
  LCDprint.h
  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
Ln 686, Col 88 Spaces: 2 UTF-8 LF C++ Win32 1
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 * A4988 is assumed for unspecified drivers.
667 *
668 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
669 *           TB6560, TB6600, TMC2100,
670 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
671 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
672 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
673 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
674 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2660', 'TMC5160']
675 */
676 #define X_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
677 #define Y_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
678 #define Z_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
679 //#define X2_DRIVER_TYPE A4988
680 //#define Y2_DRIVER_TYPE A4988
681 //#define Z2_DRIVER_TYPE A4988
682 //#define Z3_DRIVER_TYPE A4988
683 //#define Z4_DRIVER_TYPE A4988
684 #define E0_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
685 //#define E1_DRIVER_TYPE A4988
686 //#define E2_DRIVER_TYPE A4988
687 //#define E3_DRIVER_TYPE A4988
688 //#define E4_DRIVER_TYPE A4988
689 //#define E5_DRIVER_TYPE A4988
690 //#define E6_DRIVER_TYPE A4988
691 //#define E7_DRIVER_TYPE A4988
692 // Enable this feature if all enabled endstop pins are interrupt-capable.
693

```

- Go to the next page.

The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to MKS TMC2100 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The left sidebar shows the project structure under 'MARLIN-2.0.X'. The main editor window displays the Configuration.h file. A green rectangular box highlights the following code block:

```

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 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742
743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

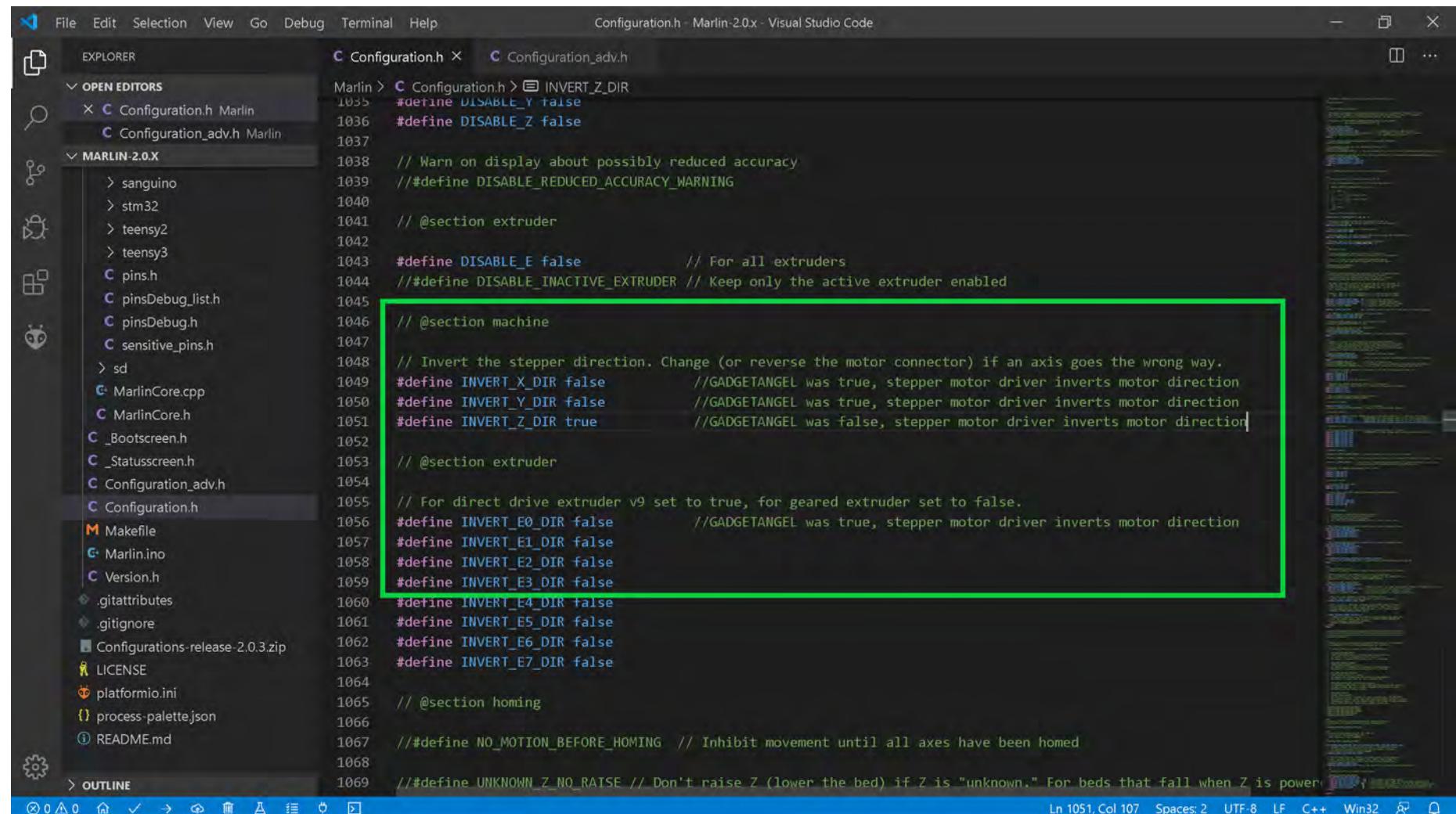
750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
754#endif
755
756 /**
757 * Default Max Acceleration (change/s) change = mm/s
758 * (Maximum start speed for accelerated moves)
759 * Override with M201
760 */

```

- Go to the next page.

The (latest release of) Marlin Setup for 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



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 > Configuration.h > INVERT_Z_DIR
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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that the setting for the Z-axis stepper motor direction has been changed from its original value.

- Go to the next page.

The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode

- The end of Marlin setup for 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.

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

EXPLORER

OPEN EDITORS

MARLIN-2.0.X

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, PONERSTEP01,  
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 OUTPUT DEBUG CONSOLE TERMINAL

1: Task - Build

TMC2209_STANDALONE	IGNORED	
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
TMC2660_STANDALONE	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	

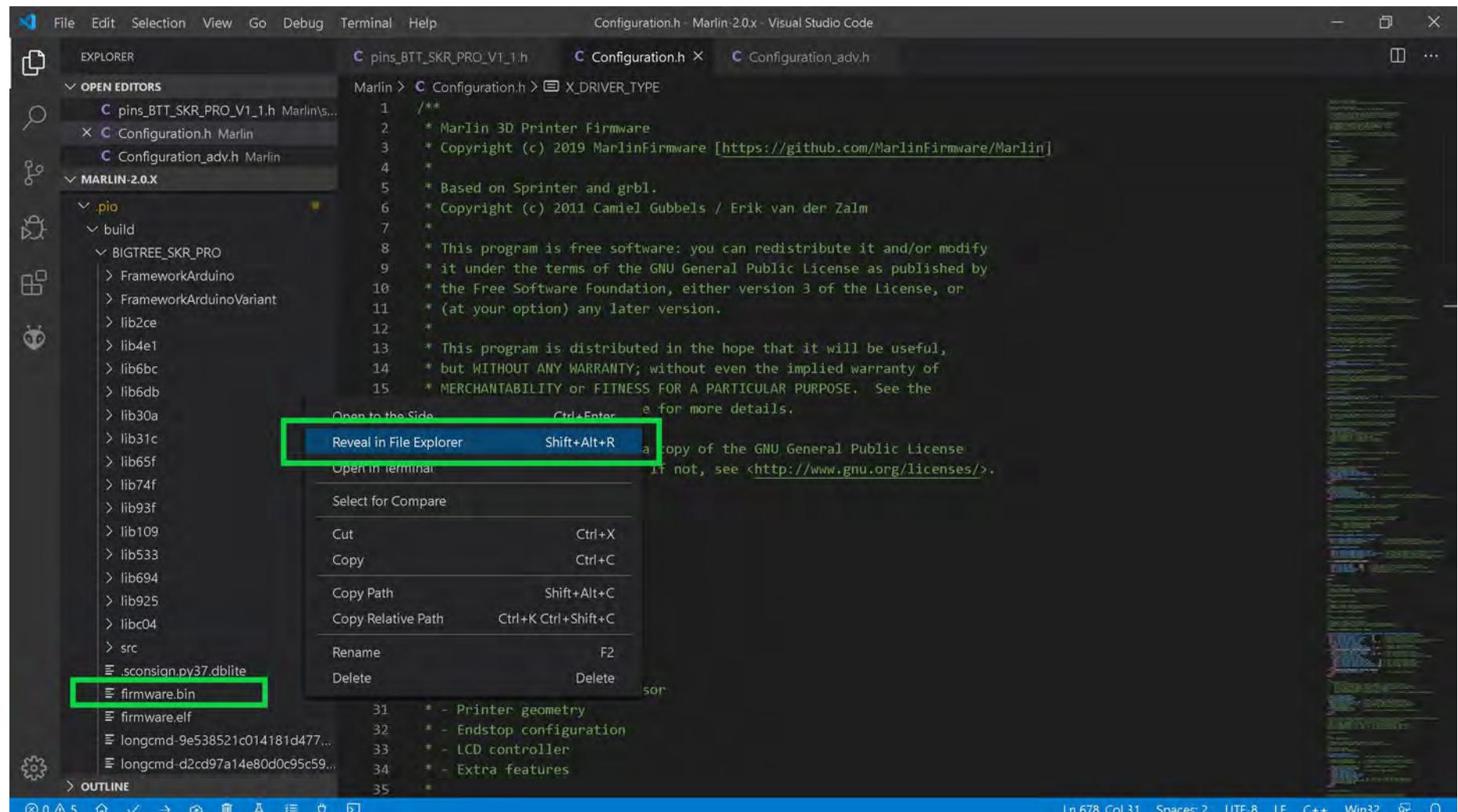
===== 1 succeeded in 00:02:31.294 =====

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

- 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 MKS TMC2100 Drivers in Stand-alone Mode

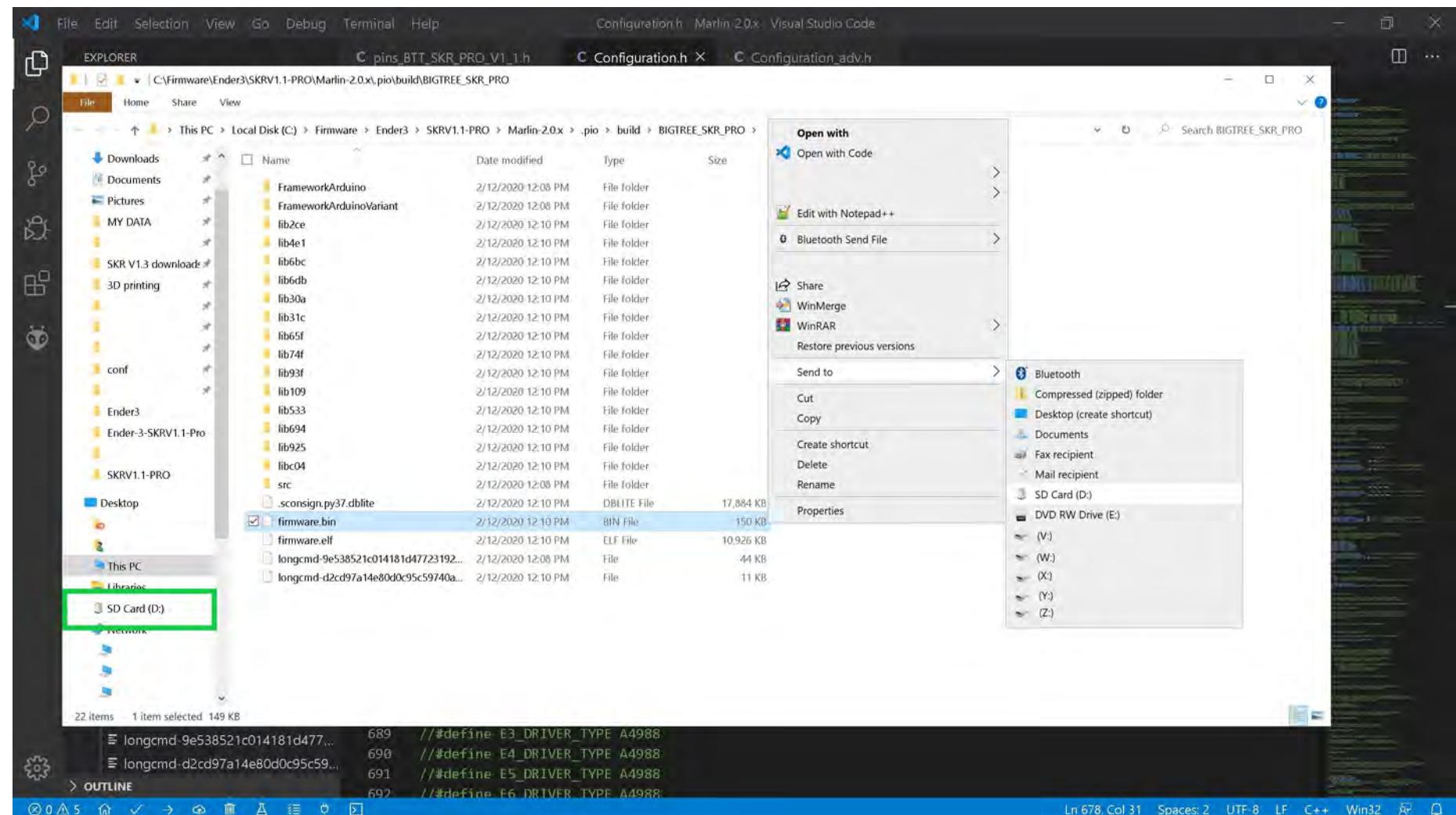
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



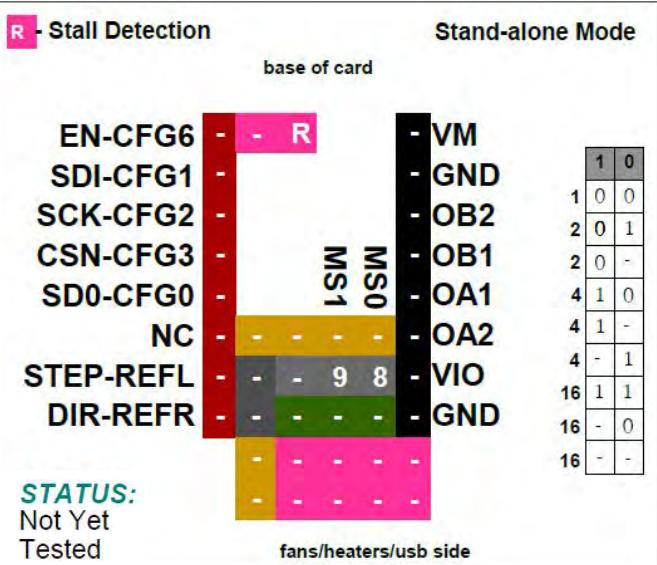
- Go to the next page.

The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".

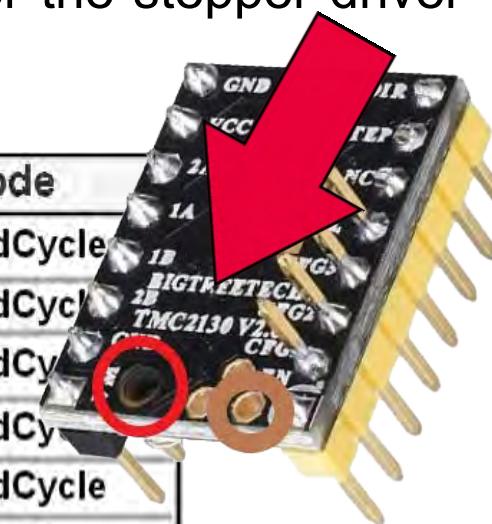


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

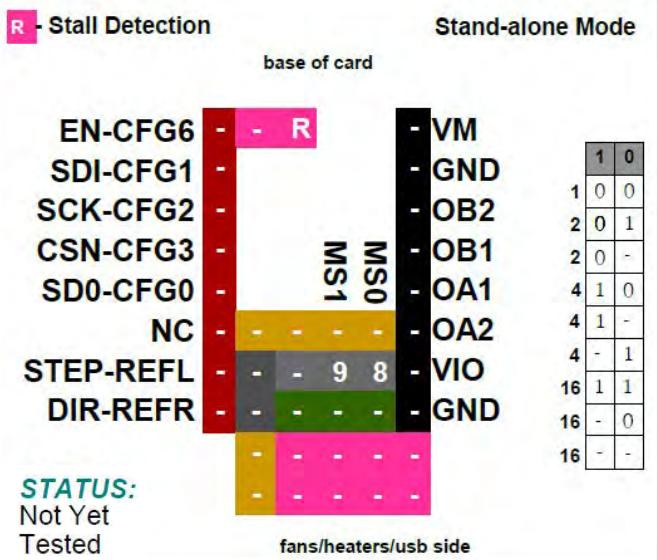
**BIQU TMC2130**Stand-alone Mode

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} , as shown in **RED**; or use the " V_{ref} Test point" location on the top of the driver board, as shown in **BROWN**. See **Appendix A** for instructions on how to set the V_{ref} for the stepper driver board.



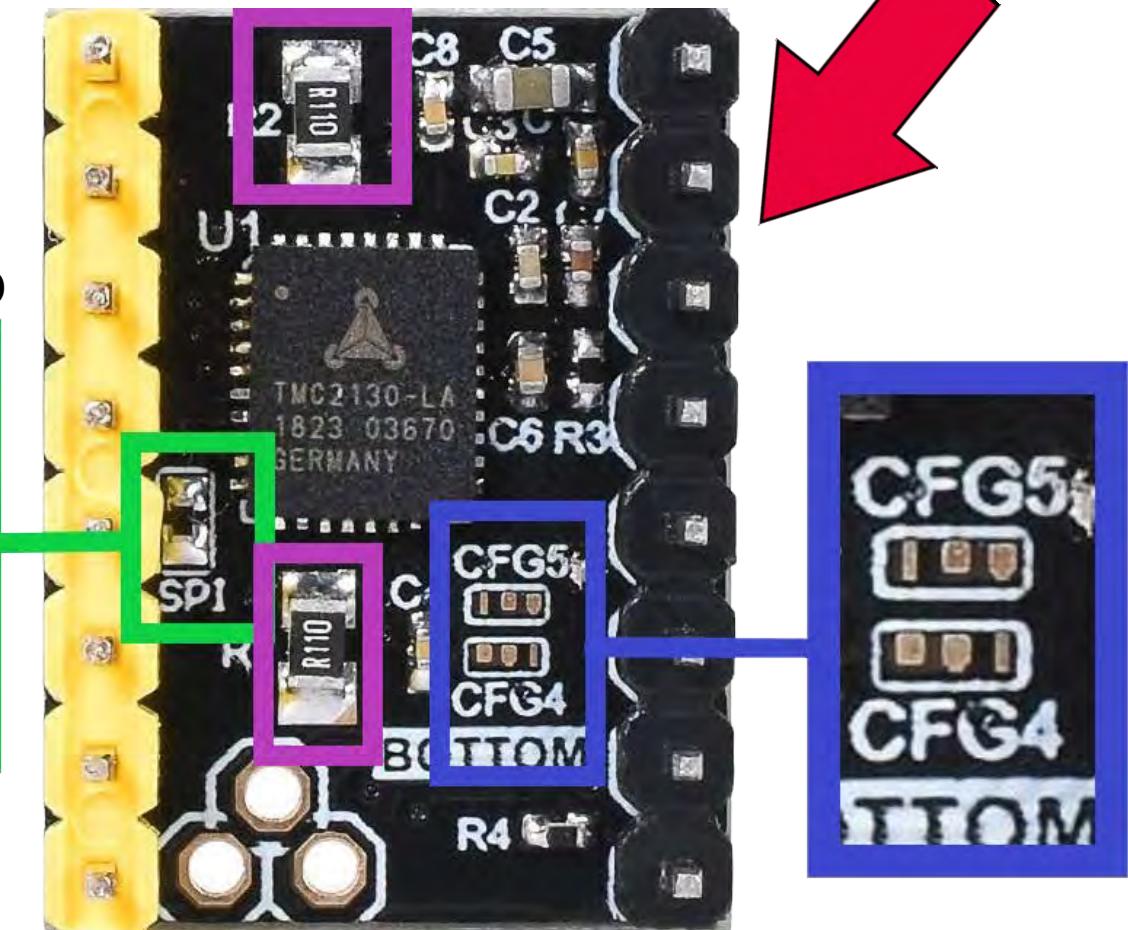
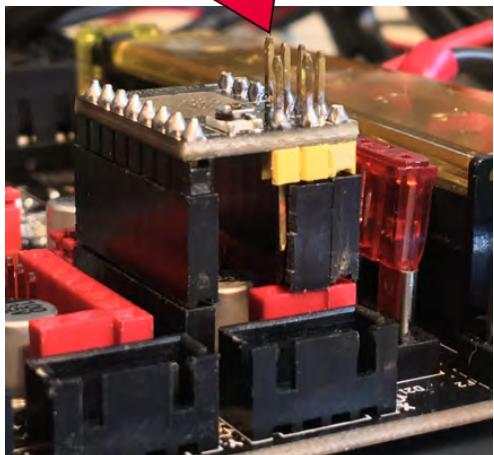
Driver Chip	MS1	MS0	Steps	Interpolation	Mode
Biqu® TMC2130 Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	GND	GND	1	NONE	spreadCycle
	GND	VIO	1 / 2	NONE	spreadCycle
	GND	OPEN	1 / 2	1 / 256	spreadCycle
	VIO	GND	1 / 4	NONE	spreadCycle
	VIO	OPEN	1 / 4	1 / 256	spreadCycle
	OPEN	VIO	1 / 4	1 / 256	stealthChop
	VIO	VIO	1 / 16	NONE	spreadCycle
	OPEN	GND	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.11Ω	$I_{MAX} = V_{ref}$			$V_{ref} = I_{MAX}$	
	See Appendix B #2. Use 50% to 90% as shown below:			See Appendix B #2. Use 50% to 90% as shown below:	
	$I_{MAX} = I_{MAX} * 0.90$			$V_{ref} = V_{ref} * 0.90$	

BIQU TMC2130Stand-alone Mode

To place the BIQU TMC2130 into Stand-alone mode:

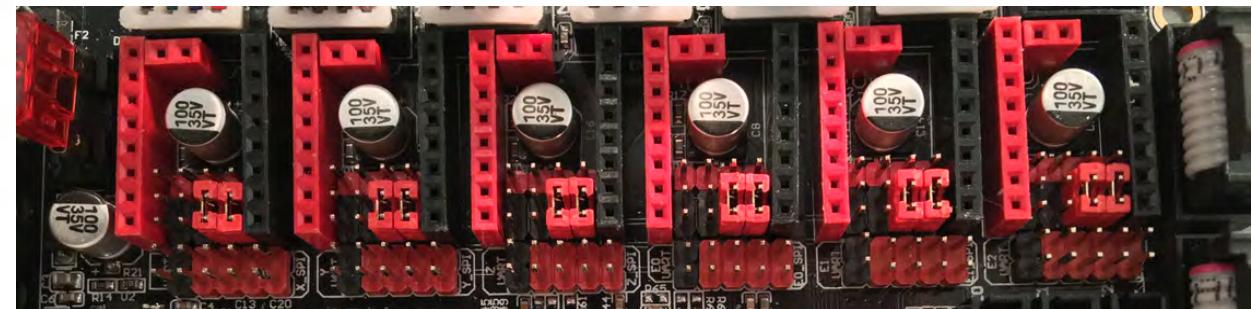
Solder the SPI Jumper together, on the bottom of the driver board to the adjacent pad, as shown in the **GREEN** box below. Ensure at CFG4 location and CFG5 location that those pads are NOT soldered together to form a bridge, as shown in **BLUE**. The **PURPLE** box shows the location of the current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

Note: When the stall-guard function is **not used**, the stall-guard pin ("R") 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.

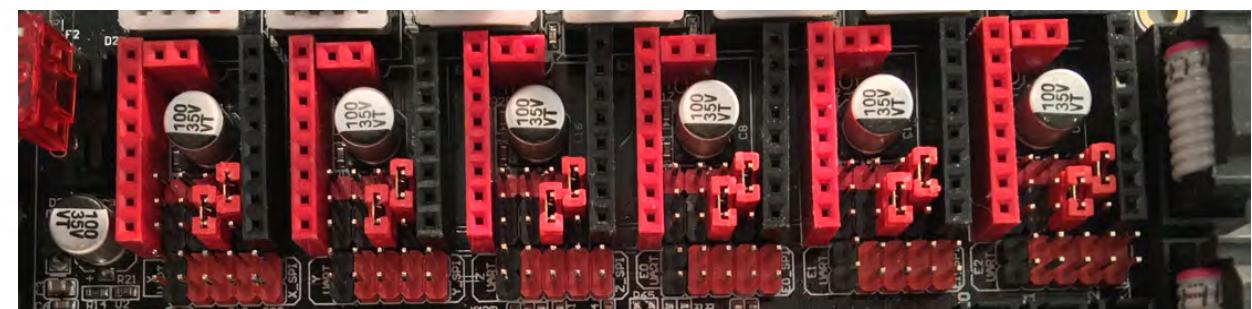


Stand-alone Mode

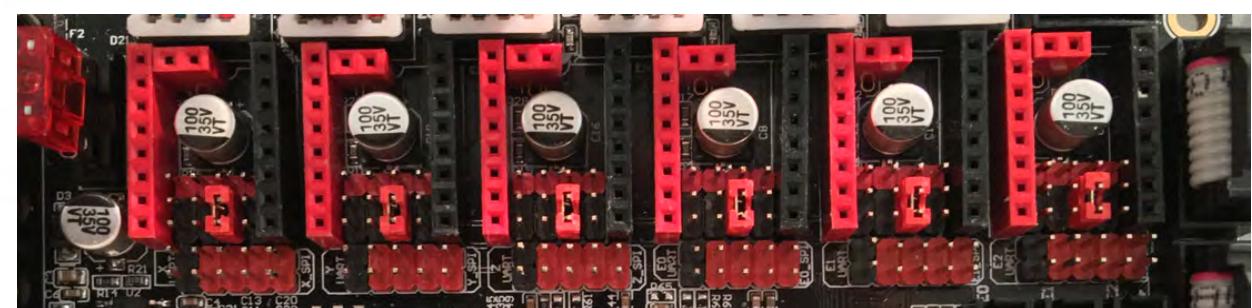
Stand-alone Mode	EN-CFG6	-	R	-	VM
STEP	SDI-CFG1	-		-	GND
	SCK-CFG2	-		-	OB2
	CSN-CFG3	-		-	OB1
Interpolation: none	SD0-CFG0	-	MS1	-	OA1
	NC	-	MS0	-	OA2
SpreadCycle	STEP-REFL	-	9	8	VIO
	DIR-REFR	-	9	8	GND



Stand-alone Mode	EN-CFG6	-	R	-	VM
1 / 2	SDI-CFG1	-		-	GND
	SCK-CFG2	-		-	OB2
	CSN-CFG3	-		-	OB1
Interpolation: none	SD0-CFG0	-	MS1	-	OA1
	NC	-	MS0	-	OA2
SpreadCycle	STEP-REFL	-	9	8	VIO
	DIR-REFR	-	9	-	GND

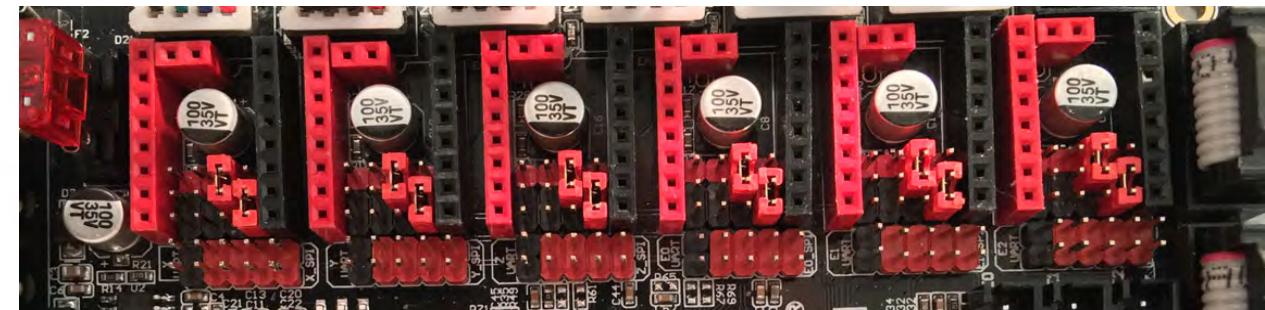


Stand-alone Mode	EN-CFG6	-	R	-	VM
1 / 2	SDI-CFG1	-		-	GND
	SCK-CFG2	-		-	OB2
	CSN-CFG3	-		-	OB1
Interpolation: 1/256	SD0-CFG0	-	MS1	-	OA1
	NC	-	MS0	-	OA2
SpreadCycle	STEP-REFL	-	9	-	VIO
	DIR-REFR	-	9	-	GND

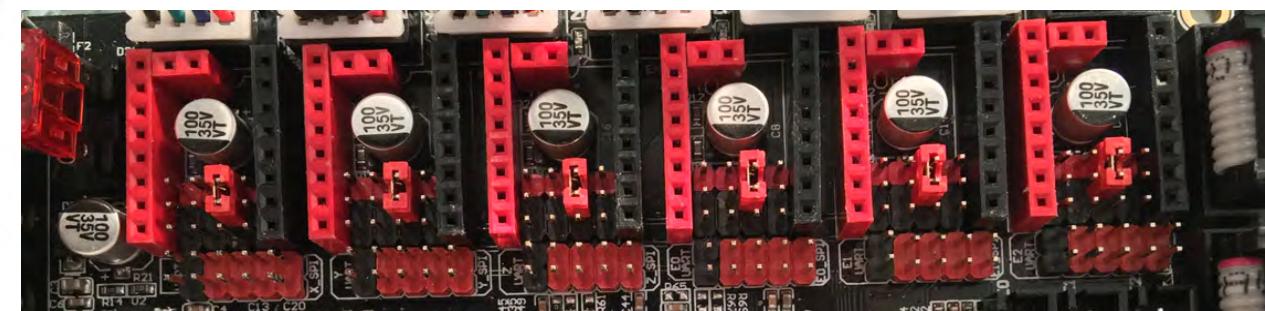


Stand-alone Mode

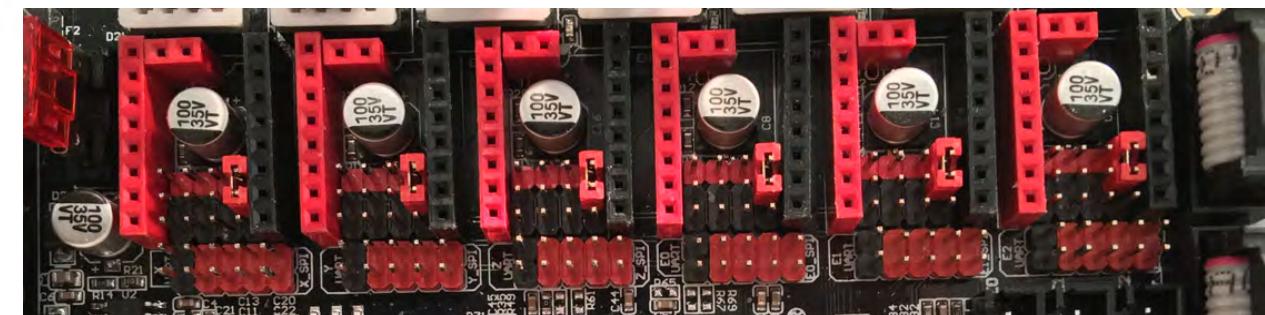
Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: none	SD0-CFG0	MS1 MS0	OA1
	NC	9	OA2
SpreadCycle	STEP-REFL	9 8	VIO
	DIR-REFR	8	GND



Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: 1/256	SD0-CFG0	MS1 MS0	OA1
	NC	9	OA2
SpreadCycle	STEP-REFL	9	VIO
	DIR-REFR	-	GND

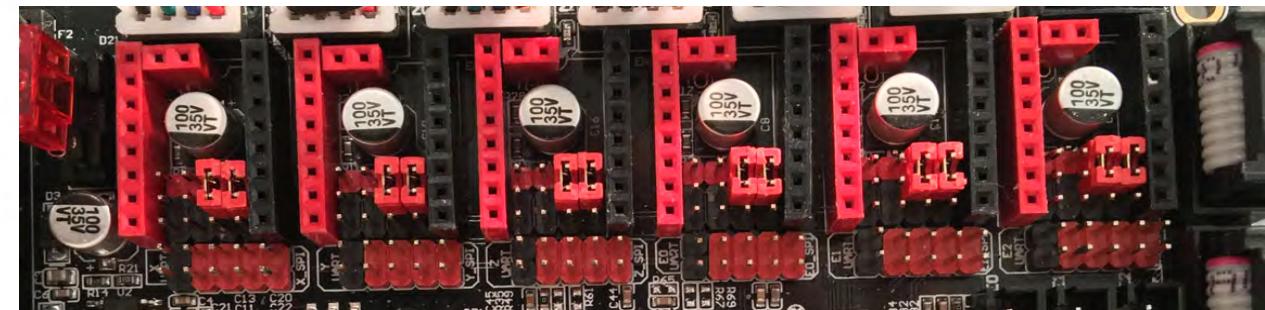


Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: 1/256	SD0-CFG0	MS1 MS0	OA1
	NC	- - 8	OA2
StealthChop	STEP-REFL	- - 8	VIO
	DIR-REFR	- - -	GND



Stand-alone Mode

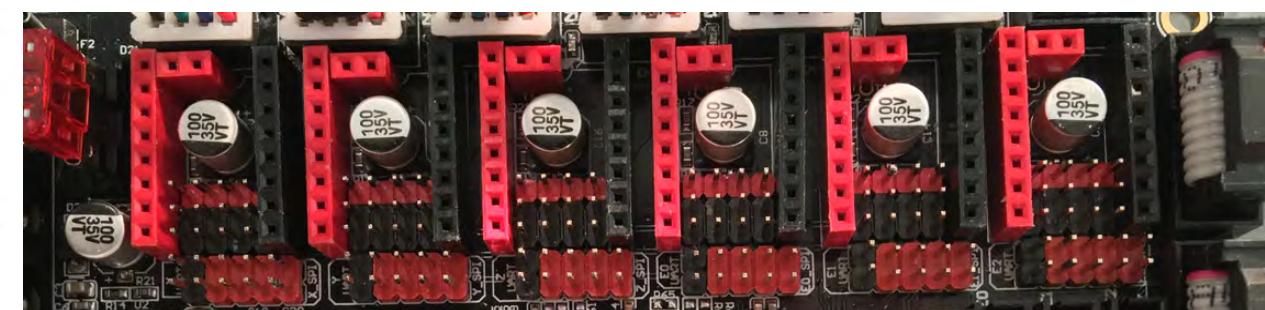
Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: none	SD0-CFG0	MS0	OA1
	NC	MS1	OA2
SpreadCycle	STEP-REFL	9 8	VIO
	DIR-REFR	9 8	GND



Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: 1/256	SD0-CFG0	MS0	OA1
	NC	MS1	OA2
SpreadCycle	STEP-REFL	8	VIO
	DIR-REFR	8	GND



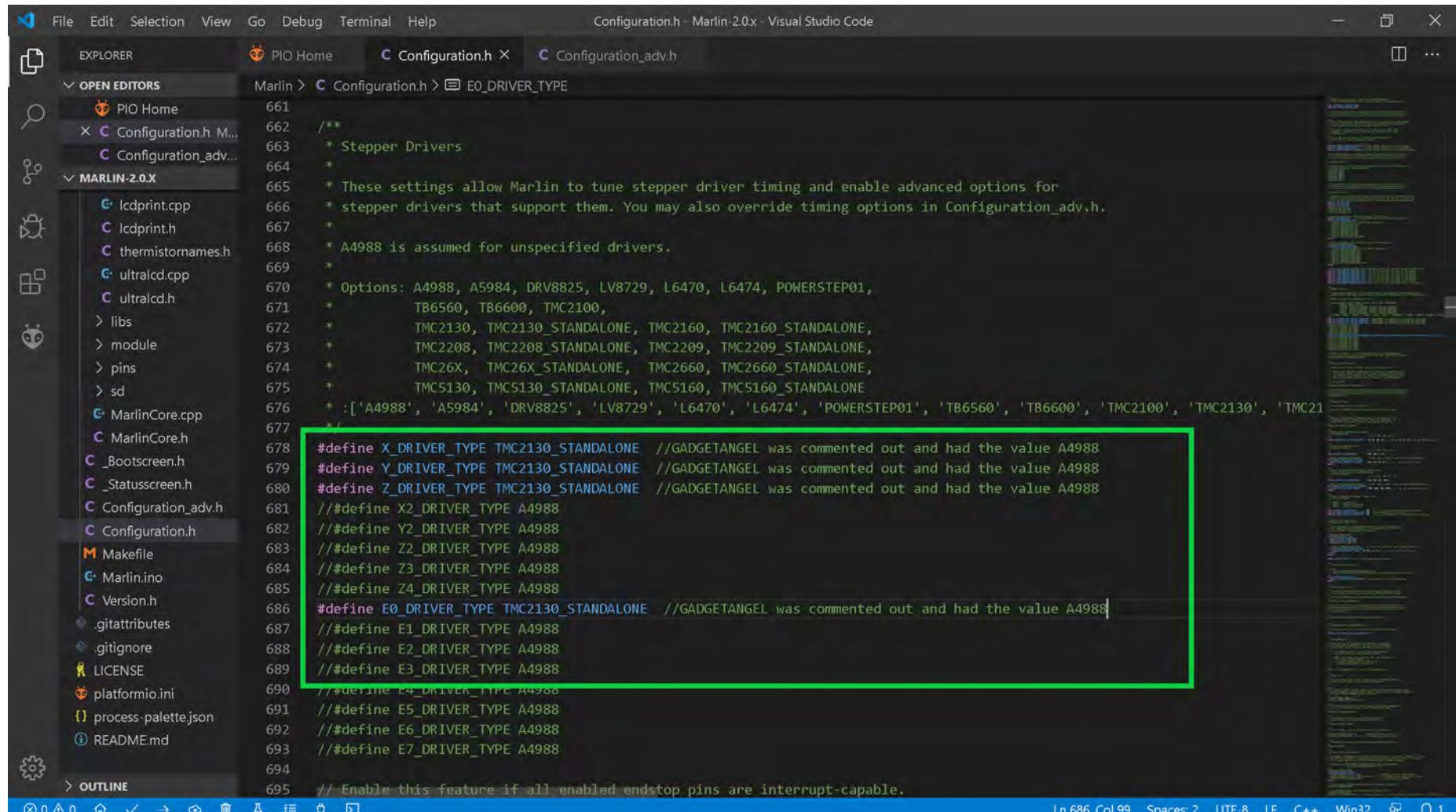
Stand-alone Mode	EN-CFG6	R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
Interpolation: 1/256	SD0-CFG0	MS0	OA1
	NC	MS1	OA2
StealthChop	STEP-REFL	-	VIO
	DIR-REFR	-	GND



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

NOTE: Go to Appendix C, and 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 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 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677
678 #define X_DRIVER_TYPE TMC2130_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2130_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2130_STANDALONE //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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.

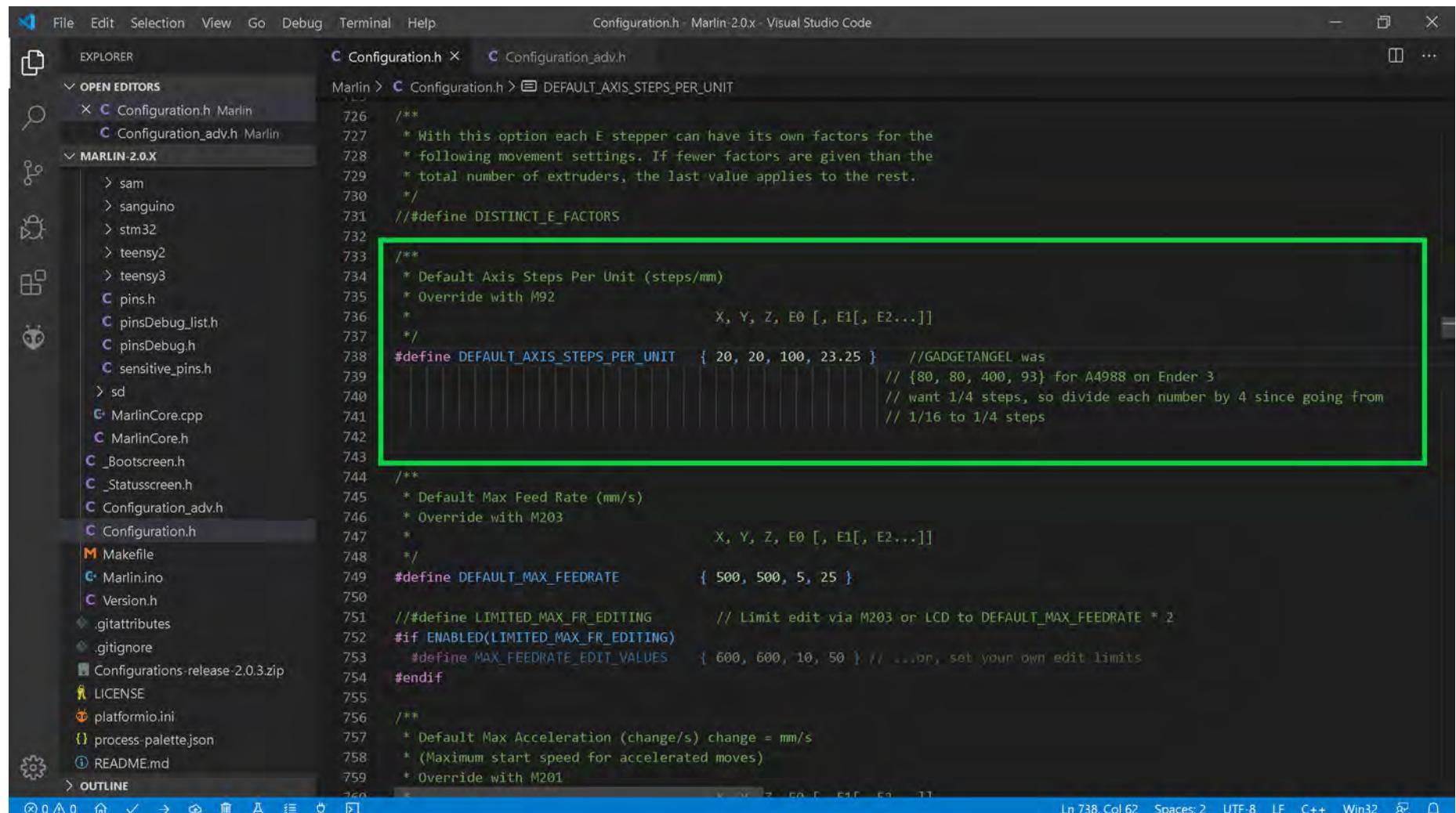
```

A green rectangular box highlights the driver type definitions (lines 678-693), specifically the lines starting with `#define`. These lines were previously commented out with `//` at the start of the line. The rest of the code is standard Marlin configuration for TMC2130 drivers.

- Go to the next page.

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

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to BIQU TMC2130 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, 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 configuration files. A green rectangular box highlights the following code snippet in the 'Configuration.h' file:

```

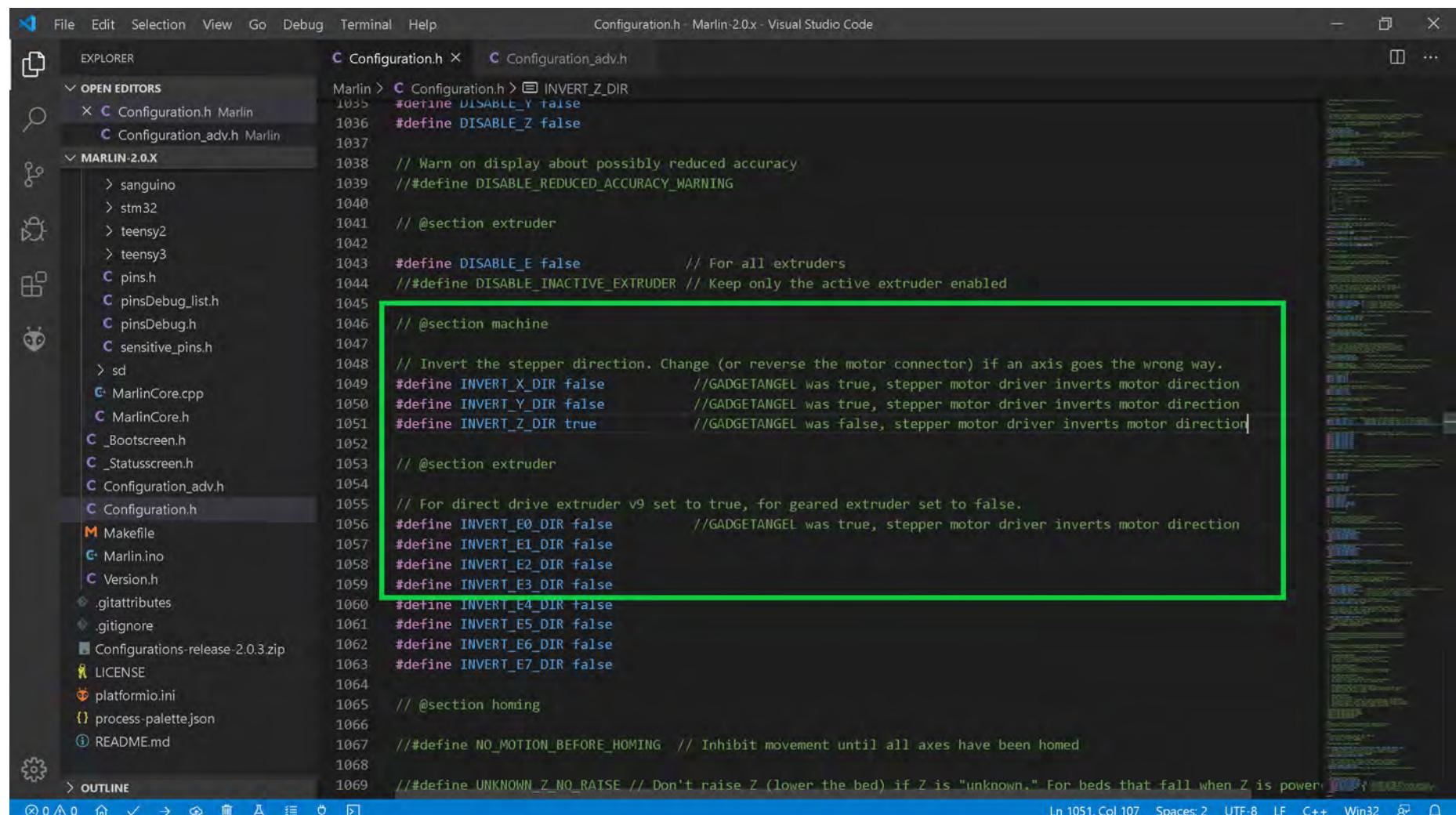
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 *
737 X, Y, Z, E0 [, E1[, E2...]]
738
739 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
740 // {80, 80, 400, 93} for A4988 on Ender 3
741 // want 1/4 steps, so divide each number by 4 since going from
742 // 1/16 to 1/4 steps
743
744 /**
745 * Default Max Feed Rate (mm/s)
746 * Override with M203
747 *
748 X, Y, Z, E0 [, E1[, E2...]]
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
754 #endif
755
756 /**
757 * Default Max Acceleration (change/s) change = mm/s
758 * (Maximum start speed for accelerated moves)
759 * Override with M201

```

- 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



```

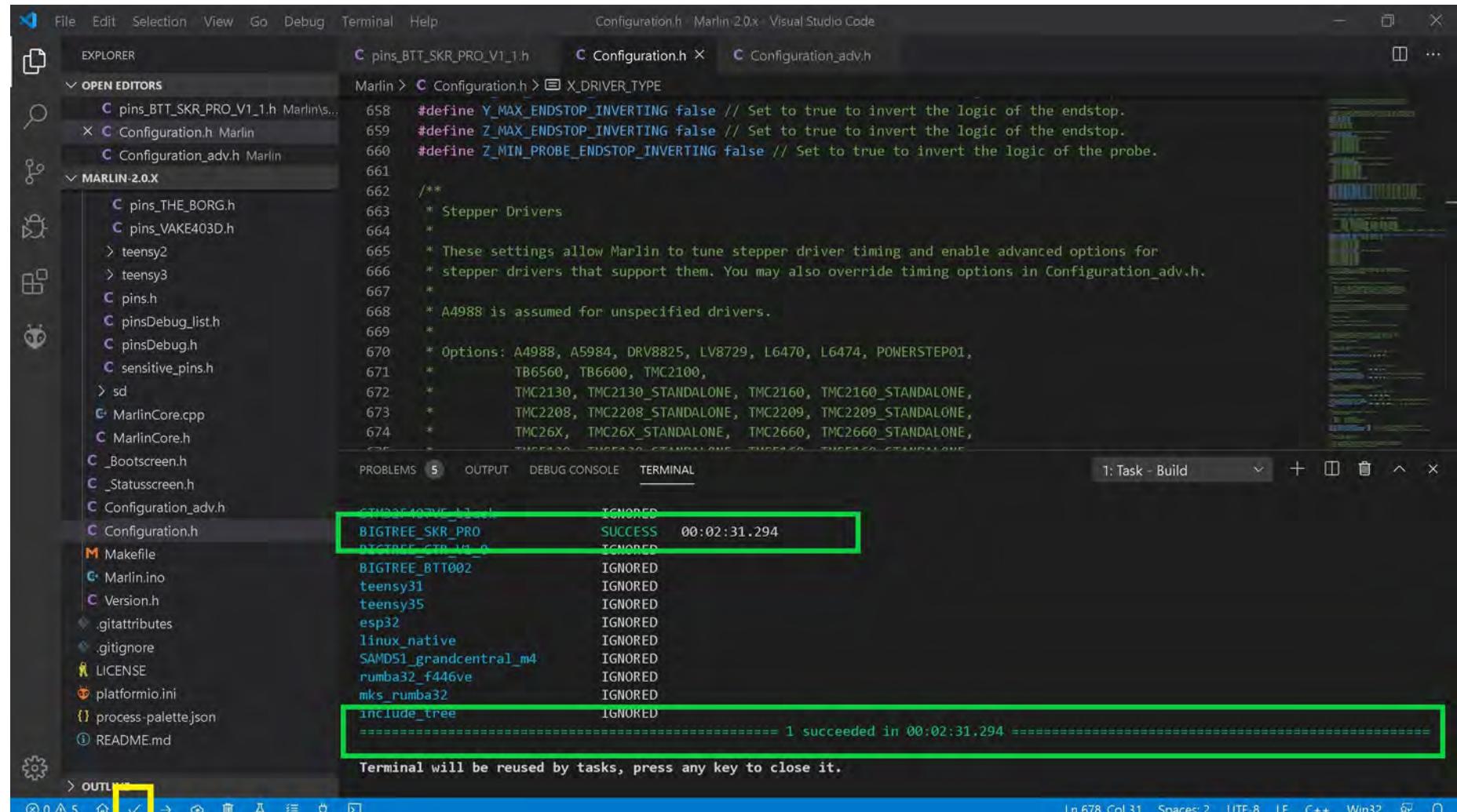
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  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
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palettejson
  README.md
  OUTLINE
  1035 #define DISABLE_Y false
  1036 #define DISABLE_Z false
  1037
  // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  // @section extruder
  1041
  1043 #define DISABLE_E false          // For all extruders
  // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1044
  1046
  // @section machine
  1048
  // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1052
  // @section extruder
  1053
  // For direct drive extruder v9 set to true, for geared extruder set to false.
  1055 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1056 #define INVERT_E1_DIR false
  1057 #define INVERT_E2_DIR false
  1058 #define INVERT_E3_DIR false
  1059 #define INVERT_E4_DIR false
  1060 #define INVERT_E5_DIR false
  1061 #define INVERT_E6_DIR false
  1062 #define INVERT_E7_DIR false
  1063
  // @section homing
  1065 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1066
  1067 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  1068
  1069
  
```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- 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.



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

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Editor Area:** Shows three open files: Configuration.h, pins_BTT_SKR_PRO_V1_1.h, and Configuration_adv.h.
- Left Sidebar (EXPLORER):** Shows the project structure under MARLIN-2.0.X, including pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and various driver and board configuration files like pins_VAKE403D.h, pins.h, pinsDebug_list.h, pinsDebug.h, sensitive_pins.h, and MarlinCore.cpp.
- Terminal Tab:** Shows the build log output.
- Output Tab:** Shows the status of the build tasks.
- Status Bar:** Shows the current file (Configuration.h), line (Ln 678), column (Col 31), spaces (Spaces: 2), encoding (UTF-8), and other system information.

Terminal Output (TMC2130 Standalone Build):

```

TMC2130_STANDALONE
BIGTREE_SKR_PRO          IGNORED
BIGTREE_SKR_PRO           SUCCESS 00:02:31.294
BIGTREE_SKR_PRO           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 =====

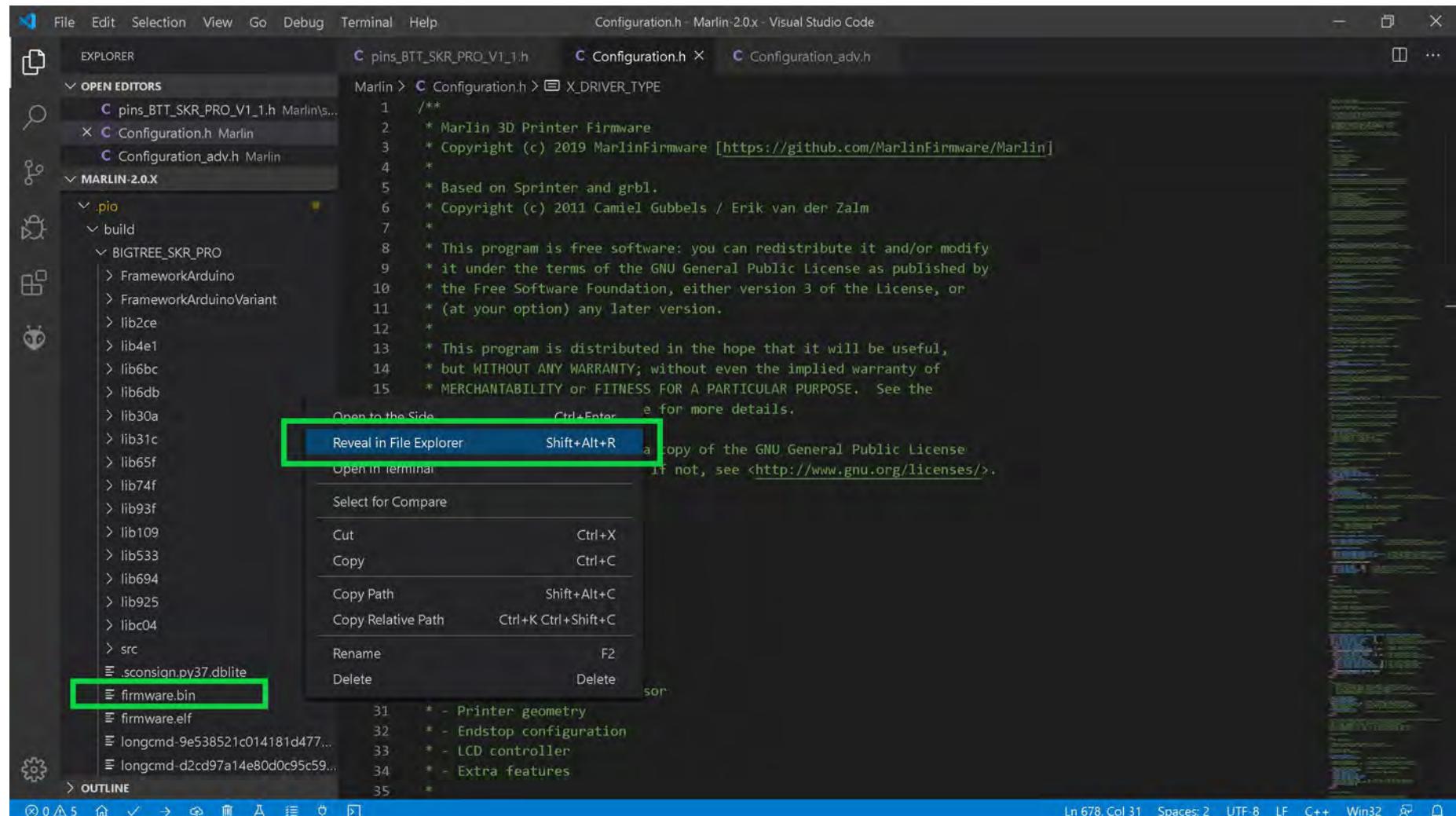
```

Output Tab Status: 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 TMC2130 Drivers in Stand-alone Mode

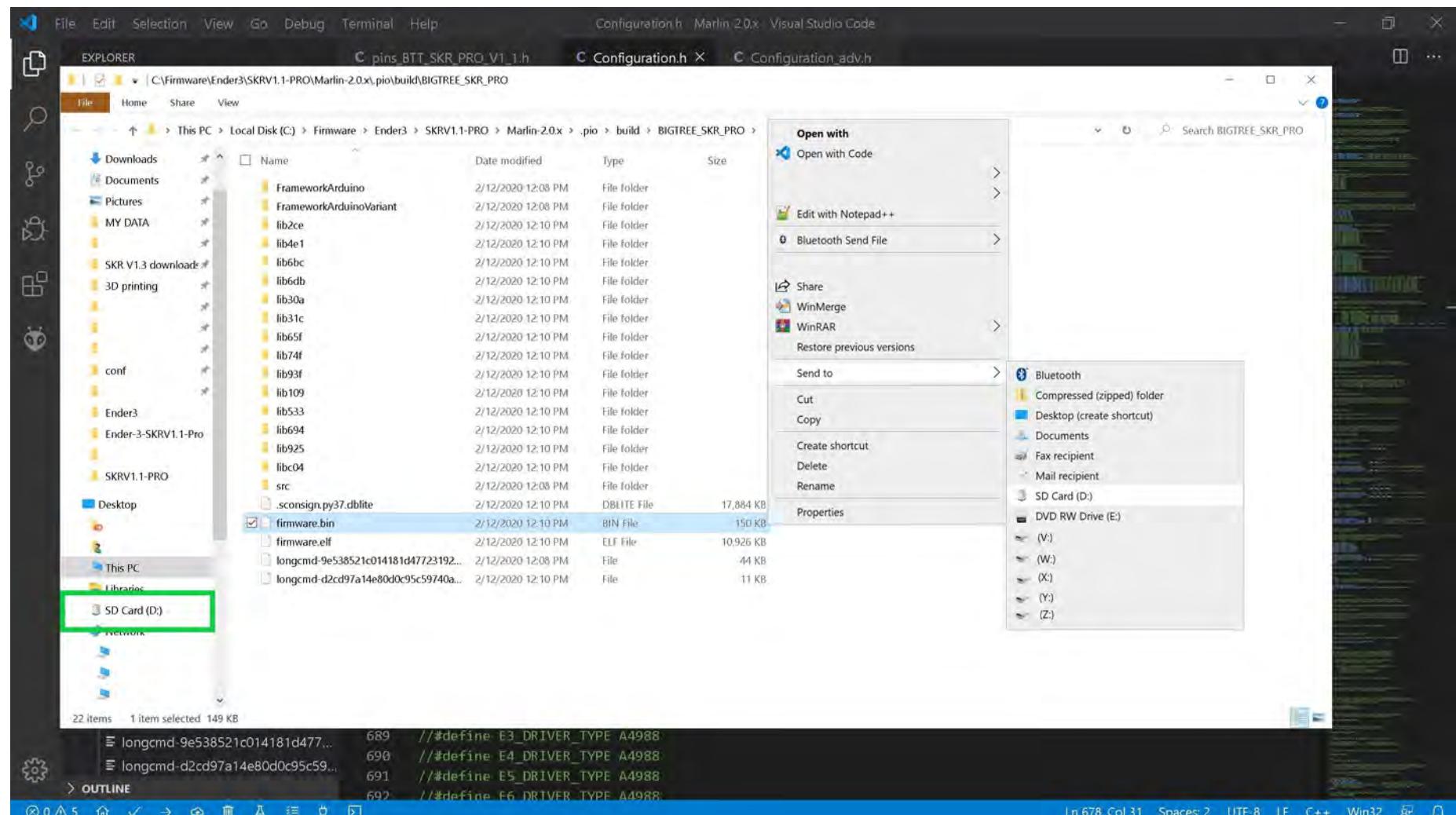
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file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



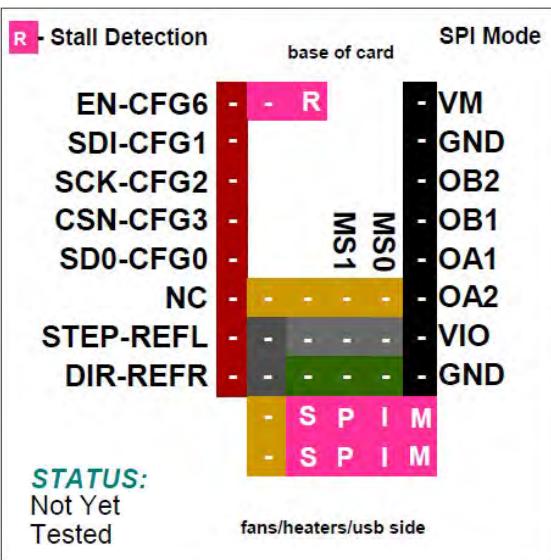
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

**BIQU TMC2130**SPI Mode

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

See the next page for further information.

<p>Driver Chip</p> <p>B<small>BIQU</small>[®]</p> <p>TMC2130</p> <p>SPI Mode</p> <p>Maximum 256 Subdivision</p> <p>46V DC 2.5A (peak)</p>	<p>Steps are set inside of your Firmware</p>
<p>Driving Current Calculation Formula</p> <p>R_S(Typical Sense Resistor)= 0.11Ω</p>	<p>$I_{MAX}=V_{ref}$</p> <p>See Appendix B #2. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$</p> <p>$V_{ref}=I_{MAX}$</p> <p>See Appendix B #2. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$</p>

R - Stall Detection		base of card	SPI Mode
EN-CFG6	R		VM
SDI-CFG1			GND
SCK-CFG2			OB2
CSN-CFG3			OB1
SD0-CFG0		MOSI	OA1
NC		MOSI	OA2
STEP-REFL			VIO
DIR-REFR			GND
		- SPI M	- SPI M
STATUS:			
Not Yet Tested		fans/heaters/usb side	

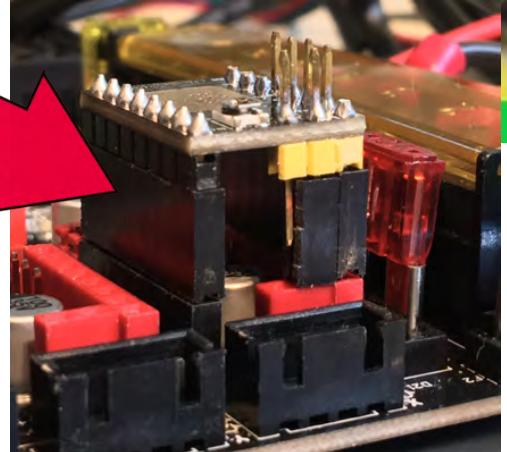
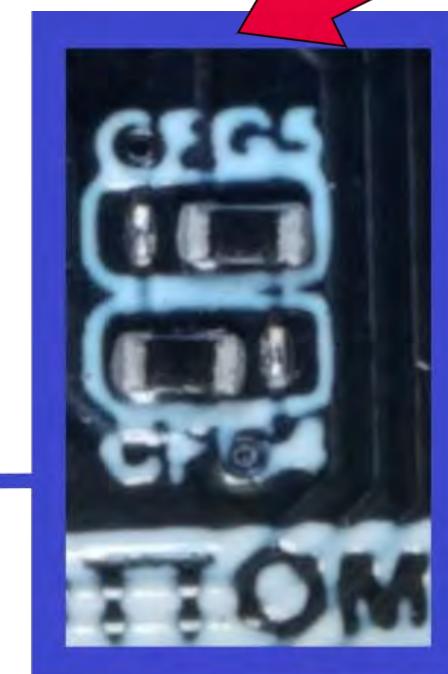
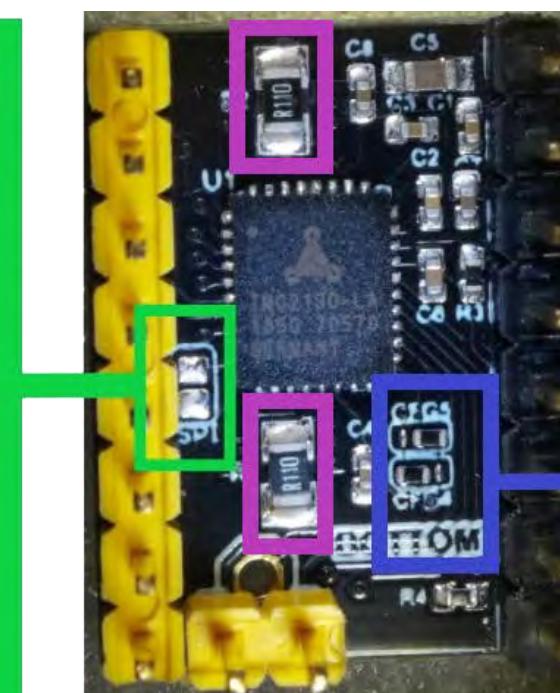
BIQU TMC2130

SPI Mode

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 the correct two pads are soldered together to form a bridge, as shown in the **BLUE** box. The **PURPLE** box shows the location of the current sense resistors (R_s). 

Note: When the stall-guard function is **not used**, the stall-guard pin ("R") 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. 



NOTICE: pin labels are relative to stepper driver board; double check compatibility before use. Numbered pairs denote required jumpers.

SPI Mode**BIQU TMC2130**SPI Mode

R - Stall Detection

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

$R_s = R050$ is 0.05 Ohms

$R_s = R062$ is 0.062 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R075$ is 0.075 Ohms

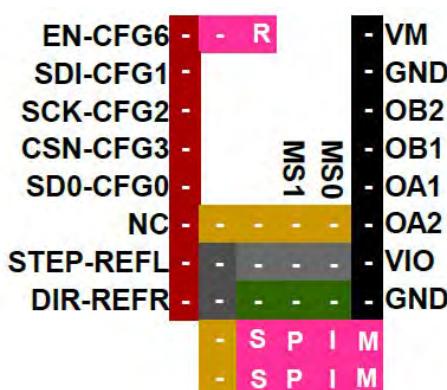
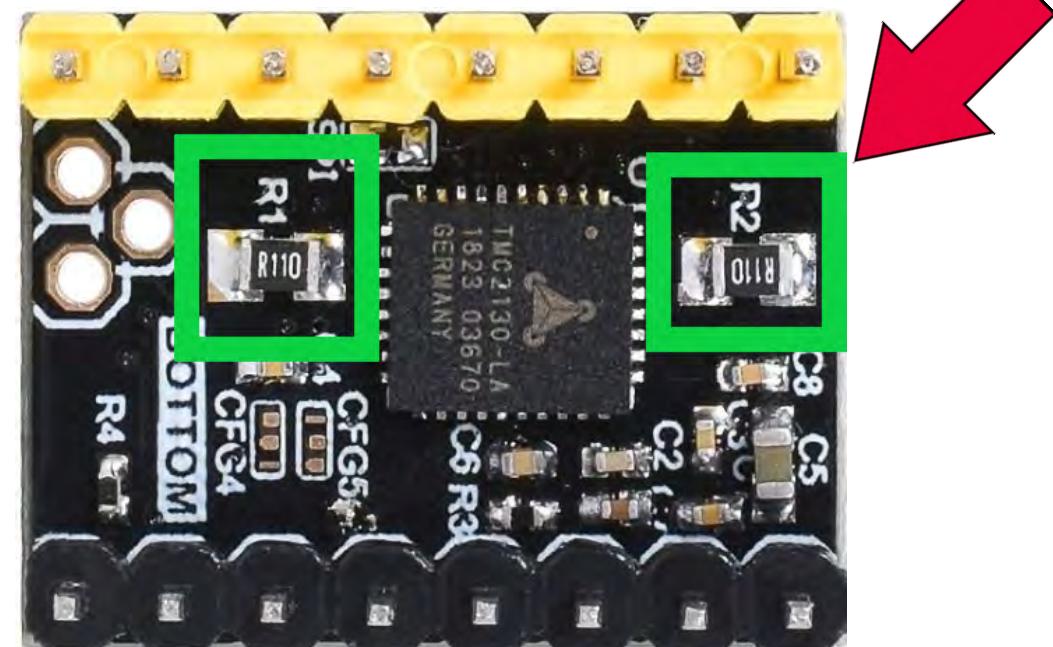
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

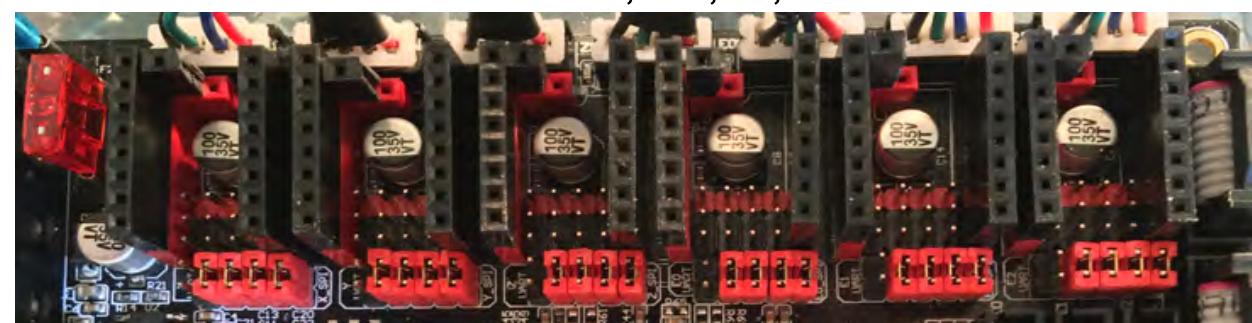
$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

**SPI**

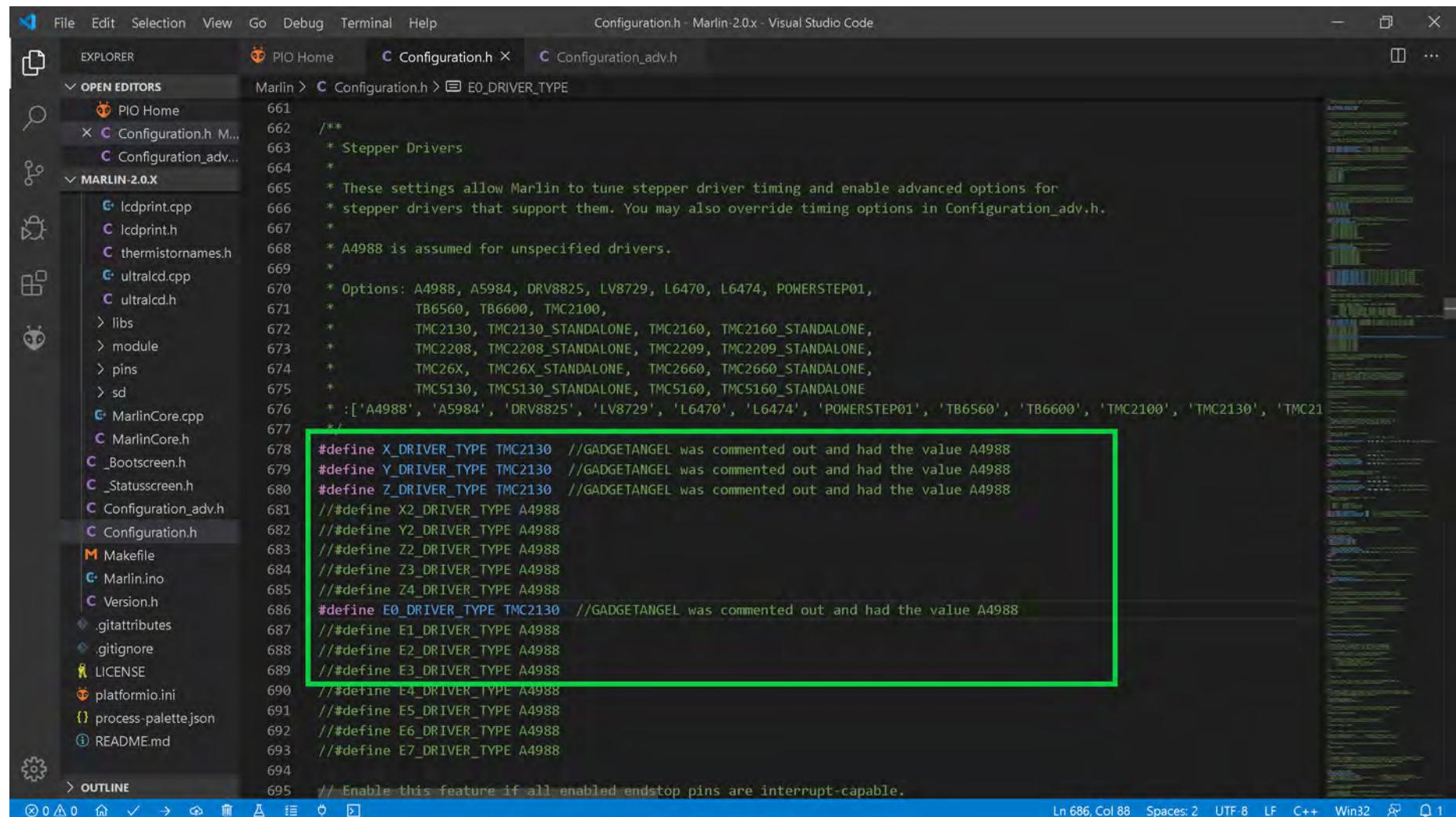
Note: Set JUMPERS "S","P","I","M" on the board!!



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

NOTE: Go to Appendix C, and 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 BIQU 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 ("//").



```

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
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home 661 /**
  * Stepper Drivers
  *
  * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  *
  * A4988 is assumed for unspecified drivers.
  *
  * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
  *          TB6560, TB6600, TMC2100,
  *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
  *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
  *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
  *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
  *          :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
  */
#define X_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value 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 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 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

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):** Shows the project structure under MARLIN-2.0.X, including files like sanguino, stm32, teensy2, teensy3, pins.h, MarlinCore.cpp, MarlinCore.h, Bootscreen.h, Statusscreen.h, Configuration_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, Configurations-release-2.0.3.zip, LICENSE, platformio.ini, process-palette.json, and README.md.
- Central Area:** The code editor displays Configuration.h with the following relevant snippet highlighted by a green box:

```
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        // GCODE was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GCODE was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GCODE was false, stepper motor driver inverts motor direction
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       // GCODE was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```
- Right Sidebar:** Shows a vertical stack of code snippets for different printer models (sanguino, stm32, teensy2, teensy3).

- Go to the next page.

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

- Next you want to set your V_{ref} 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 V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
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. //GADGETANGEL was 580
2004         #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005         #define X_MICROSTEPS       16       // 0..256
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      //GADGETANGEL was 580
2020         #define Y_CURRENT_HOME    Y_CURRENT
2021         #define Y_MICROSTEPS       16
2022         #define Y_RSENSE            0.11
2023         #define Y_CHAIN_POS         -1
2024     #endif

```

Ln 199, Col 66 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

- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the **GREEN** boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.11. Ensure "E0_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "Z_CURRENT" and/or "E0_CURRENT".

```

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

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760           //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME Z_CURRENT
2037   #define Z_MICROSTEPS    16
2038   #define Z_RSENSE        0.11
2039   #define Z_CHAIN_POS     -1
2040 #endif

```

```

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

Marlin > Configuration.h > Configuration_adv.h > E0_CURRENT

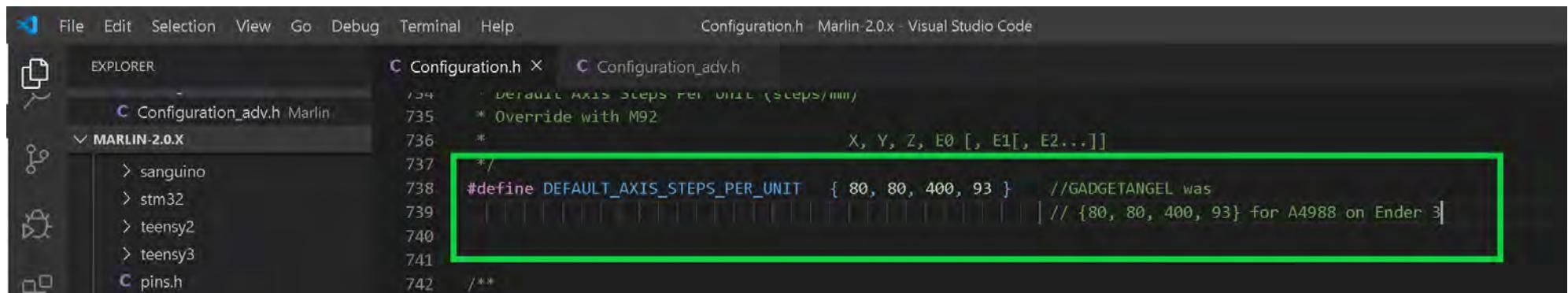
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900           //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- Go to the next page.

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

- 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 Configuration.h X Configuration_adv.h

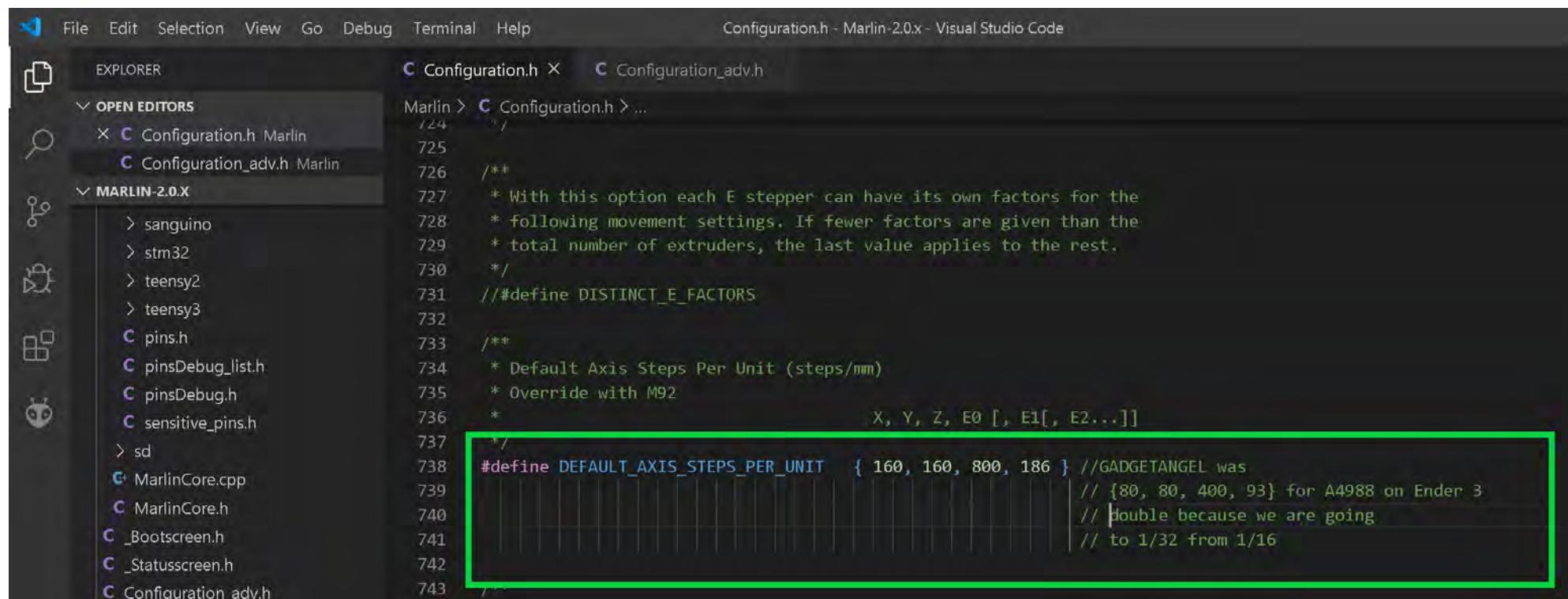
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

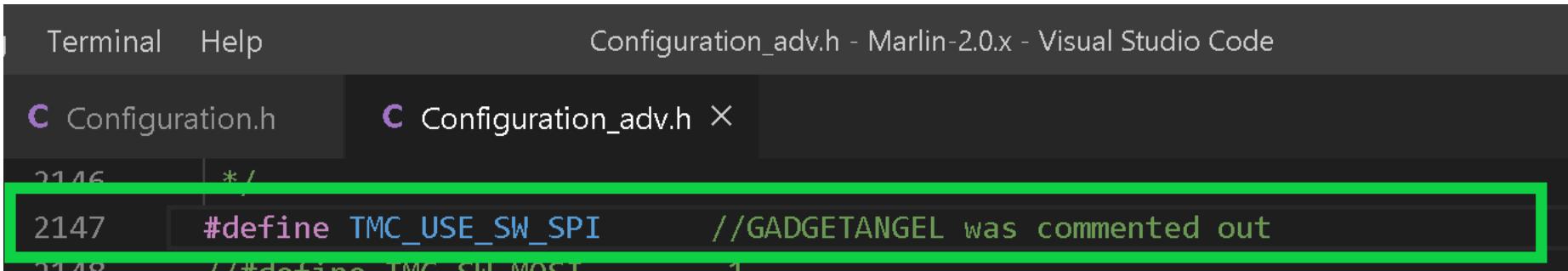
724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 /**

```

- 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 its SPI pins

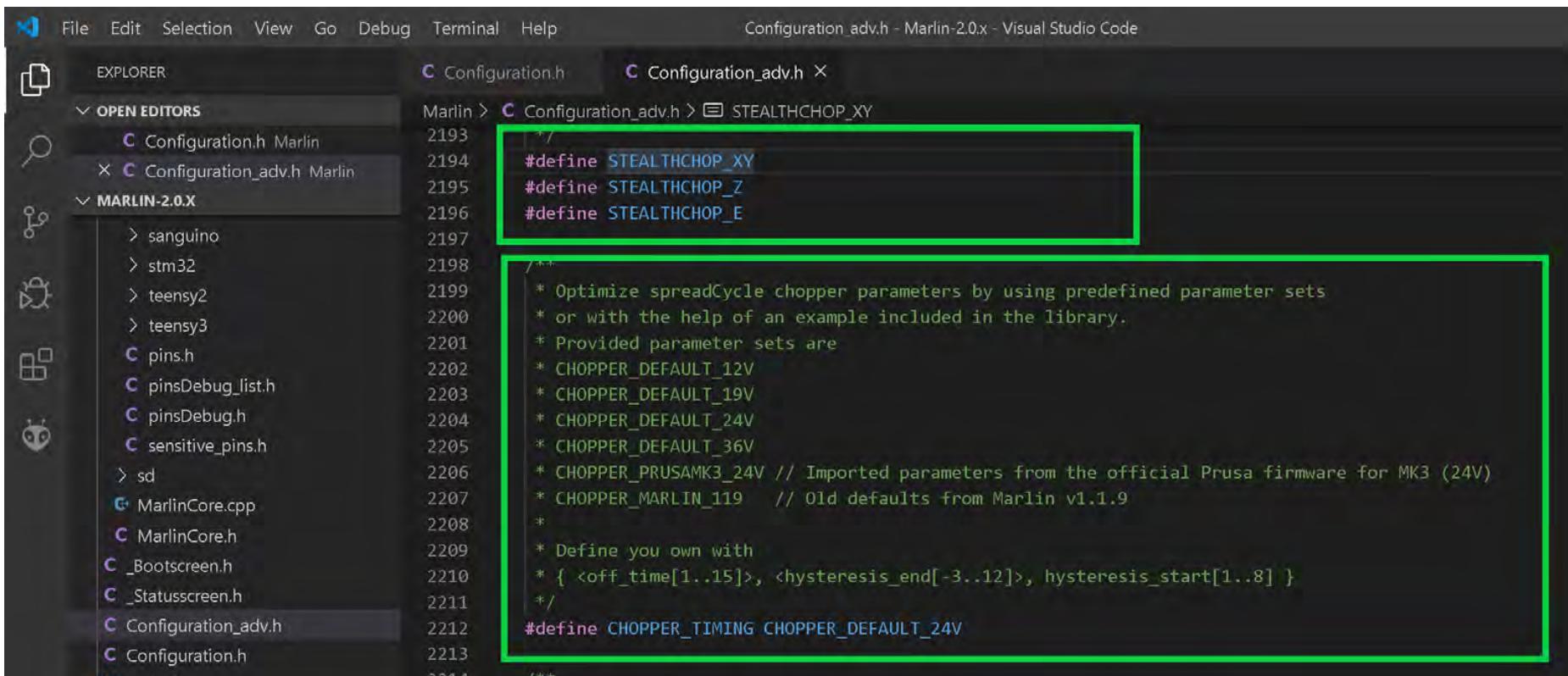


Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

C Configuration.h C Configuration_adv.h X

```
2146 */  
2147 #define TMC_USE_SW_SPI //GADGETANGEL was commented out  
2148 /*#define TMC_SW_MOST
```

- 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 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 Configuration.h Configuration_adv.h X

Marlin > C Configuration_adv.h > STEALTHCHOP_XY

```
2193 /*  
2194 #define STEALTHCHOP_XY  
2195 #define STEALTHCHOP_Z  
2196 #define STEALTHCHOP_E  
2197 */  
2198 * Optimize spreadCycle chopper parameters by using predefined parameter sets  
2199 * or with the help of an example included in the library.  
2200 * Provided parameter sets are  
2201 * CHOPPER_DEFAULT_12V  
2202 * CHOPPER_DEFAULT_19V  
2203 * CHOPPER_DEFAULT_24V  
2204 * CHOPPER_DEFAULT_36V  
2205 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)  
2206 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9  
2207 *  
2208 * Define your own with  
2209 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }  
2210 * /  
2211 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V  
2212 */  
2213 */
```

- 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 following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
- Left Sidebar (Explorer):** Shows the project structure under "MARLIN-2.0.X". The "Configuration_adv.h" file is currently selected.
- Central Area:** The code editor displays the "Configuration_adv.h" file. A green box highlights the section of code defining the HYBRID_THRESHOLD constant and its sub-thresholds for X, Y, Z, and E axes.

```
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259

/** 
 * 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 //GADGETANGEL was commented out

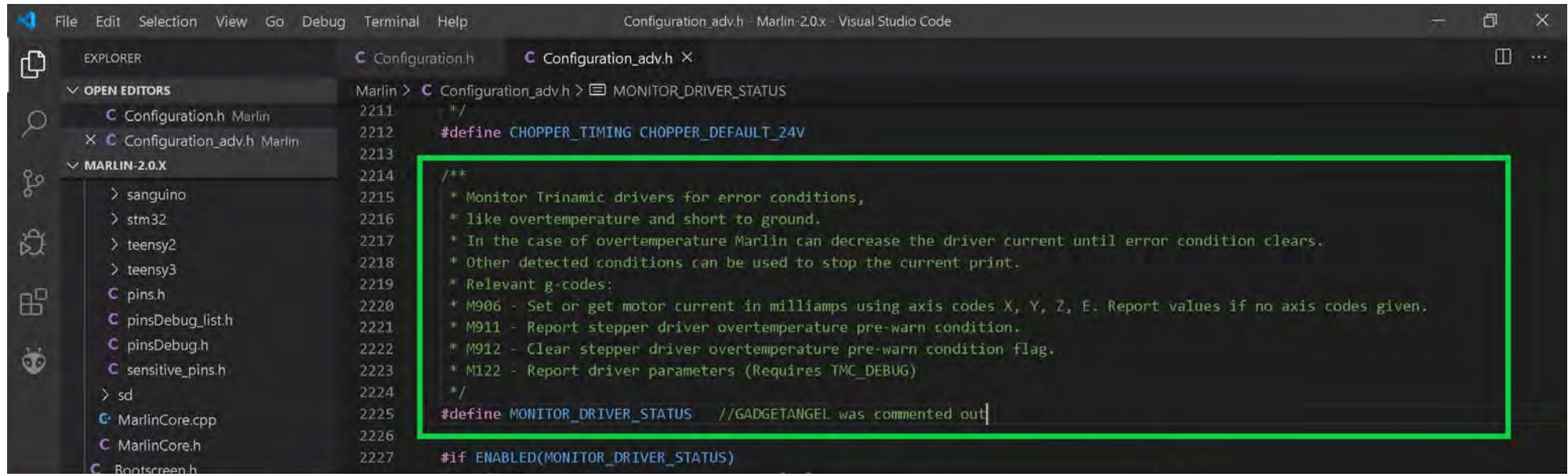
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 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

/**
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2130 Drivers in SPI 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](#).



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

EXPLORER Configuration.h Configuration_adv.h

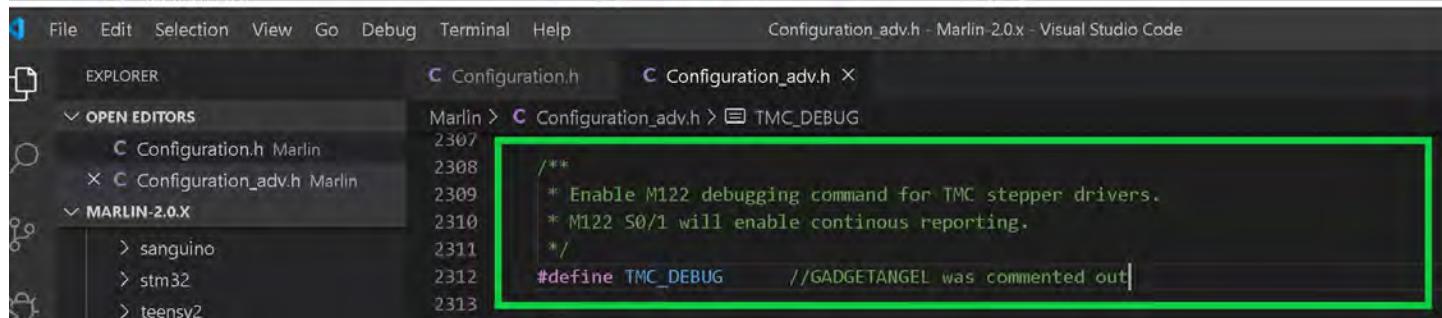
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

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 //GADGETANGEL 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

EXPLORER Configuration.h Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

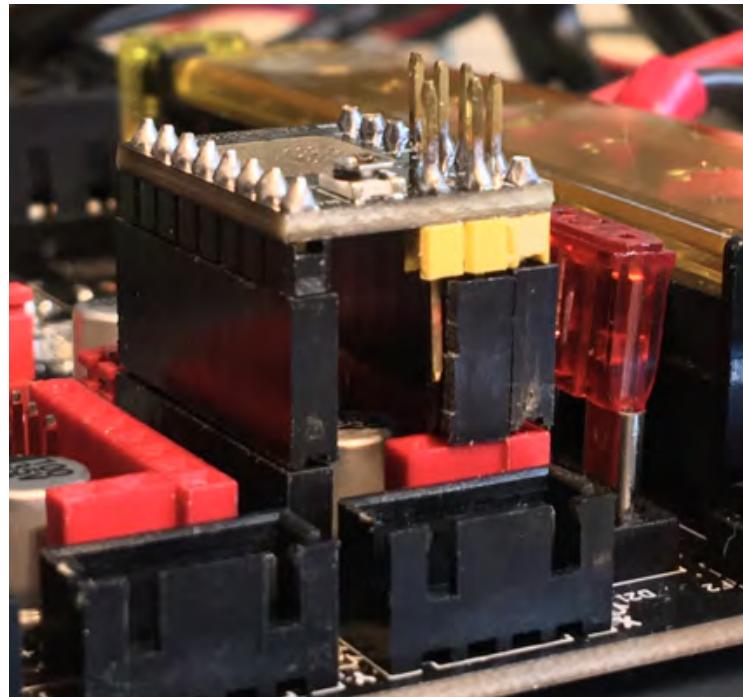
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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 DIAG1 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 DIAG1 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



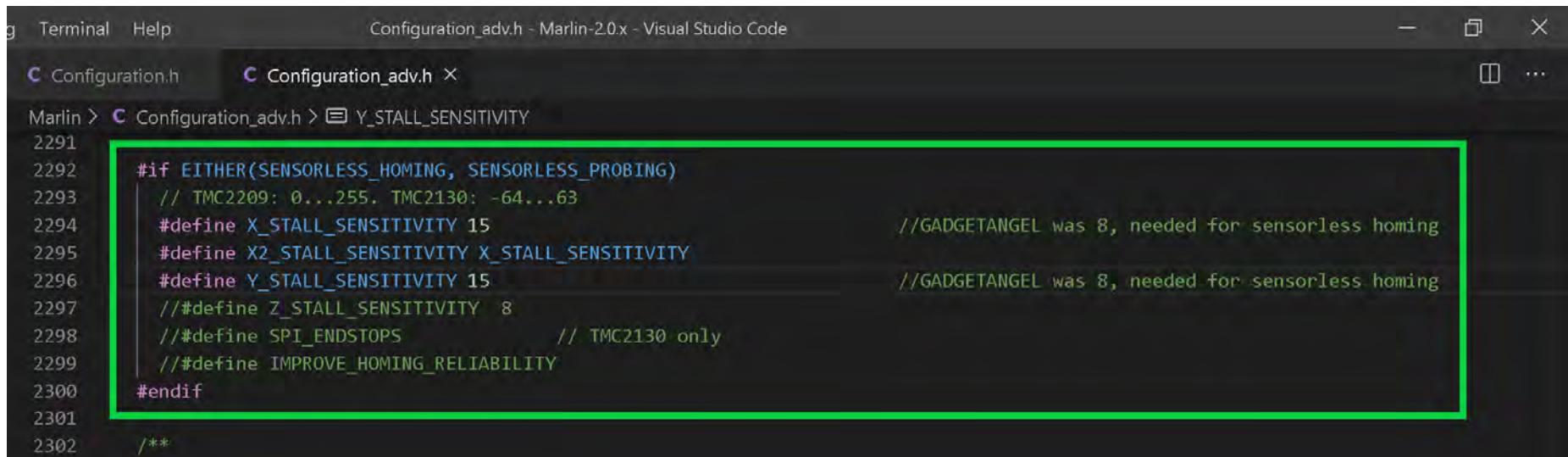
- Sensor-less homing is commented out by default. So I remove the two leading "://" to un-comment "SENSORLESS_HOMING"

```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code  
EXPLORER Configuration.h Configuration_adv.h X  
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING  
2281 */  
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out  
2283
```

- Go to the next page.

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

- Next we set the "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".

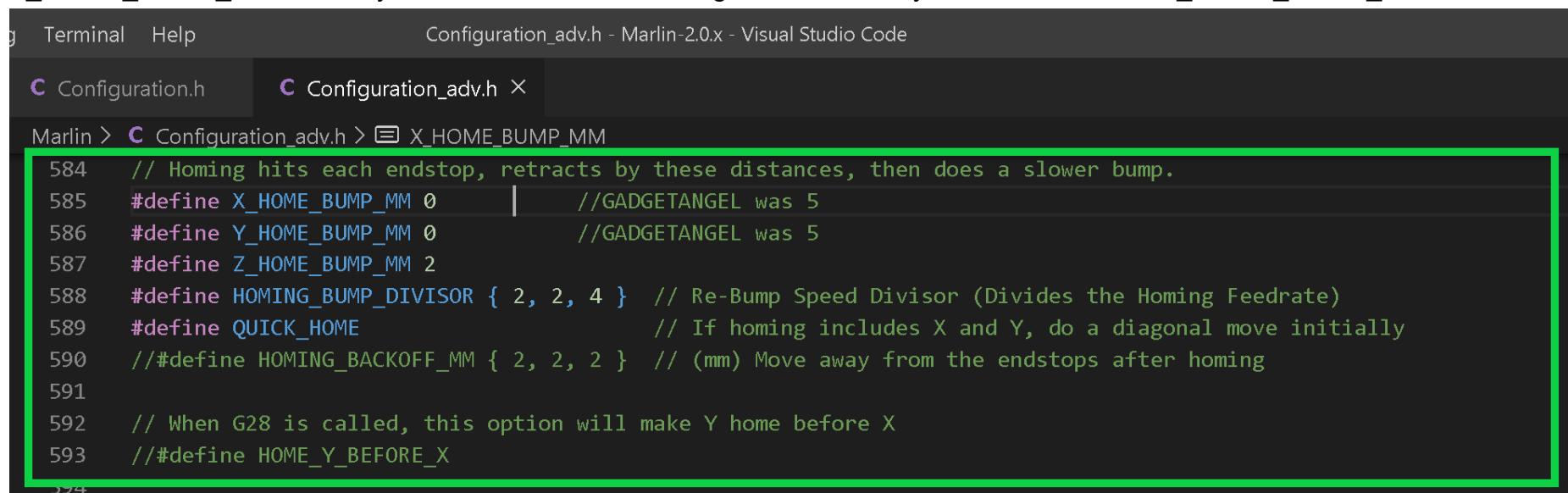


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- 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.



```

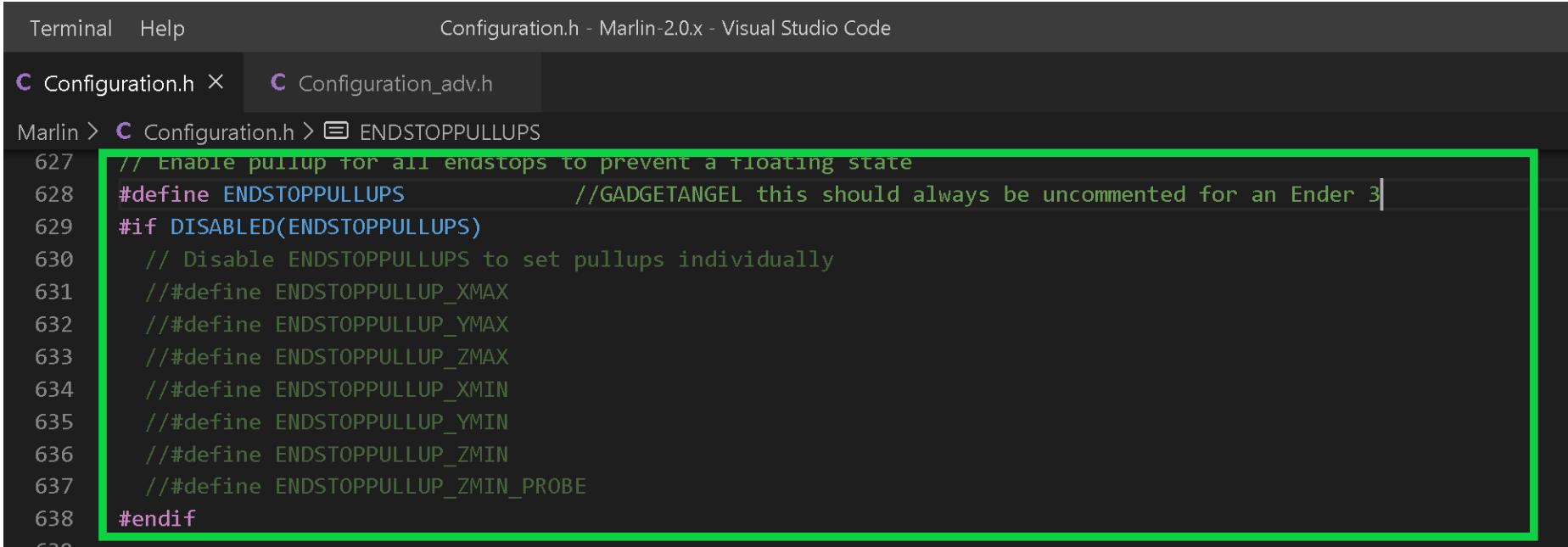
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL 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

```

- 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.



```

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

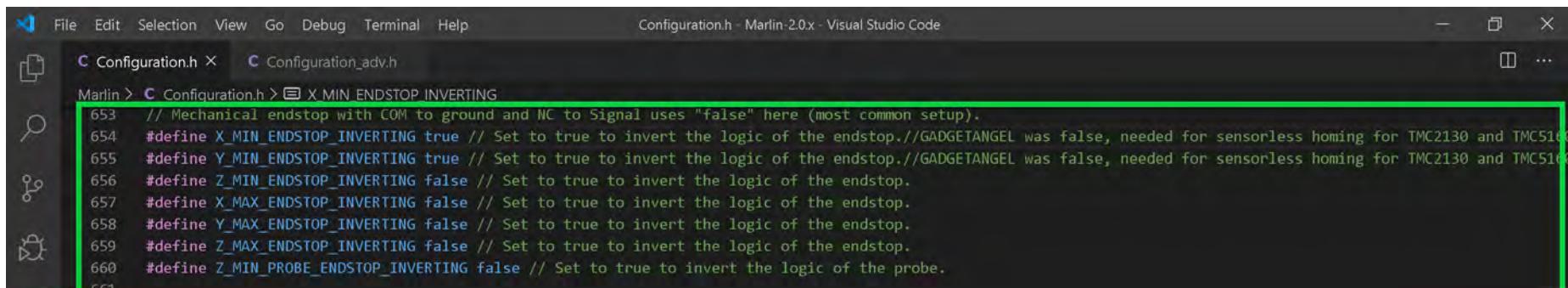
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > ENDSTOPPULLUPS

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL 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

```

- Next to allow sensor-less homing to work (while using the BIQU TMC2130) 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

C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > X-MIN ENDSTOP INVERTING

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.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5100
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5100
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

```

- 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.

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

EXPLORER OPEN EDITORS MARLIN-2.0.X

pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h

Marlin > Configuration.h > X_DRIVER_TYPE

```
#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.  
#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.  
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
```

/*
 * Stepper Drivers
 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 * A4988 is assumed for unspecified drivers.
 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 * TB6560, TB6600, TMC2100,
 * TMC2130, TMC2130_STANDALONE, TMC2100, TMC2100_STANDALONE,
 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build

BIGTREETECH_BTT002	IGNORED	SUCCESS	00:02:31.294
BIGTREETECH_BTT002	IGNORED		
teensy31	IGNORED		
teensy35	IGNORED		
esp32	IGNORED		
linux_native	IGNORED		
SAMD51_grandcentral_m4	IGNORED		
rumba32_f446ve	IGNORED		
mks_rumba32	IGNORED		
include_tree	IGNORED		

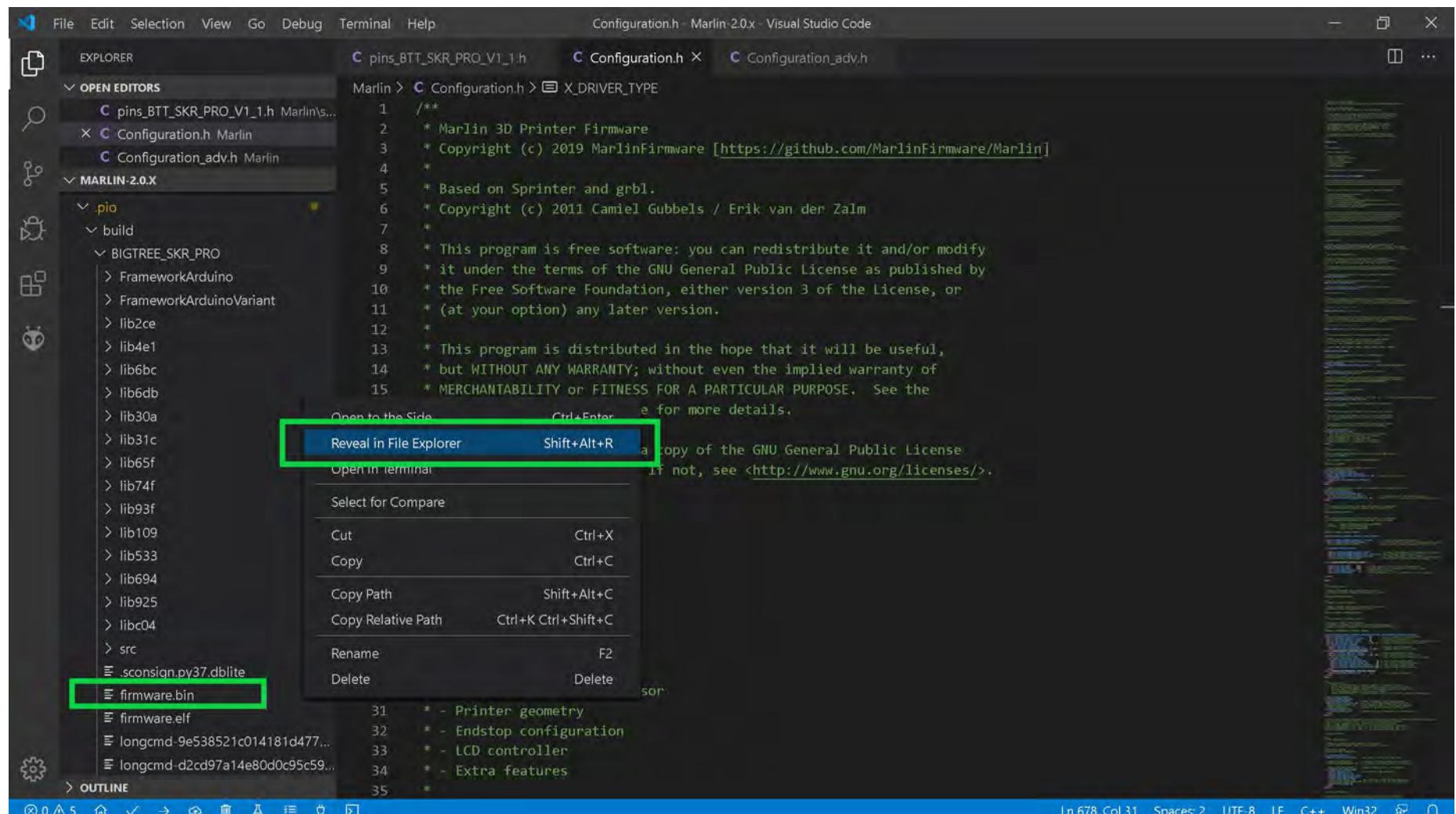
```
===== 1 succeeded in 00:02:31.294 =====
```

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 TMC2130 Drivers in SPI Mode

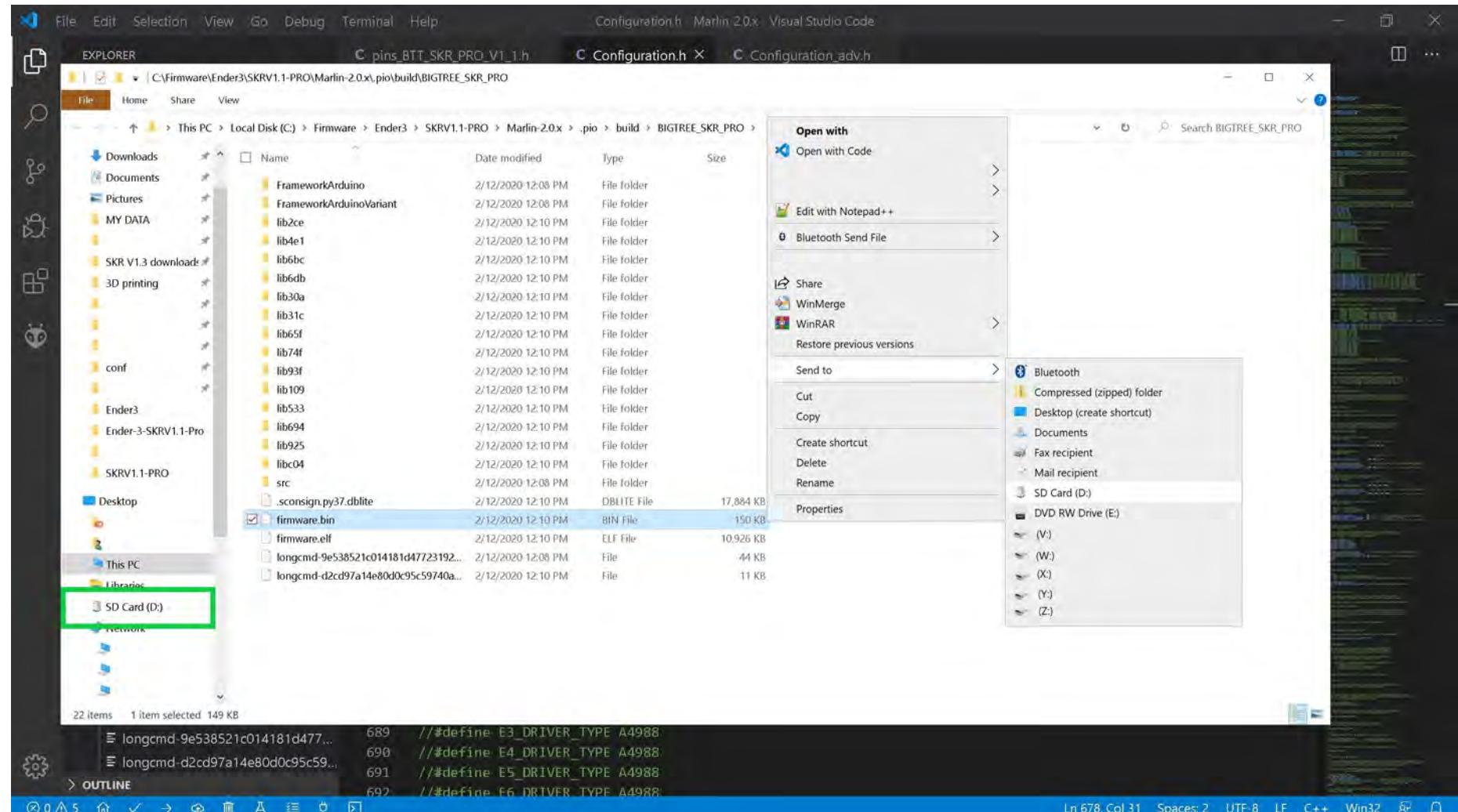
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



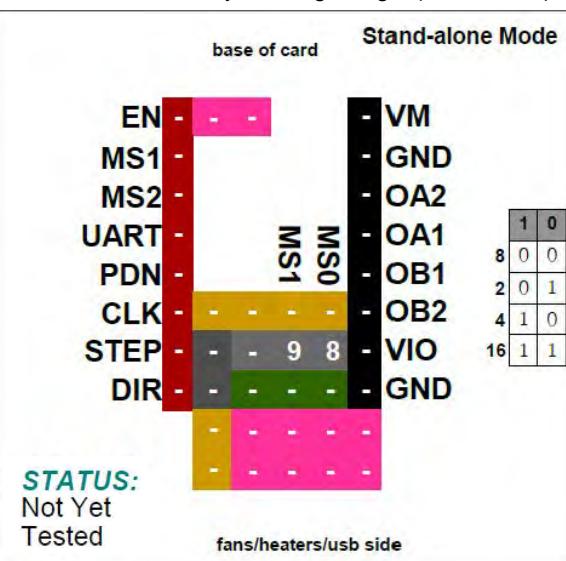
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".

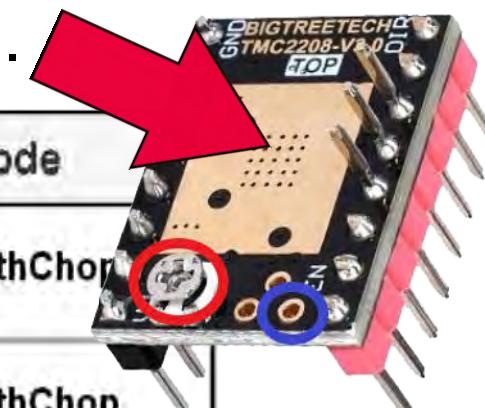


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

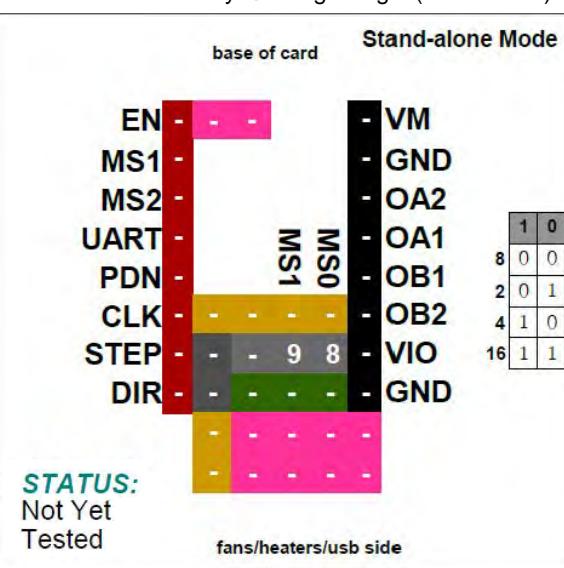
BIQU TMC2208 V3.0**Stand-alone Mode**

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.

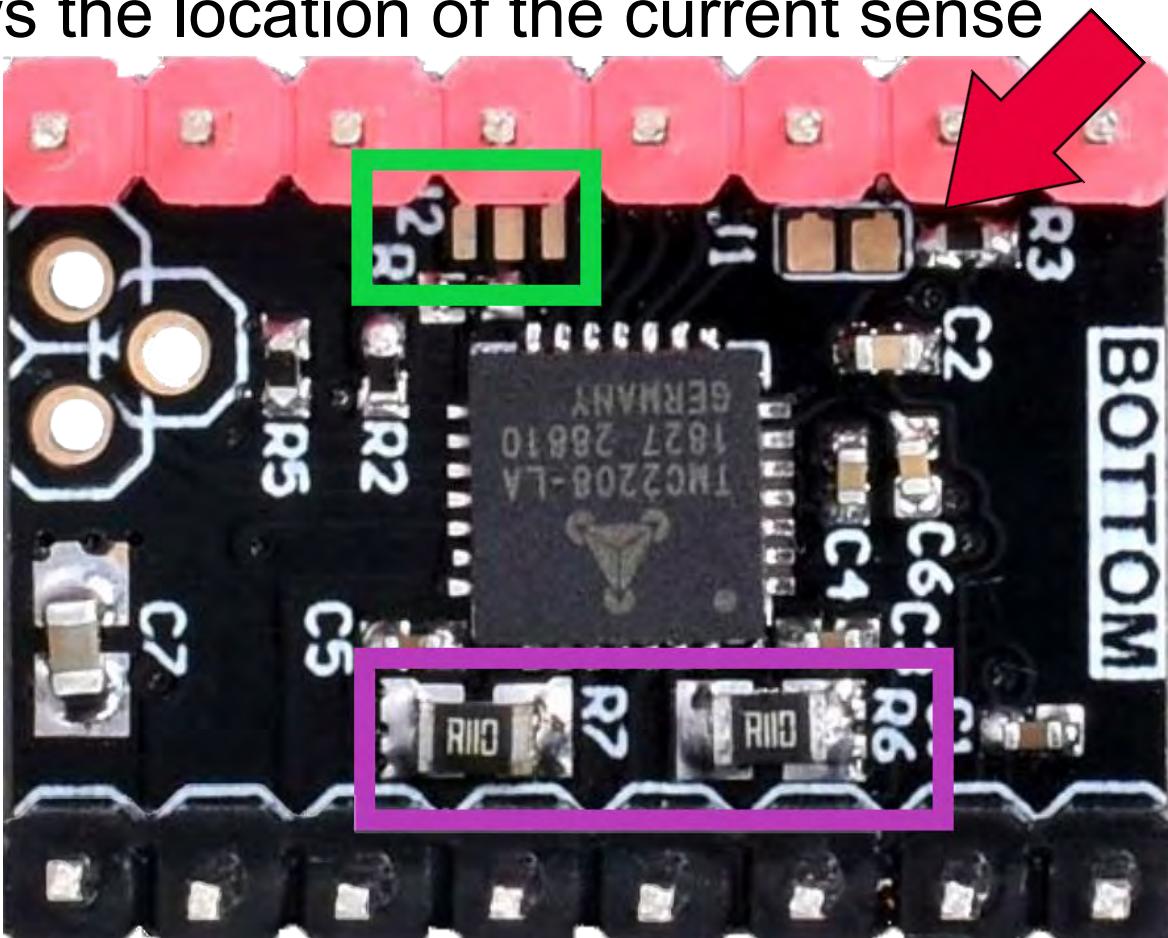


Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2208 <small>Stand Alone Mode Maximum 16 Subdivision 35V DC 2A (peak)</small>	GND	GND	1 / 8	1 / 256	stealthChop
	GND	VIO	1 / 2	1 / 256	stealthChop
	VIO	GND	1 / 4	1 / 256	stealthChop
	VIO	VIO	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula	$I_{MAX} = V_{ref} * 0.9286$			$V_{ref} = I_{MAX} * 1.0769$	
R_S (Typical Sense Resistor) = 0.11Ω	See Appendix B #3. Use 50% to 90% as shown below:			See Appendix B #3. Use 50% to 90% as shown below:	
	$I_{MAX} = (V_{ref} * 0.9286) * 0.90$			$V_{ref} = (I_{MAX} * 1.0769) * 0.90$	

BIQU TMC2208 V3.0**Stand-alone Mode**

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 stand-alone mode for the TMC2208 V3.0, as seen in the **GREEN** box below. The **PURPLE** box shows the location of the current sense resistors (R_s).

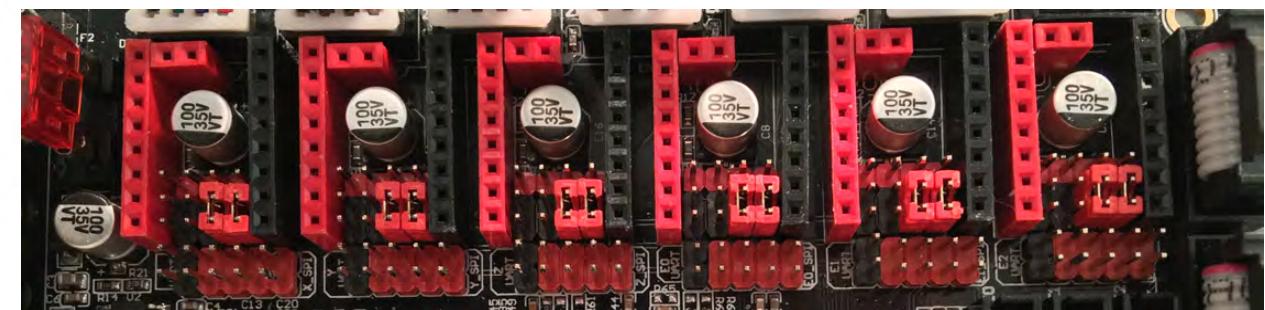
Note: MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).



Stand-alone Mode

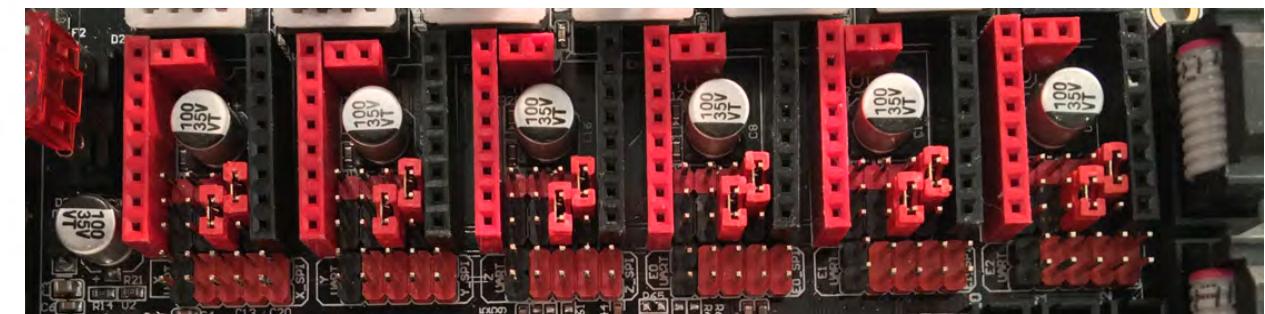
Stand-alone Mode
1 / 8
 Interpolation: **1/256**
 StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	-	OA1
PDN	-	MS1	MS0	OB1
CLK	-	-	-	OB2
STEP	-	-	9 8	VIO
DIR	-	9	8	GND
	-	-	-	-



Stand-alone Mode
1 / 2
 Interpolation: **1/256**
 StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	-	OA1
PDN	-	MS1	MS0	OB1
CLK	-	-	-	OB2
STEP	-	-	9 8	VIO
DIR	-	9	-	GND
	-	-	-	-



Stand-alone Mode

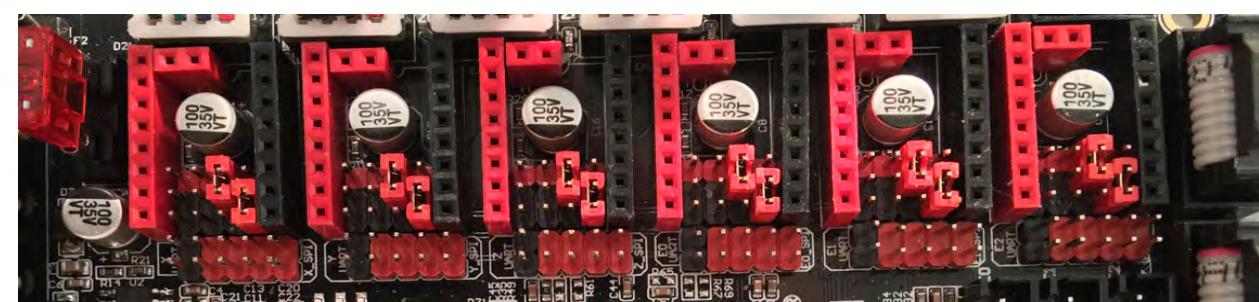
Stand-alone Mode

1 / 4

Interpolation: 1/256

StealthChop

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	MS1	OA1
PDN	-	MS0	OB1
CLK	-	9	OB2
STEP	-	9 8	VIO
DIR	-	8	GND



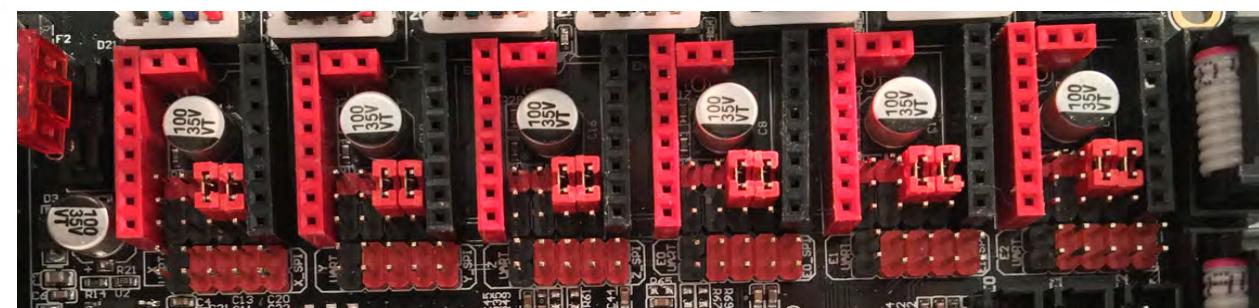
Stand-alone Mode

1 / 16

Interpolation: 1/256

StealthChop

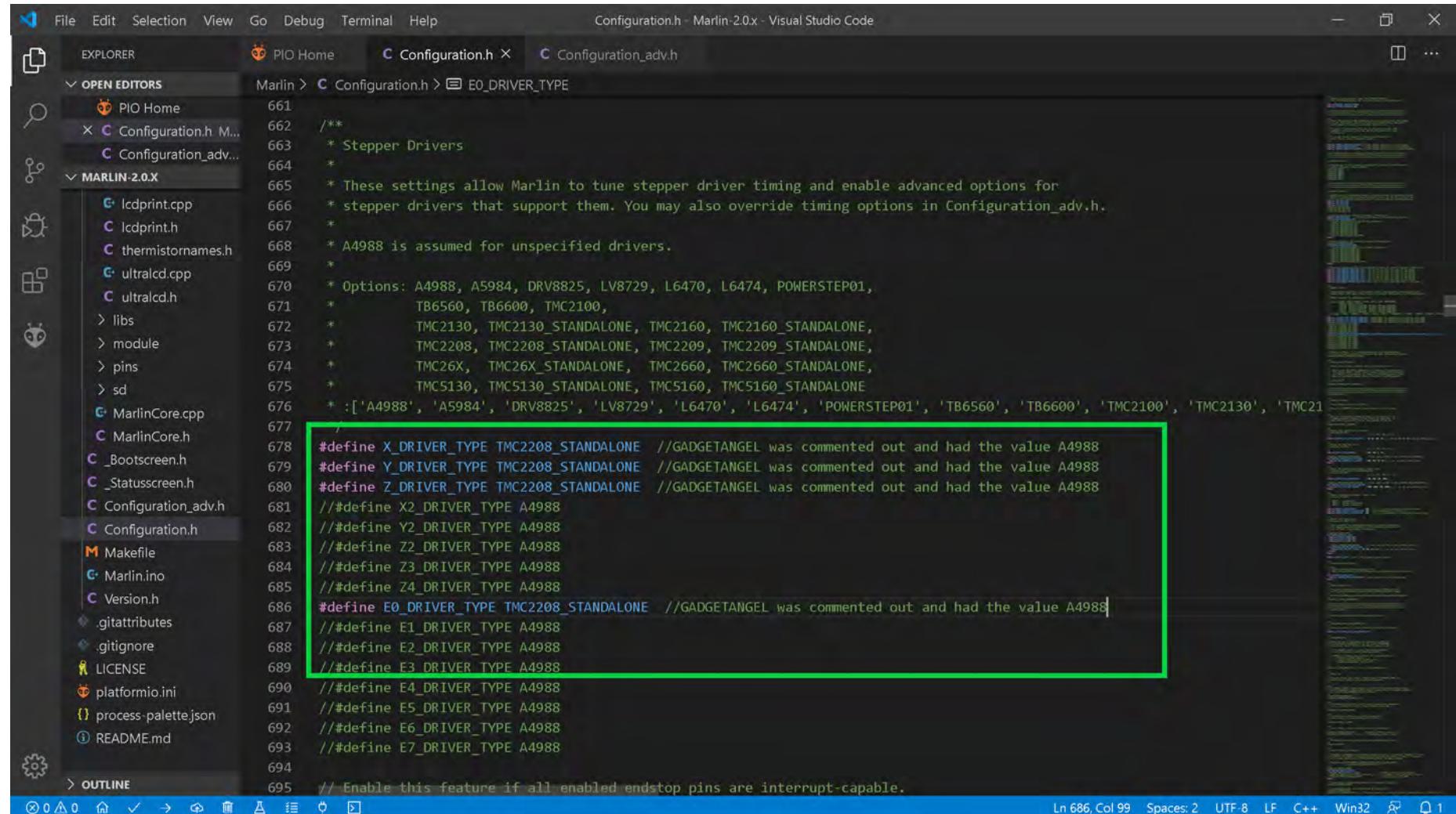
EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	MS1	OA1
PDN	-	MS0	OB1
CLK	-	9 8	OB2
STEP	-	9 8	VIO
DIR	-	-	GND



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

NOTE: Go to Appendix C, and 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 X Configuration_adv.h
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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
677 */
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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 TMC2208 V3.0 Drivers in Stand-alone Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, 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 following snippet:

```

    /**
     * With this option each E stepper can have its own factors for the
     * following movement settings. If fewer factors are given than the
     * total number of extruders, the last value applies to the rest.
     */
#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *
 * X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 *
 * X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

#ifndef LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
#if ENABLED(LIMITED_MAX_FR_EDITING)
#define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif
#endif

/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 *
 * Override with M201
 */

```

A green rectangular box highlights the line `#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 }`. The status bar at the bottom right shows 'Ln 738, Col 62'.

- 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

```

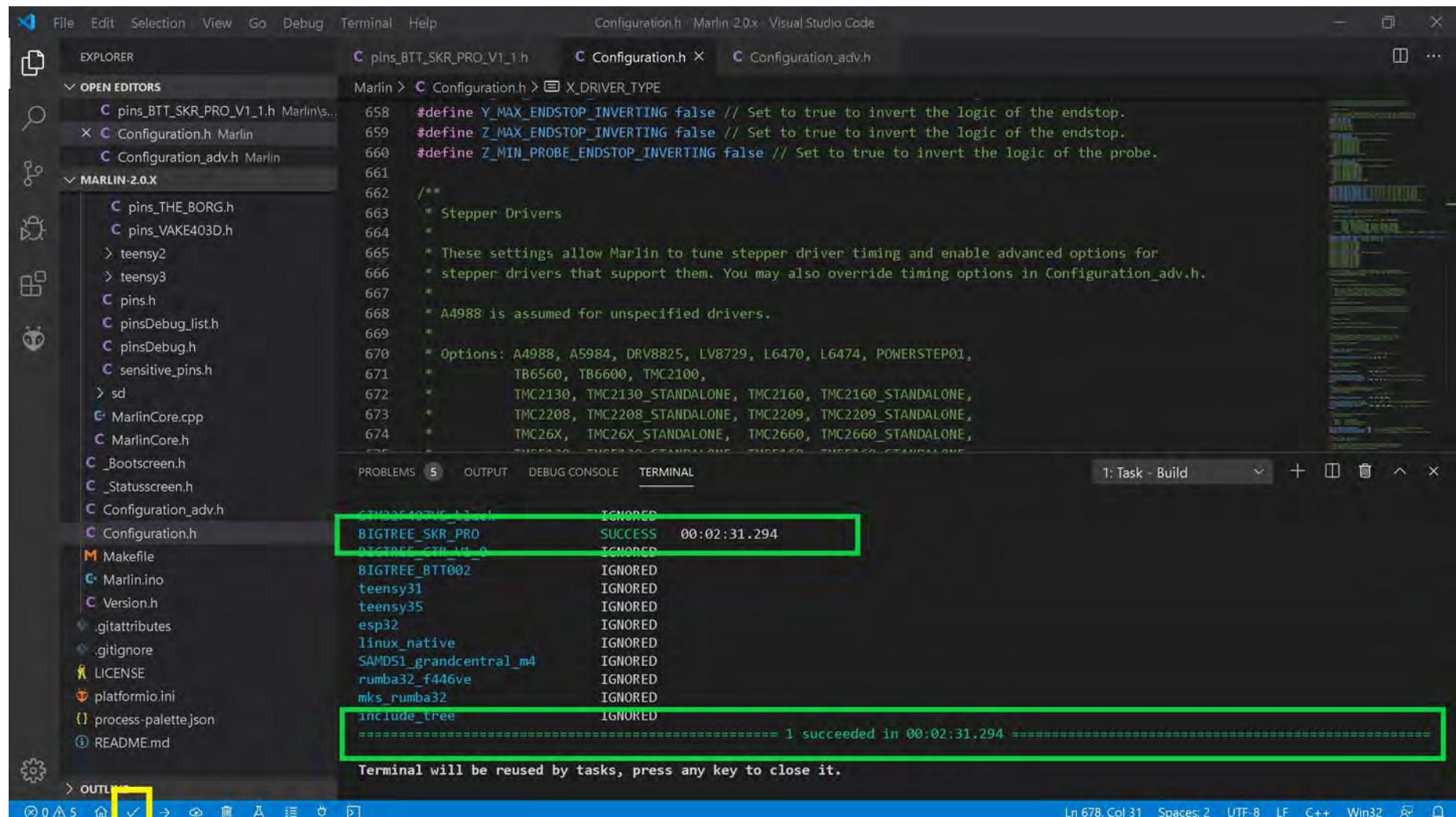
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  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
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palette.json
  README.md
  OUTLINE
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

```

- 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.



```

Configuration.h Marlin 2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help Configuration.h X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\pins\pins_BTT_SKR_PRO_V1_1.h
Configuration.h Marlin\pins\pins.h
Configuration_adv.h Marlin\pins\pinsDebug_list.h
pins.h pinsDebug.h
pinsDebug_list.h sensitive_pins.h
teensy2 teensy3
pinsDebug.h
pinsDebug_list.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 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 ☰

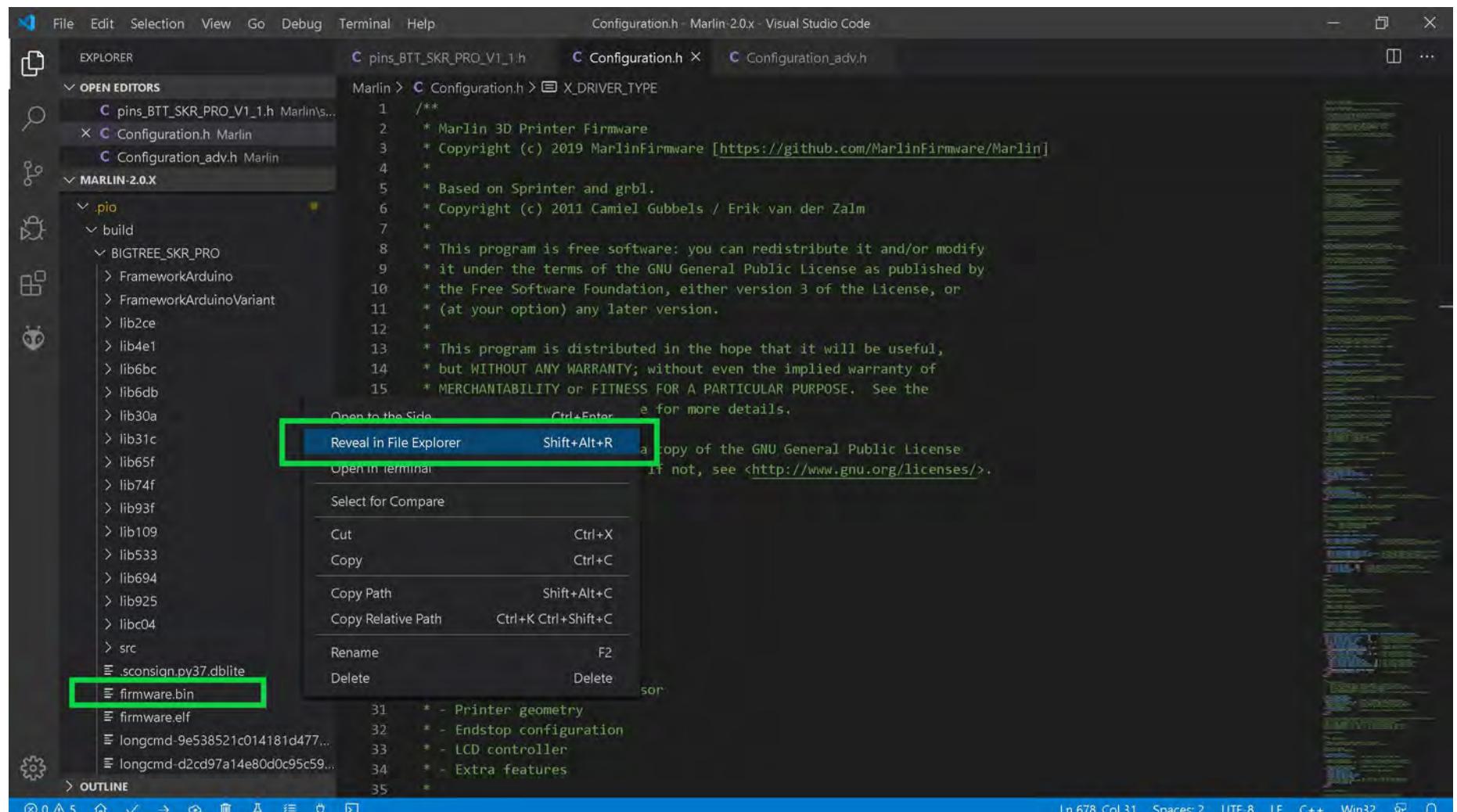
```

The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The terminal tab is active, displaying the build output. A yellow box highlights the checkmark icon in the terminal toolbar. A green box highlights the successful compilation message "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

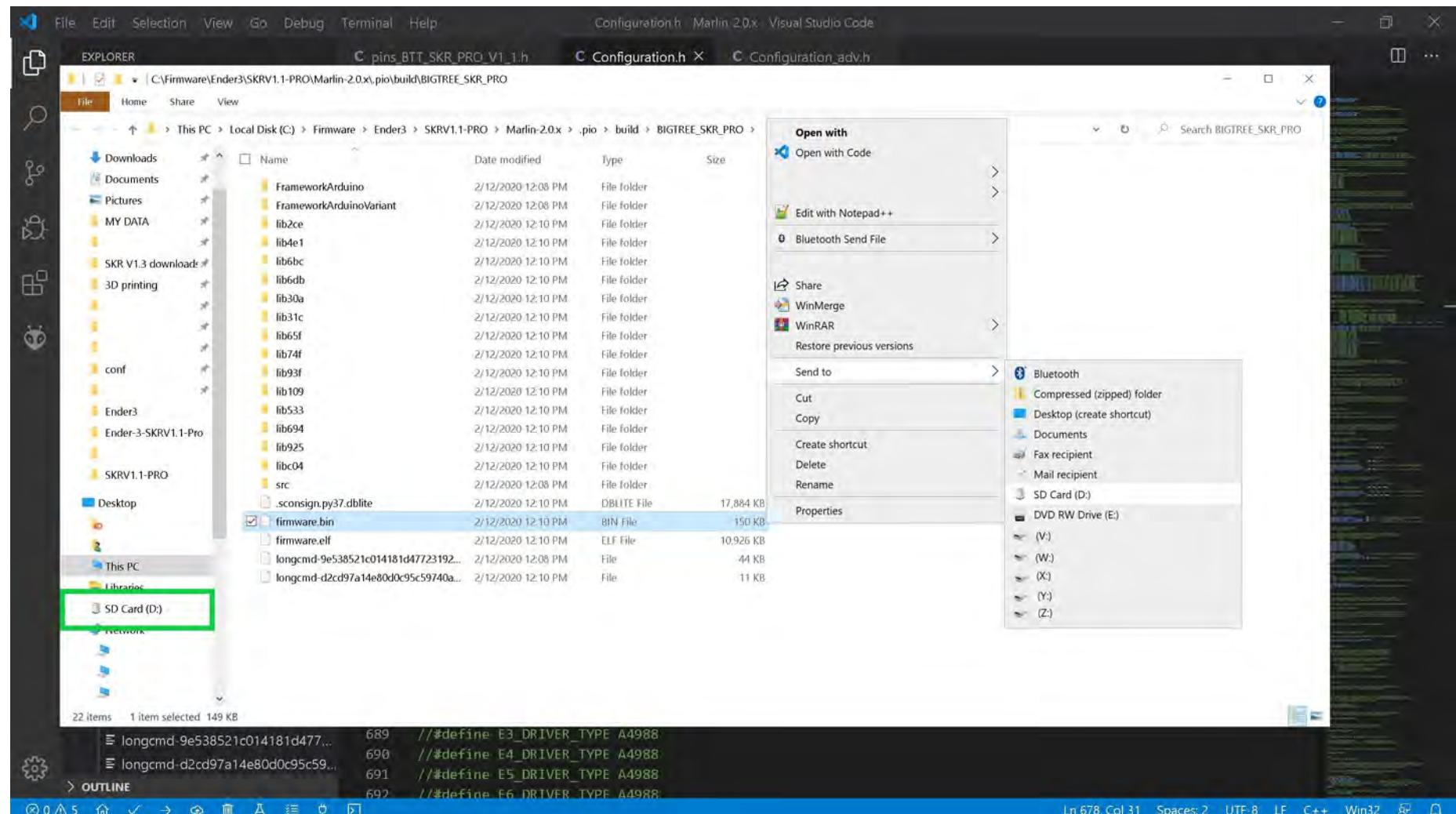
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



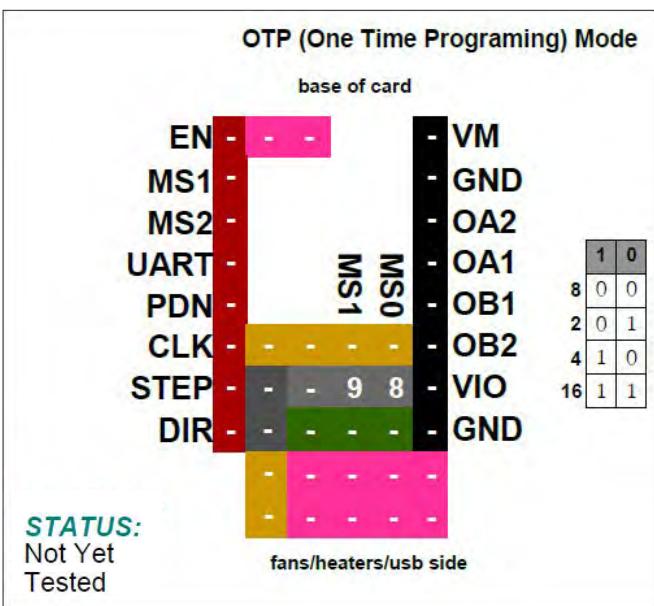
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".

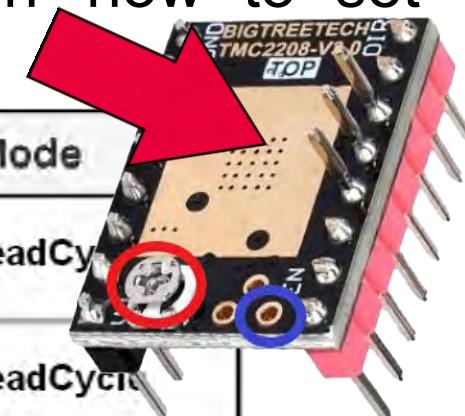


- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

One Time Programming (OTP) Mode

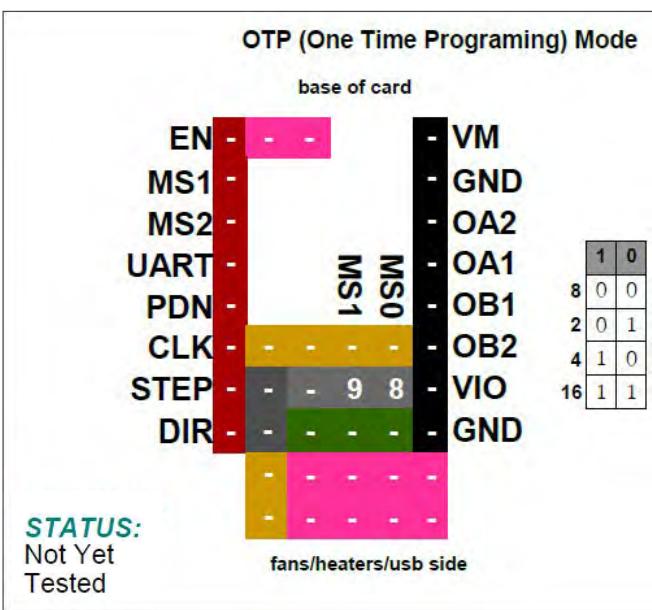
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.



Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BBIQU® TMC2208 <small>OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)</small>	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

Driving Current Calculation Formula R_S (Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}*0.9286$ See Appendix B #3. Use 50% to 90% as shown below: $I_{MAX}=(V_{ref}*0.9286)*0.90$	$V_{ref}=I_{MAX}*1.0769$ See Appendix B #3. Use 50% to 90% as shown below: $V_{ref}=(I_{MAX}*1.0769)*0.90$
---	--	--

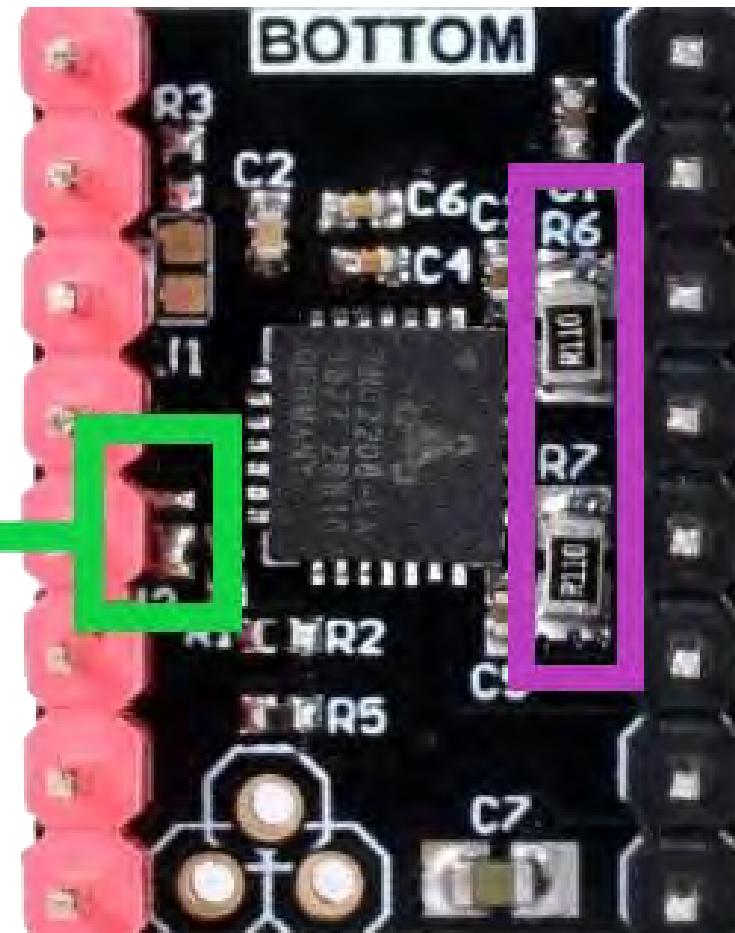


As an example, the picture shows the TOP two pads soldered together, as shown in the **GREEN box**, so that OTP mode can be obtained. To do One-Time-Programming (OTP), the TMC2208 must be placed in UART mode to program it. The **PURPLE box** shows the location of the current sense resistors (R_s).

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

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.



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).

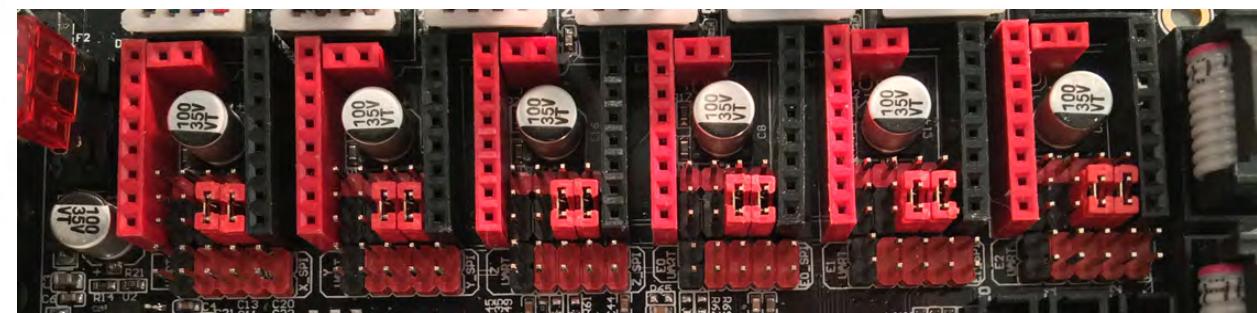
OTP (One Time Programming) Mode

BIGU TMC2208 V3.0

One Time Programming (OTP) Mode

OTP
1 / 8Interpolation: 1/256
SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	-	OA1
PDN	-	MS1	OB1
CLK	-	-	OB2
STEP	-	9 8	VIO
DIR	-	9 8	GND
	-	-	-

OTP
1 / 2Interpolation: 1/256
SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	-	OA1
PDN	-	MS1	OB1
CLK	-	-	OB2
STEP	-	9 8	VIO
DIR	-	9 -	GND
	-	-	-



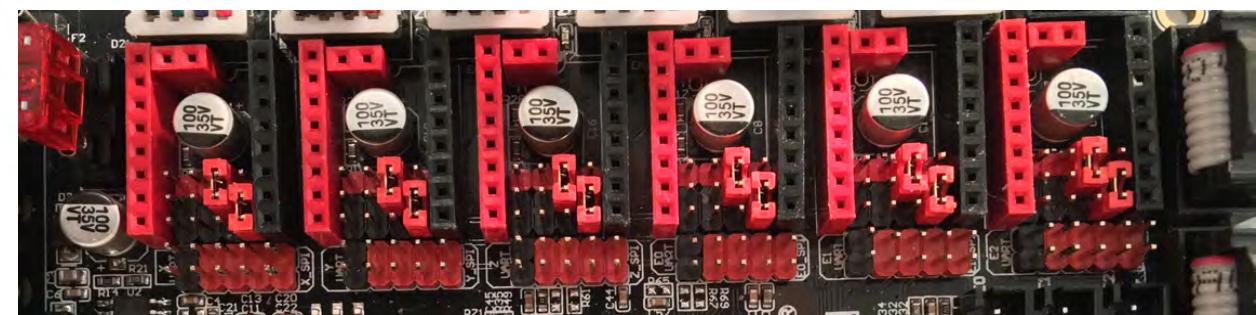
OTP (One Time Programming) Mode

OTP
1 / 4

Interpolation: 1/256
SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	-	OA1
PDN	MS1	MS0	OB1
CLK	-	9	OB2
STEP	-	9 8	VIO
DIR	-	-	GND
	-	-	8

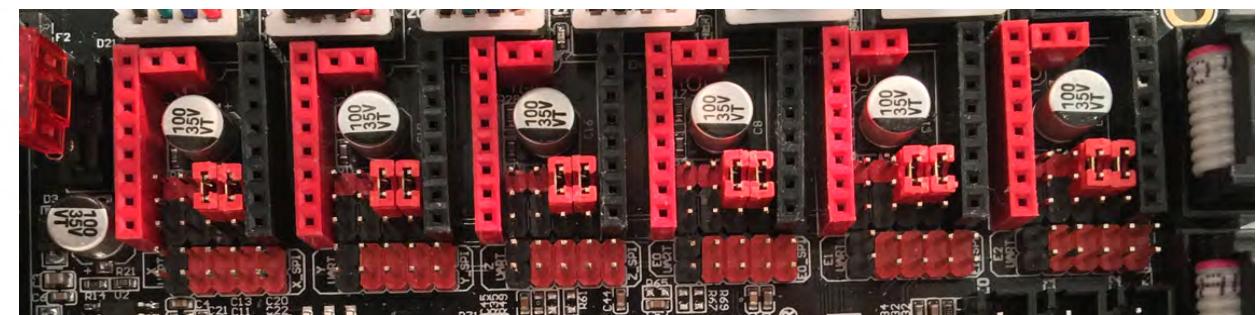
One Time Programming (OTP) Mode



OTP
1 / 16

Interpolation: 1/256
SpreadCycle

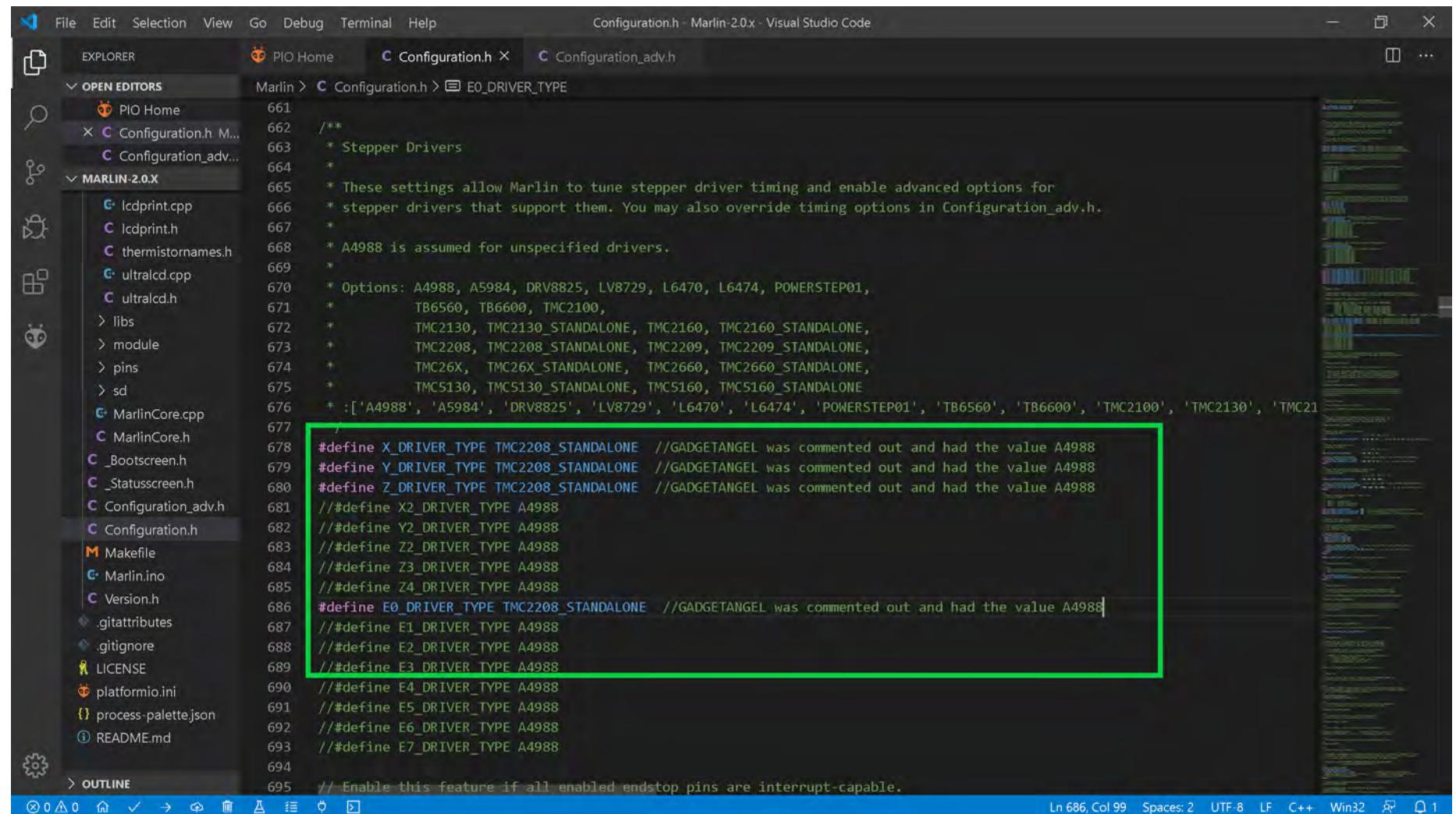
EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	-	OA1
PDN	MS1	MS0	OB1
CLK	-	9 8	OB2
STEP	-	9 8	VIO
DIR	-	-	GND
	-	-	8



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

NOTE: Go to Appendix C, and 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 ("//").



```

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
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
  PIO Home
  Configuration.h M...
  Configuration_adv...
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

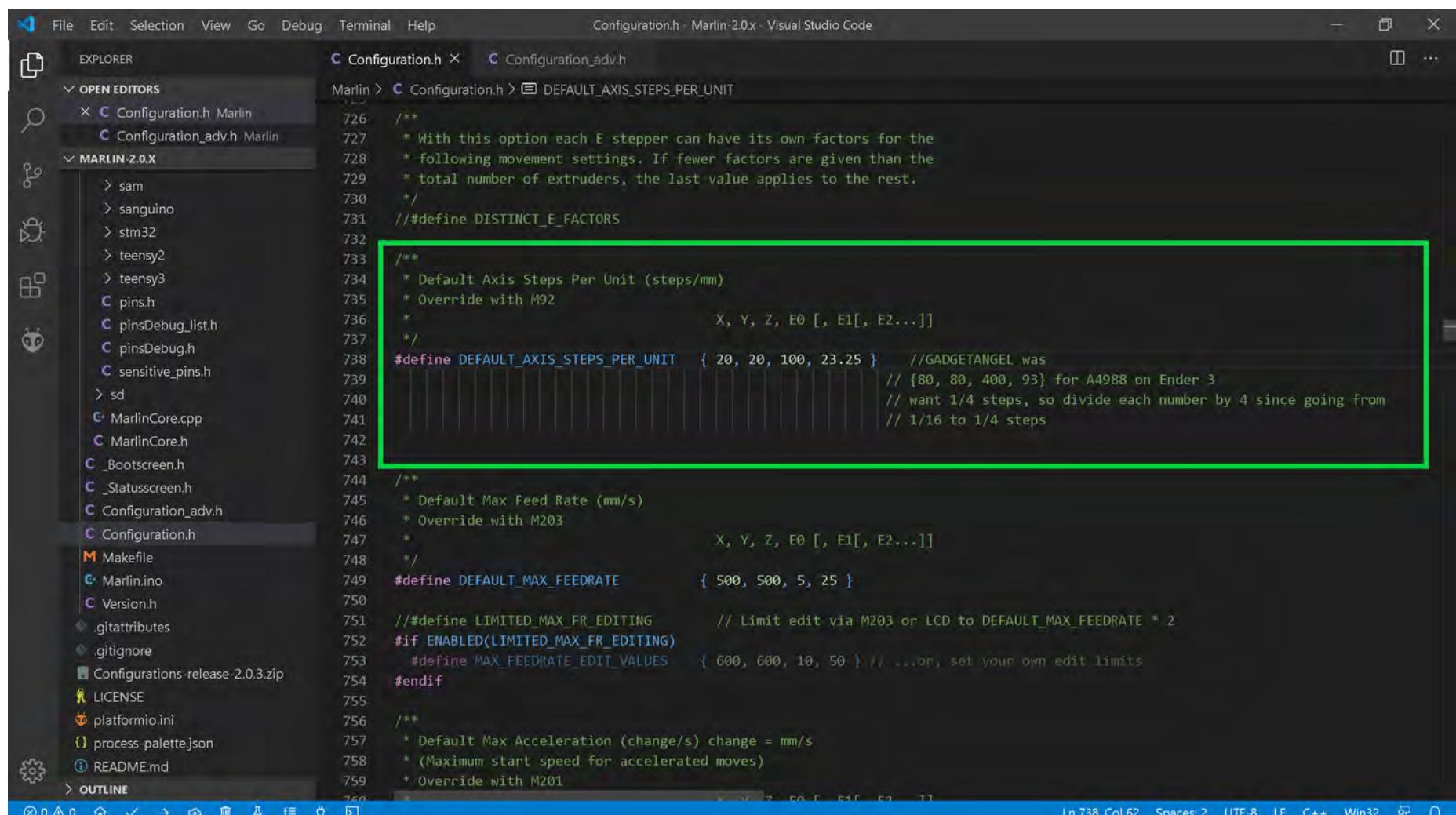
  661 /**
  662 * Stepper Drivers
  663 *
  664 */
  665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  667 *
  668 * A4988 is assumed for unspecified drivers.
  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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
  677 */
  678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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 TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, as seen in the GREEN box below.



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

```

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 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742
743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

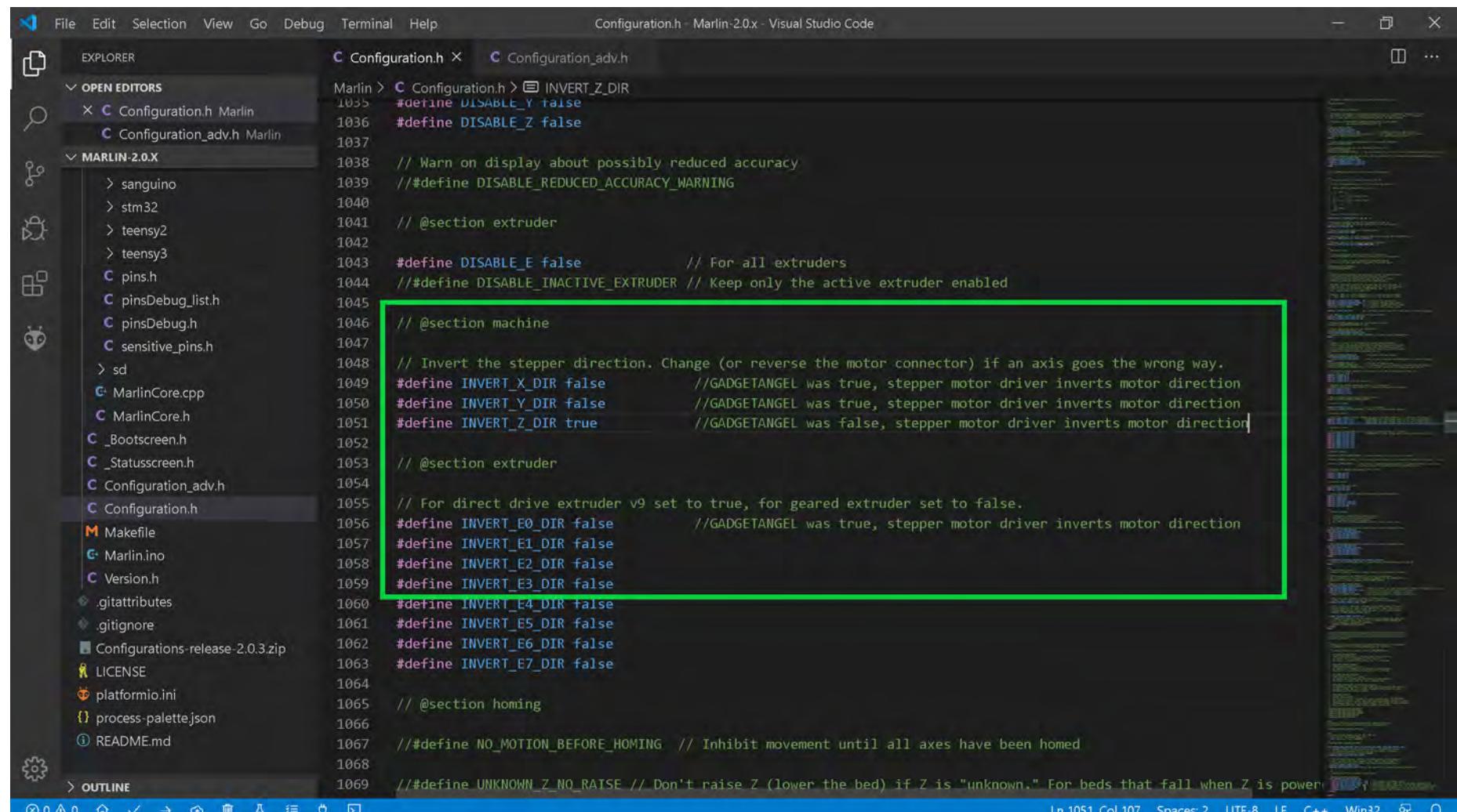
750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
754#endif
755
756 /**
757 * Default Max Acceleration (change/s) change = mm/s
758 * (Maximum start speed for accelerated moves)
759 * Override with M201
760 */

```

- 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



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

```

Marlin > Configuration.h > INVERT_Z_DIR
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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

A green rectangular box highlights the line of code: `#define INVERT_Z_DIR true`. This line is part of a block of code that defines the direction for each axis (X, Y, Z, E0-E7). The code is located in the 'Machine' section of the configuration file.

- 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.

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

EXPLORER OPEN EDITORS Marlin > Configuration.h X DRIVER_TYPE

pins_BTT_SKR_PRO_V1_1.h Configuration.h 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

```
STV32F103VE_L1_01 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
SAMDS1_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```

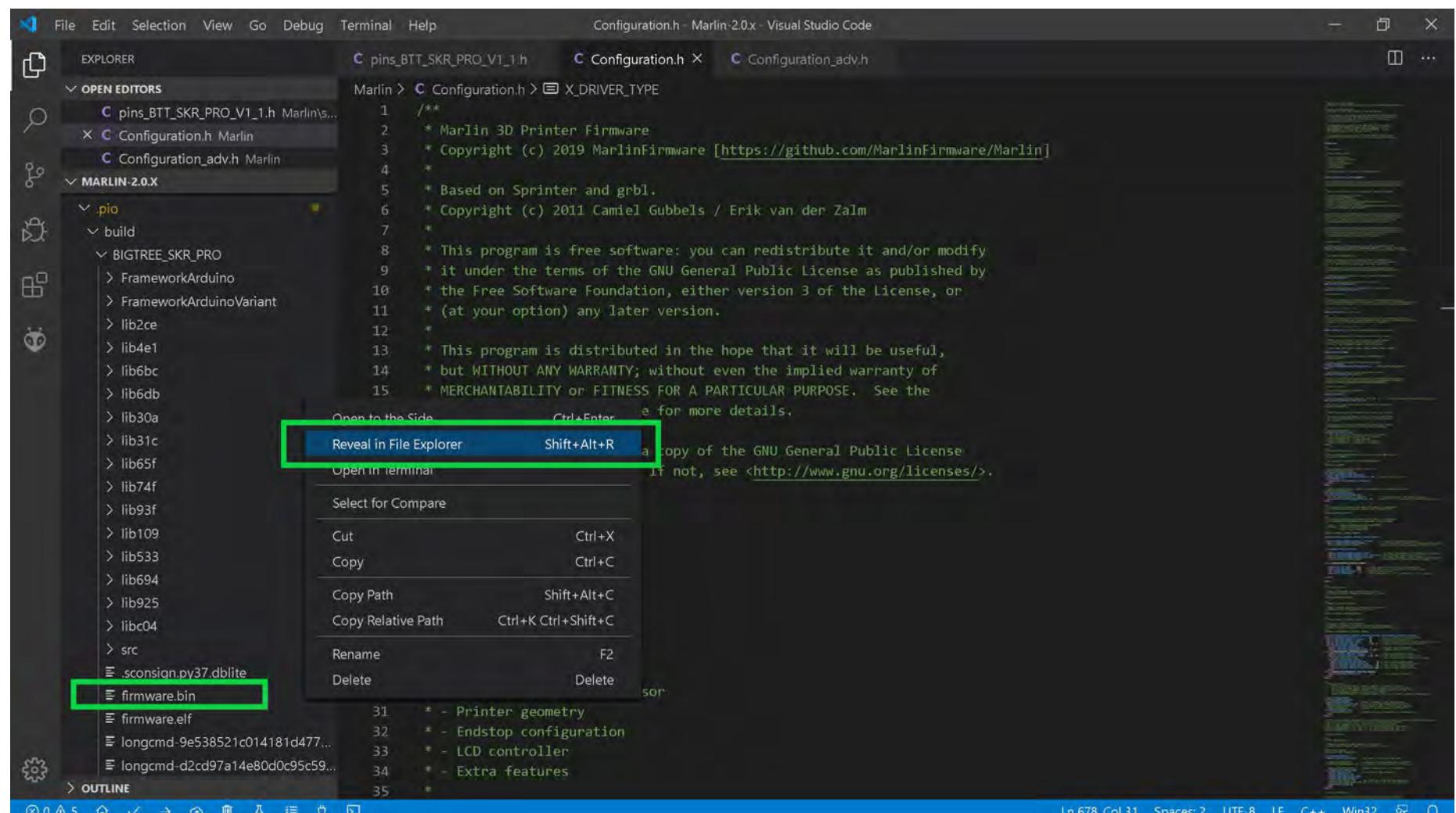
Terminal will be reused by tasks, press any key to close it.

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 One Time Programming (OTP) Mode

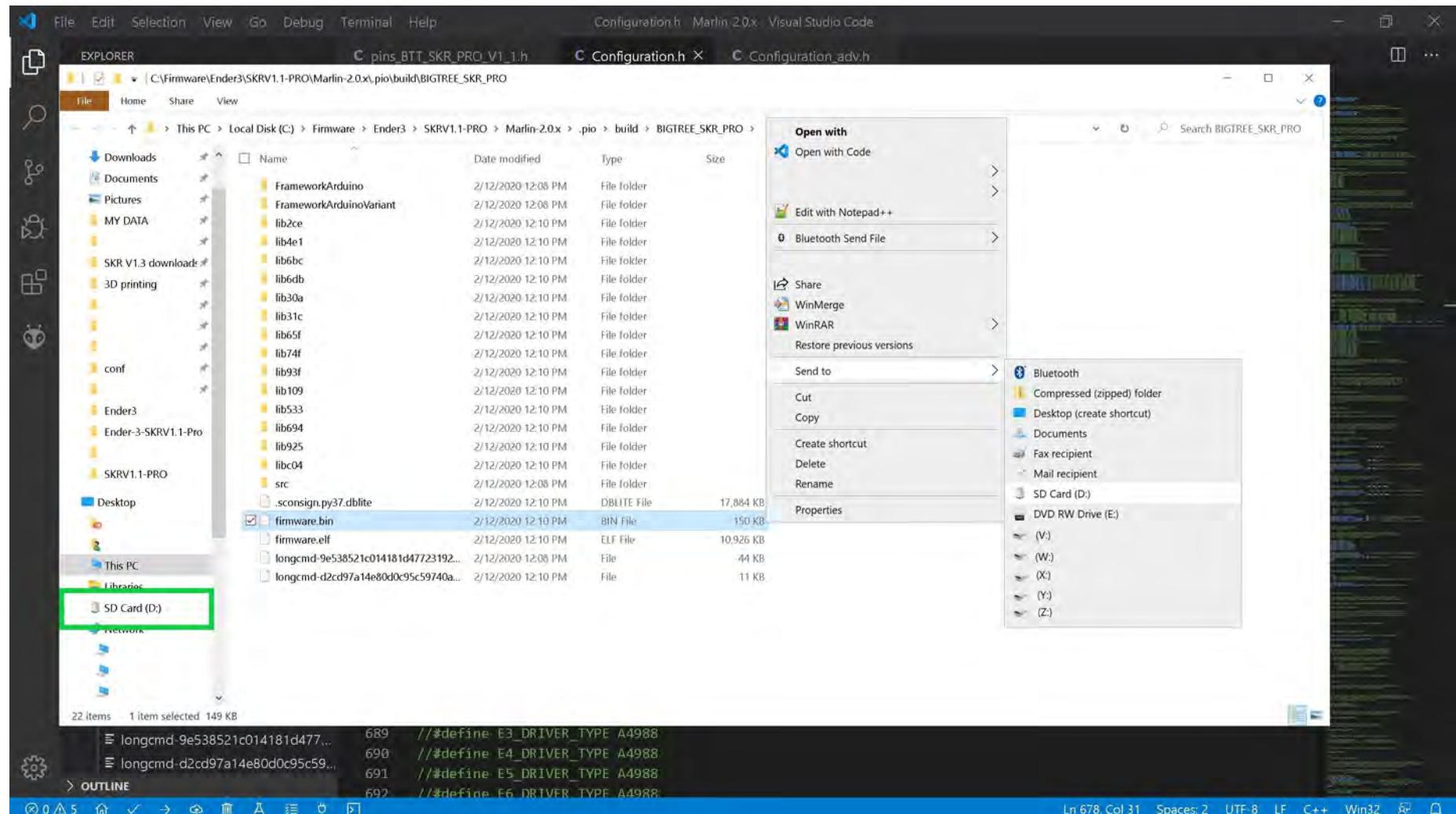
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



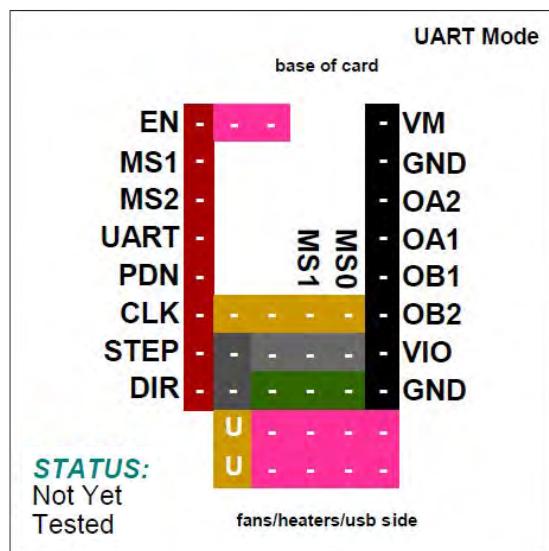
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



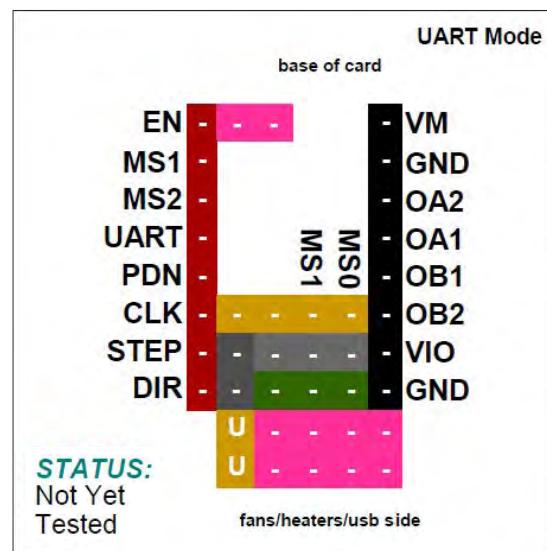
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

BIQU TMC2208 V3.0UART Mode

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

See the next page for further information.

Driver Chip  TMC2208 UART Mode Maximum 256 Subdivision 35V DC 2A (peak)	Steps are set inside of your Firmware
Driving Current Calculation Formula R_S (Typical Sense Resistor)= 0.11Ω	$I_{MAX} = V_{ref}$ See Appendix B #4. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$
	$V_{ref} = I_{MAX}$ See Appendix B #4. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

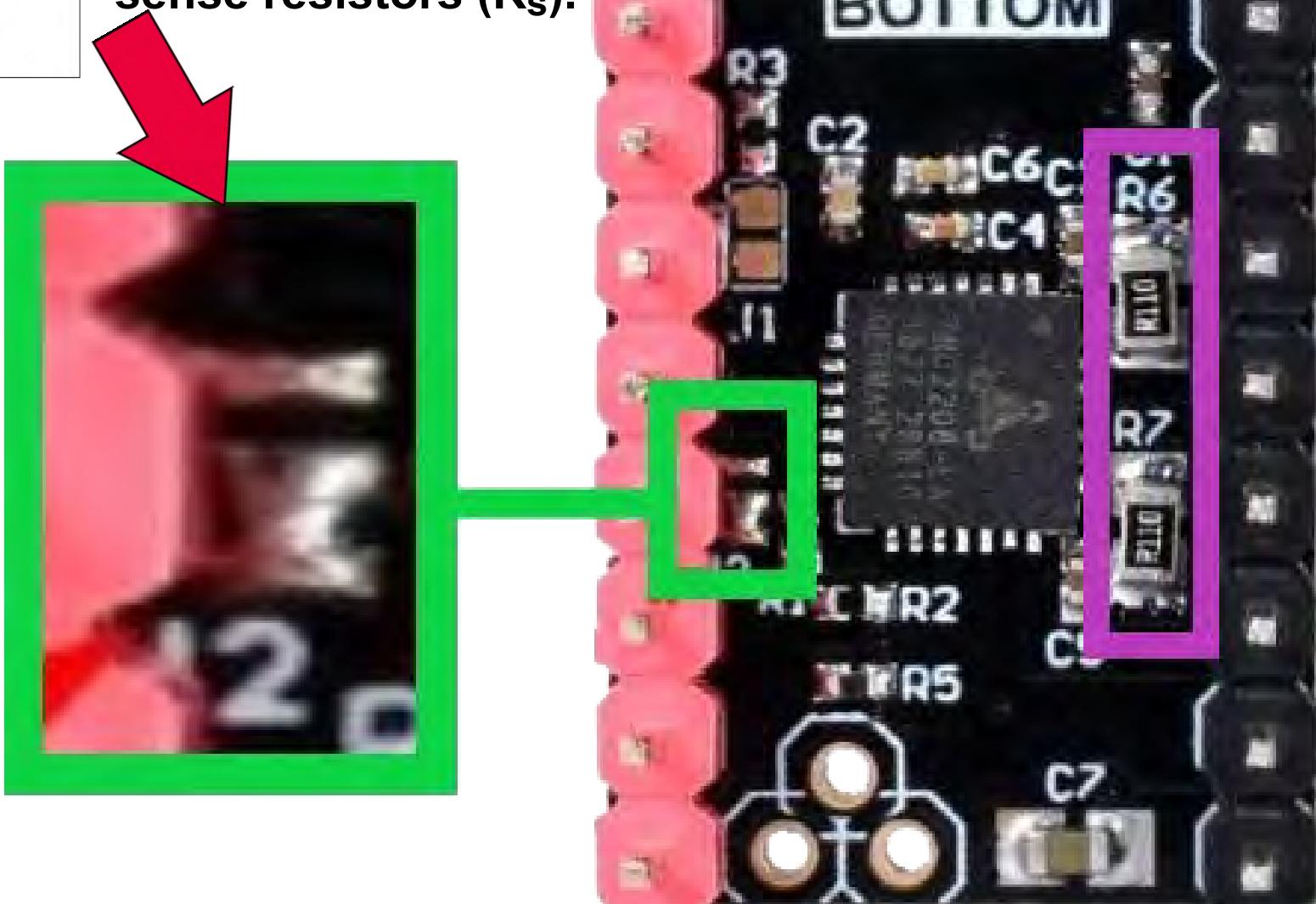


MOST Biqu TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

BIQU TMC2208 V3.0

UART Mode

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, as seen in the **GREEN box** below. The **PURPLE box** shows the location of the current sense resistors (R_s).



UART Mode**UART Mode**

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

$R_s = R_{050}$ is 0.05 Ohms

$R_s = R_{062}$ is 0.062 Ohms

$R_s = R_{068}$ is 0.068 Ohms

$R_s = R_{075}$ is 0.075 Ohms

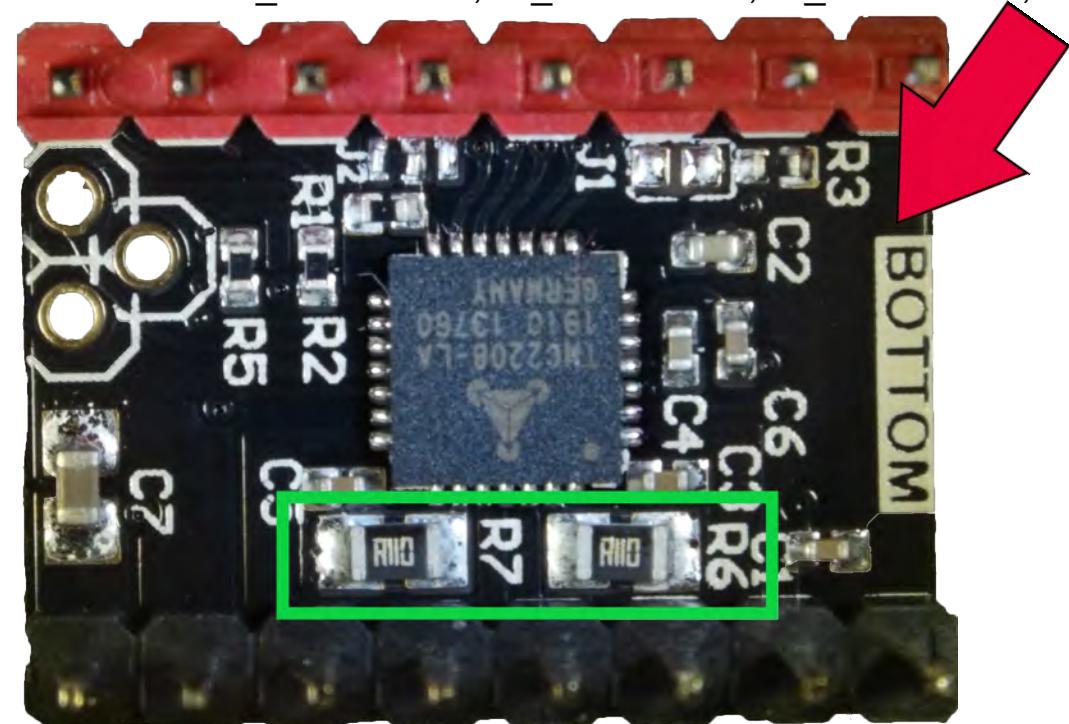
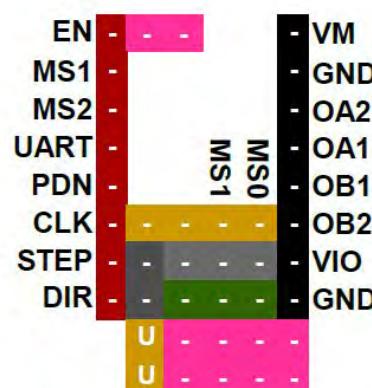
$R_s = R_{100}$ is 0.1 Ohms

$R_s = R_{110}$ is 0.11 Ohms

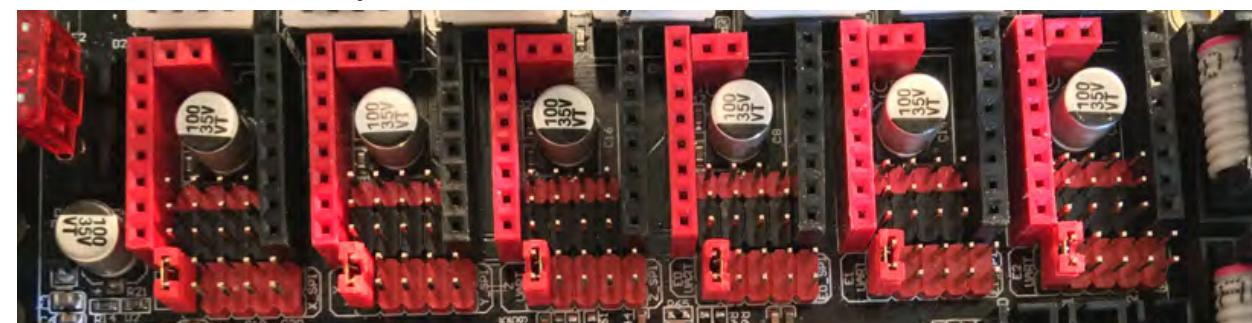
$R_s = R_{150}$ is 0.15 Ohms

$R_s = R_{200}$ is 0.2 Ohms

$R_s = R_{220}$ is 0.22 Ohms

**UART**

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](#), and 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 ("//").

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):** Shows the project structure under MARLIN-2.0.X, including files like LCDprint.cpp, thermistornames.h, and Configuration.h.
- Central Area:** The code editor displays Configuration.h with the following content (lines 661-955 shown):

```
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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
677 */
678 #define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
679 #define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
680 #define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the 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 TMC2208 //GADGETANGEL was commented out and had the 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 box highlights the section starting at line 686, specifically the `#define E0_DRIVER_TYPE TMC2208` line and the subsequent comments.

- 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
X Configuration.h Marlin
C Configuration_adv.h Marlin
MARLIN-2.0.X
> sanguino
> stm32
> 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
Configurations-release-2.0.3.zip
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 🔍

// Warn on display about possibly reduced accuracy
#define DISABLE_REDUCED_ACCURACY_WARNING

// @section extruder
#define DISABLE_E false // For all extruders
#define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled

// @section machine
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false //GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Y_DIR false //GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Z_DIR true //GADGETANGEL was false, stepper motor driver inverts motor direction

// @section extruder
// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false //GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false
#define INVERT_E5_DIR false
#define INVERT_E6_DIR false
#define INVERT_E7_DIR false

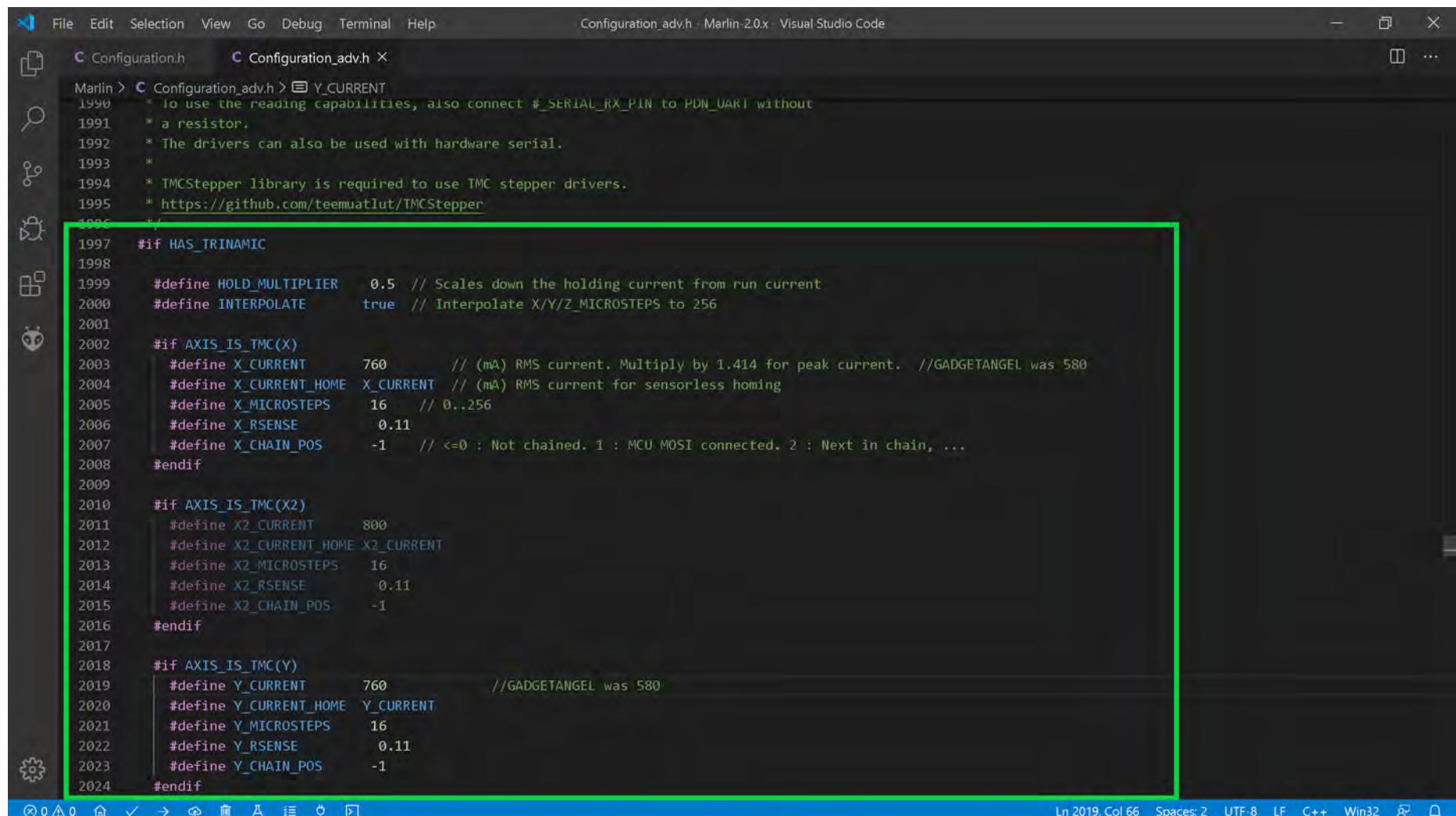
// @section homing
#define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
#define UNKNOWN_Z_NO_RATE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

```

- 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 V_{ref} 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 V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

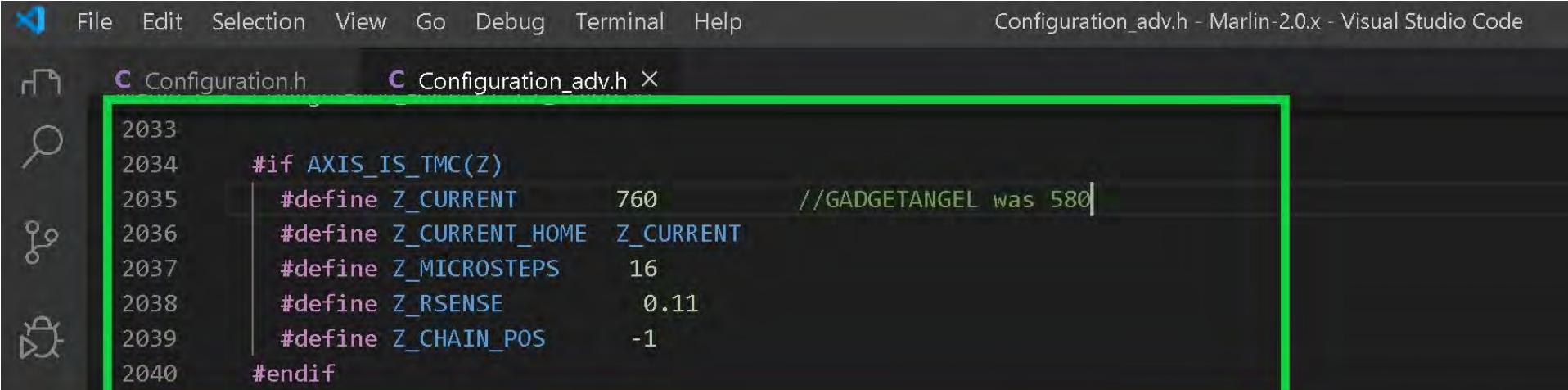
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
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. //GADGETANGEL was 580
2004     #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2005     #define X_MICROSTEPS       16       // 0..256
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      //GADGETANGEL was 580
2020     #define Y_CURRENT_HOME    Y_CURRENT
2021     #define Y_MICROSTEPS       16
2022     #define Y_RSENSE            0.11
2023     #define Y_CHAIN_POS         -1
2024 #endif

```

- 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 V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.11. Ensure "E0_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".



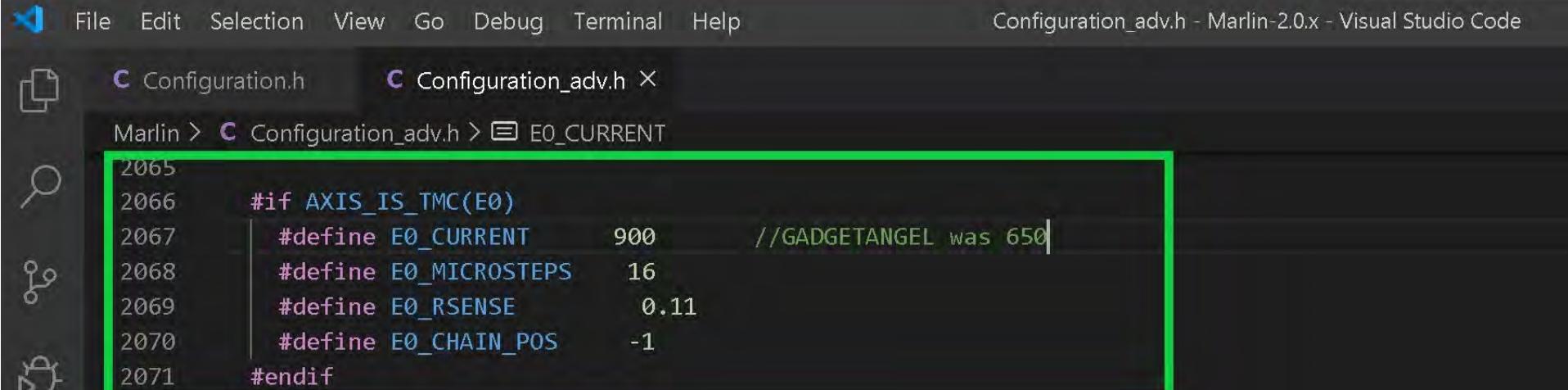
```

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

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760           //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME Z_CURRENT
2037   #define Z_MICROSTEPS    16
2038   #define Z_RSENSE        0.11
2039   #define Z_CHAIN_POS     -1
2040 #endif

```



```

File Edit Selection View Go Debug Terminal Help
Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > E0_CURRENT

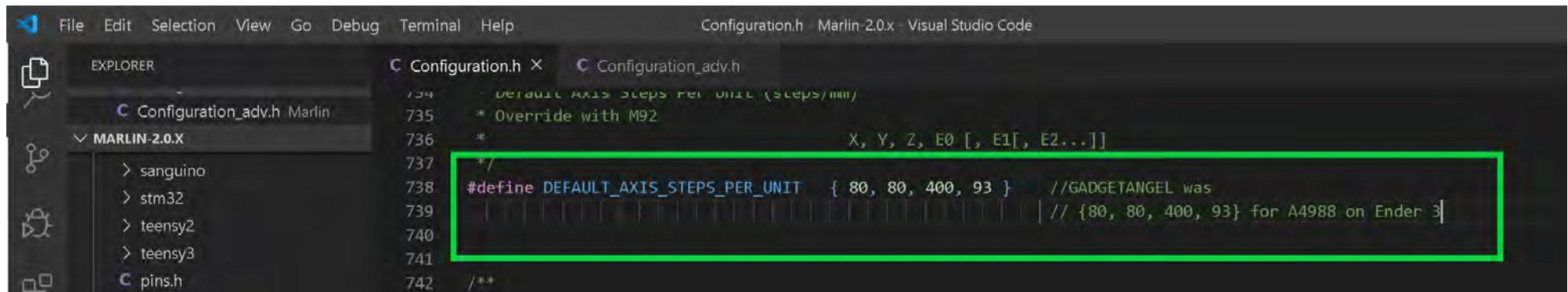
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900           //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- Go to the next page.

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

- 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 Configuration.h X Configuration_adv.h

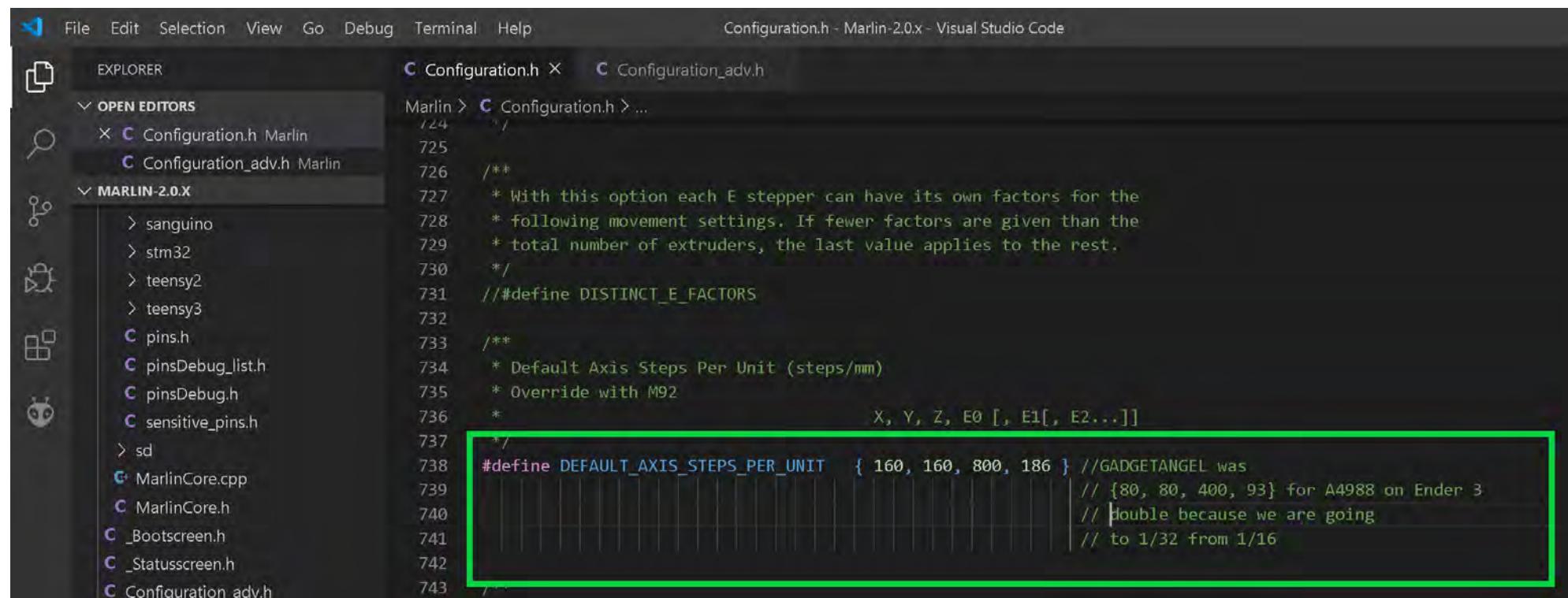
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

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 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */

```

- 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 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

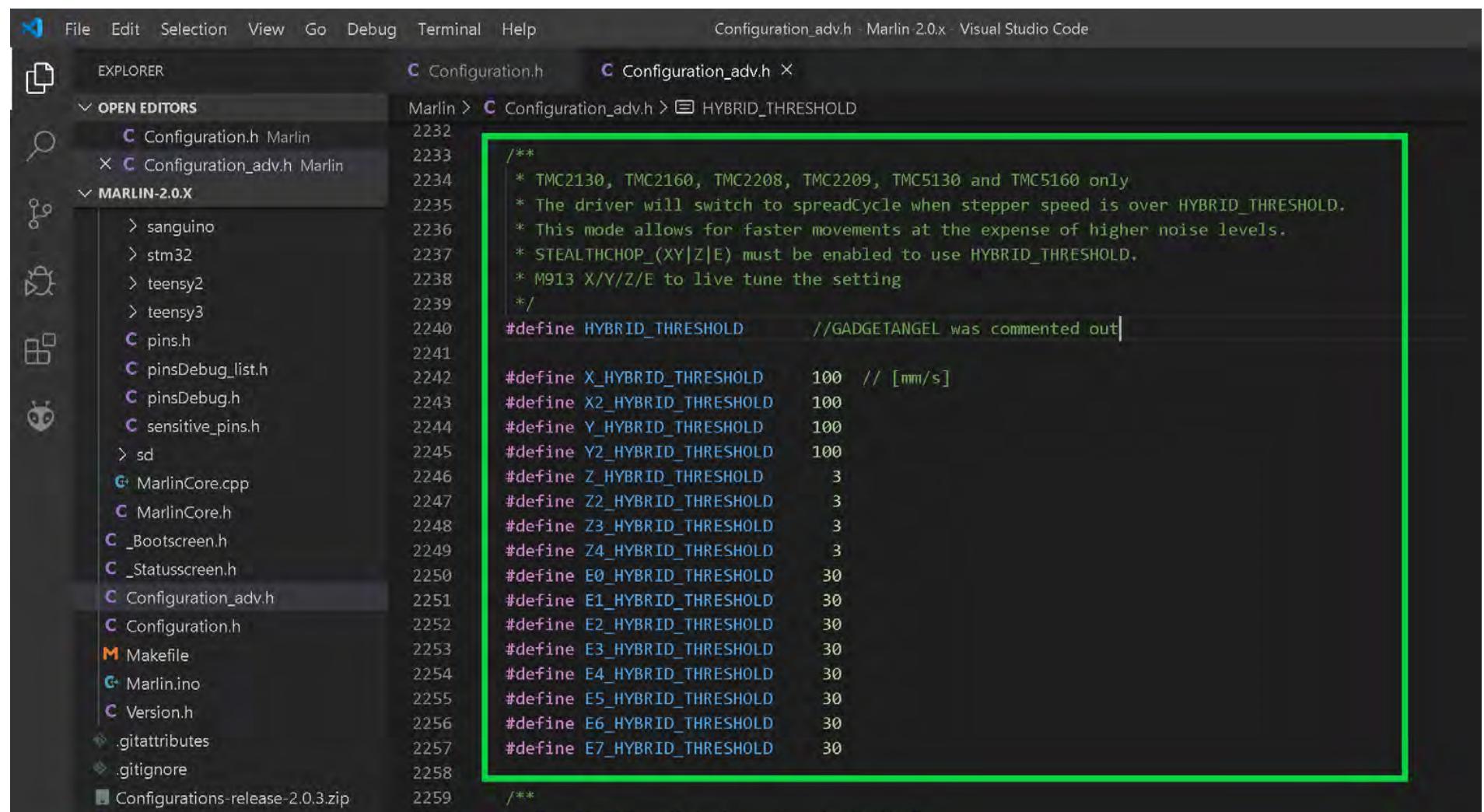
EXPLORER Configuration.h Configuration_adv.h ×
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
Configuration.h Marlin Configuration_adv.h Marlin
Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
> sanguino
> stm32
> teensy2
> teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
> sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197 /**
2198 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2199 * or with the help of an example included in the library.
2200 * Provided parameter sets are
2201 * CHOPPER_DEFAULT_12V
2202 * CHOPPER_DEFAULT_19V
2203 * CHOPPER_DEFAULT_24V
2204 * CHOPPER_DEFAULT_36V
2205 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2206 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2207 *
2208 * Define your own with
2209 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2210 */
2211 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2212 /**
2213 */

```

- 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.



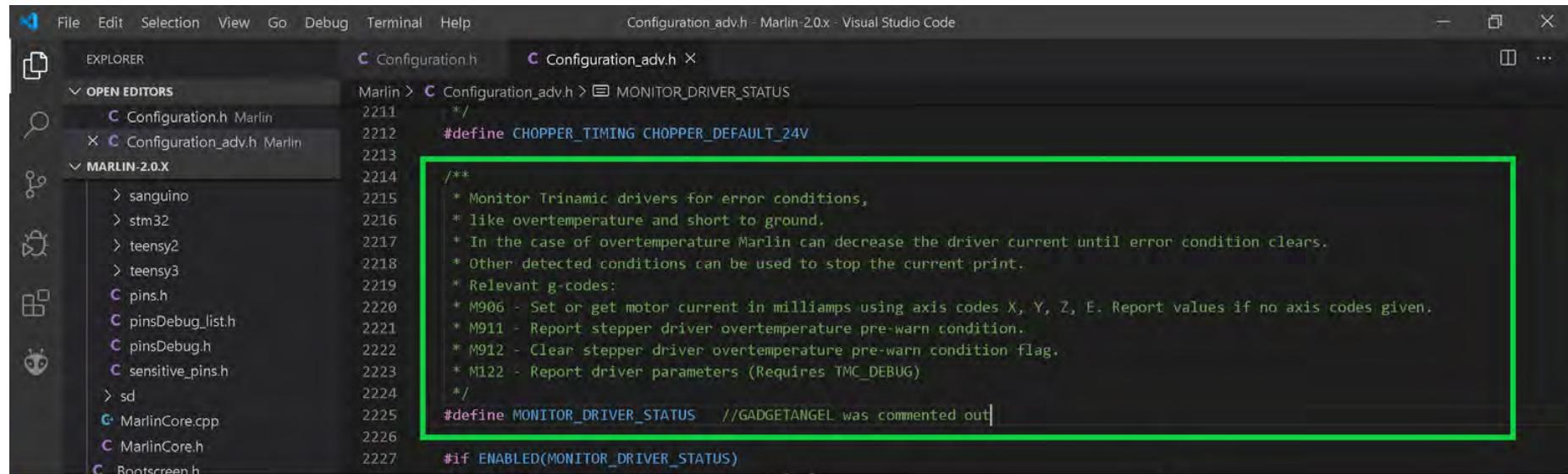
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
- Explorer:** Shows the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X' (including sanguino, stm32, 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, Configurations-release-2.0.3.zip).
- Code Editor:** Displays the content of Configuration_adv.h. A specific line is highlighted with a green border:


```
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3797
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3897
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3997
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4189
4190
4191
4192
4193
4194
4195
4196
4197
4197
4198
4199
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
42
```

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



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

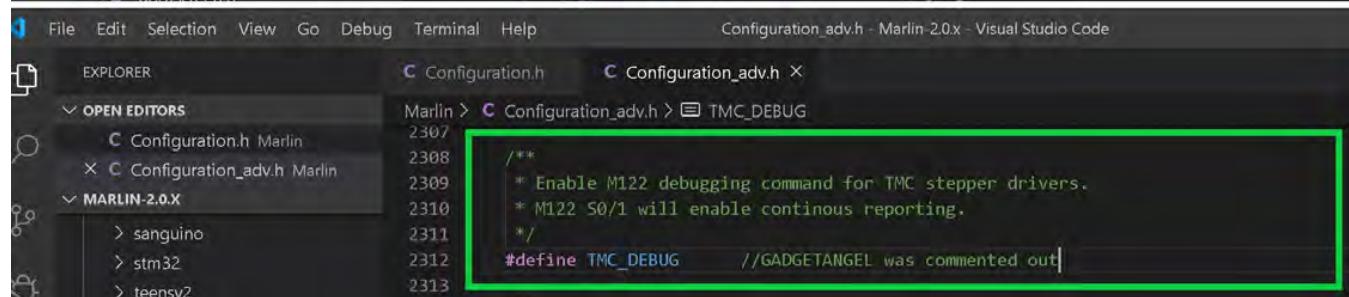
EXPLORER Configuration.h Configuration_adv.h

OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

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 millamps 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 //GADGETANGEL 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

EXPLORER Configuration.h Configuration_adv.h

OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG

```

2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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.

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

EXPLORER OPEN EDITORS MARLIN-2.0.X

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 OUTPUT DEBUG CONSOLE TERMINAL

1: Task - Build

BIGTREETECH_11_0	IGNORED	
BIGTREETECH_SKR_PRO	SUCCESS	00:02:31.294
BIGTREETECH_11_2	IGNORED	
BIGTREETECH_BTT002	IGNORED	
teensy31	IGNORED	
teensy35	IGNORED	
esp32	IGNORED	
linux_native	IGNORED	
SAMD51_grandcentral_m4	IGNORED	
rumba32_f446ve	IGNORED	
mks_rumba32	IGNORED	
INCLUDE_TREE	IGNORED	

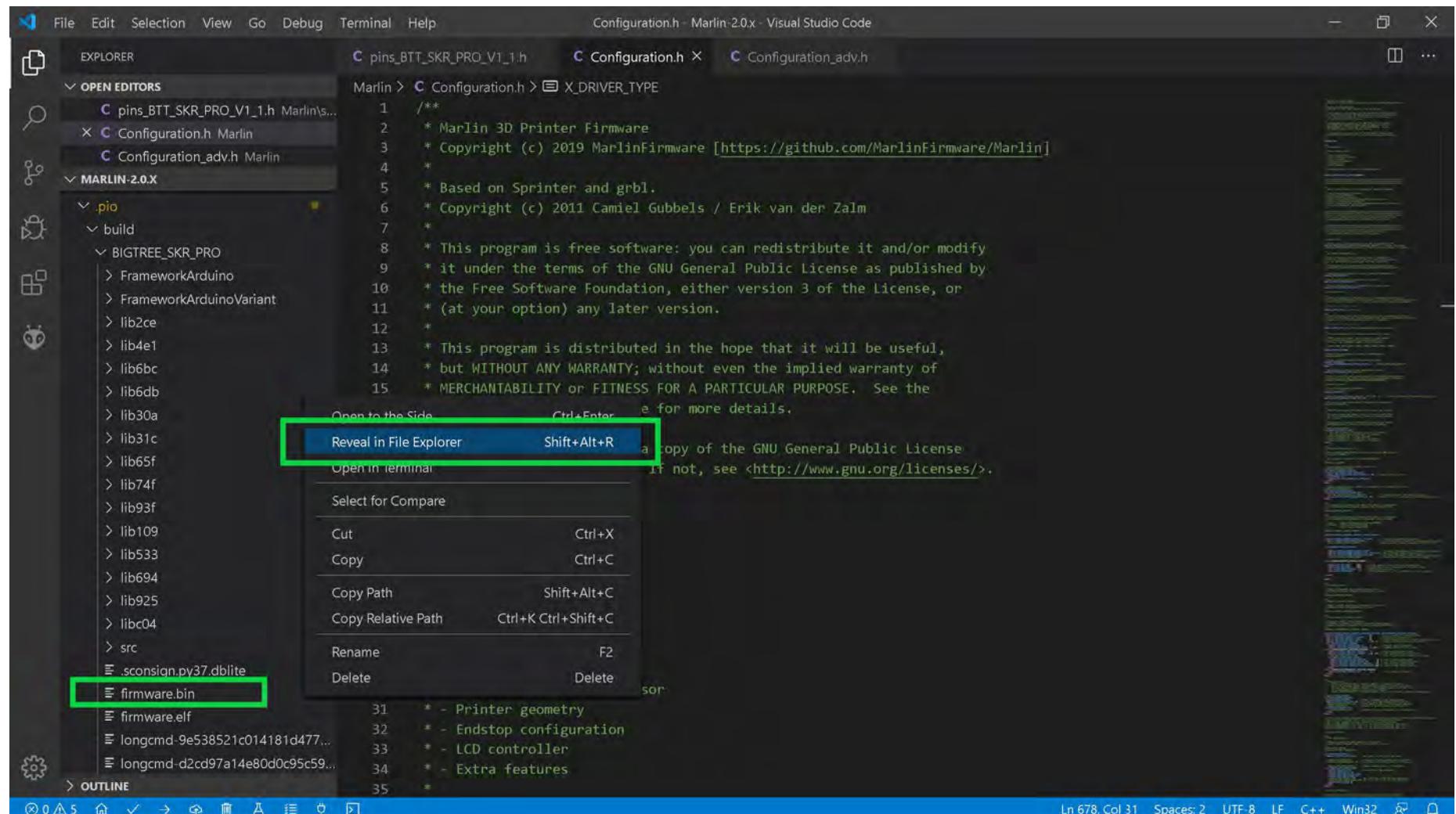
```
===== 1 succeeded in 00:02:31.294 =====
```

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 UART Mode

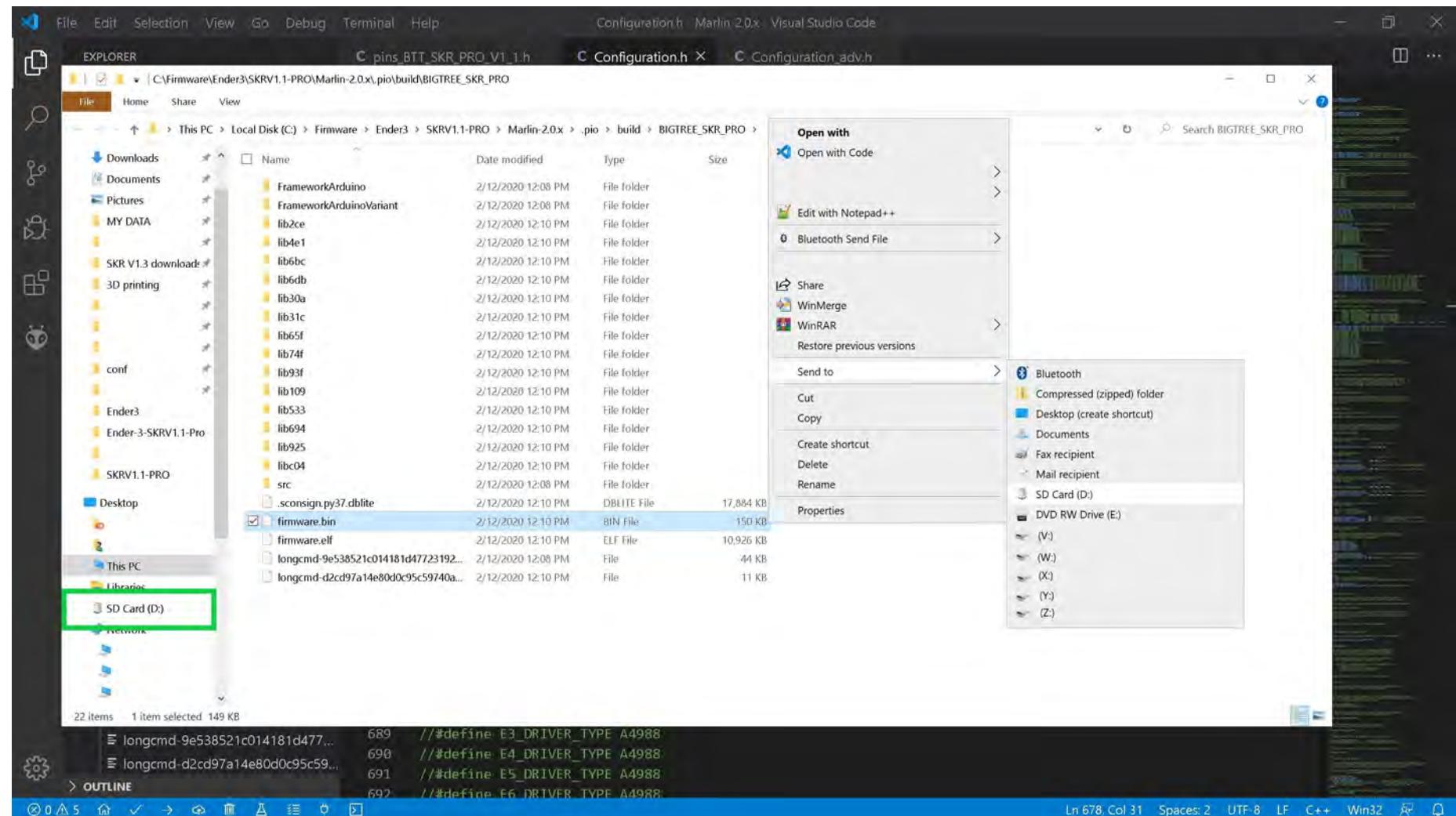
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



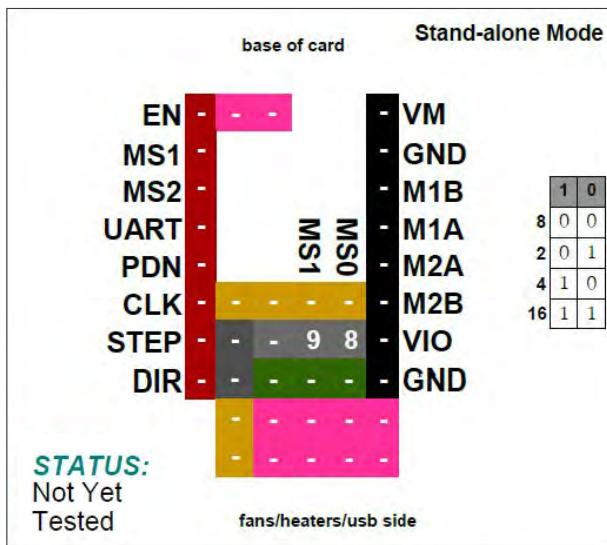
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



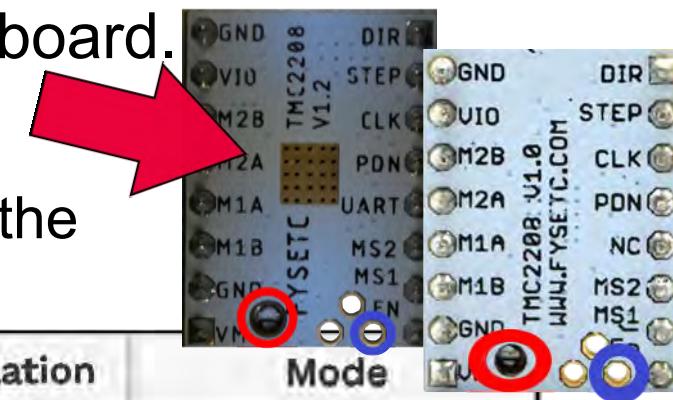
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



FYSETC TMC2208 V1.2

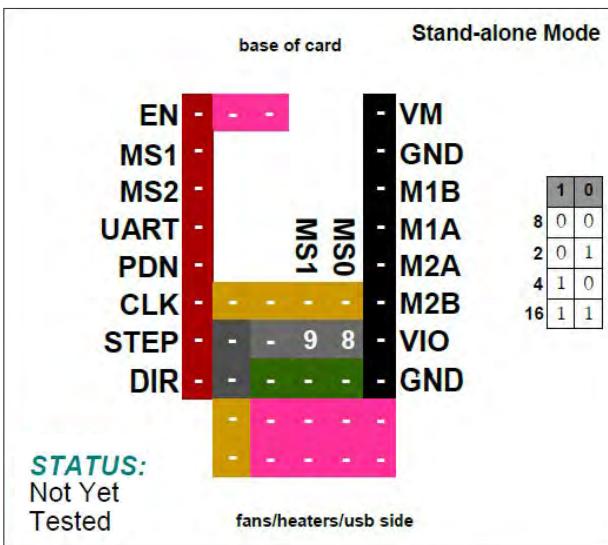
Stand-alone Mode

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Driver Chip	MS1	MS0	Steps	Interpolation	Mode
FYSETC TMC2208 Stand Alone Mode Maximum 16 Subdivision 35V DC 2A (peak)	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
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.11Ω	$I_{MAX} = V_{ref} * 0.9286$ See Appendix B #3. Use 50% to 90% as shown below: $I_{MAX} = (V_{ref} * 0.9286) * 0.90$			$V_{ref} = I_{MAX} * 1.0769$ See Appendix B #3. Use 50% to 90% as shown below: $V_{ref} = (I_{MAX} * 1.0769) * 0.90$	

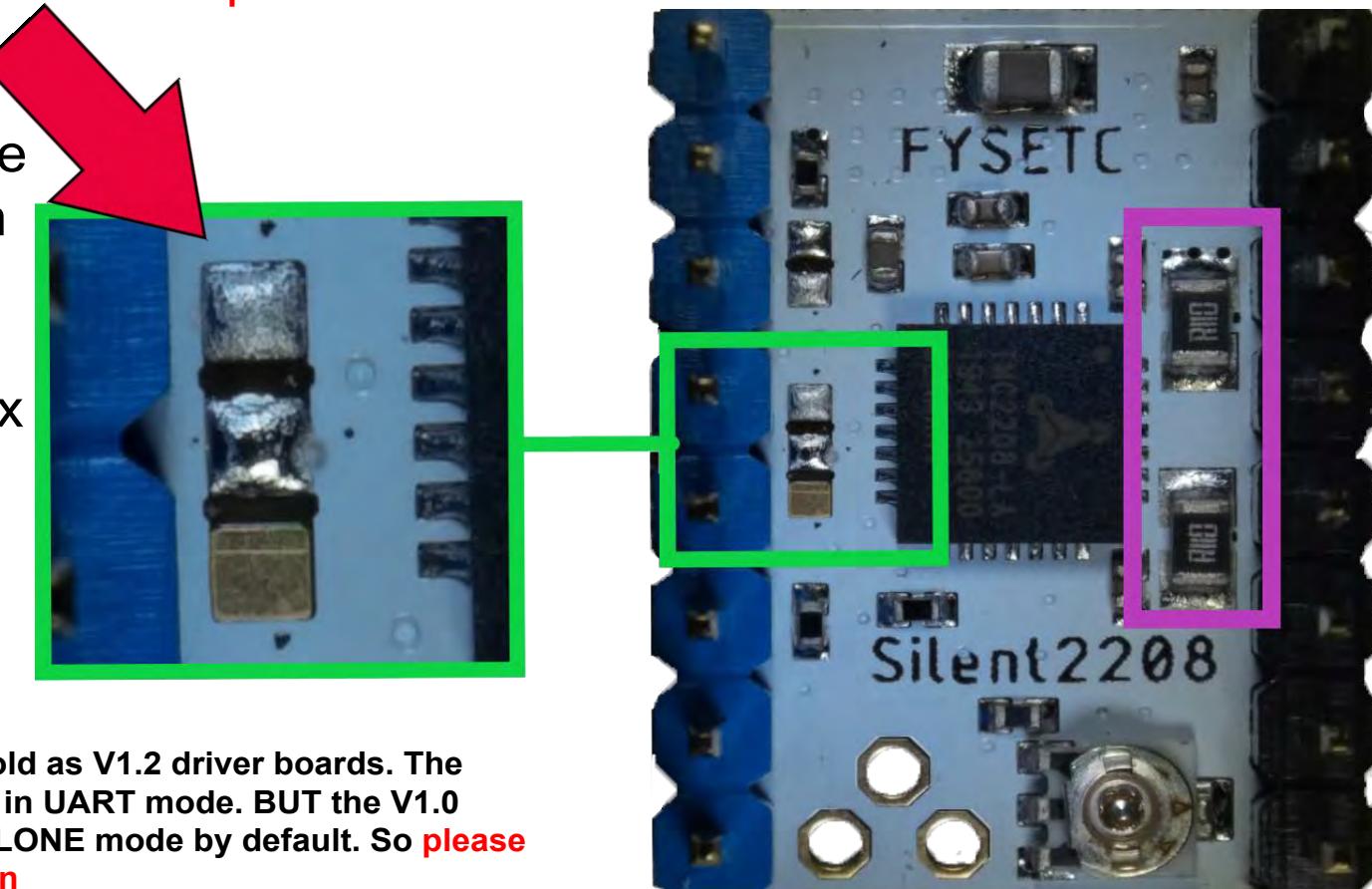


FYSETC TMC2208 V1.2

Stand-alone Mode

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 on all (three or two) "J2" pads to obtain Standalone Mode for the FYSETC TMC2208 V1.X, as seen in the **GREEN** box below. The **PURPLE** box shows the location of the current sense resistors (R_s).



MOST FYSETC TMC2208 boards 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!!**

FYSETC TMC2208 V1.2 for Standalone Mode

FYSETC TMC2208 V1.2

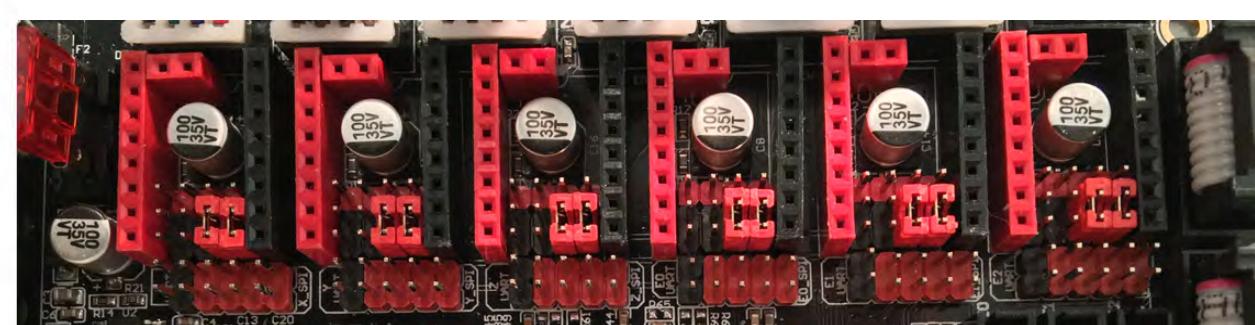
Stand-alone Mode

Stand-alone Mode

1 / 8Interpolation: **1/256**

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	-	-	M2B
STEP	-	-	9	VIO
DIR	-	9	8	GND

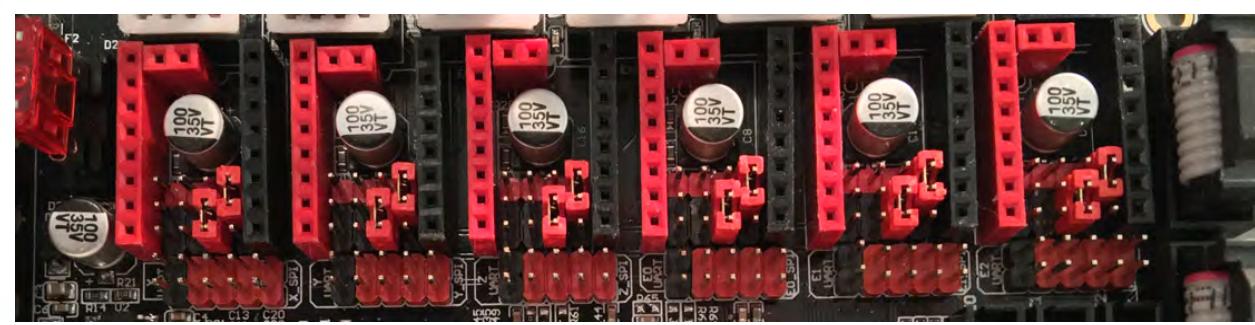


Stand-alone Mode

1 / 2Interpolation: **1/256**

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	-	8	M2B
STEP	-	-	9	VIO
DIR	-	9	-	GND



FYSETC TMC2208 V1.2

Stand-alone Mode

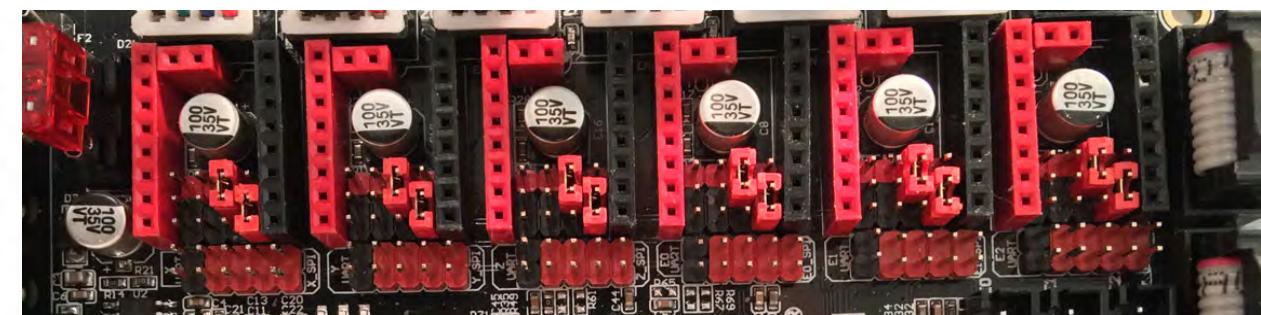
Stand-alone Mode

1 / 4

Interpolation: 1/256

StealthChop

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	M1B
UART	-	-	M1A
PDN	MS1	MS0	M2A
CLK	9	-	M2B
STEP	-	9 8	VIO
DIR	-	-	GND



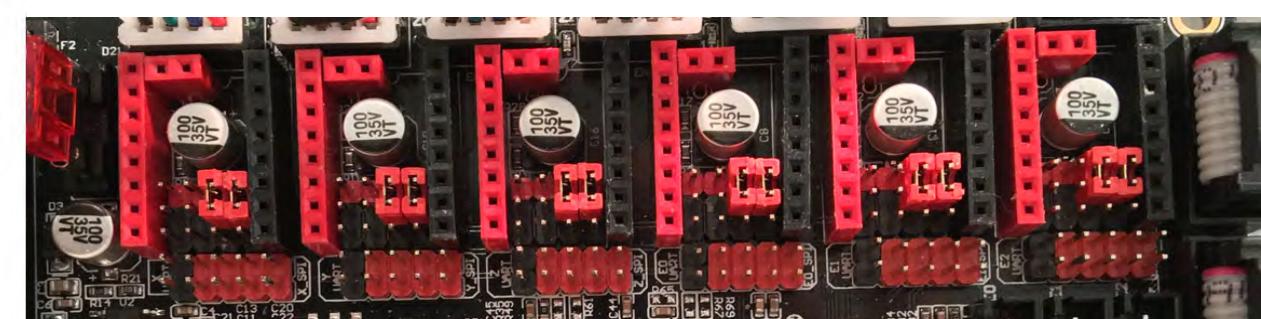
Stand-alone Mode

1 / 16

Interpolation: 1/256

StealthChop

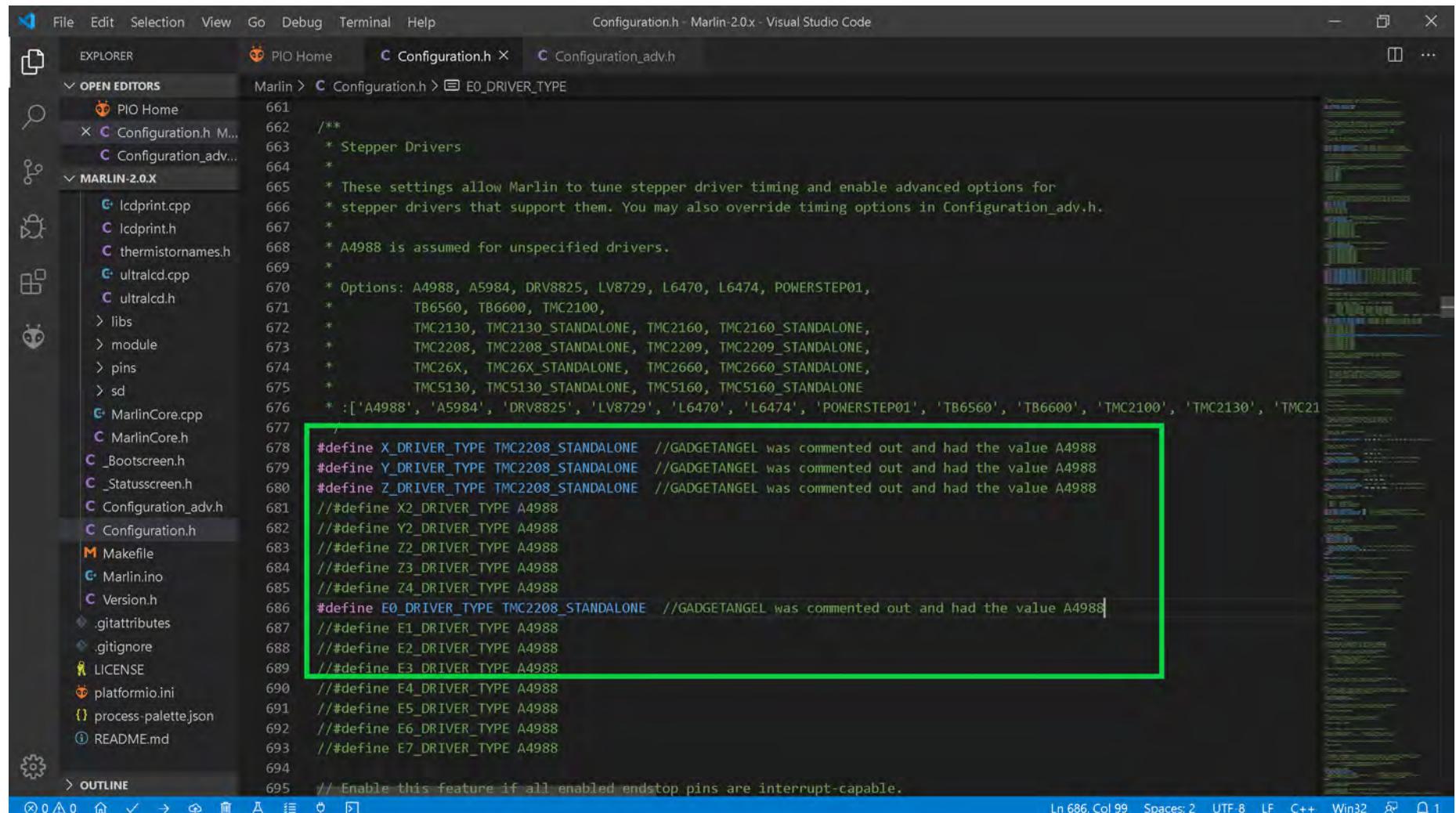
EN	-	-	VM
MS1	-	-	GND
MS2	-	-	M1B
UART	-	-	M1A
PDN	MS1	MS0	M2A
CLK	9	8	M2B
STEP	-	9 8	VIO
DIR	-	-	GND



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

NOTE: Go to Appendix C, and 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 ("//").



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 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208',
676 * TMC26X', 'TMC5130'],
677 */
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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.

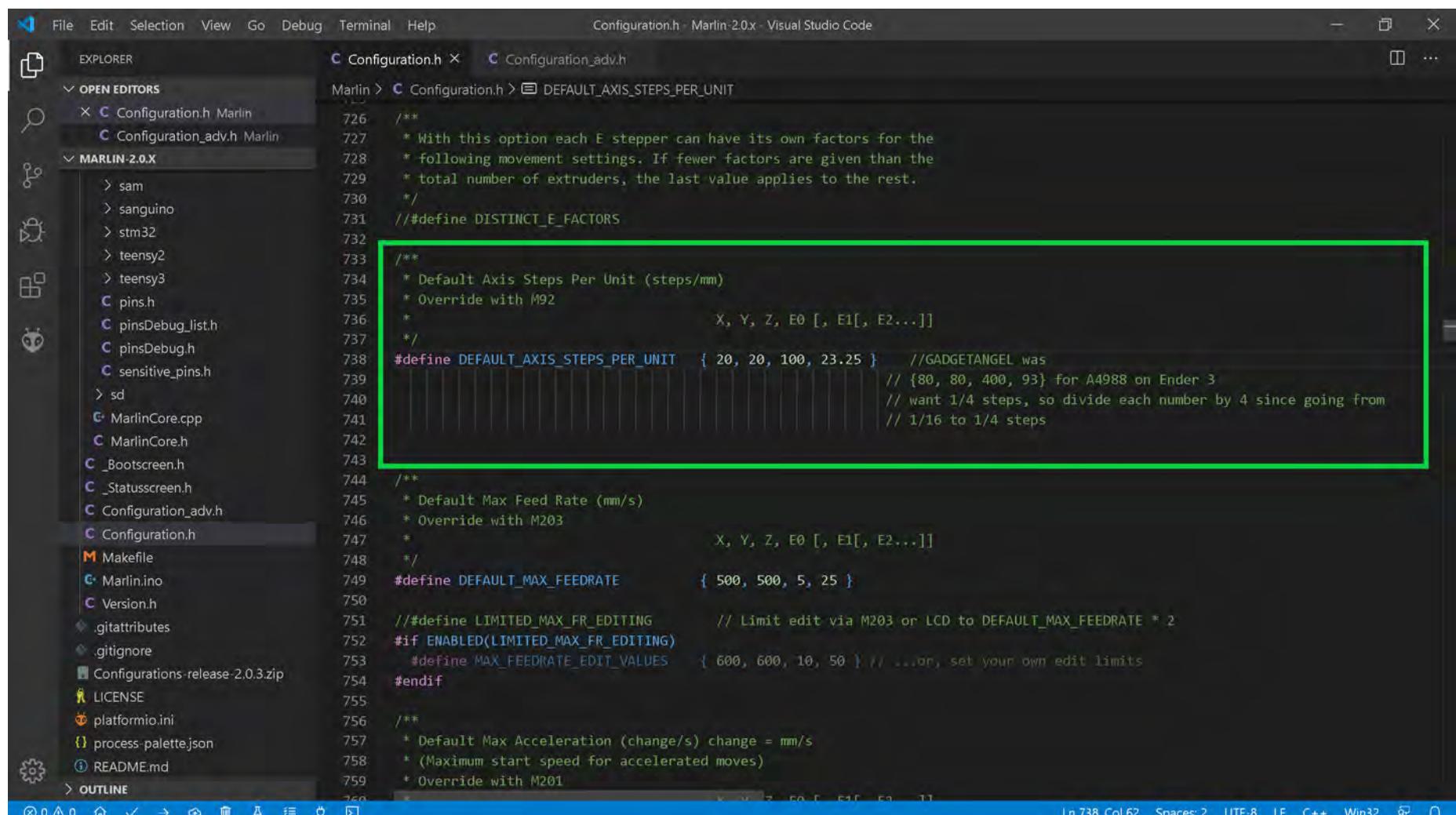
```

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

- Go to the next page.

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

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The left sidebar shows the project structure under 'MARLIN-2.0.X'. The main editor window displays the Configuration.h file. A green rectangular box highlights the following code block:

```

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 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742

743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
754#endif

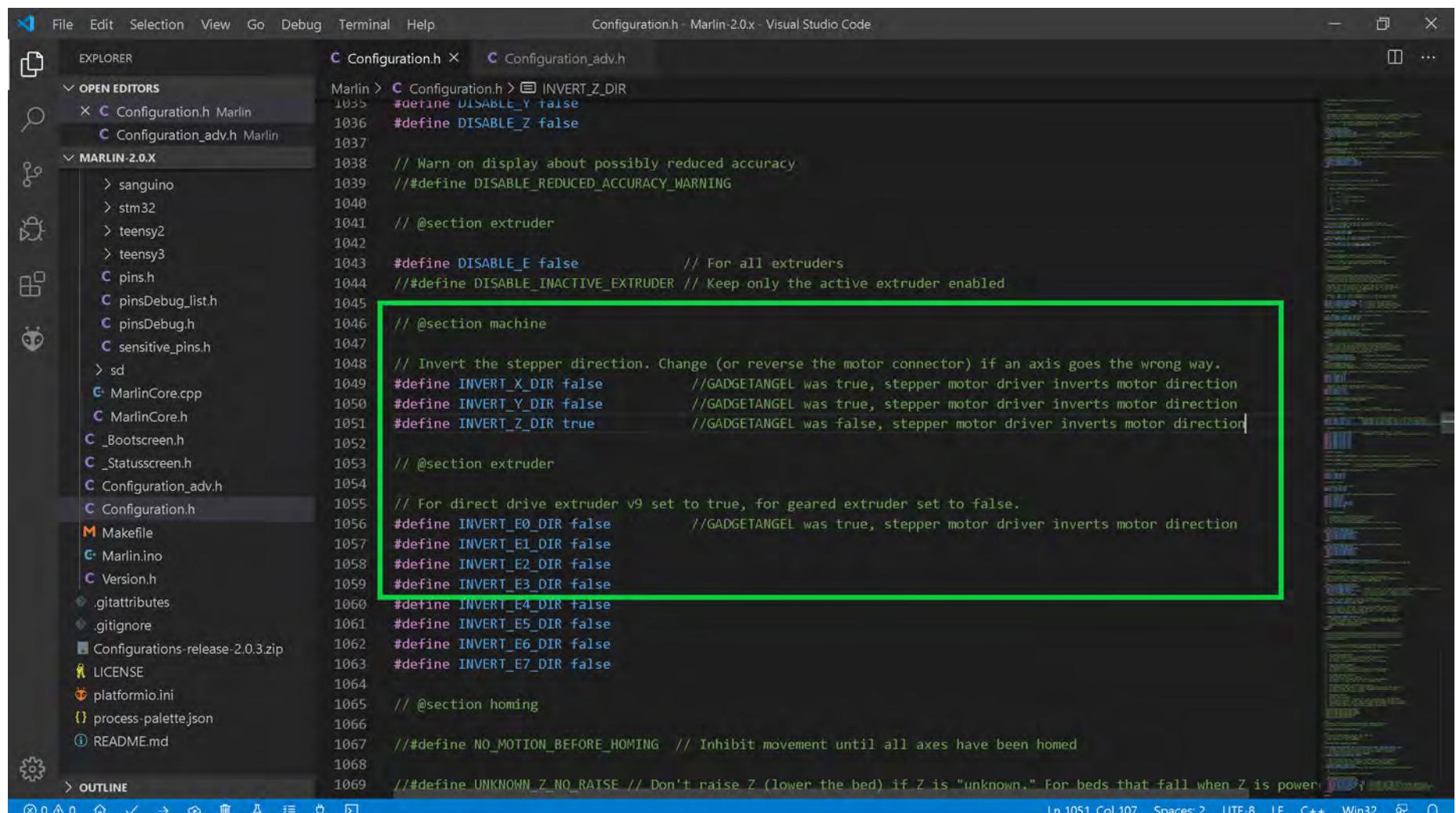
755 /**
756 * Default Max Acceleration (change/s) change = mm/s
757 * (Maximum start speed for accelerated moves)
758 *
759 * Override with M201
760 */

```

- 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



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

```

File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  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      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  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     // GADGETANGEL was true, stepper motor driver inverts motor direction
  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
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

Ln 1051, Col 107 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.

The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The Explorer sidebar on the left lists files and folders, including pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and various Marlin Core files. The main editor area displays code for stepper drivers, with comments about tuning options for different driver types. The bottom right features a terminal window showing the results of a build command for different boards. The terminal output includes:

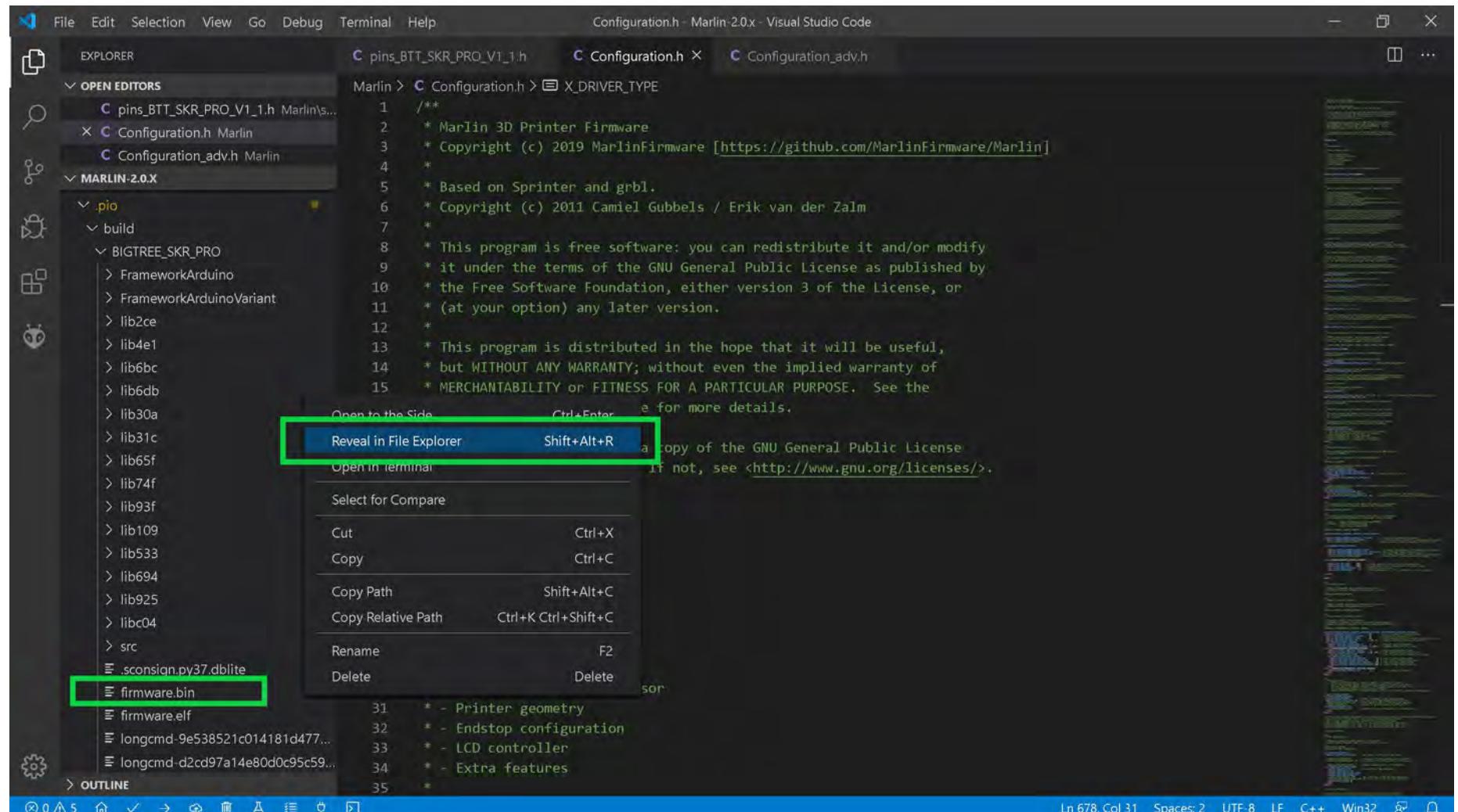
```
1: Task - Build
[...]
BIGTREETECH SKR PRO SUCCESS 00:02:31.294
BIGTREETECH SKR V1.0 IGNORED
[...]
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
```

The terminal output is highlighted with a green box.

- 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

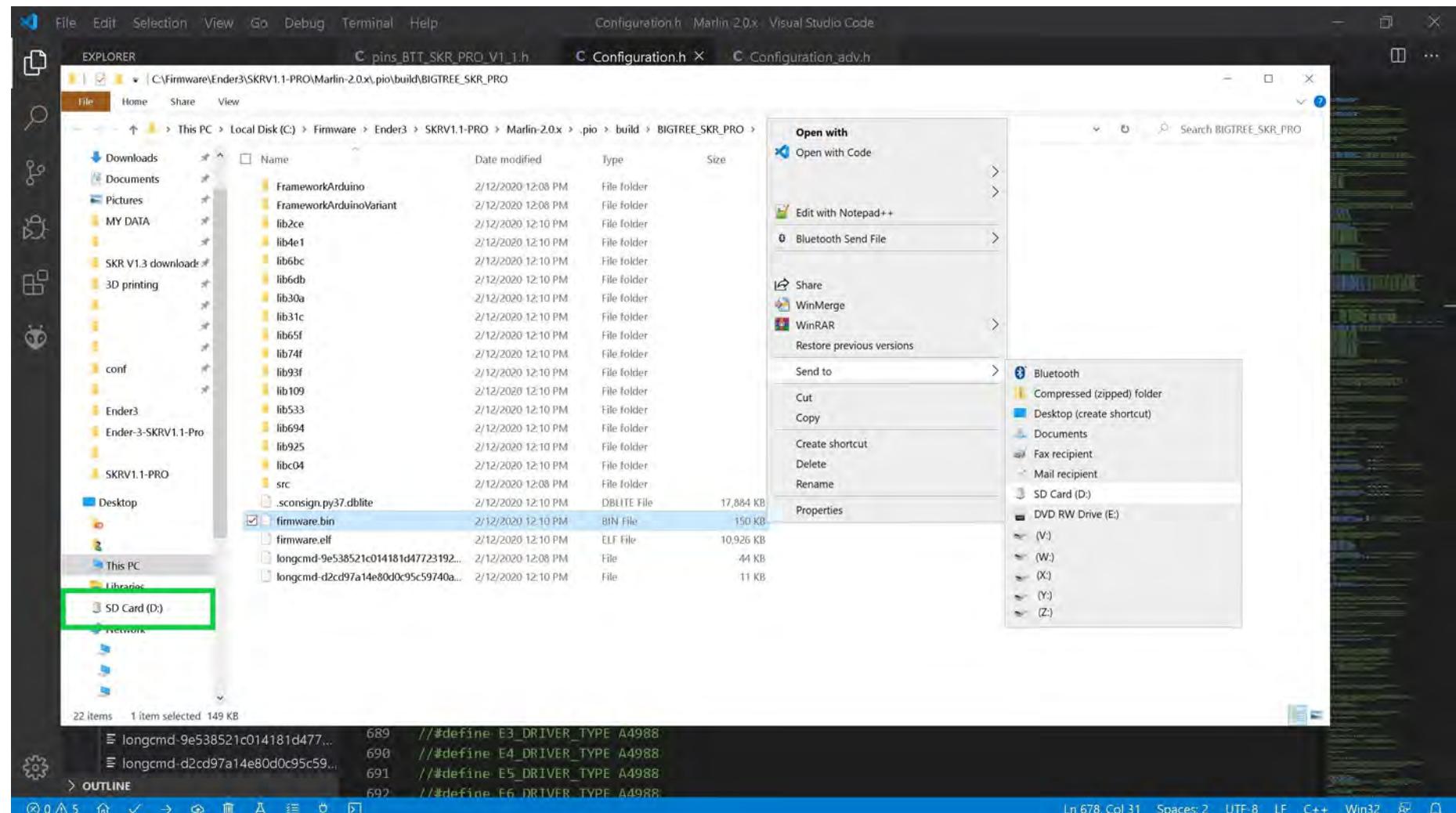
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file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



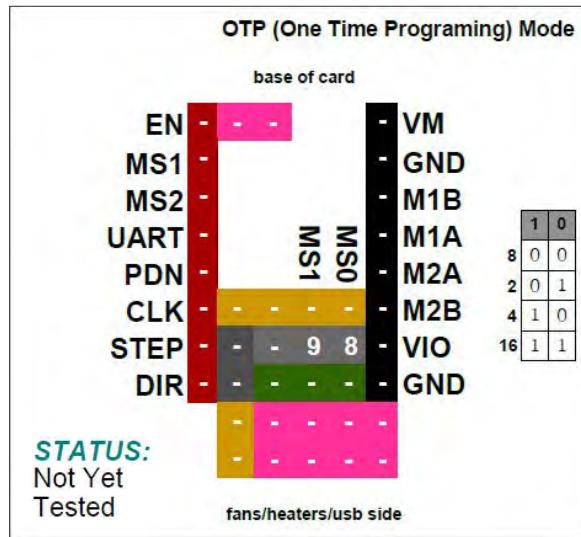
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



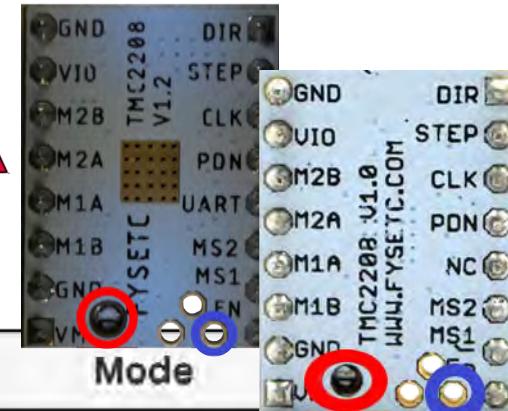
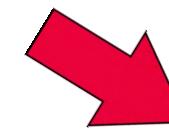
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



FYSETC TMC2208 V1.2

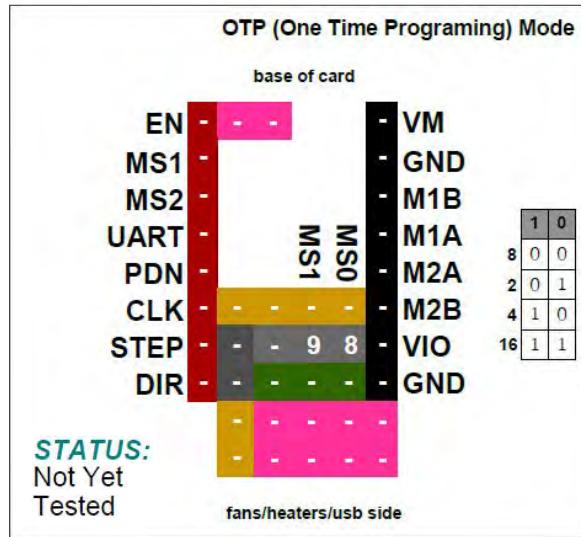
One Time Programming (OTP) Mode

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Driver Chip	MS1	MS0	Steps	Interpolation	Mode
FYSETC TMC2208 OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)	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
Driving Current Calculation Formula R_S (Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}*0.9286$ See Appendix B #3. Use 50% to 90% as shown below: $I_{MAX}=(V_{ref} * 0.9286)*0.90$			$V_{ref}=I_{MAX}*1.0769$ See Appendix B #3. Use 50% to 90% as shown below: $V_{ref}=(I_{MAX} * 1.0769)*0.90$	



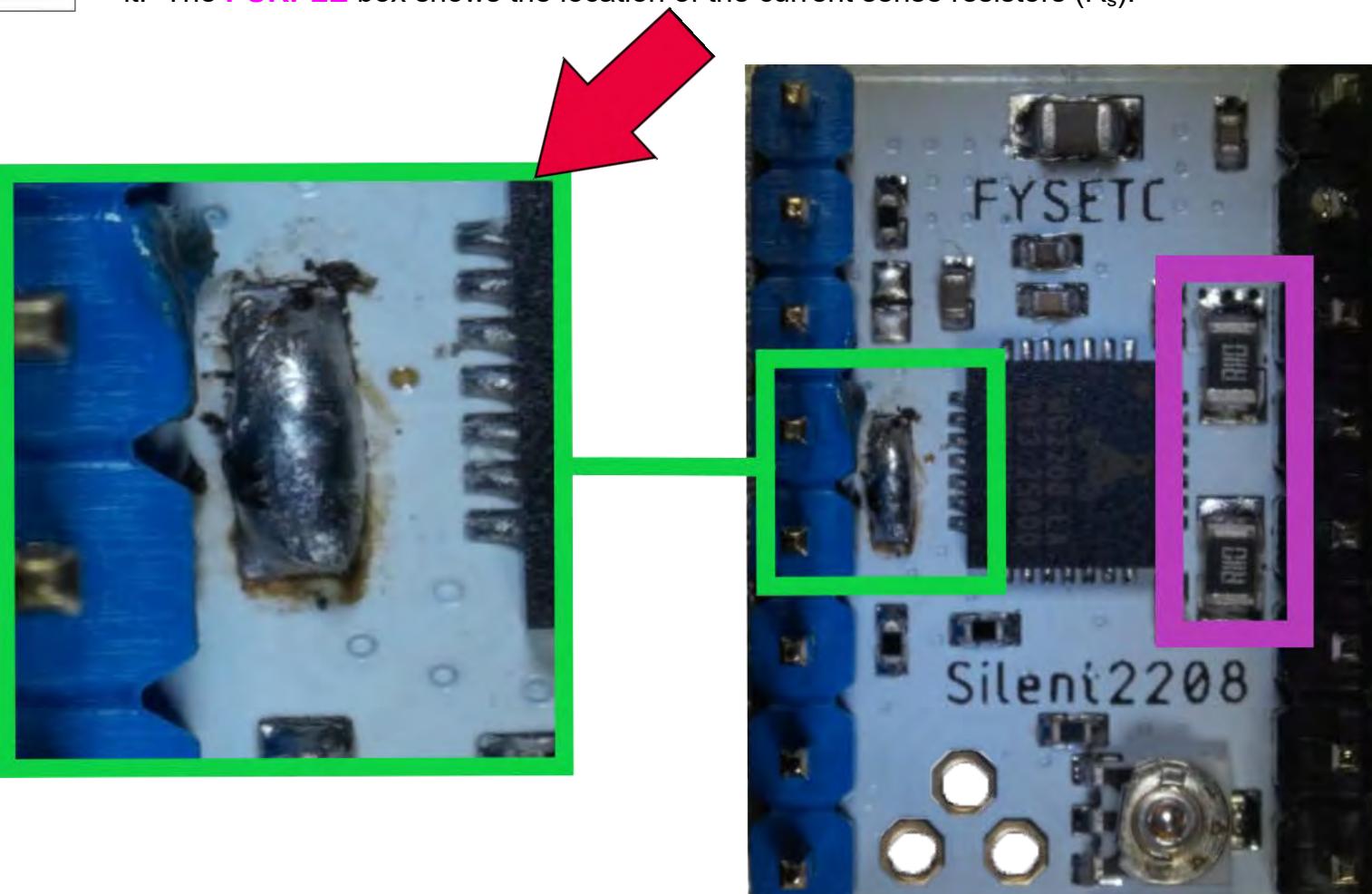
MOST FYSETC TMC2208 boards are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. So **please check your boards to ensure they are in the correct mode you desire!** To be able to program the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

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

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. The **PURPLE** box shows the location of the current sense resistors (R_s).



OTP (One Time Programming) Mode

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

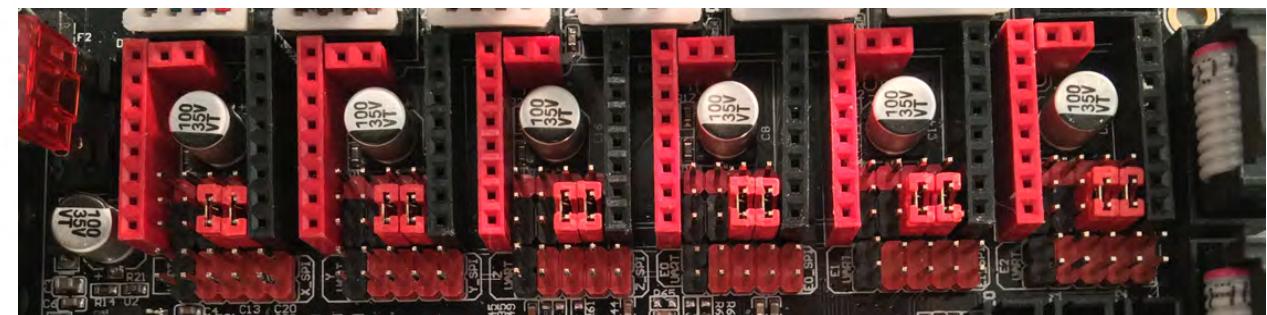
OTP

1 / 8

Interpolation: 1/256

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	M1B
UART	-	-	M1A
PDN	-	MS1	M2A
CLK	MS0	-	M2B
STEP	-	9 8	VIO
DIR	-	9 8	GND



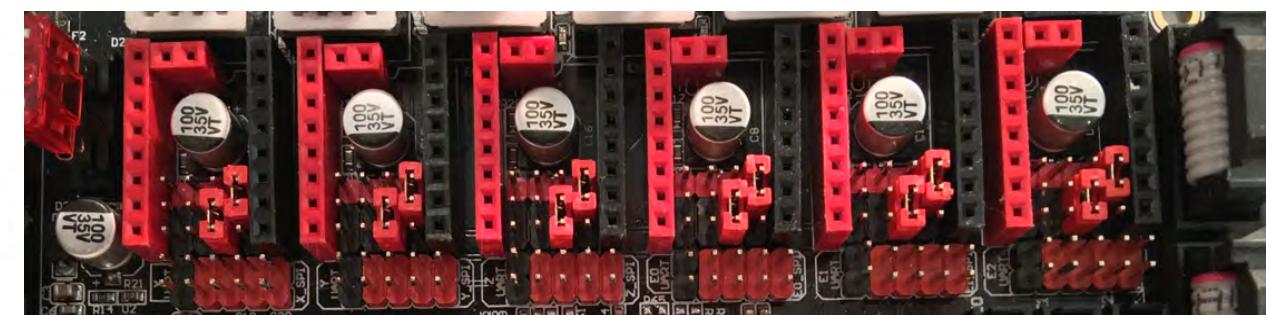
OTP

1 / 2

Interpolation: 1/256

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	M1B
UART	-	-	M1A
PDN	-	MS1	M2A
CLK	MS0	-	M2B
STEP	-	9 8	VIO
DIR	-	9 -	GND



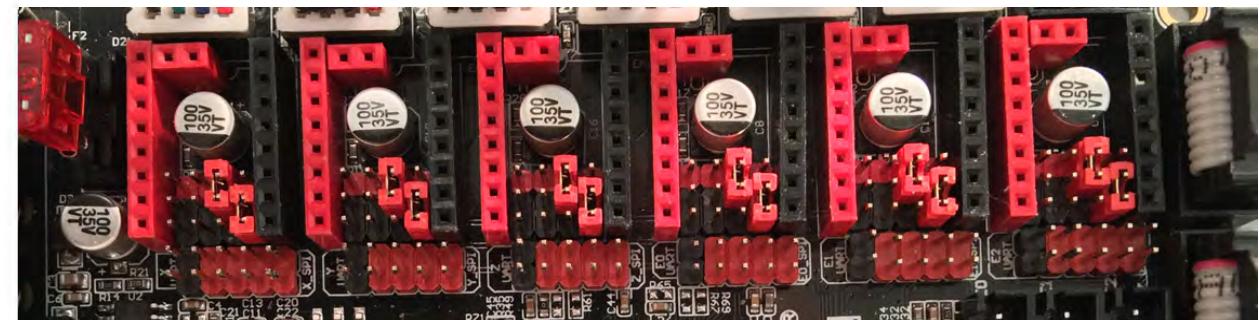
OTP (One Time Programming) Mode

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

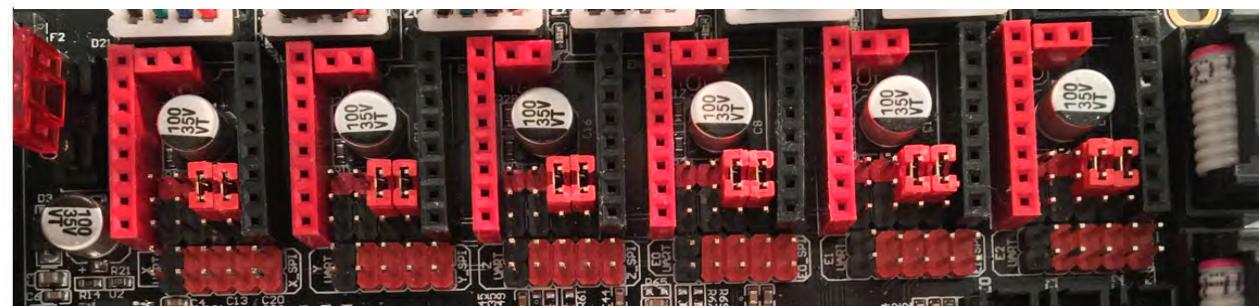
OTP
1 / 4
Interpolation: 1/256
SpreadCycle

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	MS0	-	M2A
CLK	9	-	-	M2B
STEP	-	9	8	VIO
DIR	-	-	8	GND



OTP
1 / 16
Interpolation: 1/256
SpreadCycle

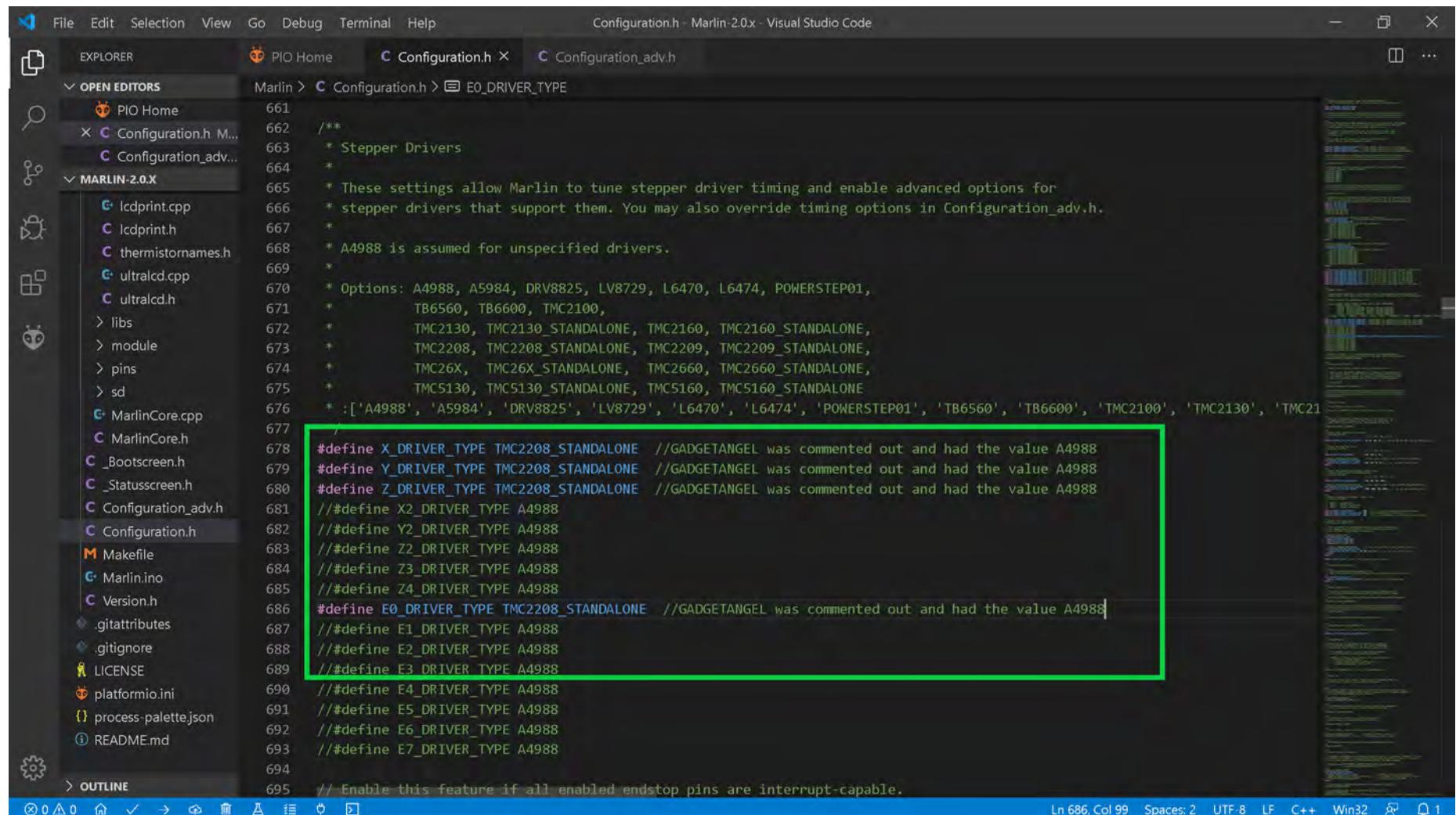
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	MS0	-	M2A
CLK	9	8	-	M2B
STEP	-	9	8	VIO
DIR	-	-	-	GND



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

NOTE: Go to Appendix C, and 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 ("//").



```

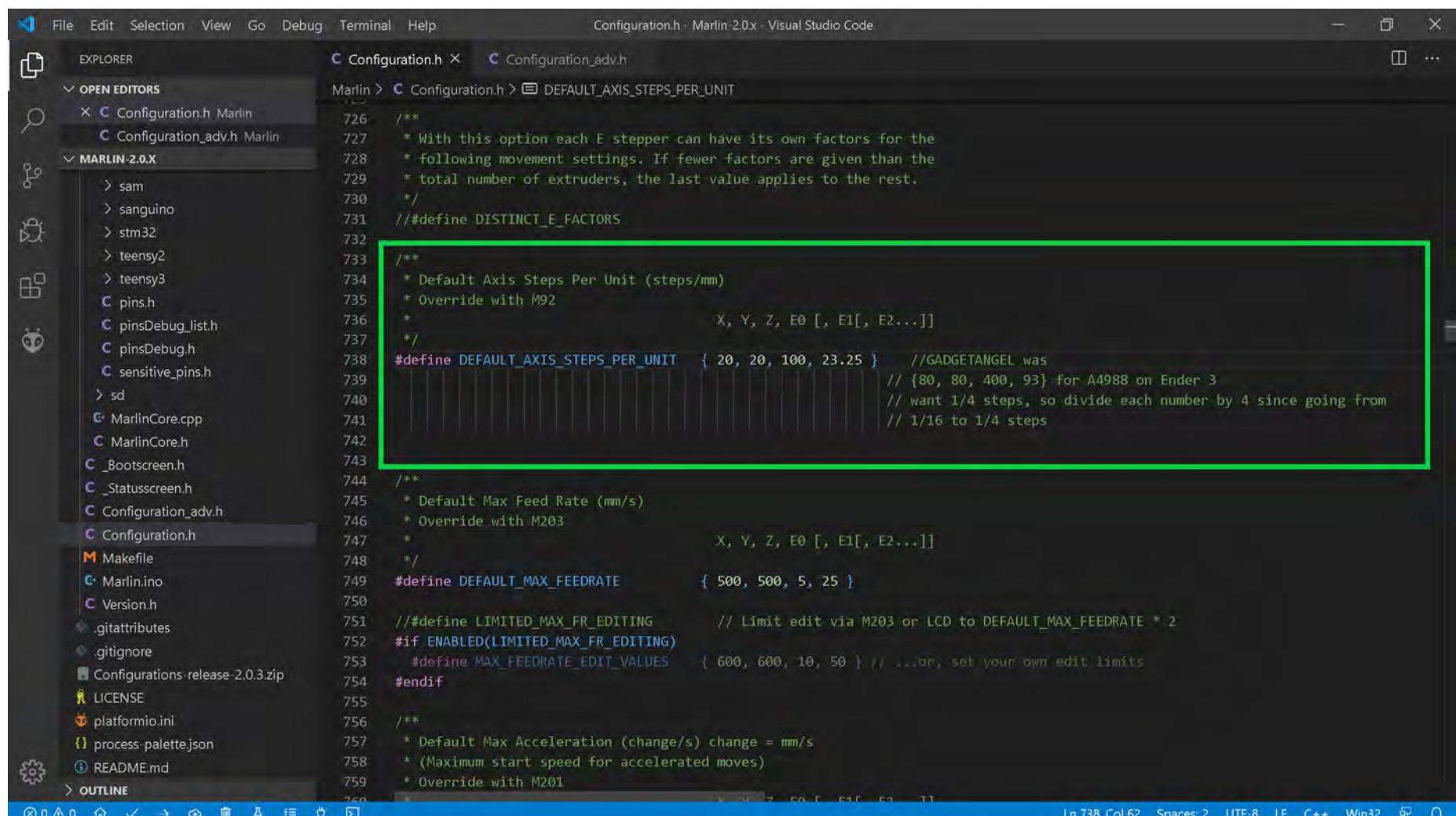
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home 661 /**
X Configuration.h M... 662 */
C Configuration.h M... 663 * Stepper Drivers
C Configuration_adv.h 664 *
MARLIN-2.0.X 665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
Lcdprint.cpp 666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
Lcdprint.h 667 *
thermistornames.h 668 * A4988 is assumed for unspecified drivers.
ultralcd.cpp 669 *
ultralcd.h 670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
> libs 671 * TB6560, TB6600, TMC2100,
> module 672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
> pins 673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
> sd 674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
MarlinCore.cpp 675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
MarlinCore.h 676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
Configuration.h 677 */
#define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value 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_STANDALONE //GADGETANGEL was commented out and had the value 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 FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, 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 line of code:

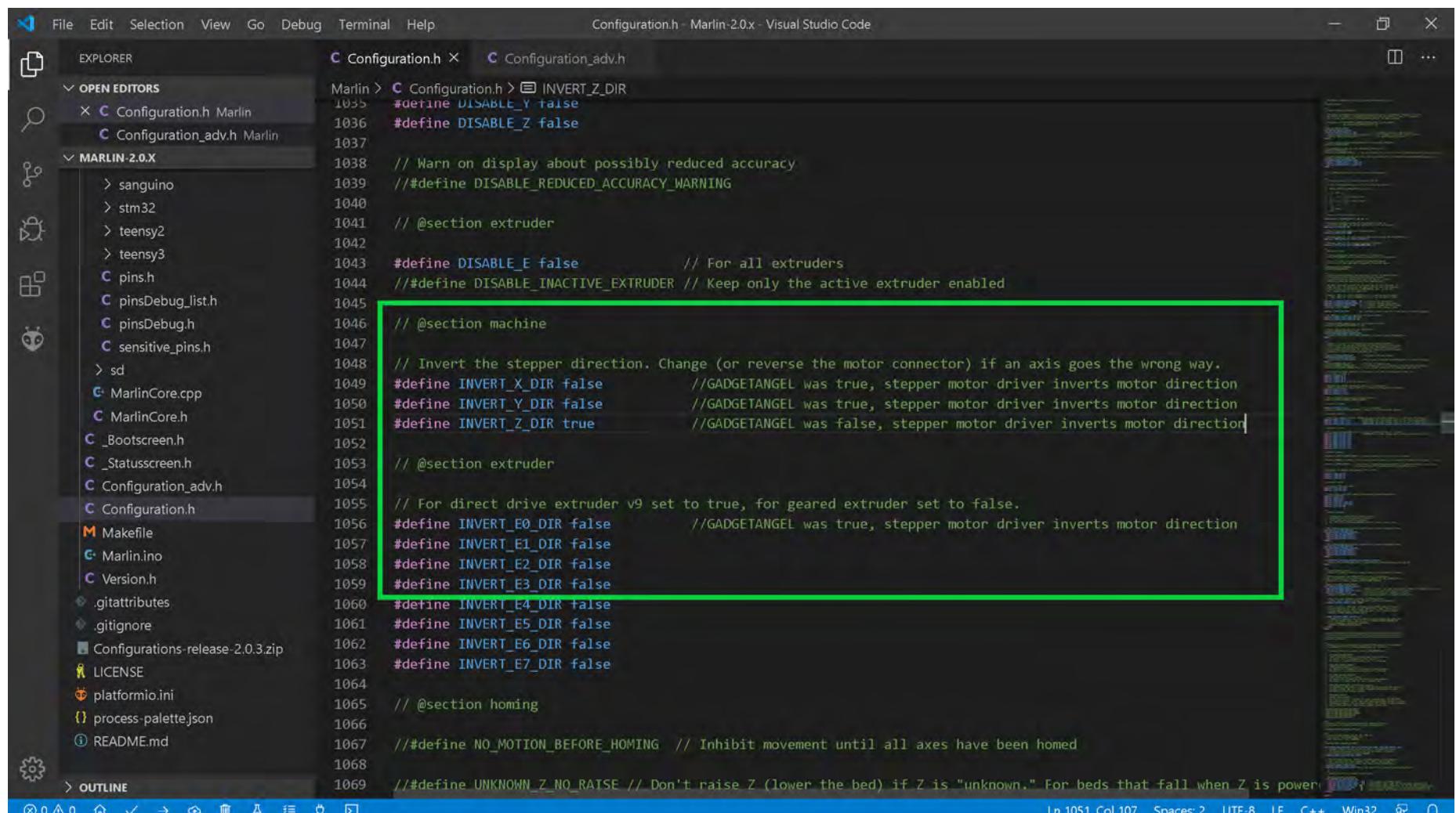
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps
```

The code editor's status bar at the bottom right indicates: Ln 738, Col 62, 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



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

```

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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

A green rectangular box highlights the following lines of code, indicating they are the focus of the instructions:

```

1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction

```

- 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 following details:

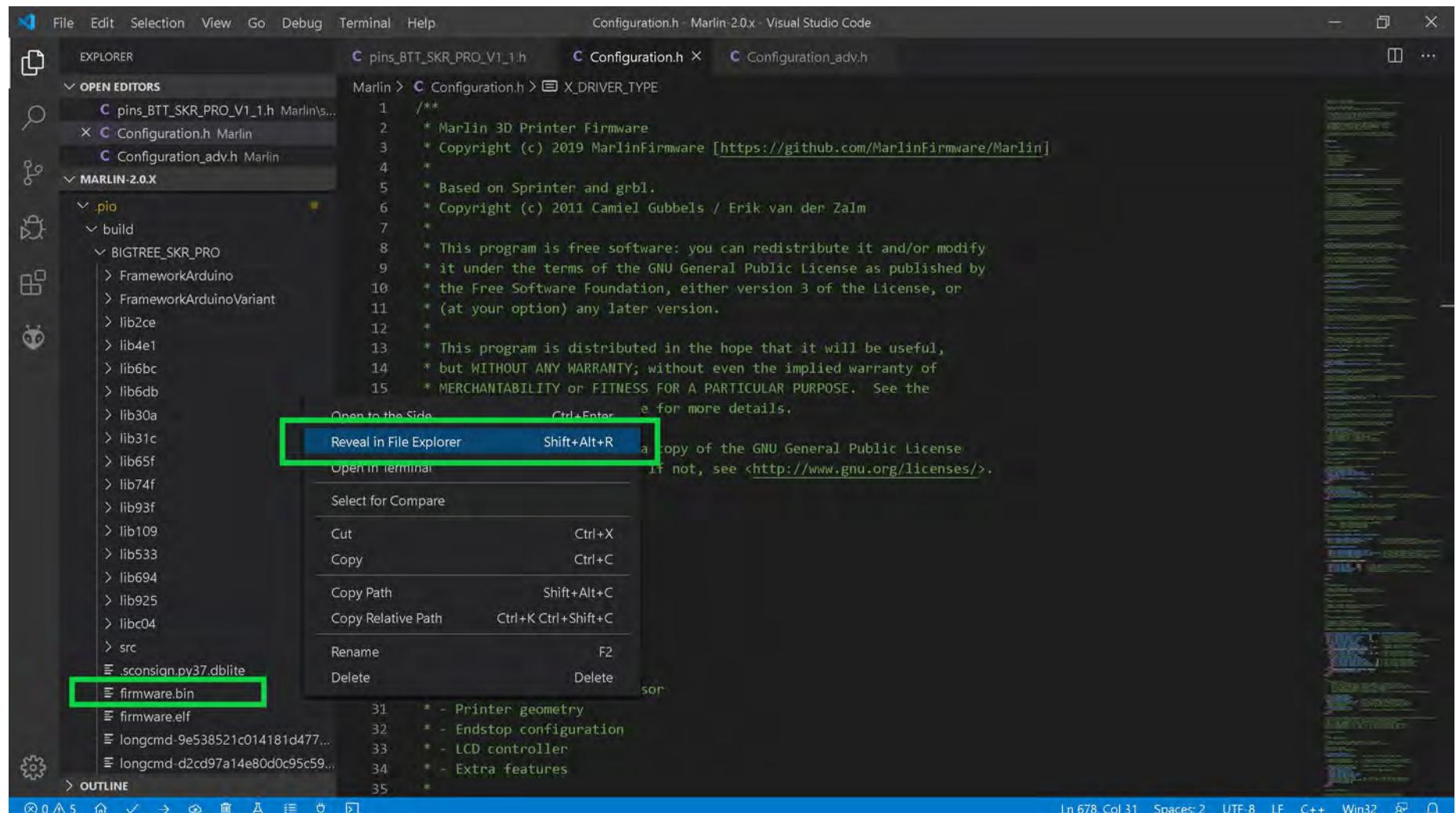
- File Explorer:** On the left, it lists project files under "MARLIN-2.0.X".
- Editors:** Three tabs are open: "pins_BTT_SKR_PRO_V1_1.h", "Configuration.h", and "Configuration_adv.h".
- Terminal:** The terminal shows a successful build process:

```
STMD2E407HE_1101: 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 =====
```
- Status Bar:** Shows file count (1), line count (678), column count (Col 31), space count (Spaces 2), line end (LF), C++ mode, Win32, and a battery icon.

- 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

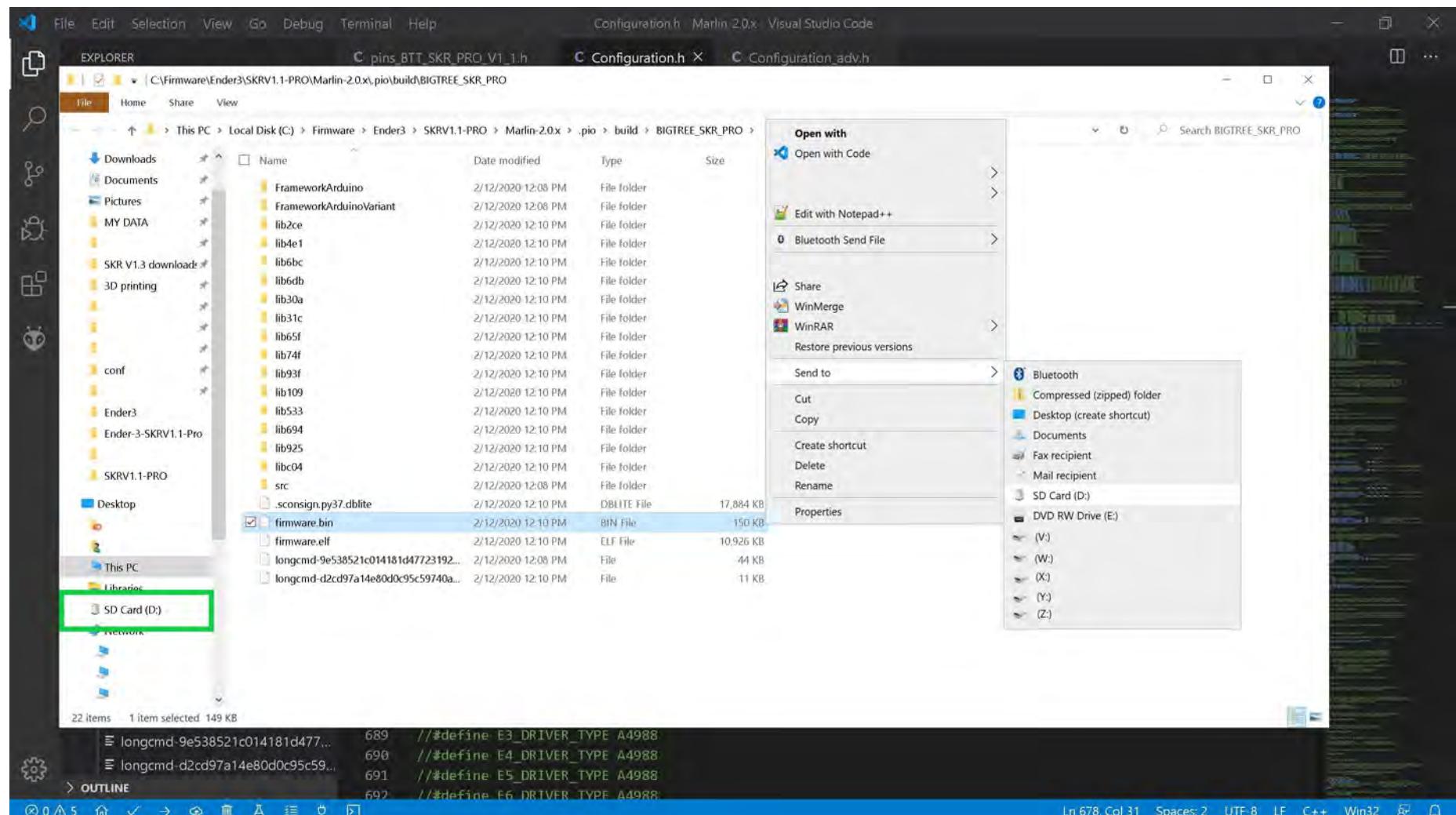
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



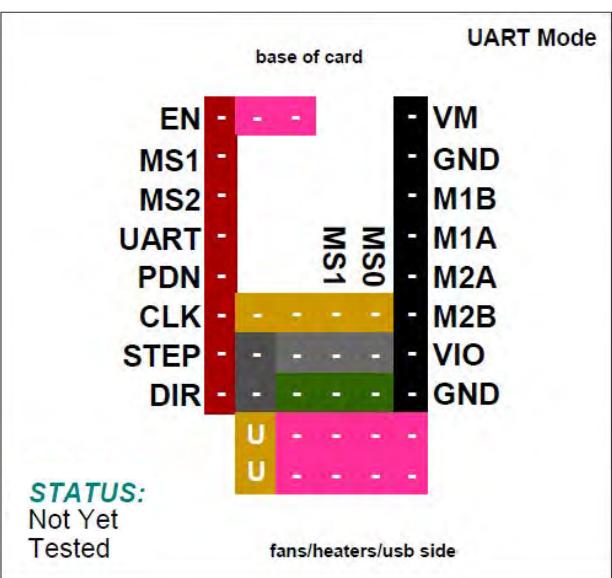
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



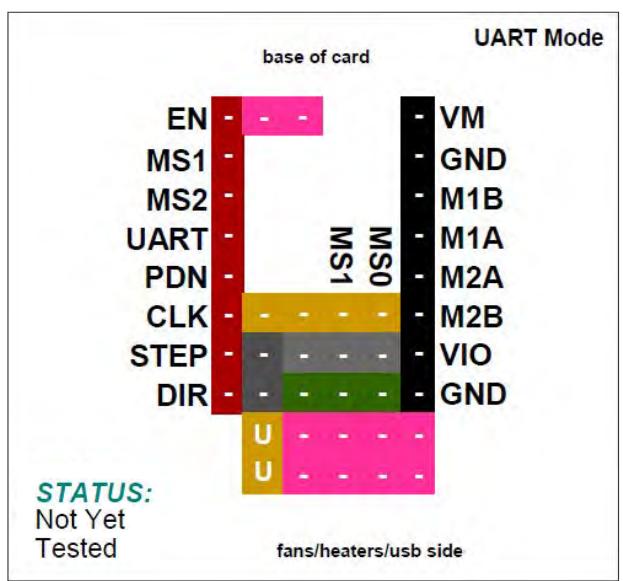
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

FYSETC TMC2208 V1.2**UART Mode**

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

See the next page for further information.

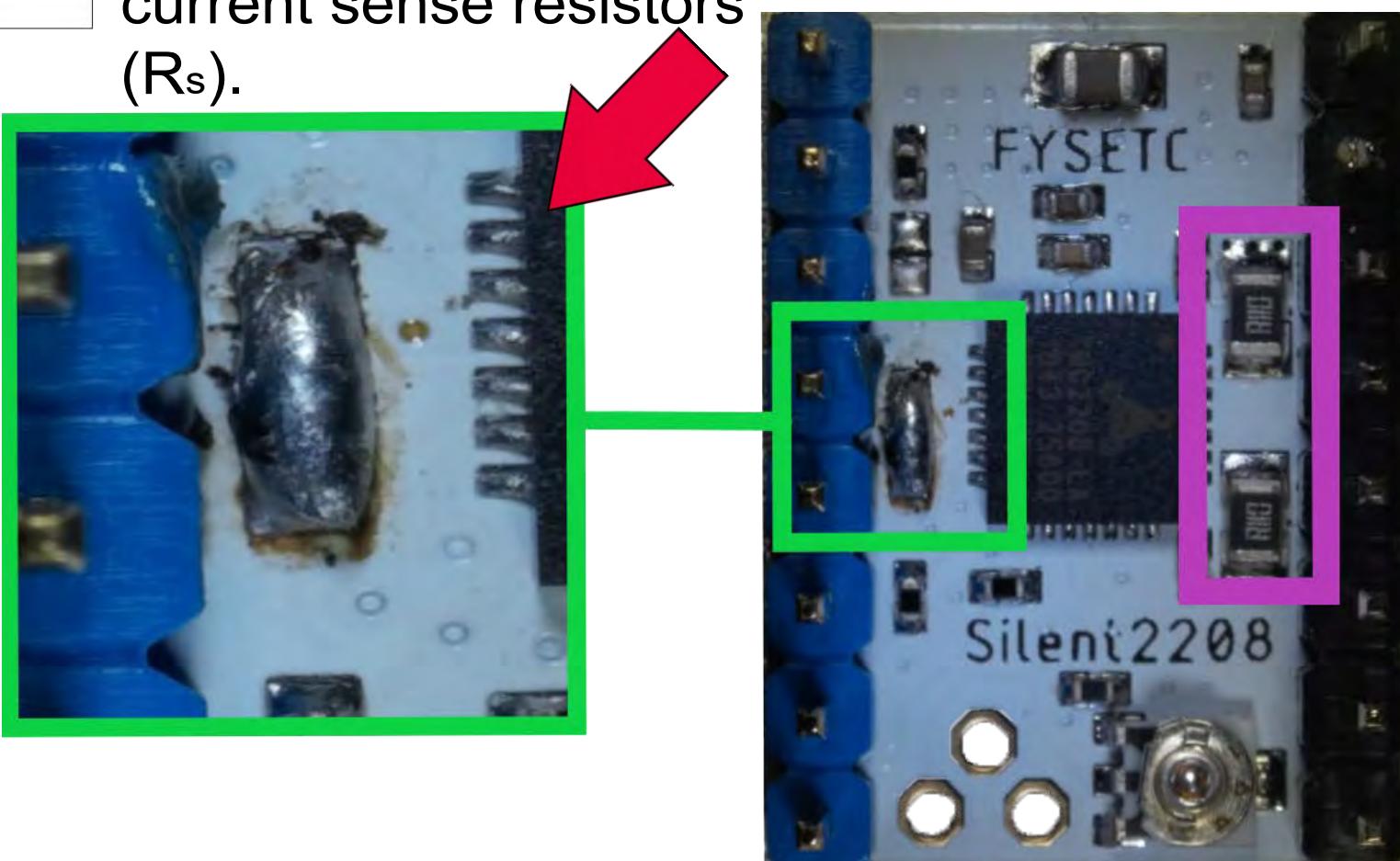
<p>Driver Chip</p> <p>FYSETC</p> <p>TMC2208</p> <p>UART Mode</p> <p>Maximum 256 Subdivision</p> <p>35V DC</p> <p>2A (peak)</p>	<h1>Steps are set inside of your Firmware</h1>	
<p>Driving Current Calculation Formula</p> <p>R_S (Typical Sense Resistor) = 0.11Ω</p>	$I_{MAX} = V_{ref}$ See Appendix B #4. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	$V_{ref} = I_{MAX}$ See Appendix B #4. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$



FYSETC TMC2208 V1.2

UART Mode

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 the **GREEN** box. The **PURPLE** box shows the location of the current sense resistors (R_s).



MOST FYSETC TMC2208 boards 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 stand-alone mode by default. **So please check your boards to ensure they are in the correct mode you desire!**

FYSETC TMC2208 V1.2 in **UART Mode**

UART Mode[UART Mode](#)

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

$R_s = R_{050}$ is 0.05 Ohms

$R_s = R_{062}$ is 0.062 Ohms

$R_s = R_{068}$ is 0.068 Ohms

$R_s = R_{075}$ is 0.075 Ohms

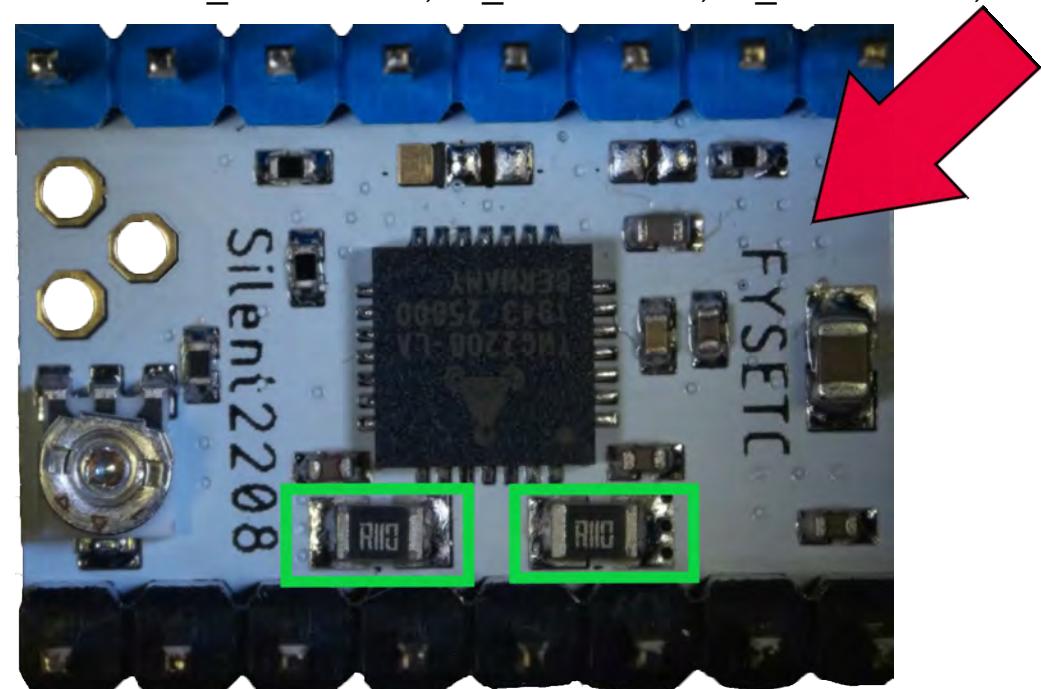
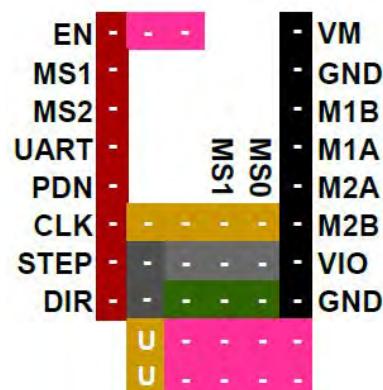
$R_s = R_{100}$ is 0.1 Ohms

$R_s = R_{110}$ is 0.11 Ohms

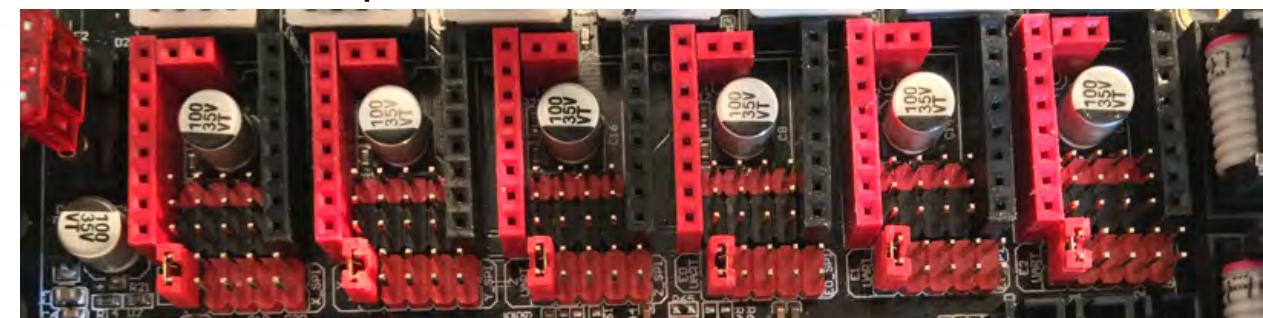
$R_s = R_{150}$ is 0.15 Ohms

$R_s = R_{200}$ is 0.2 Ohms

$R_s = R_{220}$ is 0.22 Ohms

**UART**

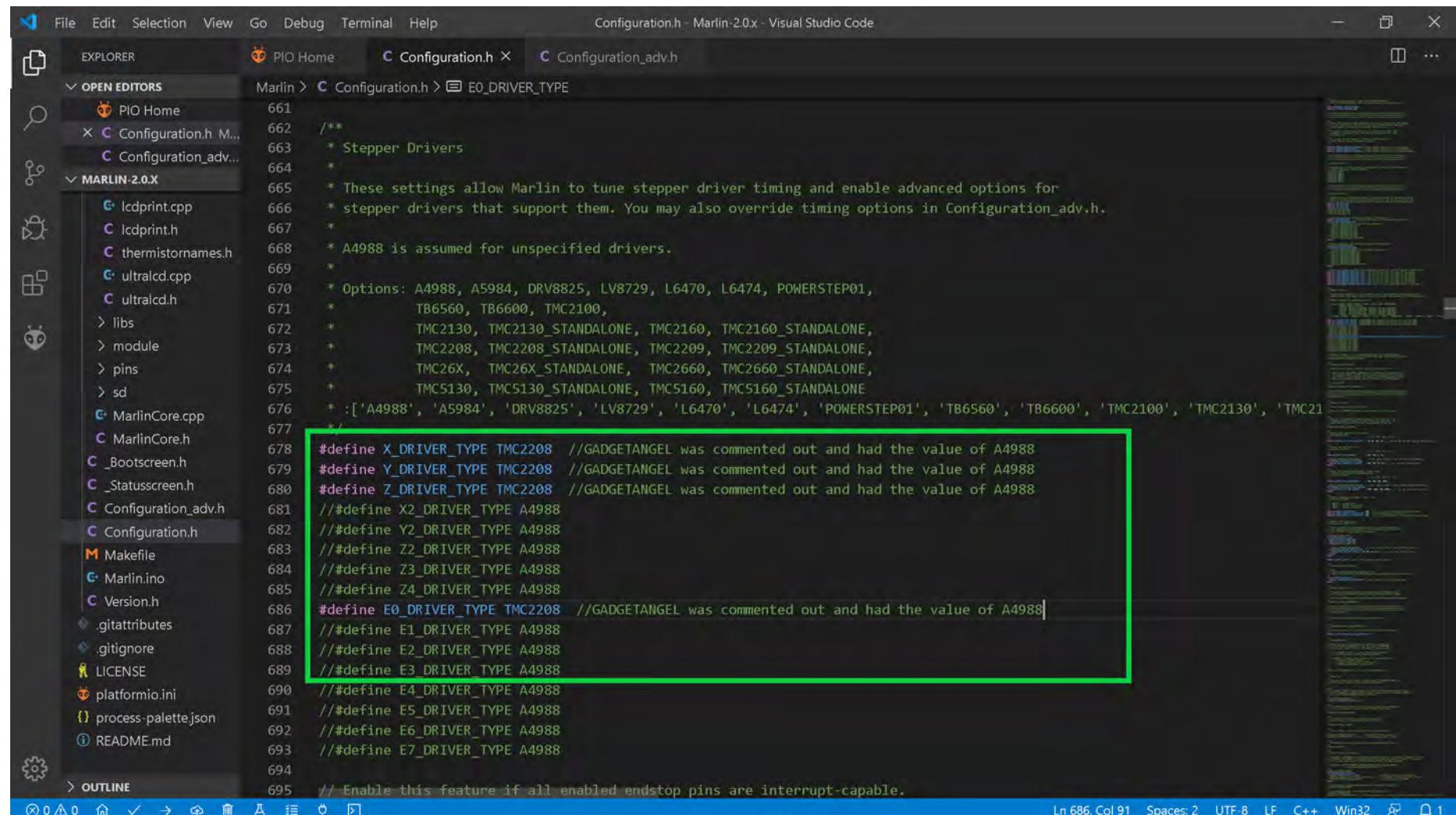
Note: Set Jumper "U" for UART MODE!!



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

NOTE: Go to Appendix C, and 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 Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h ...
Configuration_adv.h
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
Ln 686, Col 91 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌂ 1

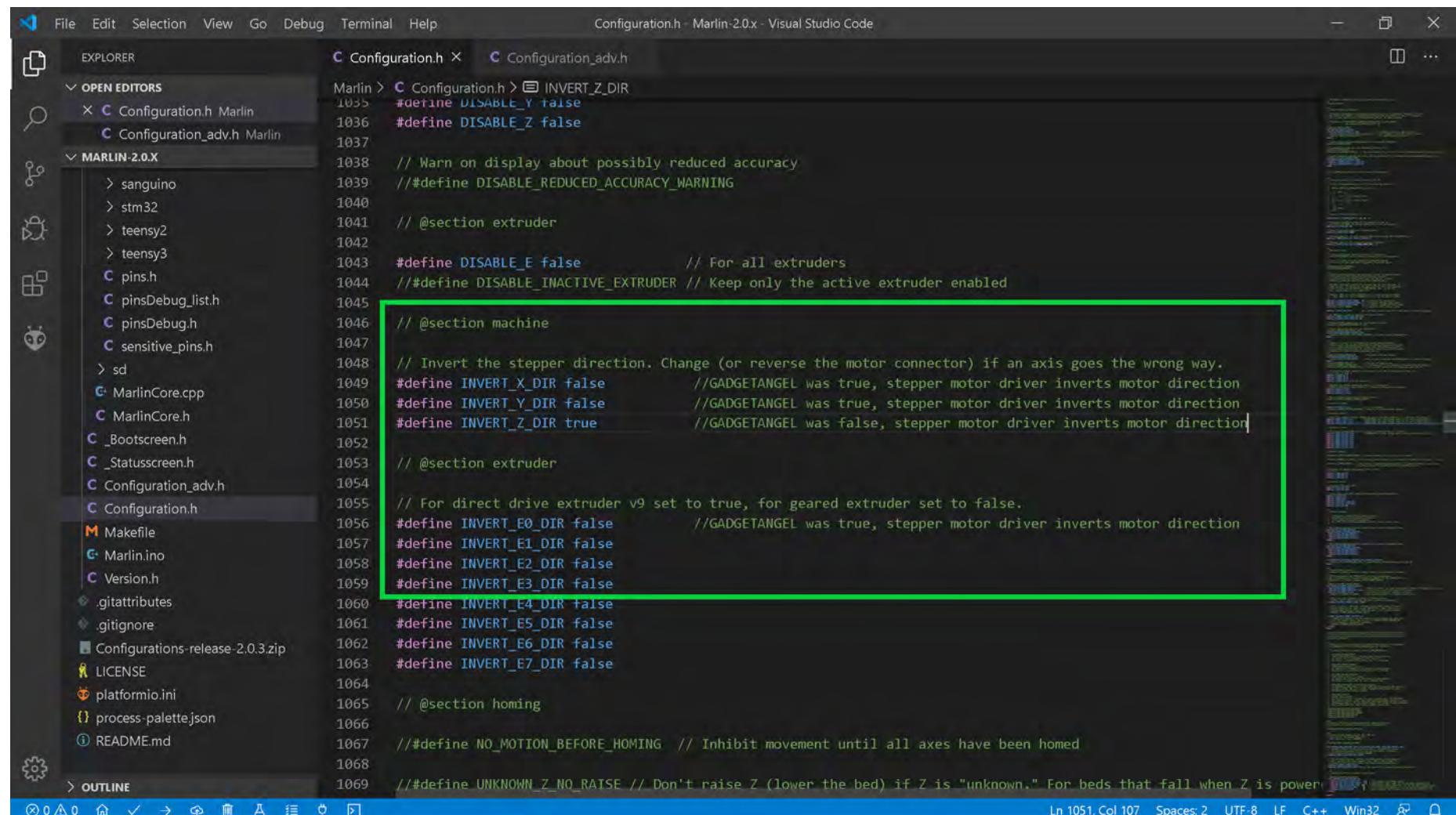
661 /**
662 * Stepper Drivers
663 *
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
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', 'TMC2
677 */
678 #define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
679 #define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
680 #define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the 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 TMC2208 //GADGETANGEL was commented out and had the 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 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 Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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    // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

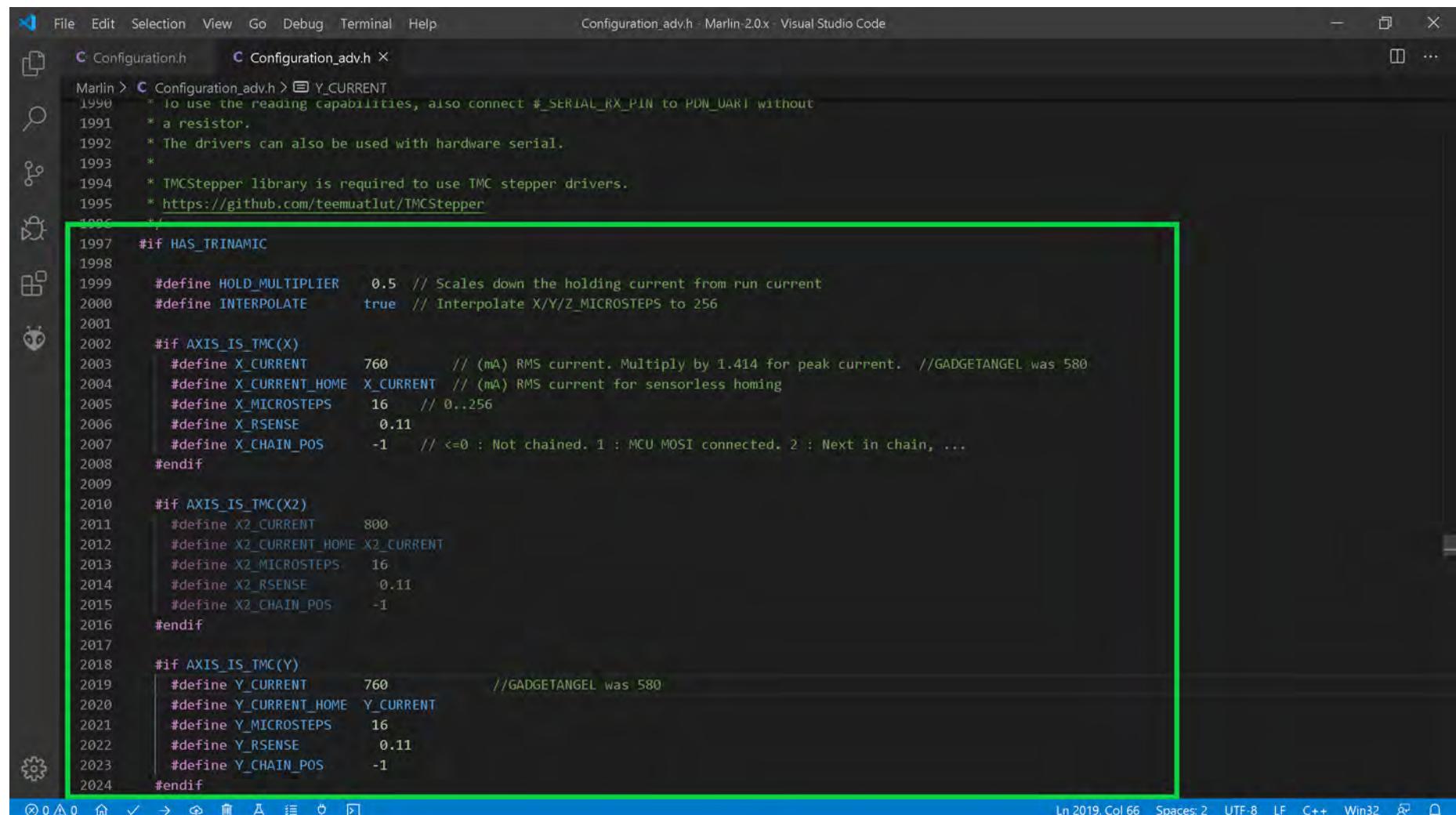
```

Ln 1051, Col 107 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 V_{ref} 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 V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

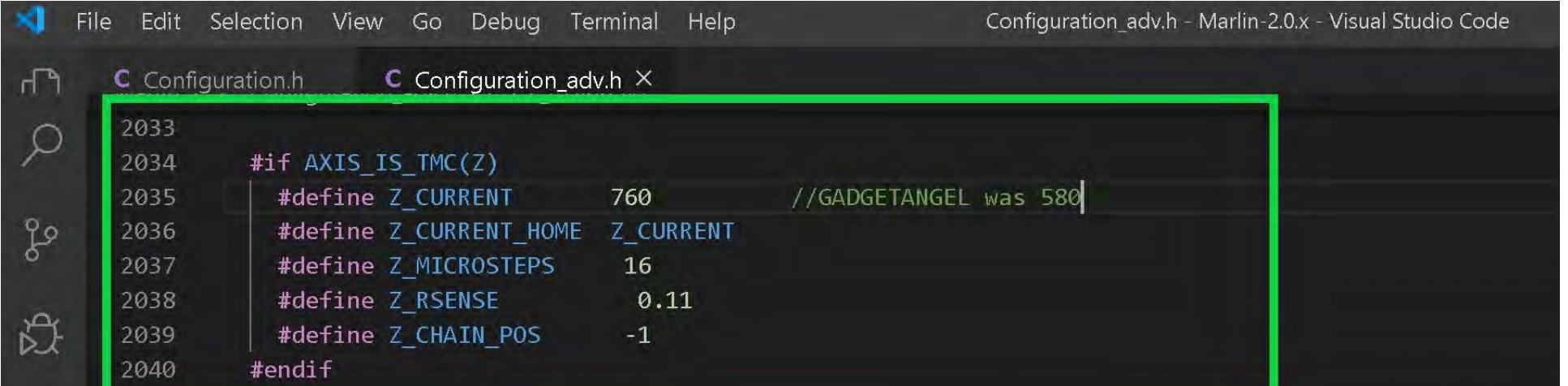
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
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. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
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 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.11
2023 #define Y_CHAIN_POS -1
2024#endif

```

- 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 V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.11. Ensure "E0_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".



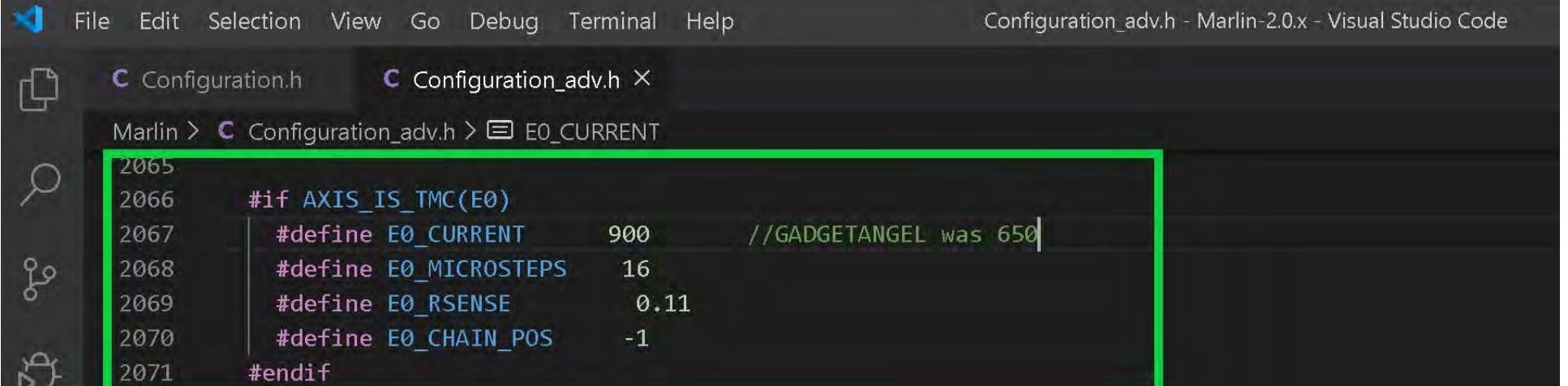
```

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

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760          //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME Z_CURRENT
2037   #define Z_MICROSTEPS    16
2038   #define Z_RSENSE        0.11
2039   #define Z_CHAIN_POS     -1
2040 #endif

```



```

File Edit Selection View Go Debug Terminal Help
Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > E0_CURRENT

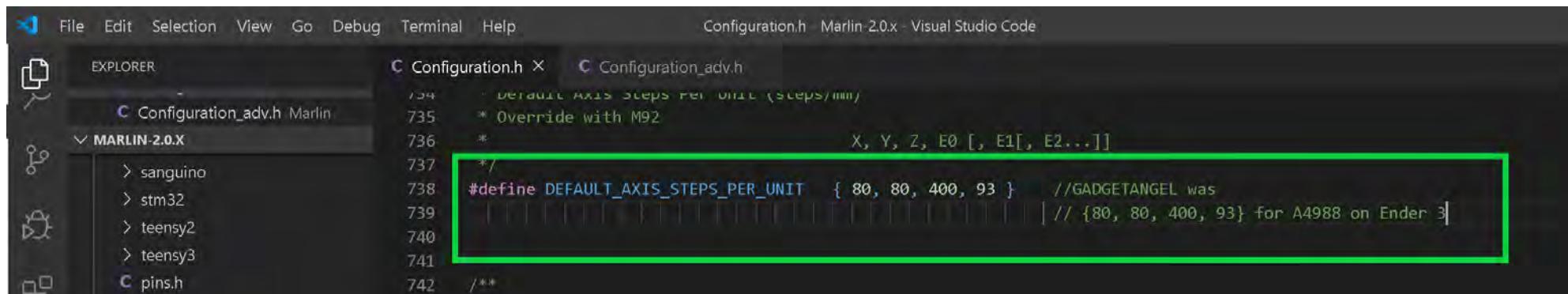
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900          //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- Go to the next page.

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

- 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 Configuration.h X Configuration_adv.h

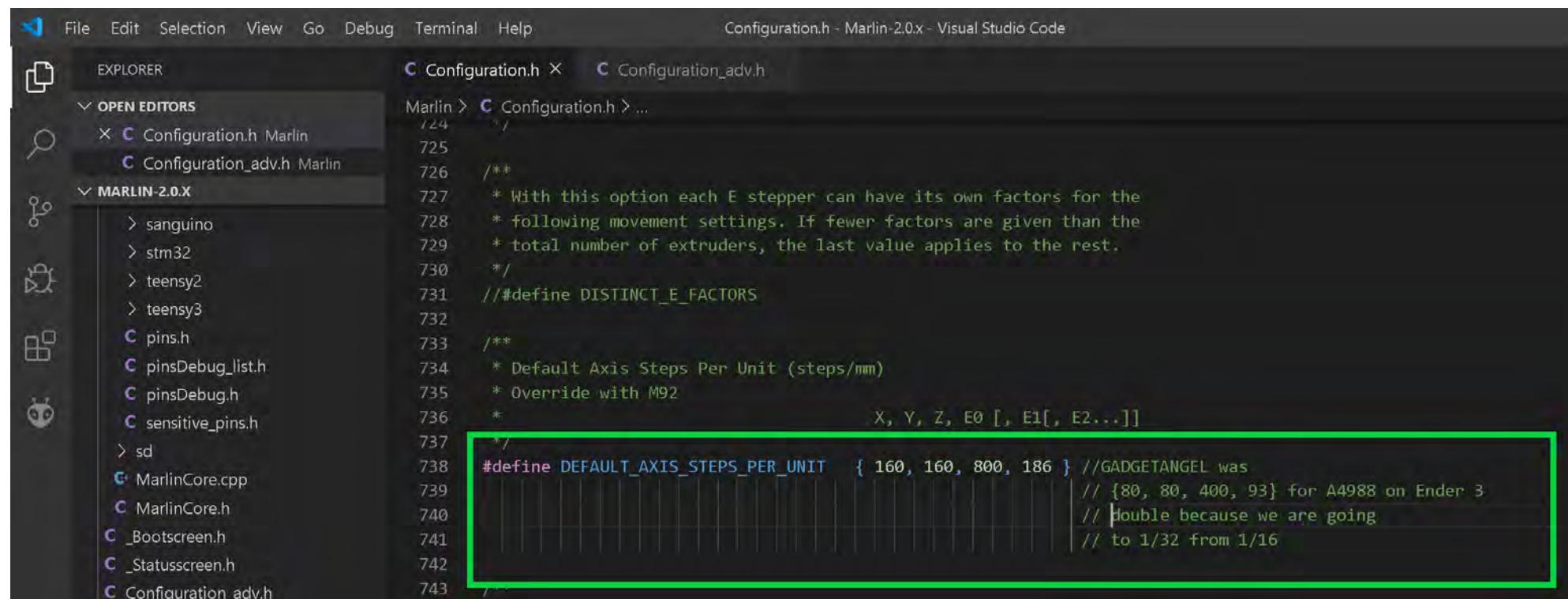
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

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 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 /**

```

- 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 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
Configuration.h Marlin Configuration_adv.h Marlin
Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
sanguino
stm32
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

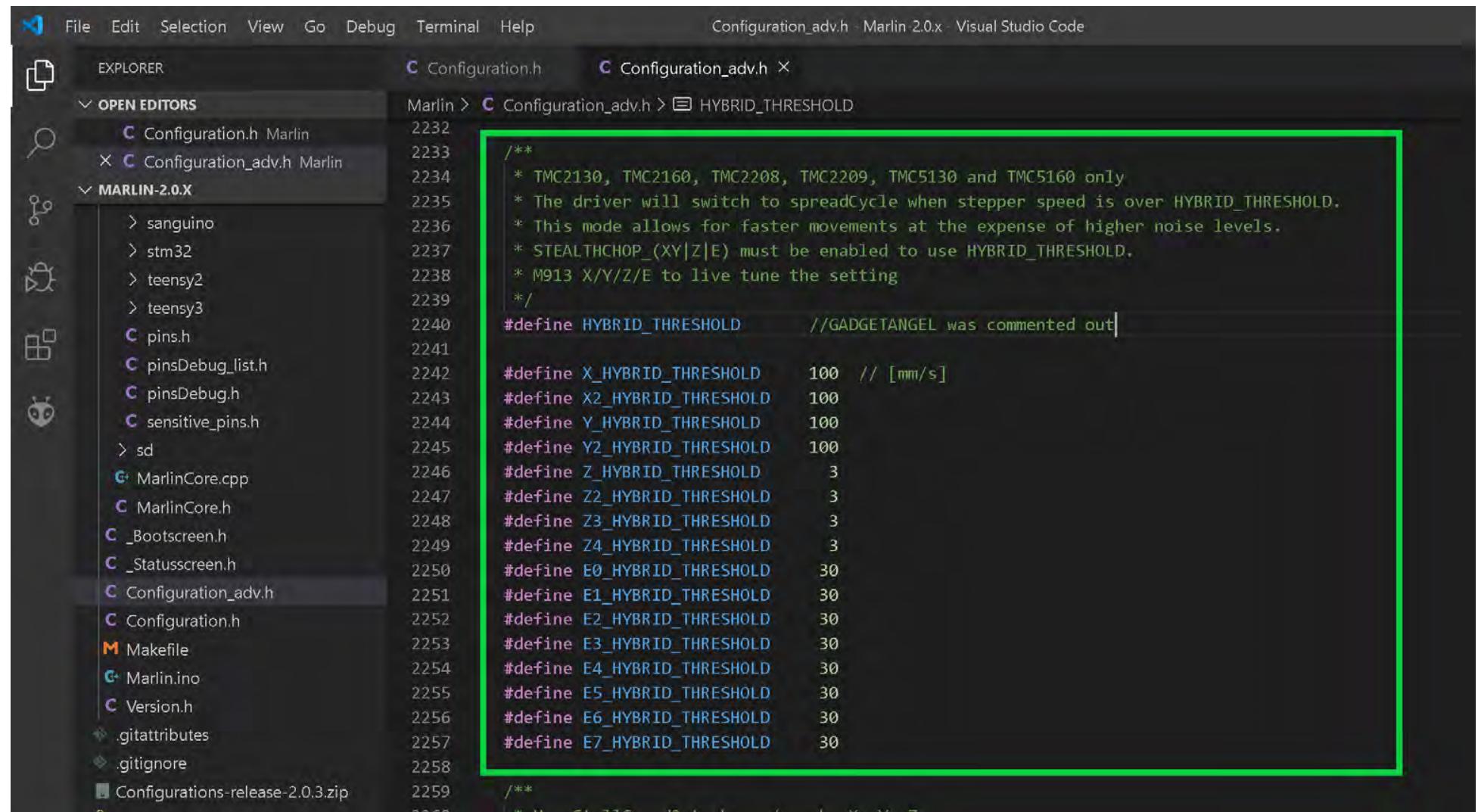
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 */
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
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 /**

```

- 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 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:** Shows the project structure under OPEN EDITORS and MARLIN-2.0.X. The Configuration_adv.h file is listed under MARLIN-2.0.X.
- Editor:** Displays the Configuration_adv.h code. A specific section is highlighted with a green border:


```

      /**
       * 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 //GADGETANGEL 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
      
```

- Go to the next page.

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

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

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h Configuration_adv.h X
Marlin > C Configuration_adv.h > #define MONITOR_DRIVER_STATUS
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 //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```

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

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h Configuration_adv.h X
Marlin > C Configuration_adv.h > #define TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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.

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

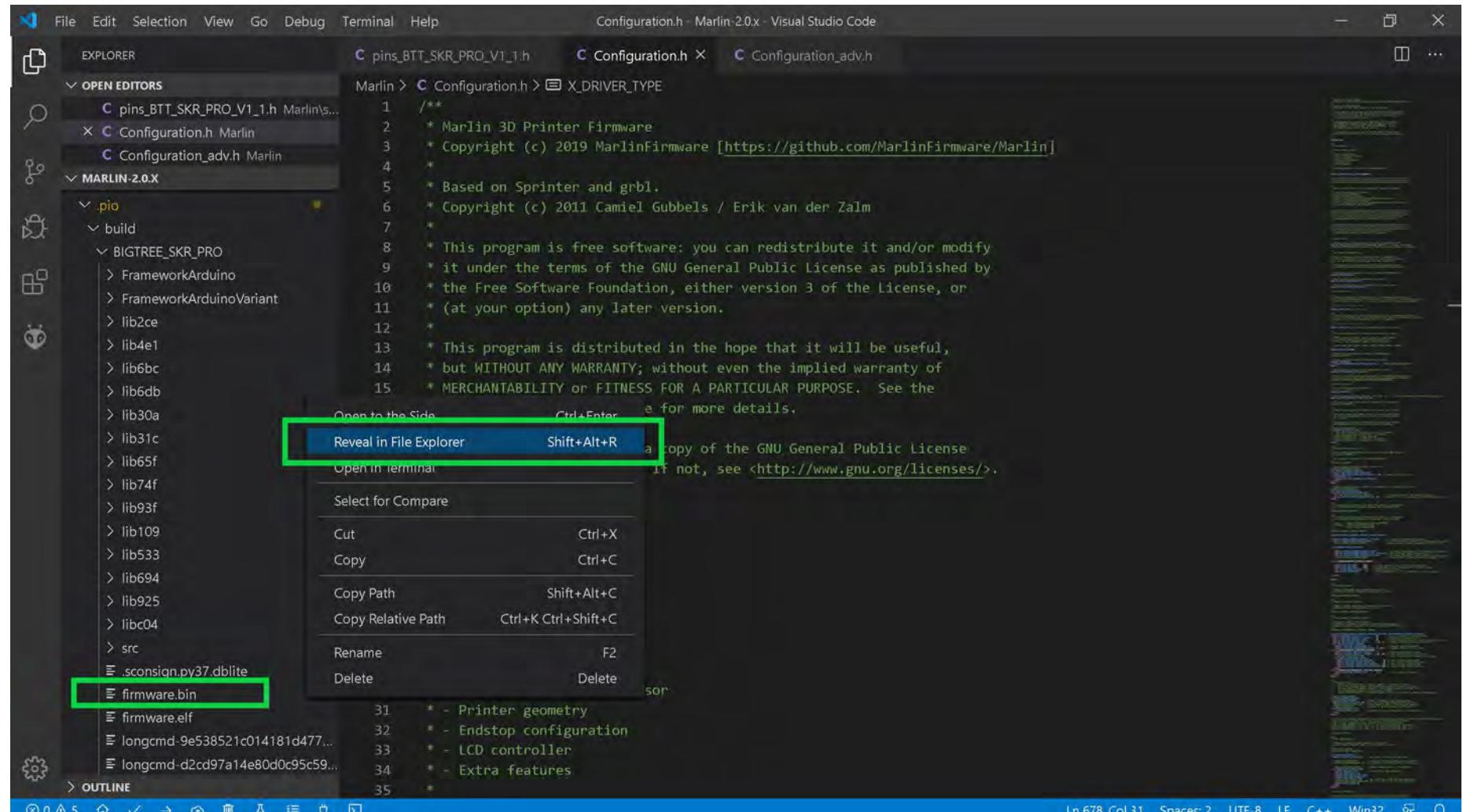
- File Explorer:** Shows the project structure under "MARLIN-2.0.X".
- Editors:** Three tabs are open: "pins_BTT_SKR_PRO_V1_1.h", "Configuration.h", and "Configuration_adv.h".
- Terminal:** The terminal tab shows a build log:

```
1: Task - Build
=====
BIGTREE_SKR_PRO          SUCCESS  00:02:31.294
BIGTREE_CTR_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 =====
```
- Output:** The output tab shows the message: "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

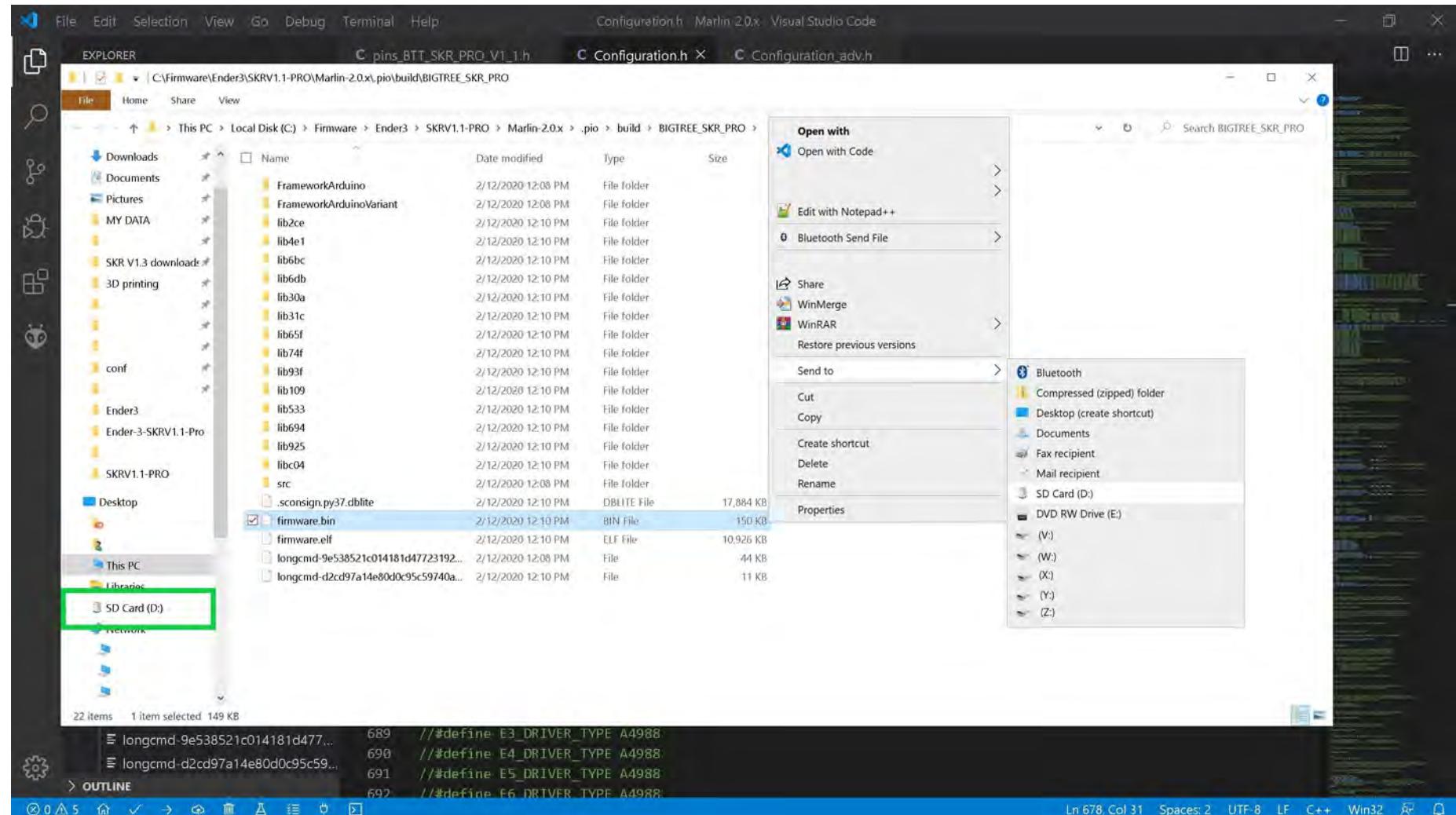
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



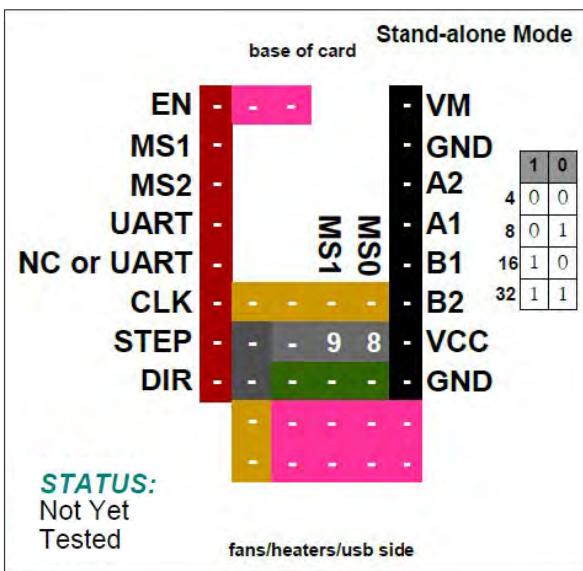
- Go to the next page.

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

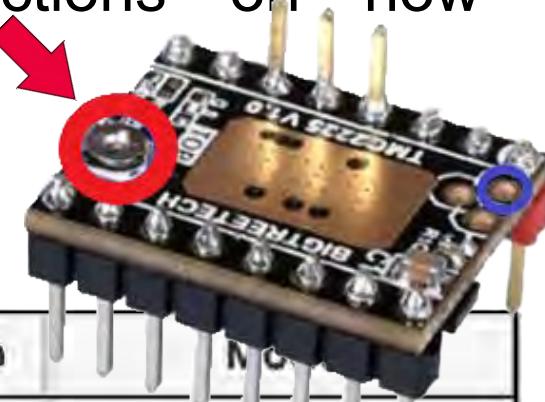
- Ensure your micro SD card, which came with your SKR PRO V1.1 board, with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
- From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

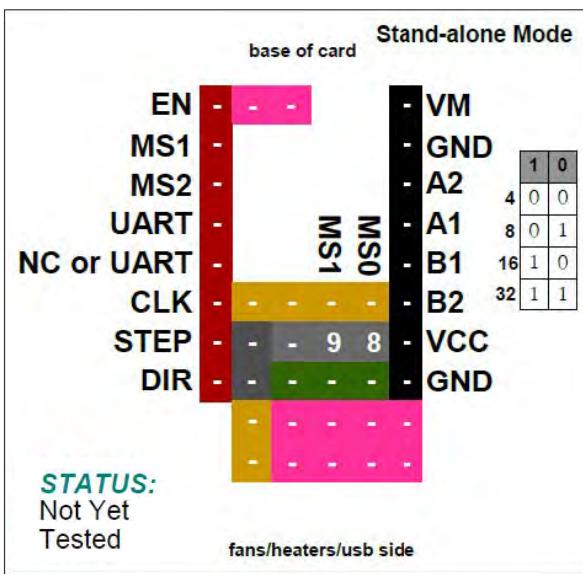
**BIQU TMC2225 V1.0****Stand-alone Mode**

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

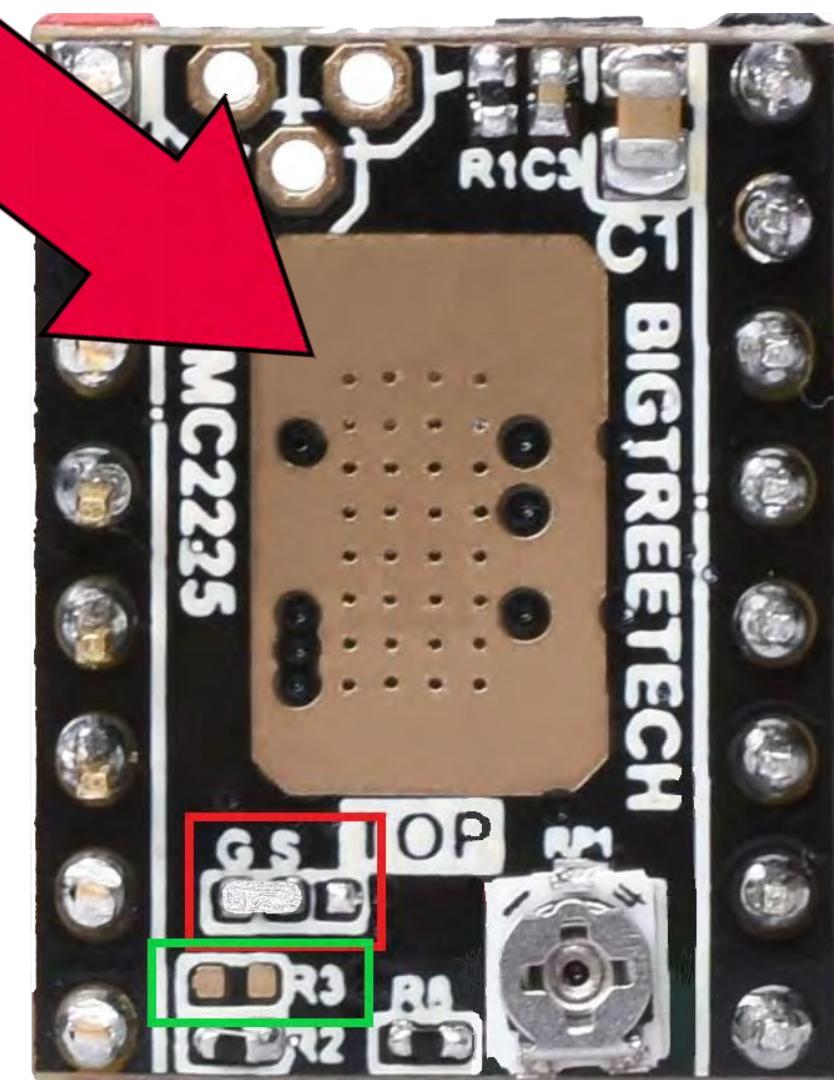
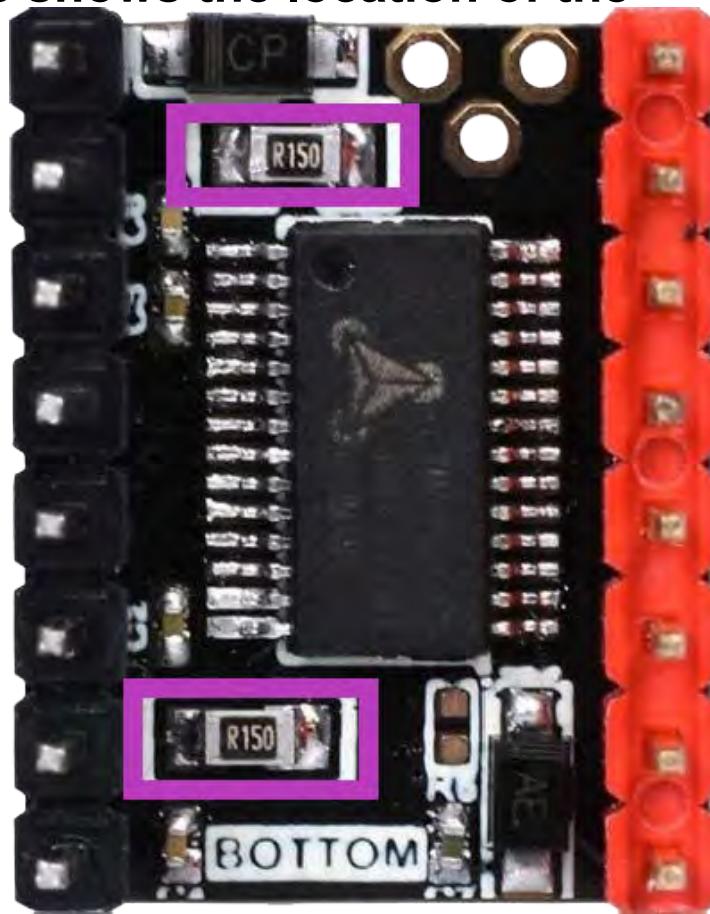
Driver Chip	MS1	MS0	Steps	Interpolation	
BIQU® TMC2225 <small>Stand Alone Mode Maximum 32 Subdivision 35V DC 2A (peak)</small>	GND	GND	1 / 4	1 / 256	stealthChop
	GND	VIO	1 / 8	1 / 256	stealthChop
	VIO	GND	1 / 16	1 / 256	stealthChop
	VIO	VIO	1 / 32	1 / 256	stealthChop
Driving Current Calculation Formula <small>R_S (Typical Sense Resistor) = 0.15Ω</small>	$I_{MAX} = V_{ref} * 0.7222$ See Appendix B #10. Use 50% to 90% as shown below:			$V_{ref} = I_{MAX} * 1.3846$ See Appendix B #10. Use 50% to 90% as shown below:	
	$I_{MAX} = (V_{ref} * 0.7222) * 0.90$			$V_{ref} = (I_{MAX} * 1.3846) * 0.90$	

**BIQU TMC2225 V1.0****Stand-alone Mode**

Note: To obtain **stand-alone mode** for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as seen in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for StealthChop as seen in **RED** below.

The **PURPLE** boxes shows the location of the current sense resistors (R_s).

Note: MOST BIQU TMC2225 V1.0 driver boards, when purchased for **UART mode**, will have two R3 pads (located on the top of the driver board), which are **NOT soldered together**. This indicates the driver board can use the UART pin for the UART single wire interface (if the UART, "U", jumper is in place on the SKR PRO V1.1 board)



Stand-alone Mode

Stand-alone Mode

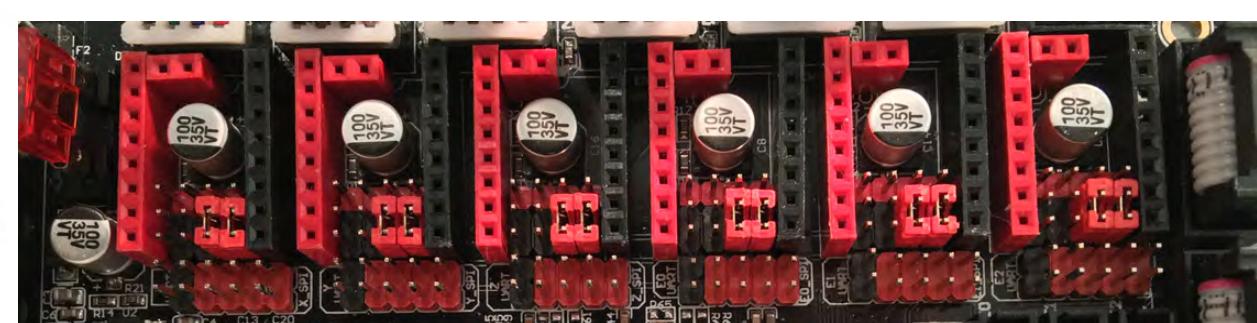
Stand-alone Mode

1 / 4

Interpolation: 1/256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	-	MS1	A1
CLK	-	-	MS0	B1
STEP	-	-	9 8	B2
DIR	-	9	8	VCC
	-	-	-	GND



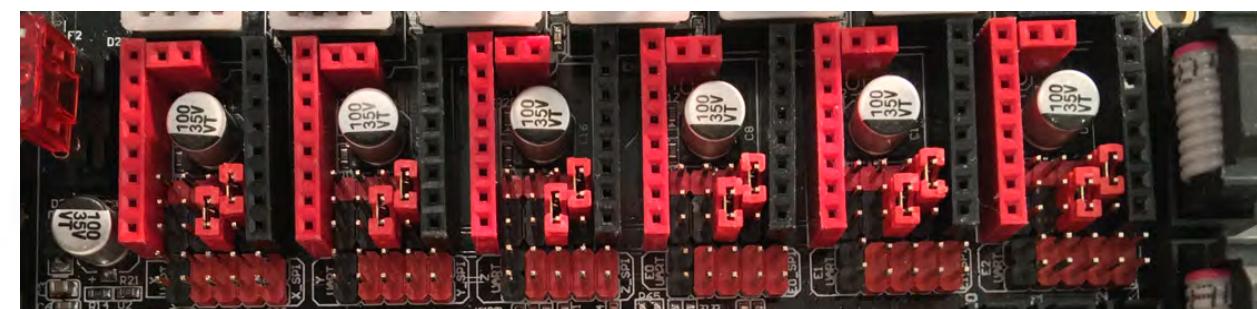
Stand-alone Mode

1 / 8

Interpolation: 1/256

StealthChop

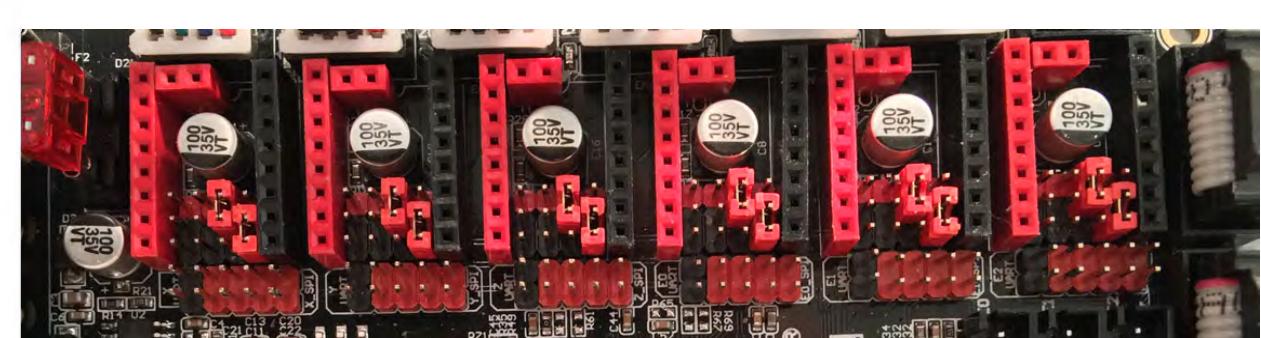
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	-	MS1	A1
CLK	-	-	MS0	B1
STEP	-	-	8	B2
DIR	-	9	8	VCC
	-	-	-	GND



Stand-alone Mode**Stand-alone Mode****1 / 16**Interpolation: **1/256**

StealthChop

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	-	A1
CLK	-	9	B1
STEP	-	9 8	B2
DIR	-	8	VCC
	-	-	GND

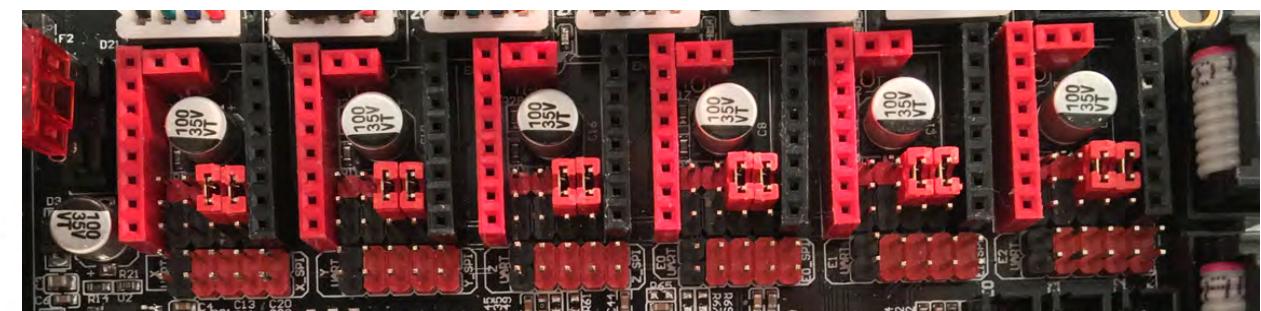


Stand-alone Mode

1 / 32Interpolation: **1/256**

StealthChop

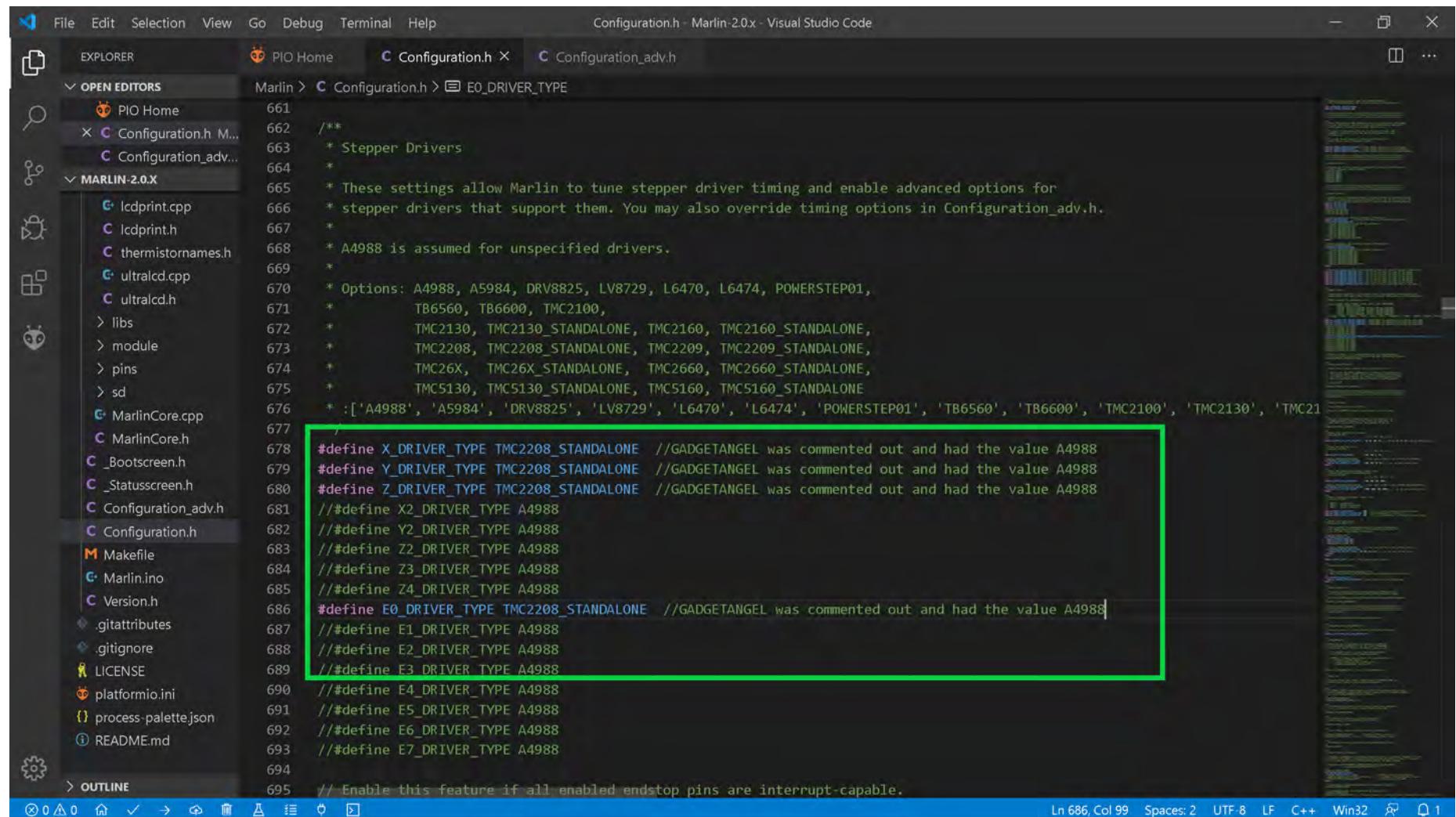
EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	-	A1
NC or UART	-	-	B1
CLK	-	9 8	B2
STEP	-	9 8	VCC
DIR	-	-	GND
	-	-	-



The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in stand-alone mode.

- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 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 TMC2225 drivers in stand-alone mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208_STANDALONE" for the DRIVER_TYPE. 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 */
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 */
669 */
670 */
671 */
672 */
673 */
674 */
675 */
676 */
677 */
678 */
679 */
680 */
681 */
682 */
683 */
684 */
685 */
686 */
687 */
688 */
689 */
690 */
691 */
692 */
693 */
694 */
695 */

```

A green rectangular box highlights the driver type definitions for all axes (X, Y, Z, E0, E1, E2, E3, E4, E5, E6, E7). The highlighted code is as follows:

```

#define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value 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

```

The status bar at the bottom of the code editor shows: Ln 686, Col 99 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1

- Go to the next page.

[The \(latest release of\) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode](#)

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, as seen in the GREEN box 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):** Shows the project structure with files like Configuration.h, Configuration_adv.h, MarlinCore.cpp, and Makefile.
- Central Area:** The code editor displays Configuration.h with the following content:

```
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

733 /**
734  * Default Axis Steps Per Unit (steps/mm)
735  * Override with M92
736  *
737  *          X, Y, Z, E0 [, E1[, E2...]]
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 }    //GADGETANGEL was
739                                         // {80, 80, 400, 93} for A4988 on Ender 3
740                                         // want 1/4 steps, so divide each number by 4 since going from
741                                         // 1/16 to 1/4 steps

744 /**
745  * Default Max Feed Rate (mm/s)
746  * Override with M203
747  *
748  *          X, Y, Z, E0 [, E1[, E2...]]
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

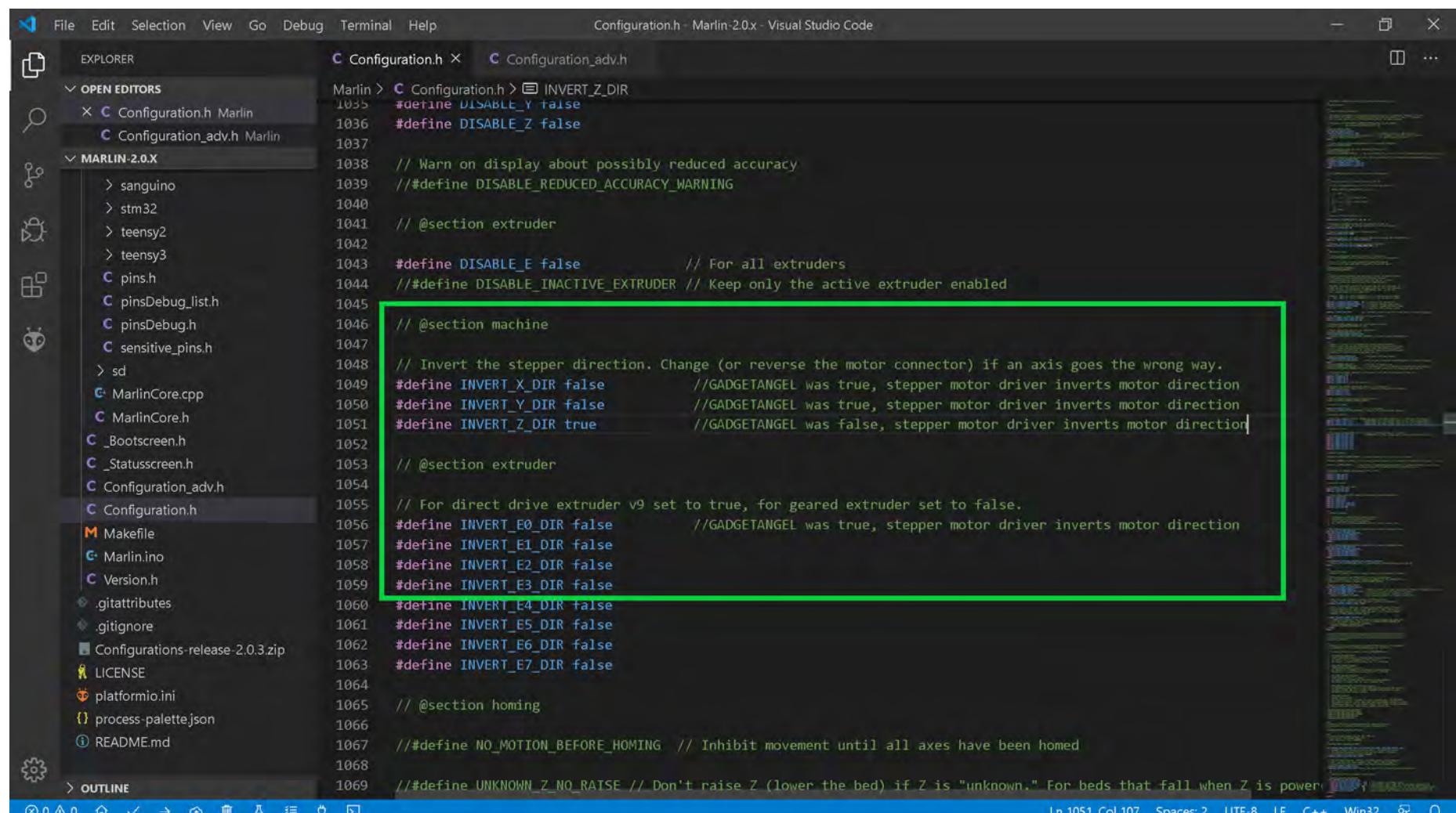
751 // #define LIMITED_MAX_FR_EDITING      // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753   #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
754 #endif

756 /**
757  * Default Max Acceleration (change/s) change = mm/s
758  * (Maximum start speed for accelerated moves)
759  * Override with M201
```
- Bottom Status Bar:** Line 738, Col 62, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 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 TMC2225 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 Configuration.h Configuration_adv.h

MARLIN-2.0.X

```

Marlin > Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 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.

```

Configuration.h Marlin 2.0.x Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h X_DRIVER_TYPE
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...
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 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
TMC2225_V1_0
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_GTR_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 =====
Terminal will be reused by tasks, press any key to close it.

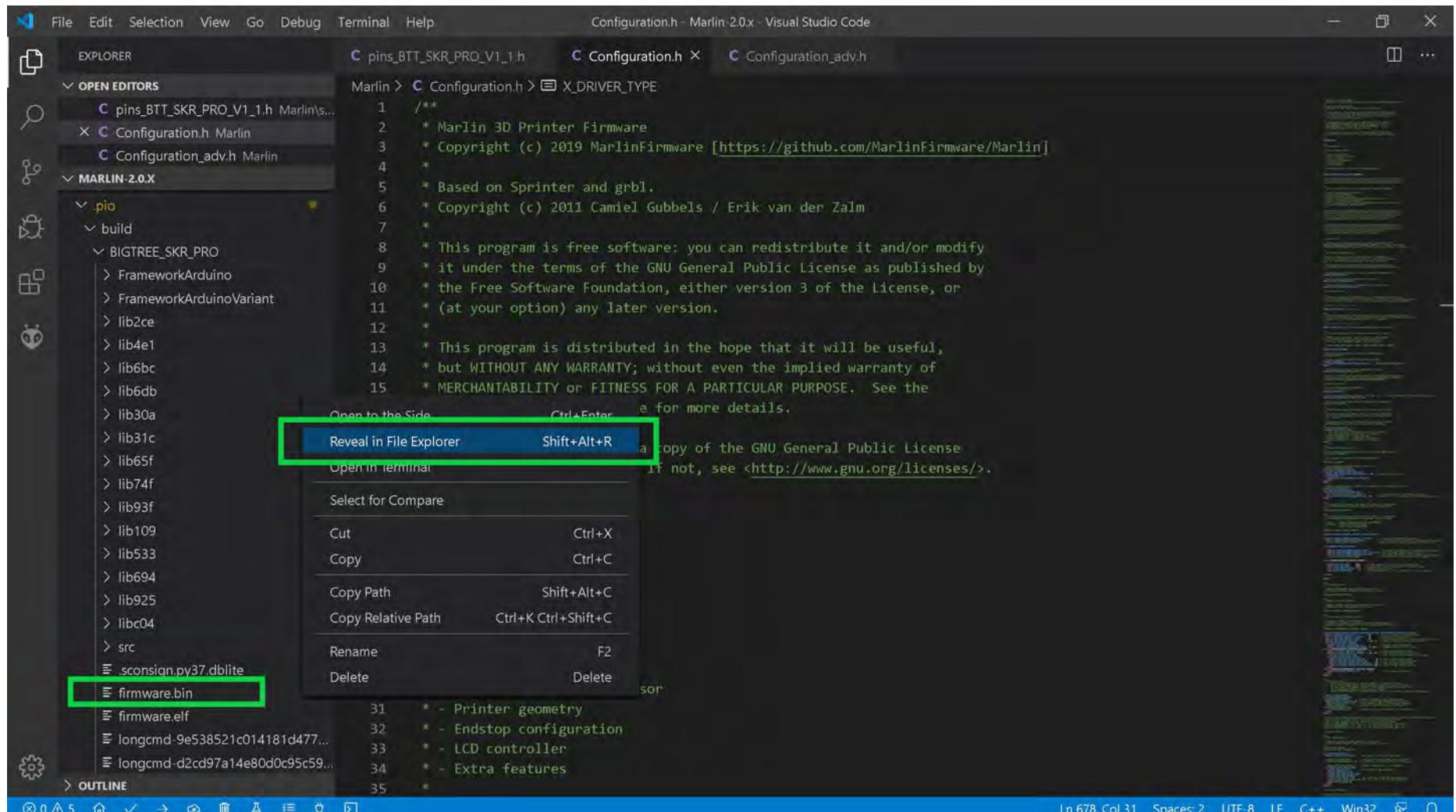
```

The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab selected. The terminal window displays the build log for Marlin 2.0.x. A yellow box highlights the checkmark icon in the bottom-left corner of the terminal tab. A green box highlights the terminal output, which shows a successful build with a duration of 00:02:31.294. The terminal also indicates that one task succeeded.

- 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 TMC2225 V1.0 Drivers in Stand-alone Mode

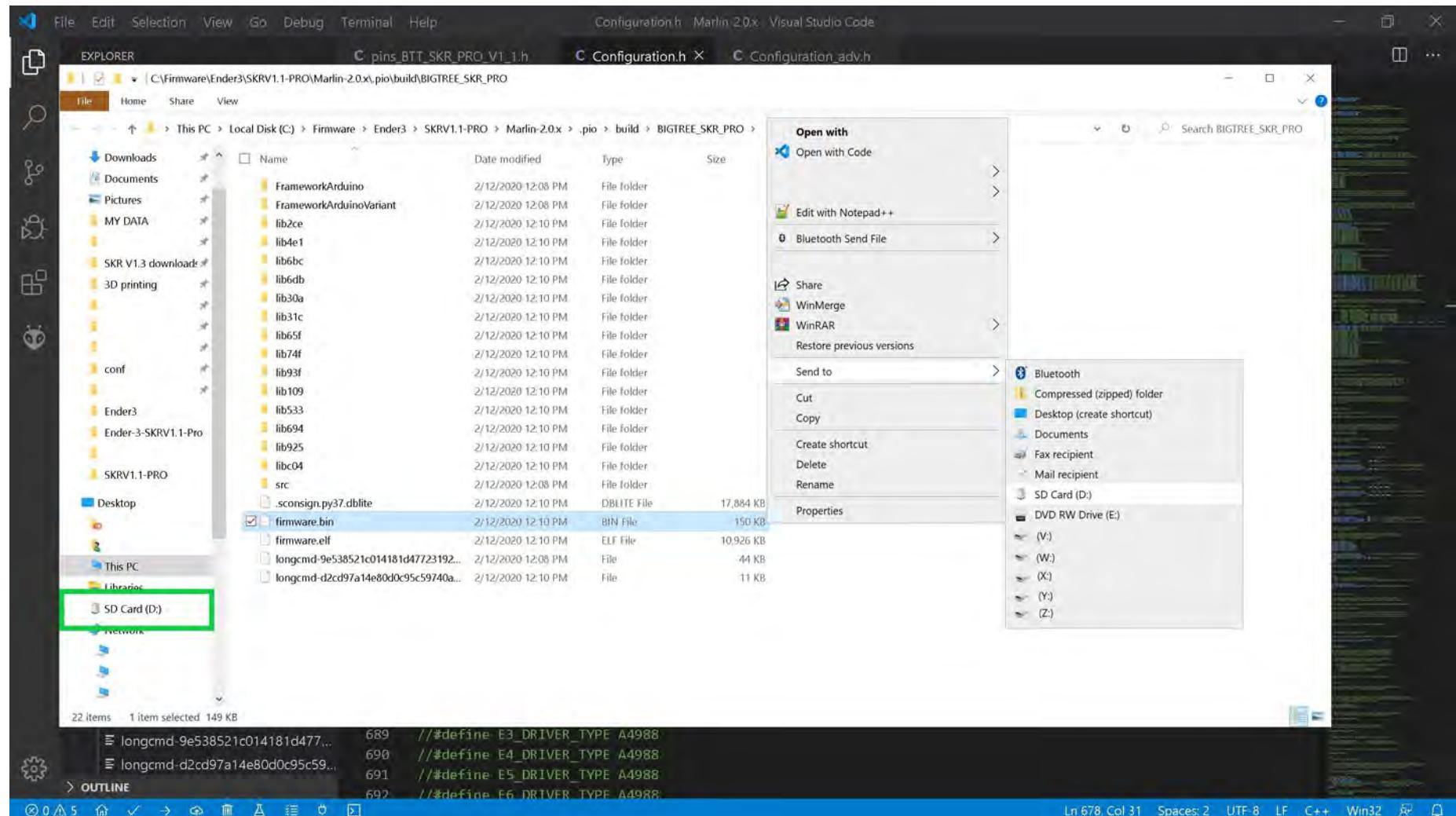
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



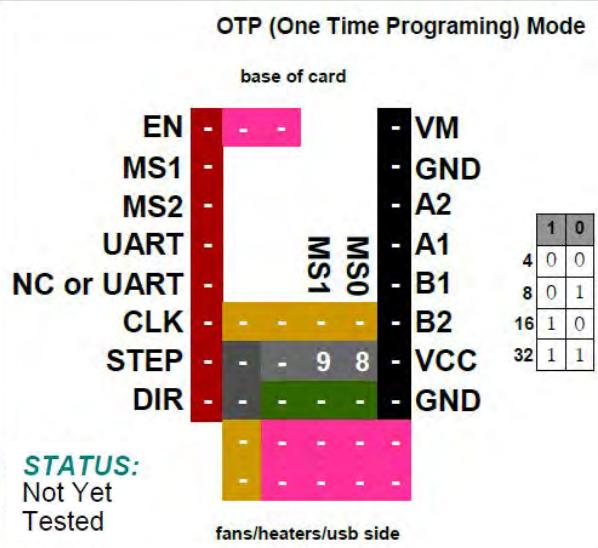
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

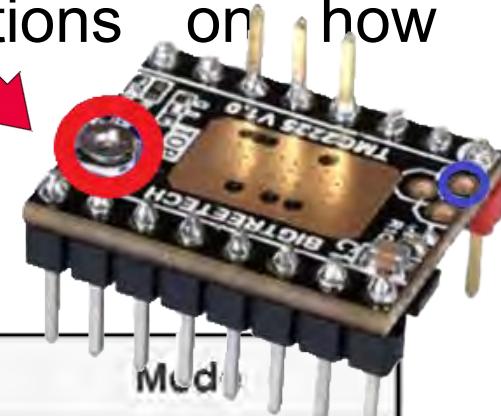
- Ensure your micro SD card, which came with your SKR PRO V1.1 board, with your SKR PRO V1.1 board is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
- From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

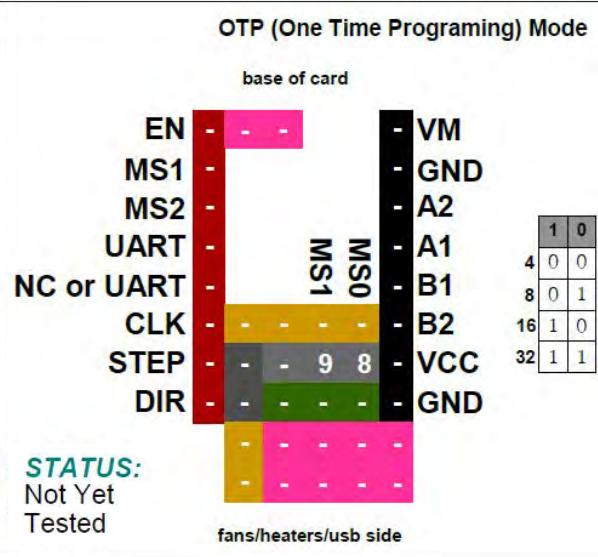
BIQU TMC2225 V1.0One Time Programming (OTP) Mode

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2225 <small>OTP Mode Maximum 32 Subdivision 35V DC 2A (peak)</small>	GND	GND	1 / 4	1 / 256	spreadCycle
	GND	VIO	1 / 8	1 / 256	spreadCycle
	VIO	GND	1 / 16	1 / 256	spreadCycle
	VIO	VIO	1 / 32	1 / 256	spreadCycle
Driving Current Calculation Formula <small>R_S (Typical Sense Resistor)=0.15Ω</small>	$I_{MAX}=V_{ref}*0.7222$ See Appendix B#10. Use 50% to 90% as shown below: $I_{MAX}=(V_{ref} * 0.7222) * 0.90$				$V_{ref}=I_{MAX}*1.3846$ See Appendix B#10. Use 50% to 90% as shown below: $V_{ref}=(I_{MAX} * 1.3846) * 0.90$

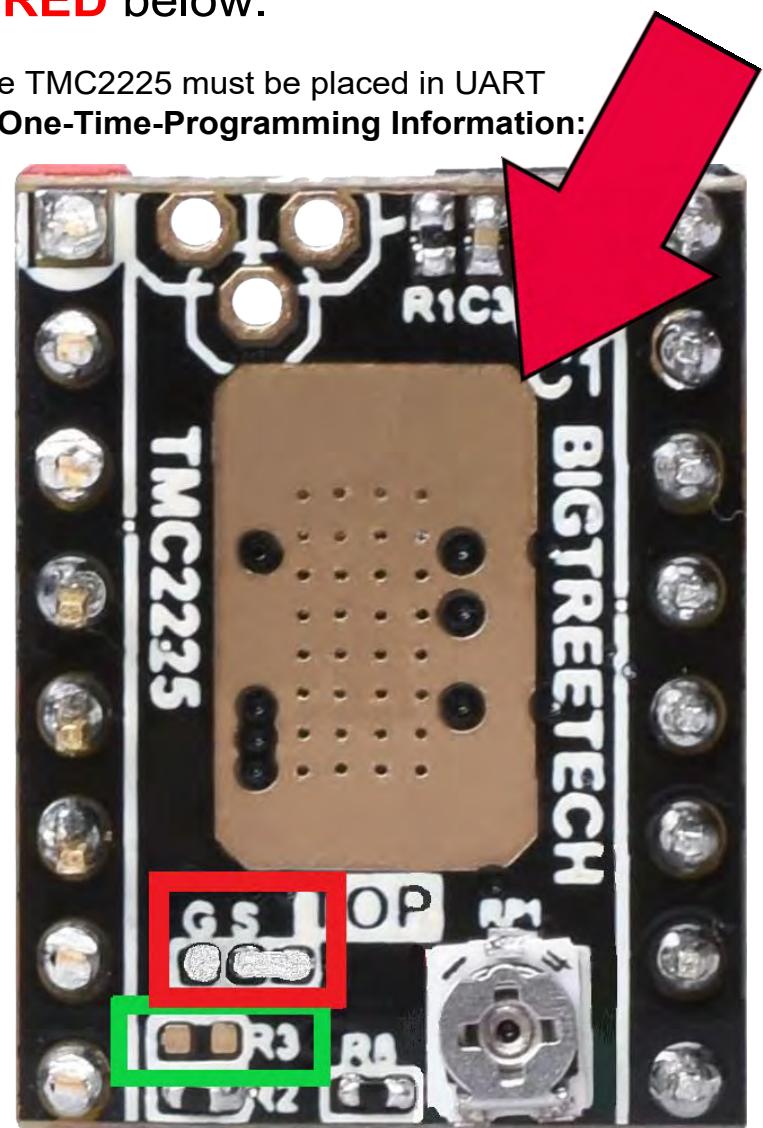
BIQU TMC2225 V1.0**One Time Programming (OTP) Mode**

The **PURPLE** boxes shows the location of the current sense resistors (R_s).

Note: MOST BIQU TMC2225 V1.0 driver boards, when purchased for UART mode, will have two R3 pads (located on the top of the driver board), which are NOT soldered together. This indicates the driver board can use the UART pin for the UART single wire interface (if the UART,"U", jumper is in place on the SKR PRO V1.1 board)

Note: Stand-alone mode by default uses stealthChop, if you want spreadCycle, you MUST use OTP mode. To obtain **One Time Programming (OTP) mode**, for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as shown in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for spread Cycle as seen in **RED** below.

To do One-Time-Programming (OTP), the TMC2225 must be placed in UART mode. See **TMC220x Configurator** for One-Time-Programming Information: [TMC220x Configurator](#).



OTP (One Time Programming) Mode

BIQU TMC2225 V1.0

One Time Programming (OTP) Mode

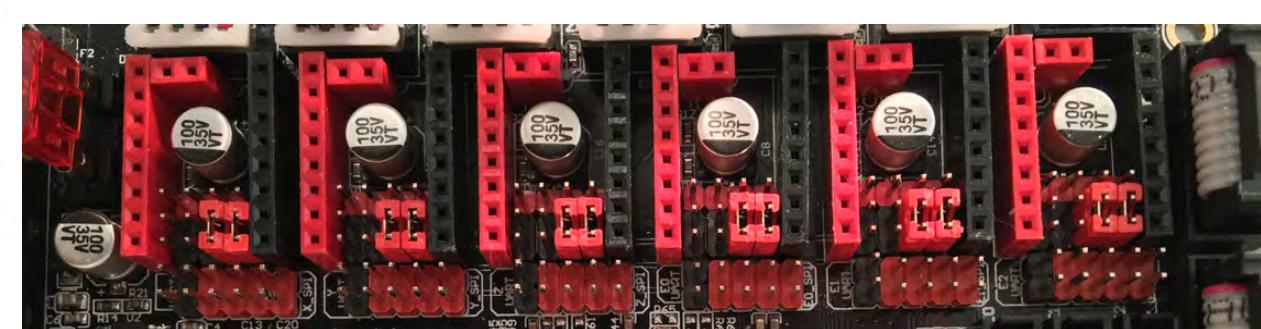
OTP

1 / 4

Interpolation: 1/256 NC or UART

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	MS1	A1
CLK	MS0	-	B1
STEP	-	9 8	B2
DIR	-	9 8	VCC
	-	-	GND



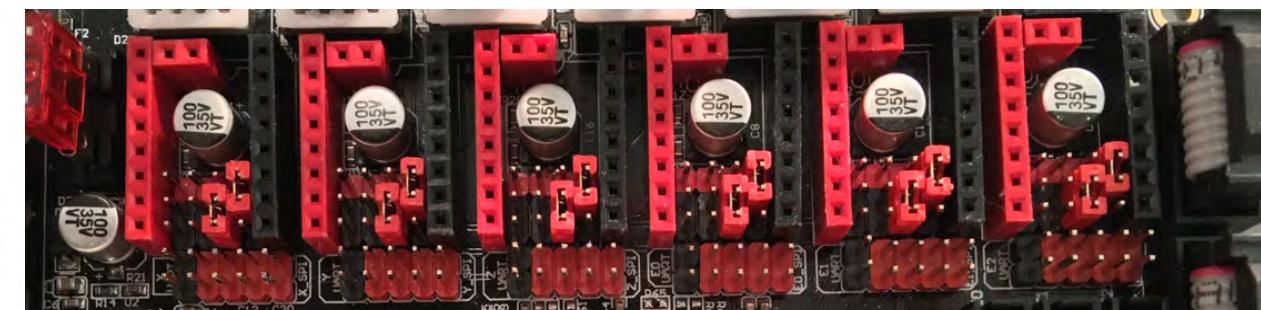
OTP

1 / 8

Interpolation: 1/256 NC or UART

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	MS1	A1
CLK	MS0	-	B1
STEP	-	8	B2
DIR	-	9 8	VCC
	-	-	GND



OTP (One Time Programming) Mode

BIQU TMC2225 V1.0

One Time Programming (OTP) Mode

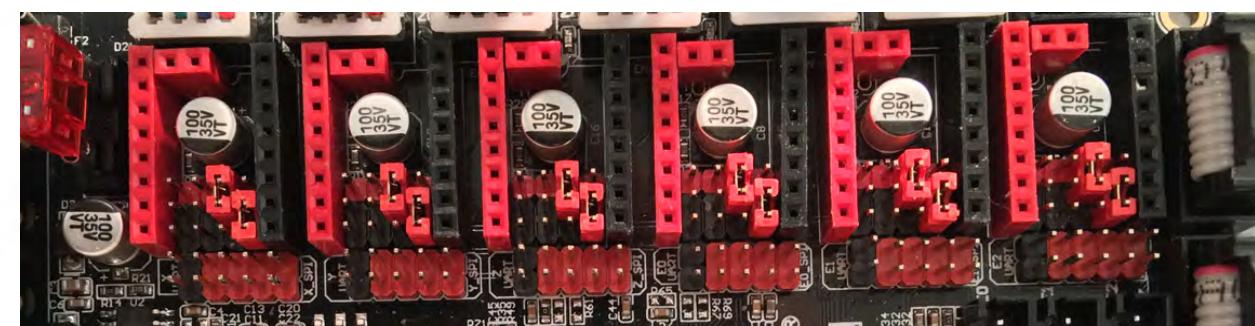
OTP

1 / 16

Interpolation: 1/256 NC or UART

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	MS1	A1
CLK	-	9	B1
STEP	-	9 8	B2
DIR	-	8	VCC
	-	-	GND



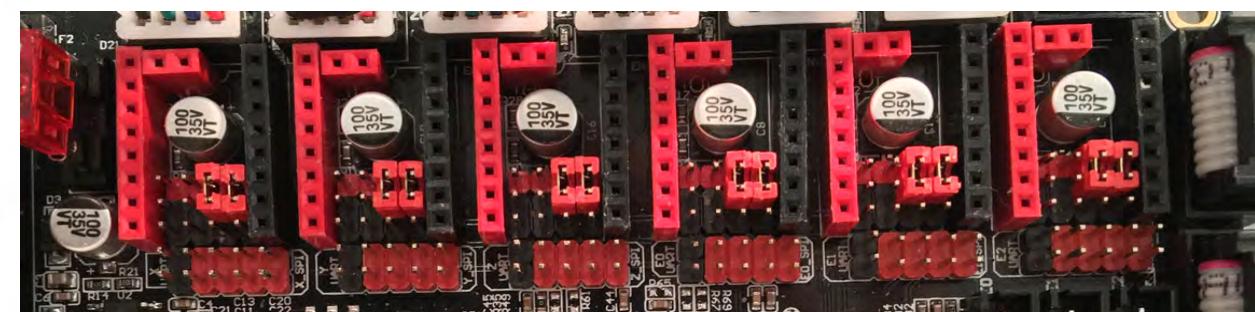
OTP

1 / 32

Interpolation: 1/256 NC or UART

SpreadCycle

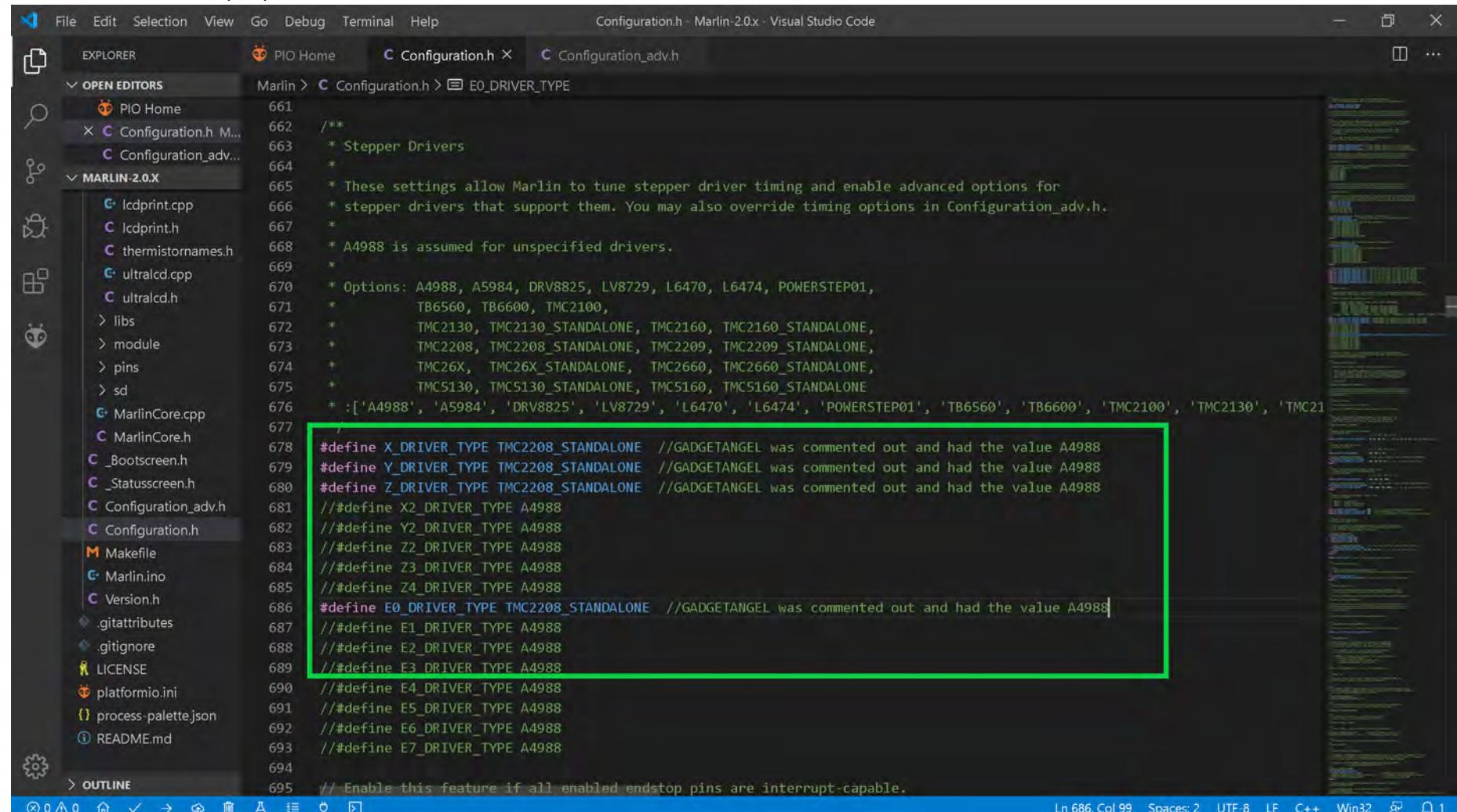
EN	-	-	VM
MS1	-	-	GND
MS2	-	-	A2
UART	-	MS1	A1
CLK	-	9 8	B1
STEP	-	9 8	B2
DIR	-	-	VCC
	-	-	GND



The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 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 BIQU TMC2225 drivers in OTP mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208_STANDALONE" for the DRIVER_TYPE. 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 ("//").



```

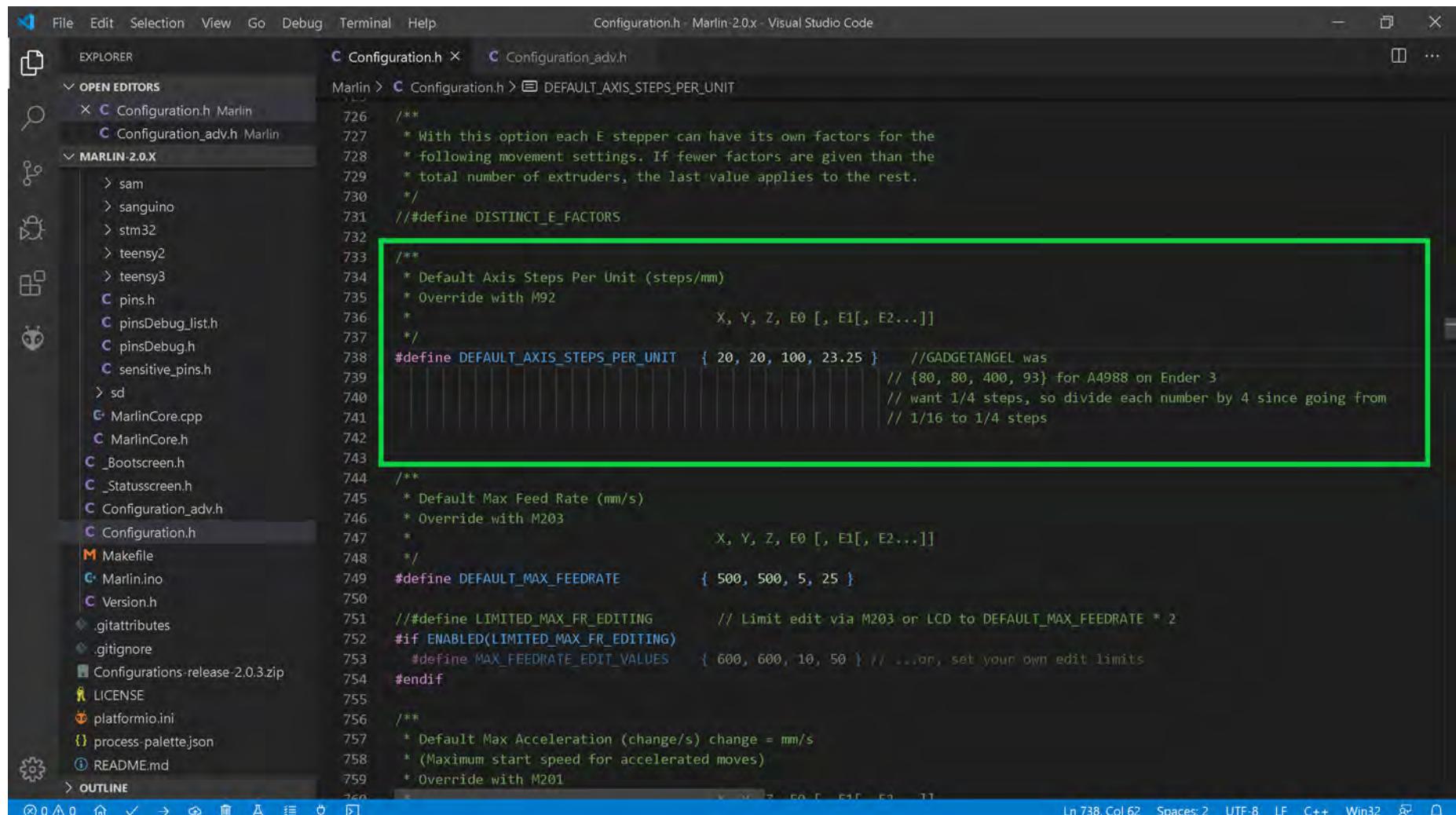
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 * A4988 is assumed for unspecified drivers.
667 *
668 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
669 * TB6560, TB6600, TMC2100,
670 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
671 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
672 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
673 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
674 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2
675 */
676 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
677 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 //##define X2_DRIVER_TYPE A4988
680 //##define Y2_DRIVER_TYPE A4988
681 //##define Z2_DRIVER_TYPE A4988
682 //##define Z3_DRIVER_TYPE A4988
683 //##define Z4_DRIVER_TYPE A4988
684 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
685 //##define E1_DRIVER_TYPE A4988
686 //##define E2_DRIVER_TYPE A4988
687 //##define E3_DRIVER_TYPE A4988
688 //##define E4_DRIVER_TYPE A4988
689 //##define E5_DRIVER_TYPE A4988
690 //##define E6_DRIVER_TYPE A4988
691 //##define E7_DRIVER_TYPE A4988
692
693
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 TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. 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 {20, 20, 100, 23.25}, 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h DEFAULT_AXIS_STEPS_PER_UNIT
sam
sanguino
stm32
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarinCore.cpp
MarinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
Configurations-release-2.0.3.zip
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 738, Col 62 Spaces: 2 UTF-8 LF C++ Win32
20 20 100 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 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 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 OPEN EDITORS MARLIN-2.0.X
  Configuration.h X Configuration_adv.h
  Configuration.h Marlin Configuration_adv.h Marlin
  sanguino
  stm32
  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
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palettejson
  README.md

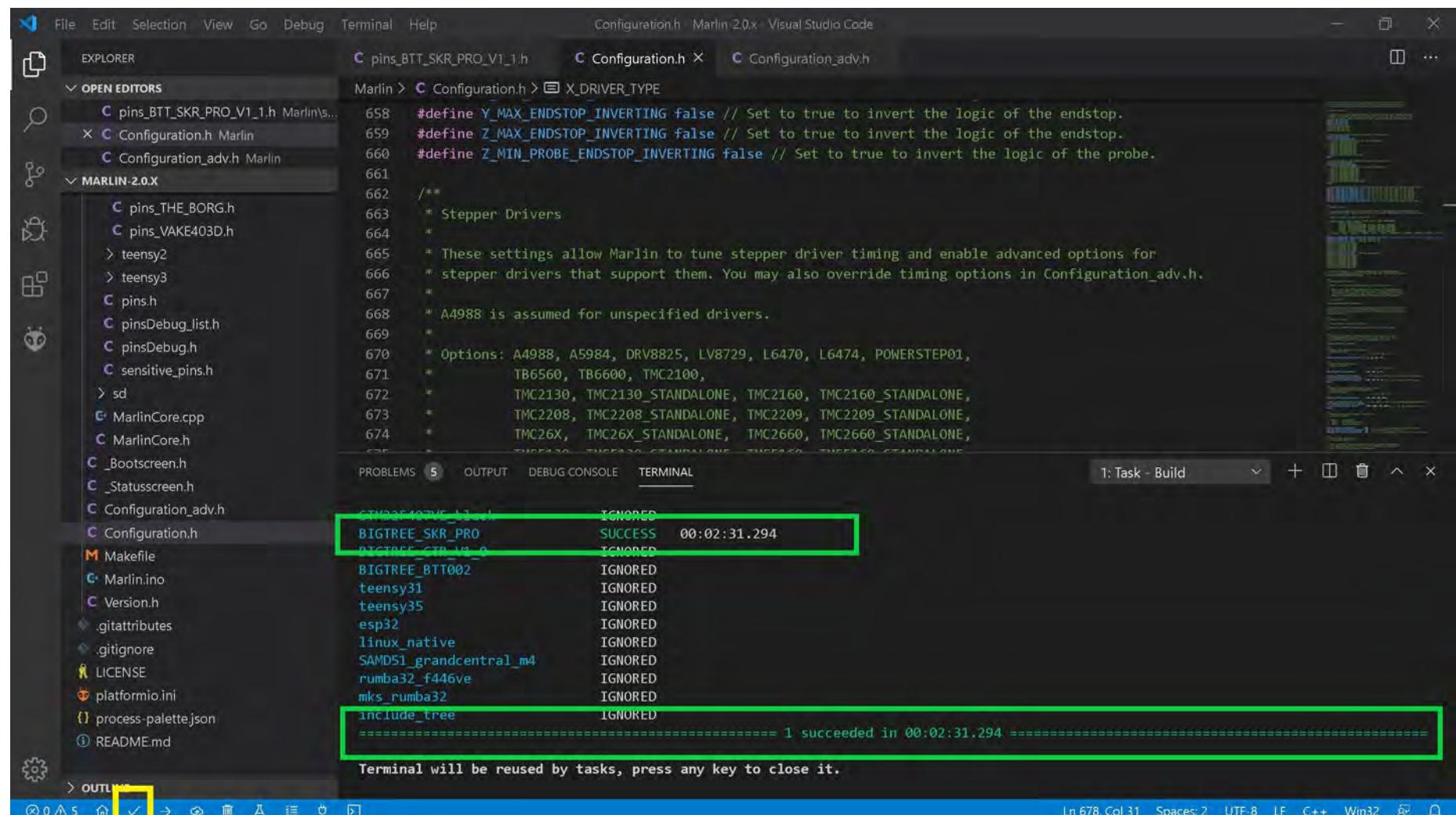
OUTLINE

C Configuration.h X C Configuration_adv.h
Marlin > C Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 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 X_DRIVER_TYPE
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 * 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 * TMC2661, TMC2661_STANDALONE, TMC2662, TMC2662_STANDALONE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
Task: 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

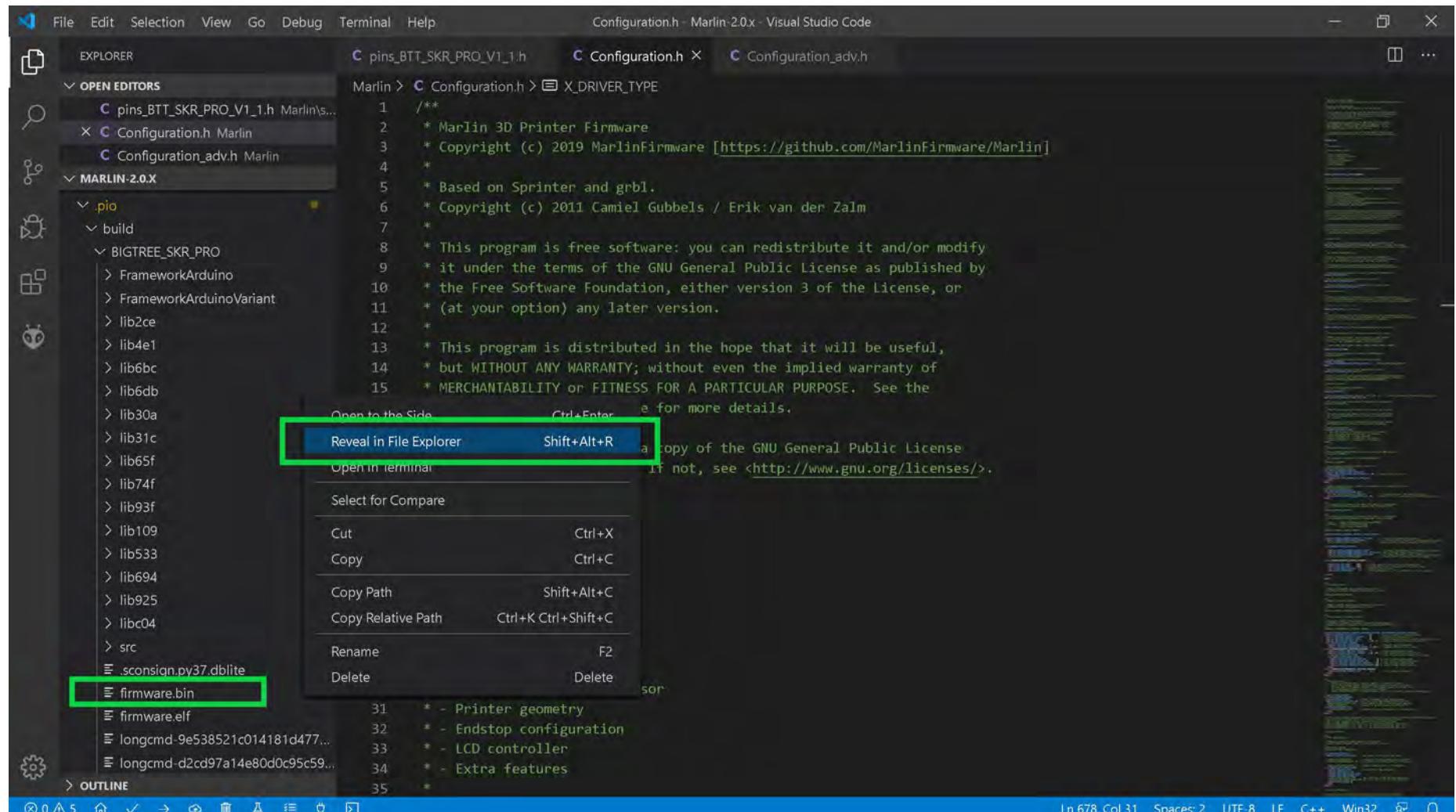
```

The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The terminal tab is active, displaying the build log. The log shows a successful build with a duration of 00:02:31.294. The output is highlighted with green boxes. The status bar at the bottom indicates the terminal will be reused by tasks.

- 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 TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

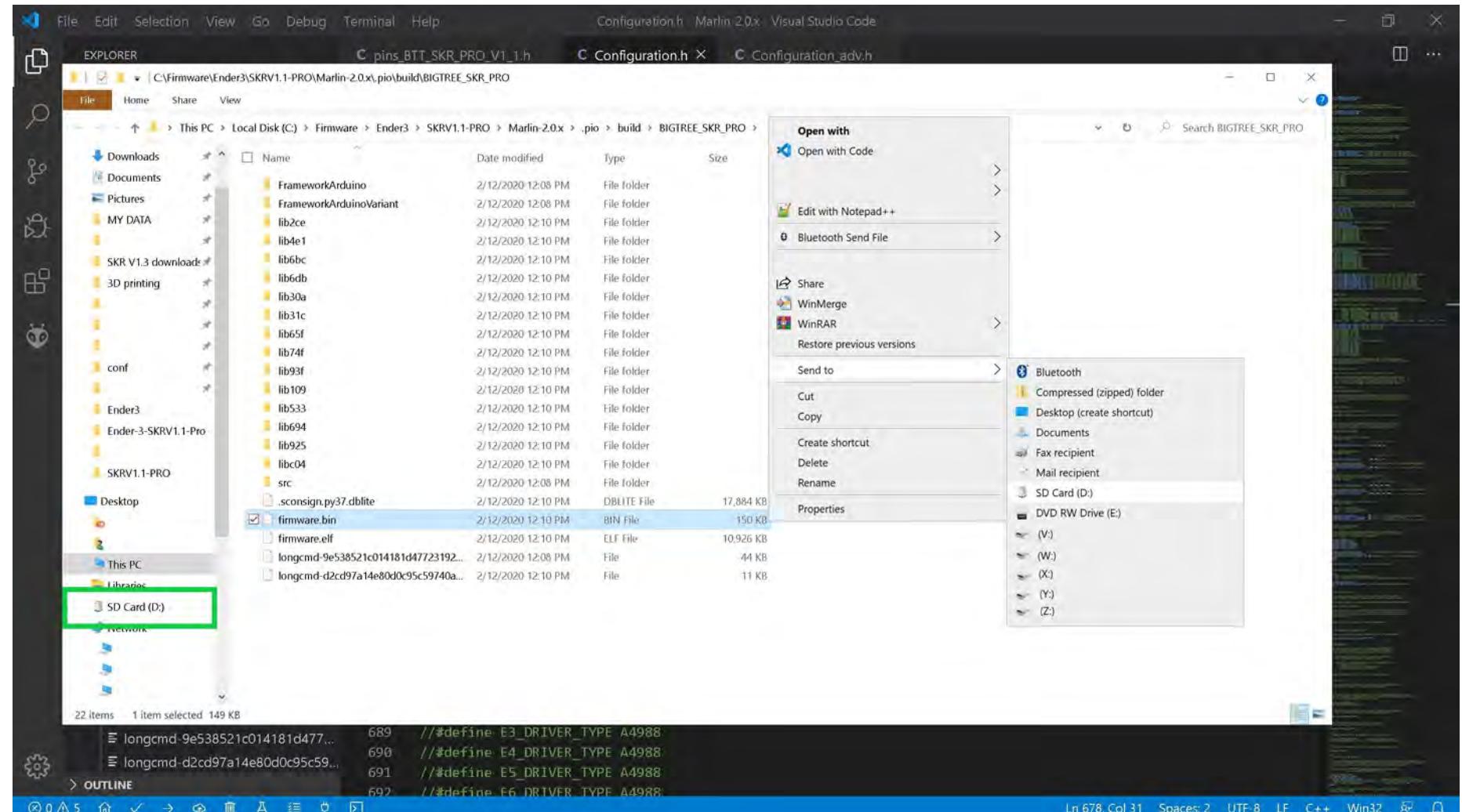
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



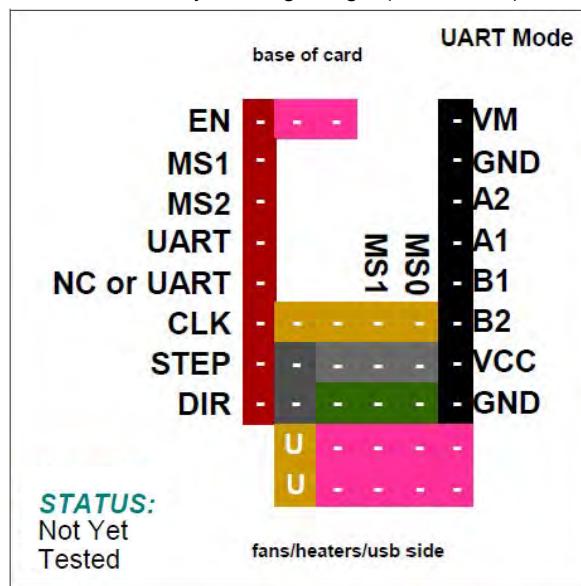
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



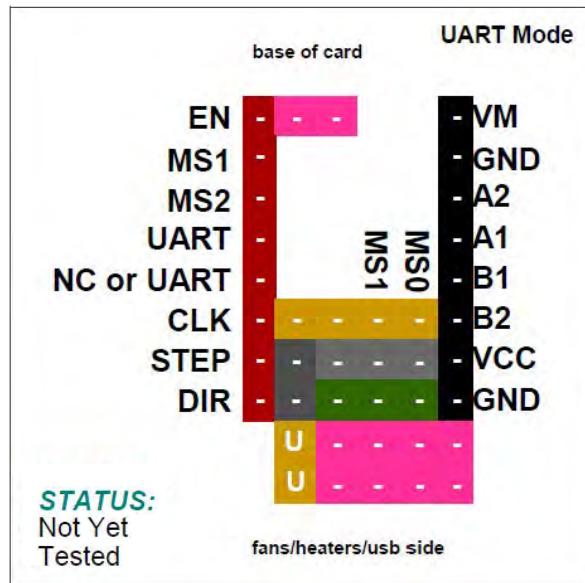
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

**BIQU TMC2225 V1.0****UART Mode**

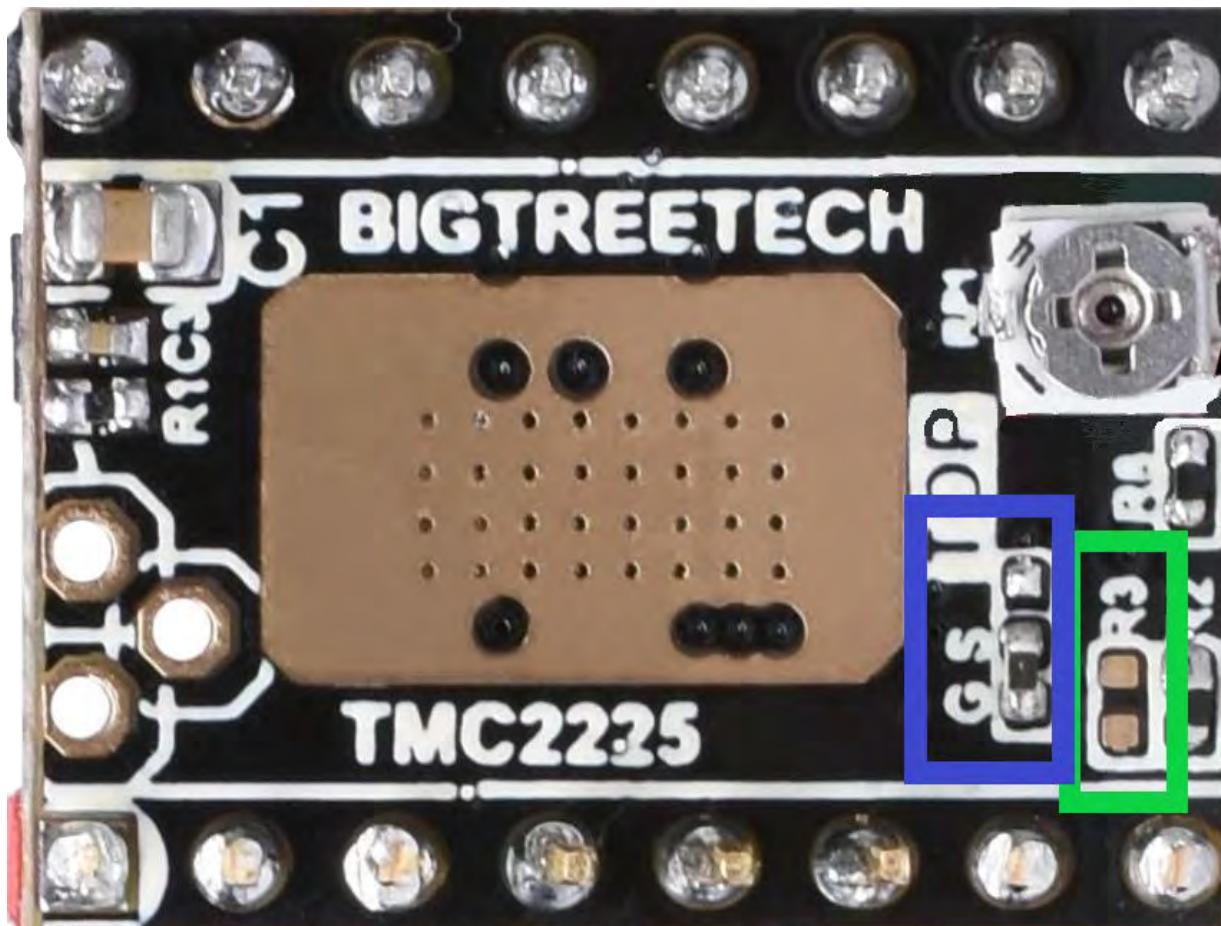
Note: You can use 50% to 90% of the calculated I_{RMS} ($I_{MAX}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

Driver Chip	Steps are set inside of your Firmware	
BIQU® TMC2225 UART Mode Maximum 256 Subdivision 35V DC 2A (peak)		
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.15Ω	$I_{MAX} = V_{ref} * 0.7647$ See Appendix B #7. Use 50% to 90% as shown below: $I_{MAX} = (V_{ref} * 0.7647) * 0.90$	$V_{ref} = I_{MAX} * 1.3077$ See Appendix B #7. Use 50% to 90% as shown below: $V_{ref} = (I_{MAX} * 1.3077) * 0.90$

**BIQU TMC2225 V1.0****UART Mode**

Important: To ensure that the BIQU TMC2225 V1.0 is in UART Mode, check to see if the two pads located at R3 have a gap between them, as seen in **GREEN** below. Ensure the "U" jumper is in place on the SKR PRO V1.1 board.



The **BLUE** box show the device has StealthChop capability.

UART Mode**UART Mode**

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

R_s = R050 is 0.05 Ohms

R_s = R062 is 0.062 Ohms

R_s = R068 is 0.068 Ohms

R_s = R075 is 0.075 Ohms

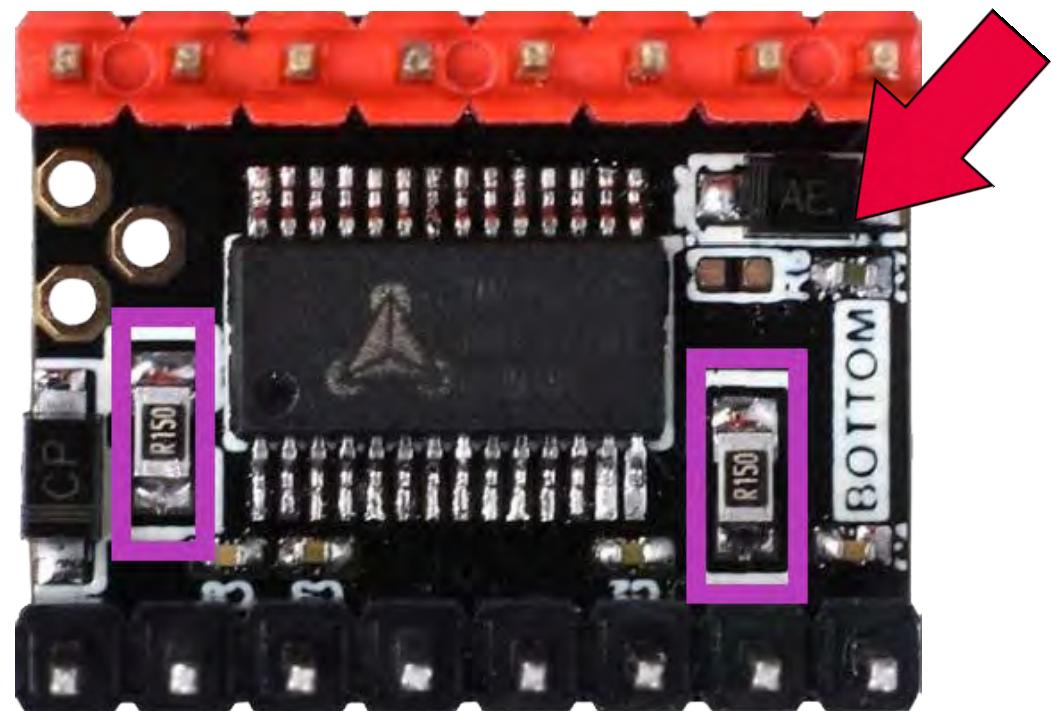
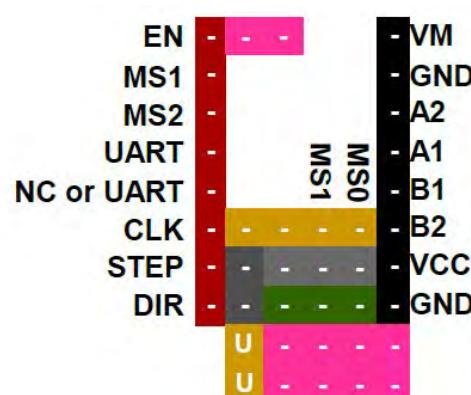
R_s = R100 is 0.1 Ohms

R_s = R110 is 0.11 Ohms

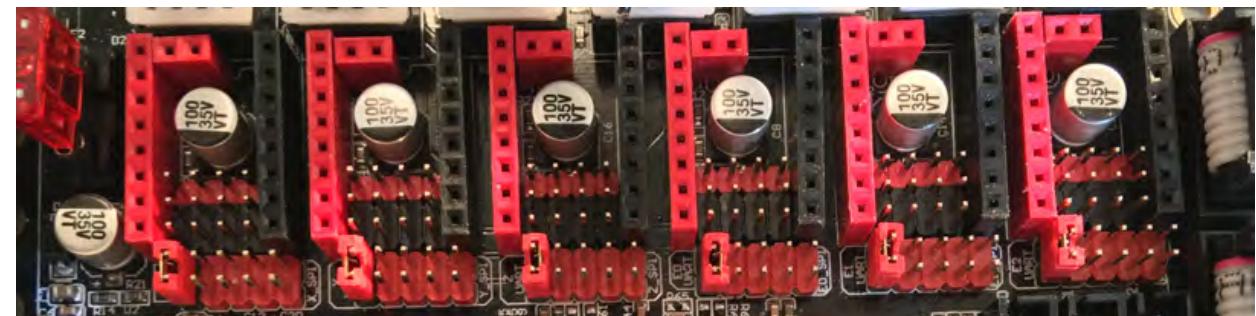
R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

R_s = R220 is 0.22 Ohms

**UART**

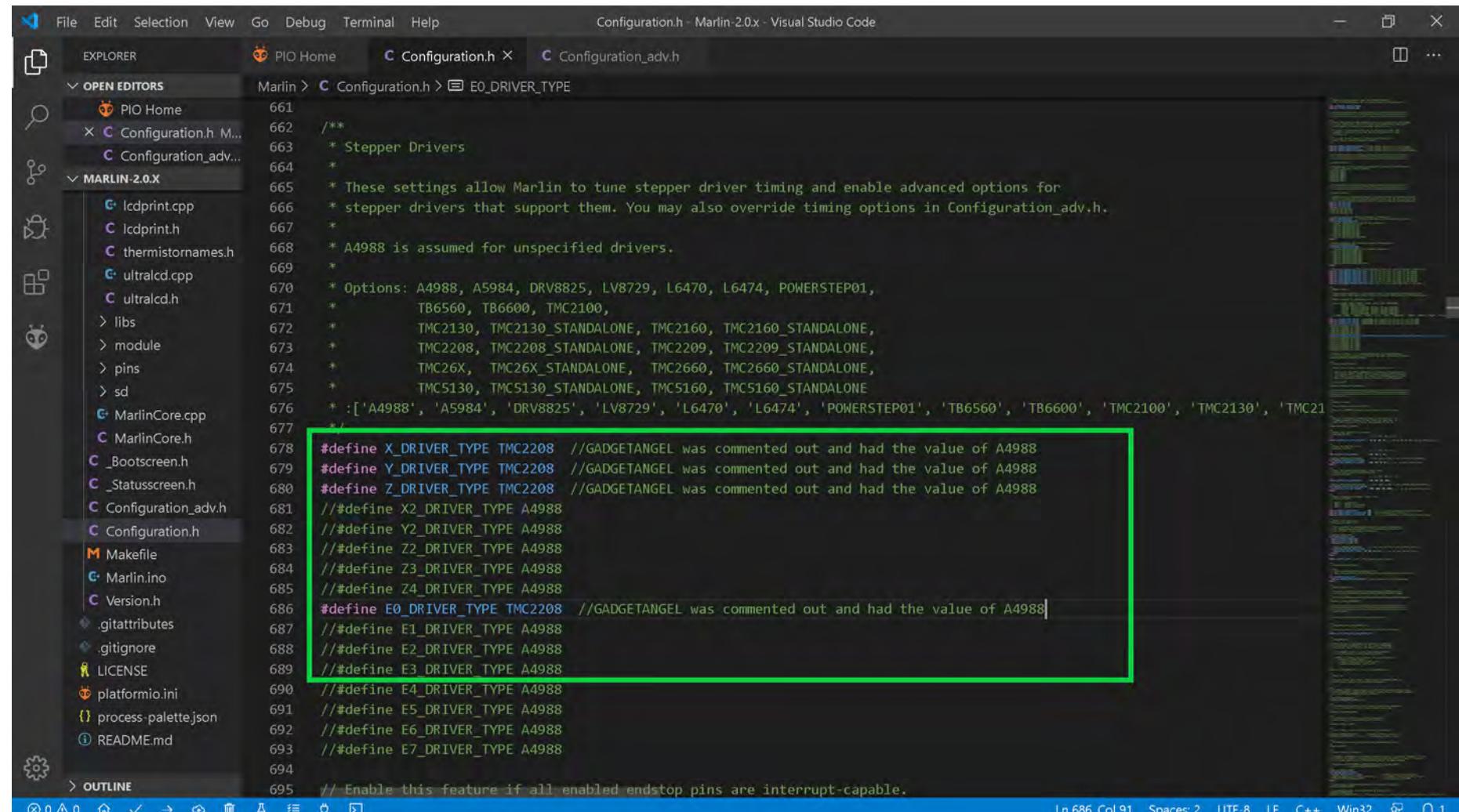
Note: Set Jumper "U" for UART MODE!!



The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

NOTE: [Go to Appendix C](#), and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2225 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 TMC2225 drivers in UART mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208" for the DRIVER_TYPE. 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-2.0x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
  PIO Home
  Configuration.h M...
  Configuration_adv.h
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
Ln 686, Col 91  Spaces: 2  UTF-8  LF  C++  Win32  ⚡  1

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', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
677 */
678 #define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
679 #define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
680 #define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the 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 TMC2208 //GADGETANGEL was commented out and had the 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 BIQU TMC2225 V1.0 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 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 TMC2225 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 following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Sidebar:** EXPLORER, OPEN EDITORS, MARLIN-2.0.X (containing sanguino, stm32, 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, Configurations-release-2.0.3.zip, LICENSE, platformio.ini, process-palettejson, README.md).
- Editor Area:** The Configuration.h file is open. A green box highlights the following code block:

```
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR false          // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Y_DIR false          // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Z_DIR true           // GADGETANGEL was false, stepper motor driver inverts motor direction
```
- Bottom Status Bar:** Ln 1051 Col 107 Spaces: 2 UTF-8 LF C++ Win32.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2225 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.15. Ensure "Y_RSENSE" is set to 0.15.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".

```

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

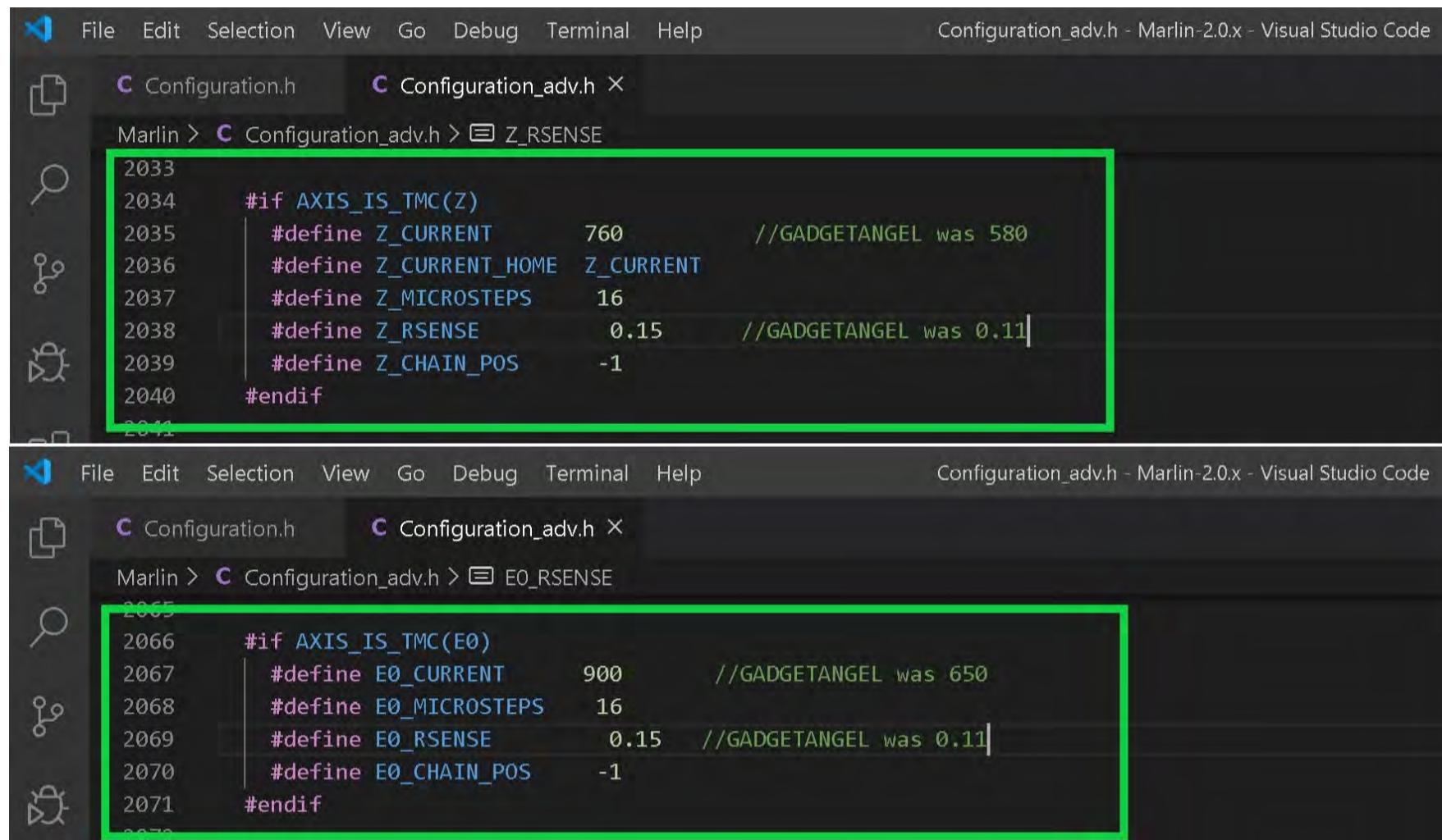
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_RSENSE
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current 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. //GADGETANGEL was 580
2004     #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005     #define X_MICROSTEPS 16 // 0..256
2006     #define X_RSENSE 0.15 //GADGETANGEL was 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 //GADGETANGEL was 580
2020     #define Y_CURRENT_HOME Y_CURRENT
2021     #define Y_MICROSTEPS 16
2022     #define Y_RSENSE 0.15 //GADGETANGEL was 0.11
2023     #define Y_CHAIN_POS -1
2024 #endif
2025
2026 //LFB: ATC TS TMC2225

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.15. Ensure "E0_RSENSE" is set to 0.15.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z_RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035     #define Z_CURRENT      760          //GADGETANGEL was 580
2036     #define Z_CURRENT_HOME Z_CURRENT
2037     #define Z_MICROSTEPS   16
2038     #define Z_RSENSE        0.15        //GADGETANGEL was 0.11
2039     #define Z_CHAIN_POS    -1
2040 #endif
2041

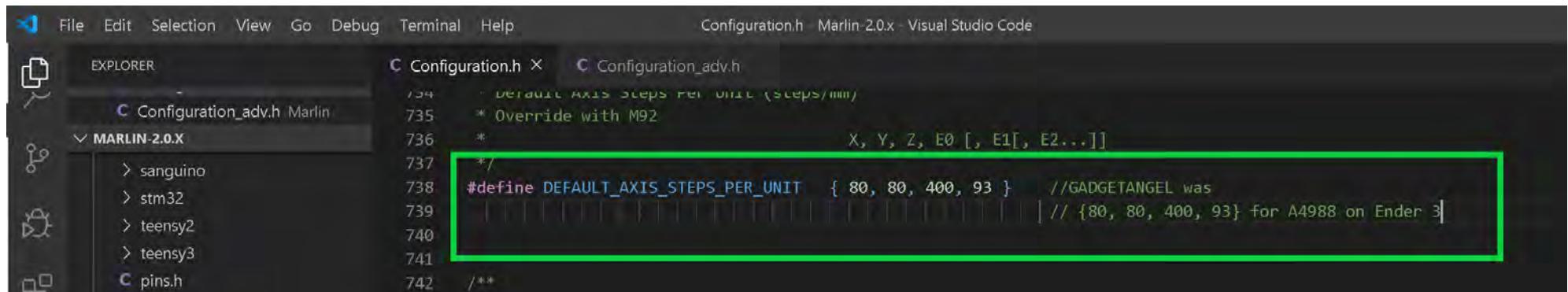
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > E0_RSENSE
2065
2066 #if AXIS_IS_TMC(E0)
2067     #define E0_CURRENT      900          //GADGETANGEL was 650
2068     #define E0_MICROSTEPS   16
2069     #define E0_RSENSE        0.15        //GADGETANGEL was 0.11
2070     #define E0_CHAIN_POS    -1
2071 #endif
2072

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- 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 Configuration.h X Configuration_adv.h

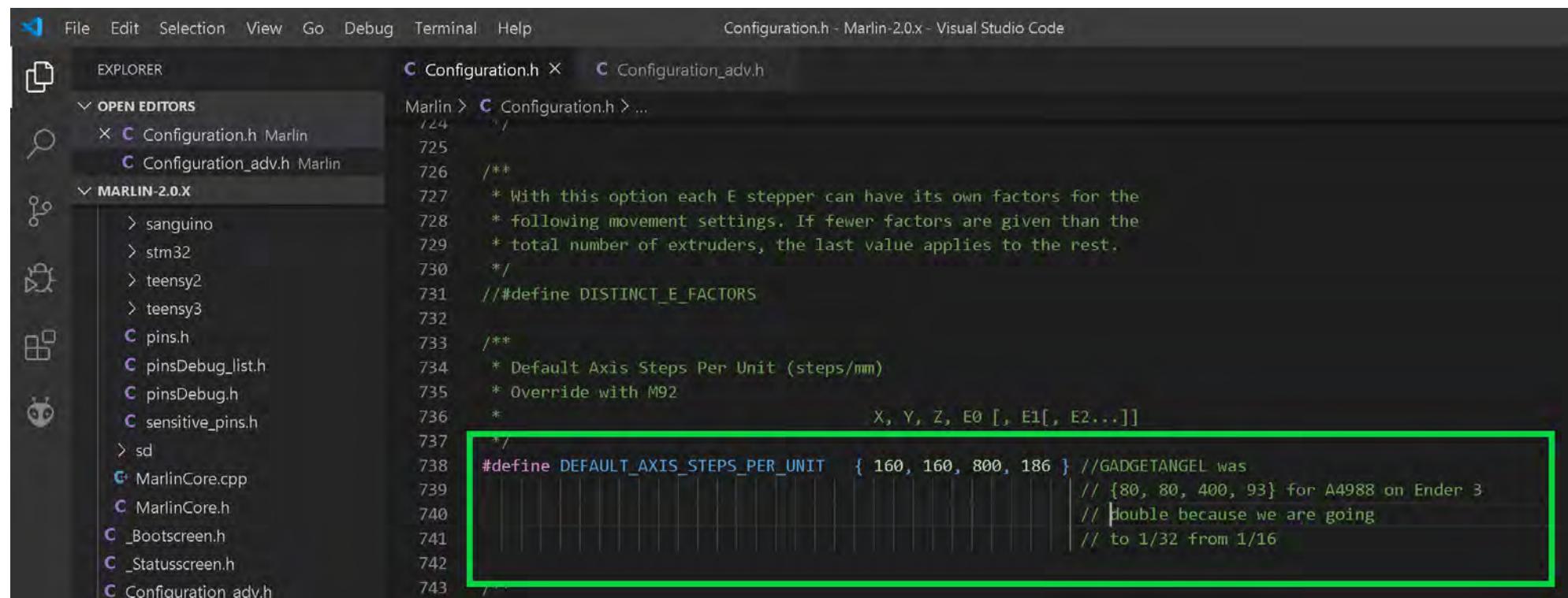
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

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 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 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 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

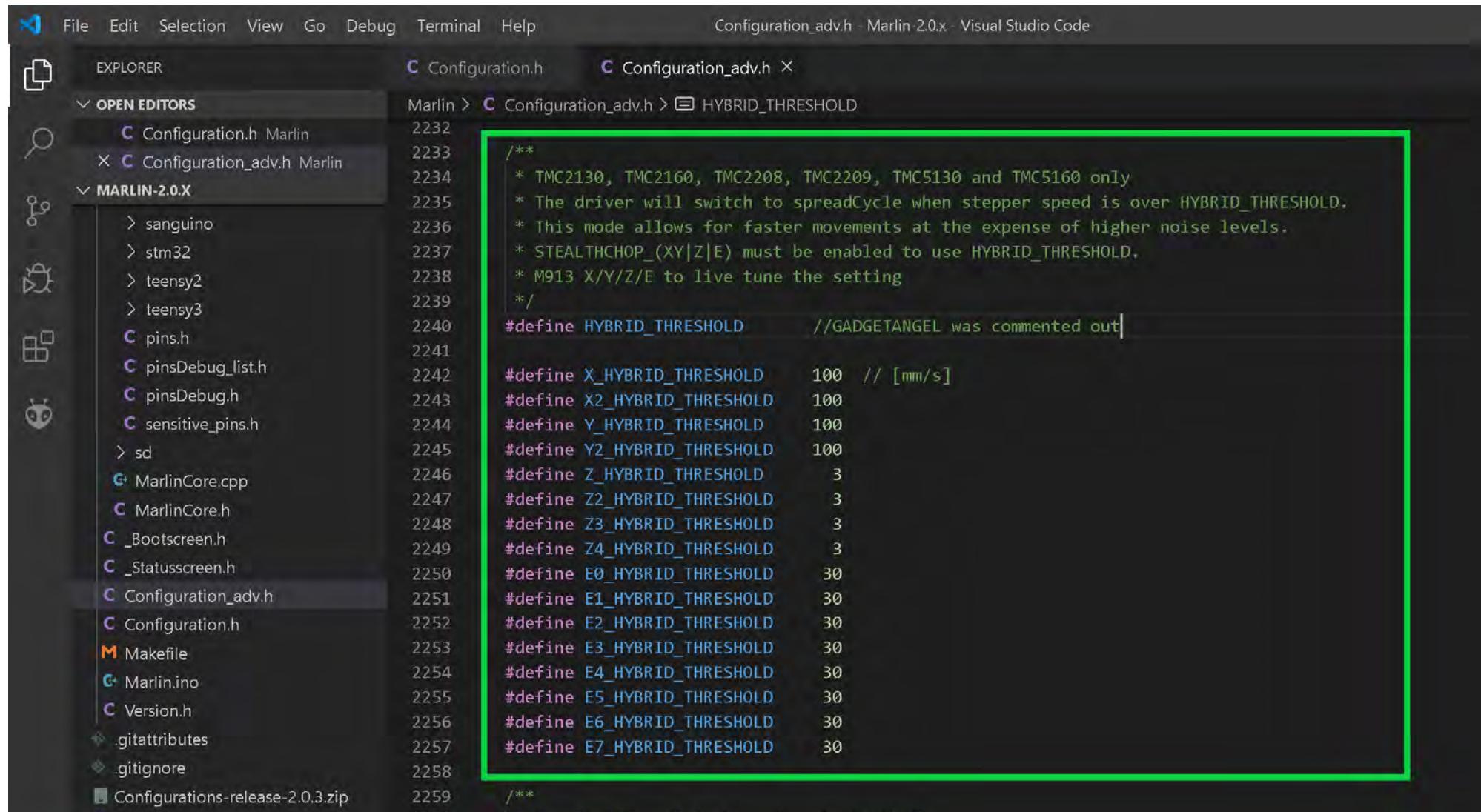
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  C pins.h
  C pinsDebug_list.h
  C pinsDebug.h
  C sensitive_pins.h
  > sd
  C MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C _Statusscreen.h
  C Configuration_adv.h
  C Configuration.h
Marlin > C Configuration_adv.h > STEALTHCHOP_XY
2193 /**
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
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 /**

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 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 following configuration for the BIQU TMC2225 V1.0 Drivers in UART Mode:

```

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

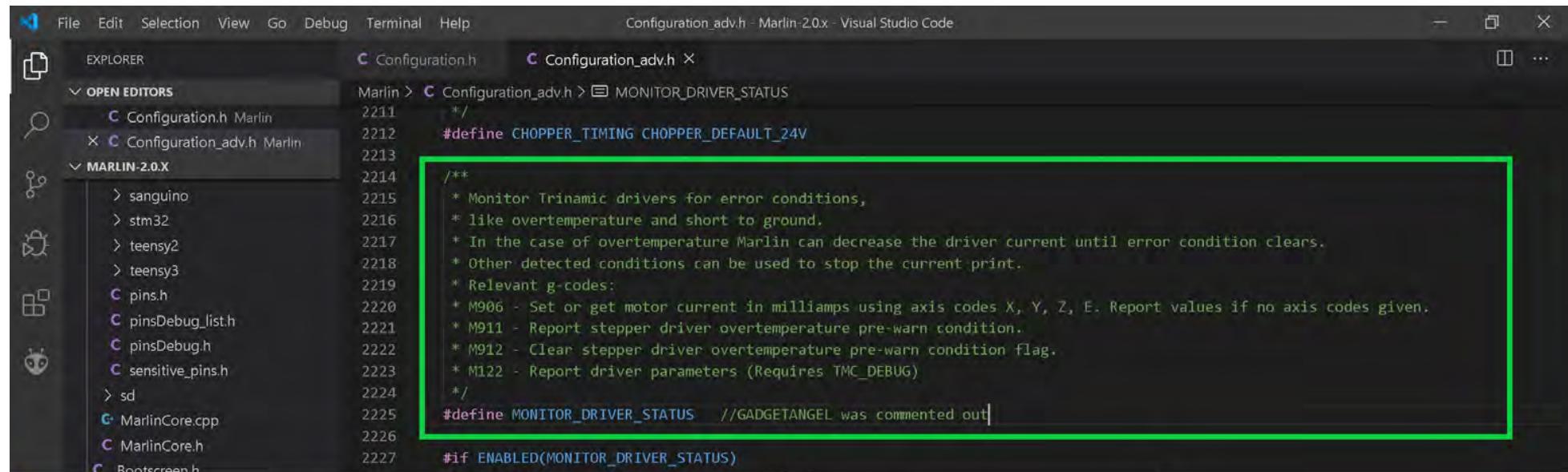
```

A green rectangular box highlights the block of code starting with `/*` and ending with `#define HYBRID_THRESHOLD`. The line `//GADGETANGEL was commented out` is visible after the `#define` line. The code editor's status bar indicates the file is Marlin 2.0.x - Visual Studio Code.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART 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.](#)



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

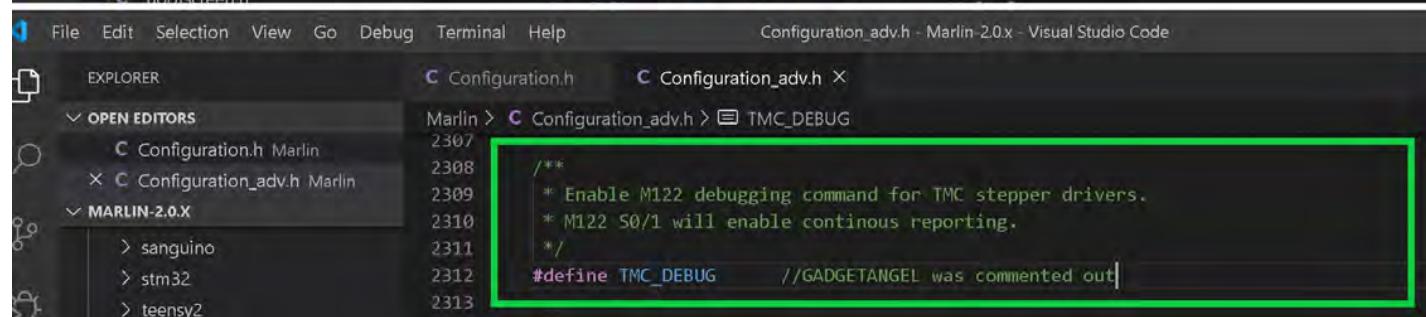
EXPLORER Configuration.h Configuration_adv.h

Marlin > Configuration.h > MONITOR_DRIVER_STATUS

```

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 millamps 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 //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



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

EXPLORER Configuration.h Configuration_adv.h

Marlin > Configuration_adv.h > TMC_DEBUG

```

2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

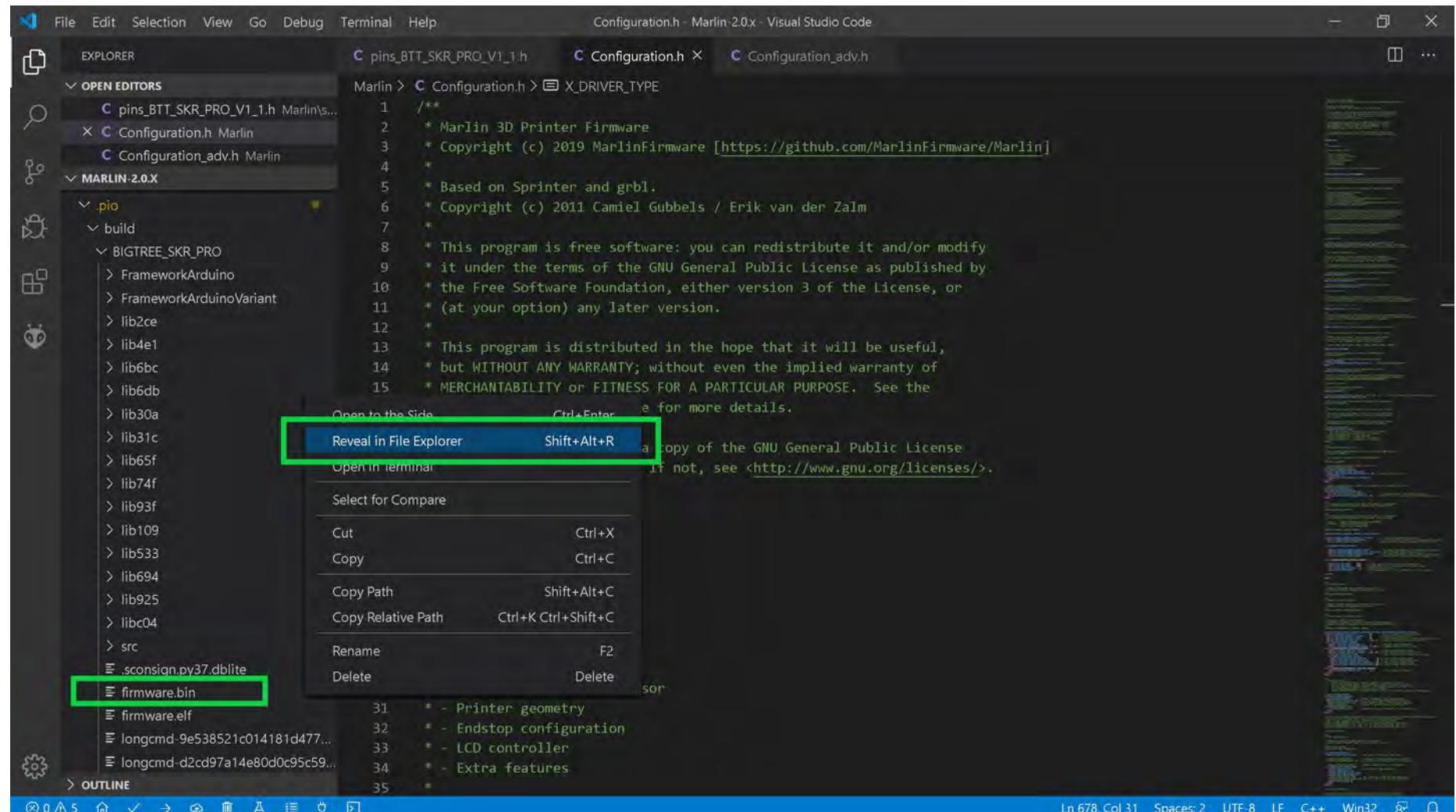
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 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.

- 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 TMC2225 V1.0 Drivers in UART Mode

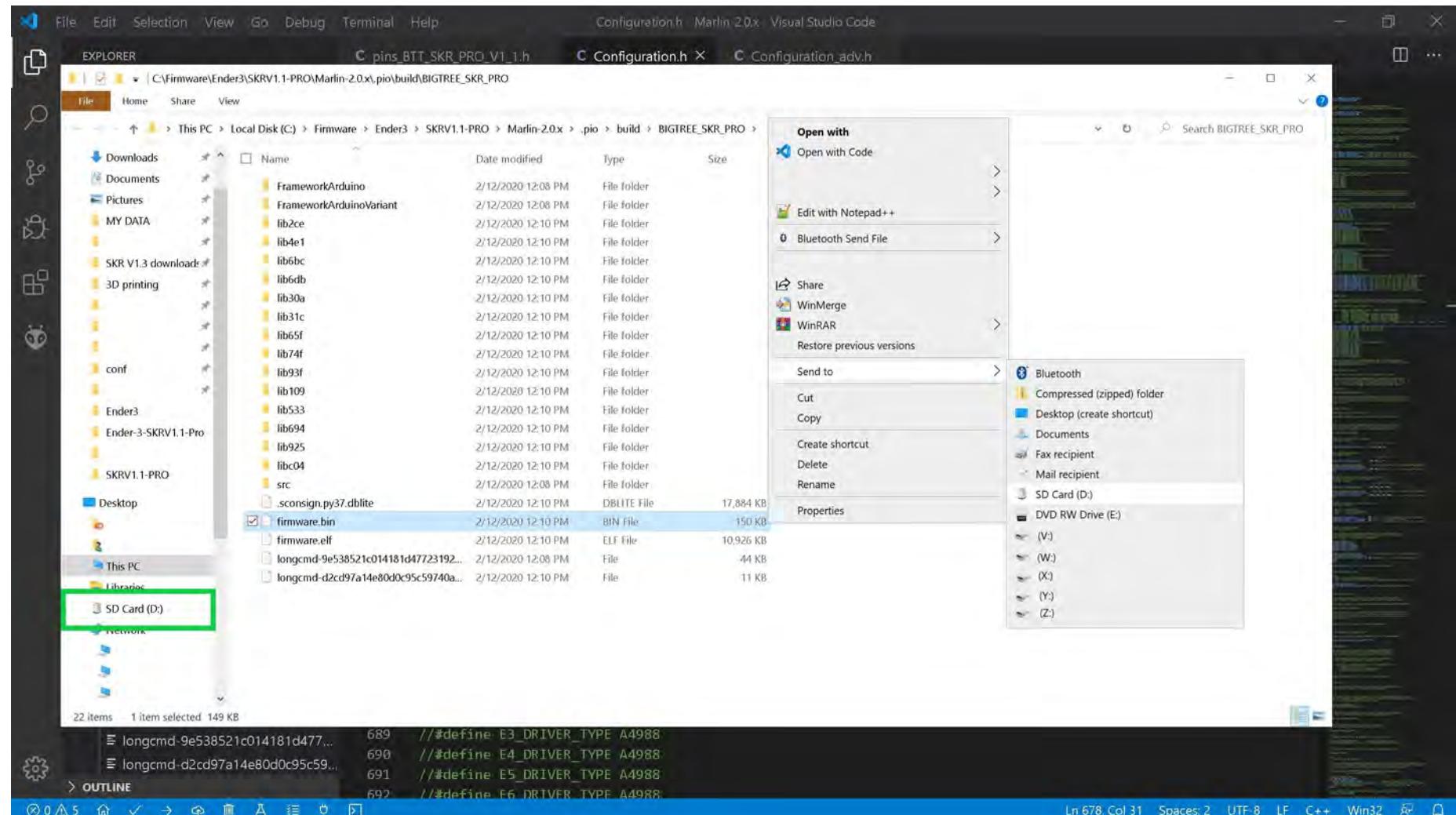
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



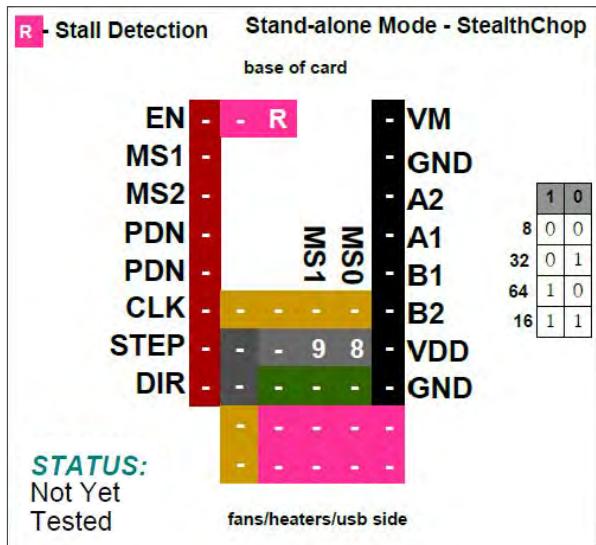
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.



BIQU TMC2209 V1.2

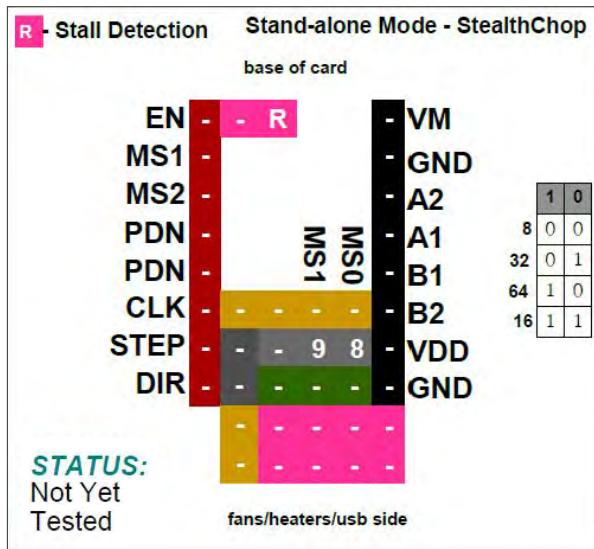
Stand-alone Mode for StealthChop

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Driver Chip	MS1	MS0	Steps	Interpolation	Mode
 TMC2209 Stand Alone Mode Maximum 64 Subdivision 28V DC 2.8A (peak)	GND	GND	1 / 8	1 / 256	stealthChop
	GND	VIO	1 / 32	1 / 256	stealthChop
	VIO	GND	1 / 64	1 / 256	stealthChop
	VIO	VIO	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S (Typical Sense Resistor)= 0.11Ω	$I_{MAX} = V_{ref}$ See Appendix B #5. Use 50% to 90% as shown below:			$V_{ref} = I_{MAX}$ See Appendix B #5. Use 50% to 90% as shown below:	
	$I_{MAX} = I_{MAX} * 0.90$			$V_{ref} = V_{ref} * 0.90$	



BIQU TMC2209 V1.2

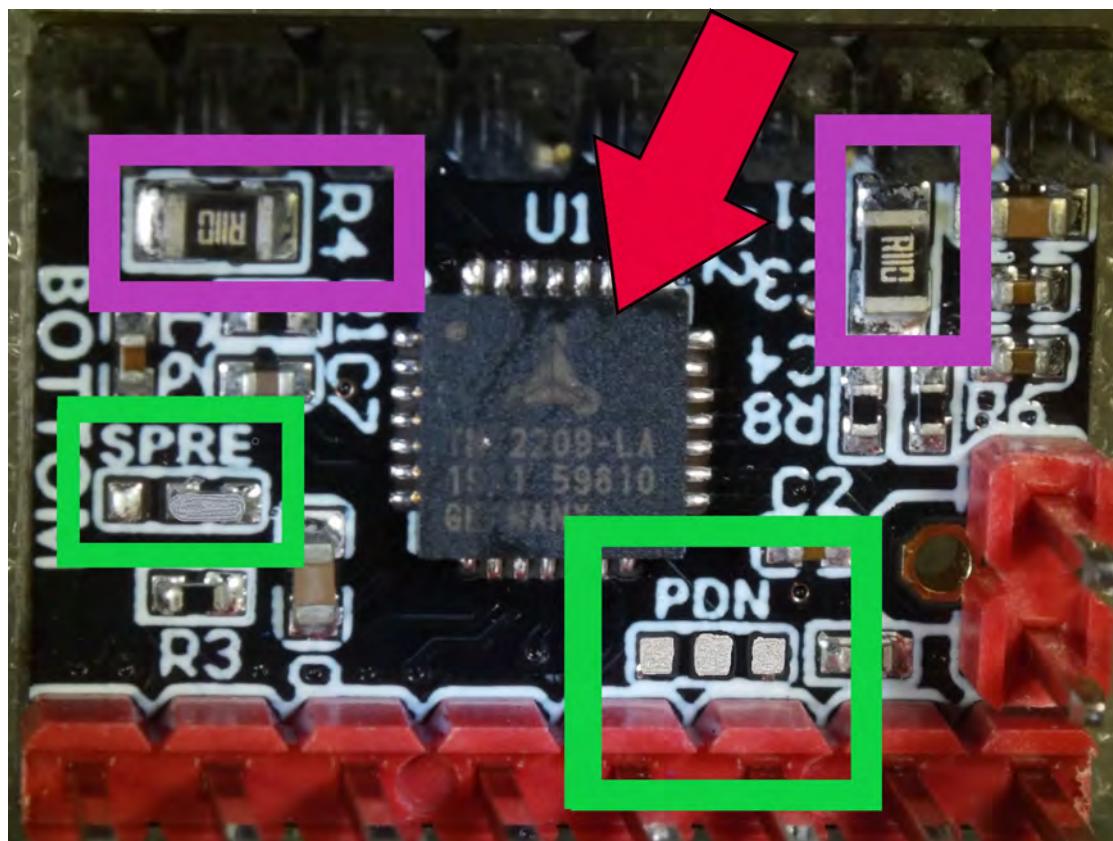
Stand-alone Mode for StealthChop

NOTE: The SPRE jumper is located on the bottom of the driver board. In stand-alone 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. The **PURPLE boxes** below show the location of the current sense resistors (R_s) for the TMC2209.

Note: To switch to stand-alone mode, none of the PDN pads should be bridged. The picture below, as seen in the **GREEN** boxes, shows stand-alone mode with StealthChop.

Note: When the stall-guard function is **not used**, the stall-guard pin ("R") 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.



Stand-alone Mode - StealthChop

BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

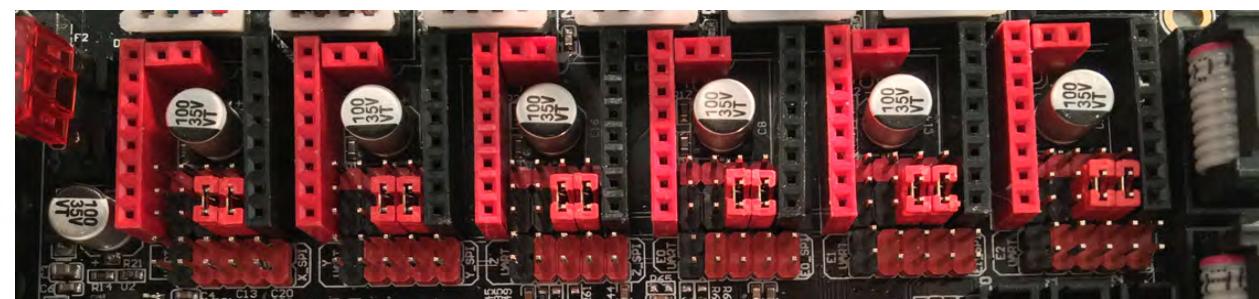
Stand-alone Mode

1 / 8

Interpolation: 1/256

StealthChop

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	-	-	B2
STEP	-	9	8	VDD
DIR	-	9	8	GND



Stand-alone Mode

1 / 32

Interpolation: 1/256

StealthChop

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	-	8	B2
STEP	-	-	9	VDD
DIR	-	9	-	GND



Stand-alone Mode - StealthChop

BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

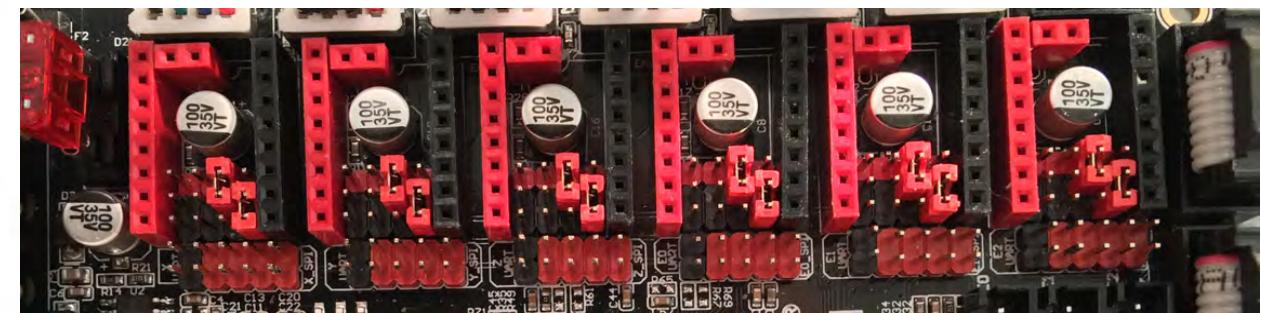
Stand-alone Mode

1 / 64

Interpolation: 1/256

StealthChop

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	9	-	B2
STEP	-	9	8	VDD
DIR	-	8	-	GND
	-		-	



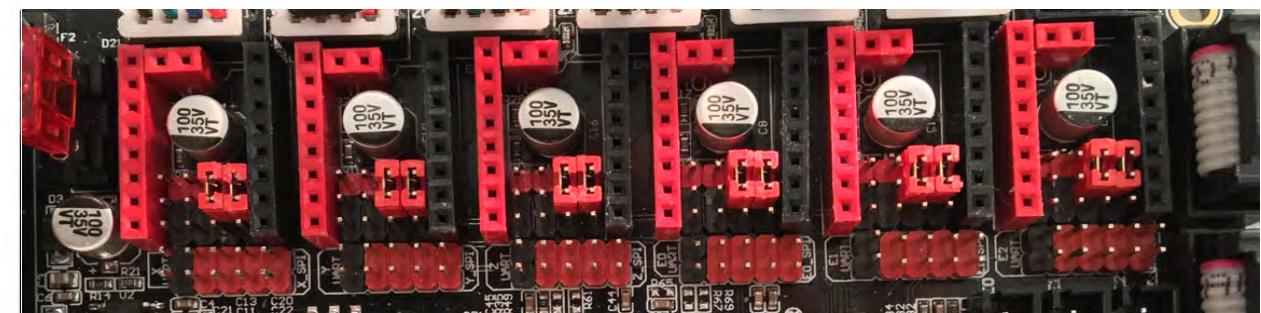
Stand-alone Mode

1 / 16

Interpolation: 1/256

StealthChop

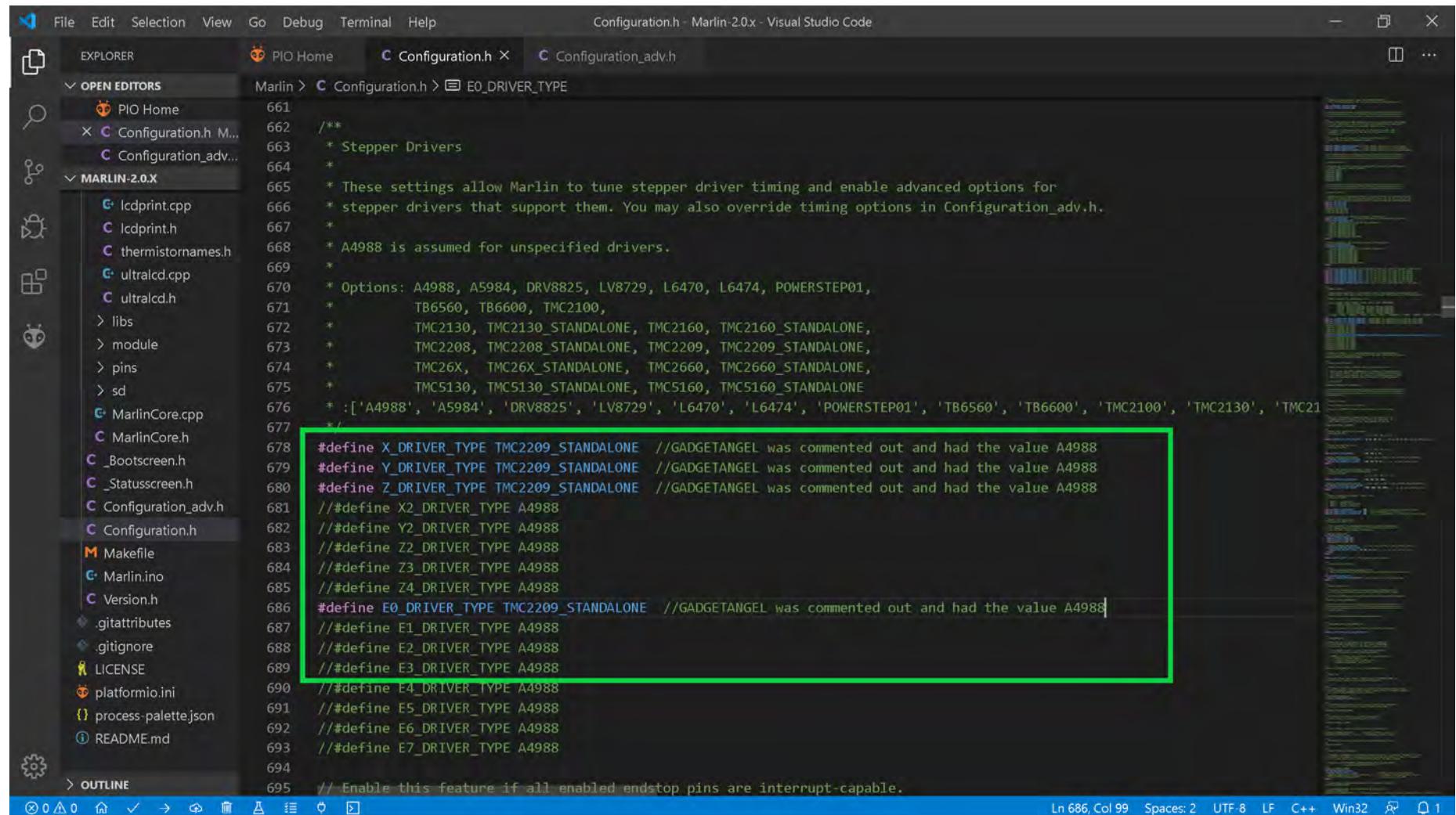
EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	9	8	B2
STEP	-	9	8	VDD
DIR	-	8	-	GND
	-		-	



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

NOTE: Go to Appendix C, and 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 ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file 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 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value 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 E1_DRIVER_TYPE A4988
686 //##define E2_DRIVER_TYPE A4988
687 //##define E3_DRIVER_TYPE A4988
688 //##define E4_DRIVER_TYPE A4988
689 //##define E5_DRIVER_TYPE A4988
690 //##define E6_DRIVER_TYPE A4988
691 //##define E7_DRIVER_TYPE A4988
692 //##define E8_DRIVER_TYPE A4988
693 //##define E9_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

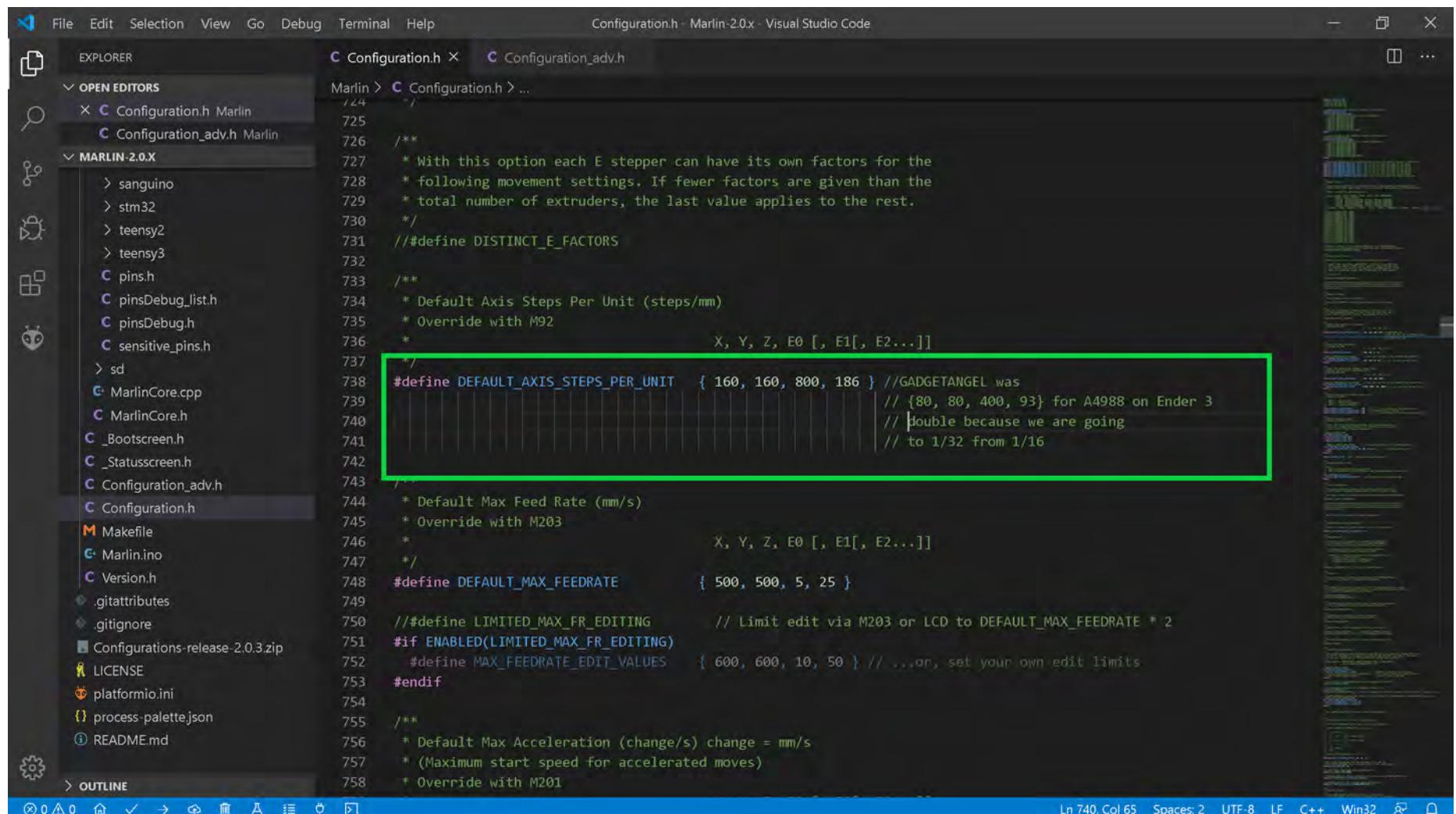
```

A green rectangular box highlights the driver type definitions for X, Y, Z, and E axes, showing that GADGETANGEL's values (A4988) were commented out and replaced with TMC2209_STANDALONE.

- Go to the next page.

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

- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 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.



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

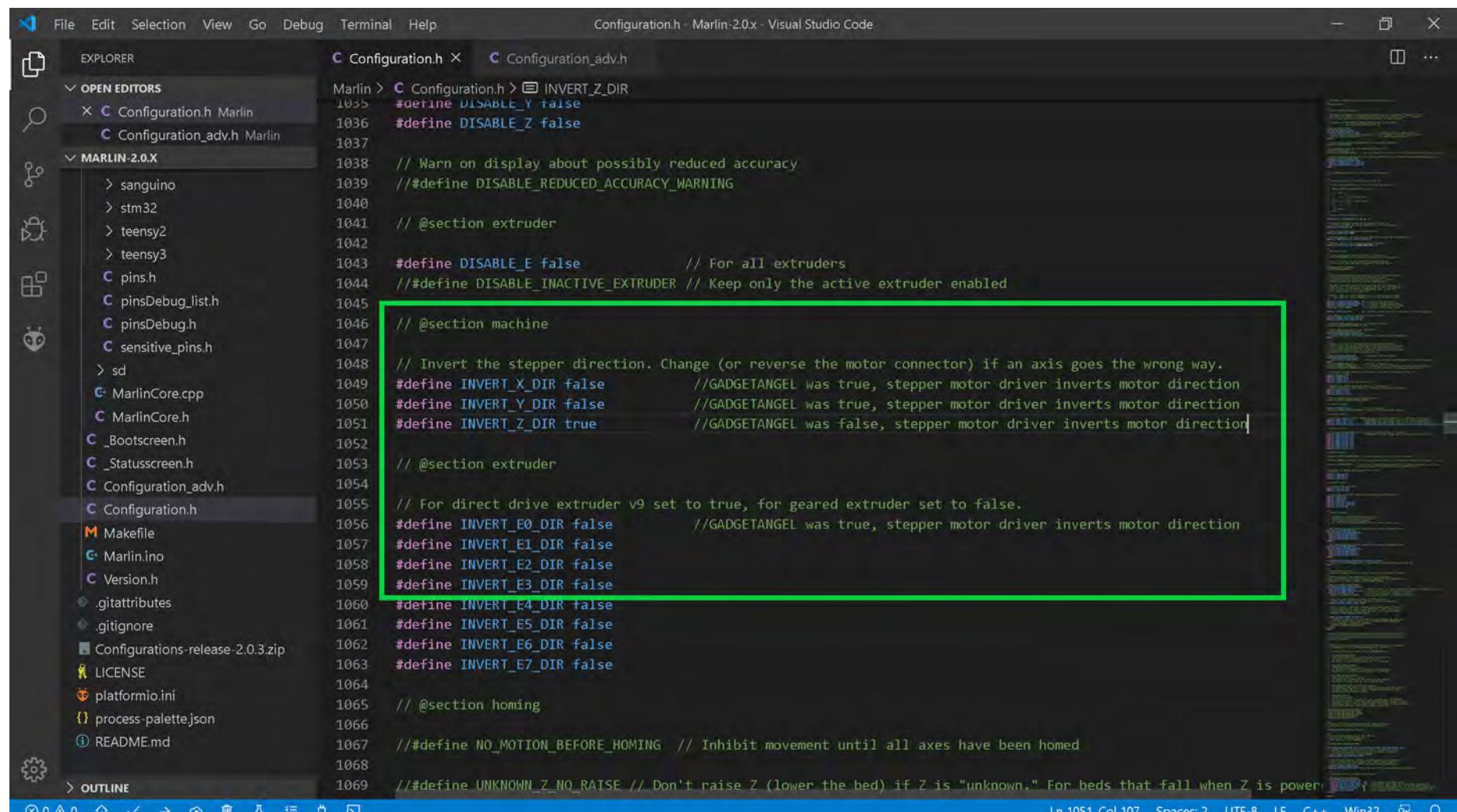
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current line (Ln 740), column (Col 65), and other settings like spaces (Spaces: 2), encoding (UTF-8), line endings (LF), and file type (C++). The left sidebar shows the project structure with various Marlin source files and configuration files.

- 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



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a green rectangular highlight box around the following lines of code:

```

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        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
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       // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

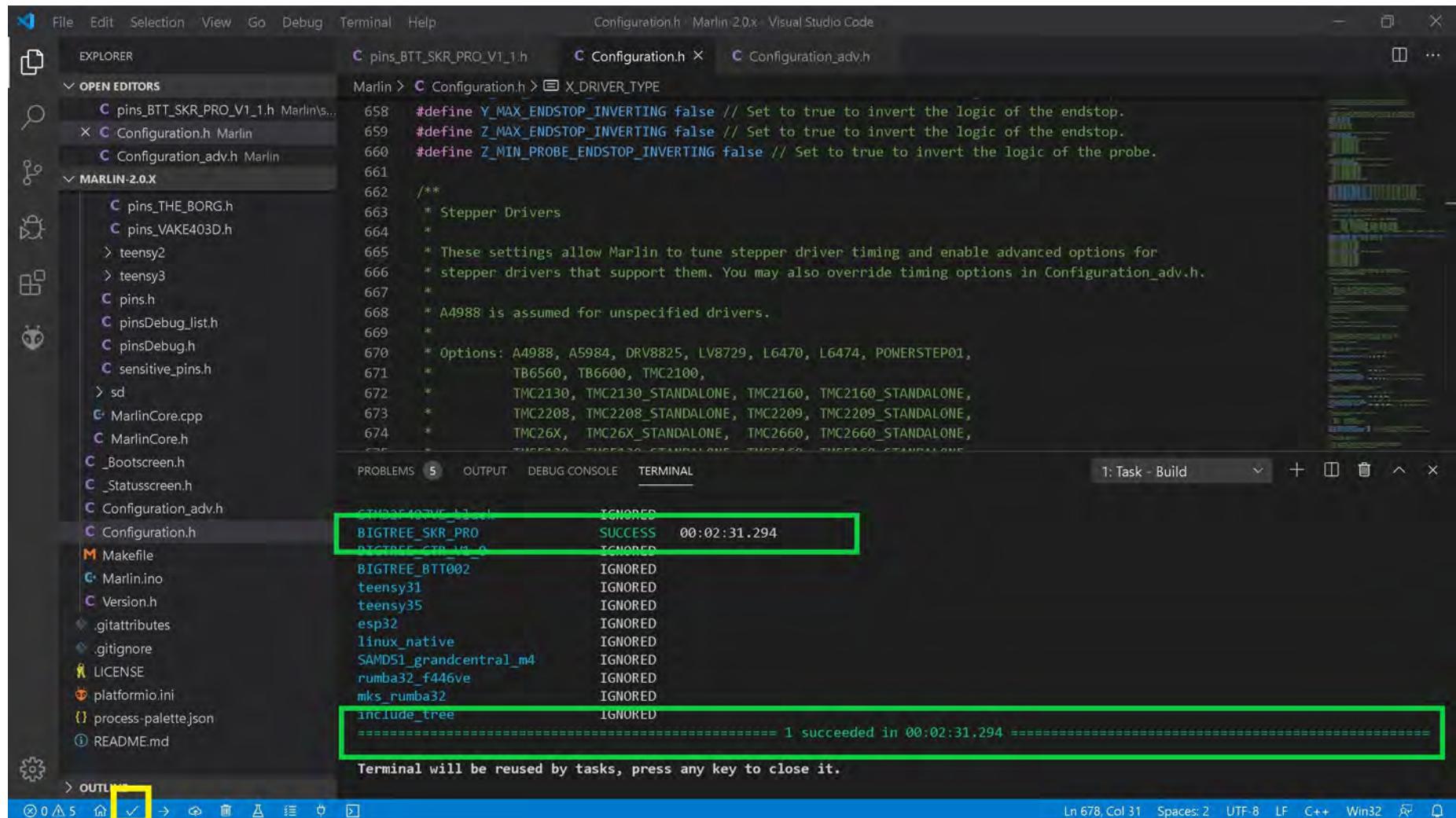
```

The status bar at the bottom right of the code editor shows: Ln 1051, Col 107 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 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.



```

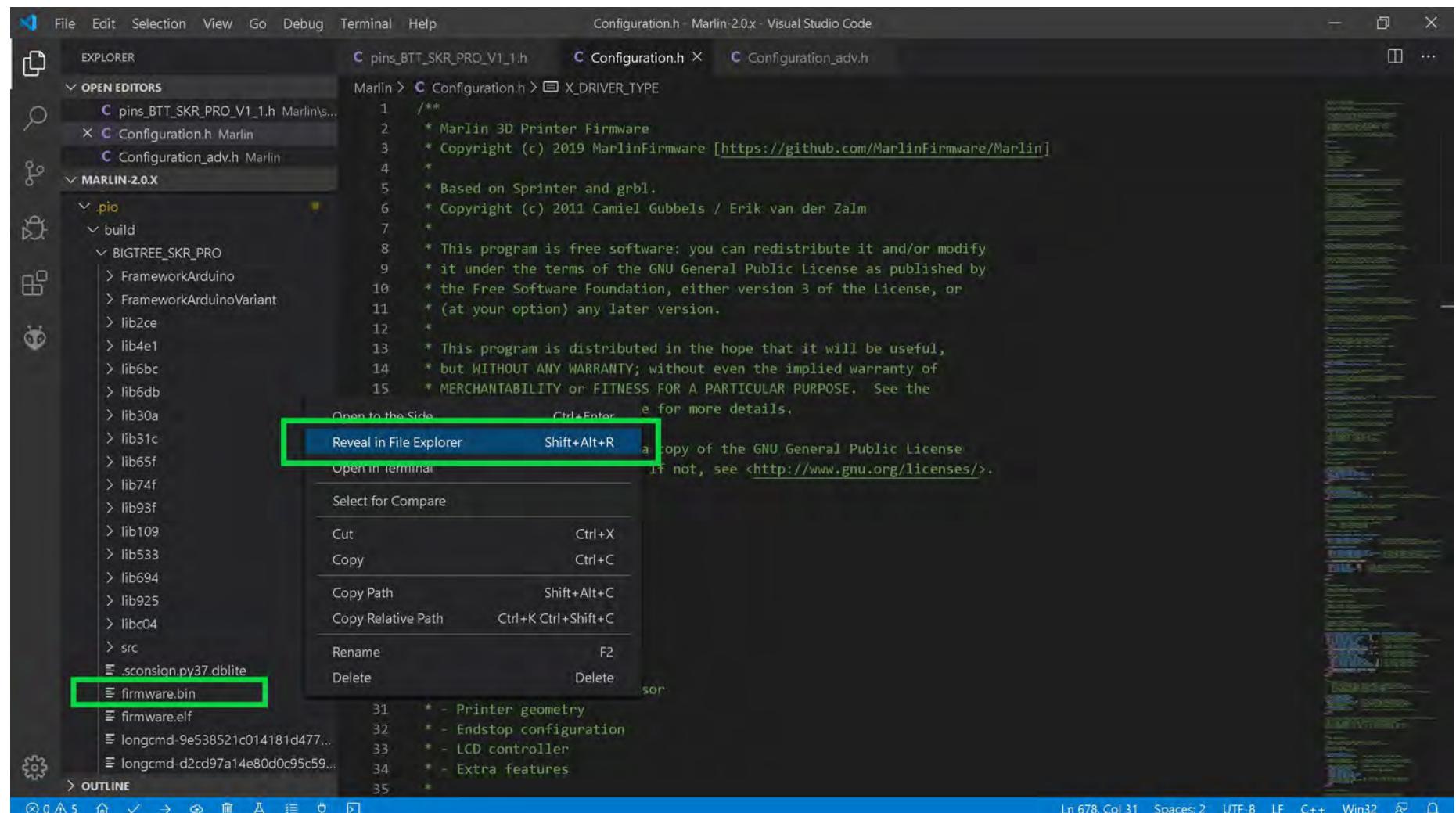
Configuration.h Marlin 2.0.x Visual Studio Code
pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h
Marlin > Configuration.h > X_DRIVER_TYPE
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true 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 * 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,
TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + □ ×
STM256107VE_1.1.0 IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_1.0.0 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMDS1_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 TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

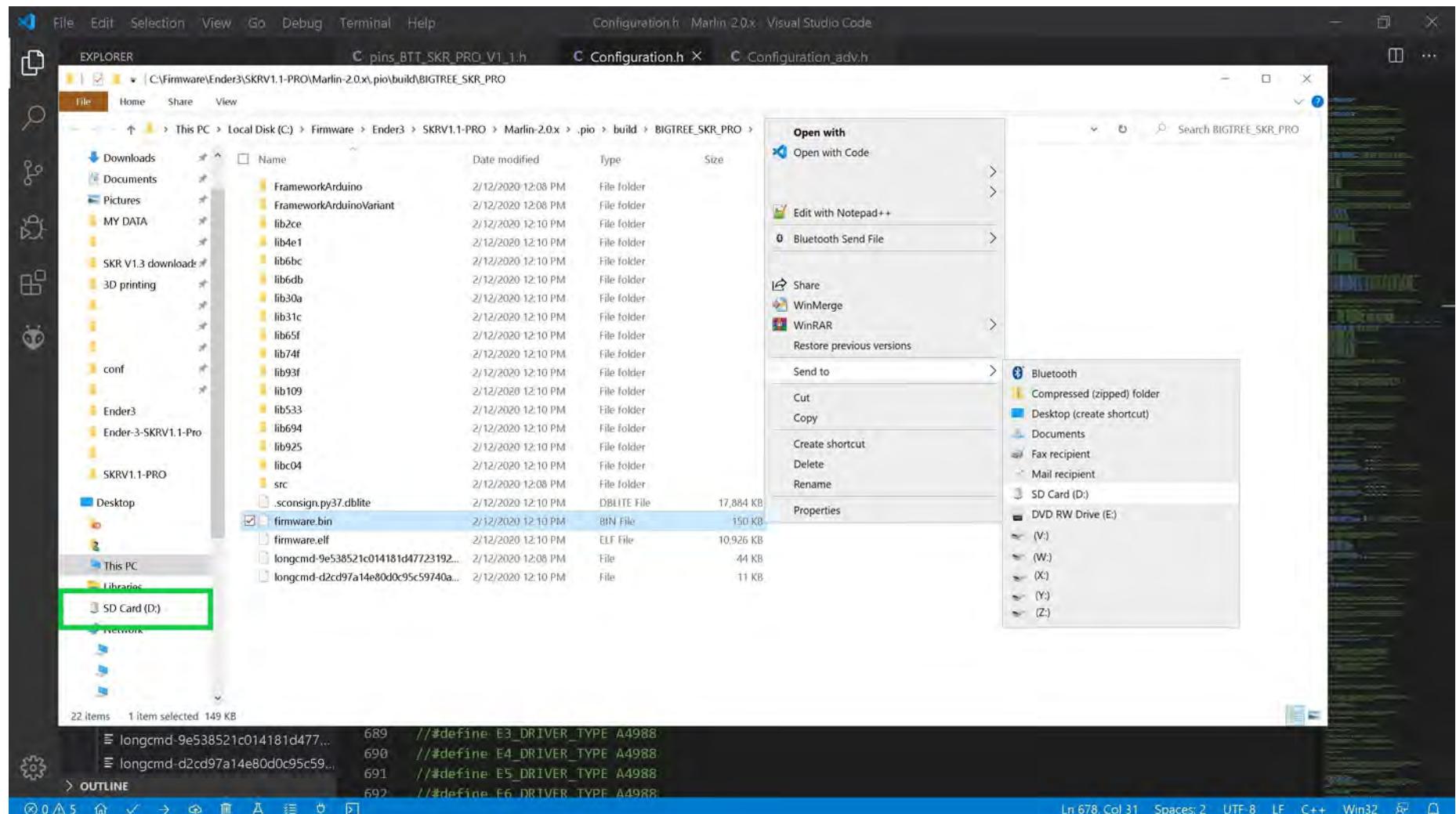
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer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



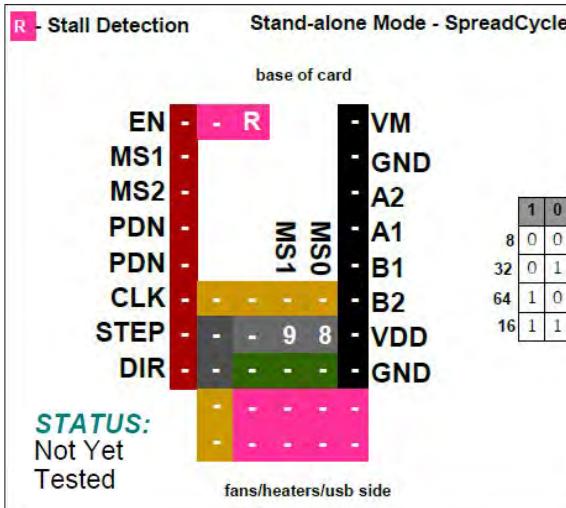
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



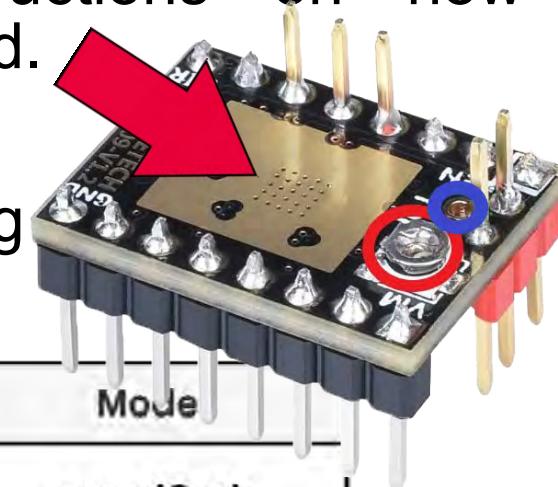
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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



BIQU TMC2209 V1.2

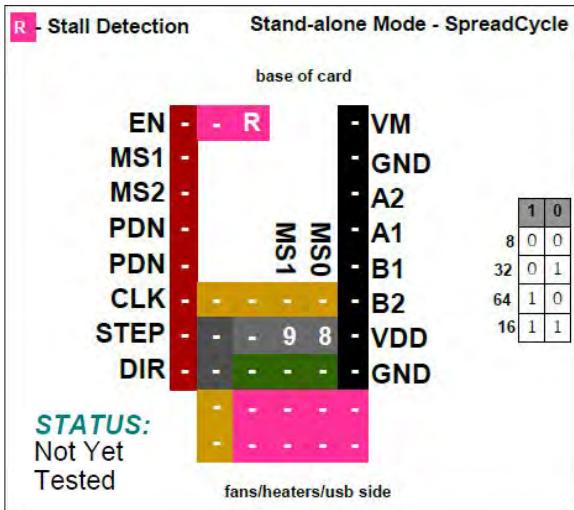
Stand-alone Mode for SpreadCycle

NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2209 <small>Stand Alone Mode Maximum 64 Subdivision 28V DC 2.8A (peak)</small>	GND	GND	1 / 8	1 / 256	spreadCycle
	GND	VIO	1 / 32	1 / 256	spreadCycle
	VIO	GND	1 / 64	1 / 256	spreadCycle
	VIO	VIO	1 / 16	1 / 256	spreadCycle
Driving Current Calculation Formula <small>R_S(Typical Sense Resistor)= 0.11Ω</small>	$I_{MAX} = V_{ref}$ See Appendix B #5. Use 50% to 90% as shown below:			$V_{ref} = I_{MAX}$ See Appendix B #5. Use 50% to 90% as shown below:	
	$I_{MAX} = I_{MAX} * 0.90$			$V_{ref} = V_{ref} * 0.90$	



BIQU TMC2209 V1.2

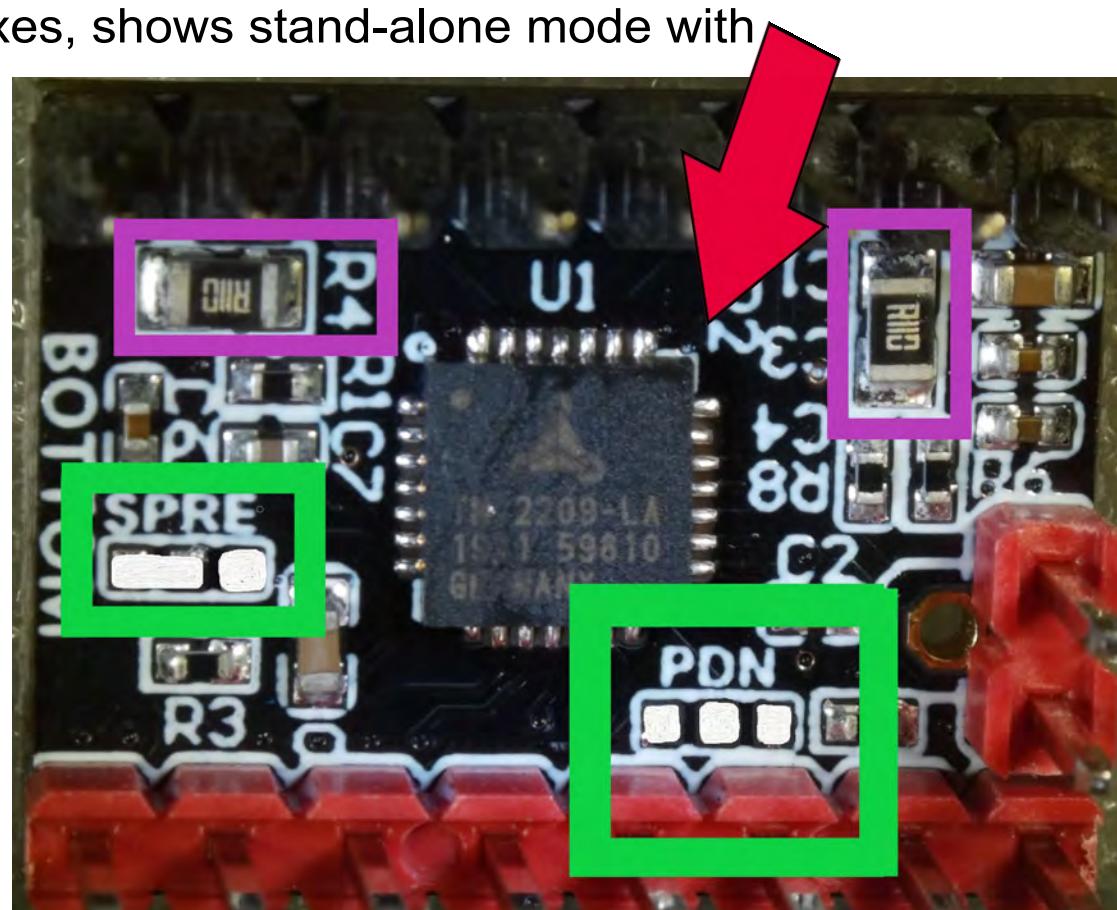
Stand-alone Mode for SpreadCycle

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. The **PURPLE boxes** below show the location of the current sense resistors (R_s) for the TMC2209.

Note: To switch to stand-alone mode, none of the PDN pads should be bridged. The picture below, as seen in the **GREEN** boxes, shows stand-alone mode with SpreadCycle.

Note: When the stall-guard function is **not used**, the stall-guard pin ("R") 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.



Stand-alone Mode - SpreadCycle

BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

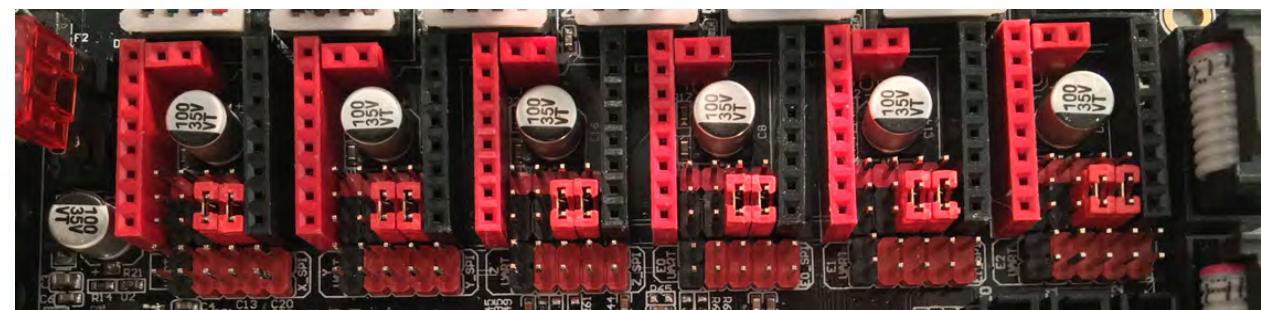
Stand-alone Mode

1 / 8

Interpolation: 1/256

SpreadCycle

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	-	-	B2
STEP	-	-	9 8	VDD
DIR	-	9	8	GND
	-			



Stand-alone Mode

1 / 32

Interpolation: 1/256

SpreadCycle

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	-	8	B2
STEP	-	-	9 8	VDD
DIR	-	9	-	GND
	-			



Stand-alone Mode - SpreadCycle

Stand-alone Mode

1 / 64

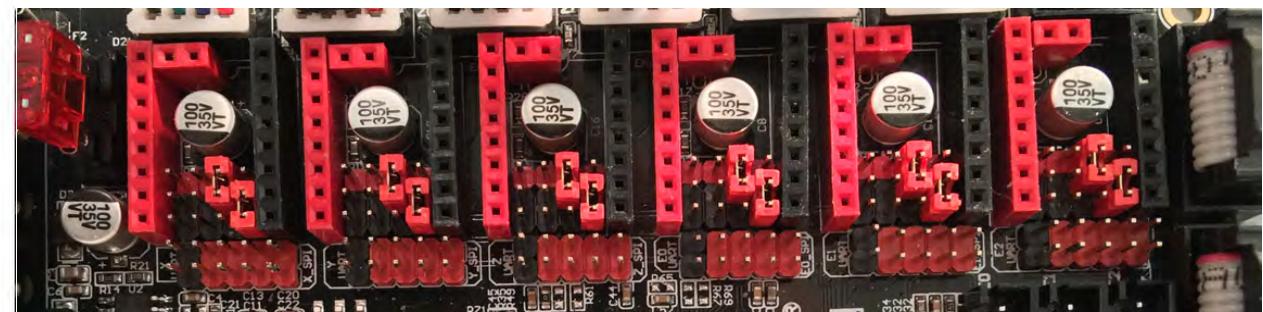
Interpolation: 1/256

SpreadCycle

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	9	-	B2
STEP	-	9	8	VDD
DIR	-	8	-	GND

BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle



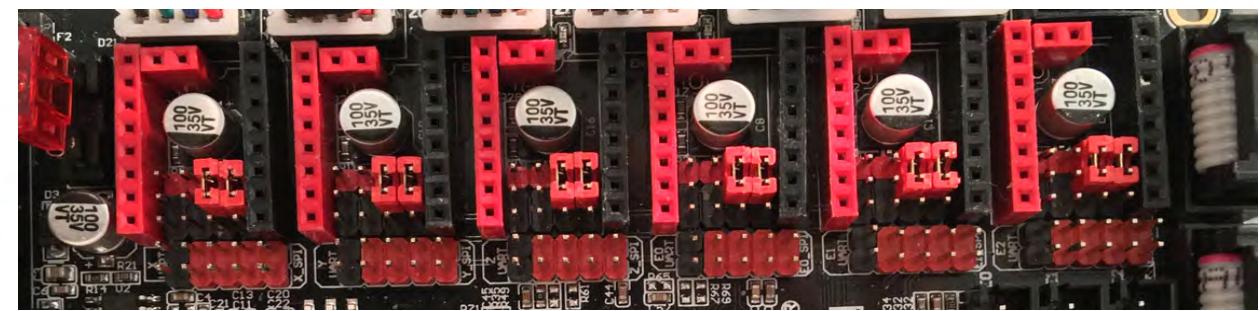
Stand-alone Mode

1 / 16

Interpolation: 1/256

SpreadCycle

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	MS0	B1
CLK	-	9	8	B2
STEP	-	9	8	VDD
DIR	-	8	-	GND



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

NOTE: [Go to Appendix C](#), and 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 ("//").

```

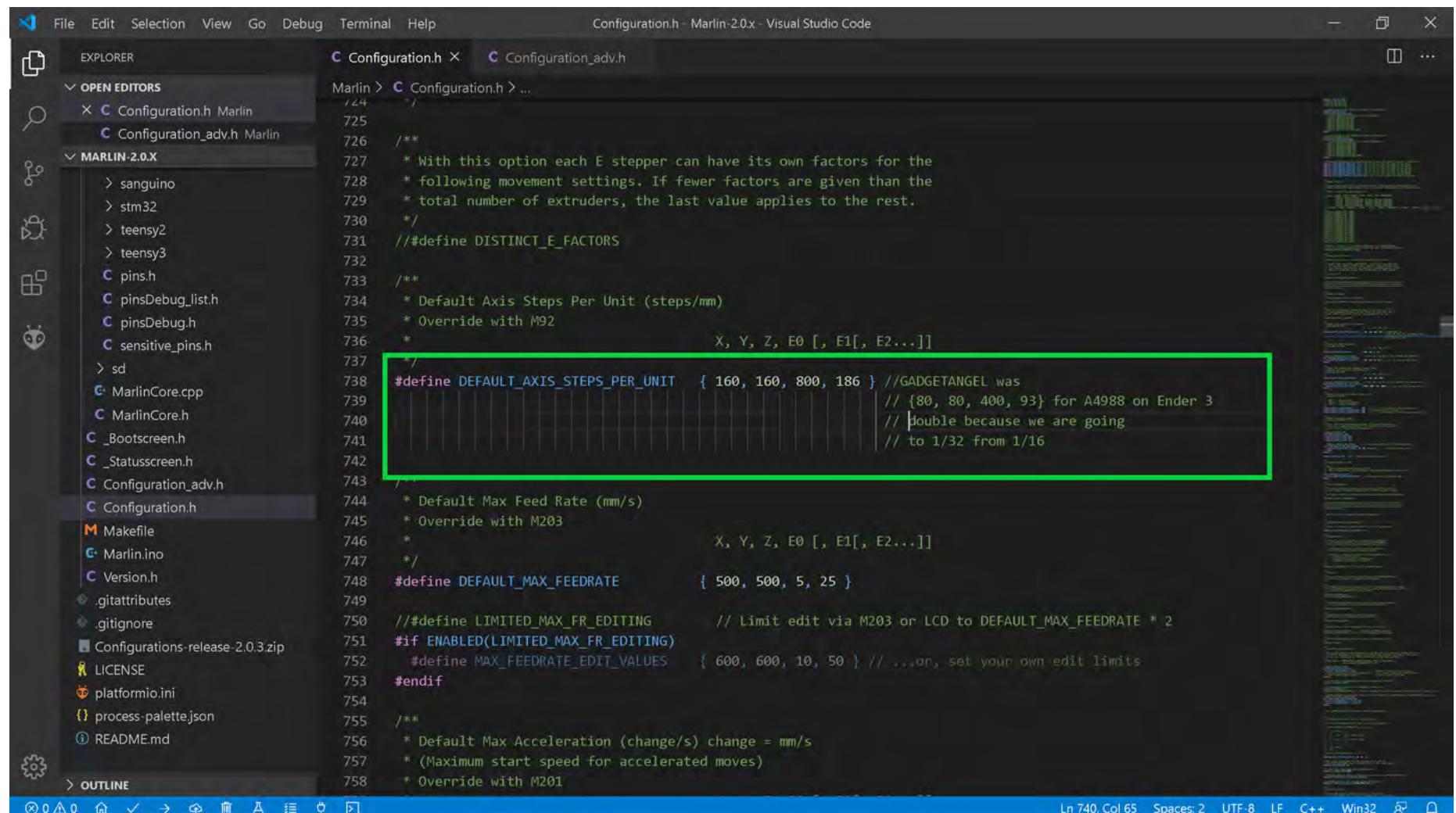
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M...
Configuration_adv.h
MARLIN-2.0.X
Lcdprint.cpp
Lcdprint.h
thermistornames.h
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
Ln 686, Col 99 Spaces: 2 UTF-8 LF C++ Win32 1
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', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value 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 TMC2209_STANDALONE //GADGETANGEL was commented out and had the value 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.

```

- Go to the next page.

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

- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 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.



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

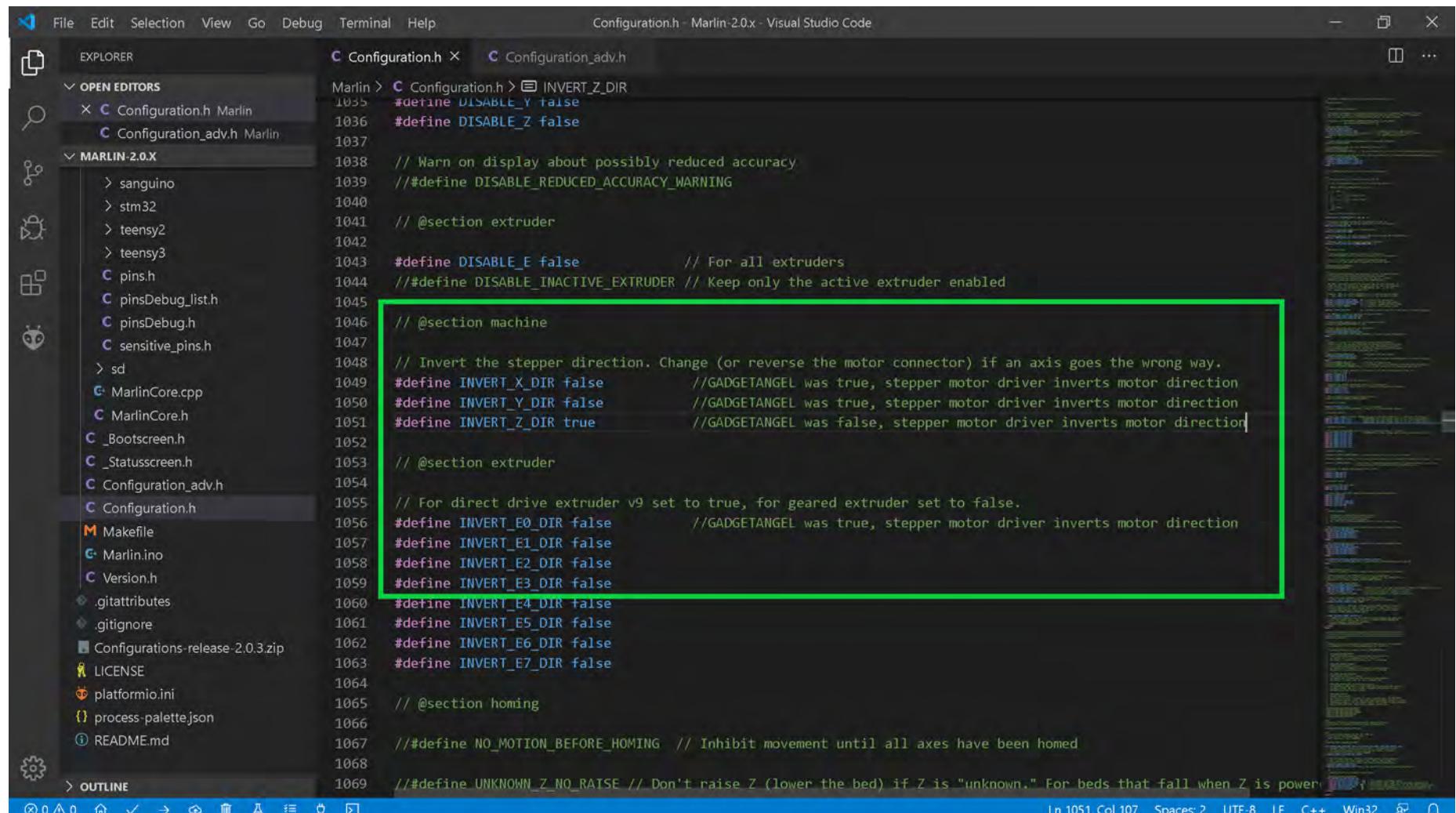
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom right indicates: Ln 740, Col 65, 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

- 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



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 Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  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      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  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     // GADGETANGEL was true, stepper motor driver inverts motor direction
  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
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

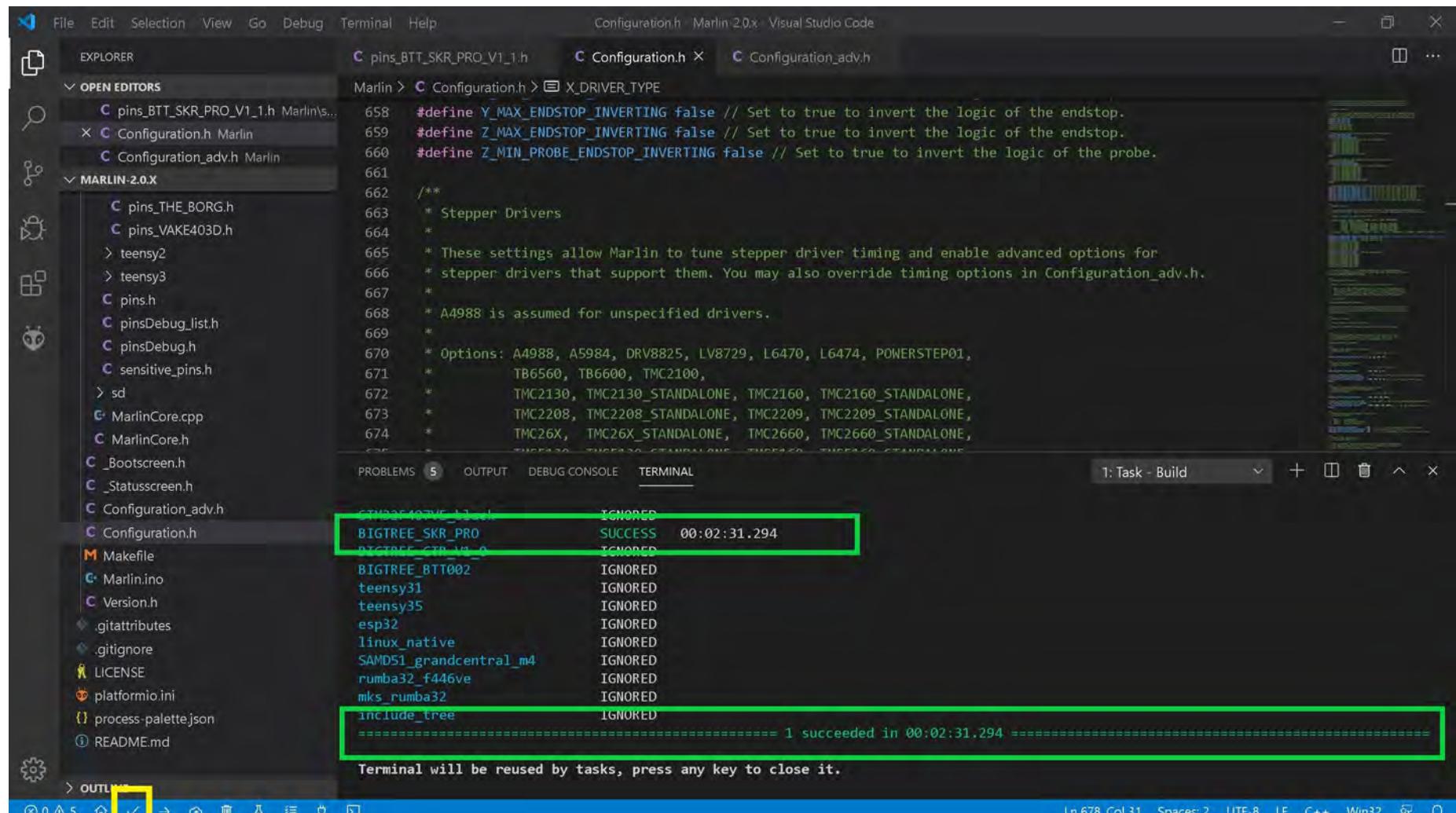
```

Ln 1051, Col 107 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 Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Editor Area:** Shows files like Configuration.h, pins_BTT_SKR_PRO_V1_1.h, and Configuration_adv.h.
- Terminal Tab:** Displays the build log for the Marlin firmware. 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 compilation results.
- Output Tab:** Shows the build progress with tasks like "BIGTREE_SKR_PRO" and "BIGTREE_BTT002" marked as "SUCCESS".
- Bottom Status Bar:** Shows the current line (Ln 678), column (Col 31), spaces (Spaces: 2), and encoding (UTF-8).

```

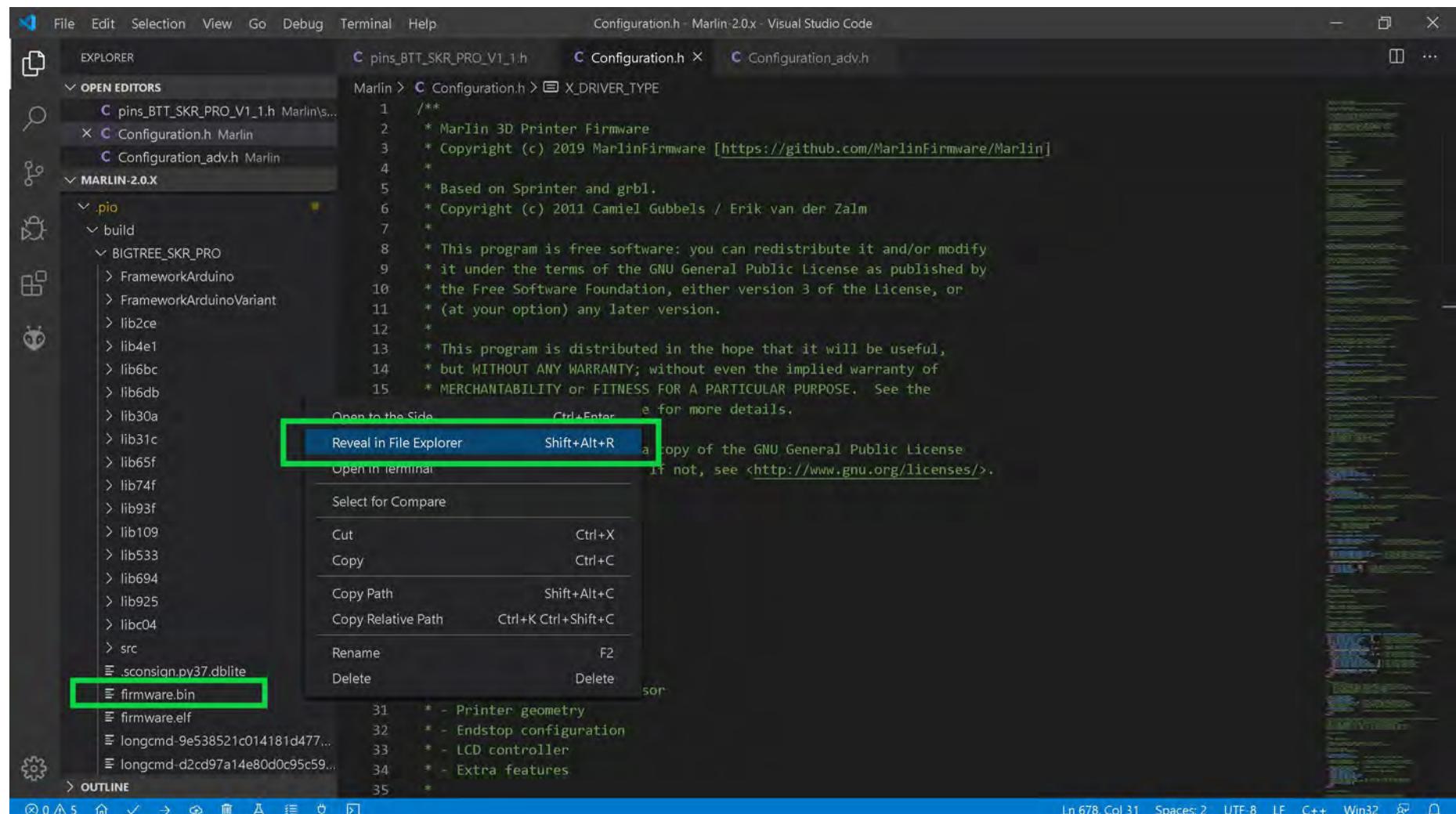
Configuration.h Marlin 2.0.x Visual Studio Code
pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
Marlin > Configuration.h > X_DRIVER_TYPE
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to 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 * 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, PONERSTEP01,
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 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ×
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
=====
===== 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 TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle

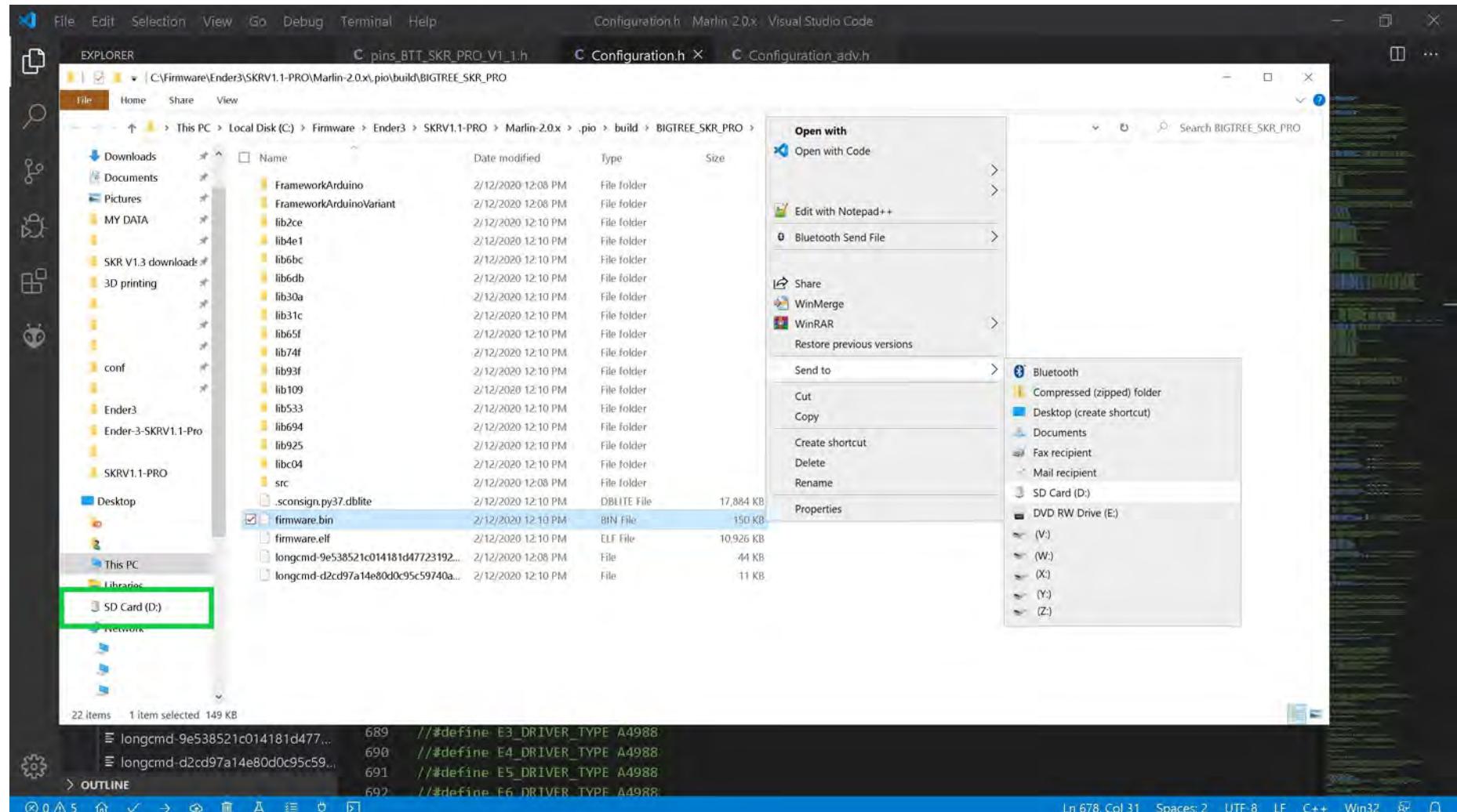
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer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



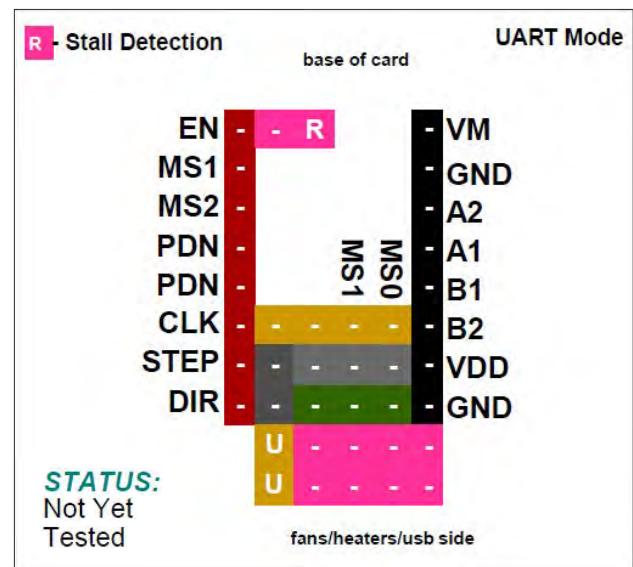
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".

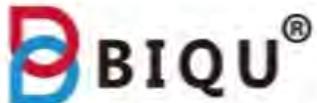


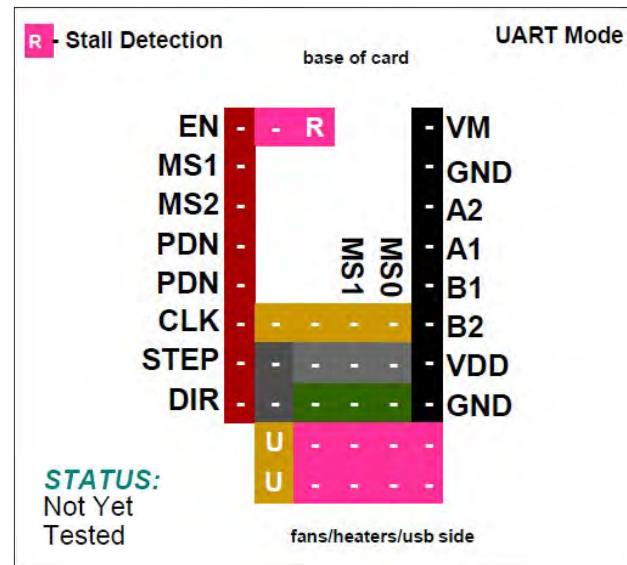
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

**BIQU TMC2209 V1.2****UART Mode**

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

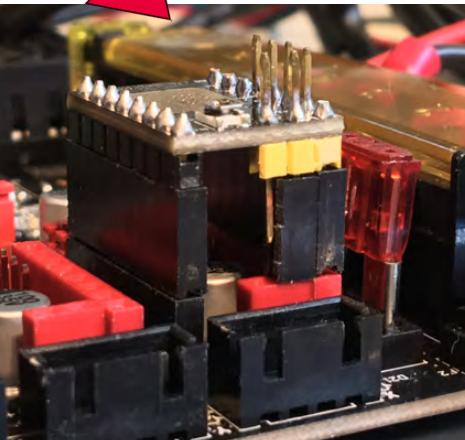
See the next page for further information.

Driver Chip  TMC2209 UART Mode Maximum 256 Subdivision 28V DC 2.8A (peak)	Steps are set inside of your Firmware	
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.11Ω	$I_{MAX} = V_{ref}$ See Appendix B #6. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	$V_{ref} = I_{MAX}$ See Appendix B #6. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$



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.

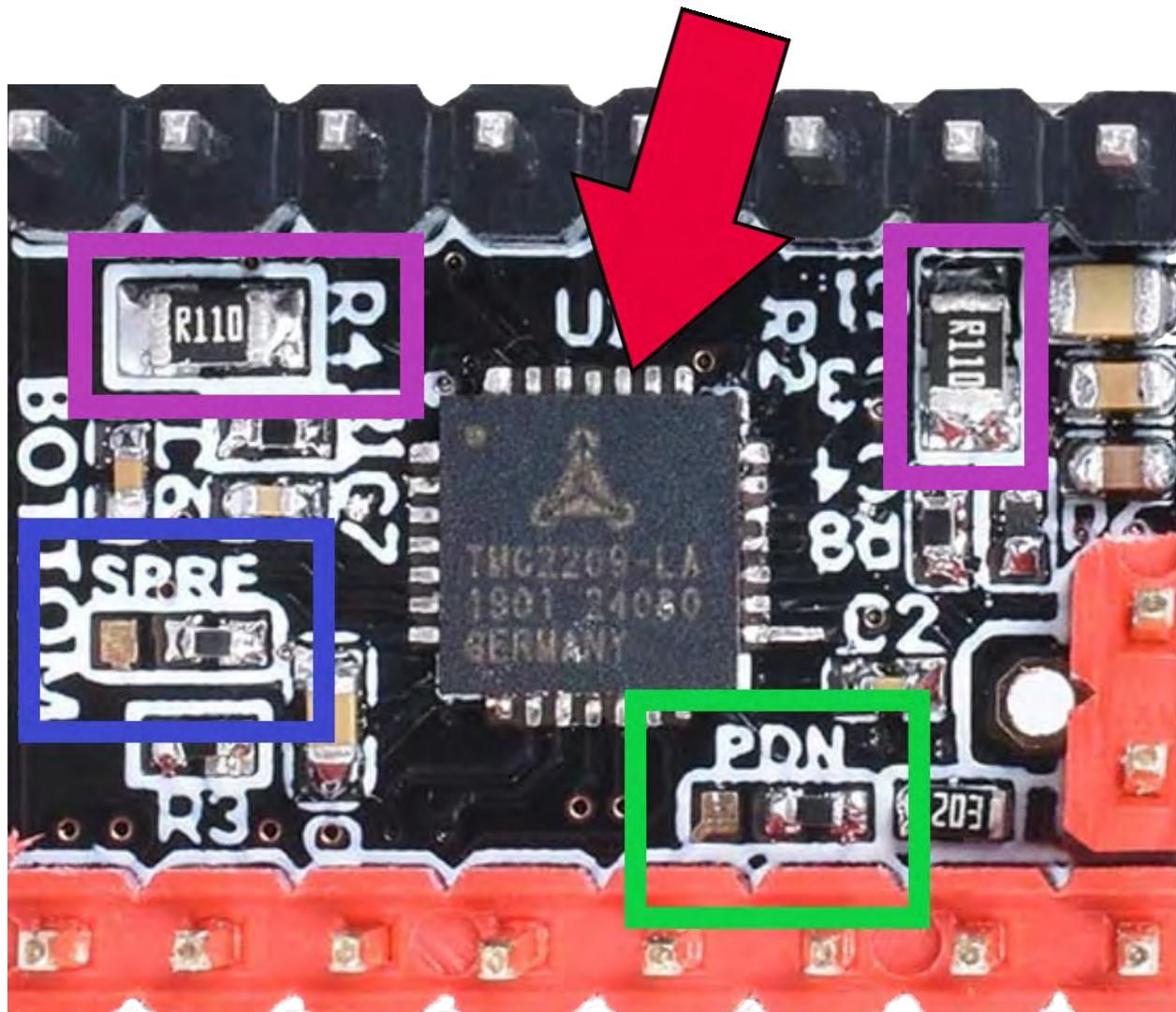
Note: When the stall-guard function is **not used**, the stall-guard pin ("R") of the TMC2209 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board



The **PURPLE** boxes show the location of the current sense resistors (R_s).

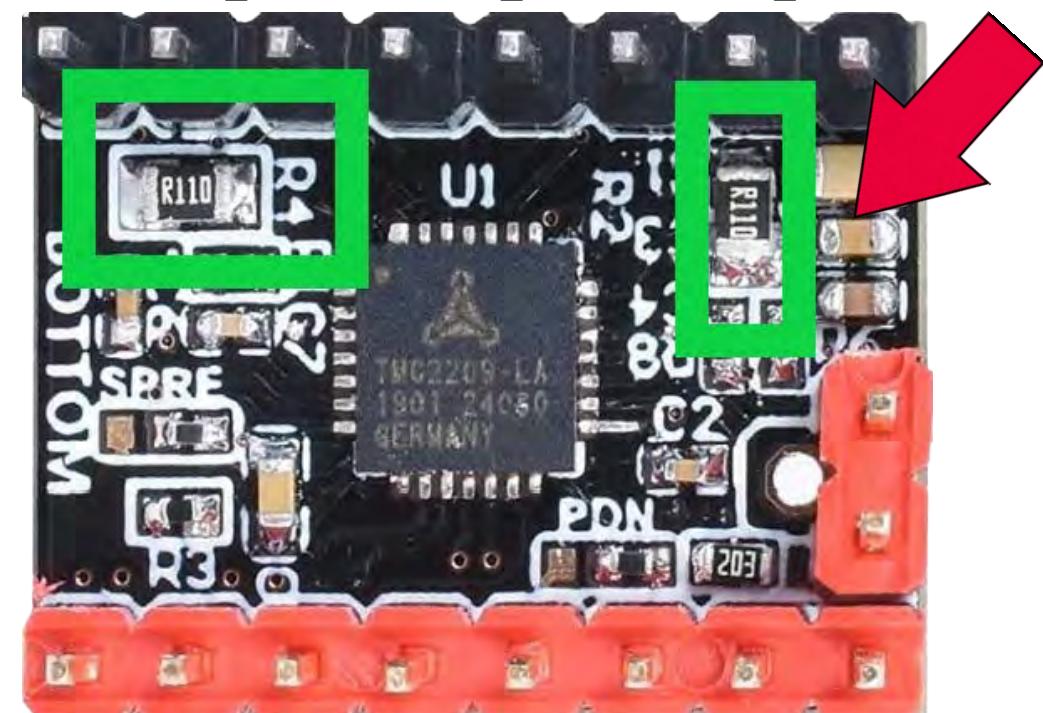
Note: To ensure your TMC2209 is in UART mode, look on the bottom of the driver board for the PDN pads. Two of the three pads should be bridged together. **If a bridge exists then the device is in UART Mode, as seen in the GREEN box.**

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



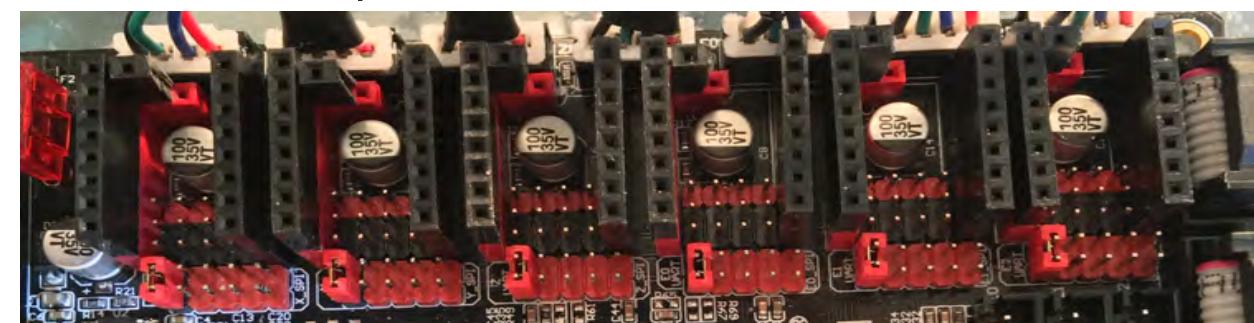
UART Mode**R - Stall Detection**

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

 $R_s = R050$ is 0.05 Ohms **$R_s = R062$ is 0.062 Ohms** **$R_s = R068$ is 0.068 Ohms** **$R_s = R075$ is 0.075 Ohms** **$R_s = R100$ is 0.1 Ohms** **$R_s = R110$ is 0.11 Ohms** **$R_s = R150$ is 0.15 Ohms** **$R_s = R200$ is 0.2 Ohms** **$R_s = R220$ is 0.22 Ohms****UART**

EN	-	R	-	VM
MS1	-		-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	-	B1
CLK	-		-	B2
STEP	-		-	VDD
DIR	-		-	GND
	U	-	-	
	U	-	-	

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



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

NOTE: Go to Appendix C, and 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 * 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', 'TMC26X', 'TMC5130']
676 */
677
678 #define X_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value 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 //GADGETANGEL was commented out and had the value 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.

```

The lines from 678 to 693 are highlighted with a green box, indicating they were previously commented out. The code editor status bar at the bottom shows: Ln 686, Col 88, 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

- 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 Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
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      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
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     // GADGETANGEL was true, stepper motor driver inverts motor direction
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
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

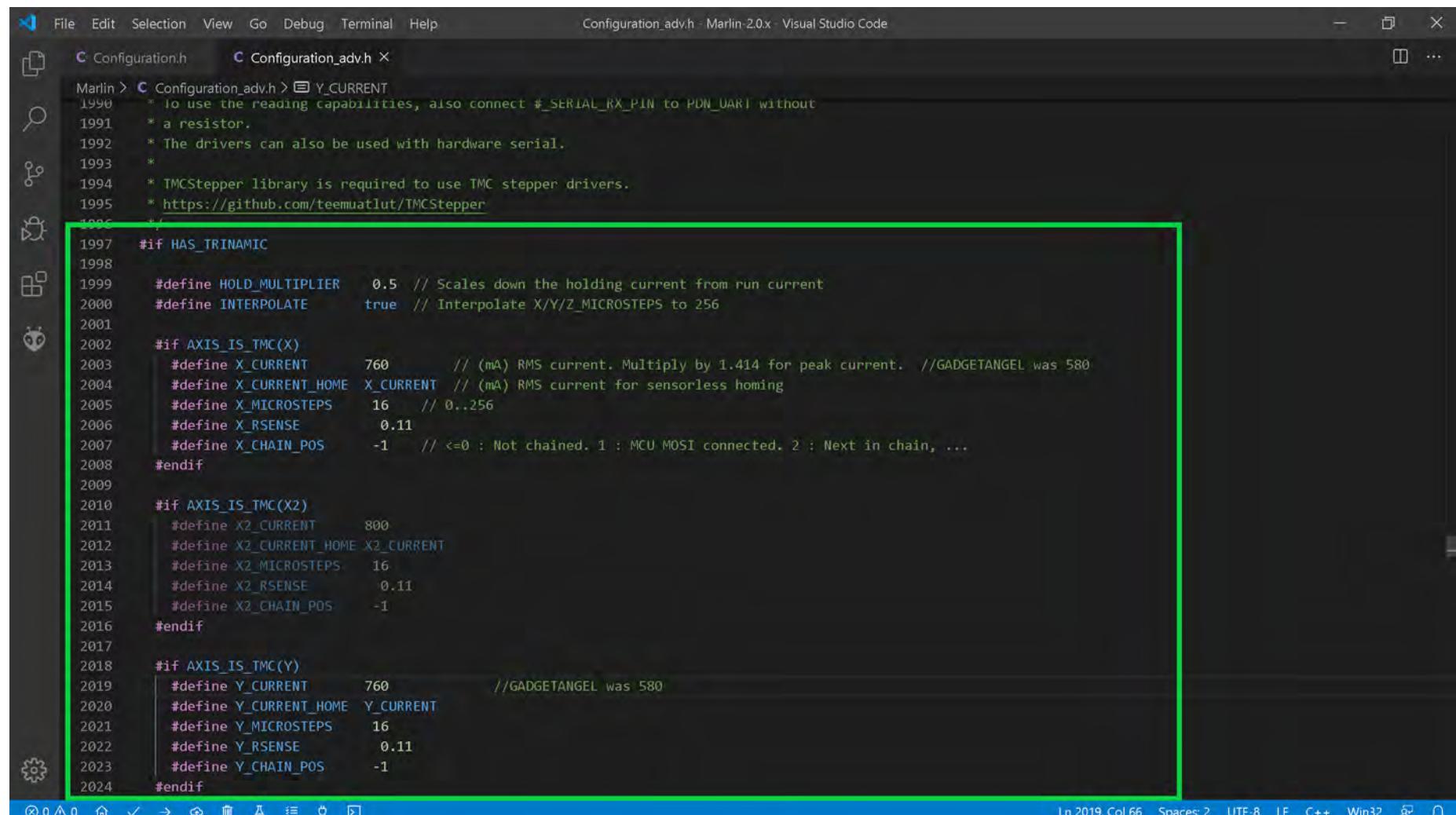
```

Ln 1051, Col 107 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 V_{ref} in the Marlin firmware for each axis that has the TMC2209 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
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. //GADGETANGEL was 580
2004     #define X_CURRENT_HOME   X_CURRENT // (mA) RMS current for sensorless homing
2005     #define X_MICROSTEPS     16       // 0..256
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      //GADGETANGEL was 580
2020     #define Y_CURRENT_HOME   Y_CURRENT
2021     #define Y_MICROSTEPS     16
2022     #define Y_RSENSE          0.11
2023     #define Y_CHAIN_POS       -1
2024 #endif

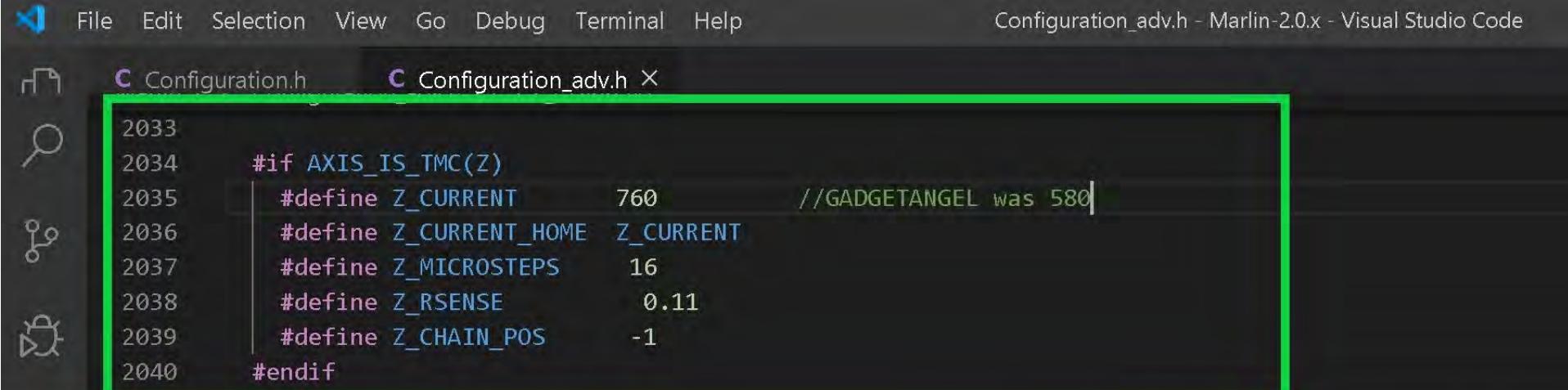
```

Ln 199, Col 66 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

- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.11. Ensure "E0_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".



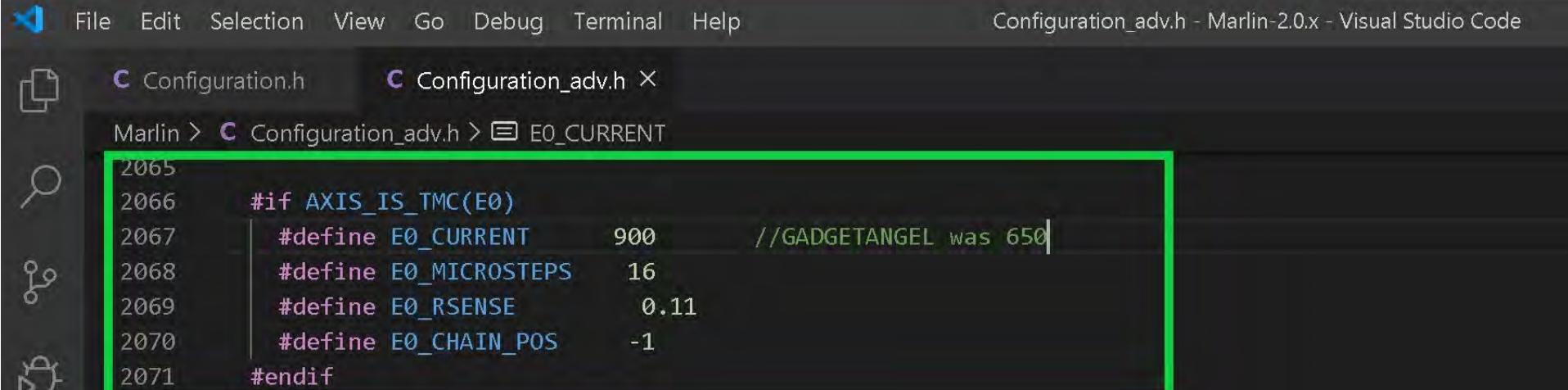
```

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

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760           //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME Z_CURRENT
2037   #define Z_MICROSTEPS    16
2038   #define Z_RSENSE        0.11
2039   #define Z_CHAIN_POS     -1
2040 #endif

```



```

File Edit Selection View Go Debug Terminal Help
Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > E0_CURRENT

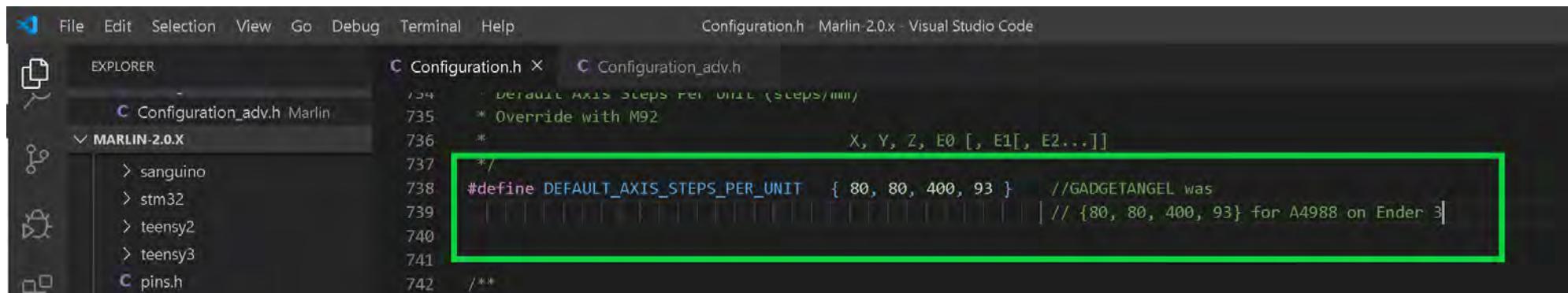
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900           //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- Go to the next page.

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

- 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 Configuration.h X Configuration_adv.h

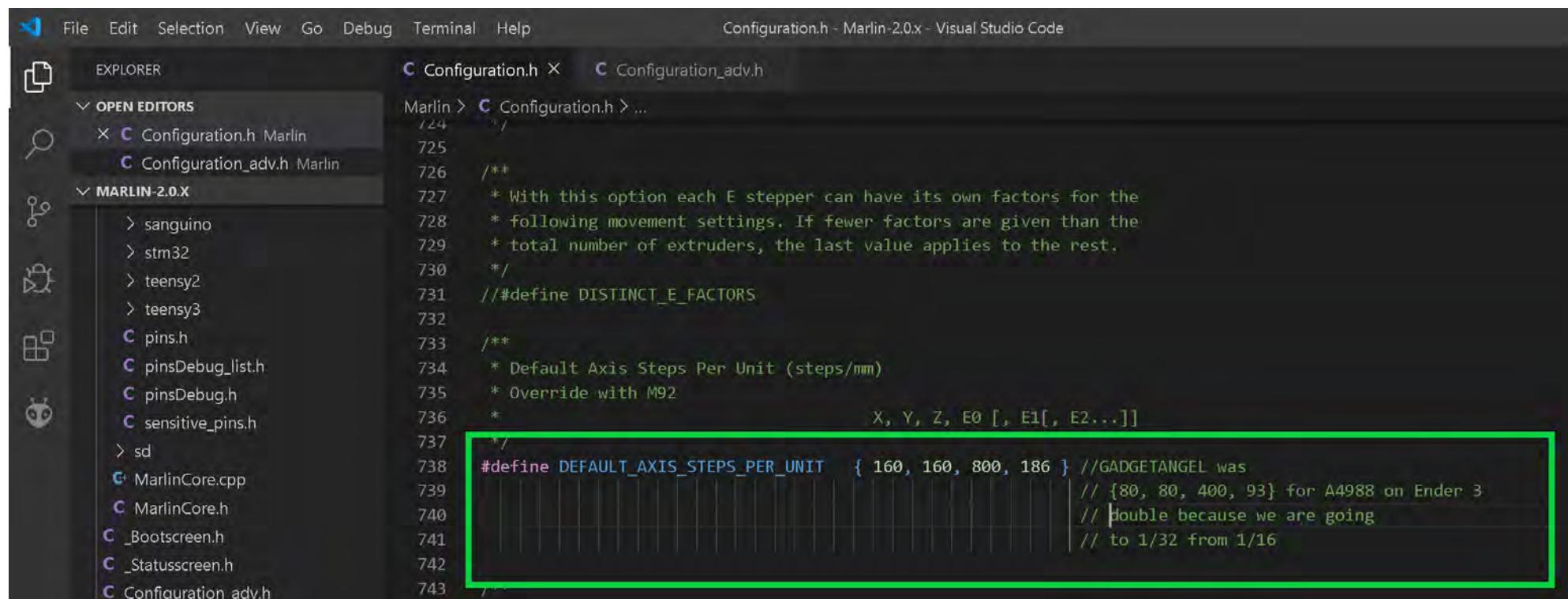
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

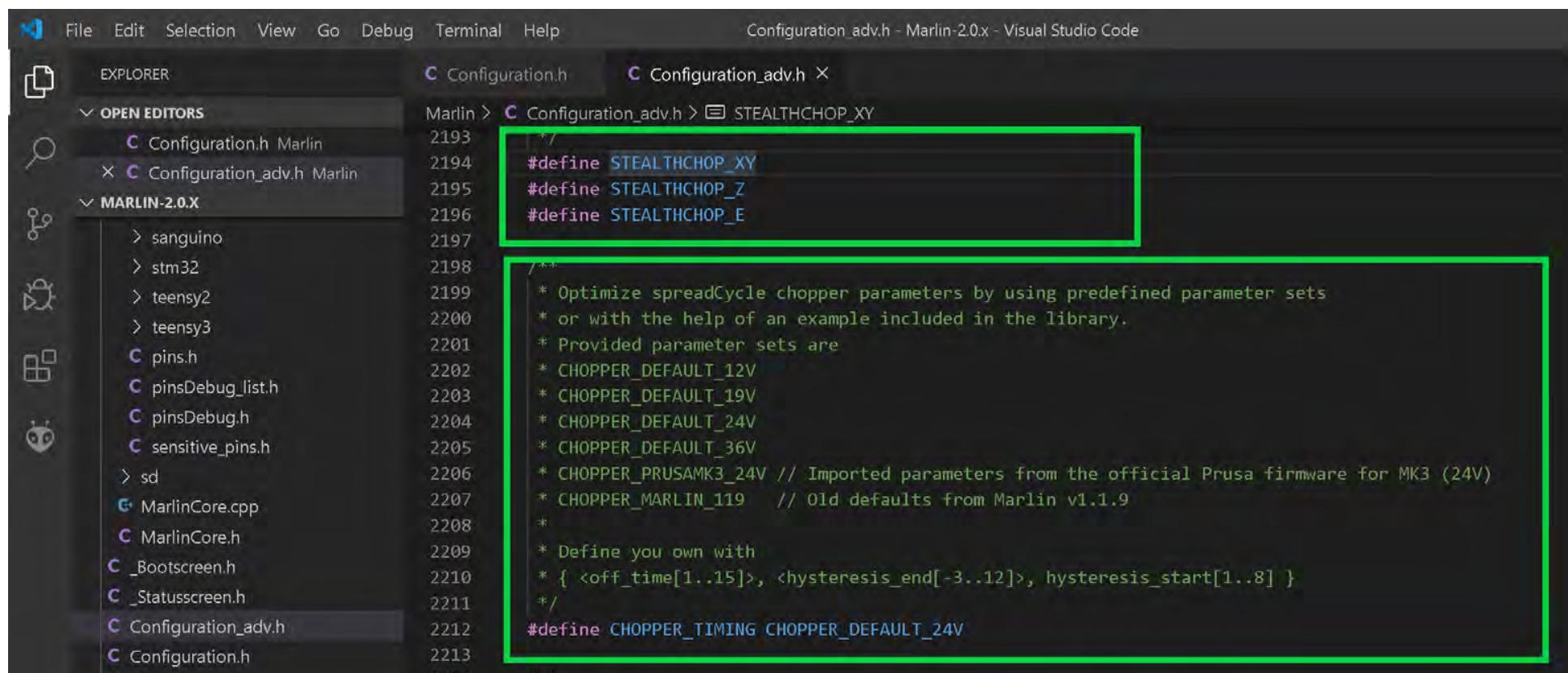
724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */

```

- 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 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

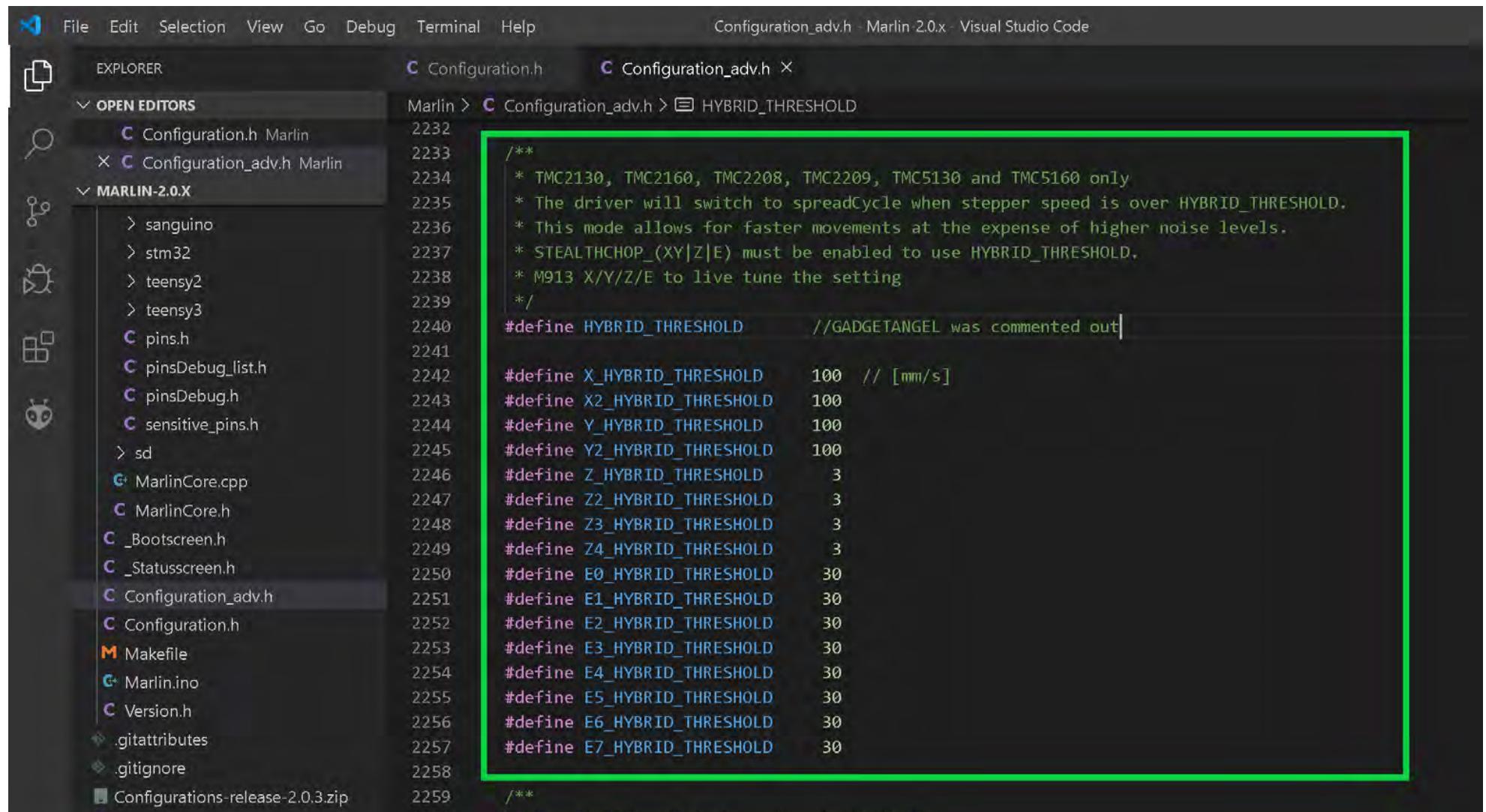
EXPLORER Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > STEALTHCHOP_XY
2193 /**
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help 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 /**

```

- 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 following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
- Explorer:** Shows the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X' (including files like Configuration.h, Configuration_adv.h, pins.h, etc.)
- Editor:** Displays the content of Configuration_adv.h with a green box highlighting the HYBRID_THRESHOLD section.

```

/*
 * 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 //GADGETANGEL was commented out

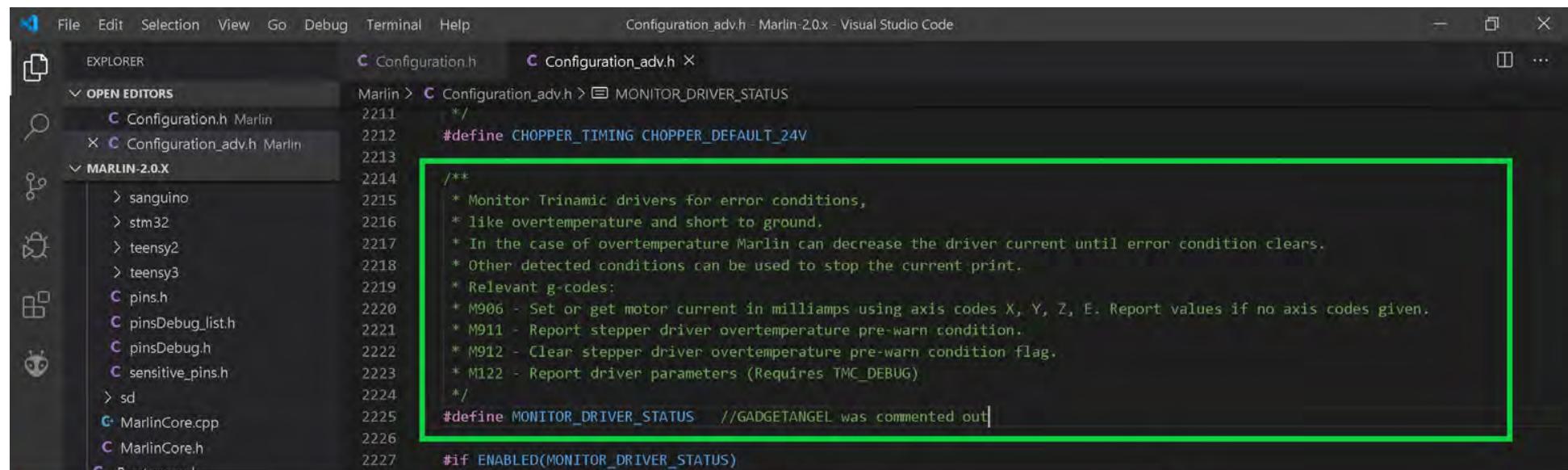
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 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

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART 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.](#)



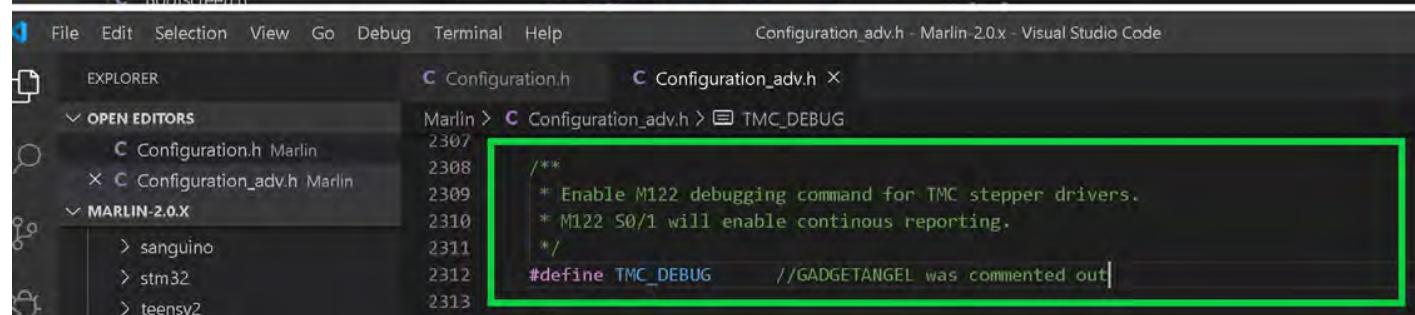
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Marlin > Configuration.h Configuration_adv.h X
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 //GADGETANGEL 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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

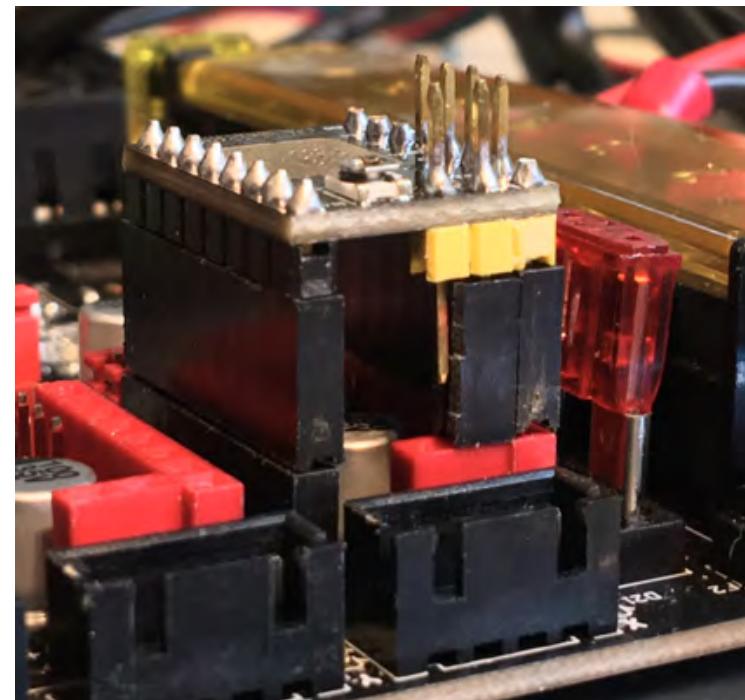
Marlin > Configuration.h Configuration_adv.h X
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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.



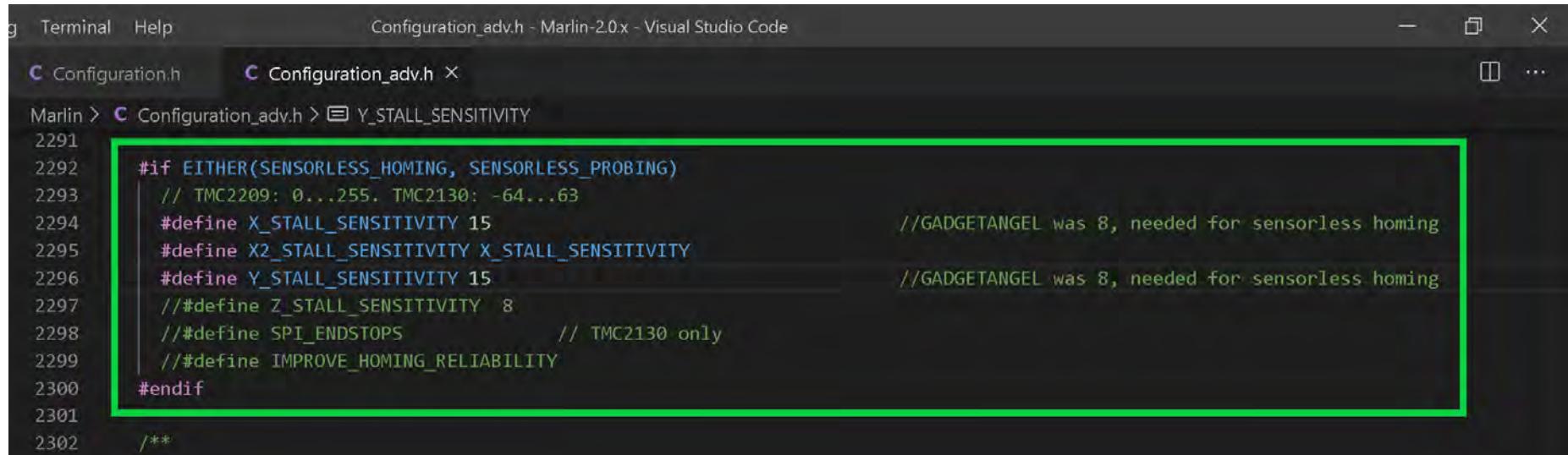
- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code  
EXPLORER Configuration.h Configuration_adv.h X  
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING  
2281 */  
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out  
2283  
MARLIN-2.0.X
```

- Go to the next page.

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

- Next we set the "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".

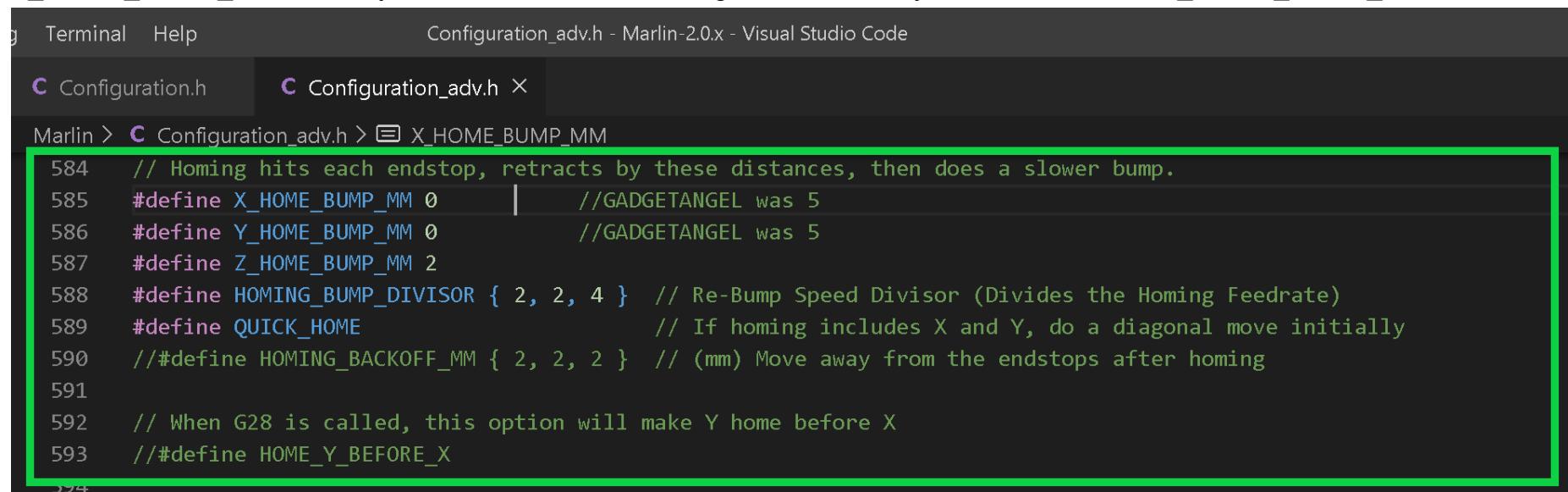


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- 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.



```

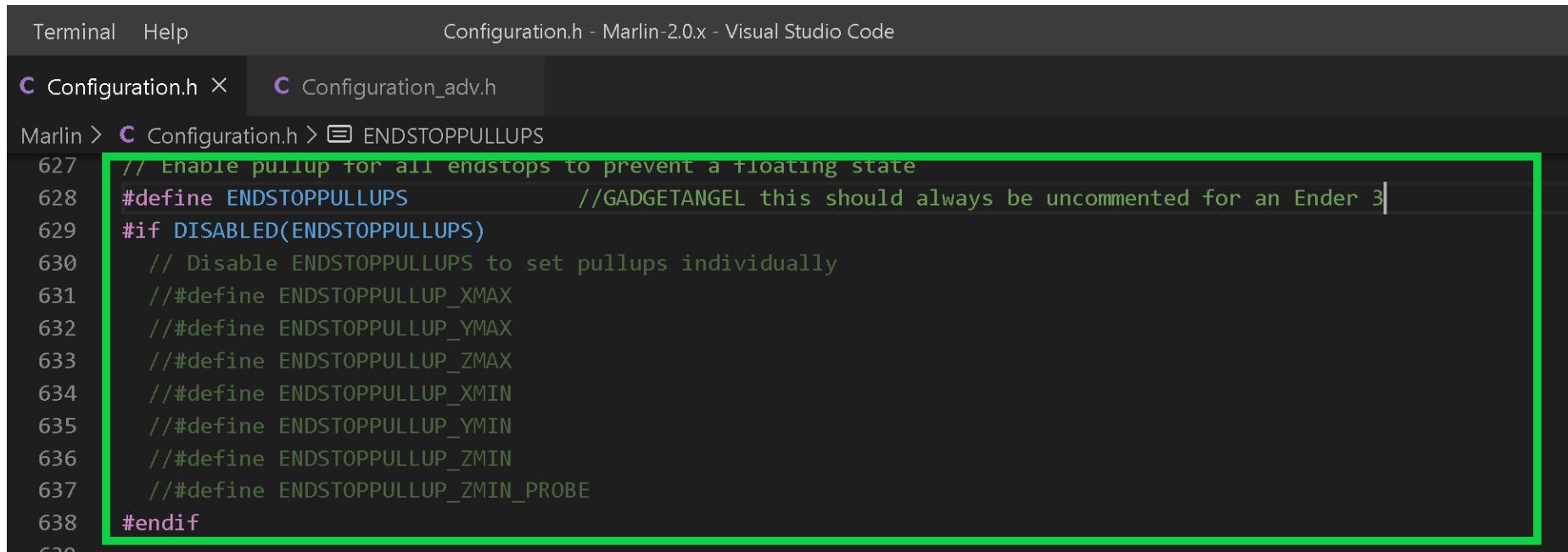
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL 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

```

- 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.



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

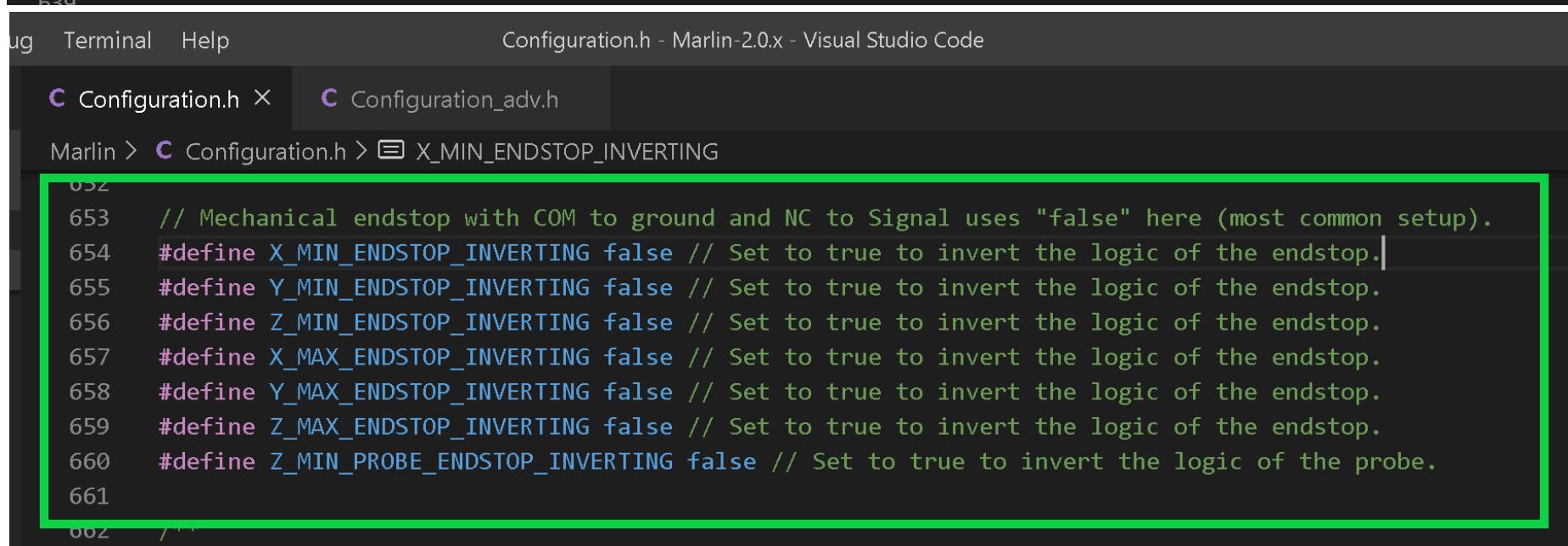
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > ENDSTOPPULLUPS

```

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL 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

```



Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > X_MIN_ENDSTOP_INVERTING

```

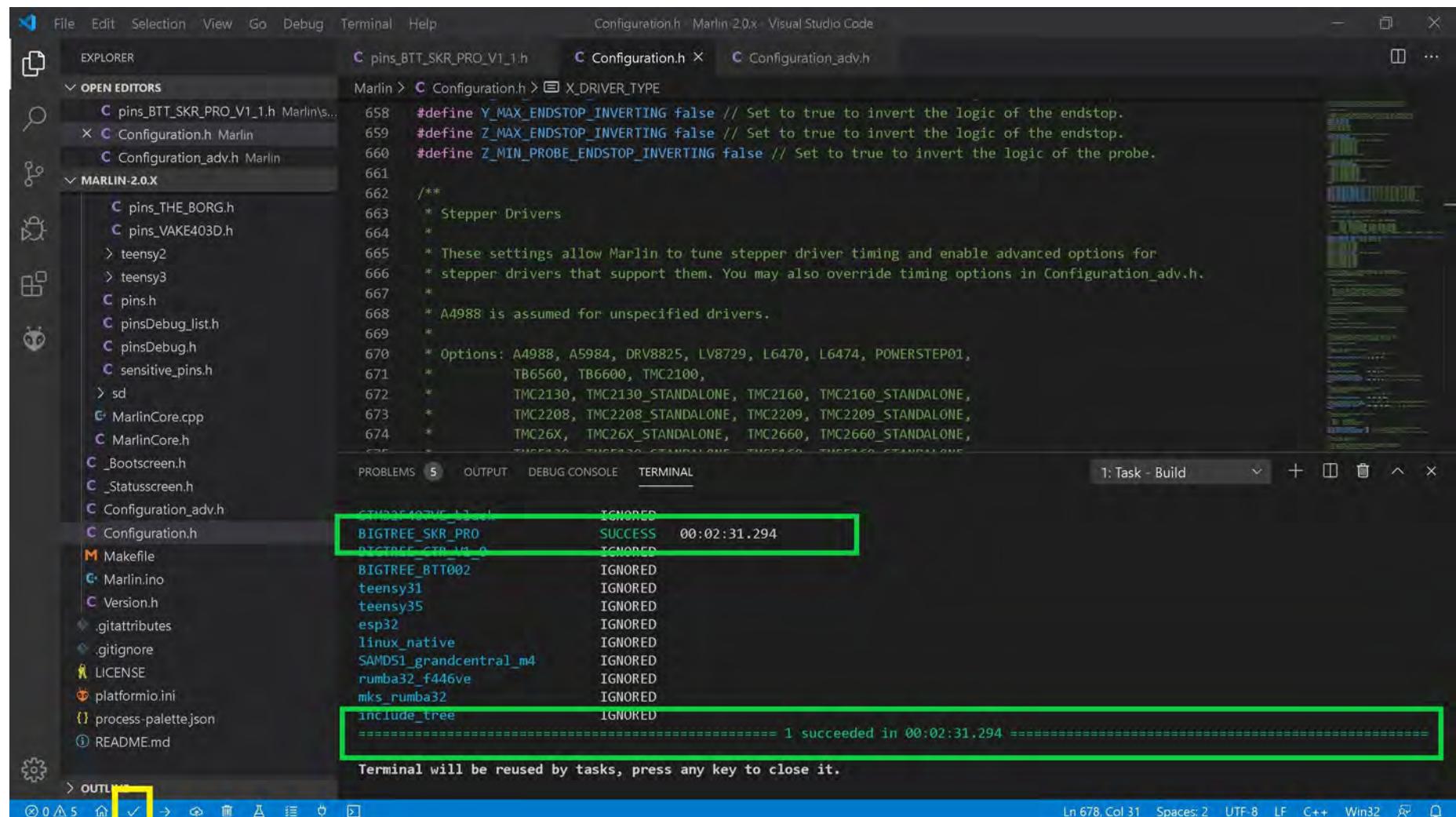
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
655 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
656 #define 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 /**

```

- 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.



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".
- Code Editor:** Displays the Configuration.h file with code related to endstop inversion and stepper driver timing.
- Terminal:** Shows the build log output. A yellow box highlights the "COMPILED FIRMWARE" section, which includes the following entries:

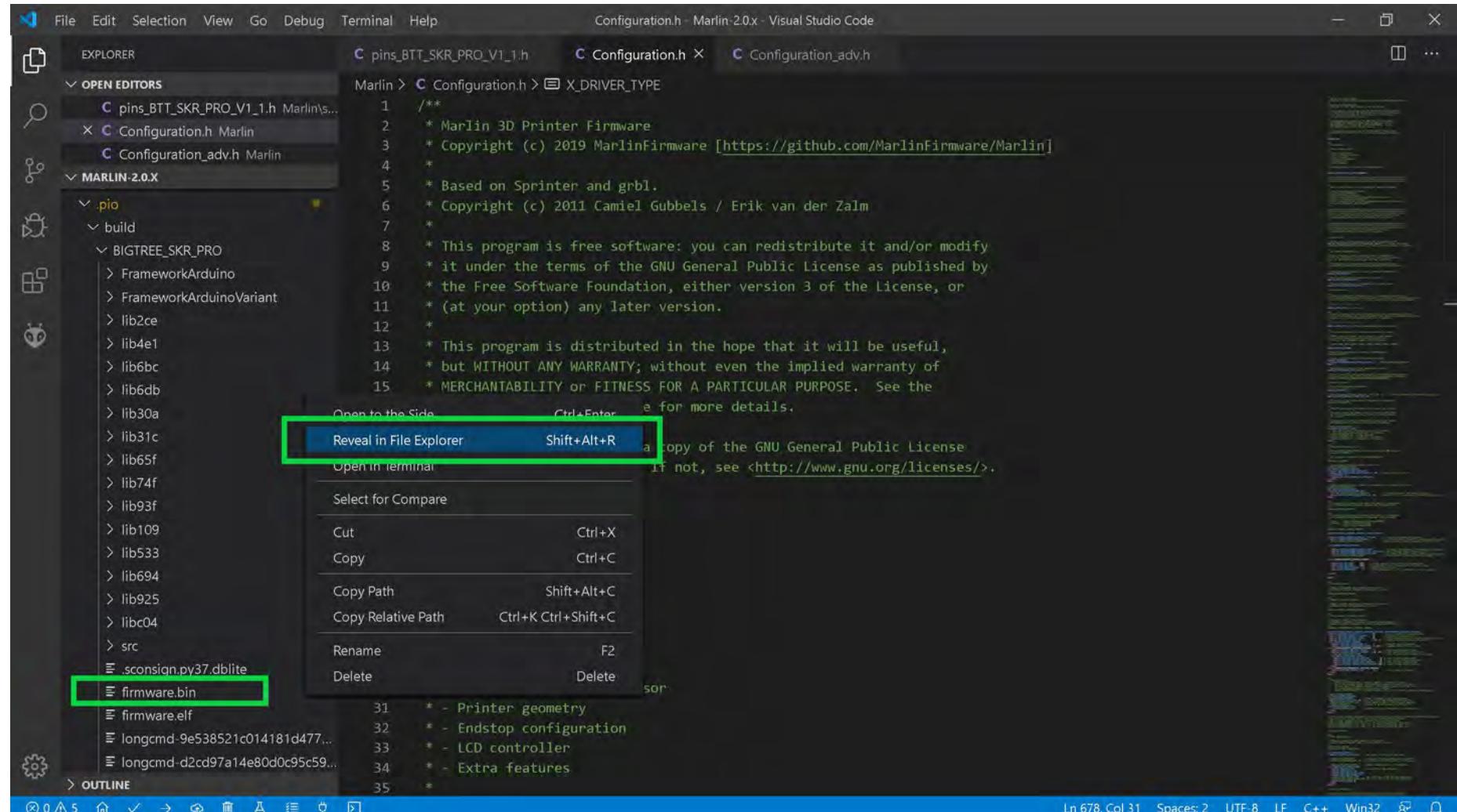
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
BIGTREE_BTT002	IGNORED	
teensy31	IGNORED	
teensy35	IGNORED	
esp32	IGNORED	
linux_native	IGNORED	
SAMD51_grandcentral_m4	IGNORED	
rumba32_f446ve	IGNORED	
mks_rumba32	IGNORED	
include_tree	IGNORED	

 Below this, a green box highlights the message: "===== 1 succeeded in 00:02:31.294 =====".
- Bottom Status Bar:** Shows the terminal status as "Terminal will be reused by tasks, press any key to close it." and the current file status as "0 A 5" with a yellow checkmark icon.

- 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

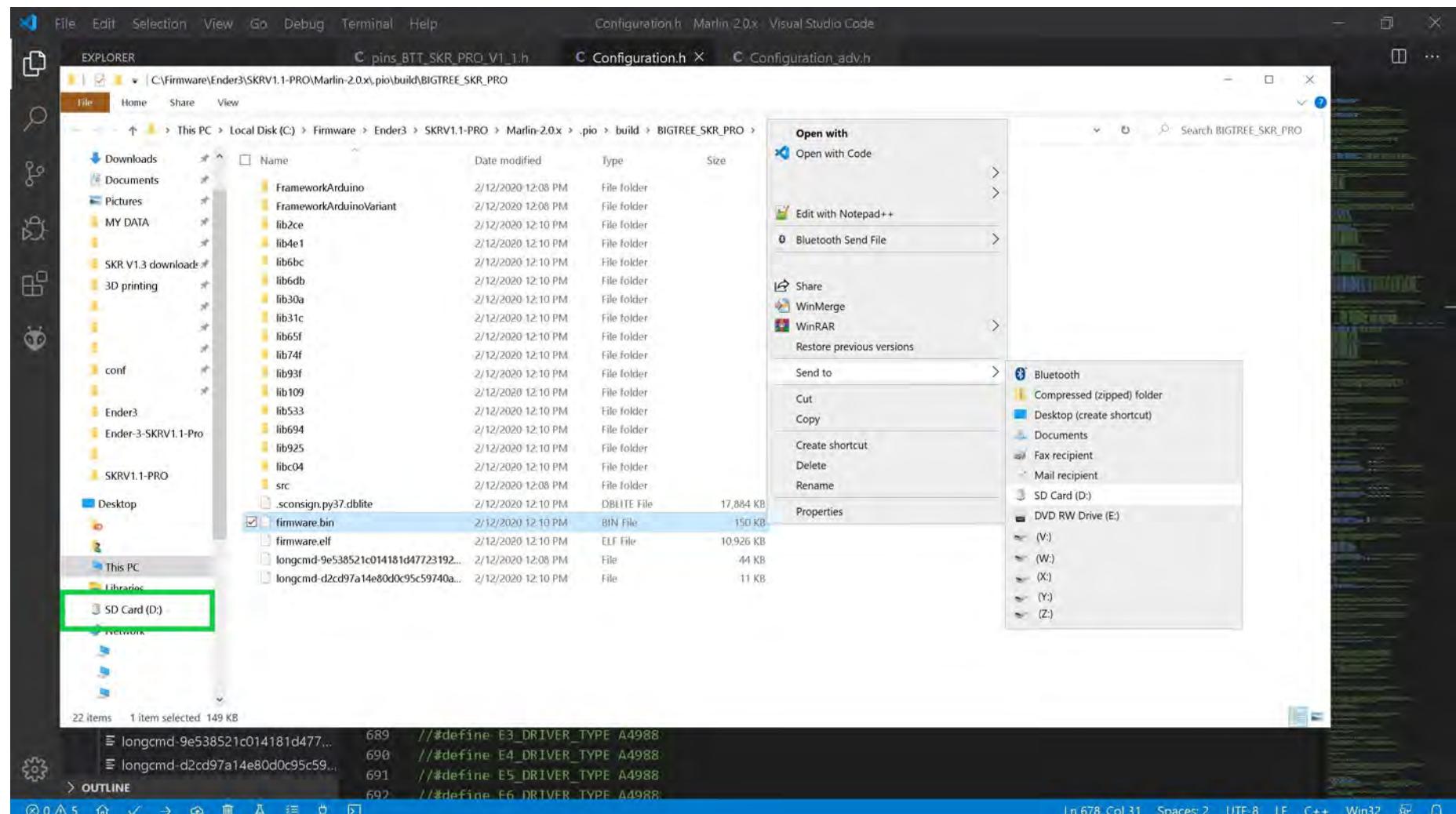
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



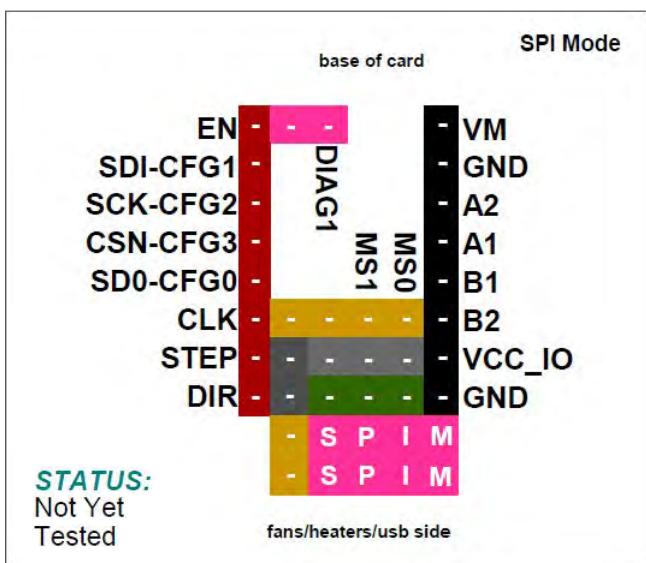
- Go to the next page.

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

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



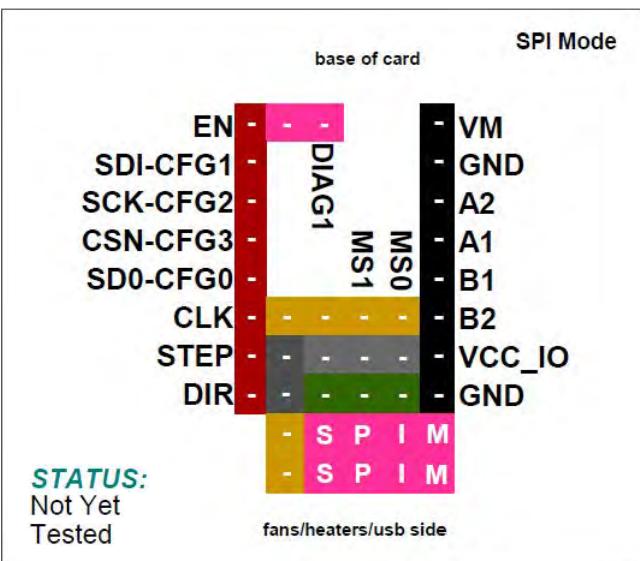
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

**BIQU TMC5160 V1.2**SPI Mode

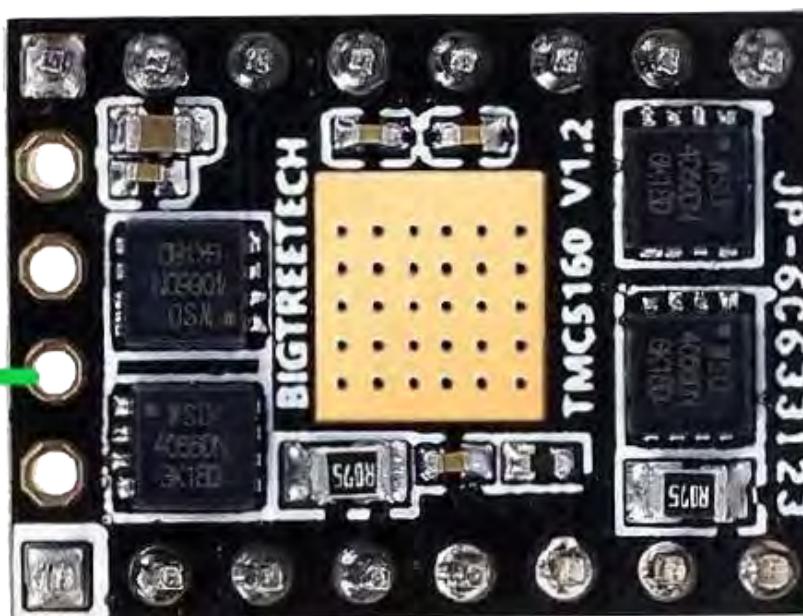
Note: You can use 50% to 90% of the calculated I_{RMS} ($I_{MAX}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

Driver Chip	<h1>Steps are set inside of your Firmware</h1>	
BBIQU® TMC5160 SPI Mode Maximum 256 Subdivision 40V DC 4.3A (peak)		
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.075 Ω	$I_{MAX} = 4.333$ See Appendix B #8. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90 = 3.900$	Current Limit is set by the current sense resistors (R_s). Use 50% - 90% of I_{MAX}.

BIQU TMC5160 V1.2SPI Mode

Note: The TMC5160 V1.2 by default comes in SPI mode. The BIQU TMC5160 does NOT come with a POT or "V_{ref} Test point" location because the IRMS is set inside of the Firmware.



NOTE: BIQU TMC5160 has the ability to do sensor-less homing. By default the DIAG1 pin is **NOT soldered onto the driver board. Therefore, for any axis you want sensor-less homing enabled, YOU WILL HAVE to solder on the DIAG1 pin.**

SPI Mode

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

$R_s = R_{050}$ is 0.05 Ohms

$R_s = R_{062}$ is 0.062 Ohms

$R_s = R_{068}$ is 0.068 Ohms

$R_s = R_{075}$ is 0.075 Ohms

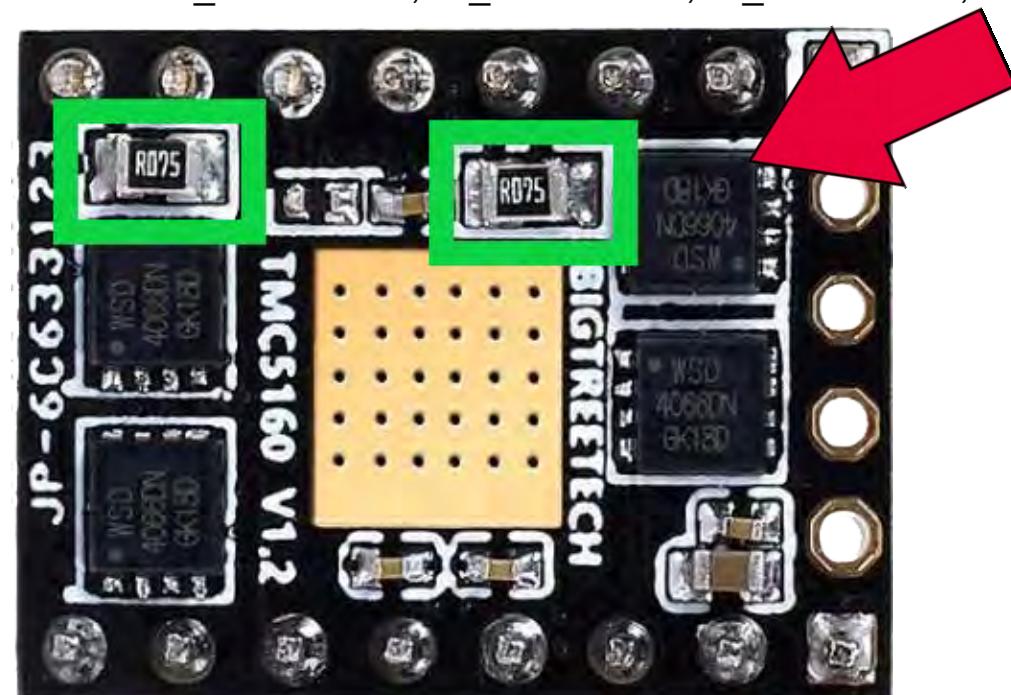
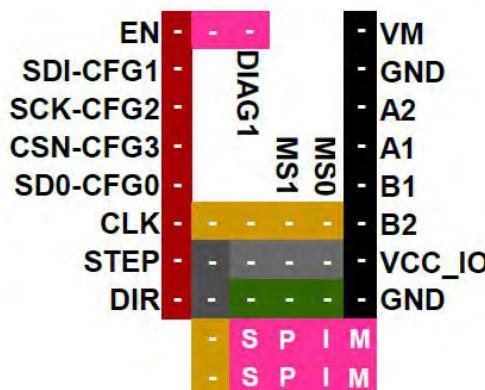
$R_s = R_{100}$ is 0.1 Ohms

$R_s = R_{110}$ is 0.11 Ohms

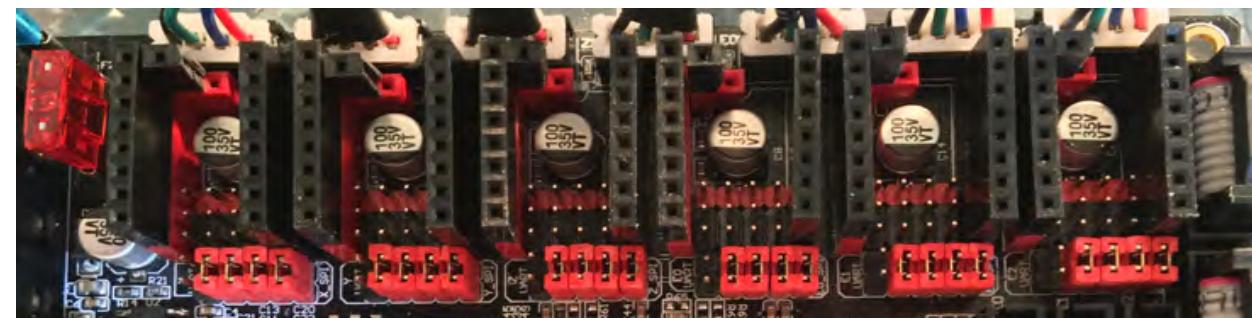
$R_s = R_{150}$ is 0.15 Ohms

$R_s = R_{200}$ is 0.2 Ohms

$R_s = R_{220}$ is 0.22 Ohms

**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](#), and 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 ("//").

```

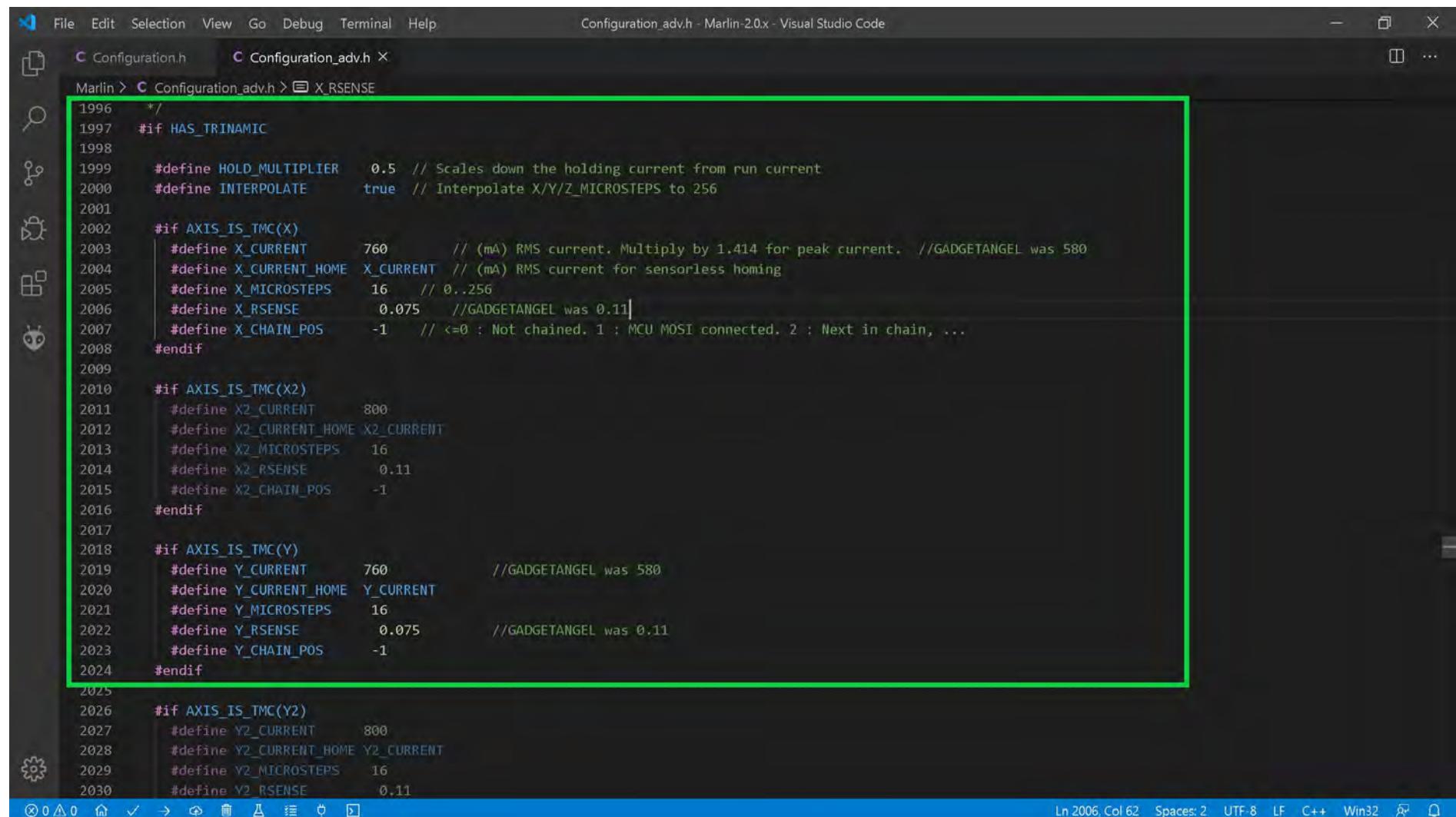
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 686, Col 1
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'T
677 "
678 #define X_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value 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 TMC5160 //GADGETANGEL was commented out and had the value 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 TMC5160 V1.2 Drivers in SPI Mode

- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5160 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.075. Ensure "Y_RSENSE" is set to 0.075.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

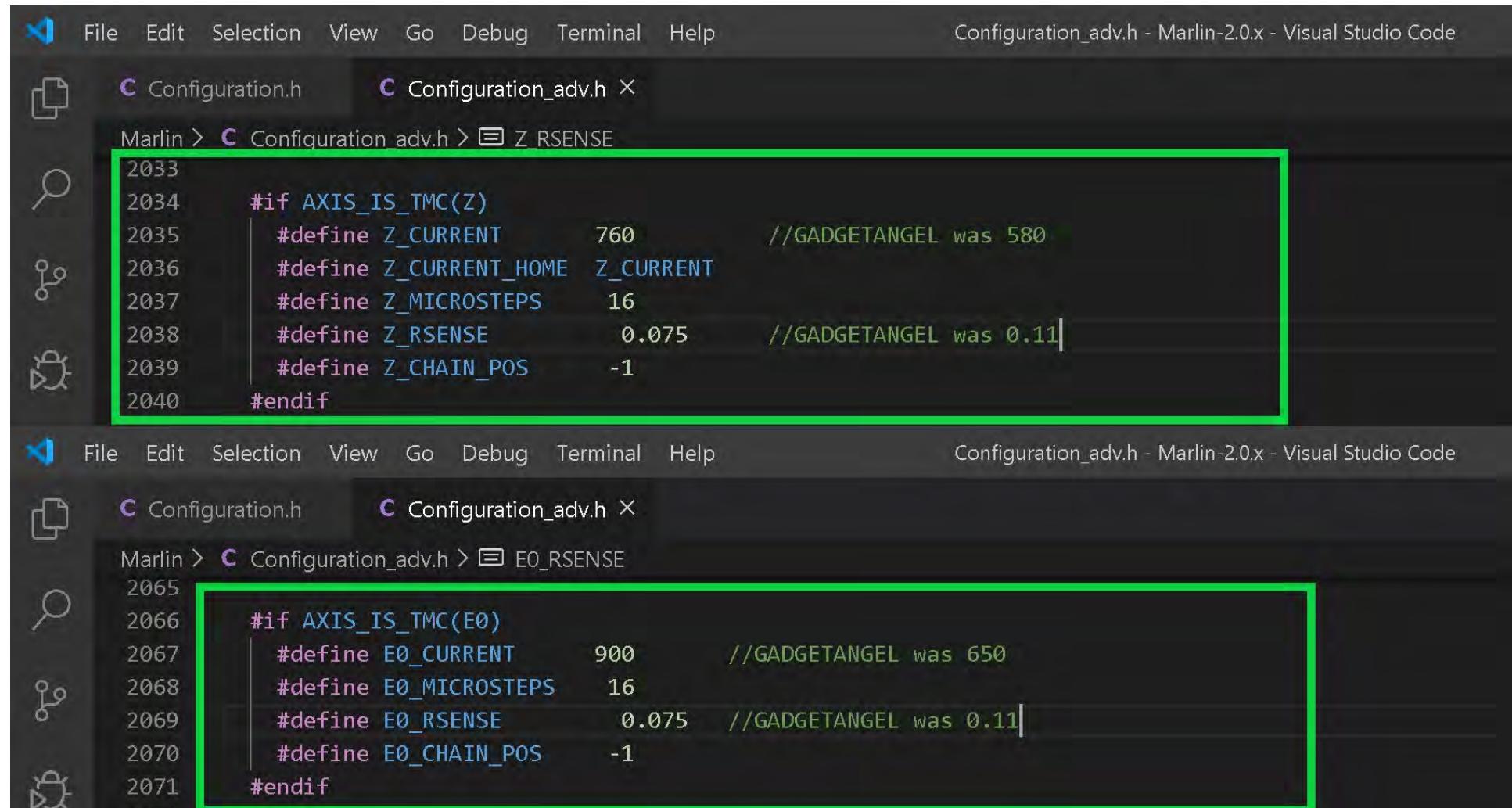
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE        true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003     #define X_CURRENT      760      // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004     #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005     #define X_MICROSTEPS   16       // 0..256
2006     #define X_RSENSE        0.075    //GADGETANGEL was 0.11]
2007     #define X_CHAIN_POS     -1       // <=0 : Not chained. 1 : MCU MOST 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      //GADGETANGEL was 580
2020     #define Y_CURRENT_HOME Y_CURRENT
2021     #define Y_MICROSTEPS   16
2022     #define Y_RSENSE        0.075    //GADGETANGEL was 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
2029     #define Y2_MICROSTEPS  16
2030     #define Y2_RSENSE       0.11
2031

```

- 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 V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.075. Ensure "E0_RSENSE" is set to 0.075.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".



```

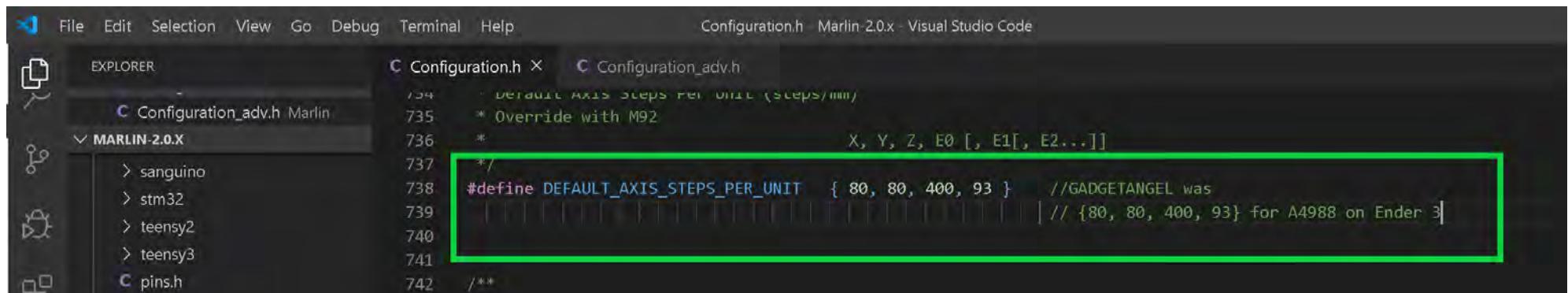
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z RSENSE
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT      760          //GADGETANGEL was 580
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS   16
2038         #define Z_RSENSE        0.075        //GADGETANGEL was 0.11
2039         #define Z_CHAIN_POS    -1
2040     #endif
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > E0_RSENSE
2065
2066     #if AXIS_IS_TMC(E0)
2067         #define E0_CURRENT      900          //GADGETANGEL was 650
2068         #define E0_MICROSTEPS   16
2069         #define E0_RSENSE        0.075        //GADGETANGEL was 0.11
2070         #define E0_CHAIN_POS    -1
2071     #endif

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- 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 Configuration.h X Configuration_adv.h

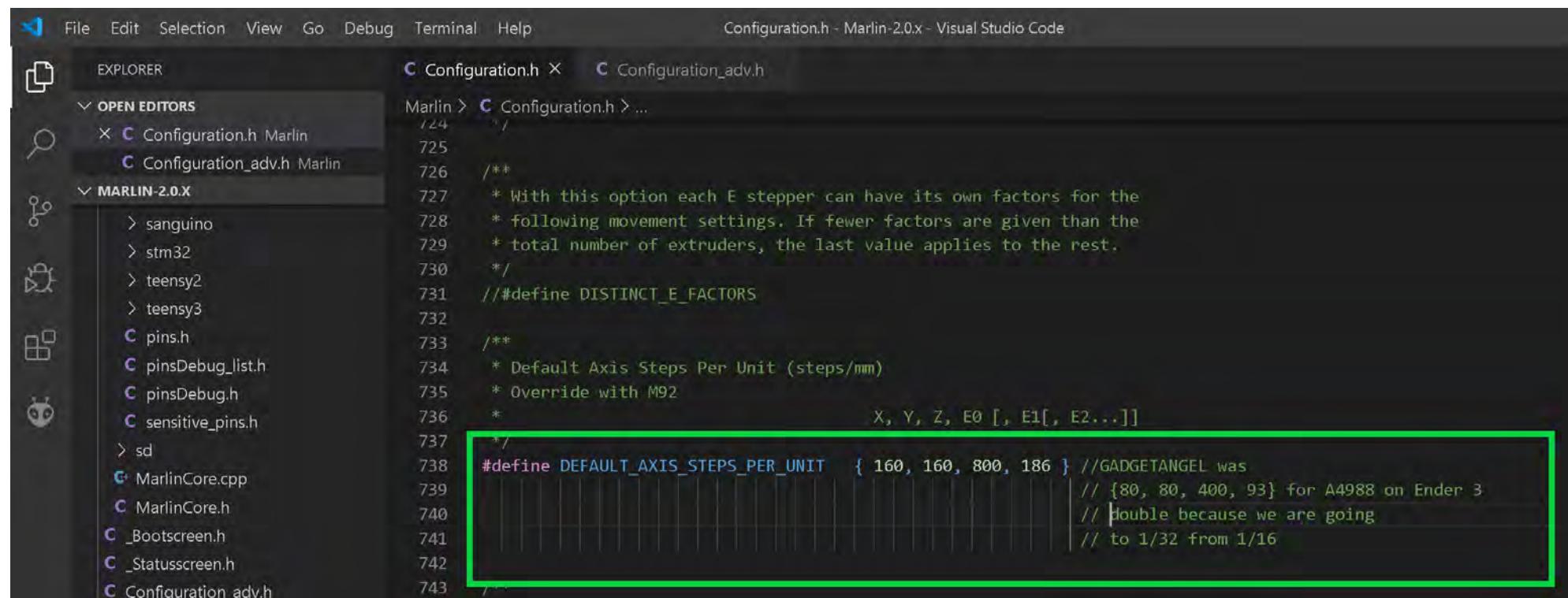
MARLIN-2.0.X Configuration.h Marlin Configuration_adv.h Marlin

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X Configuration.h Marlin Configuration_adv.h Marlin

```

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 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 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 its SPI pins

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

EXPLORER Configuration.h Configuration_adv.h ×

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X sanguino

```
Marlin > Configuration_adv.h > TMC_USE_SW_SPI
2144 * The default SW SPI pins are defined the respective pins files,
2145 * but you can override or define them here.
2146 */
2147 #define TMC_USE_SW_SPI //GADGETANGEL was commented out
2148 //#define TMC_SW_MOST -1
```

- 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 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 Configuration.h Configuration_adv.h ×

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X sanguino stm32 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

```
Marlin > Configuration_adv.h > STEALTHCHOP_XY
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
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 */
```

- 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 Configuration.h Configuration_adv.h X
OPEN EDITORS Marlin > Configuration_adv.h > HYBRID_THRESHOLD
Configuration.h Marlin 2232
Configuration_adv.h Marlin 2233
MARLIN-2.0.X 2234
sanguino 2235
stm32 2236
teensy2 2237
teensy3 2238
pins.h 2239
pinsDebug_list.h 2240
pinsDebug.h 2241
sensitive_pins.h 2242
sd 2243
MarlinCore.cpp 2244
MarlinCore.h 2245
Bootscreen.h 2246
_Statusscreen.h 2247
Configuration_adv.h 2248
Configuration.h 2249
Makefile 2250
Marlin.ino 2251
Version.h 2252
.gitattributes 2253
.gitignore 2254
Configurations-release-2.0.3.zip 2255
2256
2257
2258
2259

/*
 * 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 //GADGETANGEL 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

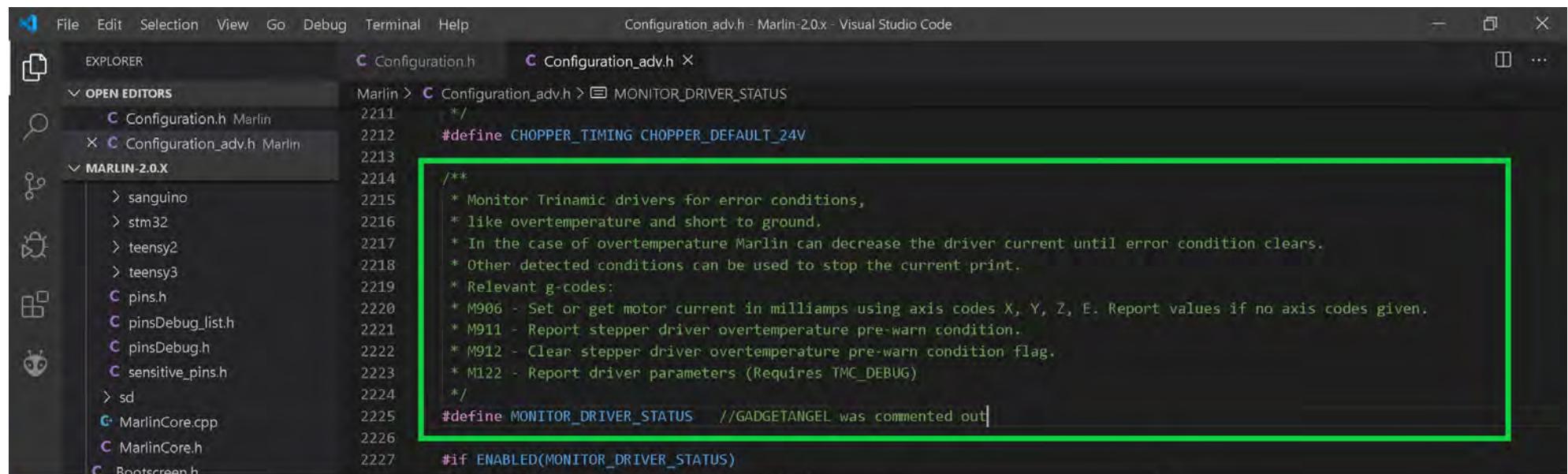
/*
 */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

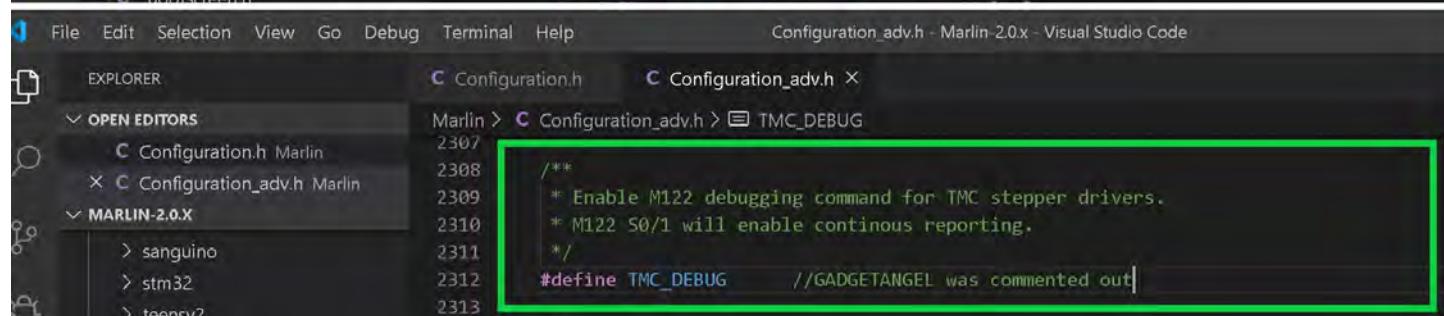
- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these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



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
  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 millamps 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 //GADGETANGEL was commented out
  2226
  2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



```

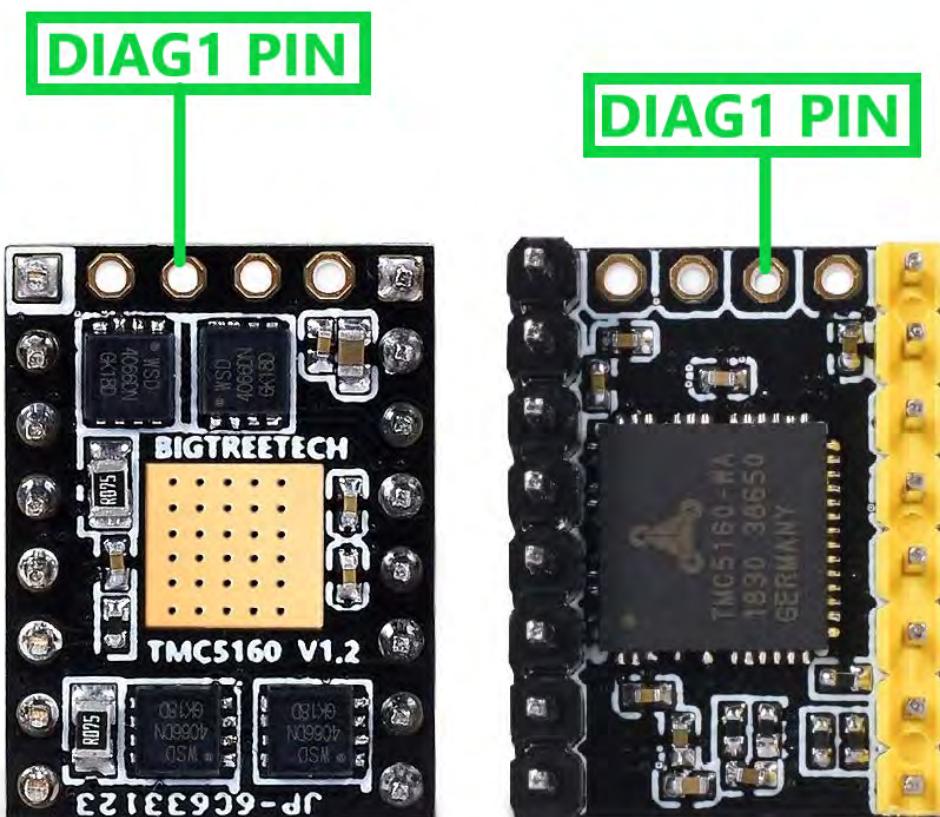
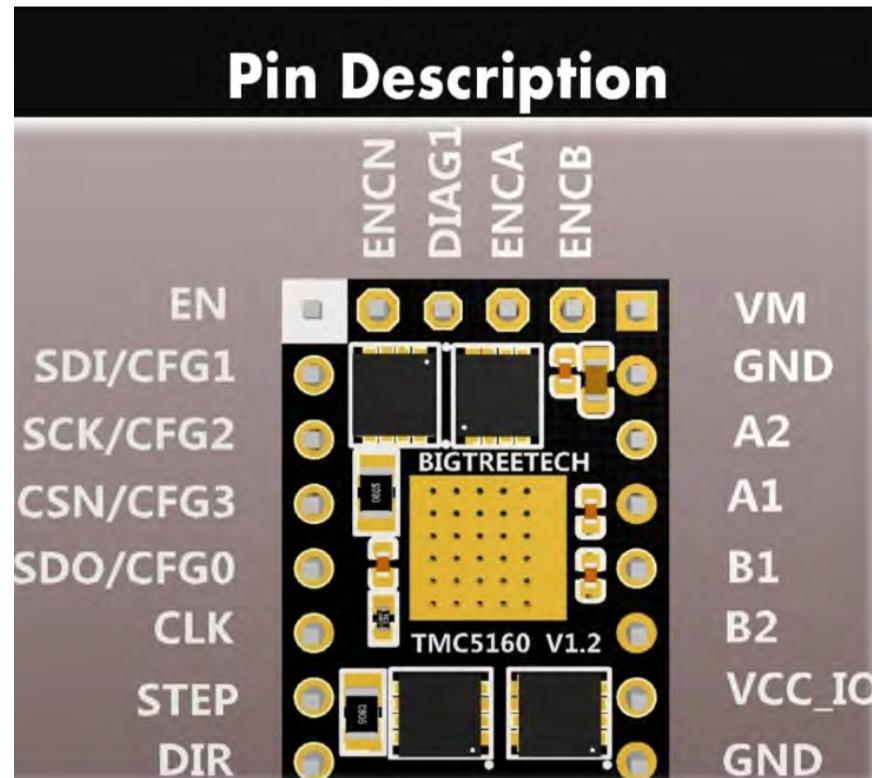
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG
  2307
  2308 /**
  2309 * Enable M122 debugging command for TMC stepper drivers.
  2310 * M122 S0/1 will enable continuous reporting.
  2311 */
  2312 #define TMC_DEBUG //GADGETANGEL was commented out
  2313

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

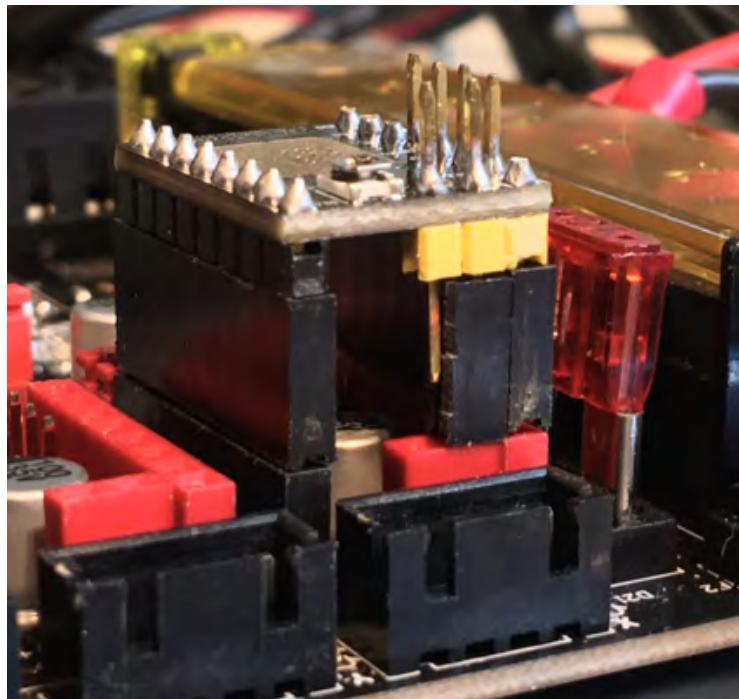
- This next section covers sensor-less homing which is available for the TMC5160 in SPI mode. I want to enable it BUT for the TMC5160 I first have to solder on the DIAG1 pin onto each TMC5160 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and 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.



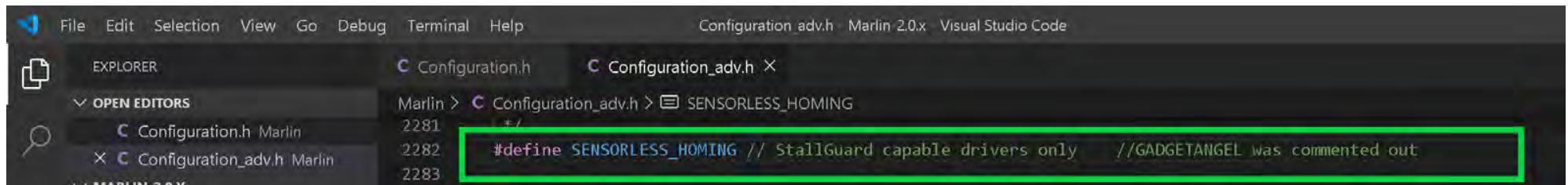
- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5160 into the SKR PRO V1.1 board the DIAG1 pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

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"



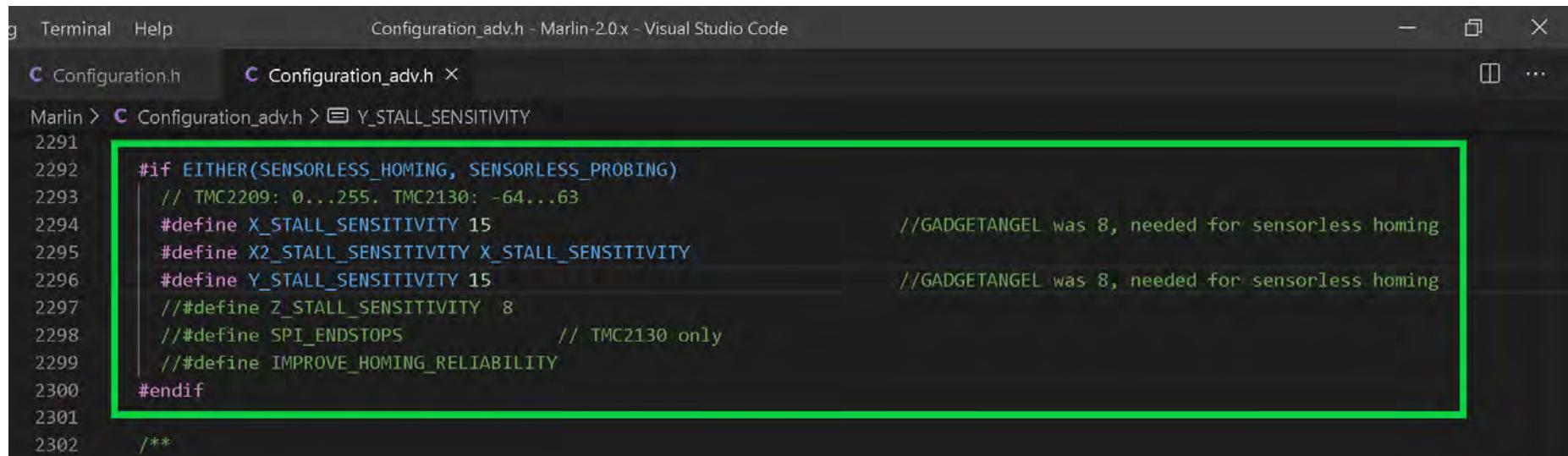
```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h X
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING
 2281 */
 2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out
 2283
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Next we set the "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".

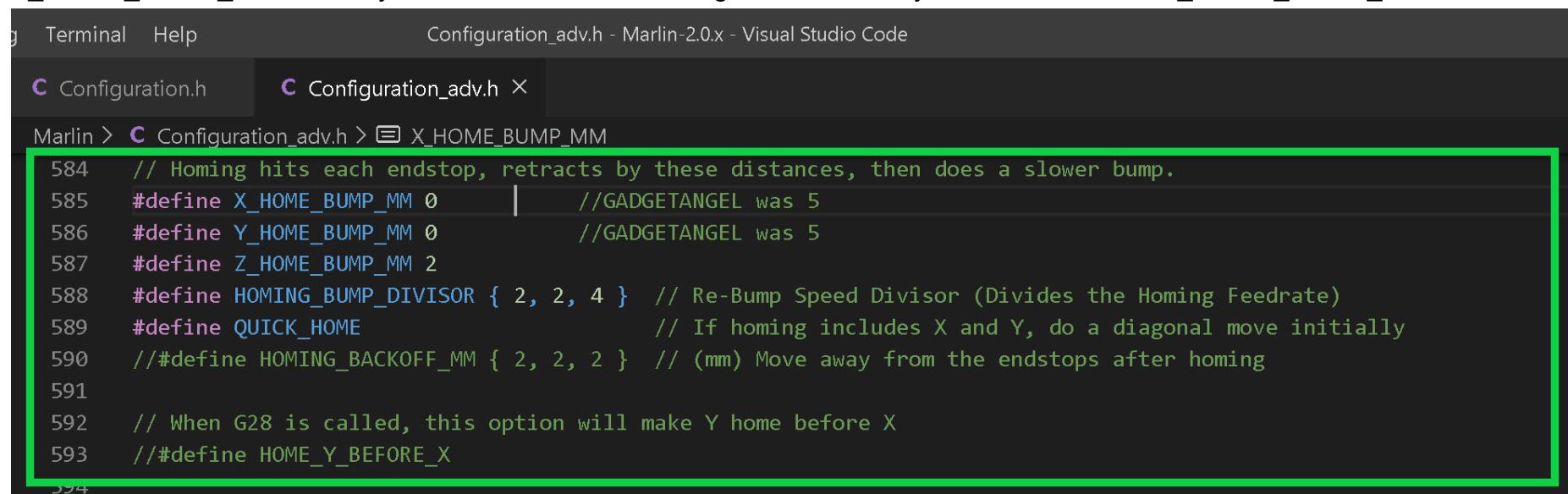


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- 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.



```

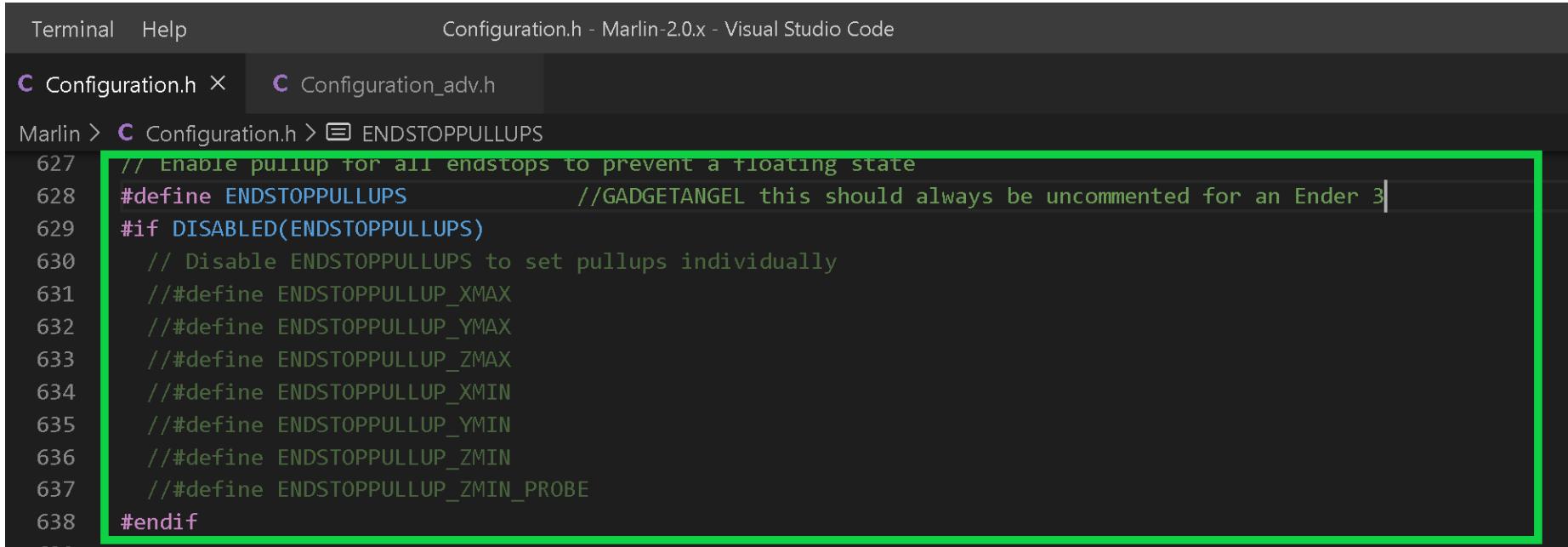
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL 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

```

- 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.



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

Configuration.h X Configuration_adv.h

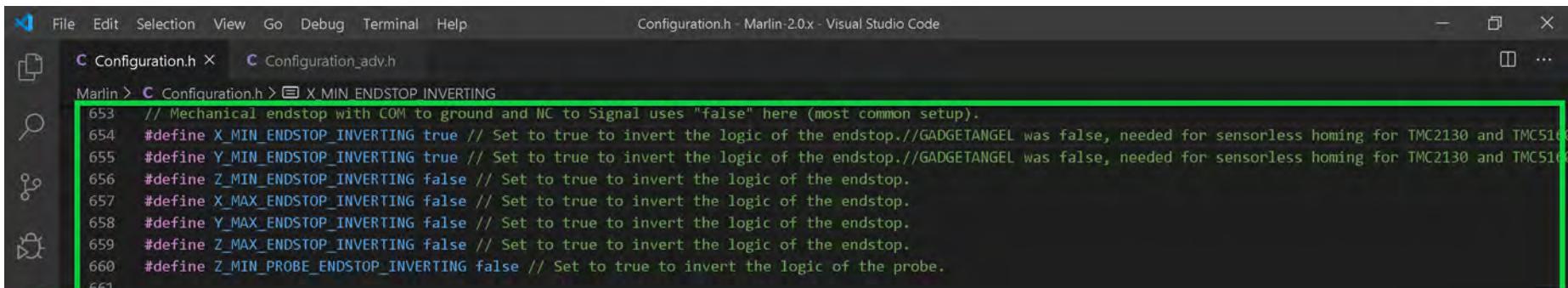
Marlin > Configuration.h > ENDSTOPPULLUPS

```

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL 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

```

- Next to allow sensor-less homing to work (while using the BIQU TMC5160) 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

Configuration.h X Configuration_adv.h

Marlin > Configuration.h > X MIN ENDSTOP INVERTING

```

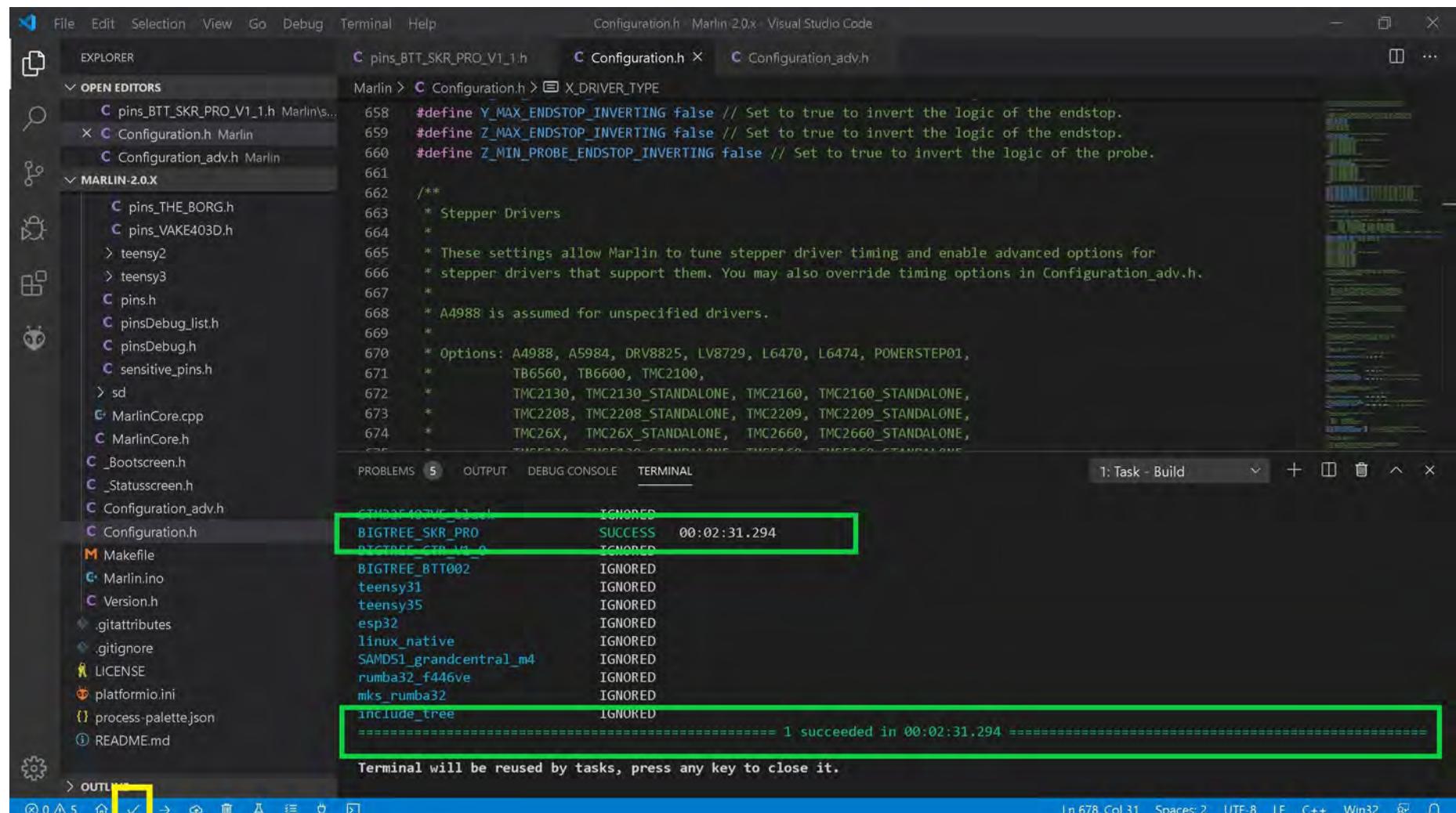
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for 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

```

- 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.



```

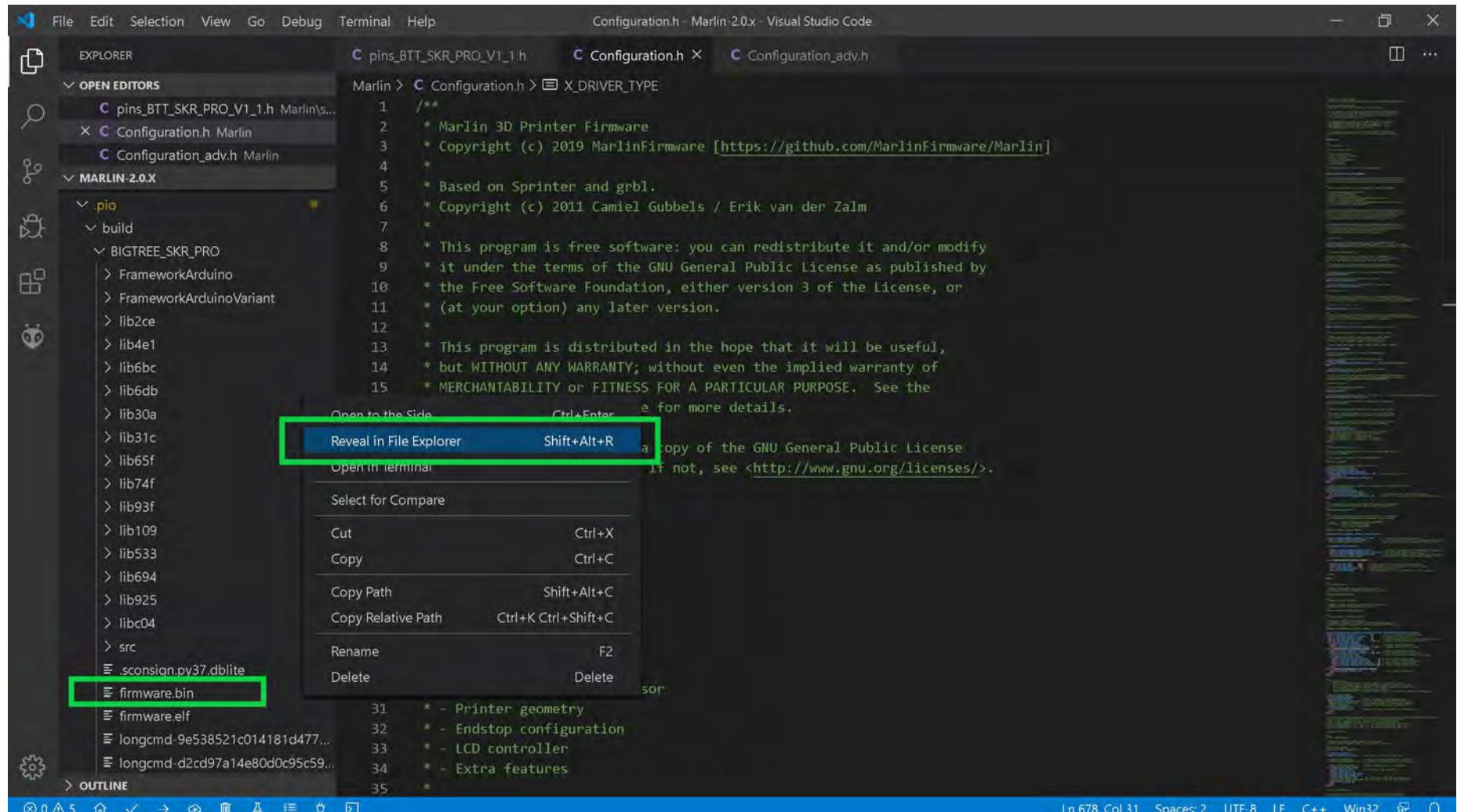
File Edit Selection View Go Debug Terminal Help Configuration.h Marlin 2.0.x Visual Studio Code
EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\pins.h
Configuration.h Marlin Configuration.h
Configuration_adv.h Marlin 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 + ×
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
=====
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 TMC5160 V1.2 Drivers in SPI Mode

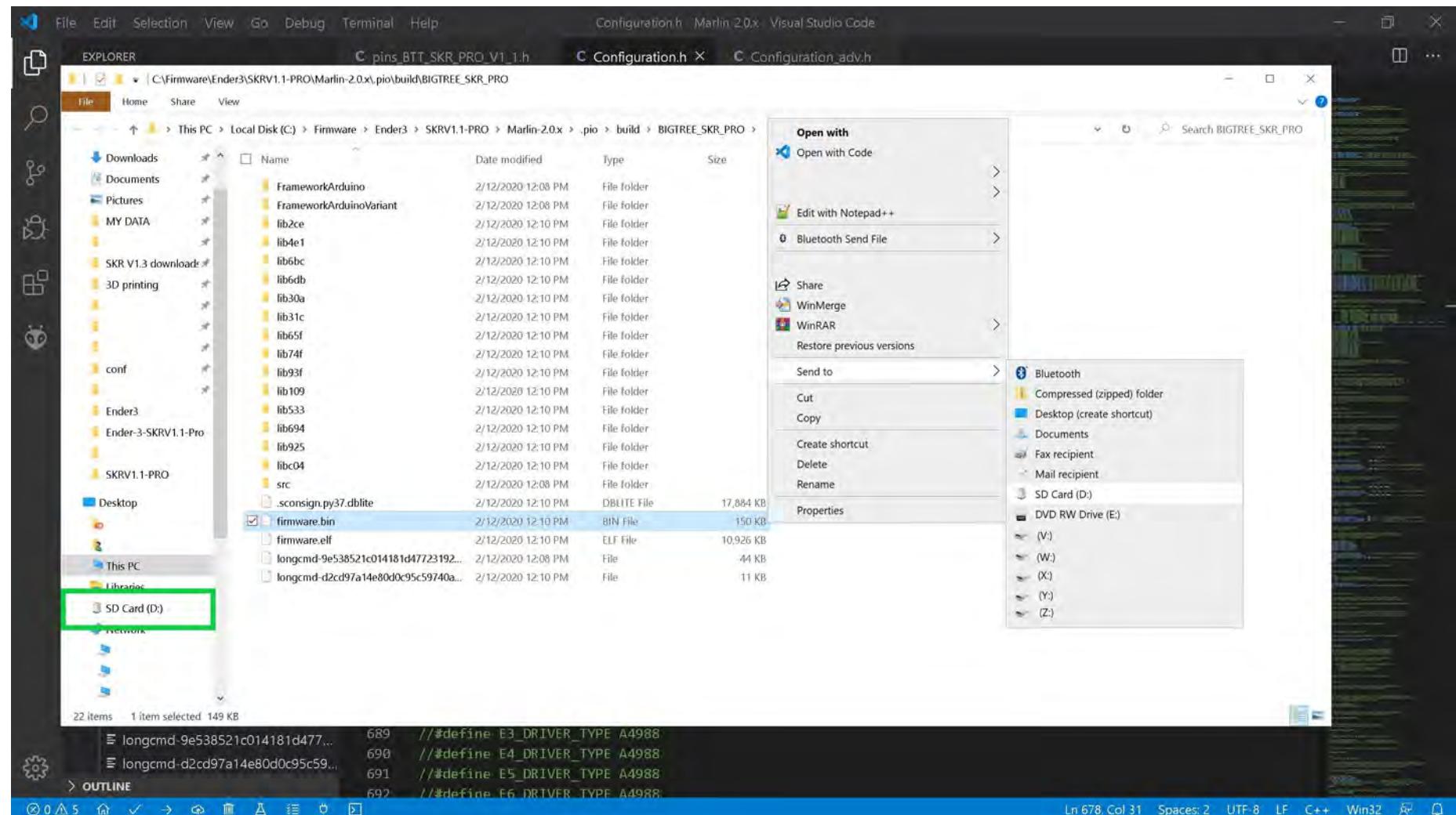
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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



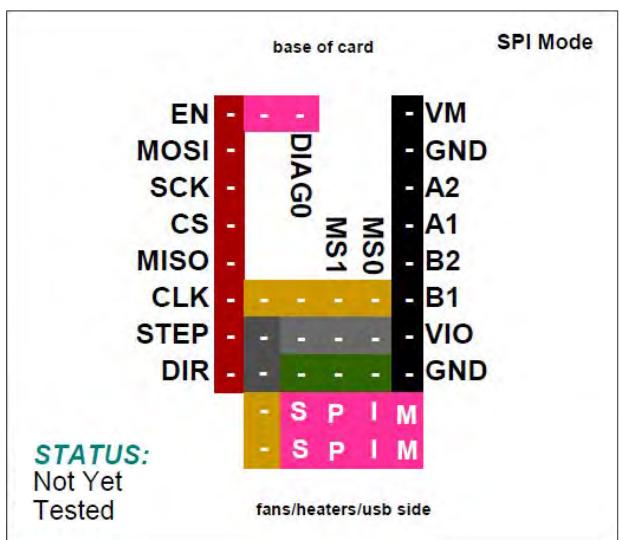
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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".



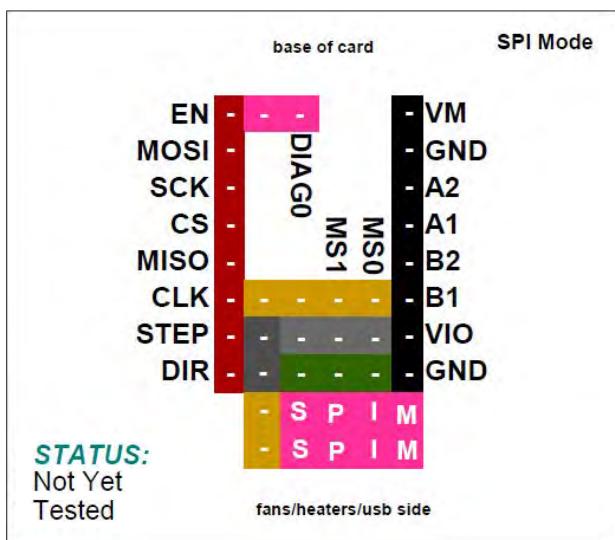
- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR PRO V1.1 board. Turn on the power to the printer (i.e. the SKR PRO V1.1 board) and the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

**BIQU TMC5161 V1.0****SPI Mode**

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

See the next page for further information.

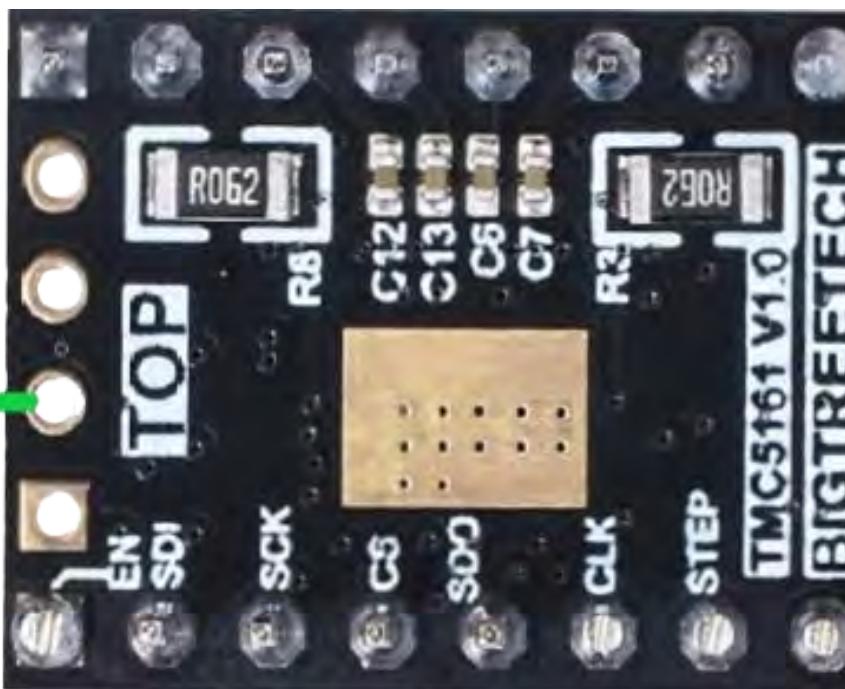
<p>Driver Chip</p> <p>BIQU® TMC5161</p> <p>SPI Mode Maximum 256 Subdivision 40V DC 5.2A (peak)</p>		
<p>Driving Current Calculation Formula</p> <p>R_S(Typical Sense Resistor)=0.062Ω</p>	<p>$I_{MAX}=5.2419$ See Appendix B #9. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90=4.718$</p>	<p>Current Limit is set by the current sense resistors (R_s). Use 50% - 90% of I_{MAX}.</p>



BIQU TMC5161 V1.0

SPI Mode

Note: The TMC5161 V1.0 by default comes in SPI mode. The BIQU TMC5161 does NOT come with a POT or "V_{ref} Test point" location because the IRMS is set inside of the Firmware.



NOTE: BIQU TMC5161 has the ability to do sensor-less homing. By default the DIAG0 pin is **NOT** soldered onto the driver board. Therefore, for any axis you want sensor-less homing enabled, **YOU WILL HAVE** to solder on the DIAG0 pin.

BIQU TMC5161 V1.0SPI Mode**SPI Mode**

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

$R_s = R050$ is 0.05 Ohms

$R_s = R062$ is 0.062 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R075$ is 0.075 Ohms

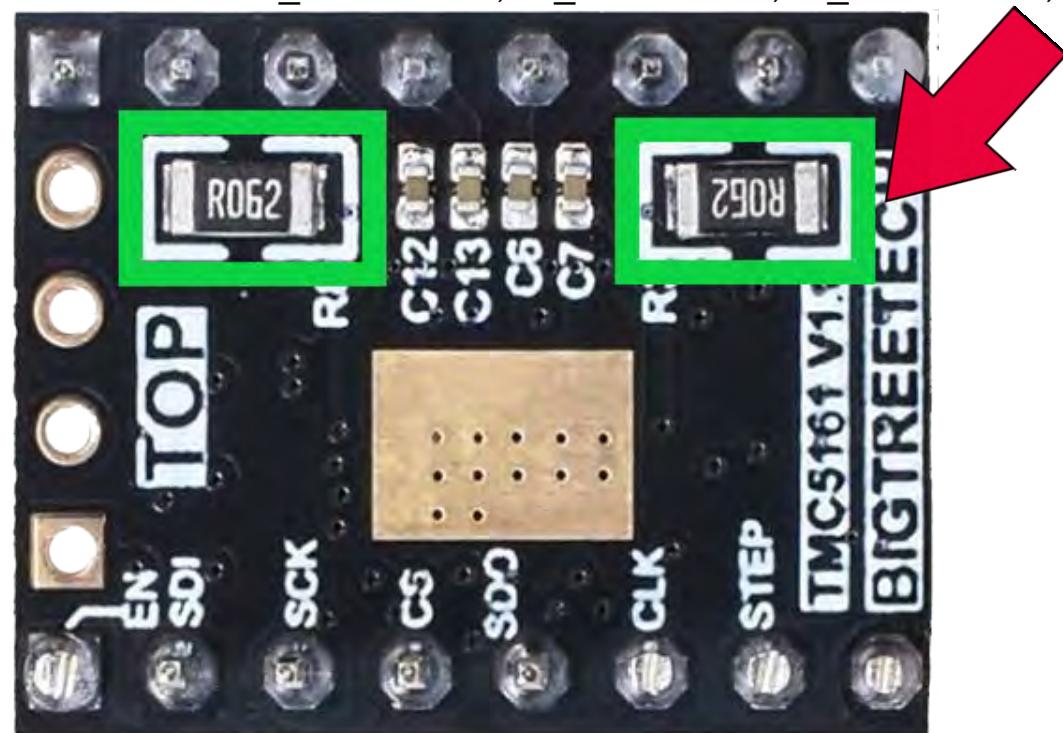
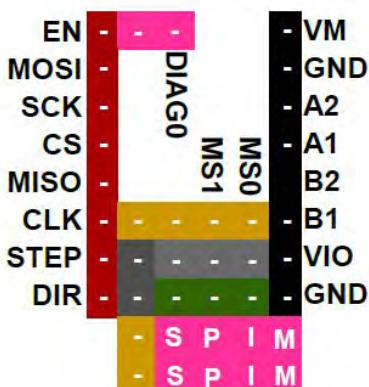
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

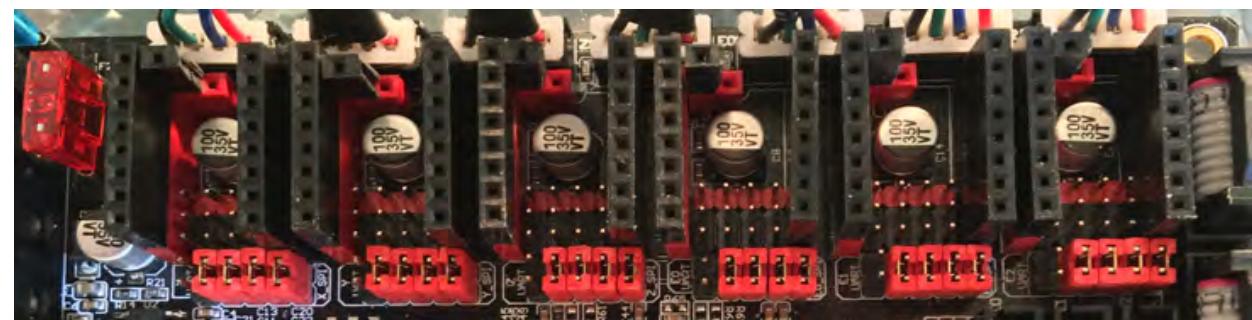
$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

**SPI**

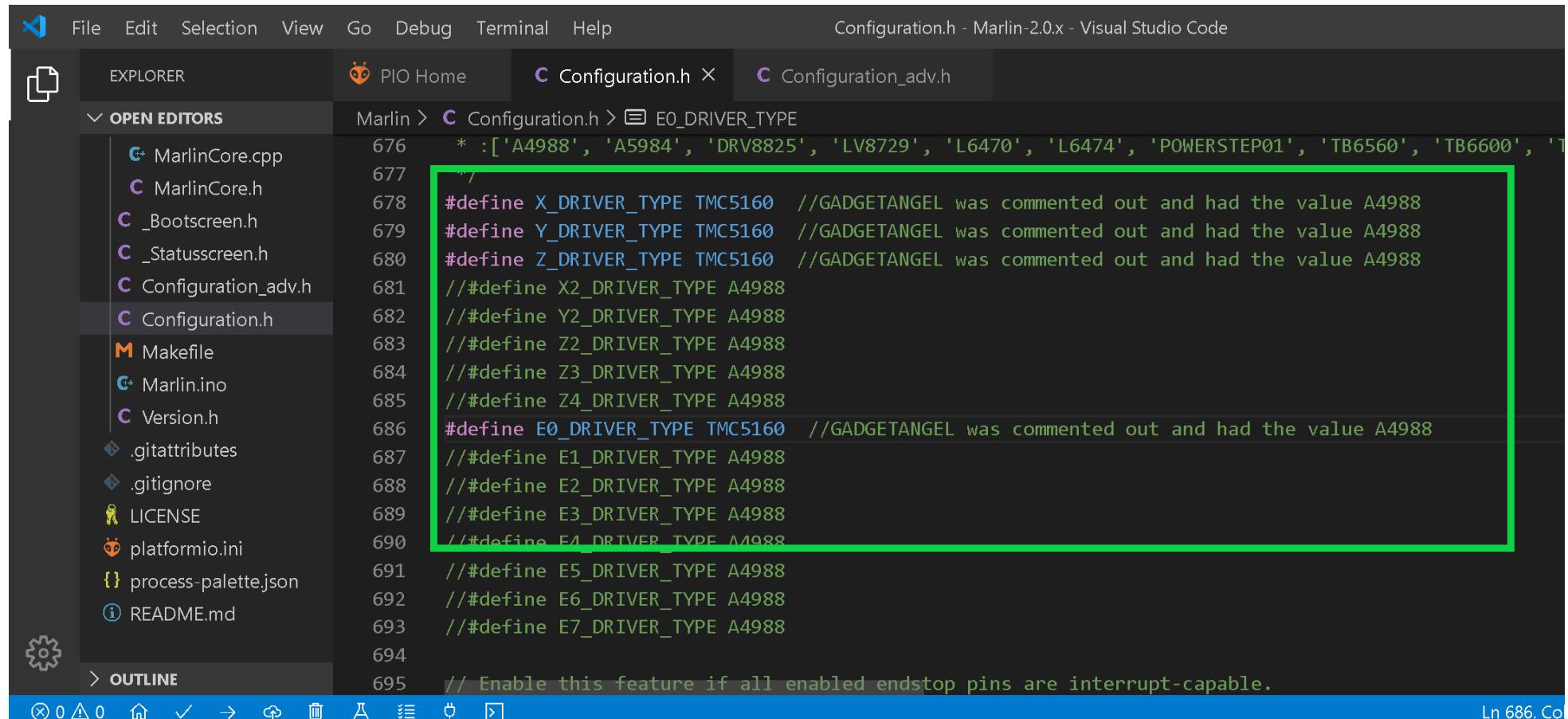
Note: Set JUMPERS "S","P","I","M" on the board!!



The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

NOTE: [Go to Appendix C](#), and then come back here for the changes to Marlin for BIQU TMC5161 V1.0 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC5161 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 TMC5161 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

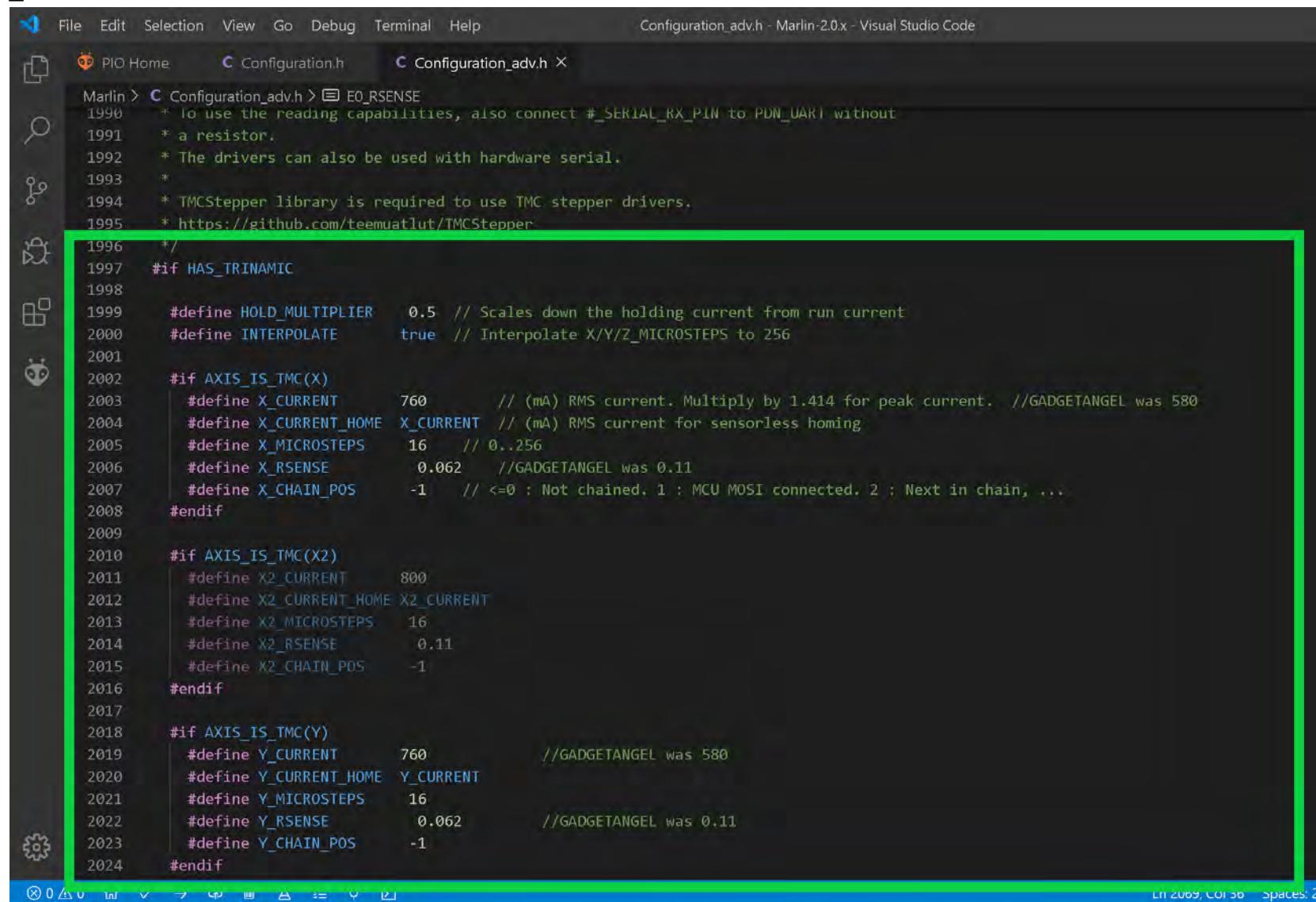
EXPLORER PIO Home Configuration.h X Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'T
677 "
678 #define X_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value 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 TMC5160 //GADGETANGEL was commented out and had the value 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 1

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5161 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.062. Ensure "Y_RSENSE" is set to 0.062.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

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 X
Marlin > C Configuration_adv.h > EO_RSENSE
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
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. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.062 //GADGETANGEL was 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 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.062 //GADGETANGEL was 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.062. Ensure "E0_RSENSE" is set to 0.062.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".

```

2034 #if AXIS_IS_TMC(Z)
2035     #define Z_CURRENT      760          //GADGETANGEL was 580
2036     #define Z_CURRENT_HOME Z_CURRENT
2037     #define Z_MICROSTEPS   16
2038     #define Z_RSENSE        0.062        //GADGETANGEL was 0.11
2039     #define Z_CHAIN_POS    -1
2040 #endif

```



```

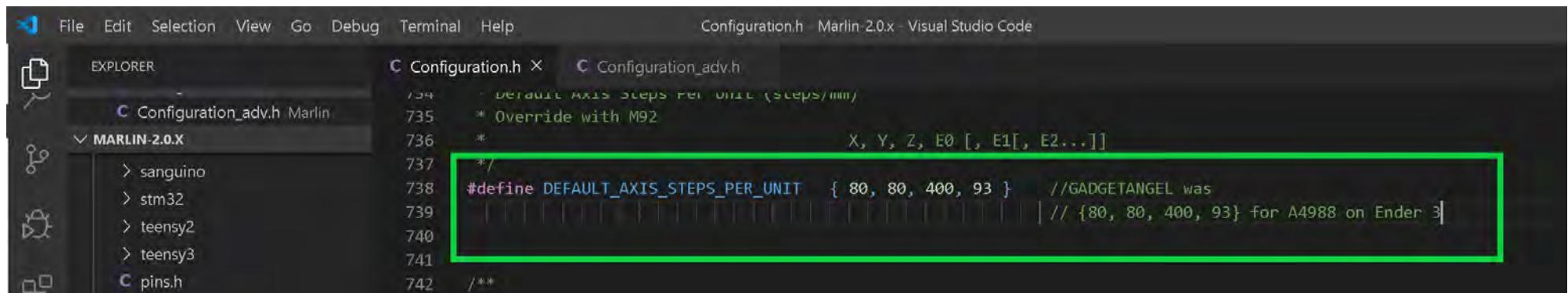
2066 #if AXIS_IS_TMC(E0)
2067     #define E0_CURRENT      900          //GADGETANGEL was 650
2068     #define E0_MICROSTEPS   16
2069     #define E0_RSENSE        0.062        //GADGETANGEL was 0.11
2070     #define E0_CHAIN_POS    -1
2071 #endif

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- 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 Configuration.h X Configuration_adv.h

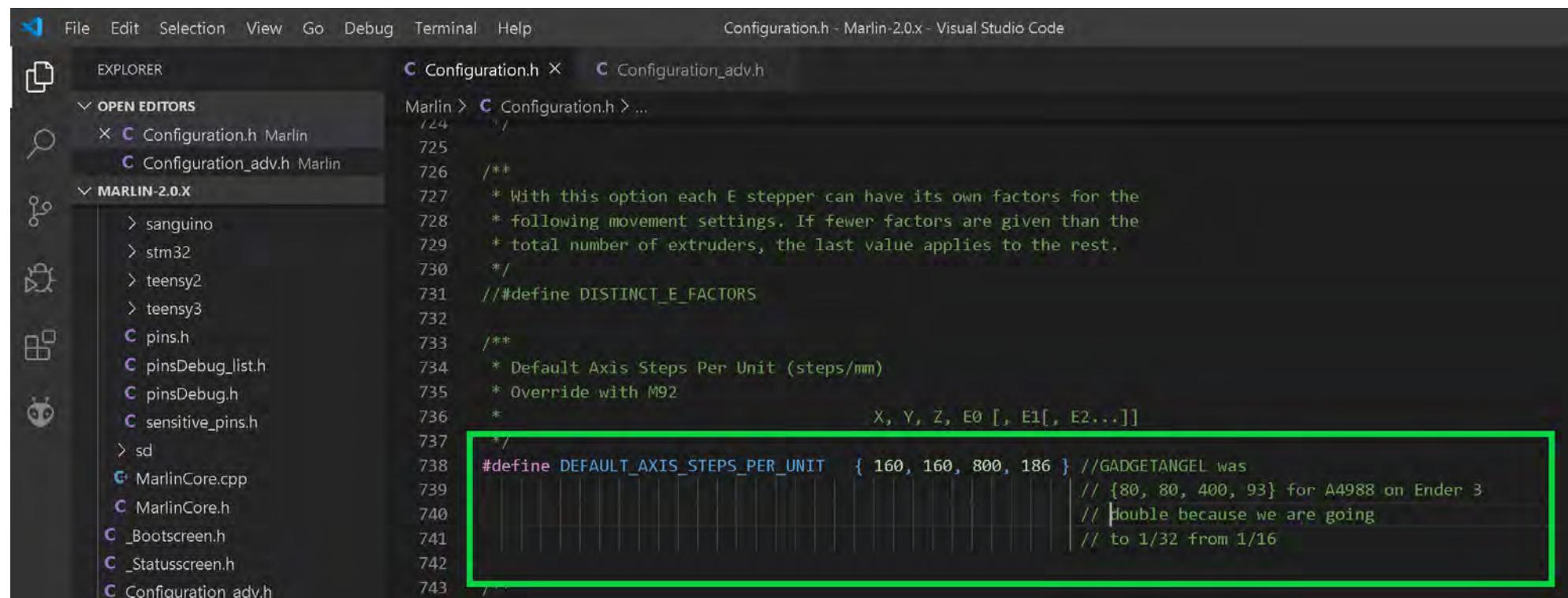
MARLIN-2.0.X

```

734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741 /**
742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your 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 Configuration.h X Configuration_adv.h

OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

```

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 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 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 its SPI pins

```

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

EXPLORER Configuration.h Configuration_adv.h
Marlin > Configuration.h > Configuration_adv.h > TMC_USE_SW_SPI
2144 * The default SW SPI pins are defined the respective pins files,
2145 * but you can override or define them here.
2146 */
2147 #define TMC_USE_SW_SPI //GADGETANGEL was commented out
2148 // #define TMC_SW_MOST -1

```

- 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 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

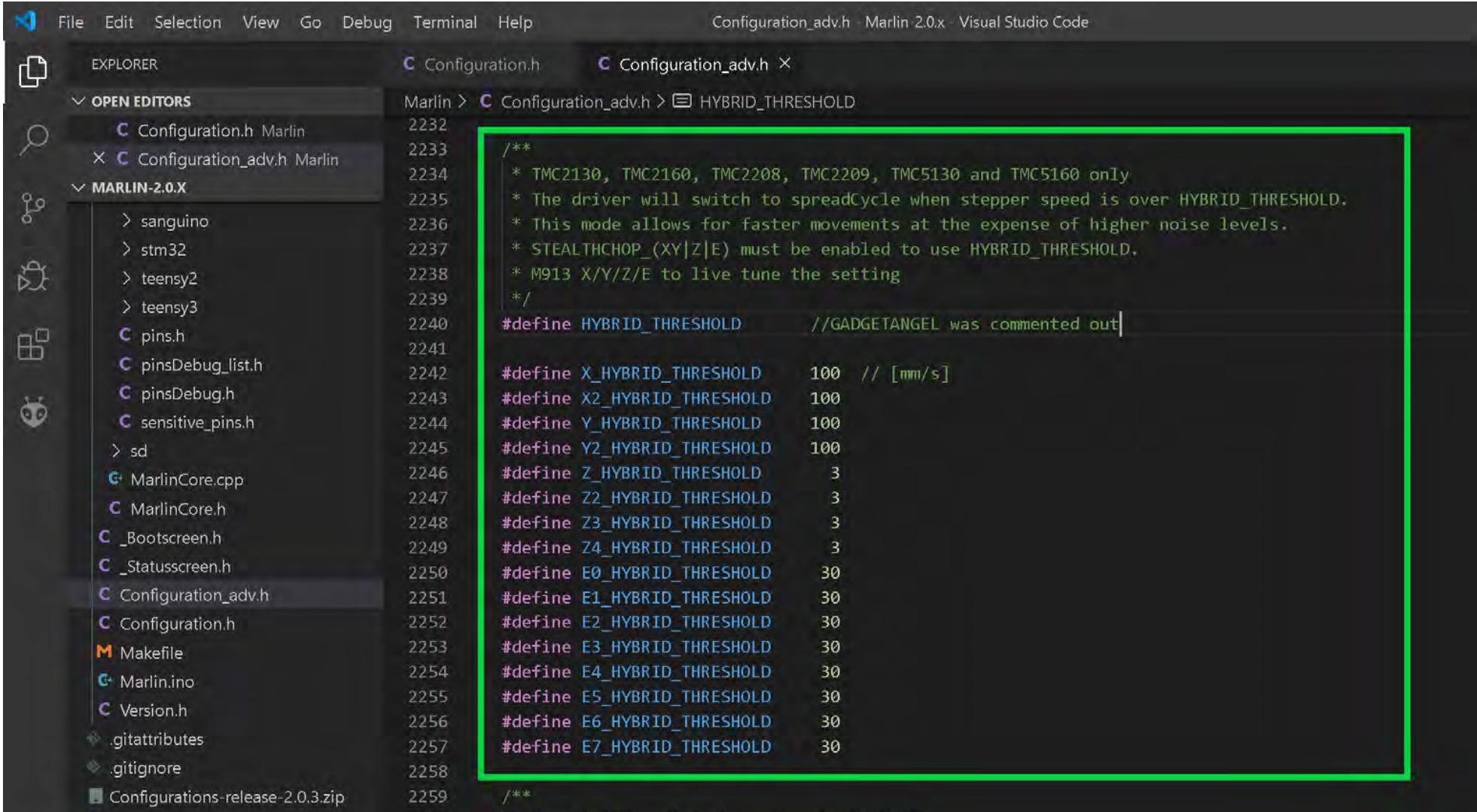
EXPLORER Configuration.h Configuration_adv.h
Marlin > Configuration.h > Configuration_adv.h > STEALTHCHOP_XY
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 */
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * 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 */

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 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 Configuration.h Configuration_adv.h

OPEN EDITORS Marlin > Configuration_adv.h > HYBRID_THRESHOLD

MARLIN-2.0.X

```

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 //GADGETANGEL 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 * H S1 S2 S3 S4 S5 S6 S7

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI 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.](#)

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
  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 millamps 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 //GADGETANGEL was commented out
  2226
  2227 #if ENABLED(MONITOR_DRIVER_STATUS)
  2228
  2229
  2230
  2231
  2232
  2233
  2234
  2235
  2236
  2237
  2238
  2239
  2240
  2241
  2242
  2243
  2244
  2245
  2246
  2247
  2248
  2249
  2250
  2251
  2252
  2253
  2254
  2255
  2256
  2257
  2258
  2259
  2260
  2261
  2262
  2263
  2264
  2265
  2266
  2267
  2268
  2269
  2270
  2271
  2272
  2273
  2274
  2275
  2276
  2277
  2278
  2279
  2280
  2281
  2282
  2283
  2284
  2285
  2286
  2287
  2288
  2289
  2290
  2291
  2292
  2293
  2294
  2295
  2296
  2297
  2298
  2299
  2300
  2301
  2302
  2303
  2304
  2305
  2306
  2307
  2308
  2309
  2310
  2311
  2312
  2313

```

```

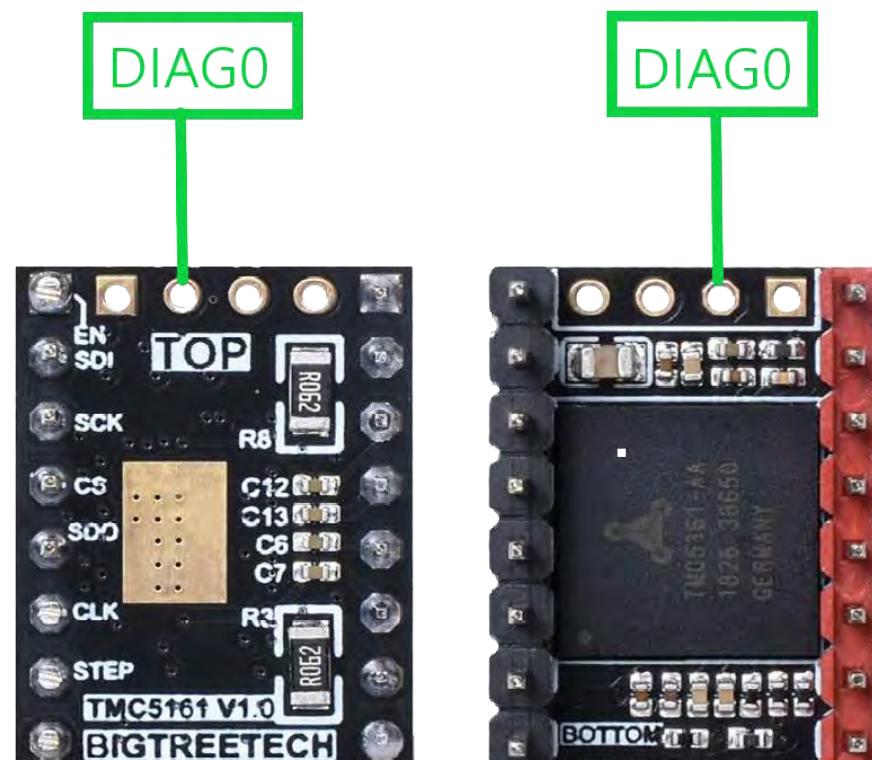
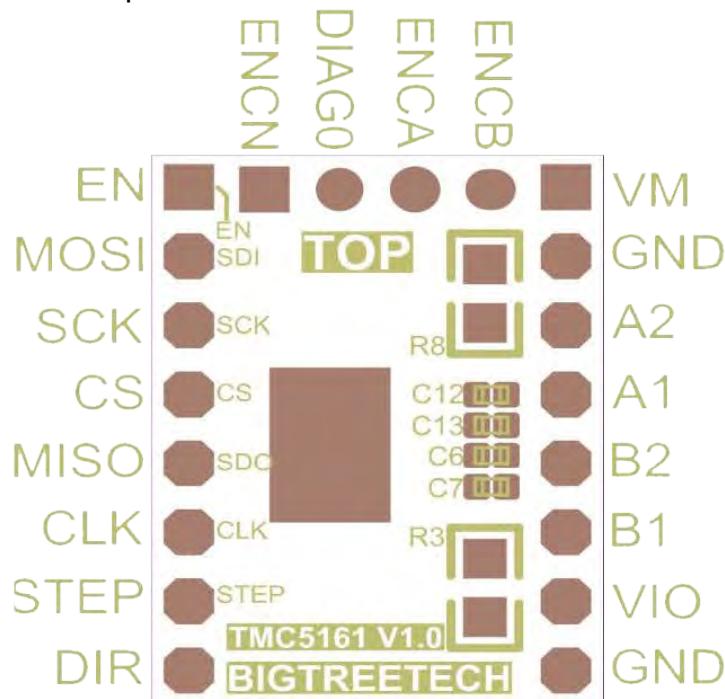
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG
  2307
  2308
  2309 /**
  2310 * Enable M122 debugging command for TMC stepper drivers.
  2311 * M122 S0/1 will enable continuous reporting.
  2312 */
  2313 #define TMC_DEBUG //GADGETANGEL was commented out

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

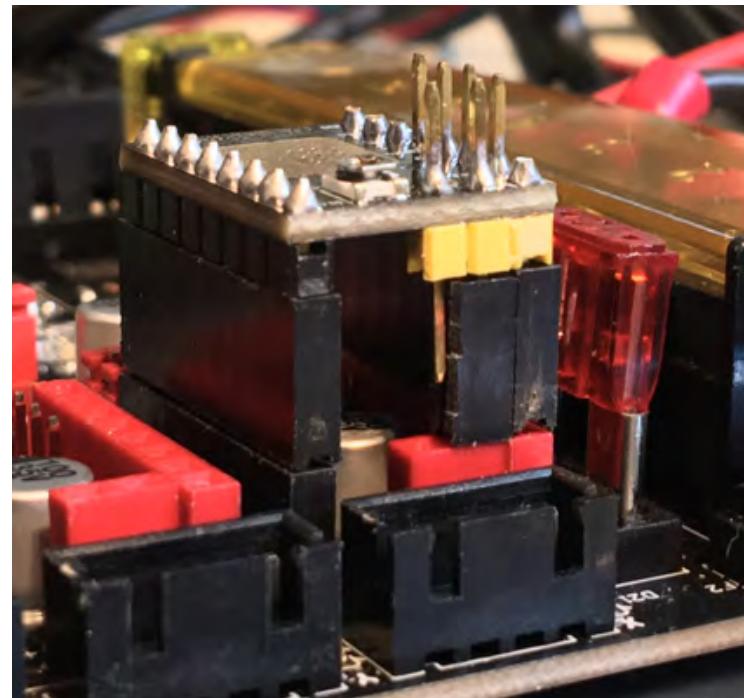
- This next section covers sensor-less homing which is available for the TMC5161 in SPI mode. I want to enable it BUT for the TMC5161 I first have to solder on the DIAG0 pin onto each TMC5161 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG0 pin for two TMC5161 drivers. Here is a picture of the TMC5161 V1.0 pin-out.



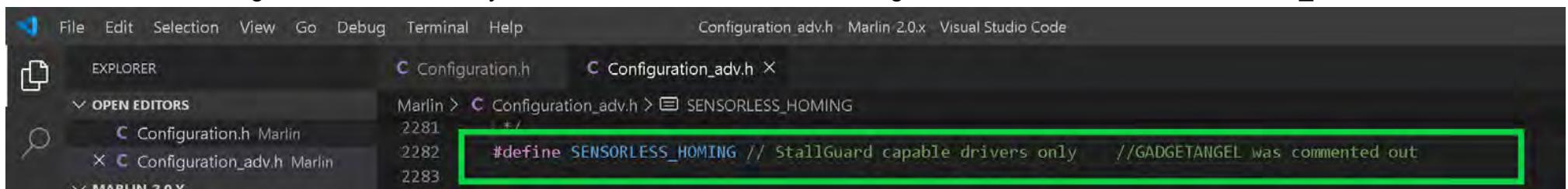
- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR PRO V1.1 board the **DIAG0** pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 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 DIAG0 pin on the TMC5161 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 DIAG0 pin on the Z axis TMC5161 is NOT connected to the board.** I plan to plug my Z axis' TMC5161 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"

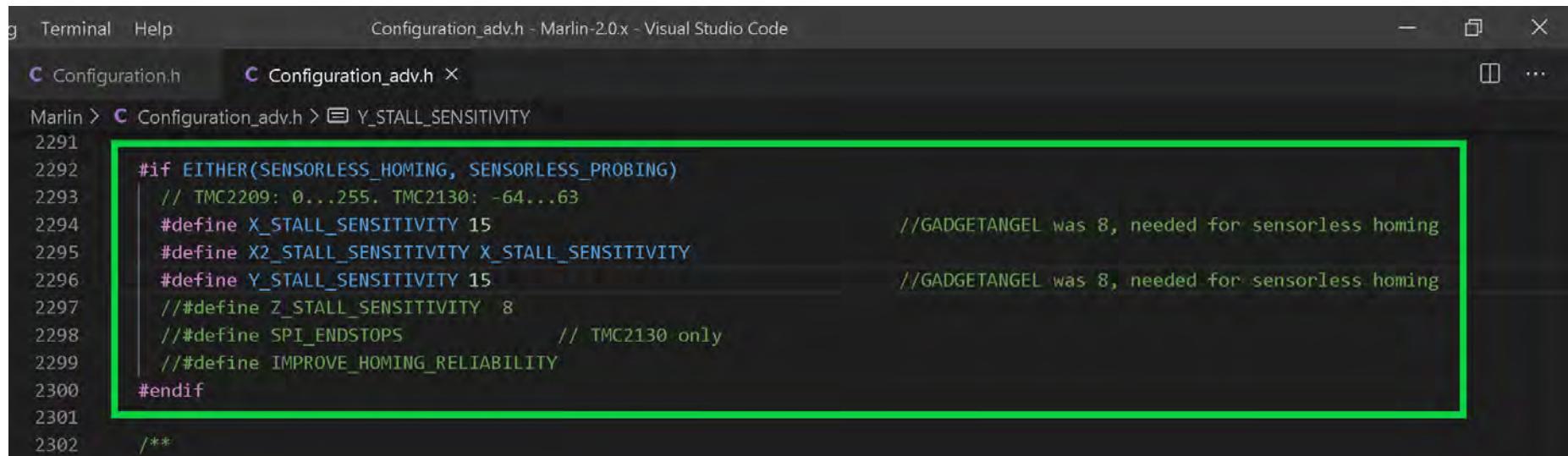


```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code  
EXPLORER C Configuration.h C Configuration_adv.h X  
OPEN EDITORS Marlin > C Configuration_adv.h > SENSORLESS_HOMING  
2281 */  
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out  
2283
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Next we set the "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".

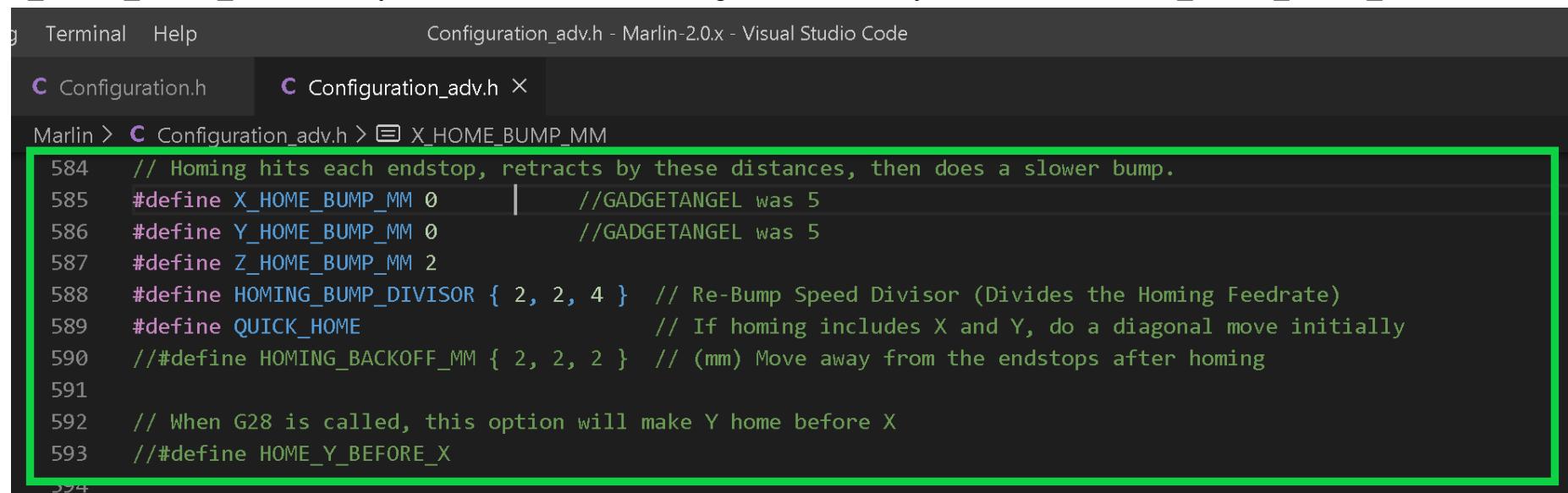


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- 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.



```

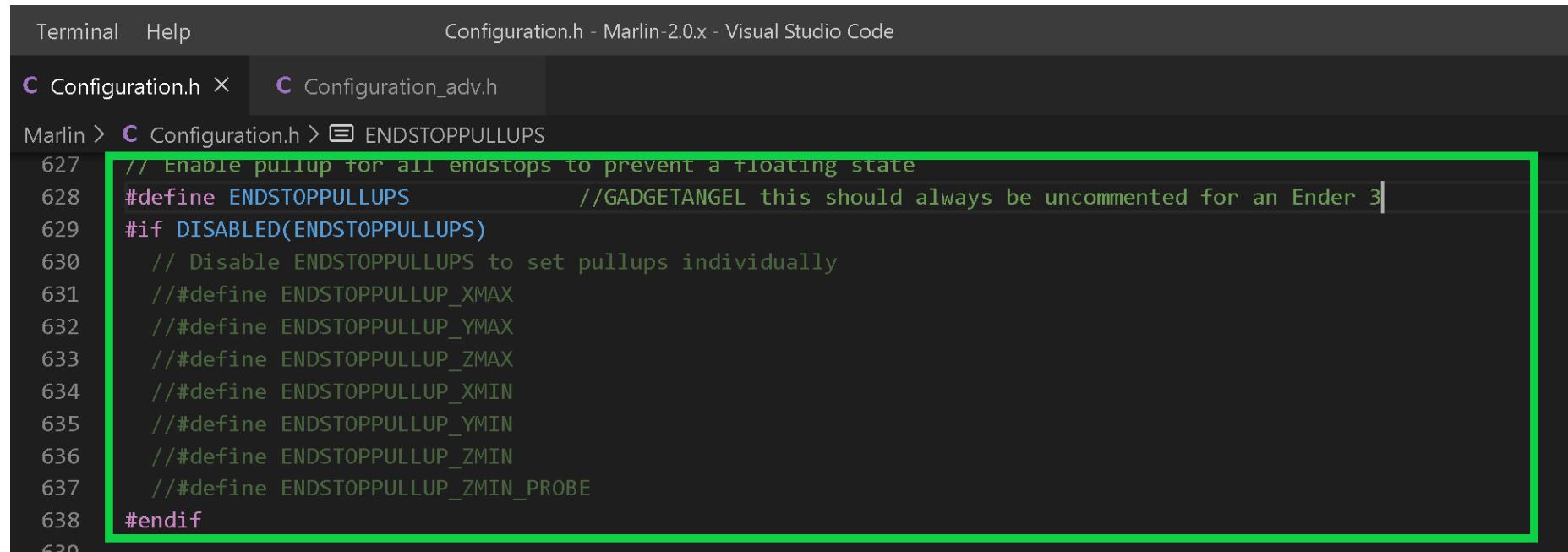
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL 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

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

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



```

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

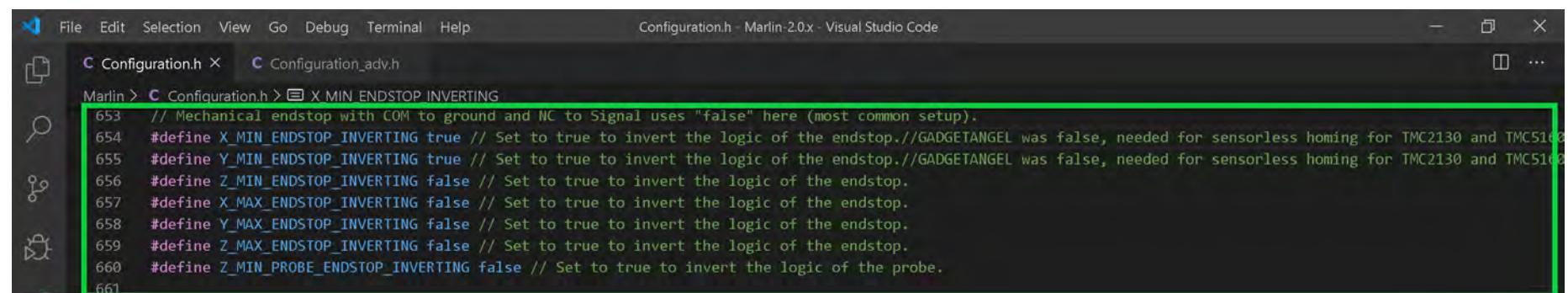
Configuration.h X Configuration_adv.h

Marlin > Configuration.h > ENDSTOPPULLUPS

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL 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

```

- Next to allow sensor-less homing to work (while using the BIQU TMC5161) 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

Configuration.h X Configuration_adv.h

Marlin > Configuration.h > X MIN ENDSTOP INVERTING

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.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
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 probe.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- The end of Marlin setup for BIQU TMC5161 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` (marked as modified), and `Configuration_adv.h`.
- Code Editor:** Displays the `Configuration.h` file with code related to stepper drivers and endstops.
- Terminal:** Shows the build output for `BIGTREE_SKR_PRO`:

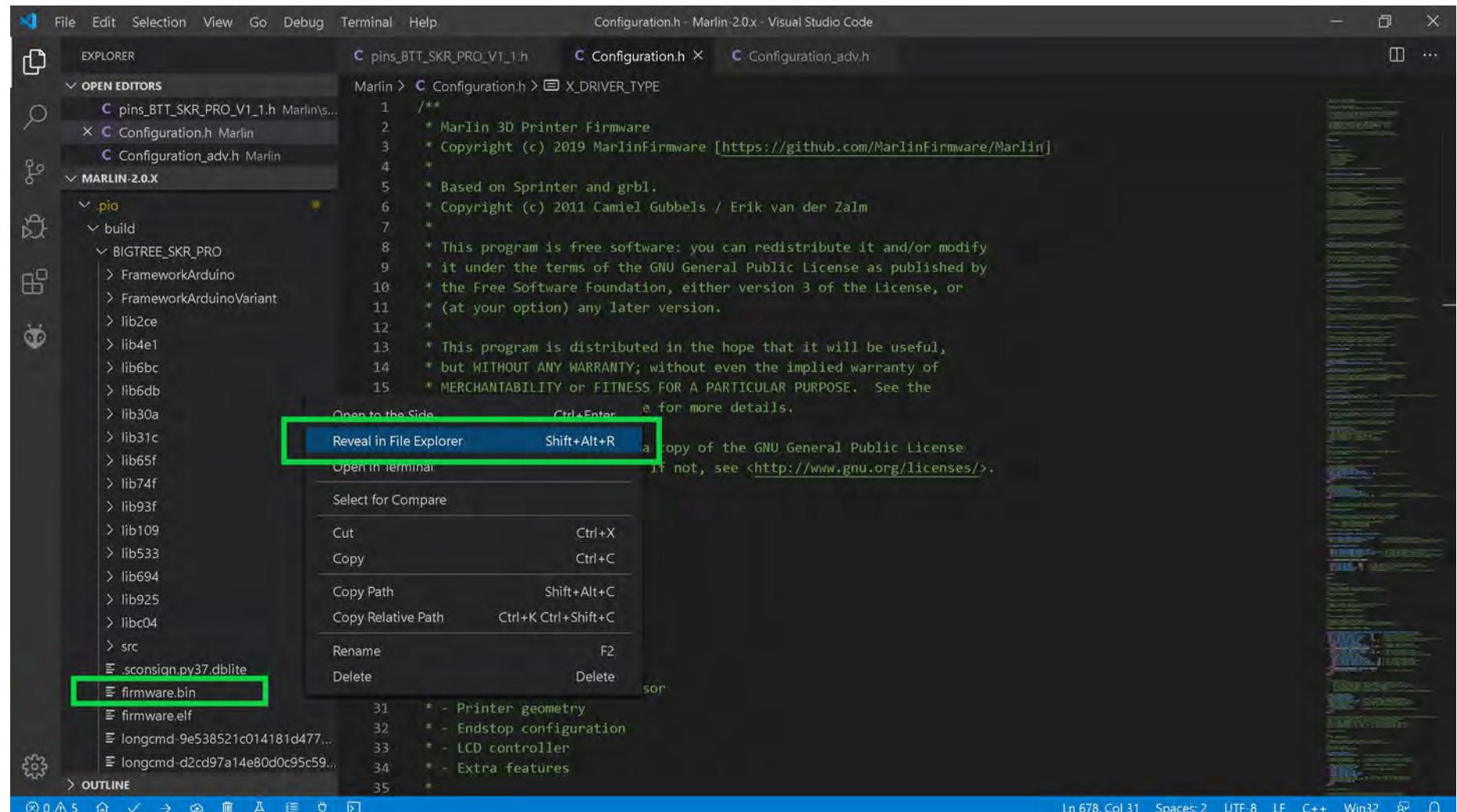
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
-----------------	---------	--------------

 Other builds listed include `BIGTREE_BTT002`, `teensy31`, `teensy35`, `esp32`, `linux_native`, `SAMD51_grandcentral_m4`, `rumba32_f446ve`, `mks_rumba32`, and `include_tree`.
- Bottom Bar:** Shows the status bar with "Ln 678, Col 31" and various icons.
- Toolbar:** Shows icons for file operations, including a yellow box around the "COMPILE" icon (the checkmark icon).

- 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 TMC5161 V1.0 Drivers in SPI Mode

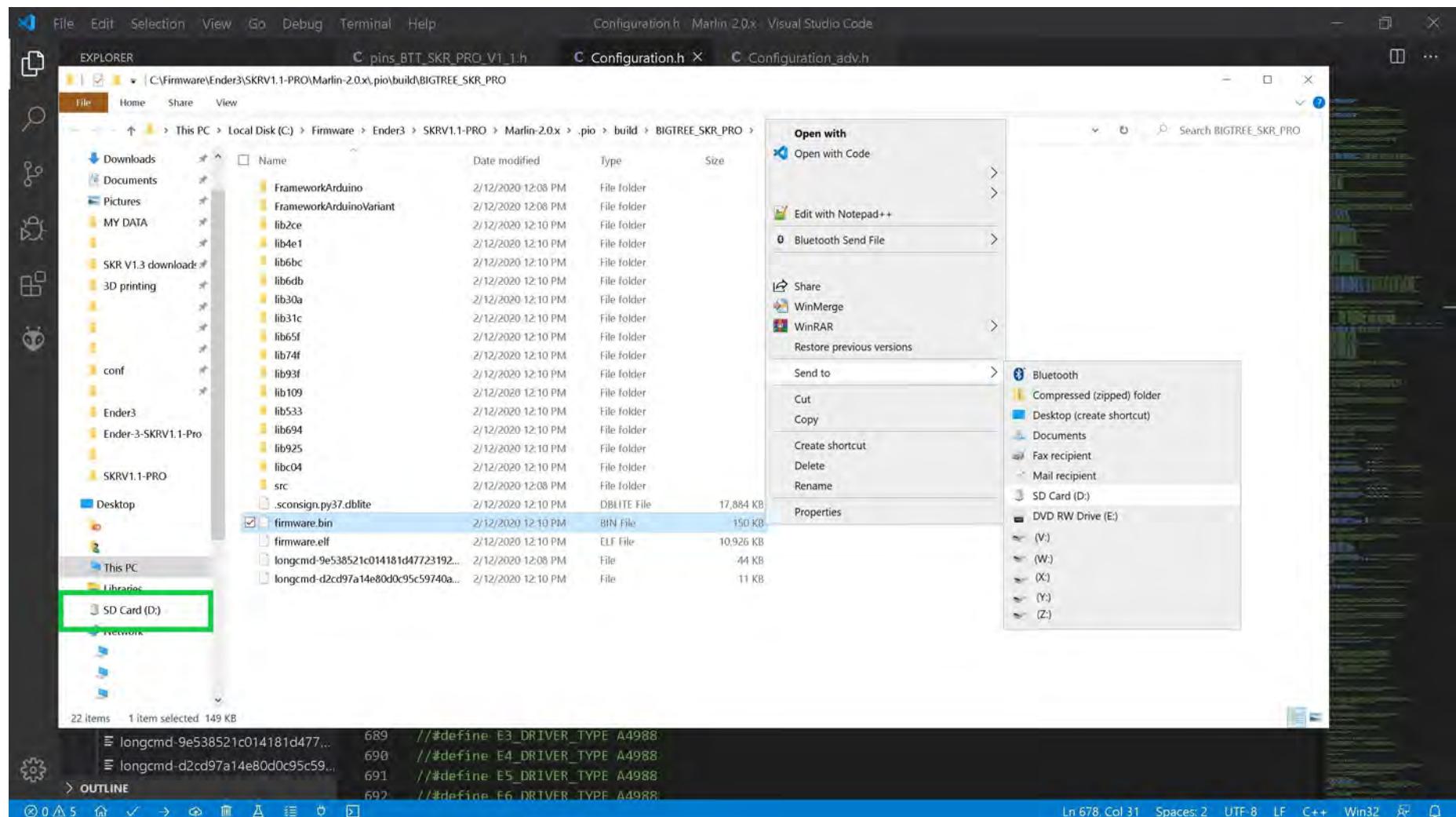
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X/.pio/build/BIGTREE_SKR_PRO/ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer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



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Ensure your micro SD card, which came with your SKR PRO V1.1 board, 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. After the upload the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

How to adjust the V_{ref} 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 " V_{ref} 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 (I_{MAX}) 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 V_{ref}) which corresponds with the current you want to set. Use a ceramic screw driver to adjust the POT. A ceramic screw driver is nonconductive 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

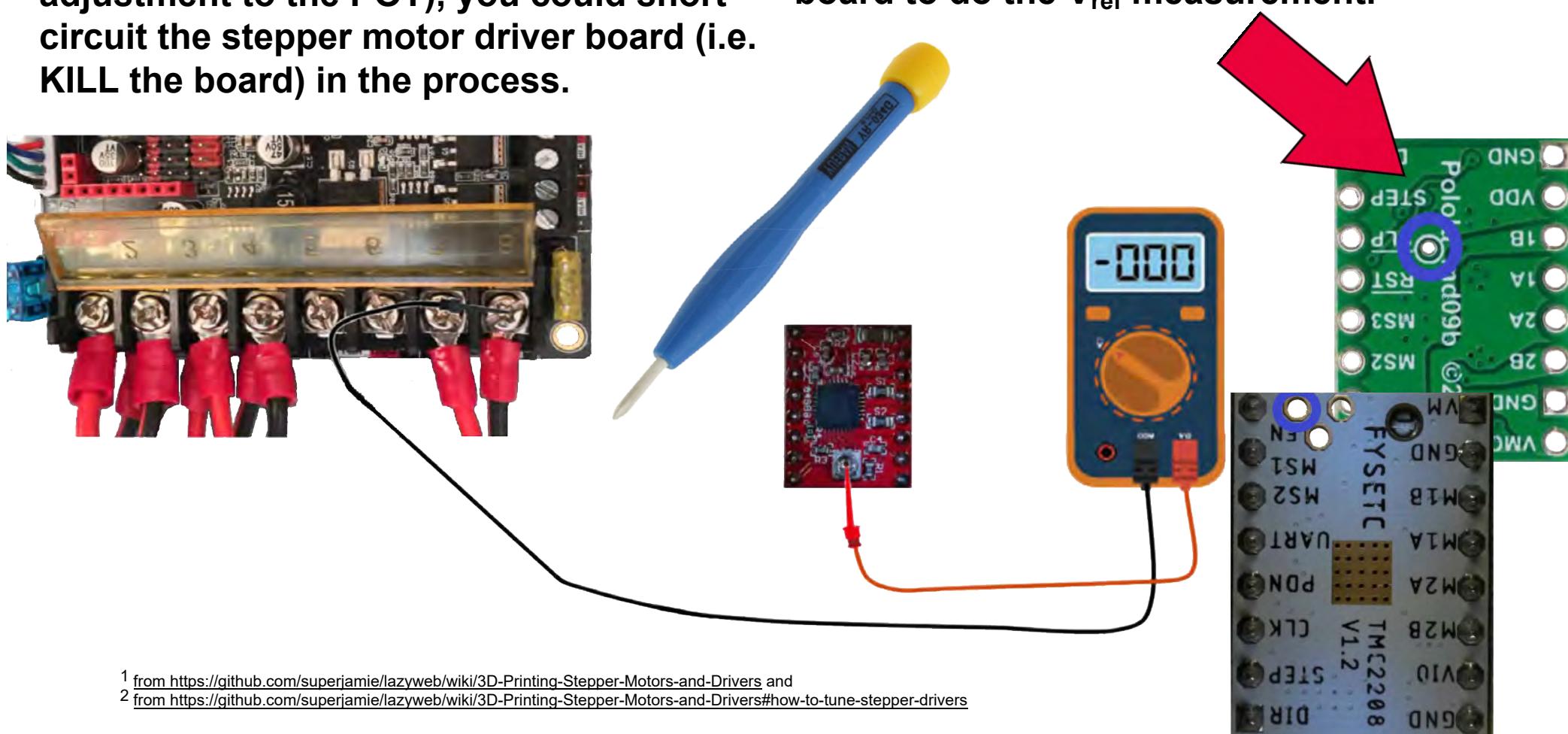
² from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

How to adjust the V_{ref} 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 V_{ref} (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 " V_{ref} Test point" location, shown in **BLUE**. Check bottom or top of your board for a location.

If a " V_{ref} Test point" location is not available, use the potentiometer on the stepper driver board to do the V_{ref} measurement.



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 ") modes?

All the TMC driver chips **EXCEPT TMC5160 and TMC5161** 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 V_{ref} for Non-TMC 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 V_{ref} formula (found on the first page of each different stepper motor driver section of this document and do the V_{ref} calculation.

Remember this V_{ref} calculation is just a suggested starting point. If your stepper motors are running **too hot** you will need to adjust the V_{ref} **downward**. If your stepper motors are **skipping steps** when printing then you will need to adjust your V_{ref} **upwards**. Our goal is to find a low enough V_{ref} where our stepper motors are cool enough without the printer missing any steps. For this example, I will use the A4988 stepper motor driver.

1. So, A4988 V_{ref} formula is $V_{ref} = I_{MAX} * (8 * R_s)$, where $R_s = 0.1\Omega$.
2. I take each of my Axis' "Rated Current" and plug it into that equation to get X-Axis V_{ref} is equal to $(0.840 * 8 * 0.100) = 0.672$ volts or 672mV.
3. Now, take 90% of that for a starting point V_{ref} value of $((0.672 * 0.90) = 0.6048)$ 0.605 volts or 605mV or X-Axis $V_{ref} = 0.605V$. Since X, Y and Z stepper motors have the same "Rated Current" we now have the V_{ref} for X, Y and Z stepper motor drivers. Their value is 0.605 volts.
4. For E (extruder) V_{ref} the equation is $(1.0 * 8 * 0.100) = 0.800$ volts. Now, take 90% of that, for a starting point, V_{ref} value of $((0.800 * 0.90) = 0.720)$ 0.720 volts or 720mV. We now have the V_{ref} for E (extruder stepper motor driver) which is 0.72 volts.
5. We use our multimeter and turn the POT on the top of the stepper motor driver until we see the wanted V_{ref} voltage displayed.

APPENDIX B

How to Calculate V_{ref} for TMC Stepper Motor Drivers

My machine is an Ender 3. The X, Y, and Z stepper motors have a "Rated Current" of 0.84 Amps, while E (extruder) stepper motor "Rated Current" is 1 Amp.

We use the I_{MAX} formula and use Algebra to find the V_{ref} formula. The I_{MAX} formula for each TMC Driver is listed on the following pages. But we will use TMC2100 drivers so we will use the below equation:

TMC2100 Stand-alone Mode, (with $R_s=110m\Omega$):

$$I_{RMS} = ((V_{ref} / 2.5) * (1 / 1.41) * ((320mV / (R_s + 20m\Omega))))$$

Since $I_{MAX}=(I_{RMS} * 1.41)$ is a known value then the above equation can be written as follows:

$$(I_{MAX} * (1 / 1.41)) = ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / (110+20))))$$

Since $(1 / 1.41)$ is on both sides of the Algebra equation they cancel each other out leaving the equation as follows:

$$I_{MAX} = (V_{ref} / 2.5) * (2.46 \approx 2 .5),$$

Therefore $I_{MAX} = V_{ref}$, and $V_{ref} = I_{MAX}$.

We use 50% to 90% of V_{ref} (i.e. I_{MAX}) to set the current limit for TMC stepper motor driver. To take 90% we do the following:

Since $V_{ref}=I_{MAX}$ we will use I_{MAX} instead of V_{ref} .

90% of V_{ref} for Ender 3's X, Y or Z $= (I_{MAX} * 0.90) = (0.84 * 0.90) = 0.756$ or 756mA. Since $I_{MAX}=V_{ref}$, then it's also equal to 756mV.

Remember this V_{ref} calculation is just a suggested starting point. If your stepper motors are running **too hot** you will need to adjust the V_{ref} **downward**. If your stepper motors are **skipping steps** when printing then you will need to adjust your V_{ref} **upwards**. Our goal is to find a low enough V_{ref} where our stepper motors are cool enough without the printer missing any steps.

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**1. TMC2100 with $R_s=0.110\Omega$ (110mΩ) :**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320mV / (R_s + 20m\Omega)))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320/110+20))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (2.46)) \\ &= (1/1.41) * V_{ref} * 0.99 \\ &= (1/1.41) * V_{ref} * 1 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

2. TMC2130 with $R_s=0.110\Omega$ (110mΩ) :

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325/110+20))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (2.5)) \\ &= (1/1.41) * V_{ref} \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

3. TMC2208 with $R_s=0.110\Omega$ (110mΩ) for Stand-alone Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 30m\Omega)))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325/110+30))) \\ &= ((1/1.41) * (V_{ref}/2.50) * (2.32143)) \\ &= (1/1.41) * V_{ref} * 0.928572 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref} * 0.928572$$

$$I_{MAX} = V_{ref} * 0.9286$$

$$V_{ref} = I_{MAX} * 1.0769$$

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**4. TMC2208 with $R_s=0.110\Omega$ (110mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)))) \\ &= ((1 / 1.41) * (V_{ref}/2.5) * ((325/(110+20)))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (2.5)) \\ &= (1/1.41) * V_{ref} * 1 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

5. TMC2209 with $R_s=0.110\Omega$ (110mΩ) for Stand-alone Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)))) \\ &= ((1/1.41) * (V_{ref}/2.5) * ((325/(110+20)))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (2.5)) \\ &= (1/1.41) * V_{ref} * 1 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**6. TMC2209 with $R_s=0.110\Omega$ (110mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325/110+20))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (2.5)) \\ &= (1/1.41) * V_{ref} * 1 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

7. TMC2225 with $R_s=0.150\Omega$ (150mΩ) for UART Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1/1.41) * (V_{ref}/2.5) * ((325/(150+20))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (1.9118)) \\ &= (1/1.41) * V_{ref} * 0.7647 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * V_{ref} * 0.7647$$

$$I_{MAX} = V_{ref} * 0.7647$$

$$V_{ref} = I_{MAX} * 1.3077$$

8. TMC5160 with $R_s=0.075\Omega$ (75mΩ) for SPI Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (325mV / R_s)) \\ &= ((1/1.41) * (325/75)) \\ &= (1/1.41) * 4.33 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$ Therefore,

$$I_{MAX} * (1/1.41) = (1/1.41) * 4.33$$

$$I_{MAX} = 4.333 \text{ Amps}$$

You will use **50% to 90%** of I_{MAX} ($4.333 * .50$ or $4.333 * .90$) which is **2.1665 Amps (2167 mA)** to **3.8997 Amps (3900 mA)** for the Marlin Firmware.

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**9. TMC5161 with $R_s=0.062\Omega$ (62mΩ) for SPI Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (325mV / R_s)) \\ &= ((1/1.41) * (325/62)) \\ &= (1/1.41) * 5.24194 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$ Therefore,

$$I_{MAX}*(1/1.41)=(1/1.41)*5.24194$$

$$I_{MAX}=5.24194 \text{ or } 5.2419$$

You will use **50% to 90%** of I_{MAX} ($5.2419*.50$ or $5.2419*.90$) which is **2.621 Amps (2621 mA)** to **4.7177 Amps (4718 mA)** for the Marlin Firmware.

10. TMC2225 with $R_s=0.150\Omega$ (150mΩ) for Stand-alone Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 30m\Omega)))) \\ &= ((1/1.41) * (V_{ref}/2.5) * ((325/(150+30)))) \\ &= ((1/1.41) * (V_{ref}/2.5) * (1.8056)) \\ &= (1/1.41) * V_{ref} * 0.7222 \end{aligned}$$

Since $I_{RMS}=I_{MAX}*(1/1.41)$

Therefore,

$$I_{MAX}*(1/1.41)=(1/1.41)*V_{ref}*0.7222$$

$$I_{MAX}=V_{ref}*0.7222$$

$$V_{ref}=I_{MAX}*1.3846$$

APPENDIX C

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

Please refer to the following documents:

- [BIGTREETECH SKR-PRO-V1.1 User Manual.pdf](#)
- [BIGTREETECH SKR-PRO-V1.1 Guide.pdf](#)

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](https://marlinfw.org/meta/download/) 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

Description	Version	Download	Configurations
Latest release Supports AVR and ARM Arduino and PlatformIO	2.0.3	2.0.x.zip	View / Download
Previous release Supports AVR Arduino and PlatformIO	1.1.9	1.1.x.zip	View / Download
Older release Supports Arduino 1.6.8 and up	1.0.2-2	1.0.x.zip	

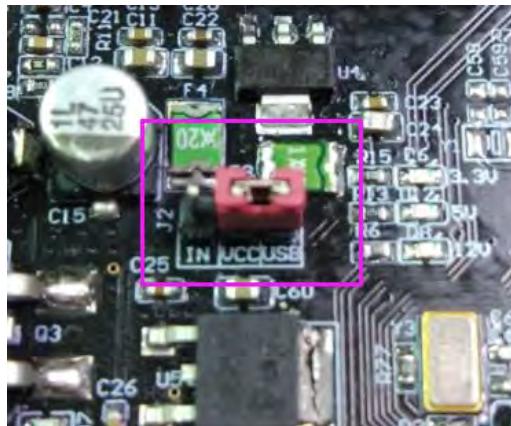
Description	Version	Download	Configurations
Patches to latest 2.0 Marlin 2.0 with bug fixes Supports AVR and ARM Arduino and PlatformIO	bugfix-2.0.x	bugfix-2.0.x.zip	View / Download
Proceed with Caution! Marlin 2.1 development Supports AVR and ARM Arduino and PlatformIO	dev-2.1.x	dev-2.1.x.zip	View / Download

- Go to the next page.

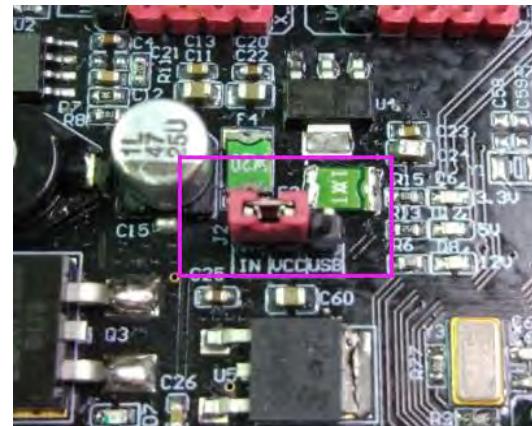
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

To ensure that any changes I make to my Marlin Firmware are permanent, I do a couple of things. I will send the G-code M503 by using the **Pronterface software** **BEFORE I change the Marlin Firmware** so I can **write down all the calibrations that I have previously set.**

I ensure that the "Power selection", as defined in "SKR PRO V1.1 user manual v1.1.pdf" is set to the correct power input source ②. Even though you can use the USB as the source of power, I prefer to use the 12V/24V DC power source, as shown below.



① USB power



② 12/24V power

Marlin2.0 Firmware Update Method:

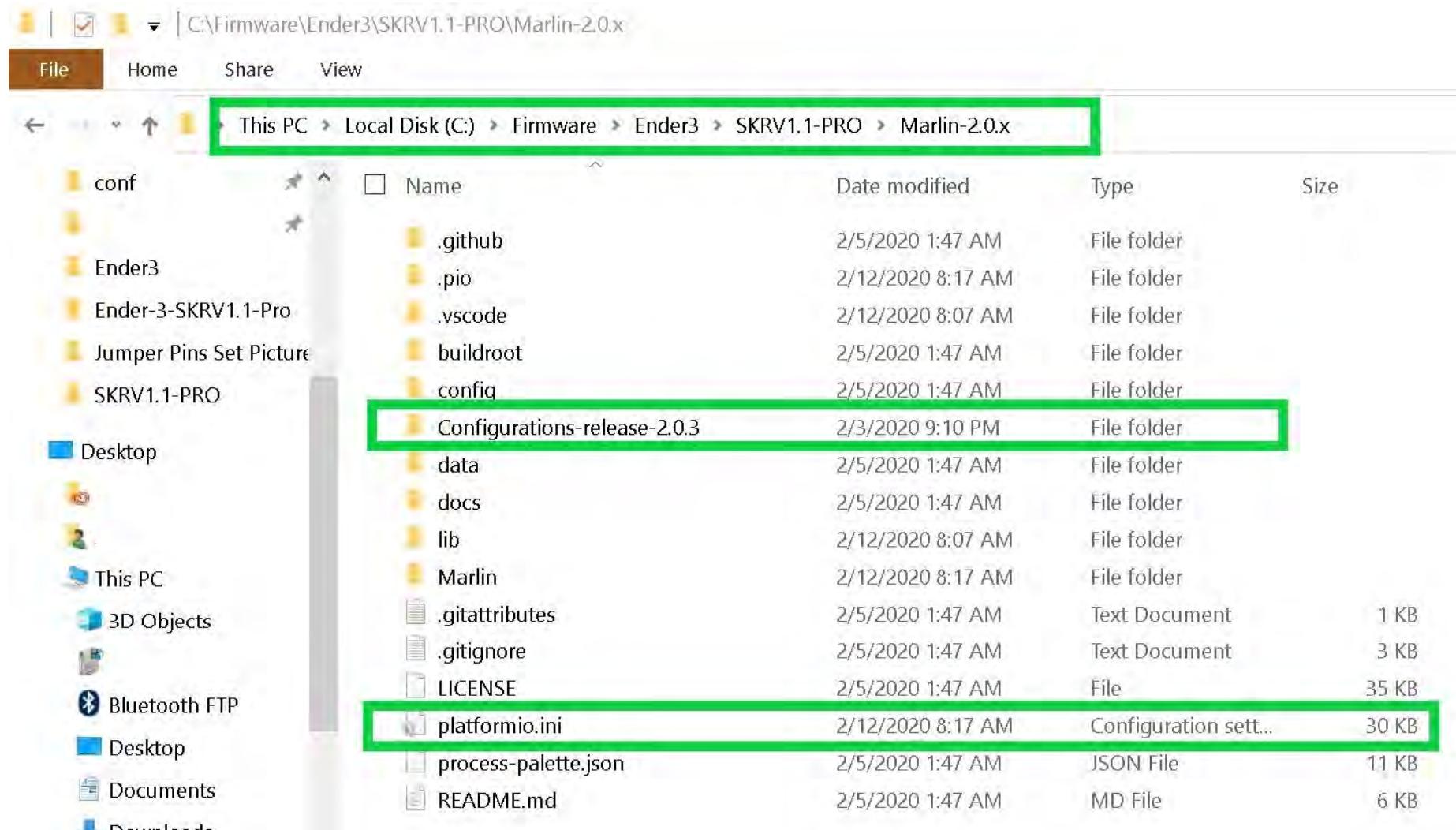
After downloading the files, use Visual Studio Code (VScode) to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the micro SD card into the board. Reboot the board; wait for about 10 seconds before doing anything else with the board. I then use the **Pronterface software**, to send the following "G-code" commands:

- M502
- M500
- M504

M502 reset all configurable settings to their factory defaults. When you follow a M502 by a M500 the M500 will also reset settings in EEPROM. The M504 command will validate the contents of the EEPROM to ensure that the EEPROM settings have been changed to the factory defaults. If the reset does not show the correct settings, find your compiled "firmware.bin" file and copy it again to the SD card, then plug the micro SD card into the SKR PRO board. Reboot the board, wait for about 10 seconds and check the settings again.

After uploading new firmware, you will need to **calibrate your 3D printer again. Please see the following document for instructions on how to calibrate your 3D printer: <https://drive.google.com/open?id=19HVpv2jNMkPamlhMGiEaGEmsnmzsoKVe>**

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers



- Go to the next page.

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!

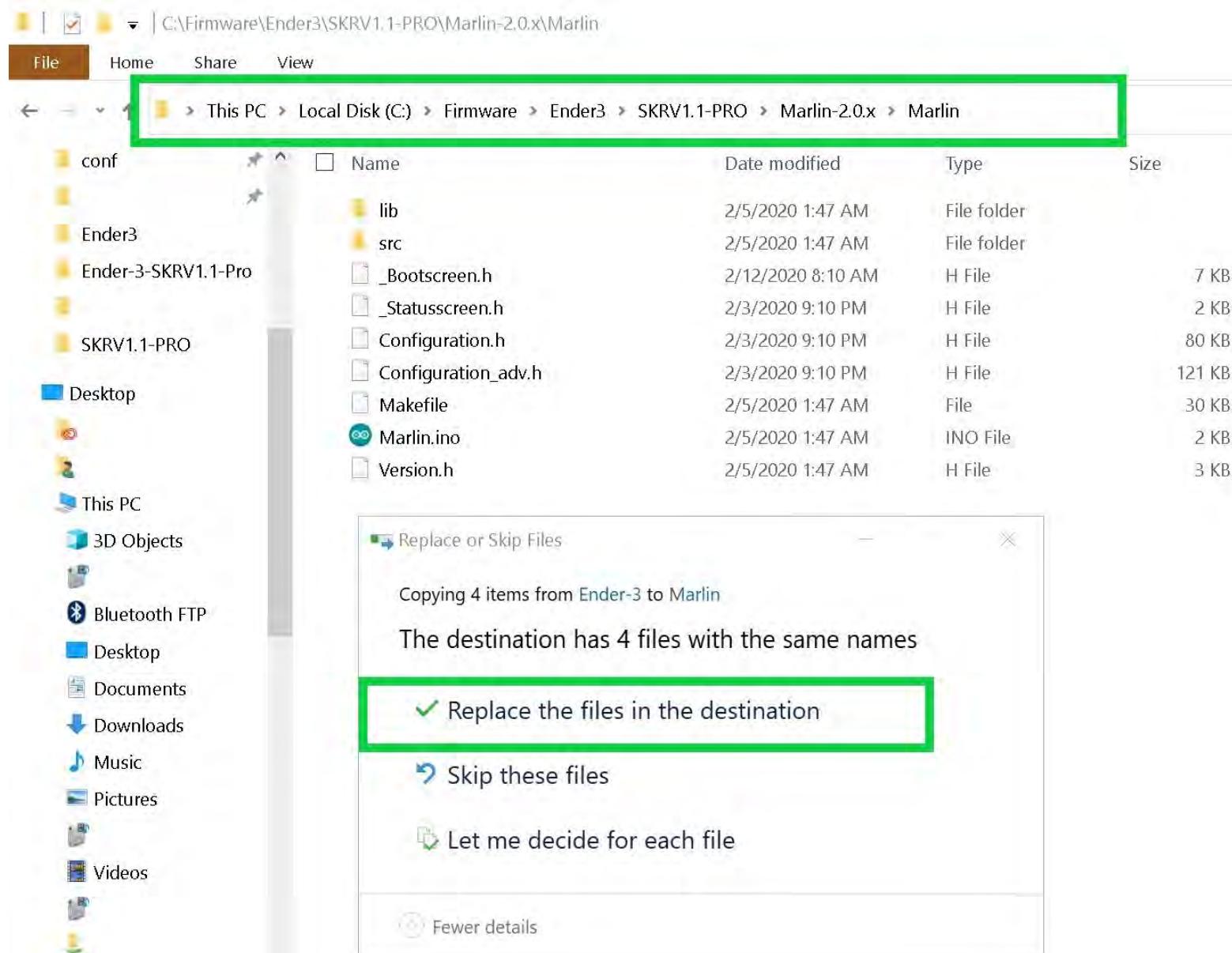
- Copy the below default configuration files (for me Ender 3 files, for you it could be another printer) to the directory where Marlin's Configuration.h and Configuration_adv.h reside. See picture below for which files you will copy. See the next page to see where to place the files.



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

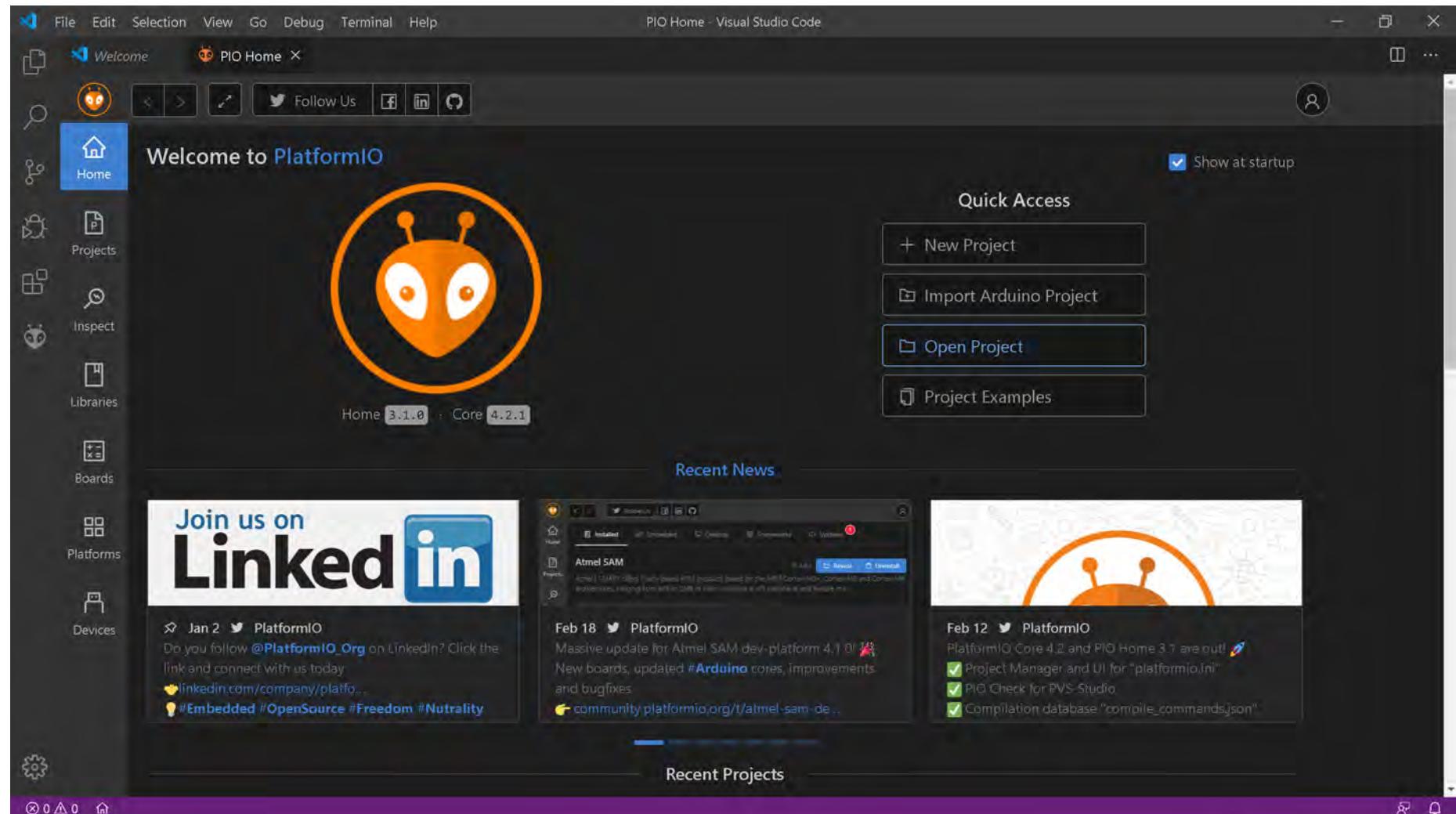
- Copy TO THIS directory so the above files reside in the same directory as Configuration.h and Configuration_adv.h. When prompted allow the files to be overwritten!



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

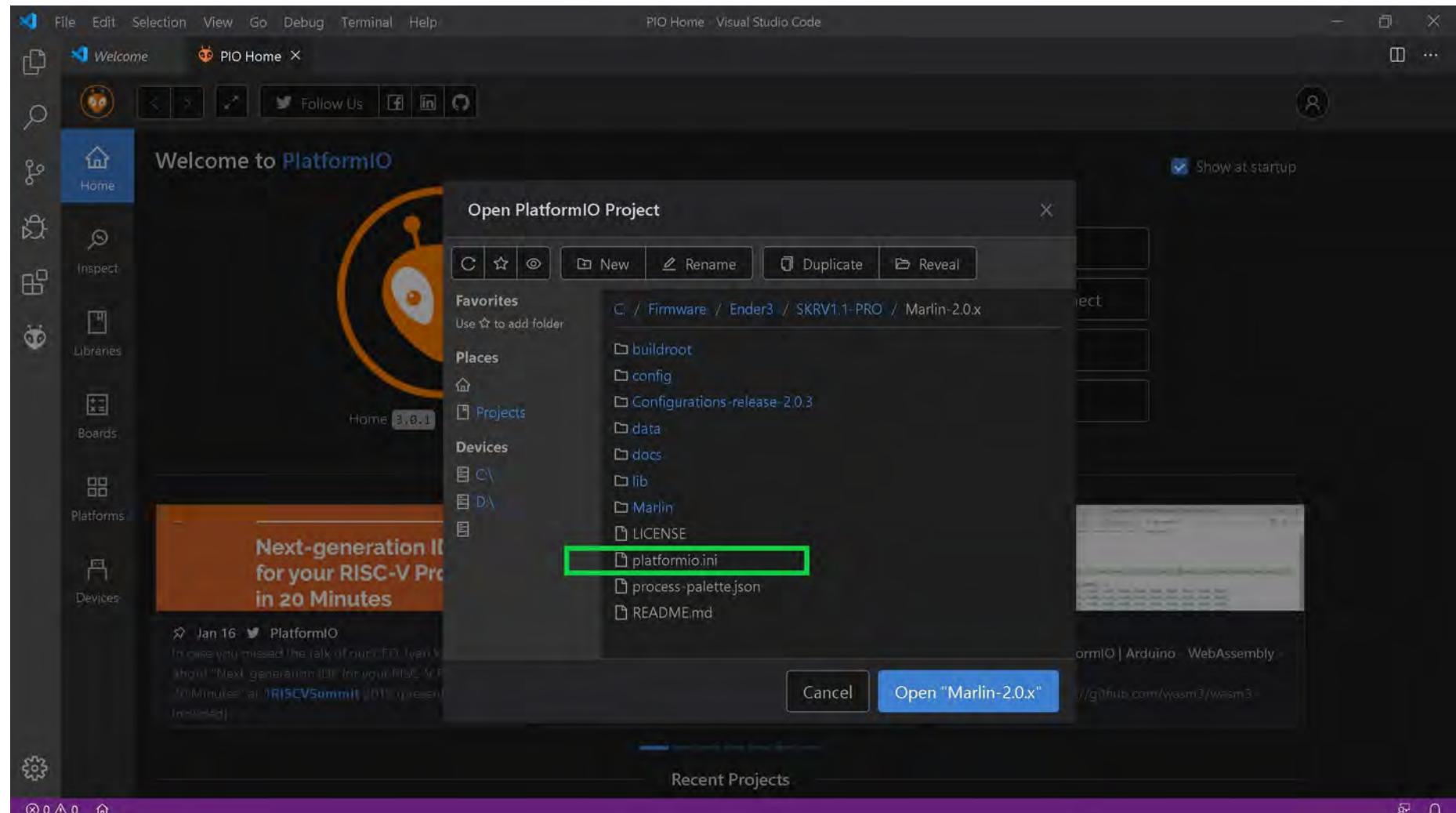
- Open VS code (see picture below) and then select "Open Project".



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

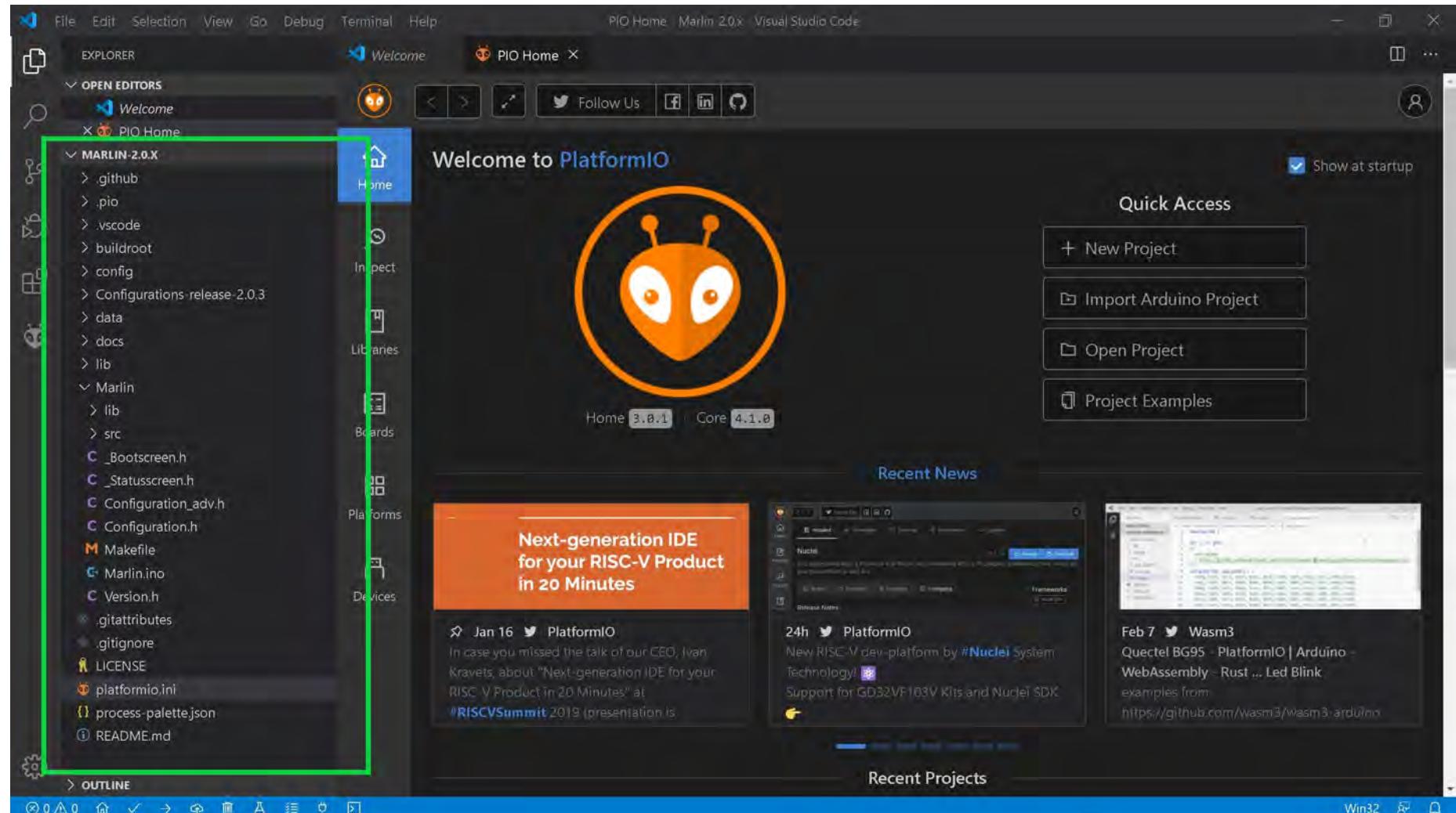
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

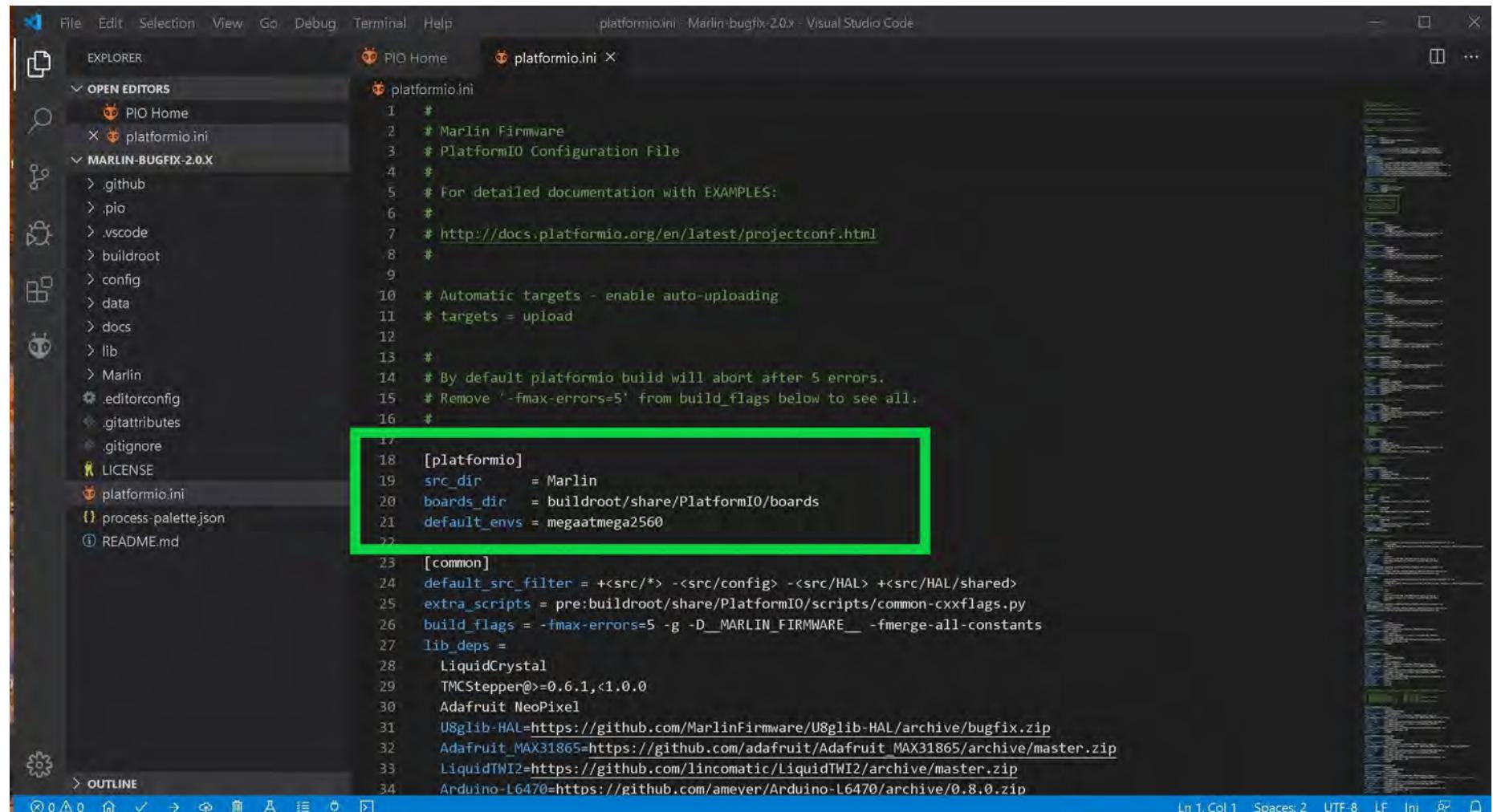
- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- What you will see when "platformio.ini" is opened up in the VS code editor window



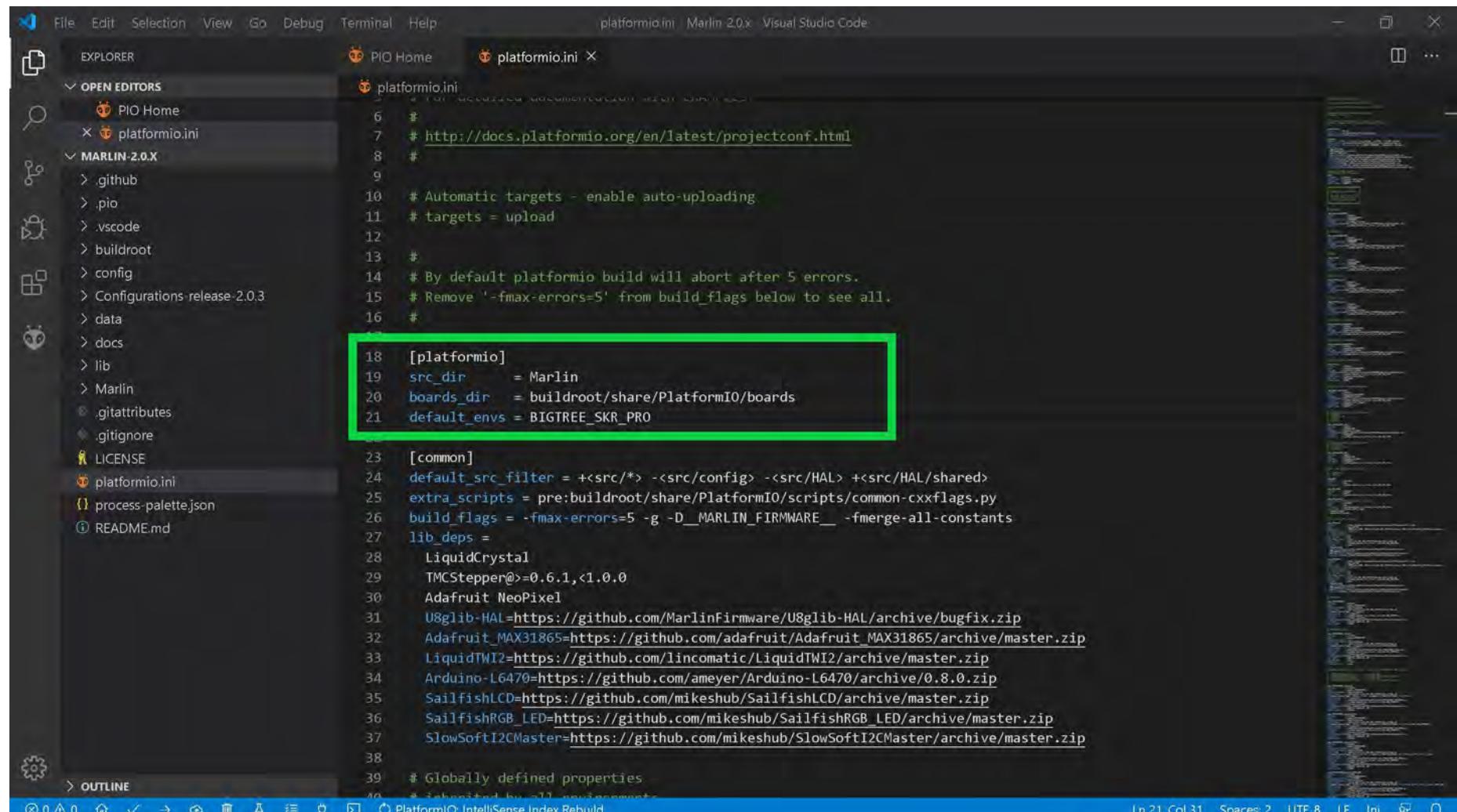
```
platformio.ini - Marlin-bugfix-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help
EXPLORER PIO Home platformio.ini
OPEN EDITORS
  PIO Home
  platformio.ini
MARLIN-BUGFIX-2.0.X
  .github
  .pio
  .vscode
  buildroot
  config
  data
  docs
  lib
  Marlin
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
[platformio]
src_dir      = Marlin
boards_dir   = buildroot/share/PlatformIO/boards
default_envs = megaatmega2560
[common]
default_src_filter = +<src/*> -<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
  Adafruit_MAX31865=https://github.com/adafruit/Adafruit_MAX31865/archive/master.zip
  LiquidTWI2=https://github.com/lincomatic/LiquidTWI2/archive/master.zip
  Arduino-L6470=https://github.com/amever/Arduino-L6470/archive/0.8.0.zip
Ln 1, Col 1  Spaces: 2  UTF-8  LF  Ini  R  B  L  O  N  E
```

- Go to the next page.

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**



```

File Edit Selection View Go Debug Terminal Help
platformio.ini Merlin 2.0.x Visual Studio Code

EXPLORER PIO Home platformio.ini
OPEN EDITORS
  PIO Home
  platformio.ini
MARLIN-2.0.X
  .github
  .pio
  .vscode
  buildroot
  config
  Configurations-release-2.0.3
  data
  docs
  lib
  Marlin
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md

[platformio]
src_dir      = Marlin
boards_dir   = buildroot/share/PlatformIO/boards
default_envs = BIGTREE_SKR_PRO

[common]
default_src_filter = +<src/*> -<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
  Adafruit_MAX31865=https://github.com/adafruit/Adafruit_MAX31865/archive/master.zip
  LiquidTWI2=https://github.com/lincomatic/LiquidTWI2/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

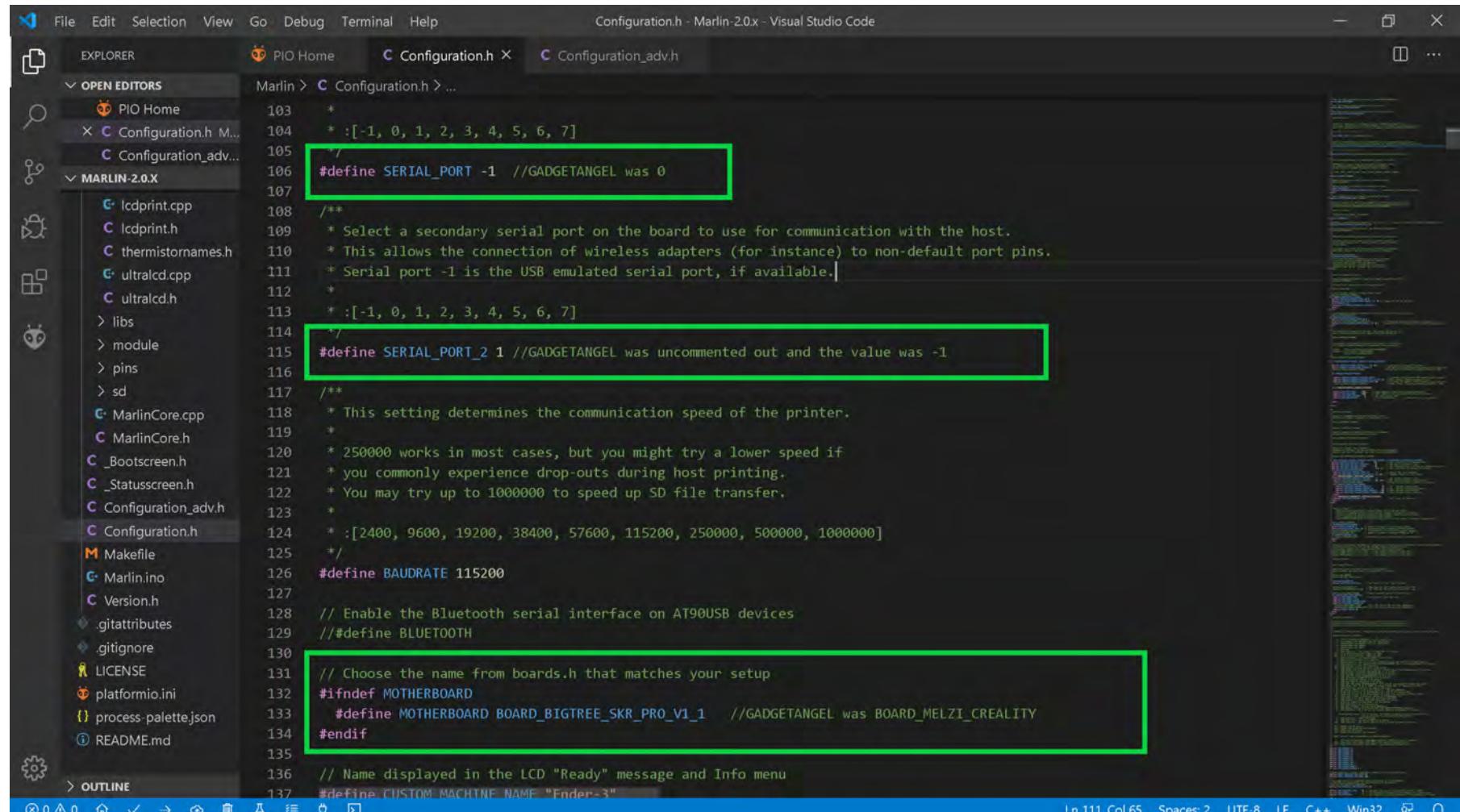
```

Ln 21, Col 31 Spaces: 2 UTF 8 LF Inl

- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Go to the Configuration.h file and change the following three items, as seen in the **3 GREEN** boxes below.



```

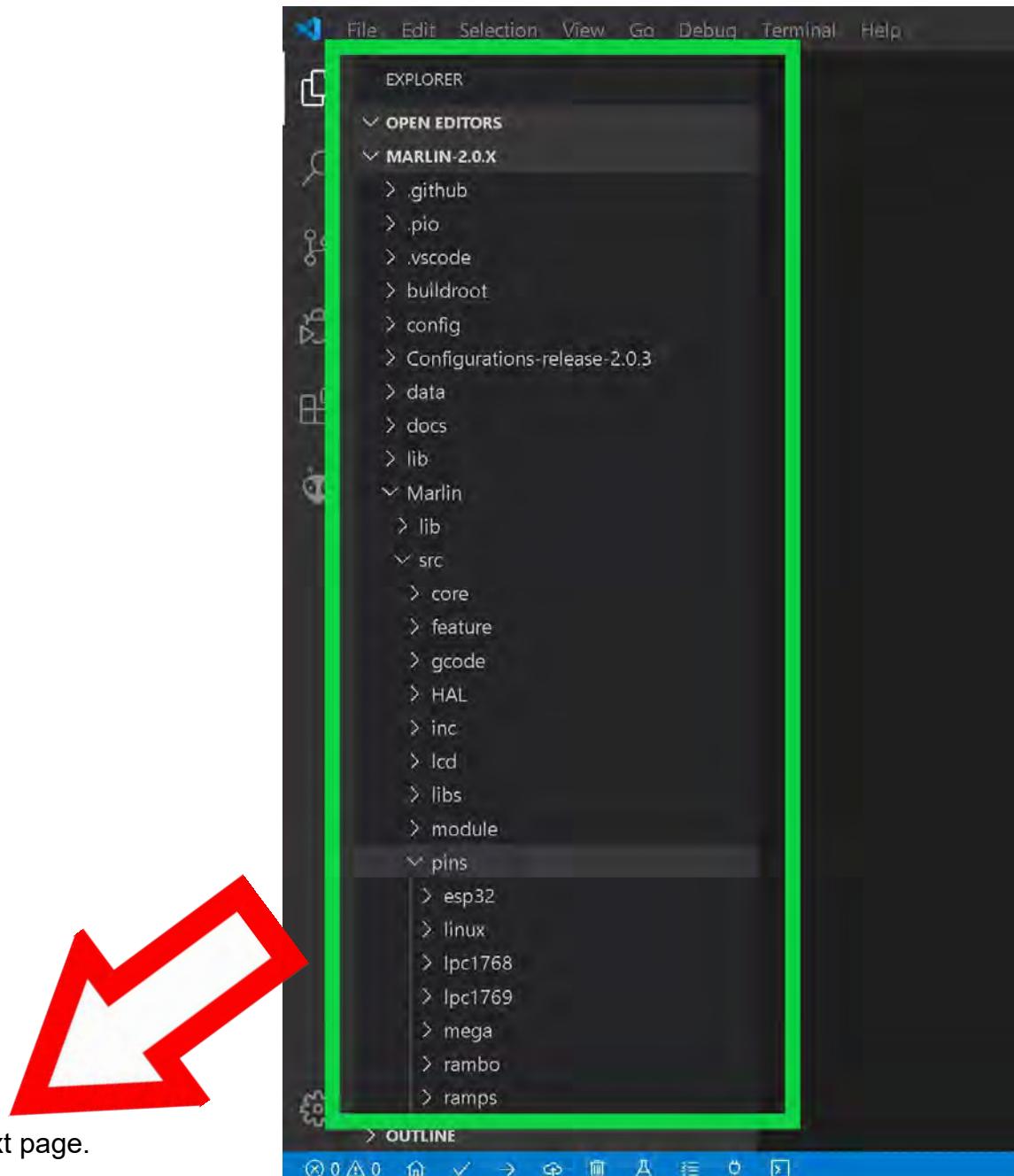
103  /*
104  * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
105 */
106 #define SERIAL_PORT -1 //GADGETANGEL was 0
107
108 /**
109 * Select a secondary serial port on the board to use for communication with the host.
110 * This allows the connection of wireless adapters (for instance) to non-default port pins.
111 * Serial port -1 is the USB emulated serial port, if available.
112 *
113 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
114 */
115 #define SERIAL_PORT_2 1 //GADGETANGEL was uncommented out and the value was -1
116
117 /**
118 * This setting determines the communication speed of the printer.
119 *
120 * 250000 works in most cases, but you might try a lower speed if
121 * you commonly experience drop-outs during host printing.
122 * You may try up to 1000000 to speed up SD file transfer.
123 *
124 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
125 */
126 #define BAUDRATE 115200
127
128 // Enable the Bluetooth serial interface on AT90USB devices
129 //#define BLUETOOTH
130
131 // Choose the name from boards.h that matches your setup
132 #ifndef MOTHERBOARD
133 #define MOTHERBOARD BOARD_BIGTREE_SKR_PRO_V1_1 //GADGETANGEL was BOARD_MELZI_CREALITY
134 #endif
135
136 // Name displayed in the LCD "Ready" message and Info menu
137 #define CUSTOM_MACHINE_NAME "Ender-3"

```

- You can set "BAUDRATE" to "115200" or "250000 ". Either setting will work but I have found that "115200" option works with any LCD that you choose to use.
- Go to the next page.

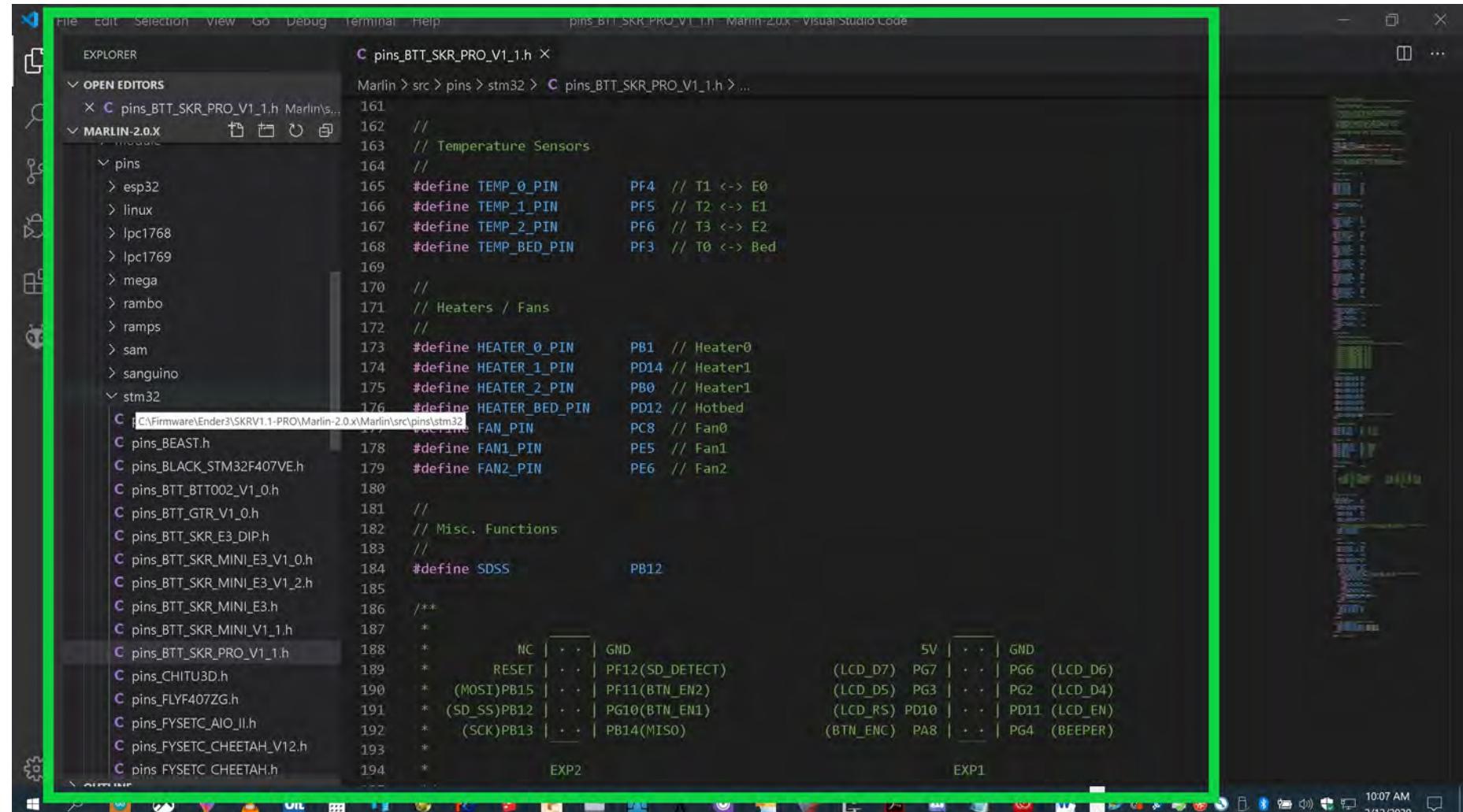
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- 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 file, pins_BTT_SKR_PRO_V1_1.h, by double clicking on it.



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers



The screenshot shows the Visual Studio Code interface with the file `pins_BTT_SKR_PRO_V1_1.h` open. The code is for the Marlin 2.0.x firmware. The `pins` section contains definitions for various pins, including temperature sensors, heaters, fans, and miscellaneous functions. The `stm32` section is expanded, showing pin assignments for the STM32 board. The pins are grouped into two columns: analog pins (A0-A5) and digital pins (PB0-PB12). The digital pins are mapped to specific functions like `TEMP_0_PIN`, `HEATER_0_PIN`, and `FAN_PIN`. The code also includes comments for required jumpers between numbered pins.

```

File Edit Selection View Go Debug Terminal Help
pins_BTT_SKR_PRO_V1_1.h Marlin-2.0.x - Visual Studio Code

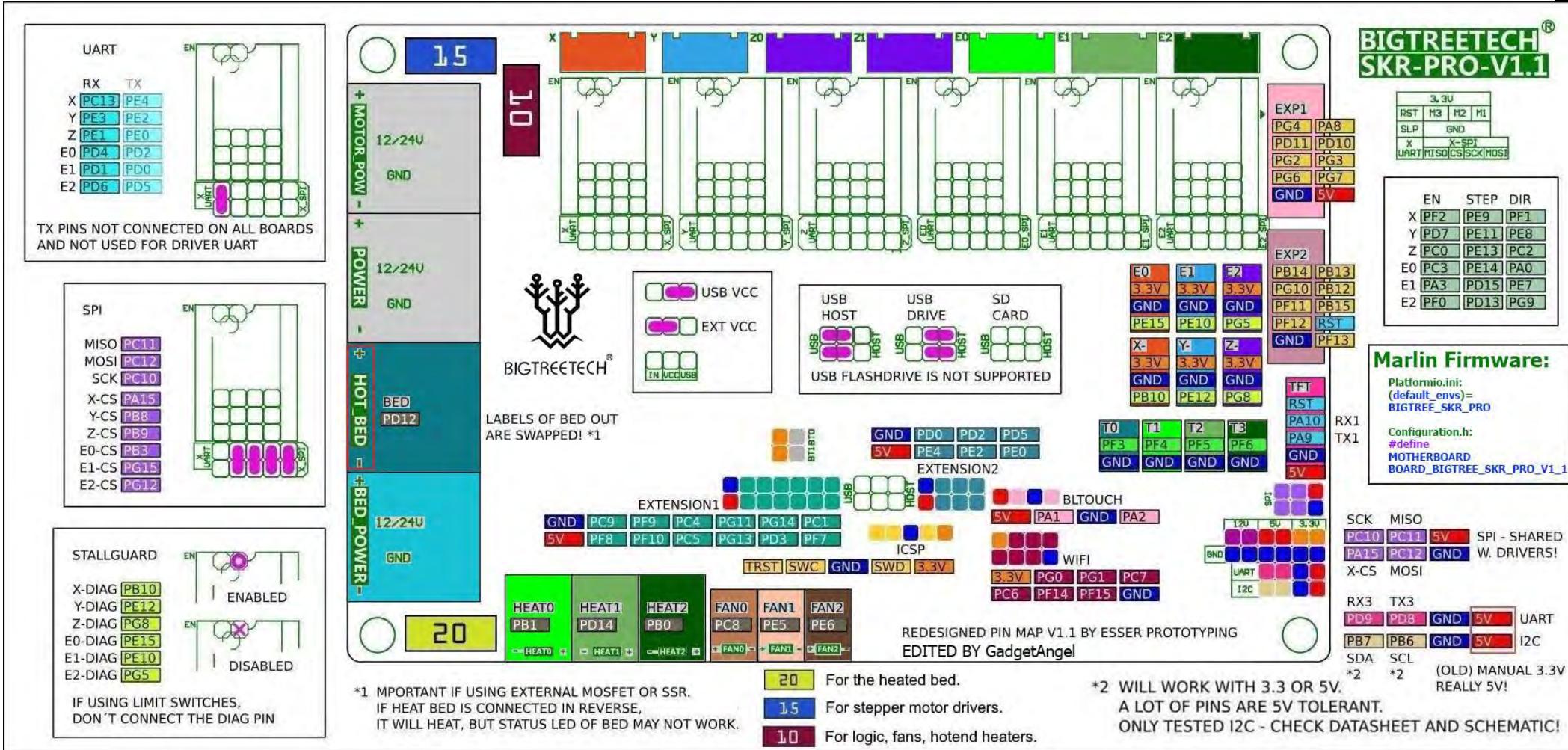
EXPLORER pins_BTT_SKR_PRO_V1_1.h ×
OPEN EDITORS Marlin > src > pins > stm32 > pins_BTT_SKR_PRO_V1_1.h ...
MARLIN-2.0.X
pins
  esp32
  linux
  lpc1768
  lpc1769
  mega
  rambo
  ramps
  sam
  sanguino
  stm32
    pins_BTT_SKR_PRO_V1_1.h
    pins_BEAST.h
    pins_BLACK_STM32F407VE.h
    pins_BTT_BTT002_V1_0.h
    pins_BTT_GTR_V1_0.h
    pins_BTT_SKR_E3_DIP.h
    pins_BTT_SKR_MINI_E3_V1_0.h
    pins_BTT_SKR_MINI_E3_V1_2.h
    pins_BTT_SKR_MINI_E3.h
    pins_BTT_SKR_MINI_V1_1.h
    pins_BTT_SKR_PRO_V1_1.h
    pins_CHITU3D.h
    pins_FLYF407ZG.h
    pins_FYSETC_AIO_ll.h
    pins_FYSETC_CHEETAH_V12.h
    pins_FYSETC_CHEETAH.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
  177 #define FAN_PIN PC8 // Fan0
  178 #define FAN1_PIN PE5 // Fan1
  179 #define FAN2_PIN PE6 // Fan2
  180 // 
  181 // 
  182 // Misc. Functions
  183 // 
  184 #define SDSS PB12
  185 // 
  186 /**
  187 *      NC | + | GND
  188 *      RESET | + | PF12(SD_DETECT)      5V | + | GND
  189 *      (MOSI)PB15 | + | PF11(BTN_EN2)  (LCD_D7) PG7 | + | PG6 (LCD_D6)
  190 *      (SD_SS)PB12 | + | PG10(BTN_EN1) (LCD_D5) PG3 | + | PG2 (LCD_D4)
  191 *      (SCK)PB13 | + | PB14(MISO)   (LCD_RS) PD10 | + | PD11 (LCD_EN)
  192 *      EXP2
  193 *      EXP1
  194 *      EXP2
  195 *      EXP1
  196 // 
  197 // 
  198 // 
  199 // 
  200 // 
  201 // 
  202 // 
  203 // 
  204 // 
  205 // 
  206 // 
  207 // 
  208 // 
  209 // 
  210 // 
  211 // 
  212 // 
  213 // 
  214 // 
  215 // 
  216 // 
  217 // 
  218 // 
  219 // 
  220 // 
  221 // 
  222 // 
  223 // 
  224 // 
  225 // 
  226 // 
  227 // 
  228 // 
  229 // 
  230 // 
  231 // 
  232 // 
  233 // 
  234 // 
  235 // 
  236 // 
  237 // 
  238 // 
  239 // 
  240 // 
  241 // 
  242 // 
  243 // 
  244 // 
  245 // 
  246 // 
  247 // 
  248 // 
  249 // 
  250 // 
  251 // 
  252 // 
  253 // 
  254 // 
  255 // 
  256 // 
  257 // 
  258 // 
  259 // 
  260 // 
  261 // 
  262 // 
  263 // 
  264 // 
  265 // 
  266 // 
  267 // 
  268 // 
  269 // 
  270 // 
  271 // 
  272 // 
  273 // 
  274 // 
  275 // 
  276 // 
  277 // 
  278 // 
  279 // 
  280 // 
  281 // 
  282 // 
  283 // 
  284 // 
  285 // 
  286 // 
  287 // 
  288 // 
  289 // 
  290 // 
  291 // 
  292 // 
  293 // 
  294 // 
  295 // 
  296 // 
  297 // 
  298 // 
  299 // 
  300 // 
  301 // 
  302 // 
  303 // 
  304 // 
  305 // 
  306 // 
  307 // 
  308 // 
  309 // 
  310 // 
  311 // 
  312 // 
  313 // 
  314 // 
  315 // 
  316 // 
  317 // 
  318 // 
  319 // 
  320 // 
  321 // 
  322 // 
  323 // 
  324 // 
  325 // 
  326 // 
  327 // 
  328 // 
  329 // 
  330 // 
  331 // 
  332 // 
  333 // 
  334 // 
  335 // 
  336 // 
  337 // 
  338 // 
  339 // 
  340 // 
  341 // 
  342 // 
  343 // 
  344 // 
  345 // 
  346 // 
  347 // 
  348 // 
  349 // 
  350 // 
  351 // 
  352 // 
  353 // 
  354 // 
  355 // 
  356 // 
  357 // 
  358 // 
  359 // 
  360 // 
  361 // 
  362 // 
  363 // 
  364 // 
  365 // 
  366 // 
  367 // 
  368 // 
  369 // 
  370 // 
  371 // 
  372 // 
  373 // 
  374 // 
  375 // 
  376 // 
  377 // 
  378 // 
  379 // 
  380 // 
  381 // 
  382 // 
  383 // 
  384 // 
  385 // 
  386 // 
  387 // 
  388 // 
  389 // 
  390 // 
  391 // 
  392 // 
  393 // 
  394 // 
  395 // 
  396 // 
  397 // 
  398 // 
  399 // 
  400 // 
  401 // 
  402 // 
  403 // 
  404 // 
  405 // 
  406 // 
  407 // 
  408 // 
  409 // 
  410 // 
  411 // 
  412 // 
  413 // 
  414 // 
  415 // 
  416 // 
  417 // 
  418 // 
  419 // 
  420 // 
  421 // 
  422 // 
  423 // 
  424 // 
  425 // 
  426 // 
  427 // 
  428 // 
  429 // 
  430 // 
  431 // 
  432 // 
  433 // 
  434 // 
  435 // 
  436 // 
  437 // 
  438 // 
  439 // 
  440 // 
  441 // 
  442 // 
  443 // 
  444 // 
  445 // 
  446 // 
  447 // 
  448 // 
  449 // 
  450 // 
  451 // 
  452 // 
  453 // 
  454 // 
  455 // 
  456 // 
  457 // 
  458 // 
  459 // 
  460 // 
  461 // 
  462 // 
  463 // 
  464 // 
  465 // 
  466 // 
  467 // 
  468 // 
  469 // 
  470 // 
  471 // 
  472 // 
  473 // 
  474 // 
  475 // 
  476 // 
  477 // 
  478 // 
  479 // 
  480 // 
  481 // 
  482 // 
  483 // 
  484 // 
  485 // 
  486 // 
  487 // 
  488 // 
  489 // 
  490 // 
  491 // 
  492 // 
  493 // 
  494 // 
  495 // 
  496 // 
  497 // 
  498 // 
  499 // 
  500 // 
  501 // 
  502 // 
  503 // 
  504 // 
  505 // 
  506 // 
  507 // 
  508 // 
  509 // 
  510 // 
  511 // 
  512 // 
  513 // 
  514 // 
  515 // 
  516 // 
  517 // 
  518 // 
  519 // 
  520 // 
  521 // 
  522 // 
  523 // 
  524 // 
  525 // 
  526 // 
  527 // 
  528 // 
  529 // 
  530 // 
  531 // 
  532 // 
  533 // 
  534 // 
  535 // 
  536 // 
  537 // 
  538 // 
  539 // 
  540 // 
  541 // 
  542 // 
  543 // 
  544 // 
  545 // 
  546 // 
  547 // 
  548 // 
  549 // 
  550 // 
  551 // 
  552 // 
  553 // 
  554 // 
  555 // 
  556 // 
  557 // 
  558 // 
  559 // 
  560 // 
  561 // 
  562 // 
  563 // 
  564 // 
  565 // 
  566 // 
  567 // 
  568 // 
  569 // 
  570 // 
  571 // 
  572 // 
  573 // 
  574 // 
  575 // 
  576 // 
  577 // 
  578 // 
  579 // 
  580 // 
  581 // 
  582 // 
  583 // 
  584 // 
  585 // 
  586 // 
  587 // 
  588 // 
  589 // 
  590 // 
  591 // 
  592 // 
  593 // 
  594 // 
  595 // 
  596 // 
  597 // 
  598 // 
  599 // 
  600 // 
  601 // 
  602 // 
  603 // 
  604 // 
  605 // 
  606 // 
  607 // 
  608 // 
  609 // 
  610 // 
  611 // 
  612 // 
  613 // 
  614 // 
  615 // 
  616 // 
  617 // 
  618 // 
  619 // 
  620 // 
  621 // 
  622 // 
  623 // 
  624 // 
  625 // 
  626 // 
  627 // 
  628 // 
  629 // 
  630 // 
  631 // 
  632 // 
  633 // 
  634 // 
  635 // 
  636 // 
  637 // 
  638 // 
  639 // 
  640 // 
  641 // 
  642 // 
  643 // 
  644 // 
  645 // 
  646 // 
  647 // 
  648 // 
  649 // 
  650 // 
  651 // 
  652 // 
  653 // 
  654 // 
  655 // 
  656 // 
  657 // 
  658 // 
  659 // 
  660 // 
  661 // 
  662 // 
  663 // 
  664 // 
  665 // 
  666 // 
  667 // 
  668 // 
  669 // 
  670 // 
  671 // 
  672 // 
  673 // 
  674 // 
  675 // 
  676 // 
  677 // 
  678 // 
  679 // 
  680 // 
  681 // 
  682 // 
  683 // 
  684 // 
  685 // 
  686 // 
  687 // 
  688 // 
  689 // 
  690 // 
  691 // 
  692 // 
  693 // 
  694 // 
  695 // 
  696 // 
  697 // 
  698 // 
  699 // 
  700 // 
  701 // 
  702 // 
  703 // 
  704 // 
  705 // 
  706 // 
  707 // 
  708 // 
  709 // 
  710 // 
  711 // 
  712 // 
  713 // 
  714 // 
  715 // 
  716 // 
  717 // 
  718 // 
  719 // 
  720 // 
  721 // 
  722 // 
  723 // 
  724 // 
  725 // 
  726 // 
  727 // 
  728 // 
  729 // 
  730 // 
  731 // 
  732 // 
  733 // 
  734 // 
  735 // 
  736 // 
  737 // 
  738 // 
  739 // 
  740 // 
  741 // 
  742 // 
  743 // 
  744 // 
  745 // 
  746 // 
  747 // 
  748 // 
  749 // 
  750 // 
  751 // 
  752 // 
  753 // 
  754 // 
  755 // 
  756 // 
  757 // 
  758 // 
  759 // 
  760 // 
  761 // 
  762 // 
  763 // 
  764 // 
  765 // 
  766 // 
  767 // 
  768 // 
  769 // 
  770 // 
  771 // 
  772 // 
  773 // 
  774 // 
  775 // 
  776 // 
  777 // 
  778 // 
  779 // 
  780 // 
  781 // 
  782 // 
  783 // 
  784 // 
  785 // 
  786 // 
  787 // 
  788 // 
  789 // 
  790 // 
  791 // 
  792 // 
  793 // 
  794 // 
  795 // 
  796 // 
  797 // 
  798 // 
  799 // 
  800 // 
  801 // 
  802 // 
  803 // 
  804 // 
  805 // 
  806 // 
  807 // 
  808 // 
  809 // 
  810 // 
  811 // 
  812 // 
  813 // 
  814 // 
  815 // 
  816 // 
  817 // 
  818 // 
  819 // 
  820 // 
  821 // 
  822 // 
  823 // 
  824 // 
  825 // 
  826 // 
  827 // 
  828 // 
  829 // 
  830 // 
  831 // 
  832 // 
  833 // 
  834 // 
  835 // 
  836 // 
  837 // 
  838 // 
  839 // 
  840 // 
  841 // 
  842 // 
  843 // 
  844 // 
  845 // 
  846 // 
  847 // 
  848 // 
  849 // 
  850 // 
  851 // 
  852 // 
  853 // 
  854 // 
  855 // 
  856 // 
  857 // 
  858 // 
  859 // 
  860 // 
  861 // 
  862 // 
  863 // 
  864 // 
  865 // 
  866 // 
  867 // 
  868 // 
  869 // 
  870 // 
  871 // 
  872 // 
  873 // 
  874 // 
  875 // 
  876 // 
  877 // 
  878 // 
  879 // 
  880 // 
  881 // 
  882 // 
  883 // 
  884 // 
  885 // 
  886 // 
  887 // 
  888 // 
  889 // 
  890 // 
  891 // 
  892 // 
  893 // 
  894 // 
  895 // 
  896 // 
  897 // 
  898 // 
  899 // 
  900 // 
  901 // 
  902 // 
  903 // 
  904 // 
  905 // 
  906 // 
  907 // 
  908 // 
  909 // 
  910 // 
  911 // 
  912 // 
  913 // 
  914 // 
  915 // 
  916 // 
  917 // 
  918 // 
  919 // 
  920 // 
  921 // 
  922 // 
  923 // 
  924 // 
  925 // 
  926 // 
  927 // 
  928 // 
  929 // 
  930 // 
  931 // 
  932 // 
  933 // 
  934 // 
  935 // 
  936 // 
  937 // 
  938 // 
  939 // 
  940 // 
  941 // 
  942 // 
  943 // 
  944 // 
  945 // 
  946 // 
  947 // 
  948 // 
  949 // 
  950 // 
  951 // 
  952 // 
  953 // 
  954 // 
  955 // 
  956 // 
  957 // 
  958 // 
  959 // 
  960 // 
  961 // 
  962 // 
  963 // 
  964 // 
  965 // 
  966 // 
  967 // 
  968 // 
  969 // 
  970 // 
  971 // 
  972 // 
  973 // 
  974 // 
  975 // 
  976 // 
  977 // 
  978 // 
  979 // 
  980 // 
  981 // 
  982 // 
  983 // 
  984 // 
  985 // 
  986 // 
  987 // 
  988 // 
  989 // 
  990 // 
  991 // 
  992 // 
  993 // 
  994 // 
  995 // 
  996 // 
  997 // 
  998 // 
  999 // 
  1000 // 
  1001 // 
  1002 // 
  1003 // 
  1004 // 
  1005 // 
  1006 // 
  1007 // 
  1008 // 
  1009 // 
  1010 // 
  1011 // 
  1012 // 
  1013 // 
  1014 // 
  1015 // 
  1016 // 
  1017 // 
  1018 // 
  1019 // 
  1020 // 
  1021 // 
  1022 // 
  1023 // 
  1024 // 
  1025 // 
  1026 // 
  1027 // 
  1028 // 
  1029 // 
  1030 // 
  1031 // 
  1032 // 
  1033 // 
  1034 // 
  1035 // 
  1036 // 
  1037 // 
  1038 // 
  1039 // 
  1040 // 
  1041 // 
  1042 // 
  1043 // 
  1044 // 
  1045 // 
  1046 // 
  1047 // 
  1048 // 
  1049 // 
  1050 // 
  1051 // 
  1052 // 
  1053 // 
  1054 // 
  1055 // 
  1056 // 
  1057 // 
  1058 // 
  1059 // 
  1060 // 
  1061 // 
  1062 // 
  1063 // 
  1064 // 
  1065 // 
  1066 // 
  1067 // 
  1068 // 
  1069 // 
  1070 // 
  1071 // 
  1072 // 
  1073 // 
  1074 // 
  1075 // 
  1076 // 
  1077 // 
  1078 // 
  1079 // 
  1080 // 
  1081 // 
  1082 // 
  1083 // 
  1084 // 
  1085 // 
  1086 // 
  1087 // 
  1088 // 
  1089 // 
  1090 // 
  1091 // 
  1092 // 
  1093 // 
  1094 // 
  1095 // 
  1096 // 
  1097 // 
  1098 // 
  1099 // 
  1100 // 
  1101 // 
  1102 // 
  1103 // 
  1104 // 
  1105 // 
  1106 // 
  1107 // 
  1108 // 
  1109 // 
  1110 // 
  1111 // 
  1112 // 
  1113 // 
  1114 // 
  1115 // 
  1116 // 
  1117 // 
  1118 // 
  1119 // 
  1120 // 
  1121 // 
  1122 // 
  1123 // 
  1124 // 
  1125 // 
  1126 // 
  1127 // 
  1128 // 
  1129 // 
  1130 // 
  1131 // 
  1132 // 
  1133 // 
  1134 // 
  1135 // 
  1136 // 
  1137 // 
  1138 // 
  1139 // 
  1140 // 
  1141 // 
  1142 // 
  1143 // 
  1144 // 
  1145 // 
  1146 // 
  1147 // 
  1148 // 
  1149 // 
  1150 // 
  1151 // 
  1152 // 
  1153 // 
  1154 // 
  1155 // 
  1156 // 
  1157 // 
  1158 // 
  1159 // 
  1160 // 
  1161 // 
  1162 // 
  1163 // 
  1164 // 
  1165 // 
  1166 // 
  1167 // 
  1168 // 
  1169 // 
  1170 // 
  1171 // 
  1172 // 
  1173 // 
  1174 // 
  1175 // 
  1176 // 
  1177 // 
  1178 // 
  1179 // 
  1180 // 
  1181 // 
  1182 // 
  1183 // 
  1184 // 
  1185 // 
  1186 // 
  1187 // 
  1188 // 
  1189 // 
  1190 // 
  1191 // 
  1192 // 
  1193 // 
  1194 // 
  1195 // 
  1196 // 
  1197 // 
  1198 // 
  1199 // 
  1200 // 
  1201 // 
  1202 // 
  1203 // 
  1204 // 
  1205 // 
  1206 // 
  1207 // 
  1208 // 
  1209 // 
  1210 // 
  1211 // 
  1212 // 
  1213 // 
  1214 // 
  1215 // 
  1216 // 
  1217 // 
  1218 // 
  1219 // 
  1220 // 
  1221 // 
  1222 // 
  1223 // 
  1224 // 
  1225 // 
  1226 // 
  1227 // 
  1228 // 
  1229 // 
  1230 // 
  1231 // 
  1232 // 
  1233 // 
  1234 // 
  1235 // 
  1236 // 
  1237 // 
  1238 // 
  1239 // 
  1240 // 
  1241 // 
  1242 // 
  1243 // 
  1244 // 
  1245 // 
  1246 // 
  1247 // 
  1248 // 
  1249 // 
  1250 // 
  1251 // 
  1252 // 
  1253 // 
  1254 // 
  1255 // 
  1256 // 
  1257 // 
  1258 // 
  1259 // 
  1260 // 
  1261 // 
  1262 // 
  1263 // 
  1264 // 
  1265 // 
  1266 // 
  1267 // 
  1268 // 
  1269 // 
  1270 // 
  1271 // 
  1272 // 
  1273 // 
  1274 // 
  1275 // 
  1276 // 
  1277 // 
  1278 // 
  1279 // 
  1280 // 
  1281 // 
  1282 // 
  1283 // 
  1284 // 
  1285 // 
  1286 // 
  1287 // 
  1288 // 
  1289 // 
  1290 // 
  1291 // 
  1292 // 
  1293 // 
  1294 // 
  1295 // 
  1296 // 
  1297 // 
  1298 // 
  1299 // 
  1300 // 
  1301 // 
  1302 // 
  1303 // 
  1304 // 
  1305 // 
  1306 // 
  1307 // 
  1308 // 
  1309 // 
  1310 // 
  1311 // 
  1312 // 
  1313 // 
  1314 // 
  1315 // 
  1316 // 
  1317 // 
  1318 // 
  1319 // 
  1320 // 
  1321 // 
  1322 // 
  1323 // 
  1324 // 
  1325 // 
  1326 // 
  1327 // 
  1328 // 
  1329 // 
  1330 // 
  1331 // 
  1332 // 
  1333 // 
  1334 // 
  1335 // 
  1336 // 
  1337 // 
  1338 // 
  1339 // 
  1340 // 
  1341 // 
  1342 // 
  1343 // 
  1344 // 
  1345 // 
  1346 // 
  1347 // 
  1348 // 
  1349 // 
  1350 // 
  1351 // 
  1352 // 
  1353 // 
  1354 // 
  1355 // 
  1356 // 
  1357 // 
  1358 // 
  1359 // 
  1360 // 
  1361 // 
  1362 // 
  1363 // 
  1364 // 
  1365 // 
  1366 // 
  1367 // 
  1368 // 
  1369 // 
  1370 // 
  1371 // 
  1372 // 
  1373 // 
  1374 // 
  1375 // 
  1376 // 
  1377 // 
  1378 // 
  1379 // 
  1380 // 
  1381 // 
  1382 // 
  1383 // 
  1384 // 
  1385 // 
  1386 // 
  1387 // 
  1388 // 
  1389 // 
  1390 // 
  1391 // 
  1392 // 
  1393 // 
  1394 // 
  1395 // 
  1396 // 
  1397 // 
  1398 // 
  1399 // 
  1400 // 
  1401 // 
  1402 // 
  1403 // 
  1404 // 
  1405 // 
  1406 // 
  1407 // 
  1408 // 
  1409 // 
  1410 // 
  1411 // 
  1412 // 
  1413 // 
  1414 // 
  1415 // 
  1416 // 
  1417 // 
  1418 // 
  1419 // 
  1420 // 
  1421 // 
  1422 // 
  1423 // 
  1424 // 
  1425 // 
  1426 // 
  1427 // 
  1428 // 
  1429 // 
  1430 // 
  1431 // 
  1432 // 
  1433 // 
  1434 // 
  1435 // 
  1436 // 
  1437 // 
  1438 // 
  1439 // 
  1440 // 
  1441 // 
  1442 // 
  1443 // 
  1444 // 
  1445 // 
  1446 // 
  1447 // 
  1448 // 
  1449 // 
  1450 // 
  1451 // 
  1452 // 
  1453 // 
  1454 // 
  1455 // 
  1456 // 
  1457 // 
  1458 // 
  1459 // 
  1460 // 
  1461 // 
  1462 // 
  1463 // 
  1464 // 
  1465 // 
  1466 // 
  1467 // 
  1468 // 
  1469 // 
  1470 // 
  1471 // 
  1472 // 
  1473 // 
  1474 // 
  1475 // 
  1476 // 
  1477 // 
  1478 // 
  1479 // 
  1480 // 
  1481 // 
  1482 // 
  1483 // 
  1484 // 
  1485 // 
  1486 // 
  1487 // 
  1488 // 
  1489 // 
  1490 // 
  1491 // 
  1492 // 
  1493 // 
  1494 // 
  1495 // 
  1496 // 
  1497 // 
  1498 // 
  1499 // 
  1500 // 
  1501 // 
  1502 // 
  1503 // 
  1504 // 
  1505 // 
  1506 // 
  1507 // 
  1508 // 
  1509 // 
  1510 // 
  1511 // 
  1512 // 
  1513 // 
  1514 // 
  1515 // 
  1516 // 
  1517 // 
  1518 // 
  1519 // 
  1520 // 
  1521 // 
  1522 // 
  1523 // 
  1524 // 
  1525 // 
  1526 // 
  1527 // 
  1528 // 
  1529 // 
  1530 // 
  1531 // 
  1532 // 
  1533 // 
  1534 // 
  1535 // 
  1536 // 
  1537 // 
  1538 // 
  1539 // 
  1540 // 
  1541 // 
  1542 // 
  1543 // 
  1544 // 
  1545 // 
  1546 // 
  1547 // 
  1548 // 
  1549 // 
  1550 // 
  1551 // 
  1552 // 
  1553 // 
  1554 // 
  15
```

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.

[See the next page](#) for a function picture of the SKR PRO V1.1 wire diagram.

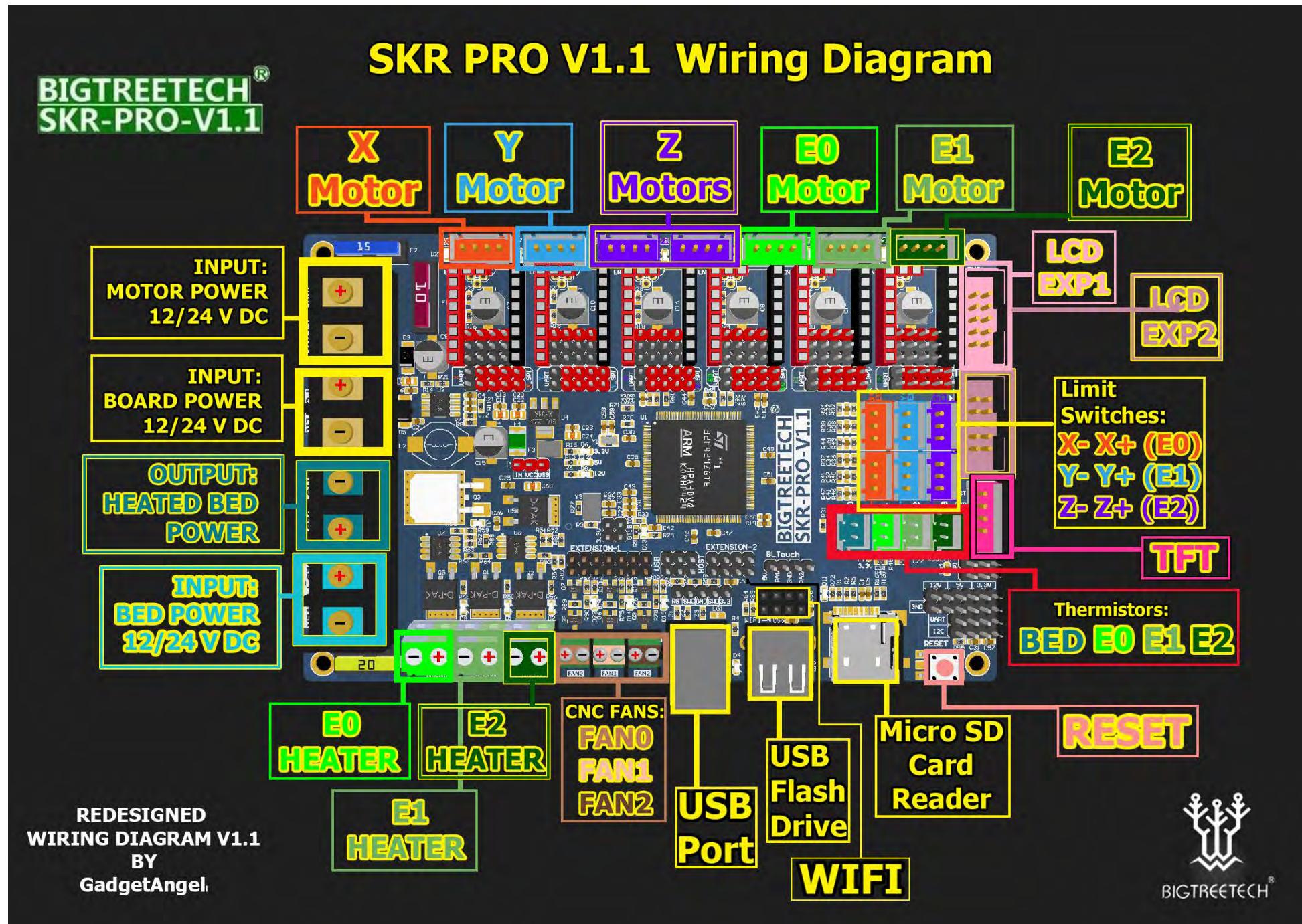
3



- 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 file we see FAN2 port is defined as PE6. So you can choose to copy "PE6" or "FAN2_PIN". Go to next page.

3 Pin Diagram is done by Thomas White and updated by GadgetAngel

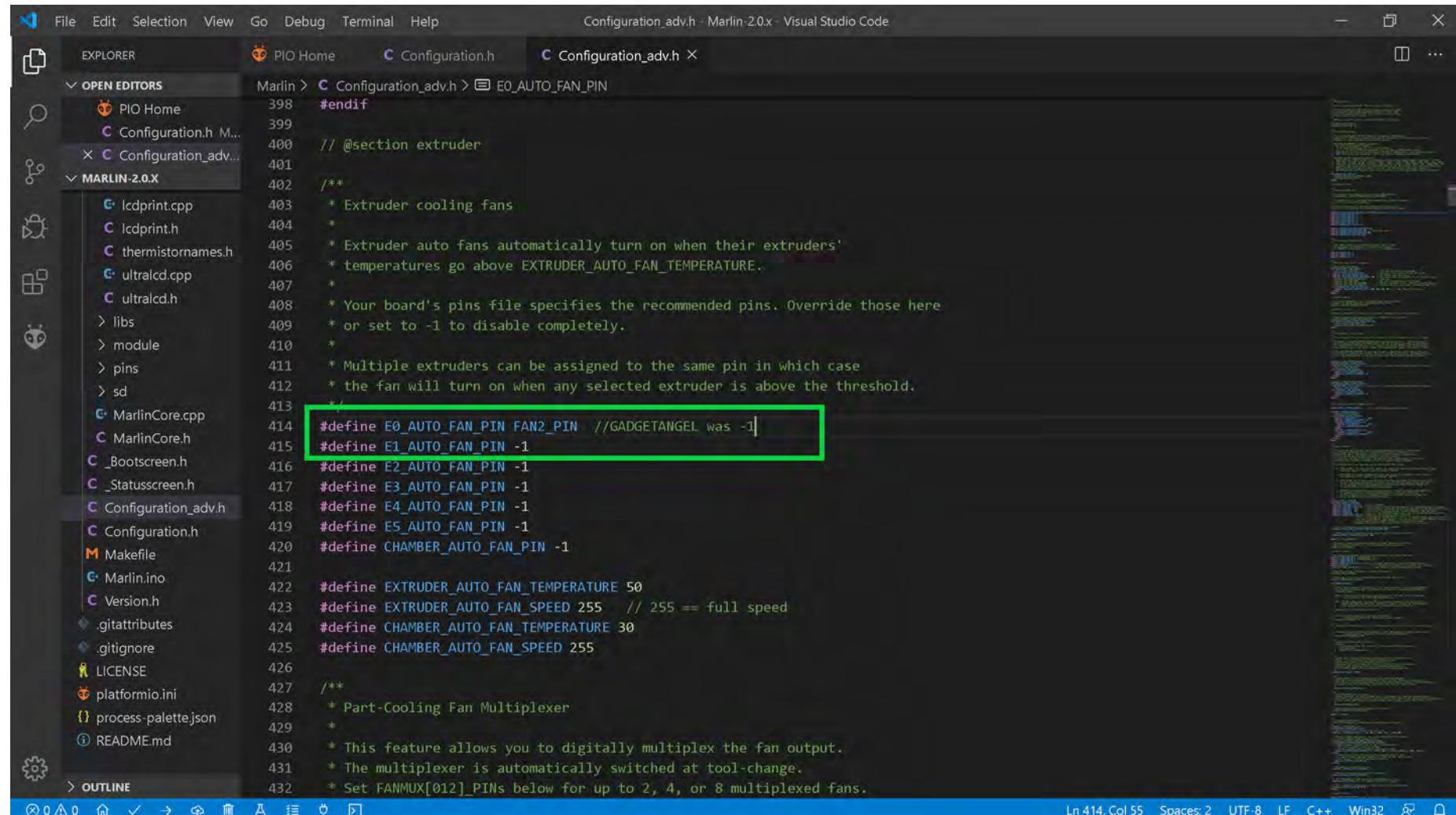
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers



- Go to the next page.

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 PIO Home Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > E0_AUTO_FAN_PIN
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 //GADGETANGEL 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.
432 * Set FANMUX[012]_PINs below for up to 2, 4, or 8 multiplexed fans.
Ln 414, Col 55 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.](#)