

Stepper Driver Configurations for SKR PRO V1.1 Board



By

@GadgetAngel



Based on Work by @rfulling

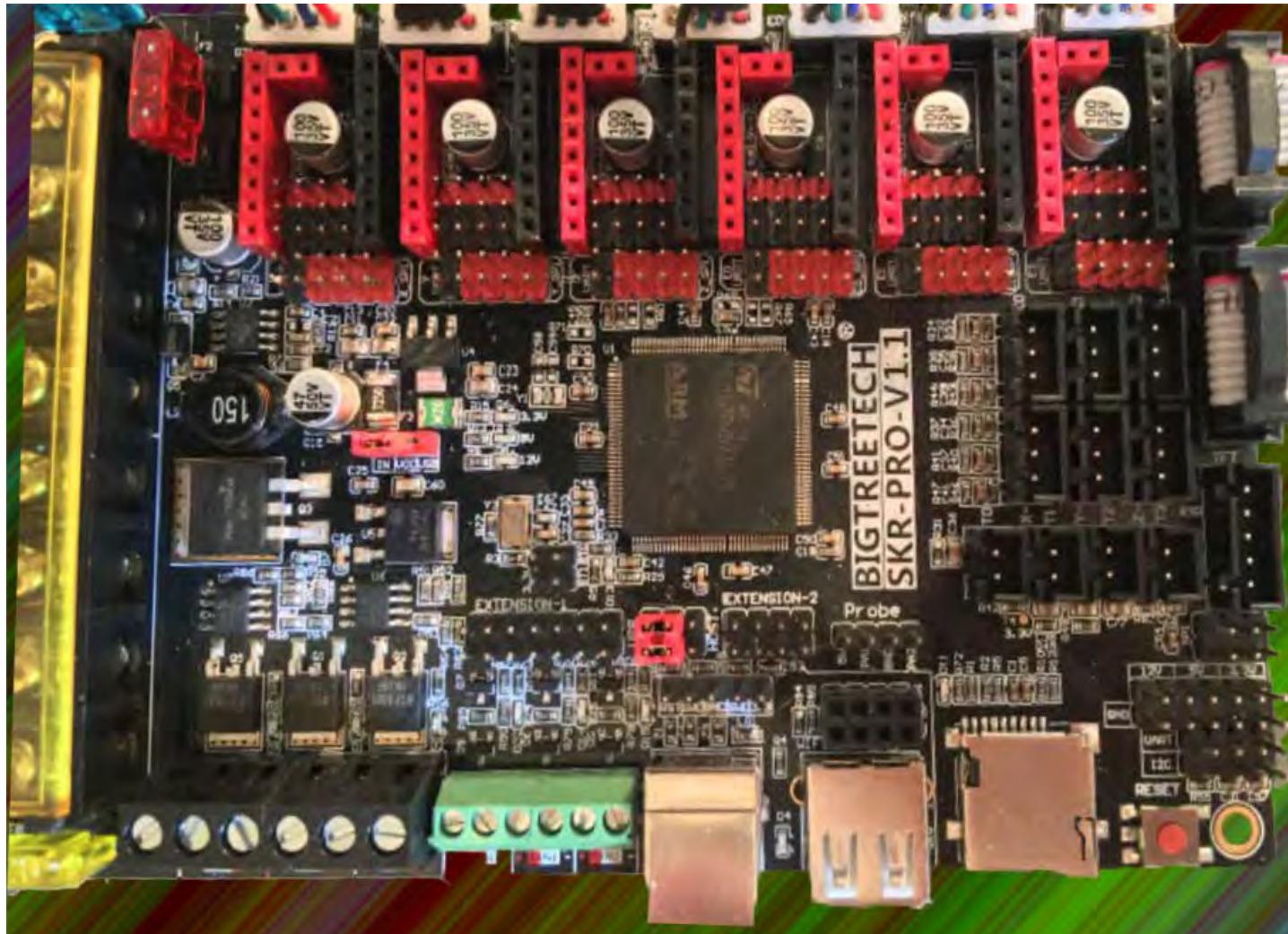


Table of Contents

POLOLU A4988	1
The (latest release of) Marlin Setup for POLOLU A4988 Drivers	6
BIQU A4988	8
The (latest release of) Marlin Setup for BIQU A4988 Drivers	13
DRV8825	15
The (latest release of) Marlin Setup for DRV8825 Drivers	21
BIQU LV8729	24
The (latest release of) Marlin Setup for BIQU LV8729 Drivers	30
FYSETC LV8729	34
The (latest release of) Marlin Setup for FYSETC LV8729 Drivers	40
LERDGE LV8729	44
The (latest release of) Marlin Setup for LERDGE LV8729 Drivers	50
MKS LV8729	54
The (latest release of) Marlin Setup for MKS LV8729 Drivers	60
FYSETC S6128 V1.1	64
The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers	70
FYSETC ST820	74
The (latest release of) Marlin Setup for FYSETC ST820 Drivers	80
BIQU ST820	85
The (latest release of) Marlin Setup for BIQU ST820 Drivers	91
POLOLU ST820 (STSPIN820)	96
The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers	102
POLOLU MP6500	107
The (latest release of) Marlin Setup for POLOLU MP6500 Drivers	112

Table of Contents

POLOLU TB67S249FTG.....	116
The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers.....	122
BIQU S109	126
The (latest release of) Marlin Setup for BIQUE S109 Drivers	132
FYSETC S109	136
The (latest release of) Marlin Setup for FYSETC S109 Drivers	142
BIQU TMC2100 Stand-alone Mode.....	146
The (latest release of) Marlin Setup for BIQUE TMC2100 Drivers in Stand-alone Mode	153
MKS TMC2100 Stand-alone Mode.....	157
The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode	164
BIQU TMC2130 Stand-alone Mode.....	168
The (latest release of) Marlin Setup for BIQUE TMC2130 Drivers in Stand-alone Mode	175
BIQU TMC2130 SPI Mode	179
The (latest release of) Marlin Setup for BIQUE TMC2130 Drivers in SPI Mode	182
BIQU TMC2208 V3.0 Stand-alone Mode	194
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in Stand-alone Mode	199
BIQU TMC2208 V3.0 One Time Programming (OTP) Mode	203
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode	208
BIQU TMC2208 V3.0 UART Mode.....	212
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in UART Mode	215
FYSETC TMC2208 V1.2 Stand-alone Mode	224
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode.....	229
FYSETC TMC2208 V1.2 One Time Programming (OTP) Mode.....	233
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode.....	238

Table of Contents

FYSETC TMC2208 V1.2 UART Mode.....	242
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode	245
BIQU TMC2225 V1.0 Stand-alone Mode	254
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode	259
BIQU TMC2225 V1.0 One Time Programming (OTP) Mode.....	263
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode ...	268
BIQU TMC2225 V1.0 UART Mode.....	272
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode	275
BIQU TMC2209 V1.2 Stand-alone Mode for StealthChop.....	284
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop	289
BIQU TMC2209 V1.2 Stand-alone Mode for SpreadCycle	293
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle	298
BIQU TMC2209 V1.2 UART Mode.....	302
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode	305
BIQU TMC5160 V1.2 SPI Mode.....	317
The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode.....	320
BIQU TMC5161 V1.0 SPI Mode.....	332
The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode.....	335
APPENDIX A.....	347
How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer	347
APPENDIX B.....	349
For the TMC drivers what's the difference between stand-alone mode and ("UART" or "SPI ") modes?.....	349
How to Calculate V_{ref} for Non-TMC Stepper Motor Divers	349
How to Calculate V_{ref} for TMC Stepper Motor Drivers.....	350

Table of Contents

Driving Current Calculation Formulas for TMC Stepper Motor Drivers	351
#1 TMC2100 with $R_s = 0.110\Omega$ (110mΩ).....	351
#2 TMC2130 with $R_s = 0.110\Omega$ (110mΩ).....	351
#3 TMC2208 with $R_s = 0.110\Omega$ (110mΩ) for Stand-alone Mode.....	351
#4 TMC2208 with $R_s = 0.110\Omega$ (110mΩ) for UART Mode	352
#5 TMC2209 with $R_s = 0.110\Omega$ (110mΩ) for Stand-alone Mode.....	352
#6 TMC2209 with $R_s = 0.110\Omega$ (110mΩ) for UART Mode	353
#7 TMC2225 with $R_s = 0.150\Omega$ (150mΩ) for UART Mode	353
#8 TMC5160 with $R_s = 0.075\Omega$ (75mΩ) for SPI Mode	353
#9 TMC5161 with $R_s = 0.062\Omega$ (62mΩ) for SPI Mode	354
#10 TMC2225 with $R_s = 0.150\Omega$ (150mΩ) for Stand-alone Mode.....	354
APPENDIX C.....	355
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers.....	355
Link to <u>BIGTREETECH SKR PRO V1.1 User Manual.pdf</u>	355
Link to <u>BIGTREETECH SKR PRO V1.1 Guide.pdf</u>	355
Link to <u>Pronterface Software</u>	356
Link to <u>How to Calibrate your 3D Printer</u>	356
SKR PRO V1.1 Color PIN Diagram	368
SKR PRO V1.1 Color Wiring Diagram.....	369
SKR PRO V1.1 Color PIN 1 Diagram	371
SKR PRO V1.1 Color Schematic Diagram	372
APPENDIX D.....	373
Legends for SKR PRO V1.1 Stepper Driver Socket Representations.....	373
Special Consideration of D Jumper for SKR PRO V1.1 Board.....	376
Examples for Stepper Driver Socket Representations	377

Table of Contents

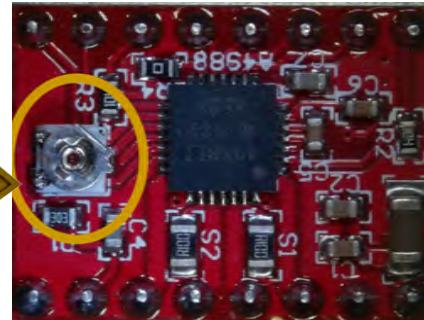
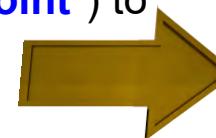
Example 1 (LV8729 Driver Board) for SKR PRO V1.1 Driver Socket Legend.....	377
Example 2 (A4988 Driver Board) for SKR PRO V1.1 Driver Socket Legend.....	378
Example 3 (UART Driver Board) for SKR PRO V1.1 Driver Socket Legend.....	379
Example 4 (SPI Driver Board) for SKR PRO V1.1 Driver Socket Legend.....	380
APPENDIX E.....	381
Location Of “firmware.bin” File from the Marlin Compilation for SKR PRO V1.1 Board.....	381
APPENDIX F	384
Links to Reference Material	384
Marlin Firmware Documentation	384
Information on Stepper Motor Drivers and Micro-stepping	384
POLOLU A4988 and BIQU A4988.....	385
DRV8825	385
BIQU LV8729, FYSETC LV8729, LERDGE LV8729, and MKS LV8729.....	386
BIQU LV8729	386
FYSETC LV8729	387
LERDGE LV8729	387
MKS LV8729	388
FYSETC S6128 V1.1	388
FYSETC ST820	389
BIQU ST820	389
POLOLU ST820 (STSPIN820)	390
POLOLU MP6500.....	390
POLOLU TB67S249FTG.....	391
BIQU S109 and FYSETC S109	391
BIQU S109.....	392

Table of Contents

FYSETC S109	392
Marlin Firmware Documentation Specific to TMC Drivers	393
Information Common to All TMC Drivers.....	393
BIQU TMC2100 and MKS TMC2100	394
BIQU TMC2130	395
Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2	395
Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2 (Continued).....	396
BIQU TMC2208 V3.0.....	396
FYSETC TMC2208 V1.2.....	396
Information Common to TMC2208 and BIQU TMC2225.....	397
BIQU TMC2225 V1.0.....	397
BIQU TMC2209 V1.2.....	398
BIQU TMC5160 V1.2.....	398
BIQU TMC5161 V1.0.....	399
SKR PRO V1.1 Board.....	399
Facebook Groups.....	400
Miscellaneous Information.....	401
APPENDIX G.....	402
BIGTREETECH Reference Material.....	402
Original PIN Diagram	402
Original Wiring Diagram 1	403
Original Wiring Diagram 2	404
Original PIN 1 Diagram	405
Original Schematic 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} .



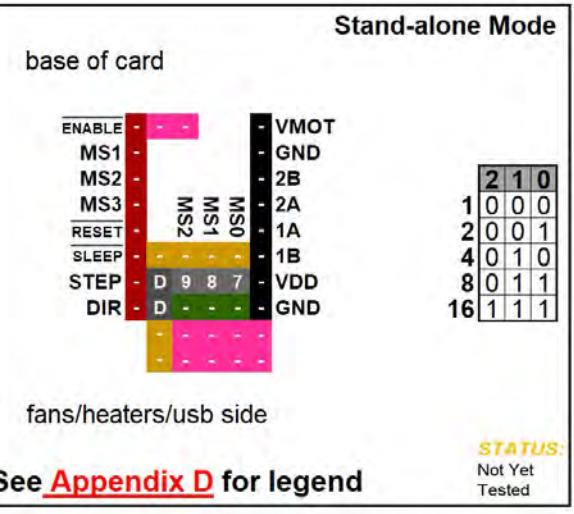
Note: " V_{ref} Test point" location is on the bottom of the driver board.



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>

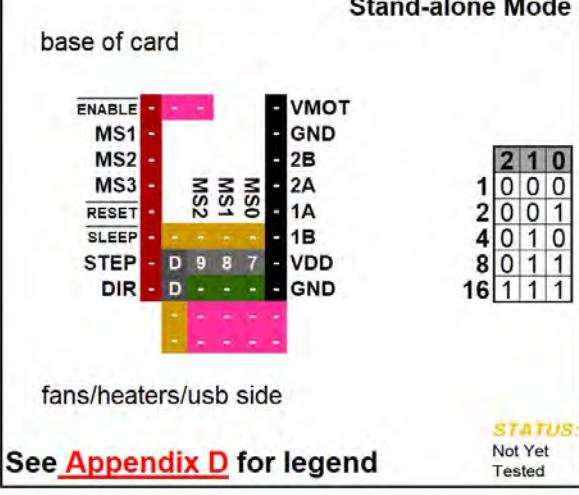


Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	2 Phase
	Low	Low	High	Half step	1-2 Phase
	Low	High	Low	Quarter step	W1-2 Phase
	Low	High	High	Eighth step	2W1-2 Phase
	High	High	High	Sixteenth step	4W1-2 Phase
Driving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

- See next page for the legend that belongs to the above chart.

POLOLU A4988SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

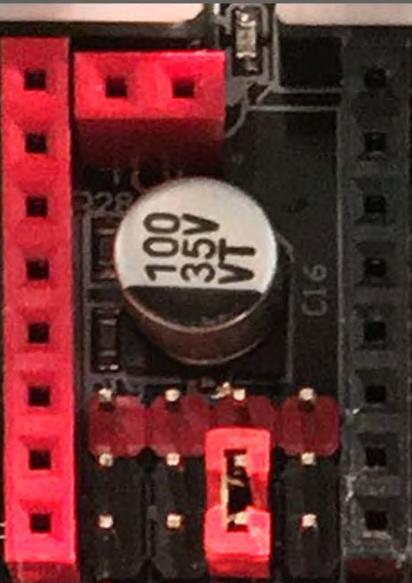
Low ➤ set Jumper between rows
2 and 3



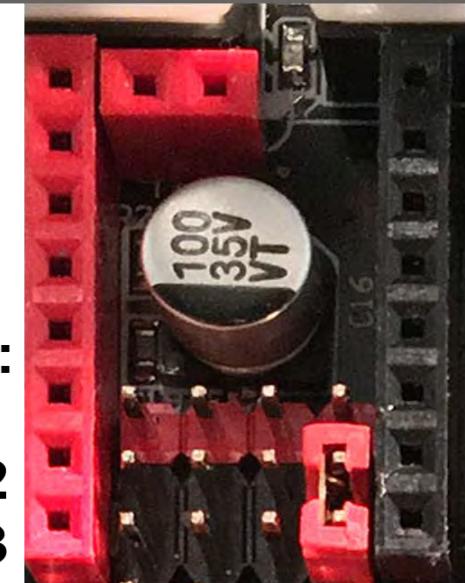
This is a
Jumper:



Row:
1
2
3

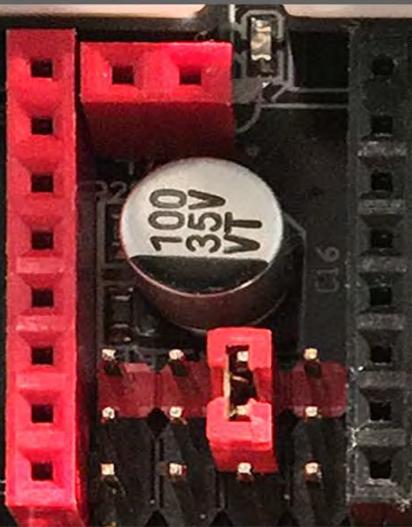


Row:
1
2
3

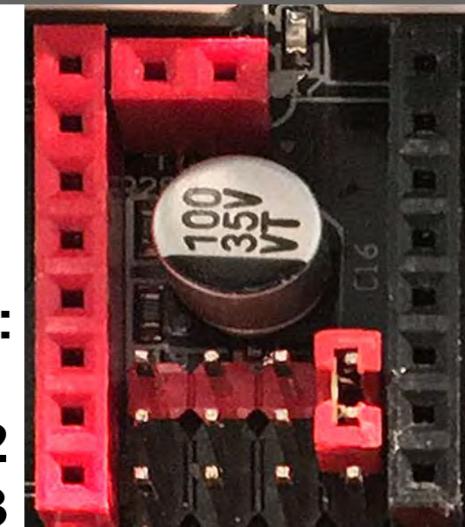


High ➤ set Jumper between rows
1 and 2

Row:
1
2
3

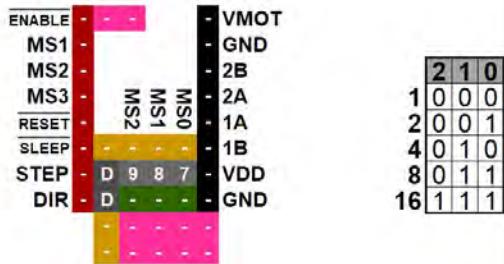


Row:
1
2
3



POLOLU A4988

base of card

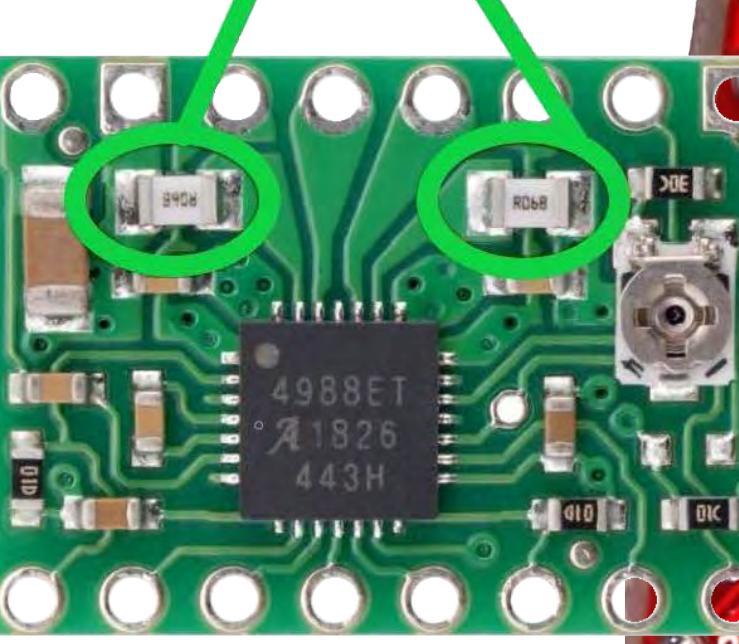


fans/heaters/usb side

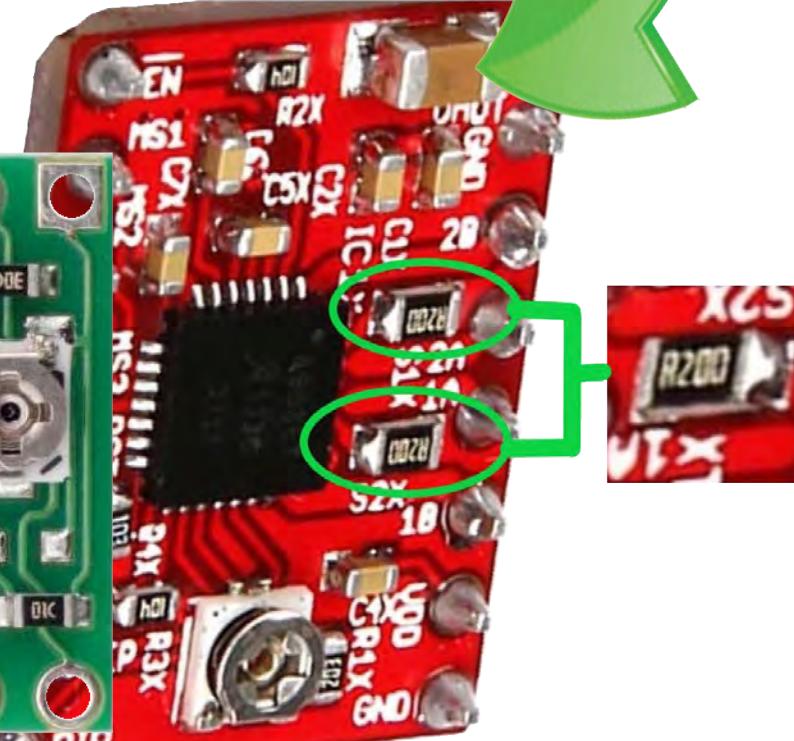
STATUS:
Not Yet
Tested

See [Appendix D](#) for legend

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



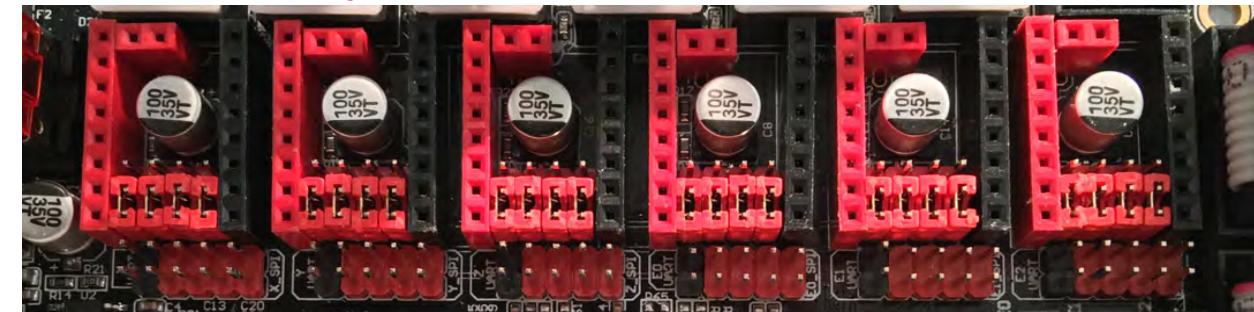
Stand-alone Mode

POLOLU A4988

STEP

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

Note: The "D" jumper MUST be SET!

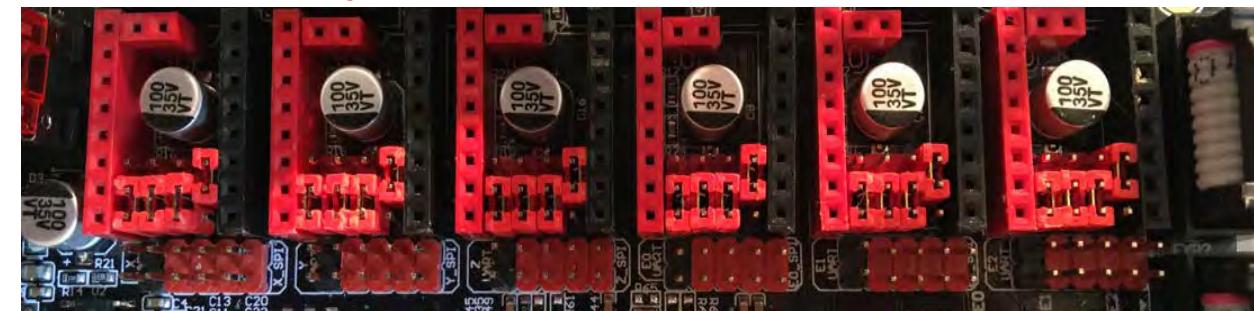


See [Appendix D](#) for legend

1 / 2

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

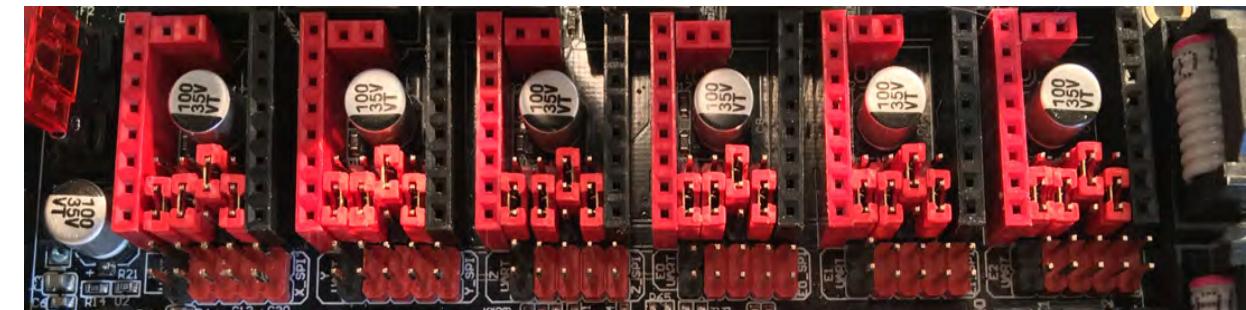
Stand-alone Mode

POLOLU A4988

1 / 4

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

Note: The "D" jumper MUST be SET!

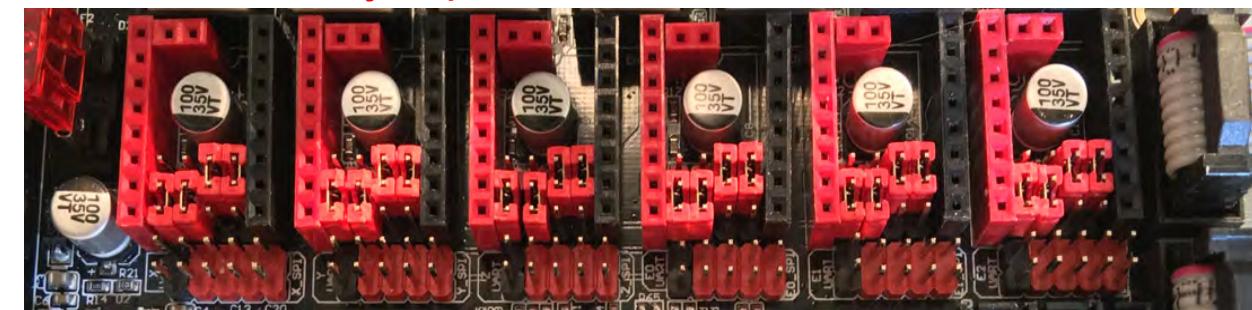


See [Appendix D](#) for legend

1 / 8

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

Note: The "D" jumper MUST be SET!

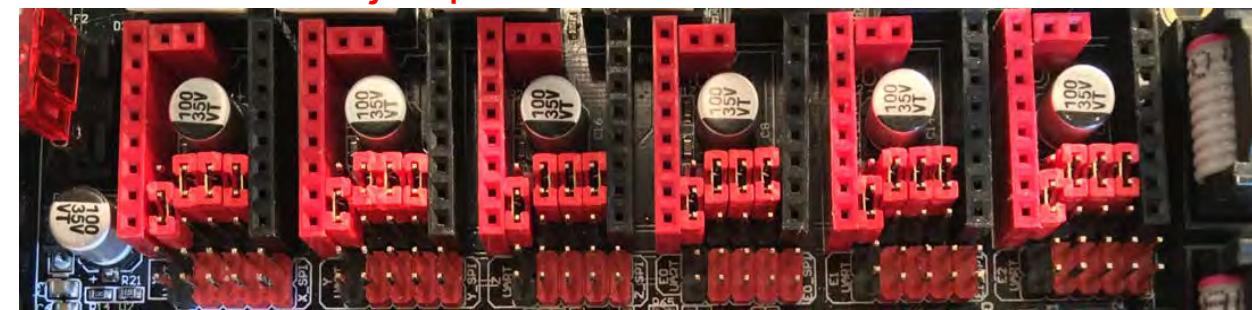


See [Appendix D](#) for legend

1 / 16

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

Note: The "D" jumper MUST be SET!

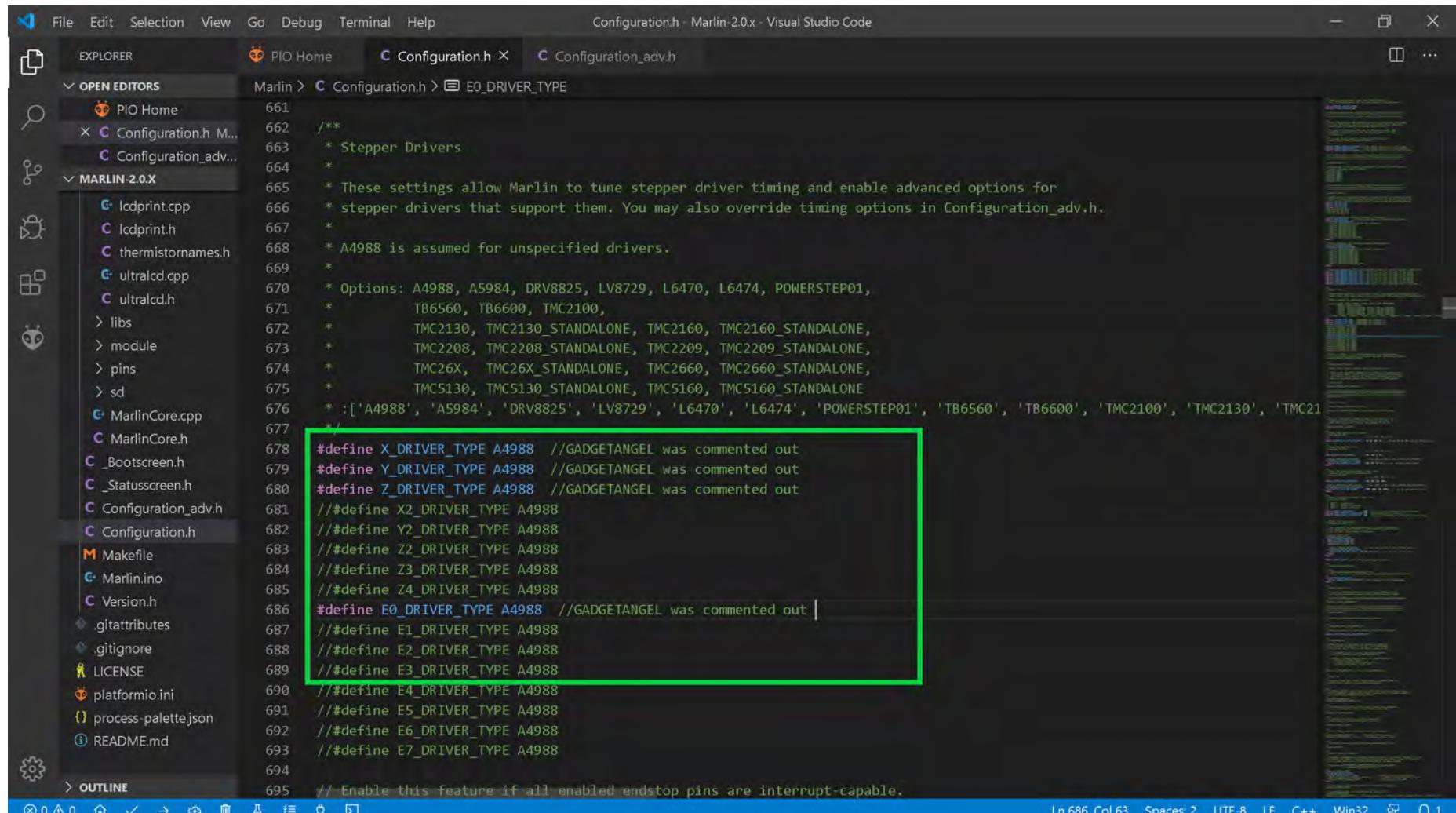


See [Appendix D](#) for legend

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 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.
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.

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

EXPLORER OPEN EDITORS pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h

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

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

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,

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

CONFIGURATION	RESULT	TIME
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	

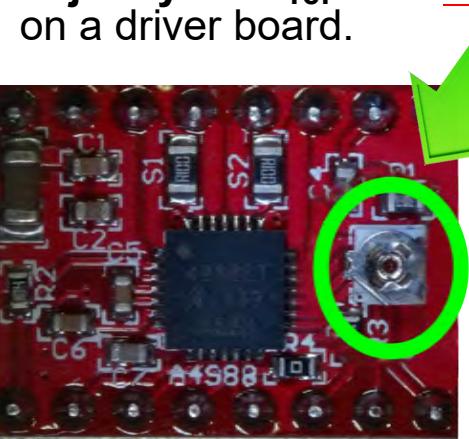
===== 1 succeeded in 00:02:31.294 =====

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 [Appendix E](#).

BIQU A4988

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	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	$I_{MAX} = V_{ref} / (8 * R_s)$			$V_{ref} = 8 * I_{MAX} * R_s$	
R_s (Typical Sense Resistor) = 0.1Ω					

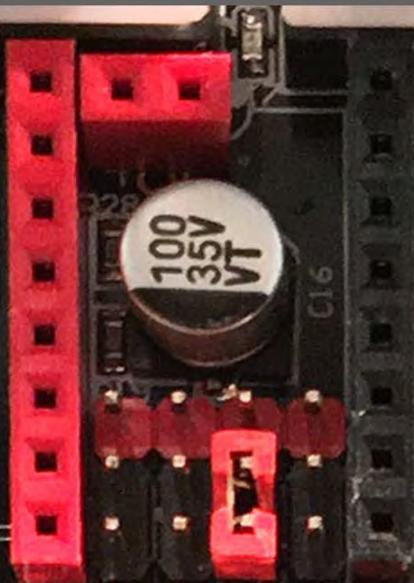
- See next page for the legend that belongs to the above chart.

BIQU A4988**SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

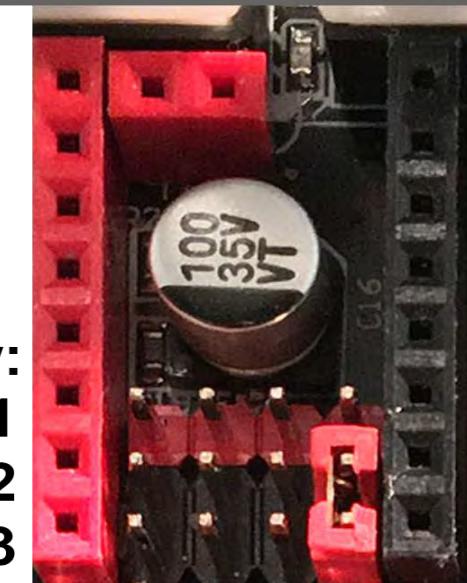
Low ➤ set Jumper between rows
2 and 3



Row:

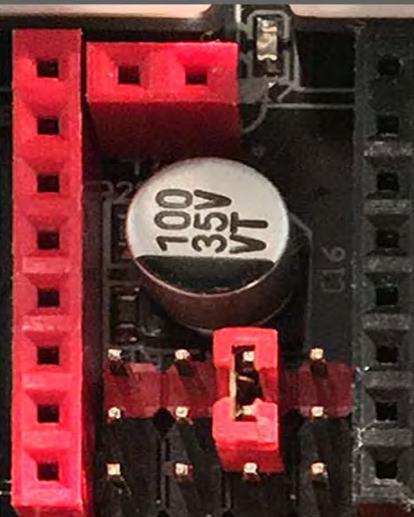
1
2
3

Row:

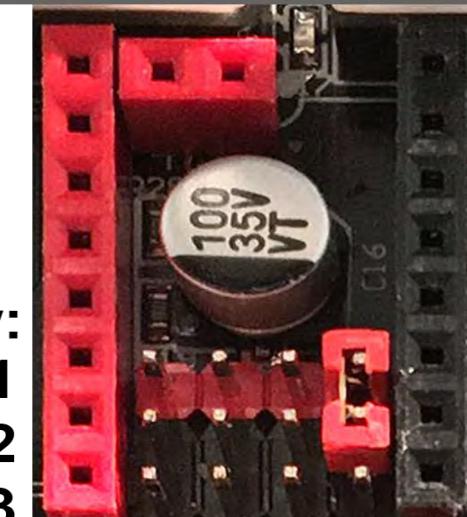
1
2
3

High ➤ set Jumper between rows
1 and 2

Row:

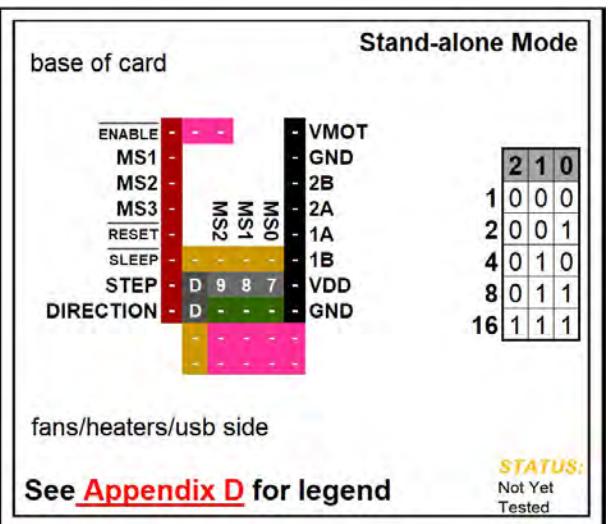
1
2
3

Row:

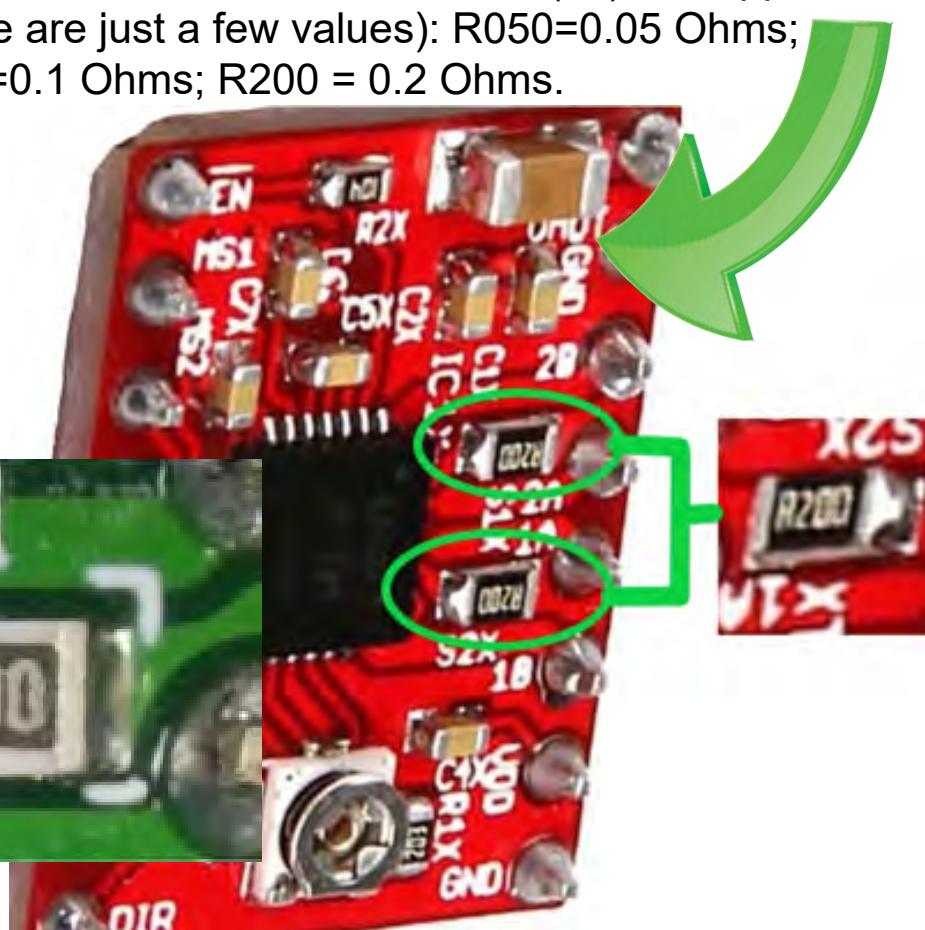
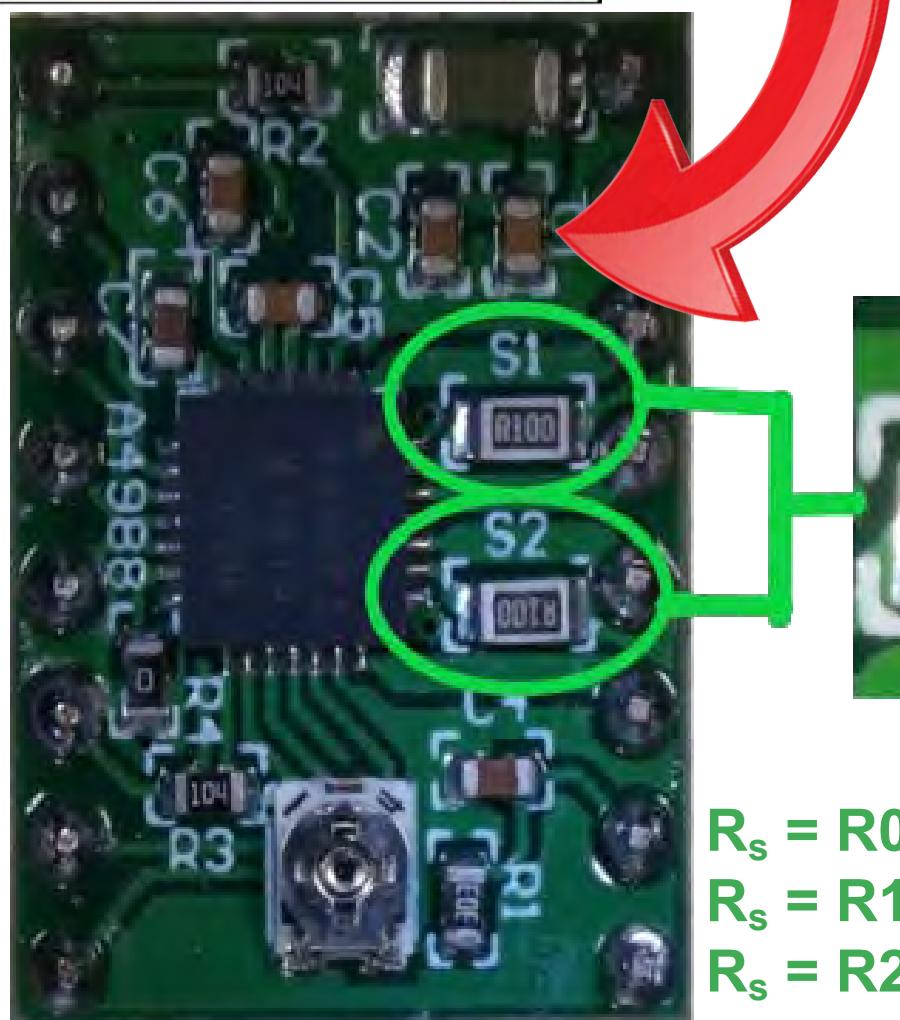
1
2
3

This is a
Jumper:



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

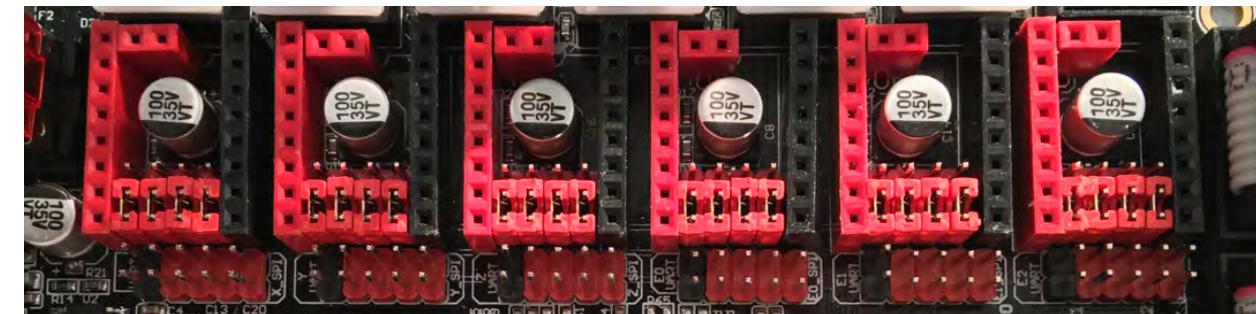
Stand-alone Mode

BIQU A4988

STEP

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

Note: The "D" jumper MUST be SET!

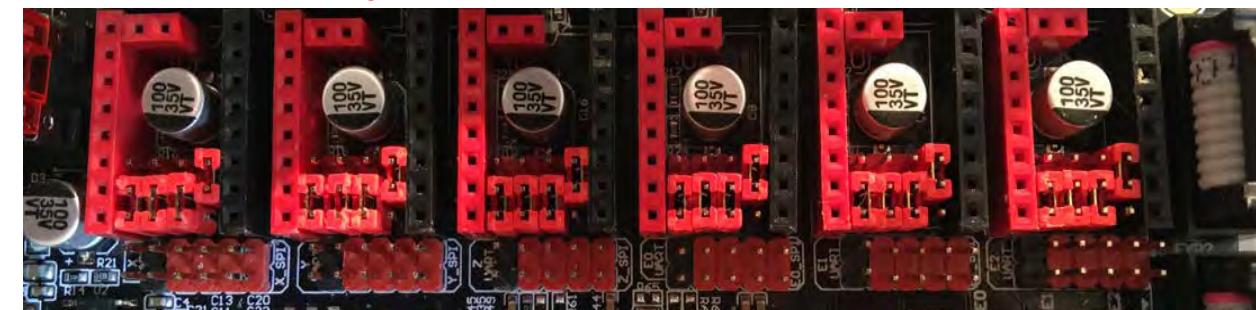


See [Appendix D](#) for legend

1 / 2

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

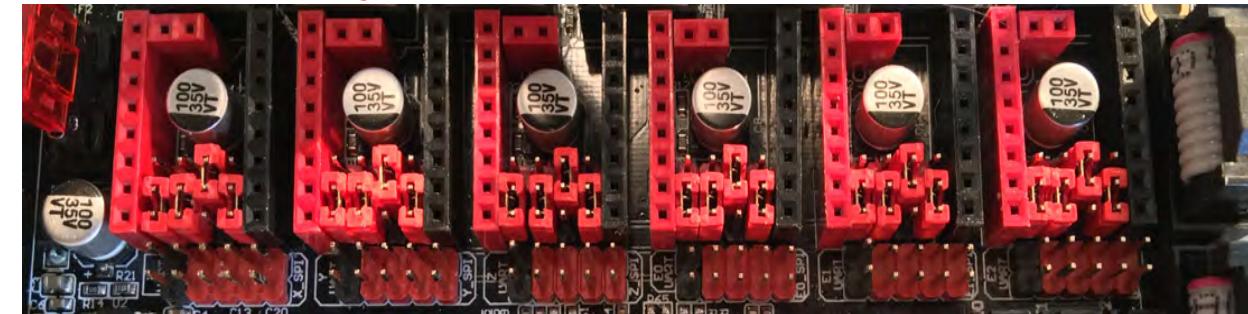
Stand-alone Mode

BINU A4988

1 / 4

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

Note: The "D" jumper MUST be SET!

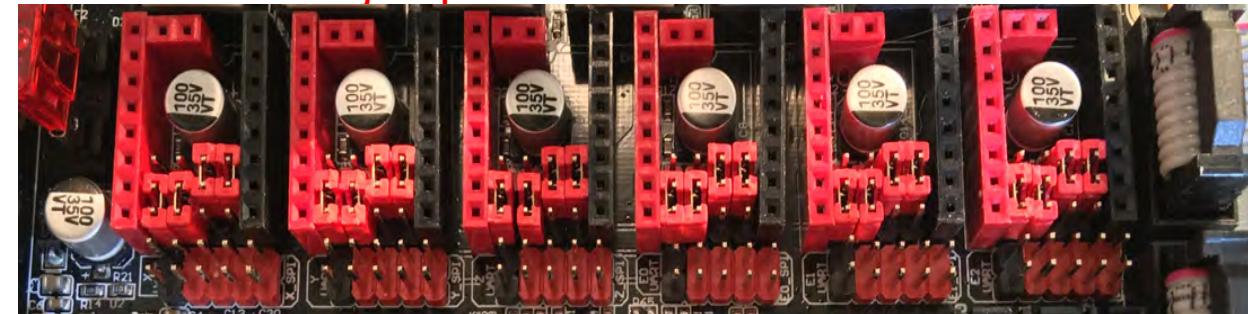


See [Appendix D](#) for legend

1 / 8

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

Note: The "D" jumper MUST be SET!

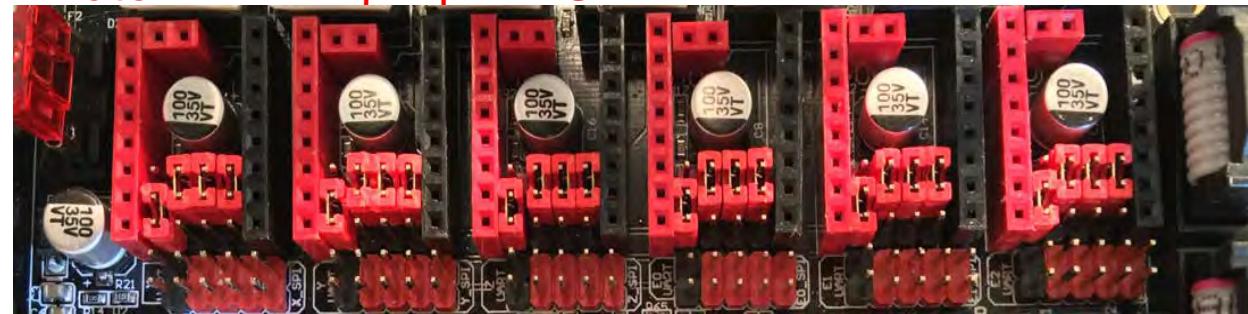


See [Appendix D](#) for legend

1 / 16

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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', 'TMC2
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 OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE

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

C pins_BTT_SKR_PRO_V1_1.h Marlin's... X 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
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

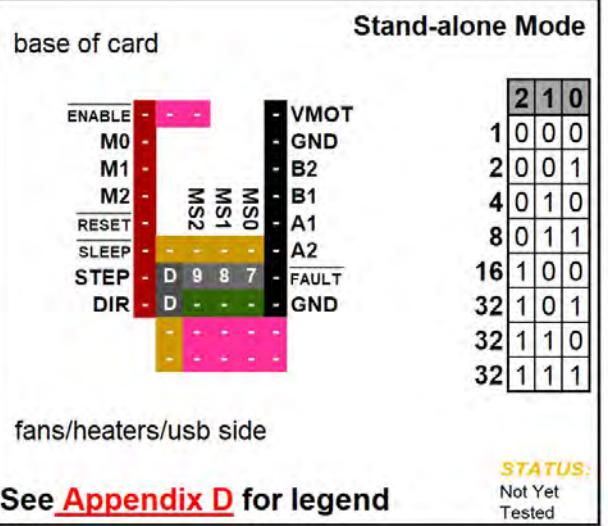
1: Task - Build

BIGTREETECH_BTT001	IGNORED		
BIGTREETECH_BTT001	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.

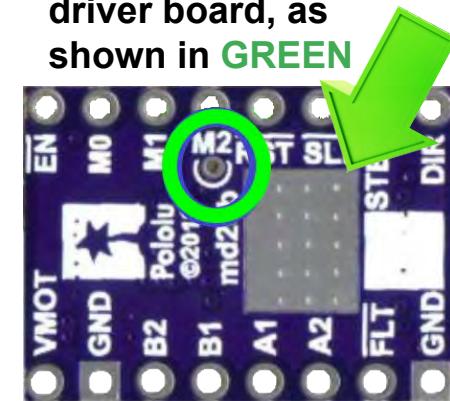
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

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.

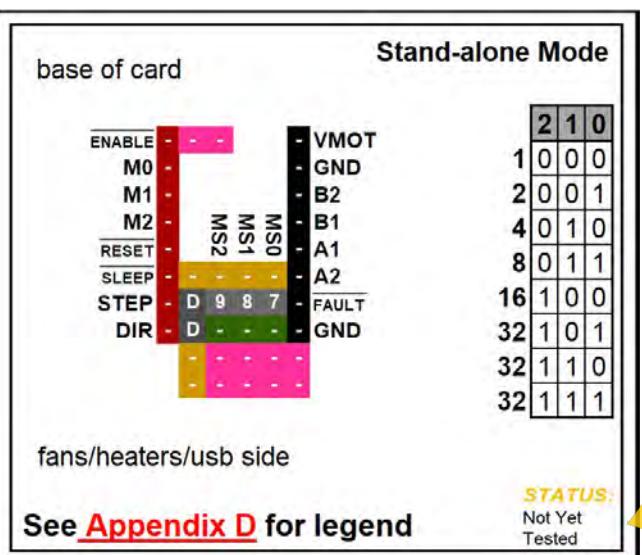
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

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 DRV8825 Maximum 32 Subdivision 45V DC 2.2A (peak)	Low	Low	Low	Full step	2 Phase
	Low	Low	High	Half 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	1/32 step	8W1-2 Phase
	High	High	High	1/32 step	8W1-2 Phase
Driving Current Calculation Formula		$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$	
R_S (Typical Sense Resistor) = 0.1Ω					

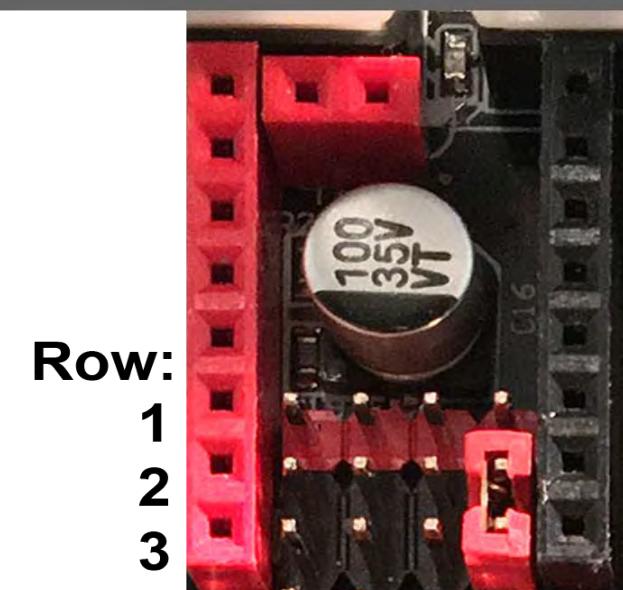
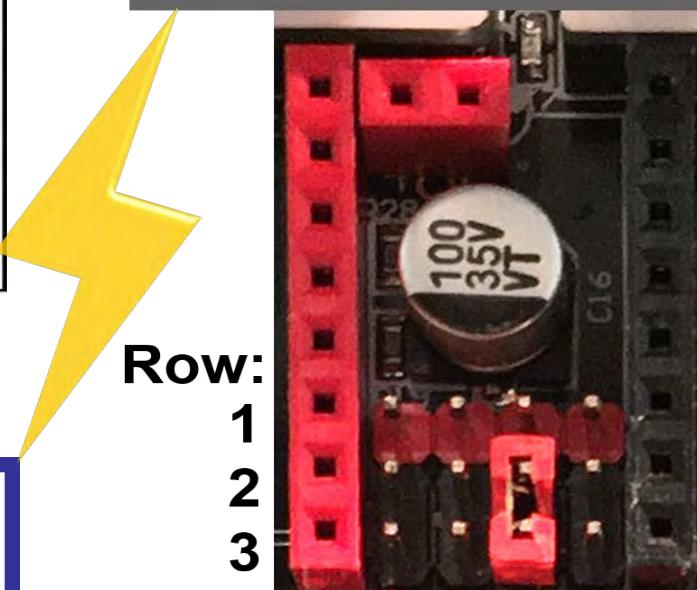
- See next page for the legend that belongs to the above chart.



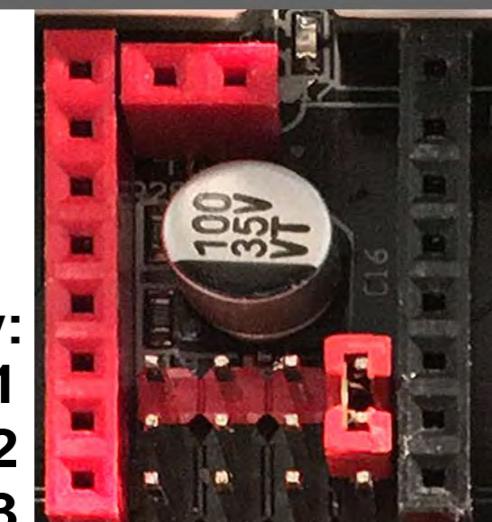
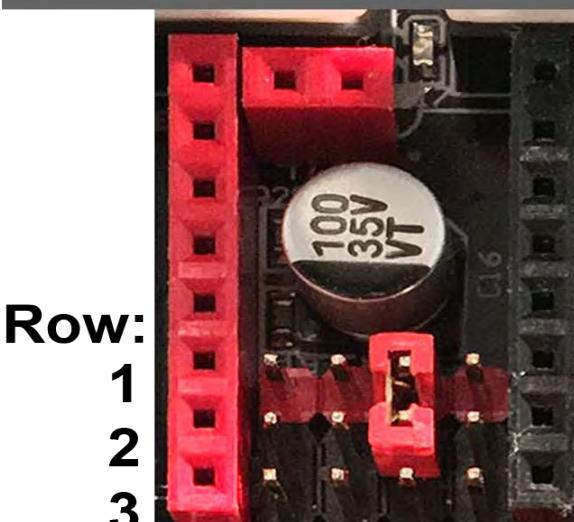
DRV8825

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows 2 and 3

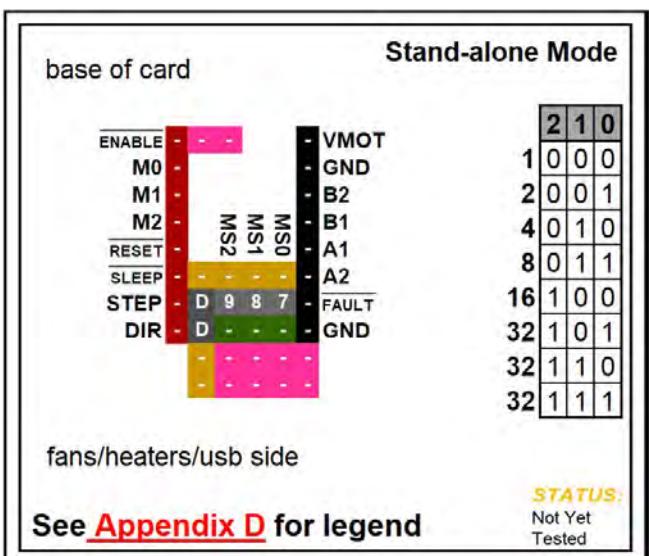


High ➡ set Jumper between rows 1 and 2



This is a Jumper:





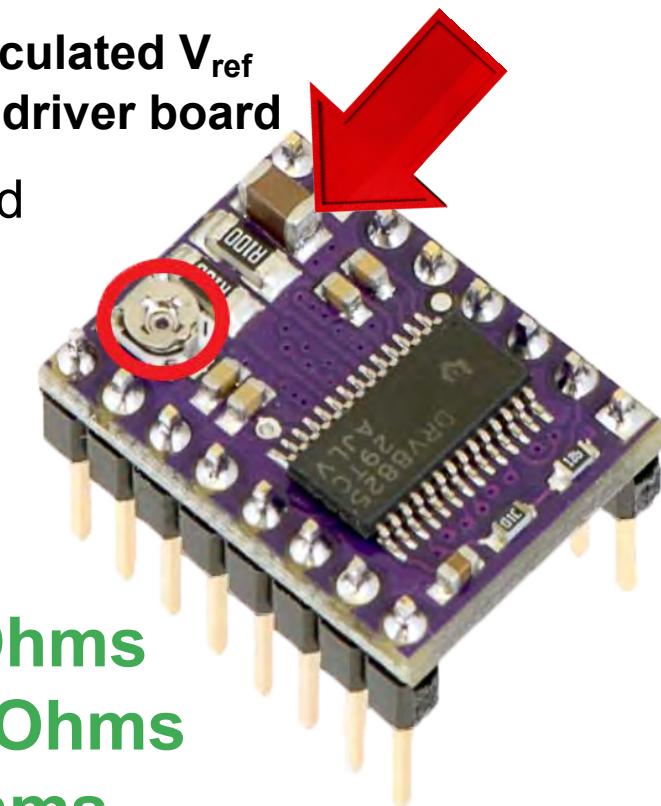
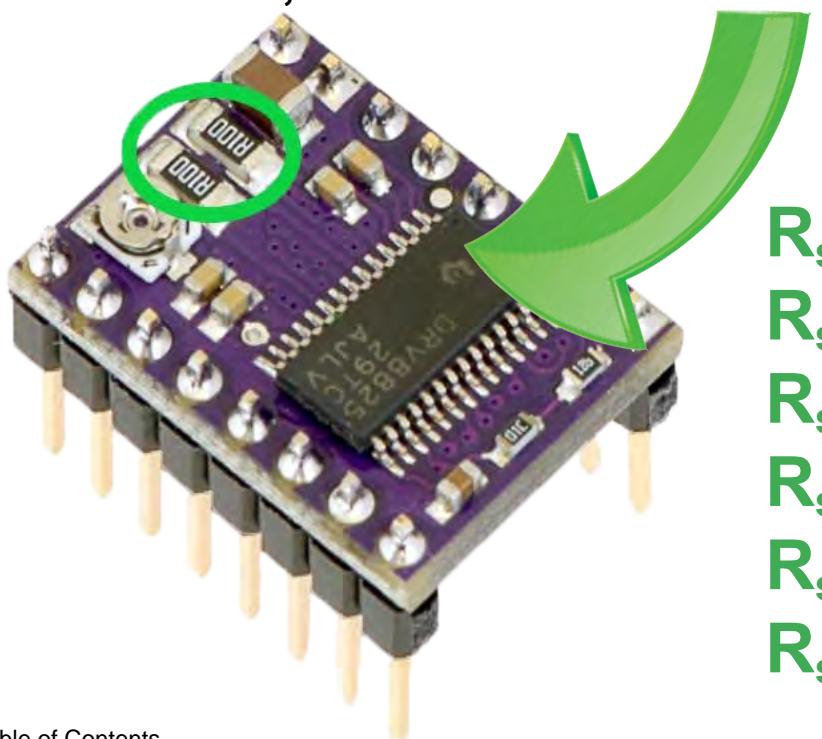
DRV8825

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

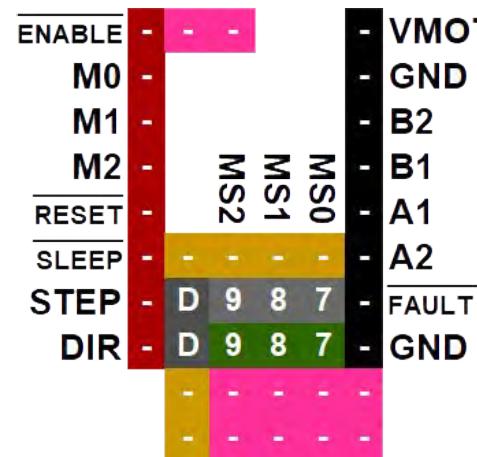
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

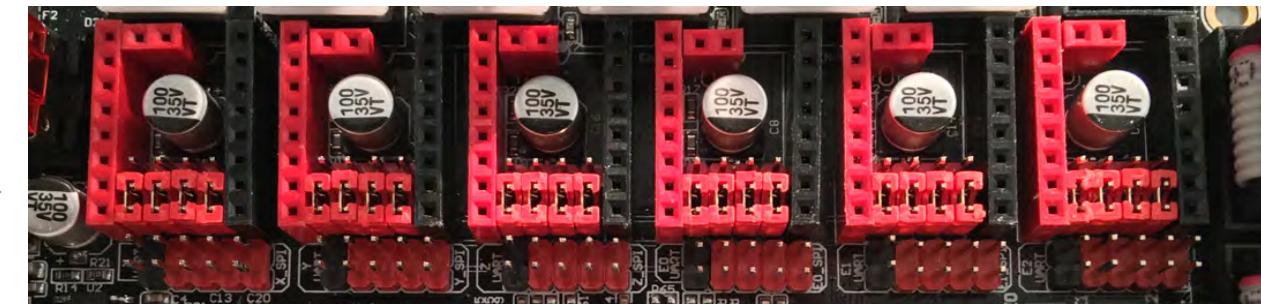
Stand-alone Mode

DRV8825

STEP

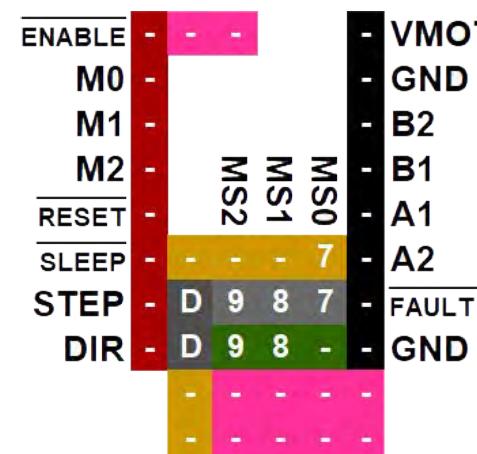


Note: The "D" jumper MUST be SET!

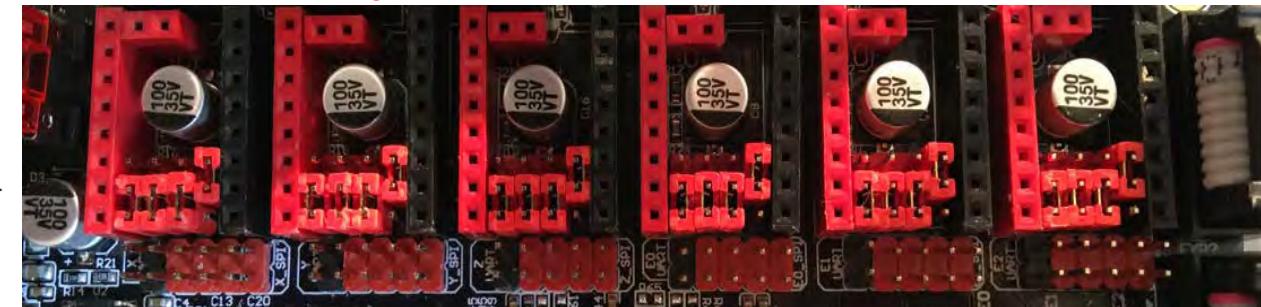


See [Appendix D](#) for legend

1 / 2



Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

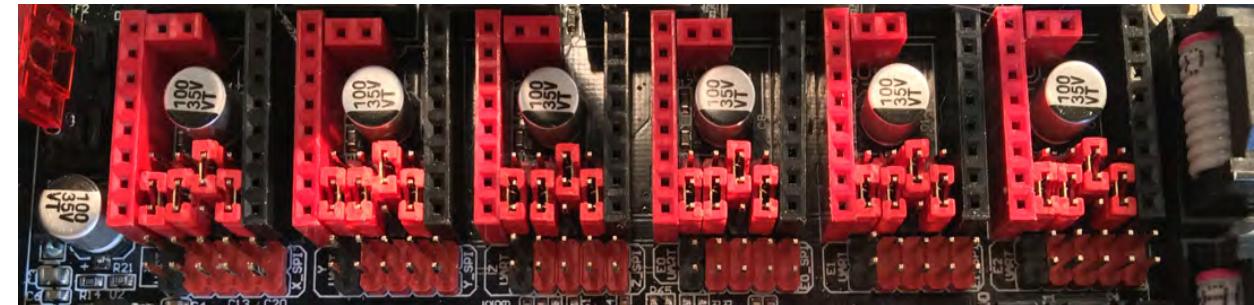
Stand-alone Mode

DRV8825

1 / 4

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
	-	-	-	-	-	-

Note: The "D" jumper MUST be SET!

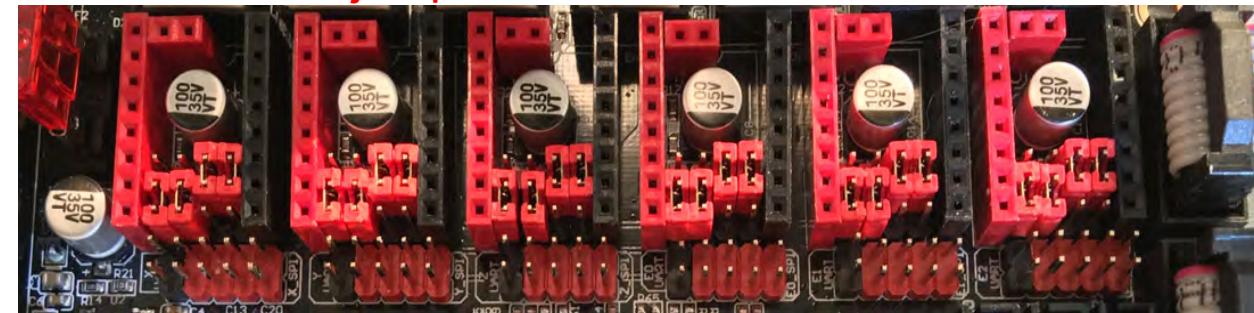


See [Appendix D](#) for legend

1 / 8

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	-	-	-	GND
	-	-	-	-	-	-

Note: The "D" jumper MUST be SET!

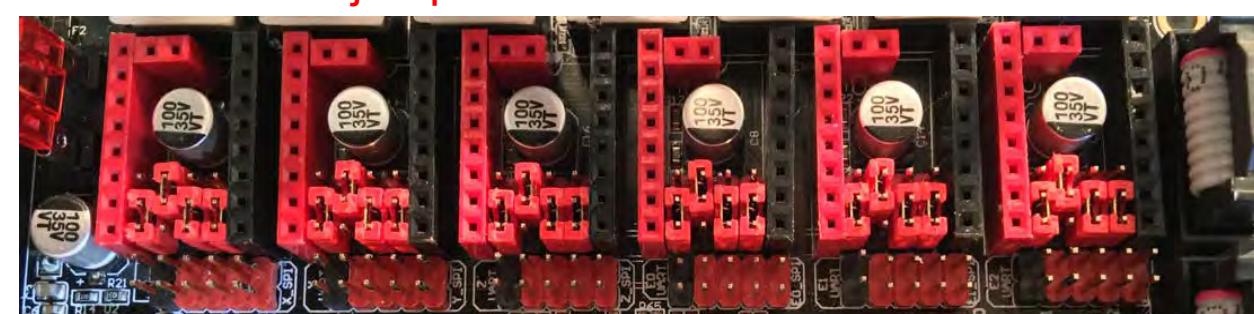


See [Appendix D](#) for legend

1 / 16

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
	-	-	-	-	-	-

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

Stand-alone Mode

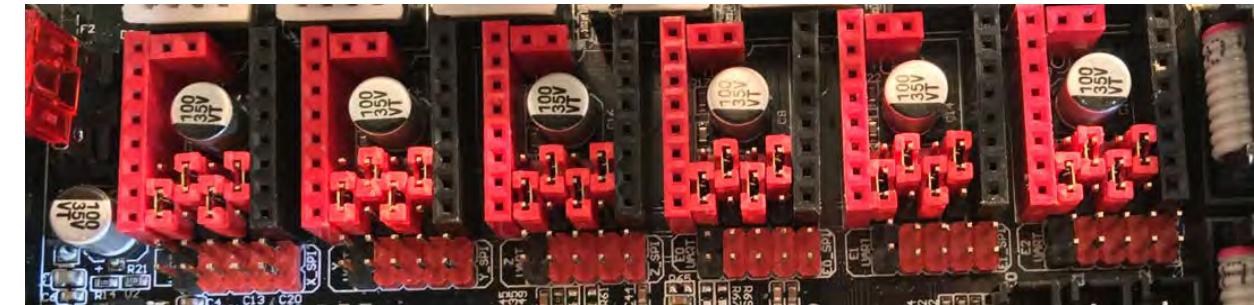
1 / 32

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

DRV8825

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

Note: The "D" jumper MUST be SET!

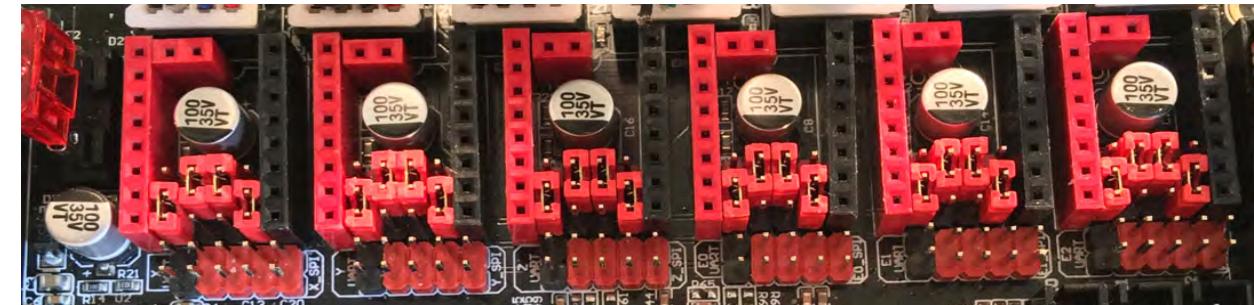


See [Appendix D](#) for legend

1 / 32

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

Note: The "D" jumper MUST be SET!

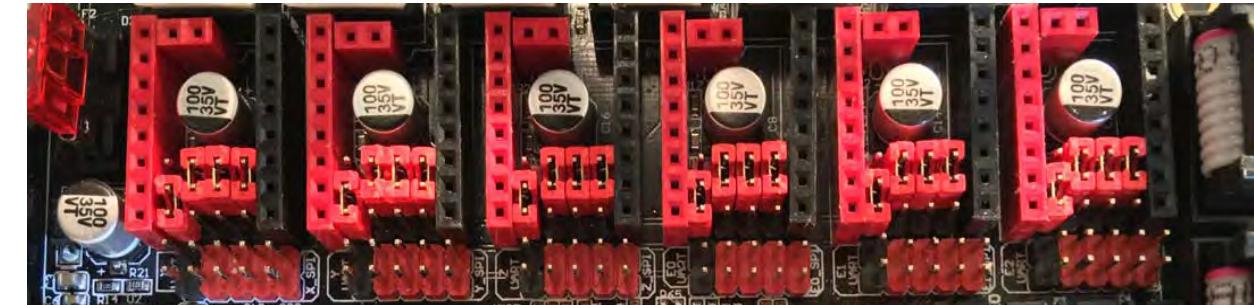


See [Appendix D](#) for legend

1 / 32

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

Note: The "D" jumper MUST be SET!

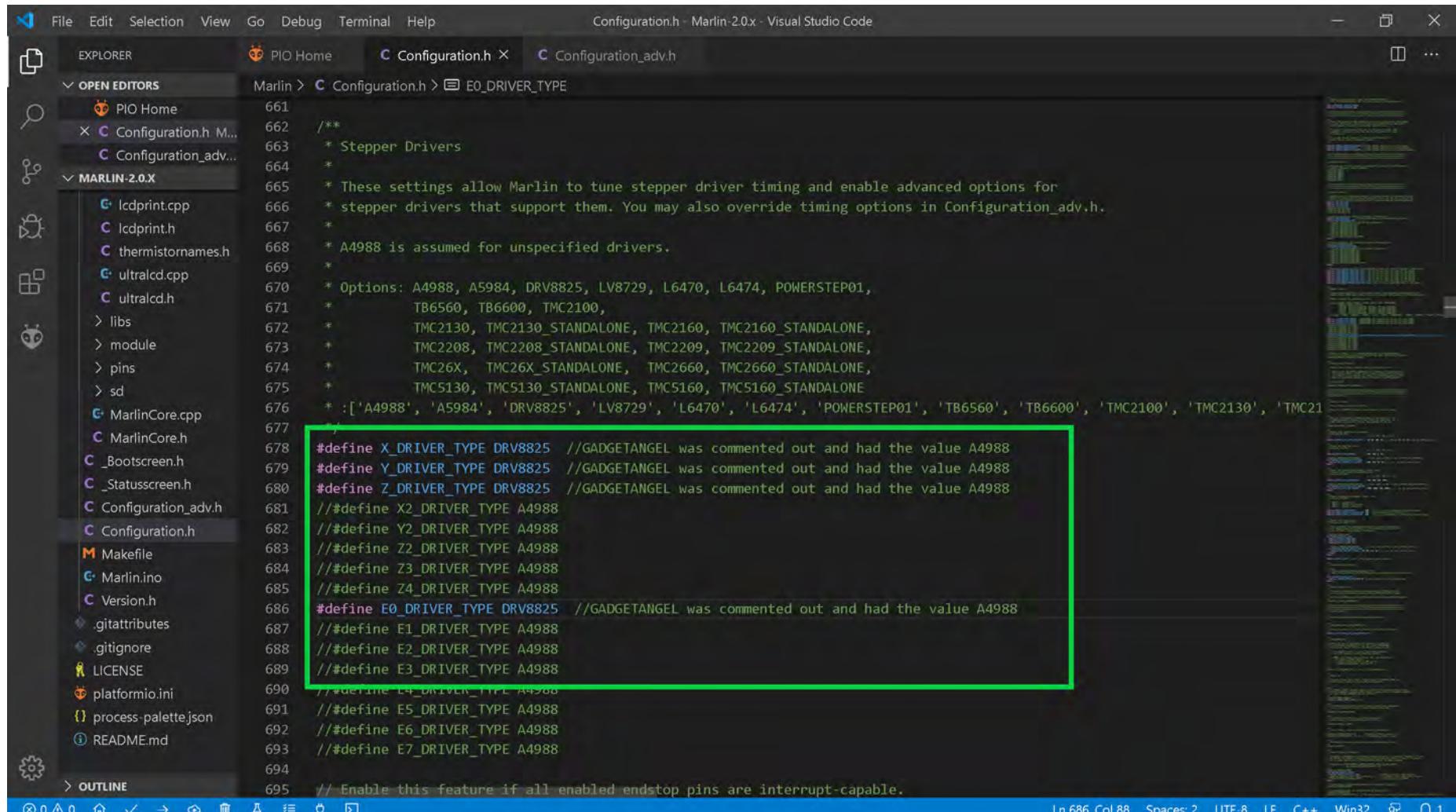


See [Appendix D](#) for legend

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.
- Title Bar:** 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, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palettejson, README.md
- Code Editor:** The code for Configuration.h is displayed, specifically the E0_DRIVER_TYPE section. A green rectangular box highlights the following lines:

```

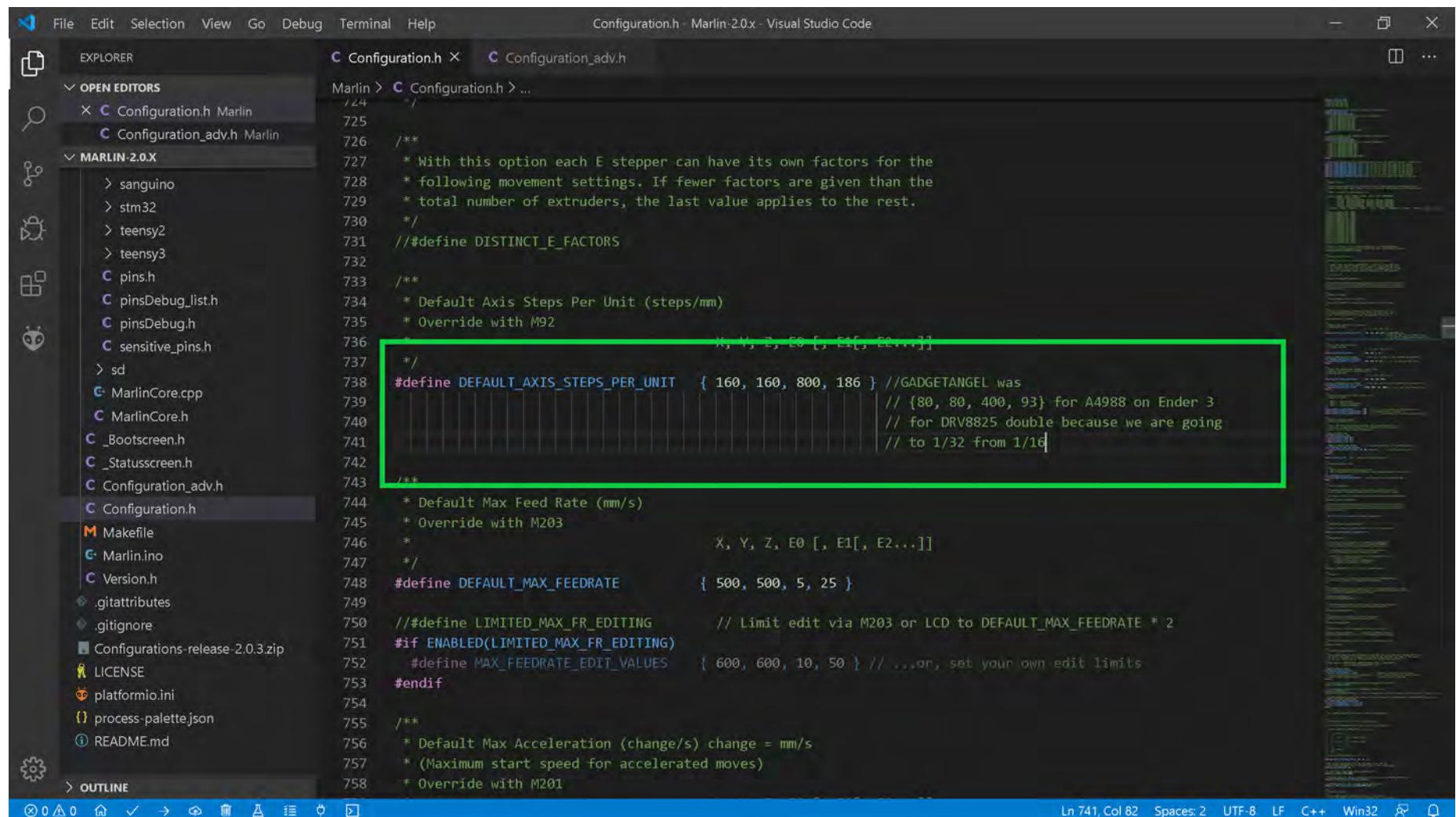
661 /**
662 * Stepper Drivers
663 */
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 */
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
677 */
678 #define X_DRIVER_TYPE DRV8825 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE DRV8825 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE DRV8825 //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 DRV8825 //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.

```
- 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 'Configuration.h' file open. The code editor displays the Marlin configuration header. A green rectangular box highlights the following line of code:

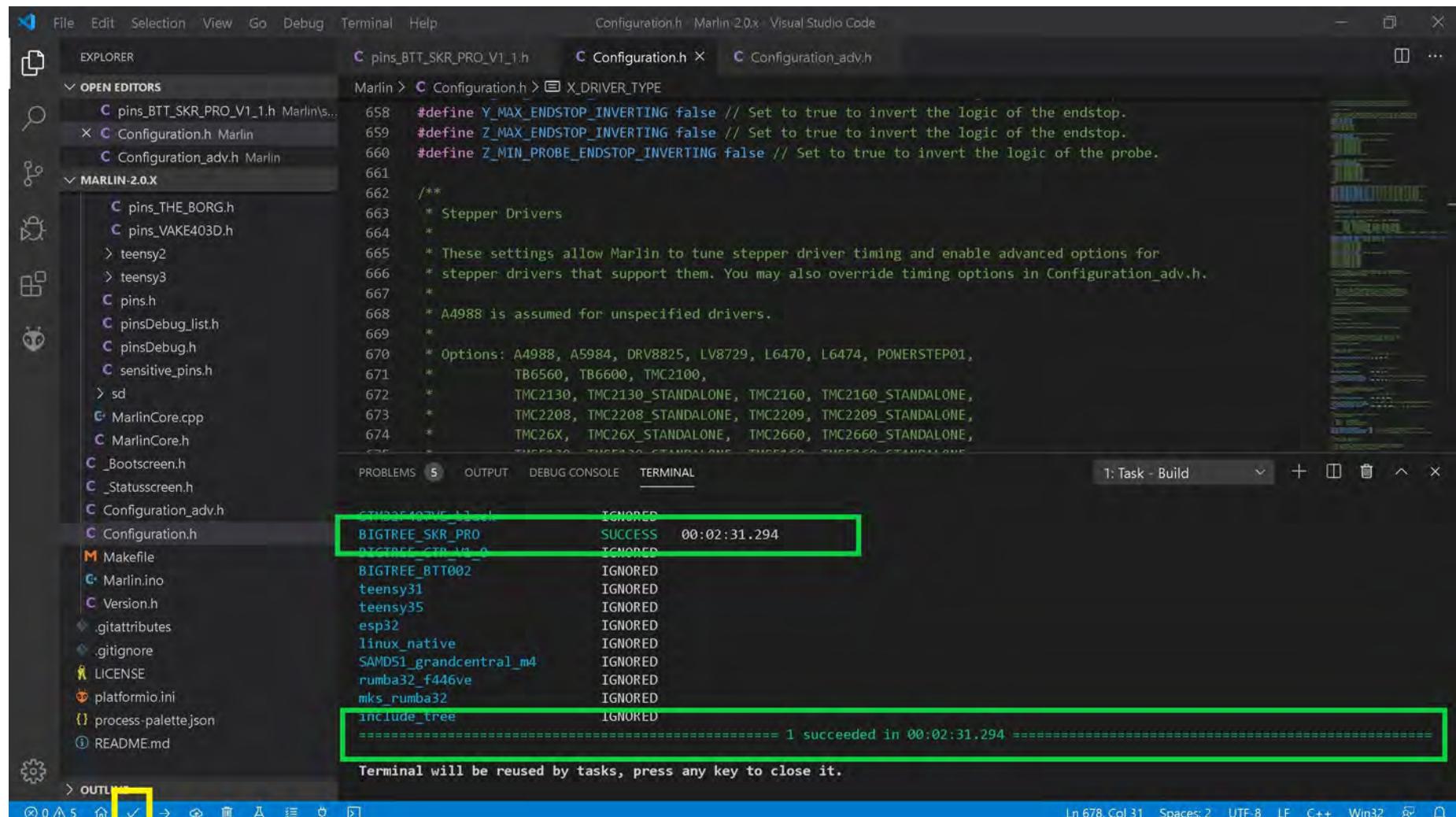
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// for DRV8825 double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom indicates the current line (Ln 741), column (Col 82), and other settings like spaces (Spaces: 2), encoding (UTF-8), 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 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.



```

Configuration.h Marlin 2.0.x Visual Studio Code

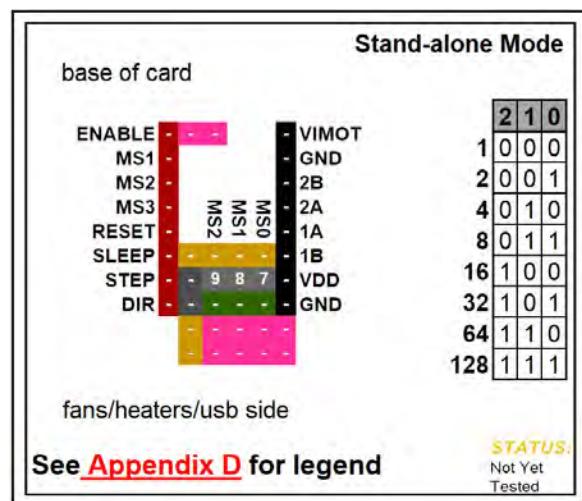
File Edit Selection View Go Debug Terminal Help Configuration.h pins_BTT_SKR_PRO_V1_1.h Configuration_adv.h
EXPLORER pins_BTT_SKR_PRO_V1_1.h Marlin\src Configuration.h Marlin Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\src\pins_BTT_SKR_PRO_V1_1.h
X 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 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + ☐ ^ ×
CTM22T40TVE_13.vcproj IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_CTR_14_v0 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 ⌂

```

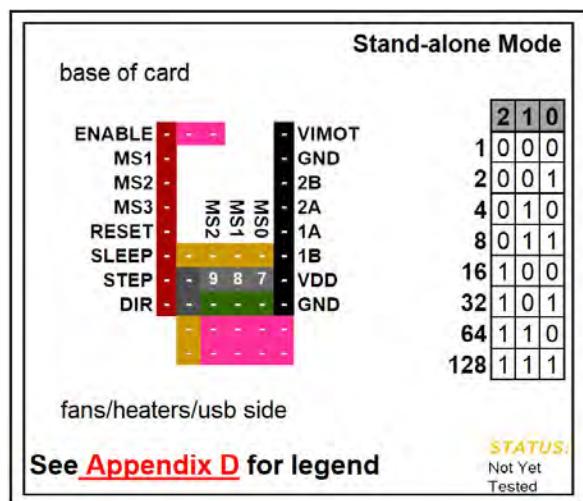
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

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
B<small>I</small>QU<small>®</small> 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	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$		
R_S (Typical Sense Resistor)=0.22Ω					

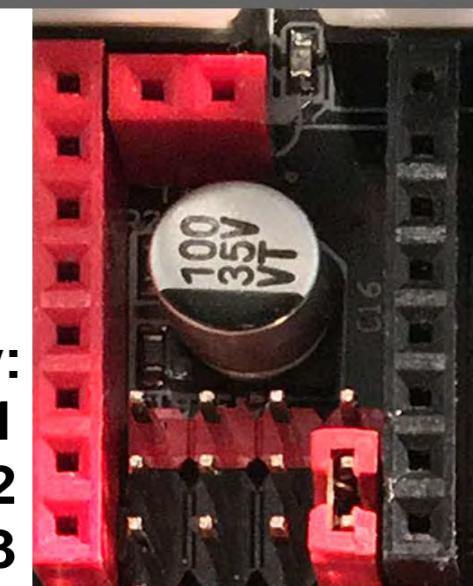
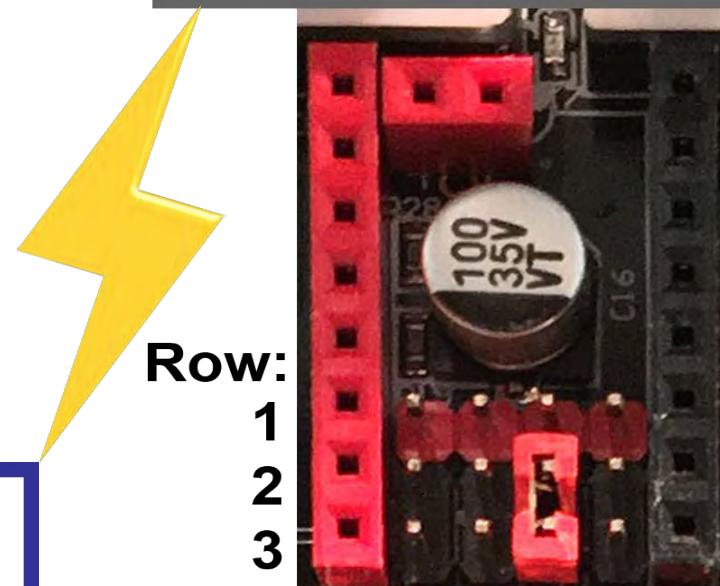
- See next page for the legend that belongs to the above chart.



BIQU LV8729

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low  **set Jumper between rows 2 and 3**

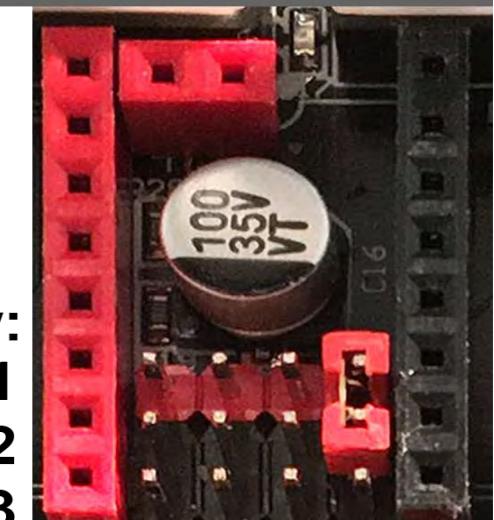
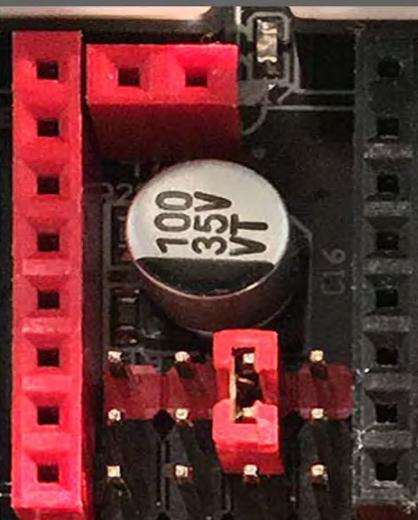


Row:
1
2
3

Row

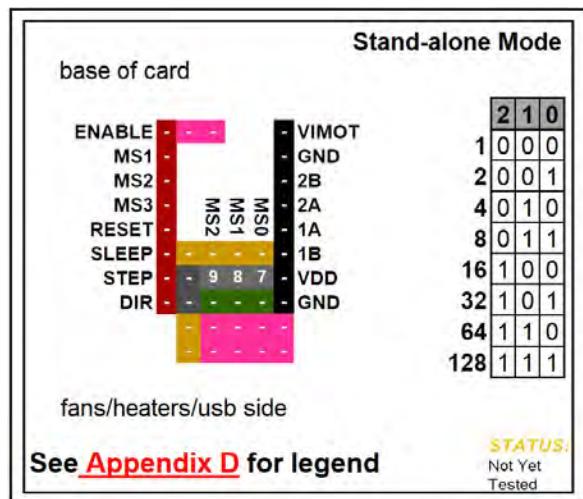
1
2
3

High ➤ set Jumper between rows
1 and 2



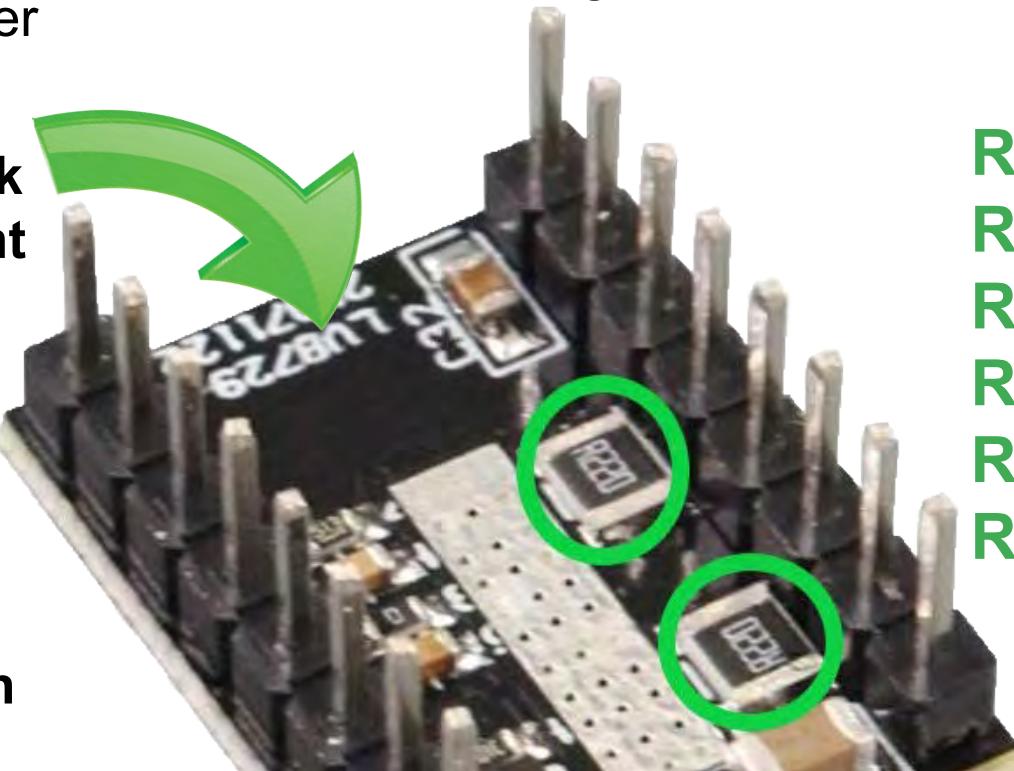
Row:
1
2
3

Row
1
2
3



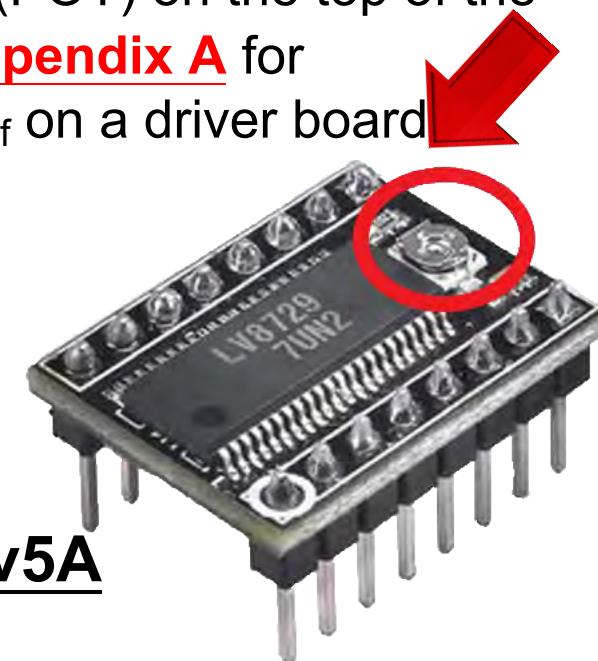
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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>

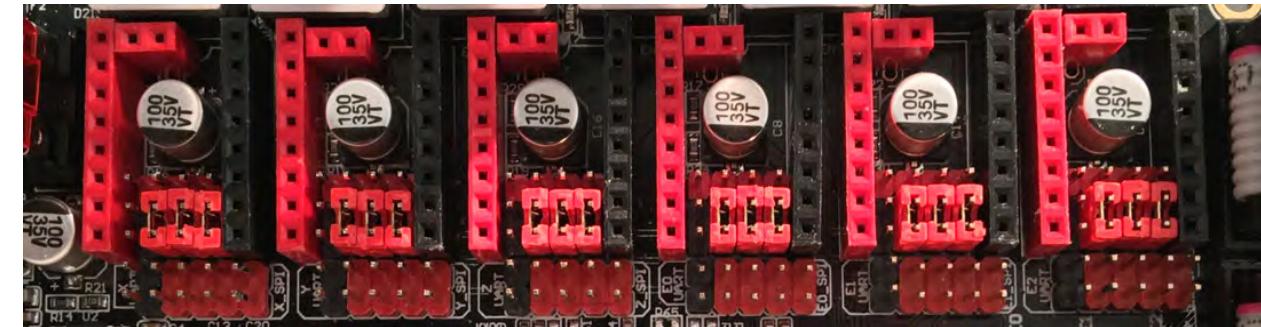
$R_s = R050$ is 0.05 Ohms
 $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms
 $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms
 $R_s = R220$ is 0.22 Ohms

Stand-alone Mode

BIQU LV8729

STEP

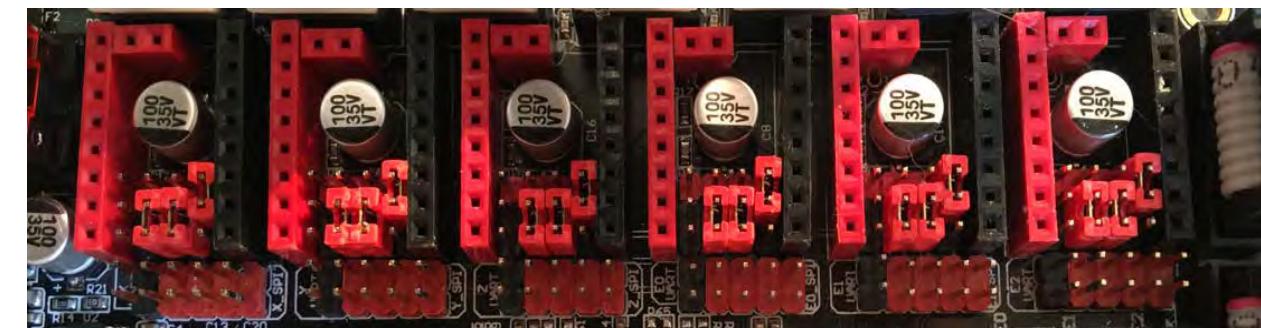
ENABLE	-	-	-	VIMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	-	-	2A	
RESET	-	MS2	MS1	MS0	1A
SLEEP	-	-	-	-	1B
STEP	-	9	8	7	VDD
DIR	-	9	8	7	GND



See [Appendix D](#) for legend

1 / 2

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

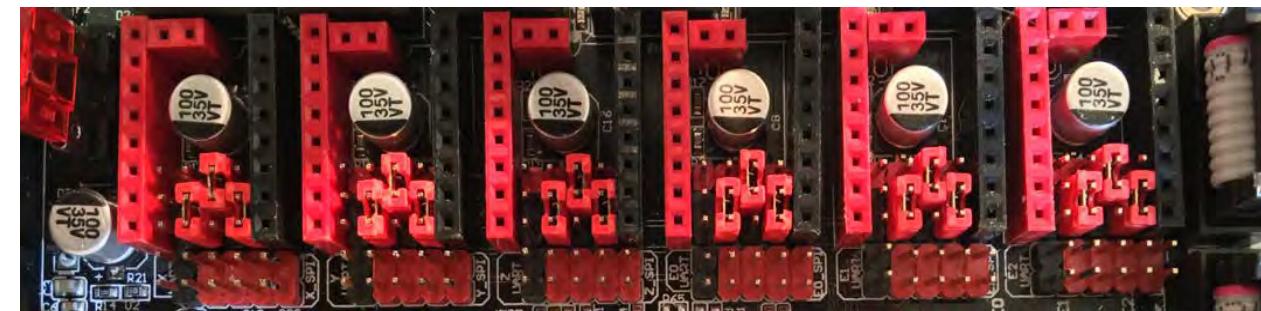


See [Appendix D](#) for legend

Stand-alone Mode

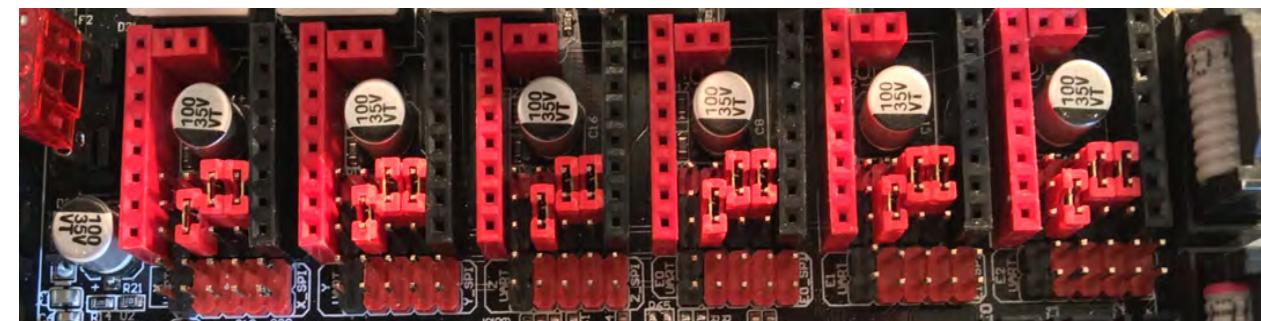
BIQU LV8729

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



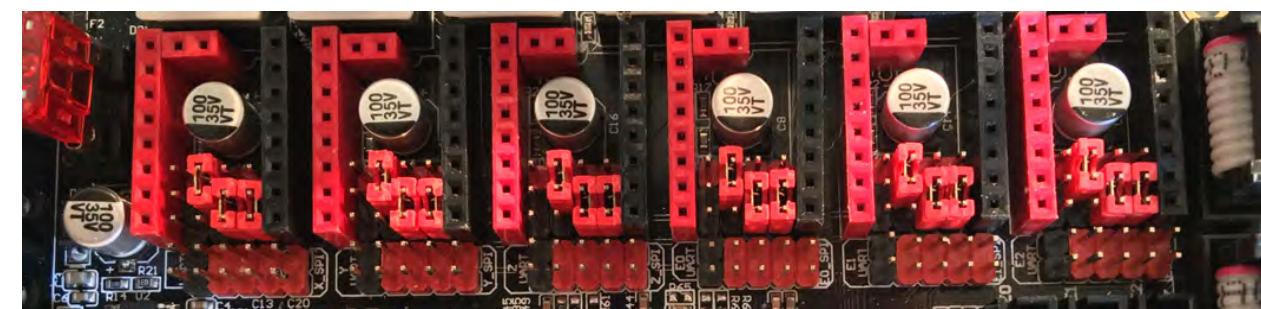
See [Appendix D](#) for legend

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



See [Appendix D](#) for legend

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



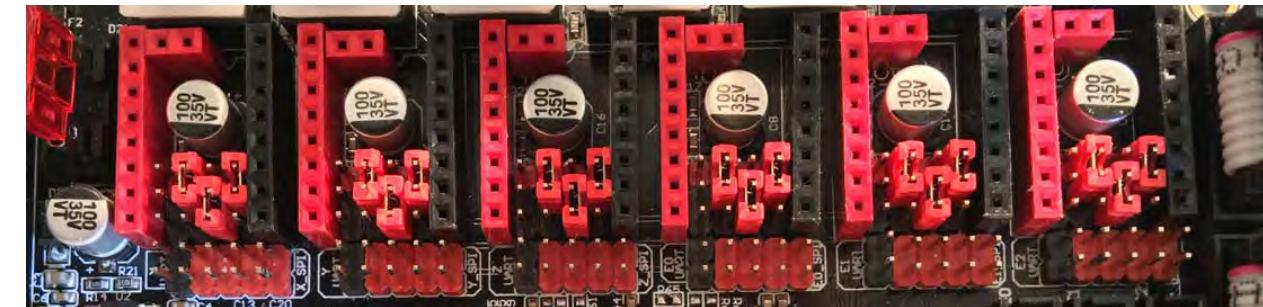
See [Appendix D](#) for legend

Stand-alone Mode

BIQU LV8729

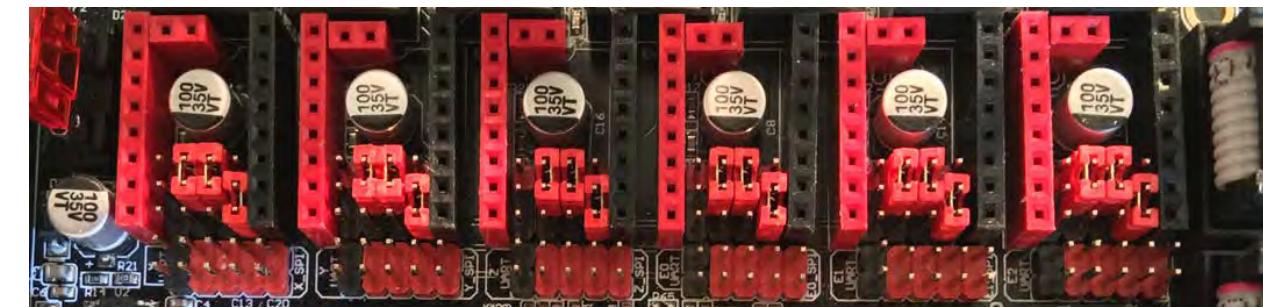
1 / 32

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

See [Appendix D](#) for legend

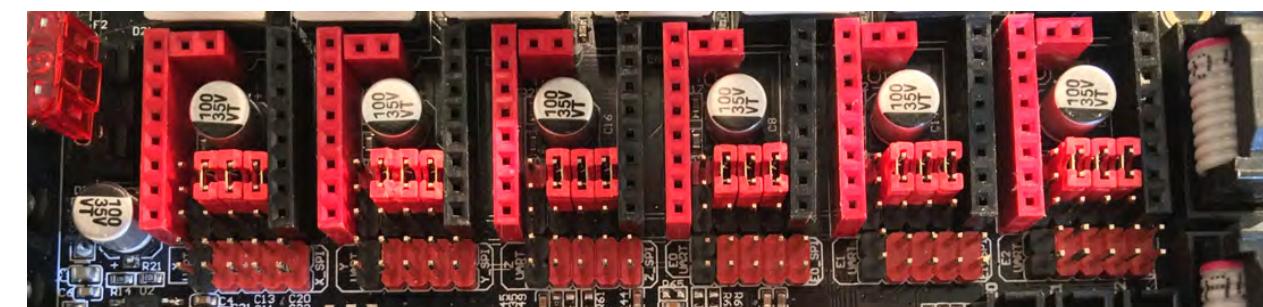
1 / 64

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

See [Appendix D](#) for legend

1 / 128

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

See [Appendix D](#) for legend

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

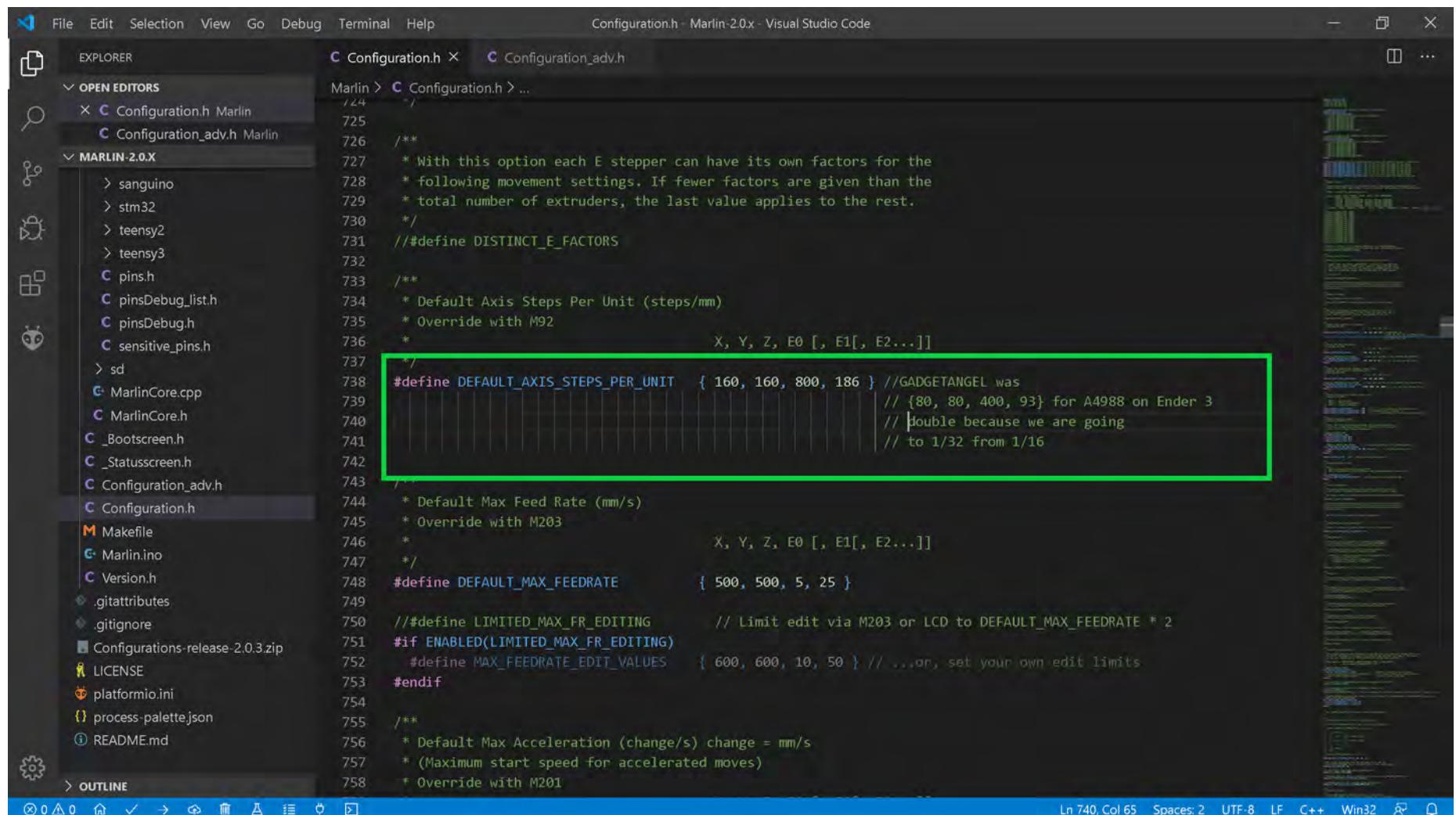
  661
  662 /**
  663 * Stepper Drivers
  664 *
  665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  667 *
  668 * A4988 is assumed for unspecified drivers.
  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', 'TMC2
  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
  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 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:

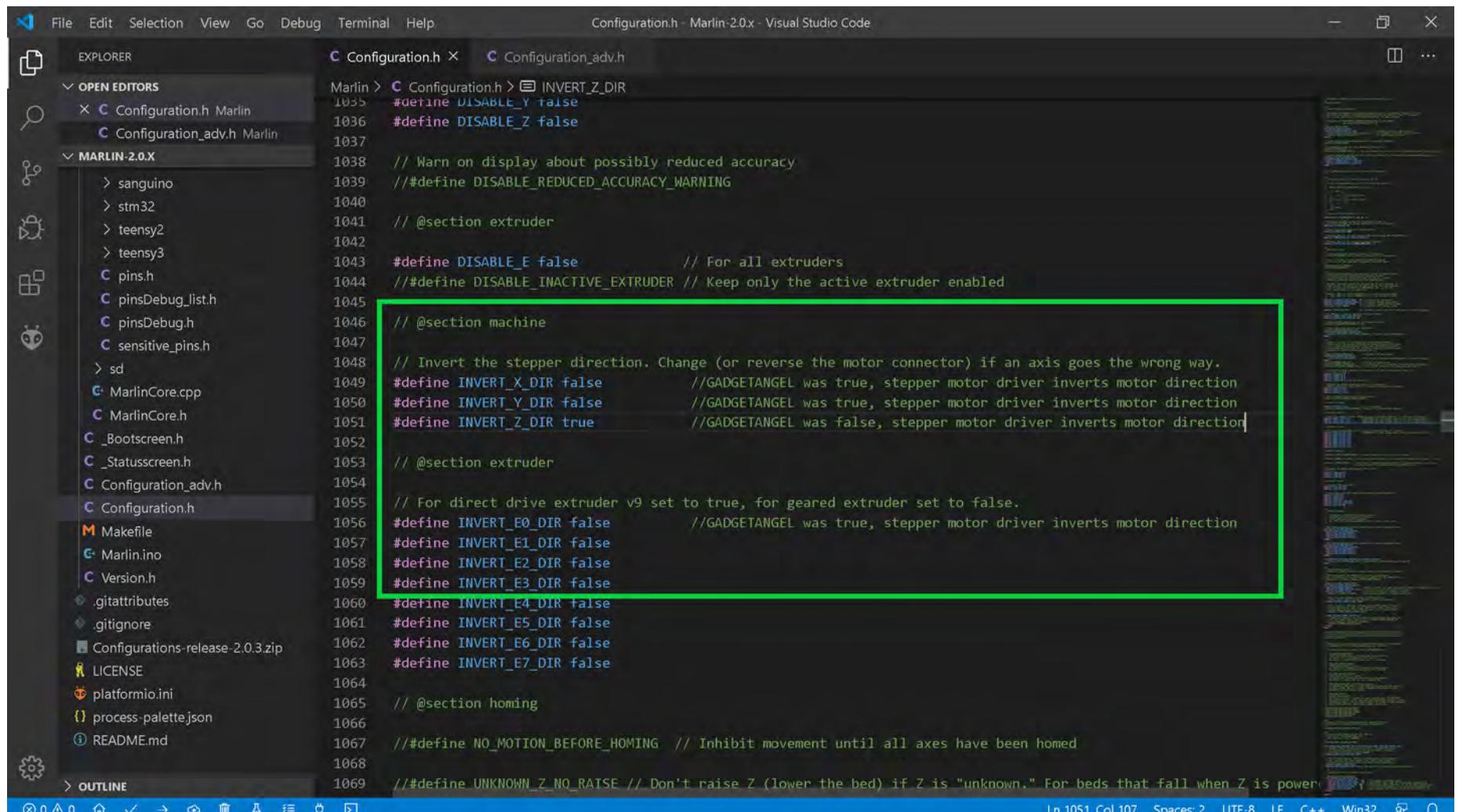
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



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

```

Marlin > C Configuration.h > C Configuration_adv.h
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

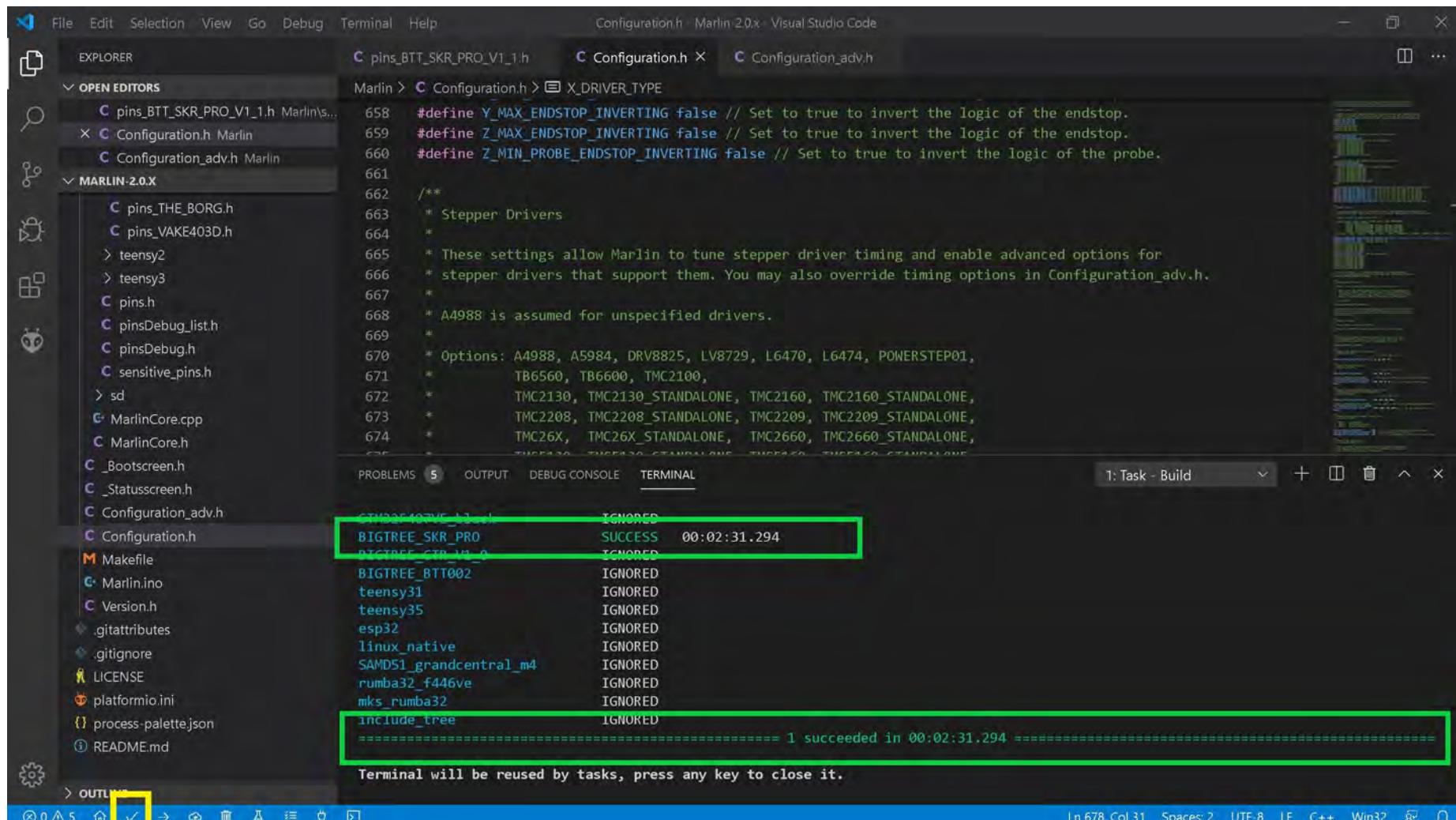
```

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

- 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".
- Editor:** Displays the `Configuration.h` file with code related to endstop inversion and stepper driver timing.
- Terminal:** Shows the build process output. A yellow box highlights the checkmark icon in the status bar at the bottom left. A green box highlights the terminal output which shows a successful build message: "1 succeeded in 00:02:31.294".

```

#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.

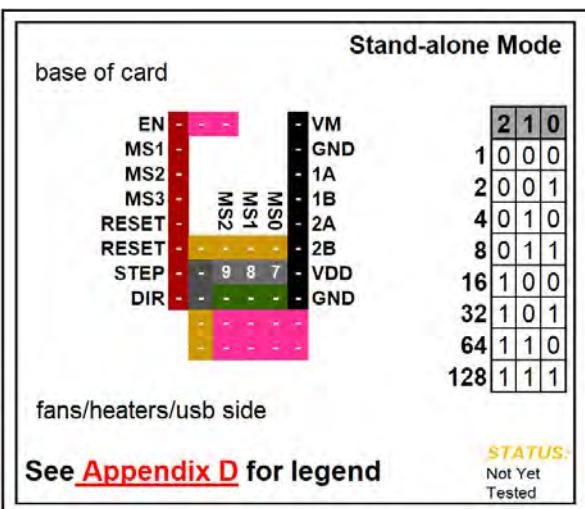
/*
 * Stepper Drivers
 *
 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 *
 * A4988 is assumed for unspecified drivers.
 *
 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 *          TB6560, TB6600, TMC2100,
 *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 *          TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
 */

BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_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 =====

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

```

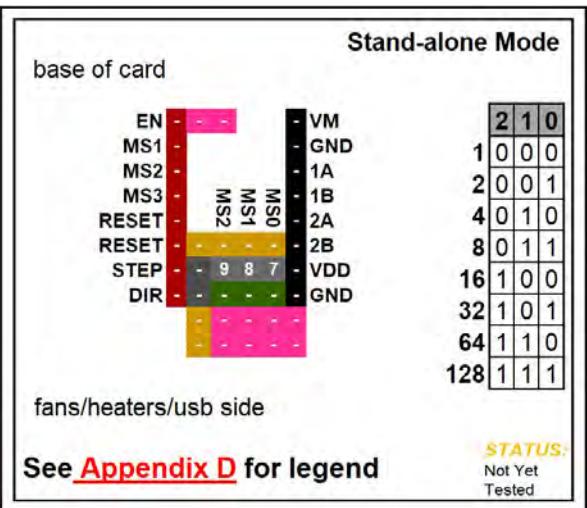
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

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	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$			$V_{ref} = 5 * I_{MAX} * R_S$	
R_S (Typical Sense Resistor) = 0.22Ω					

- See next page for the legend that belongs to the above chart.



FYSETC LV8729

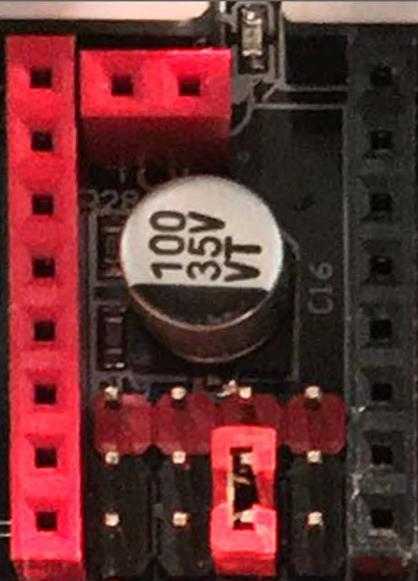
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➤ set Jumper between rows 2 and 3



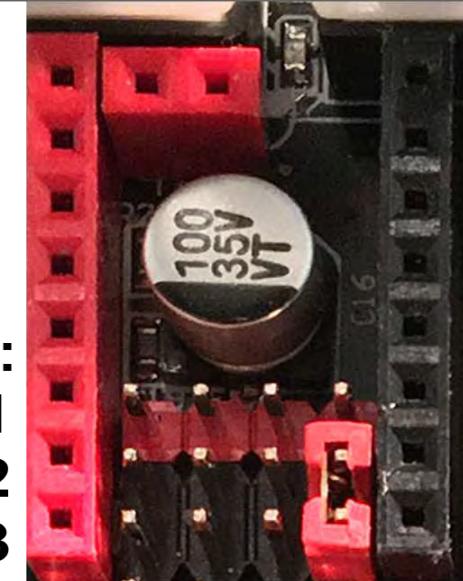
Row:

1
2
3



Row:

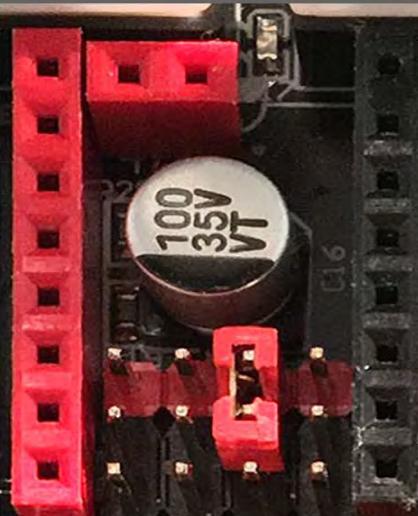
1
2
3



High ➤ set Jumper between rows 1 and 2

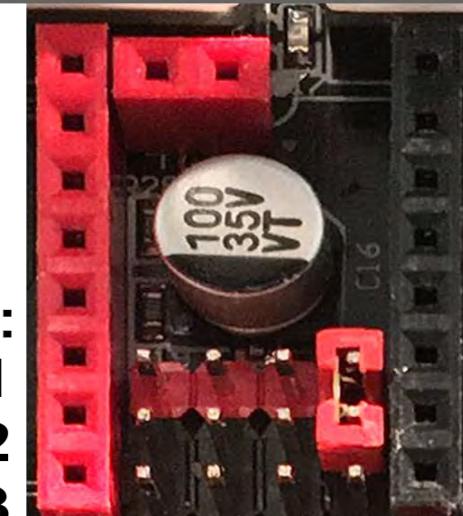
Row:

1
2
3



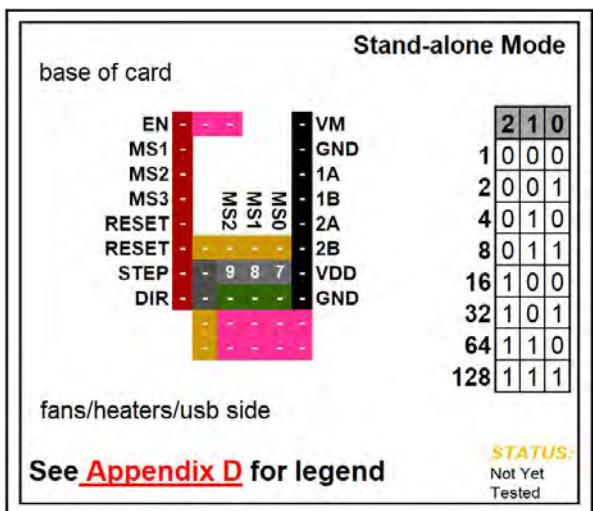
Row:

1
2
3

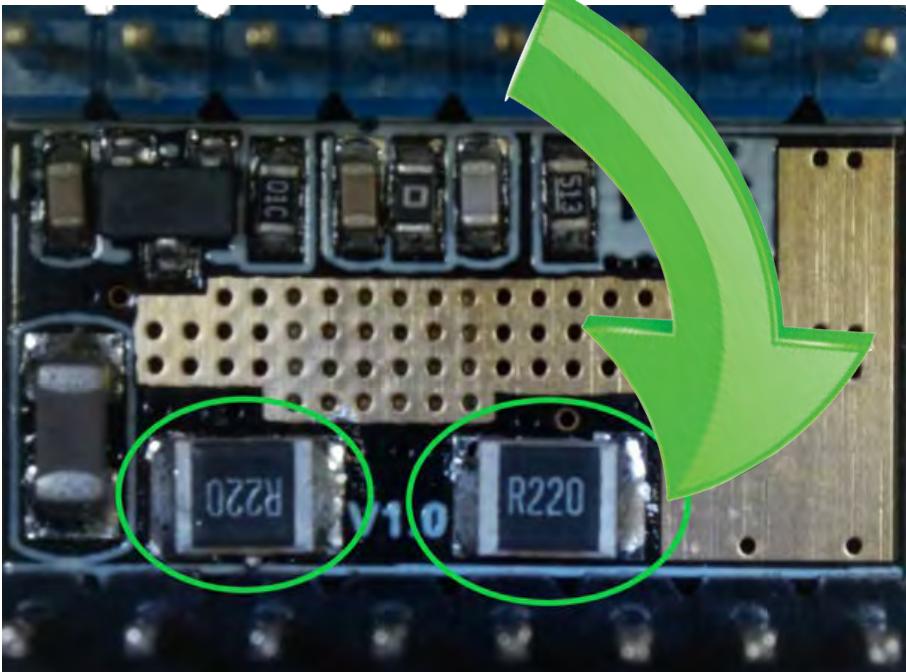


This is a Jumper:



FYSETC LV8729

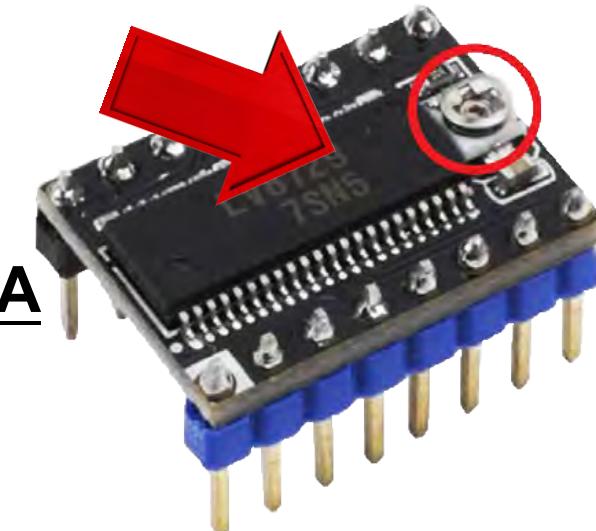
Note: Check your current sense resistors (R_s) values on the driver board, as shown in **GREEN**



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>



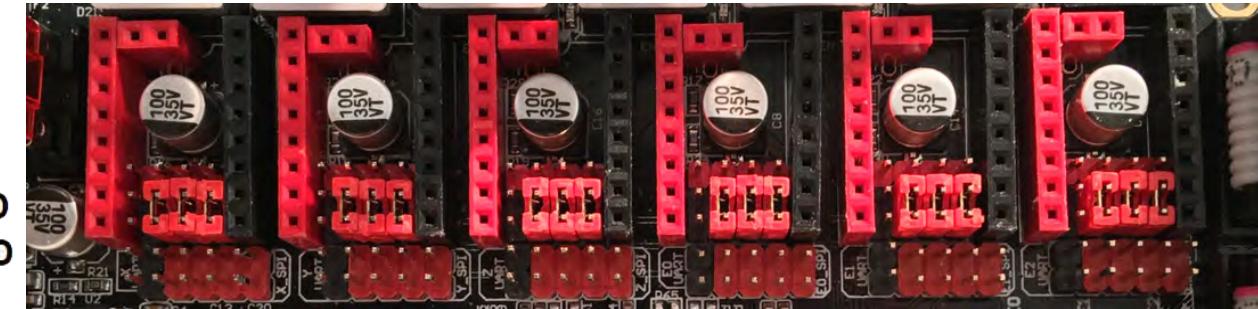
$R_s = R050$ is 0.05 Ohms;
 $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms;
 $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms;
 $R_s = R220$ is 0.22 Ohms

Stand-alone Mode

FYSETC LV8729

STEP

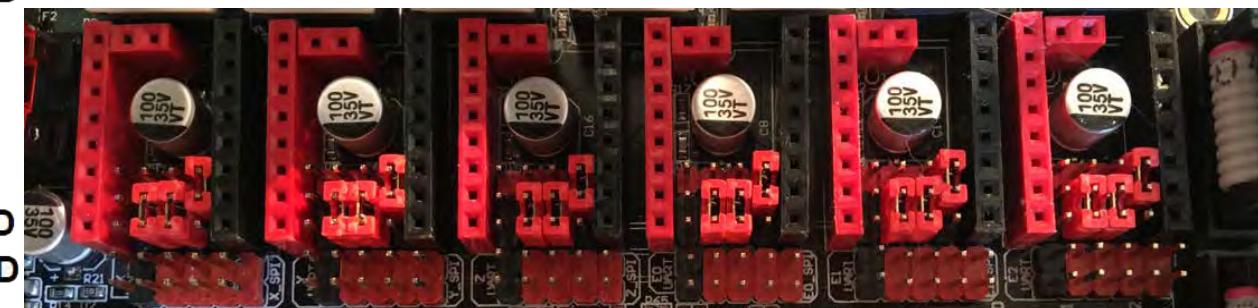
EN	-	-	-	-	VM
MS1	-	-	-	-	GND
MS2	-	-	-	-	1A
MS3	-	-	-	-	1B
RESET	-	-	MS2	MS1	MS0
RESET	-	-	-	-	2A
STEP	-	9	8	7	-
DIR	-	9	8	7	-
	-	-	-	-	VDD
	-	-	-	-	GND



See [Appendix D](#) for legend

1 / 2

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



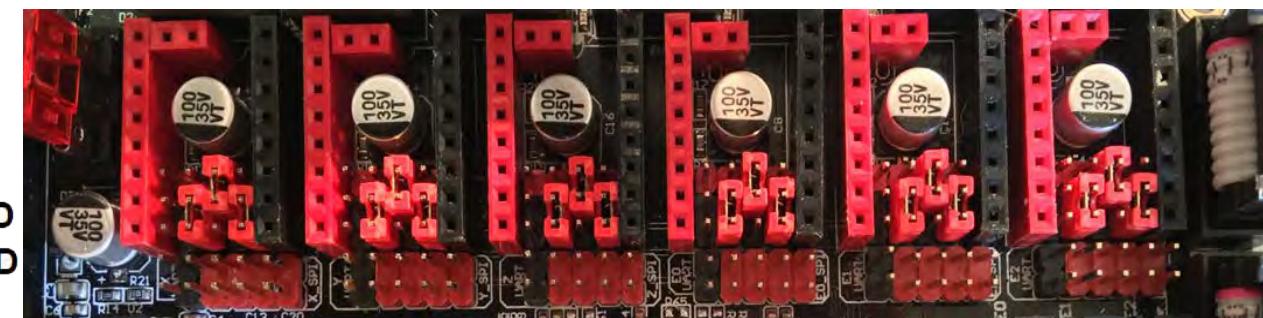
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC LV8729

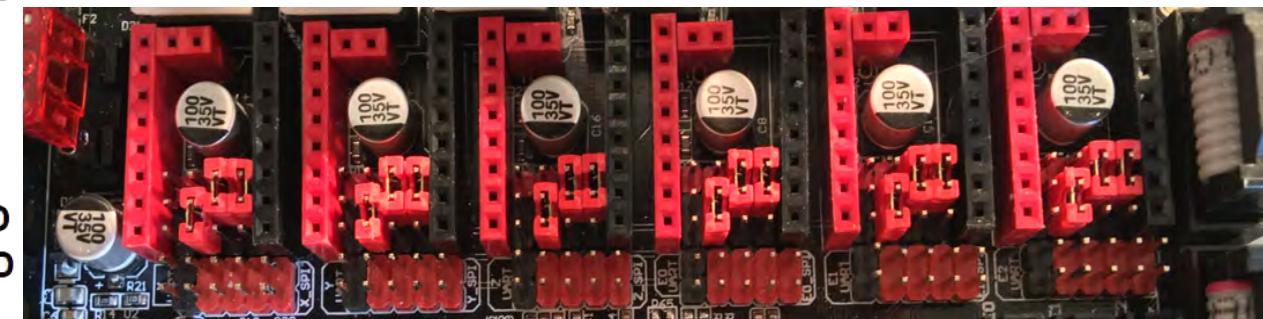
1 / 4

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

See [Appendix D](#) for legend

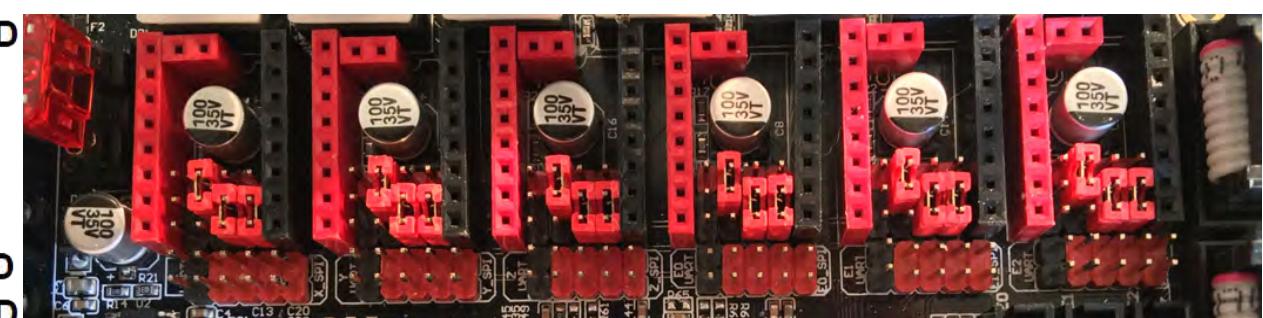
1 / 8

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

See [Appendix D](#) for legend

1 / 16

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

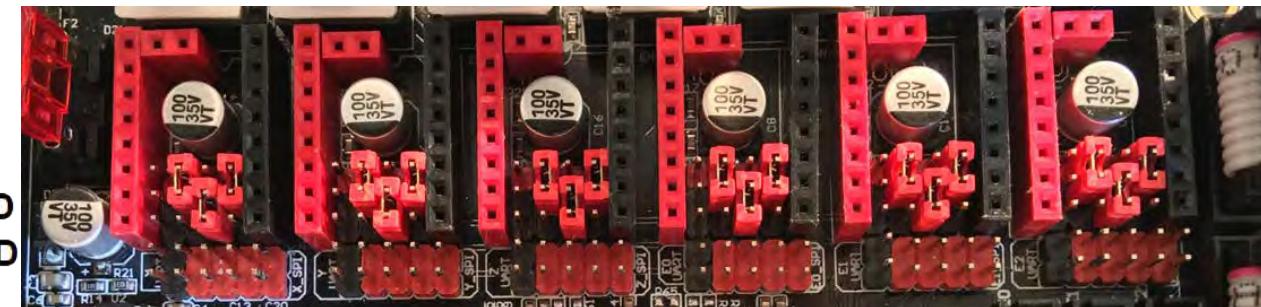
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC LV8729

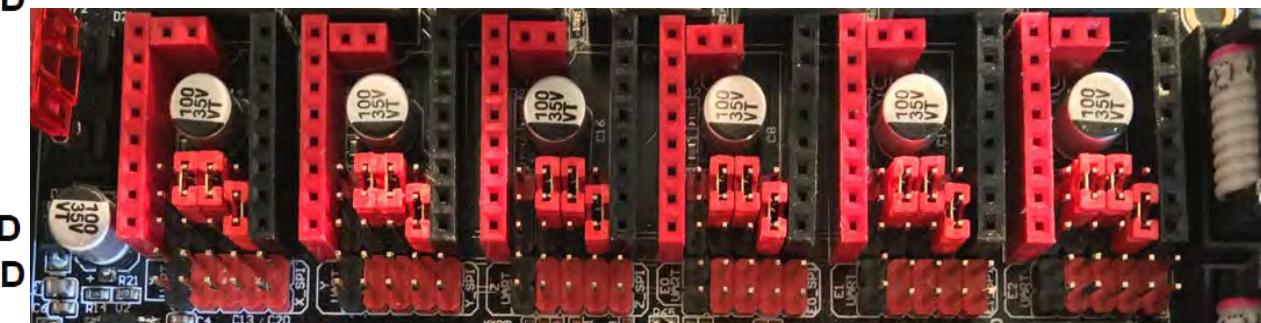
1 / 32

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

See [Appendix D](#) for legend

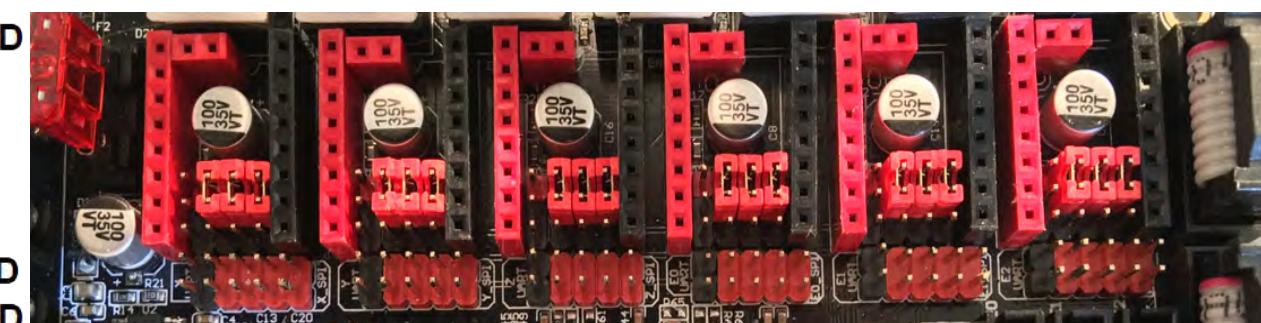
1 / 64

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

See [Appendix D](#) for legend

1 / 128

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

See [Appendix D](#) for legend

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 ("//").

The screenshot shows the Visual Studio Code interface with the file `Configuration.h` 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', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
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
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

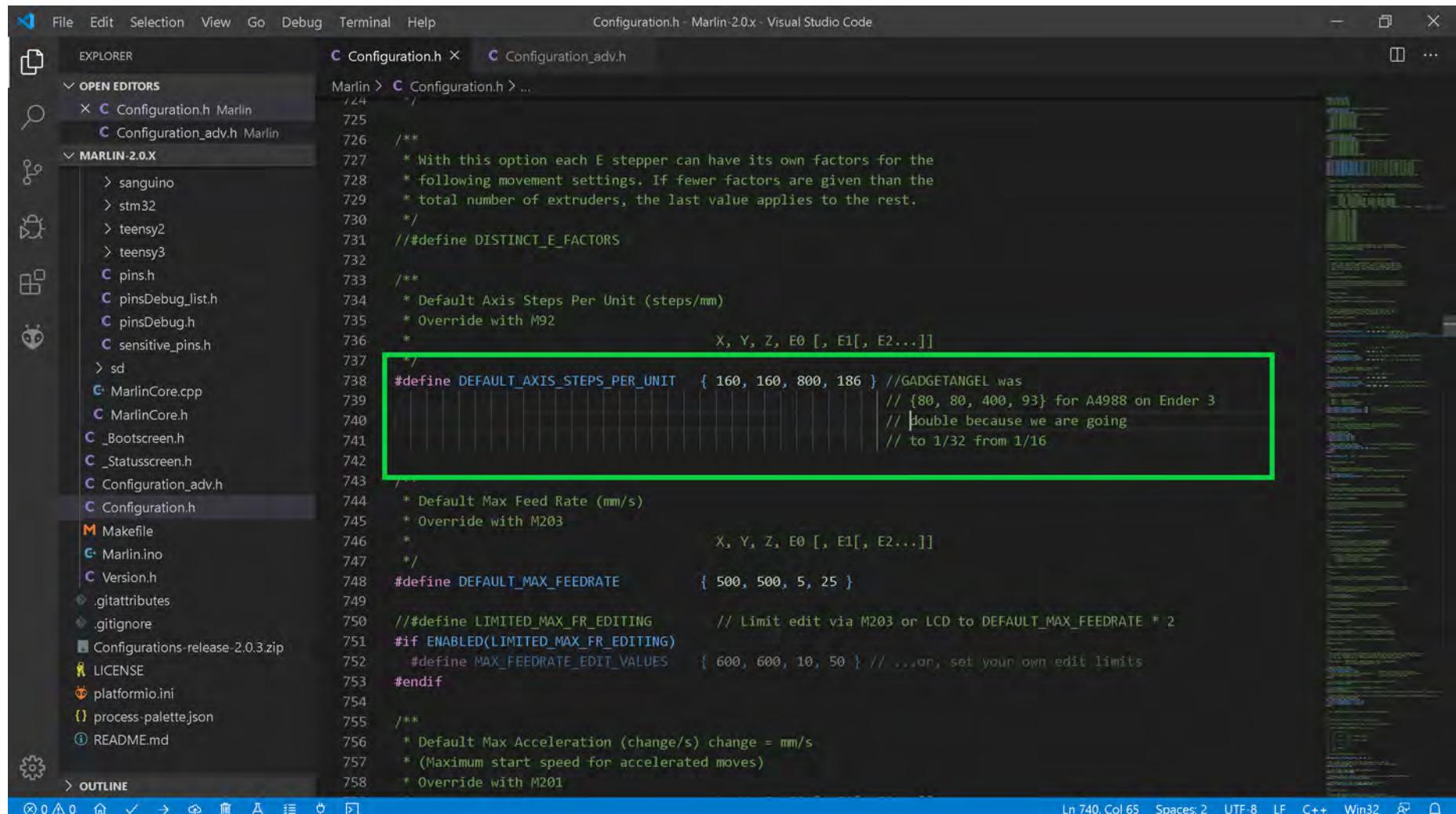
```

The code editor has a green box highlighting the driver type definitions for all axes (X, Y, Z, E0-E7). The status bar at the bottom right shows: Ln 686, Col 87 Spaces: 2 UTF-8 LF C++ Win32 1

- 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 configuration header. A green rectangular box highlights the following line of code:

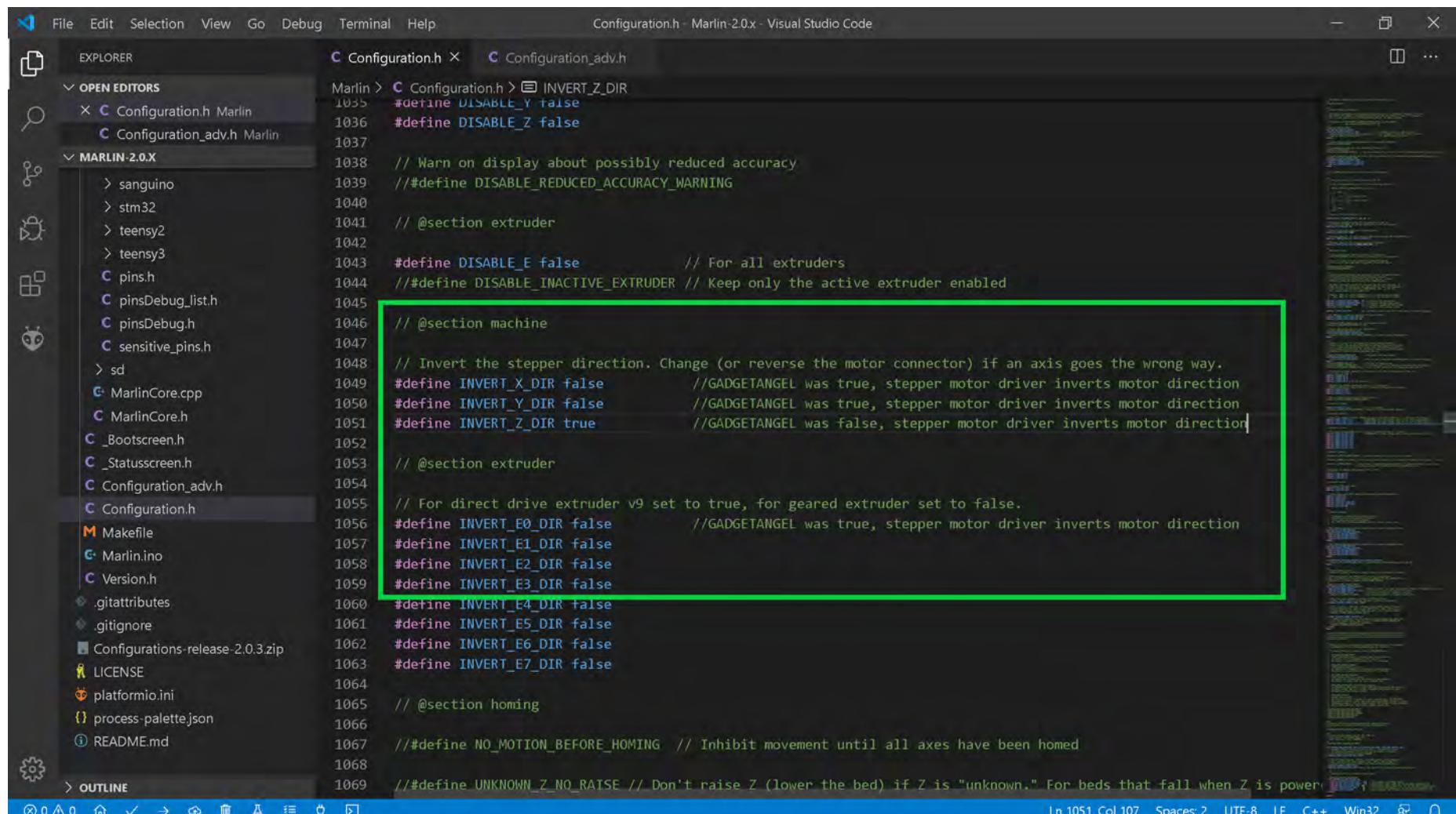
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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 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 file `Configuration.h` open. The code editor displays the following snippet:

```

Marlin > C Configuration.h > #define 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

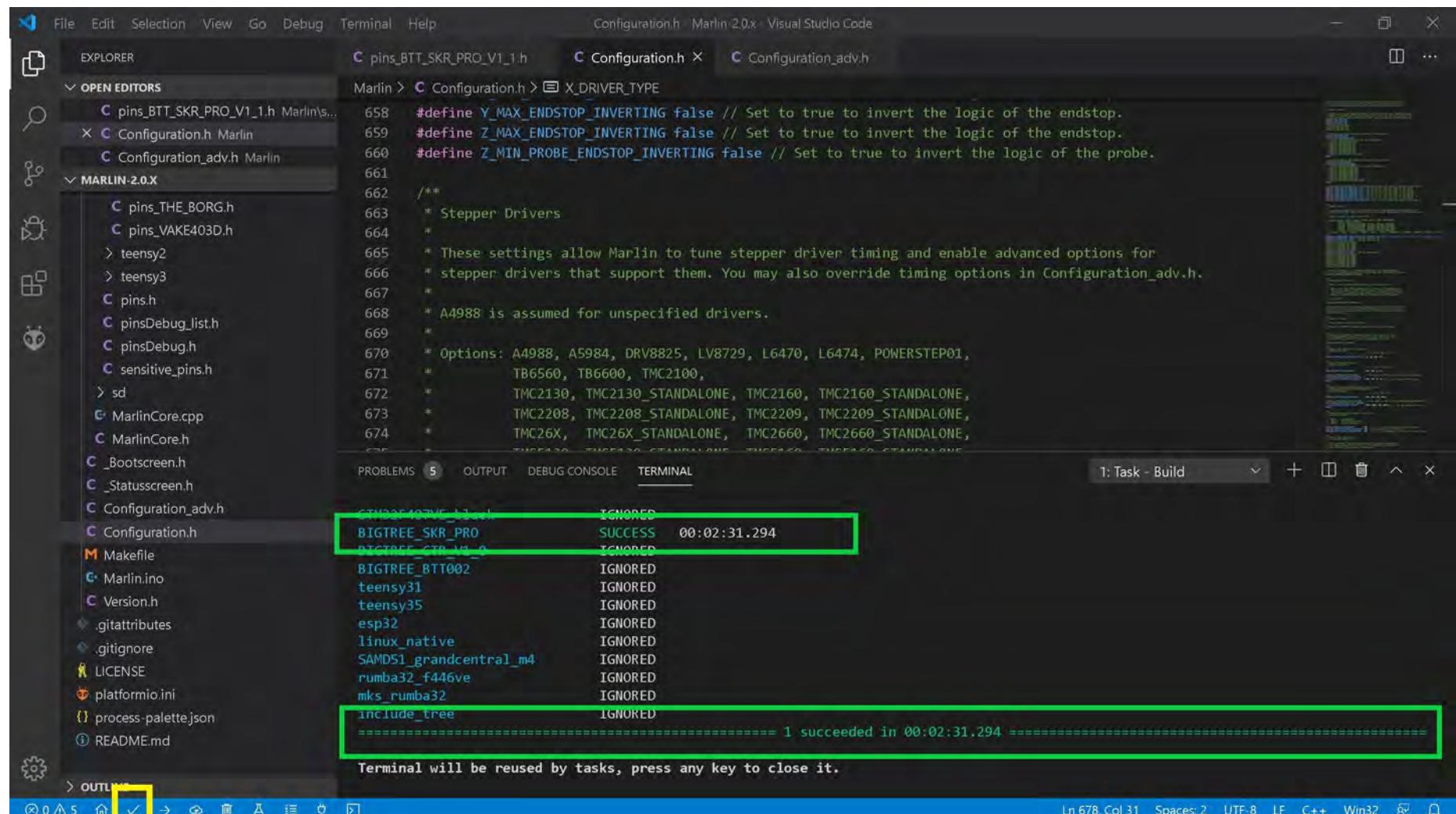
```

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

- 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.



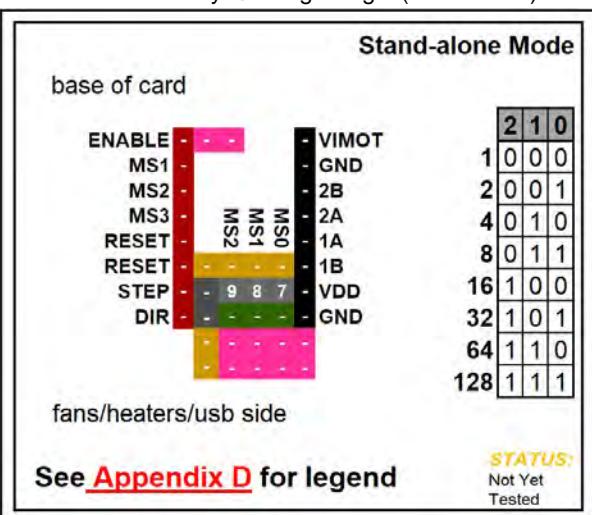
The screenshot shows the Visual Studio Code interface with the Marlin repository open. The left sidebar shows files like pins_BTT_SKR_PRO_V1_1.h, Configuration.h, and Configuration_adv.h. The main editor window displays Configuration.h with code related to stepper drivers. The terminal tab at the bottom shows a build process:

```

    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 =====
  
```

A yellow box highlights the checkmark icon in the bottom-left corner of the terminal bar, indicating the build was successful. A green box highlights the terminal output showing the build results for various boards.

- 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 [Appendix E](#).

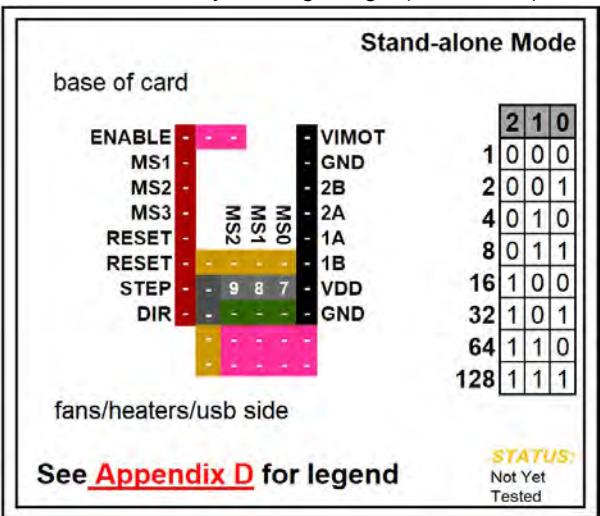


LERDGE LV8729

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	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$		
R_S (Typical Sense Resistor) = 0.22Ω					

- See next page for the legend that belongs to the above chart.



This is a
Jumper:



LERDGE LV8729

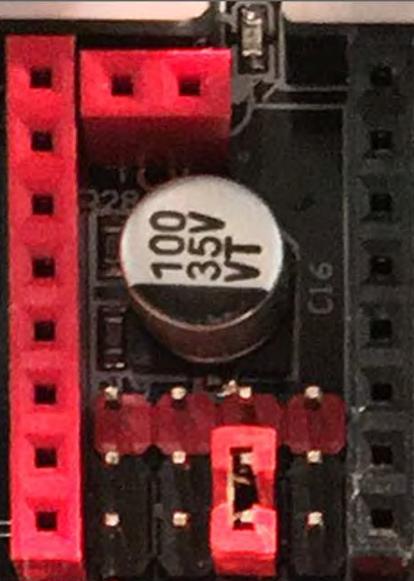
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➤ set Jumper between rows
2 and 3



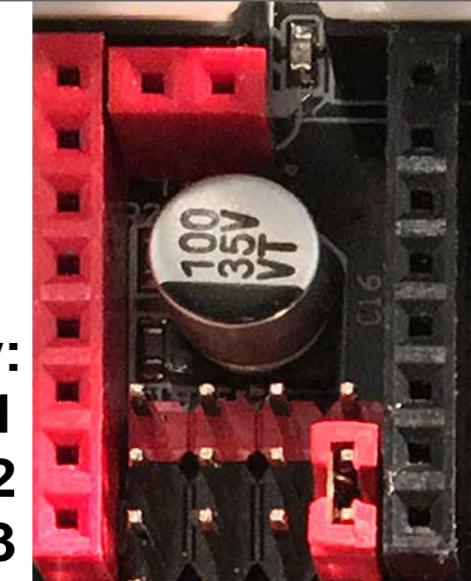
Row:

1
2
3



Row:

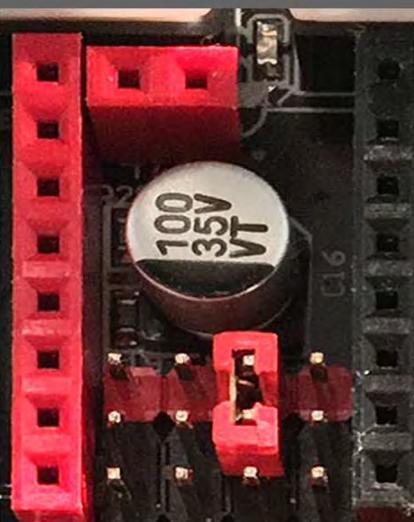
1
2
3



High ➤ set Jumper between rows
1 and 2

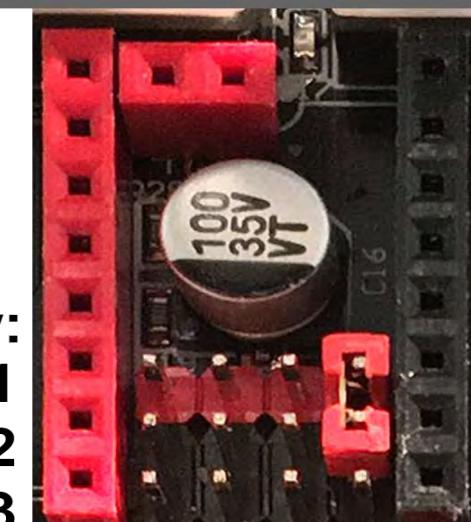
Row:

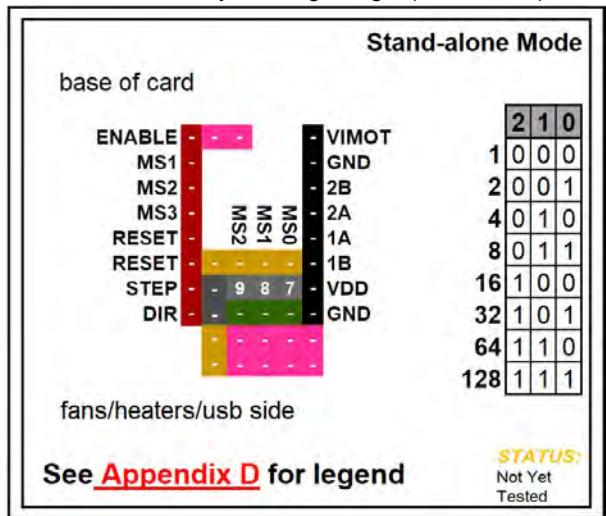
1
2
3



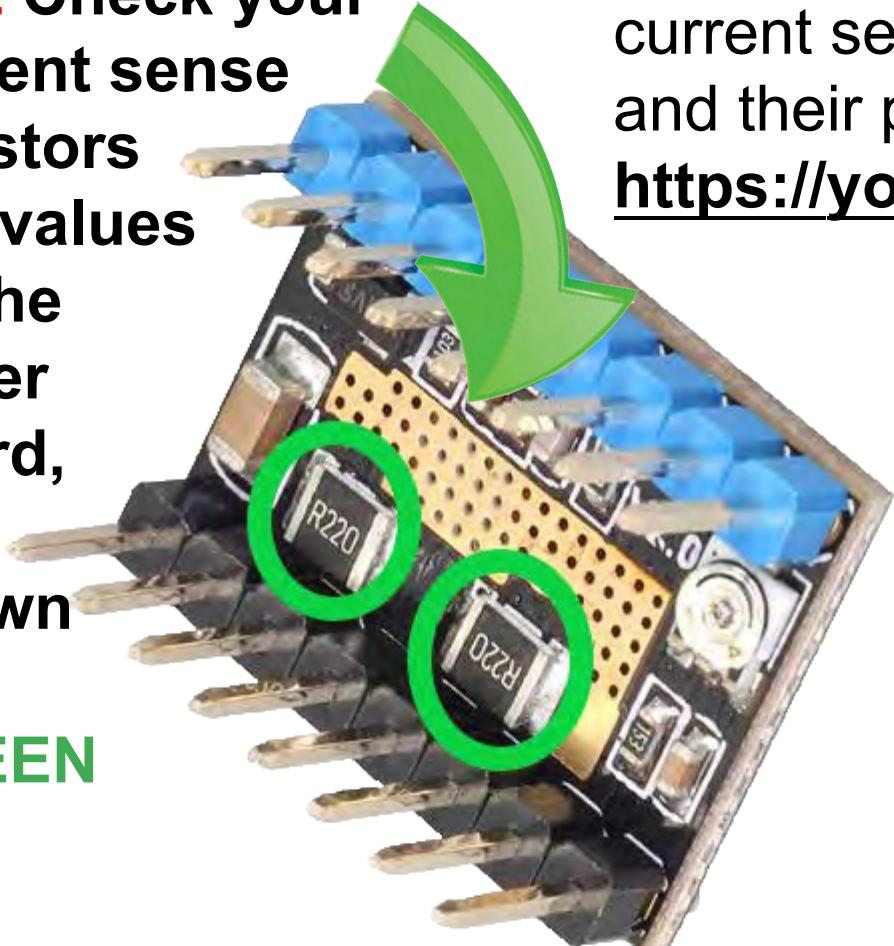
Row:

1
2
3





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

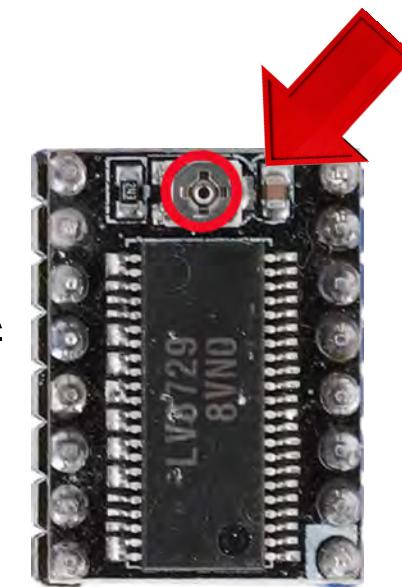


LERDGE LV8729

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: 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

LERDGE LV8729

STEP

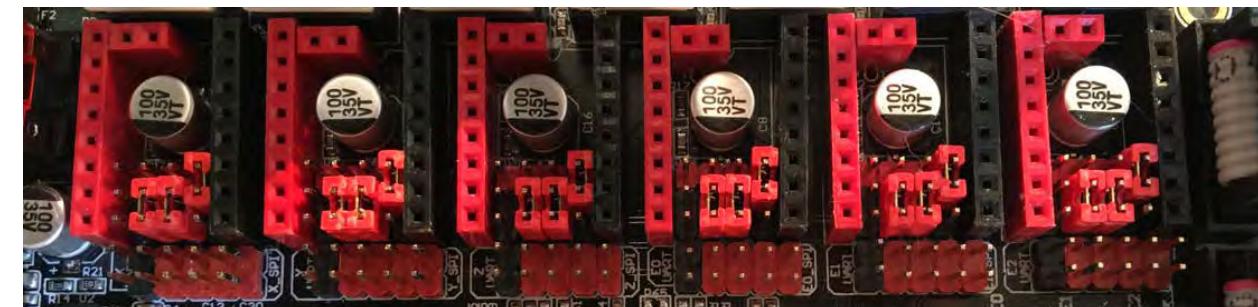
ENABLE	-	-	VIMOT
MS1	-	-	GND
MS2	-	-	2B
MS3	-	-	2A
RESET	-	MS2	1A
RESET	-	MS1	1B
STEP	-	MS0	VDD
DIR	9	8	7
	9	8	7
	-	-	GND



See [Appendix D](#) for legend

1 / 2

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

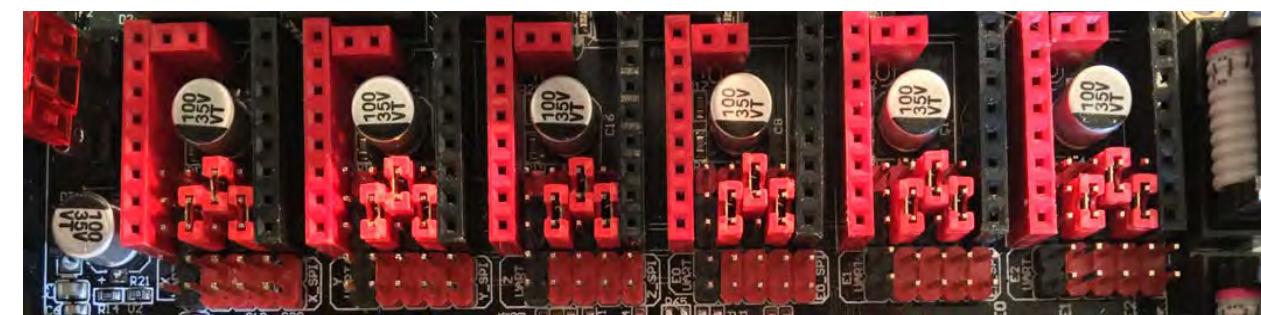


See [Appendix D](#) for legend

Stand-alone Mode

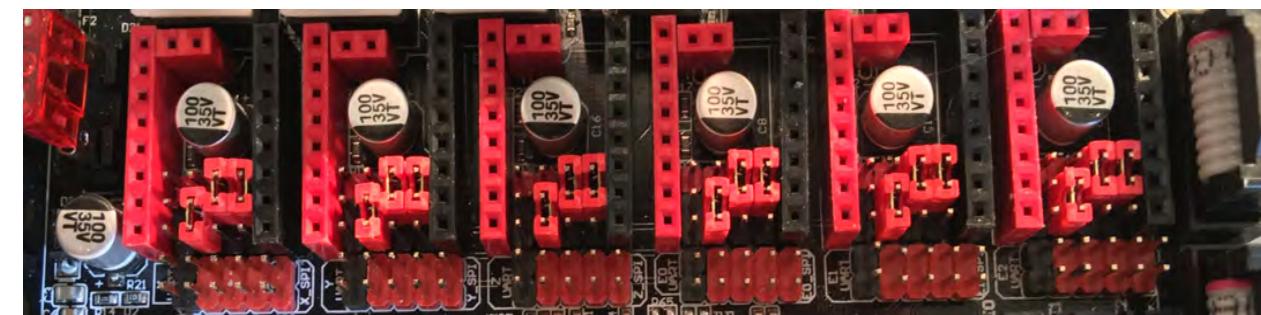
LERDGE LV8729

ENABLE	-	-	VIMOT
MS1			GND
MS2			2B
MS3			2A
RESET	MS2	MS1	MS0
RESET	-	8	-
STEP	-	9 8 7	VDD
DIR	9	-	7
			GND



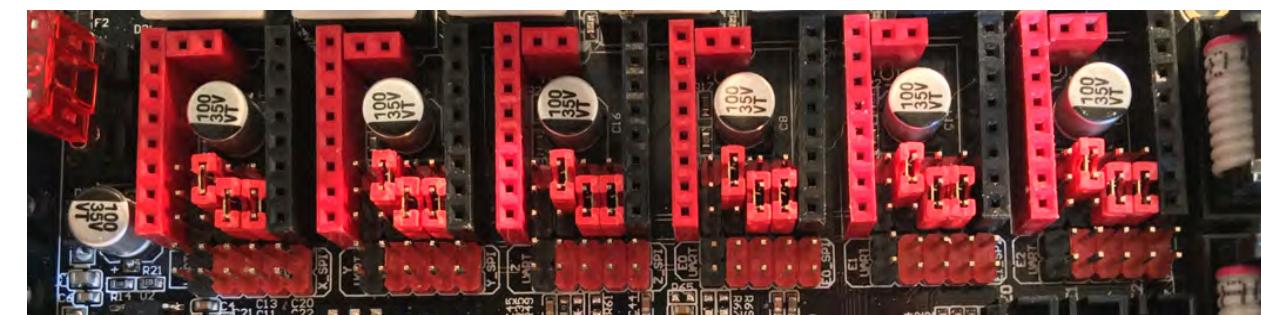
See [Appendix D](#) for legend

ENABLE	-	-	VIMOT
MS1			GND
MS2			2B
MS3			2A
RESET	MS2	MS1	MS0
RESET	-	8	7
STEP	-	9 8 7	VDD
DIR	9	-	GND



See [Appendix D](#) for legend

ENABLE	-	-	VIMOT
MS1			GND
MS2			2B
MS3			2A
RESET	MS2	MS1	MS0
RESET	-	9	-
STEP	-	9 8 7	VDD
DIR	-	8 7	GND



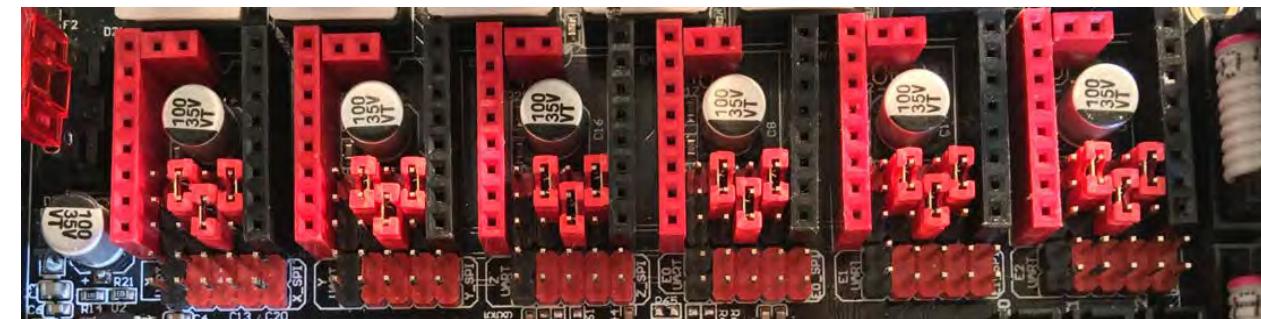
See [Appendix D](#) for legend

Stand-alone Mode

LERDGE LV8729

1 / 32

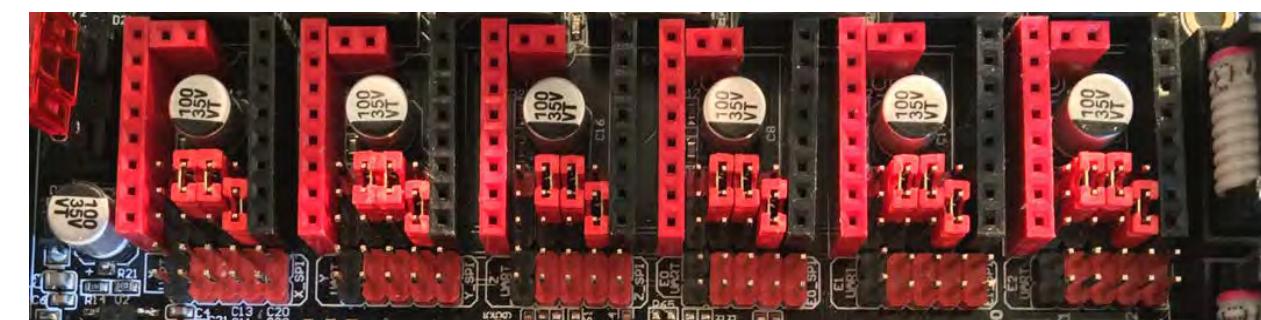
ENABLE	-	-	VIMOT
MS1	-	-	GND
MS2	-	-	2B
MS3	-	-	2A
RESET	-	MS2	1A
RESET	-	9	7
STEP	-	9	VDD
DIR	-	8	GND
	-	-	-
	-	-	-
	-	-	-



See [Appendix D](#) for legend

1 / 64

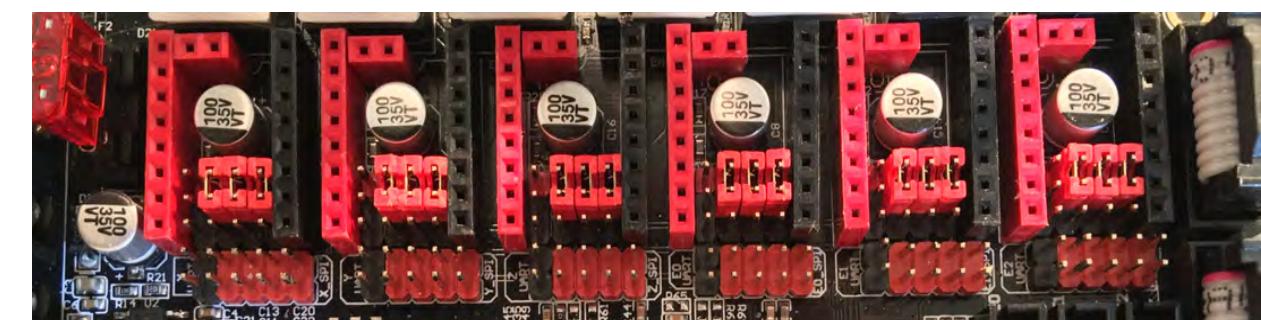
ENABLE	-	-	VIMOT
MS1	-	-	GND
MS2	-	-	2B
MS3	-	-	2A
RESET	-	MS2	1A
RESET	-	9	8
STEP	-	9	VDD
DIR	-	8	GND
	-	-	-
	-	-	-
	-	-	-



See [Appendix D](#) for legend

1 / 128

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

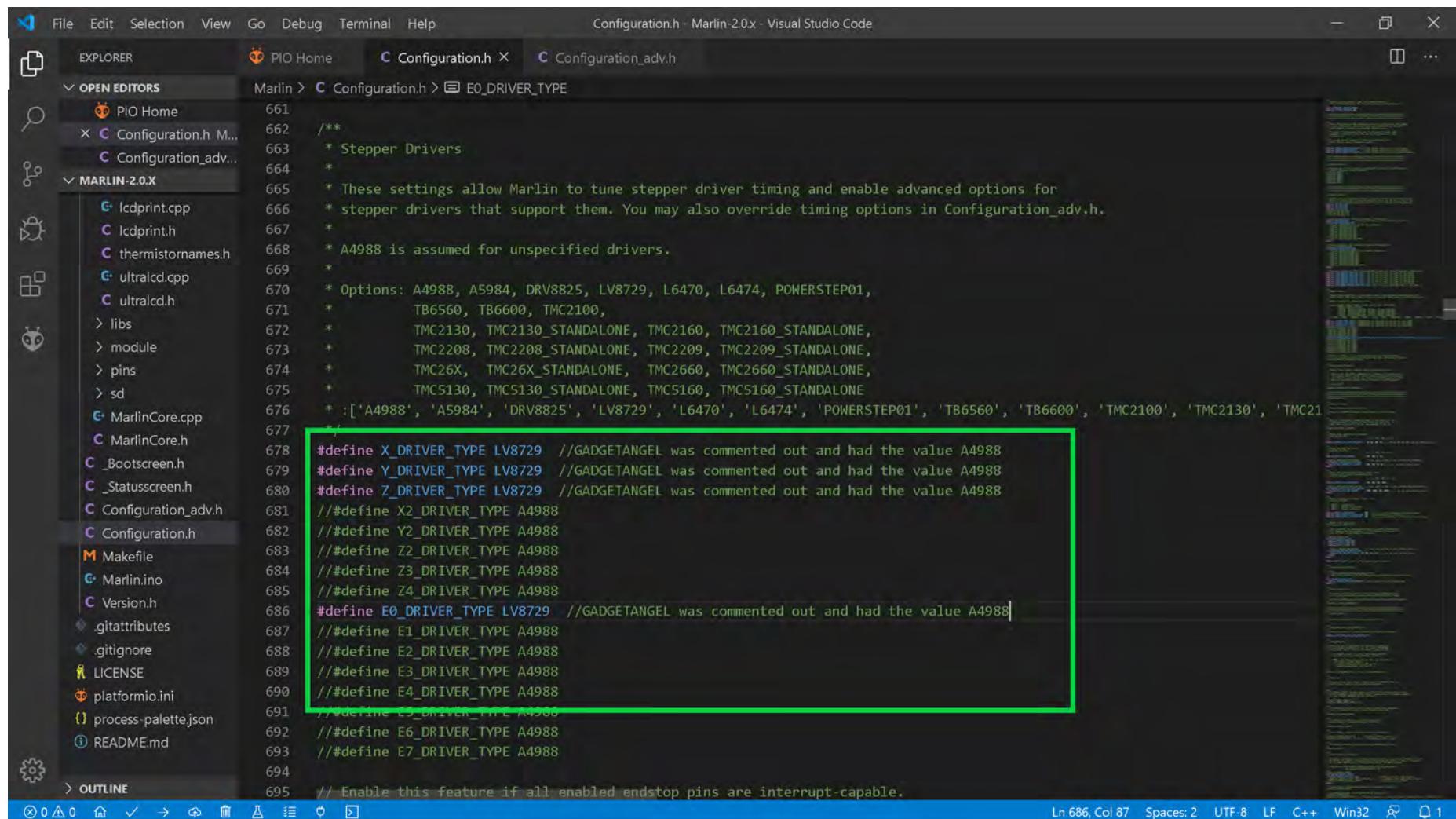


See [Appendix D](#) for legend

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 E0_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.

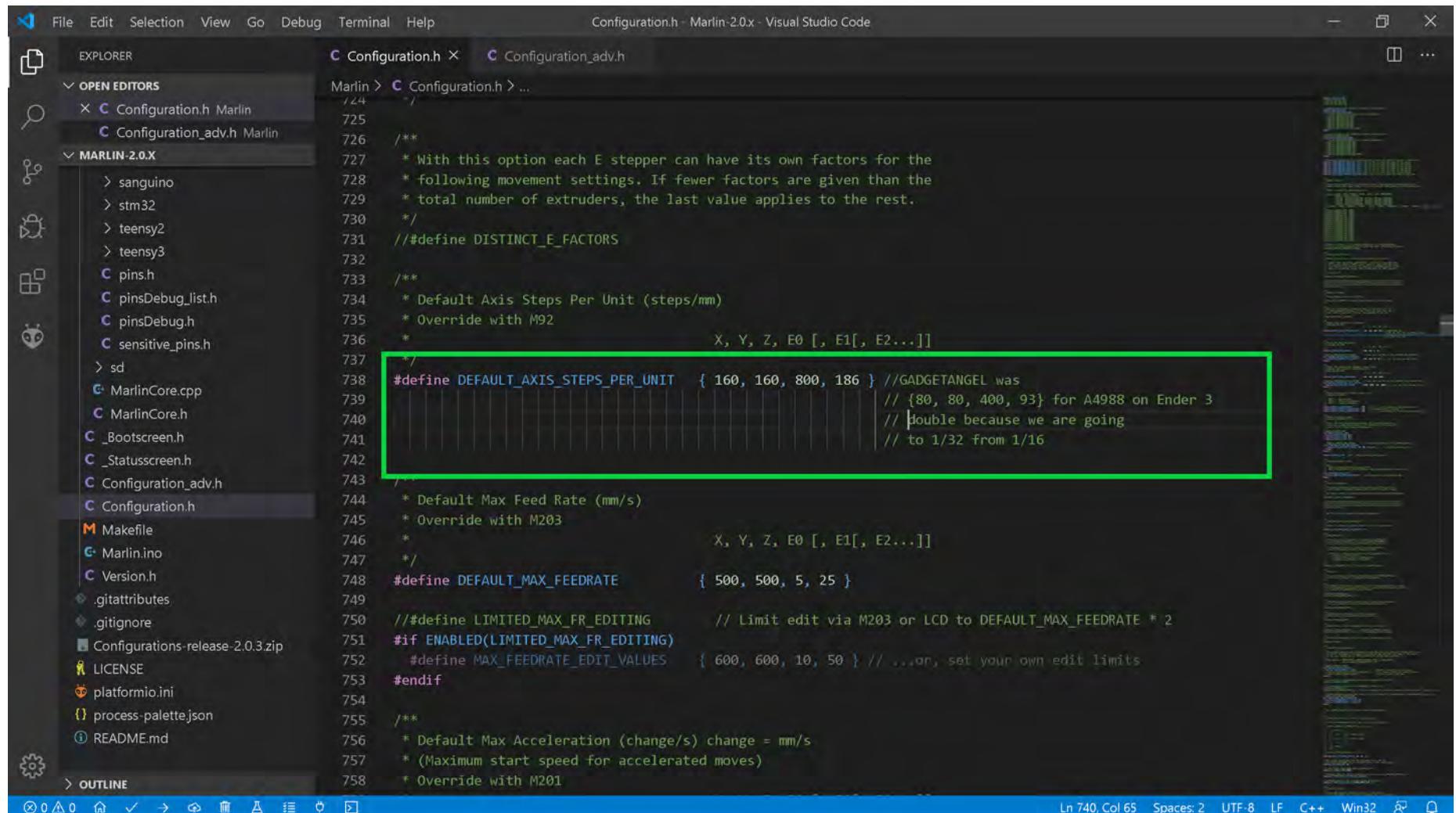
Ln 686, Col 87 Spaces: 2 UTF 8 LF C++ Win32 1

```

- 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:

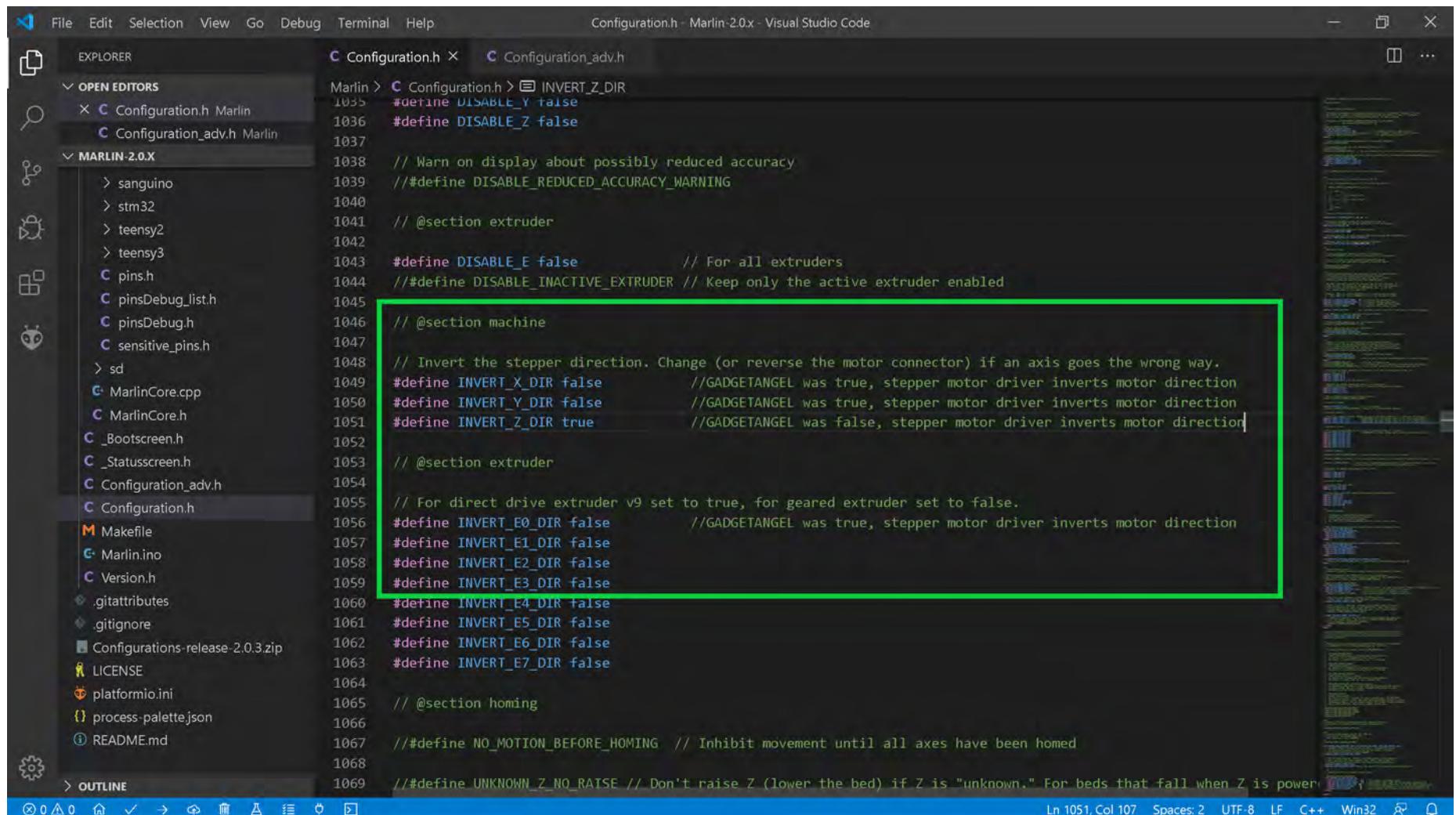
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom 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 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 file `Configuration.h` open. The code editor displays the Marlin configuration file. A green rectangular box highlights the section of code responsible for inverting stepper directions:

```

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

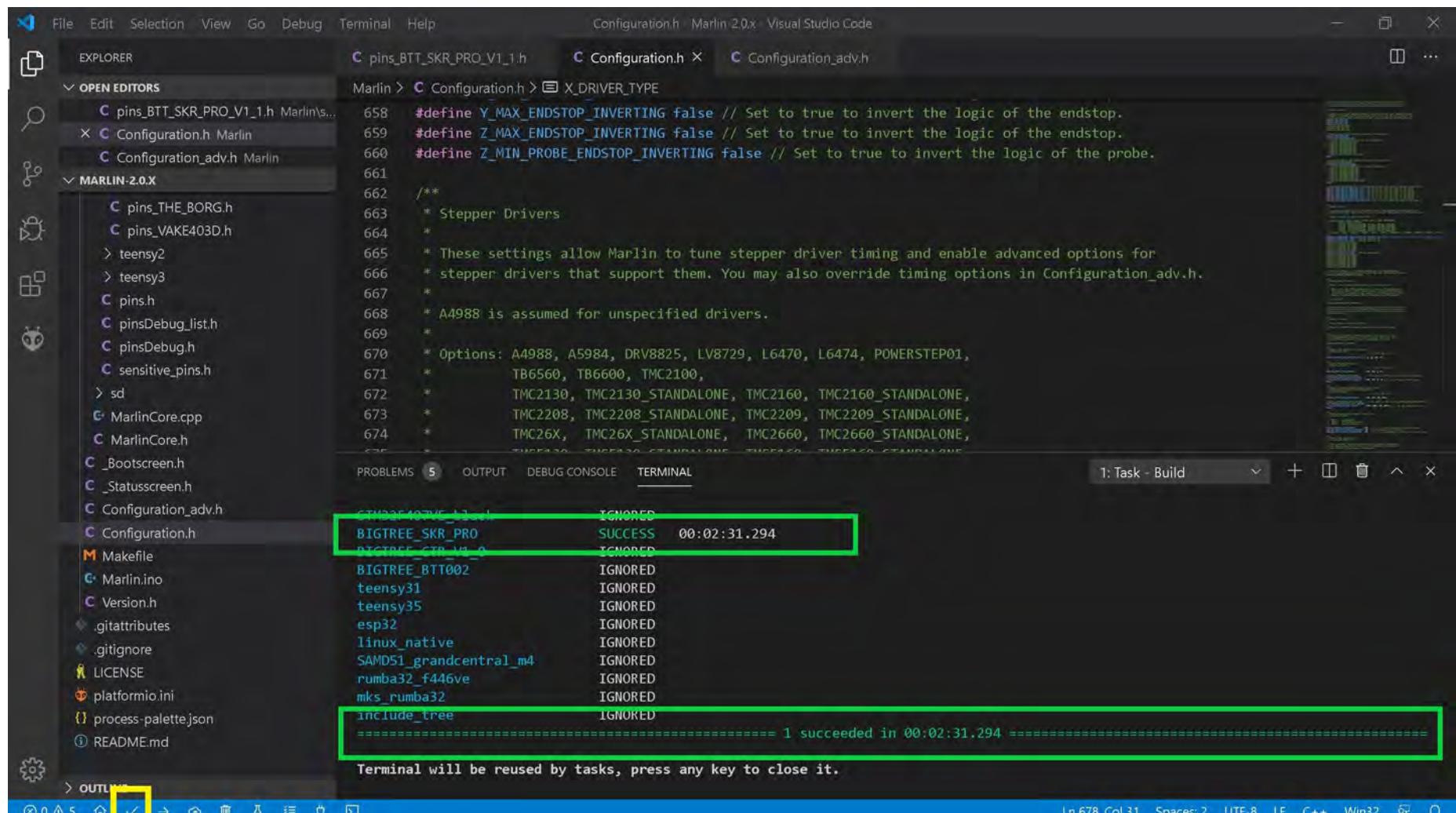
```

The code editor status bar at the bottom right indicates: Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32.

- Go to the next page.

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

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



A screenshot of the Visual Studio Code interface. The left sidebar shows project files under 'EXPLORER' and 'MARLIN-2.0.X'. The main editor window displays code from Configuration.h, specifically defining endstop inversion logic. Below the editor is the 'PROBLEMS' tab. The bottom section is the 'TERMINAL' tab, which shows the compilation results:

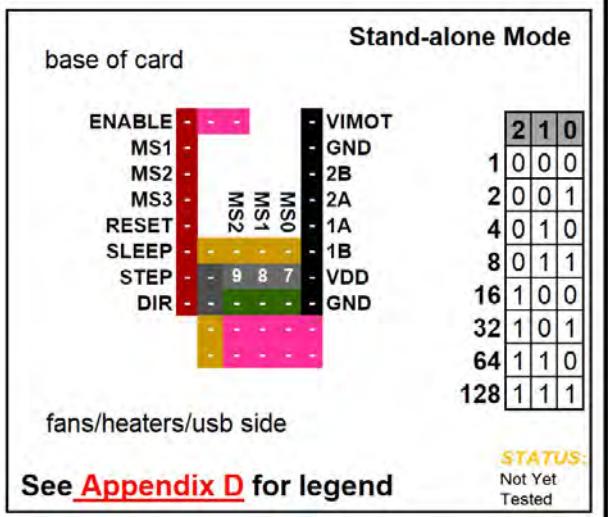
```
=====
BIGTREE_SKR_PRO      SUCCESS  00:02:31.294
BIGTREE_SKR_V1_0      IGNORED
BIGTREE_BT002        IGNORED
teensy31            IGNORED
teensy35            IGNORED
esp32              IGNORED
linux_native         IGNORED
SAMDS1_grandcentral_m4 IGNORED
rumba32_f446ve       IGNORED
mks_rumba32          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 toolbar. Two green boxes highlight the successful compilation message and the total time taken.

- 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 [Appendix E](#).

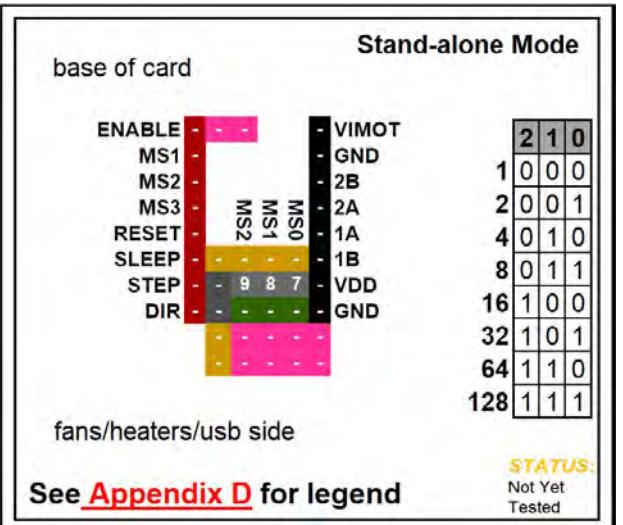
MKS LV8729

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

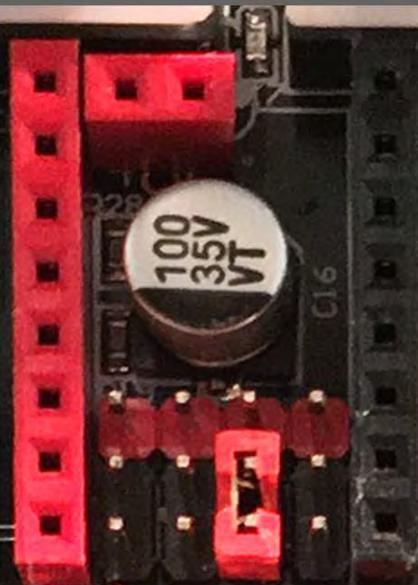
- See next page for the legend that belongs to the above chart.

**MKS LV8729****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

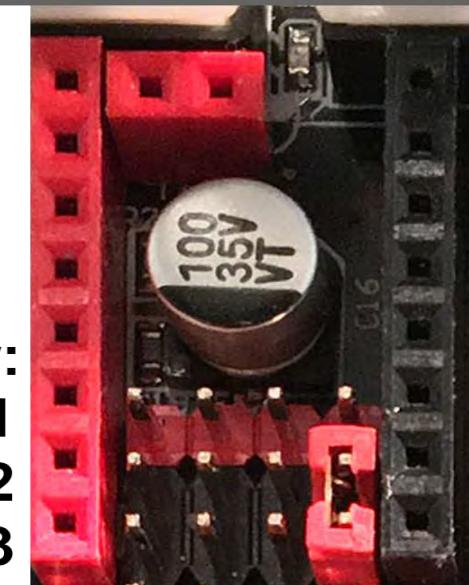
Low ➤ set Jumper between rows 2 and 3



Row:

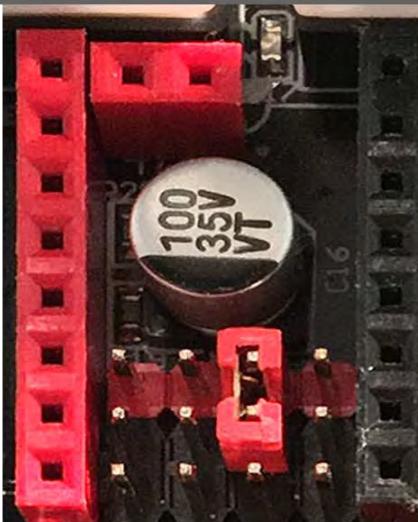
1
2
3

Row:

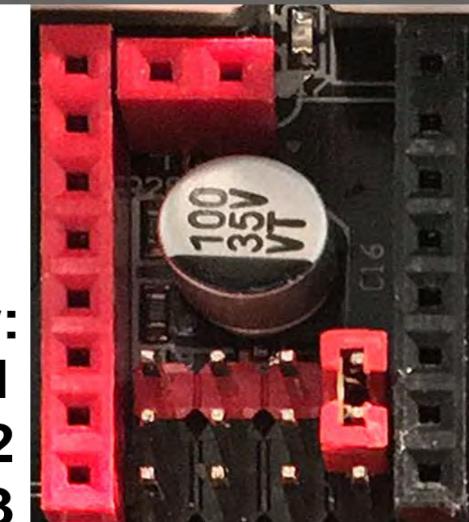
1
2
3

High ➤ set Jumper between rows 1 and 2

Row:

1
2
3

Row:

1
2
3

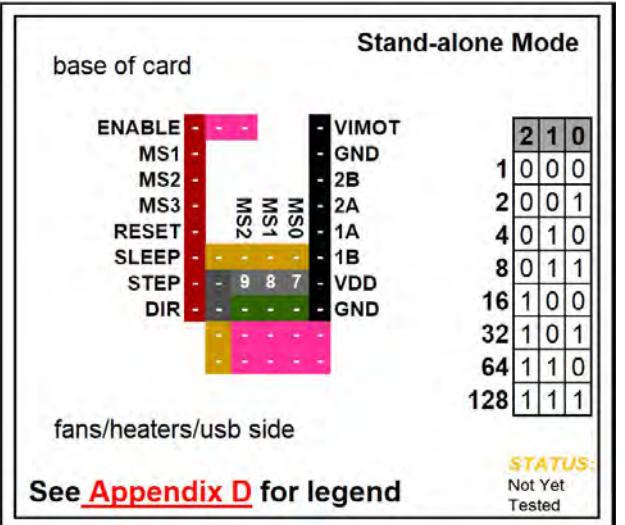
This is a Jumper:



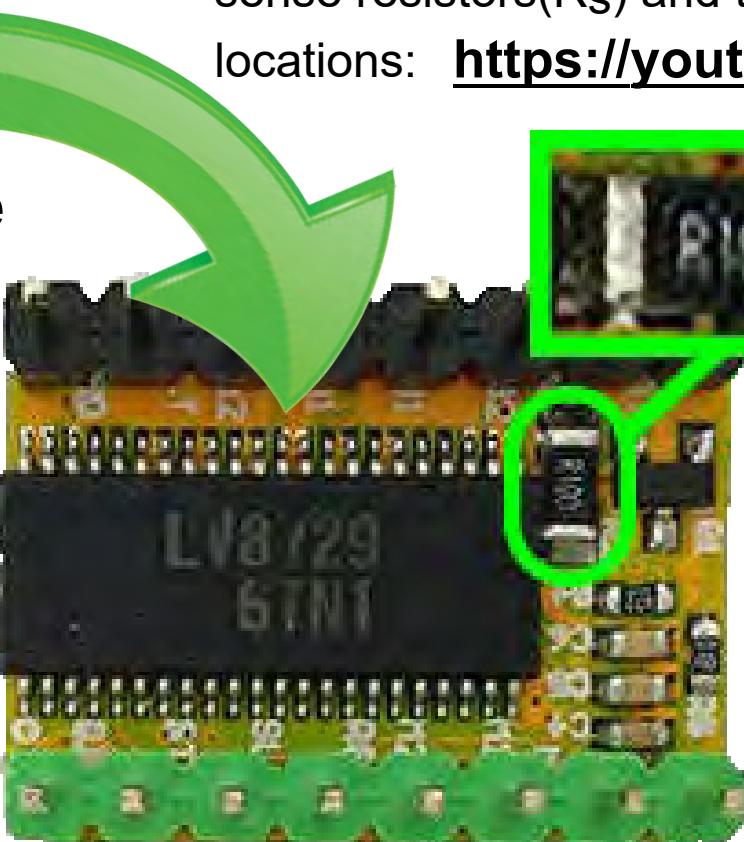
MKS LV8729

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

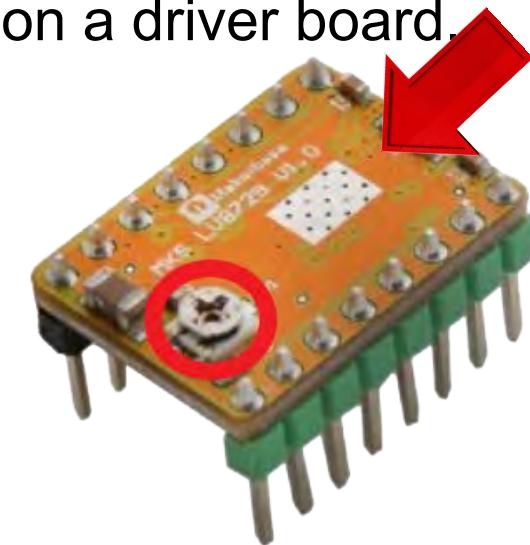
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



Note: Check your current sense resistor (R_s) value 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>



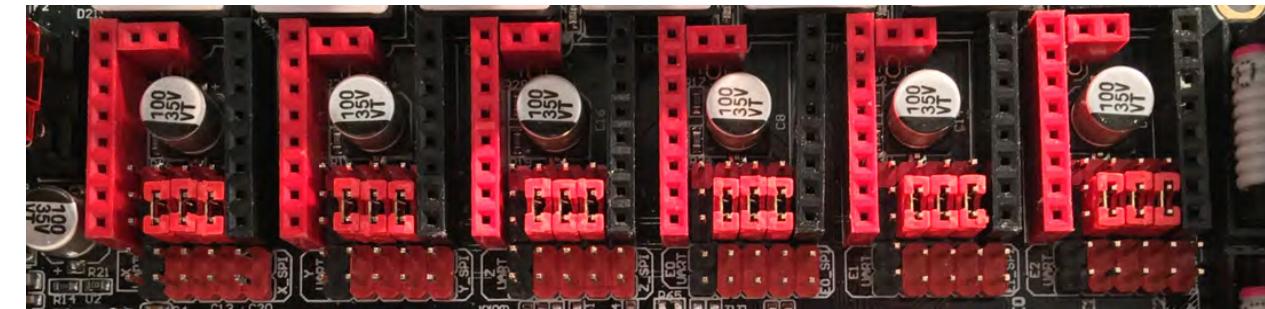
- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

Stand-alone Mode

MKS LV8729

STEP

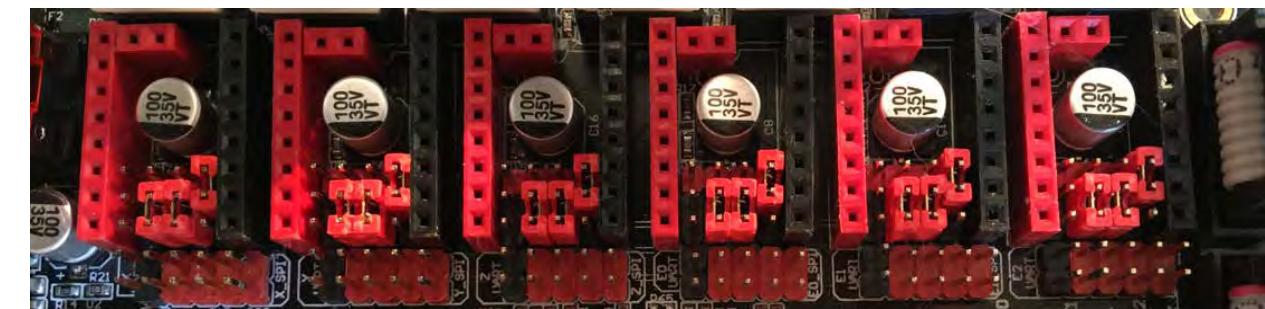
ENABLE	-	-	VIMOT
MS1			GND
MS2			2B
MS3			2A
RESET		MS2 MS1 MS0	1A
SLEEP	-	-	1B
STEP	-	9 8 7	VDD
DIR	-	9 8 7	GND



See [Appendix D](#) for legend

1 / 2

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



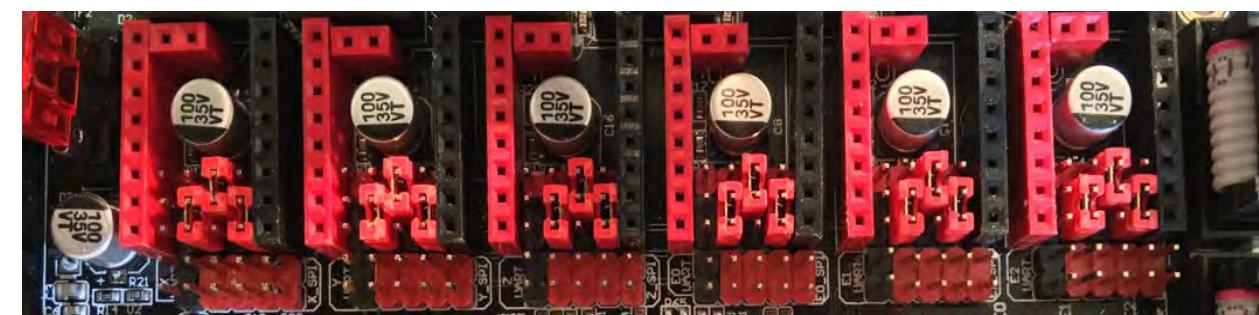
See [Appendix D](#) for legend

Stand-alone Mode

MKS LV8729

1 / 4

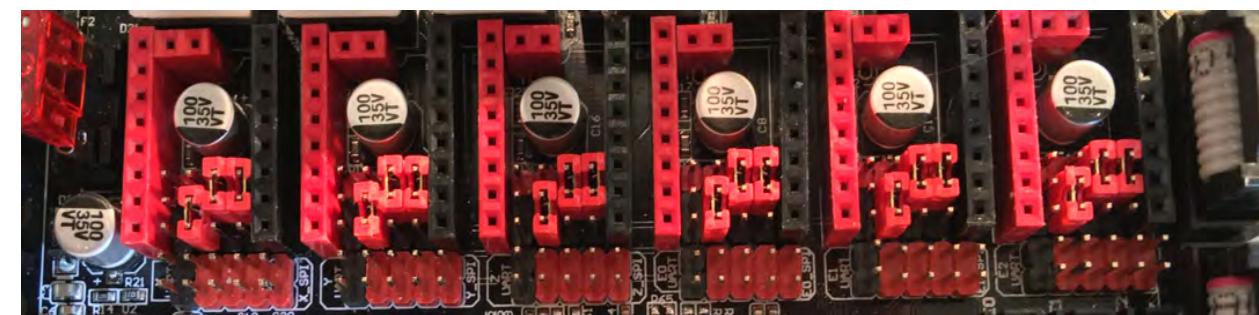
ENABLE	-	-	-	VIMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	-	-	2A	
RESET	-	MS2	MS1	MS0	1A
SLEEP	-	-	8	-	1B
STEP	-	9	8	7	VDD
DIR	-	9	-	7	GND



See [Appendix D](#) for legend

1 / 8

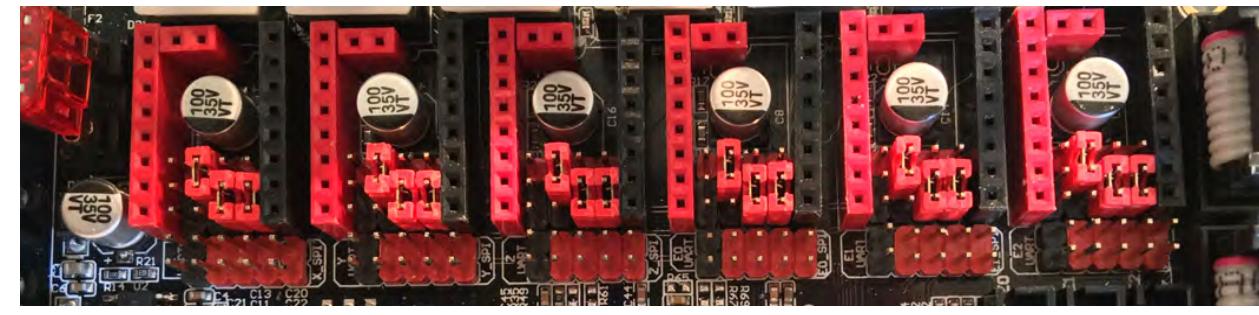
ENABLE	-	-	-	VIMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	-	-	2A	
RESET	-	MS2	MS1	MS0	1A
SLEEP	-	-	8	7	1B
STEP	-	9	8	7	VDD
DIR	-	9	-	-	GND



See [Appendix D](#) for legend

1 / 16

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



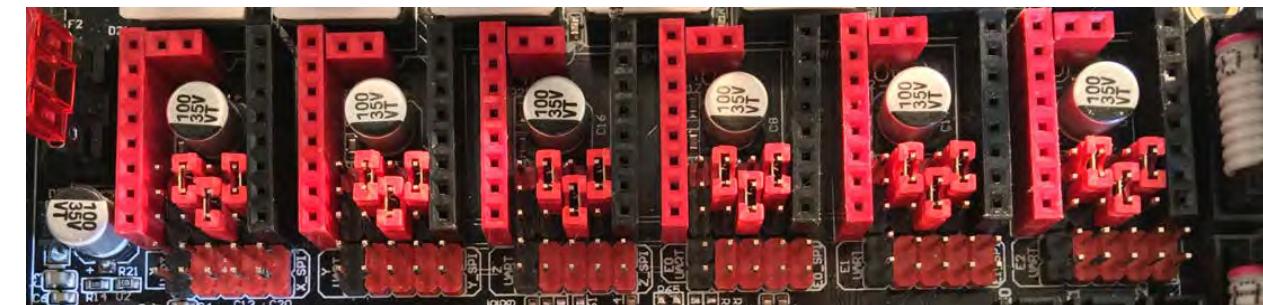
See [Appendix D](#) for legend

Stand-alone Mode

MKS LV8729

1 / 32

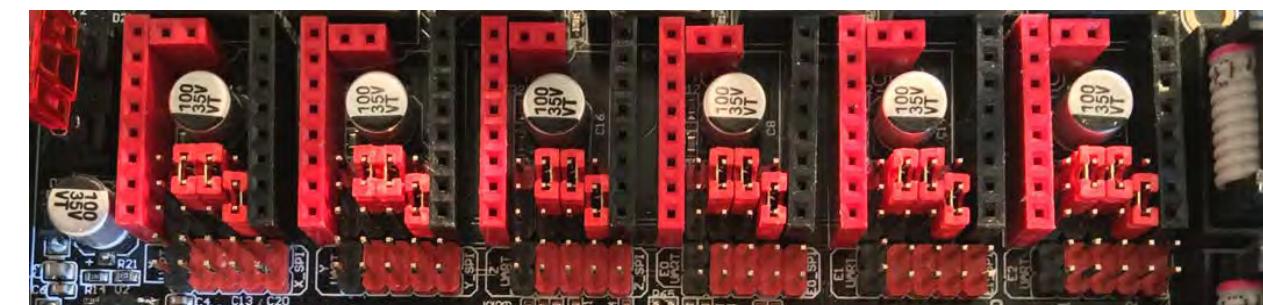
ENABLE	-	-	-	VIMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	-	-	2A	
RESET	-	MS2	MS1	MS0	1A
SLEEP	-	9	7	-	1B
STEP	-	9	8	7	VDD
DIR	-	-	8	-	GND



See [Appendix D](#) for legend

1 / 64

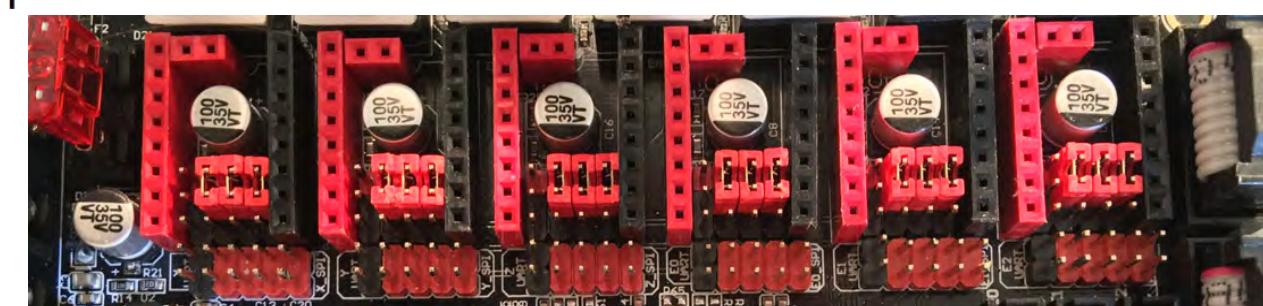
ENABLE	-	-	-	VIMOT	
MS1	-	-	-	GND	
MS2	-	-	-	2B	
MS3	-	MS2	MS1	MS0	2A
RESET	-	-	9	8	1A
SLEEP	-	9	8	-	1B
STEP	-	9	8	7	VDD
DIR	-	-	-	7	GND



See [Appendix D](#) for legend

1 / 128

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

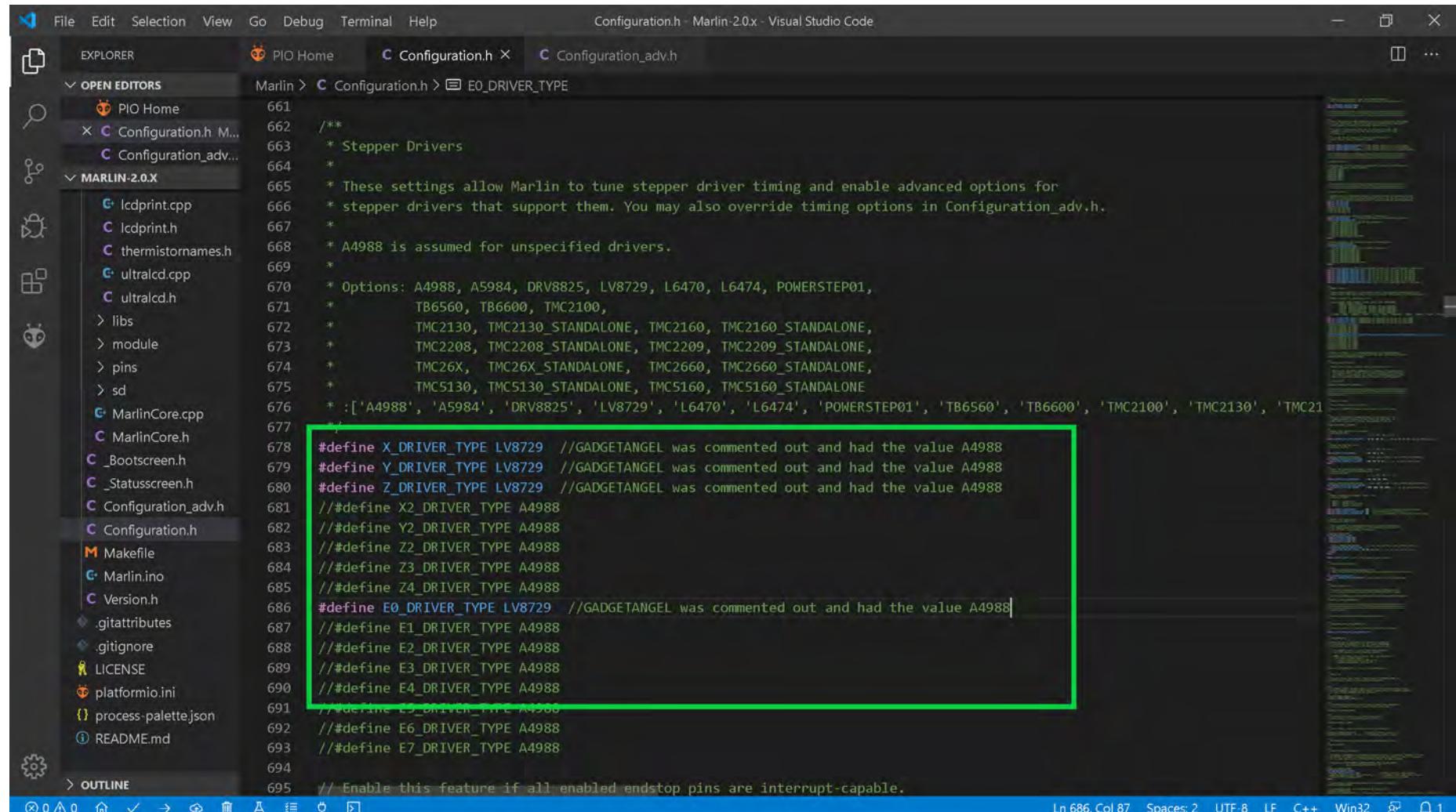


See [Appendix D](#) for legend

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.

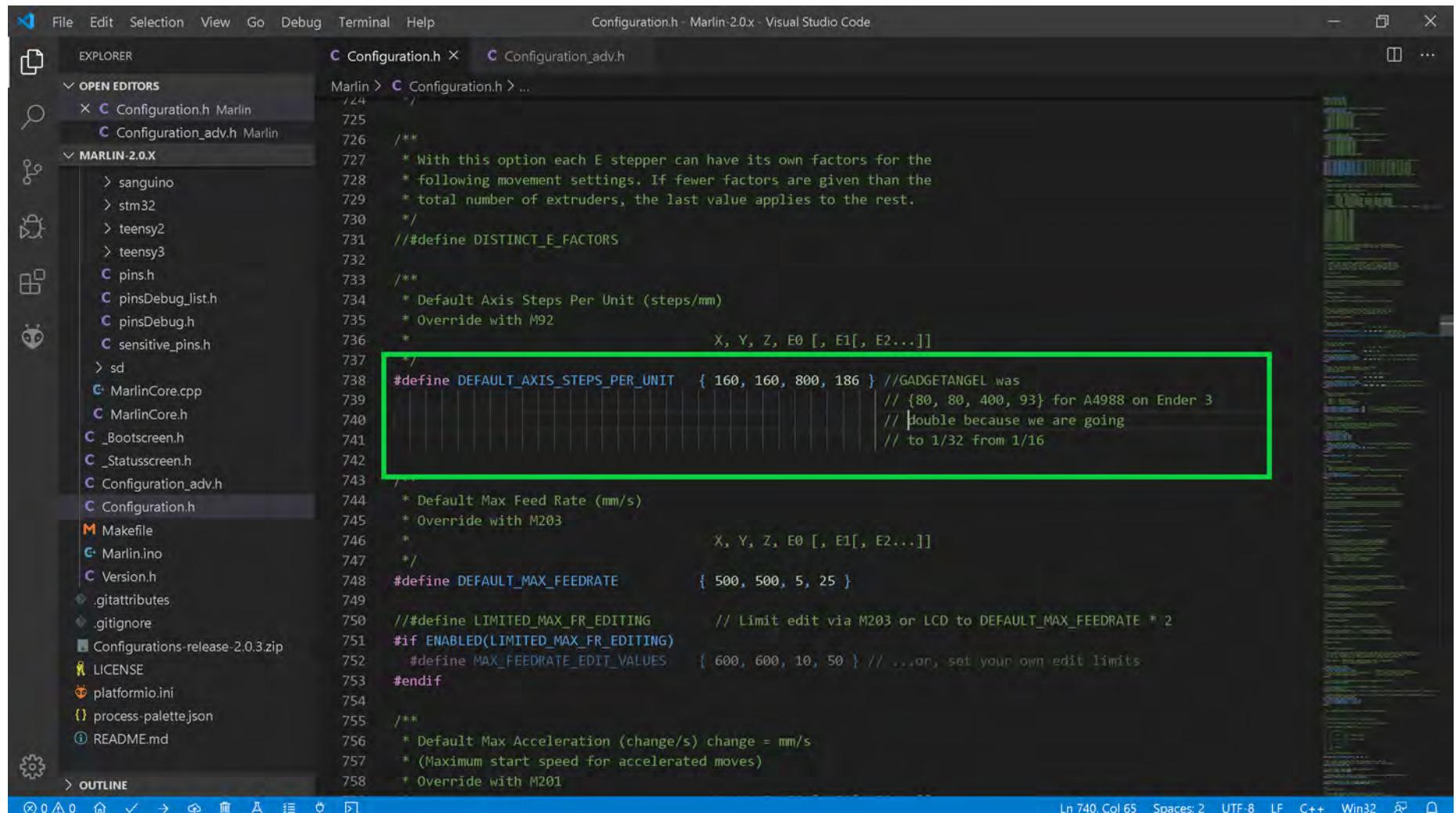
```

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.



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
Marlin > Configuration.h ...
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 * 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 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
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 } // ...or, set your own edit limits
753 #endif
754
755 /**
756 * Default Max Acceleration (change/s) change = mm/s
757 * (Maximum start speed for accelerated moves)
758 * Override with M201

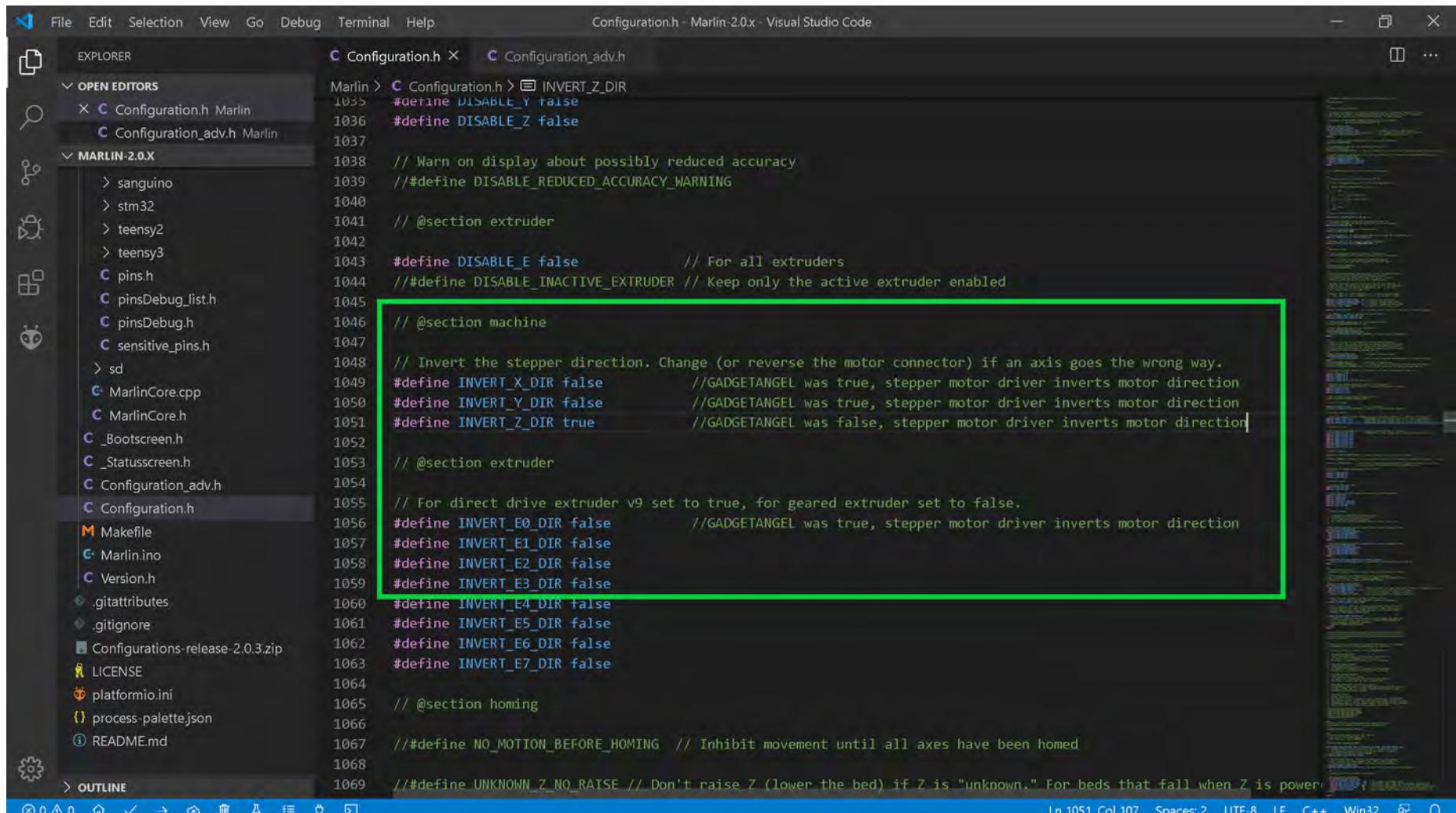
```

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

- Go to the next page.

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

EXPLORER

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
- C Marlin.ino
- C Version.h
- .gitattributes
- .gitignore
- Configurations-release-2.0.3.zip
- LICENSE
- platformio.ini
- process-palette.json
- README.md

OUTLINE

```

Marlin > C Configuration.h > #define 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

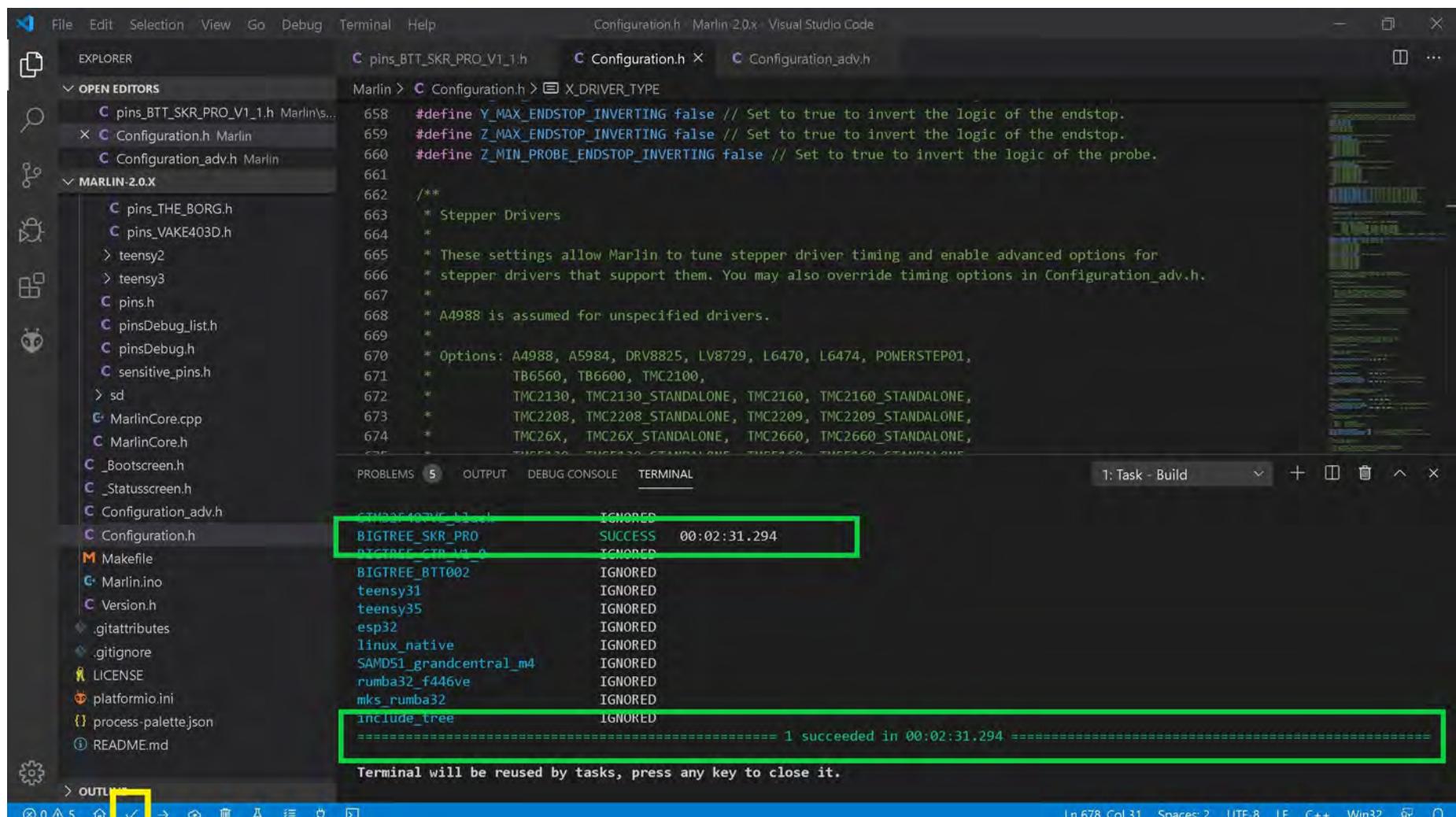
```

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

- Go to the next page.

The (latest release of) Marlin Setup for 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.



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

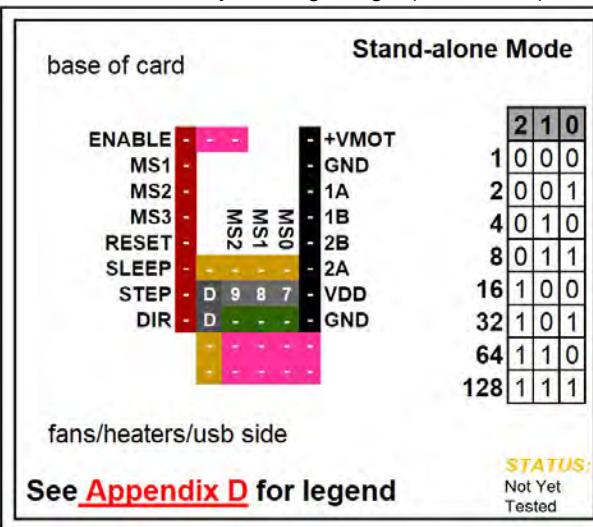
```

CTM2208_TMC2130_12_0: IGNORED
BIGTREE_SKR_PRO: SUCCESS 00:02:31.294
BIGTREE_CTP_12_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 message "1 succeeded in 00:02:31.294" is highlighted with a green box. A yellow box highlights the checkmark icon in the bottom-left corner of the terminal tab bar.

- 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 [Appendix E](#).

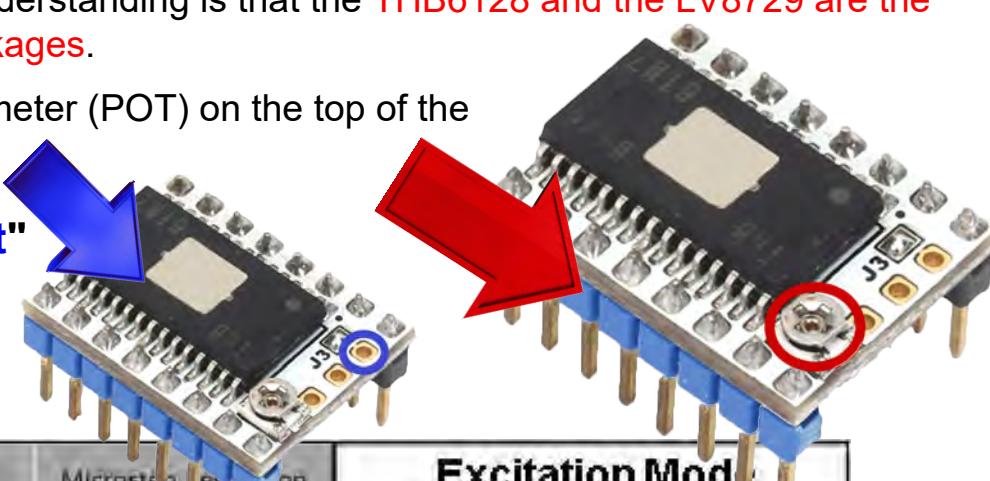


FYSETC S6128 V1.1

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.

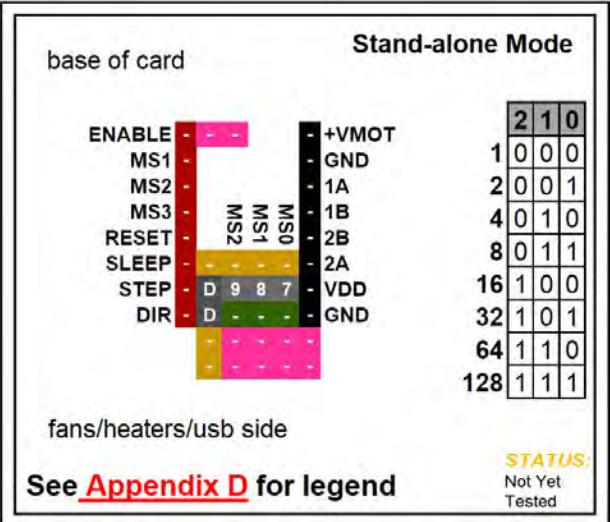
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

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

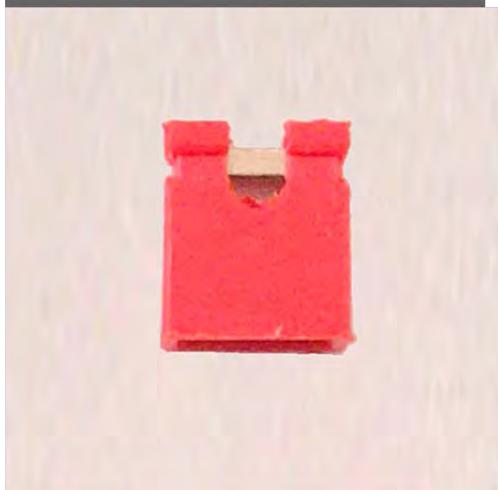


Driver Chip	MS2	MS1	MS0	Microstep Resolution	Excitation Mode
FYSETC SureStepr SD6128 Maximum 128 Subdivision 35V DC 2.2A (peak)	Low	Low	Low	Full Step	2 Phase
	Low	Low	High	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	Low	High	High	1/8 Step	2W1-2 Phase
	High	Low	Low	1/16 Step	4W1-2 Phase
	High	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	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$		
R_S (Typical Sense Resistor)=0.1Ω					

- See next page for the legend that belongs to the above chart.



This is a
Jumper:



FYSETC S6128 V1.1

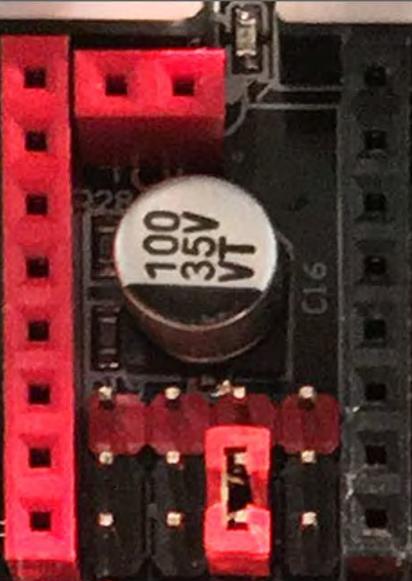
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➤ set Jumper between rows
2 and 3



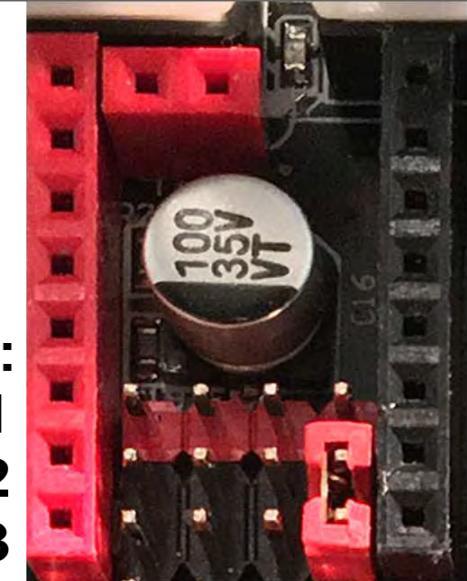
Row:

1
2
3



Row:

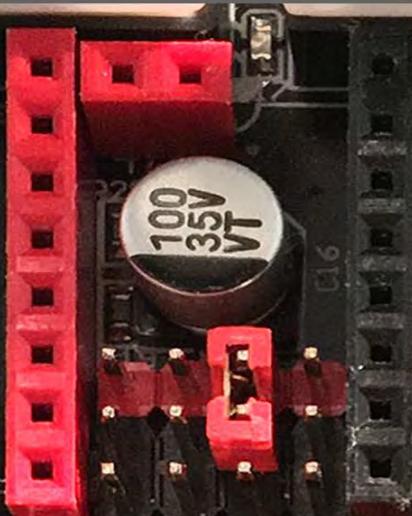
1
2
3



High ➤ set Jumper between rows
1 and 2

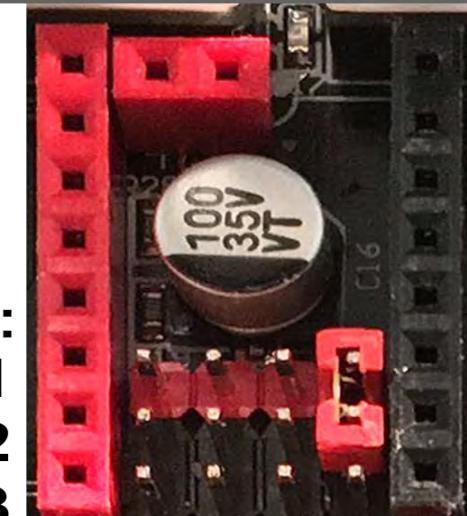
Row:

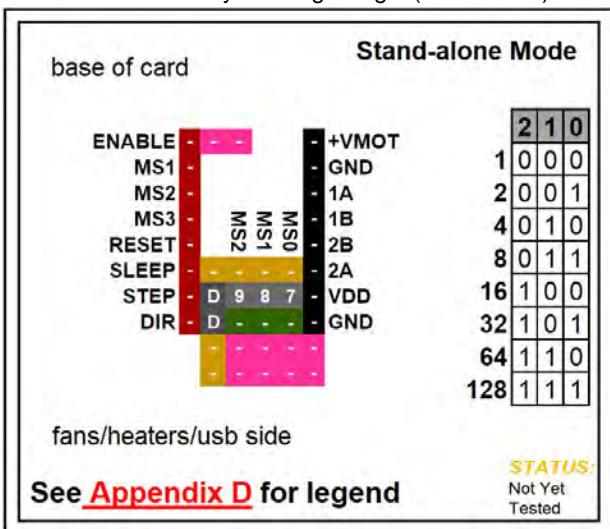
1
2
3



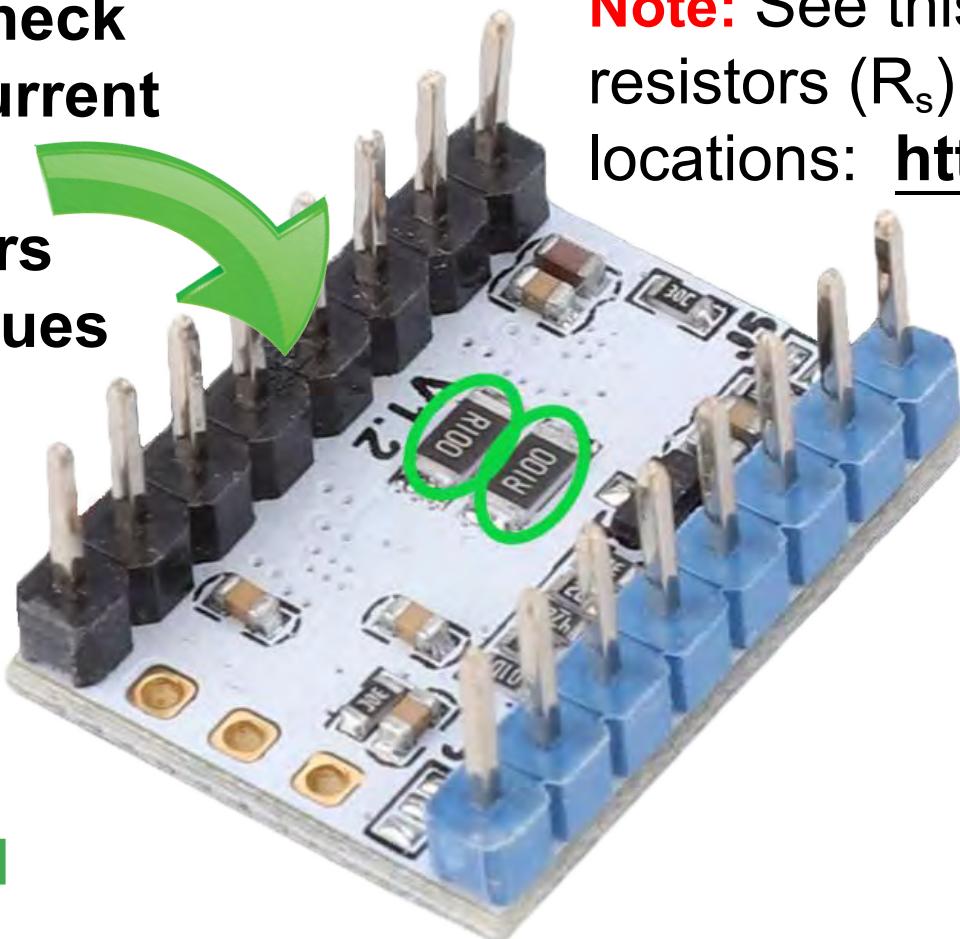
Row:

1
2
3





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



FYSETC S6128 V1.1

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

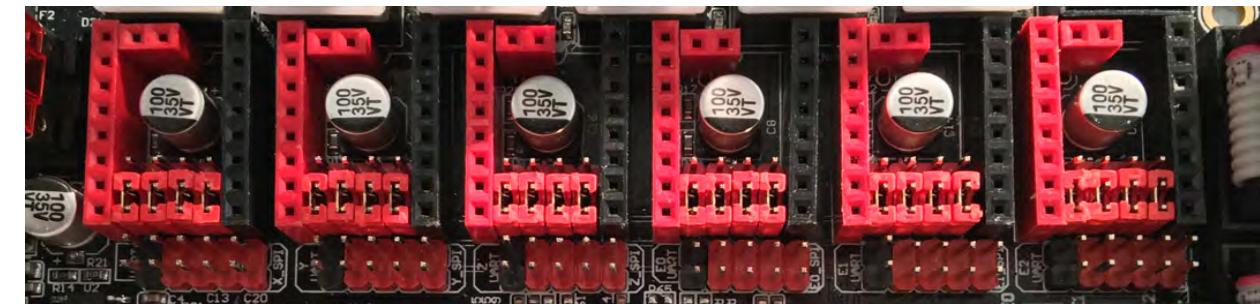
Stand-alone Mode

FYSETC S6128 V1.1

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
	-	-	-	-
	-	-	-	-

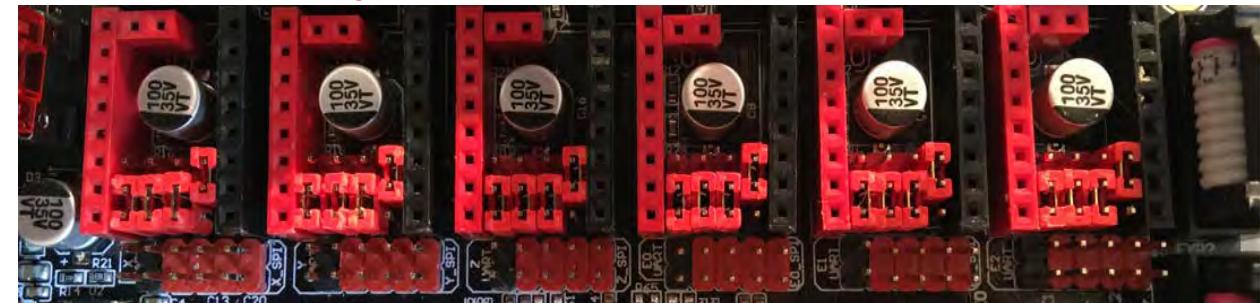
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

STEP

1 / 2

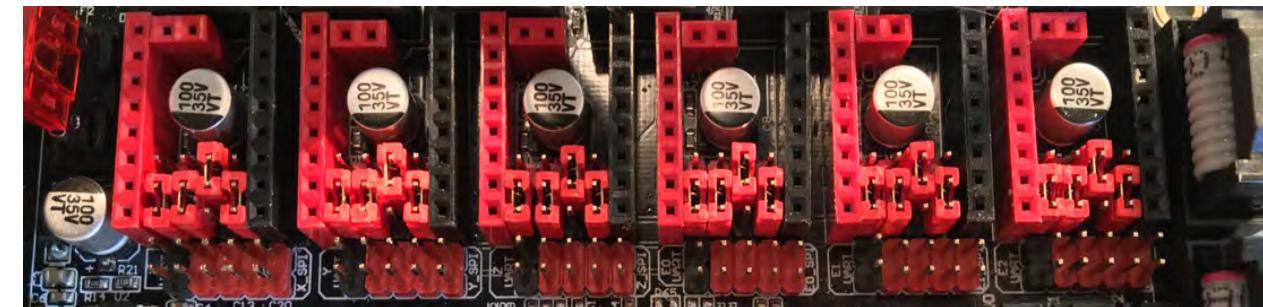
Stand-alone Mode

FYSETC S6128 V1.1

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	- 2B	
SLEEP	[]	MS1	- 2A	
STEP	D	9 8 7	- VDD	
DIR	D	9 - 7	- GND	

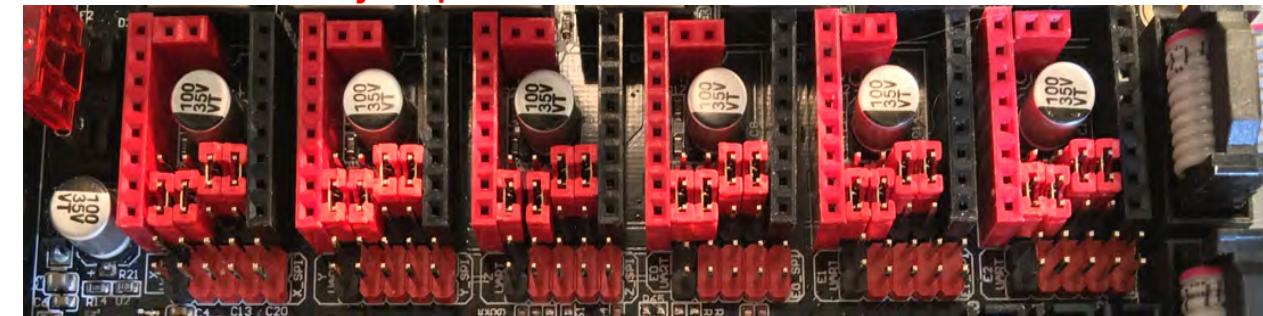
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

1 / 4

1 / 16

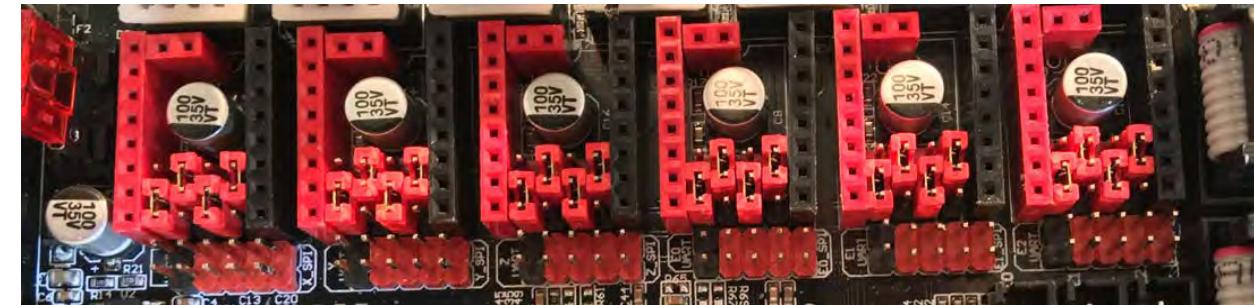
Stand-alone Mode

FYSETC S6128 V1.1

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	-	2B
SLEEP	-	9	7	-	-	2A
STEP	D	9	8	7	-	VDD
DIR	D	-	8	-	-	GND

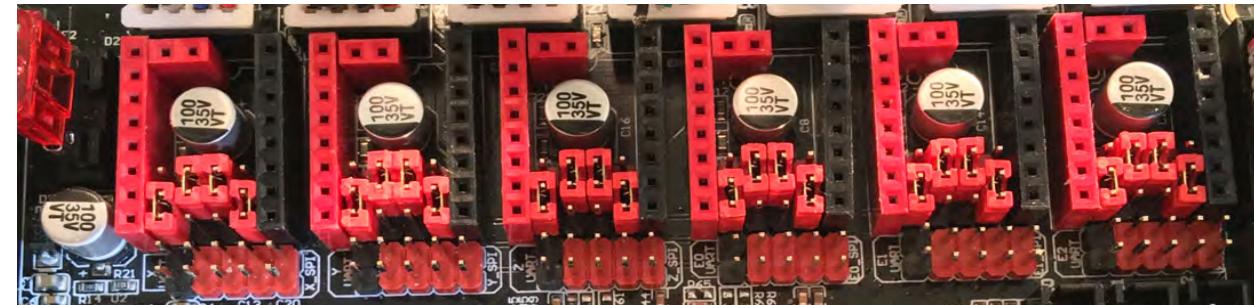
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

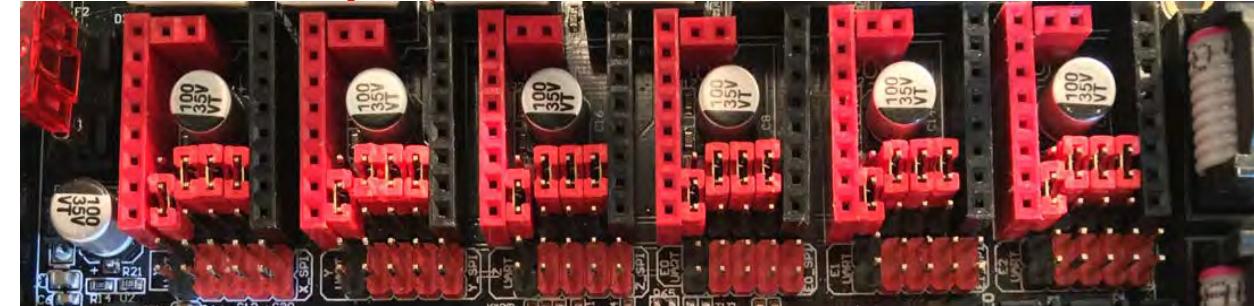
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

1 / 32

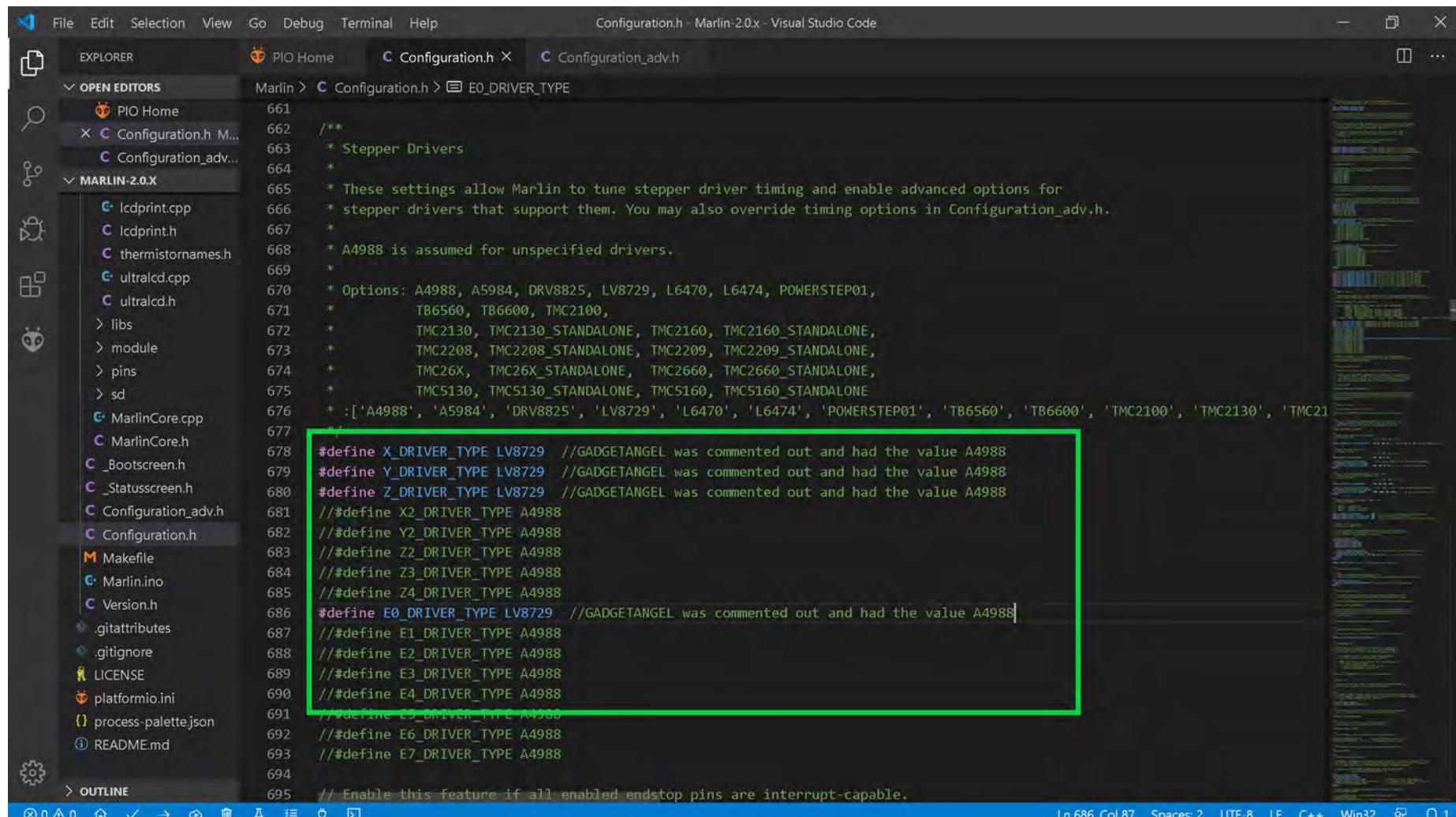
1 / 64

1 / 128

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 file `Configuration.h` open. The code editor displays the following driver configuration section:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 *           T86560, T86600, 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', 'T86560', 'T86600', '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
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

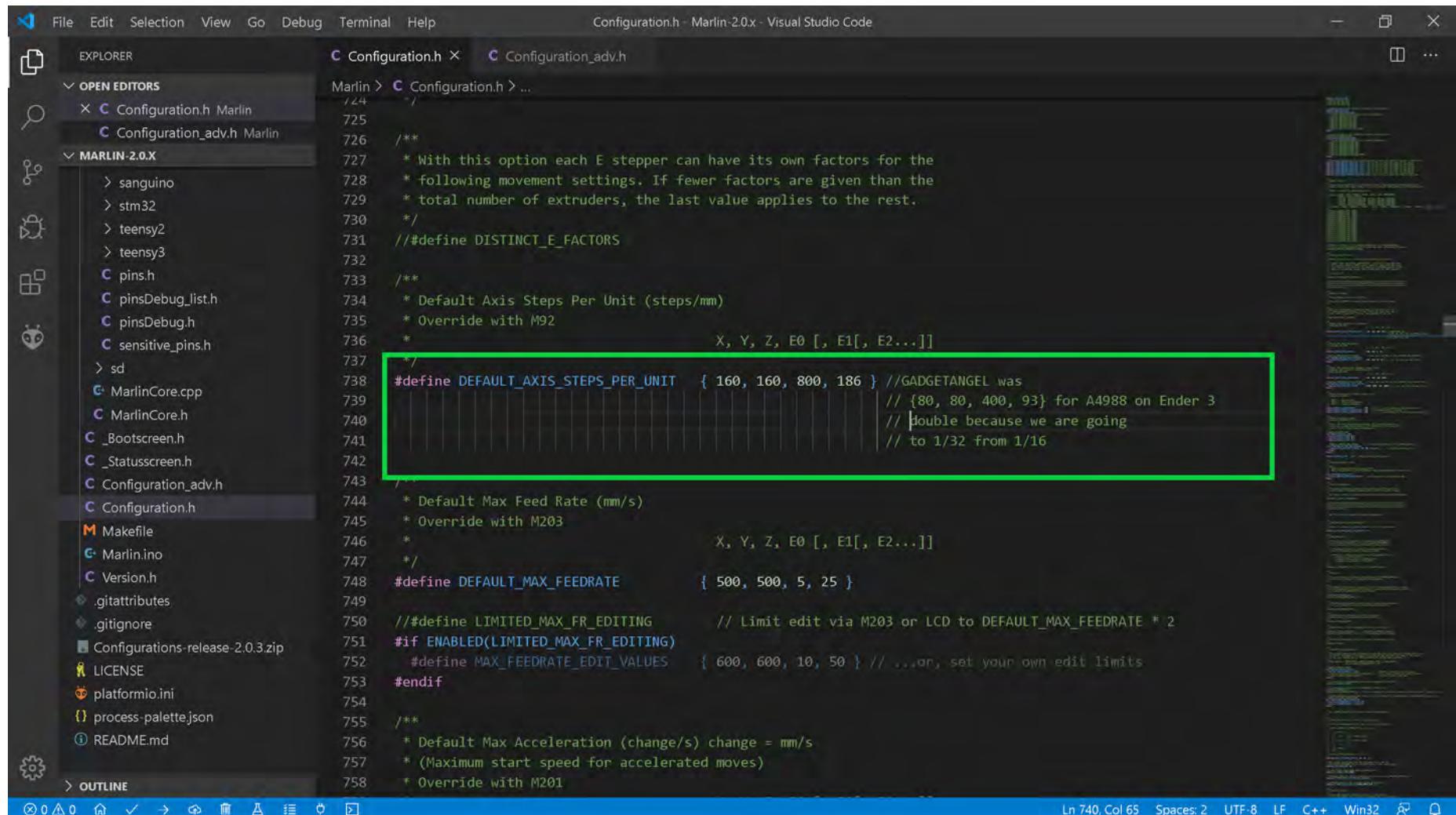
```

A green rectangular box highlights the lines from `#define X_DRIVER_TYPE LV8729` to `#define E7_DRIVER_TYPE A4988`. This indicates that the user is currently modifying these lines to use the LV8729 driver instead of the commented-out A4988 driver.

- 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.



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 ...
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 * 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 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
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 } // ...or, set your own edit limits
753 #endif
754
755 /**
756 * Default Max Acceleration (change/s) change = mm/s
757 * (Maximum start speed for accelerated moves)
758 * Override with M201

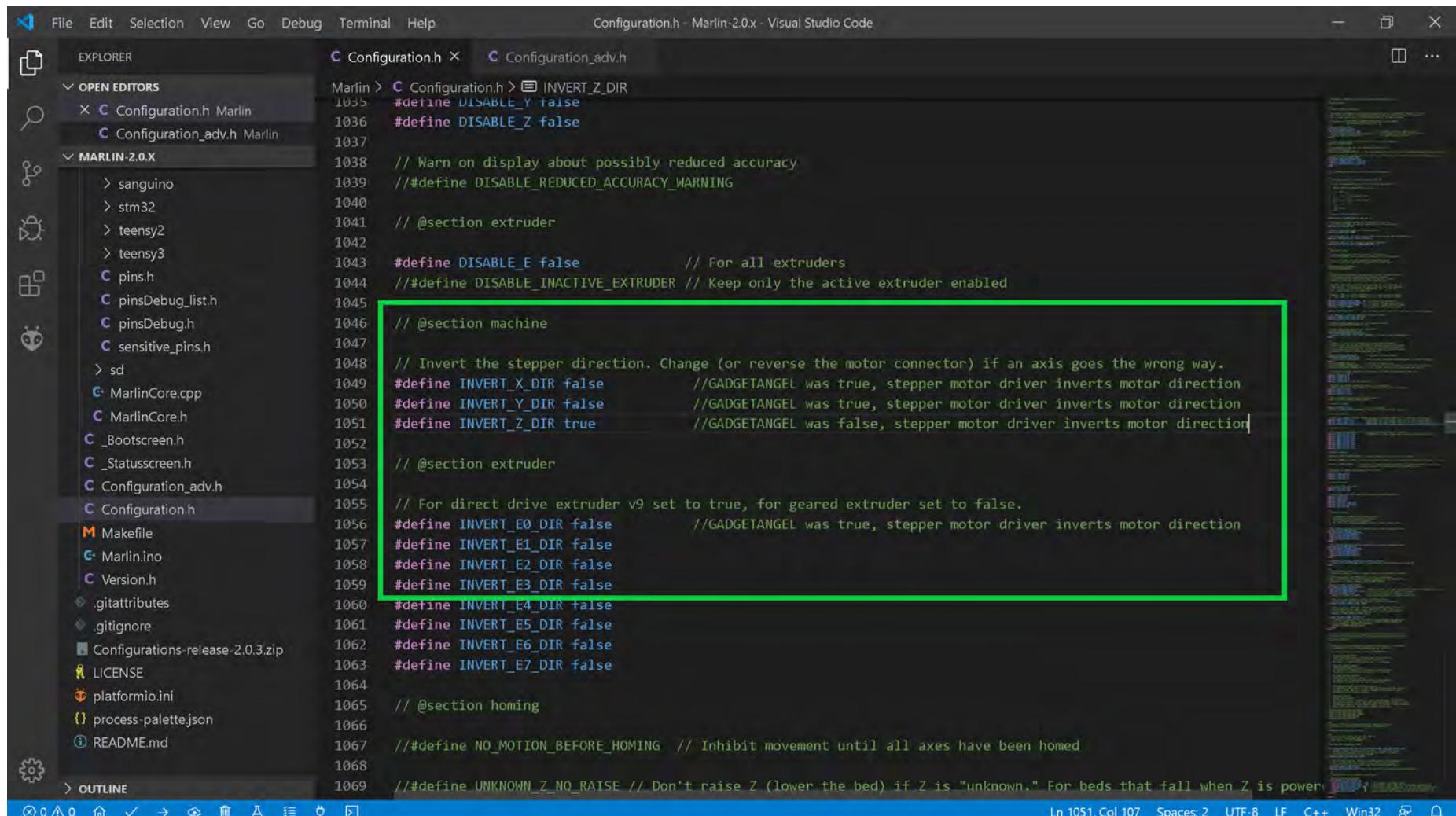
```

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

- Go to the next page.

The (latest release of) Marlin Setup for 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



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet, with the last three lines highlighted in green:

```

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 on

```

The green box highlights the following lines of code:

```

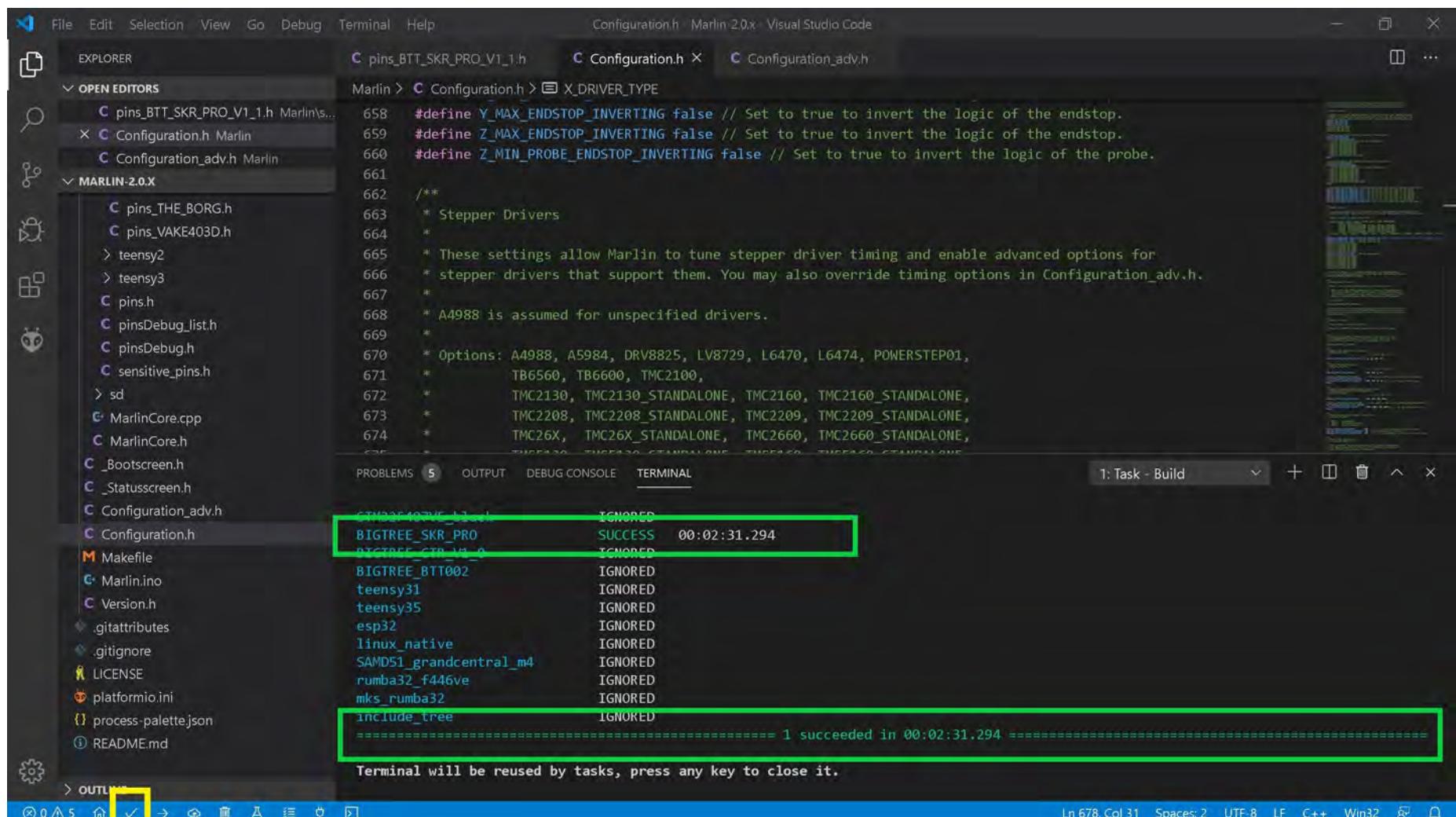
#define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor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

```

- 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 file 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' tab showing build status for various boards. The terminal tab at the bottom shows a successful build process:

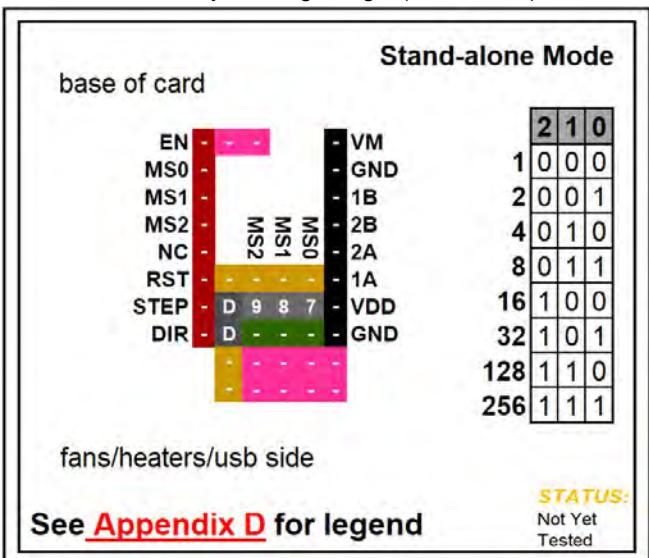
```

CTM220F_TMC_12_V1.0.0 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 also visible.

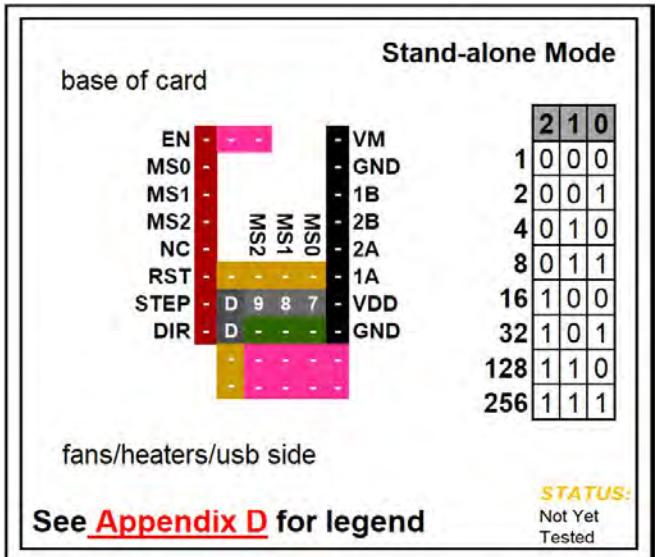
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

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

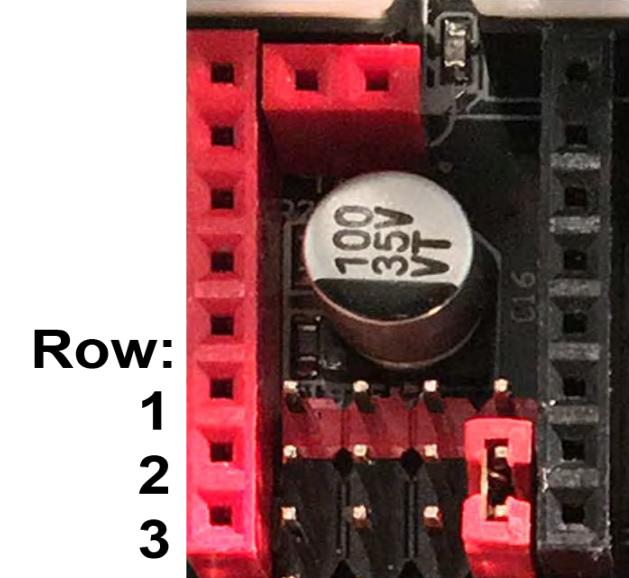
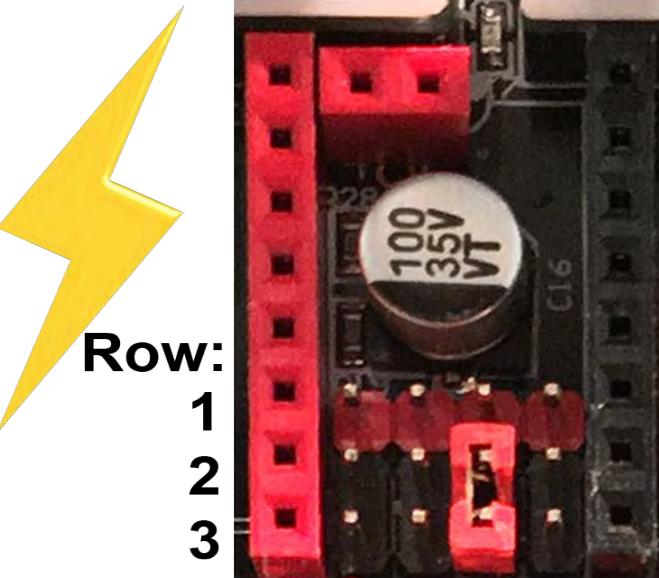
- See next page for the legend that belongs to the above chart.



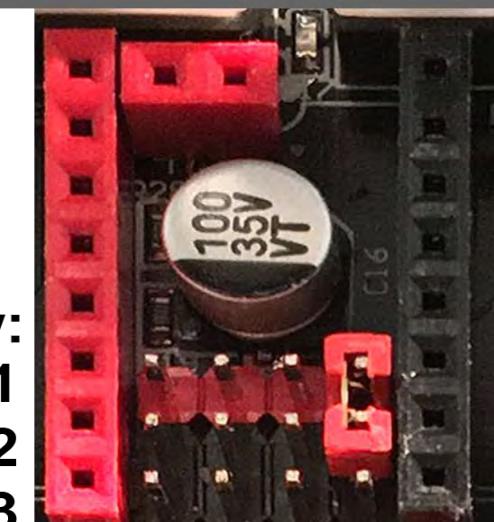
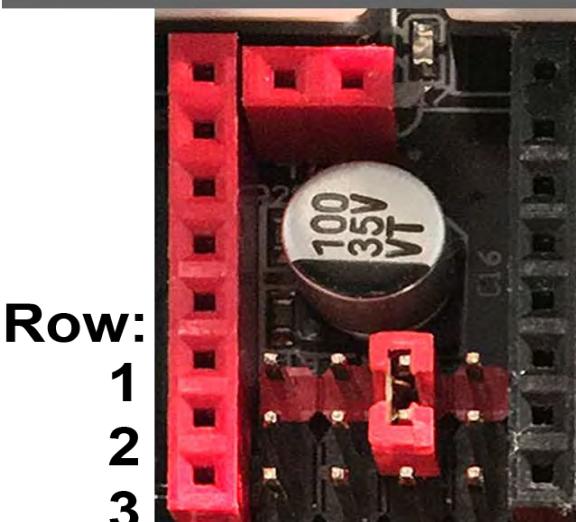
FYSETC ST820

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows 2 and 3

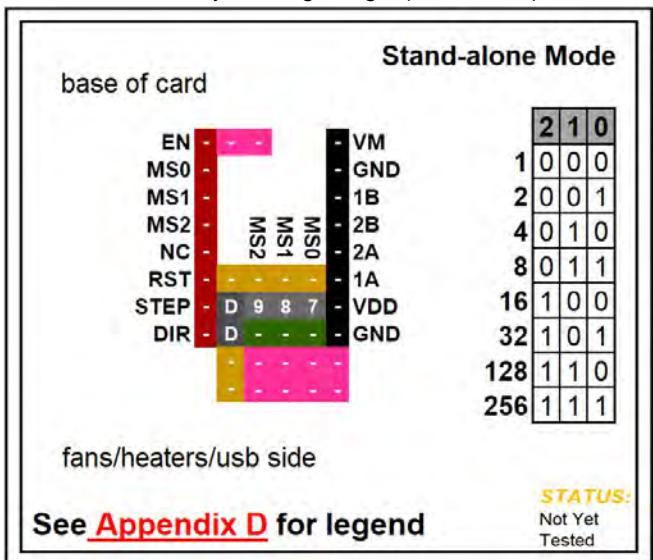


High ➡ set Jumper between rows 1 and 2



This is a
Jumper:



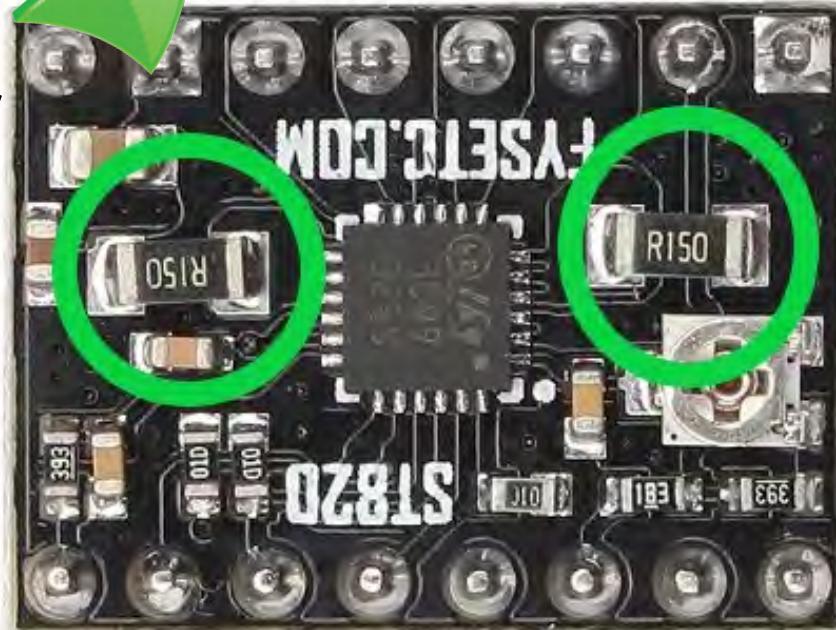


FYSETC ST820

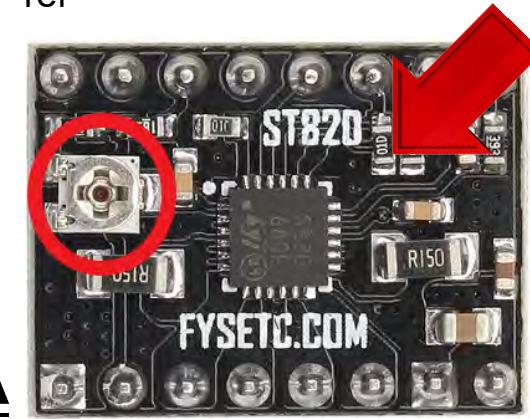
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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: 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

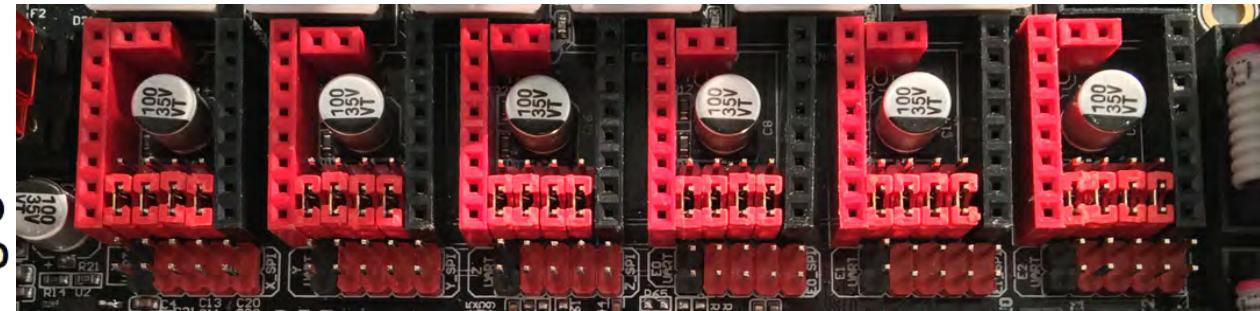
Stand-alone Mode

FYSETC ST820

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

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

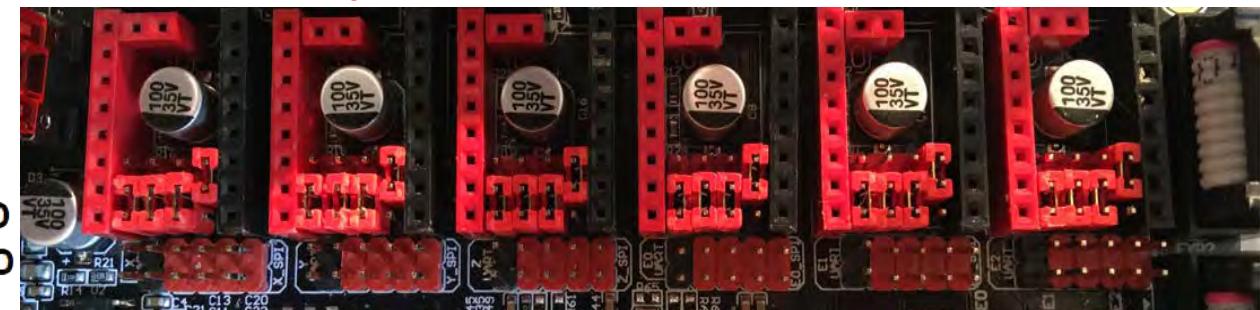
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

STEP

1 / 2

Stand-alone Mode

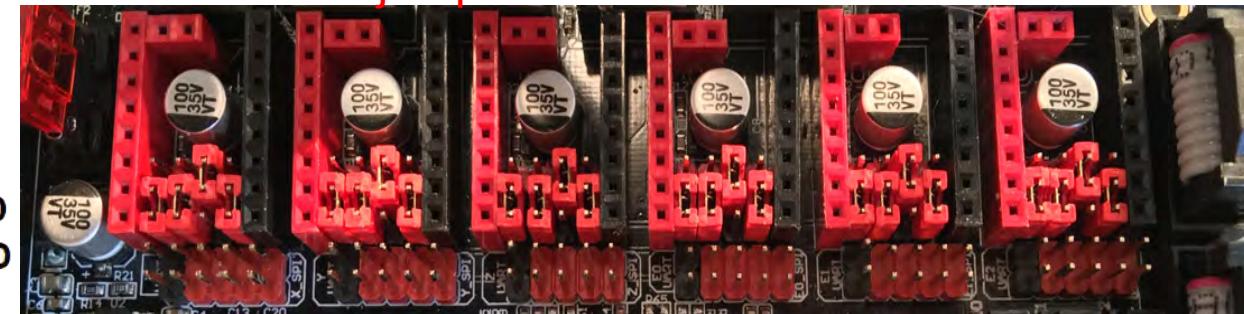
FYSETC ST820

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

1 / 4

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

Note: The "D" jumper MUST be SET!

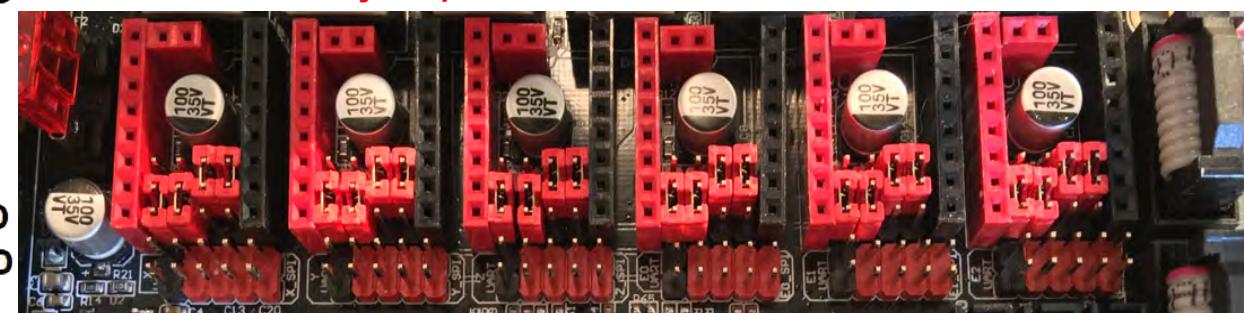


See [Appendix D](#) for legend

1 / 8

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

Note: The "D" jumper MUST be SET!

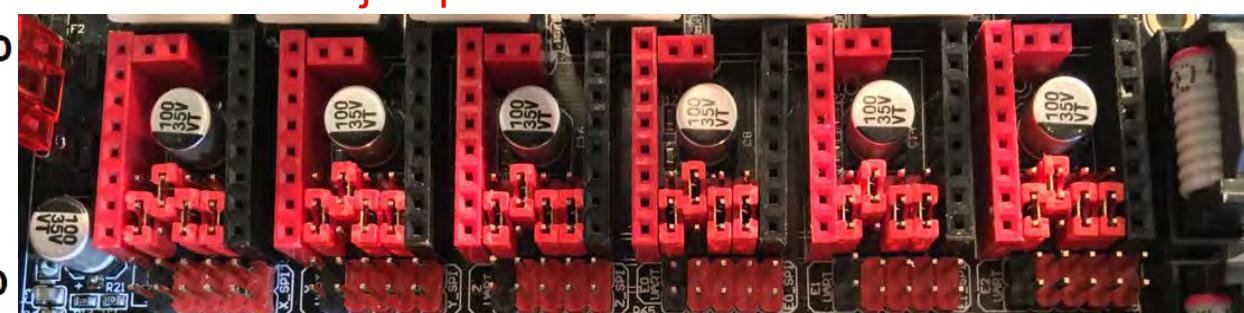


See [Appendix D](#) for legend

1 / 16

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

Stand-alone Mode

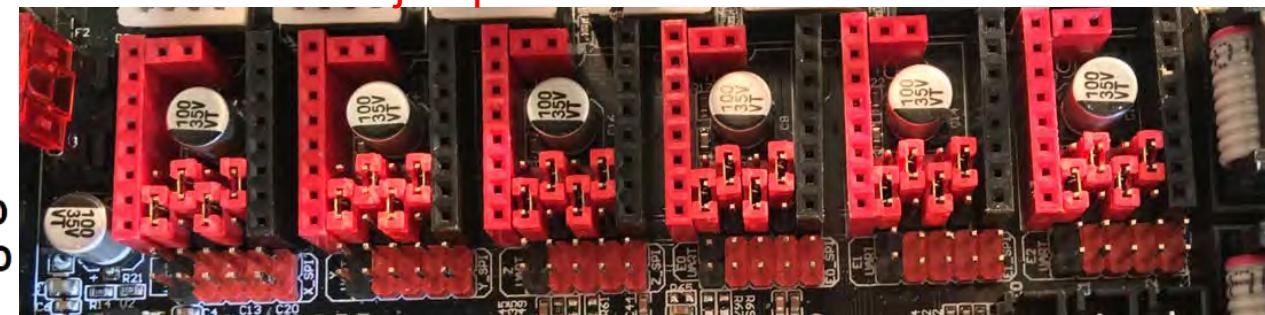
1 / 32

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

FYSETC ST820

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

Note: The "D" jumper MUST be SET!

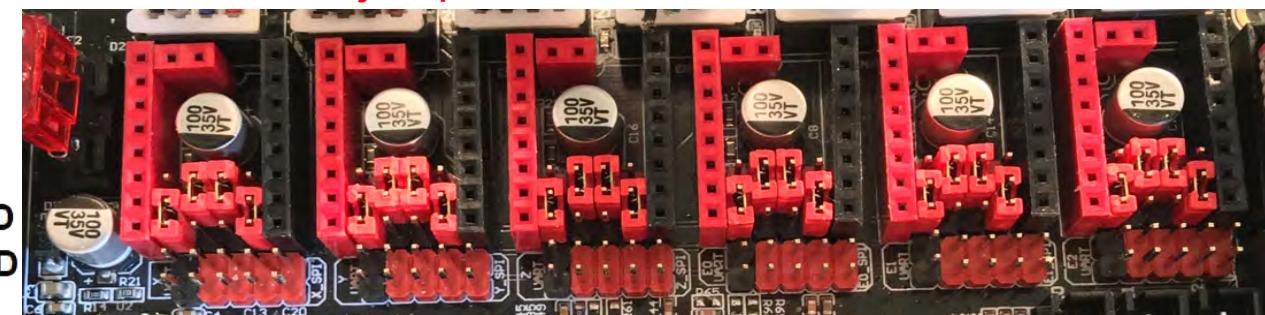


See [Appendix D](#) for legend

1 / 128

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

Note: The "D" jumper MUST be SET!

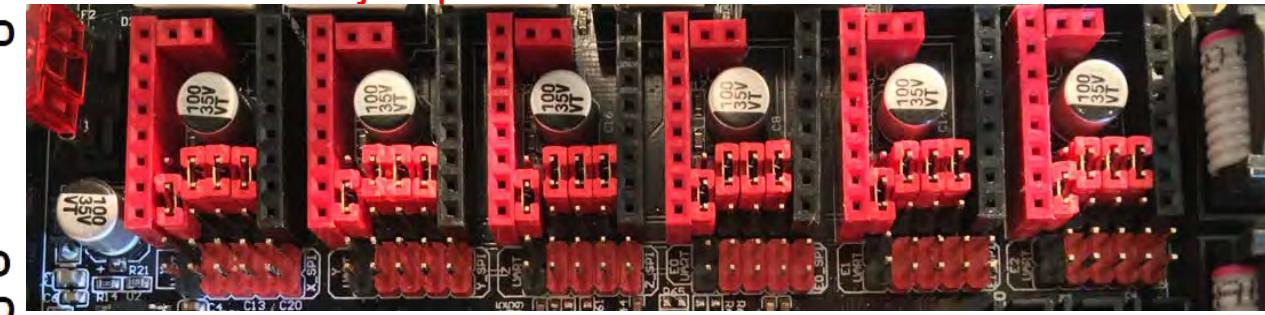


See [Appendix D](#) for legend

1 / 256

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

Note: The "D" jumper MUST be SET!

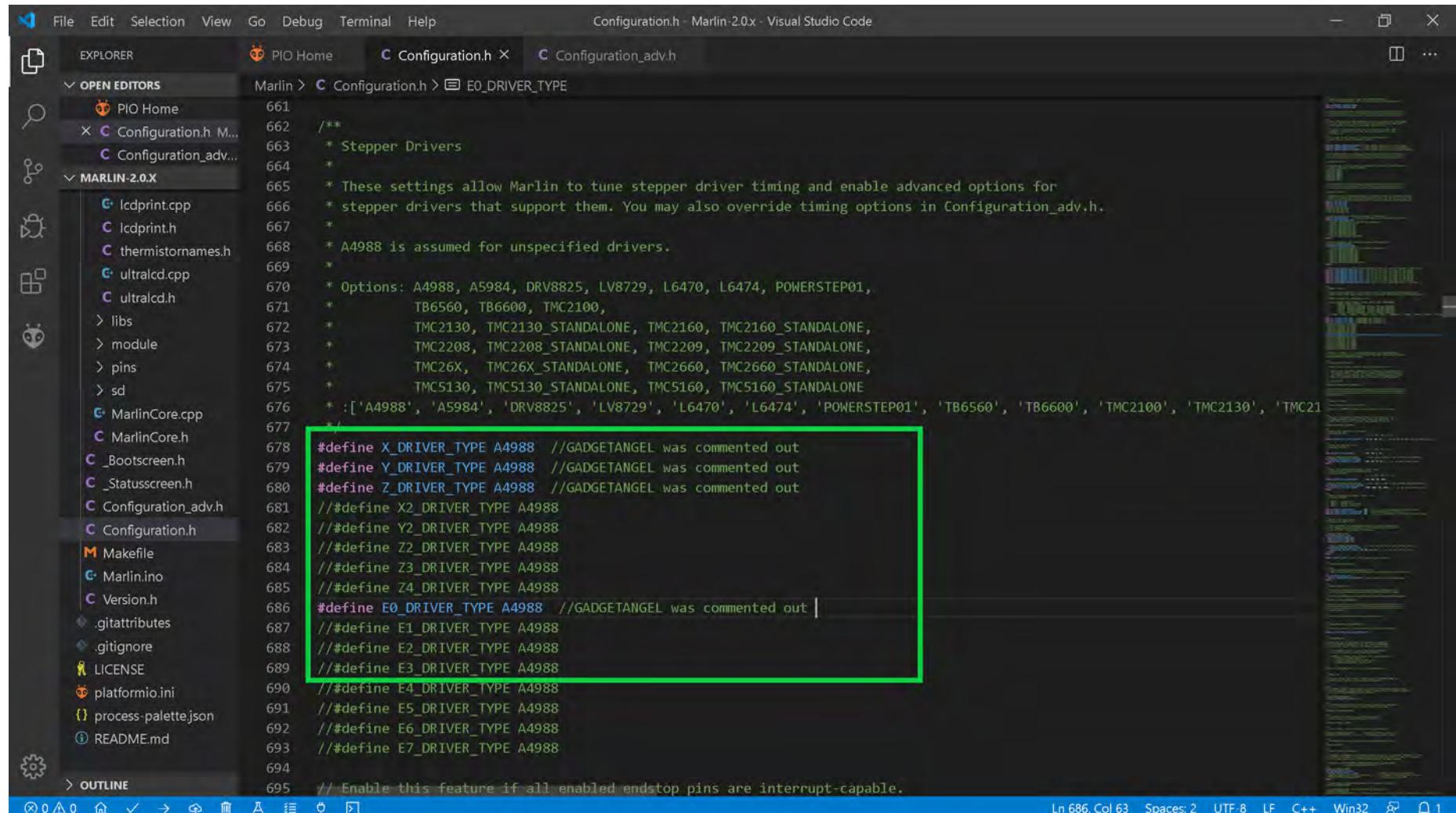


See [Appendix D](#) for legend

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 > E0_DRIVER_TYPE

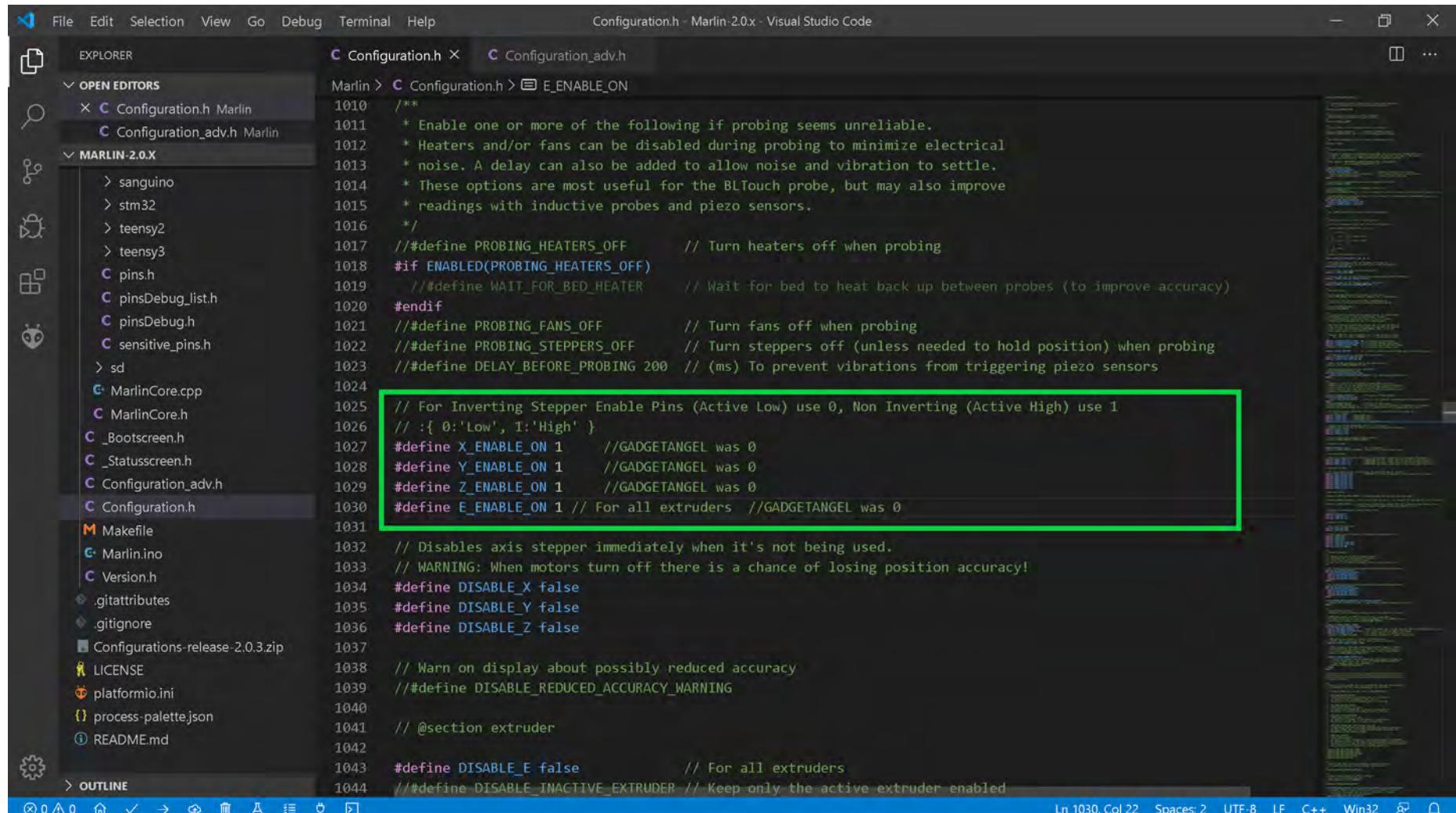
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: 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', 'TMC2660', '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.



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

```

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

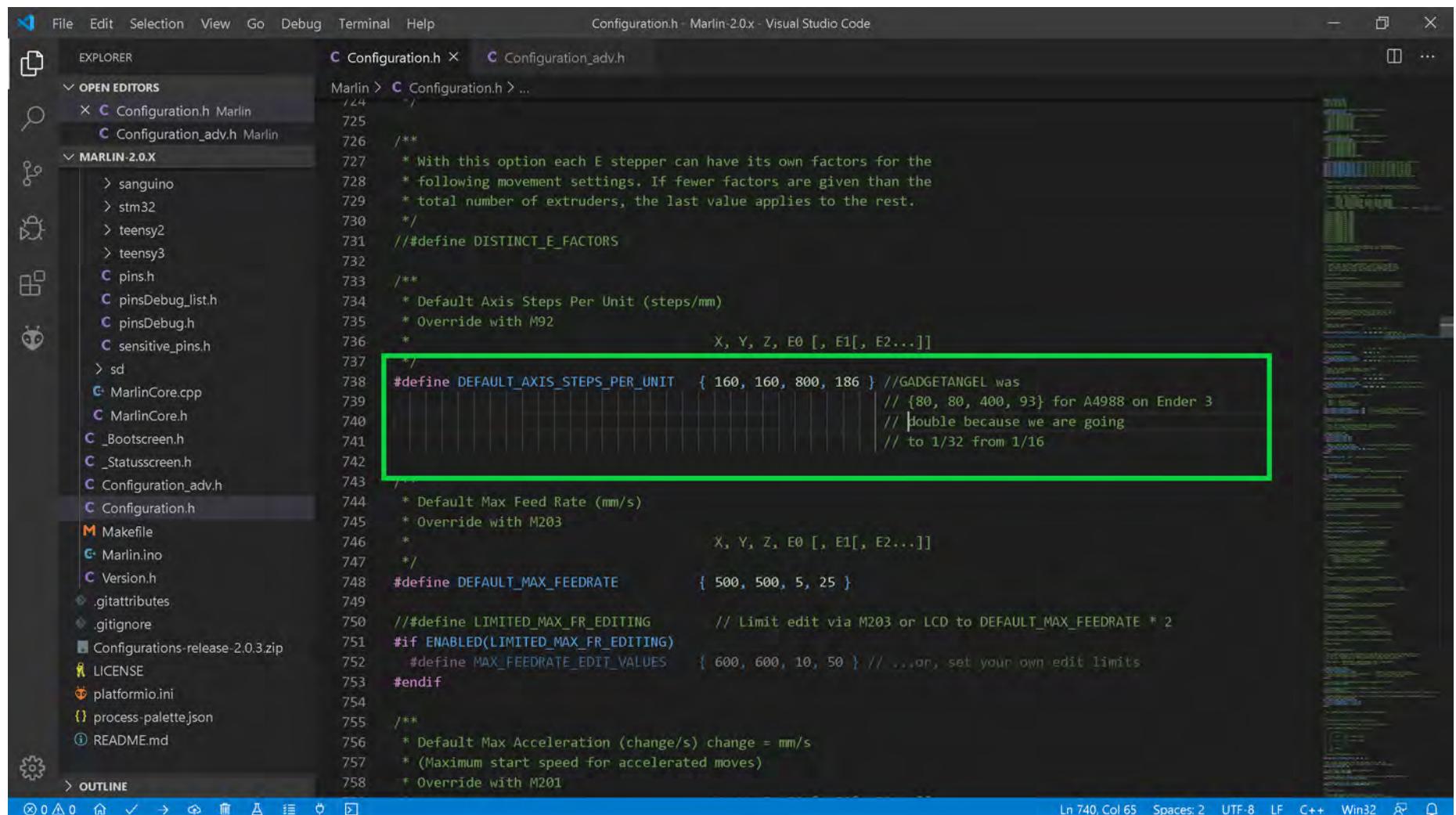
```

The rest of the code in the file includes comments about probing, heater/fan control, and axis disable logic. The status bar at the bottom indicates the code length is 1030 lines, column 22, with spaces, and the file is saved in 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 configuration header. A green rectangular box highlights the following line of code:

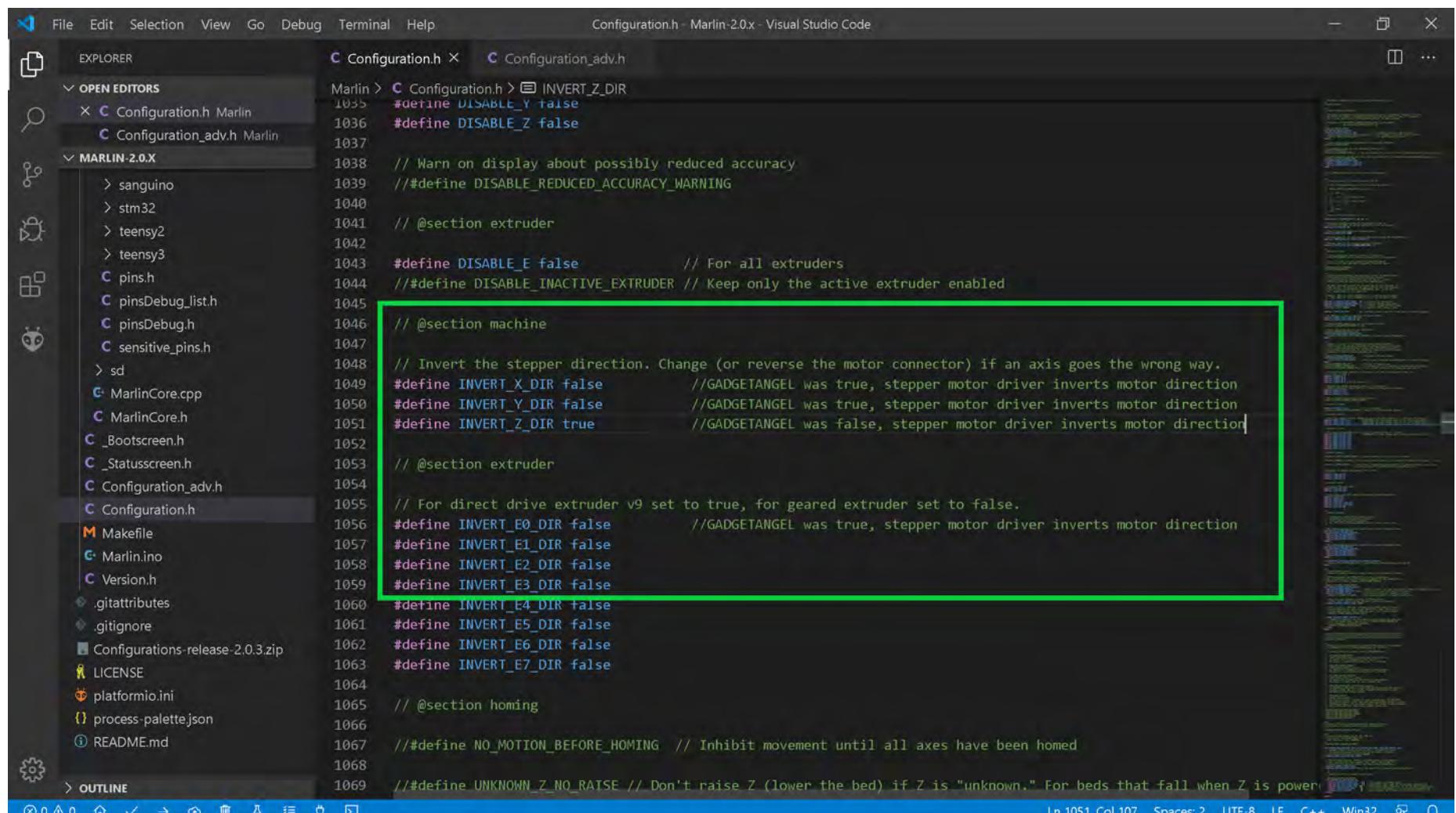
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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



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 a specific section of the code where the 'INVERT_X_DIR', 'INVERT_Y_DIR', and 'INVERT_Z_DIR' defines are located. These defines are currently set to 'false'. The code also includes comments indicating that these settings are based on GADGETANGEL values: 'false' for X and Y, and 'true' for Z.

```

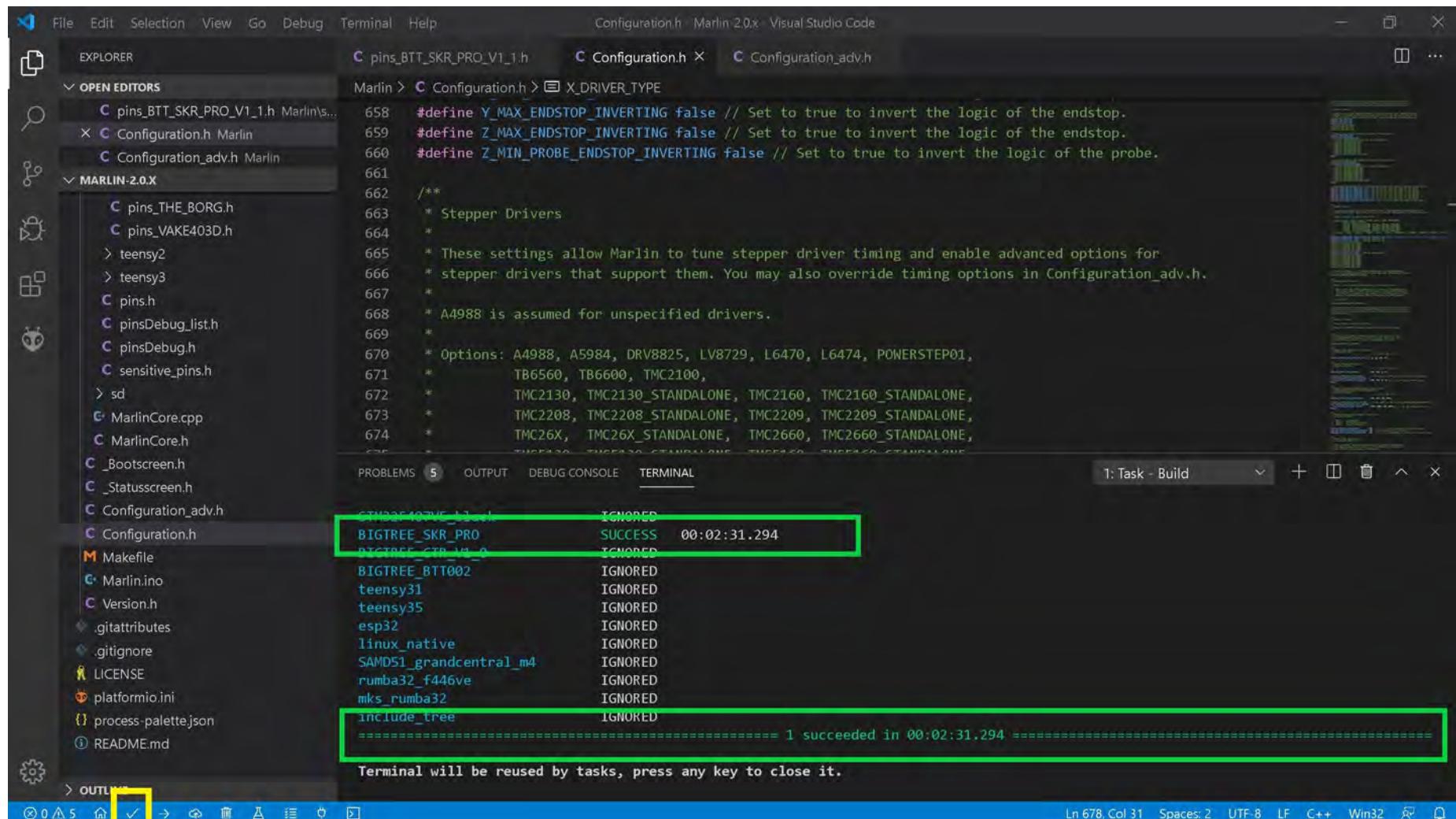
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // 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_RATE // 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.



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

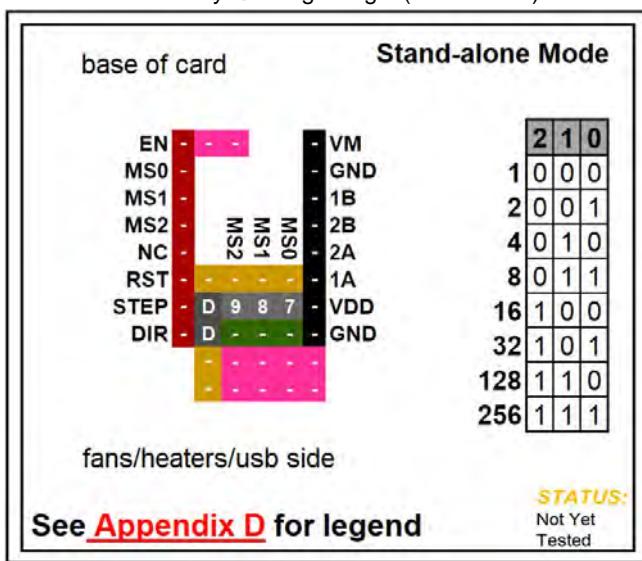
```

TMC2208_STANDALONE
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_1 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

```

A yellow box highlights the checkmark icon in the bottom-left corner of the terminal tab bar, indicating the build was successful. A green box highlights the terminal output showing the build results.

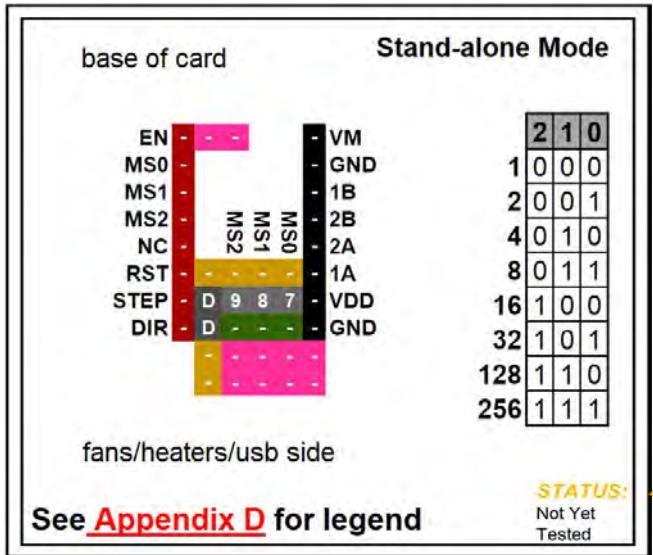
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

BIQU ST820

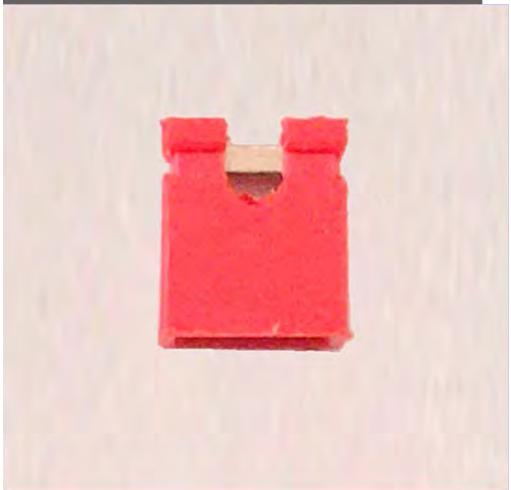
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

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 1.5A (peak)	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	$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{R_S}$		$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * R_S$		
$V_{DD} = 3.3V \text{ or } 5V \text{ DC}$ $R_S (\text{Typical Sense Resistor}) = 0.15\Omega$					

- See next page for the legend that belongs to the above chart.



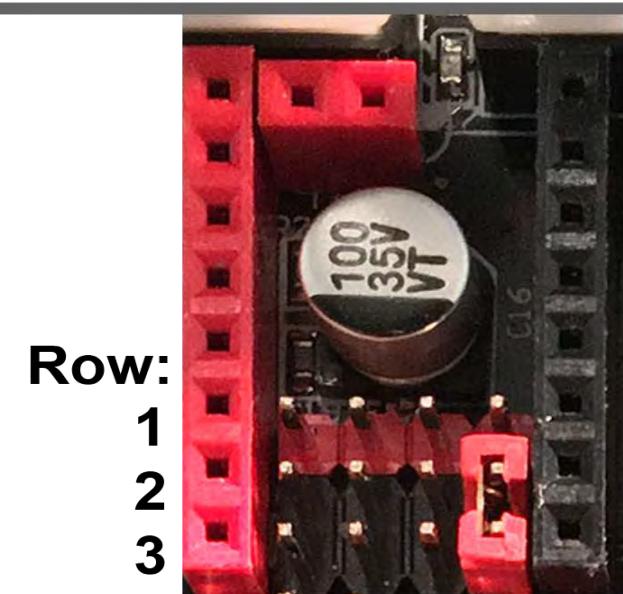
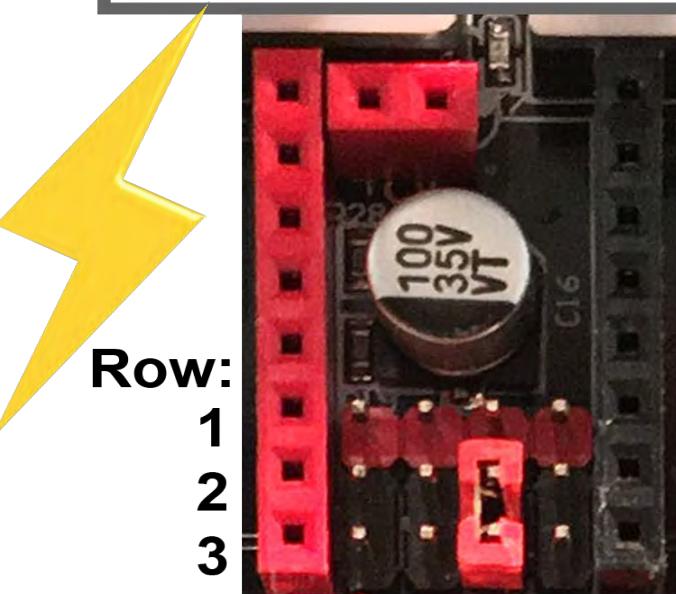
This is a
Jumper:



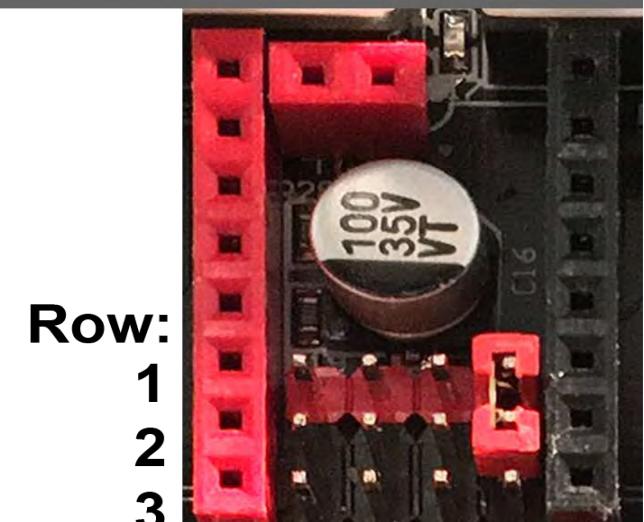
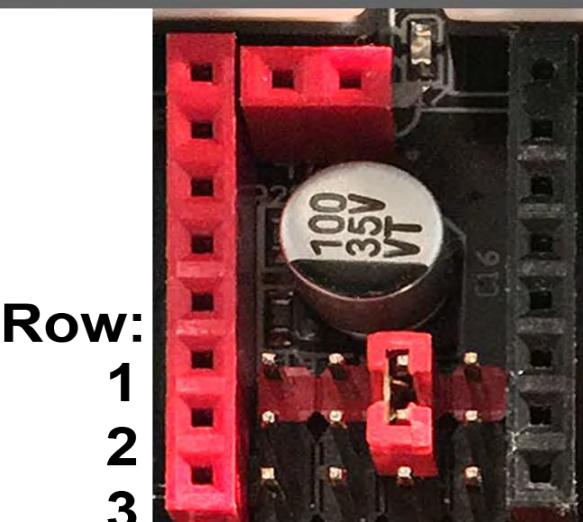
BIQU ST820

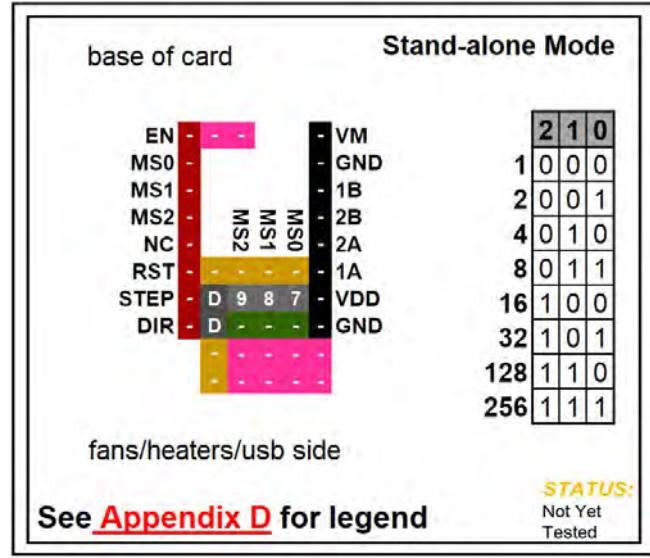
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows 2 and 3

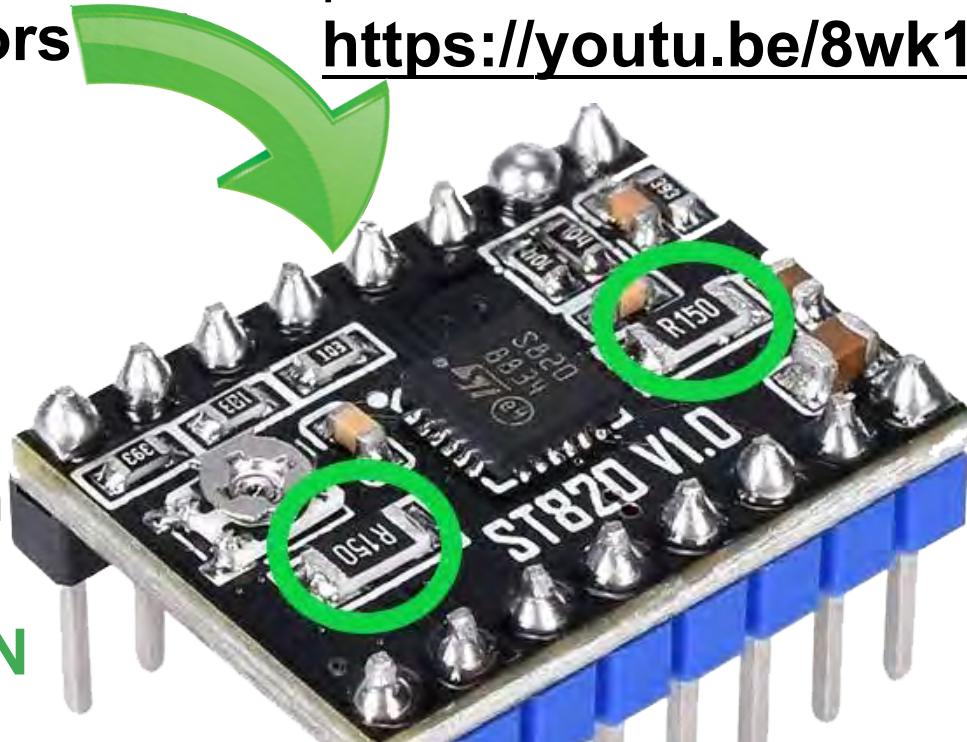


High ➡ set Jumper between rows 1 and 2





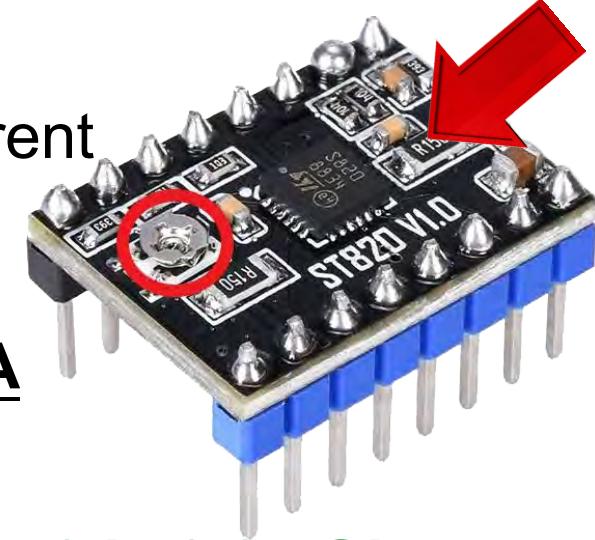
Note: Check your current sense resistors values on the driver board, as shown in GREEN



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: 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

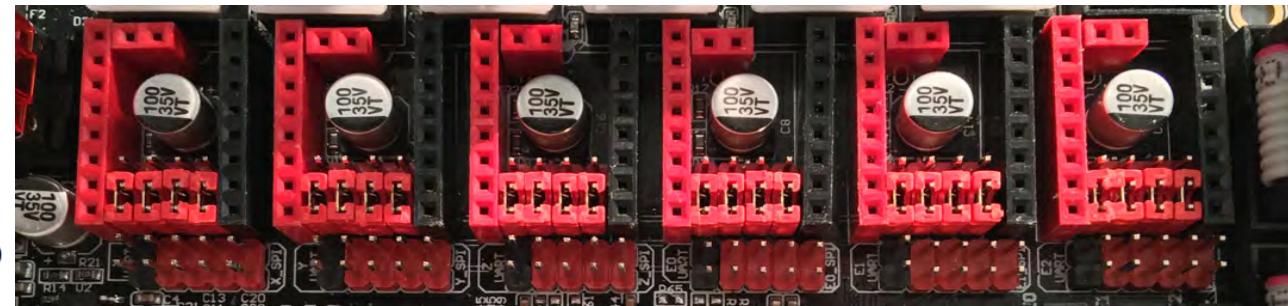
Stand-alone Mode

BIQU ST820

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

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

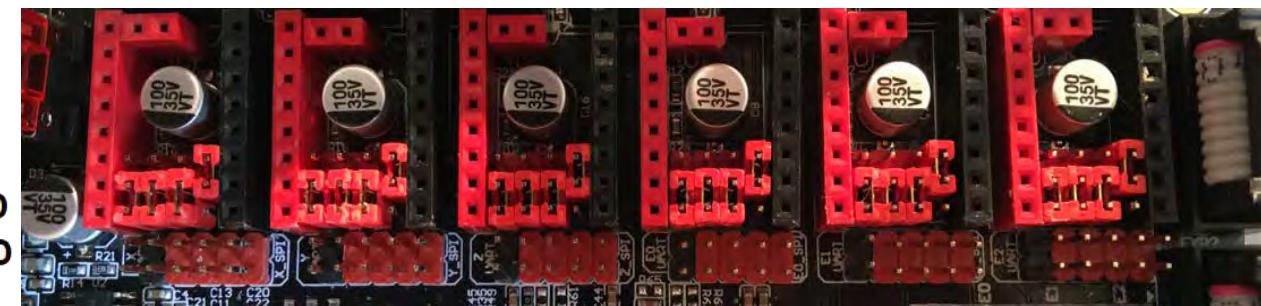
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

STEP

1 / 2

Stand-alone Mode

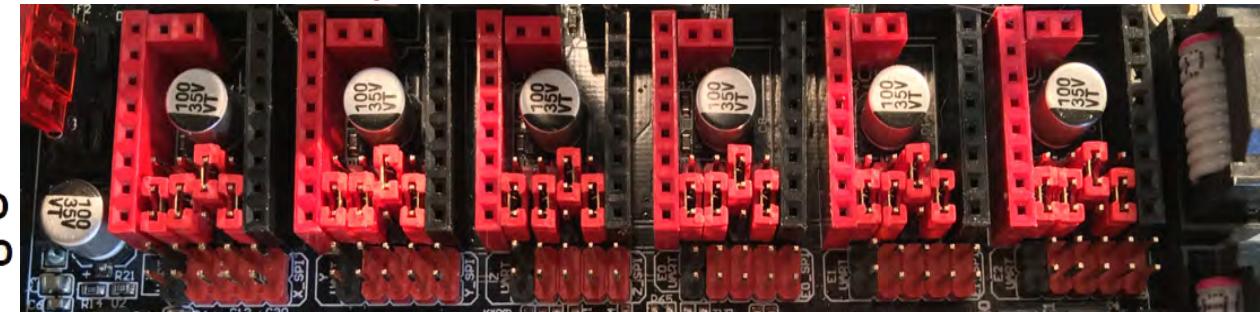
1 / 4

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

BIQU ST820

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

Note: The "D" jumper MUST be SET!

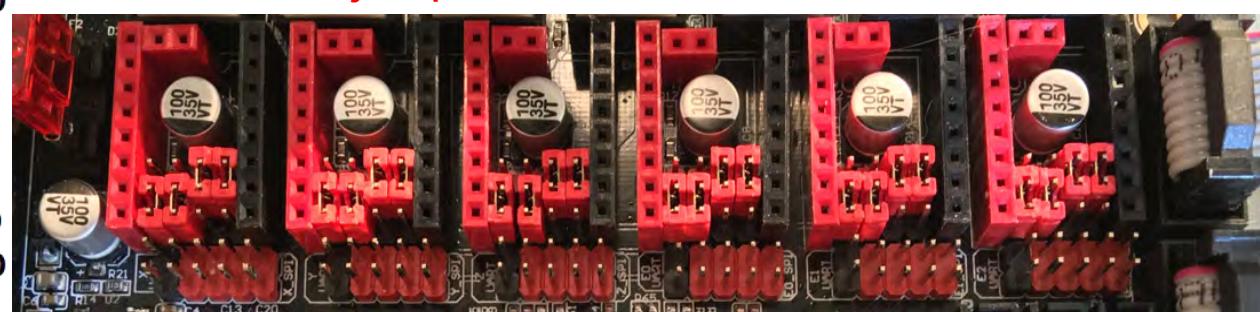


See [Appendix D](#) for legend

1 / 8

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

Note: The "D" jumper MUST be SET!

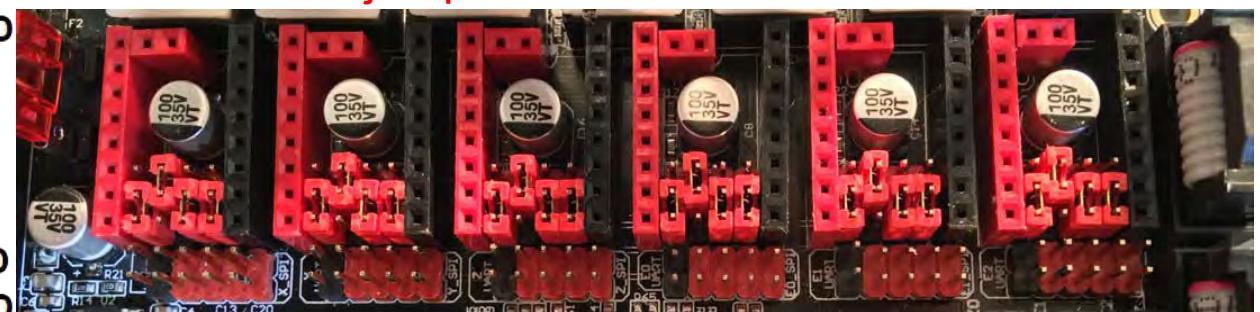


See [Appendix D](#) for legend

1 / 16

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

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

Stand-alone Mode

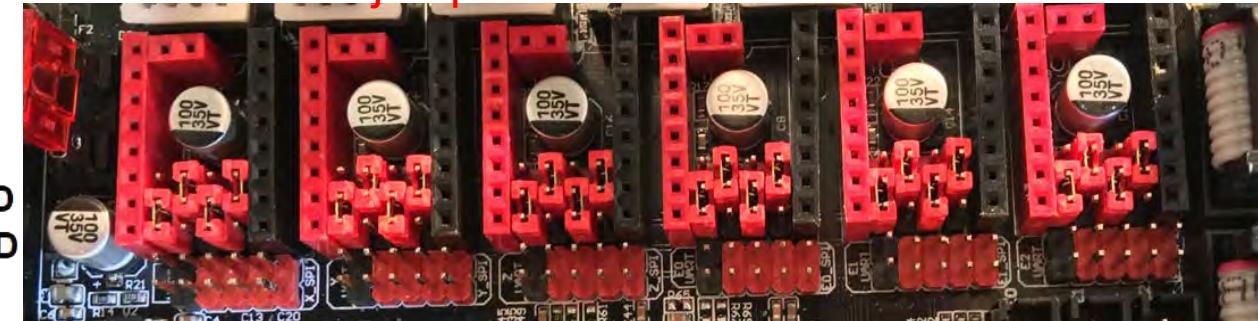
1 / 32

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

BIQU ST820

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

Note: The "D" jumper MUST be SET!

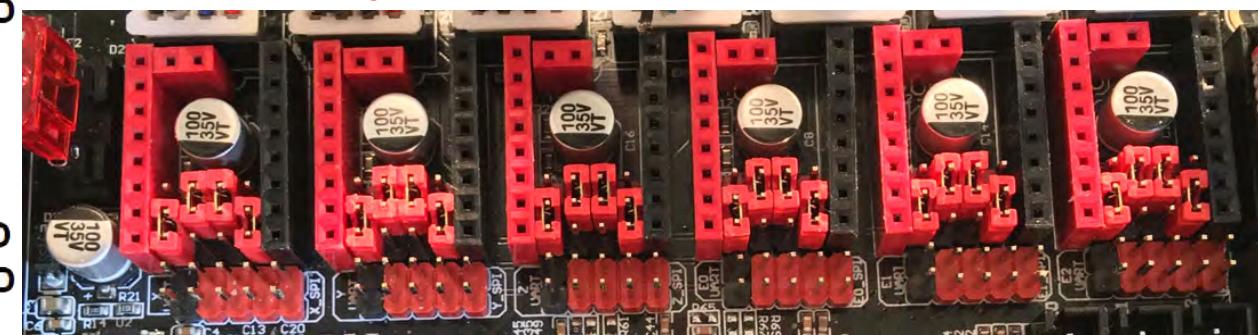


See [Appendix D](#) for legend

1 / 128

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

Note: The "D" jumper MUST be SET!

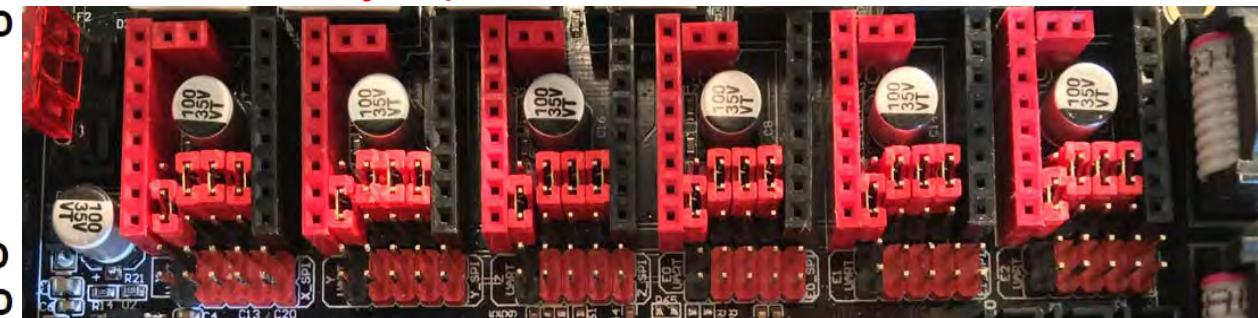


See [Appendix D](#) for legend

1 / 256

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

Note: The "D" jumper MUST be SET!

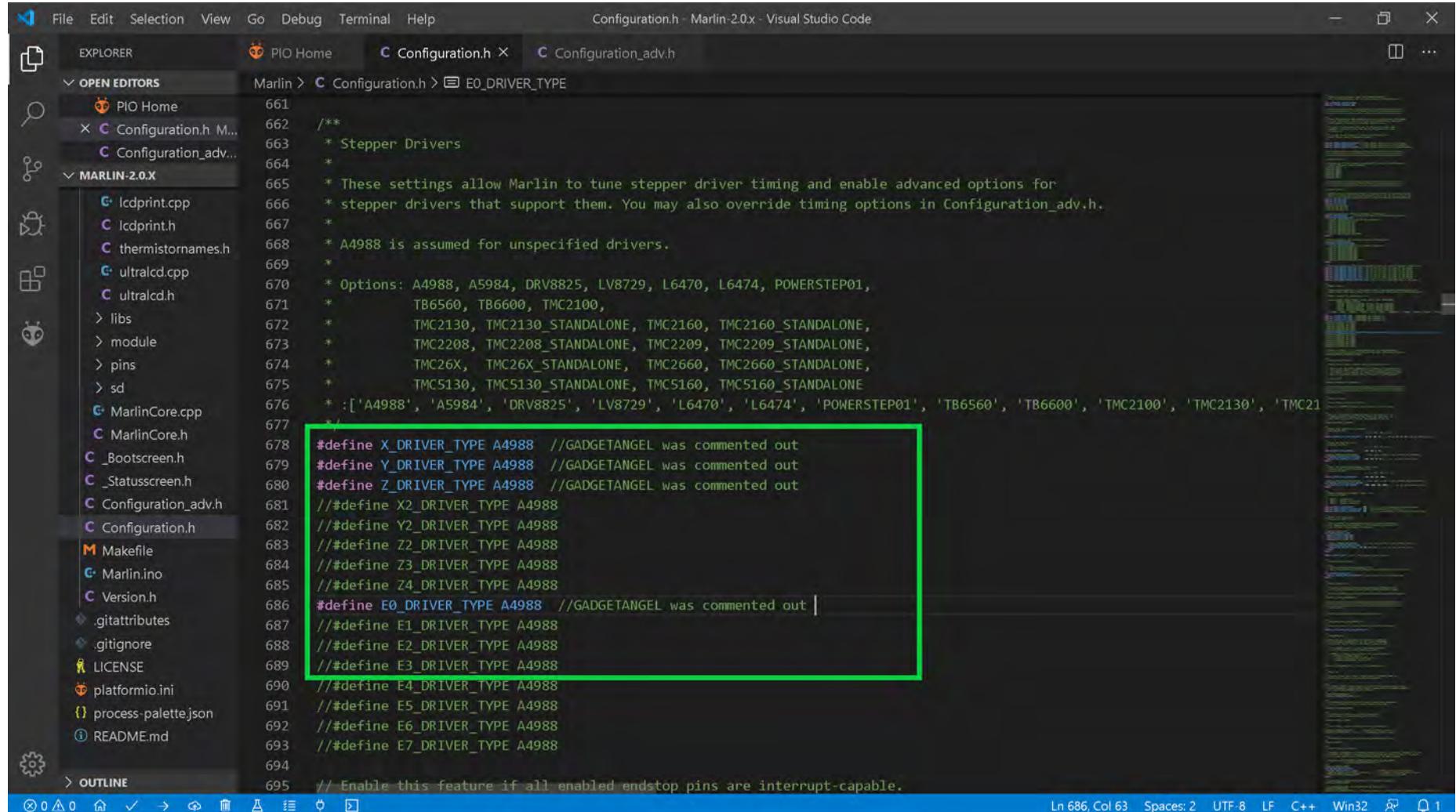


See [Appendix D](#) for legend

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.



```

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 > EO_DRIVER_TYPE
PIO Home 661 /**
Configuration.h M... 662 */
Configuration_adv.h 663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160',
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.
Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 Q 1

```

- 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 OPEN EDITORS Configuration.h Configuration_adv.h

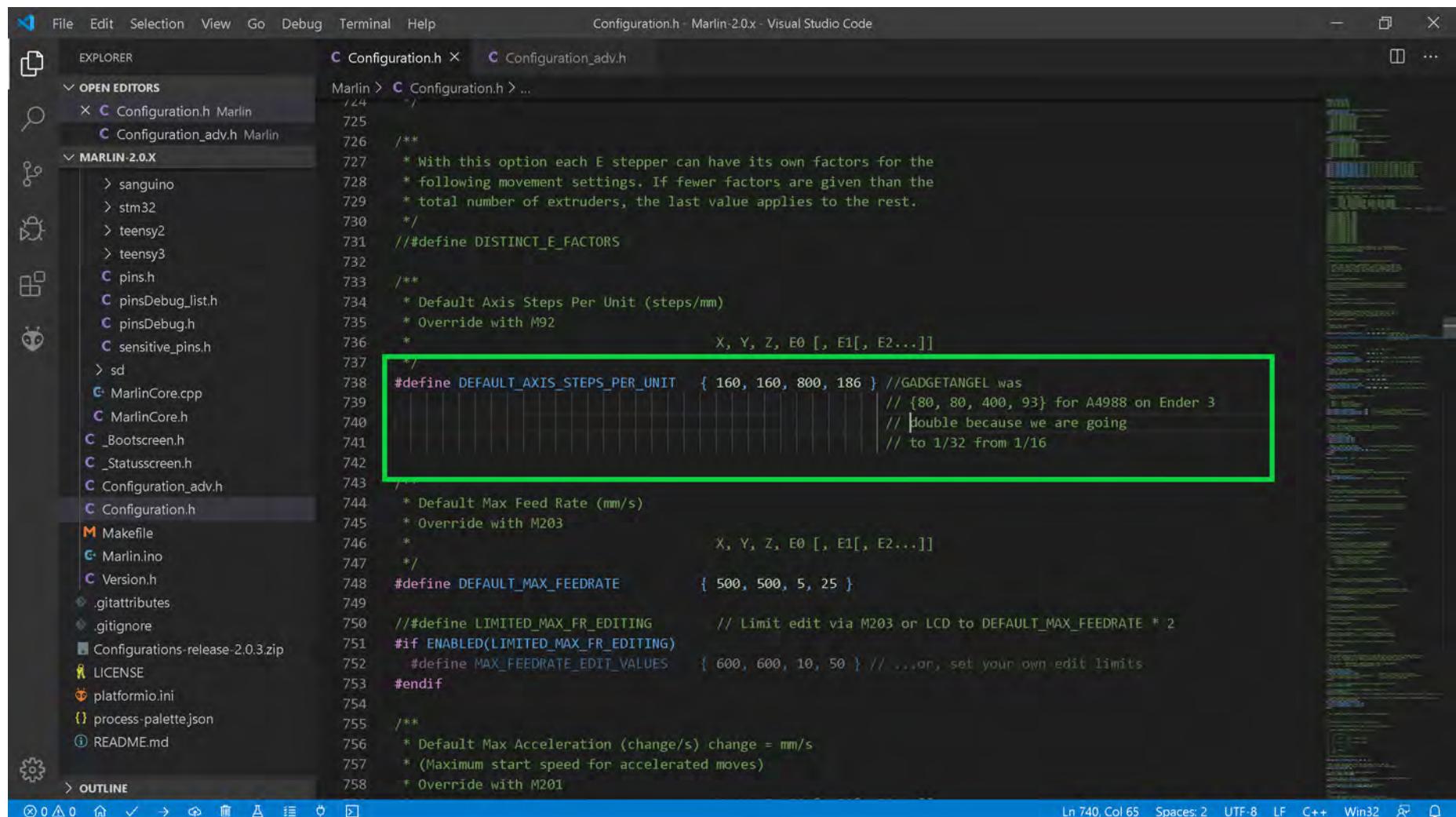
Marlin > Configuration.h > E_ENABLE_ON

```
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
```

- 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:

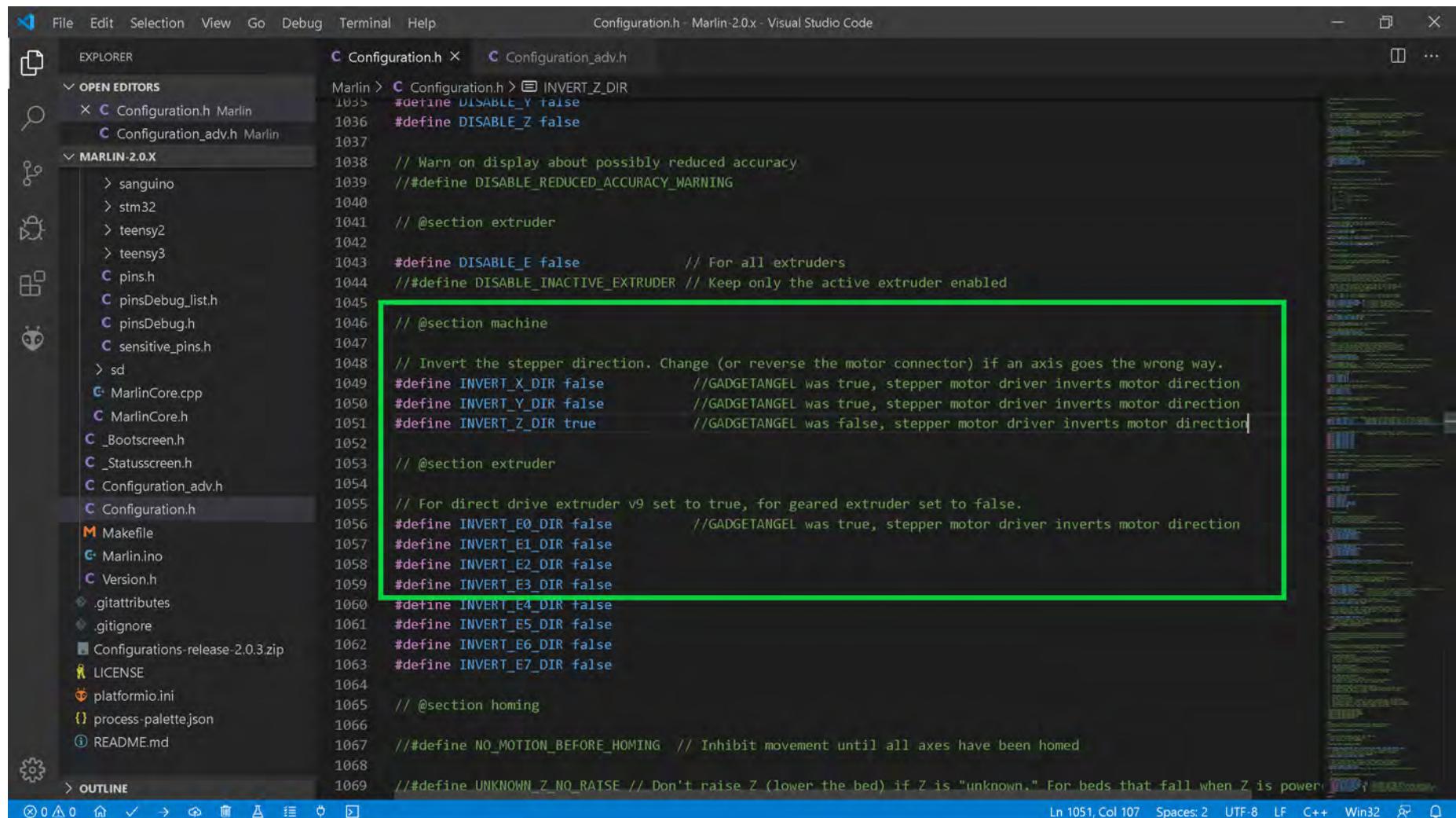
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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



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

```

A green rectangular box highlights the code block from line 1049 to line 1063, which defines the `INVERT_X_DIR` through `INVERT_E7_DIR` macros. The code within this box is annotated with comments explaining the condition for each macro based on the GADGETANGEL configuration.

- Go to the next page.

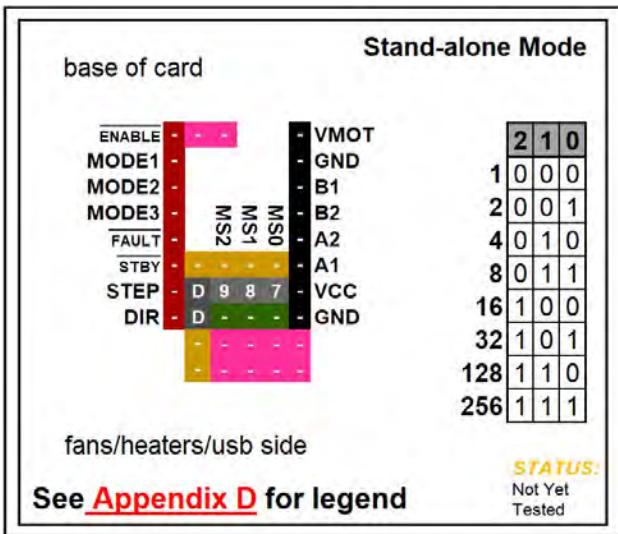
The (latest release of) Marlin Setup for BIQU ST820 Drivers

- 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".
- Code Editor:** Displays the "Configuration.h" file with code related to endstop inversion and stepper driver timing.
- Terminal:** Shows the build output for "BIGTREETECH_V1.1" and "BIGTREETECH_V1.0". The output indicates a success message: "1 succeeded in 00:02:31.294".
- Status Bar:** Shows the current line (Ln 678), column (Col 31), and encoding (UTF-8).

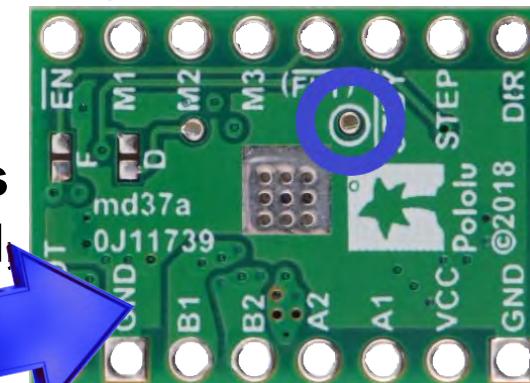
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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



POLOLU ST820 (STSPIN820)

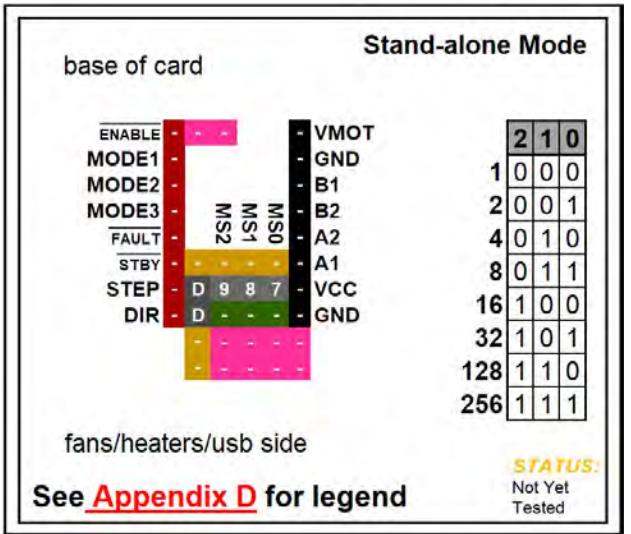
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

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
	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	$I_{MAX} = V_{ref} * \frac{1}{R_S}$		$V_{ref} = I_{MAX} * R_S$		
R_S (Typical Sense Resistor) = 0.2Ω					

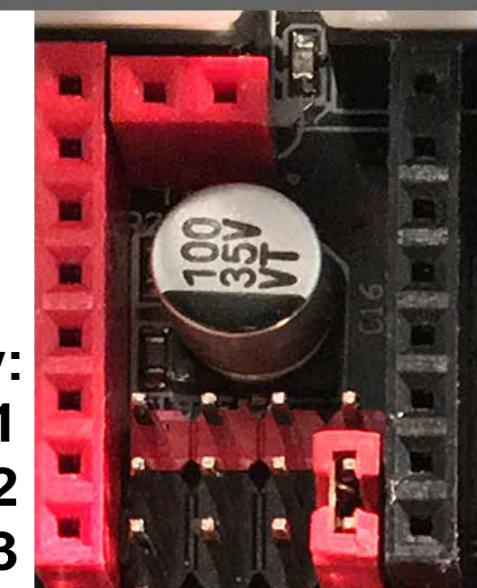
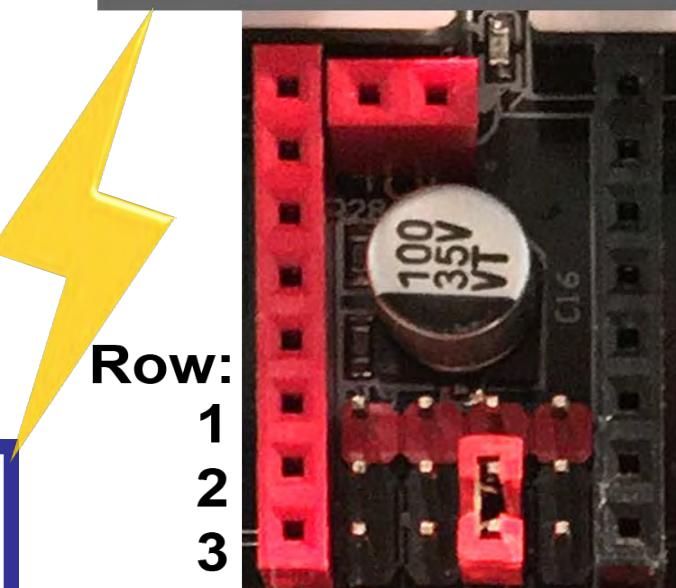
- See next page for the legend that belongs to the above chart.



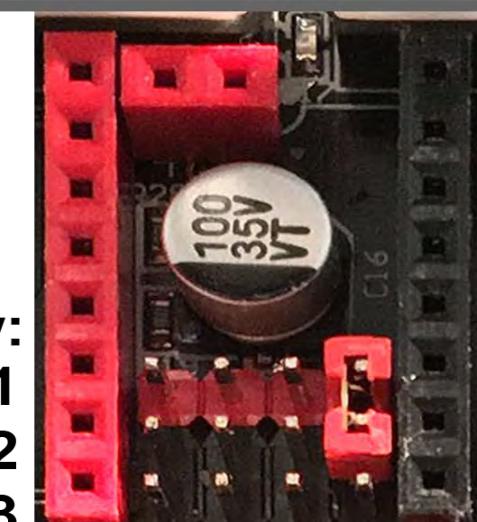
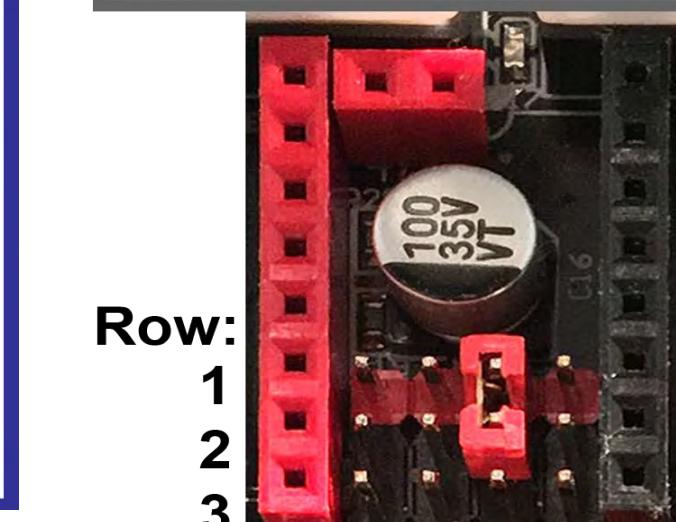
POLOLU ST820 (STSPIN820)

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➤ set Jumper between rows 2 and 3

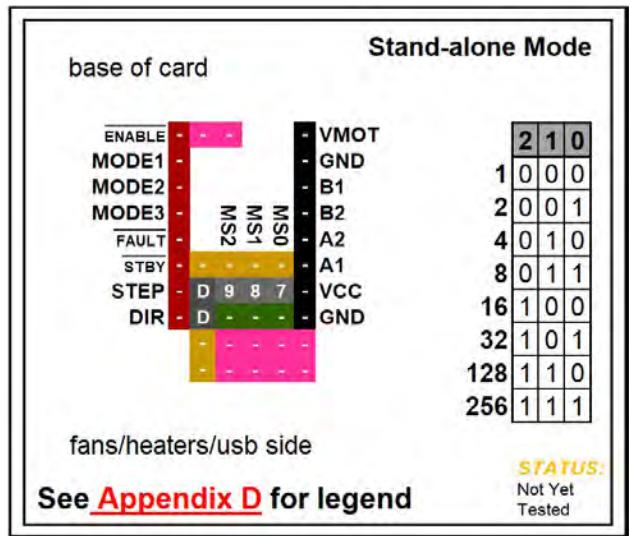


High ➤ set Jumper between rows 1 and 2



This is a
Jumper:



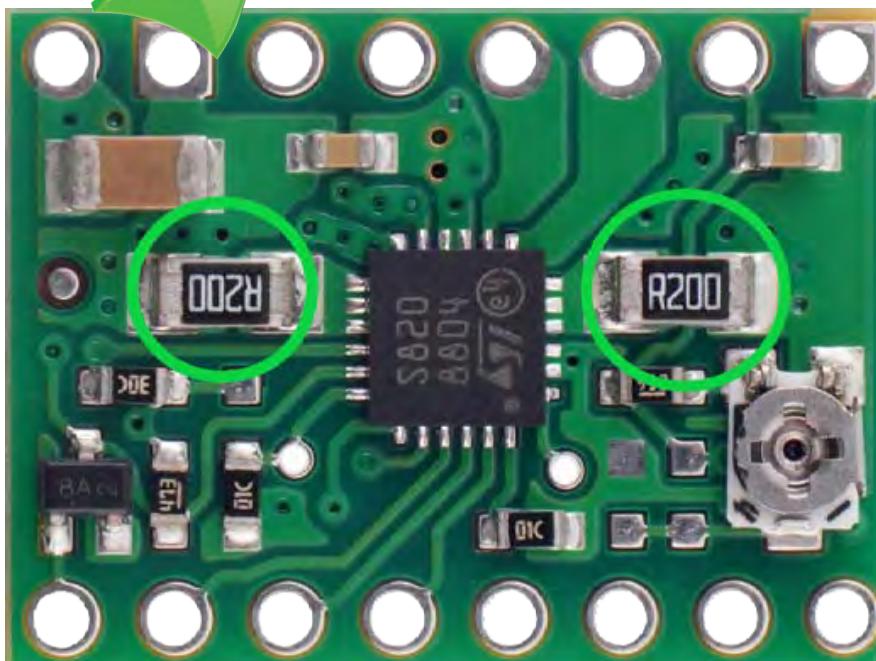


POLOLU ST820 (STSPIN820)

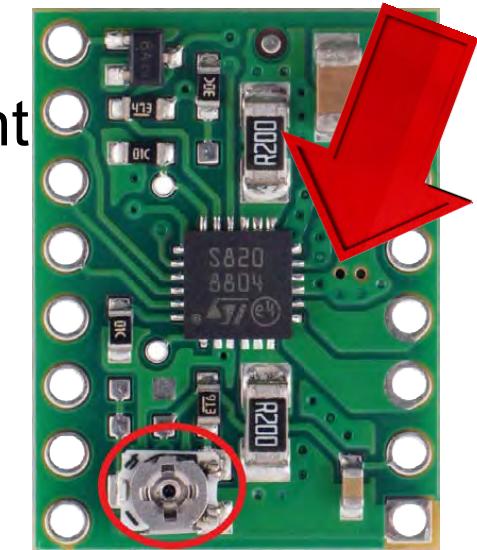
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

POLOLU ST820 (STSPIN820)

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

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	-	-	B2
FAULT	-	-	-	A2
STBY	-	-	-	A1
STEP	D	9	8	VCC
DIR	D	9	8	GND

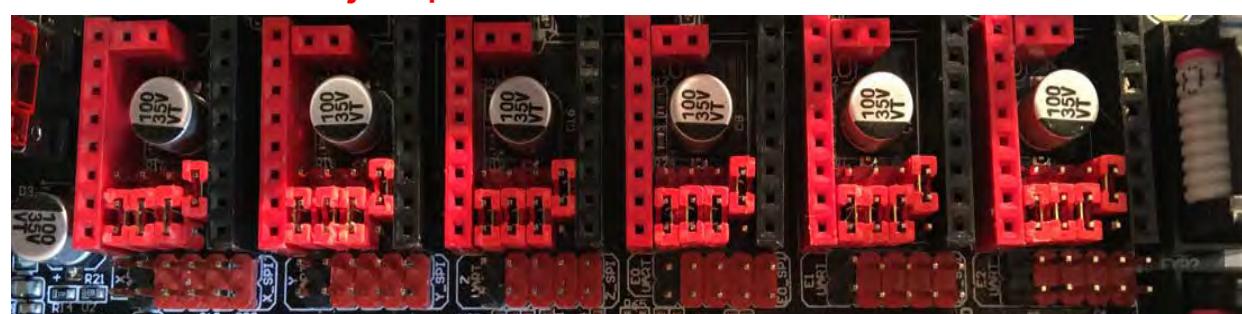
Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	-	-	B2
FAULT	-	-	-	A2
STBY	-	-	-	A1
STEP	D	9	8	VCC
DIR	D	9	8	GND

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

STEP

1 / 2

Stand-alone Mode

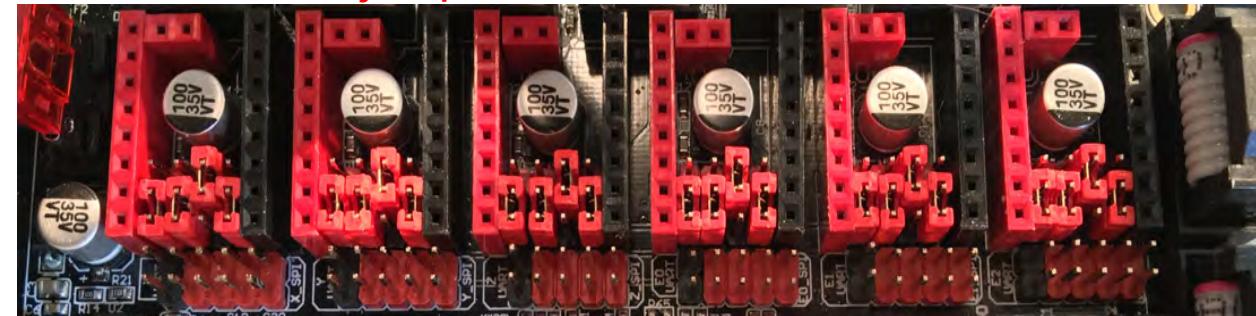
1 / 4

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	-	MS2	B2
FAULT	-	-	MS1	A2
STBY	-	-	MS0	A1
STEP	D	9	8	VCC
DIR	D	9	-	GND

POLOLU ST820 (STSPIN820)

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

Note: The "D" jumper MUST be SET!

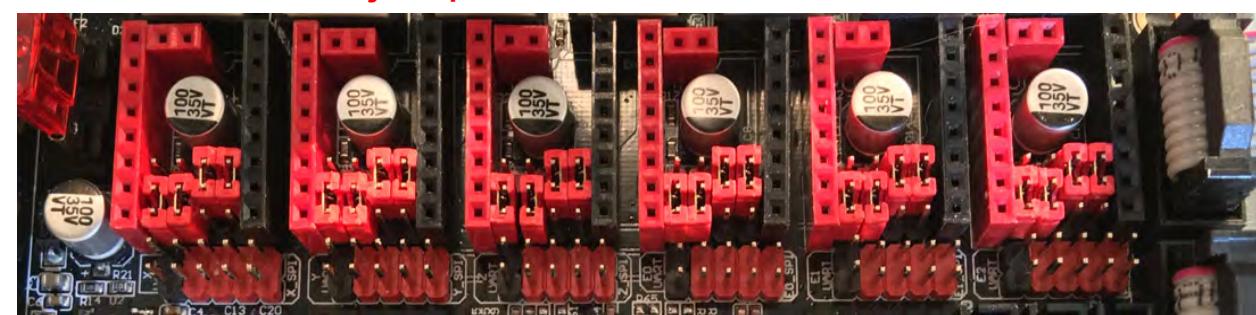


See [Appendix D](#) for legend

1 / 8

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	-	MS2	B2
FAULT	-	-	MS1	A2
STBY	-	-	MS0	A1
STEP	D	9	8	VCC
DIR	D	9	-	GND

Note: The "D" jumper MUST be SET!

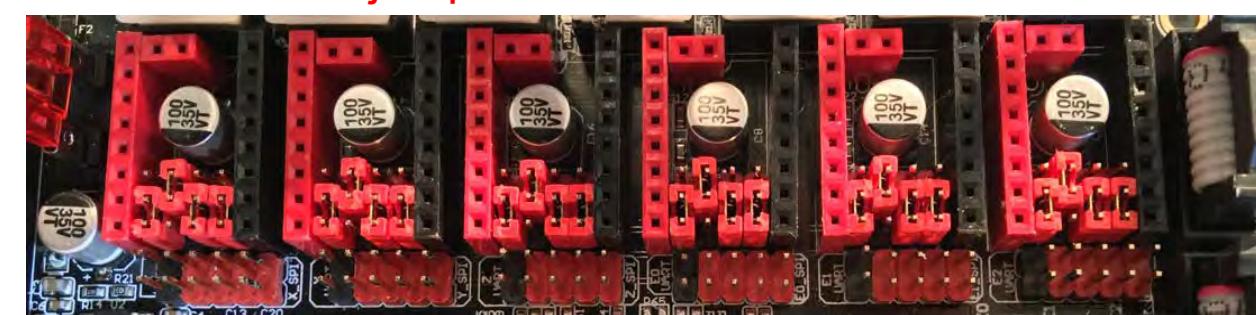


See [Appendix D](#) for legend

1 / 16

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	-	MS2	B2
FAULT	-	-	MS1	A2
STBY	-	-	MS0	A1
STEP	D	9	8	VCC
DIR	D	9	-	GND

Note: The "D" jumper MUST be SET!



See [Appendix D](#) for legend

Stand-alone Mode

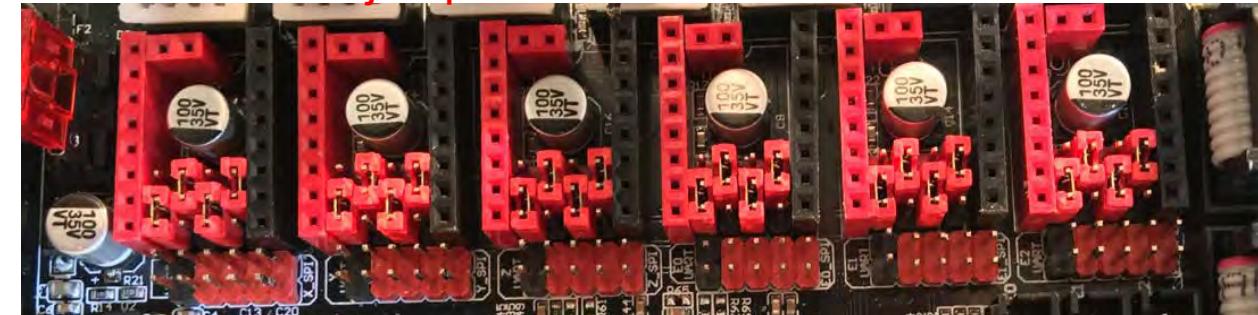
1 / 32

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	MS2	MS1	MS0
FAULT	-	-	-	A2
STBY	-	9	7	-
STEP	D	9	8	7
DIR	D	-	8	-
	-	-	-	GND

POLOLU ST820 (STSPIN820)

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

Note: The "D" jumper MUST be SET!

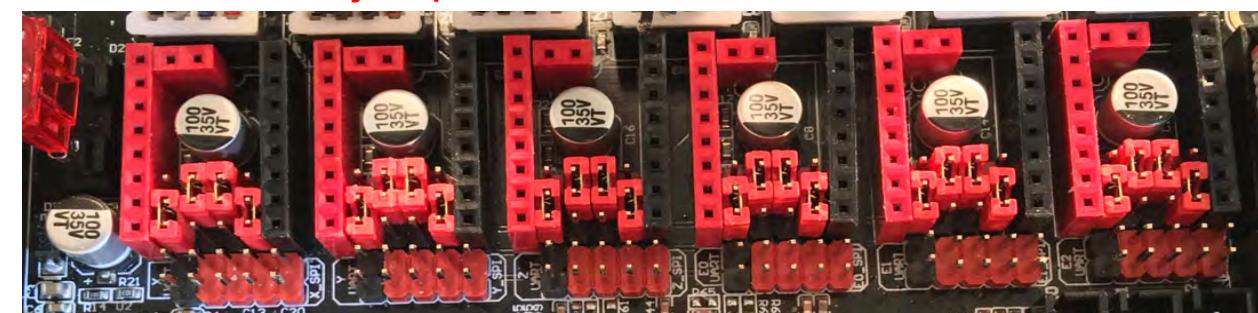


See [Appendix D](#) for legend

1 / 128

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	MS2	MS1	MS0
FAULT	-	-	-	A2
STBY	-	9	8	-
STEP	D	9	8	7
DIR	D	-	7	-
	-	-	-	GND

Note: The "D" jumper MUST be SET!

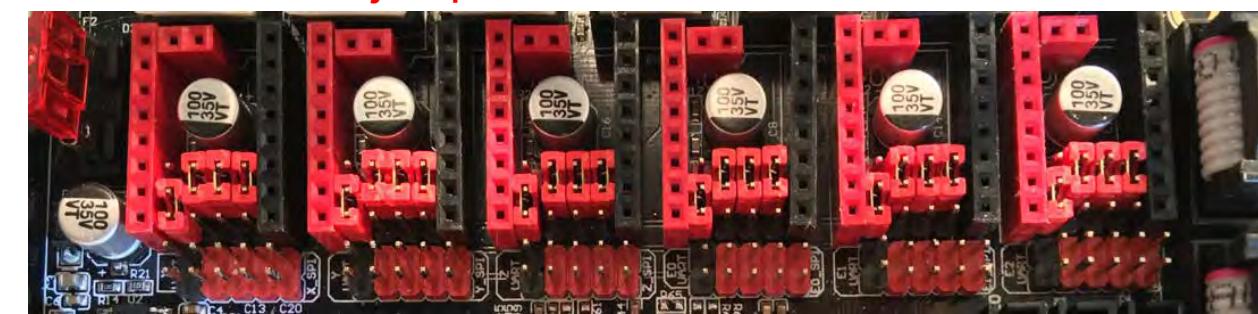


See [Appendix D](#) for legend

1 / 256

ENABLE	-	-	-	VMOT
MODE1	-	-	-	GND
MODE2	-	-	-	B1
MODE3	-	MS2	MS1	MS0
FAULT	-	-	-	A2
STBY	-	9	8	7
STEP	D	9	8	7
DIR	D	-	-	-
	-	-	-	GND

Note: The "D" jumper MUST be SET!

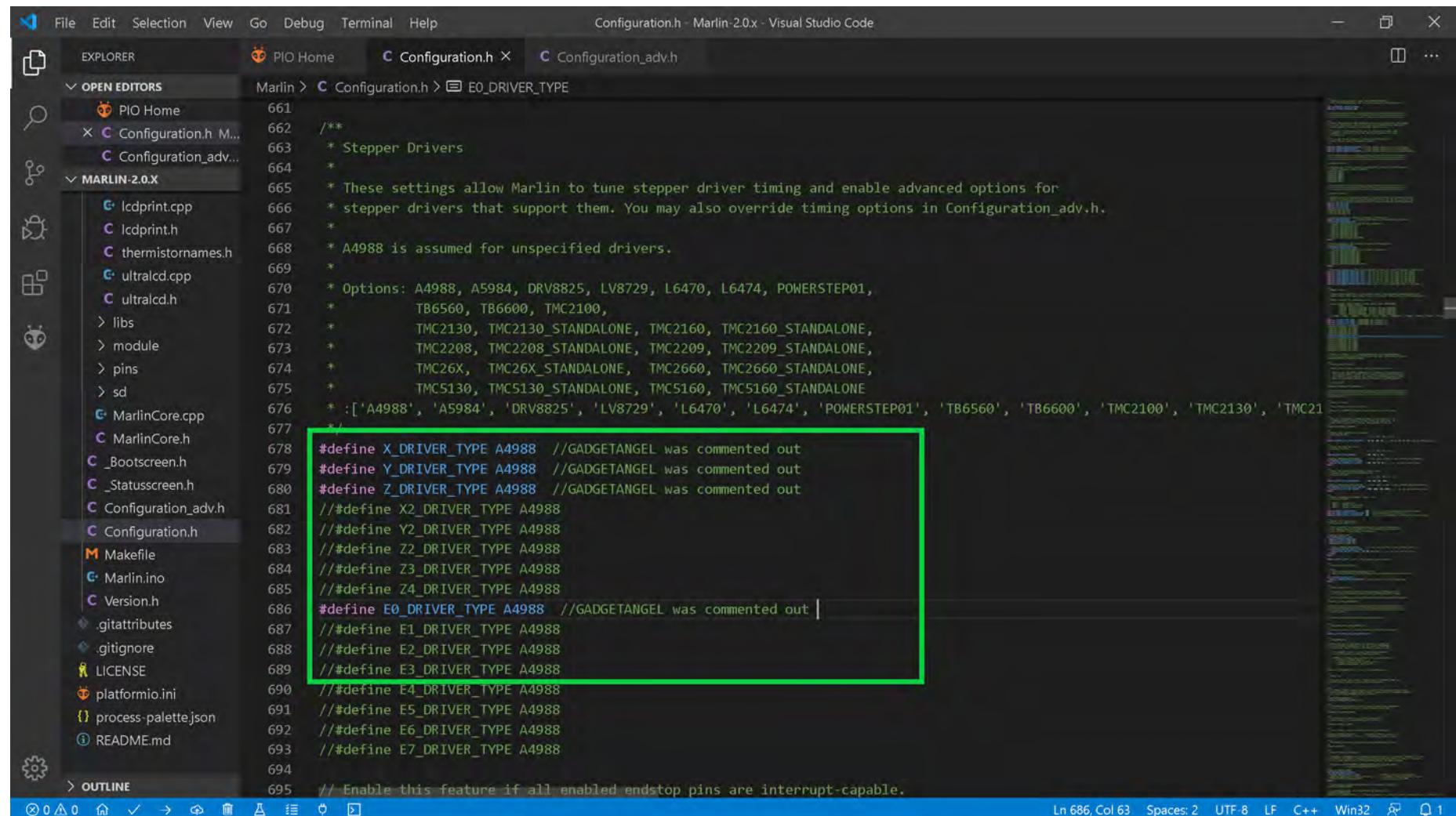


See [Appendix D](#) for legend

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.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet under the `E0_DRIVER_TYPE` section:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160',
676 * 'TMC2208', 'TMC2208_STANDALONE', 'TMC2209', 'TMC2209_STANDALONE',
677 * 'TMC26X', 'TMC26X_STANDALONE', 'TMC2660', 'TMC2660_STANDALONE',
678 * 'TMC5130', 'TMC5130_STANDALONE', 'TMC5160', 'TMC5160_STANDALONE']
679 */
680 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
681 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
682 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
683 //#define X2_DRIVER_TYPE A4988
684 //#define Y2_DRIVER_TYPE A4988
685 //#define Z2_DRIVER_TYPE A4988
686 //#define Z3_DRIVER_TYPE A4988
687 //#define Z4_DRIVER_TYPE A4988
688 #define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out
689 //#define E1_DRIVER_TYPE A4988
690 //#define E2_DRIVER_TYPE A4988
691 //#define E3_DRIVER_TYPE A4988
692 //#define E4_DRIVER_TYPE A4988
693 //#define E5_DRIVER_TYPE A4988
694 //#define E6_DRIVER_TYPE A4988
695 //#define E7_DRIVER_TYPE A4988
696
697 // Enable this feature if all enabled endstop pins are interrupt-capable.

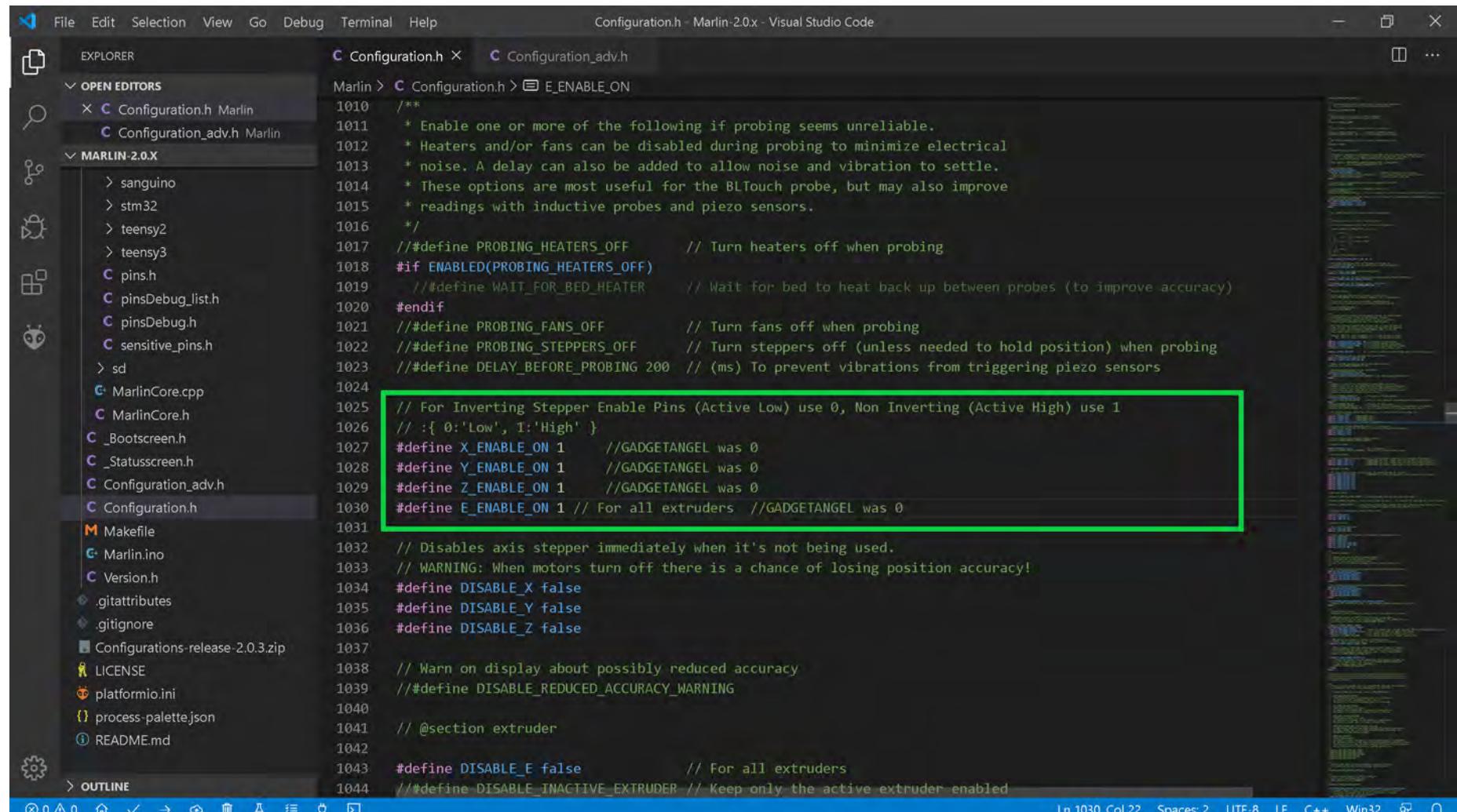
```

A green rectangular box highlights the lines from 680 to 696, indicating the area where comments are being removed or added.

- 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
OPEN EDITORS Marlin > Configuration.h > E_ENABLE_ON
  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
Ln 1030, Col 22 Spaces: 2 UTF-8 LF C++ Win32
  
```

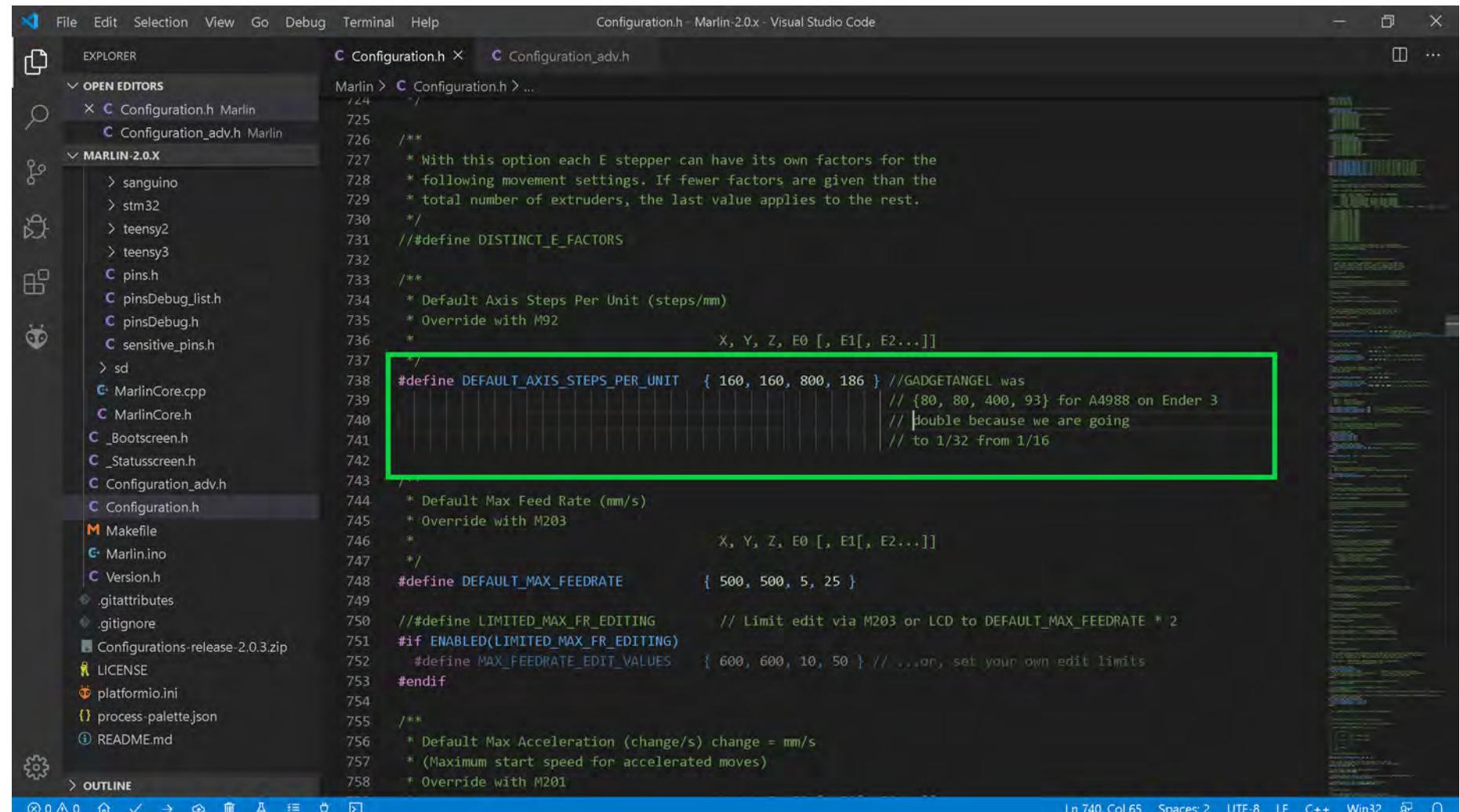
```

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 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:

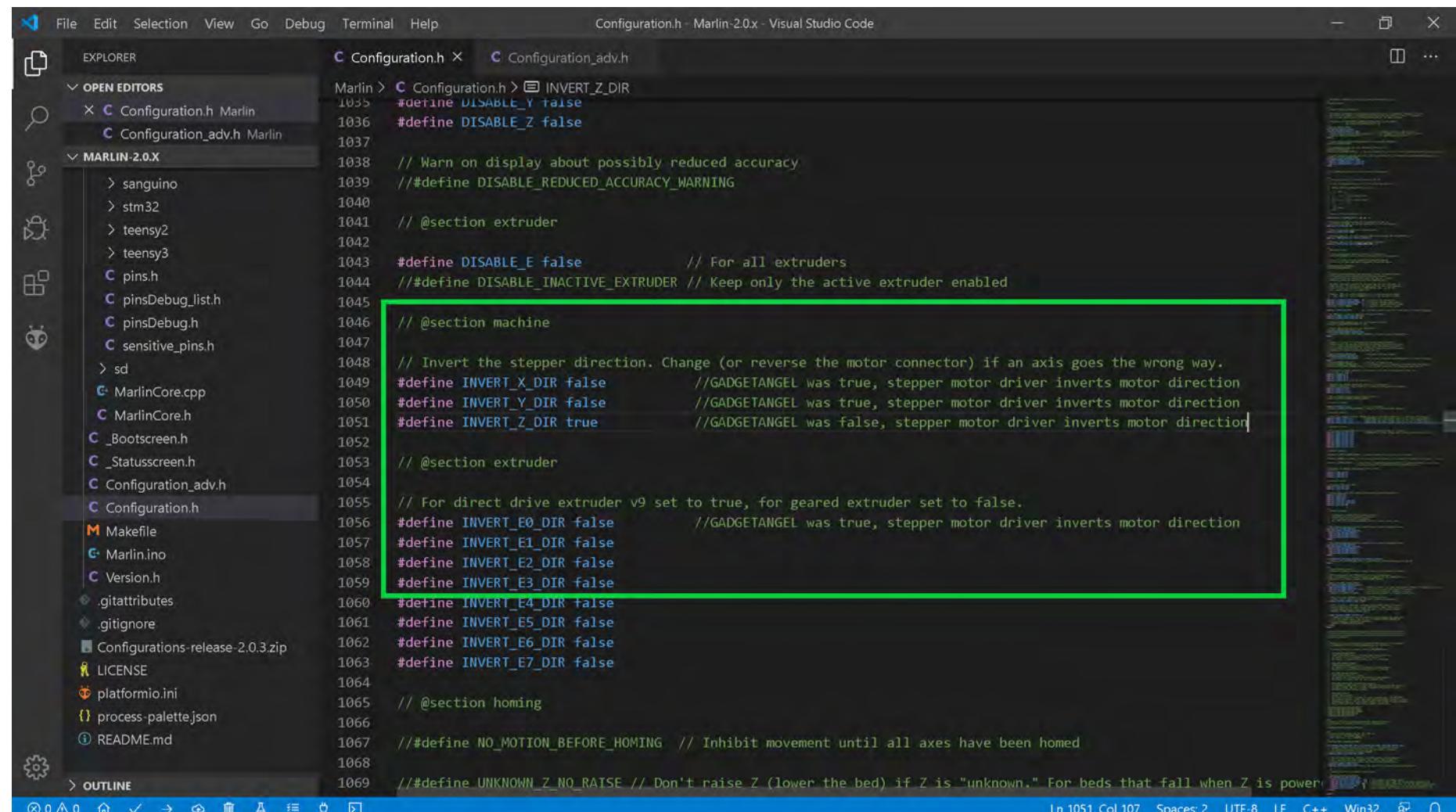
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

The 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 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
  1051 // @section extruder
  1052
  1053 // For direct drive extruder v9 set to true, for geared extruder set to false.
  1054
  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
  1064 // @section homing
  1065
  1066 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1067
  1068 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  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.

The screenshot shows the Visual Studio Code interface with the Marlin repository open. The left sidebar shows the file structure under 'EXPLORER' and 'MARLIN-2.0.X'. The main editor area displays Configuration.h with code related to endstop inversion and stepper driver timing. The bottom right shows the 'TERMINAL' tab with the following output:

```

STM26403VE_L1_1.1.1
BIGTREE_SKR_PRO          IGNORED
BIGTREE_CTP_11_0          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 =====

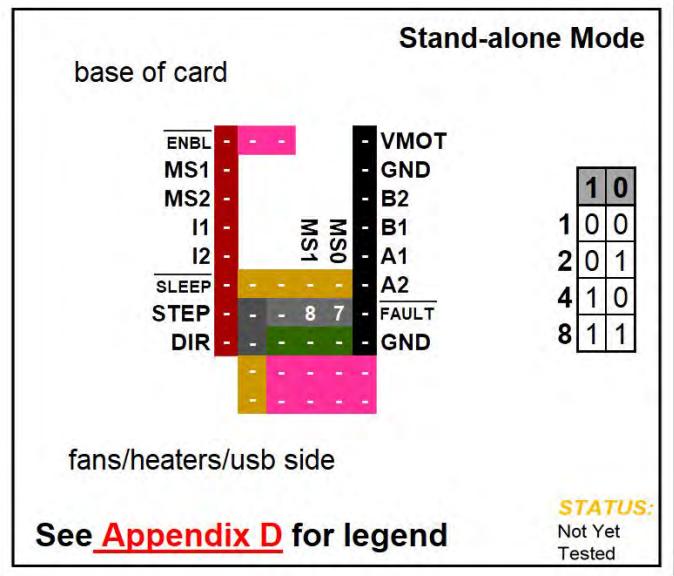
```

The terminal output is highlighted with a green box, and the checkmark icon in the bottom-left corner of the terminal tab is highlighted with a yellow 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 [Appendix E](#).

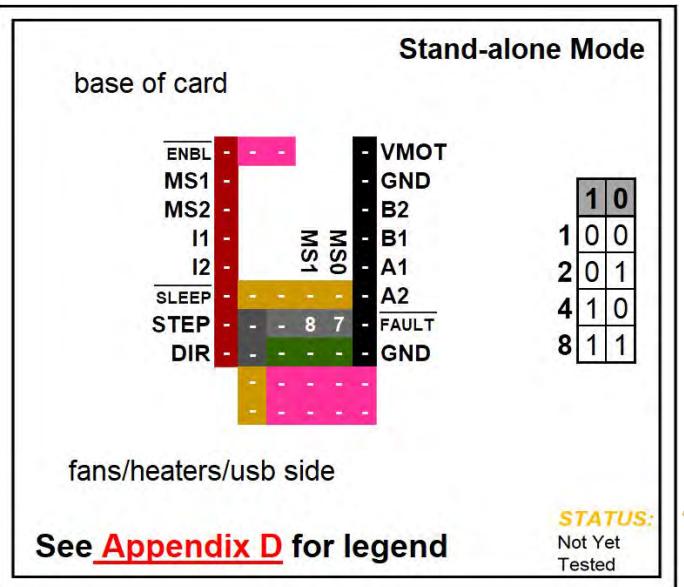
POLOLU MP6500

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.

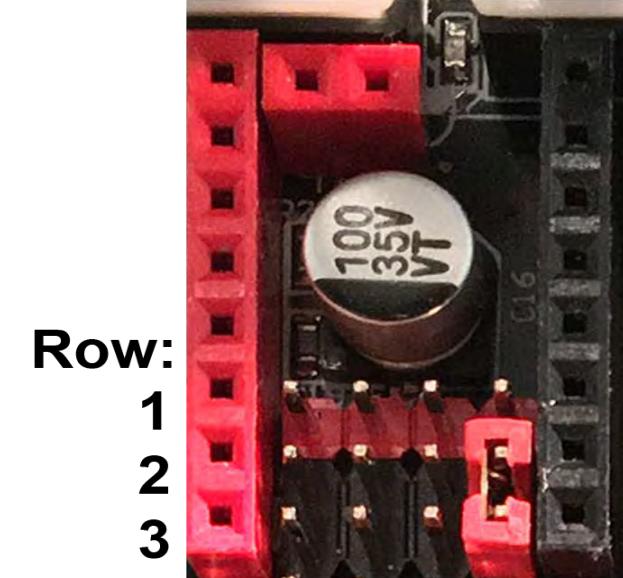
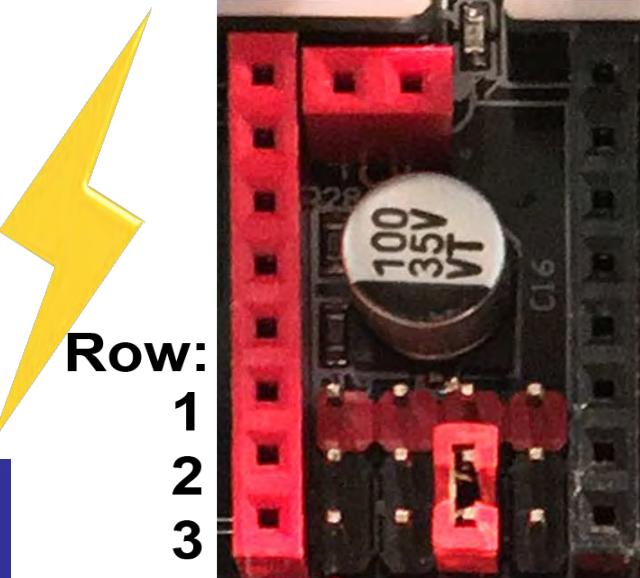


Driver Chip	MS1	MS0	Microstep Resolution	Excitation Mode
Pololu MP6500 Maximum 8 Subdivision 35V DC 2.5A (peak)	Low	Low	Full step	2 Phase
	Low	High	Half (1/2) step	1-2 Phase
	High	Low	Quarter (1/4) step	W1-2 Phase
	High	High	Eighth (1/8) step	2W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} =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

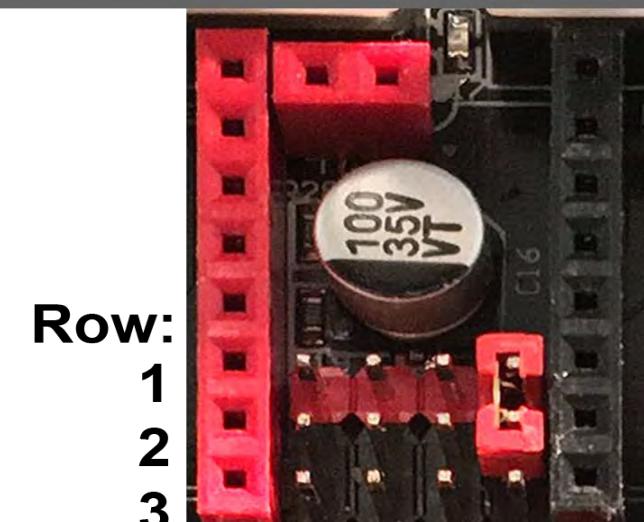
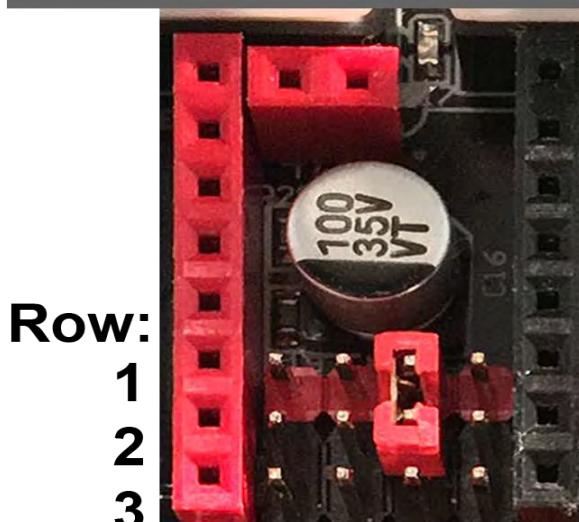
- See next page for the legend that belongs to the above chart.

**POLOLU MP6500****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

Low ➤ set Jumper between rows 2 and 3

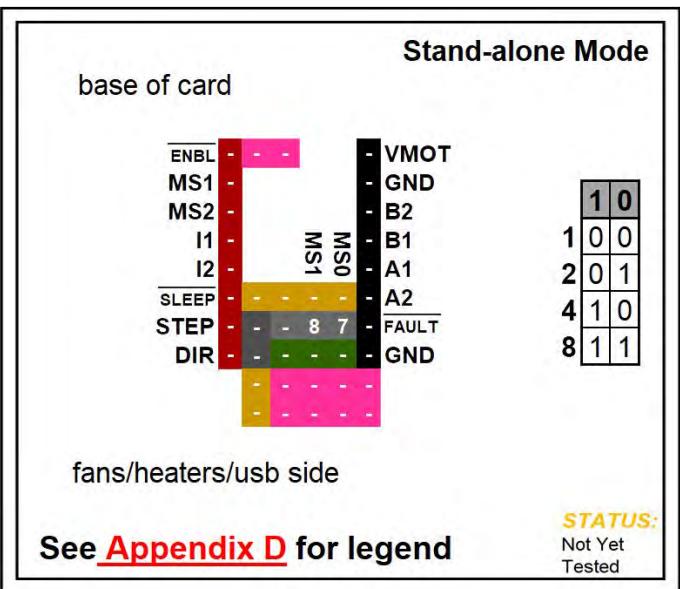


High ➤ set Jumper between rows 1 and 2



This is a
Jumper:



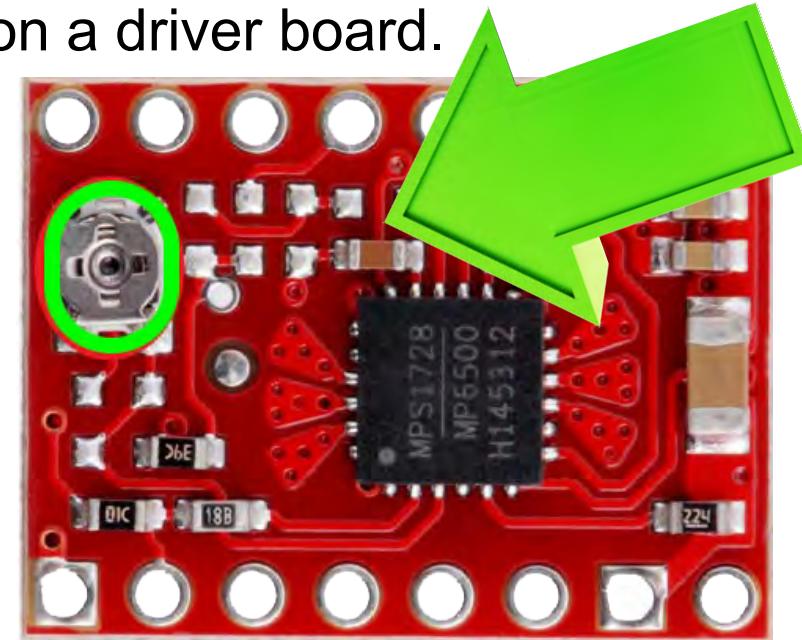
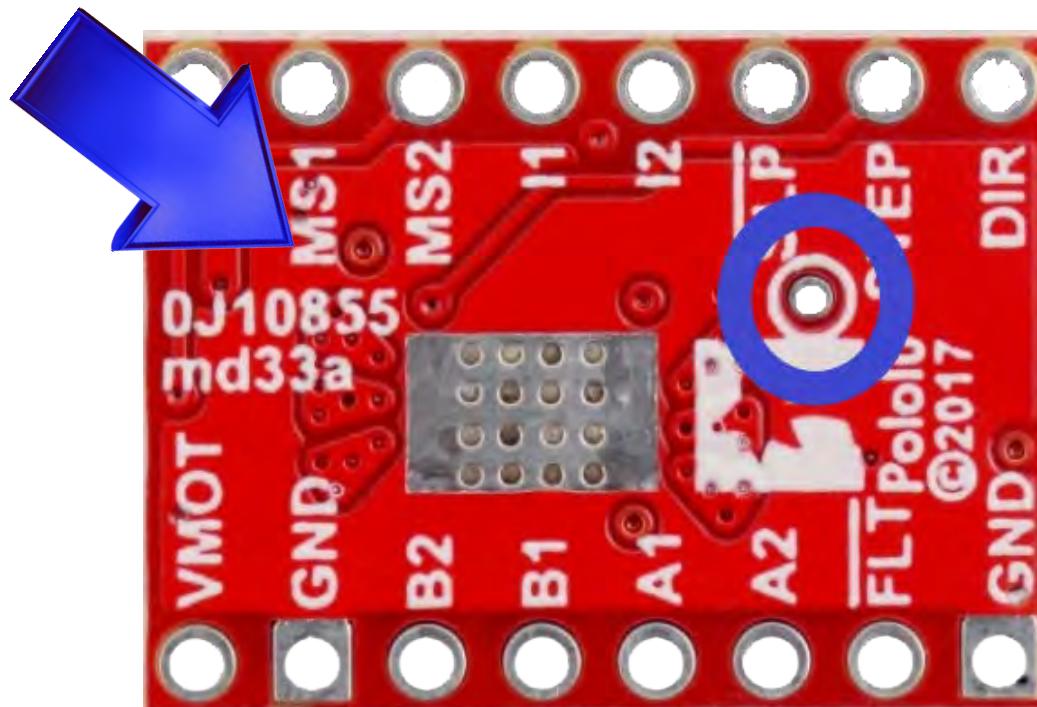


POLOLU MP6500

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



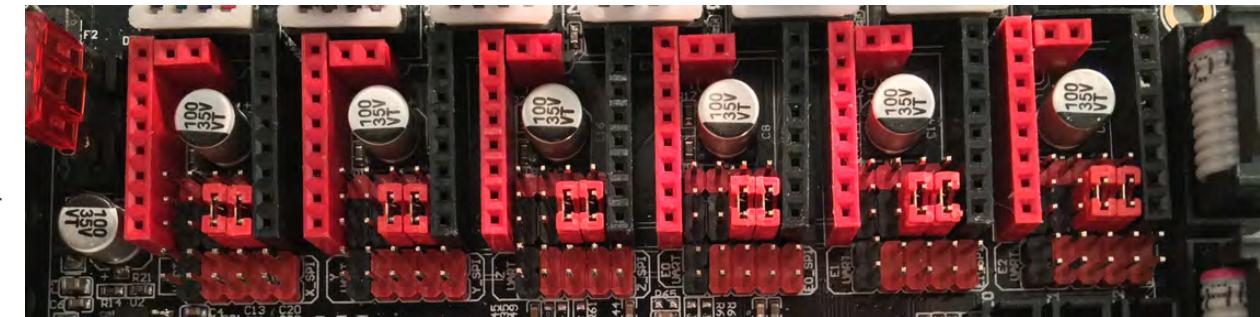
Note: MP6500 driver board does not use external current sense resistors (R_s).

Stand-alone Mode

POLOLU MP6500

STEP

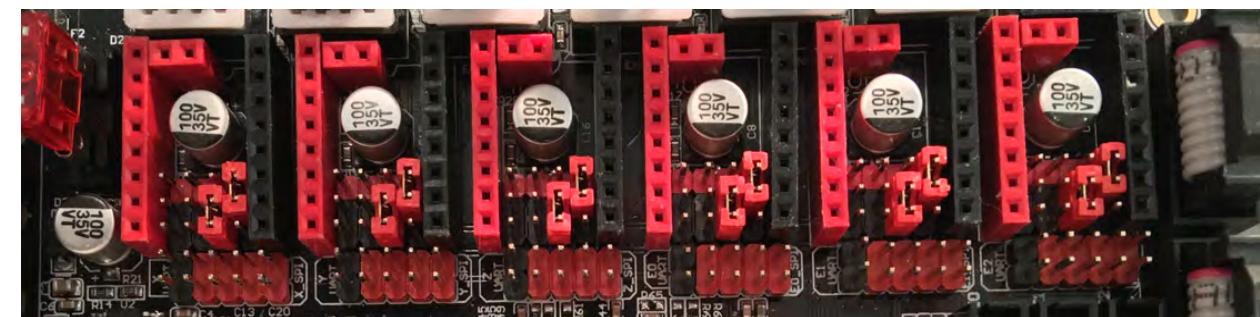
ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	MS1	MS0	B1
I2	-	MS0	-	A1
SLEEP	-	-	-	A2
STEP	-	-	8 7	FAULT
DIR	-	8 7	-	GND
	-	-	-	-



See [Appendix D](#) for legend

1 / 2

ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	MS1	MS0	B1
I2	-	MS0	-	A1
SLEEP	-	-	7	A2
STEP	-	-	8 7	FAULT
DIR	-	8	-	GND
	-	-	-	-



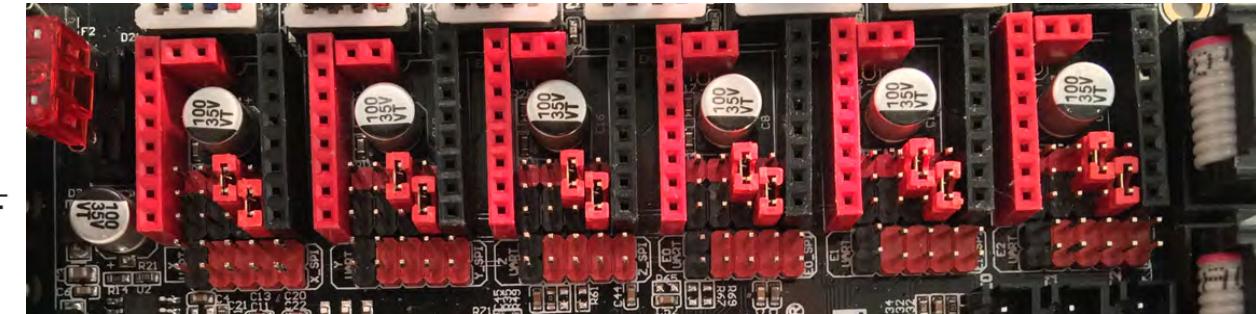
See [Appendix D](#) for legend

Stand-alone Mode

POLOLU MP6500

1 / 4

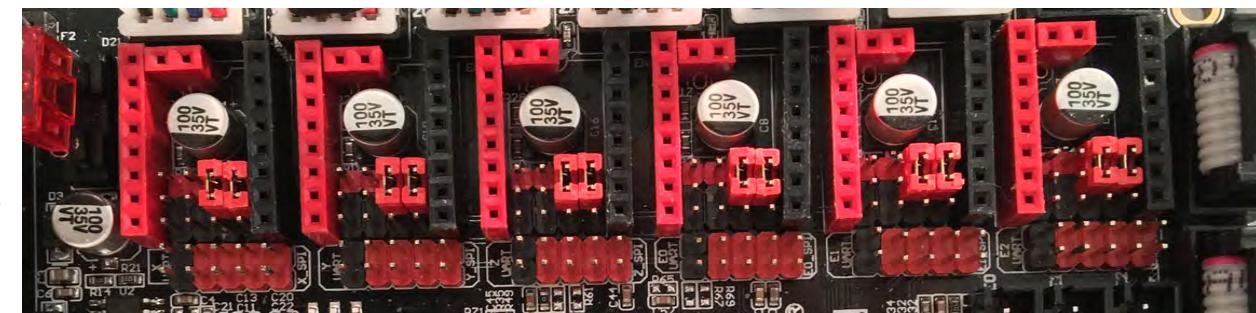
ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	-	MS1	B1
I2	-	-	MS0	A1
SLEEP	-	8	-	A2
STEP	-	8	7	FAULT
DIR	-	7	-	GND



See [Appendix D](#) for legend

1 / 8

ENBL	-	-	-	VMOT
MS1	-	-	-	GND
MS2	-	-	-	B2
I1	-	-	MS1	B1
I2	-	-	MS0	A1
SLEEP	-	8	7	A2
STEP	-	8	7	FAULT
DIR	-	-	-	GND

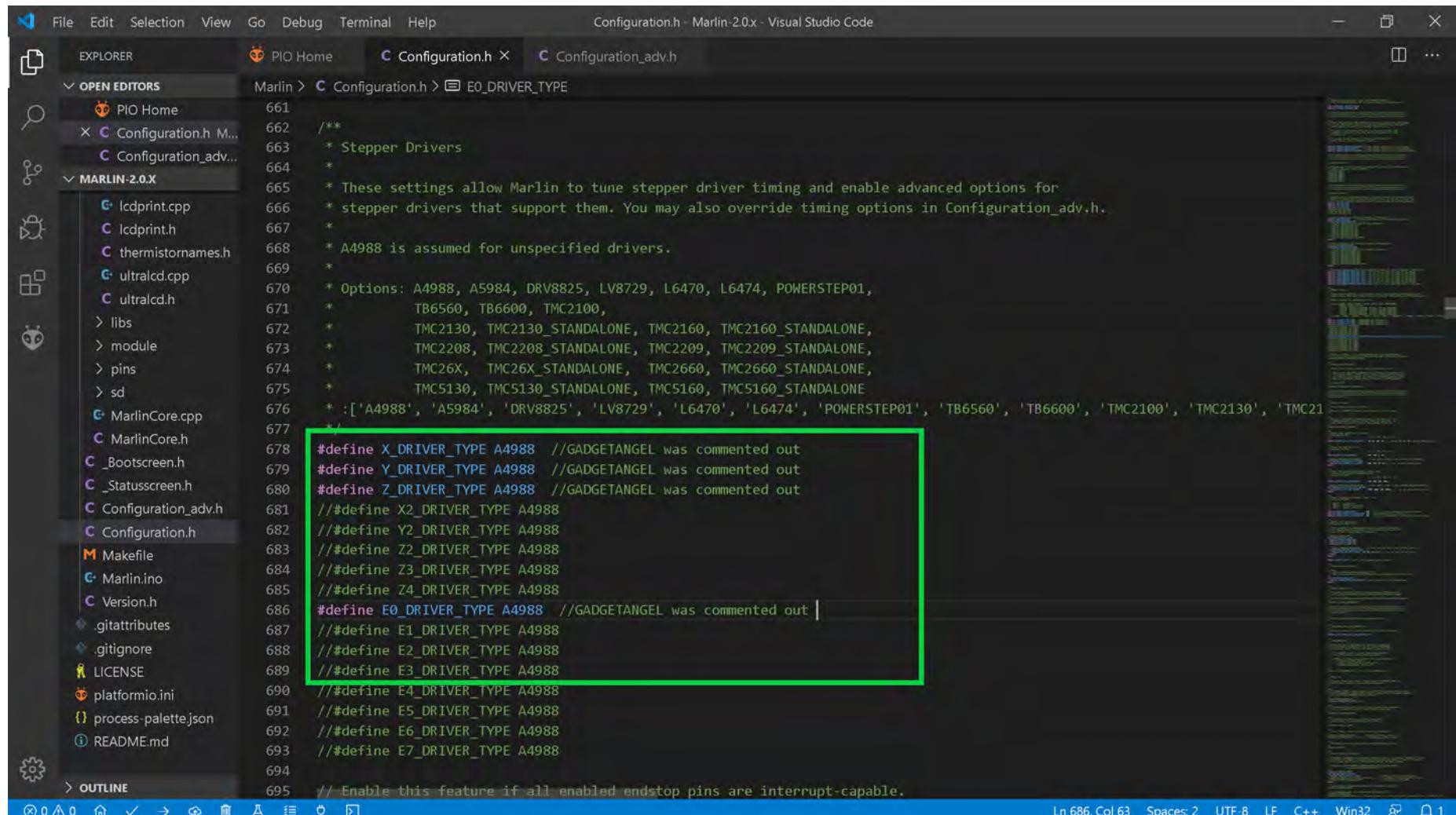


See [Appendix D](#) for legend

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.**



```

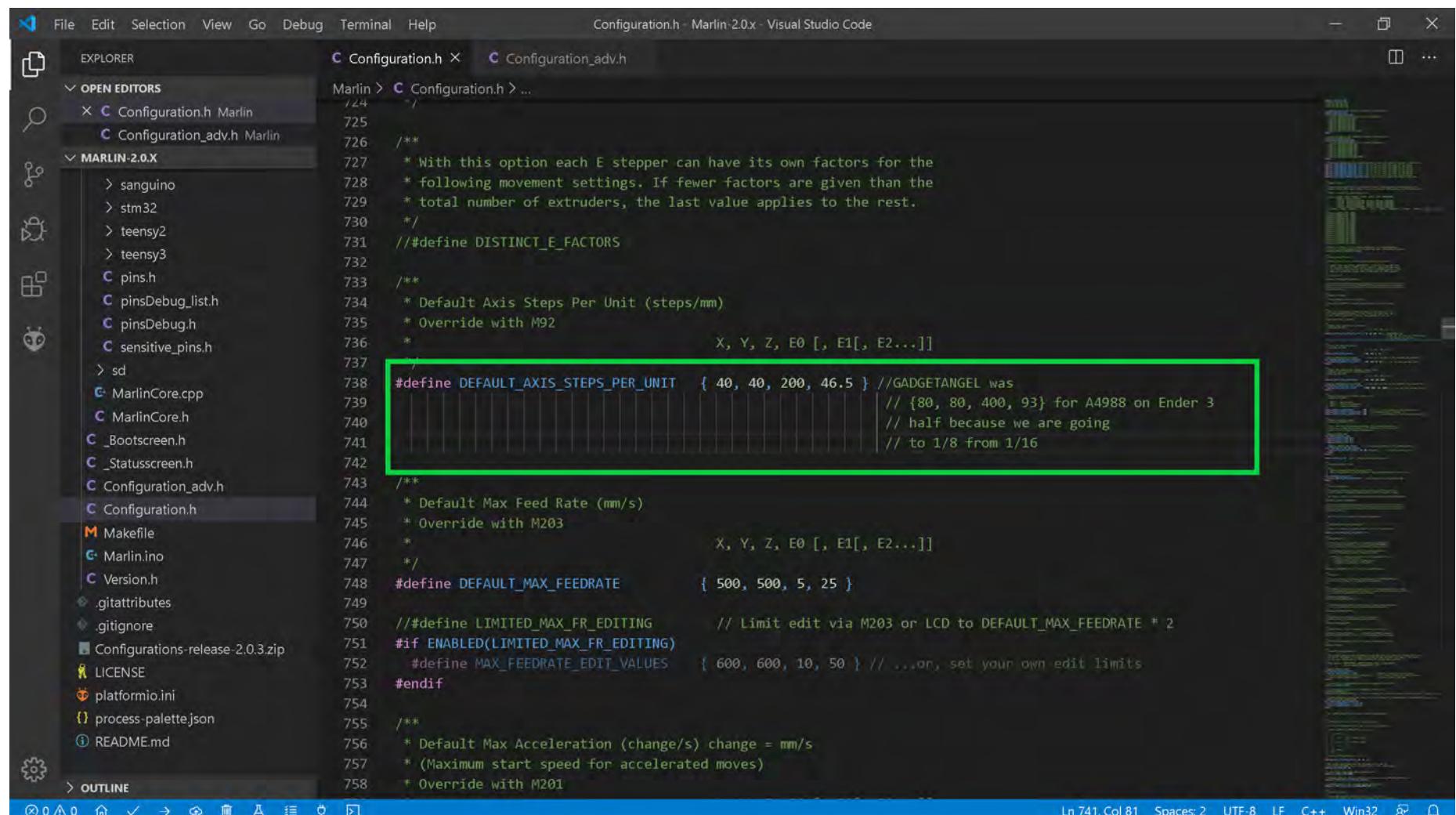
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable 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 //##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.
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:

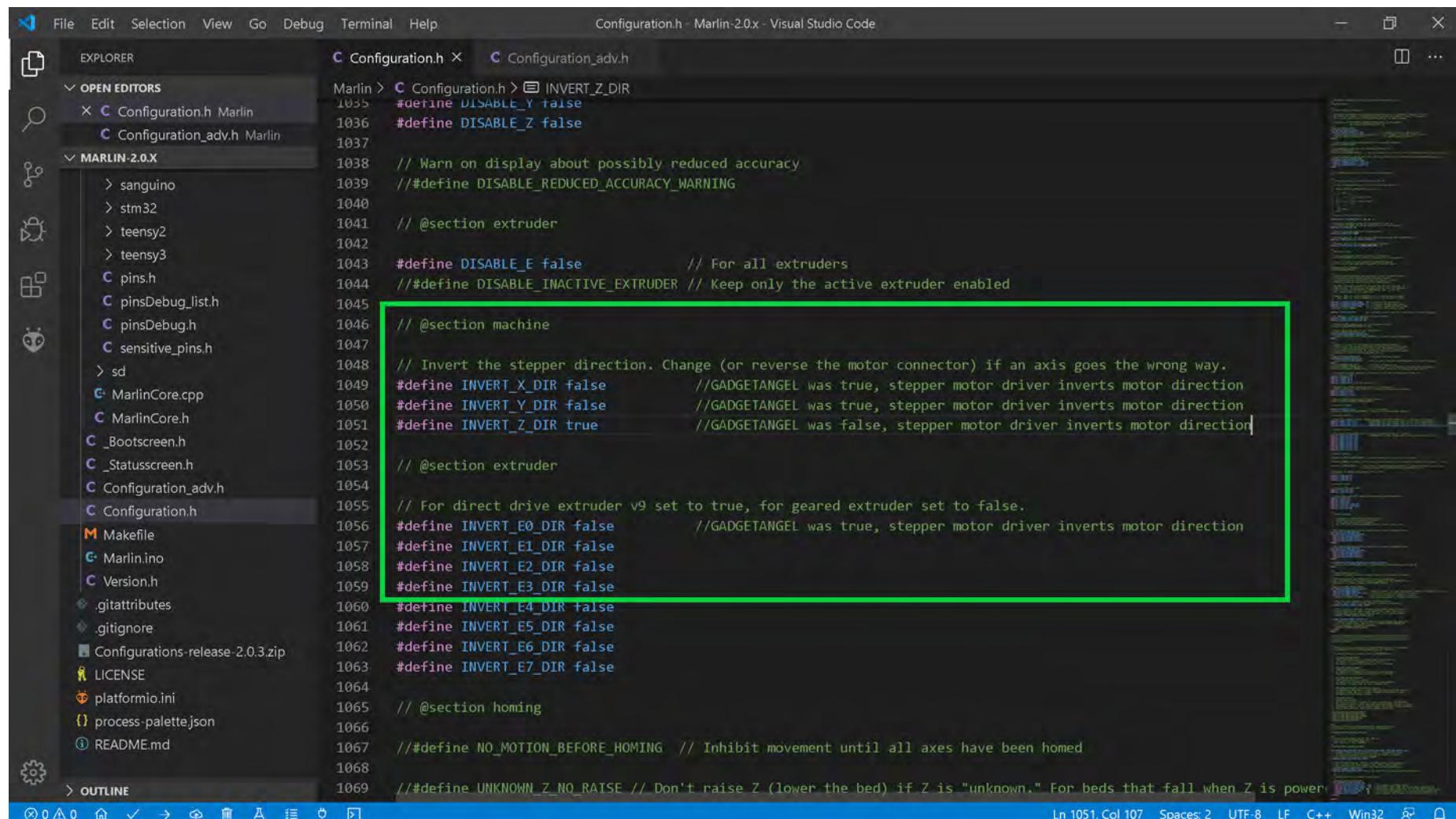
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 40, 40, 200, 46.5 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// half because we are going
// to 1/8 from 1/16
```

The 'EXPLORER' sidebar on the left shows the project structure, including files like 'pins.h', 'pinsDebug.list.h', and 'MarlinCore.cpp'. The status bar at the bottom indicates 'Ln 741, Col 81'.

- 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



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code defines several macro constants for inverting stepper motor directions. A green rectangular box highlights the following section of code:

```

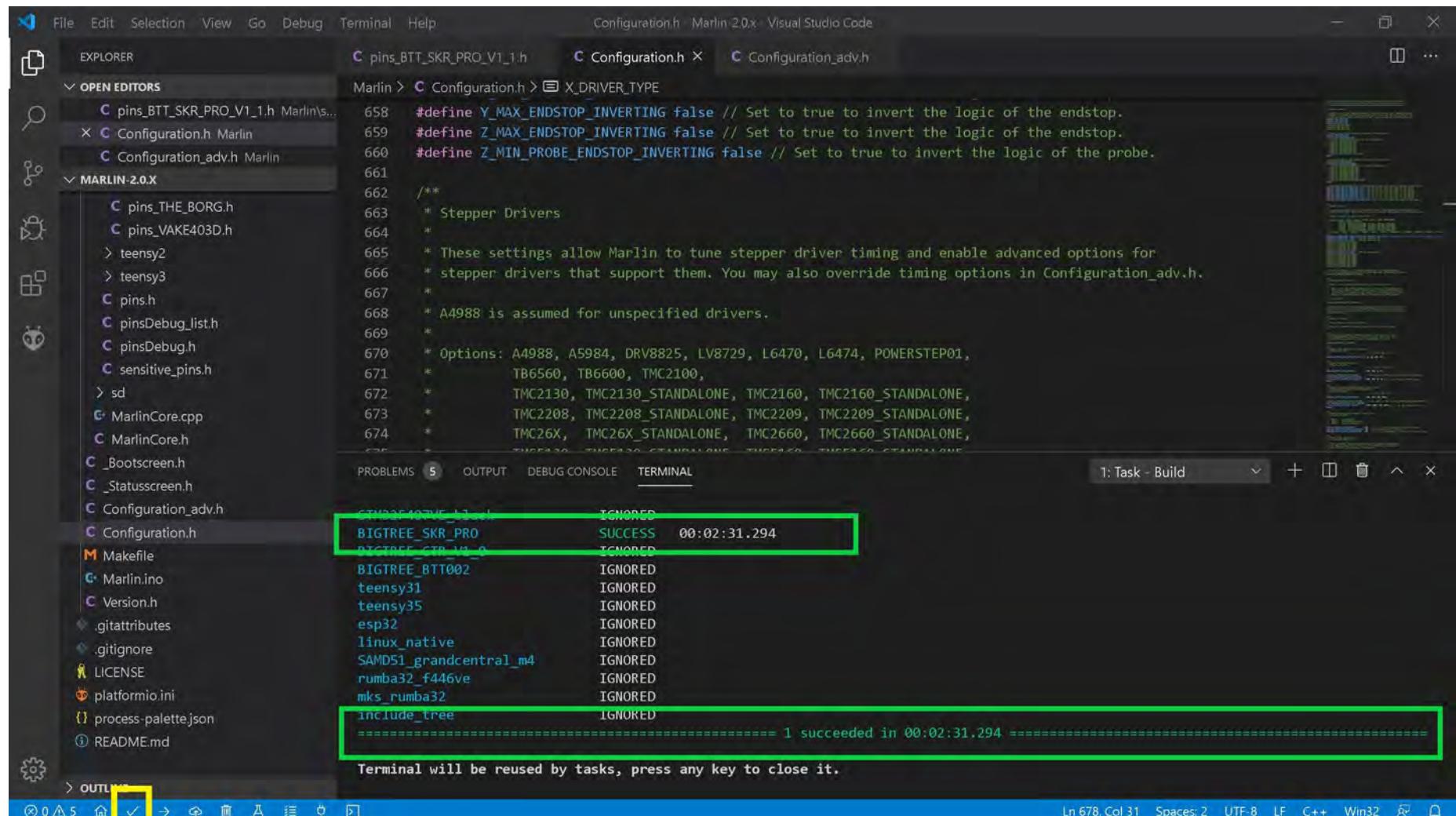
1049 #define INVERT_X_DIR false           //GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false           //GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true            //GADGETANGEL was false, stepper motor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

```

- Go to the next page.

The (latest release of) Marlin Setup for 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.

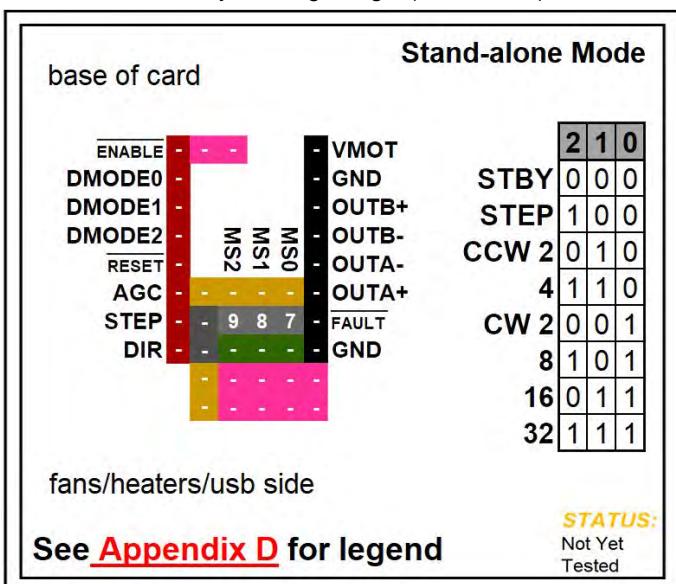


```

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 /**
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
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + □ ^ ×
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_1 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.
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 [Appendix E](#).

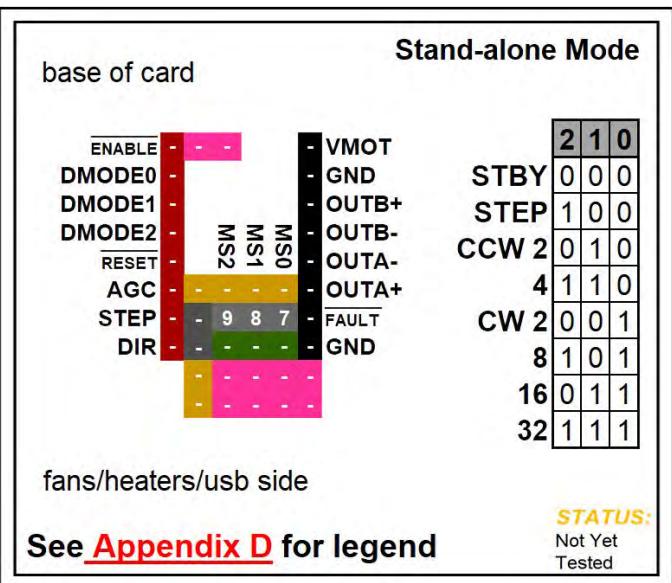


POLOLU TB67S249FTG

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 Maximum 32 Subdivision 47V DC 4.5A (peak)	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	$I_{MAX} = V_{ref} * 1.25$			$V_{ref} = \frac{I_{MAX}}{1.25}$

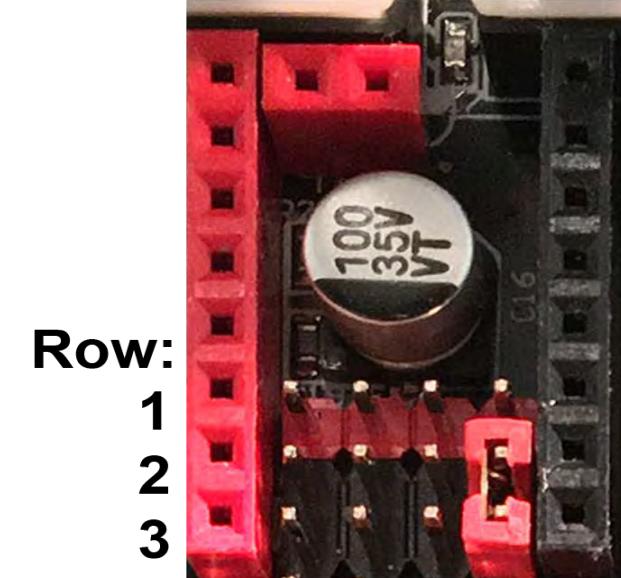
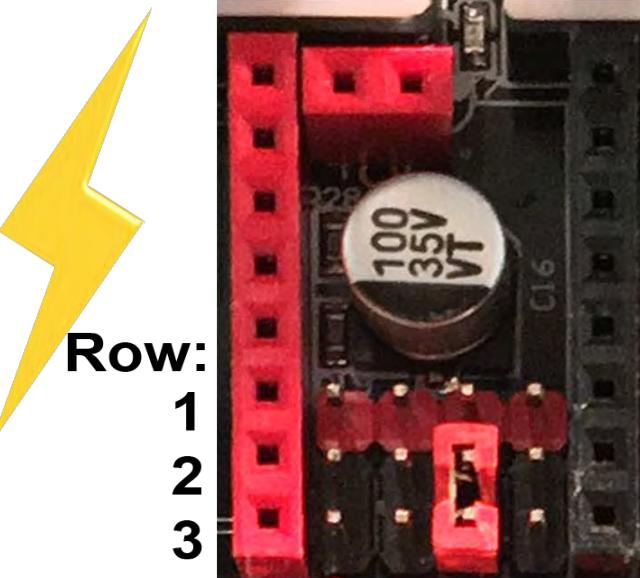
- See next page for the legend that belongs to the above chart.



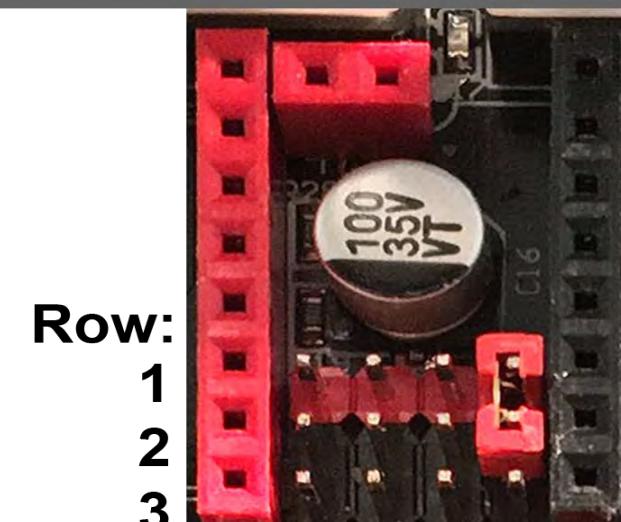
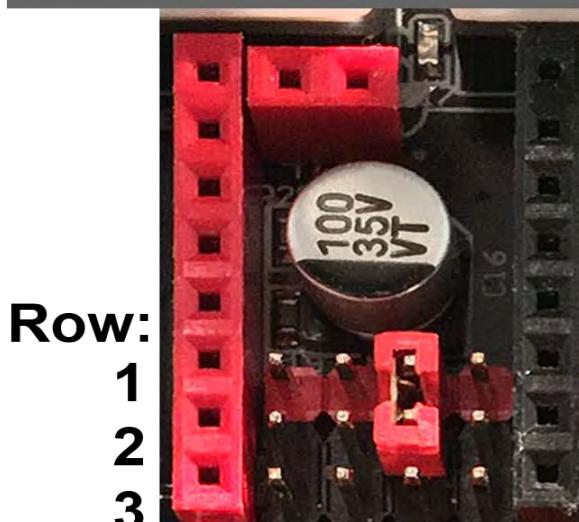
POLOLU TB67S249FTG

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows
2 and 3

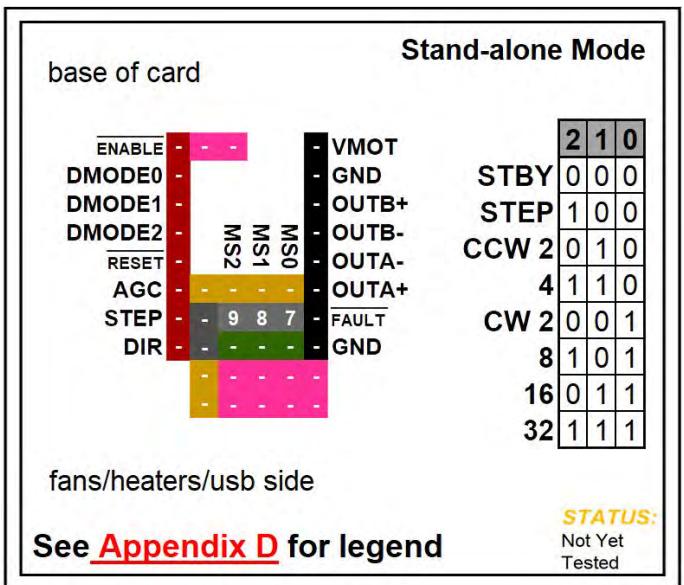


High ➡ set Jumper between rows
1 and 2



This is a
Jumper:



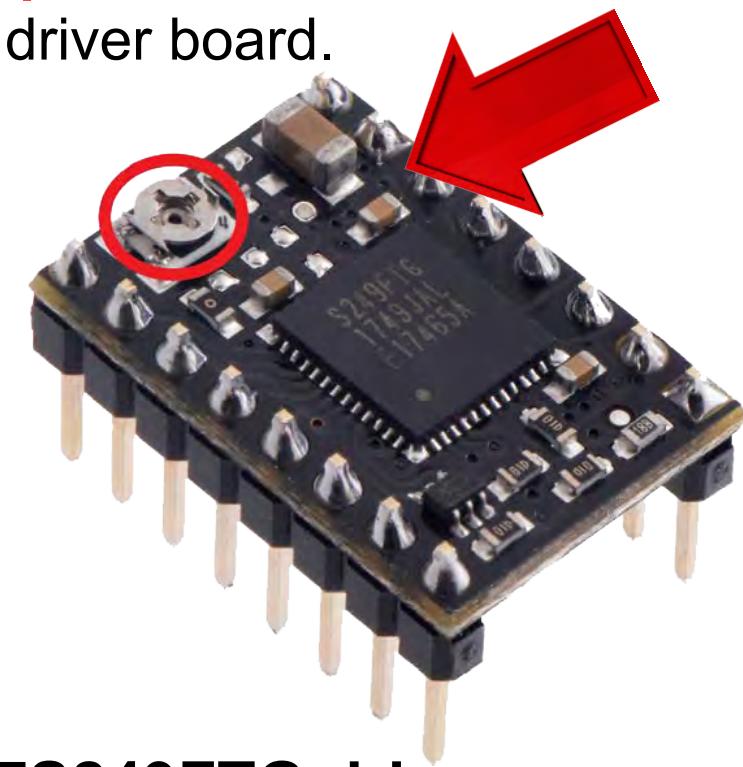
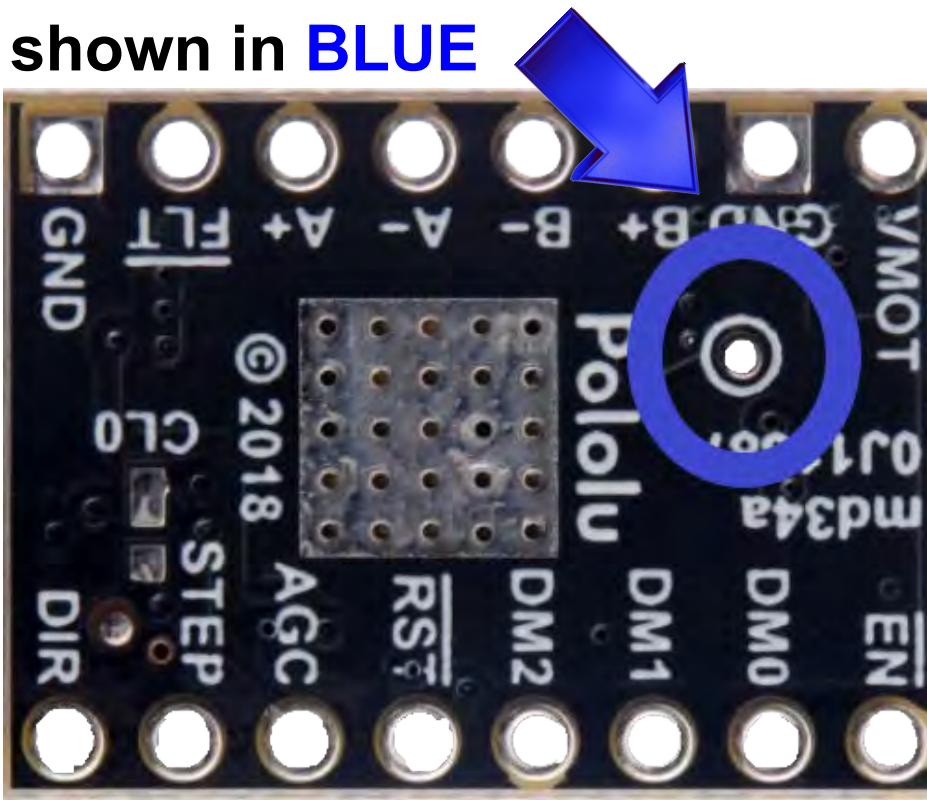


POLOLU TB67S249FTG

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**



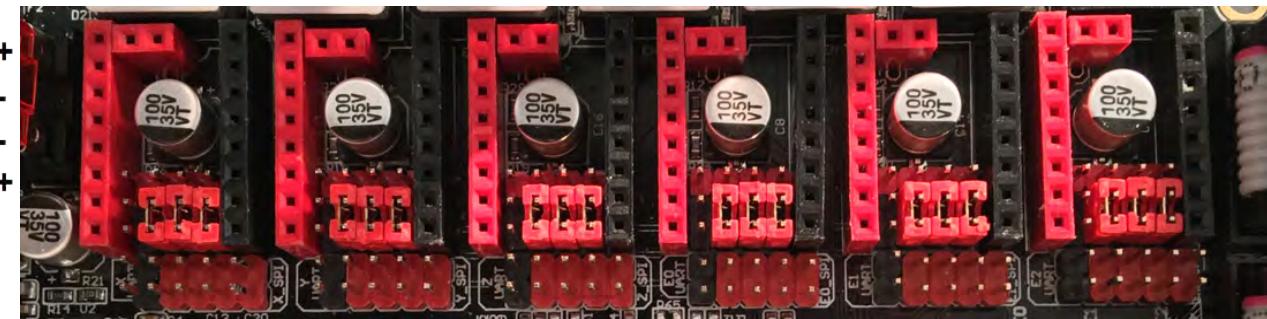
Note: TB67S249FTG driver board does not use external current sense resistors (R_s).

Stand-alone Mode

POLOLU TB67S249FTG

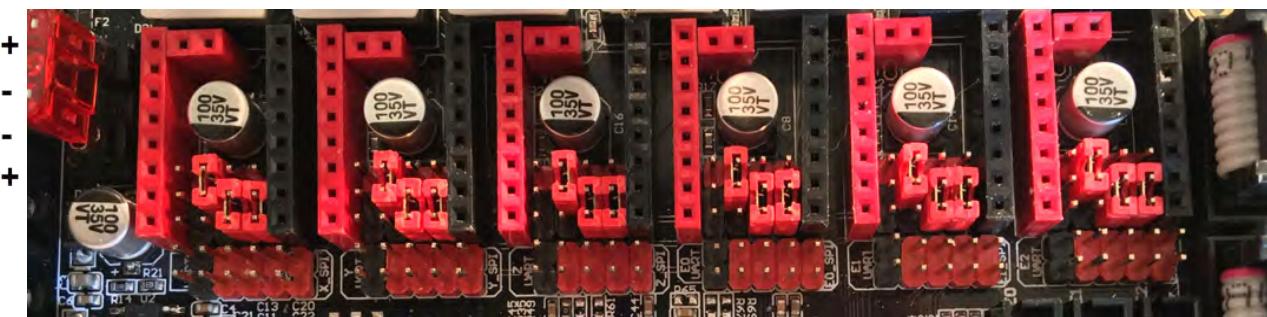
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

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



See [Appendix D](#) for legend

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



See [Appendix D](#) for legend

**STAND
BY**

STEP

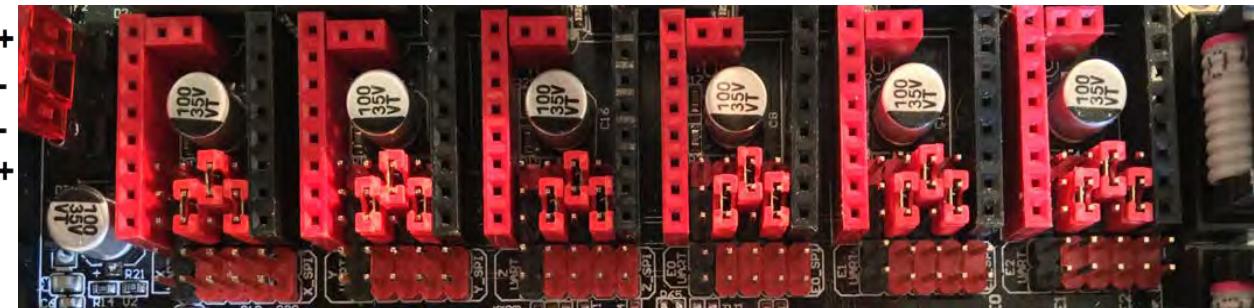
Stand-alone Mode

CCW
1 / 2

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

POLOLU TB67S249FTG

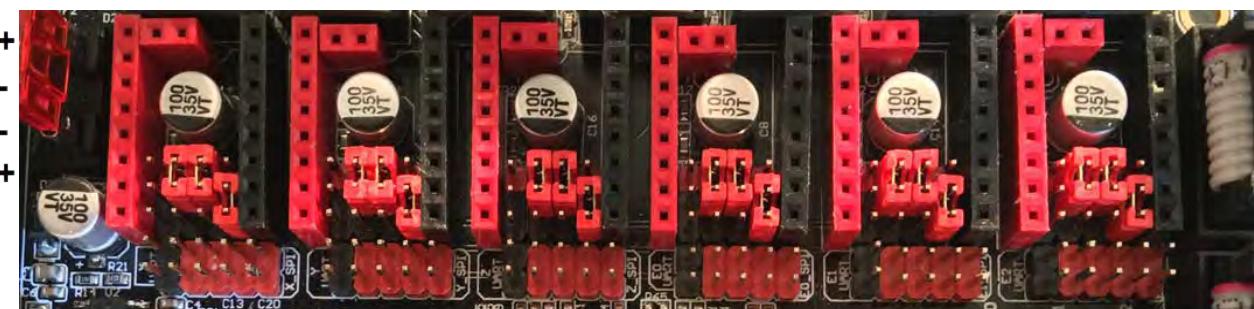
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".



See [Appendix D](#) for legend

1 / 4

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



See [Appendix D](#) for legend

CW
1 / 2

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



See [Appendix D](#) for legend

Stand-alone Mode

POLOLU TB67S249FTG

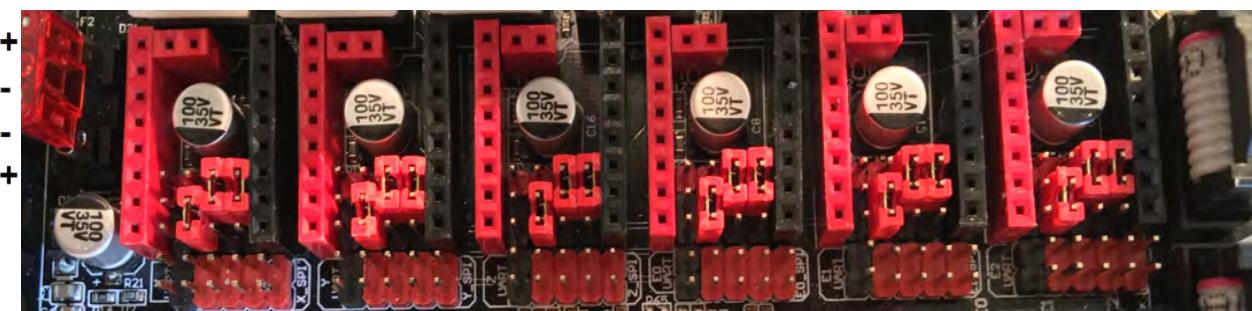
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

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



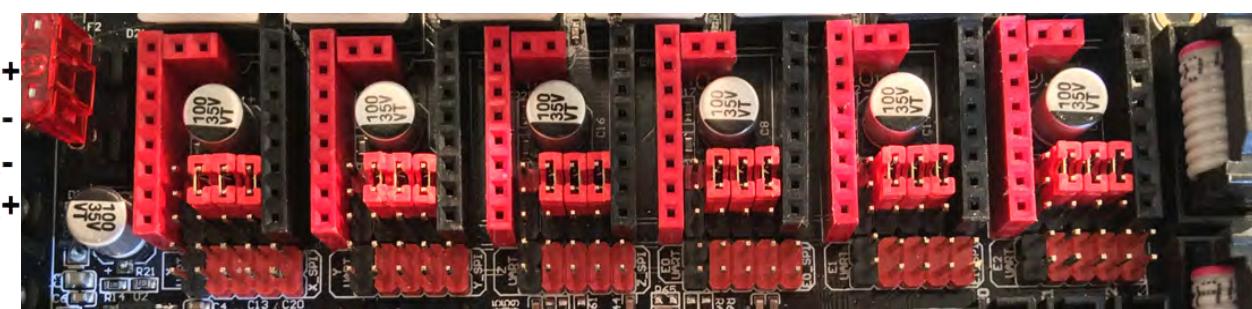
See [Appendix D](#) for legend

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



See [Appendix D](#) for legend

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

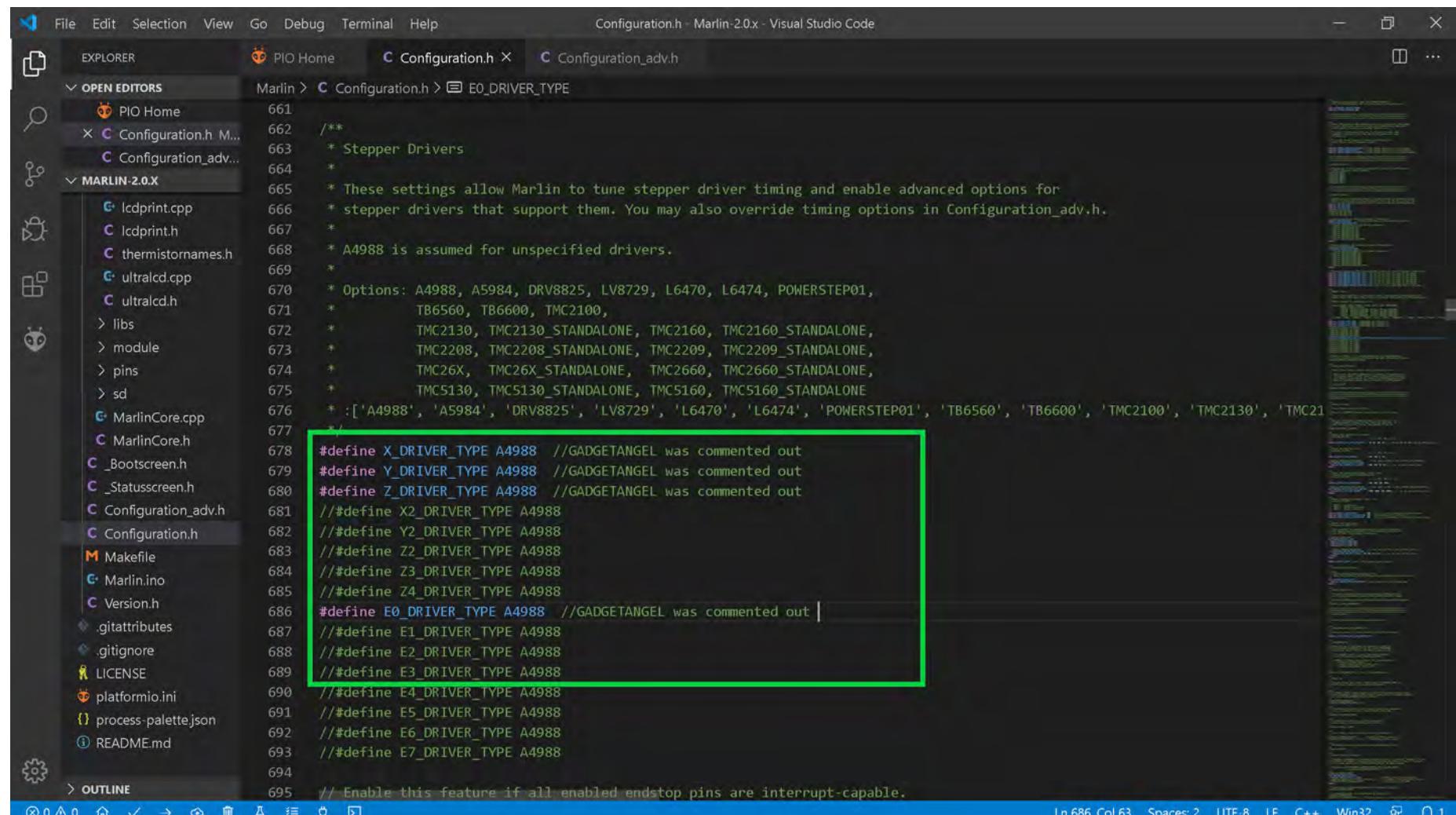


See [Appendix D](#) for legend

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 file `Configuration.h` open. The code editor displays the following snippet, with a green box highlighting the driver type definitions:

```

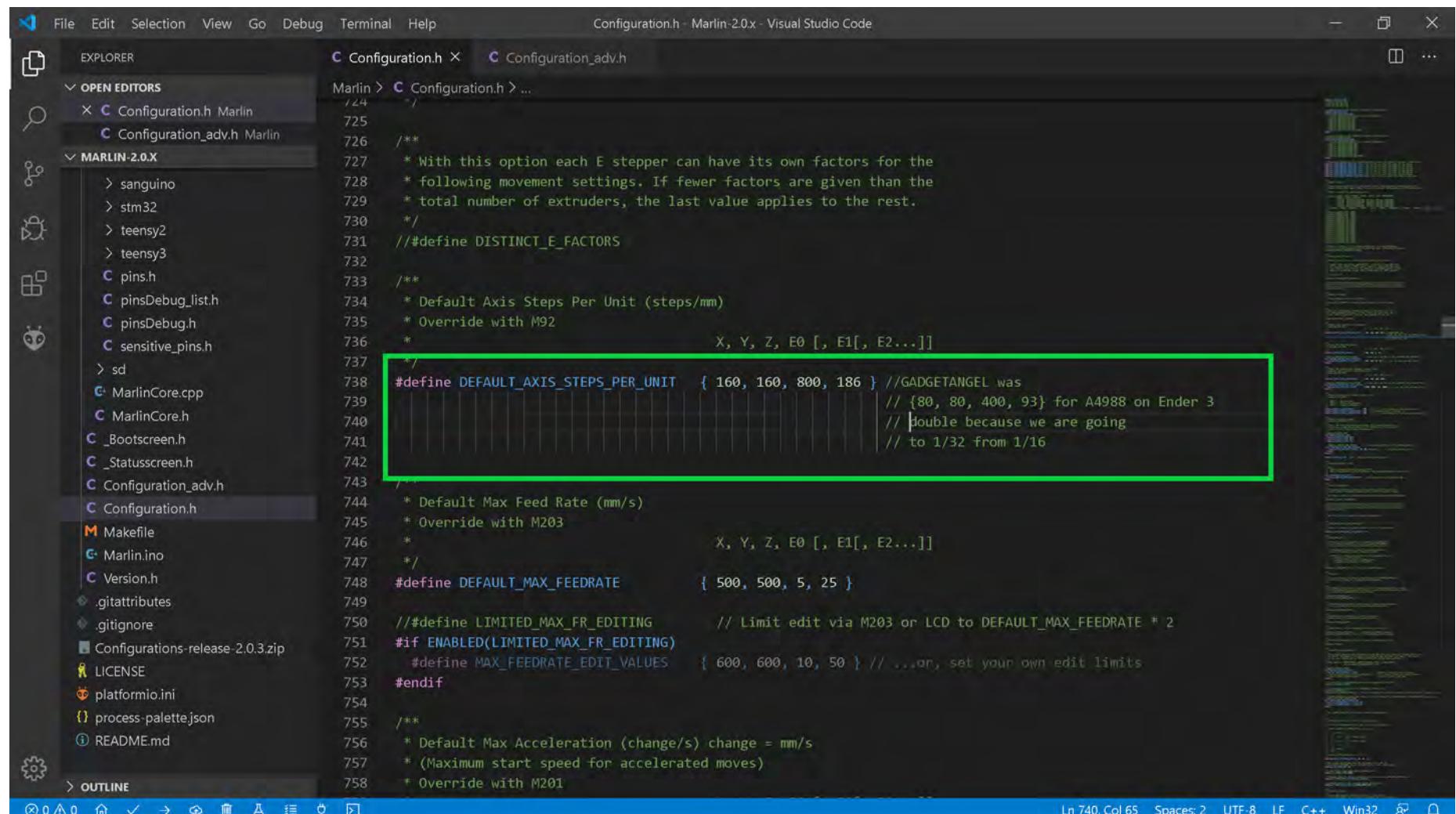
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: 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', 'TMC2
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.

```

- 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-2.0.X

```

725 /**
726 * With this option each E stepper can have its own factors for the
727 * following movement settings. If fewer factors are given than the
728 * total number of extruders, the last value applies to the rest.
729 */
730 // #define DISTINCT_E_FACTORS
731 /**
732 * Default Axis Steps Per Unit (steps/mm)
733 * Override with M92
734 */
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 /**
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 } // ...on, 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 */
760 /**
761 * Override with M201
762 */

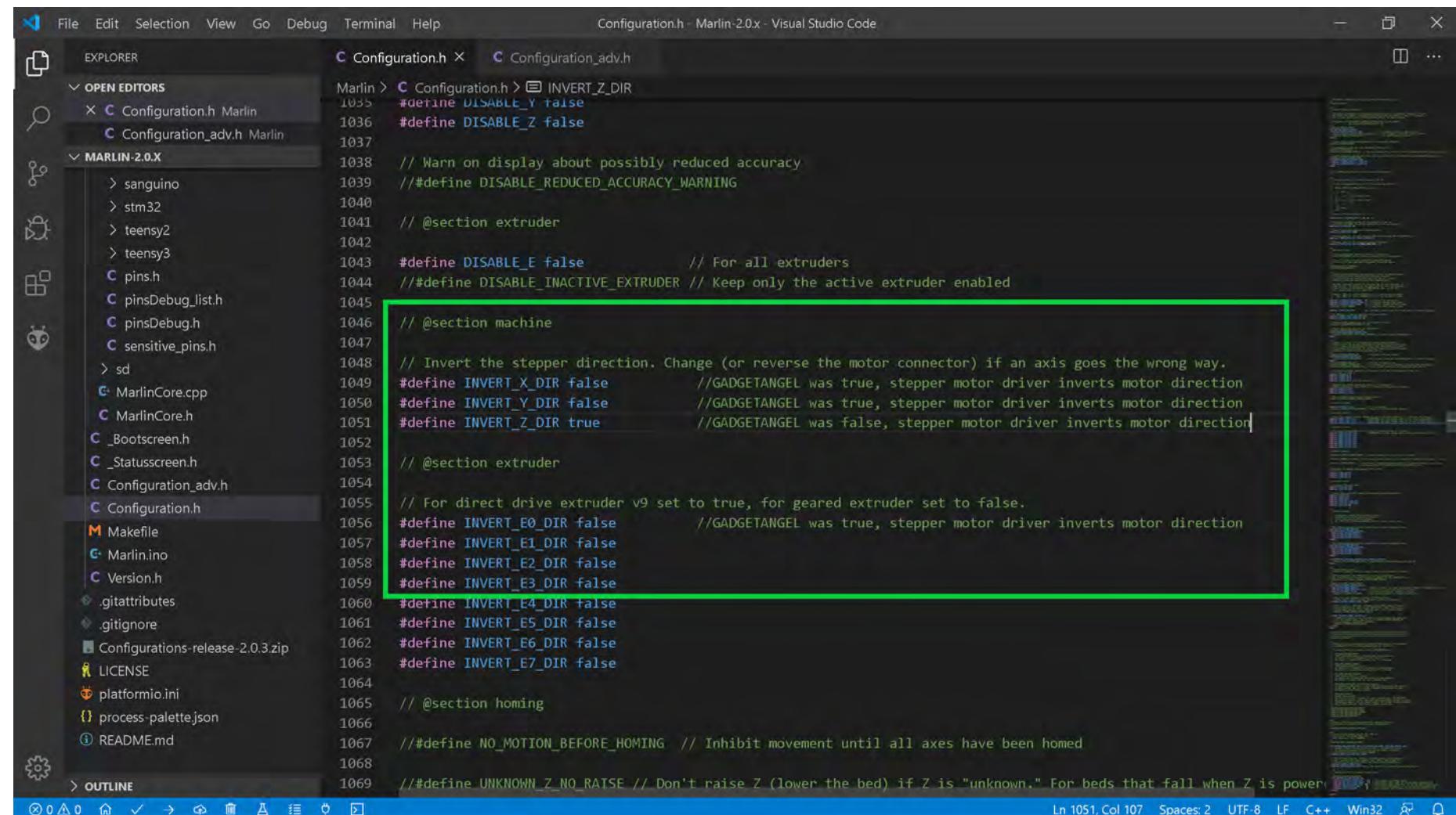
```

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



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 > Configuration.h
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 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 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 like `pins.h` and `MarlinCore.cpp`. The main editor area displays the `Configuration.h` file with code related to stepper driver timing and options for TB67S249FTG drivers. Below the editor is a terminal window showing the build process:

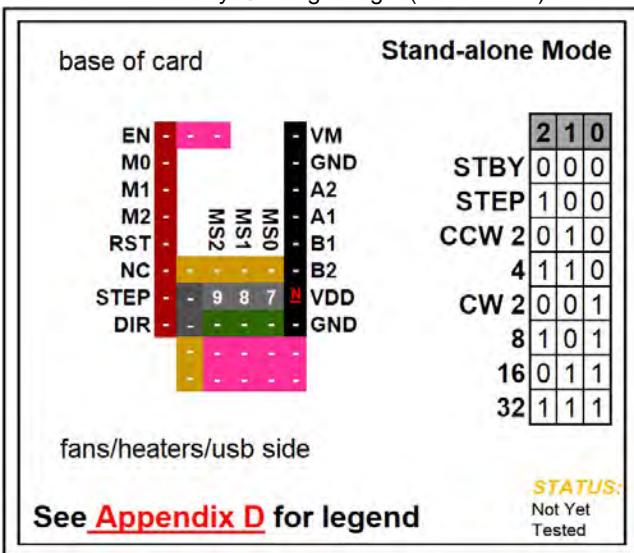
```

TMC26X
BIGTREE_SKR_PRO
BIGTREE_CTP_11_0
BIGTREE_BT1002
teensy31
teensy35
esp32
linux_native
SAMD51_grandcentral_m4
rumba32_f446ve
mks_rumba32
include_tree
=====
1 succeeded in 00:02:31.294

```

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

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



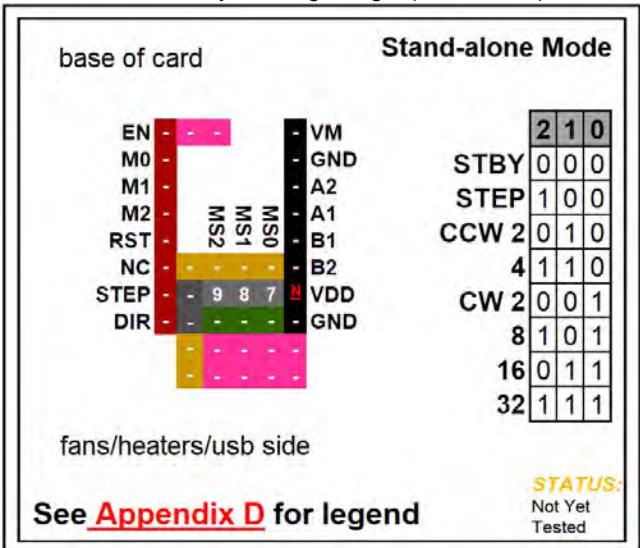
BIQU S109

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

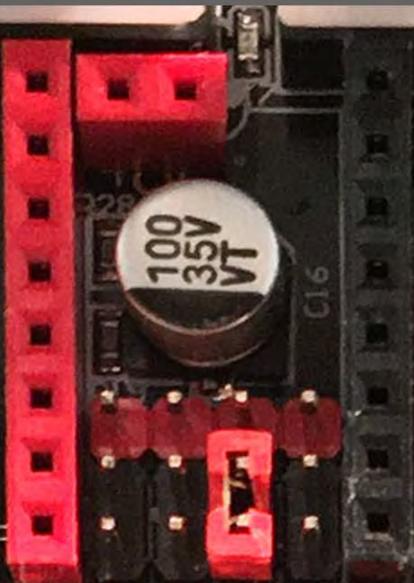
- See next page for the legend that belongs to the above chart.

**BIQU S109****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

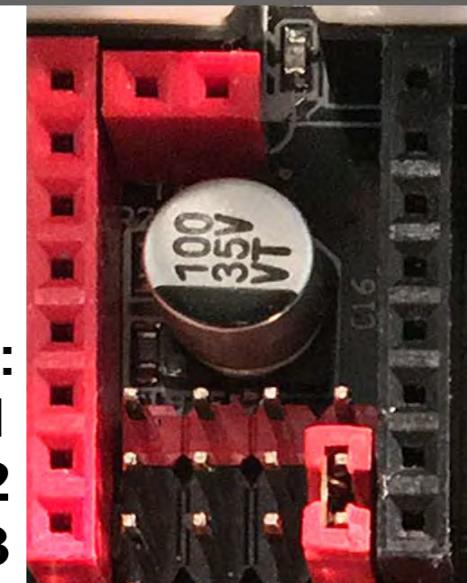
Low ➤ set Jumper between rows 2 and 3



Row:

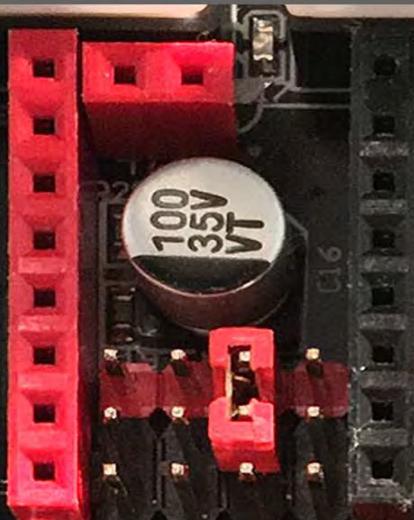
1
2
3

Row:

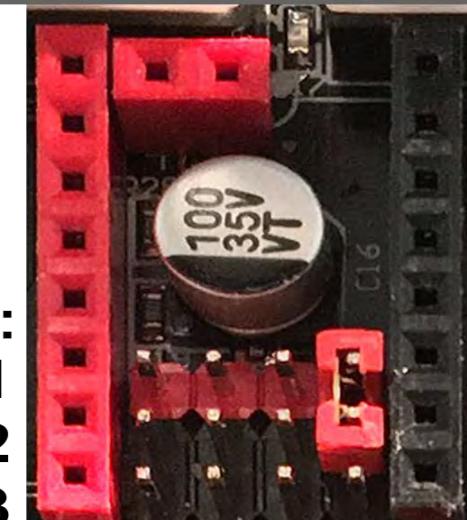
1
2
3

High ➤ set Jumper between rows 1 and 2

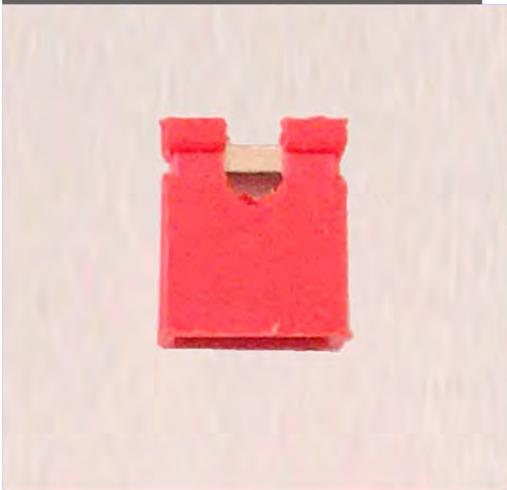
Row:

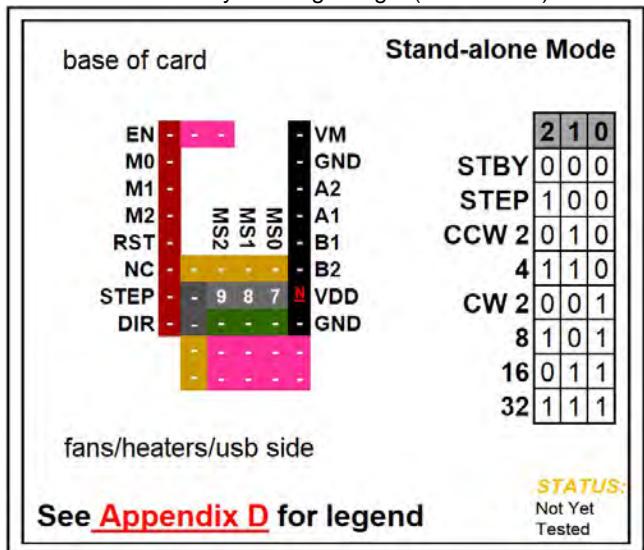
1
2
3

Row:

1
2
3

This is a
Jumper:





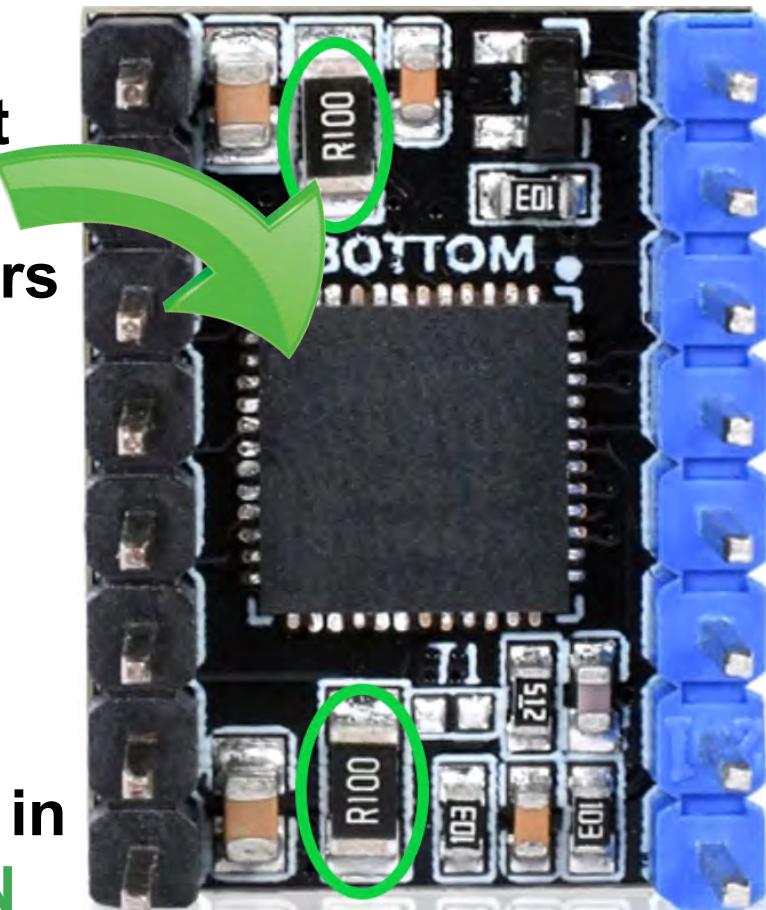
BIQU S109

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>

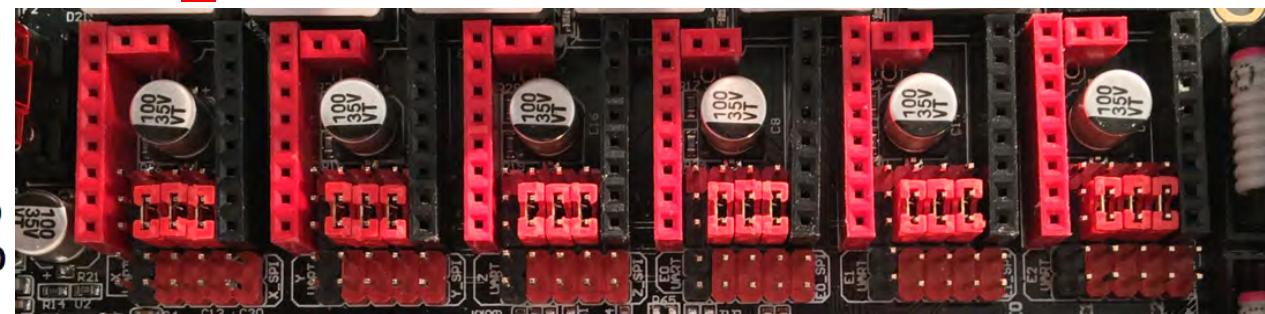
Stand-alone Mode

BQU S109

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

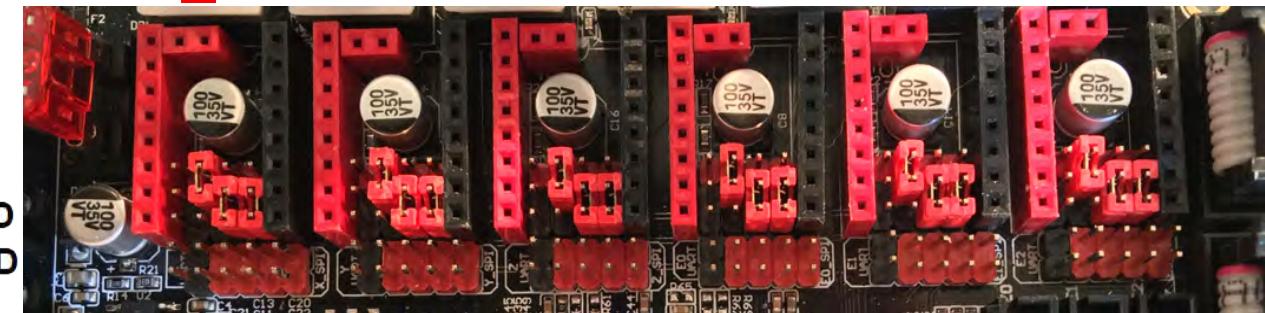
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



See [Appendix D](#) for legend

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

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



See [Appendix D](#) for legend

**STAND
BY**

STEP

Stand-alone Mode

CCW
1 / 2

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	-	-	A1
RST	-	MS2	MS1	B1
NC	-	8	-	B2
STEP	-	9	8	7
DIR	-	9	-	7
	-	-	-	VDD
	-	-	-	GND

B1QU S109

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".

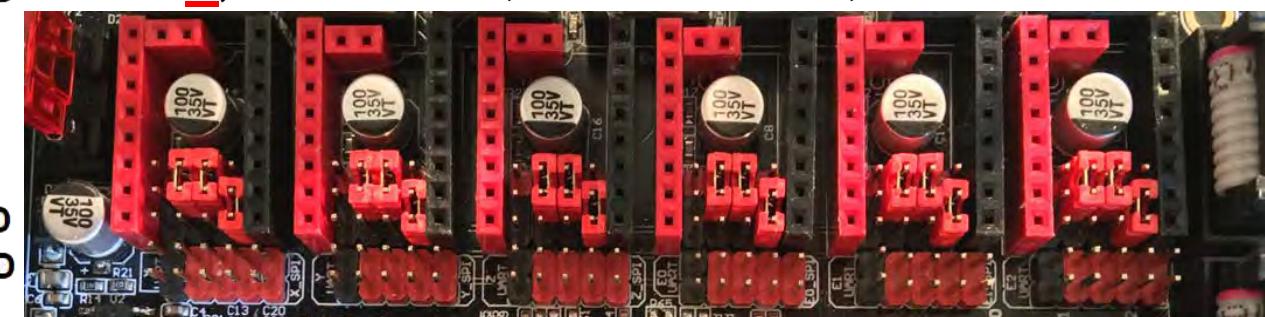


See [Appendix D](#) for legend

1 / 4

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	A1
RST	-	MS0	-	B1
NC	-	9	8	-
STEP	-	9	8	7
DIR	-	-	7	-
	-	-	-	VDD
	-	-	-	GND

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

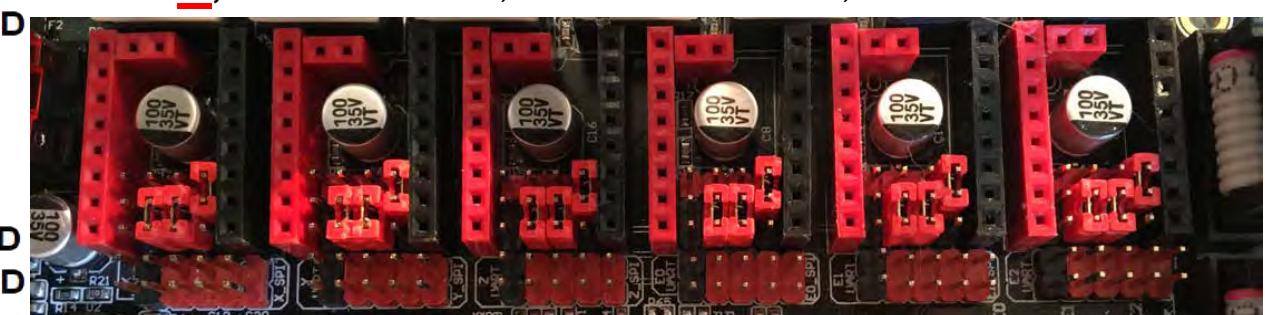


See [Appendix D](#) for legend

CW
1 / 2

EN	-	-	-	VM
M0	-	-	-	GND
M1	-	-	-	A2
M2	-	MS2	MS1	A1
RST	-	MS0	-	B1
NC	-	-	7	B2
STEP	-	9	8	7
DIR	-	9	8	-
	-	-	-	VDD
	-	-	-	GND

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



See [Appendix D](#) for legend

Stand-alone Mode

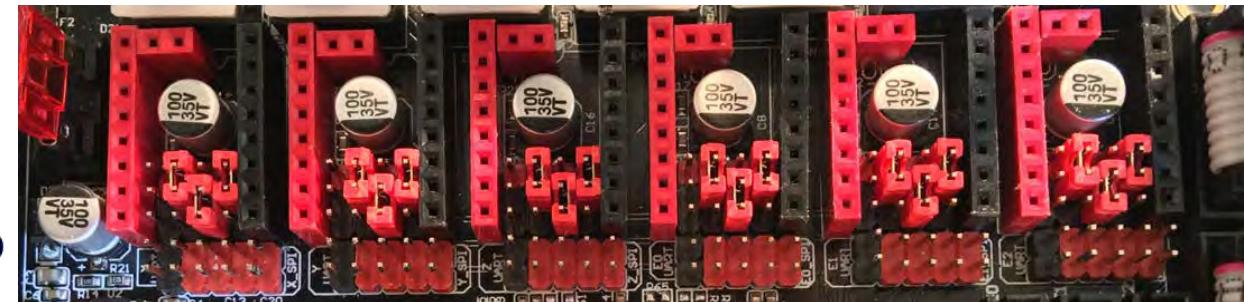
BQU S109

1 / 8

EN	-	-	-	-	VM		
M0	-	-	-	-	GND		
M1	-	-	-	-	A2		
M2	-	-	-	-	A1		
RST	-	-	MS2	MS1	MS0	-	B1
NC	-	9	-	7	-	-	B2
STEP	-	9	8	7	N	VDD	-
DIR	-	-	8	-	-	GND	-

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!

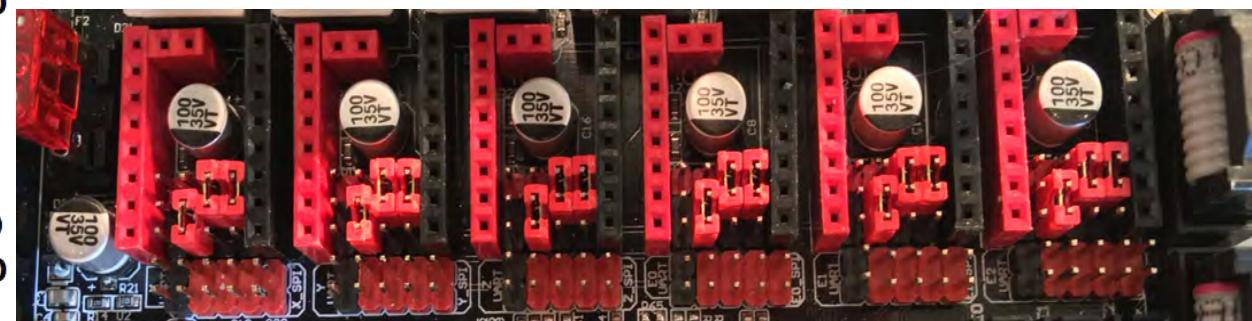


See [Appendix D](#) for legend

1 / 16

EN	-	-	-	-	VM		
M0	-	-	-	-	GND		
M1	-	-	-	-	A2		
M2	-	-	MS2	MS1	MS0	-	A1
RST	-	-	-	-	-	-	B1
NC	-	-	8	7	-	-	B2
STEP	-	9	8	7	N	VDD	-
DIR	-	9	-	-	-	GND	-

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

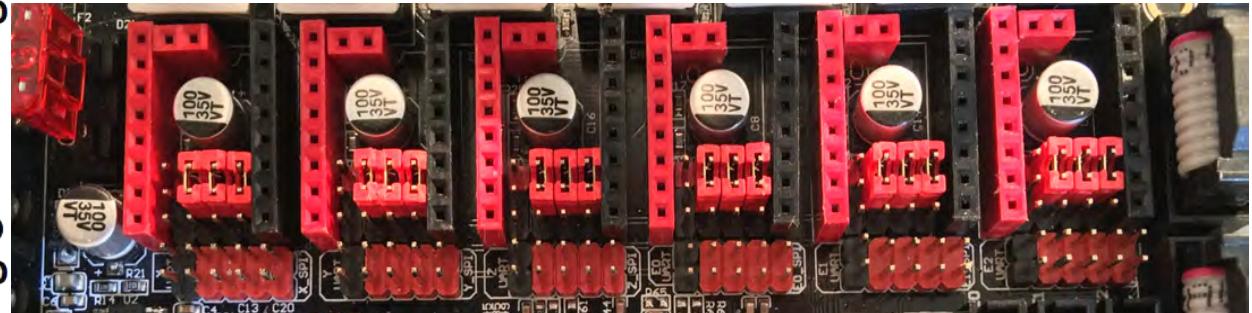


See [Appendix D](#) for legend

1 / 32

EN	-	-	-	-	VM		
M0	-	-	-	-	GND		
M1	-	-	-	-	A2		
M2	-	-	-	-	A1		
RST	-	-	MS2	MS1	MS0	-	B1
NC	-	9	8	7	-	-	B2
STEP	-	9	8	7	N	VDD	-
DIR	-	-	8	-	-	GND	-

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

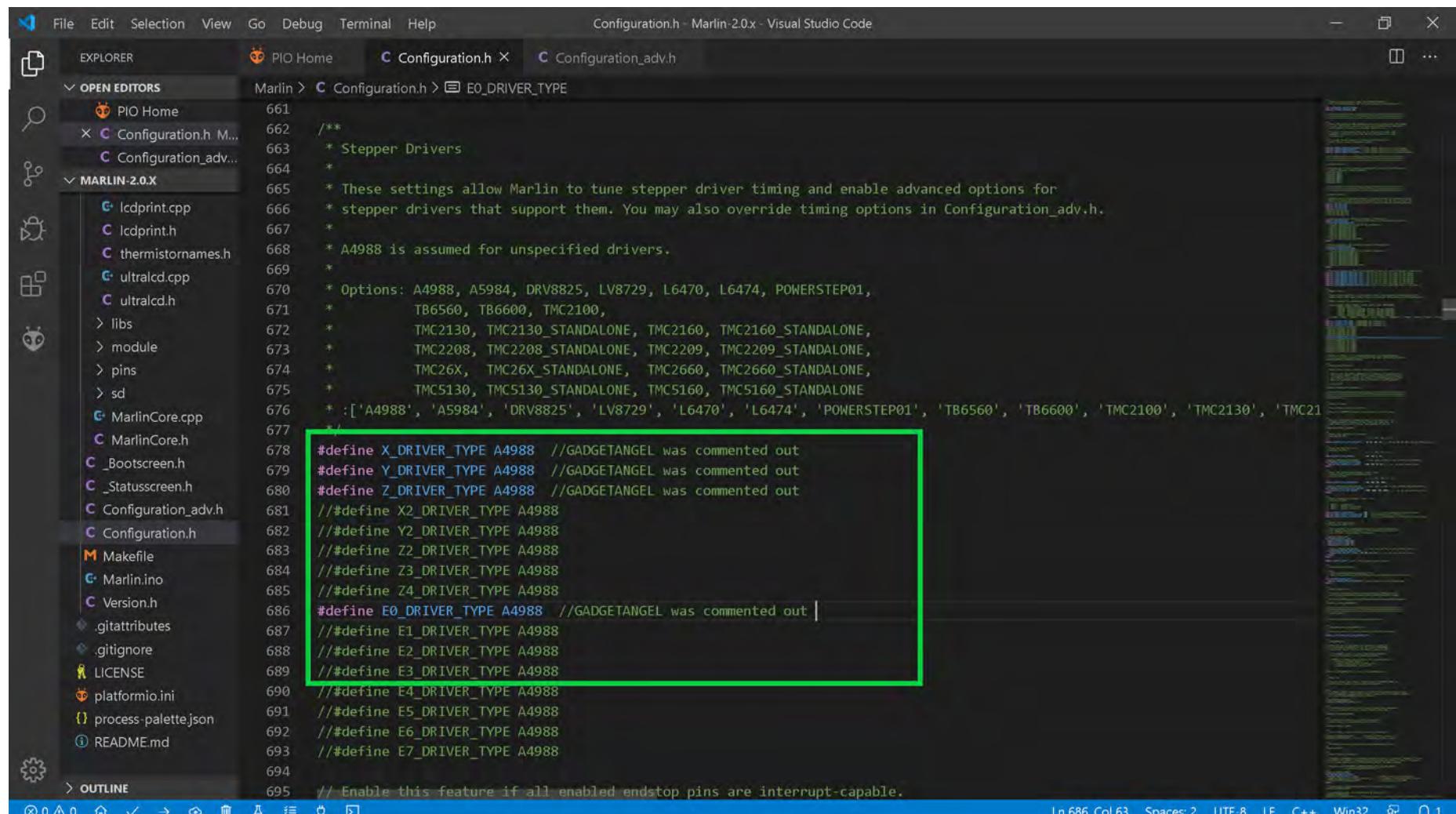


See [Appendix D](#) for legend

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.**



```

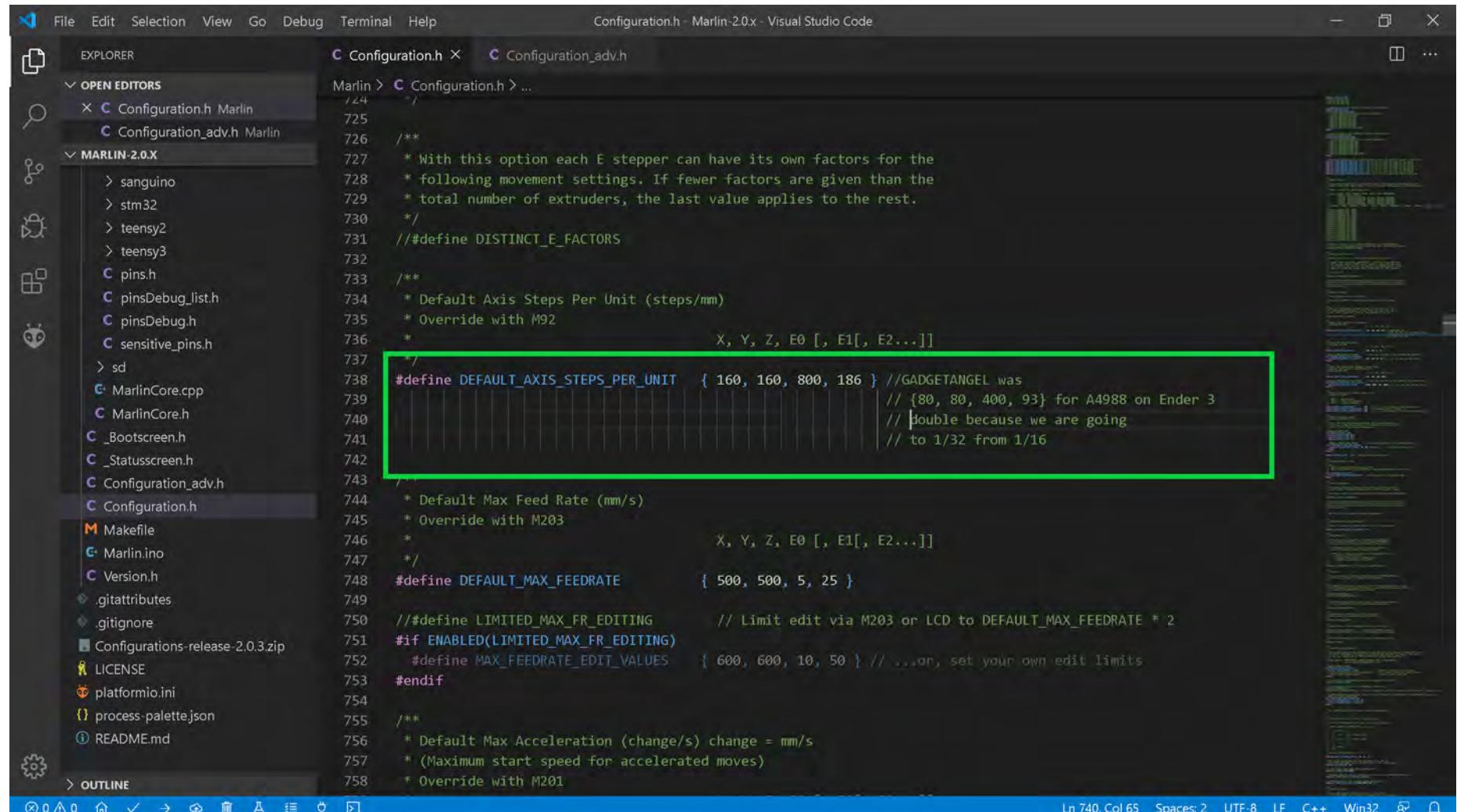
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
MARLIN-2.0.X
Lcdprint.cpp Lcdprint.h thermistornames.h
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
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 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.
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 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:

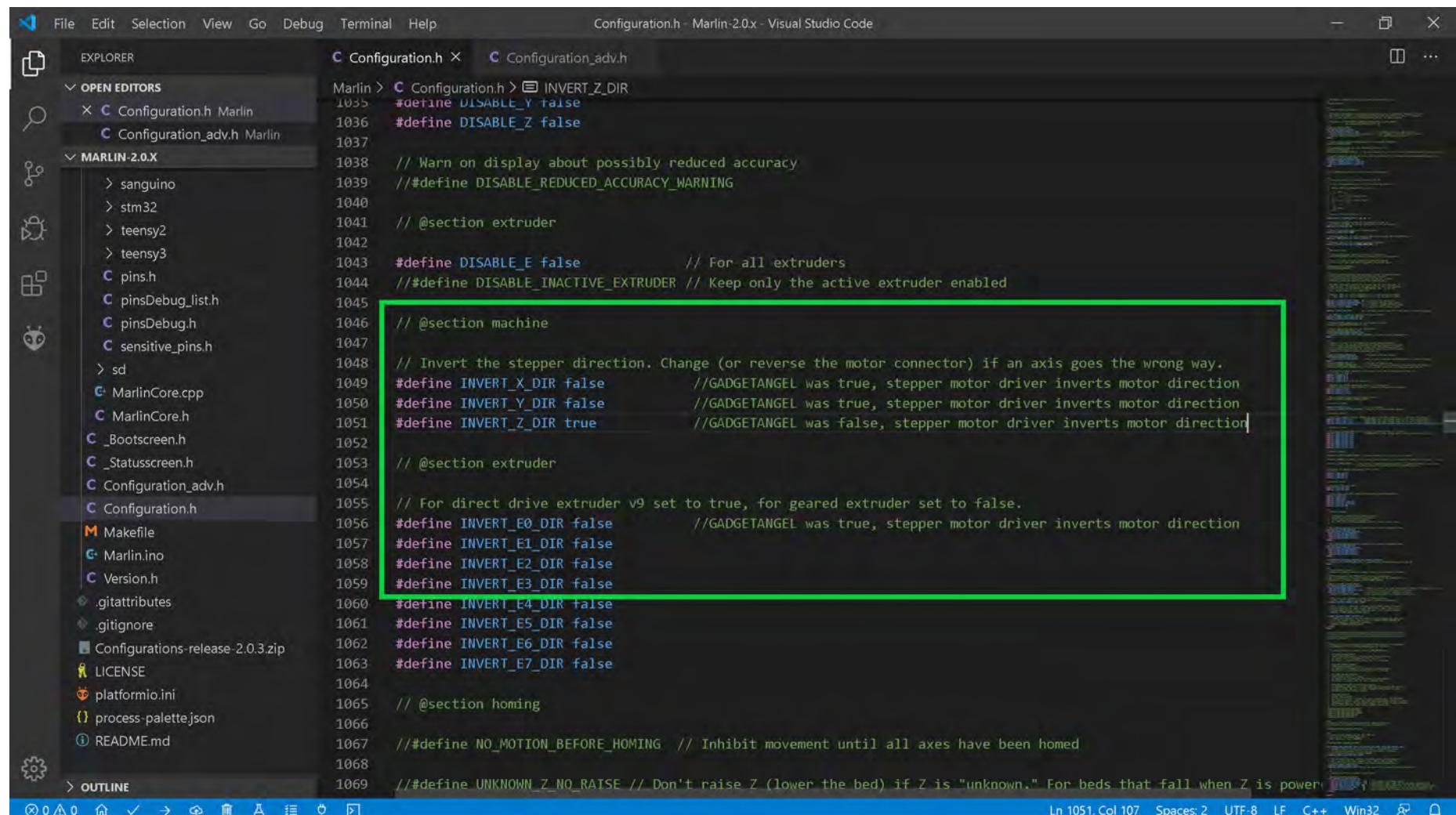
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

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



```

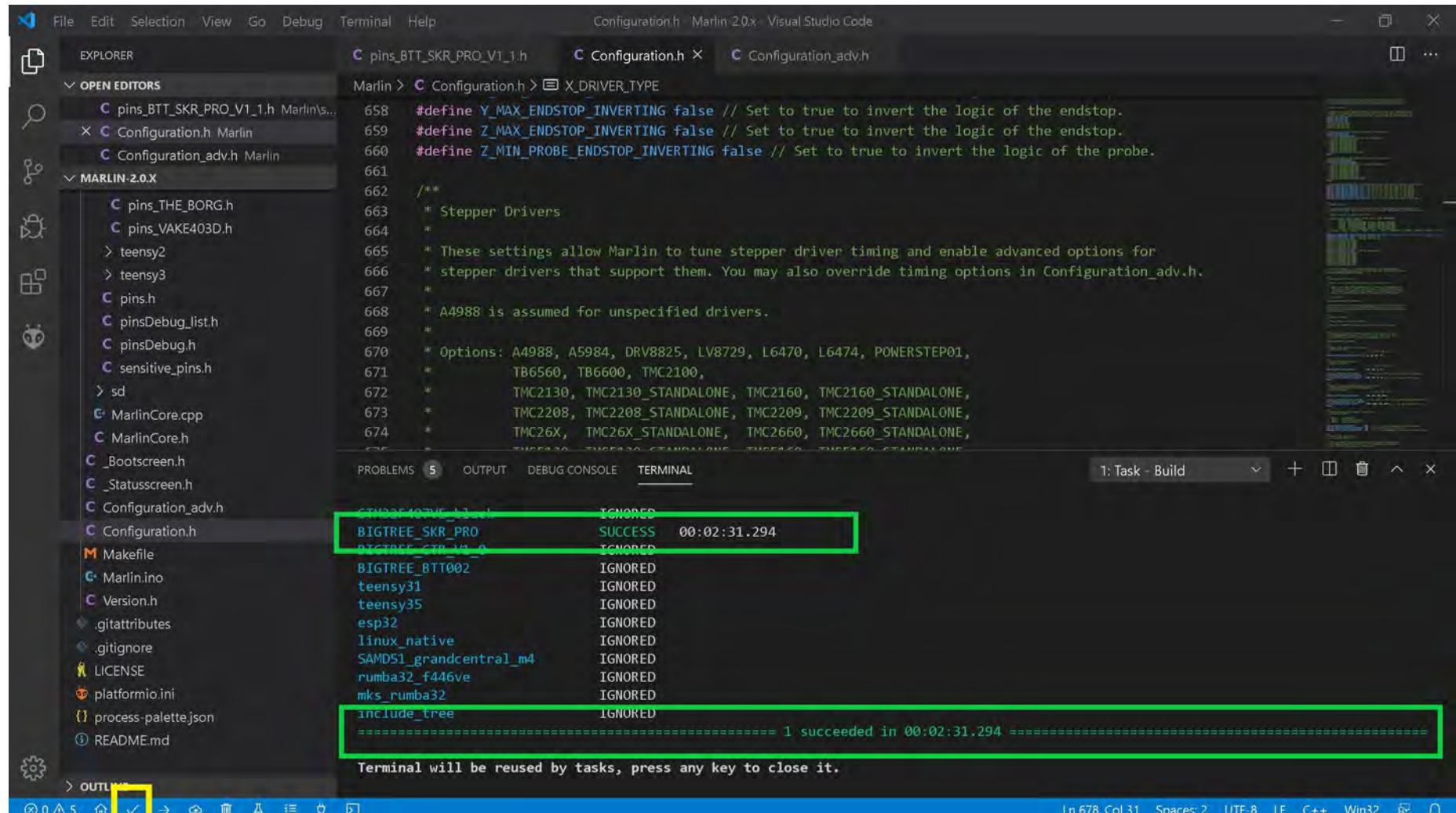
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 1036 #define DISABLE_Z 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.



The screenshot shows the Visual Studio Code interface with the Marlin repository open. The Explorer sidebar on the left lists files and folders related to Marlin 2.0.x, including Configuration.h, pins_BTT_SKR_PRO_V1_1.h, and Configuration_adv.h. The main editor area displays the Configuration.h file with code for endstop inversion and stepper driver timing. The bottom right terminal window shows the build process:

```

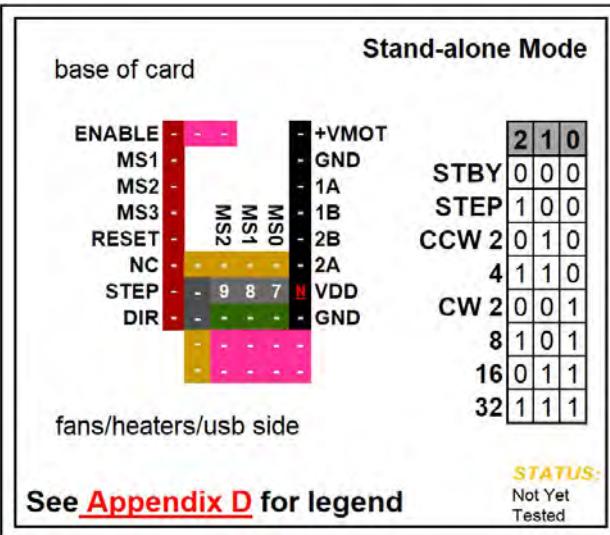
  _TMC26X.h          IGNORED
BIGTREE_SKR_PRO      SUCCESS  00:02:31.294
BIGTREE_CTP_11_0     IGNORED
BIGTREE_BT8002       IGNORED
teensy31             IGNORED
teensy35             IGNORED
esp32                IGNORED
linux_native          IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve        IGNORED
mks_rumba32          IGNORED
include_tree          IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====

```

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

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

FYSETC S109

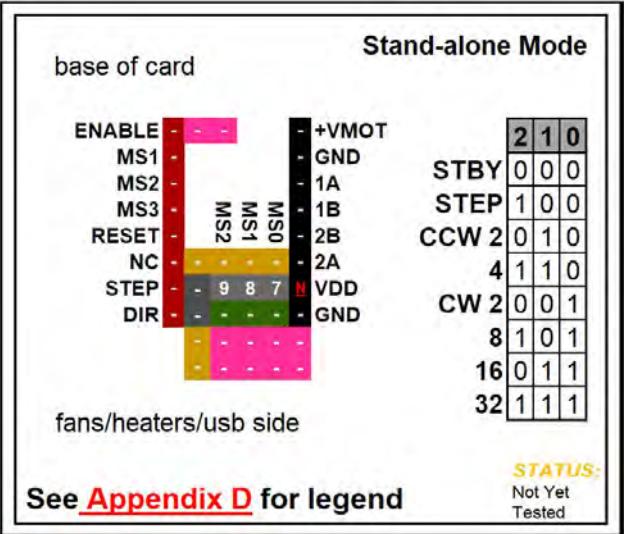


Note: V_{DD} is an Output (**N**), *on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!!*

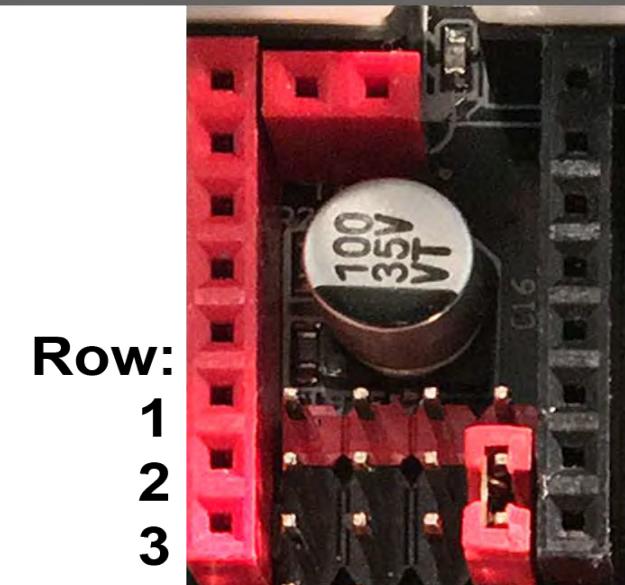
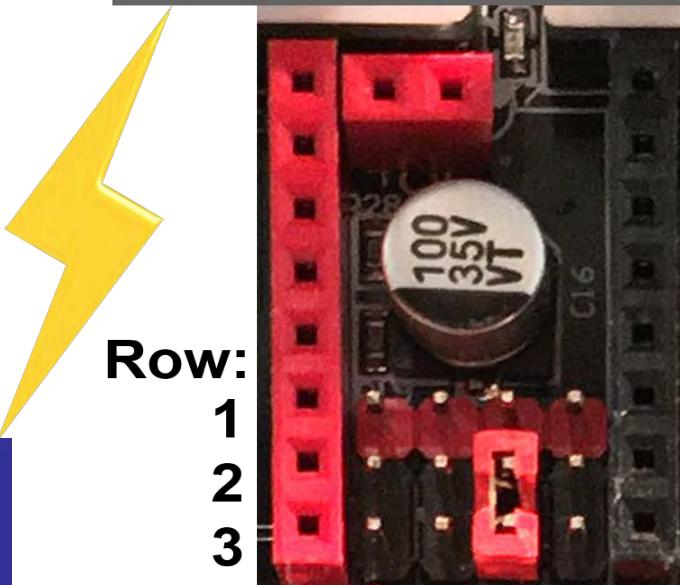
Note: See the next page for information about location of the current sense resistors and how to set V_{ref} on the stepper motor driver board.

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

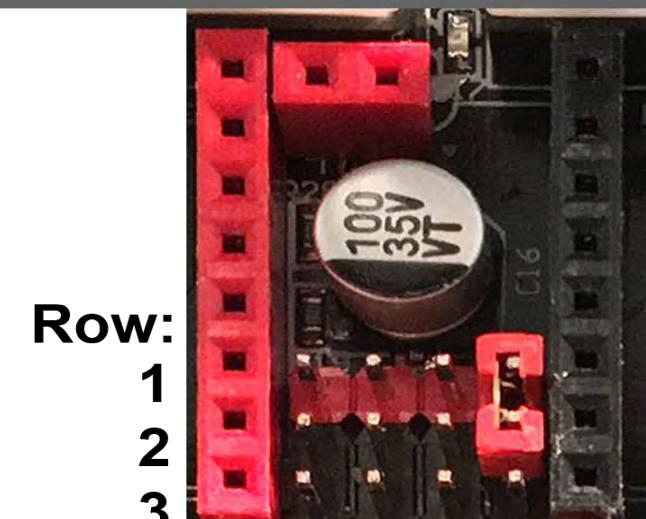
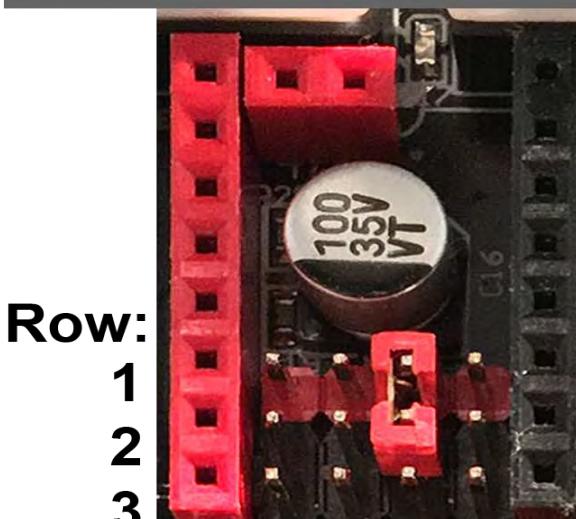
- See next page for the legend that belongs to the above chart.

**FYSETC S109****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

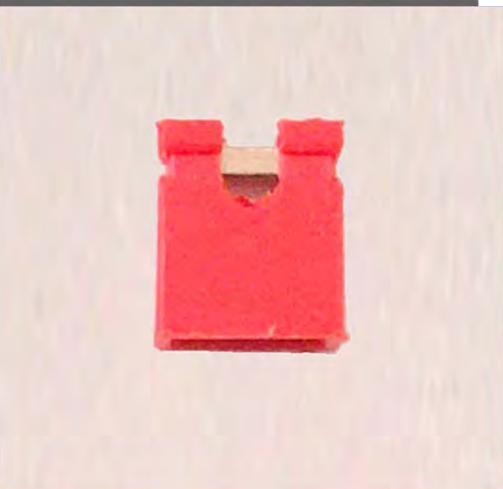
Low ➡ set Jumper between rows 2 and 3

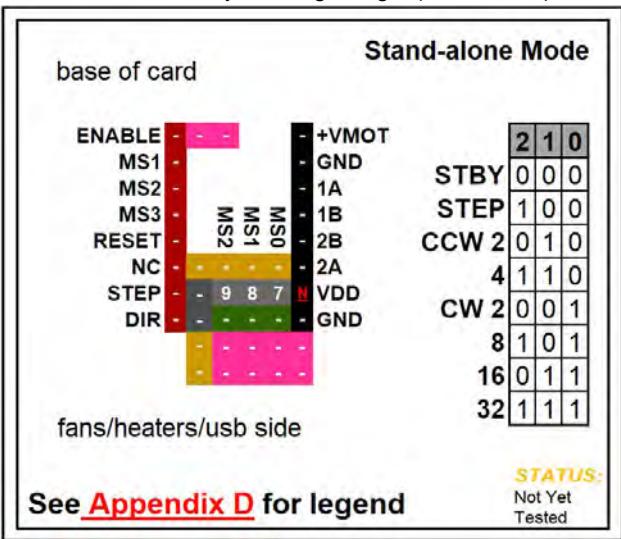


High ➡ set Jumper between rows 1 and 2



This is a Jumper:



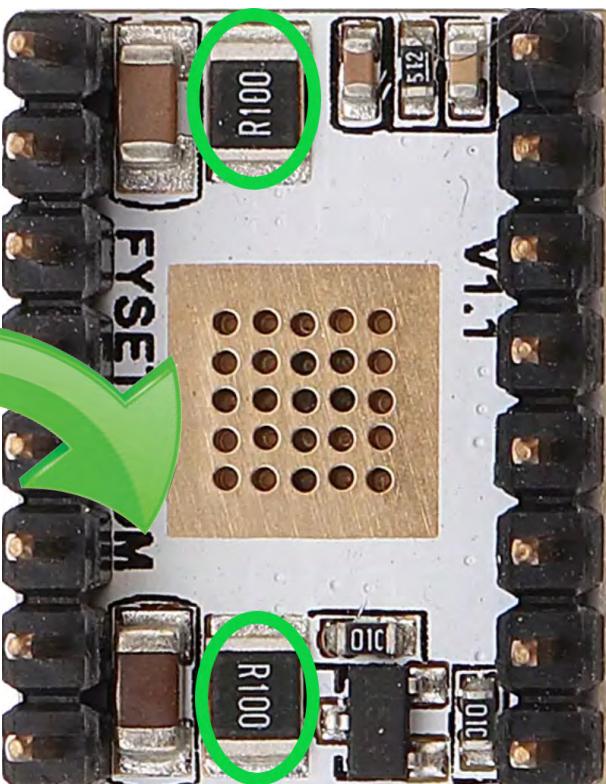


FYSETC S109

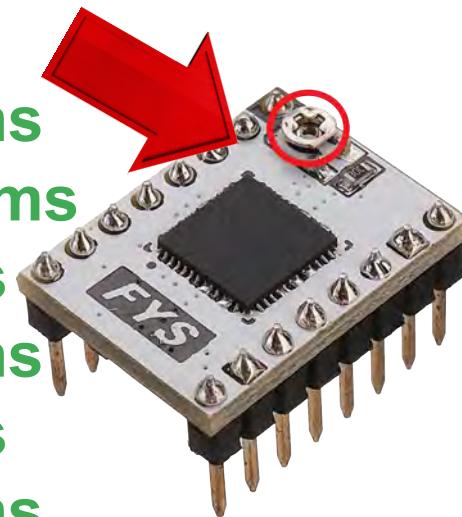
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board. V_{DD} is an Output (N), on a 3.3 V MCU like SKR PRO V1.1 the V_{DD} must be disconnected!

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>

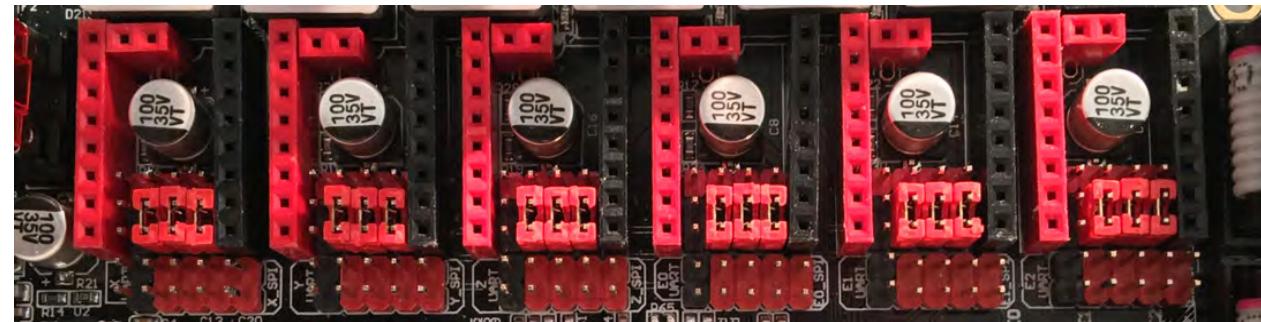
Stand-alone Mode

FYSETC S109

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	2B
NC	-	-	-	2A
STEP	-	9	8	VDD
DIR	-	9	8	GND
	-	7	-	
	-	-	-	

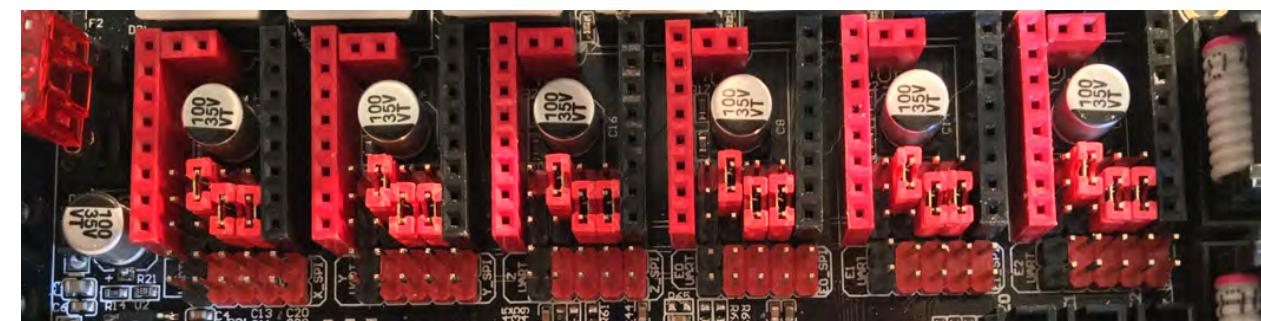
Note: N, on a 3.3 V MCU, like SKR PRO V1.1, **MUST** be disconnected!



See [Appendix D](#) for legend

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

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



See [Appendix D](#) for legend

STAND
BY

STEP

Stand-alone Mode

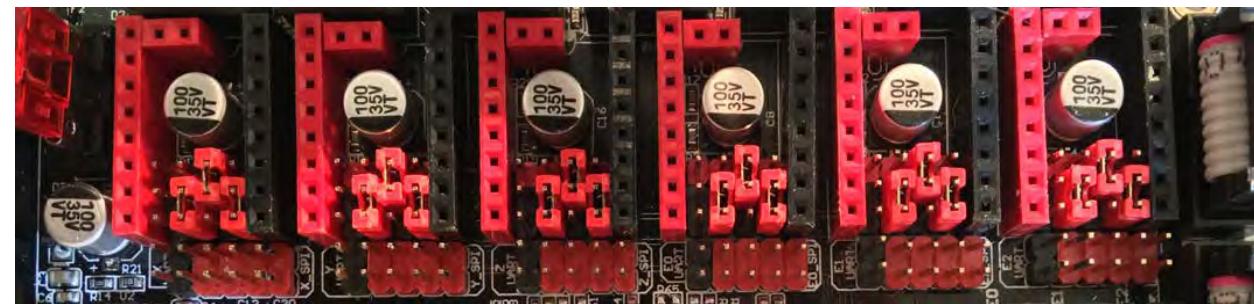
CCW
1 / 2

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

FYSETC S109

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!

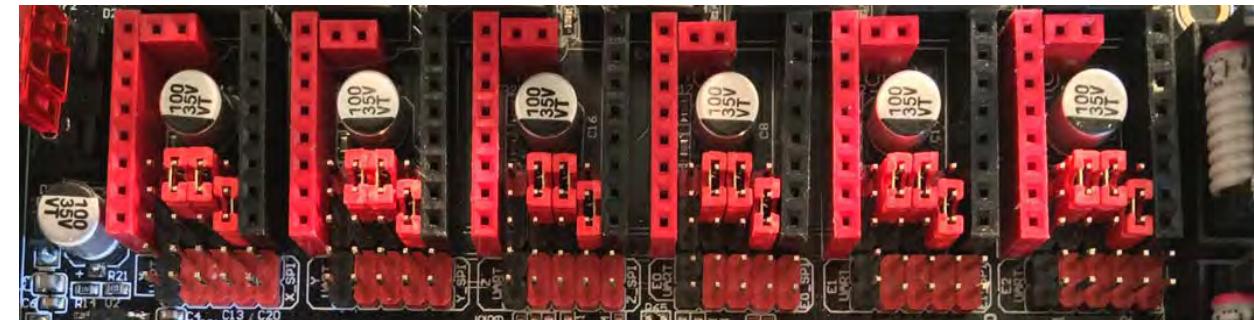


See [Appendix D](#) for legend

1 / 4

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

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

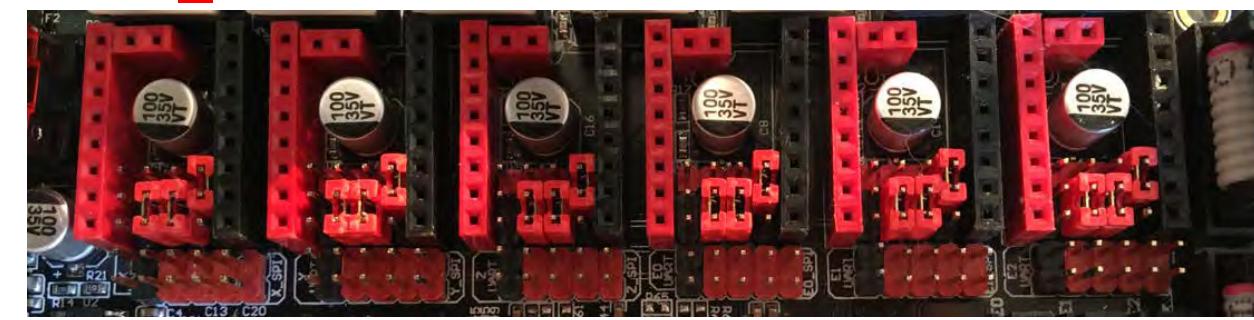


See [Appendix D](#) for legend

CW
1 / 2

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

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



See [Appendix D](#) for legend

Stand-alone Mode

1 / 8

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

FYSETC S109

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".

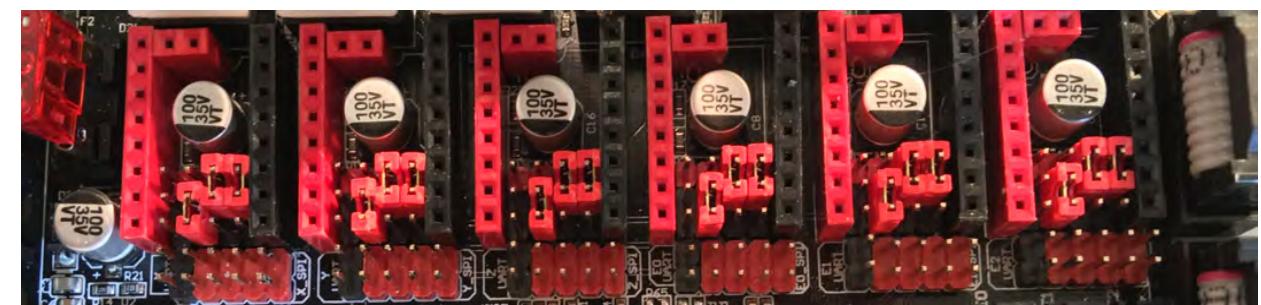


See [Appendix D](#) for legend

1 / 16

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

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

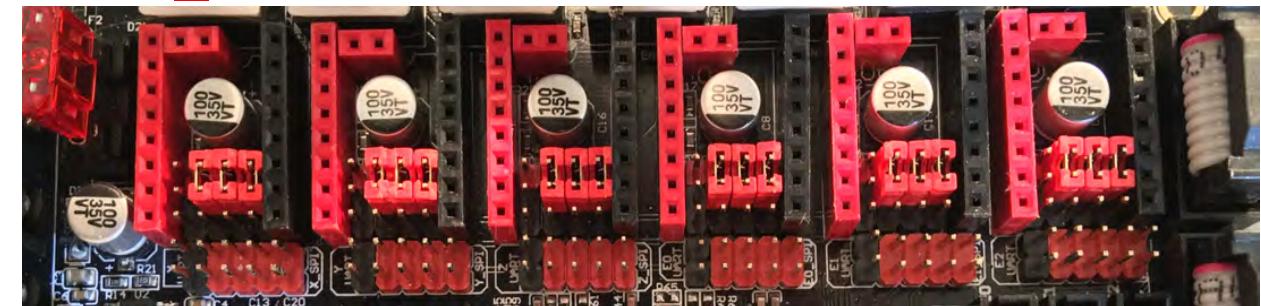


See [Appendix D](#) for legend

1 / 32

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

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

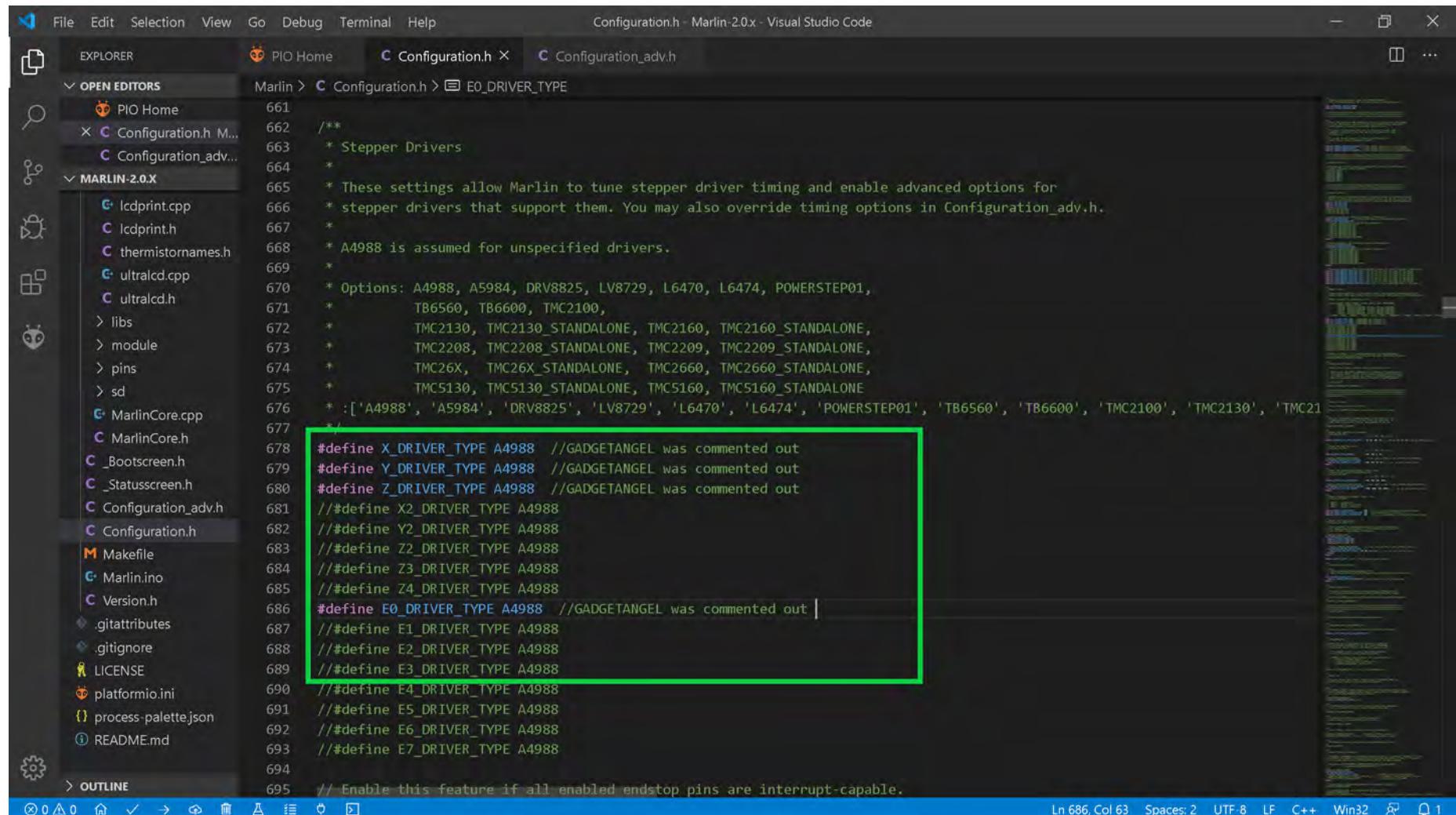


See [Appendix D](#) for legend

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 of code, with the driver definition section highlighted by a green box:

```

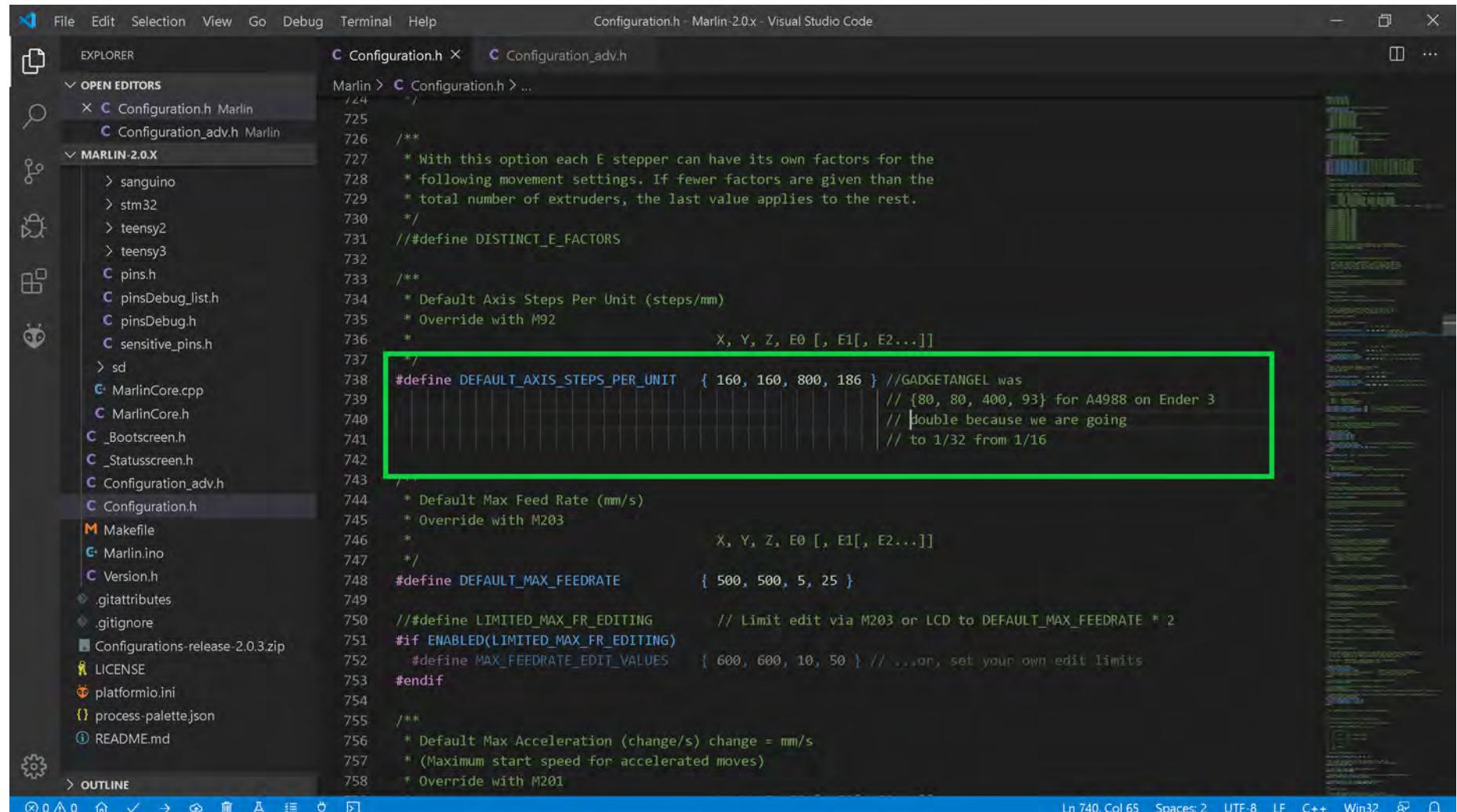
661 /**
662 * Stepper Drivers
663 */
664 /**
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 */
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.

```

- 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

```

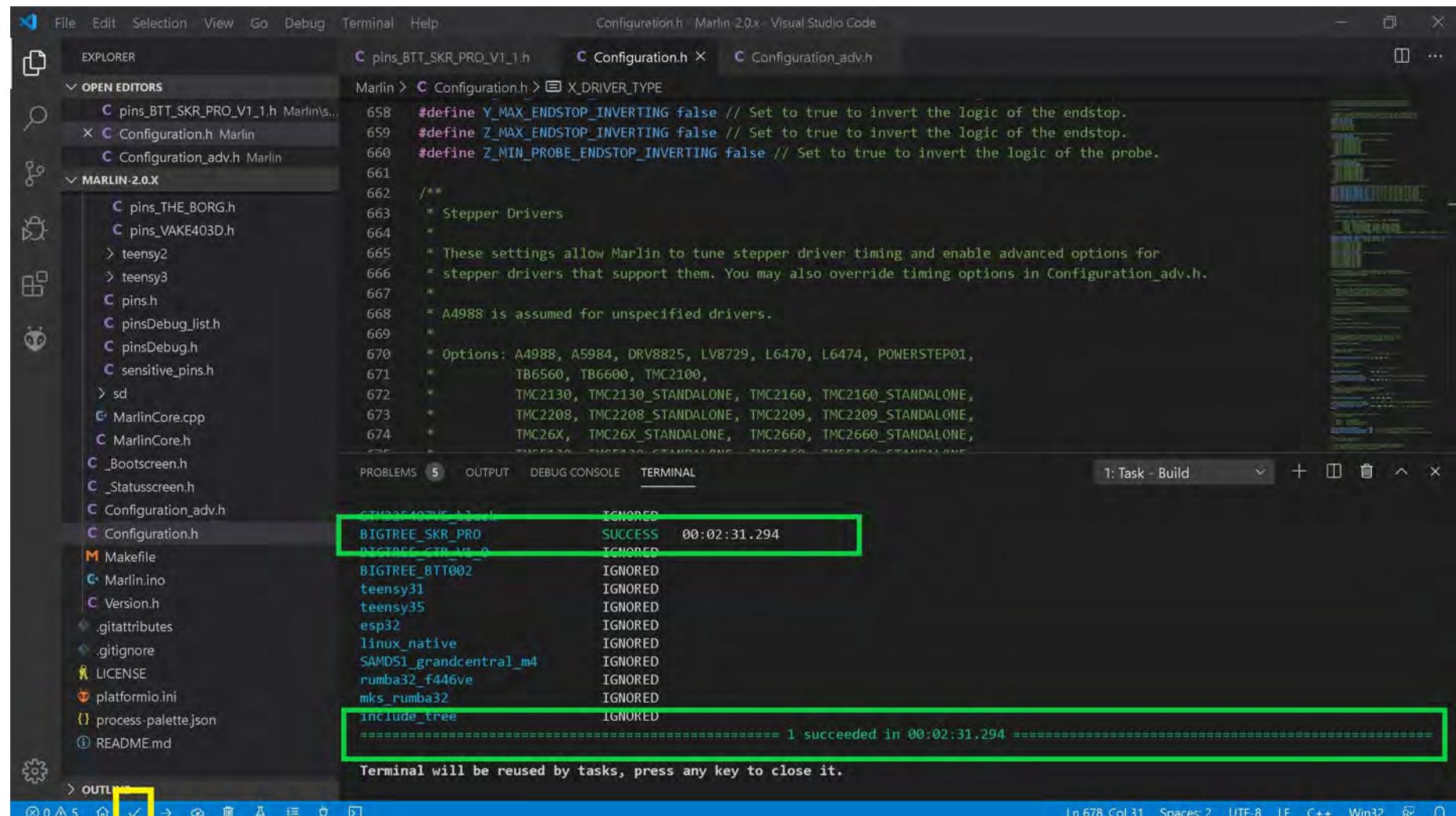
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 1036 #define DISABLE_Z false
MARLIN-2.0.X 1037
  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
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 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.

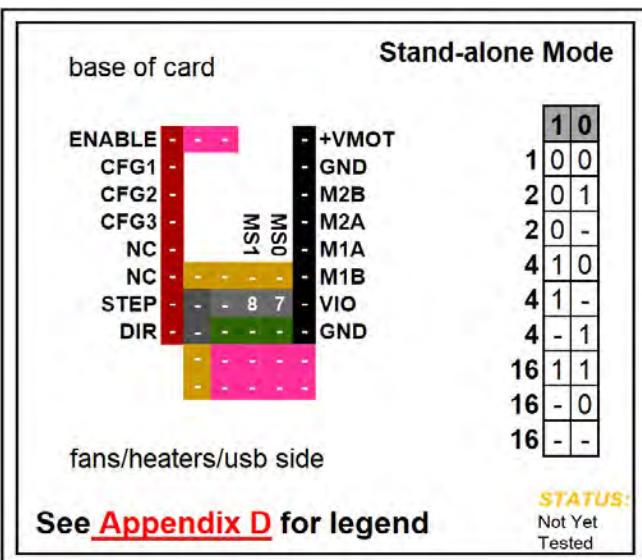


The screenshot shows the Visual Studio Code interface with the following details:

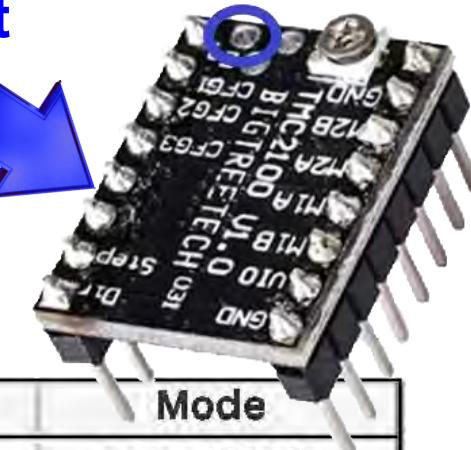
- File Explorer:** Shows files like Configuration.h, pins_BTT_SKR_PRO_V1_1.h, Configuration_adv.h, and MarlinCore.cpp.
- Editor:** Displays the Configuration.h file with code related to endstop inversion and stepper driver timing.
- Terminal:** Shows the build command and its output. The output is highlighted with a green box:


```
CTM20E403VE_L1_11_1 IGNORED
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_CTC_M1_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 =====
```
- Bottom Bar:** Shows the terminal status as "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 [Appendix E](#).

**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}

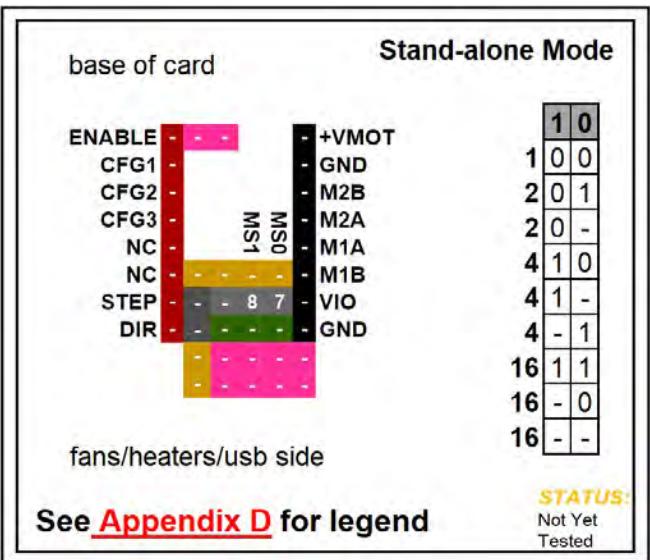


Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2100 Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	Low	High	1 / 2	NONE	spreadCycle
	Low	OPEN	1 / 2	1 / 256	spreadCycle
	High	Low	1 / 4	NONE	spreadCycle
	High	OPEN	1 / 4	1 / 256	spreadCycle
	OPEN	High	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	OPEN	Low	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop

Driving Current Calculation Formula

$$R_S \text{ (Typical Sense Resistor)} = 0.11\Omega$$

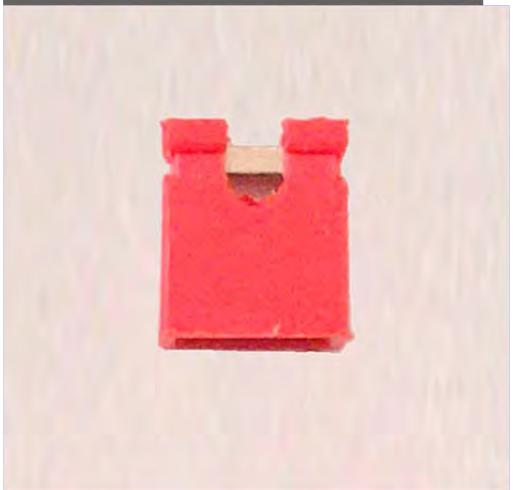
- See next page for the legend that belongs to the above chart.

**BIQU TMC2100**Stand-alone Mode**SKR PRO V1.1 LEGEND for Tri State Stepper Drivers**
PART 1

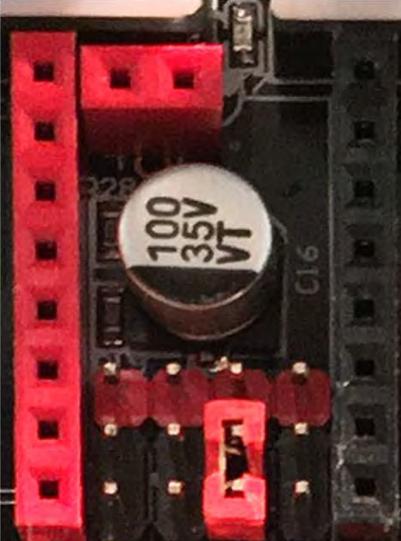
Low ➡ set Jumper between rows
2 and 3



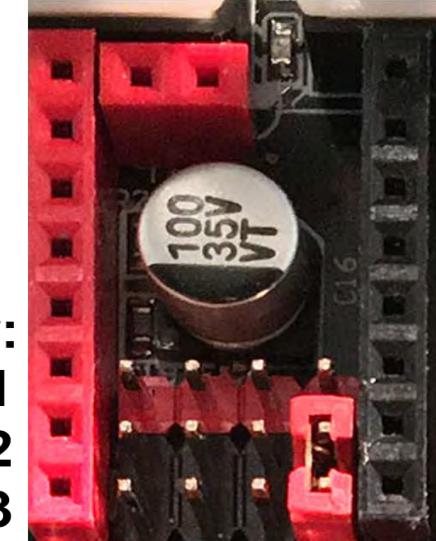
**This is a
Jumper:**



Row:
1
2
3

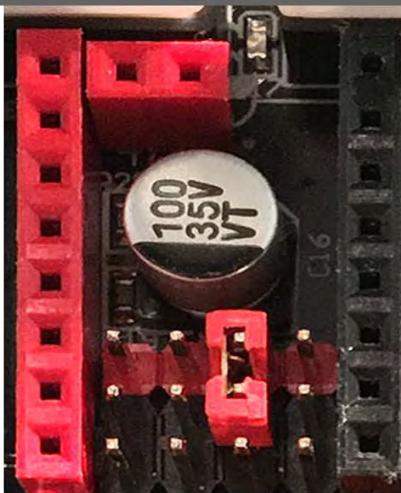


Row:
1
2
3

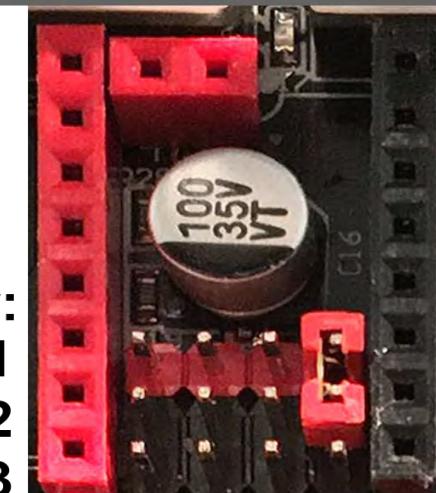


High ➡ set Jumper between rows
1 and 2

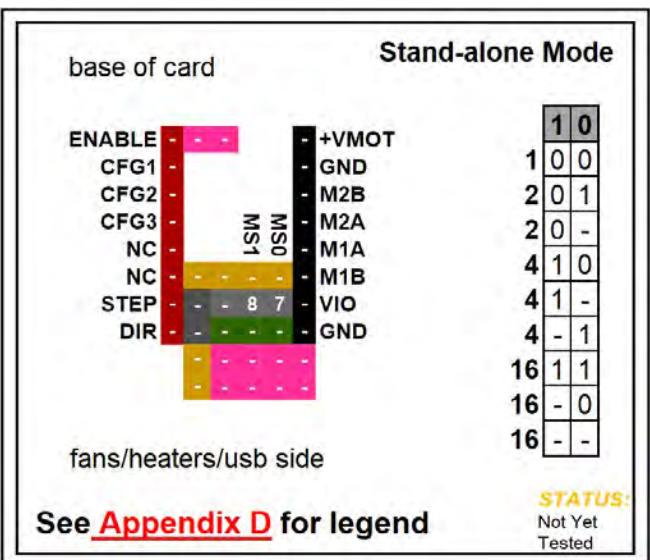
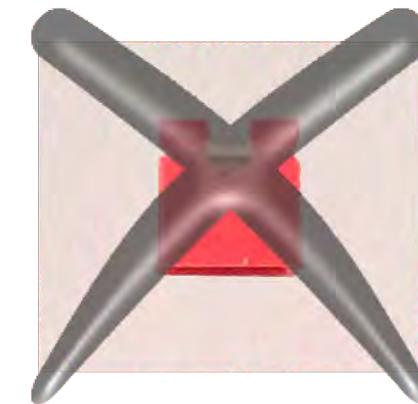
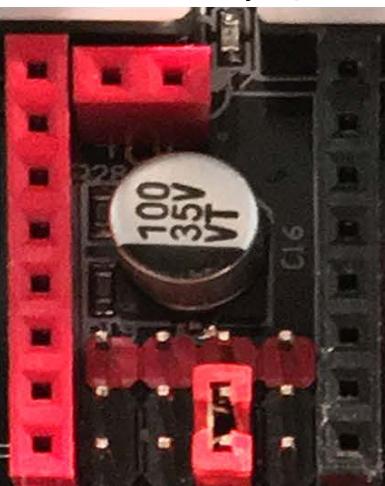
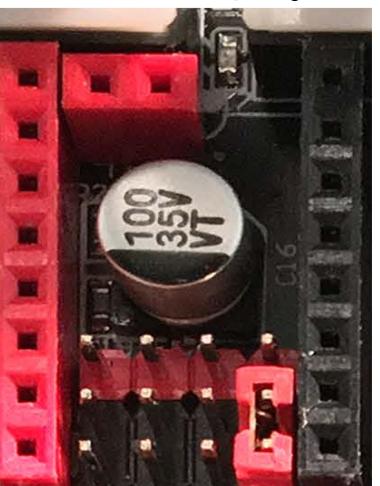
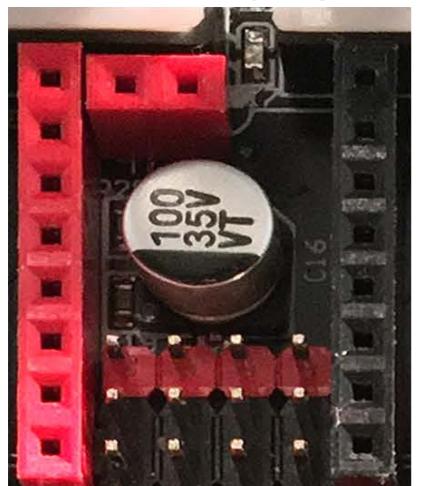
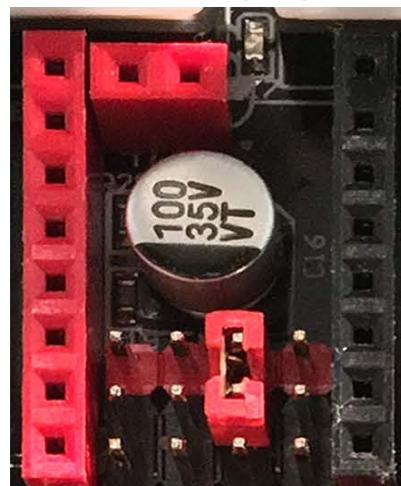
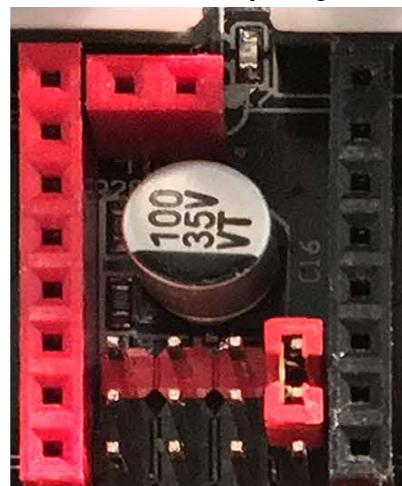
Row:
1
2
3

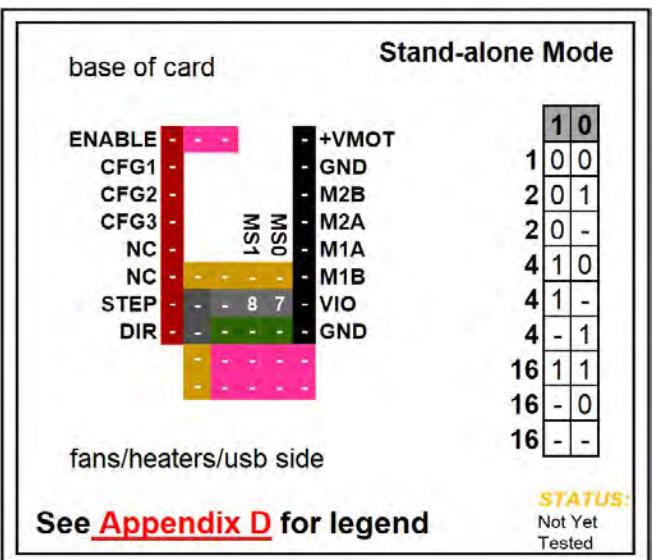


Row:
1
2
3



- Go to the next page.

**BIQU TMC2100**Stand-alone Mode**SKR PRO V1.1 LEGEND for Tri State Stepper Drivers**
PART 2**OPEN ➡ No Jumper****Examples of using OPEN:****OPEN****OPEN****Low****OPEN****Low****OPEN**M
S
1
0M
S
1
0M
S
1
0**OPEN****High****High**M
S
1
0M
S
1
0



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>

Stand-alone Mode

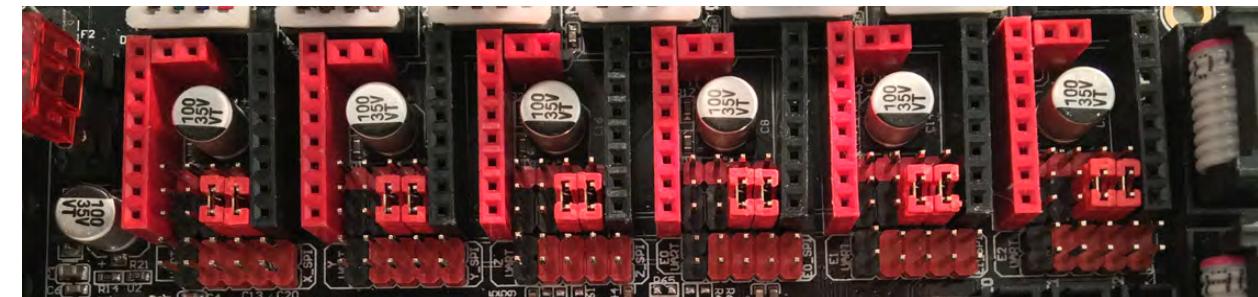
Stand-alone
Mode

STEP

Interpolation:
none

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	- GND
CFG2	-	-	- M2B
CFG3	-	-	- M2A
NC	-	MS1	- M1A
NC	-	MS0	- M1B
STEP	-	8 7	- VIO
DIR	-	8 7	- GND



See [Appendix D](#) for legend

Stand-alone
Mode

1 / 2

Interpolation:
none

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	- GND
CFG2	-	-	- M2B
CFG3	-	-	- M2A
NC	-	MS1	- M1A
NC	-	MS0	- M1B
STEP	-	8 7	- VIO
DIR	-	8 -	- GND



See [Appendix D](#) for legend

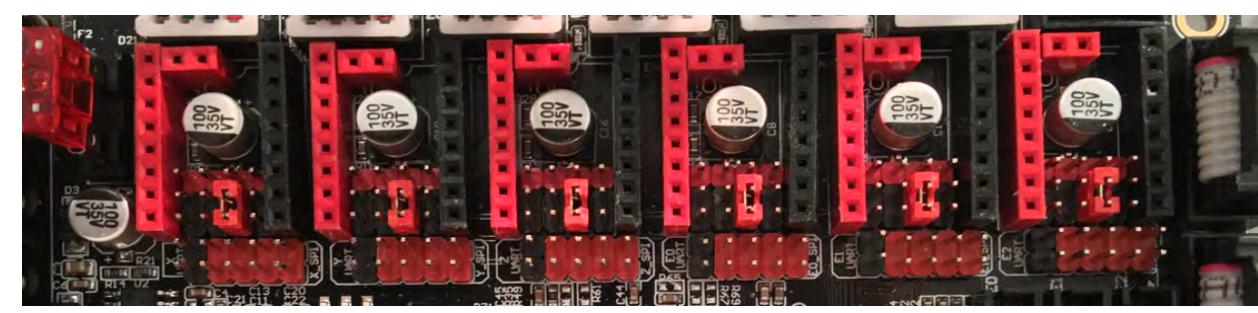
Stand-alone
Mode

1 / 2

Interpolation:
1 / 256

SpreadCycle

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



See [Appendix D](#) for legend

Stand-alone Mode

Stand-alone
Mode

1 / 4

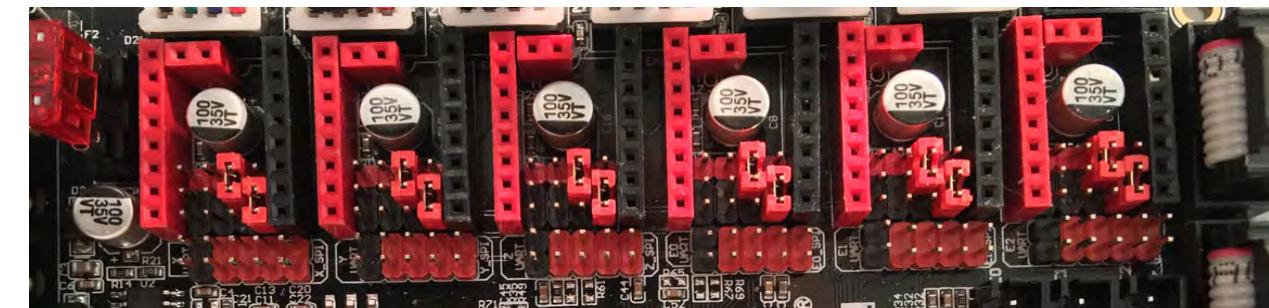
Interpolation:
none

SpreadCycle

ENABLE	[red]	[red]	-	+VMOT
CFG1	[white]	[white]	-	GND
CFG2	[white]	[white]	-	M2B
CFG3	[white]	[white]	-	M2A
NC	[yellow]	MS1	MS0	M1A
NC	[yellow]	8	-	M1B
STEP	[grey]	8	7	VIO
DIR	[green]	-	7	GND

BIGU TMC2100

Stand-alone Mode



See [Appendix D](#) for legend

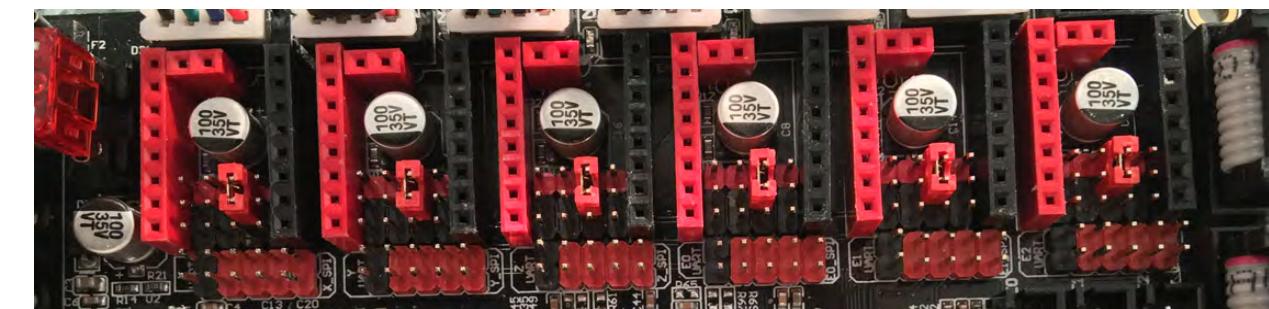
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

SpreadCycle

ENABLE	[red]	[red]	-	+VMOT
CFG1	[white]	[white]	-	GND
CFG2	[white]	[white]	-	M2B
CFG3	[white]	[white]	-	M2A
NC	[yellow]	MS1	MS0	M1A
NC	[yellow]	8	-	M1B
STEP	[grey]	8	-	VIO
DIR	[green]	-	7	GND



See [Appendix D](#) for legend

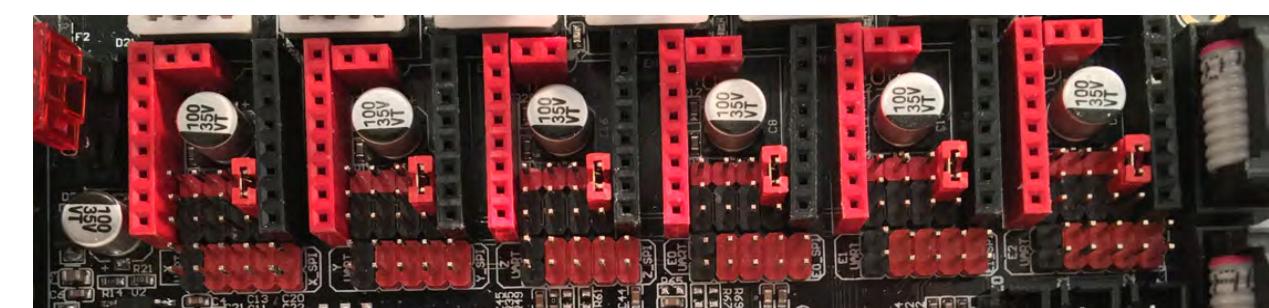
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

StealthChop

ENABLE	[red]	[red]	-	+VMOT
CFG1	[white]	[white]	-	GND
CFG2	[white]	[white]	-	M2B
CFG3	[white]	[white]	-	M2A
NC	[yellow]	MS1	MS0	M1A
NC	[yellow]	-	7	M1B
STEP	[grey]	-	7	VIO
DIR	[green]	-	-	GND



See [Appendix D](#) for legend

Stand-alone Mode

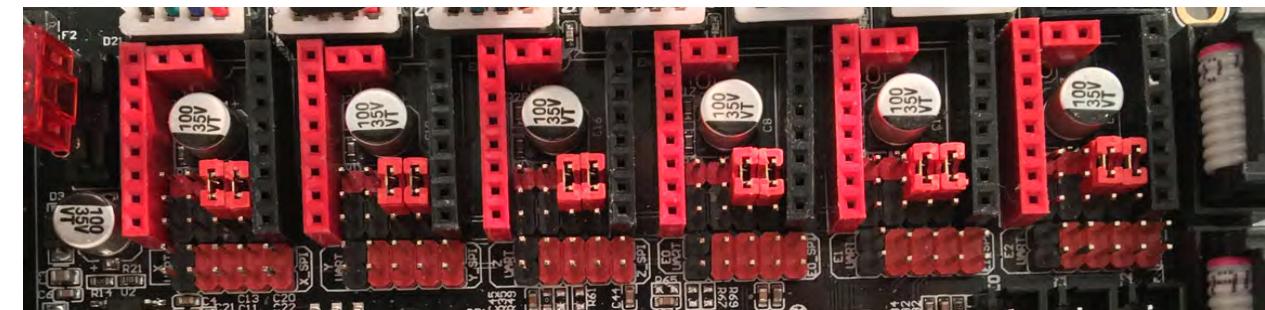
Stand-alone
Mode

1 / 16

Interpolation:
none

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	- GND
CFG2	-	-	- M2B
CFG3	-	-	- M2A
NC	-	MS1	- M1A
NC	8	7	- M1B
STEP	-	8	- VIO
DIR	-	-	- GND



See [Appendix D](#) for legend

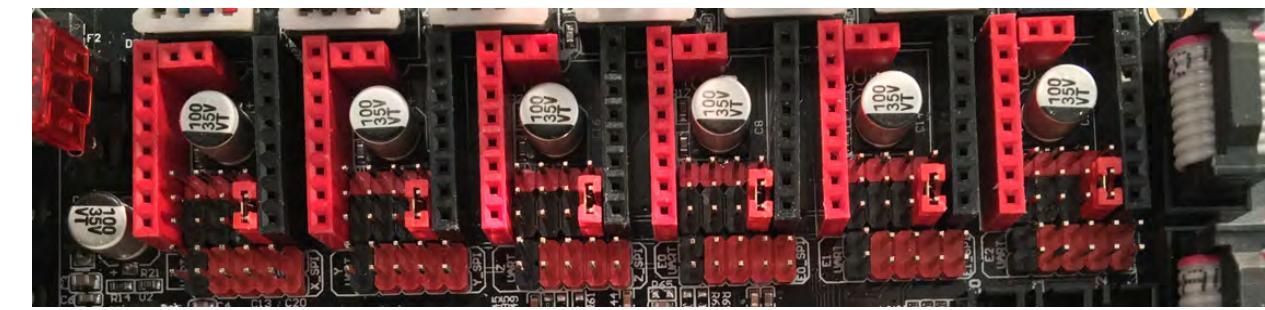
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

SpreadCycle

ENABLE	-	-	+VMOT
CFG1	-	-	- GND
CFG2	-	-	- M2B
CFG3	-	-	- M2A
NC	-	MS1	- M1A
NC	-	-	- M1B
STEP	-	-	- VIO
DIR	-	7	- GND



See [Appendix D](#) for legend

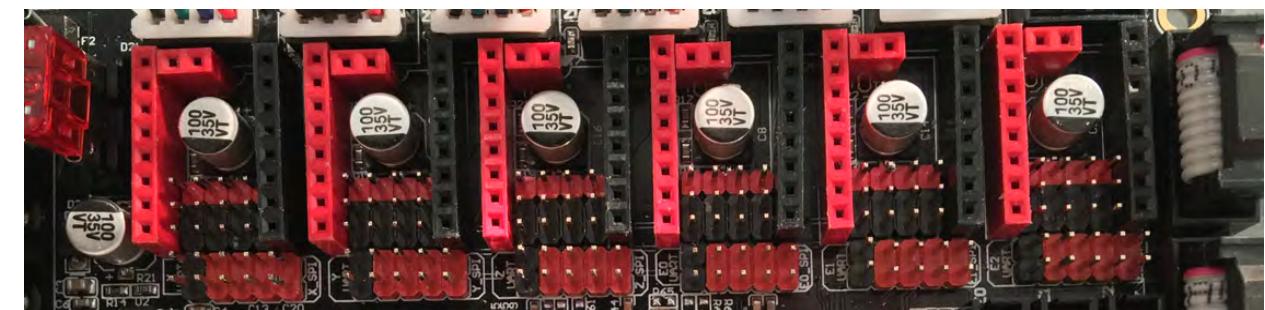
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

StealthChop

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

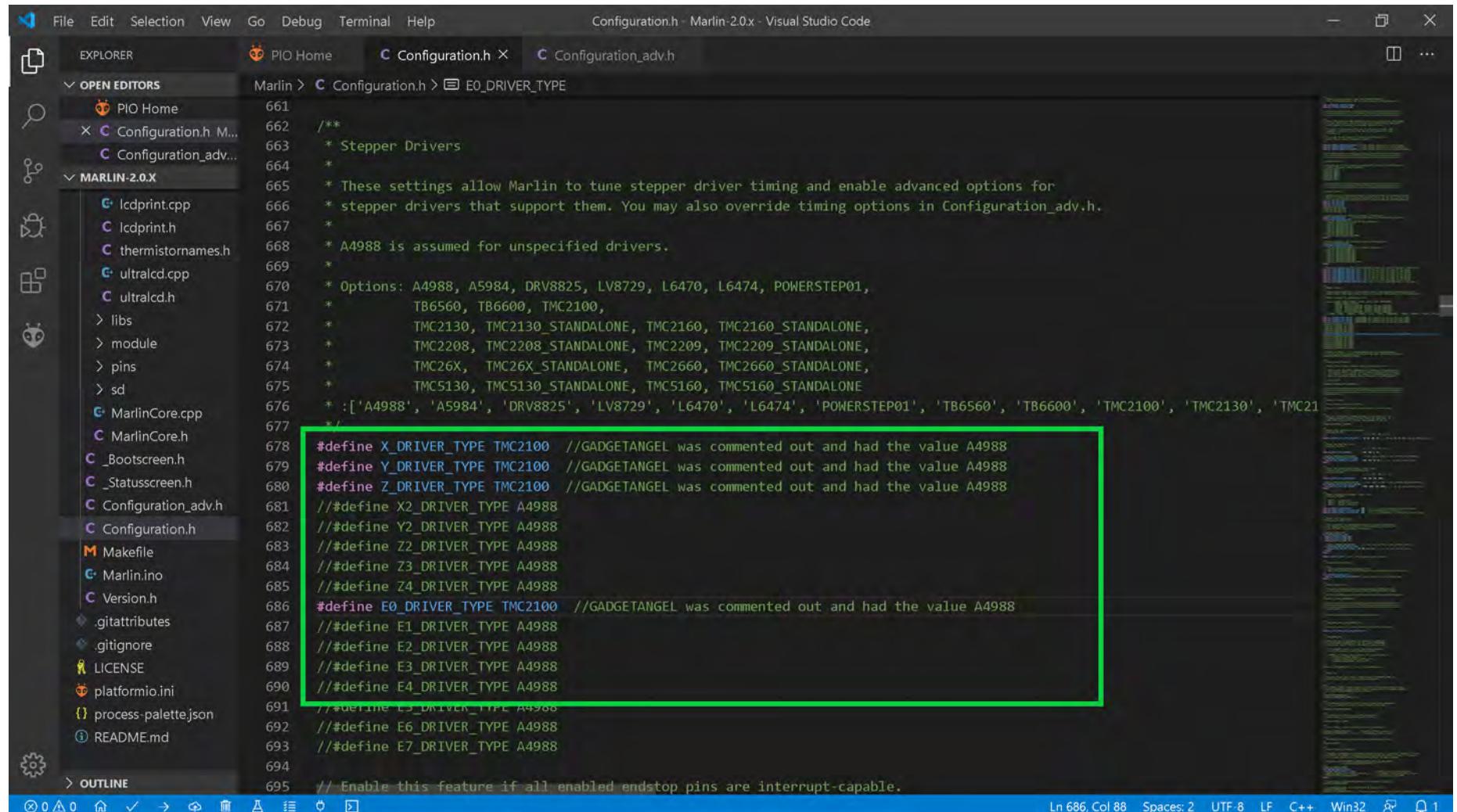


See [Appendix D](#) for legend

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 ("//").



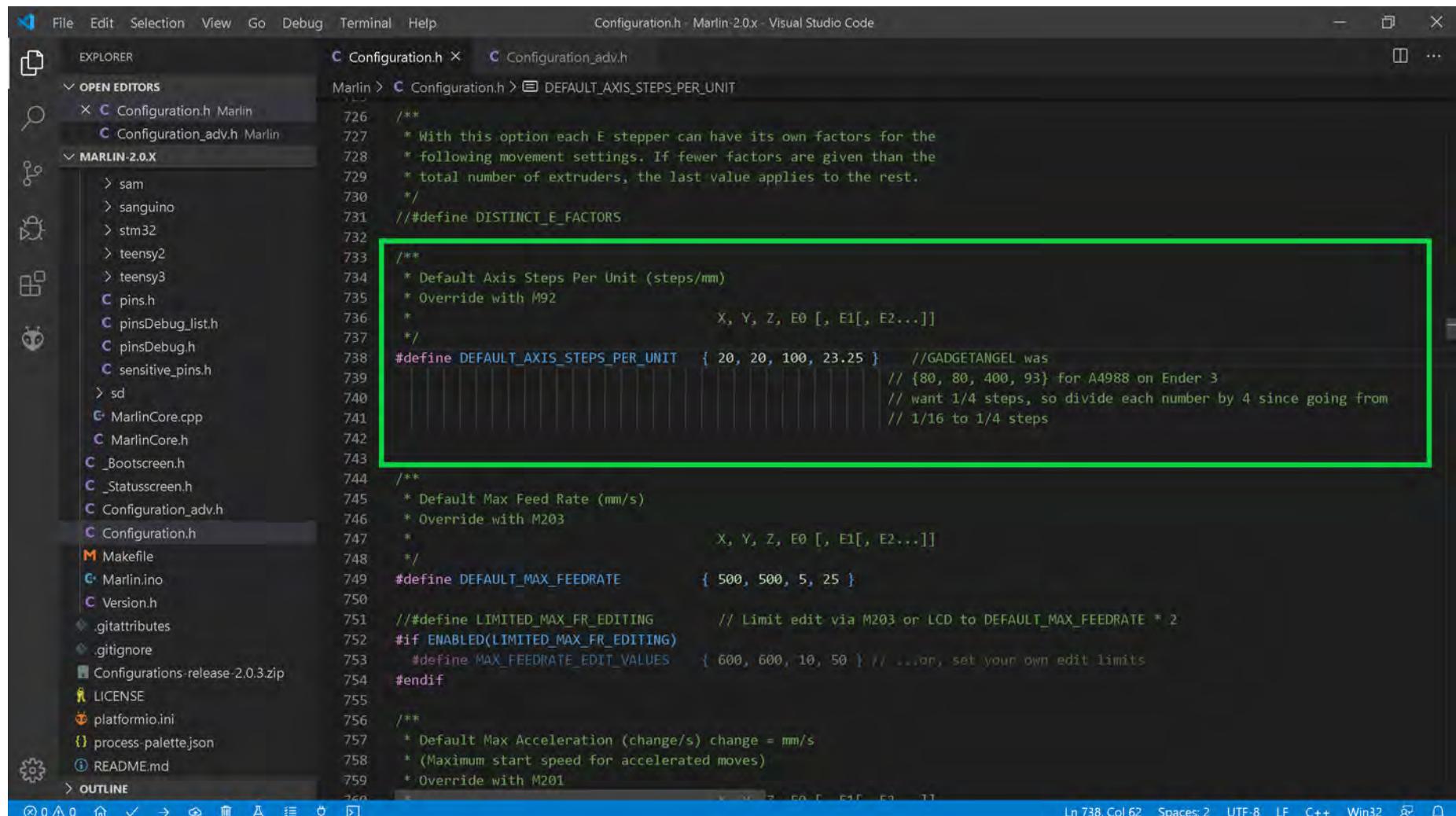
The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following driver type definitions:

```
#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 X2_DRIVER_TYPE A4988
 //#define Y2_DRIVER_TYPE A4988
 //#define Z2_DRIVER_TYPE A4988
 //#define Z3_DRIVER_TYPE A4988
 //#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2100 //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
91 // Enable this feature if all enabled endstop pins are interrupt-capable.
```

- 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.



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:

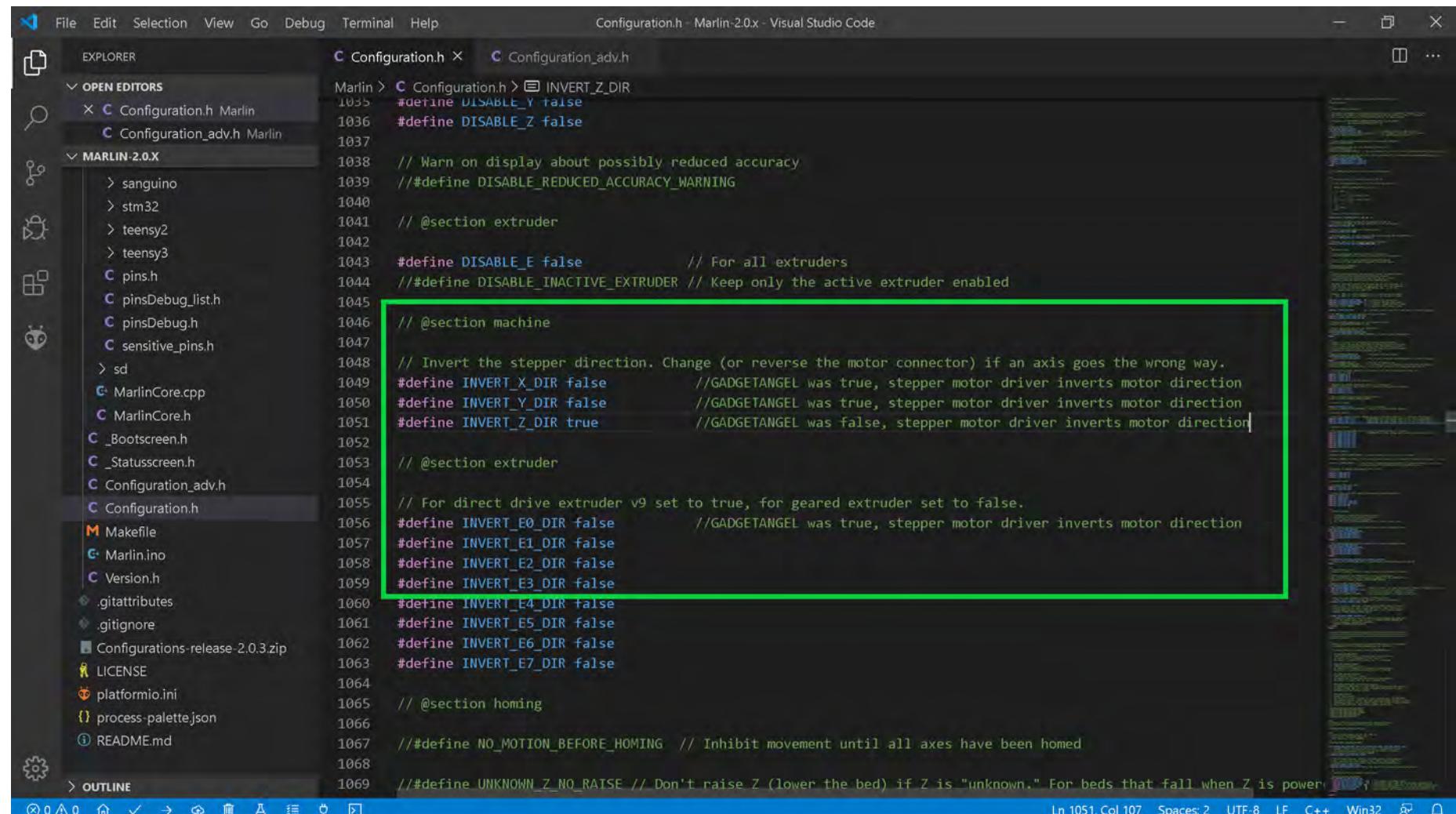
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps
```

The code editor's status bar at the bottom indicates: 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



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
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 on

```

Ln 1051, Col 107 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

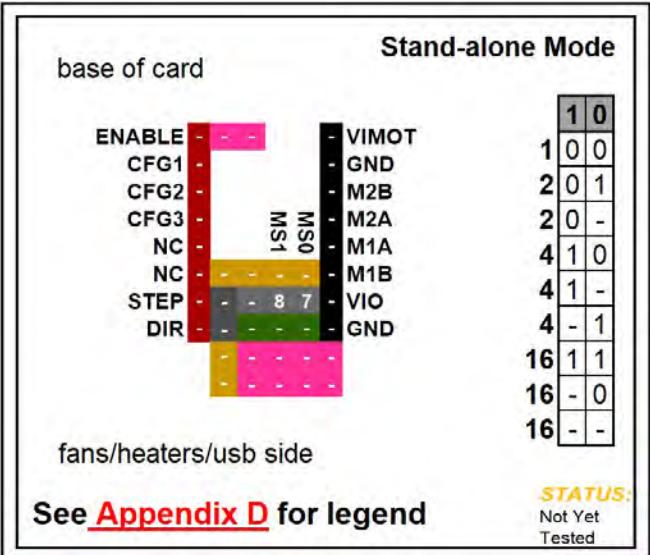
- The end of Marlin setup for BIQU TMC2100 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

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

- File Explorer:** Shows files like Configuration.h, pins_BTT_SKR_PRO_V1_1.h, Configuration_adv.h, and MarlinCore.cpp.
- Editor:** Displays the Configuration.h file with code related to TMC2100 drivers.
- Terminal:** Shows the build output:


```
===== 1 succeeded in 00:02:31.294 =====
```
- Bottom Status Bar:** Shows the terminal will be reused by tasks, and the status: Ln 678, Col 31, Spaces: 2, UTF-8, LF, C++, Win32.

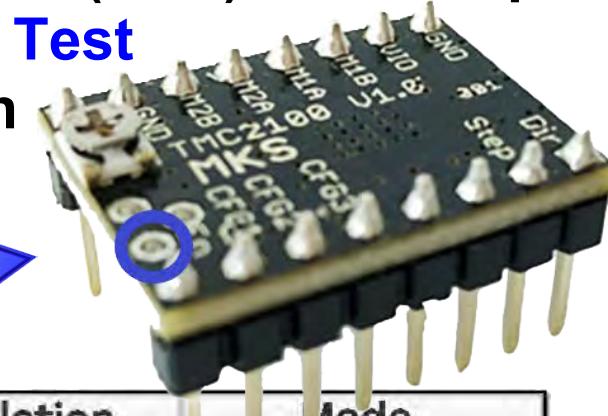
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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



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	Low	Low	1	NONE	spreadCycle
	Low	High	1 / 2	NONE	spreadCycle
	Low	OPEN	1 / 2	1 / 256	spreadCycle
	High	Low	1 / 4	NONE	spreadCycle
	High	OPEN	1 / 4	1 / 256	spreadCycle
	OPEN	High	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	OPEN	Low	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:

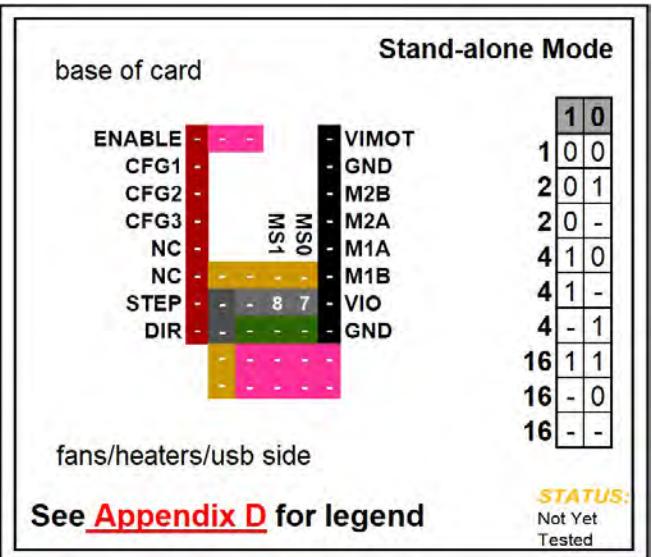
$$I_{MAX} = I_{MAX} * 0.90$$

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

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

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

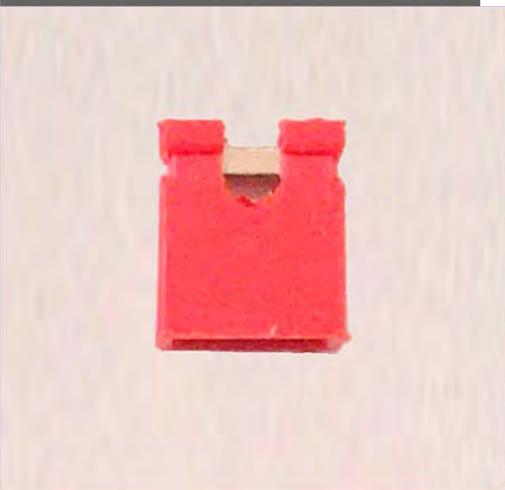
- See next page for the legend that belongs to the above chart.

**MKS TMC2100**Stand-alone Mode**SKR PRO V1.1 LEGEND for Tri State Stepper Drivers**
PART 1

Low ➤ set Jumper between rows 2 and 3

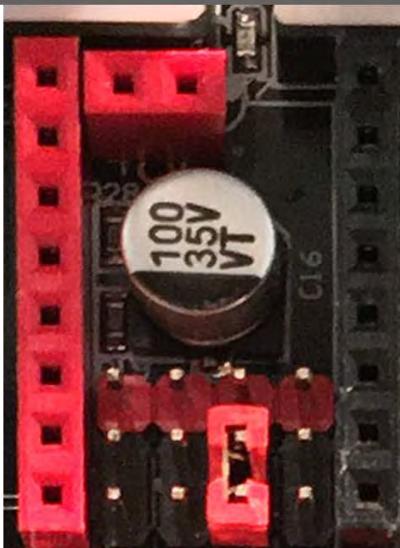


This is a Jumper:



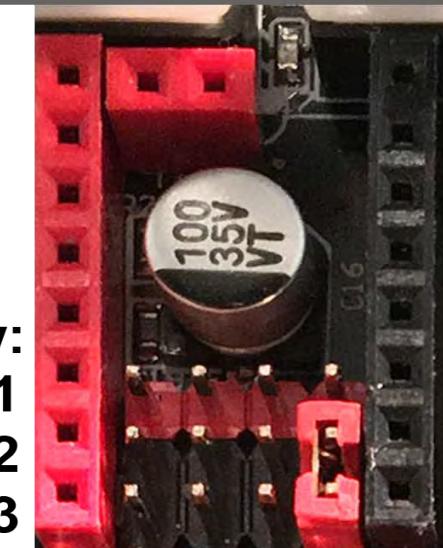
Row:

1
2
3



Row:

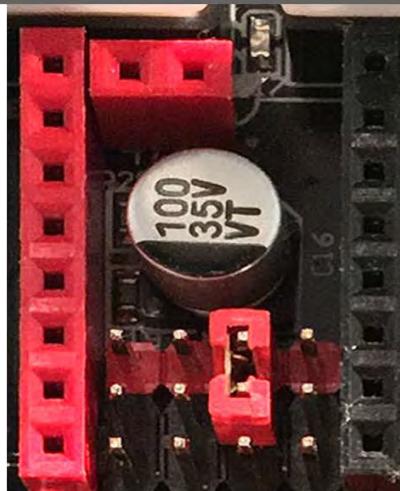
1
2
3



High ➤ set Jumper between rows 1 and 2

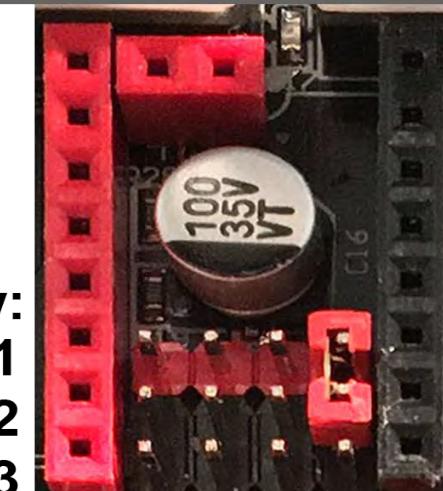
Row:

1
2
3

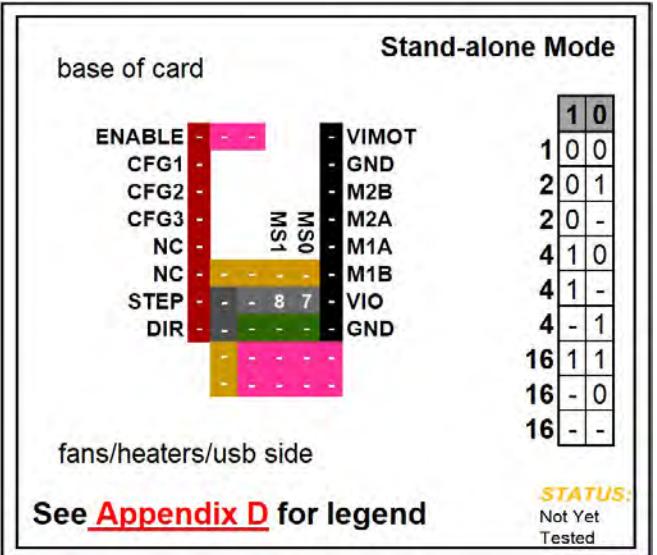


Row:

1
2
3



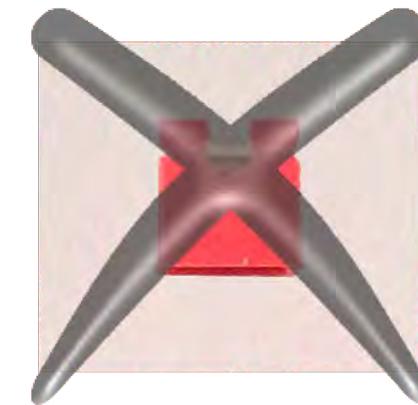
- Go to the next page.

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

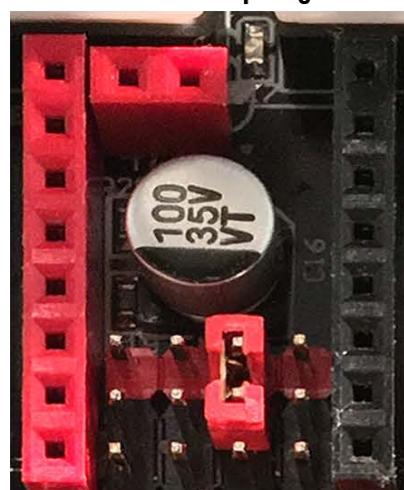
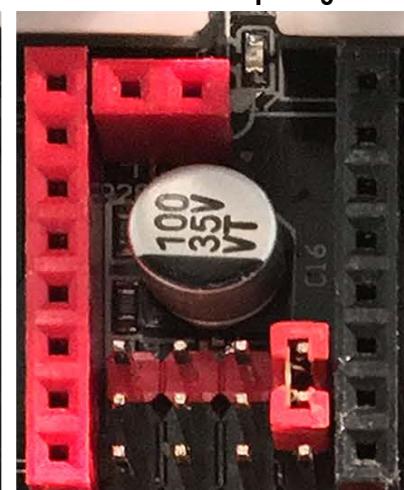
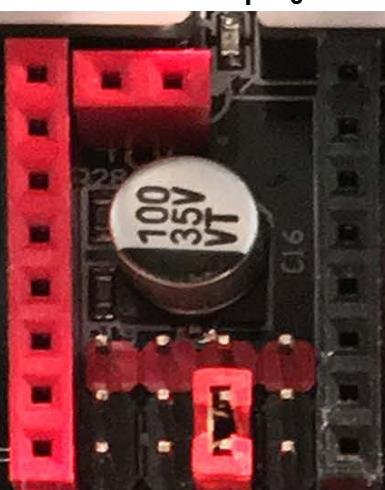
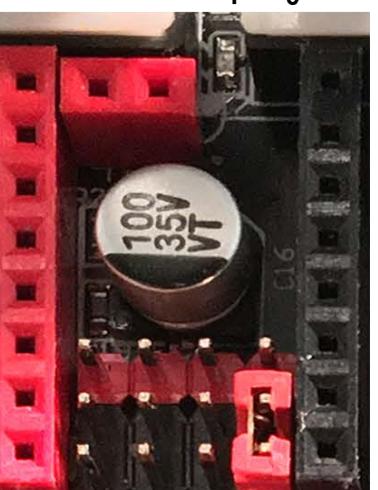
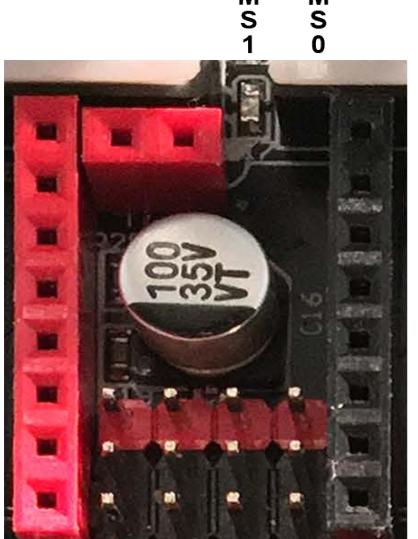
SKR PRO V1.1 LEGEND for Tri State Stepper Drivers

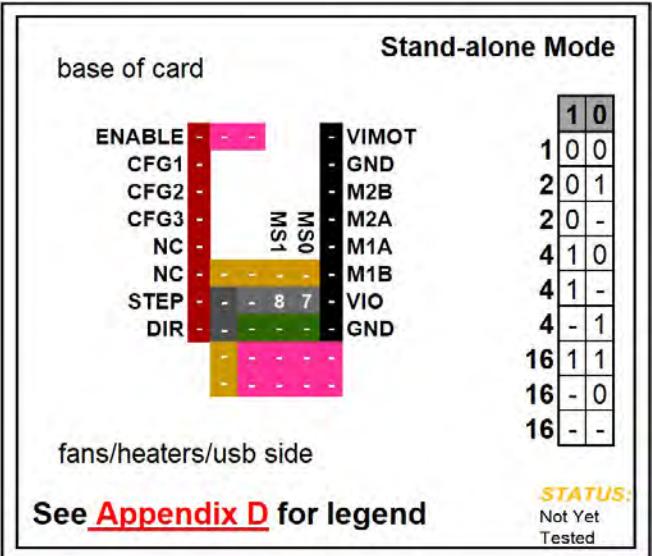
PART 2

OPEN ➡ **No Jumper**



Examples of using OPEN:

OPEN**OPEN****Low****OPEN****Low****OPEN****OPEN****High****OPEN**



MKS TMC2100

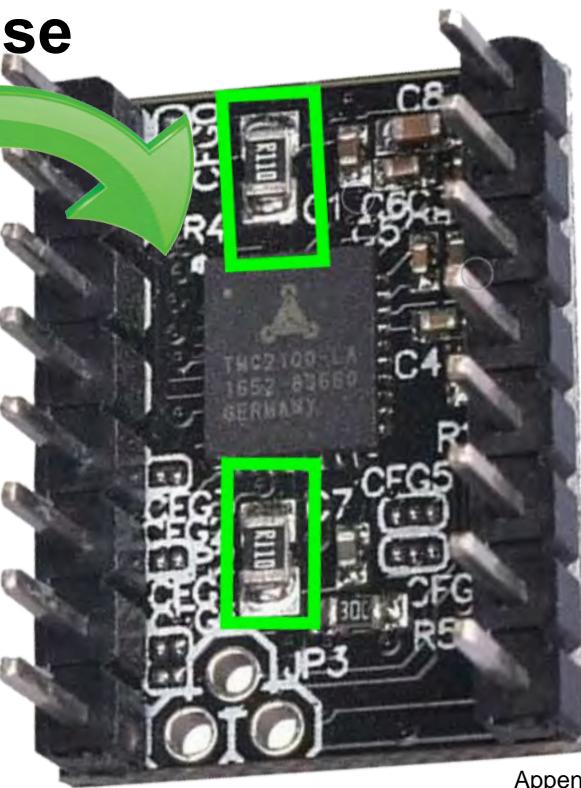
Stand-alone Mode

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

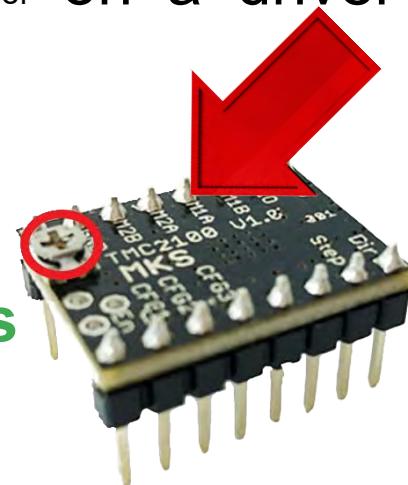
NOTE: Use the potentiometer (POT) on the top of the board (or use the board's " V_{ref} Test point" to set your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

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>

Stand-alone Mode

Stand-alone
Mode

STEP

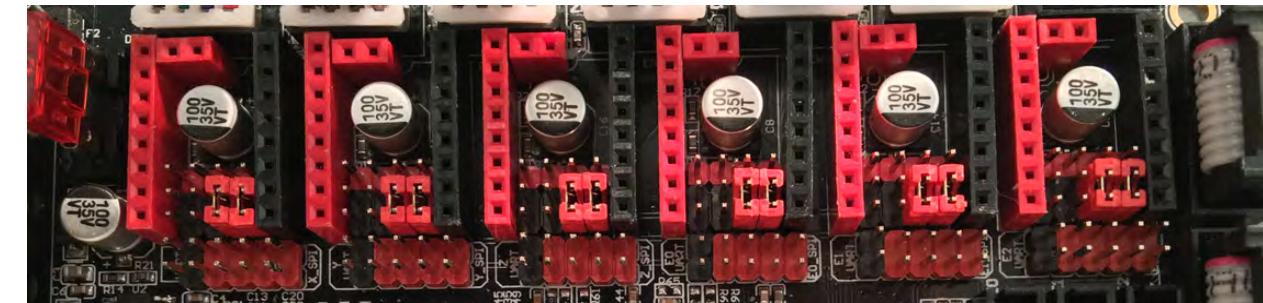
Interpolation:
none

SpreadCycle

ENABLE	-	-	VIMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	-	MS1	M1A
NC	-	MS0	M1B
STEP	-	8 7	VIO
DIR	-	8 7	GND

MKS TMC2100

Stand-alone Mode



See [Appendix D](#) for legend

Stand-alone
Mode

1 / 2

Interpolation:
none

SpreadCycle

ENABLE	-	-	VIMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	-	MS1	M1A
NC	-	MS0	M1B
STEP	-	8 7	VIO
DIR	-	8 -	GND



See [Appendix D](#) for legend

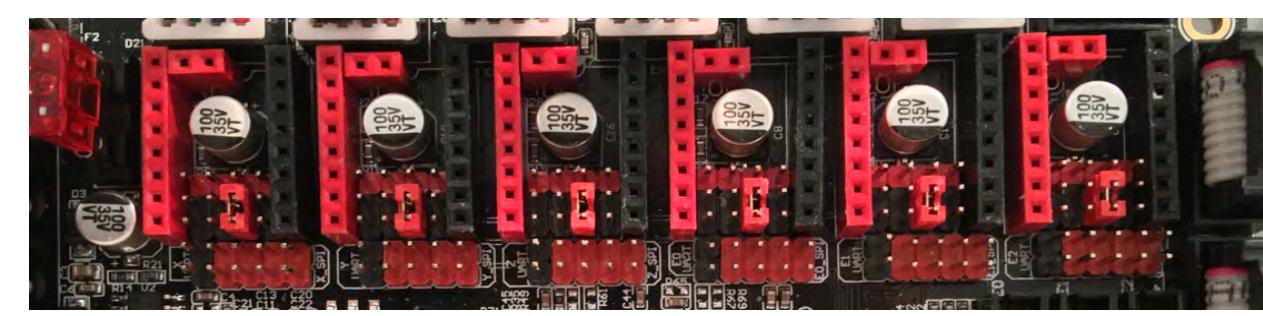
Stand-alone
Mode

1 / 2

Interpolation:
1 / 256

SpreadCycle

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



See [Appendix D](#) for legend

Stand-alone Mode

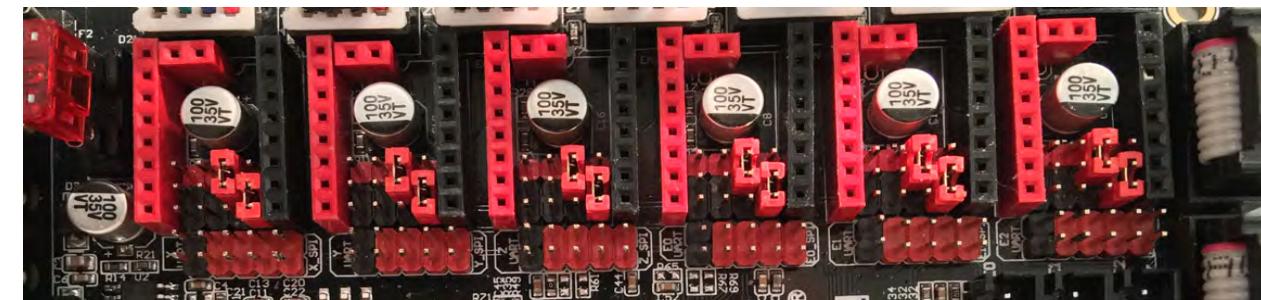
Stand-alone
Mode

1 / 4

Interpolation:
none

SpreadCycle

ENABLE	-	-	-	VIMOT
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
NC	-	-	MS0	M1A
NC	-	-	8	M1B
STEP	-	-	8	VIO
DIR	-	-	7	GND



See [Appendix D](#) for legend

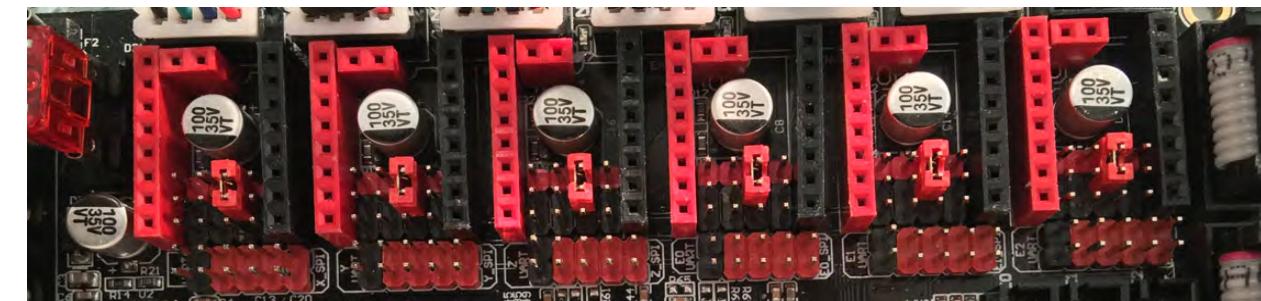
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

SpreadCycle

ENABLE	-	-	-	VIMOT
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
NC	-	-	MS0	M1A
NC	-	-	8	M1B
STEP	-	-	8	VIO
DIR	-	-	7	GND



See [Appendix D](#) for legend

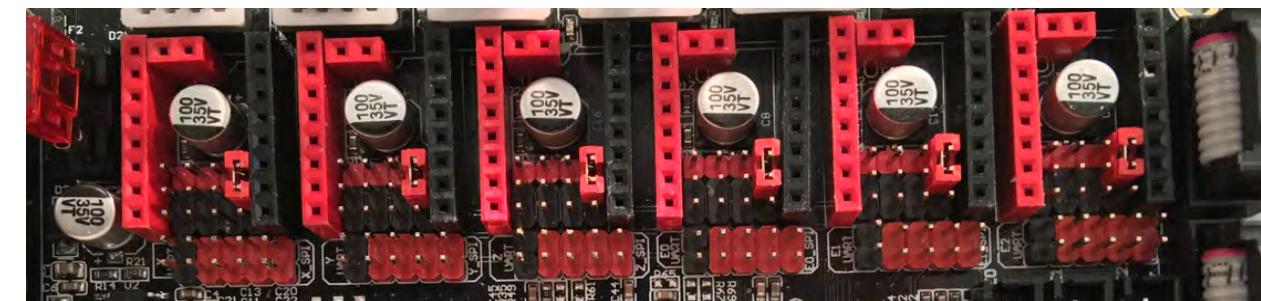
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

StealthChop

ENABLE	-	-	-	VIMOT
CFG1	-	-	-	GND
CFG2	-	-	-	M2B
CFG3	-	-	-	M2A
NC	-	-	MS0	M1A
NC	-	-	7	M1B
STEP	-	-	7	VIO
DIR	-	-	-	GND



See [Appendix D](#) for legend

Stand-alone Mode

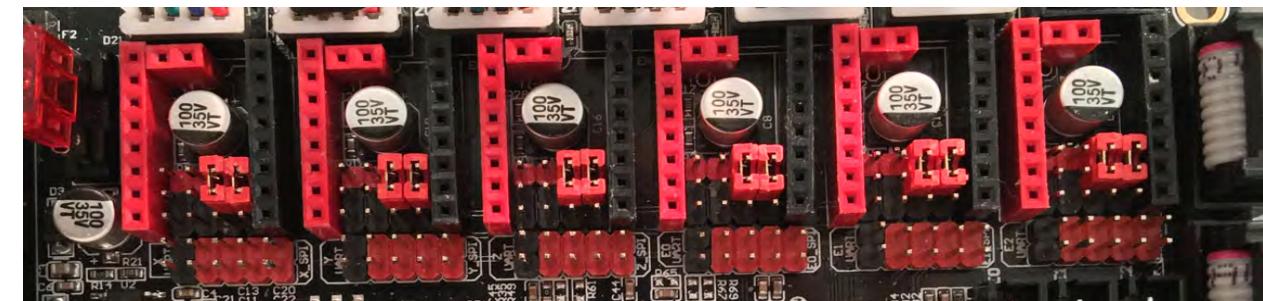
Stand-alone
Mode

1 / 16

Interpolation:
none

SpreadCycle

ENABLE	-	-	VIMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	-	MS1	M1A
NC	8	7	M1B
STEP	-	8	VIO
DIR	-	-	GND



See [Appendix D](#) for legend

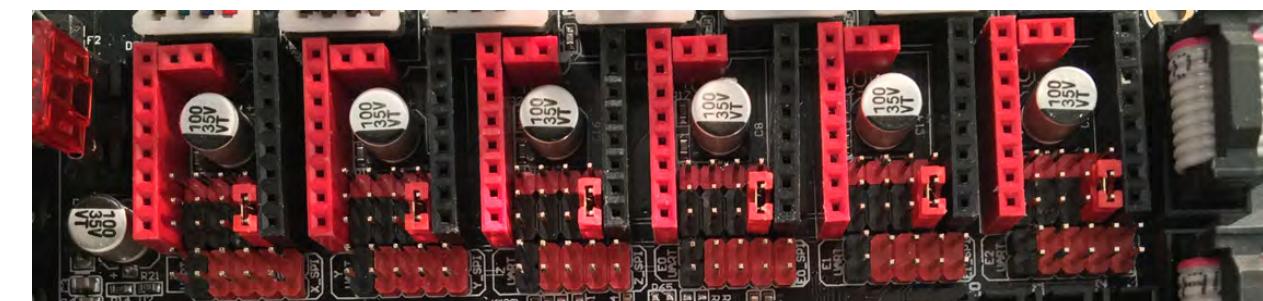
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

SpreadCycle

ENABLE	-	-	VIMOT
CFG1	-	-	GND
CFG2	-	-	M2B
CFG3	-	-	M2A
NC	-	MS1	M1A
NC	7	7	M1B
STEP	-	-	VIO
DIR	-	7	GND



See [Appendix D](#) for legend

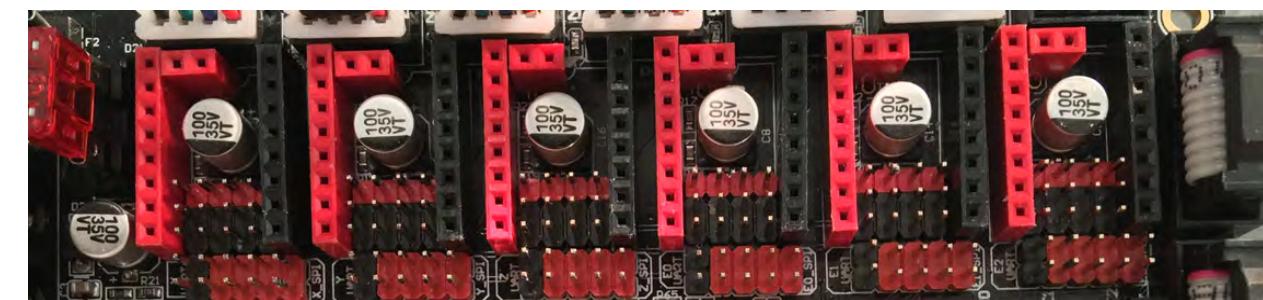
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

StealthChop

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



See [Appendix D](#) for legend

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

Marlin > Configuration.h > E0_DRIVER_TYPE

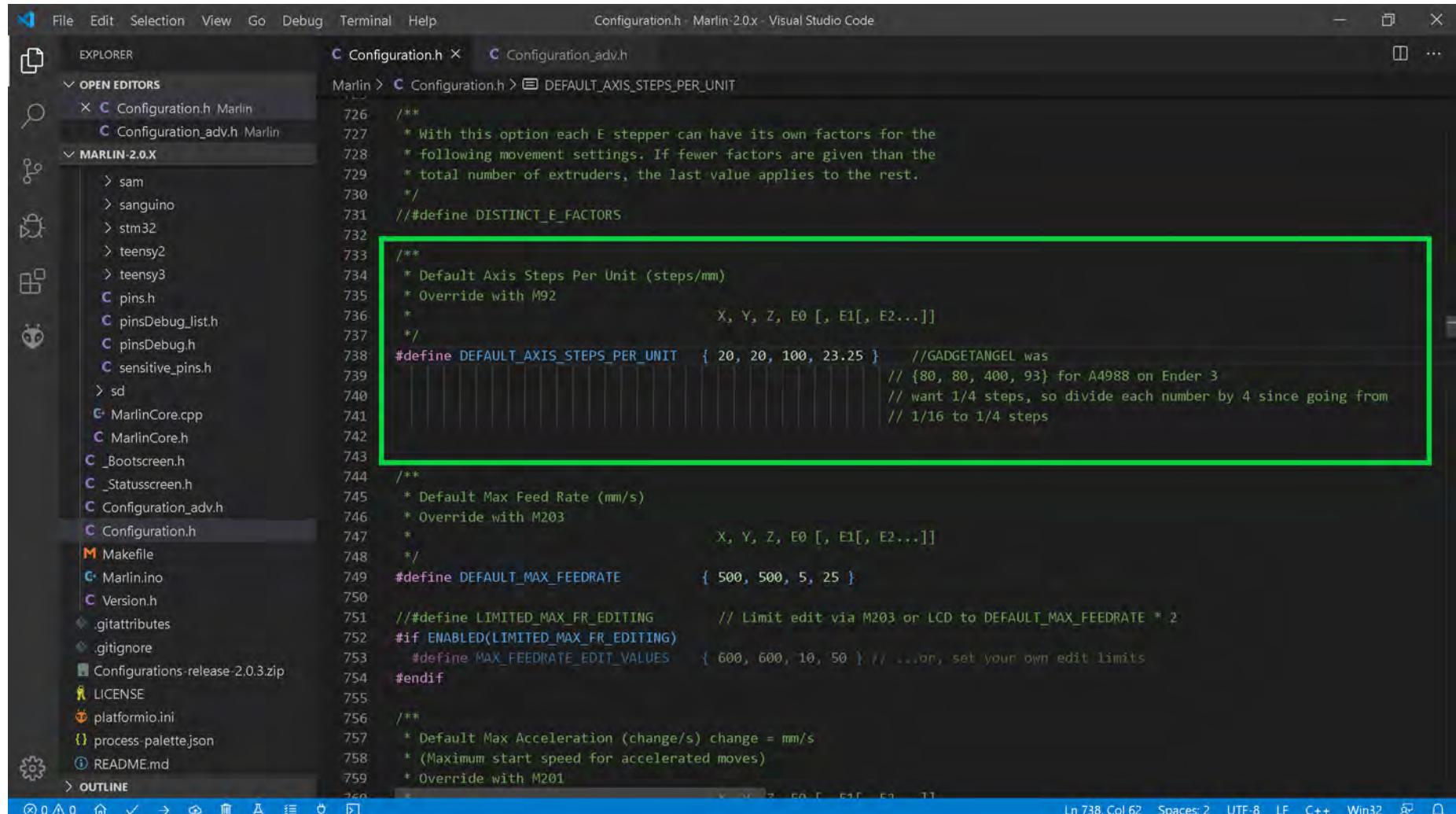
```
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * 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', 'TMC2660']
676 */
677 #define X_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2100 //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 E2_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2100 //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.
```

Ln 686, Col 88 Spaces: 2 UTF-8 LF C++ Win32 1

- 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 'Configuration.h' file open. The code editor displays the following snippet of C++ code:

```

    /**
     * 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
 */

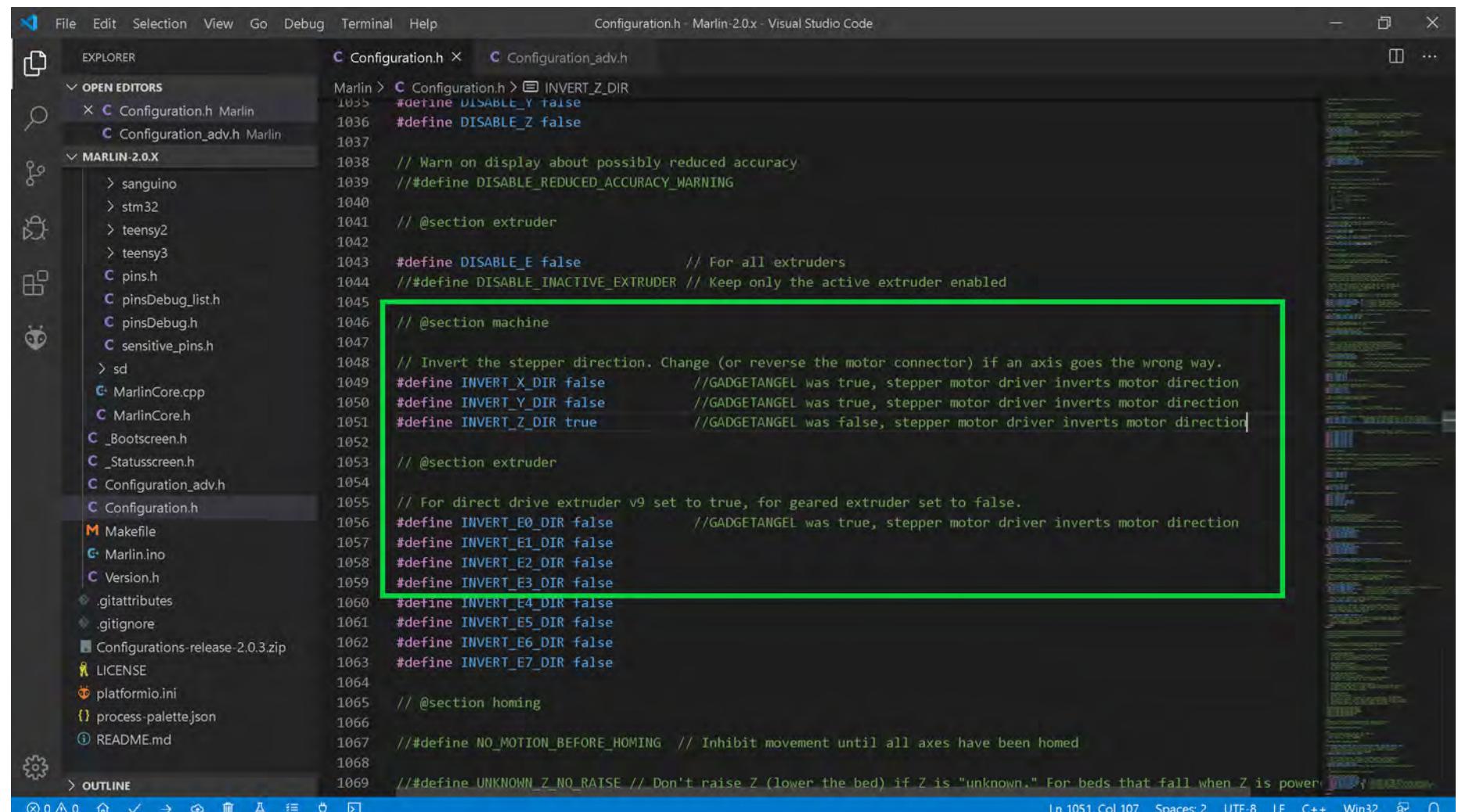
```

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

- 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

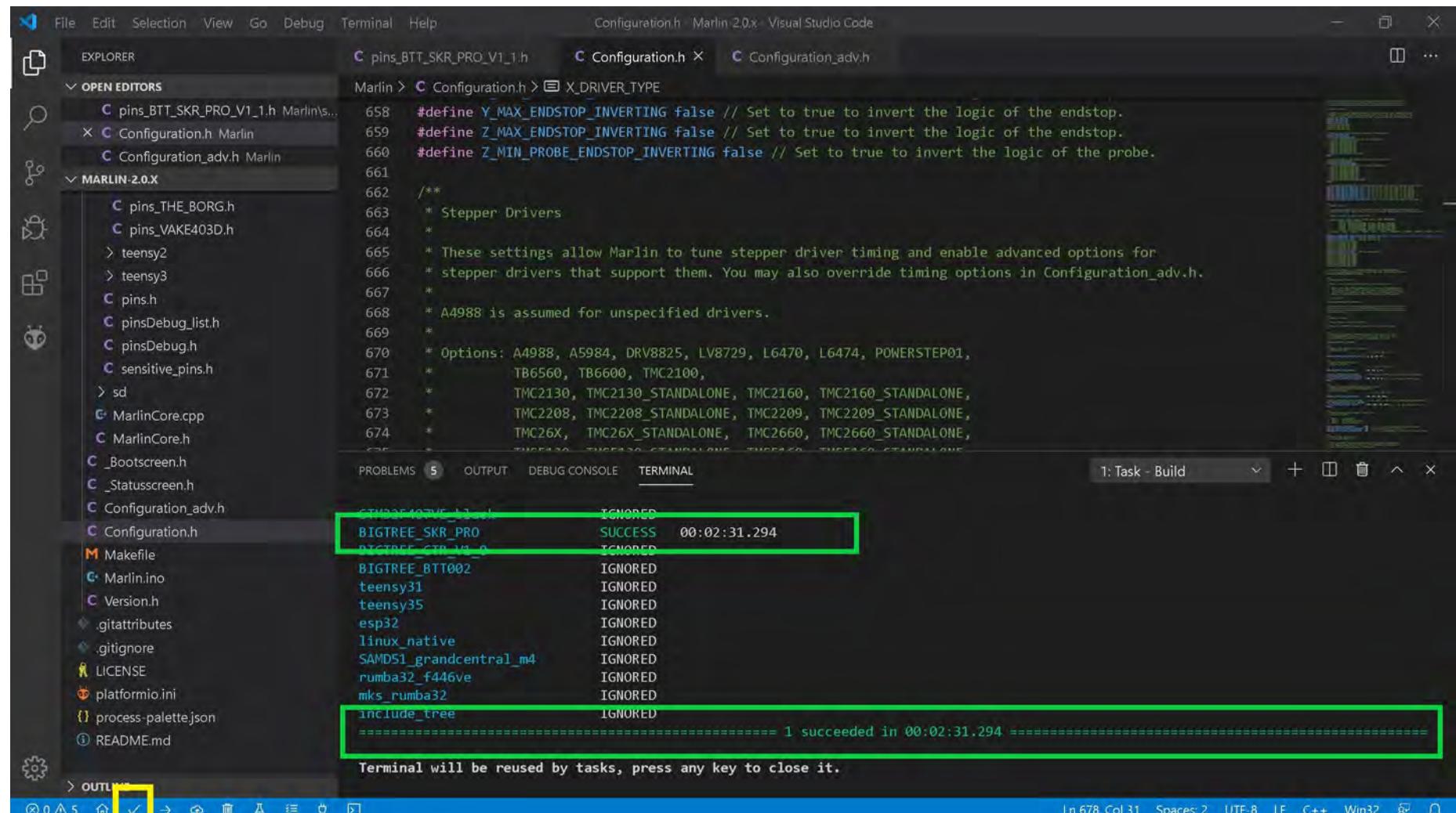
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that for TMC2100 drivers, the value should be changed from `true` to `false`.

- 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.



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

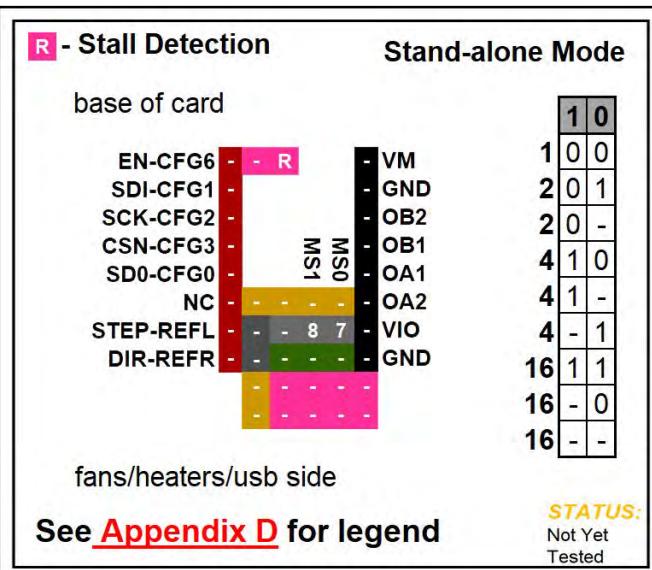
```

TMC2100_V1_1.h    IGNORED
BIGTREE_SKR_PRO   SUCCESS  00:02:31.294
BIGTREE_CTP_V1_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 =====

```

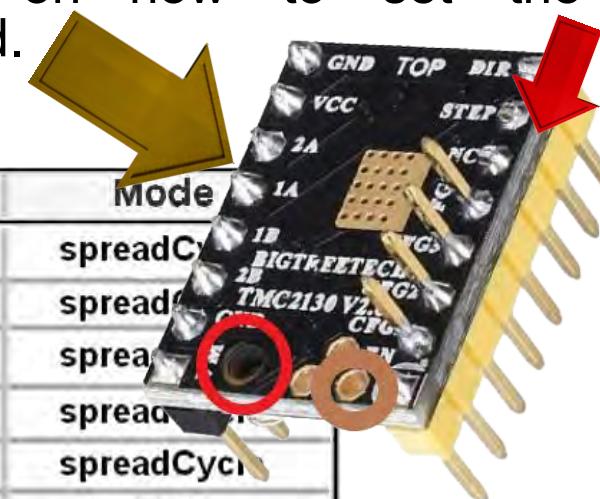
A yellow box highlights the checkmark icon in the bottom-left corner of the terminal interface, indicating the build was successful. A green box highlights the terminal output showing the build completed successfully in 00:02:31.294.

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

**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	Low	Low	1	NONE	spreadCycle
	Low	High	1 / 2	NONE	spreadCycle
	Low	OPEN	1 / 2	1 / 256	spreadCycle
	High	Low	1 / 4	NONE	spreadCycle
	High	OPEN	1 / 4	1 / 256	spreadCycle
	OPEN	High	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	OPEN	Low	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop

Driving Current Calculation Formula

$$R_S \text{ (Typical Sense Resistor)} = 0.11\Omega$$

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

See Appendix B #2. Use 50% to 90% as shown below:

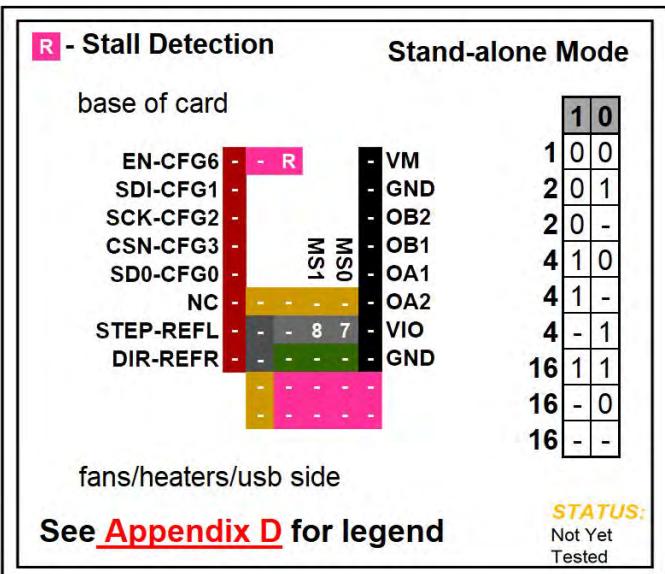
$$I_{MAX} = I_{MAX} * 0.90$$

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

See Appendix B #2. Use 50% to 90% as shown below:

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

- See next page for the legend that belongs to the above chart.



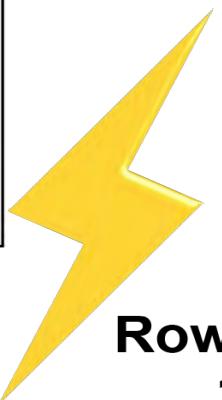
BIQU TMC2130

Stand-alone Mode

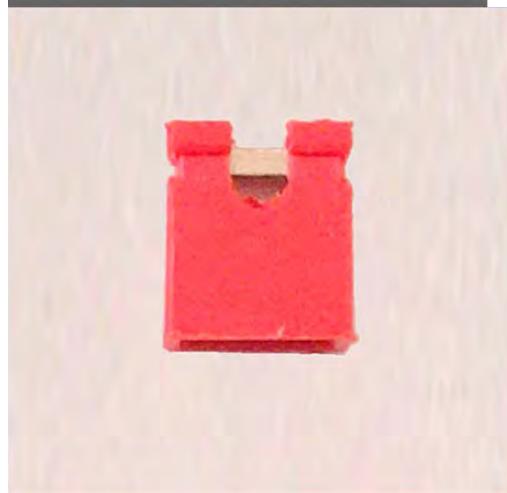
SKR PRO V1.1 LEGEND for Tri State Stepper Drivers

PART 1

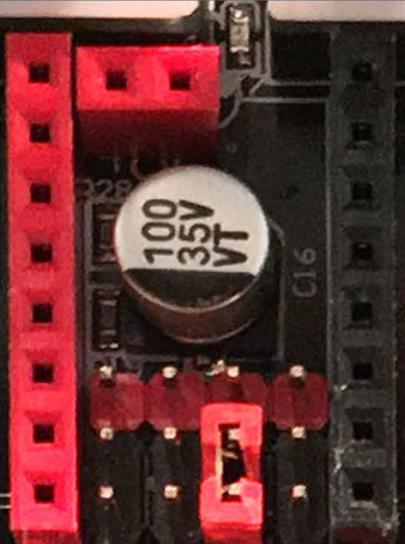
Low ➡ set Jumper between rows
2 and 3



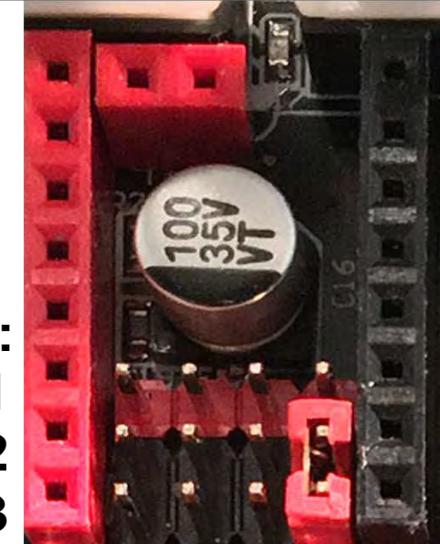
This is a
Jumper:



Row:
1
2
3

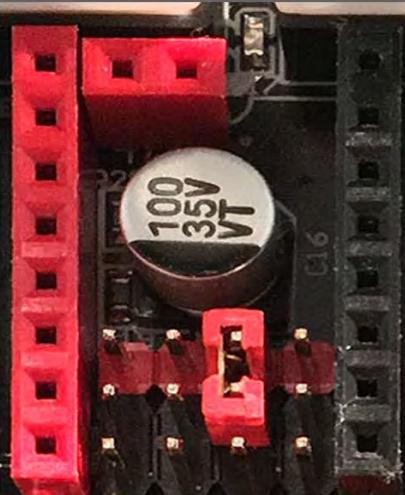


Row:
1
2
3

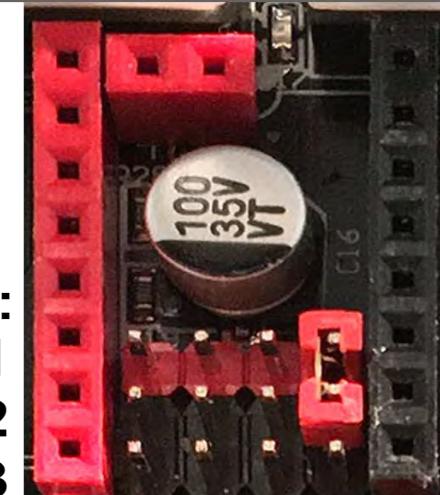


High ➡ set Jumper between rows
1 and 2

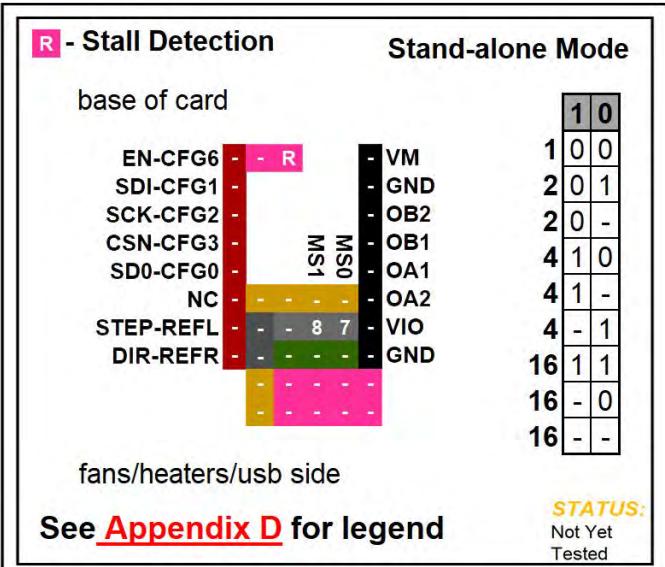
Row:
1
2
3



Row:
1
2
3



- Go to the next page.



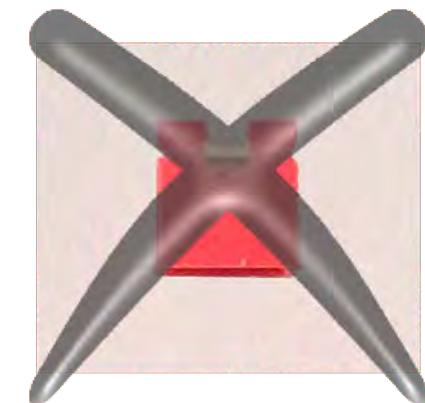
BIGU TMC2130

Stand-alone Mode

SKR PRO V1.1 LEGEND for Tri State Stepper Drivers

PART 2

OPEN ➔ No Jumper



Examples of using OPEN:

OPEN

OPEN

Low

OPEN

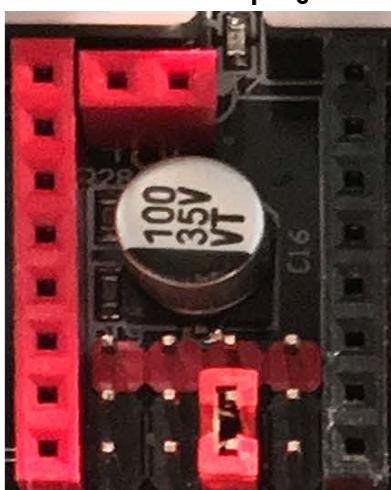
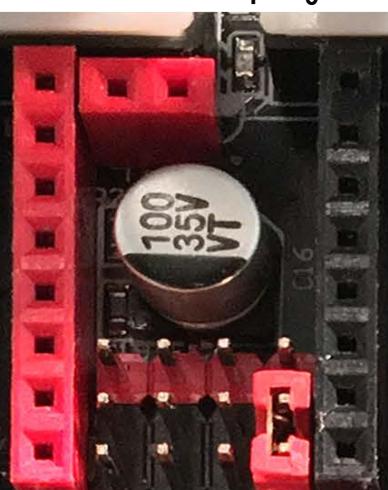
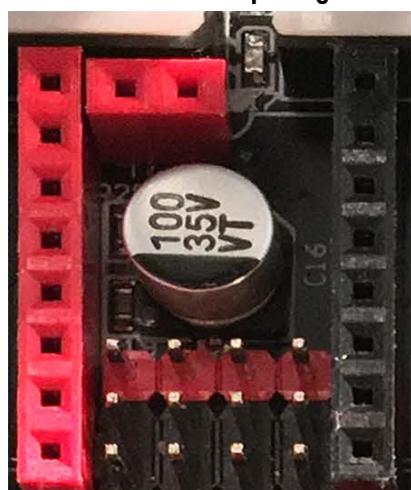
Low

OPEN

M
S
1
0

M
S
1
0

M
S
1
0



OPEN

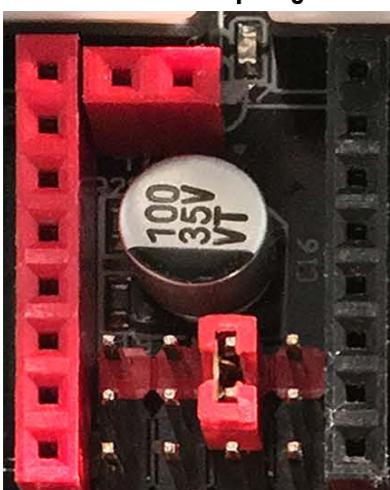
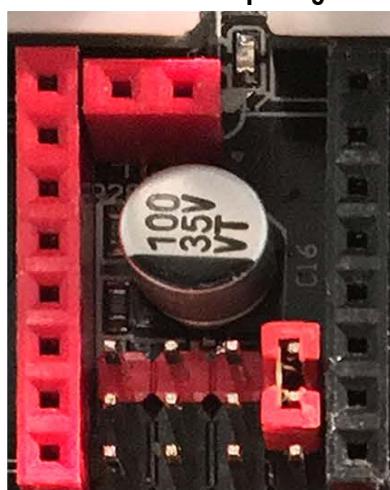
High

High

OPEN

M
S
1
0

M
S
1
0



R - Stall Detection		Stand-alone Mode
base of card		
EN-CFG6	- R	VM
SDI-CFG1		GND
SCK-CFG2		OB2
CSN-CFG3		OB1
SD0-CFG0	ISW	OA1
NC	OSW	OA2
STEP-REFL		VIO
DIR-REFR		GND
fans/heaters/usb side		
See Appendix D for legend		STATUS: Not Yet Tested

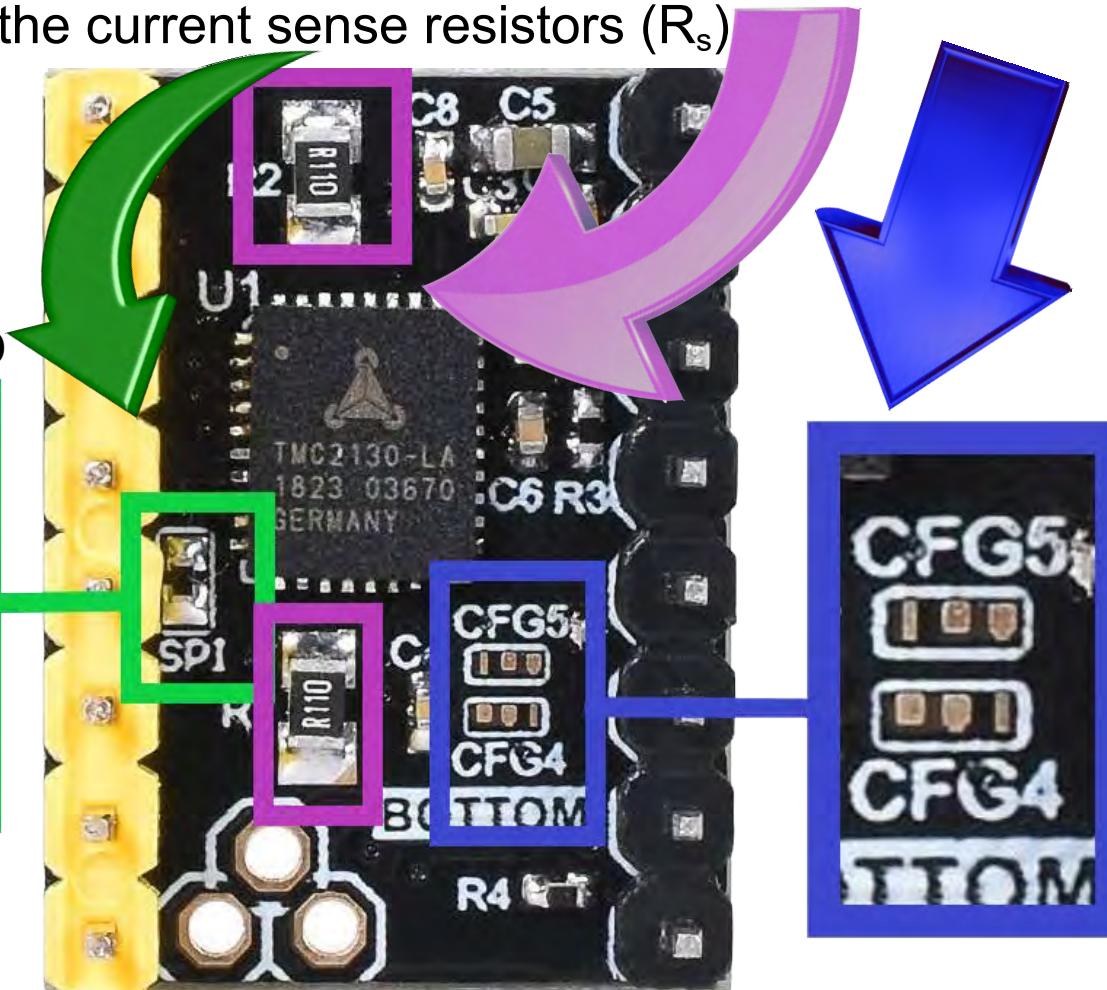
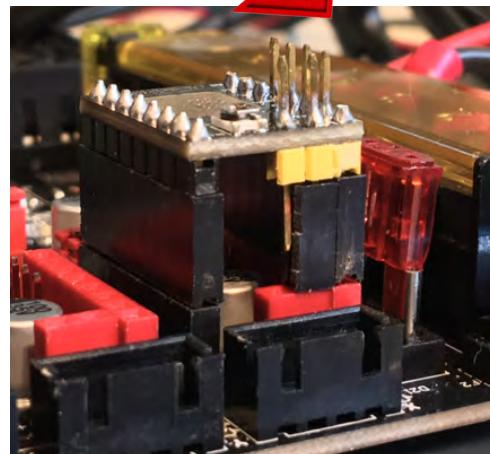
BIQU TMC2130

Stand-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.

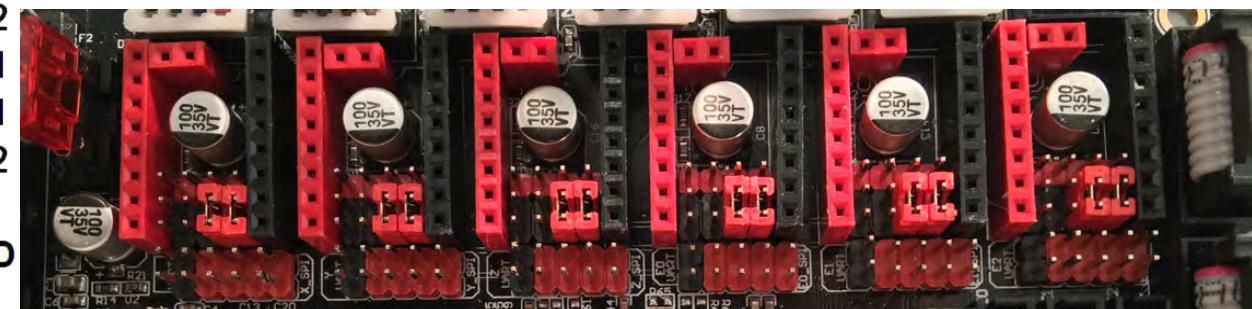


BIQU TMC2130

Stand-alone Mode

Stand-alone Mode

	EN-CFG6	-	R	-	VM
Stand-alone Mode	SDI-CFG1	-		-	GND
	SCK-CFG2	-		-	OB2
	CSN-CFG3	-		-	OB1
	SD0-CFG0	-	MS1	-	OA1
	NC	-	MS0	-	OA2
Interpolation:	STEP-REFL	-	-	8 7	VIO
none	DIR-REFR	-	-	8 7	GND
SpreadCycle		-	-	-	



See [Appendix D](#) for legend

Stand-alone Mode

1 / 2

Interpolation:

none

SpreadCycle

Stand-alone Mode

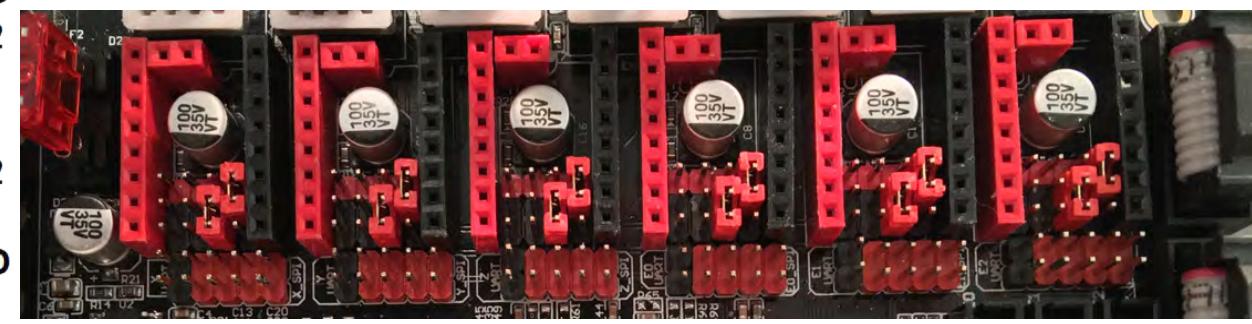
1 / 2

Interpolation:

1 / 256

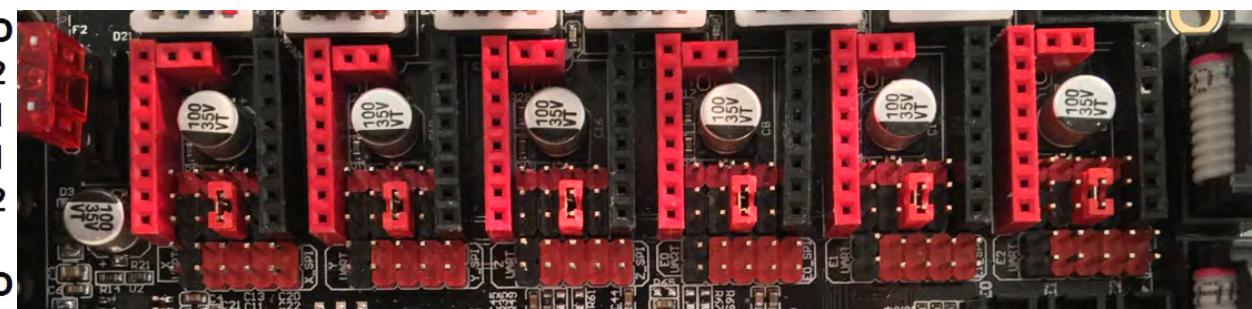
SpreadCycle

	EN-CFG6	-	R	-	VM
Stand-alone Mode	SDI-CFG1	-		-	GND
	SCK-CFG2	-		-	OB2
	CSN-CFG3	-		-	OB1
	SD0-CFG0	-	MS1	-	OA1
	NC	-	MS0	-	OA2
Interpolation:	STEP-REFL	-	-	7	VIO
none	DIR-REFR	-	-	8 7	GND
SpreadCycle		-	-	-	



See [Appendix D](#) for legend

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



See [Appendix D](#) for legend

BIQU TMC2130

Stand-alone Mode

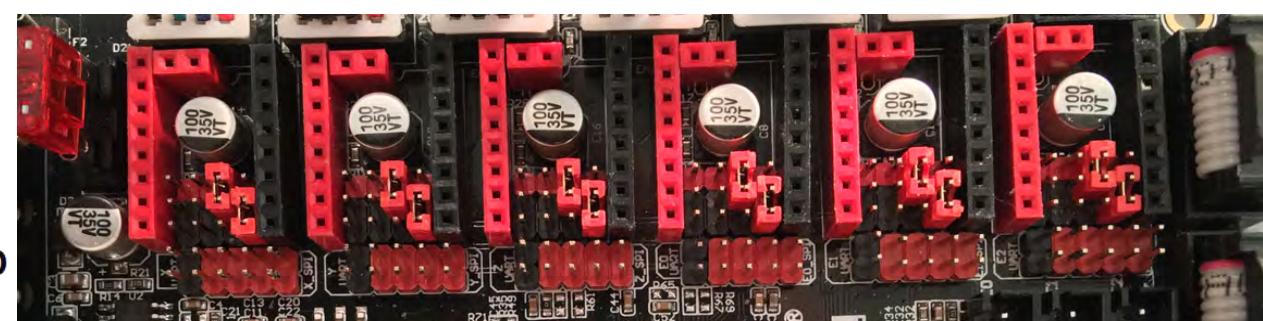
Stand-alone Mode

1 / 4

Interpolation:
none

SpreadCycle

EN-CFG6	-	R	-	VM
SDI-CFG1	-	-	-	GND
SCK-CFG2	-	-	-	OB2
CSN-CFG3	-	-	-	OB1
SD0-CFG0	-	-	MS0	OA1
NC	-	8	-	OA2
STEP-REFL	-	-	8	VIO
DIR-REFR	-	-	7	GND



See [Appendix D](#) for legend

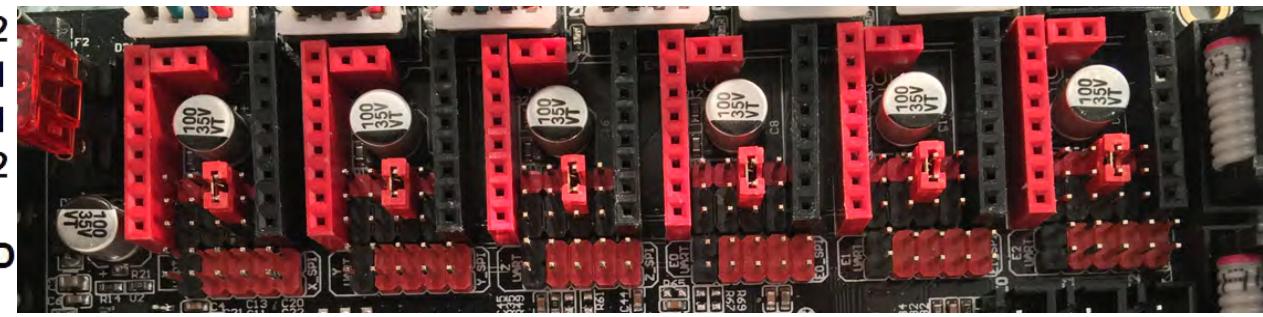
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

SpreadCycle

EN-CFG6	-	R	-	VM
SDI-CFG1	-	-	-	GND
SCK-CFG2	-	-	-	OB2
CSN-CFG3	-	-	-	OB1
SD0-CFG0	-	-	MS0	OA1
NC	-	8	-	OA2
STEP-REFL	-	-	8	VIO
DIR-REFR	-	-	7	GND



See [Appendix D](#) for legend

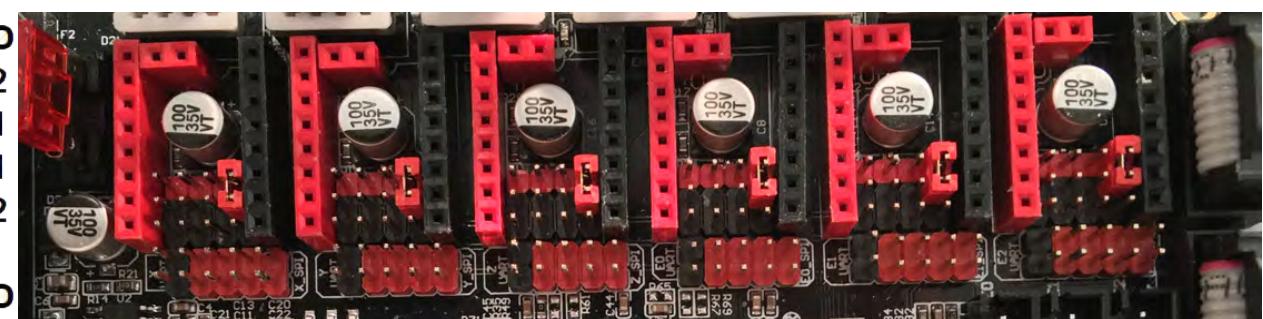
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

StealthChop

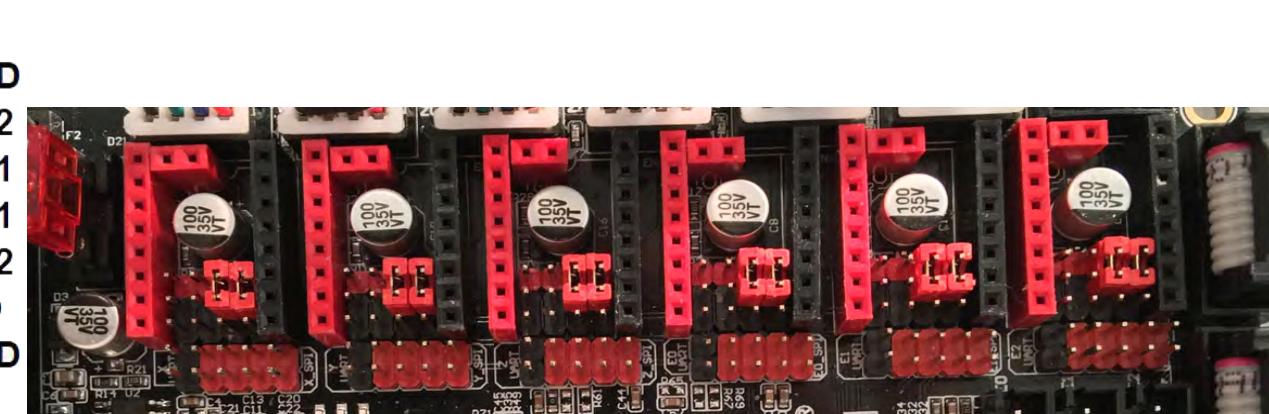
EN-CFG6	-	R	-	VM
SDI-CFG1	-	-	-	GND
SCK-CFG2	-	-	-	OB2
CSN-CFG3	-	-	-	OB1
SD0-CFG0	-	-	MS0	OA1
NC	-	7	-	OA2
STEP-REFL	-	-	7	VIO
DIR-REFR	-	-	-	GND



See [Appendix D](#) for legend

BIQU TMC2130Stand-alone Mode**Stand-alone Mode****1 / 16**

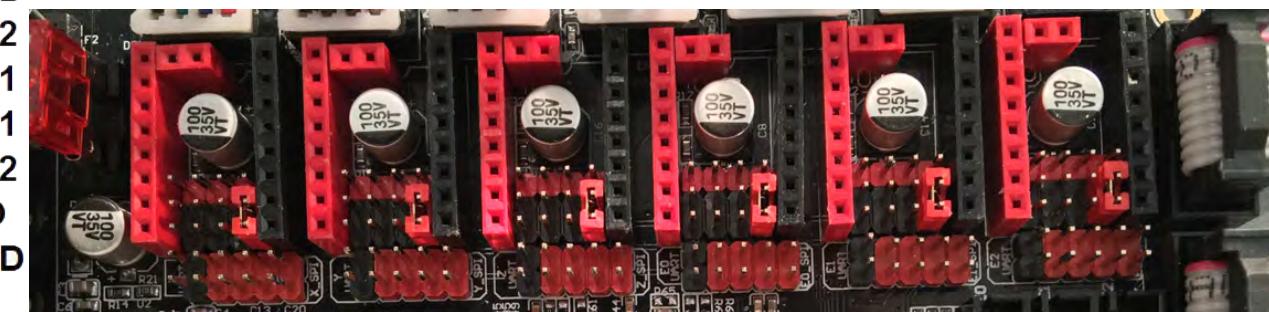
Stand-alone Mode	EN-CFG6	- R	VM
	SDI-CFG1	-	GND
	SCK-CFG2	-	OB2
	CSN-CFG3	-	OB1
	SD0-CFG0	-	OA1
	NC	8 7	OA2
Interpolation:	STEP-REFL	-	VIO
none	DIR-REFR	-	GND
SpreadCycle			

See [Appendix D](#) for legend

Stand-alone Mode

1 / 16

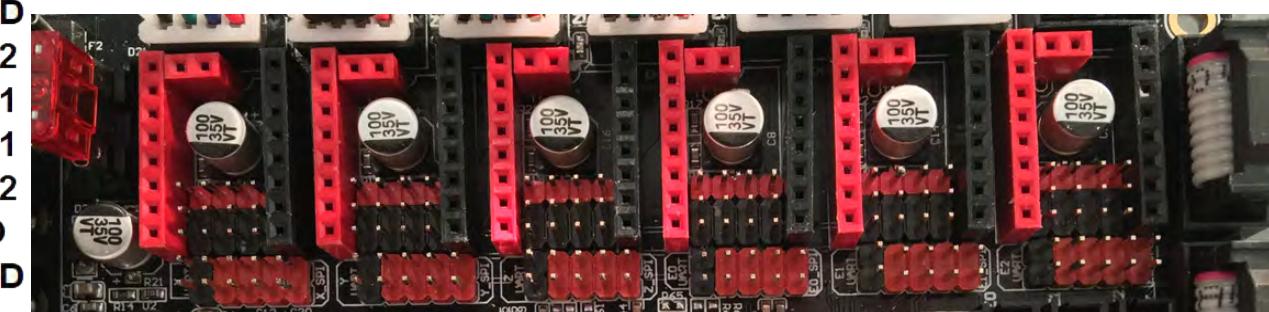
Interpolation:	STEP-REFL	-	VM
1 / 256	DIR-REFR	-	GND
SpreadCycle			

See [Appendix D](#) for legend

Stand-alone Mode

1 / 16

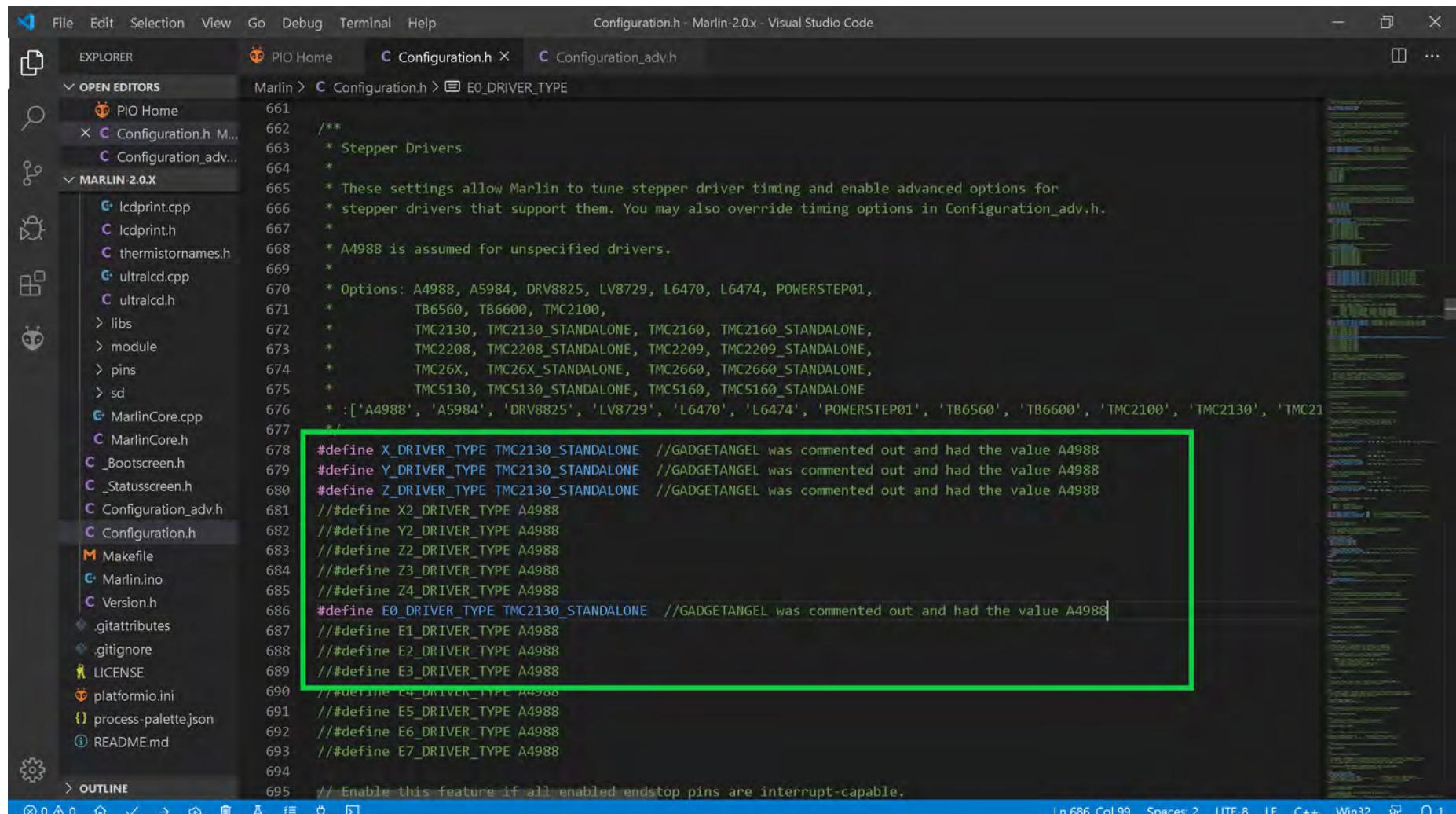
Interpolation:	STEP-REFL	-	VM
1 / 256	DIR-REFR	-	GND
StealthChop			

See [Appendix D](#) for legend

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.

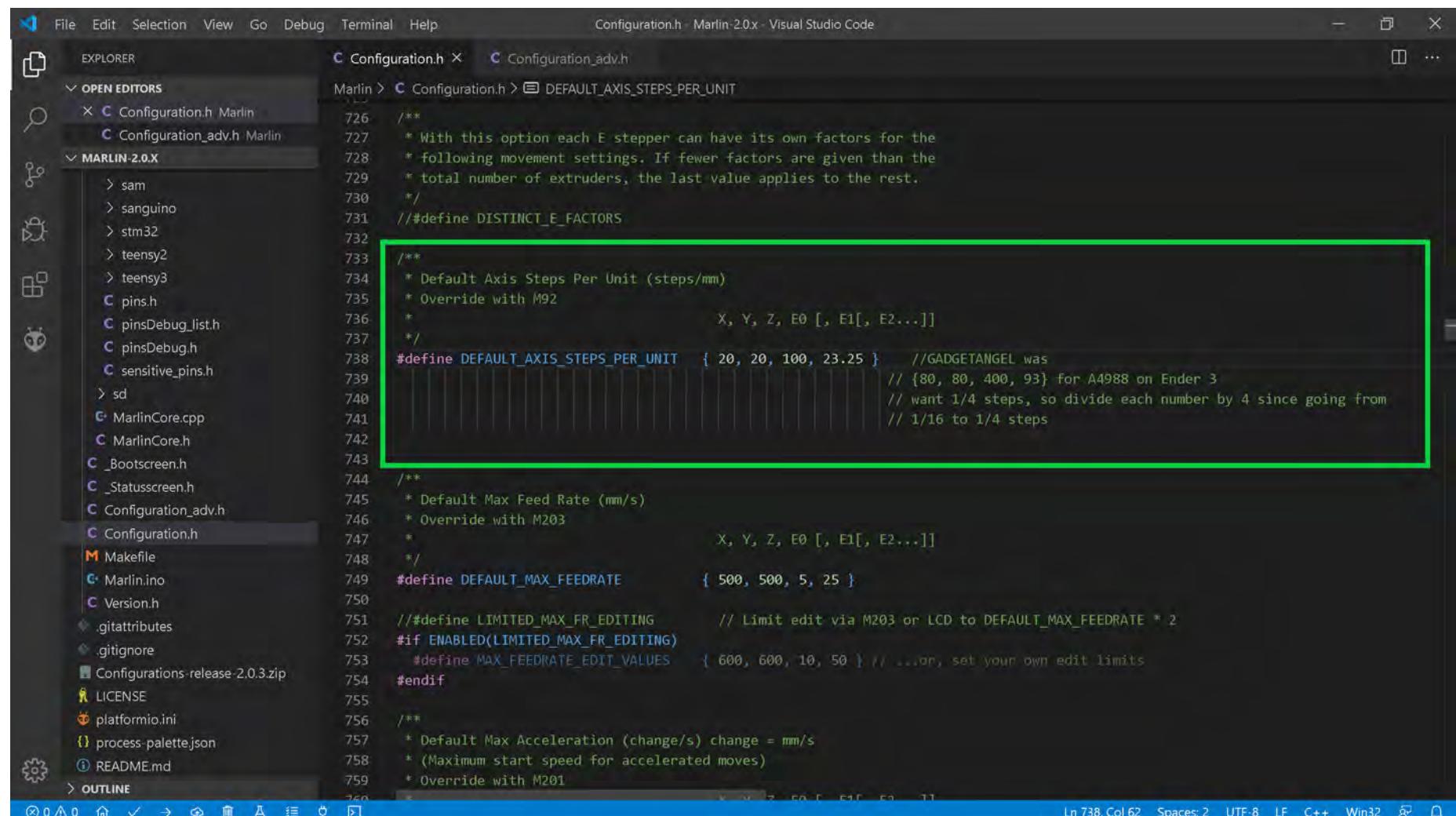
```

A green rectangular box highlights the driver type definitions (lines 678-693), specifically the lines starting with `#define`. The text within these lines is preceded by two forward slashes (//), indicating they are currently commented out. The status bar at the bottom of the code editor shows the current line (Ln 686) and column (Col 99).

- 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 file `Configuration.h` open. The code editor displays the Marlin configuration files. A green rectangular box highlights the following code snippet in the `Configuration.h` file:

```

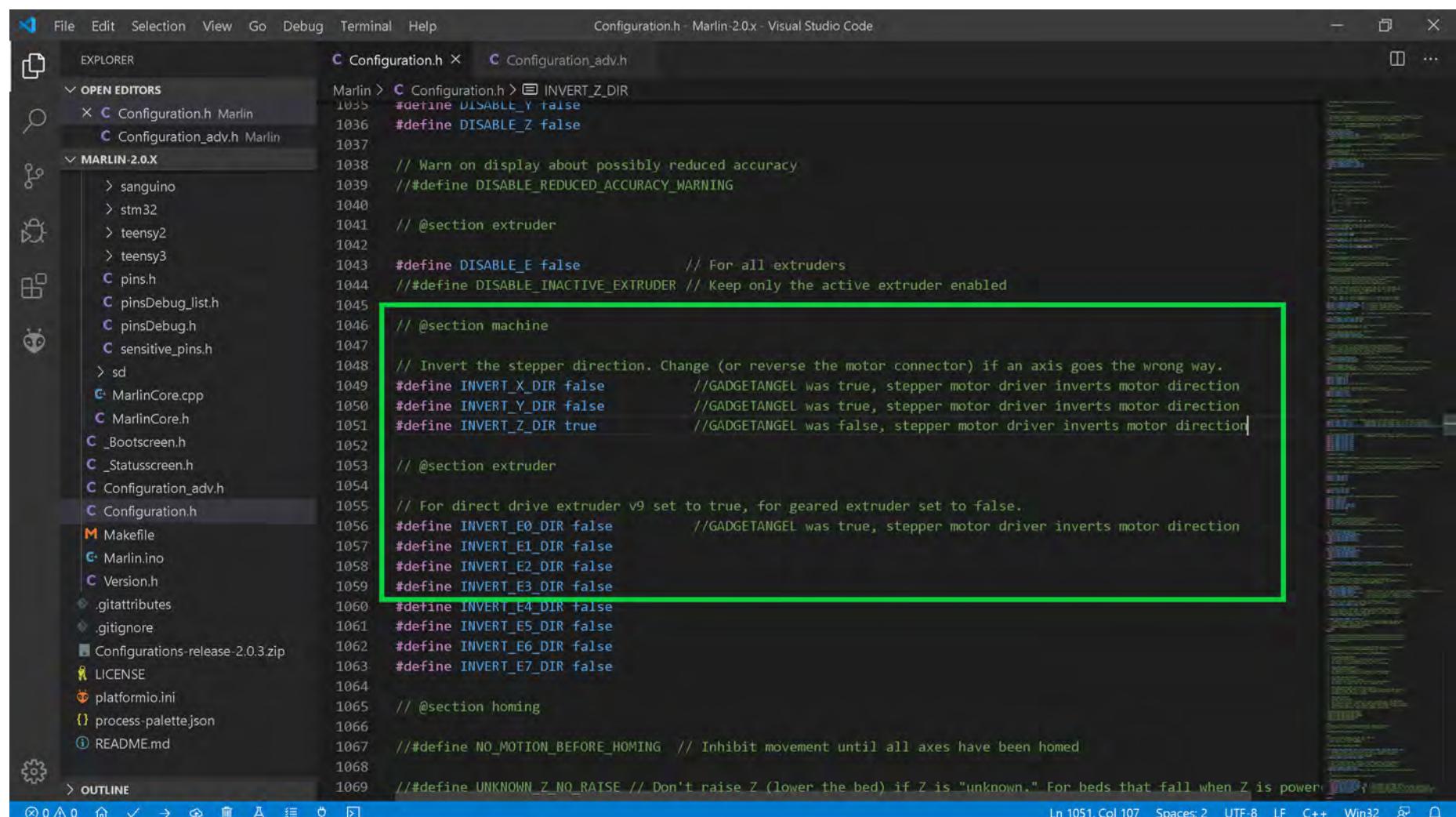
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per 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



```

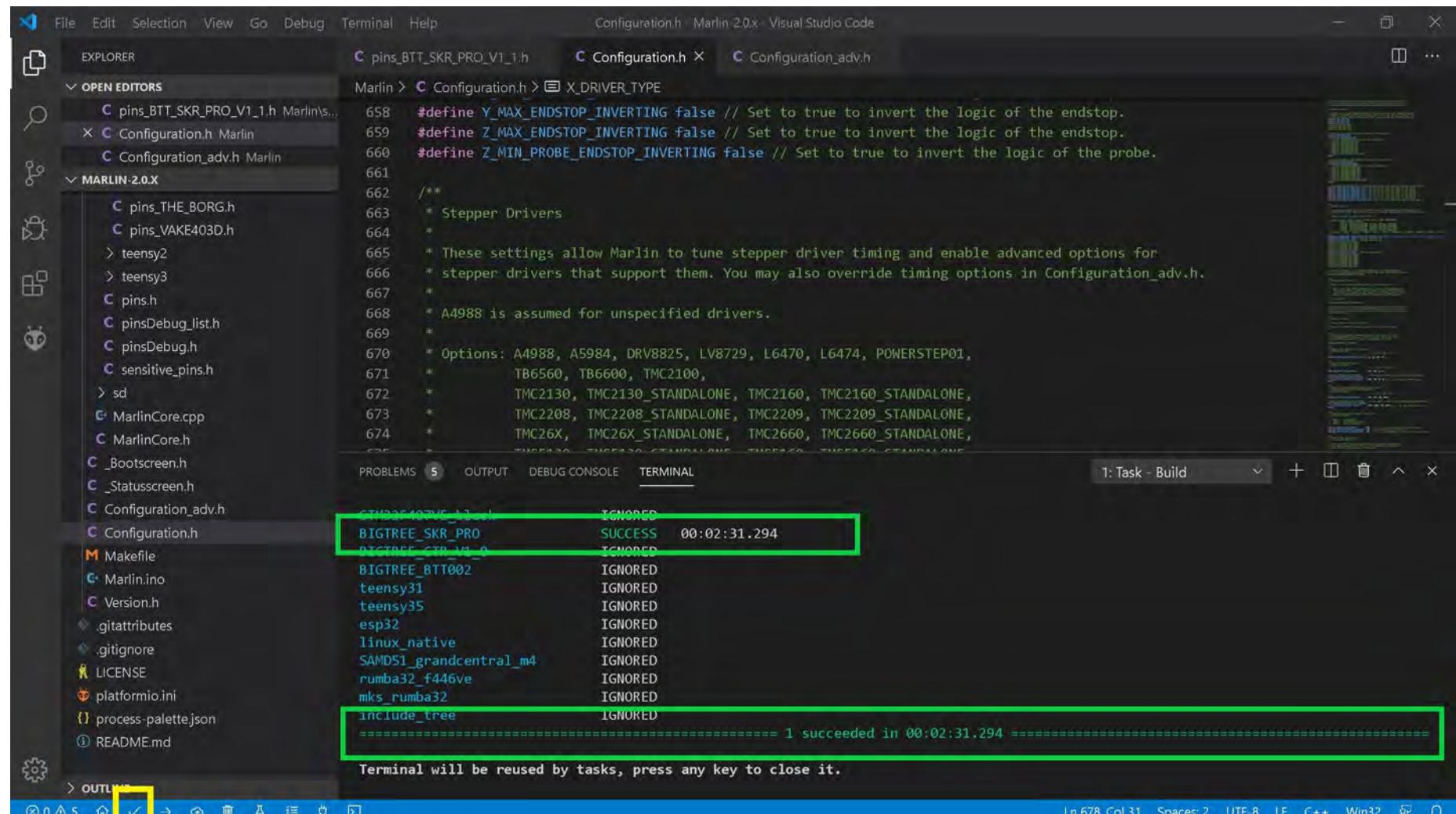
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  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
  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
  // 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

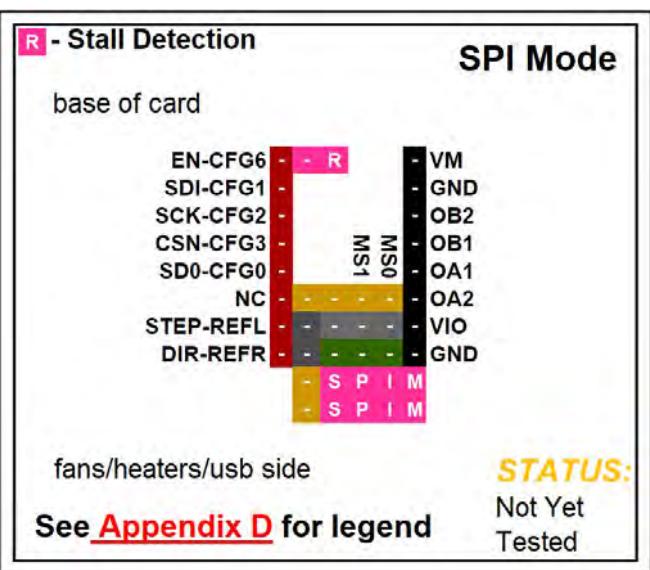
- The end of Marlin setup for BIQU TMC2130 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



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

- File Explorer:** Shows files like pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and MarlinCore.cpp.
- Editor:** Displays the Configuration.h file with code related to stepper drivers and endstops.
- Terminal:** Shows the build command and its output. The output is highlighted with a green box, showing successful builds for various boards: BIGTREE_SKR_PRO, BIGTREE_BT002, teensy31, teensy35, esp32, linux_native, SAMD51_grandcentral_m4, rumba32_f446ve, mks_rumba32, and include_tree. The message "1 succeeded in 00:02:31.294" is at the bottom.
- Status Bar:** Shows the terminal will be reused by tasks, and the current session details: Ln 678, Col 31, Spaces: 2, UTF-8, LF, C++, Win32.

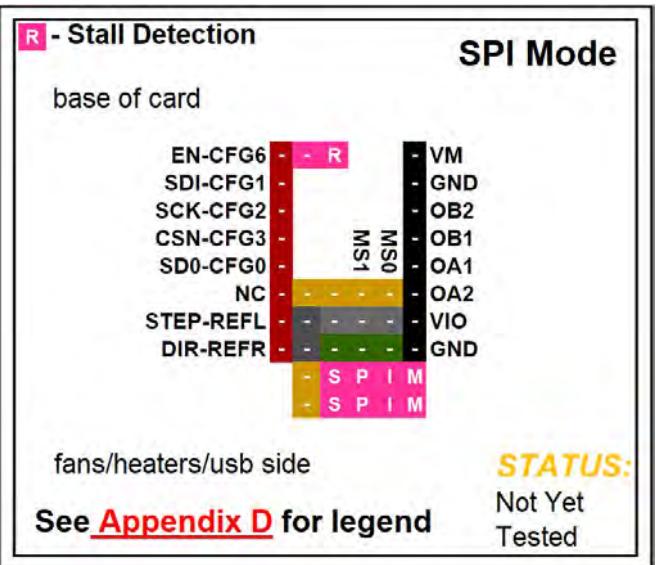
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

**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.

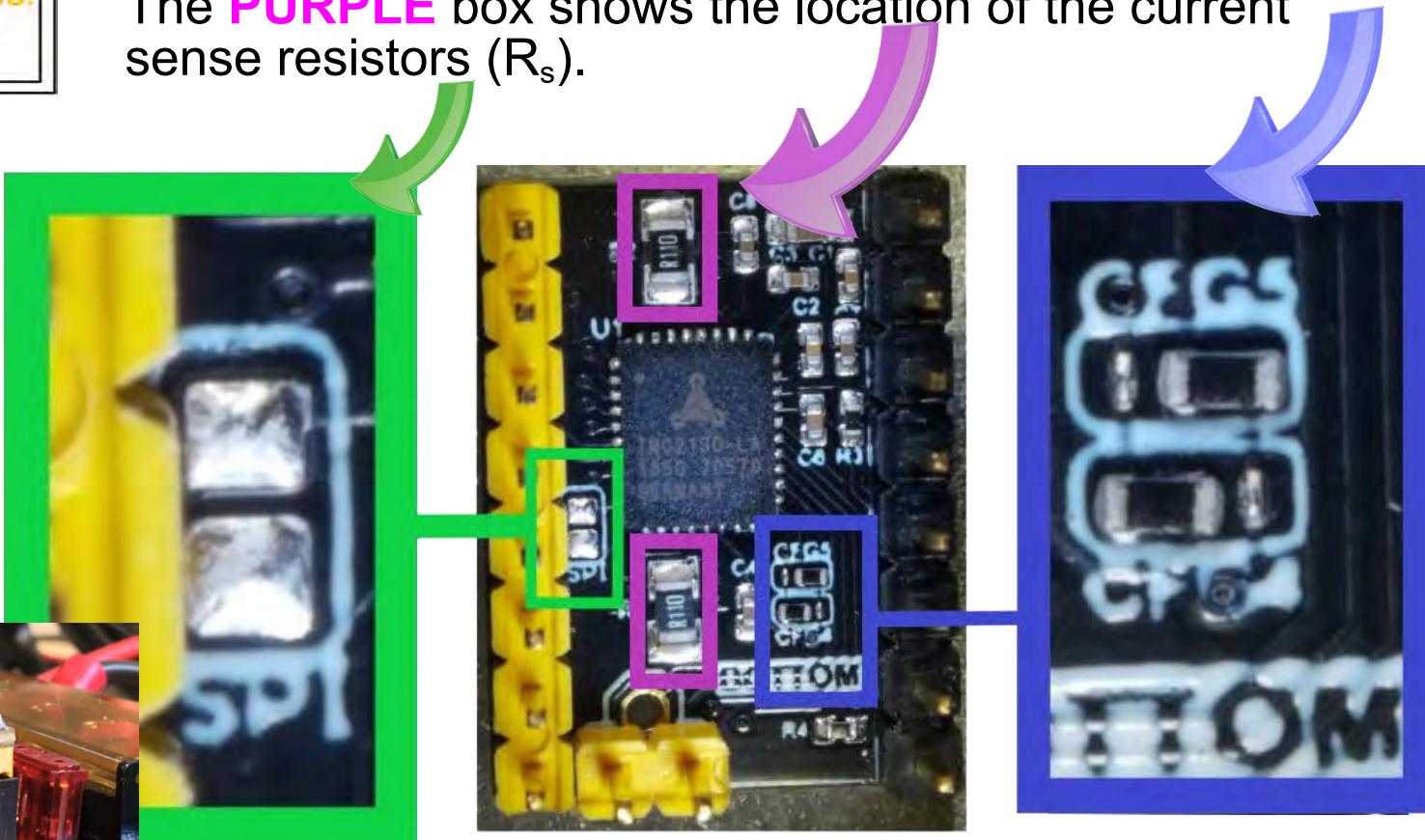
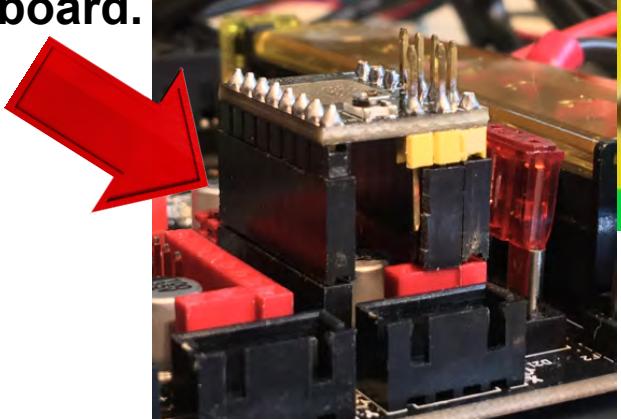
See the next page for further information.

Driver Chip  TMC2130 SPI Mode Maximum 256 Subdivision 46V DC 2.5A (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 #2. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	$V_{ref} = I_{MAX}$ See Appendix B #2. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

**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.



SPI Mode**R - Stall Detection****BIQU TMC2130**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

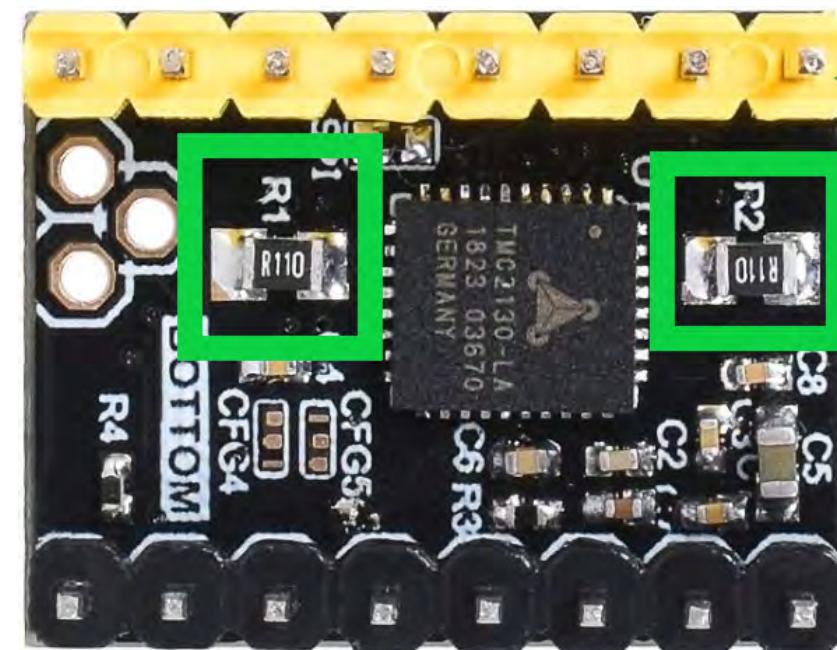
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

$R_s = R150$ is 0.15 Ohms

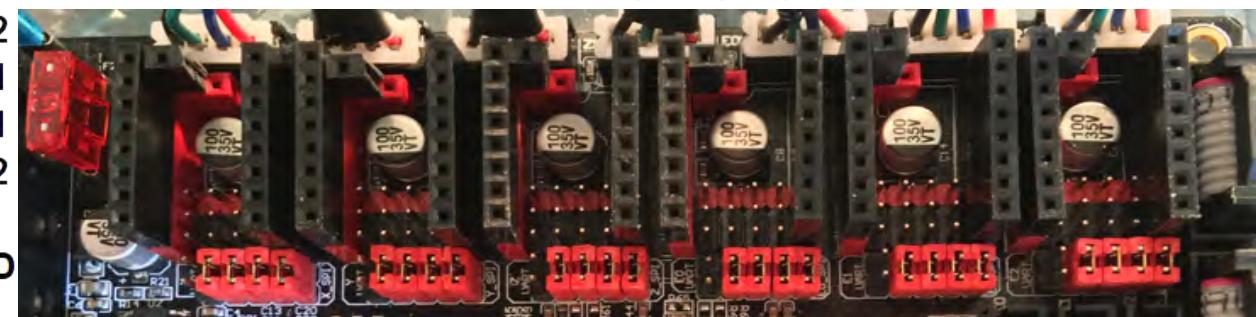
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



EN-CFG6	-	R	-	VM
SDI-CFG1	-	-	-	GND
SCK-CFG2	-	-	-	OB2
CSN-CFG3	-	-	-	OB1
SD0-CFG0	-	MS1	MS0	OA1
NC	-	-	-	OA2
STEP-REFL	-	-	-	VIO
DIR-REFR	-	-	-	GND
	-	S P I M	-	
	-	S P I M	-	

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



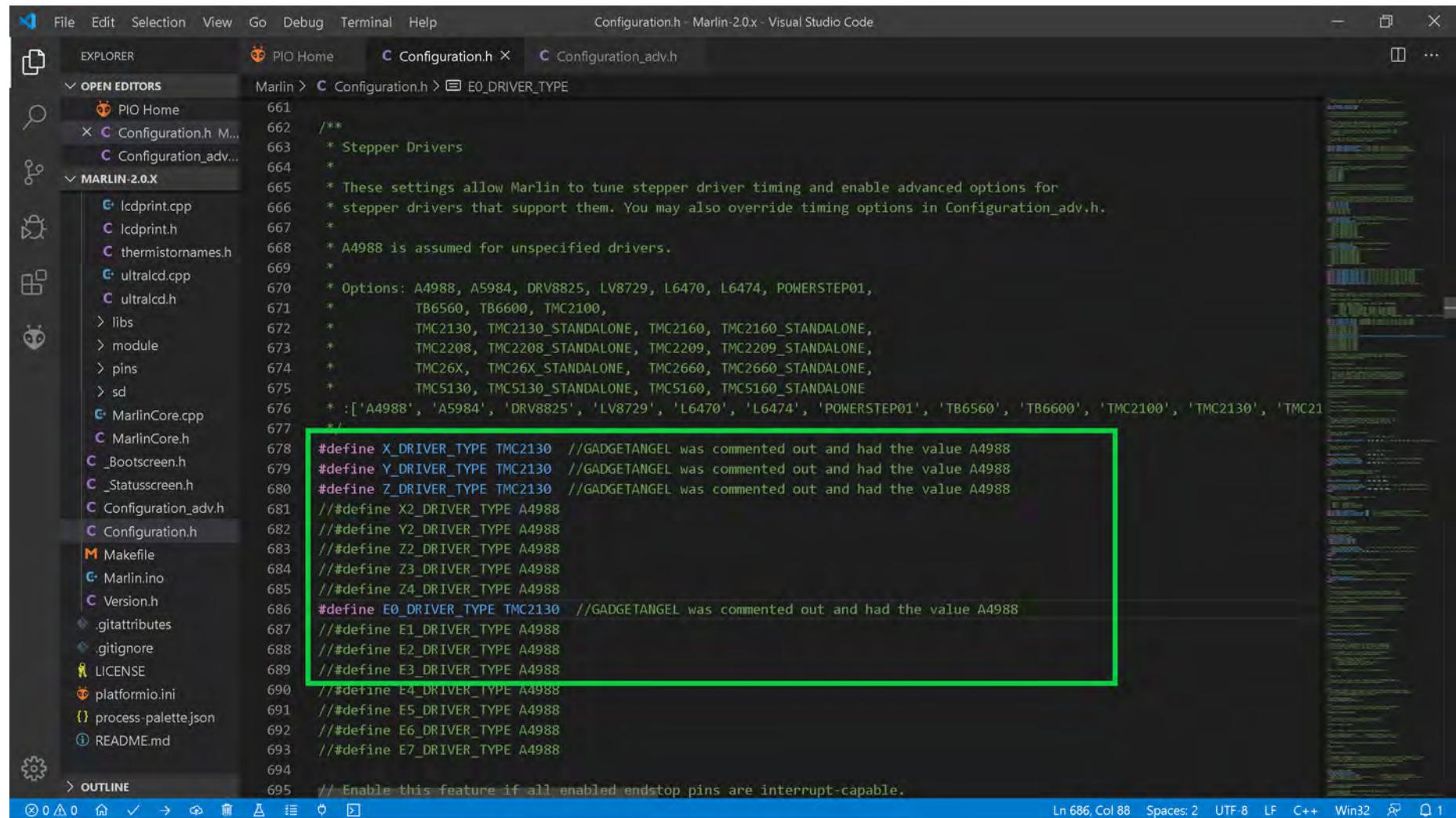
See [Appendix D](#) for legend

SPI

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
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 661, Col 88  Spaces: 2  UTF-8  LF  C++  Win32  ⚡  🔍  1

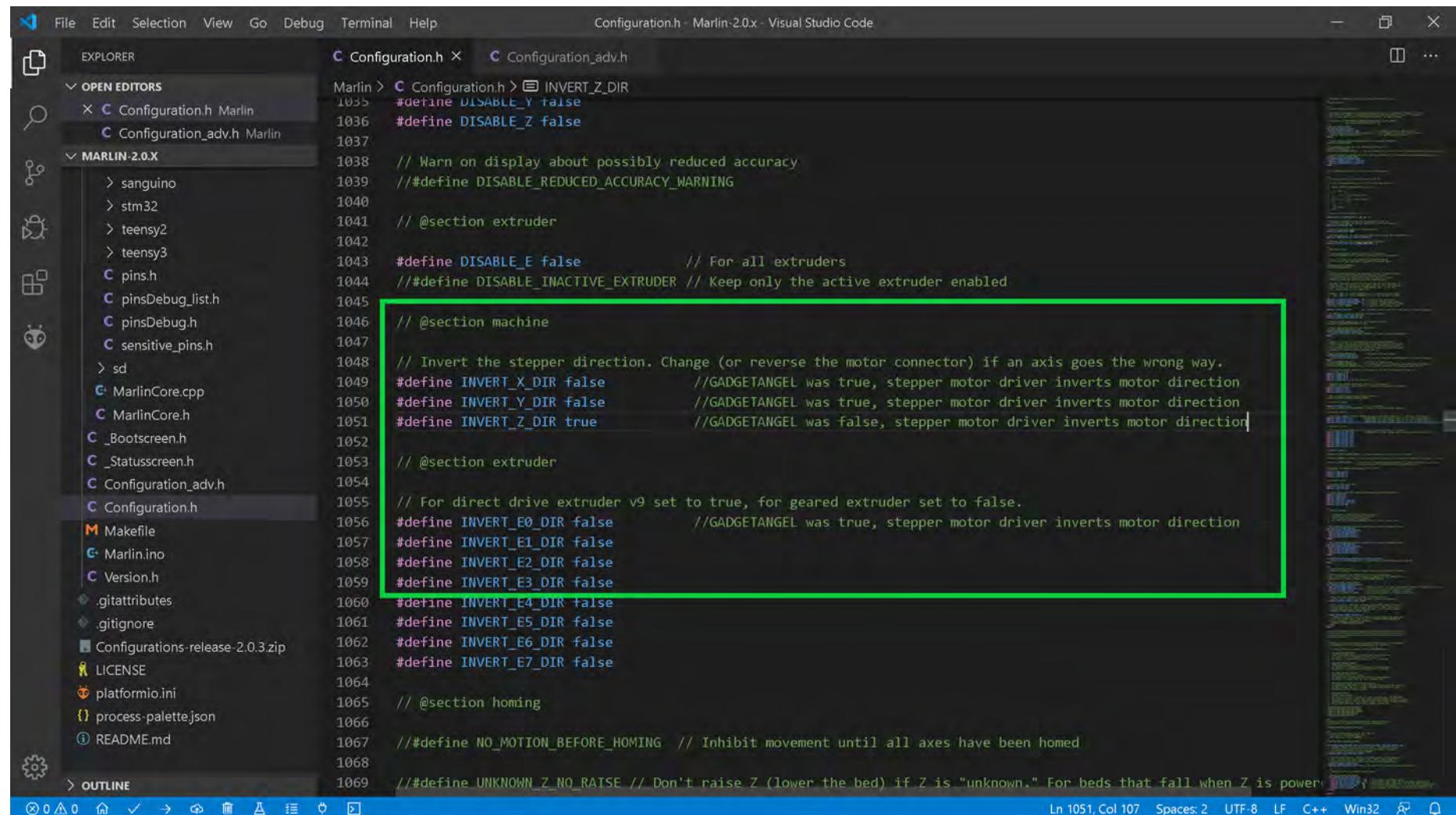
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: 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 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2130 //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 //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 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 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

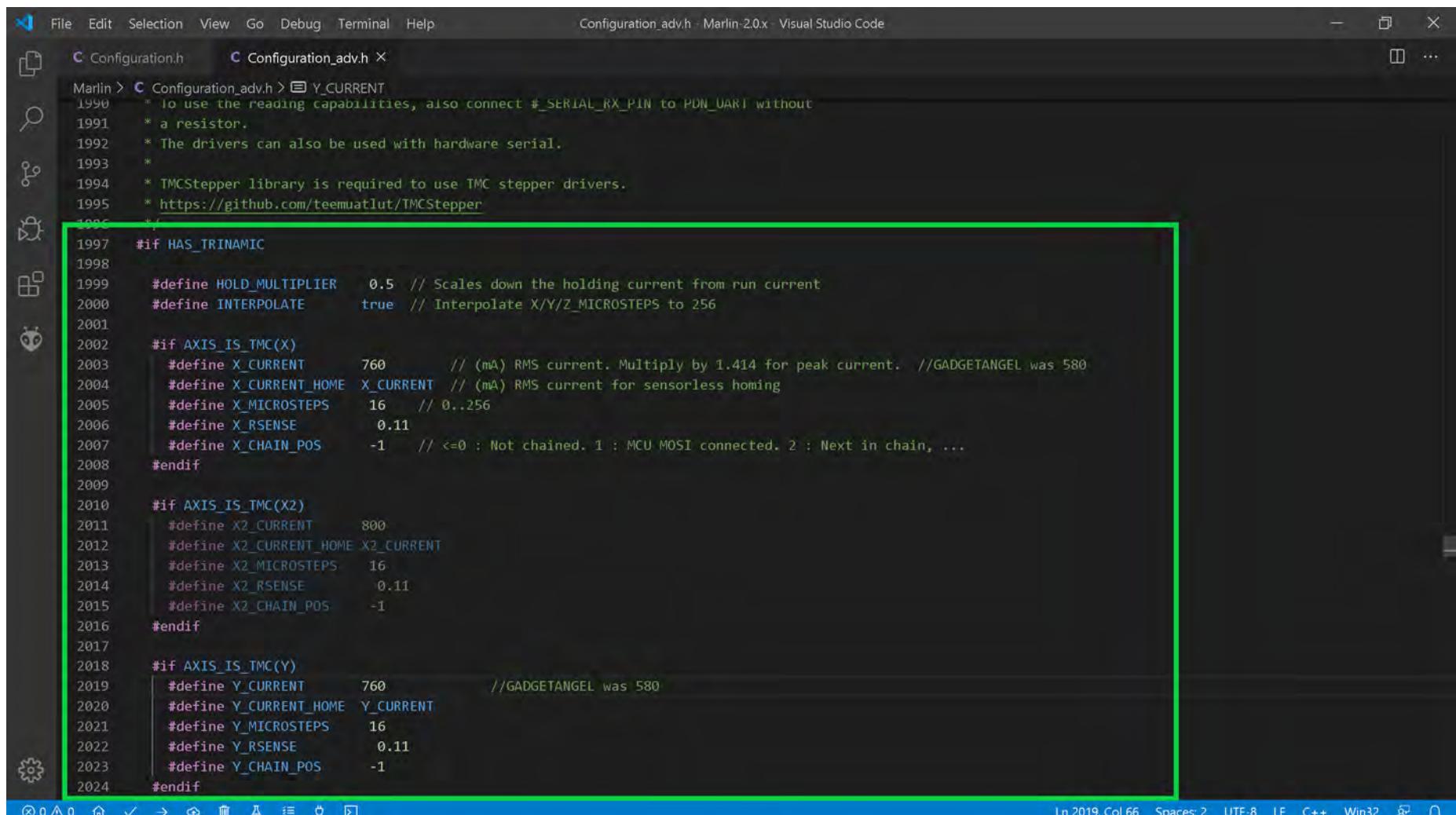
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that this line needs to be changed from `true` to `false` if the TMC2130 driver is being used.

- 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-2.0.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

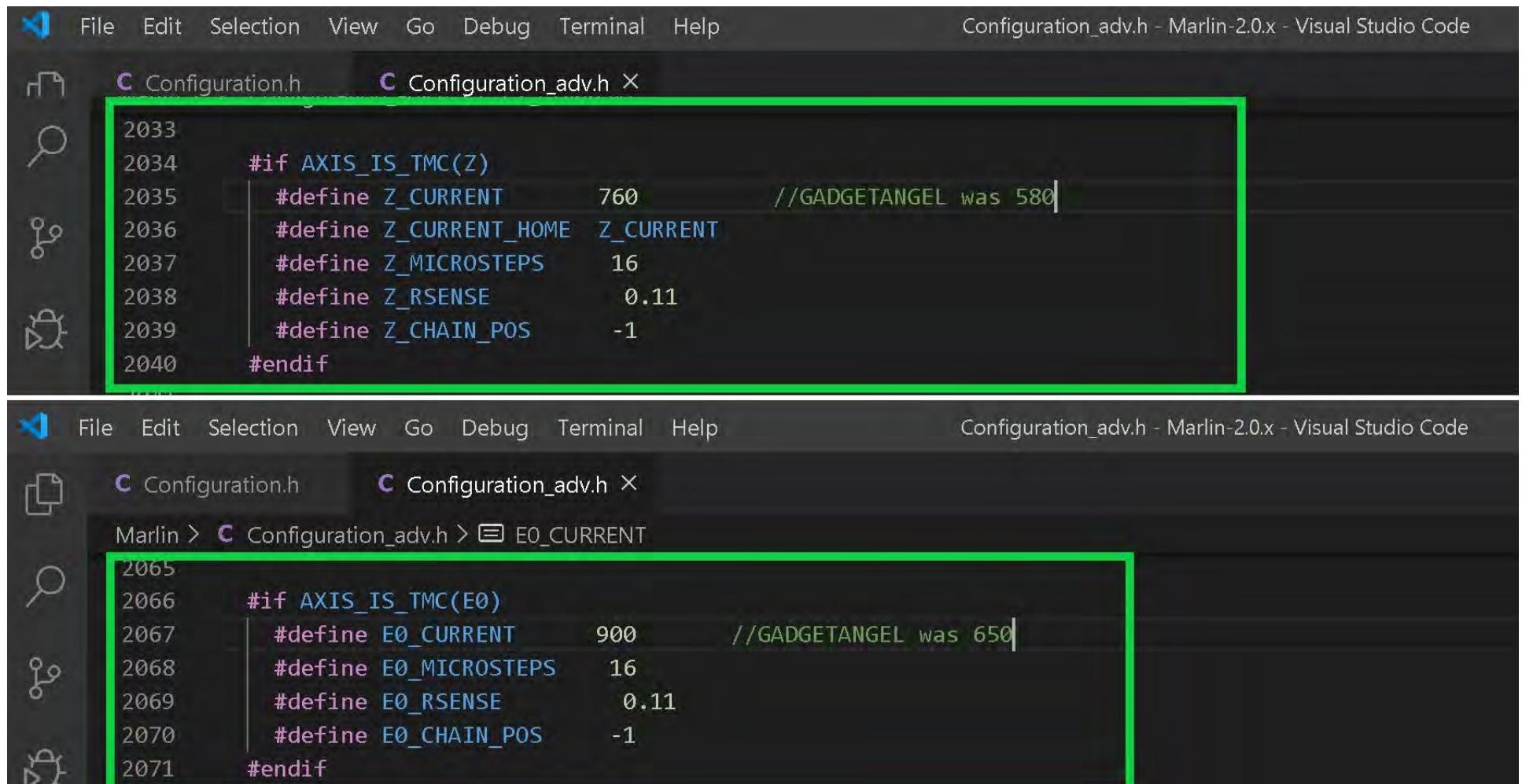
```

Ln 2019, 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".

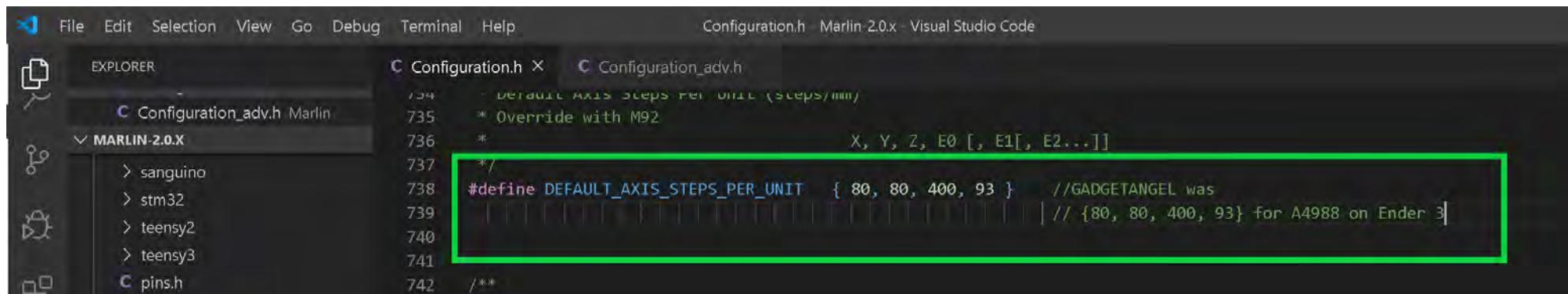


```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h X
2033
2034     #if AXIS_IS_TMC(Z)
2035         #define Z_CURRENT      760          //GADGETANGEL was 580
2036         #define Z_CURRENT_HOME Z_CURRENT
2037         #define Z_MICROSTEPS   16
2038         #define Z_RSENSE        0.11
2039         #define Z_CHAIN_POS    -1
2040     #endif
2041
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h X
Marlin > Configuration.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

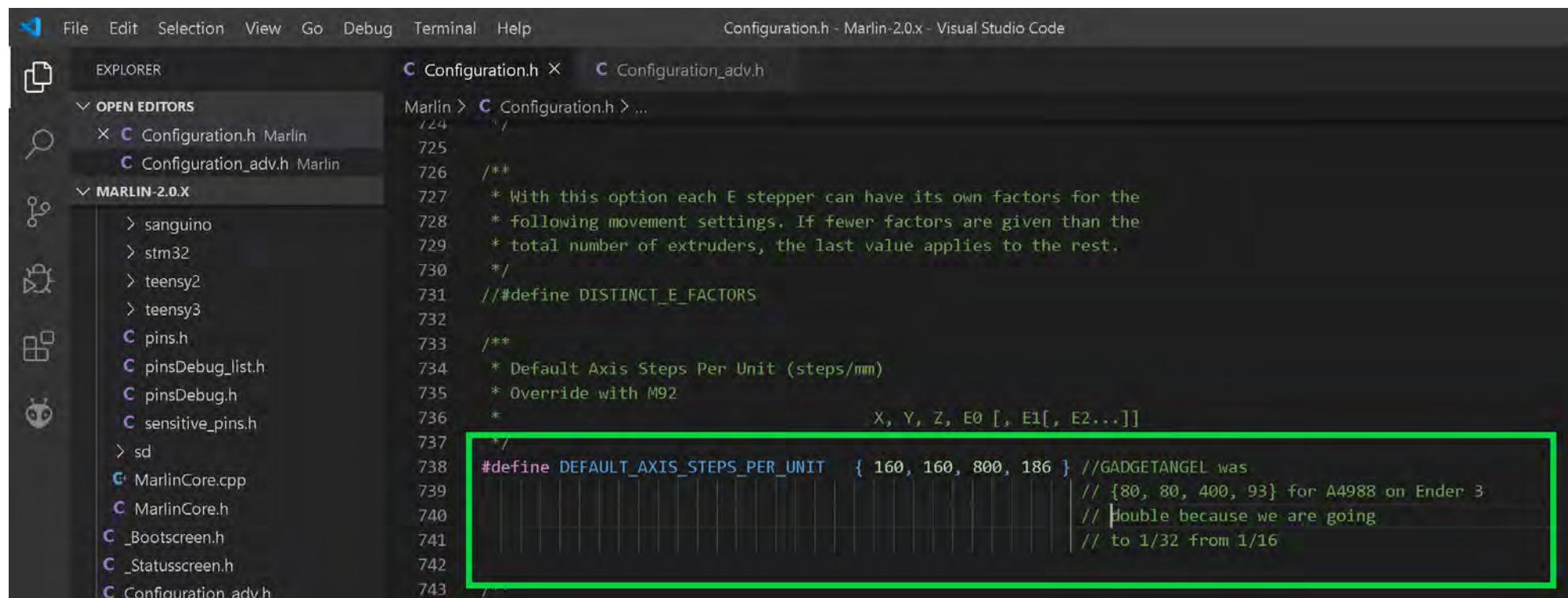
MARLIN-2.0.X

```

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

MARLIN-2.0.X

```

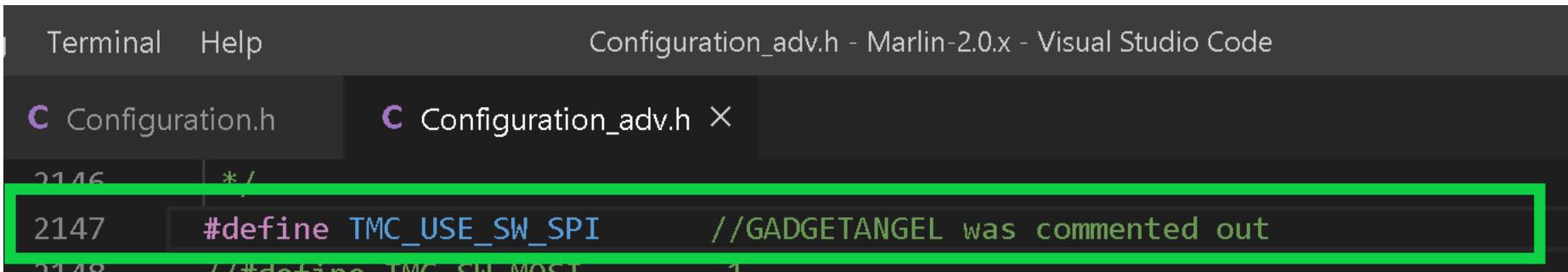
724 */
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis 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

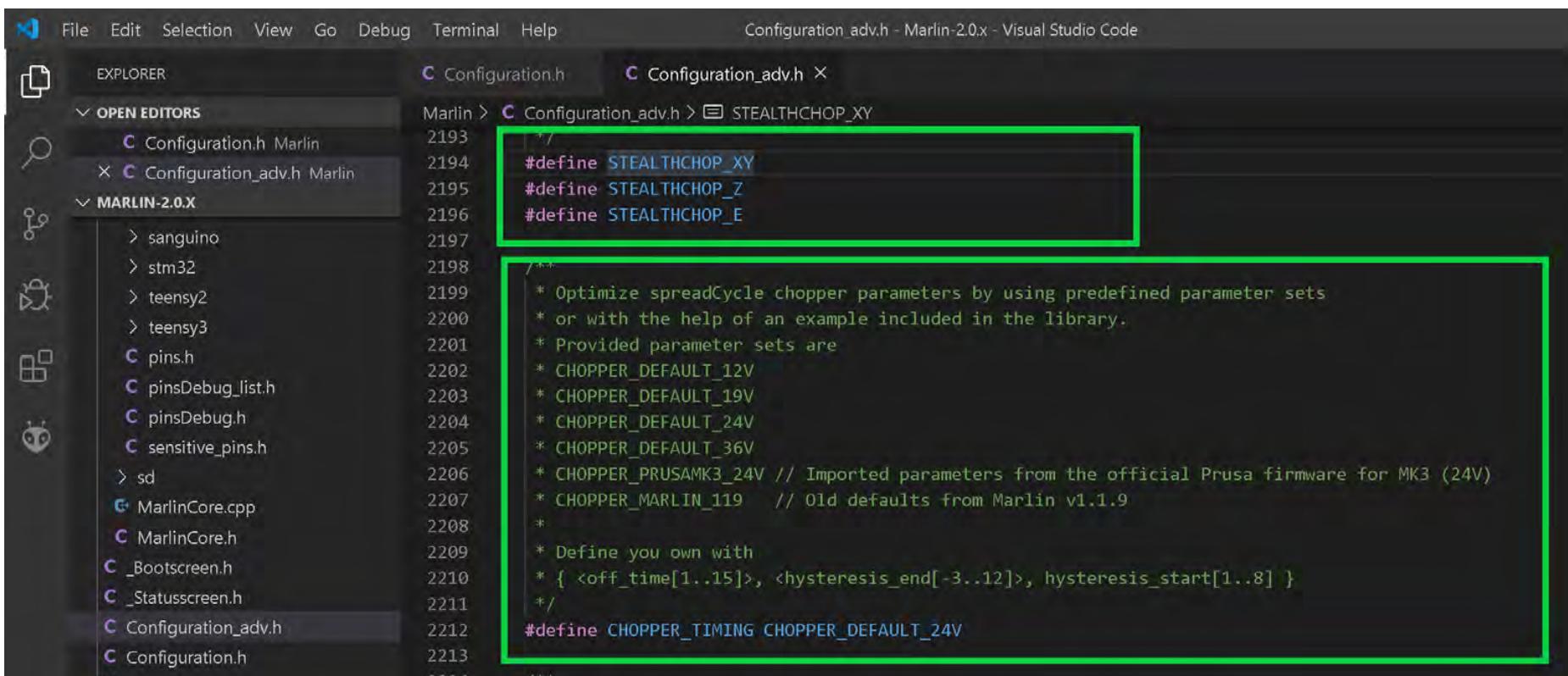


Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

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

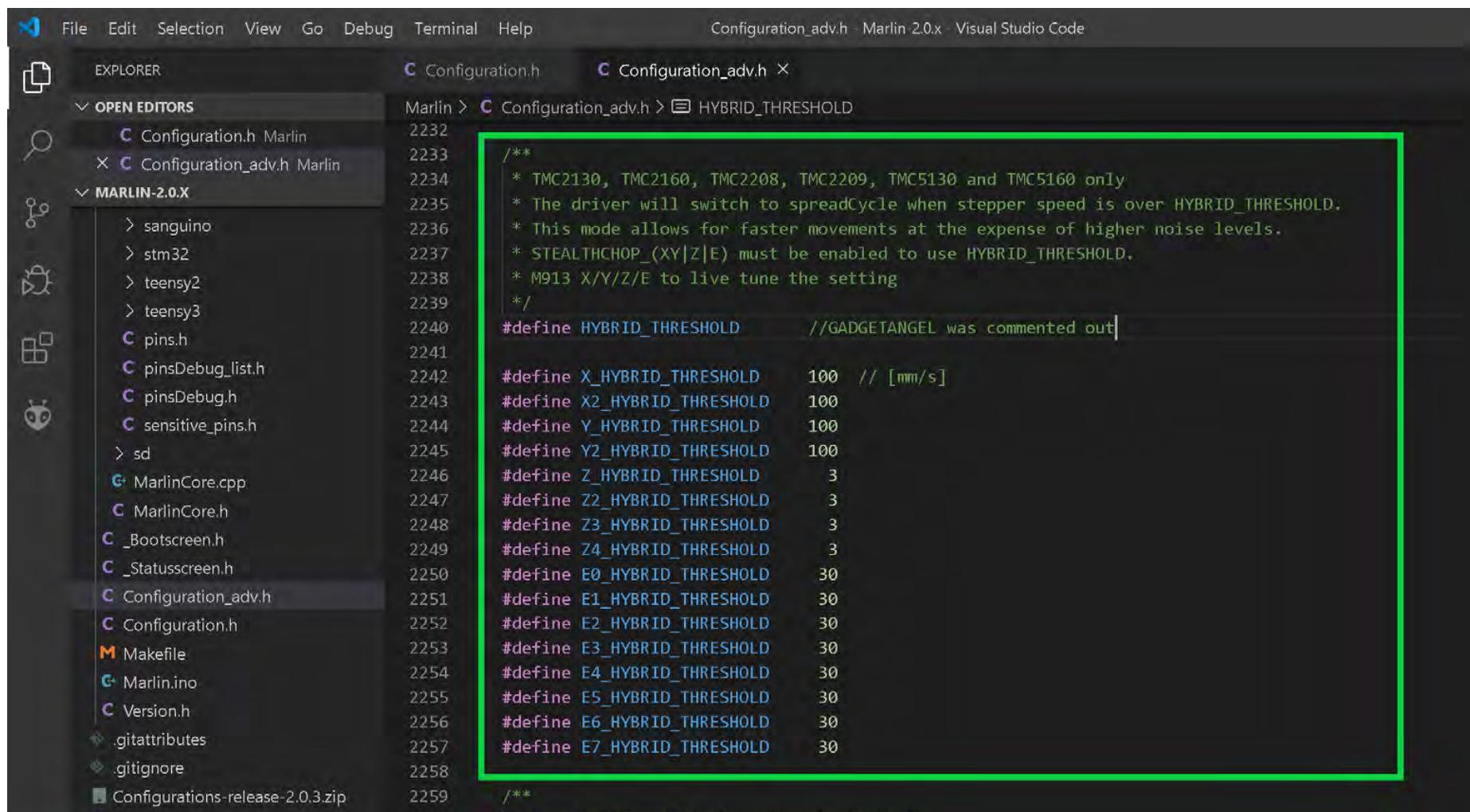
Marlin > C Configuration_adv.h > STEALTHCHOP_XY

```
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
- Explorer:** Shows the project structure under MARLIN-2.0.X, including files like 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 (selected), Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, and Configurations-release-2.0.3.zip.
- Editor:** Displays the content of Configuration_adv.h. A specific section for TMC2130 drivers is highlighted with a green border. The code within this section is as follows:

```


```

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

#define HYBRID_THRESHOLD //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

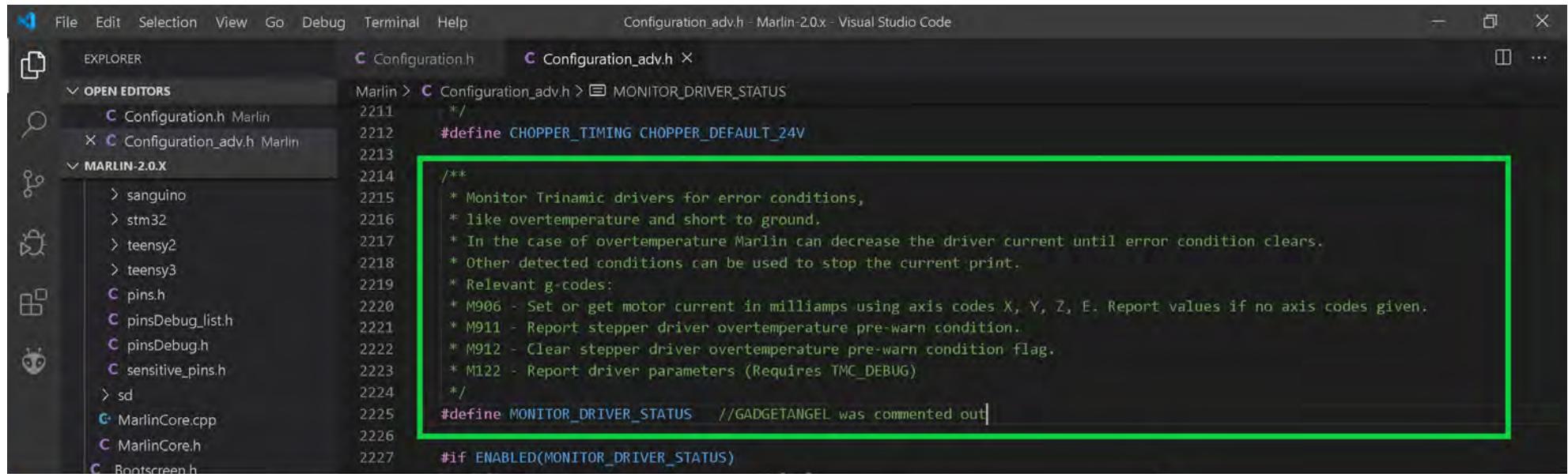
```


```

- 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](#).



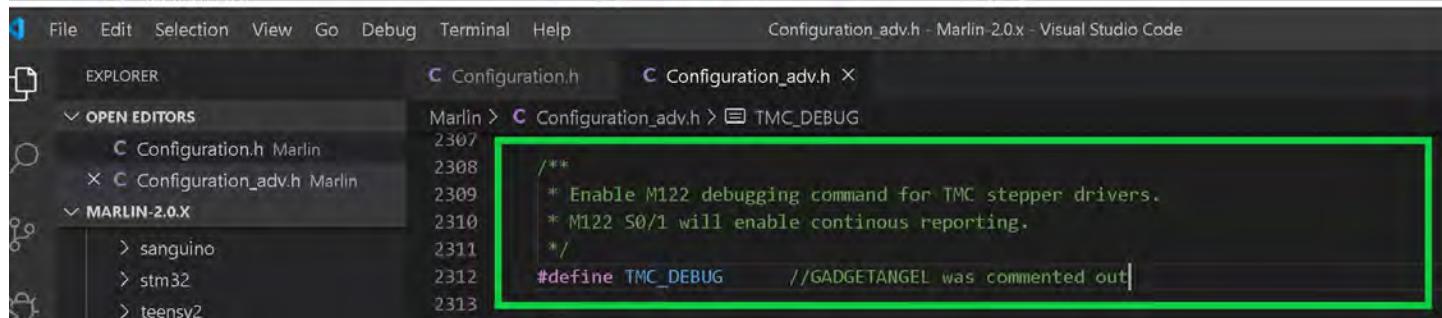
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h Configuration_adv.h 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 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 Configuration.h Configuration_adv.h Marlin > Configuration_adv.h > TMC_DEBUG

```

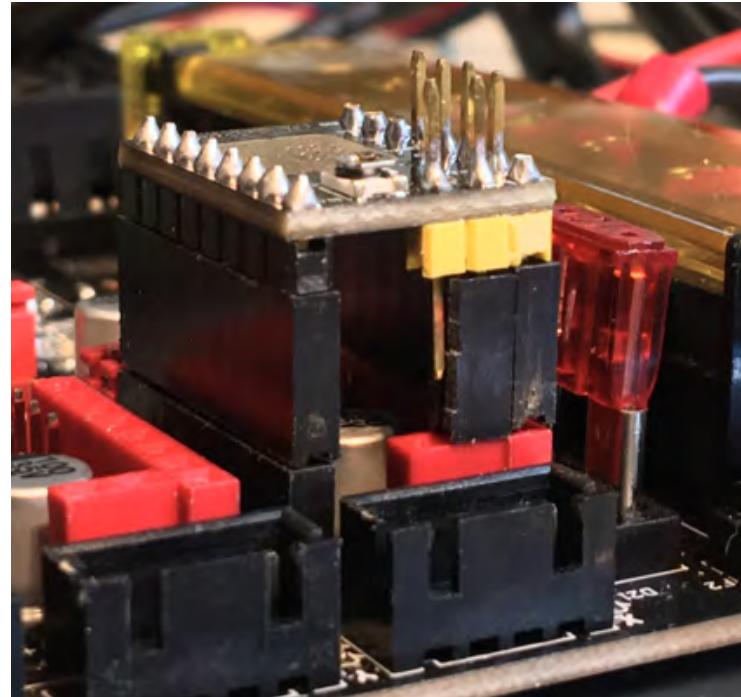
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

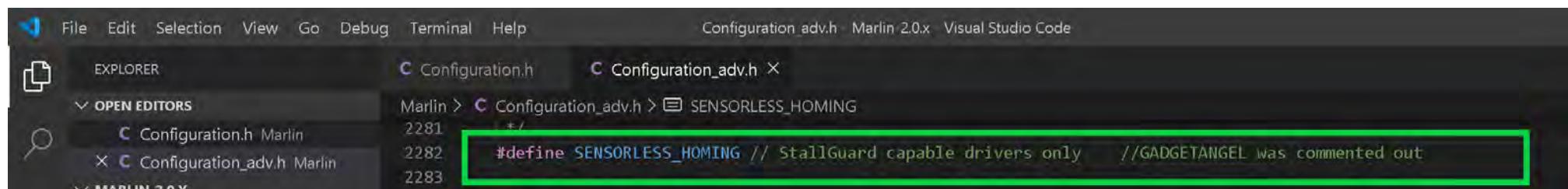
- Go to the next page.

The (latest release of) Marlin Setup for BIQU 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"

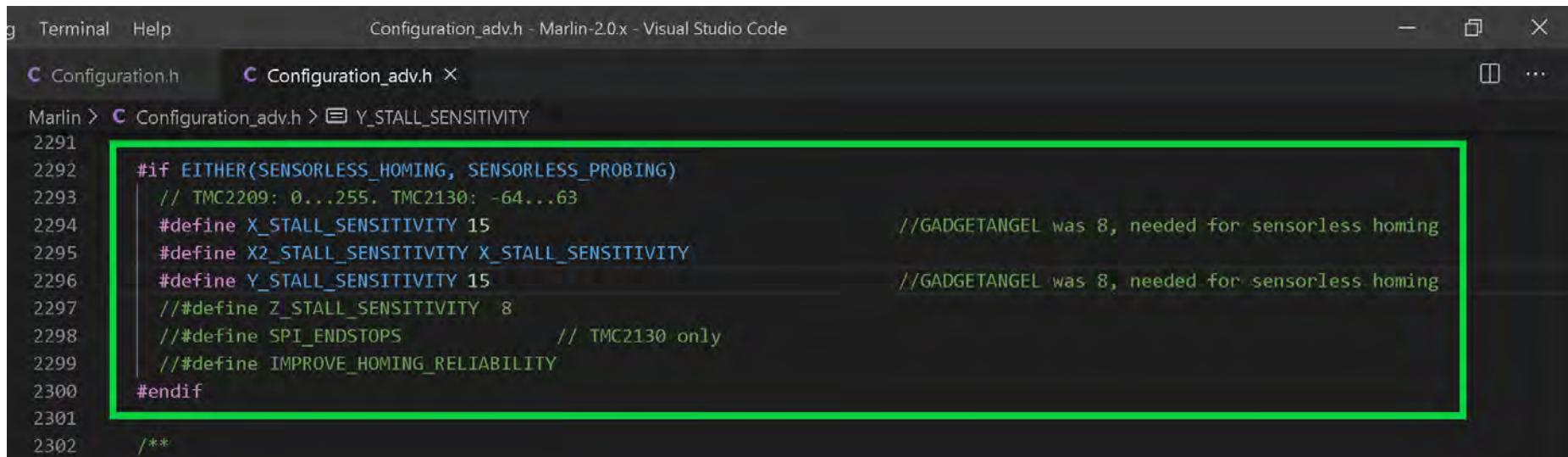


```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code  
EXPLORER Configuration.h Configuration_adv.h X  
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING  
2281 */  
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL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

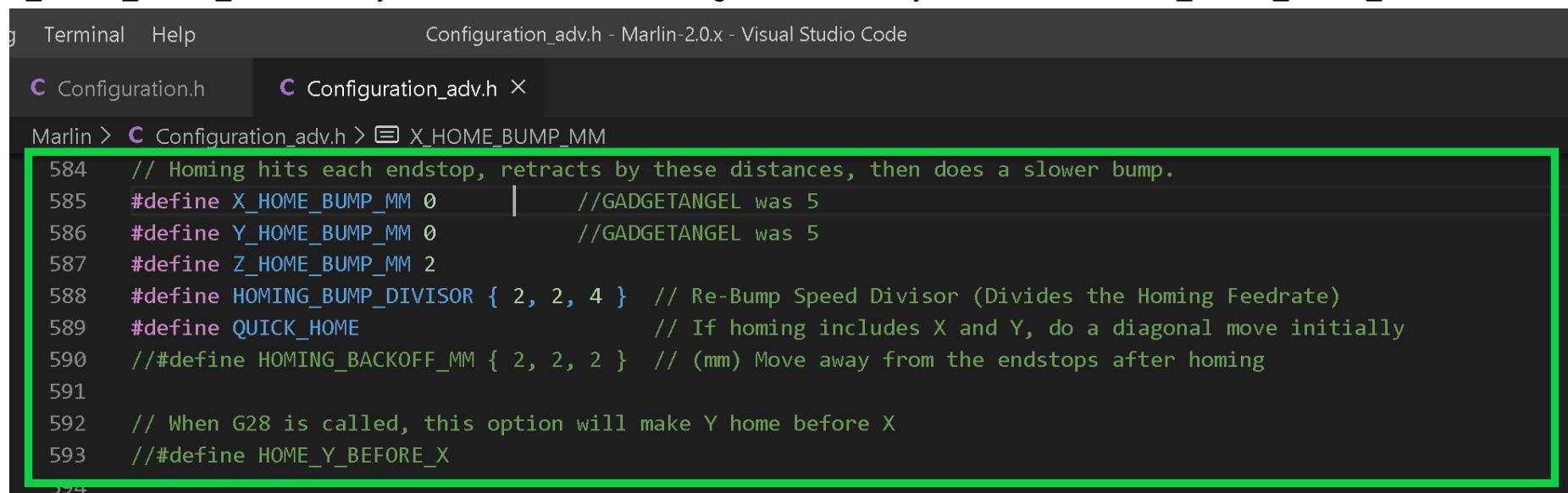


```

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.



```

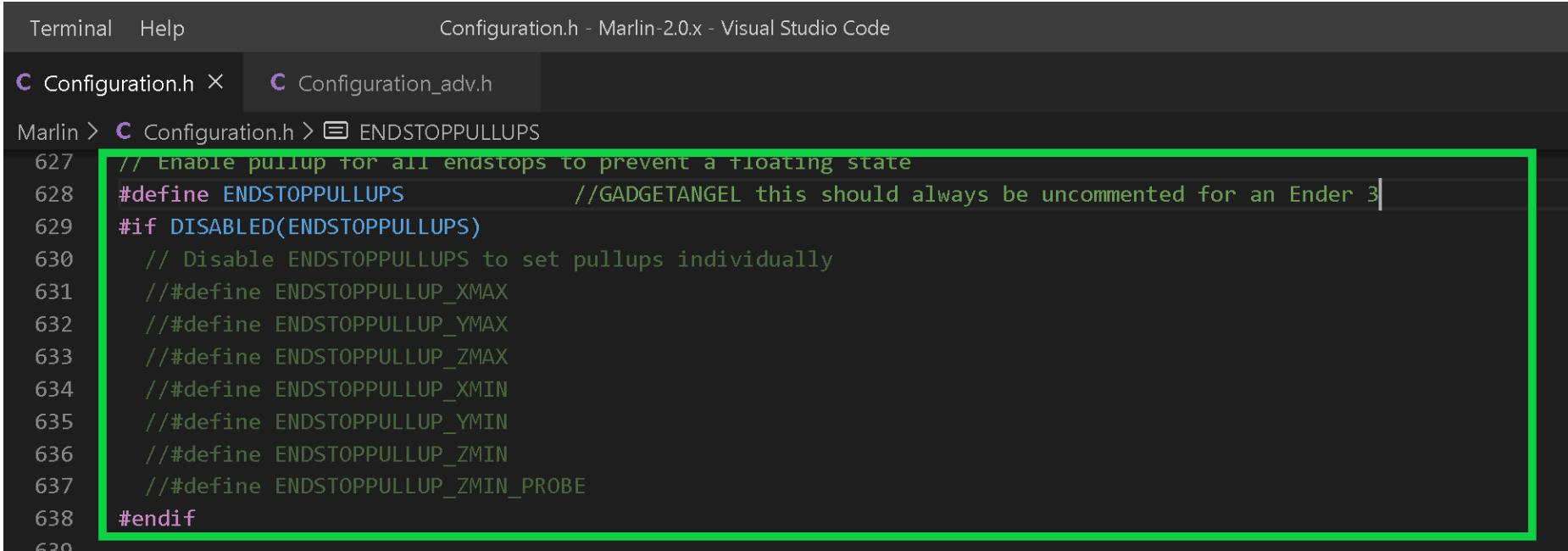
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

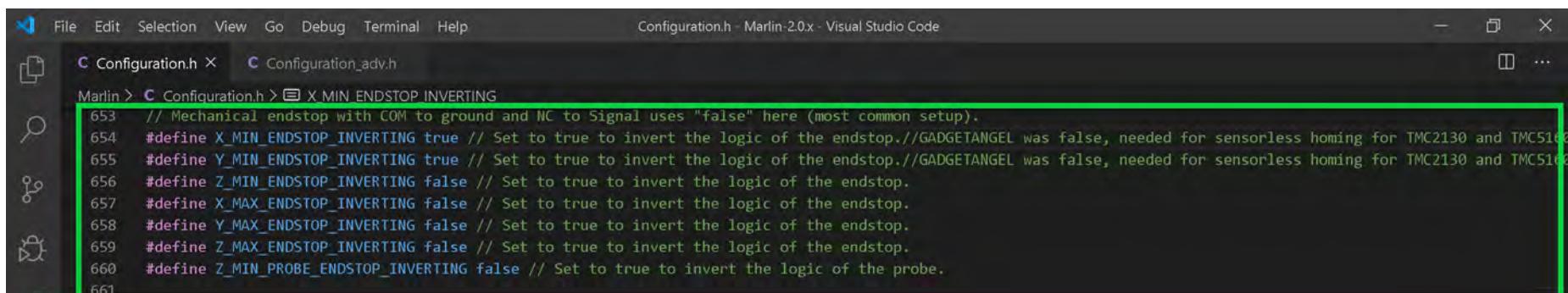
C Configuration.h X C Configuration_adv.h

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

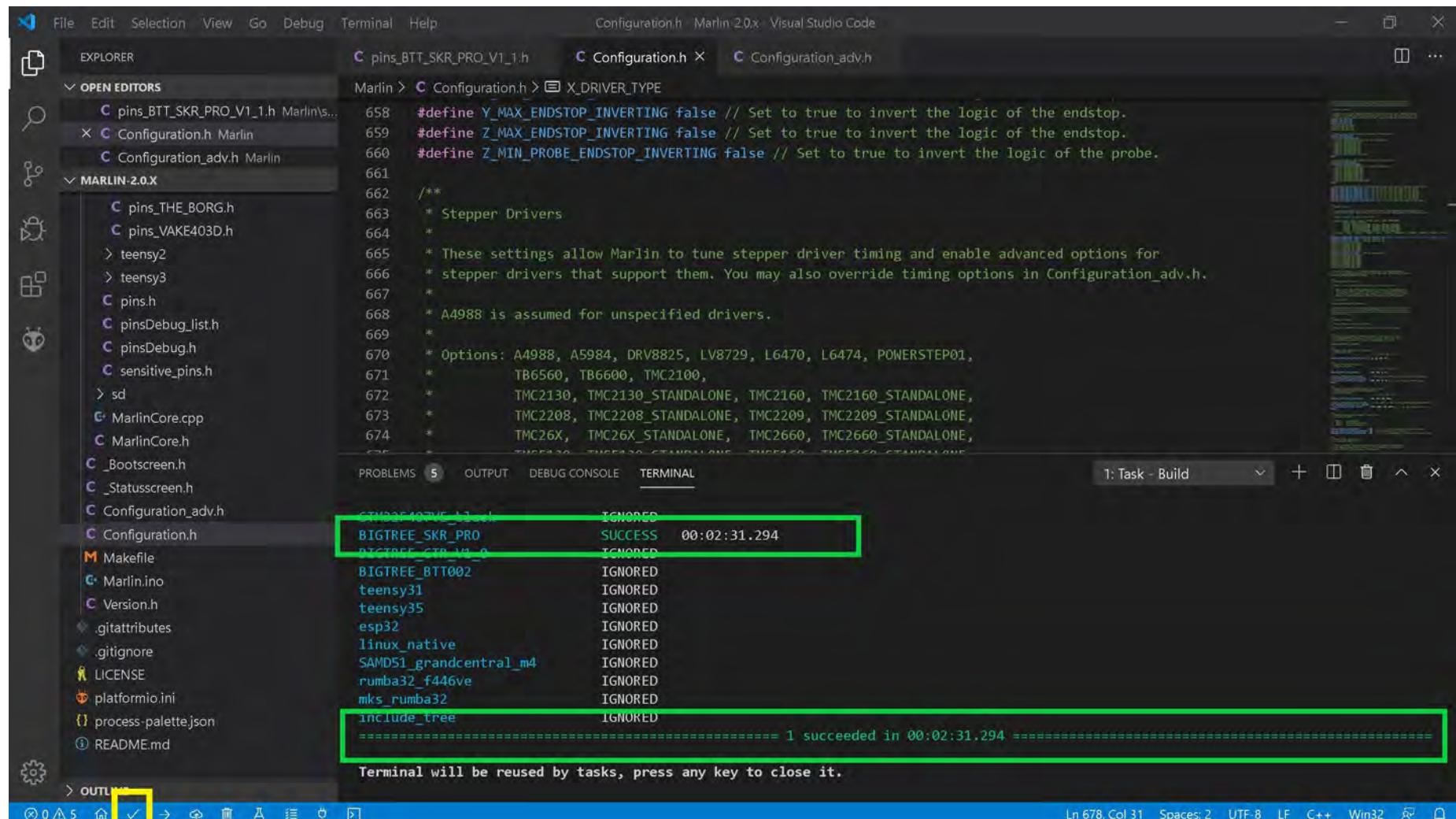
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5100
655 #define Y_MIN_ENDSTOP_INVERTING 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.

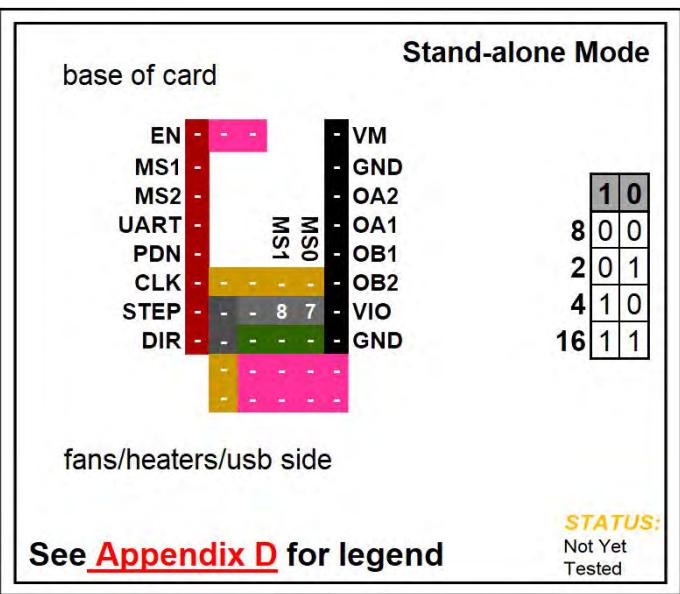


```

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 /**
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
1: Task - Build + ⌂ ⌄ ⌈ ⌉ ×
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
TMC2660_STANDALONE
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_CTP_M4_0 IGNORED
BIGTREE_BTT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
SAMD51_grandcentral_m4 IGNORED
rumba32_f446ve IGNORED
mks_rumba32 IGNORED
include_tree IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
Terminal will be reused by tasks, press any key to close it.
Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌄ ⌈ ⌉

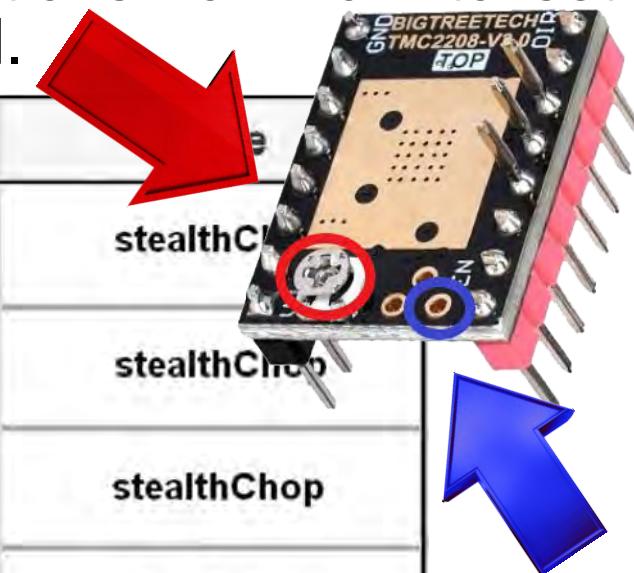
```

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

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	
BIQU® TMC2208 <small>Stand Alone Mode Maximum 16 Subdivision 35V DC 2A (peak)</small>	Low	Low	1 / 8	1 / 256	stealthChop
	Low	High	1 / 2	1 / 256	stealthChop
	High	Low	1 / 4	1 / 256	stealthChop
	High	High	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:

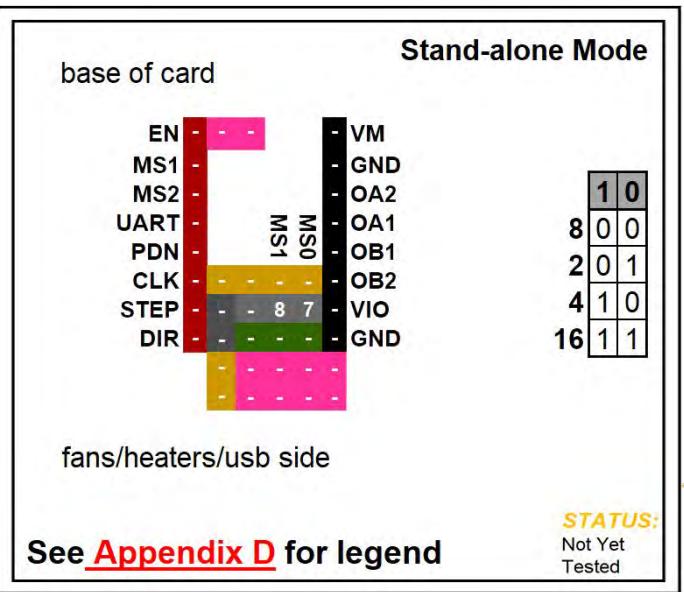
$$I_{MAX} = (V_{ref} * 0.9286) * 0.90$$

$$V_{ref} = I_{MAX} * 1.0769$$

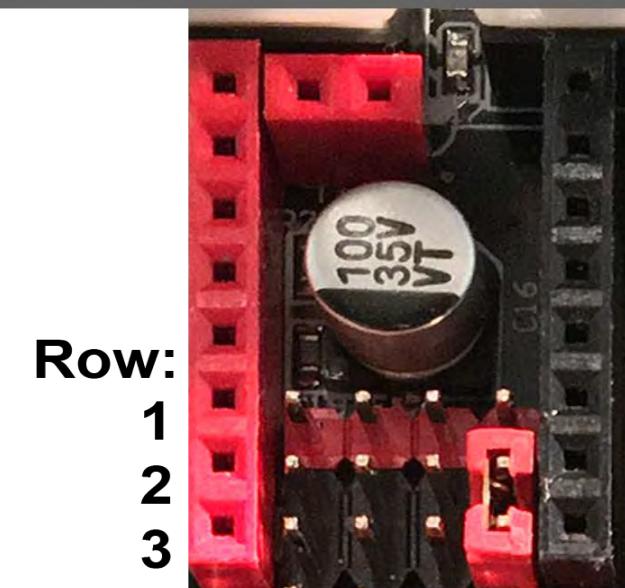
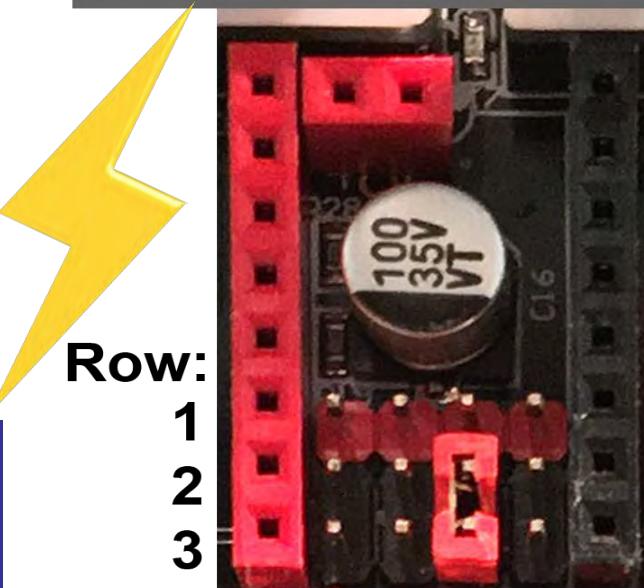
See Appendix B #3. Use 50% to 90% as shown below:

$$V_{ref} = (I_{MAX} * 1.0769) * 0.90$$

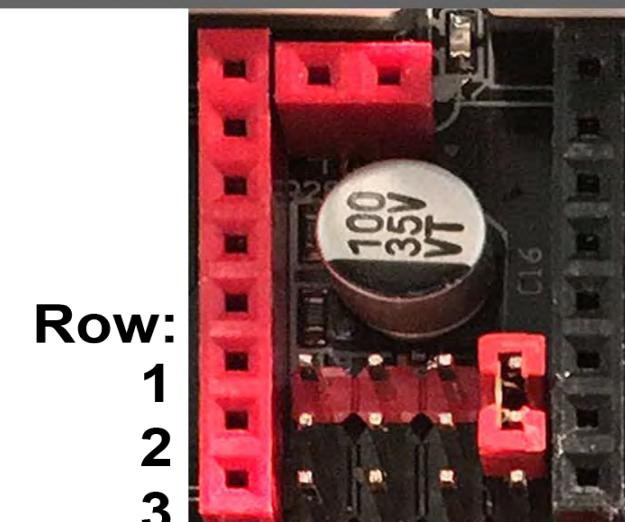
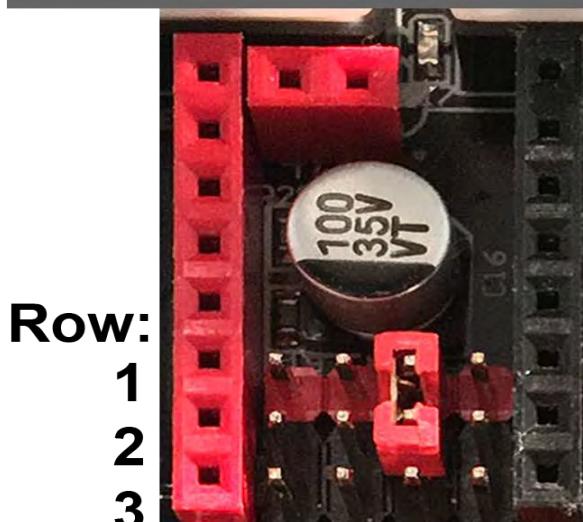
- See next page for the legend that belongs to the above chart.

**BIQU TMC2208 V3.0**Stand-alone Mode**SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

Low ➤ set Jumper between rows 2 and 3

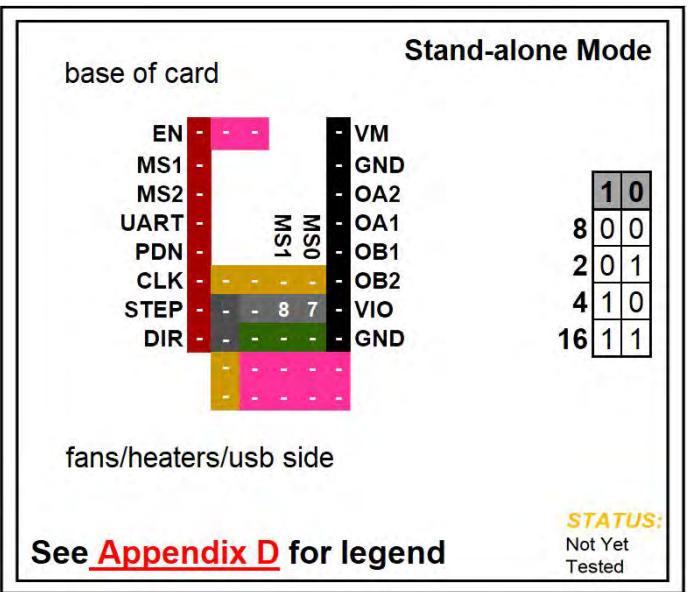


High ➤ set Jumper between rows 1 and 2

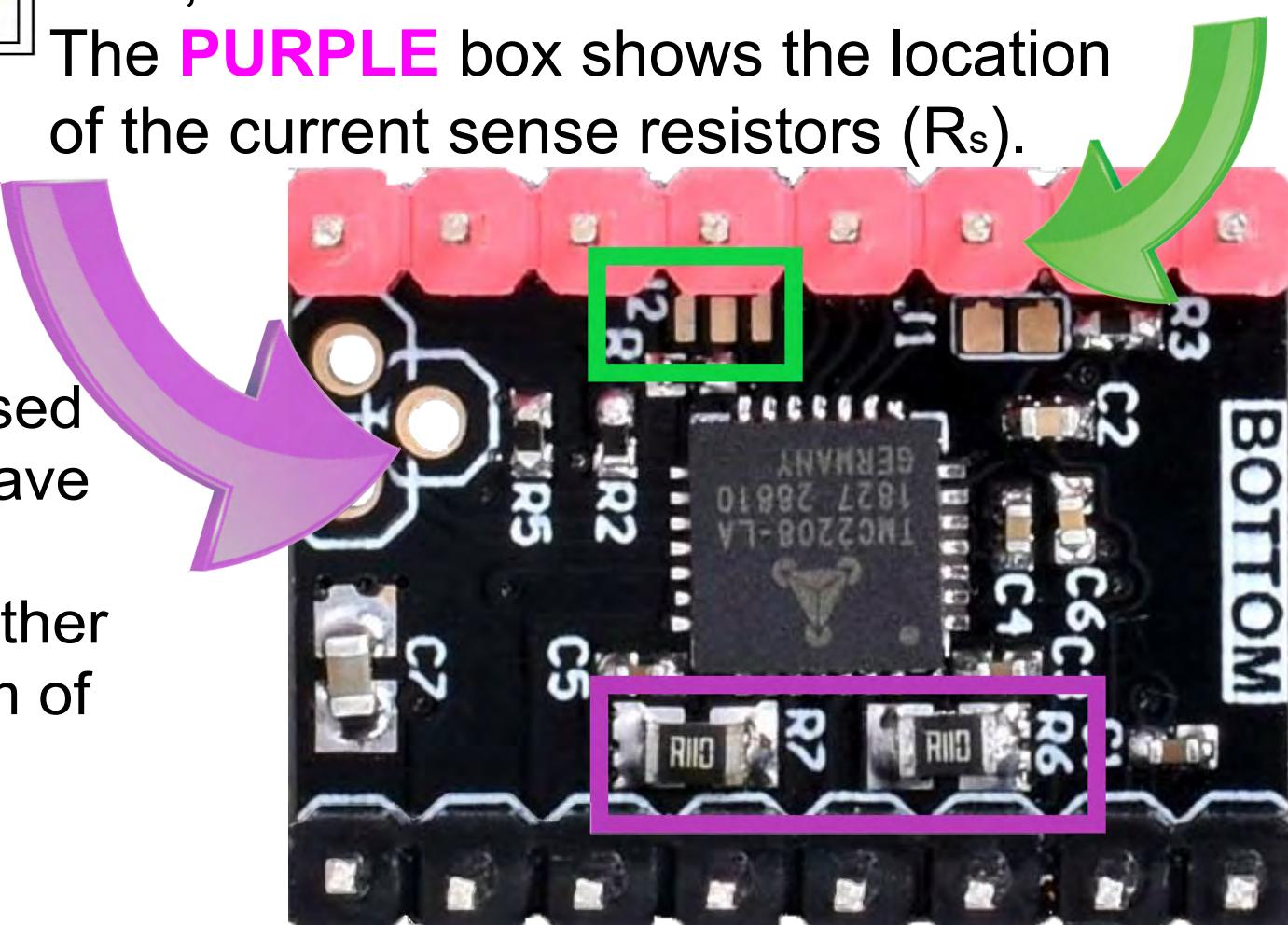


This is a
Jumper:



**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

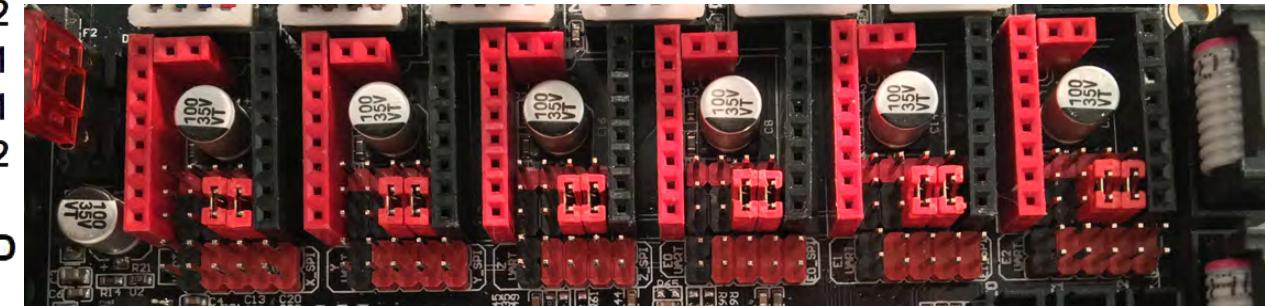
Stand-alone
Mode

1 / 8

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	MS1	OA1
PDN	-	-	MS0	OB1
CLK	-	-	-	OB2
STEP	-	8	7	VIO
DIR	-	8	7	GND



See [Appendix D](#) for legend

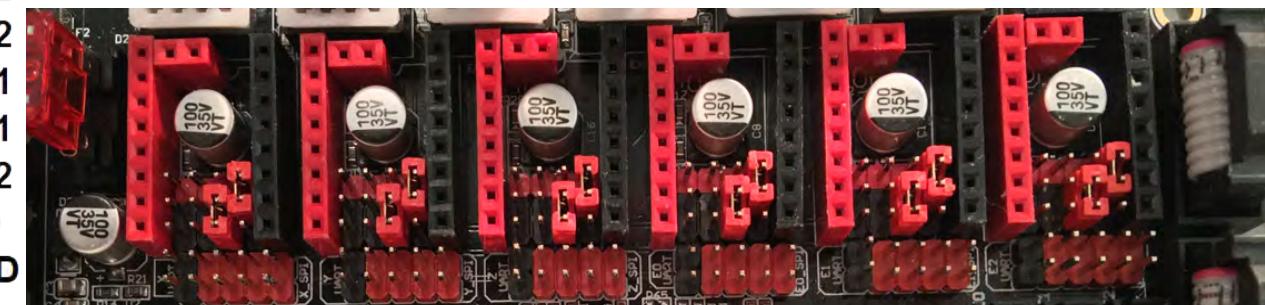
Stand-alone
Mode

1 / 2

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	MS1	OA1
PDN	-	-	MS0	OB1
CLK	-	-	7	OB2
STEP	-	8	7	VIO
DIR	-	8	7	GND



See [Appendix D](#) for legend

Stand-alone Mode

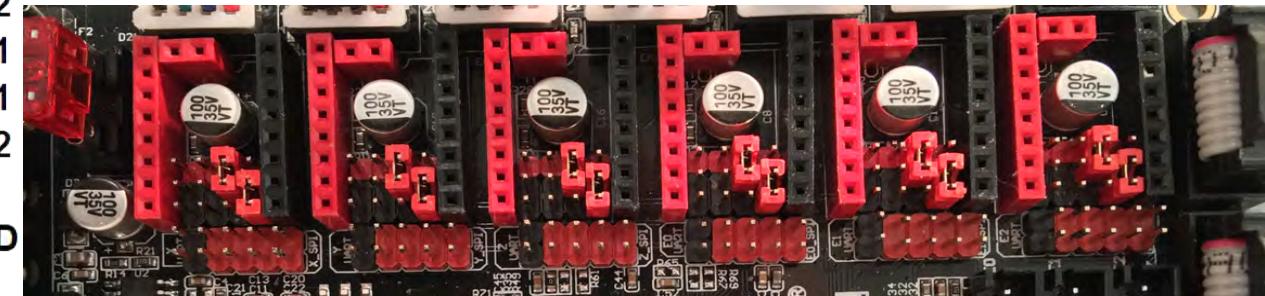
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	MS1	MS0	OA1
PDN	-	-	-	OB1
CLK	-	8	-	OB2
STEP	-	8	7	VIO
DIR	-	-	7	GND



See [Appendix D](#) for legend

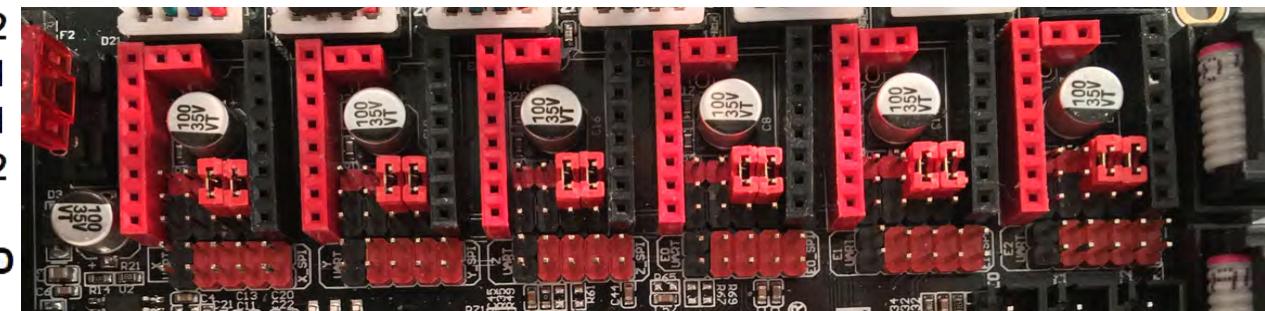
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	MS1	MS0	OA1
PDN	-	-	-	OB1
CLK	-	8	7	OB2
STEP	-	8	7	VIO
DIR	-	-	-	GND

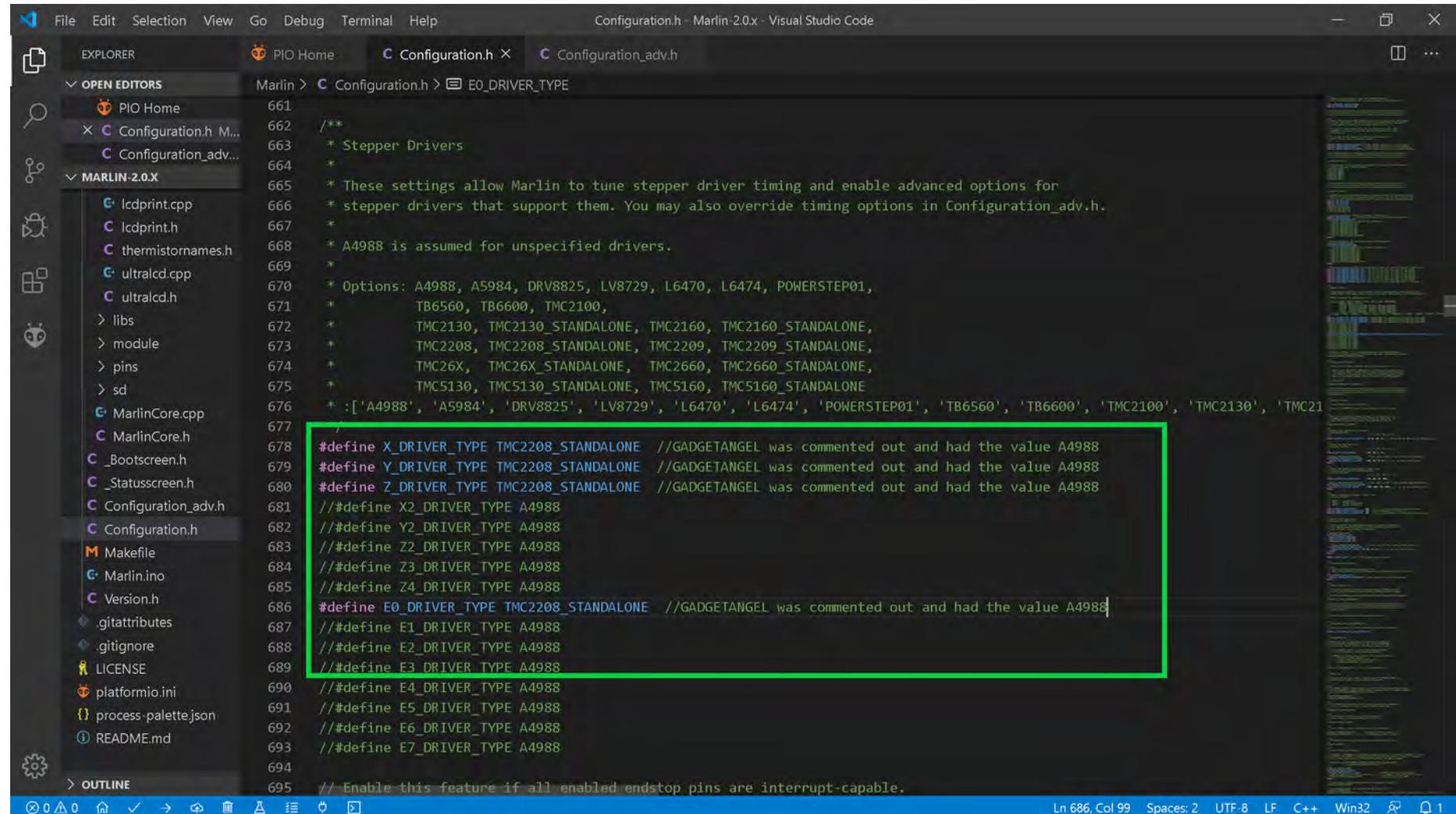


See [Appendix D](#) for legend

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
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... 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', 'TMC2208', 'TMC2209', 'TMC2660', 'TMC5130', 'TMC5160']
  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
  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 indicates '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

```

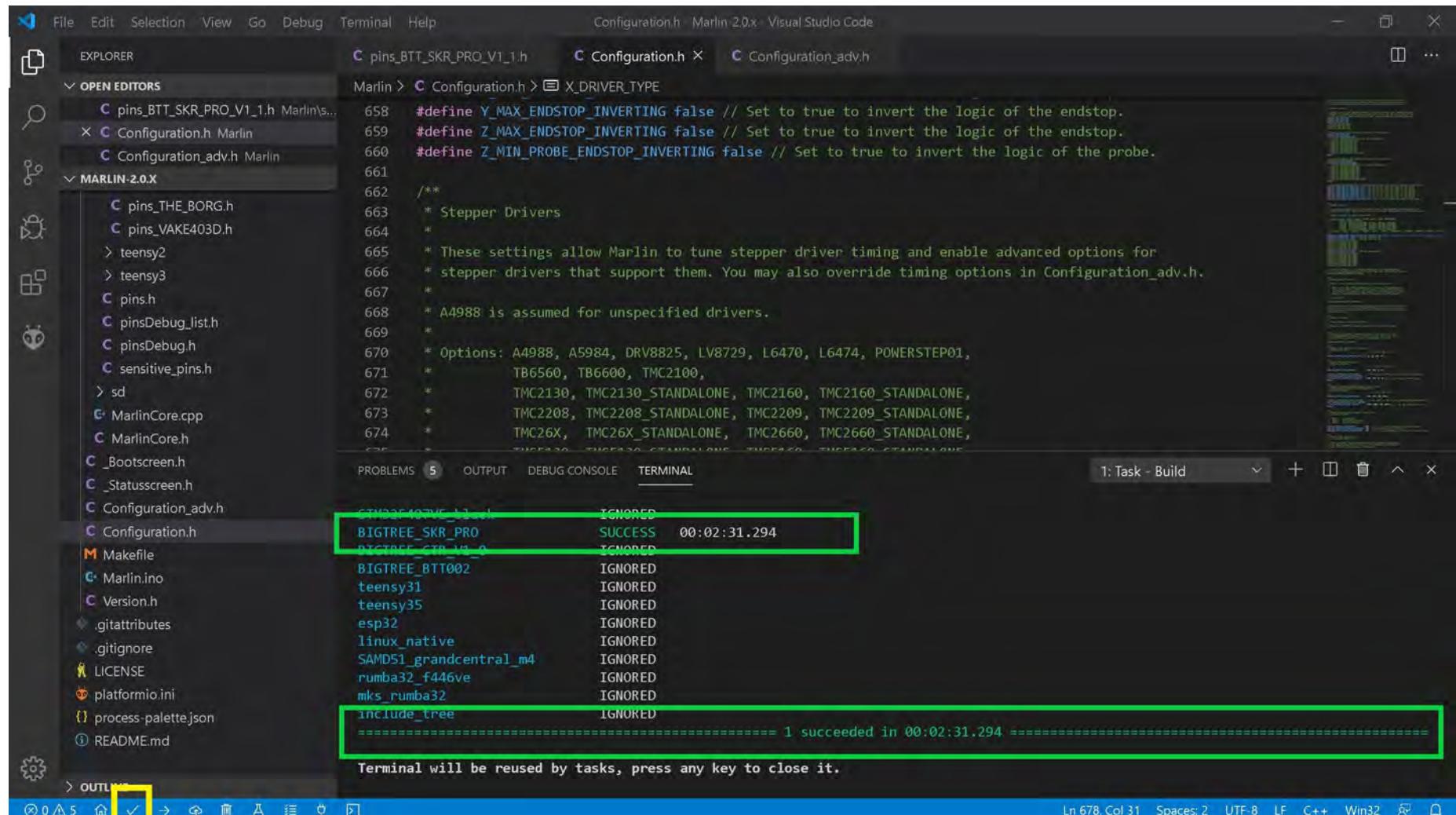
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  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.



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 pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h

MARLIN-2.0.X Configuration.h X X_DRIVER_TYPE

```
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, 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 * 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		

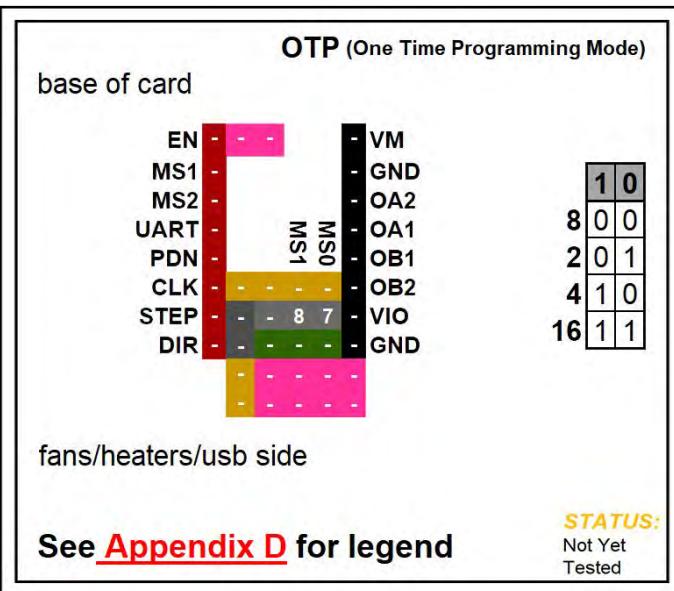
===== 1 succeeded in 00:02:31.294 =====

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 [Appendix E](#).

One Time Programming (OTP) Mode

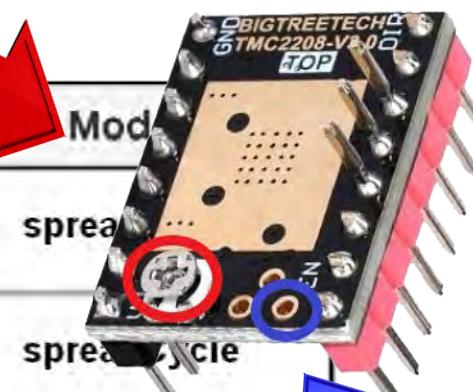


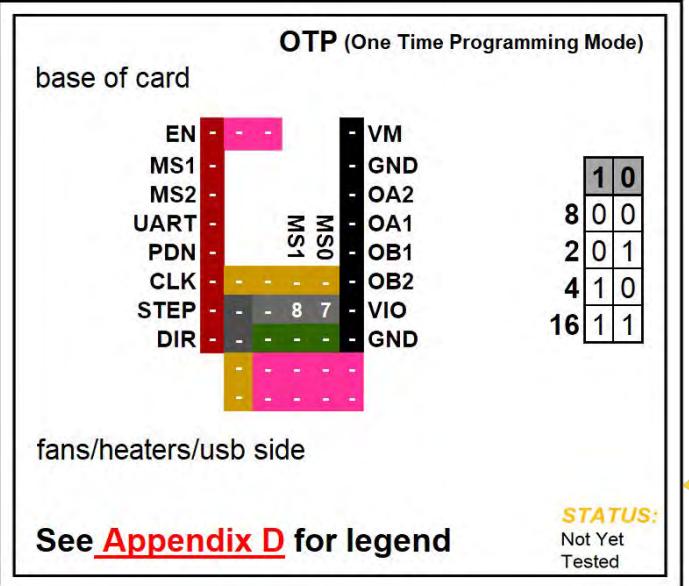
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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.

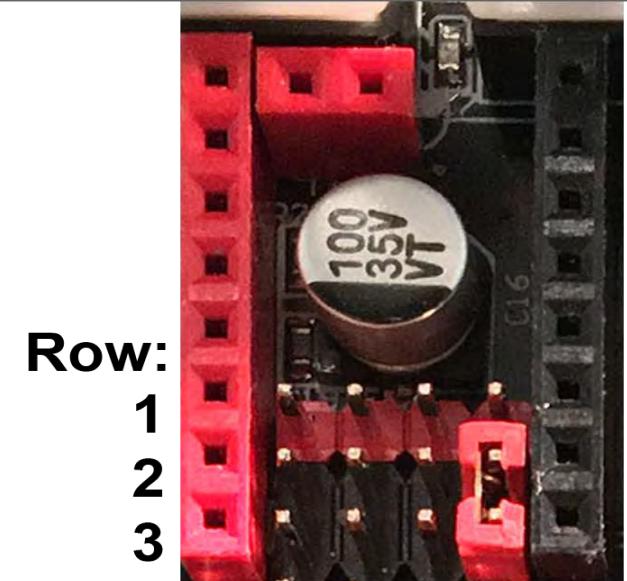
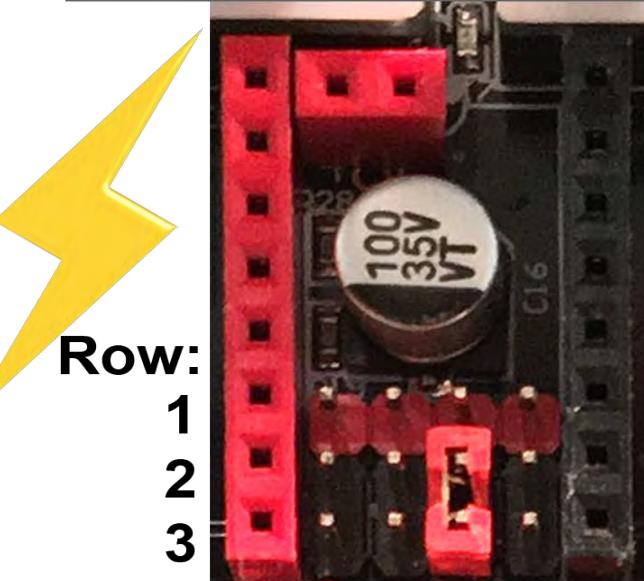
Driver Chip	MS1	MS0	Steps	Interpolation	Mod
Biqu® TMC2208 OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	1 / 8	1 / 256	spreadCycle
	Low	High	1 / 2	1 / 256	spreadCycle
	High	Low	1 / 4	1 / 256	spreadCycle
	High	High	1 / 16	1 / 256	spreadCycle
Driving Current Calculation Formula R_S (Typical Sense 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: $V_{ref} = (I_{MAX} * 1.0769) * 0.90$	

- See next page for the legend that belongs to the above chart.

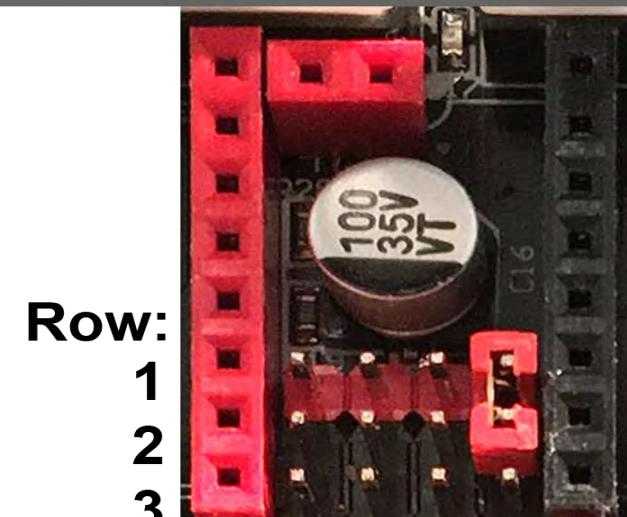
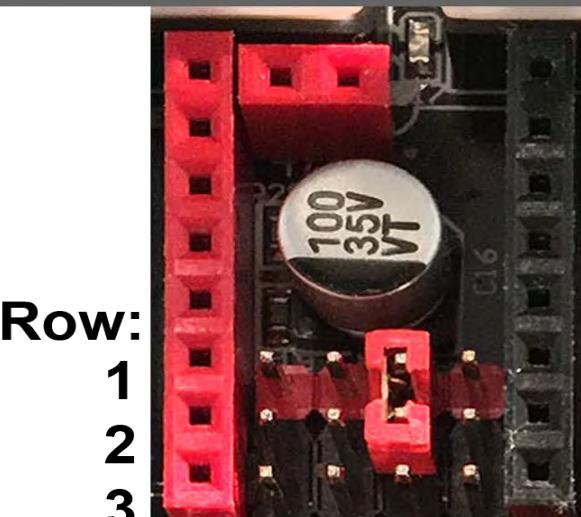


BIQU TMC2208 V3.0**One Time Programming (OTP) Mode****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

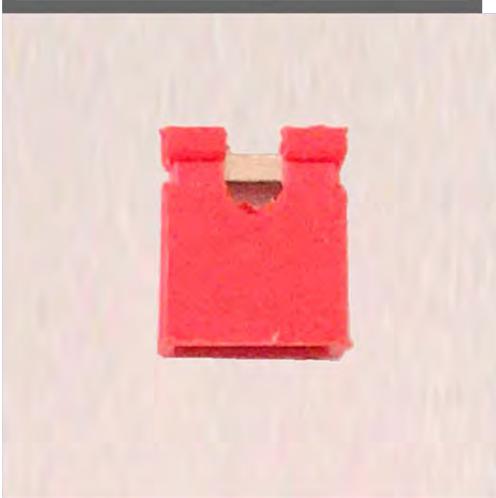
Low ➡ set Jumper between rows 2 and 3



High ➡ set Jumper between rows 1 and 2



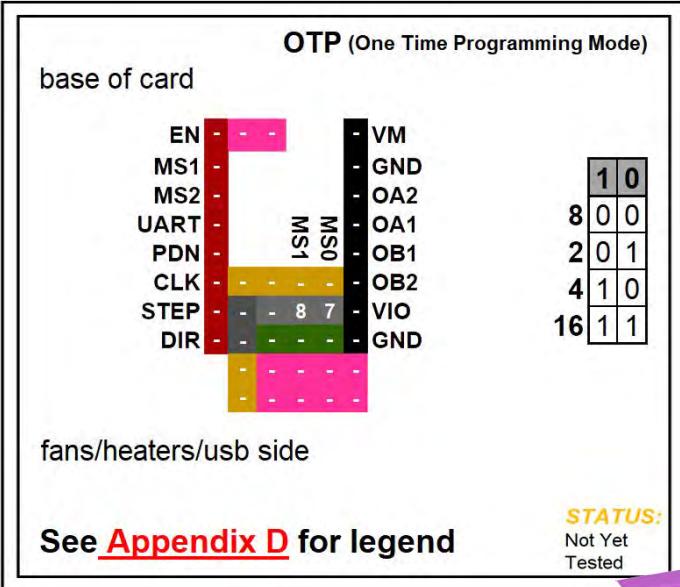
This is a
Jumper:



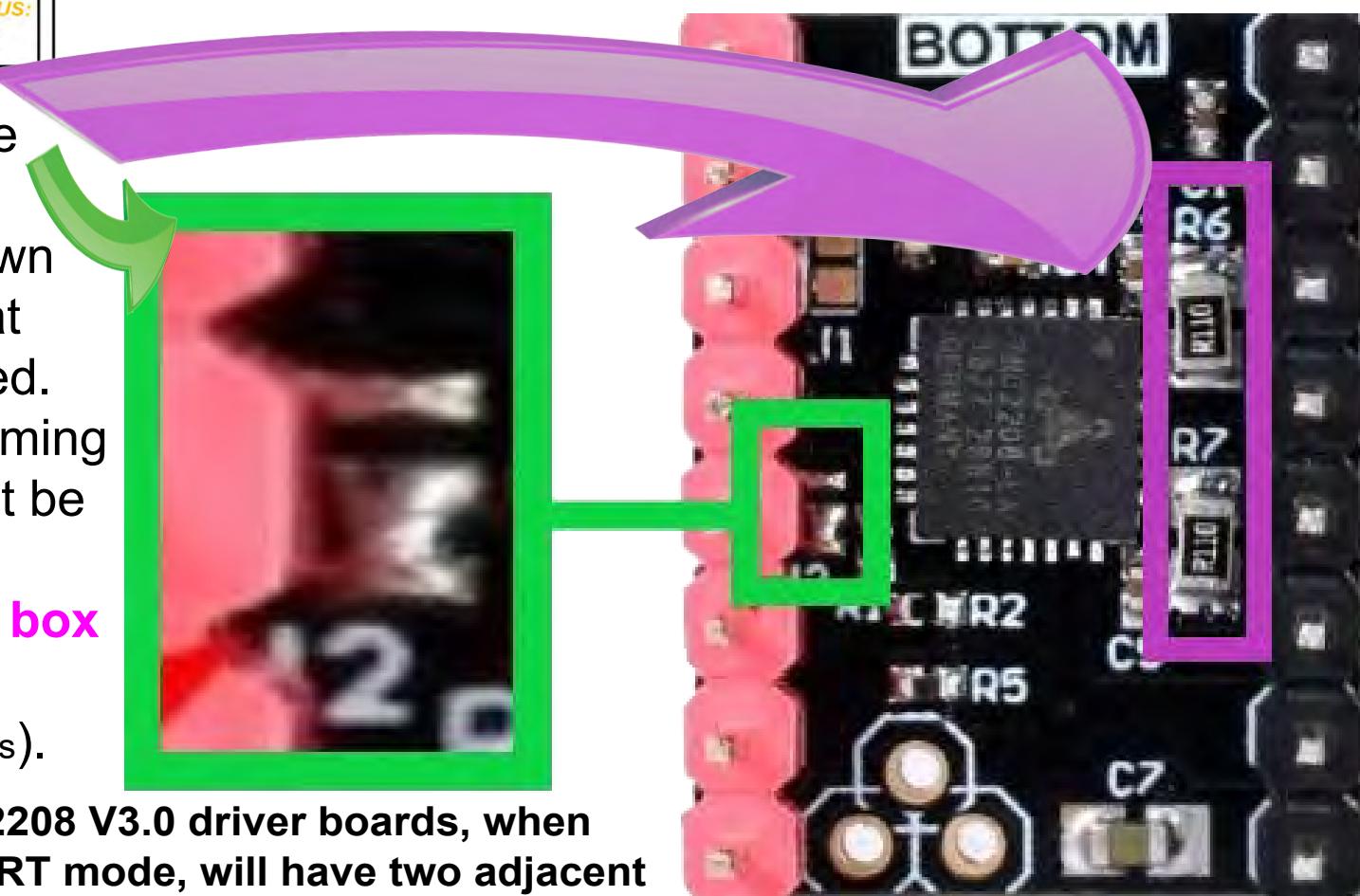
BIQU TMC2208 V3.0**One Time Programming (OTP) Mode**

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.



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).



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**One Time Programming (OTP) Mode**

OTP

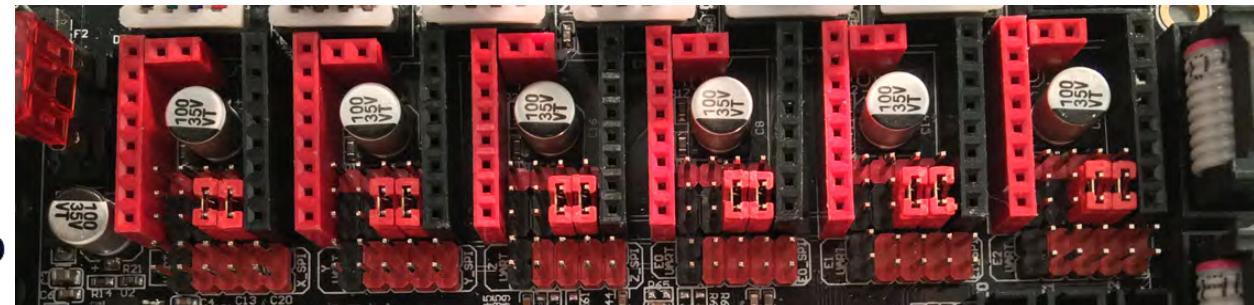
1 / 8

Interpolation:

1 / 256

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	MS1	OA1
PDN	-	MS0	OB1
CLK	-	-	OB2
STEP	-	8 7	VIO
DIR	-	8 7	GND

See [Appendix D](#) for legend

OTP

1 / 2

Interpolation:

1 / 256

SpreadCycle

EN	-	-	VM
MS1	-	-	GND
MS2	-	-	OA2
UART	-	MS1	OA1
PDN	-	MS0	OB1
CLK	-	-	OB2
STEP	-	8 7	VIO
DIR	-	8 -	GND

See [Appendix D](#) for legend

OTP (One Time Programming) Mode**One Time Programming (OTP) Mode**

OTP

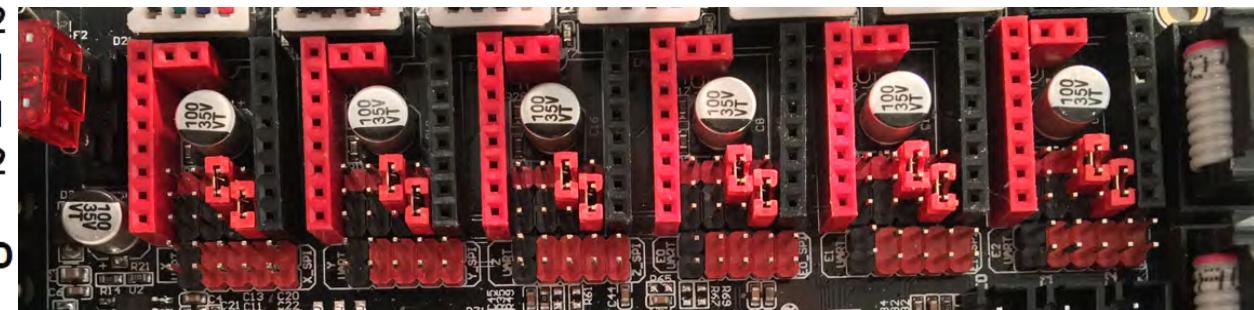
1 / 4

Interpolation:

1 / 256

SpreadCycle

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	MS1	OA1
PDN	-	-	MS0	OB1
CLK	-	8	-	OB2
STEP	-	8	7	VIO
DIR	-	-	7	GND

See [Appendix D](#) for legend

OTP

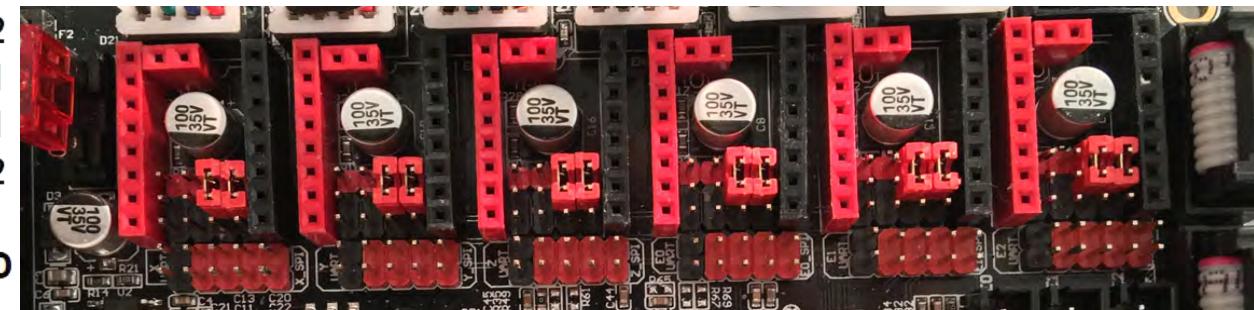
1 / 16

Interpolation:

1 / 256

SpreadCycle

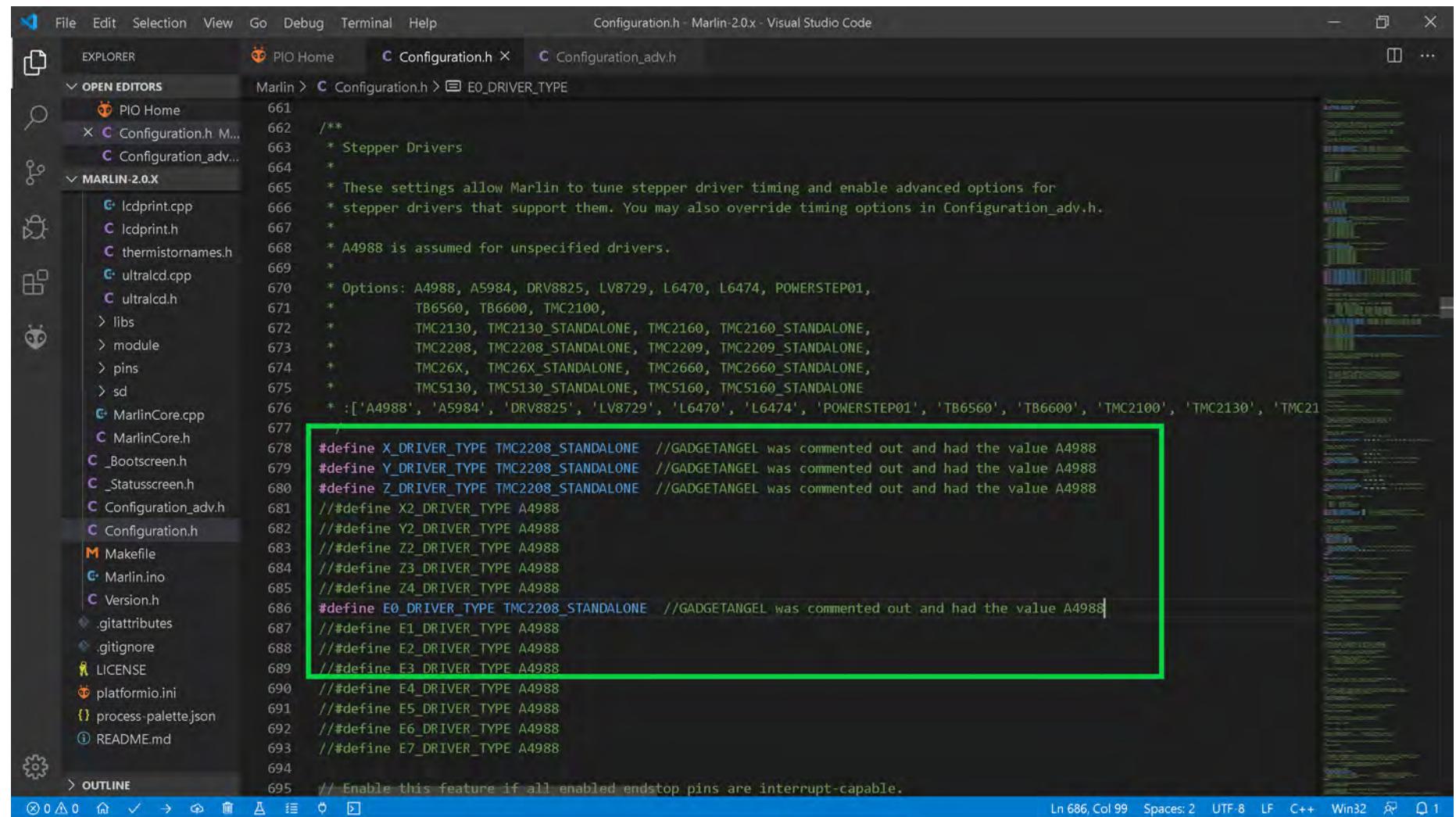
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	OA2
UART	-	-	MS1	OA1
PDN	-	-	MS0	OB1
CLK	-	8	7	OB2
STEP	-	8	7	VIO
DIR	-	-	-	GND

See [Appendix D](#) for legend

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

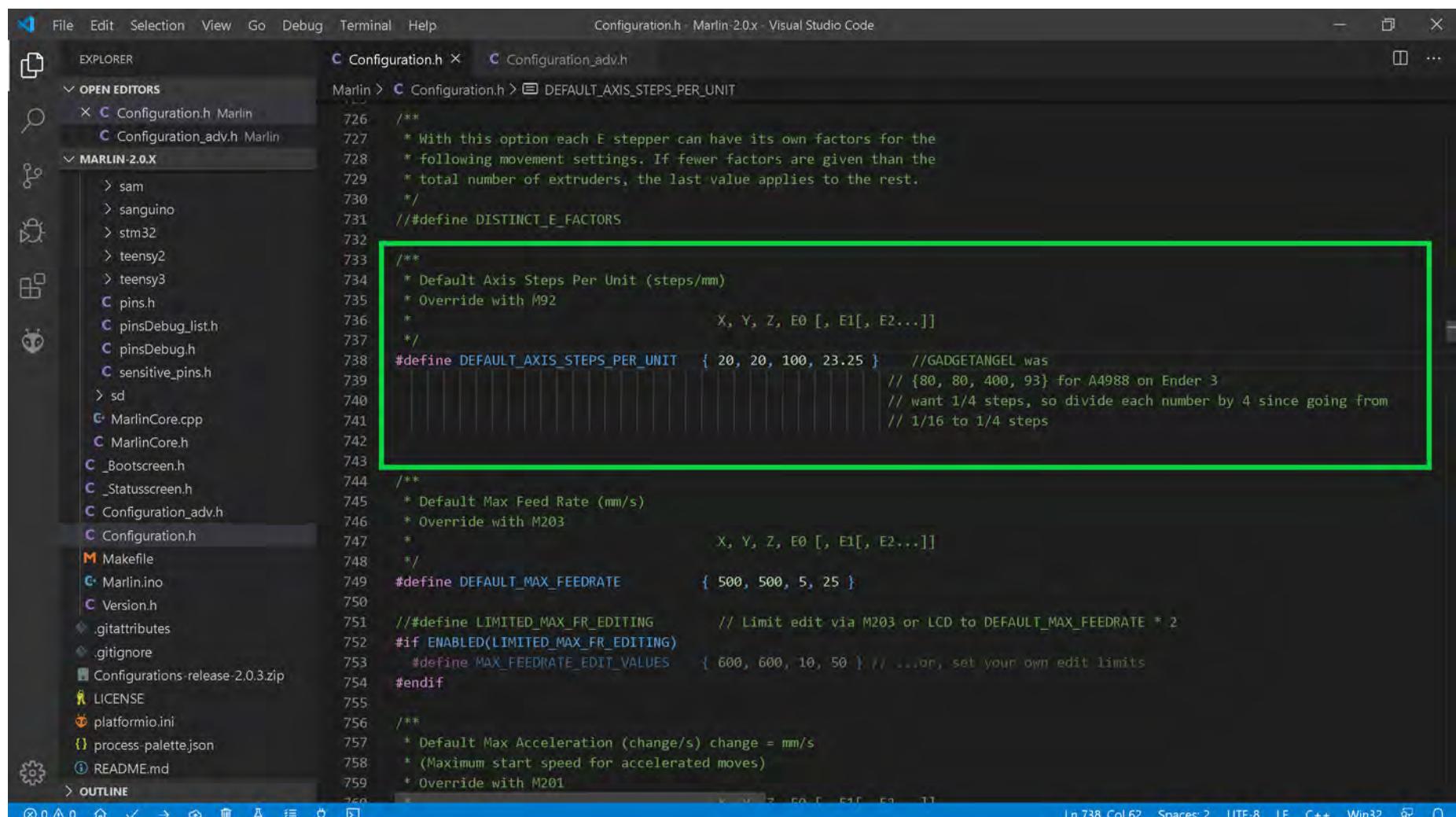
  661 /**
  662 * Stepper Drivers
  663 *
  664 */
  665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  667 *
  668 * A4988 is assumed for unspecified drivers.
  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', '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 }

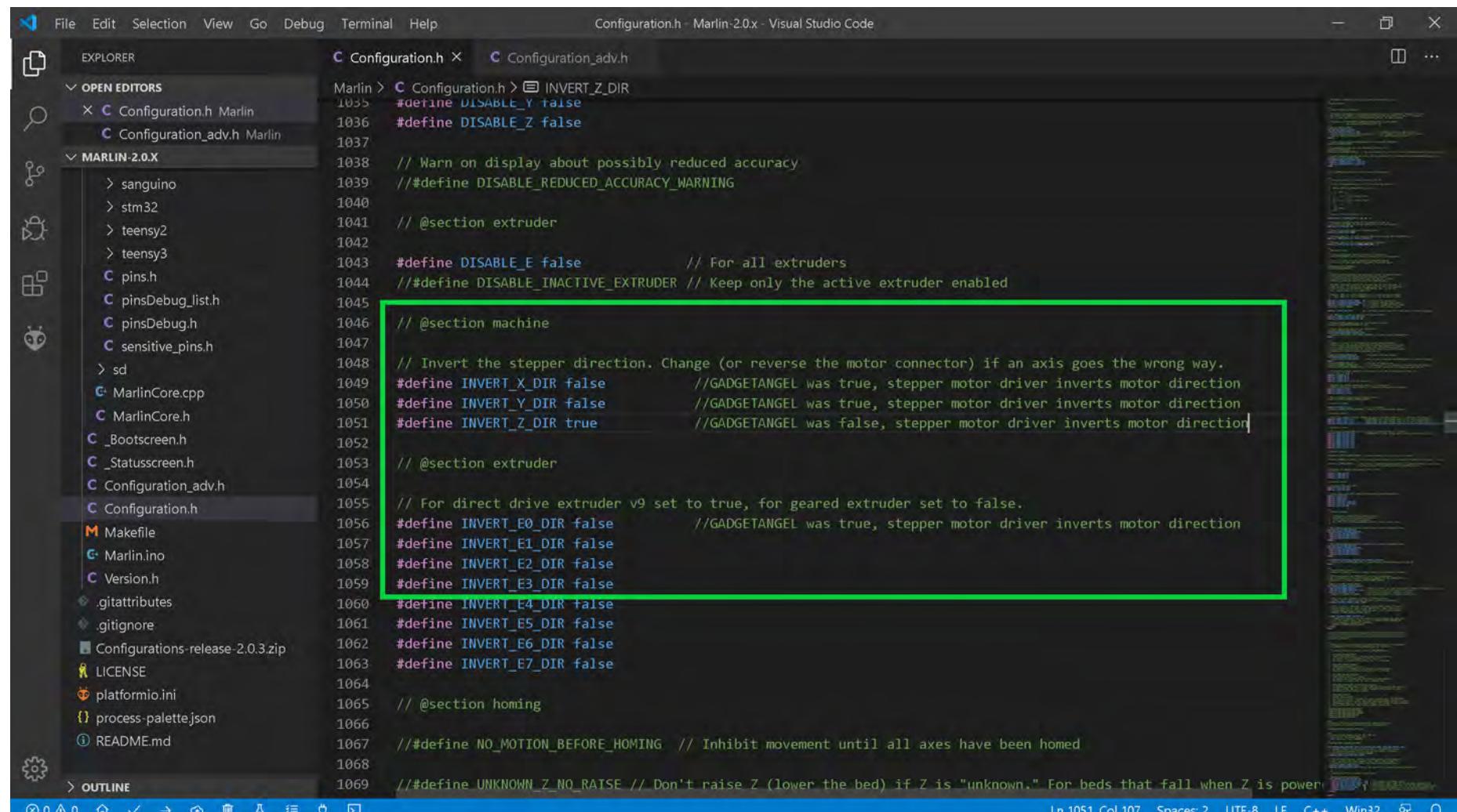
750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
754#endif
755
756 /**
757 * Default Max Acceleration (change/s) change = 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

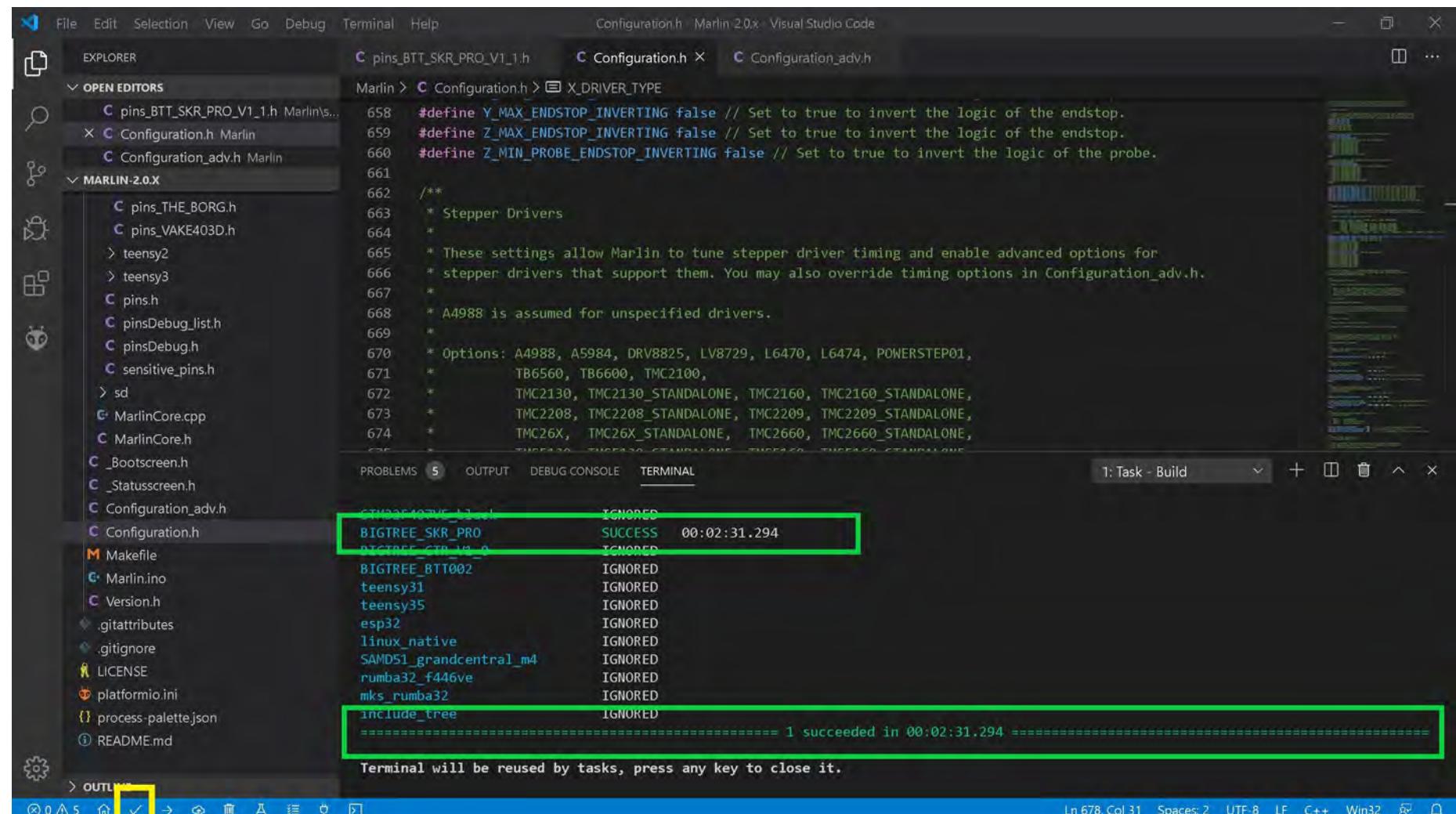
```

A green rectangular box highlights the line of code: '#define INVERT_Z_DIR true'. This indicates that the code has been modified to invert the Z-axis direction, which is necessary for BIQU TMC2208 V3.0 drivers.

- Go to the next page.

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

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



```

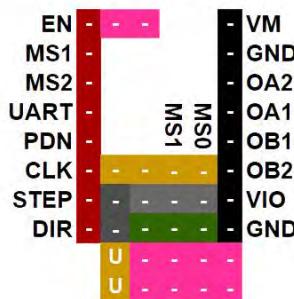
Configuration.h Marlin 2.0.x Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
EXPLORER OPEN EDITORS Marlin > Configuration.h X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin's... Configuration.h Configuration_adv.h
Configuration.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-palettejson
README.md
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build + ☰ ^ ×
CTB2208TVE_V1_1_1 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 [Appendix E](#).

BIQU TMC2208 V3.0**UART Mode**

base of card

UART Mode

fans/heaters/usb side

STATUS:Not Yet
TestedSee [Appendix D](#) for legend

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

 **BIQU®**
TMC2208

UART Mode

Maximum 256
Subdivision35V DC
2A (peak)

**Steps are set inside
of your Firmware**

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

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

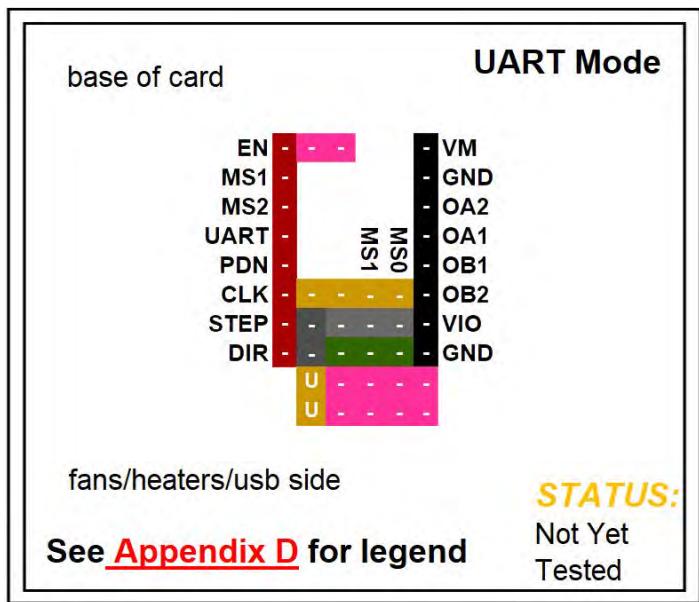
See Appendix B #4. Use 50% to 90%
as shown below:

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

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

See Appendix B #4. Use 50% to 90%
as shown below:

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

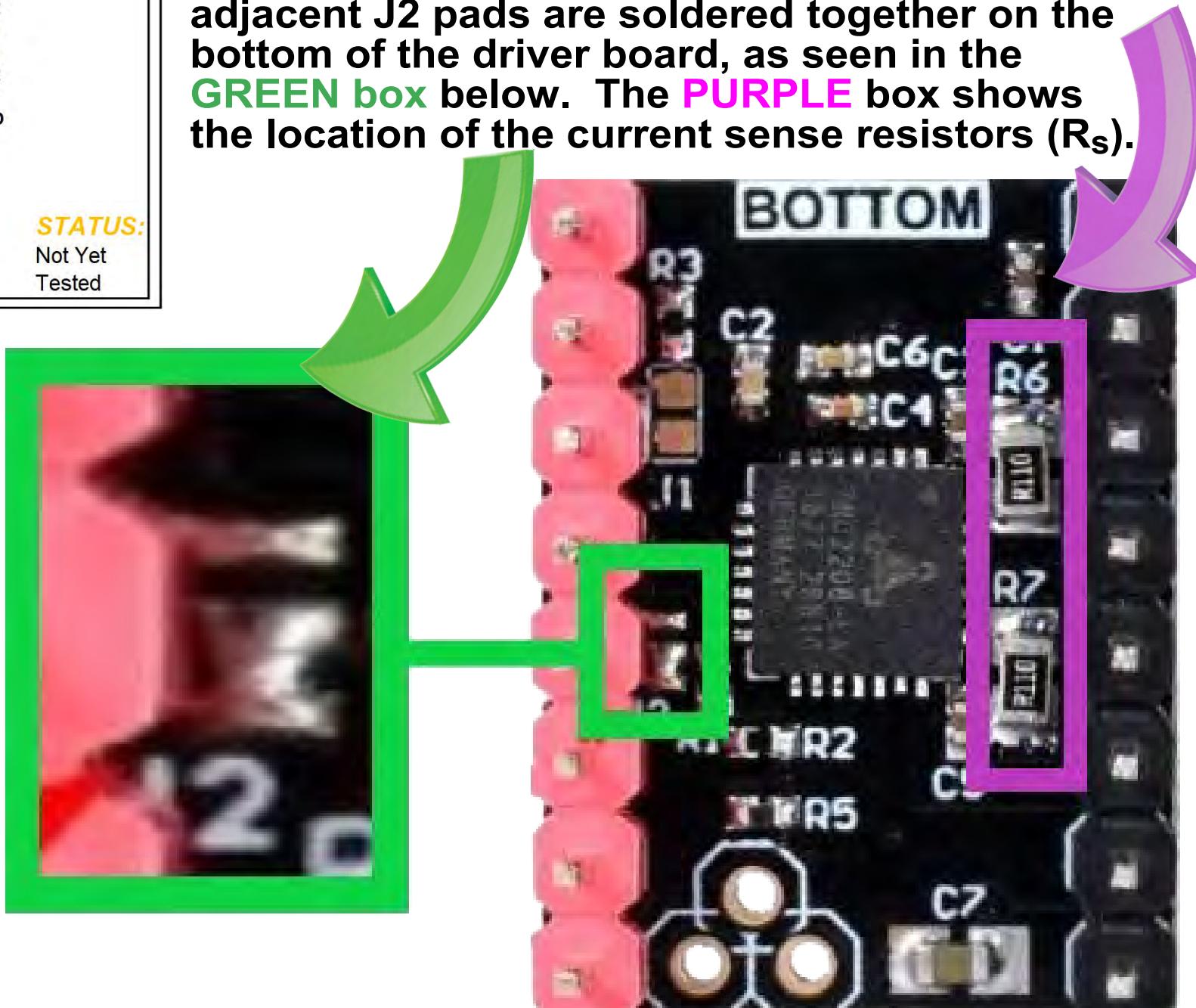


MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

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

BIQU TMC2208 V3.0

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

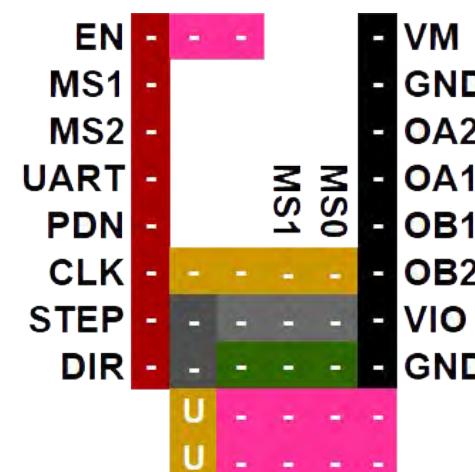
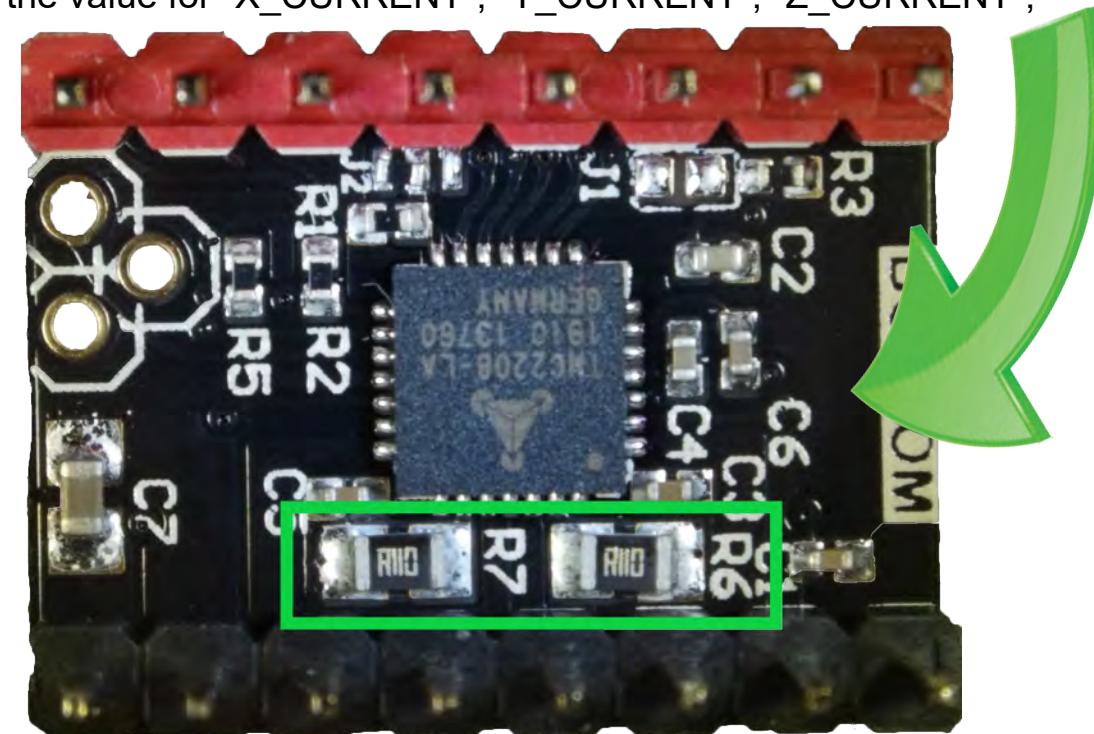
$R_s = R_{100}$ is 0.1 Ohms

$R_s = R_{110}$ is 0.11 Ohms

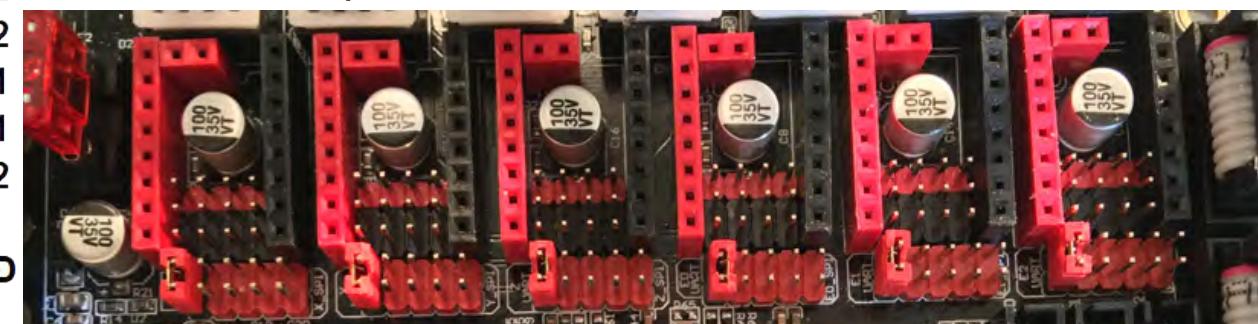
$R_s = R_{150}$ is 0.15 Ohms

$R_s = R_{200}$ is 0.2 Ohms

$R_s = R_{220}$ is 0.22 Ohms



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



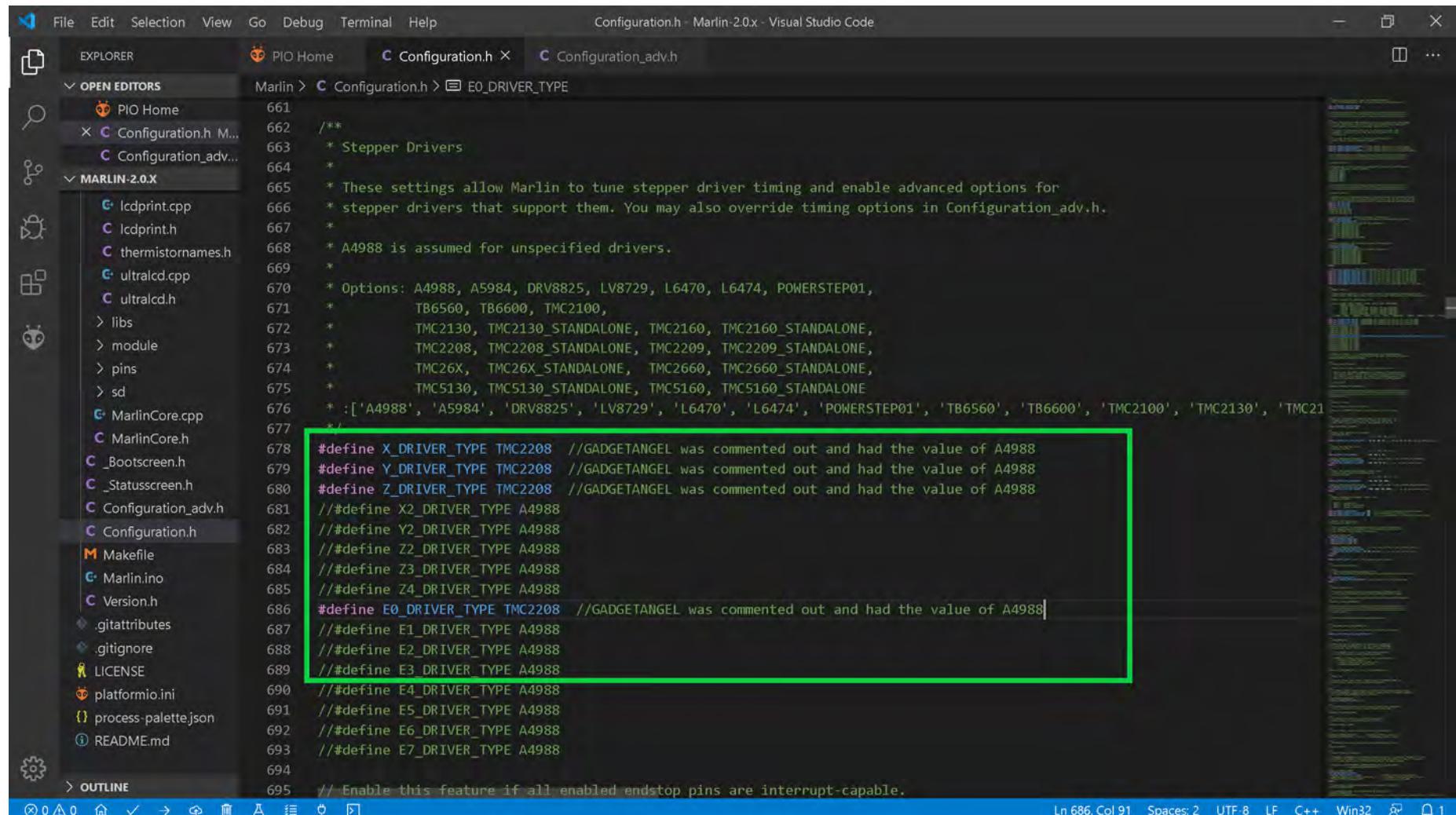
See [Appendix D](#) for legend

UART

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 ("//").



```

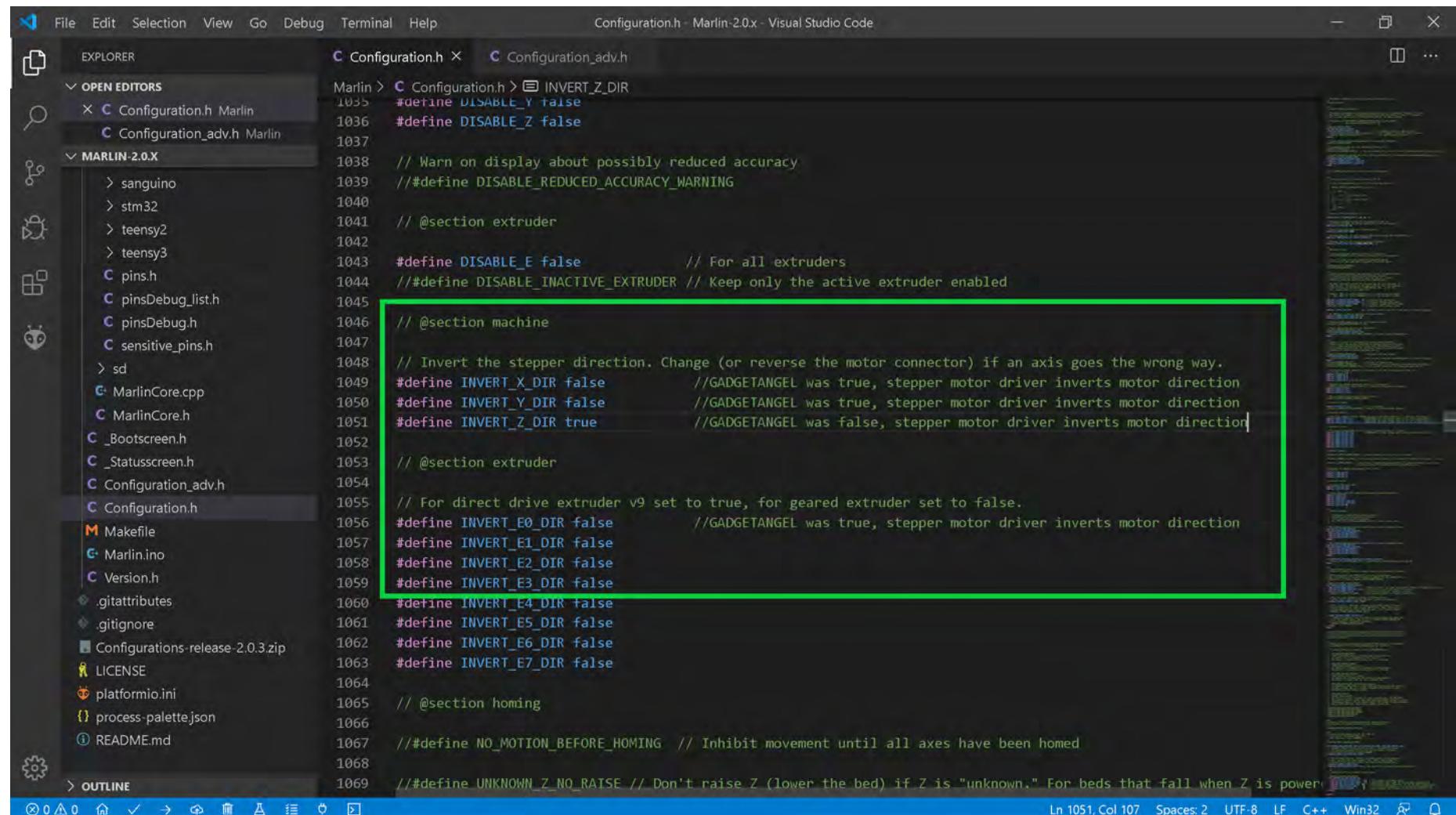
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h M... Configuration_adv.h
PIO Home
Configuration.h M...
Configuration_adv...
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
OUTLINE
Ln 686, Col 91 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', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
676 */
677
#define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define Z3_DRIVER_TYPE A4988
#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
694
// Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



```

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

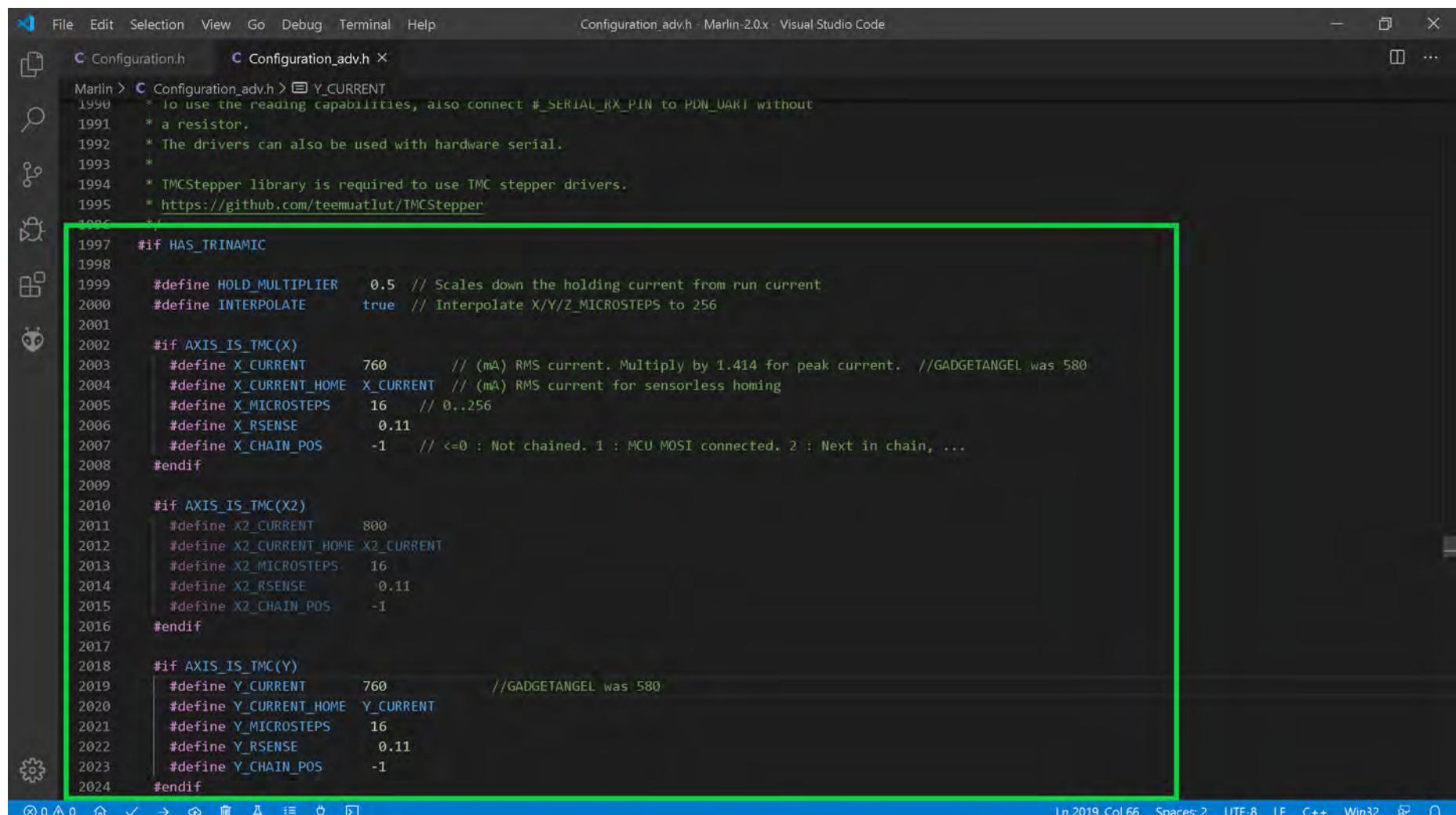
EXPLORER OPEN EDITORS Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037 // Warn on display about possibly reduced accuracy
1038 // #define DISABLE_REDUCED_ACCURACY_WARNING
1039 // @section extruder
1040
1041 #define DISABLE_E false           // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 // GADGETANGEL was true, stepper motor driver inverts motor direction
1048 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1049 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
1051
1052 // @section extruder
1053
1054 // For direct drive extruder v9 set to true, for geared extruder set to false.
1055 #define INVERT_E0_DIR false     // 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
1064 // @section homing
1065
1066 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1067
1068 // #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
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 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".



```

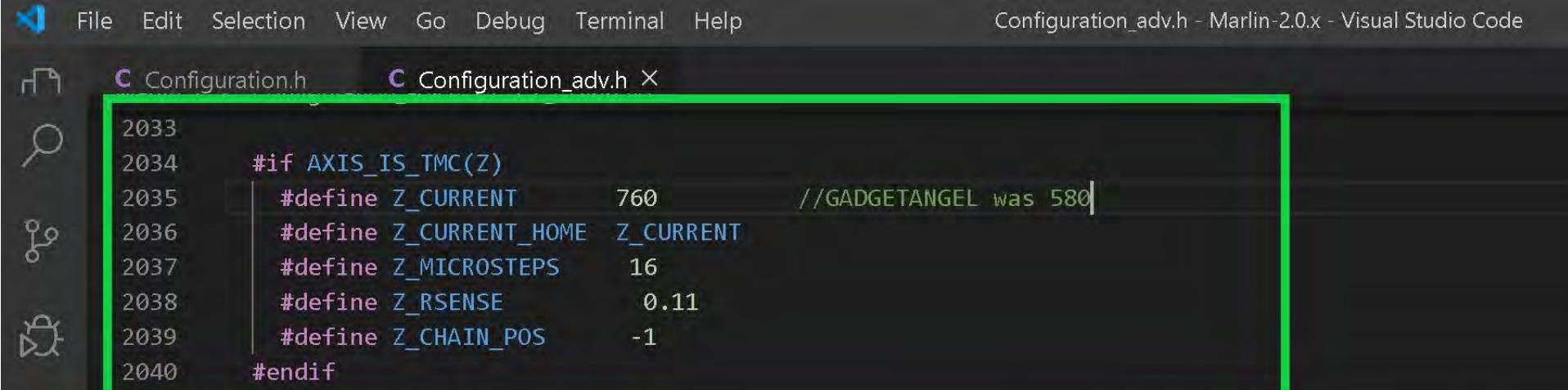
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers 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 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".



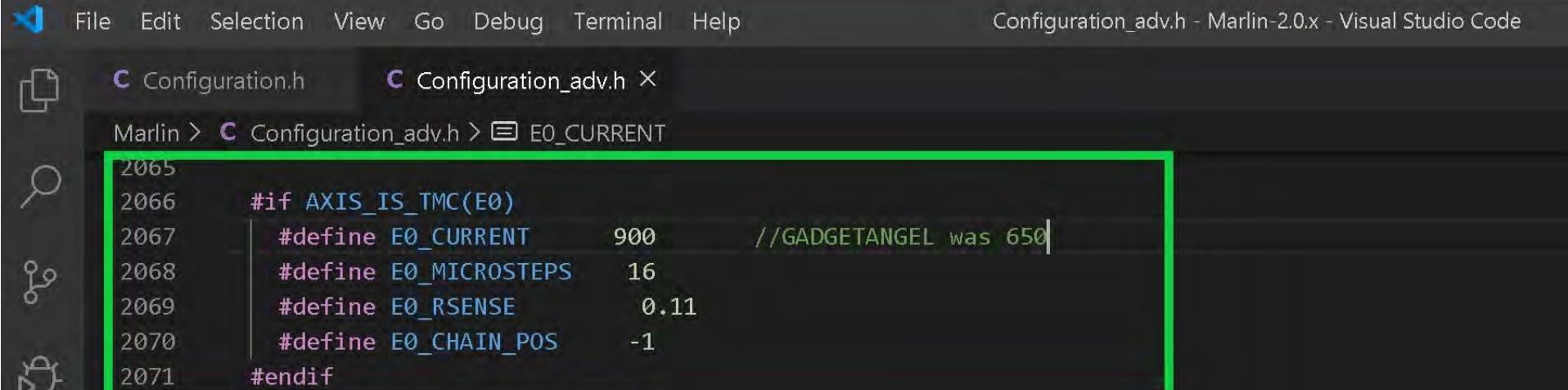
```

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

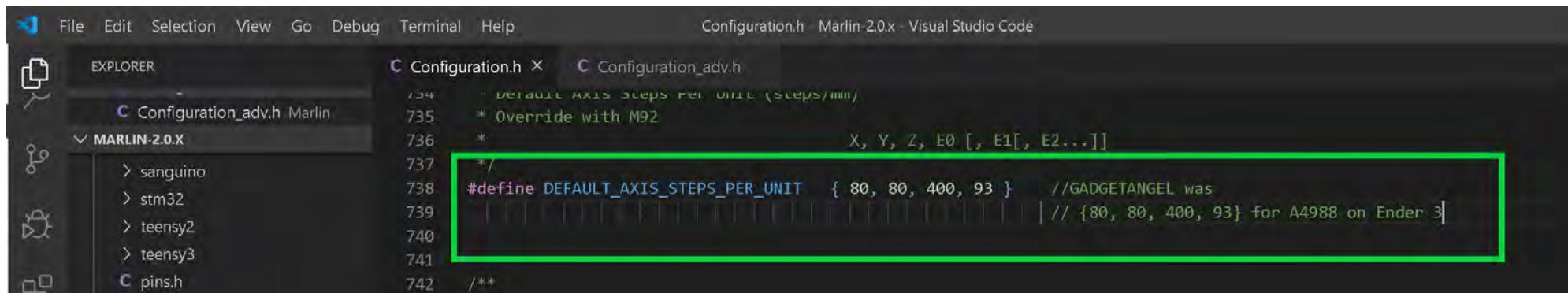
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900           //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- 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

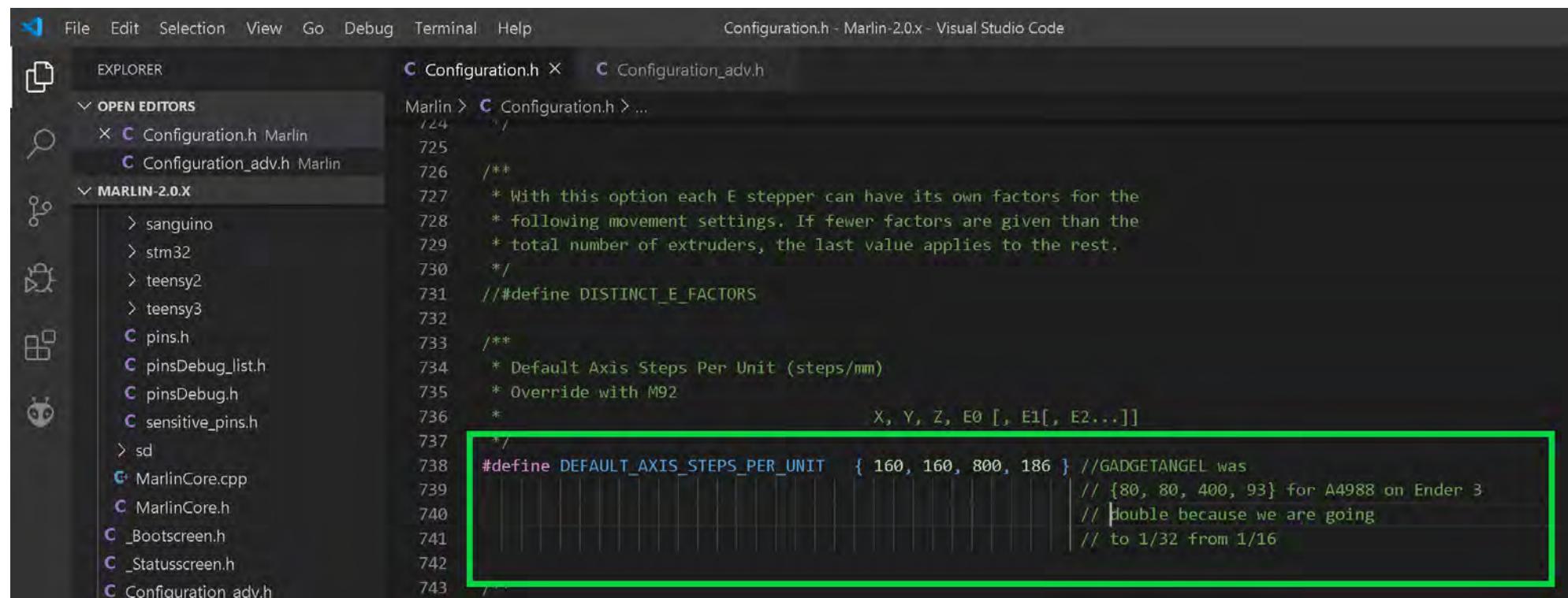
MARLIN-2.0.X

```

 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

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

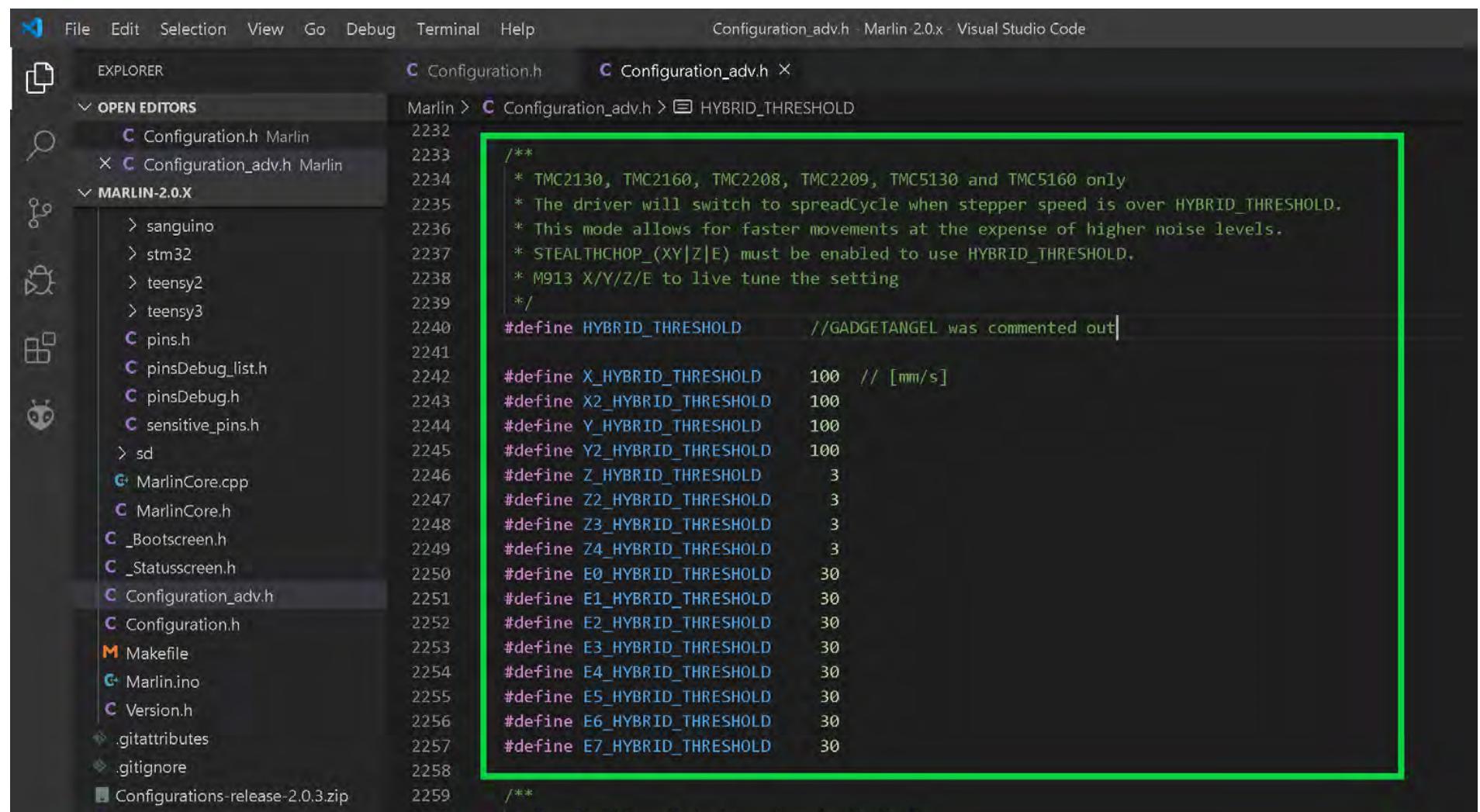
EXPLORER Configuration.h Configuration_adv.h ×
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
> sanguino
> stm32
> teensy2
> teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
> sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_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
2214 */

```

- 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.



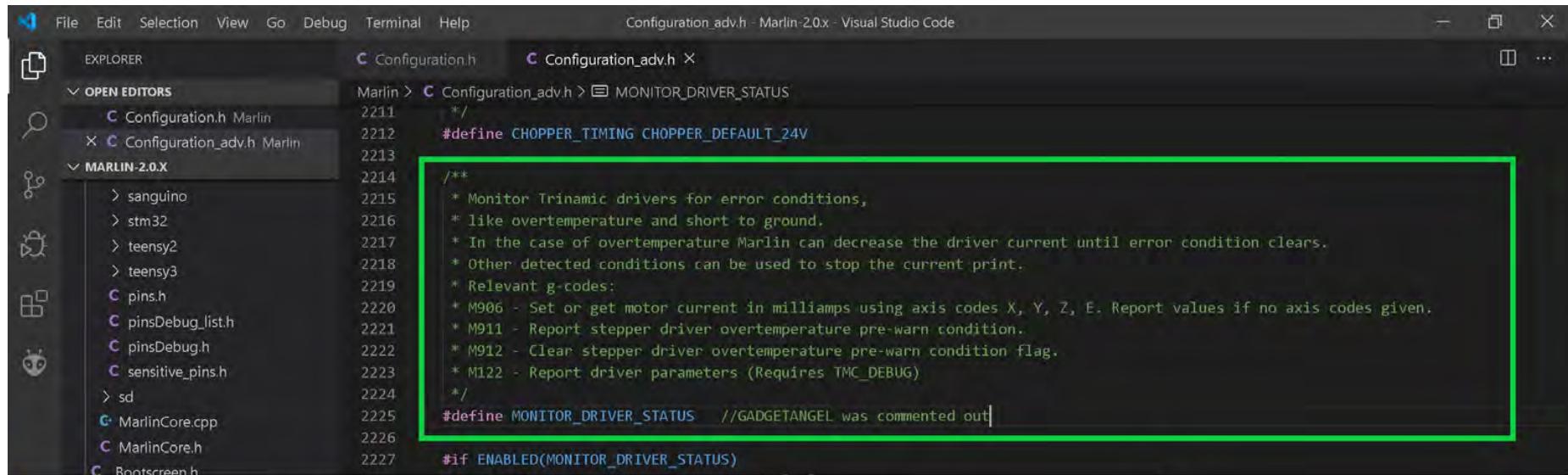
The screenshot shows the Visual Studio Code interface with the 'Configuration_adv.h' file open. The 'EXPLORER' sidebar on the left lists various Marlin 2.0.x files. The main editor window shows the code for 'Configuration_adv.h'. A specific line of code is highlighted with a green box:

```
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 *
```

- Go to the next page.

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 Configuration.h Marlin Configuration_adv.h Marlin

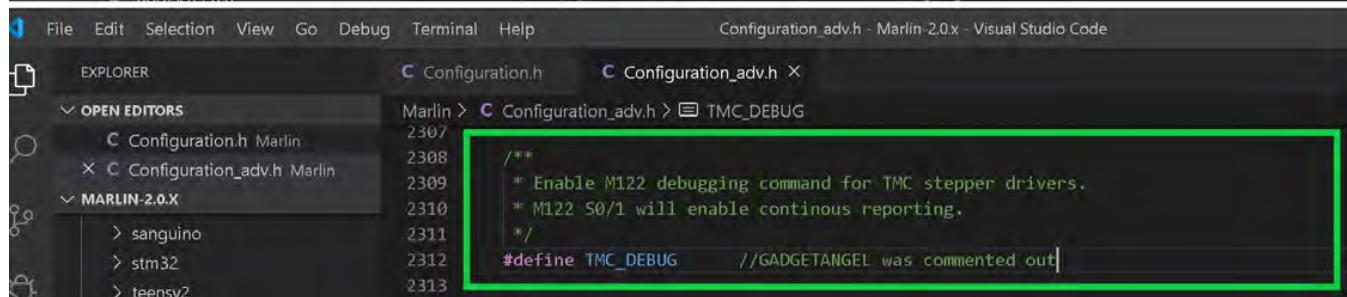
MARLIN-2.0.X

- > 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

```

Marlin > C 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 Configuration.h Marlin Configuration_adv.h Marlin

MARLIN-2.0.X

- > sanguino
- > stm32
- > teensy2

```

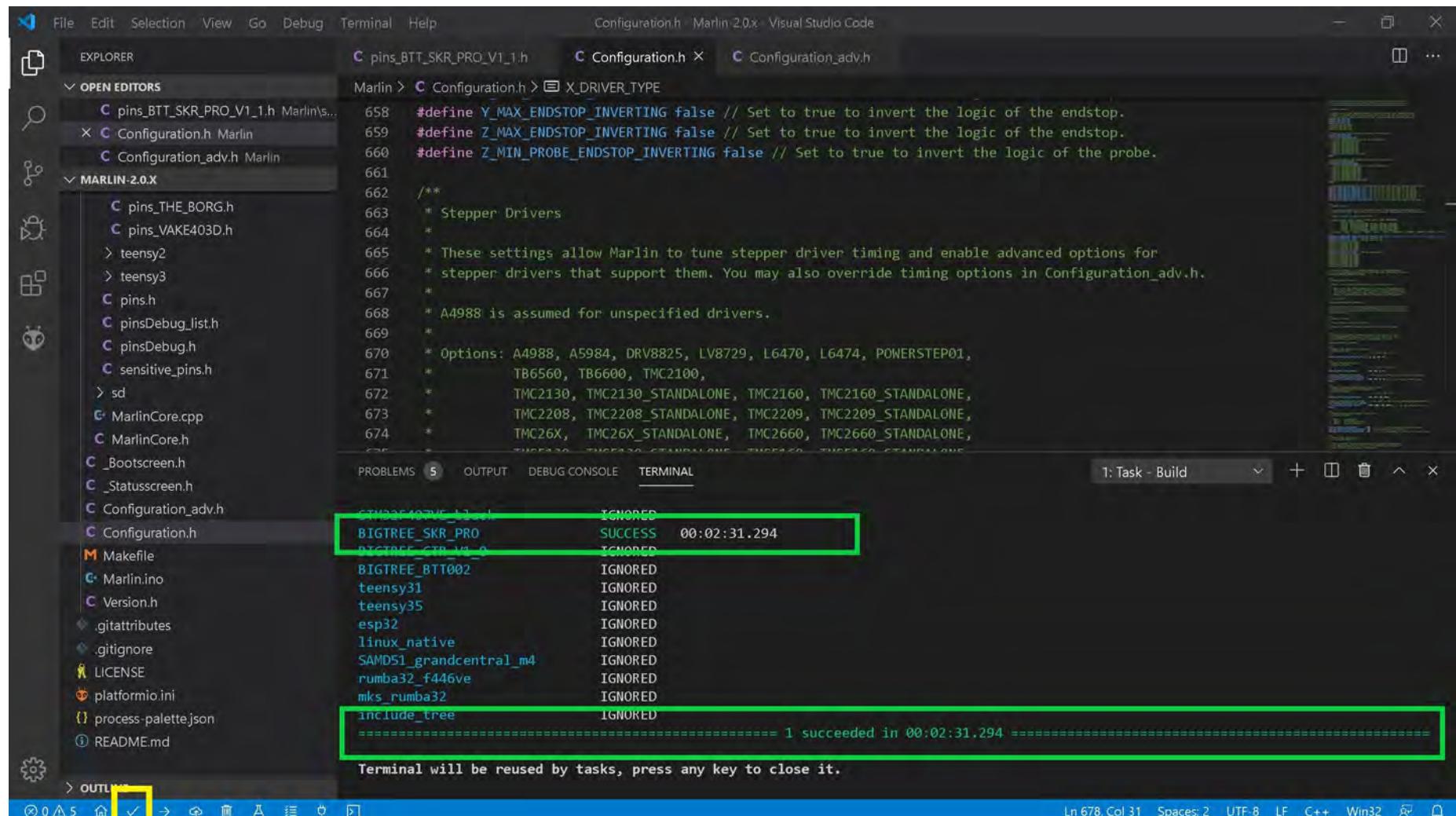
Marlin > C Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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.



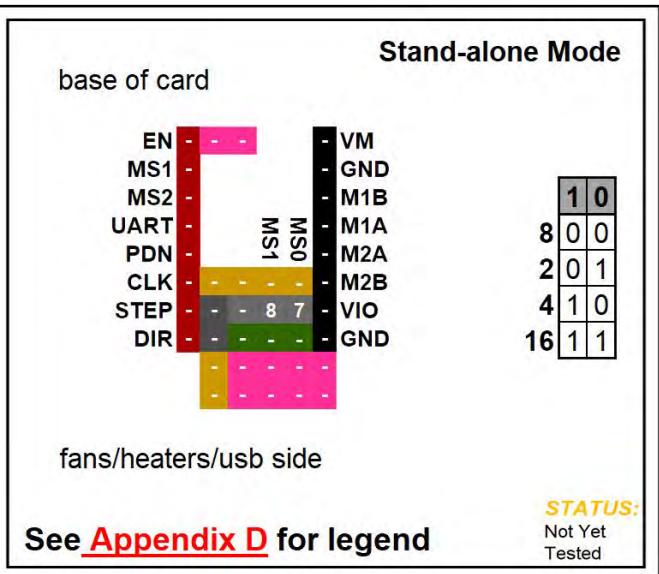
The screenshot shows the Visual Studio Code interface with the following details:

- 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:** Labeled "TERMINAL".
- Build Output:** Shows the compilation results for different boards:

Board	Status	Time
BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
BIGTREE_STM32	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	

 The output concludes with "===== 1 succeeded in 00:02:31.294 =====".
- Bottom Status Bar:** 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 [Appendix E](#).

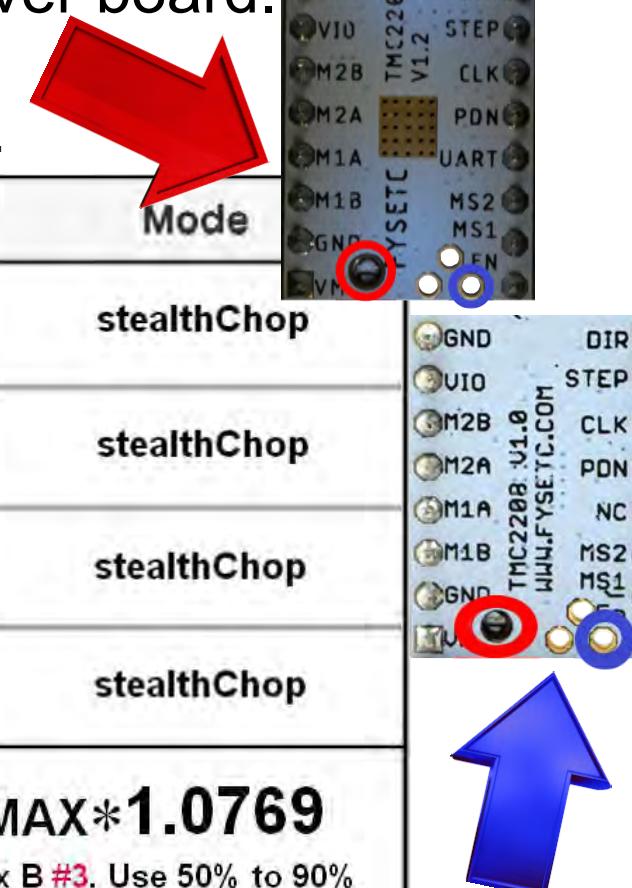


FYSETC TMC2208 V1.2

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.

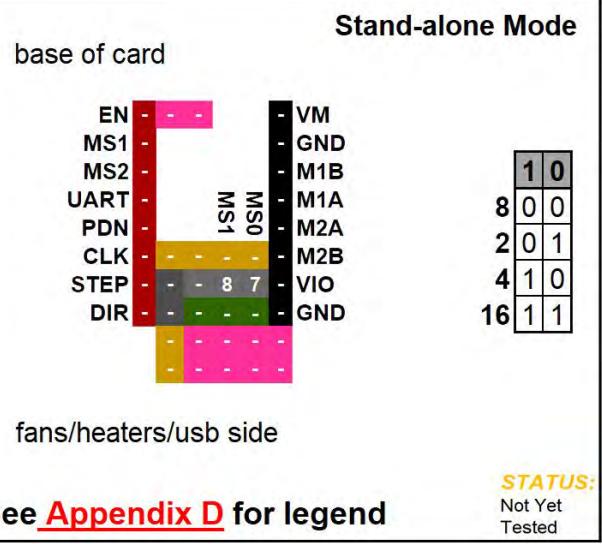
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.



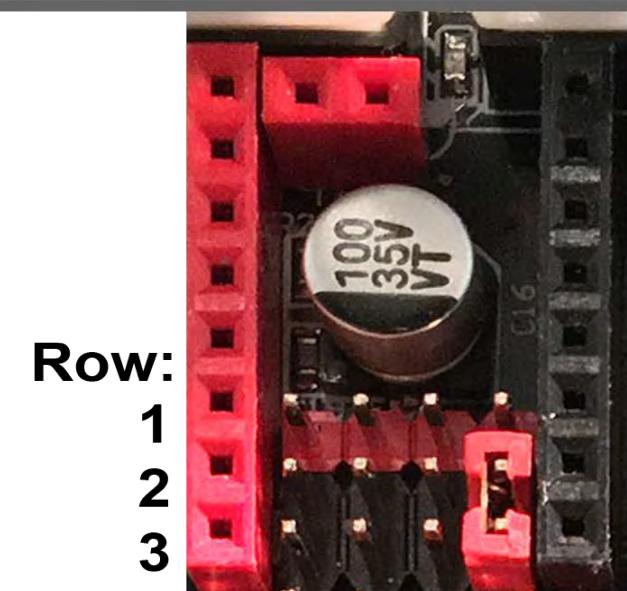
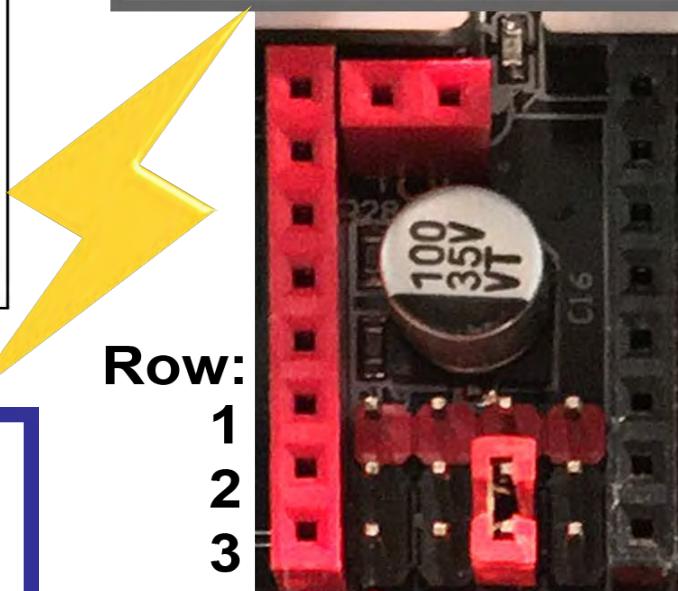
Driver Chip	MS1	MS0	Steps	Interpolation	Mode
FYSETC TMC2208 Stand Alone Mode Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	1 / 8	1 / 256	stealthChop
	Low	High	1 / 2	1 / 256	stealthChop
	High	Low	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	1 / 256	stealthChop

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: $V_{ref} = (I_{MAX} * 1.0769) * 0.90$
--	--	--

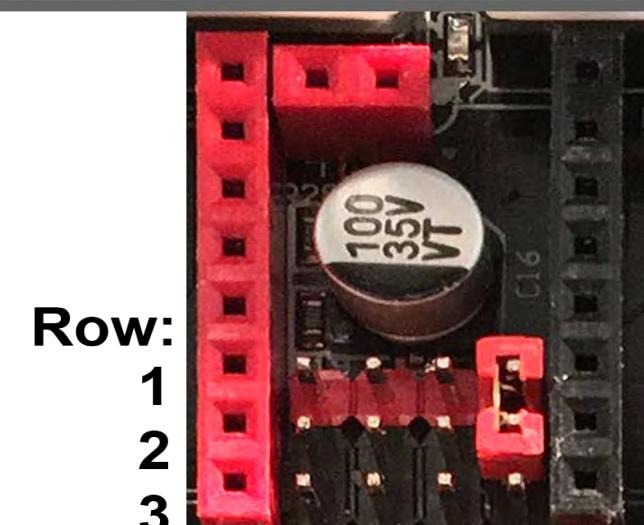
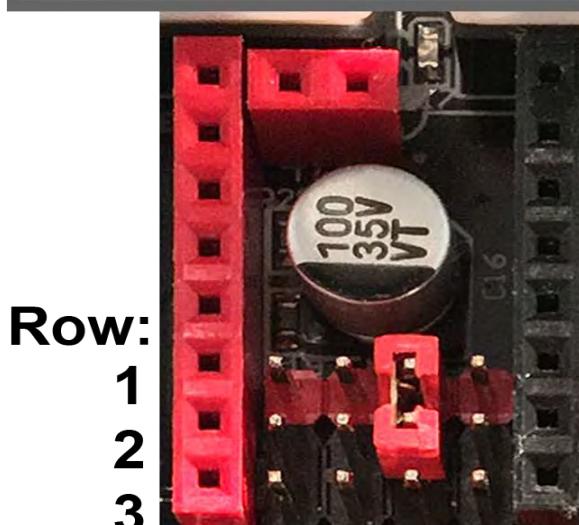
- See next page for the legend that belongs to the above chart.

**FYSETC TMC2208 V1.2****Stand-alone Mode****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

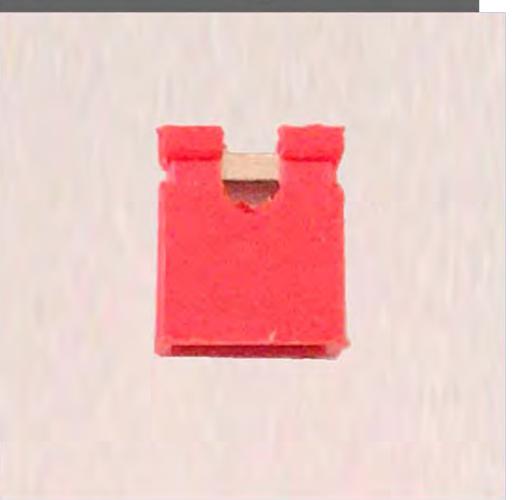
Low ➡ set Jumper between rows 2 and 3



High ➡ set Jumper between rows 1 and 2

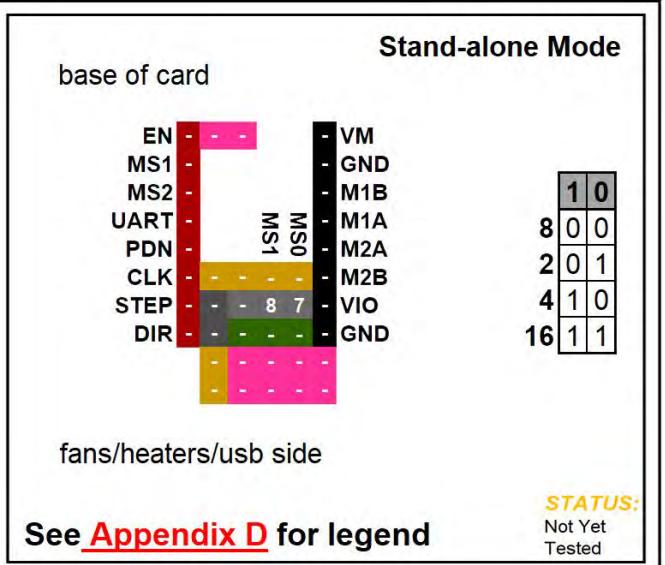


This is a Jumper:



FYSETC TMC2208 V1.2

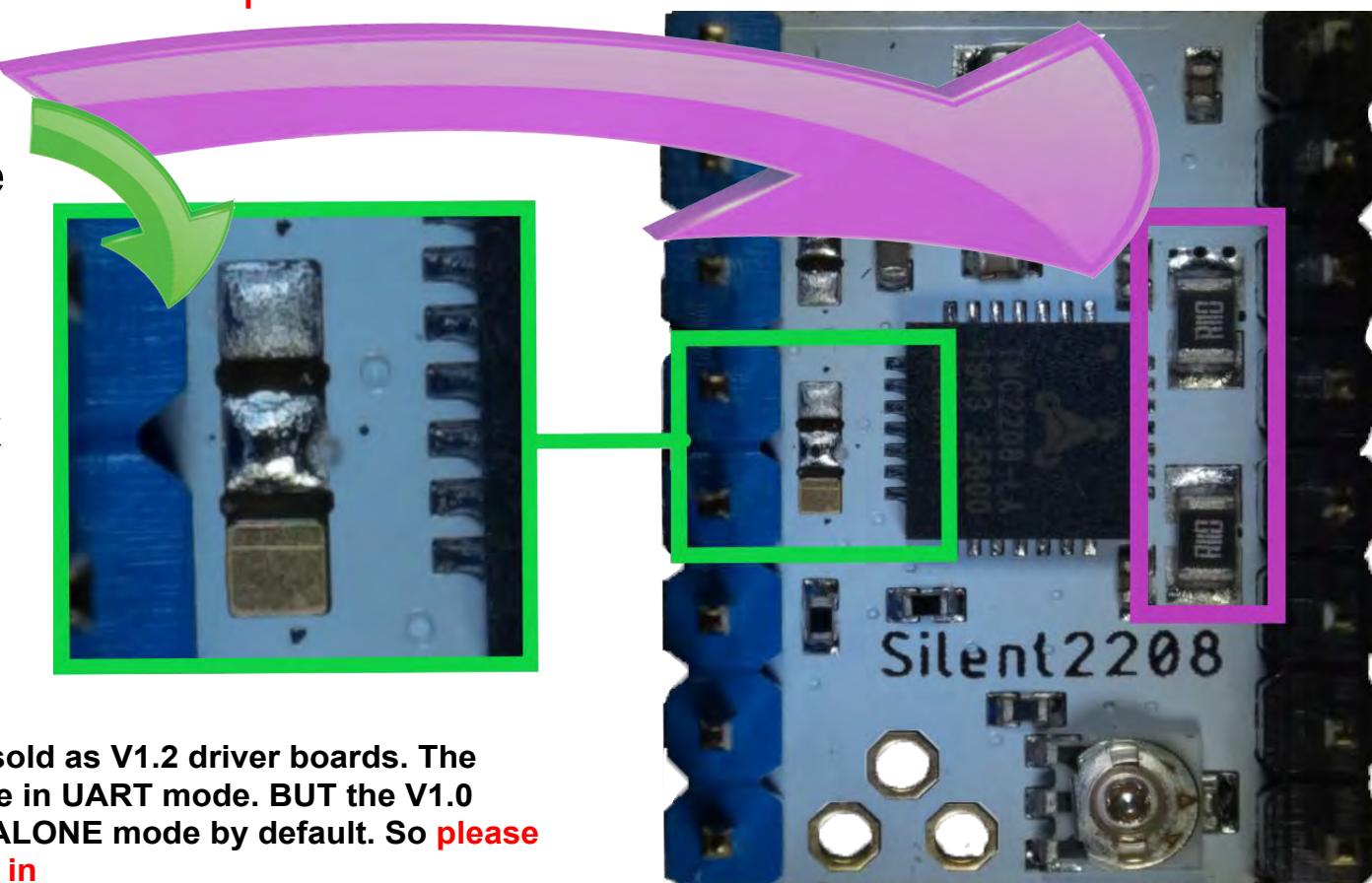
Stand-alone Mode



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

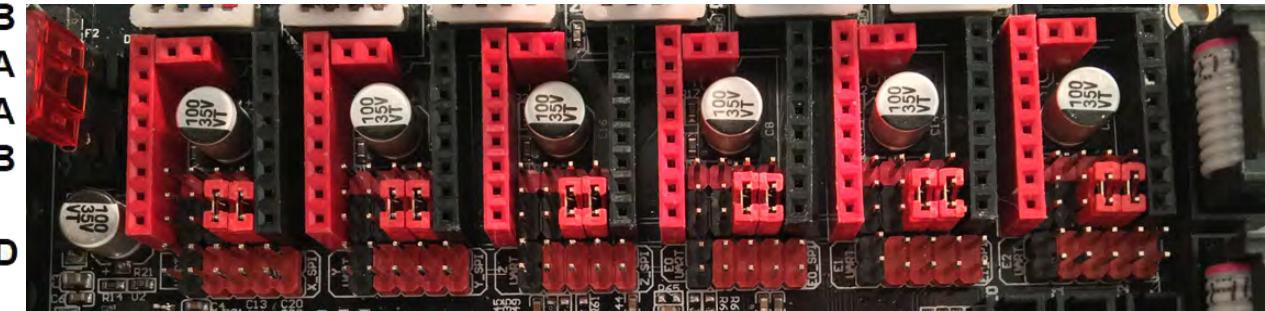
1 / 8

Interpolation:

1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	7	-	M2B
STEP	8	7	-	VIO
DIR	-	8	7	GND

See [Appendix D](#) for legendStand-alone
Mode**1 / 2**

Interpolation:

1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	7	-	M2B
STEP	8	7	-	VIO
DIR	-	8	-	GND

See [Appendix D](#) for legend

Stand-alone Mode

FYSETC TMC2208 V1.2

Stand-alone Mode

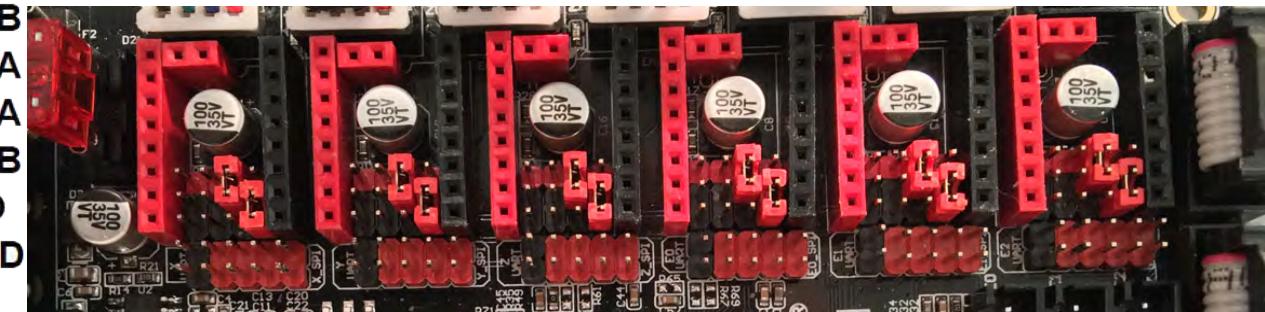
Stand-alone
Mode

1 / 4

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	8	-	M2B
STEP	-	8	7	VIO
DIR	-	-	7	GND



See [Appendix D](#) for legend

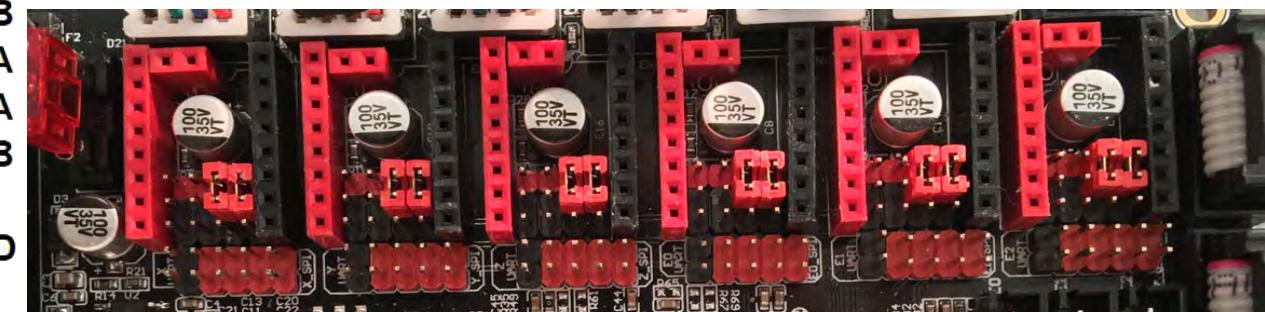
Stand-alone
Mode

1 / 16

Interpolation:
1 / 256

StealthChop

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	8	7	M2B
STEP	-	8	7	VIO
DIR	-	-	-	GND

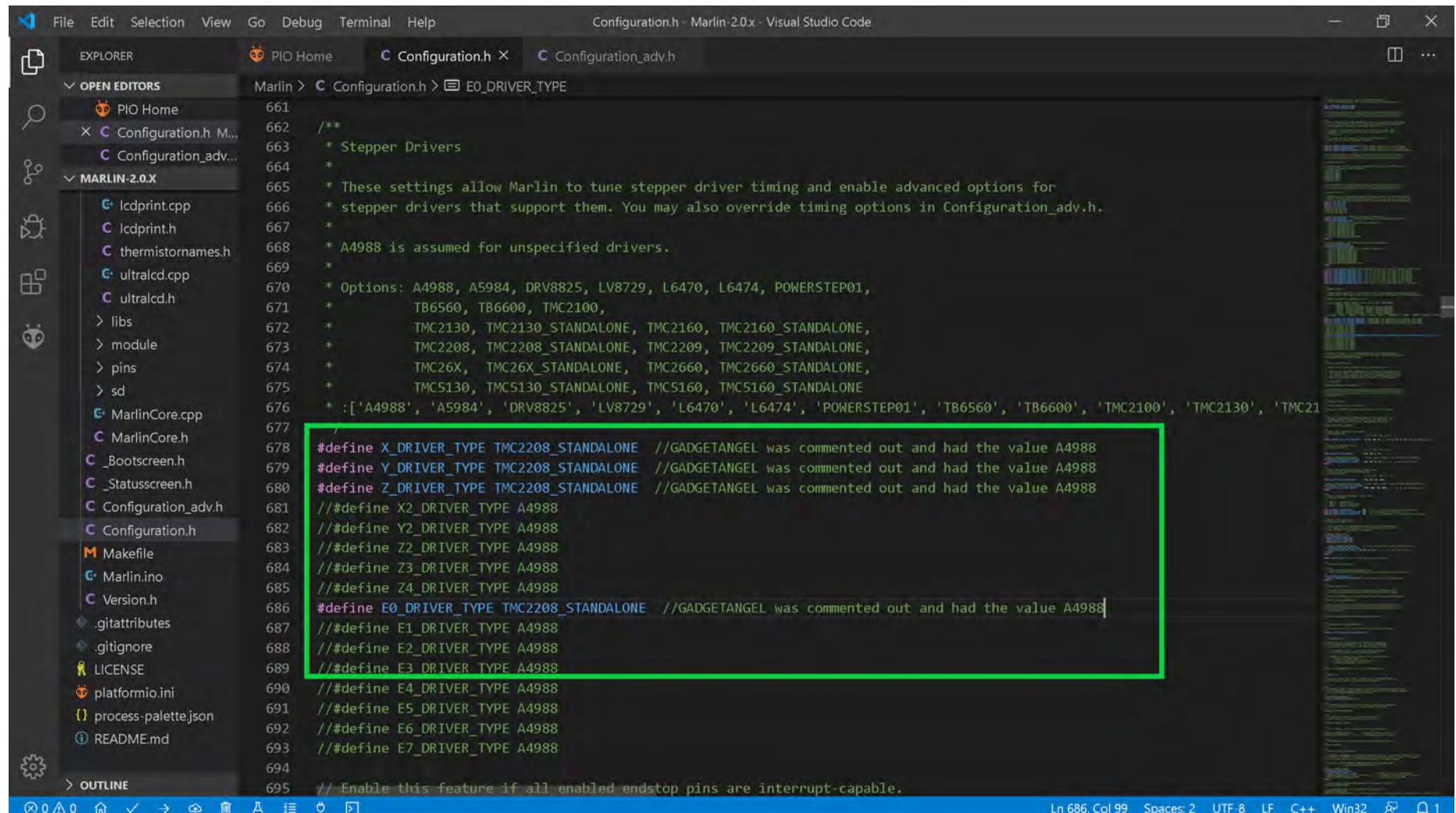


See [Appendix D](#) for legend

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', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
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.

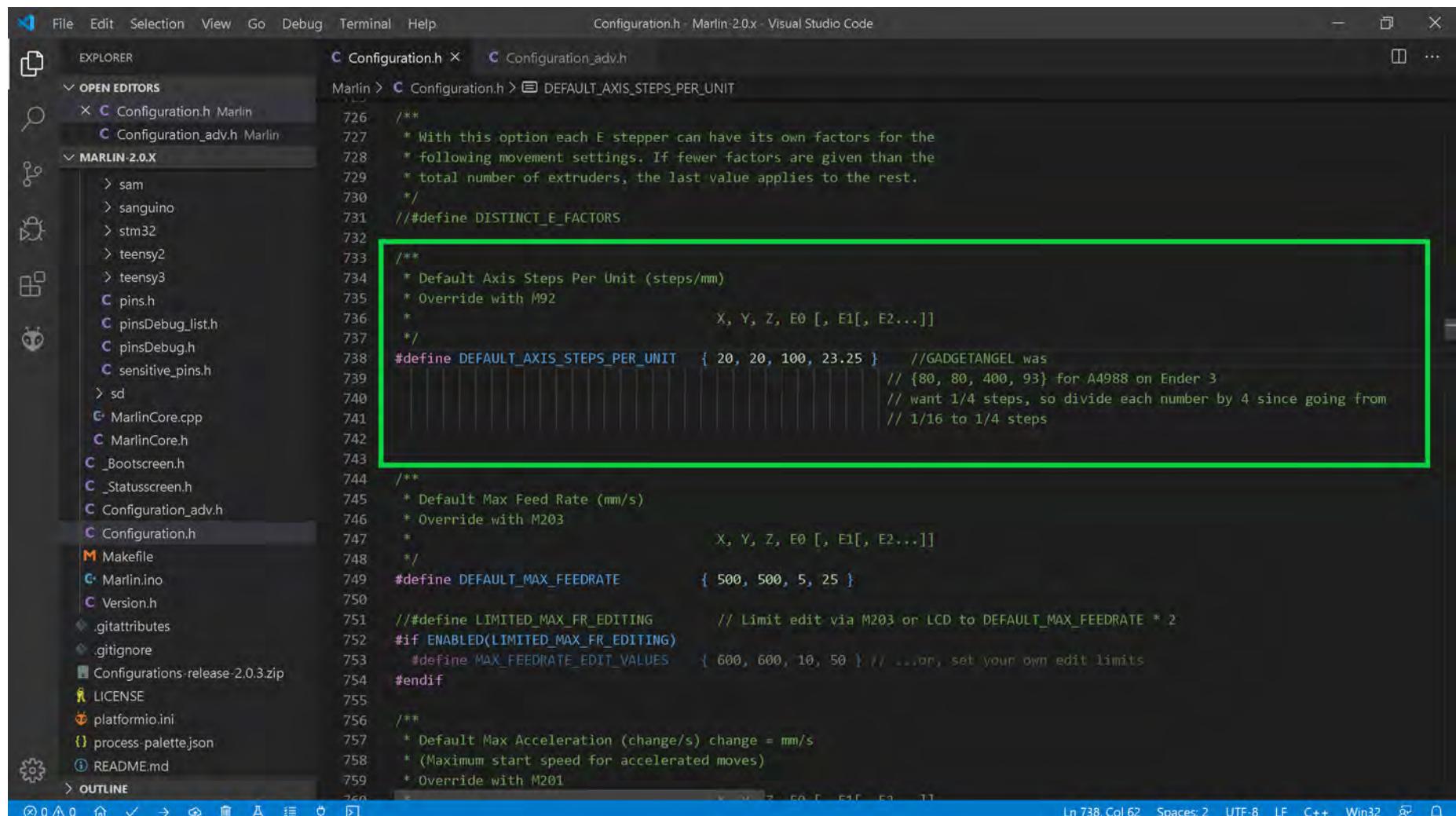
```

A green rectangular box highlights the driver type definitions for the X, Y, Z, and E0 axes, specifically the lines starting with `#define`. Below these, there are additional lines starting with `//##define`, which are commented out. The status bar at the bottom of the code editor shows the line number (Ln 686), column (Col 99), spaces (Spaces: 2), and encoding (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

- 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 various files and folders, including Configuration.h, Configuration_adv.h, and several pins and core files. 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 */

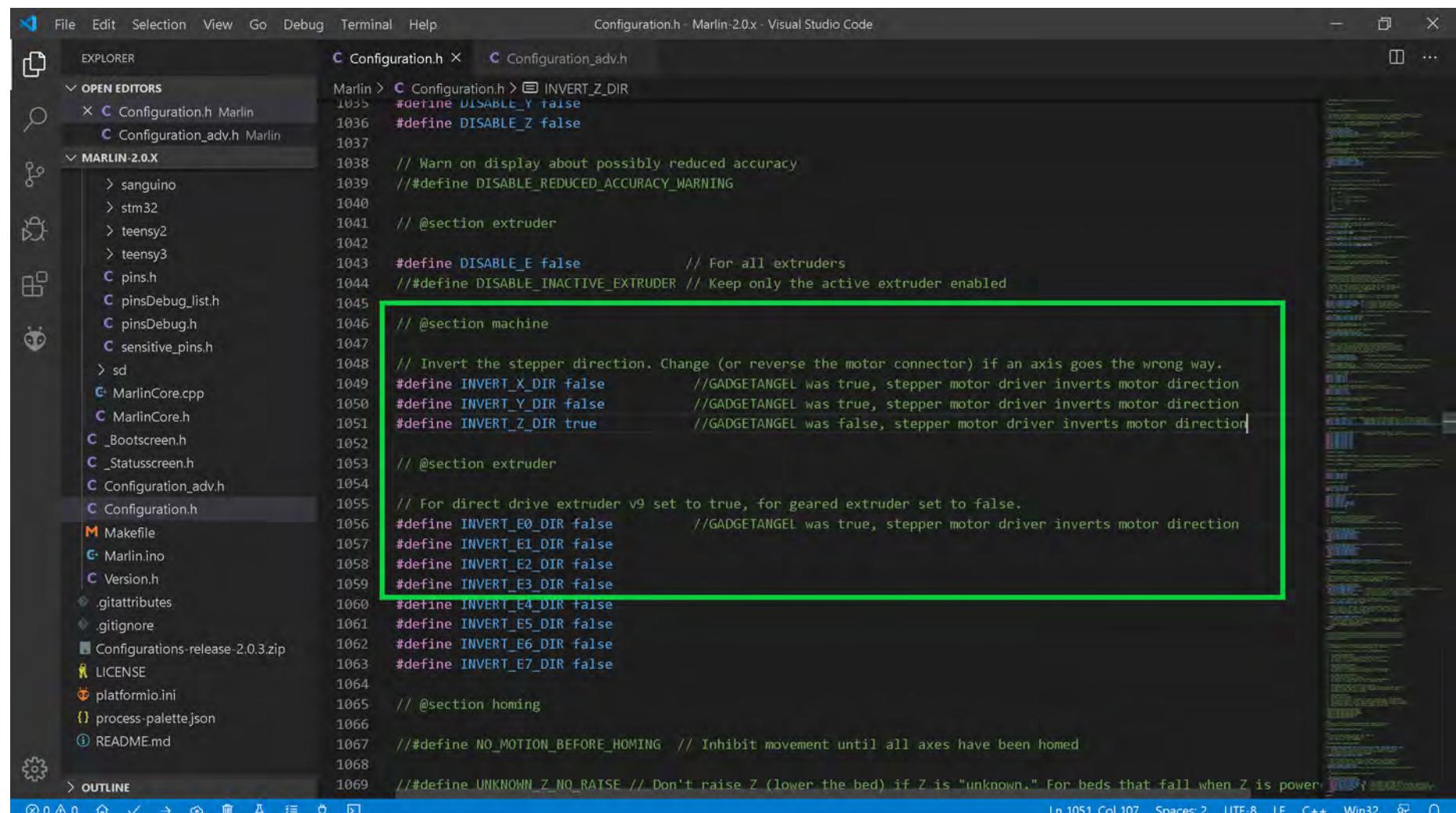
```

The status bar at the bottom indicates the current line (Ln 738), column (Col 62), and other settings like 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

- 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 sections:

```

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

```

A green rectangular box highlights the following lines of code in the `Machine` section:

```

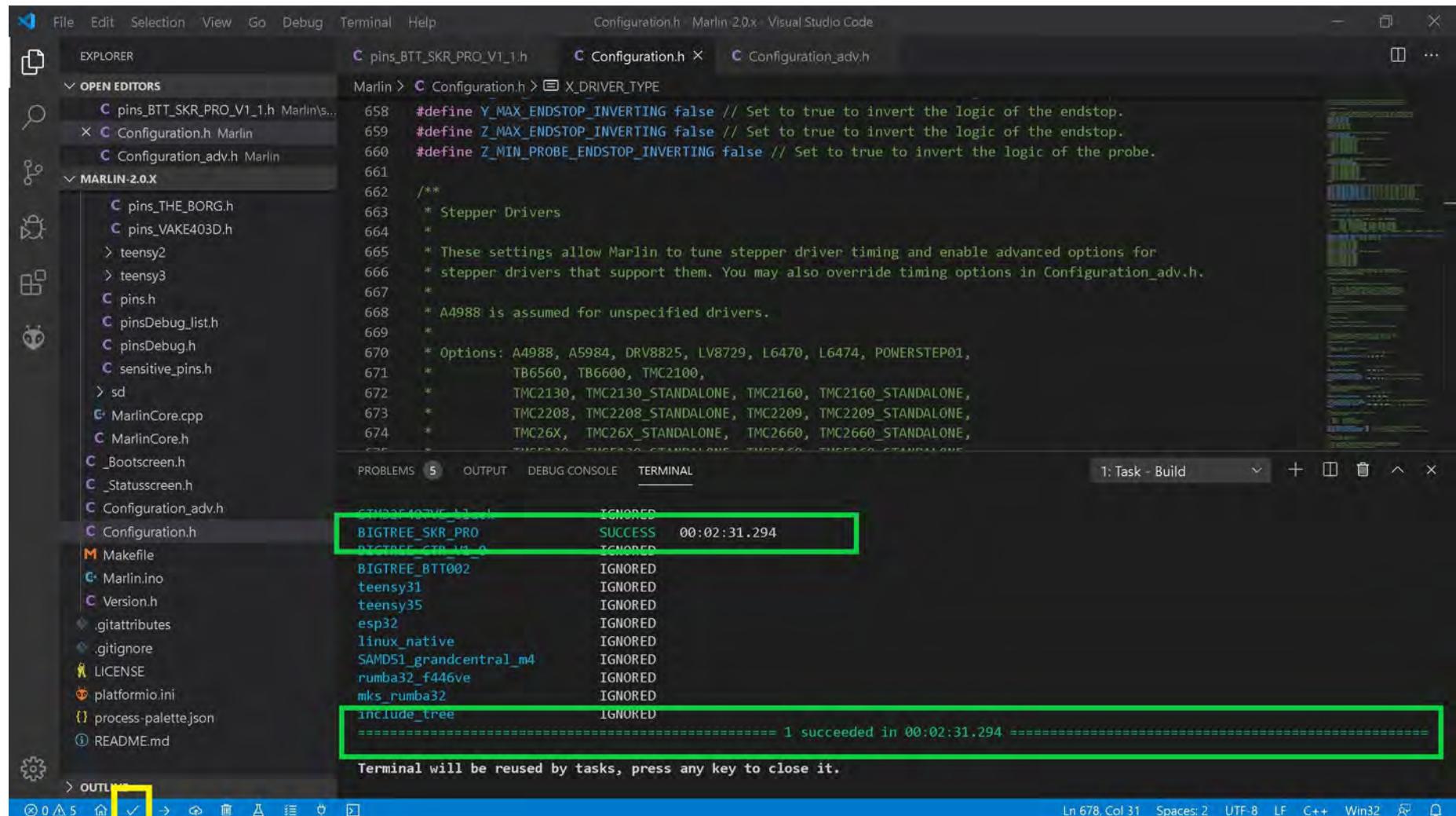
#define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction

```

- 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.



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 * 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 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

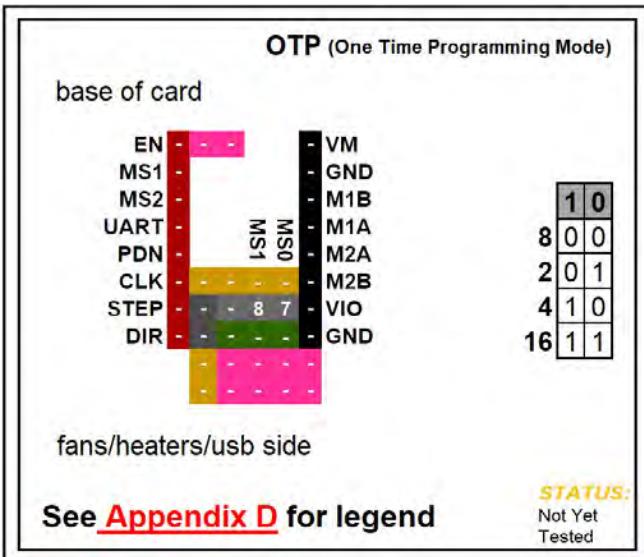
1: Task - Build

BIGTREE_SKR_PRO	SUCCESS	00:02:31.294
include_tree	IGNORED	

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 [Appendix E](#).

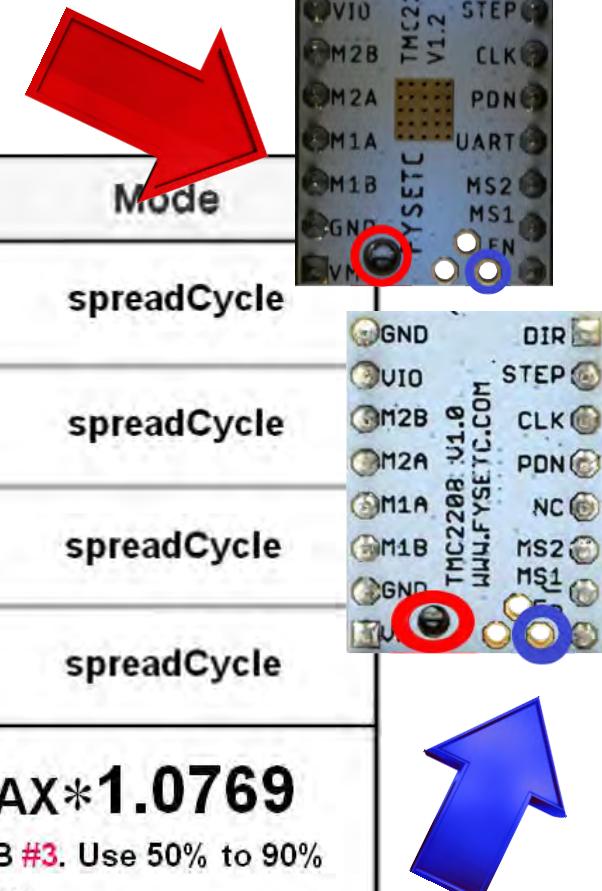


FYSETC TMC2208 V1.2

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.

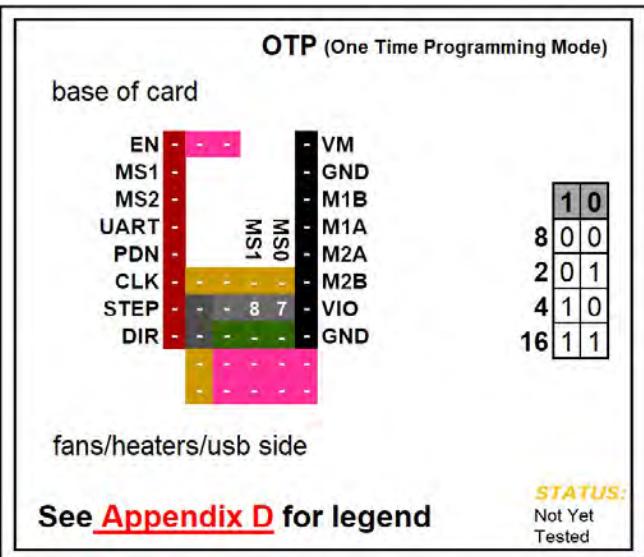
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.



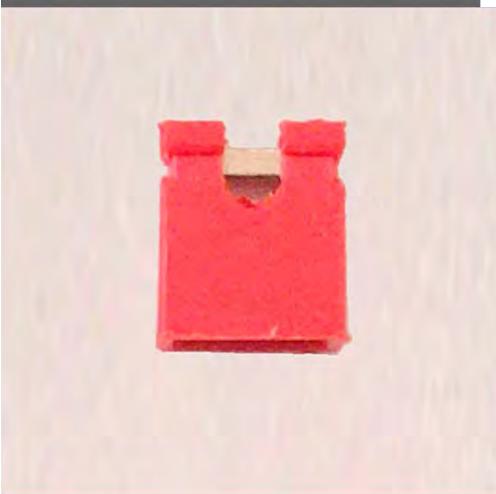
Driver Chip	MS1	MS0	Steps	Interpolation	Mode
FYSETC TMC2208 OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	1 / 8	1 / 256	spreadCycle
	Low	High	1 / 2	1 / 256	spreadCycle
	High	Low	1 / 4	1 / 256	spreadCycle
	High	High	1 / 16	1 / 256	spreadCycle

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: $V_{ref} = (I_{MAX} * 1.0769) * 0.90$
--	--	--

- See next page for the legend that belongs to the above chart.



This is a
Jumper:

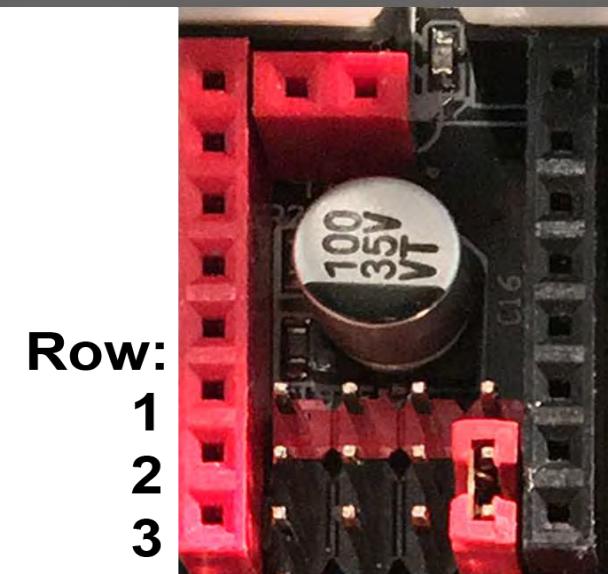
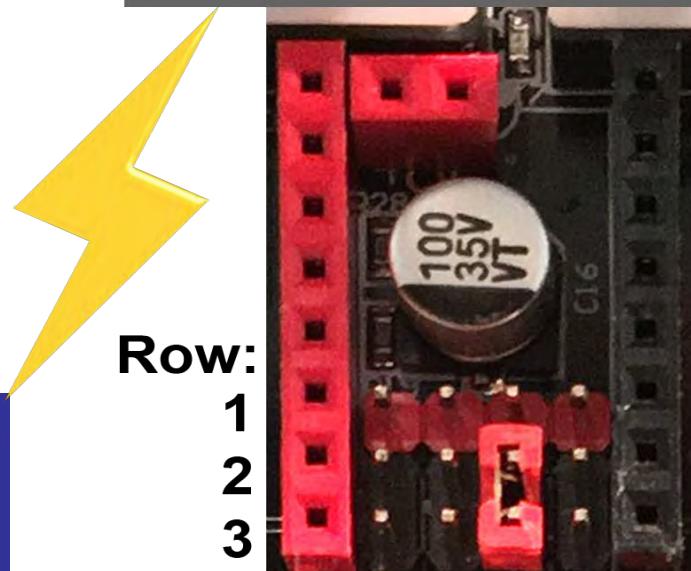


FYSETC TMC2208 V1.2

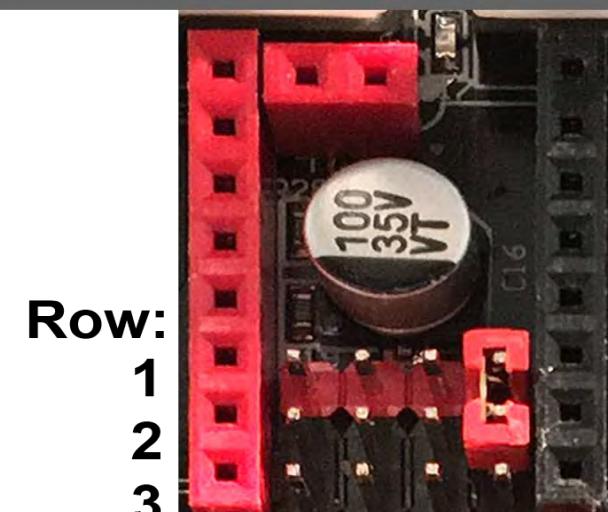
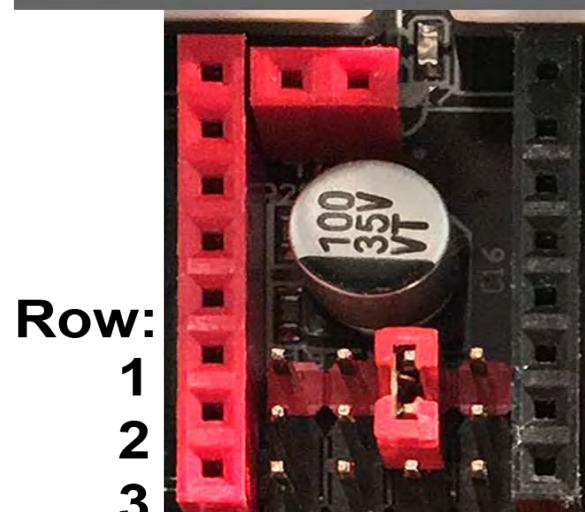
One Time Programming (OTP) Mode

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows
2 and 3



High ➡ set Jumper between rows
1 and 2

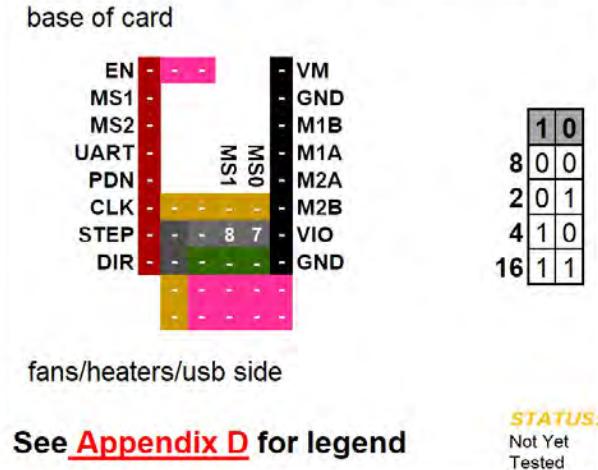


FYSETC TMC2208 V1.2

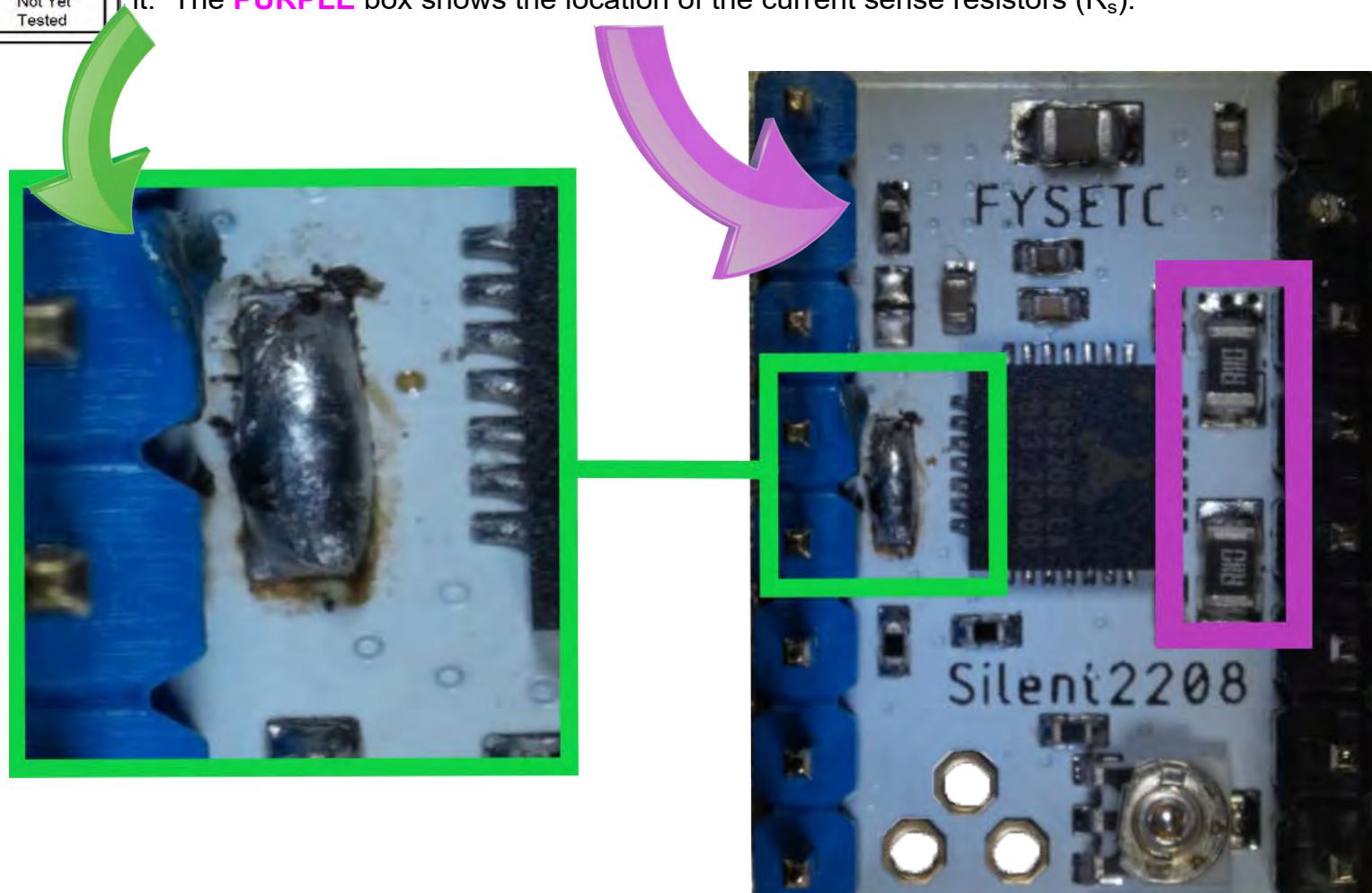
One Time Programming (OTP) Mode

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).



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****OTP (One Time Programming) Mode**

OTP

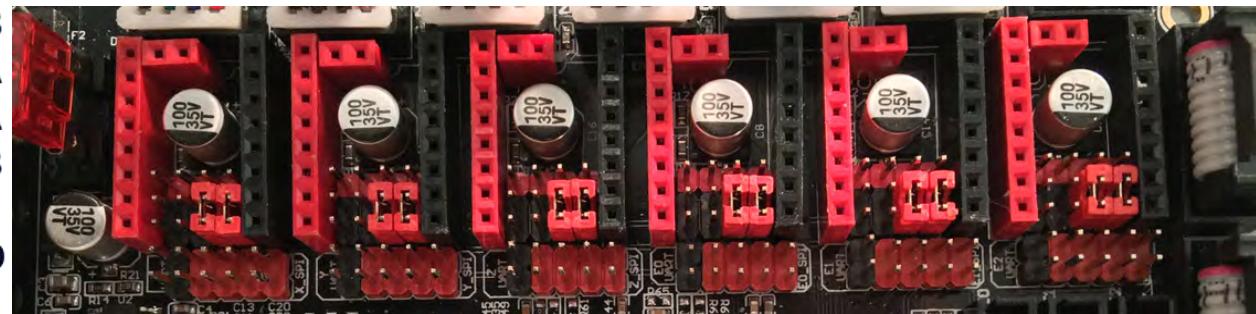
1 / 8

Interpolation:

1 / 256

SpreadCycle

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	-	-	M2B
STEP	-	8	7	VIO
DIR	-	8	7	GND

See [Appendix D](#) for legend

OTP

1 / 2

Interpolation:

1 / 256

SpreadCycle

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	-	-	M2B
STEP	-	-	8	VIO
DIR	-	8	7	GND

See [Appendix D](#) for legend

FYSETC TMC2208 V1.2One Time Programming (OTP) Mode**OTP (One Time Programming) Mode**

OTP

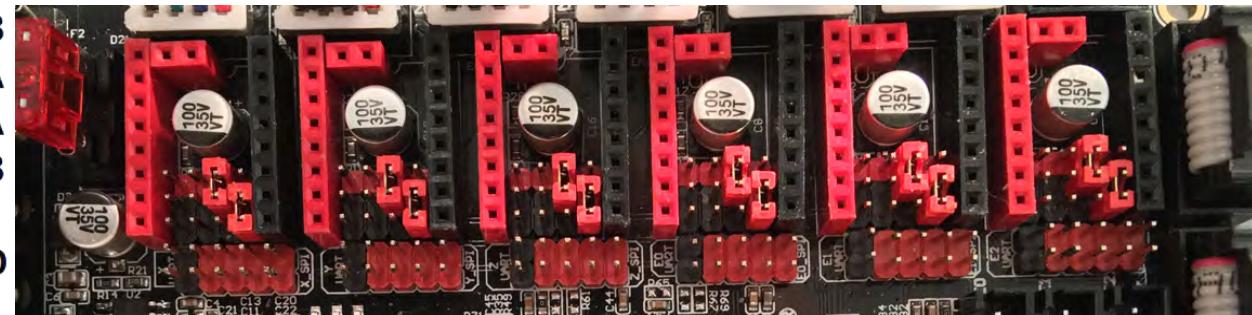
1 / 4

Interpolation:

1 / 256

SpreadCycle

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	8	-	M2B
STEP	-	8	7	VIO
DIR	-	-	7	GND

See [Appendix D](#) for legend

OTP

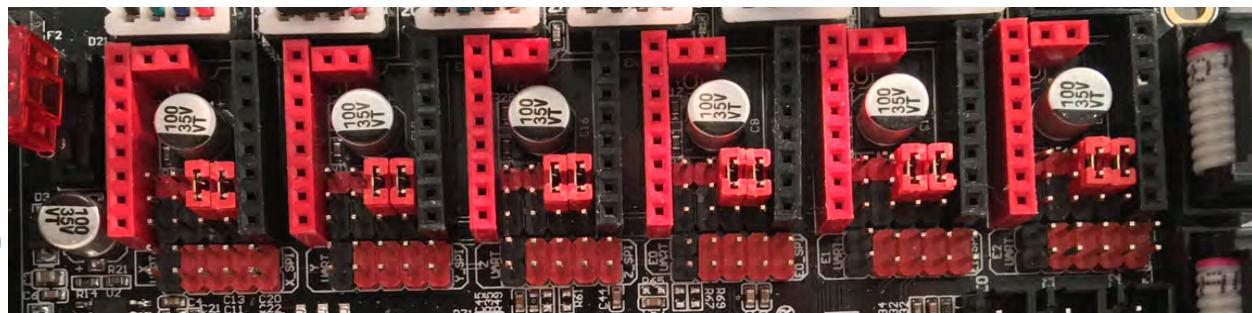
1 / 16

Interpolation:

1 / 256

SpreadCycle

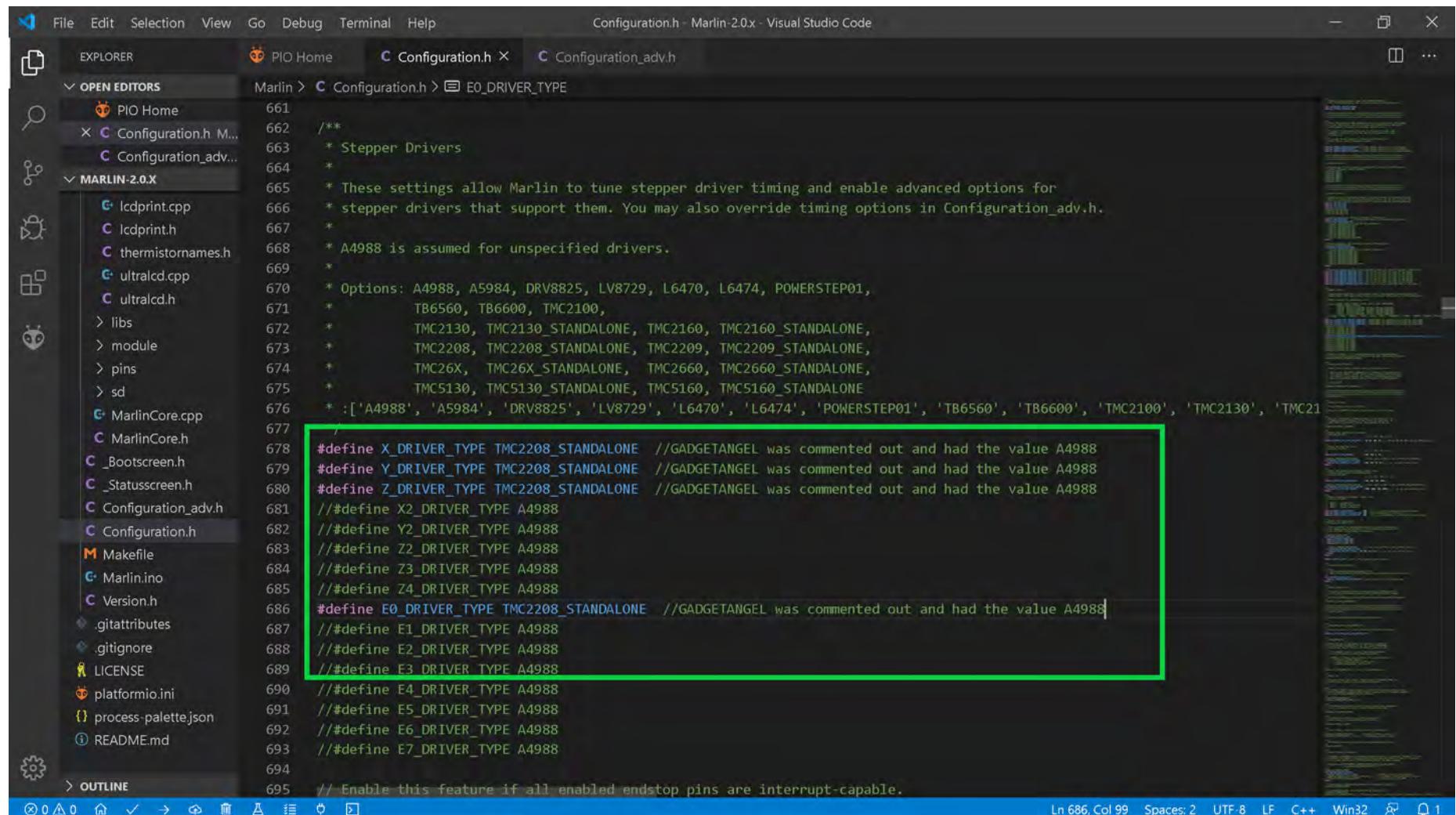
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	M1B
UART	-	-	MS1	M1A
PDN	-	-	MS0	M2A
CLK	-	8	7	M2B
STEP	-	8	7	VIO
DIR	-	-	-	GND

See [Appendix D](#) for legend

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 ("//").



```

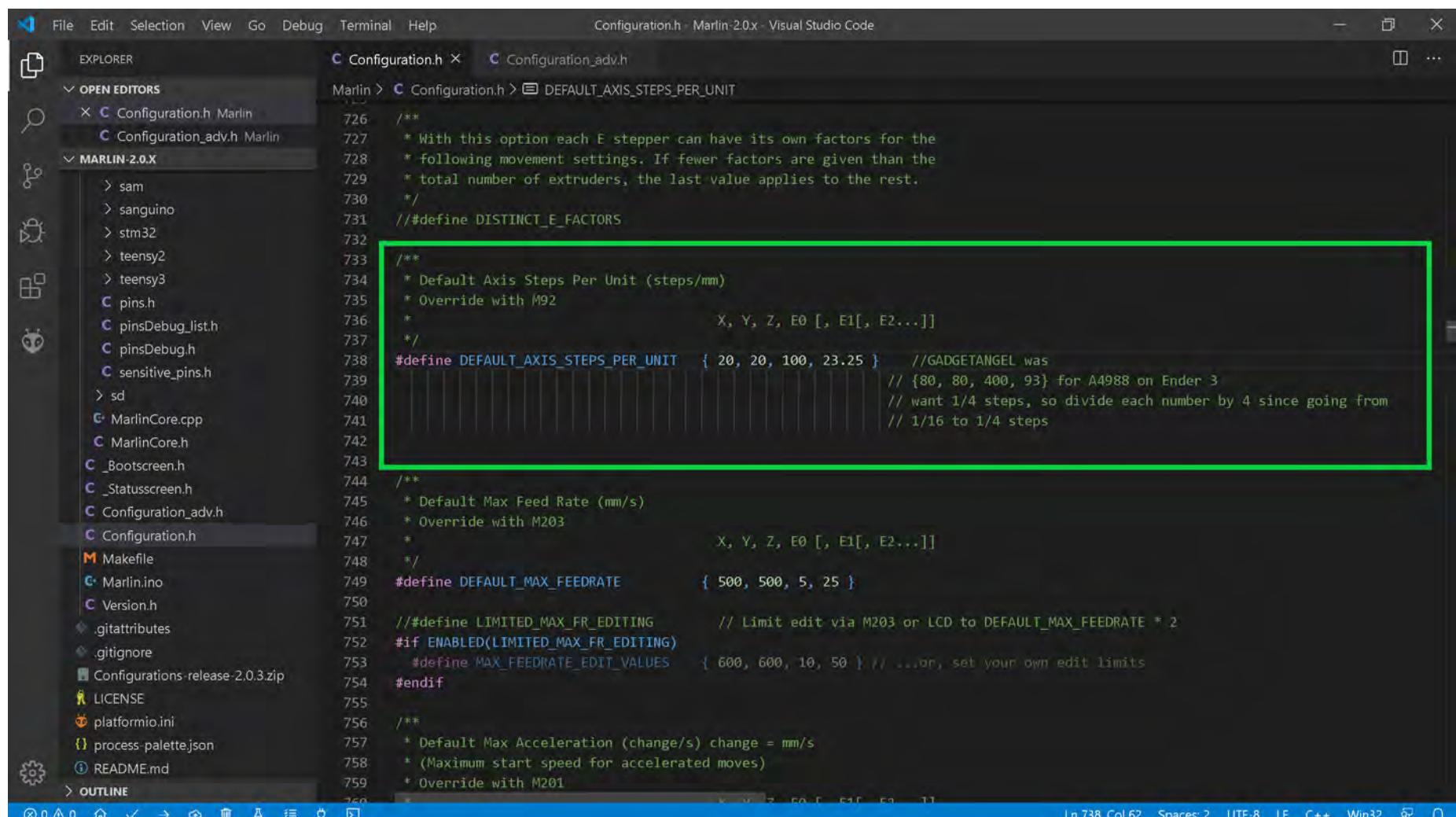
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home 661 /**
X Configuration.h M... 662 */
C Configuration.h M... 663 * Stepper Drivers
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', 'TMC2
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
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 file `Configuration.h` open. The code editor displays the Marlin 2.0.x configuration 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

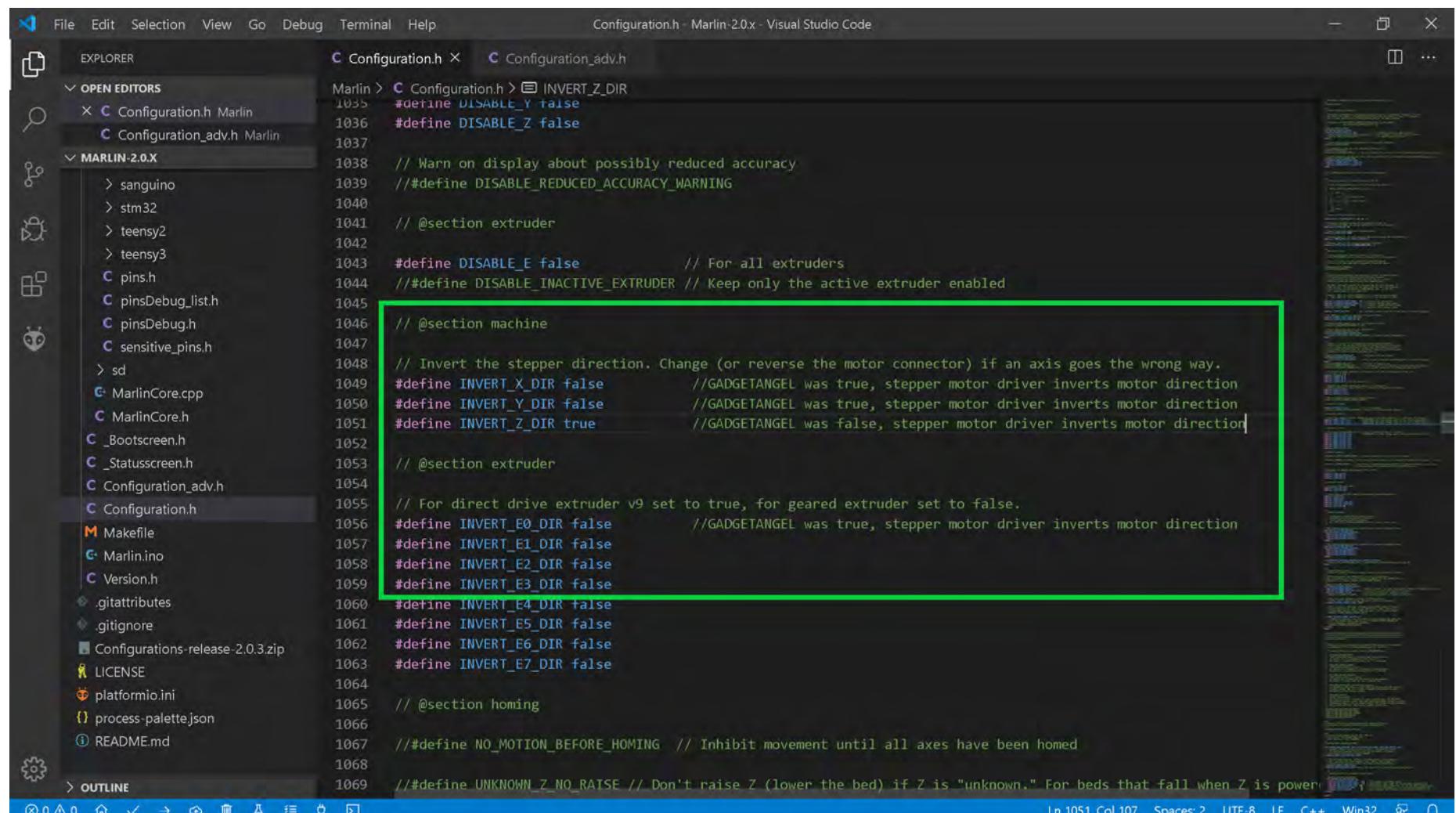
755 /**
756 * Default Max Acceleration (change/s) change = mm/s
757 * (Maximum start speed for accelerated moves)
758 * Override with M201
759 */

```

- 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 section:

```

1035 #define DISABLE_X false
1036 #define DISABLE_Y 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:

```

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

```

The status bar at the bottom right shows: 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 One Time Programming (OTP) Mode

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

The screenshot shows the Visual Studio Code interface with the Marlin repository open. The left sidebar shows files like Configuration.h, pins_BTT_SKR_PRO_V1_1.h, and Configuration_adv.h. The main editor window displays code related to stepper drivers and endstops. The bottom right shows the terminal output:

```

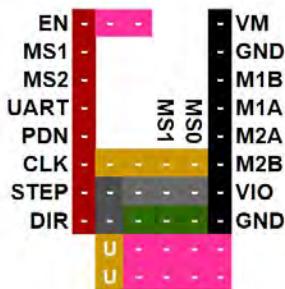
    C:\Users\GadgetAngel\Documents\GitHub\Marlin> make
    task : Task - Build
    [ 0%] BIGTREE_SKR_PRO SUCCESS 00:02:31.294
    [ 0%] BIGTREE_BT8002 IGNORED
    [ 0%] teensy31 IGNORED
    [ 0%] teensy35 IGNORED
    [ 0%] esp32 IGNORED
    [ 0%] linux_native IGNORED
    [ 0%] SAMD51_grandcentral_m4 IGNORED
    [ 0%] rumba32_f446ve IGNORED
    [ 0%] mks_rumba32 IGNORED
    [ 0%] include_tree IGNORED
    ===== 1 succeeded in 00:02:31.294 =====
  
```

The terminal also displays 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 [Appendix E](#).

FYSETC TMC2208 V1.2**UART Mode**

base of card

UART Mode

fans/heaters/usb side

STATUS:
Not Yet
Tested

See [Appendix D](#) for legend

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

FYSETC
TMC2208

UART Mode

Maximum 256
Subdivision35V DC
2A (peak)

Driving Current Calculation Formula

R_S (Typical Sense Resistor) = 0.11Ω

**Steps are set inside
of your Firmware**

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

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:

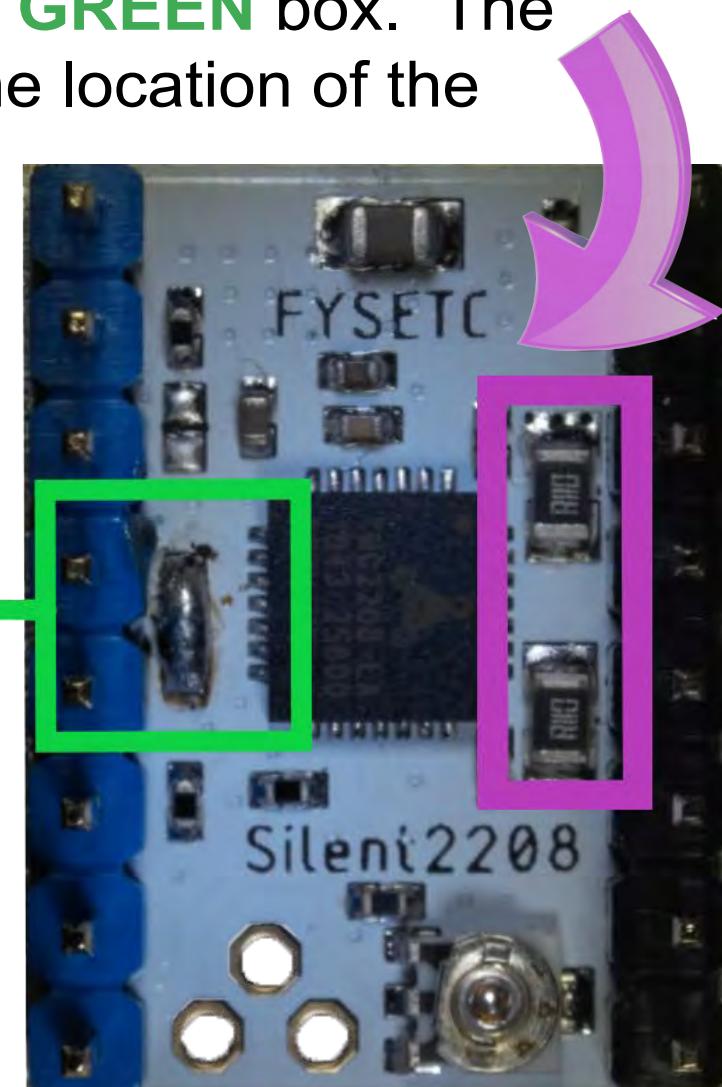
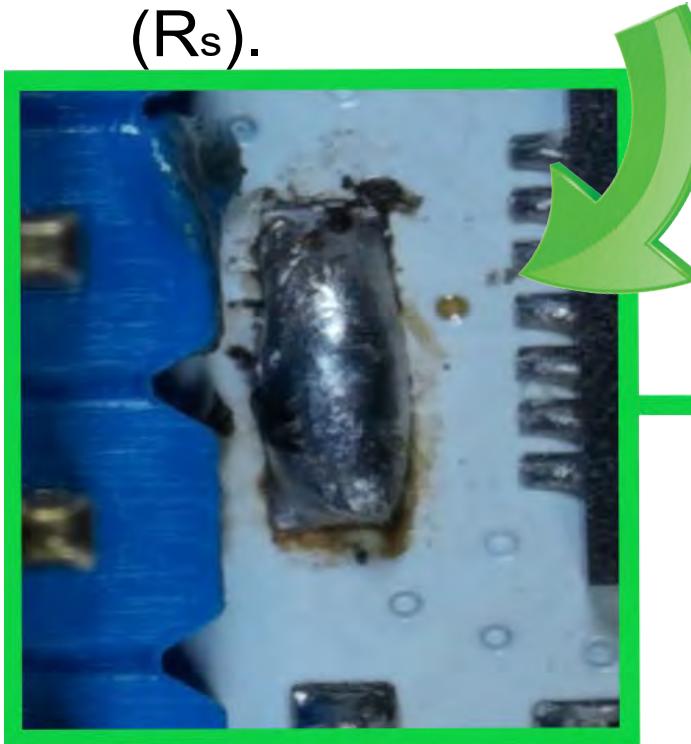
$$V_{ref} = V_{ref} * 0.90$$

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

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

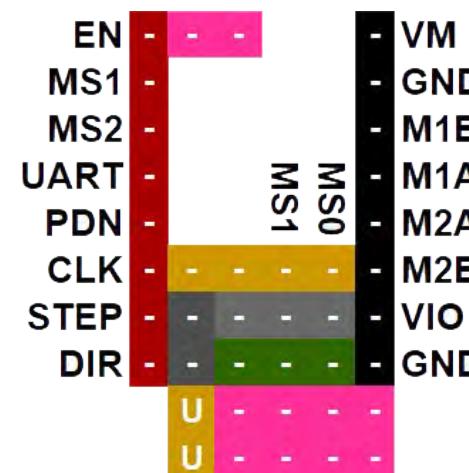
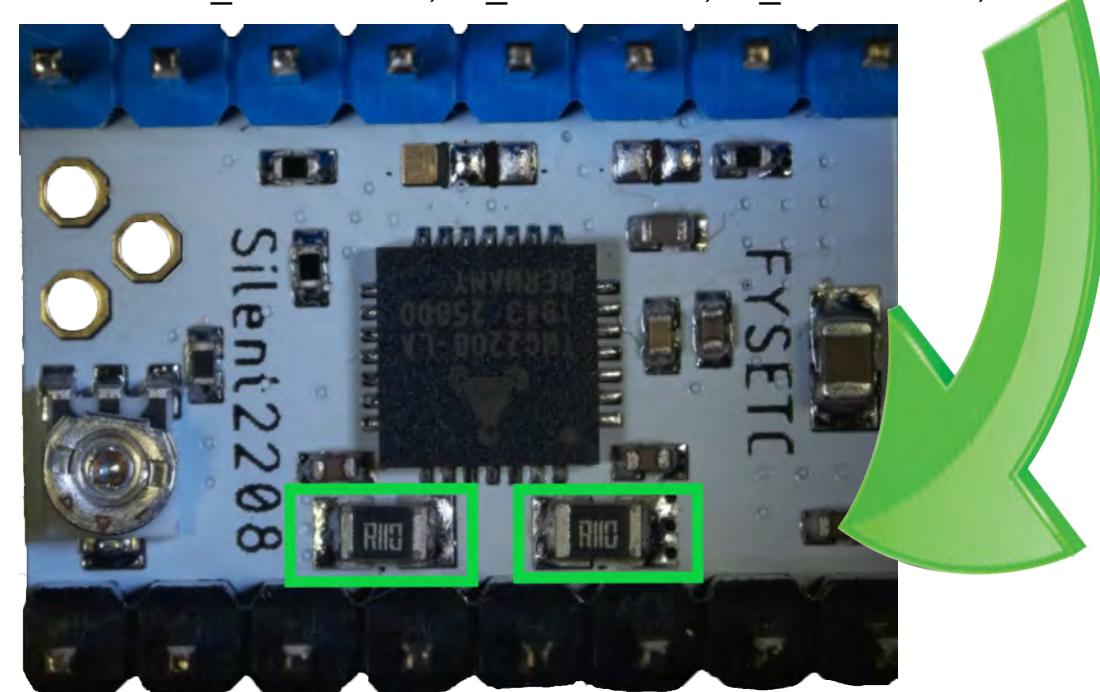
$R_s = R_{100}$ is 0.1 Ohms

$R_s = R_{110}$ is 0.11 Ohms

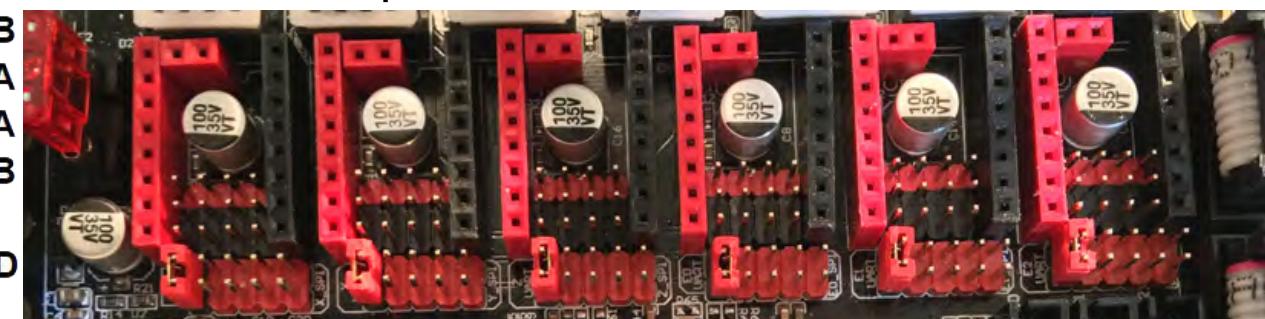
$R_s = R_{150}$ is 0.15 Ohms

$R_s = R_{200}$ is 0.2 Ohms

$R_s = R_{220}$ is 0.22 Ohms



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



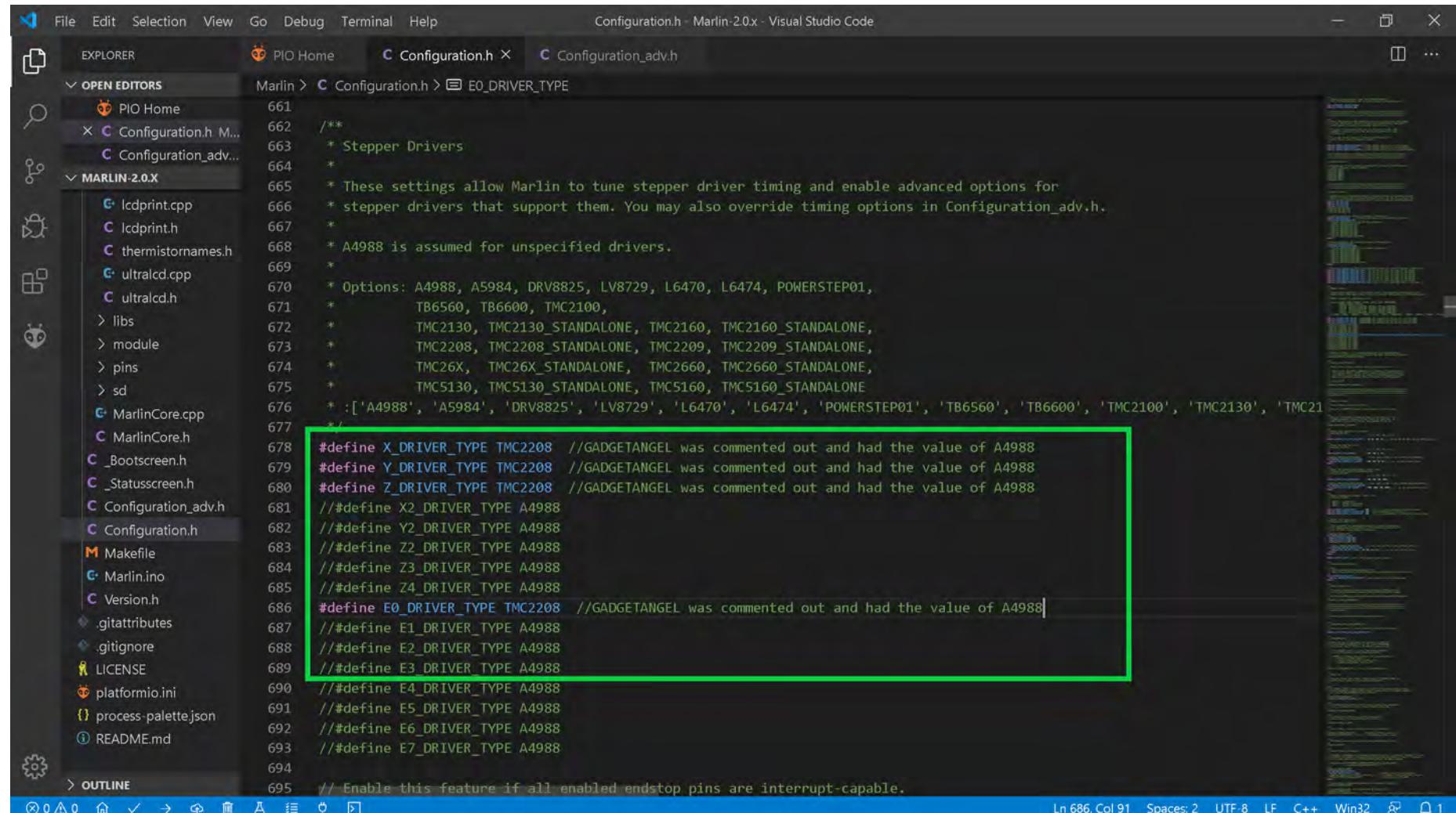
See [Appendix D](#) for legend

UART

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

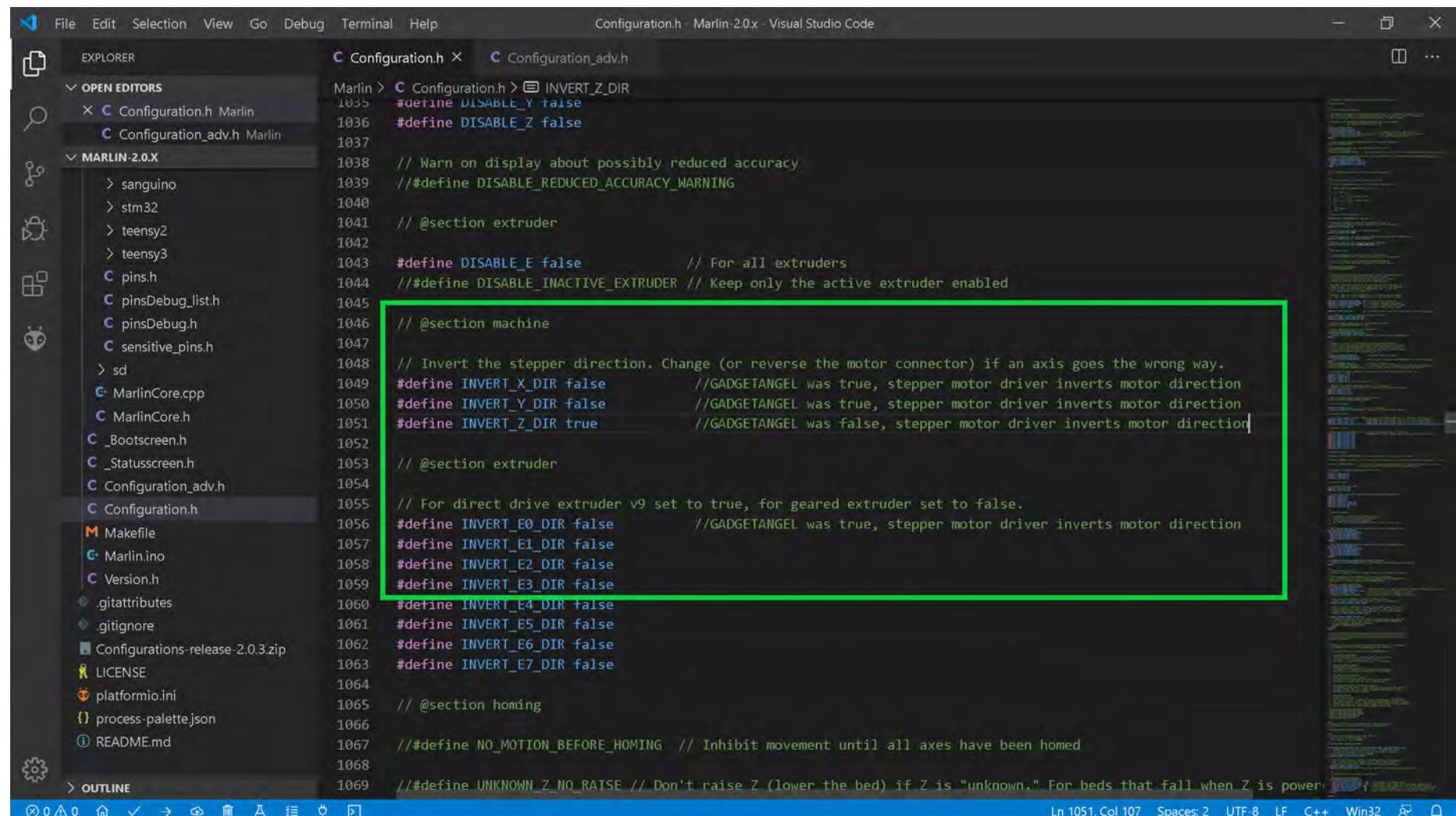
661 /**
662 * Stepper Drivers
663 *
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
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

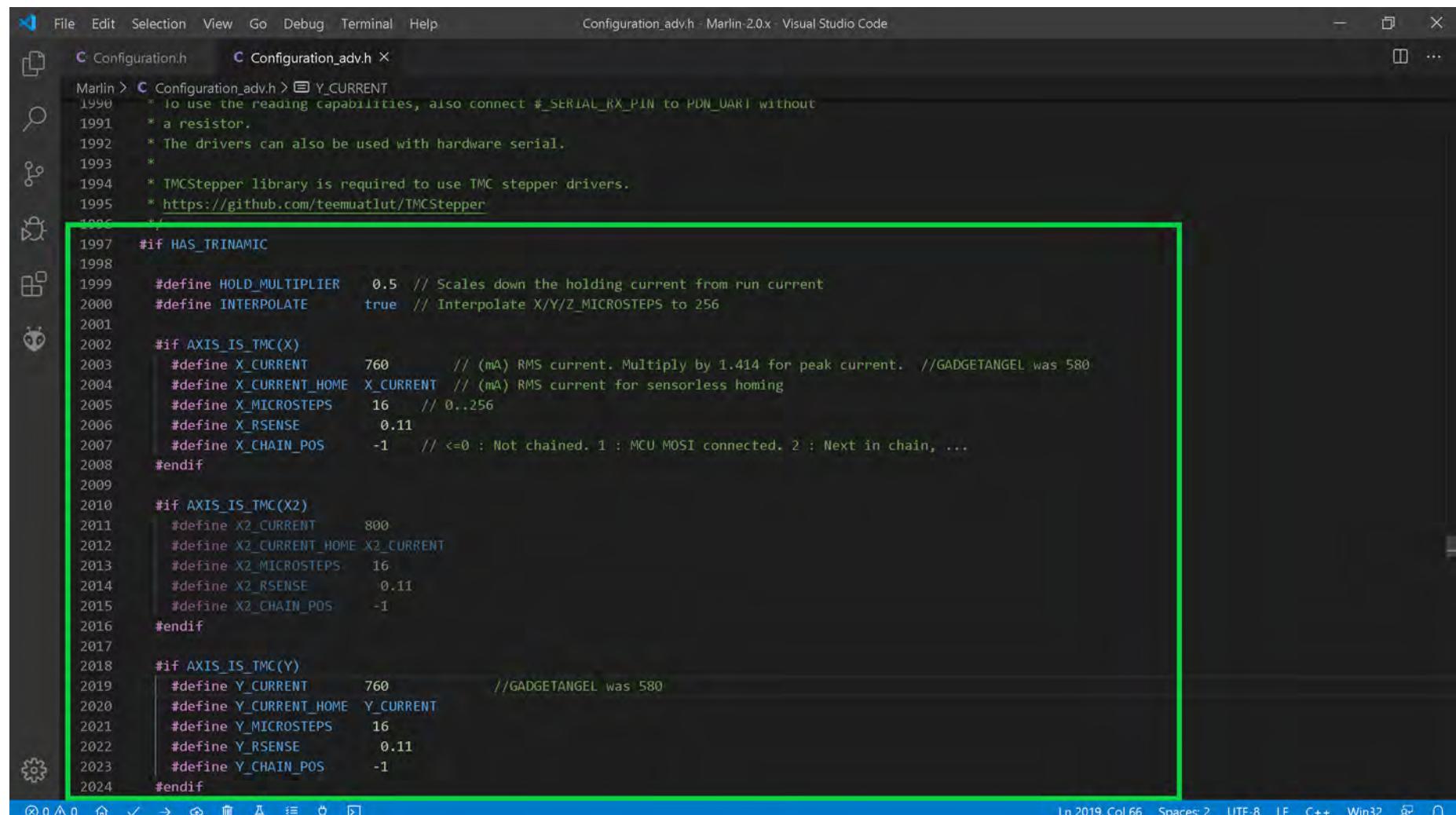
```

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".



```

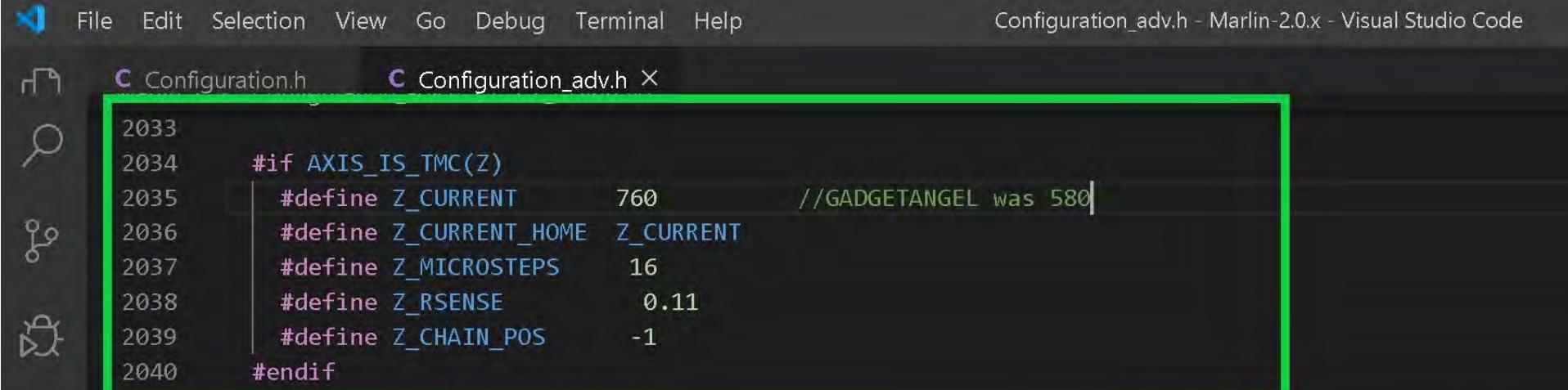
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers 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".



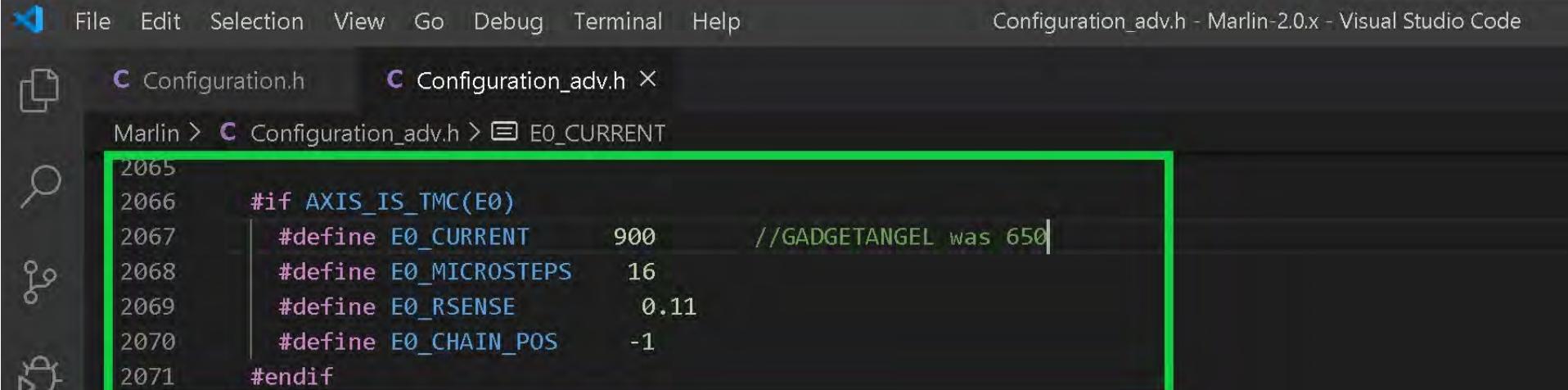
```

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

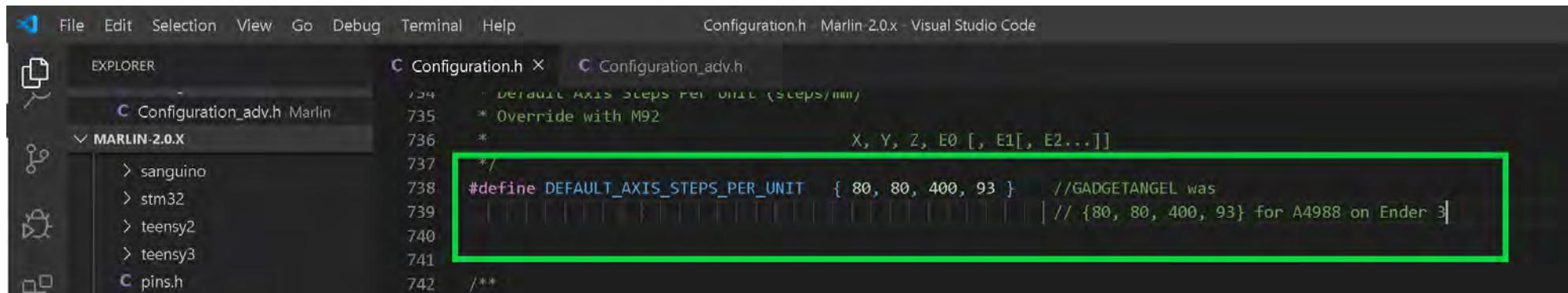
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900           //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- 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

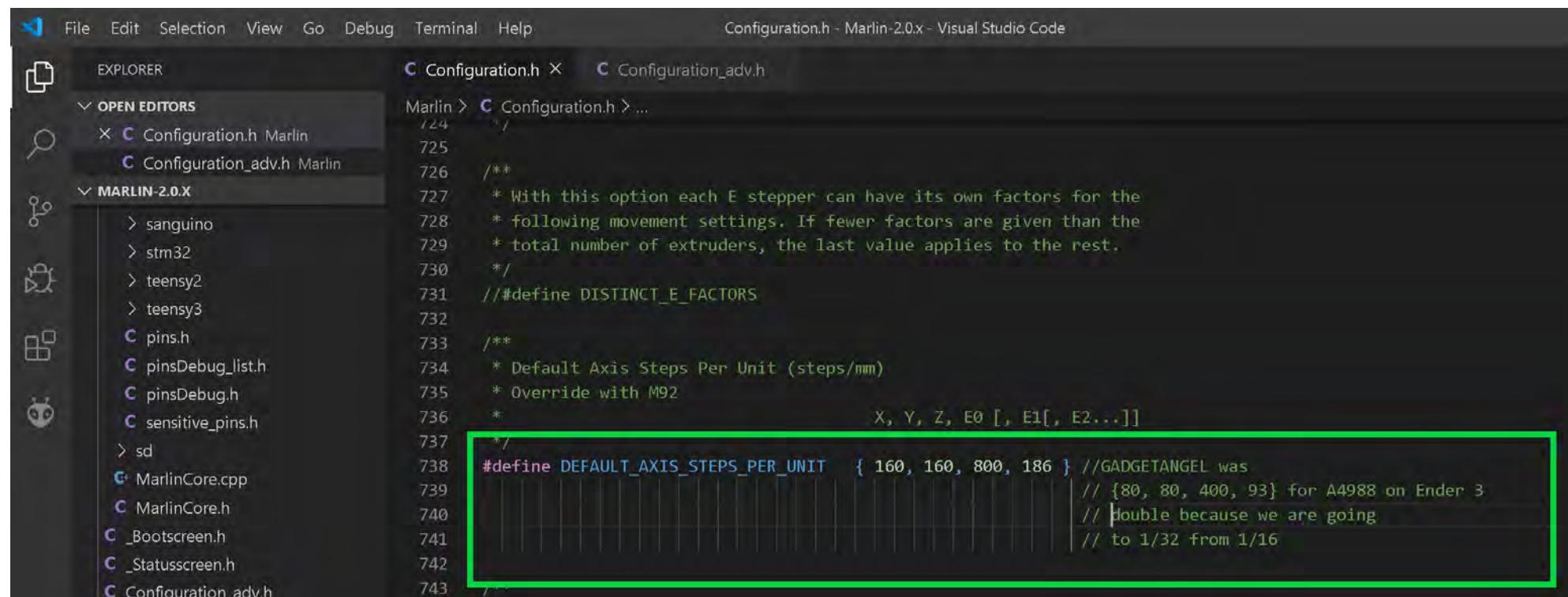
MARLIN-2.0.X

```

 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

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

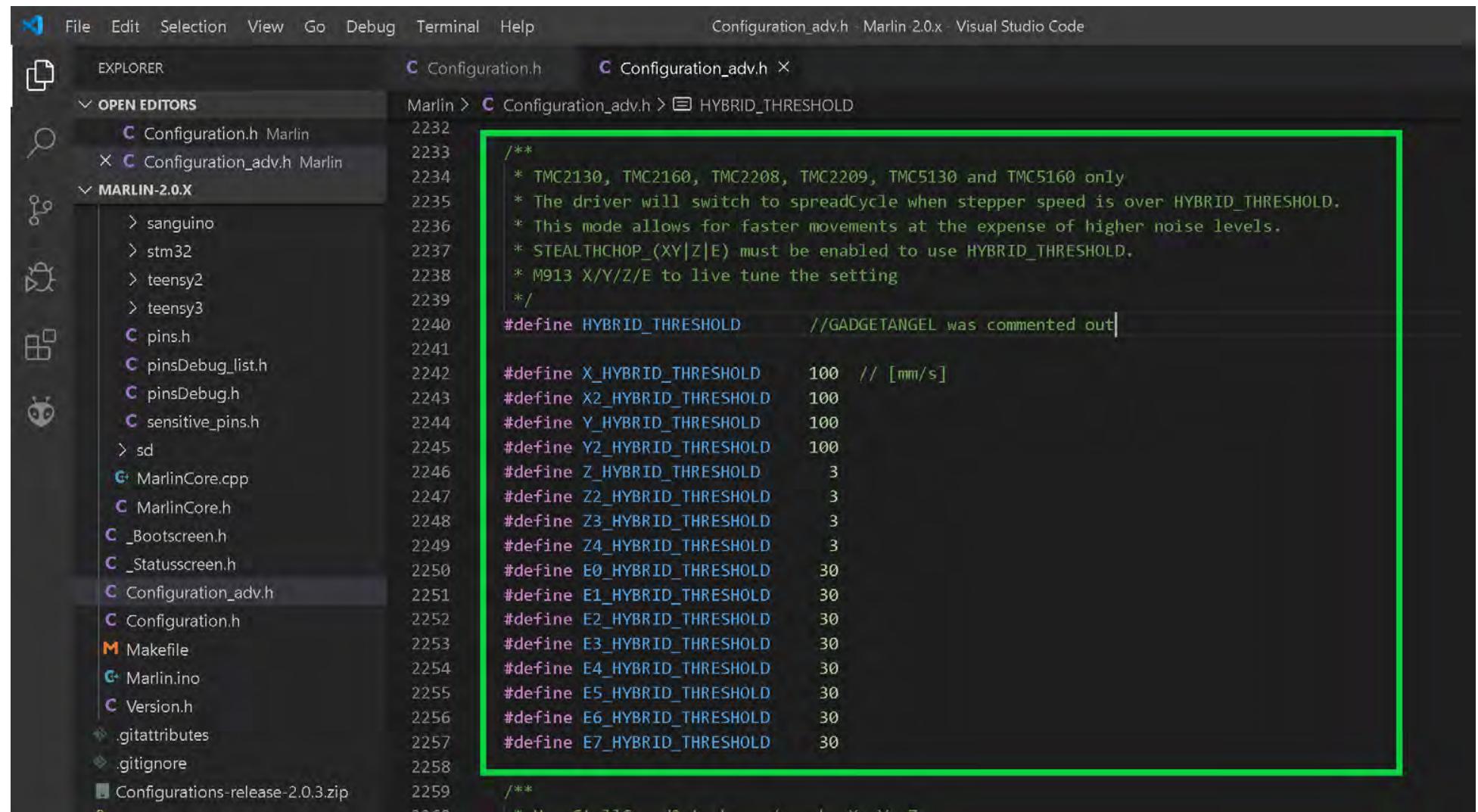
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  pins.h
  pinsDebug_list.h
  pinsDebug.h
  sensitive_pins.h
  > sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h

Configuration.h Configuration_adv.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
2214 /**
```

- 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
- Left Sidebar (EXPLORER):**
 - 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
 - C Marlin.ino
 - C Version.h
 - .gitattributes
 - .gitignore
 - Configurations-release-2.0.3.zip
- Central Area (Editor):** Shows the content of Configuration_adv.h, specifically the HYBRID_THRESHOLD section. The code is as follows:


```

2232
2233
2234 /**
2235 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
2236 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
2237 * This mode allows for faster movements at the expense of higher noise levels.
2238 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
2239 * M913 X/Y/Z/E to live tune the setting
2240 */
2241 #define HYBRID_THRESHOLD //GADGETANGEL was commented out
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 */
      
```

- 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 ×
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

```

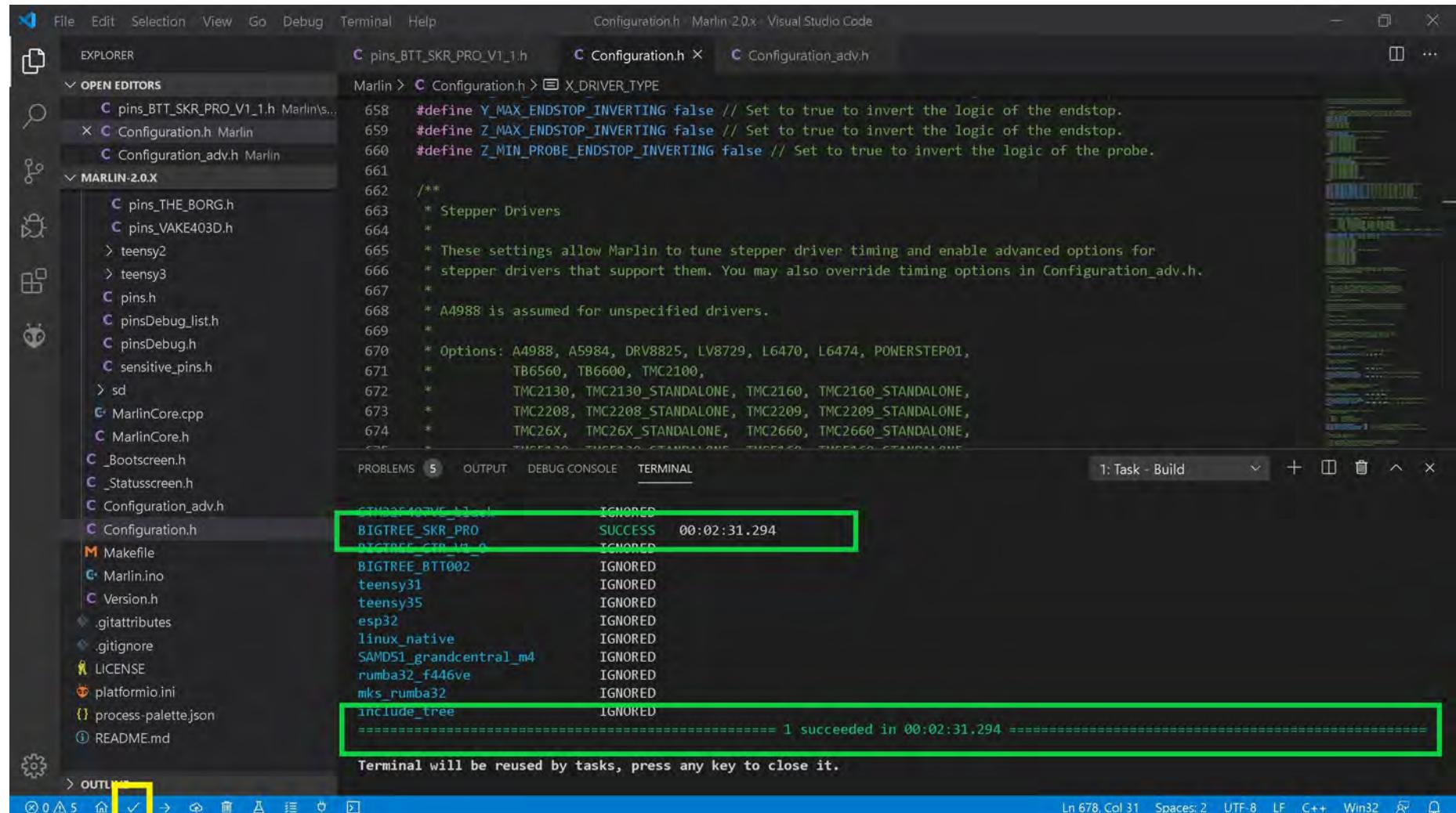
File Edit Selection View Go Debug Terminal Help Configuration_adv.h × Configuration_adv.h ×
Marlin > C Configuration_adv.h > #define TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //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.



```

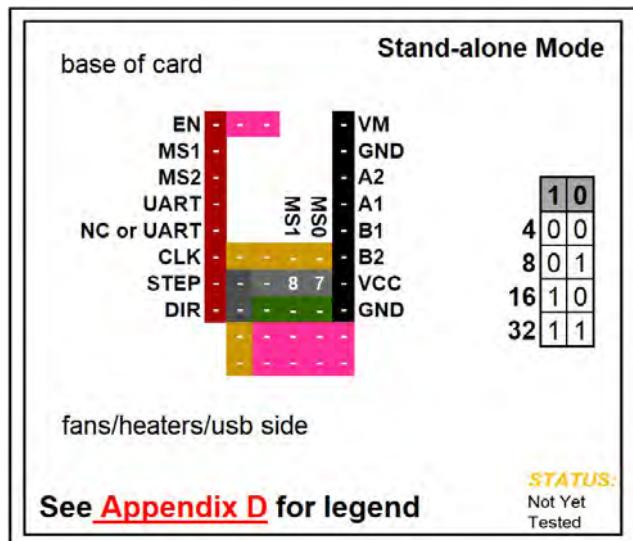
Configuration.h - Marlin 2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS pins_BTT_SKR_PRO_V1_1.h Configuration.h Configuration_adv.h
Marlin > Configuration.h > X_DRIVER_TYPE
pins_BTT_SKR_PRO_V1_1.h Marlin\pins.h Configuration.h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

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + □ ✘ ×
BIGTREE_SKR_PRO SUCCESS 00:02:31.294
BIGTREE_SKR_V1_0 IGNORED
BIGTREE_BT002 IGNORED
teensy31 IGNORED
teensy35 IGNORED
esp32 IGNORED
linux_native IGNORED
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 ⌂

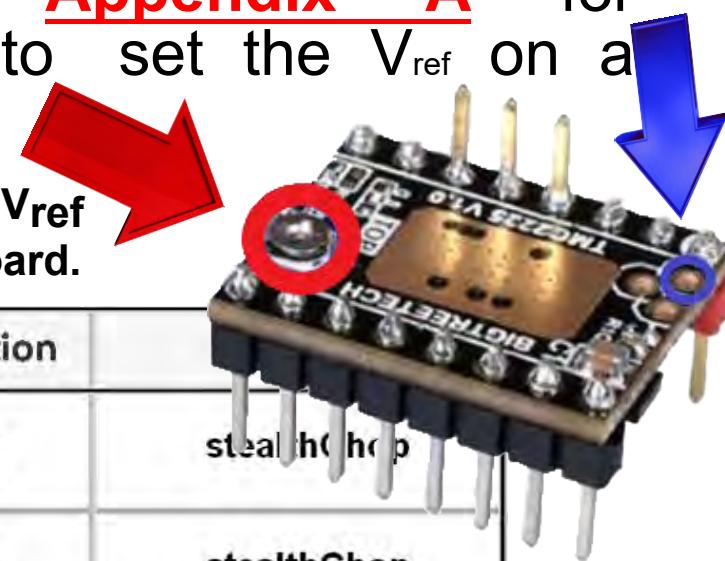
```

- 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 [Appendix E](#).



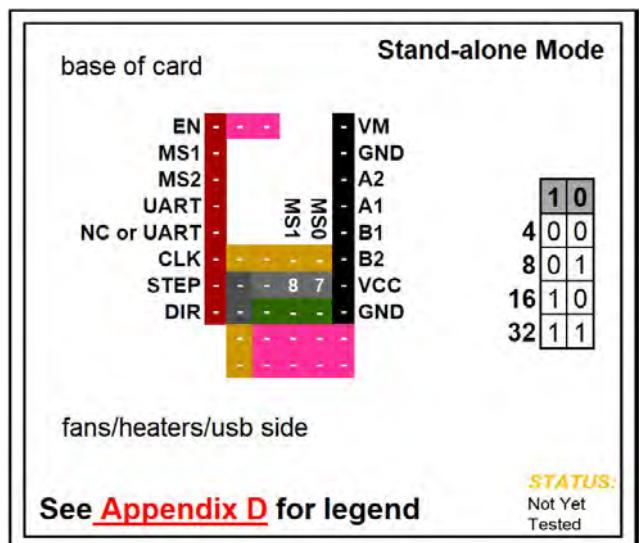
NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V_{ref}** **Test point**" location, as shown in **BLUE**, to set your **V_{ref}**. See [**Appendix A**](#) for instructions on how to set the **V_{ref}** on a driver board.

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



Driver Chip	MS1	MS0	Steps	Interpolation	
BBIQU® TMC2225 Stand Alone Mode Maximum 32 Subdivision 35V DC 2A (peak)	Low	Low	1 / 4	1 / 256	stealthChop
	Low	High	1 / 8	1 / 256	stealthChop
	High	Low	1 / 16	1 / 256	stealthChop
	High	High	1 / 32	1 / 256	stealthChop
Driving Current Calculation Formula	I_{MAX}=V_{ref}*0.7222			V_{ref}=I_{MAX}*1.3846	
<i>R_S</i> (Typical Sense Resistor)=0.15Ω	See Appendix B#10. Use 50% to 90% as shown below: I_{MAX}=(V_{ref}*0.7222)*0.90			See Appendix B#10. Use 50% to 90% as shown below: V_{ref}=(I_{MAX}*1.3846)*0.90	

- See next page for the legend that belongs to the above chart.



This is a
Jumper:

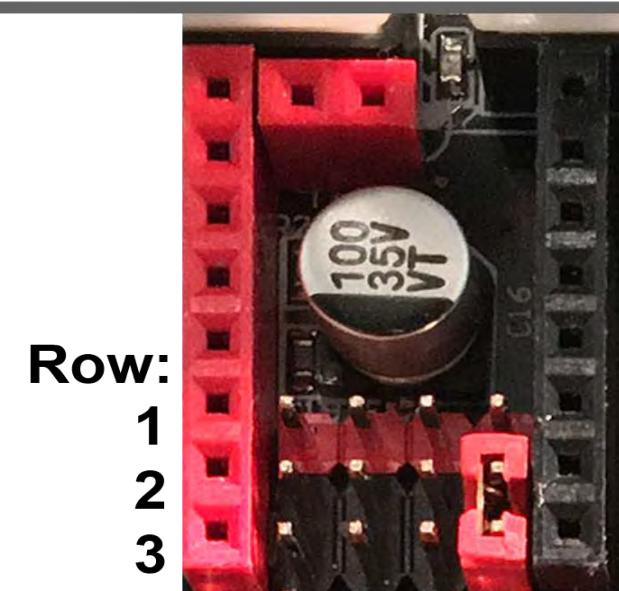
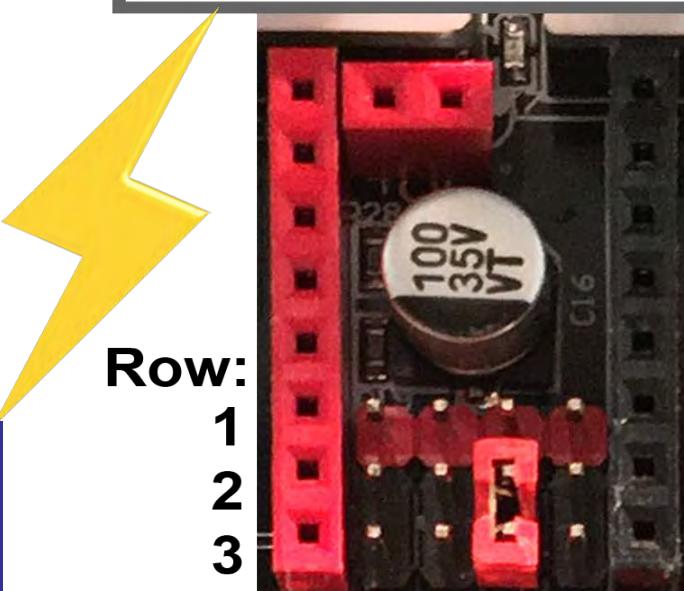


BIQU TMC2225 V1.0

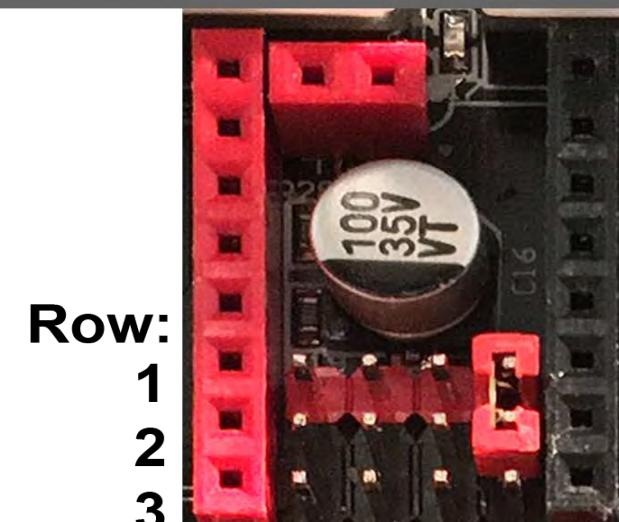
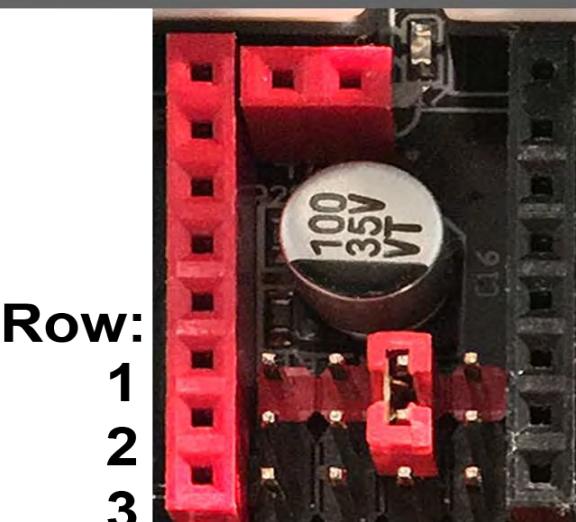
Stand-alone Mode

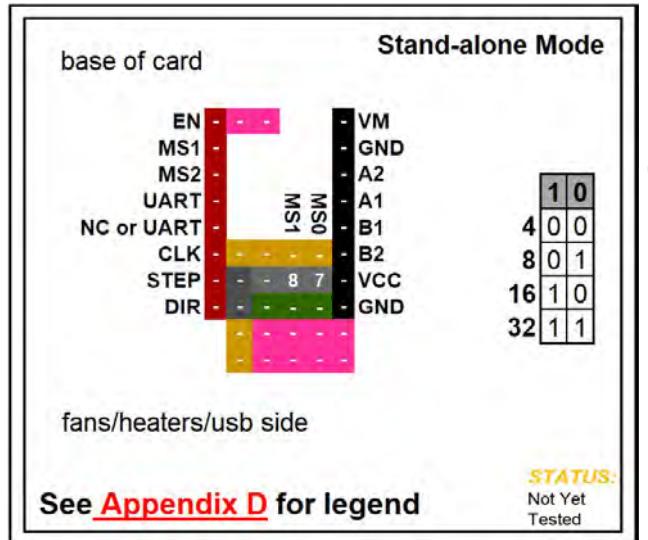
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows
2 and 3



High ➡ set Jumper between rows
1 and 2

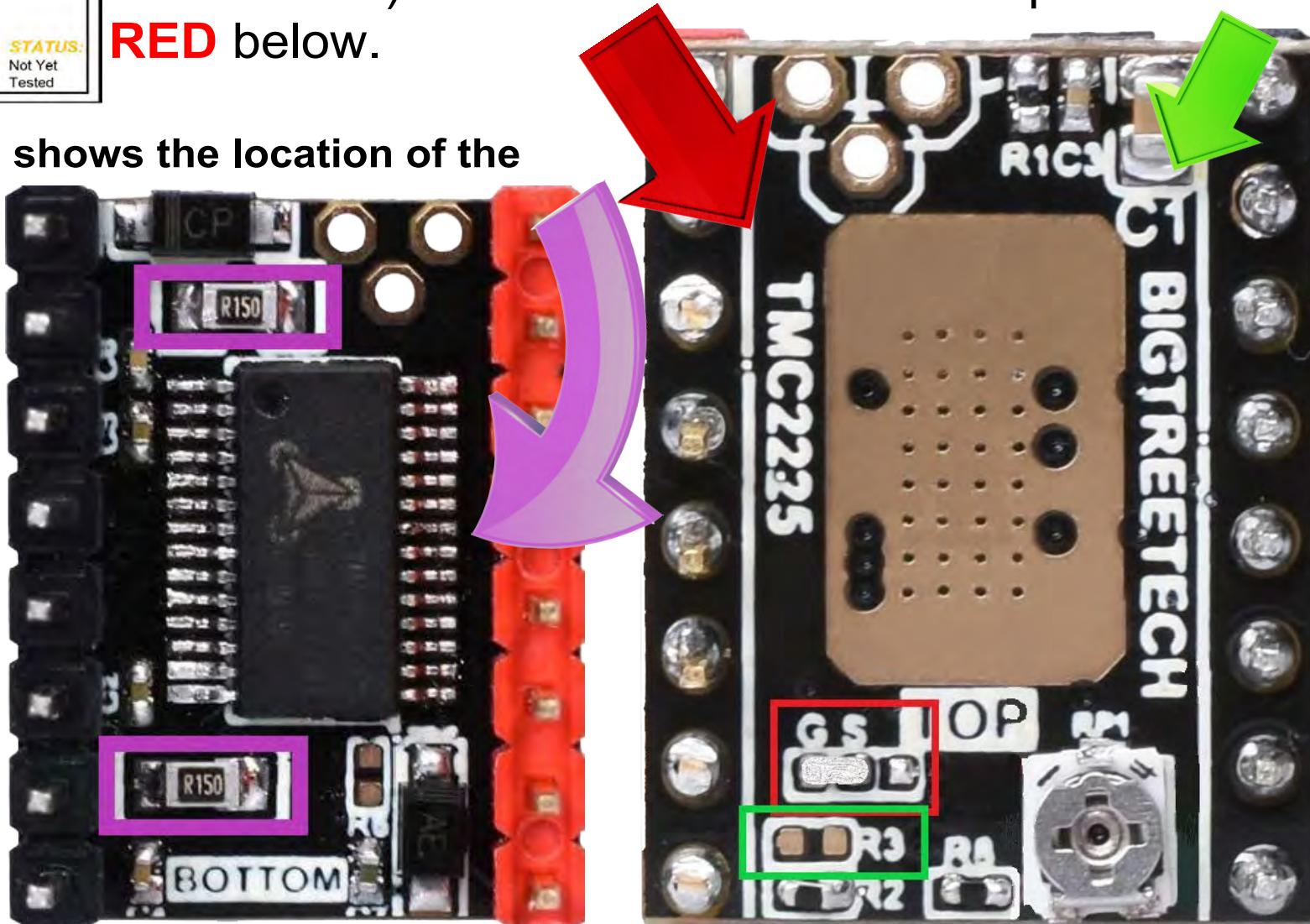


**BIQU TMC2225 V1.0****Stand-alone Mode**

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).

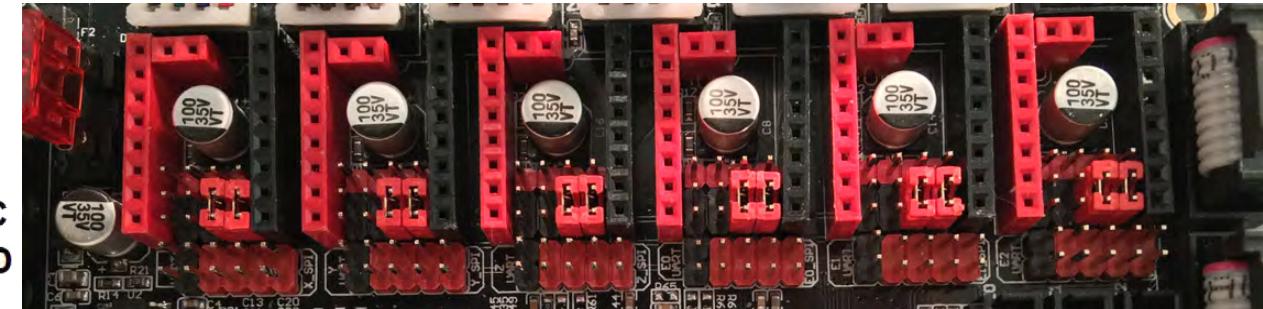
Note: MOST BIQU TMC2225 V1.0 driver boards, when purchased for **UART mode**, will have two R3 pads (located on the top of the driver board), which are **NOT soldered together**. This indicates the driver board can use the UART pin for the UART single wire interface (if the **UART, "U"**, jumper is in place on the SKR PRO V1.1 board)



BIQU TMC2225 V1.0**Stand-alone Mode****Stand-alone Mode****1 / 4**

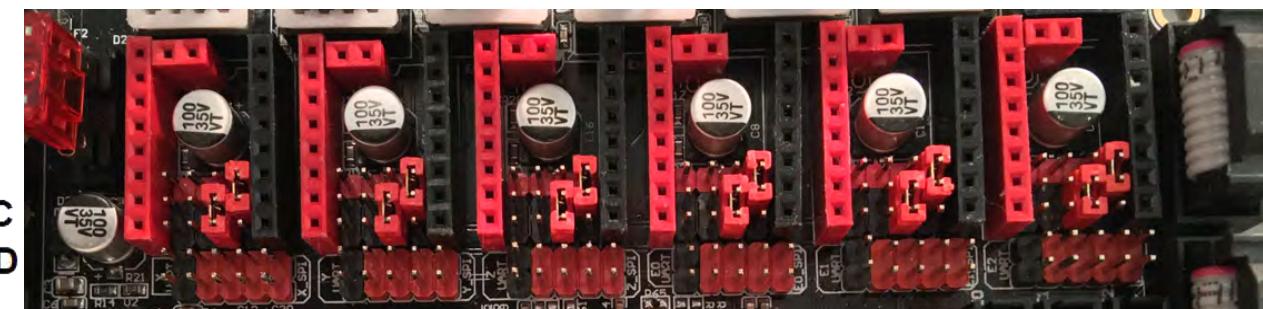
NC or

Stand-alone Mode	EN	-	-	VM
	MS1	-	-	GND
	MS2	-	-	A2
	UART	-	-	A1
Interpolation: 1 / 256	NC or	UART	MS0	B1
	CLK	-	-	B2
StealthChop	STEP	-	8 7	VCC
	DIR	-	8 7	GND

See [Appendix D](#) for legend**Stand-alone
Mode****1 / 8**

NC or

Stand-alone Mode	EN	-	-	VM
	MS1	-	-	GND
	MS2	-	-	A2
	UART	-	-	A1
Interpolation: 1 / 256	NC or	UART	MS0	B1
	CLK	-	7	B2
StealthChop	STEP	-	8 7	VCC
	DIR	-	8	GND

See [Appendix D](#) for legend

Stand-alone Mode

BIQU TMC2225 V1.0

Stand-alone Mode

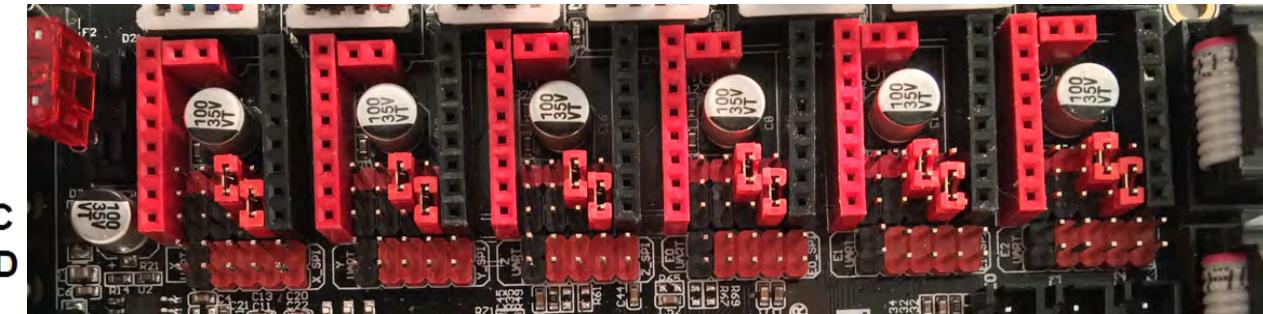
Stand-alone
Mode

1 / 16

NC or

UART

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	-	-	A1
CLK	8	-	MS0	B1
STEP	-	7	-	B2
DIR	-	7	-	VCC
	-	-	-	GND



See [Appendix D](#) for legend

Interpolation:
1 / 256
StealthChop

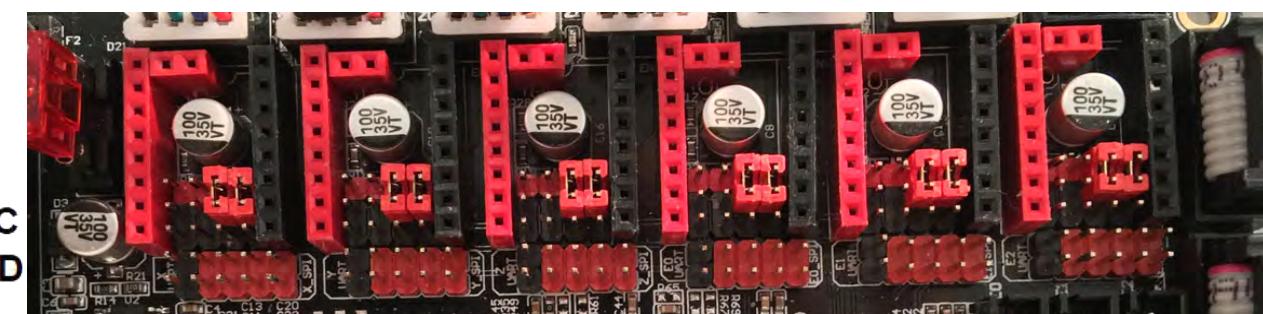
Stand-alone
Mode

1 / 32

NC or

UART

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	-	-	A1
CLK	8	7	MS0	B1
STEP	-	7	-	B2
DIR	-	-	-	VCC
	-	-	-	GND



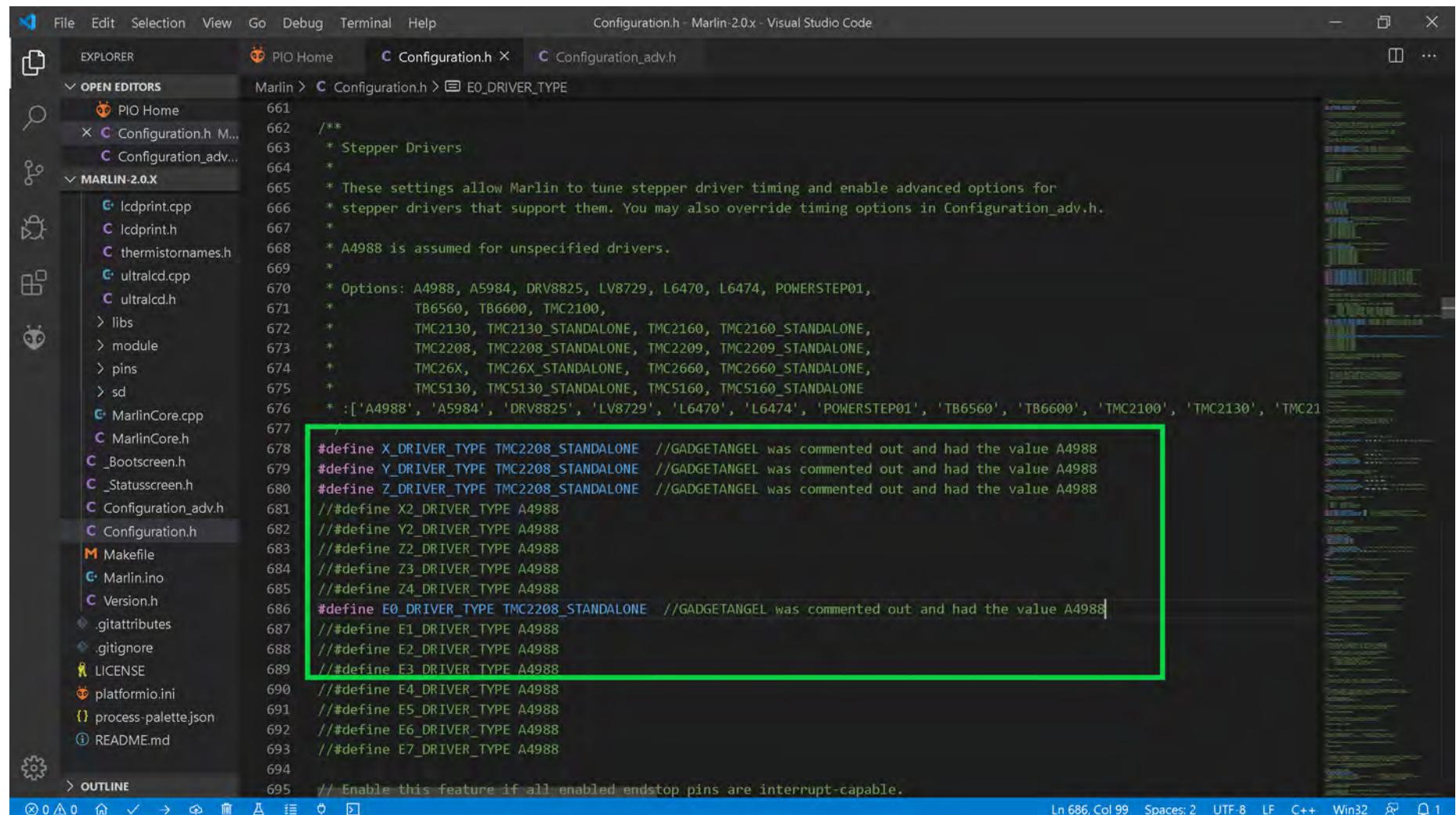
See [Appendix D](#) for legend

Interpolation:
1 / 256
StealthChop

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 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 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
668 *           TB6560, TB6600, TMC2100,
669 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
670 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
671 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
672 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
673 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC26X', 'TMC5130']
674 */
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
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 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
691 //##define E4_DRIVER_TYPE A4988
692 //##define E5_DRIVER_TYPE A4988
693 //##define E6_DRIVER_TYPE A4988
694 //##define E7_DRIVER_TYPE A4988
695
696 // 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 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 Marlin 2.0.x repository open. The left sidebar shows various files and folders, including Configuration.h, Configuration_adv.h, and several pins and core files. The main editor window displays the Configuration.h file. A green rectangular box highlights the following line of code:

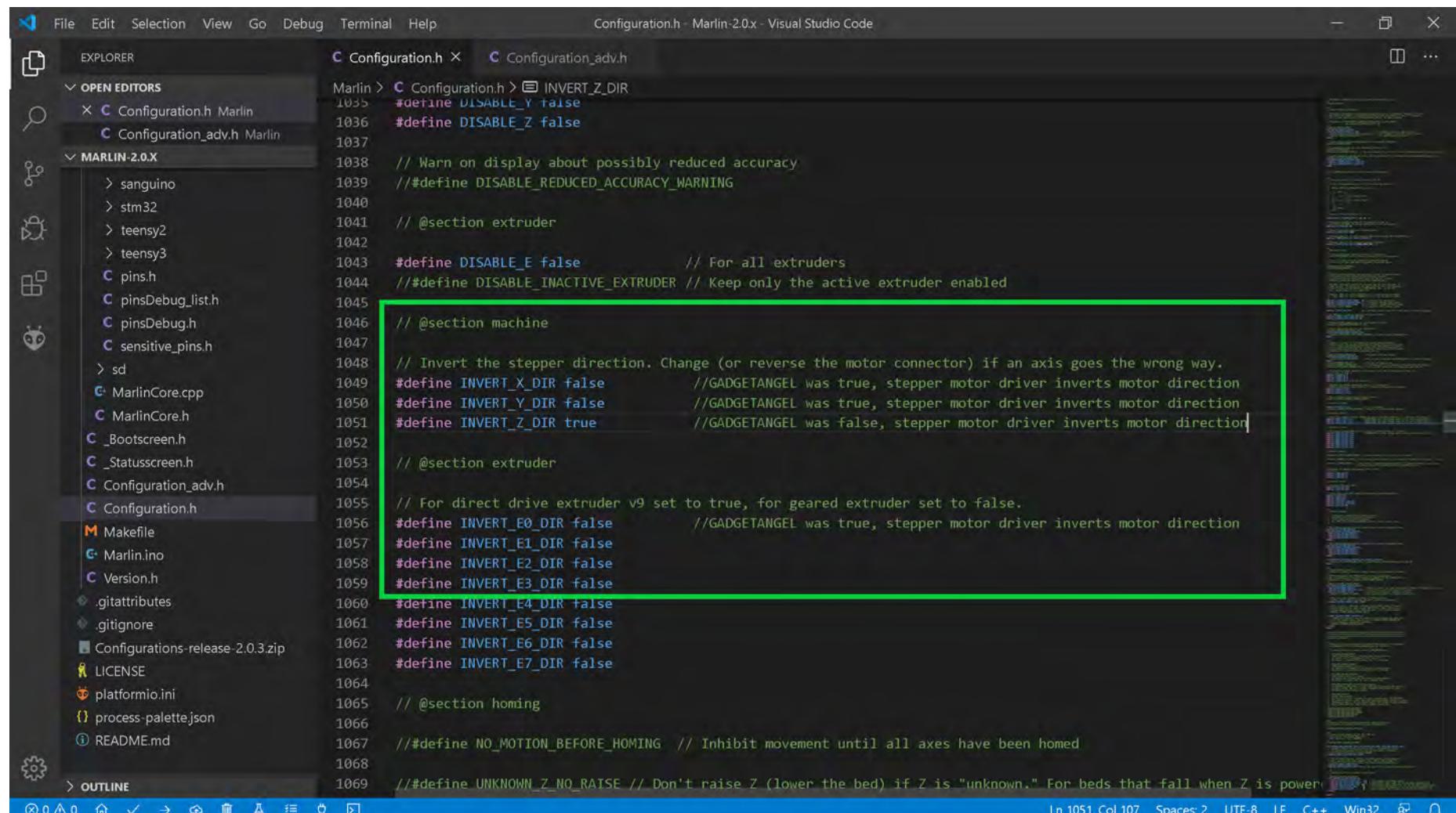
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps
```

The status bar at the bottom indicates the line number (Ln 738), column (Col 62), and other settings like spaces, encoding, and file type.

- 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



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

```

Marlin > C Configuration.h > #define 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

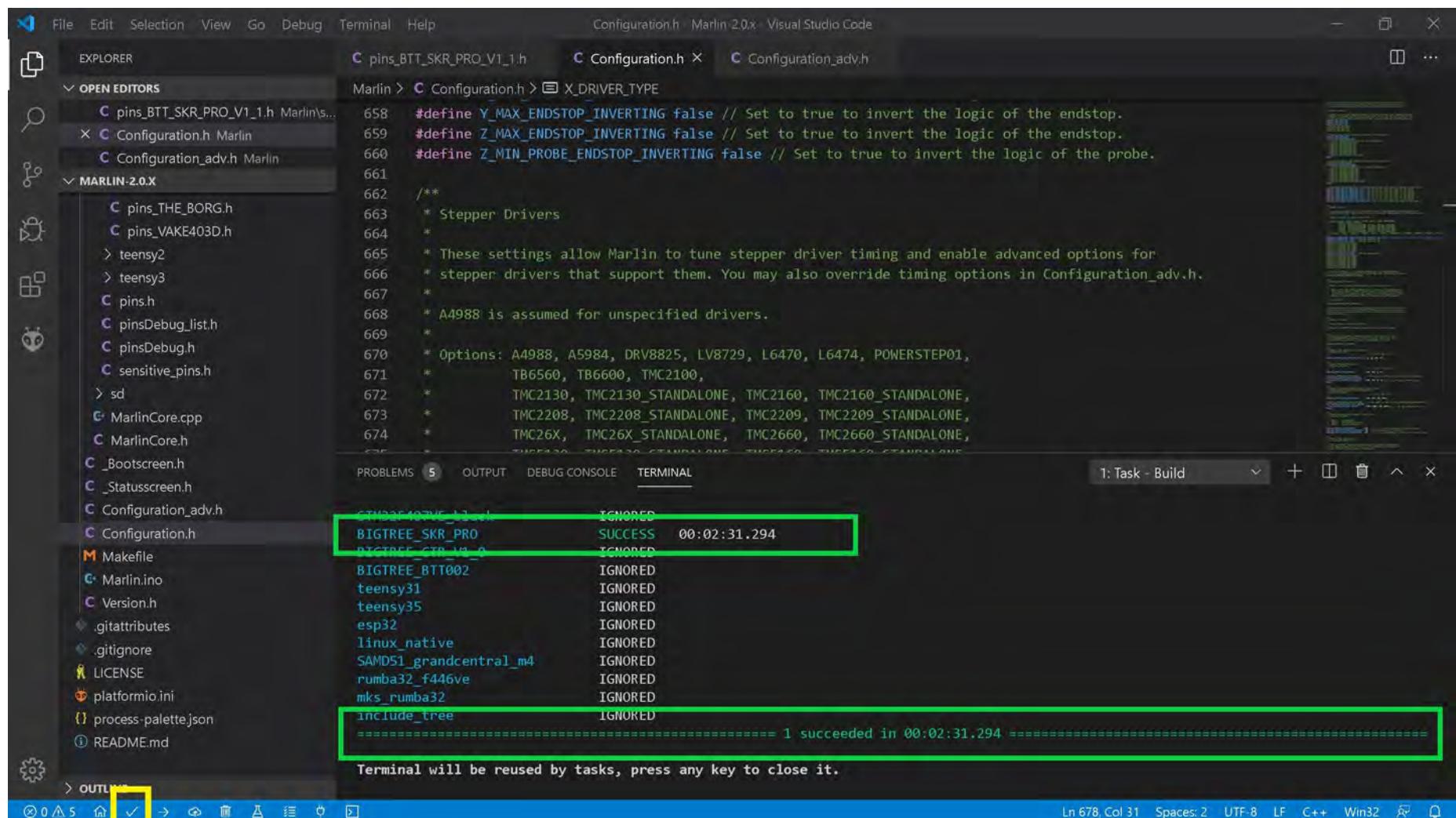
```

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

- 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.



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

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

```
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, 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 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Task - Build + ☰ ×

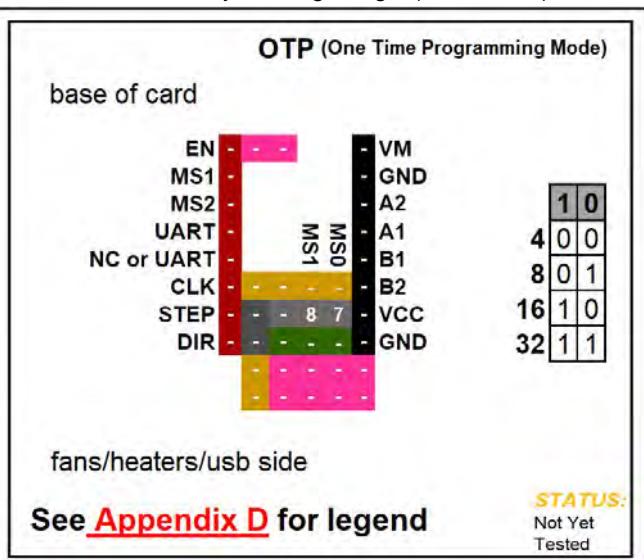
STDOUPLOAD	IGNORED		
BIGTREETREE_SKR_PRO	SUCCESS	00:02:31.294	
BIGTREETREE_SKR_V1_0	IGNORED		
BIGTREETREE_BT1002	IGNORED		
teensy31	IGNORED		
teensy35	IGNORED		
esp32	IGNORED		
linux_native	IGNORED		
SAMD51_grandcentral_m4	IGNORED		
rumba32_f446ve	IGNORED		
mks_rumba32	IGNORED		
include_tree	IGNORED		
===== 1 succeeded in 00:02:31.294 =====			

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

OUTLINE (Yellow box highlights the checkmark icon)

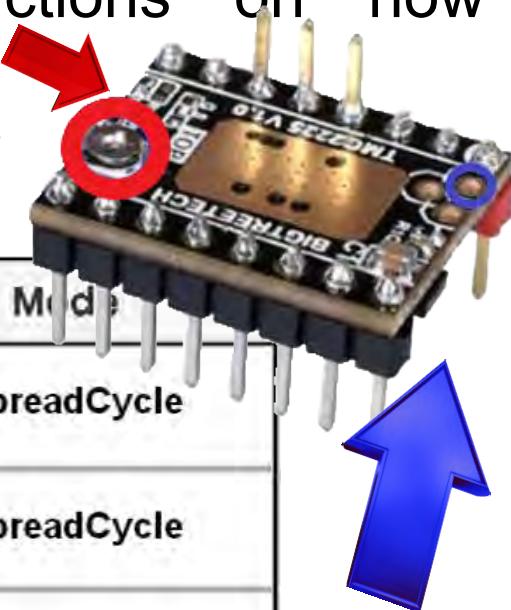
Ln 678, Col 31 Spaces: 2 UTF-8 LF C++ Win32 ☰

- 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 [Appendix E](#).

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.



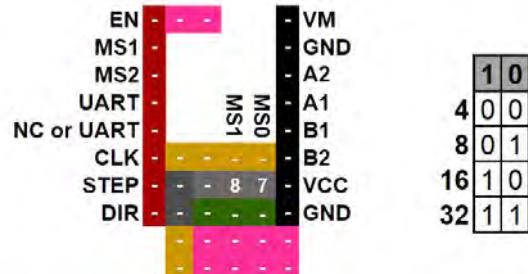
Driver Chip	MS1	MS0	Steps	Interpolation	Mode
BIQU® TMC2225 <small>OTP Mode Maximum 32 Subdivision 35V DC 2A (peak)</small>	Low	Low	1 / 4	1 / 256	spreadCycle
	Low	High	1 / 8	1 / 256	spreadCycle
	High	Low	1 / 16	1 / 256	spreadCycle
	High	High	1 / 32	1 / 256	spreadCycle

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: $V_{ref} = (I_{MAX} * 1.3846) * 0.90$
--	---	---

- See next page for the legend that belongs to the above chart.

OTP (One Time Programming Mode)

base of card



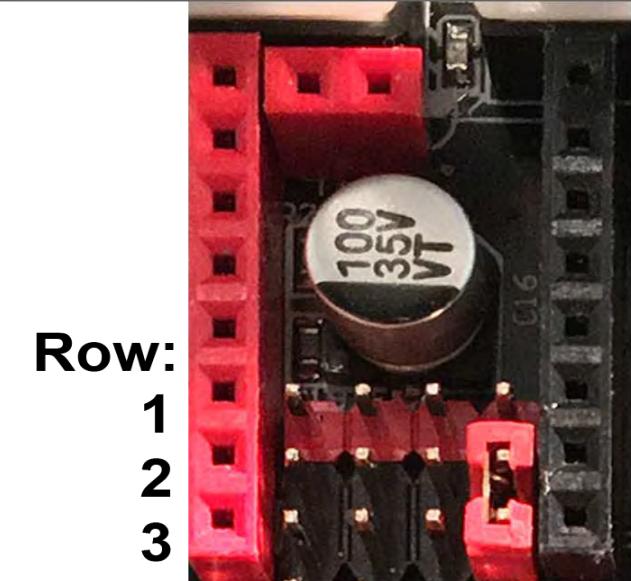
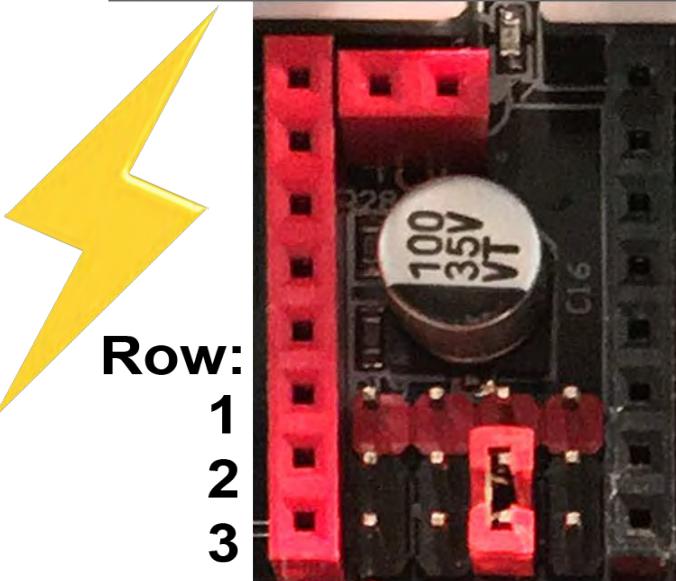
fans/heaters/usb side

See [Appendix D](#) for legend**STATUS:**
Not Yet
Tested

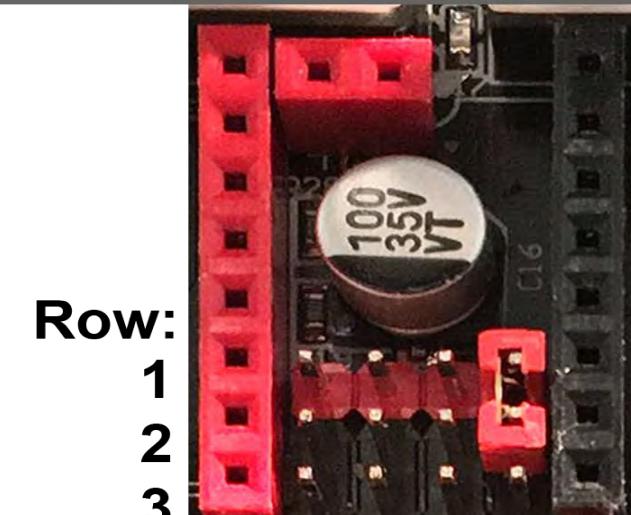
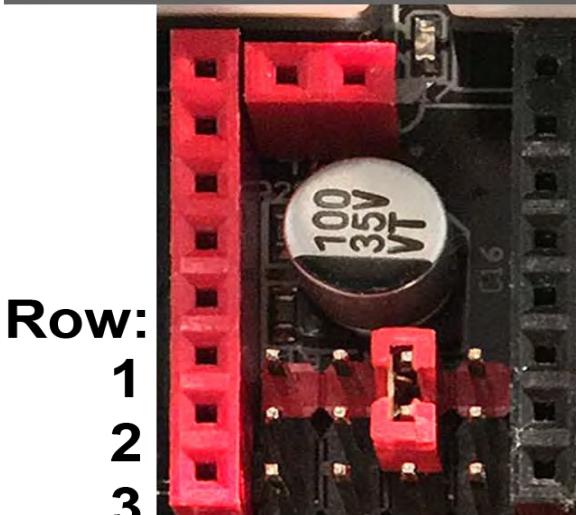
**This is a
Jumper:**

**BIQU TMC2225 V1.0****One Time Programming (OTP) Mode****SKR PRO V1.1 LEGEND for Binary State Stepper Drivers**

Low ➡ set Jumper between rows
2 and 3



High ➡ set Jumper between rows
1 and 2

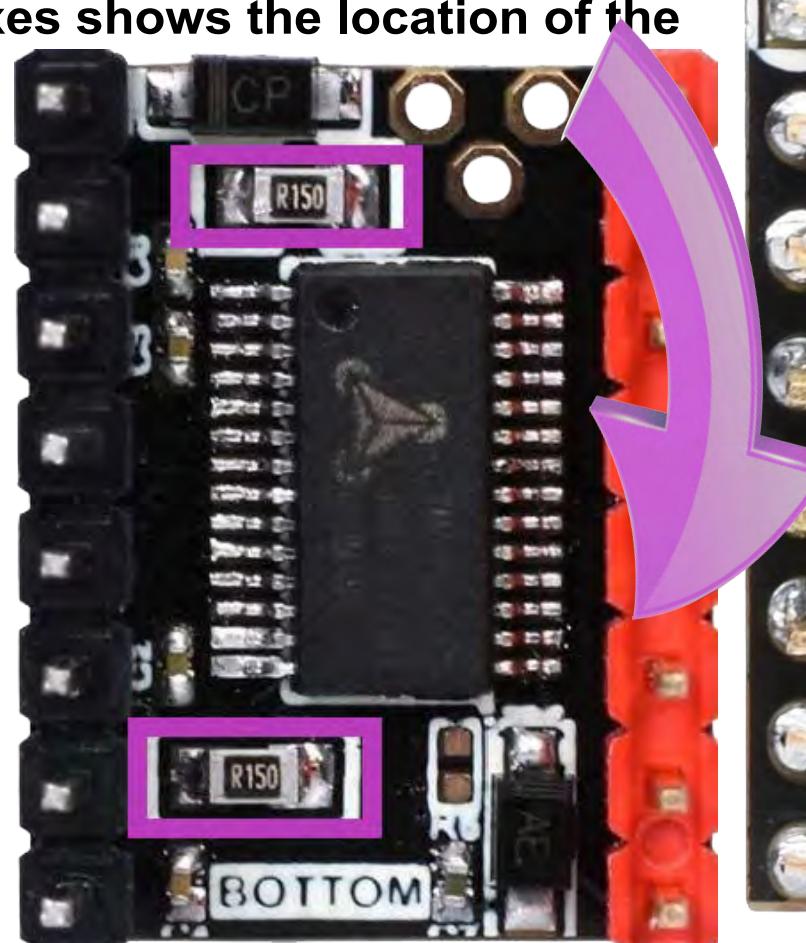


BIQU TMC2225 V1.0**One Time Programming (OTP) Mode**

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.

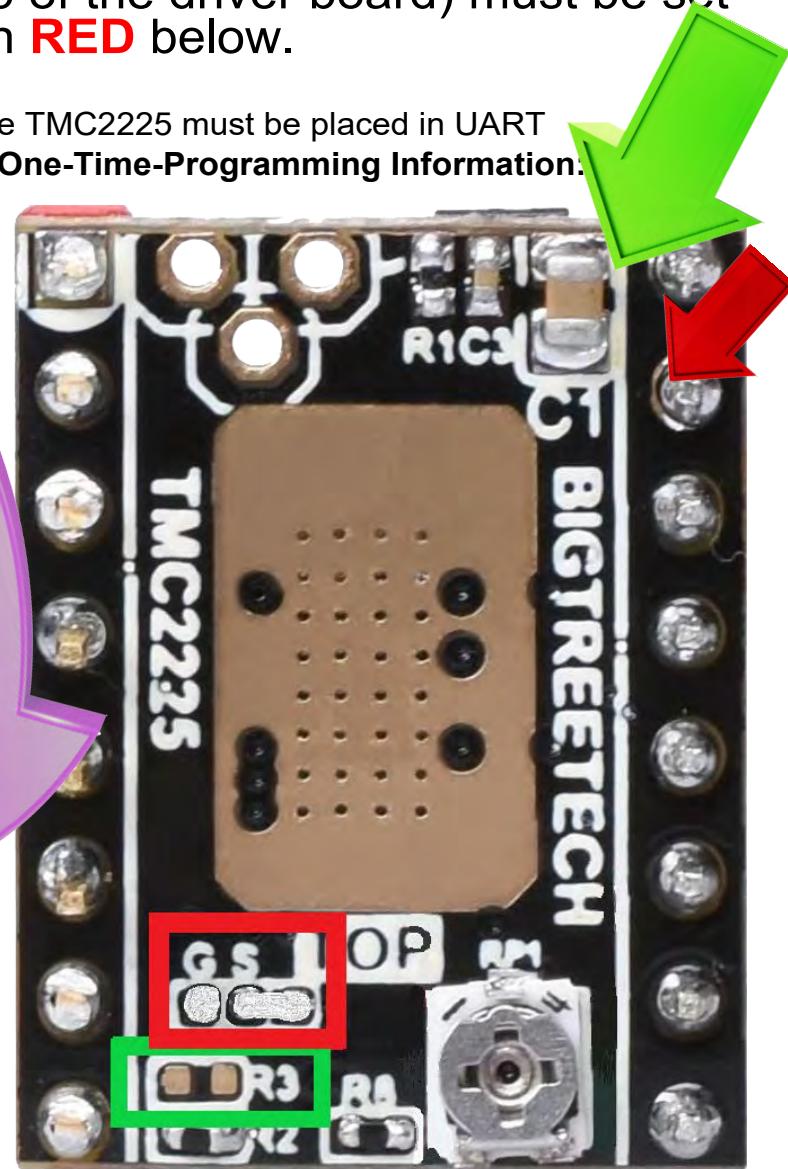
To do One-Time-Programming (OTP), the TMC2225 must be placed in UART mode. See **TMC220x Configurator for One-Time-Programming Information: [TMC220x Configurator](#).**

See [Appendix D](#) for legend



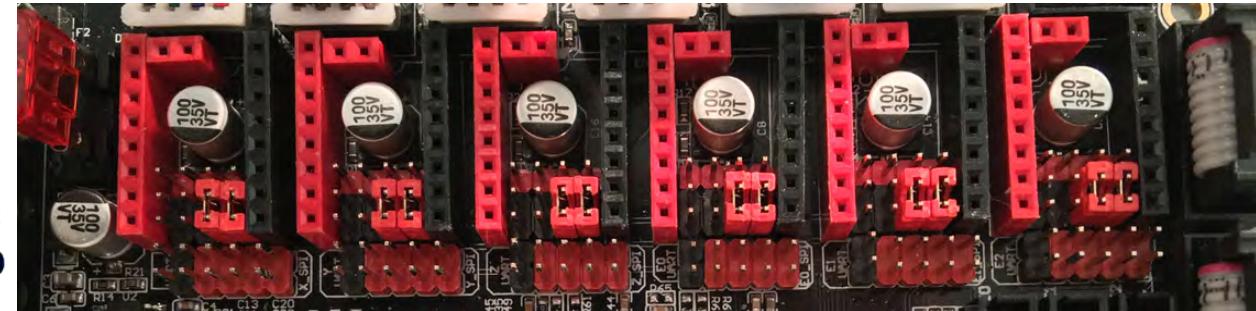
The **PURPLE** boxes shows the location of the current sense resistors (R_s).

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)



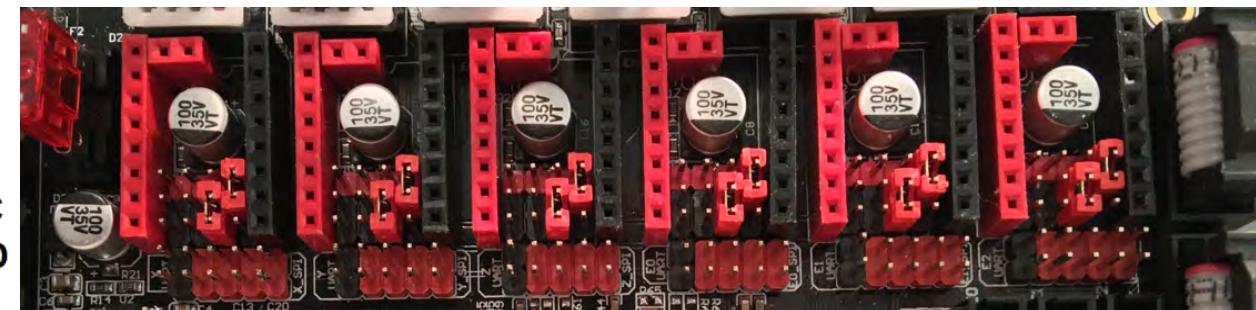
OTP (One Time Programming) Mode One Time Programming (OTP) Mode

OTP	NC or UART	EN	-	-	VM
1 / 4	NC or UART	MS1	-	-	GND
Interpolation:	UART	MS2	-	-	A2
1 / 256	UART	CLK	-	-	A1
SpreadCycle	STEP	STEP	8 7	-	B1
	DIR	DIR	8 7	-	B2
			-	-	VCC
			-	-	GND



See [Appendix D](#) for legend

OTP	NC or UART	EN	-	-	VM
1 / 8	NC or UART	MS1	-	-	GND
Interpolation:	UART	MS2	-	-	A2
1 / 256	UART	CLK	-	7	A1
SpreadCycle	STEP	STEP	-	8 7	B1
	DIR	DIR	-	8	B2
			-	-	VCC
			-	-	GND



See [Appendix D](#) for legend

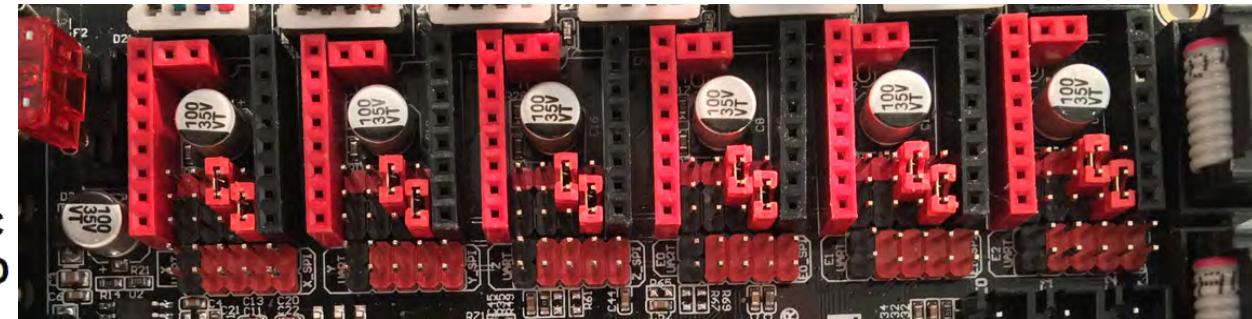
BIQU TMC2225 V1.0**OTP (One Time Programming) Mode** **One Time Programming (OTP) Mode**

OTP

1 / 16

NC or UART

EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	MS1	MS0	A1
CLK	-	8	-	B1
STEP	-	8	7	B2
DIR	-	-	7	VCC
	-	-	-	GND

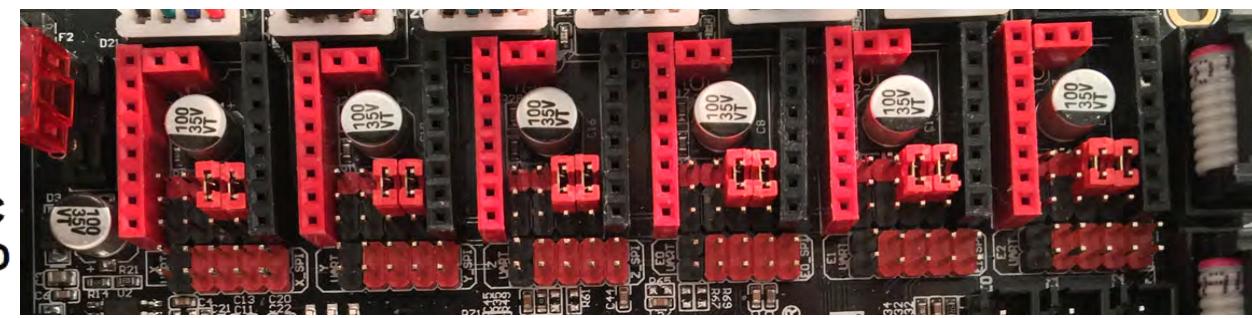
See [Appendix D](#) for legend

OTP

1 / 32

NC or UART

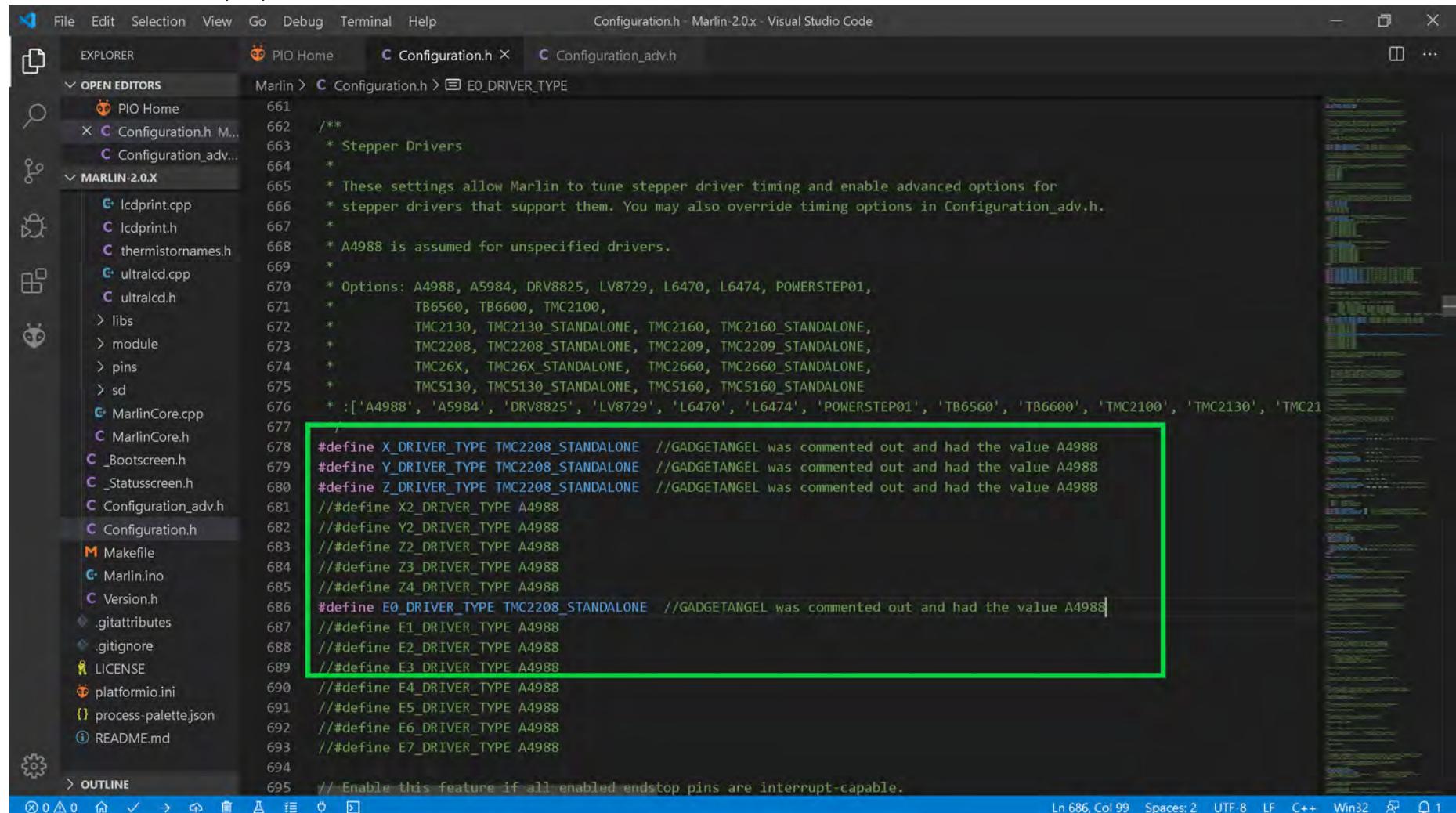
EN	-	-	-	VM
MS1	-	-	-	GND
MS2	-	-	-	A2
UART	-	MS1	MS0	A1
CLK	-	8	7	B1
STEP	-	8	7	B2
DIR	-	-	-	VCC
	-	-	-	GND

See [Appendix D](#) for legend

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 ("//").



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 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

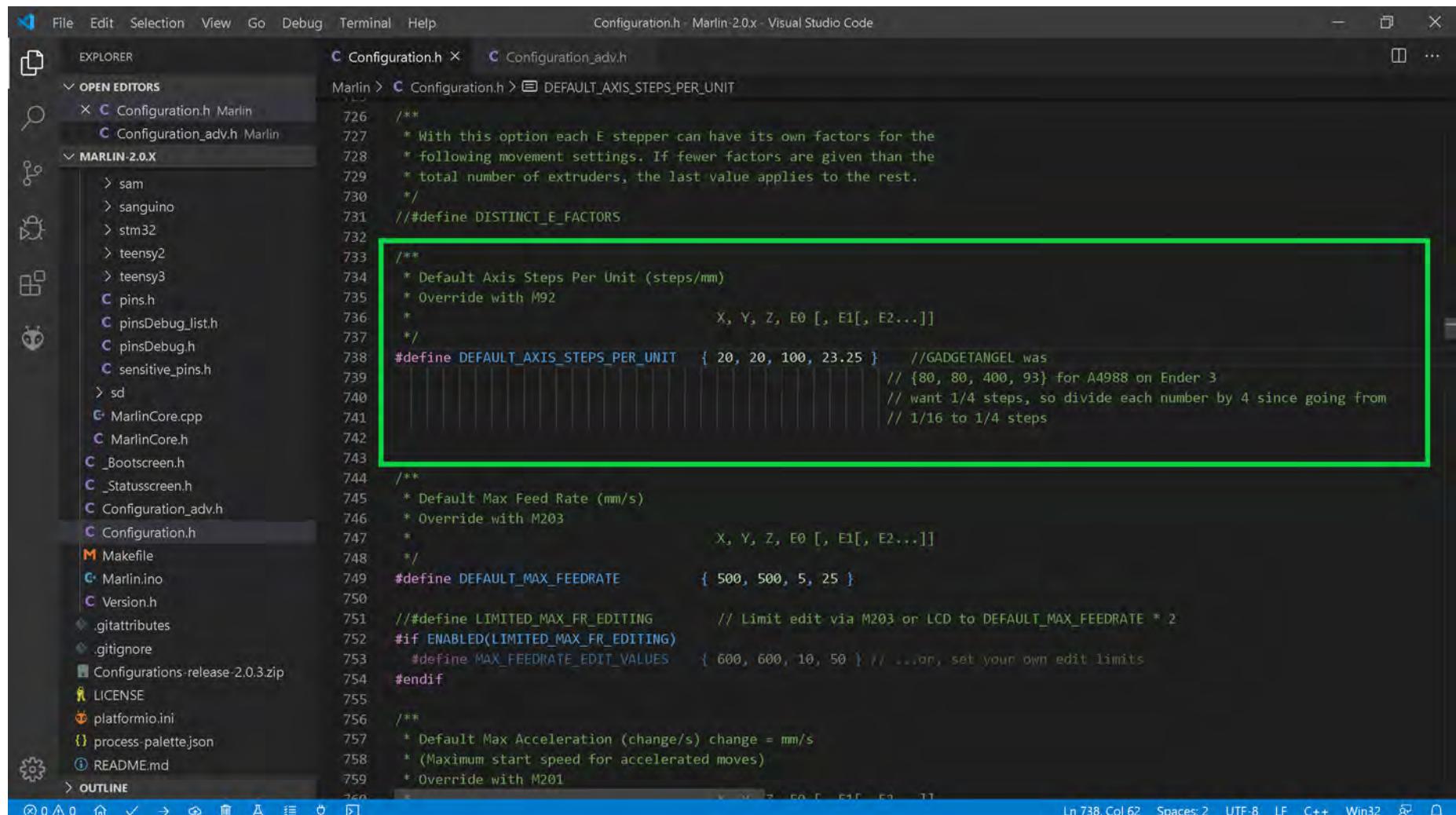
```

A green box highlights the lines for X, Y, Z, and E0, indicating they have been modified. The original values (A4988) are still present as comments.

- 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 OPEN EDITORS MARLIN-2.0.X

```

Configuration.h Configuration_adv.h
Marlin > Configuration.h > #define 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 * Override with M201
761 */

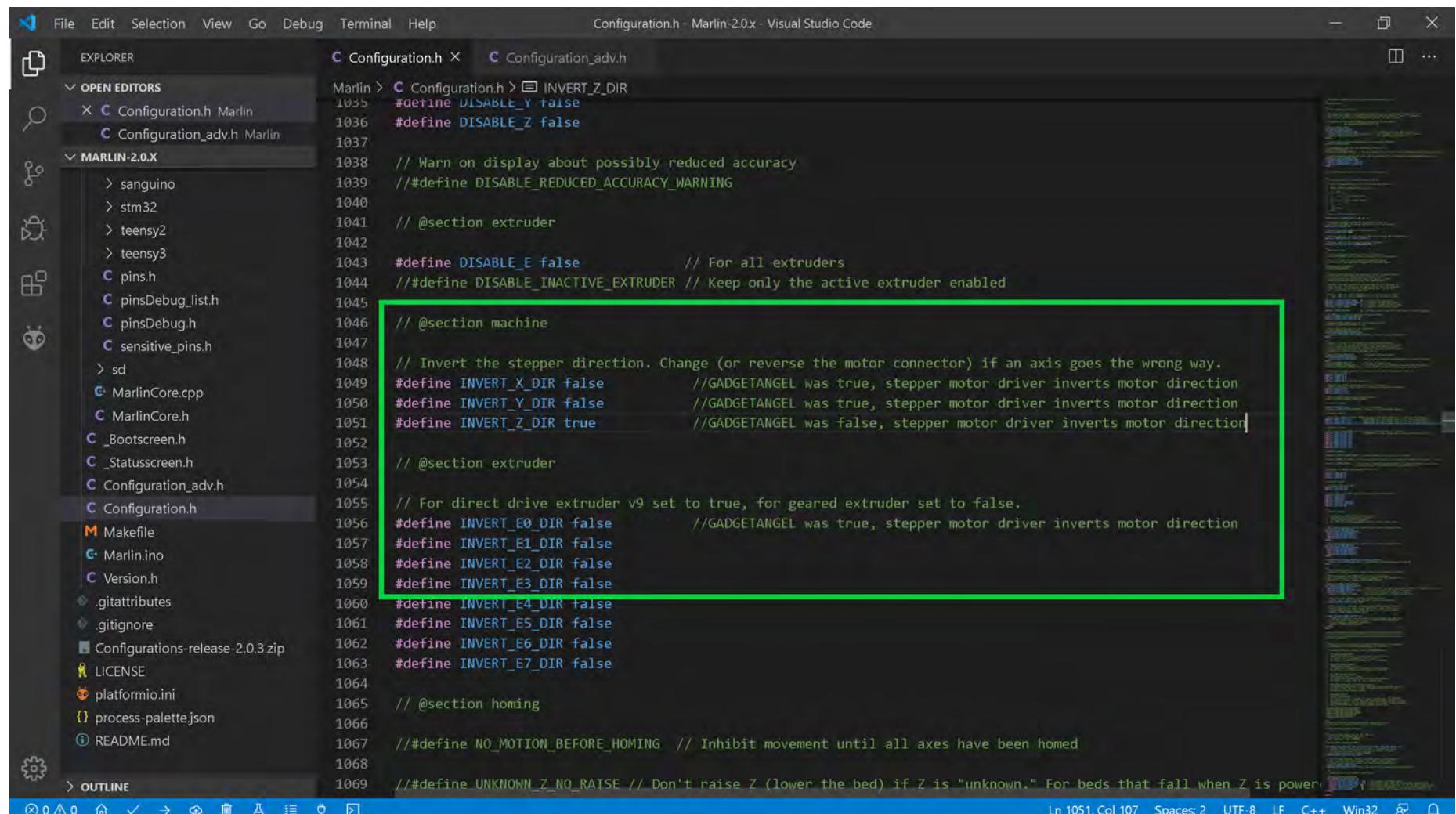
```

Ln 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 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



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 up

```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that the setting has been changed from its original value.

- 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.

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

EXPLORER Configuration.h X Configuration_adv.h

pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h

Marlin > Configuration.h > X_DRIVER_TYPE

```

558 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
559 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
560 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
561
562 /**
563 * Stepper Drivers
564 *
565 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
566 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
567 *
568 * A4988 is assumed for unspecified drivers.
569 *
570 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
571 * TB6560, TB6600, TMC2100,
572 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
573 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
574 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
575 * TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Task - Build +

STM32F407VE-L1...	IGNORED
BIGTREE_SKR_PRO	SUCCESS 00:02:31.294
ST7735R_Chip.h	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 =====	

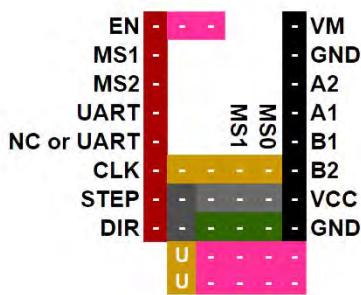
Terminal will be reused by tasks, press any key to close it.

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 [Appendix E](#).

BIQU TMC2225 V1.0UART Mode

base of card

UART Mode

fans/heaters/usb side

See [Appendix D](#) for legend

STATUS:
Not Yet
Tested

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

 **BIQU®**
TMC2225

UART Mode

Maximum 256
Subdivision

35V DC

2A (peak)

**Driving Current
Calculation
Formula**

 R_S (Typical Sense Resistor) = 0.15Ω

**Steps are set inside
of your Firmware**

$$I_{MAX} = V_{ref} * 0.7647$$

See Appendix B #7. Use 50% to 90%
as shown below:

$$I_{MAX} = (V_{ref} * 0.7647) * 0.90$$

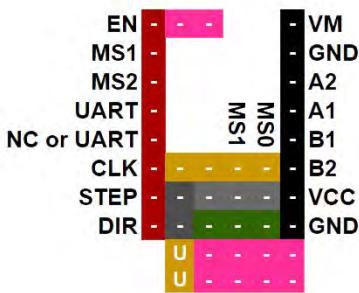
$$V_{ref} = I_{MAX} * 1.3077$$

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**

base of card

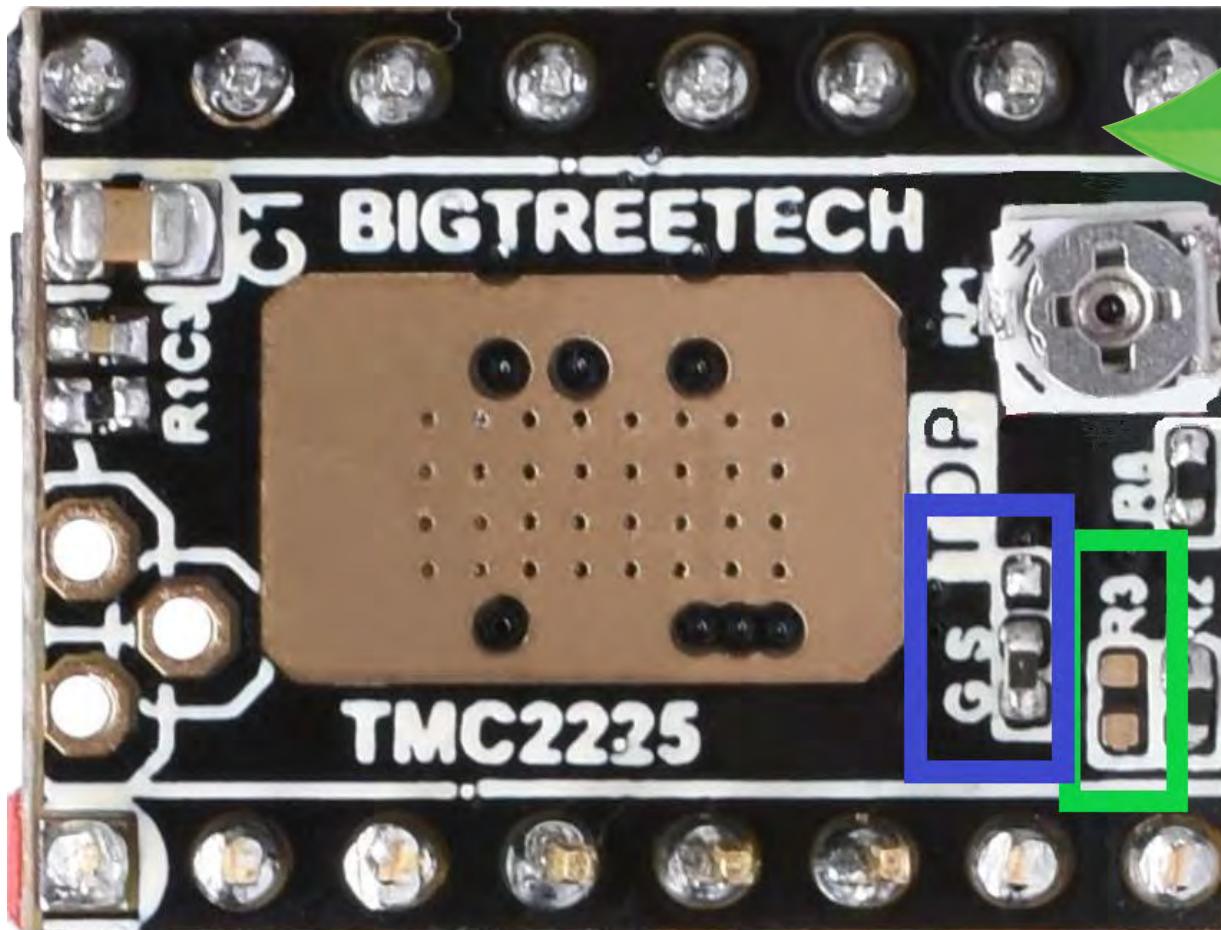
UART Mode

fans/heaters/usb side

See [Appendix D](#) for legend

STATUS:
Not Yet
Tested

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

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

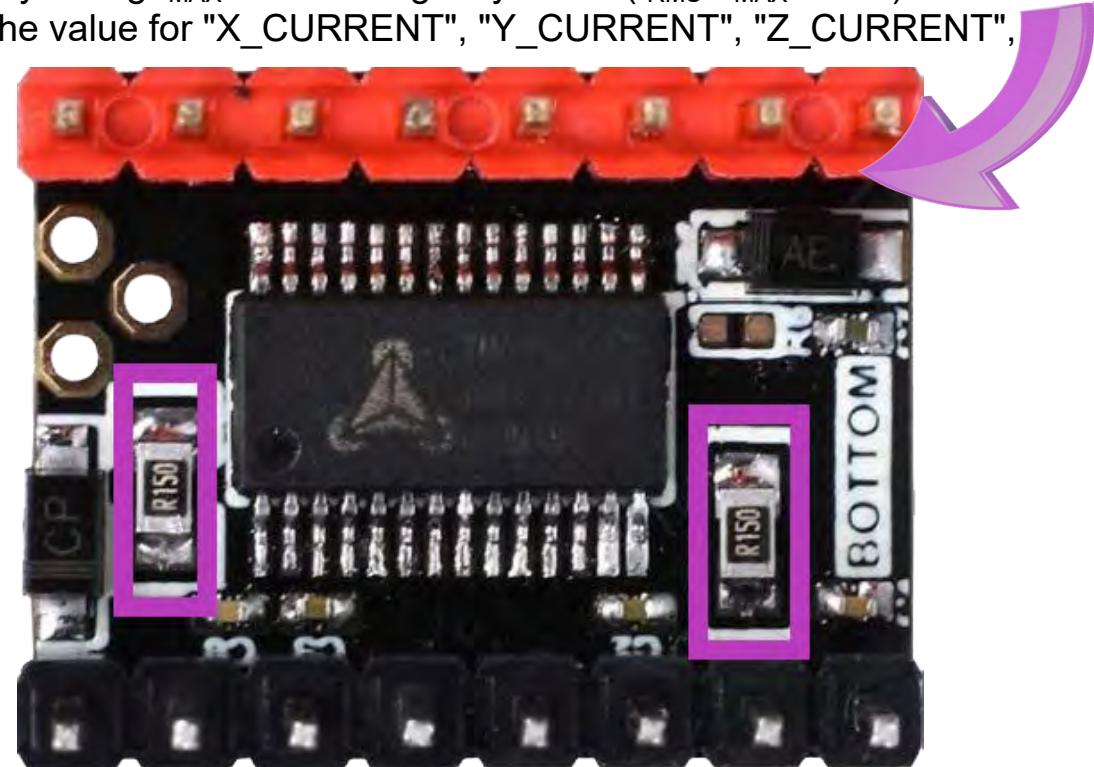
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

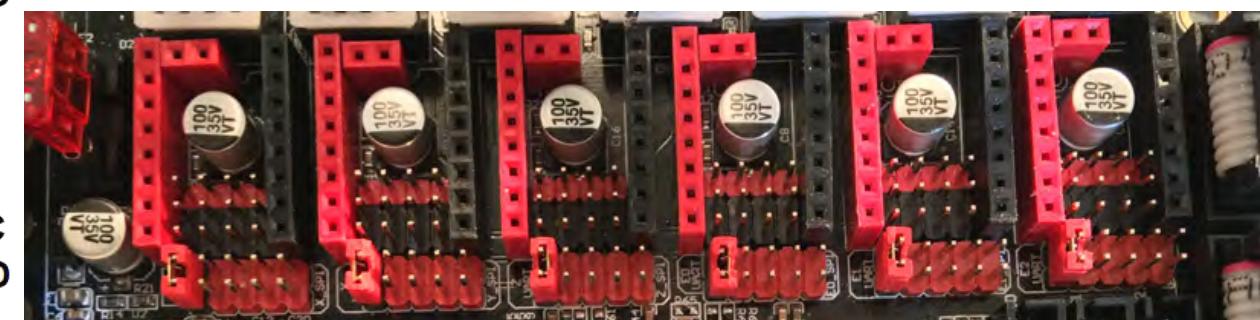
$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



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



See [Appendix D](#) for legend

UART

EN		-	VM
MS1		-	GND
MS2		-	A2
UART		MS0	A1
NC or UART		MS1	B1
CLK			B2
STEP			VCC
DIR			GND
	U		
	U		

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 file `Configuration.h` open. The code editor displays configuration settings for Marlin 2.0.x. A green rectangular box highlights the section of code responsible for inverting stepper directions:

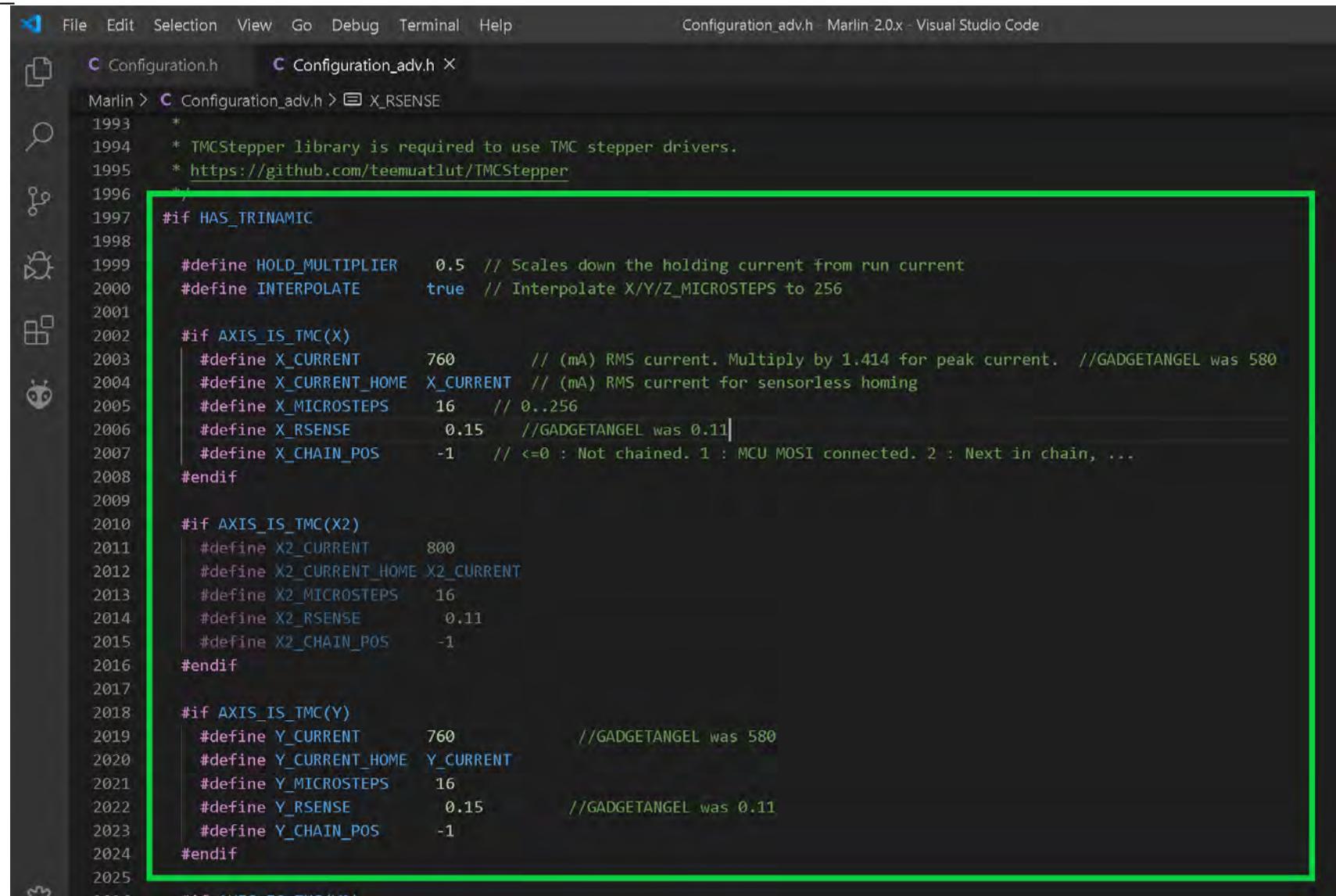
```
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // 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 indicates the code is at line 1051, column 107, in UTF-8 mode, using C++ syntax, and is running on a Win32 system.

- 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

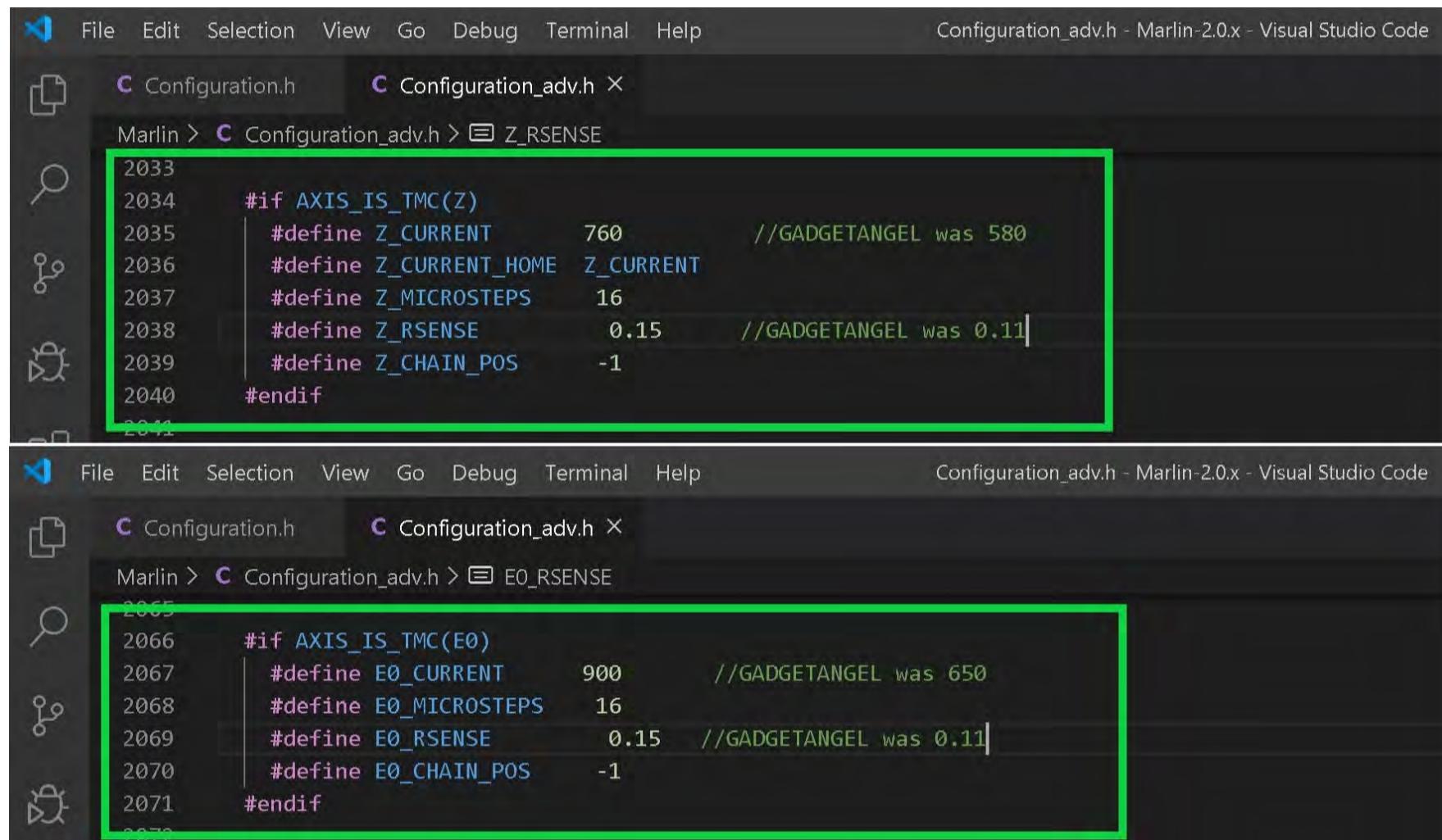
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_RSENSE
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current 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 //Lc 10/10/2021

```

- 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".



```

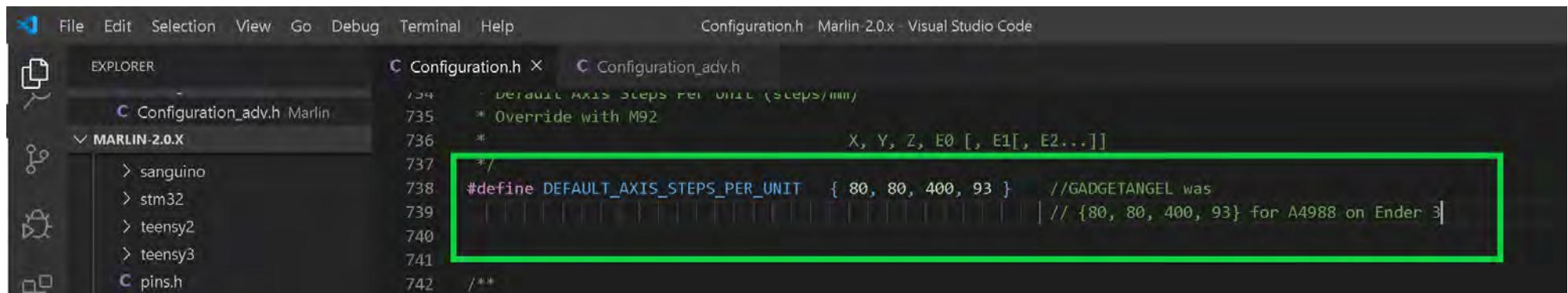
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z_RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035 | #define Z_CURRENT 760           //GADGETANGEL was 580
2036 | #define Z_CURRENT_HOME Z_CURRENT
2037 | #define Z_MICROSTEPS 16
2038 | #define Z_RSENSE 0.15          //GADGETANGEL was 0.11
2039 | #define Z_CHAIN_POS -1
2040 #endif
2041
2042
2043
2044
2045
2046 #if AXIS_IS_TMC(E0)
2047 | #define E0_CURRENT 900           //GADGETANGEL was 650
2048 | #define E0_MICROSTEPS 16
2049 | #define E0_RSENSE 0.15          //GADGETANGEL was 0.11
2050 | #define E0_CHAIN_POS -1
2051 #endif
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
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

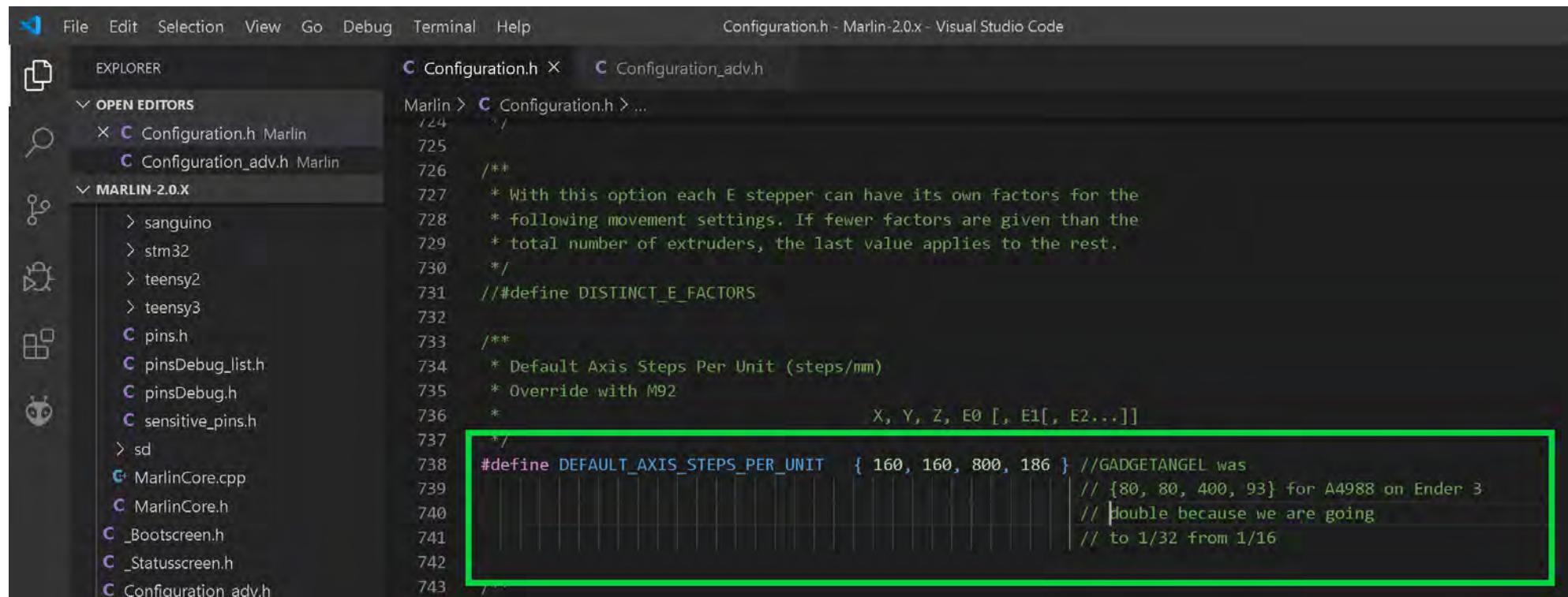
MARLIN-2.0.X

```

 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

MARLIN-2.0.X

```

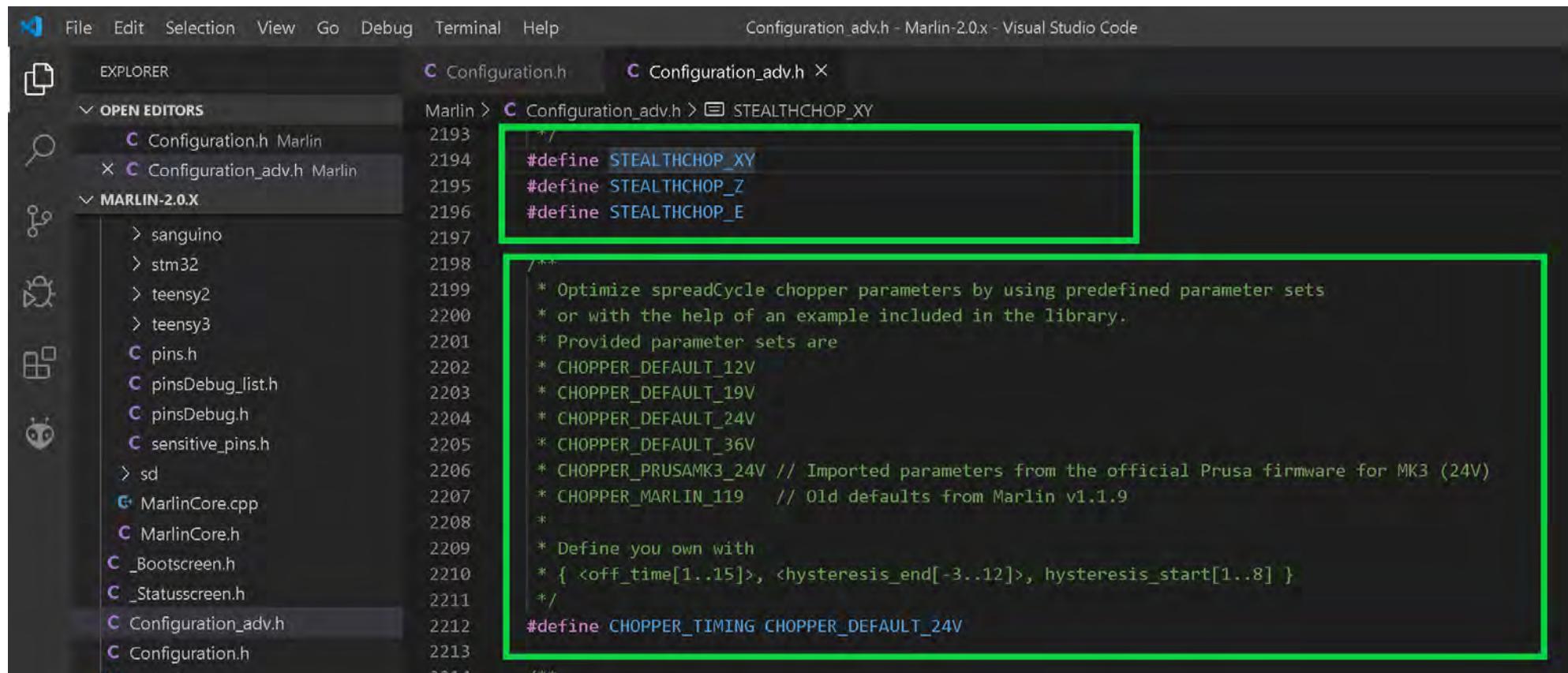
 724 */
 725
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 // #define DISTINCT_E_FACTORS
 732
 733 /**
 734 * Default 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

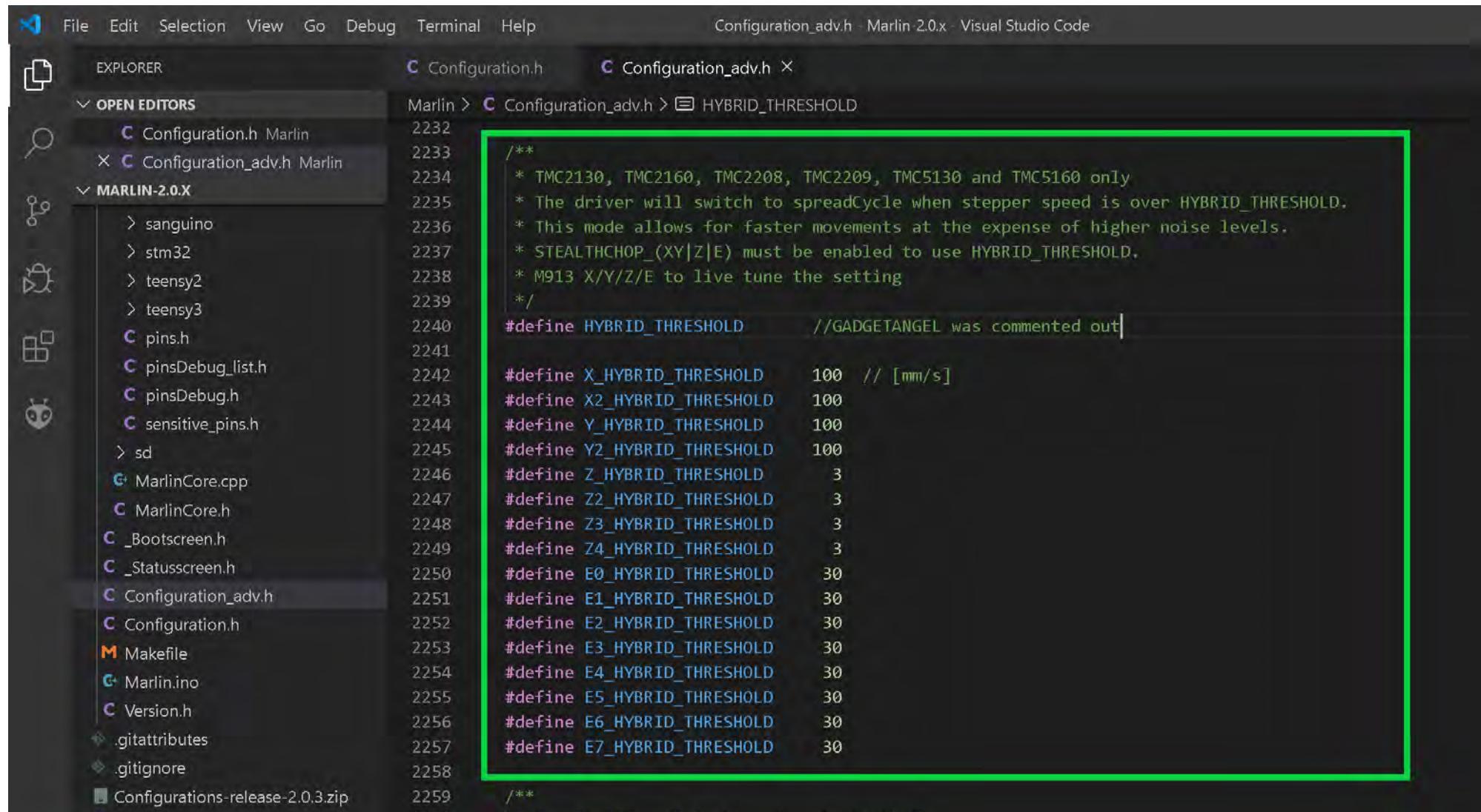
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  C pins.h
  C pinsDebug_list.h
  C pinsDebug.h
  C sensitive_pins.h
  > sd
  C MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C _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 /**
2214 */


```

- 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 */

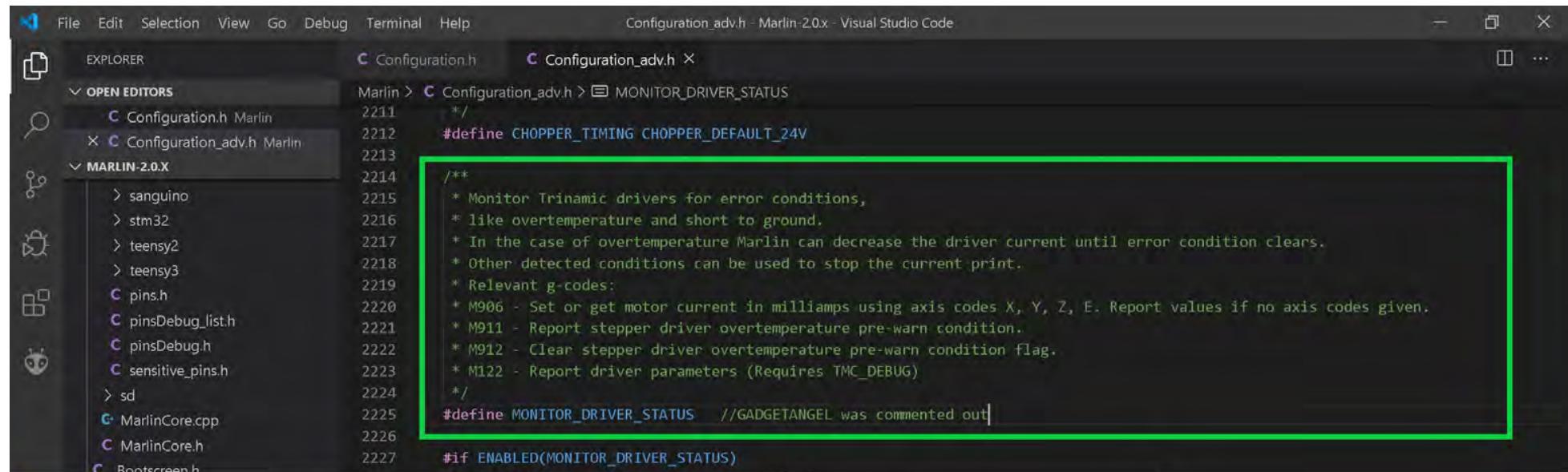
```

The line `#define HYBRID_THRESHOLD //GADGETANGEL was commented out` is highlighted with a green box, indicating it is the line that needs to be uncommented.

- 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

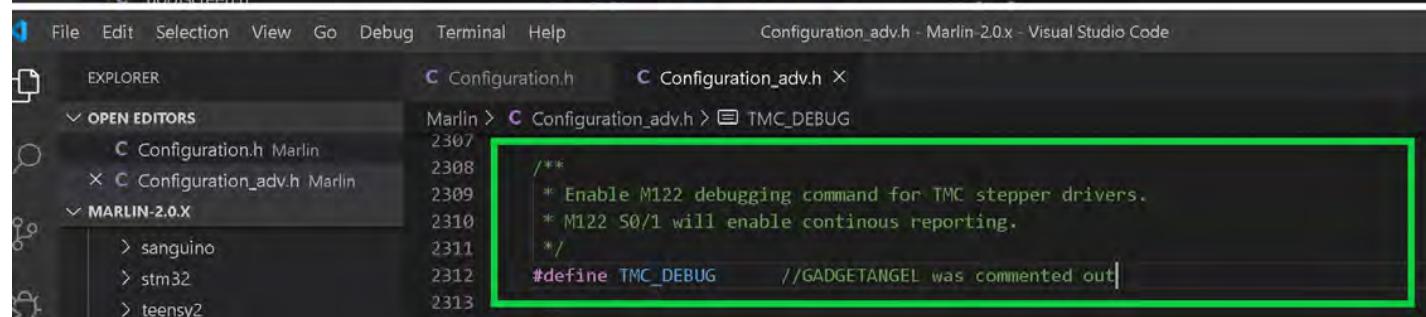
EXPLORER Configuration.h Configuration_adv.h

Marlin > Configuration.h > 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

Marlin > Configuration.h > 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.

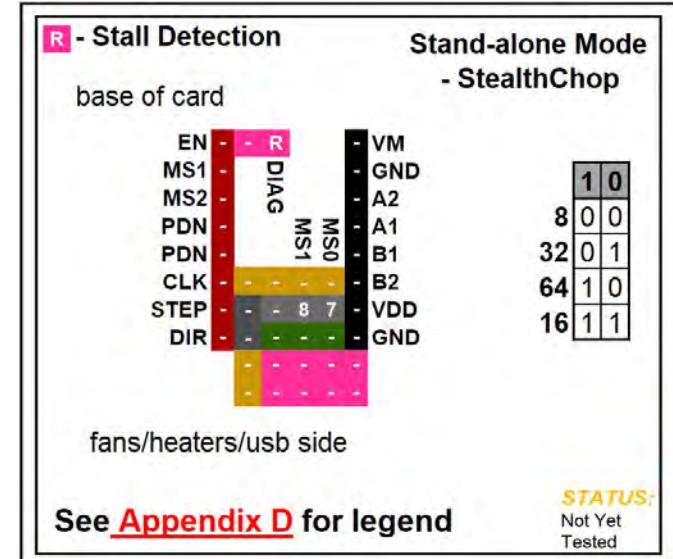
The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The Explorer sidebar on the left lists files and folders, including pins_BTT_SKR_PRO_V1_1.h, Configuration.h, Configuration_adv.h, and various pins and board configuration files. The main editor area displays code for stepper drivers, specifically for Marlin to tune stepper driver timing. The bottom right corner features a terminal window showing the output of a build process. The terminal output includes a table of build results for different boards, with the first row for BIGTREETECH SKR PRO highlighted by a green box. The terminal also shows a success message at the end.

Board	Status	Time
BIGTREETECH SKR PRO	SUCCESS	00:02:31.294
BIGTREETECH_V1_0	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 [Appendix E](#).

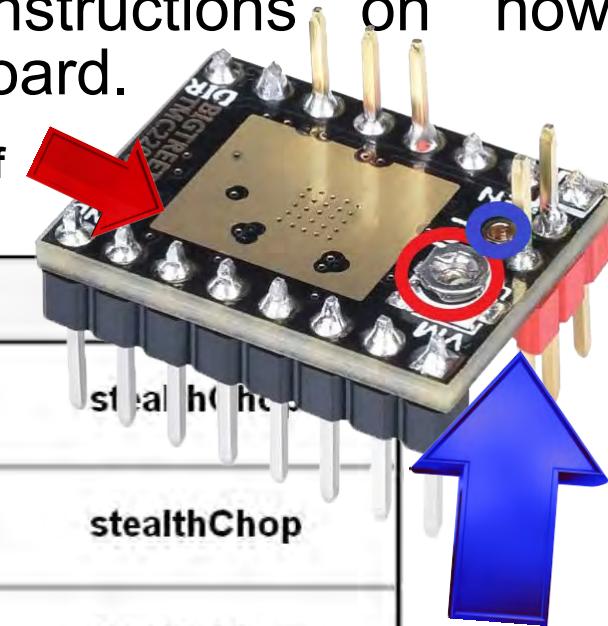


BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

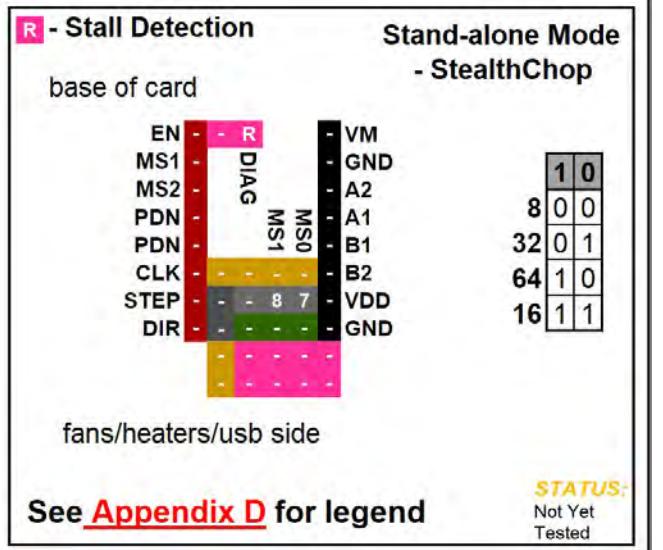
NOTE: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " V_{ref} Test point" location, as shown in **BLUE**, to set your V_{ref} . See **Appendix A** for instructions on how to set the V_{ref} on a driver board.

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



Driver Chip	MS1	MS0	Steps	Interpolation	
BIQU® TMC2209 <small>Stand Alone Mode Maximum 64 Subdivision 28V DC 2.8A (peak)</small>	Low	Low	1 / 8	1 / 256	stealthChop
	Low	High	1 / 32	1 / 256	stealthChop
	High	Low	1 / 64	1 / 256	stealthChop
	High	High	1 / 16	1 / 256	stealthChop
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: $I_{MAX} = I_{MAX} * 0.90$			$V_{ref} = I_{MAX}$ See Appendix B #5. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$	

- See next page for the legend that belongs to the above chart.

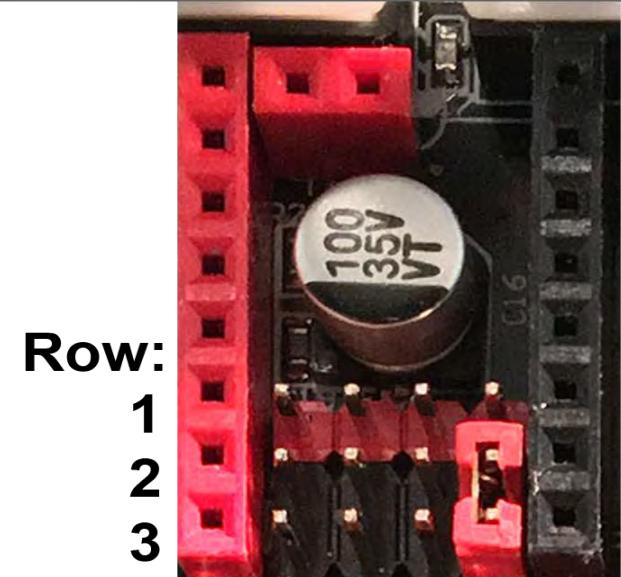
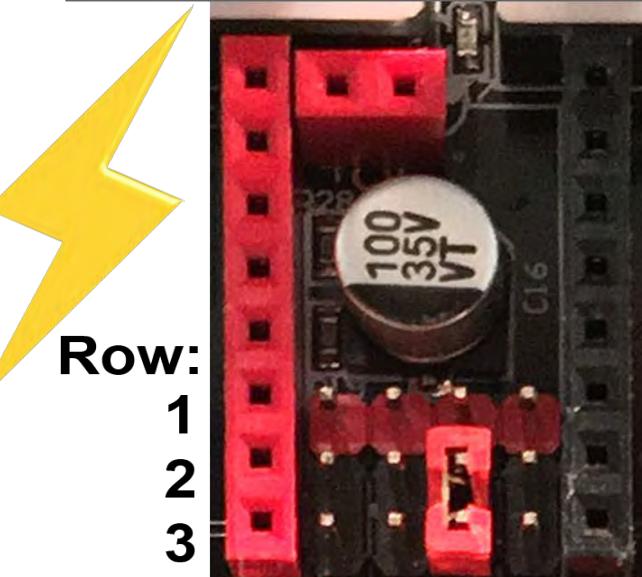


BIQU TMC2209 V1.2

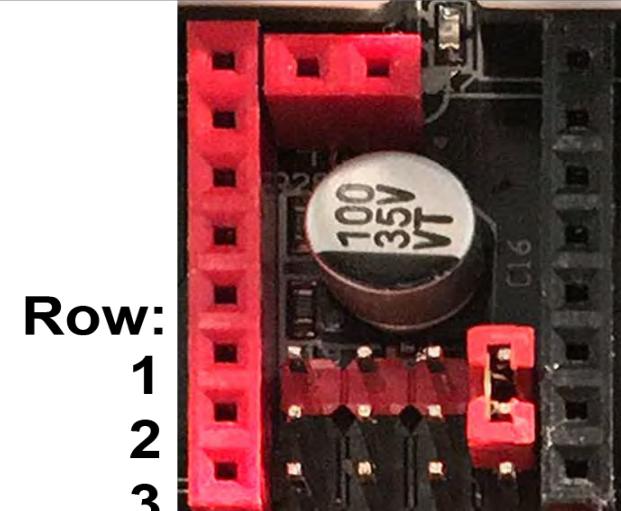
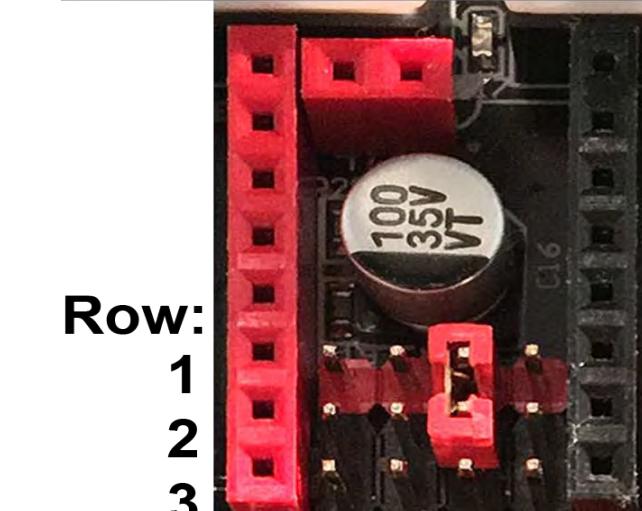
Stand-alone Mode for StealthChop

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows 2 and 3



High ➡ set Jumper between rows 1 and 2

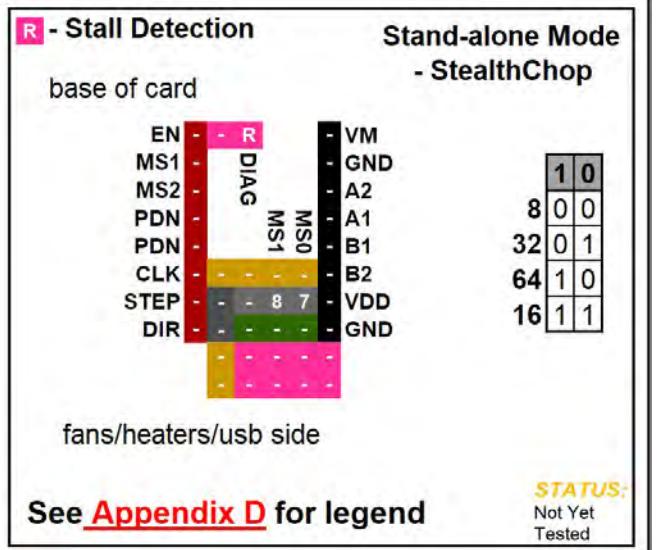


This is a Jumper:



BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

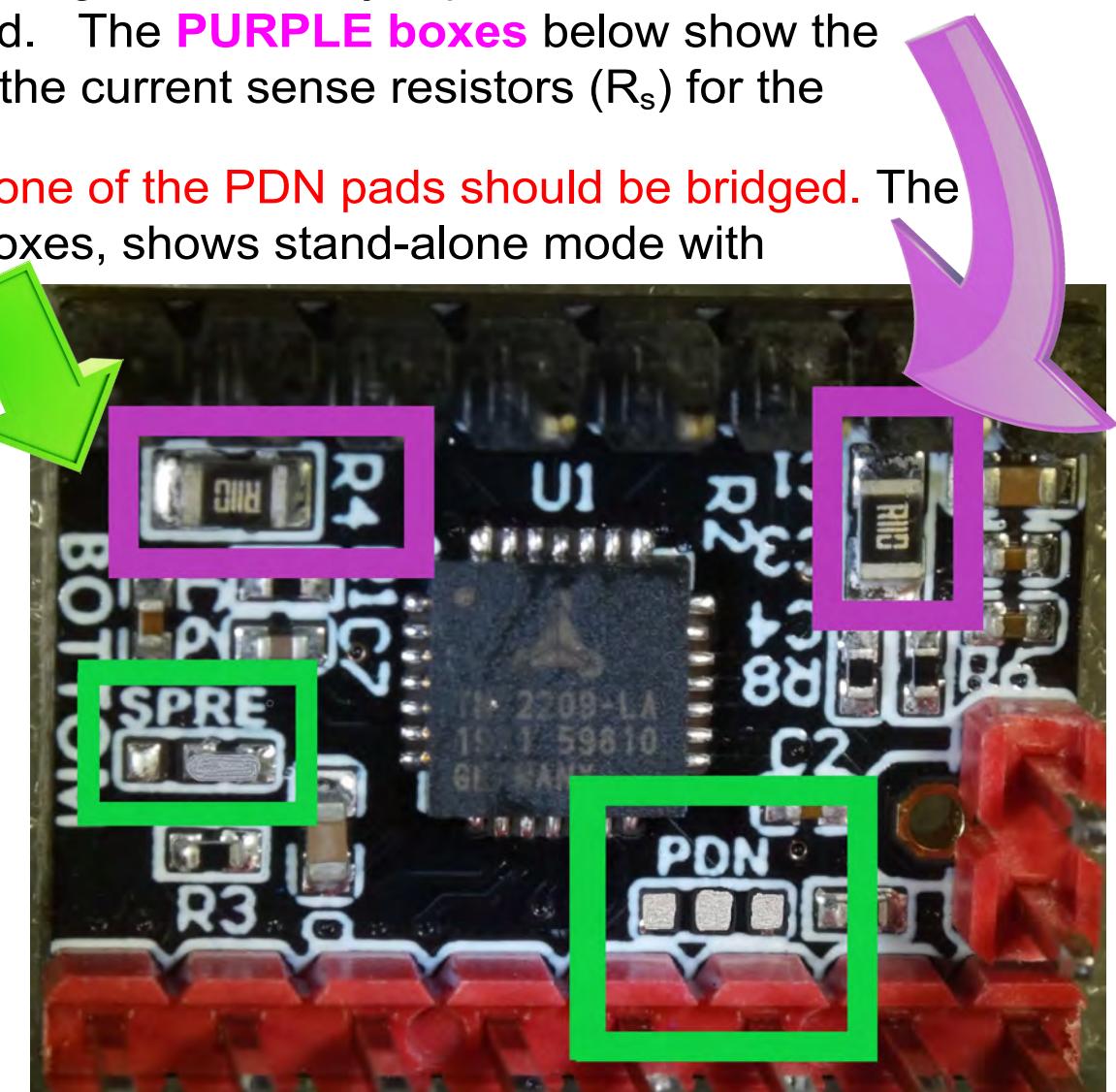


NOTE: The SPRE jumper is located on the bottom of the driver board. In stand-alone mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board. The **PURPLE boxes**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



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

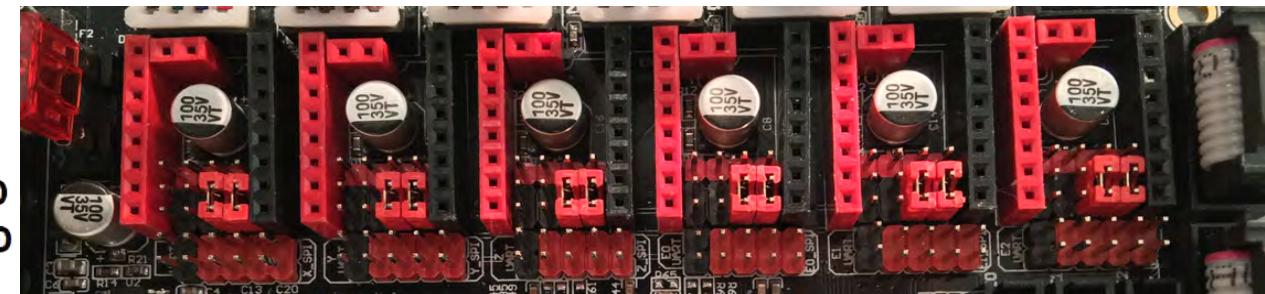
**Stand-alone
Mode**

1 / 8

Interpolation:
1 / 256

StealthChop

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	-	-	B2
STEP	-	-	8 7	VDD
DIR	-	8 7	-	GND
	-	-	-	-



See [Appendix D](#) for legend

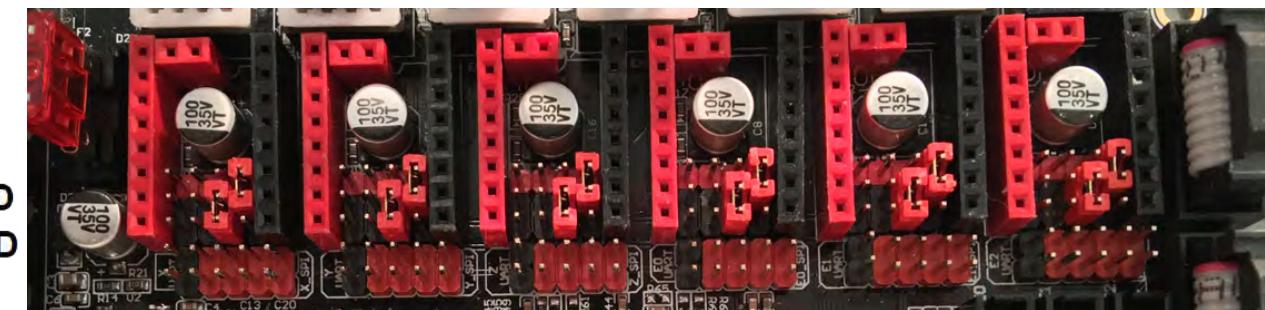
**Stand-alone
Mode**

1 / 32

Interpolation:
1 / 256

StealthChop

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	-	7	B2
STEP	-	-	8 7	VDD
DIR	-	8	-	GND
	-	-	-	-



See [Appendix D](#) for legend

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

Stand-alone
Mode

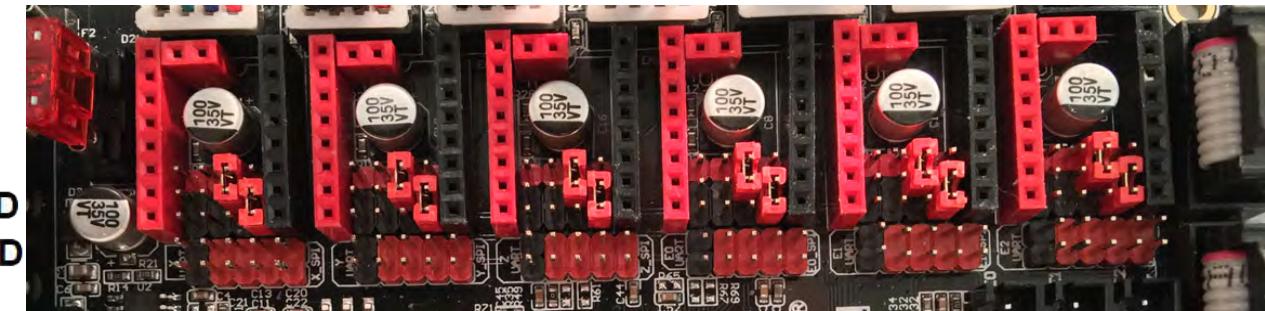
1 / 64

Interpolation:

1 / 256

StealthChop

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	8	-	B2
STEP	-	8 7	-	VDD
DIR	-	7	-	GND
	-		-	
	-		-	



See [Appendix D](#) for legend

Stand-alone
Mode

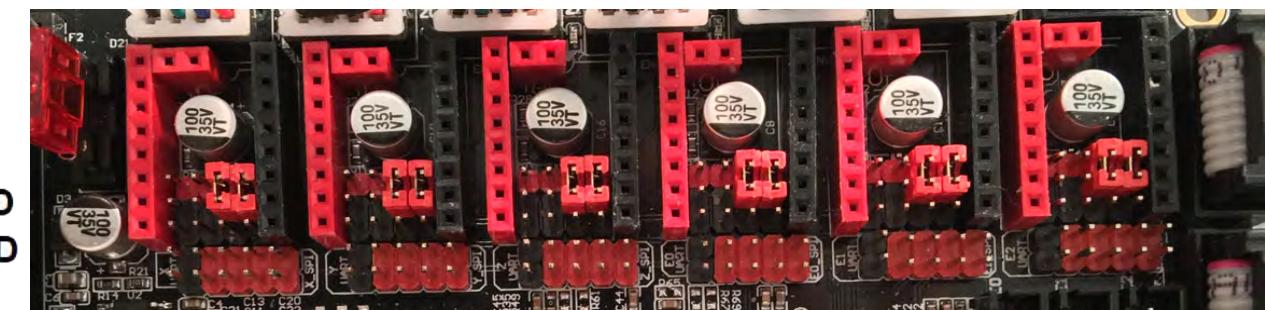
1 / 16

Interpolation:

1 / 256

StealthChop

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	8 7	-	B2
STEP	-	8 7	-	VDD
DIR	-	7	-	GND
	-		-	
	-		-	

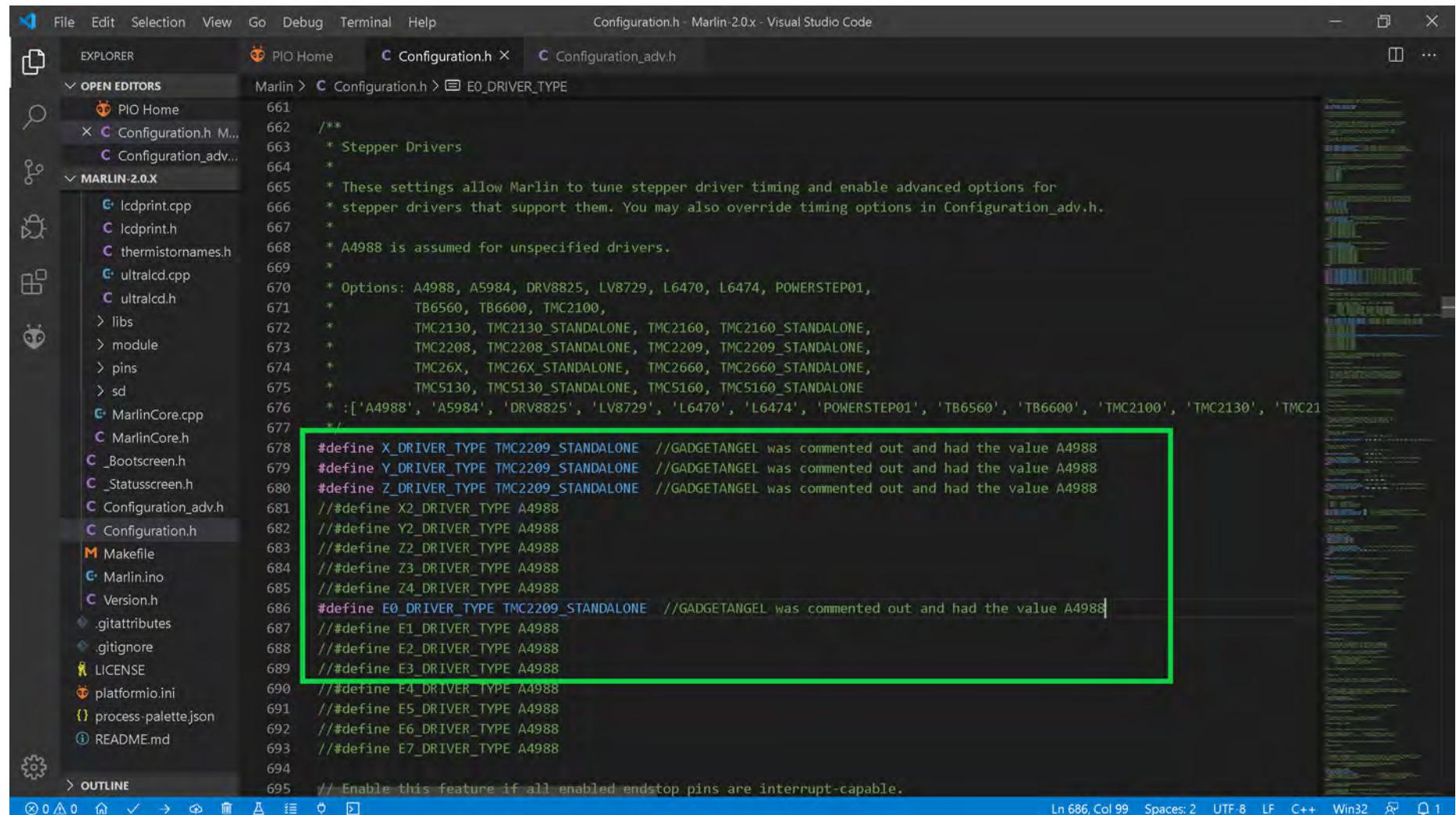


See [Appendix D](#) for legend

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.

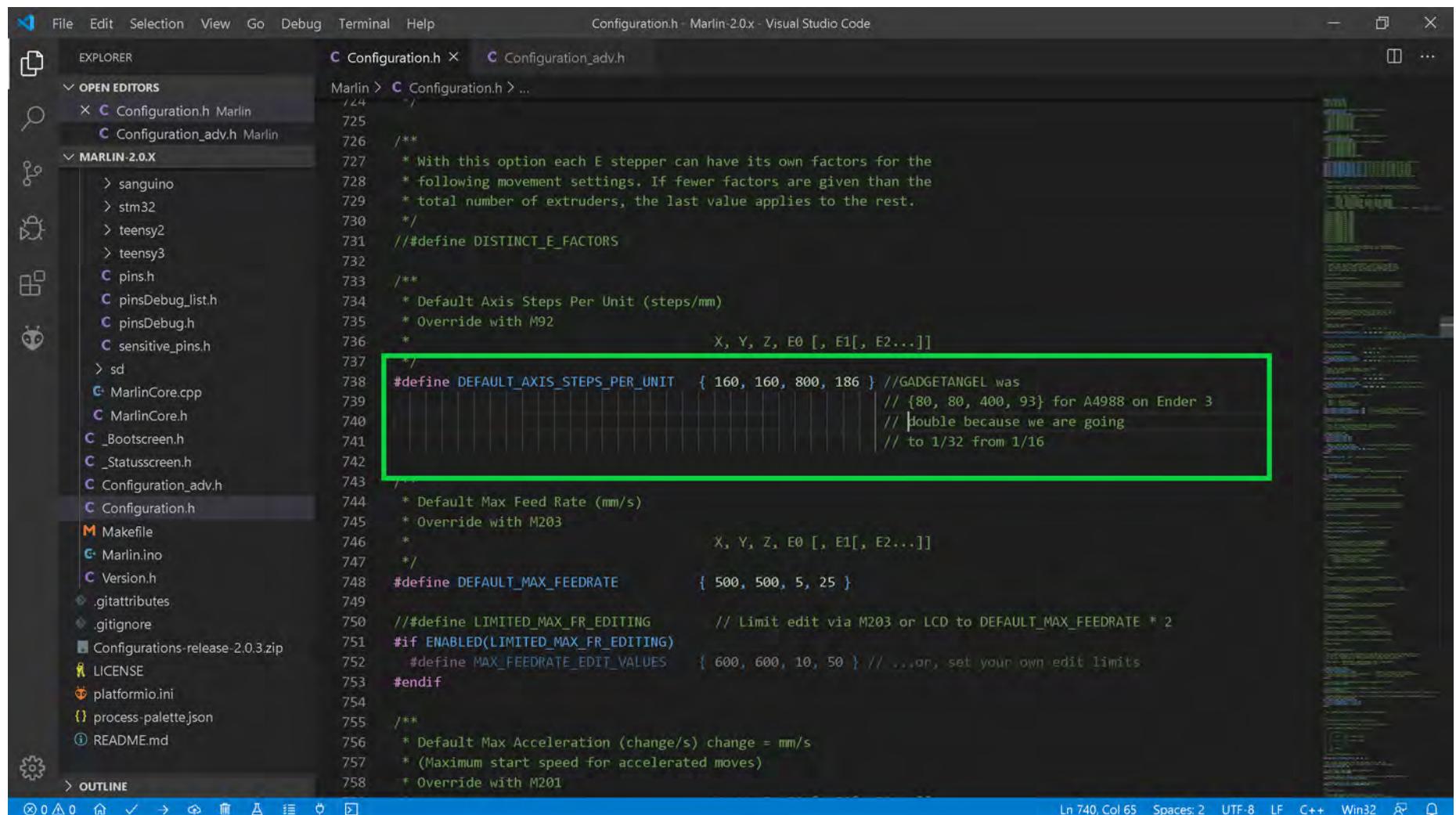
```

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. The rest of the code is standard Marlin configuration for TMC2209 drivers in stand-alone mode.

- 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 file `Configuration.h` open. The code editor displays the Marlin configuration header. A green rectangular box highlights the following line of code:

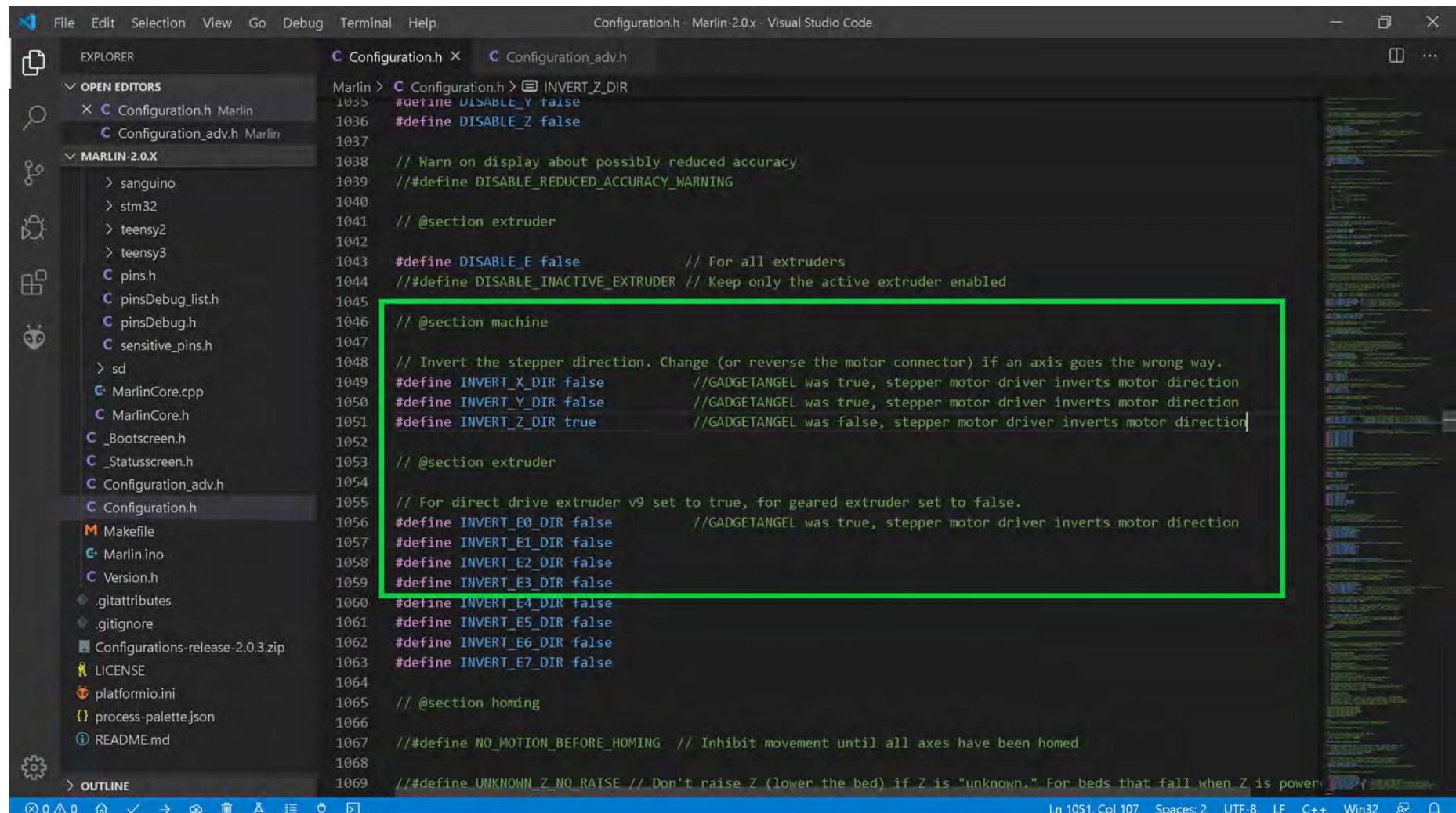
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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), and file type (C++). The left sidebar shows the project structure with files like `Configuration.h`, `Configuration_adv.h`, and `MarlinCore.cpp`.

- 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 file `Configuration.h` open. The code editor displays the following snippet:

```

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

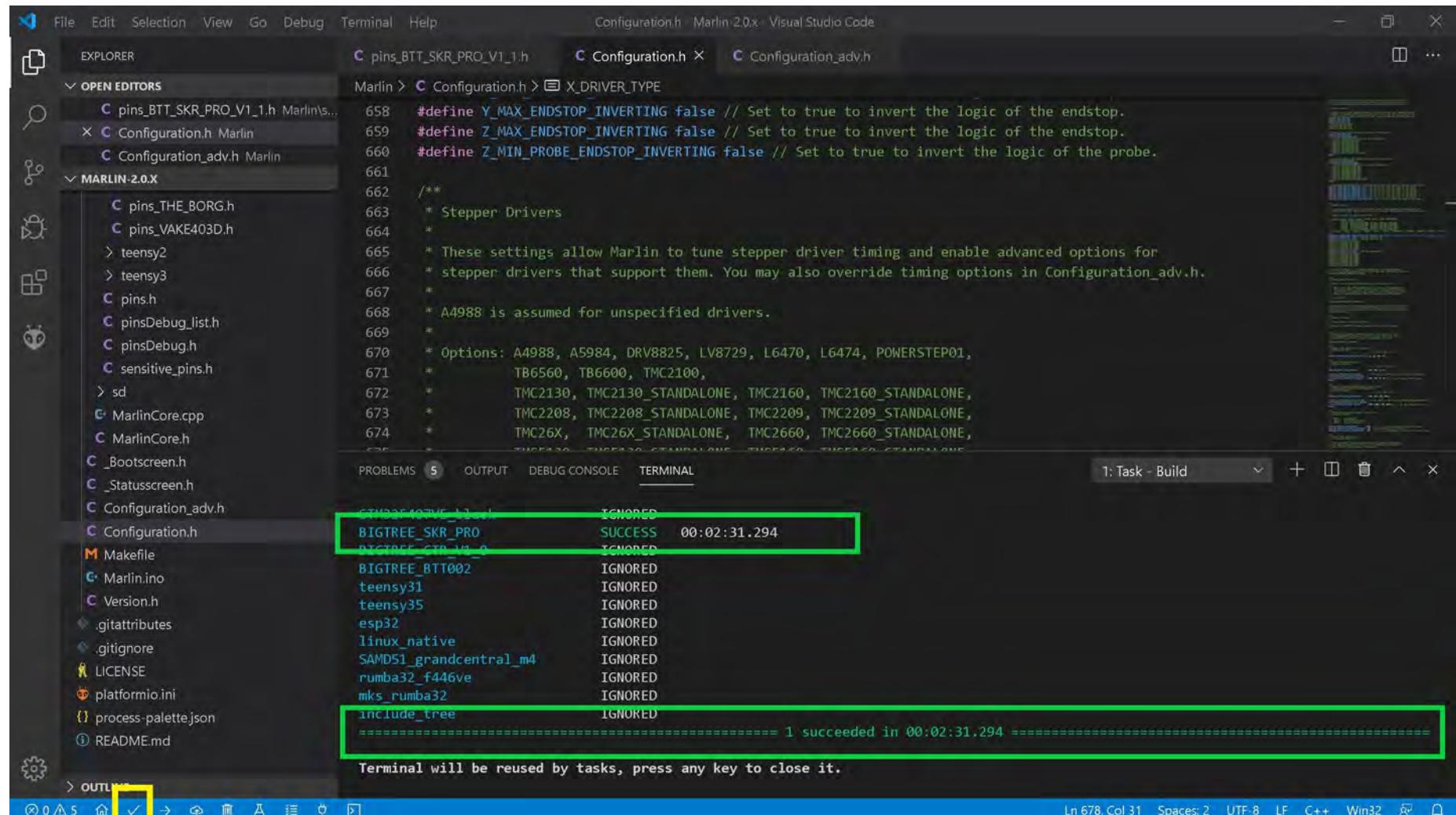
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating it is the setting to change.

- 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.

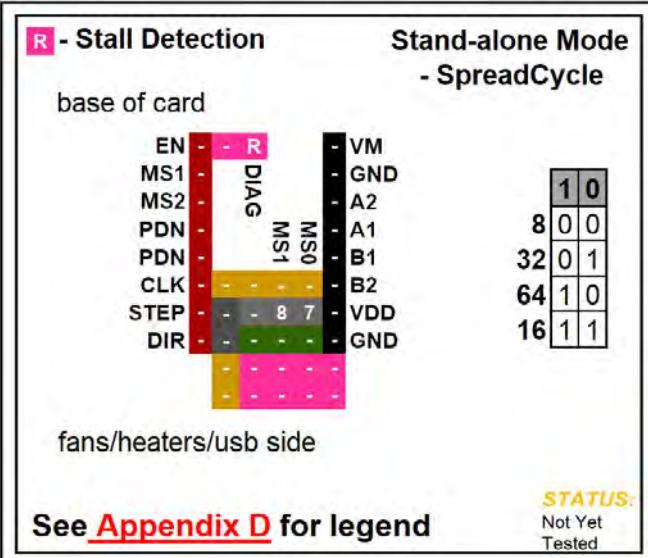


```

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 + □ ×
STM32F107VE_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 [Appendix E](#).

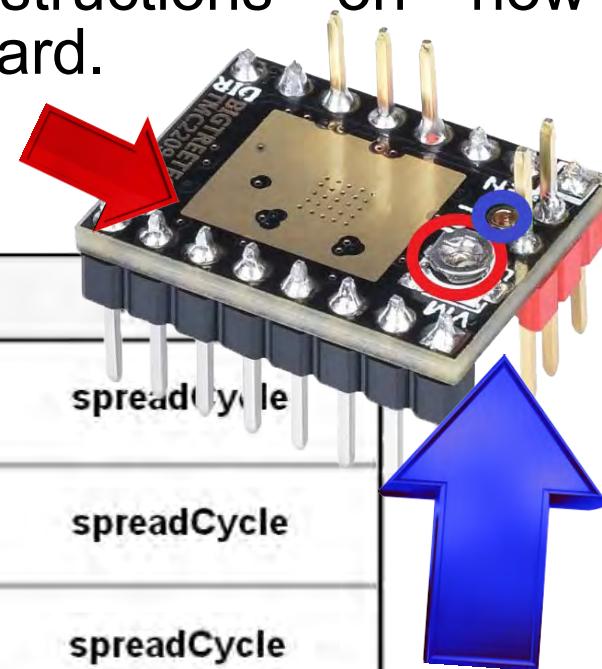


BIQU TMC2209 V1.2

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.

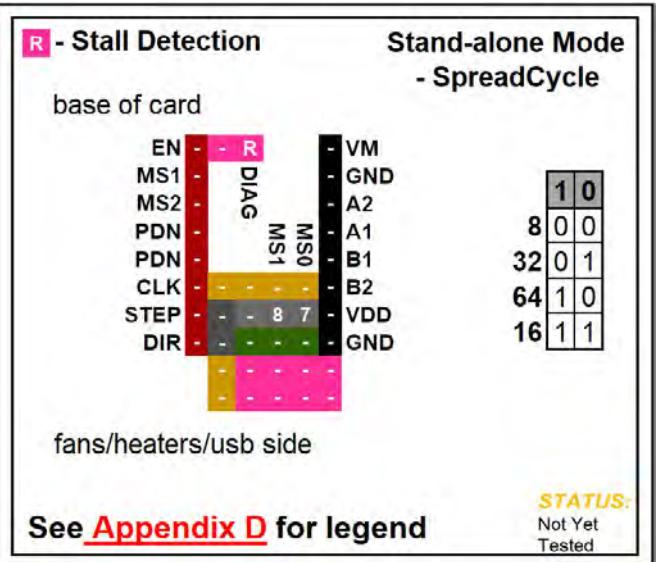
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.



Driver Chip	MS1	MS0	Steps	Interpolation	
BIQU® TMC2209 <small>Stand Alone Mode Maximum 64 Subdivision 28V DC 2.8A (peak)</small>	Low	Low	1 / 8	1 / 256	spreadCycle
	Low	High	1 / 32	1 / 256	spreadCycle
	High	Low	1 / 64	1 / 256	spreadCycle
	High	High	1 / 16	1 / 256	spreadCycle

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: $I_{MAX} = I_{MAX} * 0.90$	$V_{ref} = I_{MAX}$ See Appendix B #5. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$
---	--	--

- See next page for the legend that belongs to the above chart.

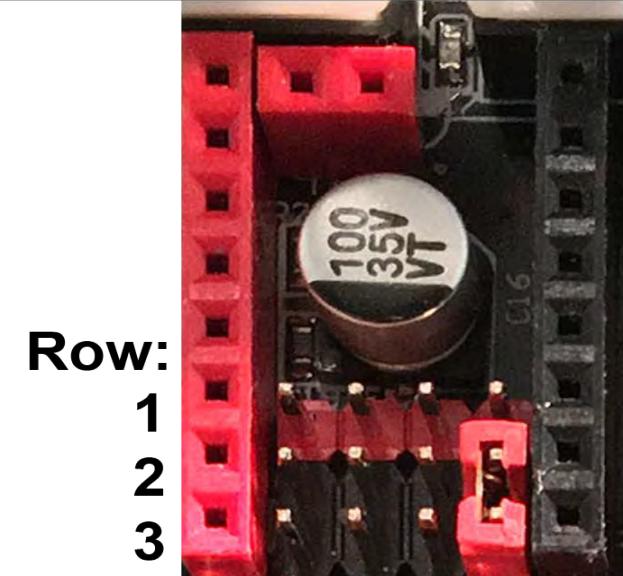
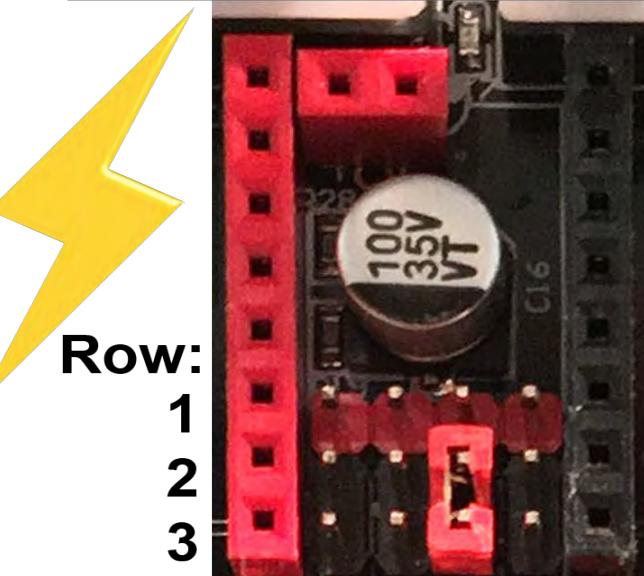


BIQU TMC2209 V1.2

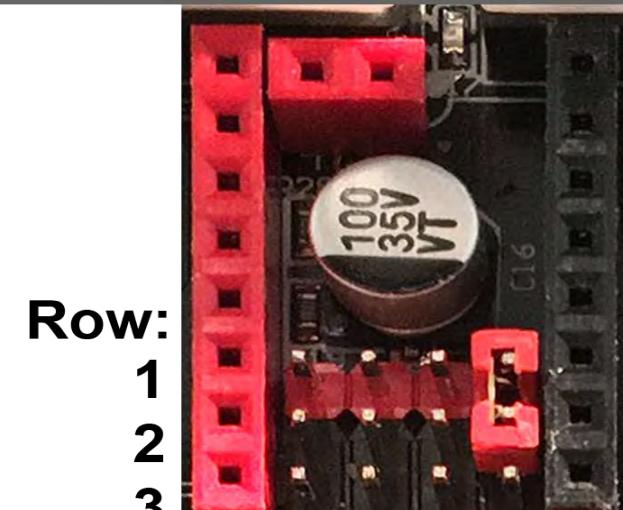
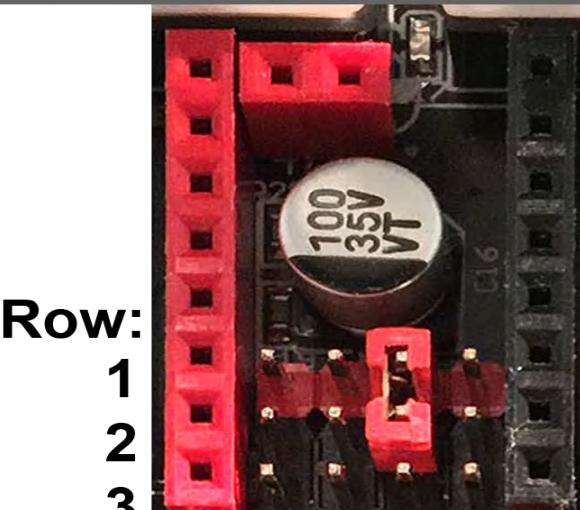
Stand-alone Mode for SpreadCycle

SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ➡ set Jumper between rows 2 and 3

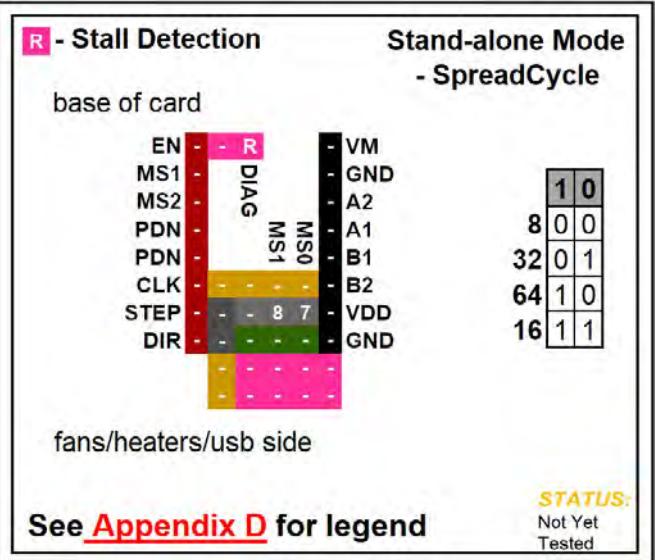


High ➡ set Jumper between rows 1 and 2



This is a
Jumper:



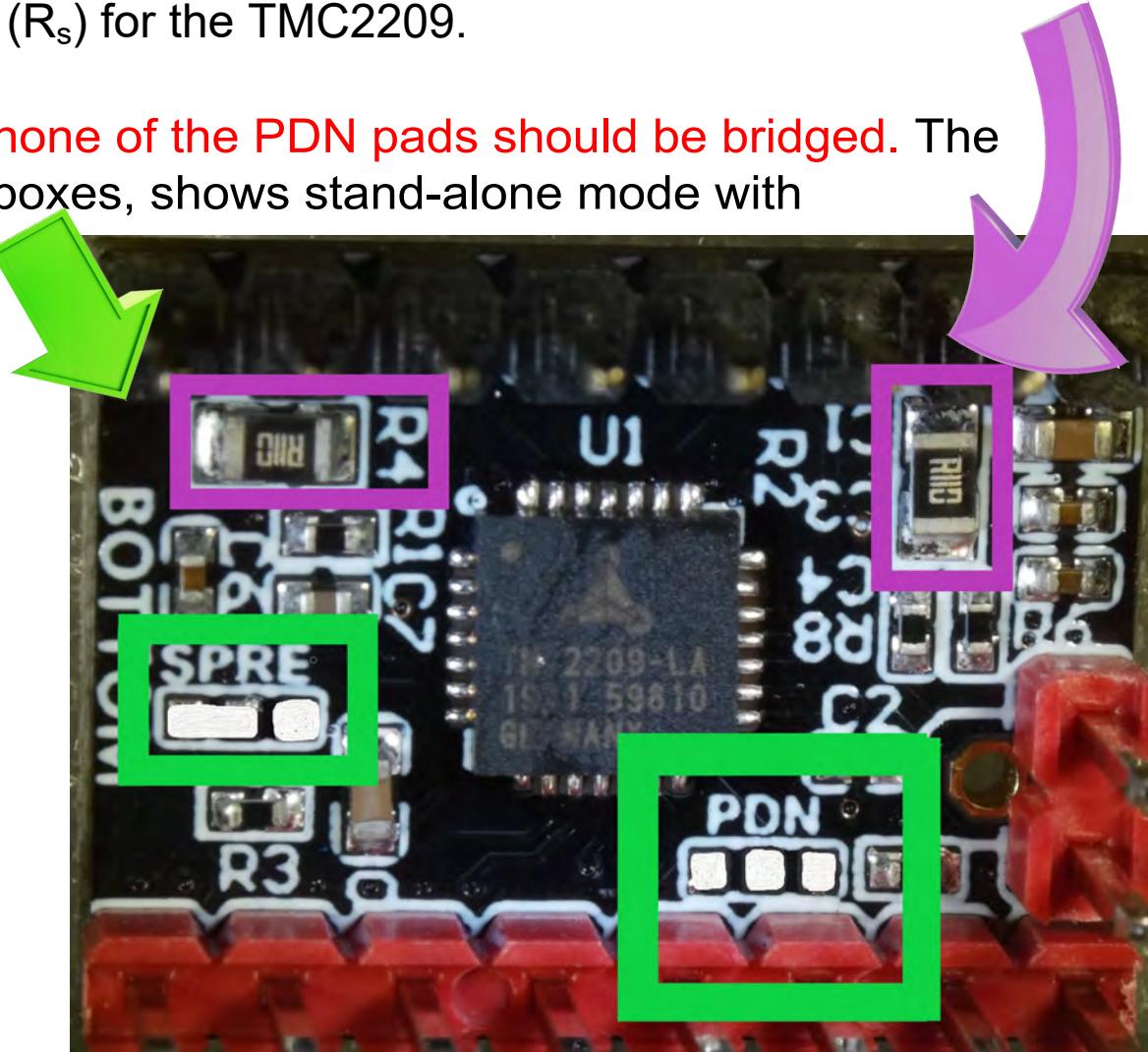
BIQU TMC2209 V1.2**Stand-alone Mode for SpreadCycle**

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 SpreadCycle Mode

Stand-alone Mode for SpreadCycle

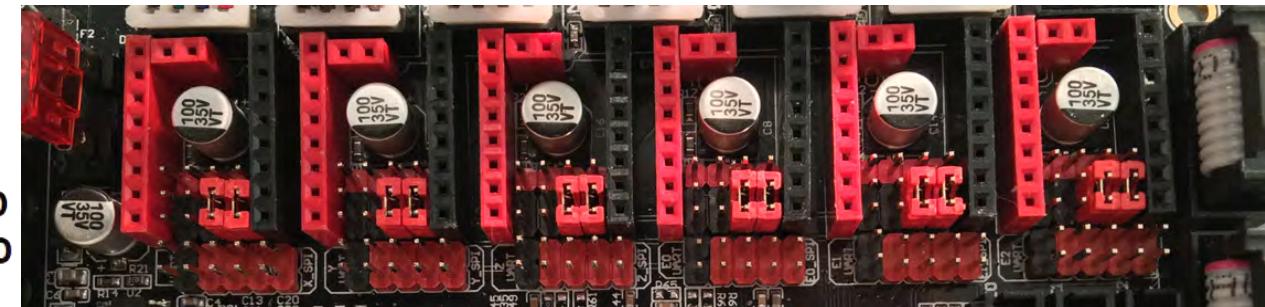
Stand-alone
Mode

1 / 8

Interpolation:
1 / 256

SpreadCycle

EN	-	R	-	VM
MS1	-	DIG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	-	B1
CLK	-		-	B2
STEP	-	8 7	-	VDD
DIR	-	8 7	-	GND



See [Appendix D](#) for legend

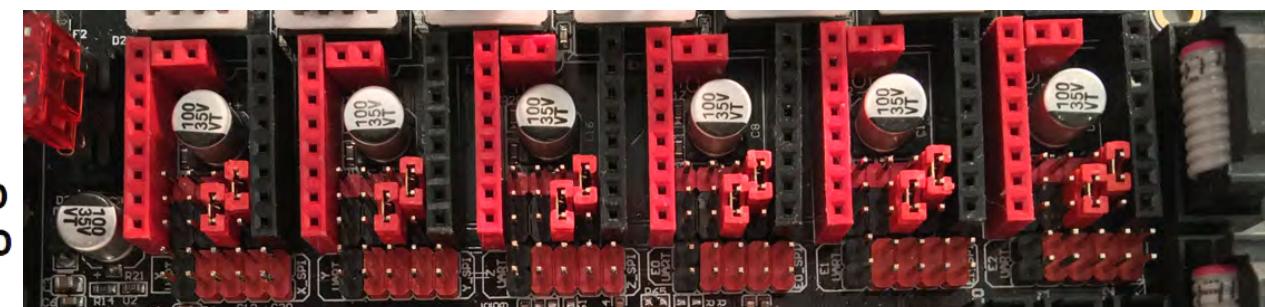
Stand-alone
Mode

1 / 32

Interpolation:
1 / 256

SpreadCycle

EN	-	R	-	VM
MS1	-	DIG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-	MS1	-	B1
CLK	-		-	B2
STEP	-	7	-	VDD
DIR	-	8 7	-	GND



See [Appendix D](#) for legend

Stand-alone SpreadCycle Mode

Stand-alone Mode for SpreadCycle

Stand-alone
Mode

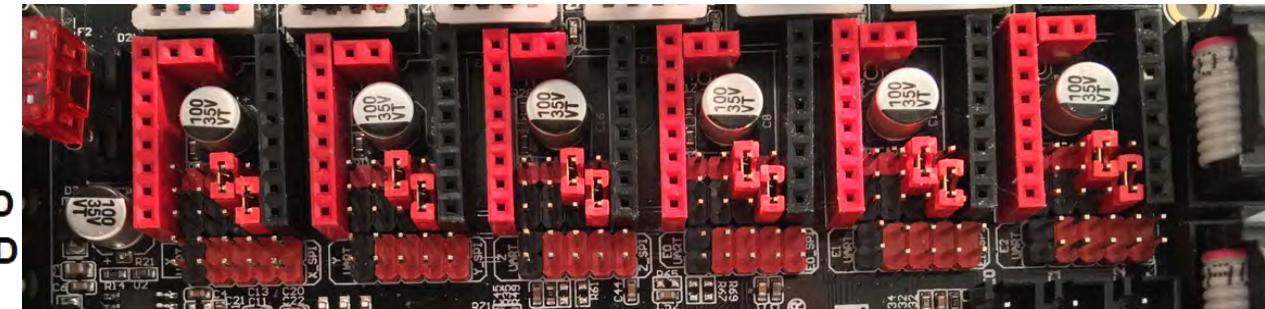
1 / 64

Interpolation:

1 / 256

SpreadCycle

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	8	-	B2
STEP	-	8 7	-	VDD
DIR	-	7	-	GND



See [Appendix D](#) for legend

Stand-alone
Mode

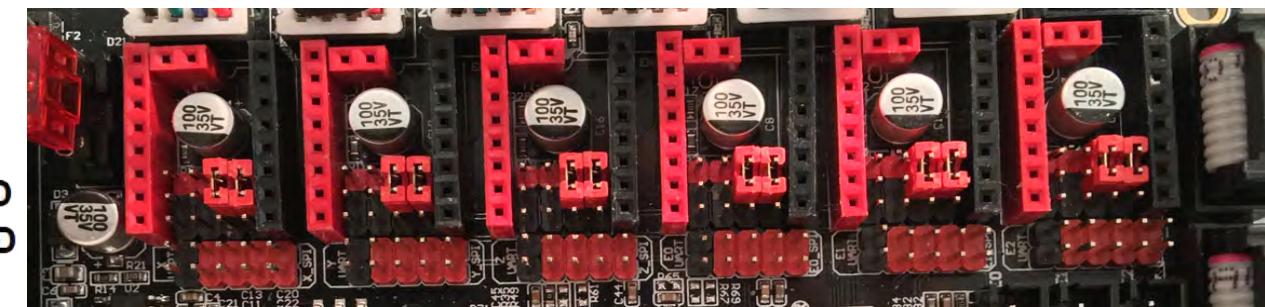
1 / 16

Interpolation:

1 / 256

SpreadCycle

EN	-	R	-	VM
MS1	-	DIAG	-	GND
MS2	-		-	A2
PDN	-		-	A1
PDN	-		-	B1
CLK	-	8 7	-	B2
STEP	-	8 7	-	VDD
DIR	-	-	-	GND

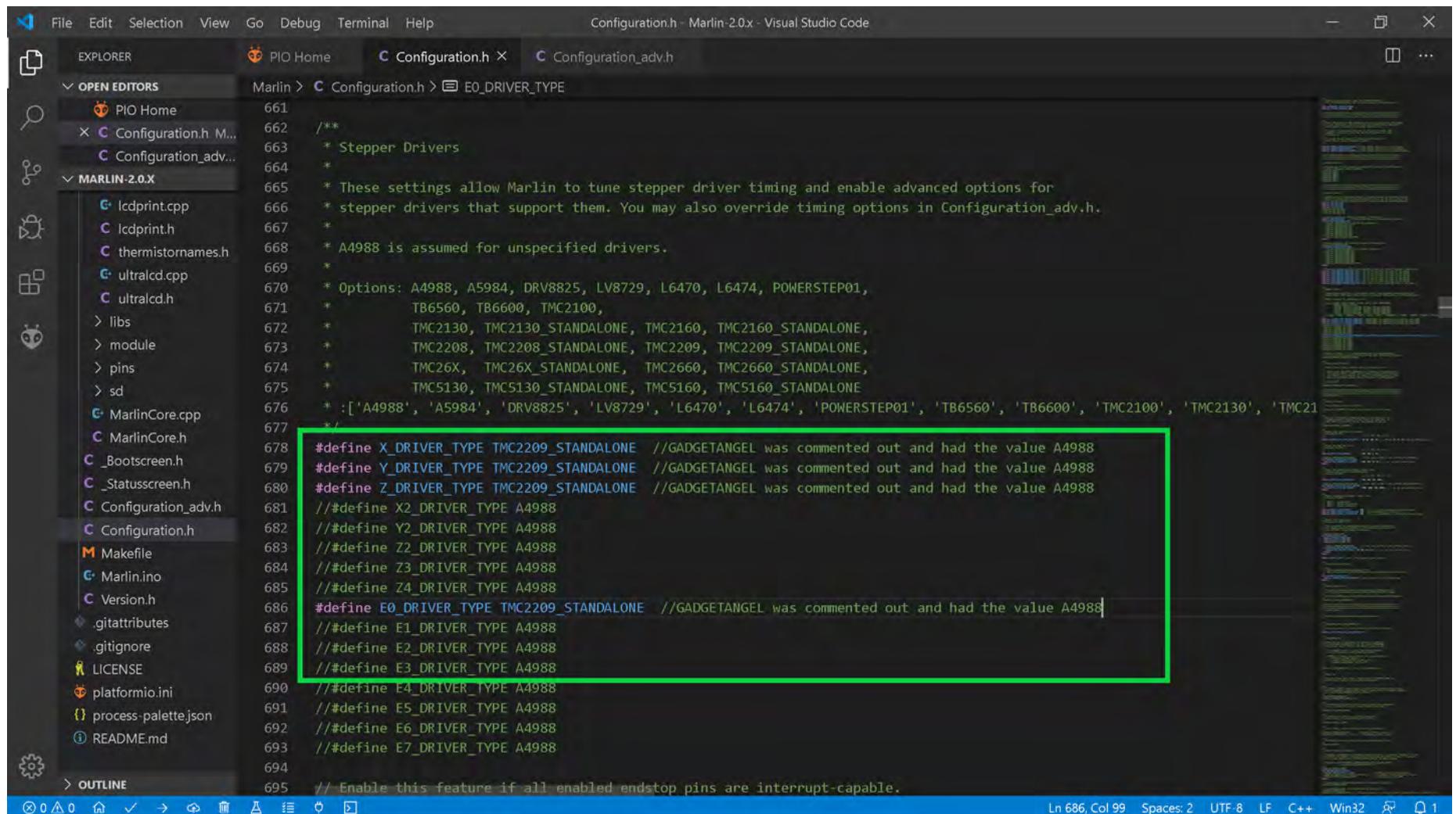


See [Appendix D](#) for legend

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 ("//").



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', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676   */
677
678 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2209_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 TMC2209_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.

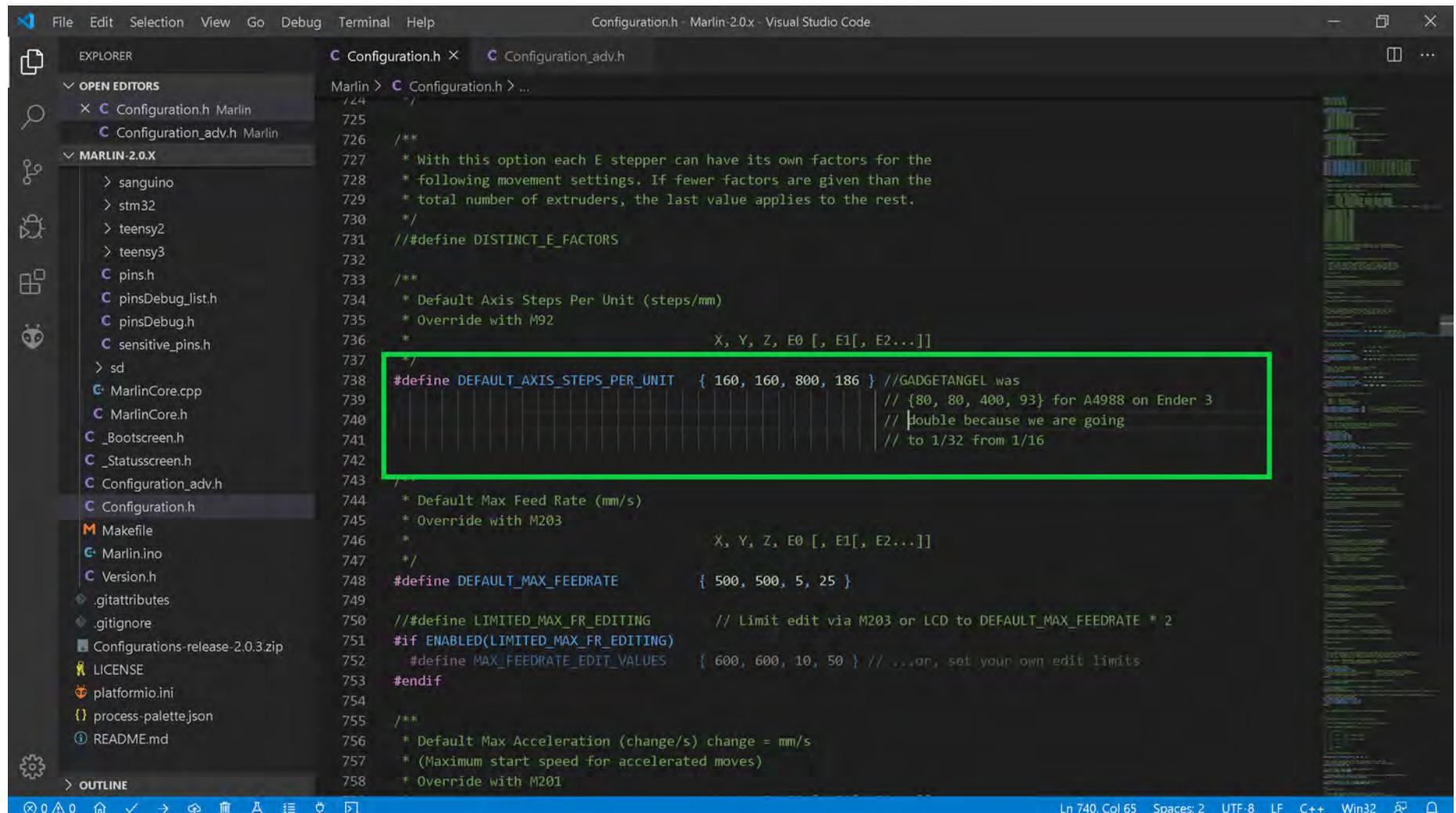
```

A green rectangular box highlights the driver type definitions starting from line 678. Lines 678 through 693 are enclosed in the box, indicating they are the lines being modified.

- 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:

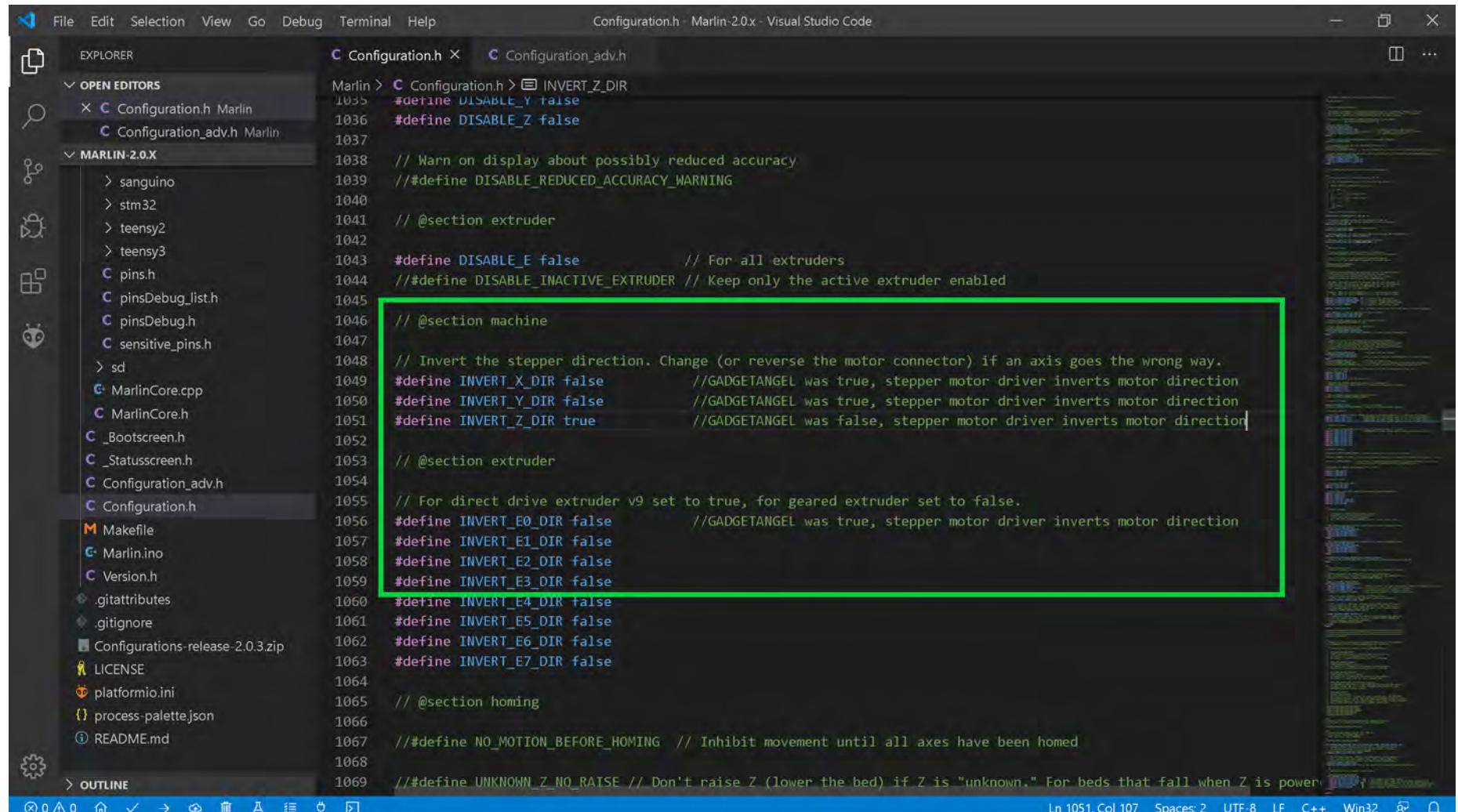
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

The code editor's status bar at the bottom 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 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



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

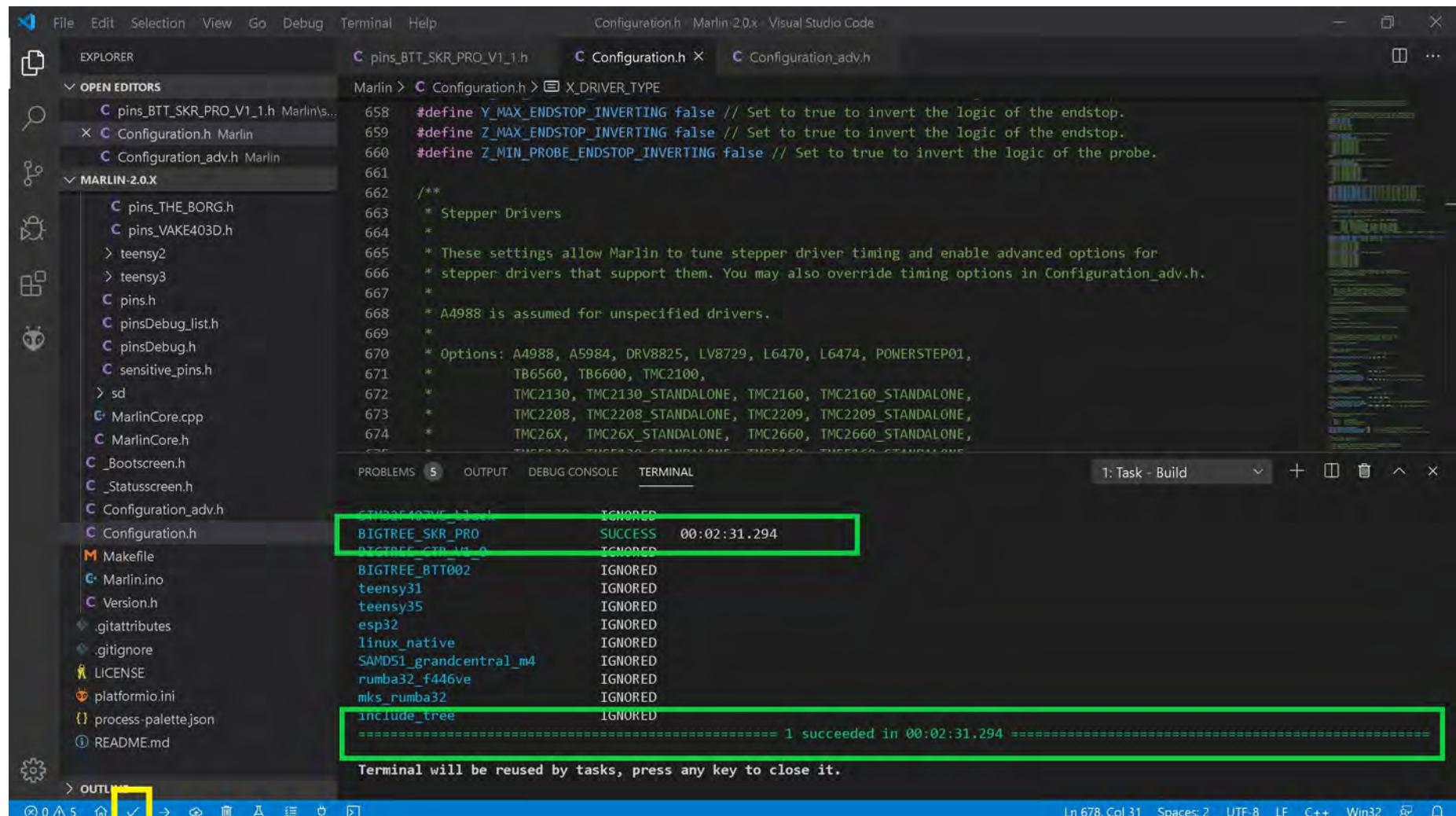
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that this setting should be changed from its current state.

- Go to the next page.

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

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



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

- File Explorer:** Shows files like Configuration.h, pins_BTT_SKR_PRO_V1_1.h, Configuration_adv.h, and MarlinCore.cpp.
- Terminal:** Displays the build log for the "BIGTREE_SKR_PRO" target, which completed successfully in 00:02:31.294. Other targets listed include BIGTREE_BTT002, teensy31, teensy35, esp32, linux_native, SAMD51_grandcentral_m4, rumba32_f446ve, mks_rumba32, and include_tree.
- Status Bar:** Shows the current line (Ln 678), column (Col 31), spaces (Spaces: 2), encoding (UTF-8), line feed (LF), C++, Win32, and other system information.

A yellow box highlights the checkmark icon in the bottom-left corner of the terminal bar, indicating the build task was successful. A green box highlights the terminal output showing the build results.

```

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

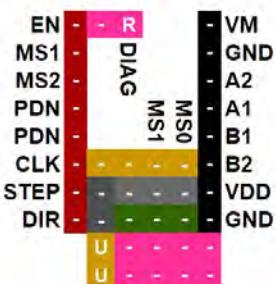
```

- 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 [Appendix E](#).

BIQU TMC2209 V1.2UART Mode**R** - Stall Detection

UART Mode

base of card



fans/heaters/usb side

See [Appendix D](#) for legend
STATUS:
 Not Yet
 Tested

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

BIQU®
TMC2209

UART Mode

Maximum 256
Subdivision28V DC
2.8A (peak)

**Steps are set inside
of your Firmware**

**Driving Current
Calculation
Formula**

 R_S (Typical Sense Resistor) = 0.11Ω

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

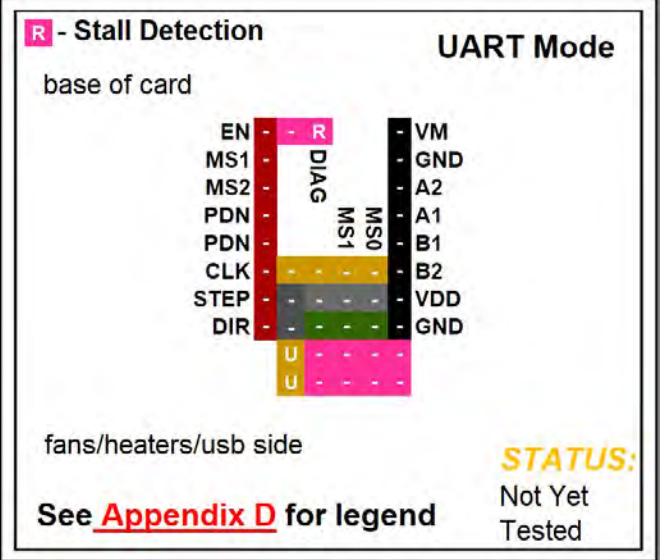
See Appendix B #6. Use 50% to 90%
as shown below:

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

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

See Appendix B #6. Use 50% to 90%
as shown below:

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

BIQU TMC2209 V1.2**UART Mode**

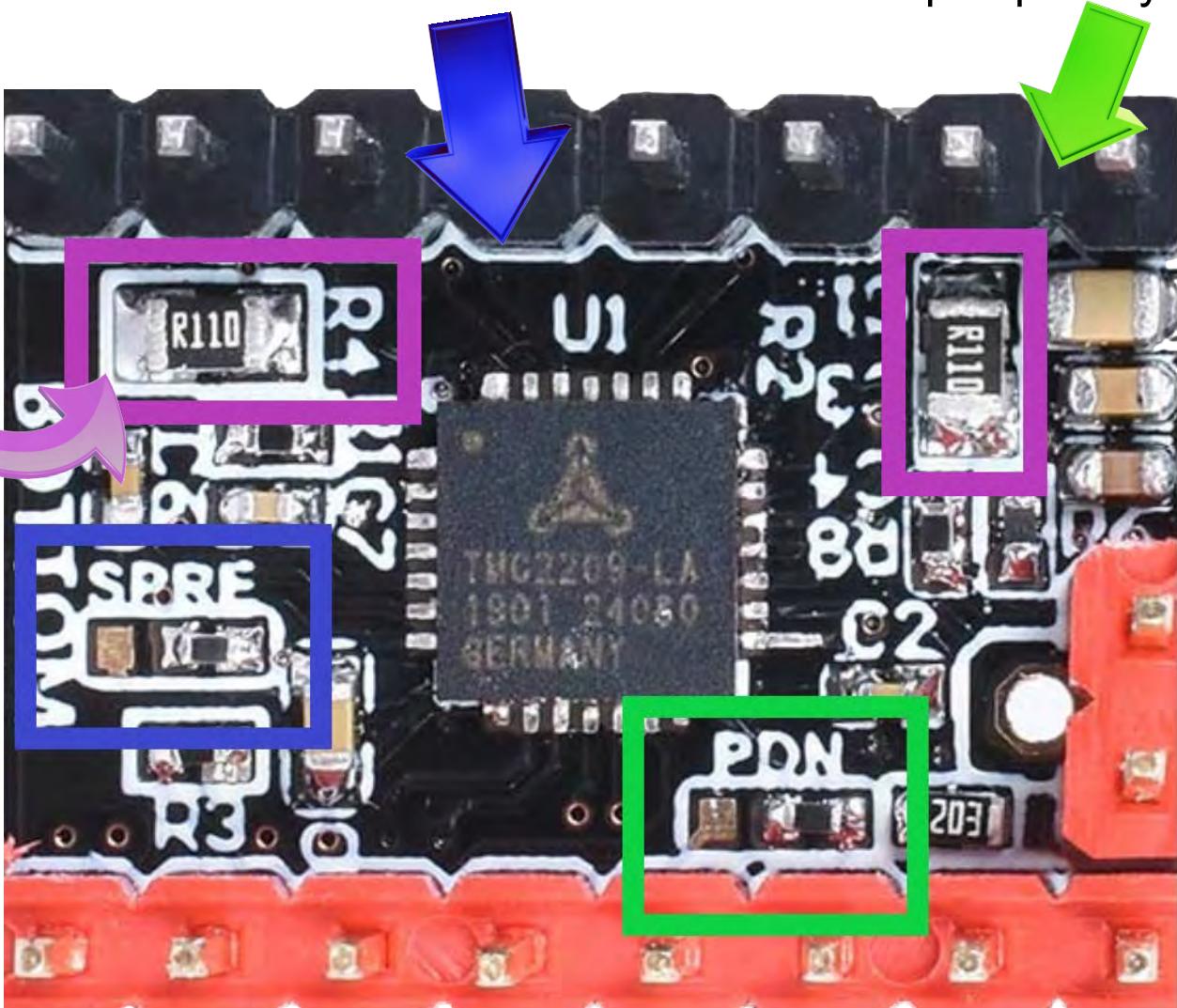
MOST Biqu TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.

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

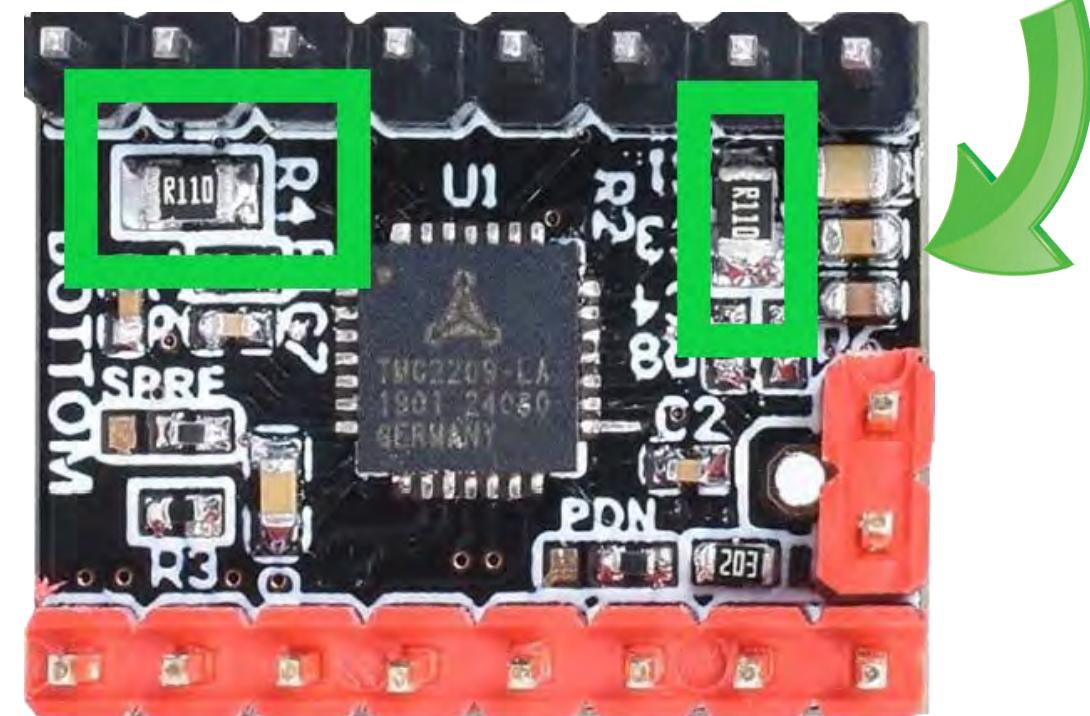
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

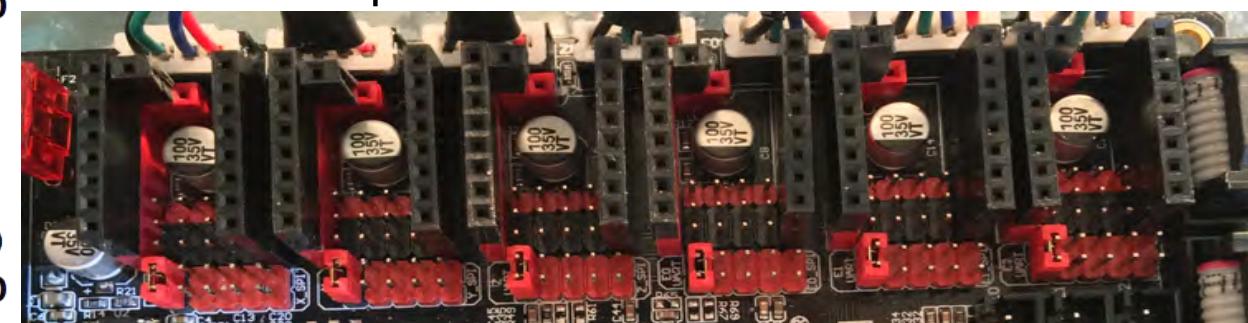
$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



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



See [Appendix D](#) for legend

UART

EN	-	R	-	VM
MS1	-	DIA	GND	
MS2	-		A2	
PDN	-		A1	
PDN	-		B1	
CLK	-		B2	
STEP	-		VDD	
DIR	-		GND	
	U			
	U			

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 code editor has a green box highlighting the driver type definitions for all axes (X, Y, Z, E0-E7). The status bar at the bottom right 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

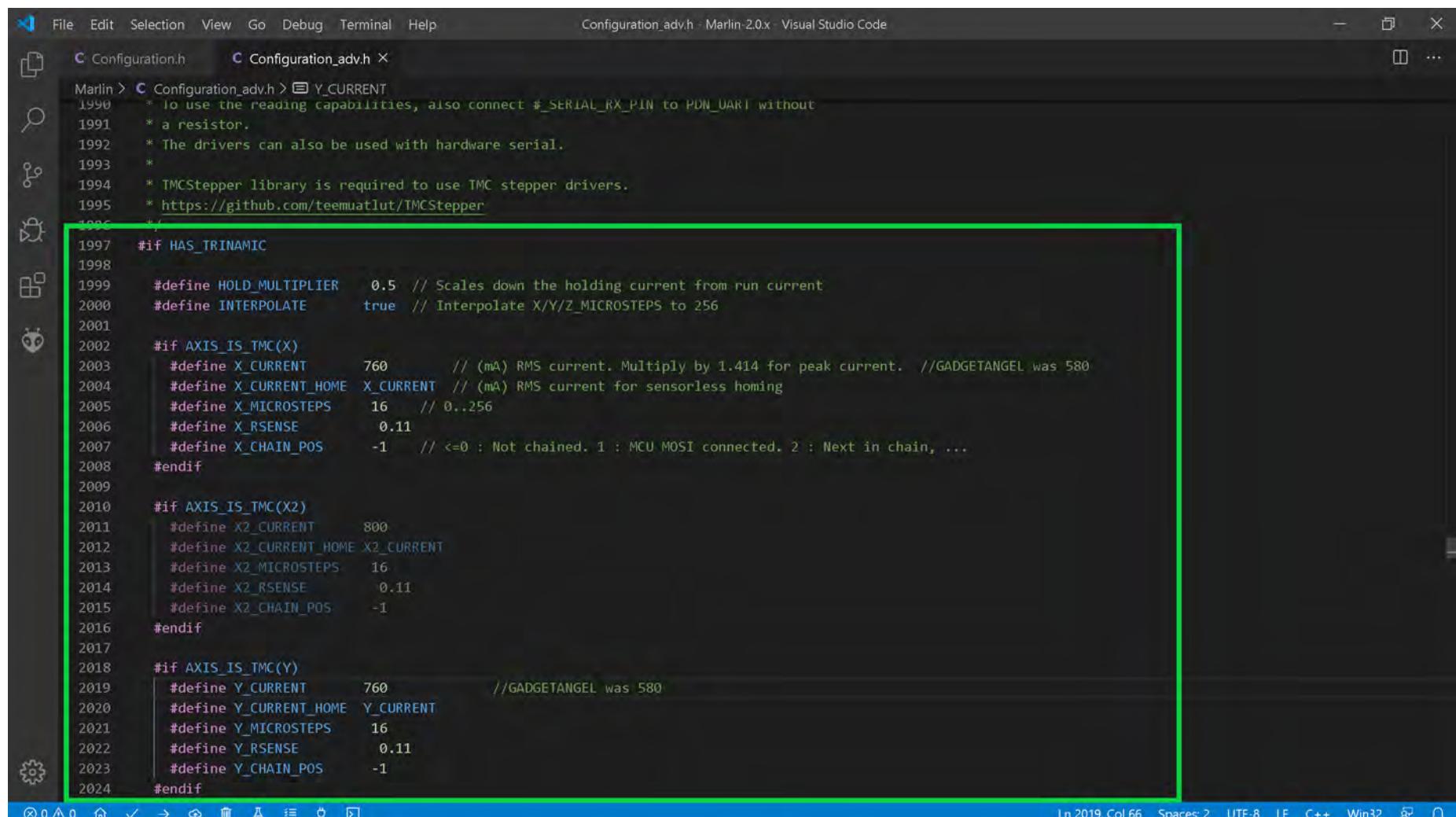
```

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".



```

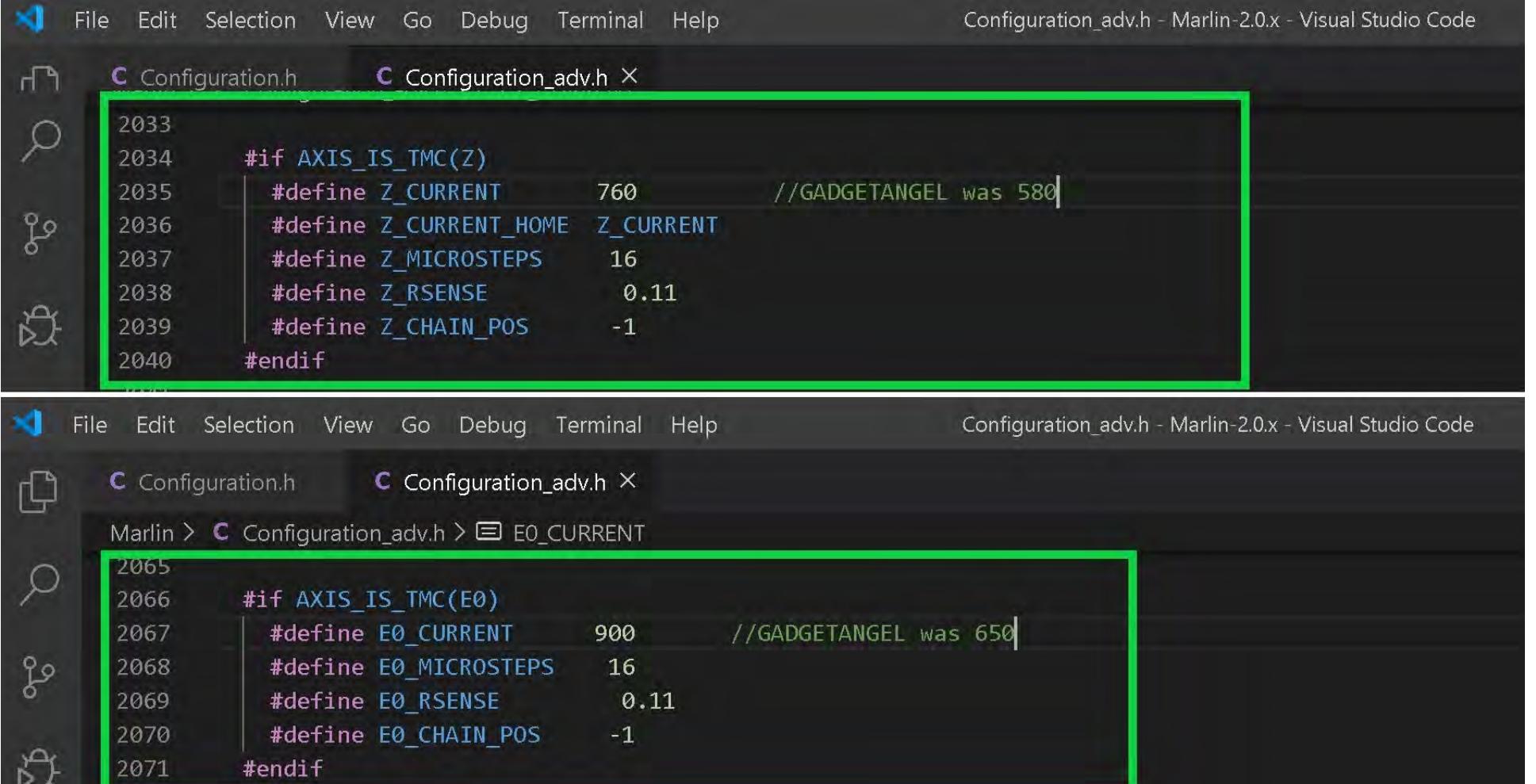
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers 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

```

- 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 ×
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

```



```

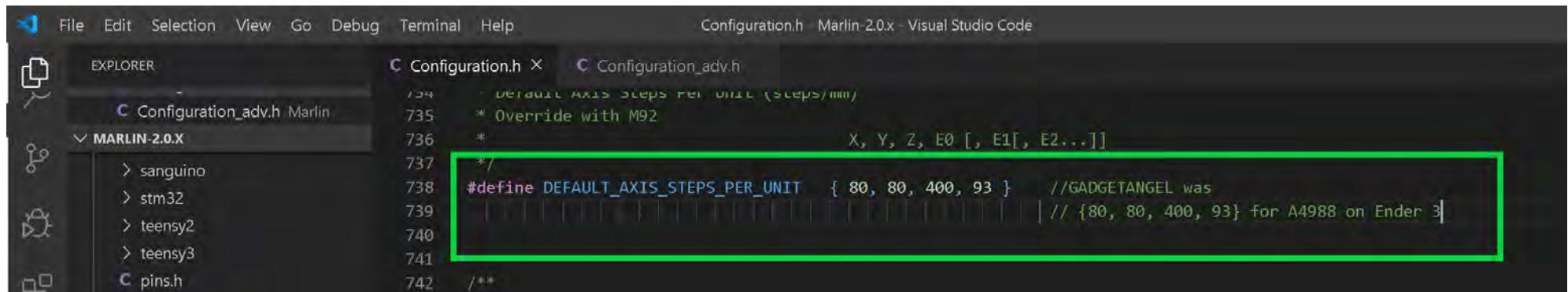
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h ×
Marlin > Configuration_adv.h > E0_CURRENT
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900          //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   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

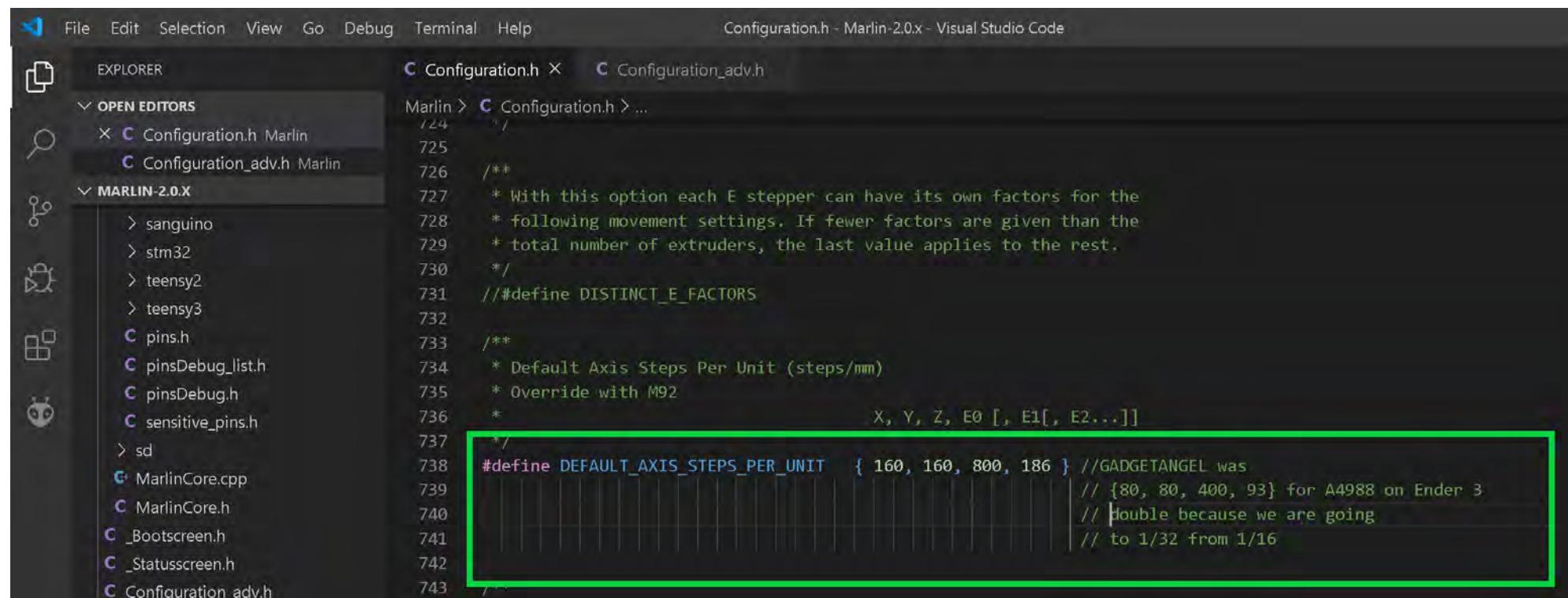
MARLIN-2.0.X

```

 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

```

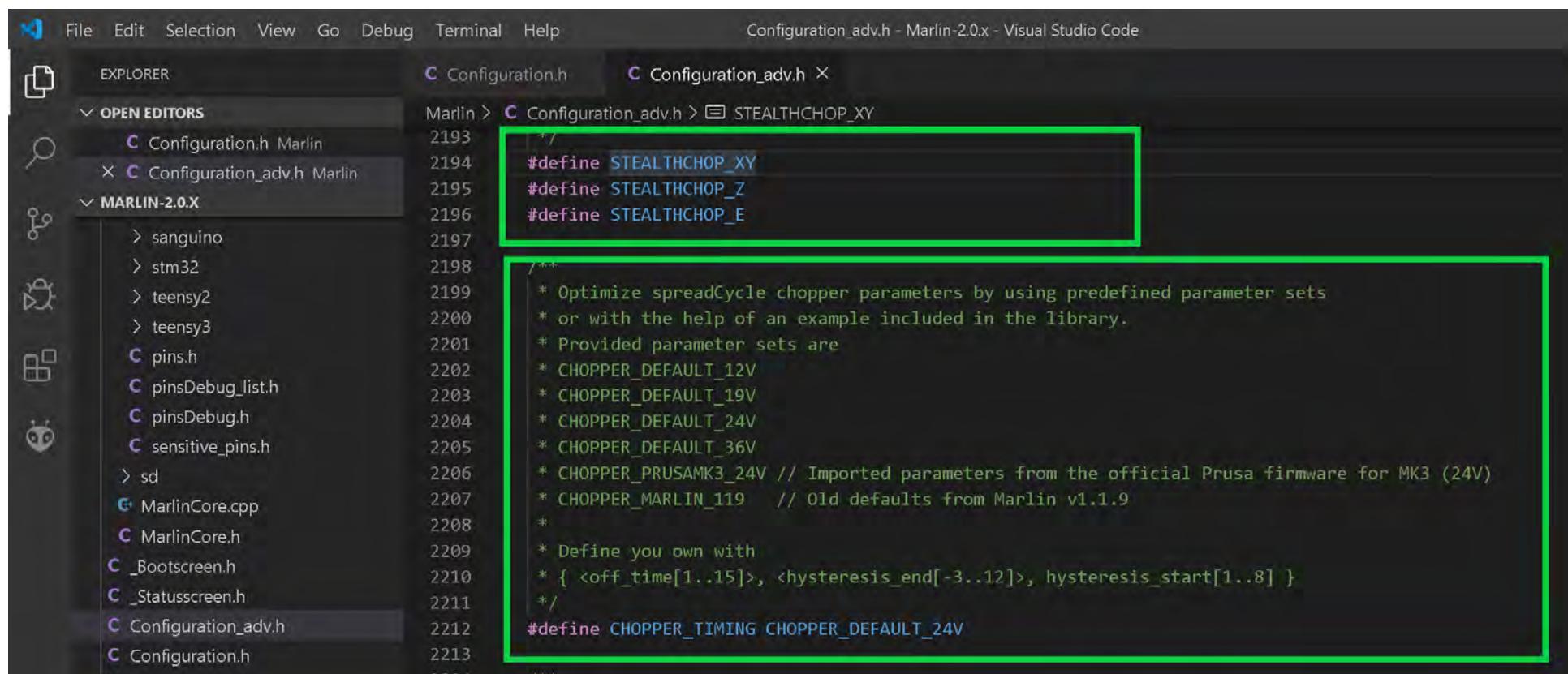
 724 */
 725
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 //#define DISTINCT_E_FACTORS
 732
 733 /**
 734 * Default 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
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
  G MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C _Statusscreen.h
  C Configuration_adv.h
  C Configuration.h

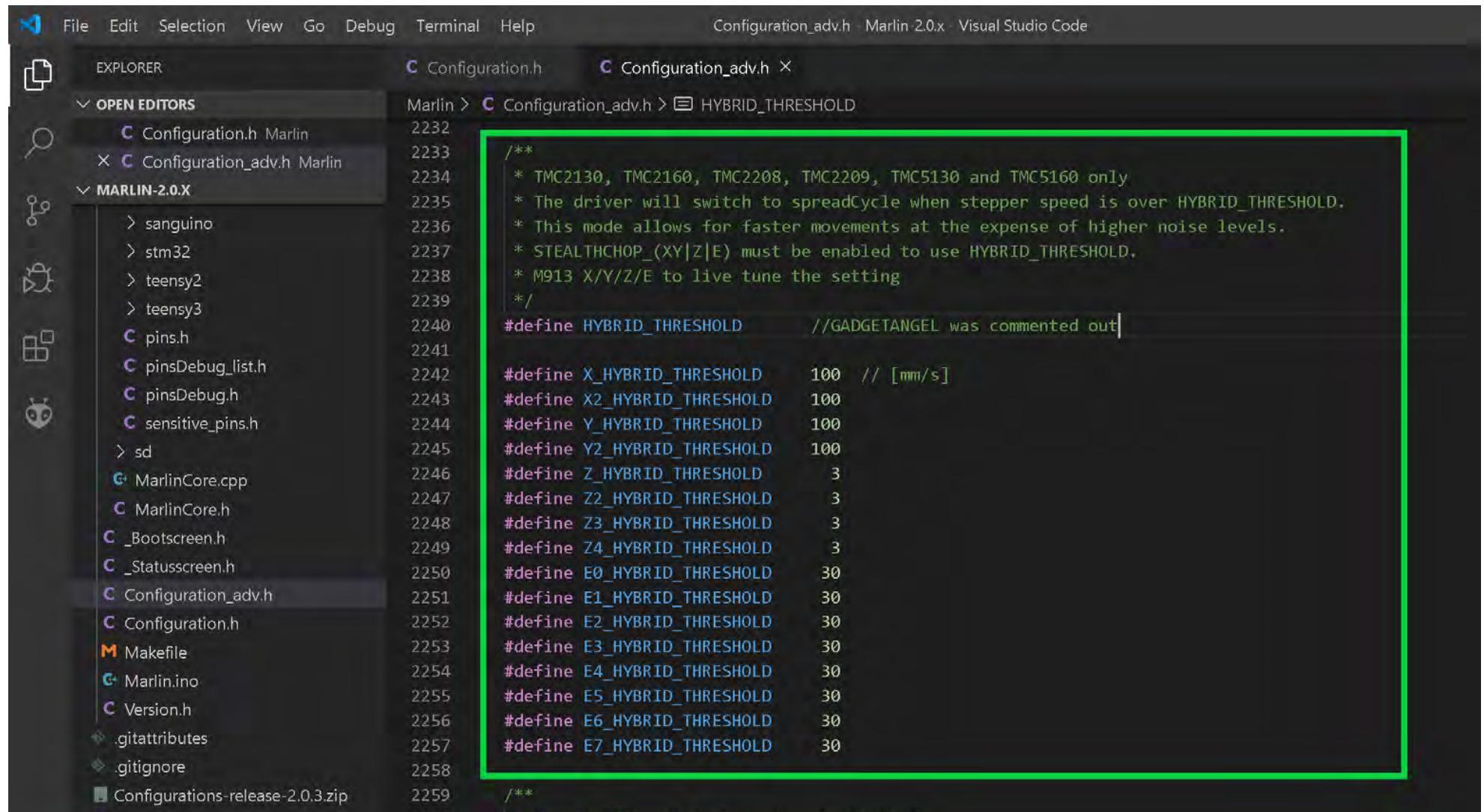
Marlin > C Configuration_adv.h > STEALTHCHOP_XY
2193 /**
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * 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
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 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'.
- Editor:** Displays the content of Configuration_adv.h. 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

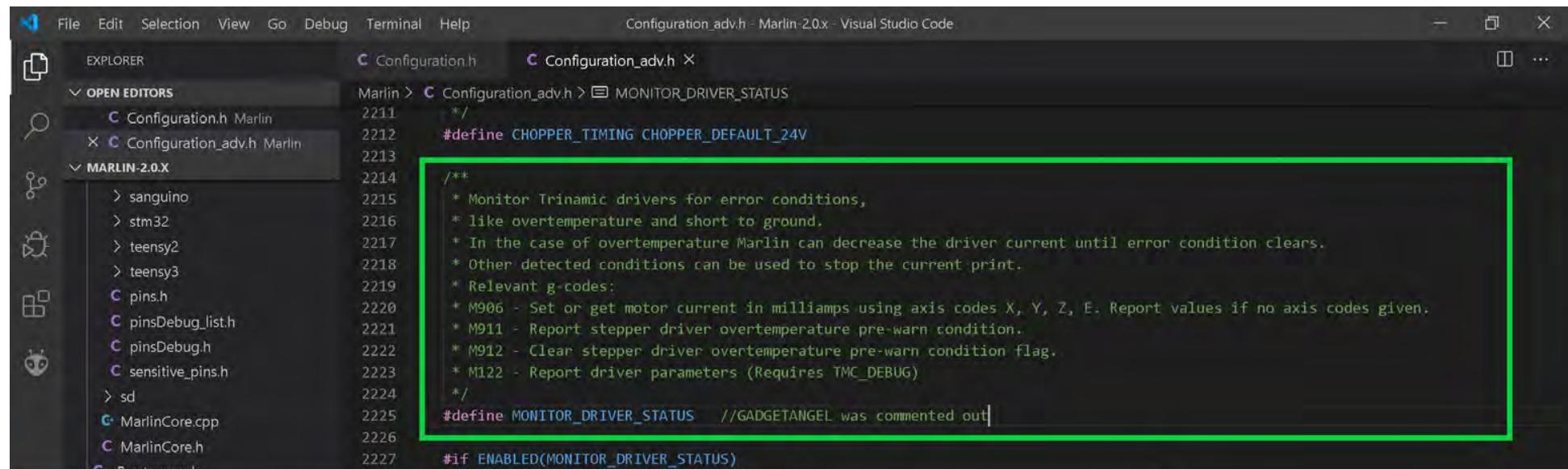
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD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



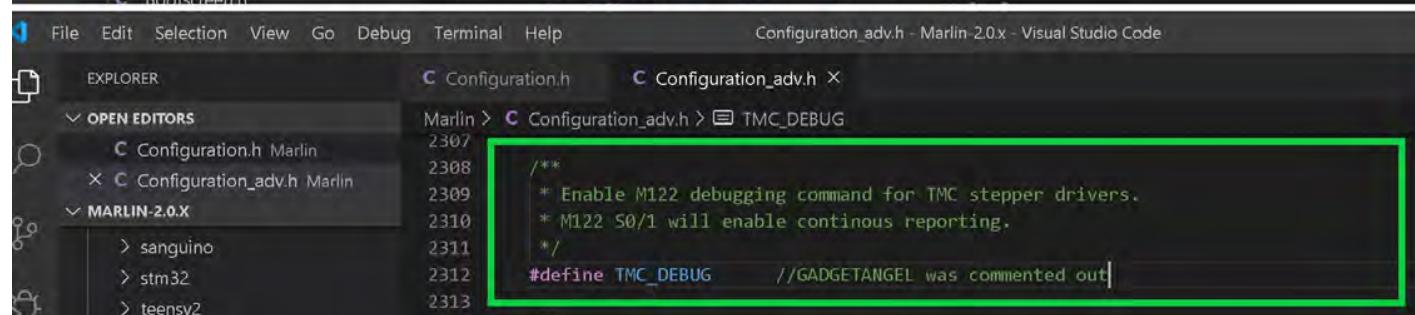
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

C Configuration.h C Configuration_adv.h X
Marlin > C 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 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

```

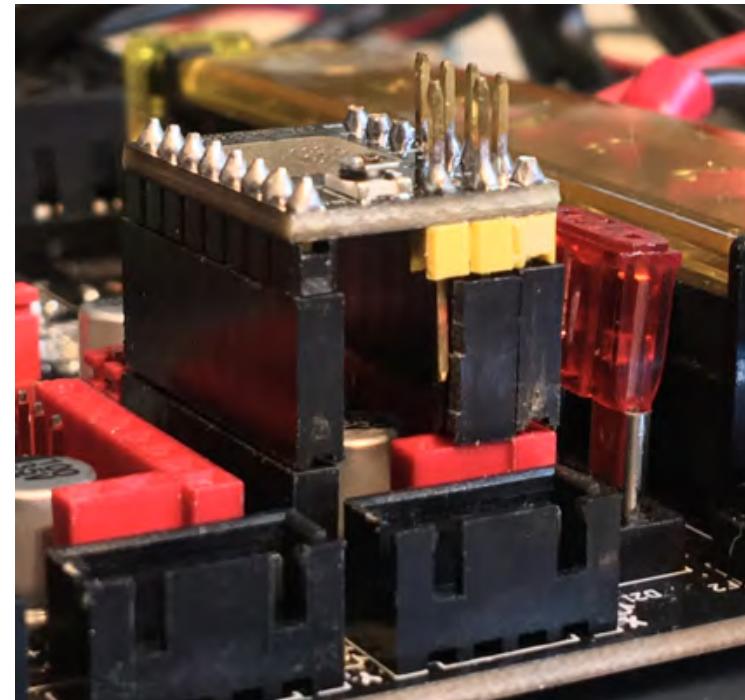
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU 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.



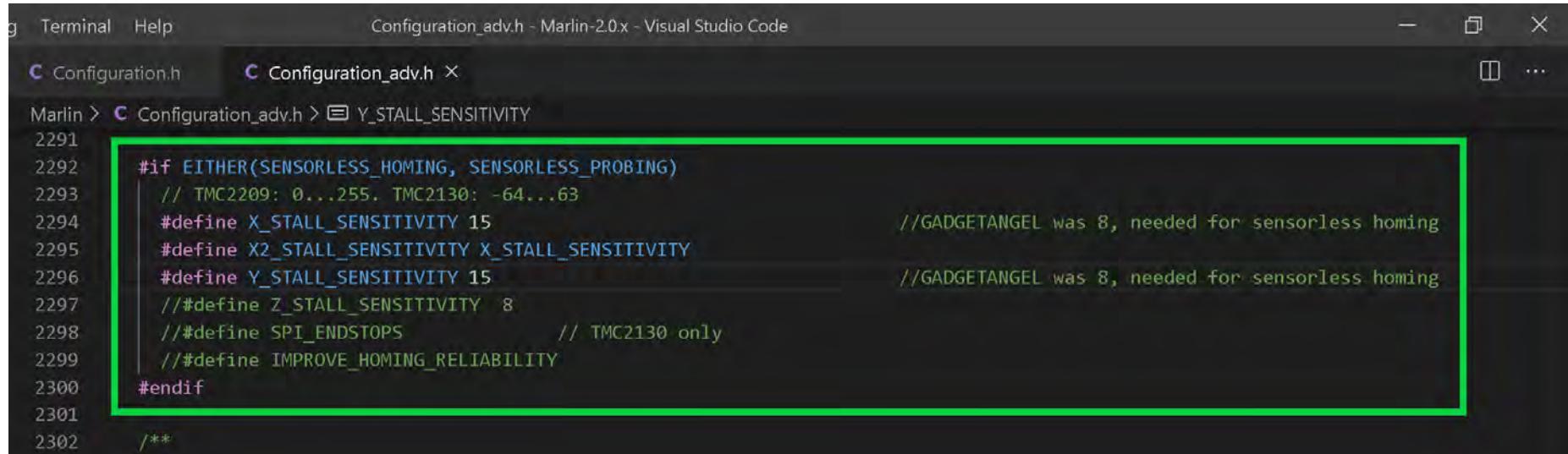
- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

```
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
```

- 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".

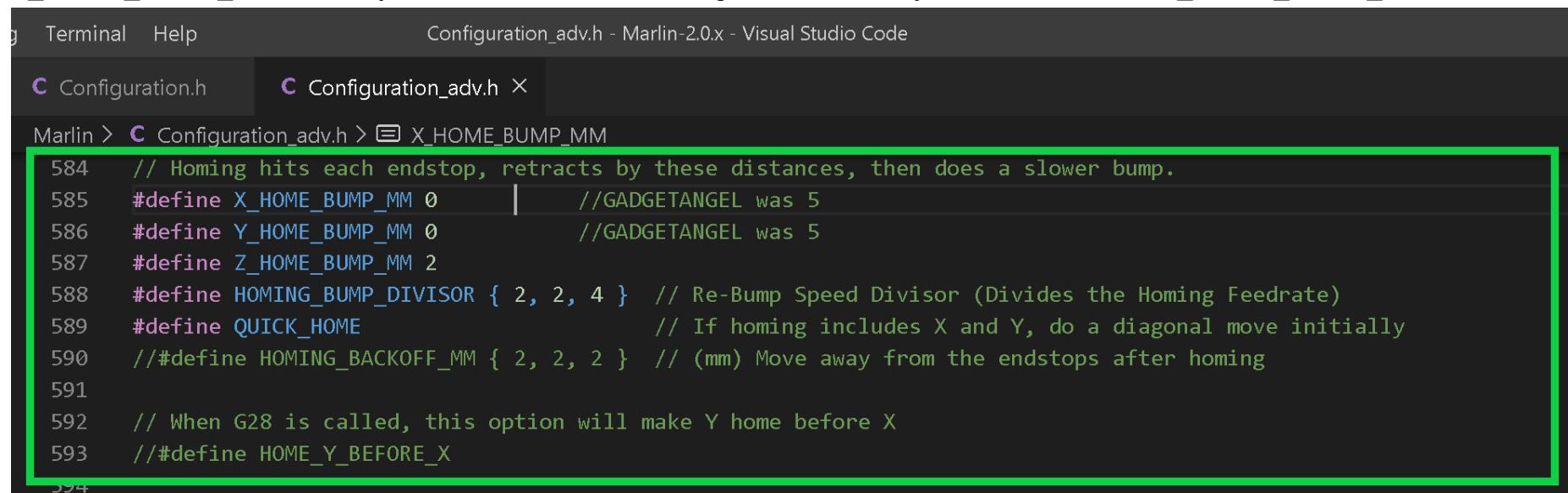


```

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.



```

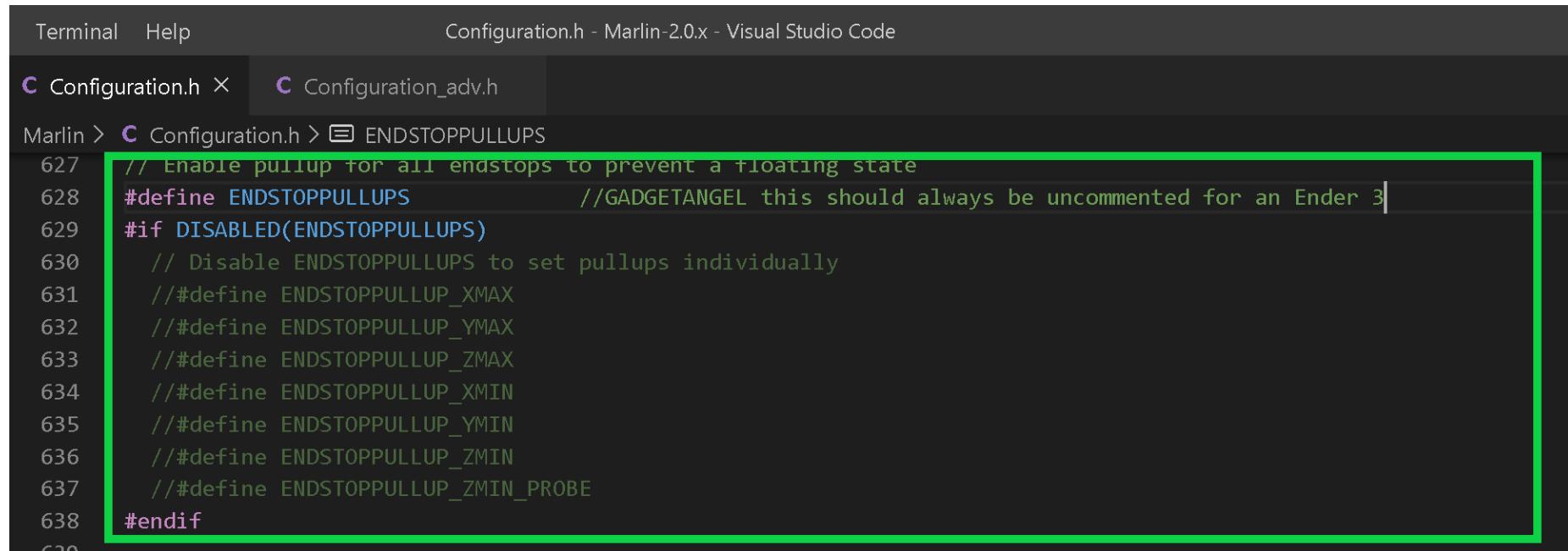
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

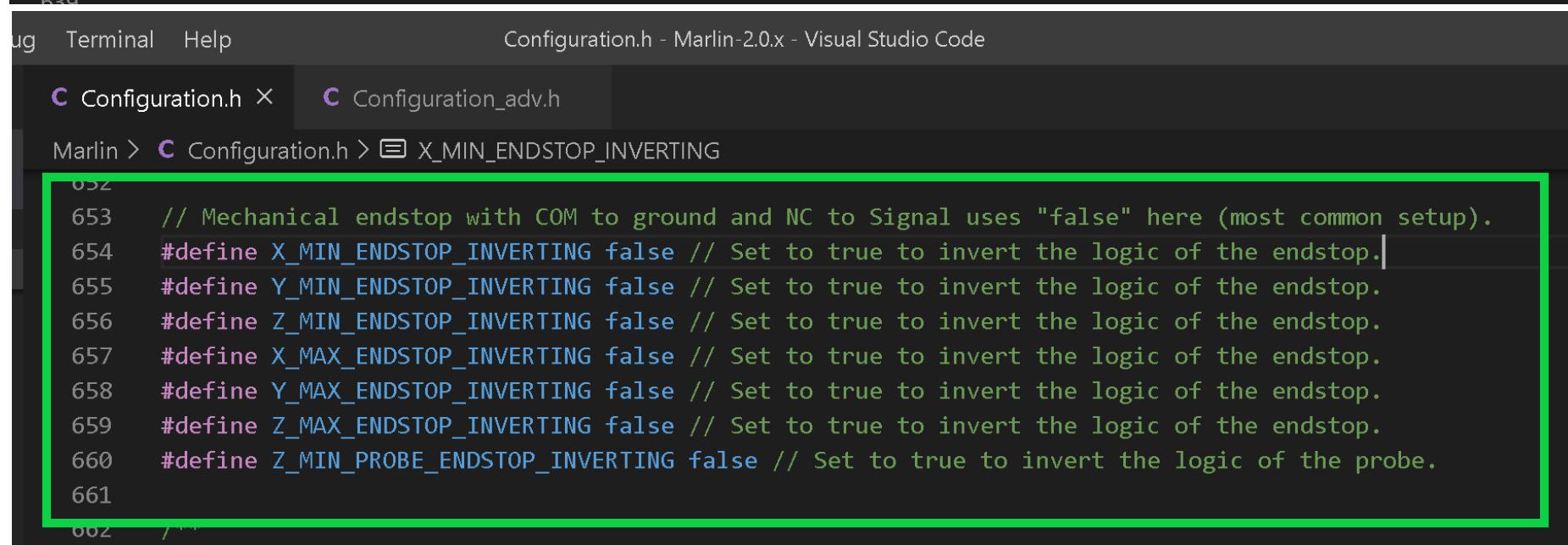
C Configuration.h X C Configuration_adv.h

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

```

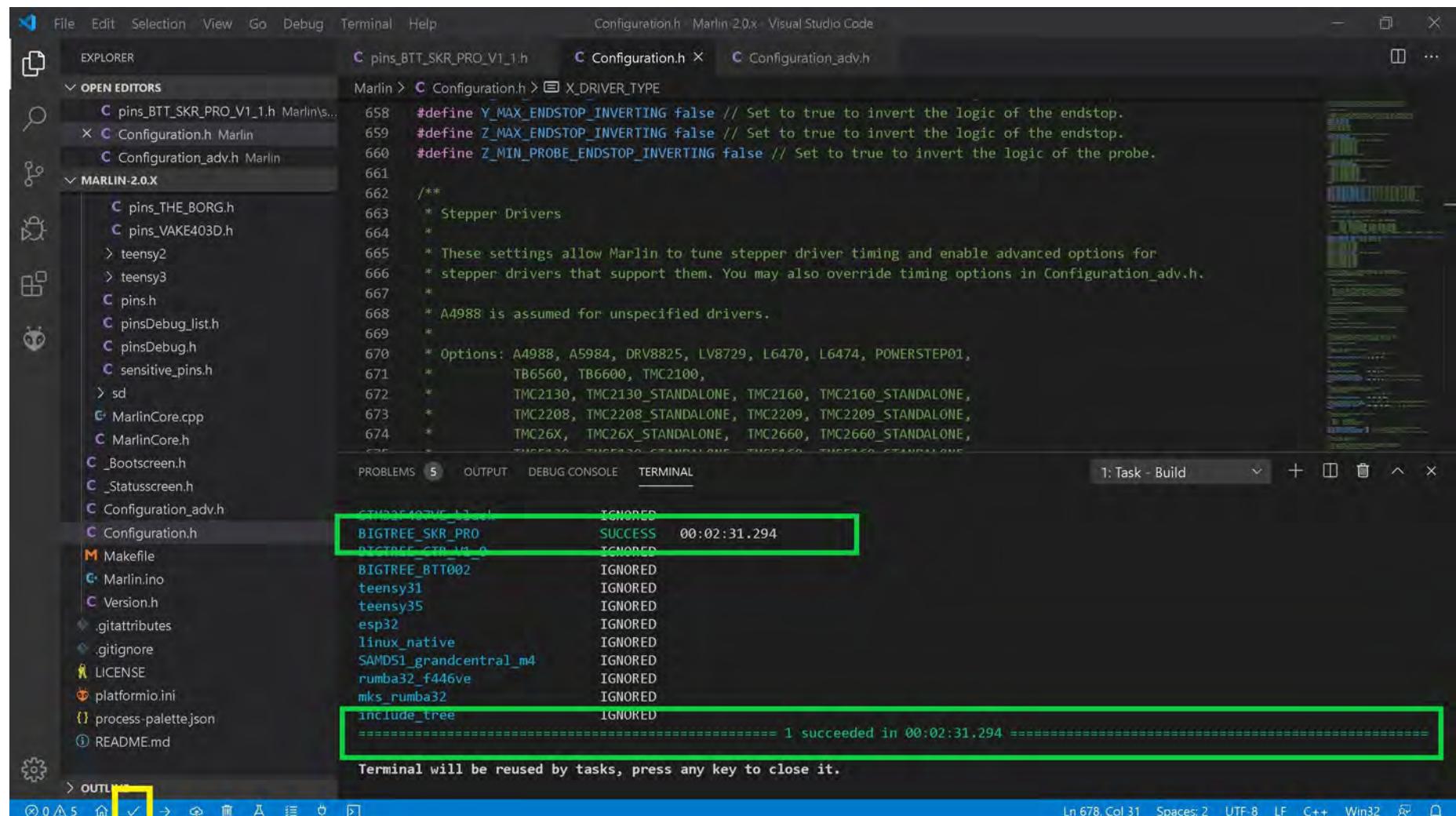
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
655 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
656 #define 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.



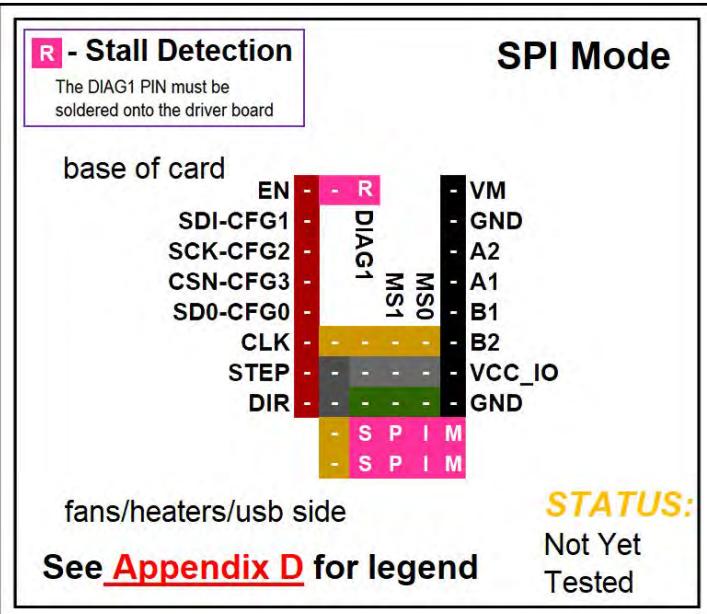
The screenshot shows the Visual Studio Code interface with the following details:

- 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 "OUTLINE" tab at the bottom left. A green box highlights the terminal output area, which shows the compilation results for various boards:

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

 The terminal also displays the message "1 succeeded in 00:02:31.294".

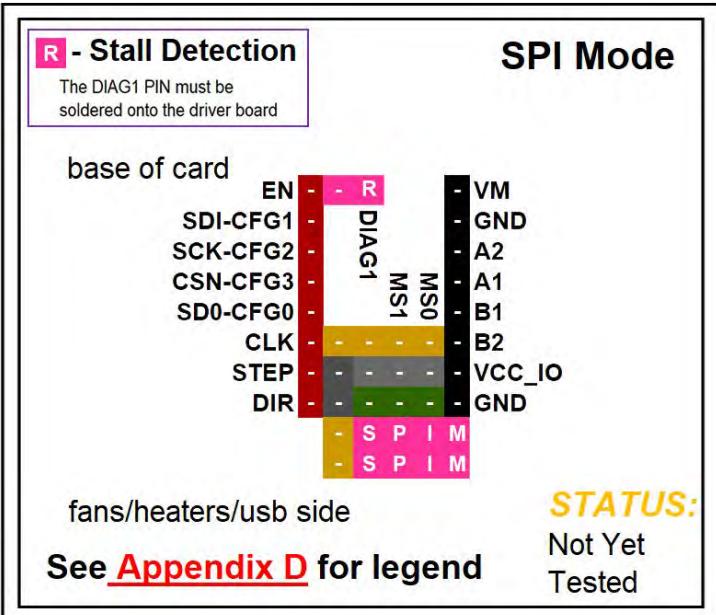
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

**BIQU TMC5160 V1.2**SPI Mode

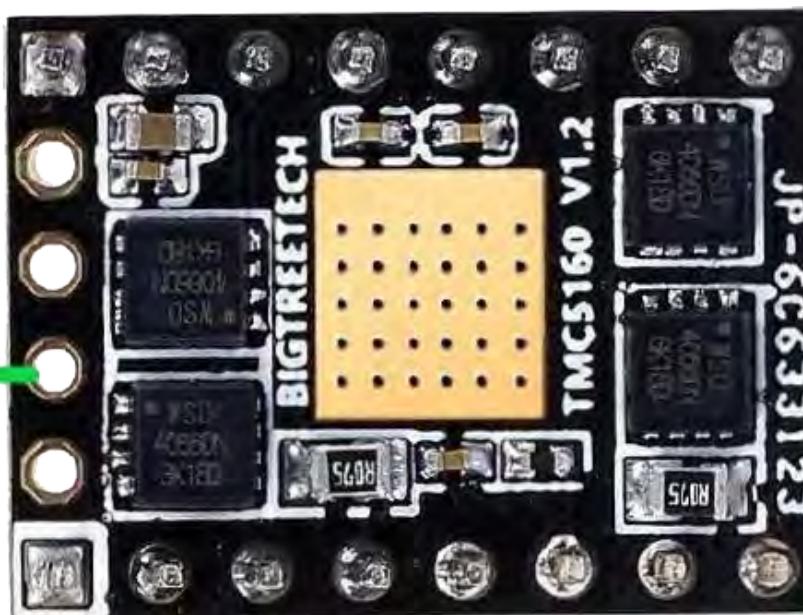
Note: You can use 50% to 90% of the calculated I_{RMS} ($I_{MAX}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

Driver Chip  TMC5160 SPI Mode Maximum 256 Subdivision 40V DC 4.3A (peak)	Steps are set inside of your Firmware
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: $I_{MAX} = I_{MAX} * 0.90 = 3.900$

**BIQU TMC5160 V1.2**SPI 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

BIQU TMC5160 V1.2

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

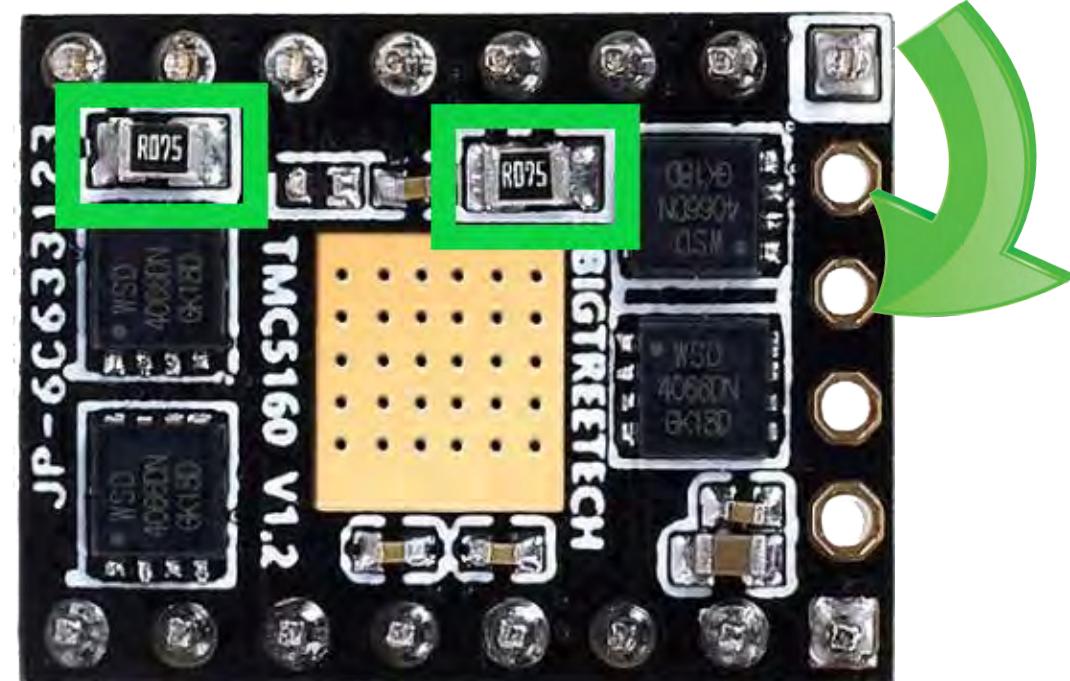
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

$R_s = R150$ is 0.15 Ohms

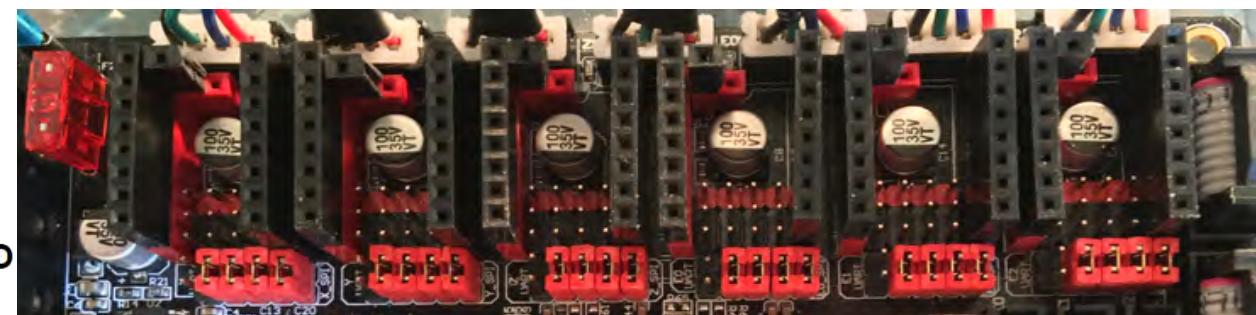
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



EN	-	R	-	VM
SDI-CFG1	-	DIAG1	-	GND
SCK-CFG2	-		-	A2
CSN-CFG3	-	MS1	-	A1
SD0-CFG0	-	MS0	-	B1
CLK	-	-	-	B2
STEP	-	-	-	VCC_IO
DIR	-	-	-	GND
	-	S P I M	-	
	-	S P I M	-	

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



See [Appendix D](#) for legend

SPI

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".

```

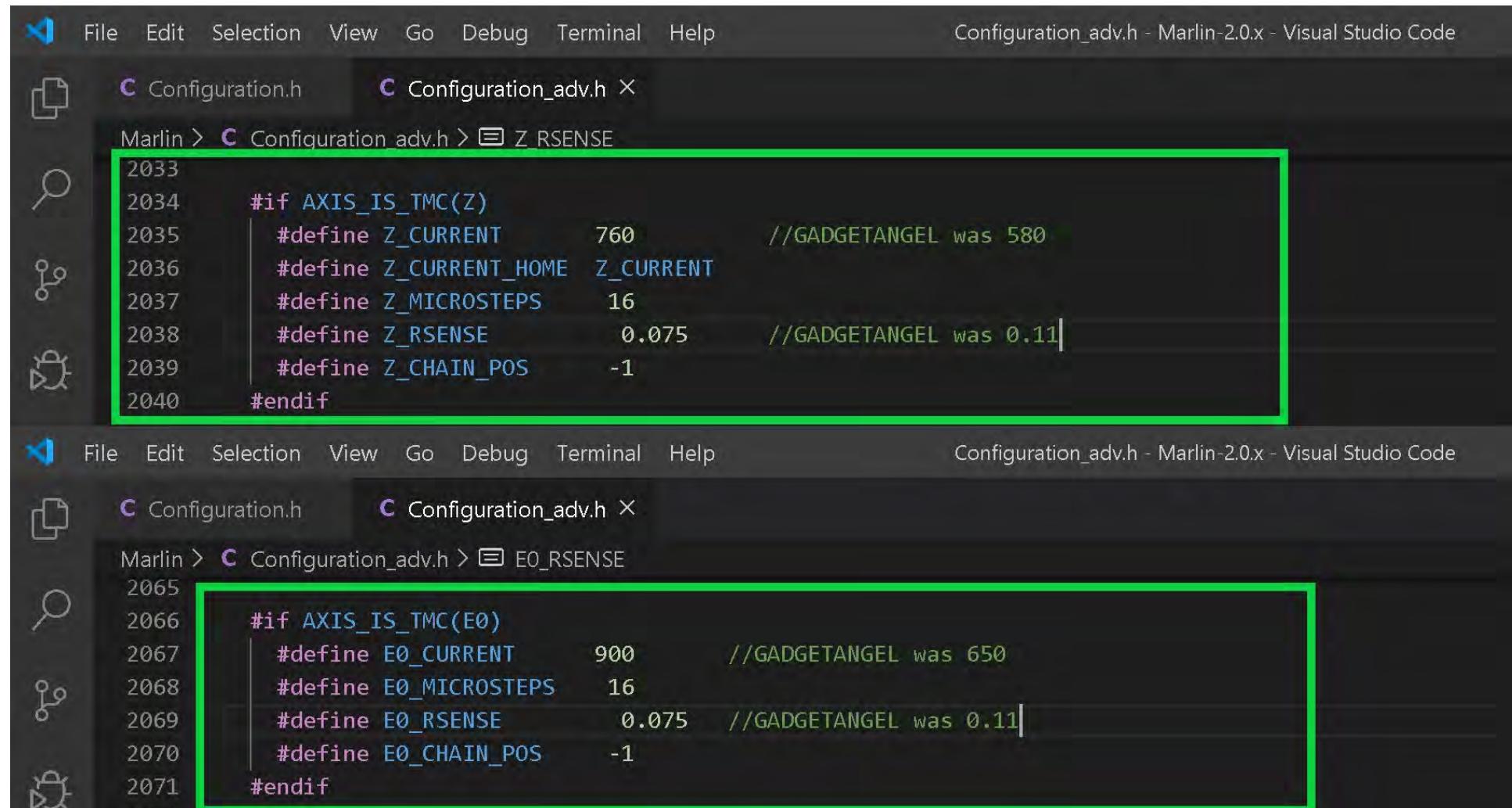
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > X.RSENSE
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #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
2032 #endif

```

- 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".



```

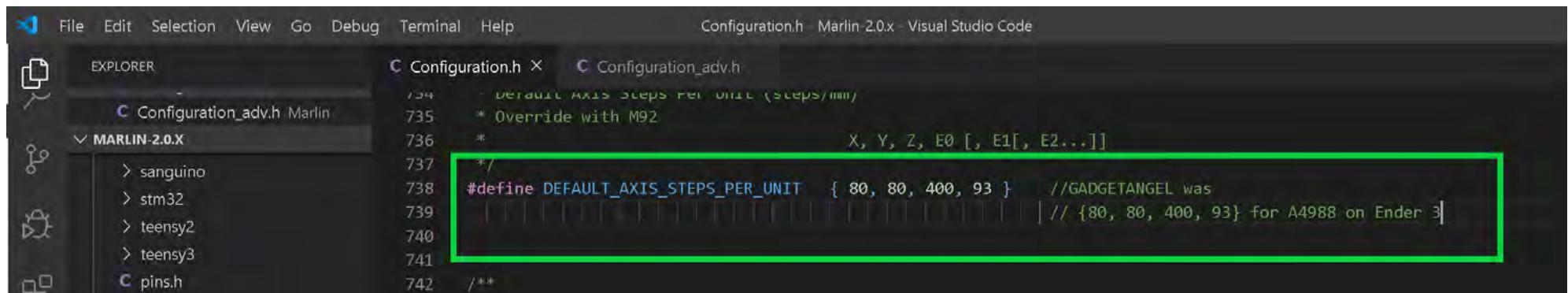
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760           //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME 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

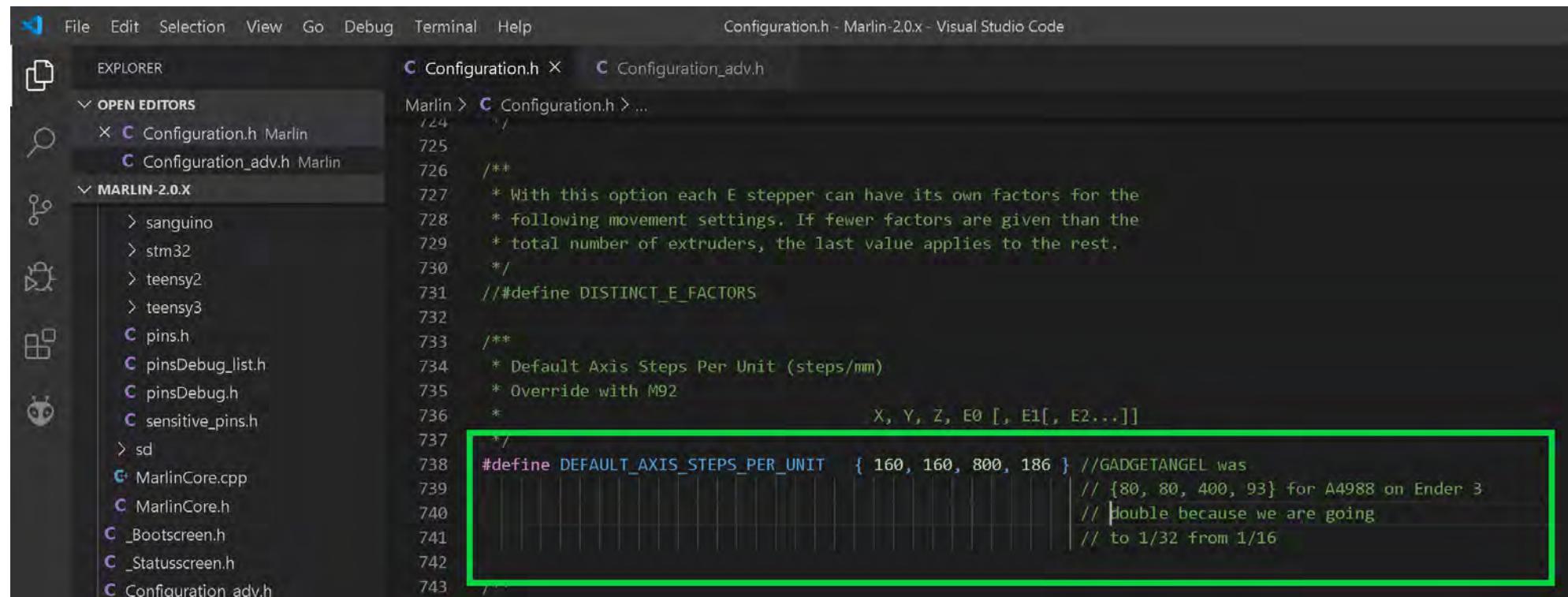
MARLIN-2.0.X

```

 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

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 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

```

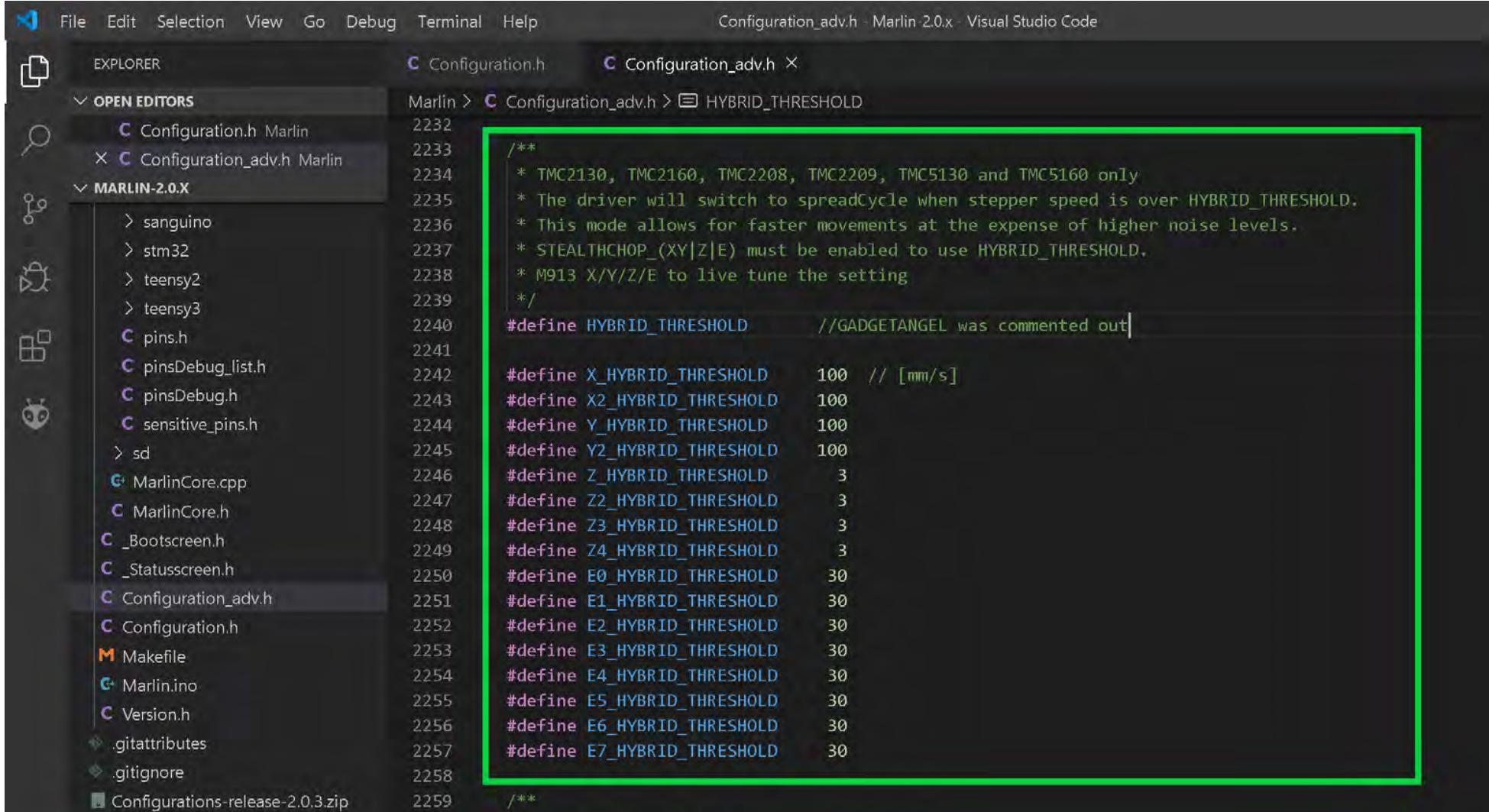
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
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.



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 MARLIN-2.0.X, including files like Configuration.h, Configuration_adv.h, pins.h, and MarlinCore.cpp.
- Editor:** The Configuration_adv.h file is open, specifically at line 2232. A green box highlights the HYBRID_THRESHOLD section. The code in this section is as follows:

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

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

```

File Edit Selection View Go Debug Terminal Help Configuration.h Configuration_adv.h X
EXPLORER Marlin > C Configuration_adv.h > MONITOR_DRIVER_STATUS
OPEN EDITORS 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)

```

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

```

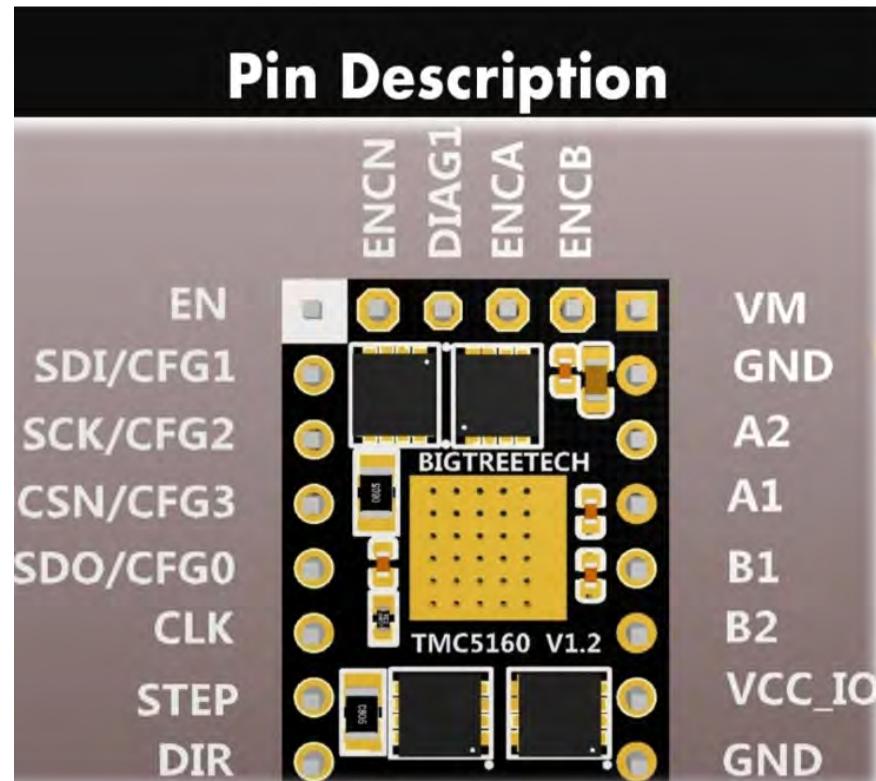
File Edit Selection View Go Debug Terminal Help Configuration.h Configuration_adv.h X
EXPLORER Marlin > C Configuration_adv.h > TMC_DEBUG
OPEN EDITORS 2307
 2308 /**
 2309 * Enable M122 debugging command for TMC stepper drivers.
 2310 * M122 S0/1 will enable continuous reporting.
 2311 */
 2312 #define TMC_DEBUG //GADGETANGEL was commented out
 2313

```

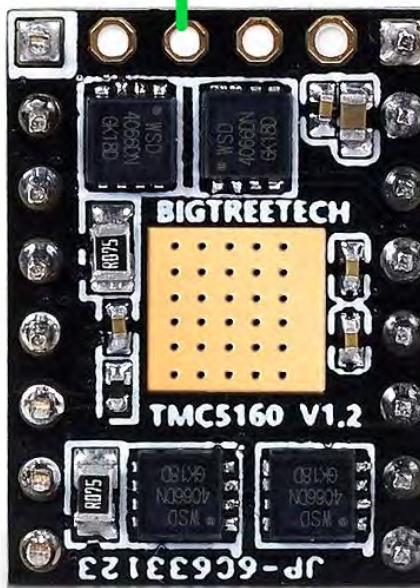
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

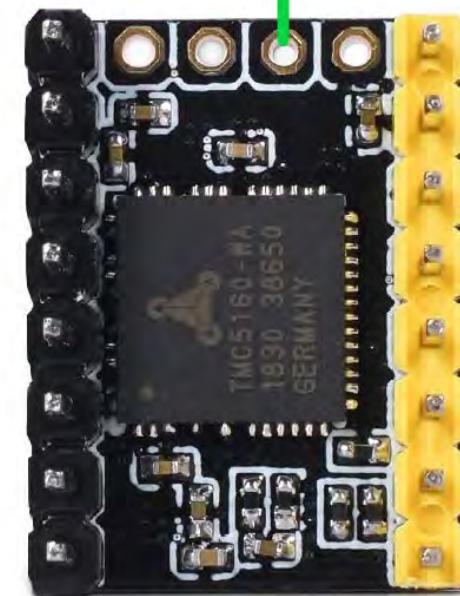
- This next section covers sensor-less homing which is available for the TMC5160 in SPI mode. I want to enable it BUT for the TMC5160 I first have to solder on the DIAG1 pin onto each TMC5160 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and 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.



DIAG1 PIN



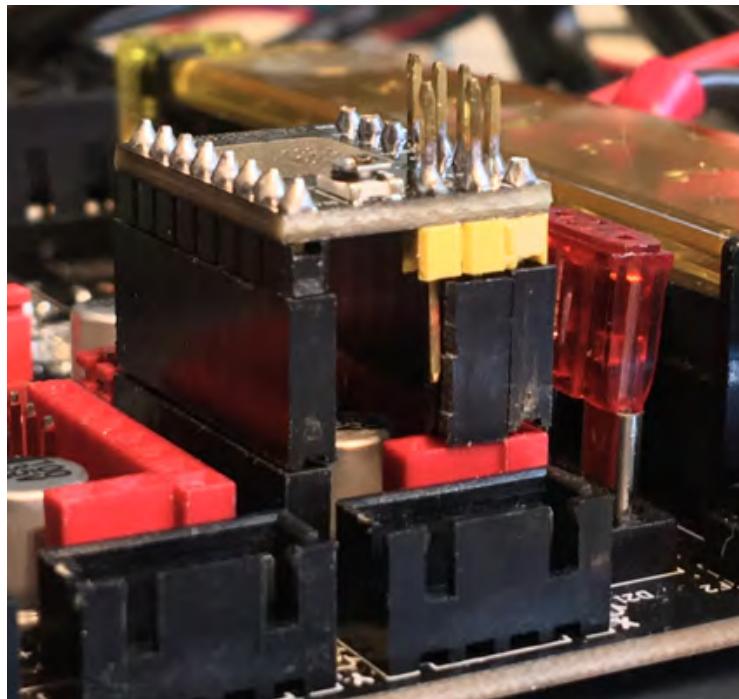
DIAG1 PIN



- 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.



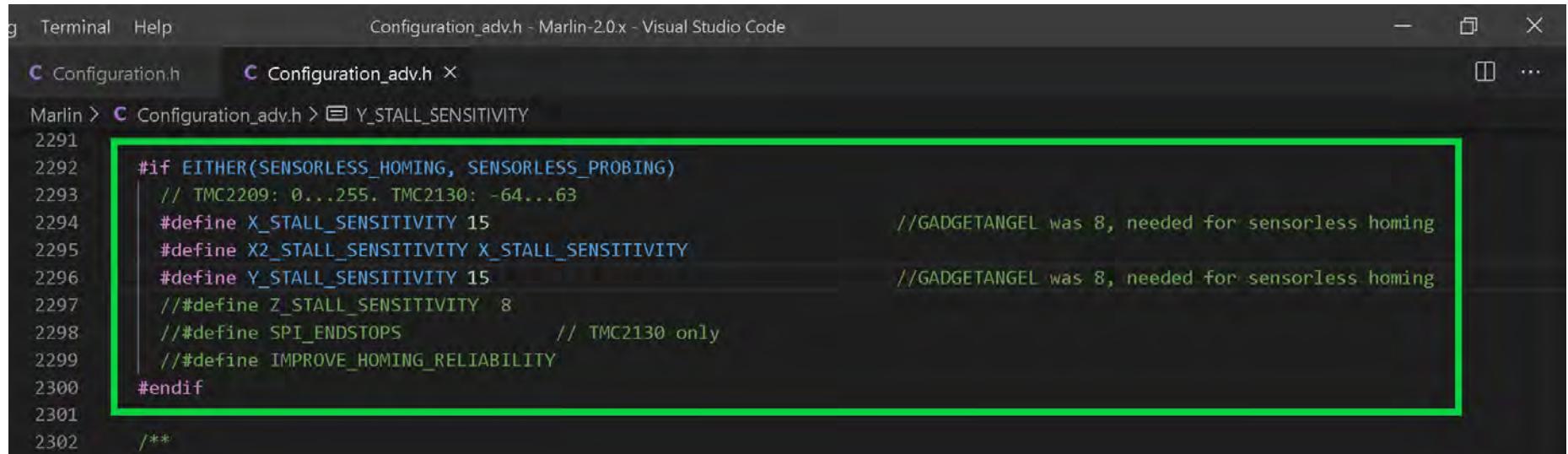
- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

```
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".

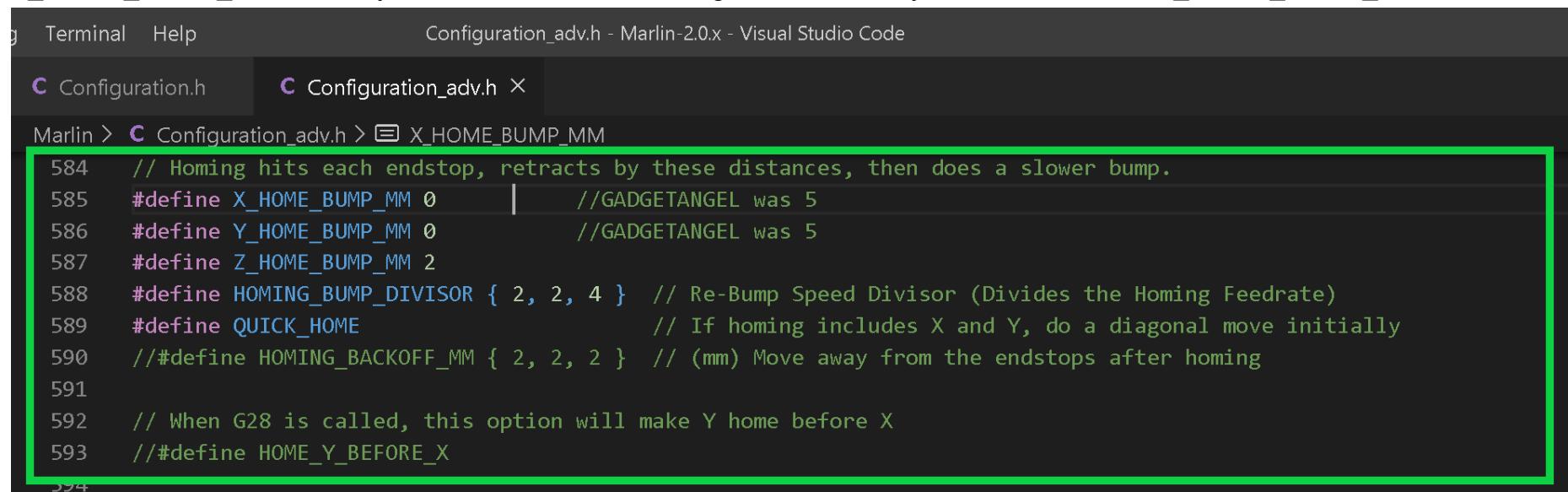


```

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.



```

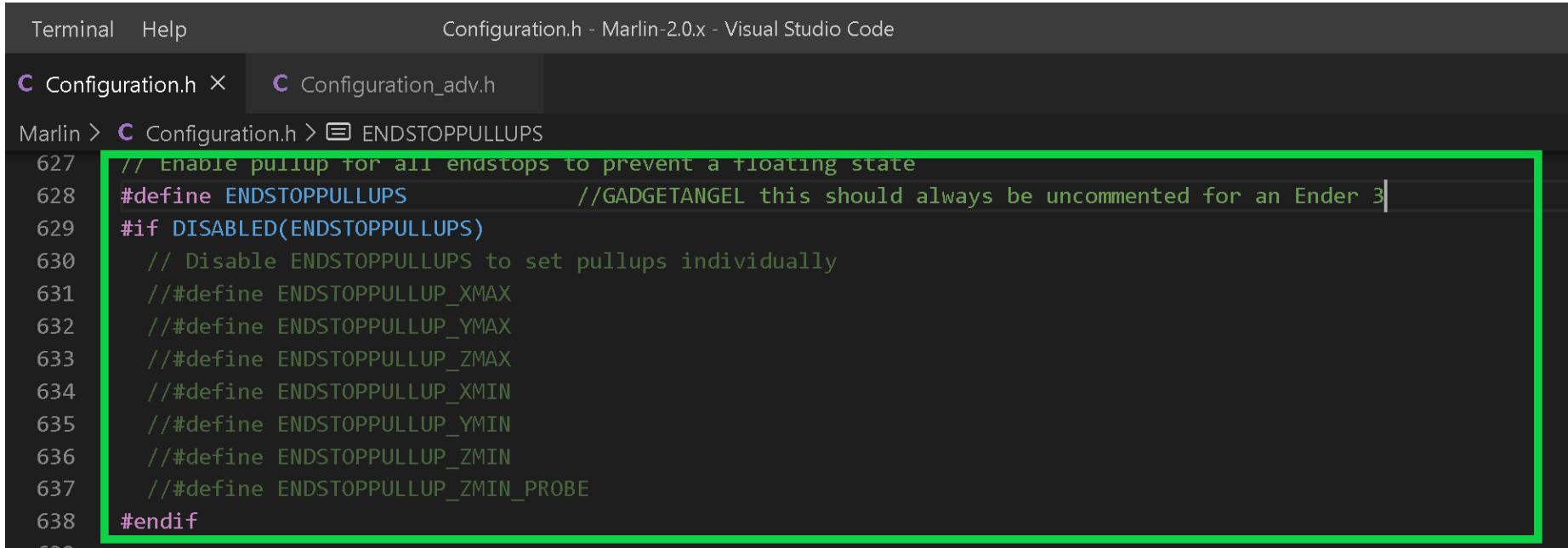
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

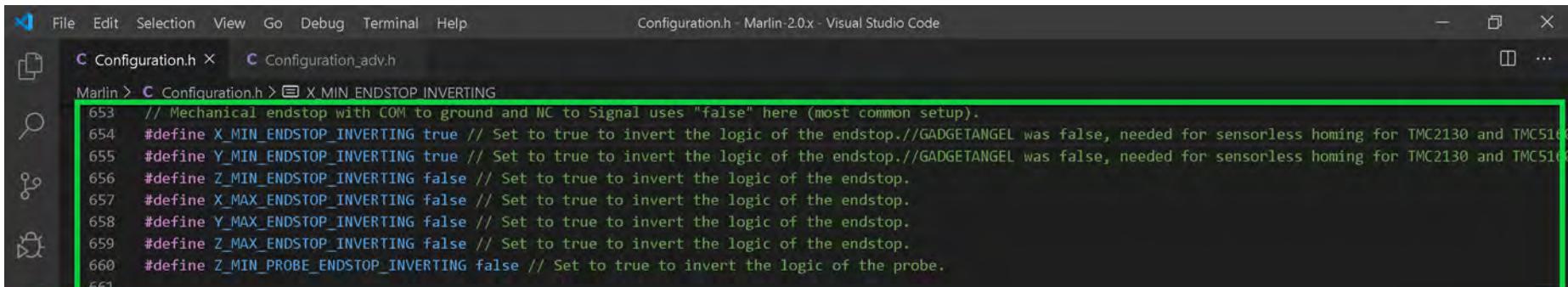
Marlin > Configuration.h > ENDSTOPPULLUPS

```

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

```

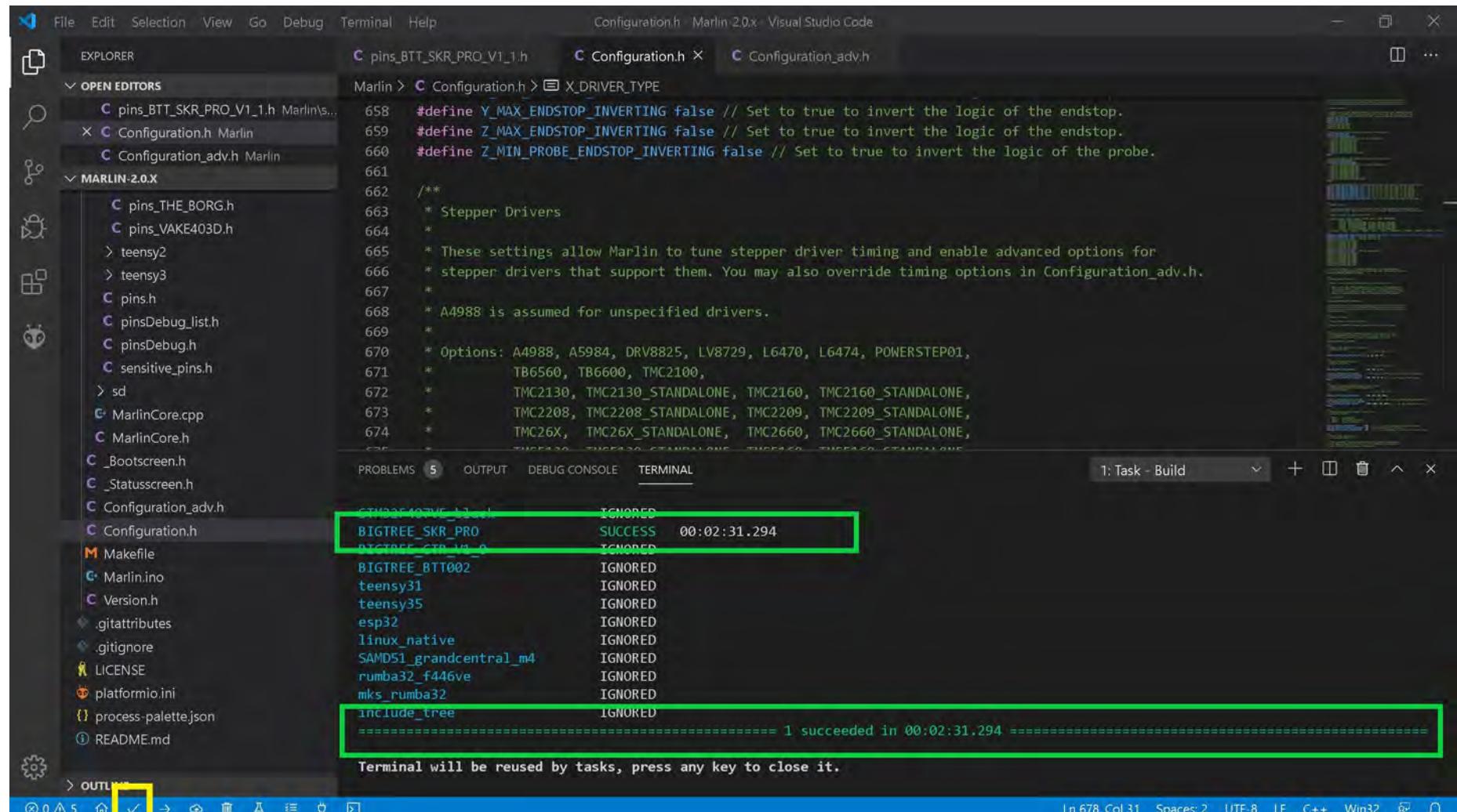
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
655 #define Y_MIN_ENDSTOP_INVERTING 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.

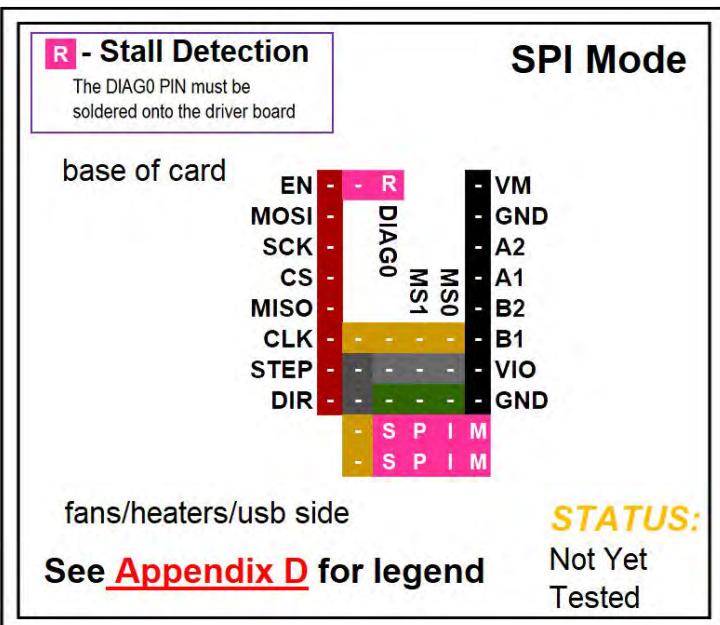


```

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 + ×
Task: 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

```

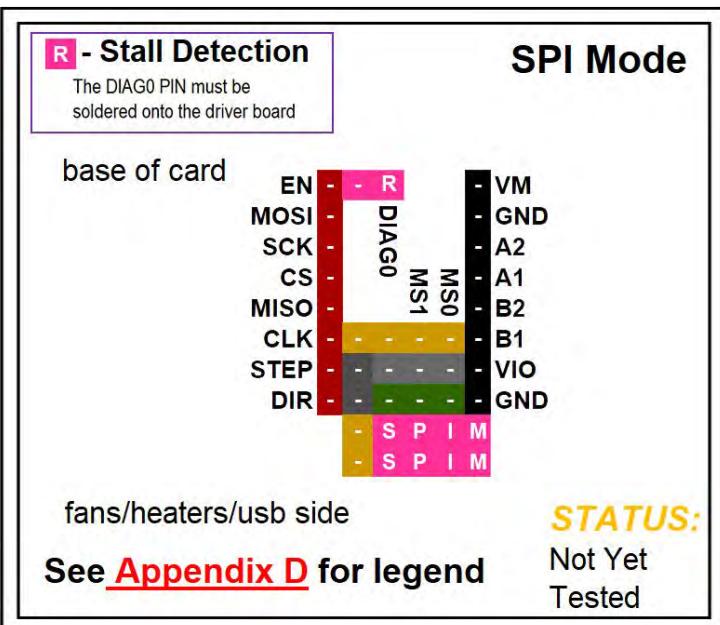
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

**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.

Driver Chip BIQU® TMC5161 SPI Mode Maximum 256 Subdivision 40V DC 5.2A (peak)	Steps are set inside of your Firmware
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.062Ω	$I_{MAX} = 5.2419$ See Appendix B #9. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90 = 4.718$
	Current Limit is set by the current sense resistors (R_s). Use 50% - 90% of I_{MAX} .

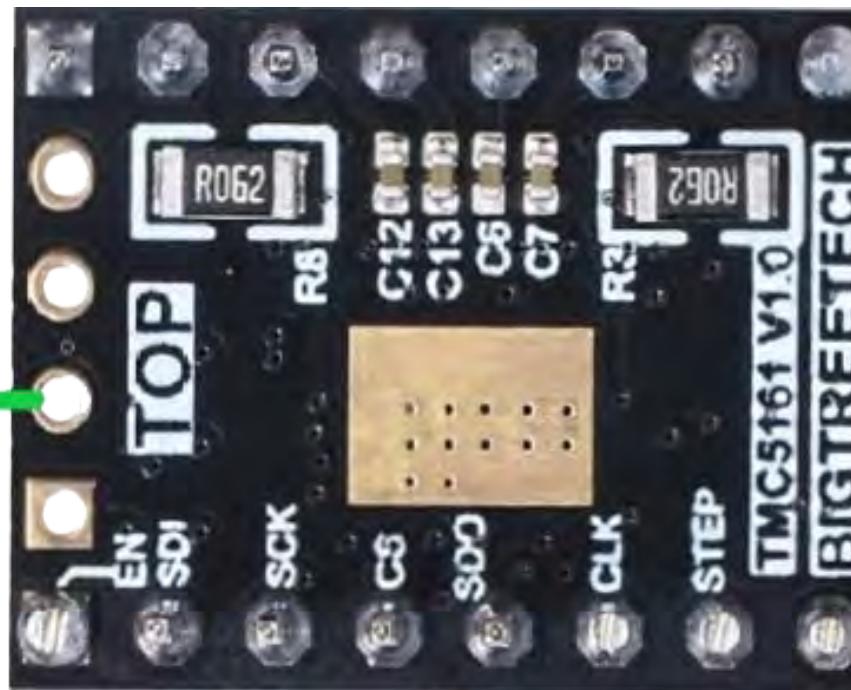


DIAG0

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.

SPI Mode

BIGTREETECH TMC5161 V1.0

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

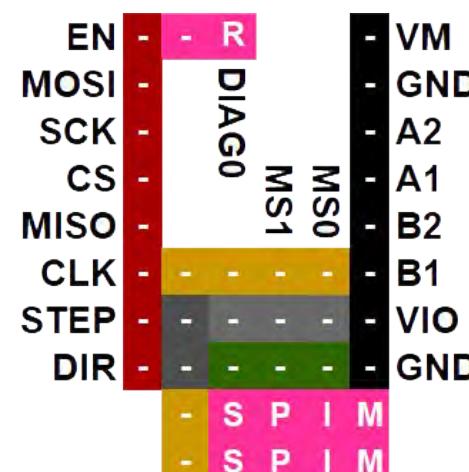
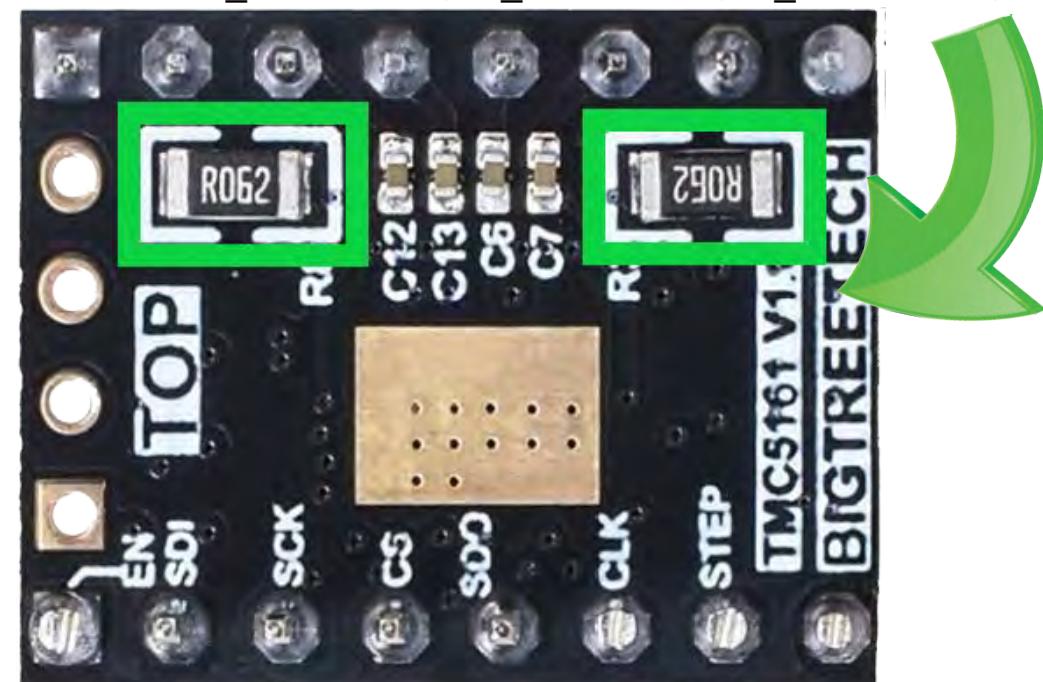
$R_s = R100$ is 0.1 Ohms

$R_s = R110$ is 0.11 Ohms

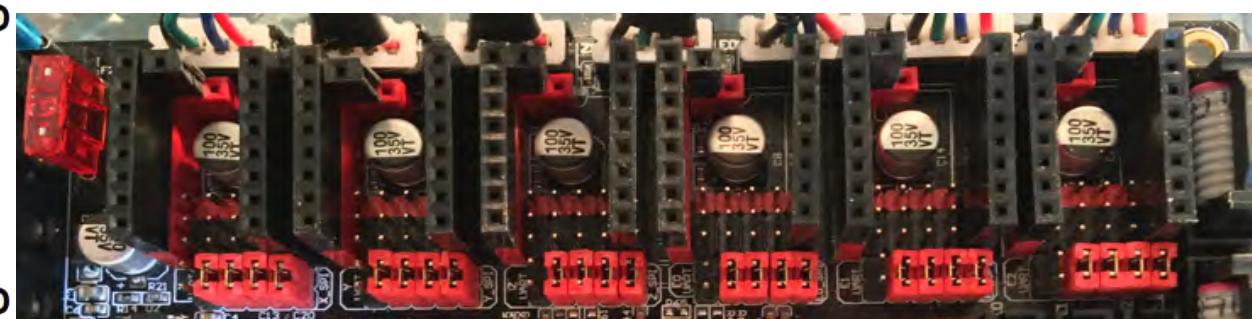
$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



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

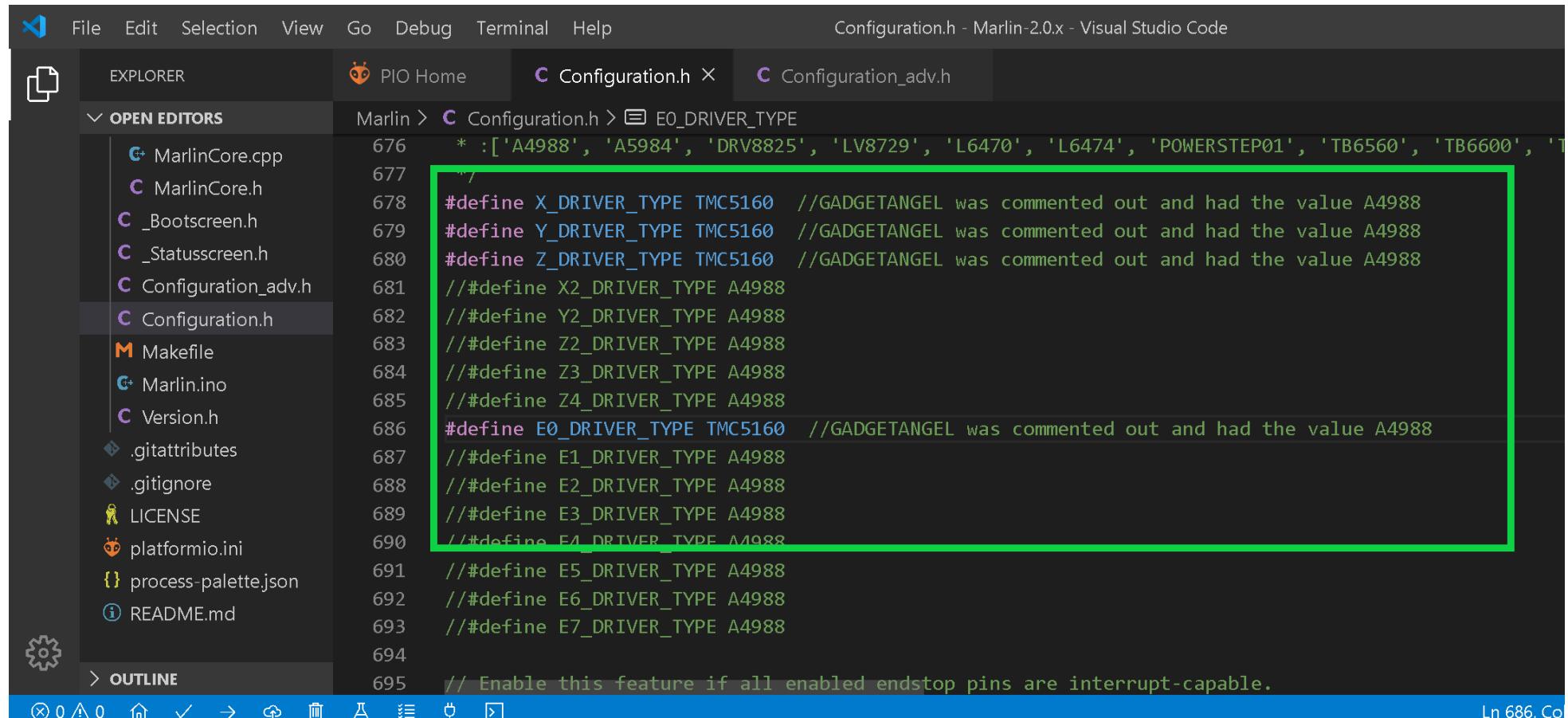


See [Appendix D](#) for legend

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.

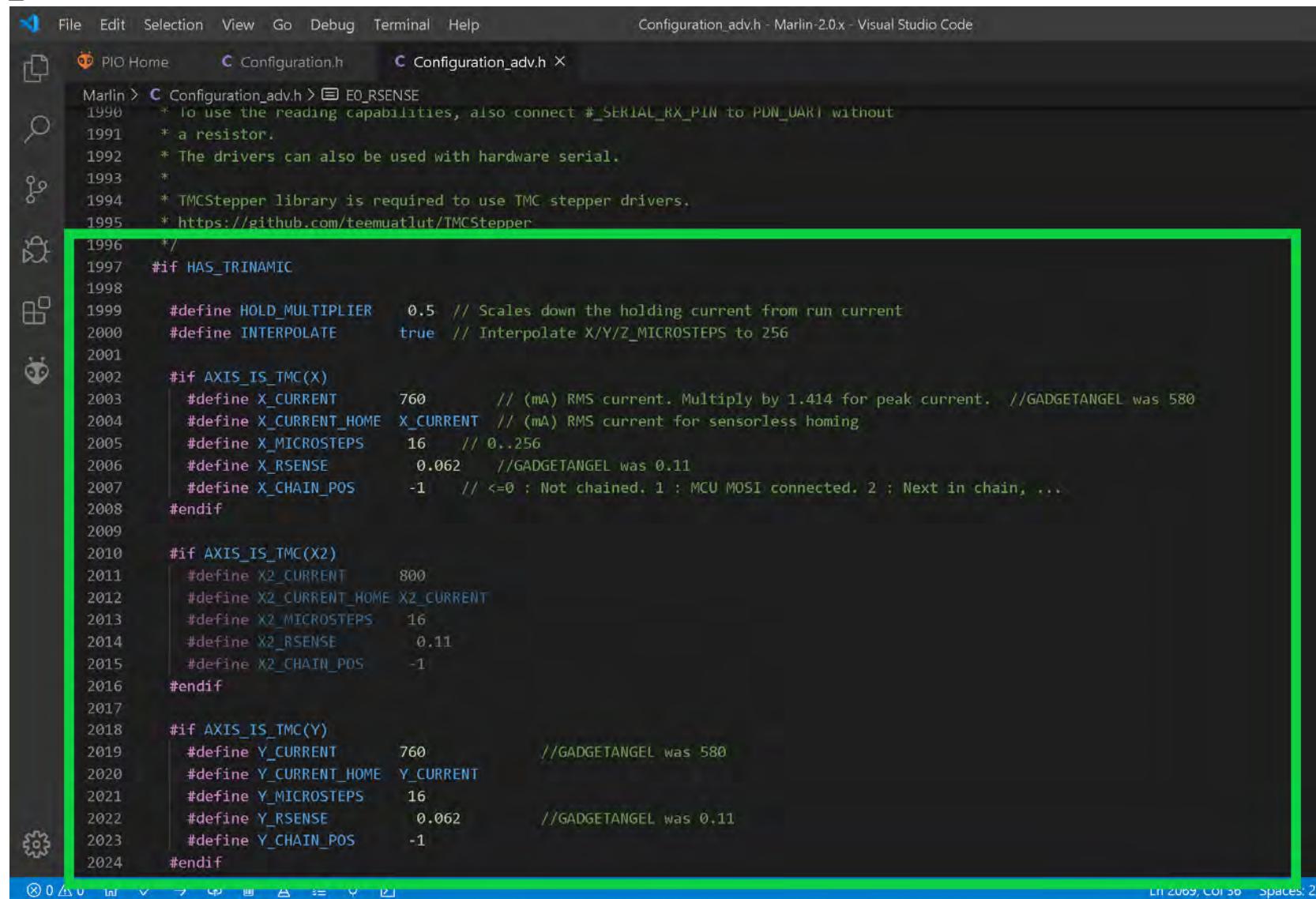
```

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".



```

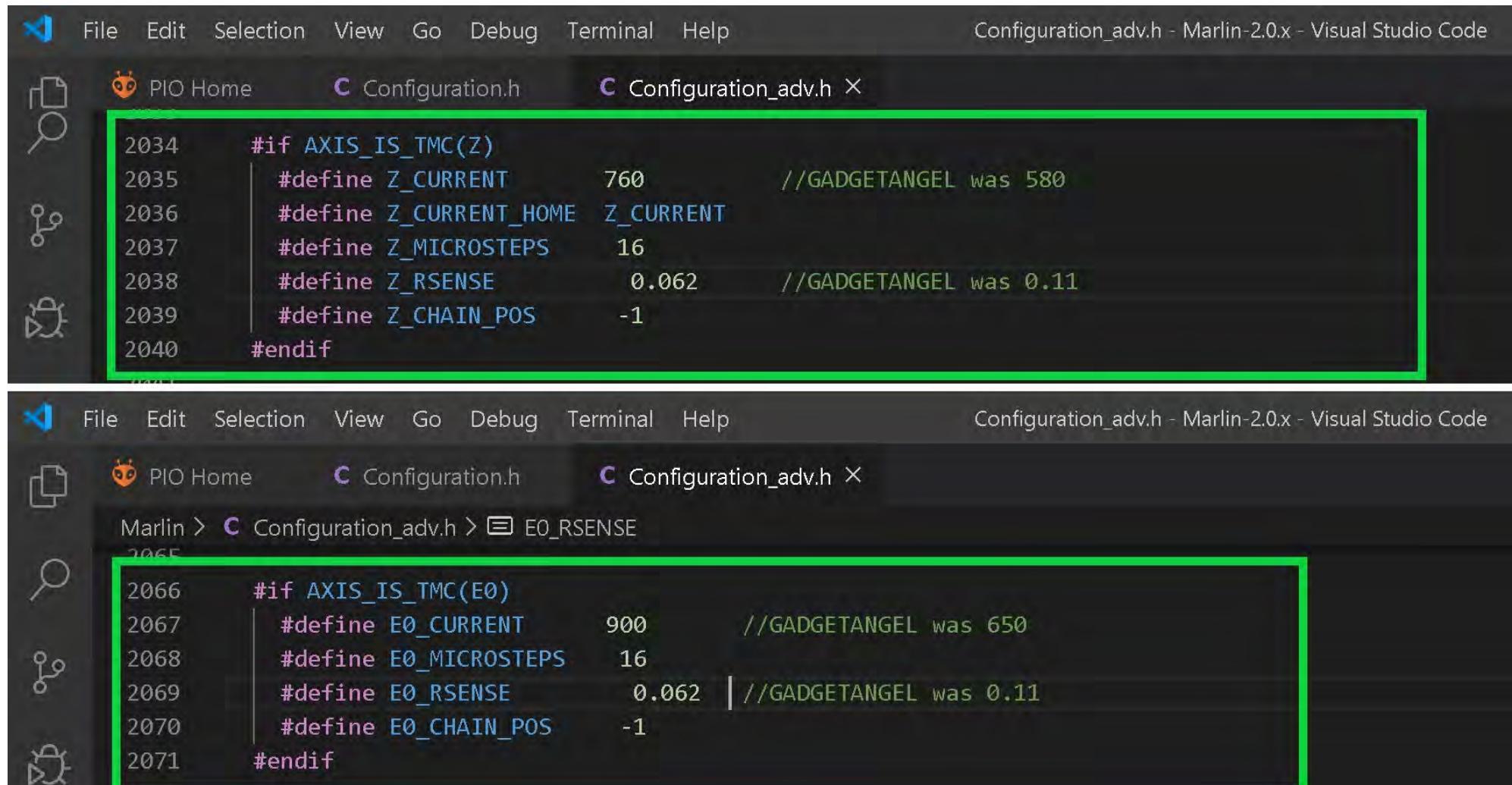
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h X
Marlin > C Configuration_adv.h > EO_RSENSE
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
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

```



```

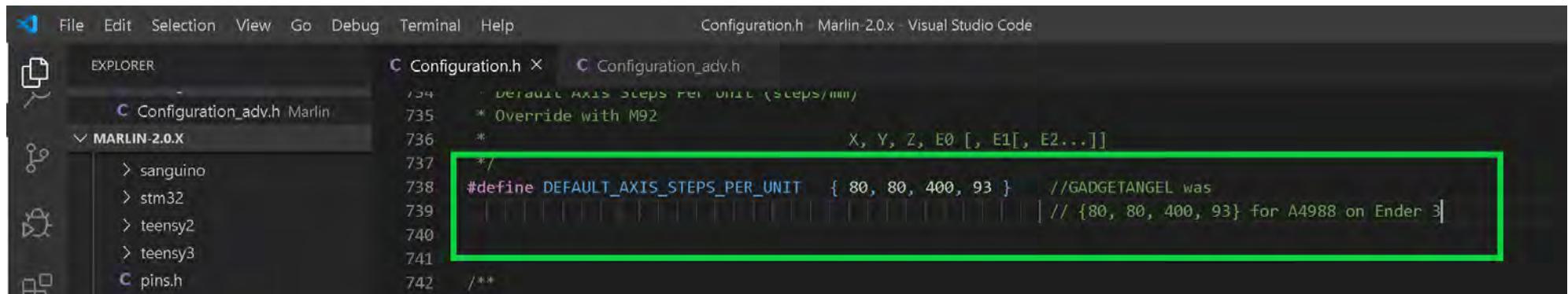
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900          //GADGETANGEL was 650
2068   #define E0_MICROSTEPS    16
2069   #define E0_RSENSE        0.062        //GADGETANGEL was 0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- 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

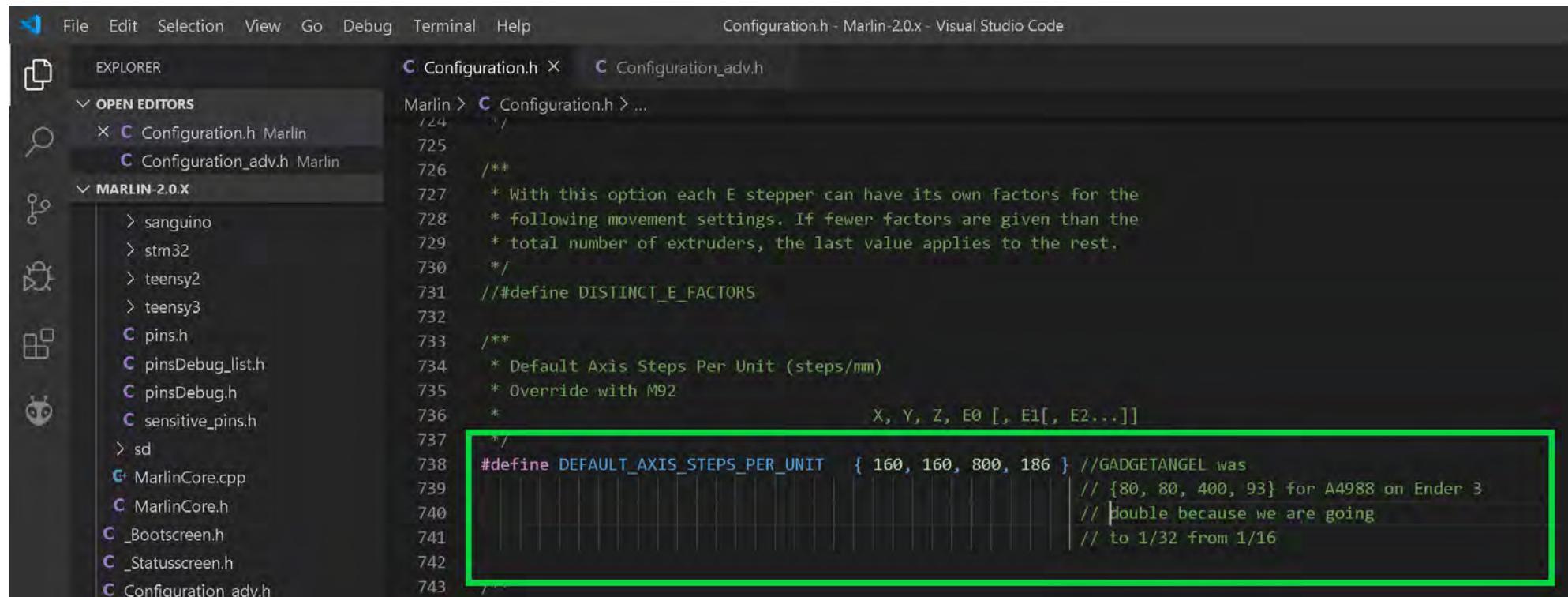
MARLIN-2.0.X Configuration.h Marlin Configuration_adv.h Marlin

```

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 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
OPEN EDITORS Marlin > Configuration_adv.h > TMC_USE_SW_SPI
  Configuration.h Marlin 2144 * The default SW SPI pins are defined the respective pins files,
  Configuration_adv.h Marlin 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.

```

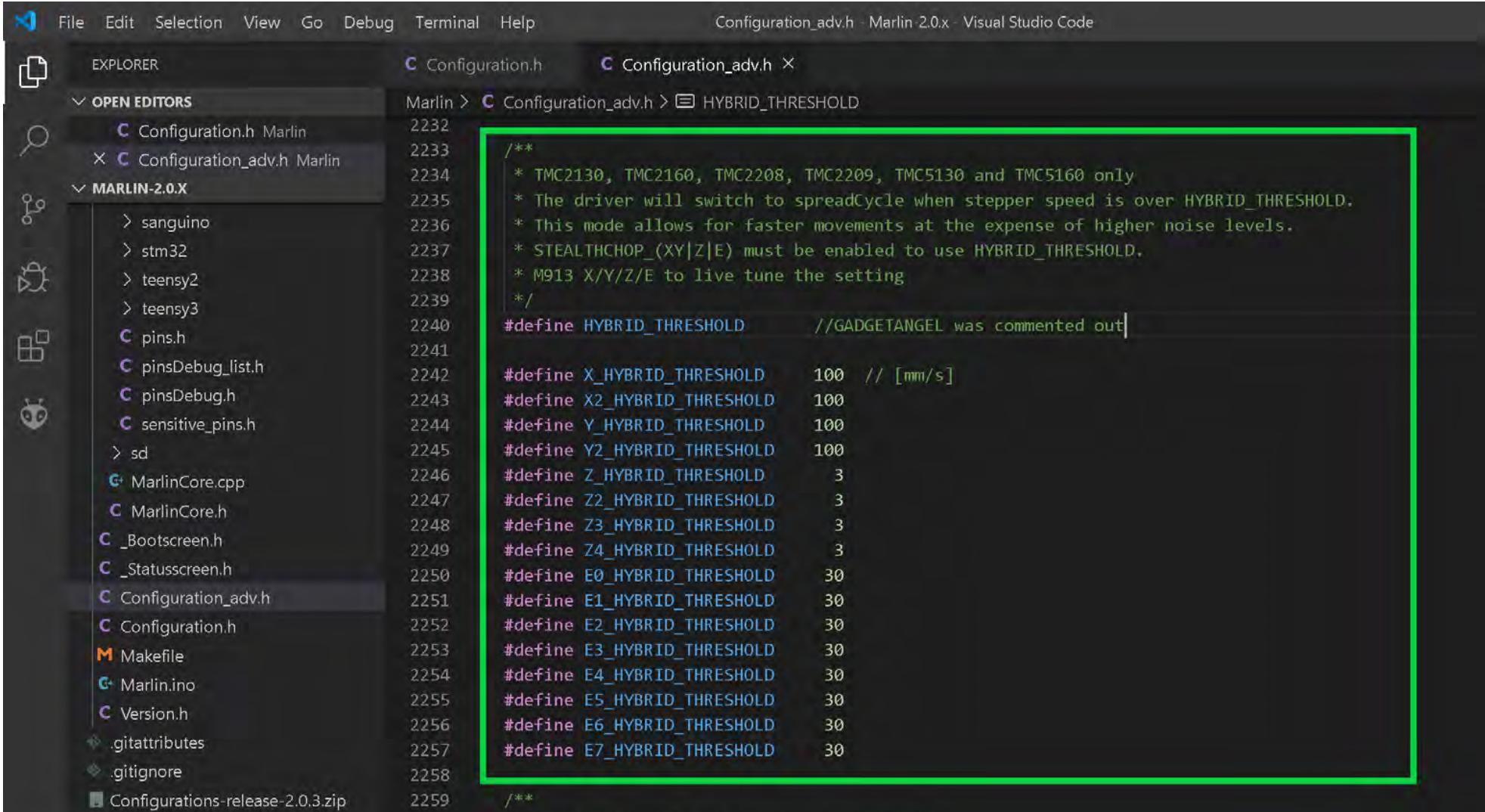
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
  Configuration.h Marlin 2193 */
  Configuration_adv.h Marlin 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
  > sanguino
  > stm32
  > teensy2
  > teensy3
  Configuration.h Marlin
  Configuration_adv.h Marlin
  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

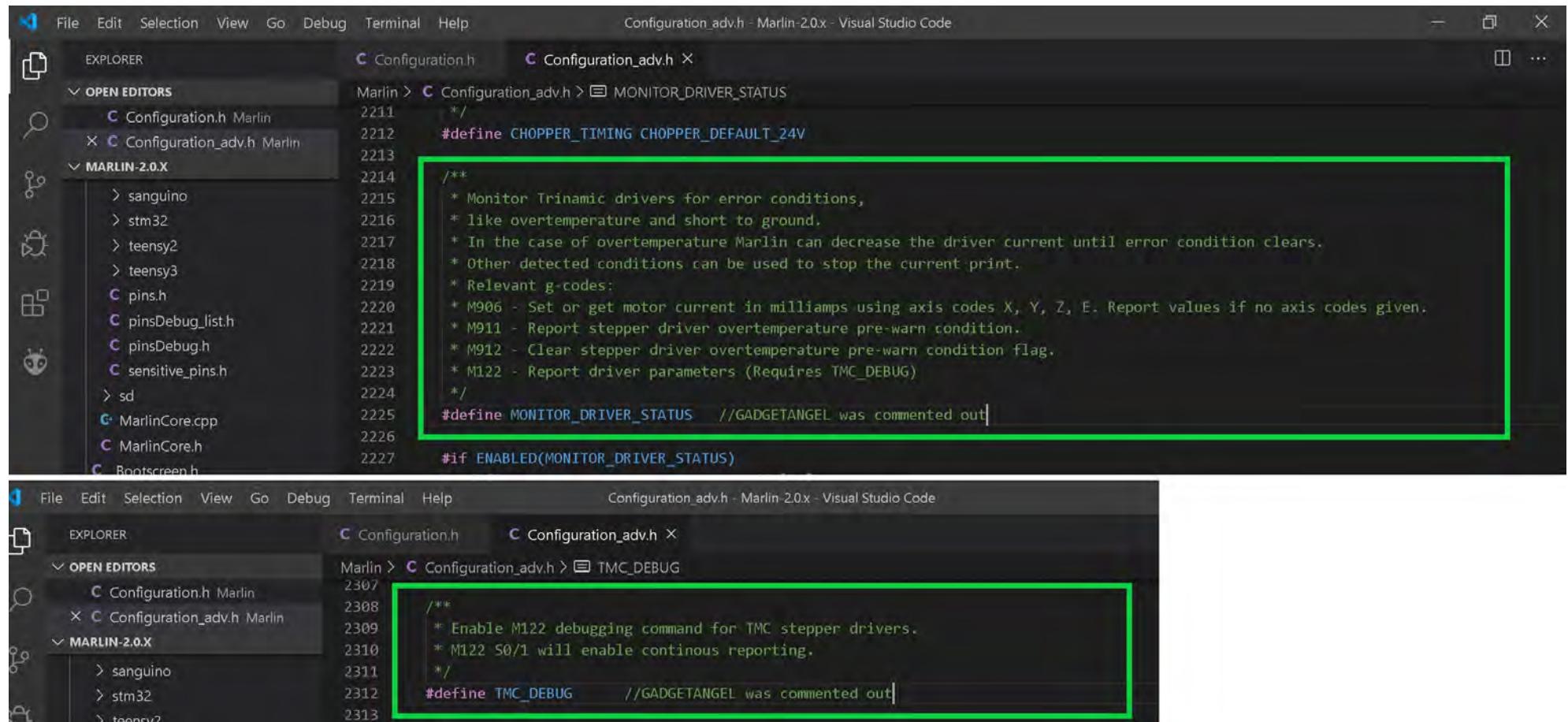
2232
2233 /**
2234 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
2235 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
2236 * This mode allows for faster movements at the expense of higher noise levels.
2237 * STEALTHCHOP_(XY|Z|E) must be enabled to 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 */

```

- 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.](#)



```

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

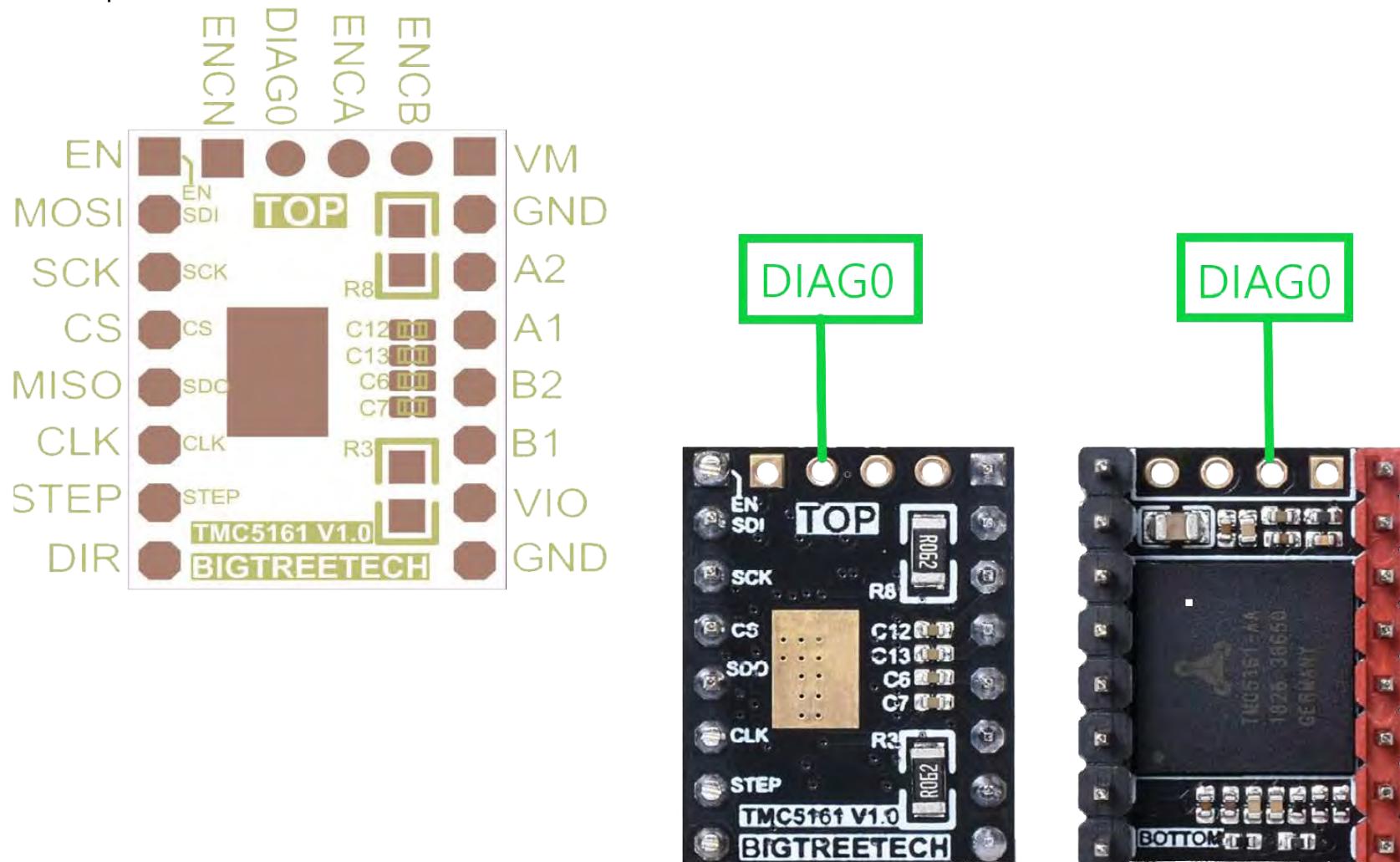
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help Configuration.h Configuration_adv.h
Marlin > C Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

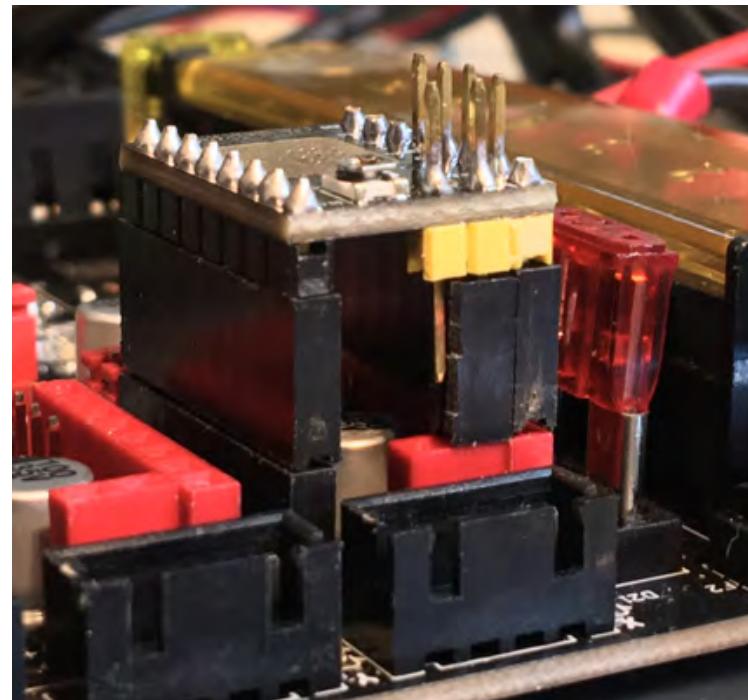
- This next section covers sensor-less homing which is available for the TMC5161 in SPI mode. I want to enable it BUT for the TMC5161 I first have to solder on the DIAG0 pin onto each TMC5161 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and 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.



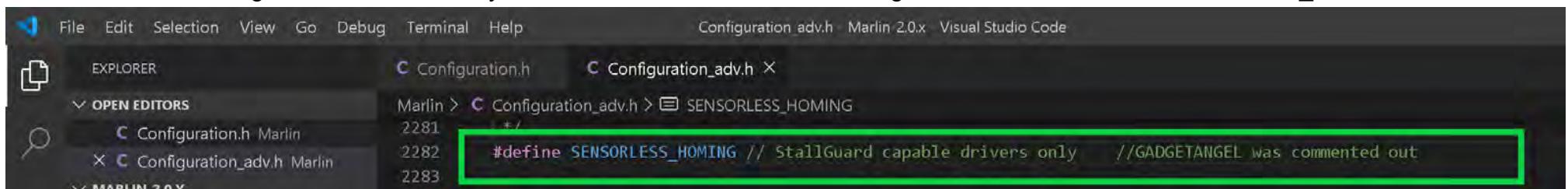
- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR PRO V1.1 board the DIAG0 pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

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

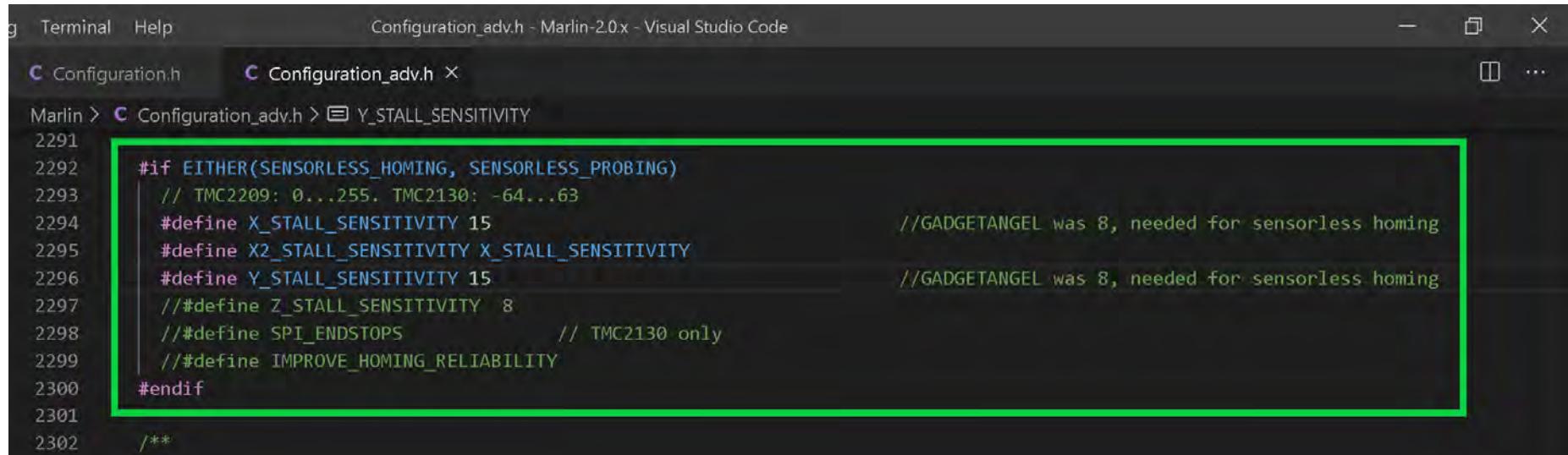
EXPLORER Configuration.h Configuration_adv.h ×
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X

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 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".

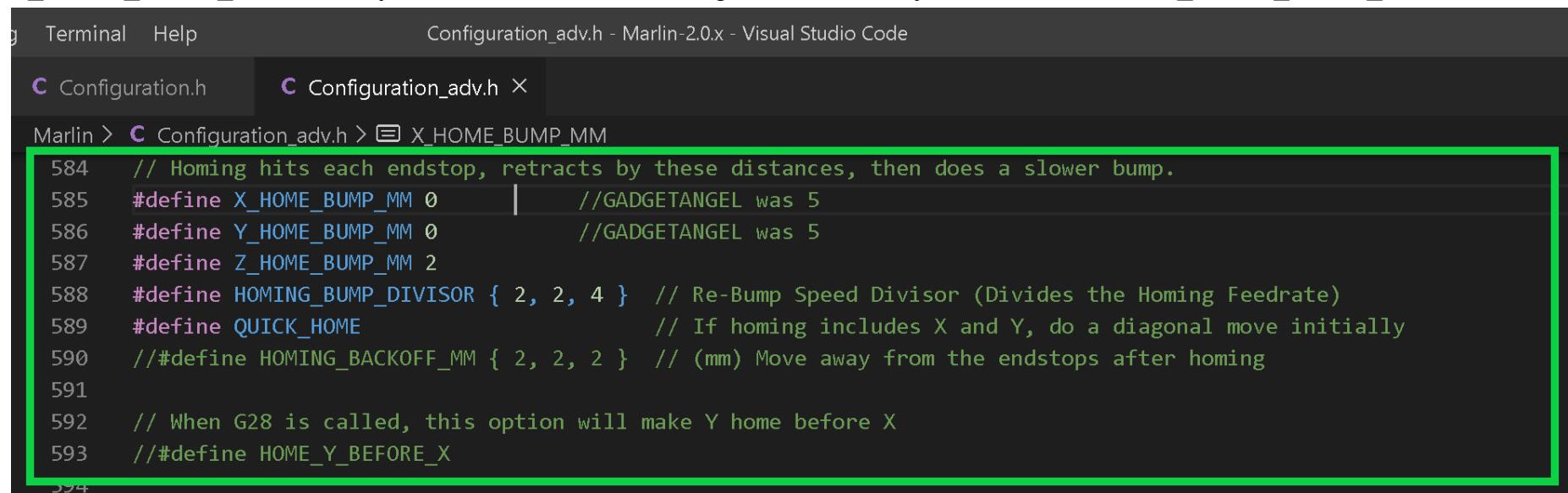


```

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.



```

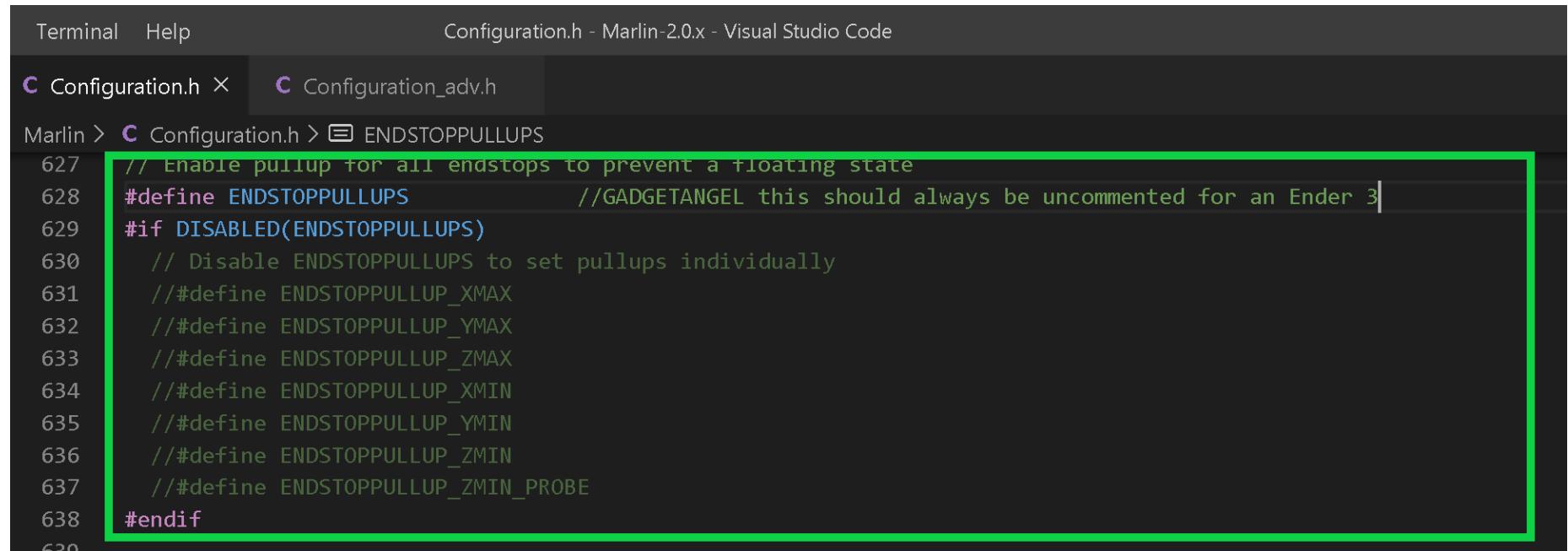
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

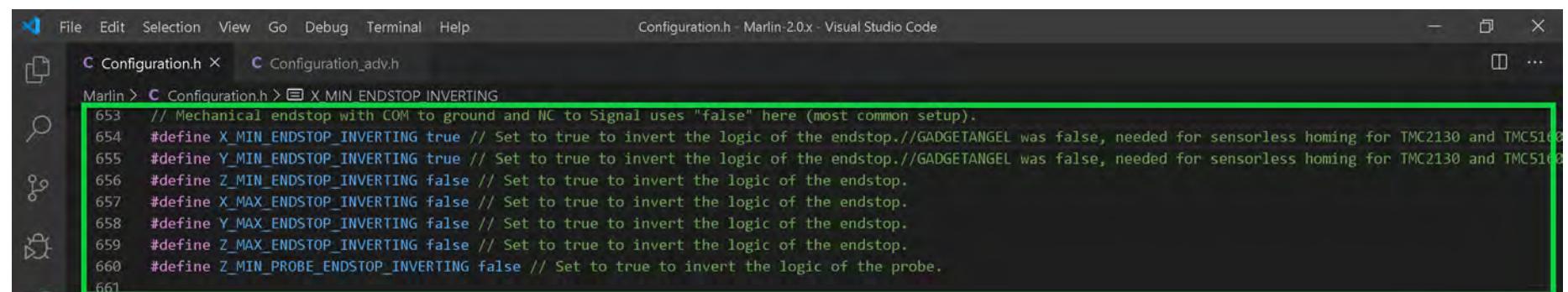
C Configuration.h X C Configuration_adv.h

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 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
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > X MIN ENDSTOP INVERTING

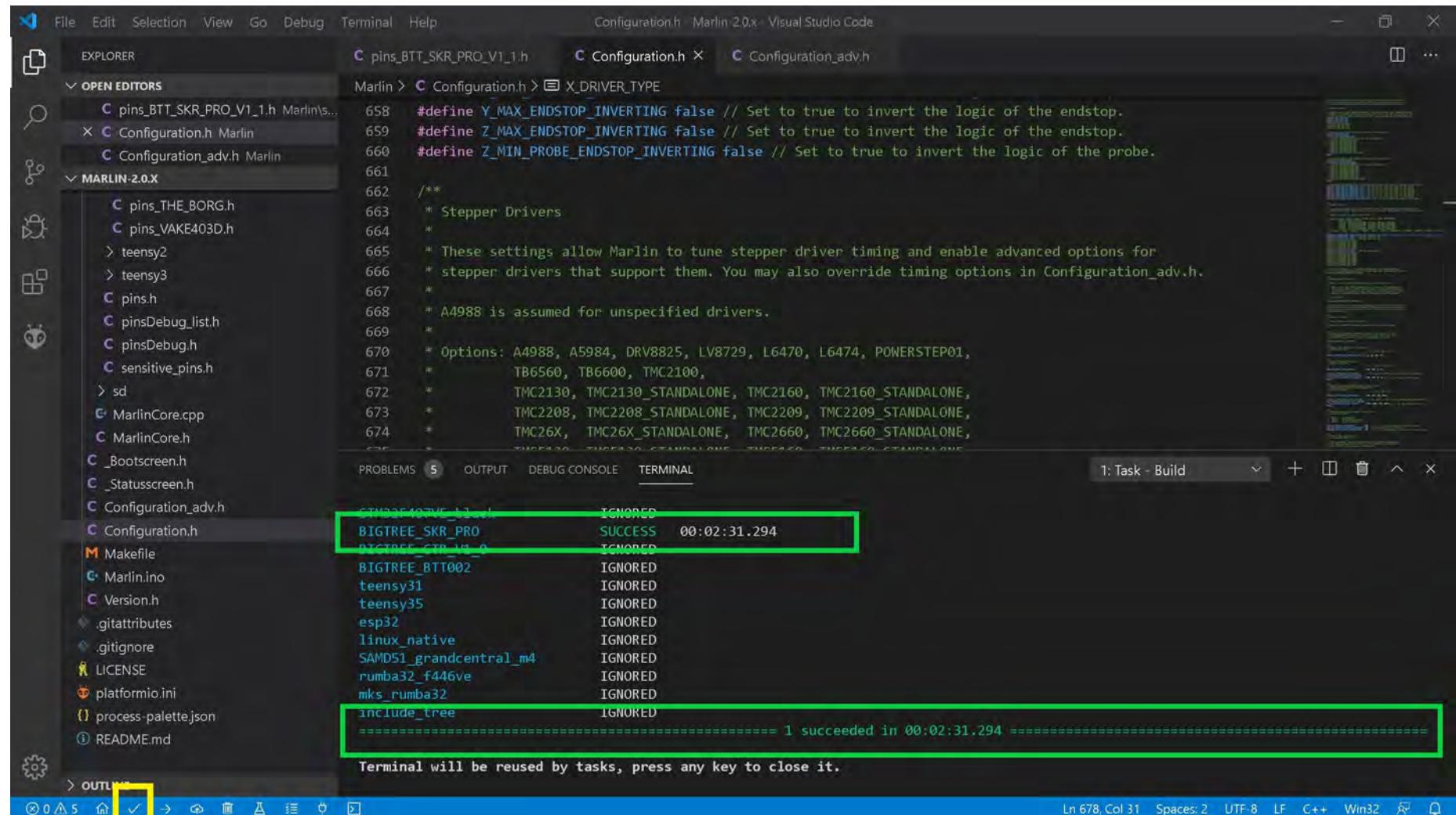
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
655 #define Y_MIN_ENDSTOP_INVERTING 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.



```

Configuration.h Marlin 2.0.x Visual Studio Code
File Edit Selection View Go Debug Terminal Help
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,
TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + - X
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 [Appendix E](#).

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

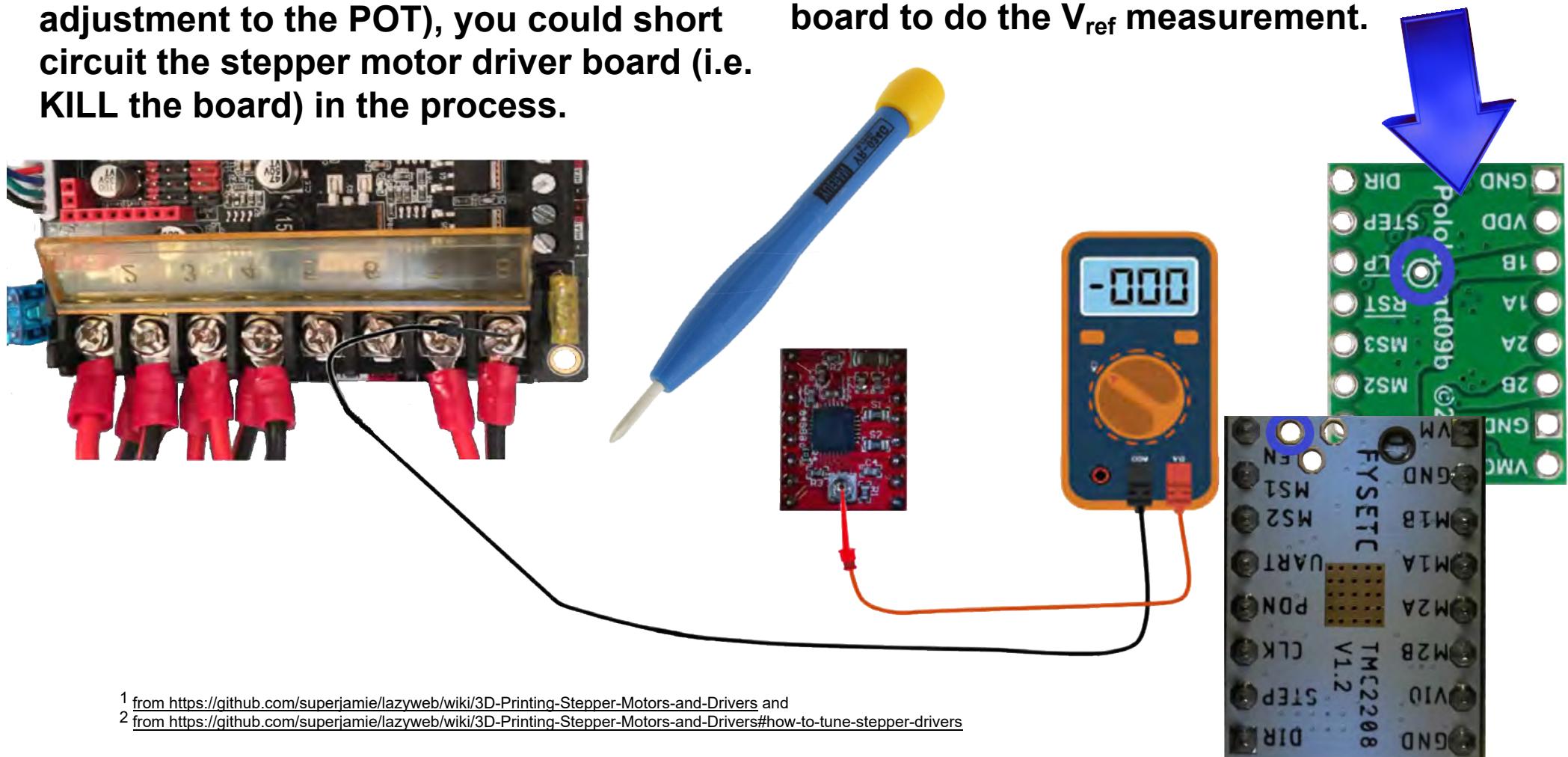
² from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

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 $R_s = 0.110\Omega$ (110m Ω) :**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320\text{mV} / (R_s + 20\text{m}\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.46)) \\ &= (1 / 1.41) * V_{ref} * 0.99 \\ &= (1 / 1.41) * V_{ref} * 1 \end{aligned}$$

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 $R_s = 0.110\Omega$ (110m Ω) :

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325\text{mV} / (R_s + 20\text{m}\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5)) \\ &= (1 / 1.41) * V_{ref} \end{aligned}$$

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 $R_s = 0.110\Omega$ (110m Ω) for Stand-alone Mode:

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325\text{mV} / (R_s + 30\text{m}\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 30))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.32143)) \\ &= (1 / 1.41) * V_{ref} * 0.928572 \end{aligned}$$

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,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * 4.333$$

$$I_{MAX} = 4.333 \text{ Amps}$$

You will use **50% to 90%** of I_{MAX} ($4.333 * 0.50$ or $4.333 * 0.90$) 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,

$$\begin{aligned} I_{MAX} * (1 / 1.41) &= (1 / 1.41) * 5.24194 \\ I_{MAX} &= 5.24194 \text{ or } 5.2419 \end{aligned}$$

You will use **50% to 90%** of I_{MAX} ($5.2419 * 0.50$ or $5.2419 * 0.90$) 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,

$$\begin{aligned} I_{MAX} * (1 / 1.41) &= (1 / 1.41) * V_{ref} * 0.7222 \\ I_{MAX} &= V_{ref} * 0.7222 \\ V_{ref} &= I_{MAX} * 1.3846 \end{aligned}$$

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](#) 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	

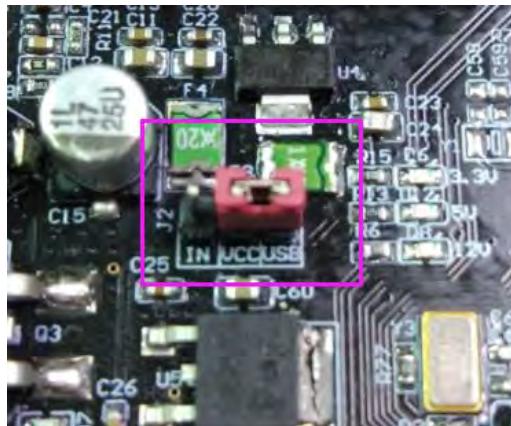
Marlin "Nightly" Builds			
Description	Version	Download	Configurations
Patches to latest 2.0.x Marlin 2.0 with bug fixes Supports AVR and ARM Arduino and PlatformIO	bugfix-2.0.x	bugfix-2.0.x.zip View / Download	
Proceed with Caution! Marlin 2.1 development Supports AVR and ARM Arduino and PlatformIO	dev-2.1.x	dev-2.1.x.zip View / Download	

- Go to the next page.

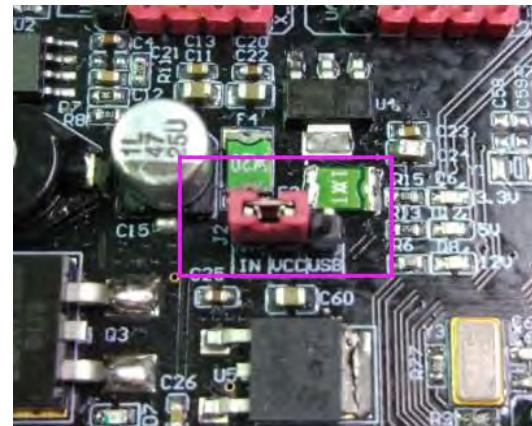
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

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:

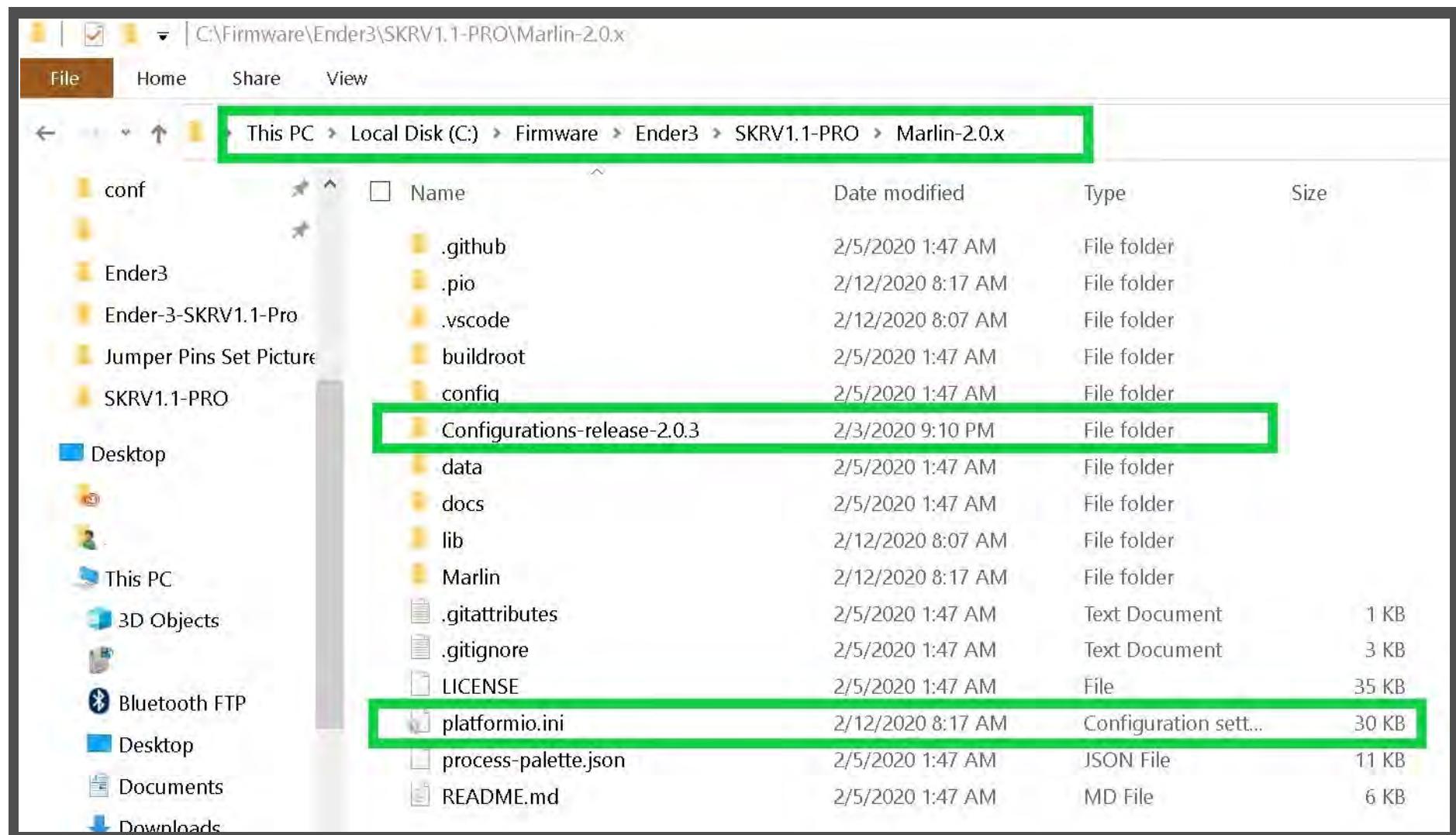
After downloading the files, use Visual Studio Code (VScode) to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the micro SD card into the board. Reboot the board; wait for 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: [How to Calibrate your 3D printer](#)**

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.

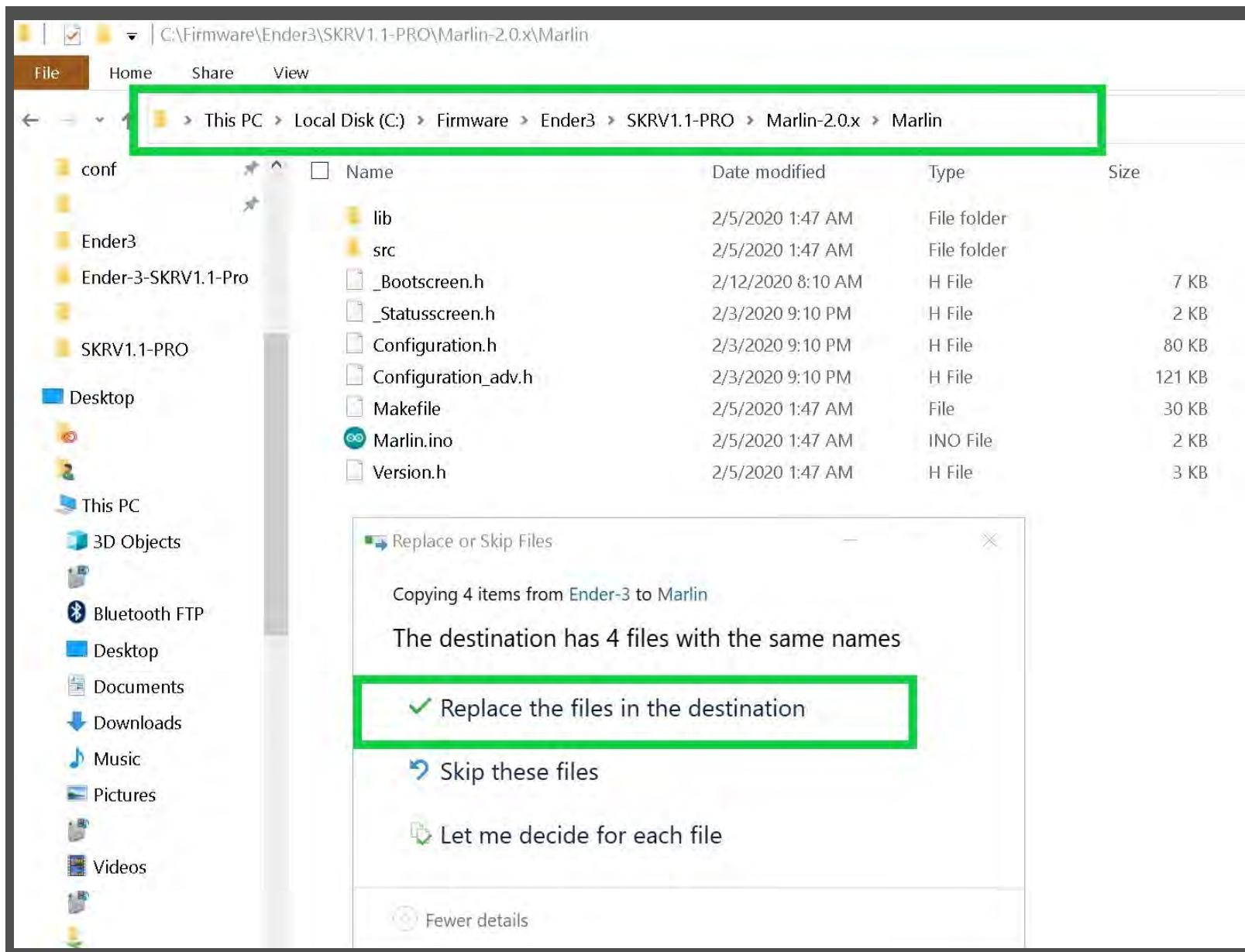


- Go to the next page.

APPENDIX C

The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

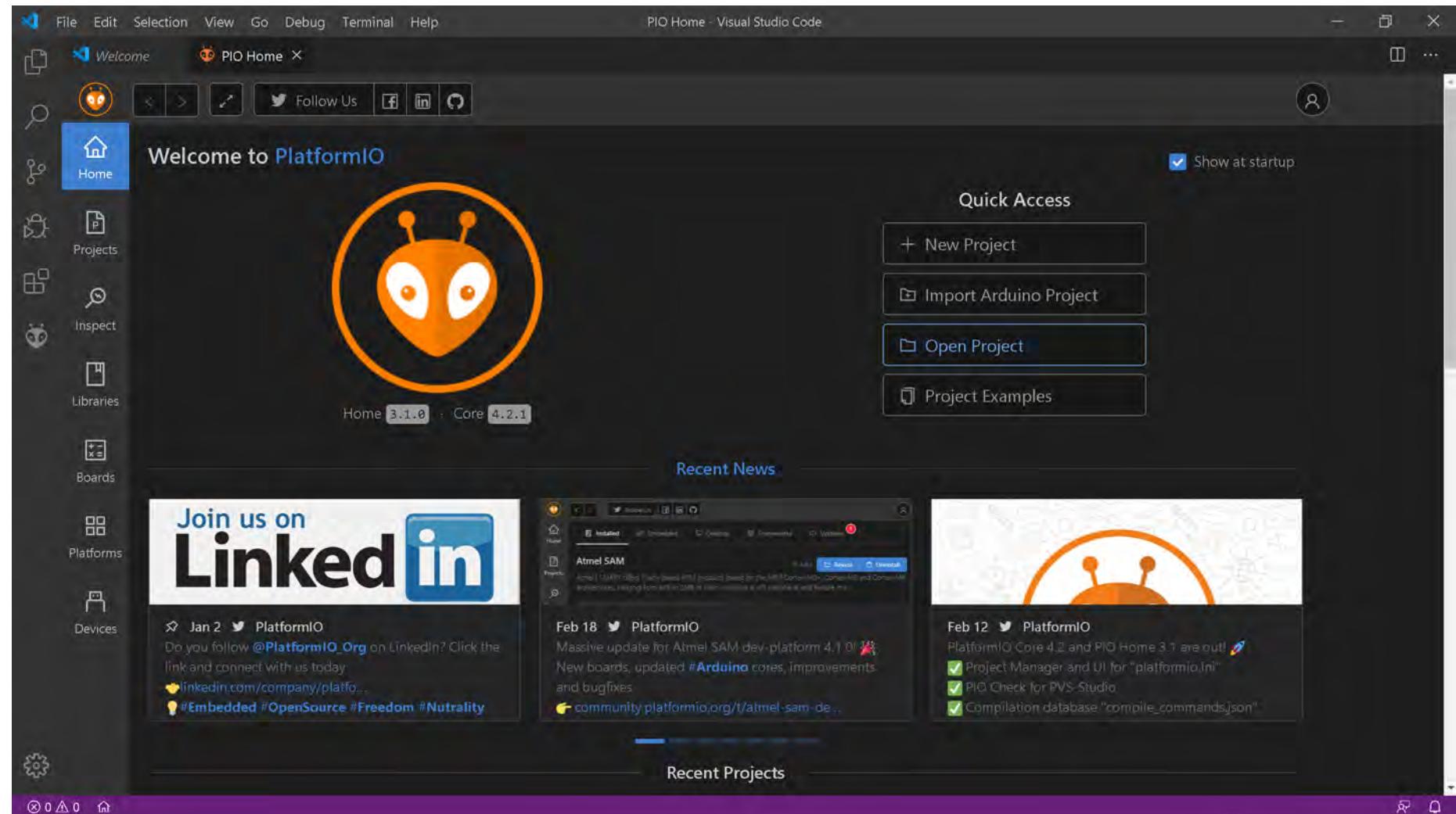
- Copy TO THIS directory so the above files reside in the same directory as Configuration.h and Configuration_adv.h. When prompted allow the files to be overwritten!



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

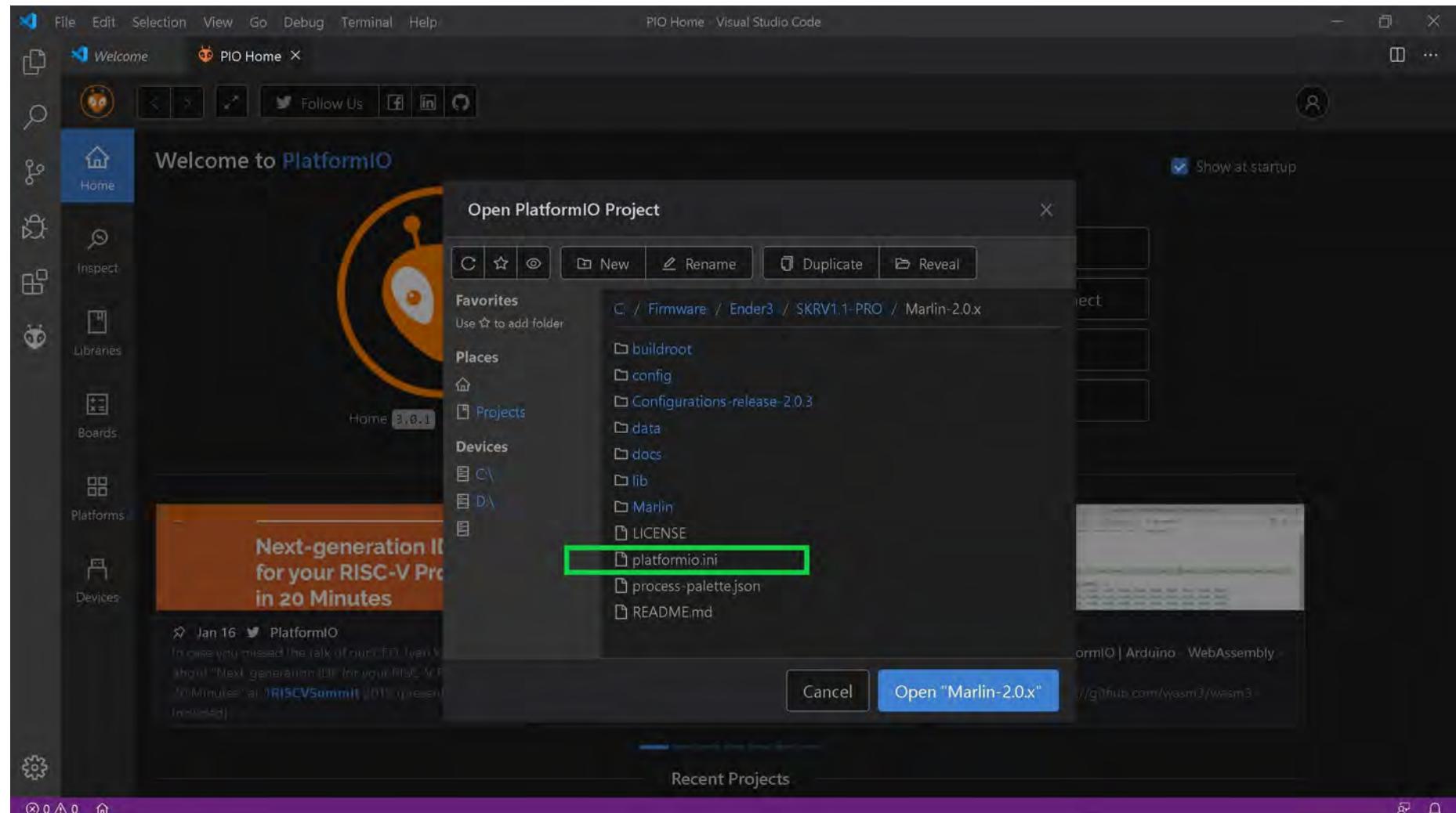
- Open VS code (see picture below) and then select "Open Project".



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

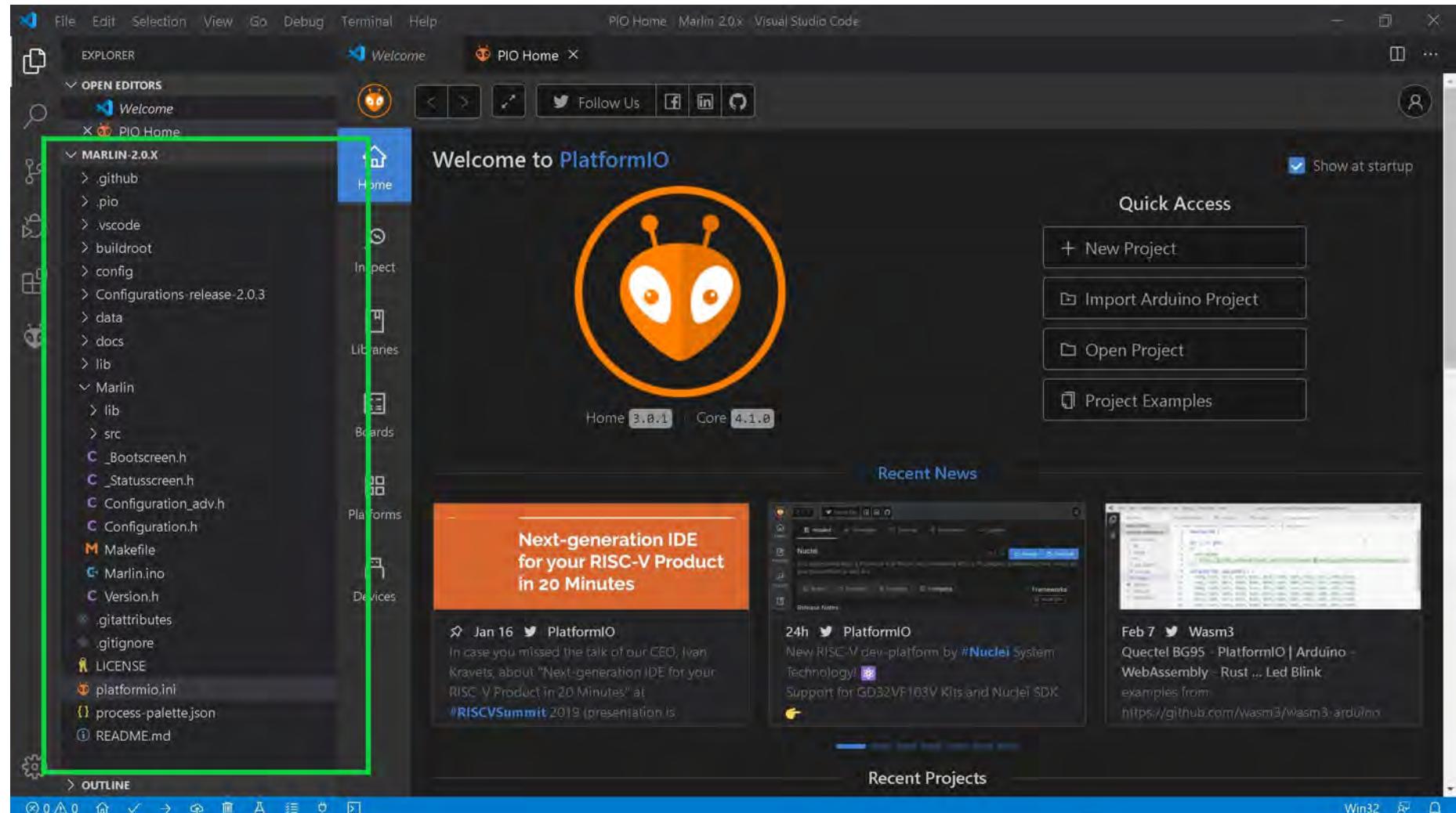
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- Go to the next page.

The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

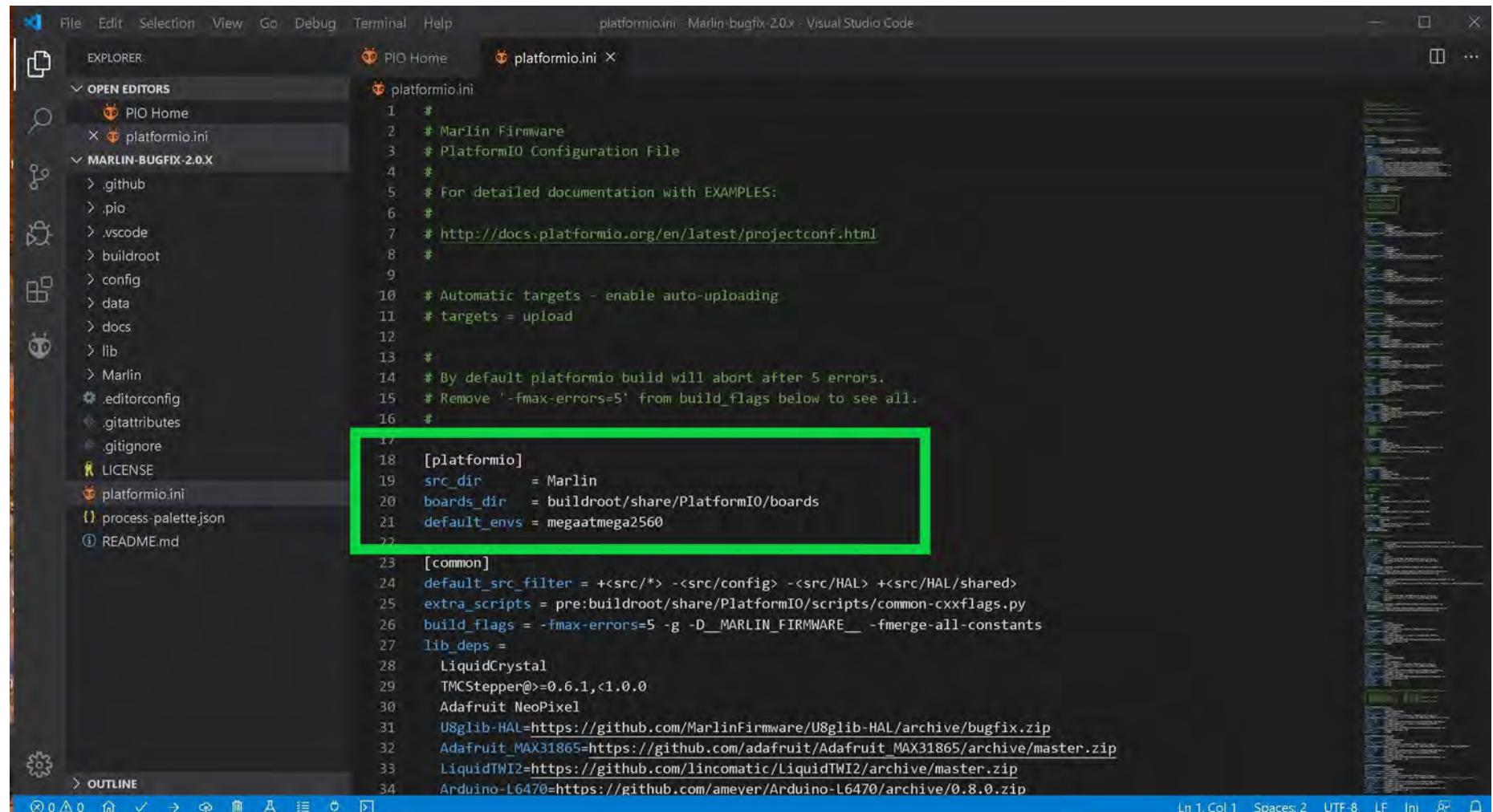
- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



- 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



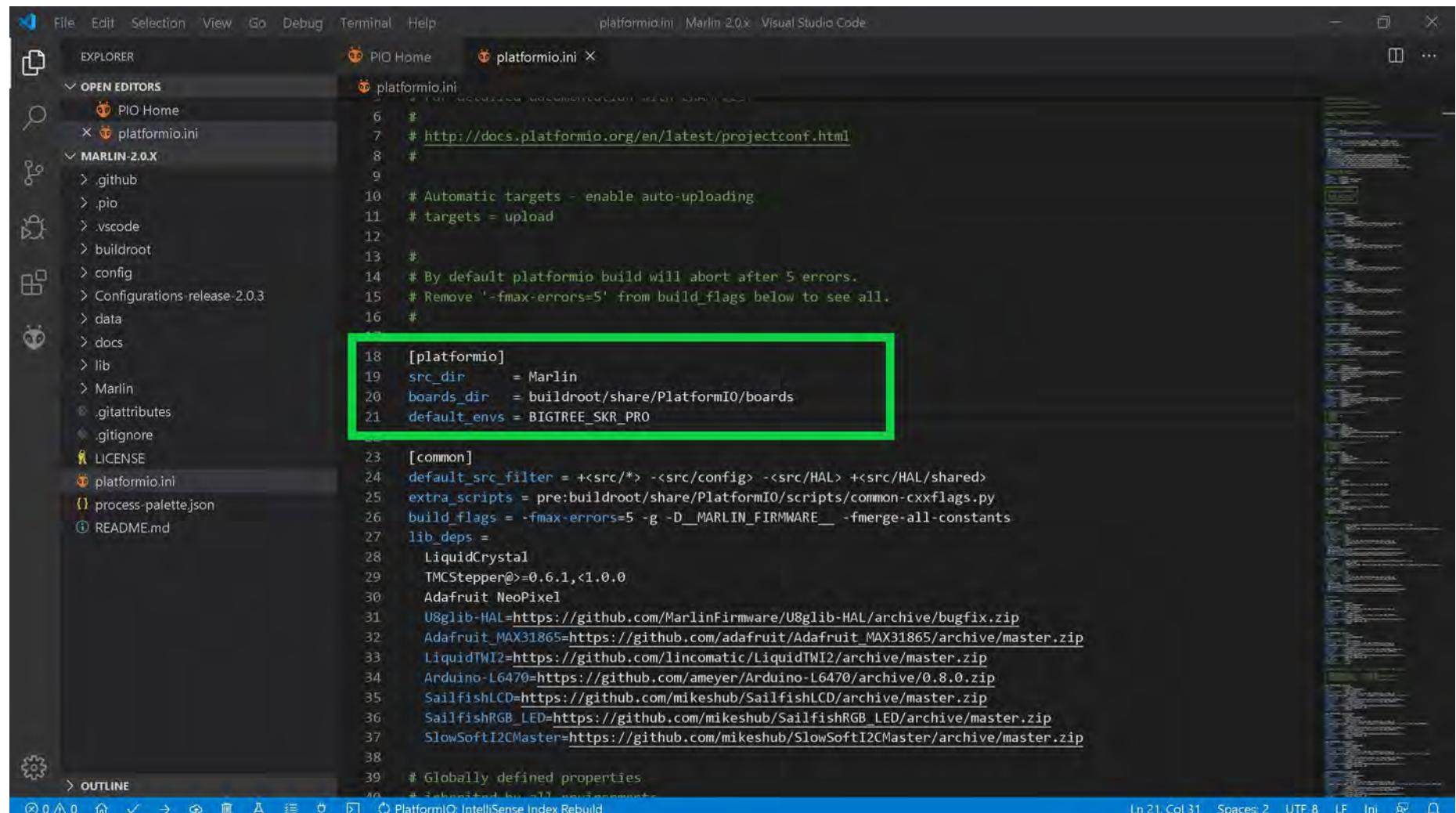
```
platformio.ini - Marlin-bugfix-2.0.x - Visual Studio Code

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  L  O  B
```

- 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**



```
# http://docs.platformio.org/en/latest/projectconf.html
# Automatic targets - enable auto-uploading
# targets = upload
#
# By default platformio build will abort after 5 errors.
# Remove '-fmax-errors=5' from build_flags below to see all.
#
[platformio]
src_dir      = Marlin
boards_dir   = buildroot/share/PlatformIO/boards
default_envs = BIGTREE_SKR_PRO

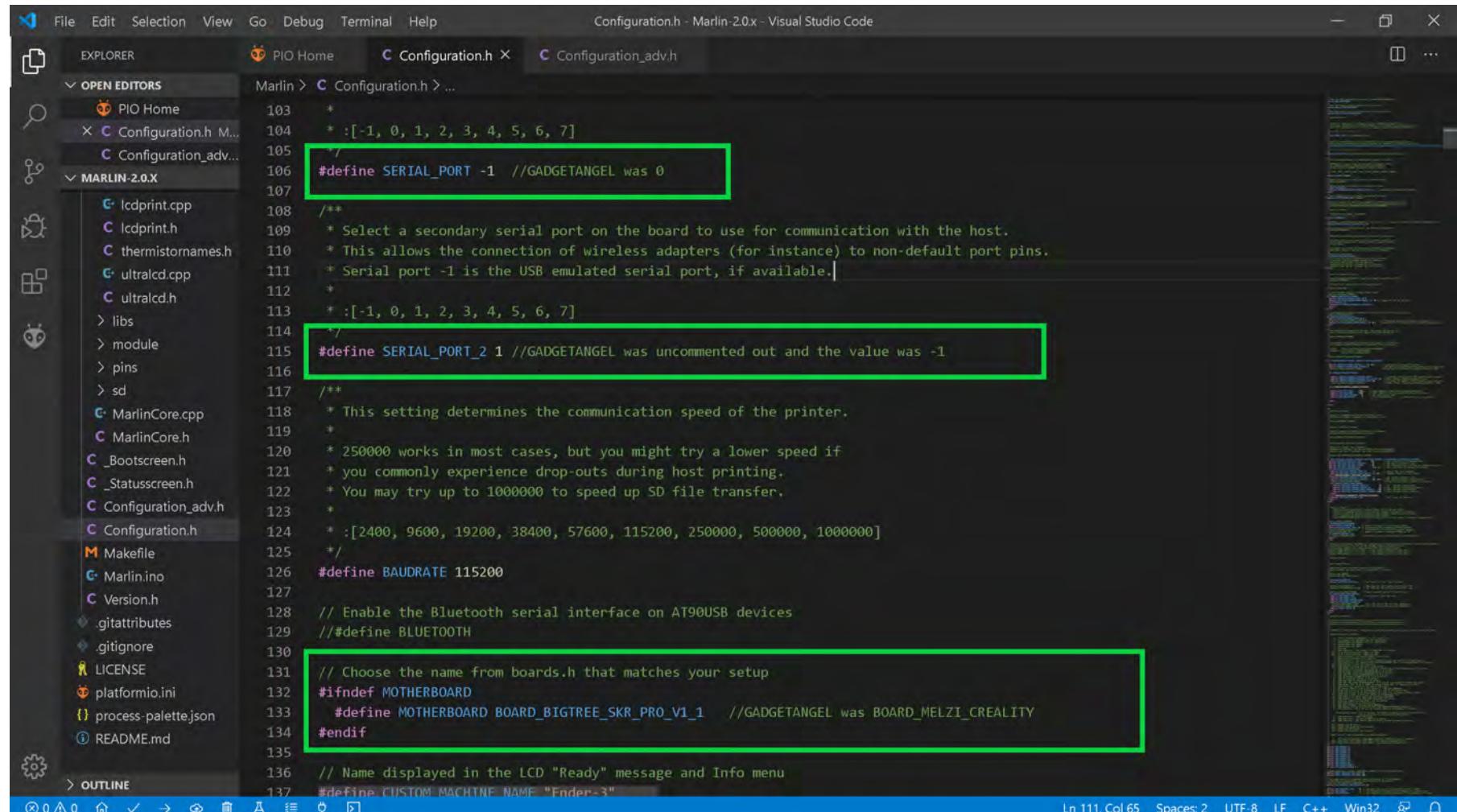
[common]
default_src_filter = +<src/*> -<src/config> -<src/HAL> +<src/HAL/shared>
extra_scripts = pre:buildroot/share/PlatformIO/scripts/common-cxxflags.py
build_flags = -fmax-errors=5 -g -D_MARLIN_FIRMWARE_ -fmerge-all-constants
lib_deps =
    LiquidCrystal
    TMCStepper@>=0.6.1,<1.0.0
    Adafruit 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
```

- 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.



```

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 ...
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

Ln 111, Col 65  Spaces: 2  UTF-8  LF  C++  Win32  ⚡  ⌂

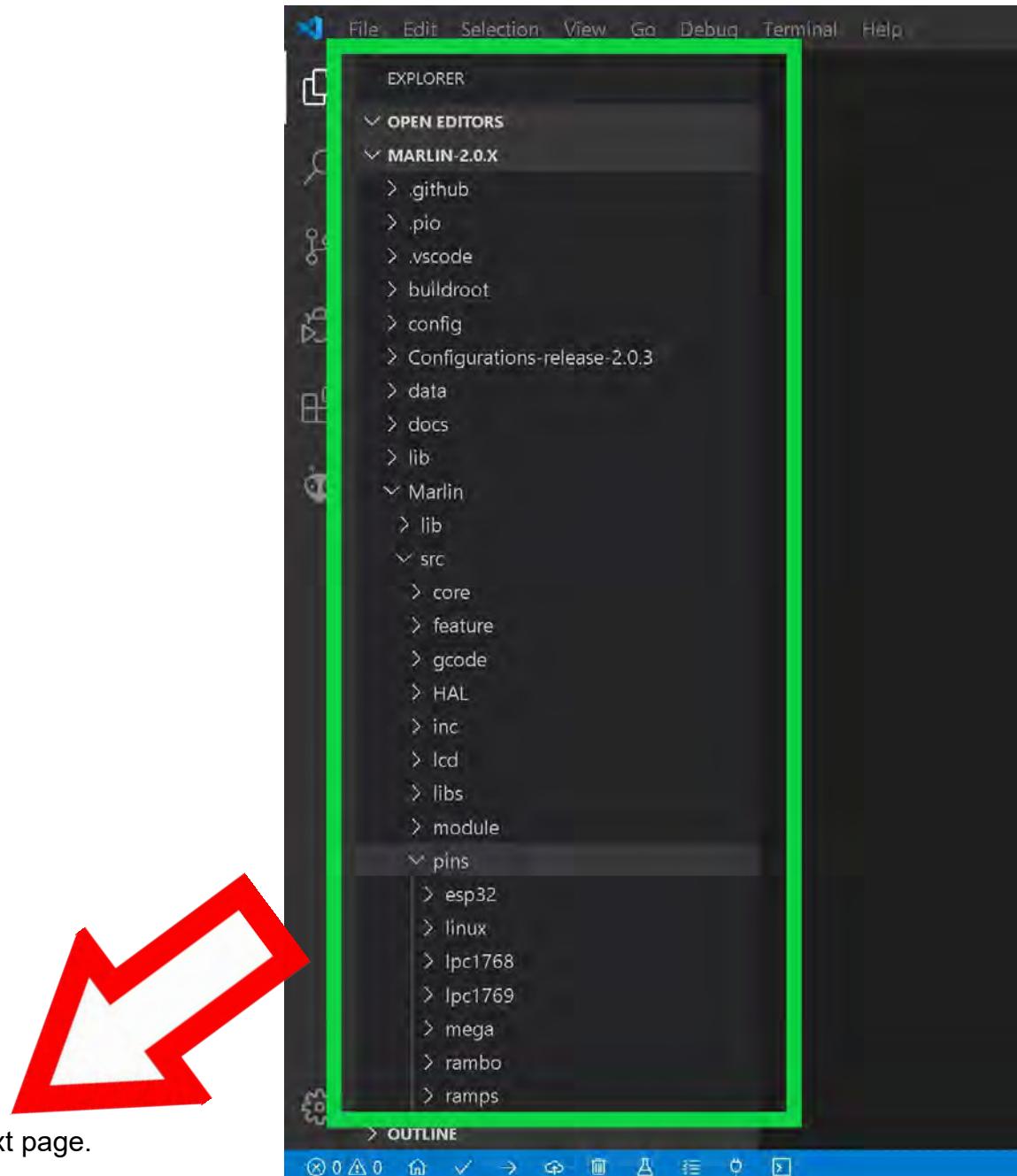
```

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.

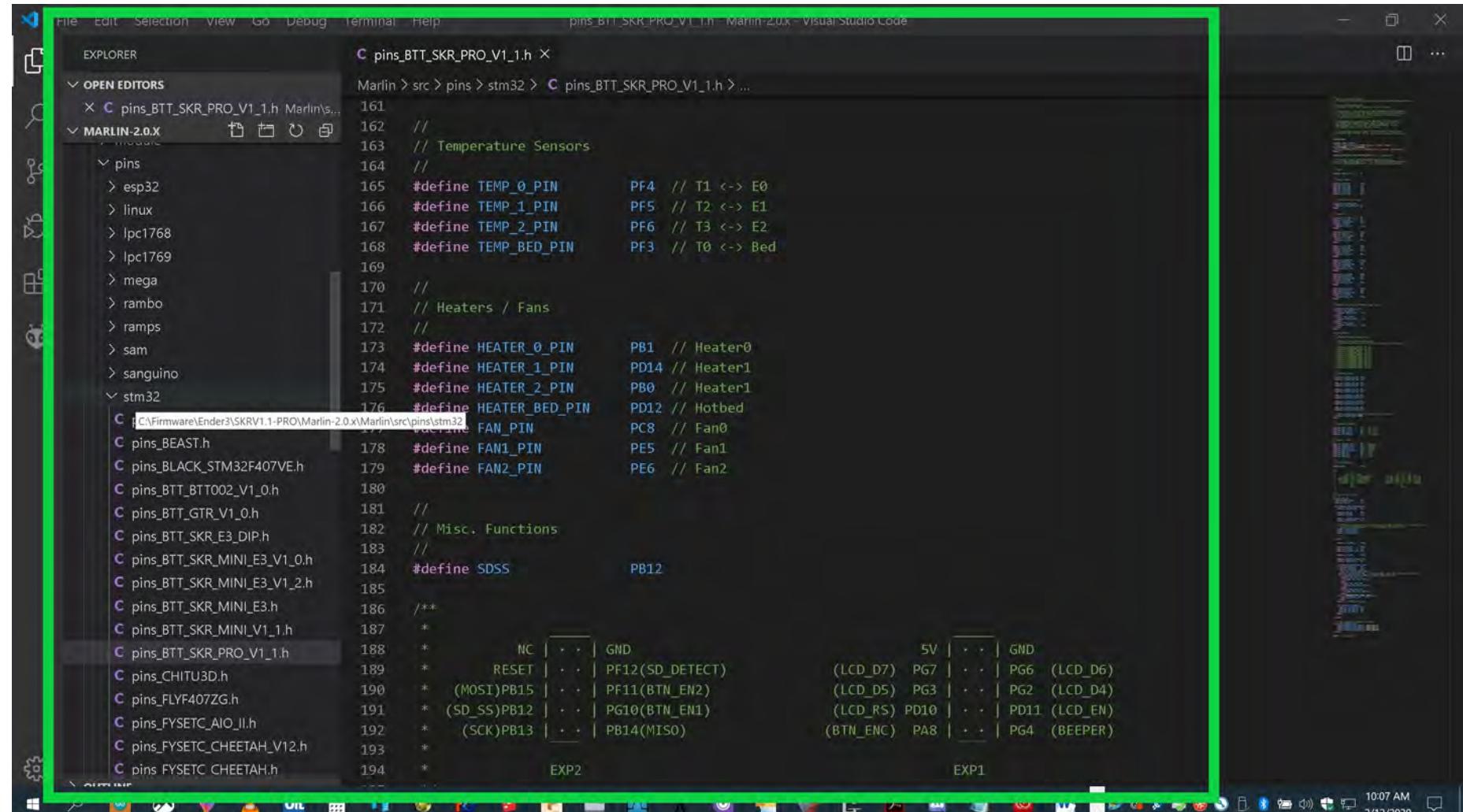
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

- 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 part of the Marlin 2.0.x source code. A green box highlights the section of the code where pins are mapped to specific pins on the STM32 board. The highlighted code includes definitions for temperature sensors, heaters/fans, and various digital 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.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
  pins_BTT.h
  pins_BEAST.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)
189 *      (MOSI)PB15 | + | PF11(BTN_EN2)
190 *      (SD_SS)PB12 | + | PG10(BTN_EN1)
191 *      (SCK)PB13 | + | PB14(MISO)
192 // 
193 // 
194 EXP2 EXP1
  5V | + | GND
(LCD_D7) PG7 | + | PG6 (LCD_D6)
(LCD_D5) PG3 | + | PG2 (LCD_D4)
(LCD_RS) PD10 | + | PD11 (LCD_EN)
(BTN_ENC) PA8 | + | PG4 (BEEPER)

```



The screenshot shows the Visual Studio Code interface with the file `pins_BTT_SKR_PRO_V1_1.h` open. The code is part of the Marlin 2.0.x source code. A green box highlights the section of the code where pins are mapped to specific pins on the Sanguino board. The highlighted code includes definitions for various pins, including heater and fan pins.

```

  > Sanguino
  > STM32
  > pins_BTT.h
  > pins_BEAST.h
  > pins_BTT.h
  > pins_BEAST.h

175 #define HEATER_2_PIN
176 #define HEATER_BED_PIN
177 #define FAN_PIN
178 #define FAN1_PIN

```

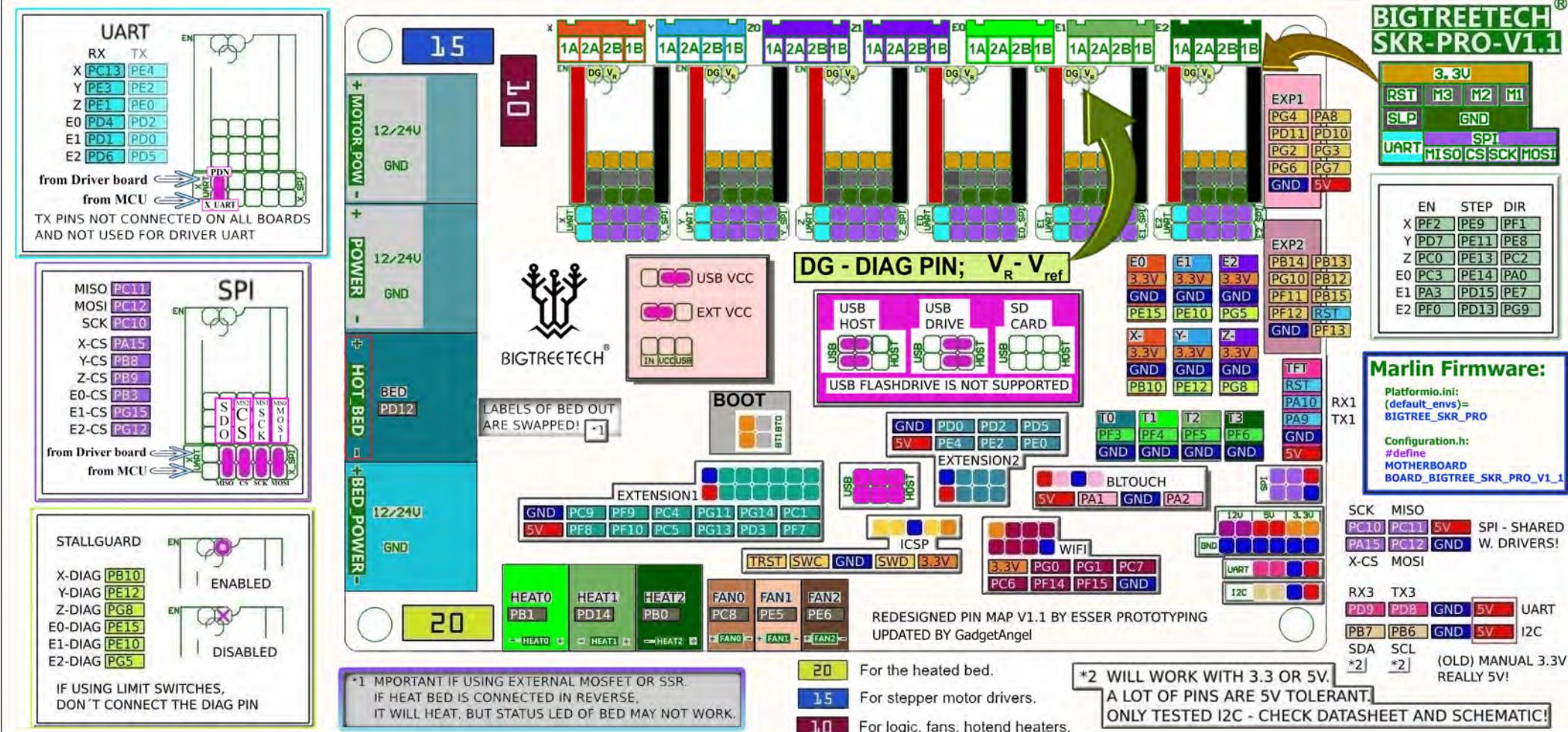
- Go to the next page.

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 functional picture of the SKR PRO V1.1 Color Wiring Diagram.

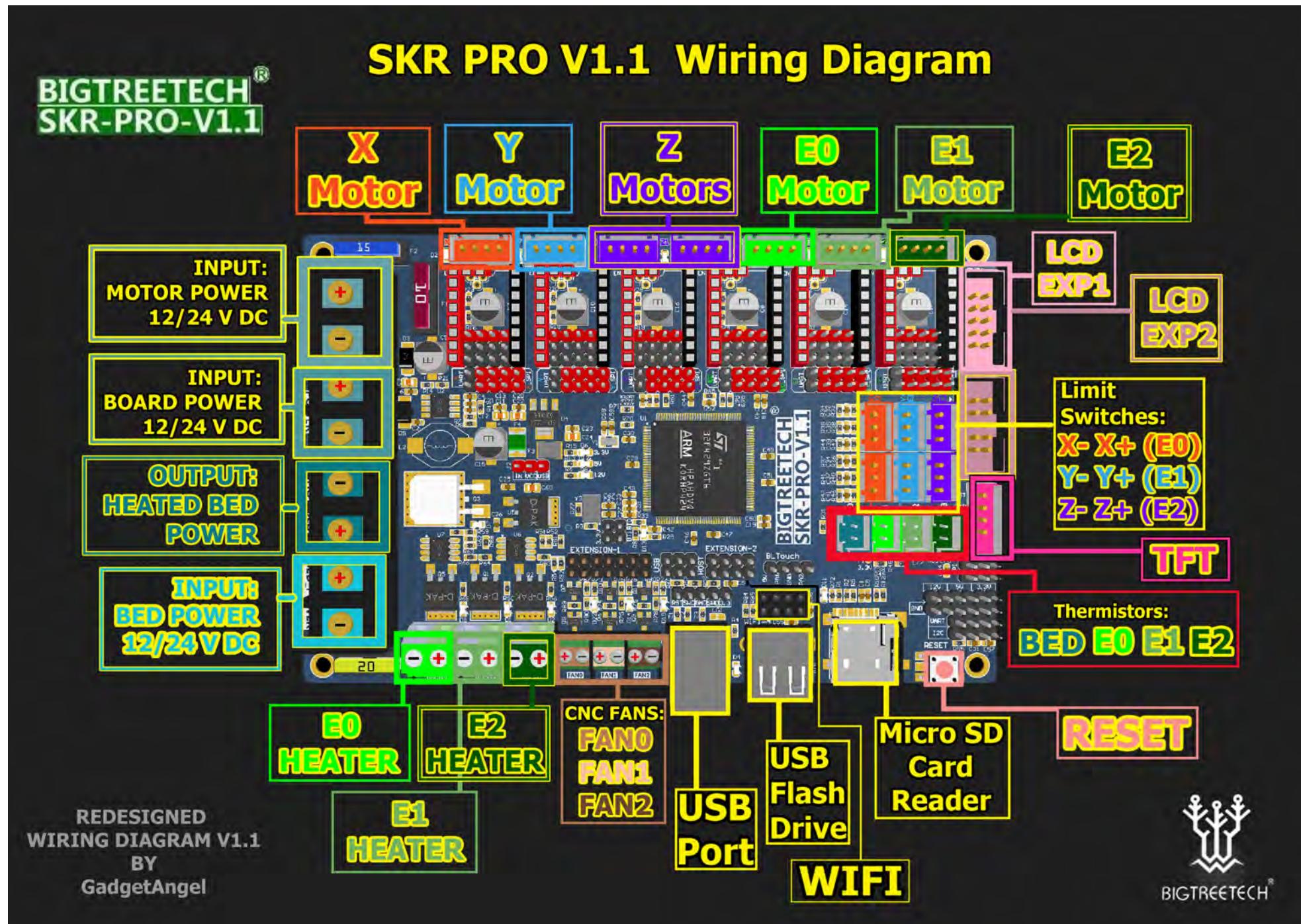
3



- 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 Color 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

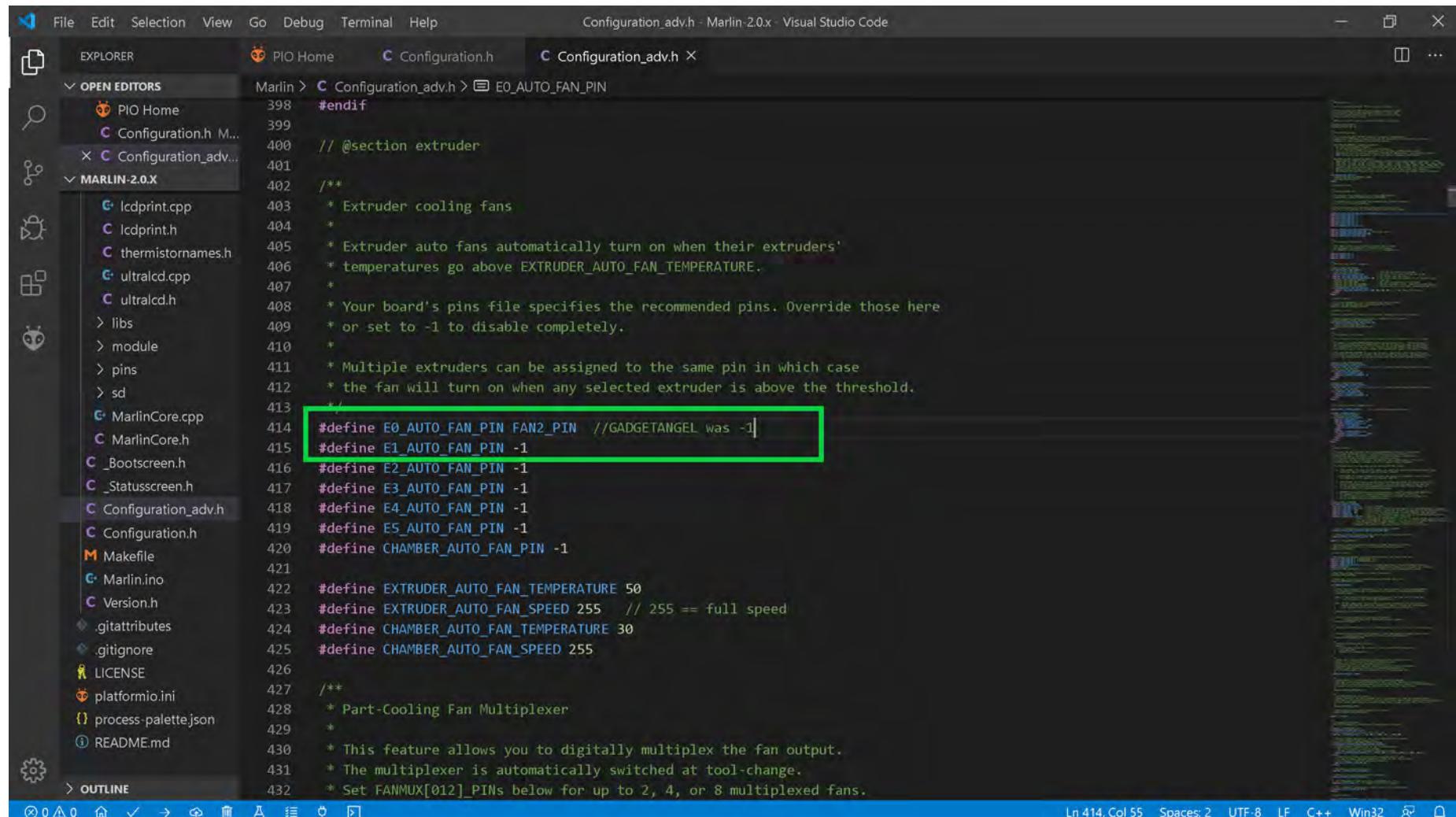


- Go to the next page.

APPENDIX C

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



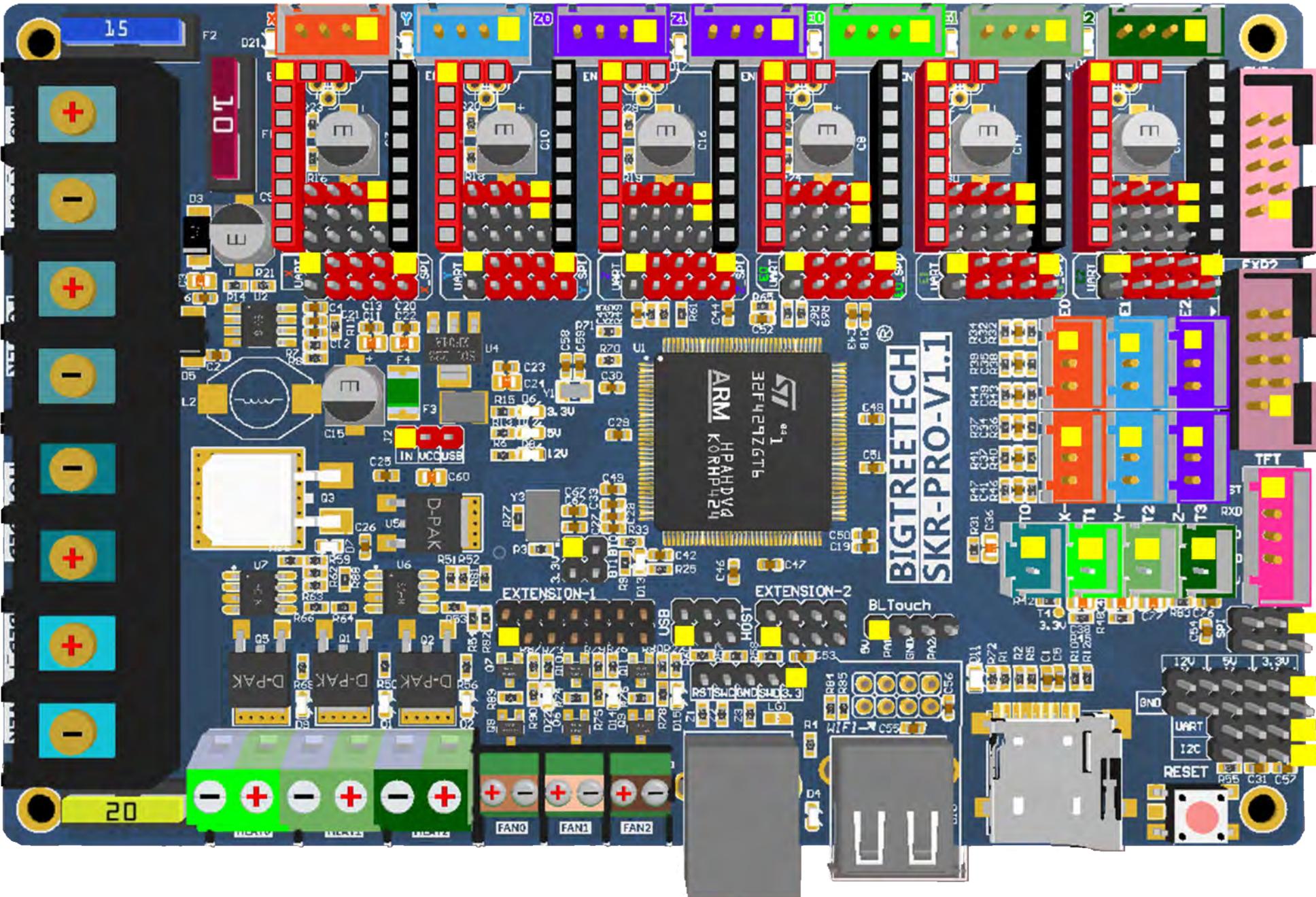
Screenshot of Visual Studio Code showing the Configuration_adv.h file. The line `#define E0_AUTO_FAN_PIN FAN2_PIN` is highlighted with a green box.

```

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[0|1]_PINs below for up to 2, 4, or 8 multiplexed fans.

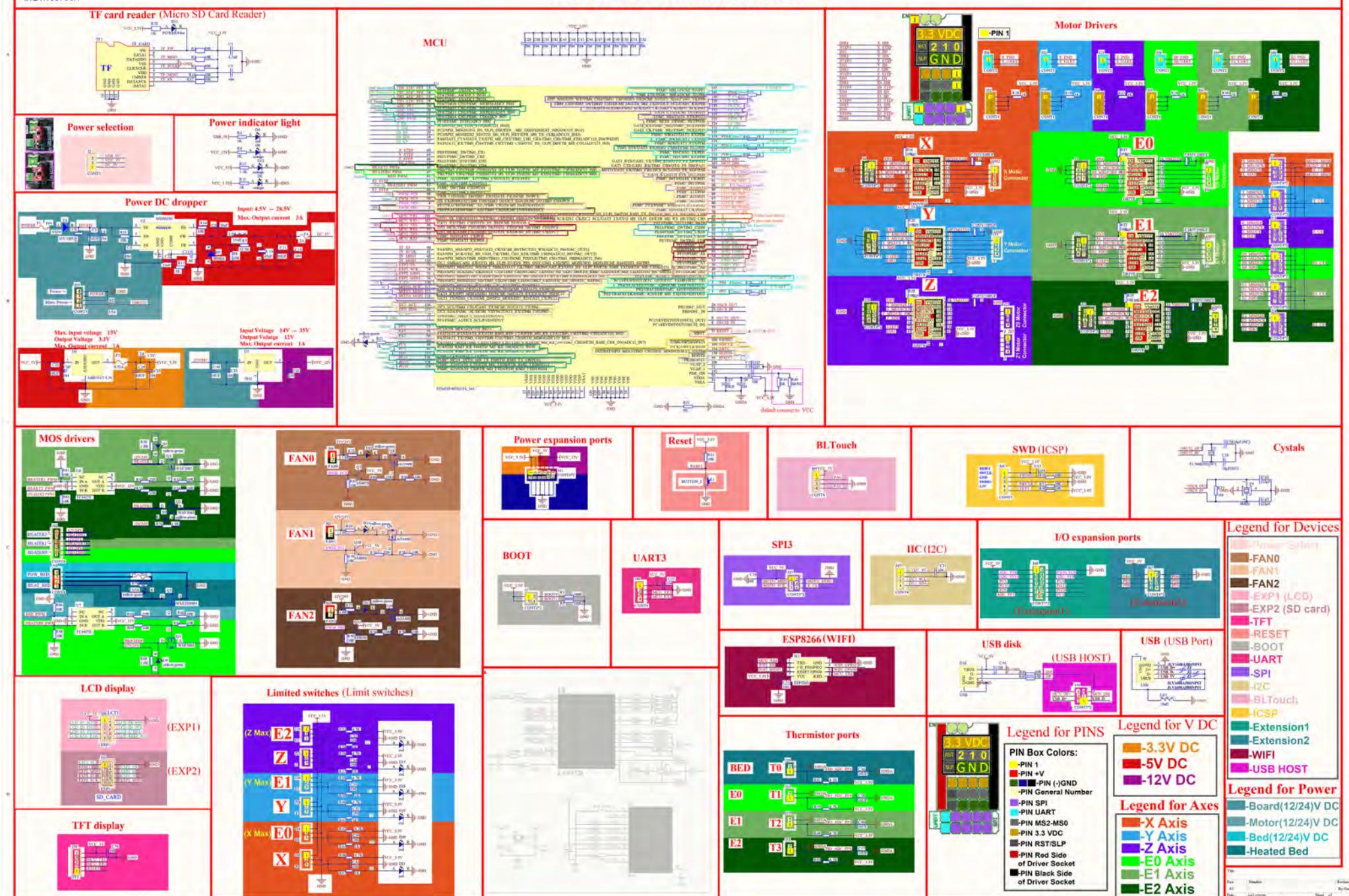
```

- 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.](#)
- [To see my updated PIN 1 diagram for SKR PRO V1.1, go to the next page.](#)

APPENDIX C - Color PIN 1 Diagram

-PIN 1

- To see my updated schematic diagram for SKR PRO V1.1, go to the next page.

APPENDIX C - Color Schematic Diagram
BIGTREETECH SKR-PRO-V1.1
WWW.BIGTREE-TECH.COM


Legends for SKR PRO V1.1 Stepper Driver Socket Representations

Row 1 is for 3.3 VDC PINS

Row 2 is for RST and MS2-MS0 PINS

Row 3 is for SLP and GND PINS

NOTICE: PIN labels are relative to stepper driver chip's carrier board, not the SKR PRO V1.1 controller board. **Double check driver compatibility before use.** Numbered pairs denote **required jumper(s)**. The example below indicates that you must place ONE jumper (8) across row 1 and 2 in MS1 column; ONE jumper (1) across row 2 and 3 in RST/SLP column; ONE jumper (2) across row 2 and 3 in MS2 column; and ONE jumper (3) across row 2 and 3 in MS0 column.

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

Legend for MS0-MS2 PINS:

Legend for Socket:

- - means: No jumper
- 7 7 means: Jumper set
- 7* - means: DuPont jumper to GND

1	3.3V					
2	RST	M3	M2	M1		
3	SLP	GND				
4	SPI	UART	MISO	CS	SCK	MOSI

DG - DIAG PIN
 V_R - V_{ref}

Row:

1

2

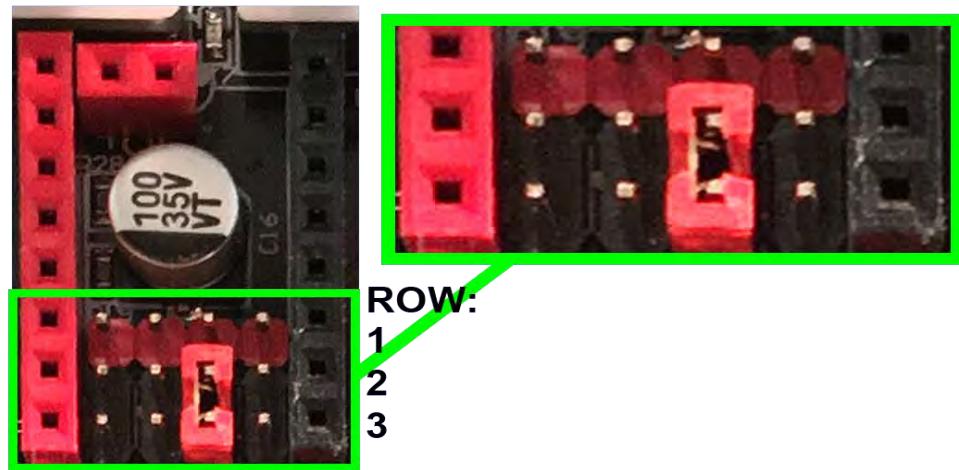
3

X UAQT Spt X

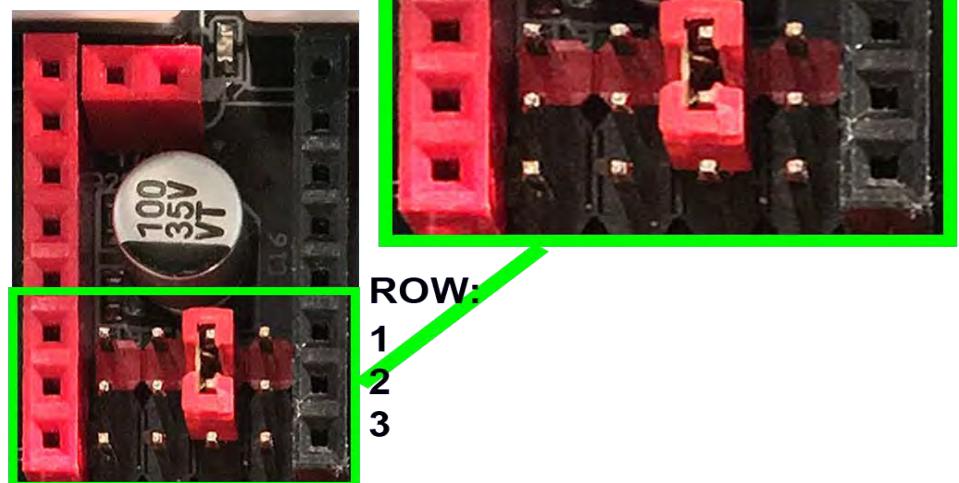
- Go to the next page.

Legends for SKR PRO V1.1 Stepper Driver Socket Representations

LOW:



High:



9 9 ➡ Jumper Set

Low ➡ set Jumper between rows 2 and 3

High ➡ set Jumper between rows 1 and 2

8 8 ➡ Jumper Set

Low ➡ set Jumper between rows 2 and 3

High ➡ set Jumper between rows 1 and 2

7 7 ➡ Jumper Set

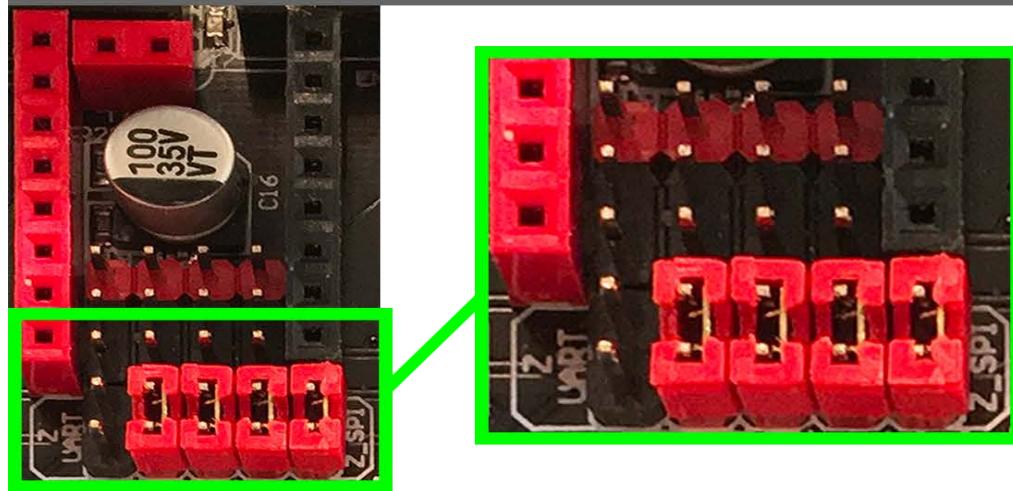
Low ➡ set Jumper between rows 2 and 3

High ➡ set Jumper between rows 1 and 2

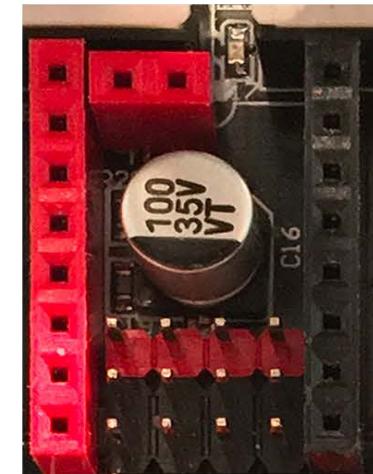
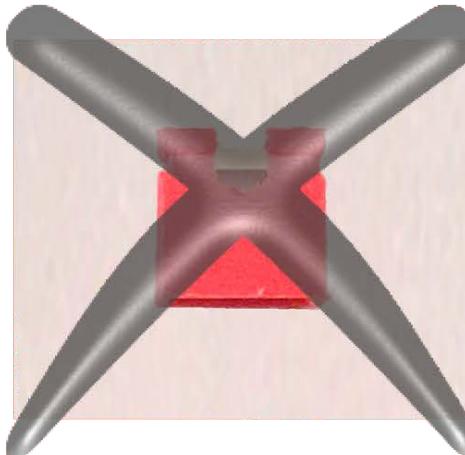
- Go to the next page.

Legends for SKR PRO V1.1 Stepper Driver Socket Representations

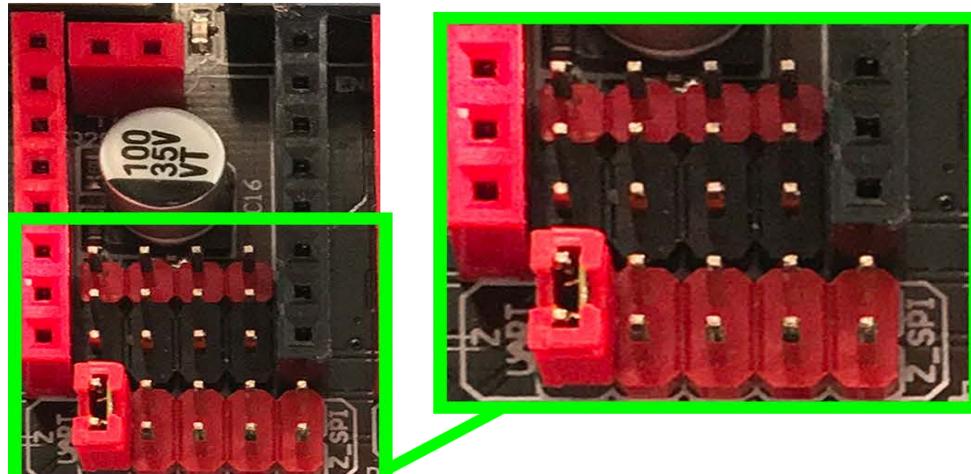
SPIM ➔ Jumper Set



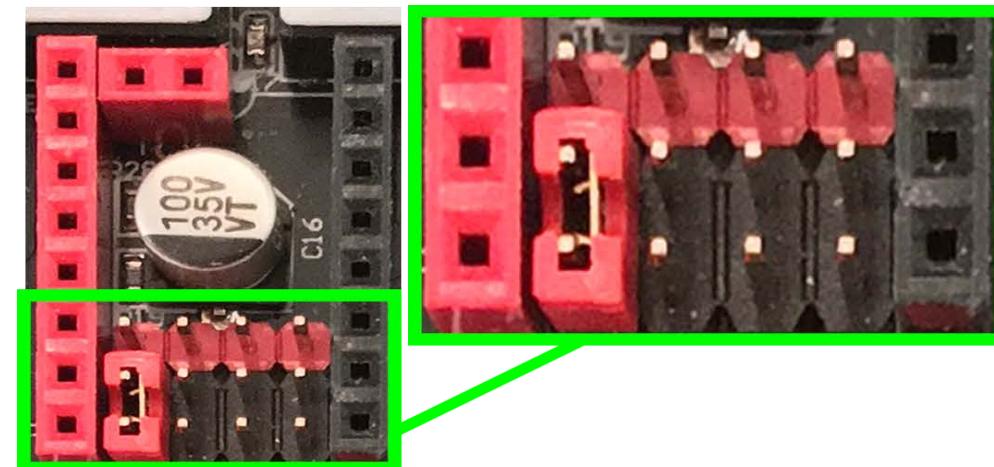
- ➔ No Jumper



U ➔ Jumper Set



D ➔ Jumper Set

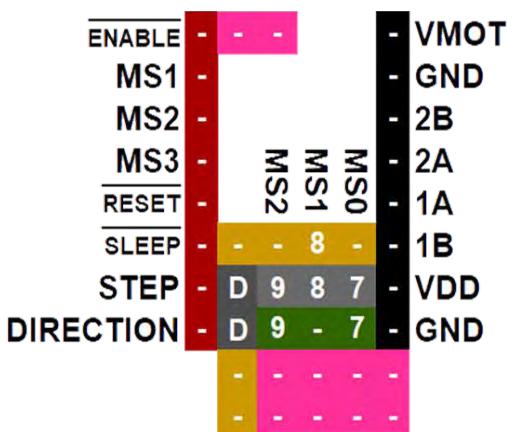


- Go to the next page.

Special Consideration of D Jumper for SKR PRO V1.1 Board

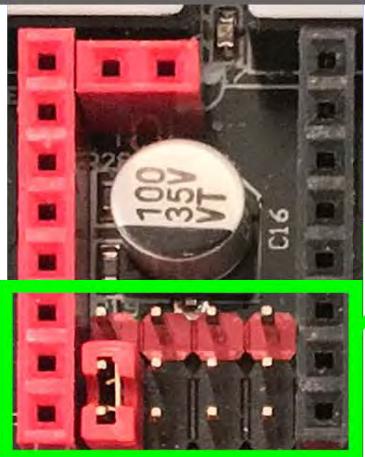
Driver Socket Representation for D Jumper:

1 / 4



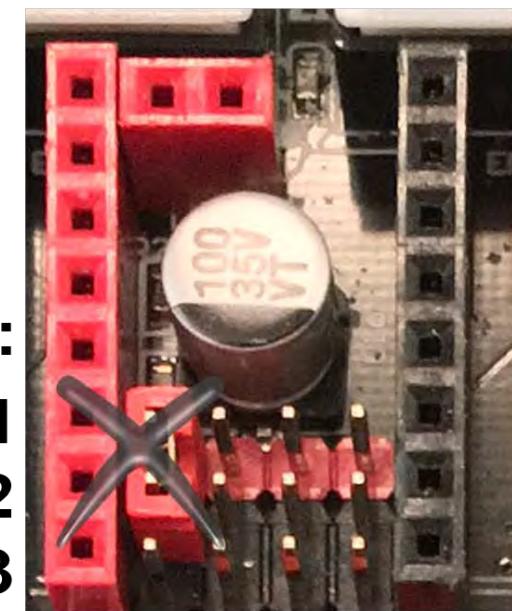
The Right Way to Set the D Jumper:

D ➔ Jumper Set



The Wrong Way to Set the D Jumper:

Example of the **WRONG** way to Set the D jumper (**do not** use row 1 and row 2 pins):



ROW:

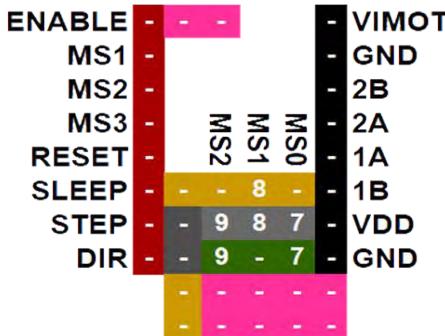
1
2
3

- Go to the next page.

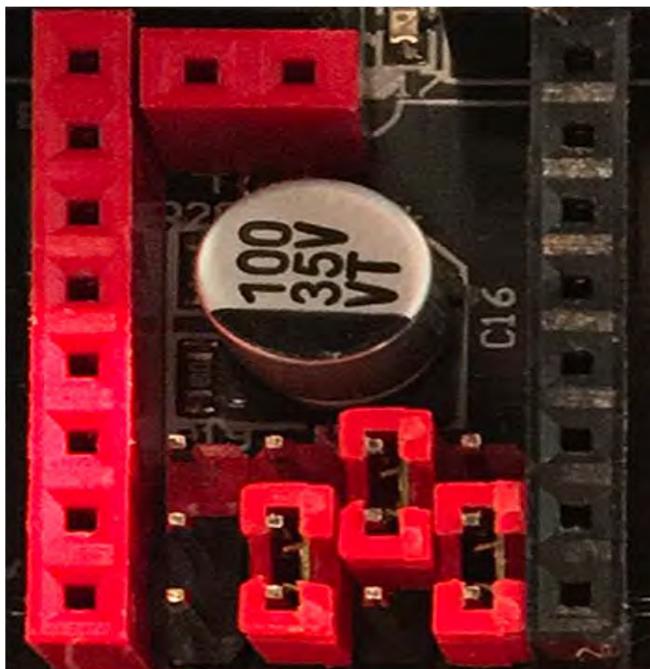
Examples for Stepper Driver Socket Representations

Example 1 (LV8729 Driver Board) for SKR PRO V1.1 Driver Socket Legend:

1 / 4



M M M
S S S
2 1 0



Row:

1
2
3

9 9 → Jumper Set

MS2 PIN:
Low → set Jumper between rows 2 and 3

8 8 → Jumper Set

MS1 PIN:
High → set Jumper between rows 1 and 2

7 7 → Jumper Set

MS0 PIN:
Low → set Jumper between rows 2 and 3

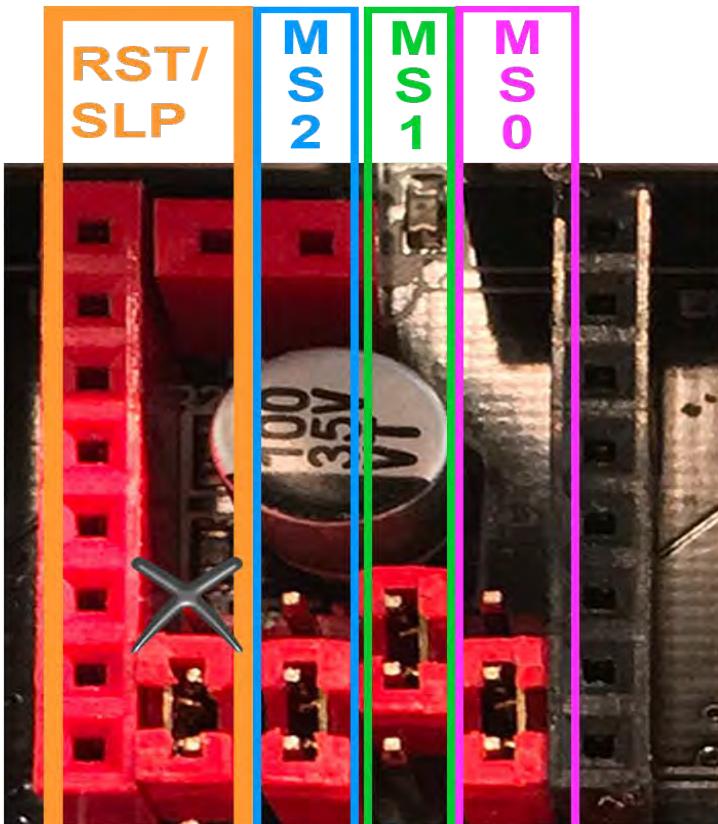
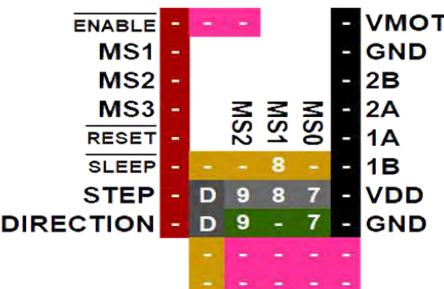
- Go to the next page.

Examples for Stepper Driver Socket Representations

**Example 2 (A4988 Driver Board) for
SKR PRO V1.1 Driver Socket**

Legend:

1 / 4



Row:

1
2
3

- Go to the next page.

**RST → Jumper Set
SLP**

RST & SLP PINS:

RST/SLP → set Jumper between rows 2 and 3

9 9 → Jumper Set

MS2 PIN:

Low → set Jumper between rows 2 and 3

8 8 → Jumper Set

MS1 PIN:

High → set Jumper between rows 1 and 2

7 7 → Jumper Set

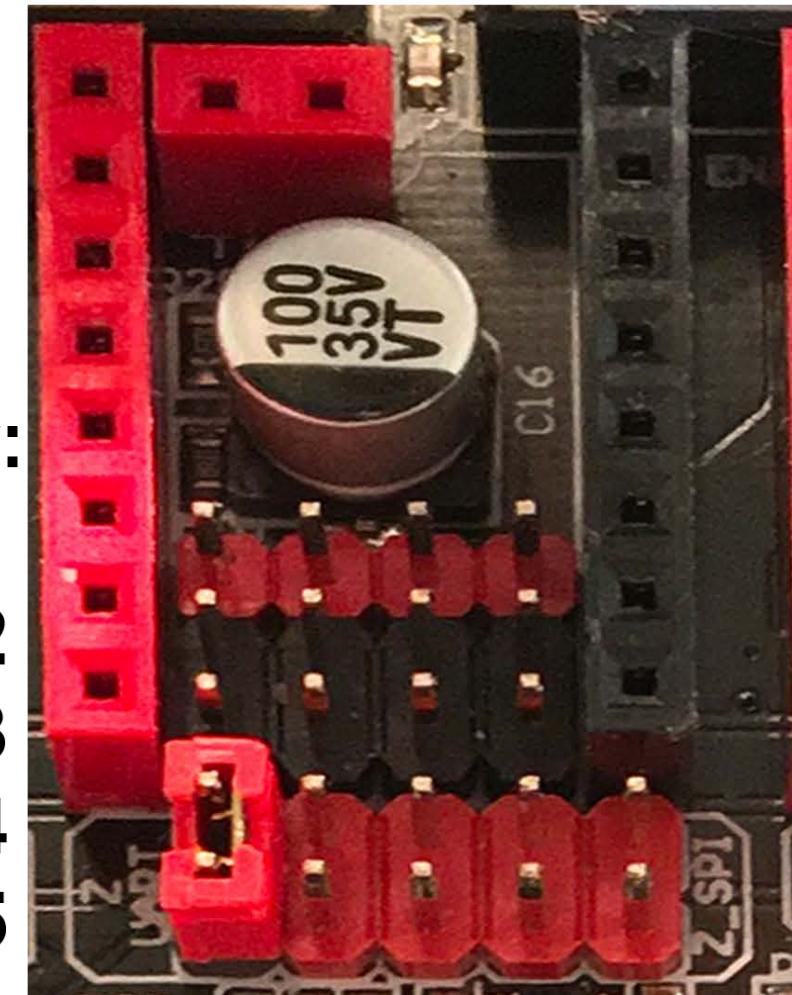
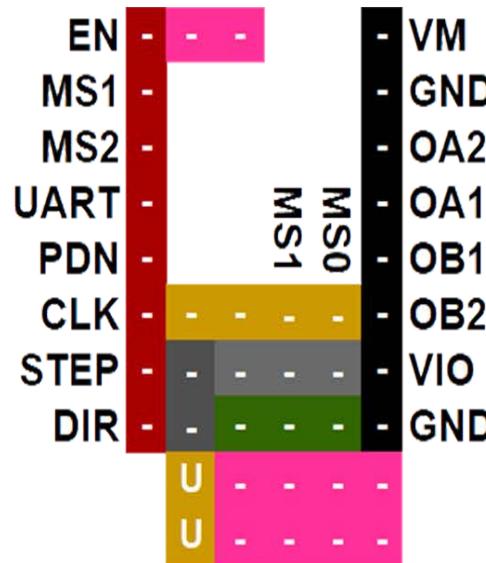
MS0 PIN:

Low → set Jumper between rows 2 and 3

Examples for Stepper Driver Socket Representations

Example 3 (UART Driver Board) for SKR PRO V1.1 Driver Socket Legend:

UART



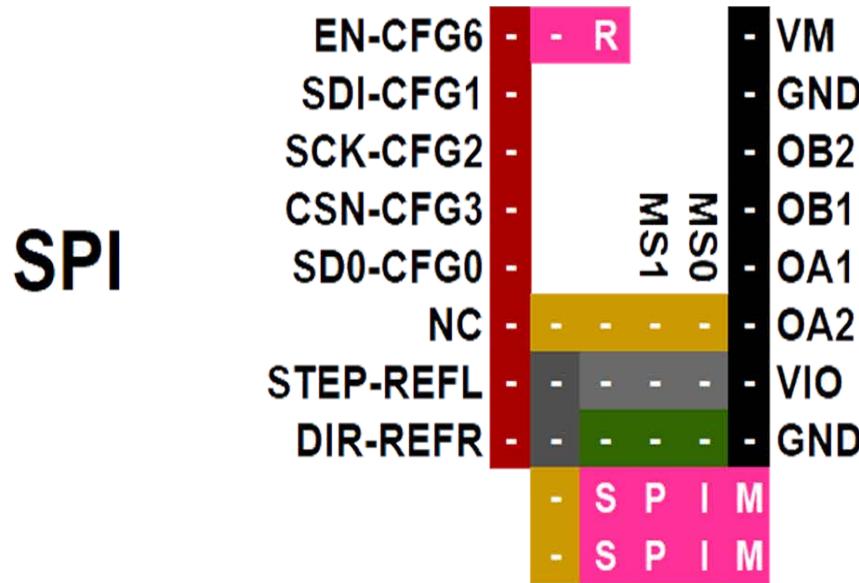
U Jumper Set

U set Jumper
between rows
4 and 5

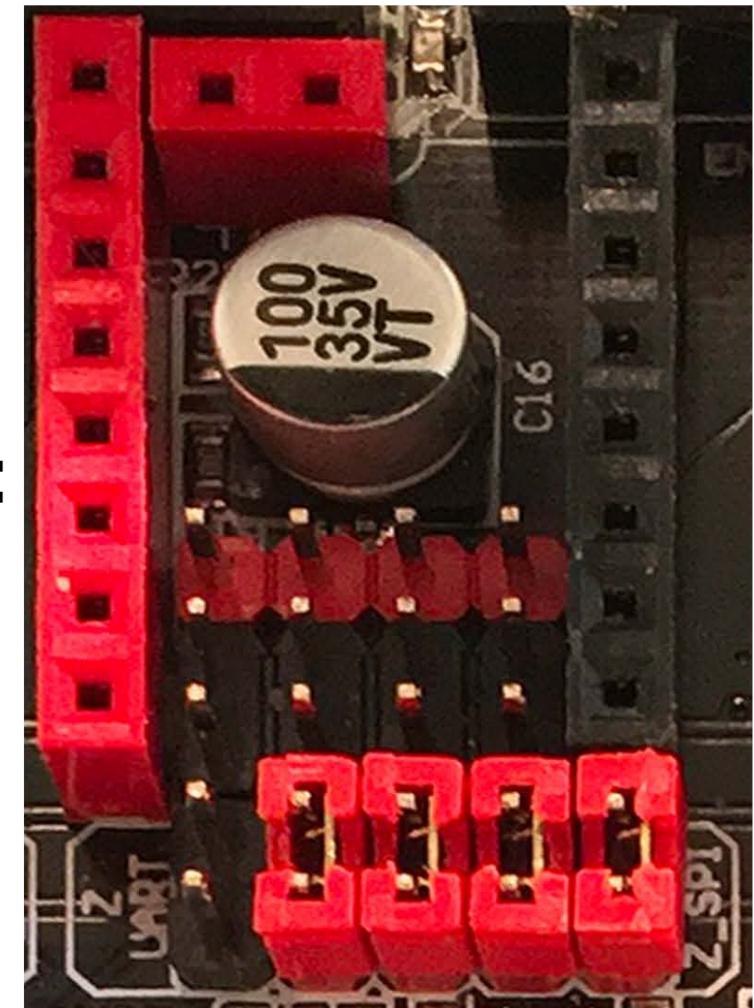
- Go to the next page.

Examples for Stepper Driver Socket Representations

Example 4 (SPI Driver Board) for SKR PRO V1.1 Driver Socket Legend:



RST/
SLP M M M
 S S S
 2 1 0

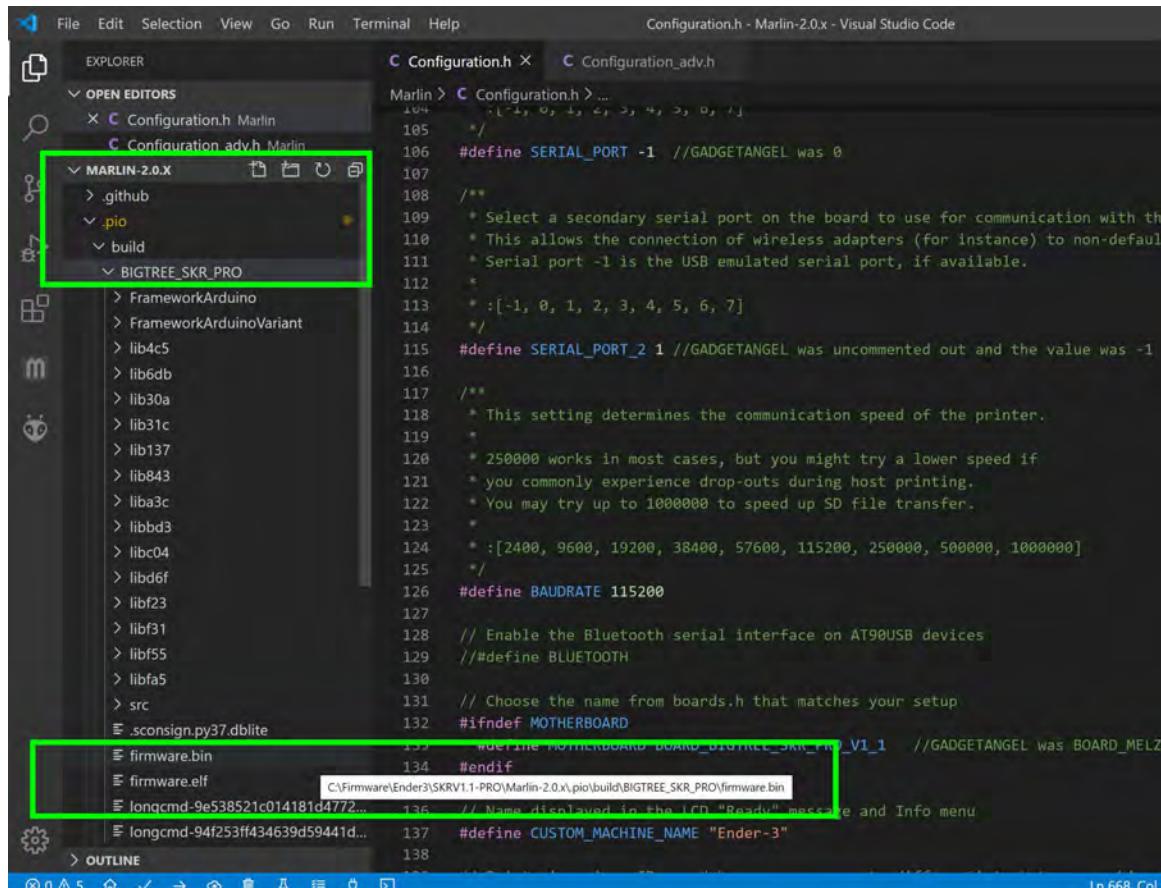


SPIM → Jumper Set

SPIM → set Jumpers between rows 4 and 5

APPENDIX E**Location Of "firmware.bin" File from the Marlin Compilation for SKR PRO V1.1 Board**

- 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.



The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar, which displays the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X'. A green box highlights the 'pio/build/BIGTREE_SKR_PRO' folder. Inside this folder, 'firmware.bin' and 'firmware.elf' files are listed, also highlighted with a green box. The main editor area shows the 'Configuration.h' file with the following code snippet:

```

104
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
110 * This allows the connection of wireless adapters (for instance) to non-default
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
134 #endif
135
136 // Name displayed in the LCD "Ready" message and Info menu
137 #define CUSTOM_MACHINE_NAME "Ender-3"
138

```

Below the editor is the 'OUTLINE' view, which lists the files 'sconsian.py37.dblite', 'firmware.bin', 'firmware.elf', 'longcmd-9e538521c014181d4772...', and 'longcmd-94f253ff434639d59441d...'. The 'firmware.elf' entry is highlighted with a green box, and its tooltip shows the full path: C:\Firmware\Ender3\SKR V1.1-PRO\Marlin-2.0.x\pio\build\BIGTREE_SKR_PRO\firmware.bin.

- Go to the next page.

Location Of "firmware.bin" File from the Marlin Compilation for SKR PRO V1.1 Board

```

    .sconsian.py37.dblite          132  #ifndef MOTHERBOARD
    firmware.bin                   133  #define MOTHERBOARD BOARD_BIGTREE_SKR_PRO_V1_1 //GADGETANGEL
    firmware.elf                   134  #endif
    longcmd-9e538521c014181d4772... 136  // Name displayed in the LCD "Ready" message and Info menu
    longcmd-9e538521c014181d4772... 137  #define CUSTOM_MACHINE_NAME "Ender-3"
    138
  
```

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

EXPLORER pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h

OPEN EDITORS Marlin > Configuration.h > X_DRIVER_TYPE

MARLIN-2.0.X

.pio build

BIGTREE_SKR_PRO

- > FrameworkArduino
- > FrameworkArduinoVariant
- > lib2ce
- > lib4e1
- > lib6bc
- > lib6db
- > lib30a
- > lib31c
- > lib65f
- > lib74f
- > lib93f
- > lib109
- > lib533
- > lib694
- > lib925
- > libc04
- > src
- > .sconsian.py37.dblite
- firmware.bin**
- > firmware.elf
- > longcmd-9e538521c014181d4772...
- > longcmd-d2cd97a14e80d0c95c59...

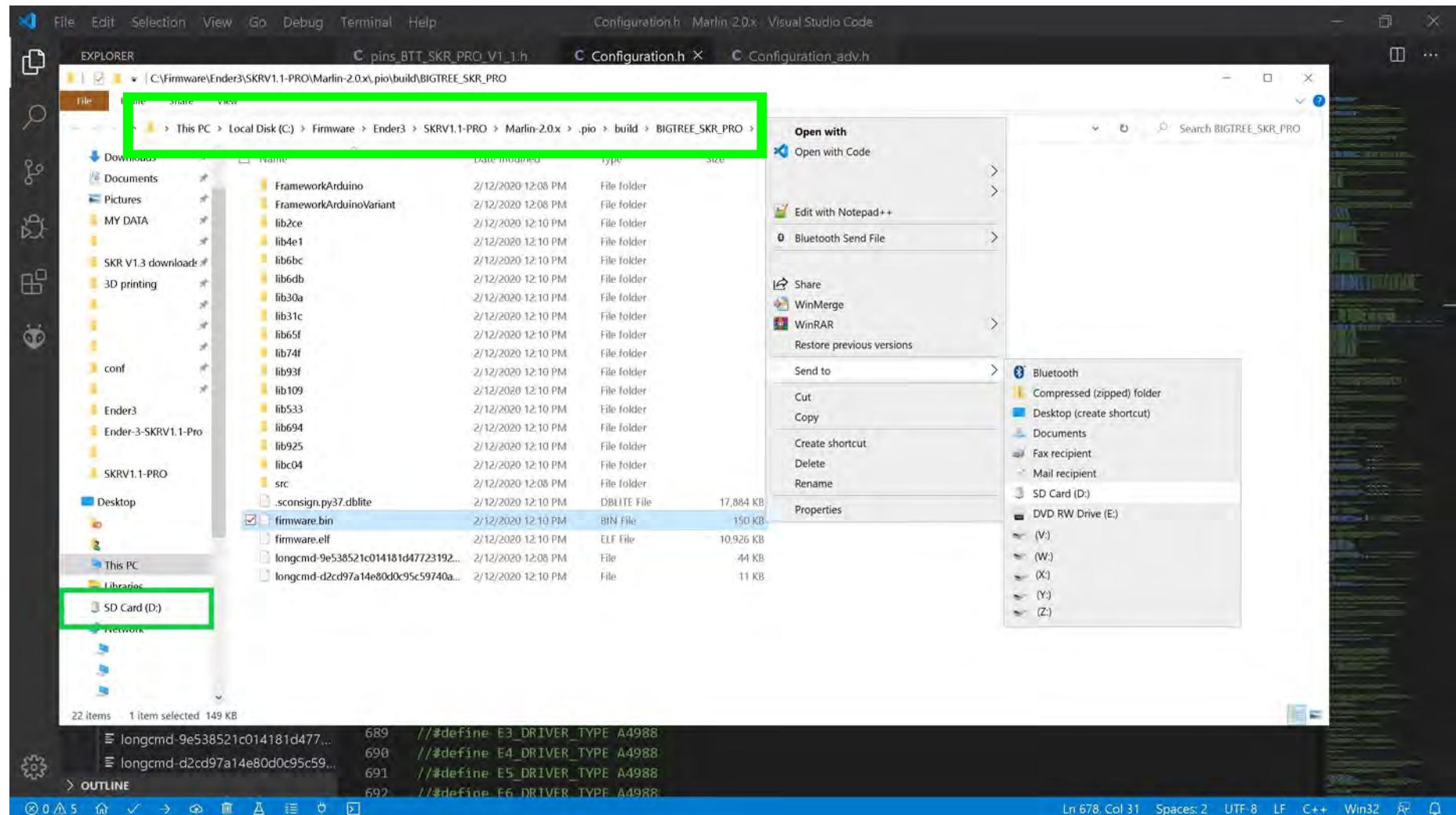
Open to the Side Ctrl+Enter Reveal in File Explorer Shift+Alt+R Open in terminal Select for Compare Cut Ctrl+X Copy Ctrl+C Copy Path Shift+Alt+C Copy Relative Path Ctrl+K Ctrl+Shift+C Rename F2 Delete Delete

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

- Go to the next page.

Location Of "firmware.bin" File from the Marlin Compilation for SKR PRO V1.1 Board

- 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.

APPENDIX F

Links to Reference Material

Marlin Firmware Documentation

- <https://marlinfw.org/docs/configuration/configuration.html>
- <https://marlinfw.org/docs/basics/introduction.html>
- <https://marlinfw.org/meta/download/>
- <https://marlinfw.org/docs/basics/install.html>
- <https://marlinfw.org/tools/u8glib/converter.html>
- https://marlinfw.org/tools/lin_advance/k-factor.html
- <https://marlinfw.org/meta/configuration/>
- <https://marlinfw.org/meta/development/>
- <https://marlinfw.org/meta/features/>
- <https://marlinfw.org/meta/gcode/>
- <https://marlinfw.org/meta/hardware/>
- <https://marlinfw.org/docs/basics/troubleshooting.html>

Information on Stepper Motor Drivers and Micro-stepping

- <https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping>
- Go to the next page.

Links to Reference Material

POLOLU A4988 and BIQU A4988

- <https://www.youtube.com/watch?v=H41hIXdB6js&t=43s>
- <https://www.pololu.com/product/1182>

DRV8825

- <https://www.youtube.com/watch?v=H41hIXdB6js&t=667s>
- <https://www.pololu.com/product/2133>

- Go to the next page.

Links to Reference Material

BIQU LV8729, FYSETC LV8729, LERDGE LV8729, and MKS LV8729

- <https://www.youtube.com/watch?v=IDljLaXWq4E>

BIQU LV8729

- <https://www.biqu.equipment/collections/stepper-motor-board/products/biqu-direct-sell-3d-printer-part-lv8729-stepper-motor-driver-ultra-quiet-driver-lv8729-driver-support-6v-36v-with-heatsink>
- Go to the next page.

Links to Reference Material

FYSETC LV8729

- https://wiki.fysetc.com/LV8729_V1.0/

LERDGE LV8729

- <https://es.aliexpress.com/item/32816789352.html>

- Go to the next page.

Links to Reference Material

MKS LV8729

- <http://3dmodularsystems.com/en/stepper-motor-drivers/470-mks-lv8729-ultra-quiet-stepper-motor-driver-15a.html>

FYSETC S6128 V1.1

- https://wiki.fysetc.com/S6128_V1.1/

- Go to the next page.

Links to Reference Material

FYSETC ST820

- https://wiki.fysetc.com/ST820_V1.0/

BIQU ST820

- https://bigtreetech.en.alibaba.com/product/60873031205-806048513/BIGTREETECH_ST820_V1_0_Step_Stepper_Motor_Driver_Microstepping_Kit_MKS_GEN_V1_4_MKS_GEN_L_SKR_V1_1_Ramps_1_4.html
 - <https://www.youtube.com/watch?v=HNciZ31GAro>
 - <https://github.com/bigtreeTech/BIGTREETECH-ST820-V1.0>
-
- Go to the next page.

Links to Reference Material

POLOLU ST820 (STSPIN820)

- <https://www.pololu.com/product/2878>

POLOLU MP6500

- <https://www.youtube.com/watch?v=1oKByOCngEI>
- https://www.pololu.com/file/0J1447/MP6500_r1.0.pdf
- <https://www.pololu.com/product/2966>

- Go to the next page.

Links to Reference Material

POLOLU TB67S249FTG

- <https://toshiba.semicon-storage.com/us/semiconductor/product/motor-driver-ics/stepping-motor-driver-ics/detail.TB67S249FTG.html>
- https://www.pololu.com/file/0J1523/TB67S249FTG_datasheet_en_20170818.pdf
- <https://www.pololu.com/product/3096>

BIQU S109 and FYSETC S109

- <https://toshiba.semicon-storage.com/us/semiconductor/product/motor-driver-ics/stepping-motor-driver-ics/detail.TB67S109AFTG.html>

- Go to the next page.

Links to Reference Material

BIQU S109

- <https://www.biqu.equipment/products/bigtreeetech-s109-v1-0-stepper-motor-driver>

FYSETC S109

- https://wiki.fysetc.com/S109_V1.1/

- Go to the next page.

Links to Reference Material

Marlin Firmware Documentation Specific to TMC Drivers

- https://marlinfw.org/docs/hardware/tmc_drivers.html

Information Common to All TMC Drivers

- <https://github.com/watterott/SilentStepStick>
- <https://www.trinamic.com/support/eval-kits/details/silentstepstick/>
- <https://www.instructables.com/id/Install-and-configure-SilentStepStick-in-RAMPS-TMC/>

- Go to the next page.

Links to Reference Material

BIQU TMC2100 and MKS TMC2100

- <https://learn.watterott.com/silentstepstick/pinconfig/tmc2100/>
- <https://reprap.org/wiki/TMC2100>
- <https://www.youtube.com/watch?v=H41hlXdB6js&t=869s>
- <https://wiki.fysetc.com/TMC2100/>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2100_datasheet_Rev1.10.pdf
- <https://www.biqu.equipment/products/biqu-bigtreeTech-tmc2100-2208-2130-v1-0-stepstick-mks-stepper-motor-driver-ultra-silent>

- Go to the next page.

Links to Reference Material

BIQU TMC2130

- <https://www.youtube.com/watch?v=Jh8iwqc1TP0>
- <https://www.youtube.com/watch?v=OUadiW5QLBE&t=34s>
- <https://learn.watterott.com/silentstepstick/pinconfig/tmc2130/>
- <https://github.com/bigtreetech/BIGTREETECH-TMC2130-V3.0/blob/master/TMC2130-V3.0RM.pdf>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2130_datasheet_Rev1.11.pdf
- <https://www.youtube.com/watch?v=HoKq7XUaGxM>
- <https://www.youtube.com/watch?v=OUadiW5QLBE&t=555s>

Information Common to BIQUE TMC2208 V3.0 and FYSETC TMC2208 V1.2

- <https://www.instructables.com/id/TMC2208-UART-on-BigTreeTechBIQUE-SKR-V11-and-V13-Co/>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC220x_TMC2224_datasheet_Rev1.09.pdf
- <https://learn.watterott.com/silentstepstick/pinconfig/tmc2208/>
- <https://www.youtube.com/watch?v=TubpTMFkbVs>
- <https://www.youtube.com/watch?v=7VHwcEroHPk&t=2s>
- <https://www.youtube.com/watch?v=7VHwcEroHPk&t=419s>

- Go to the next page.

Links to Reference Material

Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2 (Continued)

- <https://www.youtube.com/watch?v=9V8iEEAB15g&t=1s>
- <https://www.youtube.com/watch?v=FEgirEyEUbo>
- <https://www.youtube.com/watch?v=SyMNwmO5Nxo>
- <https://www.youtube.com/watch?v=7VHwcEroHPk&t=510s>

BIQU TMC2208 V3.0

- <https://github.com/bigtreeetech/BIGTREETECH-TMC2208-V3.0/blob/master/TMC2208-V3.0%20manual.pdf>

FYSETC TMC2208 V1.2

- <https://wiki.fysetc.com/TMC2208/>

- Go to the next page.

Links to Reference Material

Information Common to TMC2208 and BIQU TMC2225

- <https://www.instructables.com/id/TMC2208-UART-on-BigTreeTechBIQU-SKR-V11-and-V13-Co/>

BIQU TMC2225 V1.0

- <https://github.com/bigtreeTech/BIGTREETECH-TMC2225-V1.0>
 - <https://github.com/bigtreeTech/BIGTREETECH-TMC2225-V1.0/blob/master/TMC2225%20V1.0%20manual.pdf>
 - <http://img.ekeic.com/ekeic/9981564407175.pdf>
-
- Go to the next page.

Links to Reference Material

BIQU TMC2209 V1.2

- https://www.youtube.com/watch?v=dOJbSrWVu_Q
- <https://www.youtube.com/watch?v=H5ea2iyOscg&t=1s>
- <https://learn.watterott.com/silentstepstick/pinconfig/tmc2209/>
- <https://github.com/bigtreeTech/BIGTREETECH-TMC2209-V1.2/blob/master/manual/TMC2209-V1.2-manual.pdf>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2209_Datasheet_V104.pdf

BIQU TMC5160 V1.2

- https://www.youtube.com/watch?v=dOJbSrWVu_Q
- <https://learn.watterott.com/silentstepstick/pinconfig/tmc5160/>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC5160A_Datasheet_Rev1.13.pdf
- <https://github.com/bigtreeTech/BIGTREETECH-TMC5160-V1.0/blob/master/manual/BIGTREETECH%20TMC5160-V1.0%20manual.pdf>
- <https://www.biqu.equipment/collections/stepper-motor-board/products/bigtreeTech-tmc5160-v1-0-driver-spi-mode-silent-high-precision-stepstick-stepper-motor-driver-with-heatsink-for-skr-v1-3-gen-v1-4-reprap>

- Go to the next page.

Links to Reference Material

BIQU TMC5161 V1.0

- <https://github.com/bigtreeTech/BIGTREETECH-TMC5161-V1.0/blob/master/TMC5161%20v1.0%20manual.pdf>
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC5161A-datasheet_Rev1.05.pdf

SKR PRO V1.1 Board

- <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/Schematic>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/Virtual%20COM%20Port%20driver>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/firmware>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/manual>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/models>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/tree/master/picture>
 - <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/issues>
- Go to the next page.

Links to Reference Material

Facebook Groups

- https://www.facebook.com/groups/2264108593625228/?hc_location=ufi
- <https://www.facebook.com/groups/505736576548648/?ref=group Browse>
- <https://www.facebook.com/groups/755135144882809/?ref=group Browse>
- <https://www.facebook.com/groups/1049718498464482/?ref=group Browse>
- <https://www.facebook.com/groups/928738533837774/?ref=group Browse>
- <https://www.facebook.com/groups/197476557529090/?ref=group Browse>
- <https://www.facebook.com/groups/2264108593625228/?ref=group Browse>

- Go to the next page.

Links to Reference Material

Miscellaneous Information

- <https://youtu.be/8wk1elugv5A>
- https://reprap.org/wiki/Stepper_wiring
- <https://learn.watterott.com/silentstepstick/configurator/>
- <https://wiki.fysetc.com/TMC2208/#to-run-the-program>
- <https://marlinfw.org/meta/gcode/>
- <https://blog.prusaprinters.org/calculator/>
- <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers>
- <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>
- https://marlinfw.org/docs/basics/install_platformio.html
- <https://marlinfw.org/meta/download/>
- <https://www.pronterface.com/>
- https://www.klipper3d.org/Sensorless_Homing.html
- <https://learn.watterott.com/silentstepstick/comparison/>
- <https://reprap.org/wiki/Category:Reference>
- <https://reprap.org/wiki/Calibration>

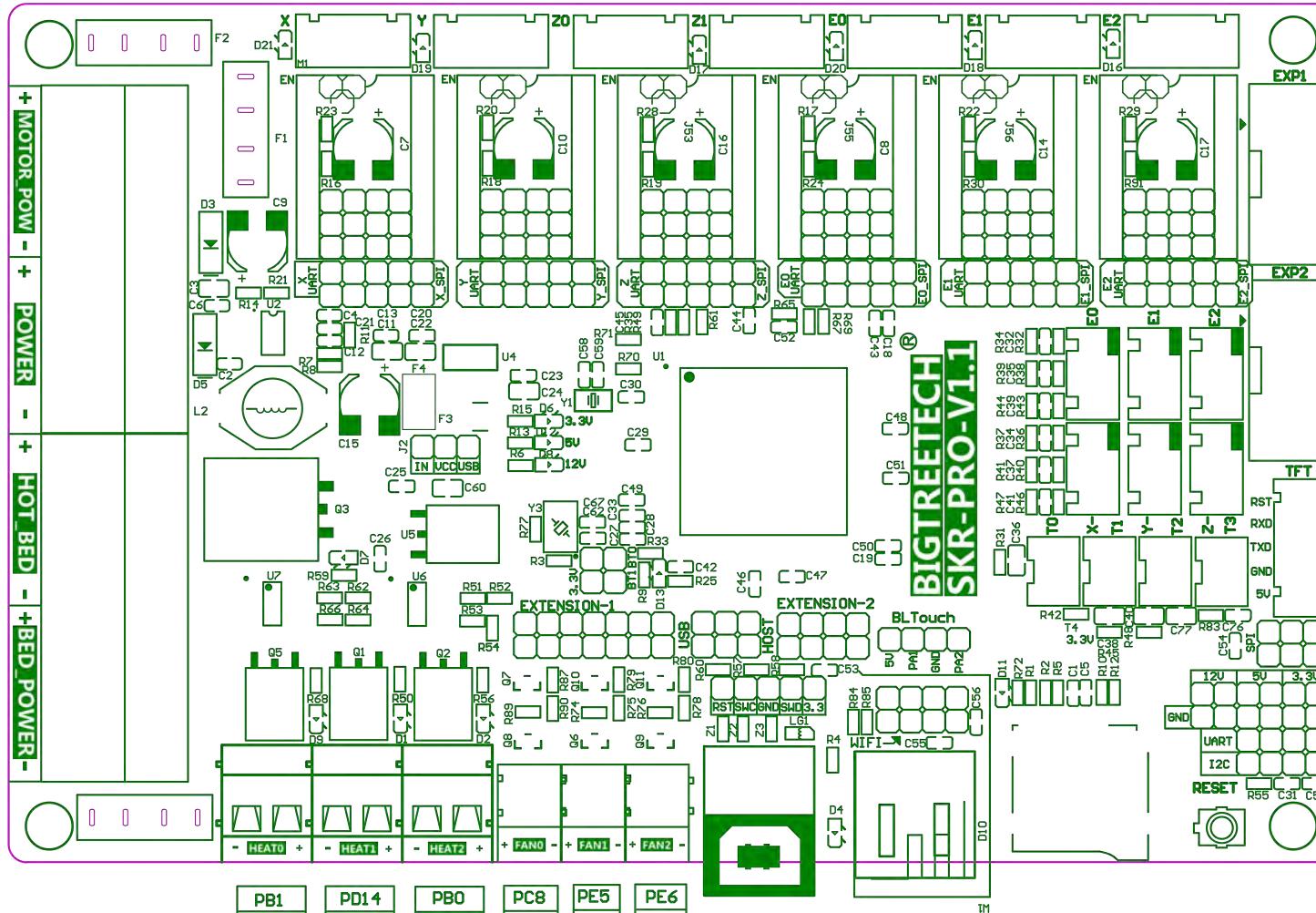
APPENDIX G

BIGTREETECH Reference Material

Original PIN Diagram:

3.3U	3.3U	3.3U	3.3U	3.3U	3.3U
RST M3 M2 M1					
SLP GND					

X UART MISO CS SCK MOSI Y UART MISO CS SCK MOSI Z UART MISO CS SCK MOSI EO UART MISO CS SCK MOSI E1 UART MISO CS SCK MOSI E2 UART MISO CS SCK MOSI

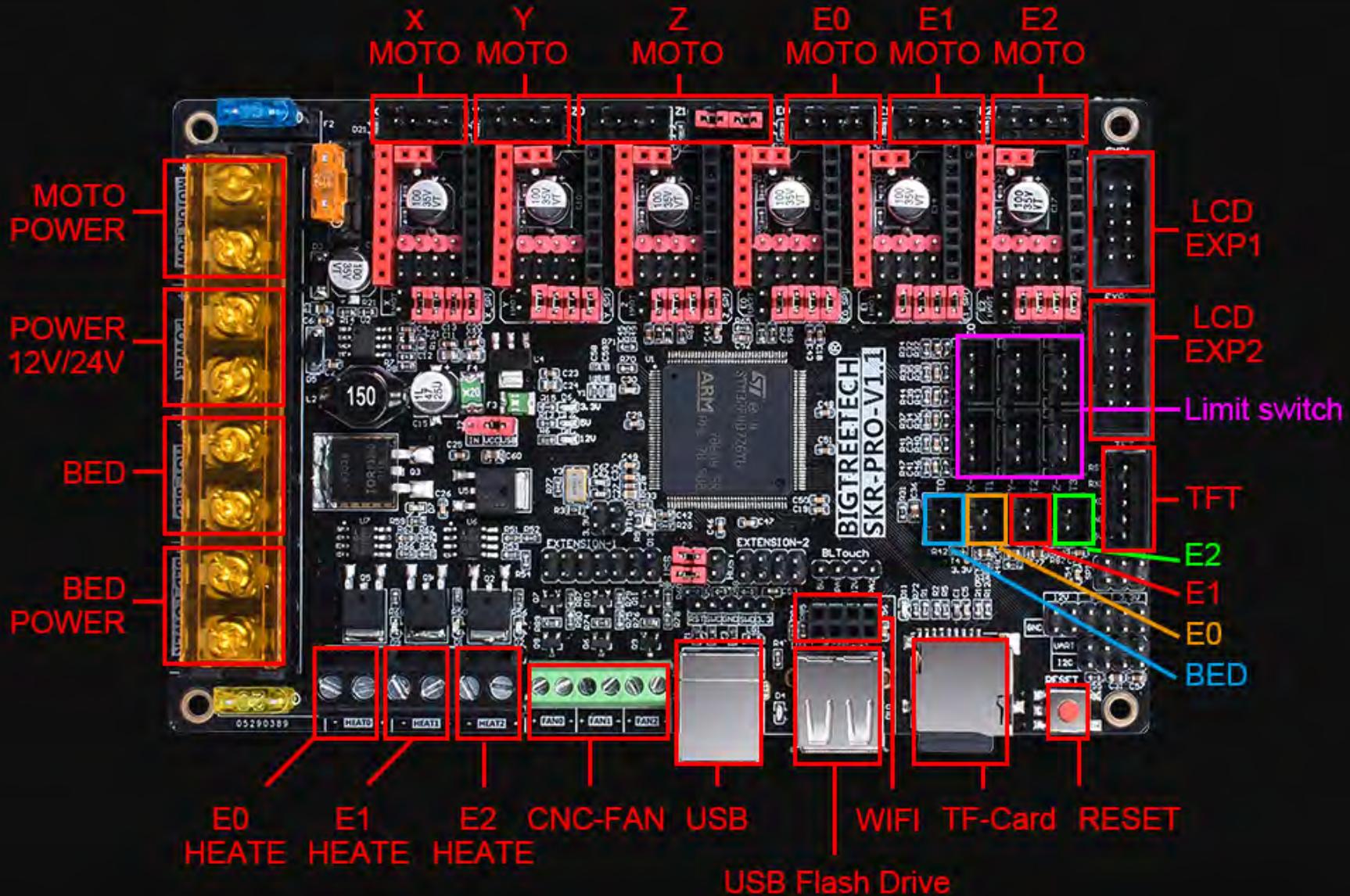


BIGTREETECH Reference Material**Original Wiring Diagram 1:**

5

5

from
<https://github.com/bigtreetech/BIGTRETECH-SKR-PRO-V1.1/blob/master/picture/BIGTRETECH-SKR-PRO-V1.1.jpg>

SKR wiring diagram

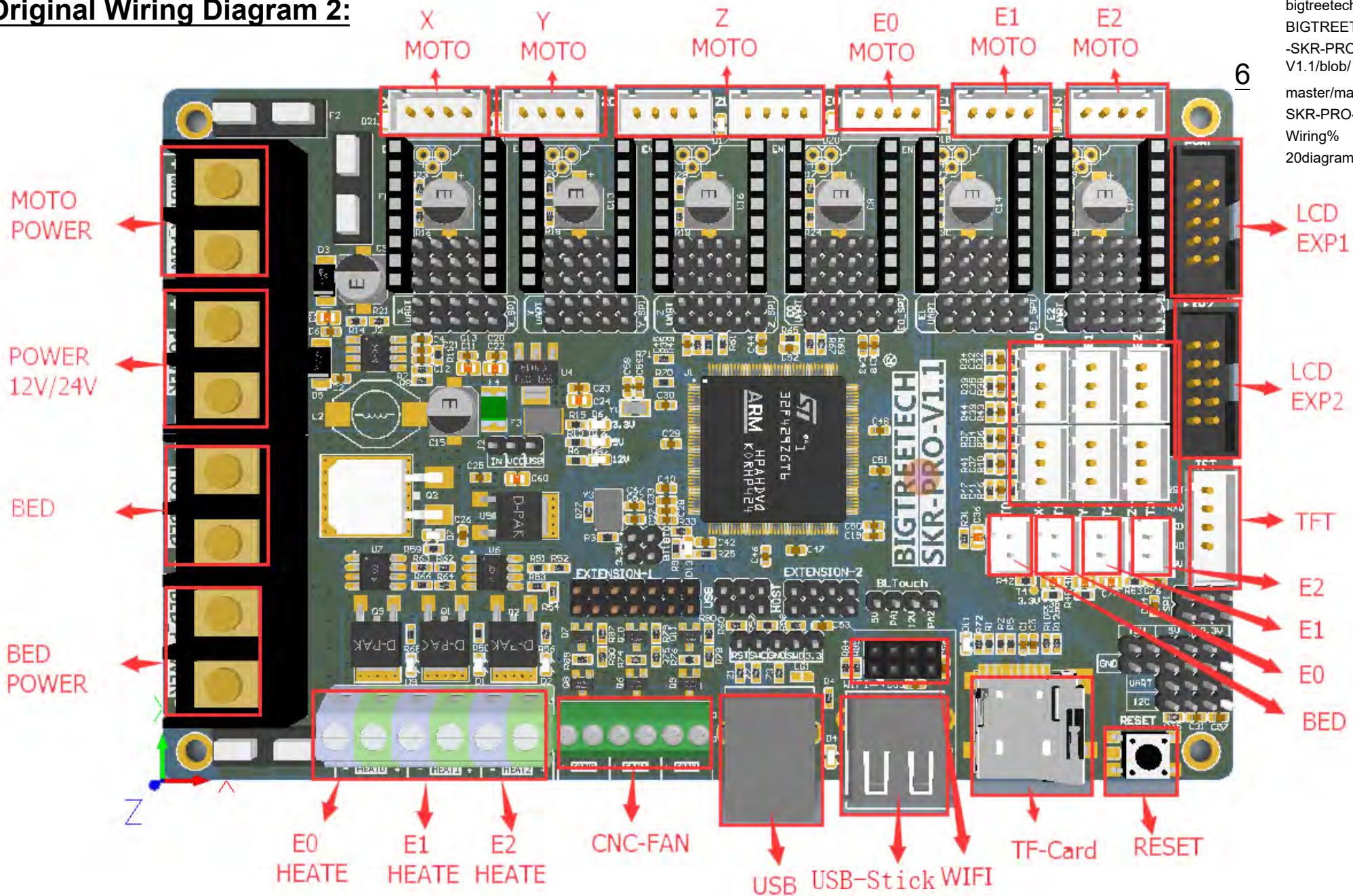
- Go to the next page.

APPENDIX G

BIGTREETECH Reference Material

6 from <https://github.com/bigtreeTech/BIGTREETECH-SKR-PRO-V1.1/blob/master/manual/SKR-PRO-V1.1-Wiring%20diagram.jpg>

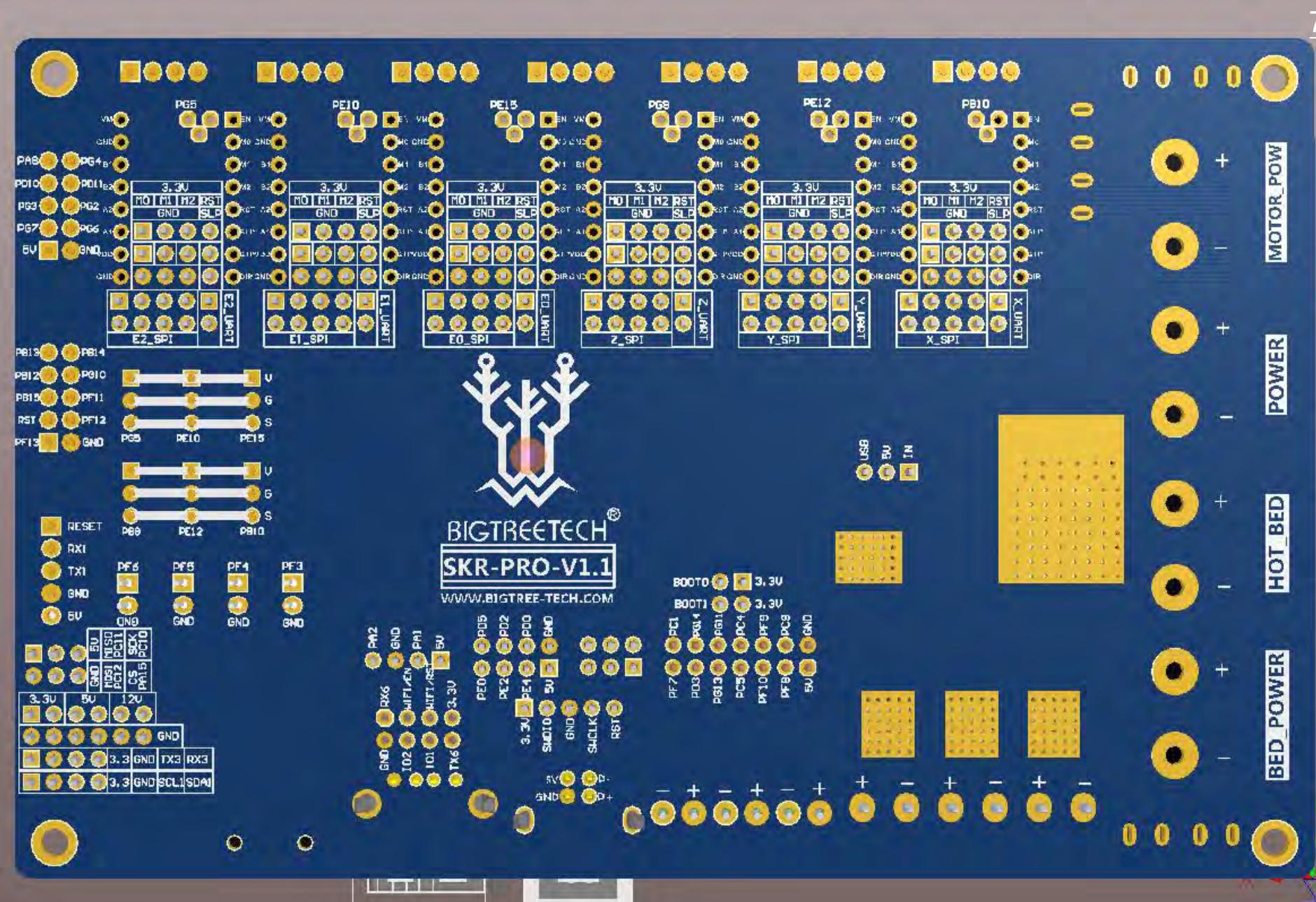
Original Wiring Diagram 2:



- Go to the next page.

BIGTREETECH Reference Material

Original PIN 1 Diagram:



7

1

from
<https://github.com/bigtreeTech/BIGTR-EETEC-H-SKR-PRO-V1.1/blob/master/picture/SKR%20PRO%20V1.1-BOTOM.png>

- Go to the next page.

APPENDIX G

Original Schematic Diagram:

BIGTREETECH Reference Material

8

