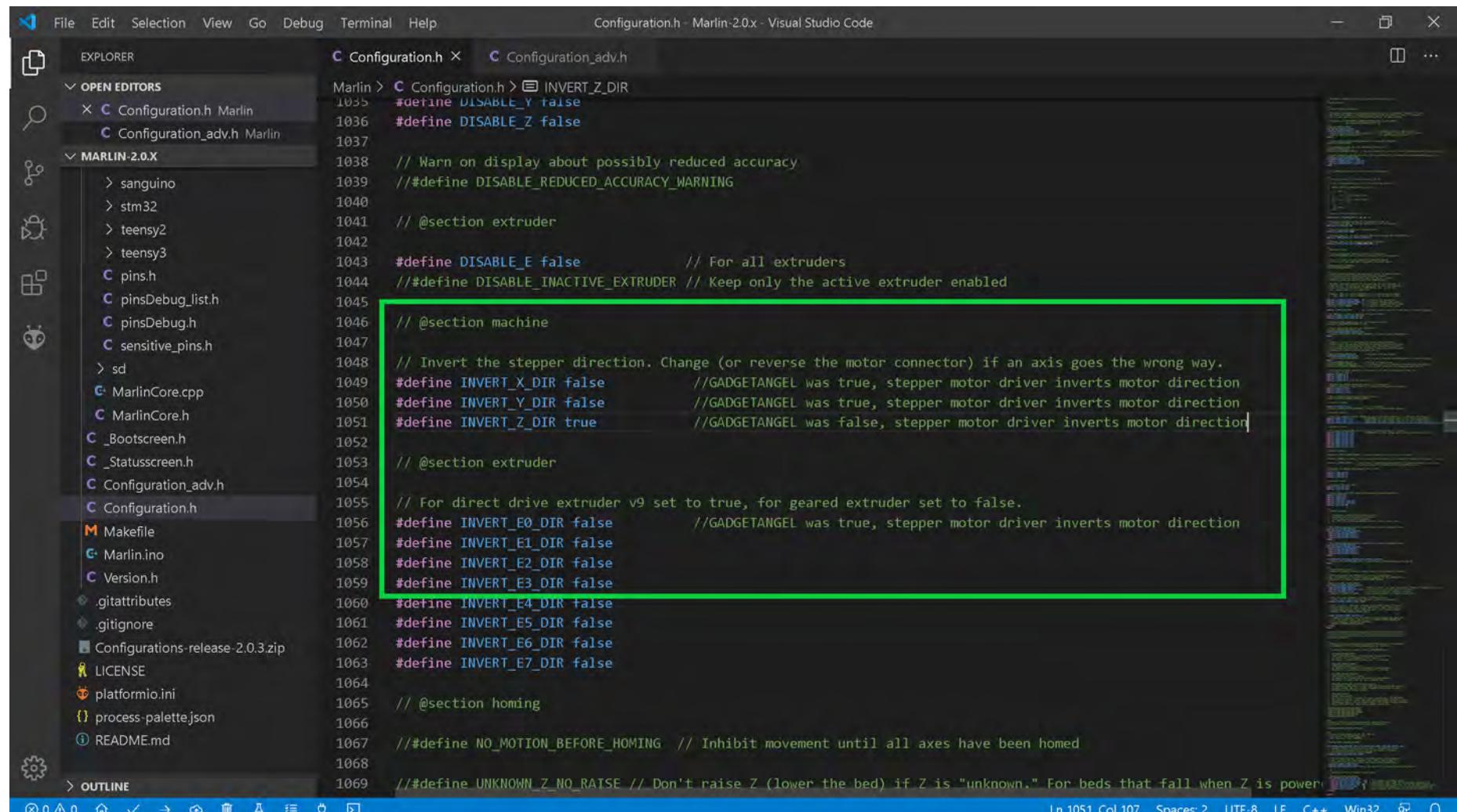


## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

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



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor 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 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

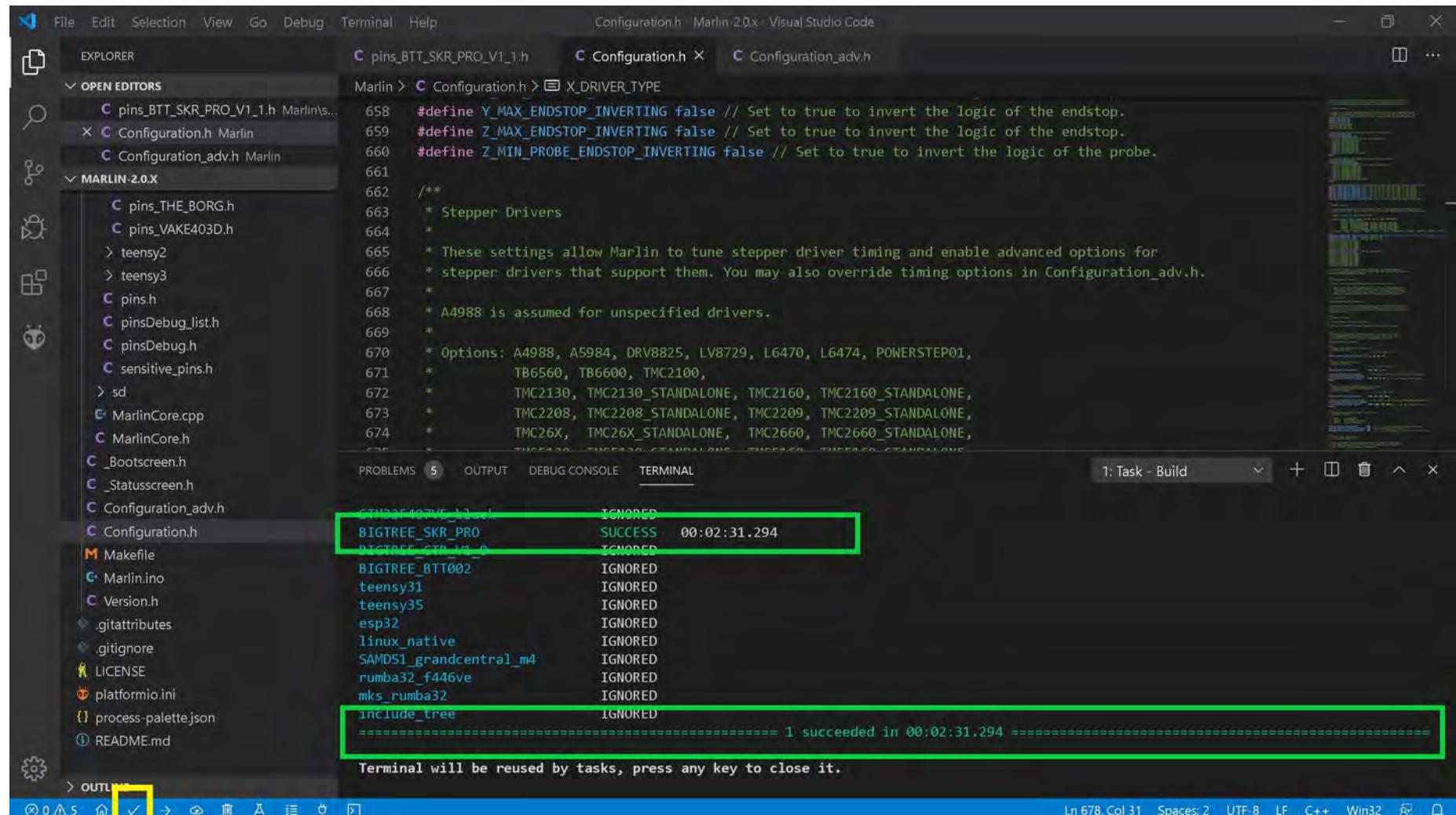
```

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

- Go to the next page.

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

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

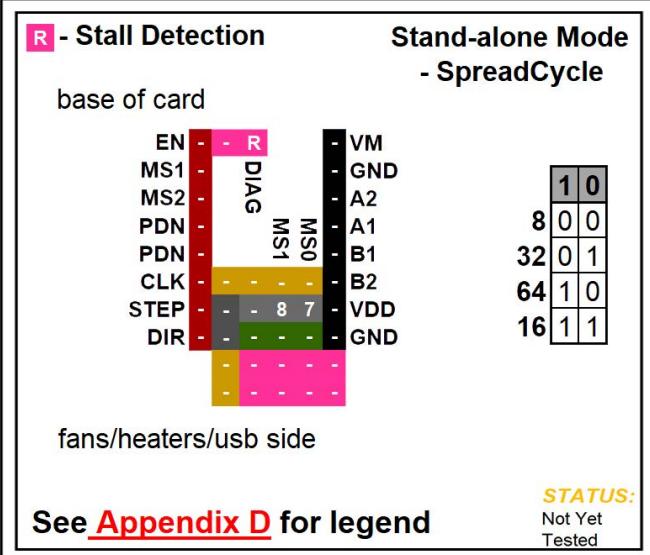


```

Configuration.h Marlin 2.0.x Visual Studio Code
pins_BTT_SKR_PRO_V1_1.h Configuration.h X Configuration_adv.h
Marlin > Configuration.h > X_DRIVER_TYPE
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE, TMC2660_STANDALONE
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
1: Task - Build + □ ×
STM32F107VE_L1_1_1 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

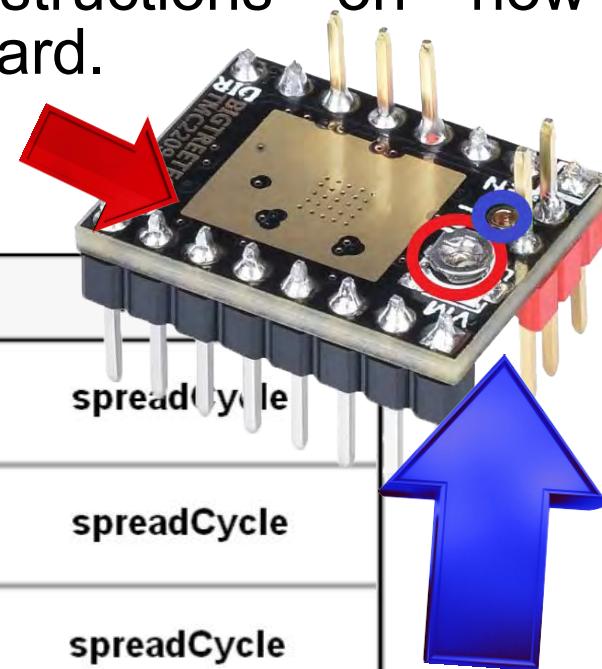
```

- 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.

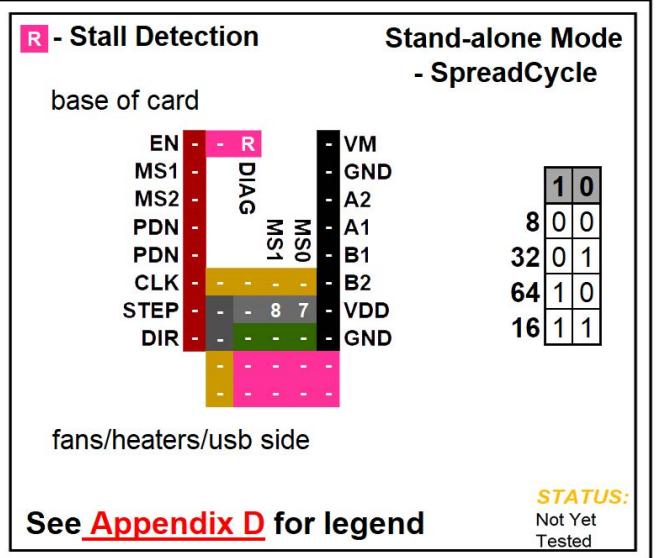


Driver Chip	MS1	MS0	Steps	Interpolation	
<b>BIQU® TMC2209</b> <small>Stand Alone Mode Maximum 64 Subdivision 28V DC 2.8A (peak)</small>	<b>Low</b>	<b>Low</b>	1 / 8	1 / 256	spreadCycle
	<b>Low</b>	<b>High</b>	1 / 32	1 / 256	spreadCycle
	<b>High</b>	<b>Low</b>	1 / 64	1 / 256	spreadCycle
	<b>High</b>	<b>High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

<b>Driving Current Calculation Formula</b> $R_S$ (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.

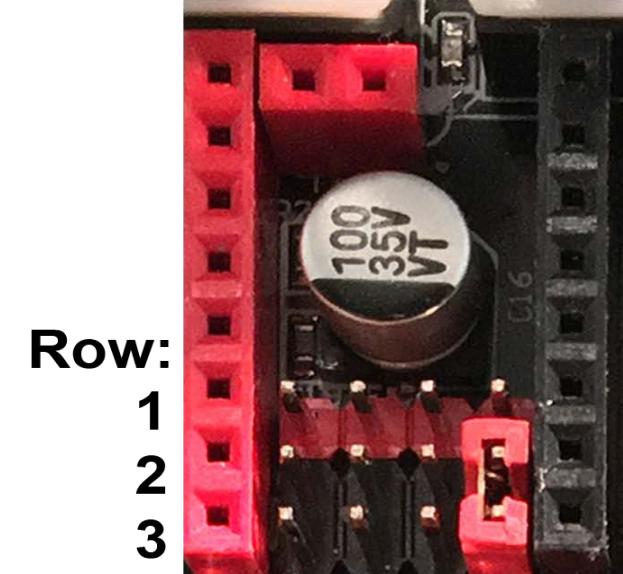
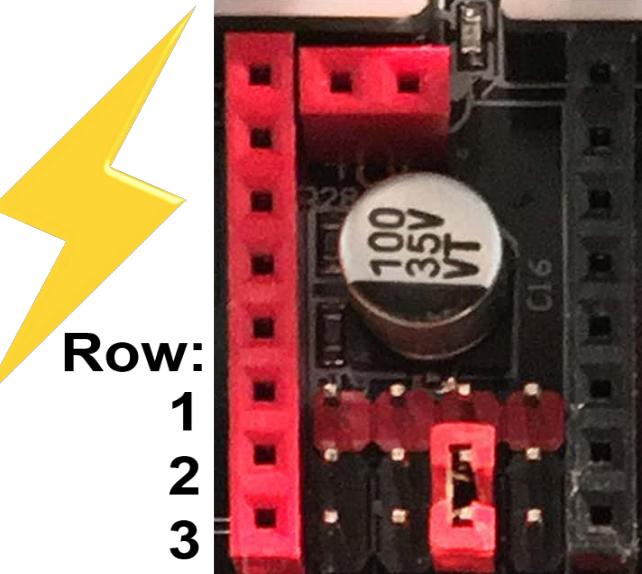


## BIQU TMC2209 V1.2

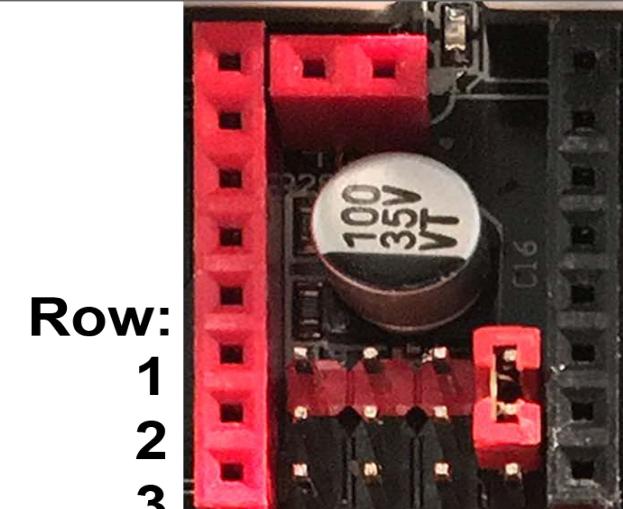
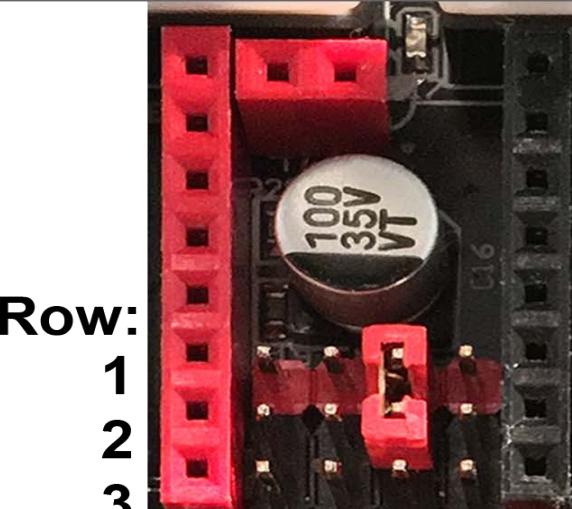
### Stand-alone Mode for SpreadCycle

## SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

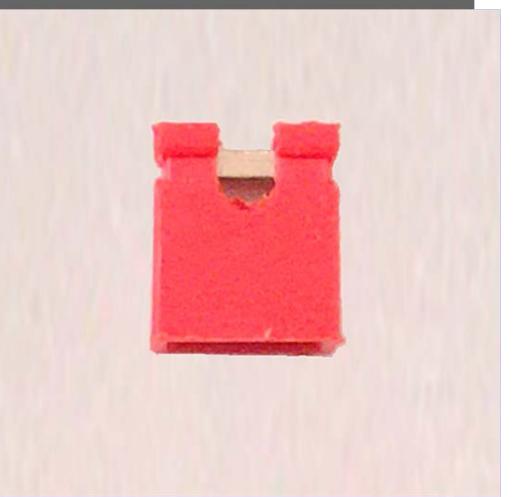
**Low** ➡ set Jumper between rows 2 and 3

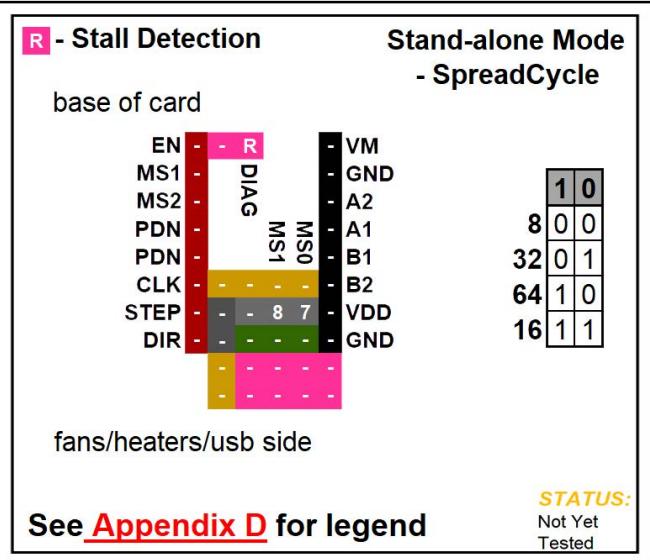


**High** ➡ set Jumper between rows 1 and 2



This is a  
Jumper:





# BIQU TMC2209 V1.2

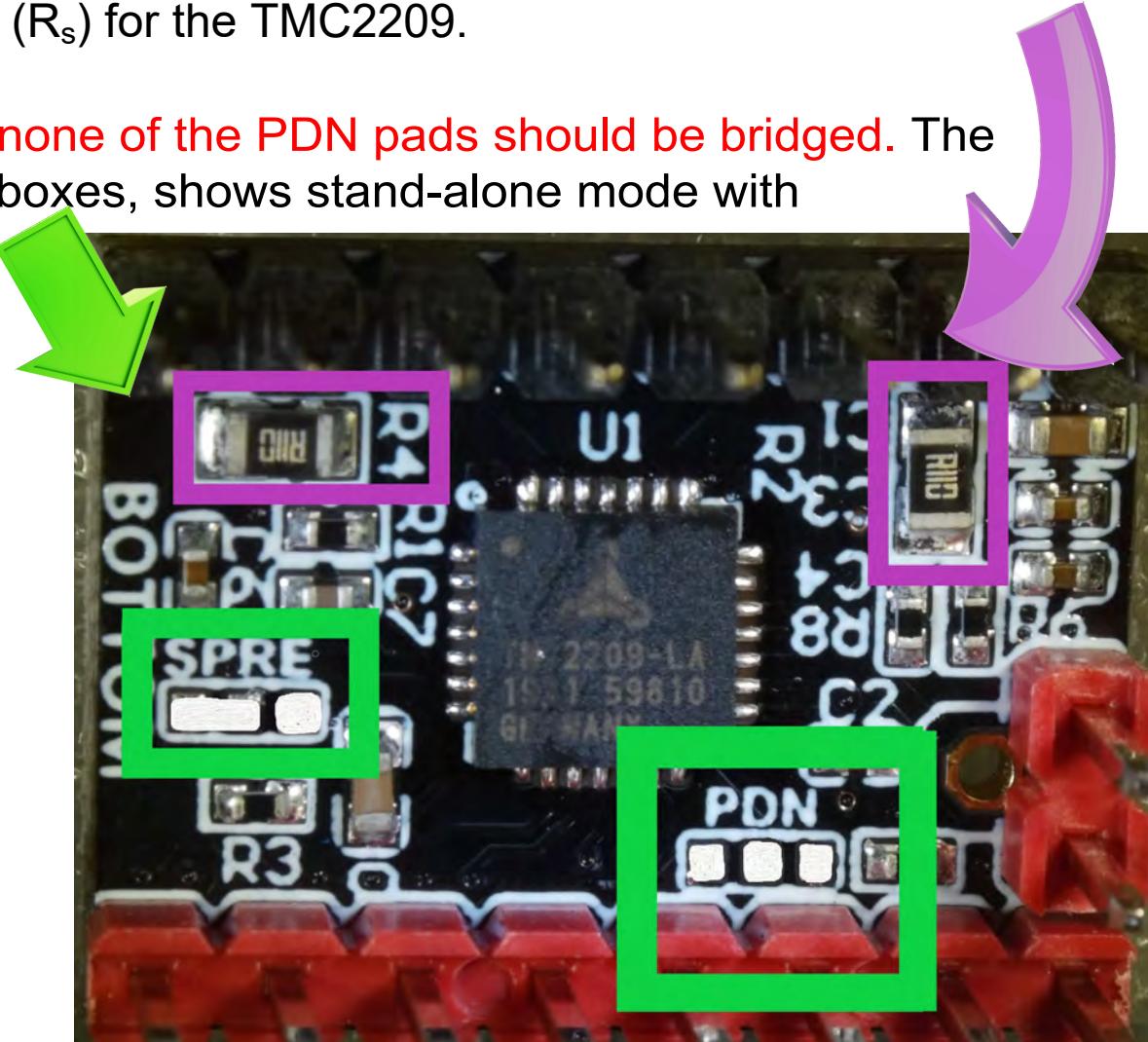
## Stand-alone Mode for SpreadCycle

**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

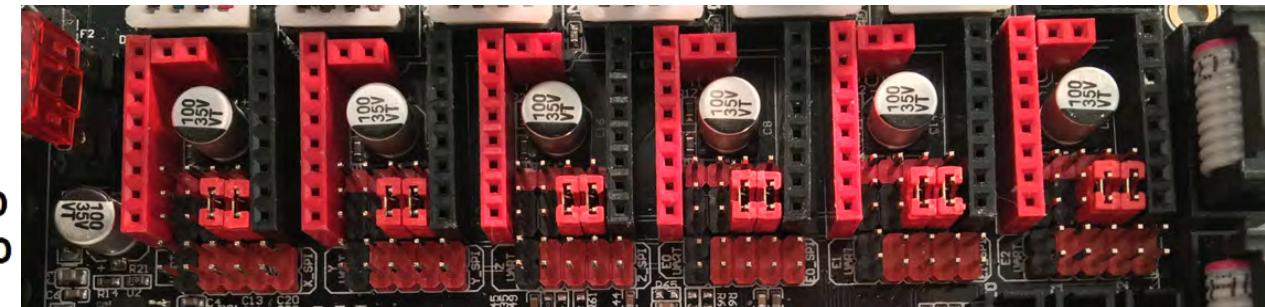
Stand-alone  
Mode

**1 / 8**

Interpolation:  
**1 / 256**

SpreadCycle

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



See [Appendix D](#) for legend

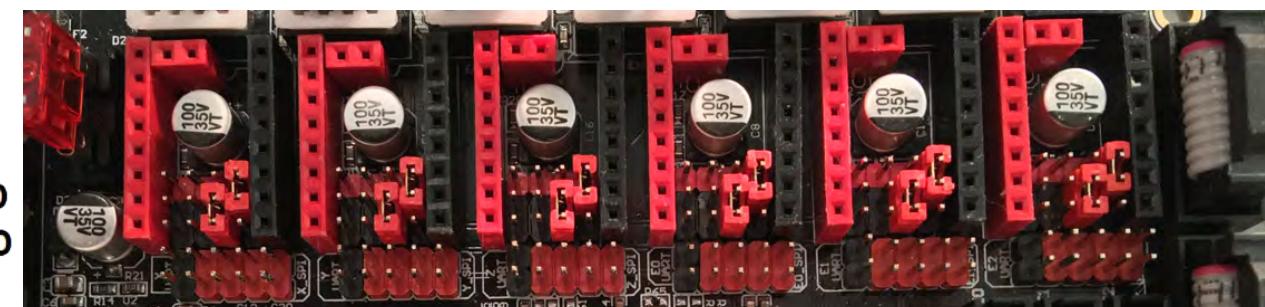
Stand-alone  
Mode

**1 / 32**

Interpolation:  
**1 / 256**

SpreadCycle

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



See [Appendix D](#) for legend

# Stand-alone SpreadCycle Mode

## Stand-alone Mode for SpreadCycle

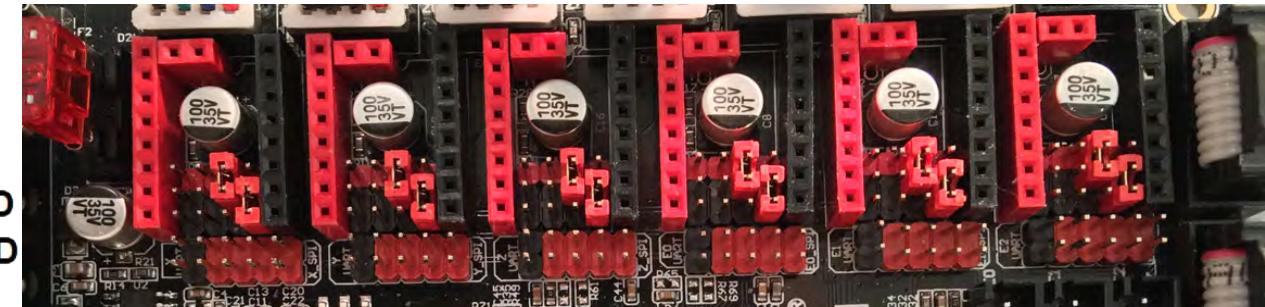
Stand-alone  
Mode

**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

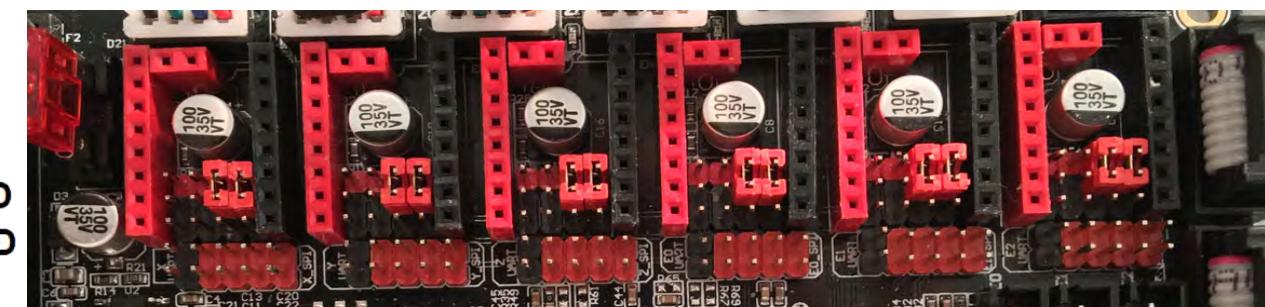
Stand-alone  
Mode

**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	-	7	-	GND

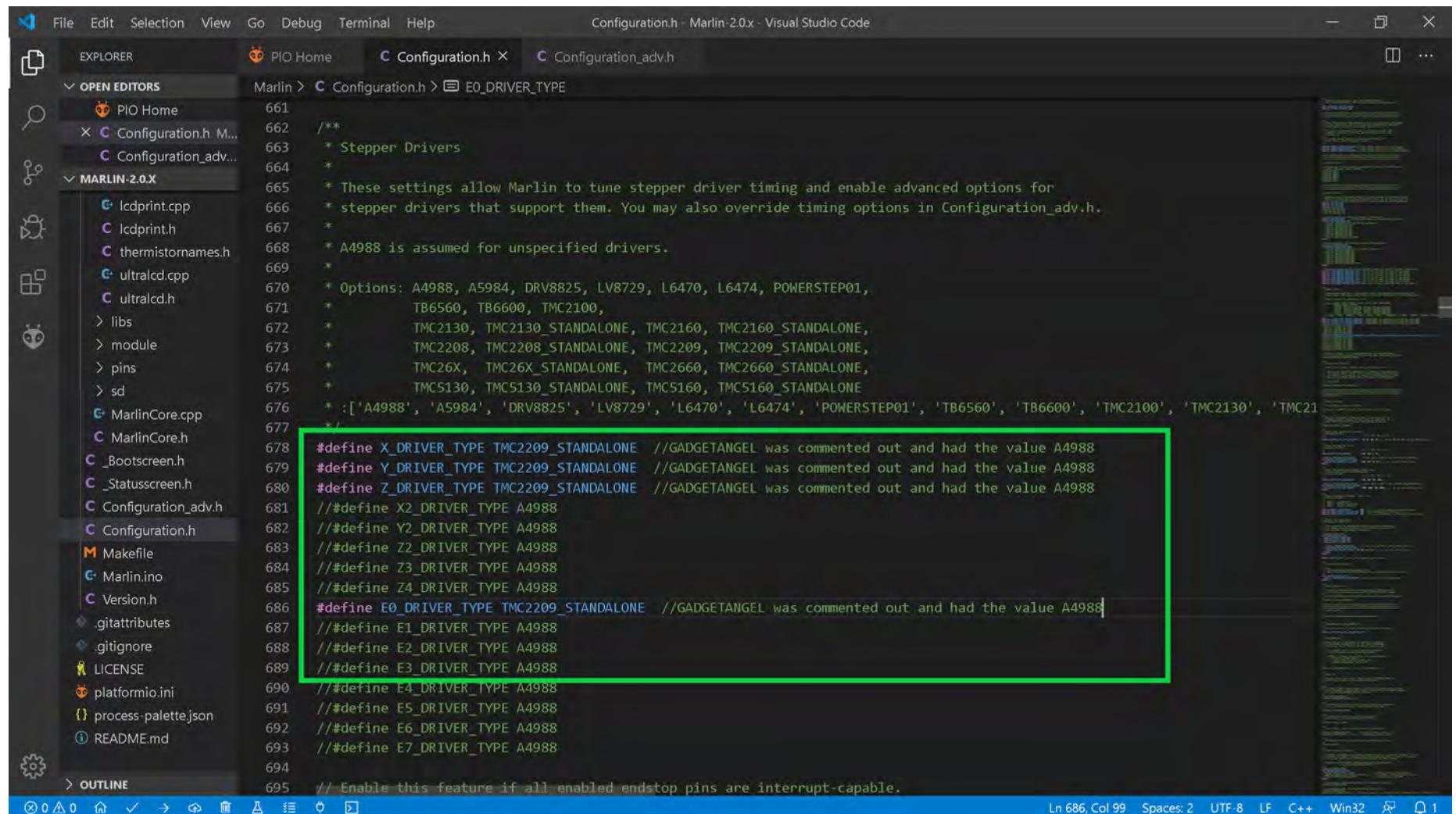


See [Appendix D](#) for legend

## 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 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 #define E0_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693
694 // Enable this feature if all enabled endstop pins are interrupt-capable.
695

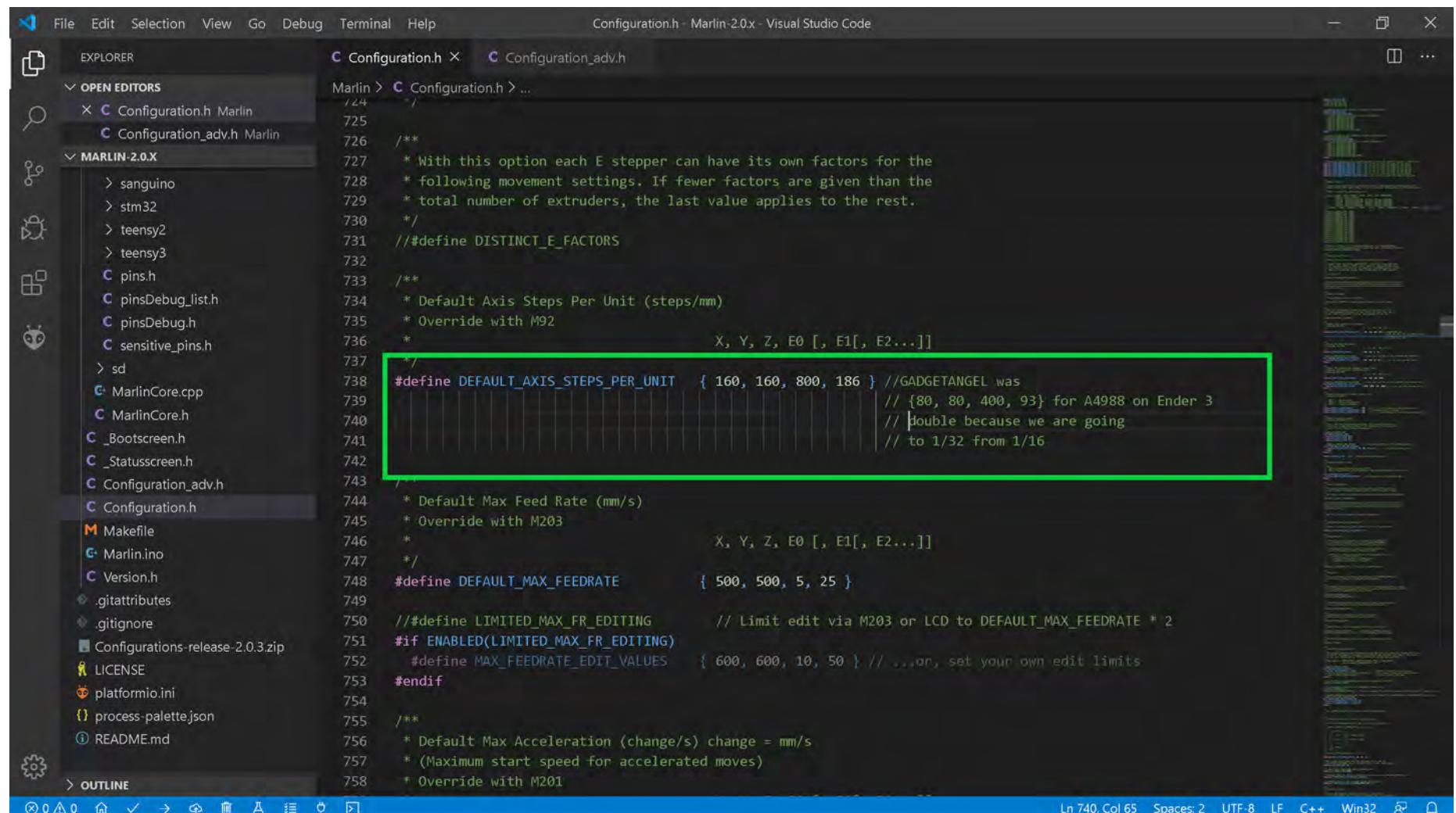
```

A green rectangular box highlights the driver type definitions starting from line 677. Lines 677 through 693 are shown with their original content, while lines 694 and 695 are shown with their commented-out versions. The status bar at the bottom of the code editor indicates the current line (Ln 686), column (Col 99), and other settings like spaces, encoding, and file type.

- 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:

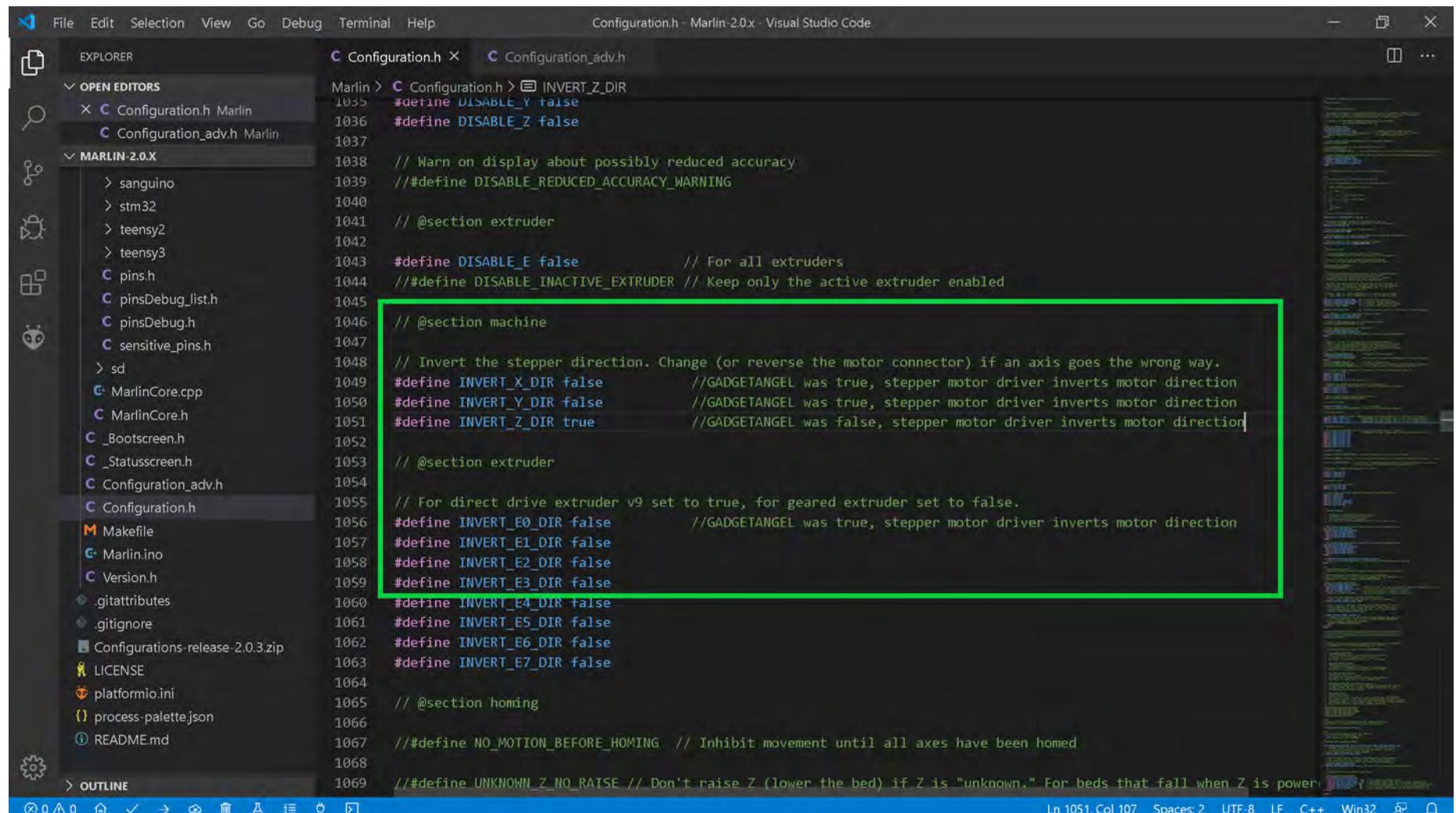
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

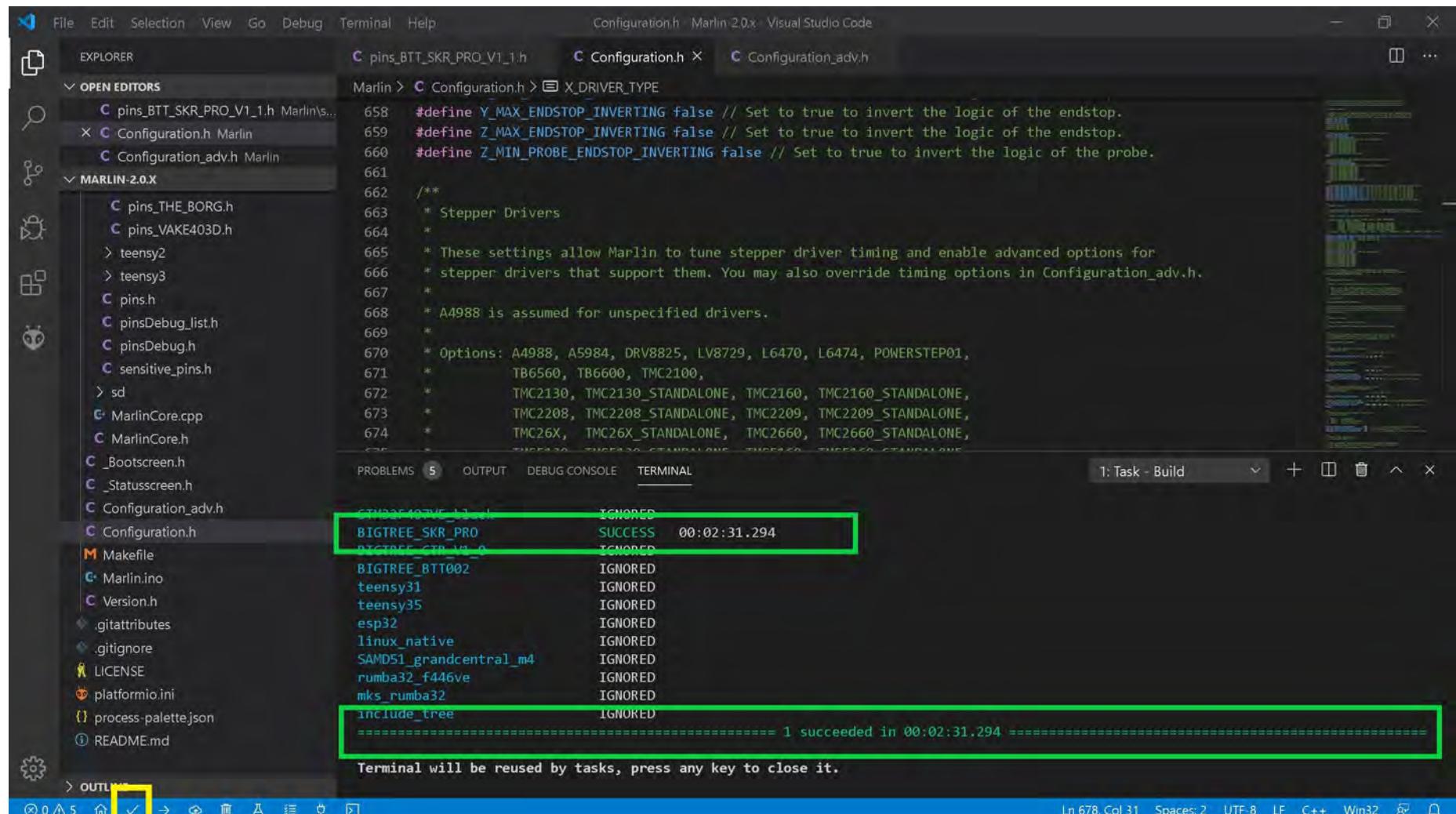
```

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

- Go to the next page.

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

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



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

- File Explorer:** Shows the project structure with files like `pins\_BTT\_SKR\_PRO\_V1\_1.h`, `Configuration.h`, and `Configuration\_adv.h`.
- Terminal:** Displays the build 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`. The log concludes with "1 succeeded in 00:02:31.294".

```

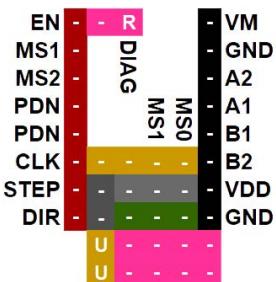
    BIGTREE_SKR_PRO           IGNORED
    BIGTREE_BTT002            IGNORED
    teensy31                  IGNORED
    teensy35                  IGNORED
    esp32                     IGNORED
    linux_native              IGNORED
    SAMD51_grandcentral_m4   IGNORED
    rumba32_f446ve            IGNORED
    mks_rumba32               IGNORED
    include_tree               IGNORED
=====
===== 1 succeeded in 00:02:31.294 =====
  
```
- Bottom Status Bar:** Shows the terminal reused message and status indicators (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**UART 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

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

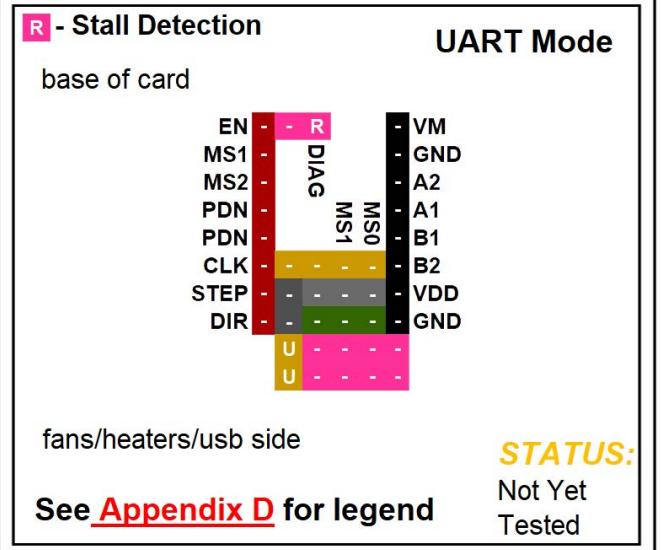
See Appendix B #6. Use 50% to 90%  
as shown below:

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

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

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

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

**BIQU TMC2209 V1.2****UART Mode**

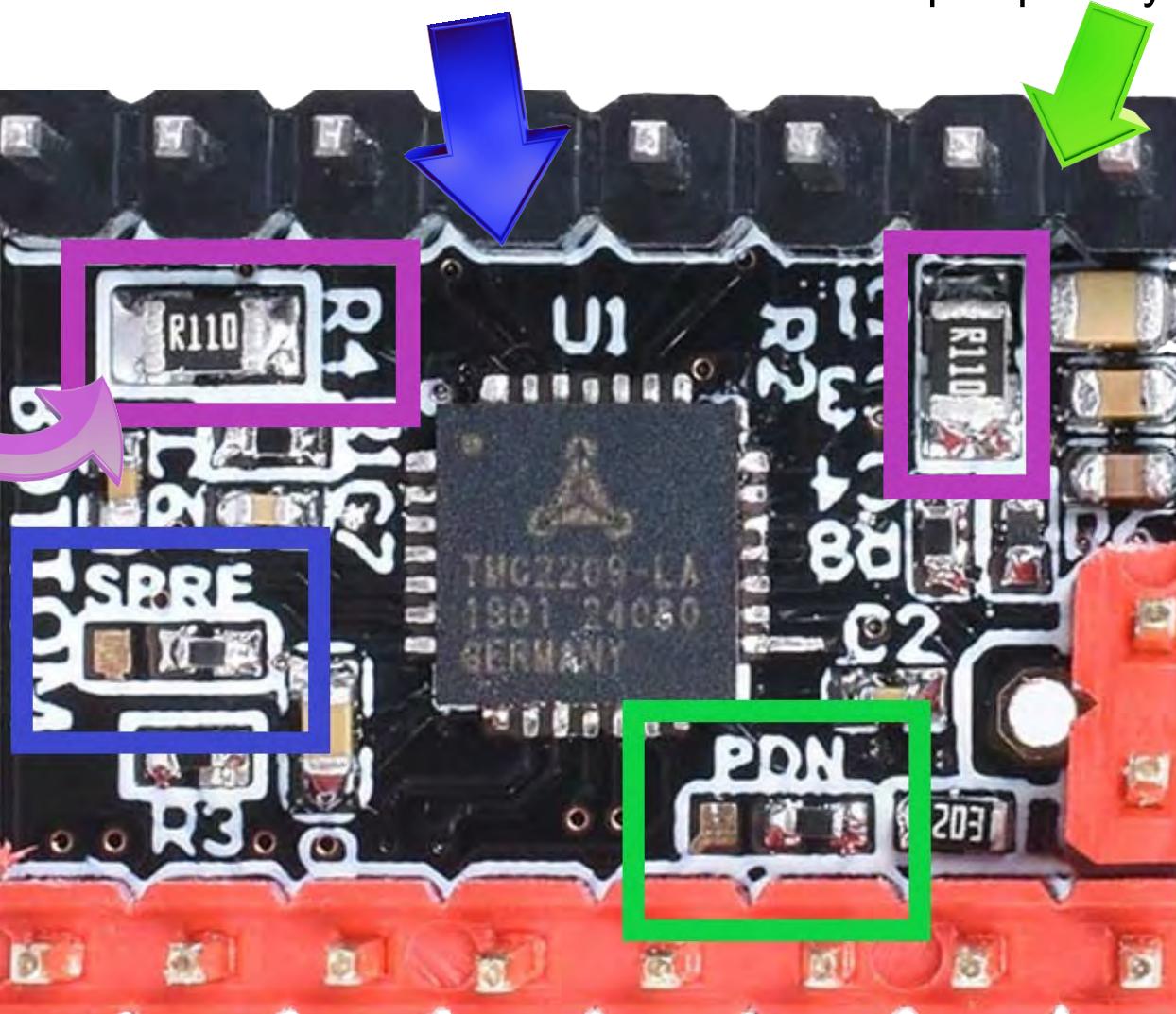
**MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.**

**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

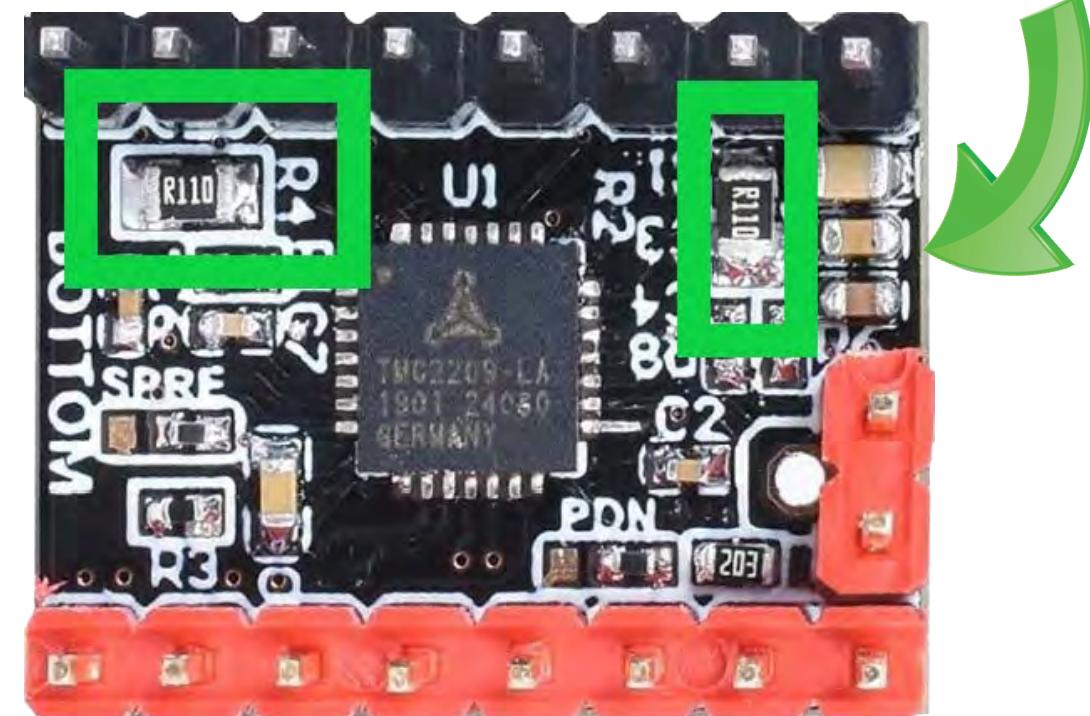
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

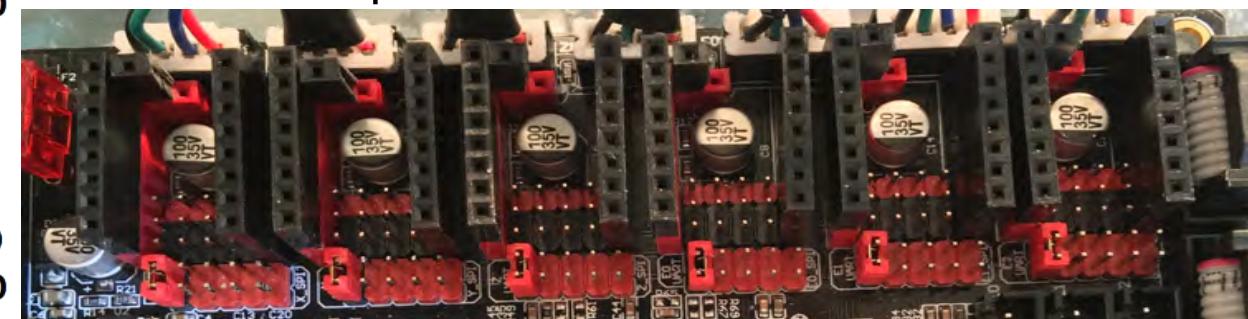
$R_s = R150$  is 0.15 Ohms

$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



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



See [Appendix D](#) for legend

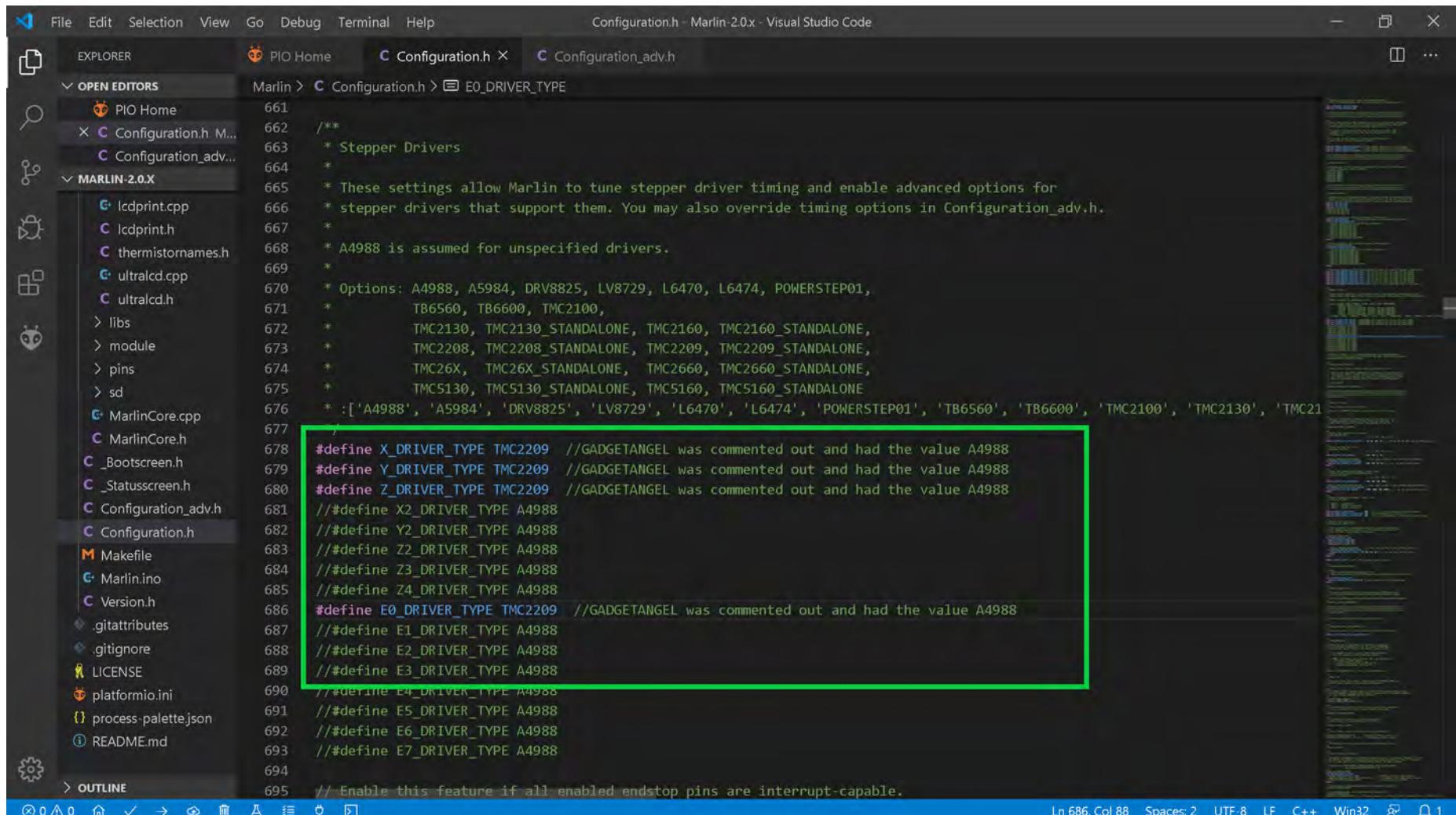
**UART**

EN	-	R	-	VM
MS1	-	DIA	GND	
MS2	-	G	A2	
PDN	-		A1	
PDN	-		B1	
CLK	---	MS1	B2	
STEP	---	MS0	VDD	
DIR	---		GND	
	U	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 snippet of code, which is part of the Marlin 2.0.x configuration:

```

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.

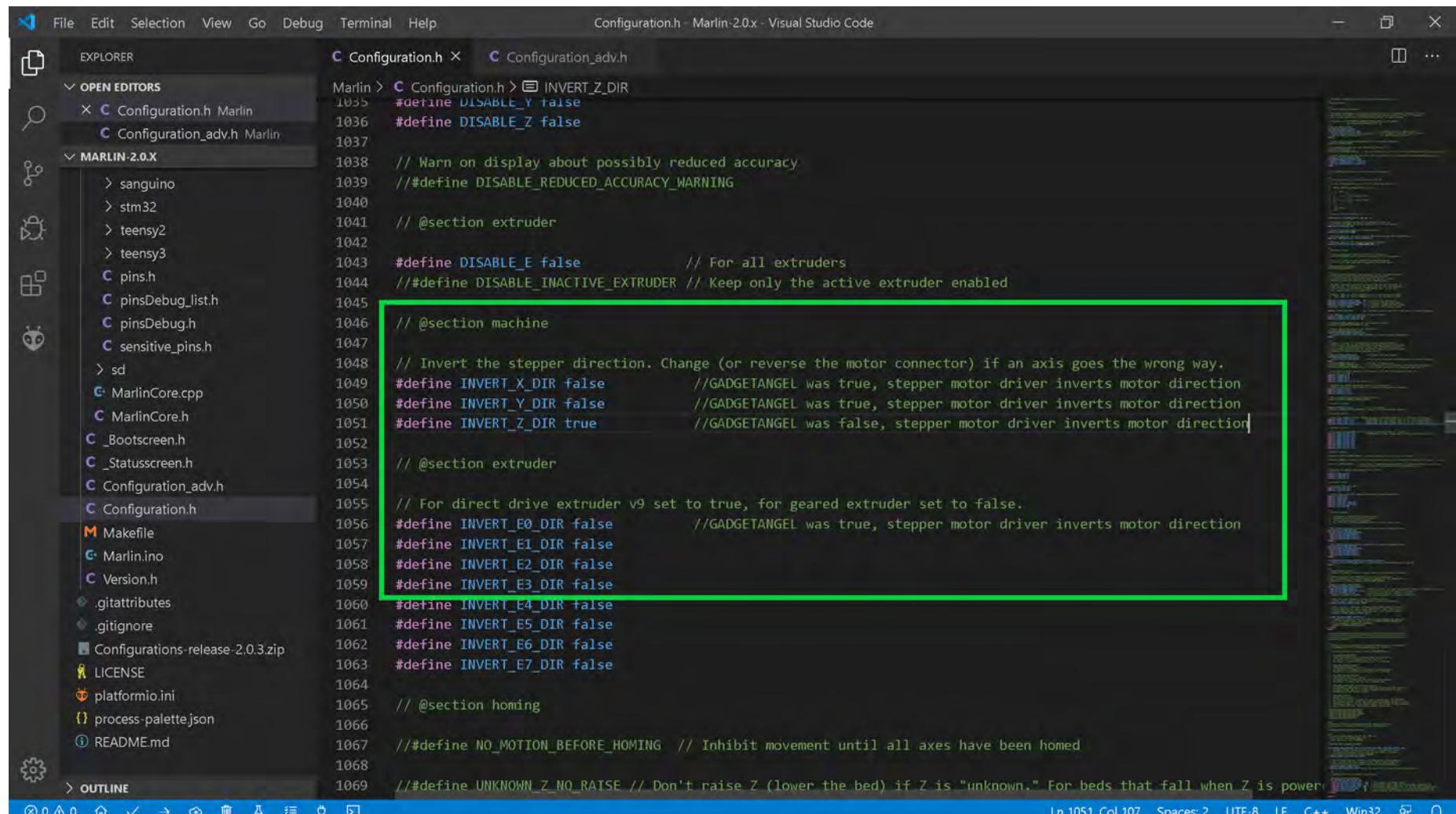
```

A green rectangular box highlights the driver configuration section, specifically the lines defining the driver type for each axis (X, Y, Z, E0, E1, E2, E3, E4, E5, E6, E7). The lines for E1 through E7 are currently commented out with double slashes ('//').

- 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



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 up

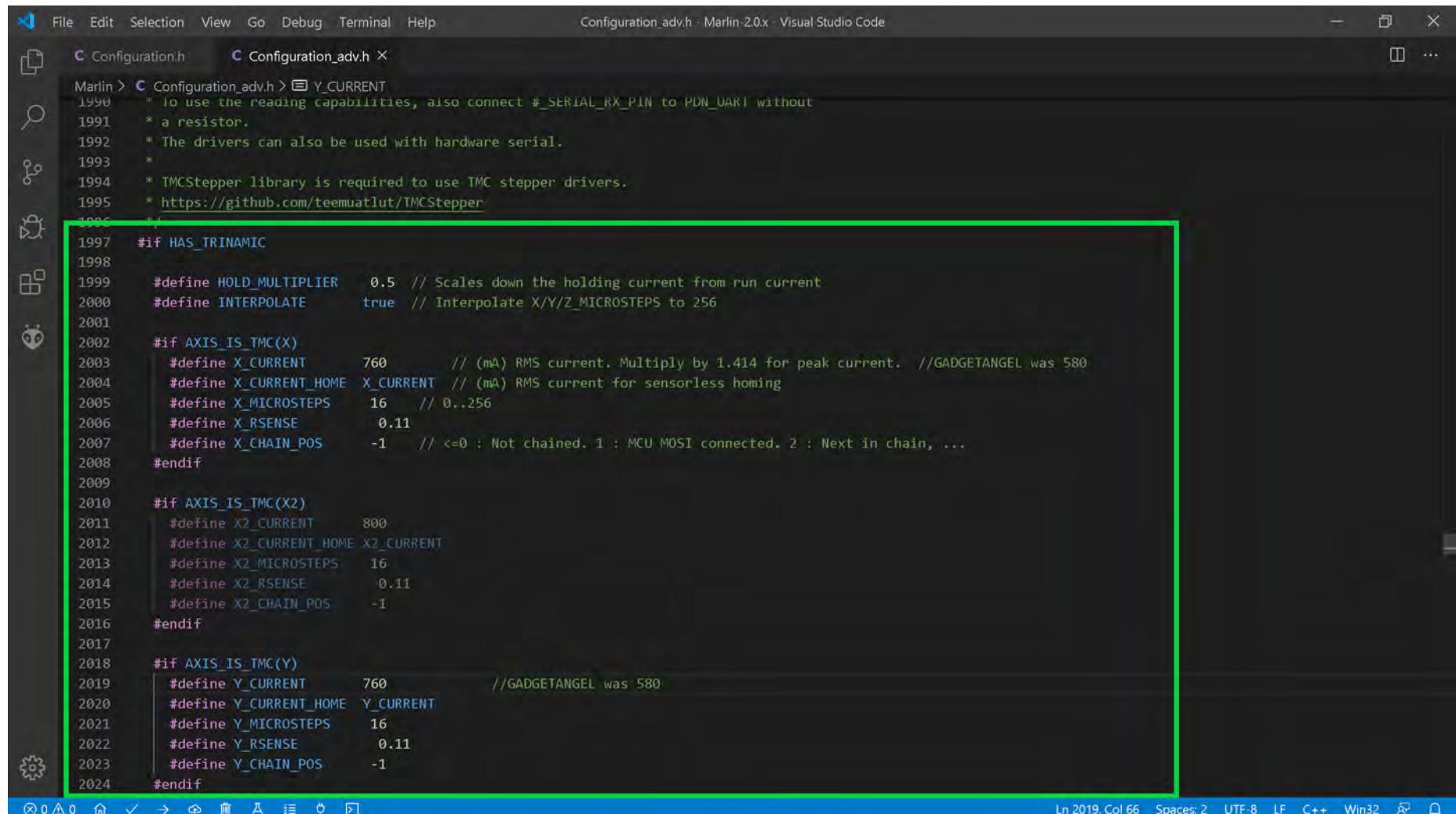
```

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".



```

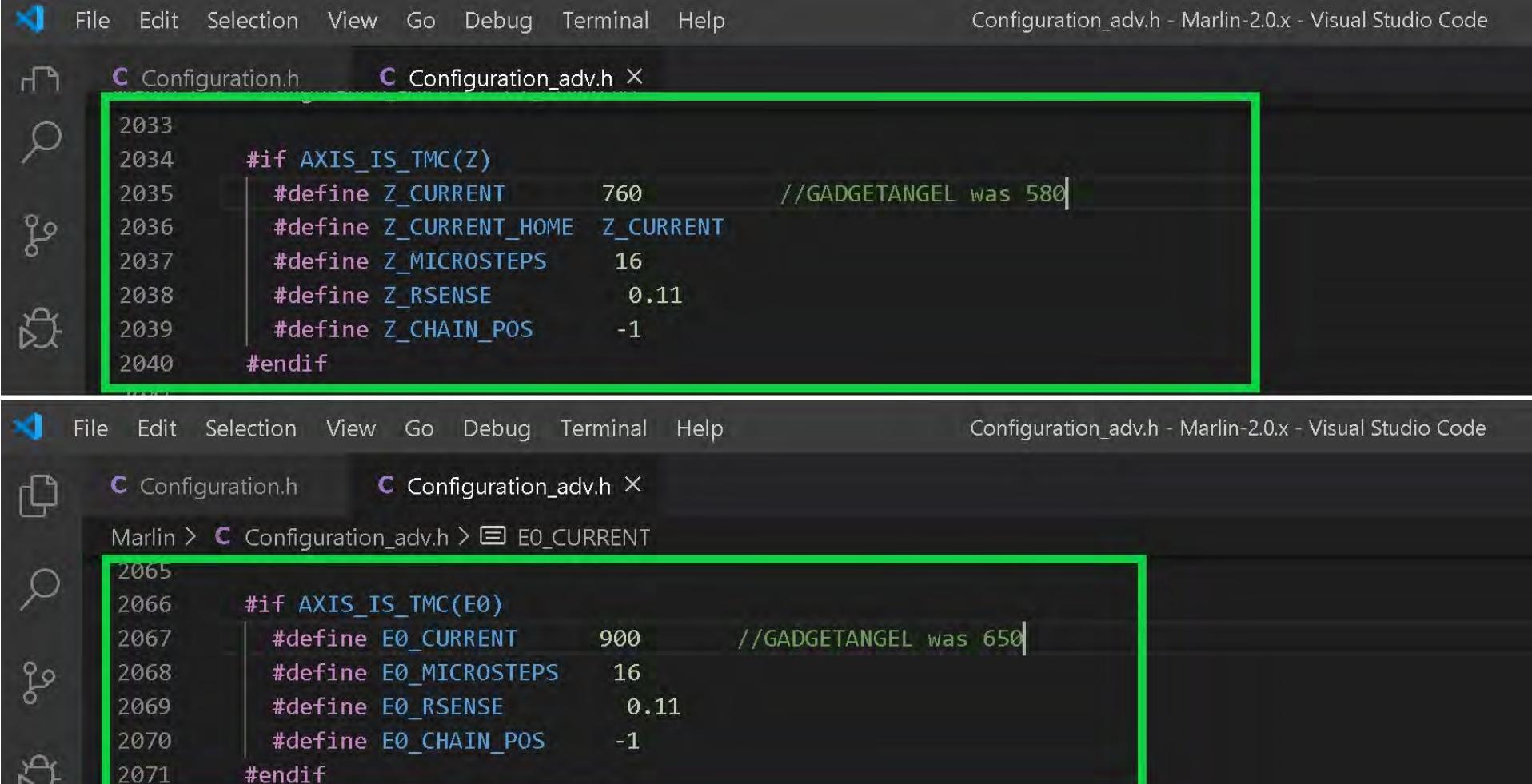
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers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

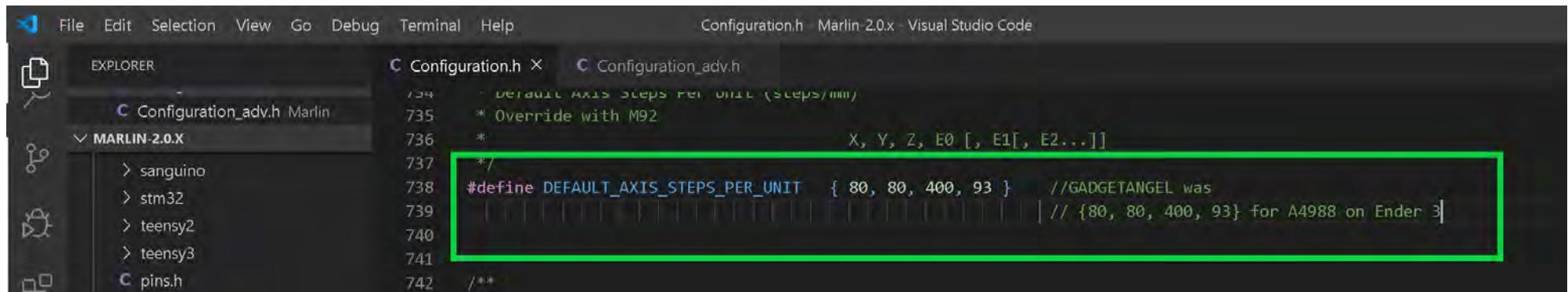


```
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h X
2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760          //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME Z_CURRENT
2037   #define Z_MICROSTEPS    16
2038   #define Z_RSENSE        0.11
2039   #define Z_CHAIN_POS     -1
2040 #endif
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
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 TMC2209 V1.2 Drivers in UART Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes



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

EXPLORER Configuration.h X Configuration\_adv.h

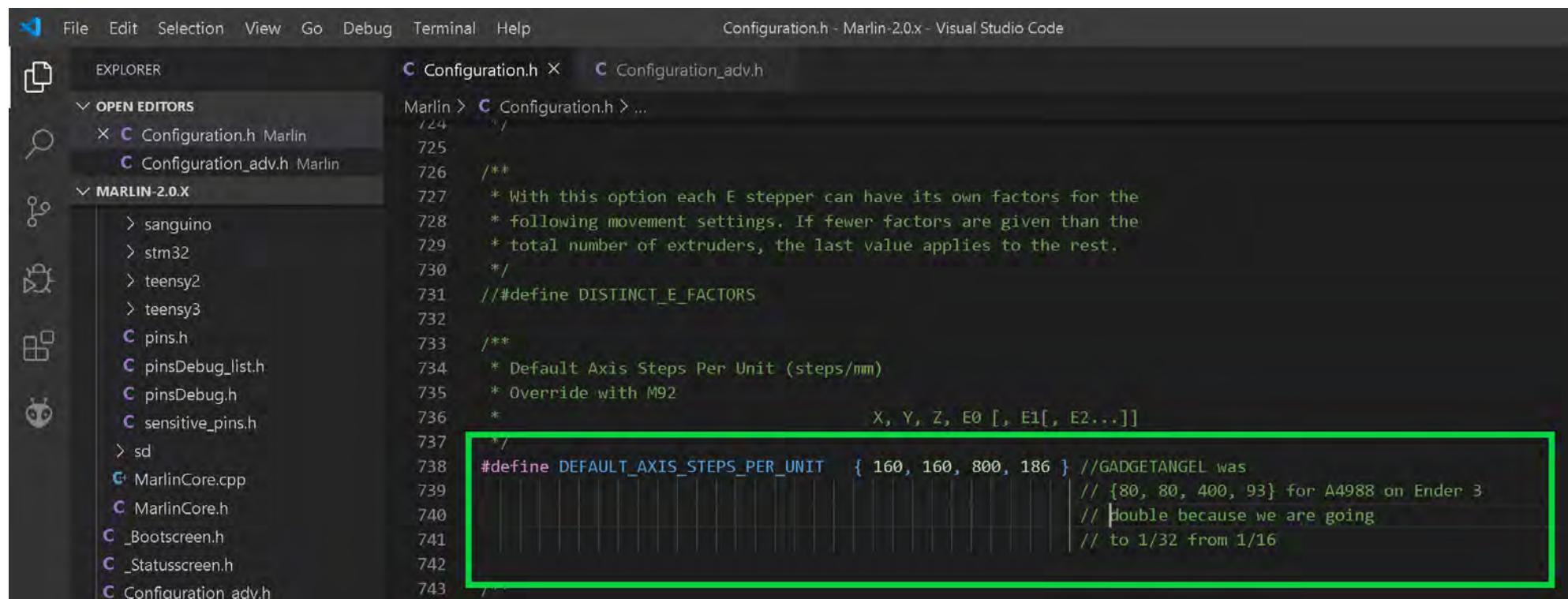
MARLIN-2.0.X

```

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

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



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

EXPLORER Configuration.h X Configuration\_adv.h

OPEN EDITORS Configuration.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X

```

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

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

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

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

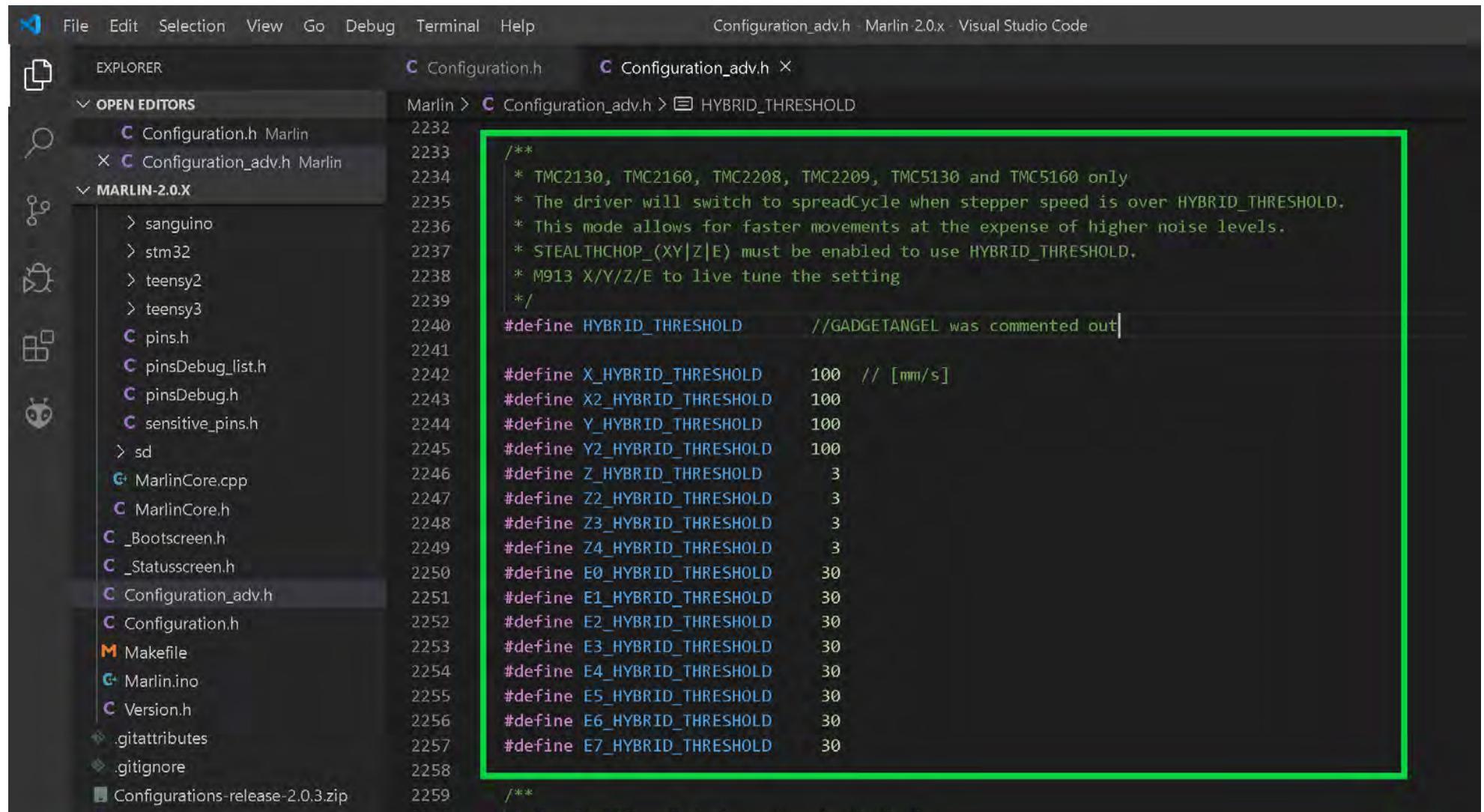
EXPLORER Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > STEALTHCHOP_XY
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 */

Table of Contents
```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

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



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

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Explorer:** Shows the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X' (including files like Configuration.h, Configuration\_adv.h, pins.h, etc.)
- Code Editor:** Displays the content of Configuration\_adv.h. A specific section of the code 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

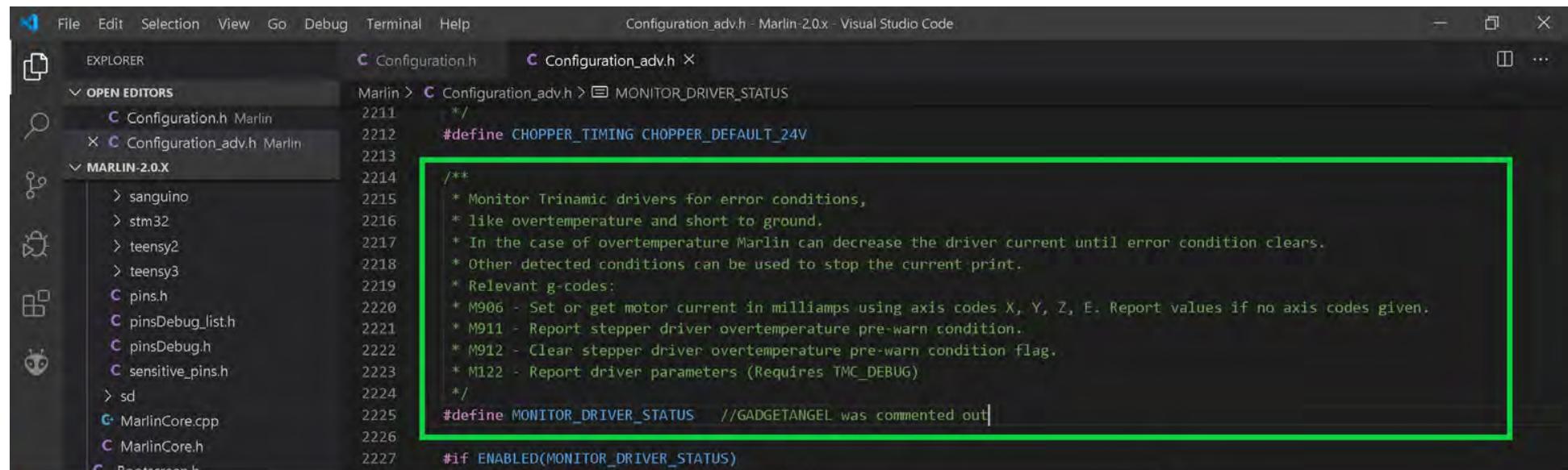
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD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



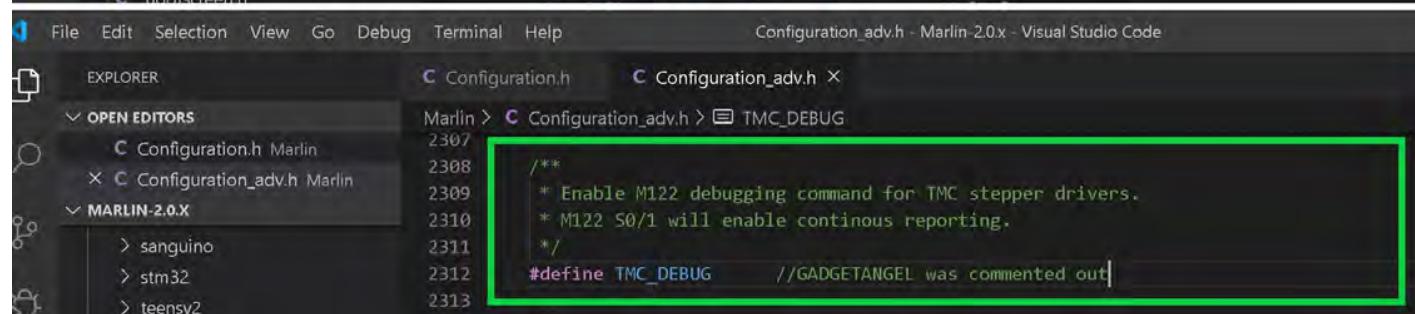
File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

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

```

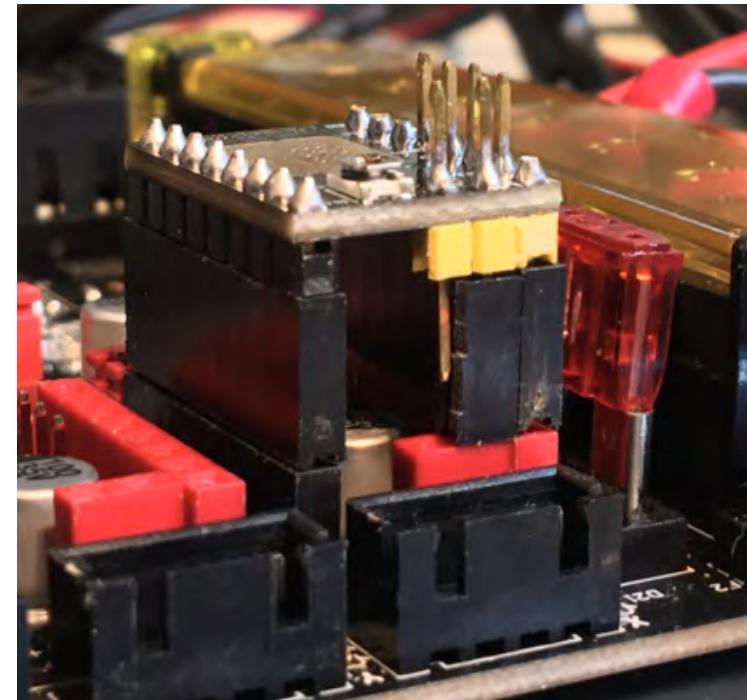
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- 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"

```

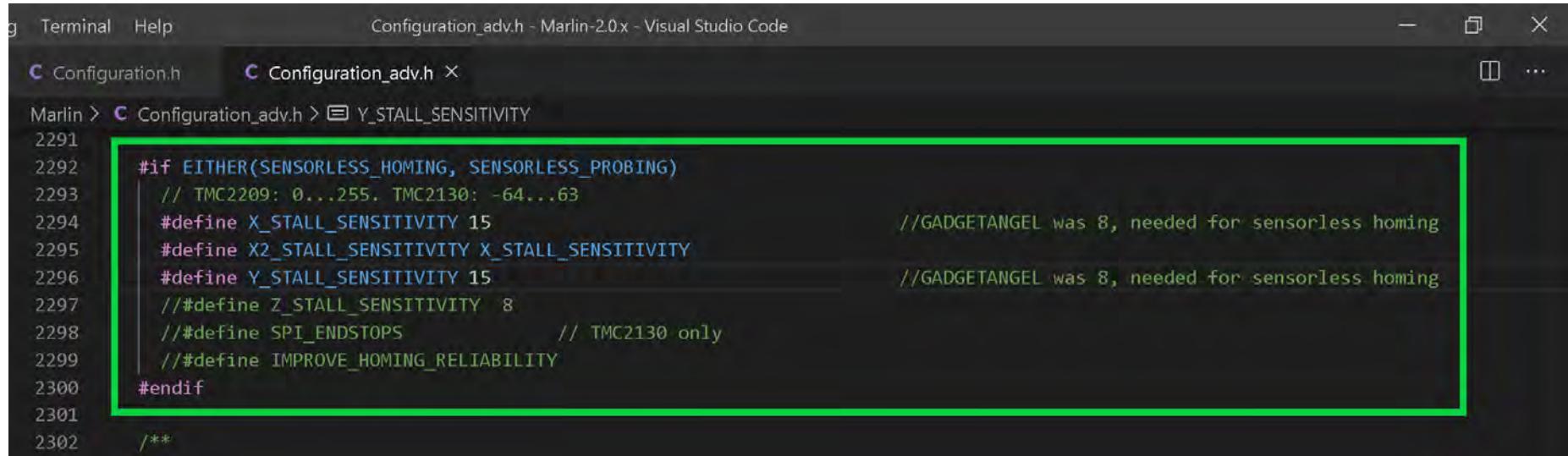
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h X
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING
  2281   */
  2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL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

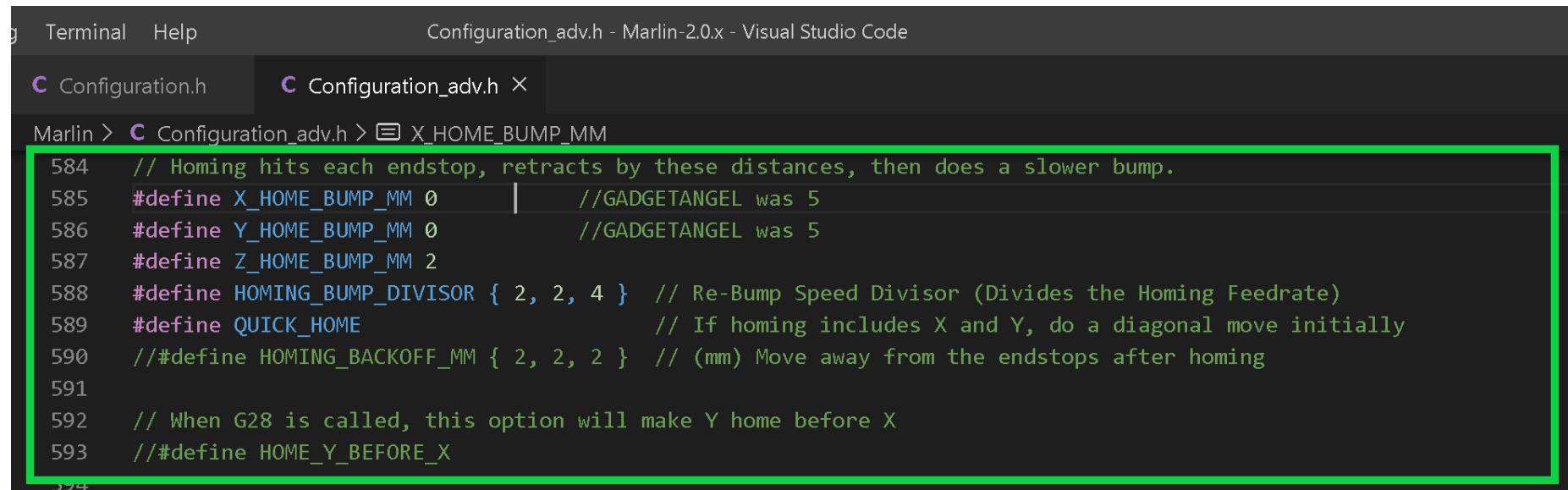


```

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.



```

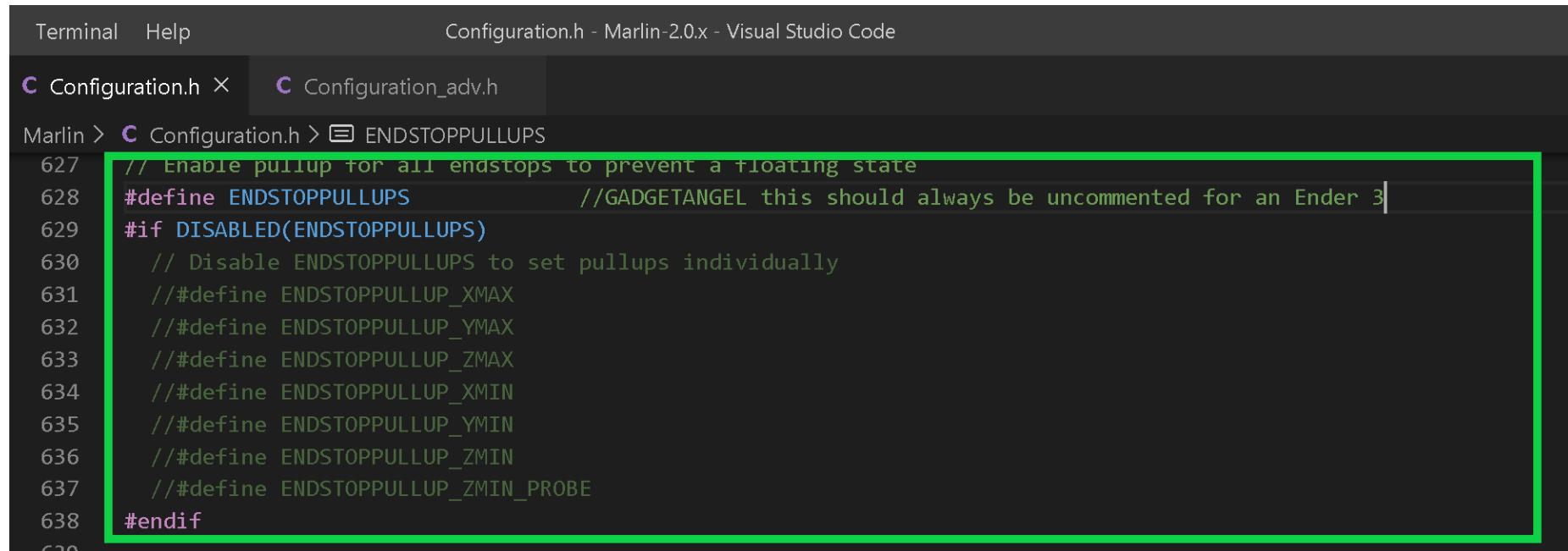
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

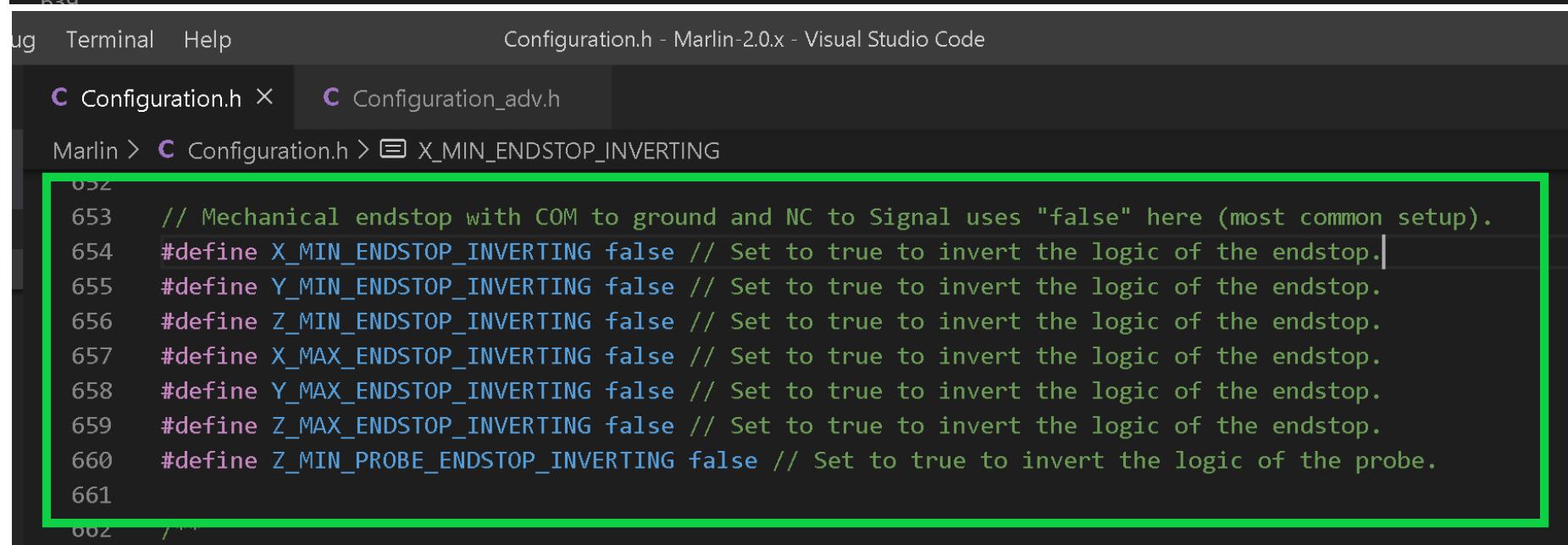
C Configuration.h X C Configuration\_adv.h

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

```

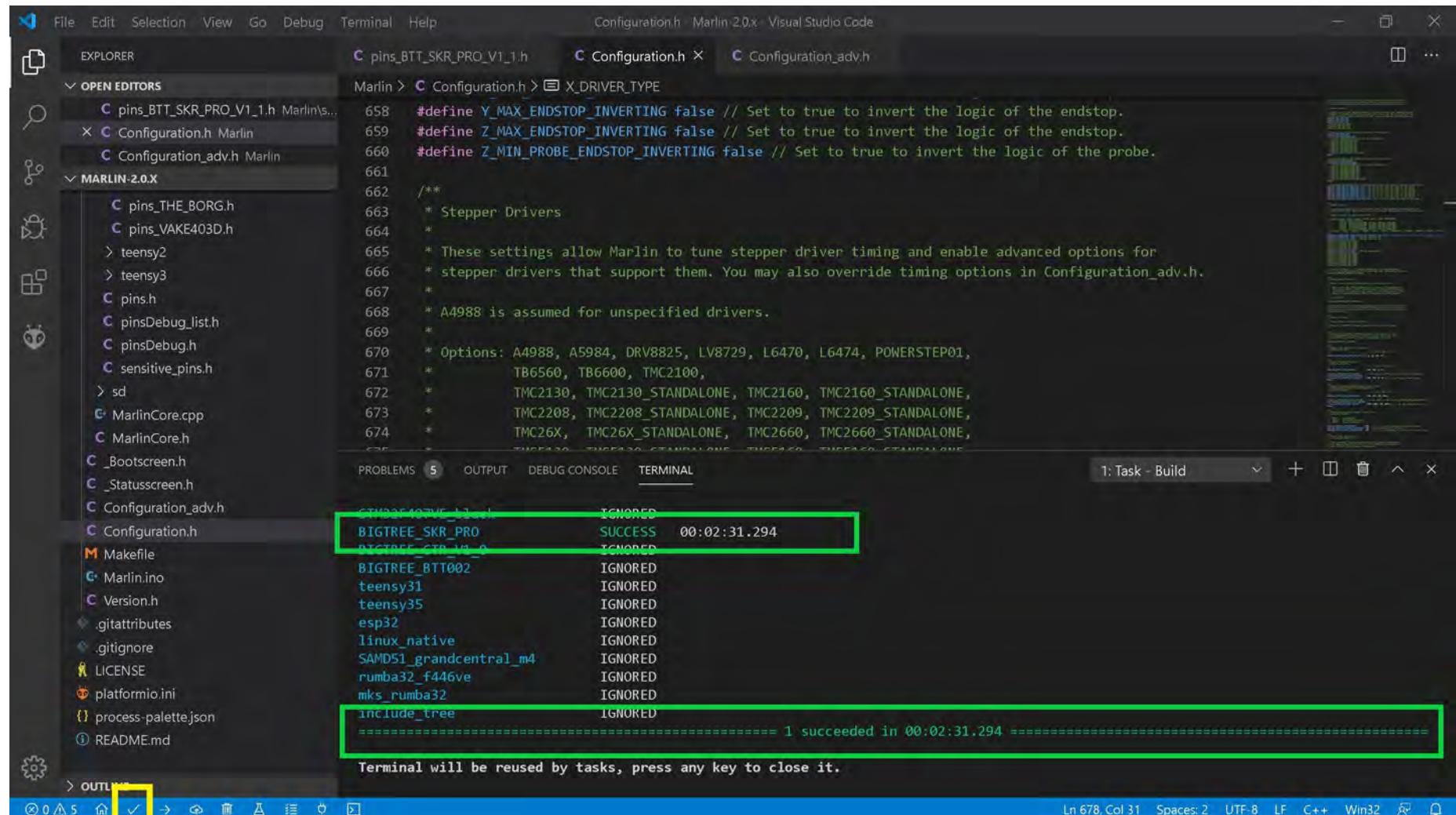
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
655 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
656 #define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



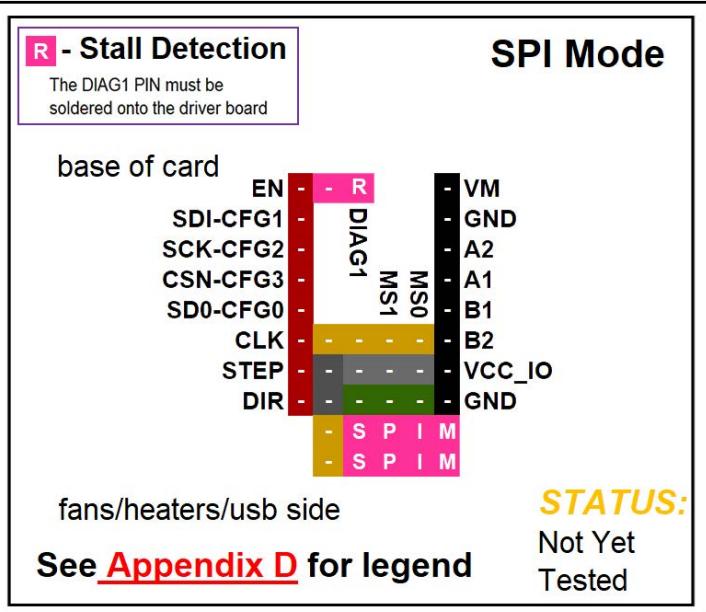
The screenshot shows the Visual Studio Code interface with the following details:

- 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 build 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".

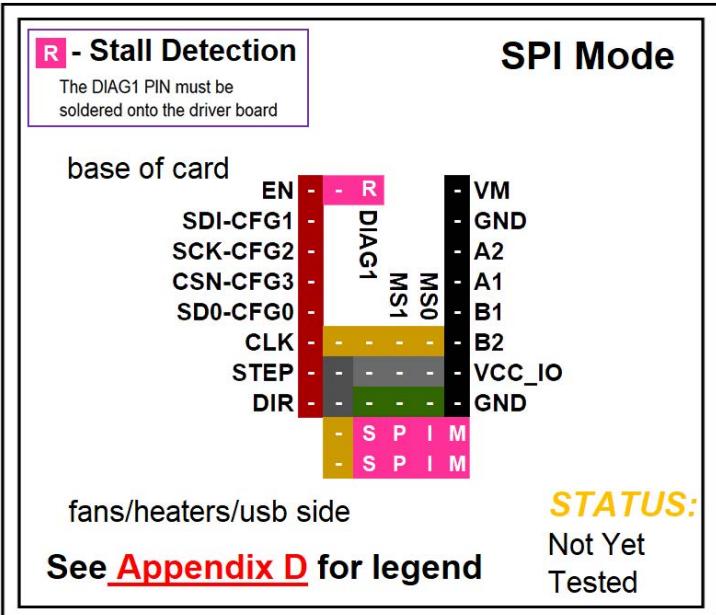
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

**BIQU TMC5160 V1.2**SPI Mode

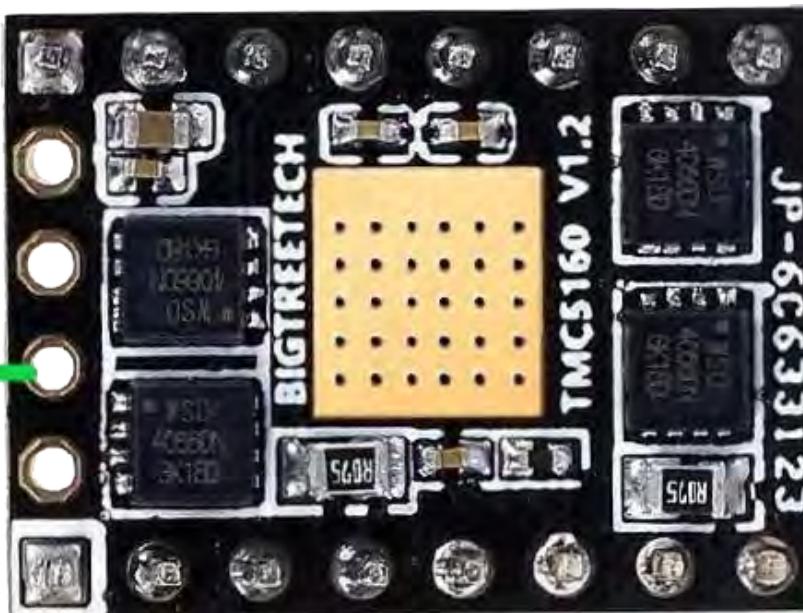
**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

<b>Driver Chip</b>  <b>TMC5160</b> SPI Mode Maximum 256 Subdivision 40V DC 4.3A (peak)	<b>Steps are set inside of your Firmware</b>
<b>Driving Current Calculation Formula</b> $R_S$ (Typical Sense Resistor) = 0.075 Ω	$I_{MAX} = 4.333$ See Appendix B #8. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90 = 3.900$
	Current Limit is set by the current sense resistors ( $R_s$ ). Use 50% - 90% of $I_{MAX}$ .

**BIQU TMC5160 V1.2**SPI Mode

**Note: The TMC5160 V1.2 by default comes in SPI mode. The BIQU TMC5160 does NOT come with a POT or "V<sub>ref</sub> 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

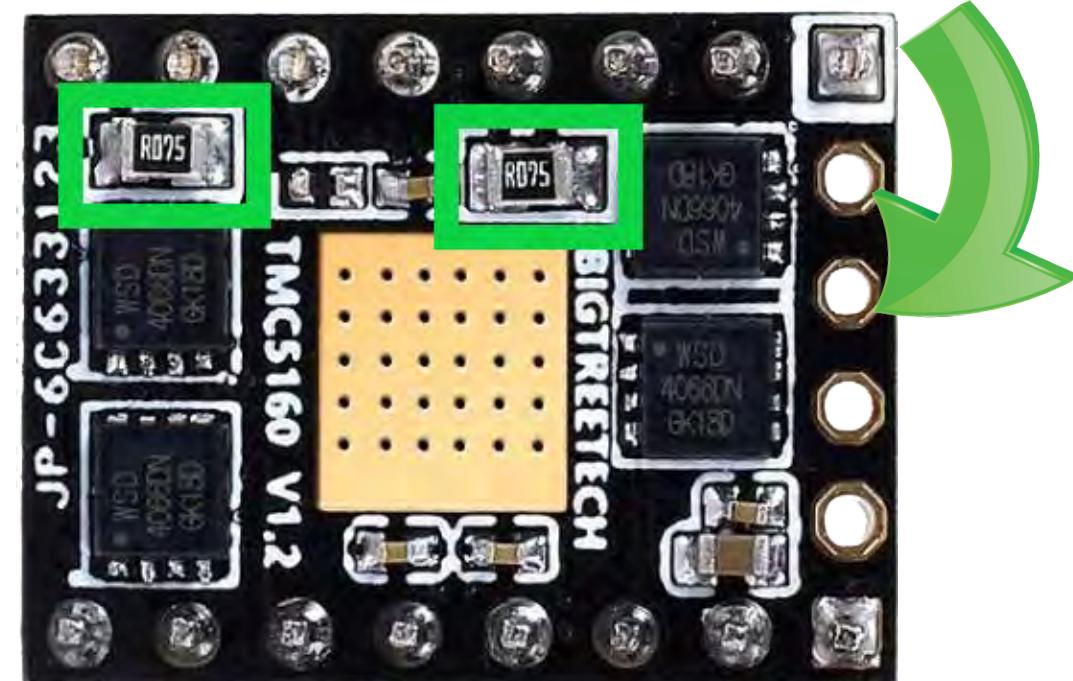
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

$R_s = R150$  is 0.15 Ohms

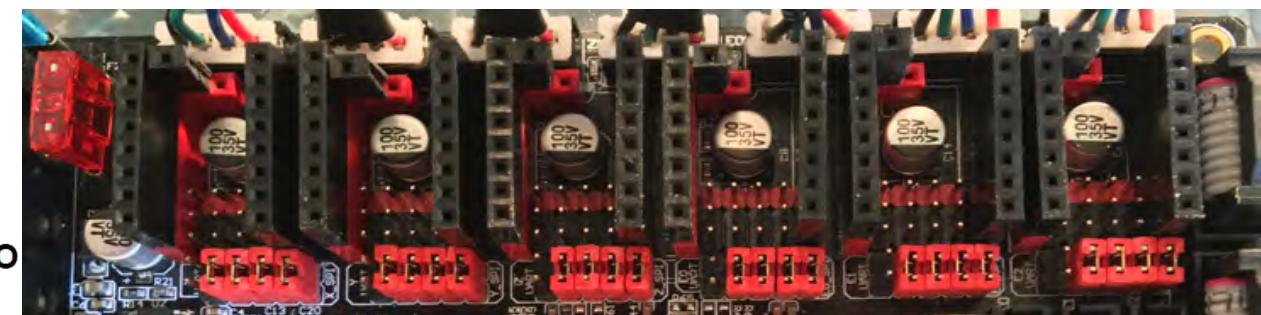
$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



EN	-	R	-	VM
SDI-CFG1	-	DIG1	-	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!!



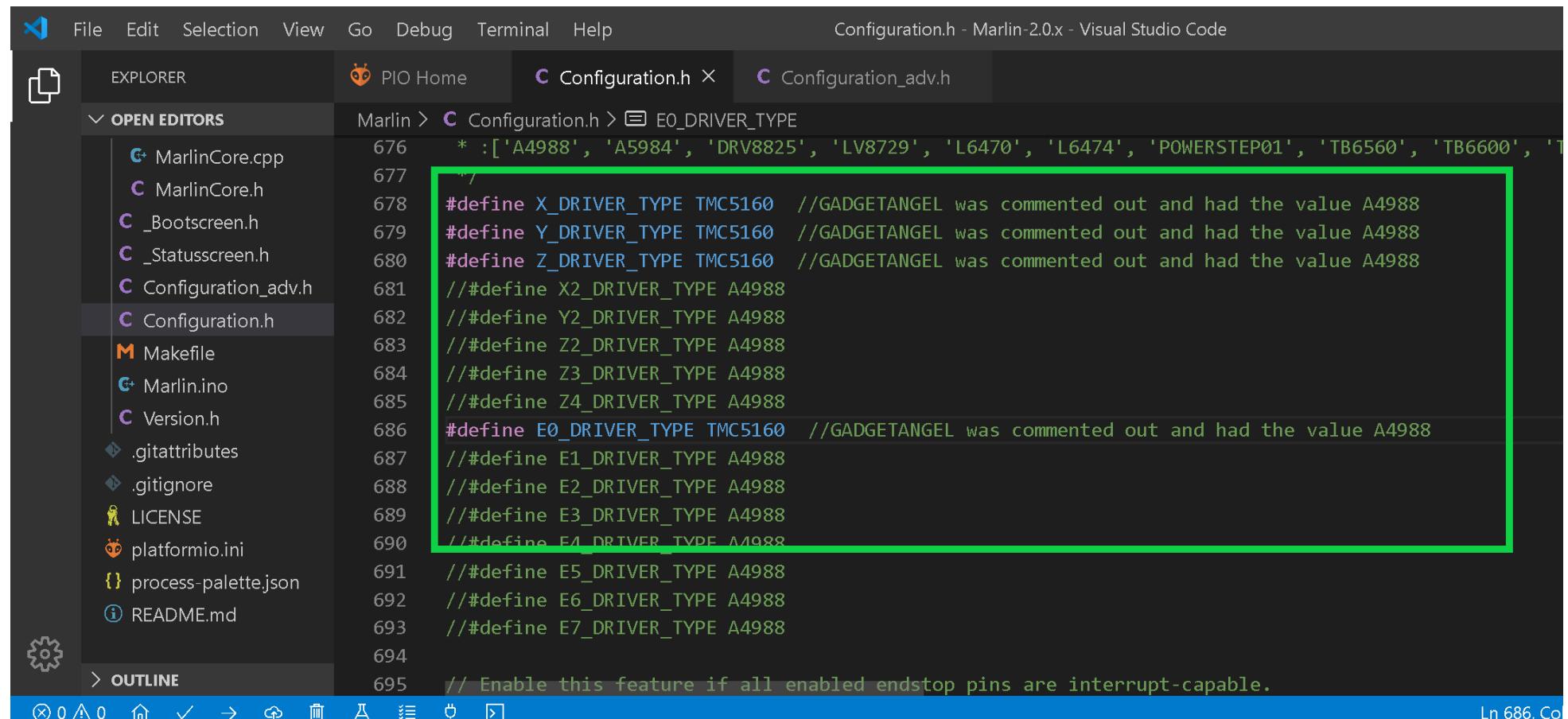
See [Appendix D](#) for legend

# 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 X 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

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 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

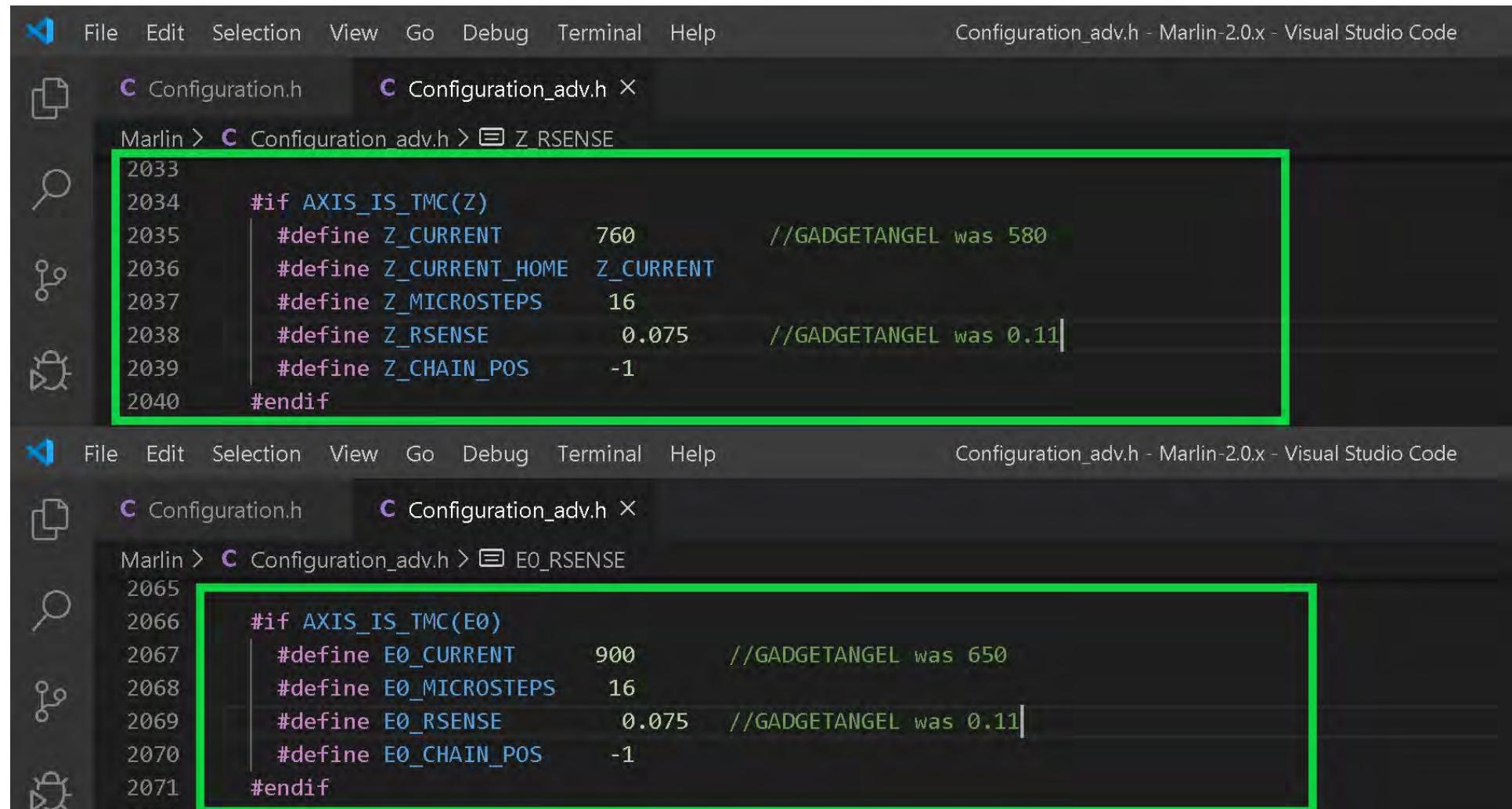
```

Ln 2006, Col 62 Spaces: 2 UTF-8 LF C++ Win32 ⚡ 🔍

- Go to the next page.

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

- Now, I am setting the  $V_{ref}$  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".



```

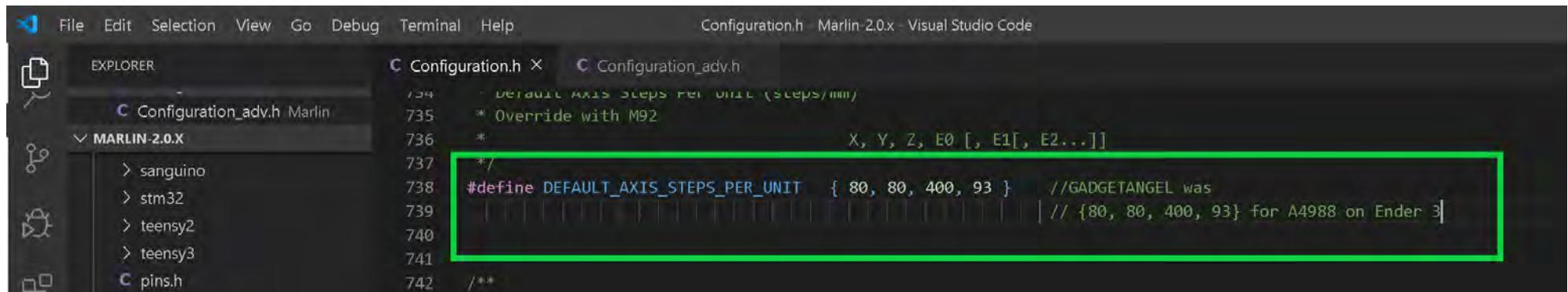
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035   #define Z_CURRENT      760           //GADGETANGEL was 580
2036   #define Z_CURRENT_HOME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

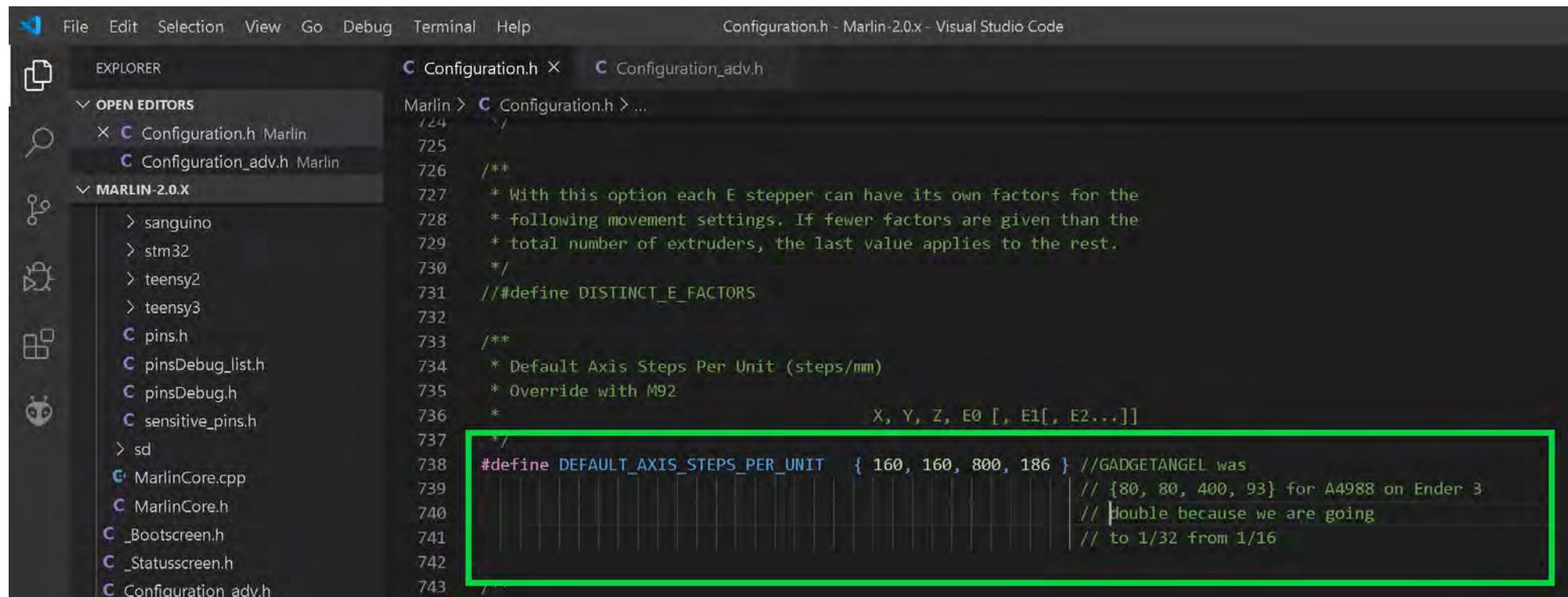
MARLIN-2.0.X

```

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

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 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

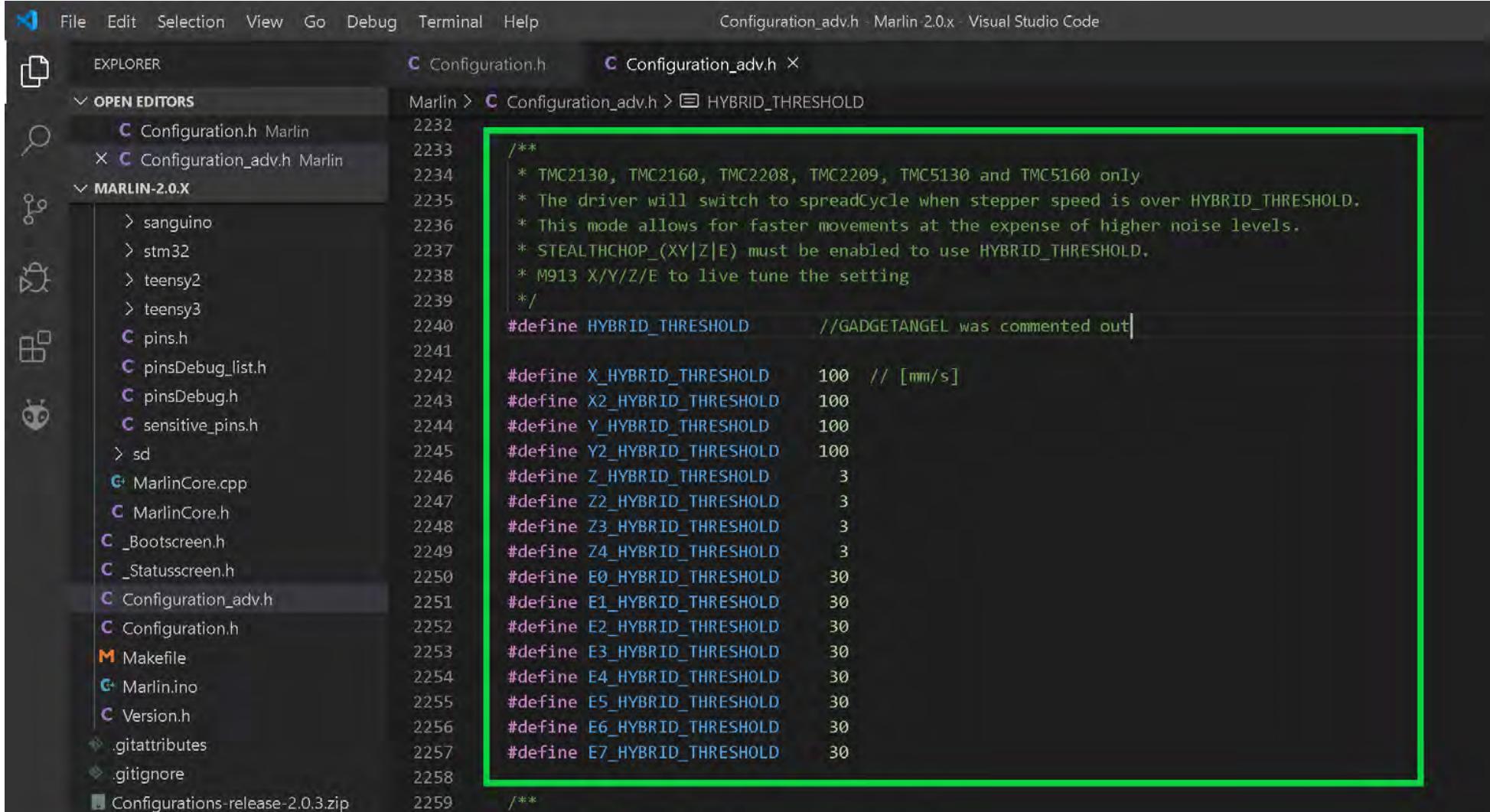
MARLIN-2.0.X sanguino stm32 teensy2 teensy3 pins.h pinsDebug\_list.h pinsDebug.h sensitive\_pins.h sd MarlinCore.cpp MarlinCore.h Bootscreen.h Statusscreen.h Configuration\_adv.h Configuration.h

```
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:

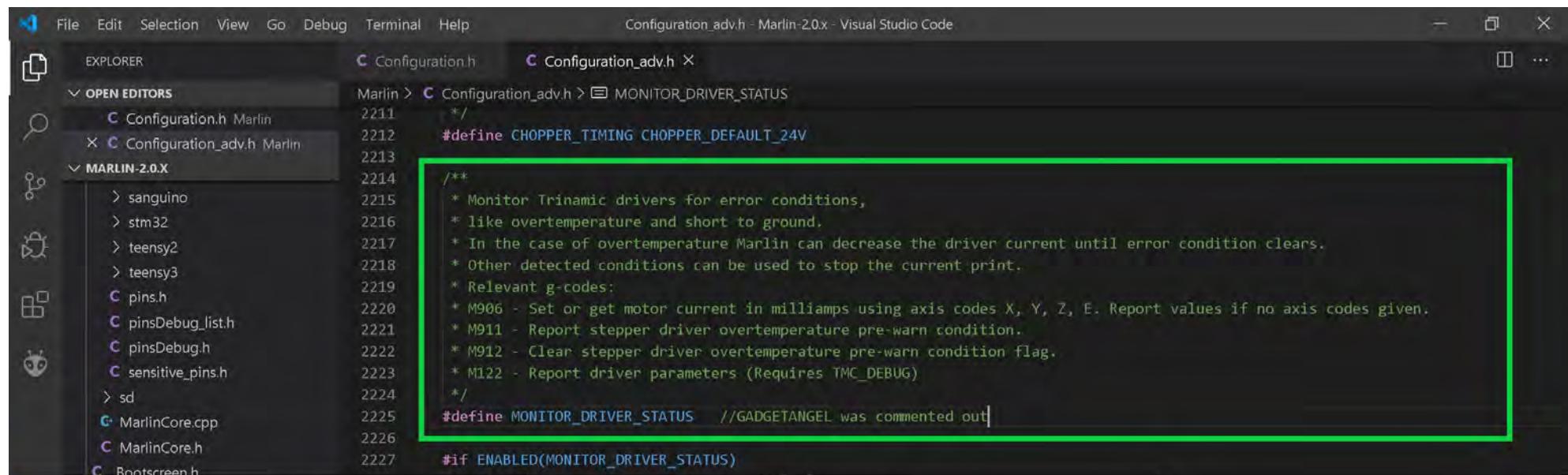
```
/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.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

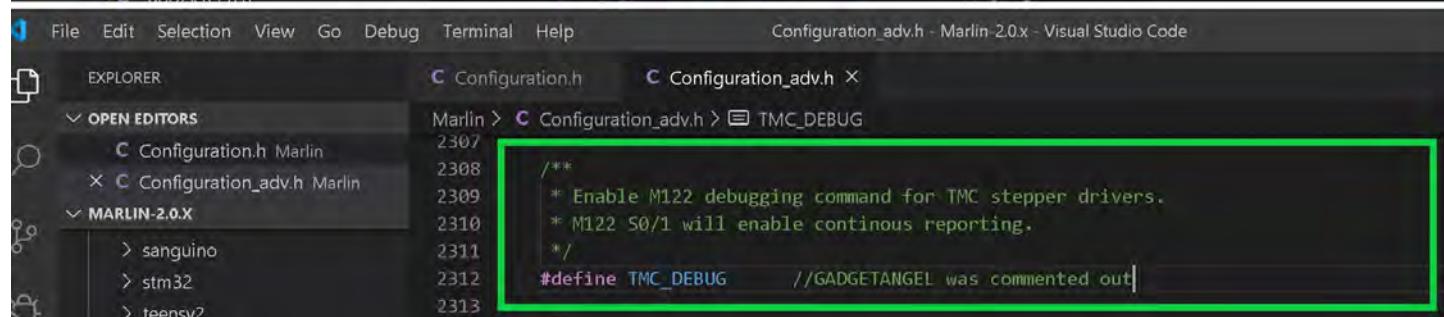
- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about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



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
  2211 */
  2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
  2213
  2214 /**
  2215 * Monitor Trinamic drivers for error conditions,
  2216 * like overtemperature and short to ground.
  2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
  2218 * Other detected conditions can be used to stop the current print.
  2219 * Relevant g-codes:
  2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
  2221 * M911 - Report stepper driver overtemperature pre-warn condition.
  2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
  2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
  2224 */
  2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
  2226
  2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



```

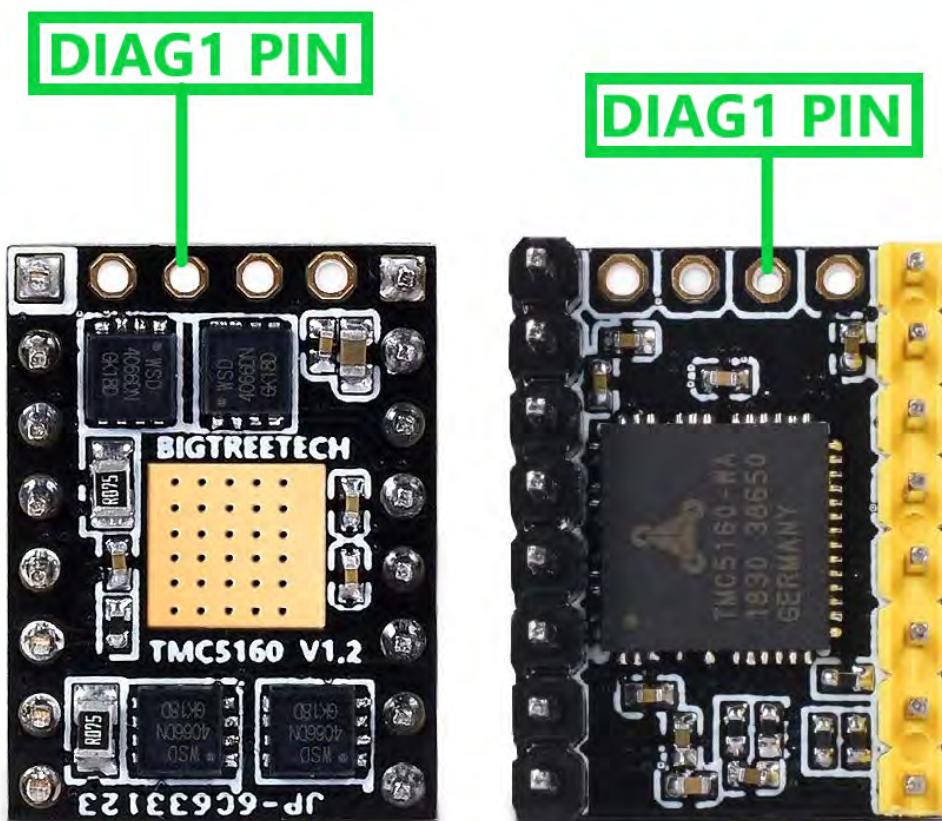
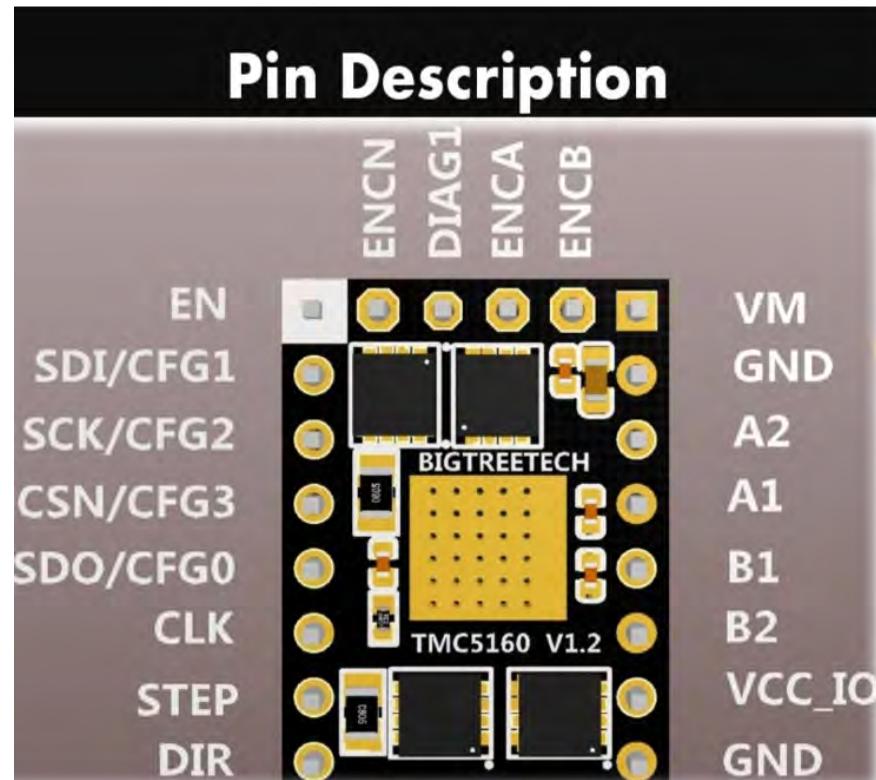
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG
  2307
  2308 /**
  2309 * Enable M122 debugging command for TMC stepper drivers.
  2310 * M122 S0/1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

```

- Go to the next page.

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

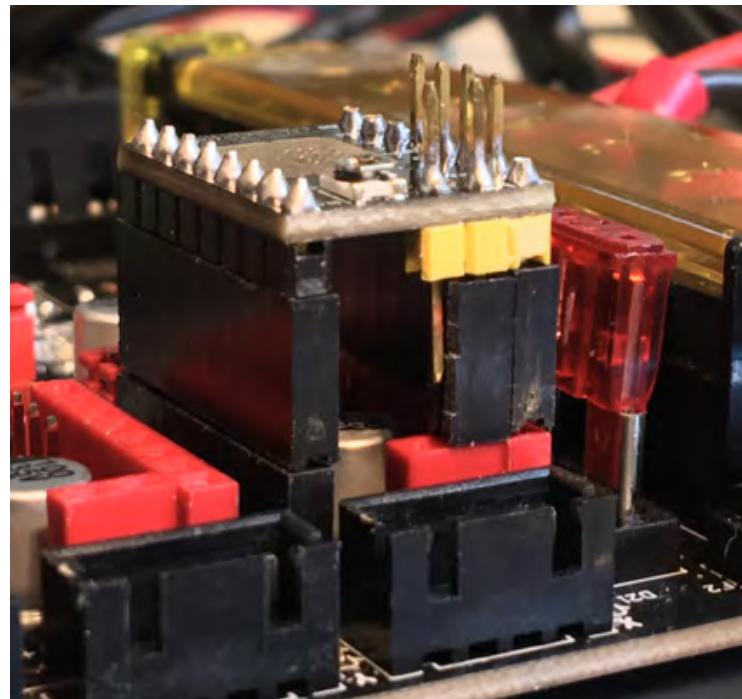
- This next section covers sensor-less homing which is available for the TMC5160 in SPI mode. I want to enable it BUT for the TMC5160 I first have to solder on the DIAG1 pin onto each TMC5160 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG1 pin for two TMC5160 drivers. Here is a picture of the TMC5160 V1.2 pin-out.



- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5160 into the SKR PRO V1.1 board the DIAG1 pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

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

- I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC5160 driver has to be plugged into the SKR PRO V1.1 board. Since I am **not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC516 is NOT connected to the board.** I plan to plug my Z axis' TMC5160 by using long stackable header pin risers, as seen in the picture below.



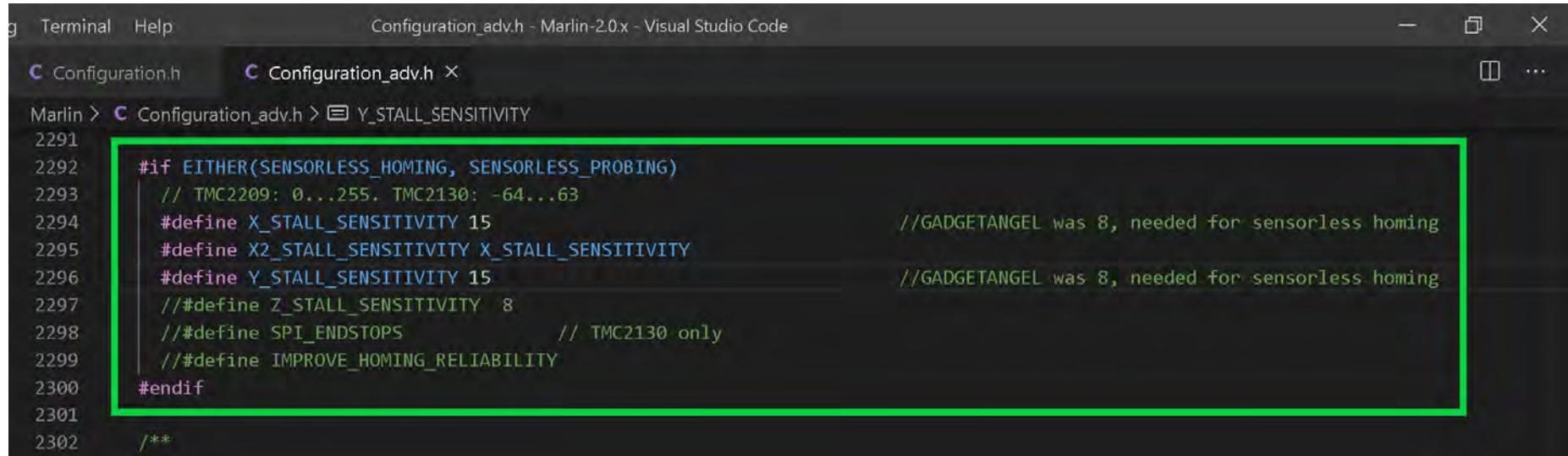
- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS\_HOMING"

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

- Go to the next page.

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

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

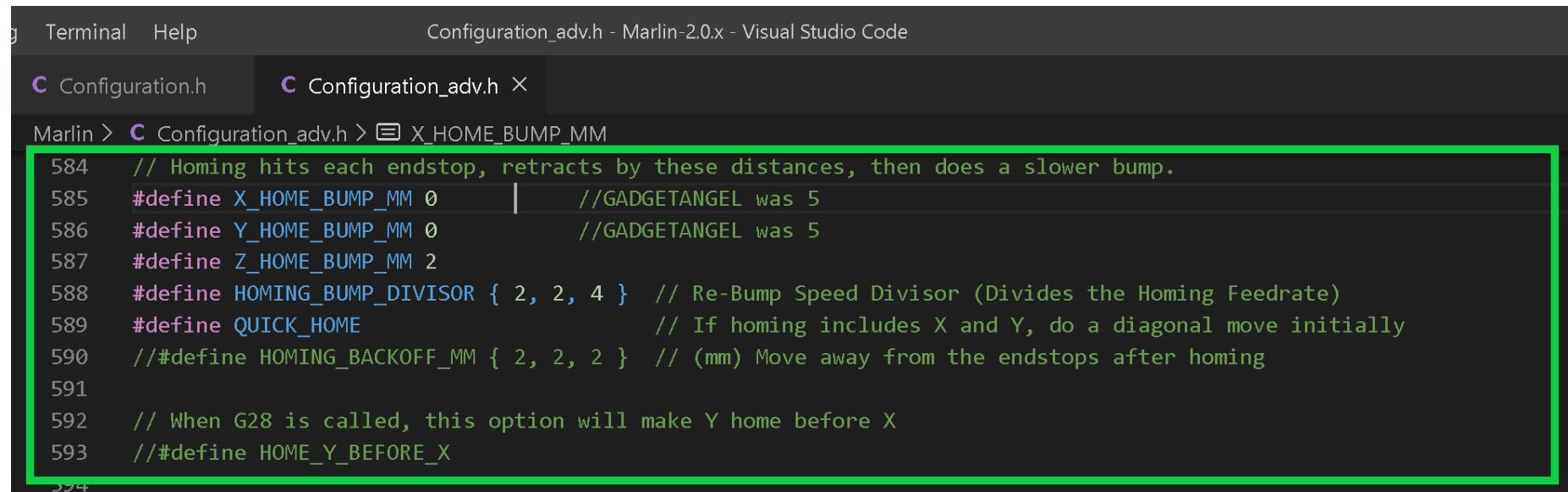


```

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

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X\_HOME\_BUMP\_MM" to 0 and "Y\_HOME\_BUMP\_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z\_HOME\_BUMP\_MM" to 0.



```

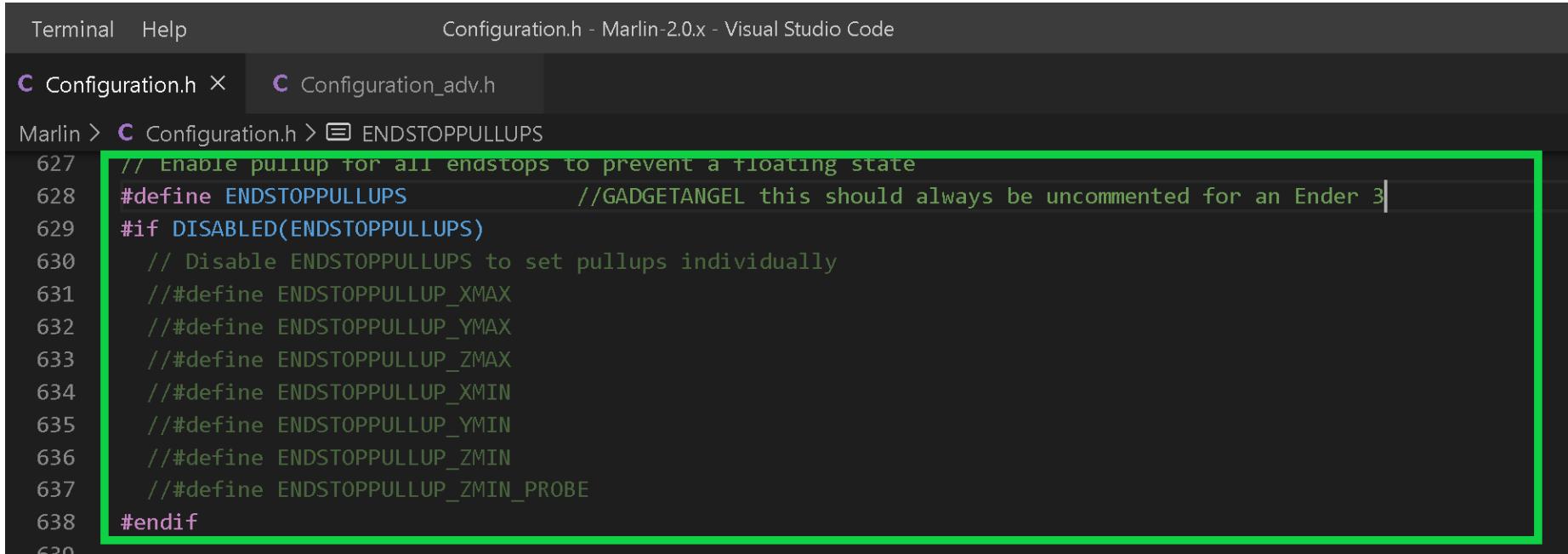
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL was 5
587 #define Z_HOME_BUMP_MM 2
588 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
589 #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
590 //">#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
591
592 // When G28 is called, this option will make Y home before X
593 //">#define HOME_Y_BEFORE_X
594

```

- Go to the next page.

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

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



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

Configuration.h X Configuration\_adv.h

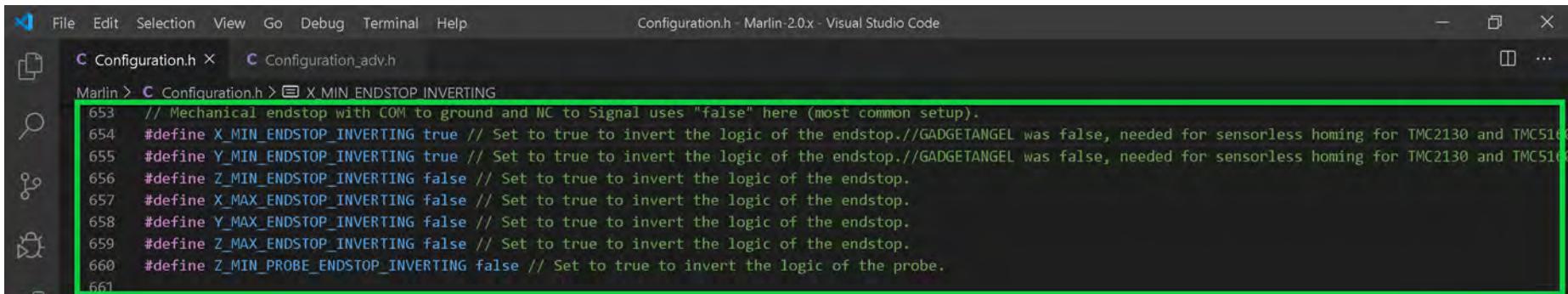
Marlin > Configuration.h > ENDSTOPPULLUPS

```

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630     // Disable ENDSTOPPULLUPS to set pullups individually
631     //#define ENDSTOPPULLUP_XMAX
632     //#define ENDSTOPPULLUP_YMAX
633     //#define ENDSTOPPULLUP_ZMAX
634     //#define ENDSTOPPULLUP_XMIN
635     //#define ENDSTOPPULLUP_YMIN
636     //#define ENDSTOPPULLUP_ZMIN
637     //#define ENDSTOPPULLUP_ZMIN_PROBE
638 #endif
639

```

- Next to allow sensor-less homing to work (while using the BIQU TMC5160) we need to change our end stop logic. Therefore I set "X\_MIN\_ENDSTOP\_INVERTING" to true and "Y\_MIN\_ENSTOP\_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z\_MIN\_ENDSTOP\_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z\_MIN\_ENDSTOP\_INVERTING" set to false.



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

Configuration.h X Configuration\_adv.h

Marlin > Configuration.h > X MIN ENDSTOP INVERTING

```

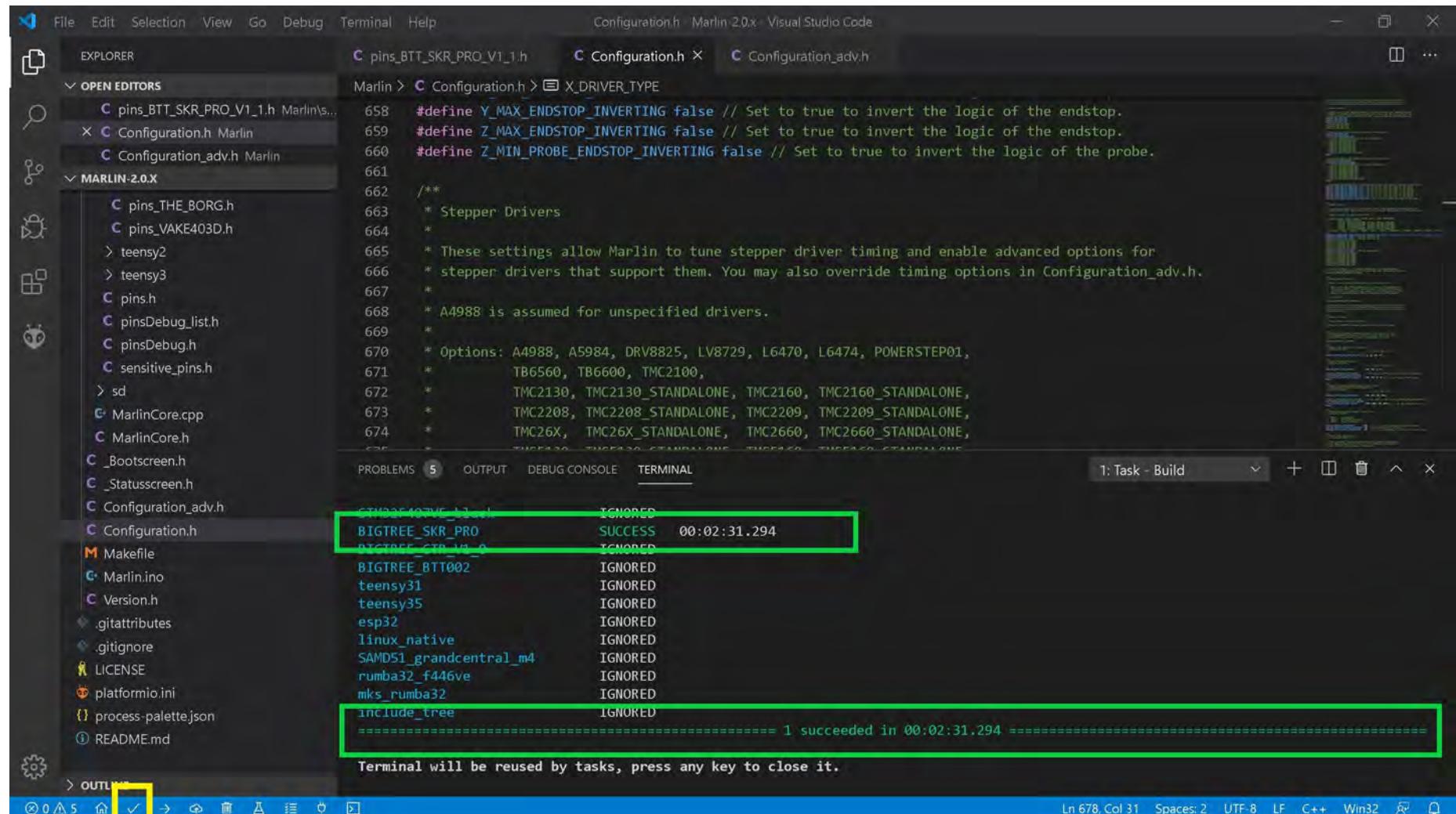
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661

```

- Go to the next page.

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

- The end of Marlin setup for BIQU TMC5160 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

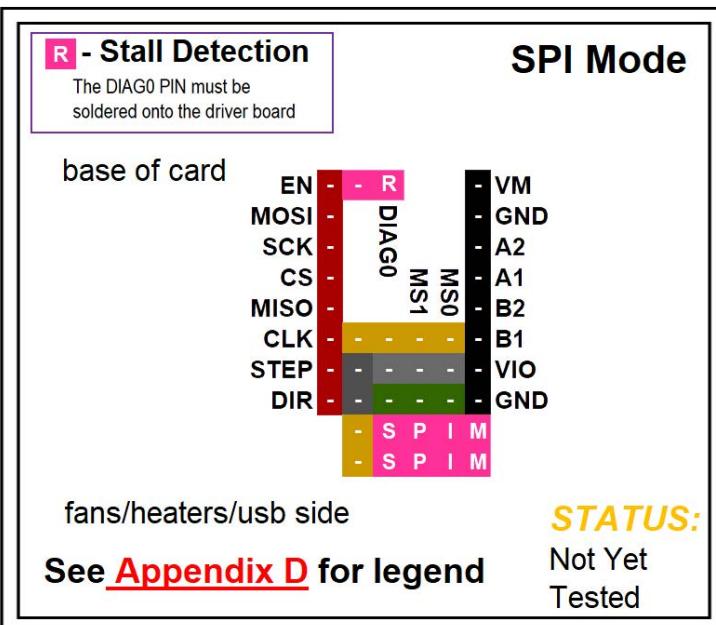


```

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

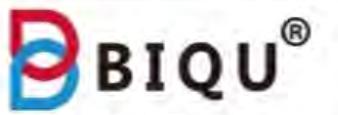
```

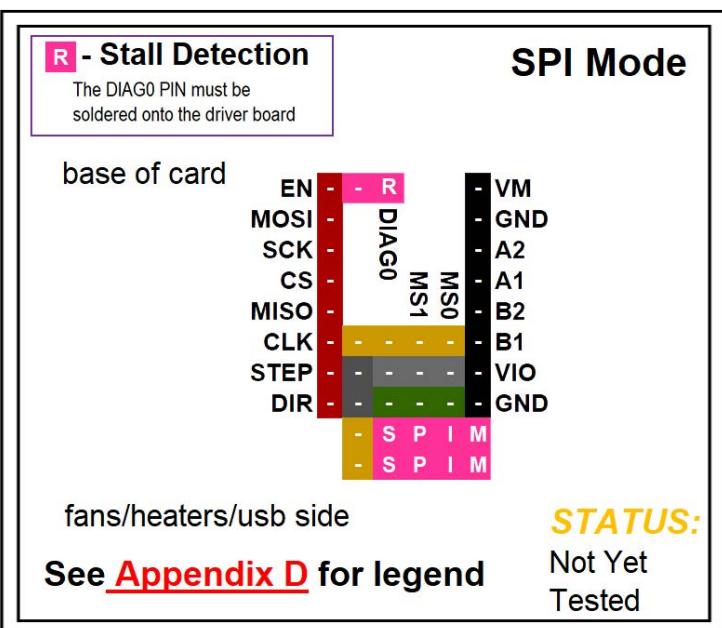
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

<b>Driver Chip</b>   <b>TMC5161</b>  SPI Mode Maximum 256 Subdivision 40V DC 5.2A (peak)	<b>Steps are set inside of your Firmware</b>	
<b>Driving Current Calculation Formula</b>  $R_S$ (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 <b>50% - 90% of <math>I_{MAX}</math></b> .

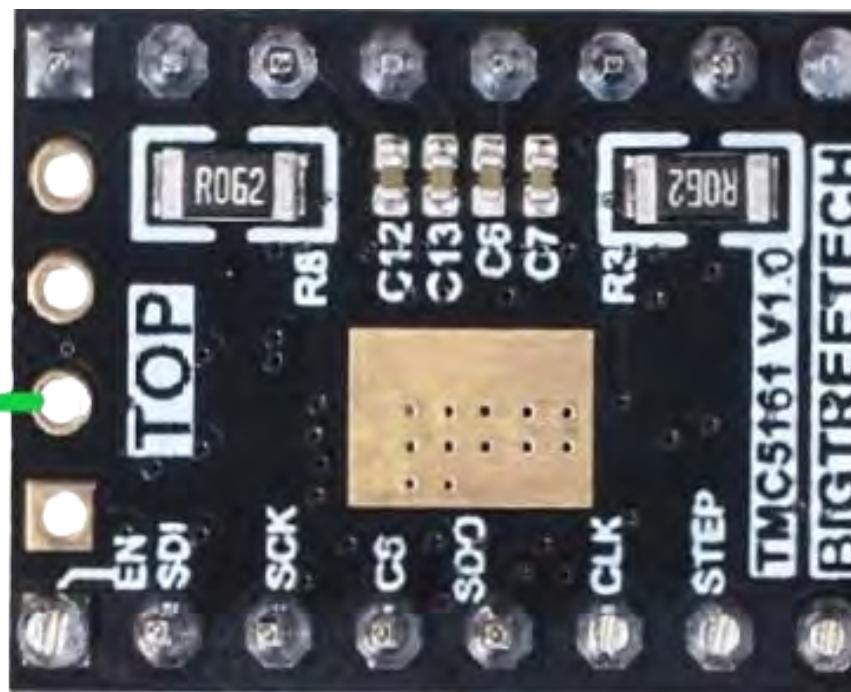


**DIAGO**

## 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<sub>ref</sub> 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

## BIGTREE 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

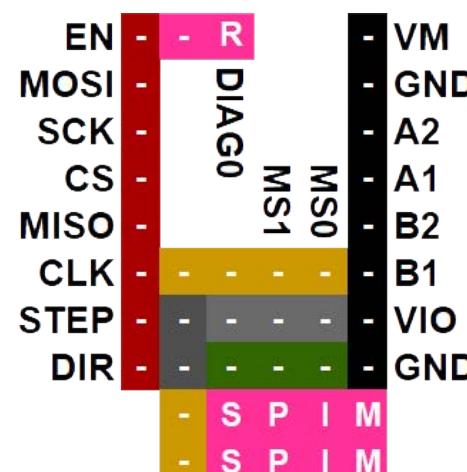
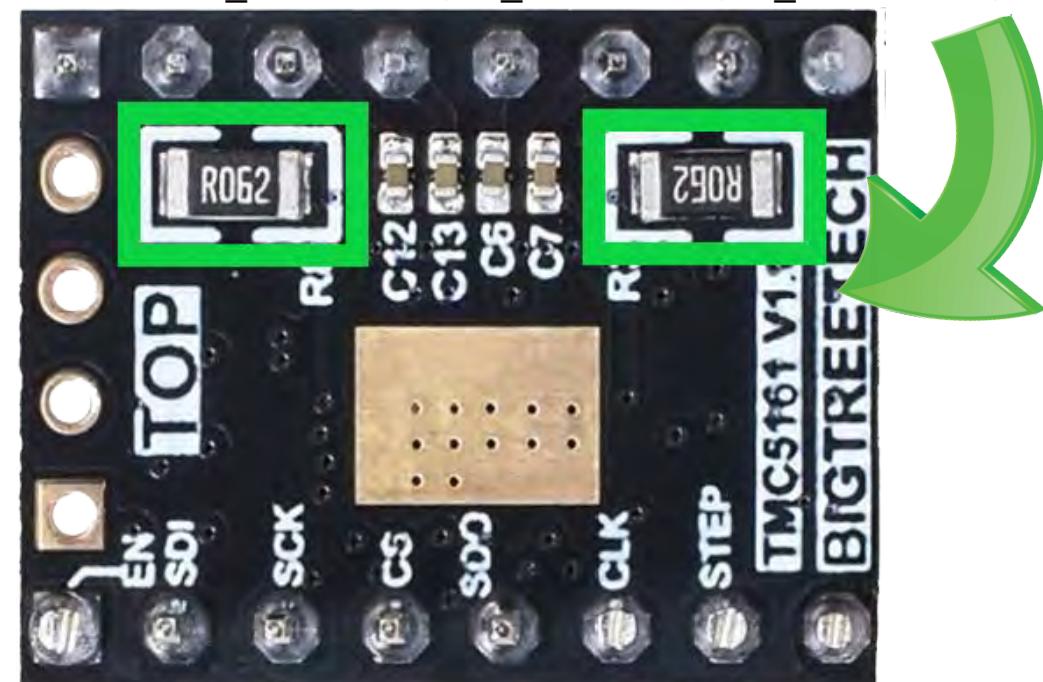
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

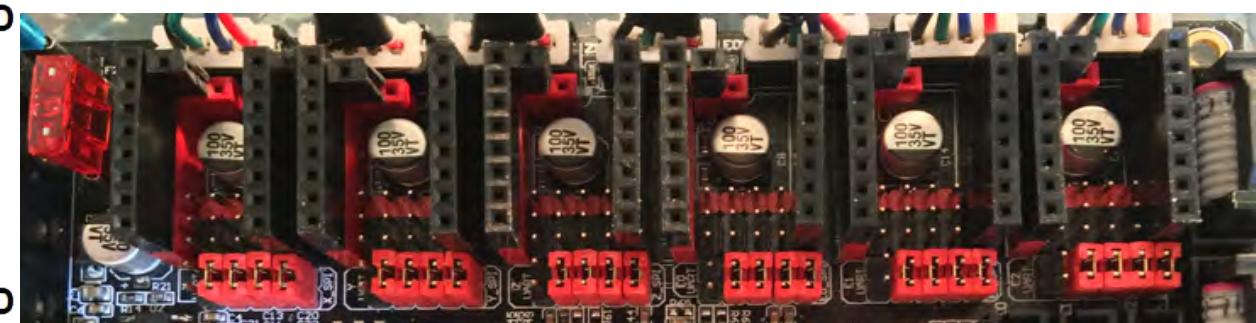
$R_s = R150$  is 0.15 Ohms

$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



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

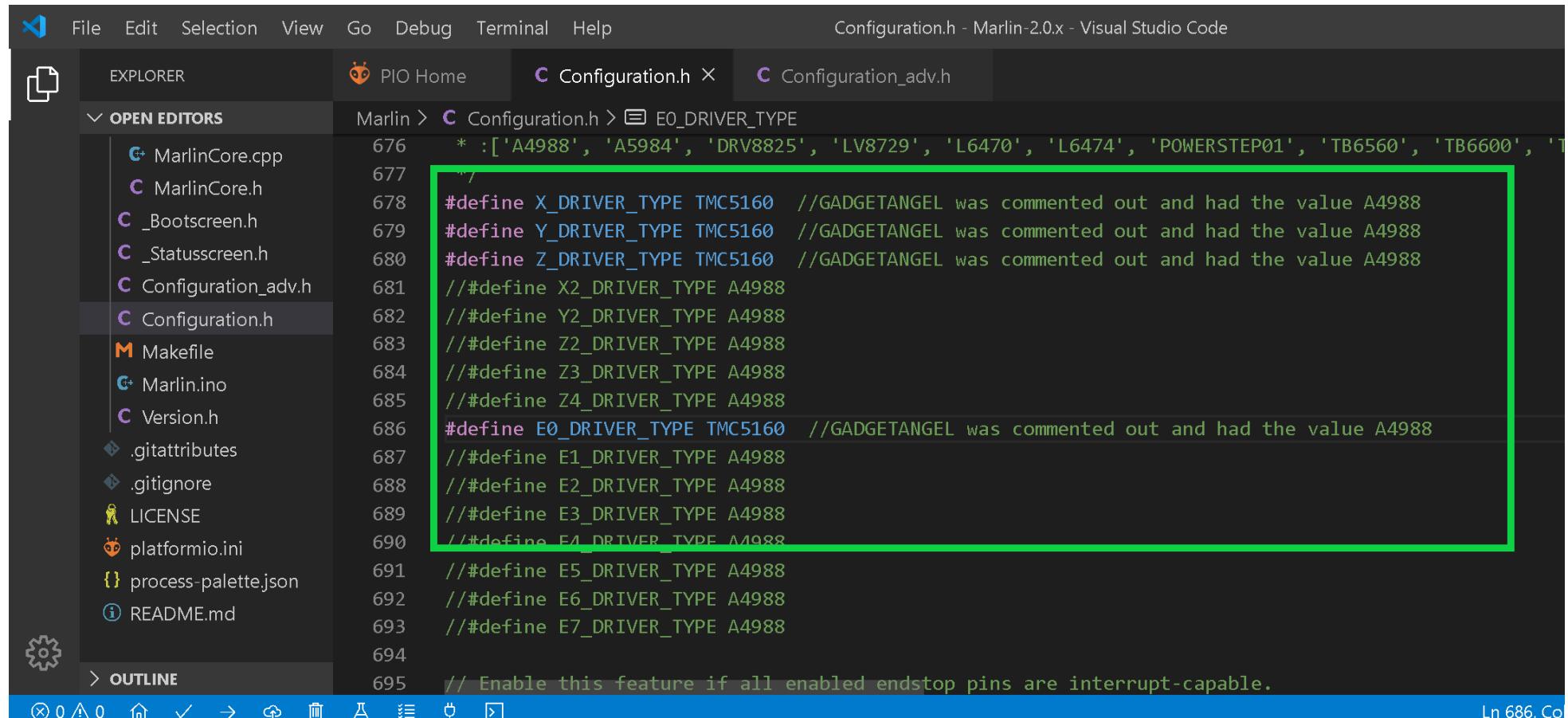


See [Appendix D](#) for legend

## 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

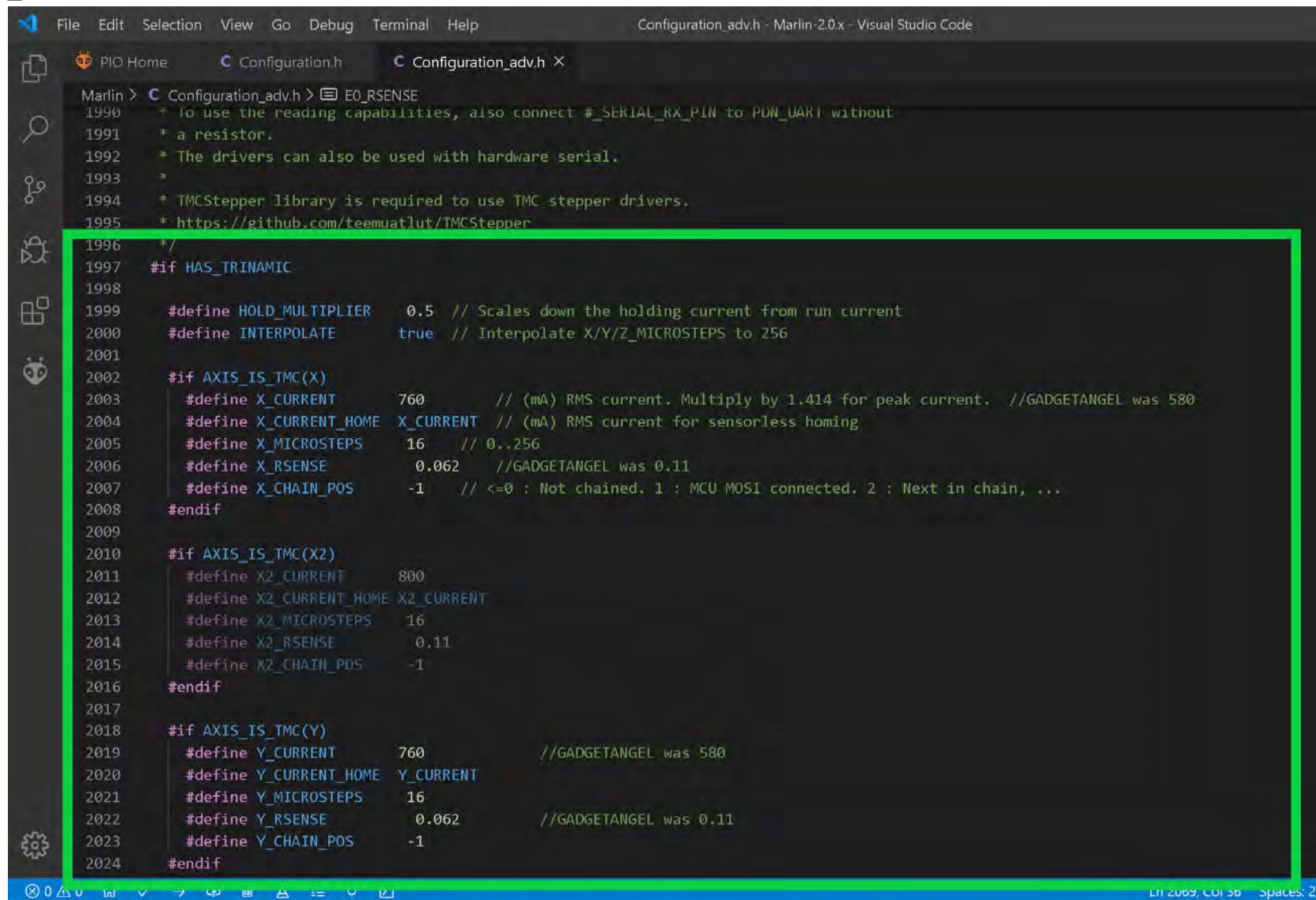
EXPLORER PIO Home Configuration.h X Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'T
677 "
678 #define X_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
679 #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

```

```

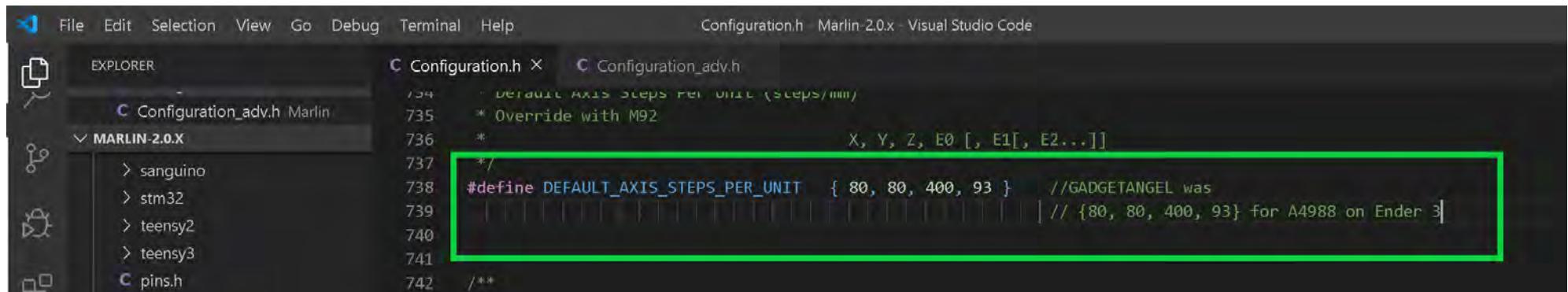
2066 #if AXIS_IS_TMC(E0)
2067     #define E0_CURRENT      900          //GADGETANGEL was 650
2068     #define E0_MICROSTEPS   16
2069     #define E0_RSENSE        0.062        //GADGETANGEL was 0.11
2070     #define E0_CHAIN_POS    -1
2071 #endif

```

- 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

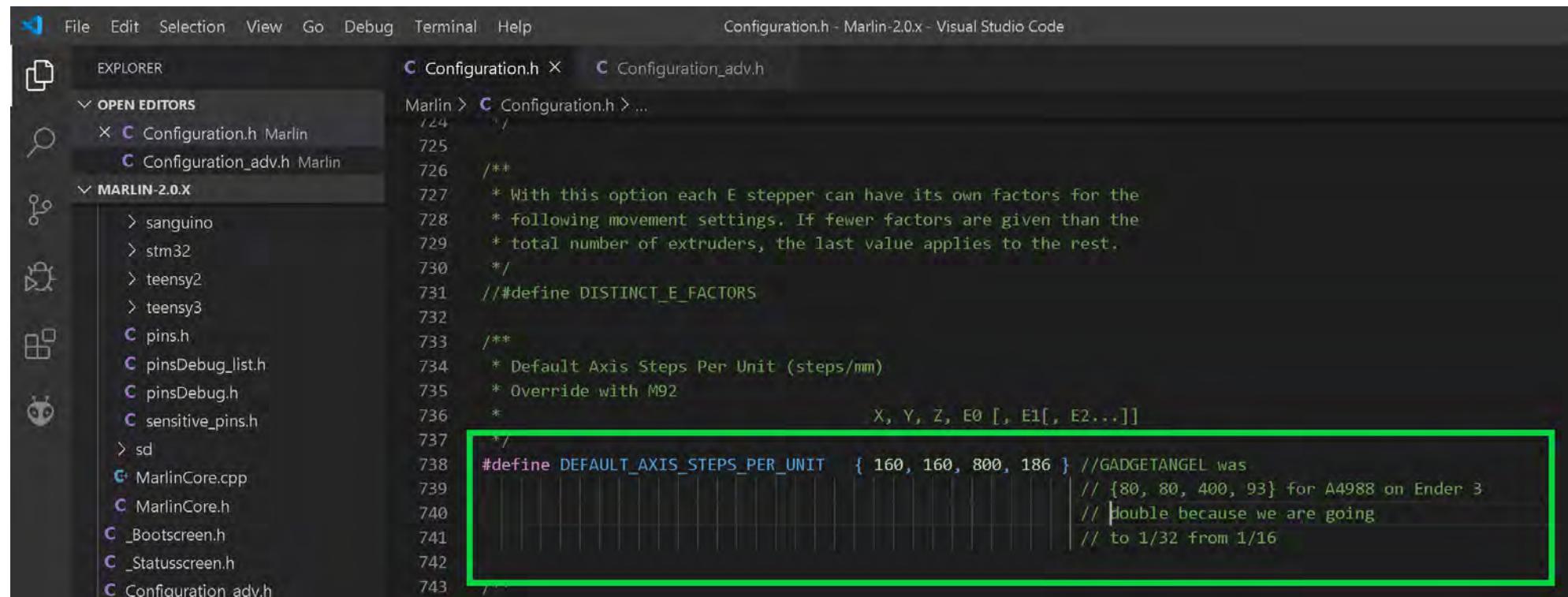
MARLIN-2.0.X

```

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

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



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

EXPLORER Configuration.h X Configuration\_adv.h

OPEN EDITORS Configuration.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X

```

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

```

- Go to the next page.

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

- We need to uncomment out the "TMC\_USE\_SW\_SPI" because the SKR PRO V1.1 pins file depends on this variable to define its SPI pins

```

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

EXPLORER Configuration.h Configuration_adv.h
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.

```

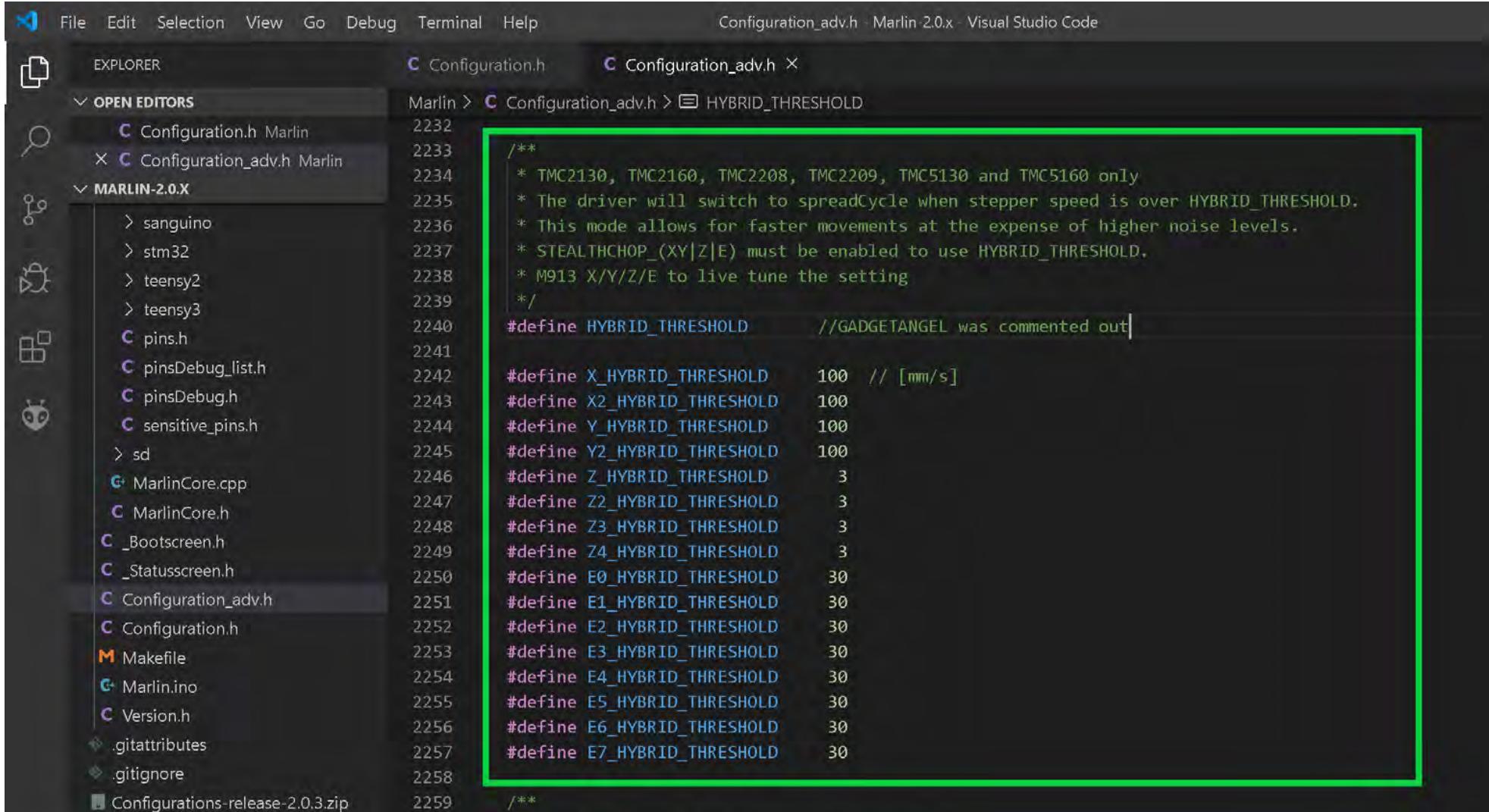
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

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.



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 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, and Configurations-release-2.0.3.zip.
- Editor:** The Configuration\_adv.h file is open, specifically at line 2232. A green box highlights the HYBRID\_THRESHOLD section. The code is as follows:

```
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 #define X_HYBRID_THRESHOLD 100 // [mm/s]
2241 #define X2_HYBRID_THRESHOLD 100
2242 #define Y_HYBRID_THRESHOLD 100
2243 #define Y2_HYBRID_THRESHOLD 100
2244 #define Z_HYBRID_THRESHOLD 3
2245 #define Z2_HYBRID_THRESHOLD 3
2246 #define Z3_HYBRID_THRESHOLD 3
2247 #define Z4_HYBRID_THRESHOLD 3
2248 #define E0_HYBRID_THRESHOLD 30
2249 #define E1_HYBRID_THRESHOLD 30
2250 #define E2_HYBRID_THRESHOLD 30
2251 #define E3_HYBRID_THRESHOLD 30
2252 #define E4_HYBRID_THRESHOLD 30
2253 #define E5_HYBRID_THRESHOLD 30
2254 #define E6_HYBRID_THRESHOLD 30
2255 #define E7_HYBRID_THRESHOLD 30
2256
2257 /**
2258 * H S T D L C R Y Z E
2259 */
```

- Go to the next page.

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

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

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
  2211 */
  2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
  2213
  2214 /**
  2215 * Monitor Trinamic drivers for error conditions,
  2216 * like overtemperature and short to ground.
  2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
  2218 * Other detected conditions can be used to stop the current print.
  2219 * Relevant g-codes:
  2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
  2221 * M911 - Report stepper driver overtemperature pre-warn condition.
  2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
  2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
  2224 */
  2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
  2226
  2227 #if ENABLED(MONITOR_DRIVER_STATUS)
  2228
  2229
  2230
  2231
  2232
  2233
  2234
  2235
  2236
  2237
  2238
  2239
  2240
  2241
  2242
  2243
  2244
  2245
  2246
  2247
  2248
  2249
  2250
  2251
  2252
  2253
  2254
  2255
  2256
  2257
  2258
  2259
  2260
  2261
  2262
  2263
  2264
  2265
  2266
  2267
  2268
  2269
  2270
  2271
  2272
  2273
  2274
  2275
  2276
  2277
  2278
  2279
  2280
  2281
  2282
  2283
  2284
  2285
  2286
  2287
  2288
  2289
  2290
  2291
  2292
  2293
  2294
  2295
  2296
  2297
  2298
  2299
  2300
  2301
  2302
  2303
  2304
  2305
  2306
  2307
  2308
  2309
  2310
  2311
  2312
  2313

```

```

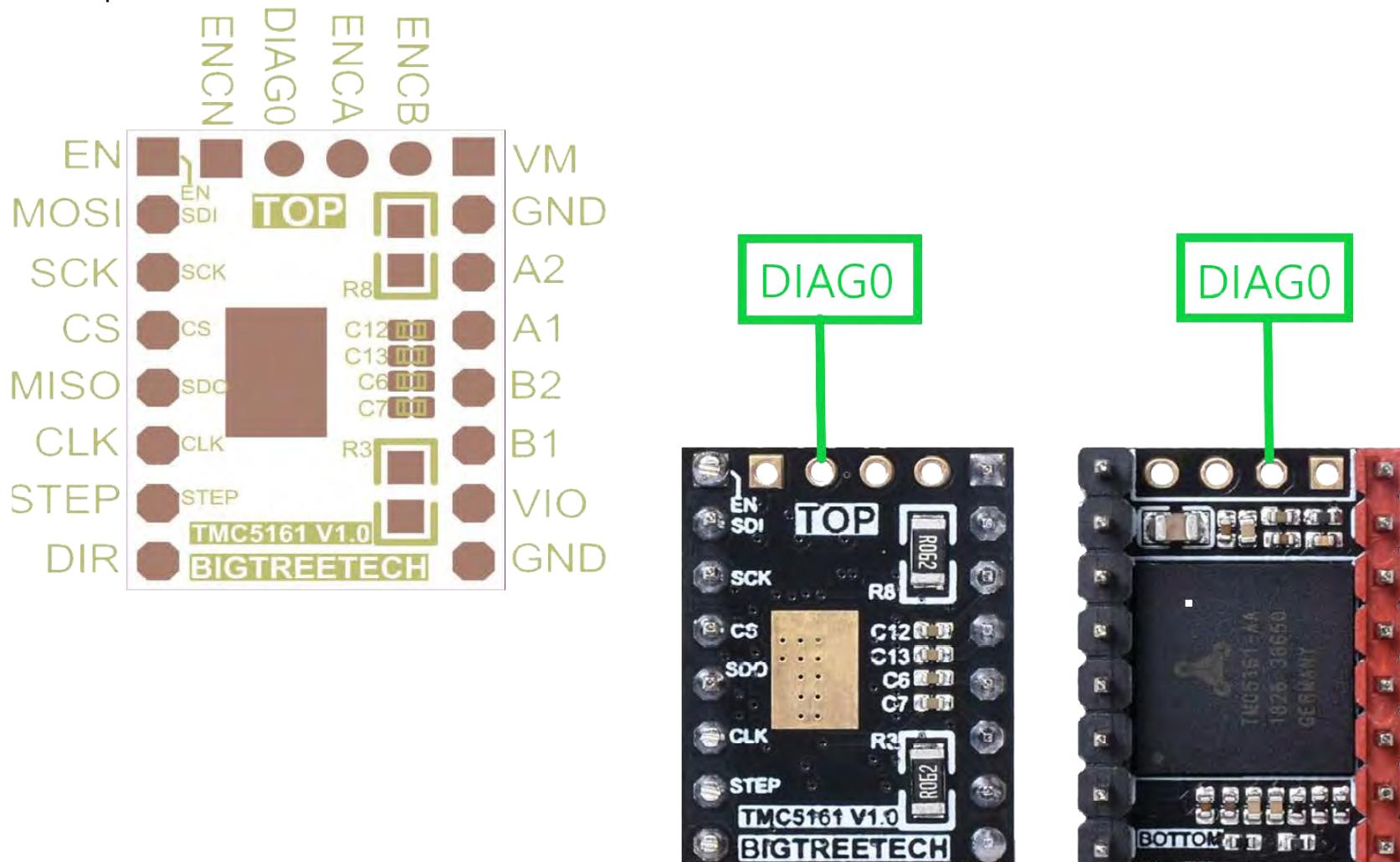
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG
  2307
  2308
  2309 /**
  2310 * Enable M122 debugging command for TMC stepper drivers.
  2311 *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

```

- Go to the next page.

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

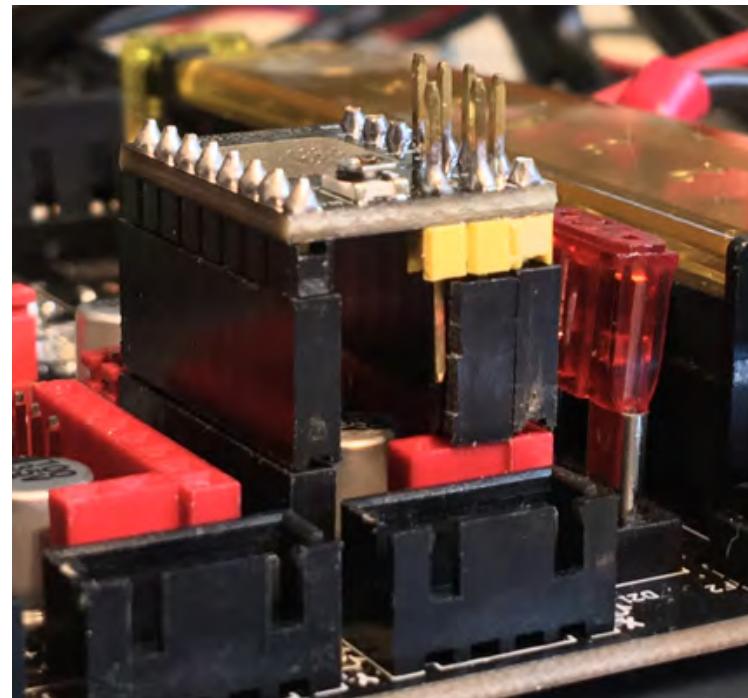
- This next section covers sensor-less homing which is available for the TMC5161 in SPI mode. I want to enable it BUT for the TMC5161 I first have to solder on the DIAG0 pin onto each TMC5161 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG0 pin for two TMC5161 drivers. Here is a picture of the TMC5161 V1.0 pin-out.



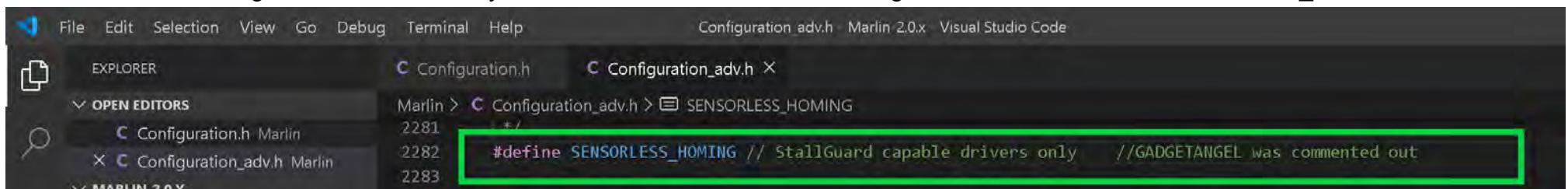
- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR PRO V1.1 board the DIAG0 pin will be seated in the SKR PRO V1.1 board.
- Go to the next page.

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

- I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG0 pin on the TMC5161 driver has to be plugged into the SKR PRO V1.1 board. Since I am **not using sensor-less homing on my Z axis I will need to ensure that my DIAG0 pin on the Z axis TMC5161 is NOT connected to the board.** I plan to plug my Z axis' TMC5161 by using long stackable header pin risers, as seen in the picture below.



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



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

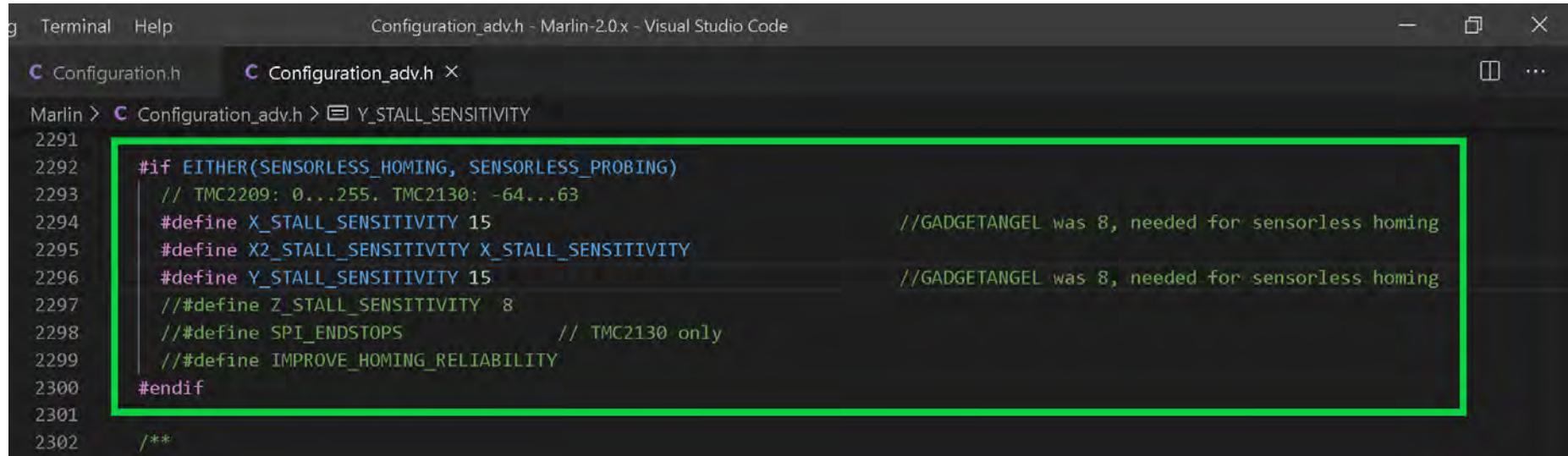
EXPLORER Configuration.h Configuration_adv.h ×
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X Configuration.h Marlin Configuration_adv.h Marlin

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".

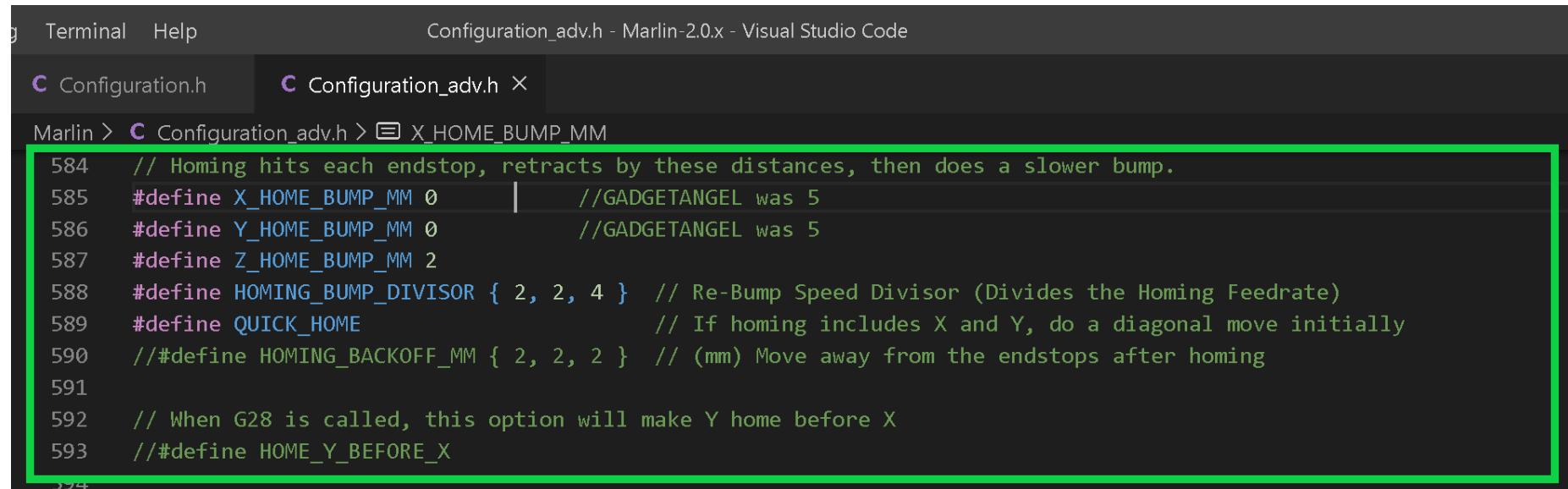


```

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.



```

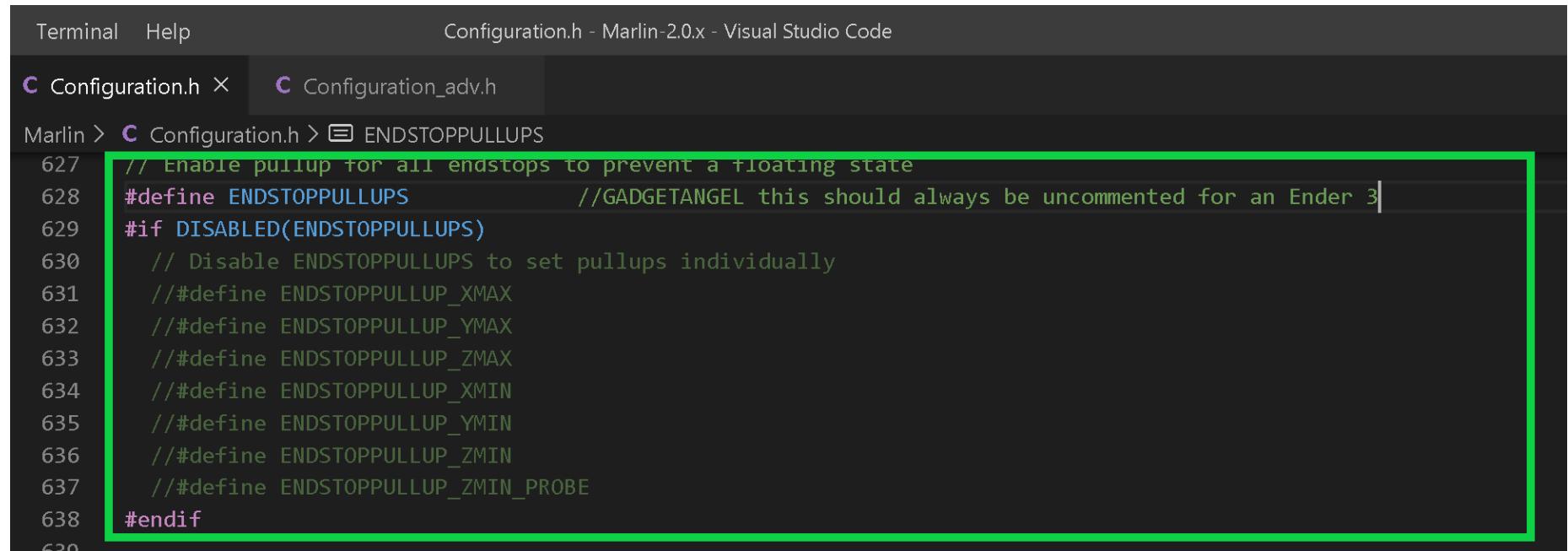
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

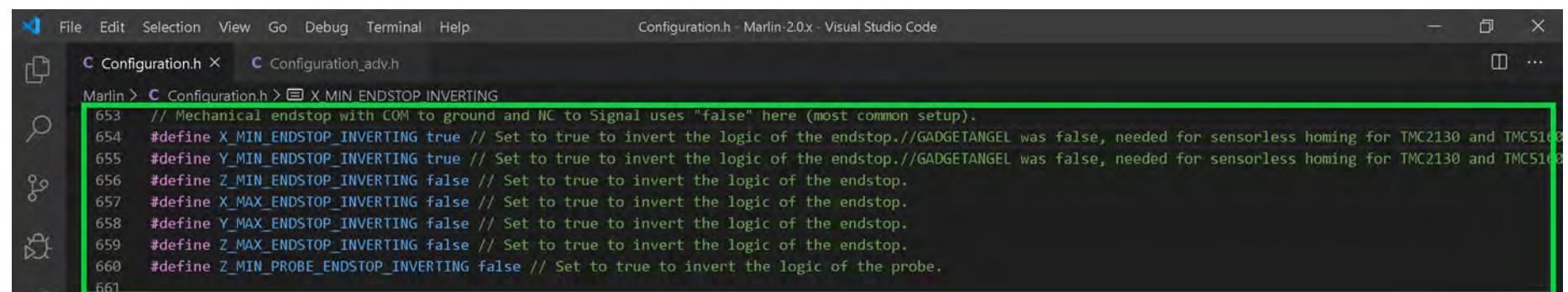
C Configuration.h X C Configuration_adv.h

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

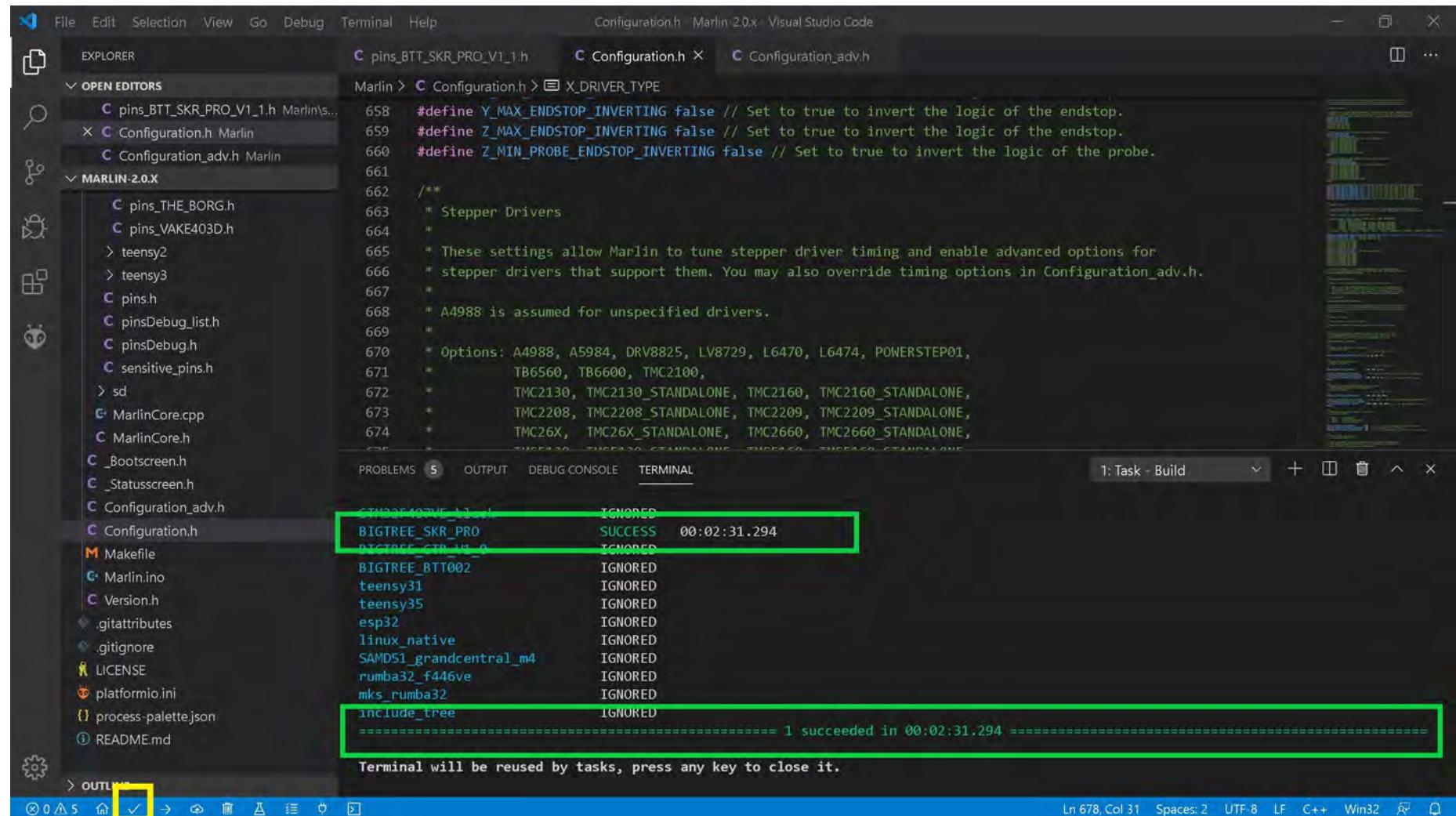
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
655 #define Y_MIN_ENDSTOP_INVERTING 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
CTM2208_STANDALONE IGNORED
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<sup>1, 2</sup>

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 Drivers<sup>2</sup>

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.

<sup>1</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers> and

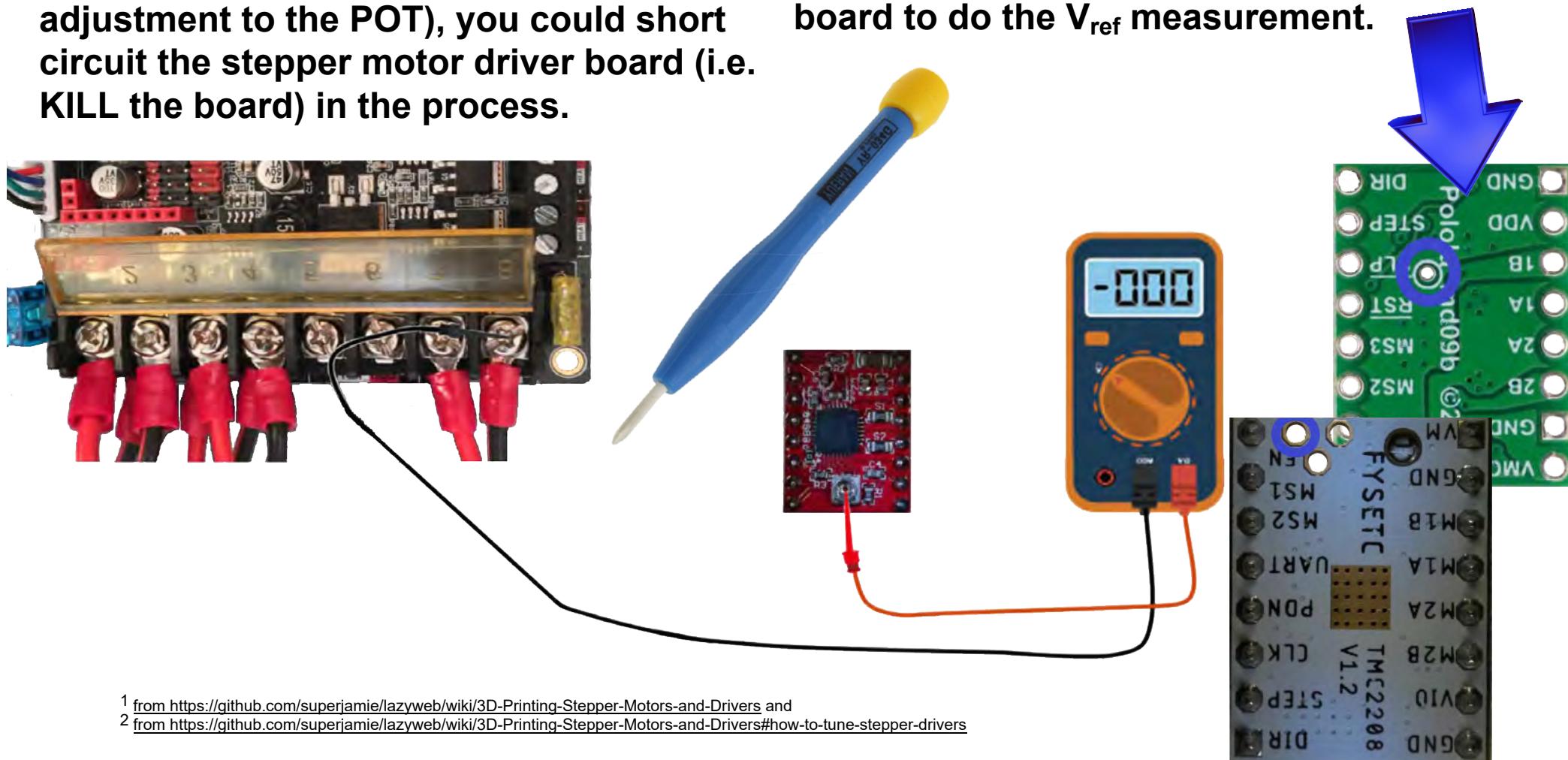
<sup>2</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

## How to adjust the $V_{ref}$ on a Stepper Motor Driver board using the Potentiometer<sup>1, 2</sup>

**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.



<sup>1</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers>

<sup>2</sup> 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 $\Omega$ ) :**

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

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 $\Omega$ ) :**

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

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 $\Omega$ ) for Stand-alone Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325\text{mV} / (R_s + 30\text{m}\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 30))) \\ &= ((1 / 1.41) * (V_{ref} / 2.50) * (2.32143)) \\ &= (1 / 1.41) * V_{ref} * 0.928572 \end{aligned}$$

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

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref} * 0.928572$$

$$I_{MAX} = V_{ref} * 0.9286$$

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

- See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**4. TMC2208 with  $R_s = 0.110\Omega$  (110mΩ) for UART Mode:**

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

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

Therefore,

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

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

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

**5. TMC2209 with  $R_s = 0.110\Omega$  (110mΩ) for Stand-alone Mode:**

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

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

Therefore,

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

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

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

- See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**6. TMC2209 with  $R_s = 0.110\Omega$  (110mΩ) for UART Mode:**

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

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

Therefore,

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

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

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

**7. TMC2225 with  $R_s = 0.150\Omega$  (150mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / (150 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (1.9118)) \\ &= (1 / 1.41) * V_{ref} * 0.7647 \end{aligned}$$

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

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref} * 0.7647$$

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

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

**8. TMC5160 with  $R_s = 0.075\Omega$  (75mΩ) for SPI Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (325mV / R_s)) \\ &= ((1 / 1.41) * (325 / 75)) \\ &= (1 / 1.41) * 4.33 \end{aligned}$$

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

Therefore,

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

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

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](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 <a href="#">View / Download</a>	
Previous release Supports AVR Arduino and PlatformIO	1.1.9	1.1.x.zip <a href="#">View / Download</a>	
Older release Supports Arduino 1.6.8 and up	1.0.7-2	1.0.x.zip	

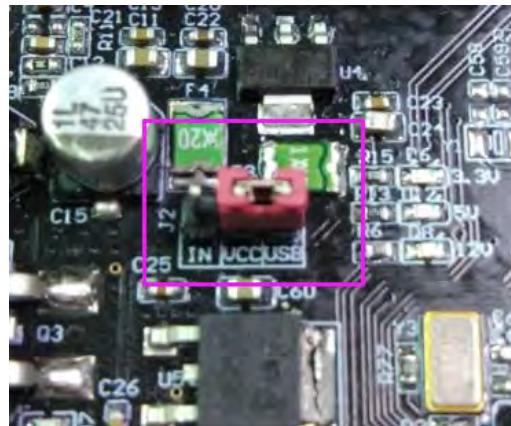
Marlin "Nightly" Builds			
Description	Version	Download	Configurations
Patches to latest 2.0 Marlin 2.0 with bug fixes Supports AVR and ARM Arduino and PlatformIO	bugfix-2.0.x	bugfix-2.0.x.zip <a href="#">View / Download</a>	
Proceed with Caution! Marlin 2.1 development Supports AVR and ARM Arduino and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

- Go to the next page.

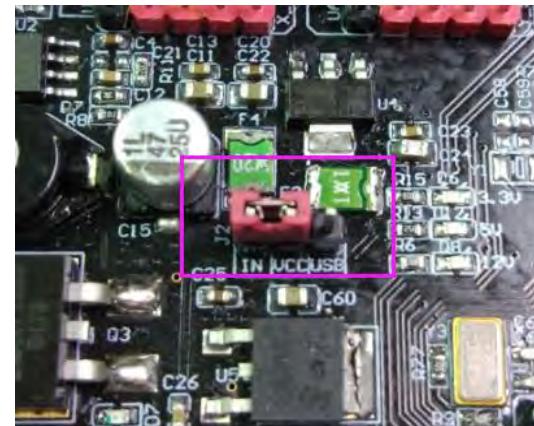
## The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

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:**

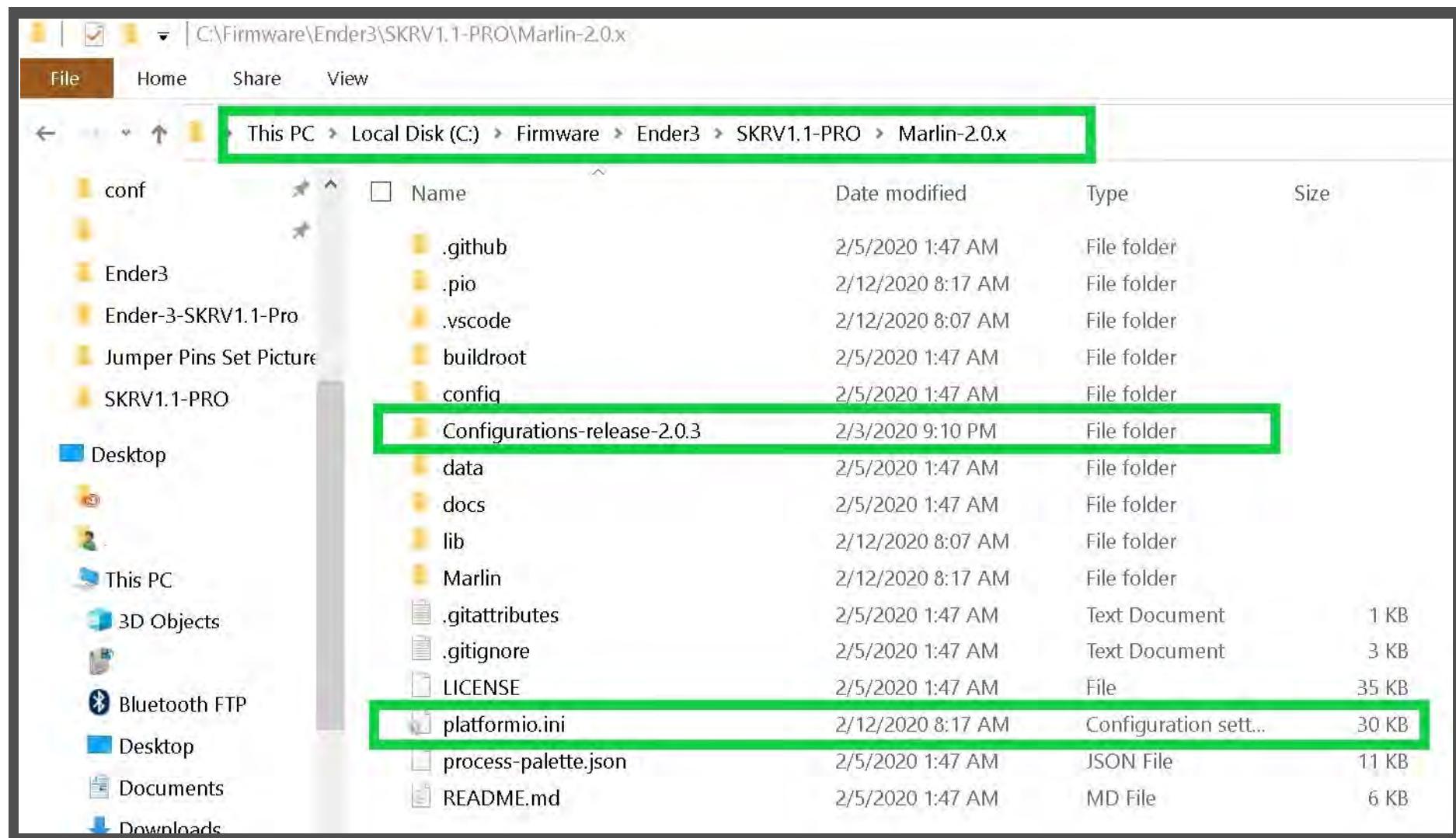
After downloading the files, use Visual Studio Code (VScode) to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the micro SD card into the board. Reboot the board; wait for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

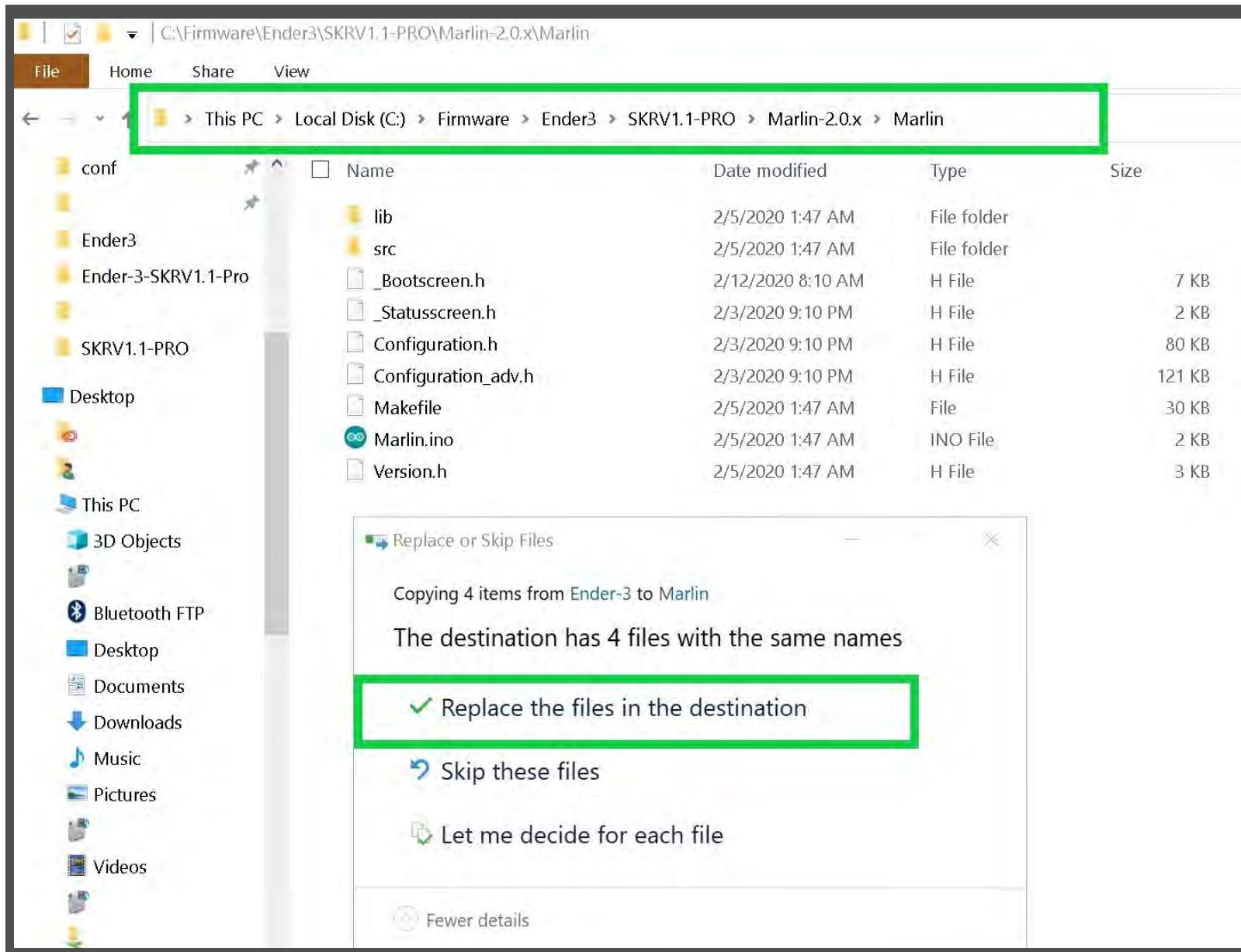
- Copy the below default configuration files (for me Ender 3 files, for you it could be another printer) to the directory where Marlin's Configuration.h and Configuration\_adv.h reside. See picture below for which files you will copy. See the next page to see where to place the files.



- Go to the next page.

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

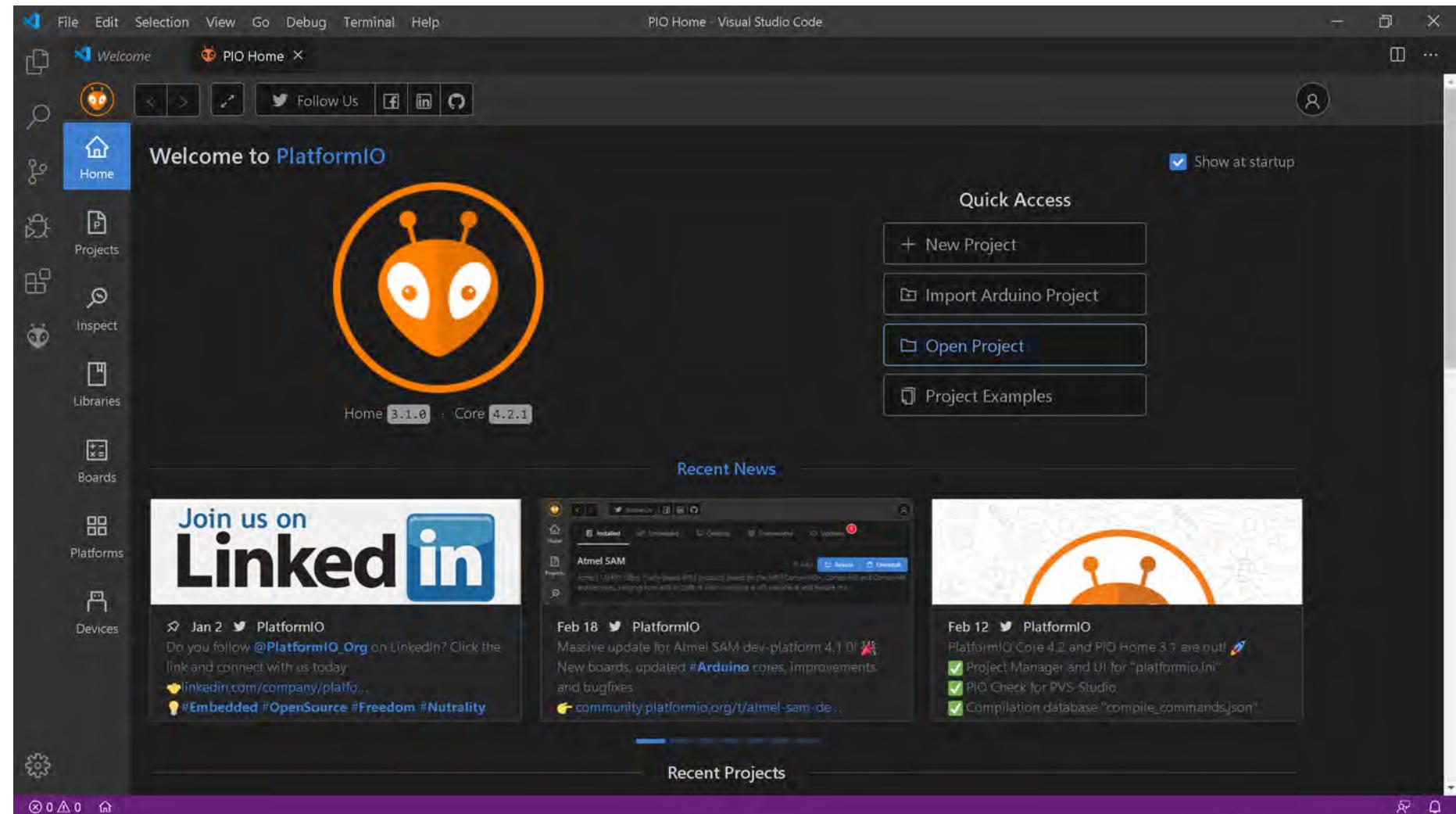
- Copy TO THIS directory so the above files reside in the same directory as Configuration.h and Configuration\_adv.h. When prompted allow the files to be overwritten!



- Go to the next page.

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

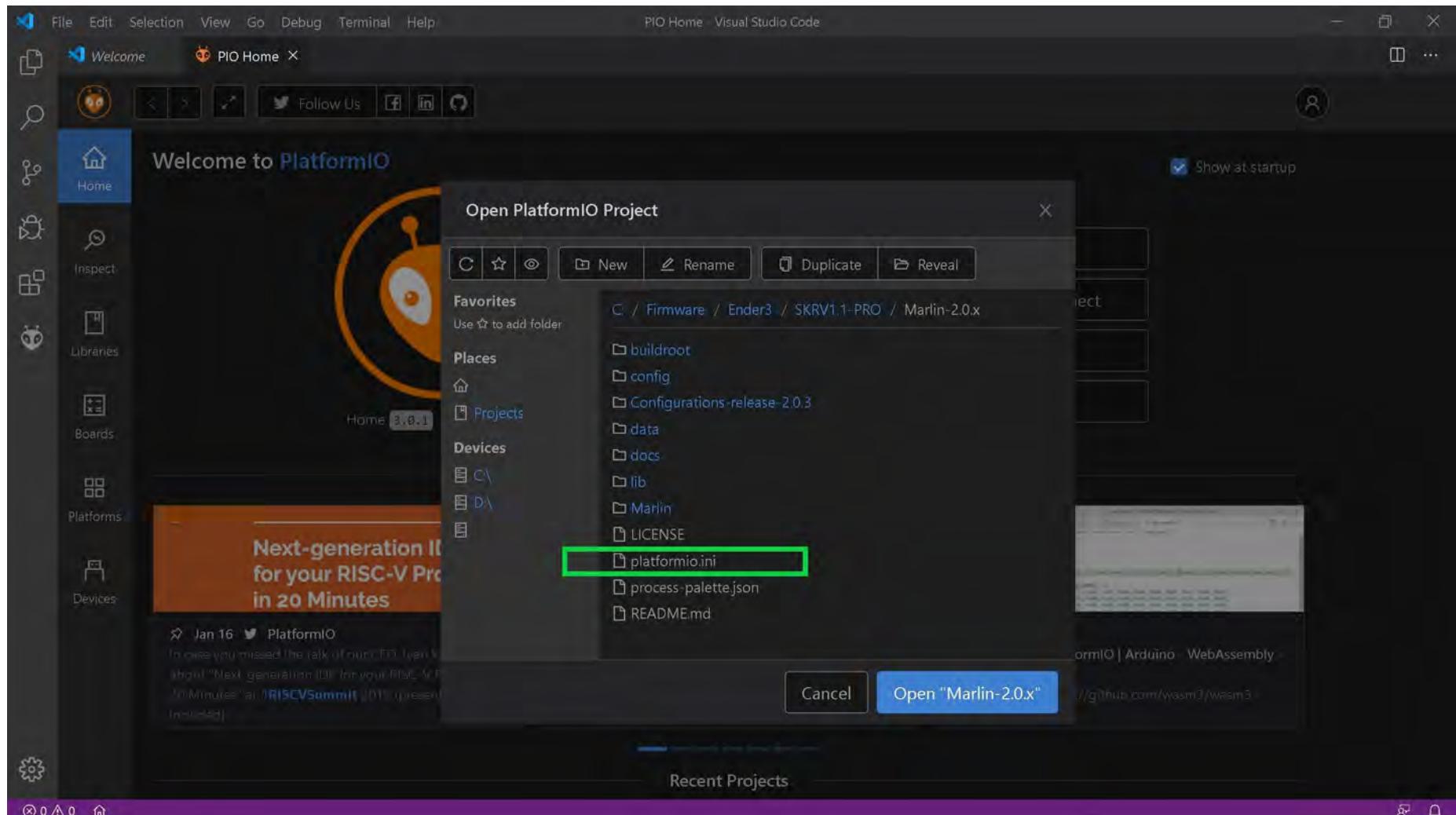
- Open VS code (see picture below) and then select "Open Project".



- Go to the next page.

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

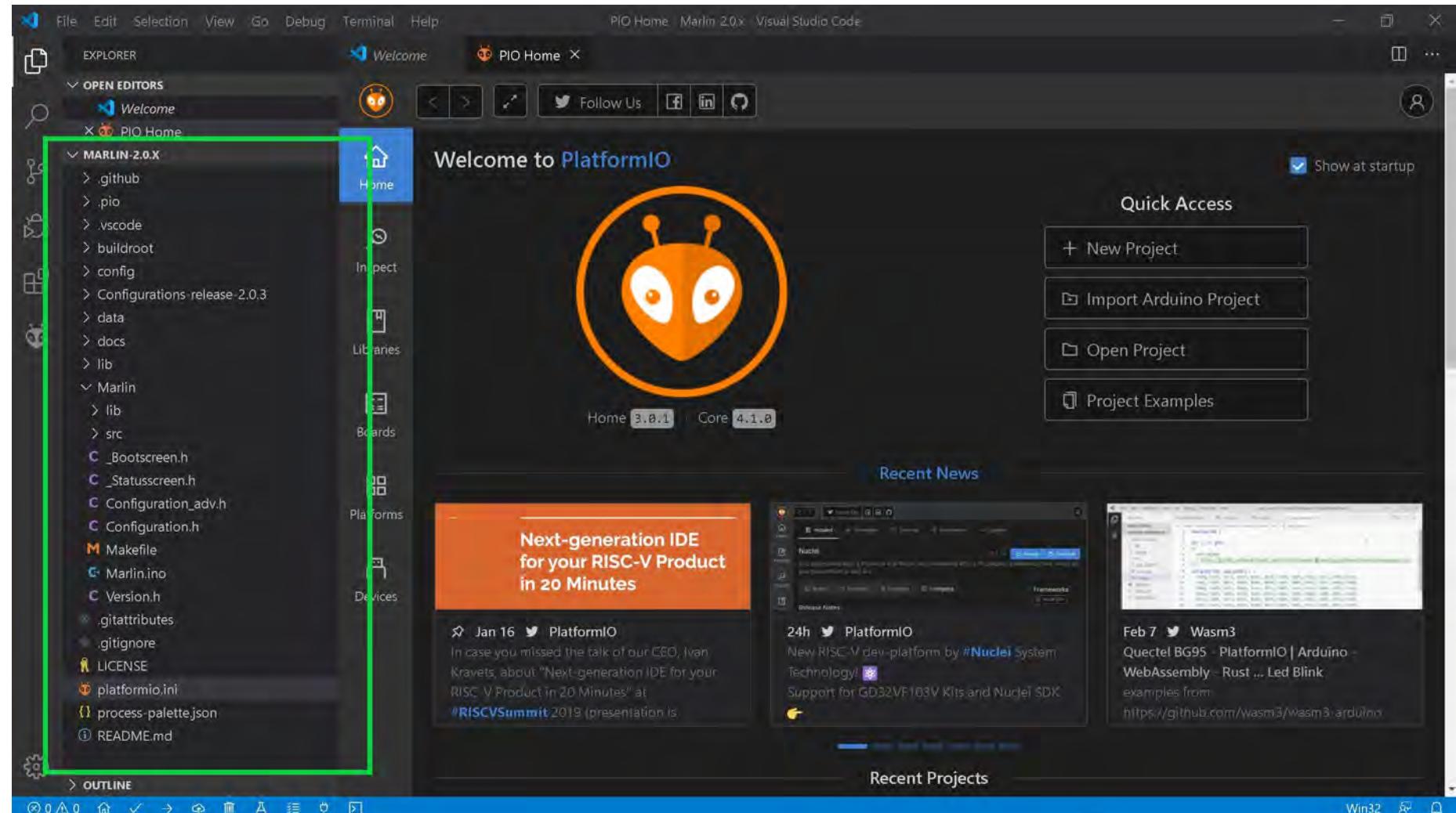
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- Go to the next page.

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

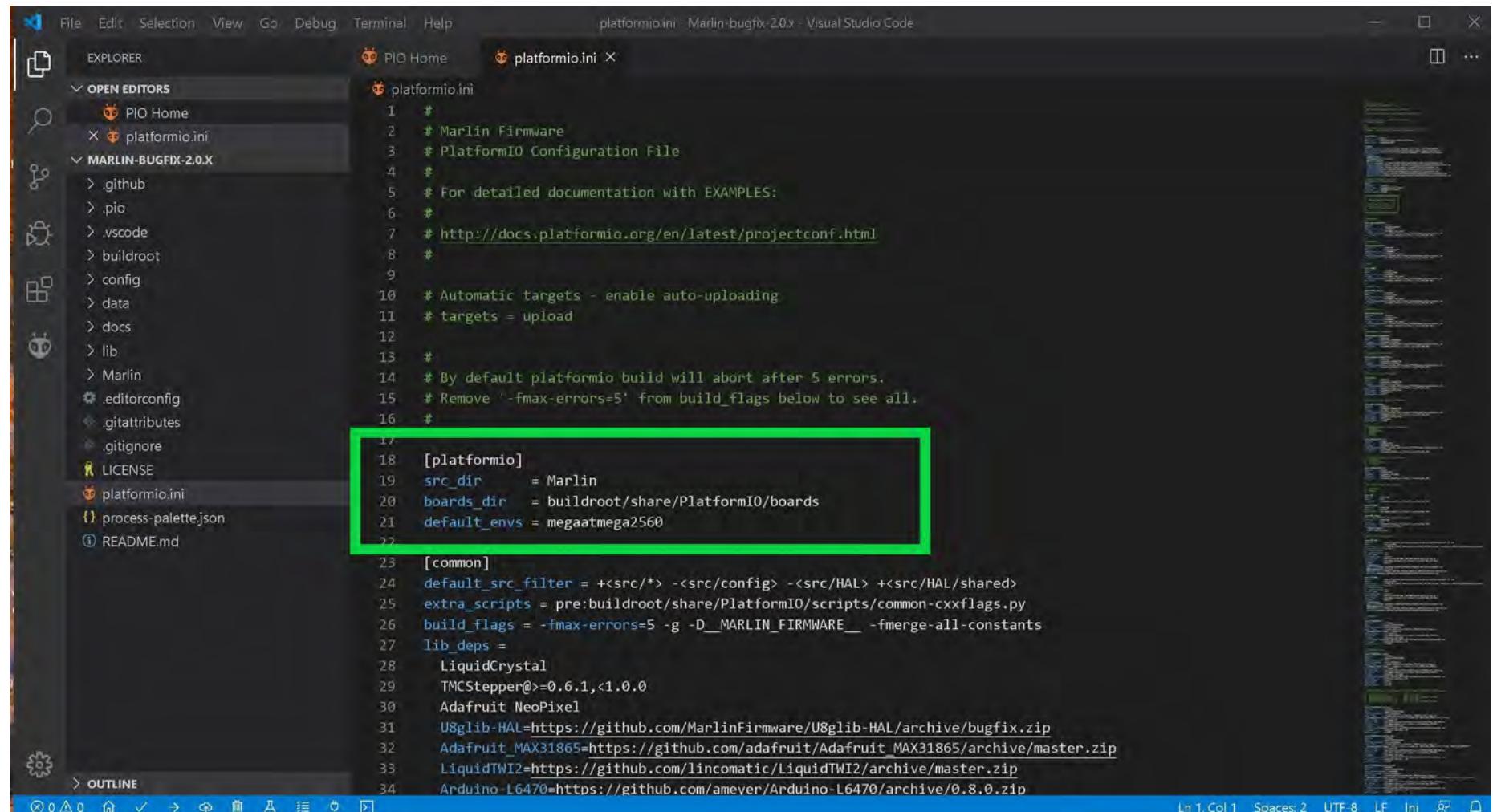
- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



- 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



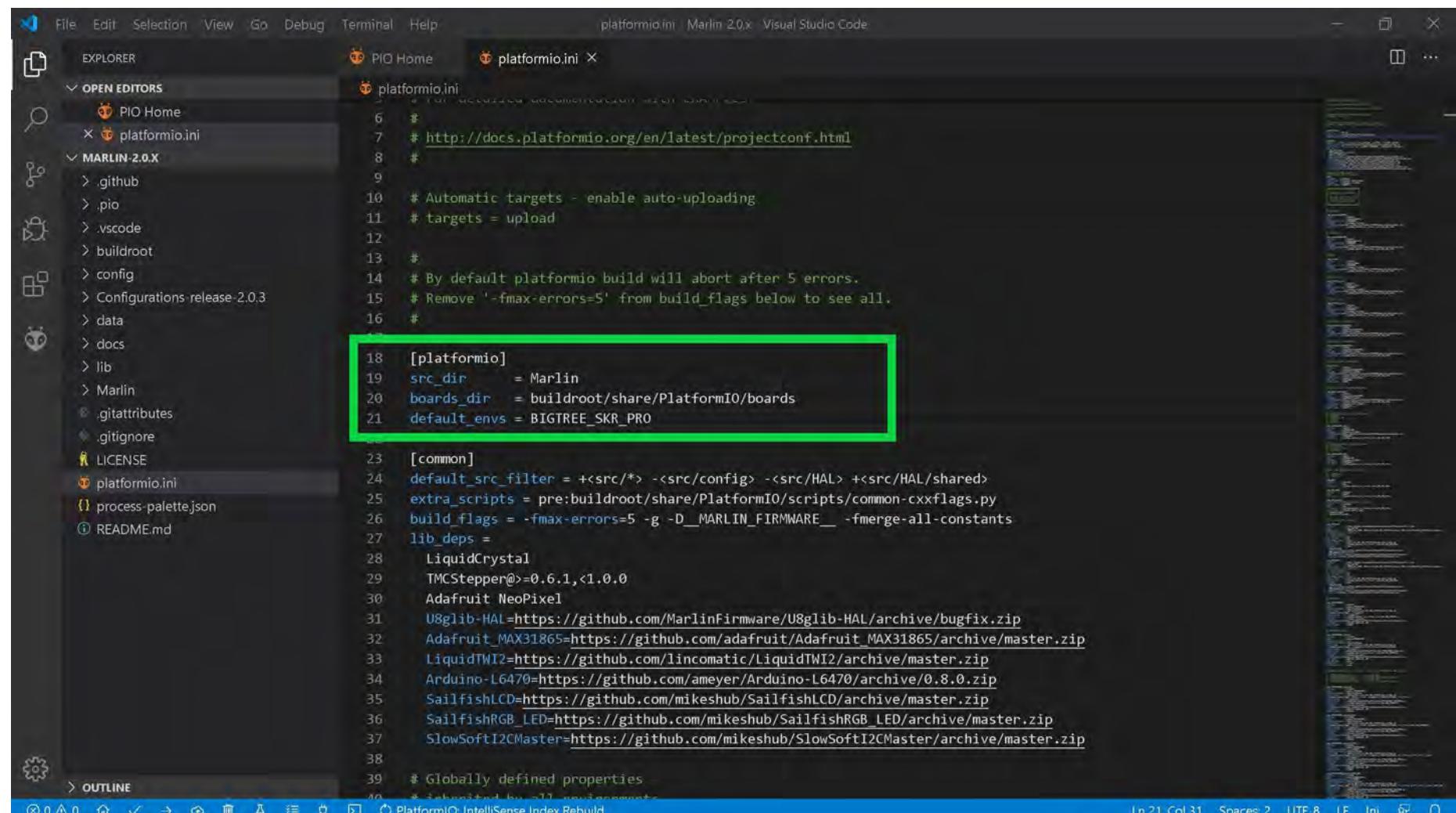
```
platformio.ini - Marlin-bugfix-2.0.x - Visual Studio Code

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

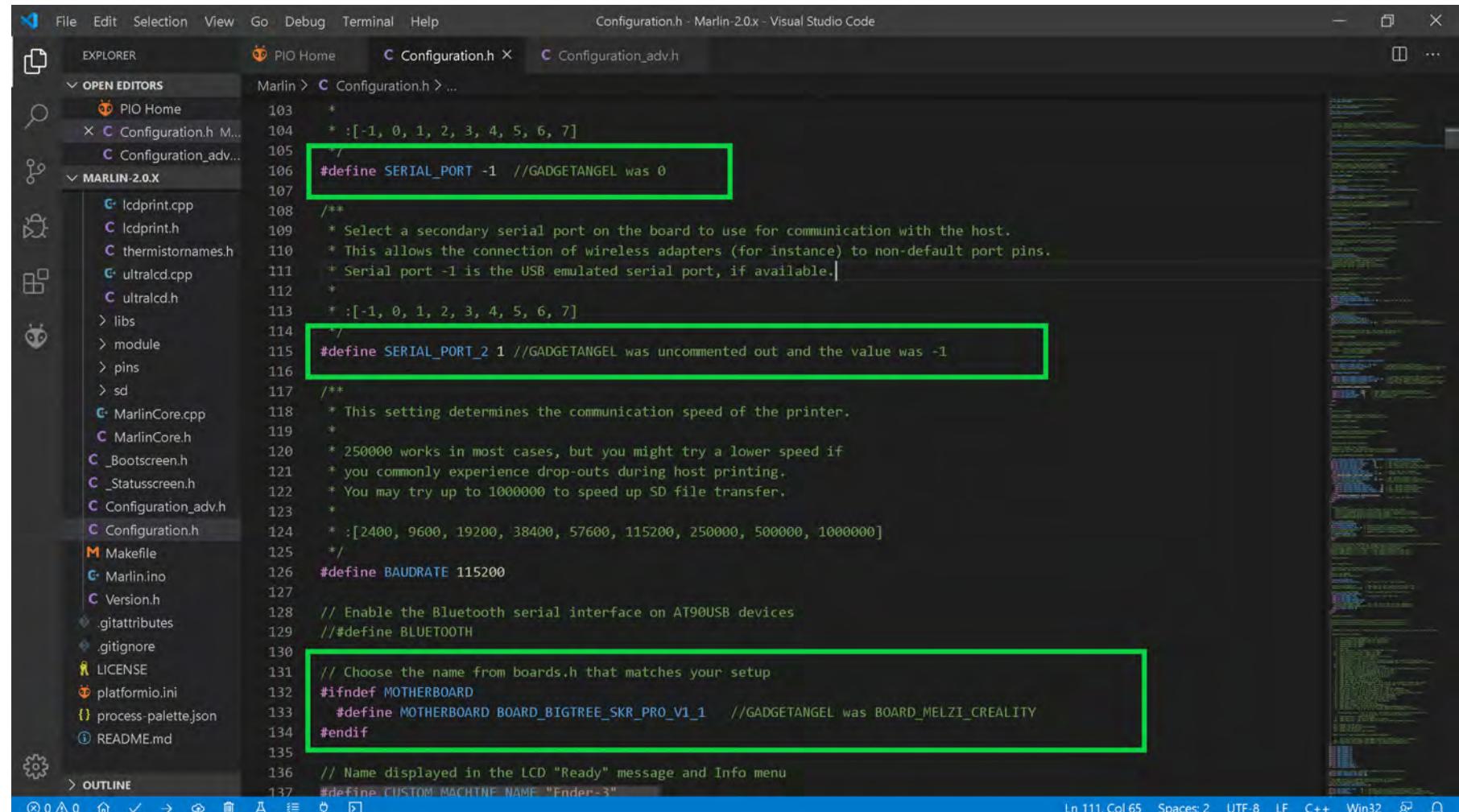
[common]
default_src_filter = +<src/*> -<src/config> -<src/HAL> +<src/HAL/shared>
extra_scripts = pre:buildroot/share/PlatformIO/scripts/common-cxxflags.py
build_flags = -fmax-errors=5 -g -D_MARLIN_FIRMWARE_ -fmerge-all-constants
lib_deps =
    LiquidCrystal
    TMCStepper@>=0.6.1,<1.0.0
    Adafruit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



```

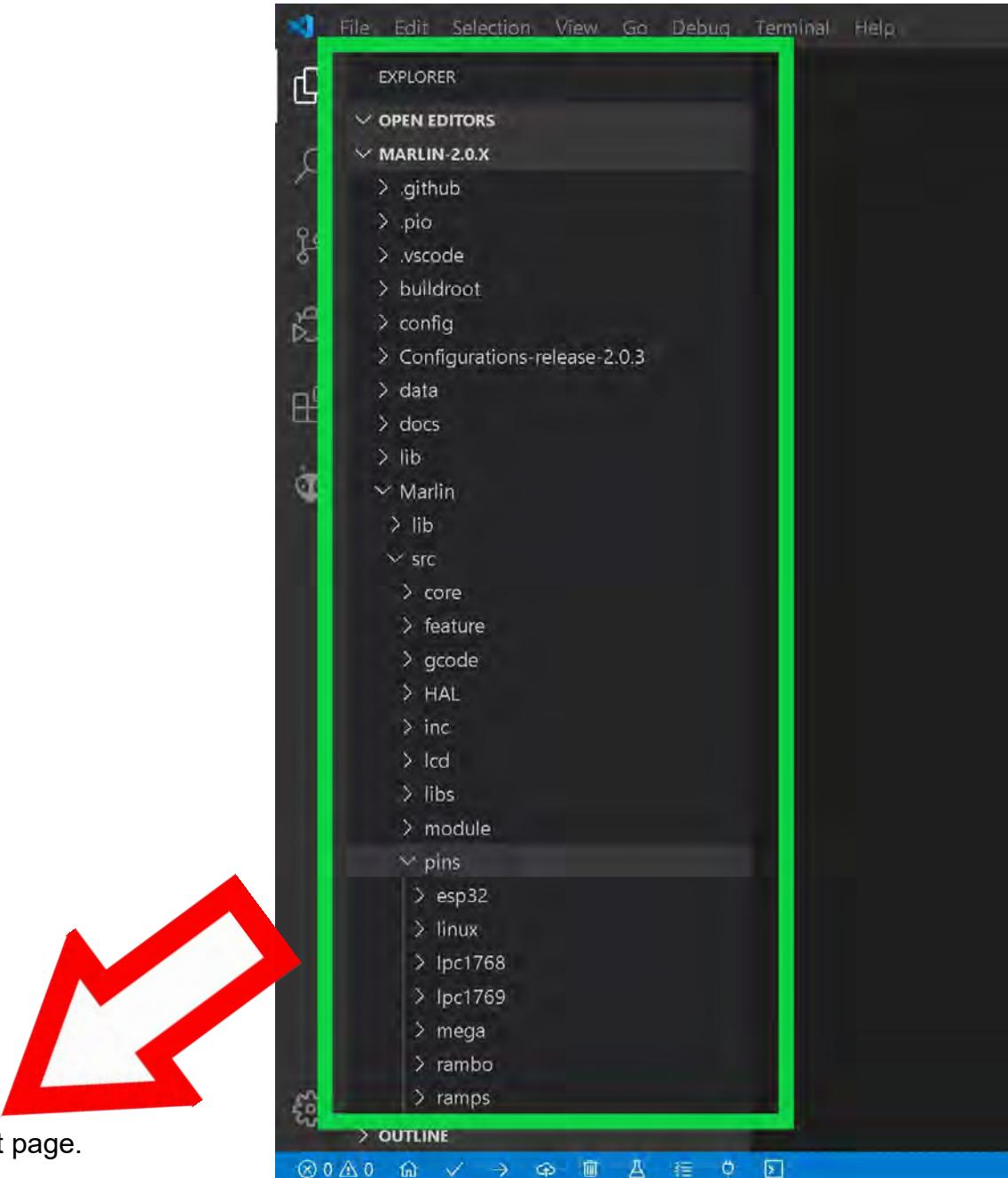
103  /*
104  * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
105 */
106 #define SERIAL_PORT -1 //GADGETANGEL was 0
107
108 /**
109 * Select a secondary serial port on the board to use for communication with the host.
110 * This allows the connection of wireless adapters (for instance) to non-default port pins.
111 * 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.

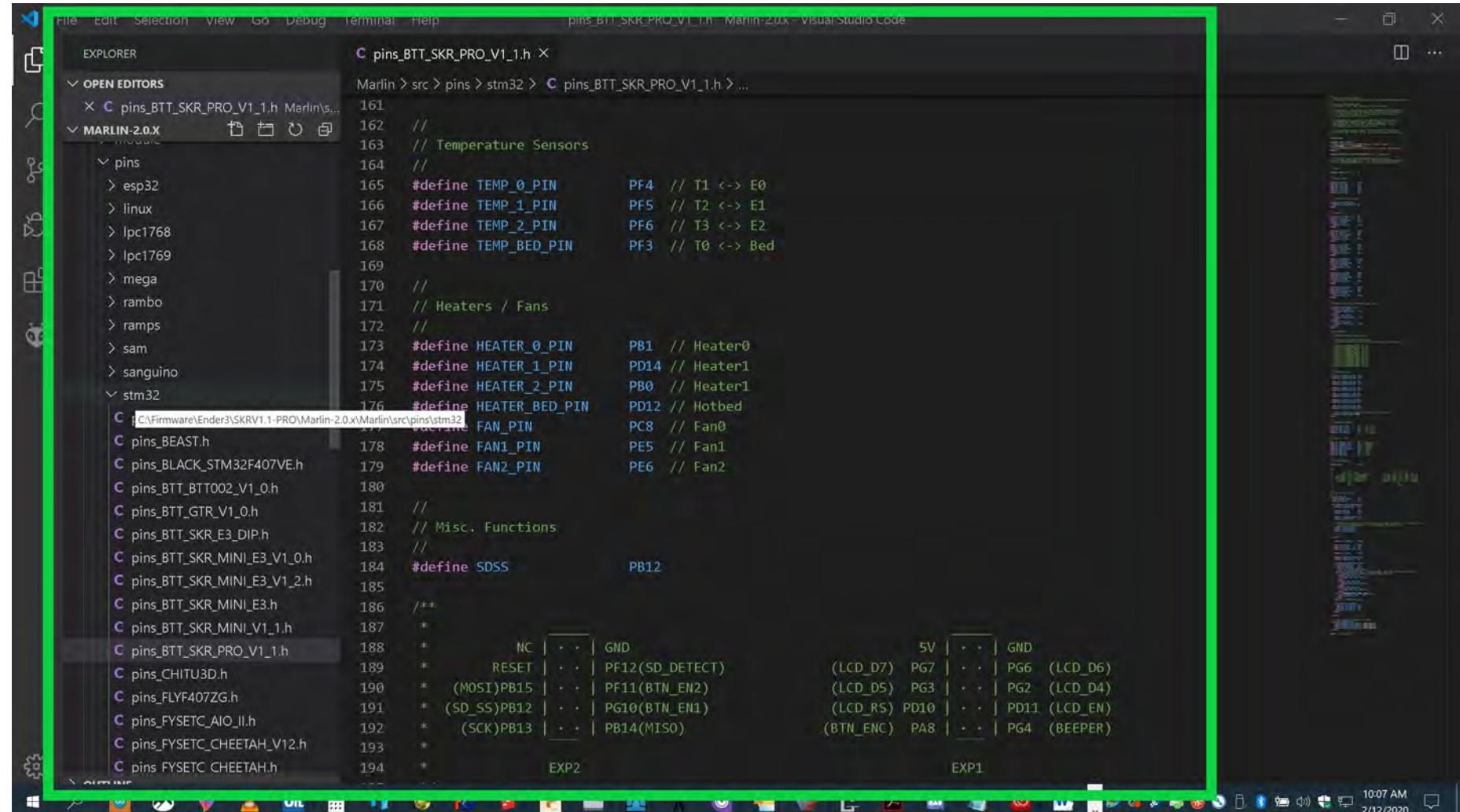
## The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

- 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



File Edit Selection View Go Debug Terminal Help pins\_BTT\_SKR\_PRO\_V1\_1.h Marlin-2.0.x - Visual Studio Code

EXPLORER      C pins\_BTT\_SKR\_PRO\_V1\_1.h X

OPEN EDITORS      Marlin > src > pins > stm32 > C pins\_BTT\_SKR\_PRO\_V1\_1.h > ...

MARLIN-2.0.X      pins\_BTT\_SKR\_PRO\_V1\_1.h Marlin\src\pins\stm32\pins\_BTT\_SKR\_PRO\_V1\_1.h

```

161 // 
162 // 
163 // Temperature Sensors
164 // 
165 #define TEMP_0_PIN PF4 // T1 <-> E0
166 #define TEMP_1_PIN PF5 // T2 <-> E1
167 #define TEMP_2_PIN PF6 // T3 <-> E2
168 #define TEMP_BED_PIN PF3 // T0 <-> Bed
169 // 
170 // 
171 // Heaters / Fans
172 // 
173 #define HEATER_0_PIN PB1 // Heater0
174 #define HEATER_1_PIN PD14 // Heater1
175 #define HEATER_2_PIN PB0 // Heater1
176 #define HEATER_BED_PIN PD12 // Hotbed
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

```

10:07 AM 2/12/2020



File Edit Selection View Go Debug Terminal Help pins\_BTT\_SKR\_PRO\_V1\_1.h Marlin-2.0.x - Visual Studio Code

EXPLORER      C pins\_BTT\_SKR\_PRO\_V1\_1.h X

OPEN EDITORS      Marlin > src > pins > stm32 > C pins\_BTT\_SKR\_PRO\_V1\_1.h > ...

MARLIN-2.0.X      pins\_BTT\_SKR\_PRO\_V1\_1.h Marlin\src\pins\stm32\pins\_BTT\_SKR\_PRO\_V1\_1.h

```

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

```

10:07 AM 2/12/2020

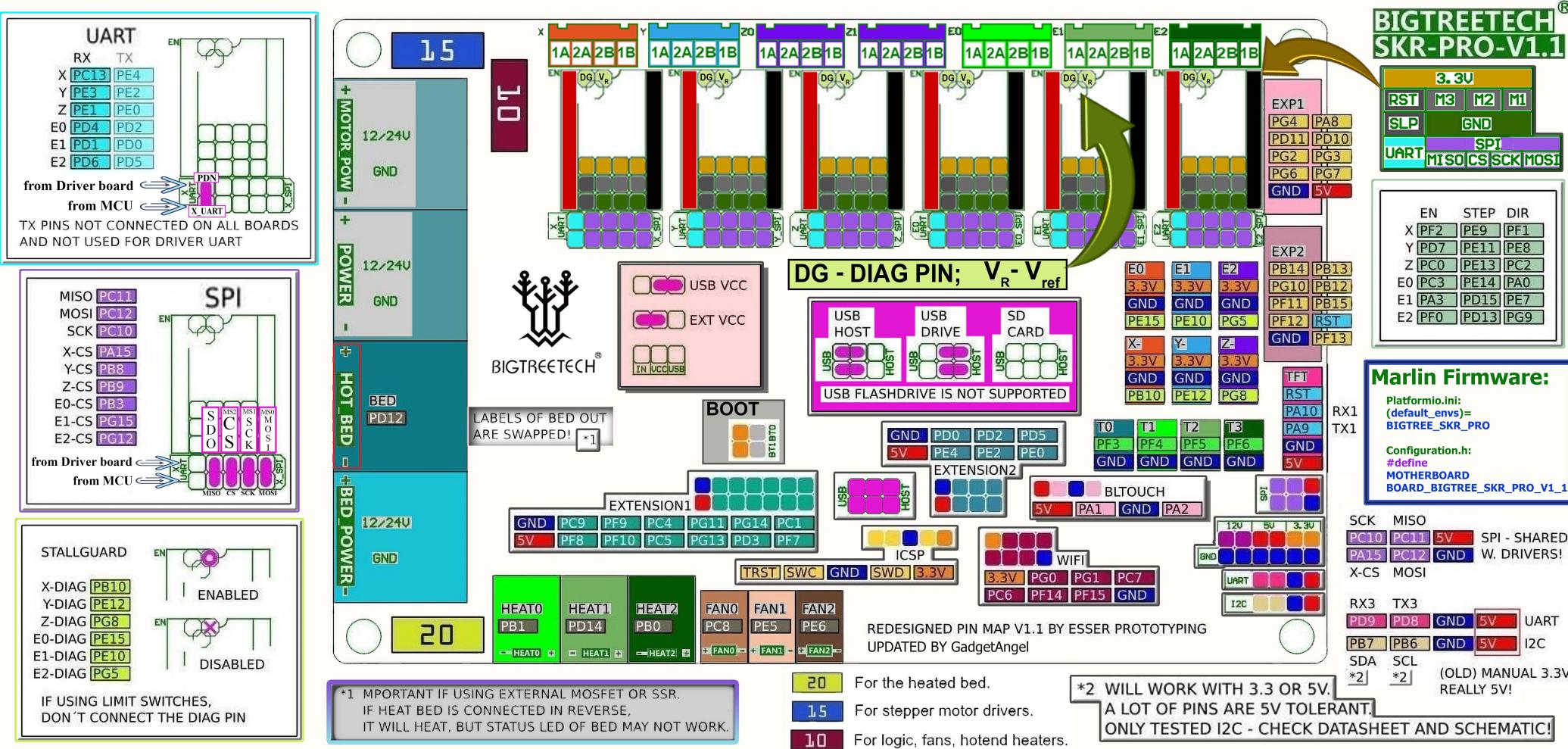
- Go to the next page.

## 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.

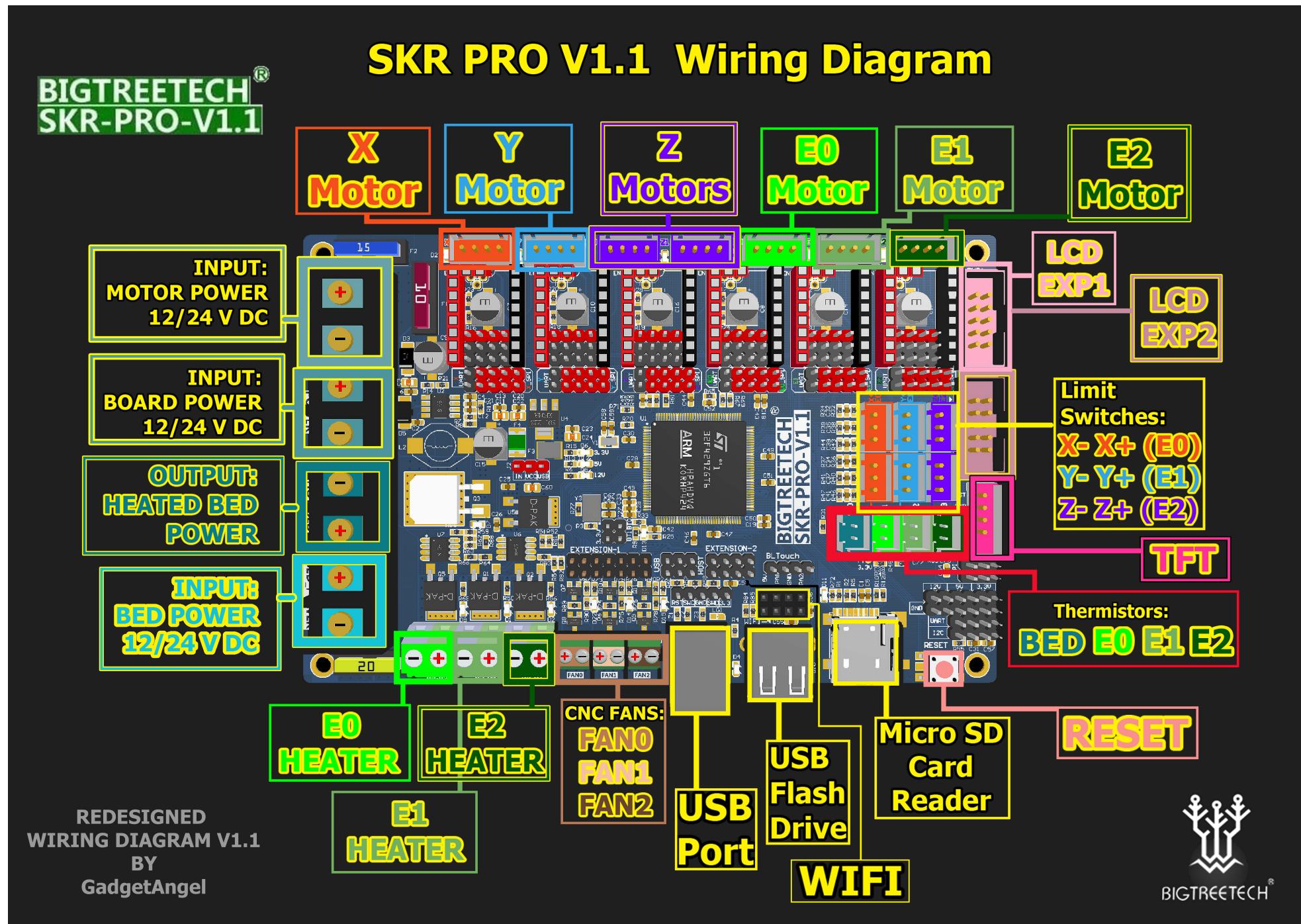
3



- 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

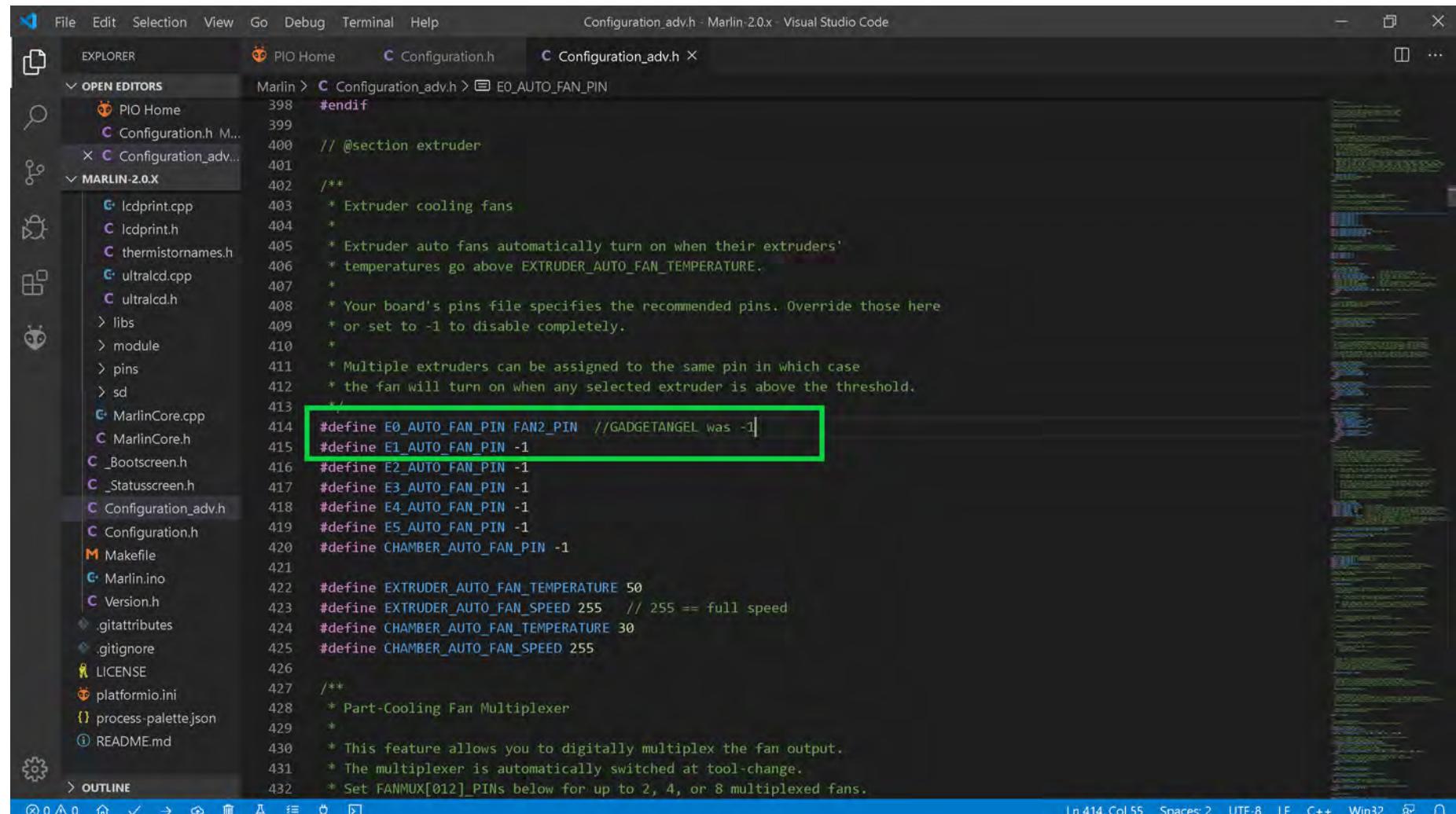


- Go to the next page.

## 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



The screenshot shows the Visual Studio Code interface with the 'Configuration\_adv.h' file open. The code editor displays the following snippet from the file:

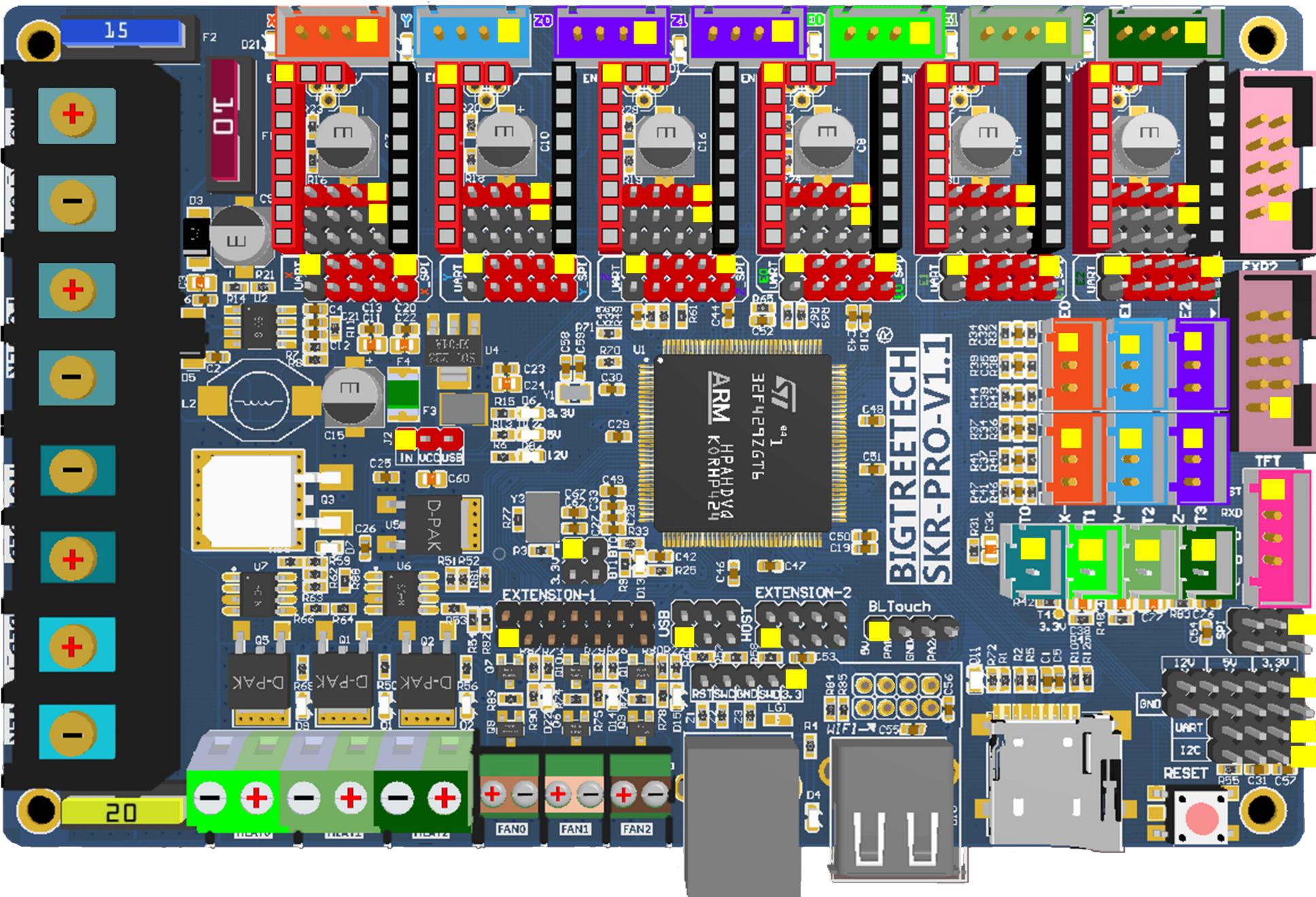
```

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|12]_PINs below for up to 2, 4, or 8 multiplexed fans.

```

A green rectangular box highlights the line `#define E0_AUTO_FAN_PIN FAN2_PIN //GADGETANGEL was -1`. The status bar at the bottom right of the code editor shows: Ln 414, Col 55, Spaces: 2, UTF-8, LF, C++, Win32.

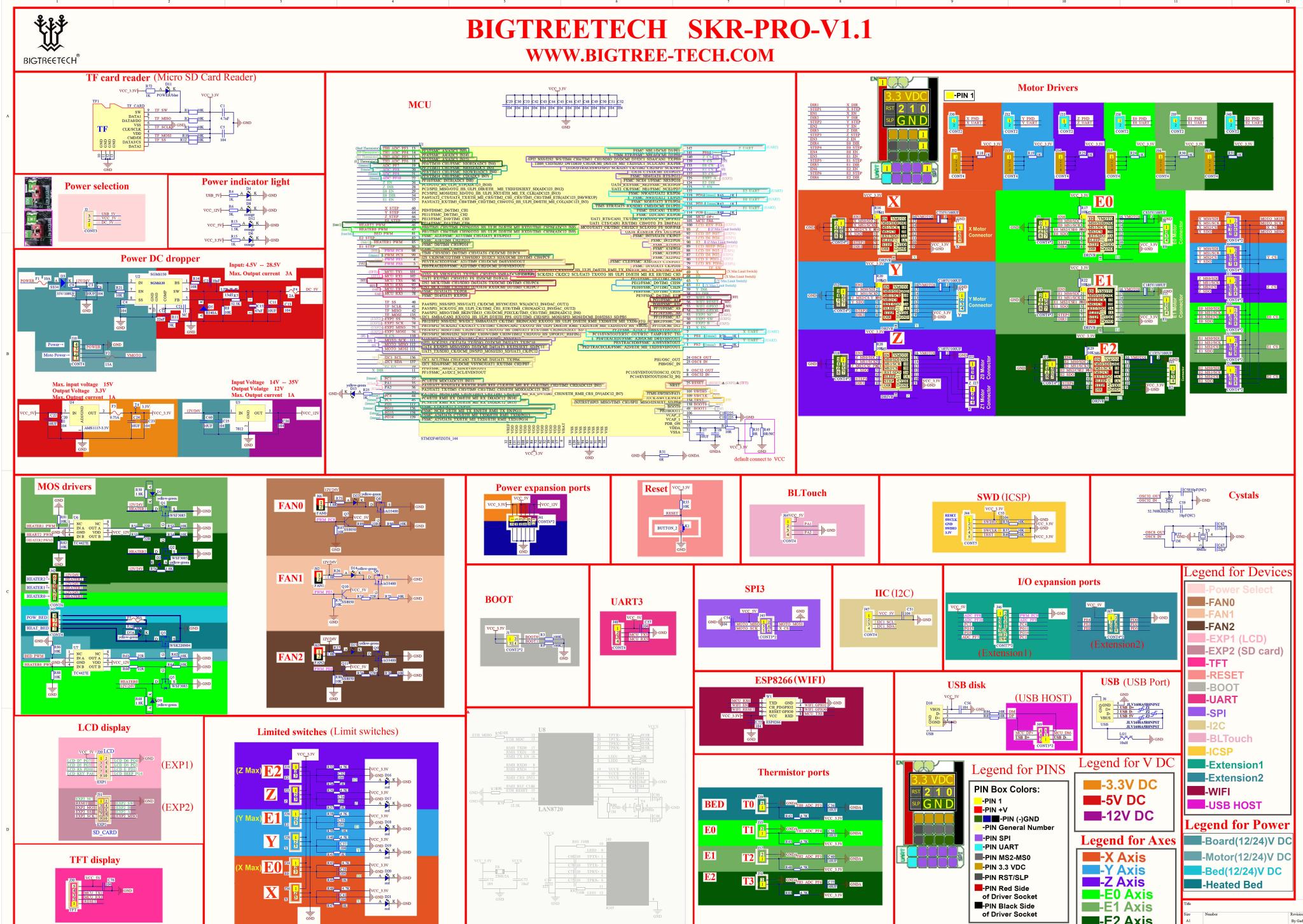
- To see more Marlin setup for the latest release, [please refer to the stepper motor driver section of this document for the stepper motor driver of your choice.](#)
- To see my updated PIN 1 diagram for SKR PRO V1.1, go to the next page.**

**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

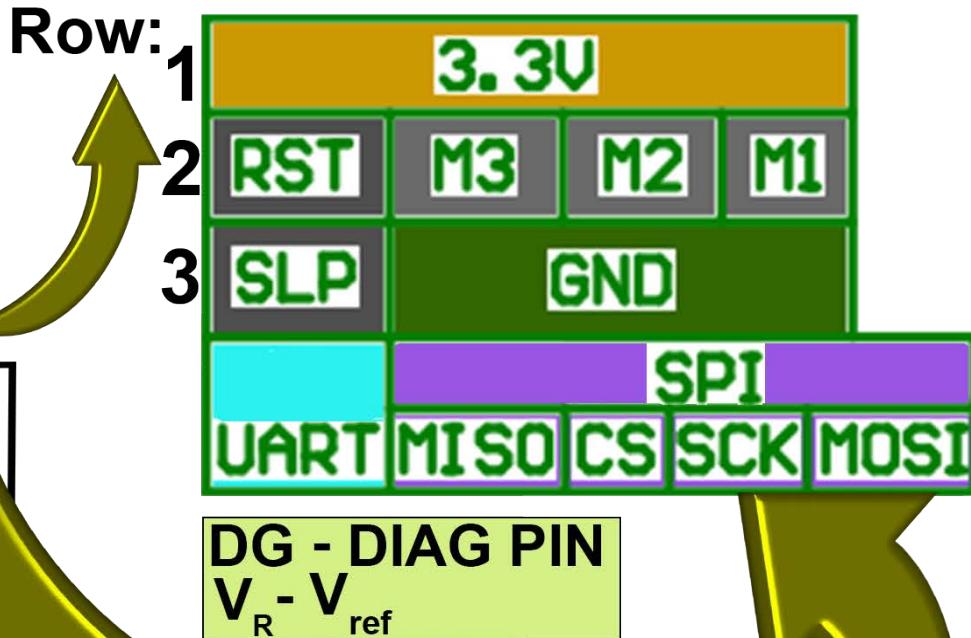
BIGTREETECH SKR-PRO-V1.1  
WWW.BIGTREE-TECH.COM

# 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 (D) across row 2 and 3 in RST/SLP column; ONE jumper (9) across row 2 and 3 in MS2 column; and ONE jumper (7) across row 2 and 3 in MS0 column.

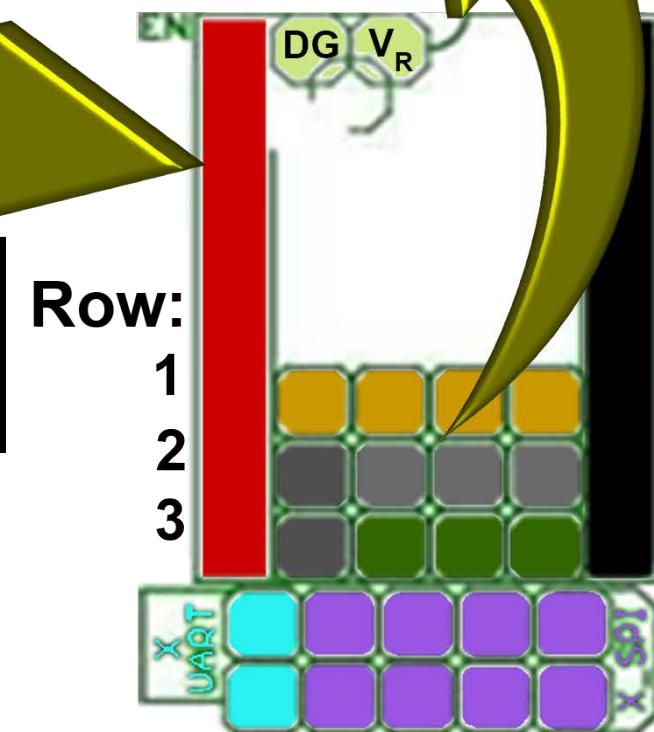
ENABLE	-	-	-	-	-	-	-	VMOT
MS1	-	-	-	-	-	-	-	GND
MS2	-	-	-	-	-	-	-	2B
MS3	-	-	MS2	MS1	MS0	-	-	2A
RESET	-	-	-	-	-	-	-	1A
SLEEP	-	-	-	8	-	-	-	1B
STEP	-	D	9	8	7	-	-	VDD
DIR	-	D	9	-	7	-	-	GND

Legend for  
MS0-MS2  
PINS:

2 - MS2  
1 - MS1  
0 - MS0

Legend for Socket:

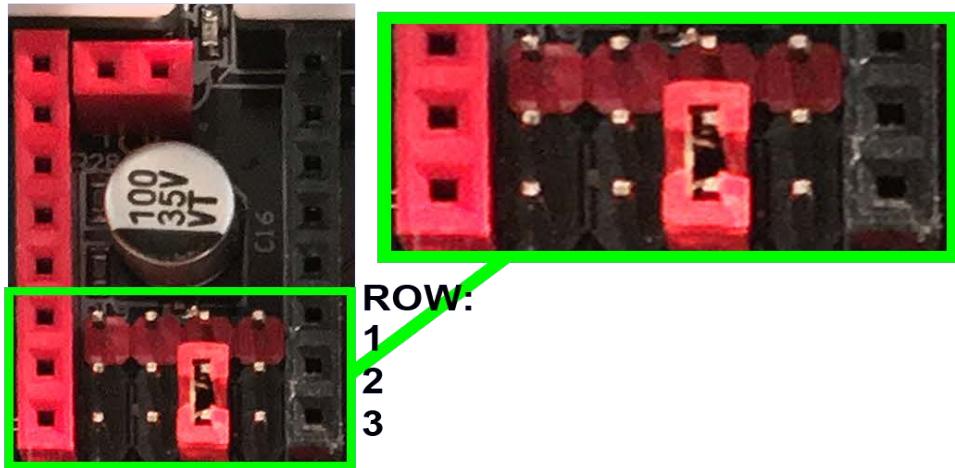
- - means: No jumper  
7 7 means: Jumper set  
7\* - means: DuPont jumper to GND



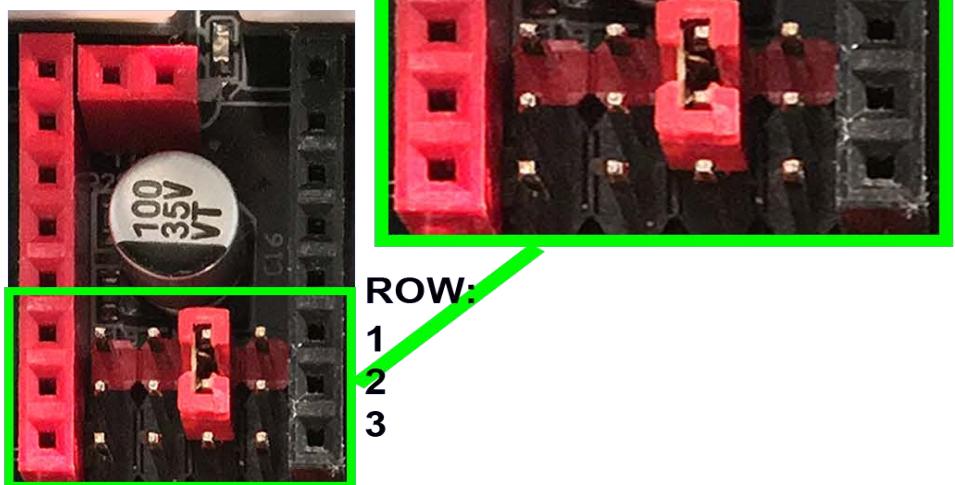
- Go to the next page.

## 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

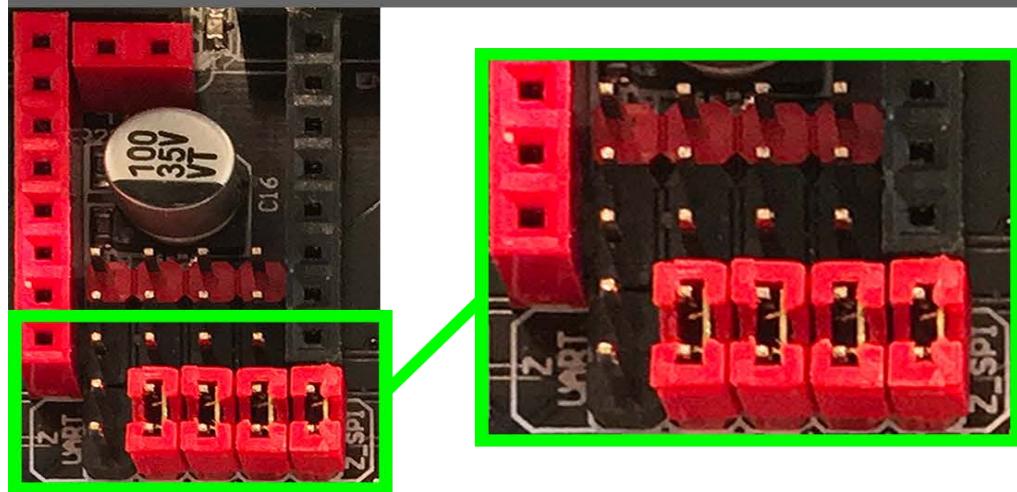
**Low** ➡ set Jumper between rows 2 and 3

**High** ➡ set Jumper between rows 1 and 2

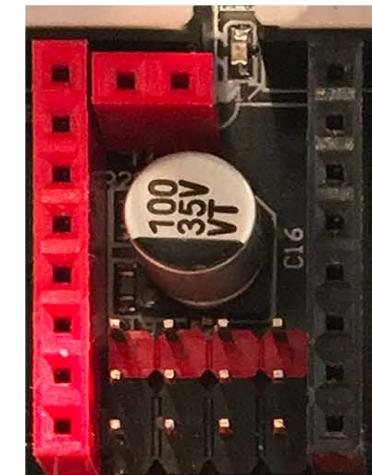
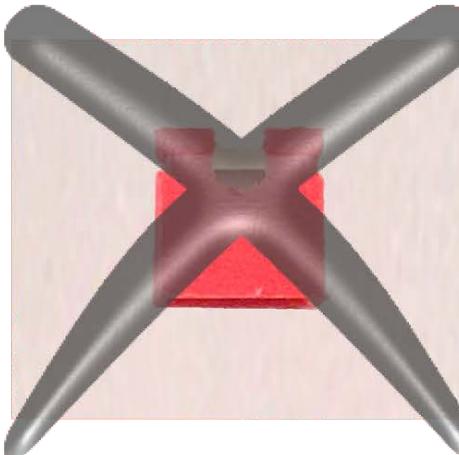
- Go to the next page.

## Legends for SKR PRO V1.1 Stepper Driver Socket Representations

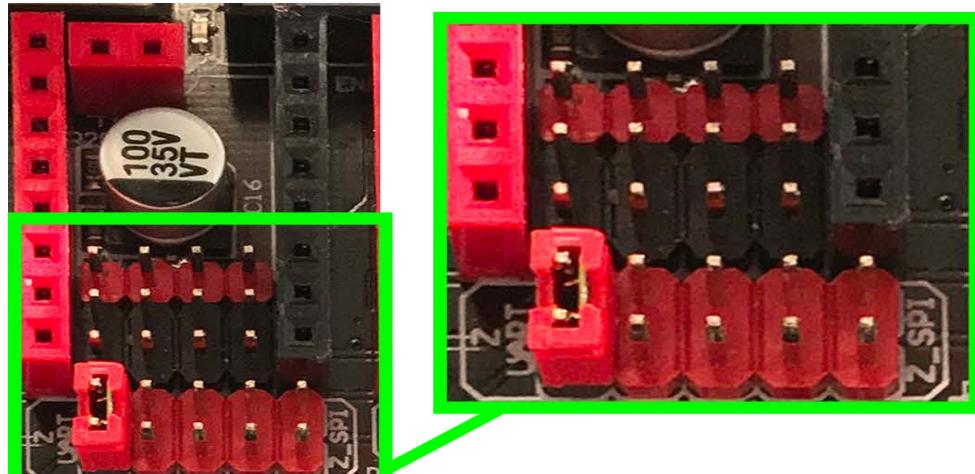
**SPIM** ➔ Jumper Set



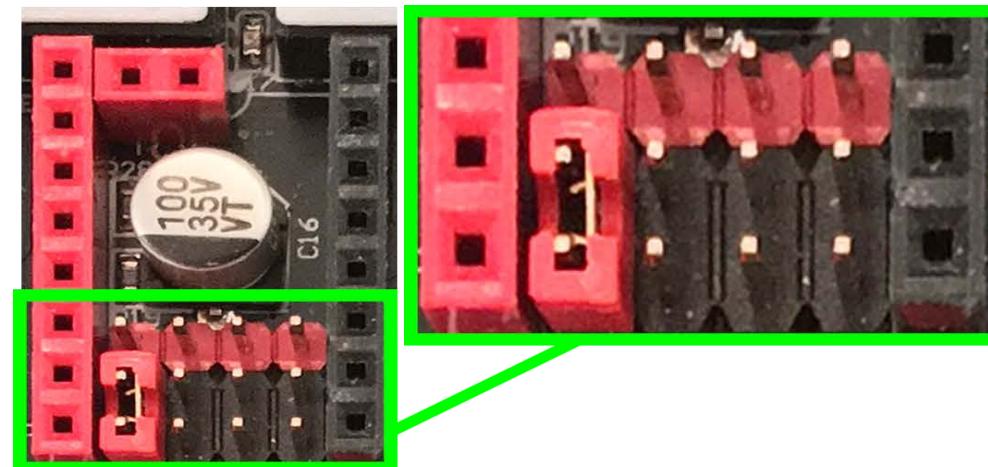
- ➔ No Jumper



**U** ➔ Jumper Set



**D** ➔ Jumper Set

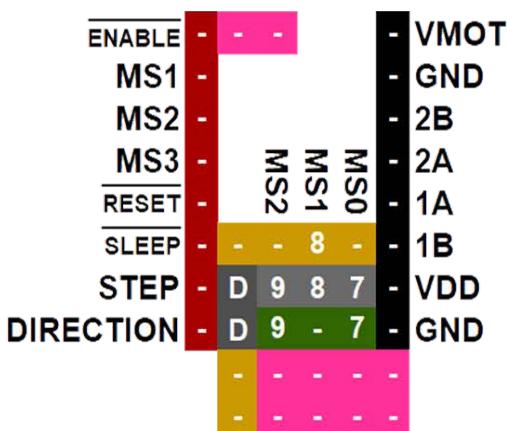


- Go to the next page.

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

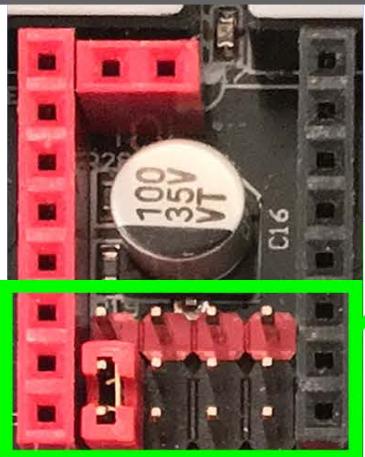
## Driver Socket Representation for D Jumper:

1 / 4



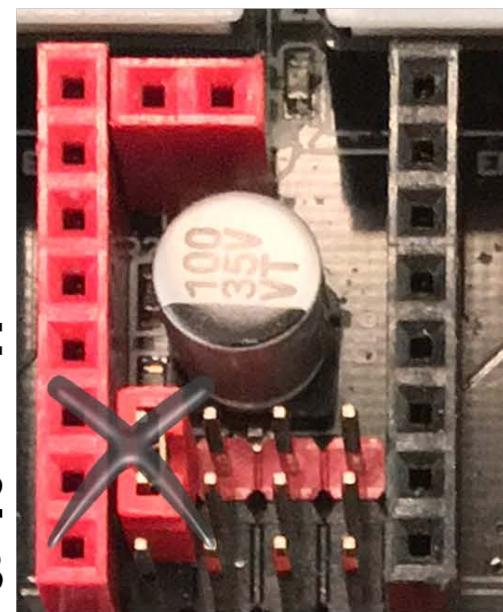
## The Right Way to Set the D Jumper:

**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:**

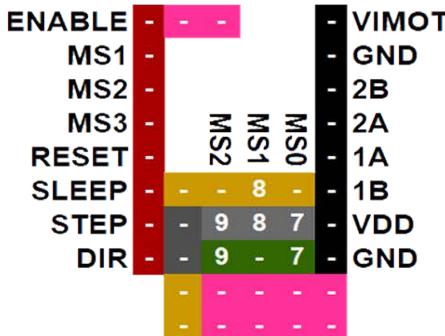
1  
2  
3

- Go to the next page.

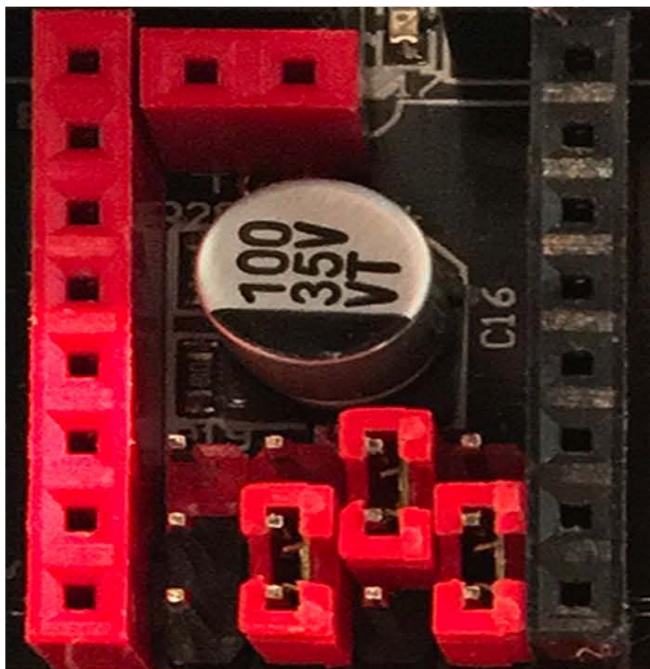
## Examples for Stepper Driver Socket Representations

### 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

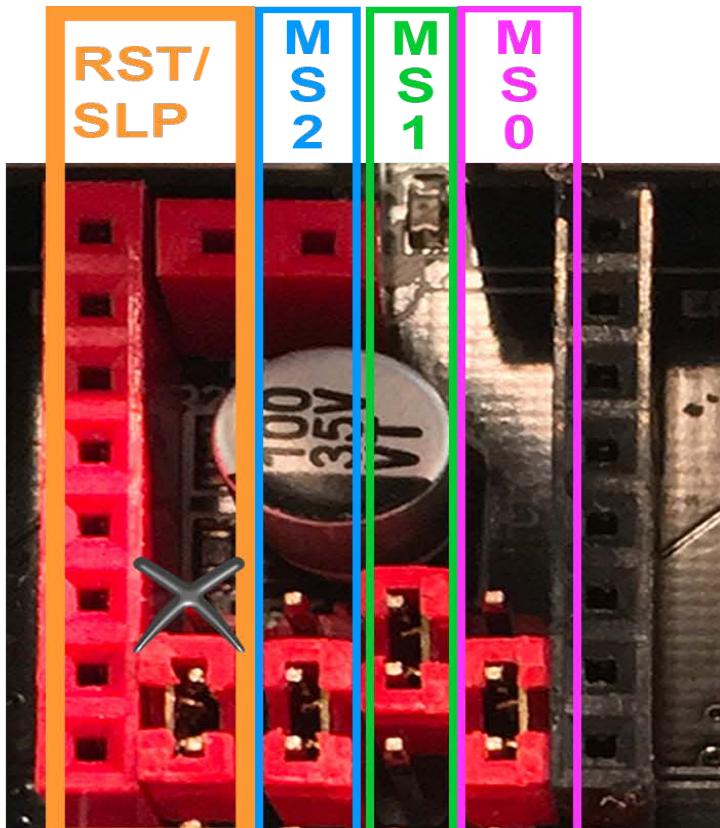
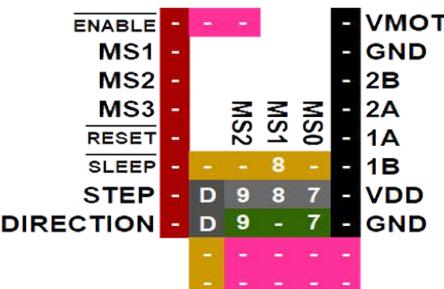
- Go to the next page.

## Examples for Stepper Driver Socket Representations

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

**Legend:**

1 / 4



- 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

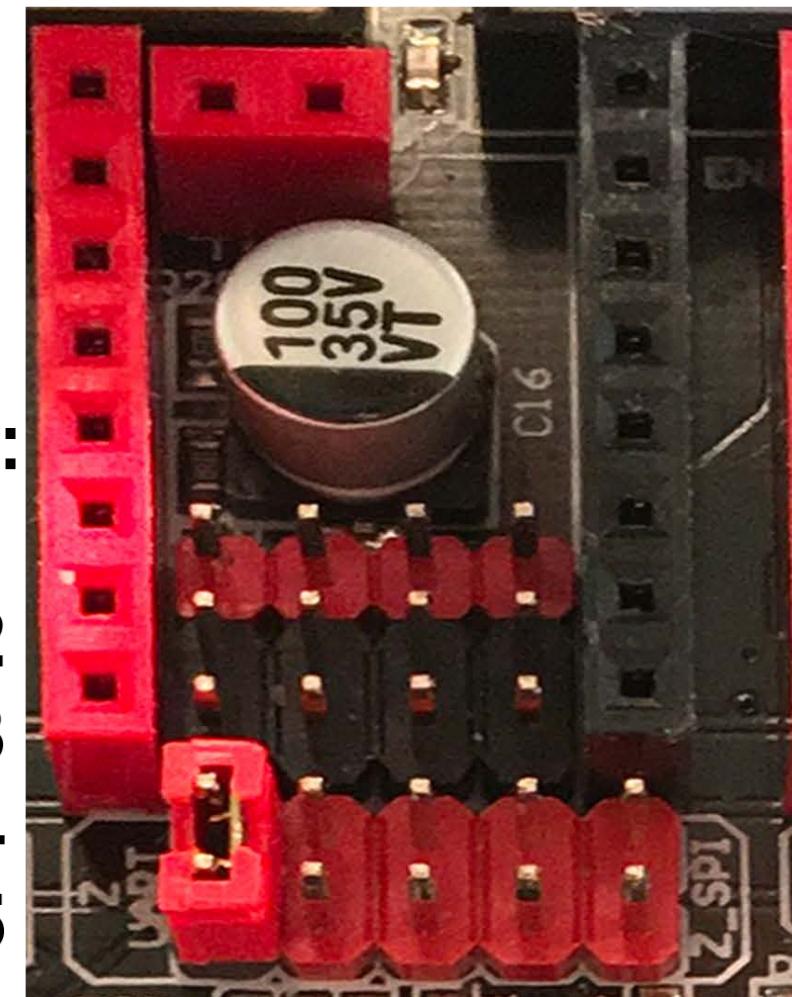
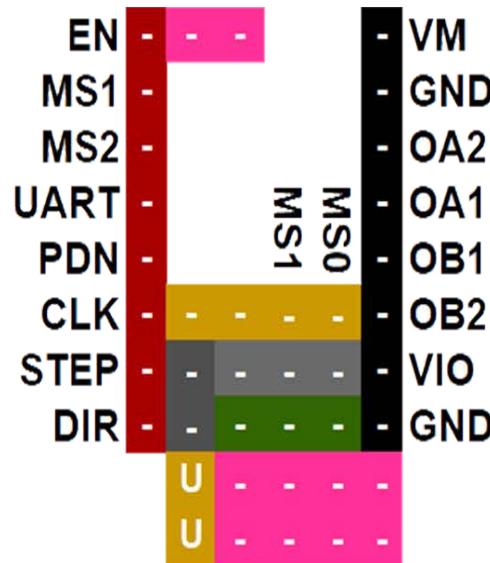
**MS0 PIN:**

**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



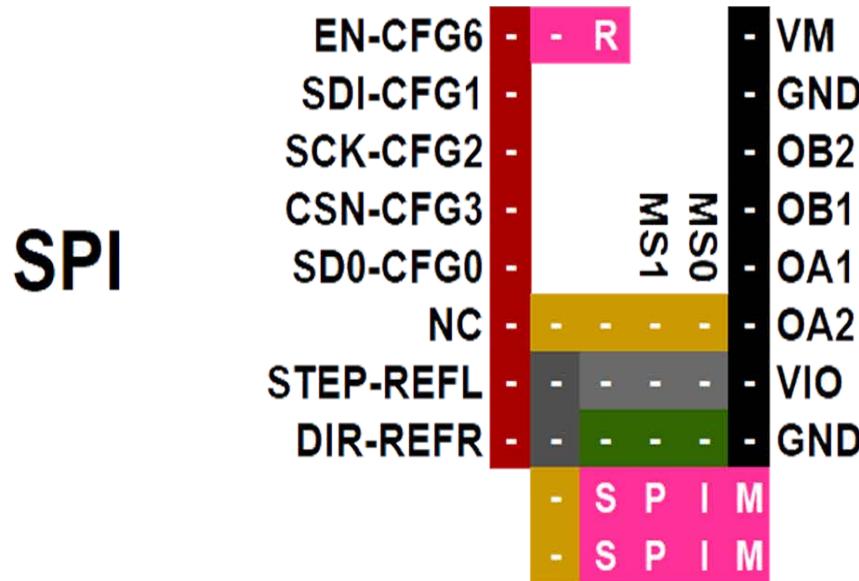
U Jumper Set

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:

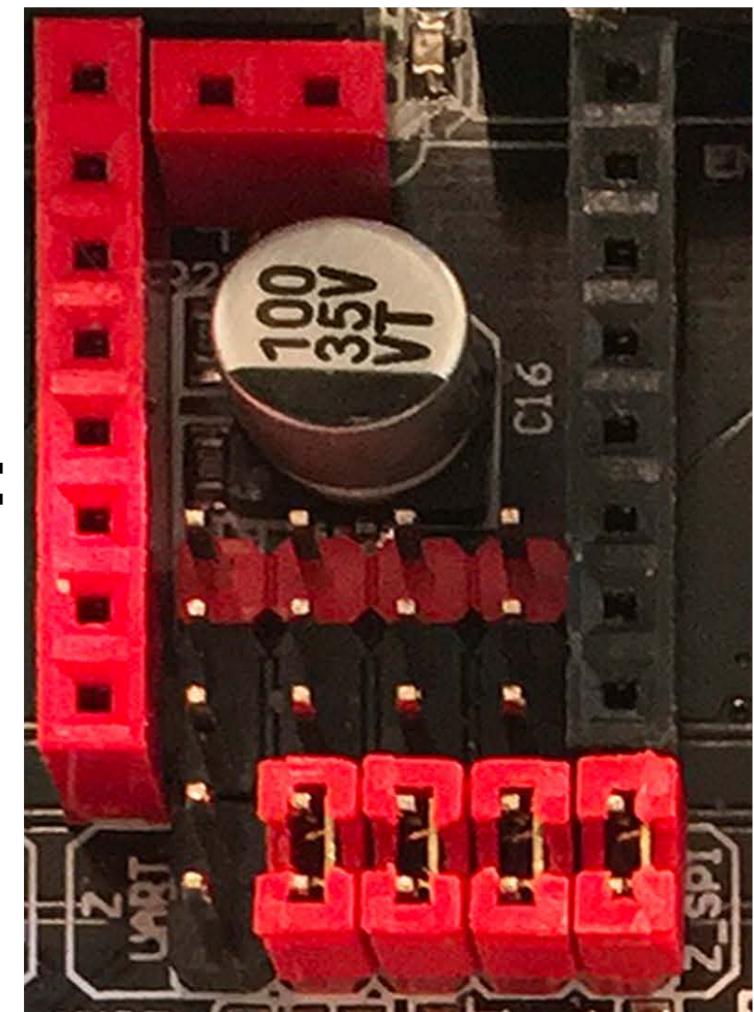


SPI

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

**SPIM → Jumper Set**

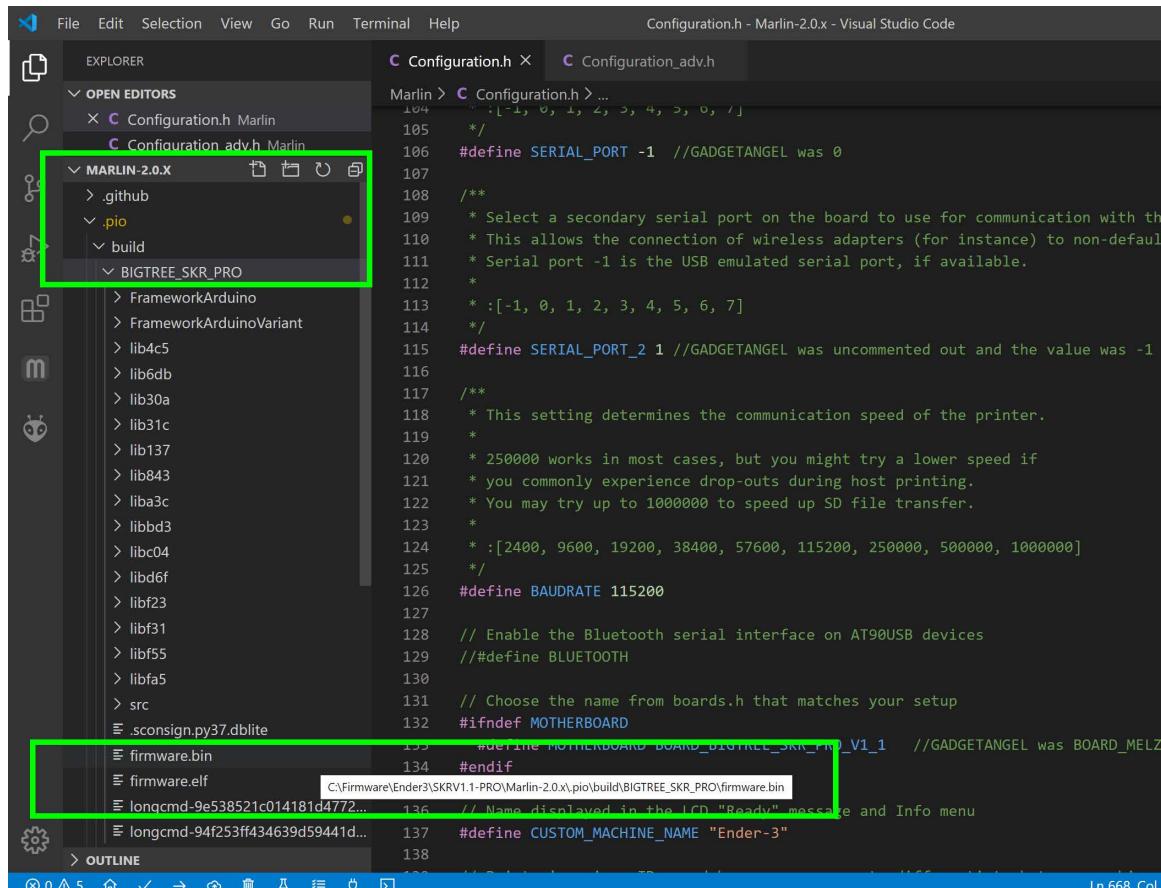
Row:

1  
2  
3  
4  
5

**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 with a tree view of the project structure. A green box highlights the folder `BIGTREE_SKR_PRO` under `Marlin-2.0.X\pio\build`. Inside this folder, the files `firmware.bin` and `firmware.elf` are also highlighted with a green box. The main editor area displays the `Configuration.h` file. A callout box points to the line of code where `firmware.bin` is defined:

```

132 #ifndef MOTHERBOARD
133 #define MOTHERBOARD BOARD_BIGTREE_SKR_PRO_V1_1 //GADGETANGEL was BOARD_MELZI
134 #endif
C:\Firmware\Ender3\SKR V1.1-PRO\Marlin-2.0.x\pio\build\BIGTREE_SKR_PRO\firmware.bin

```

Below the editor, the `OUTLINE` panel shows the contents of `sconsign.py37.dblite`, with the same two files (`firmware.bin` and `firmware.elf`) highlighted with a green box. A callout box points to the line of code defining the machine name:

```

132 #ifndef MOTHERBOARD
133 #define MOTHERBOARD BOARD_BIGTREE_SKR_PRO_V1_1 //GADGETANGEL was BOARD_MELZI
134 #endif
C:\Firmware\Ender3\SKR V1.1-PRO\Marlin-2.0.x\pio\build\BIGTREE_SKR_PRO\firmware.bin
136 // Name displayed in the LCD "Ready" message and Info menu
137 #define CUSTOM_MACHINE_NAME "Ender-3"

```

- Go to the next page.

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

```

    .sconsign.py37.dblite          132  #ifndef MOTHERBOARD
    firmware.bin                   133  #define MOTHERBOARD BOARD_BIGTREE_SKR_PRO_V1_1 //GADGETANG
    firmware.elf                   134  #endif
    longcmd-9e538521c014181d4772... 136  // Name displayed in the LCD "Ready" message and Info menu
    longcmd-d2cd97a14e80d0c95c59... 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
- > .sconsign.py37.dblite
- firmware.bin**
- > firmware.elf
- > longcmd-9e538521c014181d4772...
- > longcmd-d2cd97a14e80d0c95c59...

Reveal in File Explorer Shift+Alt+R Open in terminal Ctrl+Enter

Select for Compare

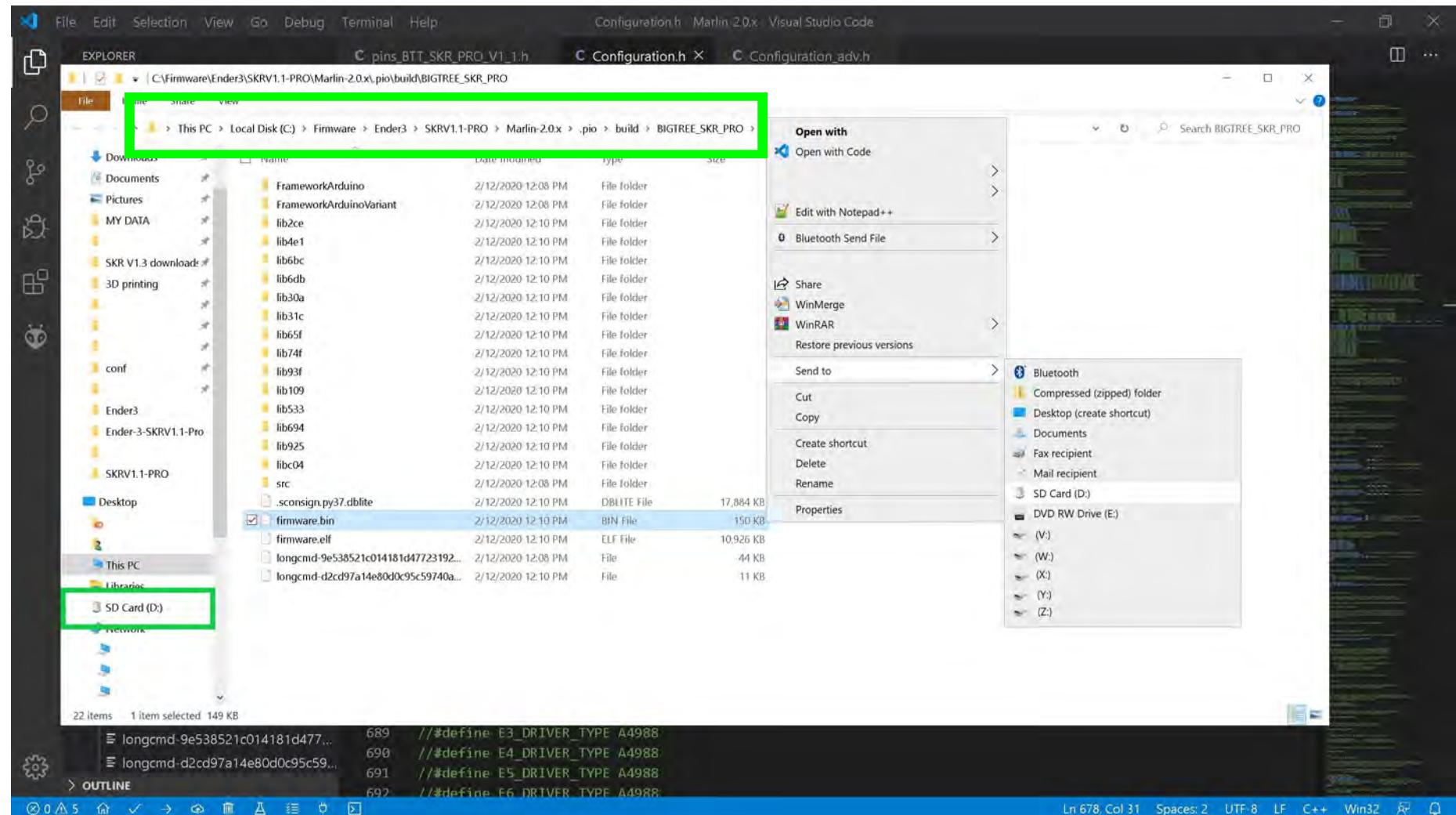
Cut Ctrl+X Copy Ctrl+C Copy Path Shift+Alt+C Copy Relative Path Ctrl+K Ctrl+Shift+C Rename F2 Delete Delete

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/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/](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/](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/](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://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/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/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/](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](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.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.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=55s>

### 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://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=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](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://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://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](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/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://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/docs/basics/install_platformio.html)
- <https://marlinfw.org/meta/download/>
- <https://www.pronterface.com/>
- [https://www.klipper3d.org/Sensorless\\_Homing.html](https://www.klipper3d.org/Sensorless_Homing.html)
- <https://learn.watterott.com/silentstepstick/comparison/>
- <https://reprap.org/wiki/Category:Reference>
- <https://reprap.org/wiki/Calibration>