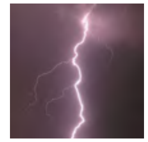
Stepper Driver Configurations for SKR V1.3 Board

By

@GadgetAngel

Based on Work By  **@rflulling**

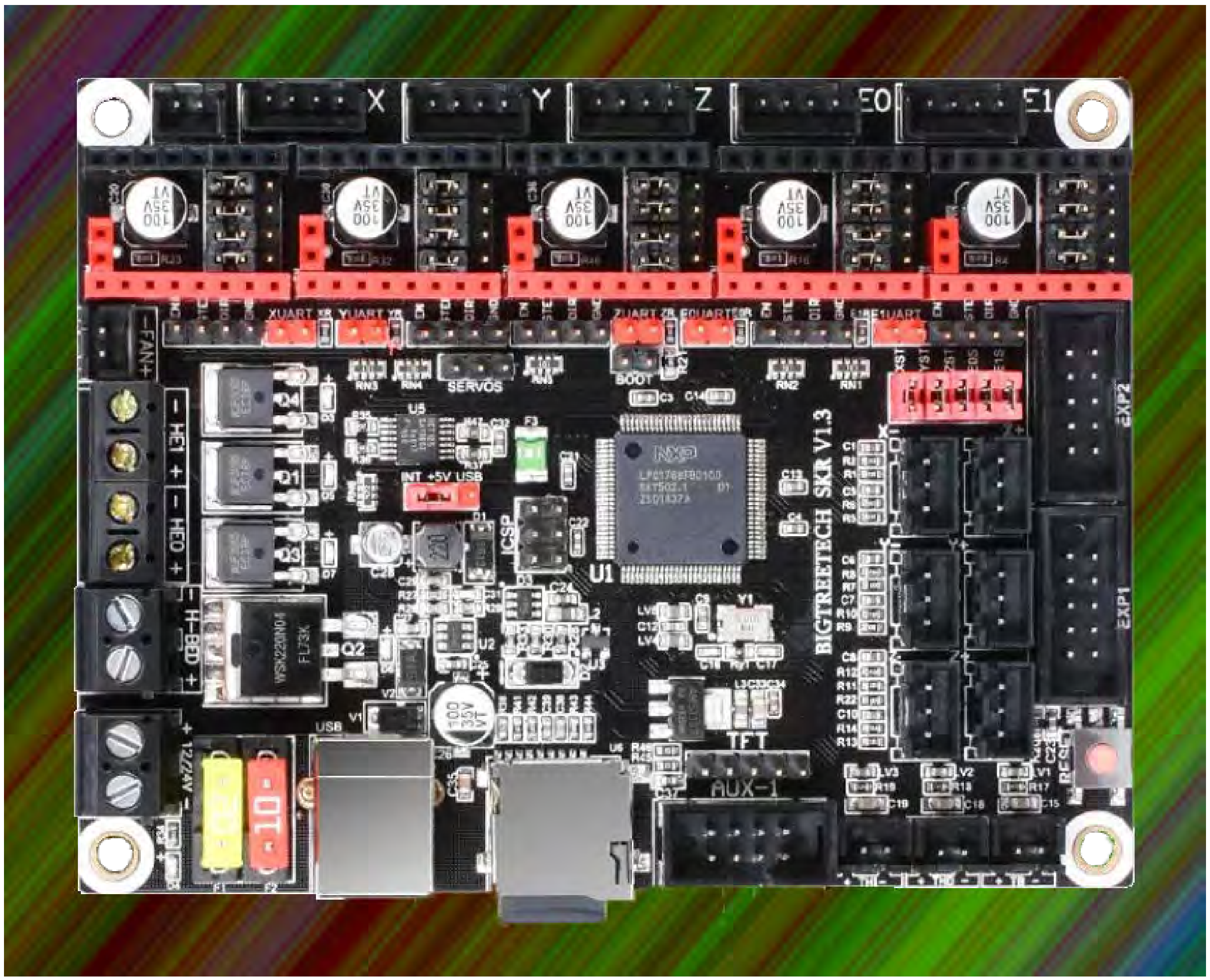


Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

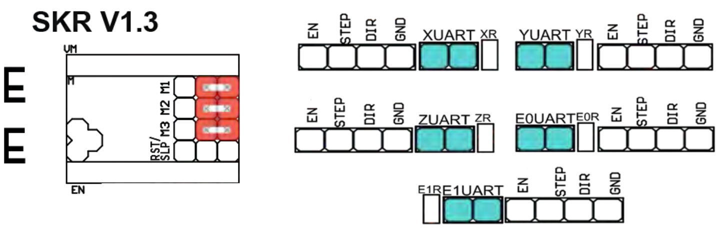
Preface

NOTE: EMF can destroy the SKR V1.3 motherboard (MCU). Static Electricity can destroy, so ALWAYS touch metal to ground yourself before inserting or removing micro-SD cards.*0

For Stepper Driver JUMPERS ( ), the reader must realize that JUMPERS should be set for ONLY ONE of THREE (3) modes. Check this manual for the mode(s) that your stepper driver or driver can support.*0

*0

STAND-ALONE NORMAL MODE

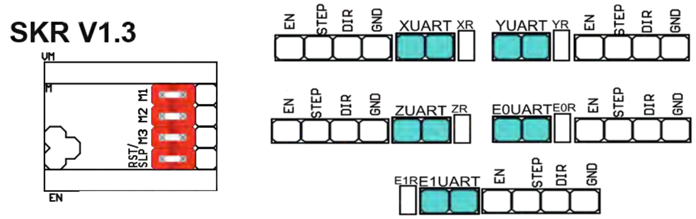


M1, M2 and M3 rows are used to set the desired micro-steps for your 3D Printer's stepper motors which are controlled by the installed stepper motor driver boards or drivers. You set M1, M2 and M3 rows by setting JUMPERS on the MCU board. See the driver's Chip Charts (Tables) located in this manual for the different M1, M2, and M3 settings to obtain the desired micro-stepping for this mode The RST/SLP row is not used for this mode on the SKR V1.3 board.

- this means a JUMPER maybe present

No JUMPERS on the UART-LINE per Axis.

SPI MODE



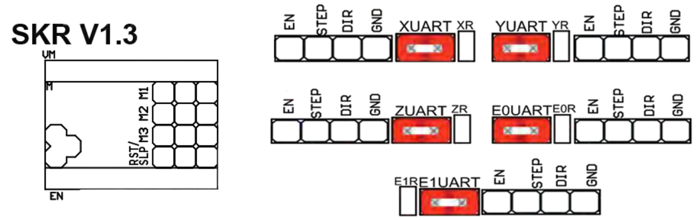
SPI mode is a software controlled mode.

- this means a JUMPER MUST be present

ONLY FOUR [4] (M1, M2, M3, and RST/SLP rows) SPI JUMPERS.

No JUMPERS on the UART-LINE per Axis.

UART MODE



UART mode is a software controlled mode.

- this means a JUMPER MUST be present

No (M1, M2, M3, and RST/SLP rows) JUMPERS.

ONE [1] JUMPER on the UART-LINE per Axis. Up to FOUR [4] JUMPERS are possible.

***0 based on Thomas White's 'Stepper jumpers PRO-GTR.pdf'**

https://reprap.org/wiki/Stepper_motor

What is a Stepper motor?

A stepper motor is one kind of electric motor and is used in 3D Printers. Stepper motors move a known interval for each pulse of power. These pulses of power are provided by a stepper motor driver and is referred to as a step. As each step moves the motor a known distance it makes them handy devices for repeatable positioning.

Properties

Stepper motors have a step angle. A full 360 degree circle divided by the step angle gives the number of steps per revolution. For example, 1.8 degree per full step is a common step size rating, equivalent to 200 steps per revolution. Most stepper motors used for 3D Printers have a step angle of 1.8 degrees It is sometimes possible to use motors with larger step angles, however for printing to be accurate, they will need to be geared down to reduce the angle moved per step, which may lead to a slower maximum speed.

Micro stepping

A stepper motor always has a fixed number of steps. Microstepping is a way of increasing the number of steps by sending a sine/cosine waveform to the coils inside the stepper motor. In most cases, micro stepping allows stepper motors to run smoother and more accurately.

Microstepping between pole-positiions is made with lower torque than with full-stepping, but has much lower tendency for mechanical oscillation around the step-positions and you can drive with much higher frequencies.

If your motors are near to mechanical limitations and you have high friction or dynamics, microsteps don't give you much more accuracy over half-stepping. When your motors are 'overpowered' and/or you don't have much friction, then microstepping can give you much higher accuracy over half-stepping. You can transfer the higher positioning accuracy to moving accuracy too.

Stall Detection and Sensor-less Homing

URL1: https://duet3d.dozuki.com/Wiki/Stall_detection_and_sensorless_homing

URL2: https://www.klipper3d.org/Sensorless_Homing.html#:~:text=Homing%20and%20Tuning-,Sensorless%20Homing,the%20stepper%20motor%20lose%20steps.

Overview

The TMC2209, TMC2130, TMC5160 and TMC5161 drivers used on SKR V1.3 (MCU) support the stallGuard™ feature. This feature allows the driver to detect motor stalls under some circumstances. Stall detection may be useful for detecting when a motor has skipped steps due to the nozzle hitting an obstruction, and for homing the printer without using endstop switches. Marlin 2.0.x and later provides facilities for configuring and using stall detection.

Requirements Needed to Make Sensor-less Homing Work:

#1: You MUST be using a stepper driver board that has the stallGuard™ feature like the following: TMC2130 in SPI mode, TMC2209 in UART mode, TMC5160 in SPI mode and TMC5161 in SPI mode. NOTE: The TMC2208 and TMC2225 DO NOT have stallGuard™ feature and CAN NOT be used for Senor-less Homing.

#2: A DIAG (DIAG/DIAG0/DIAG1) PIN MUST be installed on the stepper driver board or driver.

#3: A way to connect the DIAG PIN to the MCU Endstops or Limit switches. Most 32-bit MCU boards do this by electrical trace in the MCU PCB, others do it with a JUMPER Block and non-32-bit MCU boards will require external DuPont Jumper cables to be used.

#4: Marlin Firmware or other Firmware that can enable Sensor-less Homing on the 3D Printer. I know for Marlin >2.0.x Firmware, the sensor-less homing by default is disabled and MUST be enabled (un-commented) by the user.

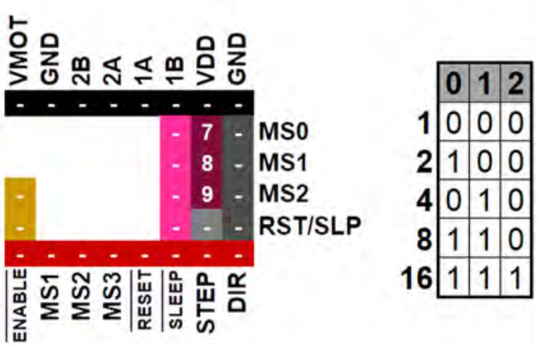
#5: Physical removal of Endstops/Limit switches MAYBE needed to make Sensor-less Homing work properly on the Axis. If using TMC2209 driver and you WANT sensor-less homing then the physical enstops MUST be REMOVED. If using TMC2209 driver and you DO NOT WANT sensor-less homing then remove the 'ST' JUMPER for the Axis from the 'ST' JUMPER block.

#6: If using a filament runout sensor, remove the DIAG PIN from the driver board that is installed in the Extruder socket (E0 and/or E1 and/or E2).

POLOLU A4988

Stand-alone Mode

base of card

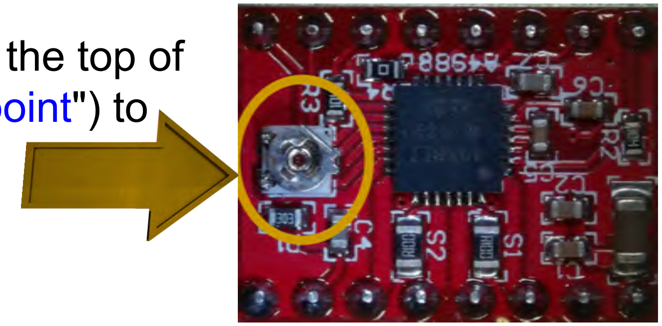


fans/heaters/usb side

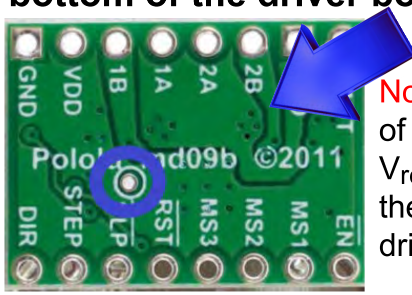
STATUS: Not Yet Tested

See Appendix D for legend

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



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



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

See Appendix A for instructions on how to set the V_{ref} on a driver board.

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
Pololu A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	2 Phase
	High	Low	Low	Half step	1-2 Phase
	Low	High	Low	Quarter step	W1-2 Phase
	High	High	Low	Eighth step	2W1-2 Phase
	High	High	High	Sixteenth step	4W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor) $= 0.1\Omega$	$I_{MAX} = V_{ref} / (8 * R_S)$	$V_{ref} = 8 * I_{MAX} * R_S$

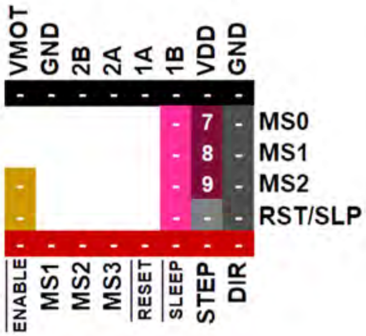
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

POLOLU A4988

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
XXA (peak)	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX} = FORMULA	V_{ref} = FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

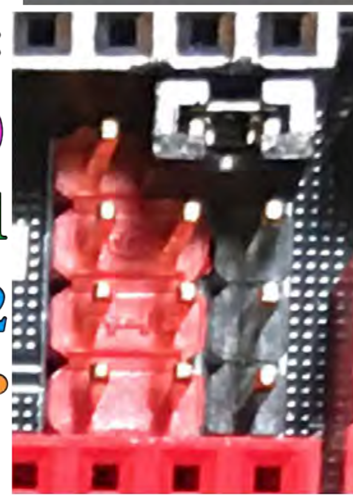
Rows:

MS0

MS1

MS2

RST/SLP



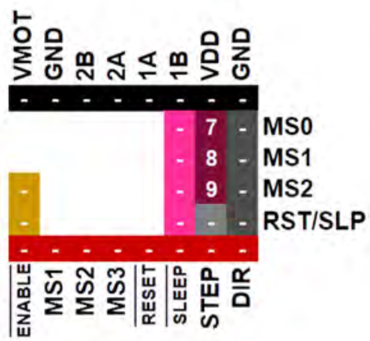
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

POLOLU A4988

Stand-alone Mode



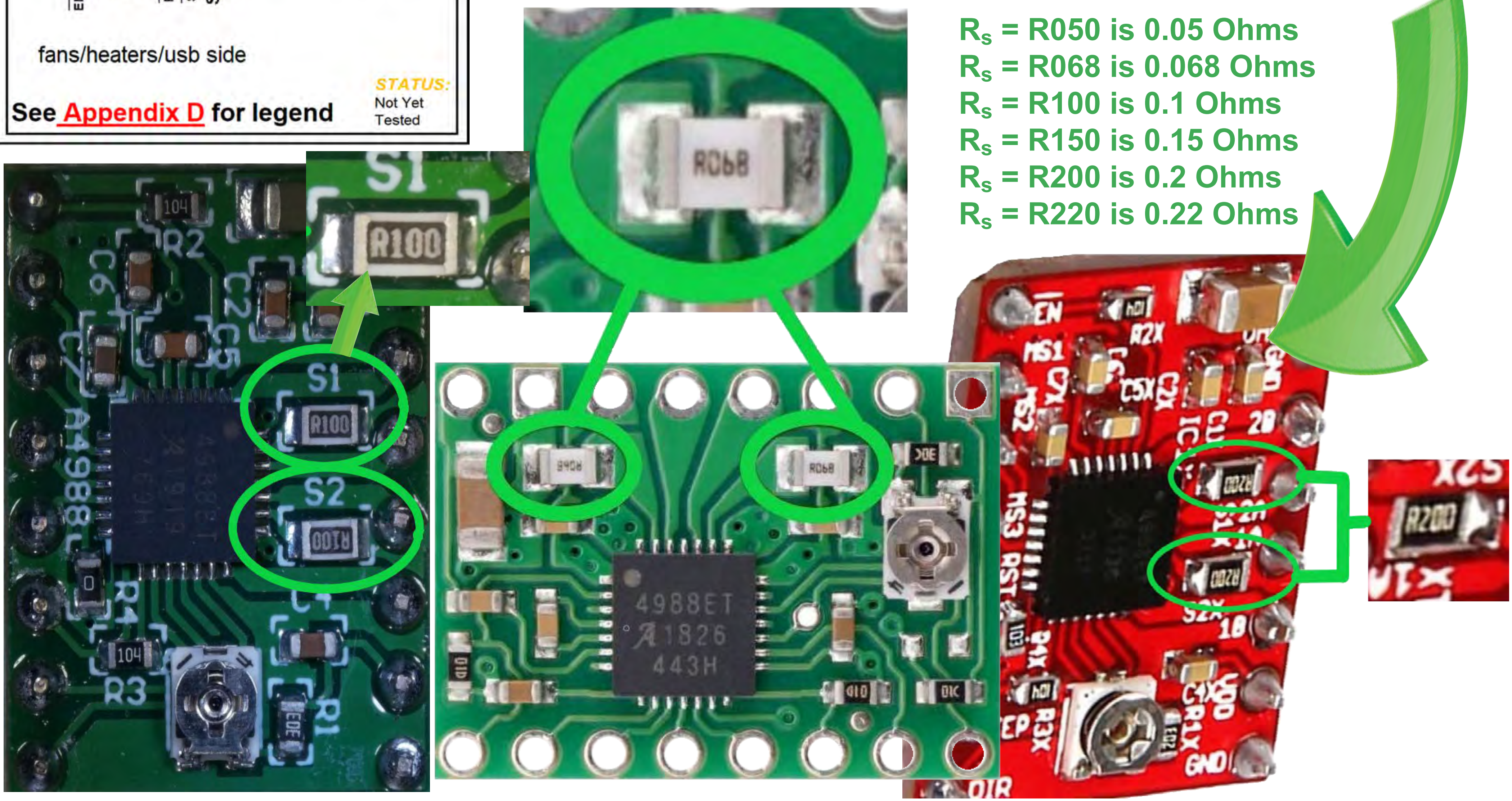
	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	1	1	1

See Appendix D for legend

STATUS: Not Yet Tested

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

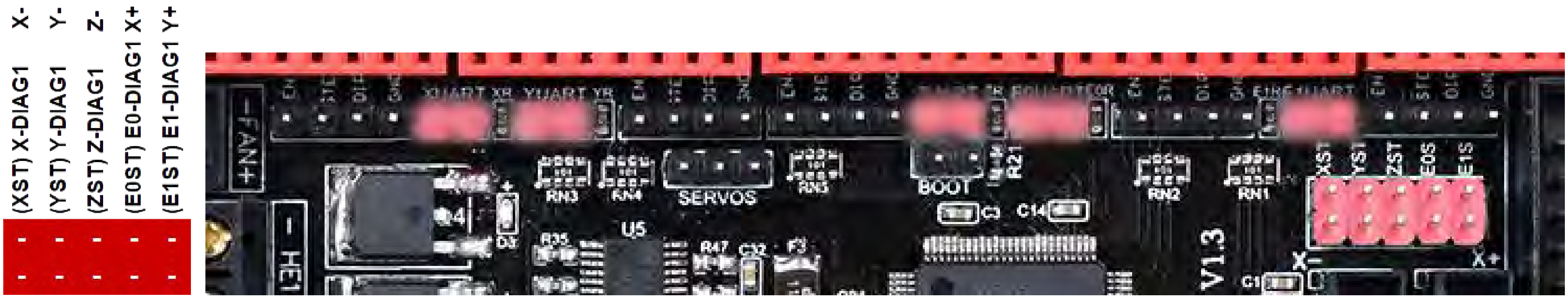
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms



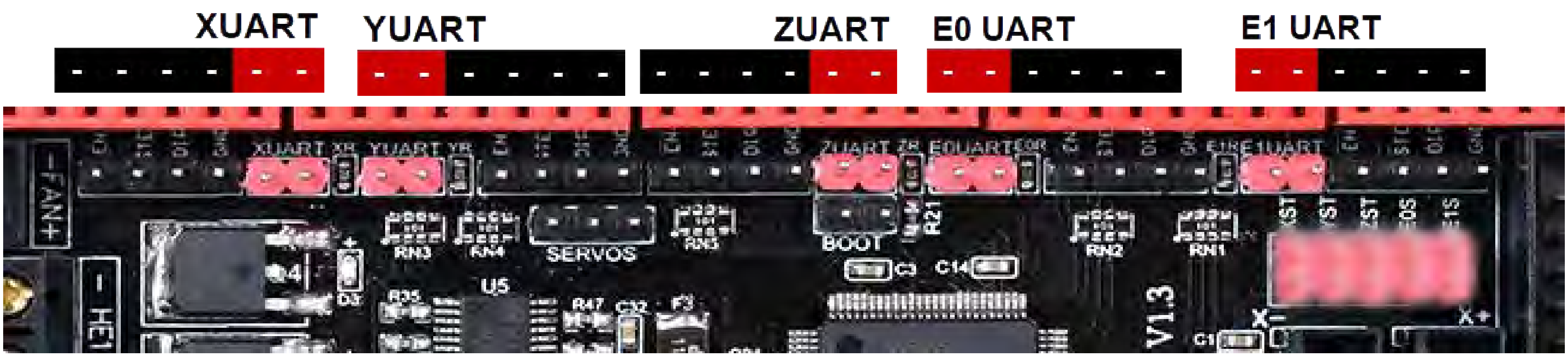
Stand-alone Mode

POLOLU A4988

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.



Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

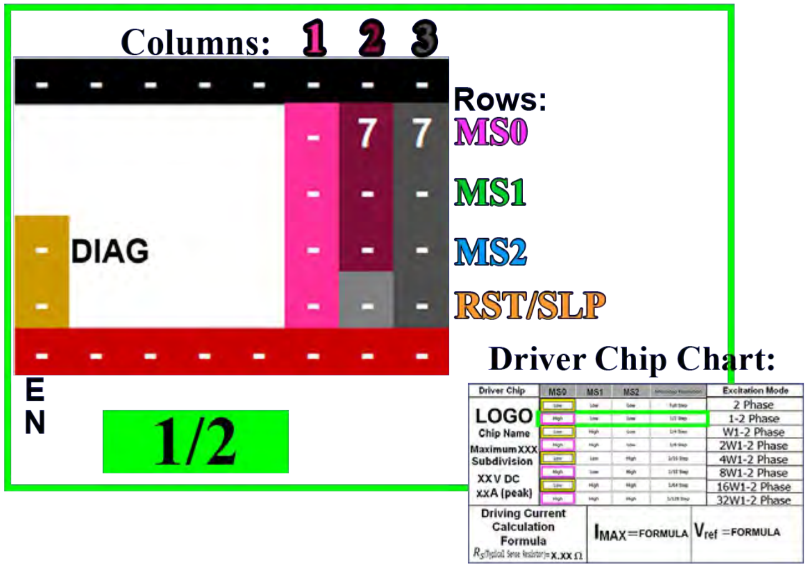


Stand-alone Mode

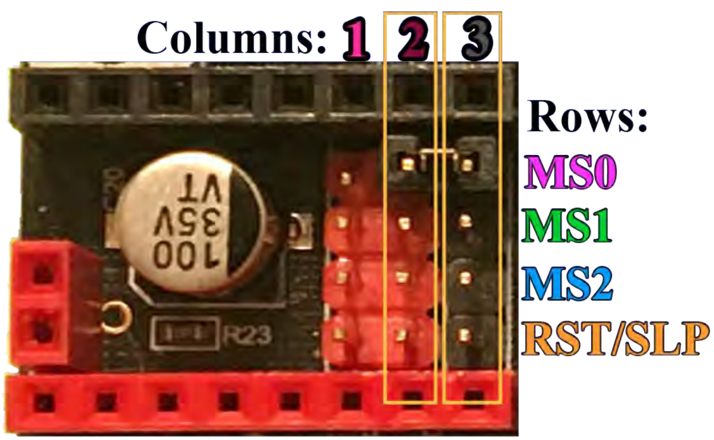
POLOLU A4988

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

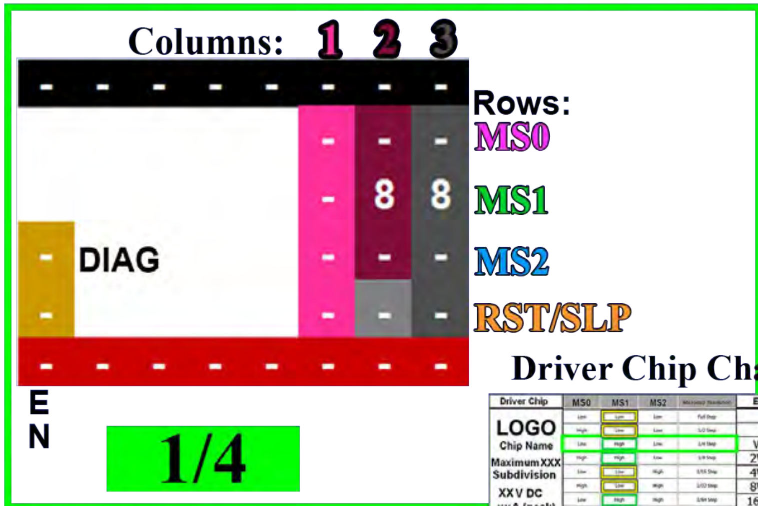
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

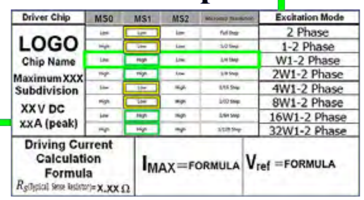
Stand-alone Mode

POLOLU A4988

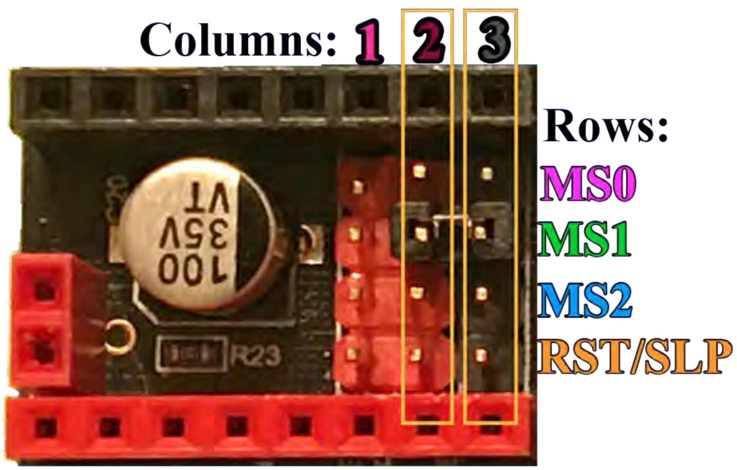
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

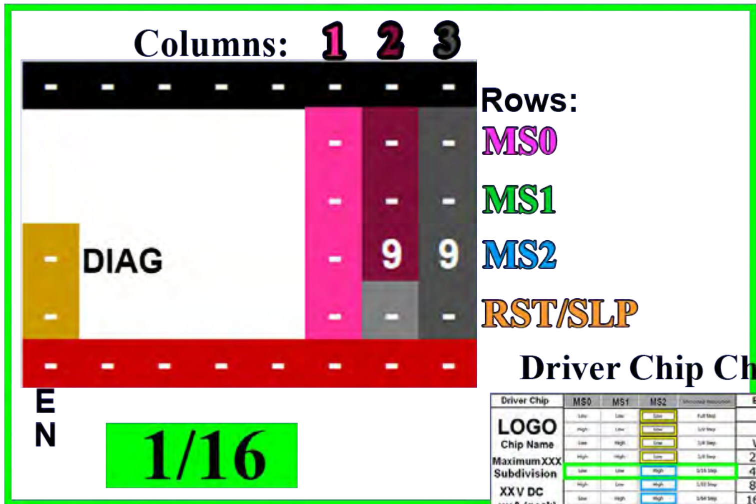
Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Driver Chip Chart:		Meaning:
High	→	set **Jumper** between column 2 and column 3 on the MS1 row
Low	→	No **Jumper** Set

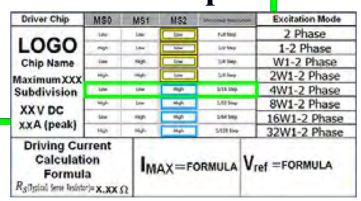
Stand-alone Mode

POLOLU A4988

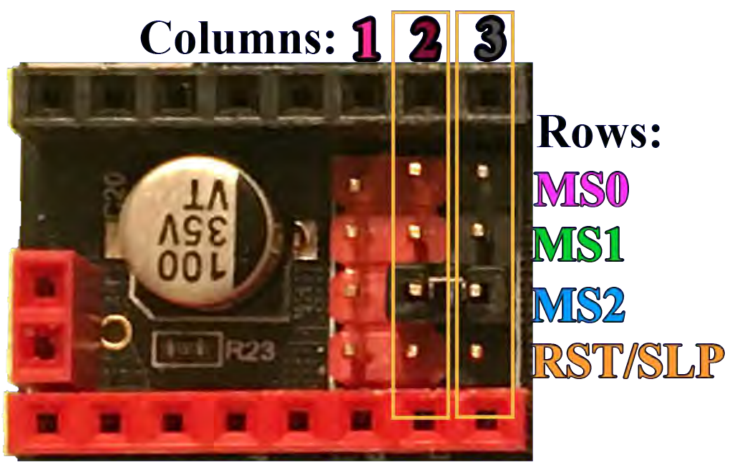
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Driver Chip Chart:

Meaning:

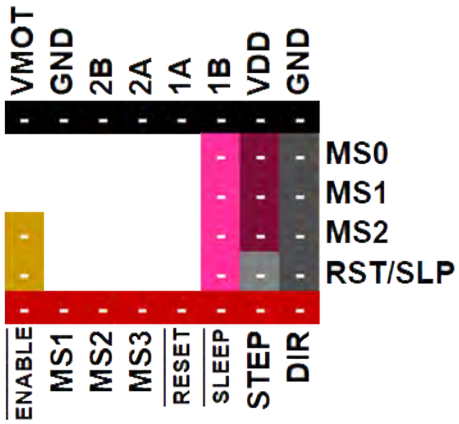
High → set Jumper between column 2 and column 3 on the MS2 row

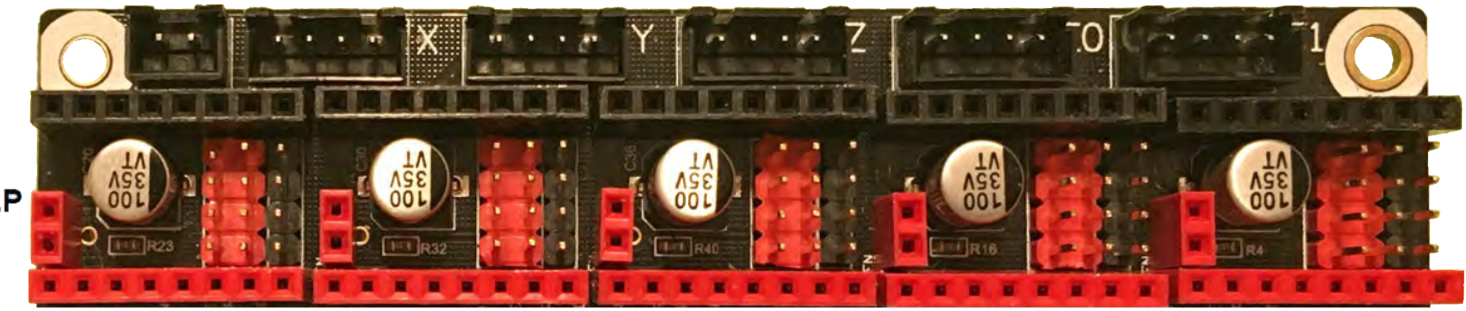
Low → No Jumper Set

Stand-alone Mode

POLOLU A4988

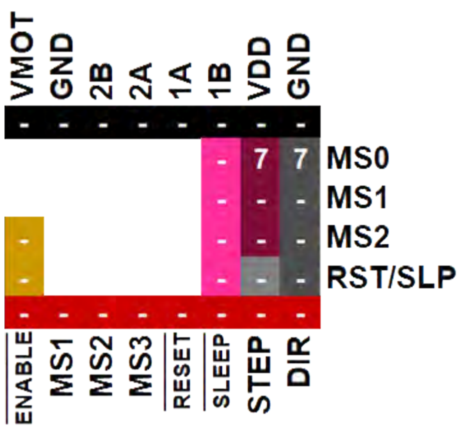
STEP

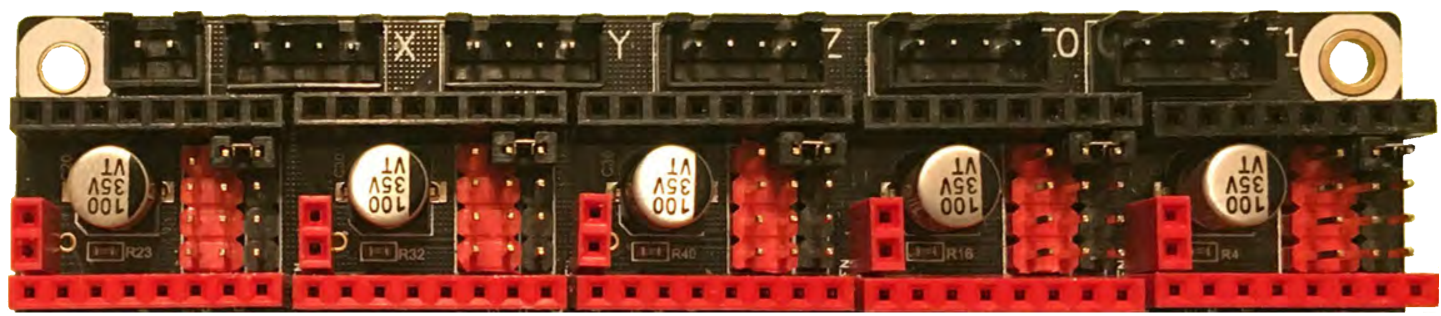




See Appendix D for legend

1 / 2



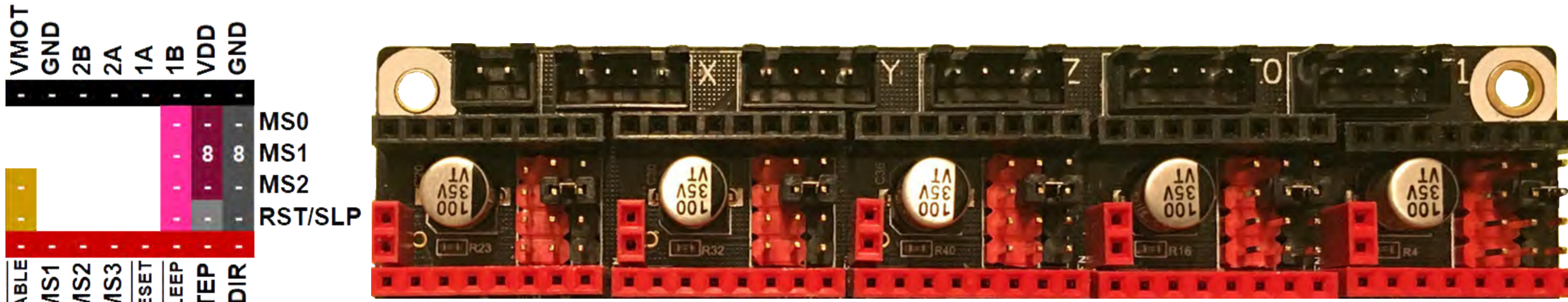


See Appendix D for legend

Stand-alone Mode

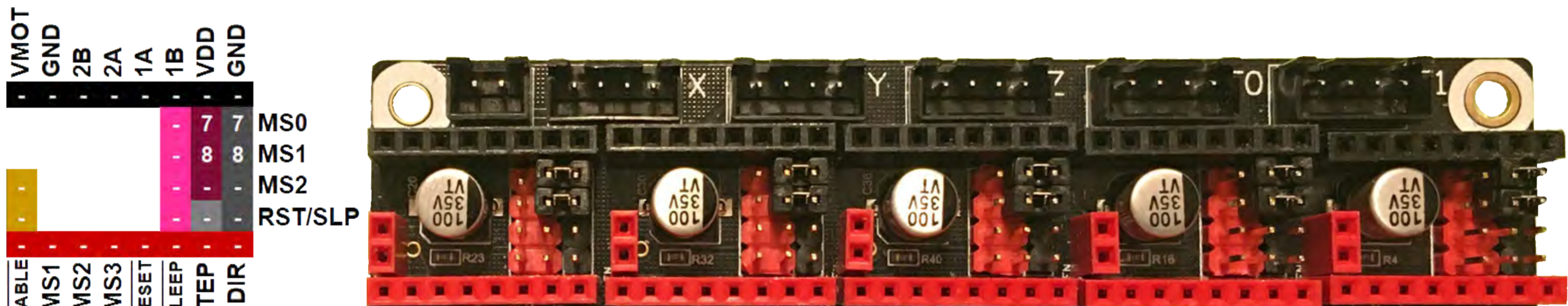
POLOLU A4988

1 / 4



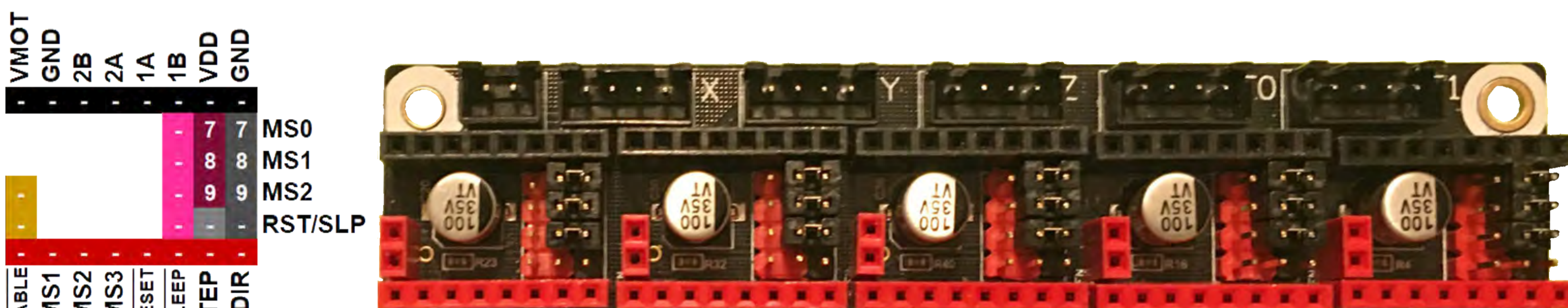
See Appendix D for legend

1 / 8



See Appendix D for legend

1 / 16

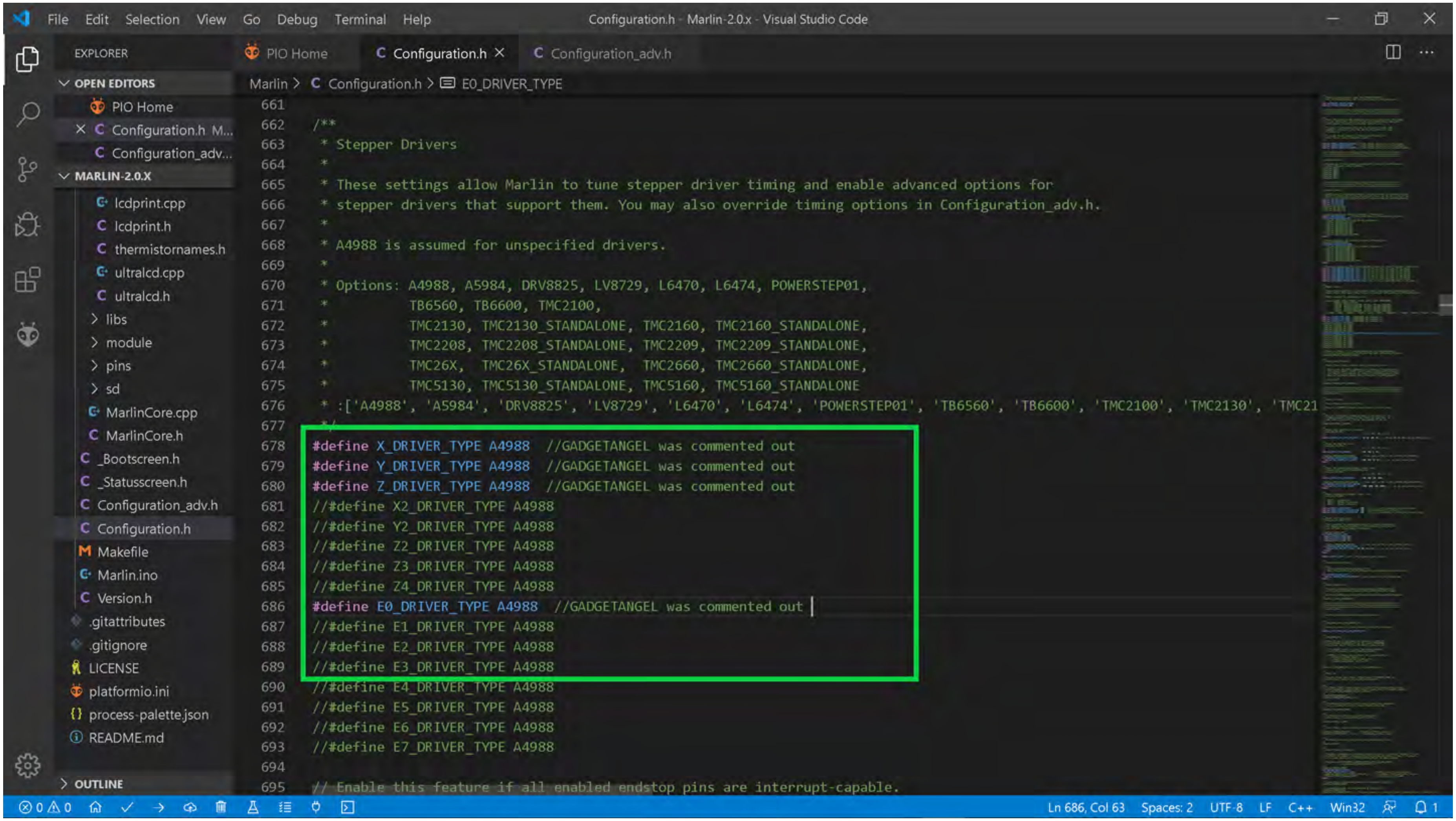


See Appendix D for legend

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

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

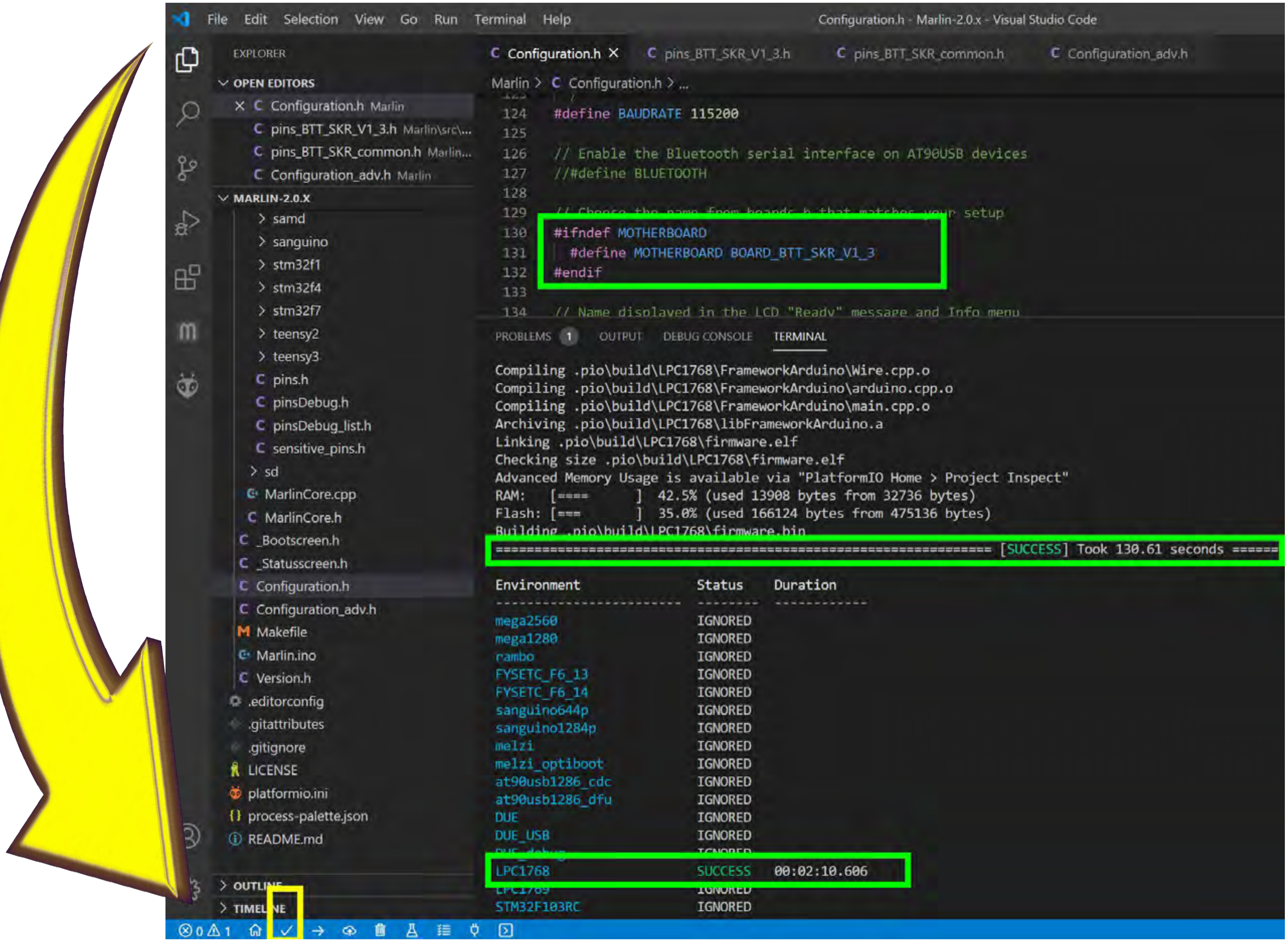
- Change the stepper motor drivers so that Marlin knows you are using POLOLU A4988 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU A4988 drivers. When two "//" appear at the beginning of a line that means that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



- Go to the next page.

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

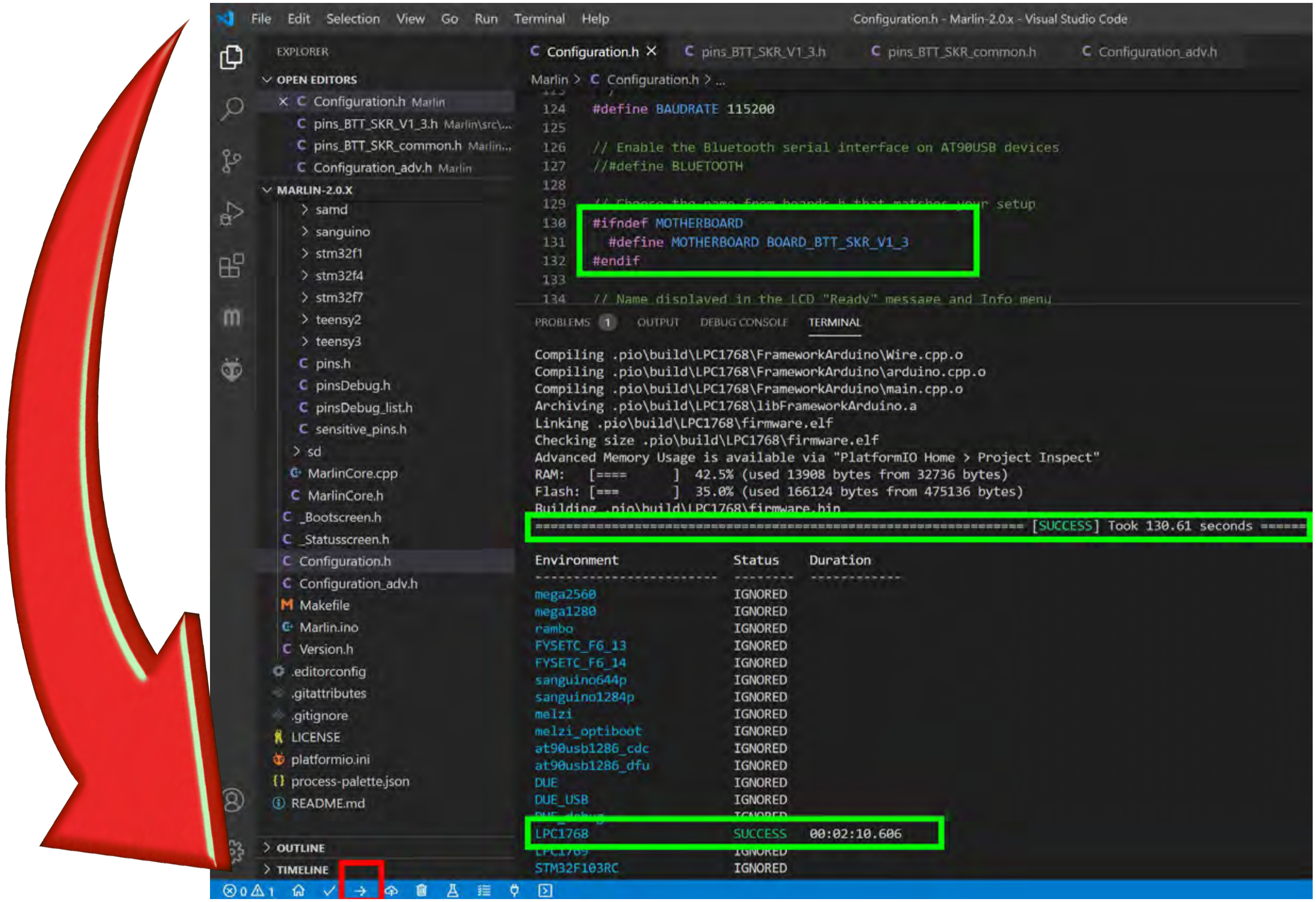
- The end of Marlin setup for POLOLU A4988 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

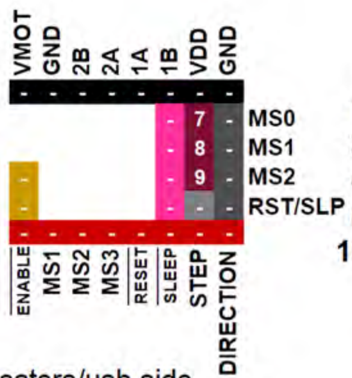
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

Stand-alone Mode

base of card



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	1	1	1

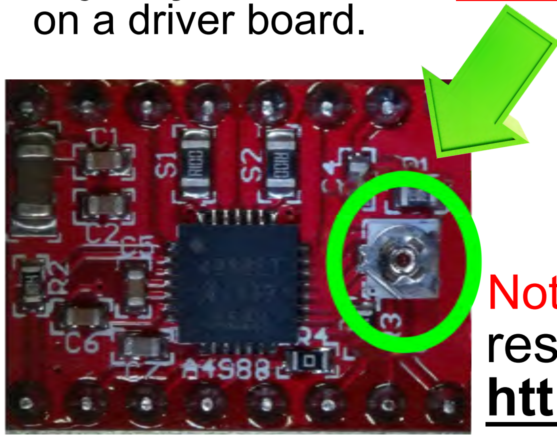
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

BIQU A4988

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



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

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
BIQU® A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	2 Phase
	High	Low	Low	Half step	1-2 Phase
	Low	High	Low	Quarter step	W1-2 Phase
	High	High	Low	Eighth step	2W1-2 Phase
	High	High	High	Sixteenth step	4W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)$=0.1\Omega$	$I_{MAX}=V_{ref}/(8*R_S)$	$V_{ref}=8*I_{MAX}*R_S$

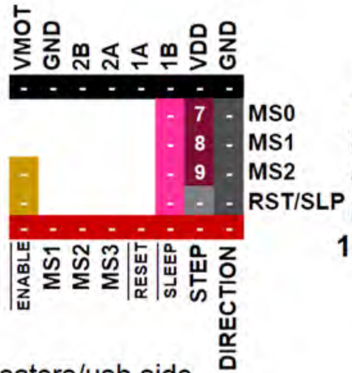
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU A4988

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Stand-alone Mode

base of card



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

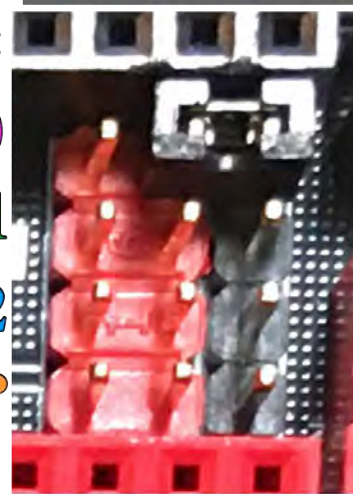
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

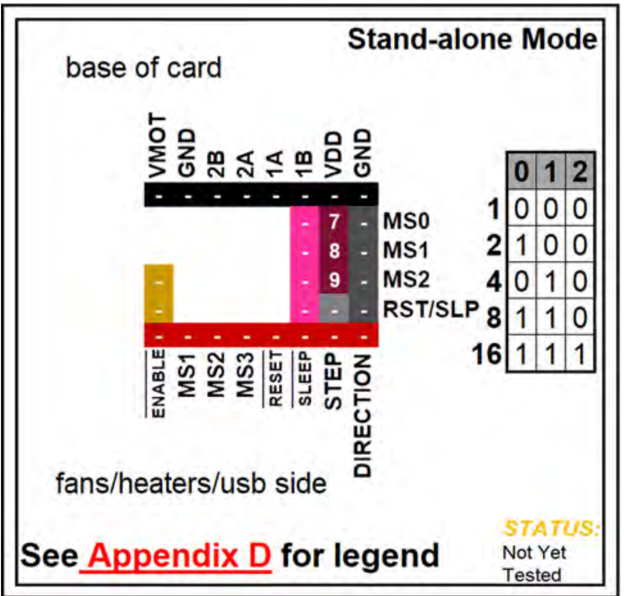
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
XXA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

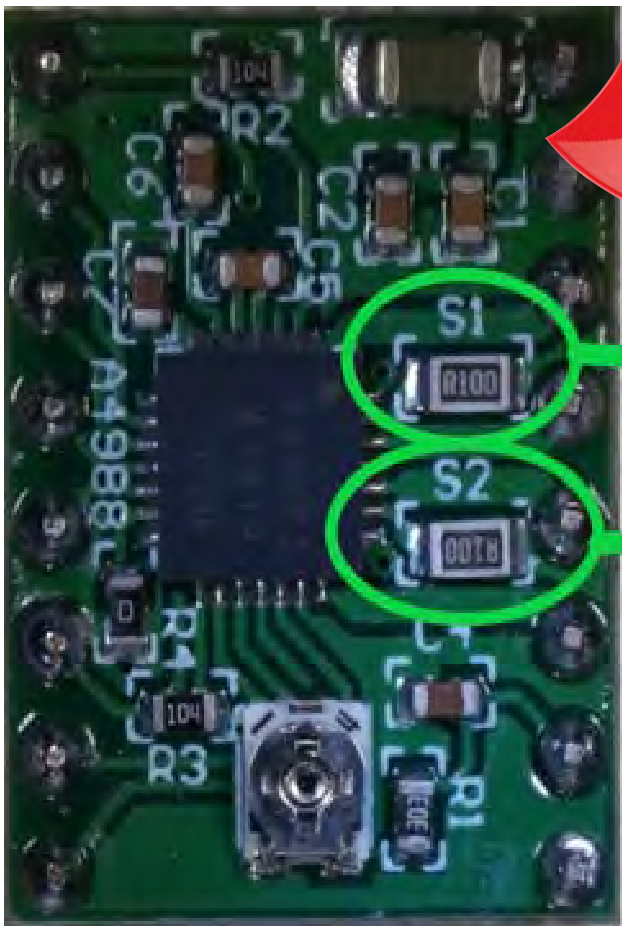
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

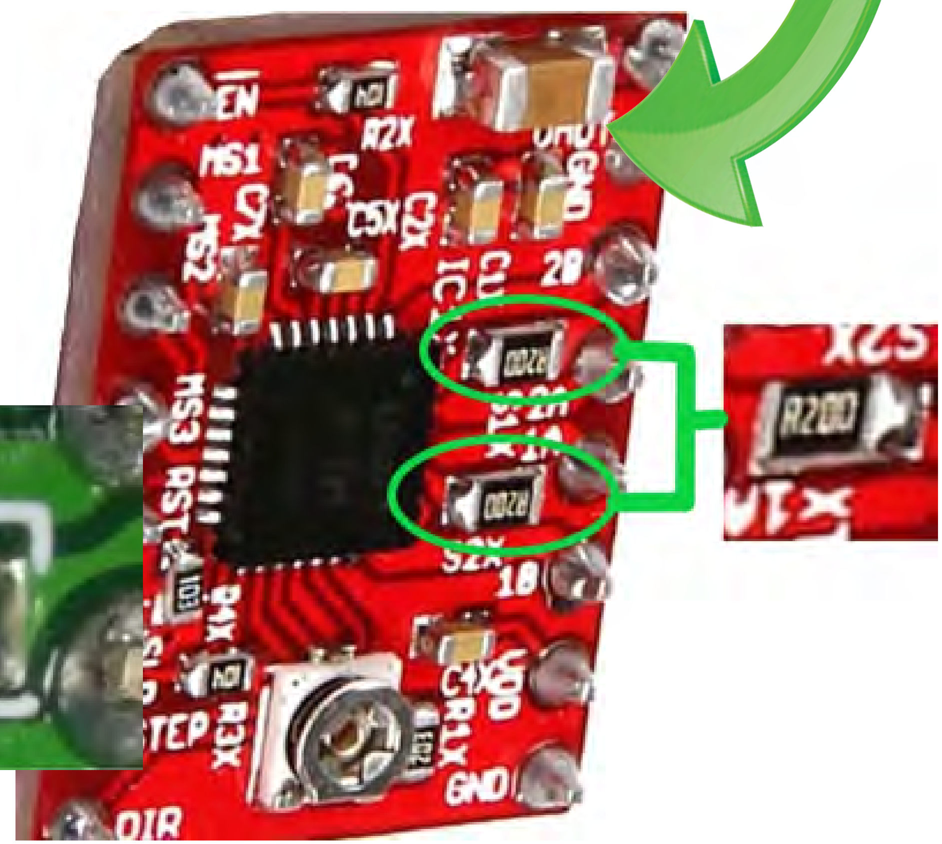
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

BIQU A4988



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





R_S = R050 is 0.05 Ohms; R_S = R068 is 0.068 Ohms

R_S = R100 is 0.1 Ohms; R_S = R150 is 0.15 Ohms

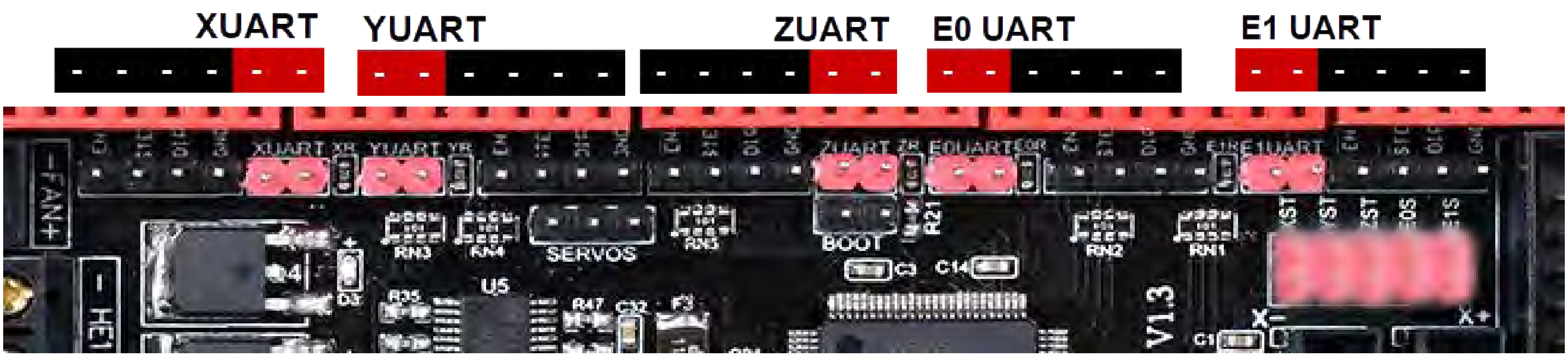
R_S = R200 is 0.2 Ohms; R_S = R220 is 0.22 Ohms

Stand-alone Mode

BIQU A4988

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

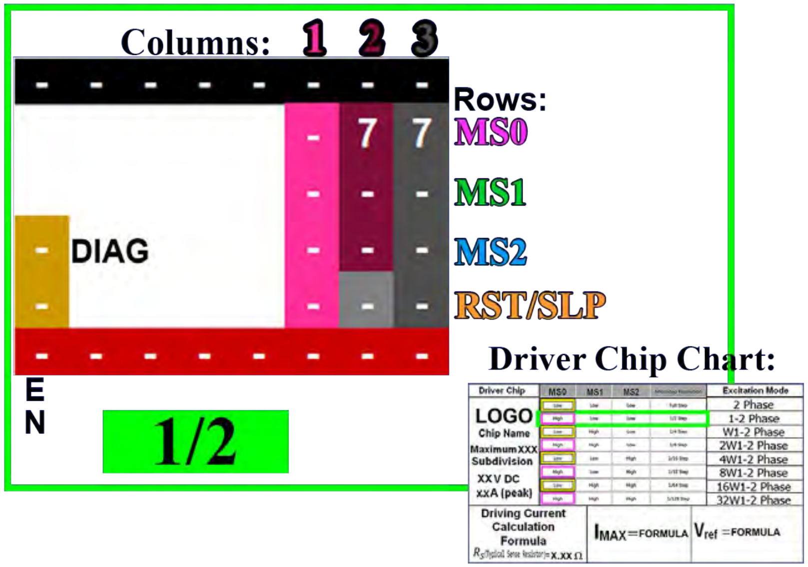


Stand-alone Mode

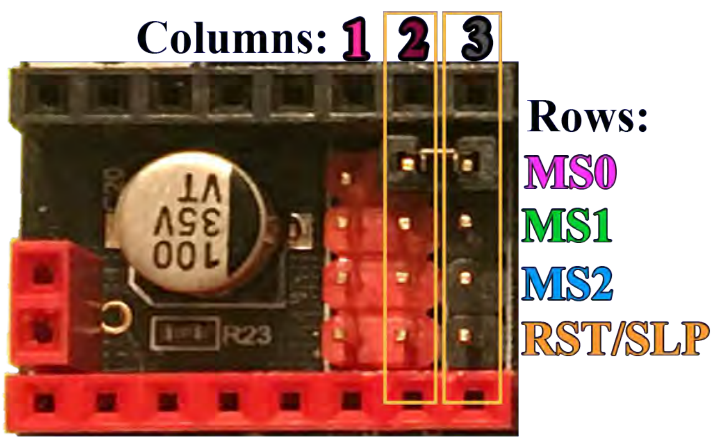
BIQU A4988

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Meaning:

Driver Chip Chart:

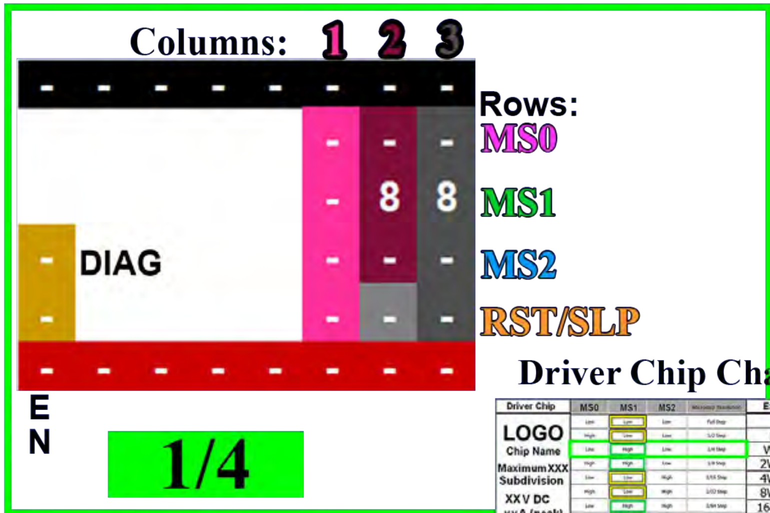
High → **set Jumper between column 2 and column 3 on the MS0 row**

Low → **No Jumper Set**

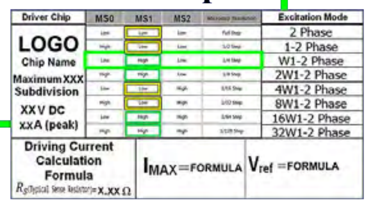
Stand-alone Mode

BIQU A4988

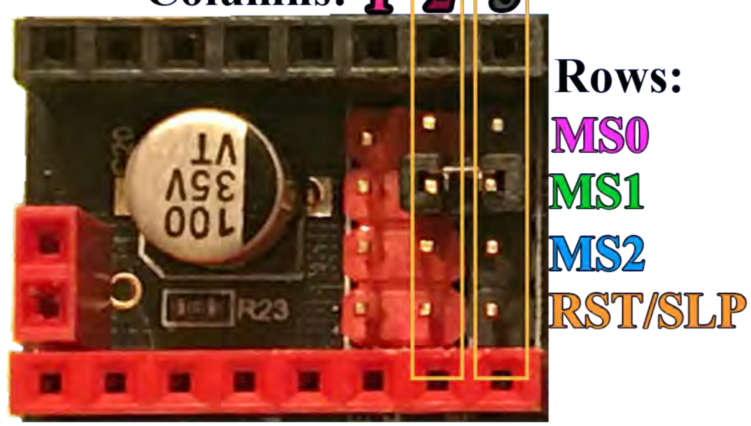
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

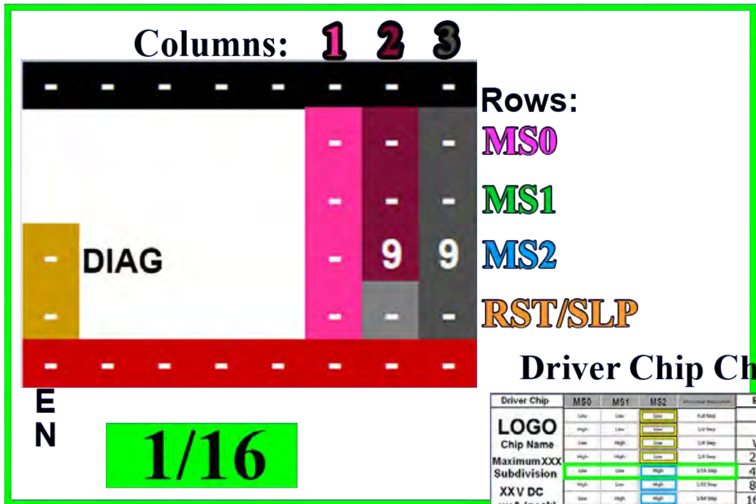
Meaning:

Driver Chip Chart:	Meaning:
High	set **Jumper** between column 2 and column 3 on the MS1 row
Low	No **Jumper** Set

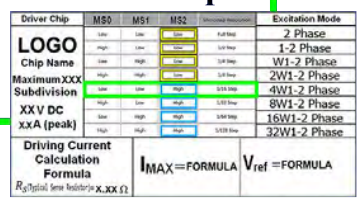
Stand-alone Mode

BIQU A4988

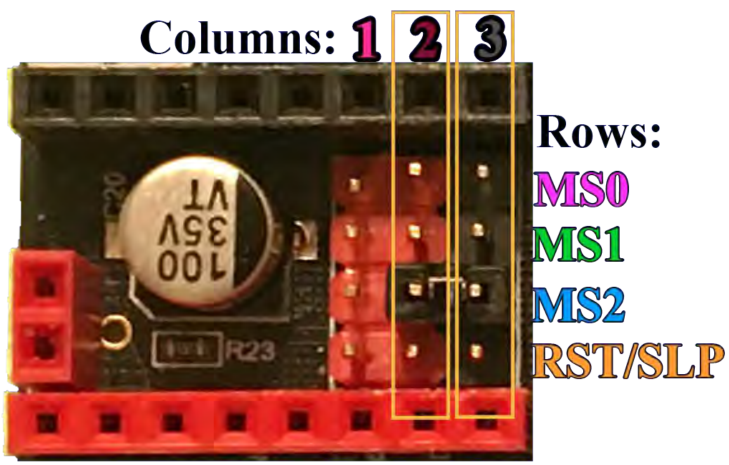
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

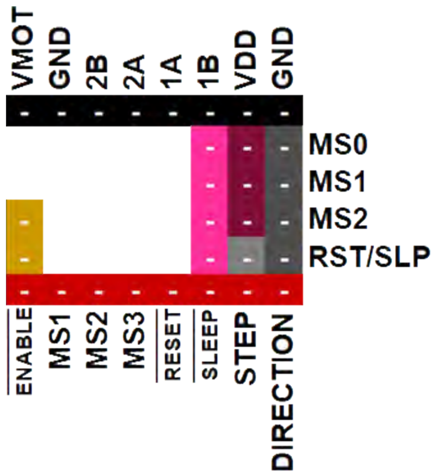
Driver Socket Representation: 9 9 → Driver Chip Chart: High

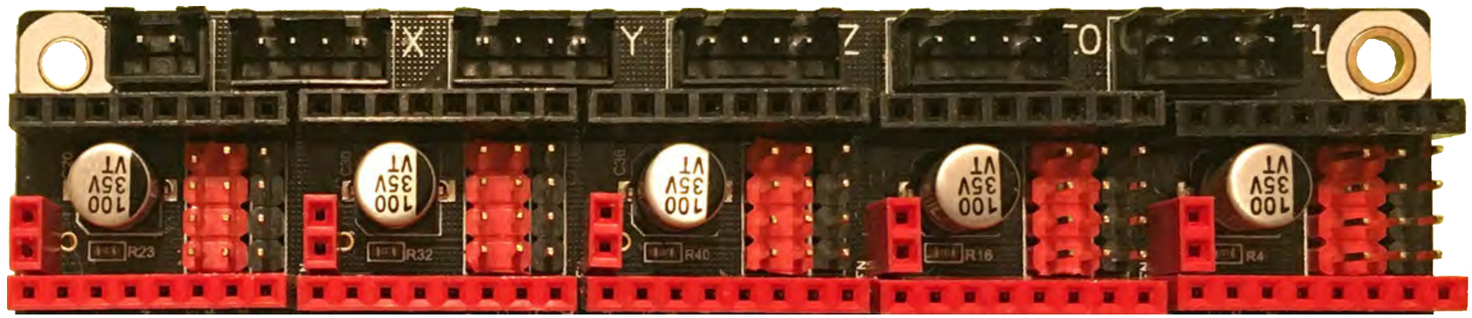
Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS2 row
Low →	No Jumper Set

Stand-alone Mode

BIQU A4988

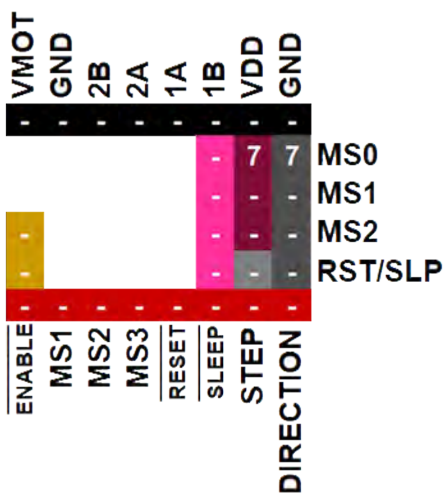
STEP

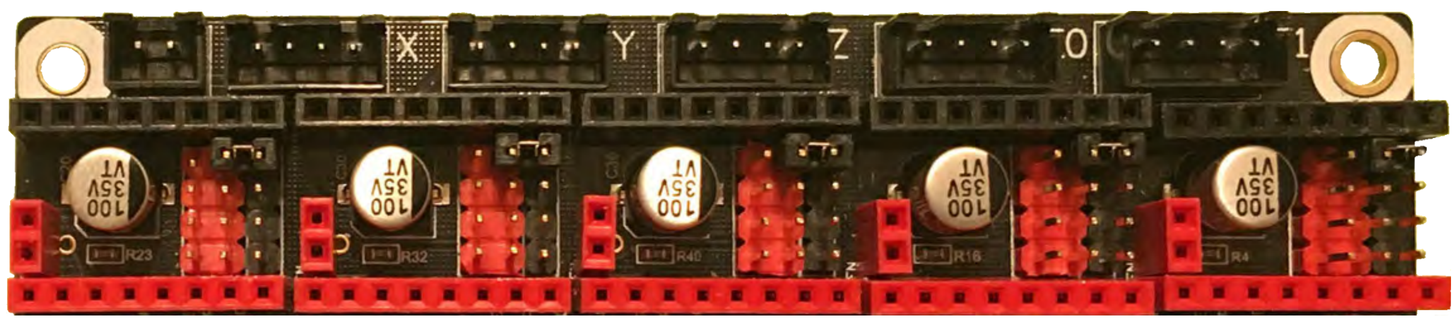




See Appendix D for legend

1 / 2



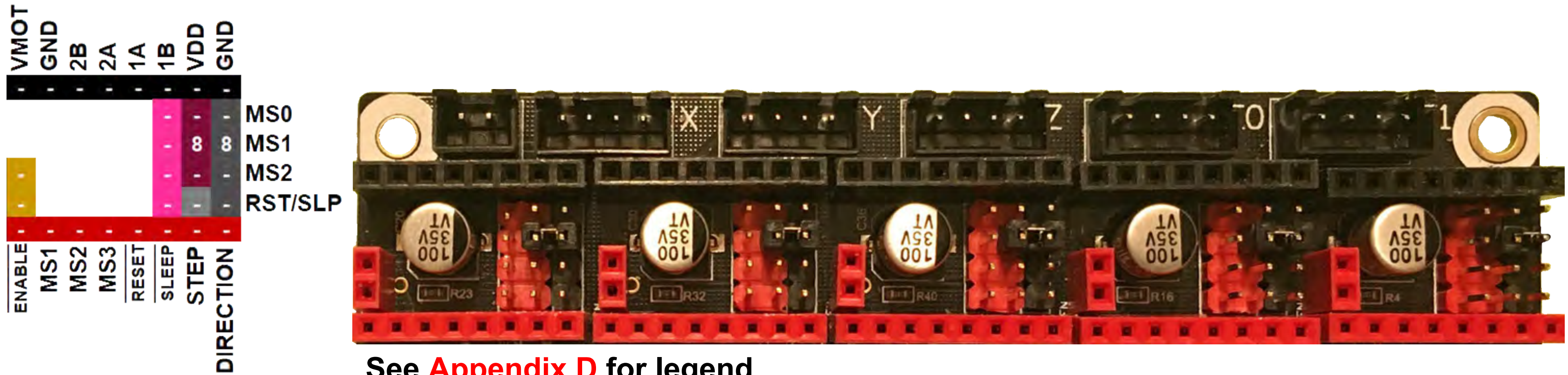


See Appendix D for legend

Stand-alone Mode

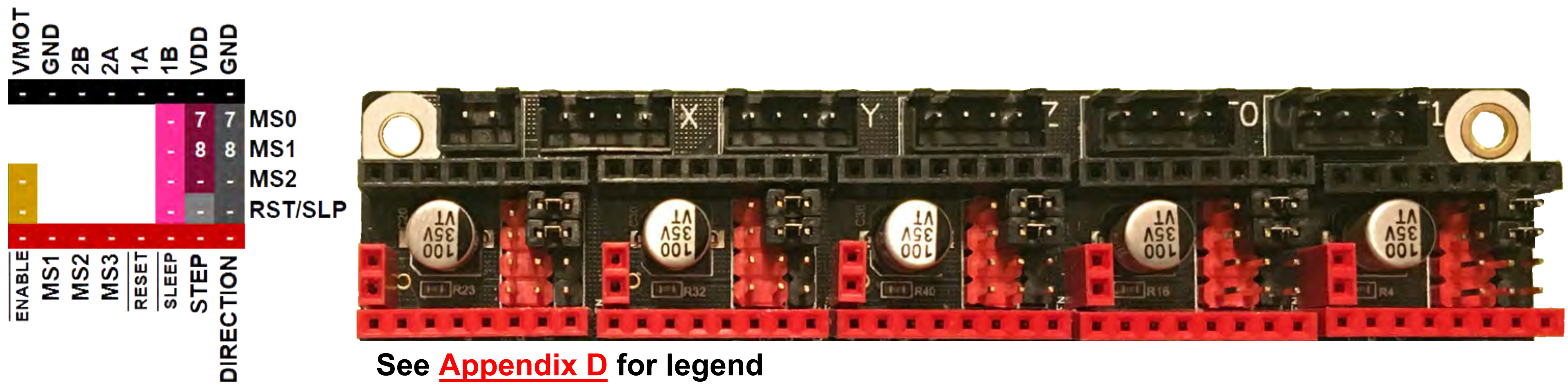
BIQU A4988

1 / 4



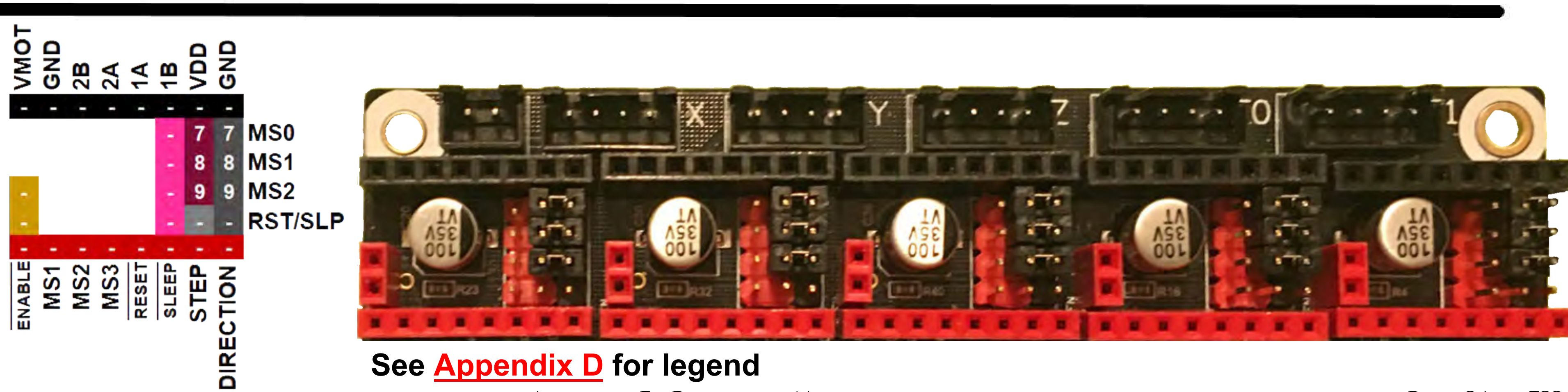
See Appendix D for legend

1 / 8



See Appendix D for legend

1 / 16

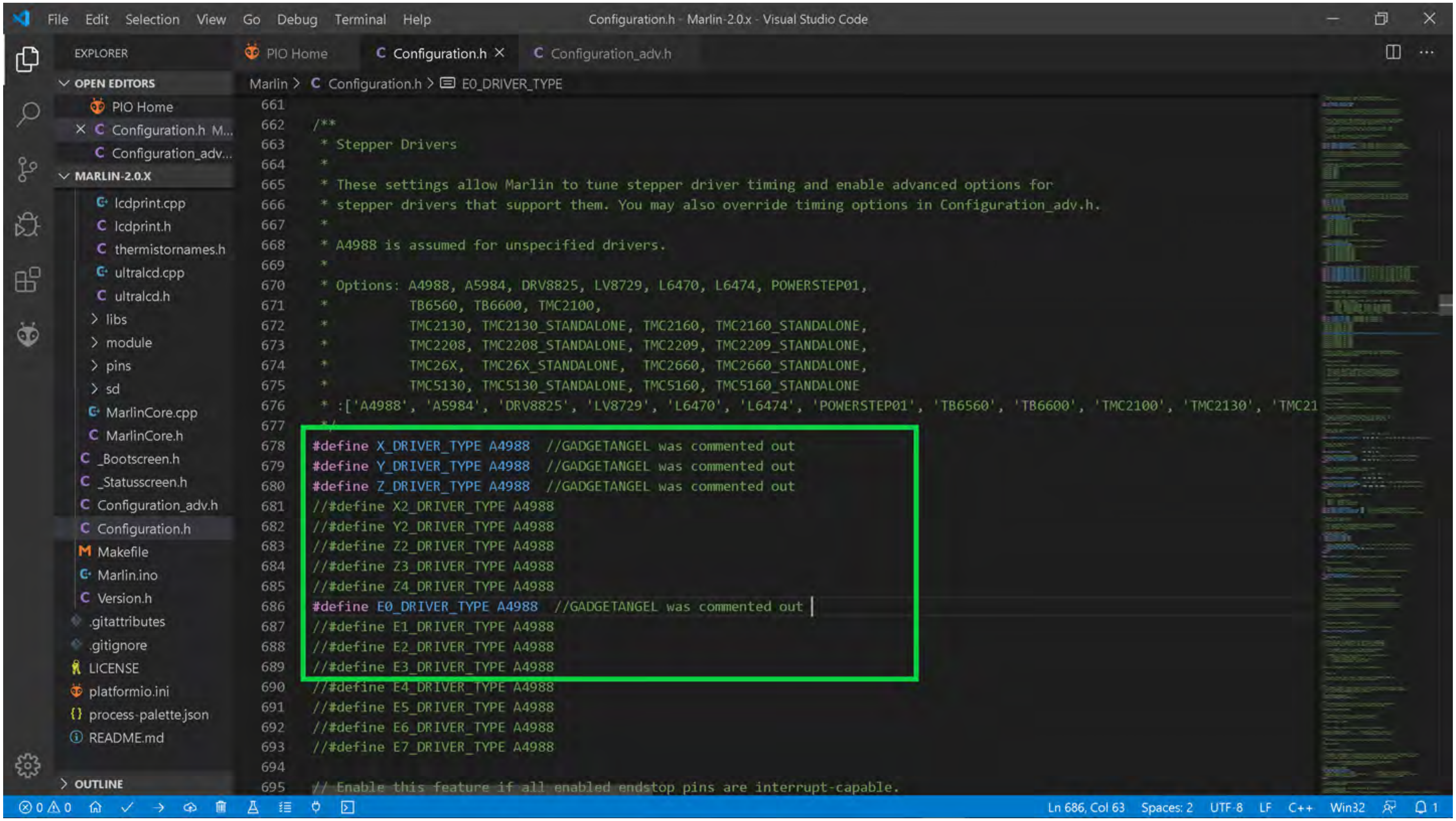


See Appendix D for legend

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

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

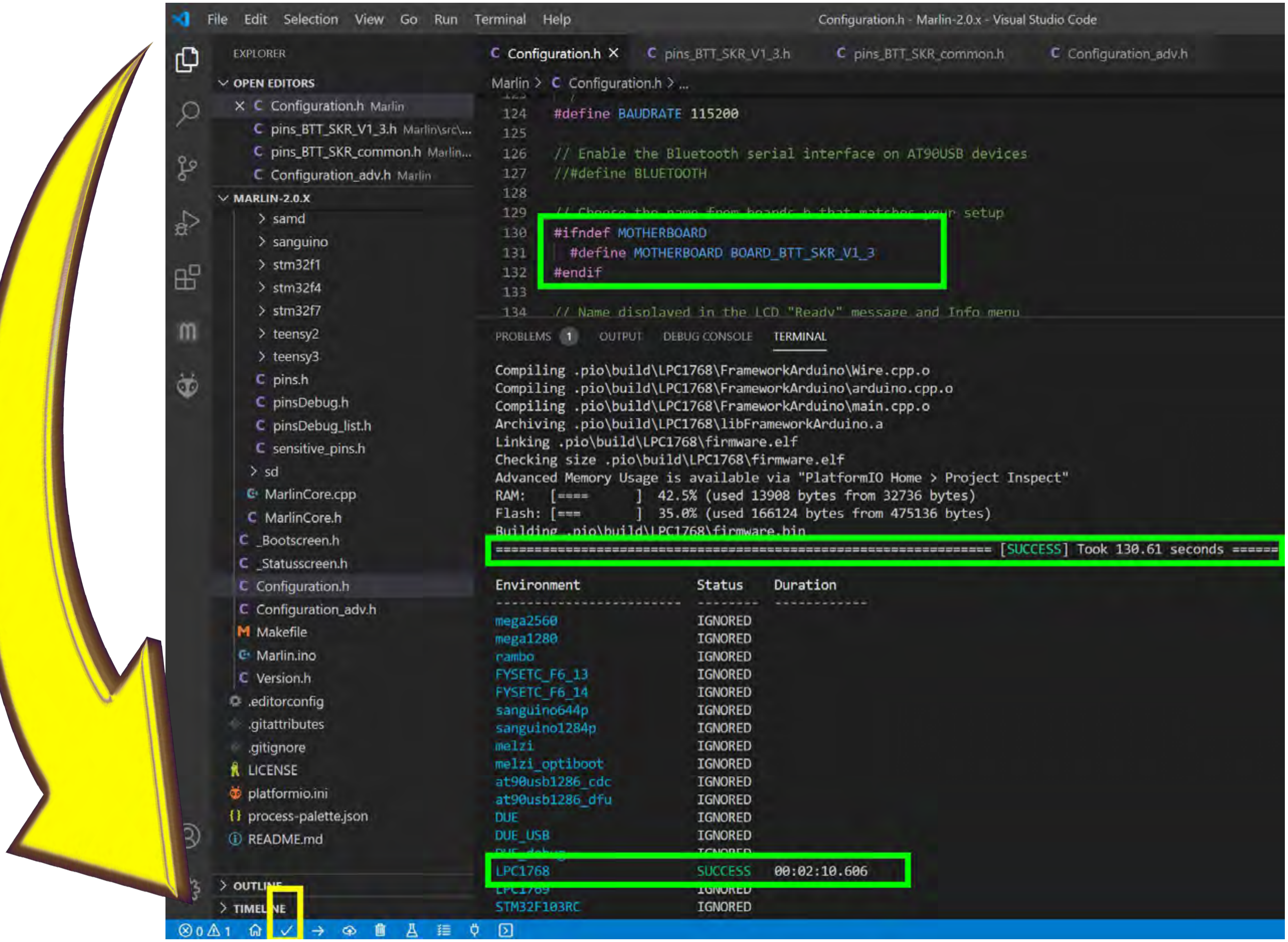
- Change the stepper motor drivers so that Marlin knows you are using BIQU A4988 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU A4988 drivers. When two "//" appear at the beginning of a line that means that line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



- Go to the next page.

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

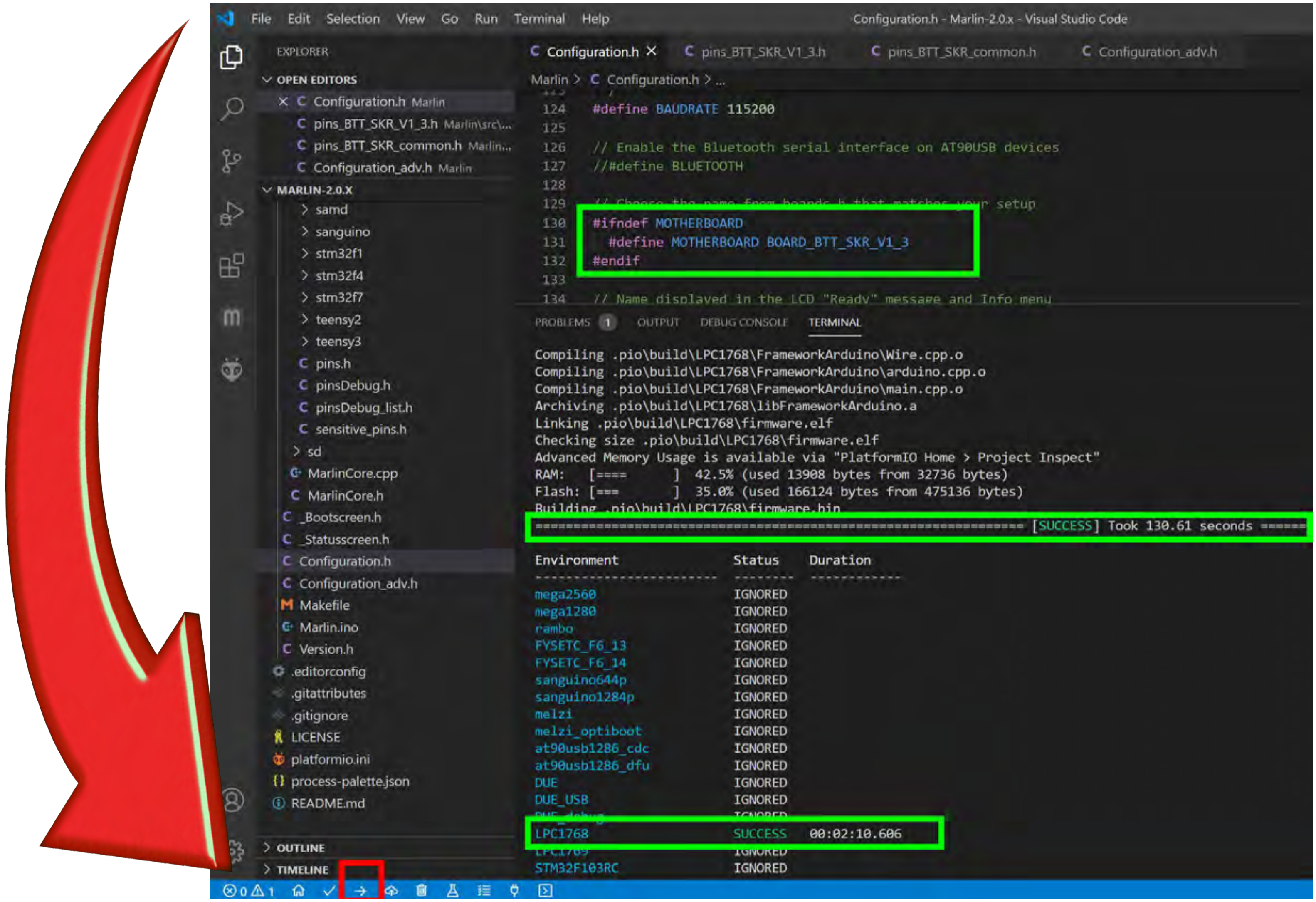
- The end of Marlin setup for BIQU A4988 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

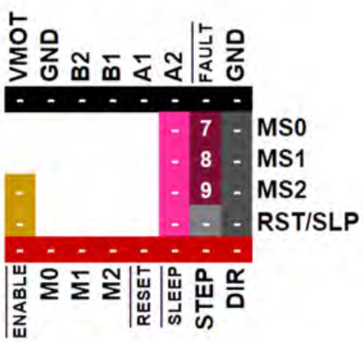


- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

DRV8825

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
32	0	1	1
32	1	1	1

fans/heaters/usb side

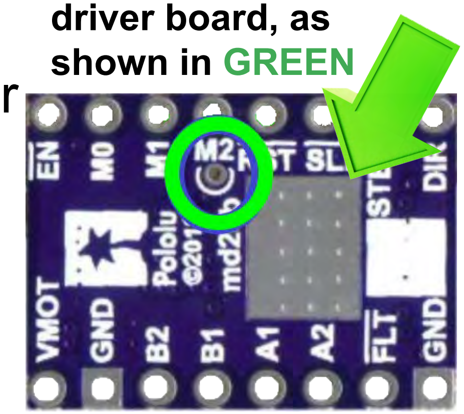
STATUS: Not Yet Tested

See Appendix D for legend

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

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

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



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

Driving Current Calculation Formula

R_S(Typical Sense Resistor)$=0.1\Omega$

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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

DRV8825

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
32	0	1	1
32	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Low → No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High → Jumper Set

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC xxA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

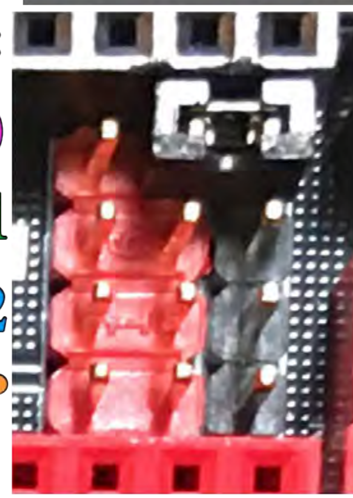
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

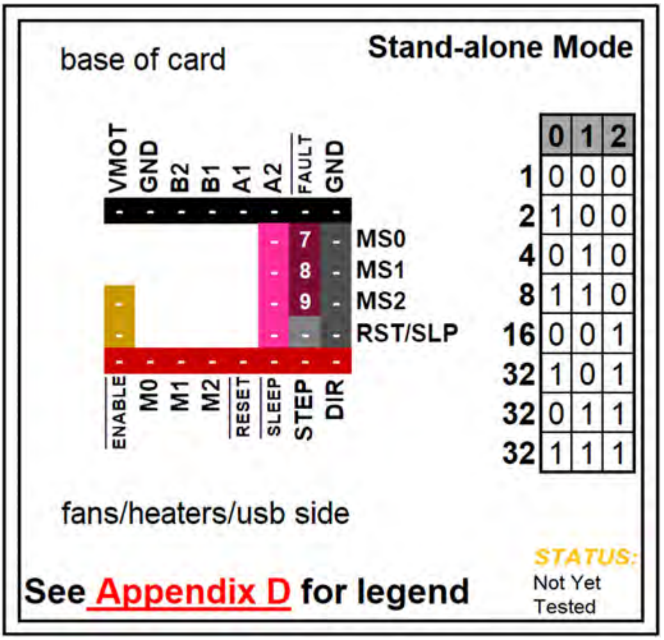
MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

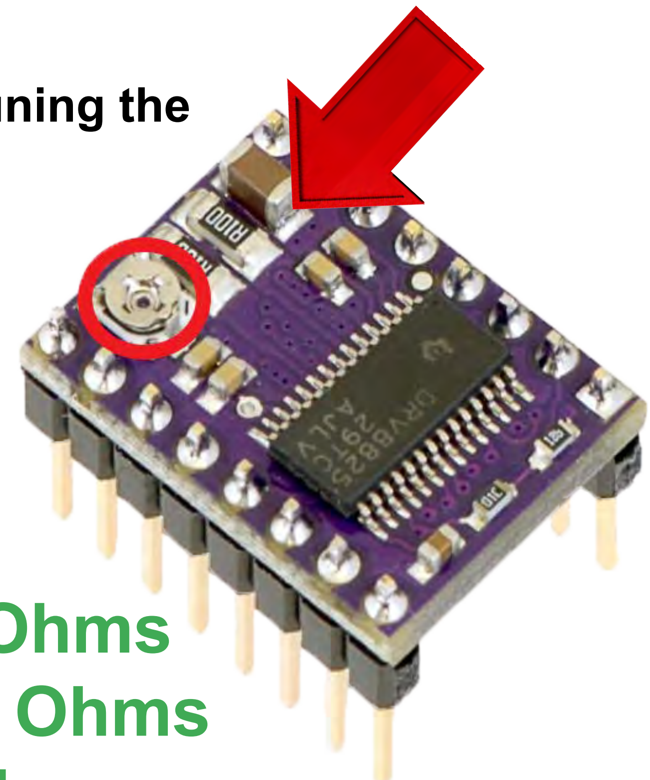
DRV8825



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
32	0	1	1
32	1	1	1

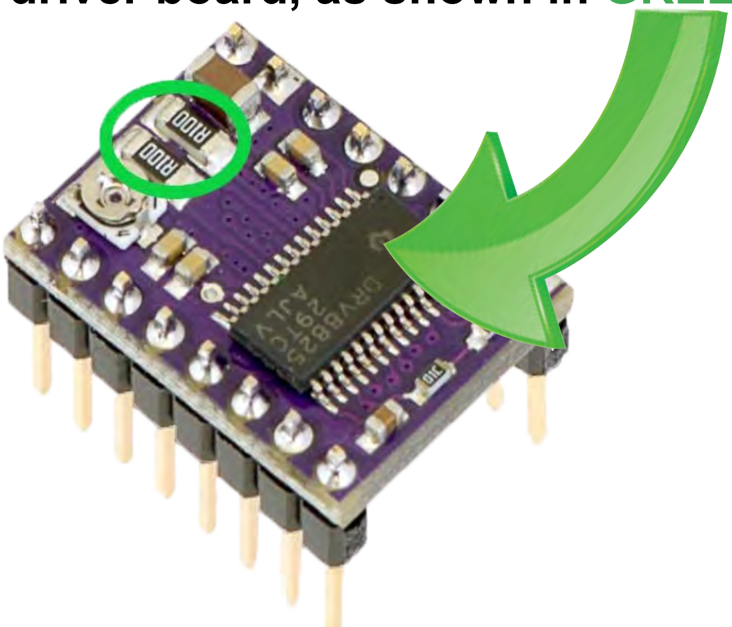
NOTE: Use the potentiometer (POT) on the top of the board (or the board's "V_{ref} Test point") to adjust your V_{ref}. See **Appendix A** for instructions on how to set the V_{ref} on a driver board.

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



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

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



R_s = R050 is 0.05 Ohms

R_s = R068 is 0.068 Ohms

R_s = R100 is 0.1 Ohms

R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

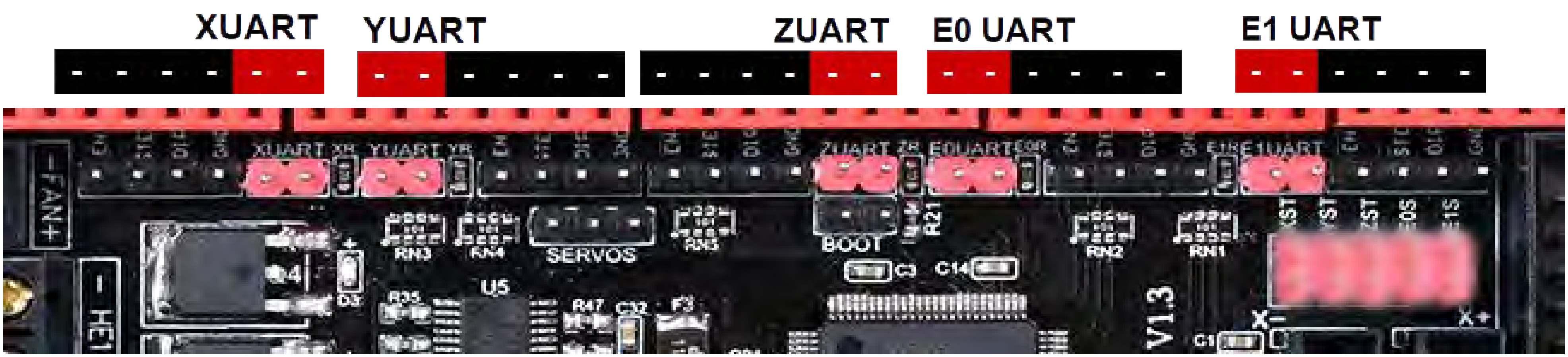
R_s = R220 is 0.22 Ohms

Stand-alone Mode

DRV8825

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

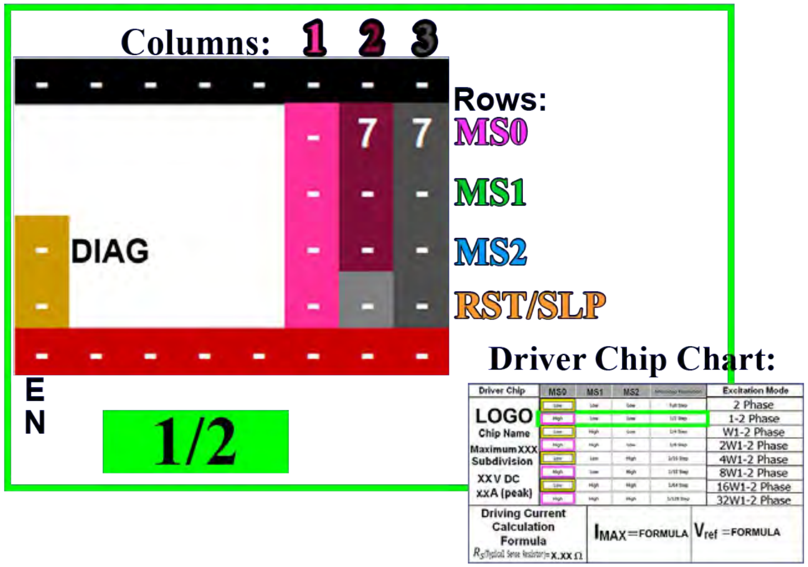


Stand-alone Mode

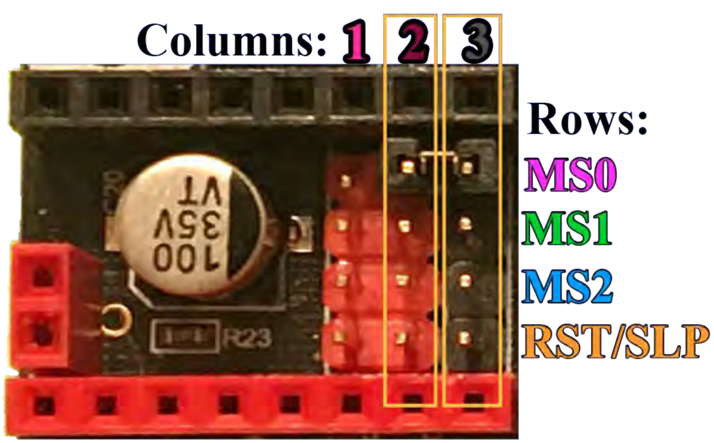
DRV8825

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

Stand-alone Mode

DRV8825

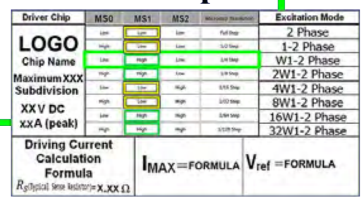
Driver Socket Representation

Columns: 1 2 3

Rows:
MS0
MS1
MS2
RST/SLP

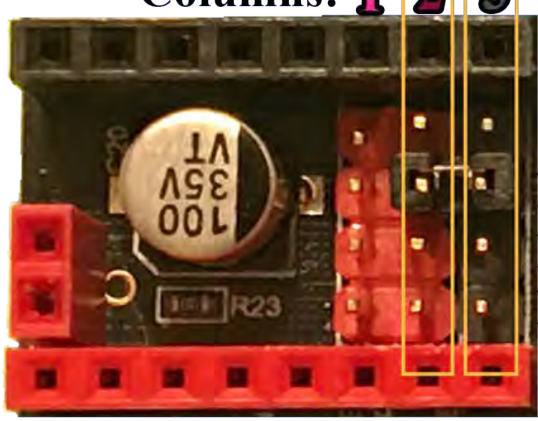
1/4

Driver Chip Chart:



High:

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

MS1 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 8 8 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

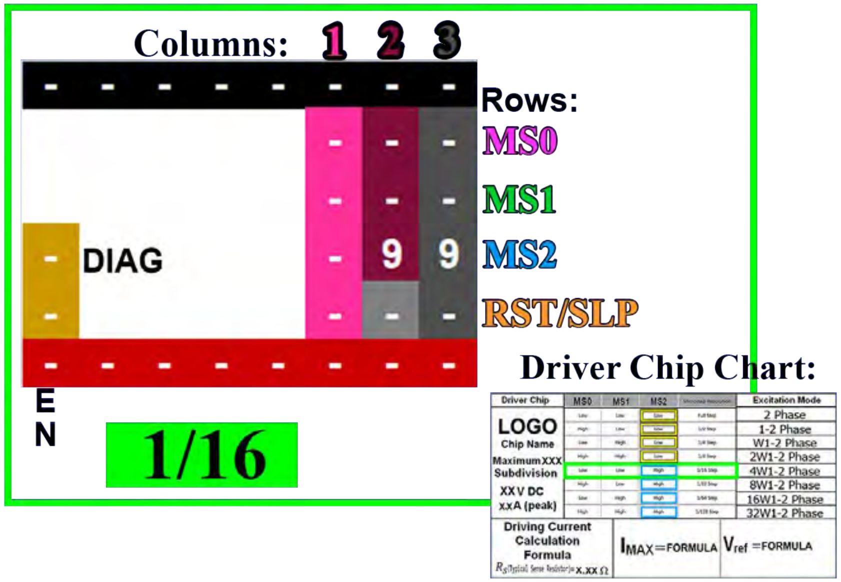
High → set <u>Jumper</u> between column 2 and column 3 on the MS1 row

Low → No <u>Jumper</u> Set

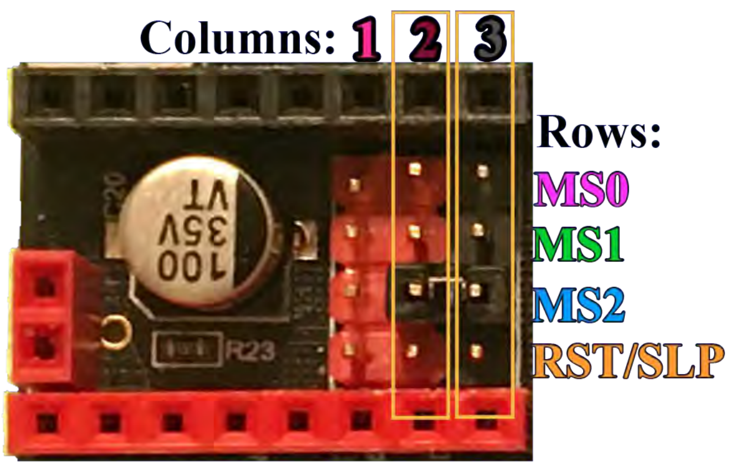
Stand-alone Mode

DRV8825

Driver Socket Representation



High:



MS2 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	⇒	Low
9 9	⇒	High

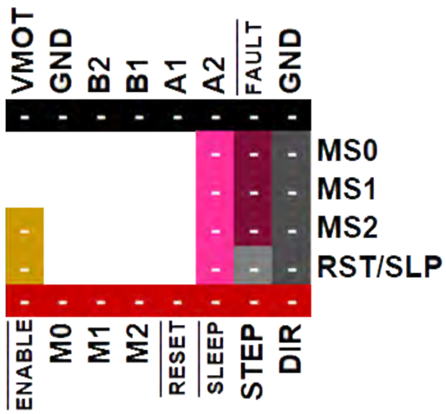
Meaning:

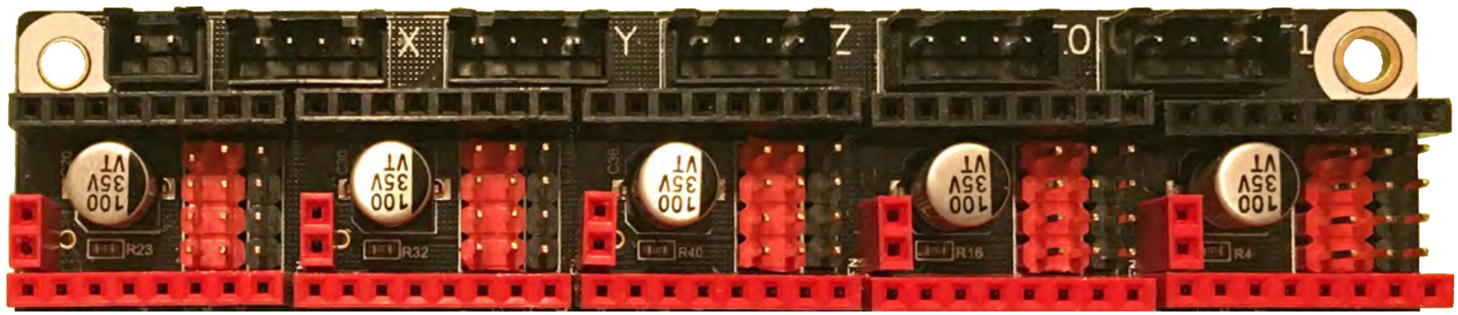
Driver Chip Chart:	Meaning:
High ⇒	set Jumper between column 2 and column 3 on the MS2 row
Low ⇒	No Jumper Set

Stand-alone Mode

DRV8825

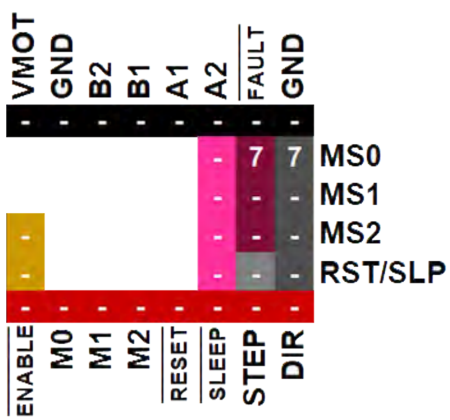
STEP





See Appendix D for legend

1 / 2

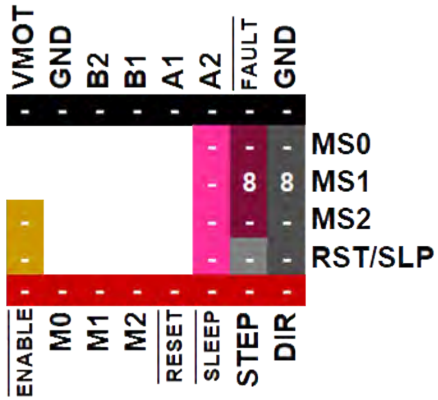


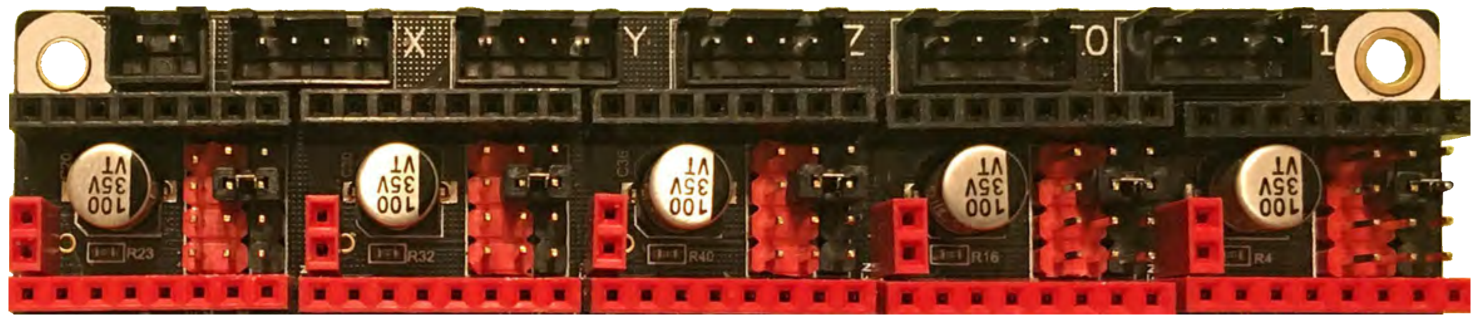
See Appendix D for legend

Stand-alone Mode

DRV8825

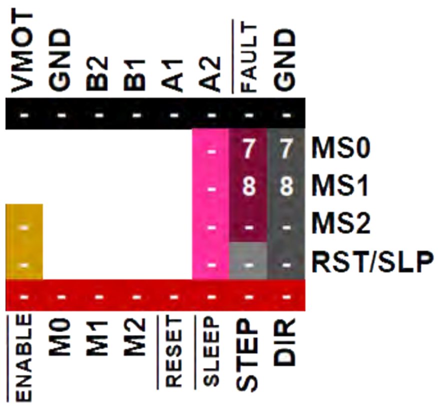
1 / 4

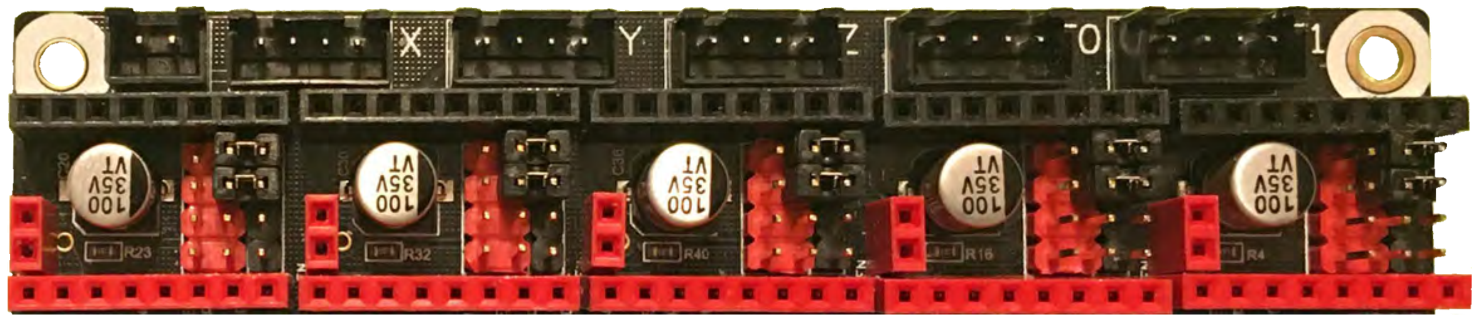




See Appendix D for legend

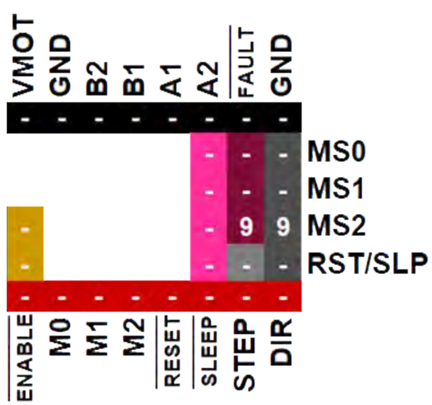
1 / 8

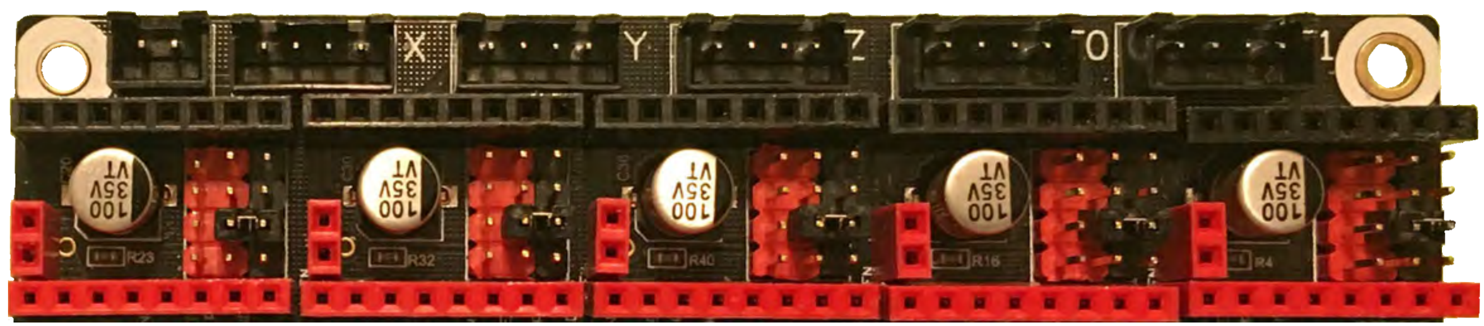




See Appendix D for legend

1 / 16





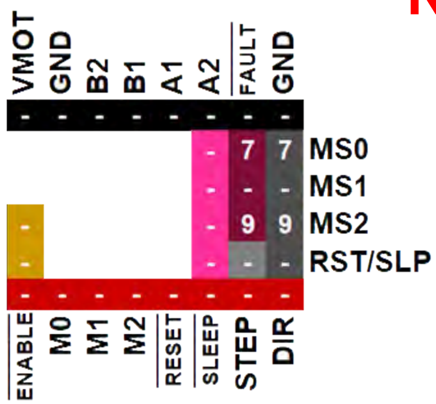
See Appendix D for legend

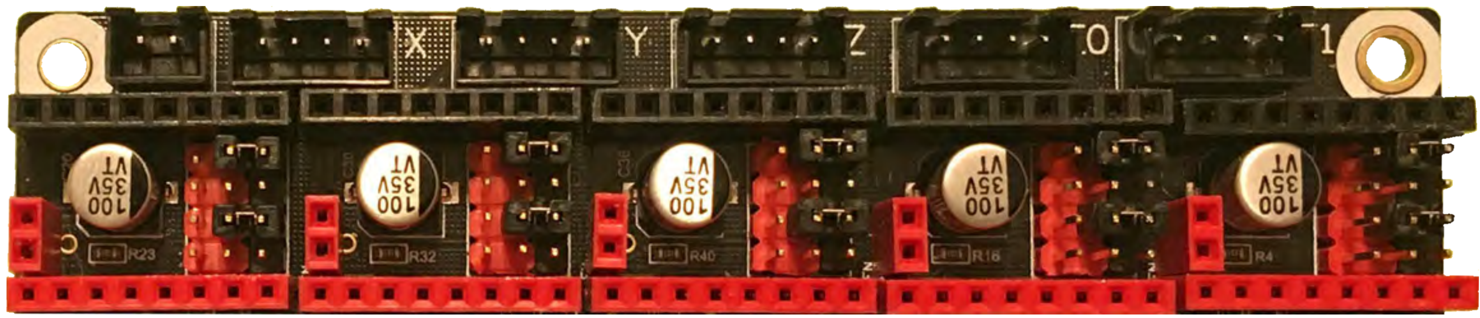
Stand-alone Mode

DRV8825

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

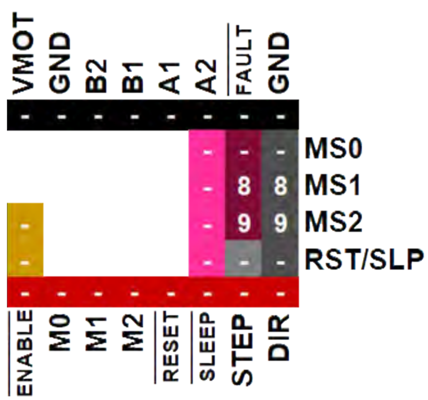
1 / 32

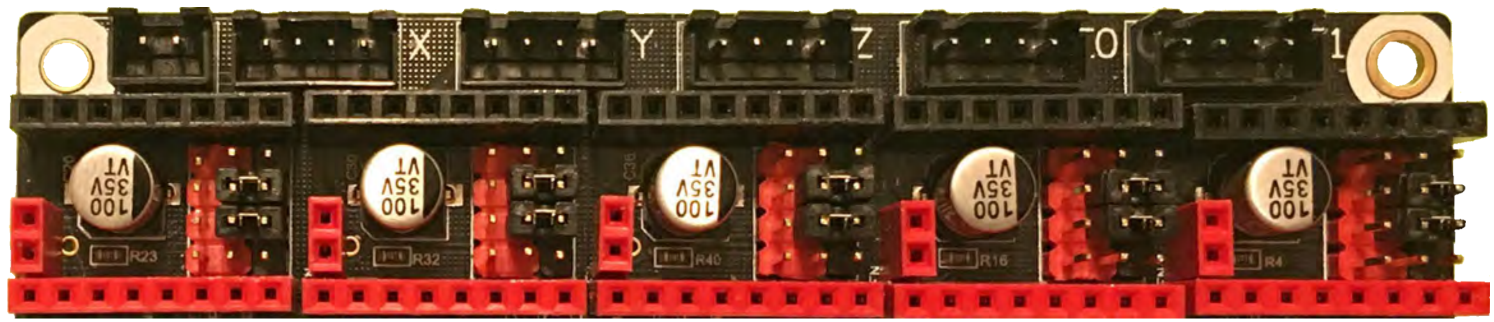




See Appendix D for legend

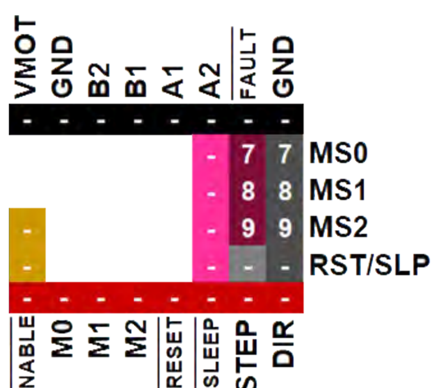
1 / 32

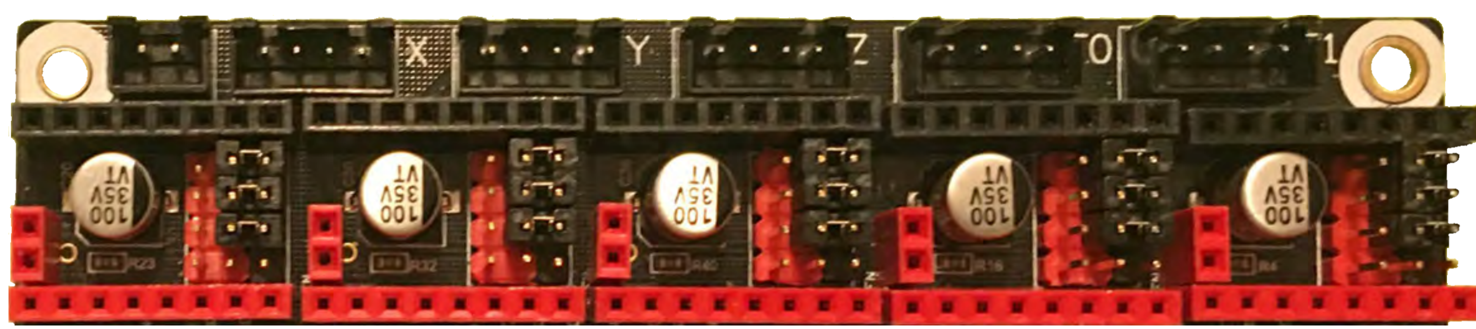




See Appendix D for legend

1 / 32



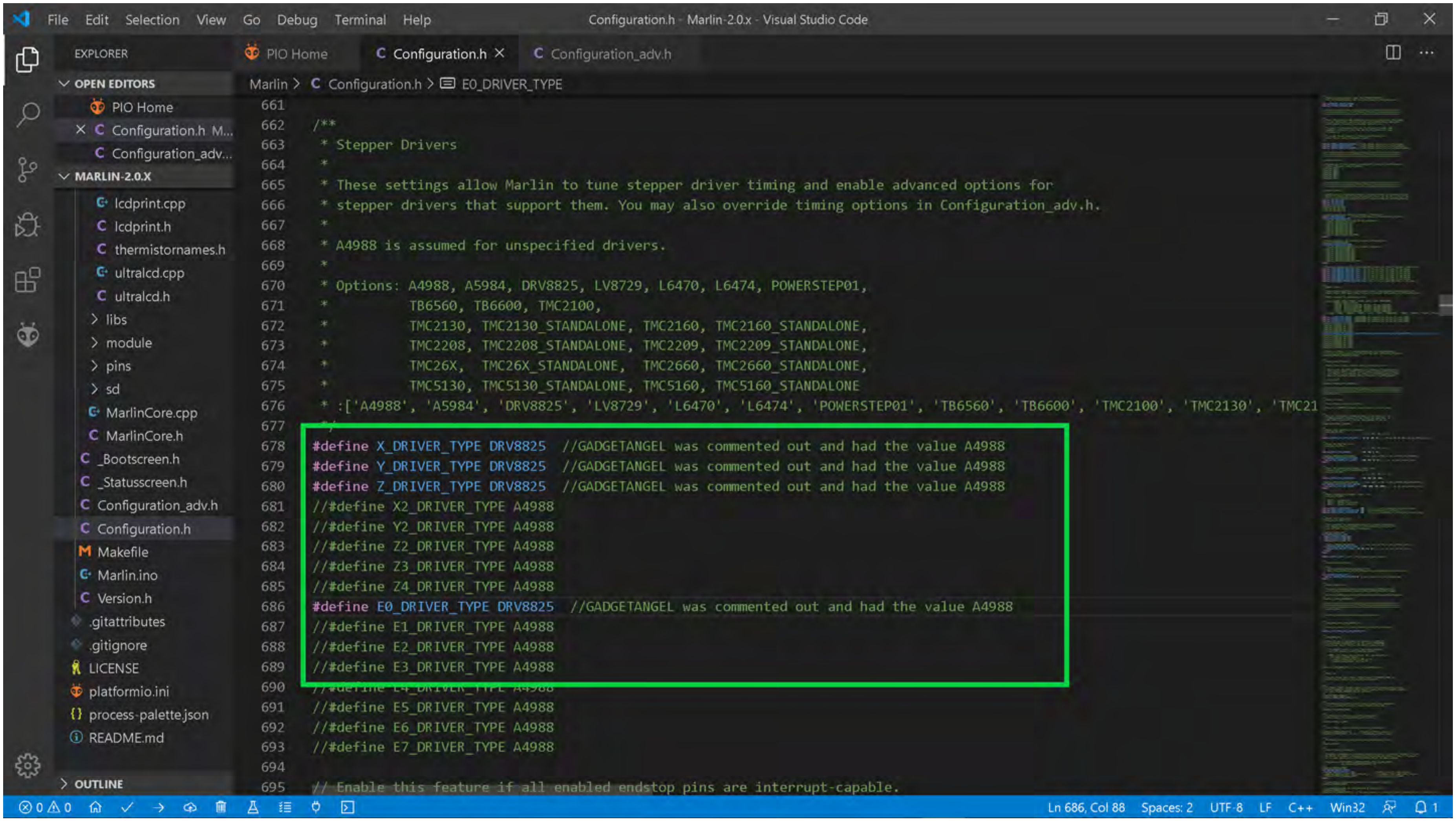


See Appendix D for legend

The (latest release of) Marlin Setup for DRV8825 Drivers

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

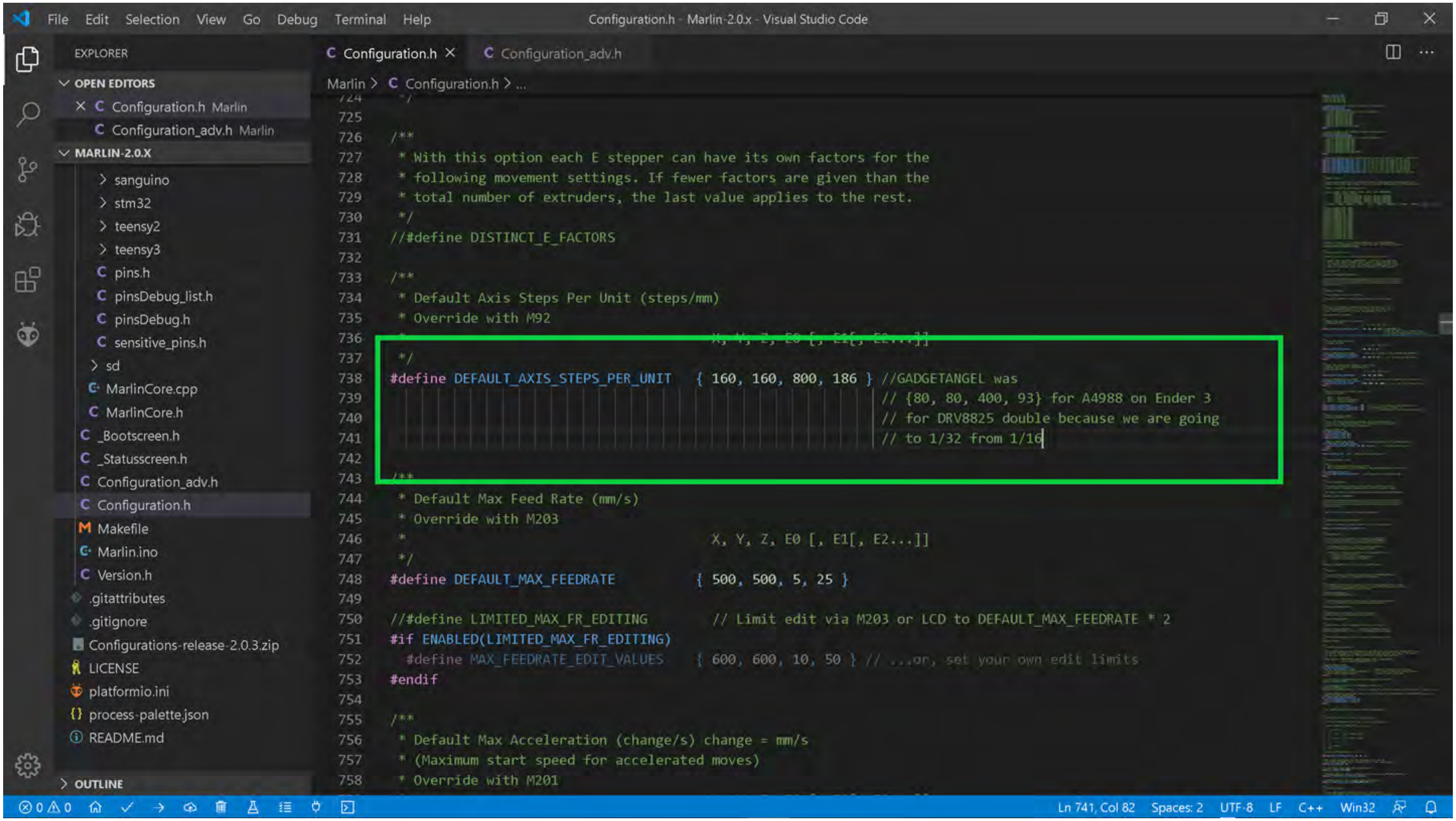
- Change the stepper motor drivers so that Marlin knows you are using DRV8825 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use DRV8825 drivers. When two "//" appear at the beginning of a line that means that line is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



- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

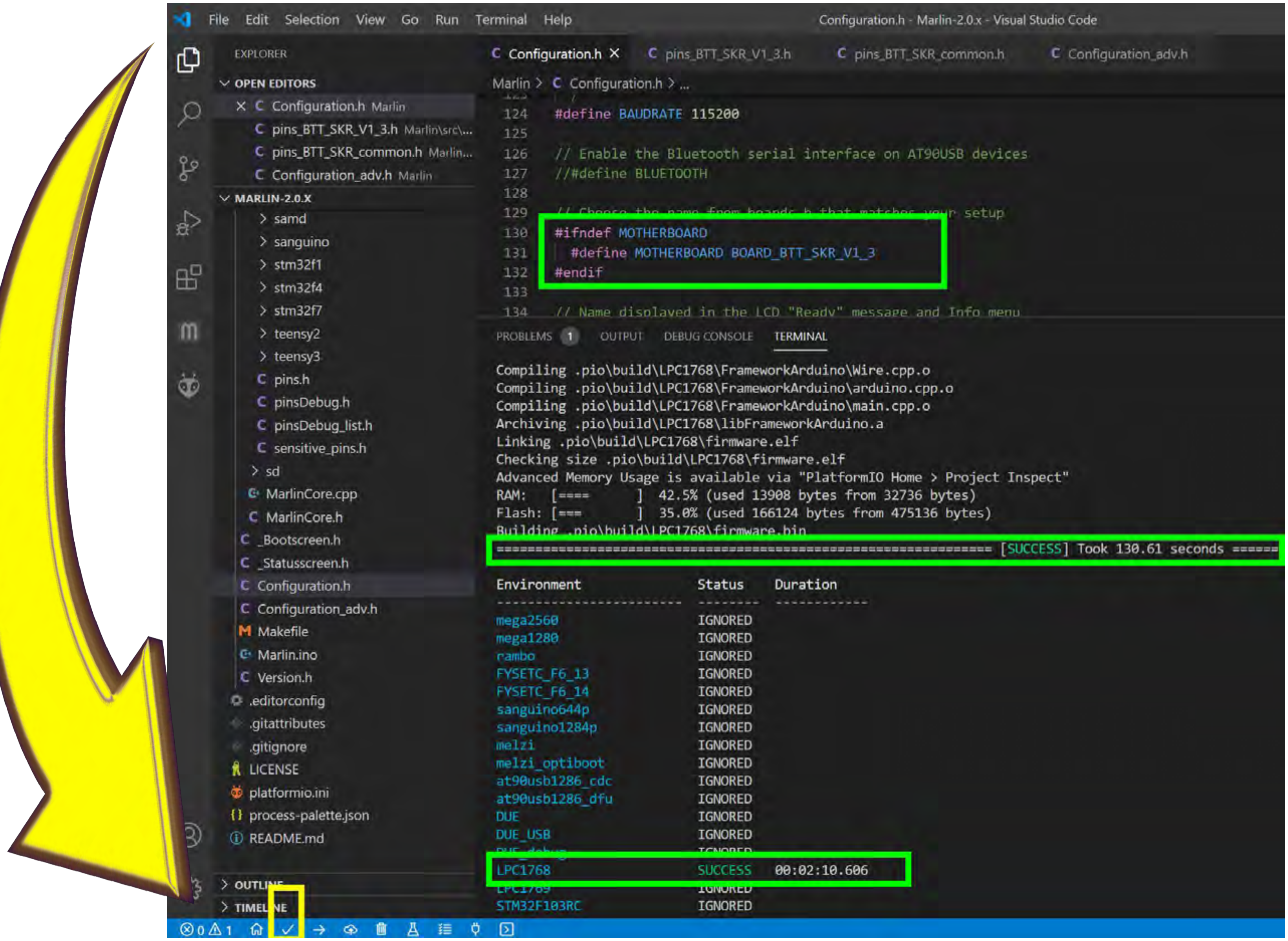
- We are changing from A4988 stepper motor drivers on the Ender 3 to DRV8825 stepper motor drivers for each axis and the extruder stepper motor driver. We will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime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



- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

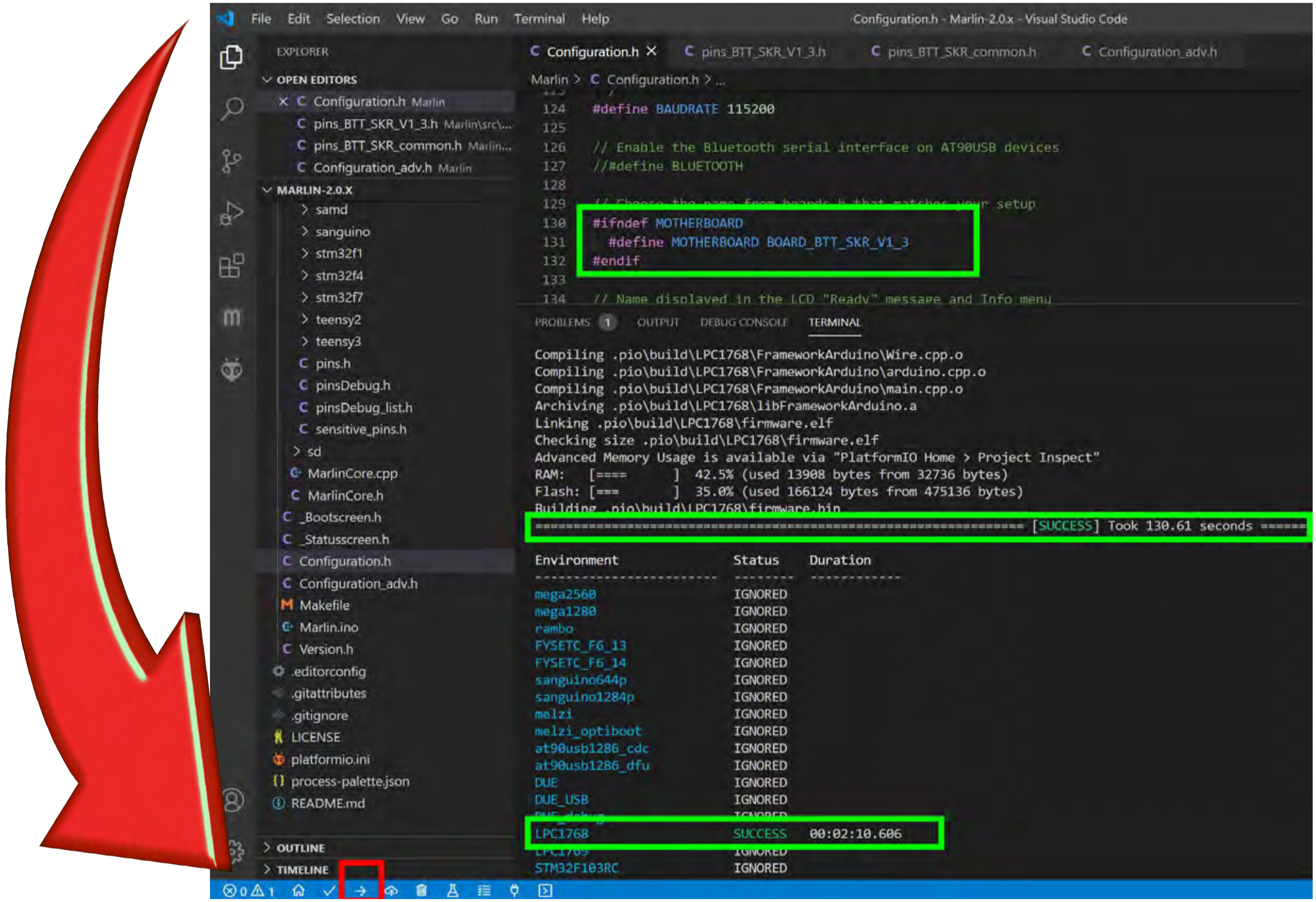
- The end of Marlin setup for DRV8825 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

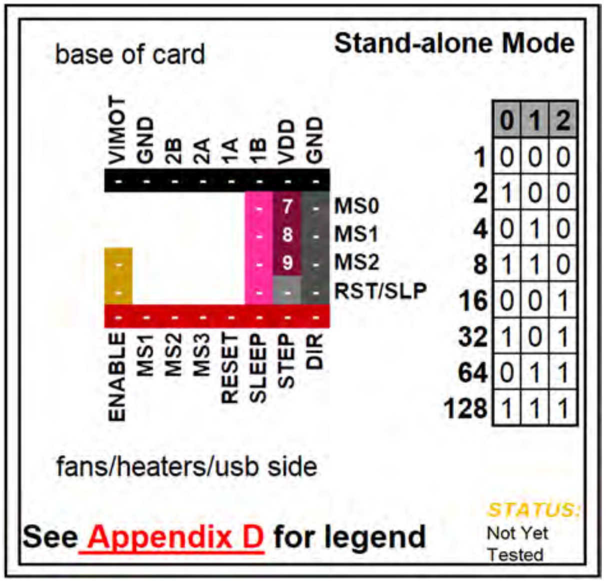
The (latest release of) Marlin Setup for DRV8825 Drivers

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU LV8729



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

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

Driving Current Calculation Formula

R_S(Typical Sense Resistor)$=0.22\Omega$

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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

BIQU LV8729

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇒ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
XX V DC	High	Low	High	1/32 Step	8W1-2 Phase
xxA (peak)	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

High ⇒ Jumper Set

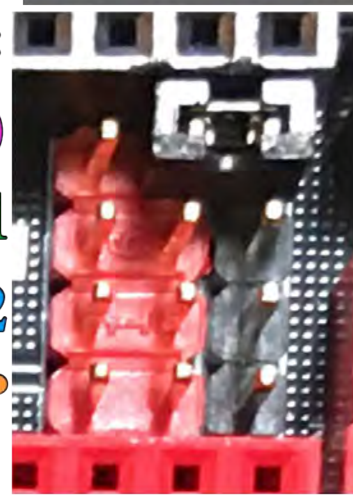
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU LV8729

base of card

Stand-alone Mode

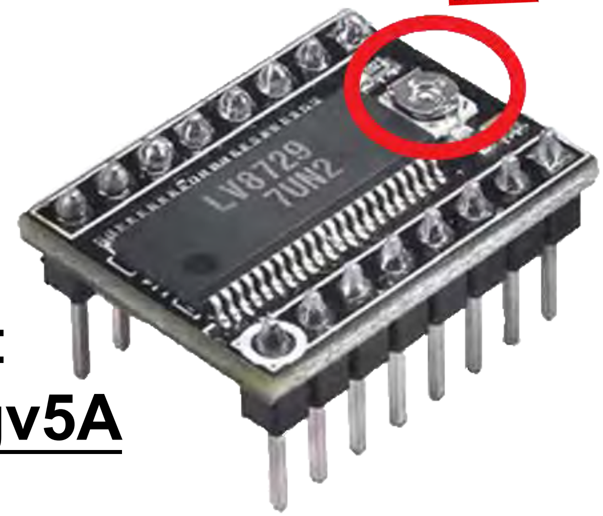
	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

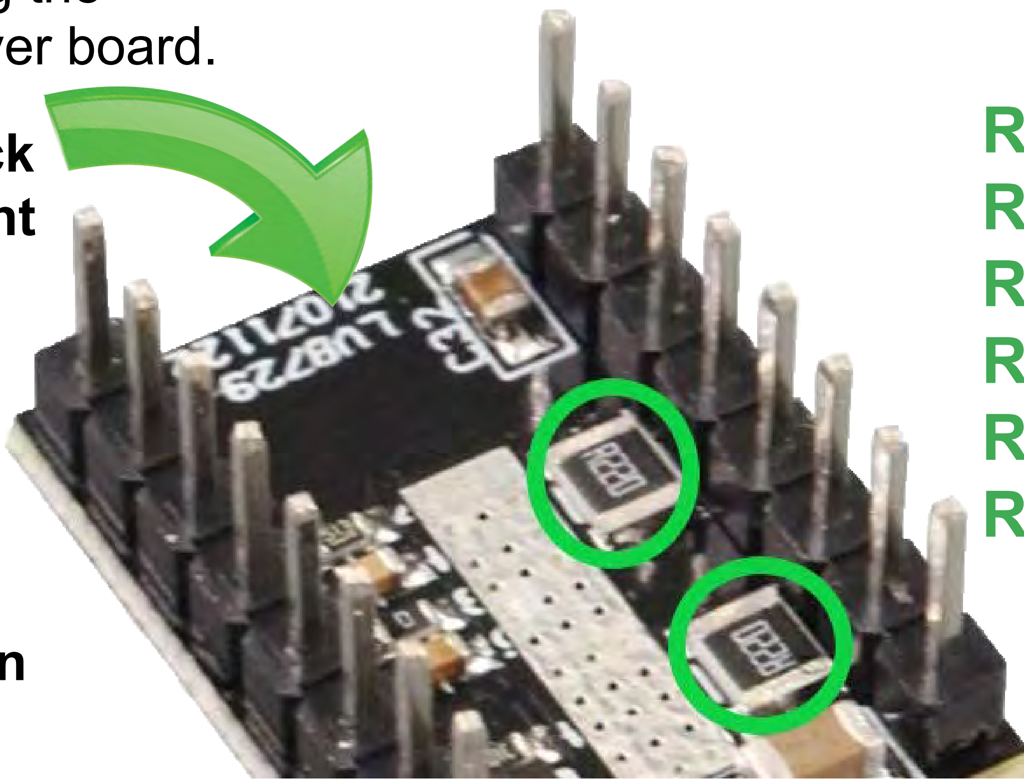
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref}. See **Appendix A** for instructions on how to set the V_{ref} on a driver board



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

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

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



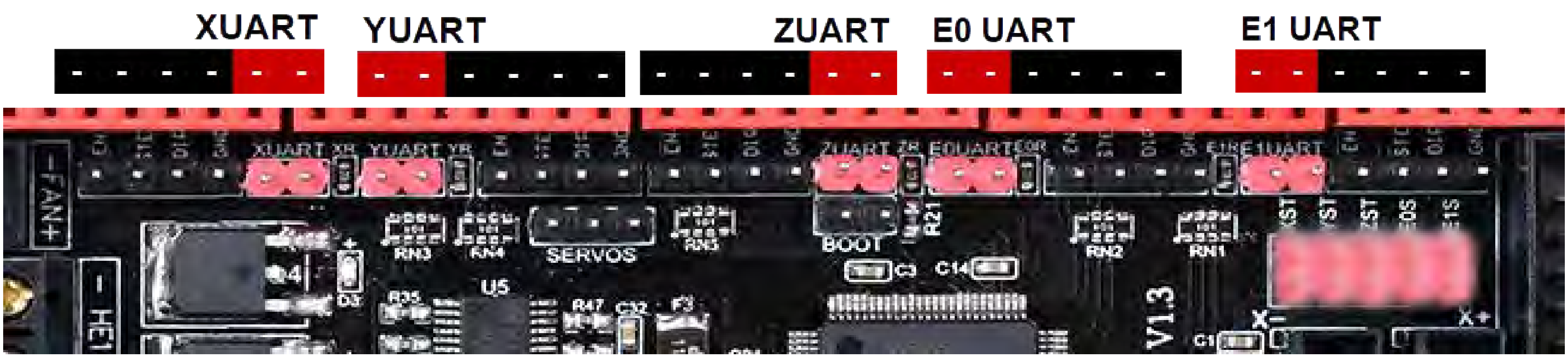
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

BIQU LV8729

Note: **Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.**

Note: **Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.**

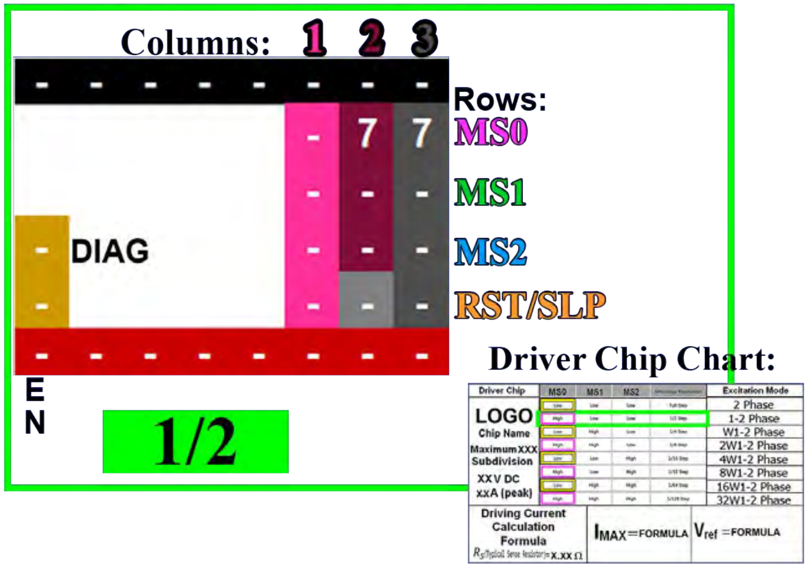


Stand-alone Mode

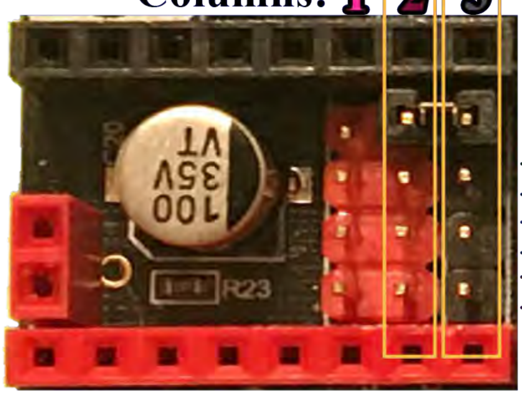
BIQU LV8729

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

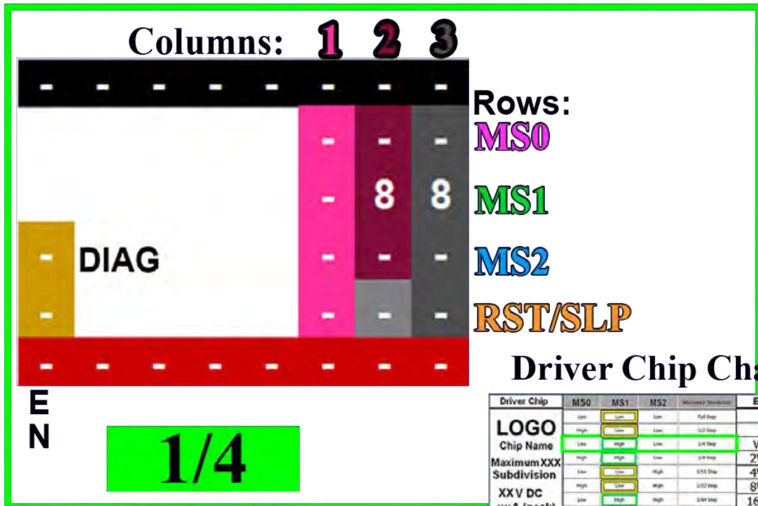
Driver Chip Chart:

Driver Chip Chart	Meaning
High	set Jumper between column 2 and column 3 on the MS0 row
Low	No Jumper Set

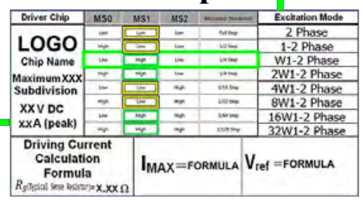
Stand-alone Mode

BIQU LV8729

Driver Socket Representation

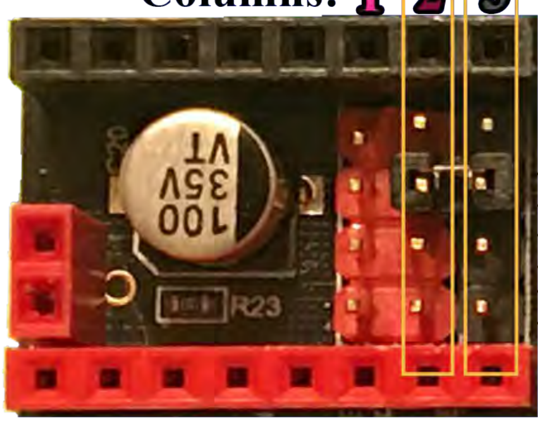


Driver Chip Chart:



High:

Columns: 1 2 3



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

Meaning:

Driver Chip Chart:

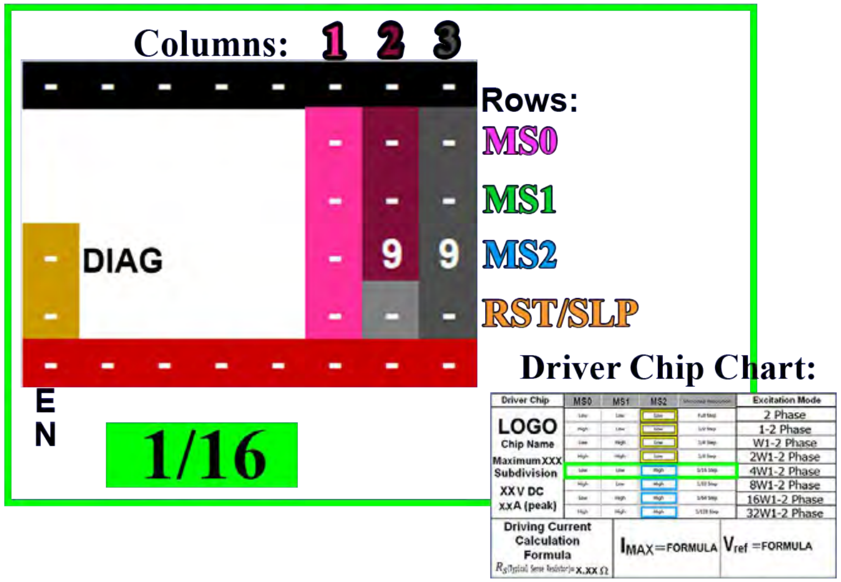
High → set Jumper between column 2 and column 3 on the MS1 row

Low → No Jumper Set

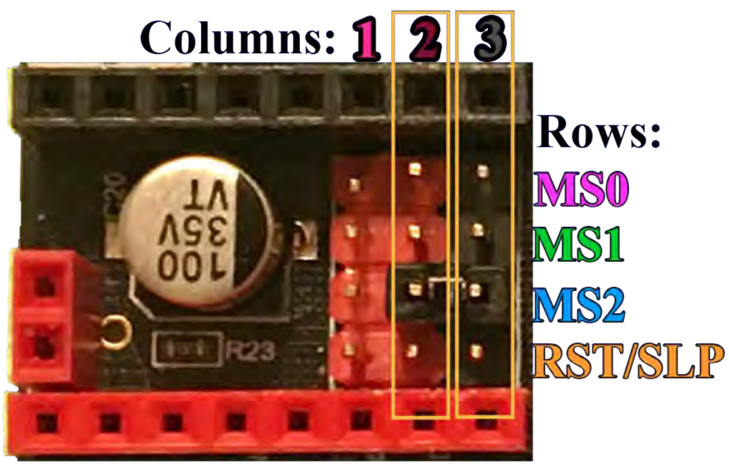
Stand-alone Mode

BIQU LV8729

Driver Socket Representation



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

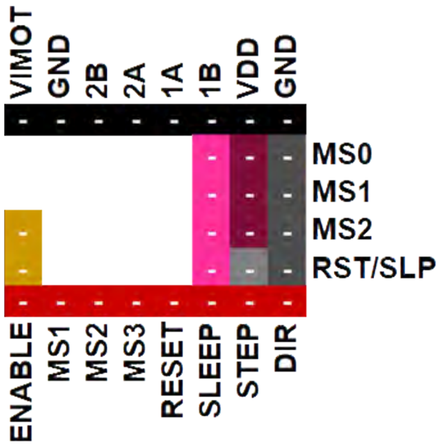
High → **set Jumper between column 2 and column 3 on the MS2 row**

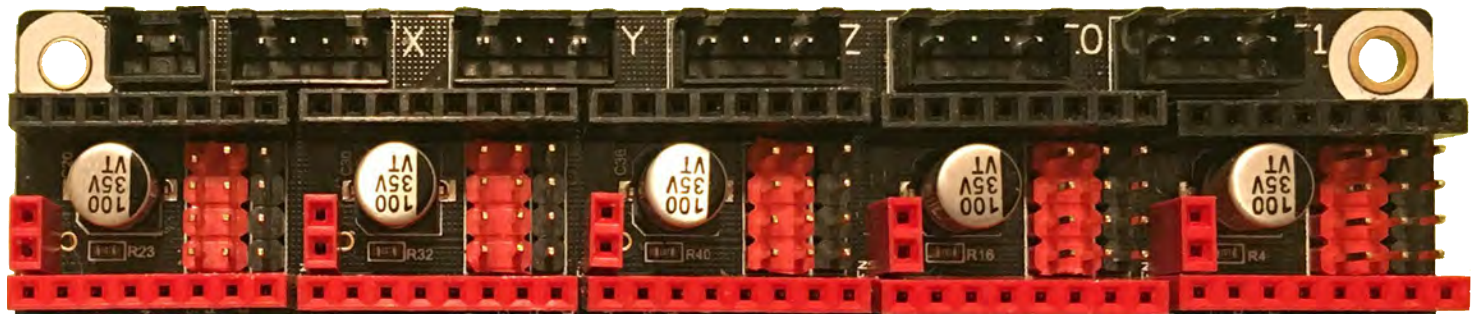
Low → **No Jumper Set**

Stand-alone Mode

BIQU LV8729

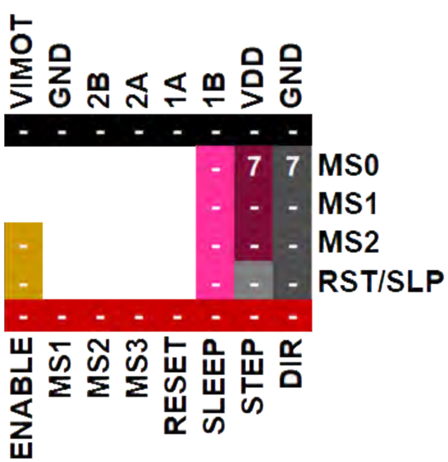
STEP

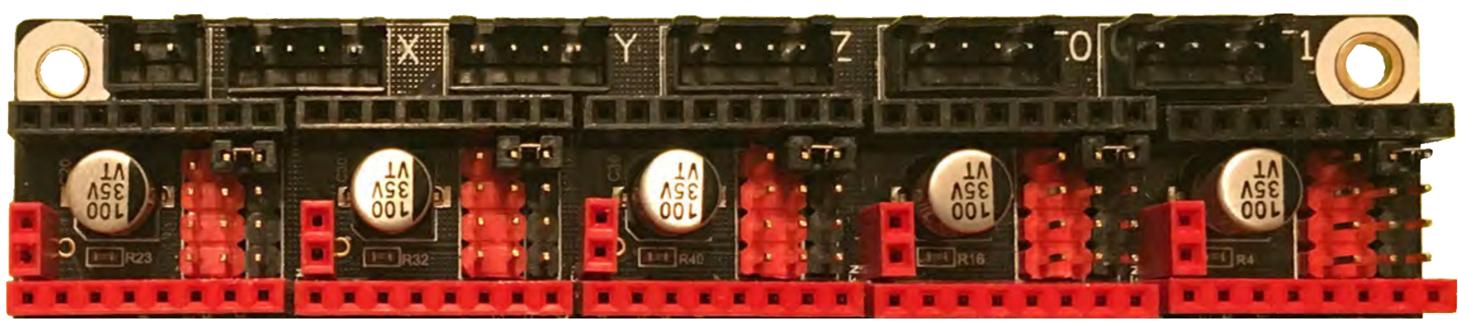




See Appendix D for legend

1 / 2



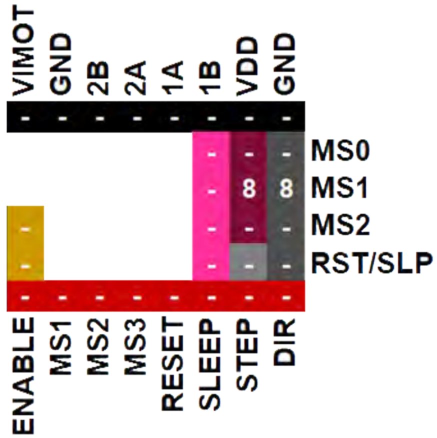


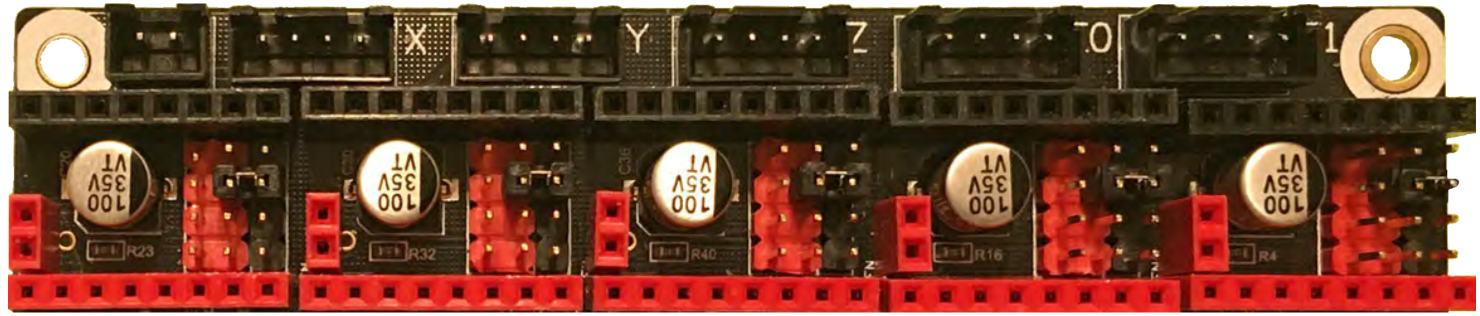
See Appendix D for legend

Stand-alone Mode

BIQU LV8729

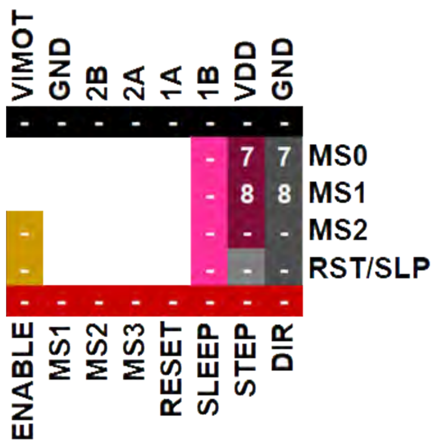
1 / 4

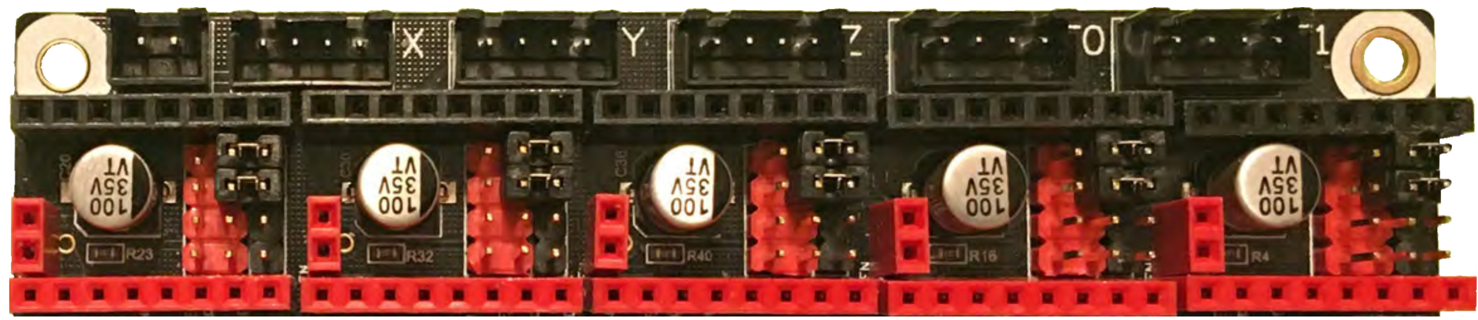




See Appendix D for legend

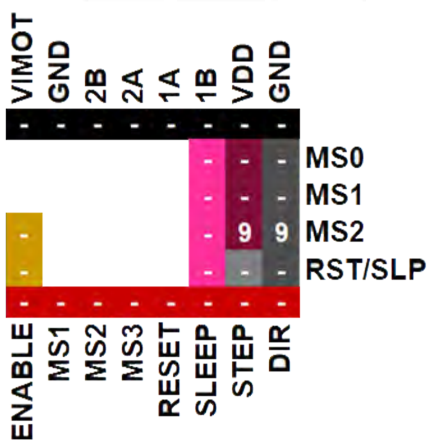
1 / 8

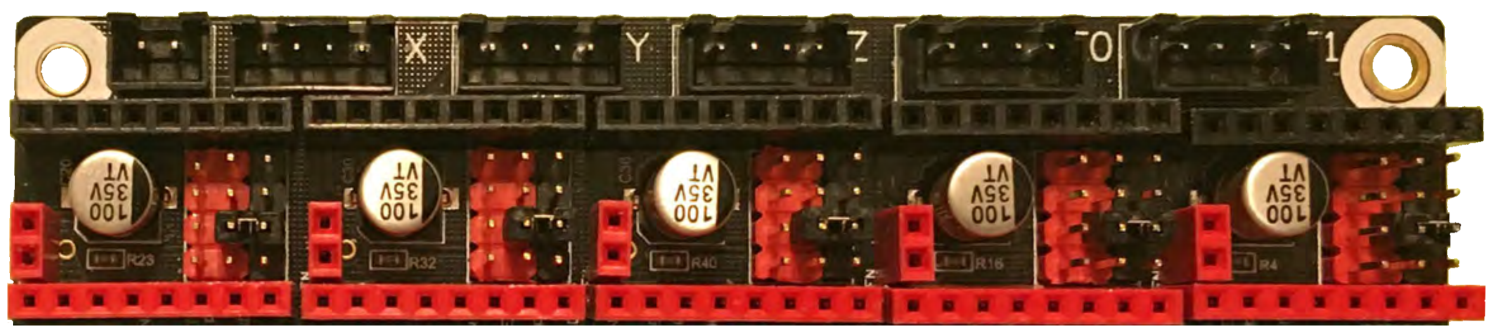




See Appendix D for legend

1 / 16



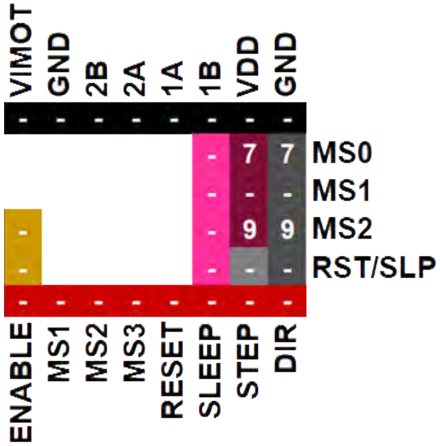


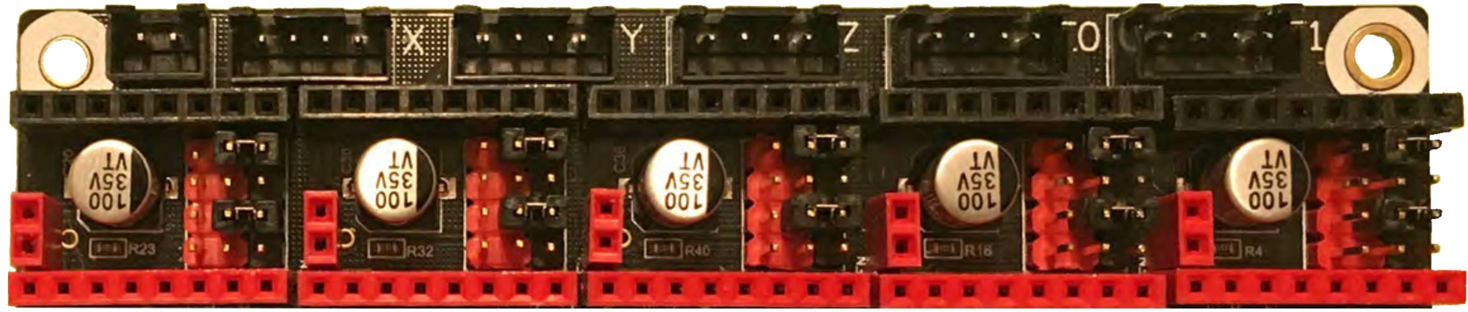
See Appendix D for legend

Stand-alone Mode

BIQU LV8729

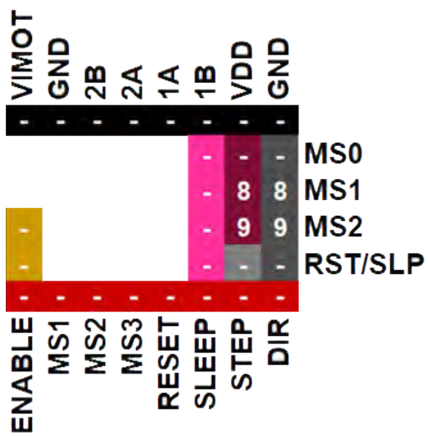
1 / 32

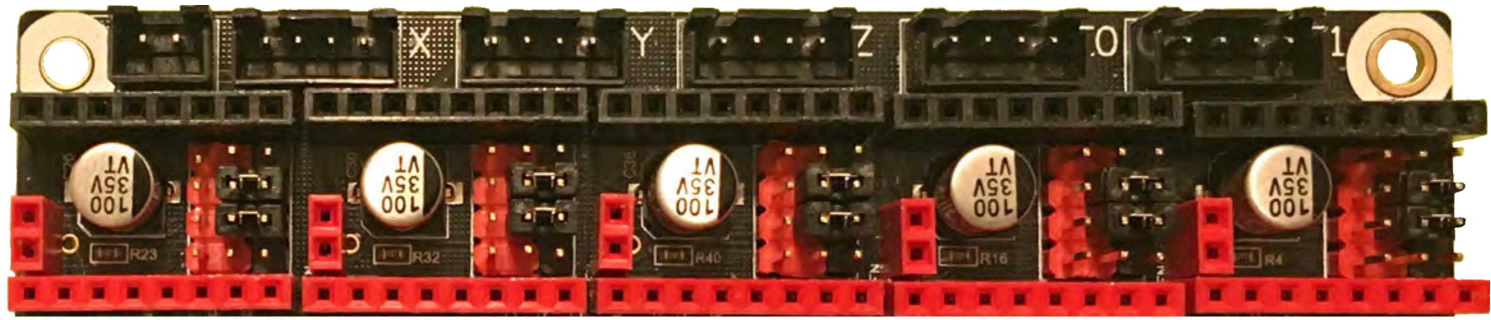




See Appendix D for legend

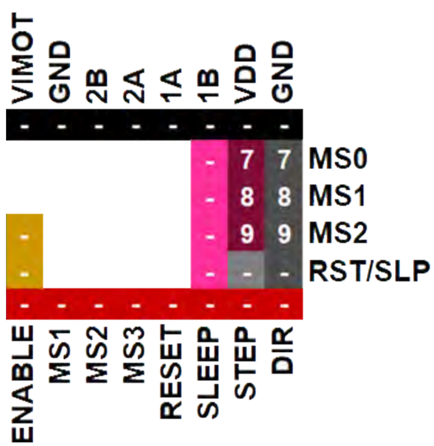
1 / 64

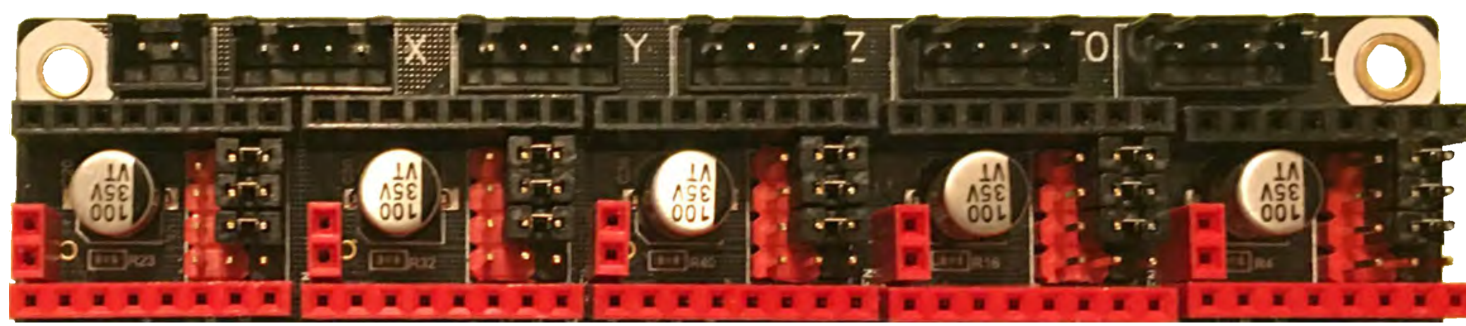




See Appendix D for legend

1 / 128



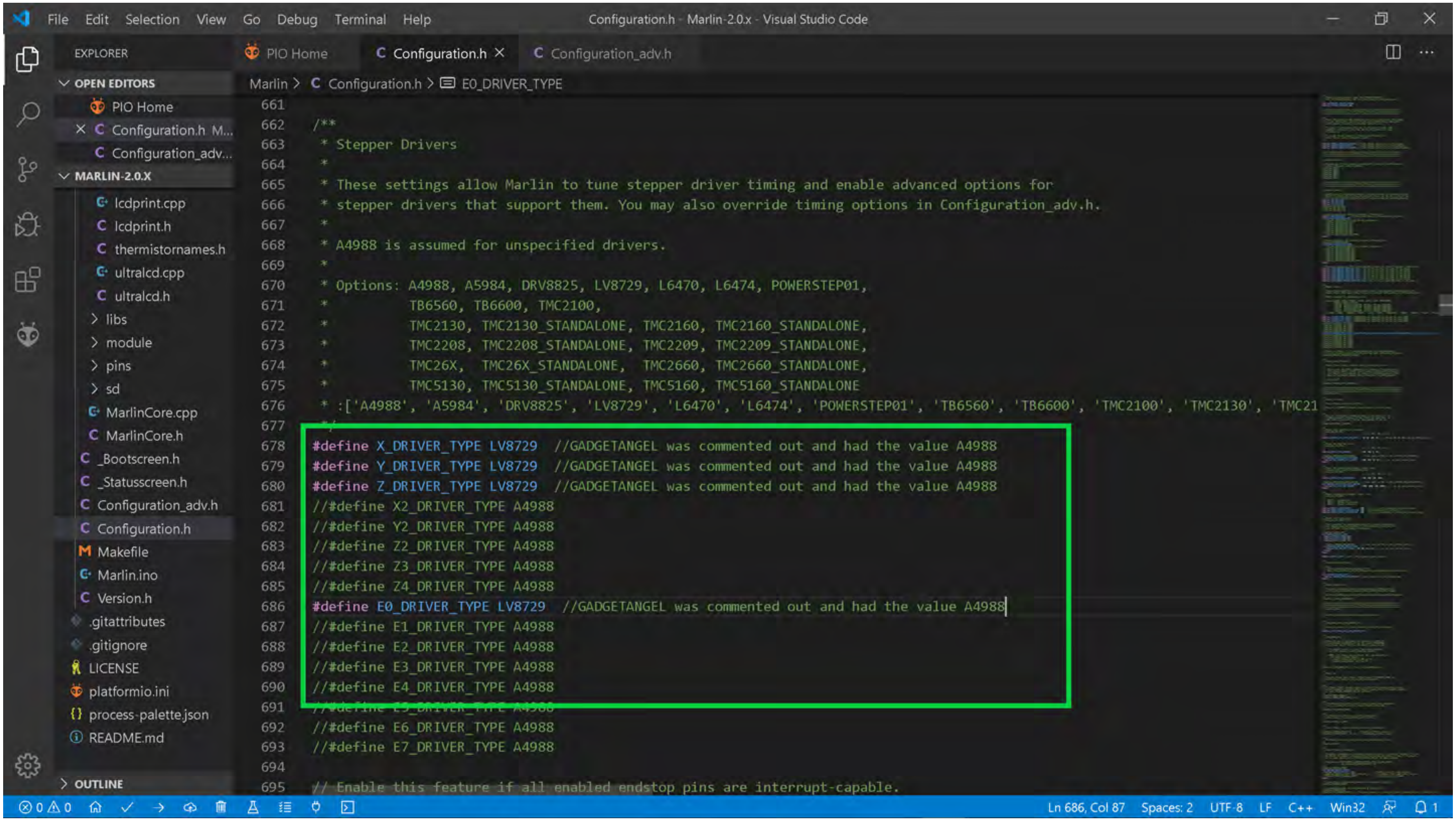


See Appendix D for legend

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

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

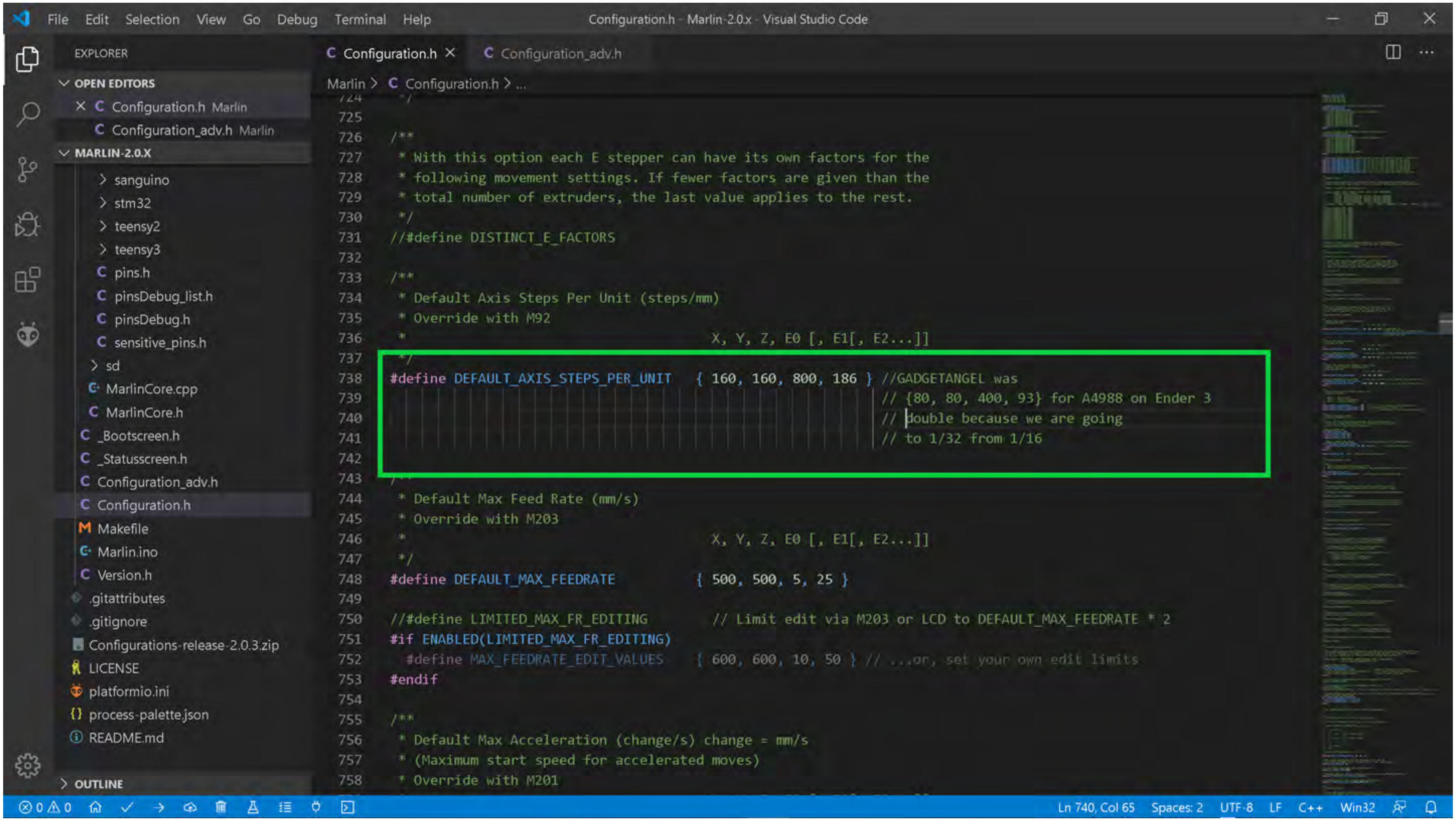
- Change the stepper motor drivers so that Marlin knows you are using BIQU LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU LV8729 drivers. When two "//" appear at the beginning of a line that means that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



- Go to the next page.

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

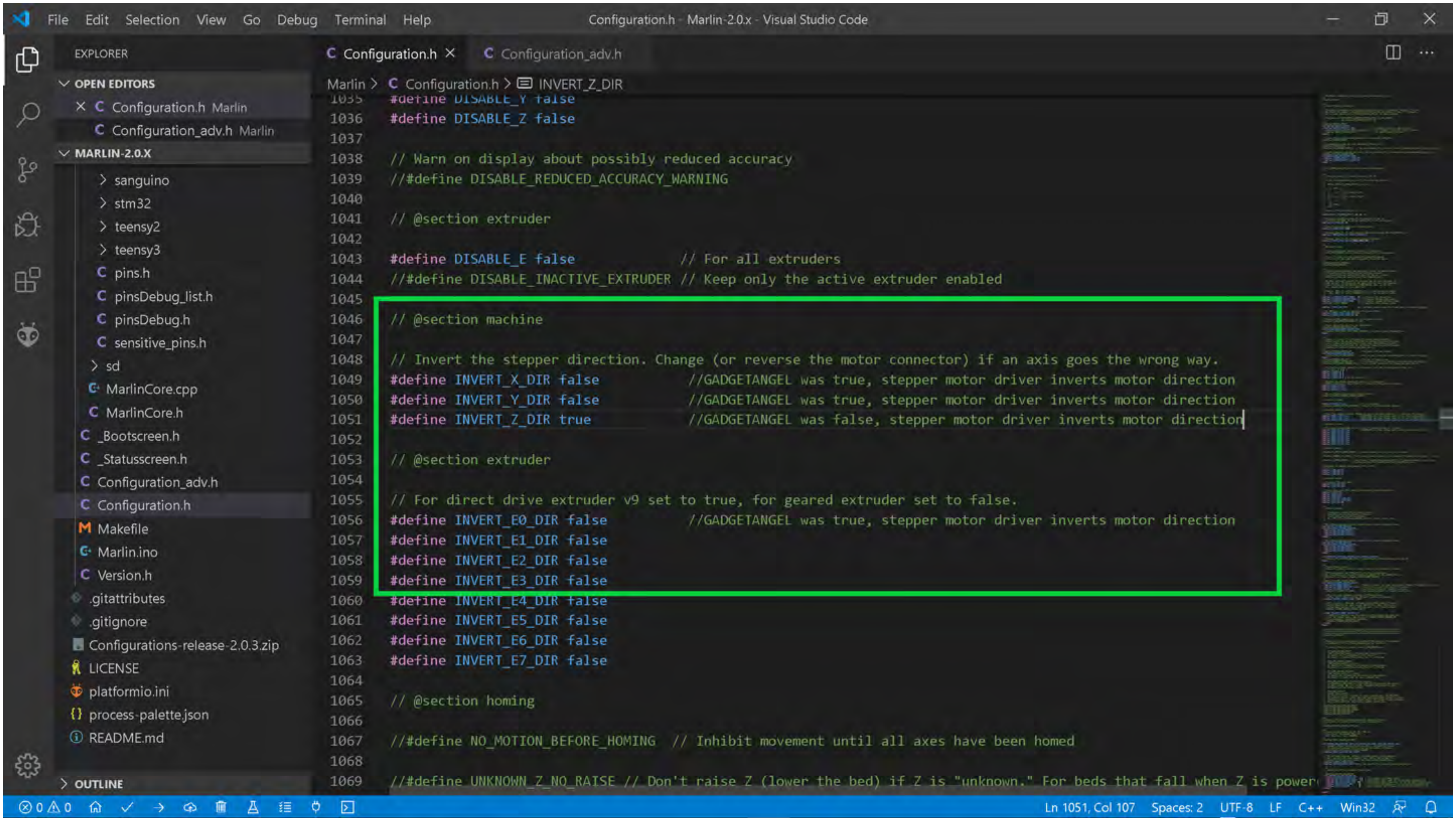
- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

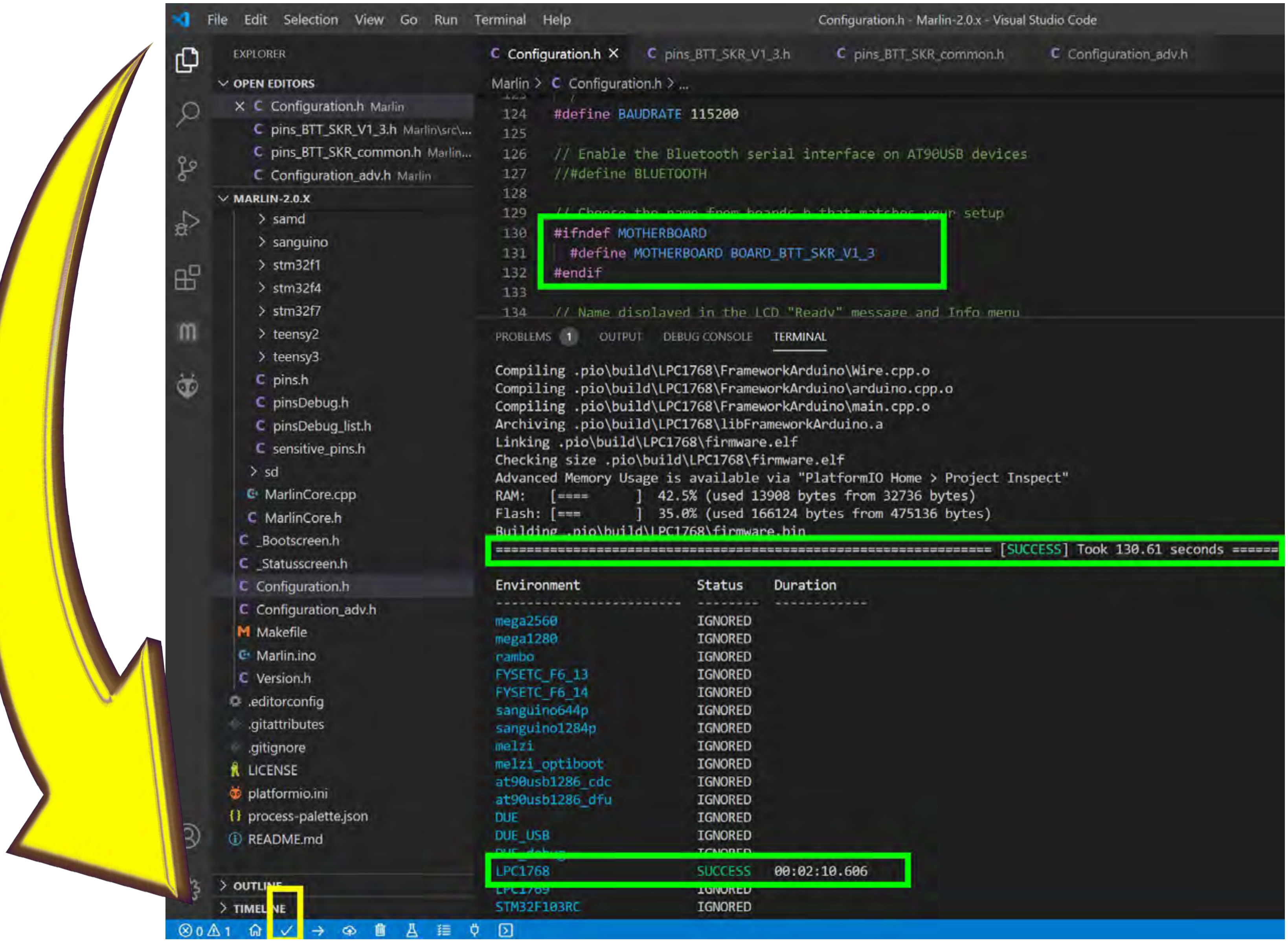
- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver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



- Go to the next page.

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

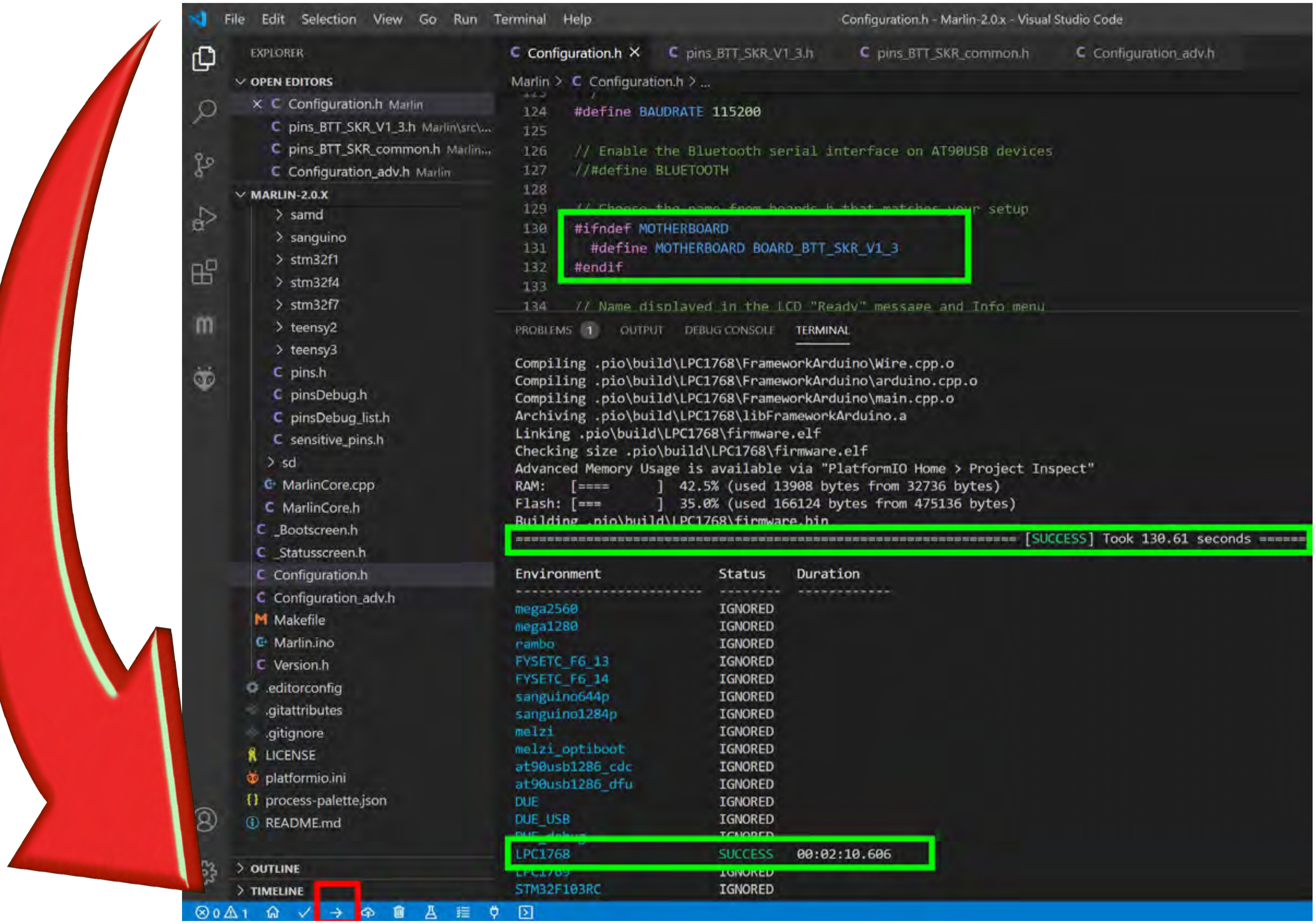
- The end of Marlin setup for BIQU LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

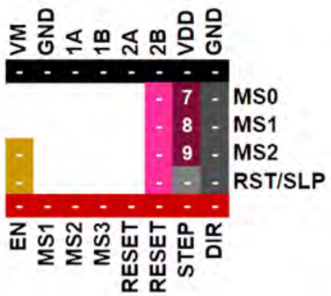
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

FYSETC LV8729

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

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

Driving Current Calculation Formula

R_S(Typical Sense Resistor)=0.22Ω

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

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

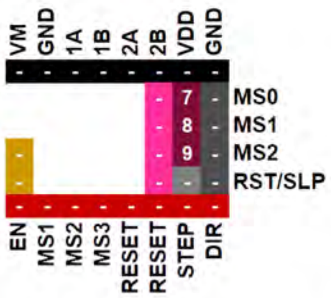
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

FYSETC LV8729

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

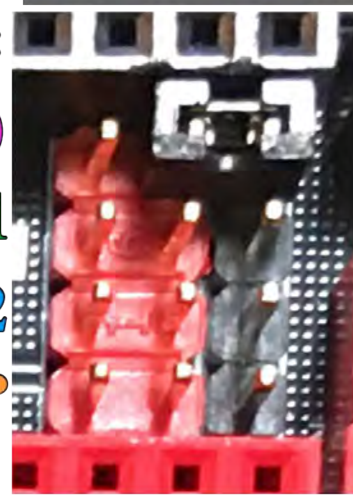
STATUS: Not Yet Tested

Low → No Jumper

Rows: MS0 MS1 MS2 RST/SLP

High → Jumper Set

Rows: MS0 MS1 MS2 RST/SLP



MS0 SET HIGH

MS1 SET HIGH

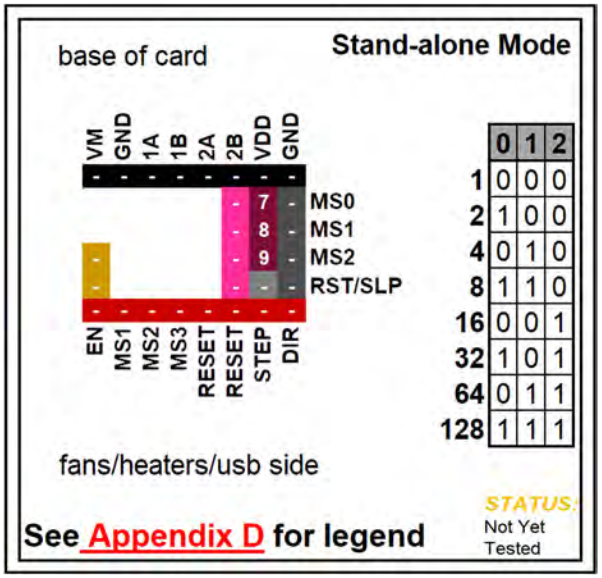
MS2 SET HIGH

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
XXA (peak)	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA		V_{ref} =FORMULA		

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

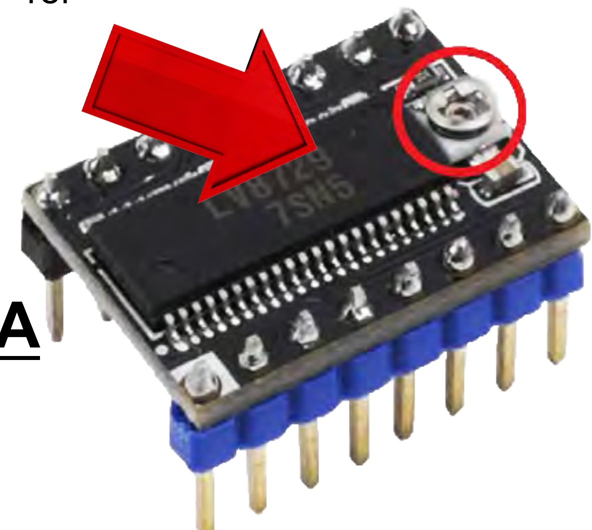


	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

FYSETC LV8729

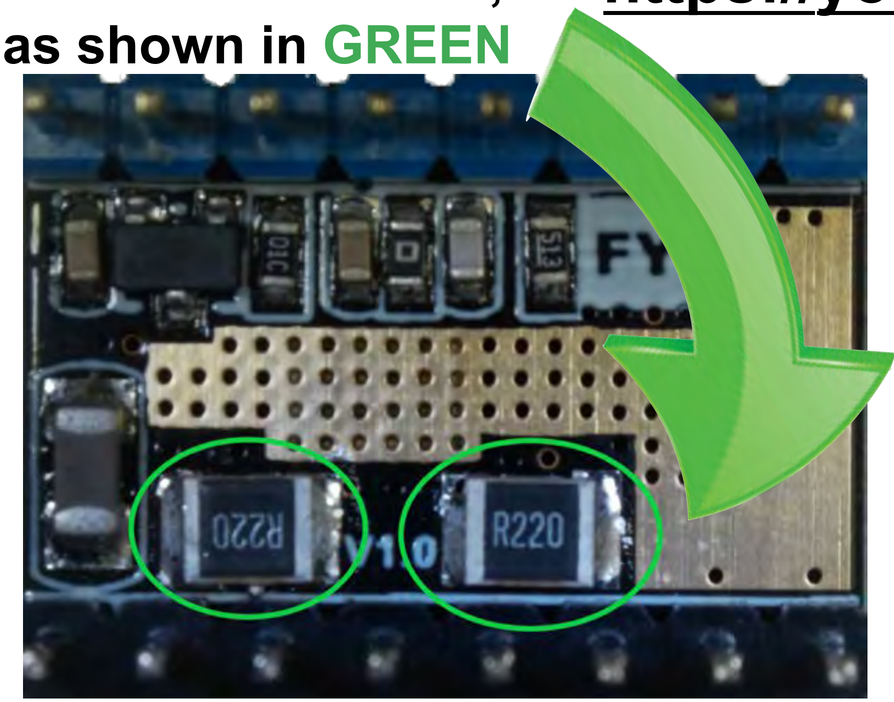
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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



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

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



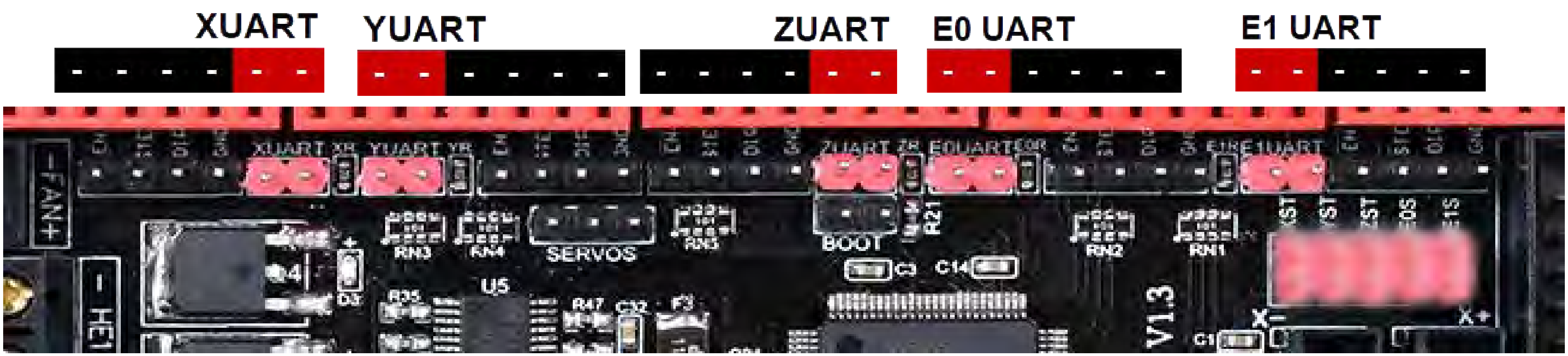
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

FYSETC LV8729

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

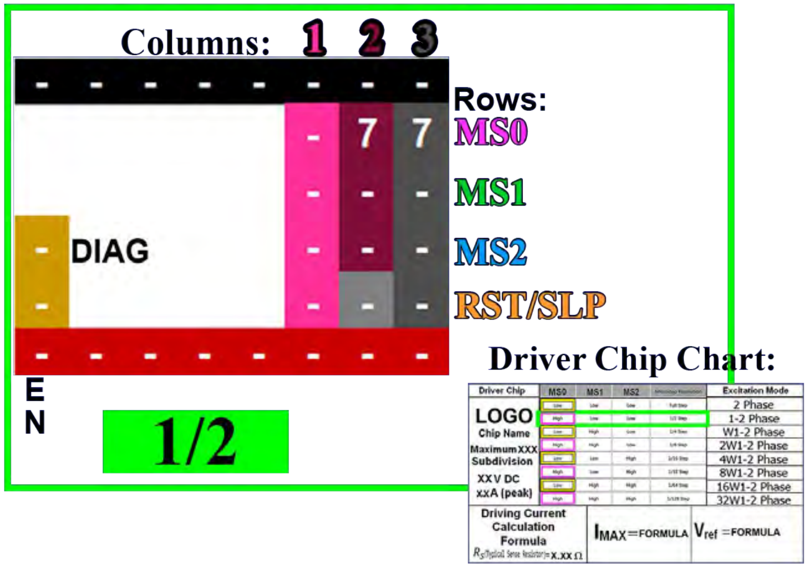


Stand-alone Mode

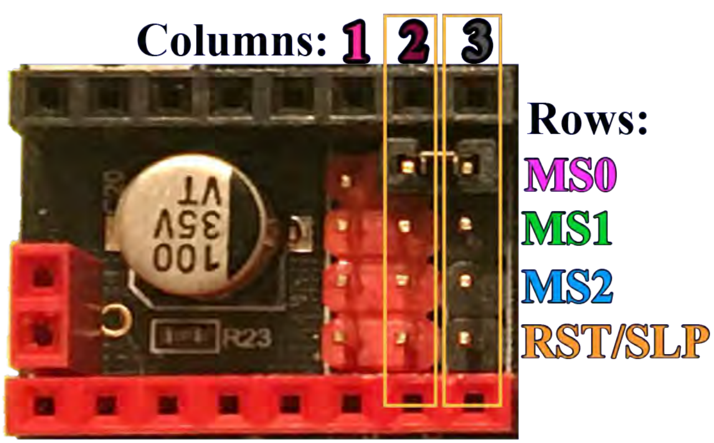
FYSETC LV8729

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

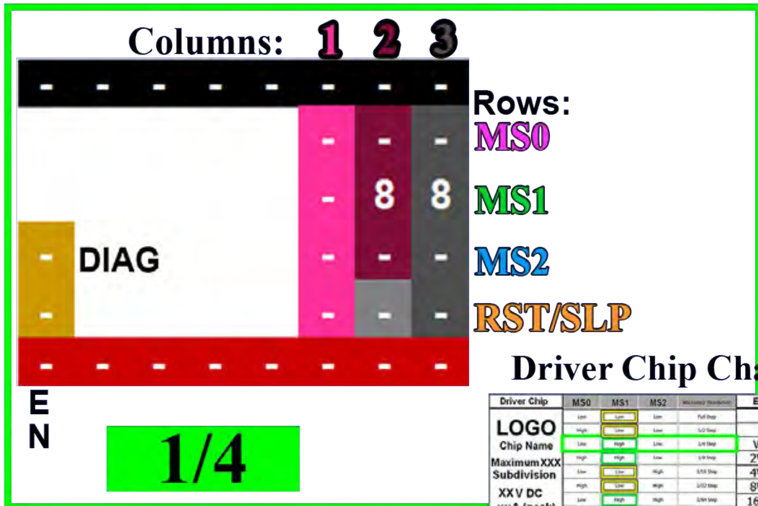
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

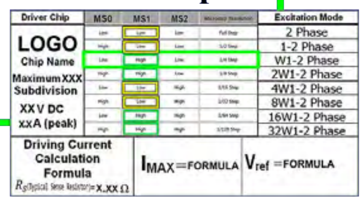
Stand-alone Mode

FYSETC LV8729

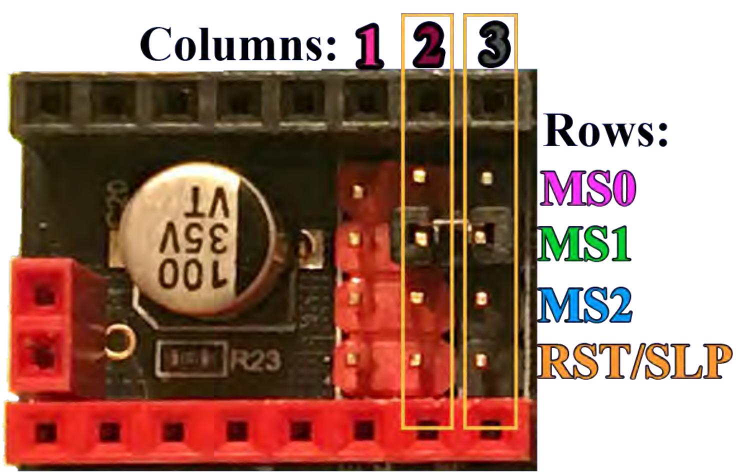
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

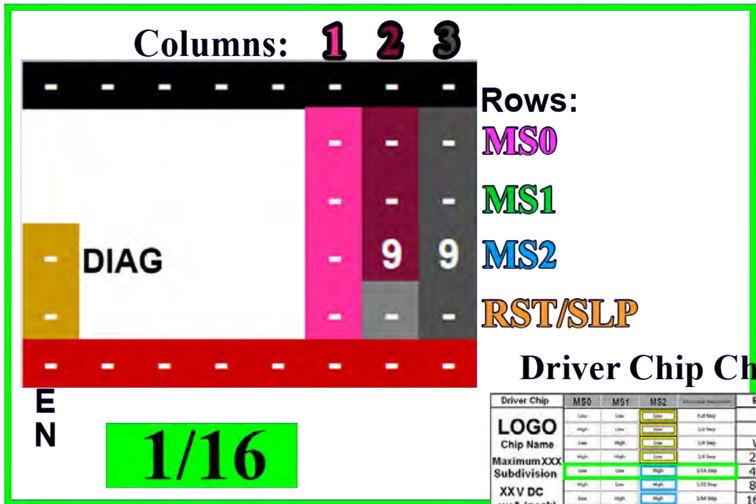
Meaning:

Driver Chip Chart:	Meaning
High	set Jumper between column 2 and column 3 on the MS1 row
Low	No Jumper Set

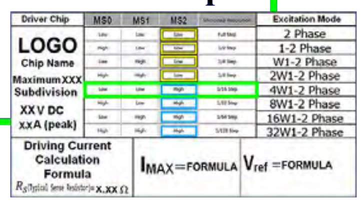
Stand-alone Mode

FYSETC LV8729

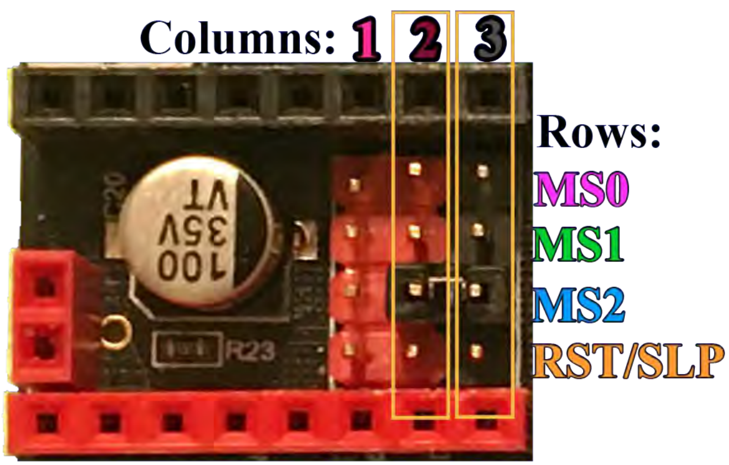
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

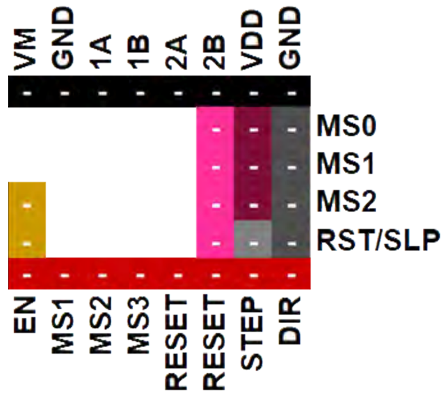
High → **set Jumper between column 2 and column 3 on the MS2 row**

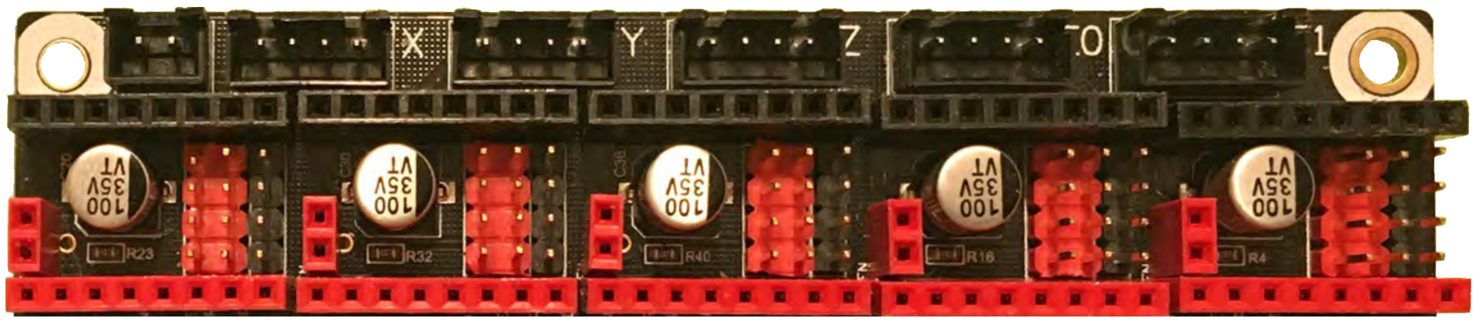
Low → **No Jumper Set**

Stand-alone Mode

FYSETC LV8729

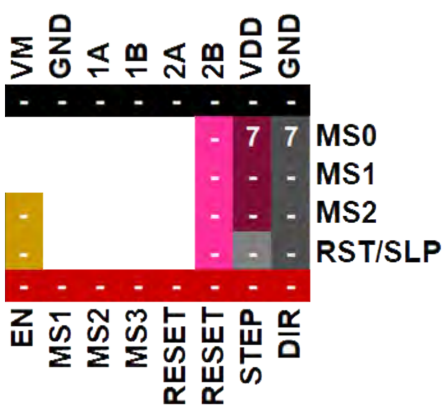
STEP

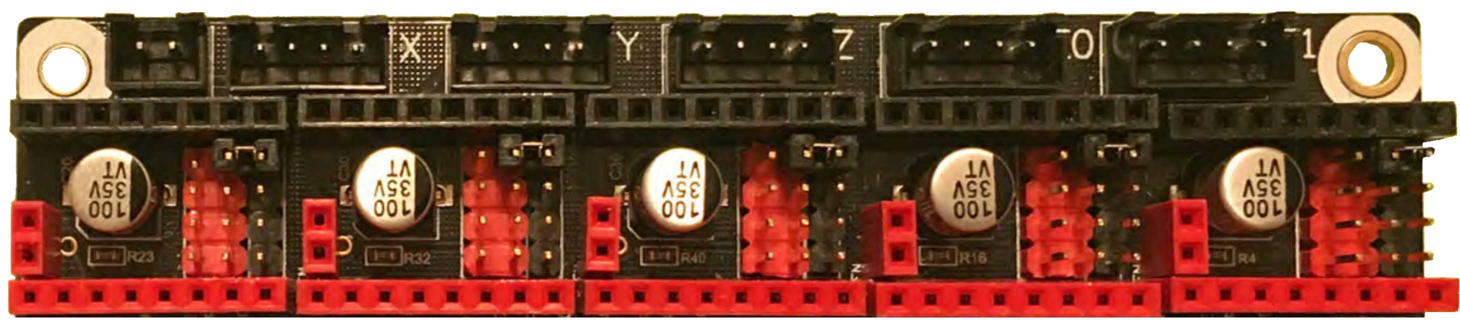




See Appendix D for legend

1 / 2



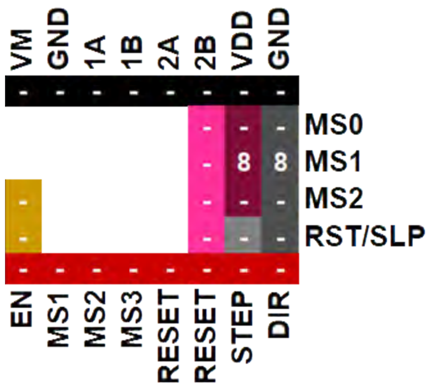


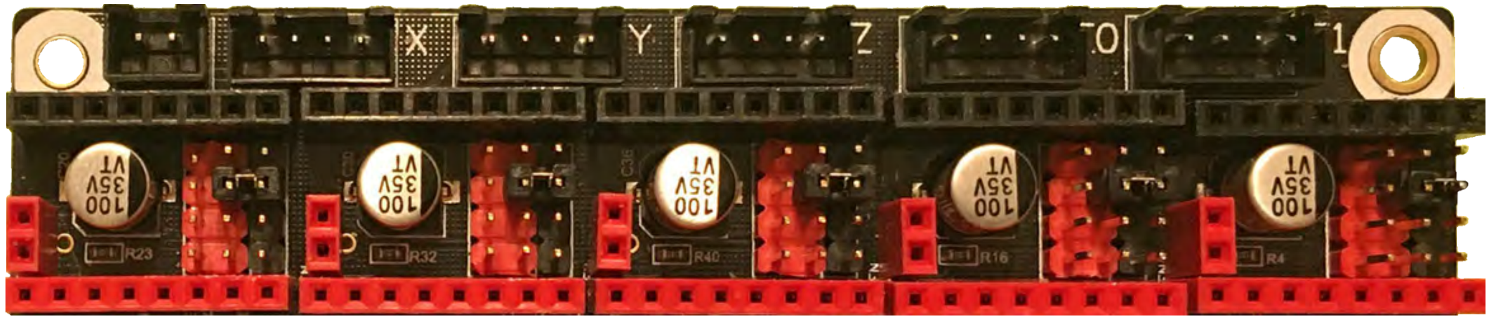
See Appendix D for legend

Stand-alone Mode

FYSETC LV8729

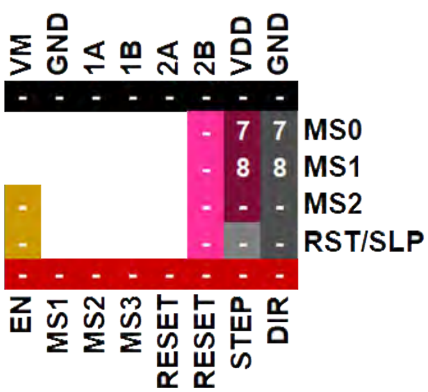
1 / 4

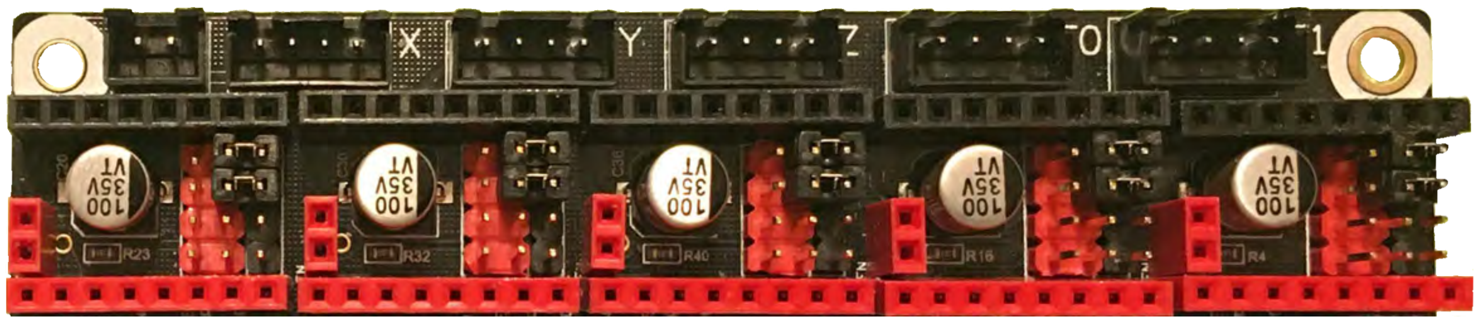




See Appendix D for legend

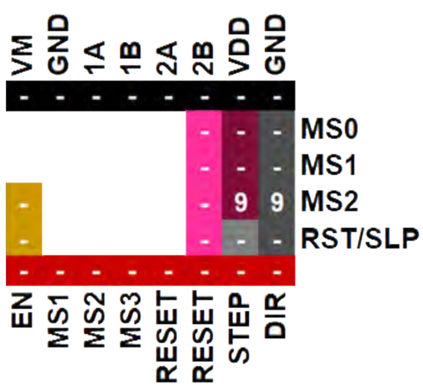
1 / 8

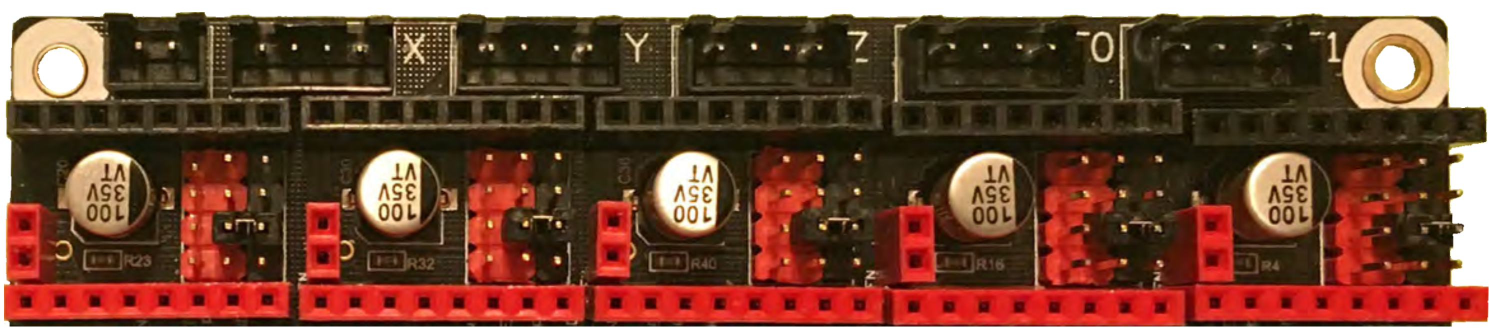




See Appendix D for legend

1 / 16



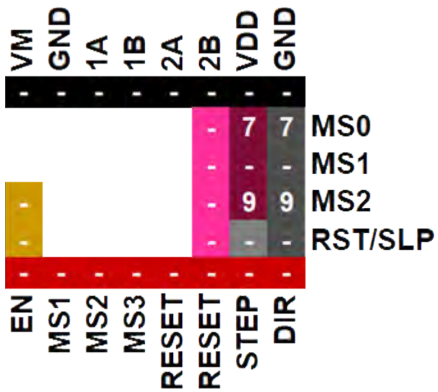


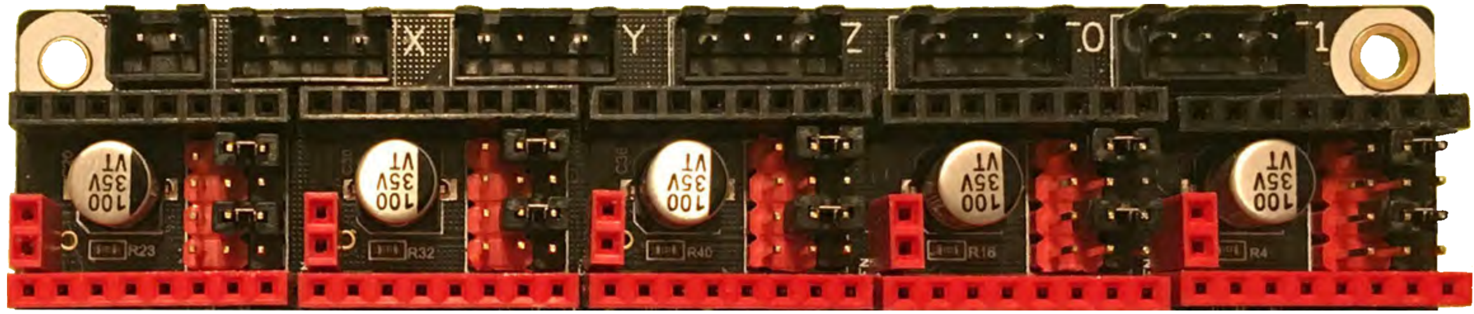
See Appendix D for legend

Stand-alone Mode

FYSETC LV8729

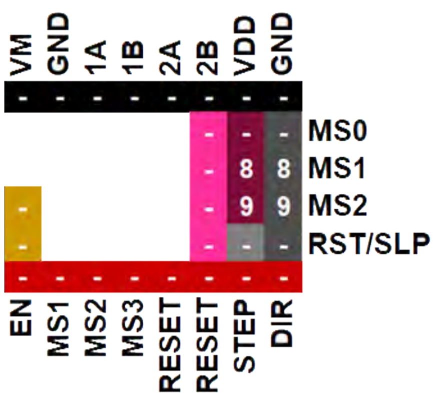
1 / 32

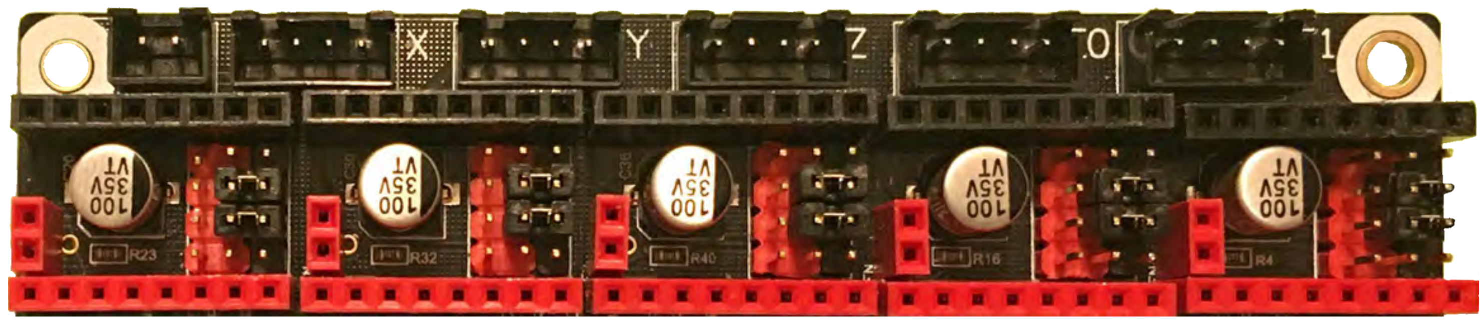




See Appendix D for legend

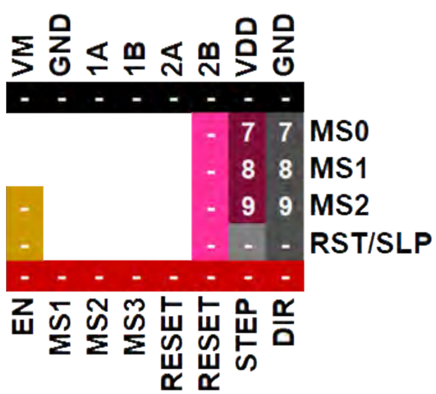
1 / 64

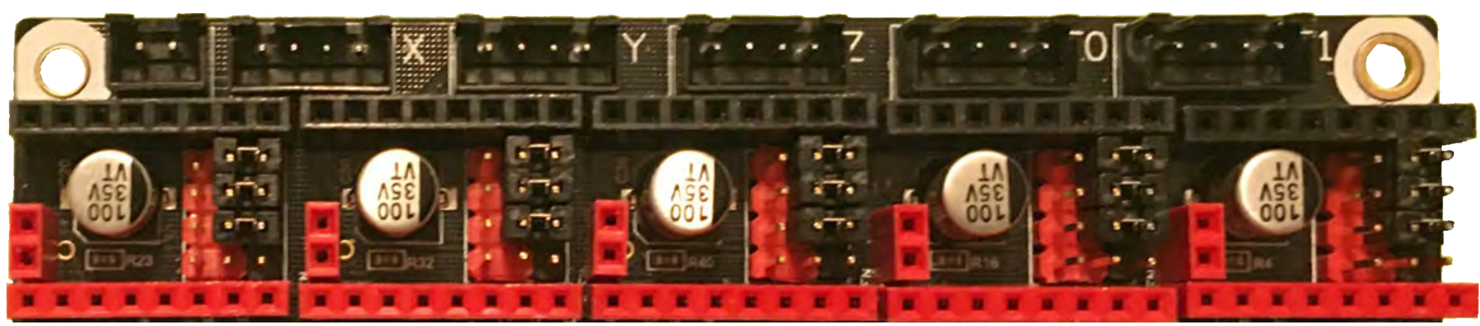




See Appendix D for legend

1 / 128



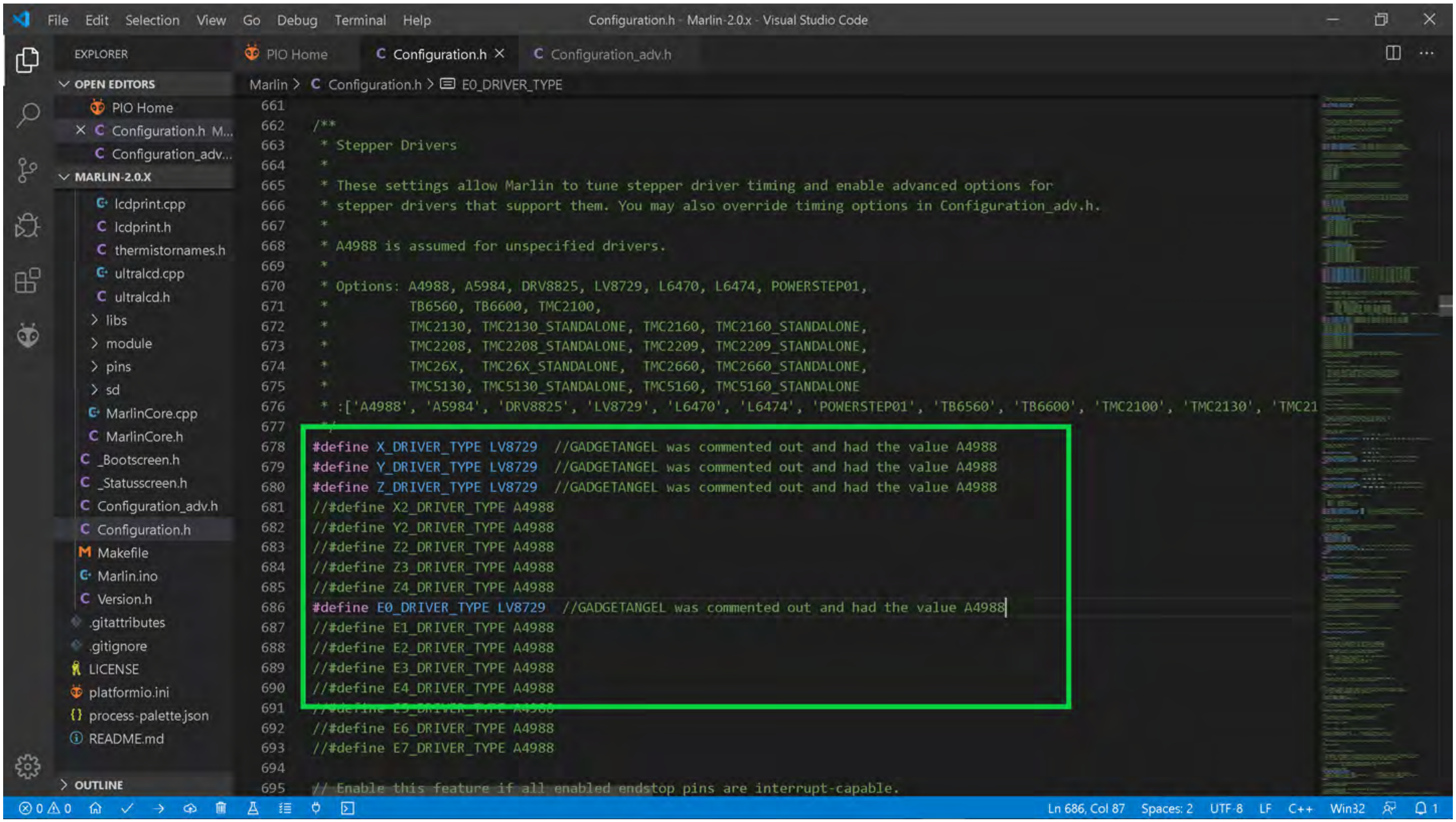


See Appendix D for legend

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

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

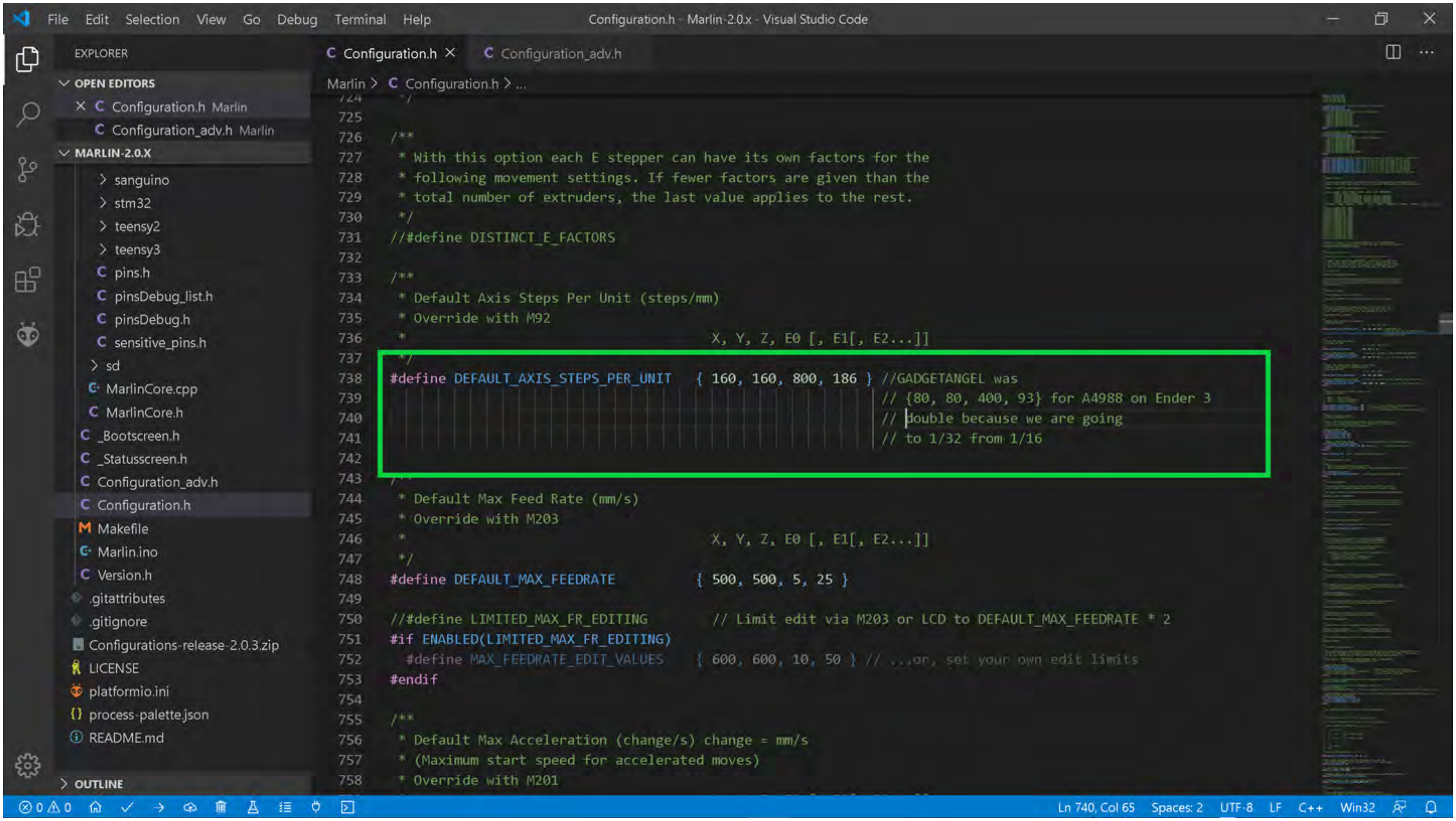
- Change the stepper motor drivers so that Marlin knows you are using FYSETC LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC LV8729 drivers. When two "//" appear at the beginning of a line that means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



- Go to the next page.

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

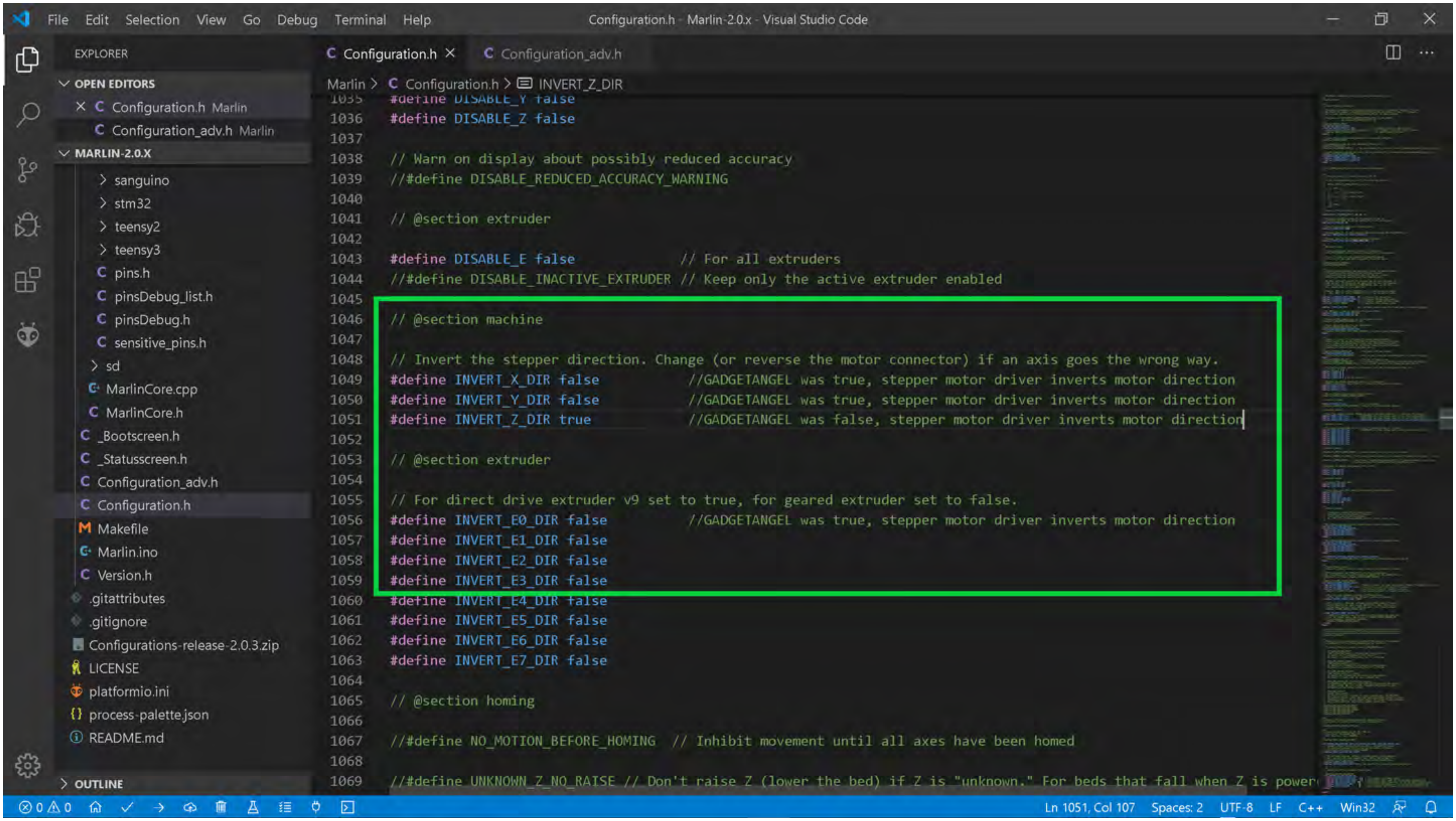
- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

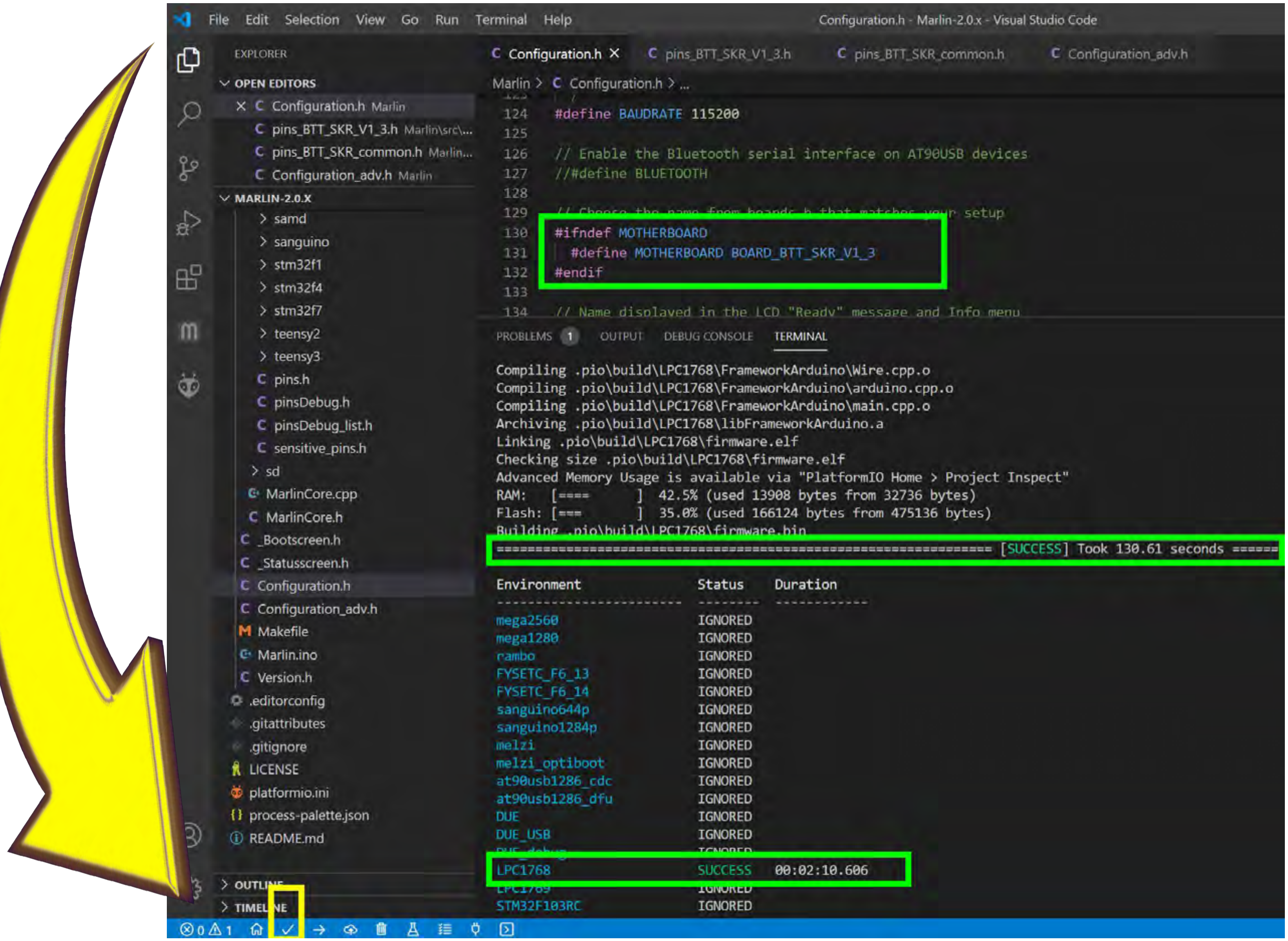
- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver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



- Go to the next page.

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

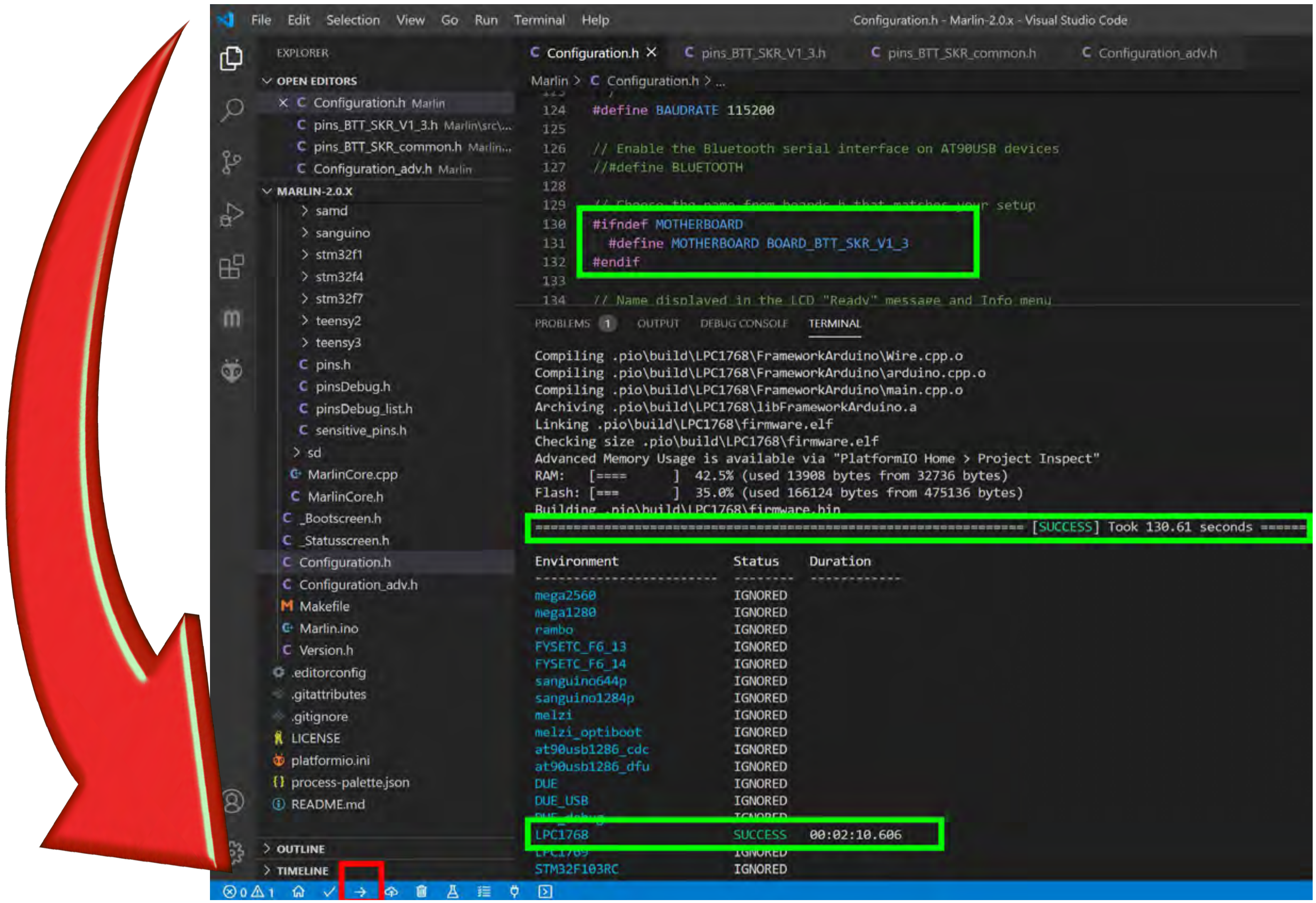
- The end of Marlin setup for FYSETC LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



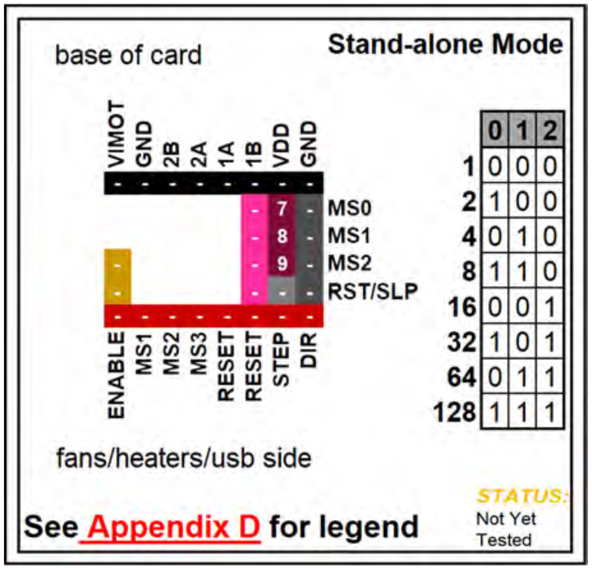
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.



LERDGE LV8729

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

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

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)$=0.22\Omega$	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$	$V_{ref} = 5 * I_{MAX} * R_S$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

LERDGE LV8729

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
x.xA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= x.xx Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

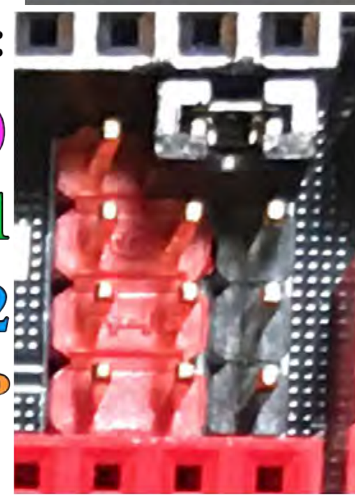
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

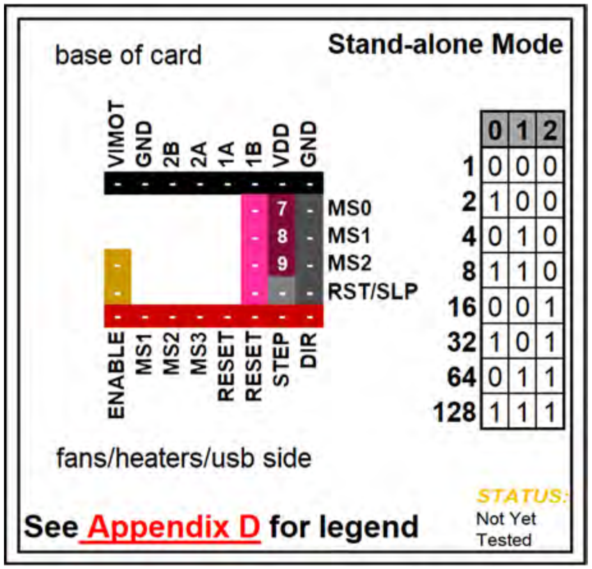
MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

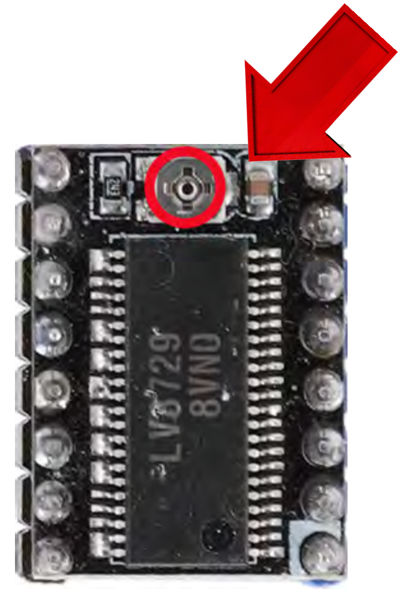
LERDGE LV8729



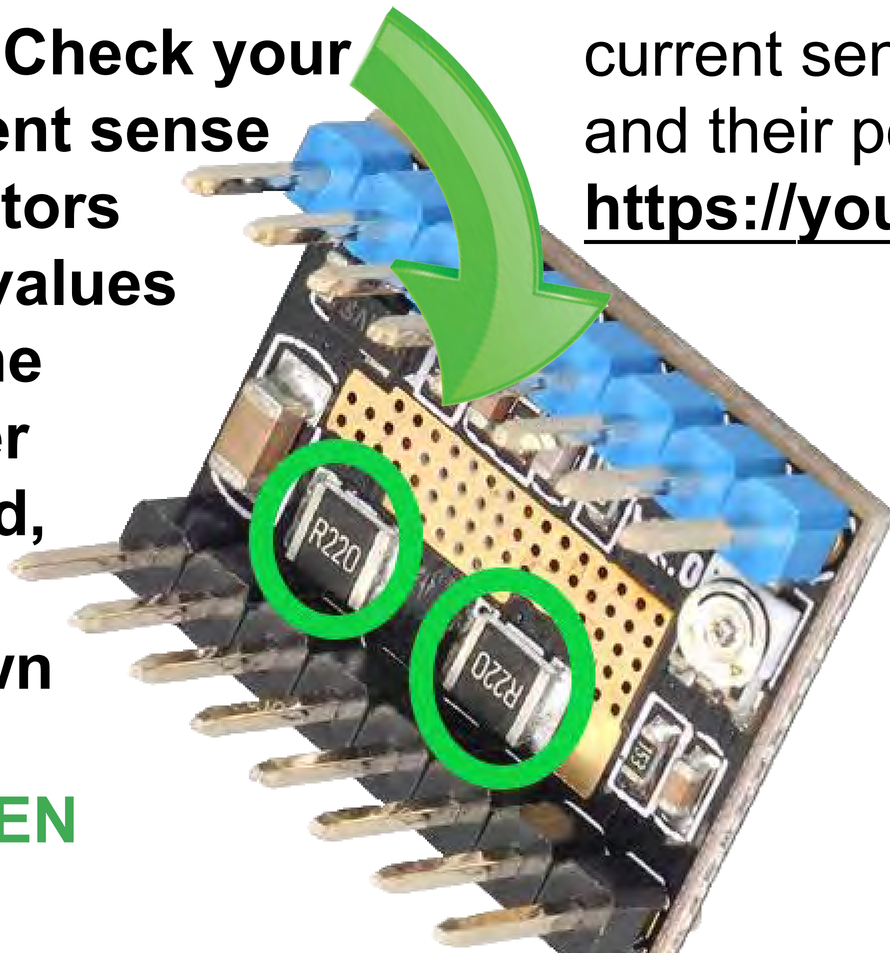
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

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



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



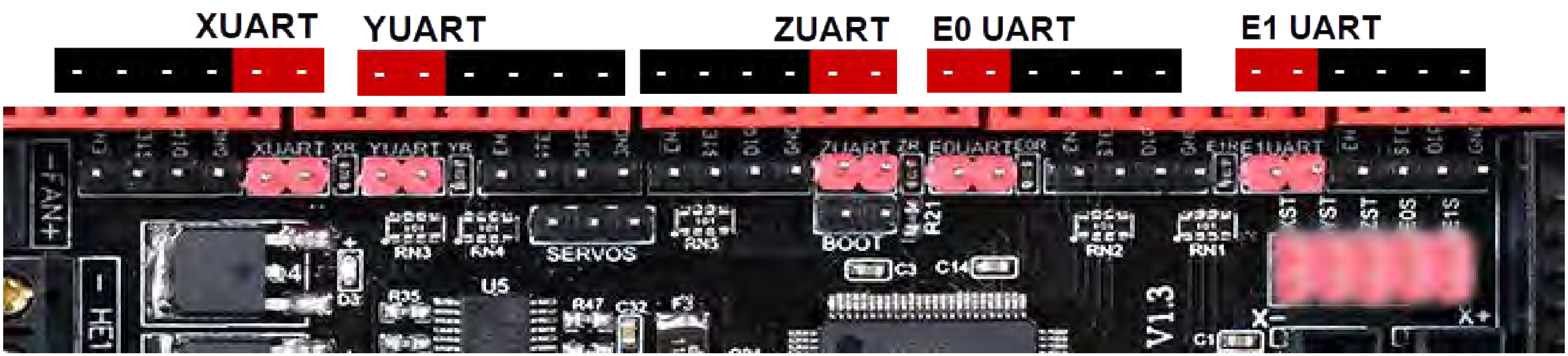
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

LERDGE LV8729

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

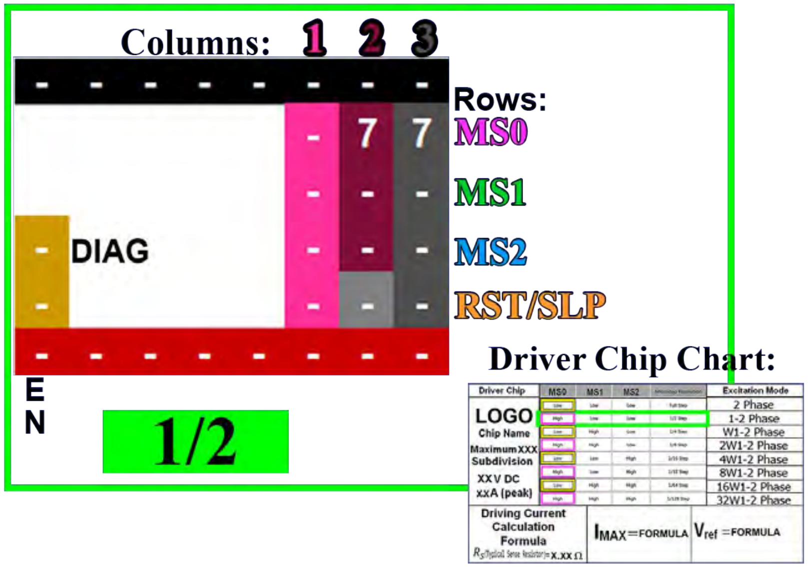


Stand-alone Mode

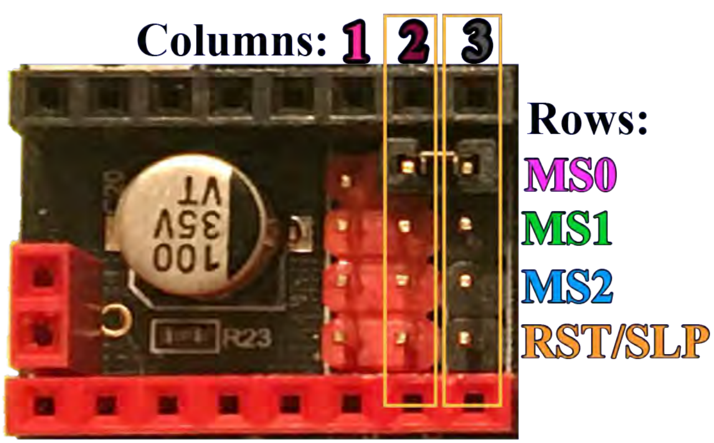
LERDGE LV8729

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Meaning:

Driver Chip Chart:

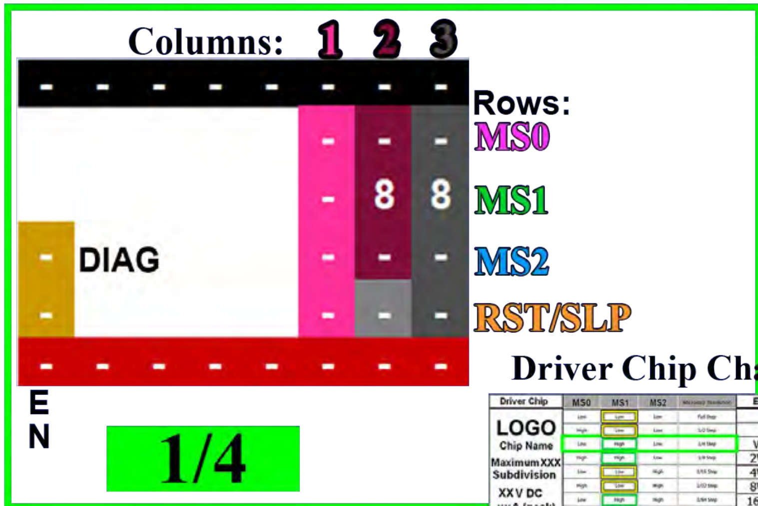
High → **set Jumper between column 2 and column 3 on the MS0 row**

Low → **No Jumper Set**

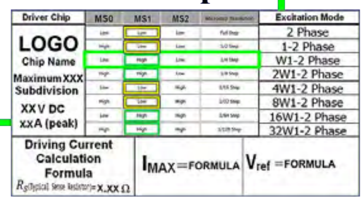
Stand-alone Mode

LERDGE LV8729

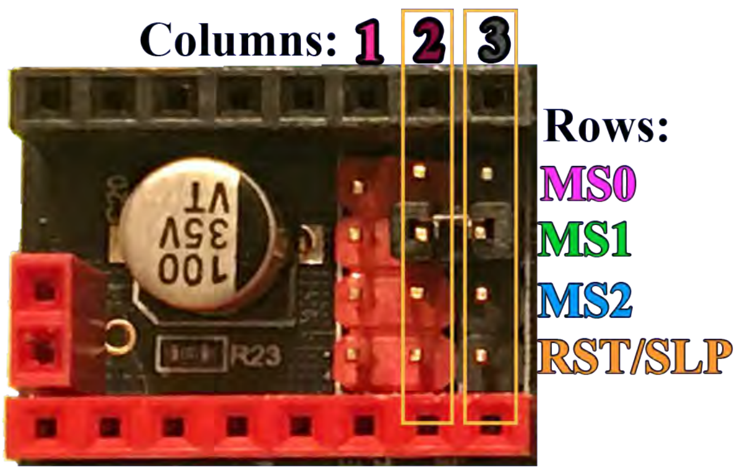
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 8 8 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:

Meaning:

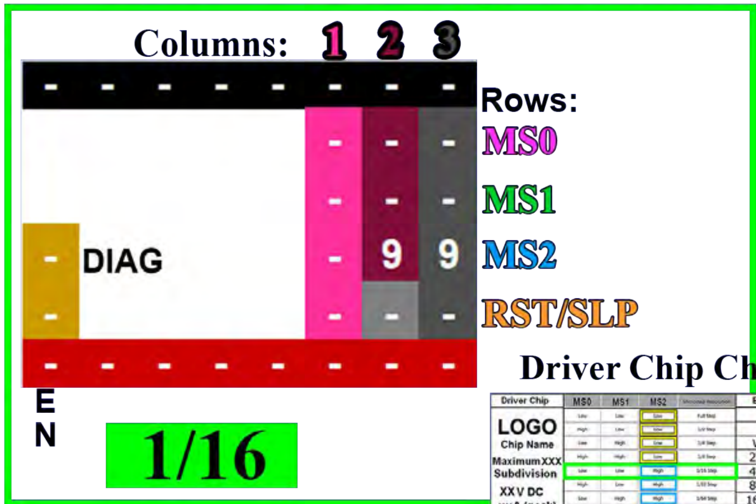
High → **set Jumper between column 2 and column 3 on the MS1 row**

Low → **No Jumper Set**

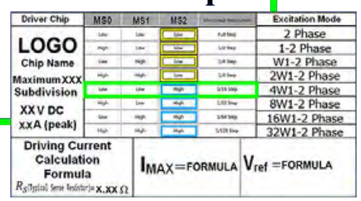
Stand-alone Mode

LERDGE LV8729

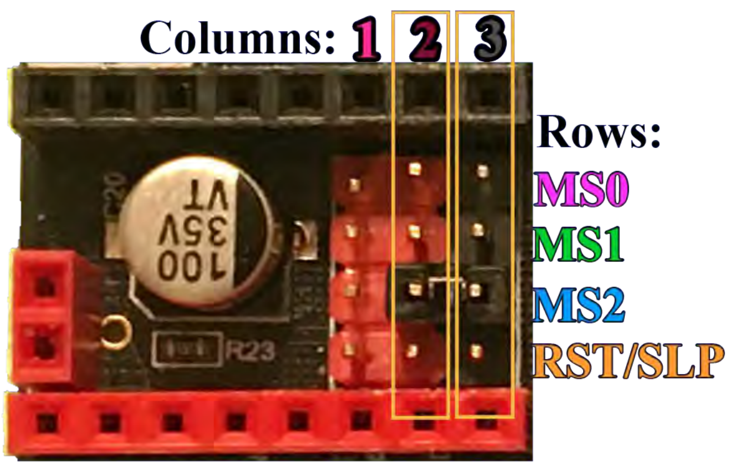
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

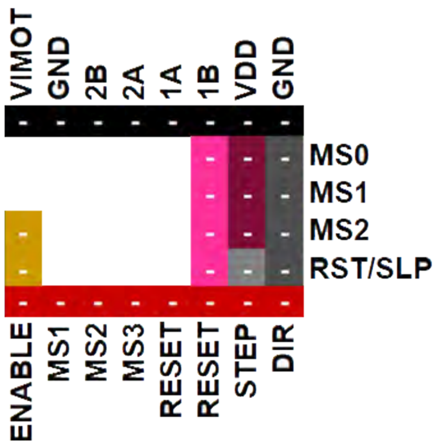
High → **set Jumper between column 2 and column 3 on the MS2 row**

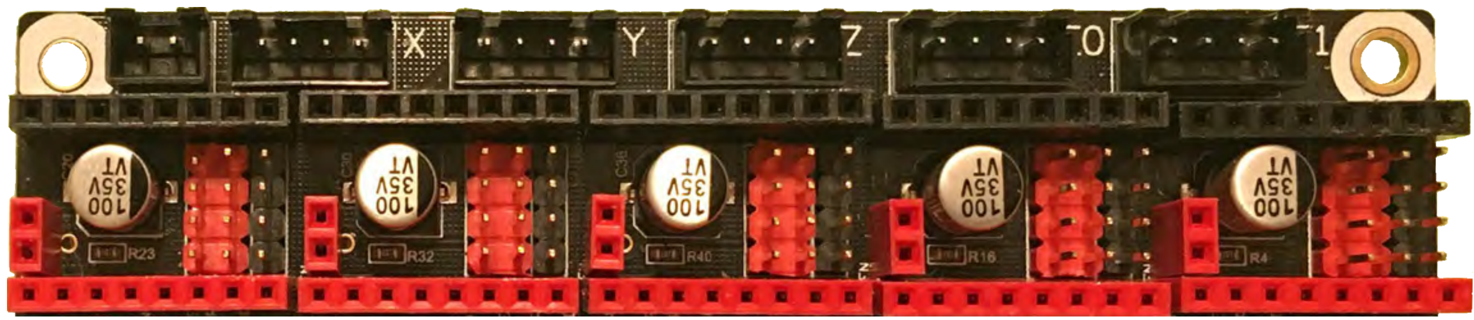
Low → **No Jumper Set**

Stand-alone Mode

LERDGE LV8729

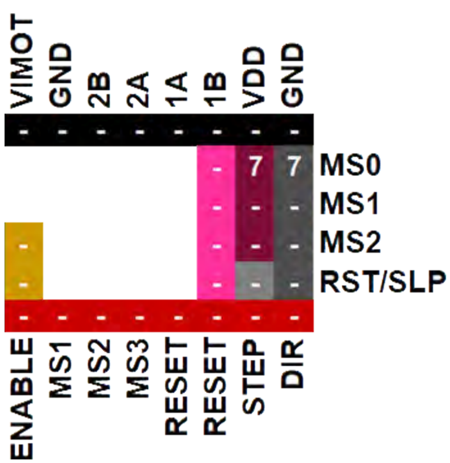
STEP

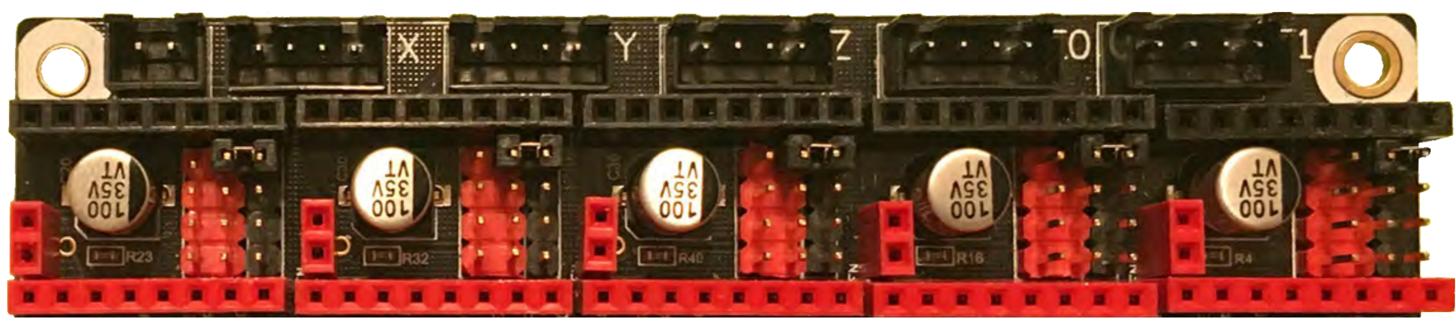




See Appendix D for legend

1 / 2



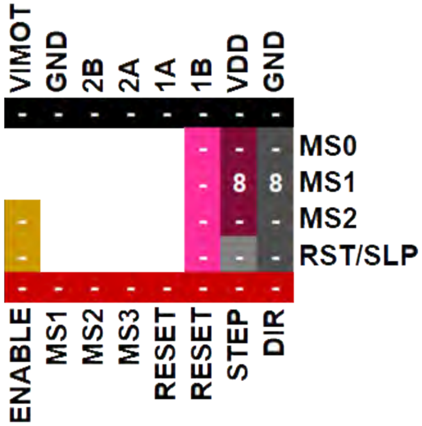


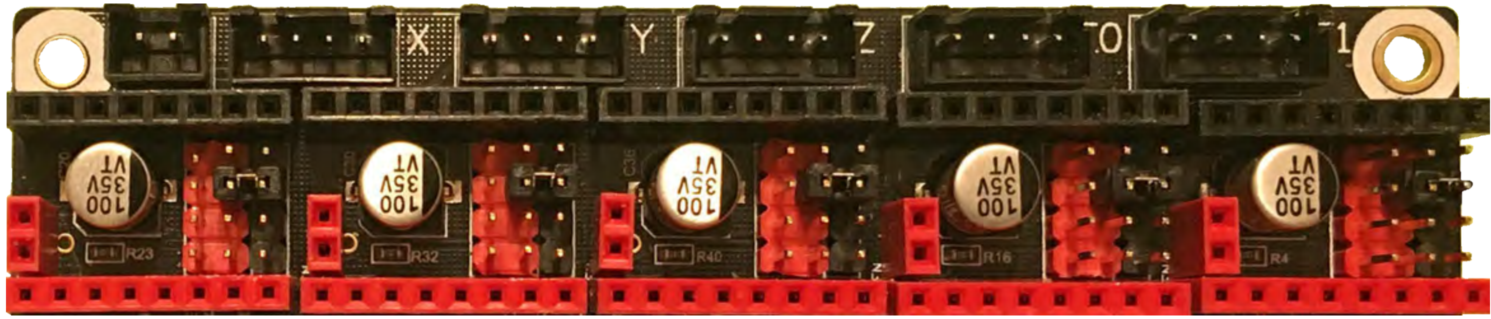
See Appendix D for legend

Stand-alone Mode

LERDGE LV8729

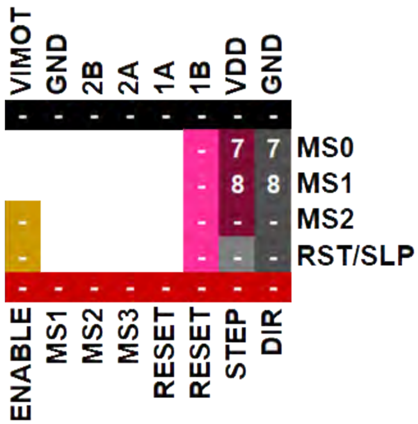
1 / 4

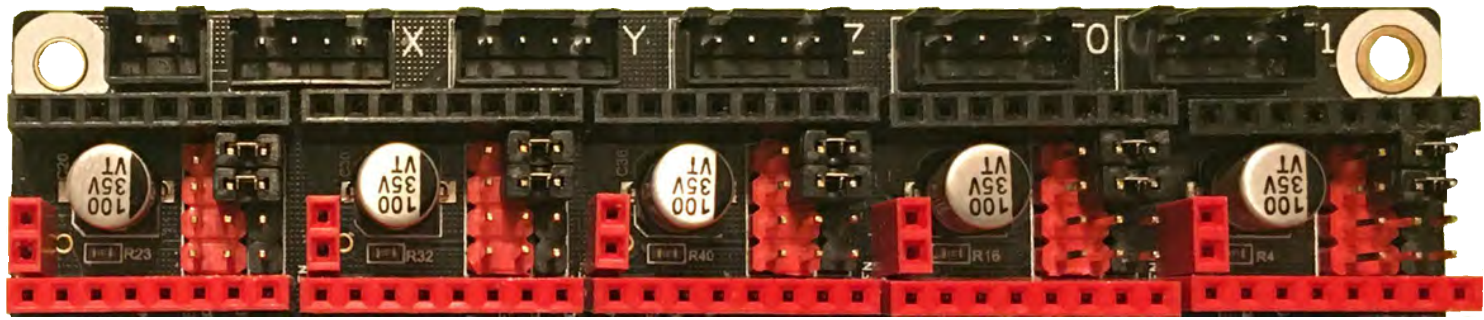




See Appendix D for legend

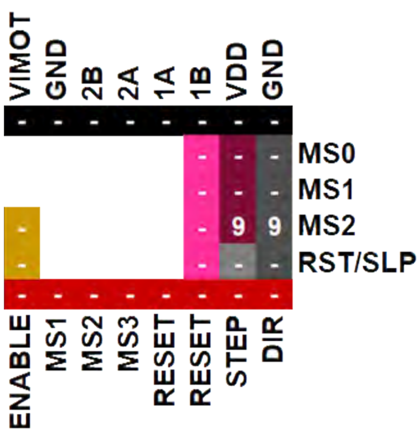
1 / 8

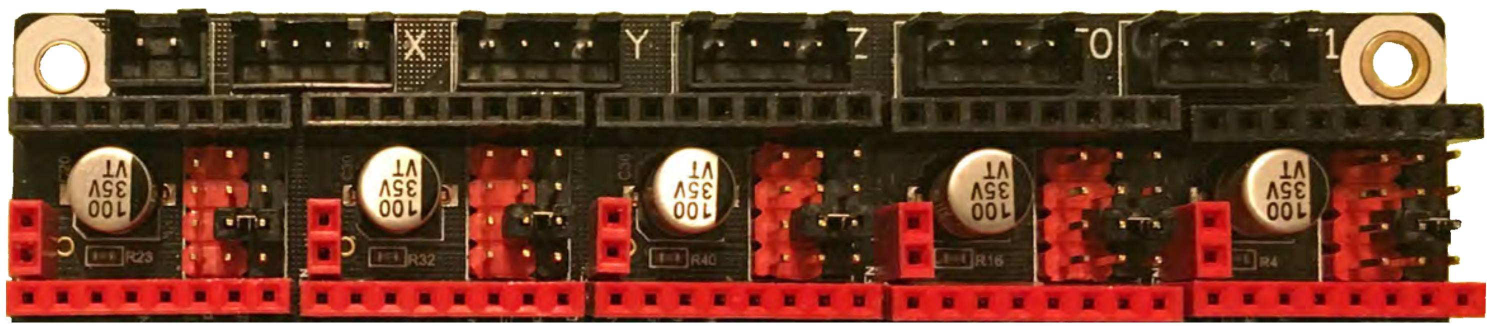




See Appendix D for legend

1 / 16



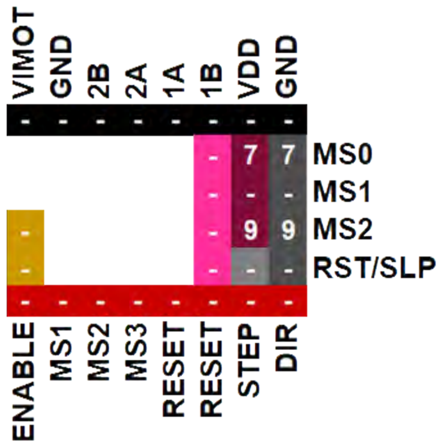


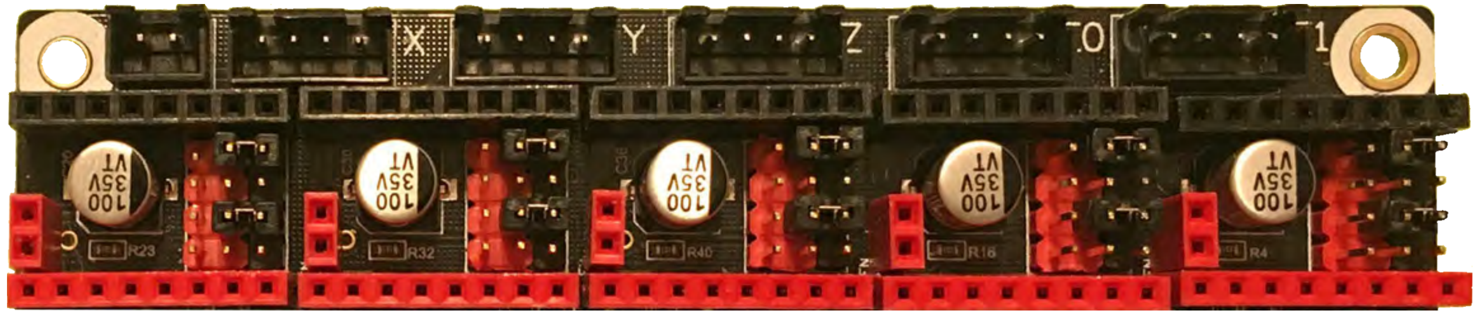
See Appendix D for legend

Stand-alone Mode

LERDGE LV8729

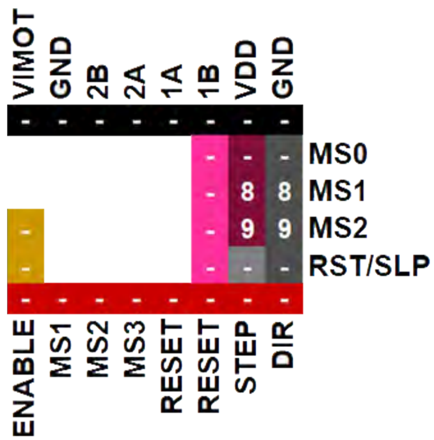
1 / 32

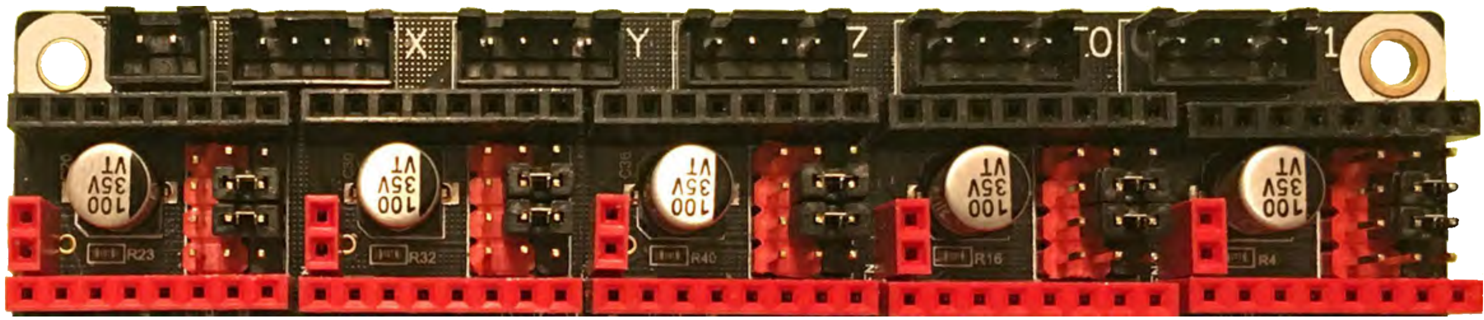




See Appendix D for legend

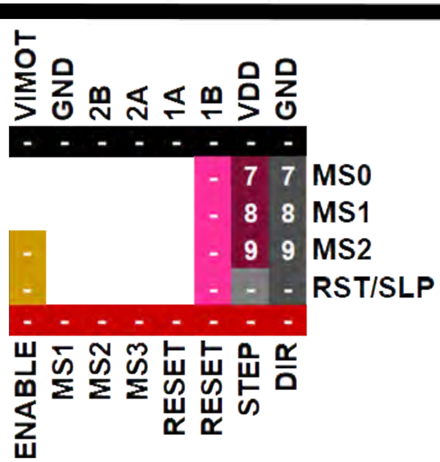
1 / 64

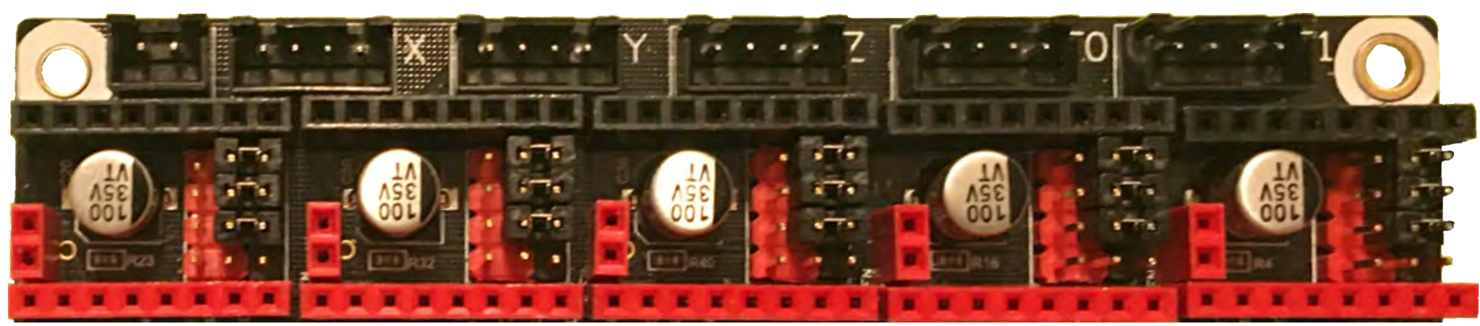




See Appendix D for legend

1 / 128



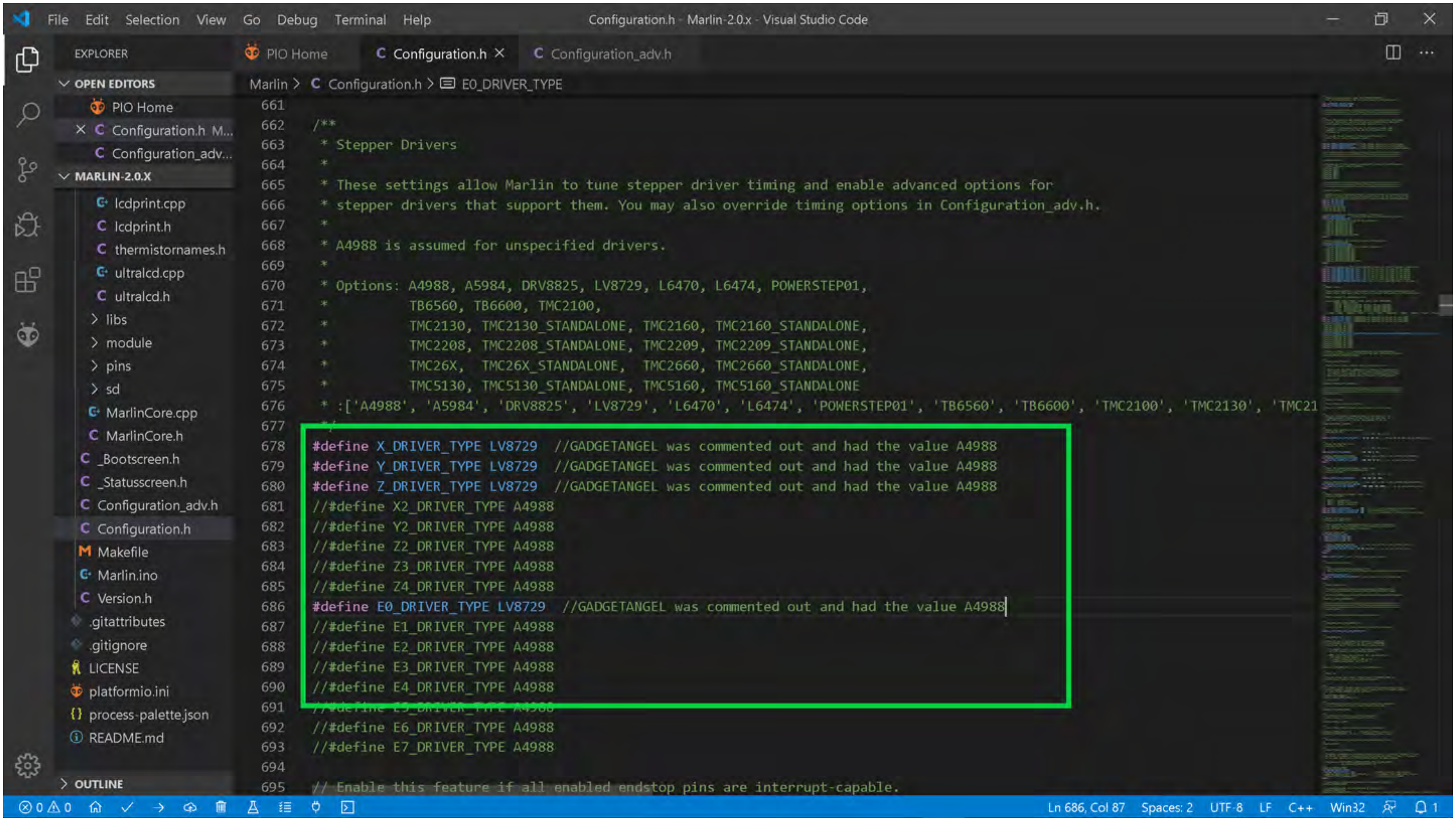


See Appendix D for legend

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

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

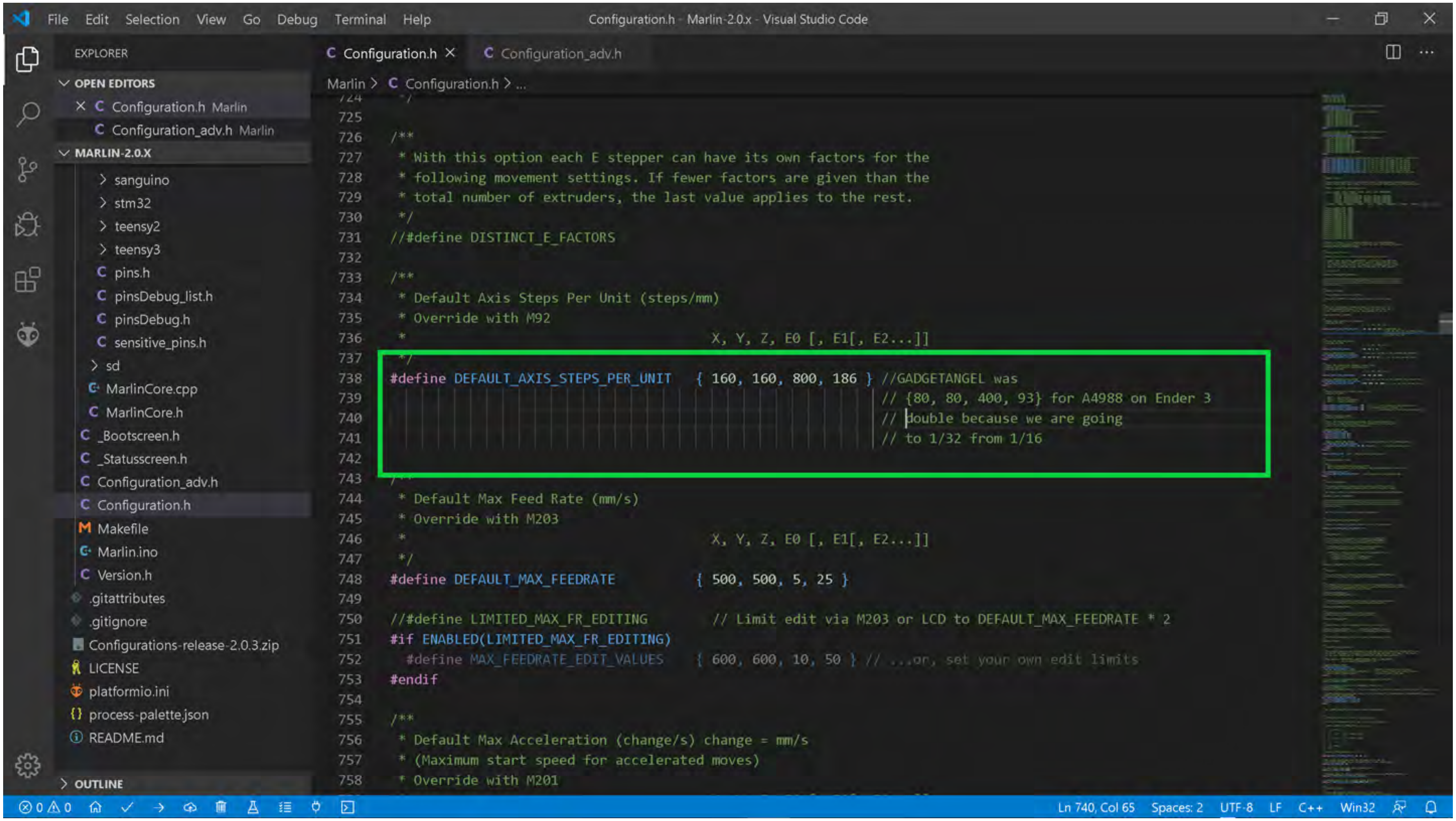
- Change the stepper motor drivers so that Marlin knows you are using Lerdge LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use Lerdge LV8729 drivers. When two "//" appear at the beginning of a line that means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



- Go to the next page.

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

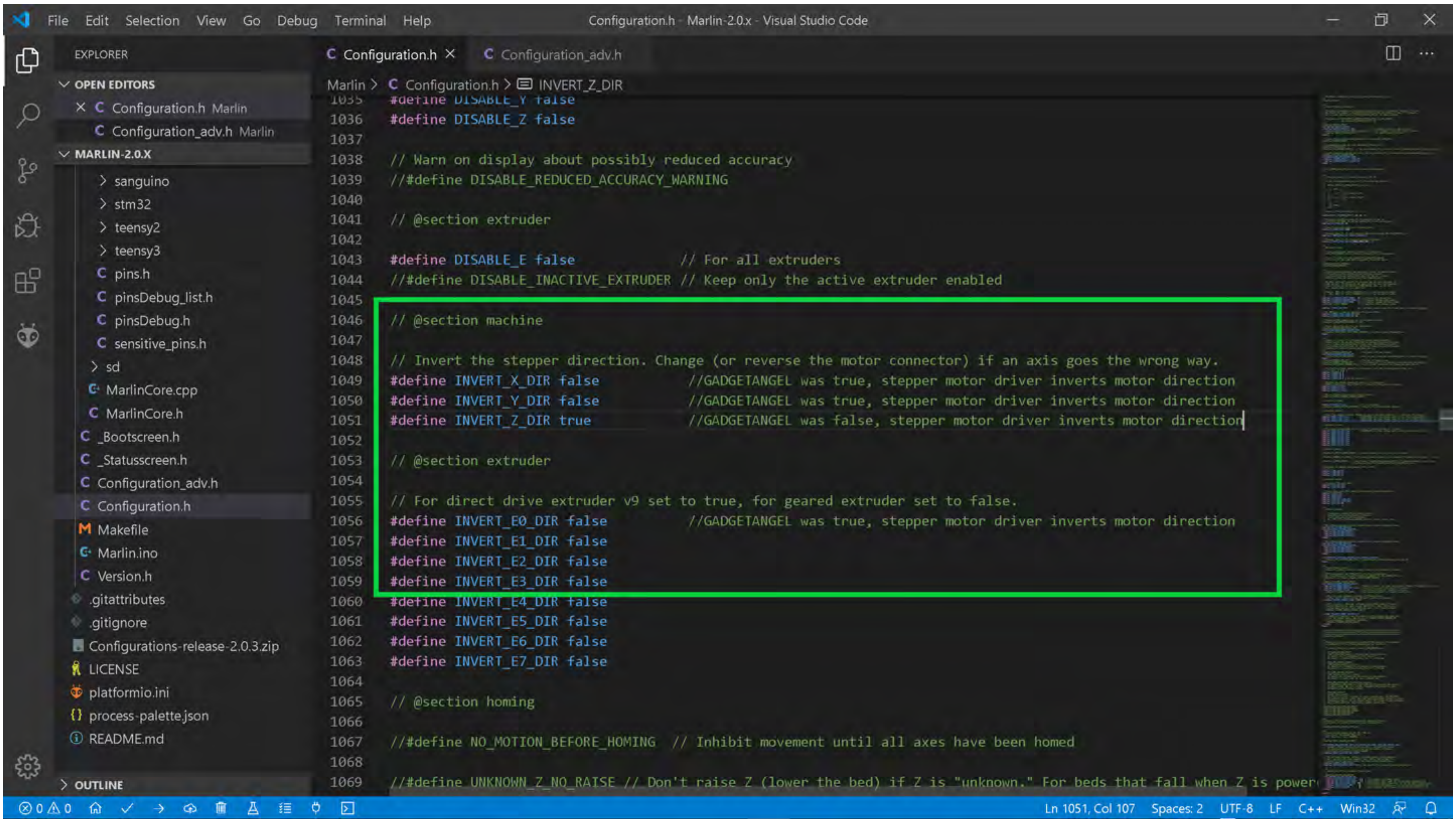
- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

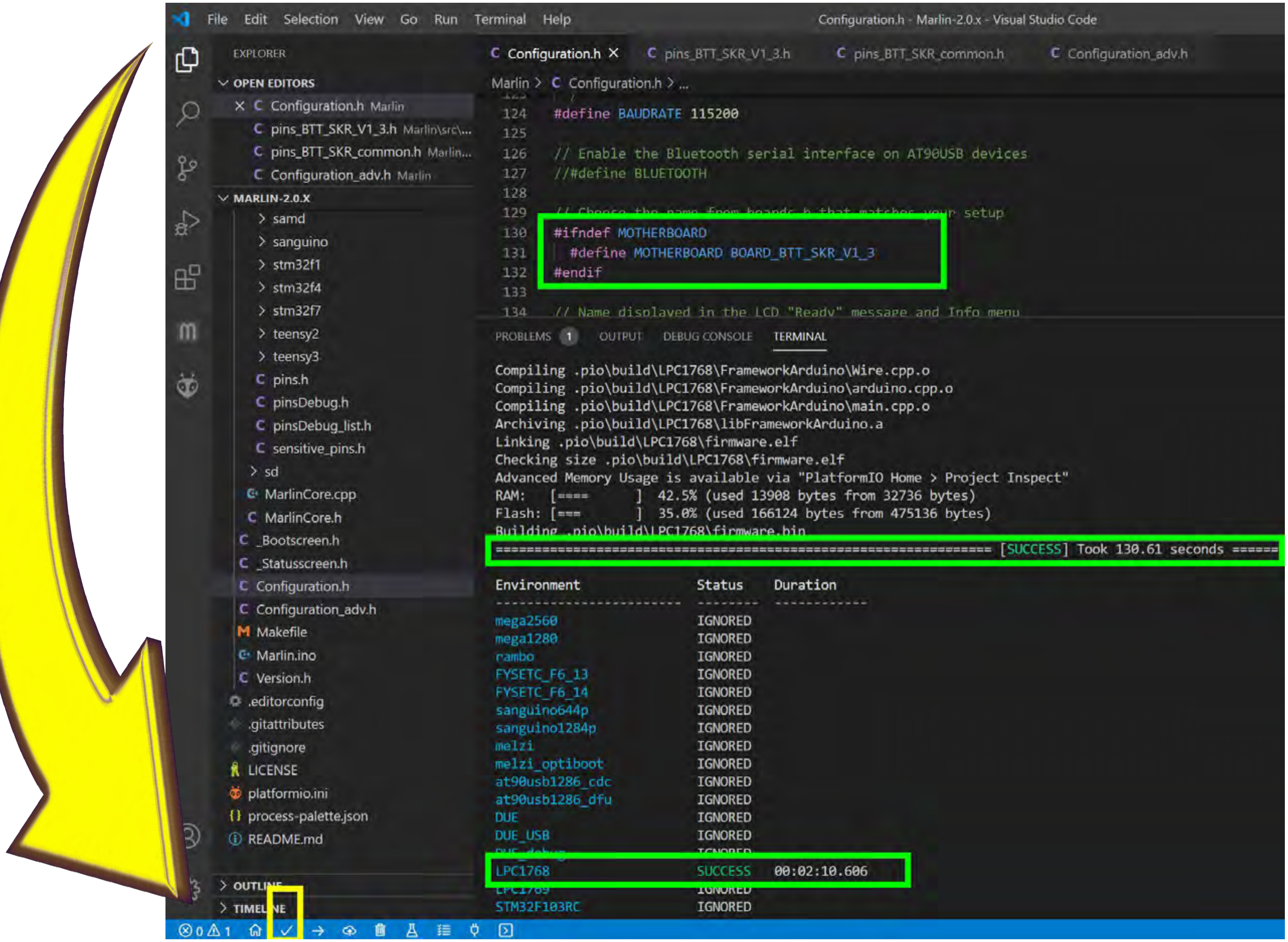
- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver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



- Go to the next page.

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

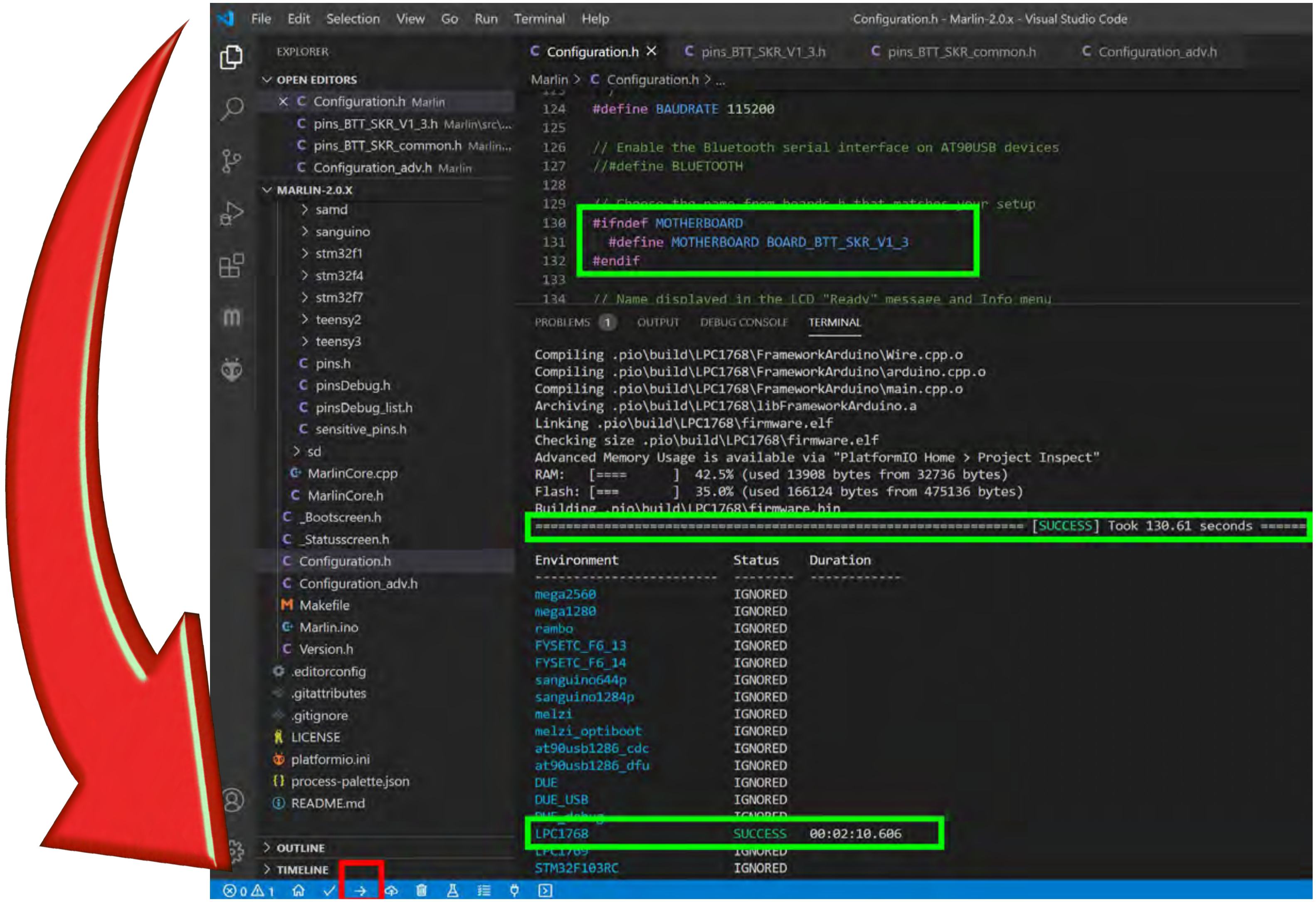
- The end of Marlin setup for LERDGE LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

MKS LV8729

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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

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

Driving Current Calculation Formula

R_S(Typical Sense Resistor)$=0.22\Omega$

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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

MKS LV8729

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
xxA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

High ⇨ Jumper Set

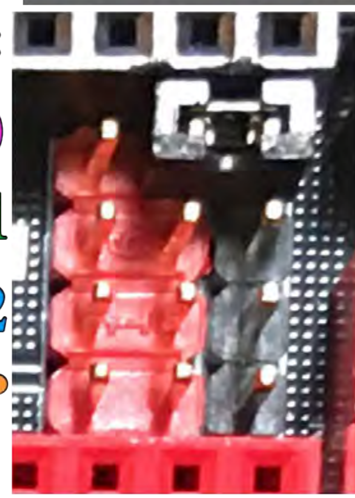
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

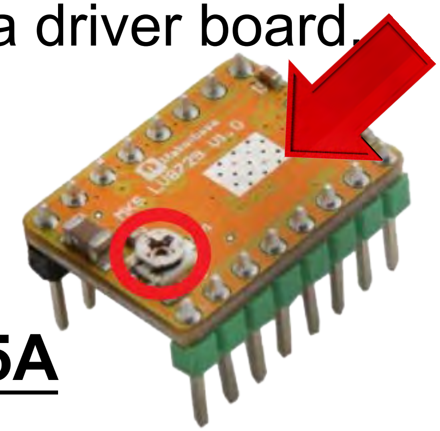
See Appendix D for legend

MKS LV8729

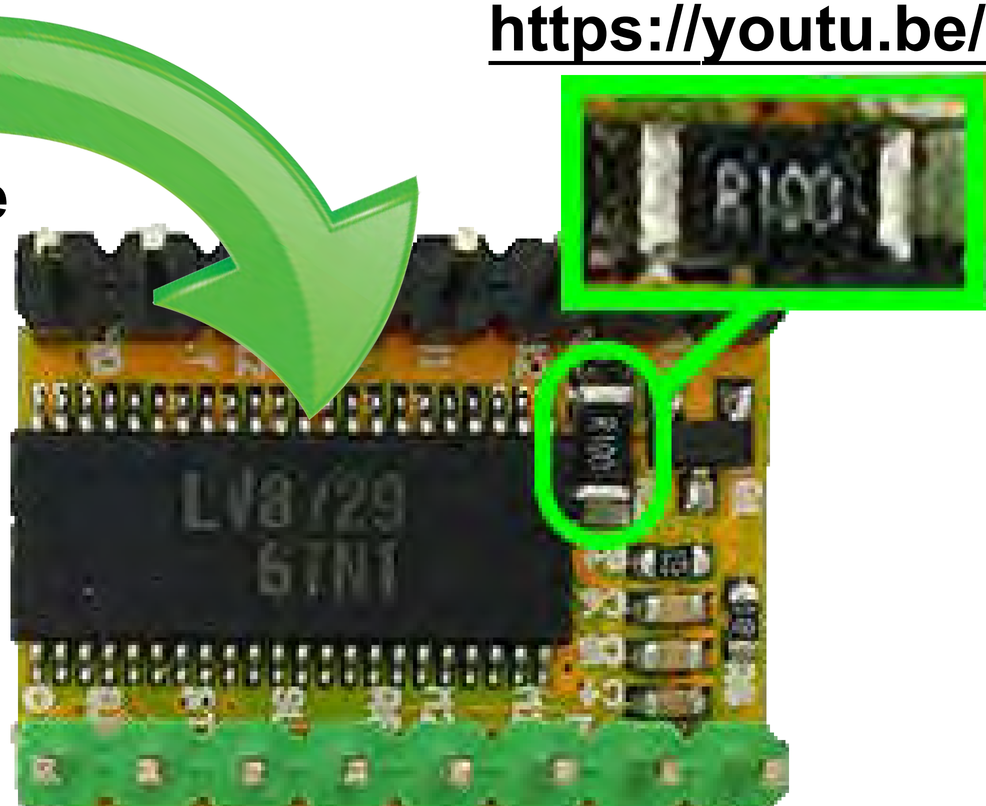
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

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



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



R_s = R050 is 0.05 Ohms

R_s = R068 is 0.068 Ohms

R_s = R100 is 0.1 Ohms

R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

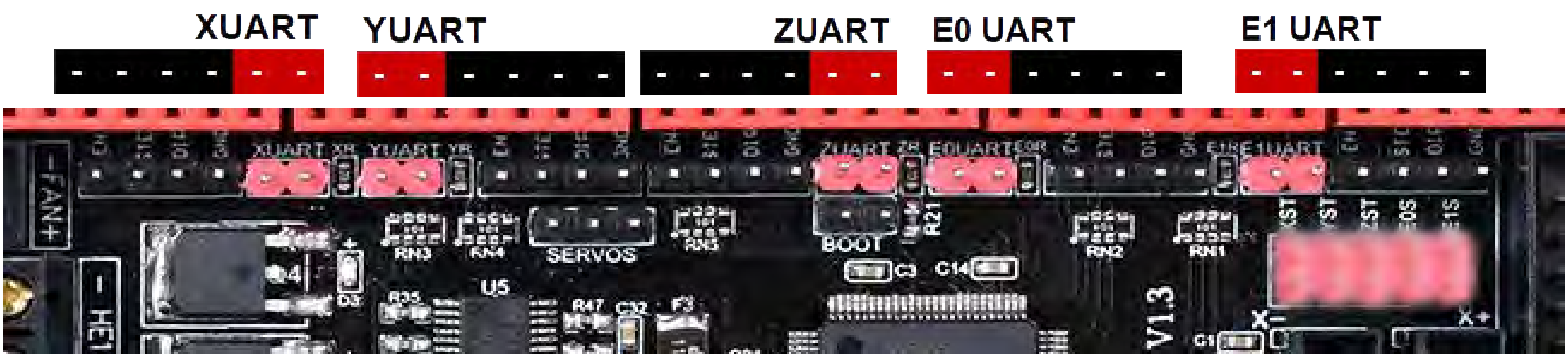
R_s = R220 is 0.22 Ohms

Stand-alone Mode

MKS LV8729

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

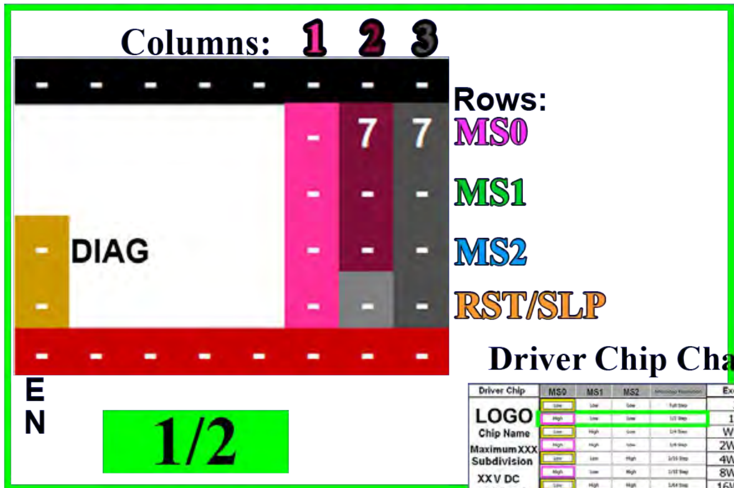


Stand-alone Mode

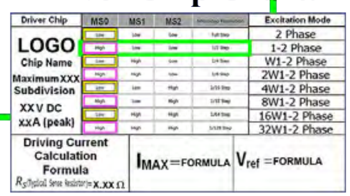
MKS LV8729

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

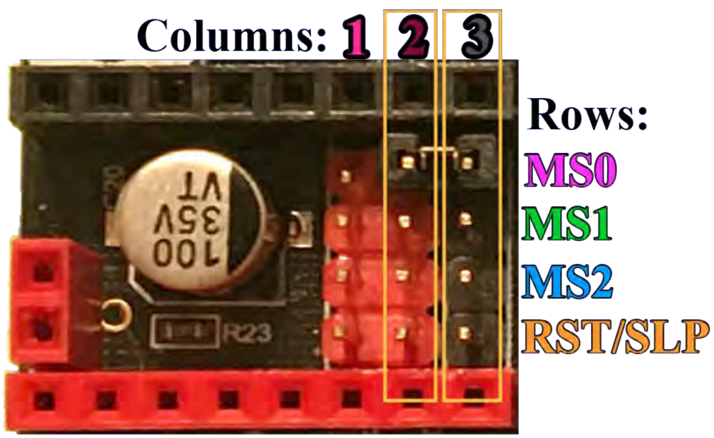
Driver Socket Representation



Driver Chip Chart:



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

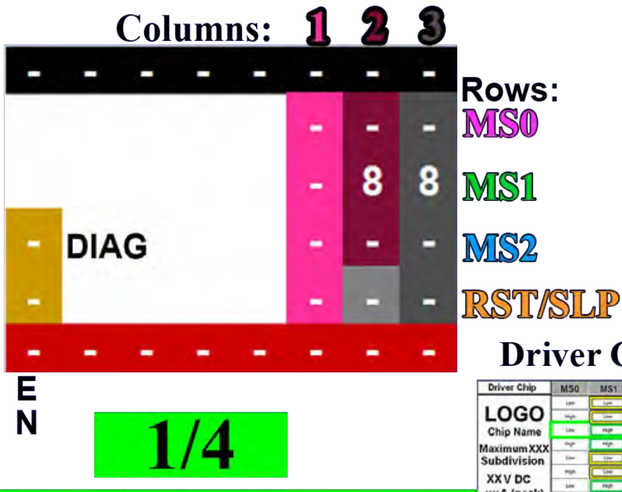
Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:	Meaning:
High →	set **Jumper** between column 2 and column 3 on the MS0 row
Low →	No **Jumper** Set

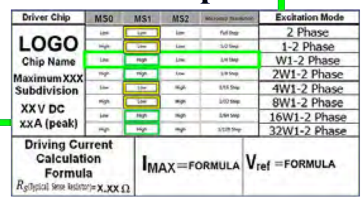
Stand-alone Mode

MKS LV8729

Driver Socket Representation

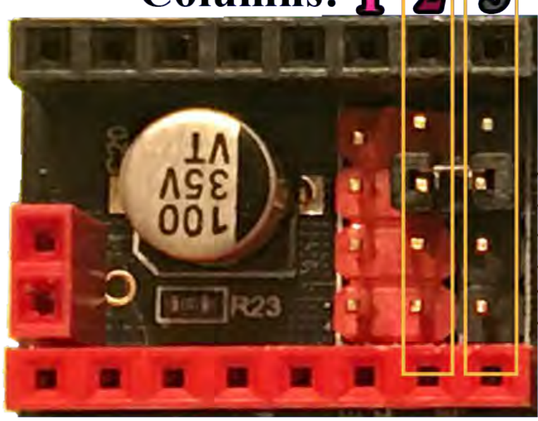


Driver Chip Chart:



High:

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

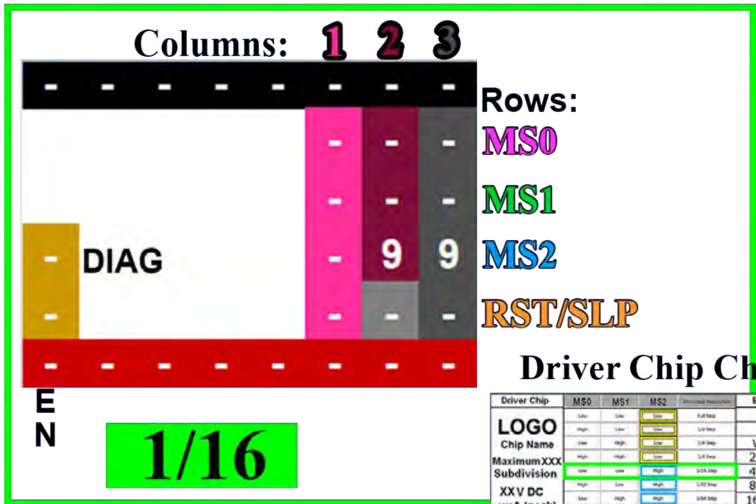
Meaning:

Driver Chip Chart:	
High	set **Jumper** between column 2 and column 3 on the MS1 row
Low	No **Jumper** Set

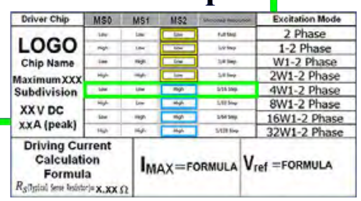
Stand-alone Mode

MKS LV8729

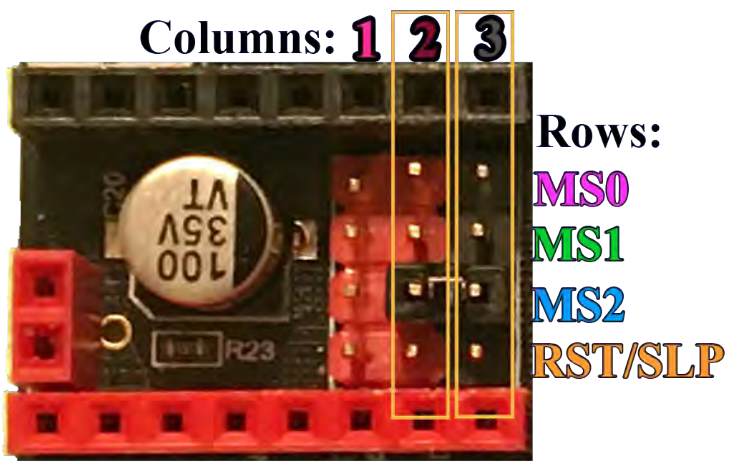
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

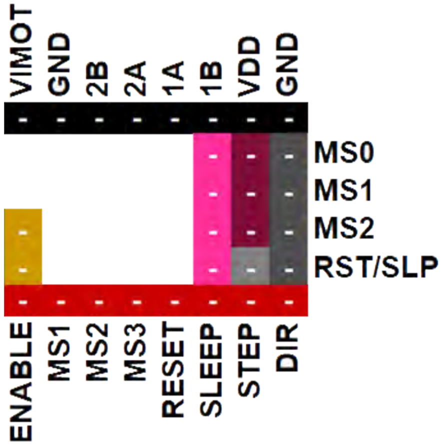
High → **set Jumper between column 2 and column 3 on the MS2 row**

Low → **No Jumper Set**

Stand-alone Mode

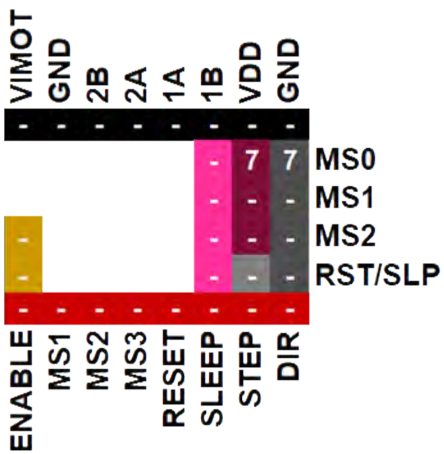
MKS LV8729

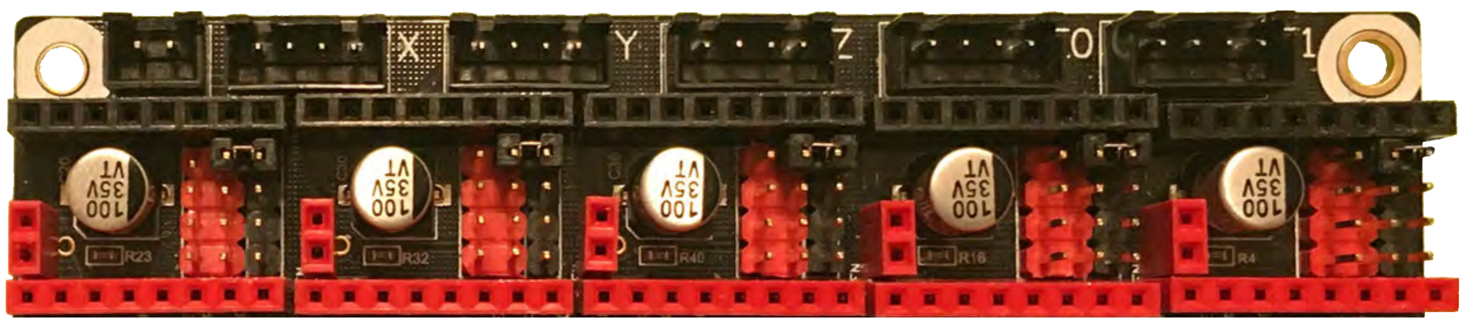
STEP



See Appendix D for legend

1 / 2



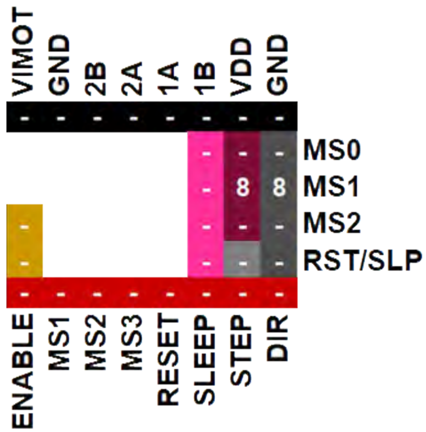


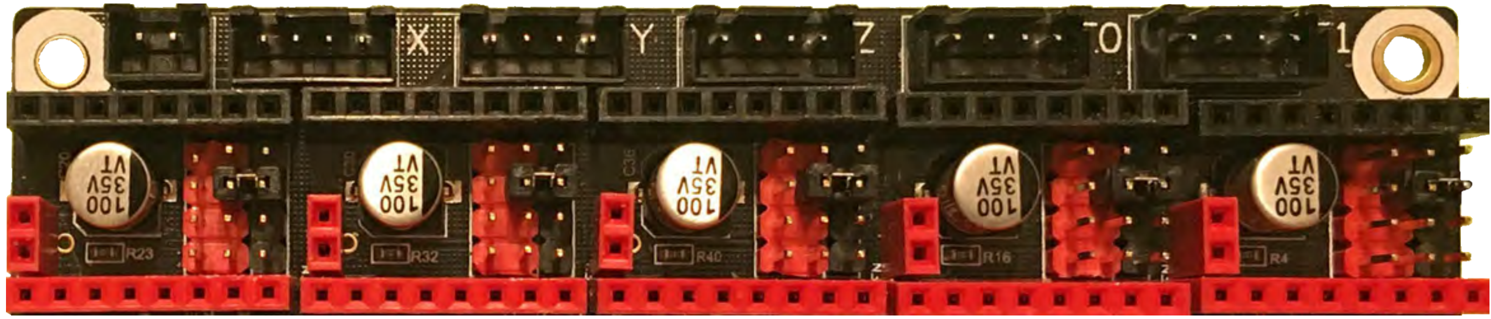
See Appendix D for legend

Stand-alone Mode

MKS LV8729

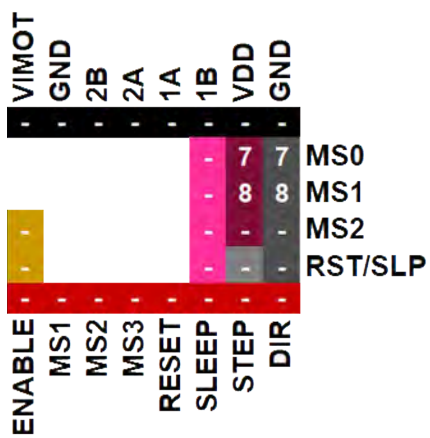
1 / 4

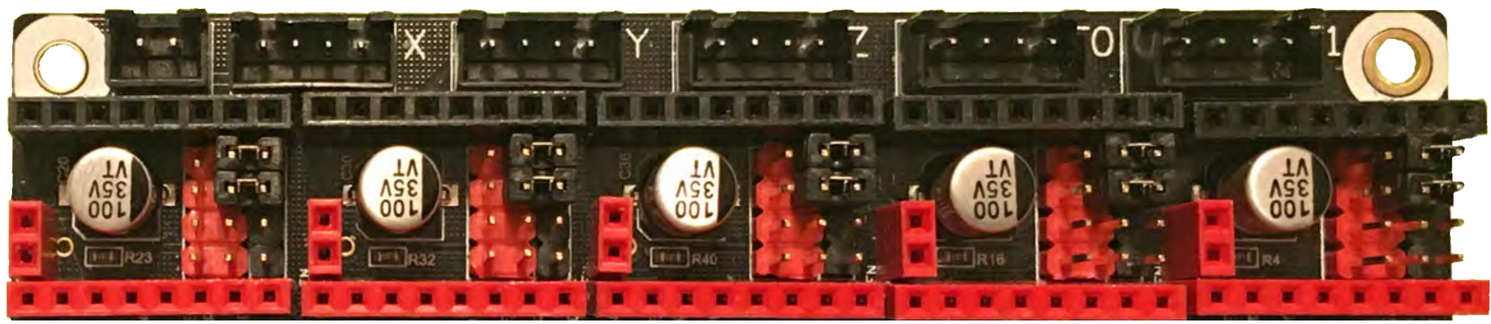




See Appendix D for legend

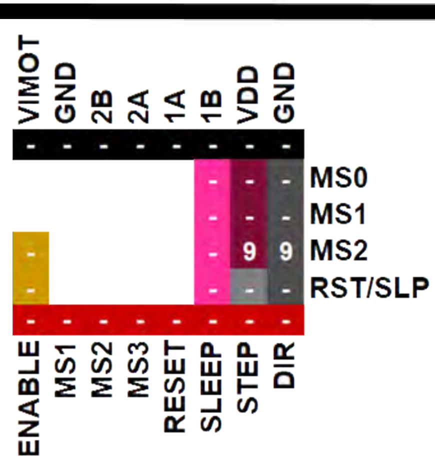
1 / 8

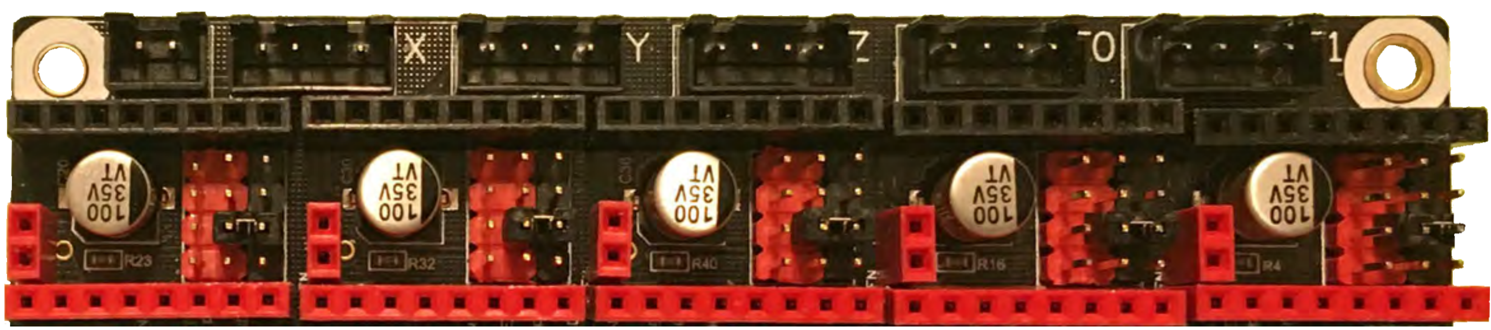




See Appendix D for legend

1 / 16



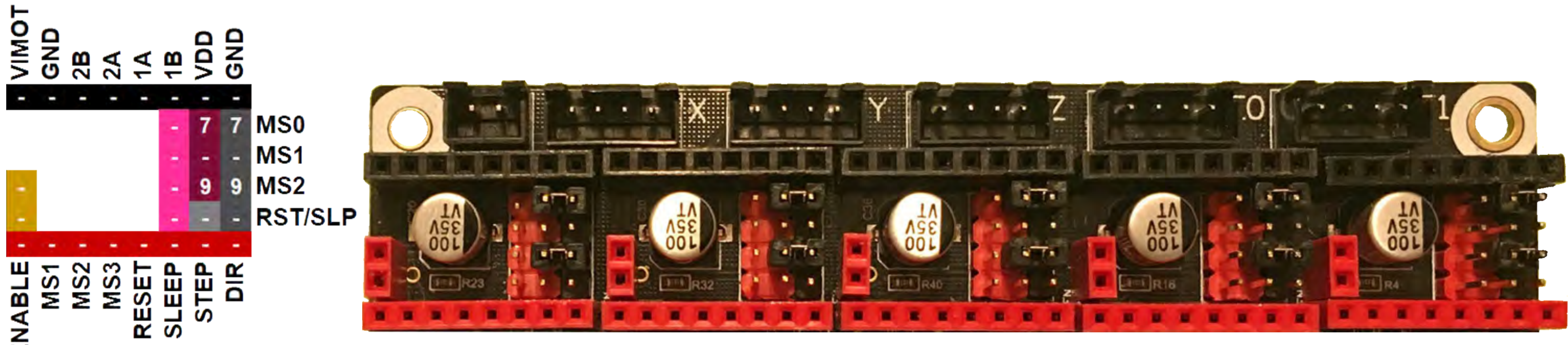


See Appendix D for legend

Stand-alone Mode

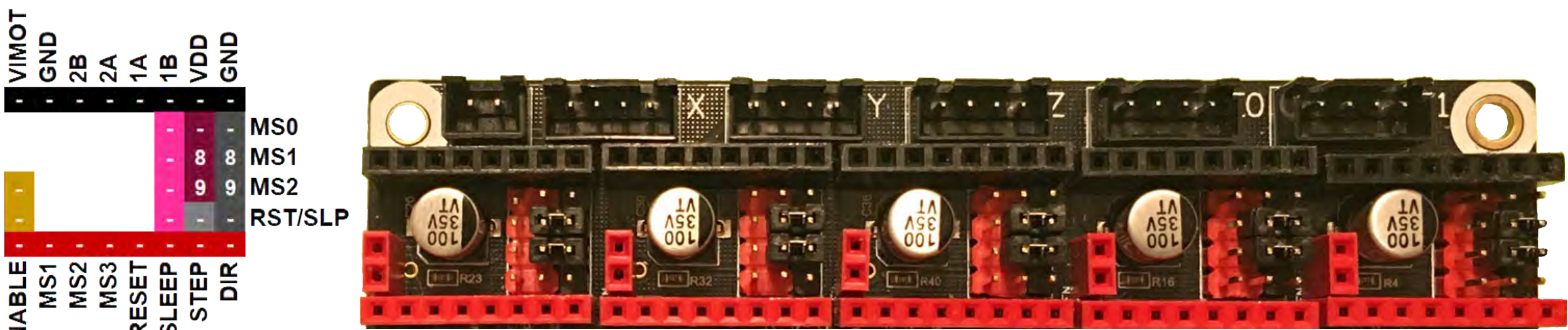
MKS LV8729

1 / 32



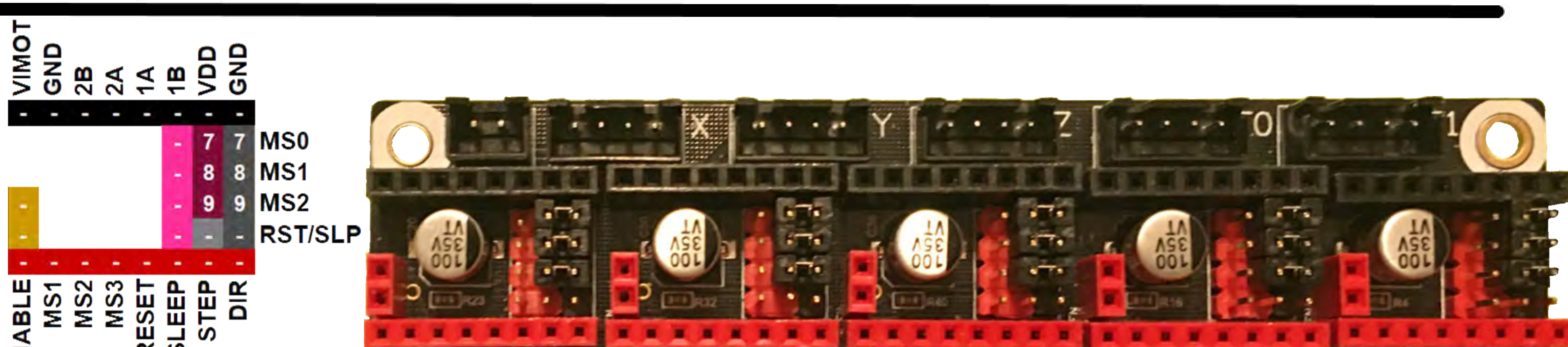
See Appendix D for legend

1 / 64



See Appendix D for legend

1 / 128

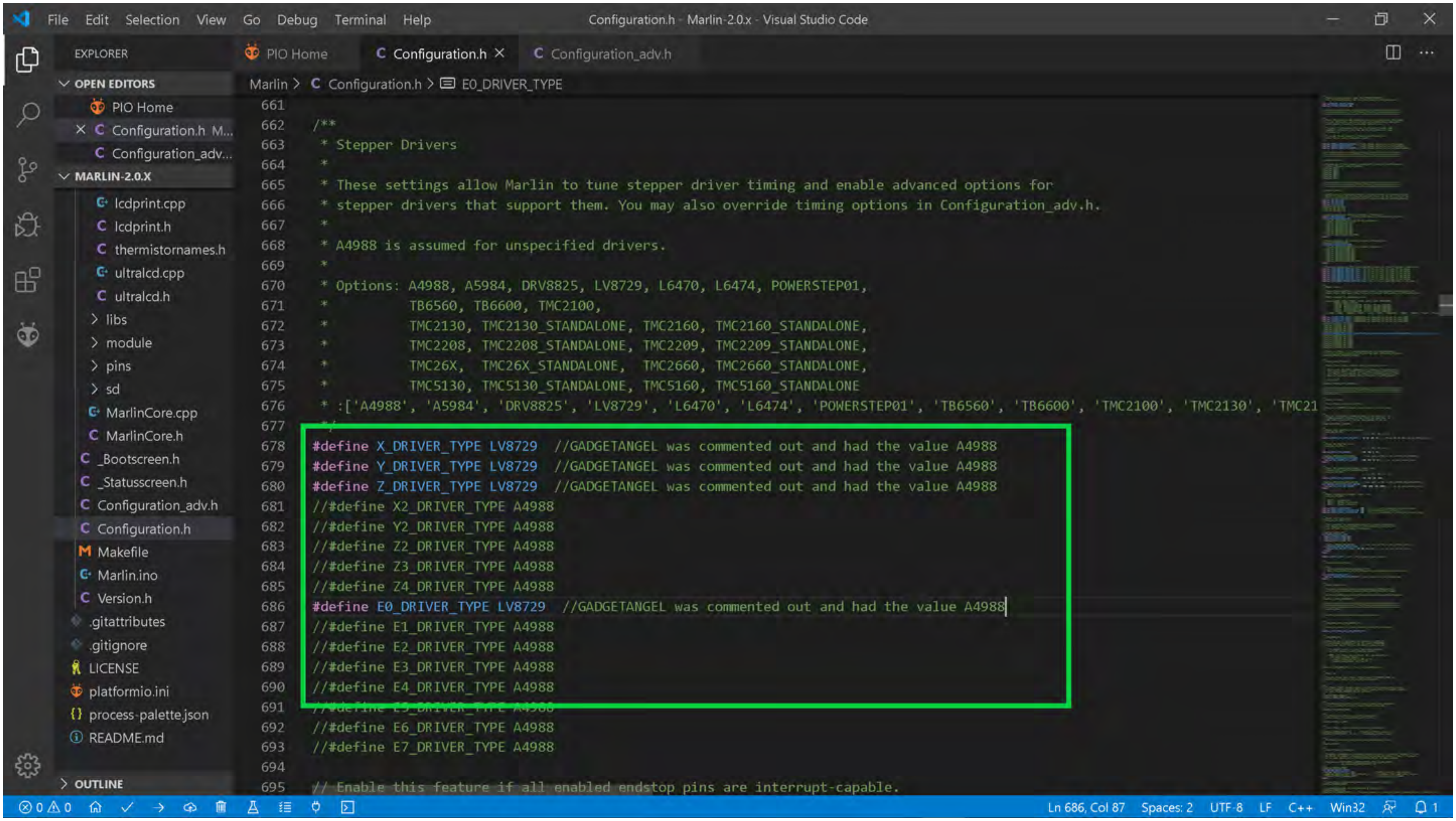


See Appendix D for legend

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

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

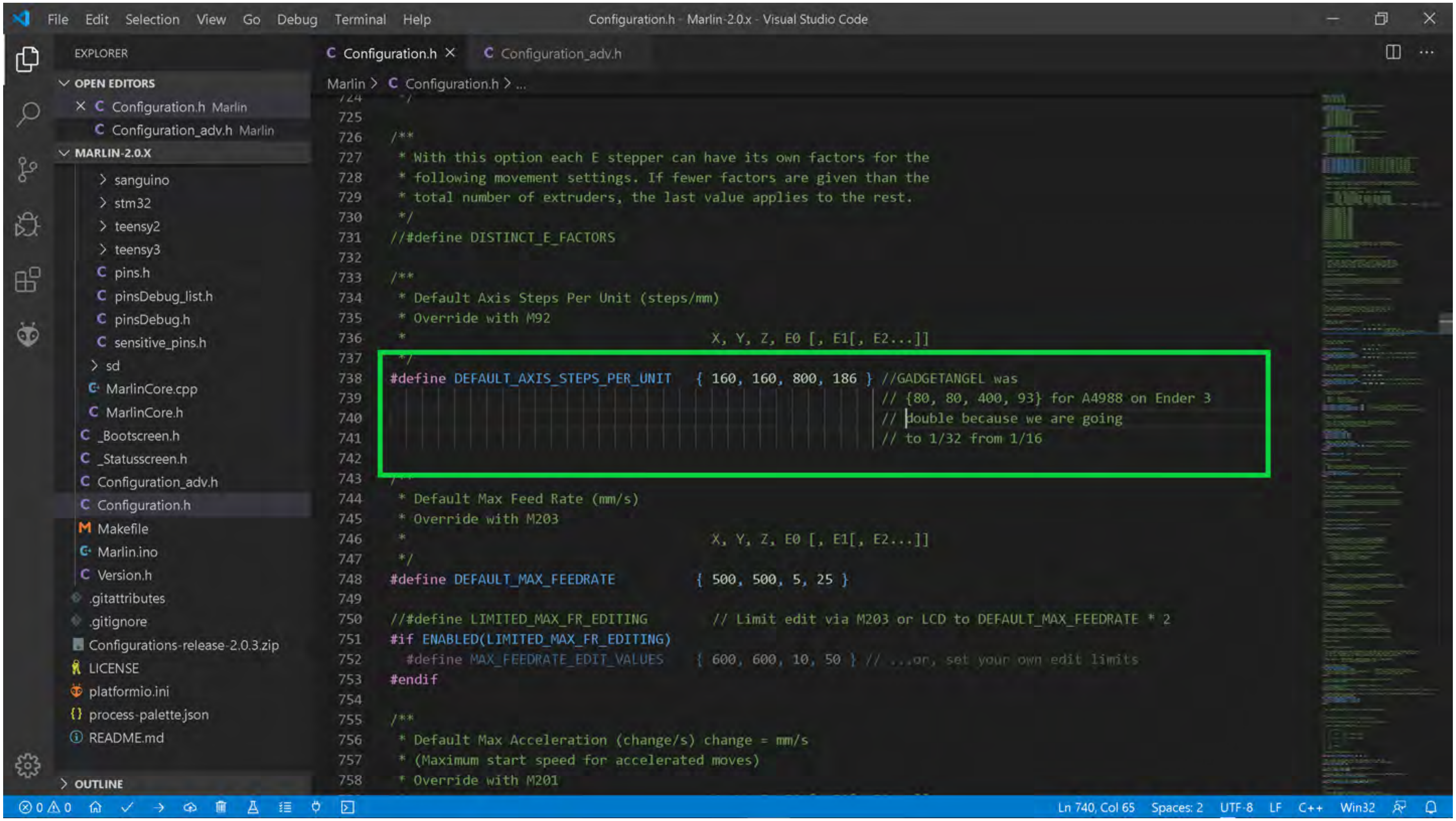
- Change the stepper motor drivers so that Marlin knows you are using MKS LV8729 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use MKS LV8729 drivers. When two "//" appear at the beginning of a line that means that line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



- Go to the next page.

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

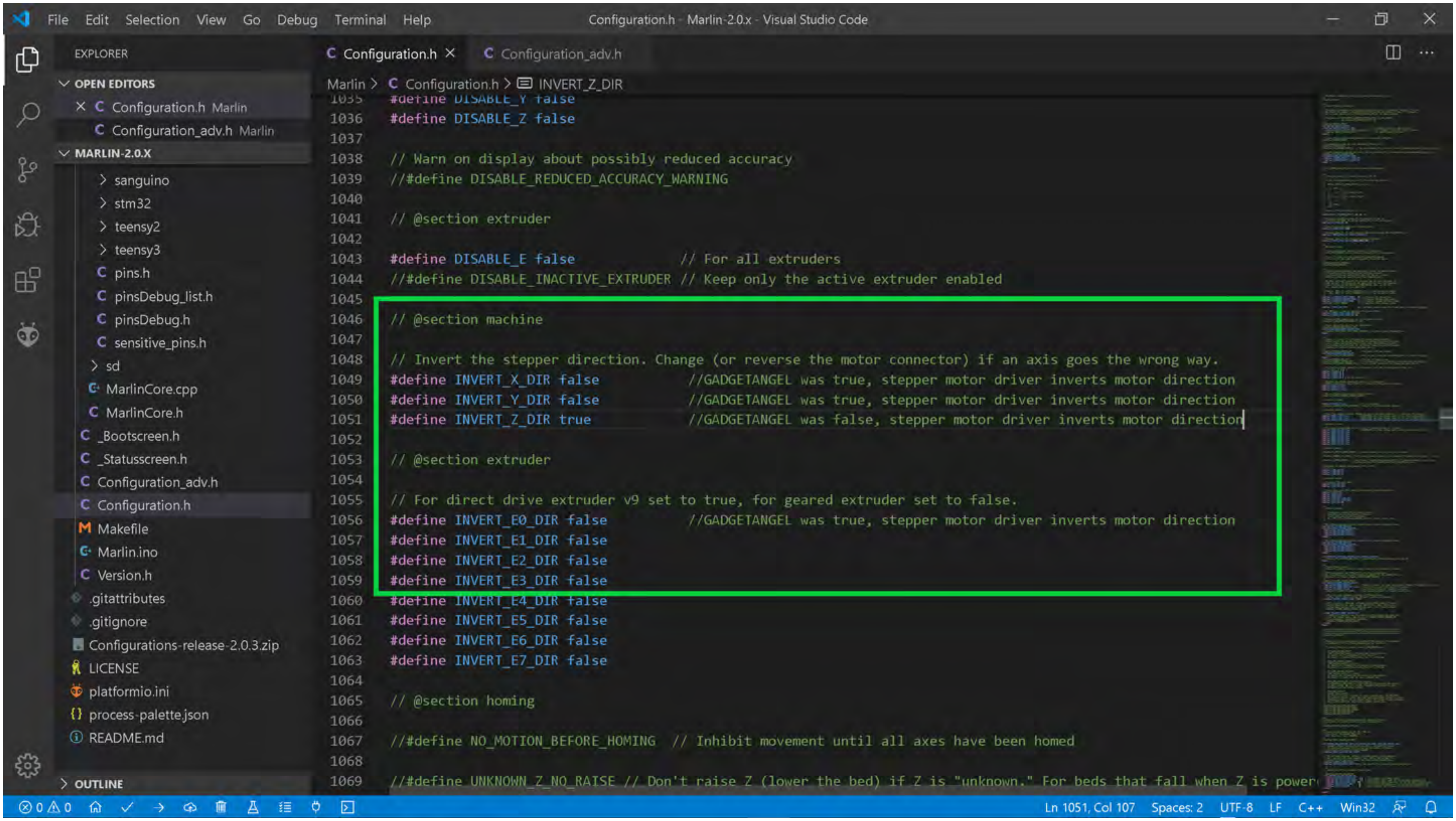
- We are changing from A4988 stepper motor drivers on the Ender 3 to LV8729 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

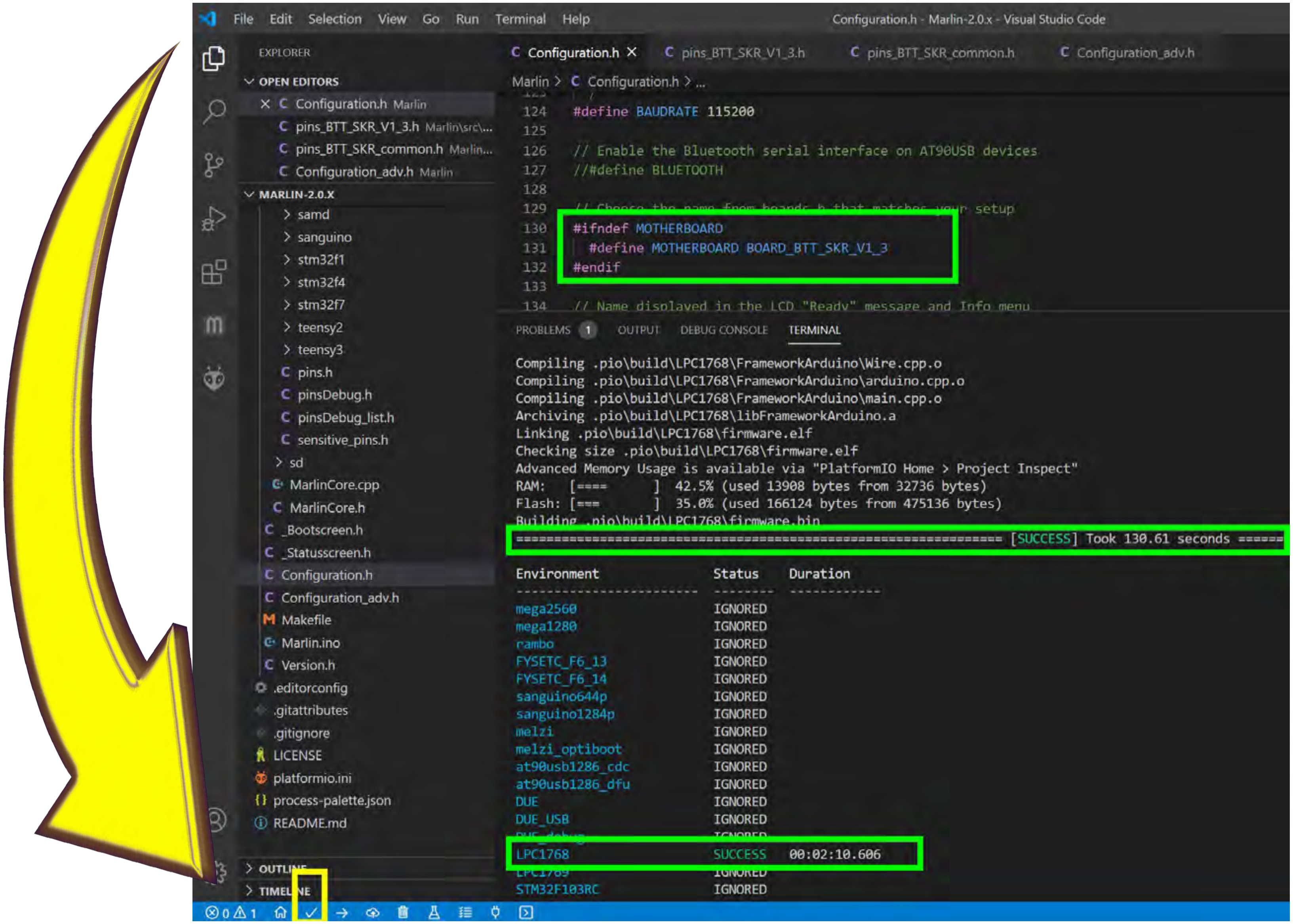
- Since the A4988 driver is what my Ender 3 used, but, now I want to use LV8729 drivers, I must invert the stepper motor direction because the LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the LV8729 driver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



- Go to the next page.

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

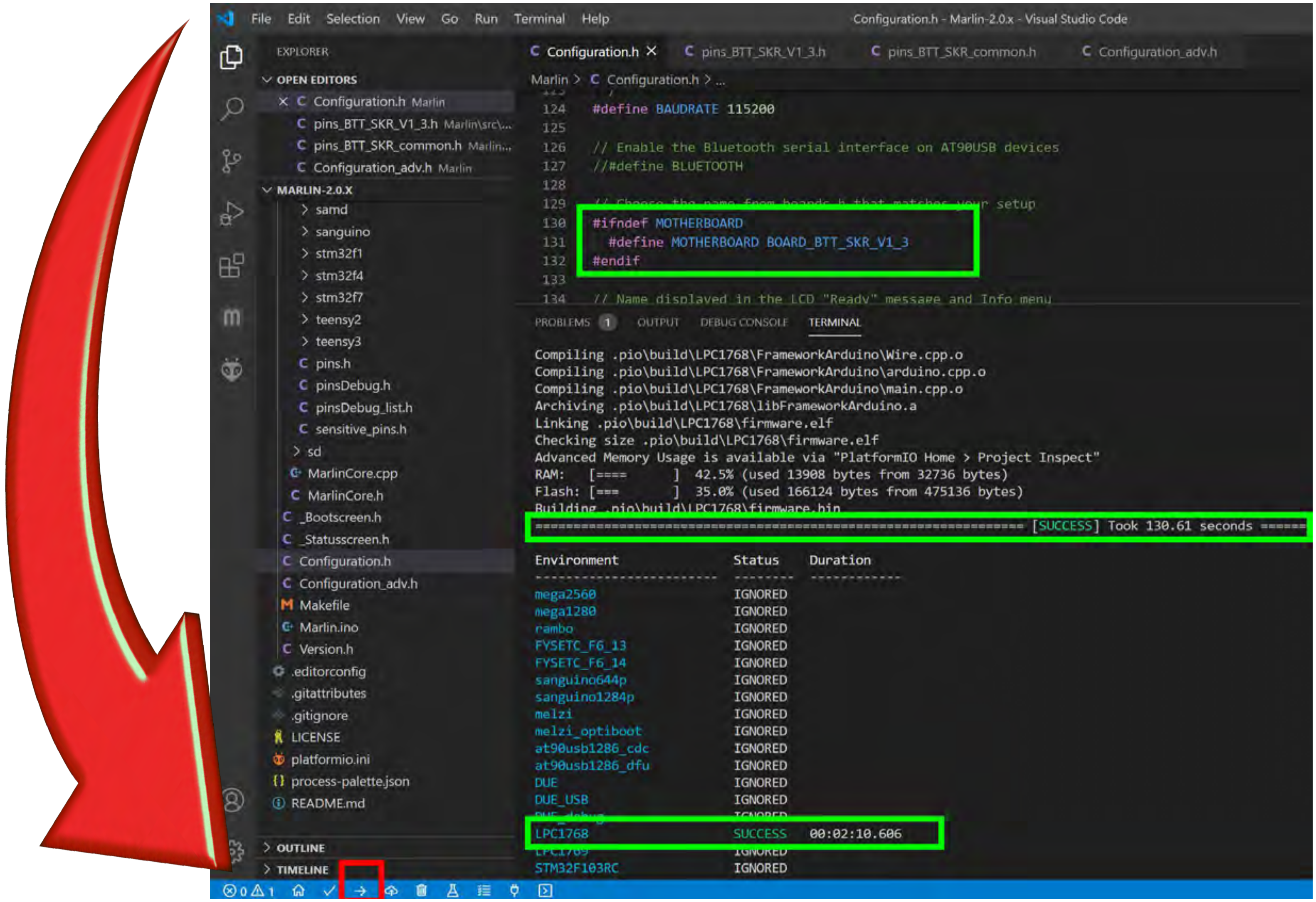
- The end of Marlin setup for MKS LV8729 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

FYSETC S6128 V1.1

base of card **Stand-alone Mode**

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

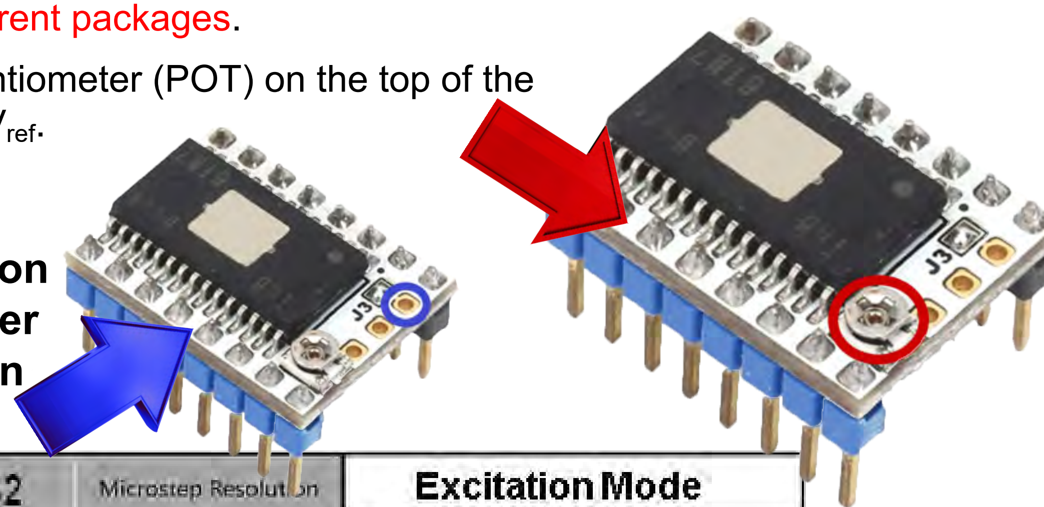
See Appendix D for legend

STATUS: Not Yet Tested

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

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

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



Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
FYSETC SureStepr SD6128 Maximum 128 Subdivision 35V DC 2.2A (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula

R_S(Typical Sense Resistor)$=0.1\Omega$

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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

FYSETC S6128 V1.1

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC X.X A (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S(Typical Sense Resistor)=X.XX Ω	I_{MAX}=FORMULA			V_{ref} =FORMULA	

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

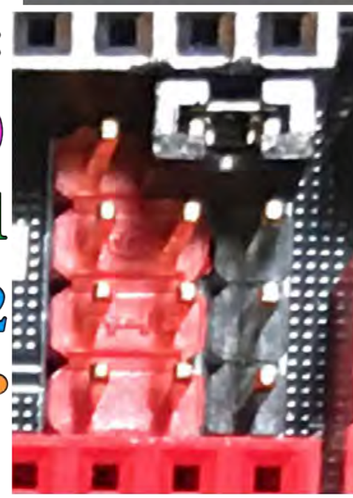
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

FYSETC S6128 V1.1

base of card **Stand-alone Mode**

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
64	0	1	1
128	1	1	1

fans/heaters/usb side

See Appendix D for legend

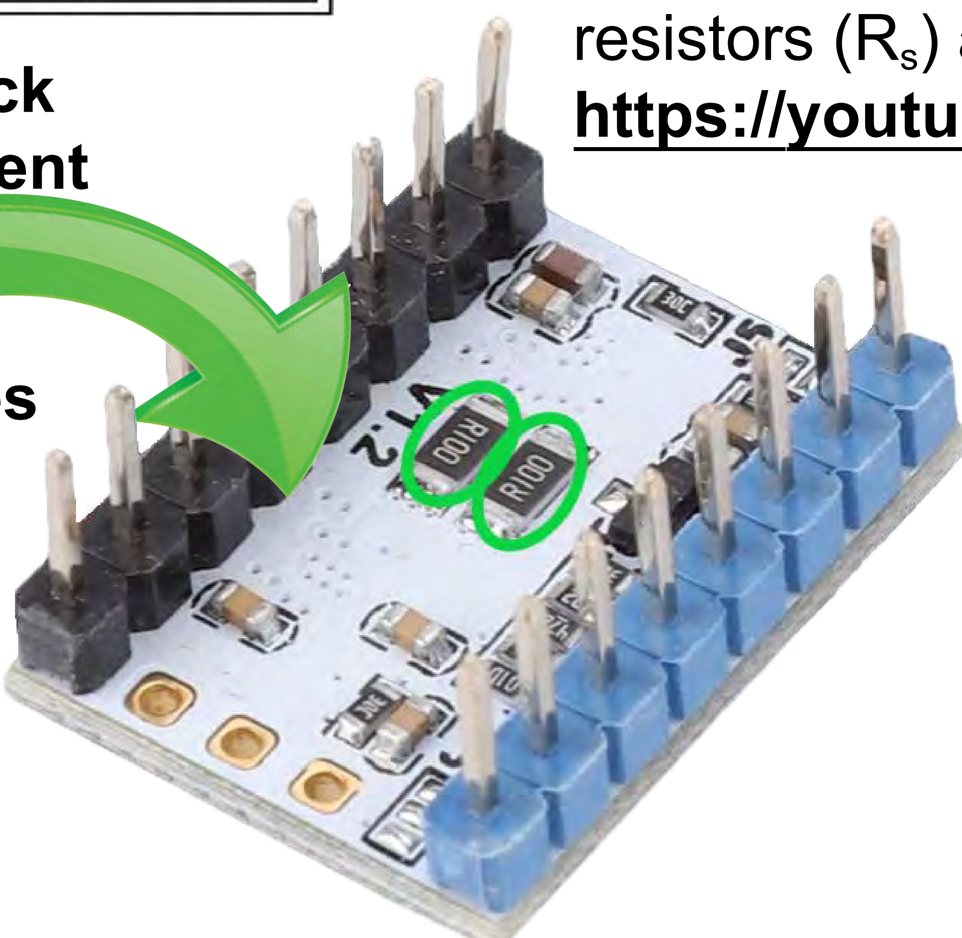
STATUS: Not Yet Tested

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

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

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

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



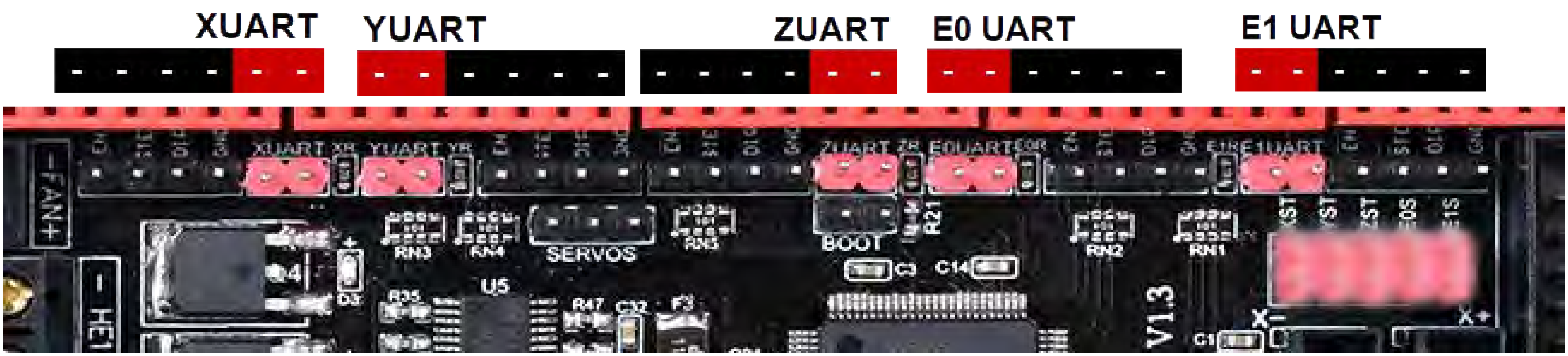
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

FYSETC S6128 V1.1

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

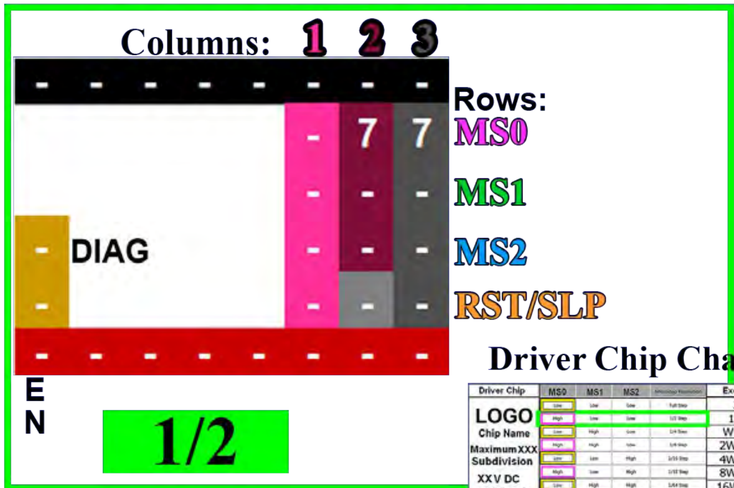


Stand-alone Mode

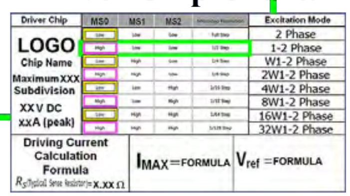
FYSETC S6128 V1.1

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

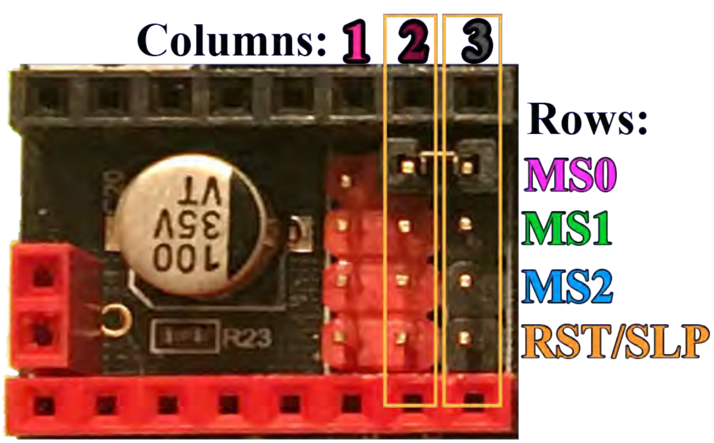
Driver Socket Representation



Driver Chip Chart:



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Meaning:

Driver Chip Chart:

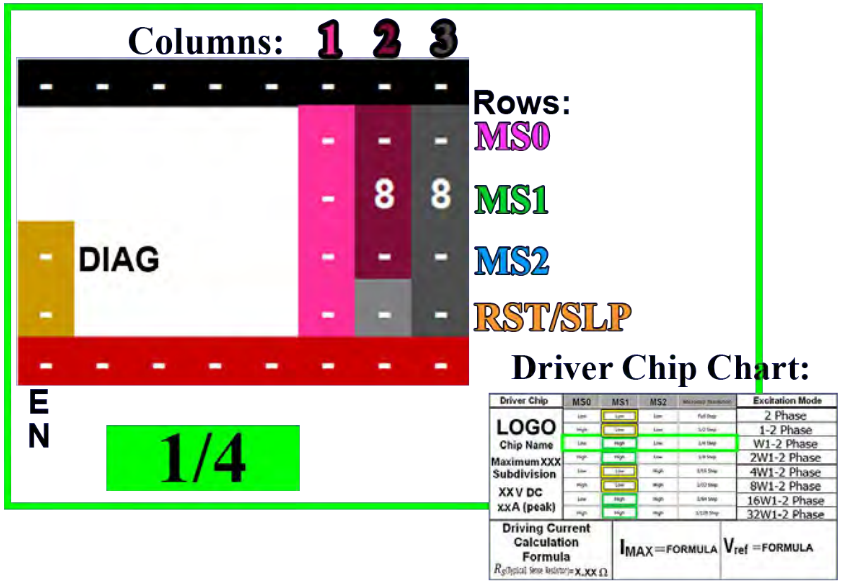
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

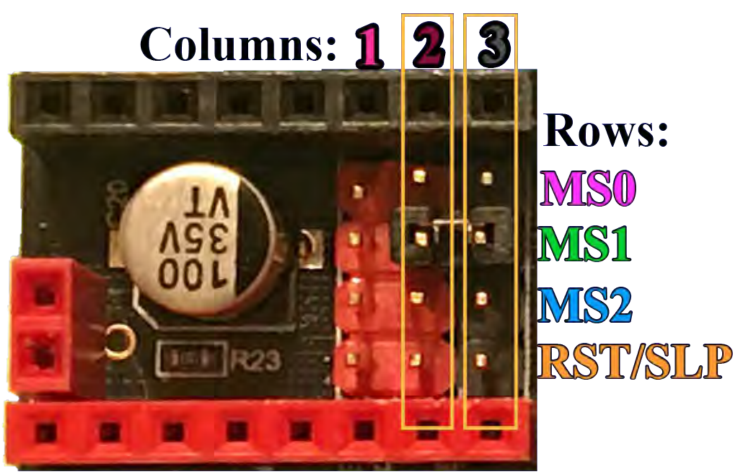
Stand-alone Mode

FYSETC S6128 V1.1

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

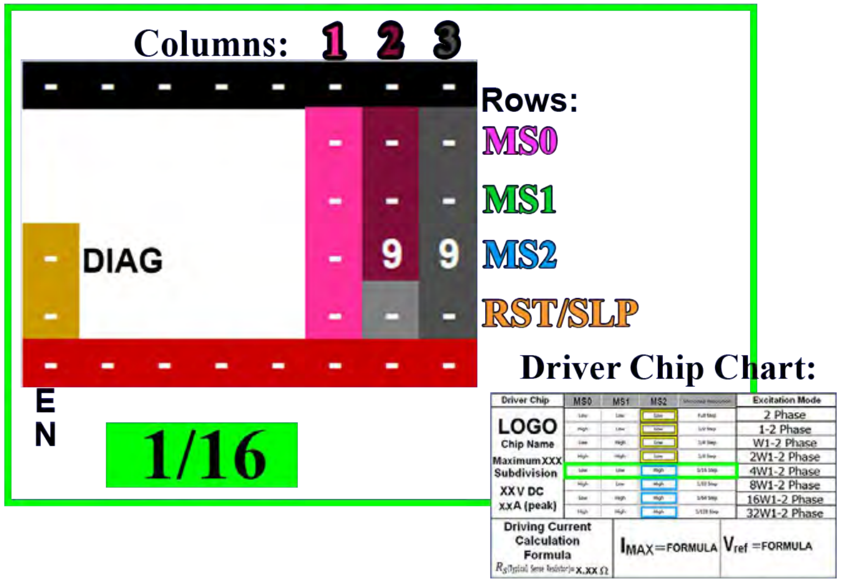
Meaning:

Driver Chip Chart:	
High	set Jumper between column 2 and column 3 on the MS1 row
Low	No Jumper Set

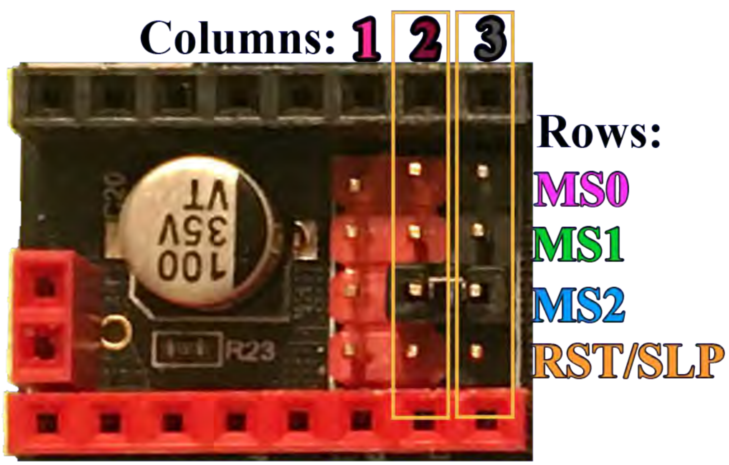
Stand-alone Mode

FYSETC S6128 V1.1

Driver Socket Representation



High:



MS2 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
9 9	→	High

Meaning:

Driver Chip Chart:

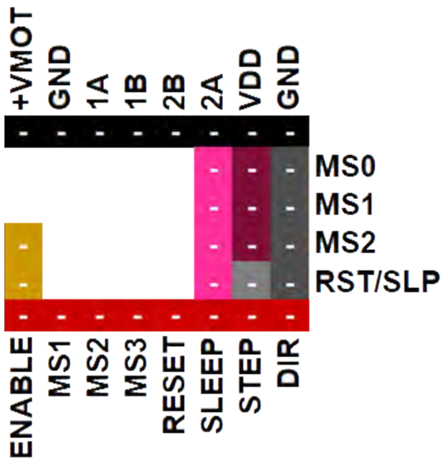
- **High** → **set Jumper between column 2 and column 3 on the MS2 row**
- **Low** → **No Jumper Set**

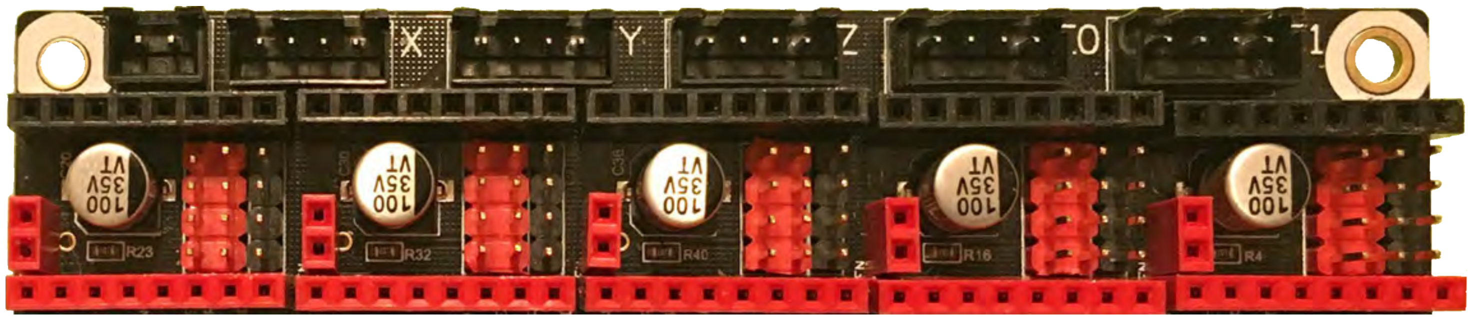
Stand-alone Mode

FYSETC S6128 V1.1

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

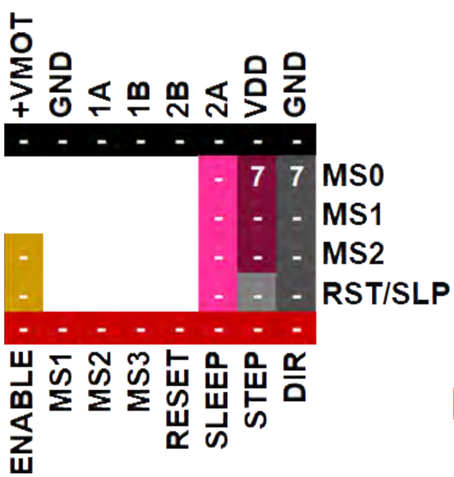
STEP

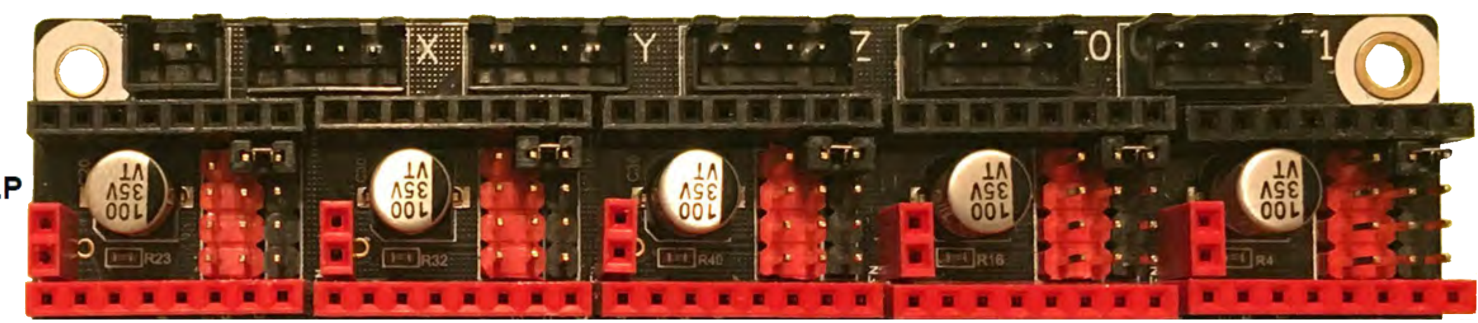




See Appendix D for legend

1 / 2





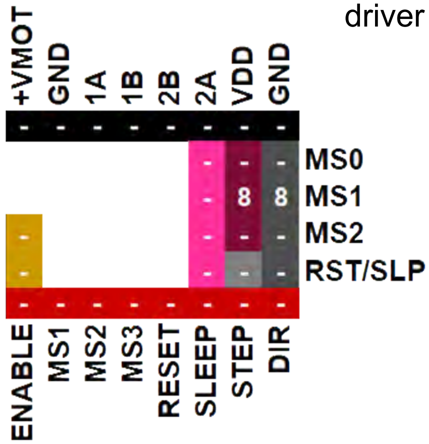
See Appendix D for legend

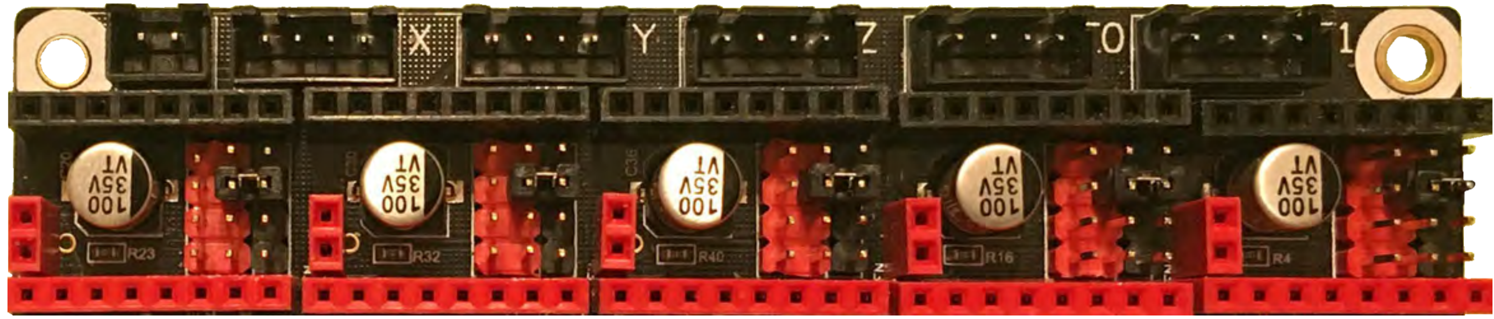
Stand-alone Mode

FYSETC S6128 V1.1

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

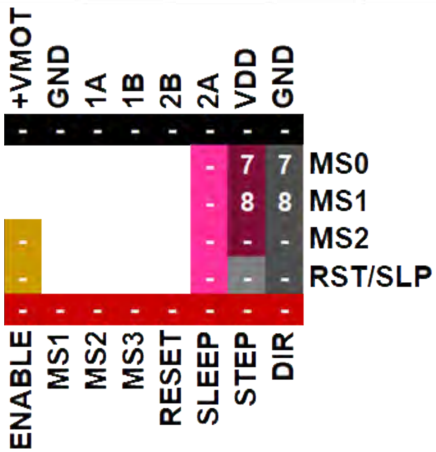
1 / 4

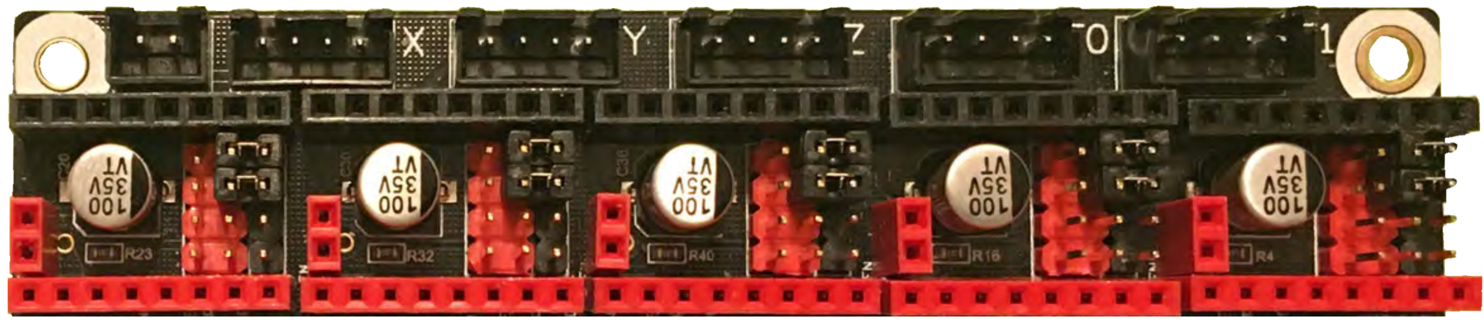




See Appendix D for legend

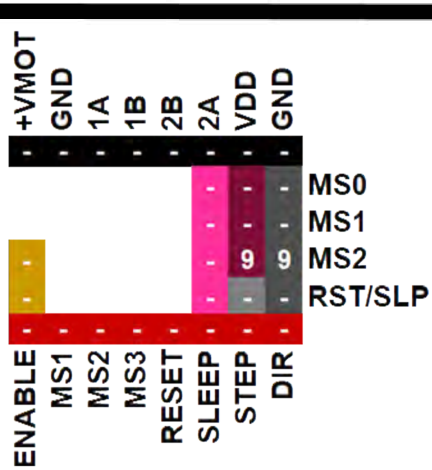
1 / 8

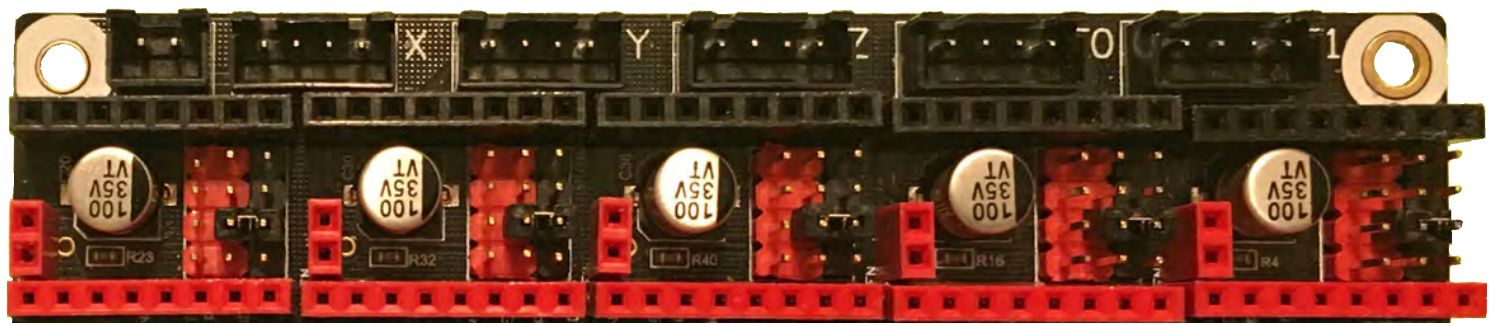




See Appendix D for legend

1 / 16





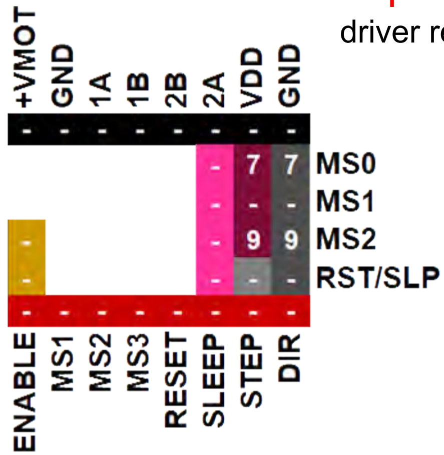
See Appendix D for legend

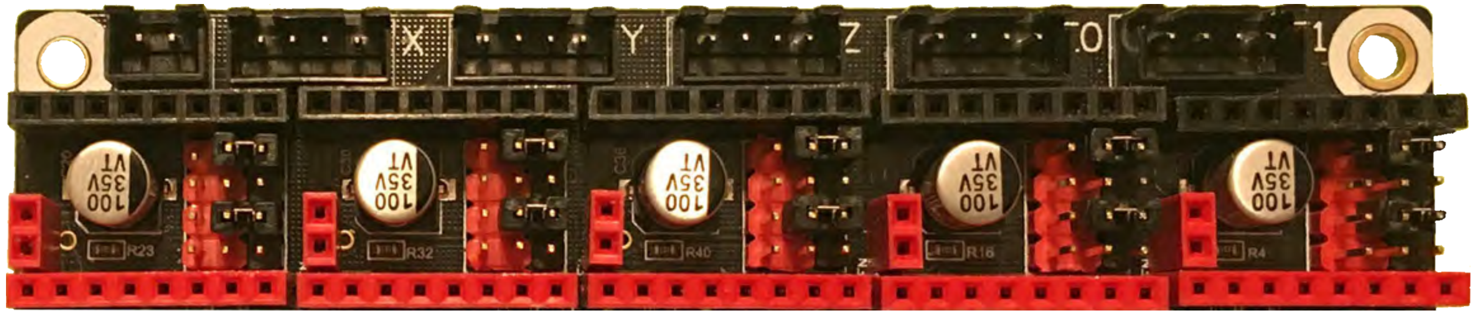
Stand-alone Mode

FYSETC S6128 V1.1

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

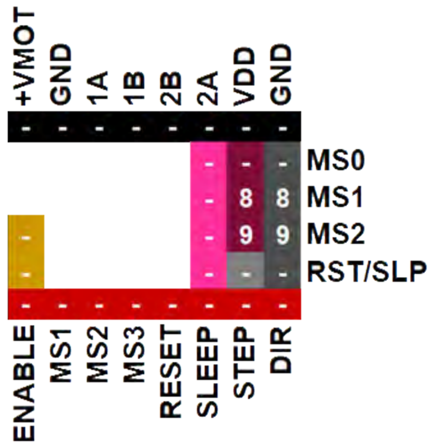
1 / 32

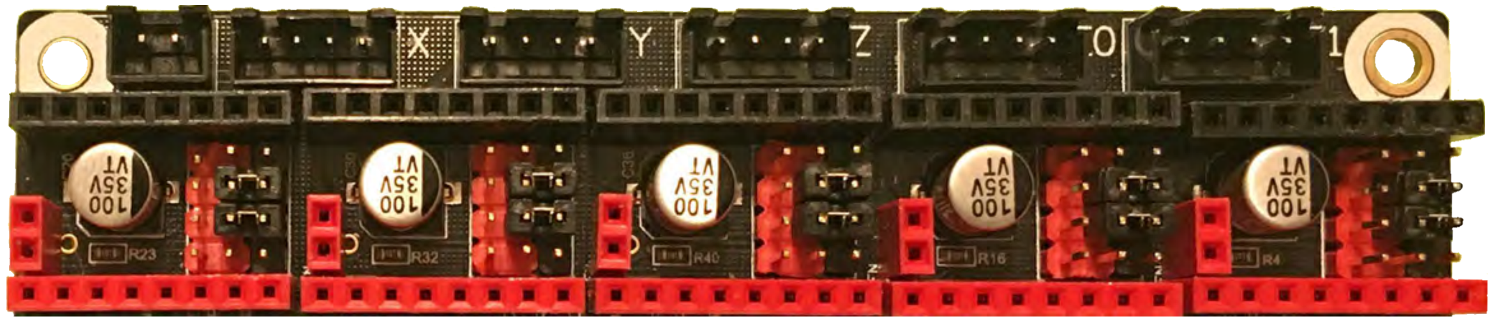




See Appendix D for legend

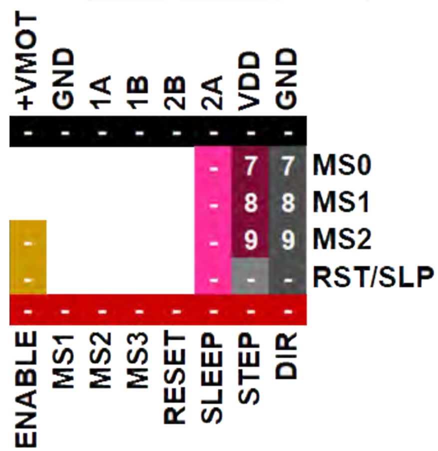
1 / 64

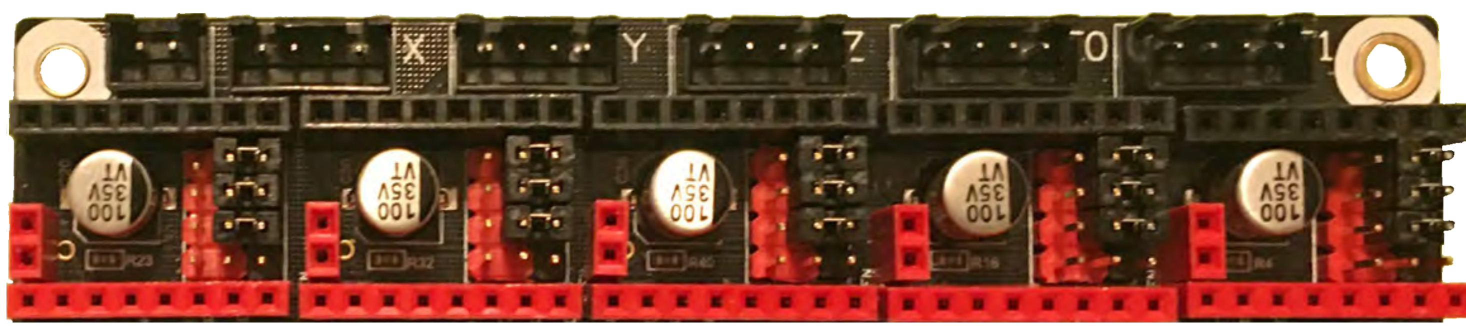




See Appendix D for legend

1 / 128



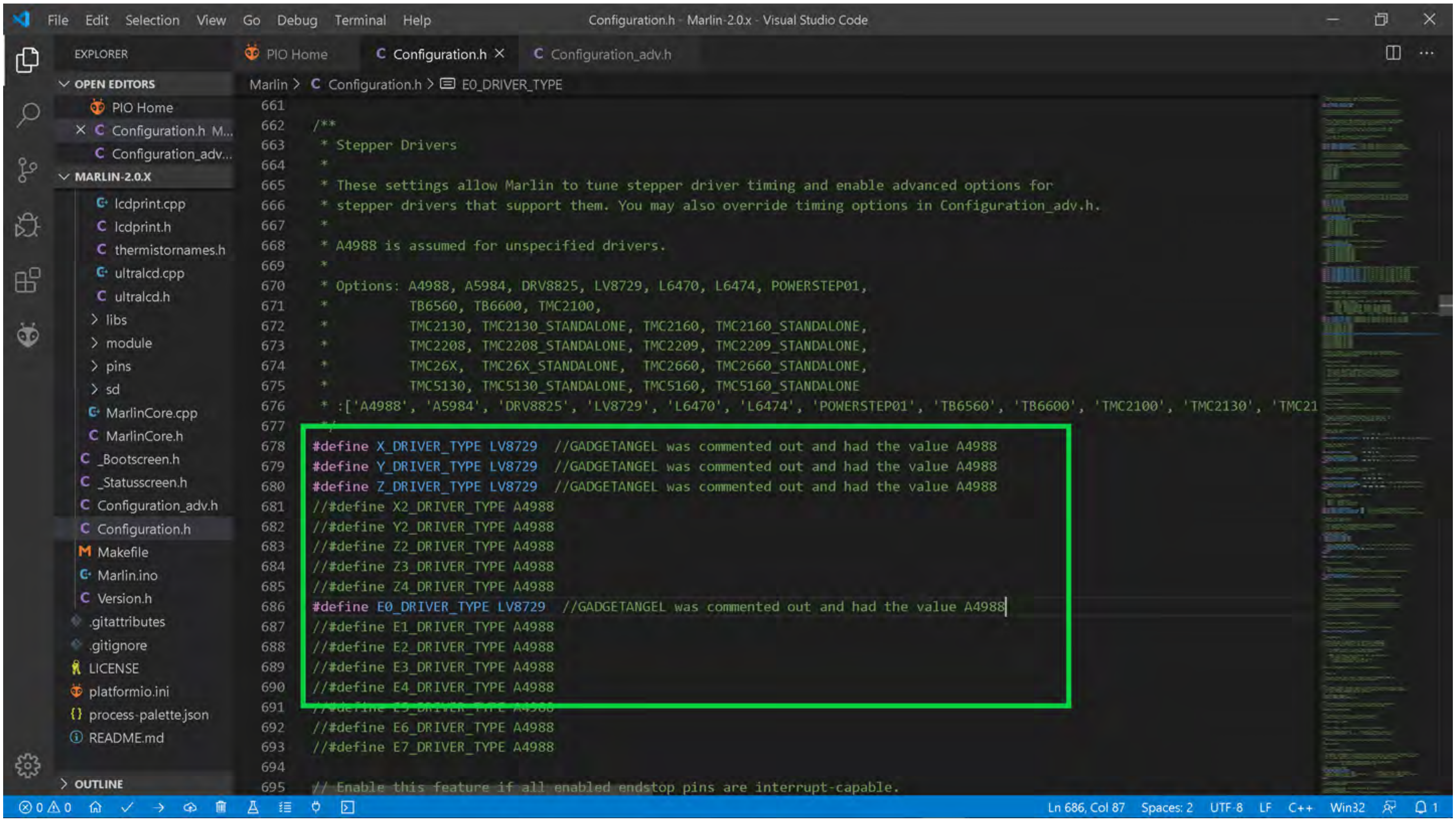


See Appendix D for legend

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

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

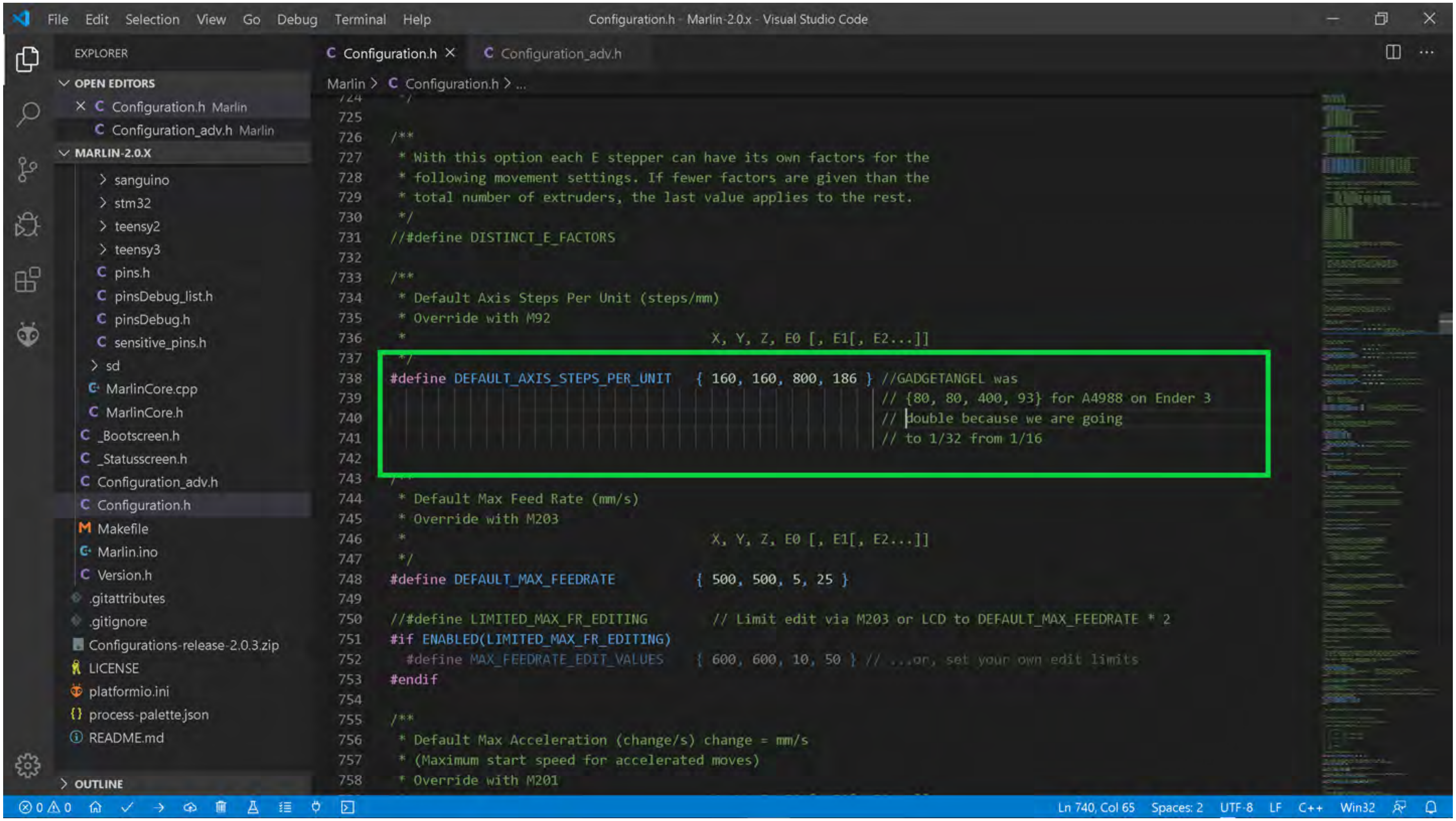
- Change the stepper motor drivers so that Marlin knows you are using FYSETC S6128 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC S6128 drivers. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The S6128 is a drop in replacement for the LV8729. Since Marlin does not have an option for S6128 we will use the LV8729 option.



- Go to the next page.

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

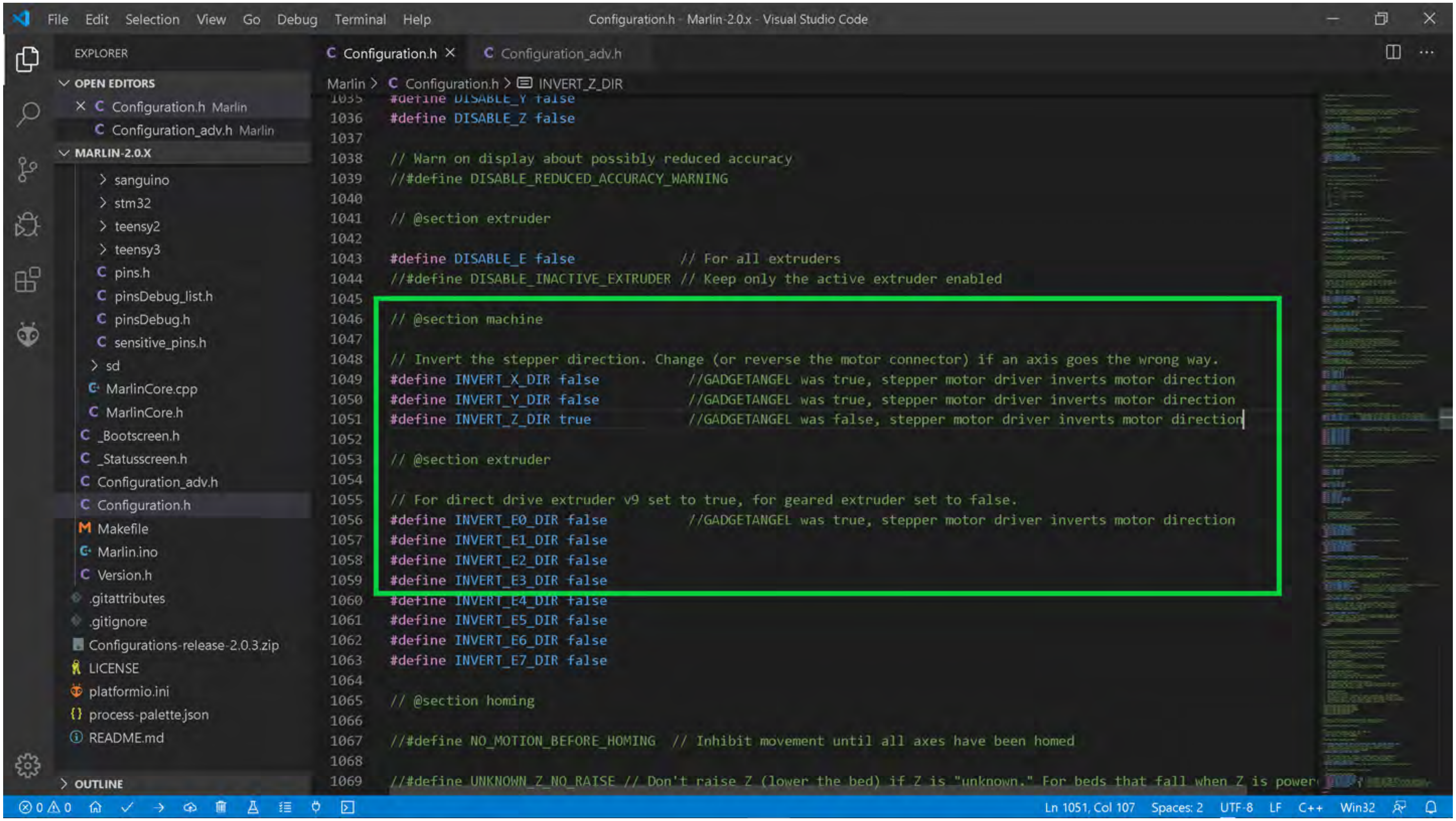
- We are changing from A4988 stepper motor drivers on the Ender 3 to S6128 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

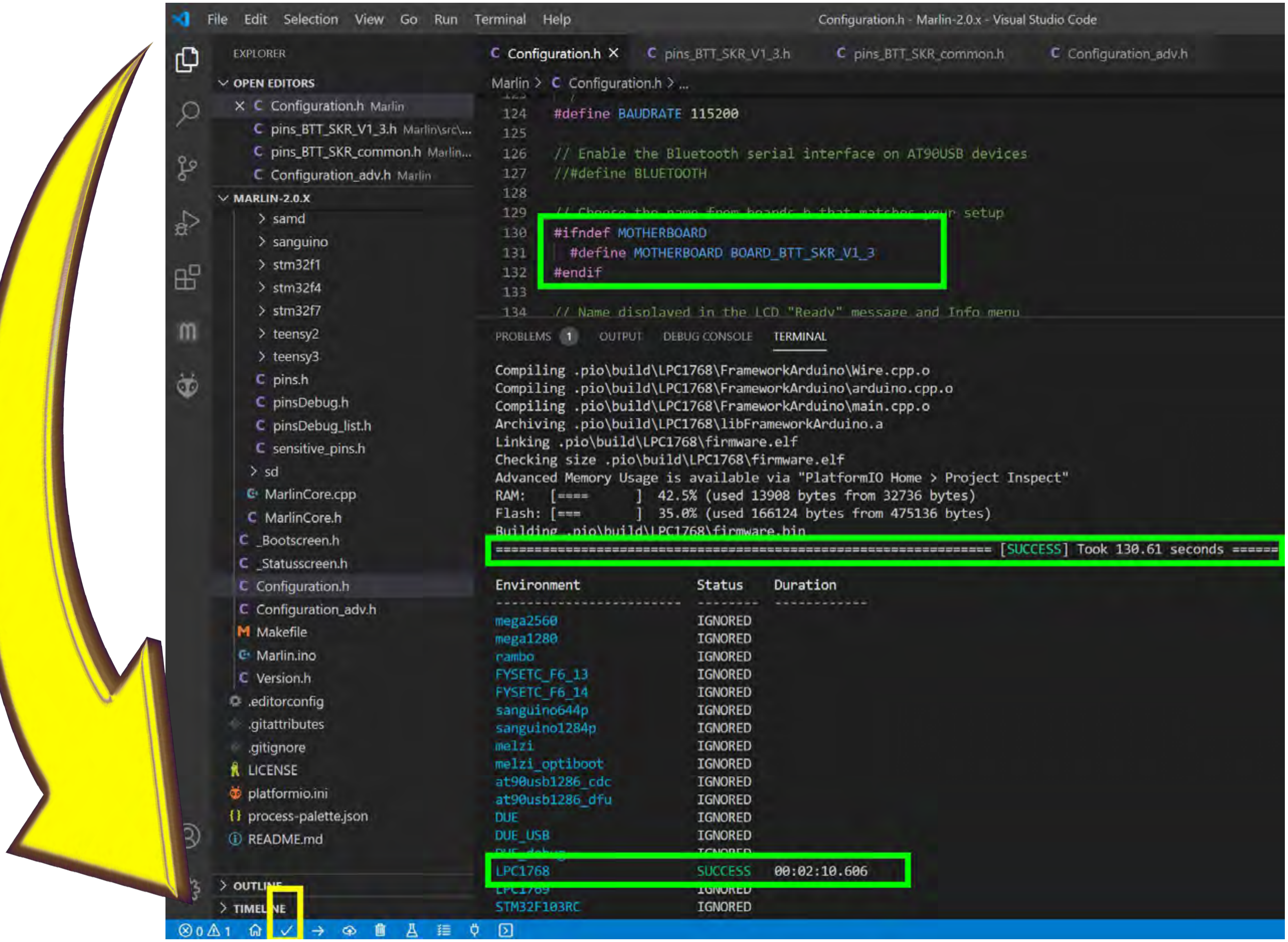
- Since the A4988 driver is what my Ender 3 used, but, now I want to use S6128 drivers, I must invert the stepper motor direction because the S6128 or LV8729 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the S6128 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as show in the **GREEN** box below



- Go to the next page.

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

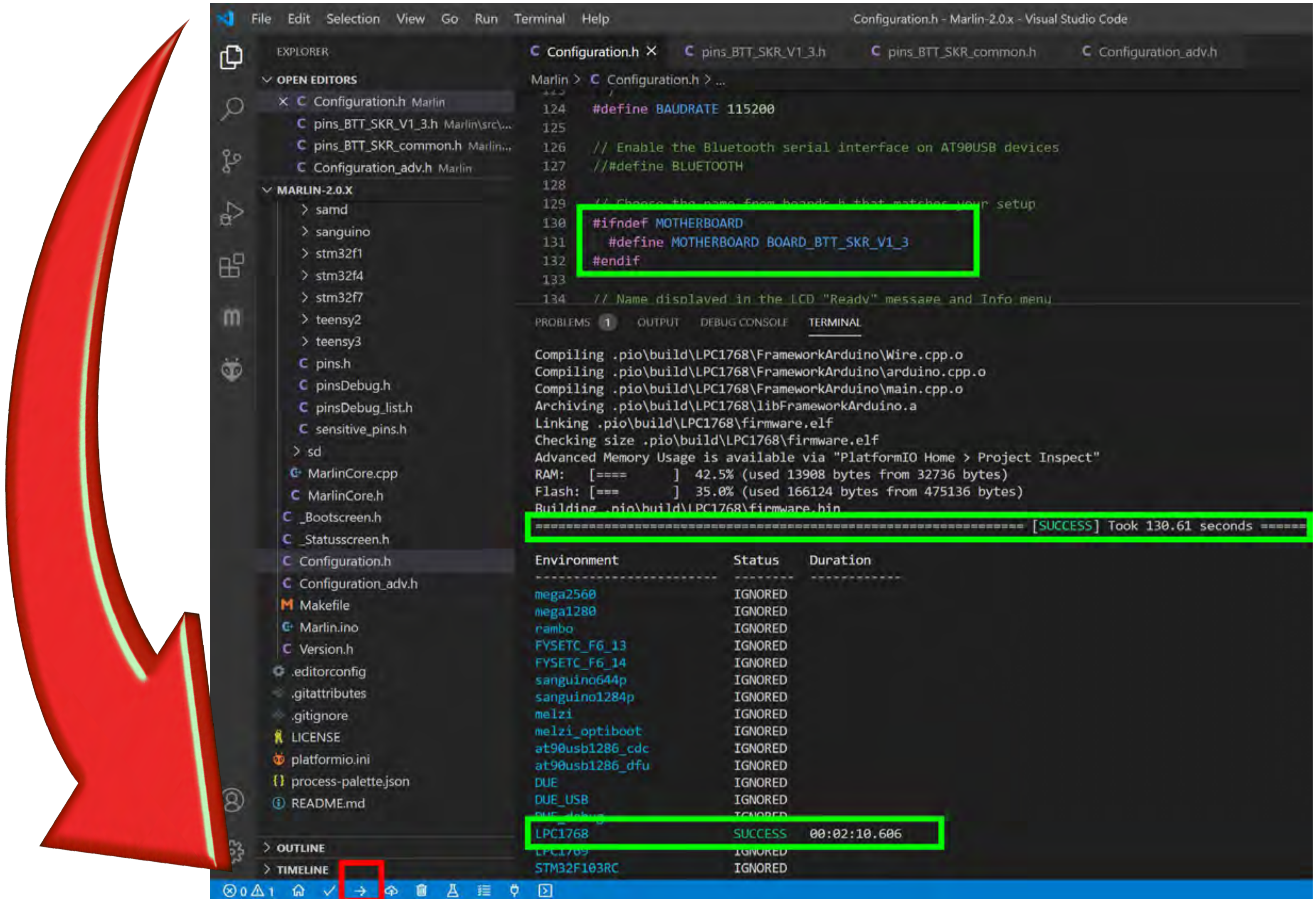
- The end of Marlin setup for FYSETC S6128 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

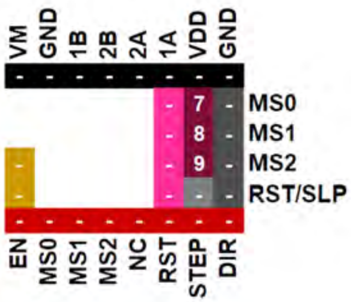


- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

FYSETC ST820

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
FYSETC ST820	Low	Low	Low	Full step	2 Phase
Maximum 256 Subdivision	High	Low	Low	Half step	1-2 Phase
45V DC	Low	High	Low	1/4 step	W1-2 Phase
1.5A (peak)	High	High	Low	1/8 step	2W1-2 Phase
	Low	Low	High	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	Low	High	High	1/128 step	16W1-2 Phase
	High	High	High	1/256 step	32W1-2 Phase

Driving Current Calculation Formula

V_{DD} = 3.3 V or 5 V DC

R_S (Typical Sense Resistor) = 0.15Ω

$$I_{MAX} = V_{ref} * \left(\frac{V_{DD}}{5}\right) * \frac{1}{R_S}$$

$$V_{ref} = I_{MAX} * \left(\frac{5}{V_{DD}}\right) * R_S$$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

FYSETC ST820

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Low → No Jumper

Rows:

MS0
MS1
MS2
RST/SLP

High → Jumper Set

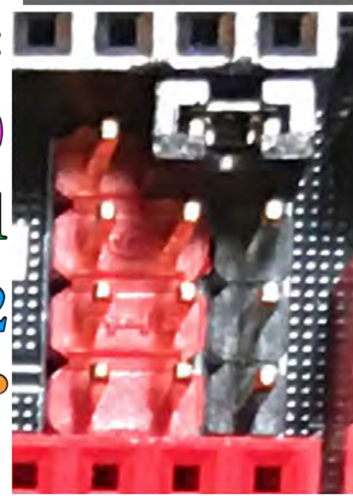
Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
XXA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Rows:

MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

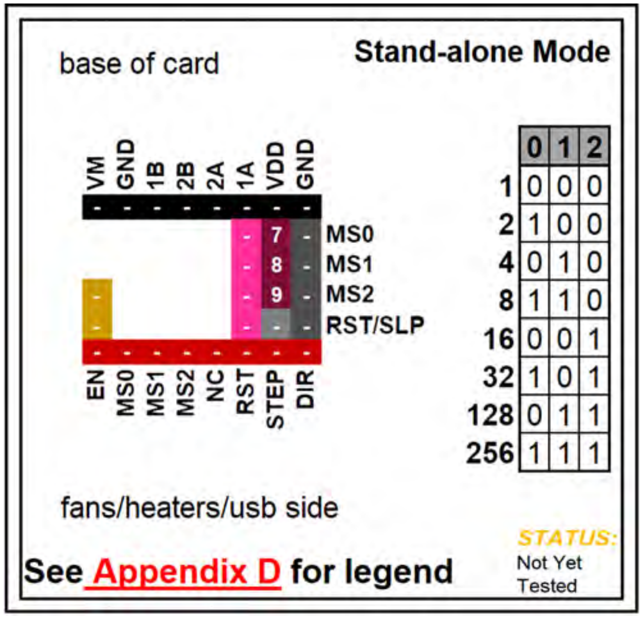
MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

FYSETC ST820



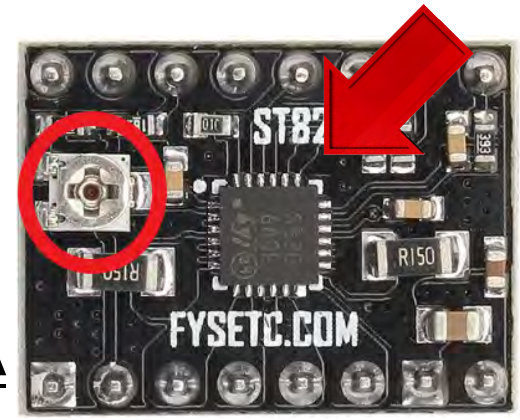
	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

See Appendix D for legend

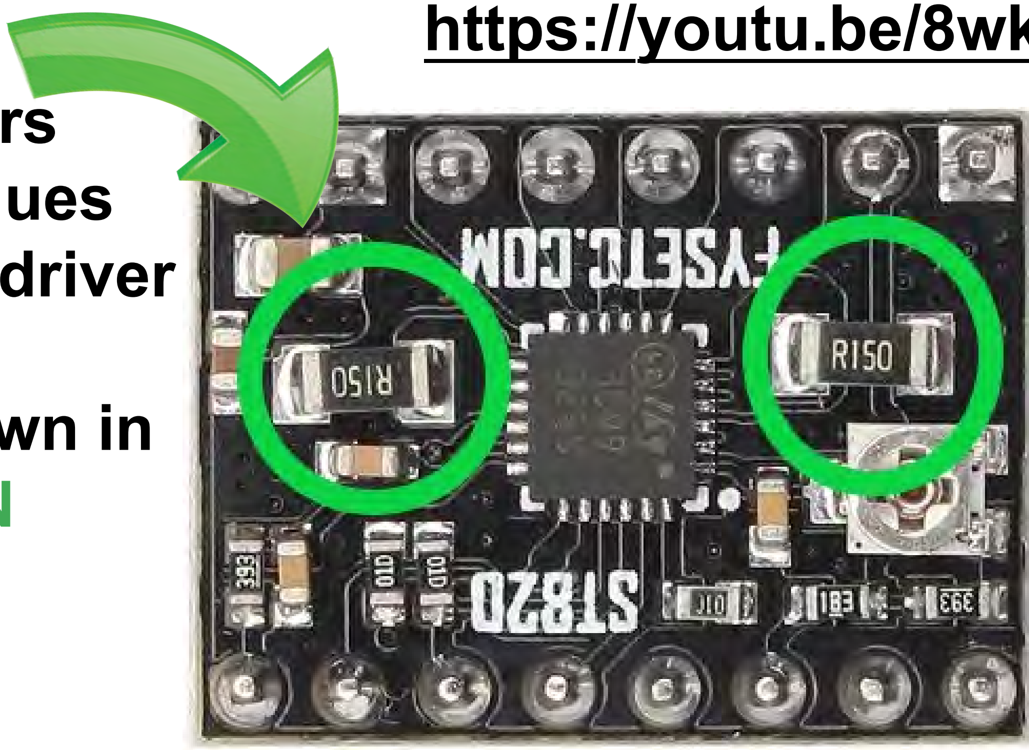
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

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



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



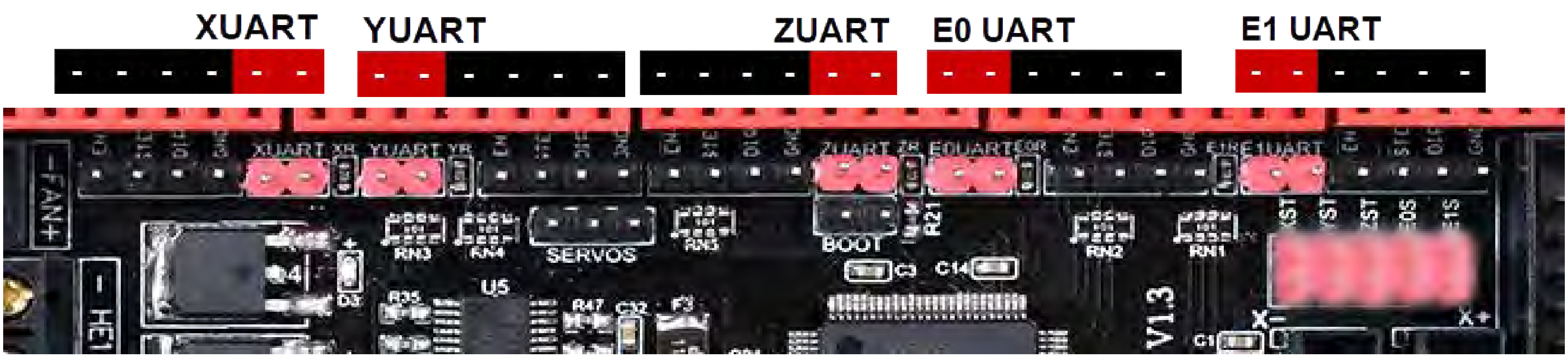
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

FYSETC ST820

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

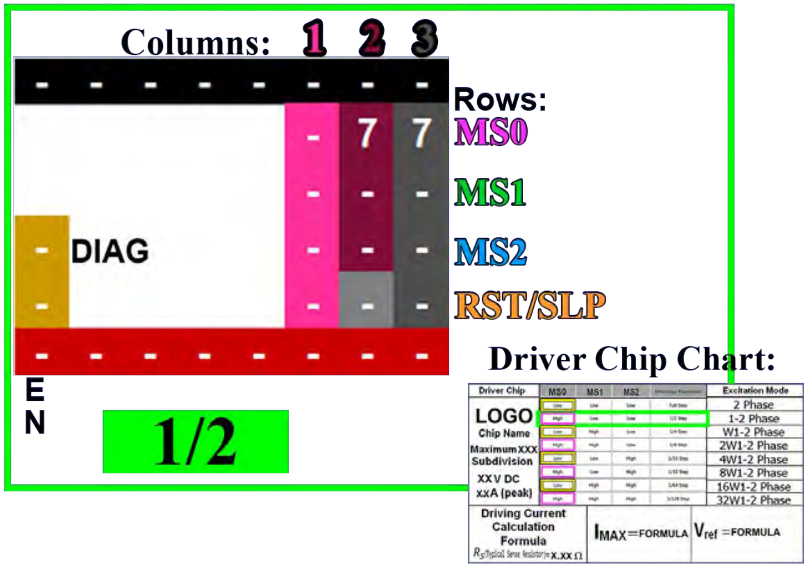


Stand-alone Mode

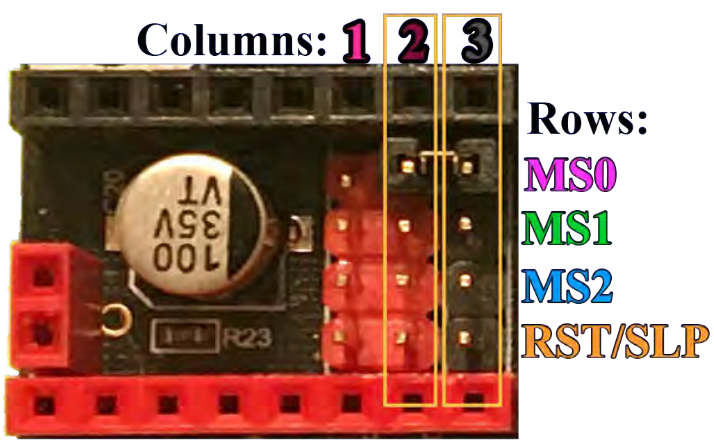
FYSETC ST820

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:

Meaning:

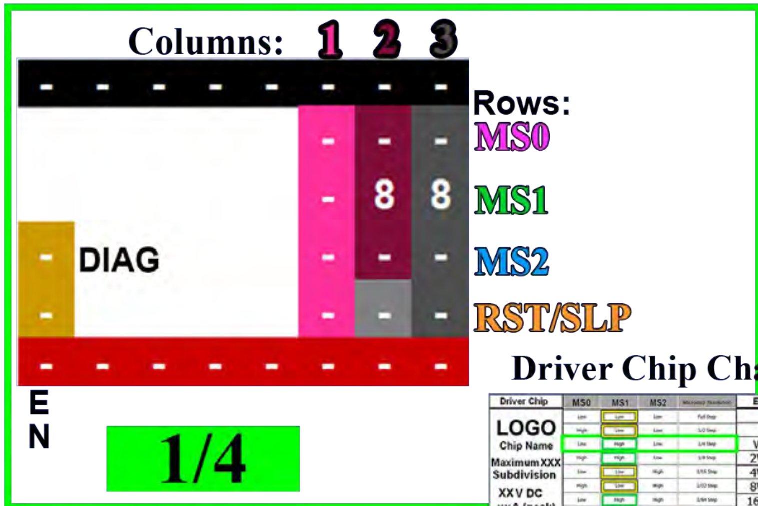
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

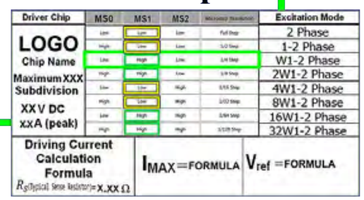
Stand-alone Mode

FYSETC ST820

Driver Socket Representation



Driver Chip Chart:



High:

MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

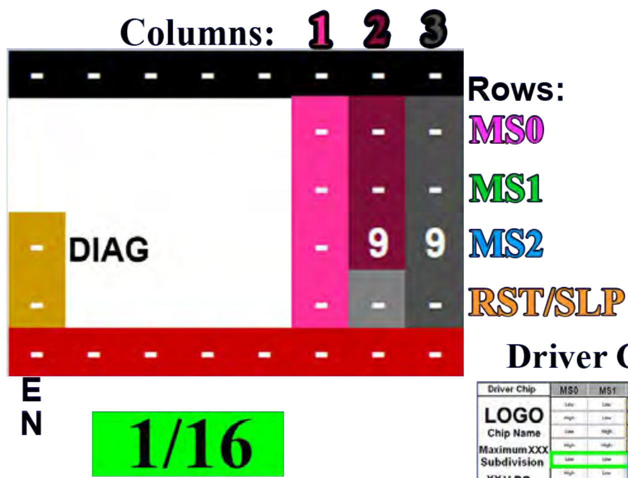
Meaning:

Driver Chip Chart:	
High	set Jumper between column 2 and column 3 on the MS1 row
Low	No Jumper Set

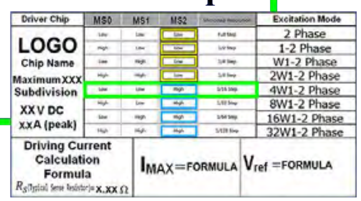
Stand-alone Mode

FYSETC ST820

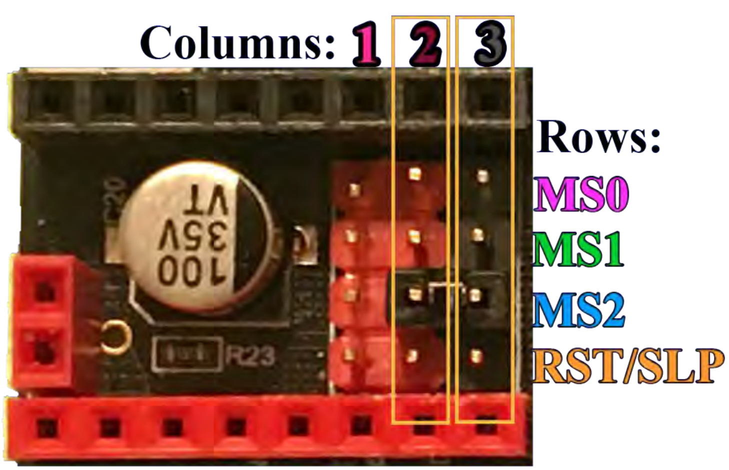
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 9 9 → Driver Chip Chart: **High**

Meaning:

Driver Chip Chart:

High → **set Jumper between column 2 and column 3 on the MS2 row**

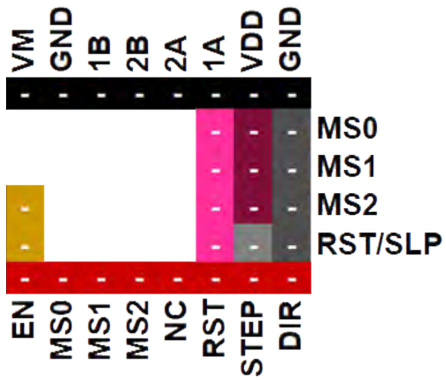
Low → **No Jumper Set**

Stand-alone Mode

FYSETC ST820

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

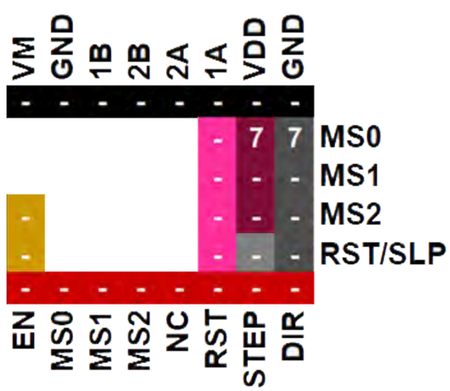
STEP

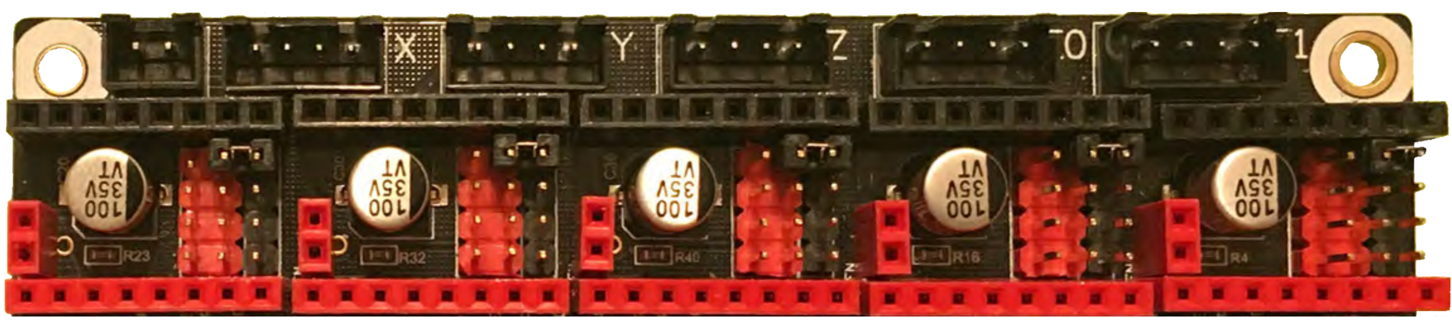


See Appendix D for legend

---

1 / 2





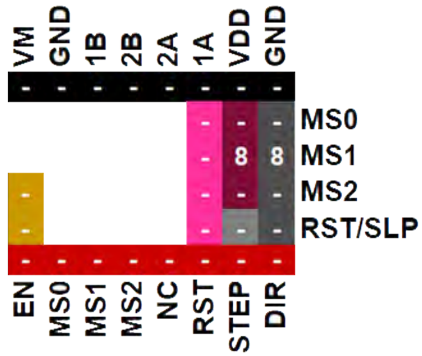
See Appendix D for legend

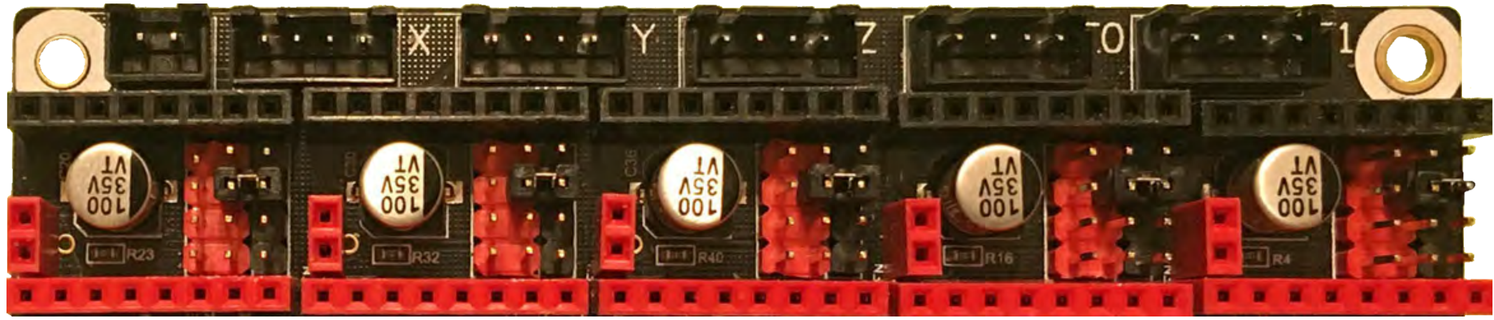
Stand-alone Mode

FYSETC ST820

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

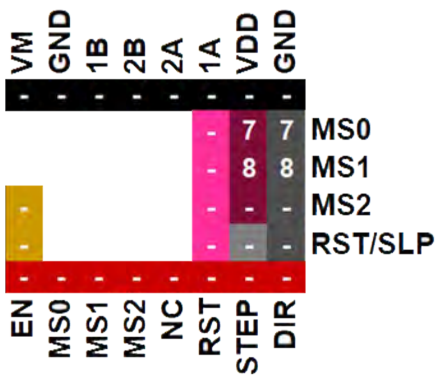
1 / 4

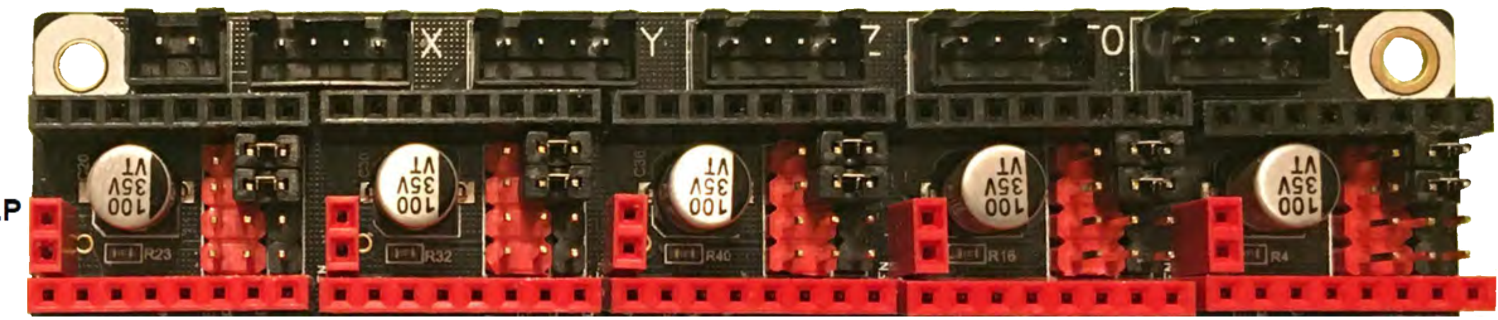




See Appendix D for legend

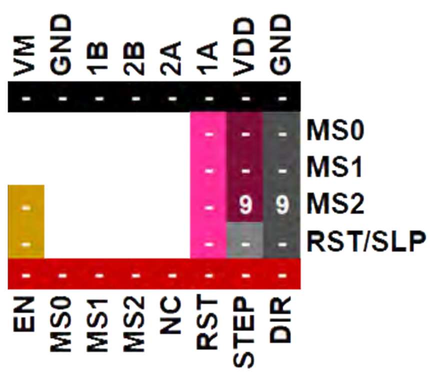
1 / 8





See Appendix D for legend

1 / 16



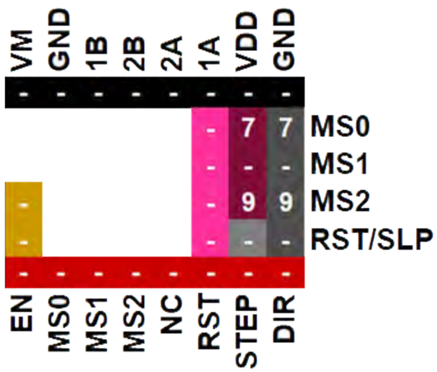
See Appendix D for legend

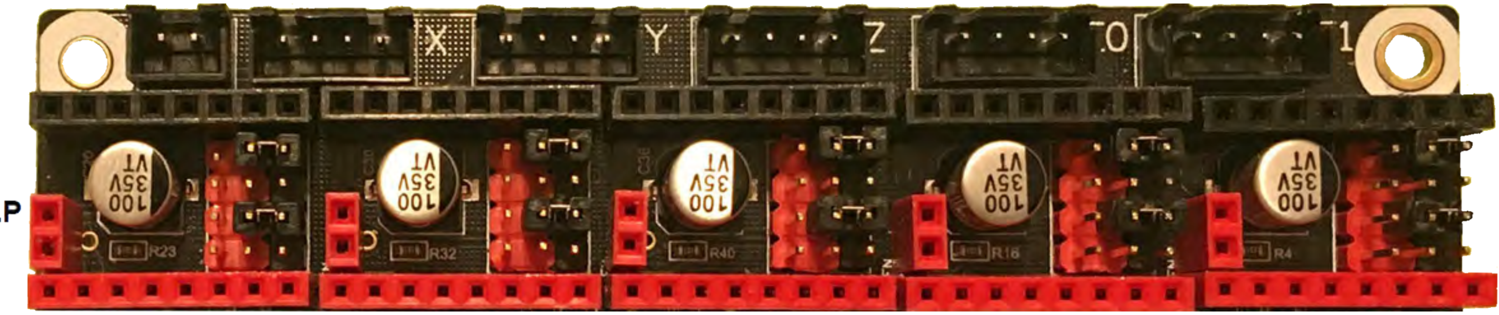
Stand-alone Mode

FYSETC ST820

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

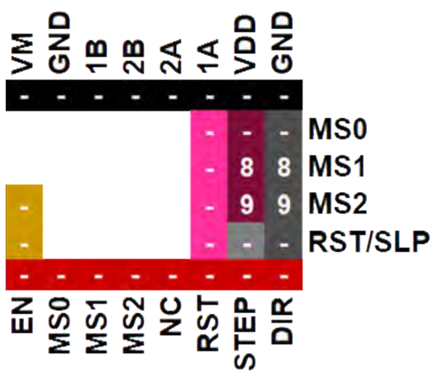
1 / 32

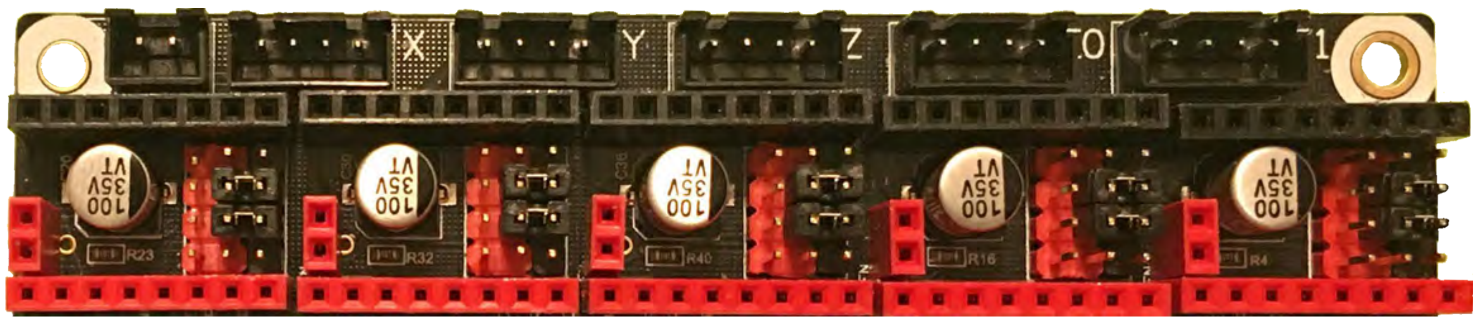




See Appendix D for legend

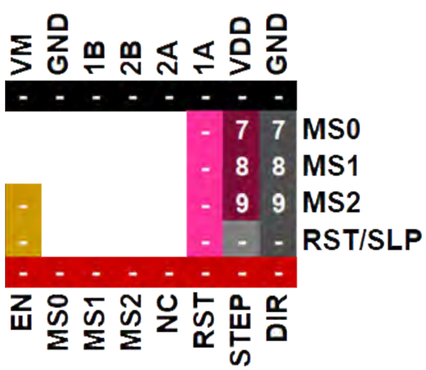
1 / 128

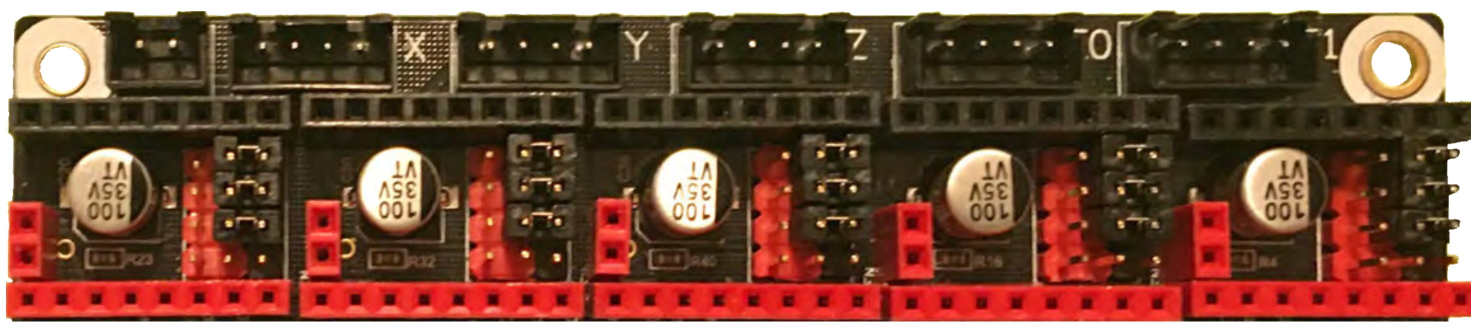




See Appendix D for legend

1 / 256



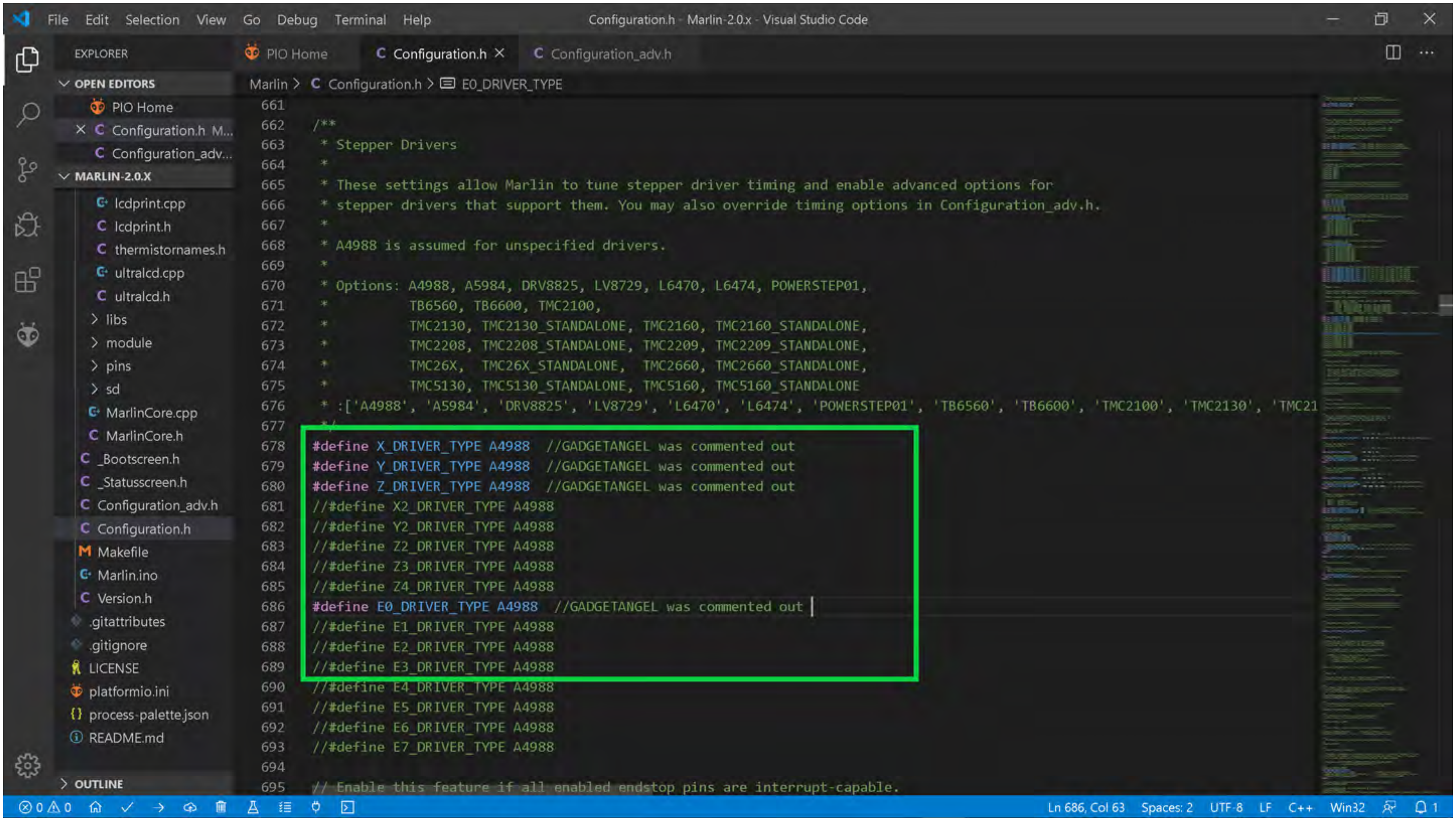


See Appendix D for legend

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

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

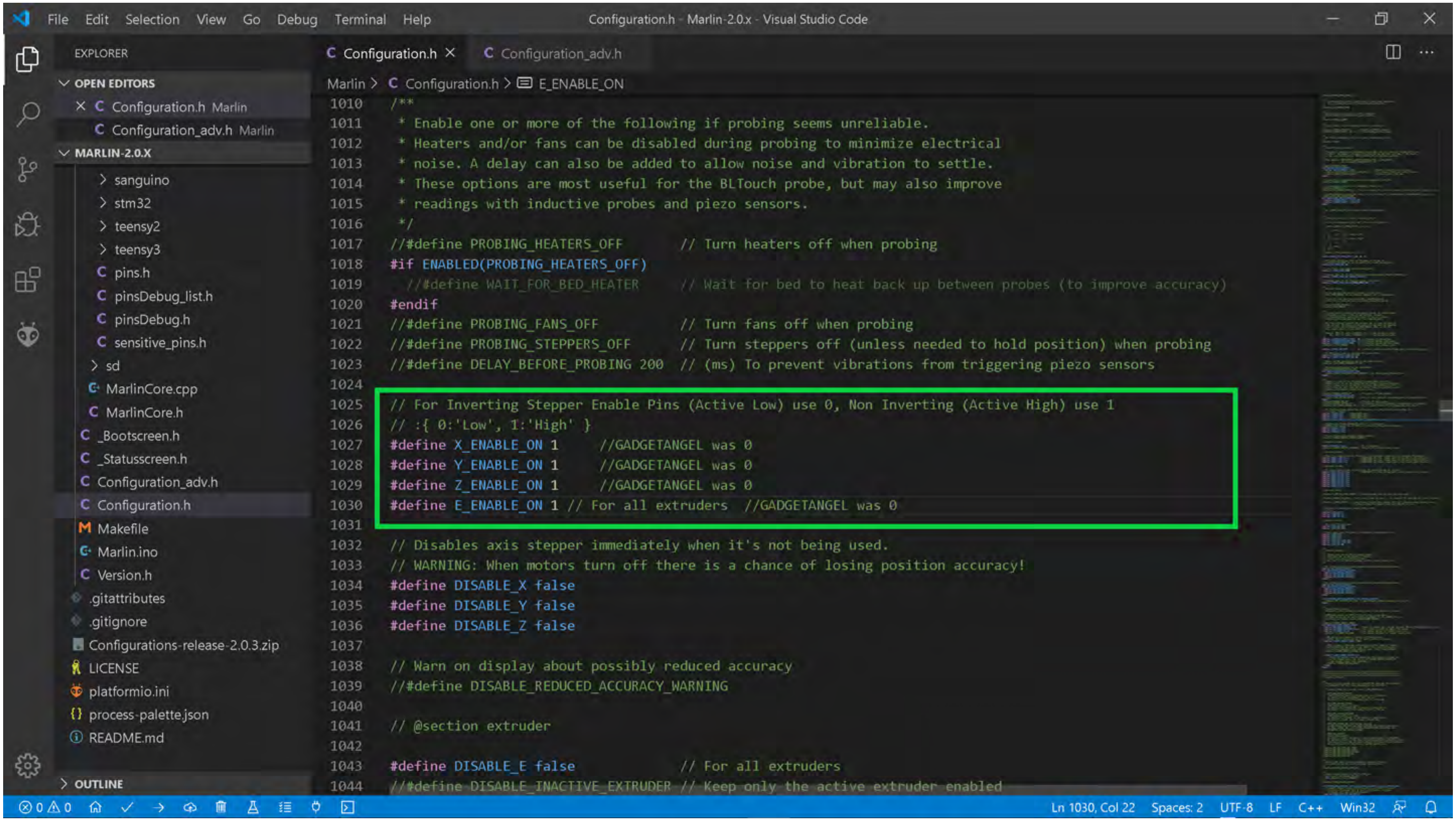
- Change the stepper motor drivers so that Marlin knows you are using FYSETC ST820 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC ST820 drivers. When two "//" appear at the beginning of a line that means that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



- Go to the next page.

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

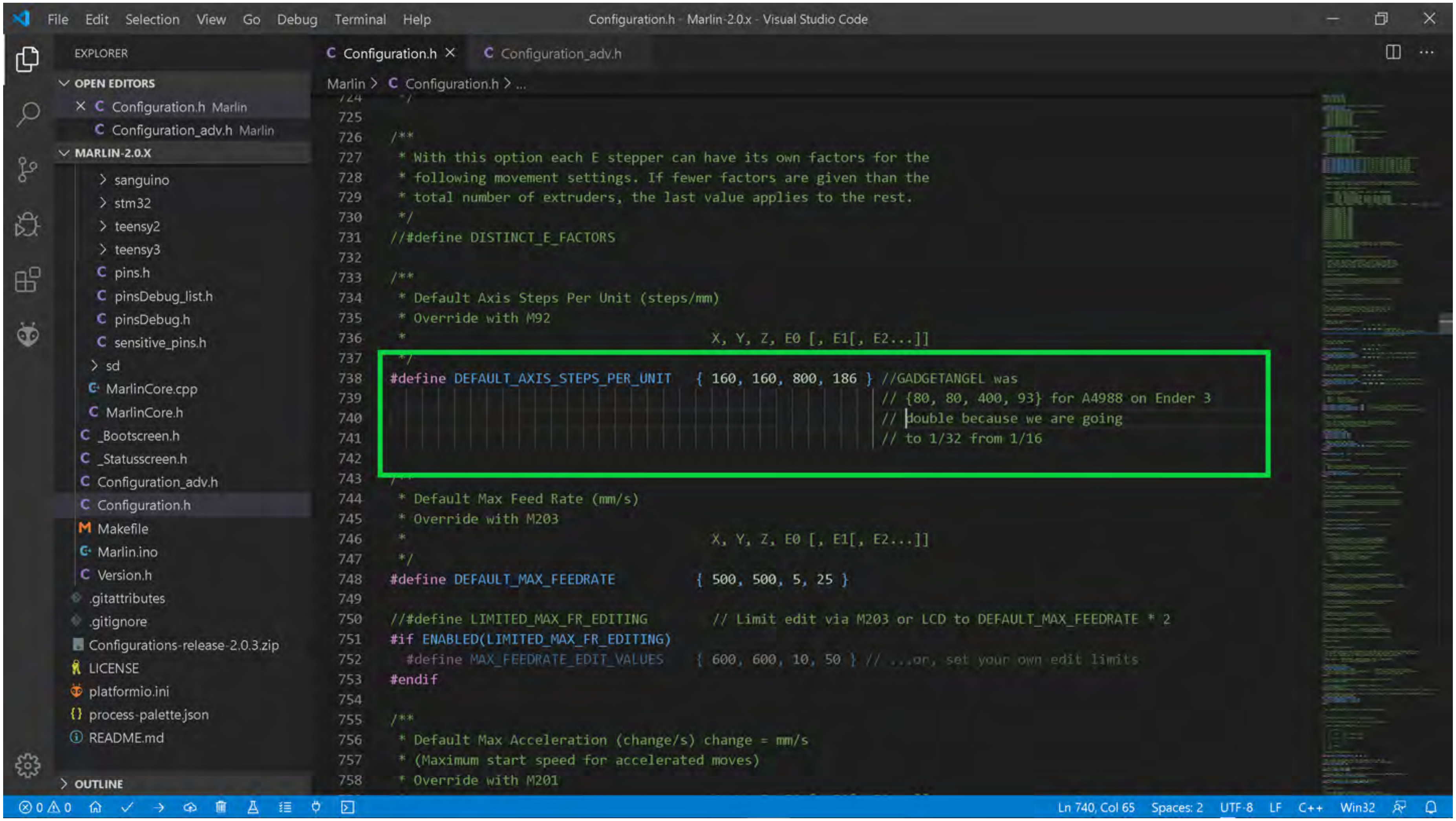
- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the **GREEN** box below.



- Go to the next page.

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

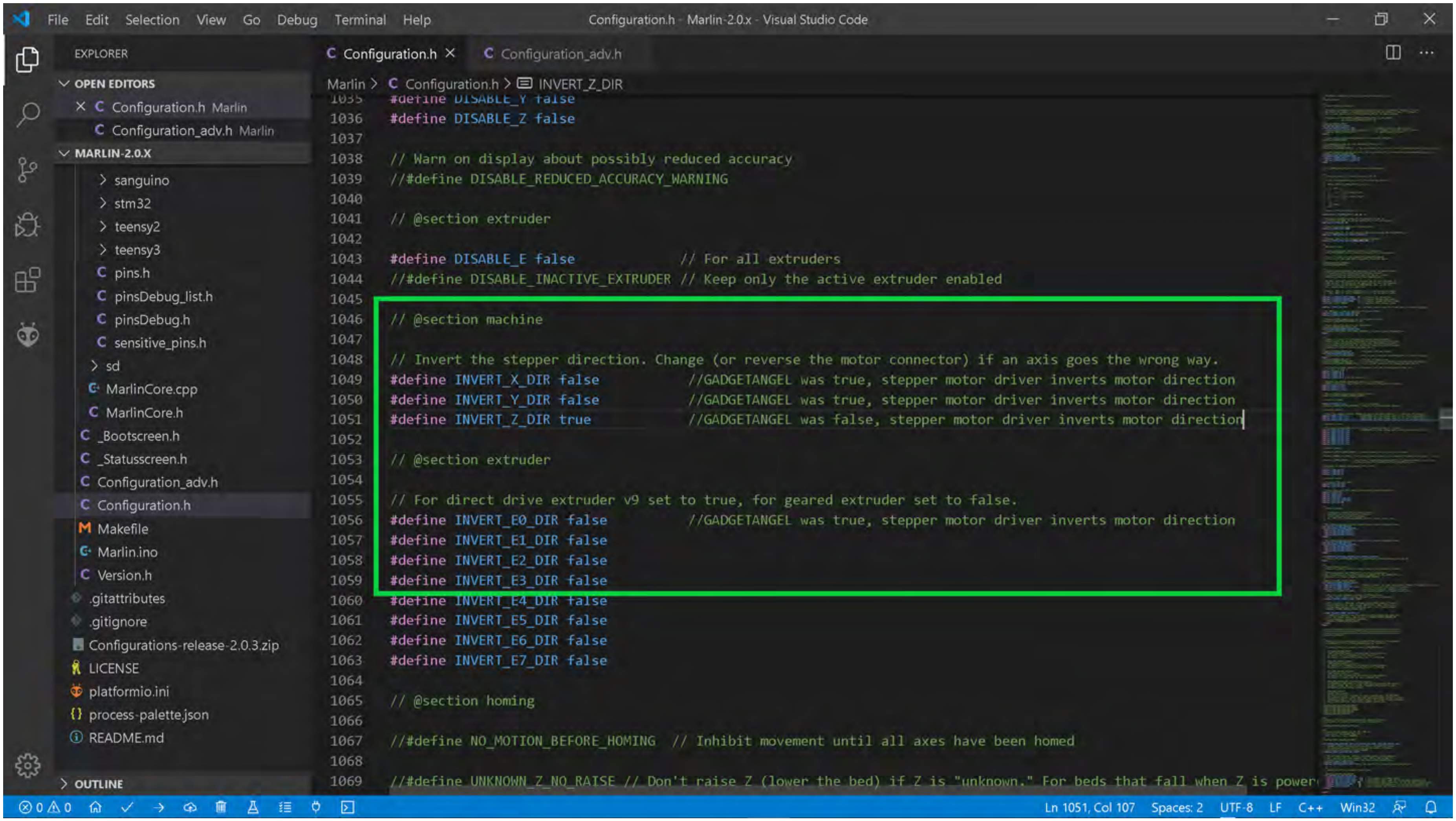
- We are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC ST820 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT"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



- Go to the next page.

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

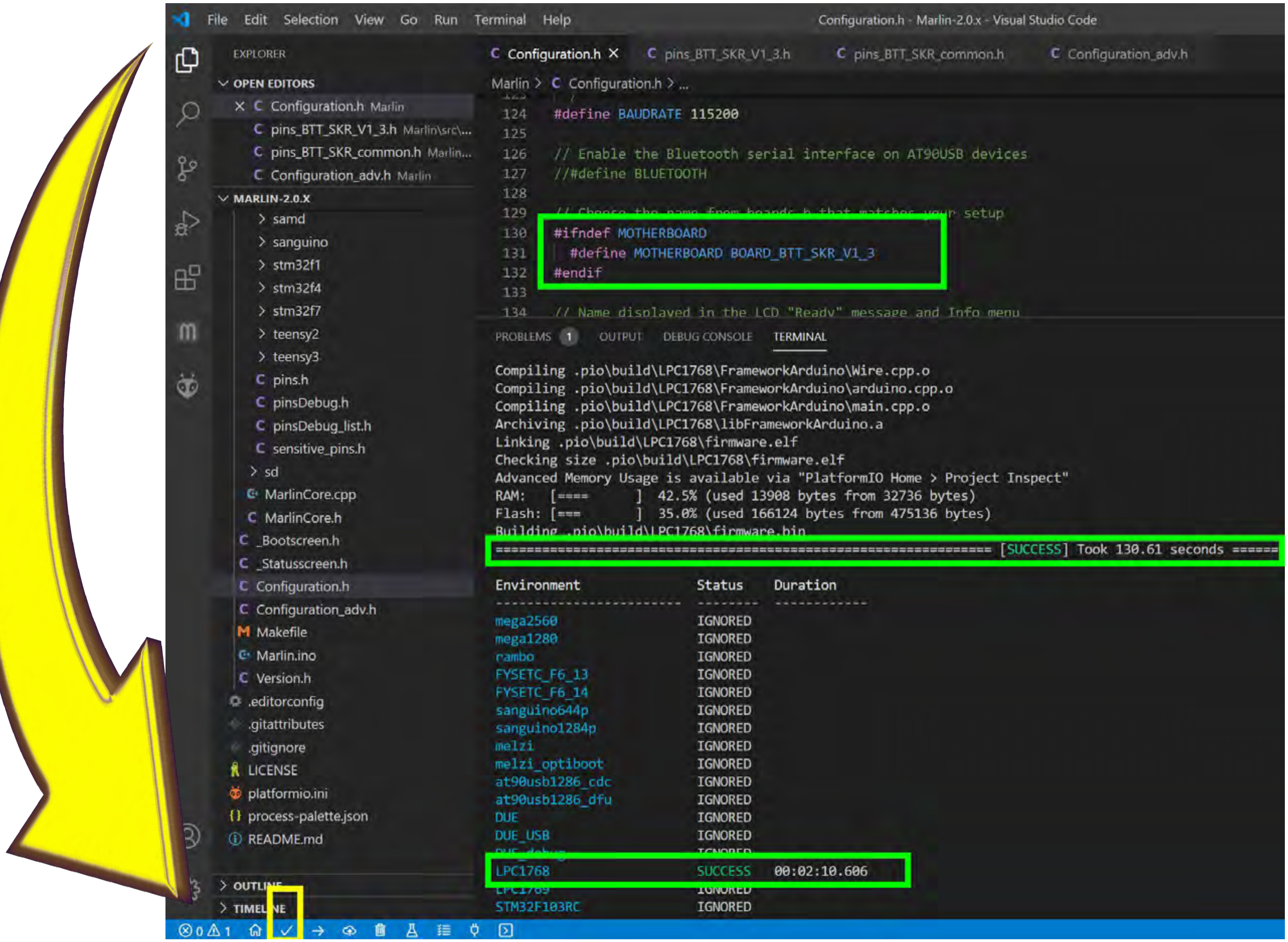
- Optional Step: I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring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



- Go to the next page.

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

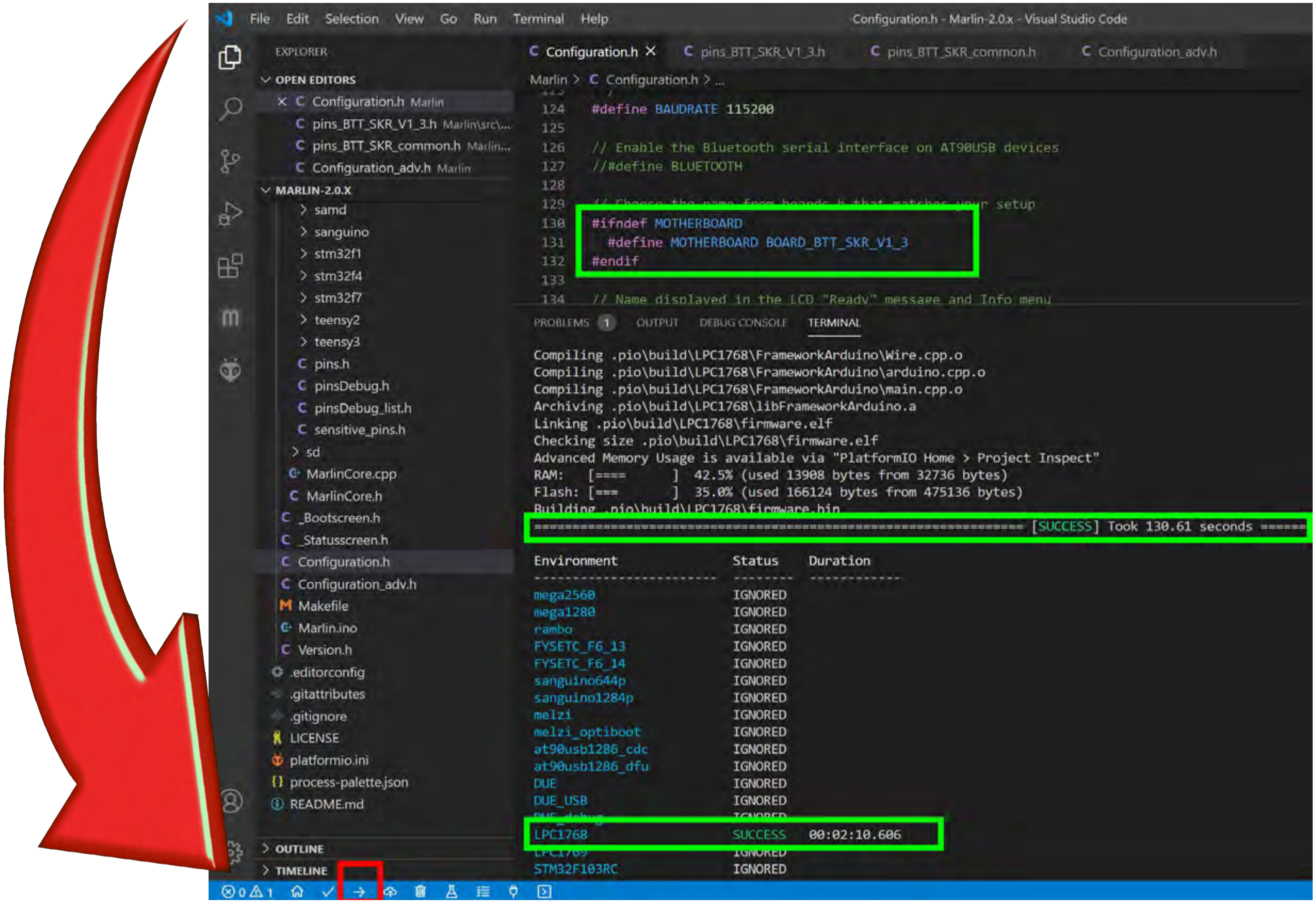
- The end of Marlin setup for FYSETC ST820 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



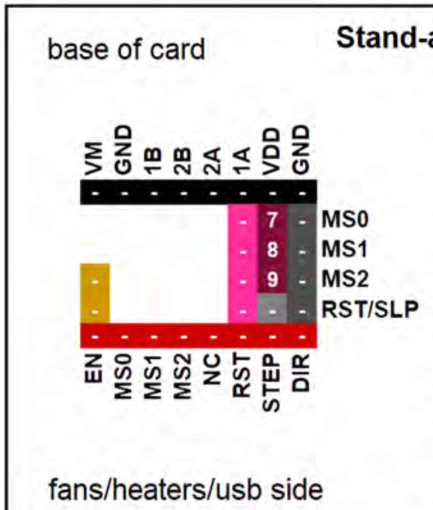
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

See Appendix D for legend

STATUS: Not Yet Tested

BIQU ST820

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
BIQU® ST820	Low	Low	Low	Full step	2 Phase
Maximum 256 Subdivision	High	Low	Low	Half step	1-2 Phase
45V DC	Low	High	Low	1/4 step	W1-2 Phase
1.5A (peak)	High	High	Low	1/8 step	2W1-2 Phase
	Low	Low	High	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	Low	High	High	1/128 step	16W1-2 Phase
	High	High	High	1/256 step	32W1-2 Phase

Driving Current Calculation Formula		
V_{DD} = 3.3 V or 5 V DC R_S(Typical Sense Resistor)= 0.15Ω	$I_{MAX}=V_{ref}*\left(\frac{V_{DD}}{5}\right)*\frac{1}{R_S}$	$V_{ref}=I_{MAX}*\left(\frac{5}{V_{DD}}\right)*R_S$

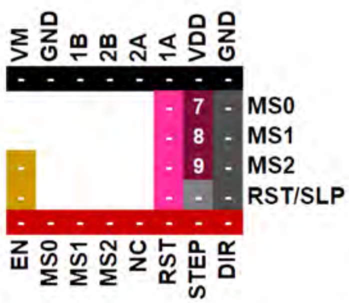
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU ST820

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Low ⇨ No Jumper

Rows:

High ⇨ Jumper Set

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
X.XA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

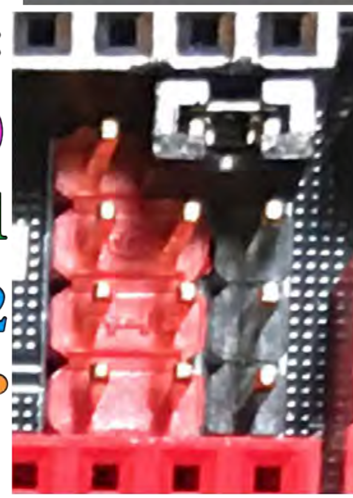
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

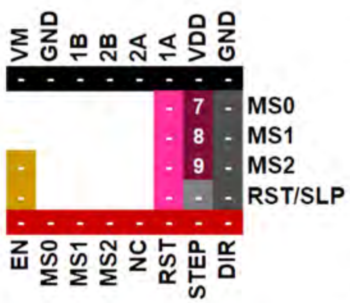
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

BIQU ST820

base of card

Stand-alone Mode



	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

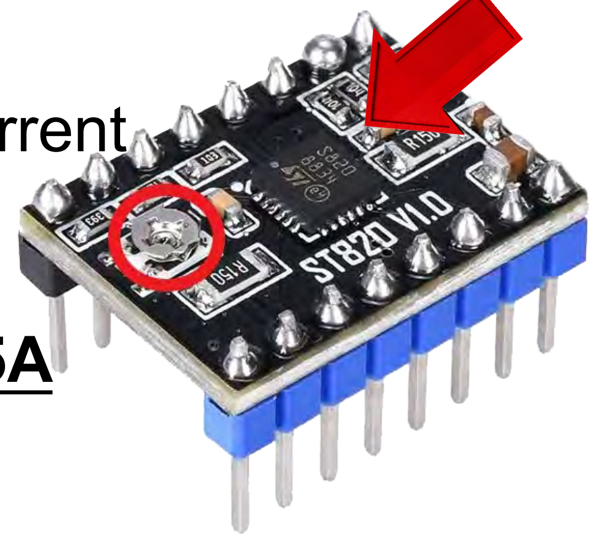
See Appendix D for legend

STATUS: Not Yet Tested

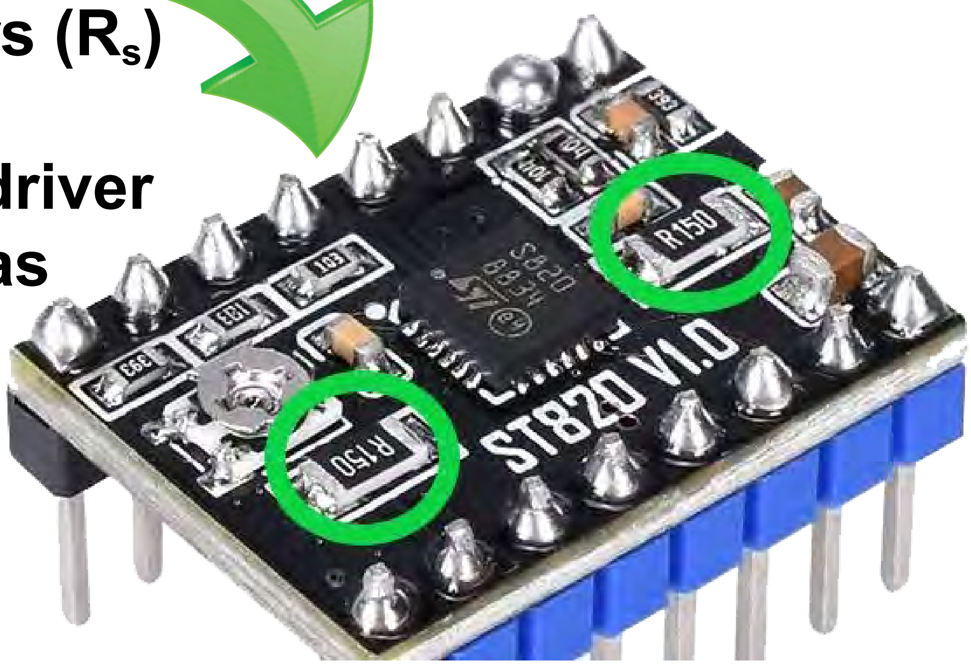
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

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



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



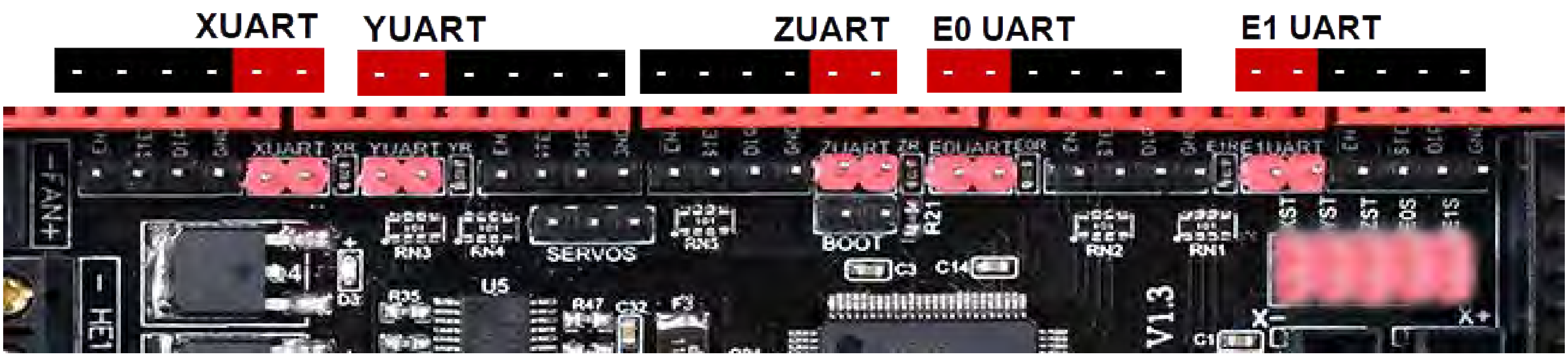
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

BIQU ST820

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

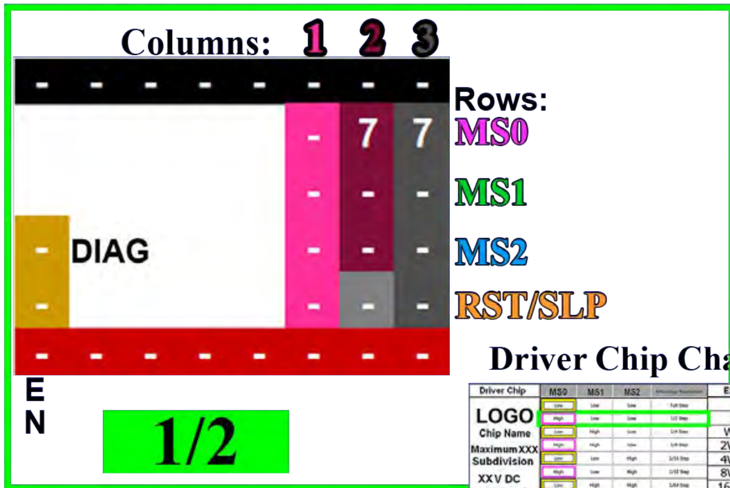


Stand-alone Mode

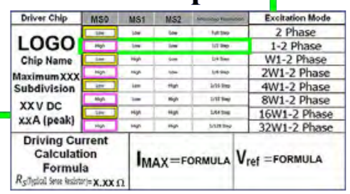
BIQU ST820

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

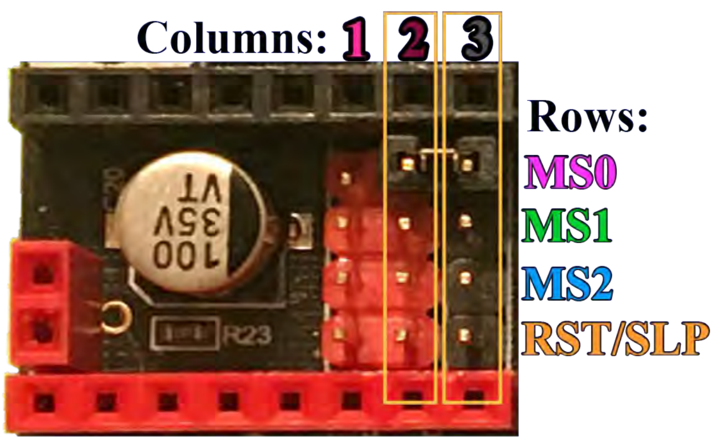
Driver Socket Representation



Driver Chip Chart:



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

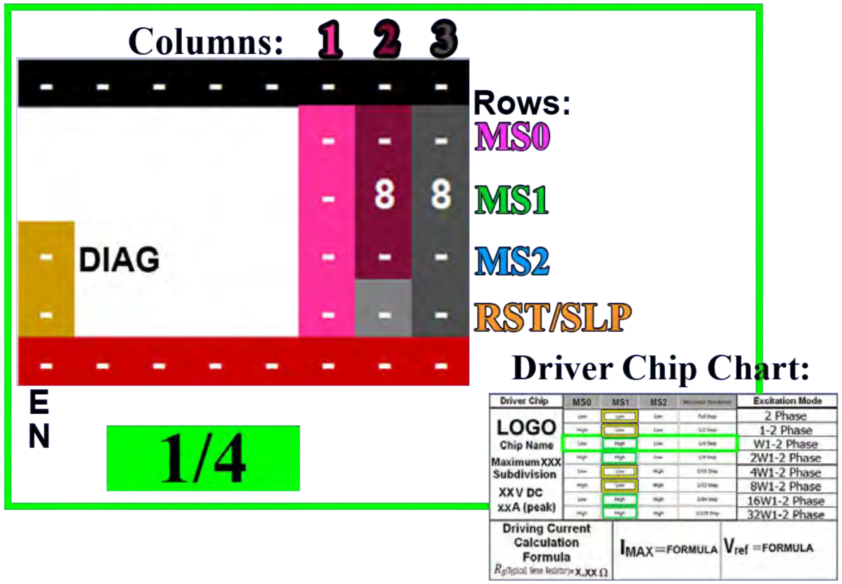
Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Chip Chart:	**Meaning:**
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

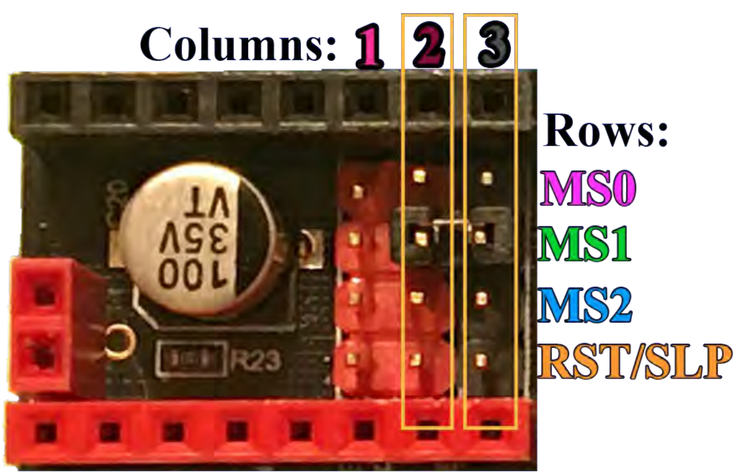
Stand-alone Mode

BIQU ST820

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 8 8 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

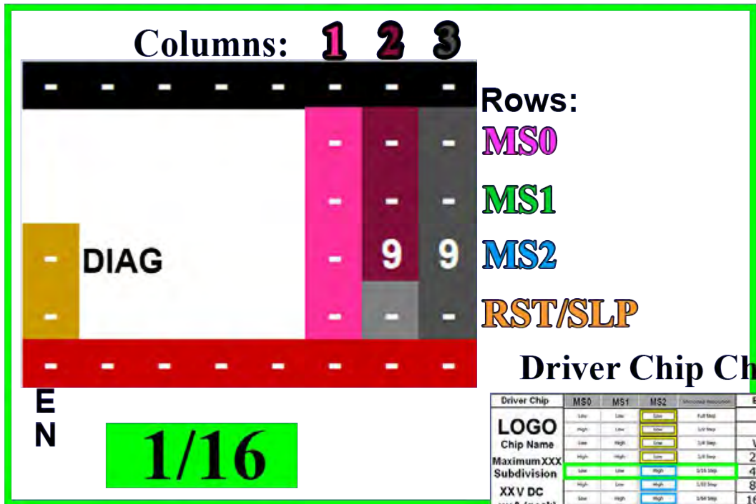
High → set Jumper between column 2 and column 3 on the MS1 row

Low → No Jumper Set

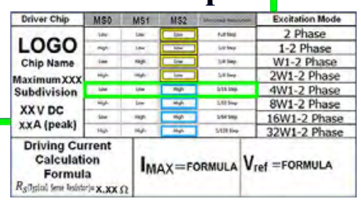
Stand-alone Mode

BIQU ST820

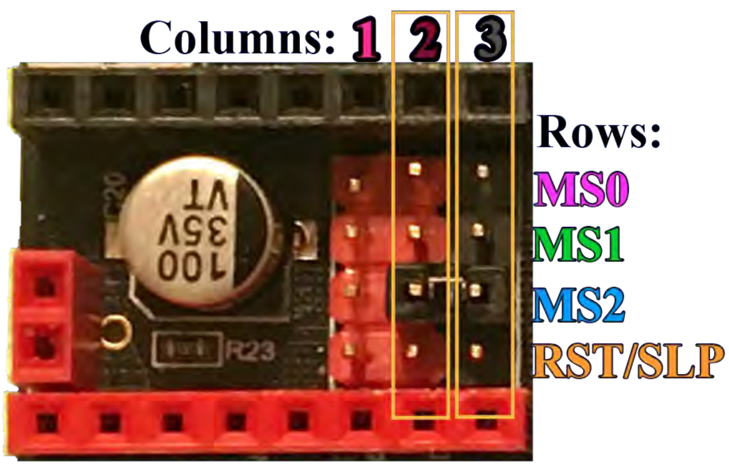
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

High → set Jumper between column 2 and column 3 on the MS2 row

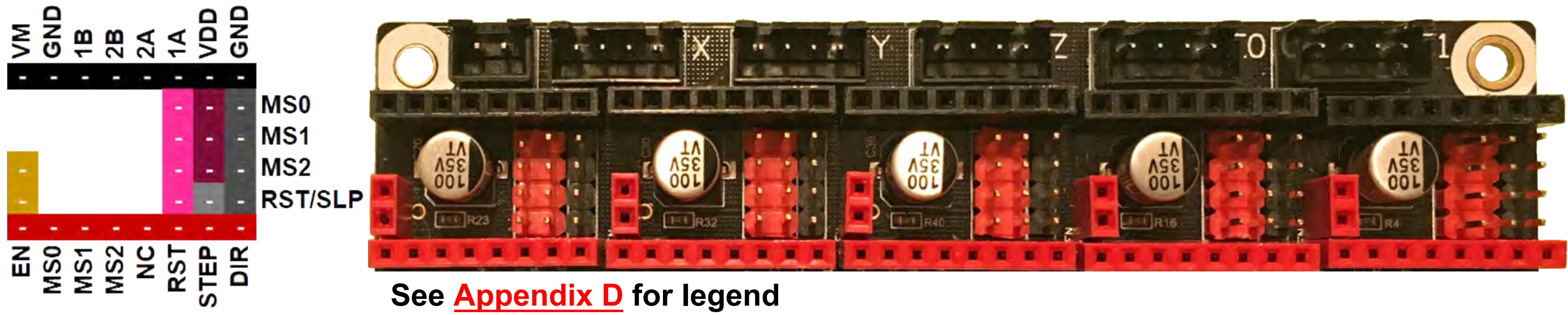
Low → No Jumper Set

Stand-alone Mode

BIQU ST820

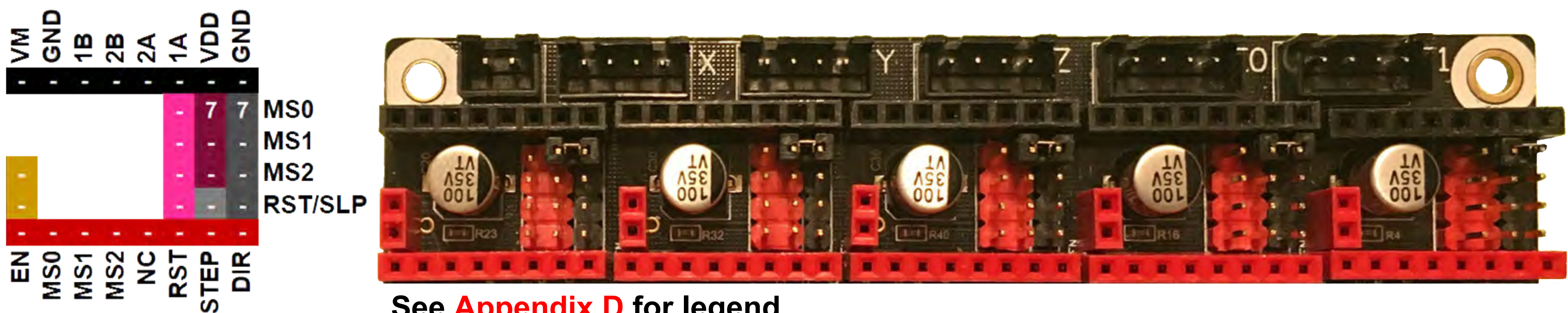
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

STEP



See Appendix D for legend

1 / 2



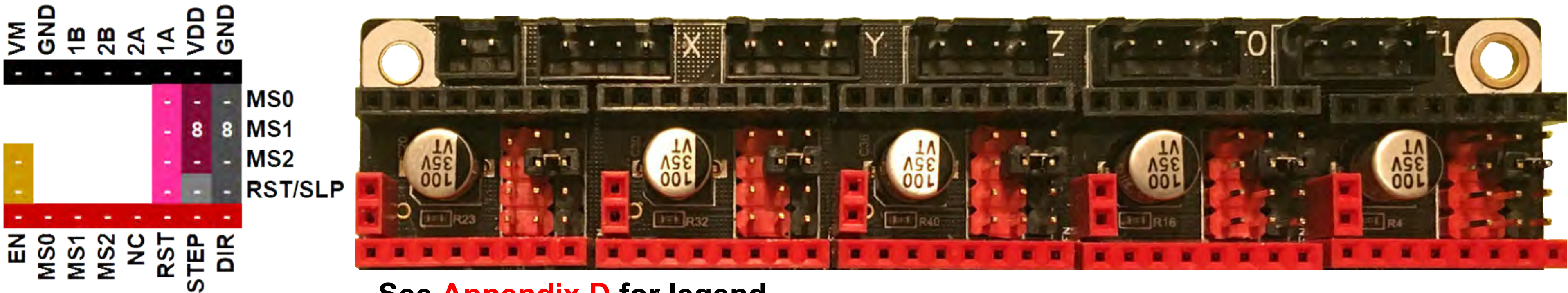
See Appendix D for legend

Stand-alone Mode

BIQU ST820

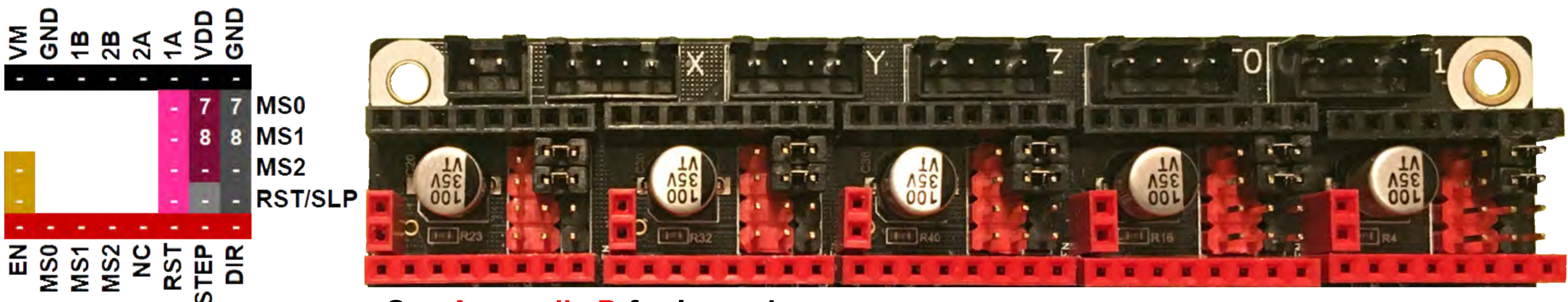
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 4



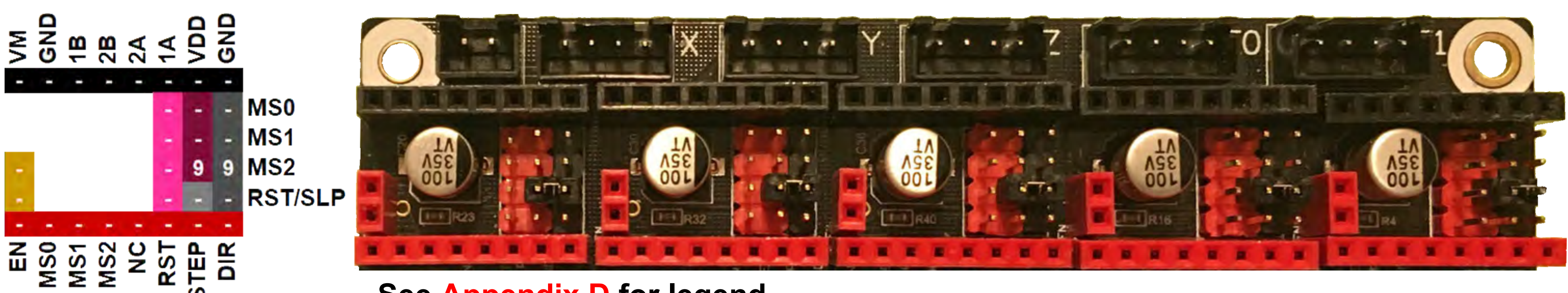
See Appendix D for legend

1 / 8



See Appendix D for legend

1 / 16



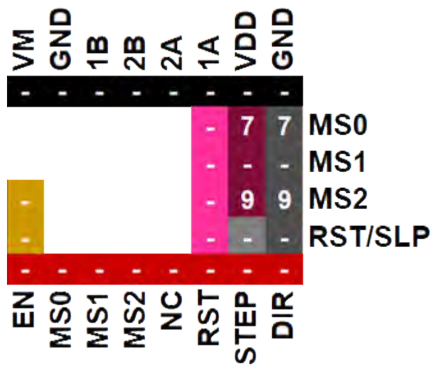
See Appendix D for legend

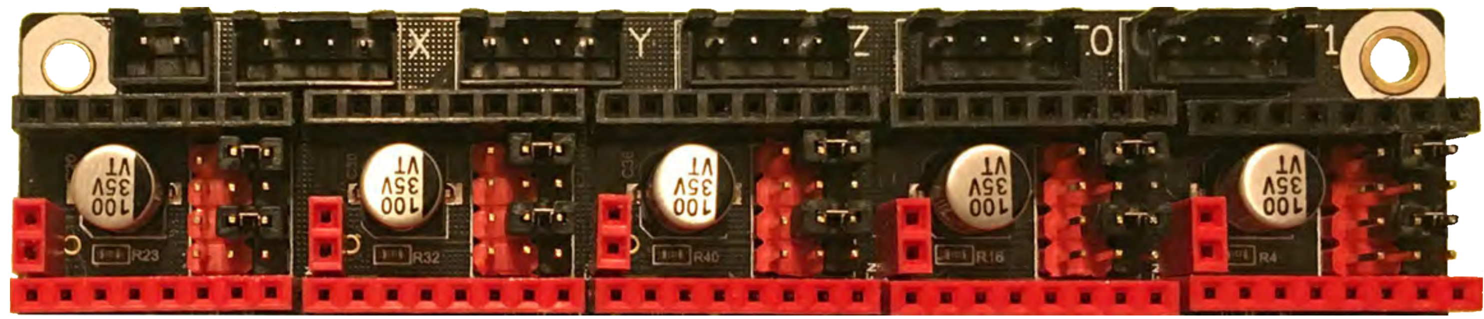
Stand-alone Mode

BIQU ST820

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

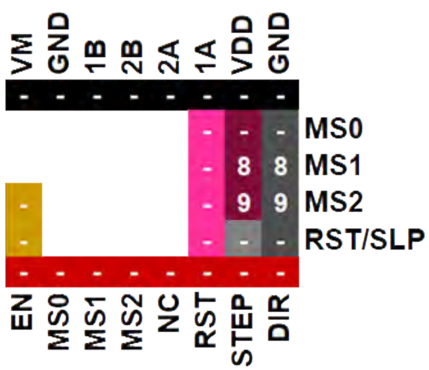
1 / 32

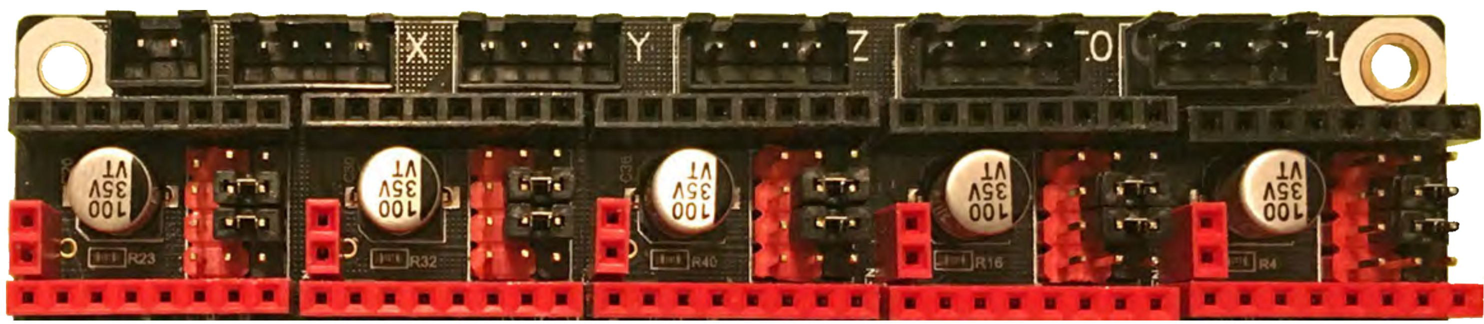




See Appendix D for legend

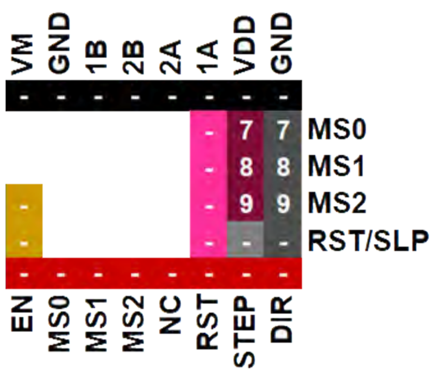
1 / 128

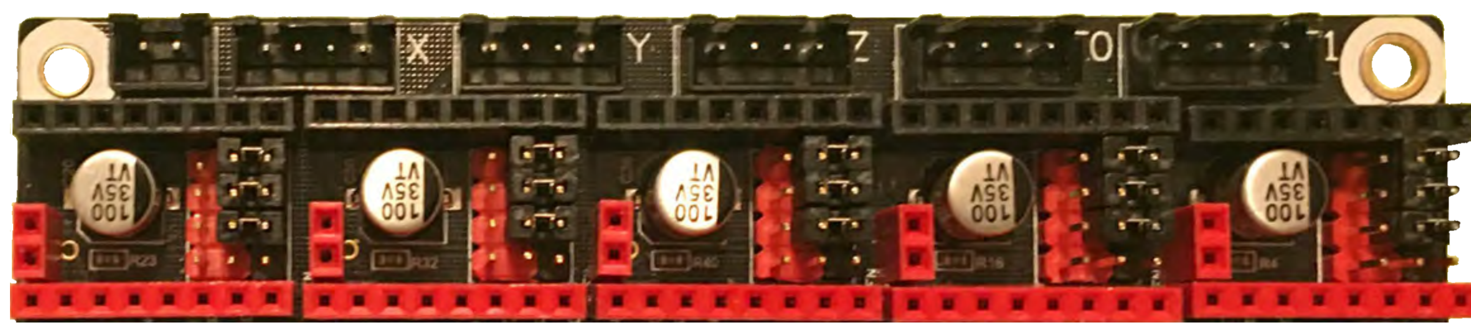




See Appendix D for legend

1 / 256



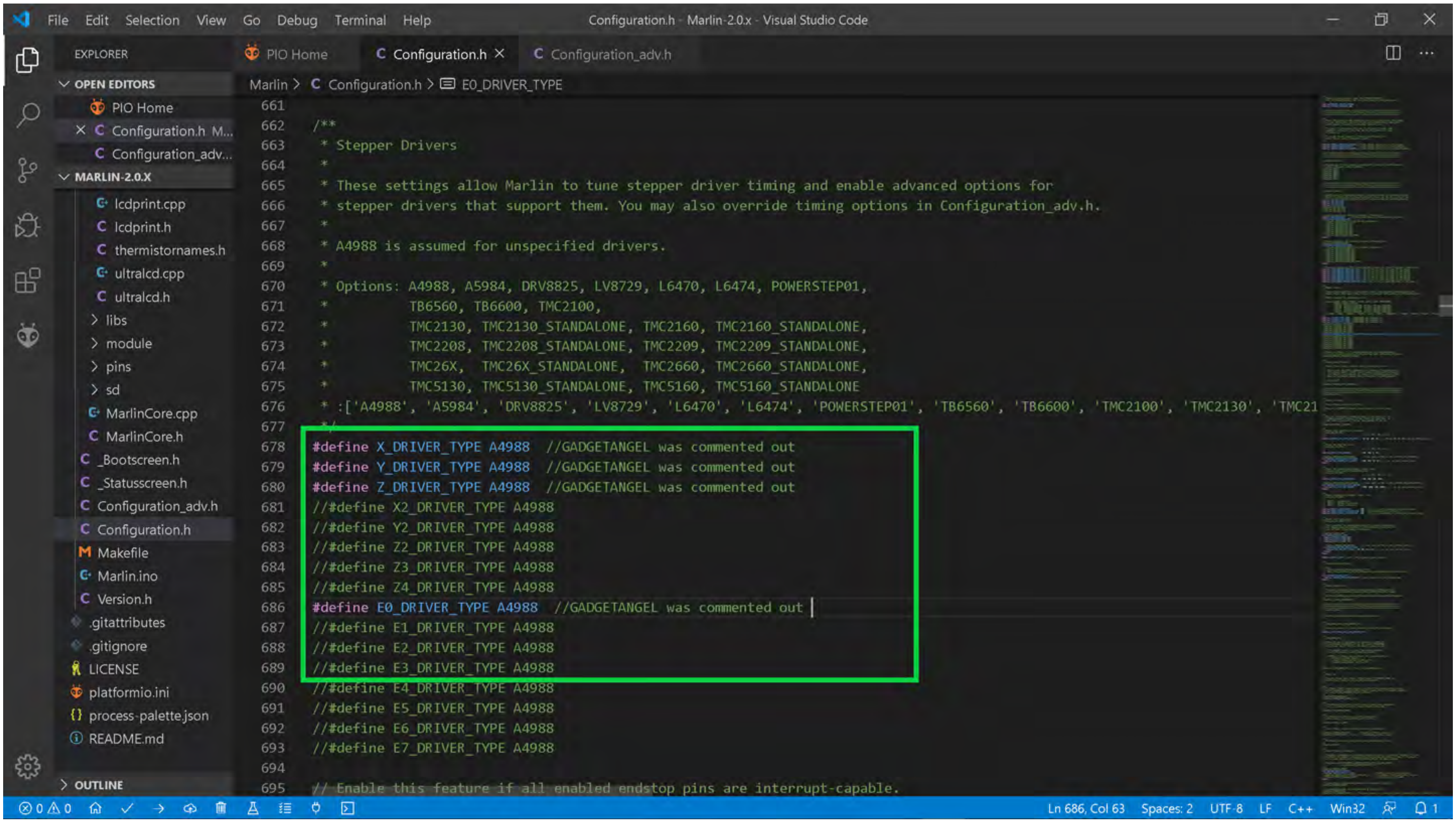


See Appendix D for legend

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

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

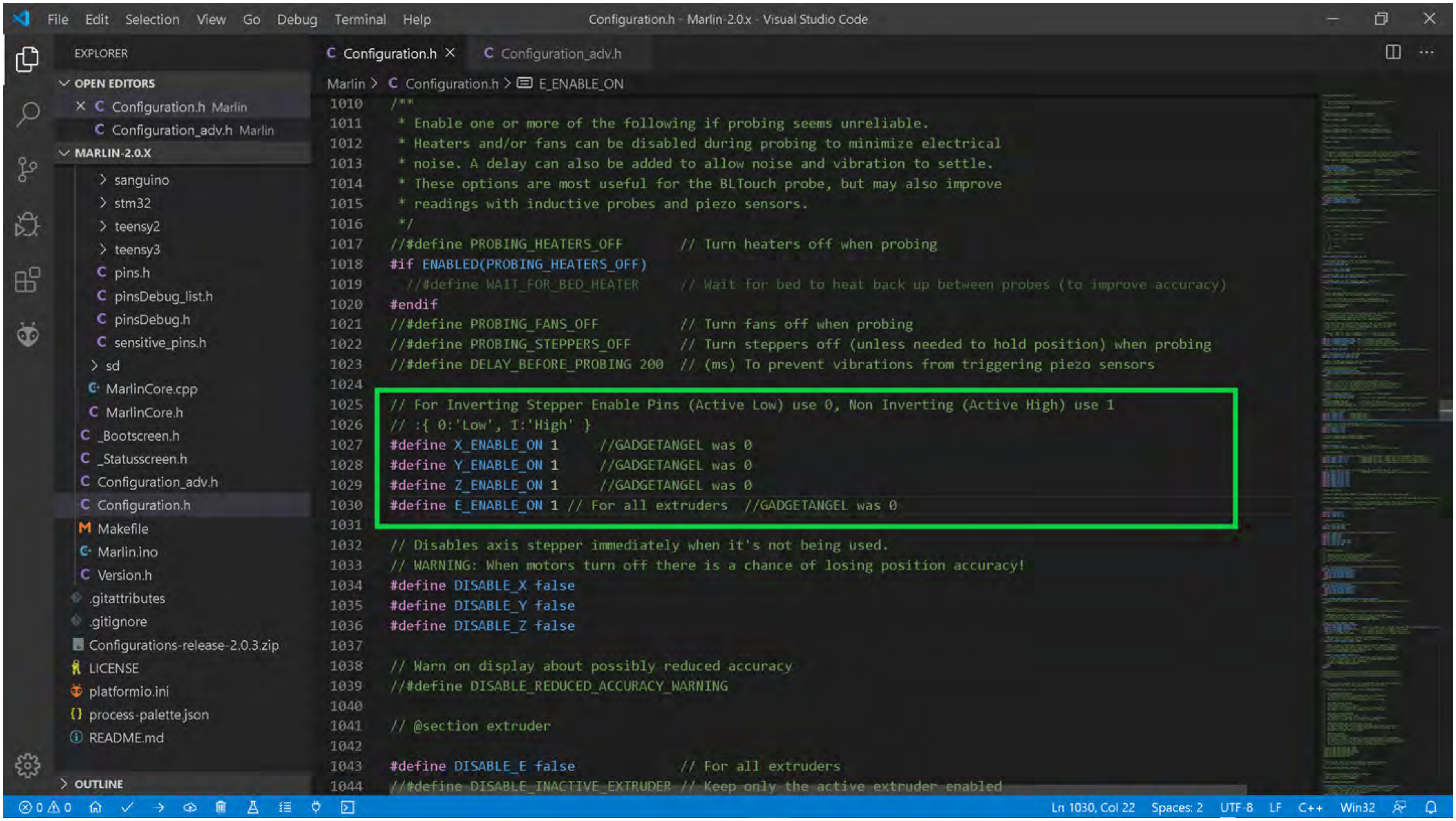
- Change the stepper motor drivers so that Marlin knows you are using BIQU ST820 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU ST820 drivers. When two "//" appear at the beginning of a line that means that line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



- Go to the next page.

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

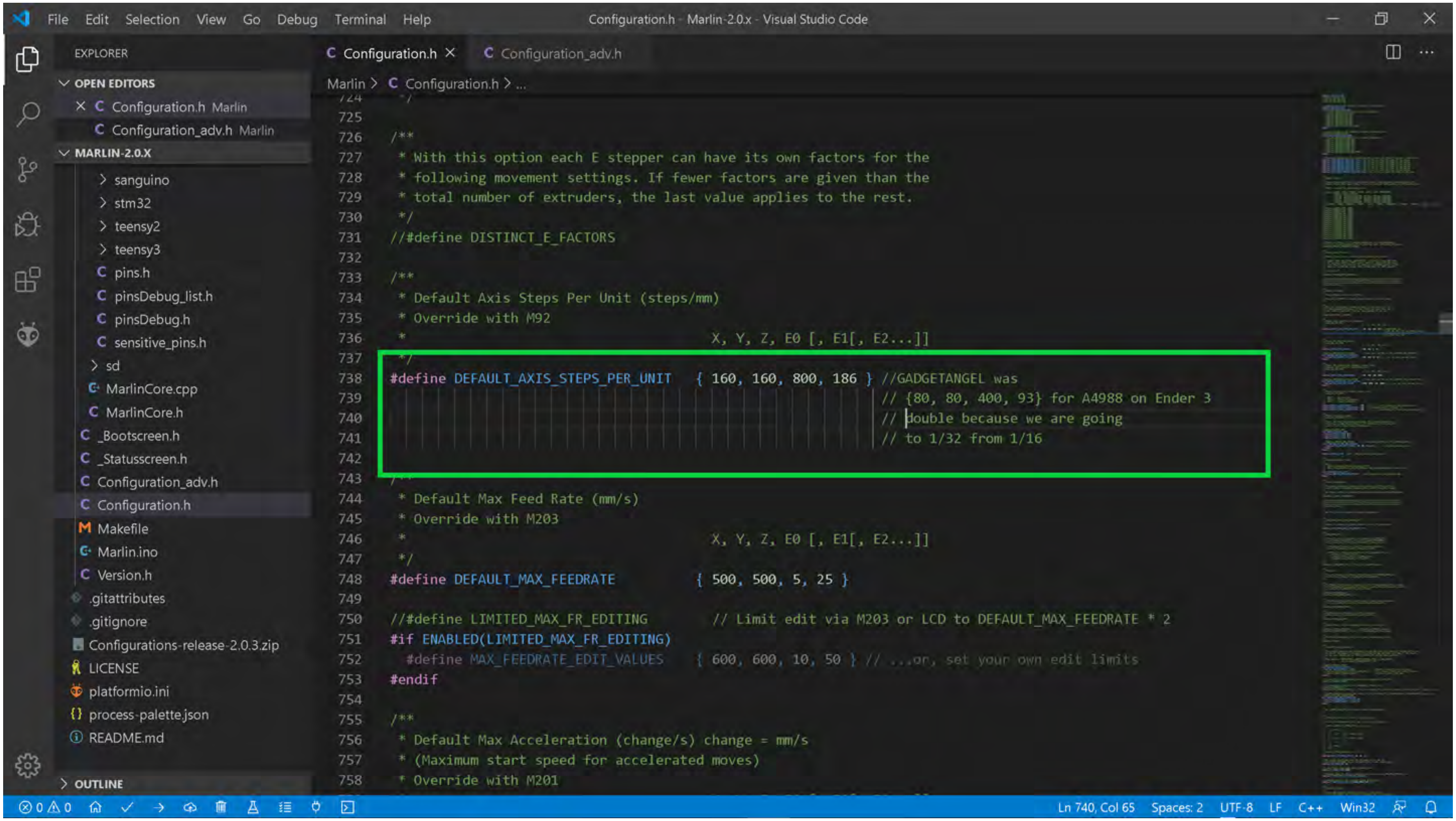
- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the **GREEN** box below.



- Go to the next page.

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

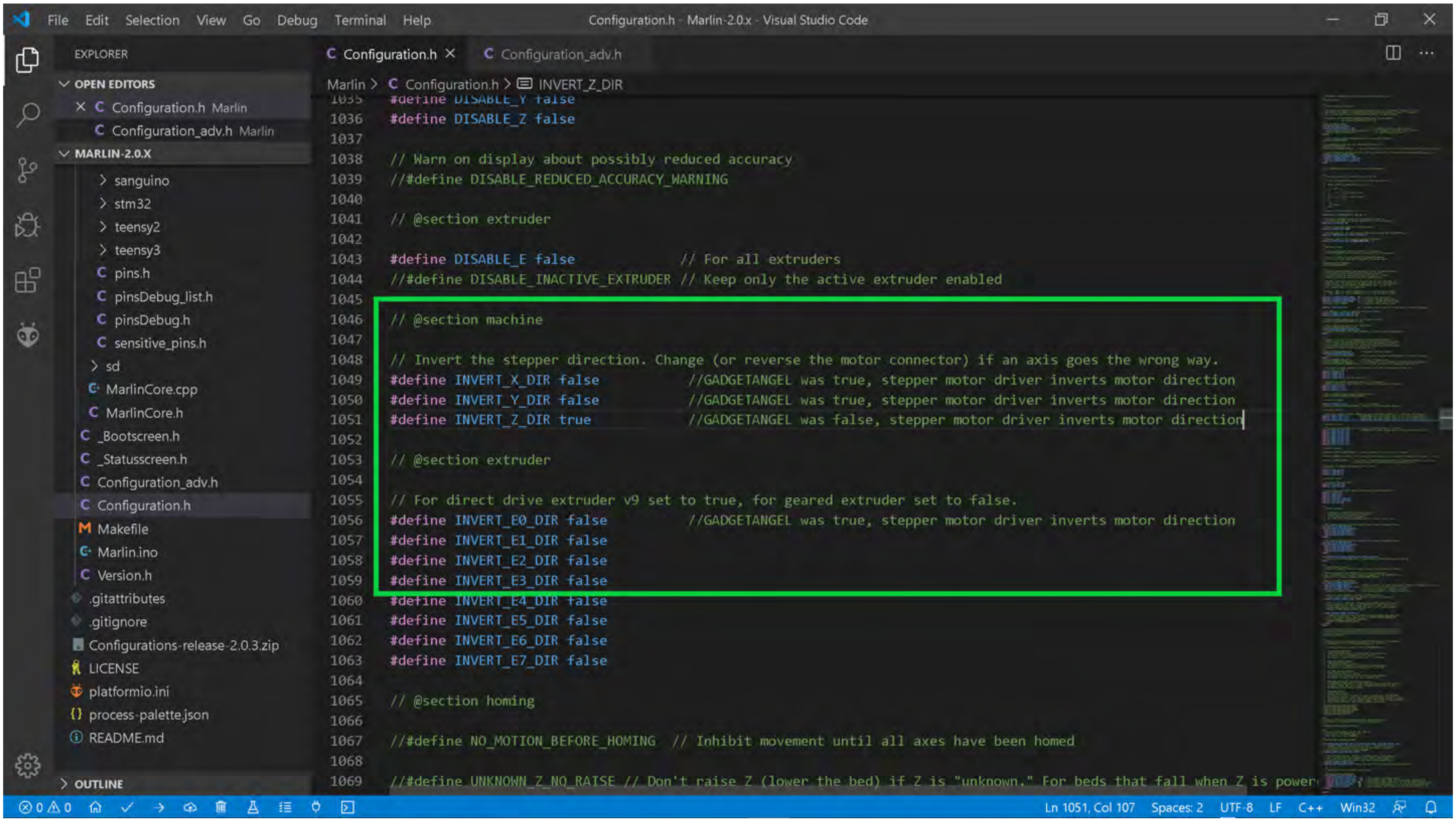
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to BIQU ST820 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT"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



- Go to the next page.

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

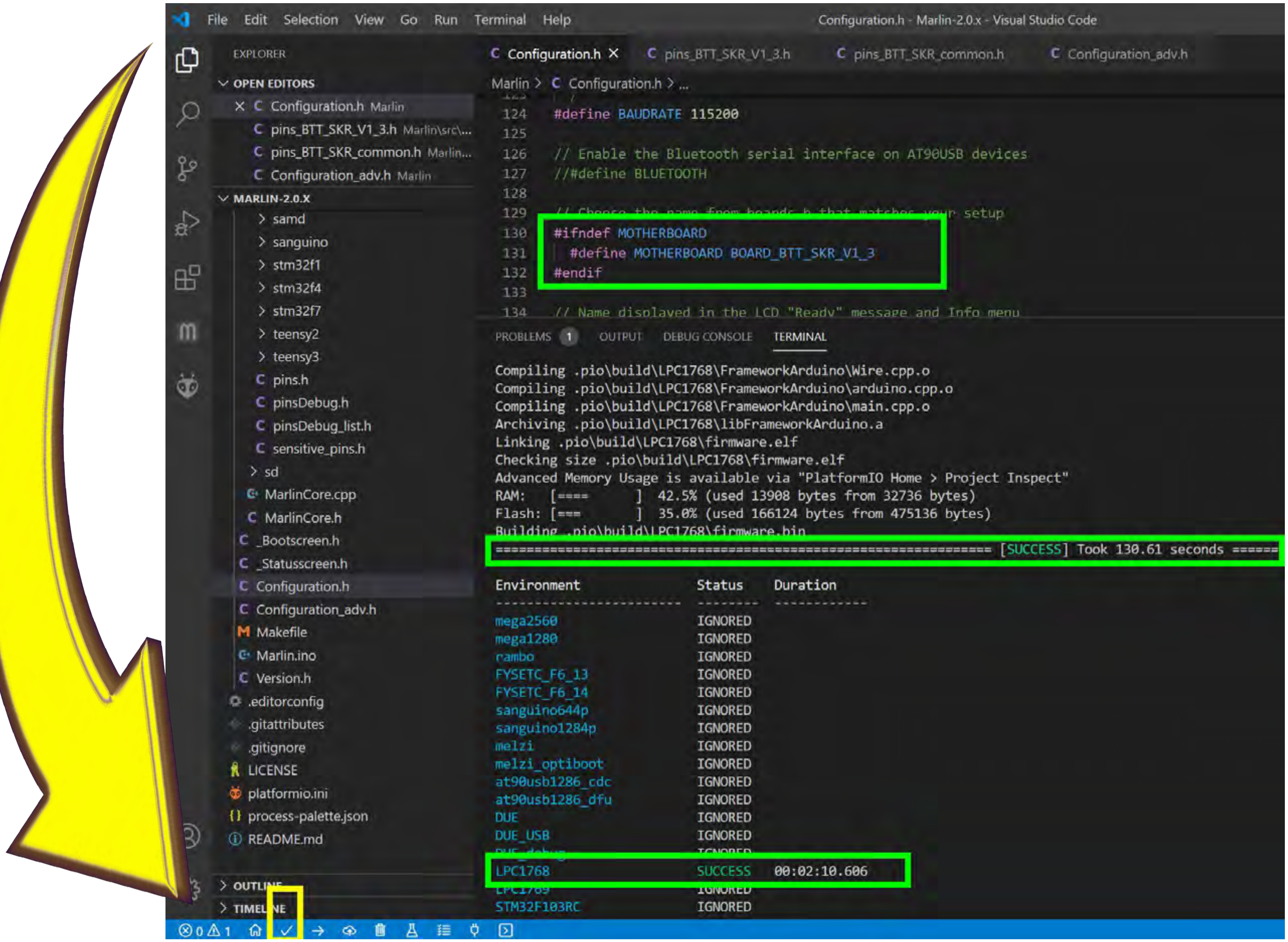
- Optional Step: I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide, the below information, in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring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



- Go to the next page.

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

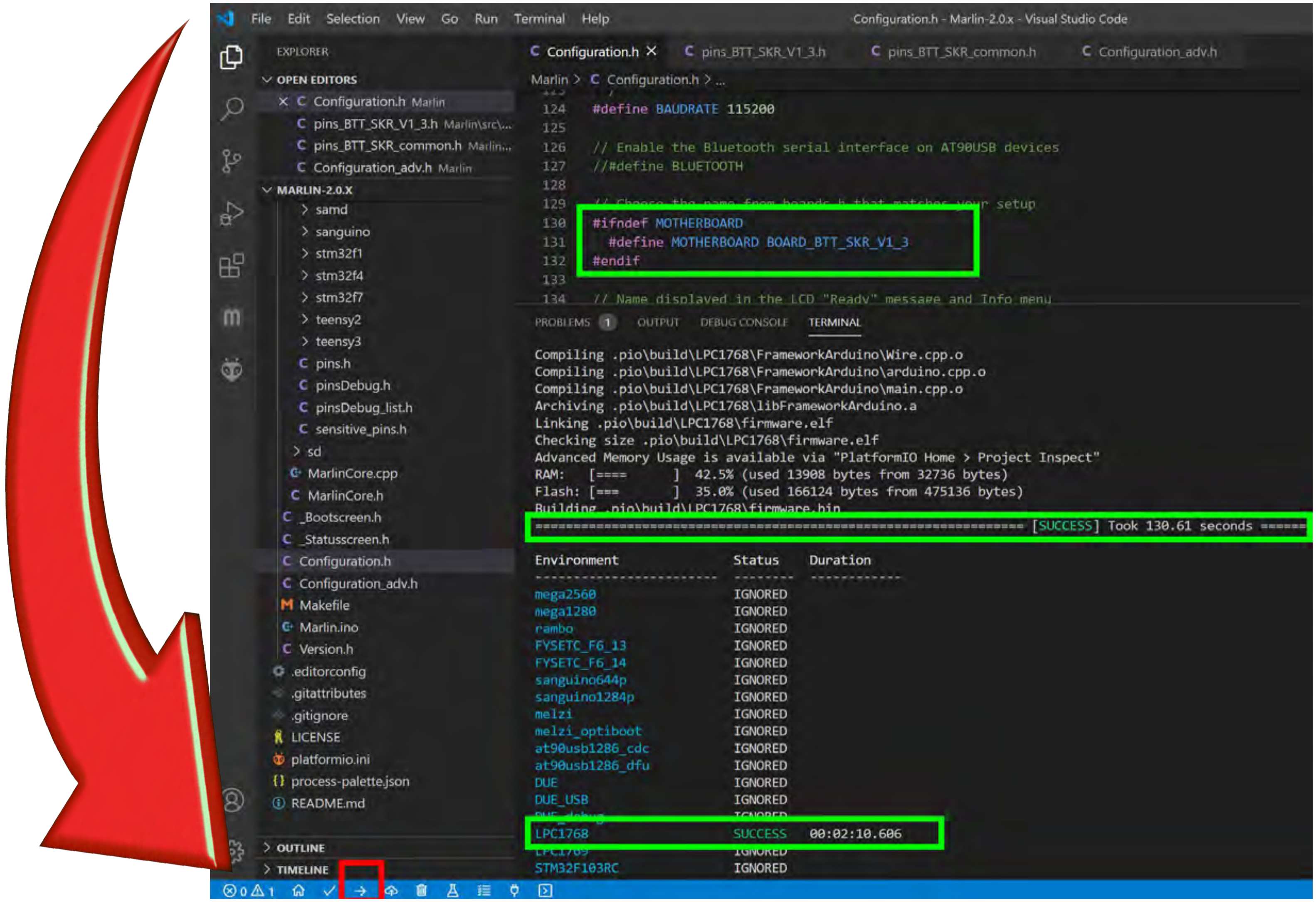
- The end of Marlin setup for BIQU ST820 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

POLOLU ST820 (STSPIN820)

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

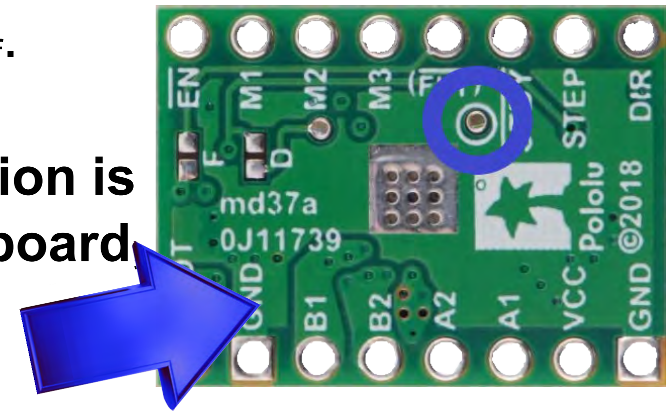
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

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



Driver Chip: Pololu ST820 — Maximum 256 Subdivision — 45V DC 1.5A (peak)

MS0	MS1	MS2	Microstep Resolution	Excitation Mode
Low	Low	Low	Full step	2 Phase
High	Low	Low	Half step	1-2 Phase
Low	High	Low	1/4 step	W1-2 Phase
High	High	Low	1/8 step	2W1-2 Phase
Low	Low	High	1/16 step	4W1-2 Phase
High	Low	High	1/32 step	8W1-2 Phase
Low	High	High	1/128 step	16W1-2 Phase
High	High	High	1/256 step	32W1-2 Phase

Driving Current Calculation Formula

R_S(Typical Sense Resistor)= 0.2Ω

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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

POLOLU ST820 (STSPIN820)

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Low ⇒ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇒ Jumper Set

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
x.xA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= x.xx Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

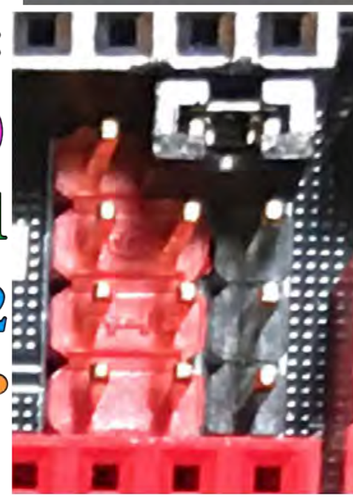
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

POLOLU ST820 (STSPIN820)

Stand-alone Mode

base of card

	0	1	2
1	0	0	0
2	1	0	0
4	0	1	0
8	1	1	0
16	0	0	1
32	1	0	1
128	0	1	1
256	1	1	1

fans/heaters/usb side

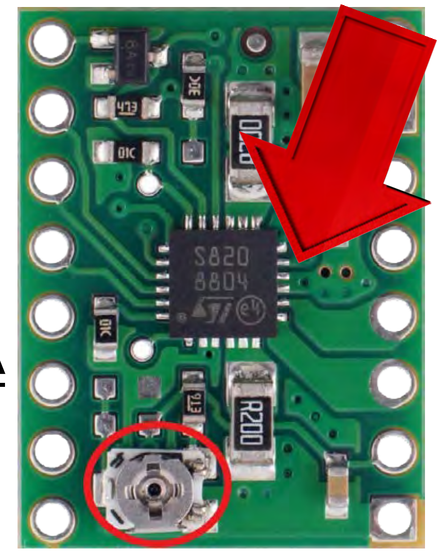
See Appendix D for legend

STATUS: Not Yet Tested

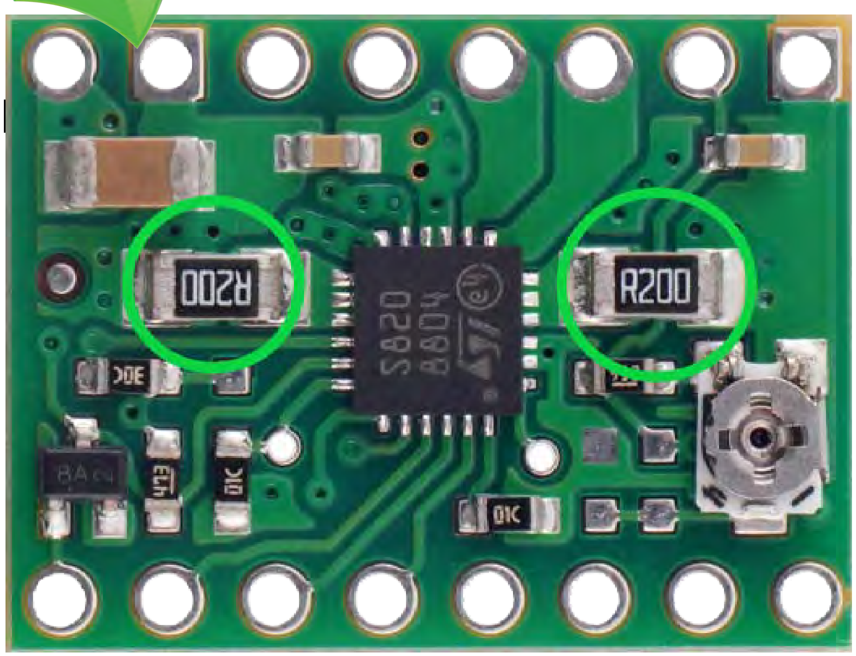
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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

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



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



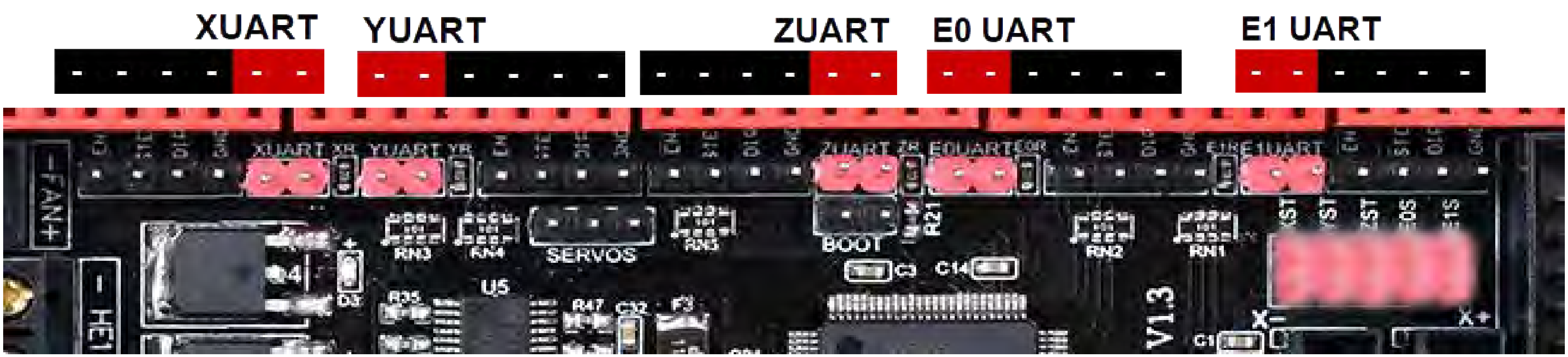
R_s = R050 is 0.05 Ohms
R_s = R068 is 0.068 Ohms
R_s = R100 is 0.1 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms

Stand-alone Mode

POLOLU ST820 (STSPIN820)

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

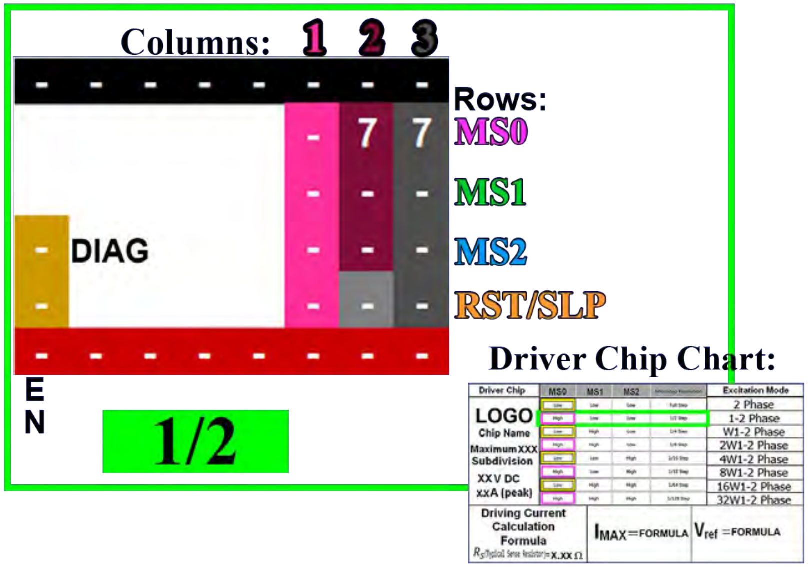


Stand-alone Mode

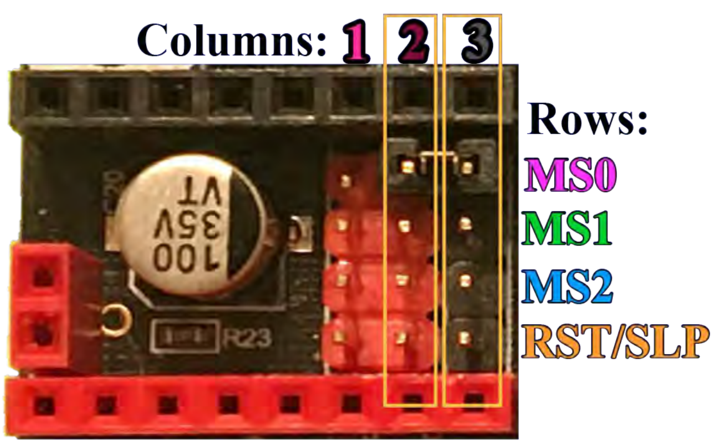
POLOLU ST820 (STSPIN820)

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

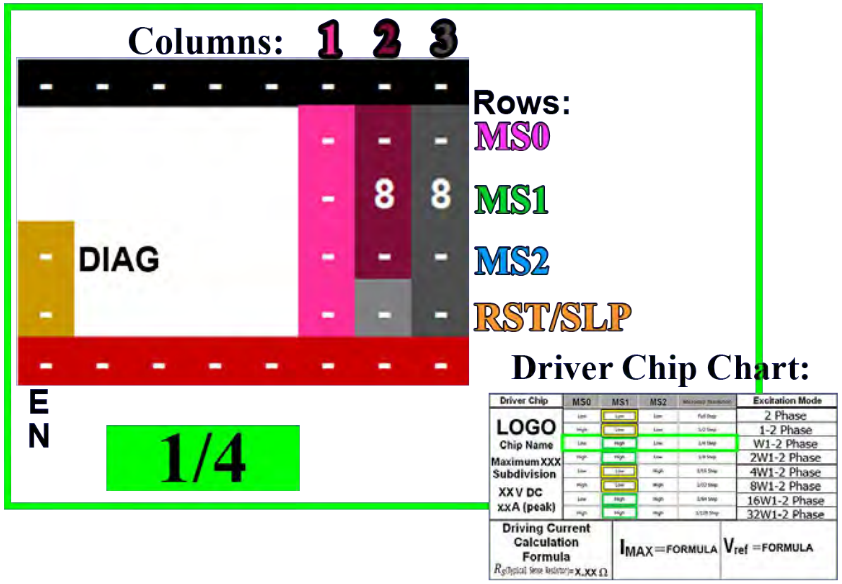
Meaning:

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

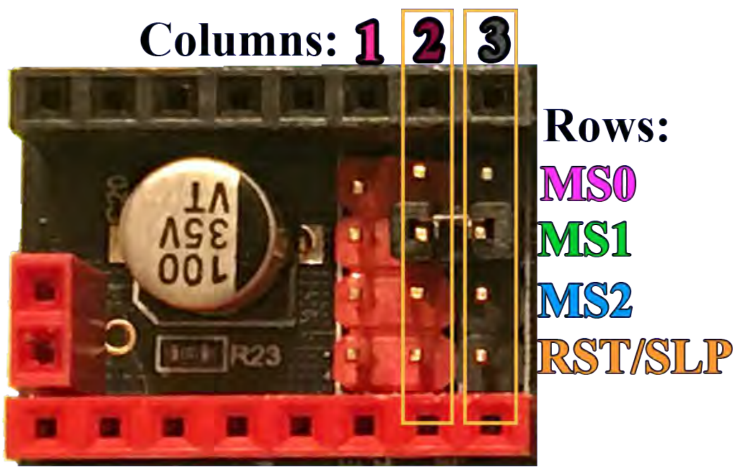
Stand-alone Mode

POLOLU ST820 (STSPIN820)

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Driver Chip Chart:

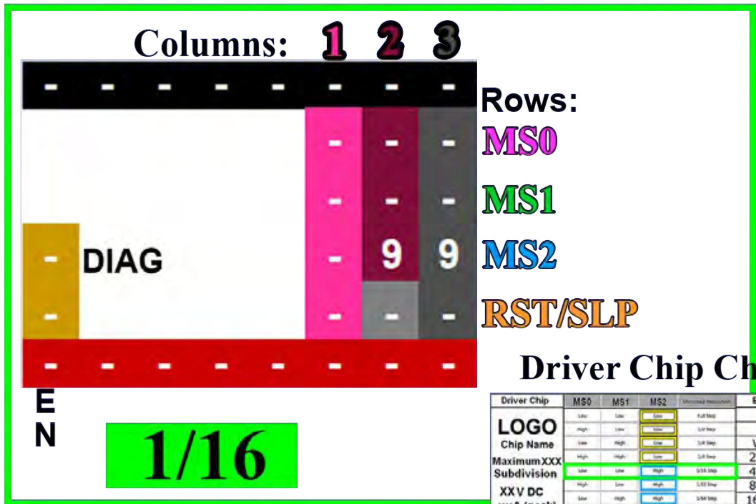
Meaning:

- **High** → **set Jumper between column 2 and column 3 on the MS1 row**
- **Low** → **No Jumper Set**

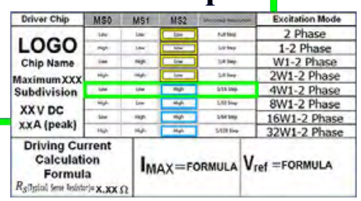
Stand-alone Mode

POLOLU ST820 (STSPIN820)

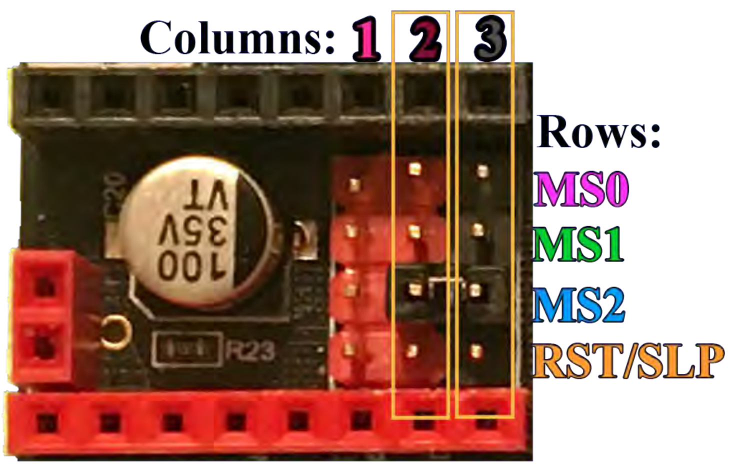
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 9 9 → Driver Chip Chart: **High**

Meaning:

Driver Chip Chart:

High → set Jumper between column 2 and column 3 on the MS2 row

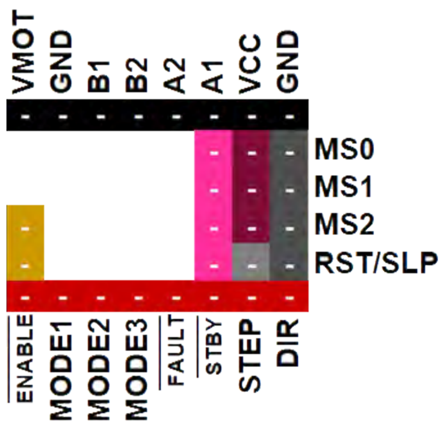
Low → No Jumper Set

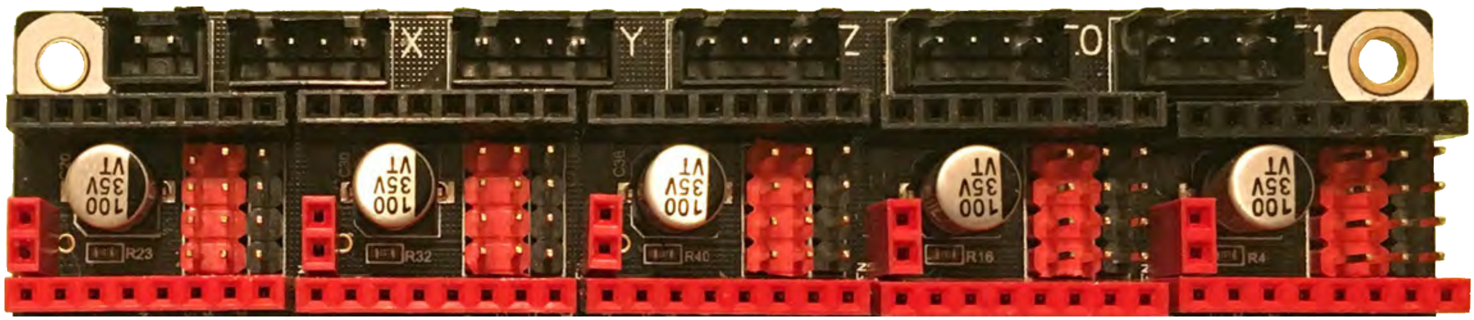
Stand-alone Mode

POLOLU ST820 (STSPIN820)

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

STEP

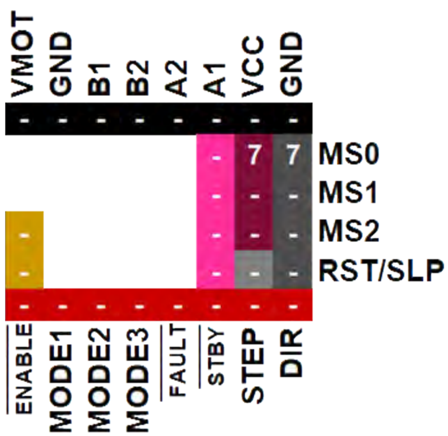


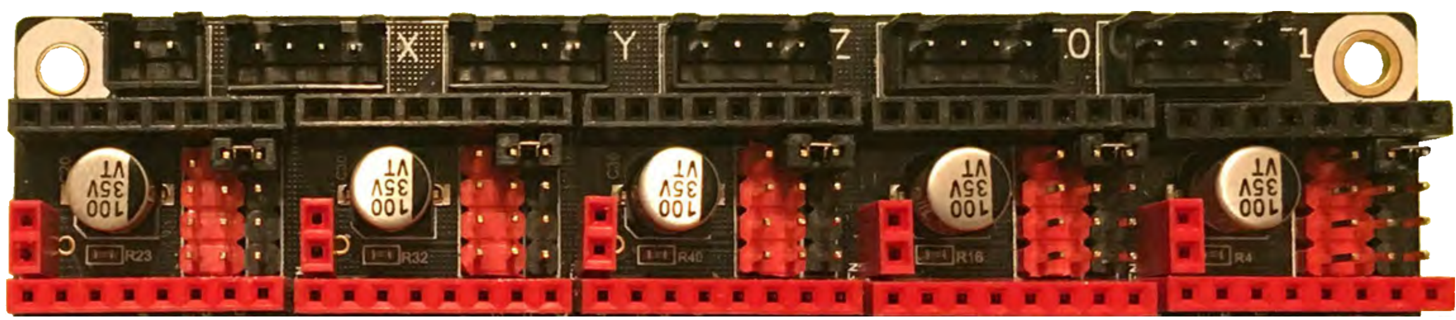


See Appendix D for legend

---

1 / 2





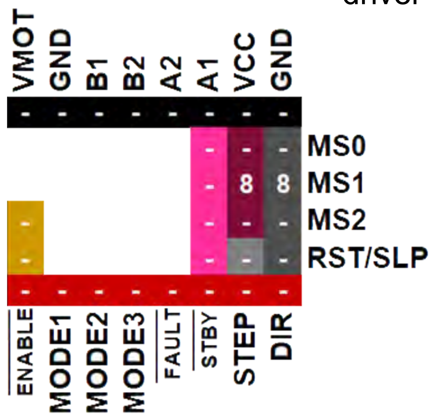
See Appendix D for legend

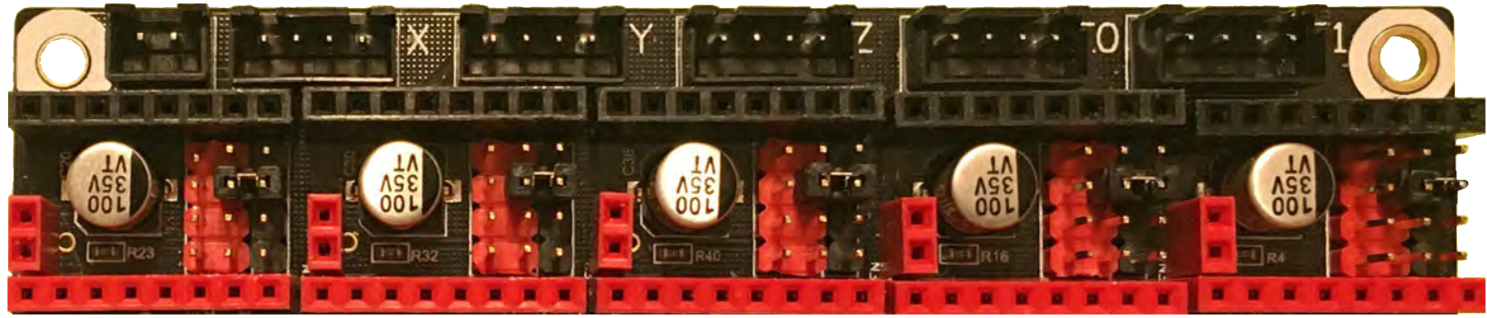
Stand-alone Mode

POLOLU ST820 (STSPIN820)

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

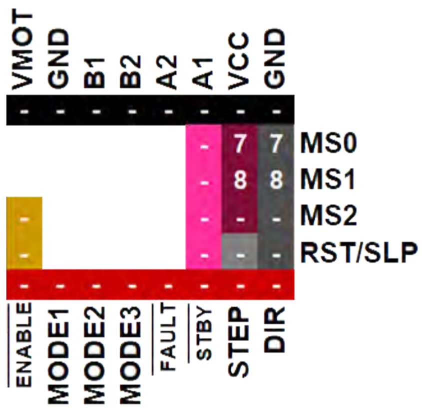
1 / 4

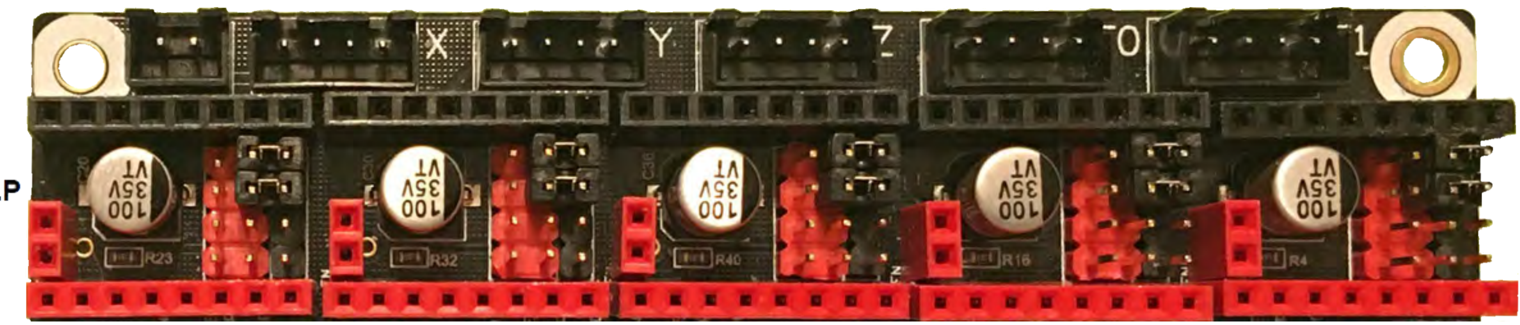




See Appendix D for legend

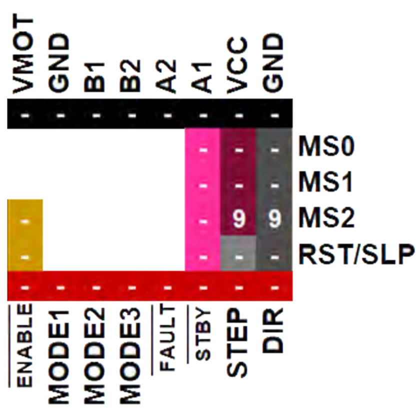
1 / 8

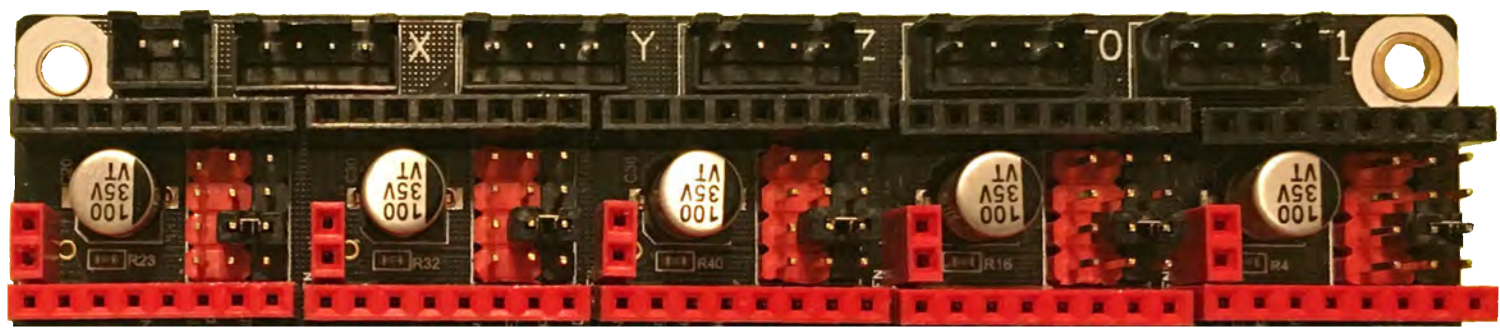




See Appendix D for legend

1 / 16





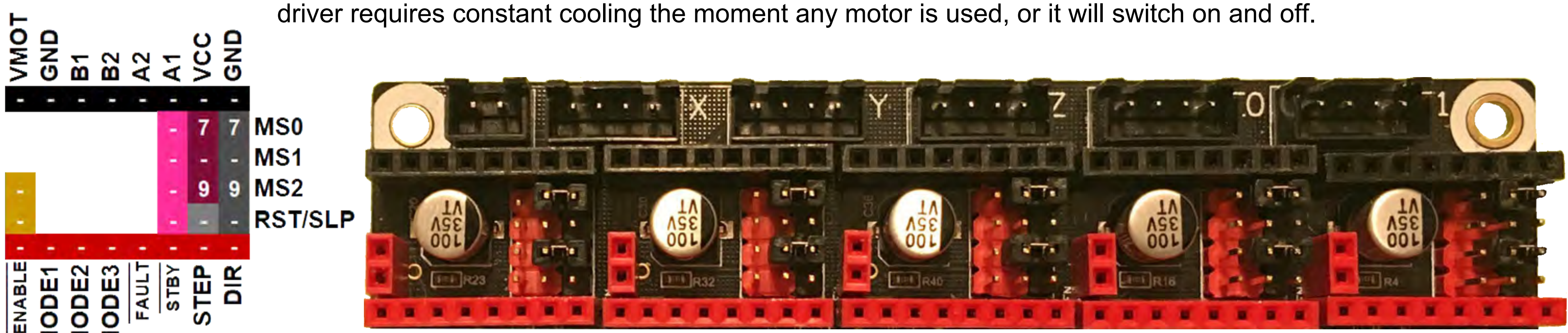
See Appendix D for legend

Stand-alone Mode

POLOLU ST820 (STSPIN820)

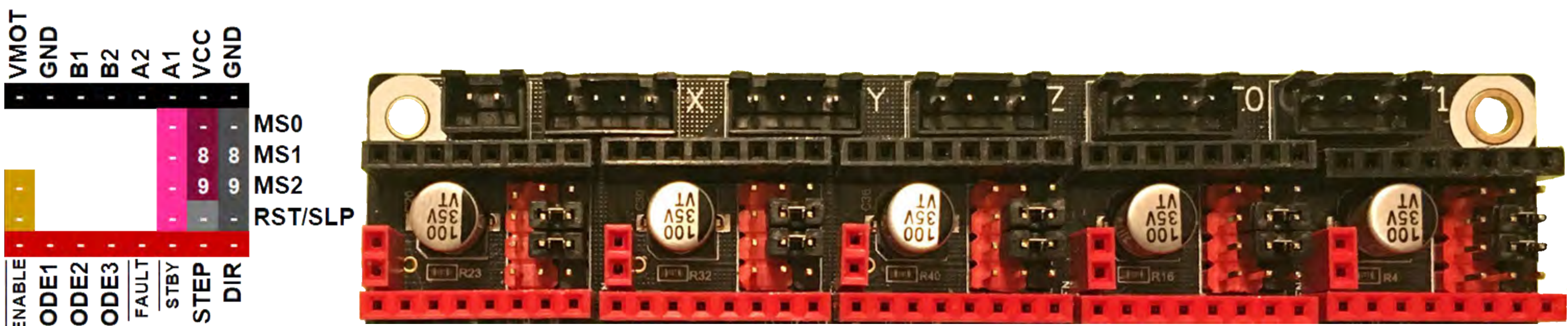
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 32



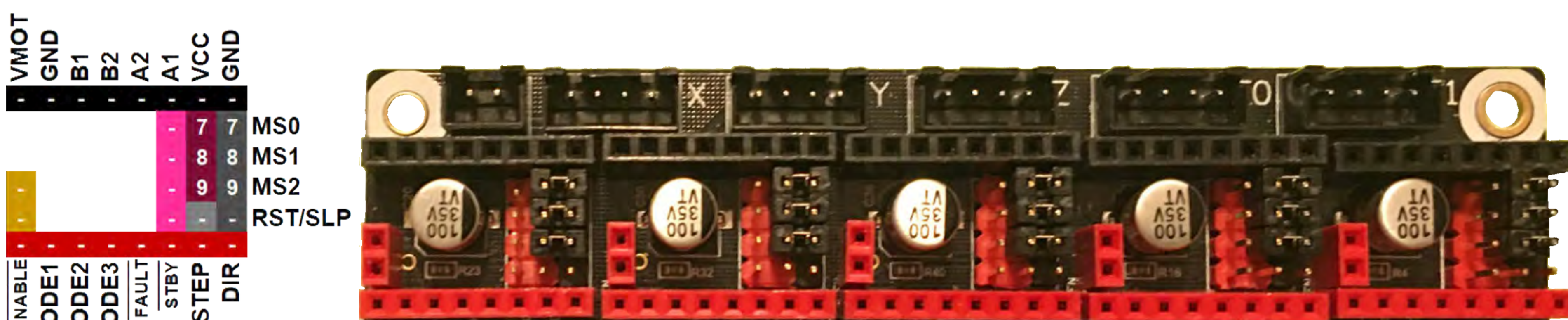
See Appendix D for legend

1 / 128



See Appendix D for legend

1 / 256

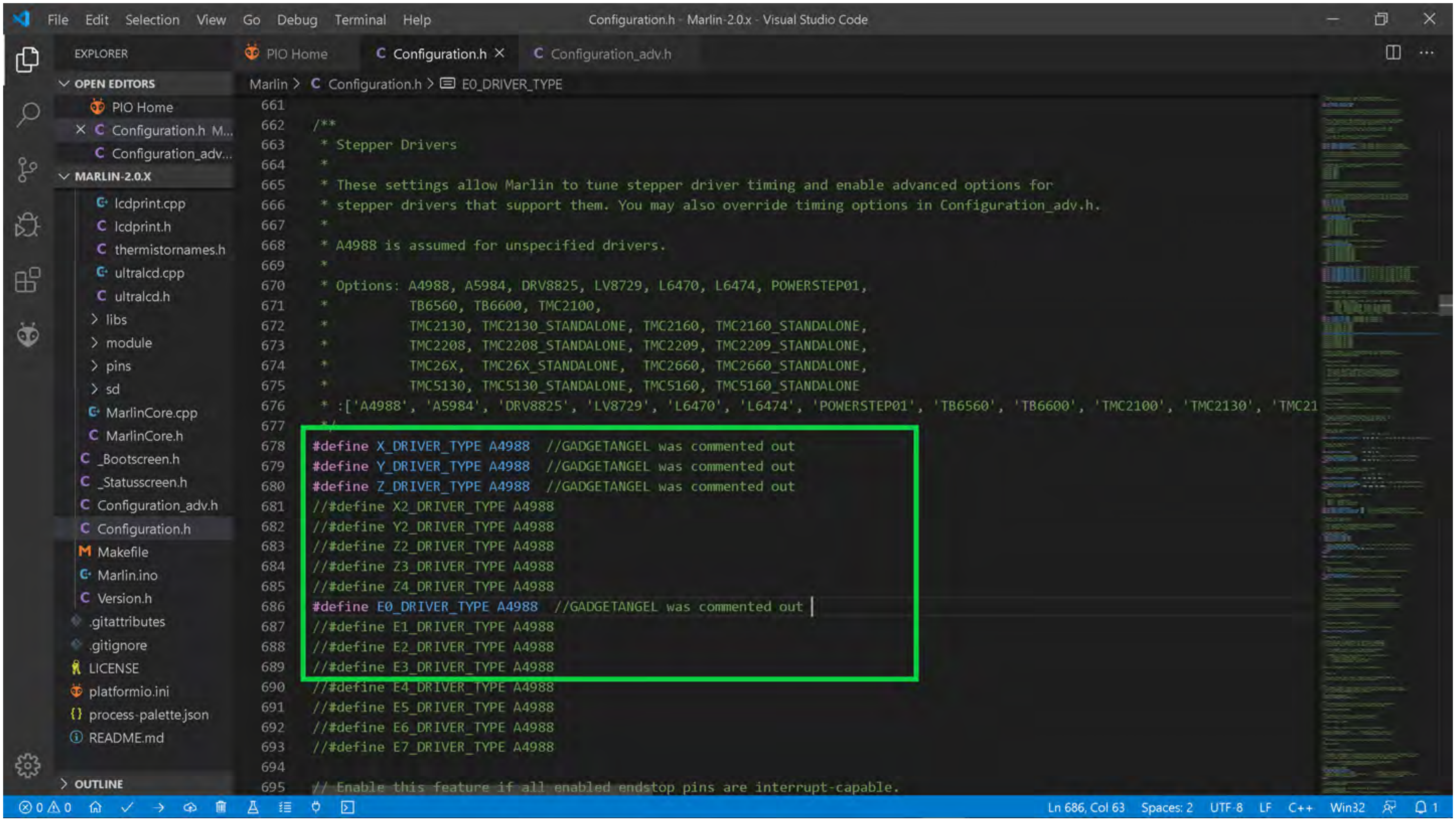


See Appendix D for legend

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

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

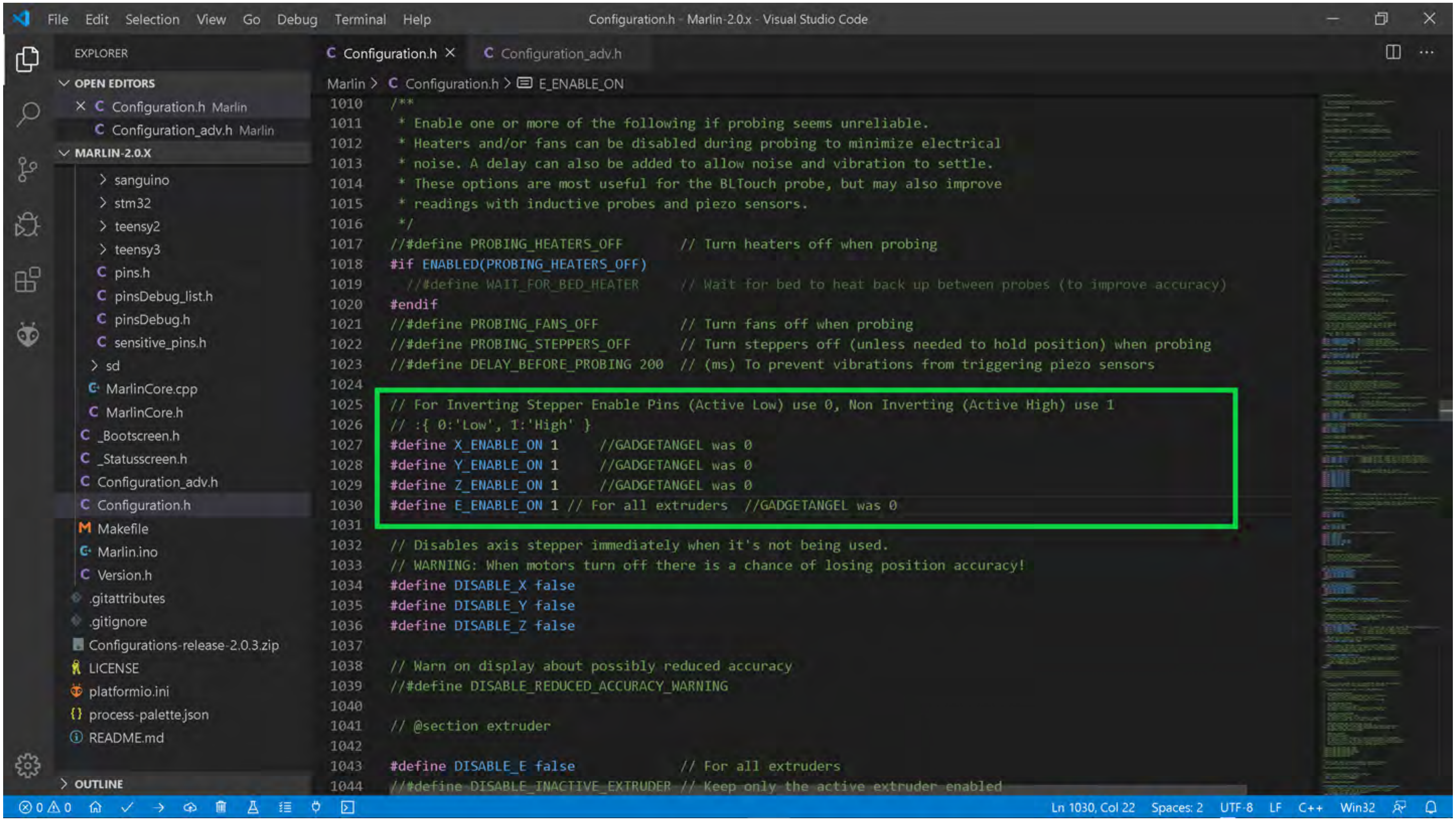
- Change the stepper motor drivers so that Marlin knows you are using POLOLU ST820 (STSPIN820) drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU ST820 (STSPIN820) drivers. When two "//" appear at the beginning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



- Go to the next page.

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

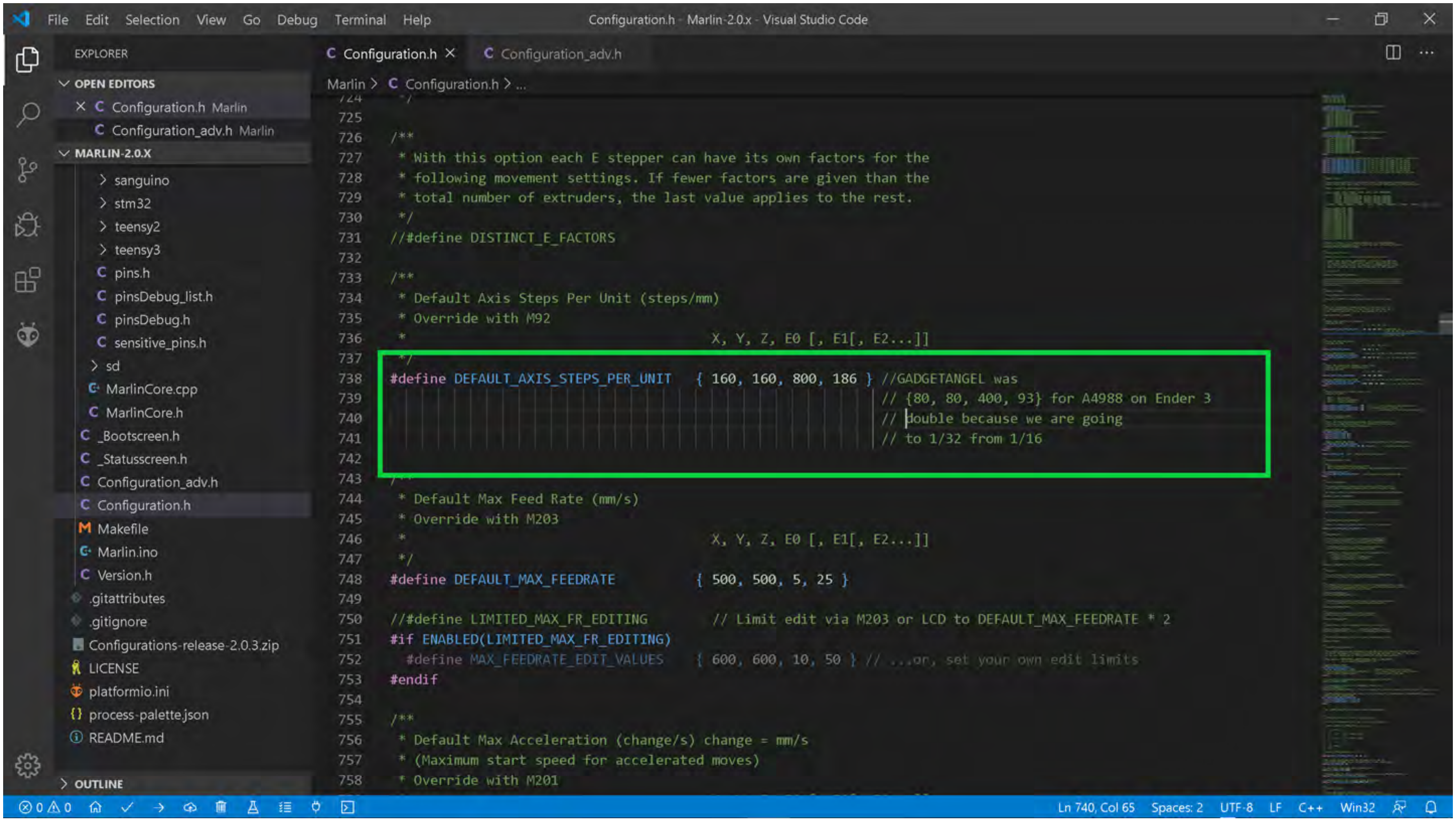
- In the Marlin firmware, the ST820 drivers needs an ACTIVE HIGH for the stepper motor driver's enable pin, so set "X_ENABLE_ON" to 1, "Y_ENABLE_ON" to 1, "Z_ENABLE_ON" to 1 and "E_ENABLE_ON" to 1, as seen in the **GREEN** box below.



- Go to the next page.

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

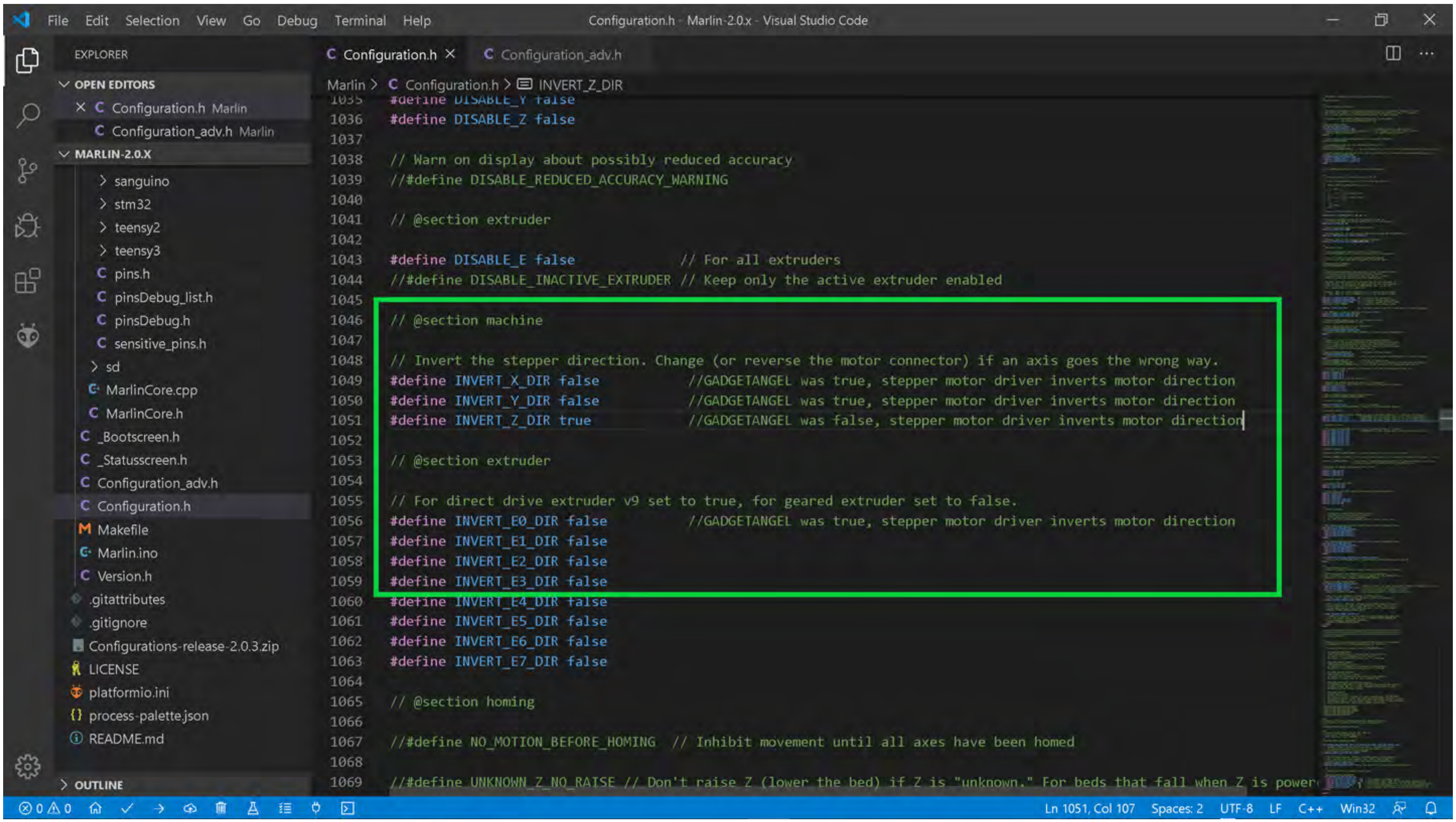
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to POLOLU ST820 (STSPIN820) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT"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



- Go to the next page.

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

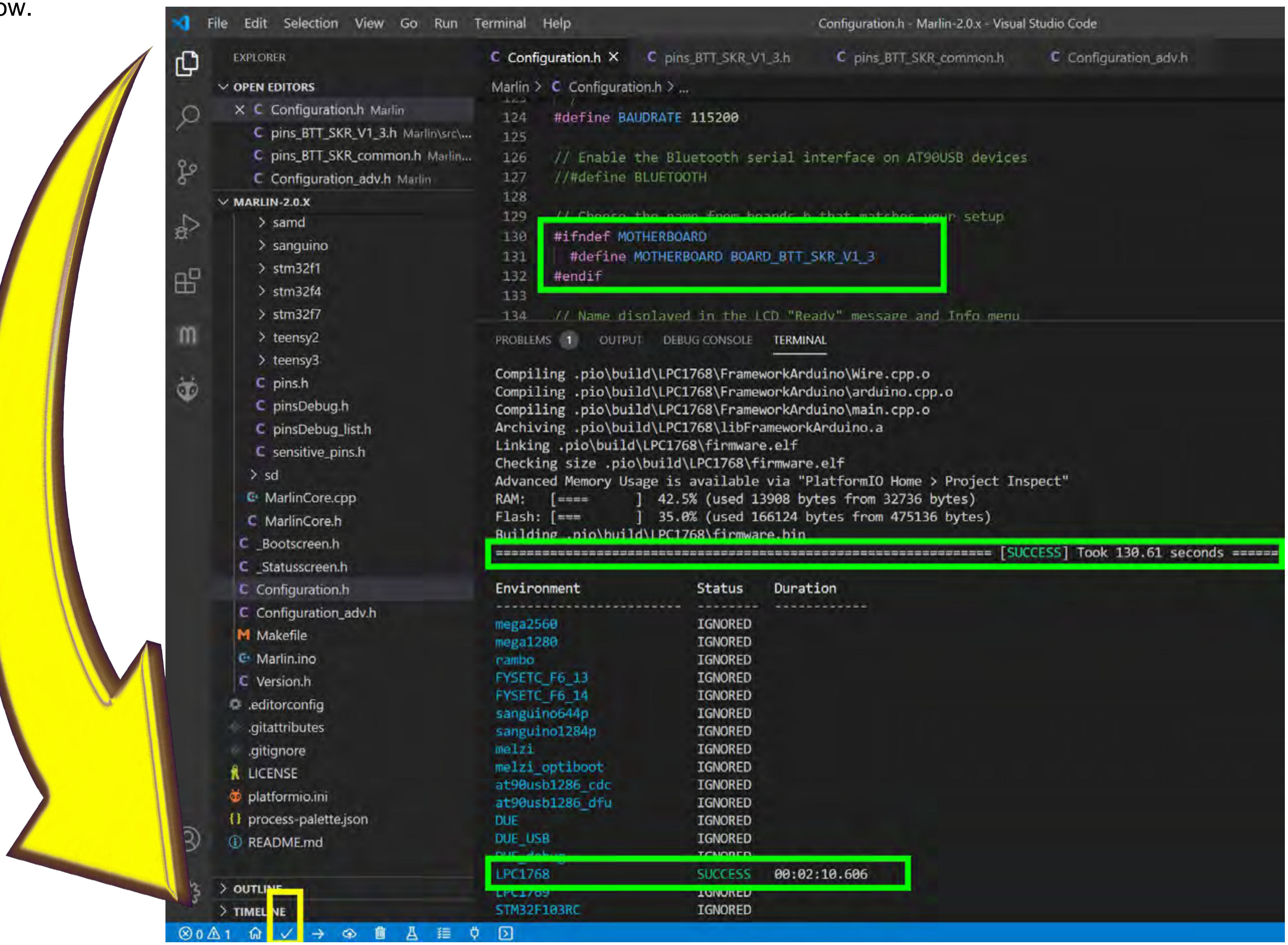
- Optional Step: I found conflicting information on the ST820 driver. Some sources say you will need to change the motor direction others say you may not. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring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



- Go to the next page.

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

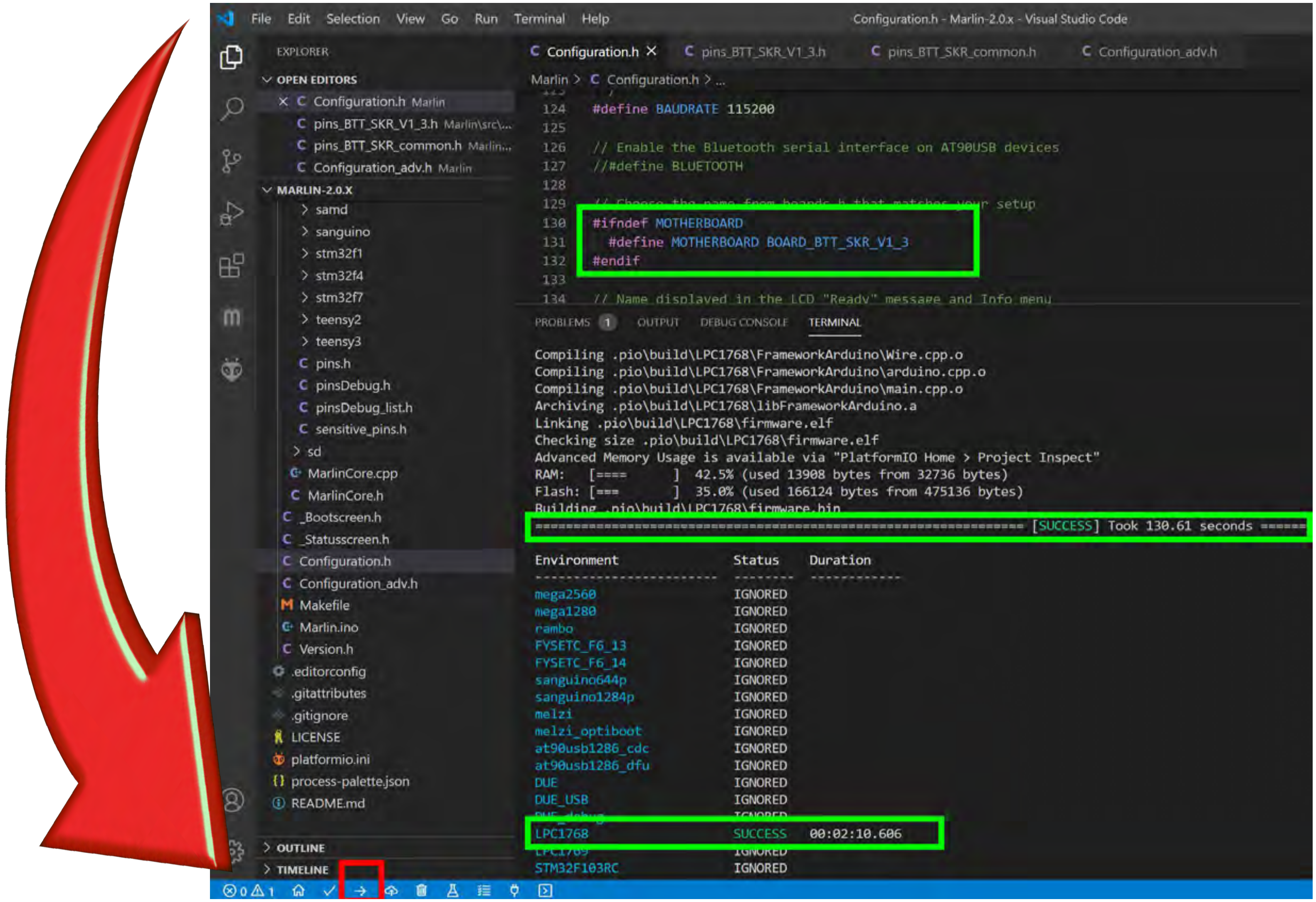
- The end of Marlin setup for POLOLU ST820 (STSPIN820) drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

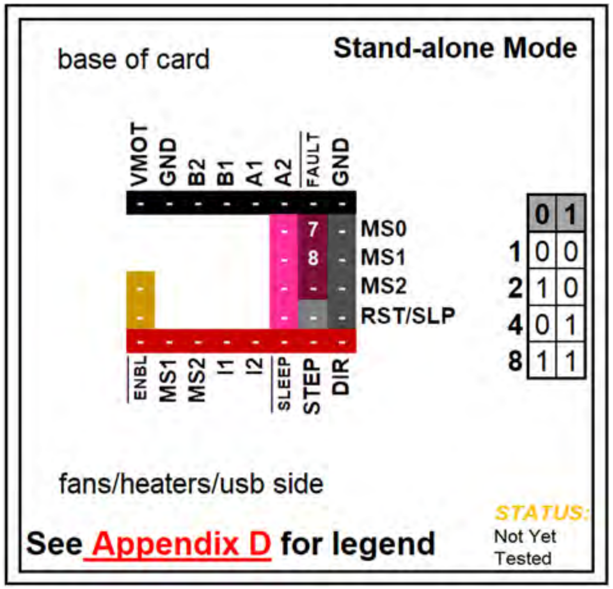
The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

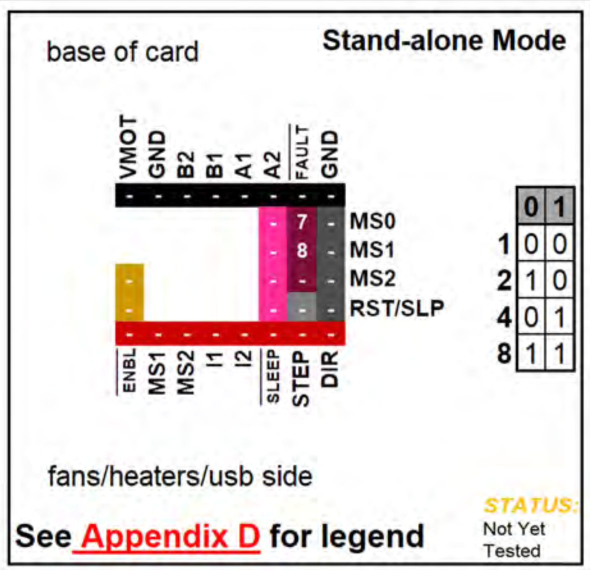
POLOLU MP6500



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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**



See Appendix D for legend

POLOLU MP6500

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low → No Jumper

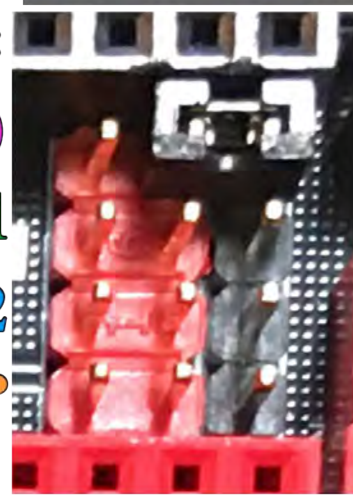
Rows:

MS0
MS1
MS2
RST/SLP

High → Jumper Set

Rows:

MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

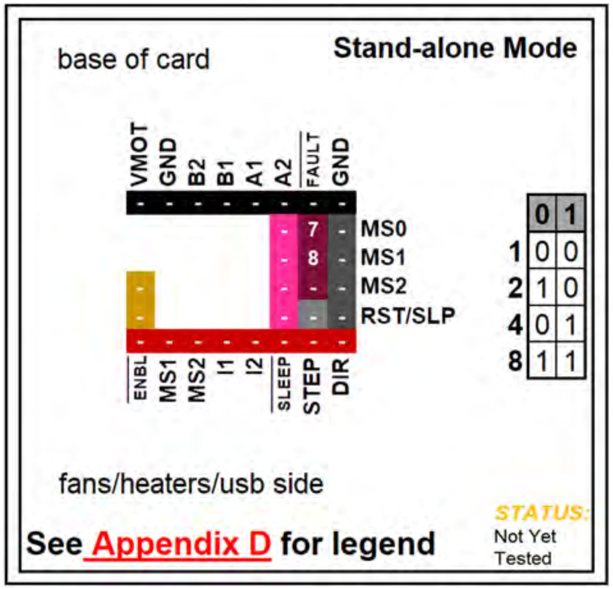
Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC x.xA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

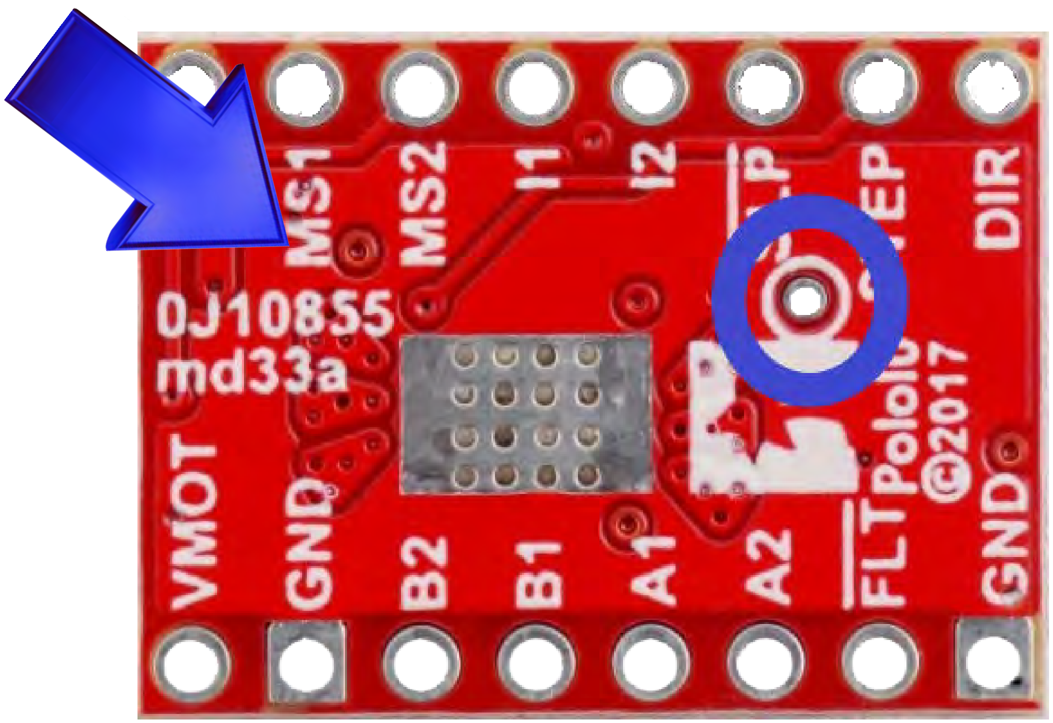


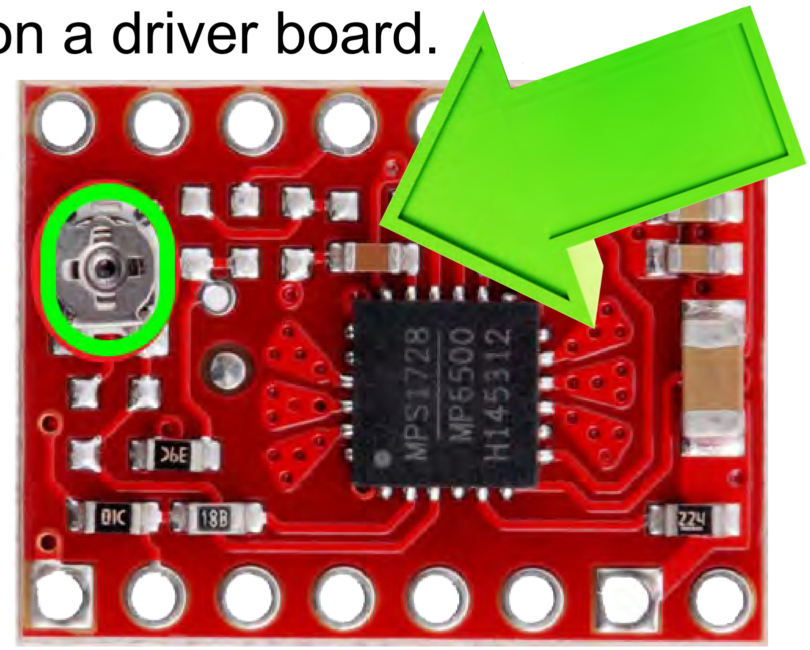
POLOLU MP6500

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

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

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





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

Stand-alone Mode

POLOLU MP6500

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

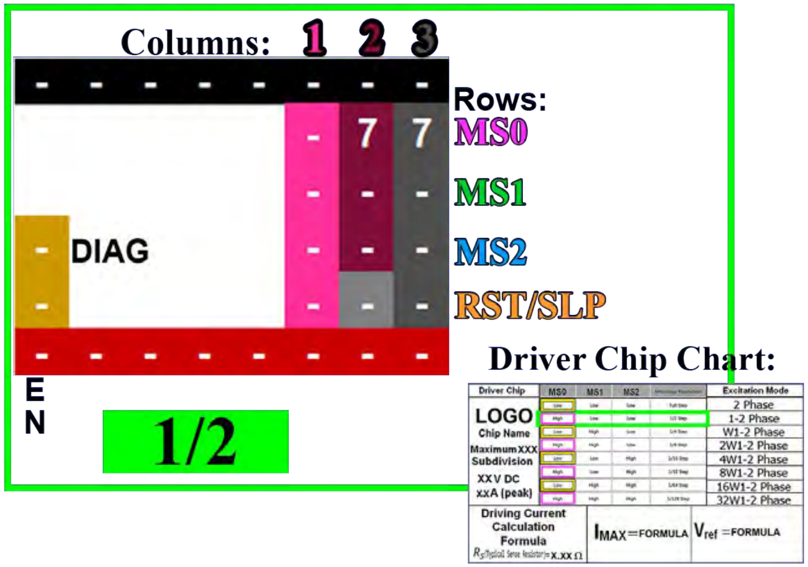
Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Stand-alone Mode

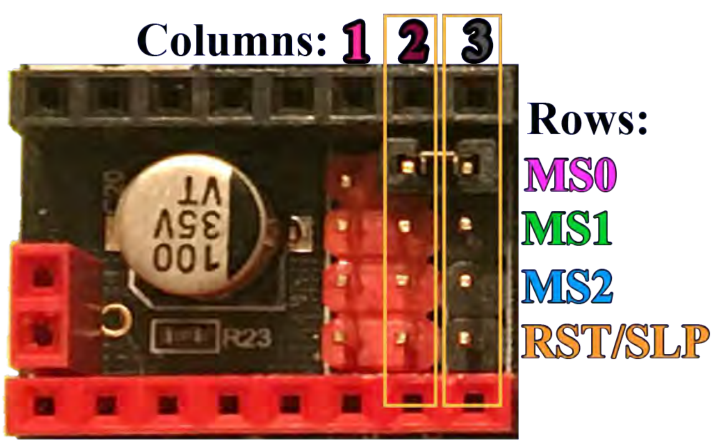
POLOLU MP6500

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

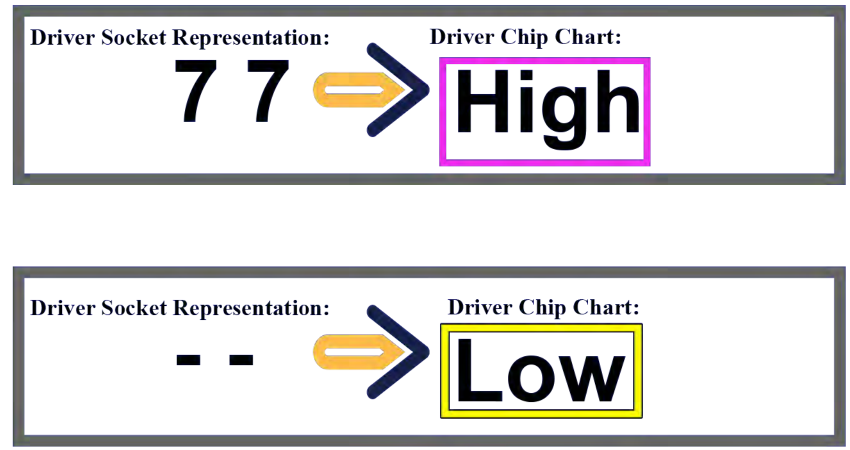
Driver Socket Representation



High:



MS0 for Binary State Drivers:

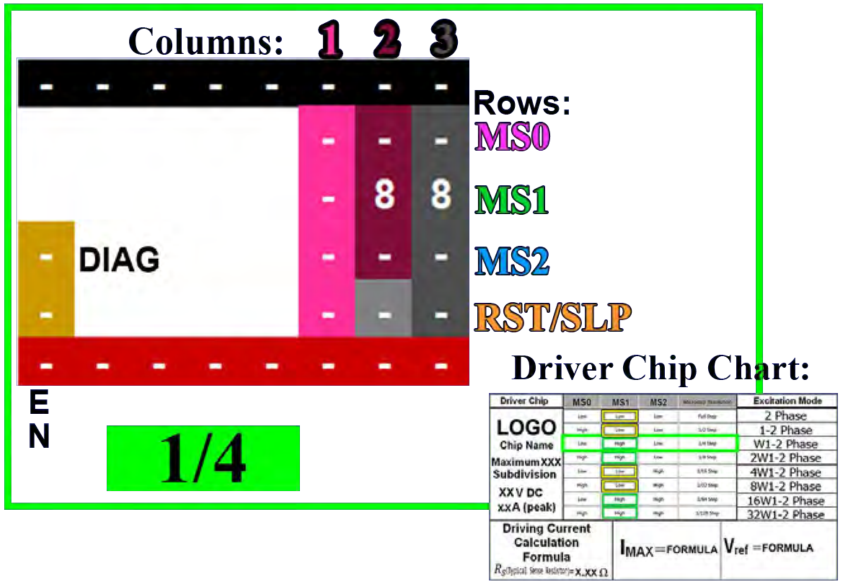


Driver Chip Chart:	Meaning:
High	set Jumper between column 2 and column 3 on the MS0 row
Low	No Jumper Set

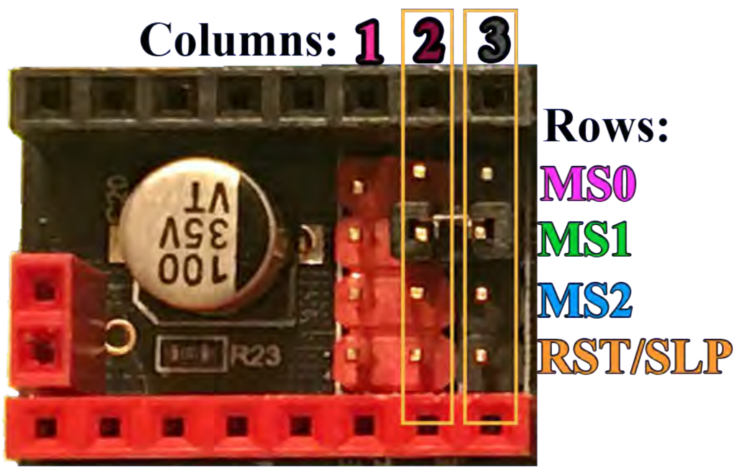
Stand-alone Mode

POLOLU MP6500

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

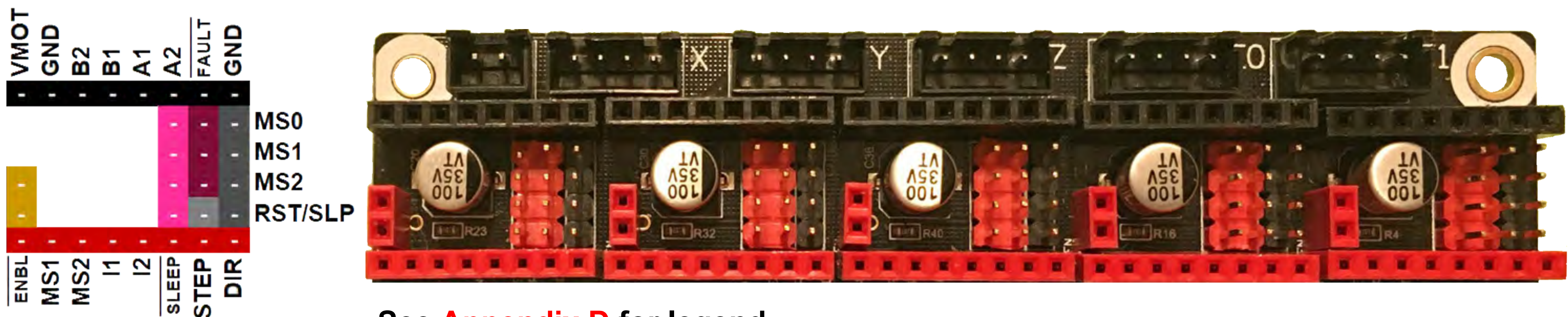
Meaning:

Driver Chip Chart:	Meaning:
High	set Jumper between column 2 and column 3 on the MS1 row
Low	No Jumper Set

Stand-alone Mode

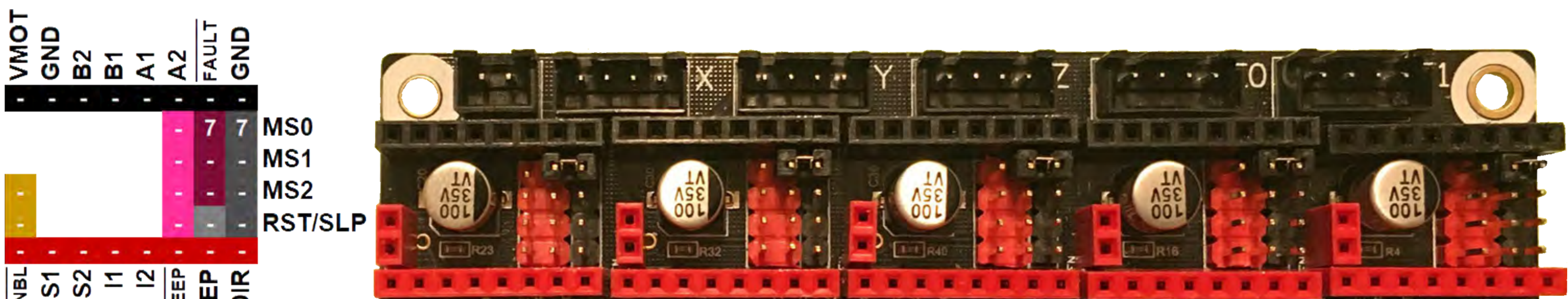
POLOLU MP6500

STEP



See Appendix D for legend

1 / 2

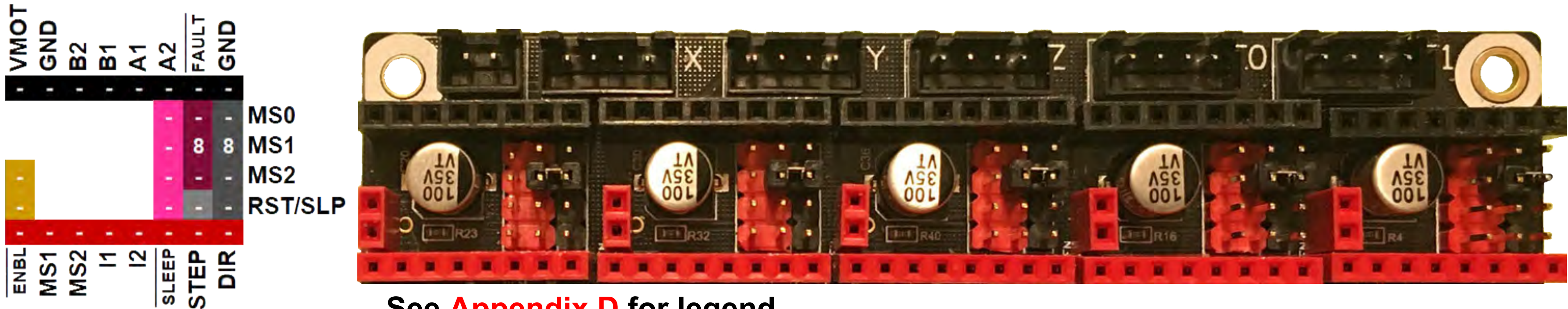


See Appendix D for legend

Stand-alone Mode

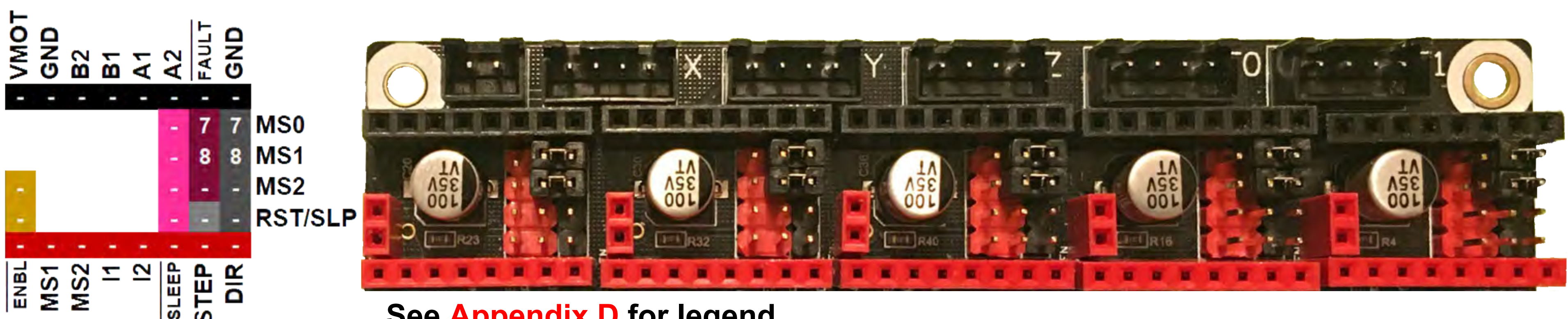
POLOLU MP6500

1 / 4



See Appendix D for legend

1 / 8

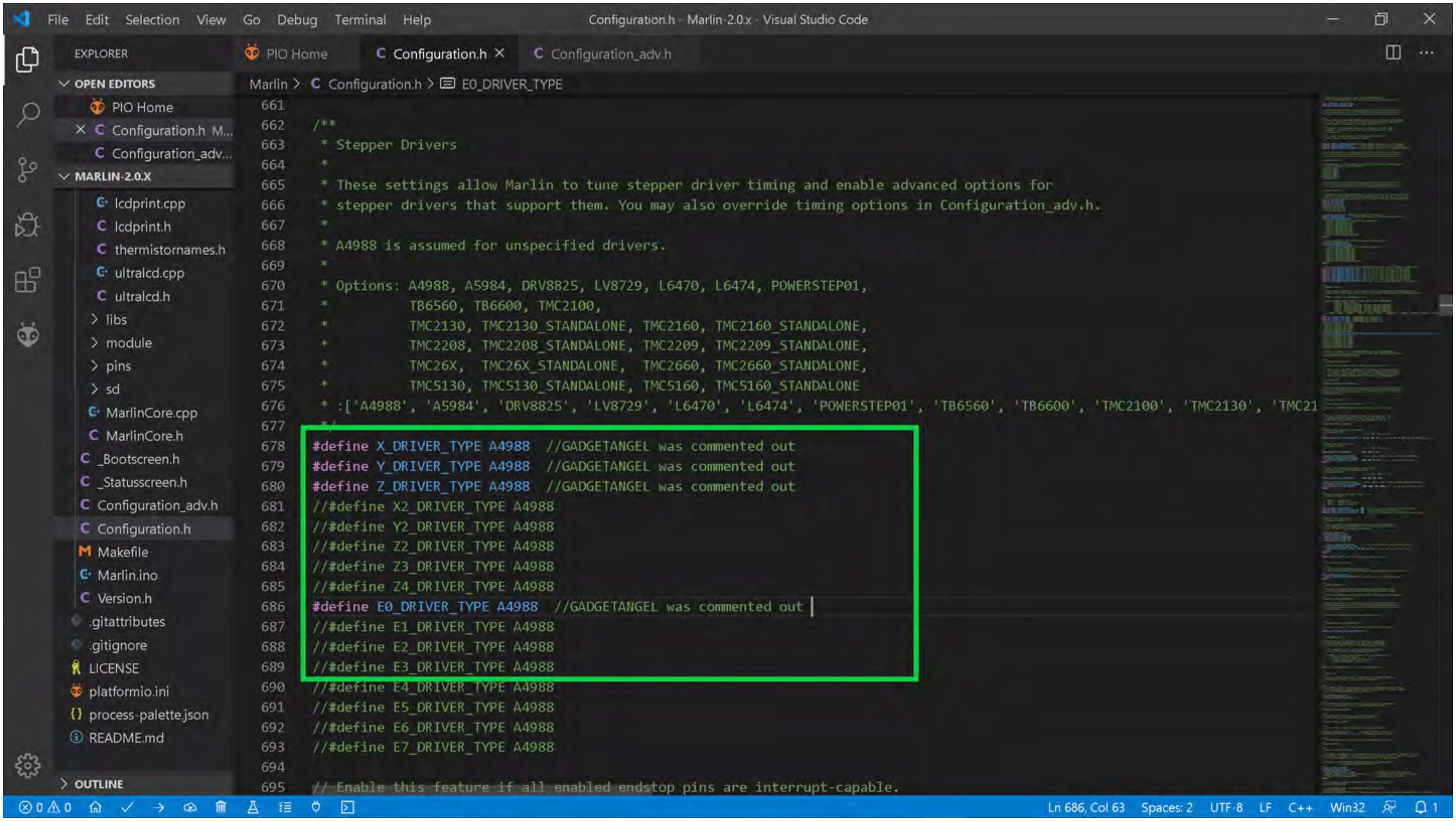


See Appendix D for legend

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

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

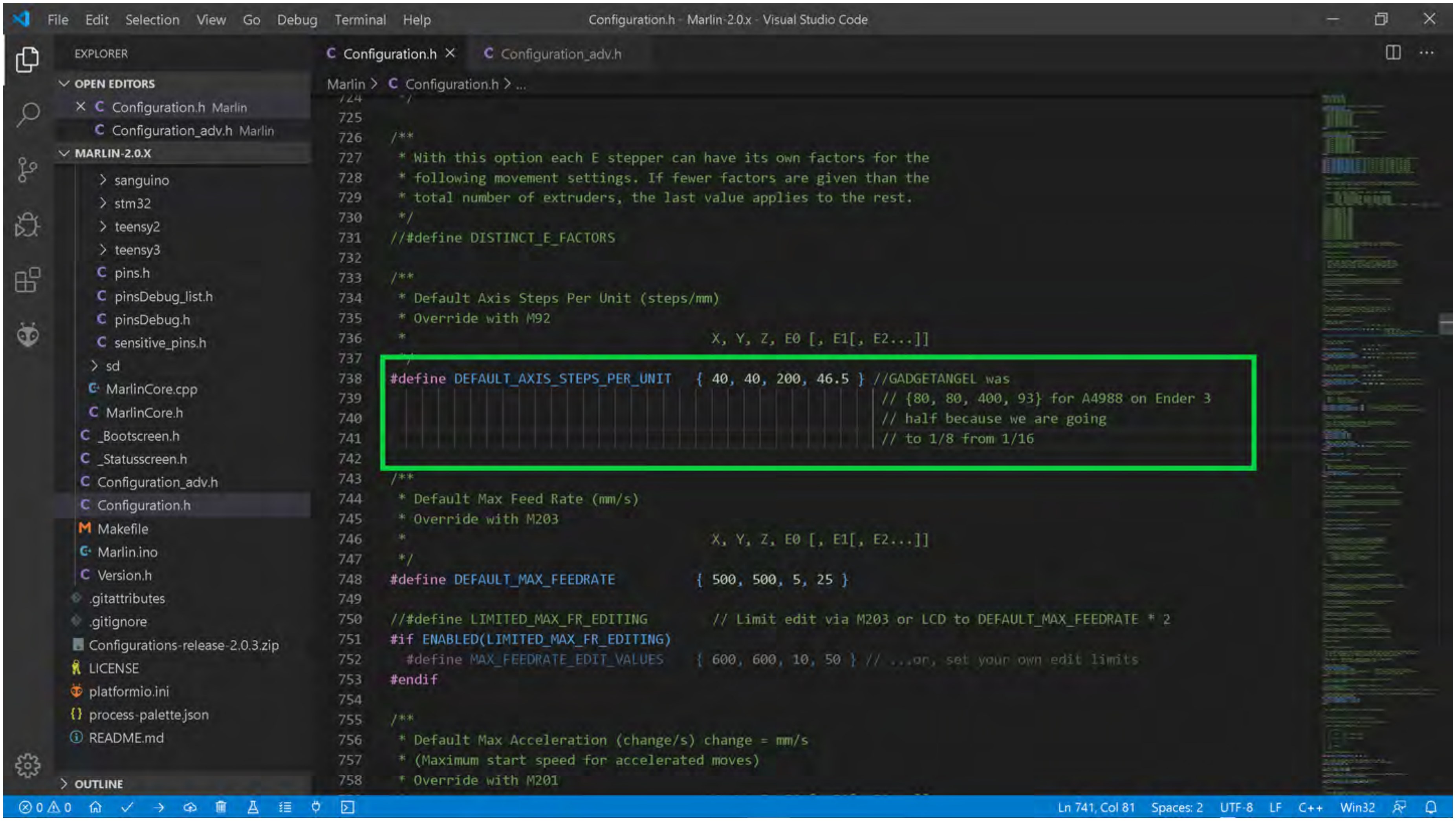
- Change the stepper motor drivers so that Marlin knows you are using POLOLU MP6500 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU MP6500 drivers. When two "//" appear at the beginning of a line that means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



- Go to the next page.

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

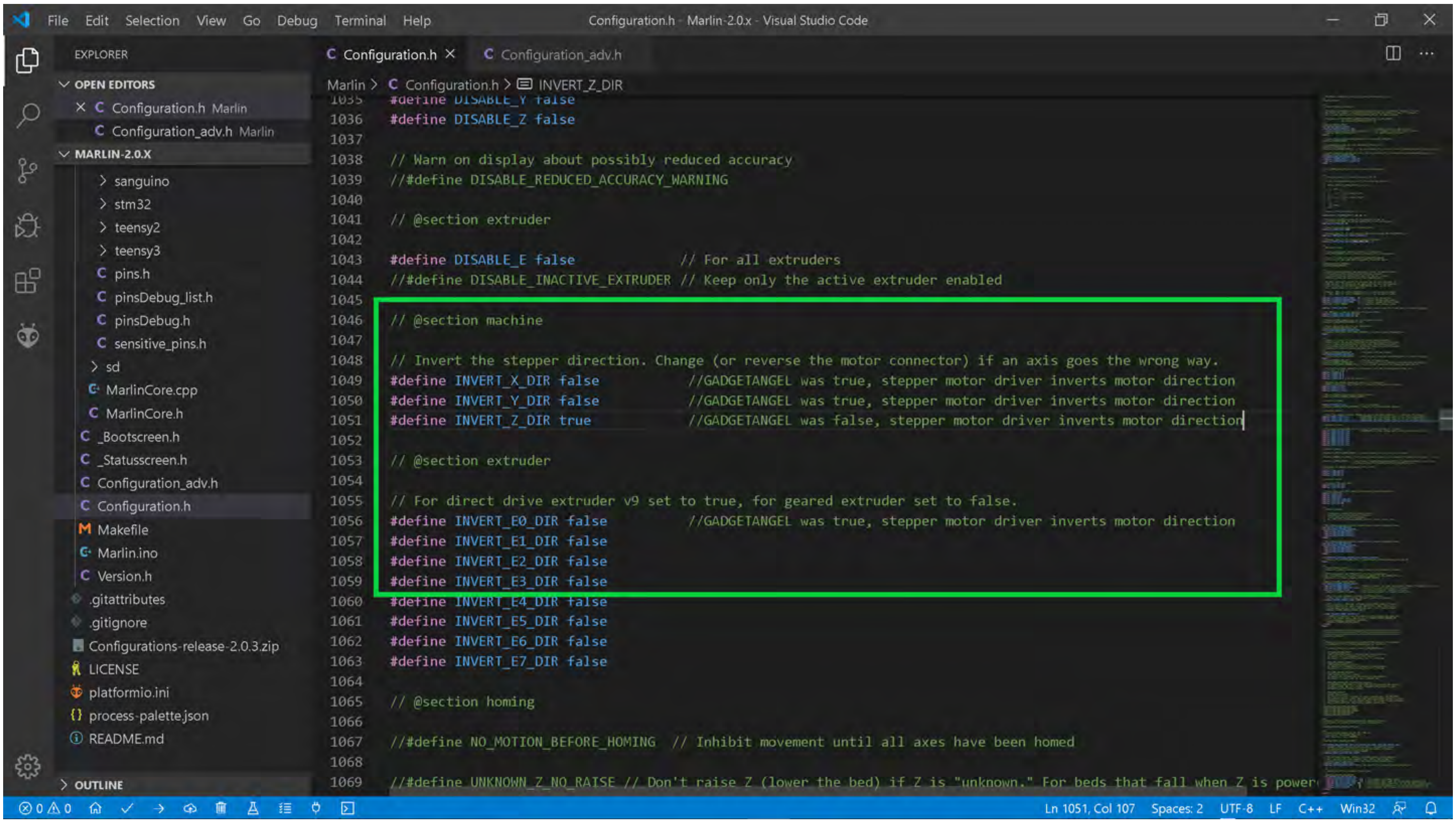
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for POLOLU MP6500 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/8 stepping. So we are cutting our STEPS in half. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT"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



- Go to the next page.

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

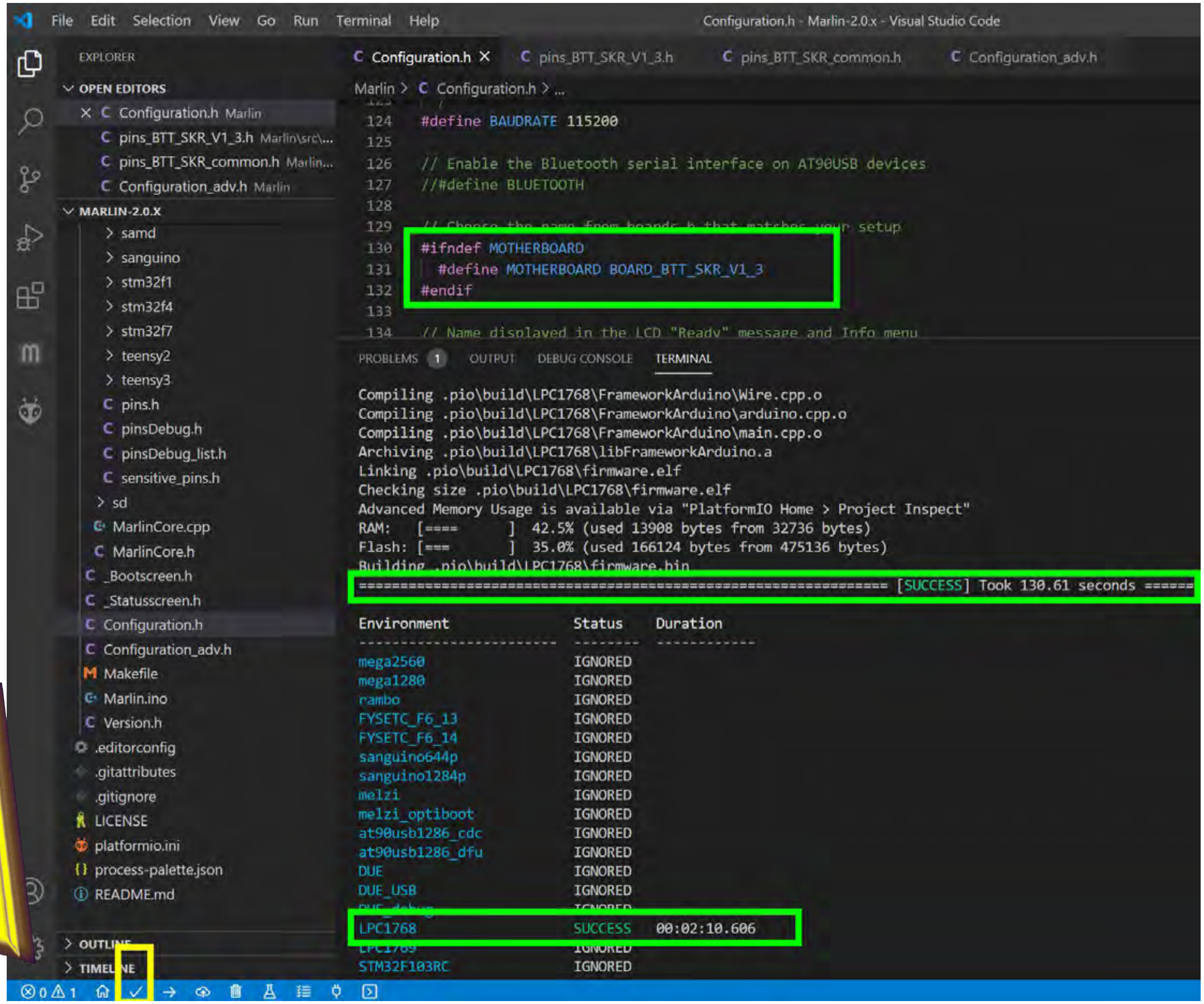
- Optional Step: I cannot find information on the POLOLU MP6500 driver's impact on motor direction. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") **https://reprap.org/wiki/Stepper_wiring**. Other people prefer to change the motor direction in the Marlin firmware. So if you want or need to change the motor direction in Marlin, then if the axis' setting you will be using the MP6500 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

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

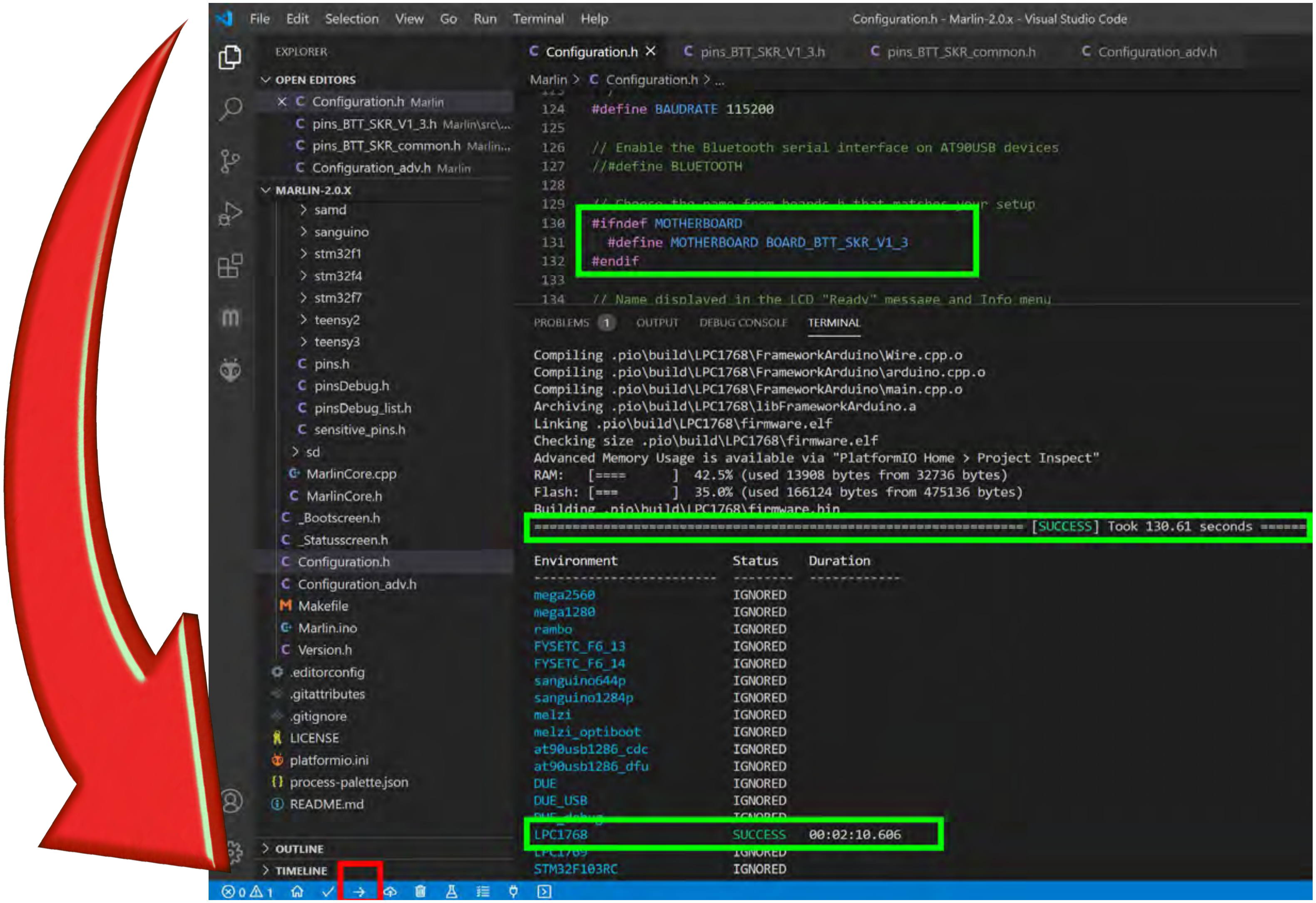
- The end of Marlin setup for POLOLU MP6500 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

POLOLU TB67S249FTG

Stand-alone Mode

base of card

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

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution
Pololu TB67S249FTG Maximum 32 Subdivision 47V DC 4.5A (peak)	Low	Low	Low	Standby mode (outputs disabled)
	Low	Low	High	Full step
	Low	High	Low	Non-circular half step ("a")
	Low	High	High	1/4 step
	High	Low	Low	Circular half step ("b")
	High	Low	High	1/8 step
	High	High	Low	1/16 step
	High	High	High	1/32 step
Driving Current Calculation Formula	$I_{MAX} = V_{ref} * 1.25$			$V_{ref} = \frac{I_{MAX}}{1.25}$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

POLOLU TB67S249FTG

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Stand-alone Mode

base of card

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

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Low ⇒ No Jumper

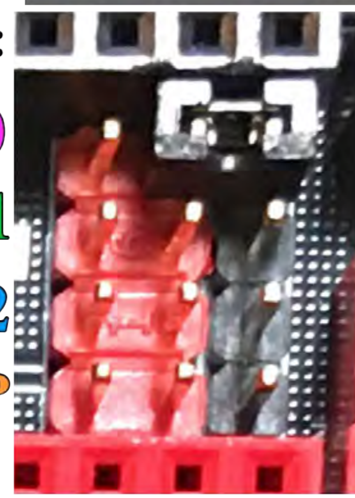
Rows: MS0 MS1 MS2 RST/SLP

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC x.xA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S(Typical Sense Resistor)= x.xx Ω	I_{MAX}=FORMULA		V_{ref} =FORMULA		

High ⇒ Jumper Set

Rows: MS0 MS1 MS2 RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

POLOLU TB67S249FTG

Stand-alone Mode

base of card

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

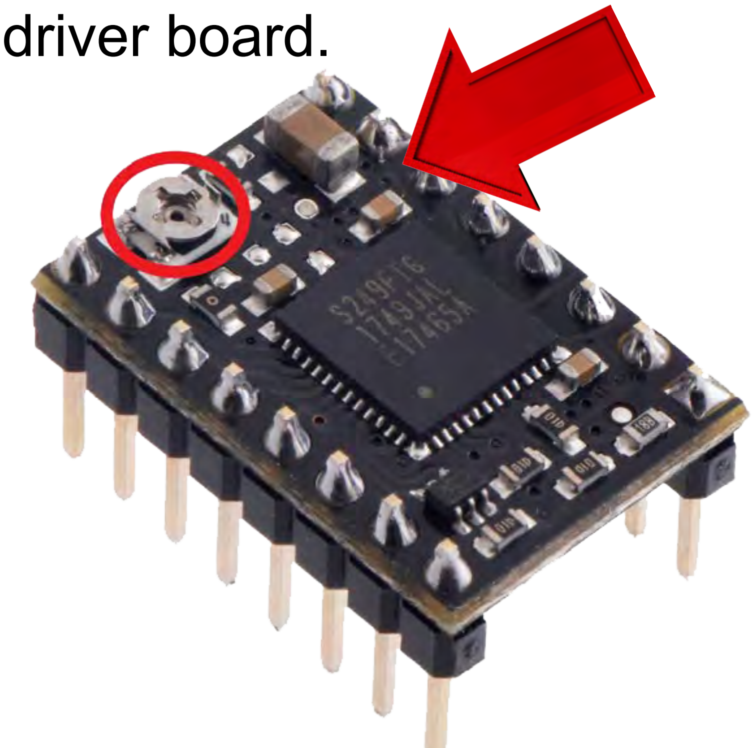
fans/heaters/usb side

STATUS: Not Yet Tested

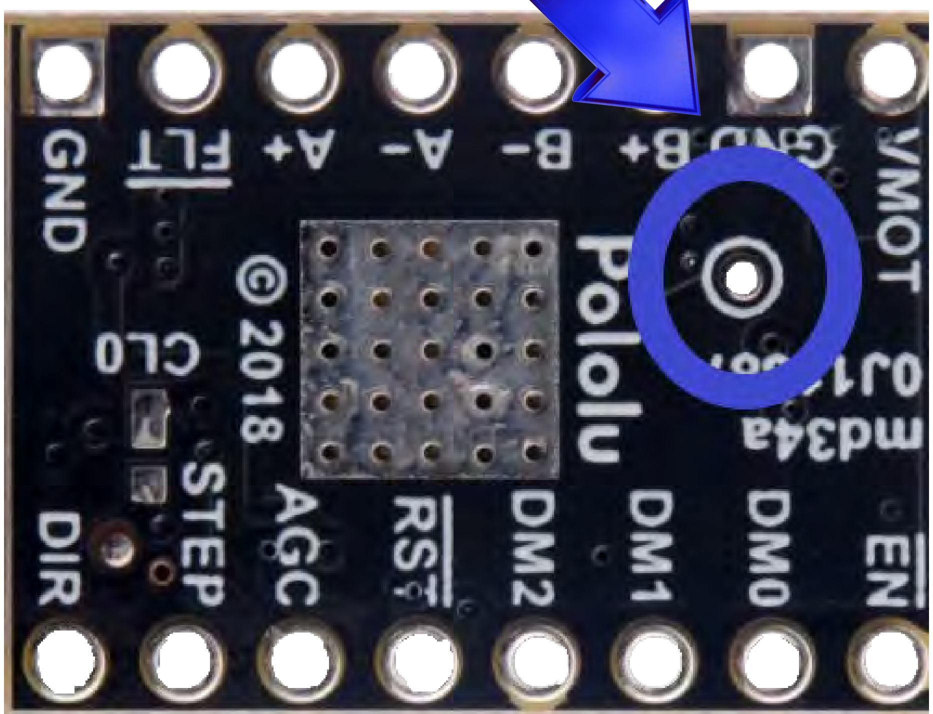
See Appendix D for legend

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

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



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



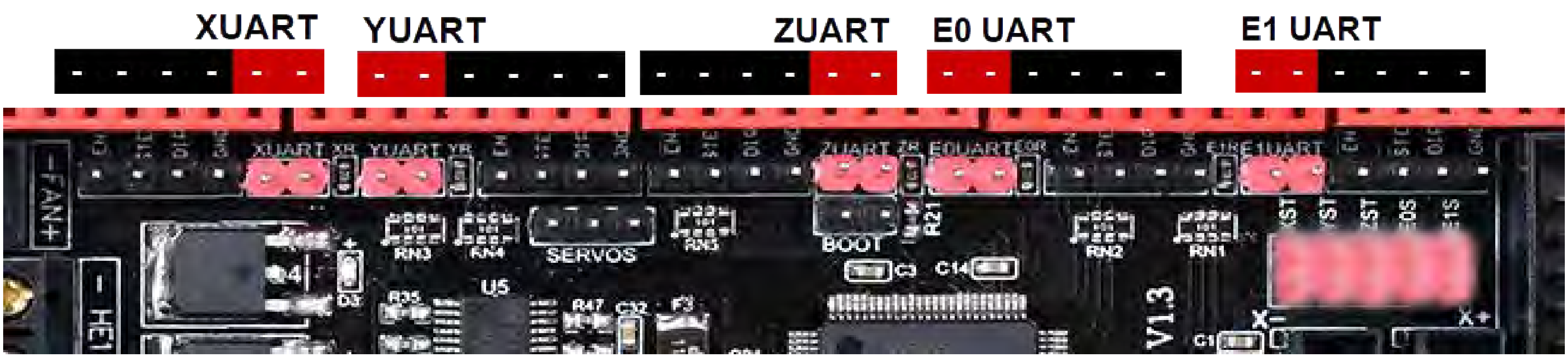
Note: **TB67S249FTG driver board does not use external current sense resistors (R_s).**

Stand-alone Mode

POLOLU TB67S249FTG

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

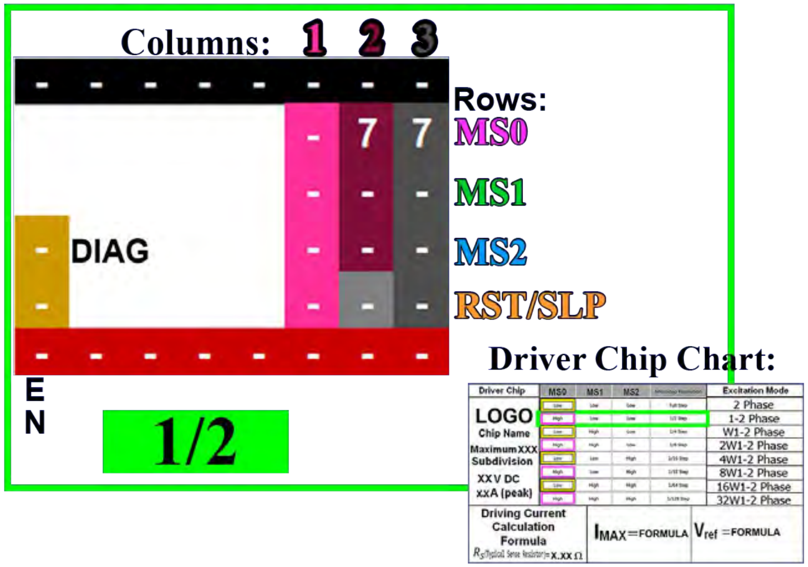


Stand-alone Mode

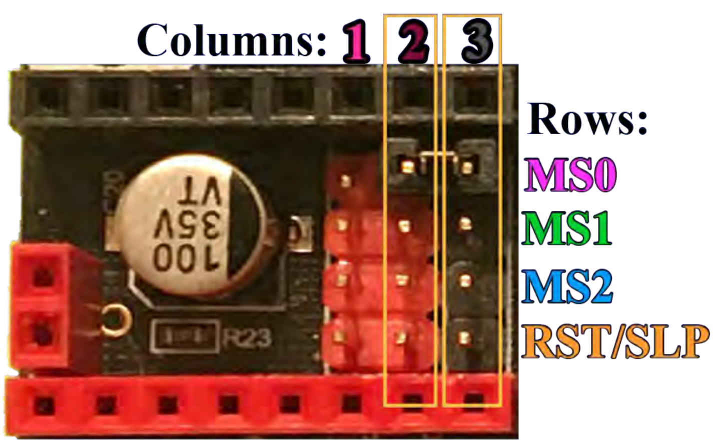
POLOLU TB67S249FTG

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

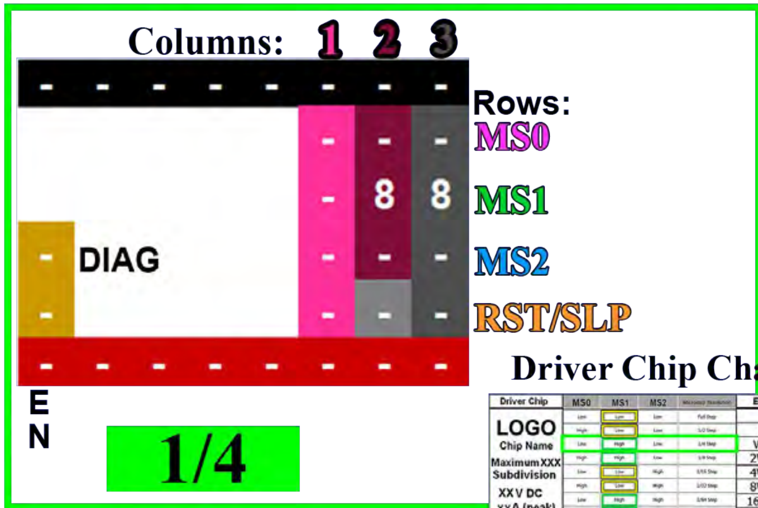
Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

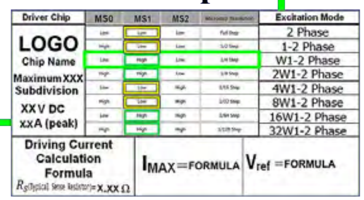
Stand-alone Mode

POLOLU TB67S249FTG

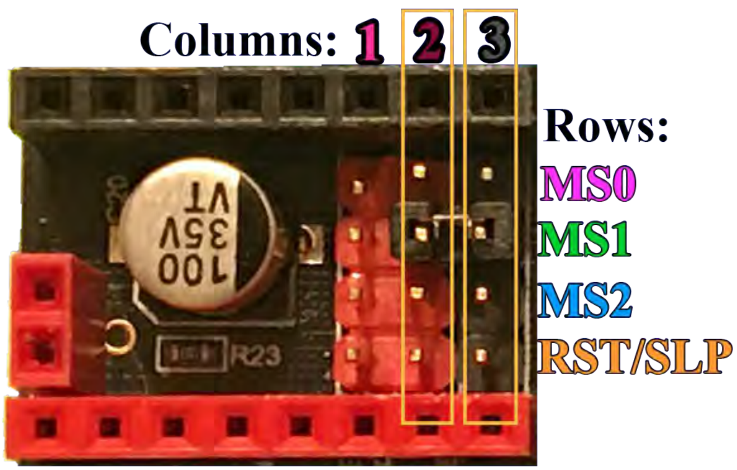
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

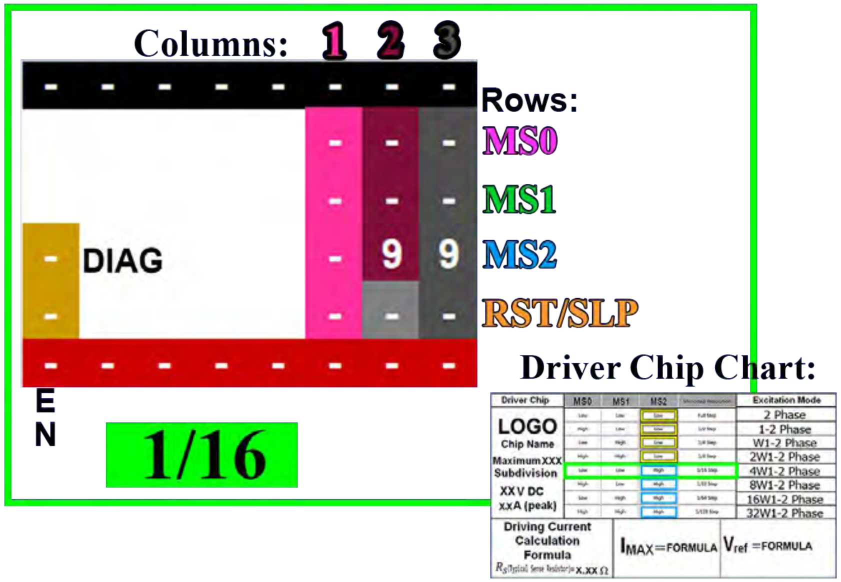
Meaning:

Driver Chip Chart:		
High	→	set Jumper between column 2 and column 3 on the MS1 row
Low	→	No Jumper Set

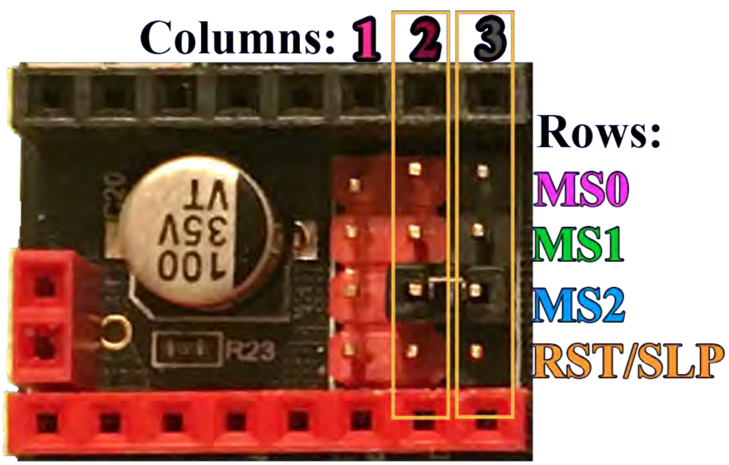
Stand-alone Mode

POLOLU TB67S249FTG

Driver Socket Representation



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 9 9 → Driver Chip Chart: **High**

Driver Chip Chart: / **Meaning:**

High → set Jumper between column 2 and column 3 on the MS2 row

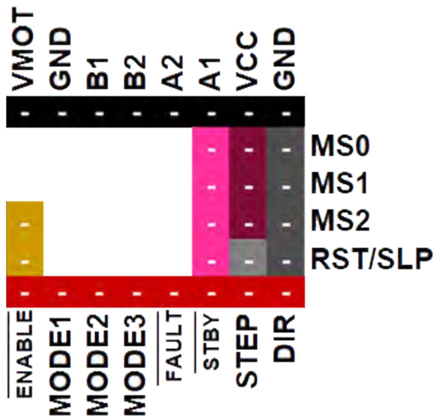
Low → No Jumper Set

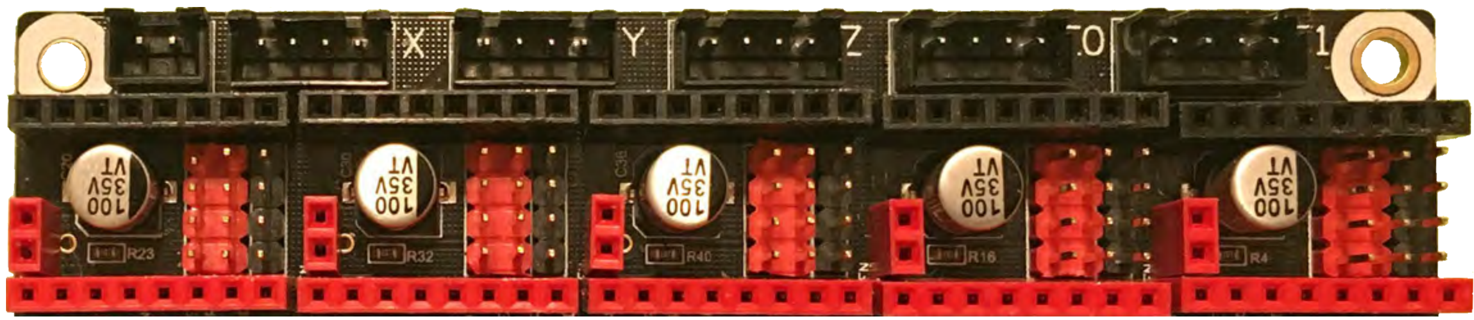
Stand-alone Mode

POLOLU TB67S249FTG

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

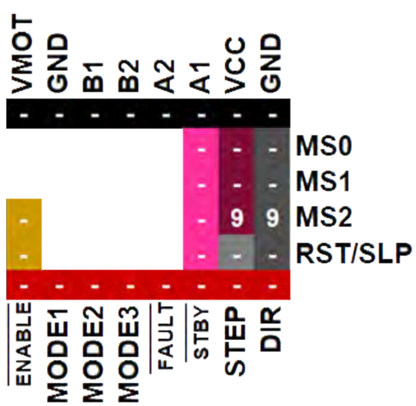
STAND BY

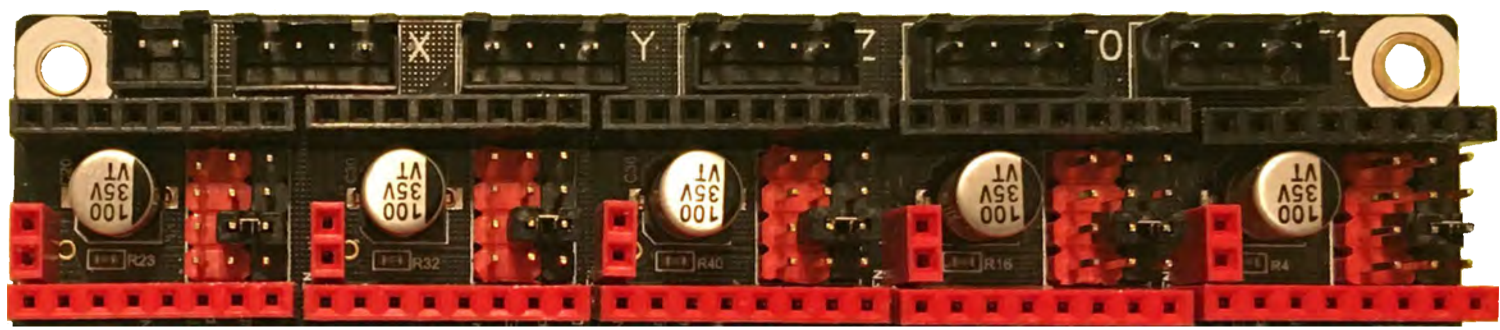




See Appendix D for legend

STEP





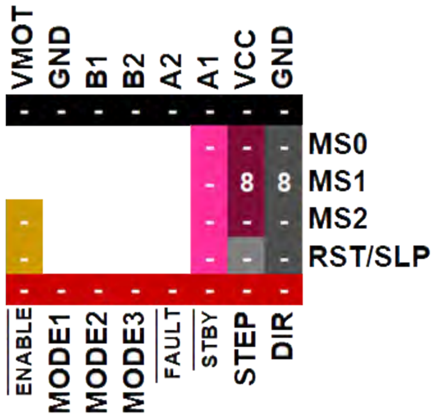
See Appendix D for legend

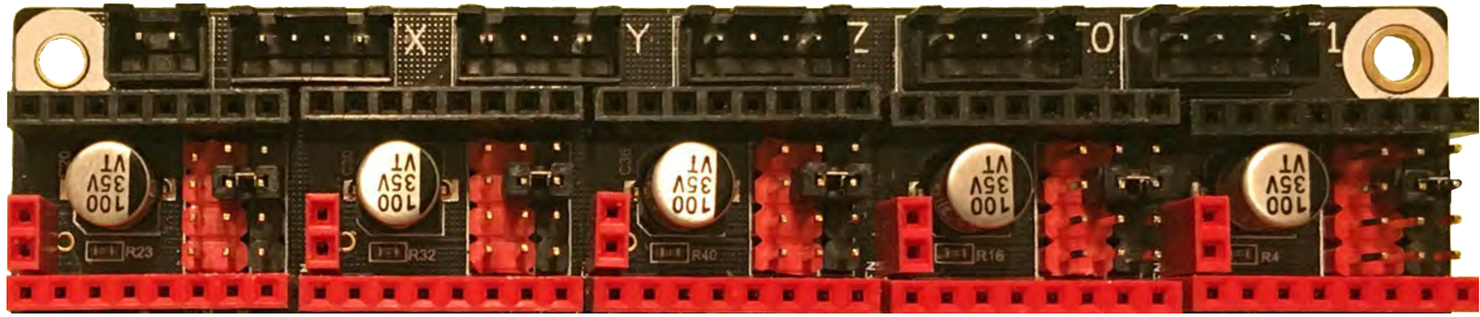
Stand-alone Mode

POLOLU TB67S249FTG

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

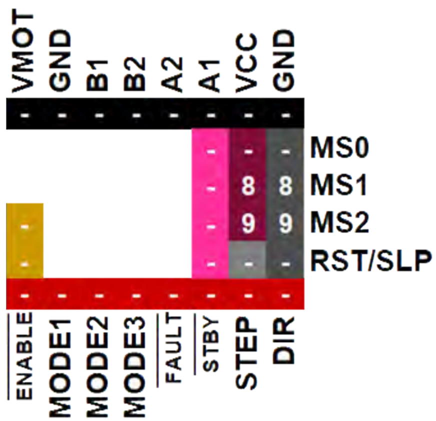
CCW
1 / 2

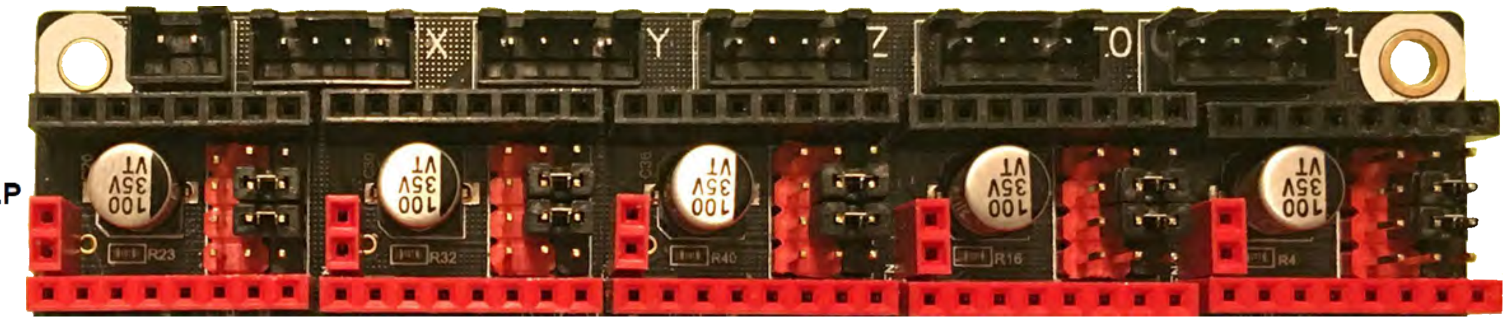




See Appendix D for legend

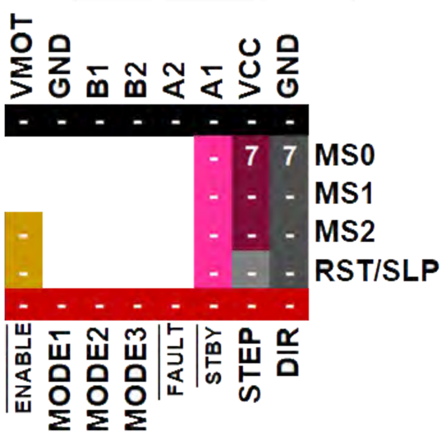
1 / 4

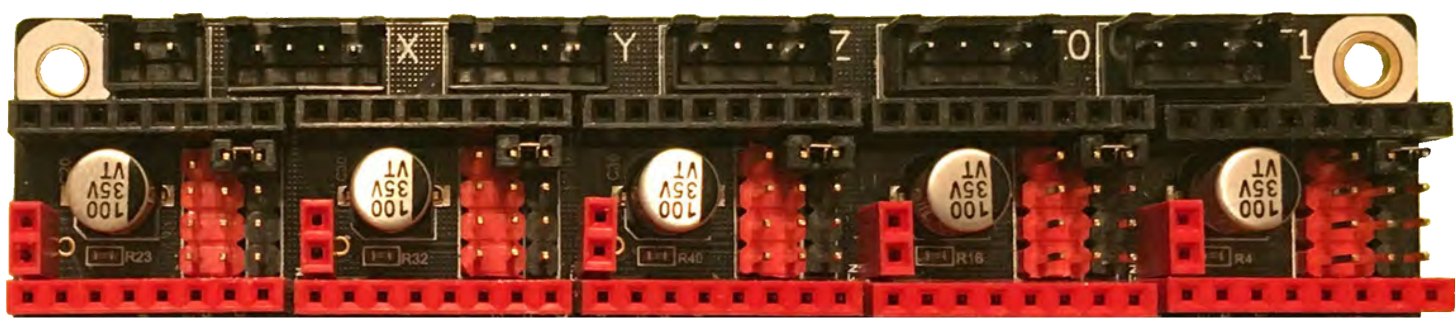




See Appendix D for legend

CW
1 / 2





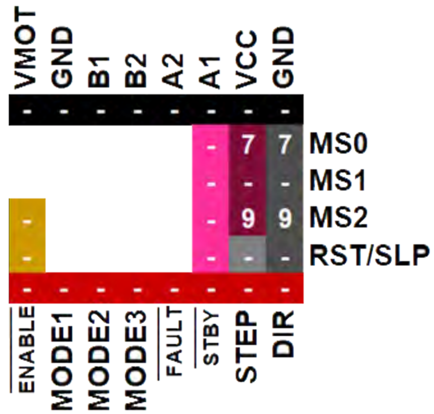
See Appendix D for legend

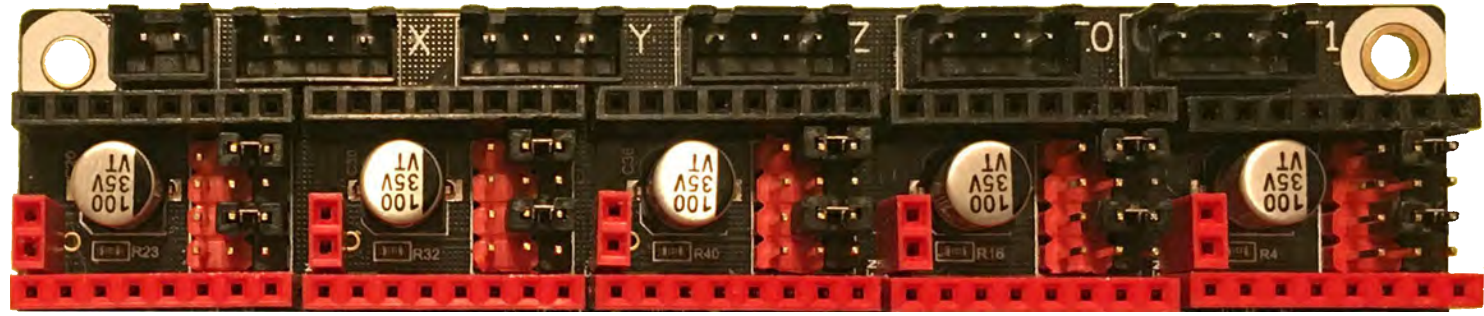
Stand-alone Mode

POLOLU TB67S249FTG

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

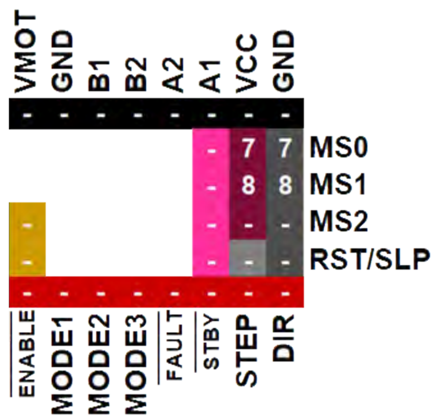
1 / 8

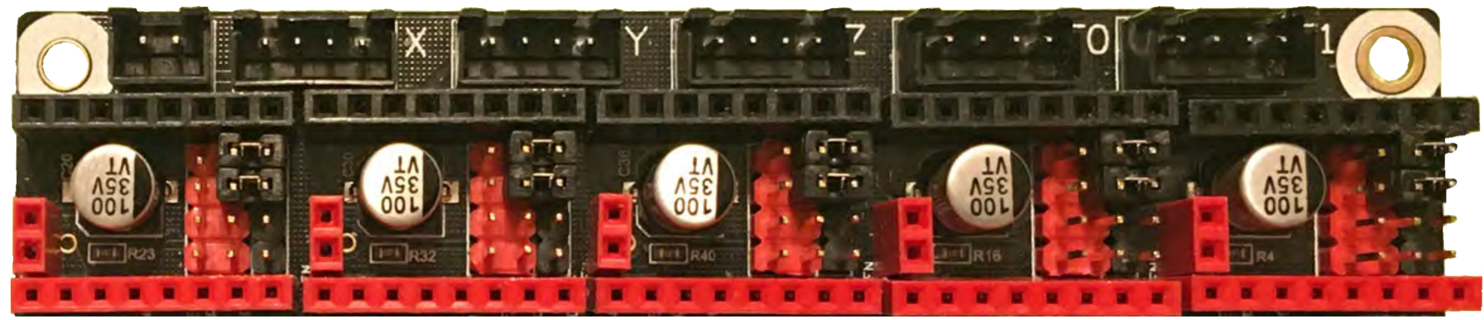




See Appendix D for legend

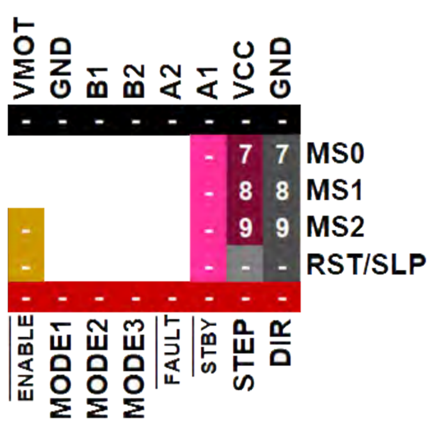
1 / 16

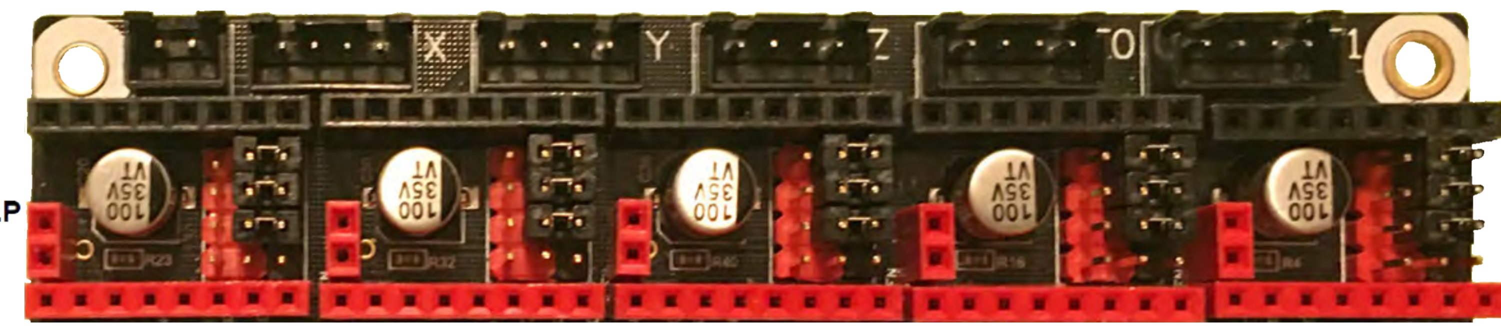




See Appendix D for legend

1 / 32



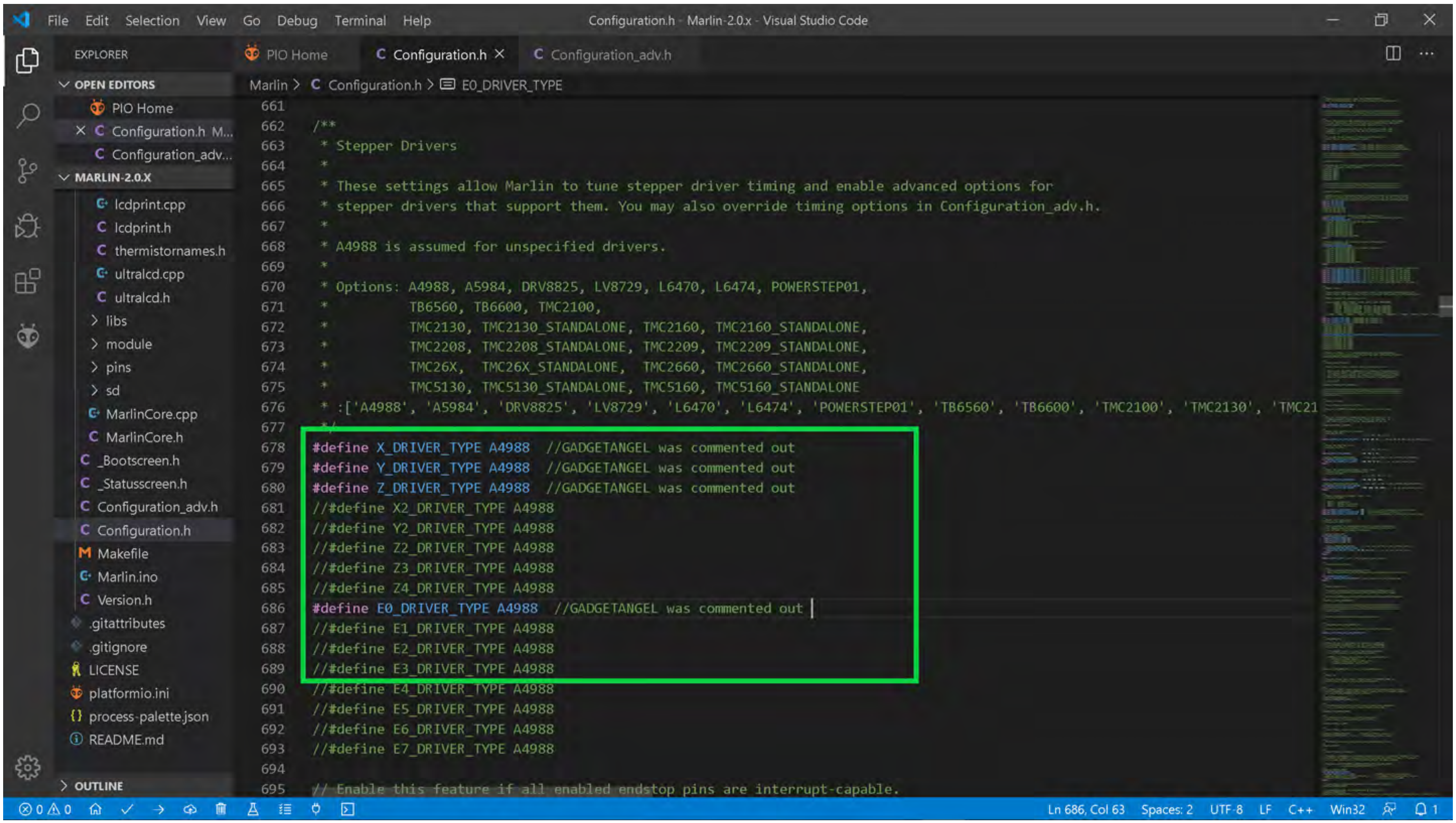


See Appendix D for legend

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

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

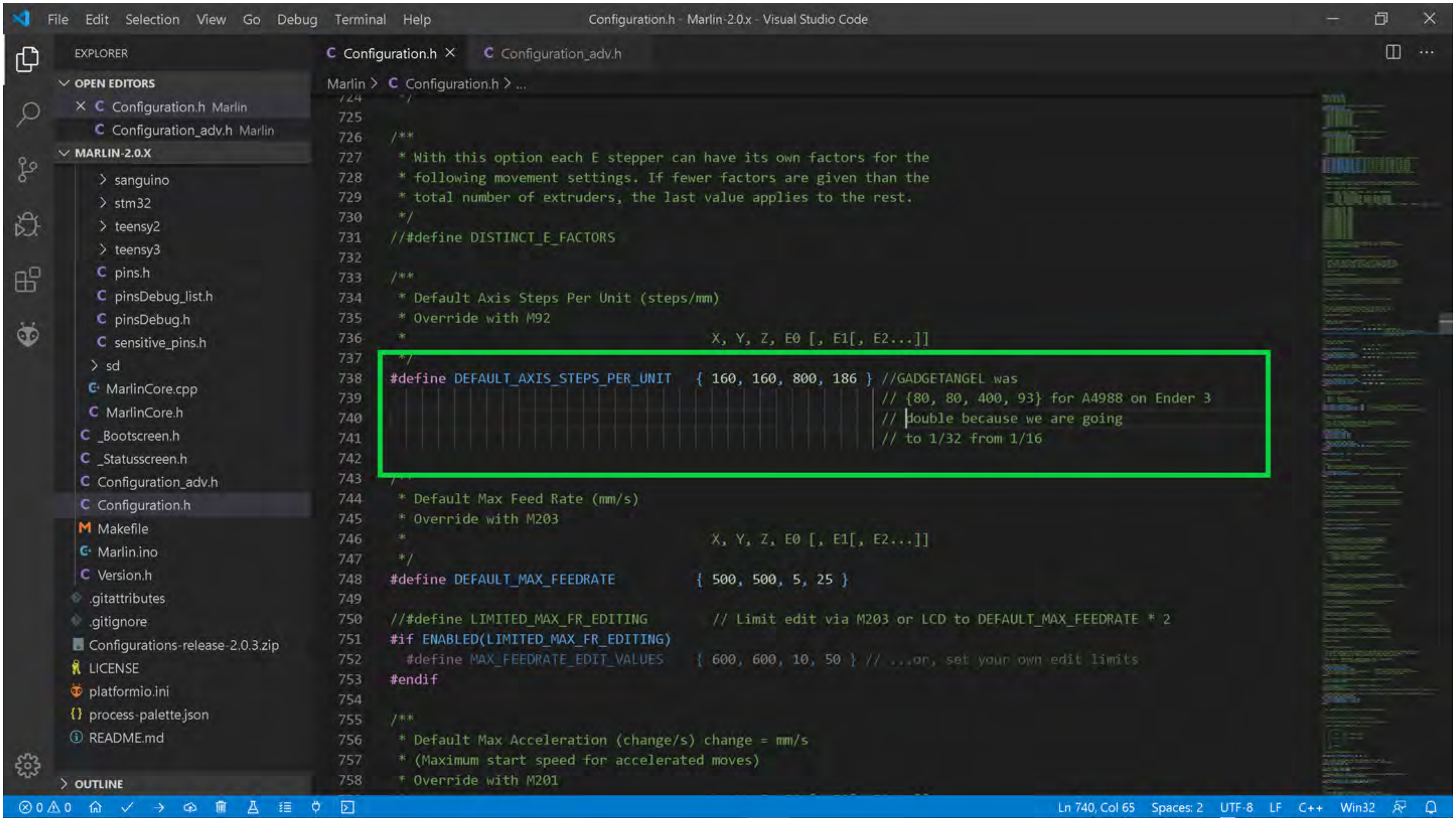
- Change the stepper motor drivers so that Marlin knows you are using POLOLU TB67S249FTG drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use POLOLU TB67S249FTG drivers. When two "//" appear at the beginning of a line that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



- Go to the next page.

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

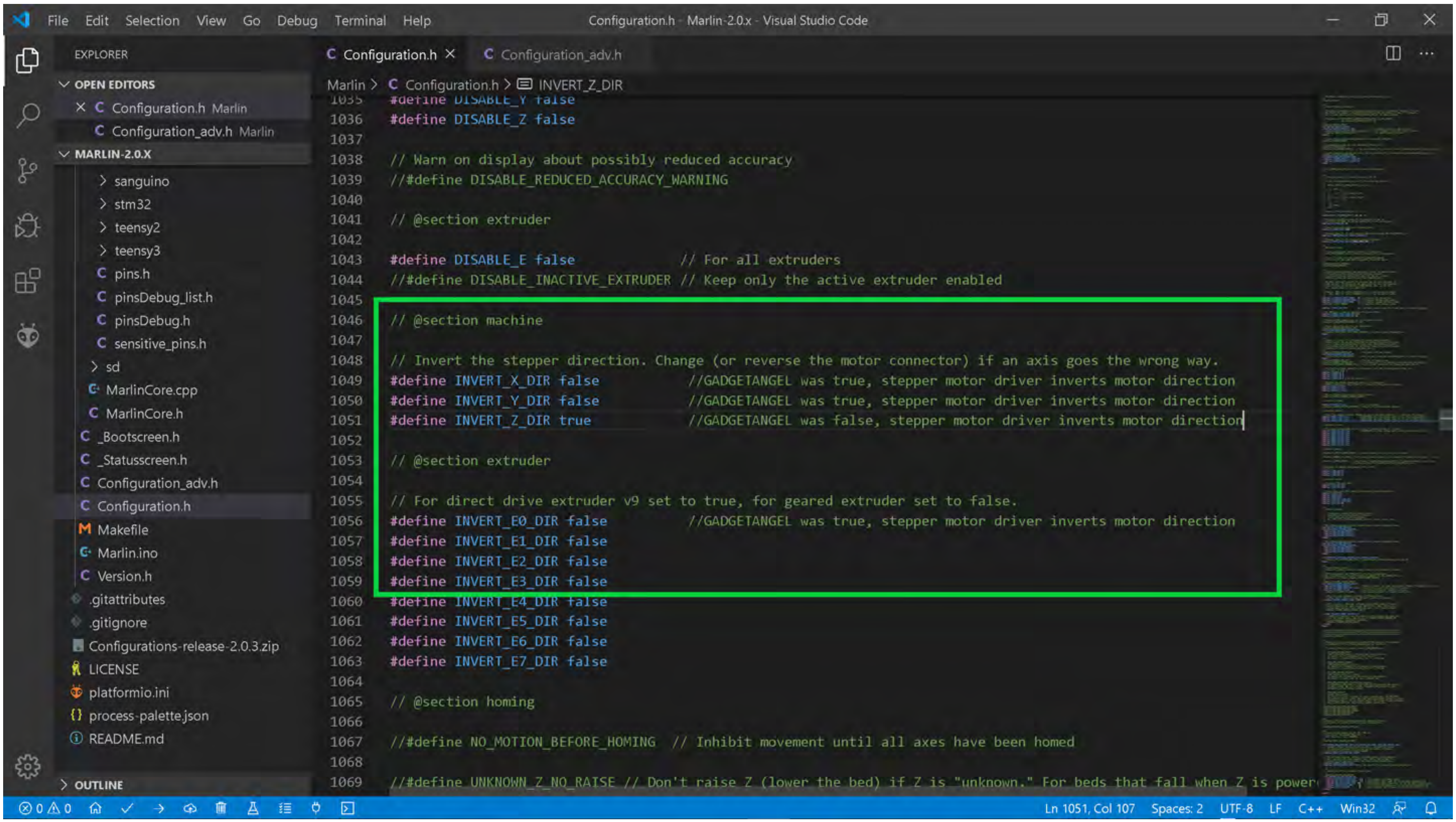
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for POLOLU TB67S249FTG stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT"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



- Go to the next page.

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

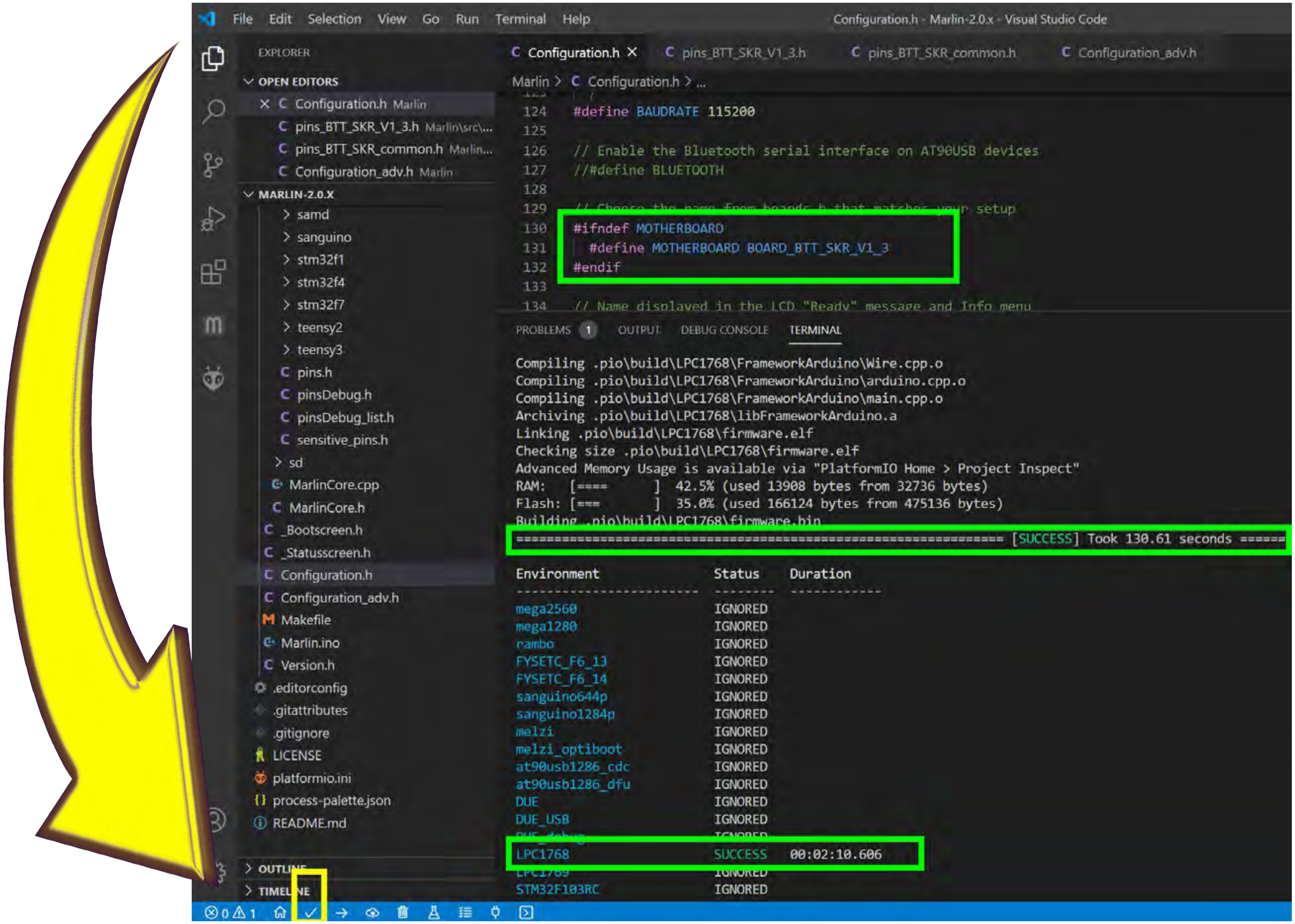
- Optional Step: I cannot find information on the POLOLU TB67S249FTG driver's impact on motor direction. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") **https://reprap.org/wiki/Stepper_wiring**. Other people prefer to change the motor direction in the Marlin firmware. So if you want or need to change the motor direction in Marlin, then if the axis' setting you will be using the TB67S249FTG driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

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

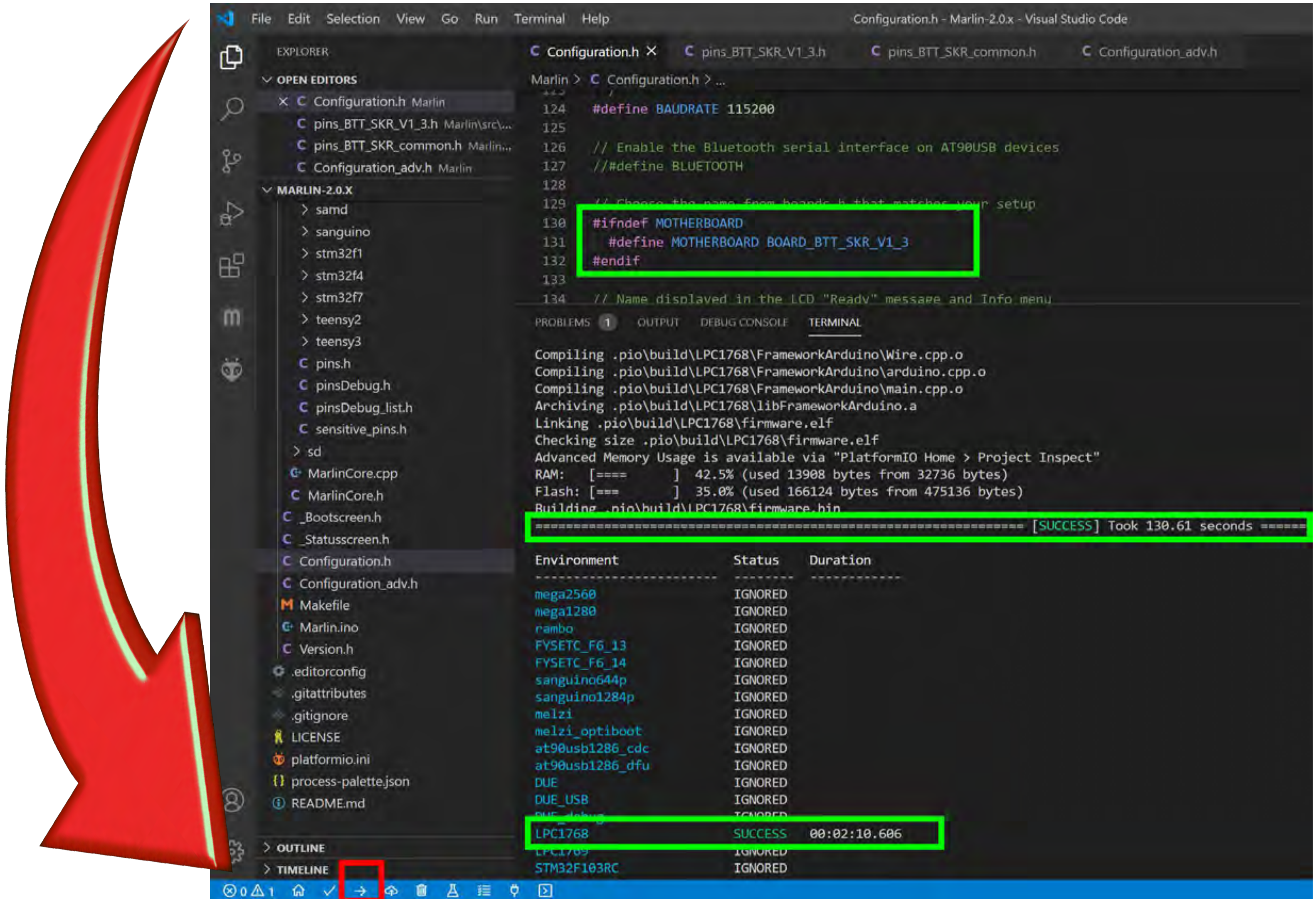
- The end of Marlin setup for POLOLU TB67S249FTG drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

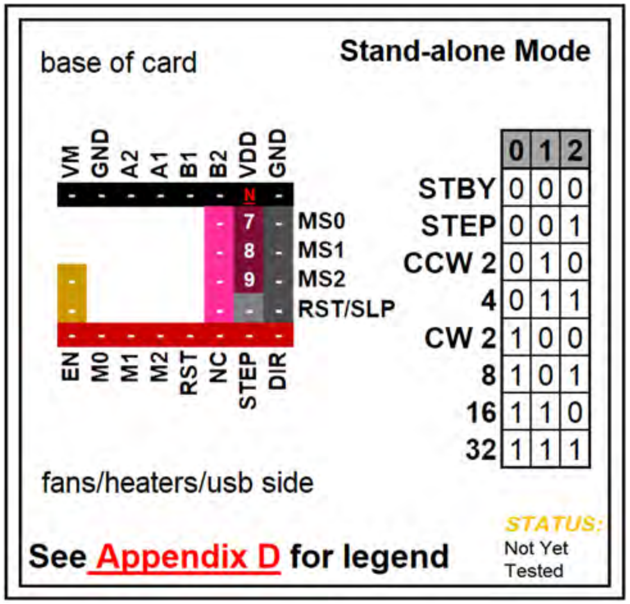
The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU S109



Note: V_{DD} is an Output (**N**), ***on a 3.3 V MCU like SKR V1.3 the V_{DD} must be disconnected!!***

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution
BIQU® S109 Maximum 32 Subdivision 25V DC 4A (peak)	Low	Low	Low	Standby mode (outputs disabled)
	Low	Low	High	Full step
	Low	High	Low	Non-circular half step ("a")
	Low	High	High	1/4 step
	High	Low	Low	Circular half step ("b")
	High	Low	High	1/8 step
	High	High	Low	1/16 step
	High	High	High	1/32 step
Driving Current Calculation Formula V_{DD} = 5 V DC R_S(Typical Sense Resistor)$=0.1\Omega$	$I_{MAX}=V_{ref}*\left(\frac{V_{DD}}{5}\right)*\frac{1}{(5*R_S)}$			$V_{ref}=I_{MAX}*\left(\frac{5}{V_{DD}}\right)*(5*R_S)$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU S109

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

base of card

Stand-alone Mode

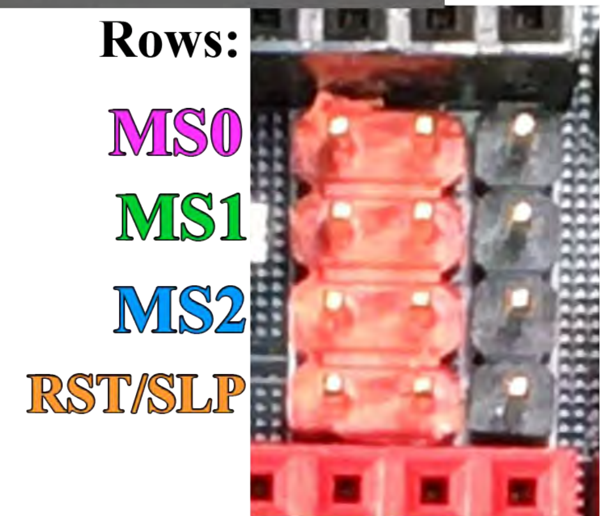
	0	1	2
STBY	0	0	0
STEP	0	0	1
CCW 2	0	1	0
4	0	1	1
CW 2	1	0	0
8	1	0	1
16	1	1	0
32	1	1	1

fans/heaters/usb side

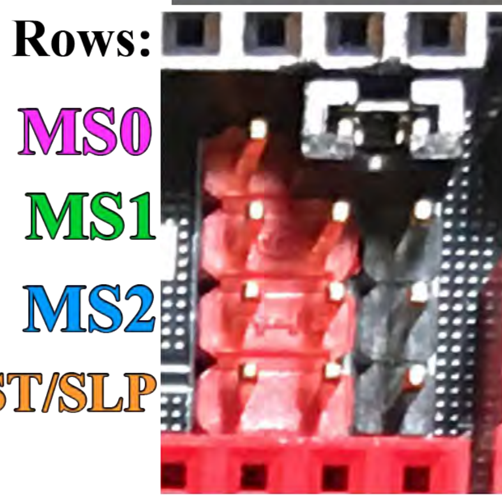
STATUS: Not Yet Tested

See Appendix D for legend

Low → No Jumper



High → Jumper Set



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
x.xA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

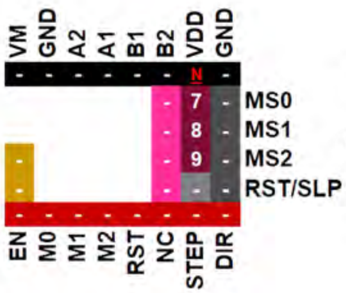
Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

base of card

Stand-alone Mode



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

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

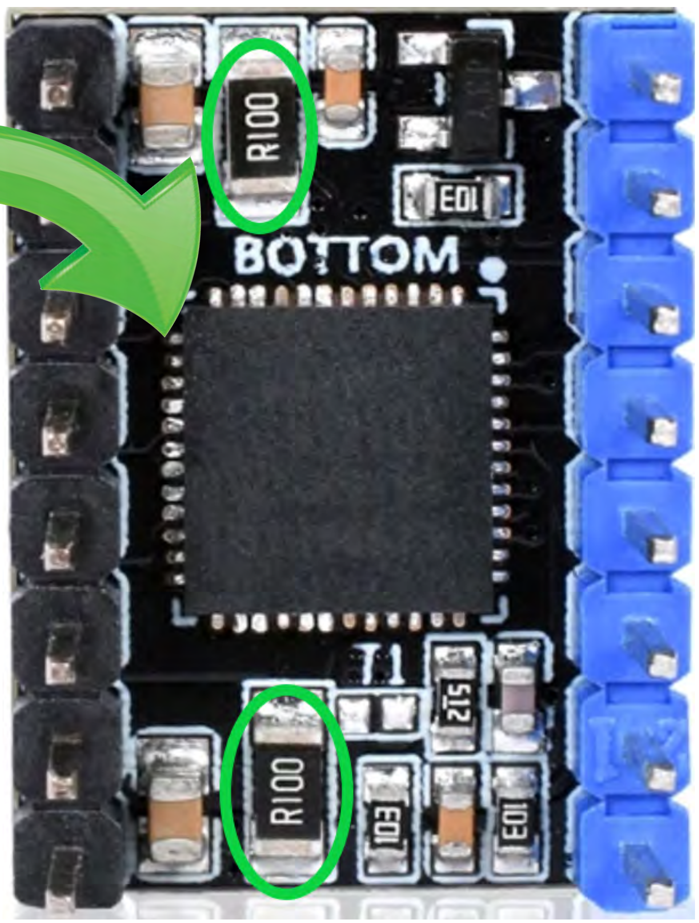
BIQU S109

Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board. V_{DD} is an Output (**N**), **on a 3.3 V MCU like SKR V1.3 the V_{DD} must be disconnected!**

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

Note:

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



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

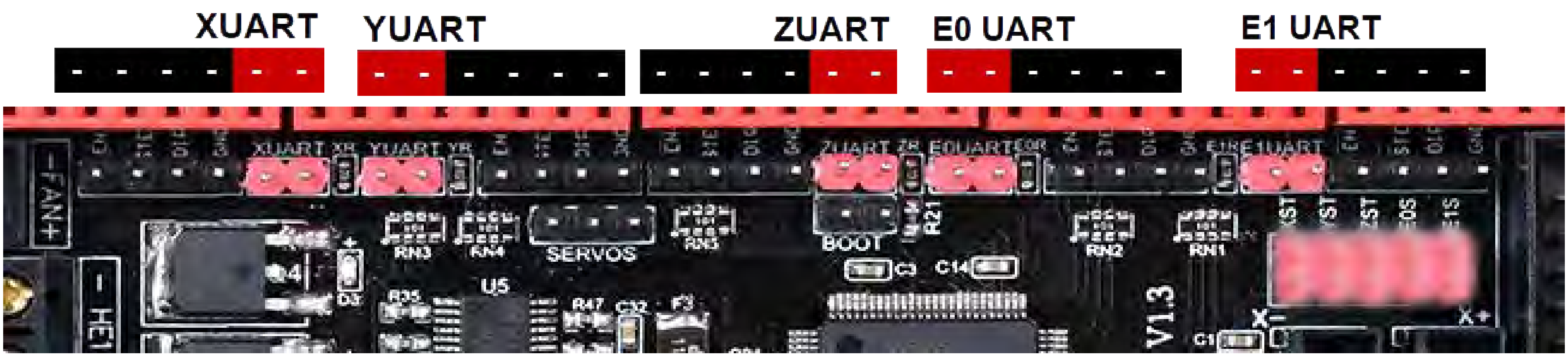
Note: See this video about current sense resistors (R_s) and their possible locations: **https://youtu.be/8wk1elugv5A**

Stand-alone Mode

BIQU S109

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

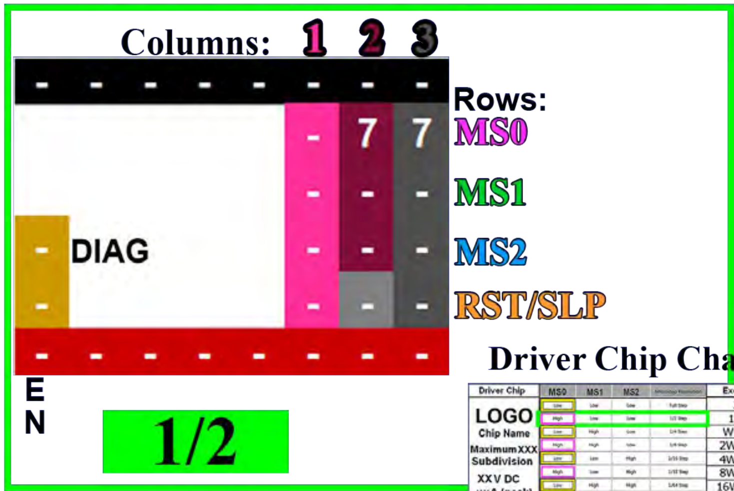


Stand-alone Mode

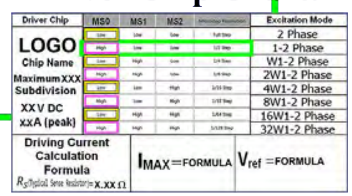
BIQU S109

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

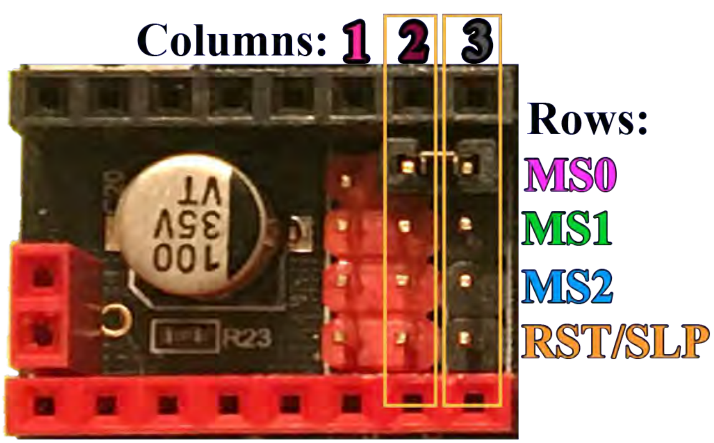
Driver Socket Representation



Driver Chip Chart:



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

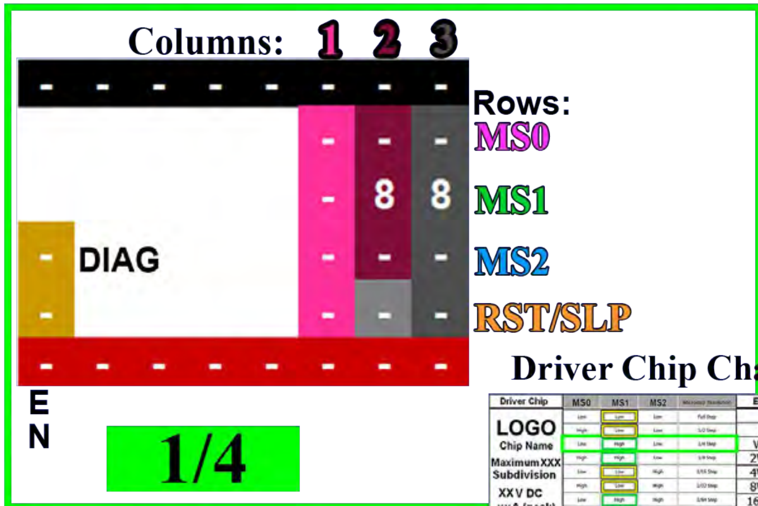
Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

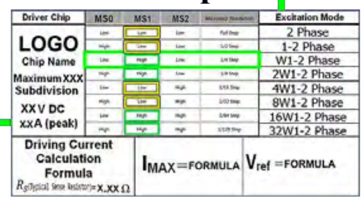
Stand-alone Mode

BIQU S109

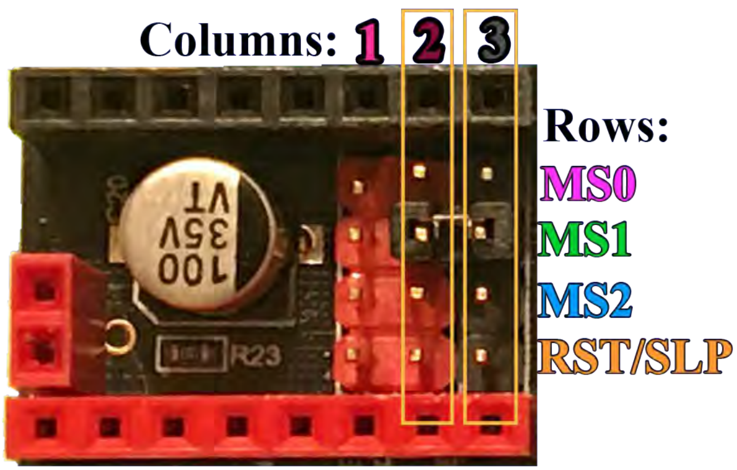
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

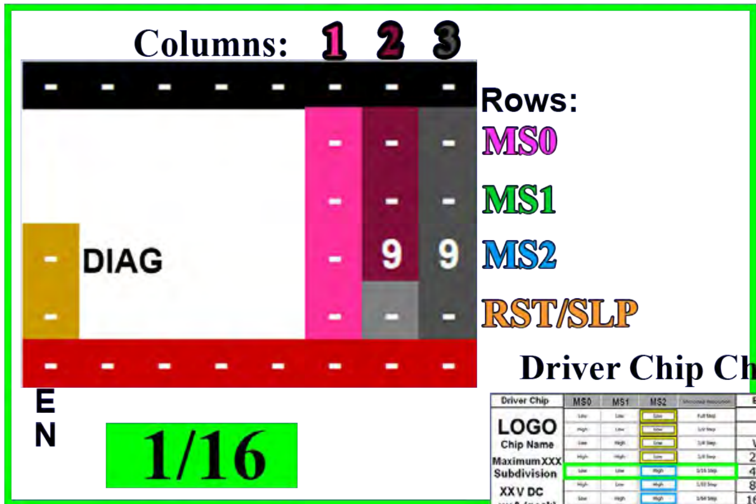
Meaning:

Driver Chip Chart:	
High	set Jumper between column 2 and column 3 on the MS1 row
Low	No Jumper Set

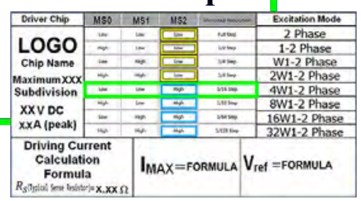
Stand-alone Mode

BIQU S109

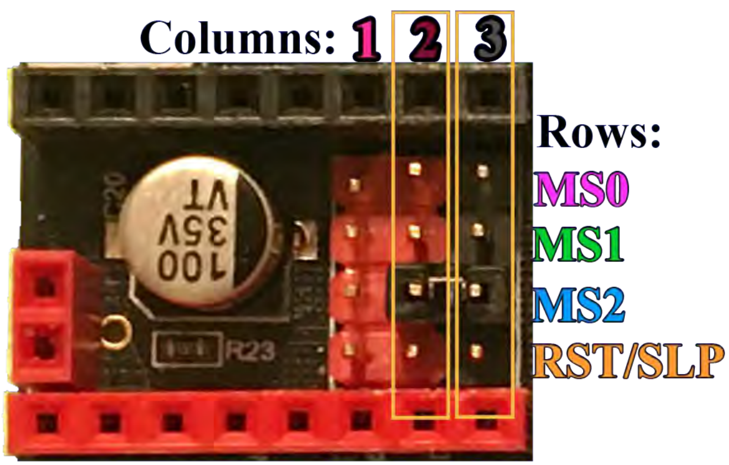
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 9 9 → Driver Chip Chart: High

Meaning:

Driver Chip Chart:

High → **set Jumper between column 2 and column 3 on the MS2 row**

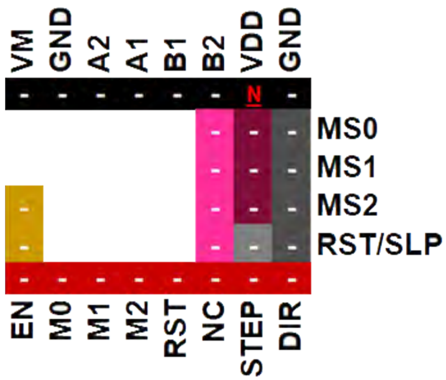
Low → **No Jumper Set**

Stand-alone Mode

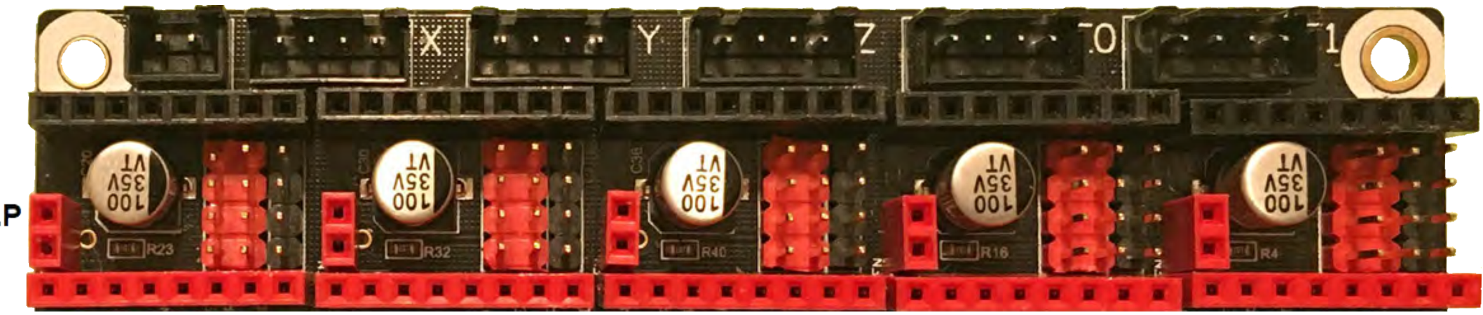
BIQU S109

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

STAND BY

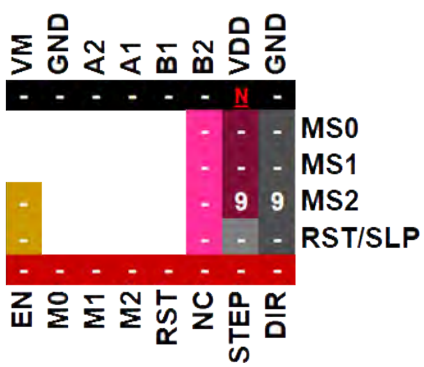


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

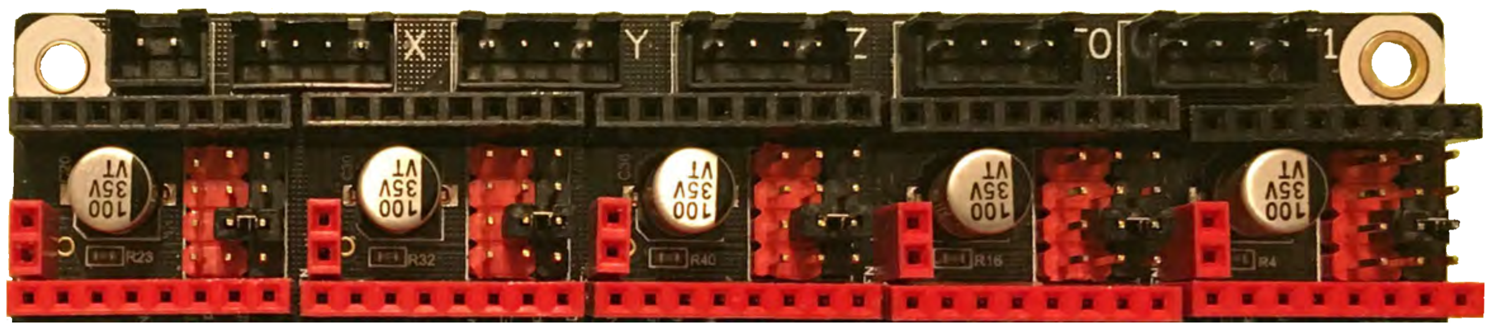


See Appendix D for legend

STEP



Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



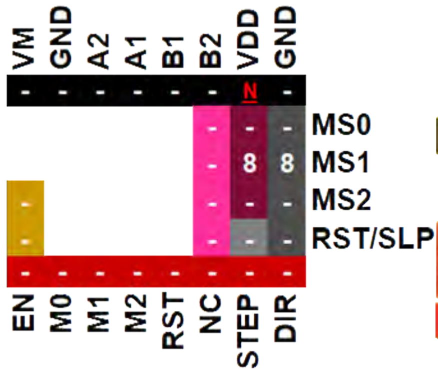
See Appendix D for legend

Stand-alone Mode

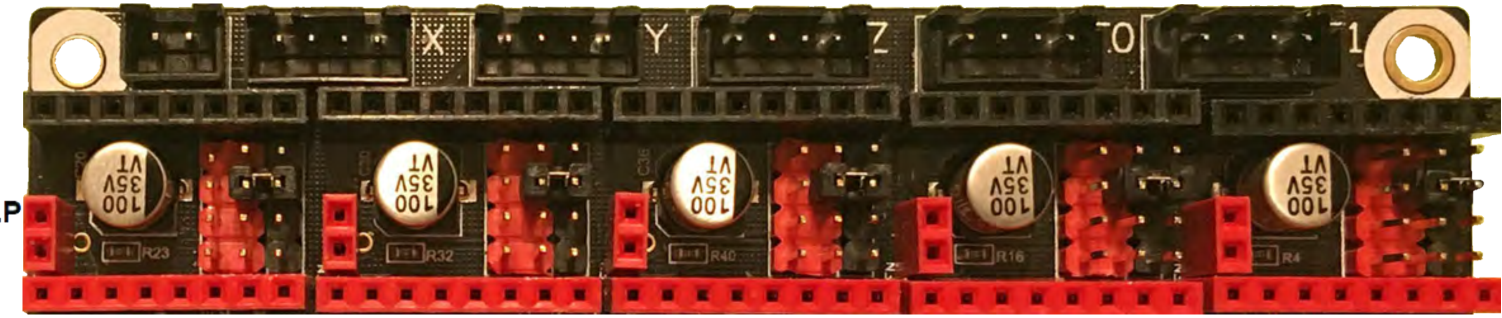
BIQU S109

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

CCW
1 / 2



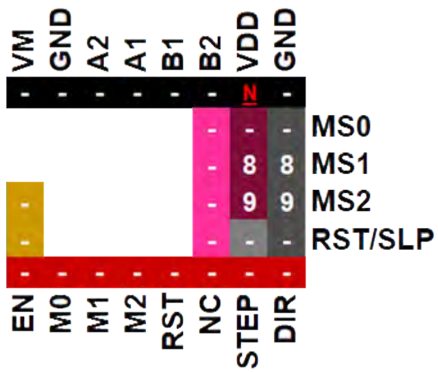
Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



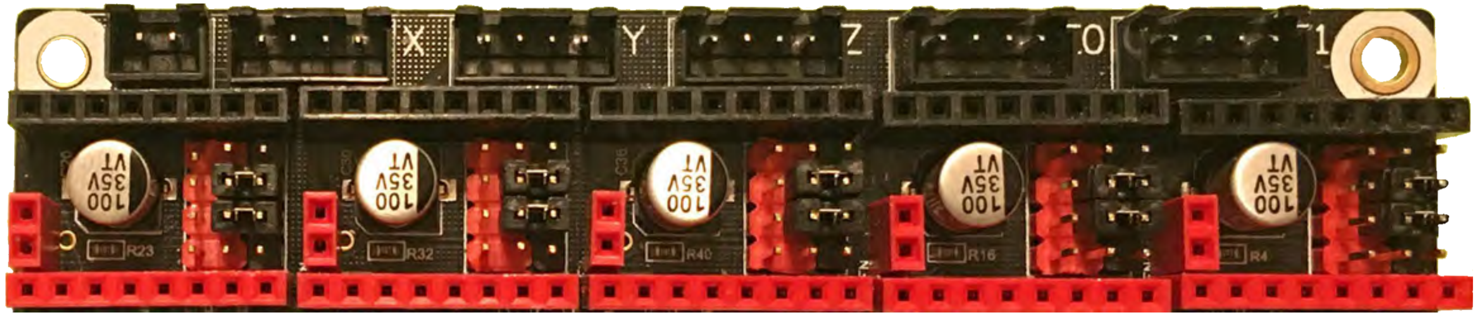
See Appendix D for legend

---

1 / 4

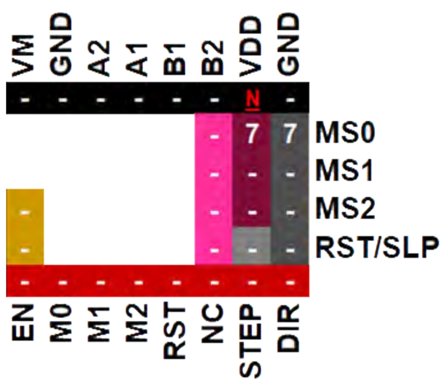


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

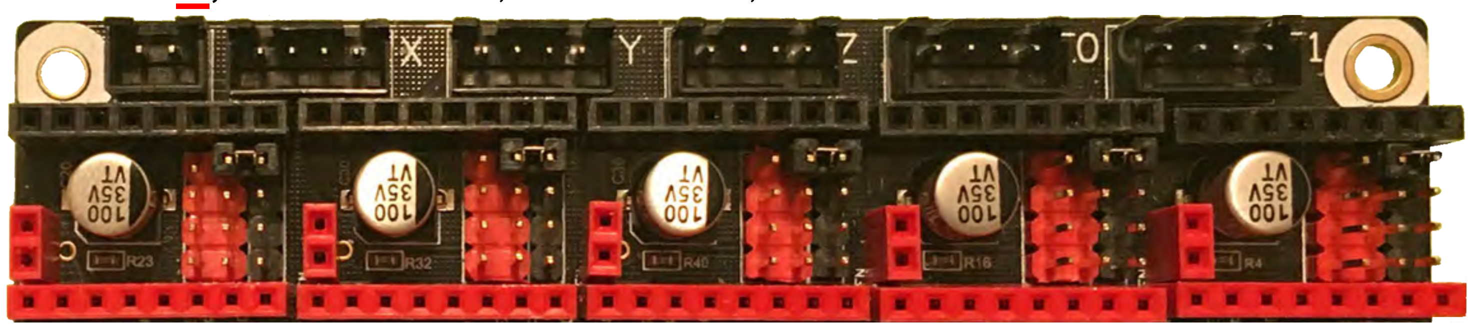


See Appendix D for legend

CW
1 / 2



Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



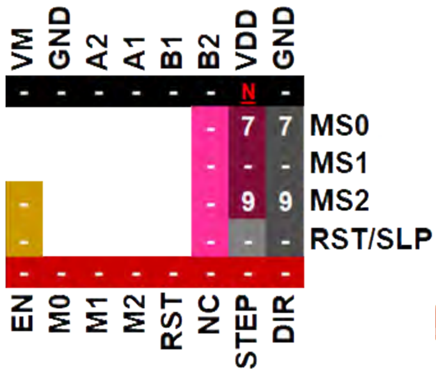
See Appendix D for legend

Stand-alone Mode

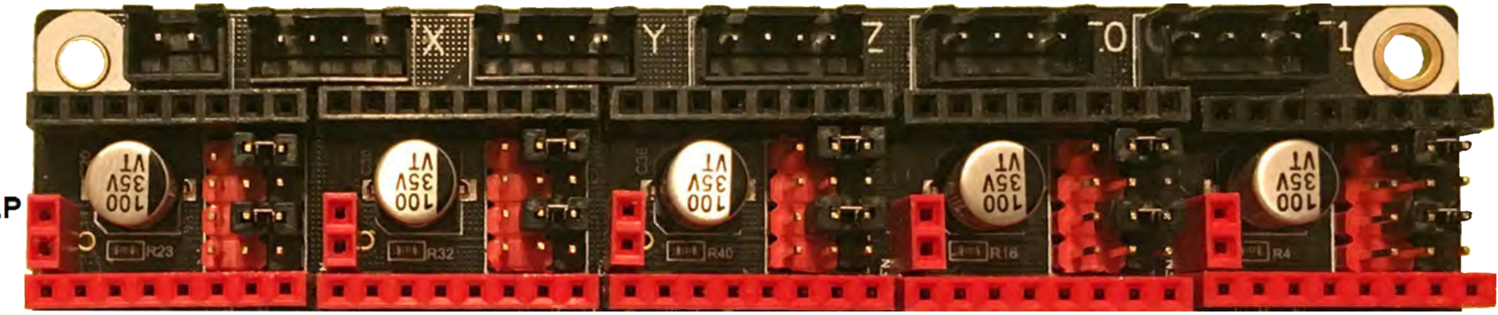
BIQU S109

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

1 / 8

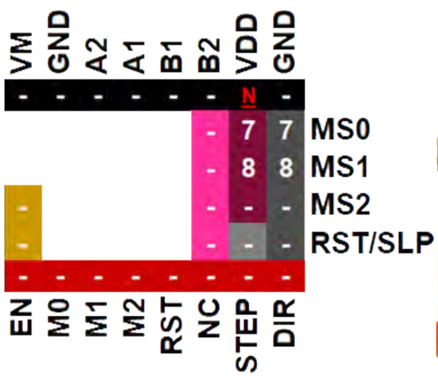


Note: **N**, on a 3.3 V MCU, like SKR V1.3 **MUST** be disconnected!

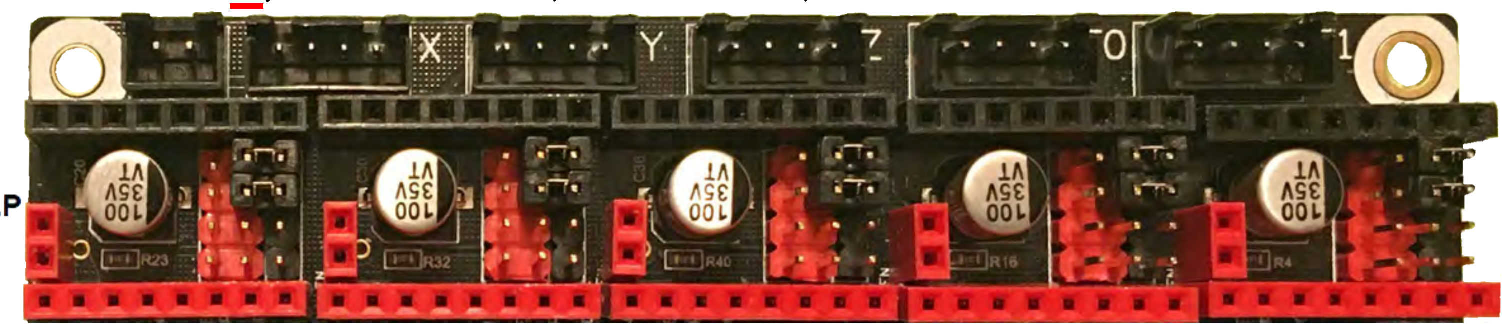


See Appendix D for legend

1 / 16

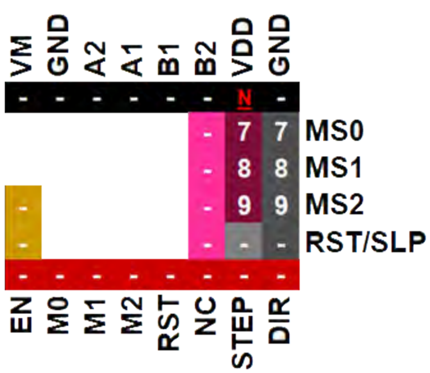


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

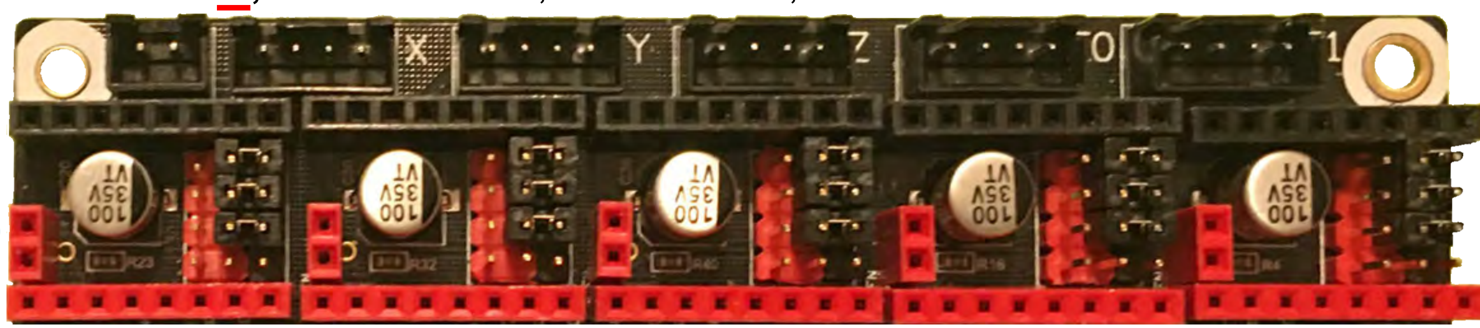


See Appendix D for legend

1 / 32



Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

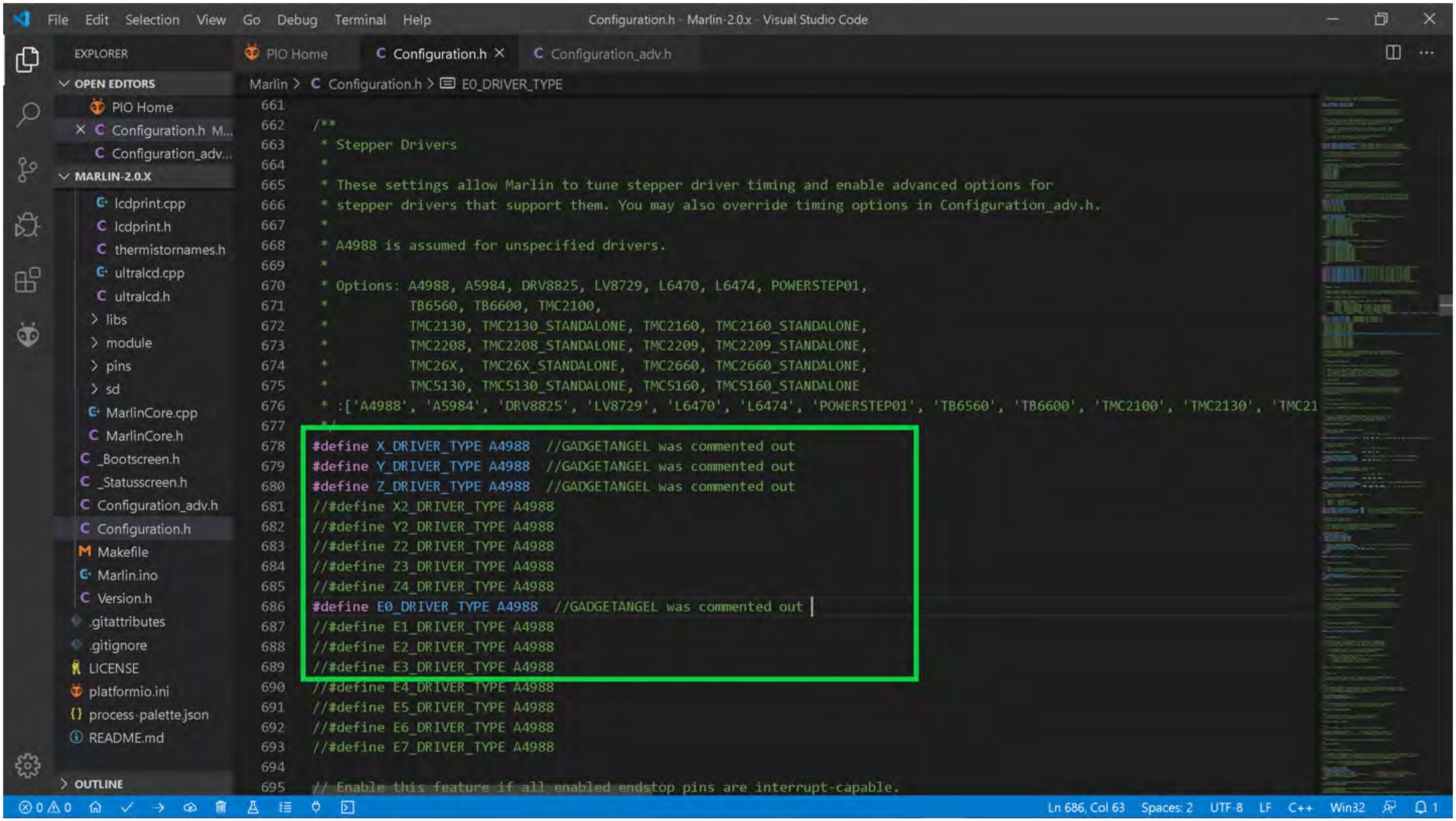


See Appendix D for legend

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

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

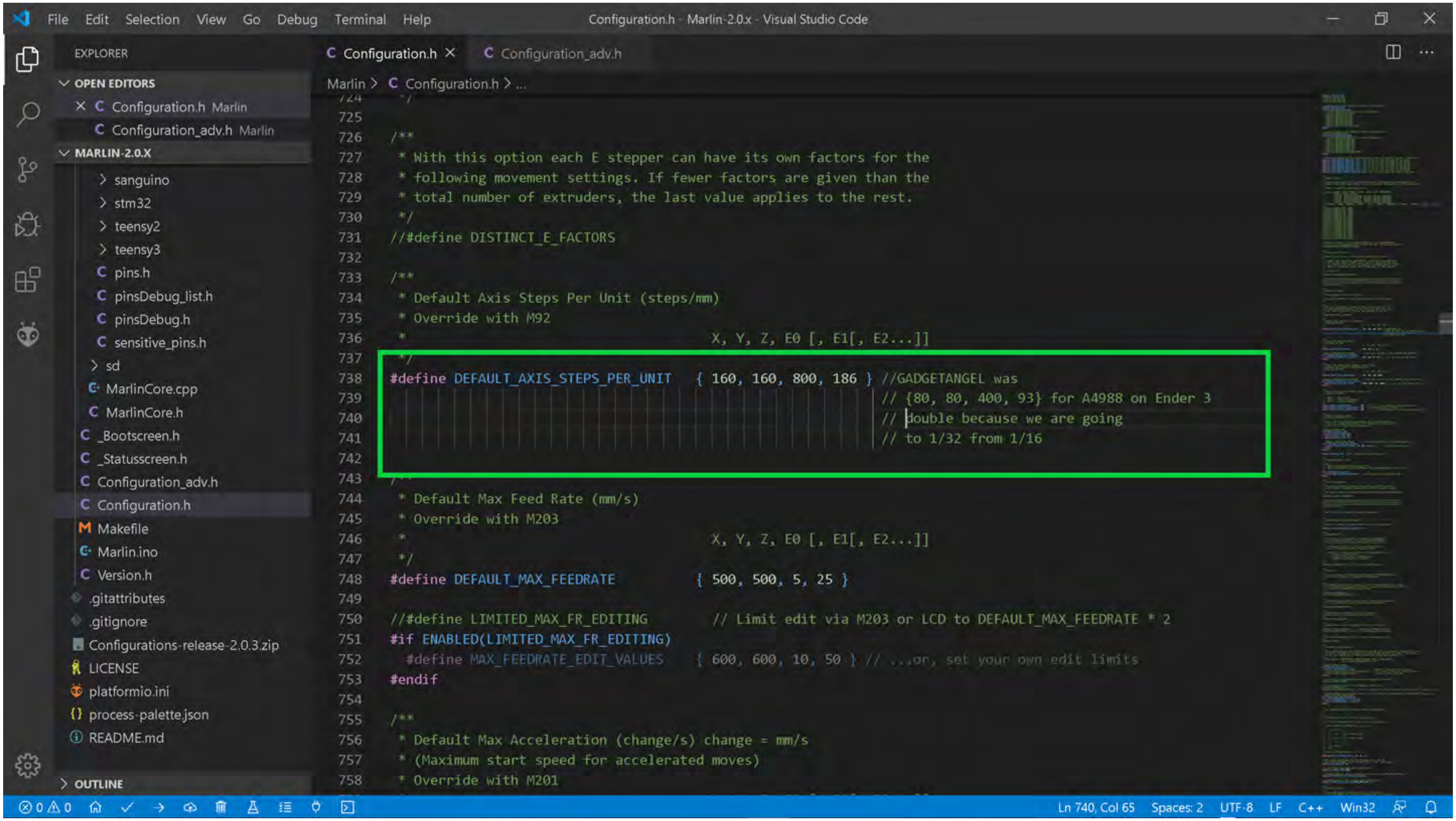
- Change the stepper motor drivers so that Marlin knows you are using BIQU S109 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU S109 drivers. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The BIQU S109 is a drop in replacement for the A4988. Since Marlin does not have an option for BIQU S109 we will use the A4988 as the driver type.



- Go to the next page.

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

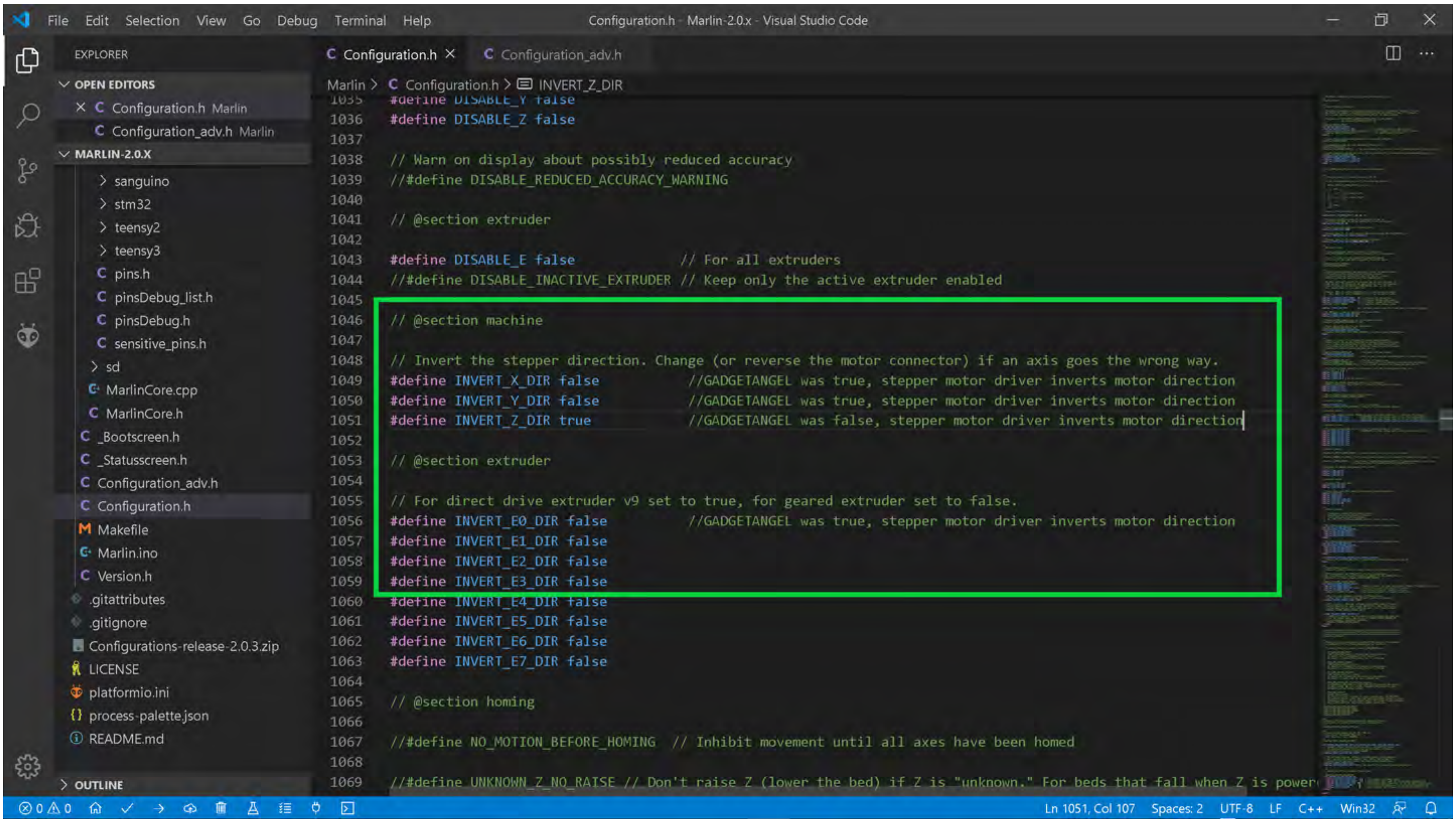
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for BIQU S109 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

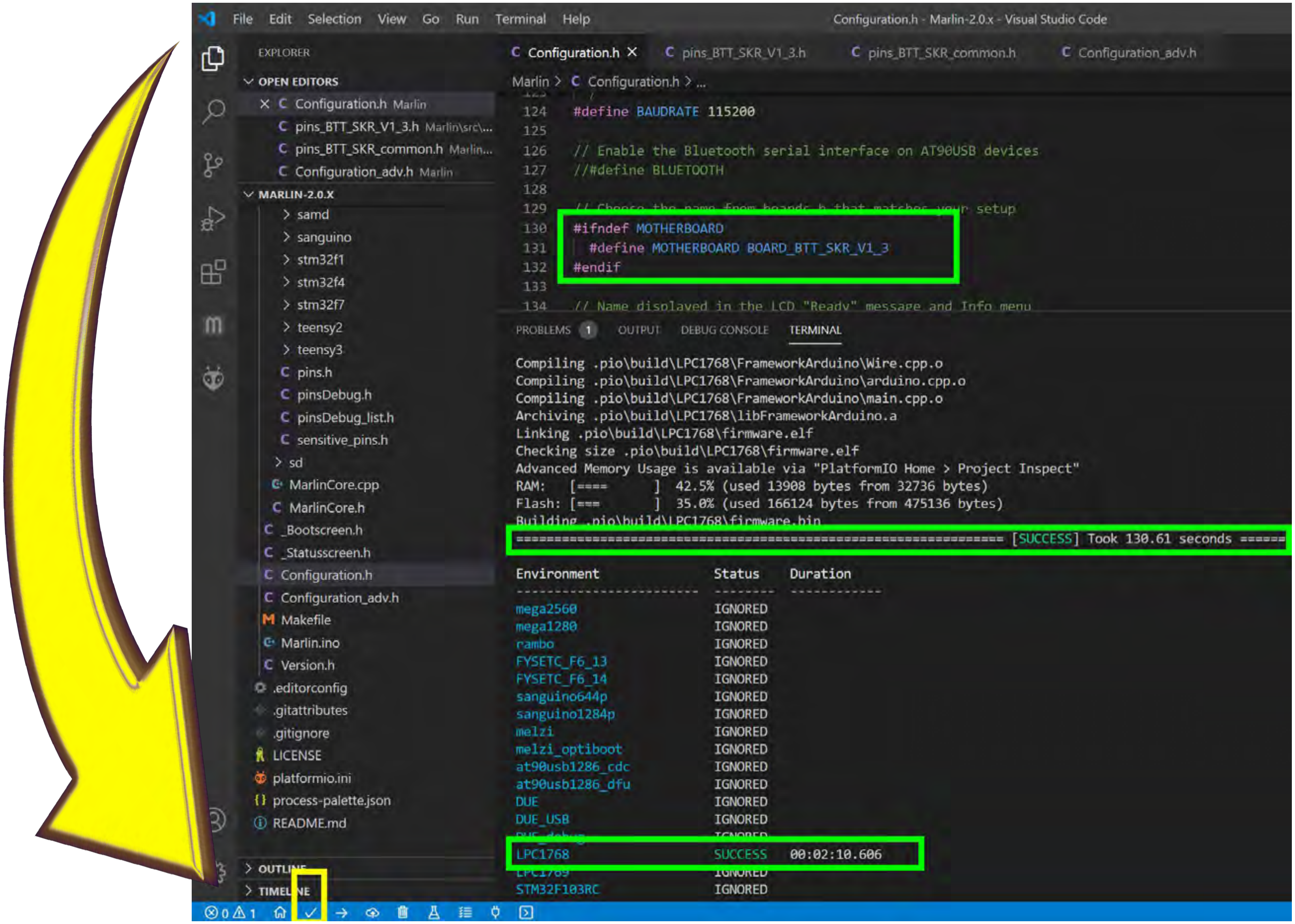
- Optional Step: I cannot find information on the BIQU S109 driver's impact on motor direction. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") **https://reprap.org/wiki/Stepper_wiring**. Other people prefer to change the motor direction in the Marlin firmware. So if you want or need to change the motor direction in Marlin, then if the axis' setting you will be using the S109 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

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

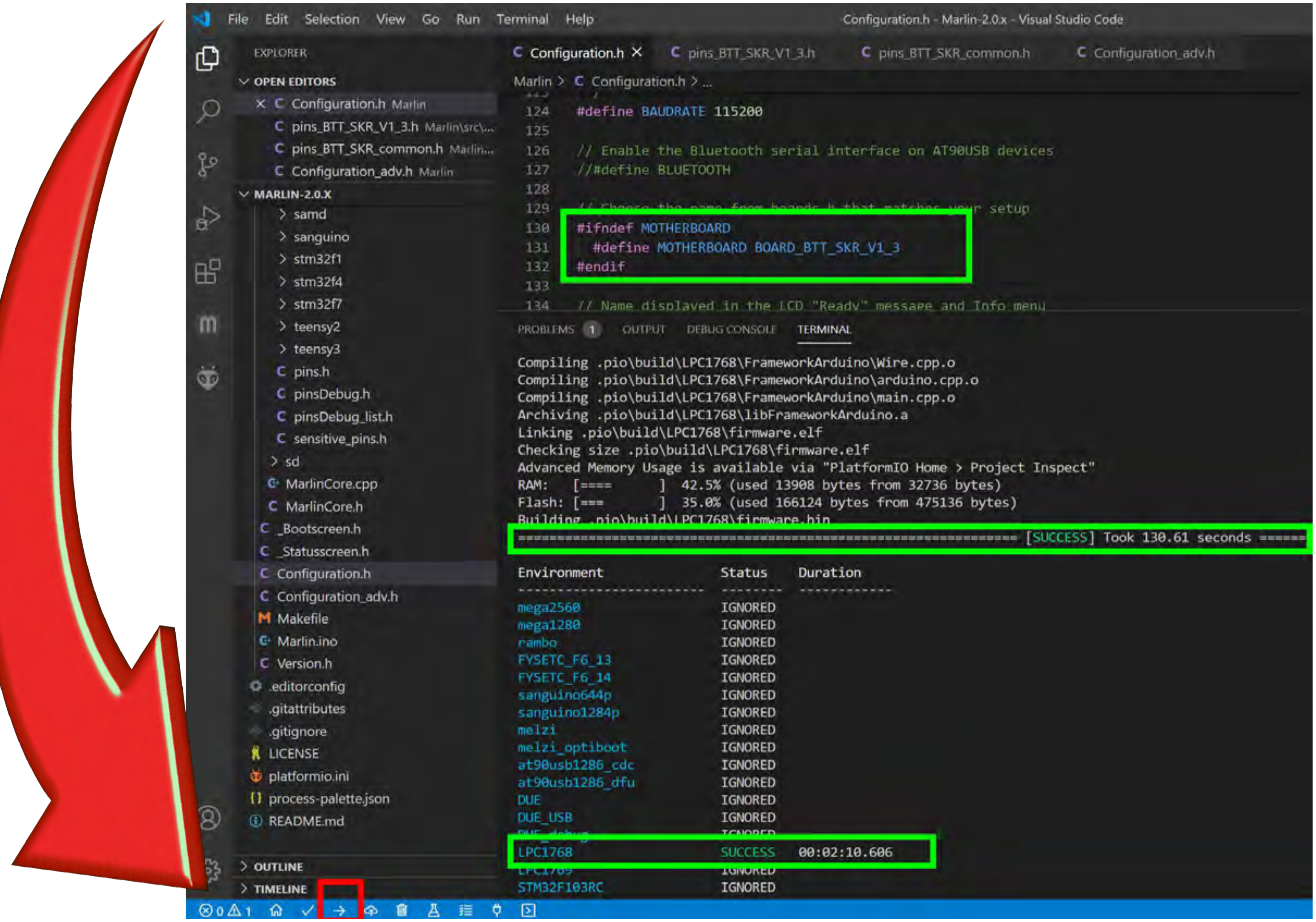
- The end of Marlin setup for BIQU S109 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

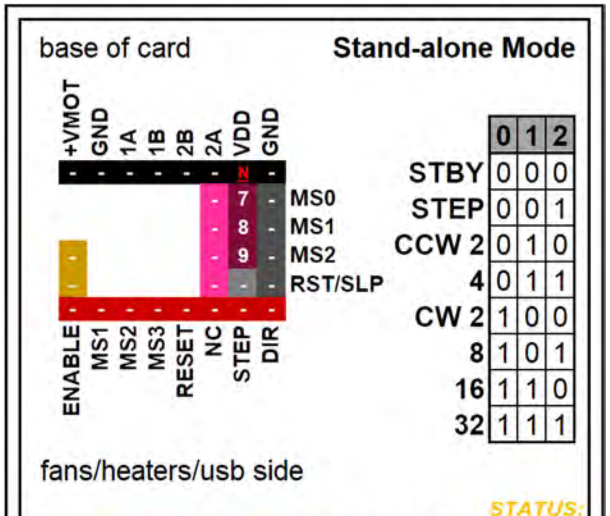
The (latest release of) Marlin Setup for BIQU S109 Drivers

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

FYSETC S109



See Appendix D for legend

Note: V_{DD} is an Output (**N**), ***on a 3.3 V MCU like SKR V1.3 the V_{DD} must be disconnected!!***

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

Driver Chip	MS0	MS1	MS2	Microstep Resolution
FYSETC **S109** **Maximum 32 Subdivision** **50V DC 4A (peak)**	Low	Low	Low	Standby mode (outputs disabled)
	Low	Low	High	Full step
	Low	High	Low	Non-circular half step ("a")
	Low	High	High	1/4 step
	High	Low	Low	Circular half step ("b")
	High	Low	High	1/8 step
	High	High	Low	1/16 step
	High	High	High	1/32 step
Driving Current Calculation Formula V_{DD} **= 5 V DC** R_S(Typical Sense Resistor)$=0.1\Omega$	$I_{MAX}=V_{ref}*\left(\frac{V_{DD}}{5}\right)*\frac{1}{(5*R_S)}$			$V_{ref}=I_{MAX}*\left(\frac{5}{V_{DD}}\right)*(5*R_S)$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

base of card

Stand-alone Mode

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

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

FYSETC S109

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇒ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇒ Jumper Set

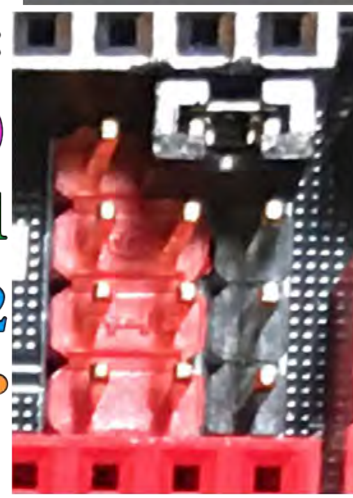
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
XX V DC	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
xxA (peak)	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

FYSETC S109

base of card

Stand-alone Mode

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

fans/heaters/usb side

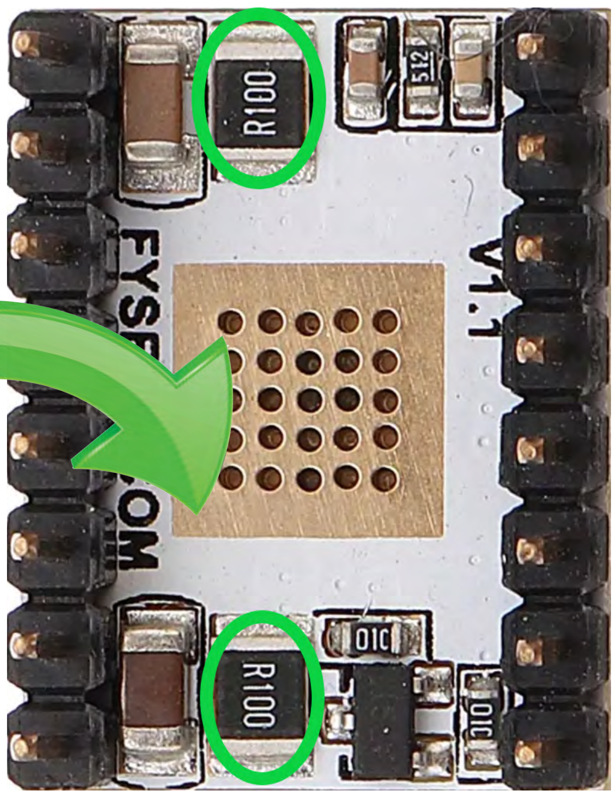
STATUS: Not Yet Tested

See Appendix D for legend

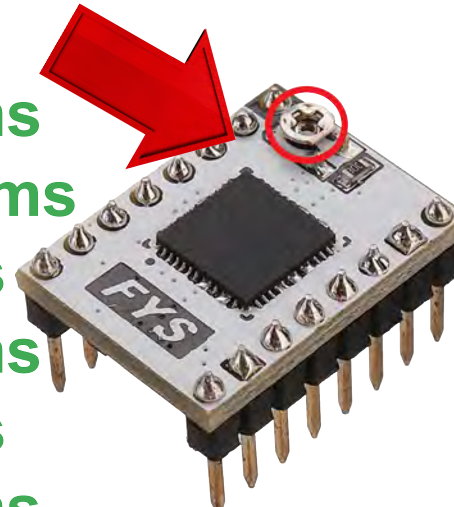
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board. V_{DD} is an Output (**N**), **on a 3.3 V MCU like SKR V1.3 the V_{DD} must be disconnected!**

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

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



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



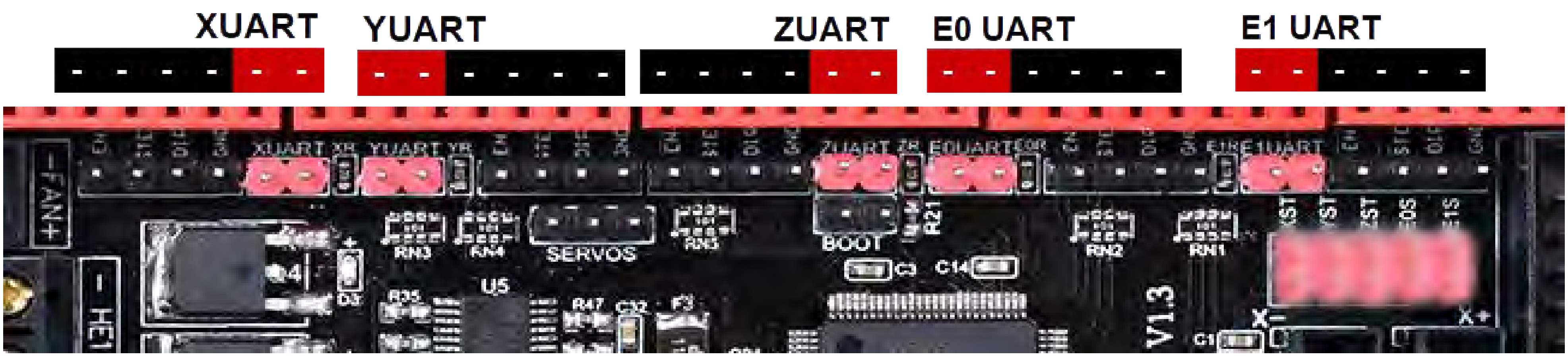
Note: See this video about current sense resistors (R_s) and their possible locations: **https://youtu.be/8wk1elugv5A**

Stand-alone Mode

FYSETC S109

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

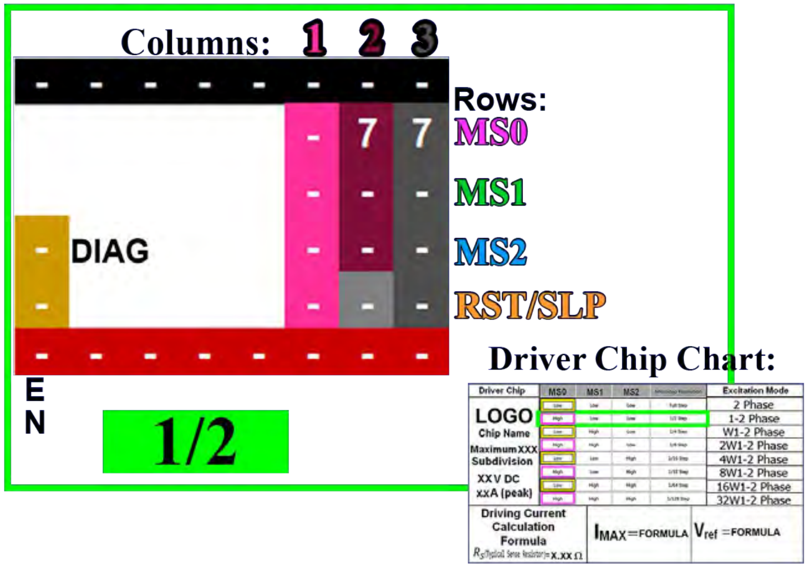


Stand-alone Mode

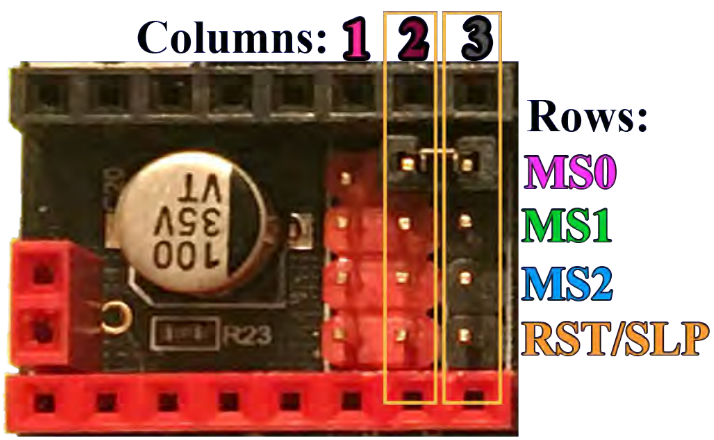
FYSETC S109

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:

Meaning:

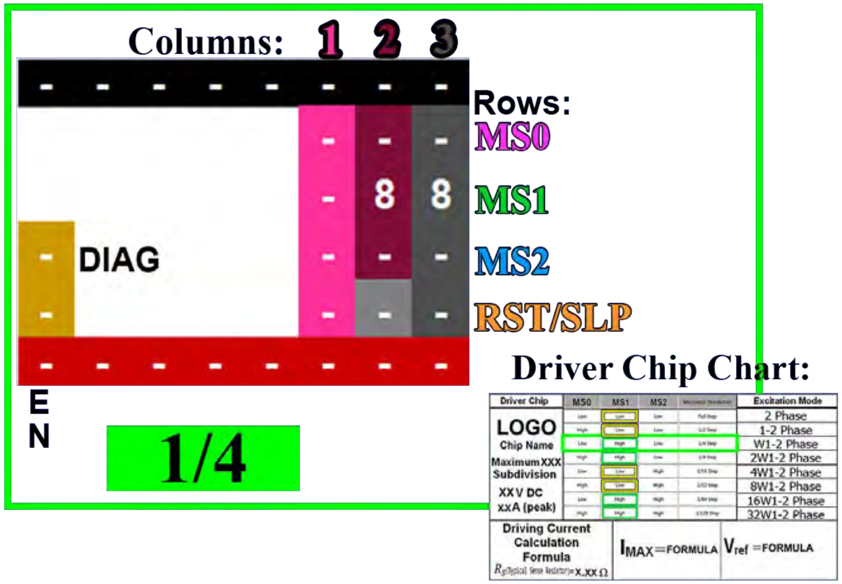
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

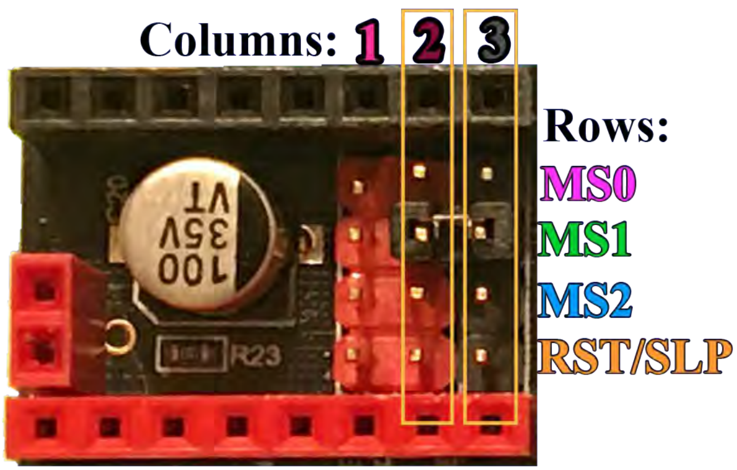
Stand-alone Mode

FYSETC S109

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 8 8 → Driver Chip Chart: **High**

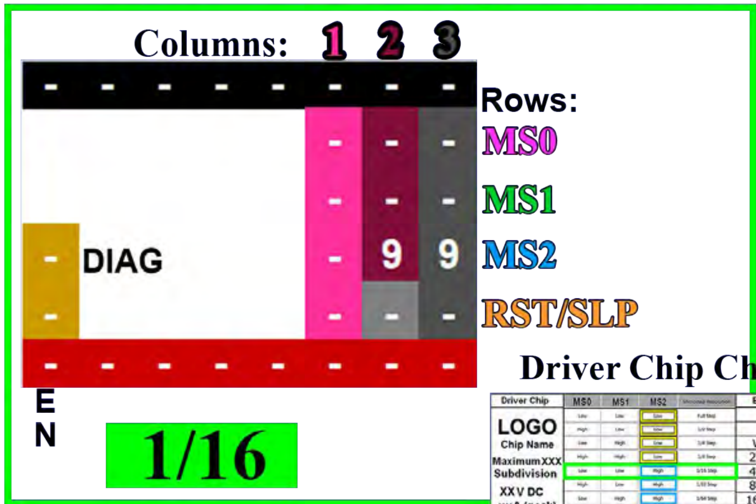
Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS1 row
Low →	No Jumper Set

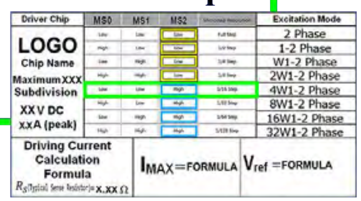
Stand-alone Mode

FYSETC S109

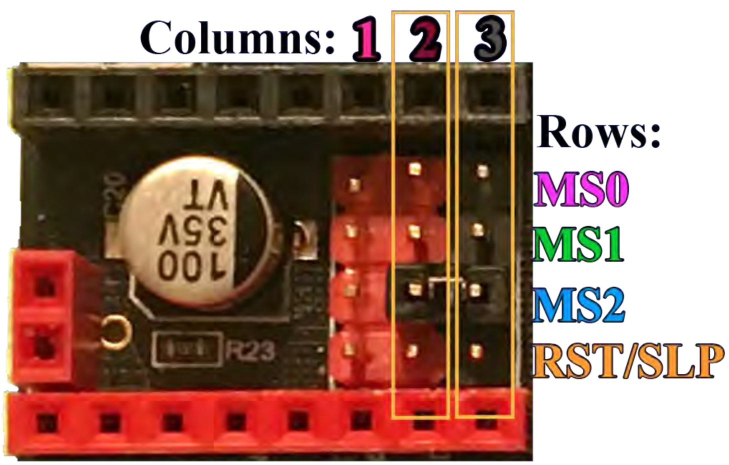
Driver Socket Representation



Driver Chip Chart:



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 9 9 → Driver Chip Chart: **High**

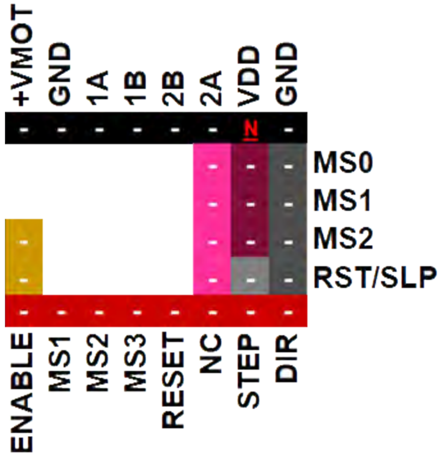
Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS2 row
Low →	No Jumper Set

Stand-alone Mode

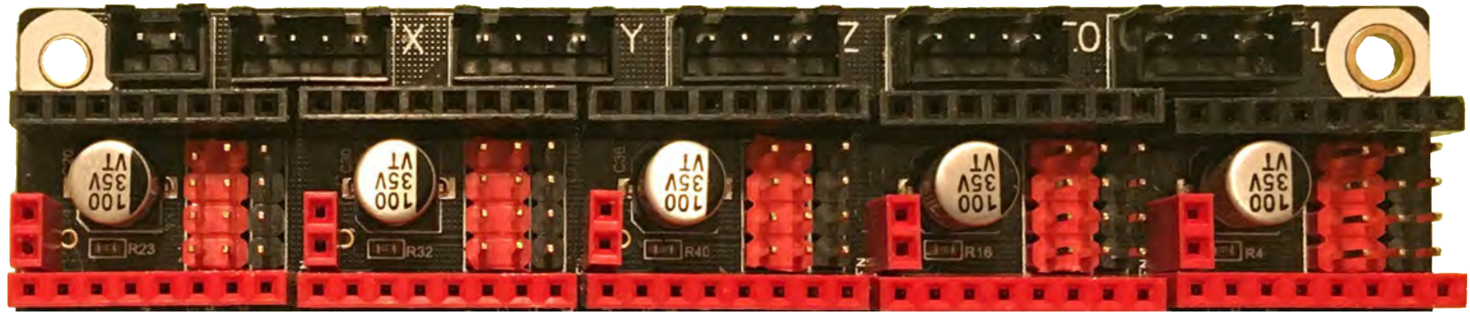
FYSETC S109

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

STAND BY

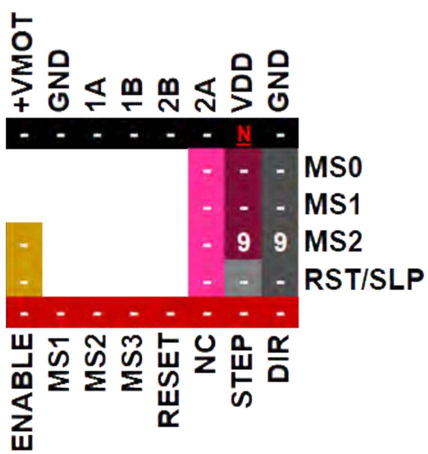


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

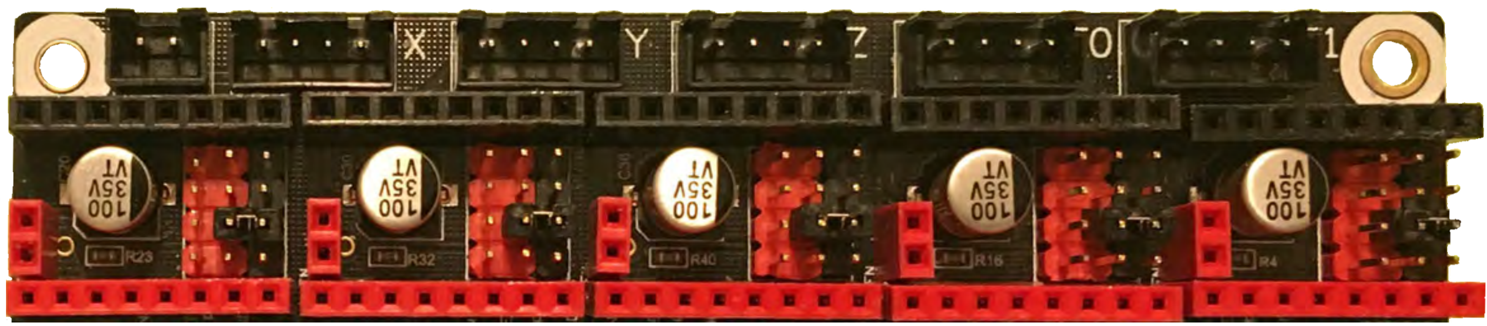


See Appendix D for legend

STEP



Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



See Appendix D for legend

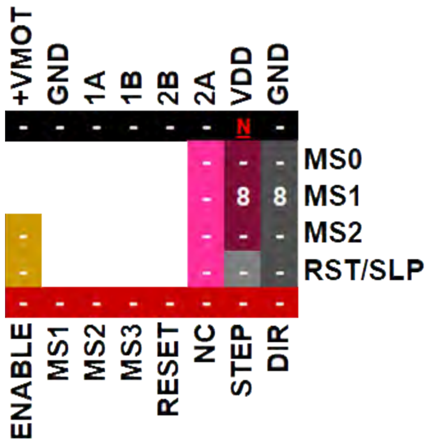
Stand-alone Mode

FYSETC S109

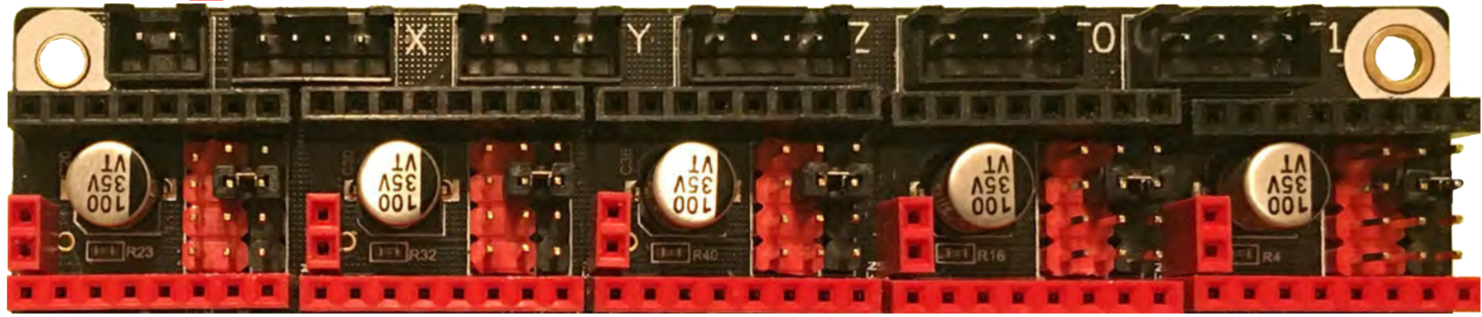
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

CCW

1 / 2

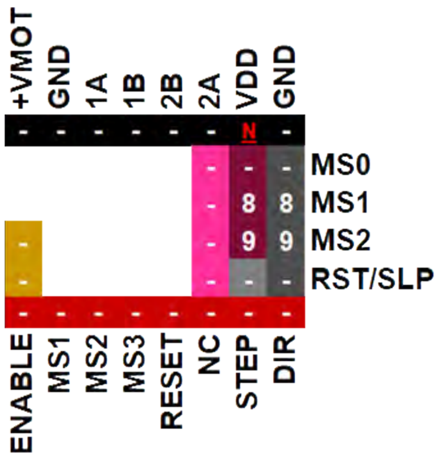


Note: **N**, on a 3.3 V MCU, like SKR V1.3 **MUST** be disconnected!

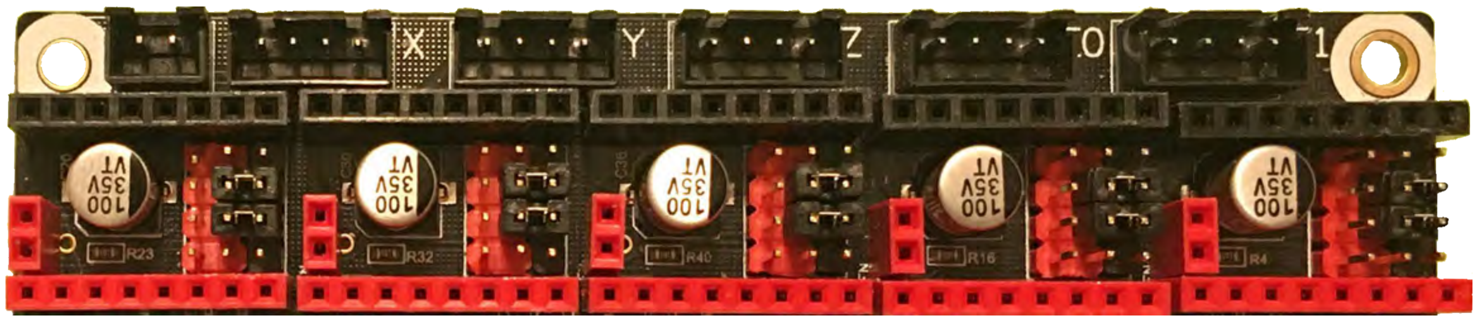


See Appendix D for legend

1 / 4



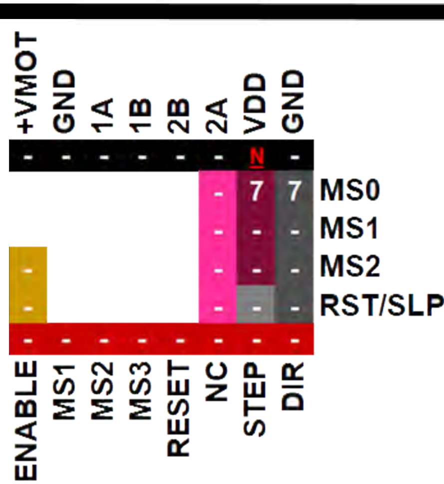
Note: **N**, on a 3.3 V MCU, like SKR V1.13 **MUST** be disconnected!



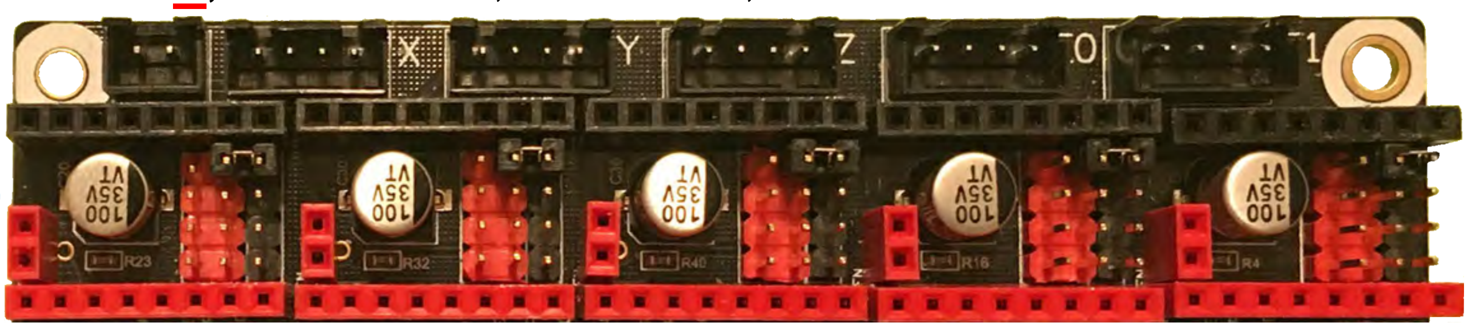
See Appendix D for legend

CW

1 / 2



Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



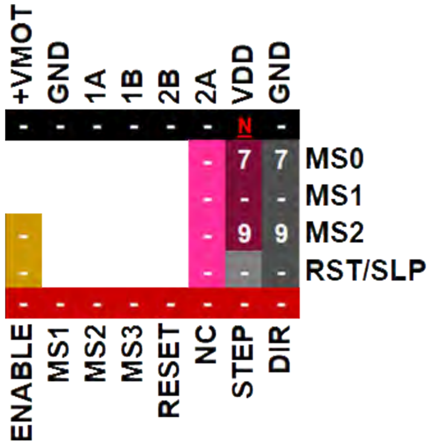
See Appendix D for legend

Stand-alone Mode

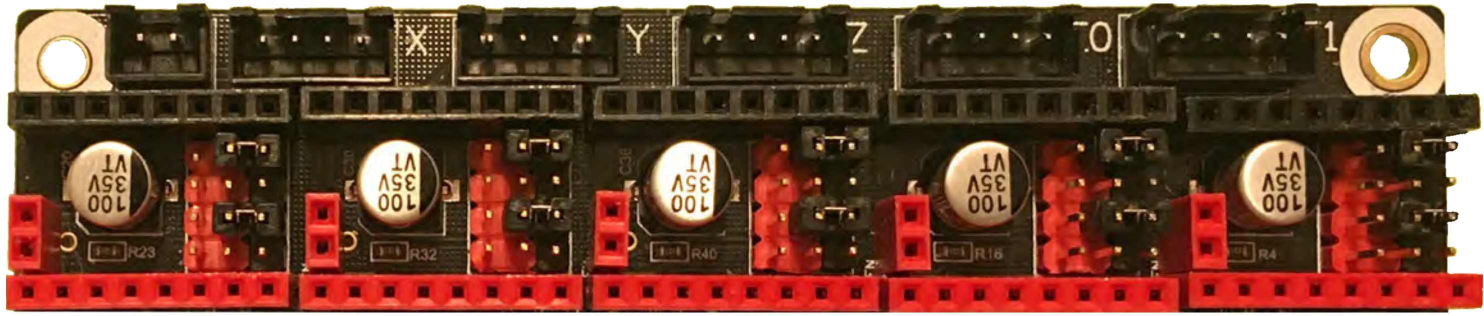
FYSETC S109

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

1 / 8

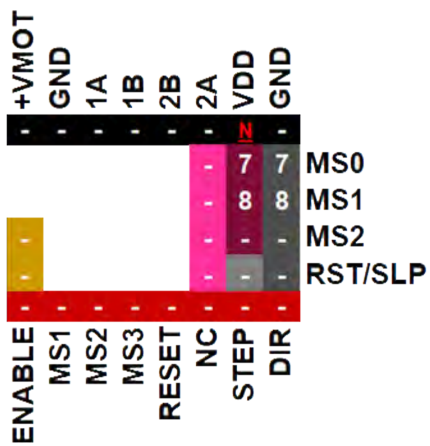


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

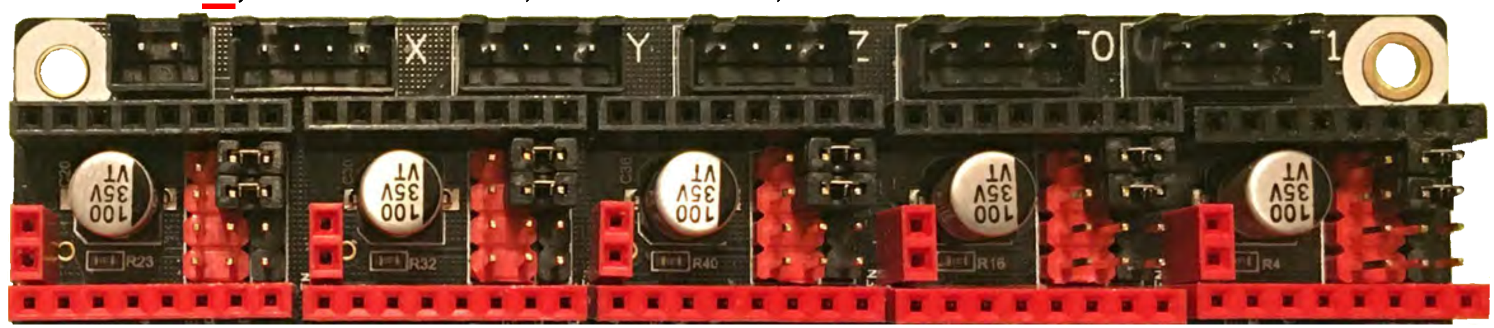


See Appendix D for legend

1 / 16

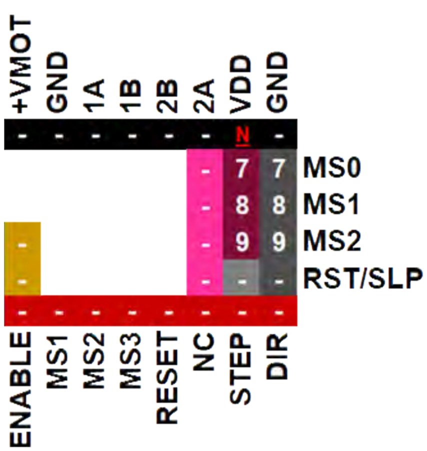


Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



See Appendix D for legend

1 / 32



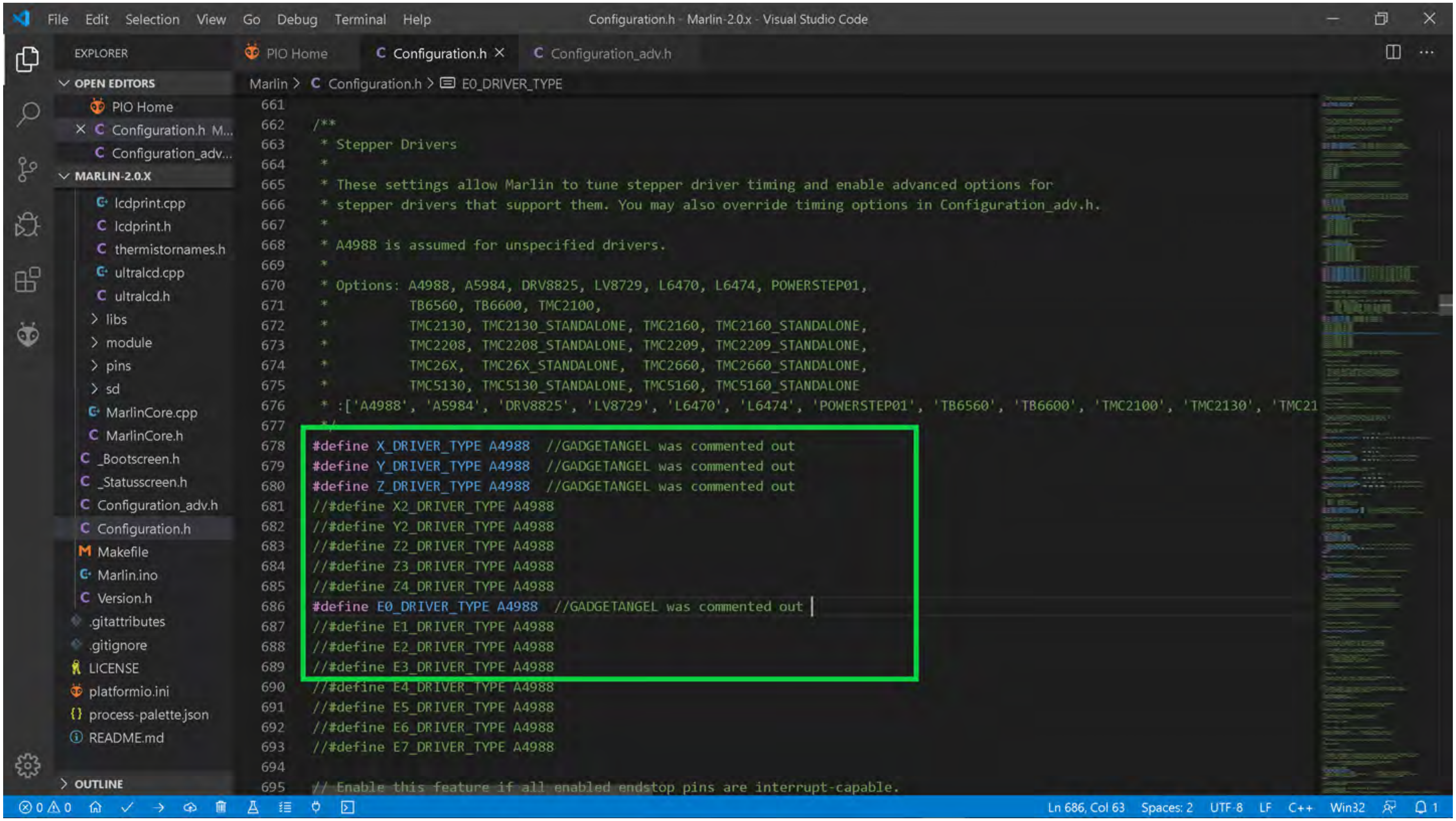
Note: **N**, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

See Appendix D for legend

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

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

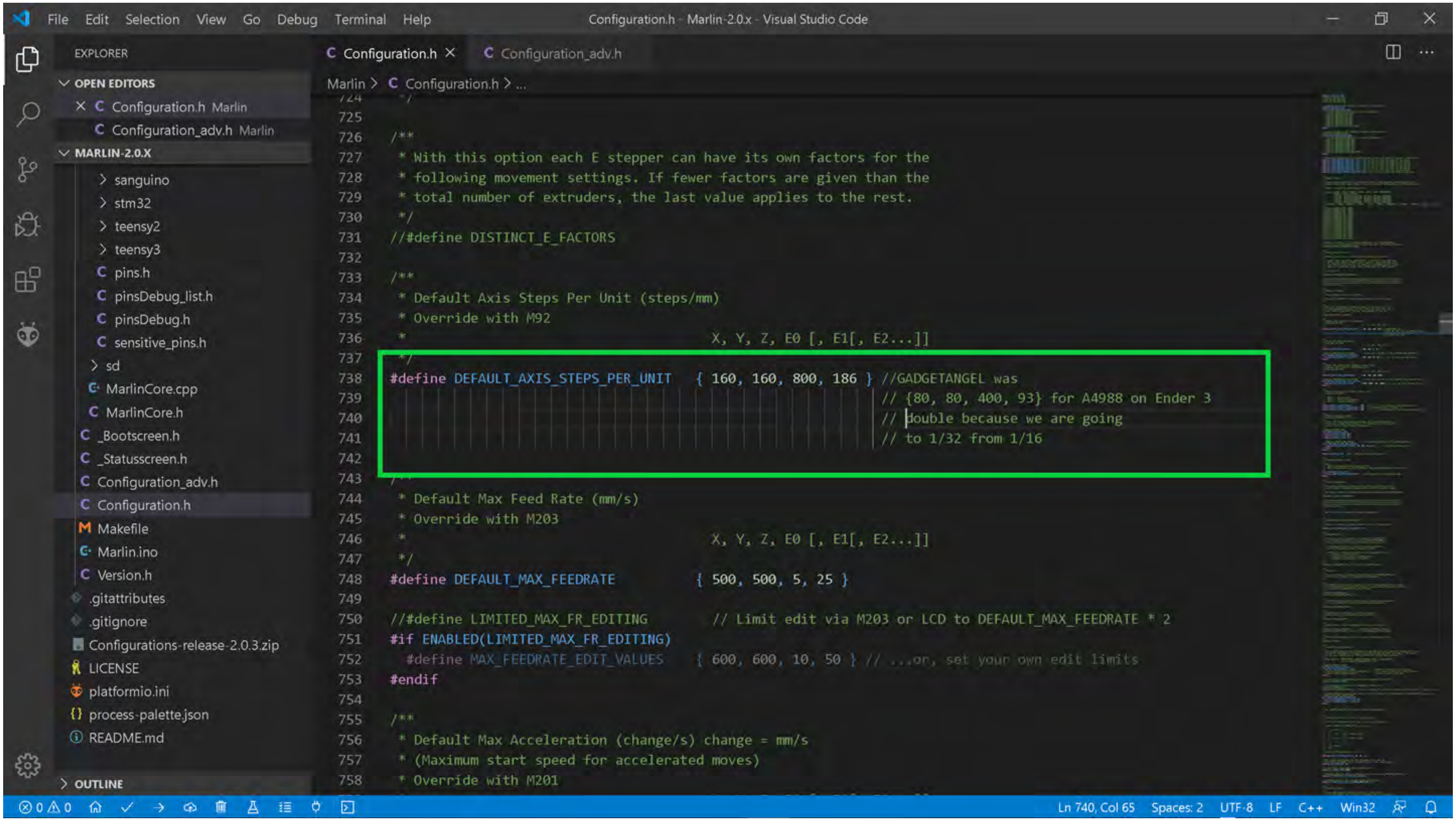
- Change the stepper motor drivers so that Marlin knows you are using FYSETC S109 drivers. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use FYSETC S109 drivers. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").
- The FYSETC S109 is a drop in replacement for the A4988. Since Marlin does not have an option for FYSETC S109 we will use the A4988 as the driver type.



- Go to the next page.

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

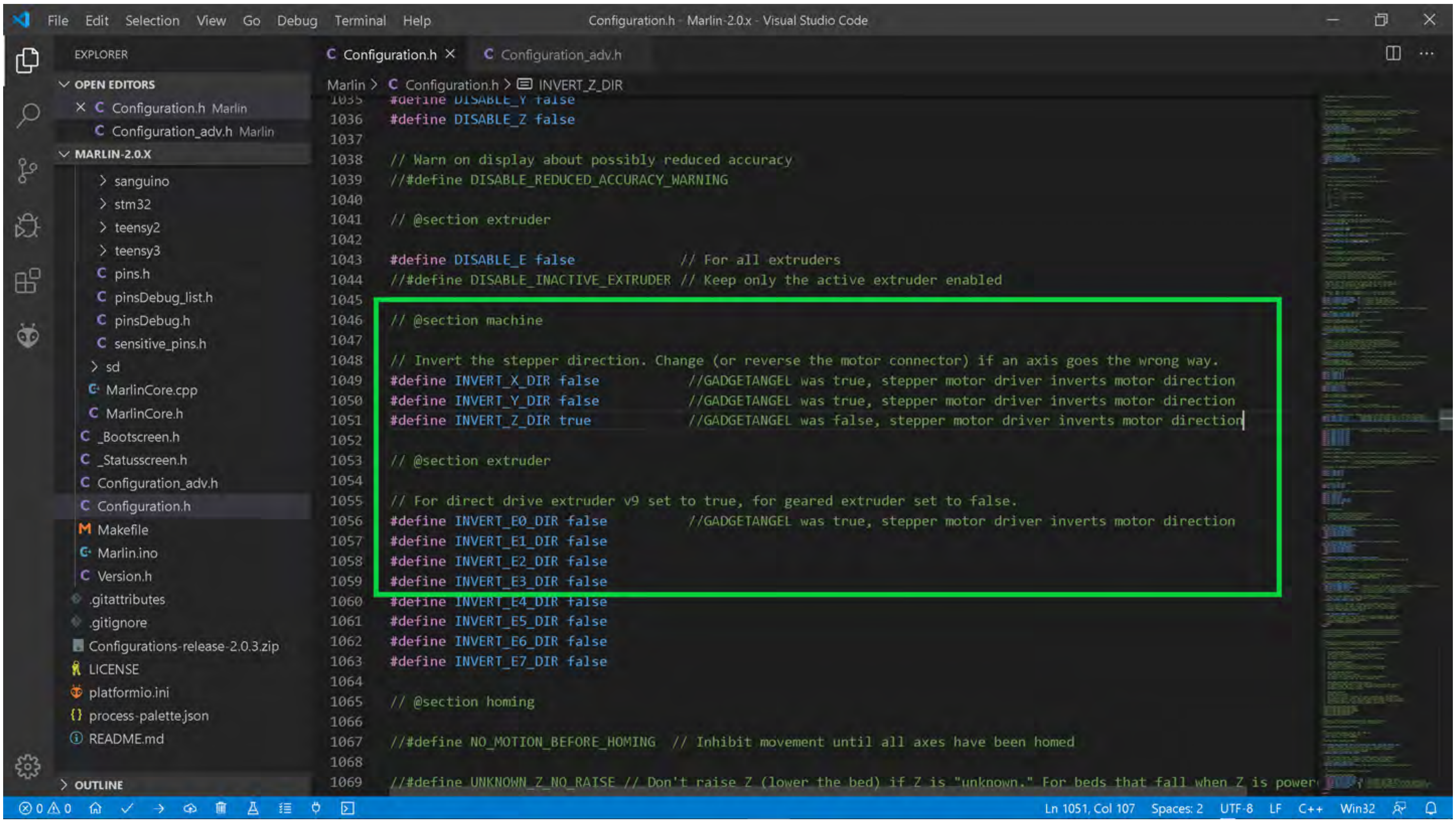
- Since we are changing from A4988 stepper motor drivers on the Ender 3 to for FYSETC S109 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

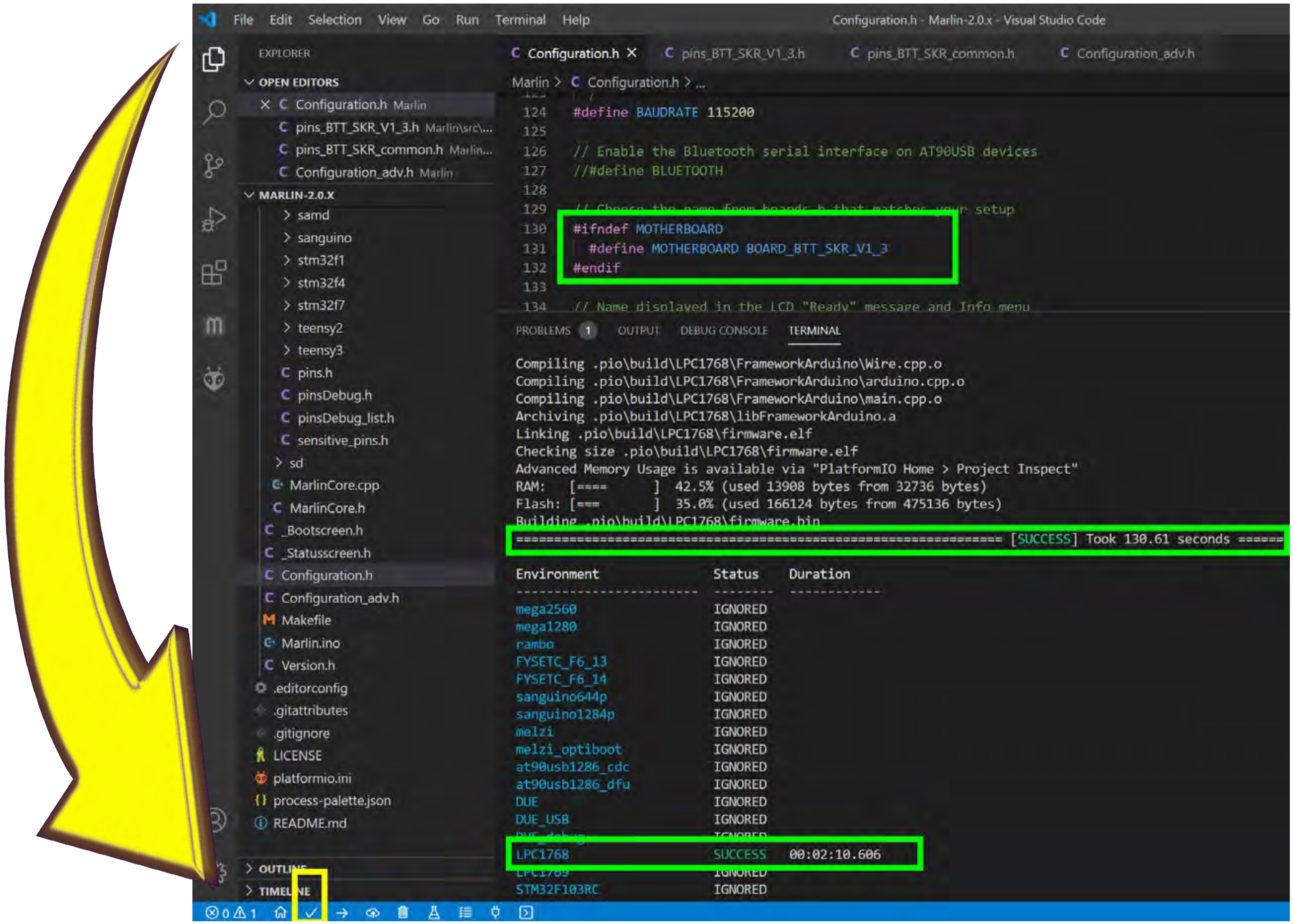
- Optional Step: I cannot find information on the FYSETC S109 driver's impact on motor direction. So I provide the below information in case you do need to change the stepper motor direction. If you prefer to change the motor direction with wiring instead of the Marlin firmware, here is a link on how to change the motor direction via the wiring (look for section labeled "Motor moving the wrong direction") **https://reprap.org/wiki/Stepper_wiring**. Other people prefer to change the motor direction in the Marlin firmware. So if you want or need to change the motor direction in Marlin, then if the axis' setting you will be using the S109 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

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

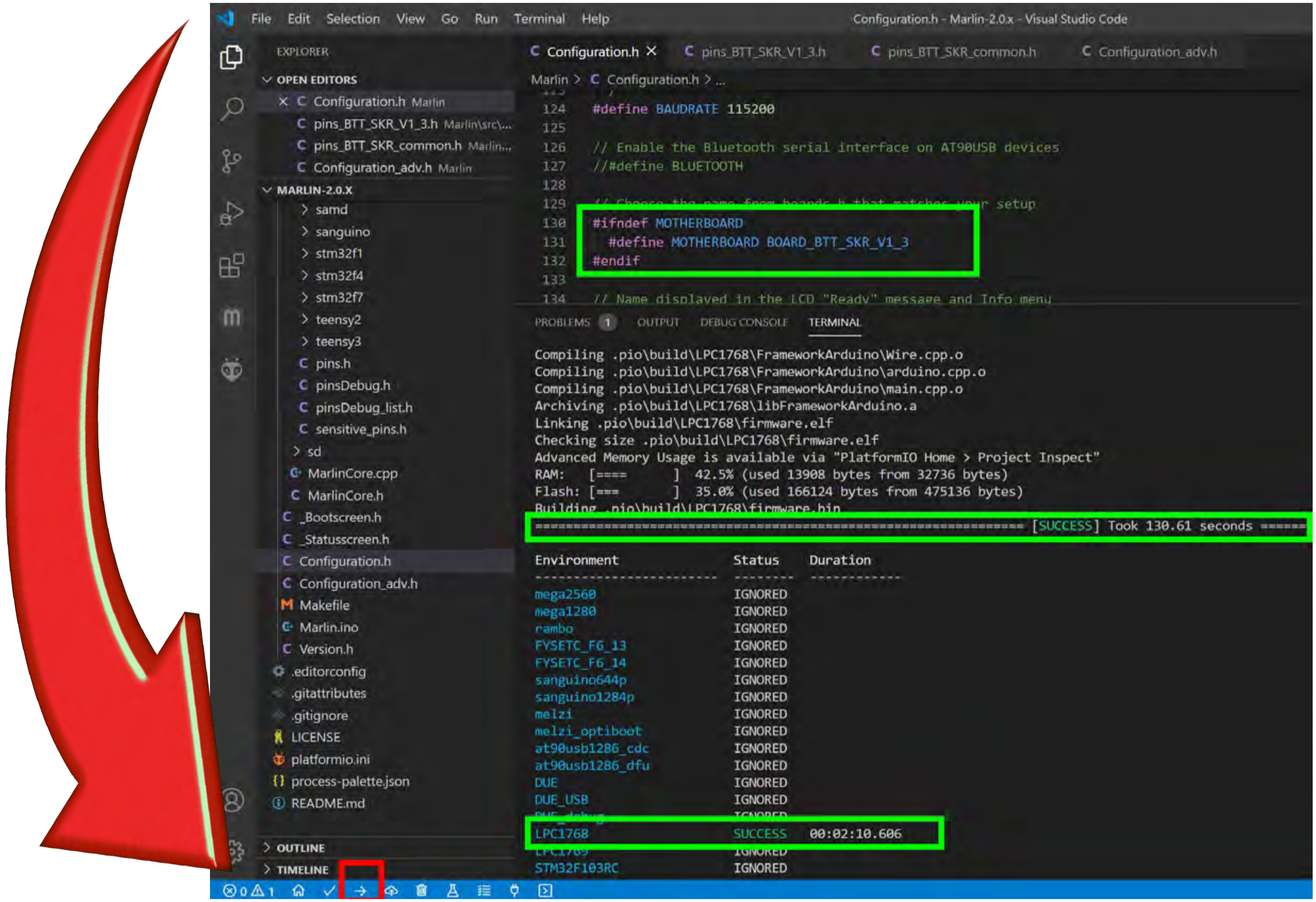
- The end of Marlin setup for FYSETC S109 drivers. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



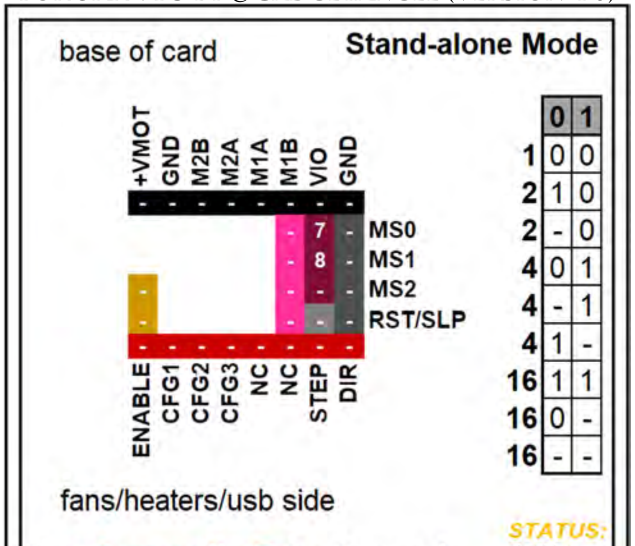
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

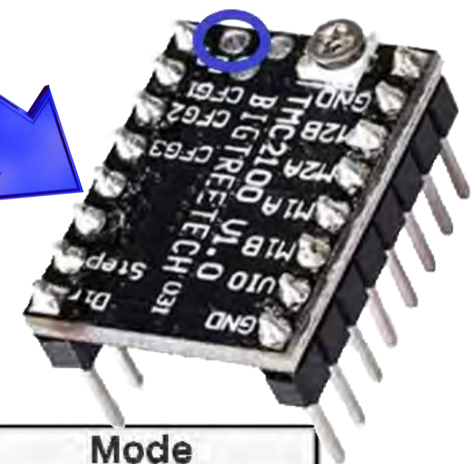


See Appendix D for legend

BIQU TMC2100

Stand-alone Mode

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



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

Driving Current Calculation Formula	$I_{MAX} = V_{ref}$	$V_{ref} = I_{MAX}$
R_S(Typical Sense Resistor)$= 0.11\Omega$	See Appendix B #1. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	See Appendix B #1. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

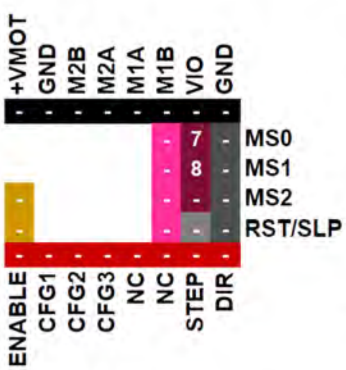
BIQU TMC2100

Stand-alone Mode

SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1

base of card

Stand-alone Mode



	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

OPEN → No Jumper

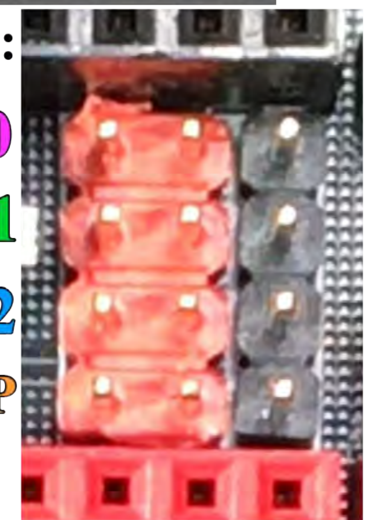
Rows:

MS0

MS1

MS2

RST/SLP



Driver Chip Chart:

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

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

High → Jumper Set

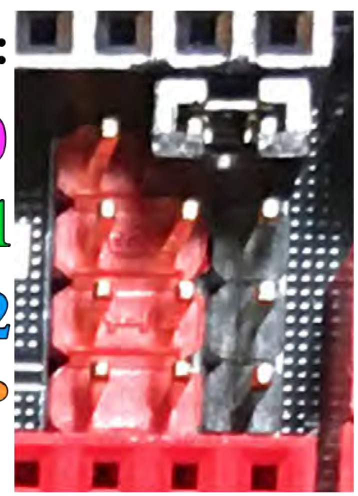
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

BIQU TMC2100

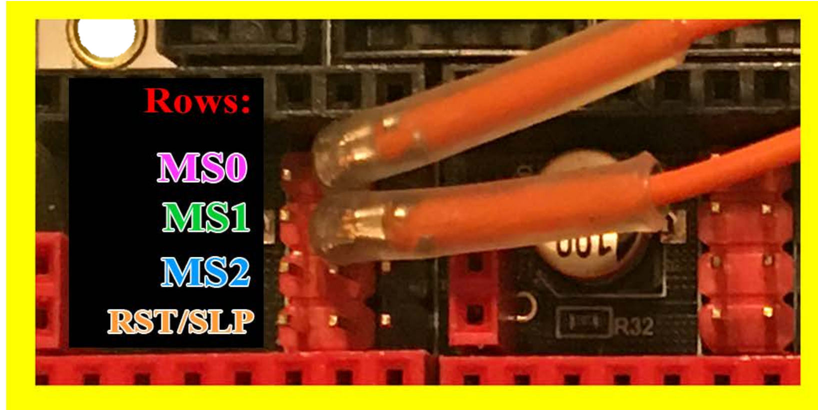
Stand-alone Mode

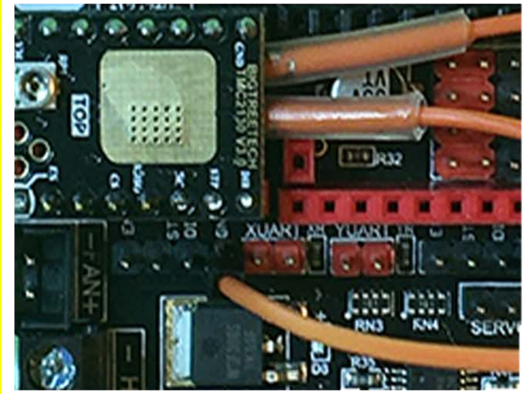
Driver Chip Chart:

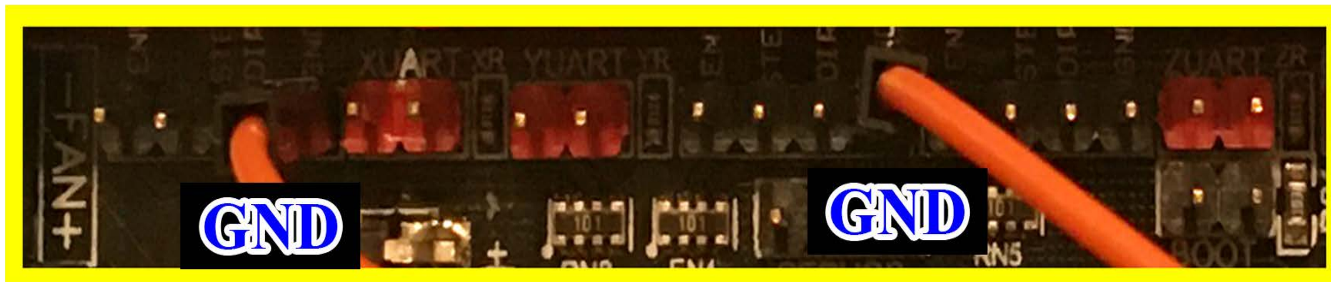
Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMCxxxx Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1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

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

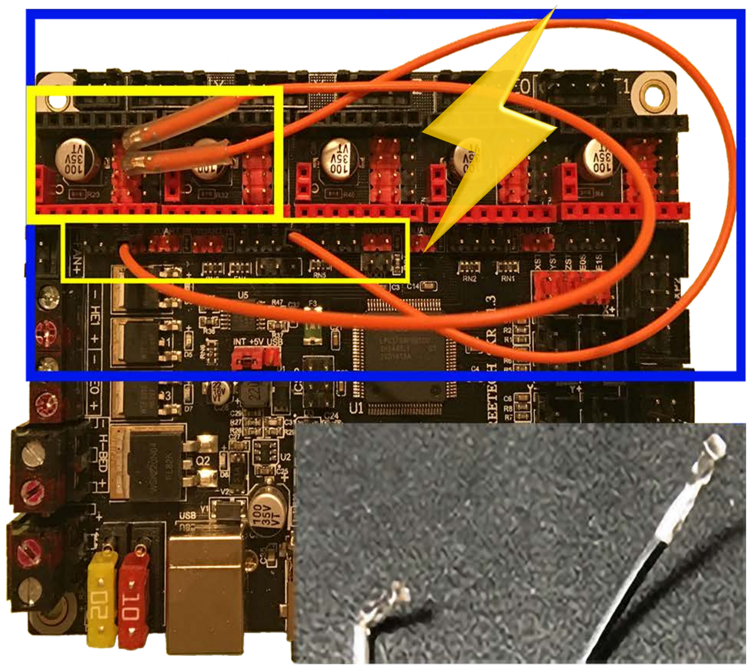
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 2

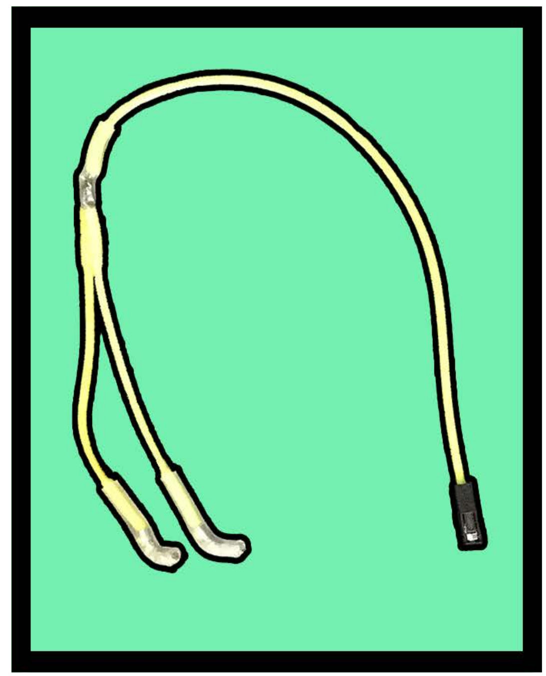






Low MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable





Note: See next page for instructions on how to create a DuPont jumper cable.

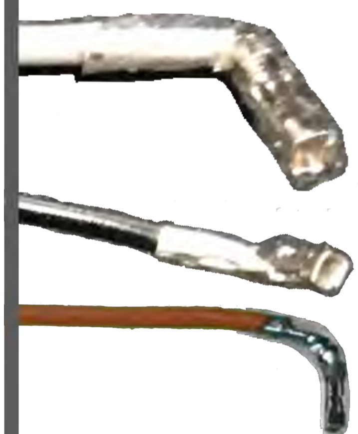
BIQU TMC2100

Stand-alone Mode

How to Create a SKR V1.3 Dupont Jumper Cable to Use with Tri State Drivers

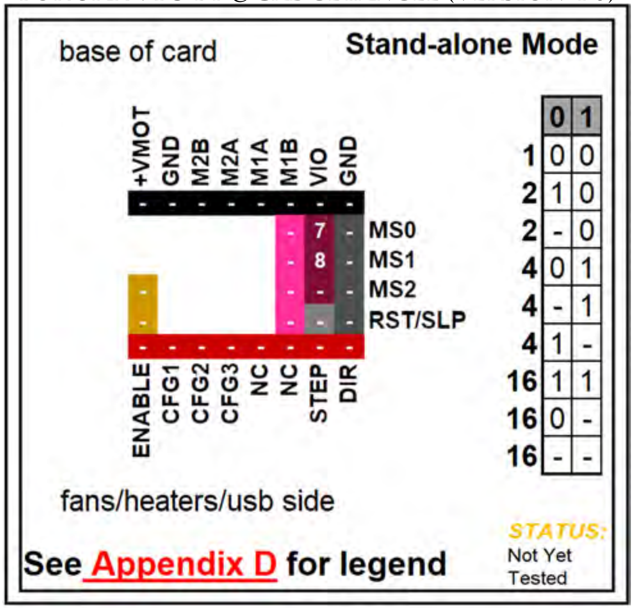
Remove DuPont cover from one end of a female to female DuPont jumper wire.

Place a single DuPont pin into exposed female end and bend the connector backwards 90 degrees.



Since bending the connector weakens it, place a bead of solder at the 90 degree bend. I inserted a single DuPont pin inside the female connector to ensure that the connector stays clear of solder. After soldering is done, move the inserted DuPont pin around to ensure a functioning right angle female connector.

Place heat shrink over the exposed female DuPont connector to protect the exposed metal from causing shorts. Picture here is a finished jumper cable with a single DuPont pin inserted.



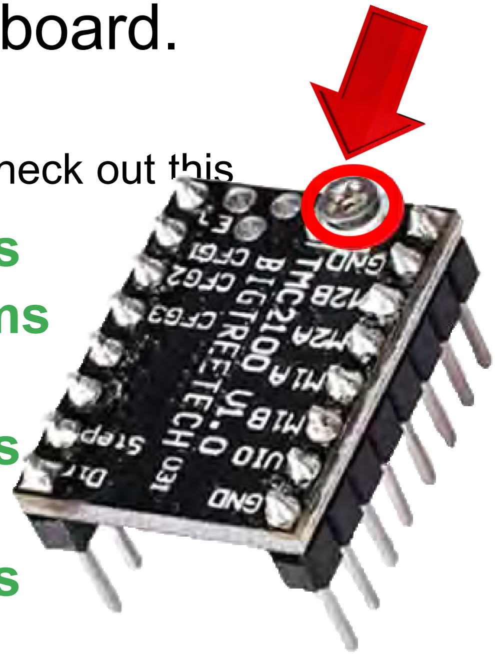
	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

BIQU TMC2100

Stand-alone Mode

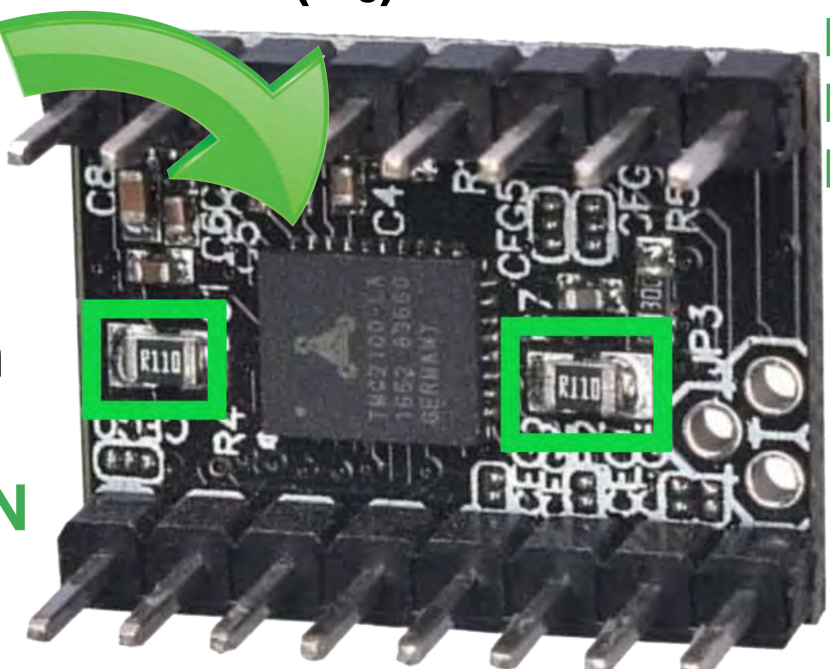
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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



To learn more, please watch this **YouTube video done by Teaching Tech** and check out this link on the **TMC2100 Driver**

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



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

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

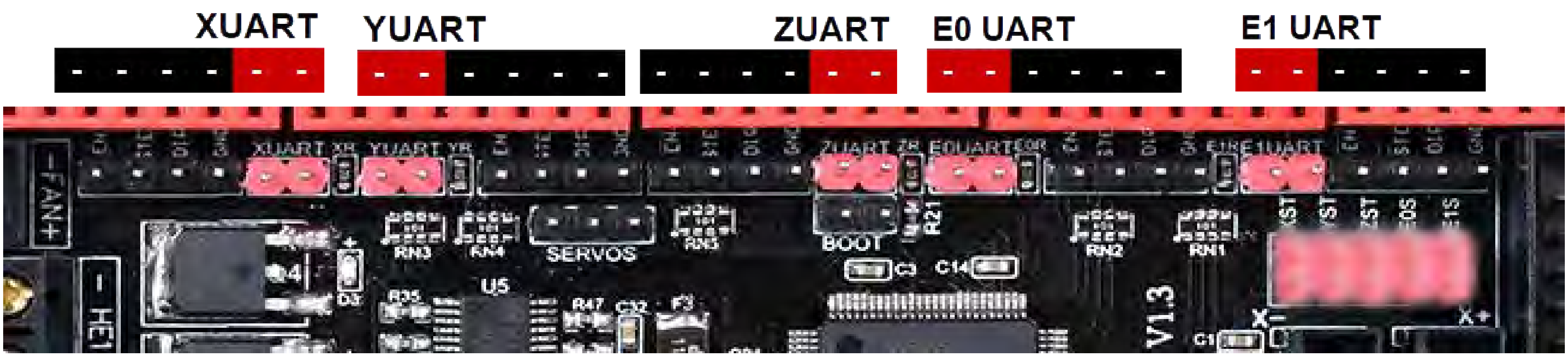
Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

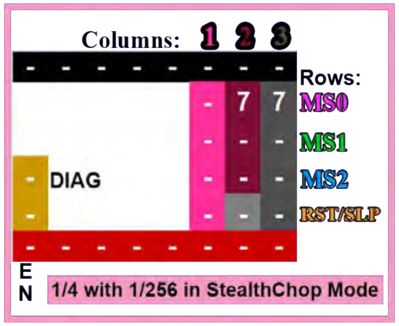


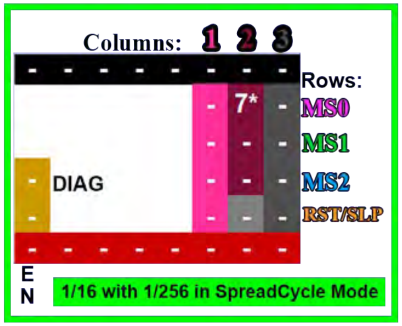
BIQU TMC2100

Stand-alone Mode

SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

Driver Socket Representation





Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX} = V_{ref}$ See Appendix B. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$			$V_{ref} = I_{MAX}$ See Appendix B. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$	

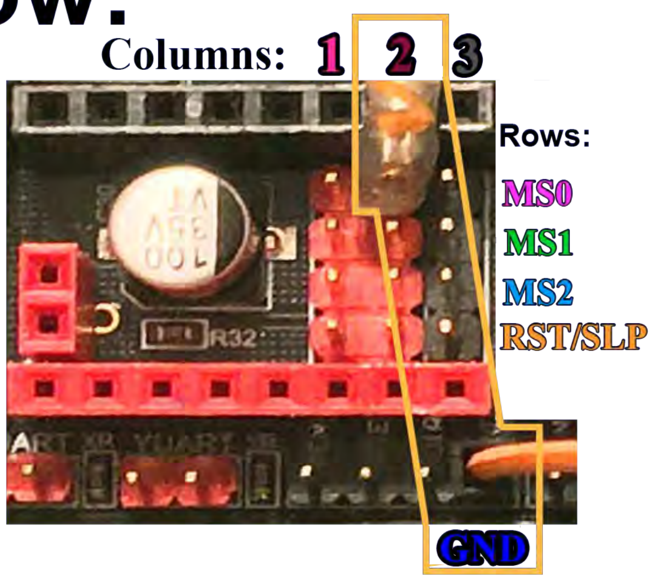
- - → No Jumper set

MS0 for Tri State Drivers ONLY (TMC2100 & TMC2130):

Driver Socket Representation:		Driver Chip Chart:
7 7	→	High
7* -	→	Low
- -	→	Open

Driver Chip Chart:		Meaning:
High	→	set Jumper between column 2 and column 3 on the MS0 row
Low	→	Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)
Open	→	No Jumper set

Low:

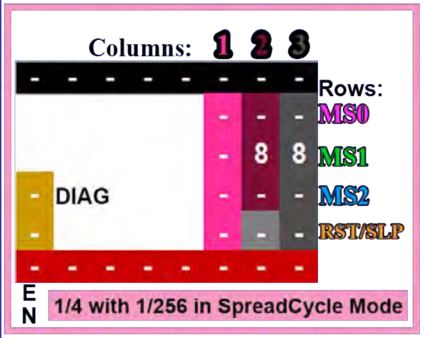


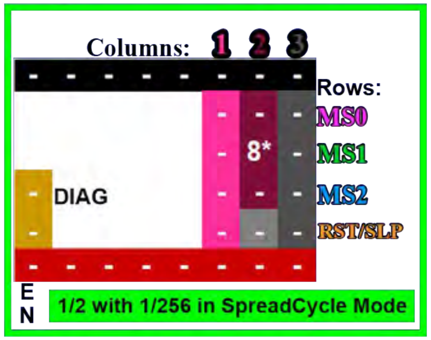
High:

BIQU TMC2100

Stand-alone Mode

Driver Socket Representation



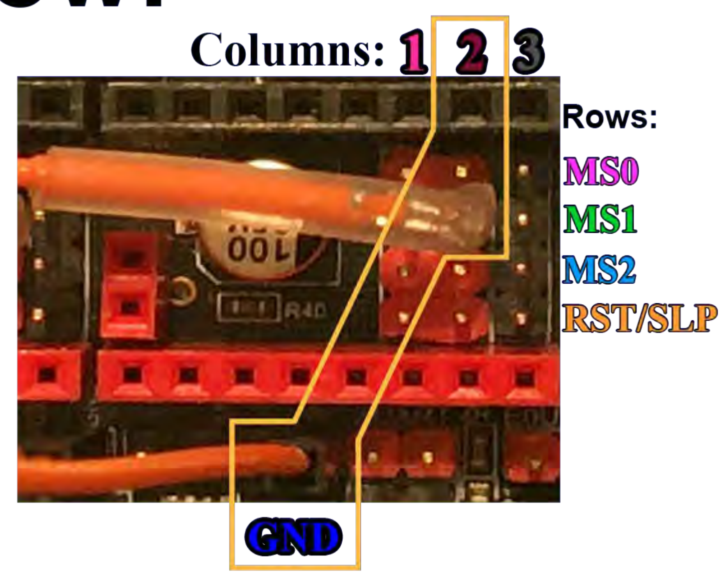


Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B #x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$			$V_{ref}=I_{MAX}$ See Appendix B #x. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$	

- - ⇒ No Jumper set

Low:



MS1 for Tri State Drivers ONLY (TMC2100 & TMC2130):

Driver Socket Represntation: 8 8 ⇒ Driver Chip Chart: High

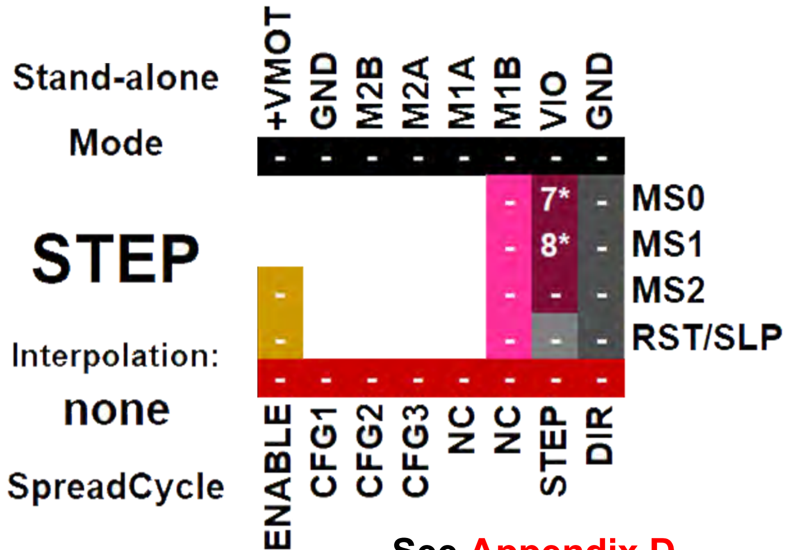
Driver Socket Represntation: 8* - ⇒ Driver Chip Chart: Low

Driver Socket Represntation: - - ⇒ Driver Chip Chart: Open

Driver Chip Chart:	Meaning:
High ⇒	set Jumper between column 2 and column 3 on the MS1 row
Low ⇒	Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)
Open ⇒	No Jumper set

High:

Stand-alone Mode



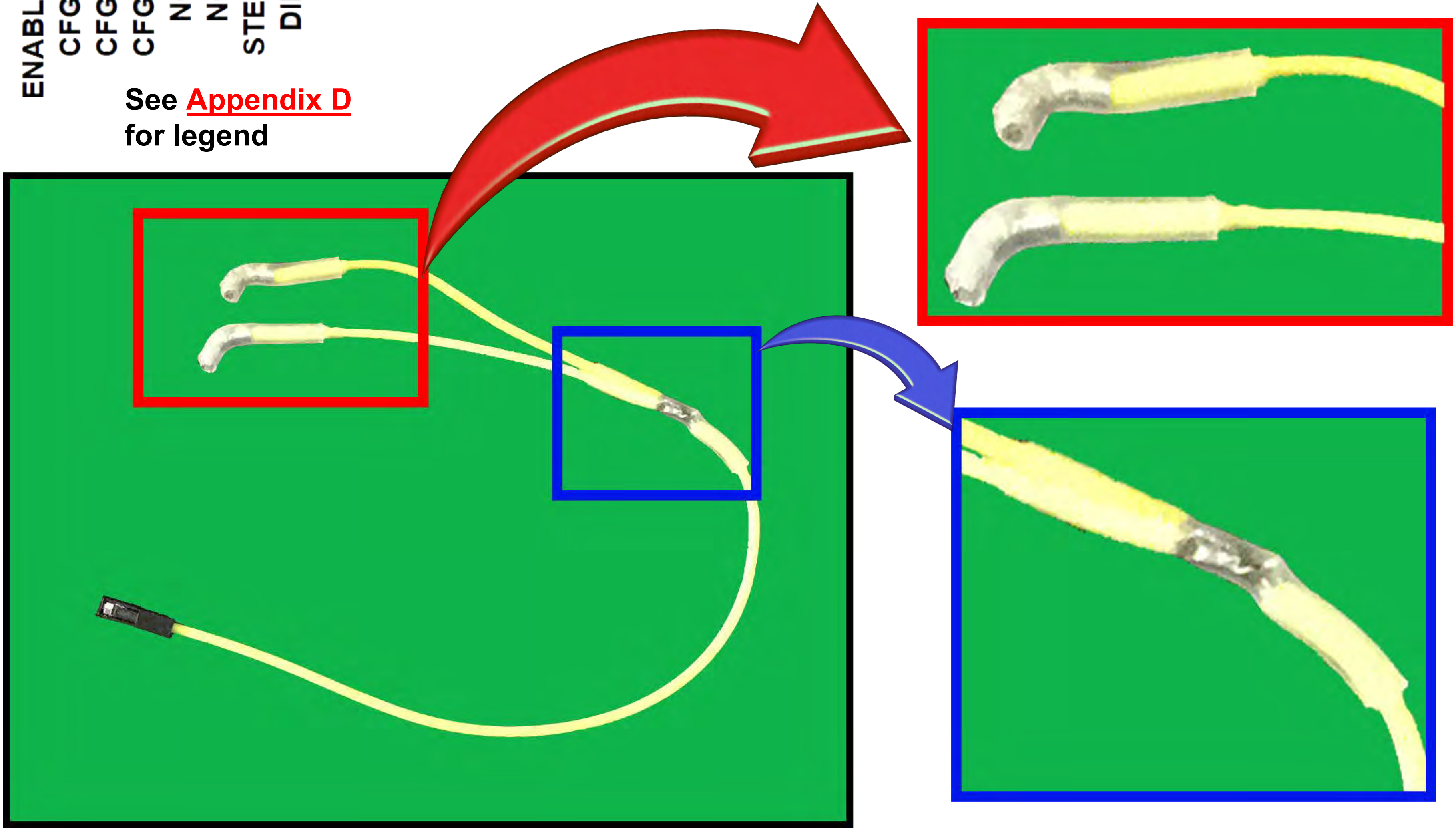
See Appendix D for legend

BIQU TMC2100

Stand-alone Mode

Additional Equipment Needed for Low Low (STEP or FULL) Configuration

You will need one DuPont jumper cable that connects to a ground point (GND) on the SKR V1.3 board but will source two PINS simultaneously. So create a Y-cable that has one female connector on one end and has two female connectors on the other end.



Stand-alone Mode

BIQU TMC2100

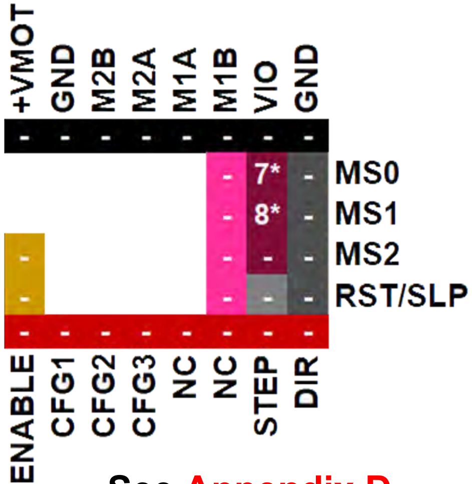
Stand-alone Mode

Stand-alone Mode

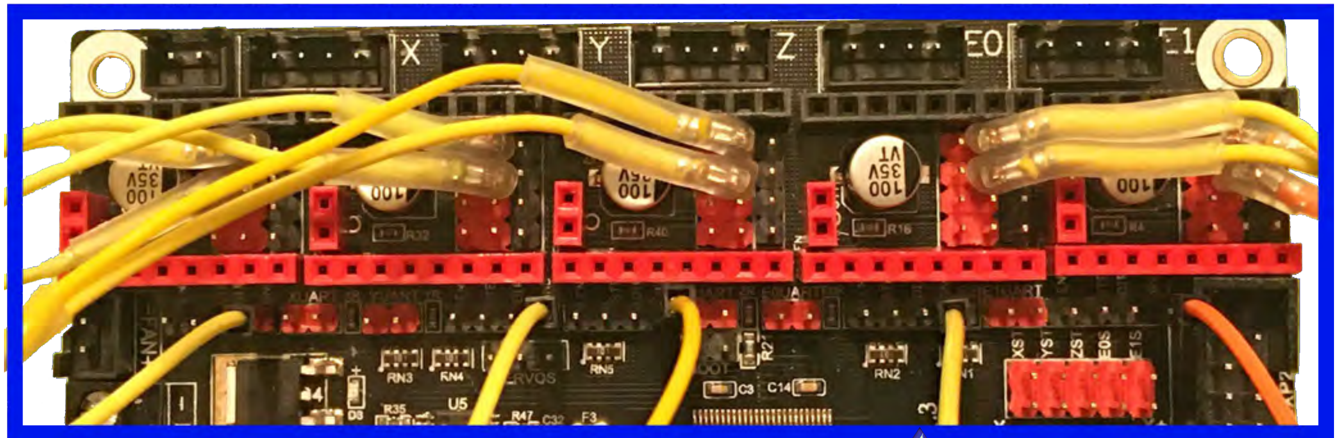
STEP

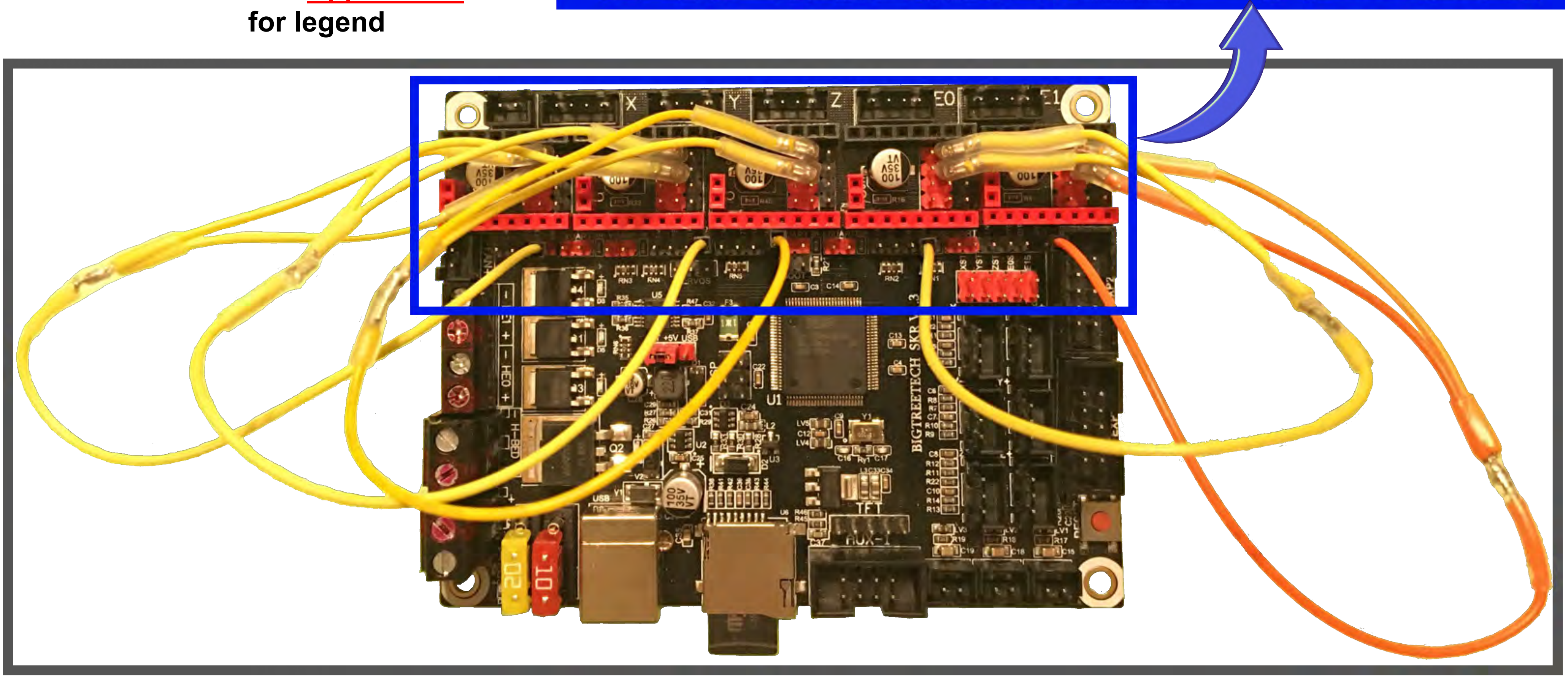
Interpolation: none

SpreadCycle



See Appendix D for legend

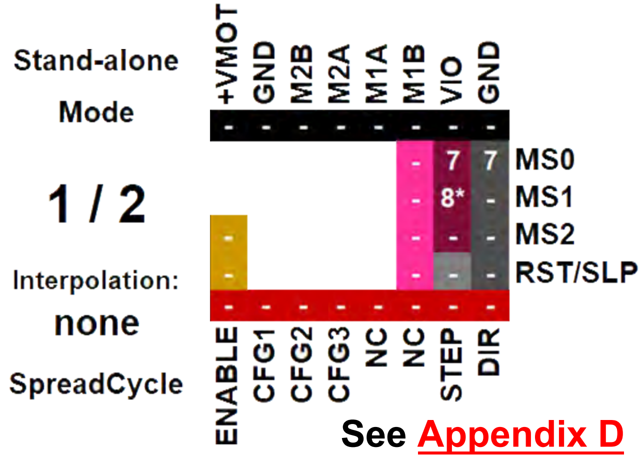




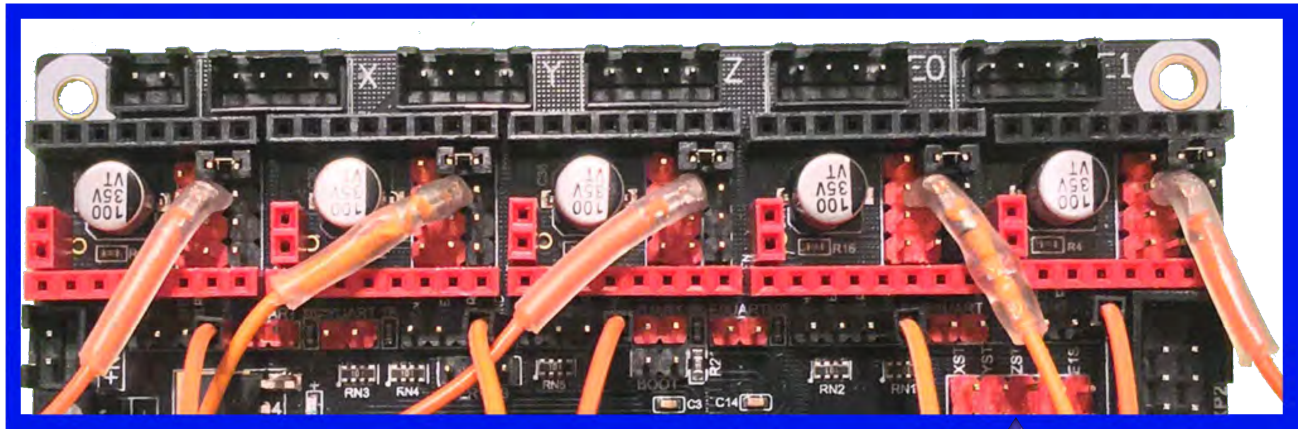
Stand-alone Mode

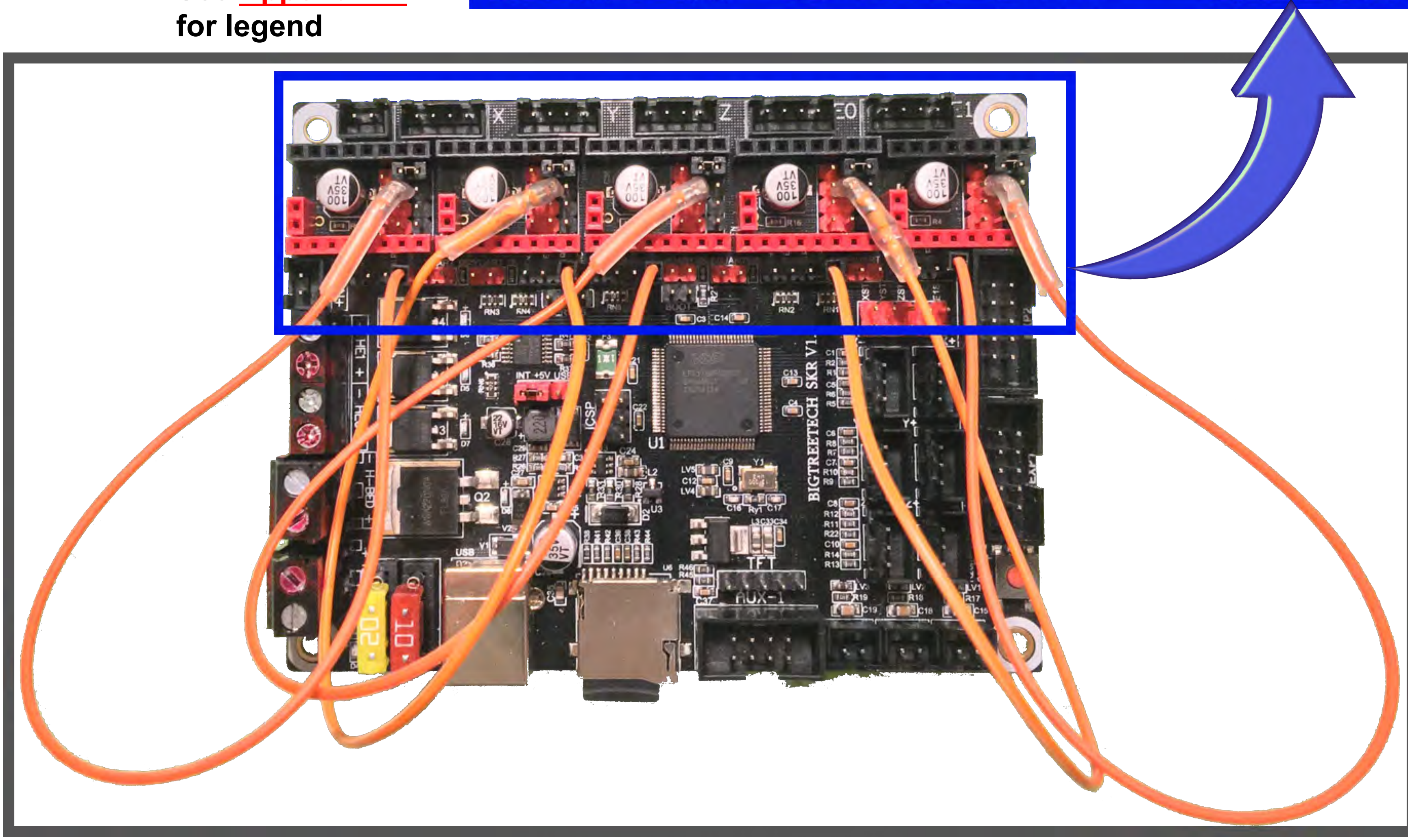
BIQU TMC2100

Stand-alone Mode



See Appendix D for legend





Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

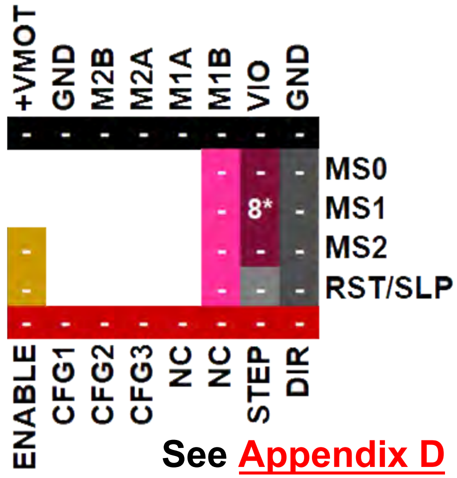
Stand-alone Mode

1 / 2

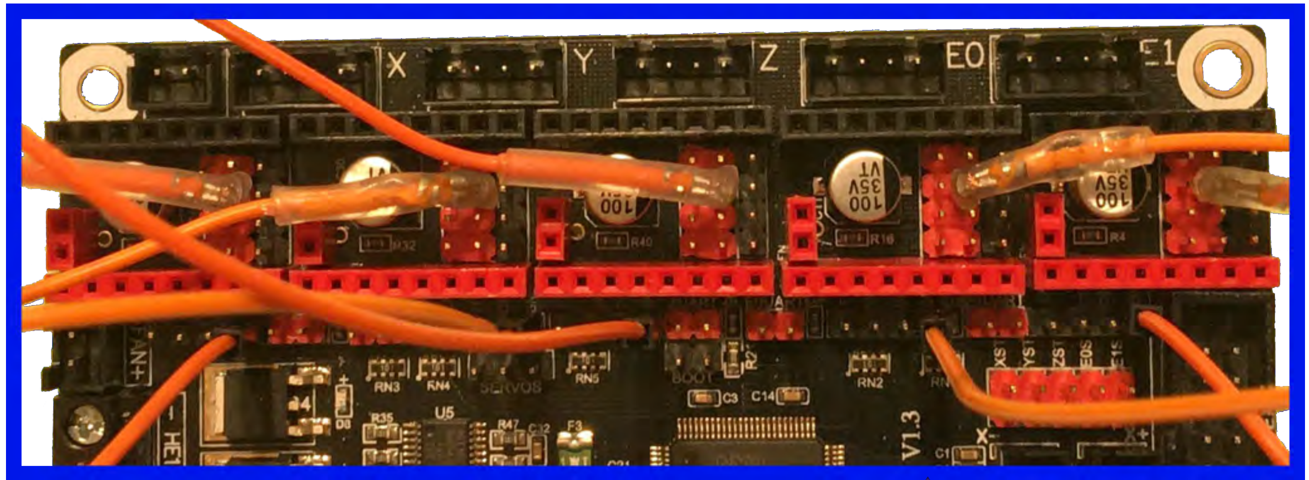
Interpolation:

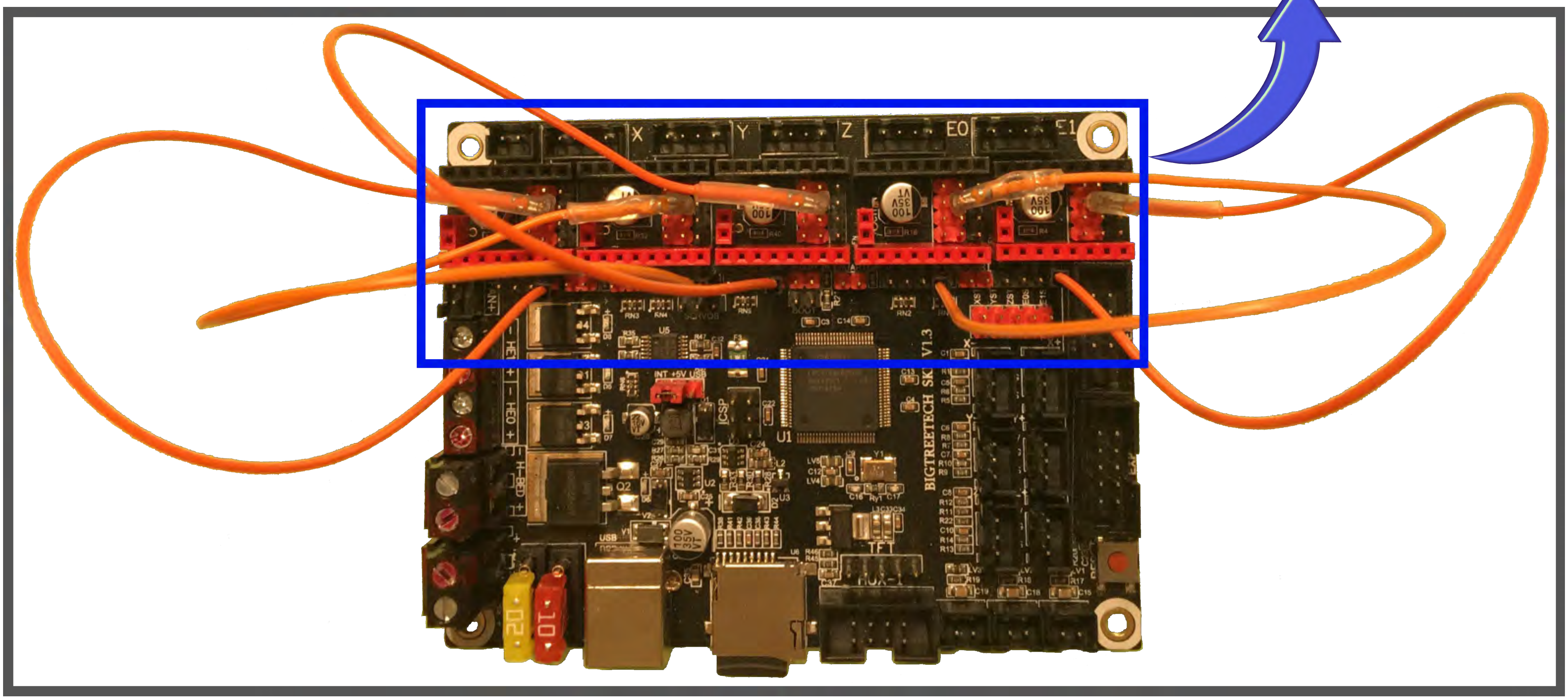
1 / 256

SpreadCycle



See Appendix D for legend





Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

Stand-alone

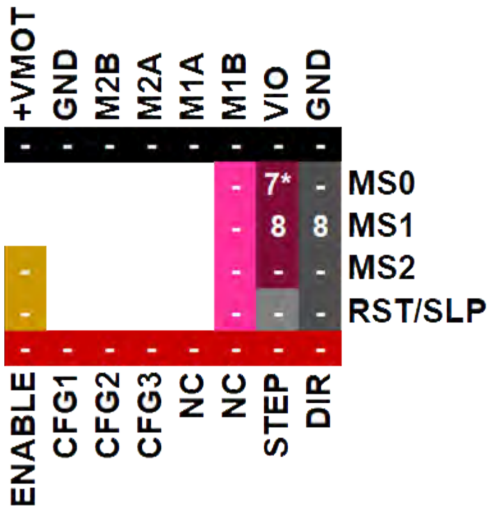
Mode

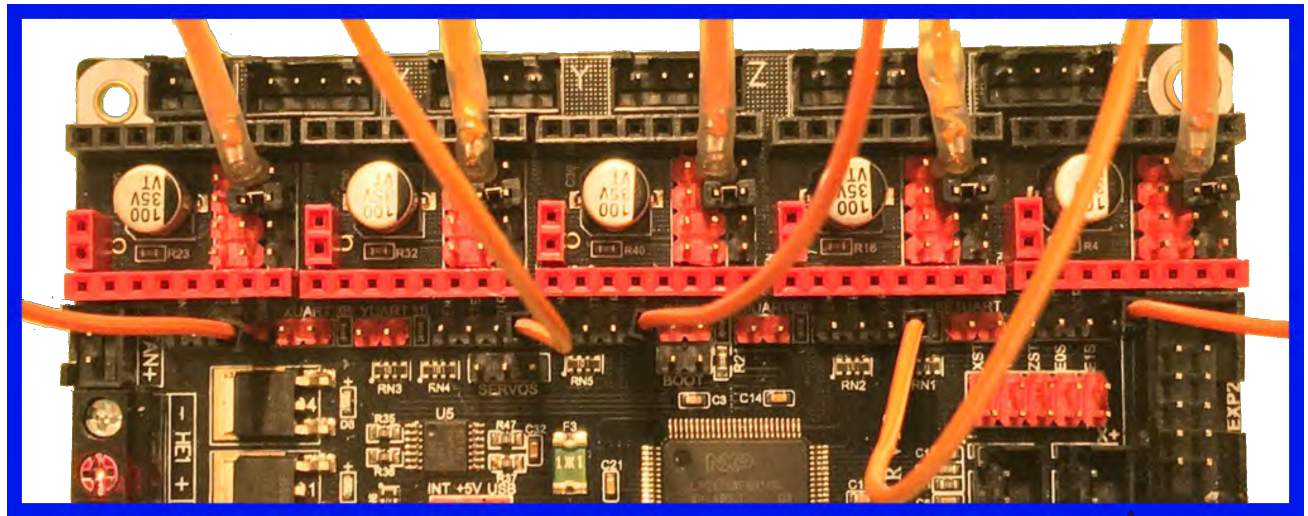
1 / 4

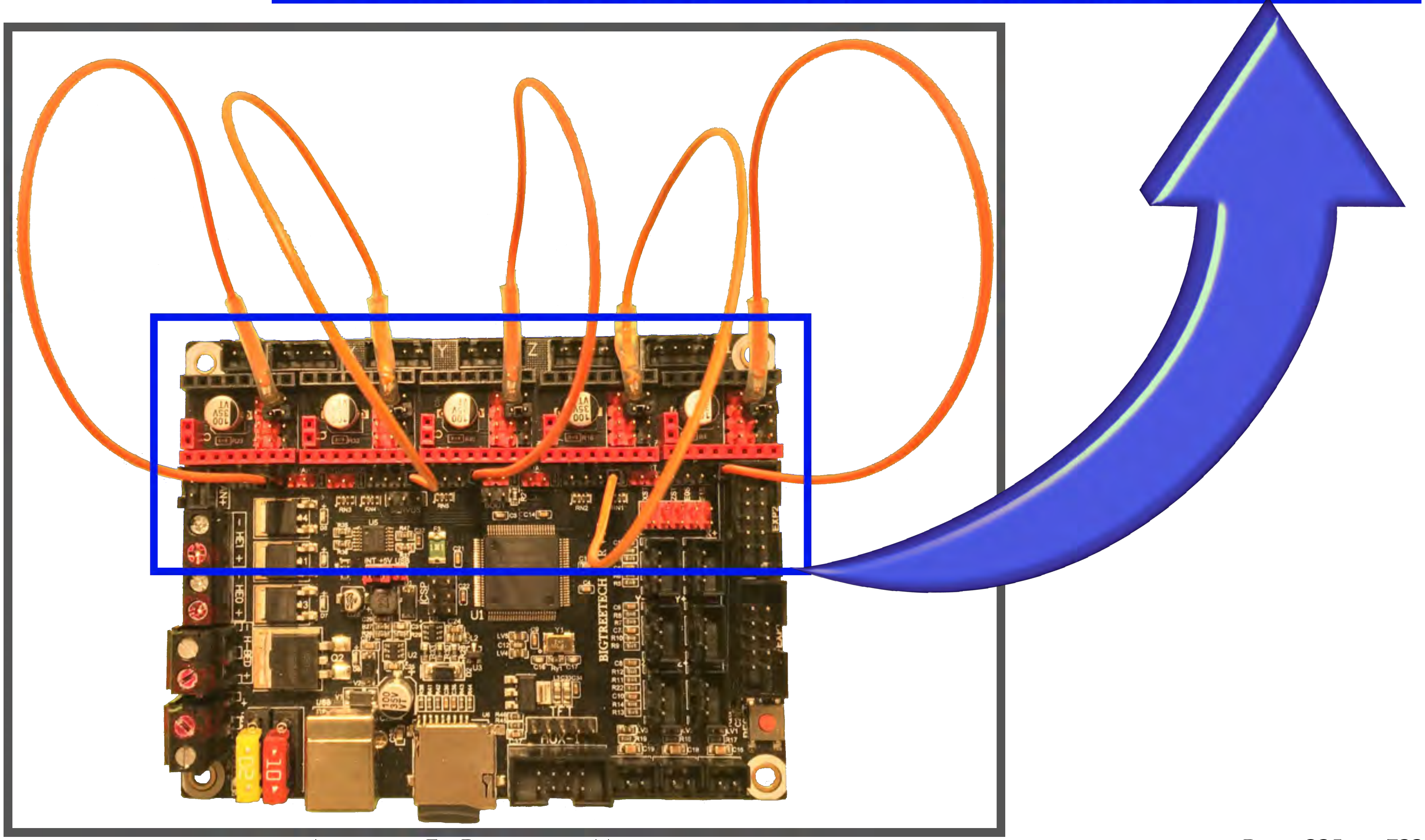
Interpolation:

none

SpreadCycle







See [Appendix D](#) for legend

Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

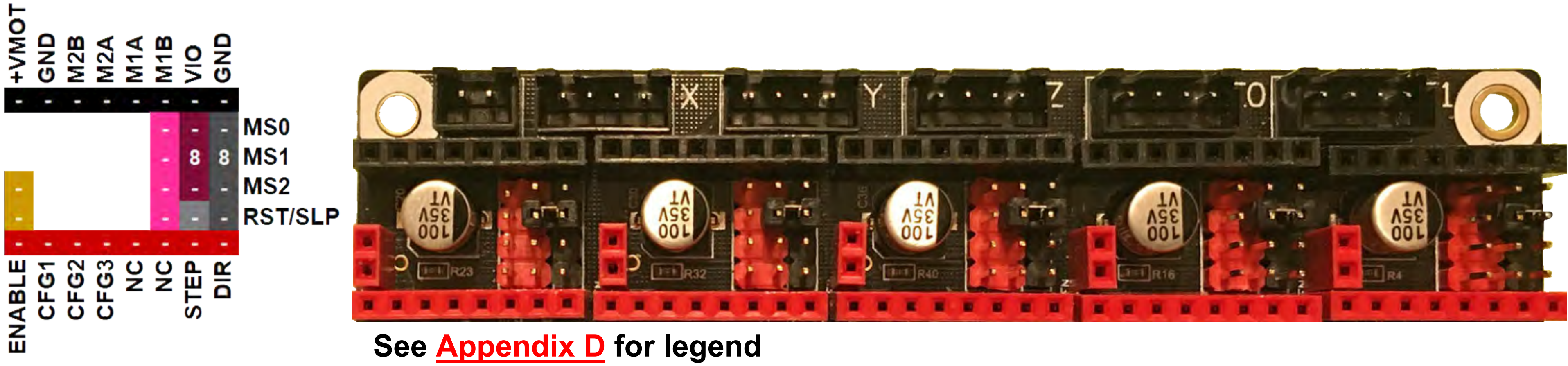
Stand-alone Mode

1 / 4

Interpolation:

1 / 256

SpreadCycle



See Appendix D for legend

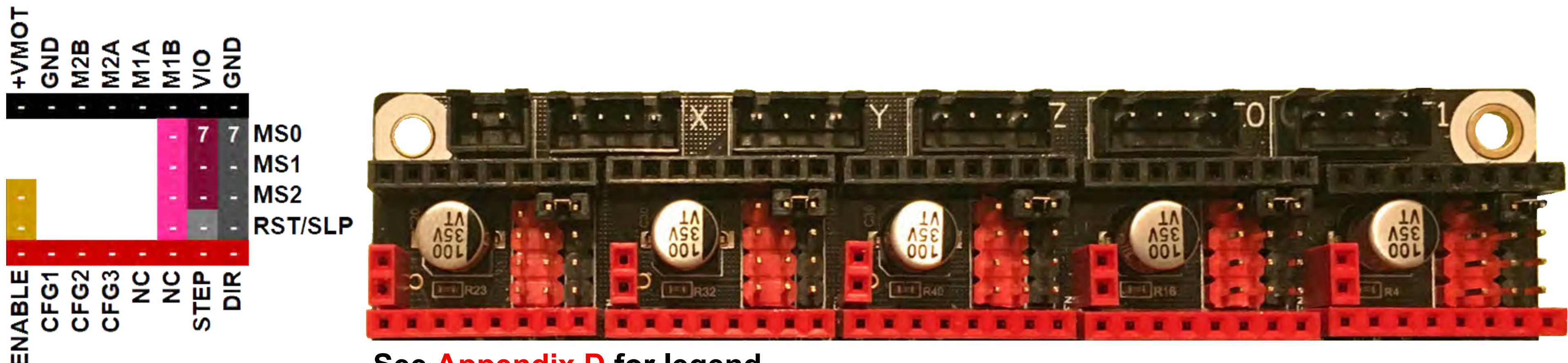
Stand-alone Mode

1 / 4

Interpolation:

1 / 256

StealthChop



See Appendix D for legend

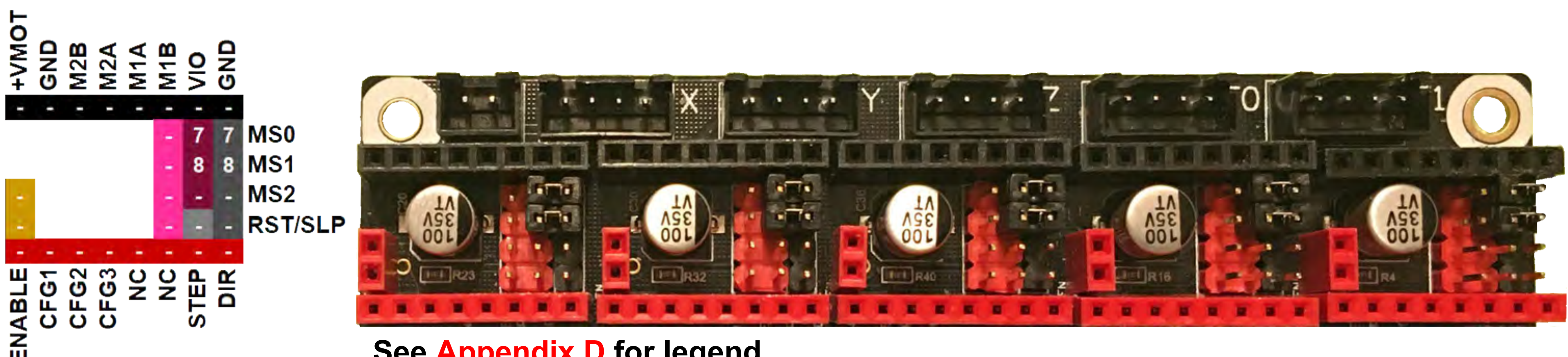
Stand-alone Mode

1 / 16

Interpolation:

none

SpreadCycle



See Appendix D for legend

Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

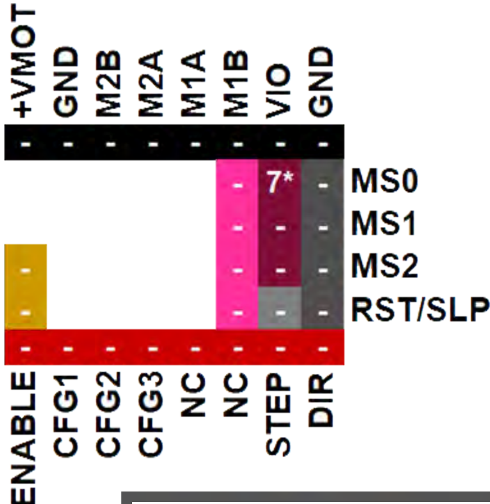
Stand-alone Mode

1 / 16

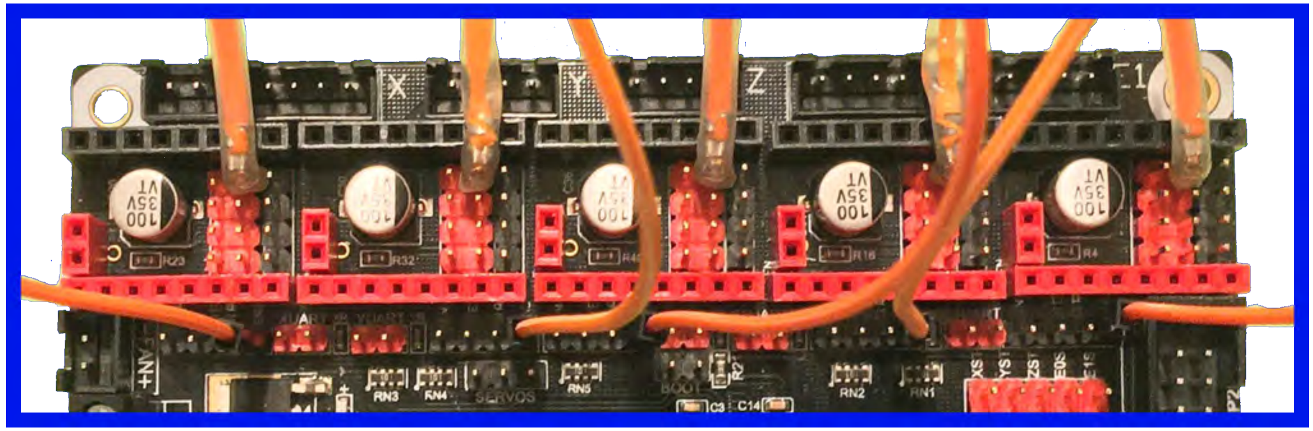
Interpolation:

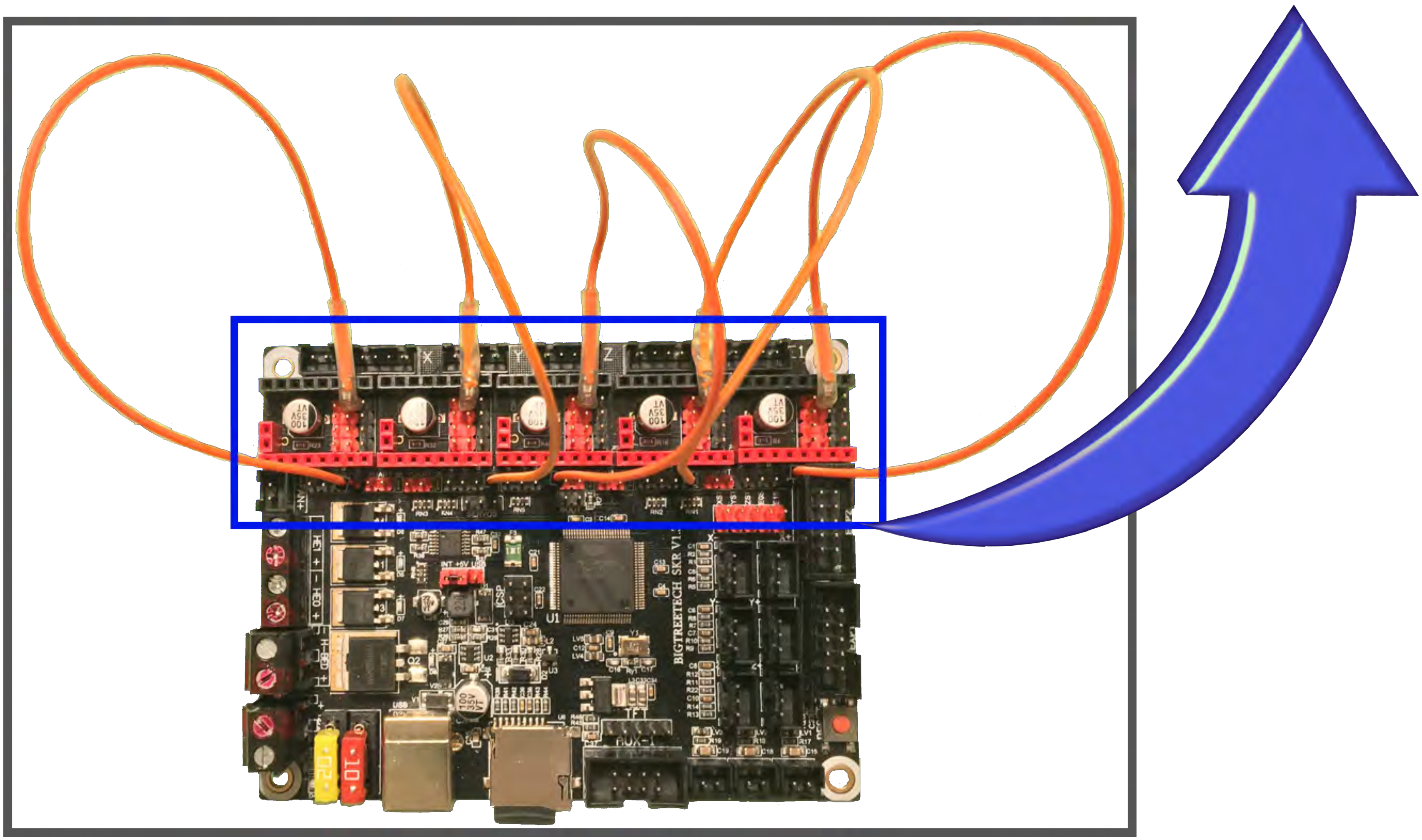
1 / 256

SpreadCycle



See Appendix D for legend





Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

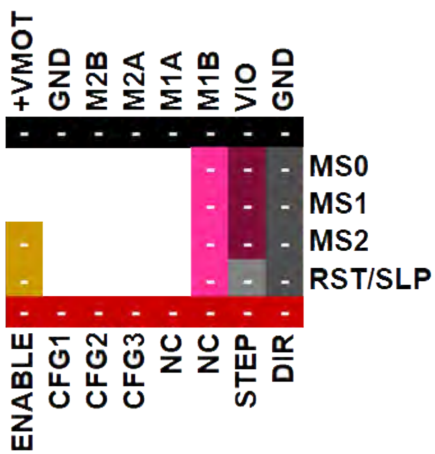
Stand-alone Mode

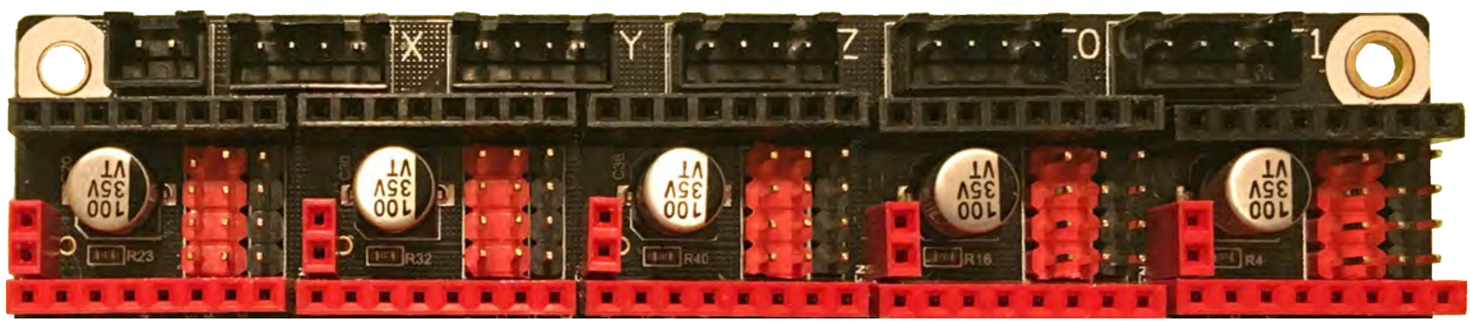
1 / 16

Interpolation:

1 / 256

StealthChop



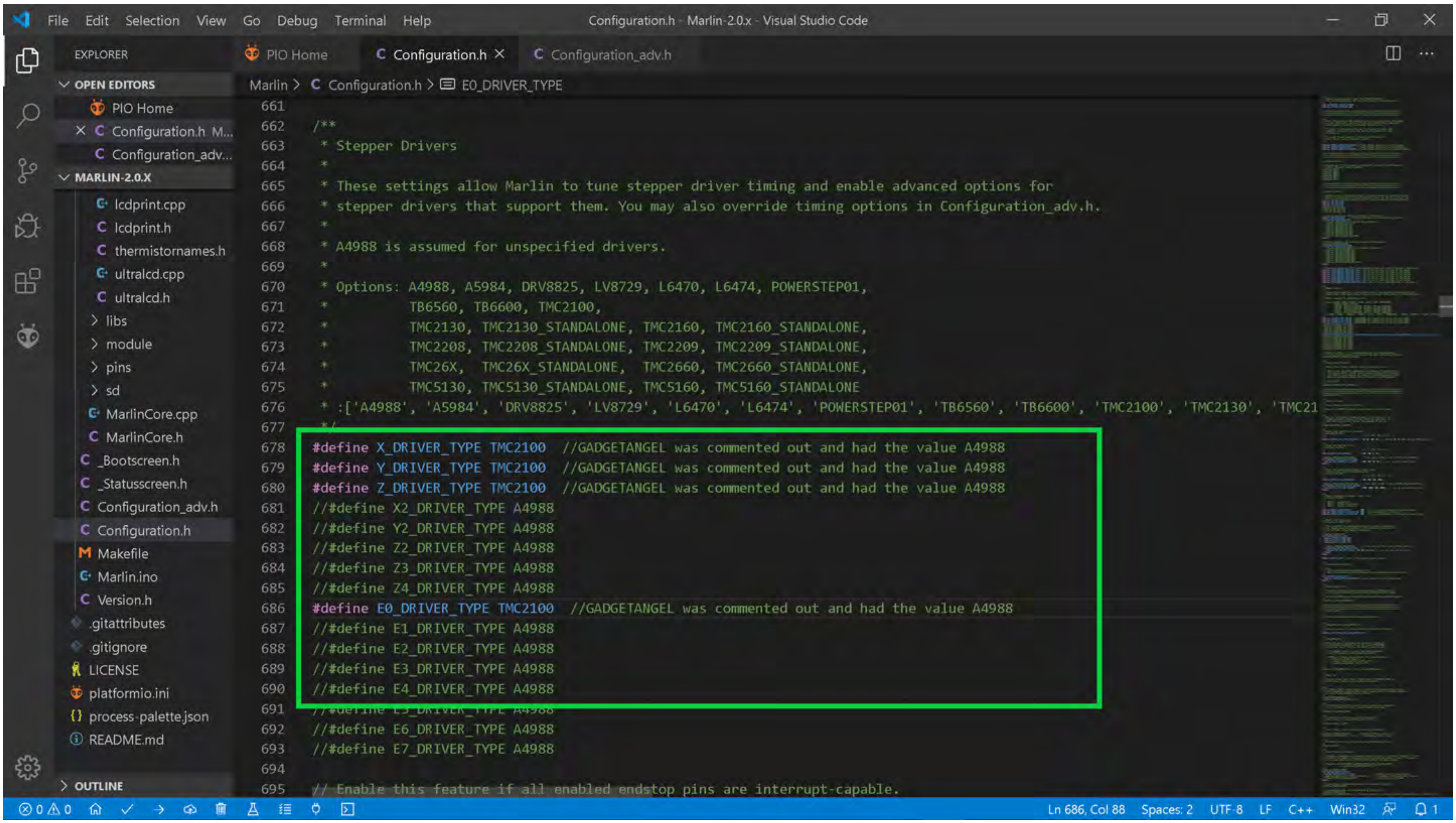


See Appendix D for legend

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

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

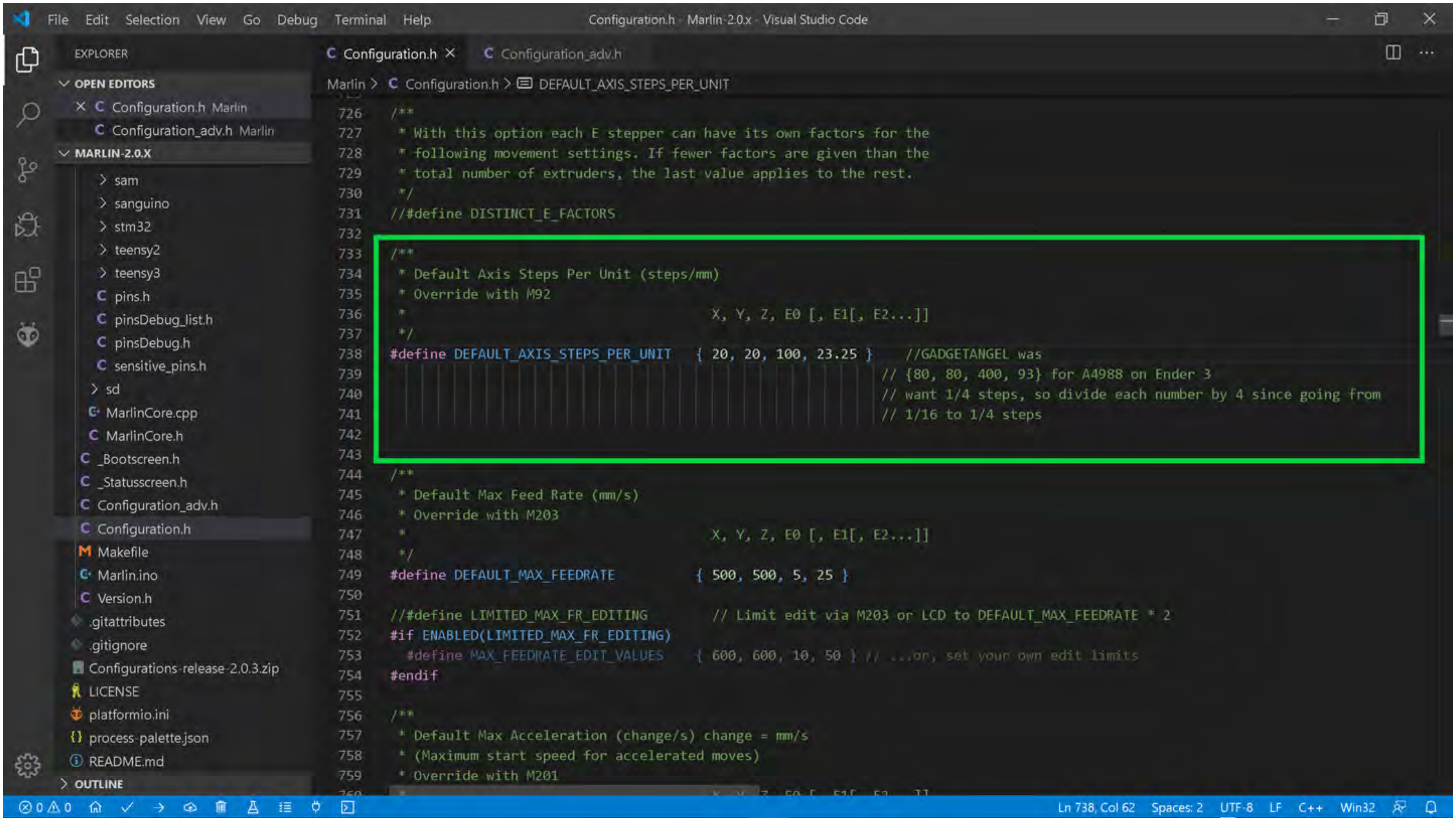
- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2100 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU TMC2100 drivers in stand-alone mode. When two "//" appear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



- Go to the next page.

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

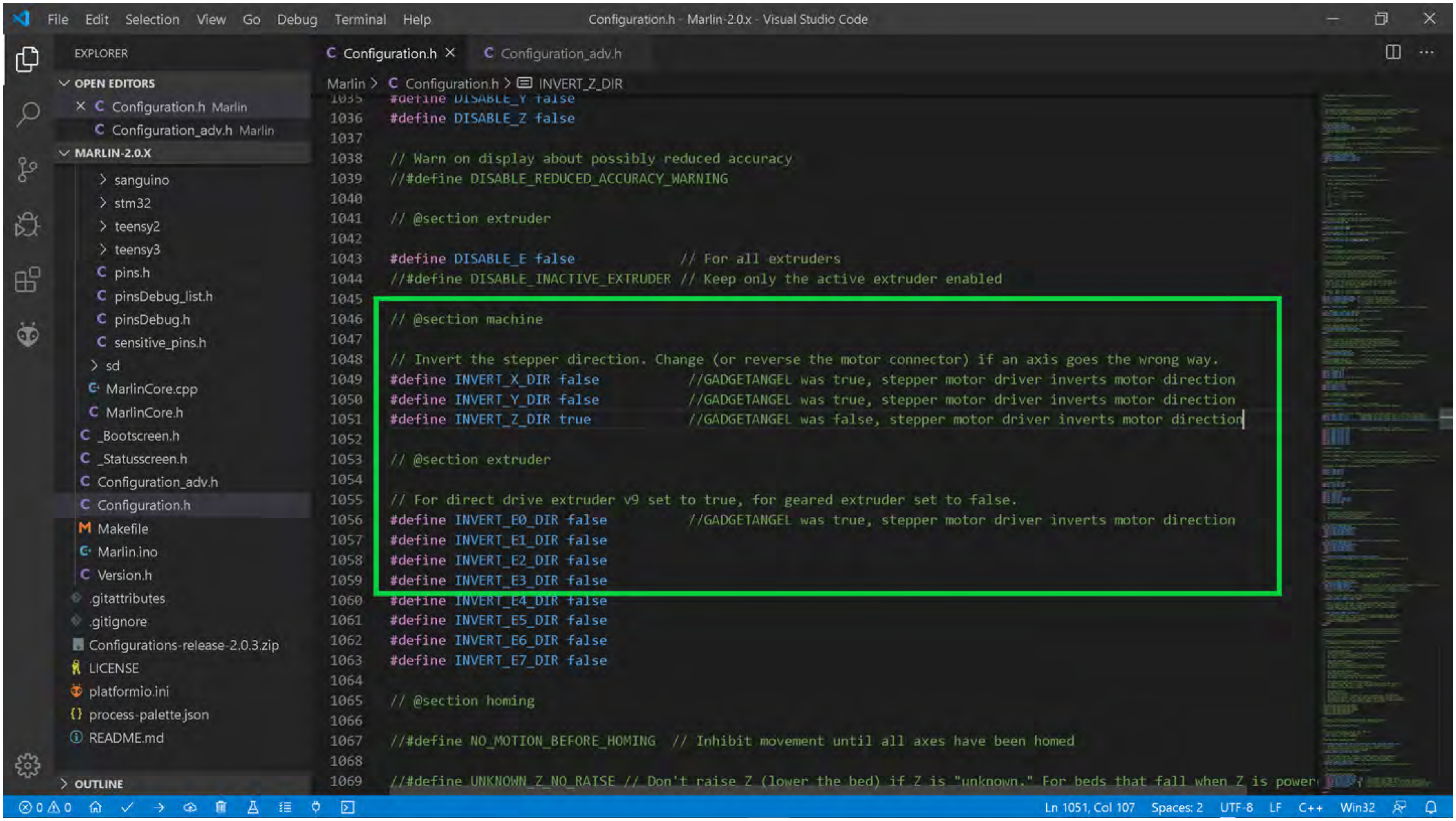
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to for BIQU TMC2100 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as shown in the **GREEN** box below.



- Go to the next page.

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

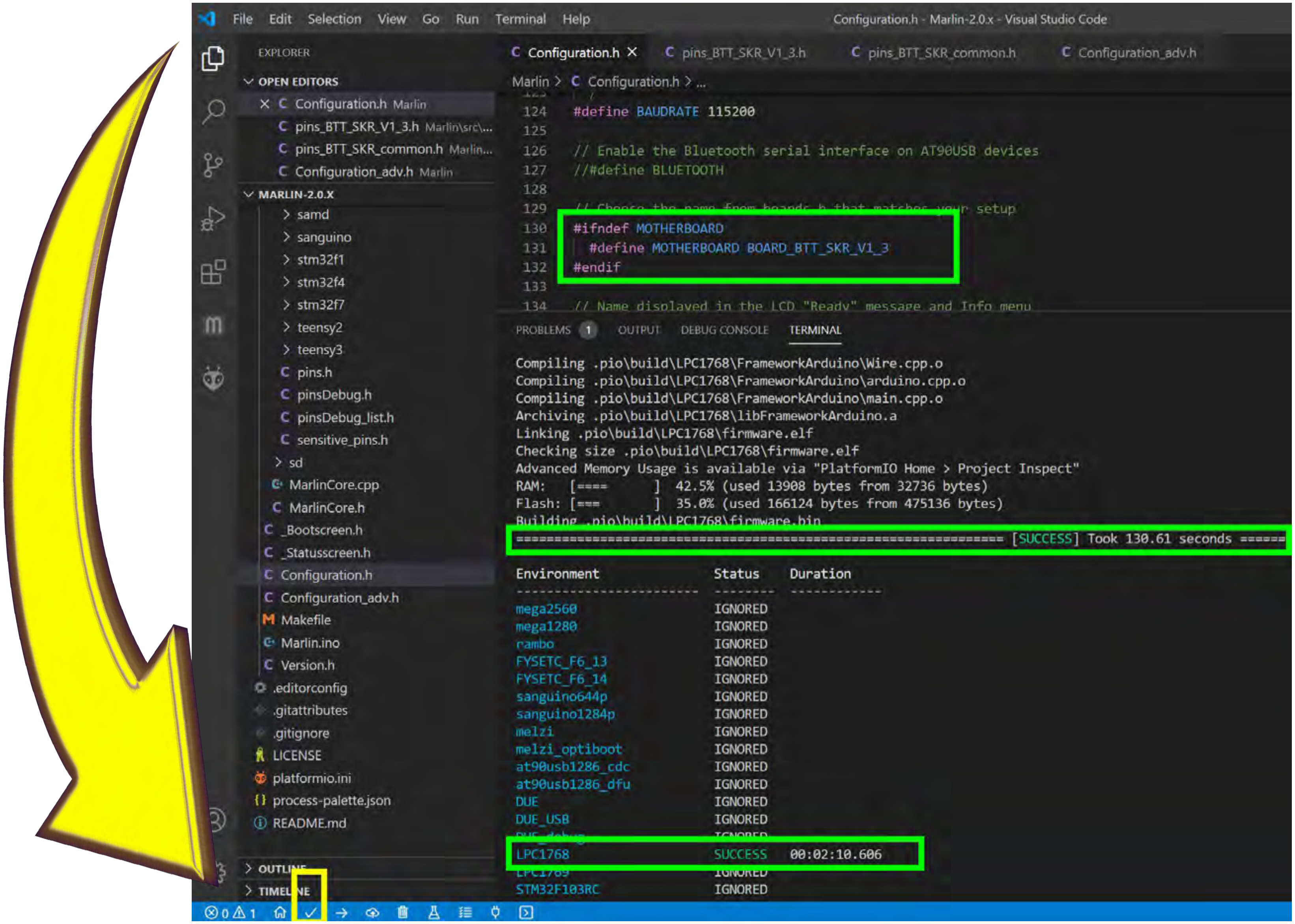
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2100 drivers, I must invert the stepper motor direction because the TMC2100 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2100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



- Go to the next page.

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

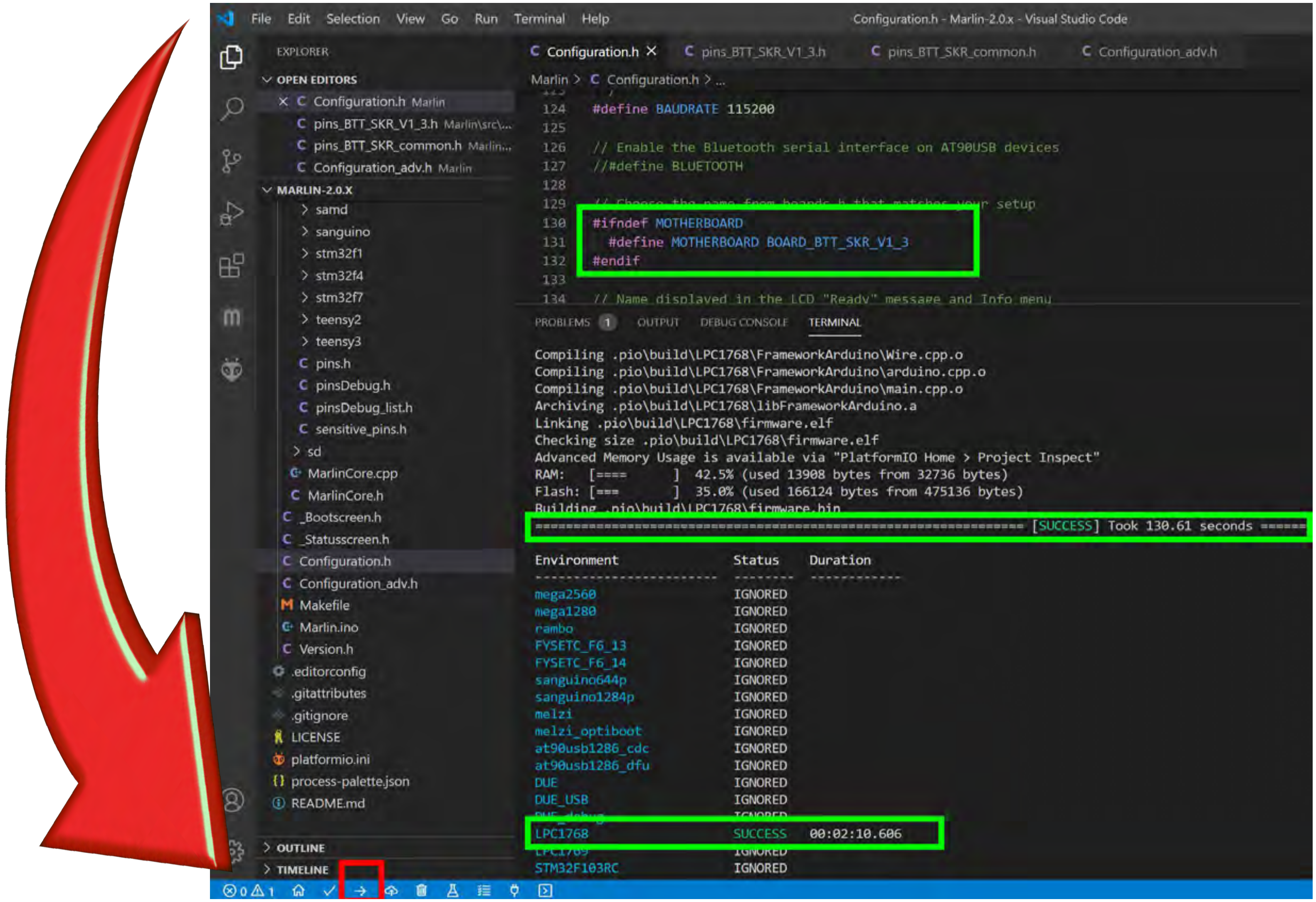
- The end of Marlin setup for BIQU TMC2100 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN boxes below.**



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

MKS TMC2100

Stand-alone Mode

base of card

Stand-alone Mode

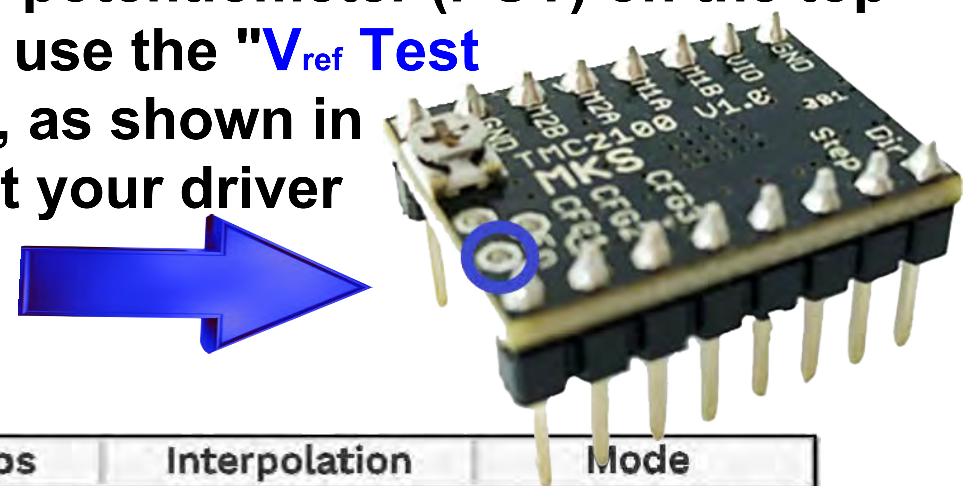
	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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



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

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

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

MKS TMC2100

Stand-alone Mode

SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1

base of card

Stand-alone Mode

	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

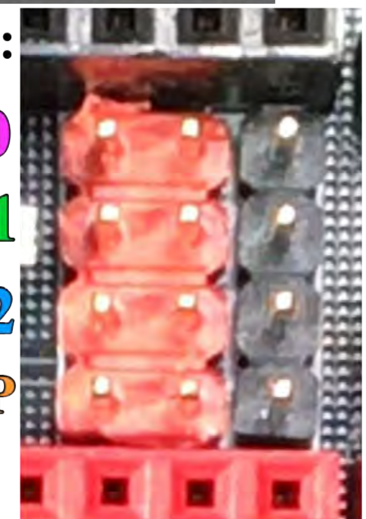
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

OPEN → No Jumper

Rows:
MS0
MS1
MS2
RST/SLP

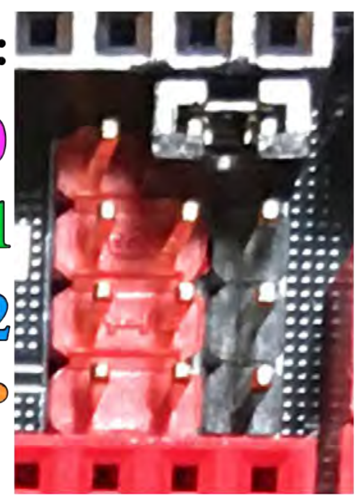


Driver Chip Chart:

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

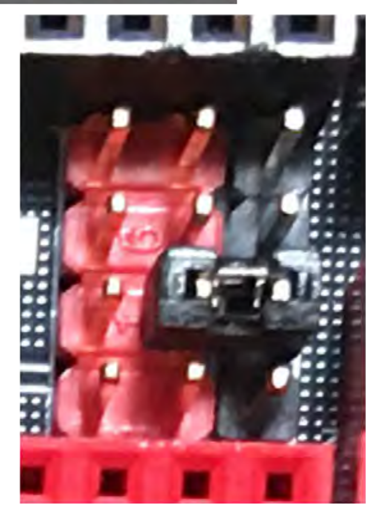
High → Jumper Set

Rows:
MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

MS1 SET HIGH



MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

MKS TMC2100

Stand-alone Mode

Driver Chip Chart:

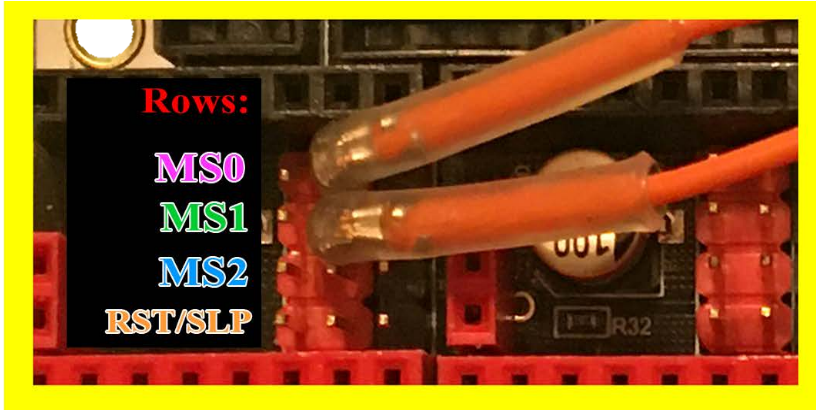
Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO	Low	Low	1	NONE	spreadCycle
TMCxxxx	High	Low	1 / 2	NONE	spreadCycle
Stand Alone Mode	OPEN	Low	1 / 2	1 / 256	spreadCycle
Maximum 16 Subdivision	Low	High	1 / 4	NONE	spreadCycle
46V DC 2.5A (peak)	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256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

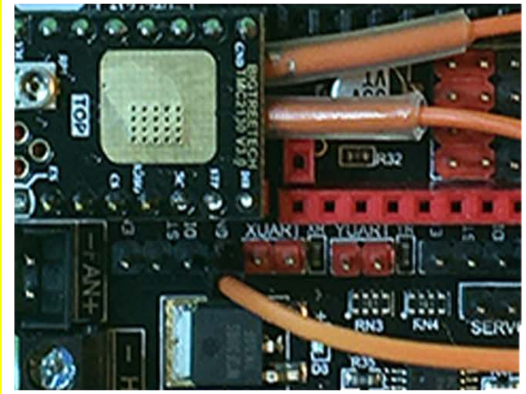
Driving Current Calculation Formula	$I_{MAX}=V_{ref}$	$V_{ref}=I_{MAX}$
R_S(Typical Sense Resistor)= 0.11Ω	See Appendix B #x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$	See Appendix B #x. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$

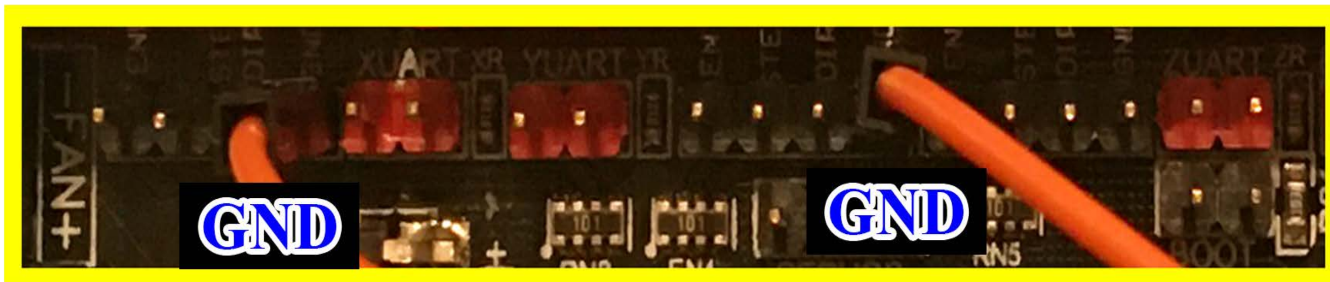
Low MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable

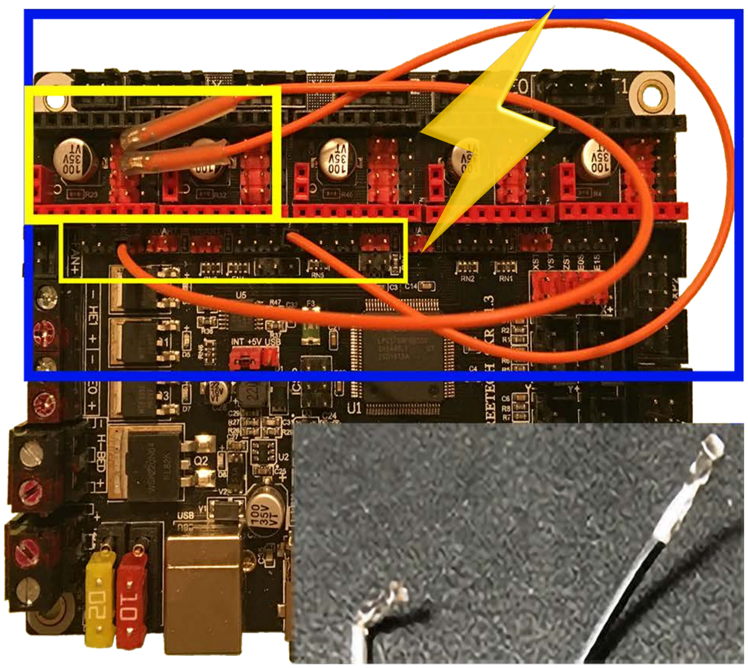
Note: See next page for instructions on how to create a DuPont jumper cable.

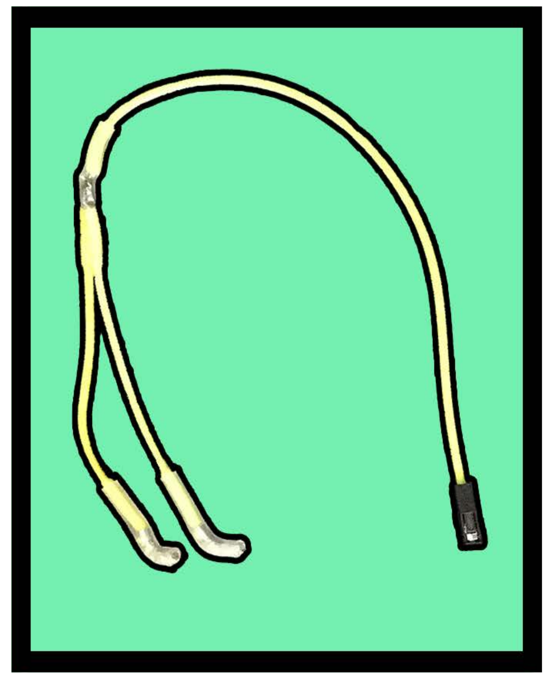
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 2







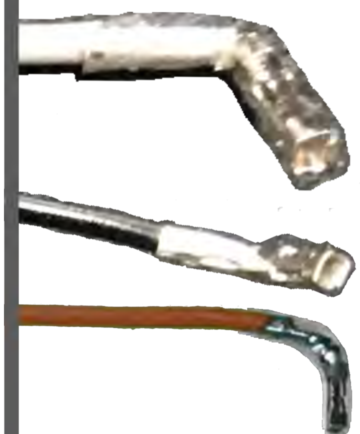




How to Create a SKR V1.3 Dupont Jumper Cable to Use with Tri State Drivers

Remove DuPont cover from one end of a female to female DuPont jumper wire.

Place a single DuPont pin into exposed female end and bend the connector backwards 90 degrees.



Since bending the connector weakens it, place a bead of solder at the 90 degree bend. I inserted a single DuPont pin inside the female connector to ensure that the connector stays clear of solder. After soldering is done, move the inserted DuPont pin around to ensure a functioning right angle female connector.

Place heat shrink over the exposed female DuPont connector to protect the exposed metal from causing shorts. Picture here is a finished jumper cable with a single DuPont pin inserted.

MKS TMC2100

Stand-alone Mode

base of card

Stand-alone Mode

	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

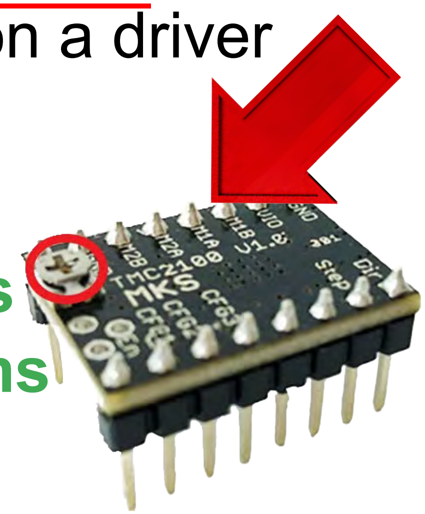
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

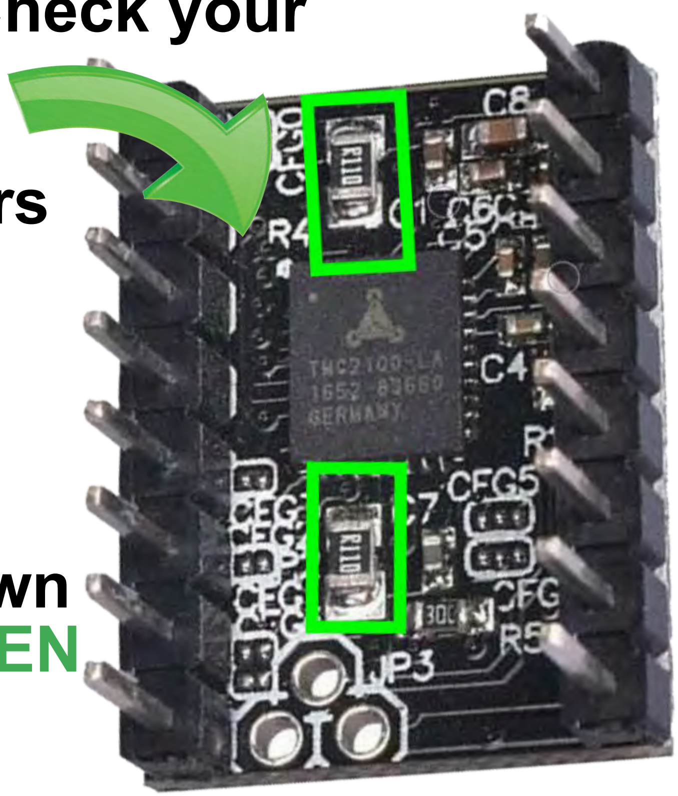
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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



To learn more, please watch this YouTube video done by Teaching Tech and check out this link on the TMC2100 Driver

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



R_s = R050 is 0.05 Ohms

R_s = R068 is 0.068 Ohms

R_s = R100 is 0.1 Ohms

R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

R_s = R220 is 0.22 Ohms

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

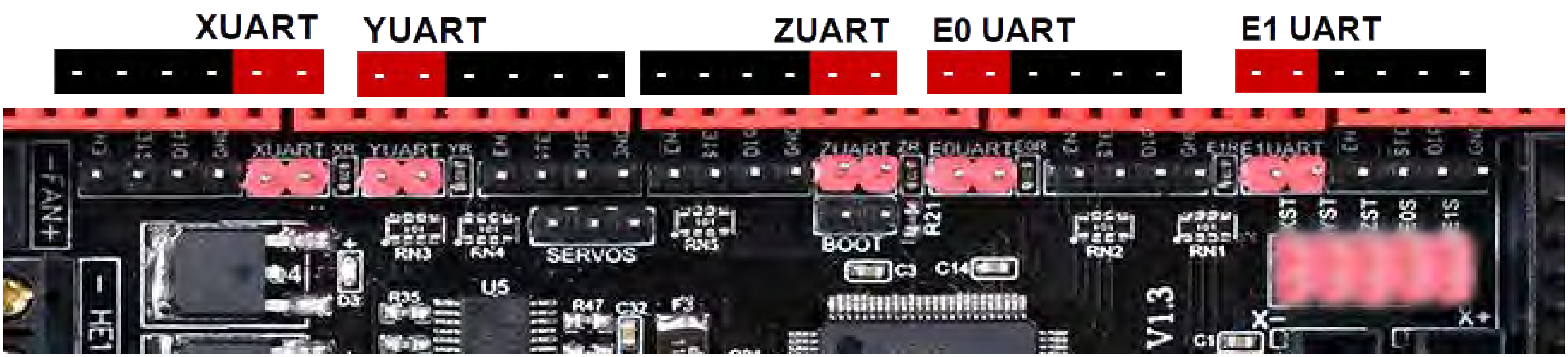
Stand-alone Mode

MKS TMC2100

Stand-alone Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.



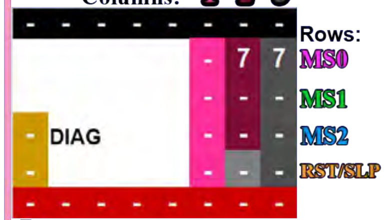
MKS TMC2100

Stand-alone Mode

SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

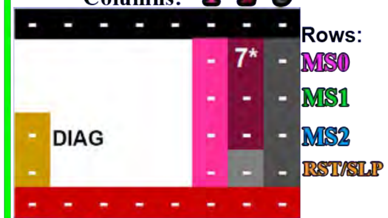
Driver Socket Representation

Columns: 1 2 3



1/4 with 1/256 in StealthChop Mode

Columns: 1 2 3



1/16 with 1/256 in SpreadCycle Mode

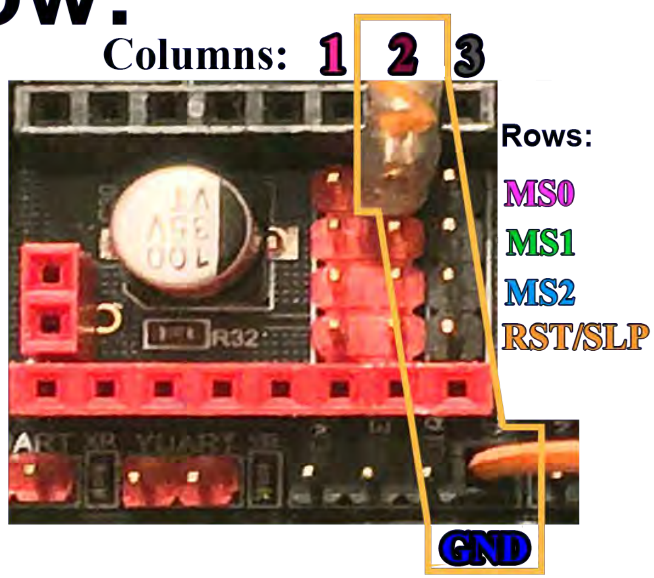
Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B x. Use 50% to 90% as shown below: $I_{max}=I_{max}$ * 0.90			$V_{ref}=I_{MAX}$ See Appendix B x. Use 50% to 90% as shown below: $V_{ref}=V_{ref}$ * 0.90	

- - → No Jumper set

Low:

Columns: 1 2 3



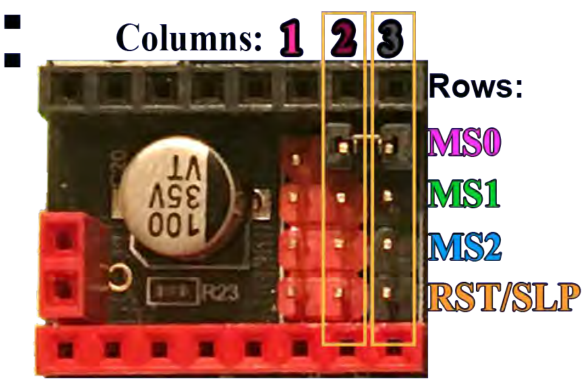
MS0 for Tri State Drivers ONLY (TMC2100 & TMC2130):

Driver Socket Representation:	Driver Chip Chart:
7 7 →	High
7* - →	Low
- - →	Open

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)
Open →	No Jumper set

High:

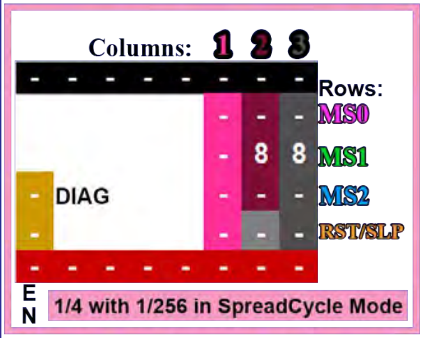
Columns: 1 2 3

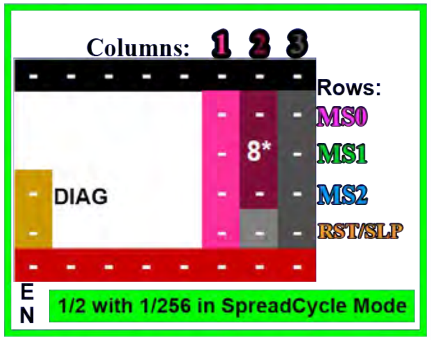


MKS TMC2100

Stand-alone Mode

Driver Socket Representation





Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B #x. Use 50% to 90% as shown below: $I_{max}=I_{max}$ * 0.90		$V_{ref}=I_{MAX}$ See Appendix B #x. Use 50% to 90% as shown below: $V_{ref}=V_{ref}$ * 0.90		

- - → No Jumper set

MS1 for Tri State Drivers ONLY (TMC2100 & TMC2130):

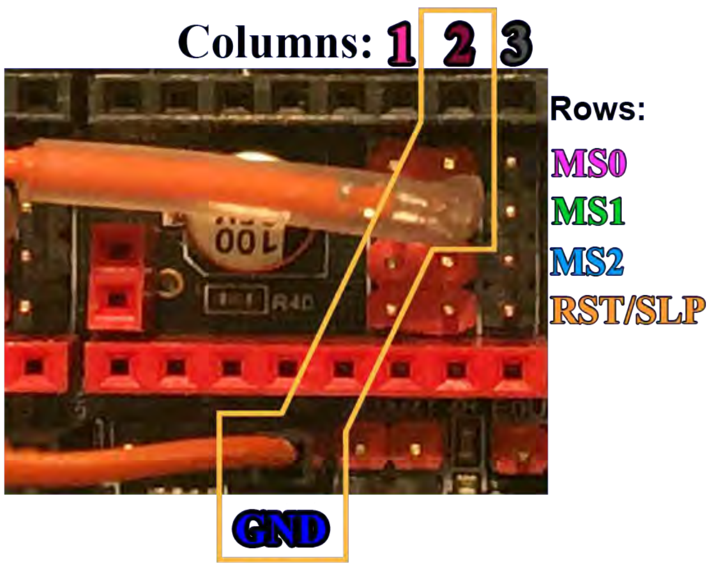
Driver Socket Represntation: 8 8 → Driver Chip Chart: High

Driver Socket Represntation: 8* - → Driver Chip Chart: Low

Driver Socket Represntation: - - → Driver Chip Chart: Open

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS1 row
Low →	Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)
Open →	No Jumper set

Low:



High:

Stand-alone Mode

MKS TMC2100

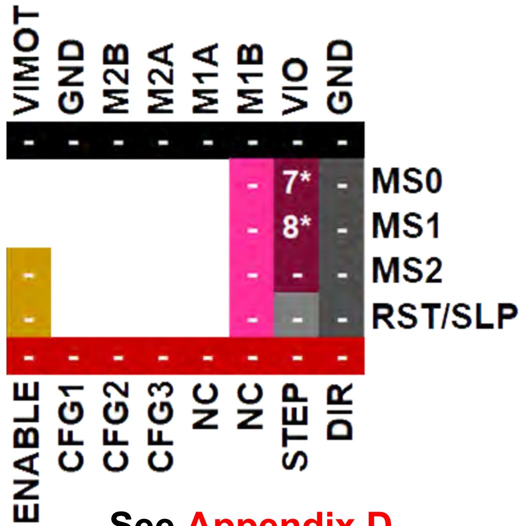
Stand-alone Mode

Stand-alone Mode

STEP

Interpolation: none

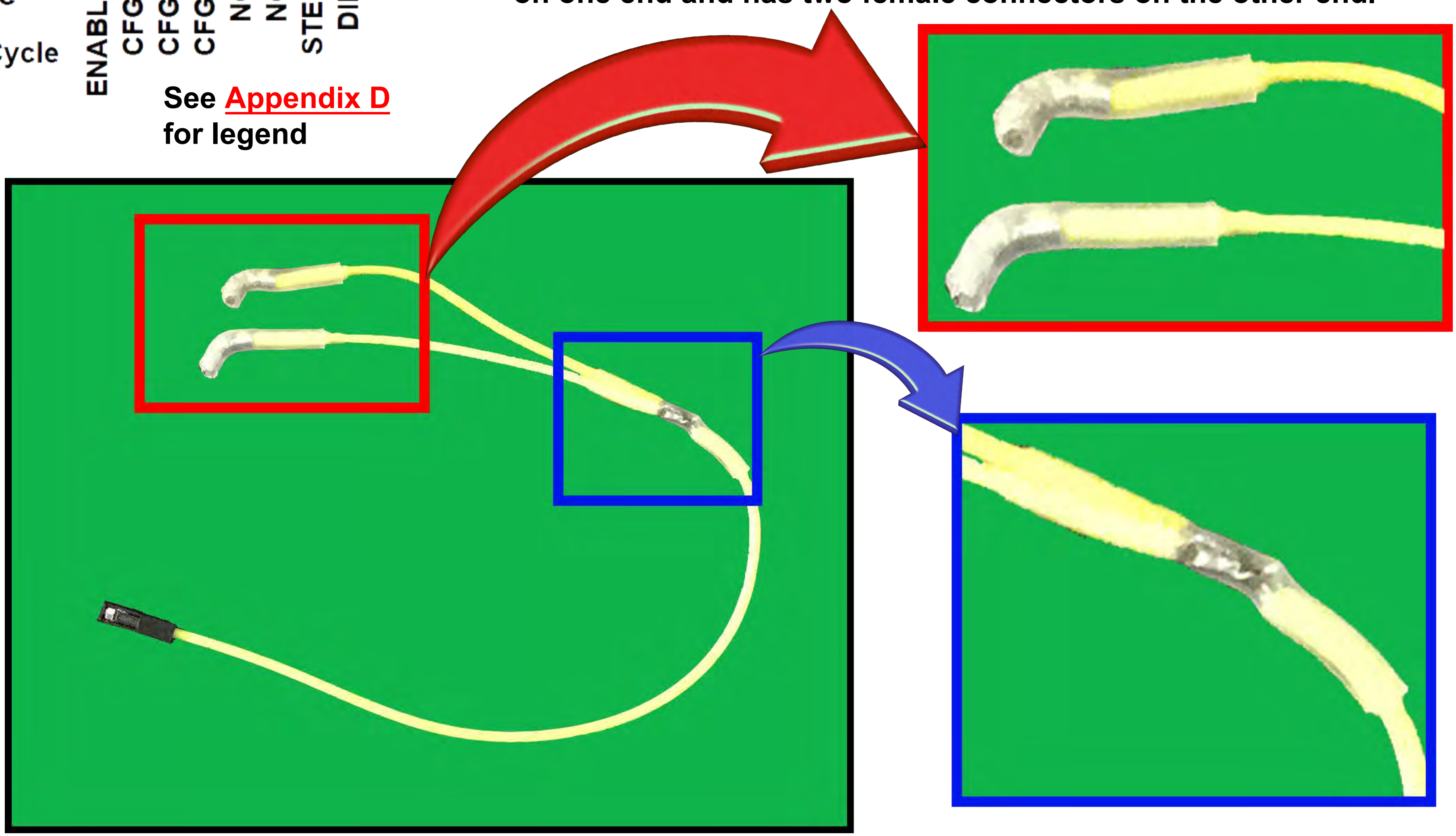
SpreadCycle



See Appendix D for legend

Additional Equipment Needed for Low Low (STEP or FULL) Configuration

You will need one DuPont jumper cable that connects to a ground point (GND) on the SKR V1.3 board but will source two PINS simultaneously. So create a Y-cable that has one female connector on one end and has two female connectors on the other end.



Stand-alone Mode

MKS TMC2100

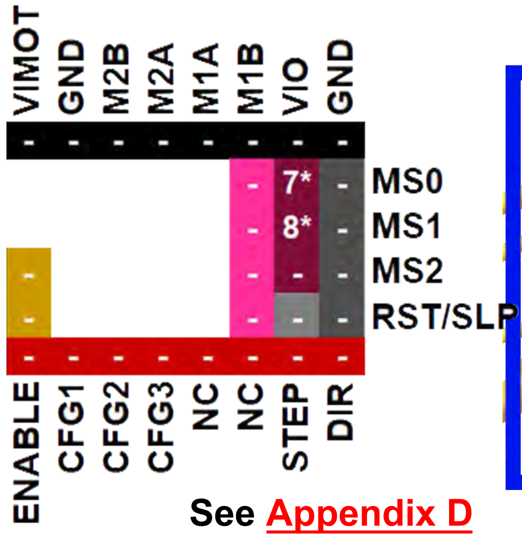
Stand-alone Mode

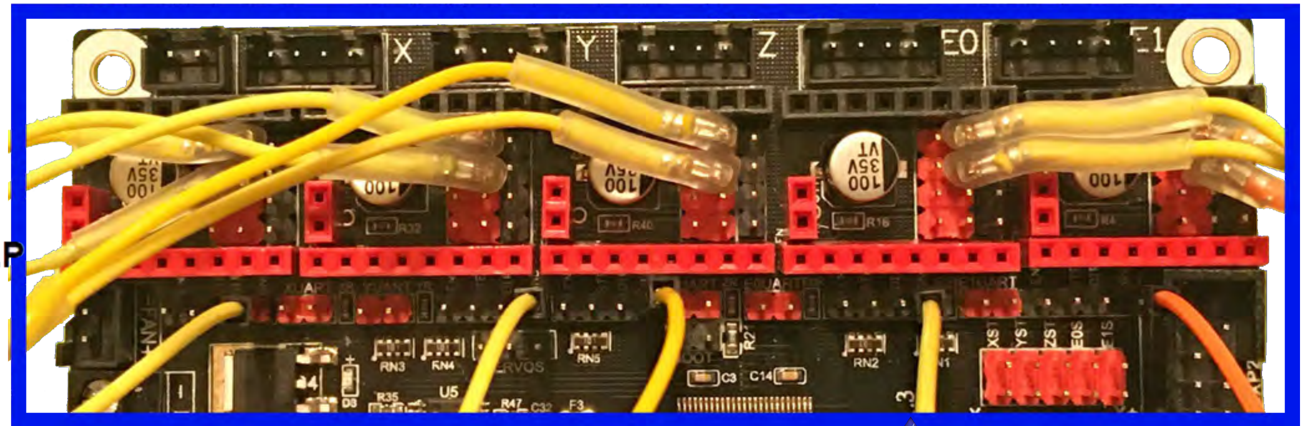
Stand-alone Mode

STEP

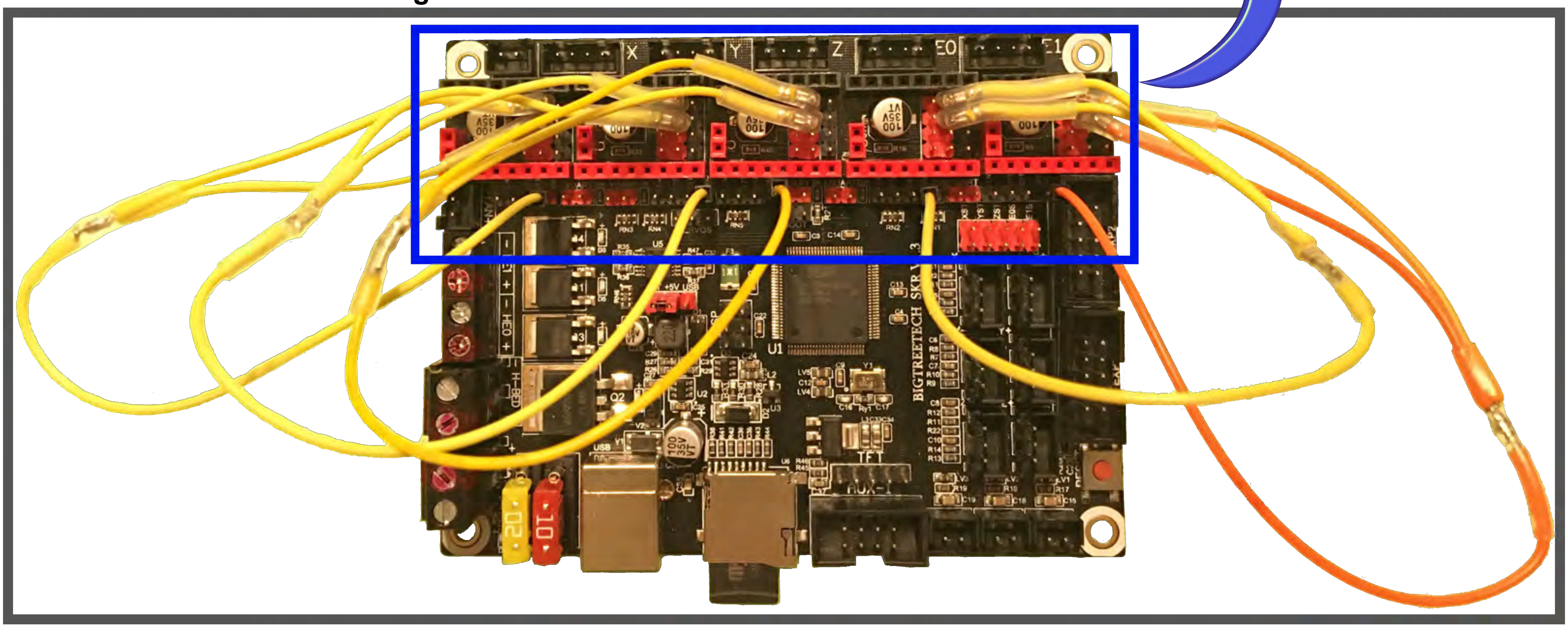
Interpolation: none

SpreadCycle

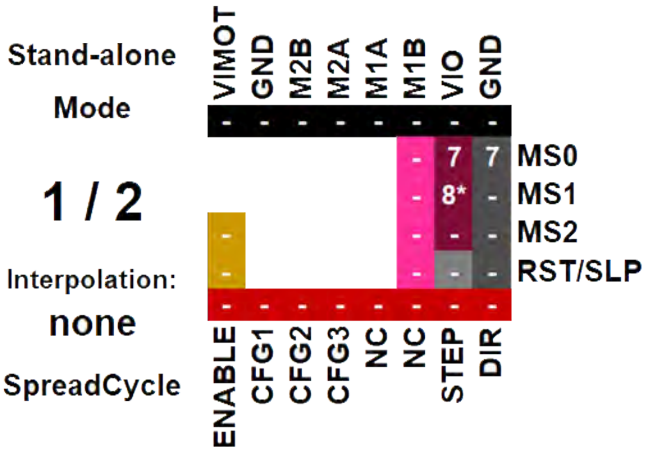




See Appendix D for legend

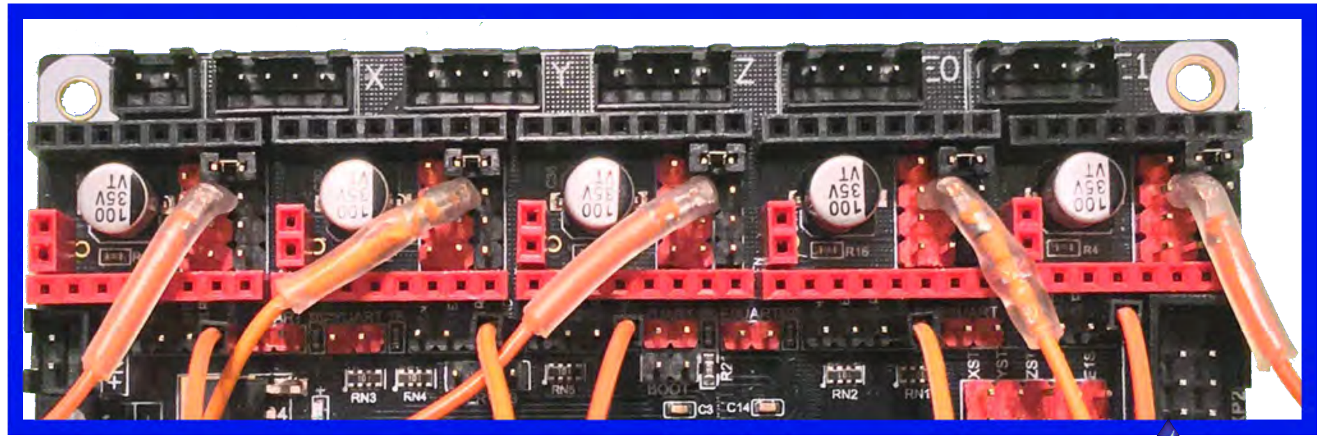


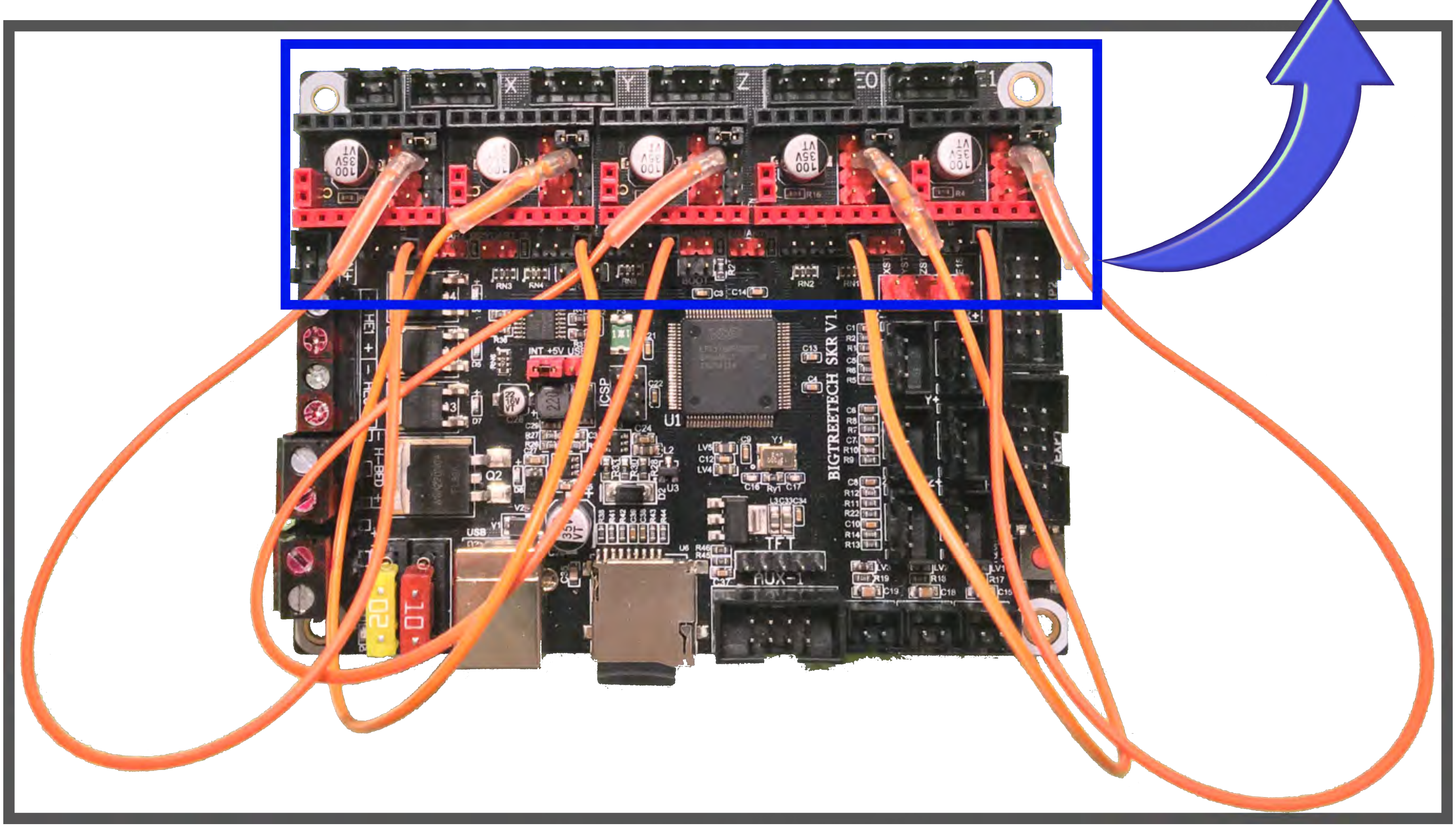
Stand-alone Mode



MKS TMC2100

Stand-alone Mode





See Appendix D for legend

Stand-alone Mode

MKS TMC2100

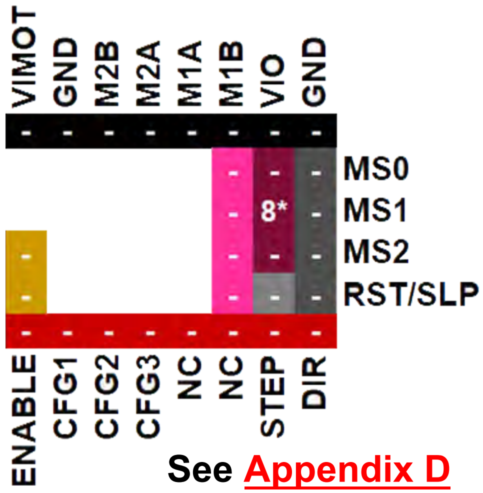
Stand-alone Mode

Stand-alone Mode

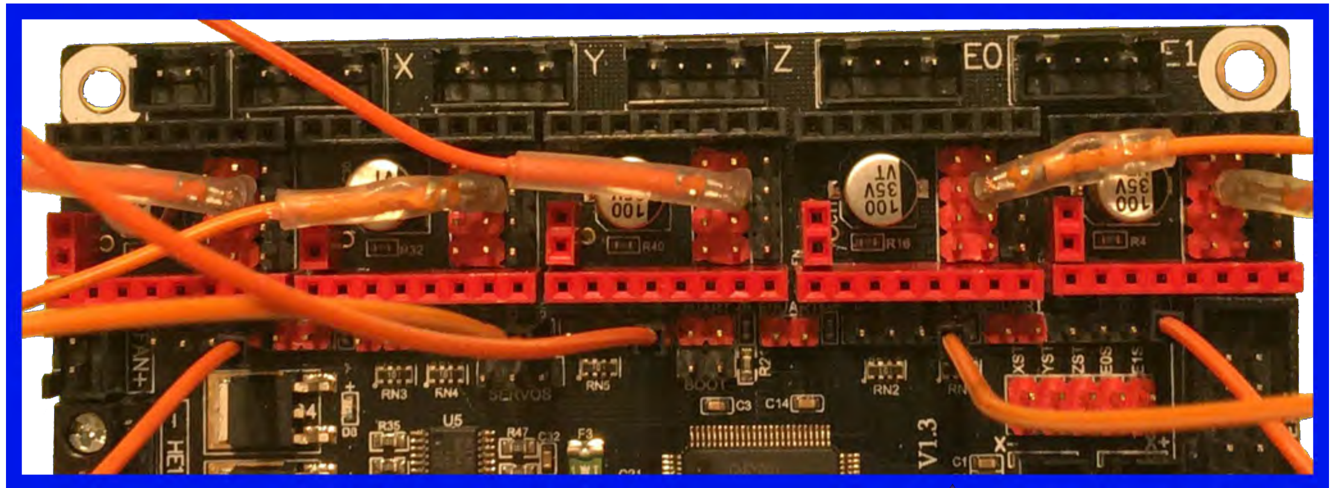
1 / 2

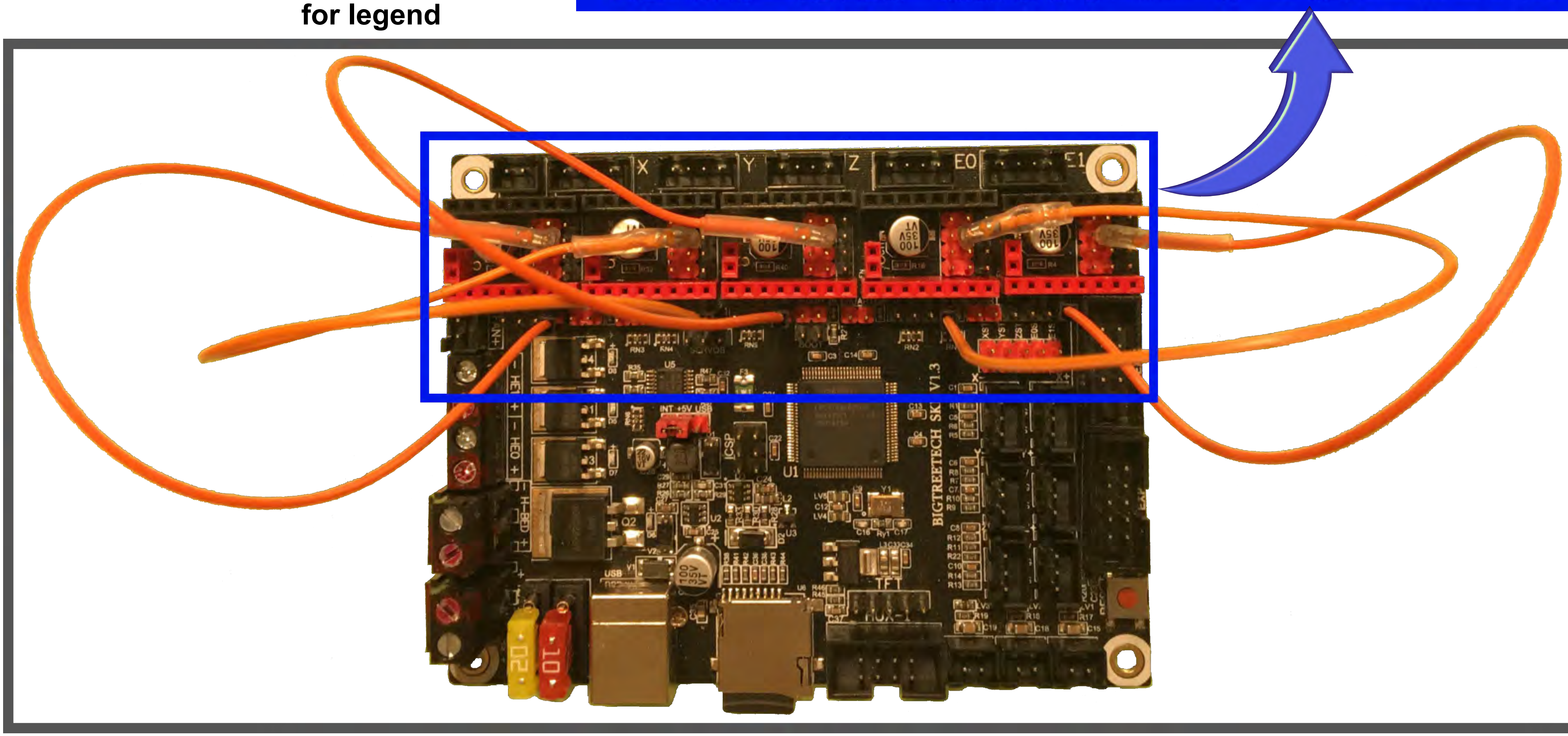
Interpolation:
1 / 256

SpreadCycle



See Appendix D for legend





Stand-alone Mode

MKS TMC2100

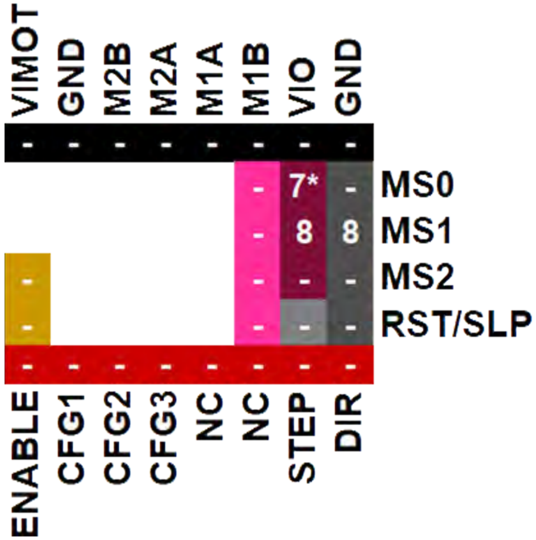
Stand-alone Mode

Stand-alone Mode

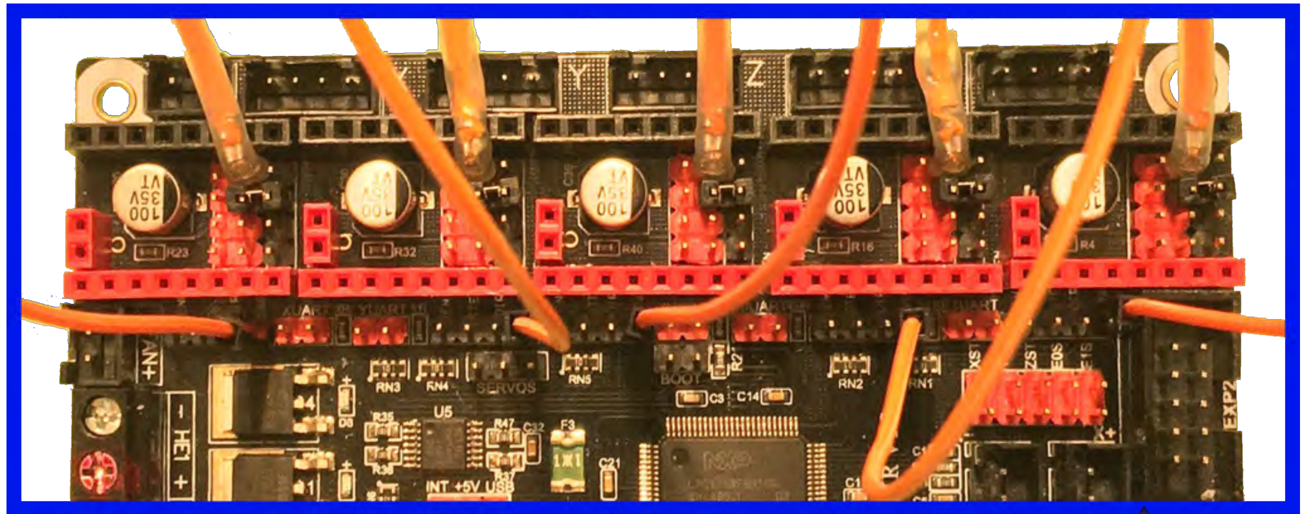
1 / 4

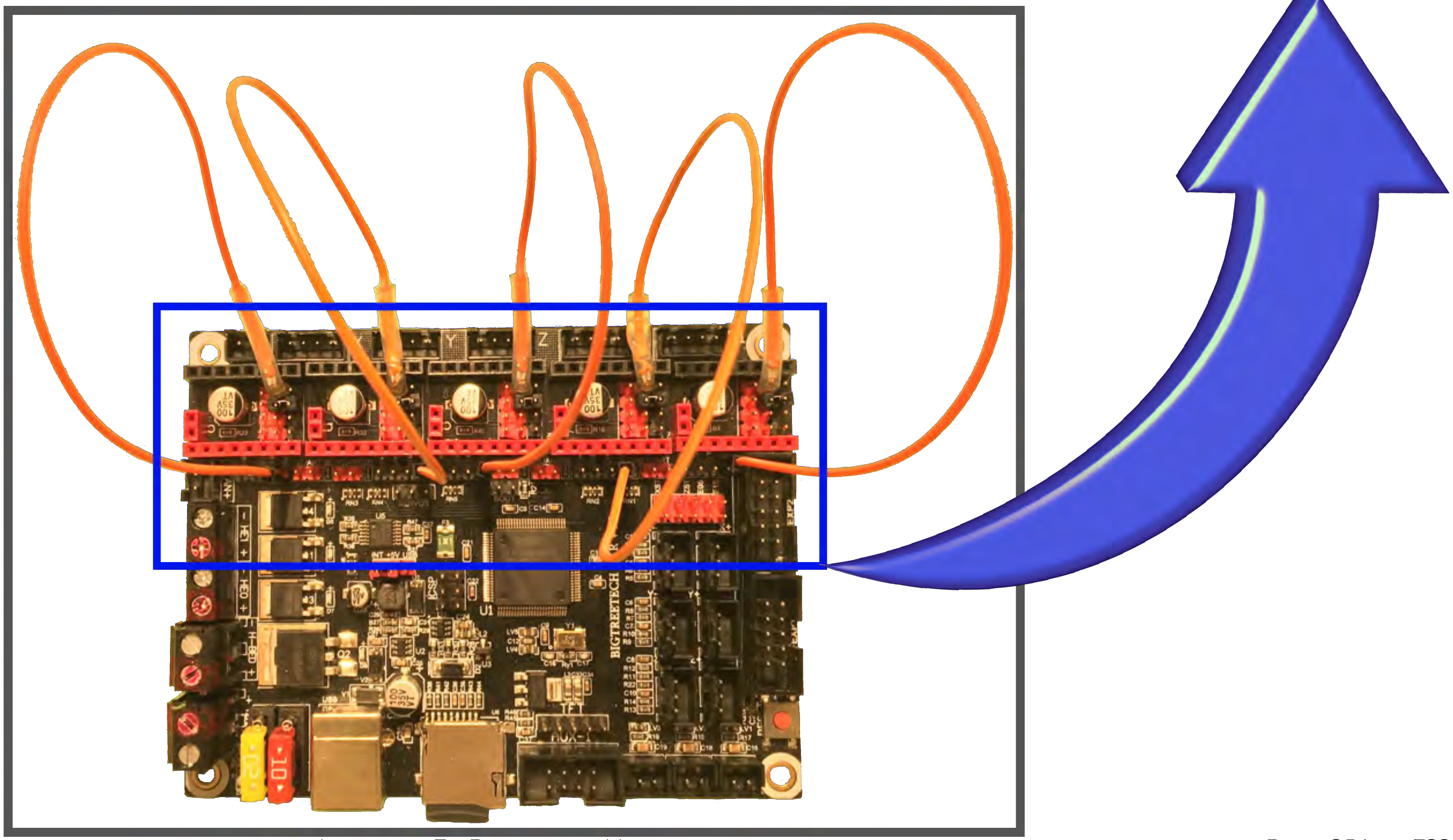
Interpolation: none

SpreadCycle



See Appendix D for legend





Stand-alone Mode

MKS TMC2100

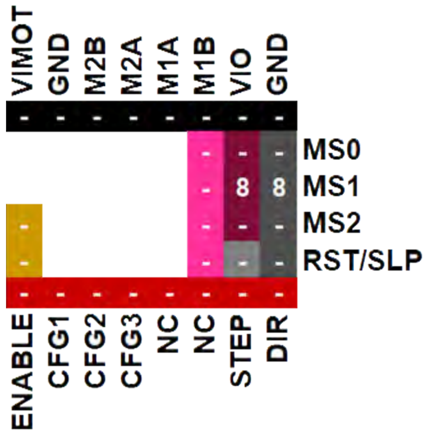
Stand-alone Mode

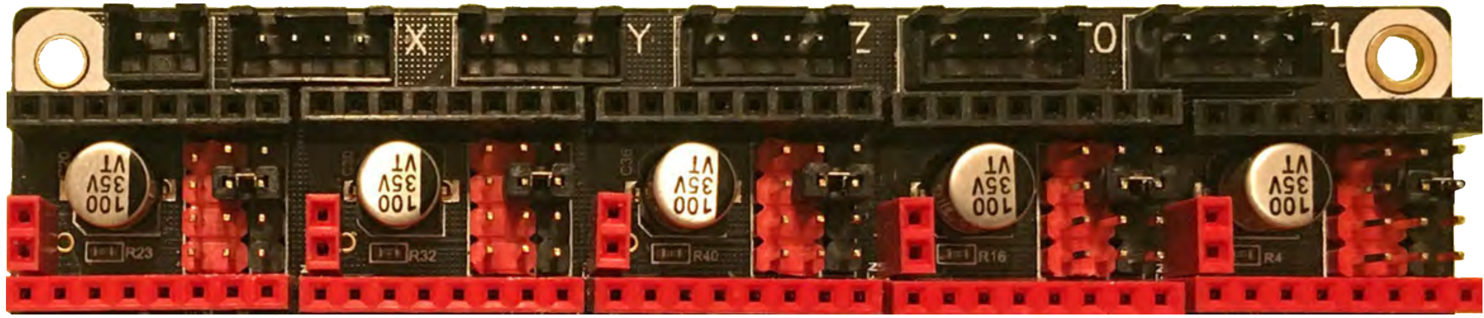
Stand-alone Mode

1 / 4

Interpolation:
1 / 256

SpreadCycle





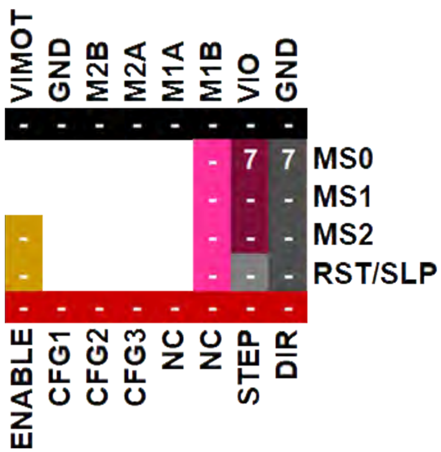
See Appendix D for legend

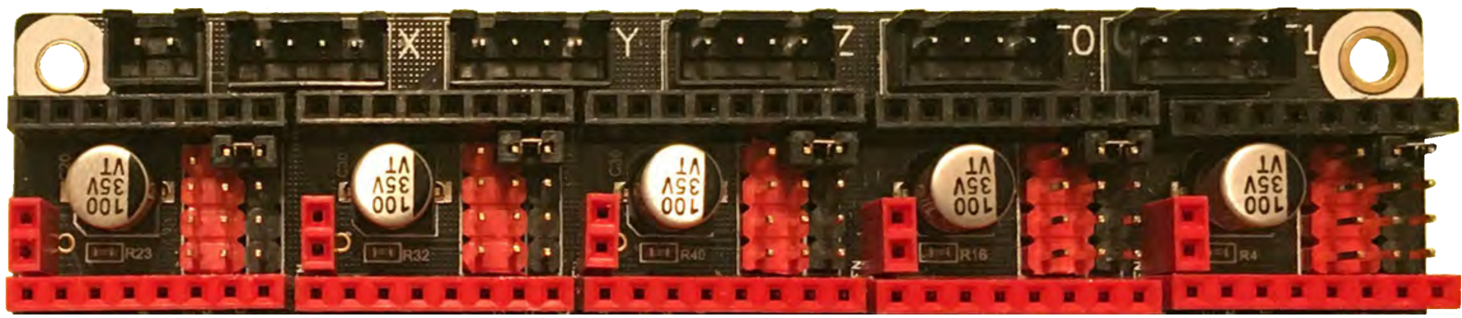
Stand-alone Mode

1 / 4

Interpolation:
1 / 256

StealthChop





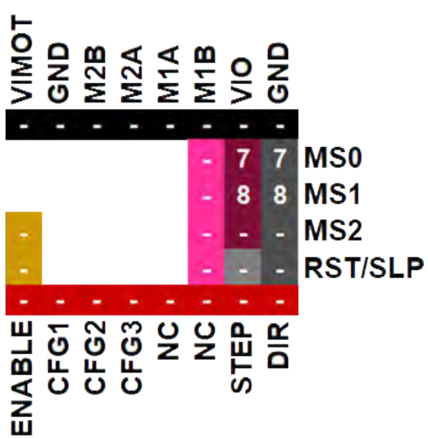
See Appendix D for legend

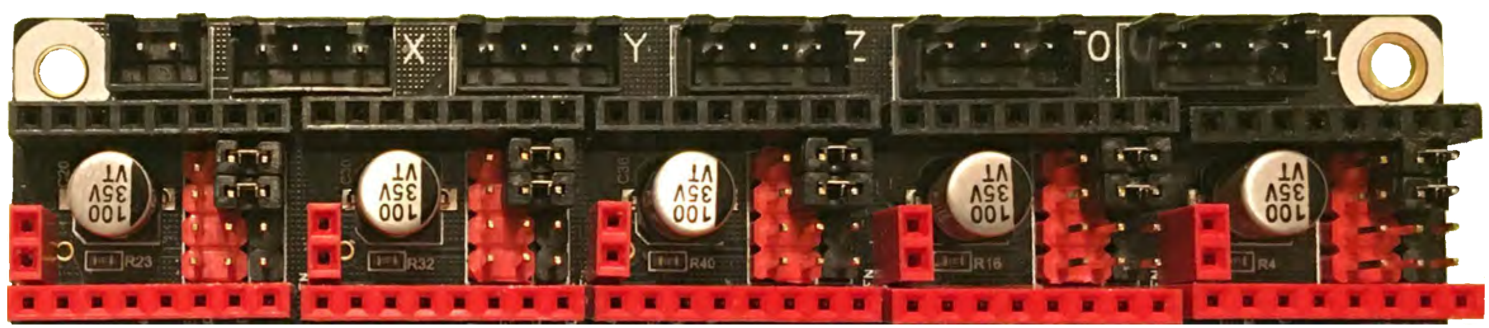
Stand-alone Mode

1 / 16

Interpolation:
none

SpreadCycle





See Appendix D for legend

Stand-alone Mode

MKS TMC2100

Stand-alone Mode

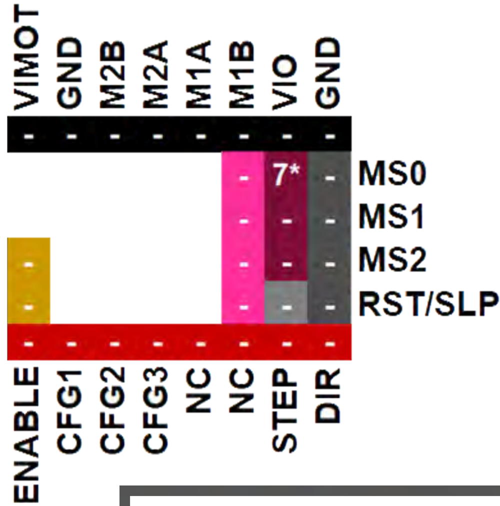
Stand-alone Mode

1 / 16

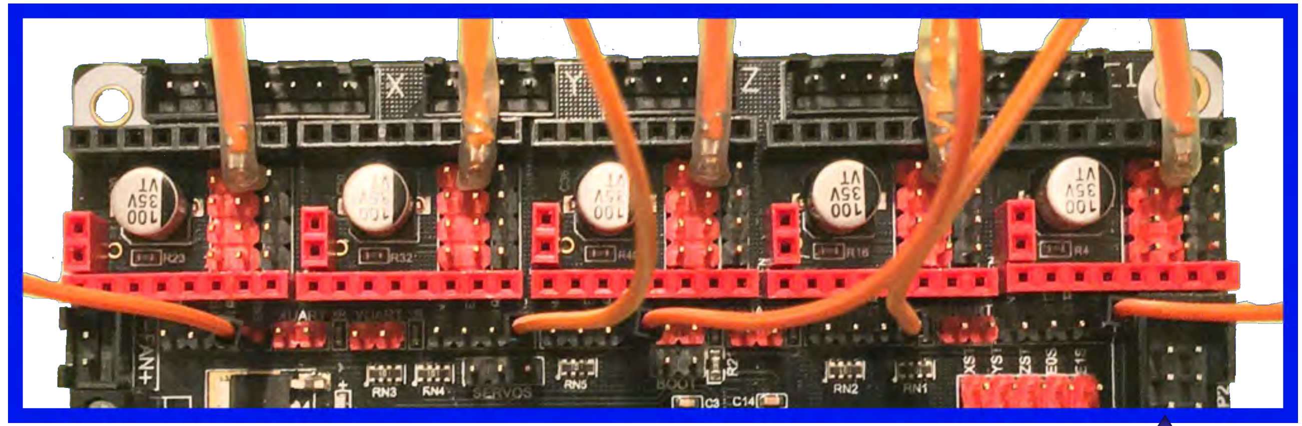
Interpolation:

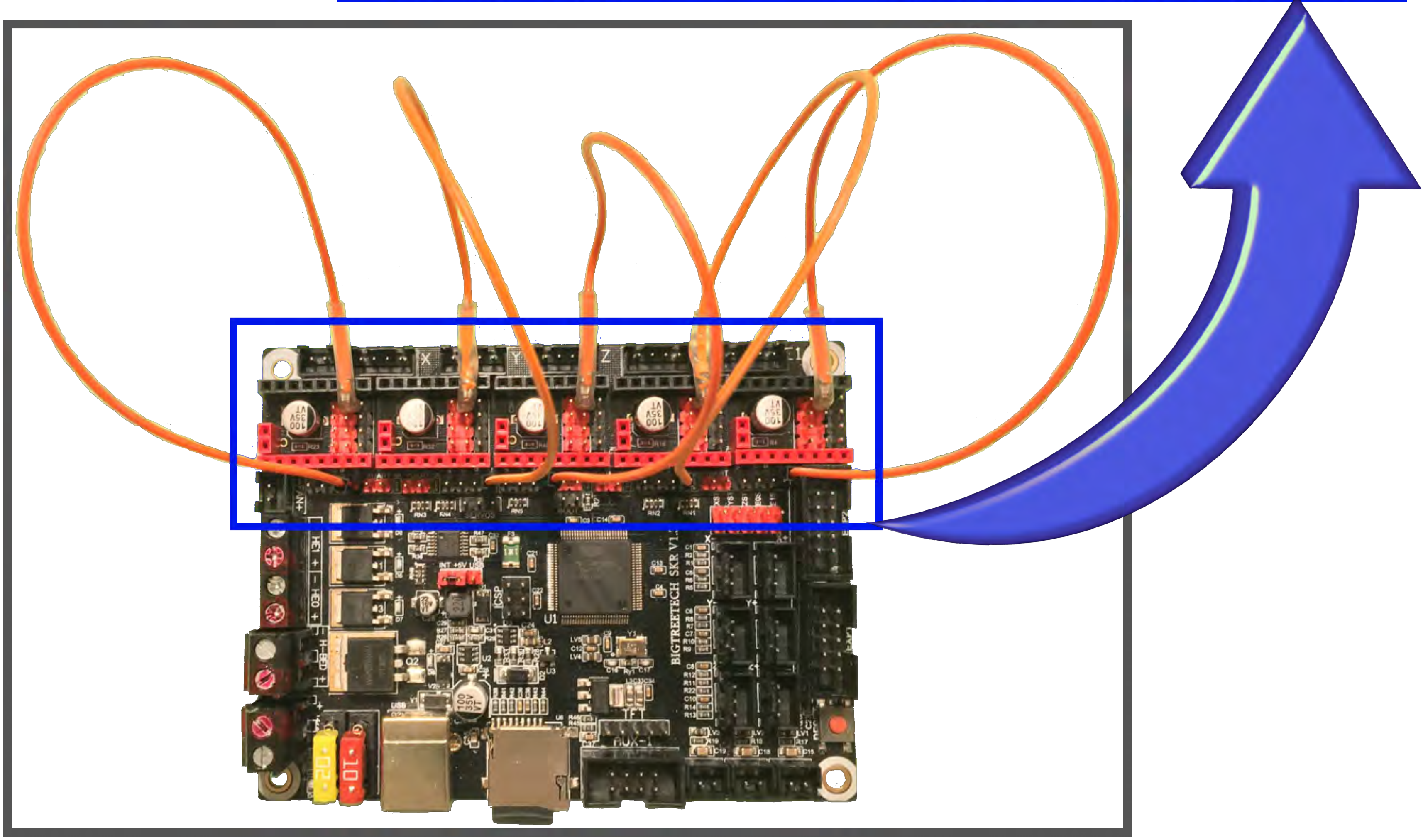
1 / 256

SpreadCycle



See Appendix D for legend





Stand-alone Mode

MKS TMC2100

Stand-alone Mode

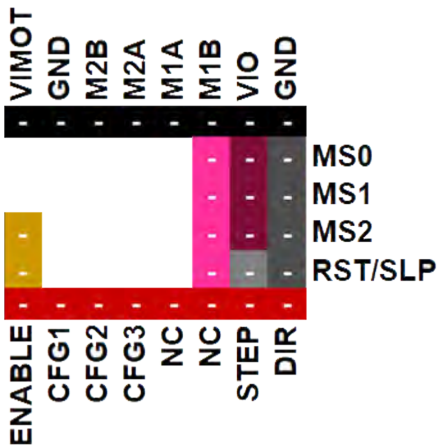
Stand-alone Mode

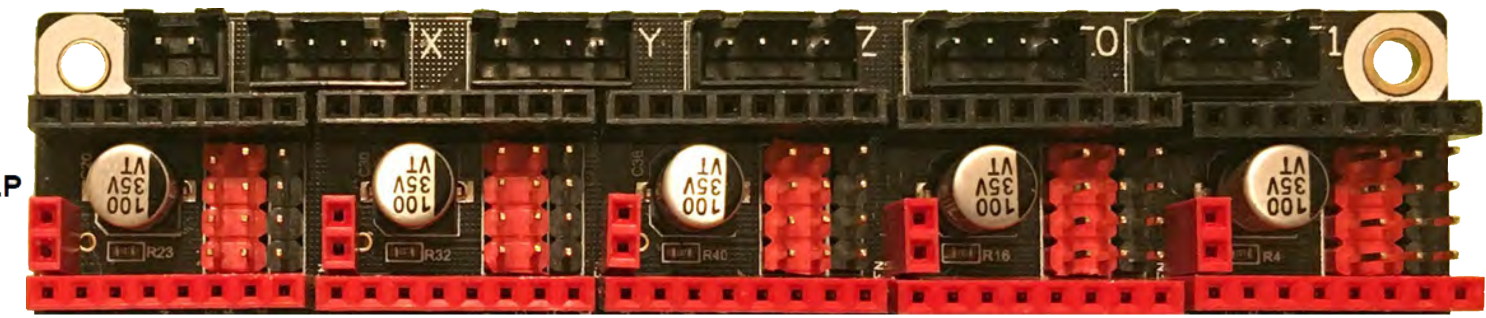
1 / 16

Interpolation:

1 / 256

StealthChop



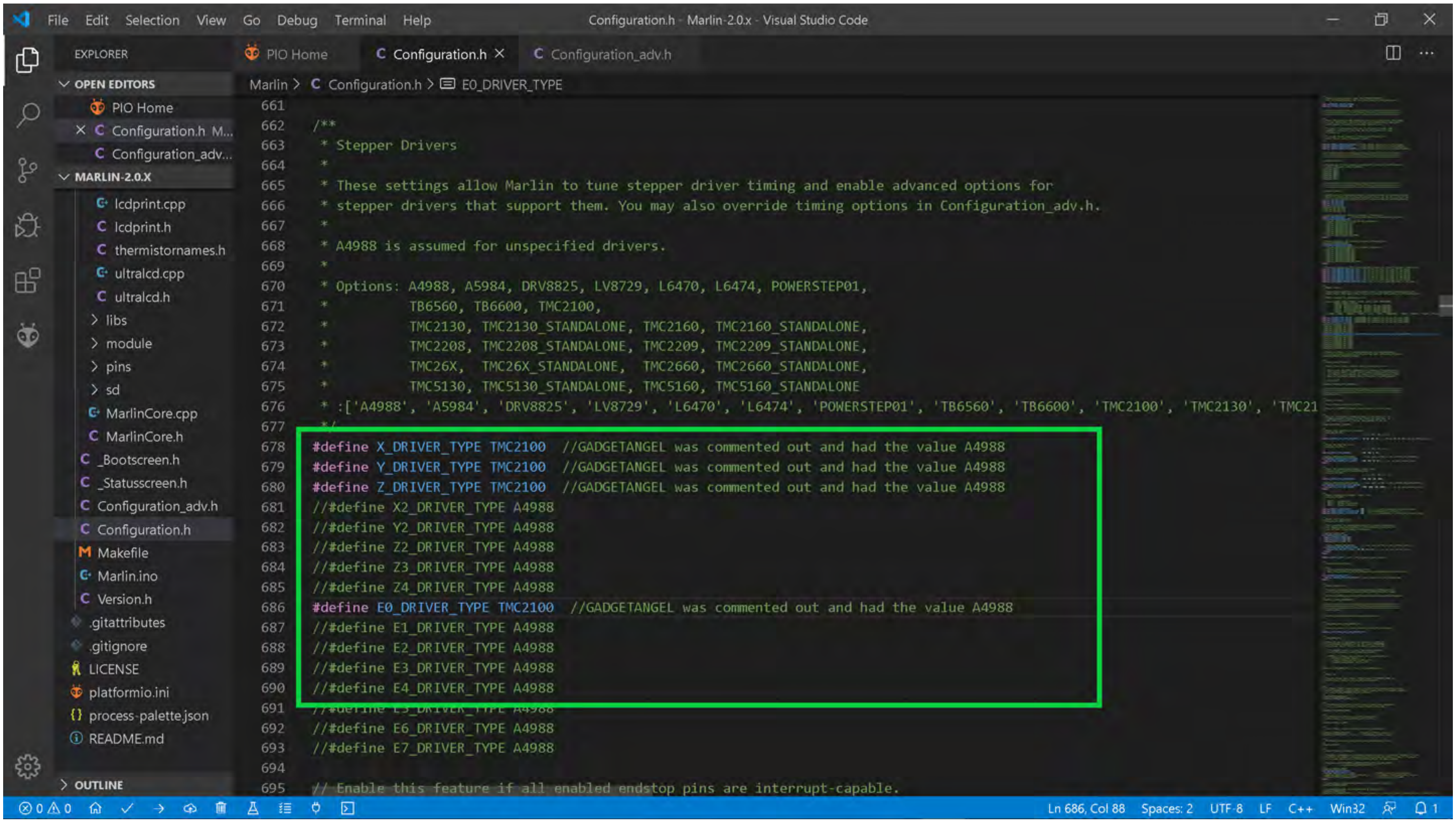


See Appendix D for legend

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

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

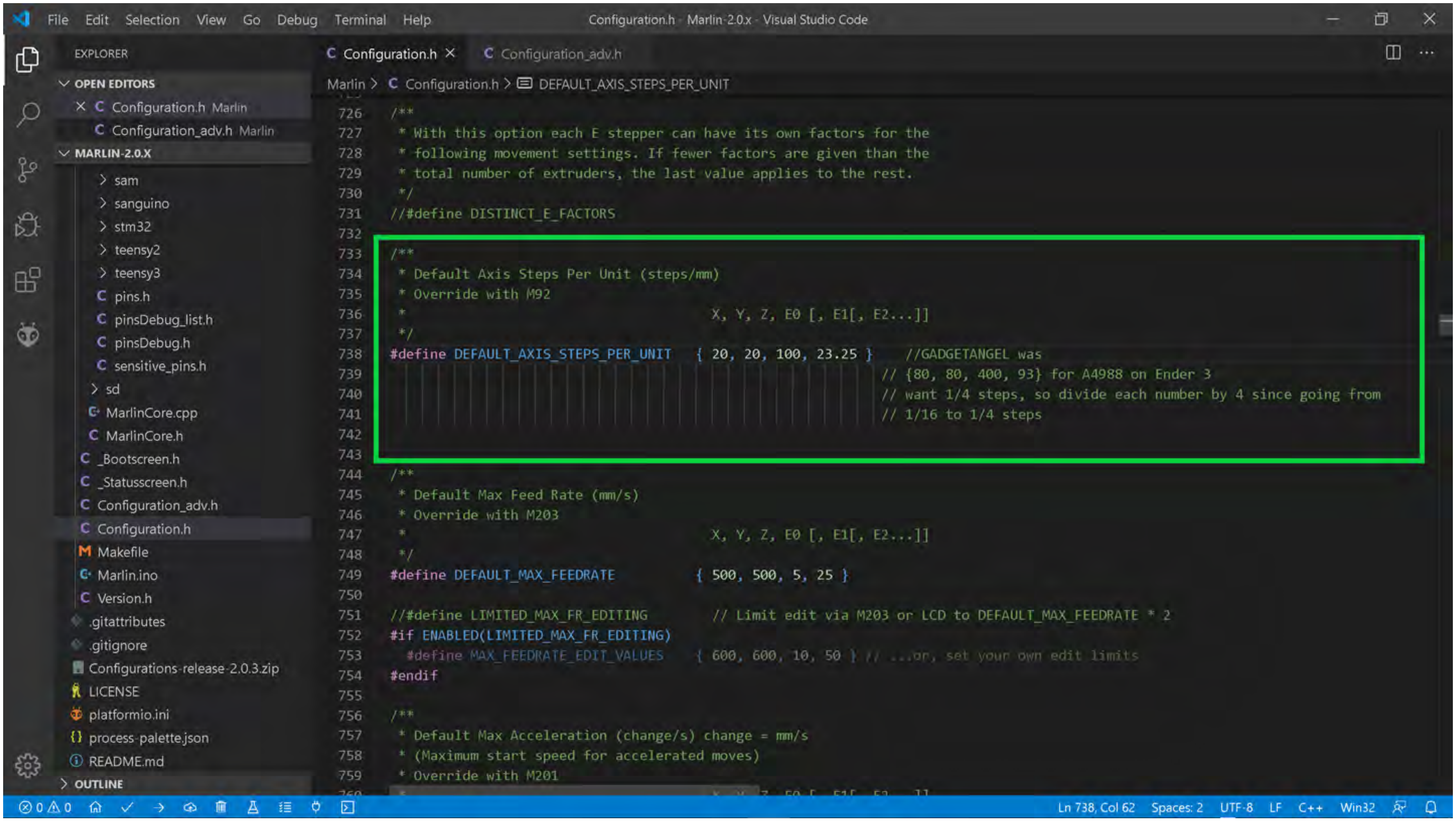
- Change the stepper motor drivers so that Marlin knows you are using TMC2100 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2100 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

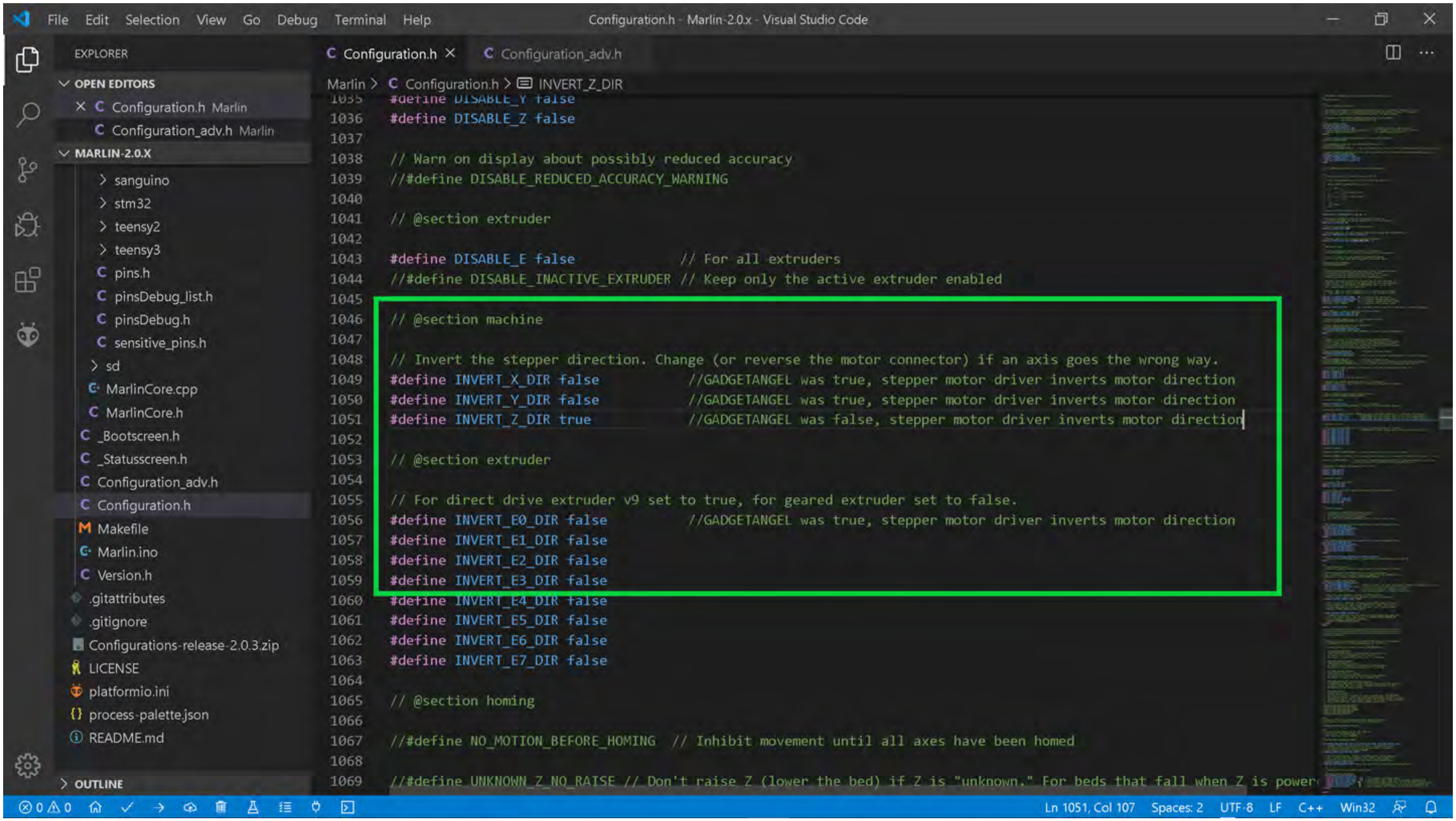
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to MKS TMC2100 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

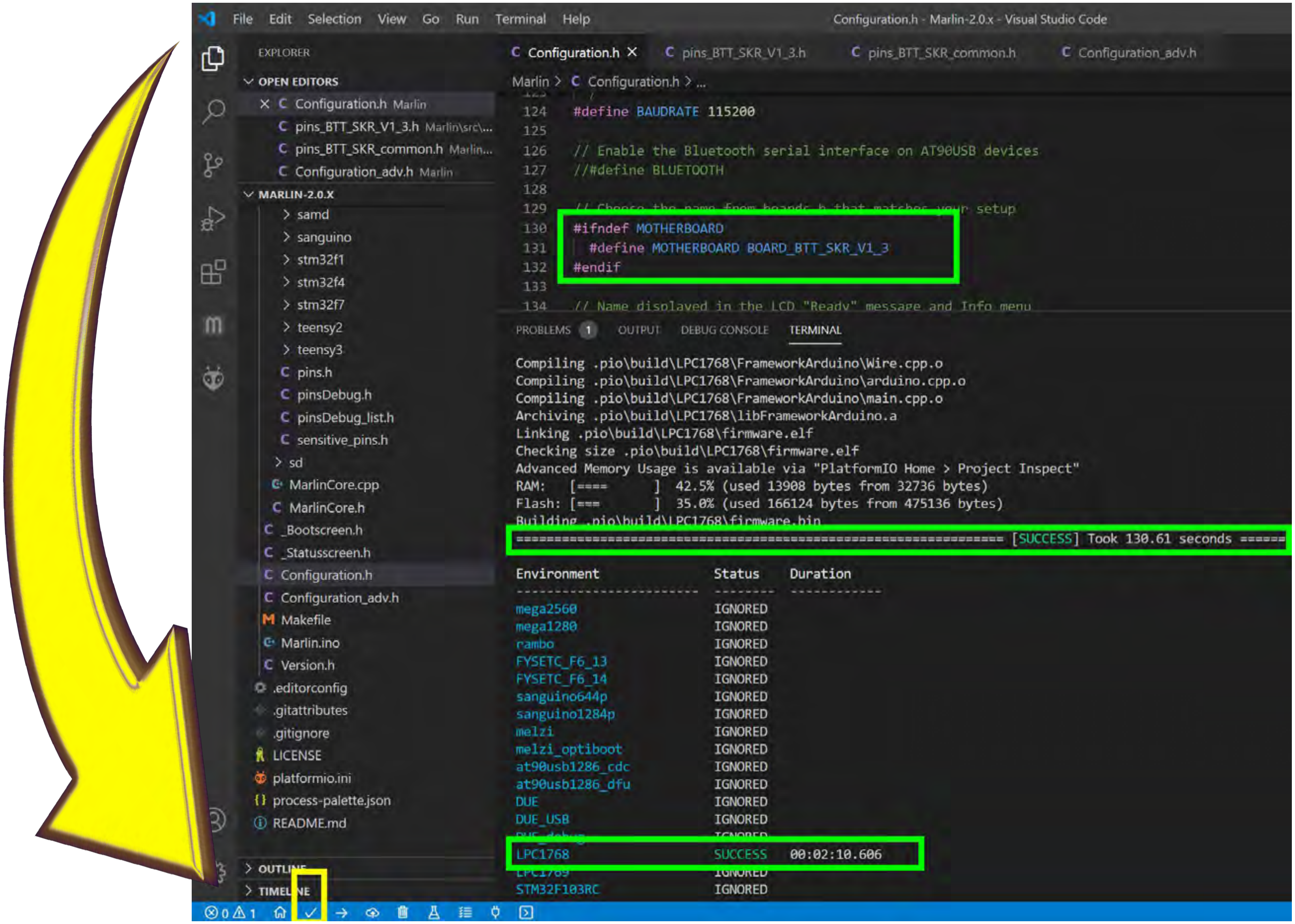
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2100 drivers, I must invert the stepper motor direction because the TMC2100 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2100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



- Go to the next page.

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

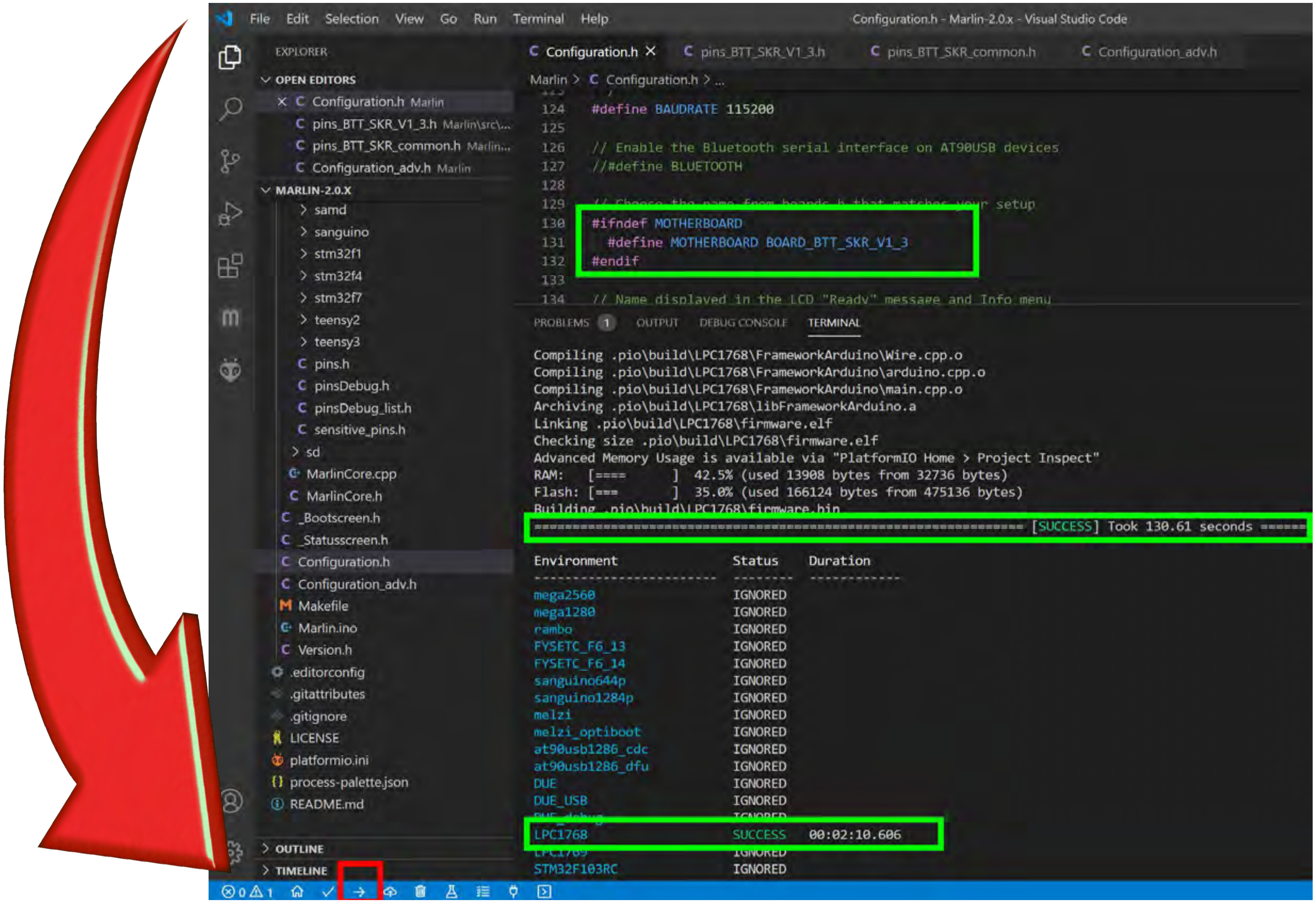
- The end of Marlin setup for MKS TMC2100 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2130

Stand-alone Mode

R - Stall Detection

Stand-alone Mode

base of card

	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

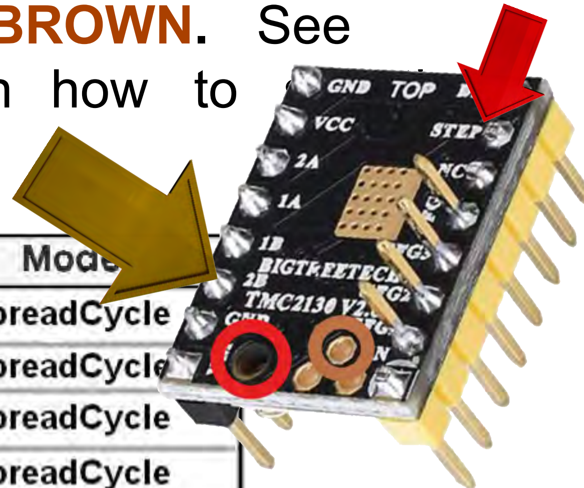
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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

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



Driver Chip	MS0	MS1	Steps	Interpolation	Mode
BIQU® TMC2130 Stand Alone Mode; Maximum 16 Subdivision; 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop

Driving Current Calculation Formula	$I_{MAX} = V_{ref}$	$V_{ref} = I_{MAX}$
R_S(Typical Sense Resistor)= 0.11Ω	See Appendix B #2. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	See Appendix B #2. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU TMC2130

Stand-alone Mode

R - Stall Detection

Stand-alone Mode

base of card

	0	1
1	0	0
2	1	0
2	-	0
4	0	1
4	-	1
4	1	-
16	1	1
16	0	-
16	-	-

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Driver Chip Chart:

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

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

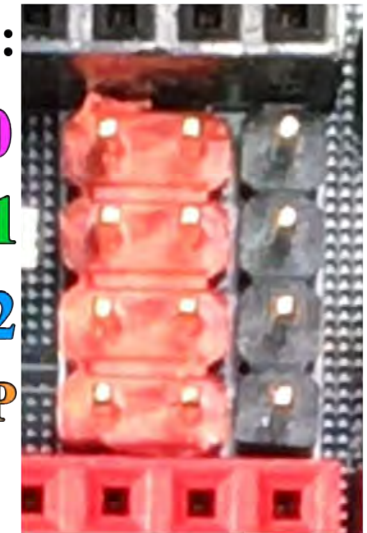
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1

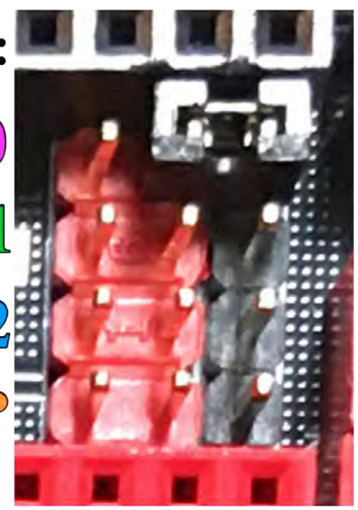
OPEN → No Jumper

Rows: MS0 MS1 MS2 RST/SLP



High → Jumper Set

Rows: MS0 MS1 MS2 RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU TMC2130

Stand-alone Mode

Driver Chip Chart:

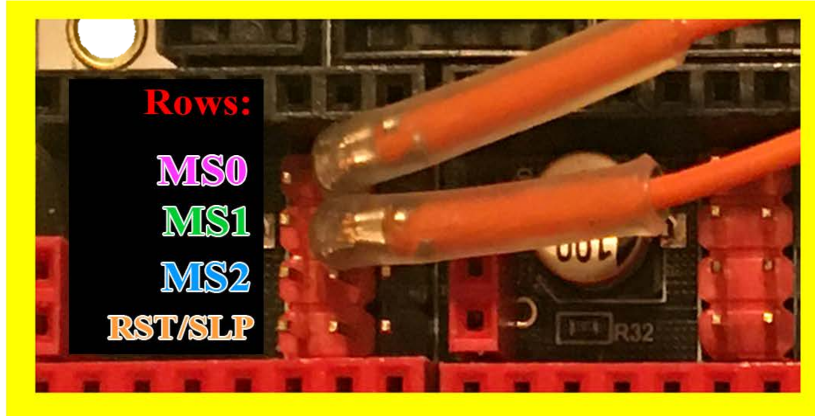
Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO	Low	Low	1	NONE	spreadCycle
TMCxxxx	High	Low	1 / 2	NONE	spreadCycle
Stand Alone Mode	OPEN	Low	1 / 2	1 / 256	spreadCycle
Maximum 16 Subdivision	Low	High	1 / 4	NONE	spreadCycle
46V DC 2.5A (peak)	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256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

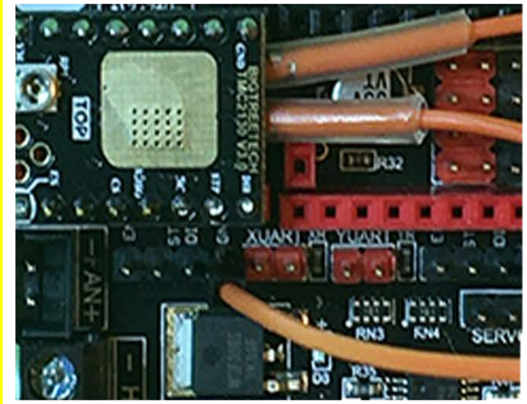
Driving Current Calculation Formula R_S(Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B #x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$	$V_{ref}=I_{MAX}$ See Appendix B #x. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$

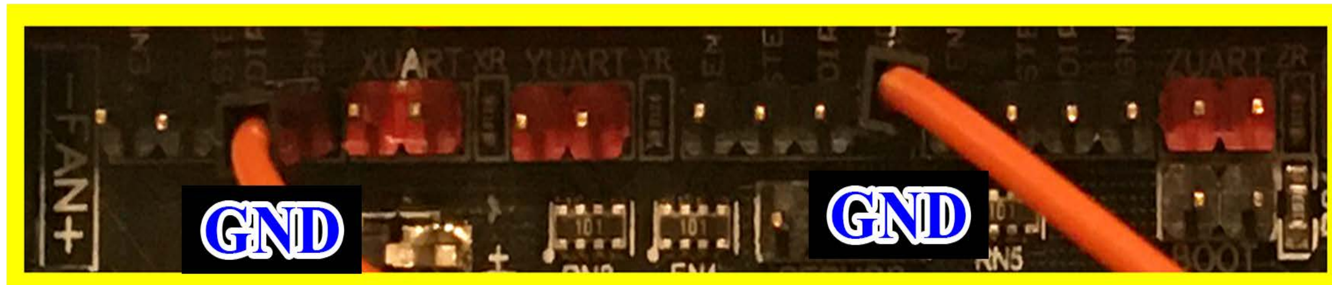
Low ⇨ MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable

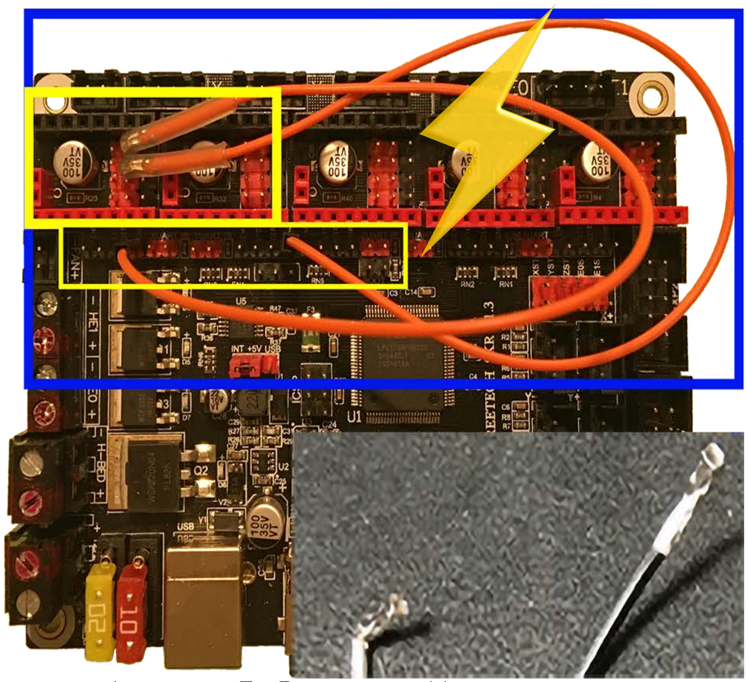
Note: See next page for instructions on how to create a DuPont jumper cable.

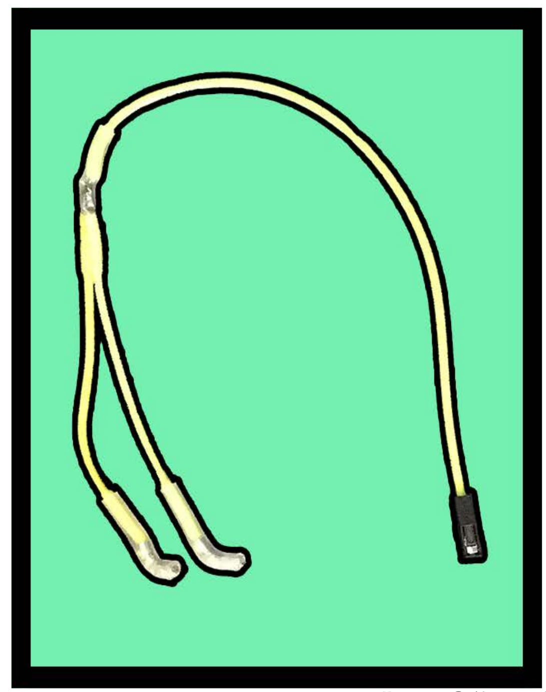
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 2







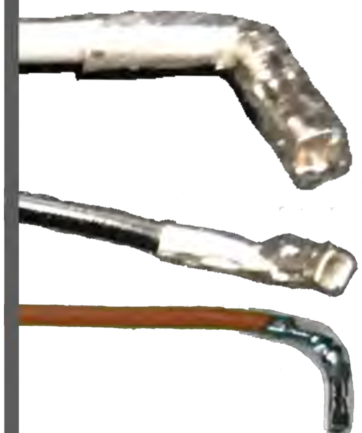




How to Create a SKR V1.3 Dupont Jumper Cable to Use with Tri State Drivers

Remove DuPont cover from one end of a female to female DuPont jumper wire.

Place a single DuPont pin into exposed female end and bend the connector backwards 90 degrees.

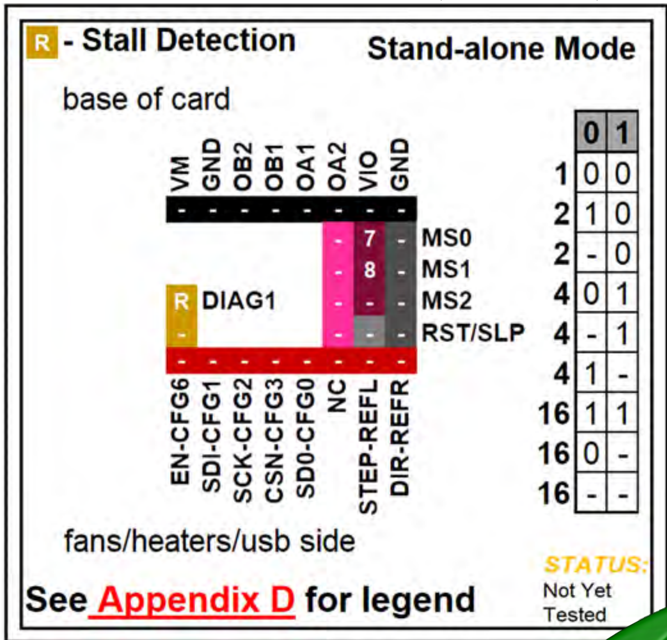


Since bending the connector weakens it, place a bead of solder at the 90 degree bend. I inserted a single DuPont pin inside the female connector to ensure that the connector stays clear of solder. After soldering is done, move the inserted DuPont pin around to ensure a functioning right angle female connector.

Place heat shrink over the exposed female DuPont connector to protect the exposed metal from causing shorts. Picture here is a finished jumper cable with a single DuPont pin inserted.

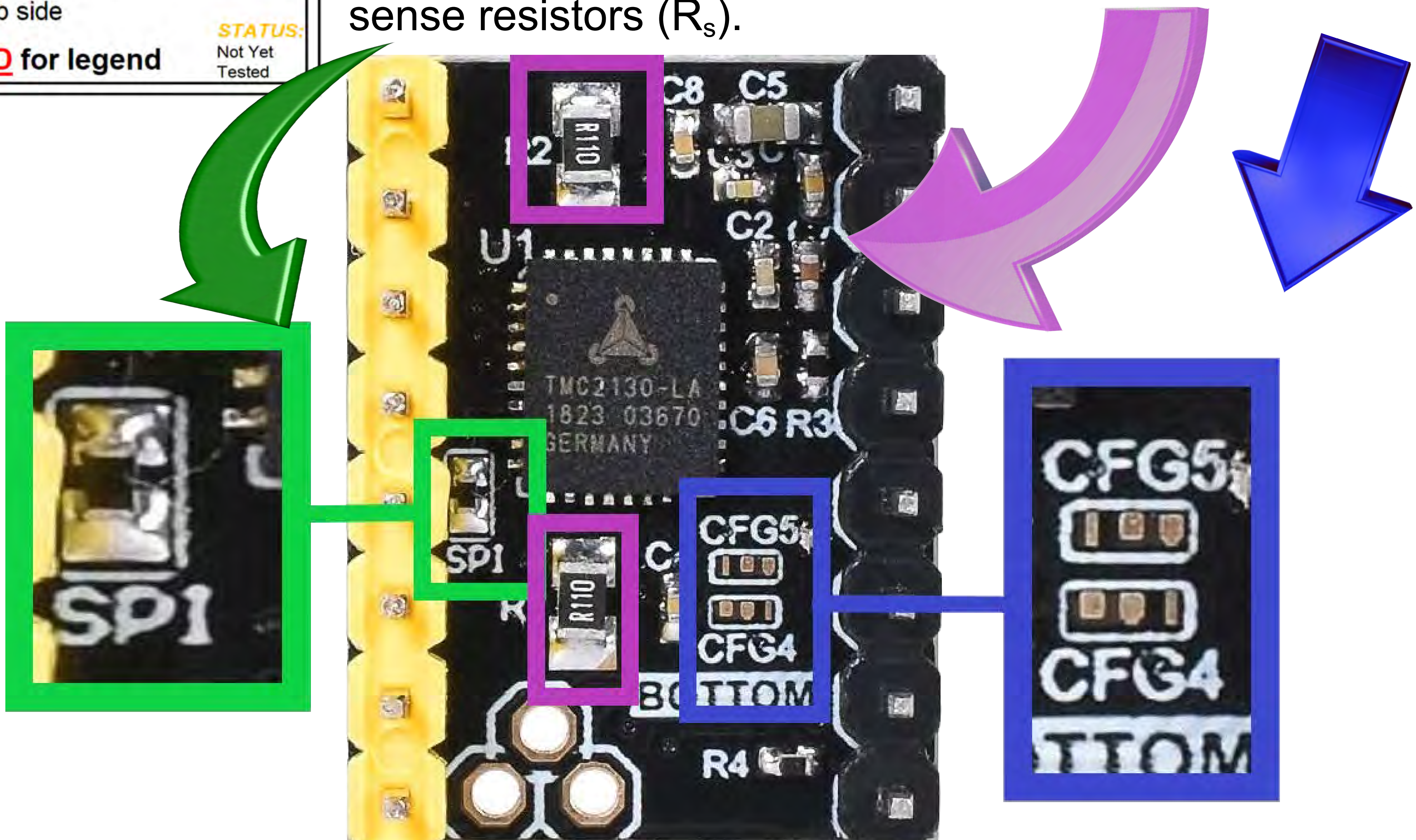
BIQU TMC2130

Stand-alone Mode



To place the BIQU TMC2130 into Stand-alone mode:

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

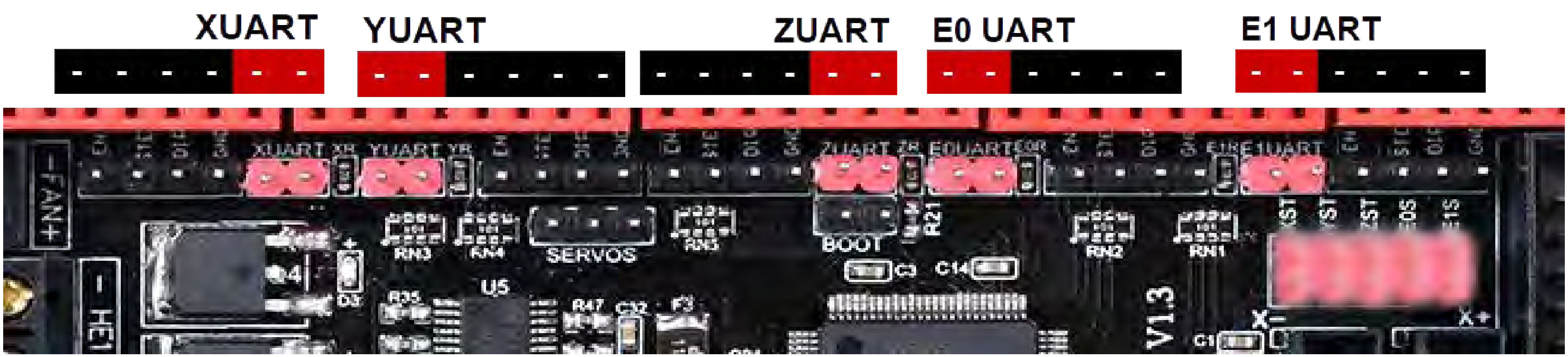


Stand-alone Mode

BIQU TMC2130
Stand-alone Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

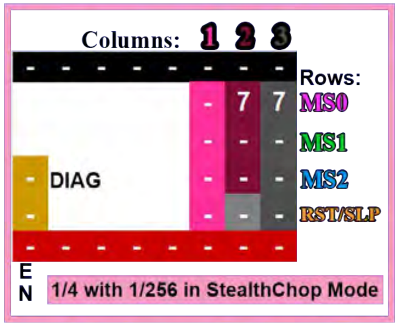
Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

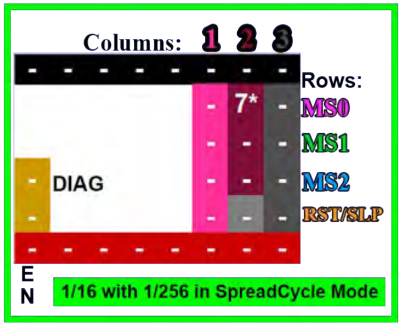


Stand-alone Mode

SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

Driver Socket Representation



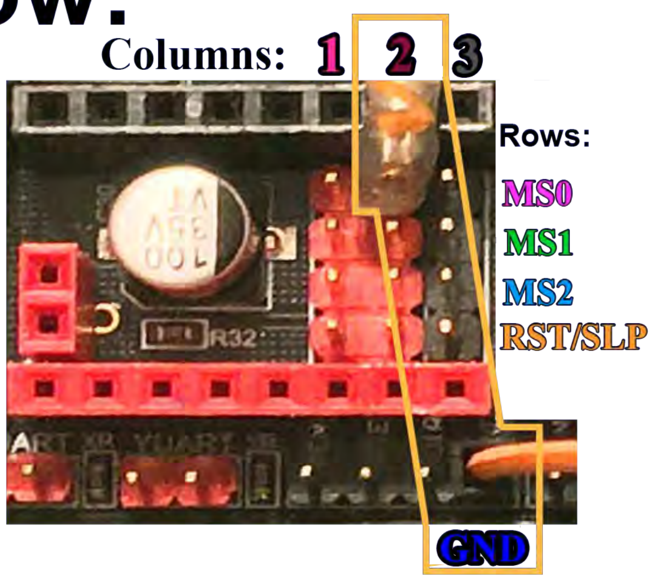


Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX}$ * 0.90			$V_{ref}=I_{MAX}$ See Appendix B x. Use 50% to 90% as shown below: $V_{ref}=V_{ref}$ * 0.90	

- - → No Jumper set

Low:



MS0 for Tri State Drivers ONLY (TMC2100 & TMC2130):

Driver Socket Representation:	Driver Chip Chart:
7 7	High
7* -	Low
- -	Open

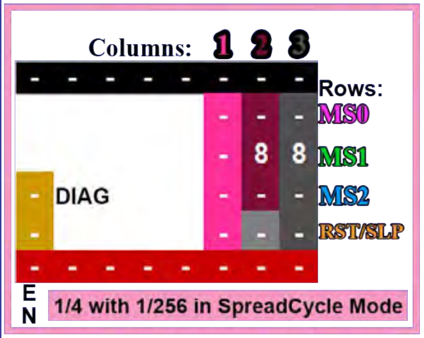
Driver Chip Chart:	Meaning:
High	set Jumper between column 2 and column 3 on the MS0 row
Low	Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)
Open	No Jumper set

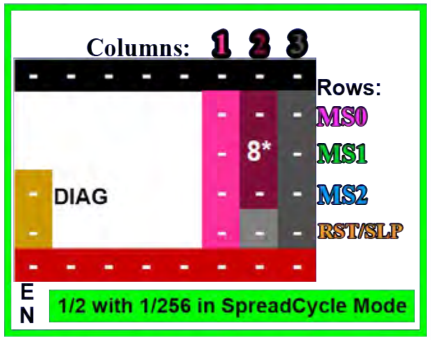
High:

Stand-alone Mode

BIQU TMC2130
Stand-alone Mode

Driver Socket Representation



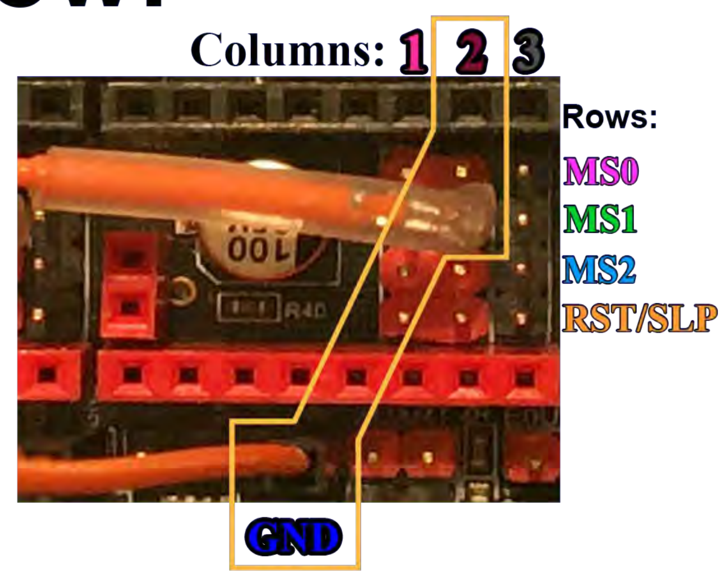


Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$		$V_{ref}=I_{MAX}$ See Appendix B x. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$		

- - ⇒ No Jumper set

Low:



MS1 for Tri State Drivers ONLY (TMC2100 & TMC2130):

Driver Socket Represntation:	Driver Chip Chart:
8 8 ⇒	High
8* - ⇒	Low
- - ⇒	Open

Driver Chip Chart:	Meaning:
High ⇒	set Jumper between column 2 and column 3 on the MS1 row
Low ⇒	Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)
Open ⇒	No Jumper set

High:

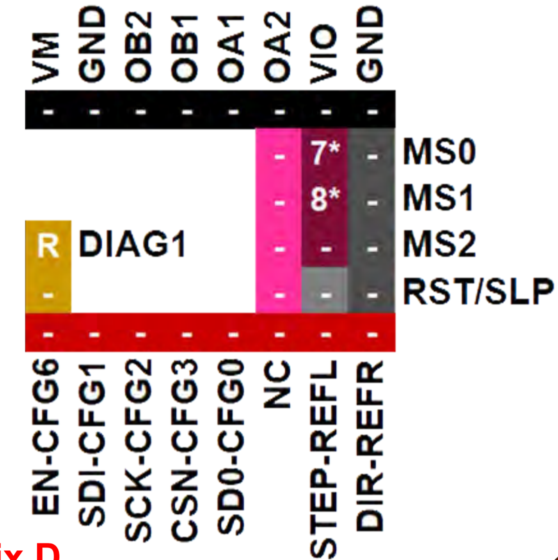
Stand-alone Mode

Stand-alone Mode

STEP

Interpolation: none

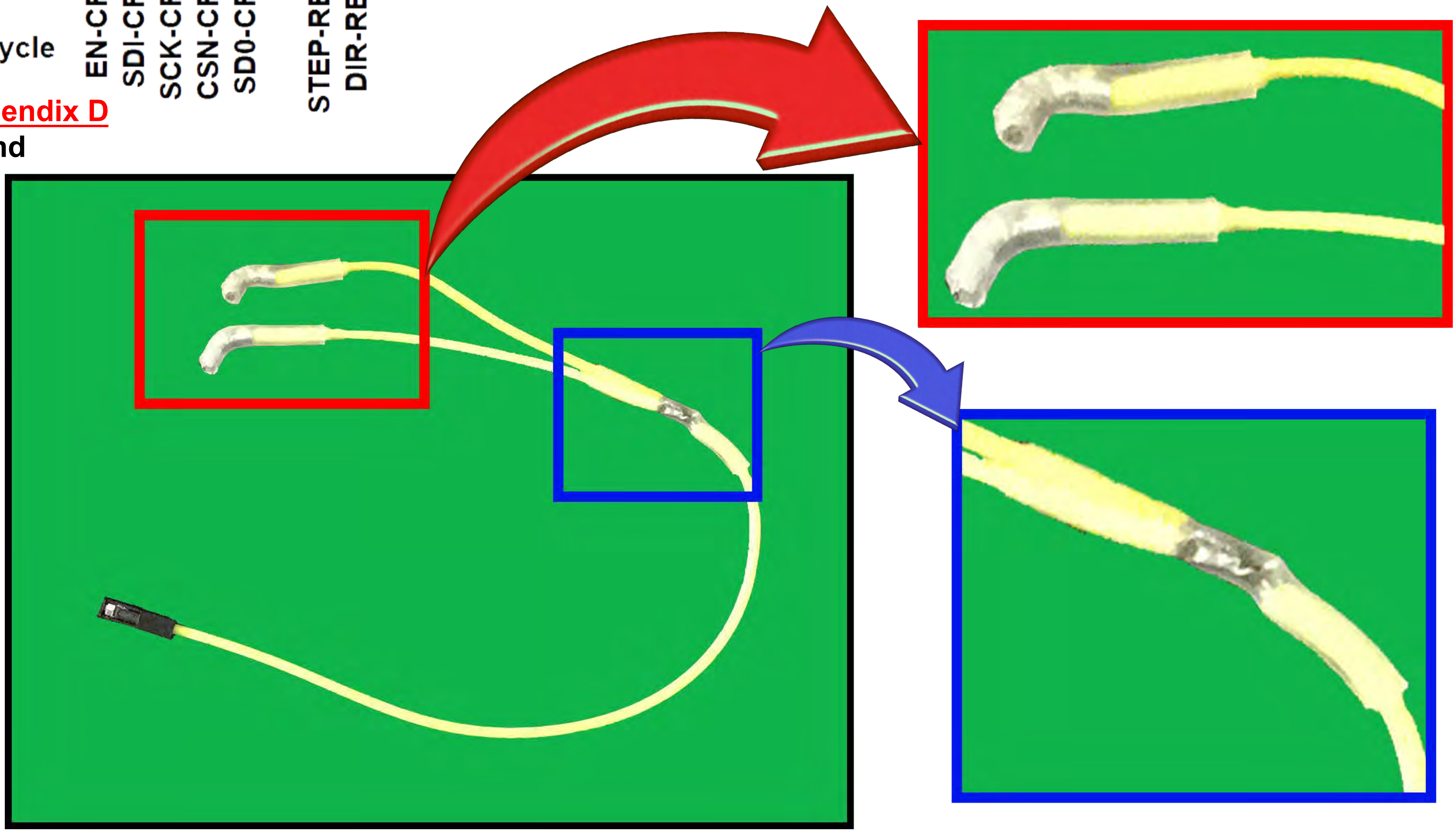
SpreadCycle



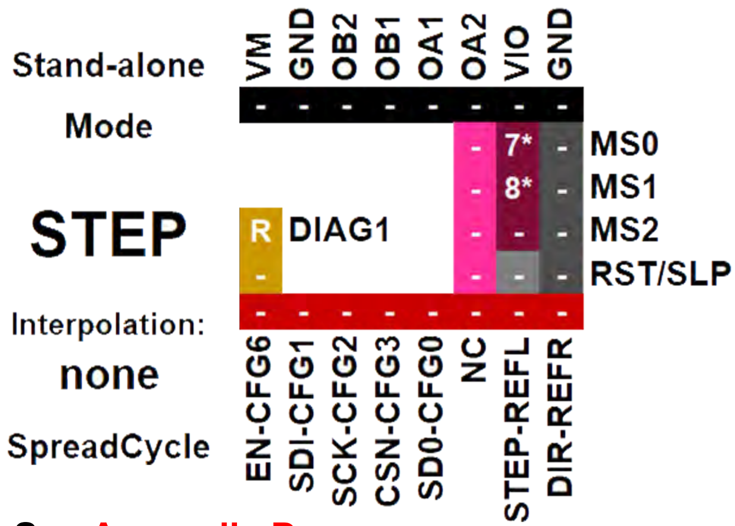
See Appendix D for legend

Additional Equipment Needed for Low Low (STEP or FULL) Configuration

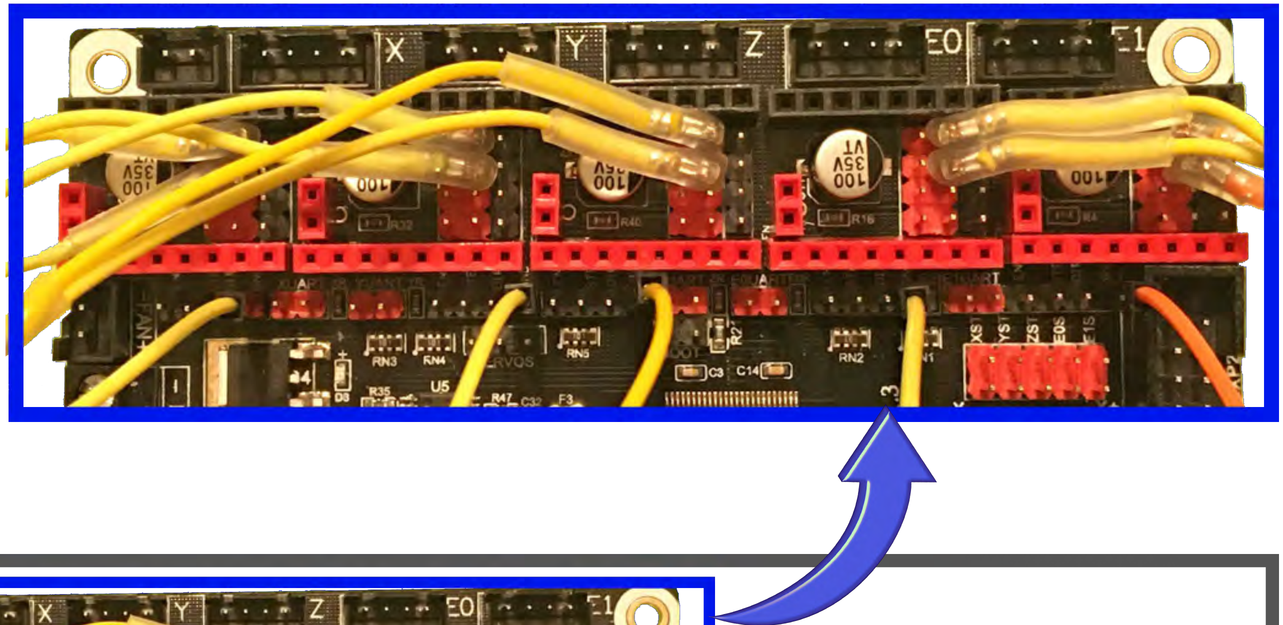
You will need one DuPont jumper cable that connects to a ground point (GND) on the SKR V1.3 board but will source two PINS simultaneously. So create a Y-cable that has one female connector on one end and has two female connectors on the other end.

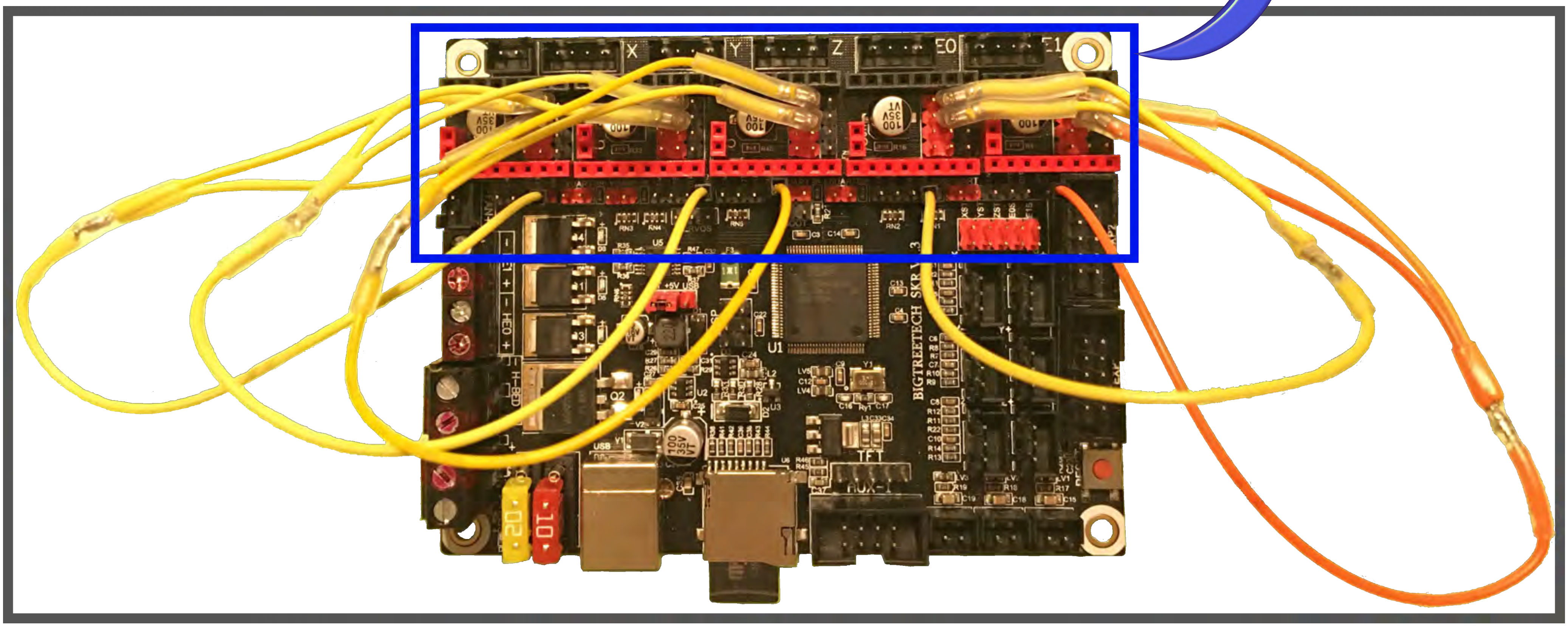


Stand-alone Mode



See Appendix D
for legend





Stand-alone Mode

Stand-alone

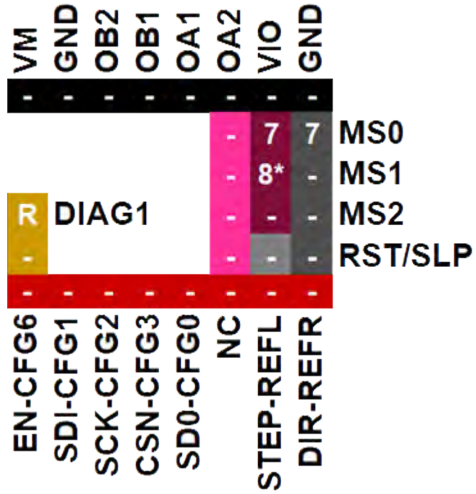
Mode

1 / 2

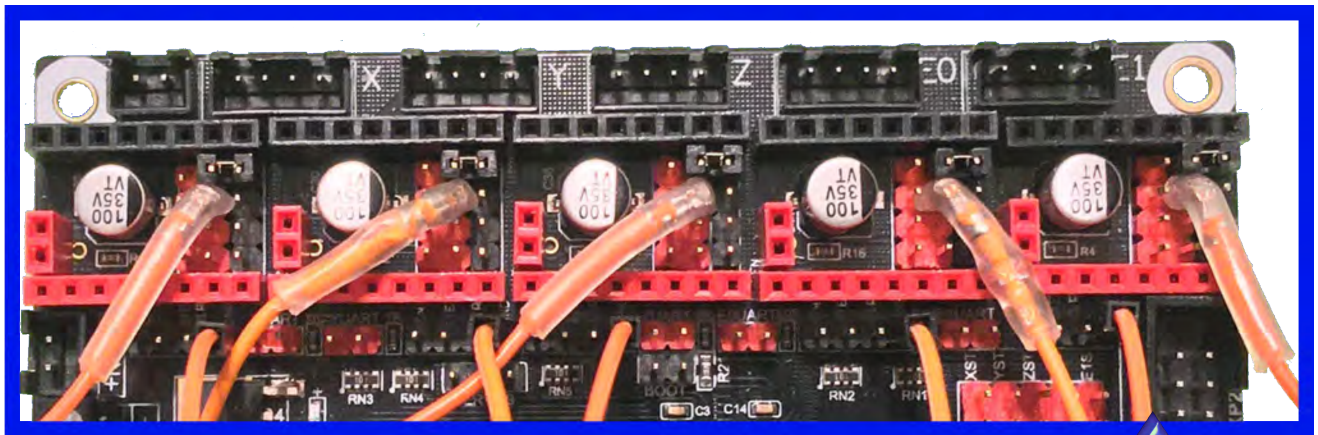
Interpolation:

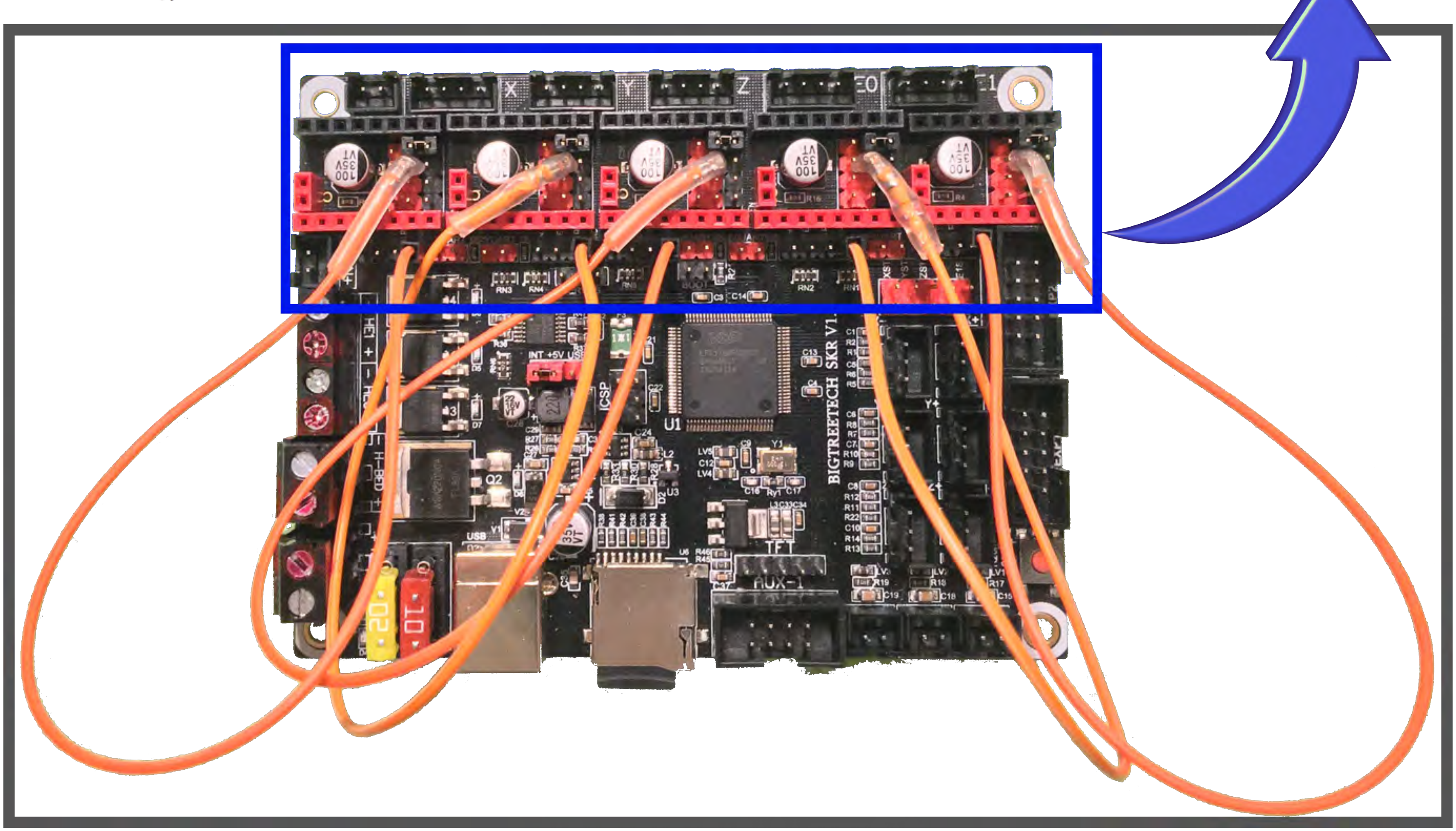
none

SpreadCycle



See Appendix D
for legend





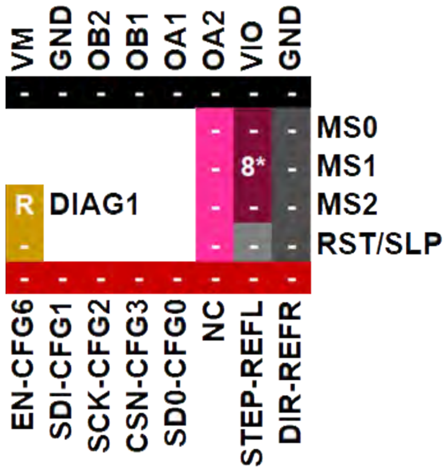
Stand-alone Mode

Stand-alone Mode

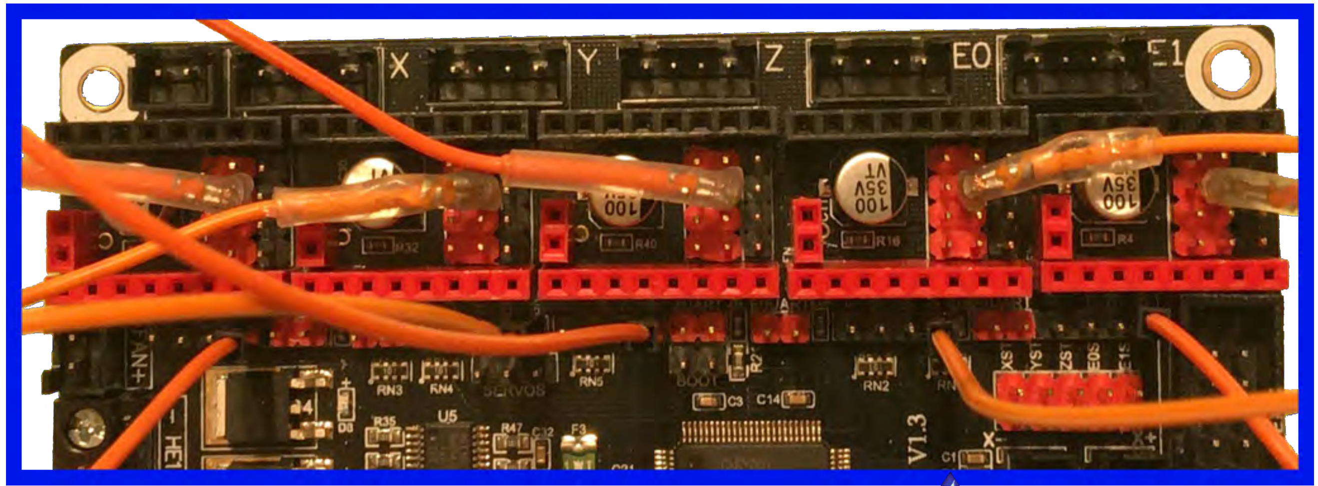
1 / 2

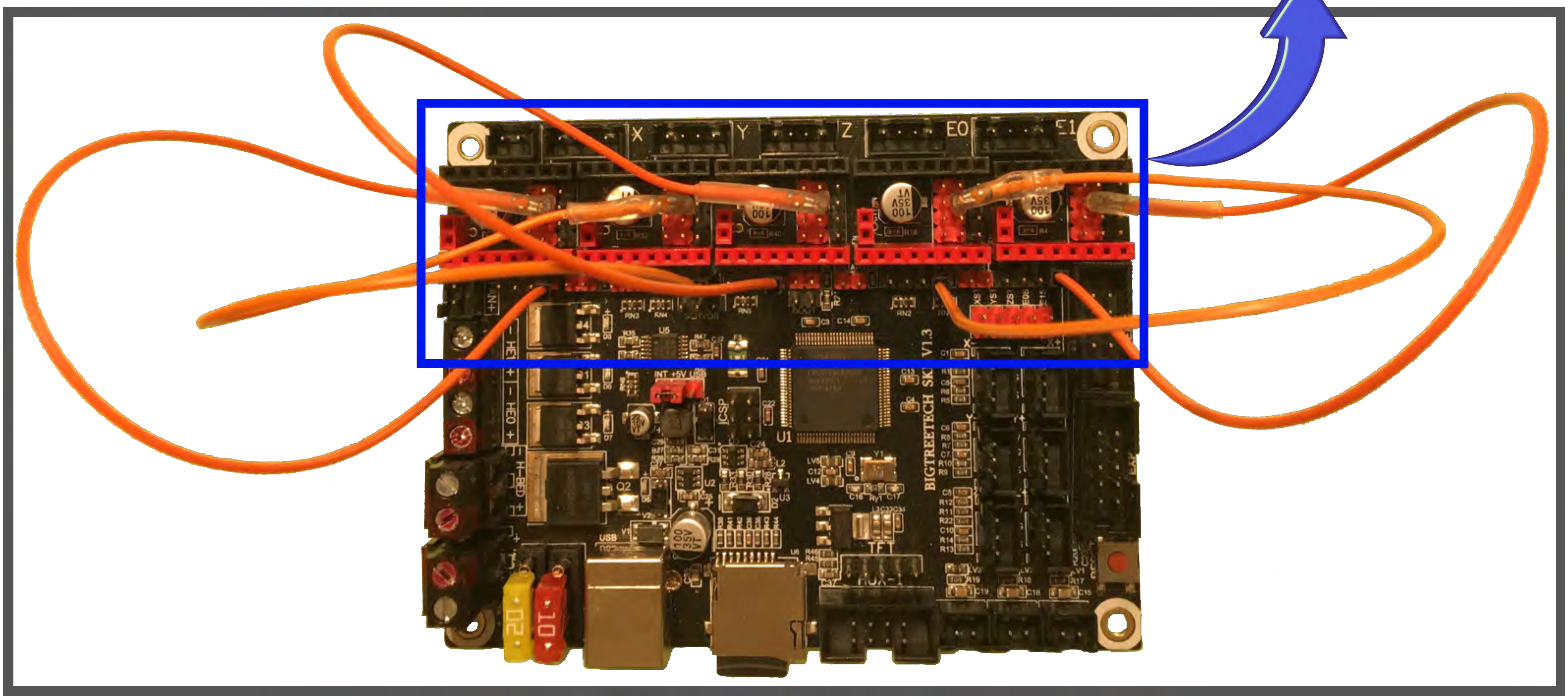
Interpolation:
1 / 256

SpreadCycle



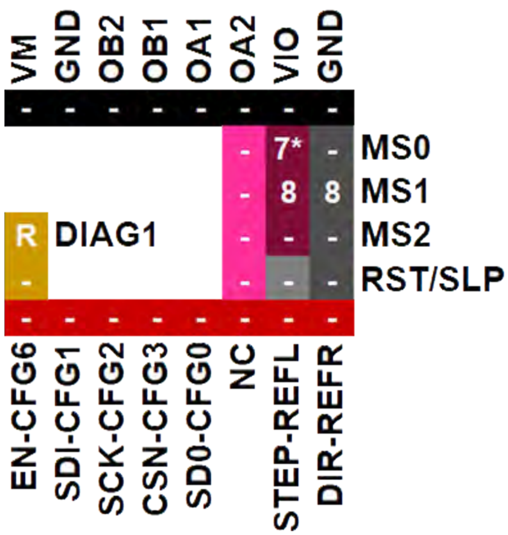
See Appendix D for legend





Stand-alone Mode

Stand-alone Mode

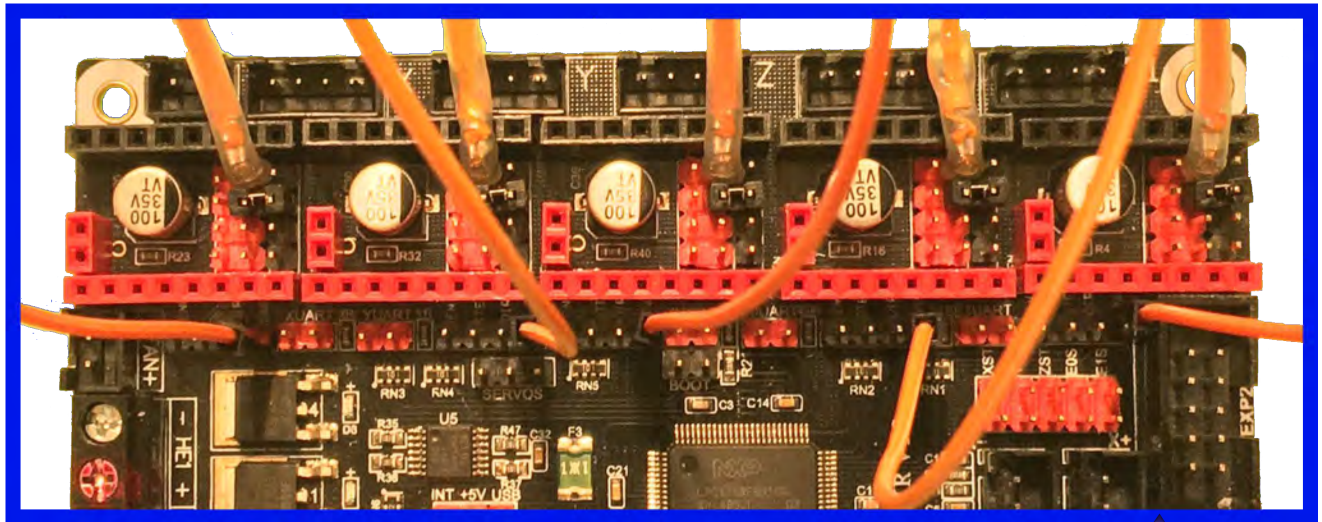


1 / 4

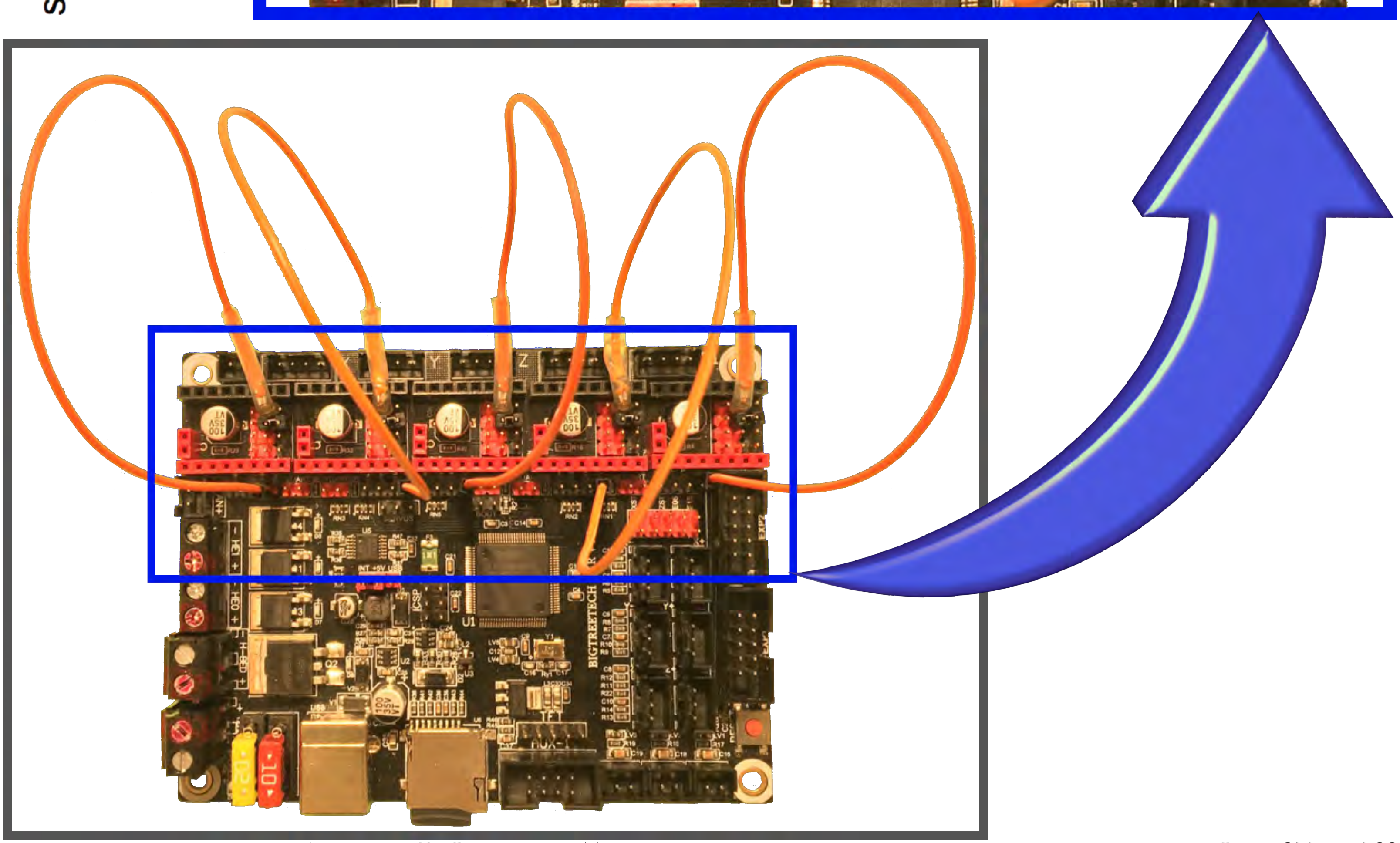
Interpolation:
none

SpreadCycle

Stand-alone Mode



See Appendix D for legend



Stand-alone Mode

BIQU TMC2130

Stand-alone Mode

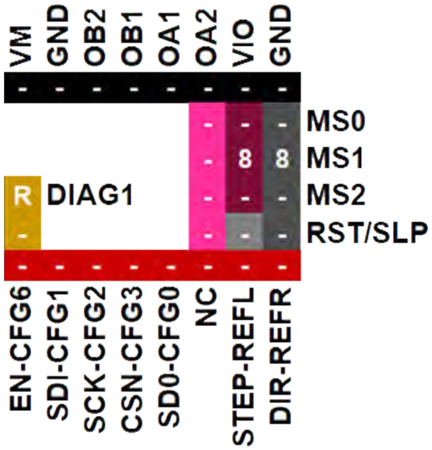
Stand-alone Mode

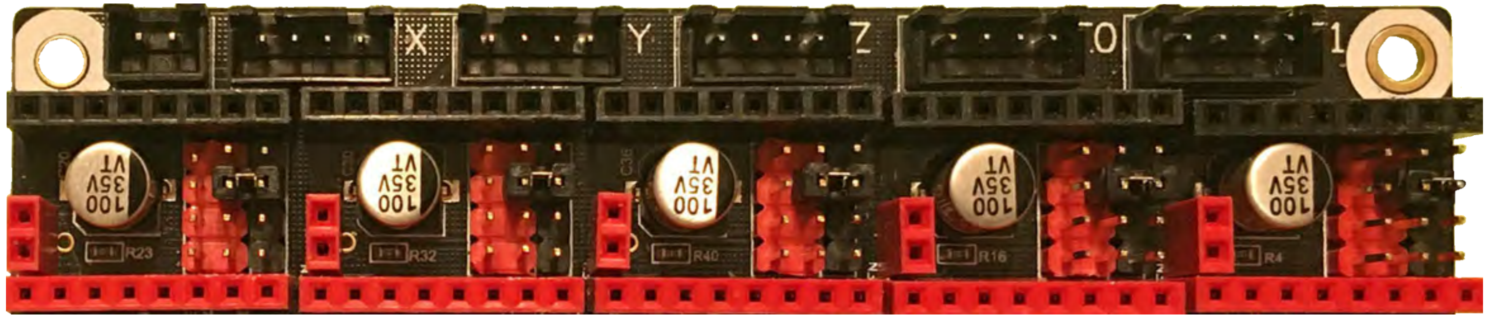
1 / 4

Interpolation:

1 / 256

SpreadCycle





See Appendix D for legend

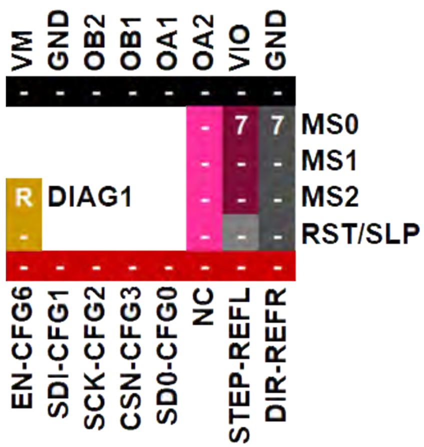
Stand-alone Mode

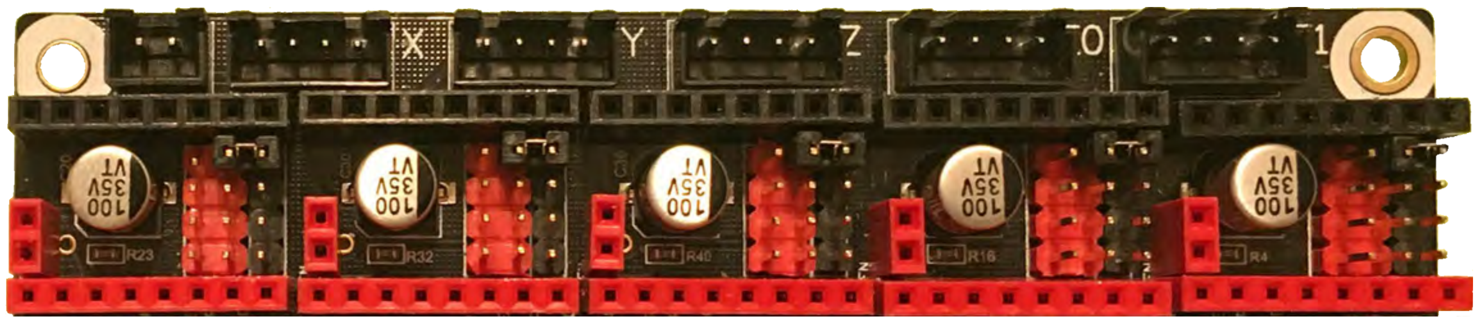
1 / 4

Interpolation:

1 / 256

StealthChop





See Appendix D for legend

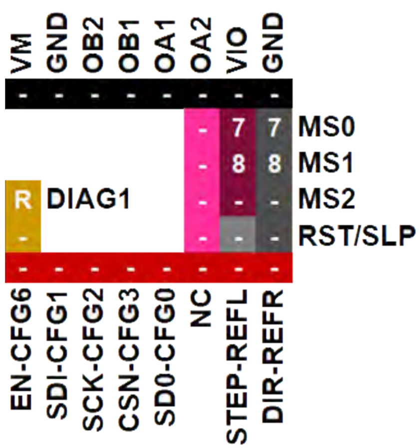
Stand-alone Mode

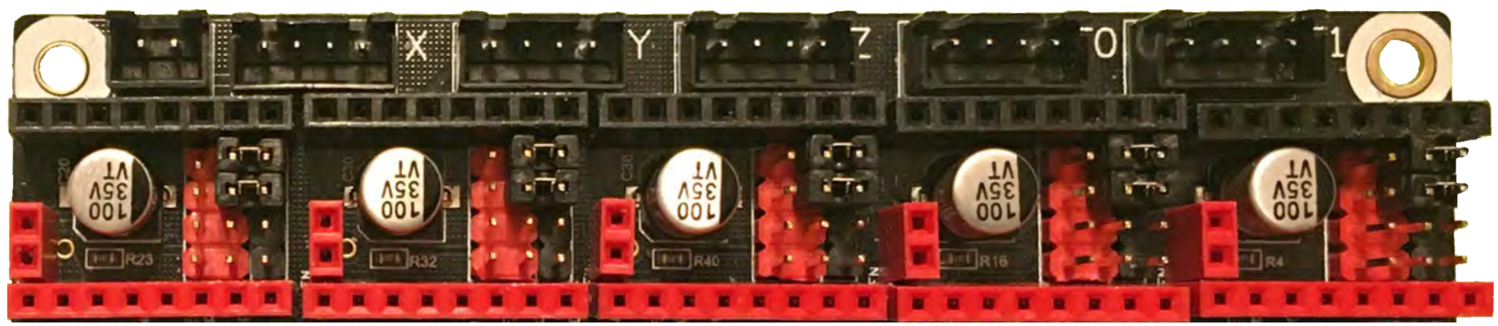
1 / 16

Interpolation:

none

SpreadCycle





See Appendix D for legend

Stand-alone Mode

BIQU TMC2130

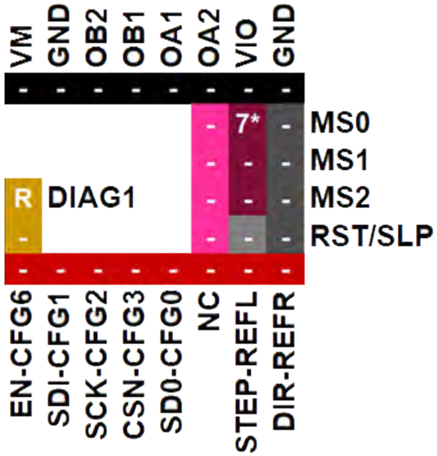
Stand-alone Mode

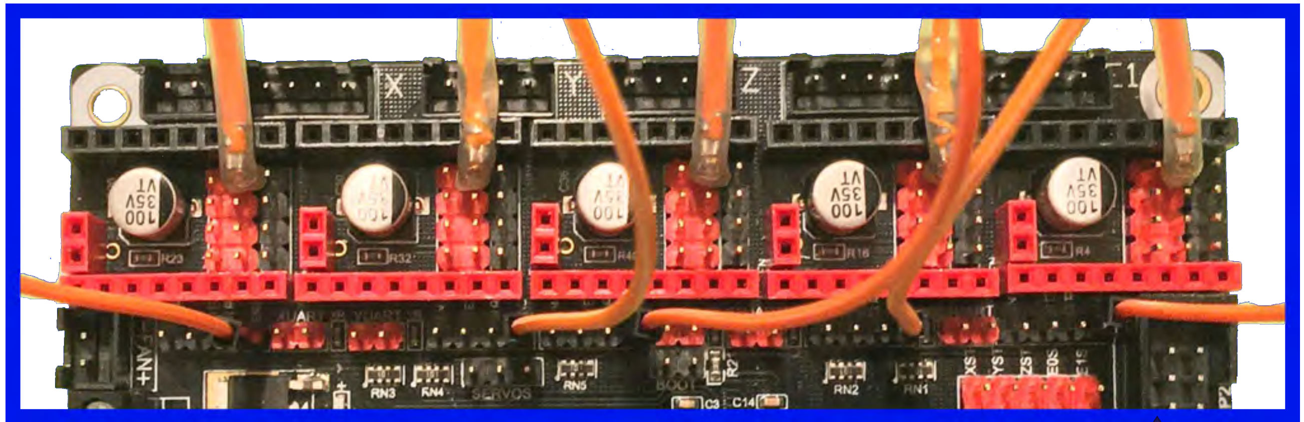
Stand-alone Mode

1 / 16

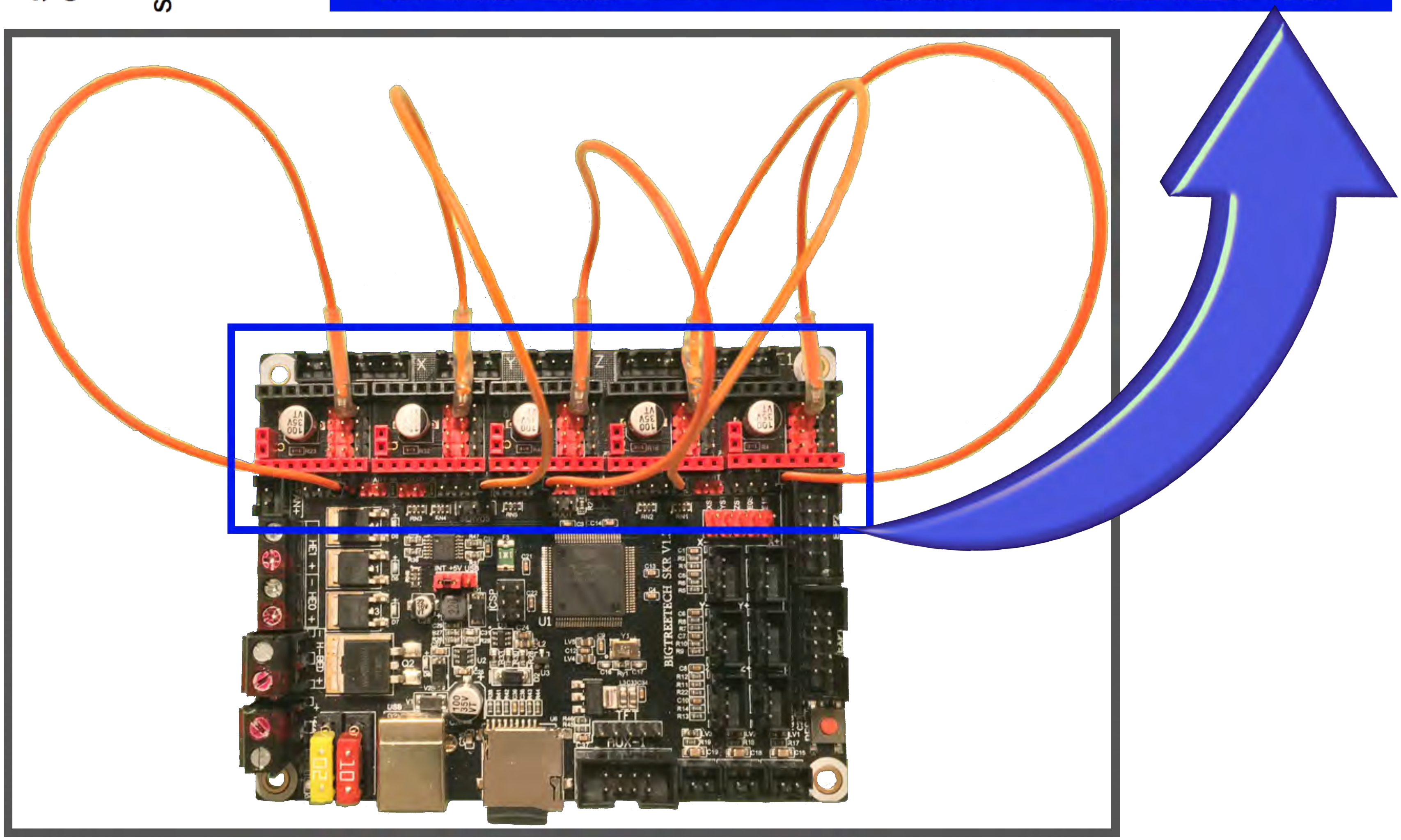
Interpolation: 1 / 256

SpreadCycle



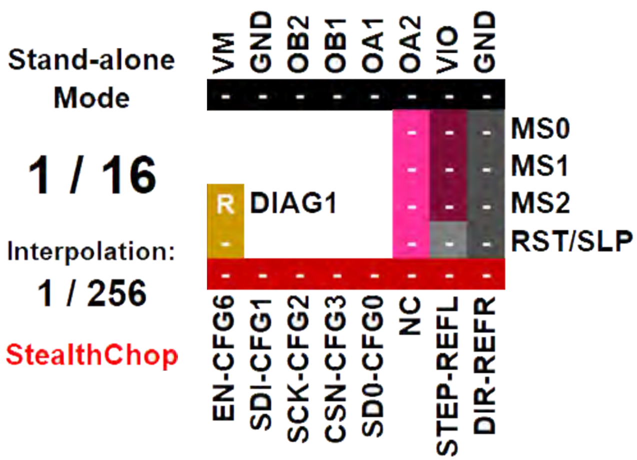


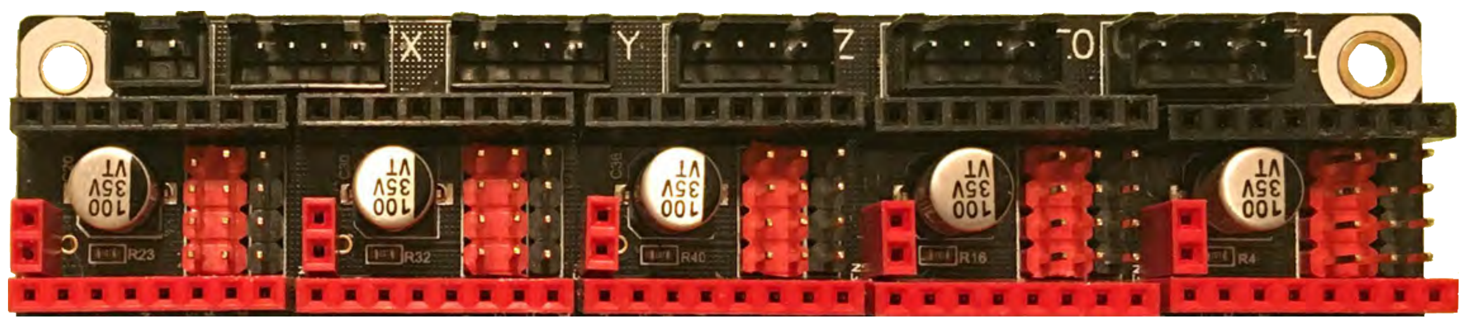
See Appendix D for legend



Stand-alone Mode

BIQU TMC2130
Stand-alone Mode



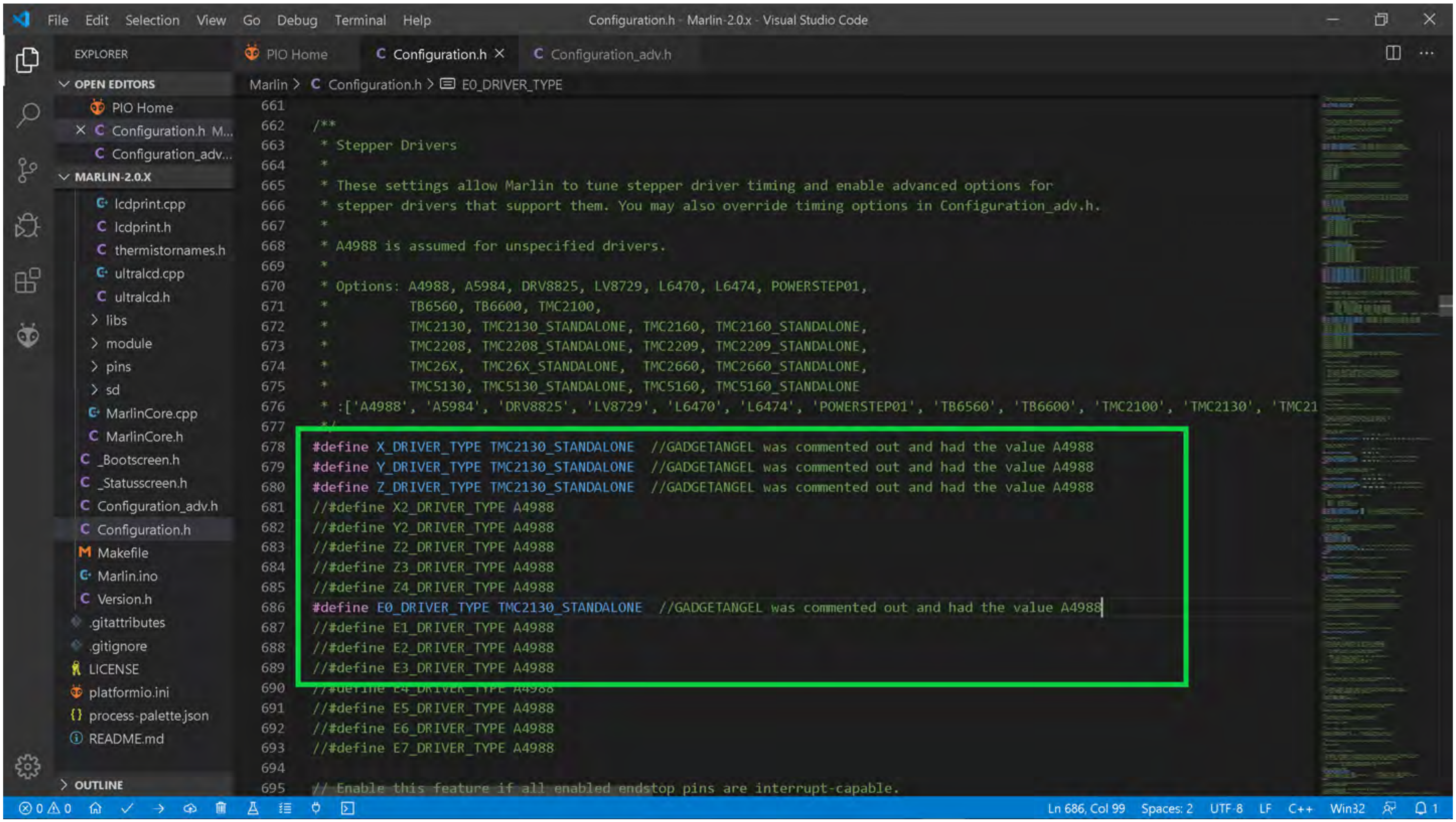


See Appendix D for legend

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

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

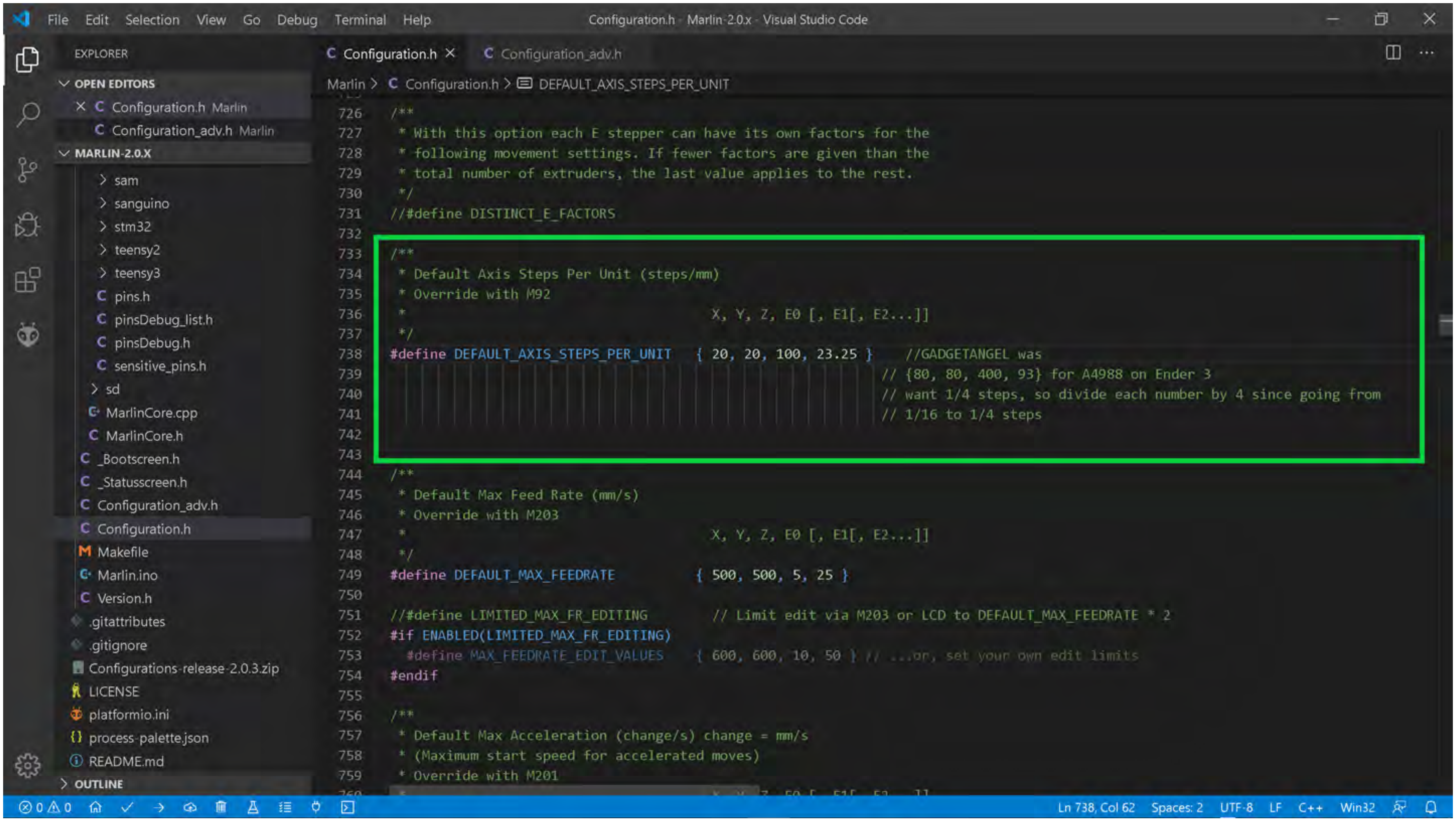
- Change the stepper motor drivers so that Marlin knows you are using TMC2130 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

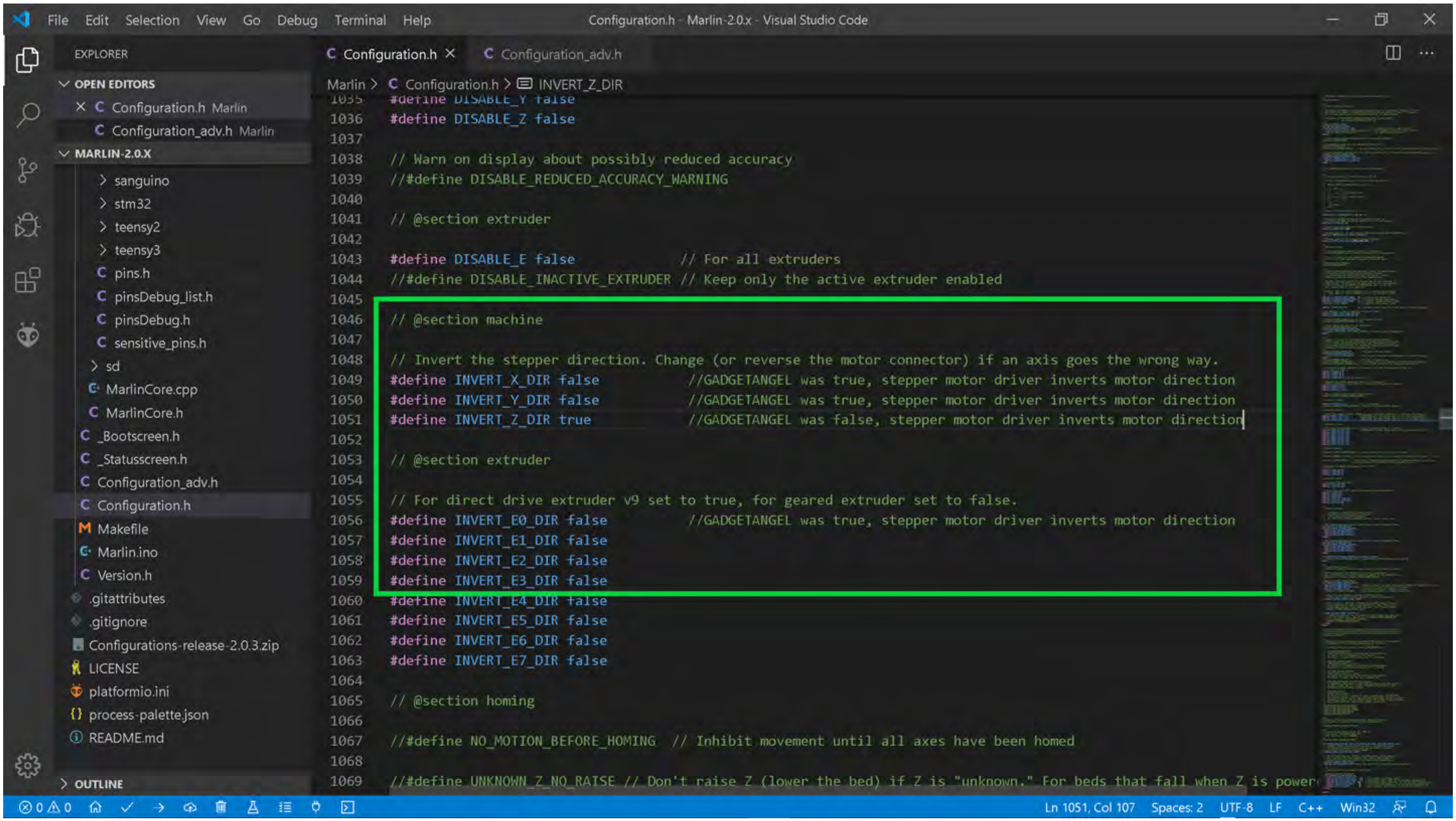
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to BIQU TMC2130 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

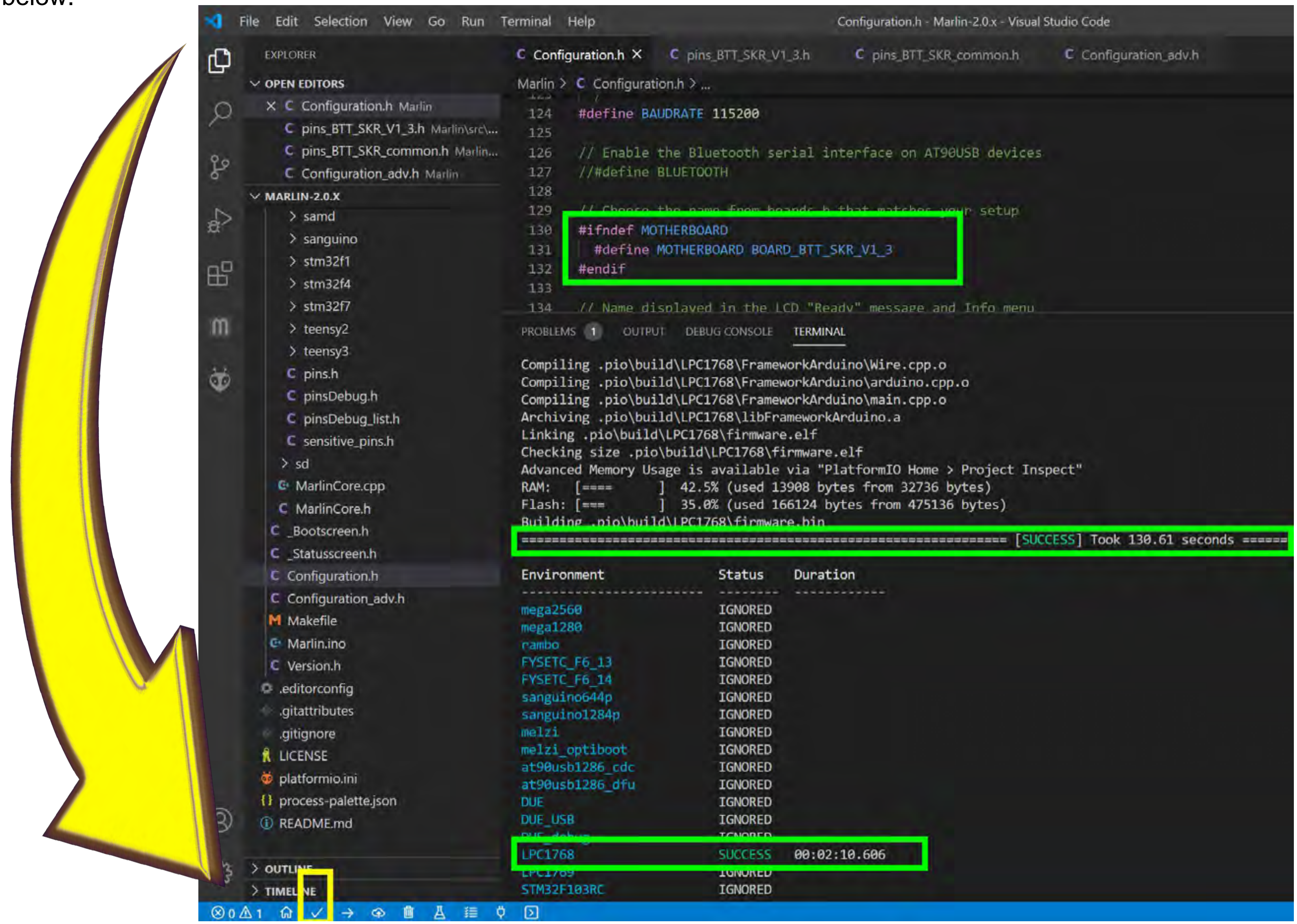
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130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



- Go to the next page.

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

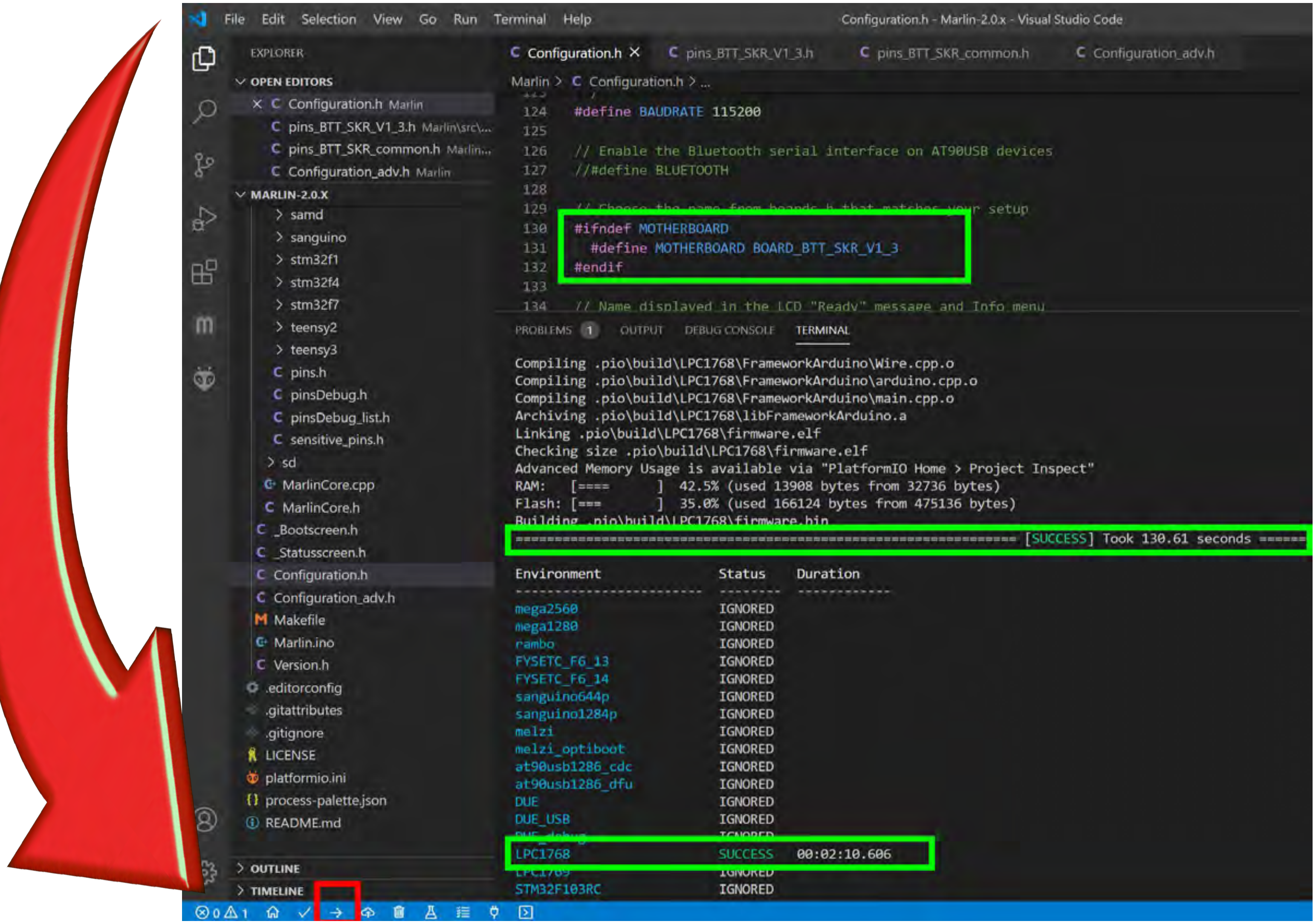
- The end of Marlin setup for BIQU TMC2130 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

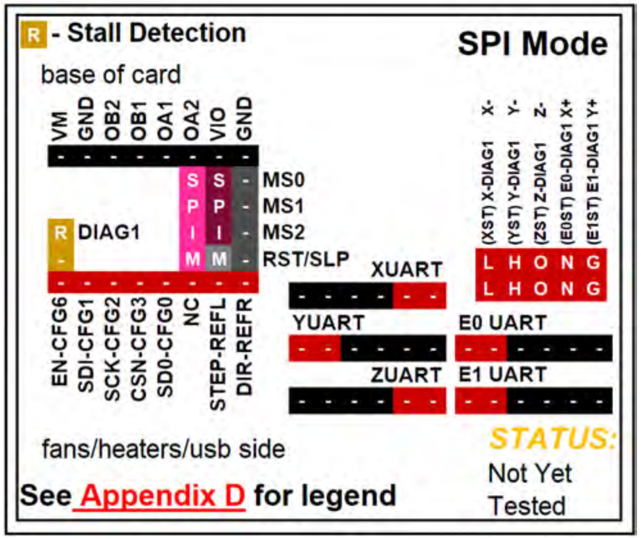
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2130

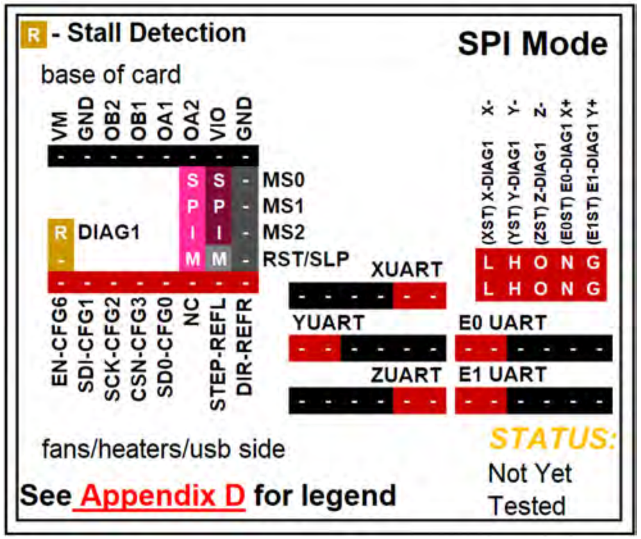
SPI Mode



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

See the next page for further information.

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

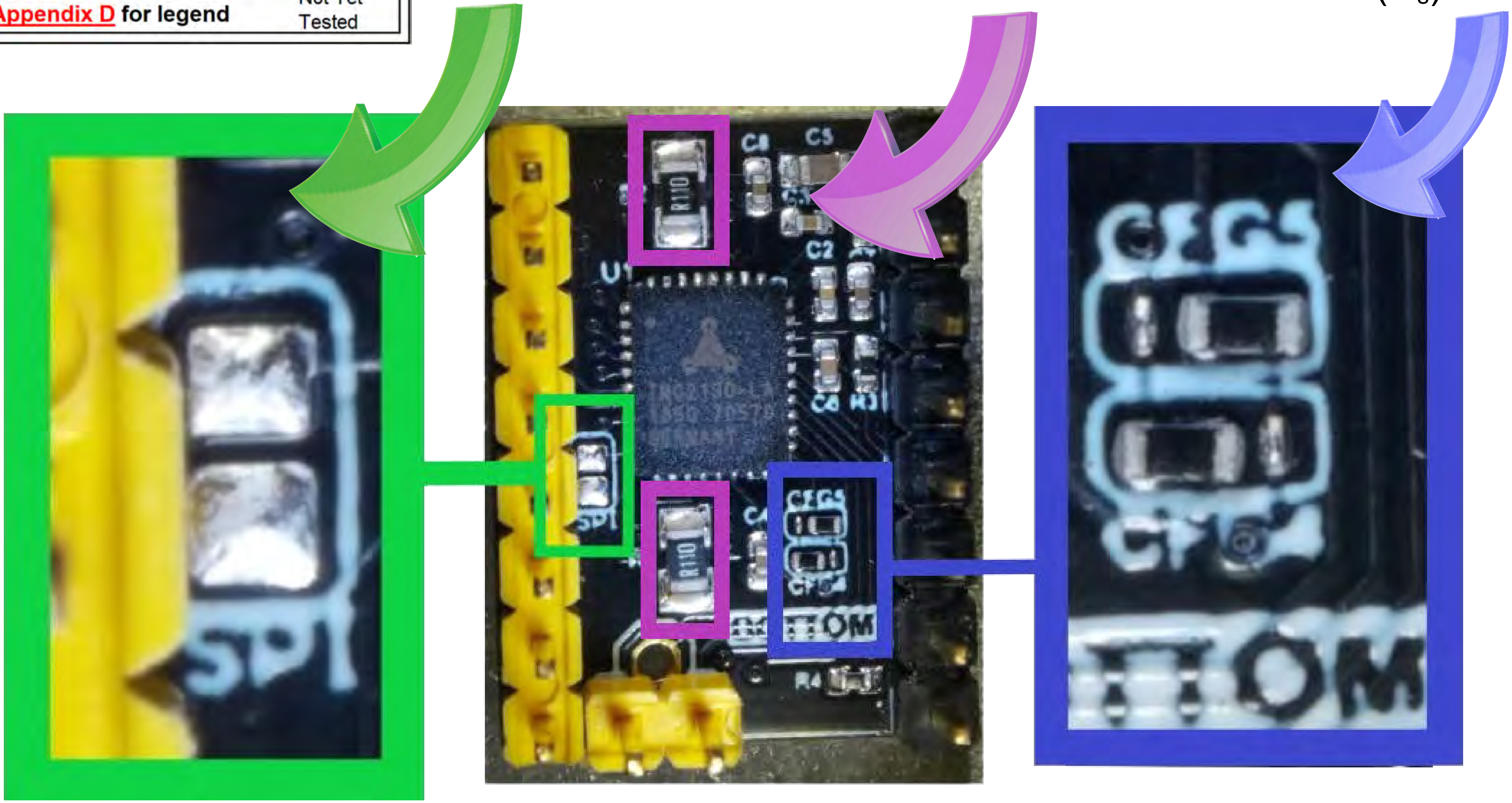


BIQU TMC2130

SPI Mode

To place the BIQU TMC2130 into SPI Mode:

The SPI Jumper must have a gap between the two SPI pads, on the bottom of the driver board, as shown in the **GREEN** box below. Ensure that at CFG4 location and CFG5 location the correct two pads are soldered together to form a bridge, as shown in the **BLUE** box. The **PURPLE** box shows the location of the current sense resistors (R_s).



SPI Mode

R - Stall Detection

BIQU TMC2130

SPI Mode

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistor's value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_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

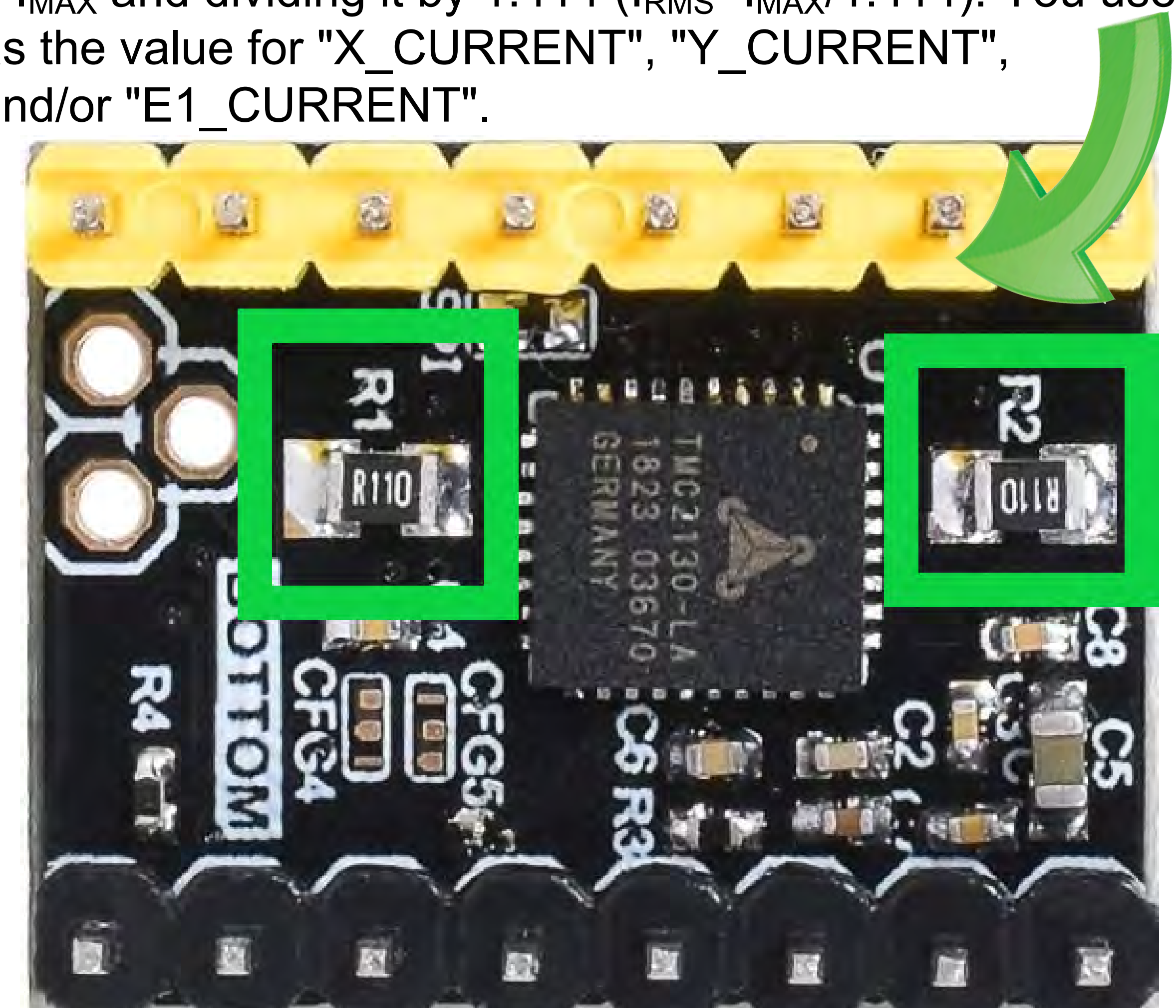
R_s = R100 is 0.1 Ohms

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



BIQU TMC2130

SPI Mode

SPI Mode

R - Stall Detection

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	-	-
L	H	O	-	-

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	N	G
L	H	O	N	G

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
-	-	-	-	-
-	-	-	-	-

Note: If sensor-less homing is not wanted ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".

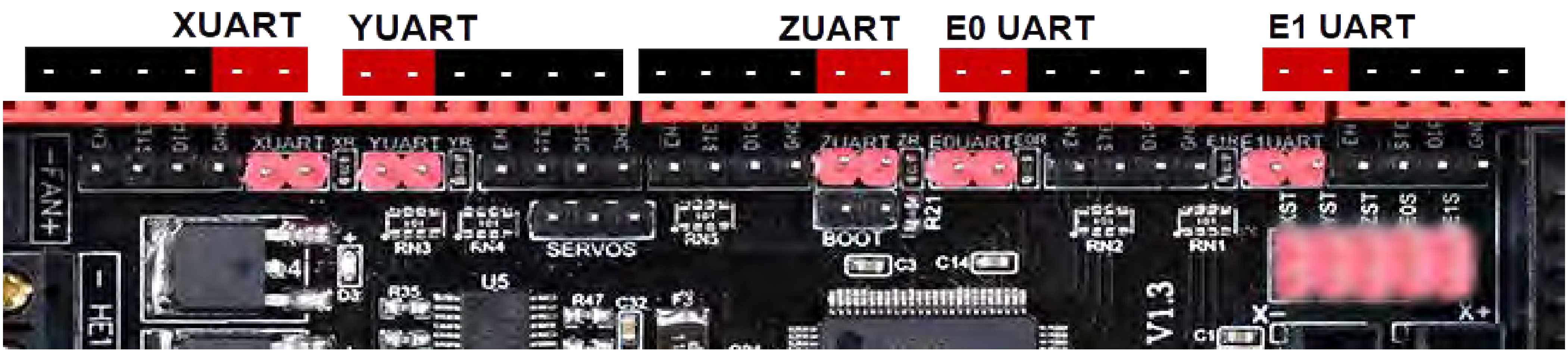
SPI Mode

R - Stall Detection

BIQU TMC2130

SPI Mode

Note: Ensure that **ALL** the UART jumpers are **EMPTY**.



SPI Mode

R - Stall Detection

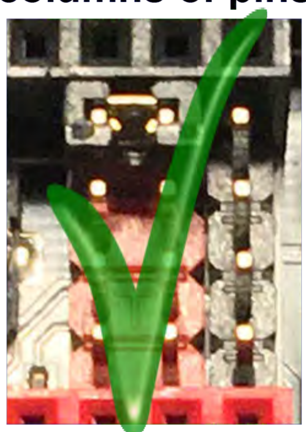
BIQU TMC2130

SPI Mode

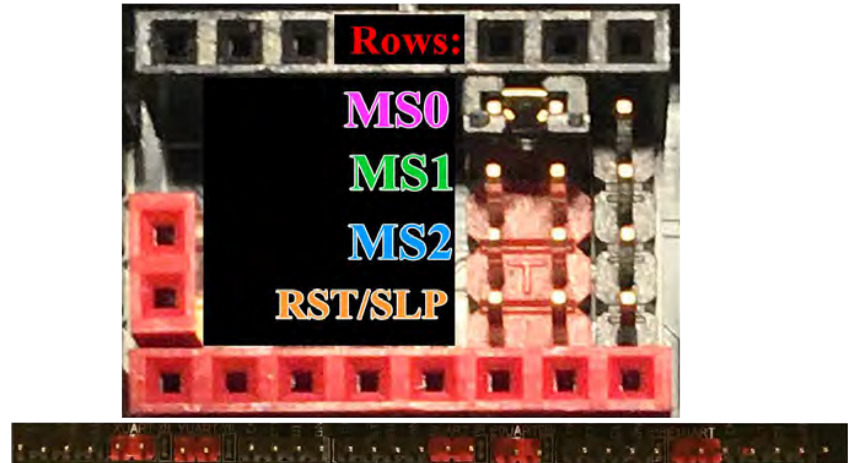
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

Example of the **WRONG** way to set the "S" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "S" jumper for SPI mode (left two columns of pins):



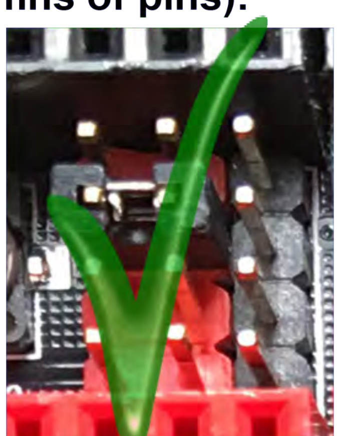
S S ⇒ Jumper set



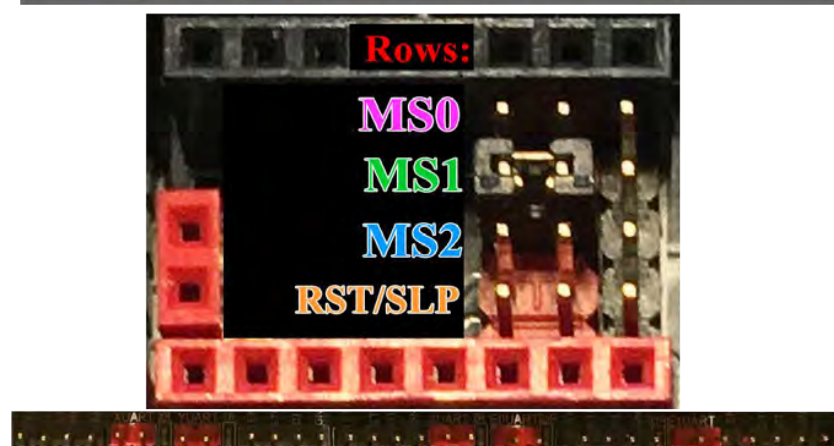
SET 'ST' JUMPER to enable Sensor-less Homing

Example of the **WRONG** way to set the "P" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "P" jumper for SPI mode (left two columns of pins):



P P ⇒ Jumper set



SET 'ST' JUMPER to enable Sensor-less Homing

BIQU TMC2130

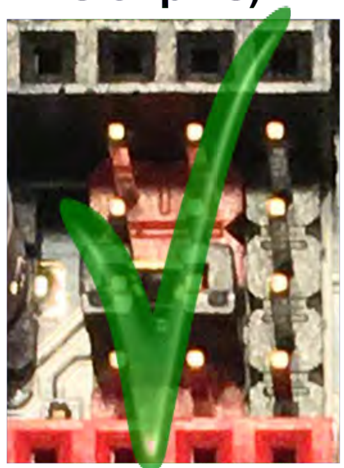
SPI Mode

SPI Mode

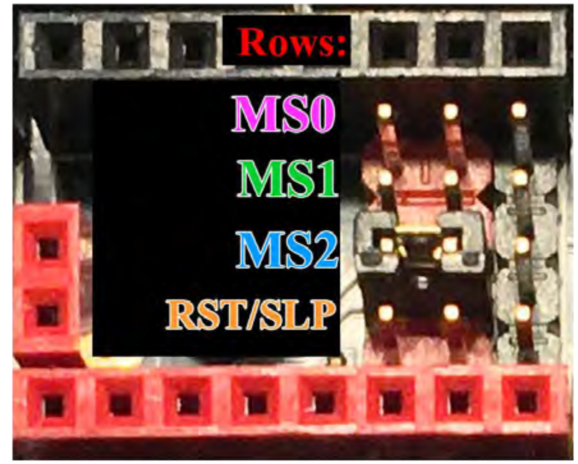
R - Stall Detection

Example of the **WRONG** way to set the "I" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "I" jumper for SPI mode (left two columns of pins):



I I ➡ **Jumper set**

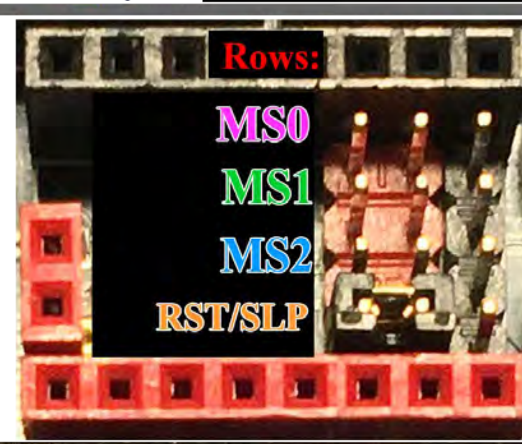


SET 'ST' JUMPER to enable Sensor-less Homing

Example of the **WRONG** way to set the "M" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "M" jumper for SPI mode (left two columns of pins):

M M ➡ **Jumper set**



SET 'ST' JUMPER to enable Sensor-less Homing

SPI Mode

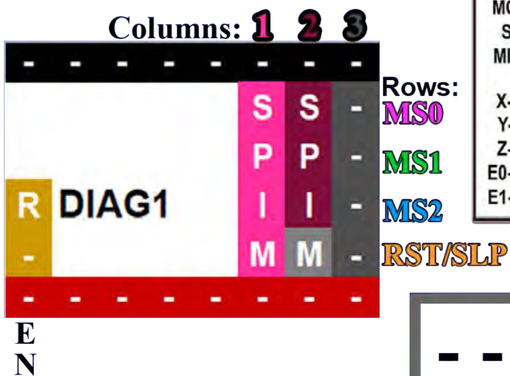
R - Stall Detection

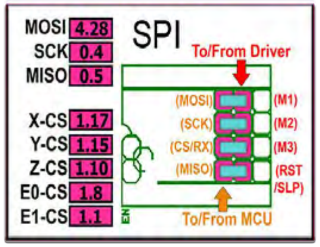
BIQU TMC2130

SPI Mode

SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

Driver Socket Representation



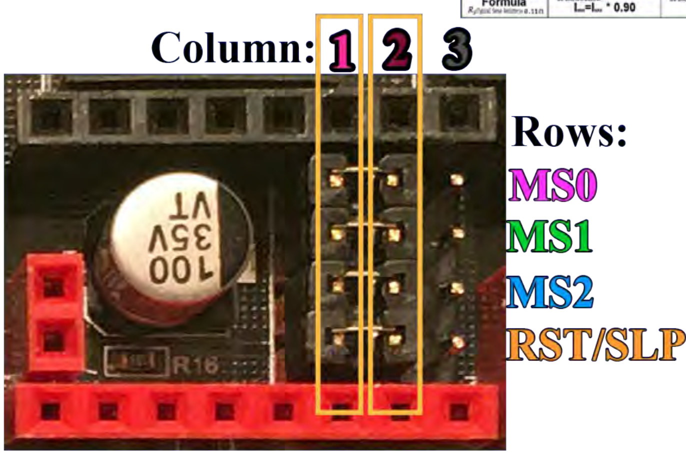


- - No Jumper set

NOTE: UART PINS must not be set if SPI mode is ensbled.

SPIM:

Driver Chip Chart:



SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

S S
P P
I I
M M

Meaning:

Jumpers Set

Driver Socket Representation: S S

Meaning: **set Jumper between column 1 and column 2 on the MS0 row**

Driver Socket Representation: P P

Meaning: **set Jumper between column 1 and column 2 on the MS1 row**

Driver Socket Representation: I I

Meaning: **set Jumper between column 1 and column 2 on the MS2 row**

Driver Socket Representation: M M

Meaning: **set Jumper between column 1 and column 2 on the RST/SLP row**

SPI Mode

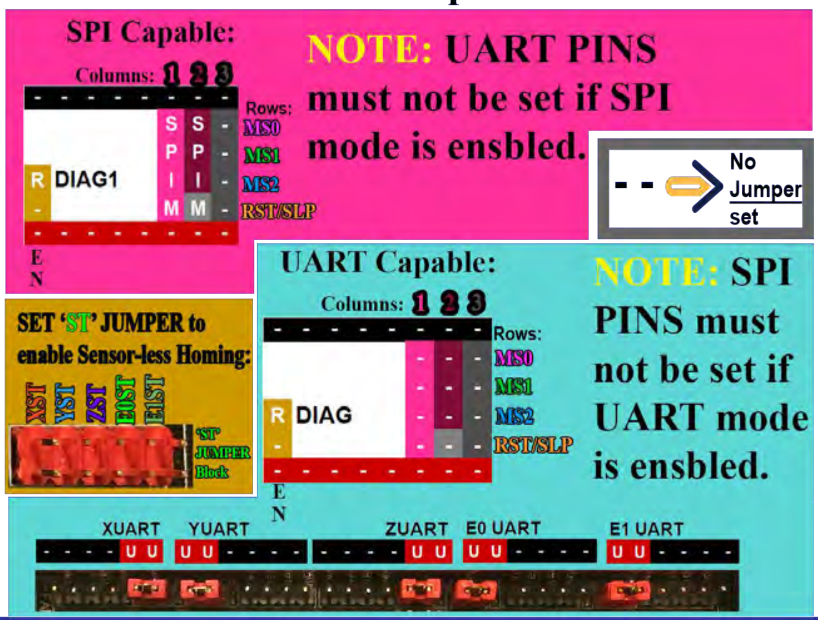
R - Stall Detection

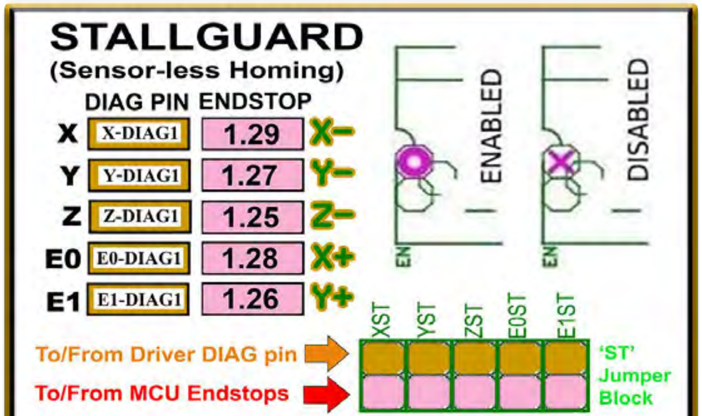
BIQU TMC2130

SPI Mode

Information on Sensor-less Homing

Driver Socket Representation





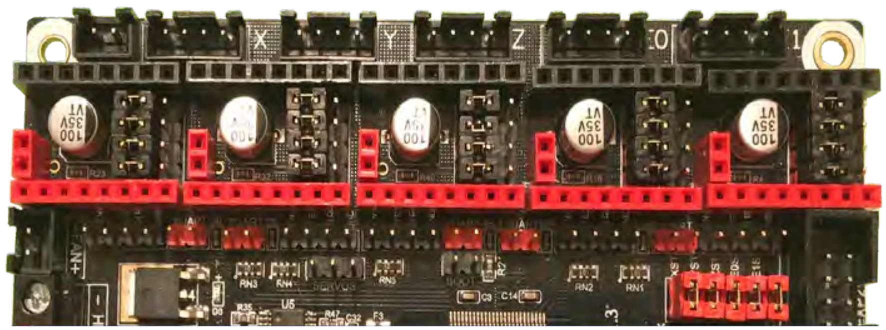
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

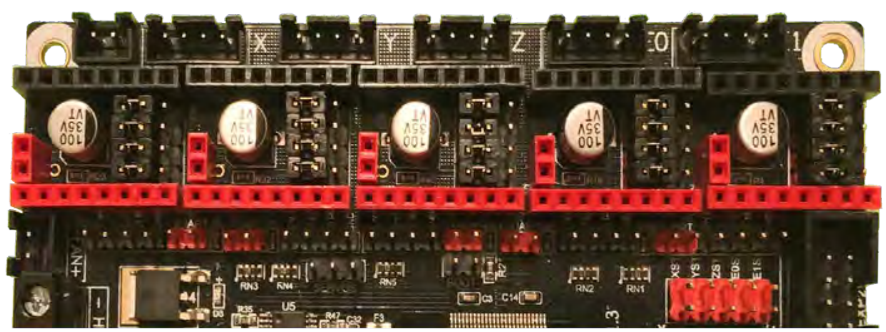
Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161
UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are SET for ALL Axes:



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are REMOVED for ALL Axes:



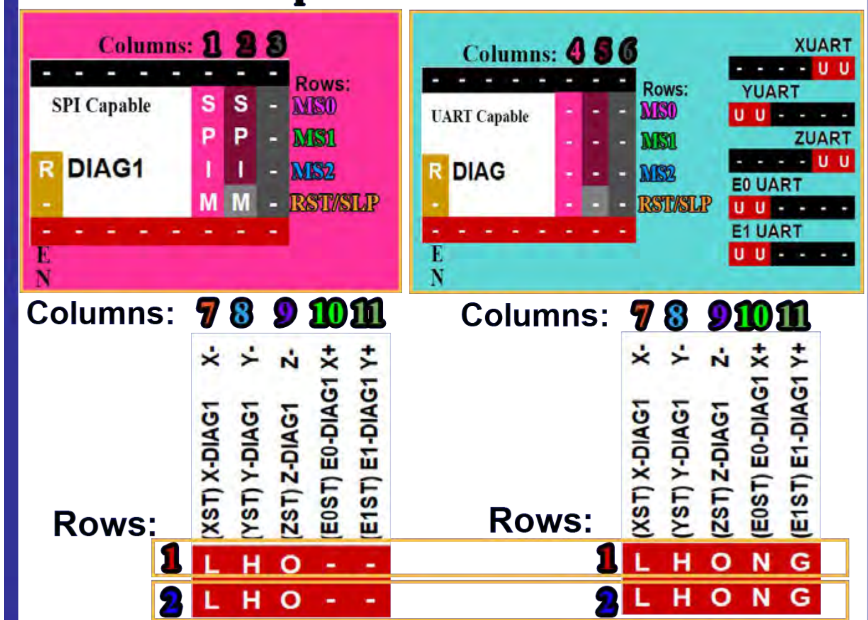
BIQU TMC2130

SPI Mode

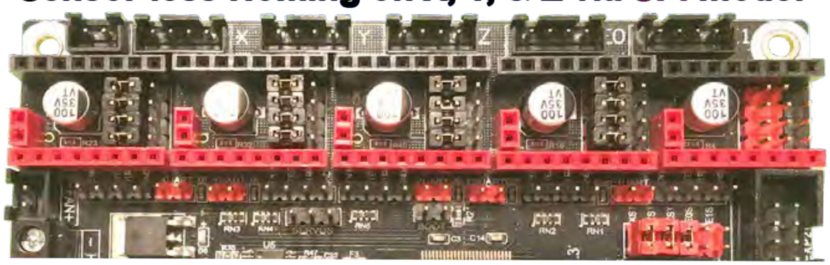
SPI Mode

R - Stall Detection

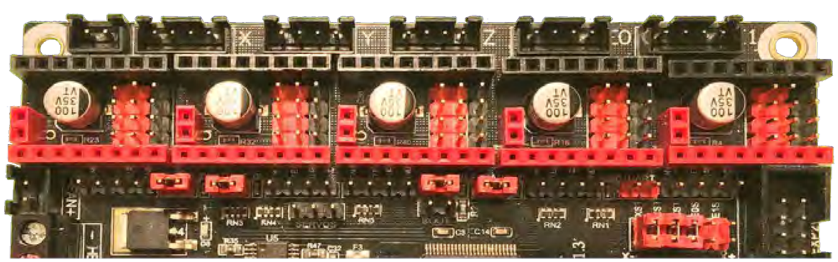
Driver Socket Representation of 'ST' JUMPER Block



Sensor-less Homing on X, Y, & Z via SPI mode:



Sensor-less Homing on X, Y, & Z via UART mode:



Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

- L L → X MIN Endstop to enable Sensor-less Homing
- H H → Y MIN Endstop to enable Sensor-less Homing
- O O → Z MIN Endstop to enable Sensor-less Homing
- N N → X MAX Endstop to enable Sensor-less Homing
- G G → Y MAX Endstop to enable Sensor-less Homing
- \- \- → No Jumper Set

Driver Socket Representation:		Meaning:
	L L	set Jumper between row 1 and row 2 in column 7
	H H	set Jumper between row 1 and row 2 in column 8
	O O	set Jumper between row 1 and row 2 in column 9
	N N	set Jumper between row 1 and row 2 in column 10
	G G	set Jumper between row 1 and row 2 in column 11

BIQU TMC2130

SPI Mode

SPI Mode

R - Stall Detection

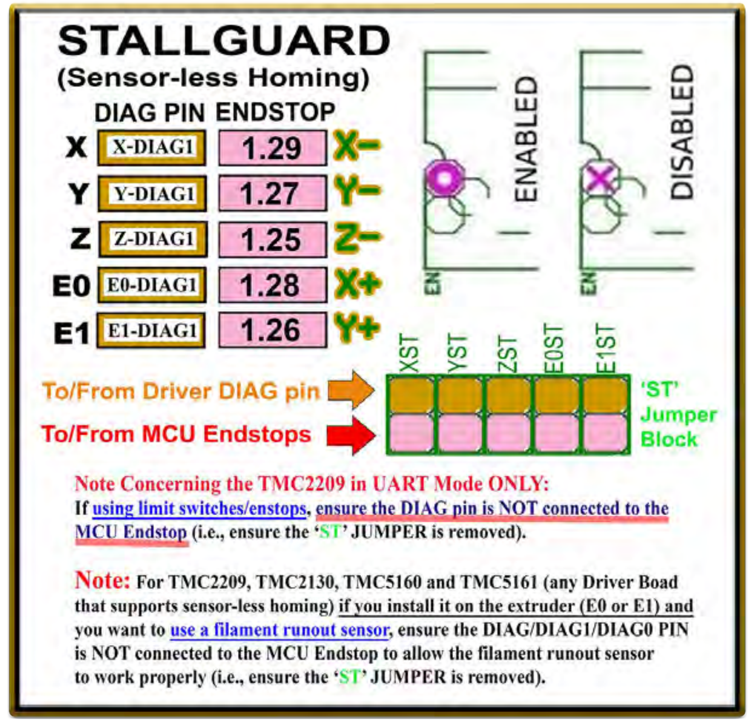
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capabliltiy of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

- **Extruder Axis (E0 or E1)**
- **Z Axis, if a BLTouch is used to Home instead of a physical endstop**

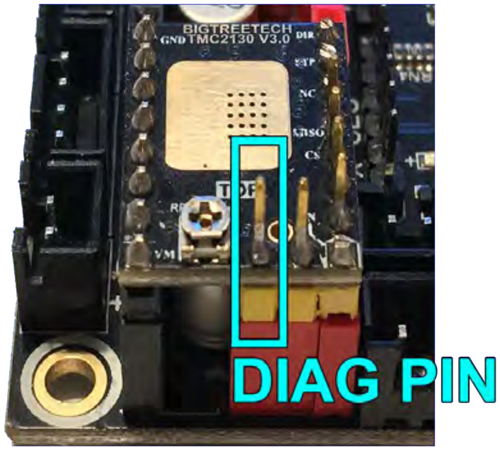
SPI Mode

R - Stall Detection

BIQU TMC2130

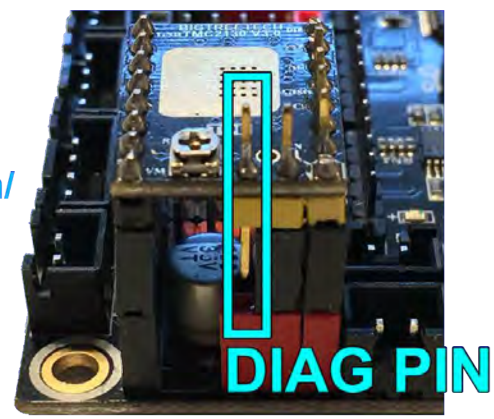
SPI Mode

To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.



To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

Link to stackable header pins:
https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P



SPI Mode

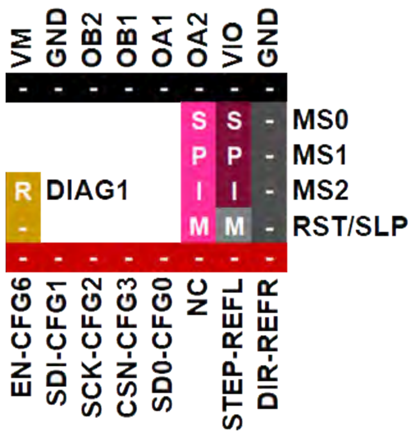
R - Stall Detection

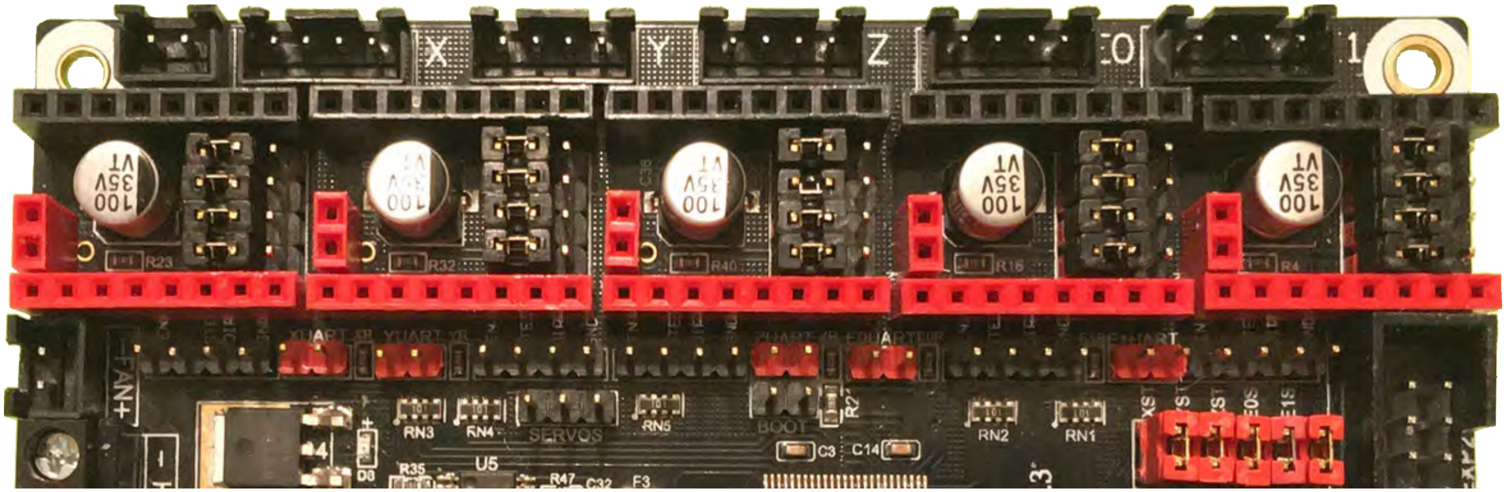
BIQU TMC2130

SPI Mode

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

SPI



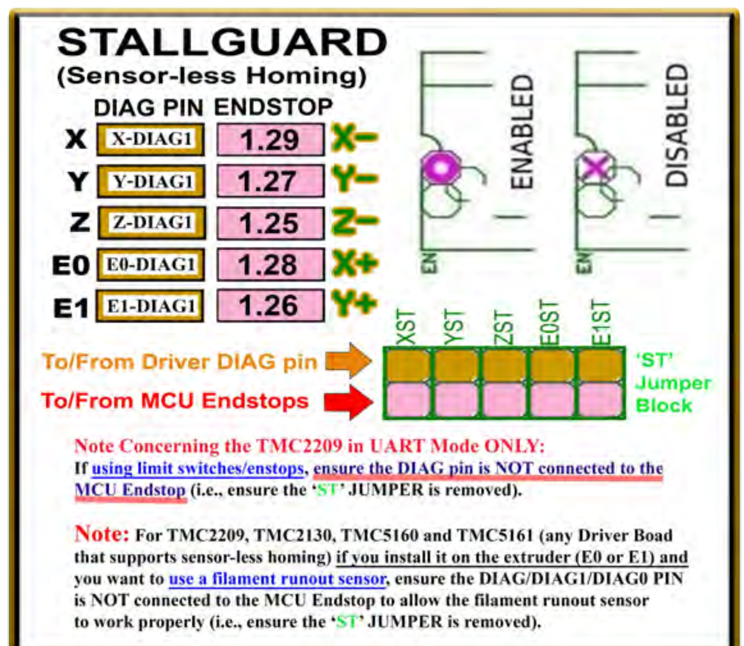


XUART YUART ZUART E0 UART E1 UART

See Appendix D for legend

X-	Y-	Z-	X+	Y+
(XST) X-DIAG1	(YST) Y-DIAG1	(ZST) Z-DIAG1	(E0ST) E0-DIAG1	(E1ST) E1-DIAG1
L	H	O	N	G
L	H	O	N	G

Note: TMC2130 has sensor-less homing capability



SPI Mode

R - Stall Detection

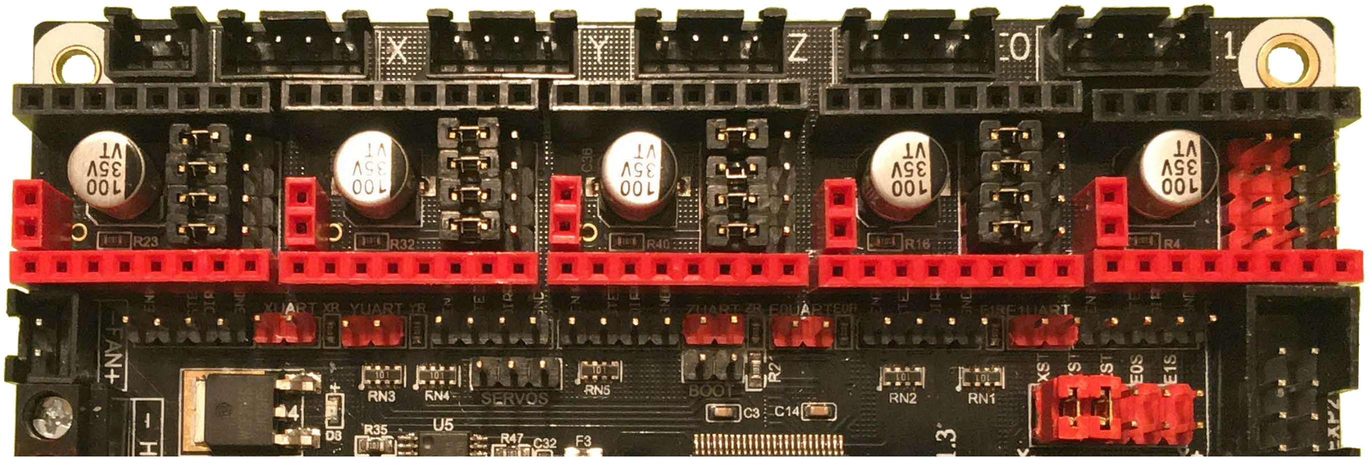
BIQU TMC2130

SPI Mode

Examples of Different SPI Configurations Part 1

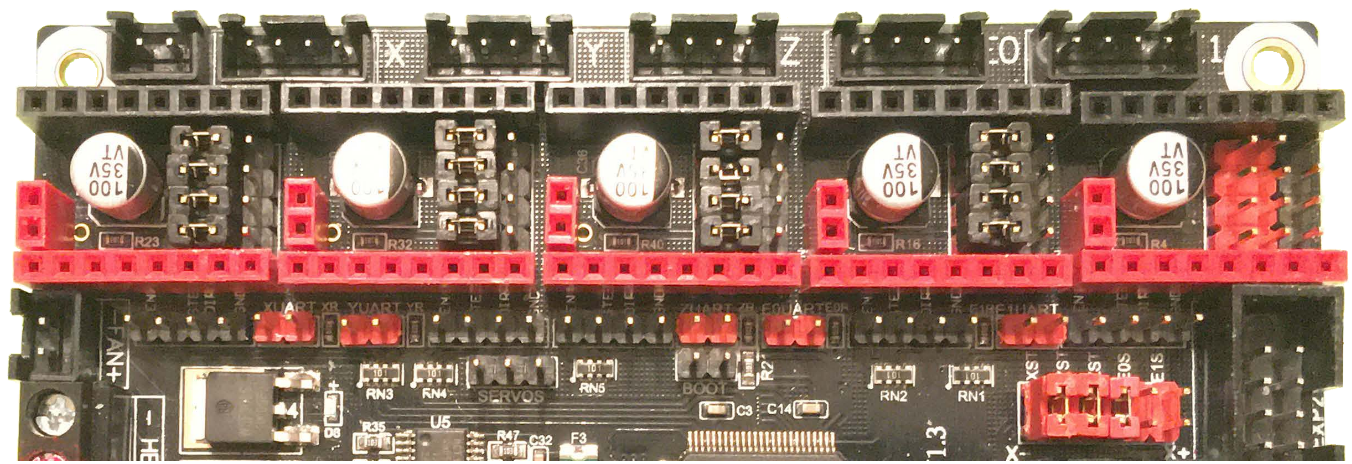
X, Y, Z and E0 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z and E0 axes configured for SPI mode.

Sensor-less homing for X, Y and Z axes.



SPI Mode

R - Stall Detection

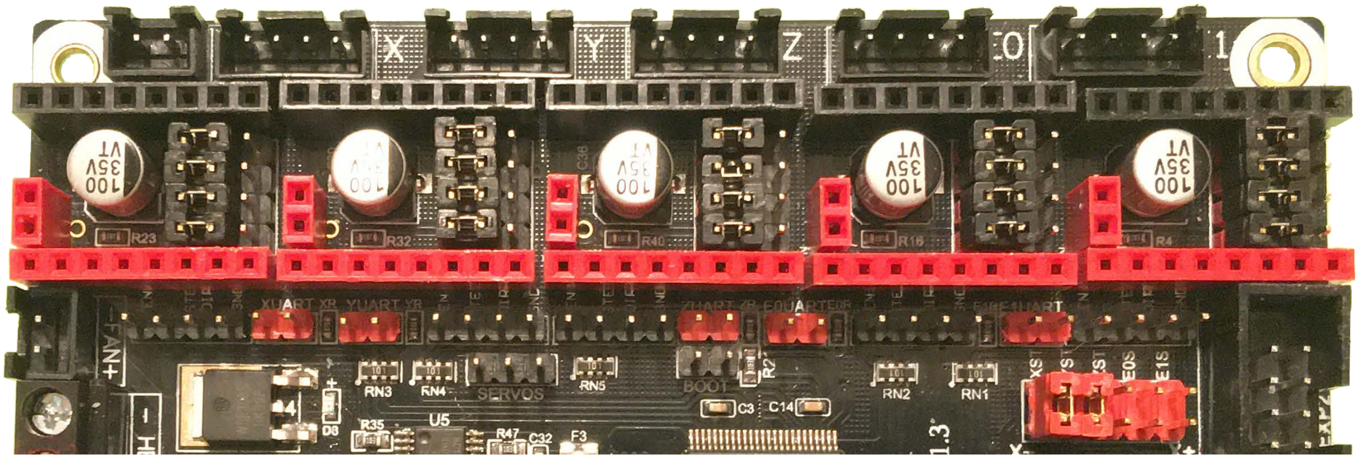
BIQU TMC2130

SPI Mode

Examples of Different SPI Configurations Part 2

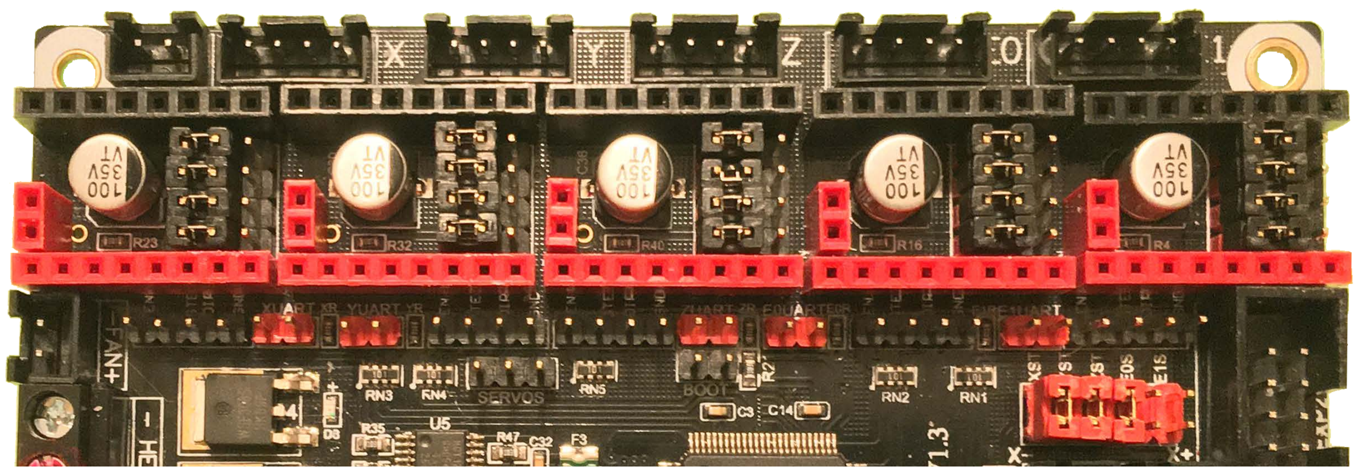
X, Y, Z, E0, and E1 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z, E0 and E1 axes configured for SPI mode.

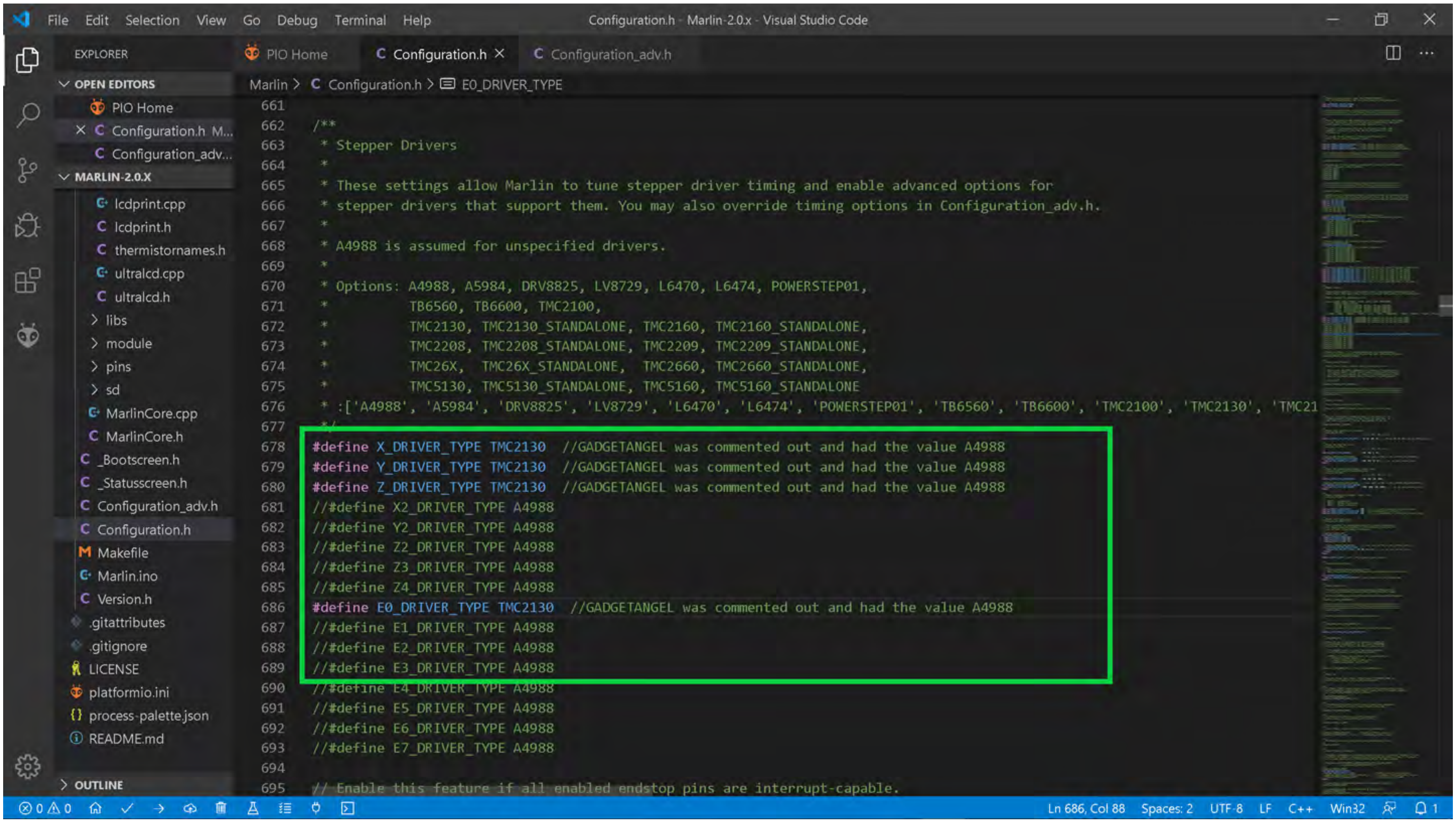
Sensor-less homing for X, Y and Z axes.



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

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

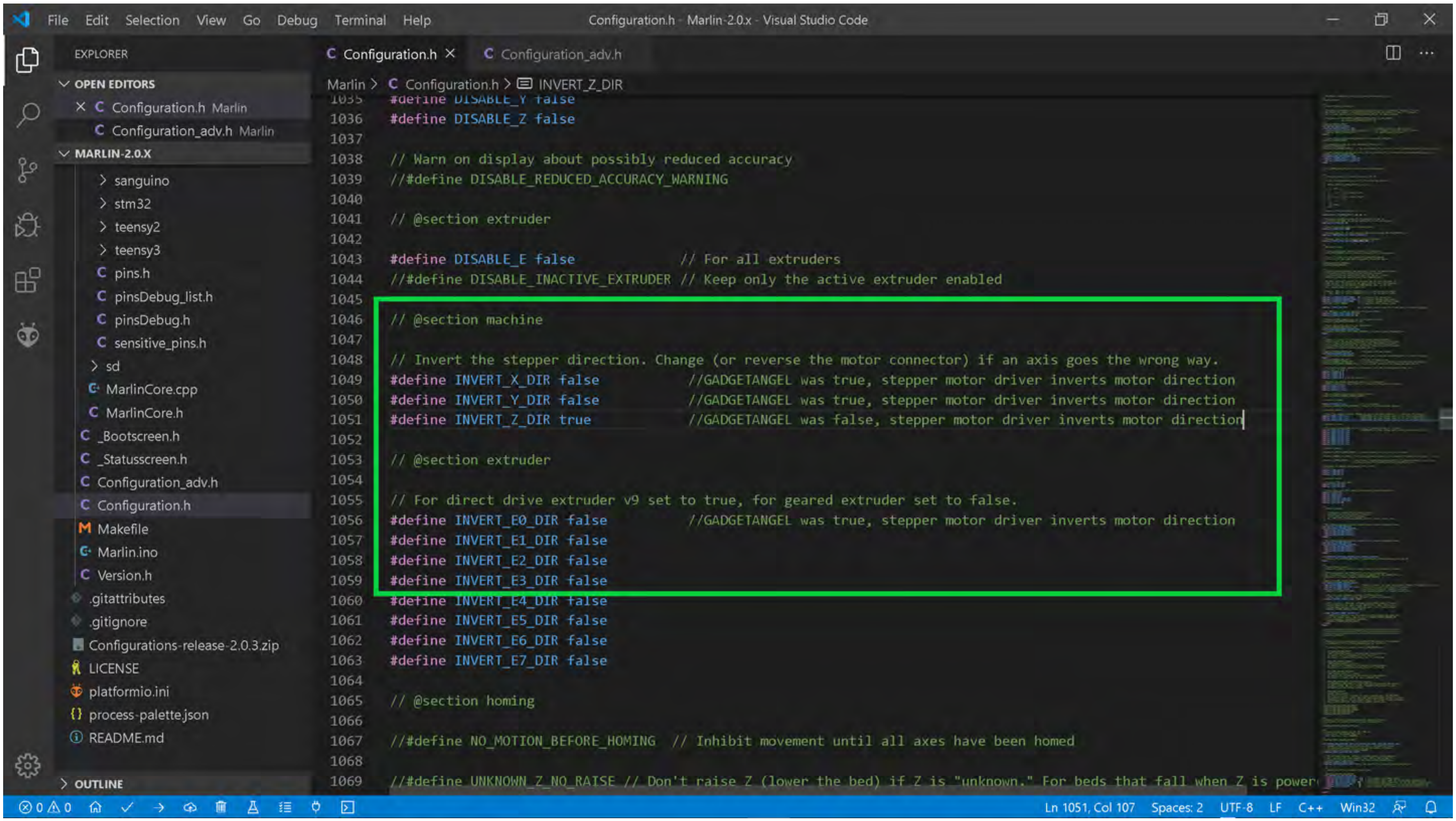
- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2130 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in SPI mode. When two "//" appear at the beginning of a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



- Go to the next page.

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

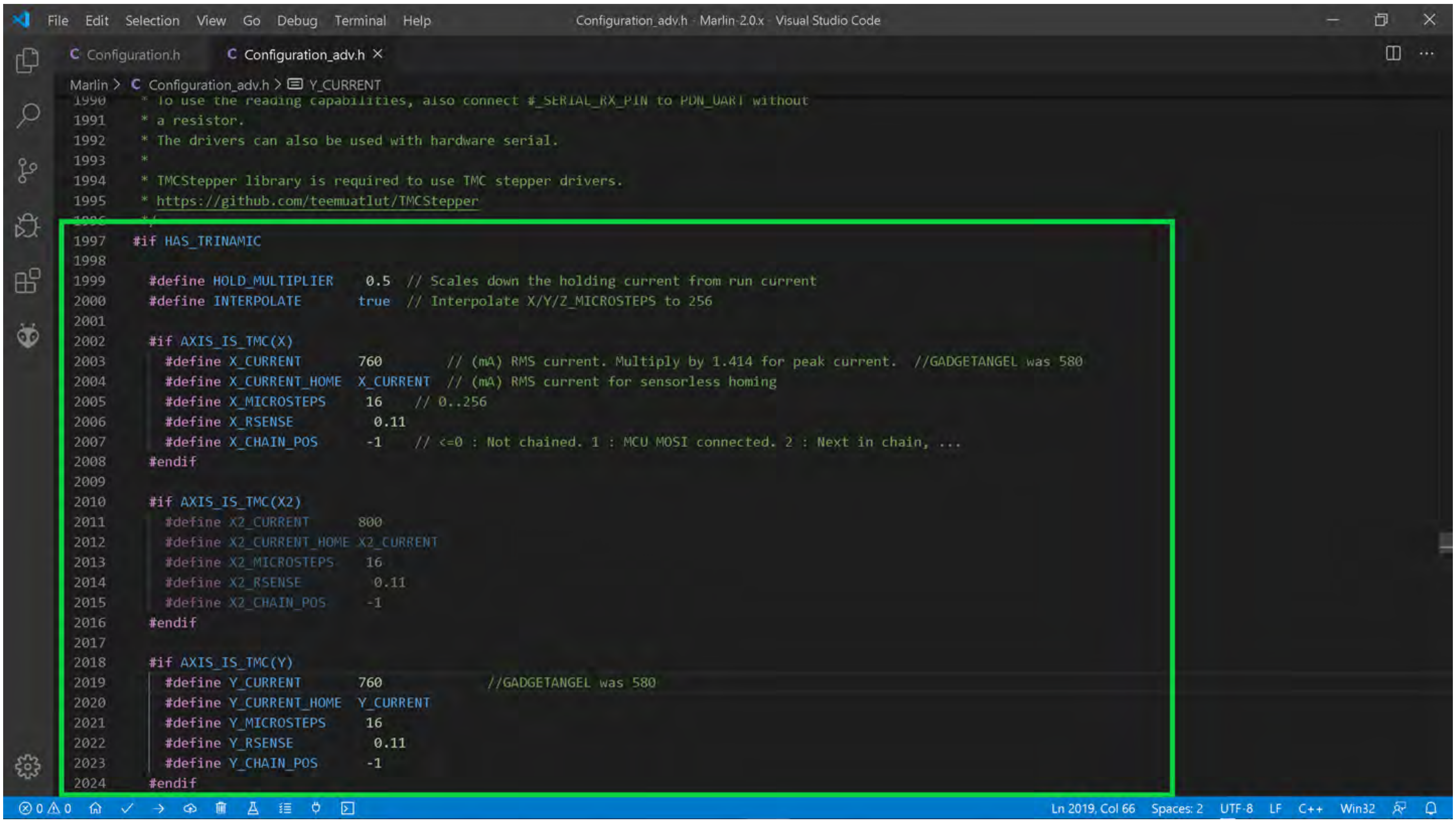
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130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



- Go to the next page.

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

- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2130 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "X_CURRENT" and/or "Y_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT" and/or "Y_CURRENT".

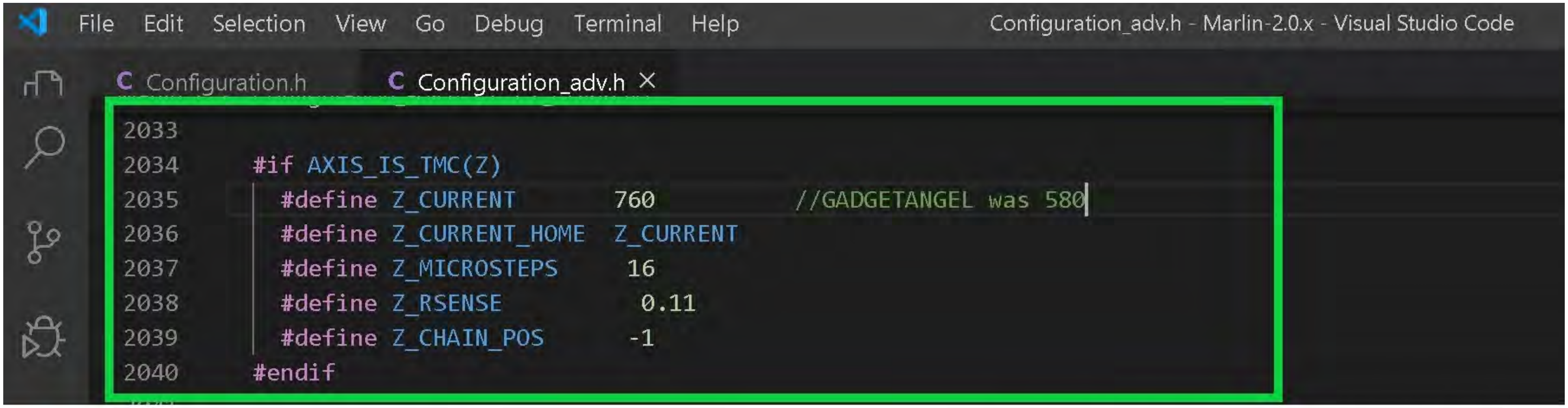


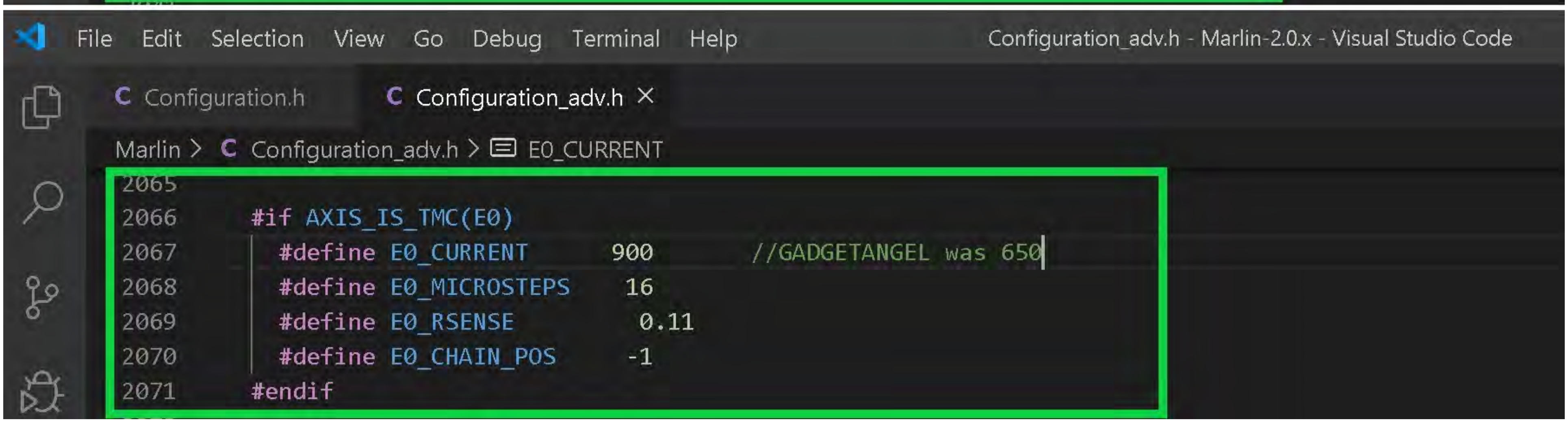
```
 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
 * a resistor.
 * The drivers can also be used with hardware serial.
 *
 * TMCStepper library is required to use TMC stepper drivers.
 * https://github.com/teemuatlut/TMCStepper
 */
#if HAS_TRINAMIC

  #define HOLD_MULTIPLIER    0.5  // Scales down the holding current from run current
  #define INTERPOLATE       true  // Interpolate X/Y/Z_MICROSTEPS to 256

  #if AXIS_IS_TMC(X)
    #define X_CURRENT       760        // (mA) RMS current. Multiply by 1.414 for peak current.  //GADGETANGEL was 580
    #define X_CURRENT_HOME  X_CURRENT  // (mA) RMS current for sensorless homing
    #define X_MICROSTEPS     16    // 0..256
    #define X_RSENSE          0.11
    #define X_CHAIN_POS      -1    // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
  #endif

  #if AXIS_IS_TMC(X2)
    #define X2_CURRENT      800
    #define X2_CURRENT_HOME X2_CURRENT
    #define X2_MICROSTEPS    16
    #define X2_RSENSE         0.11
    #define X2_CHAIN_POS     -1
  #endif

  #if AXIS_IS_TMC(Y)
    #define Y_CURRENT       760          //GADGETANGEL was 580
    #define Y_CURRENT_HOME  Y_CURRENT
    #define Y_MICROSTEPS     16
    #define Y_RSENSE          0.11
    #define Y_CHAIN_POS      -1
  #endif
```

- Go to the next page.

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

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

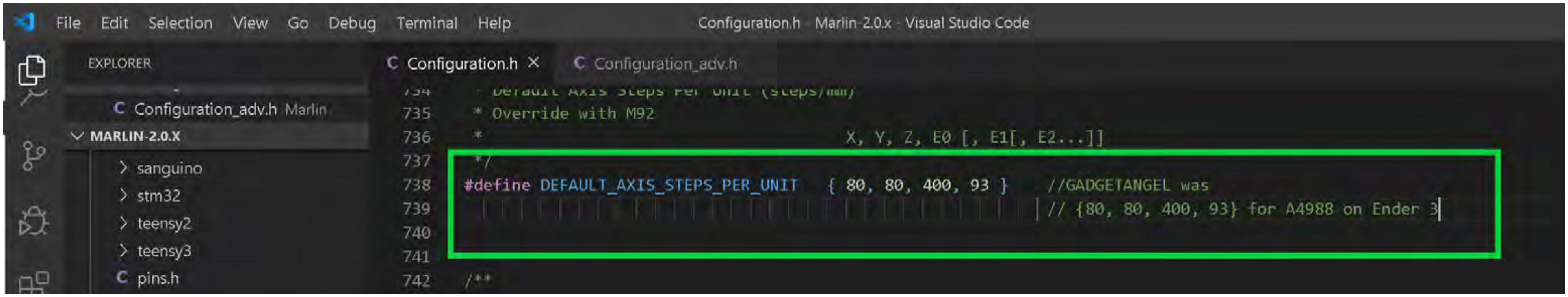




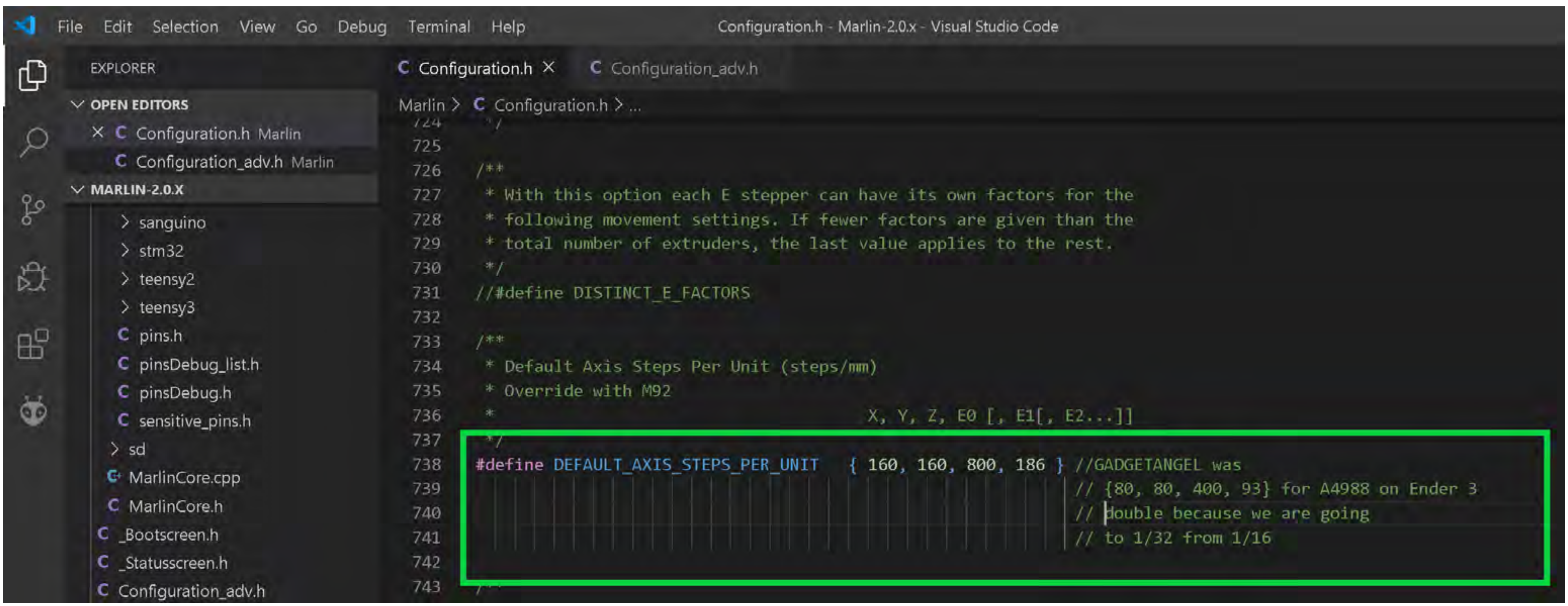
- Go to the next page.

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

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



- FOR EAMPLE 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.

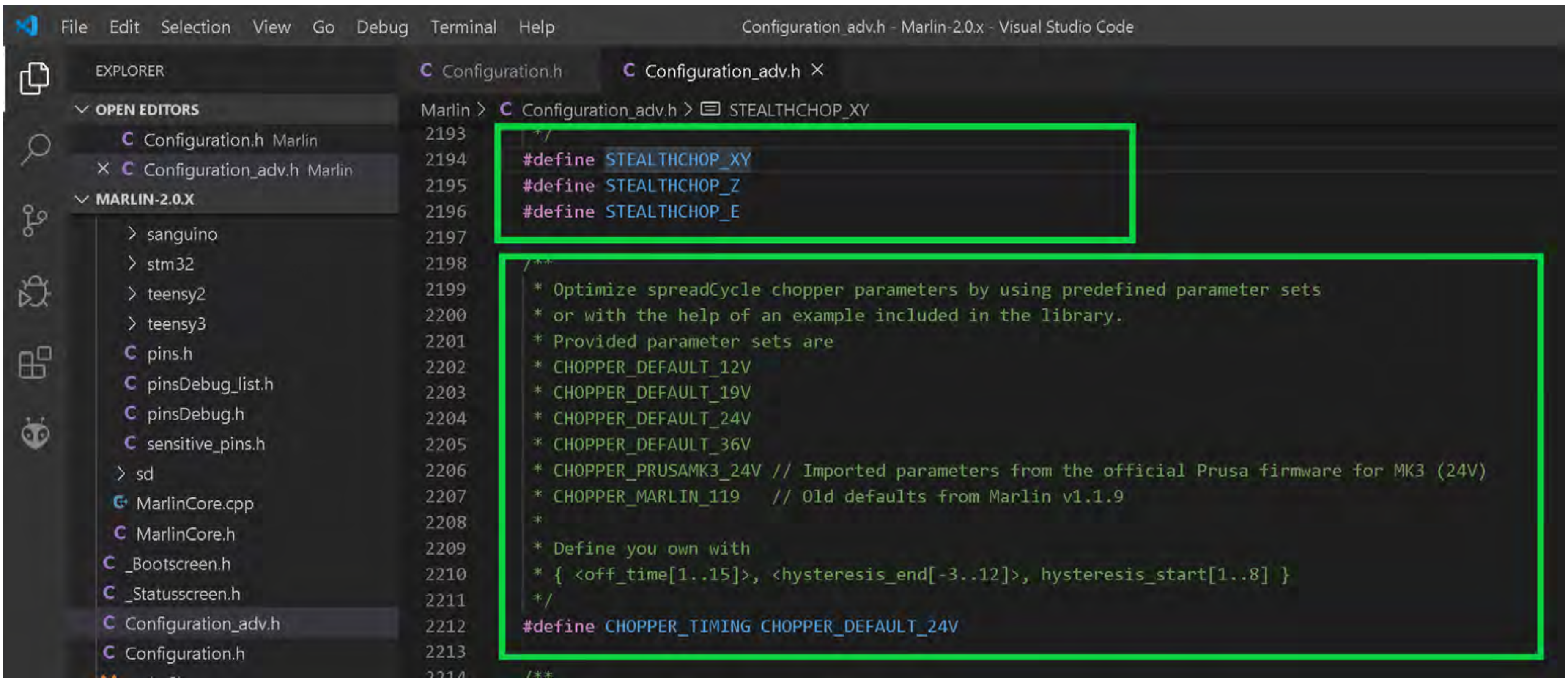


- Go to the next page.

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

- We need to uncomment out the "TMC_USE_SW_SPI" because the SKR V1.3 pins file depends on this variable to define its SPI pins

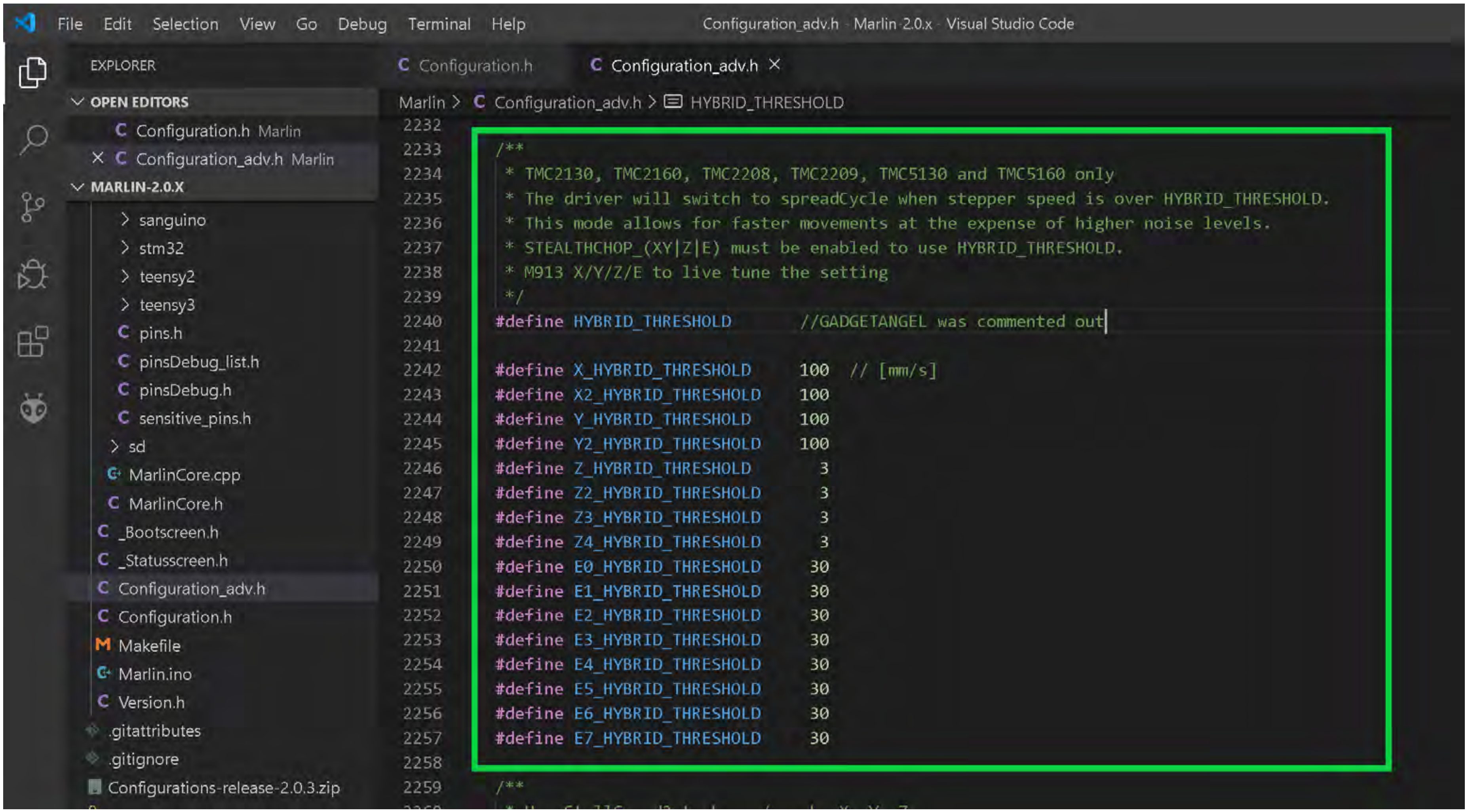
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

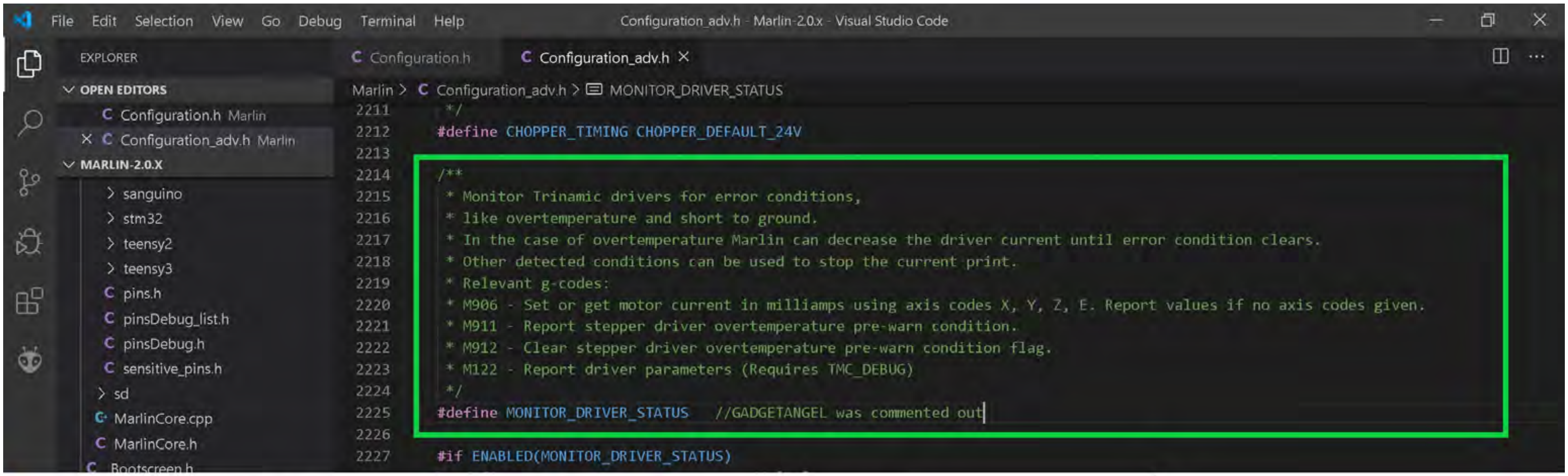
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

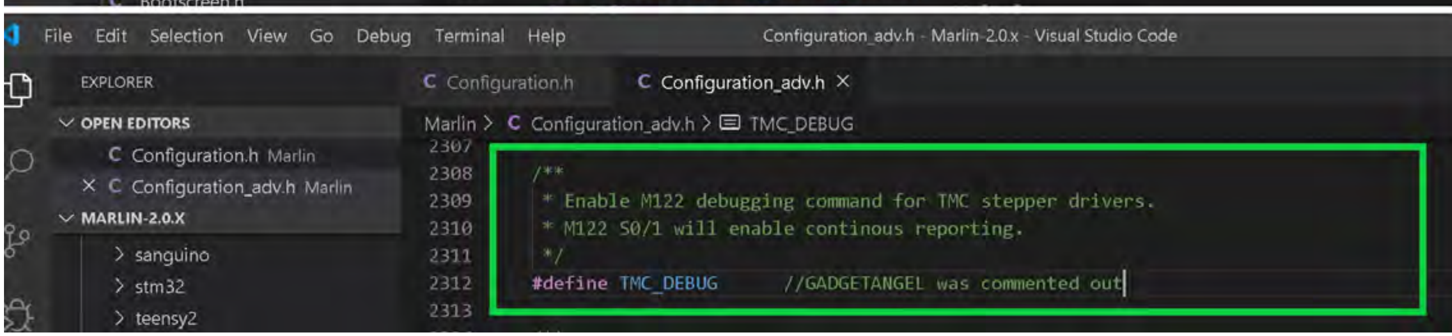


- Go to the next page.

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

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

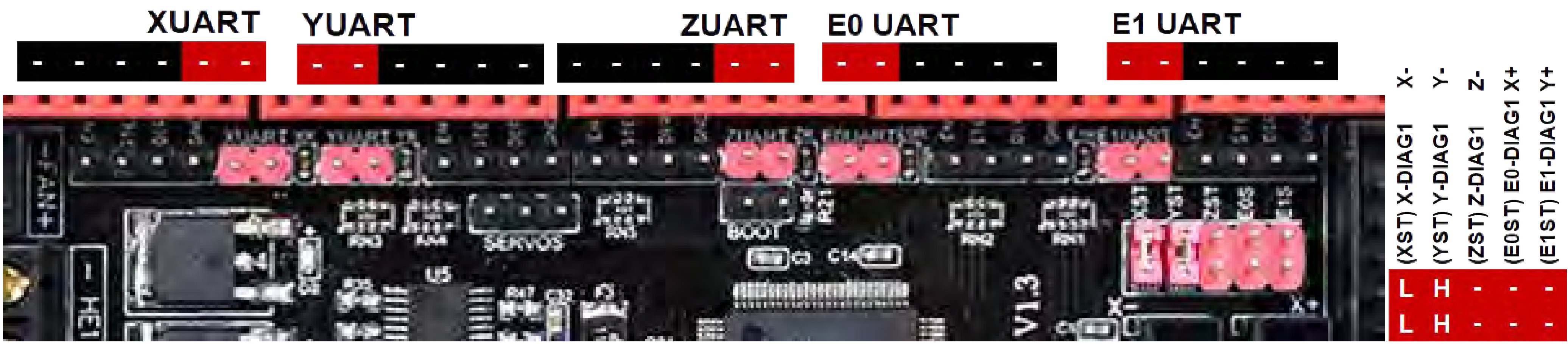




- Go to the next page.

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

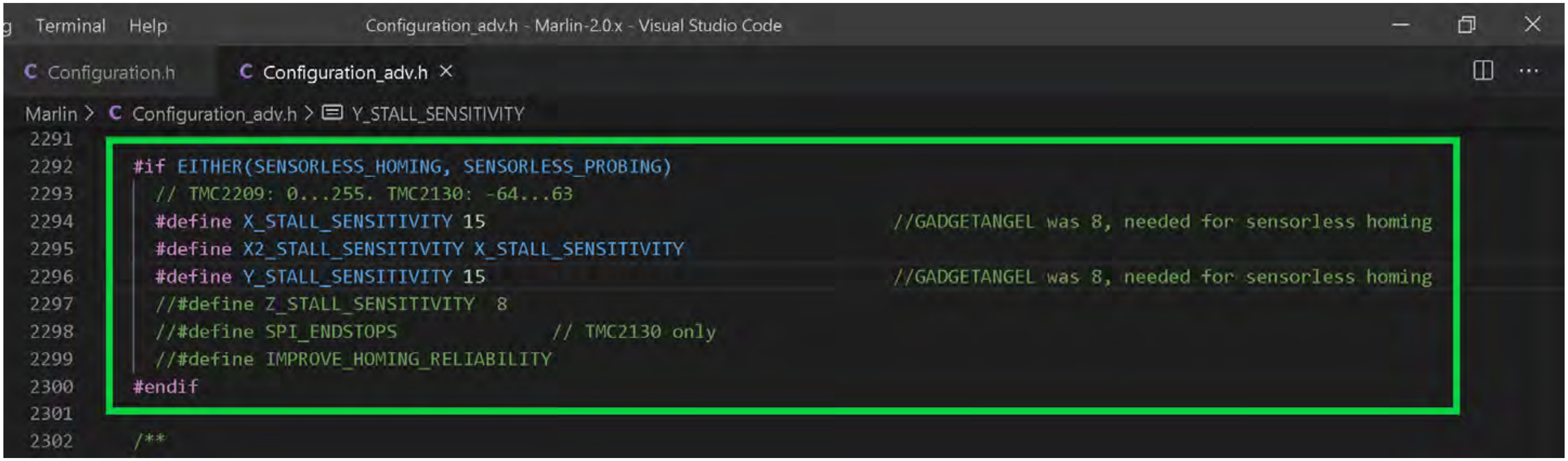
- This next section covers sensor-less homing which is available for the TMC2130 in SPI mode. I want to enable it so I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC2130 driver has to be plugged into the SKR V1.3 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC2130 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC2130 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC2130 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers are in place: the "L" and "H" jumpers.



- Go to the next page.

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

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



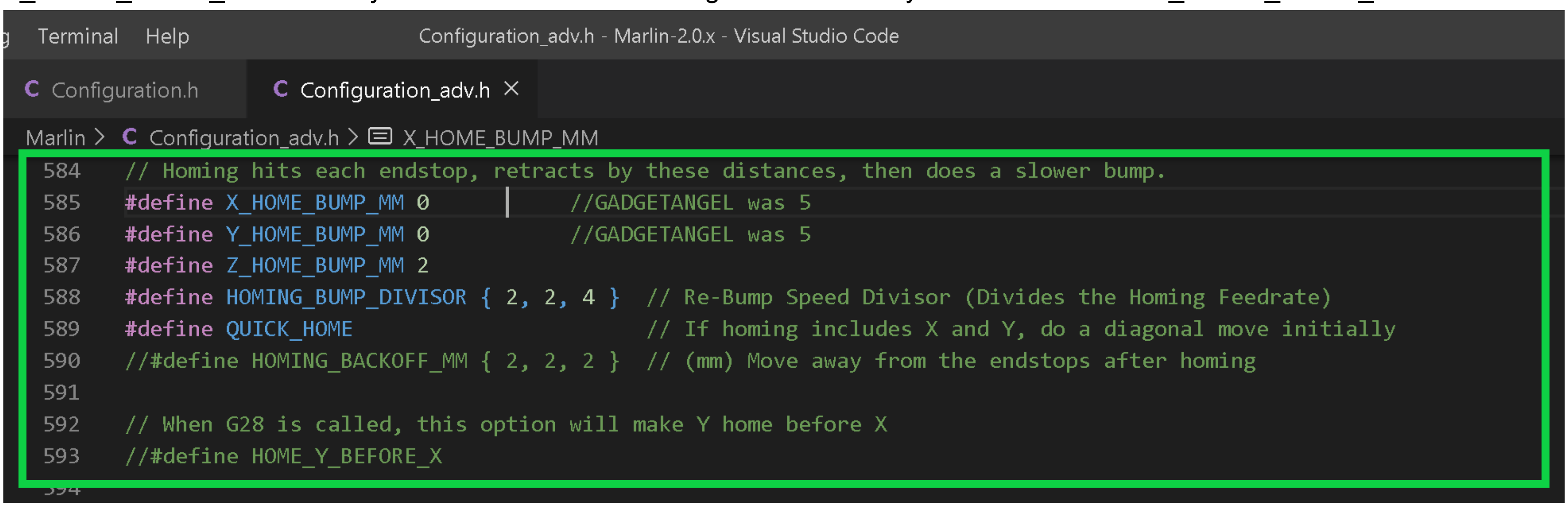
Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

Configuration.h Configuration_adv.h

Marlin > Configuration_adv.h > Y_STALL_SENSITIVITY

```
#if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
  // TMC2209: 0...255. TMC2130: -64...63
  #define X_STALL_SENSITIVITY 15                                    //GADGETANGEL was 8, needed for sensorless homing
  #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
  #define Y_STALL_SENSITIVITY 15                                    //GADGETANGEL was 8, needed for sensorless homing
  //#define Z_STALL_SENSITIVITY  8
  //#define SPI_ENDSTOPS              // TMC2130 only
  //#define IMPROVE_HOMING_RELIABILITY
#endif

/**
```

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



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

Configuration.h Configuration_adv.h

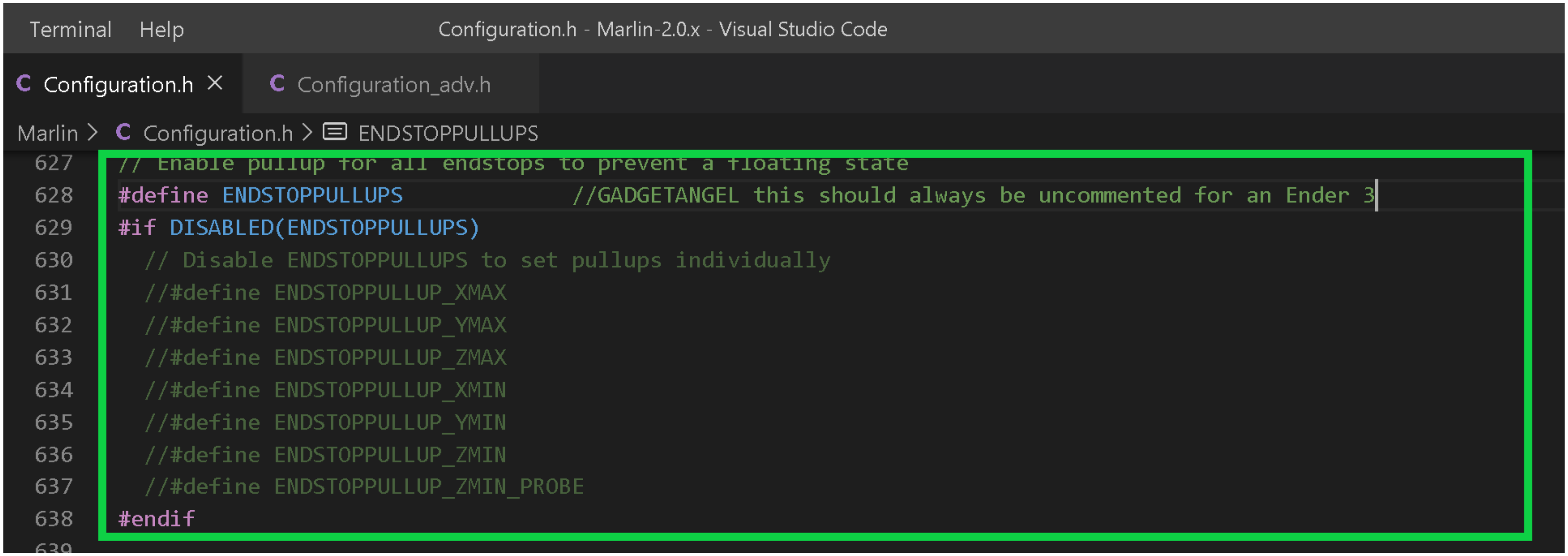
Marlin > Configuration_adv.h > X_HOME_BUMP_MM

```
// Homing hits each endstop, retracts by these distances, then does a slower bump.
#define X_HOME_BUMP_MM 0          //GADGETANGEL was 5
#define Y_HOME_BUMP_MM 0          //GADGETANGEL was 5
#define Z_HOME_BUMP_MM 2
#define HOMING_BUMP_DIVISOR { 2, 2, 4 }  // Re-Bump Speed Divisor (Divides the Homing Feedrate)
#define QUICK_HOME                       // If homing includes X and Y, do a diagonal move initially
//#define HOMING_BACKOFF_MM { 2, 2, 2 }  // (mm) Move away from the endstops after homing

// When G28 is called, this option will make Y home before X
//#define HOME_Y_BEFORE_X
```

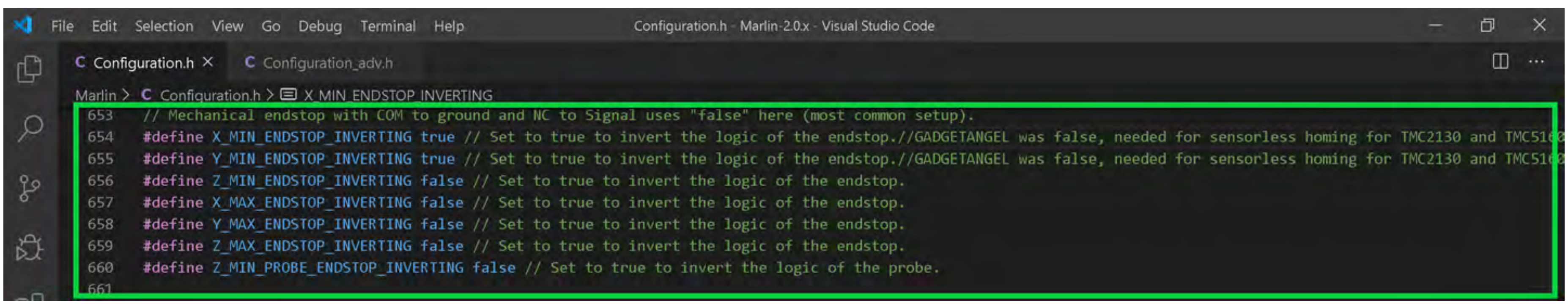
- Go to the next page.

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

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



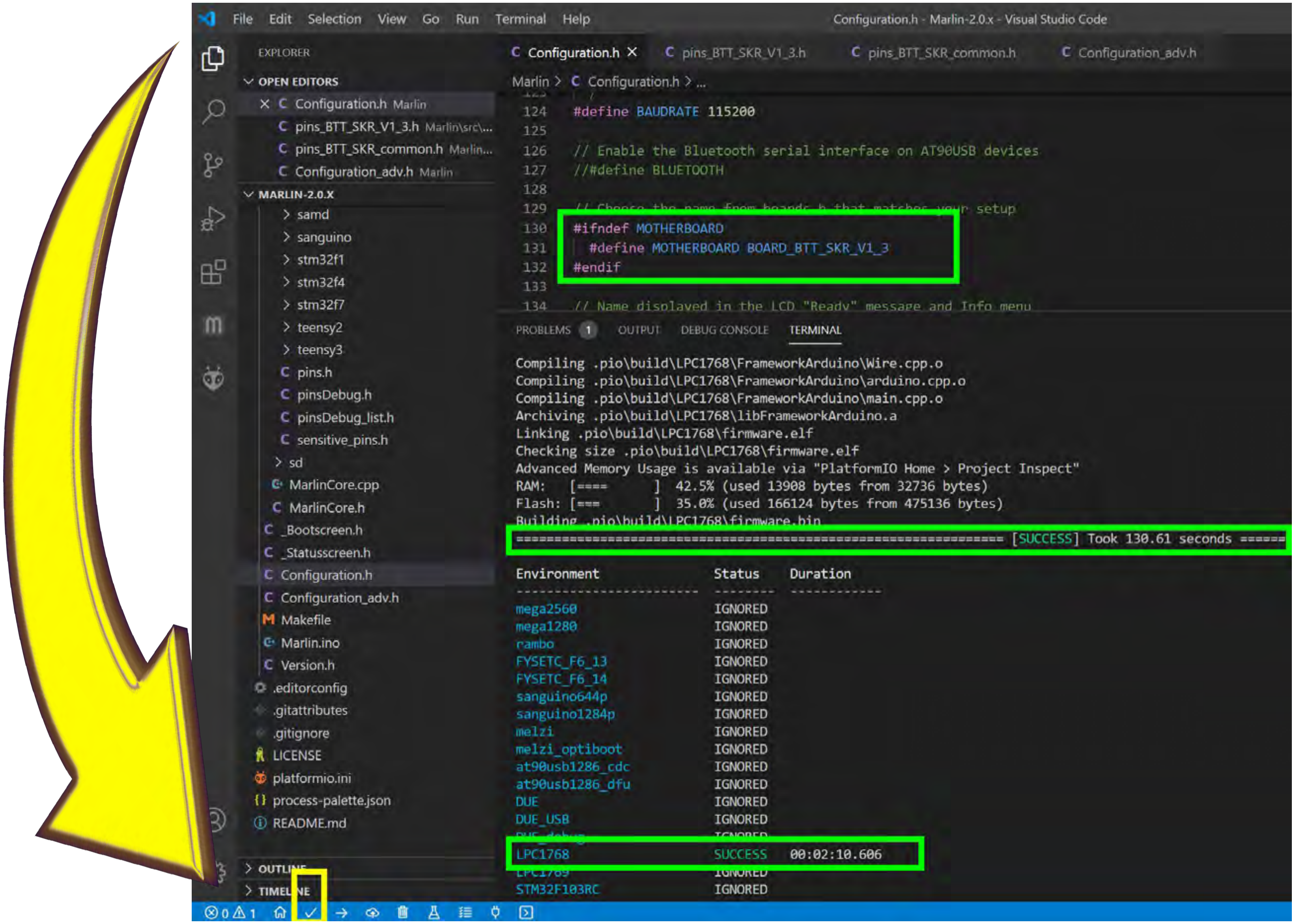
- Next to allow sensor-less homing to work (while using the BIQU TMC2130) we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z_MIN_ENDSTOP_INVERTING" set to false.



- Go to the next page.

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

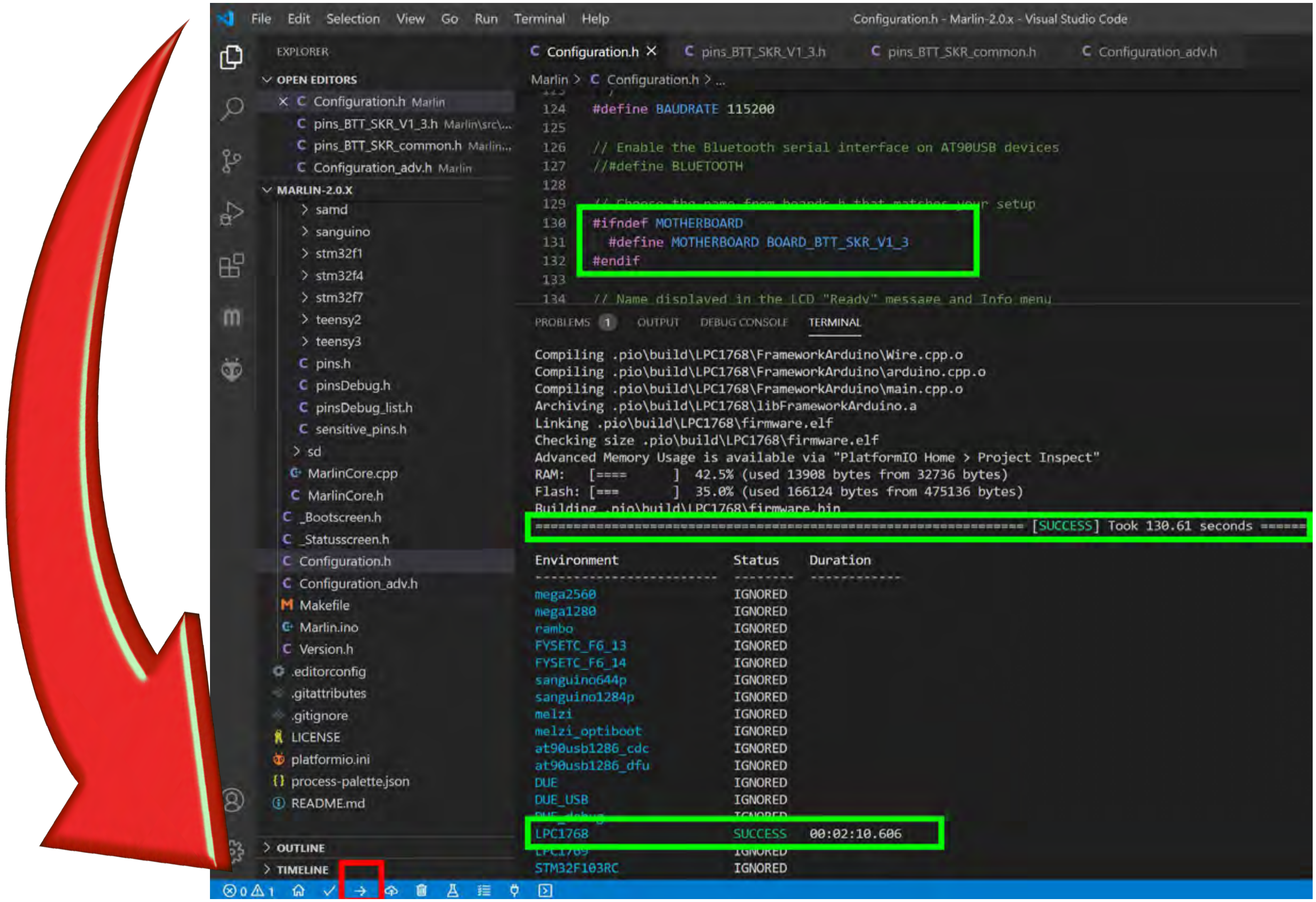
- The end of Marlin setup for BIQU TMC2130 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



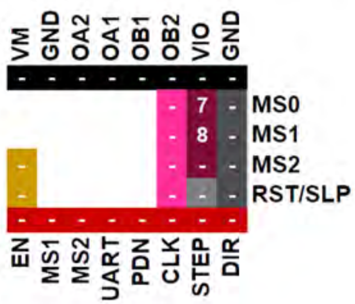
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2208 V3.0

Stand-alone Mode

Stand-alone Mode

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

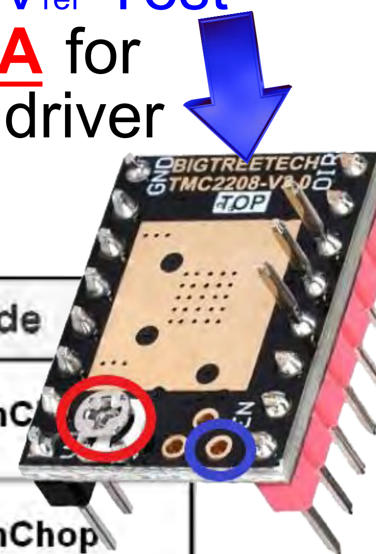
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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

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



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

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

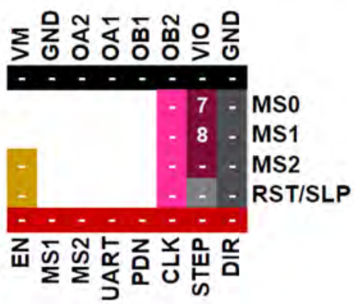
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU TMC2208 V3.0

Stand-alone Mode

Stand-alone Mode

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC xxA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)=x.xx Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

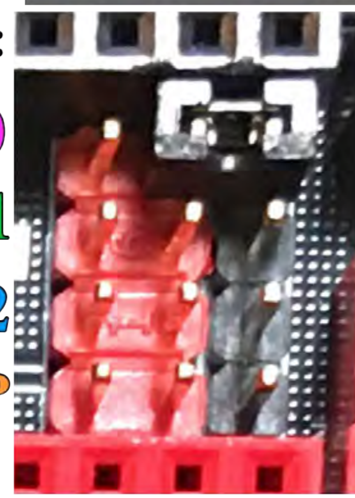
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low → **No Jumper**

Rows:
MS0
MS1
MS2
RST/SLP

High → **Jumper Set**

Rows:
MS0
MS1
MS2
RST/SLP



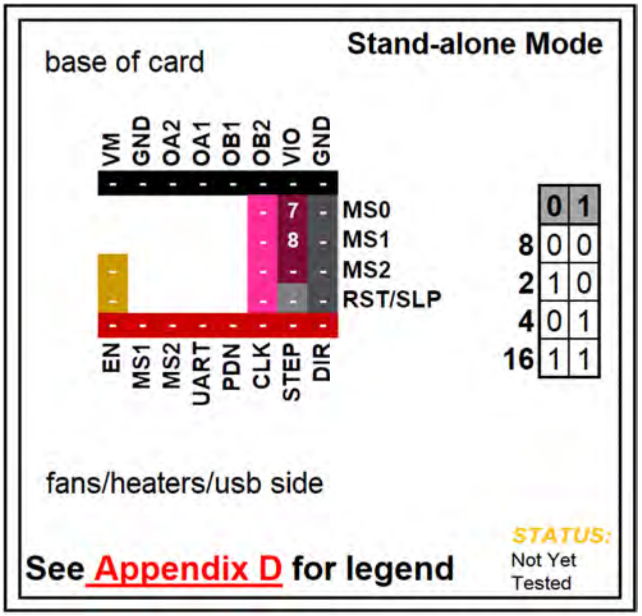
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

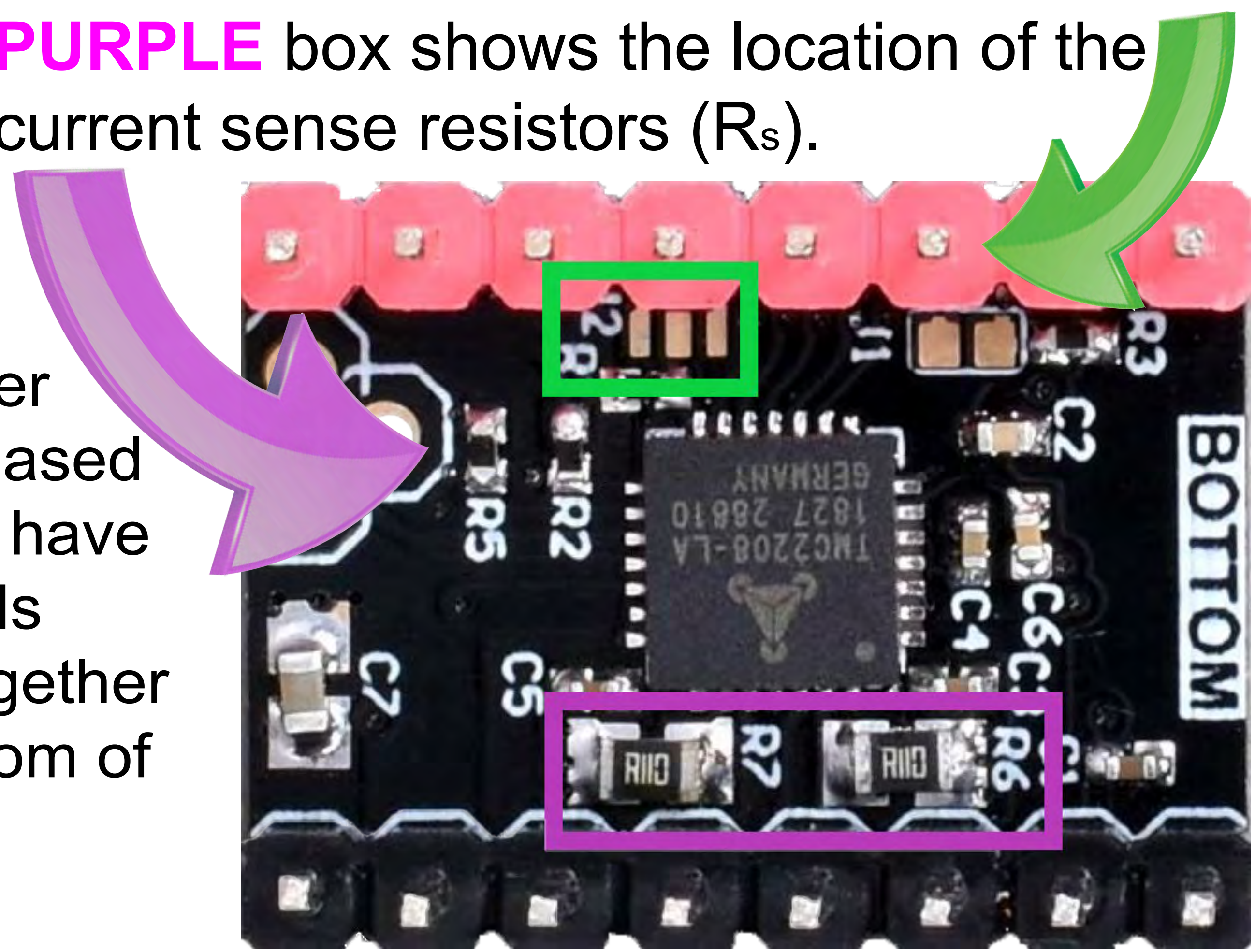
BIQU TMC2208 V3.0

Stand-alone Mode



Note: The three pads (J2), on the bottom of the BIQU TMC2208 V3.0 driver boards, **MUST NOT** be connected. Again, a gap MUST be in place between all three J2 pads to obtain stand-alone mode for the TMC2208 V3.0, as seen in the **GREEN** box below. The **PURPLE** box shows the location of the current sense resistors (R_s).

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



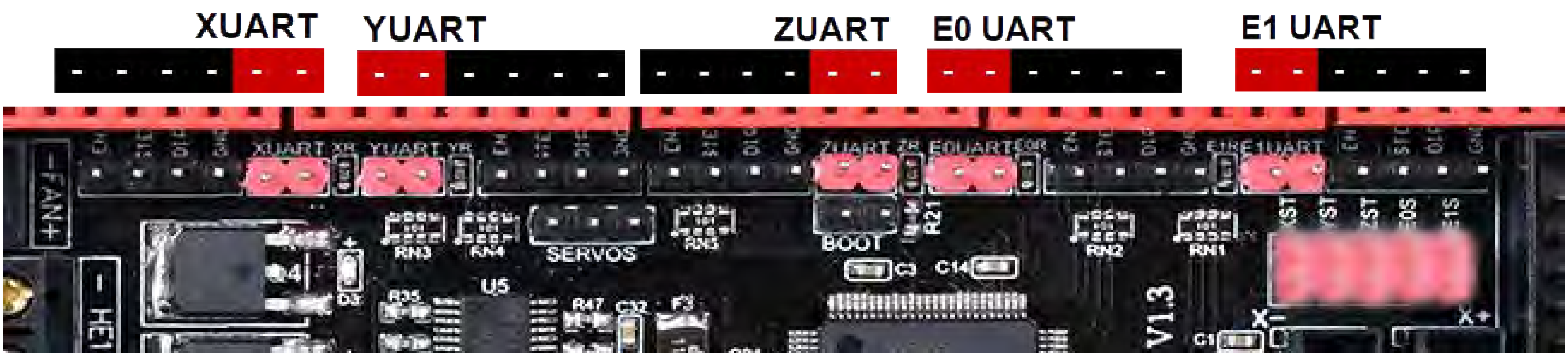
Stand-alone Mode

BIQU TMC2208 V3.0

Stand-alone Mode

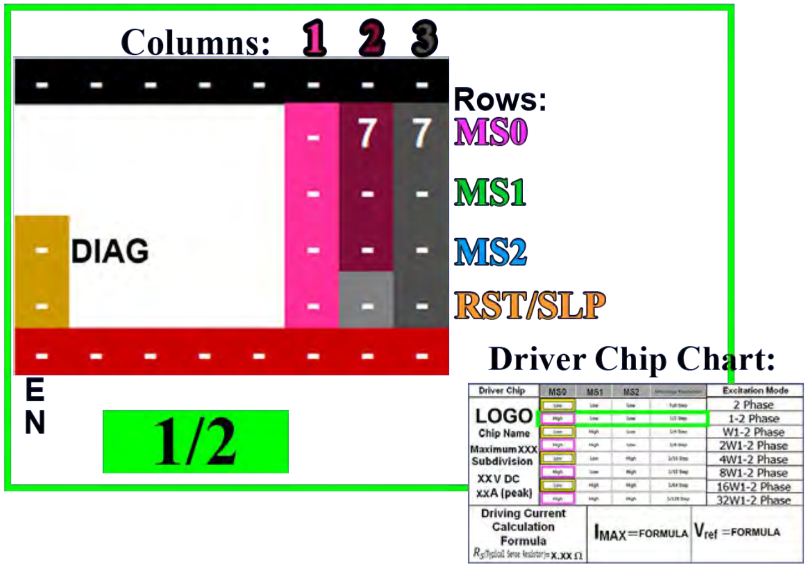
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

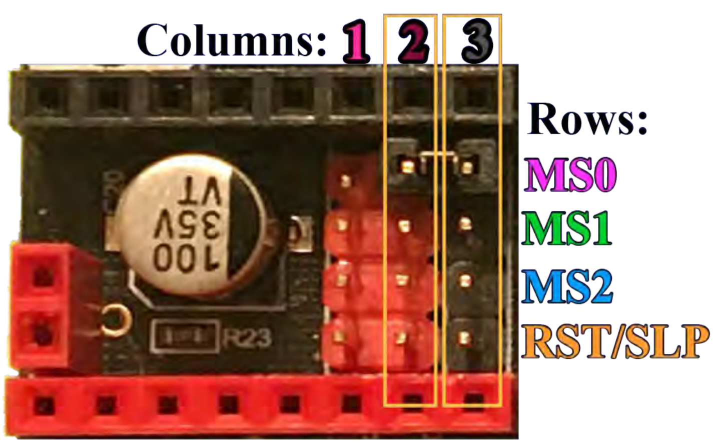


SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

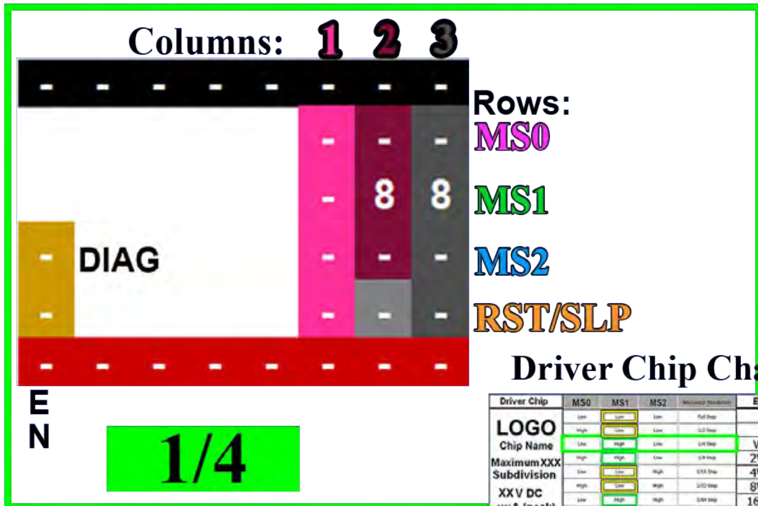
Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

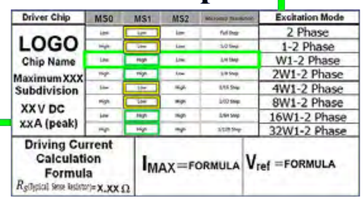
Driver Chip Chart:	Meaning:
High	set Jumper between column 2 and column 3 on the MS0 row
Low	No Jumper Set

Stand-alone Mode

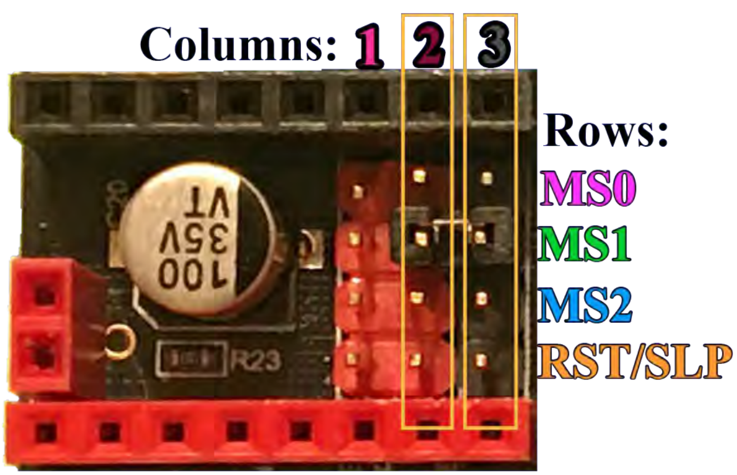
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Driver Chip Chart:		Meaning:
High	→	set Jumper between column 2 and column 3 on the MS1 row
Low	→	No Jumper Set

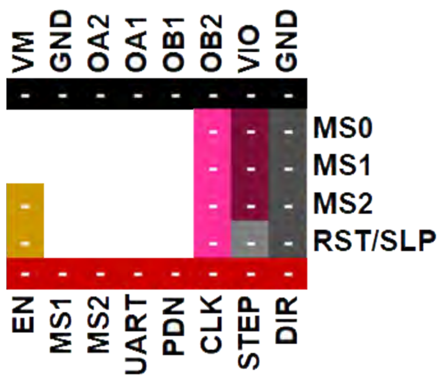
Stand-alone Mode

Stand-alone Mode

1 / 8

Interpolation:
1 / 256

StealthChop



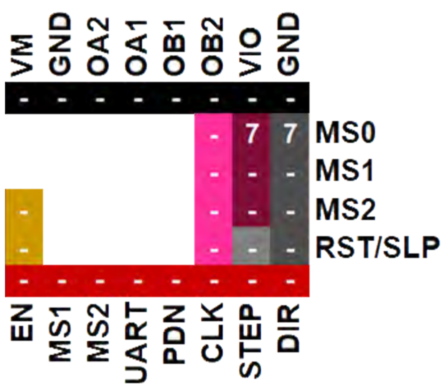
See Appendix D for legend

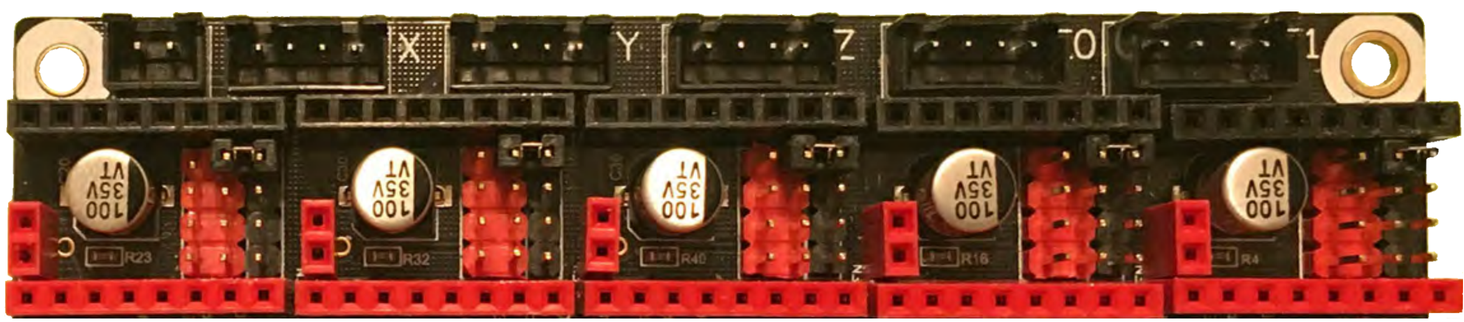
Stand-alone Mode

1 / 2

Interpolation:
1 / 256

StealthChop





See Appendix D for legend

Stand-alone Mode

BIQU TMC2208 V3.0

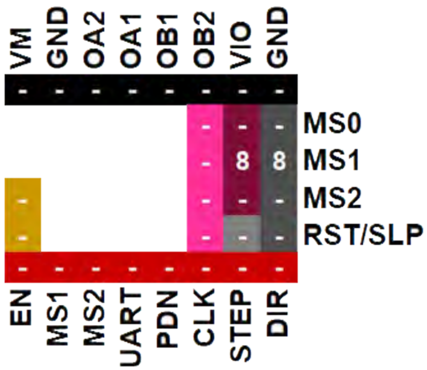
Stand-alone Mode

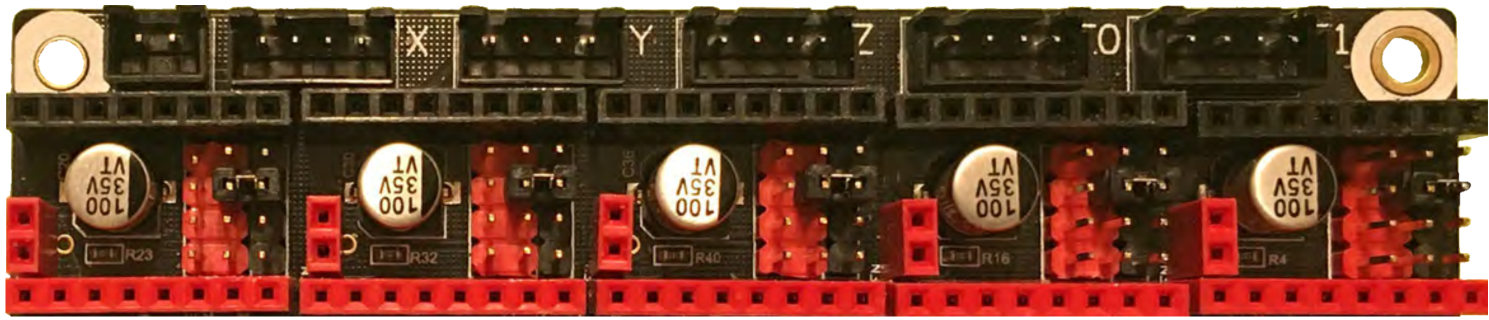
Stand-alone Mode

1 / 4

Interpolation:
1 / 256

StealthChop





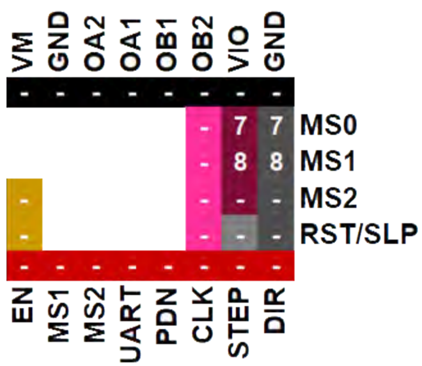
See Appendix D for legend

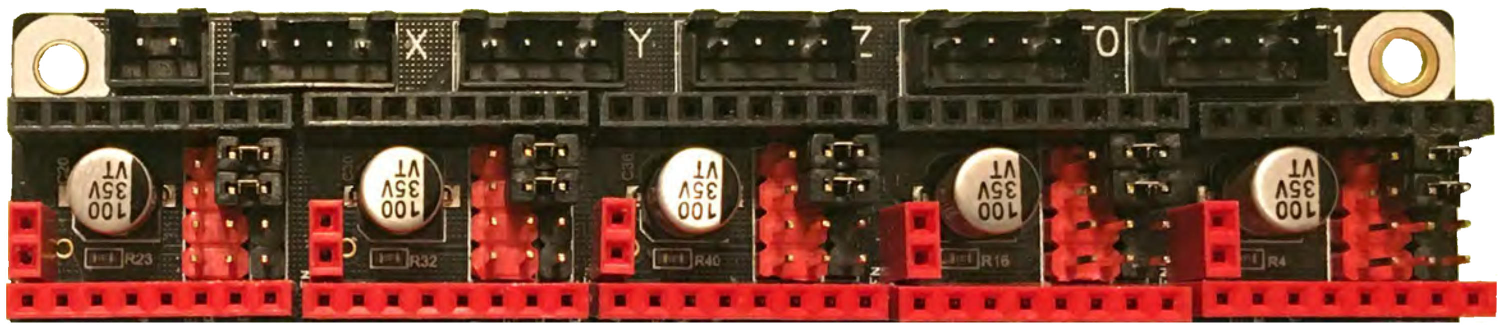
Stand-alone Mode

1 / 16

Interpolation:
1 / 256

StealthChop



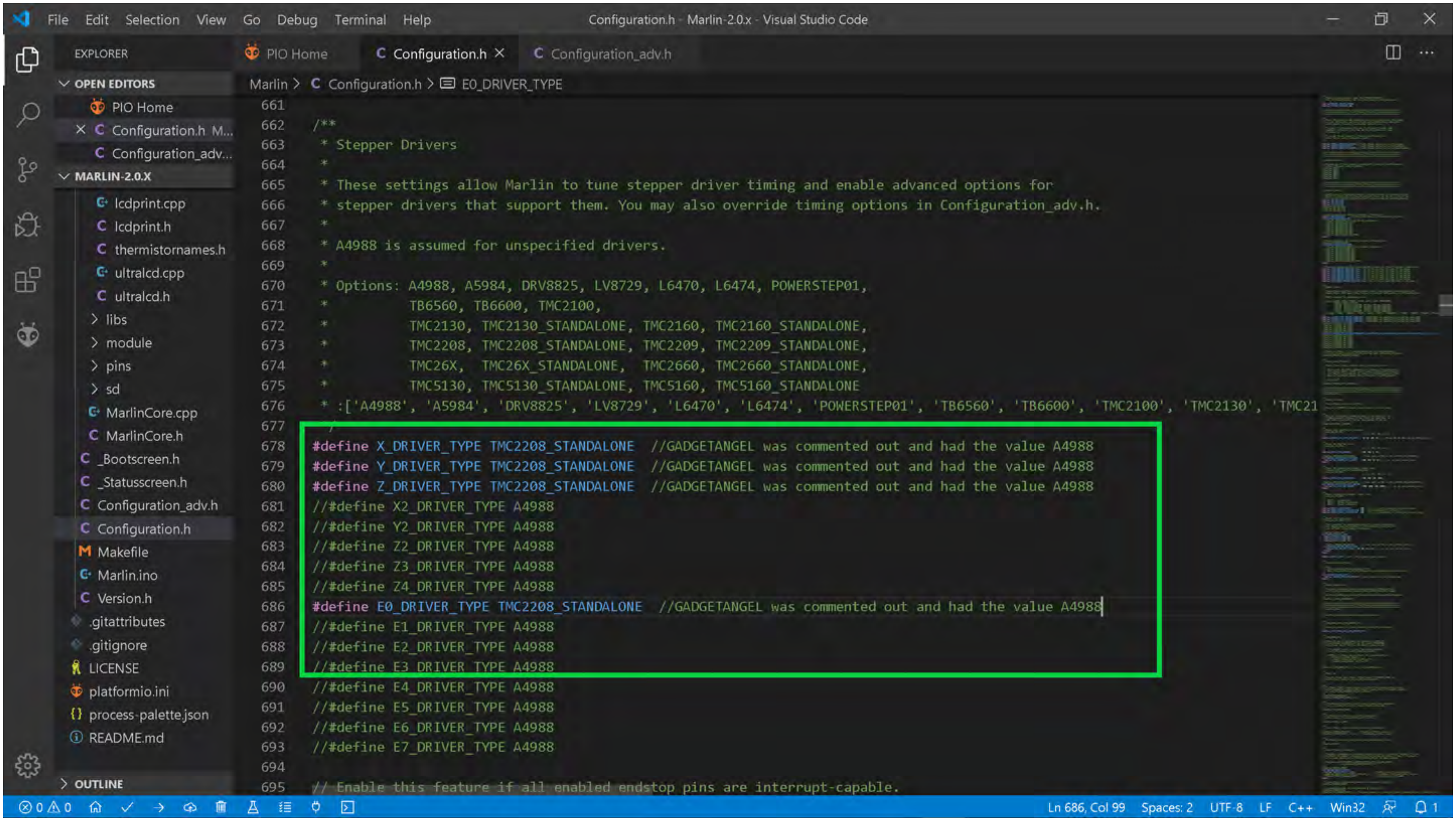


See Appendix D for legend

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

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

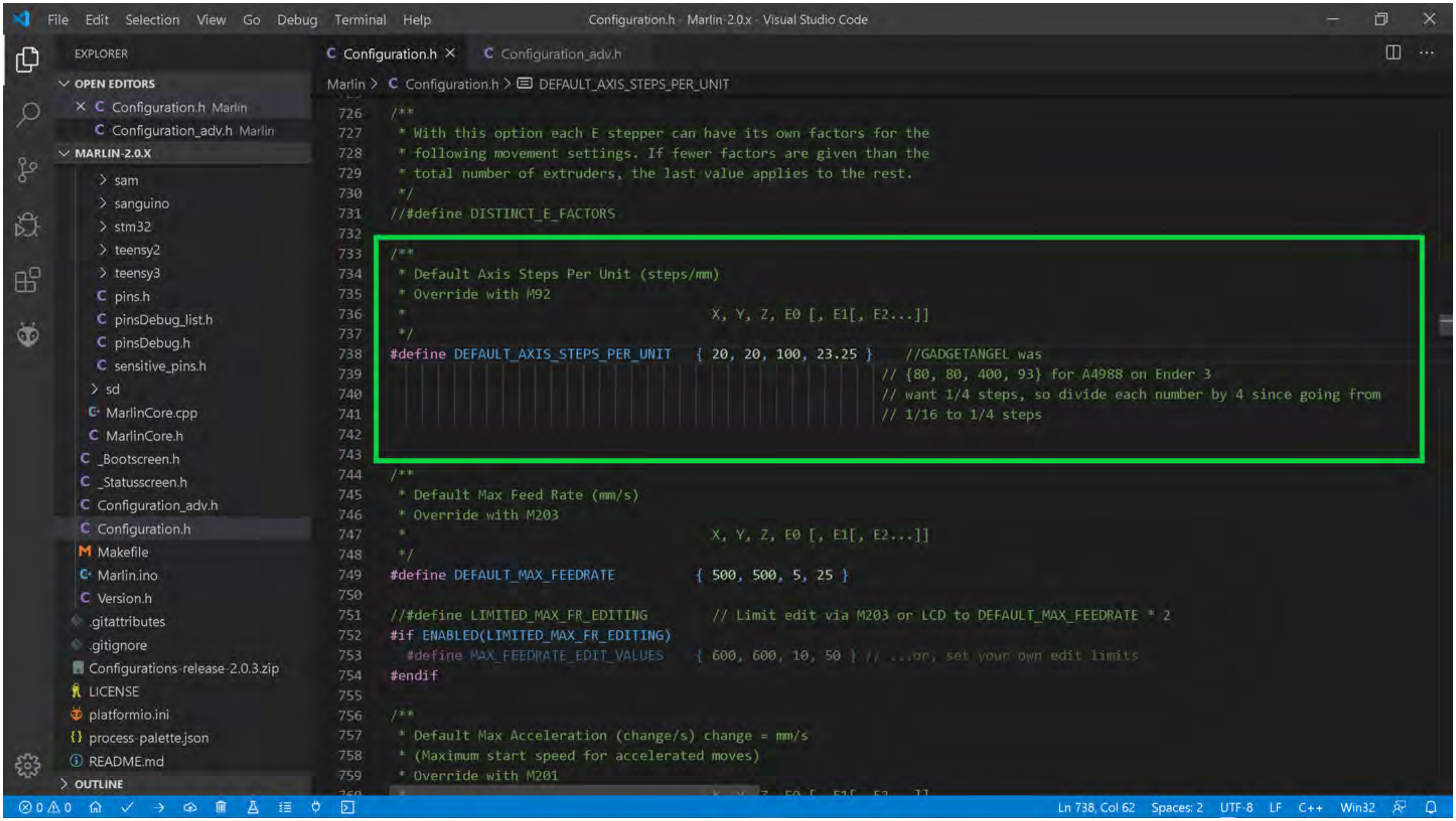
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

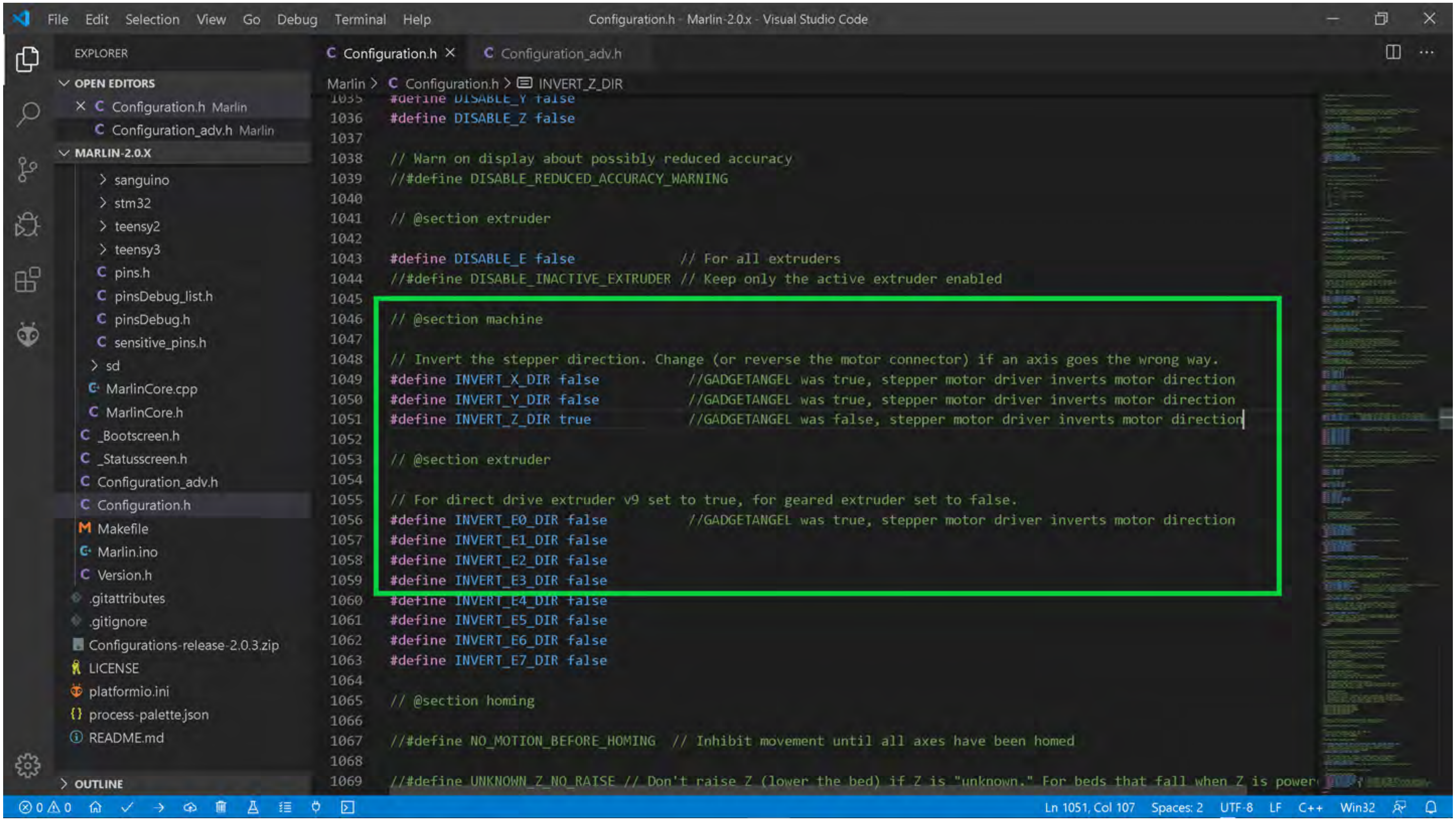
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

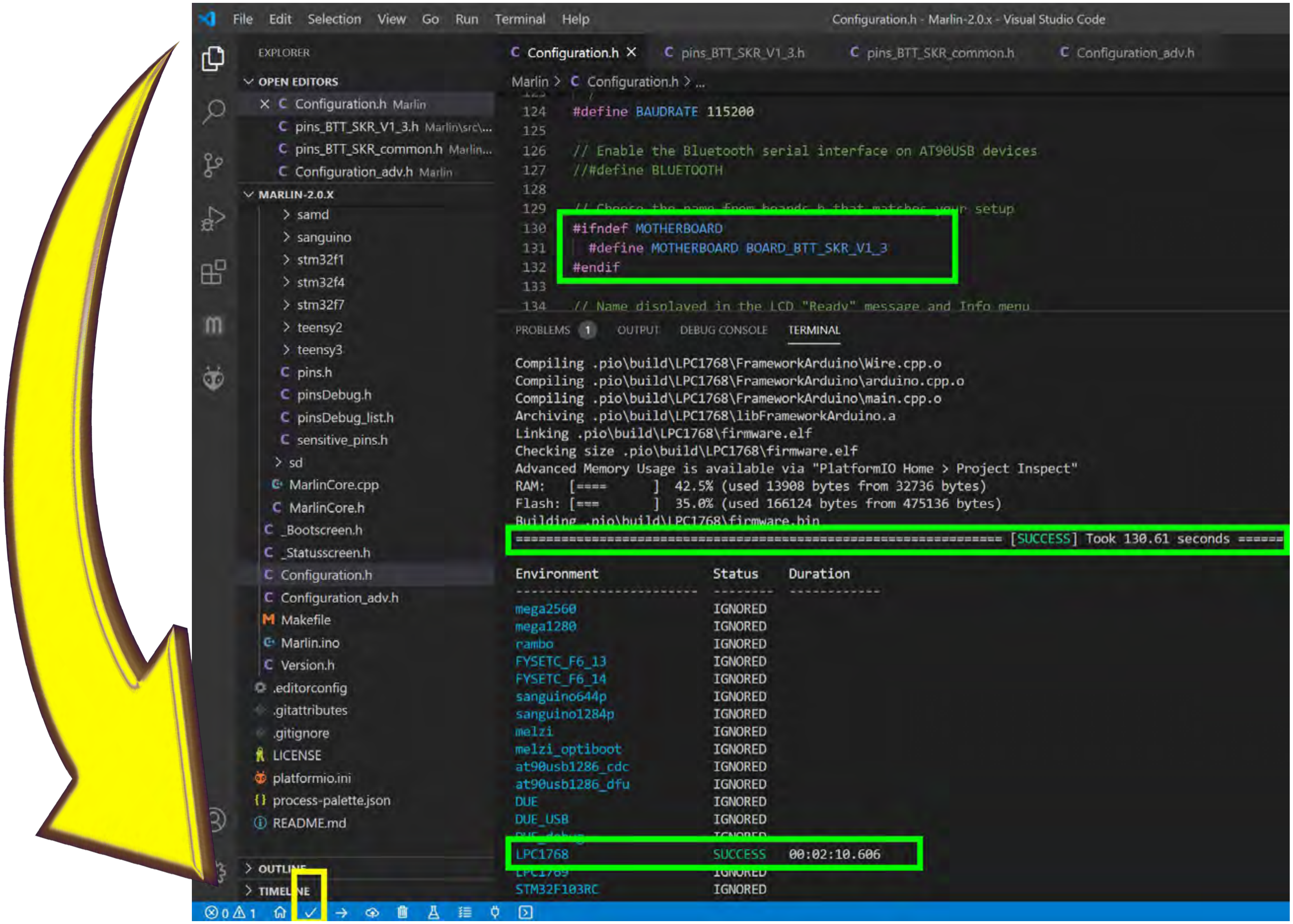
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

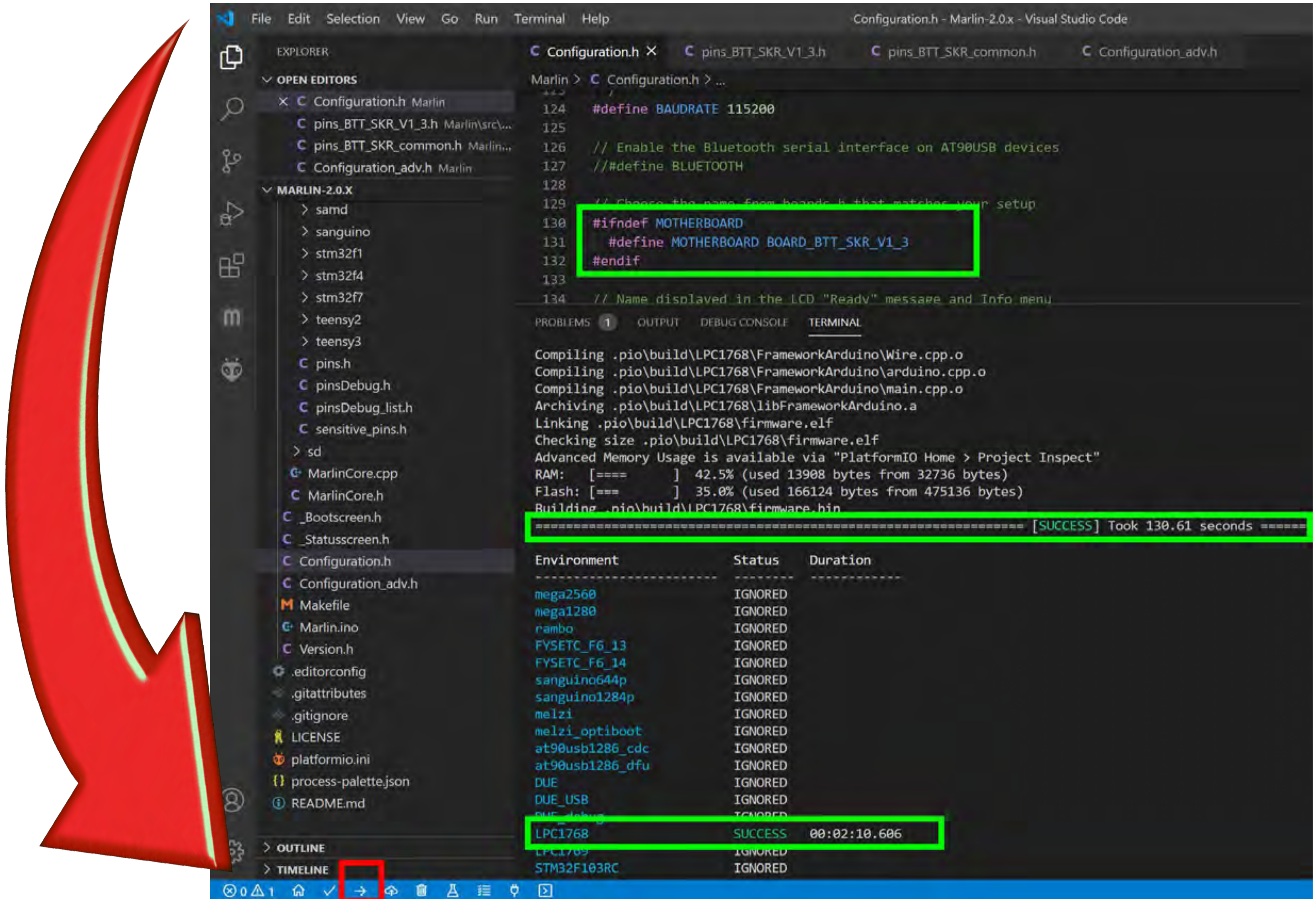
- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

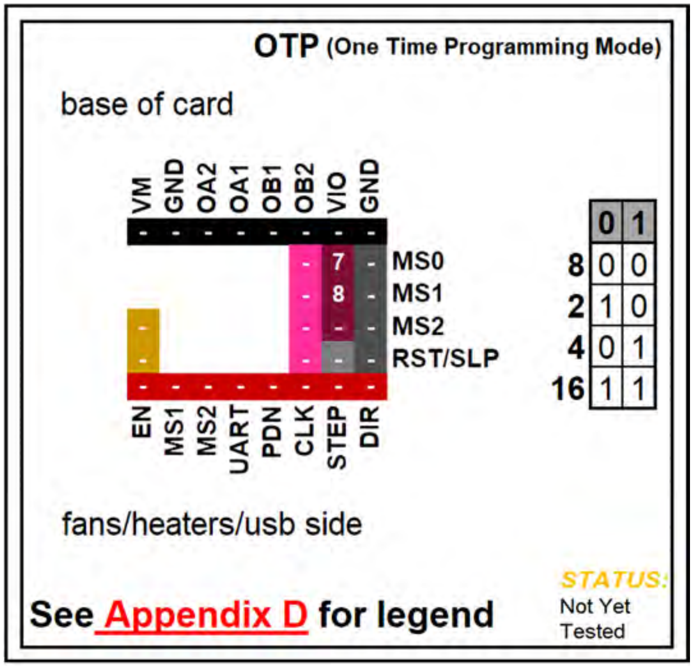
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

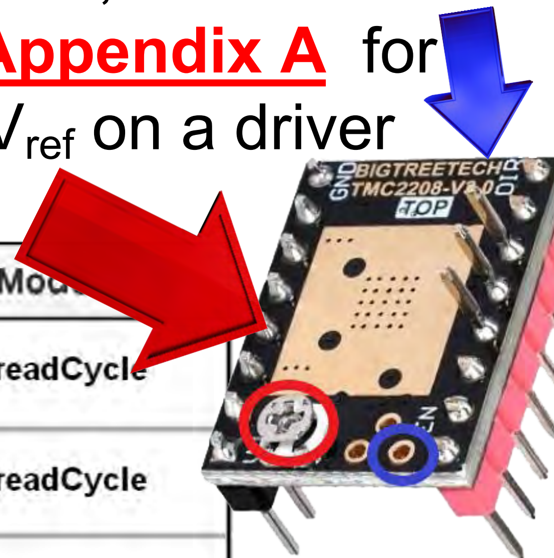
BIQU TMC2208 V3.0

One Time Programming (OTP) Mode



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

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



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

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

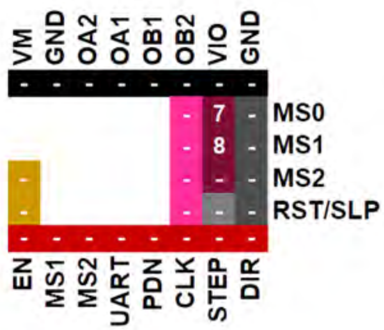
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

BIQU TMC2208 V3.0

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

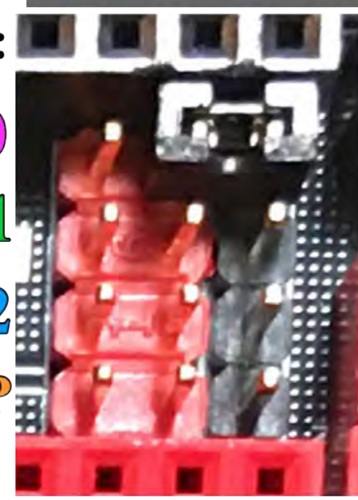
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low → No Jumper

Rows:
MS0
MS1
MS2
RST/SLP

High → Jumper Set

Rows:
MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
Chip Name	Low	High	Low	1/4 Step	W1-2 Phase
Maximum XXX	High	High	Low	1/8 Step	2W1-2 Phase
Subdivision	Low	Low	High	1/16 Step	4W1-2 Phase
XX V DC	High	Low	High	1/32 Step	8W1-2 Phase
xxA (peak)	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

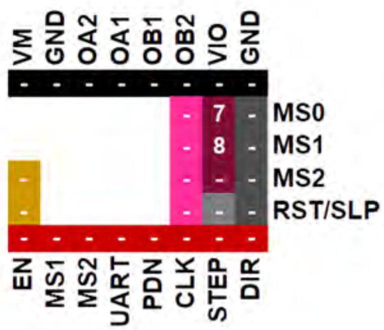
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

BIQU TMC2208 V3.0

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

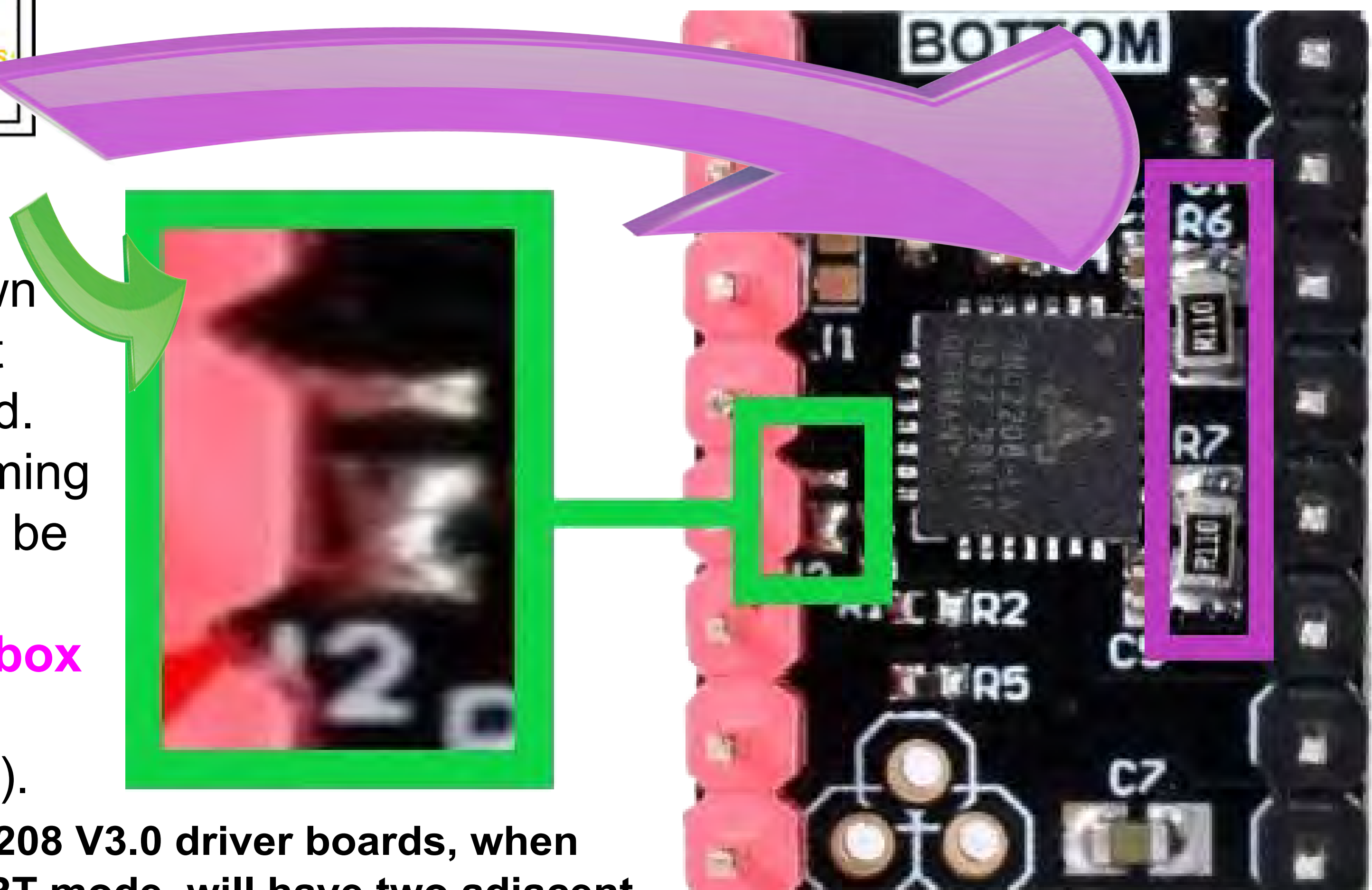
See Appendix D for legend

STATUS: Not Yet Tested

NOTE: **Stand-alone Mode by default uses StealthChop, if you want SpreadCycle, you MUST use OTP mode. See TMC220x Configurator for One-Time-Programming Information: TMC220x Configurator.**

Important: To place BIQU TMC2208 V3.0 into OTP mode, at J2, you must solder the top two pads together OR solder the bottom two pads together on the bottom of the driver board.

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



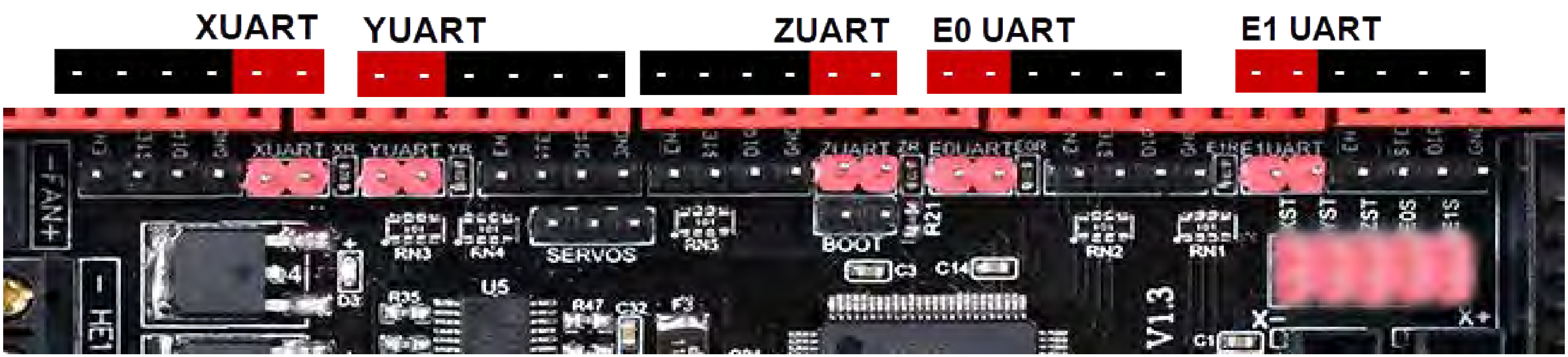
MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

OTP (One Time Programming) Mode

One Time Programming (OTP) Mode

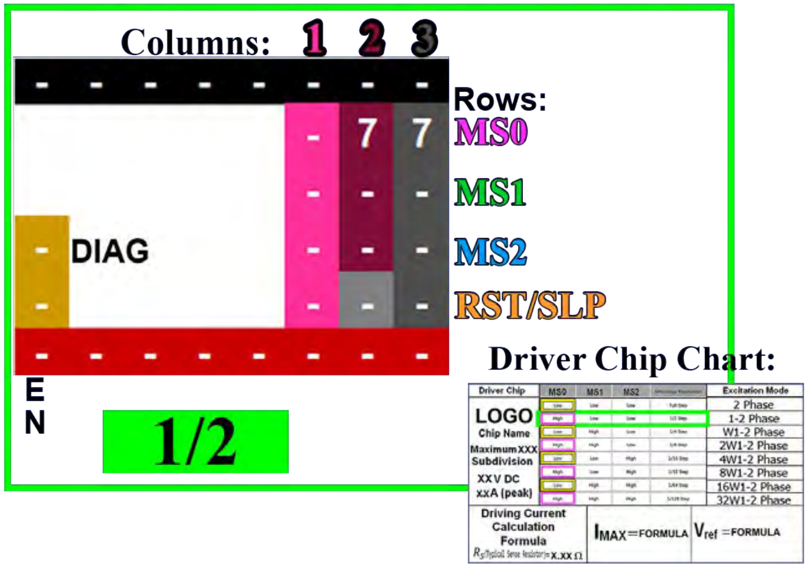
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation.

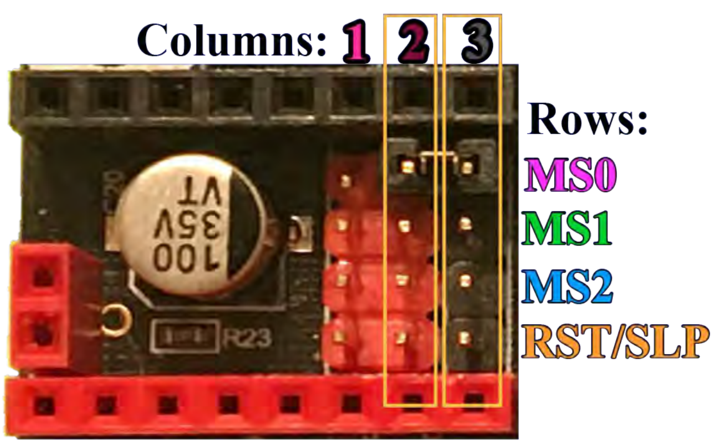


SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

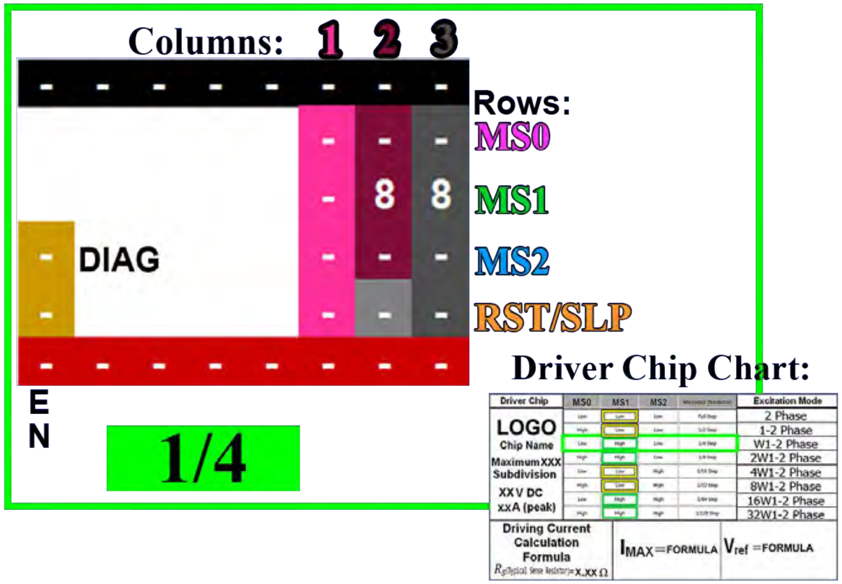
Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:	Meaning:
High	set Jumper between column 2 and column 3 on the MS0 row
Low	No Jumper Set

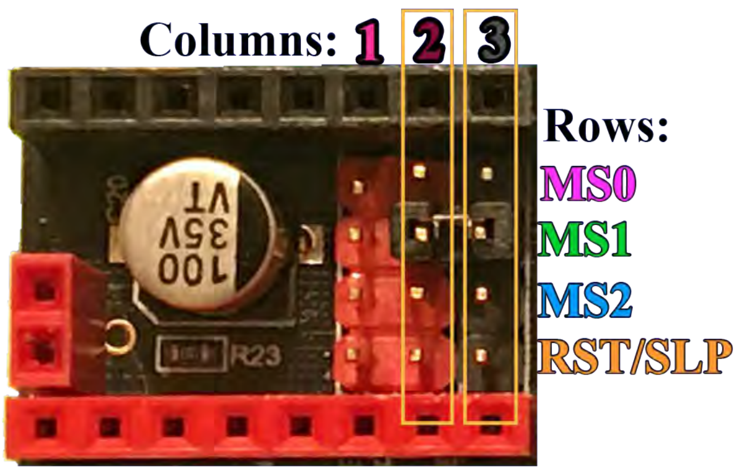
OTP (One Time Programming) Mode

One Time Programming (OTP) Mode

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Meaning:

Driver Chip Chart:		
High	→	**set Jumper between column 2 and column 3 on the MS1 row**
Low	→	**No Jumper Set**

OTP (One Time Programming) Mode

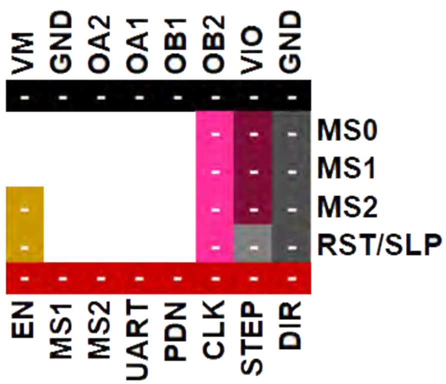
One Time Programming (OTP) Mode

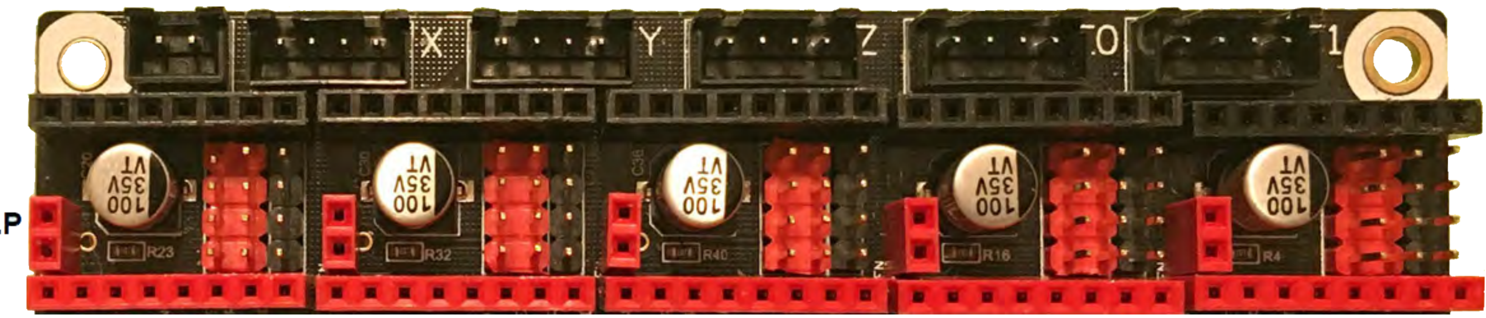
OTP

1 / 8

Interpolation:
1 / 256

SpreadCycle





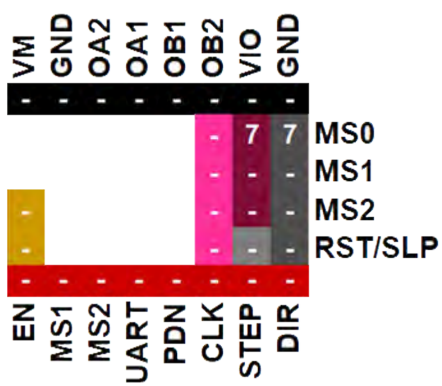
See [Appendix D](#) for legend

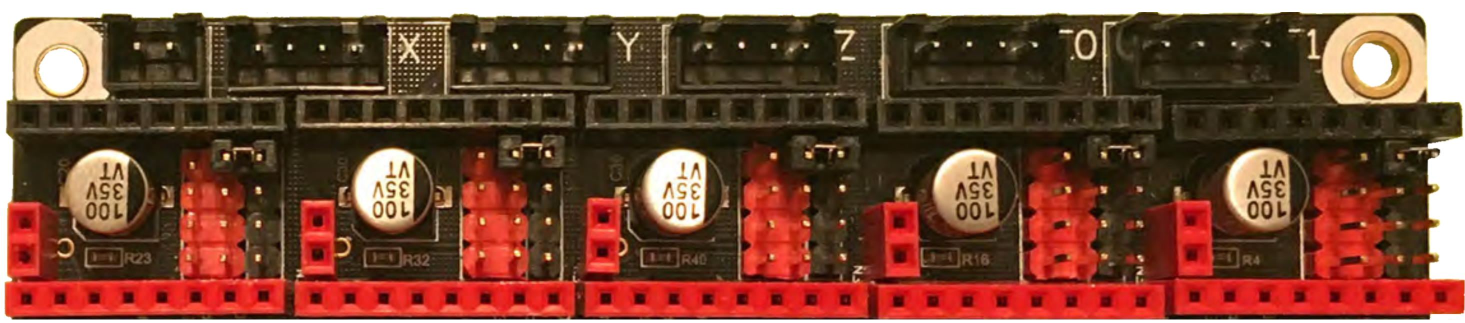
OTP

1 / 2

Interpolation:
1 / 256

SpreadCycle





See [Appendix D](#) for legend

OTP (One Time Programming) Mode

BIQU TMC2208 V3.0

One Time Programming (OTP) Mode

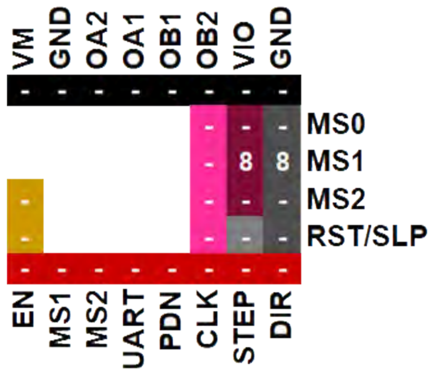
OTP

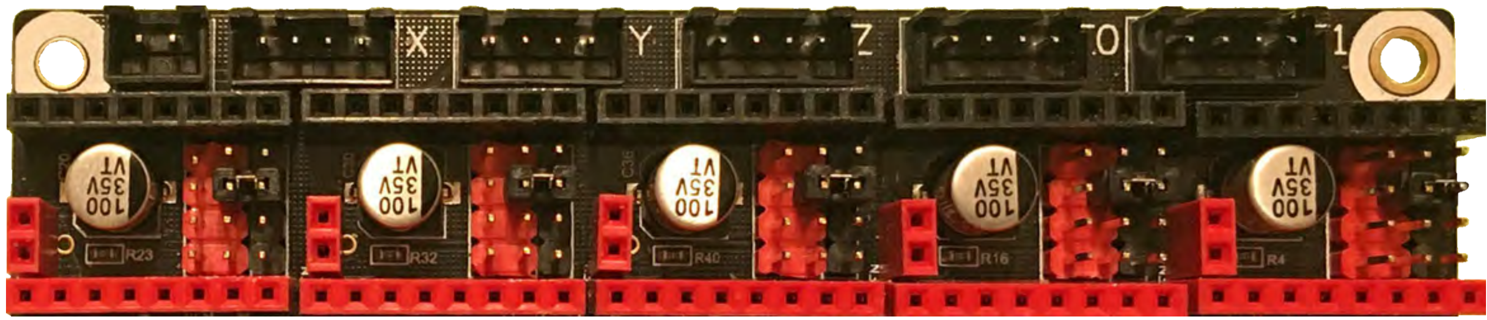
1 / 4

Interpolation:

1 / 256

SpreadCycle





See Appendix D for legend

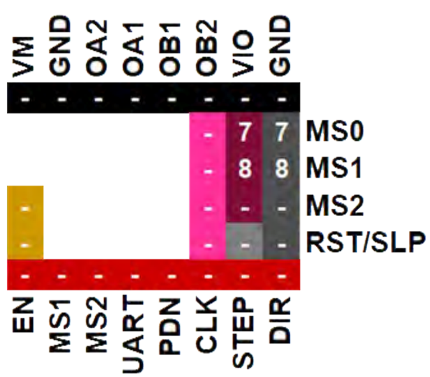
OTP

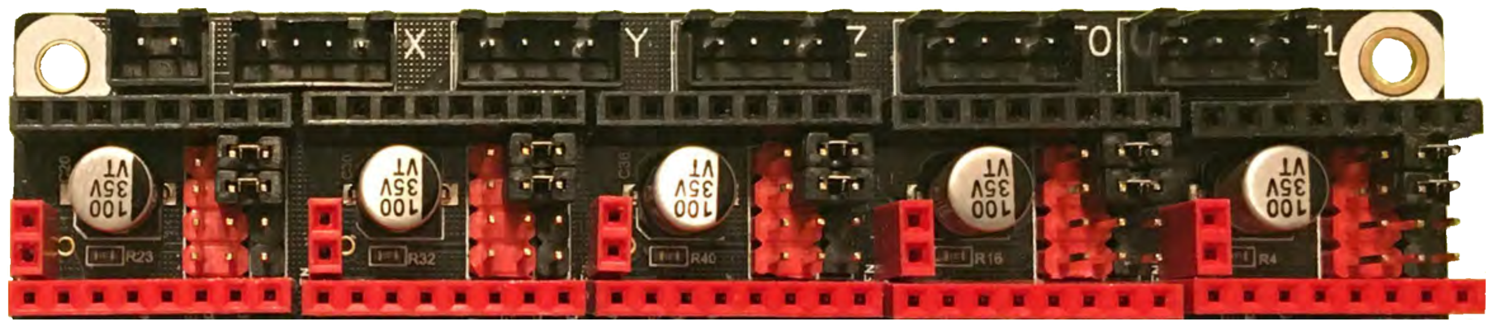
1 / 16

Interpolation:

1 / 256

SpreadCycle



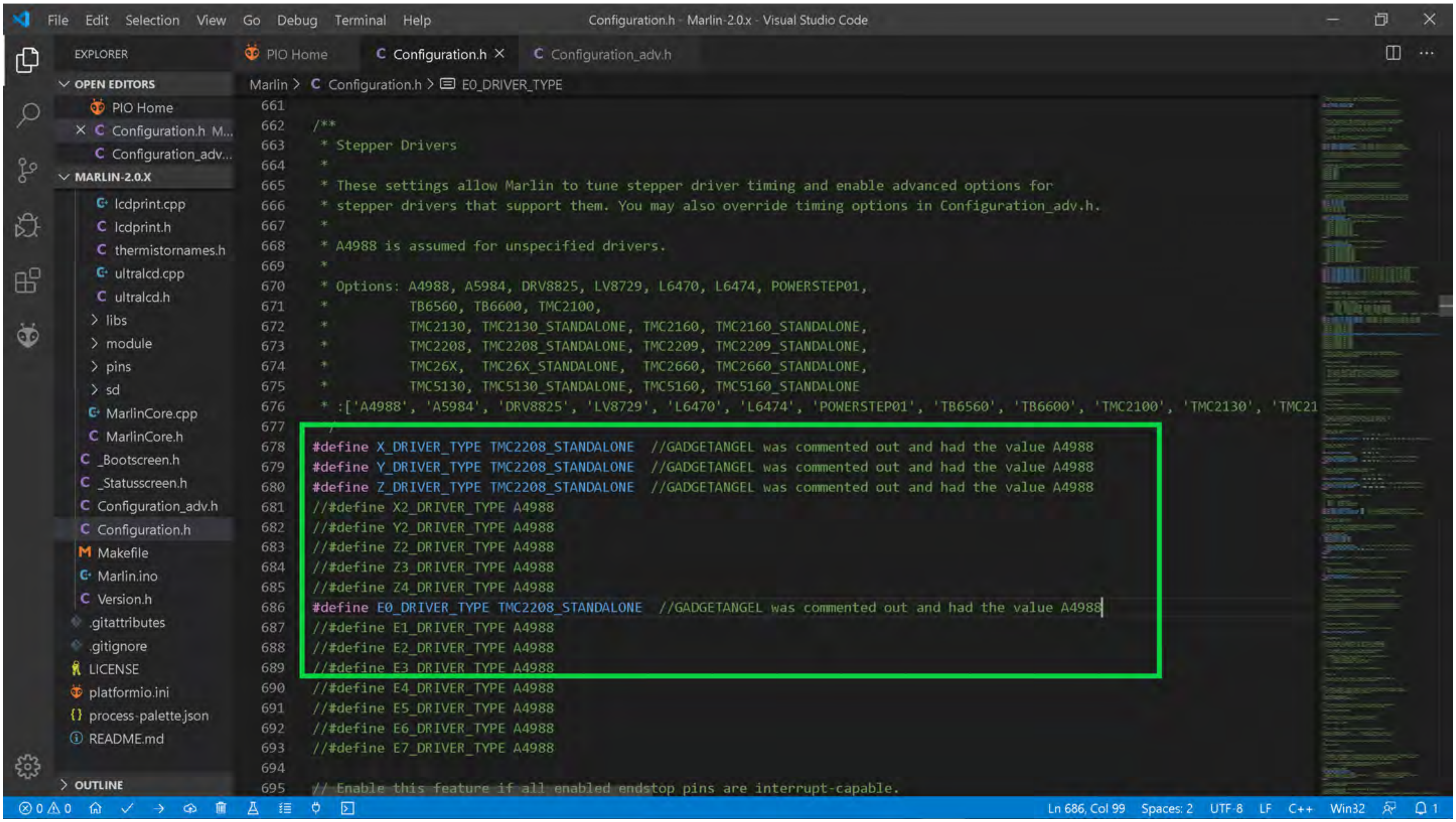


See Appendix D for legend

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

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

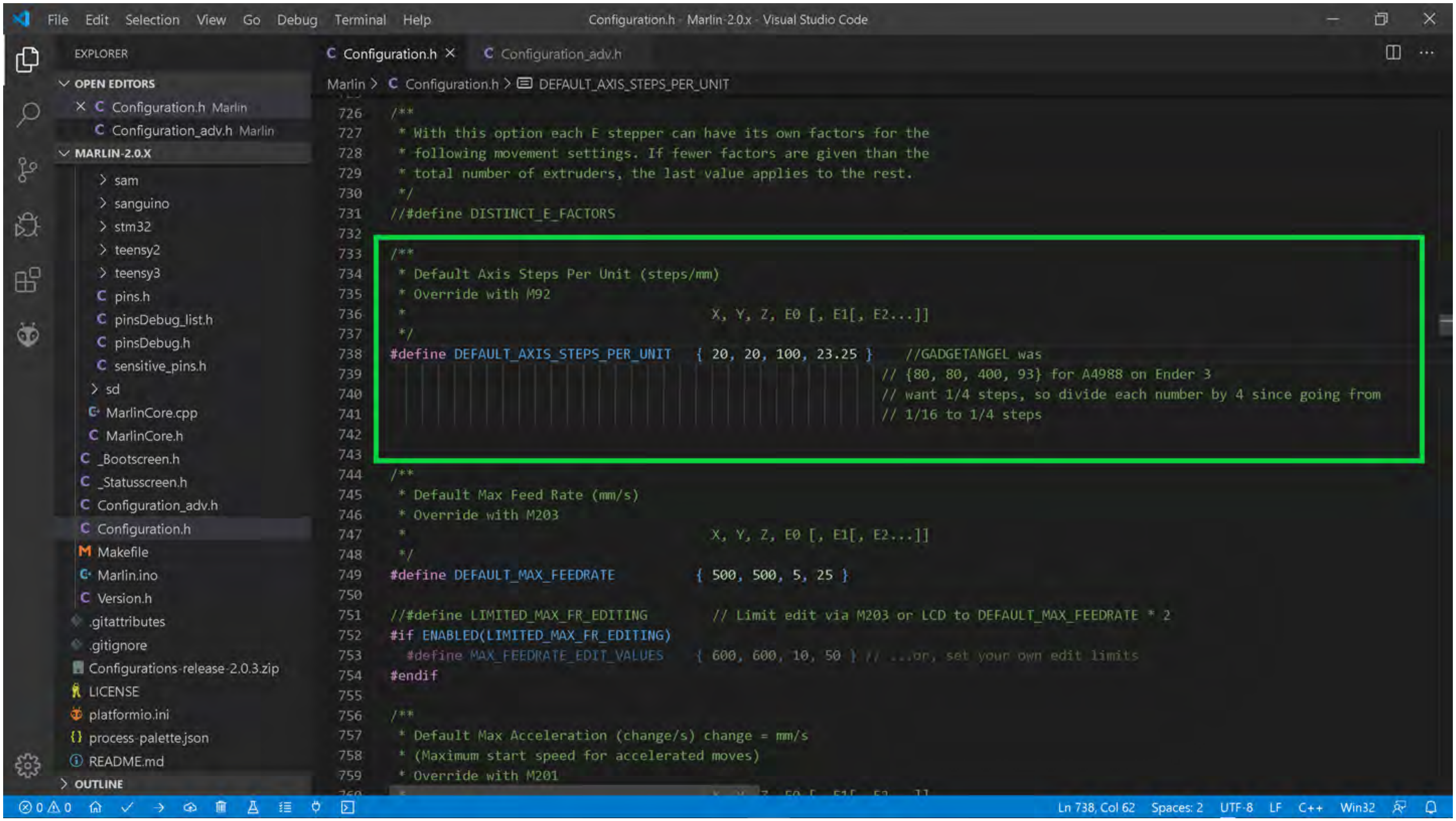
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "//" appear at the beginning of a line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



- Go to the next page.

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

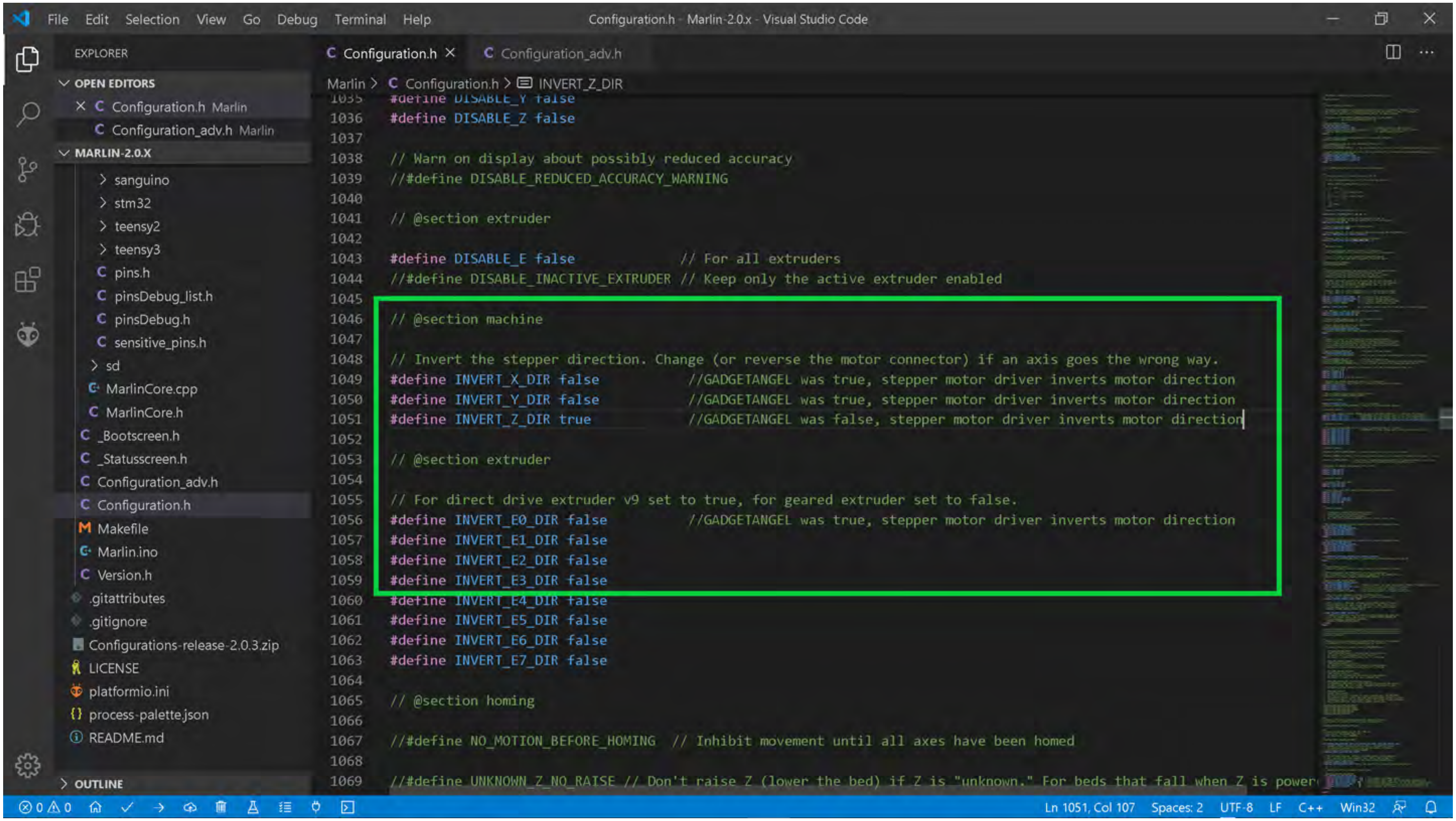
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

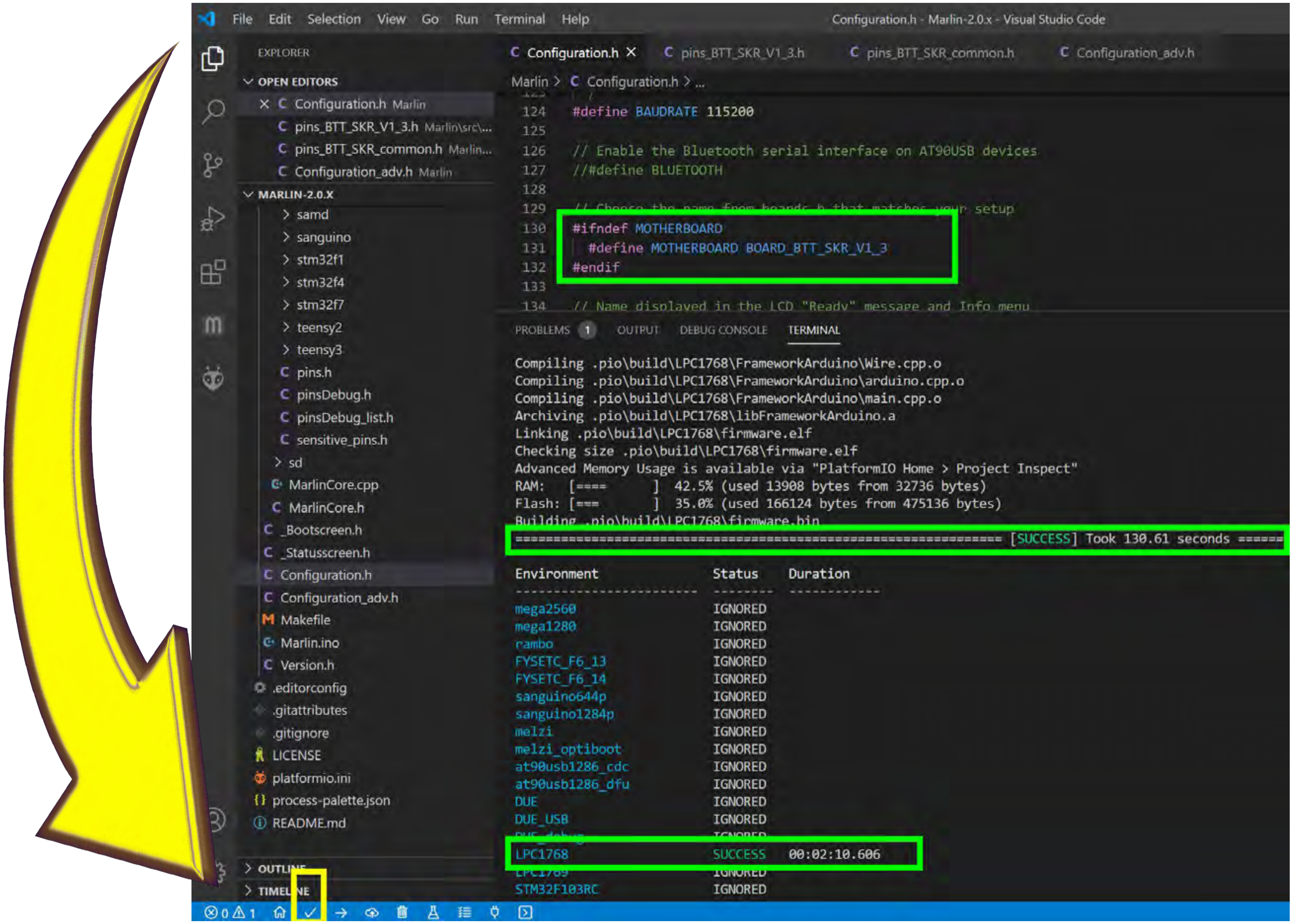
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

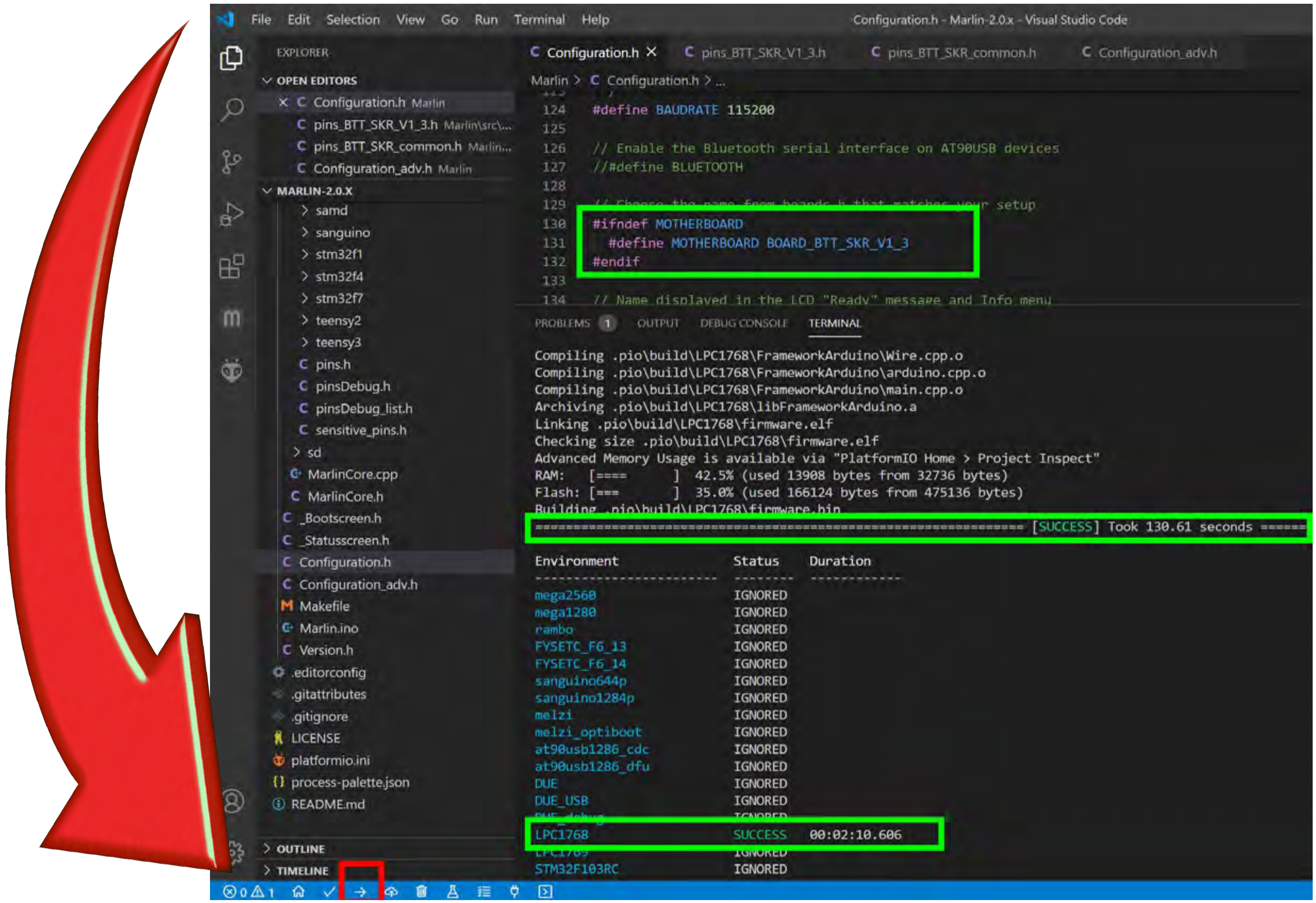
- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

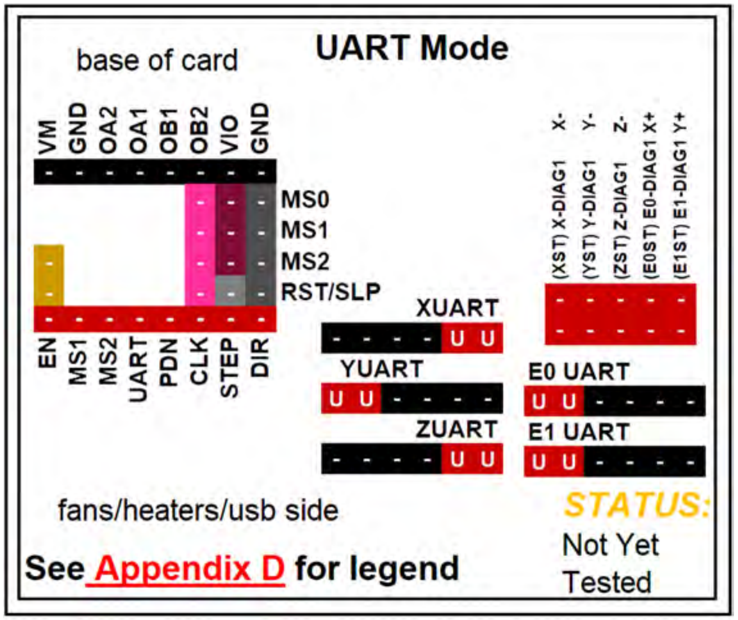
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2208 V3.0

UART Mode



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

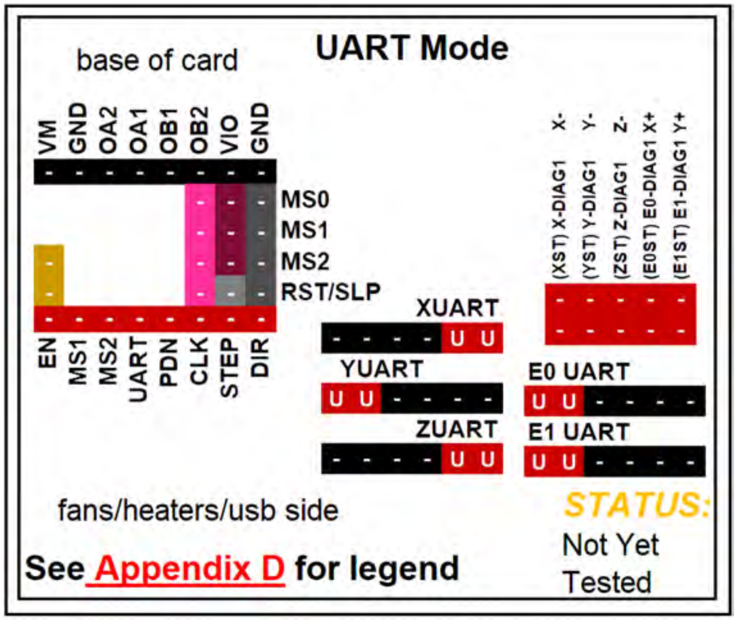
See the next page for further information.

Driver Chip	
BIQU® TMC2208 UART Mode Maximum 256 Subdivision 35V DC 2A (peak)	Steps are set inside of your Firmware

Driving Current Calculation Formula	$I_{MAX}=V_{ref}$	$V_{ref}=I_{MAX}$
R_S(Typical Sense Resistor)= 0.11Ω	See Appendix B #4. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$	See Appendix B #4. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$

BIQU TMC2208 V3.0

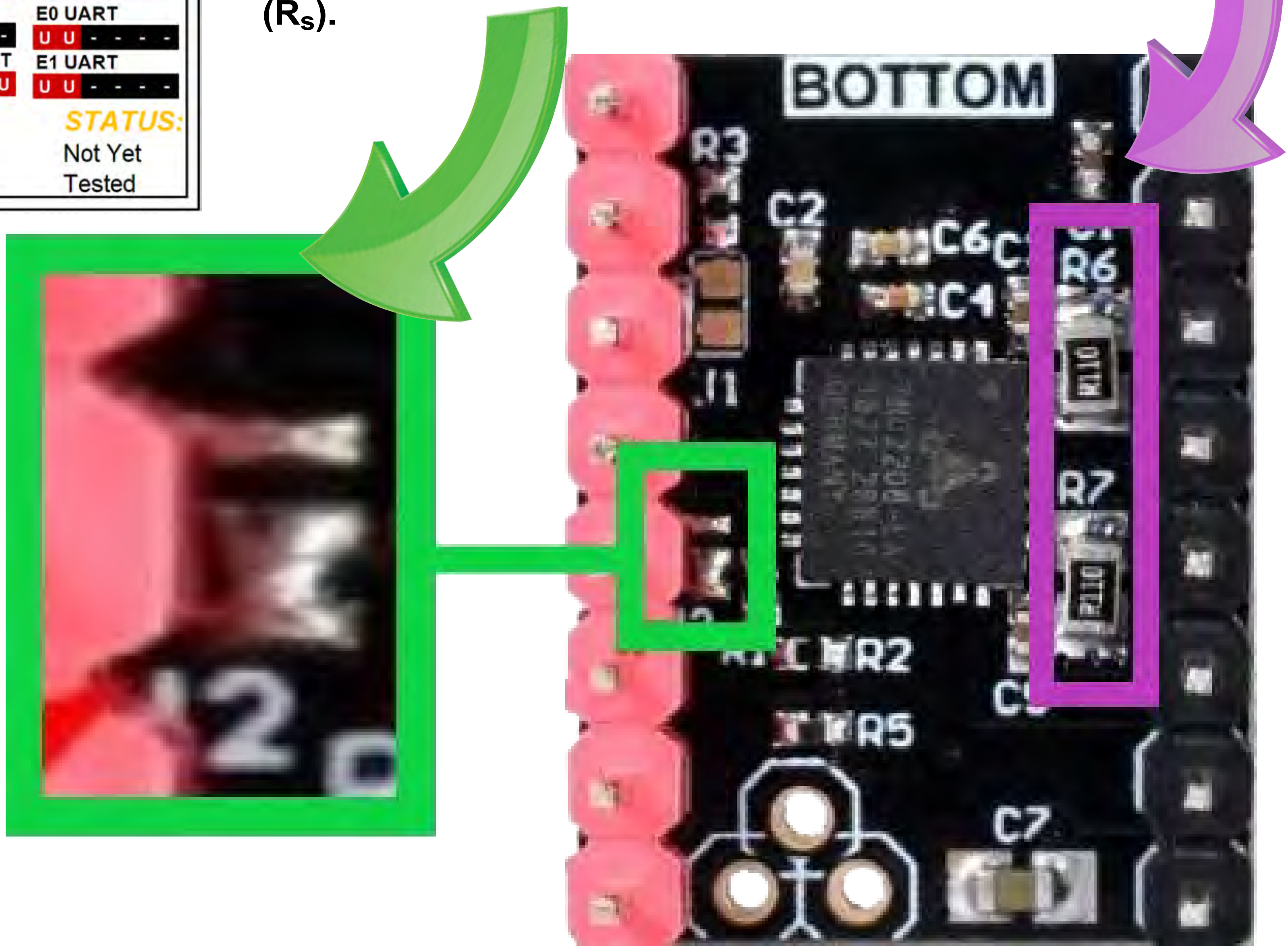
UART Mode



See Appendix D for legend

Important: **To ensure that the BIQU TMC2208 V3.0 is in UART Mode, check to see if two adjacent J2 pads are soldered together on the bottom of the driver board, as seen in the GREEN box below. The PURPLE box shows the location of the current sense resistors (R_S).**

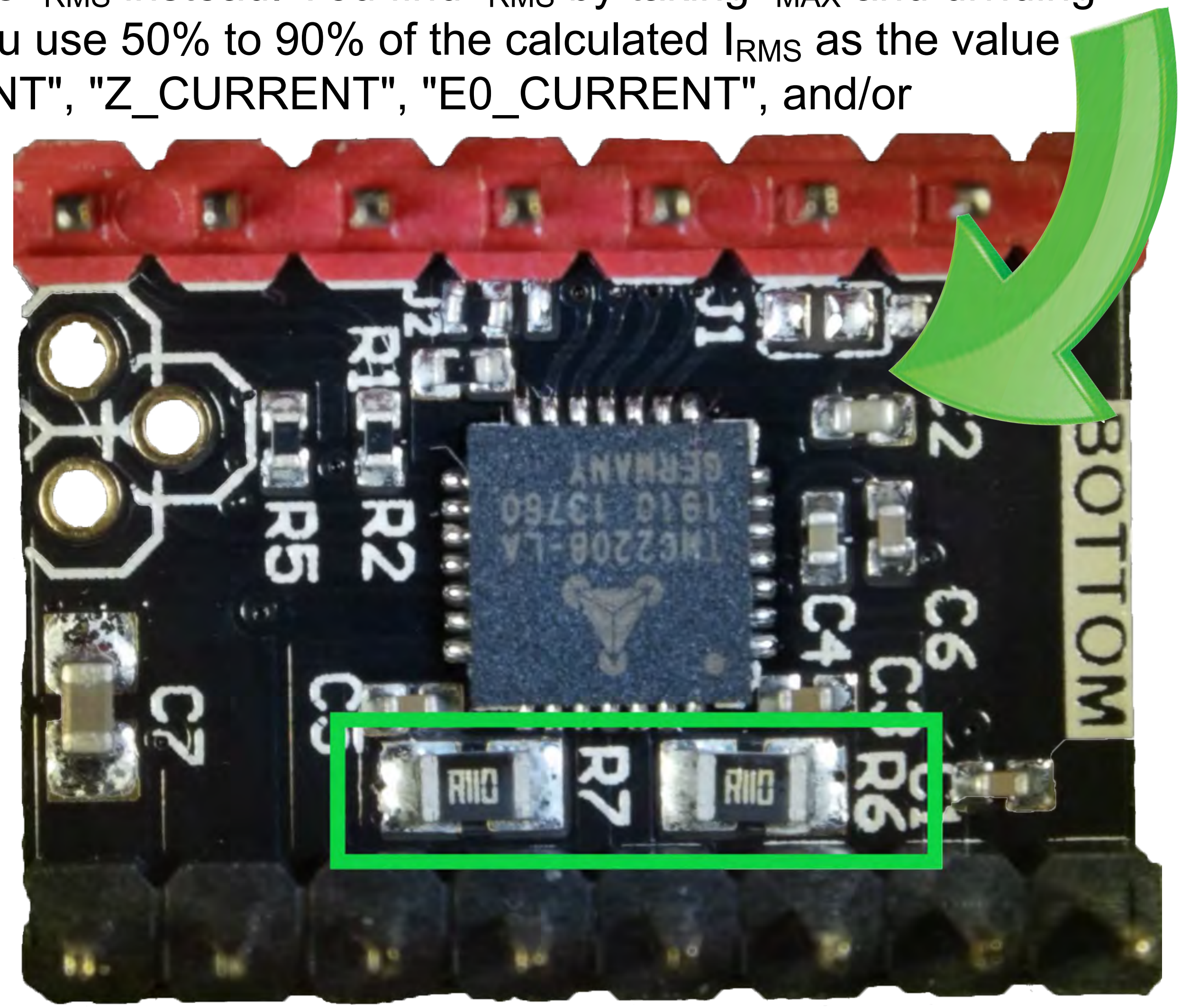
MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).



UART Mode

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_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



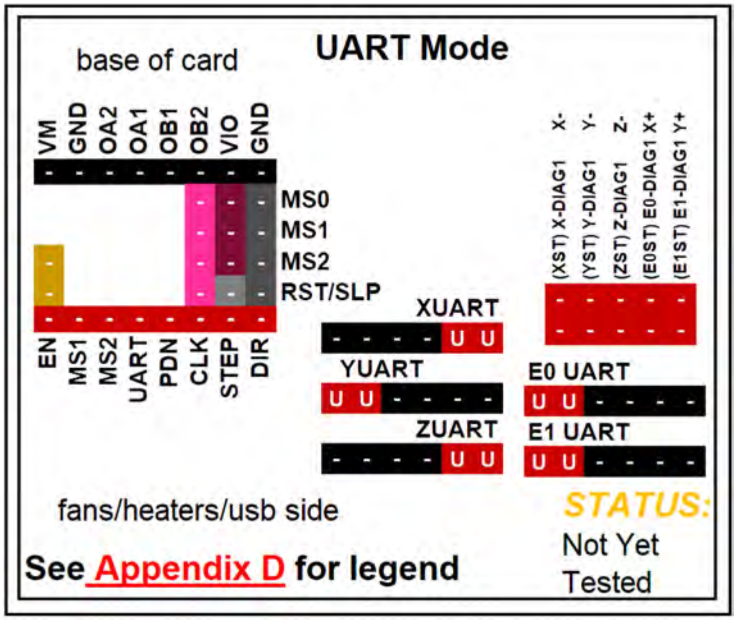
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to ensure proper operation for UART mode.

Note: Set the "U" Jumper(s) for UART MODE!

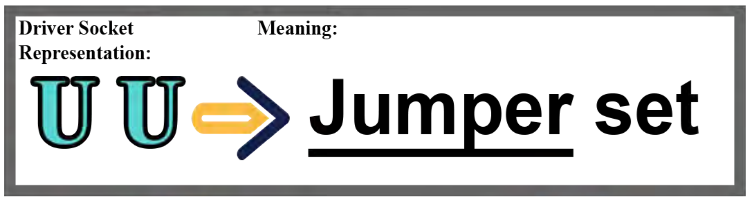
XUART	YUART	ZUART	E0 UART	E1 UART
- - - - U U	U U - - - -	- - - - U U	U U - - - -	U U - - - -

BIQU TMC2208 V3.0

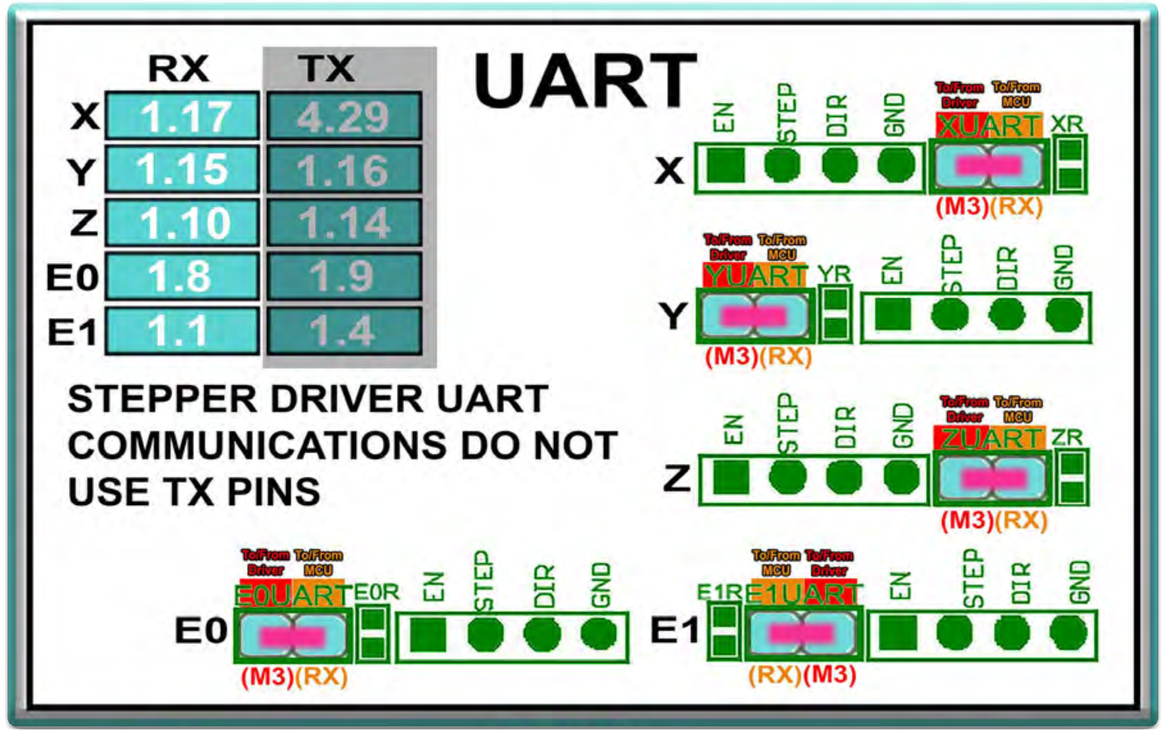
UART Mode

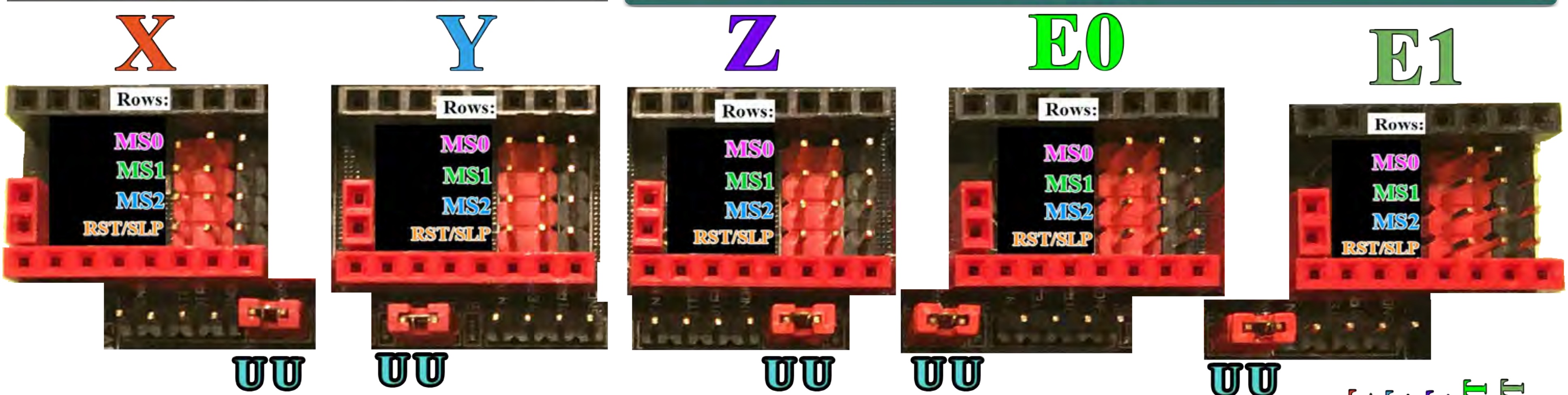


See Appendix D for legend



SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode



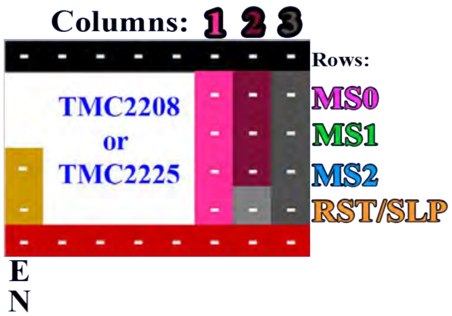


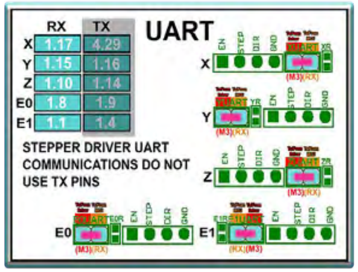
REMOVE 'ST' JUMPER(s) to disable Sensor-less Homing – for drivers without stallGaurd™ feature

UART Mode

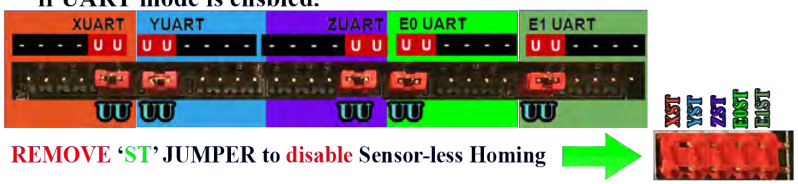
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode

Driver Socket Representation



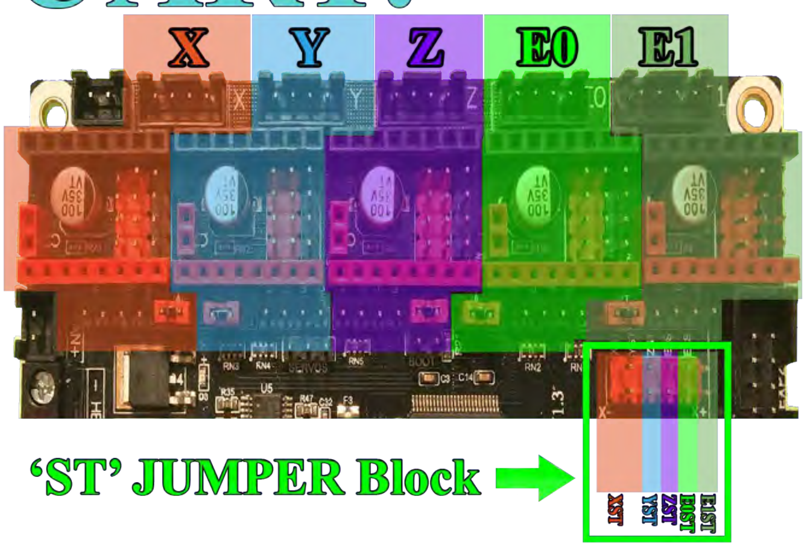


NOTE: SPI PINS must not be set if UART mode is ensbled.



REMOVE 'ST' JUMPER to disable Sensor-less Homing

UART:



'ST' JUMPER Block

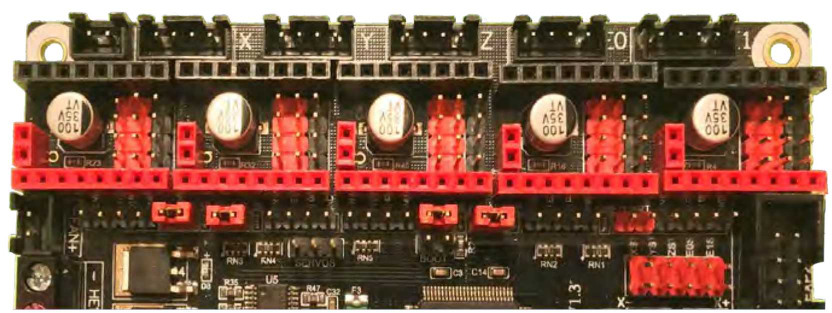
UART Capable Drivers without stallGaurd™ feature: TMC2208, and TMC2225

Driver Socket Representation:	Meaning:
U U	**Jumper Set**

Driver Socket Representation:	Meaning:
U U	set Jumper so that both U PINS are covered by the Jumper

Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS REMOVED. All other PINS need to be empty for UART mode to work properly.

Here is an example of only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS removed for ALL Axes:



BIQU TMC2208 V3.0

UART Mode

Information on Sensor-less Homing

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

NOTE: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to ensure proper operation for UART mode.

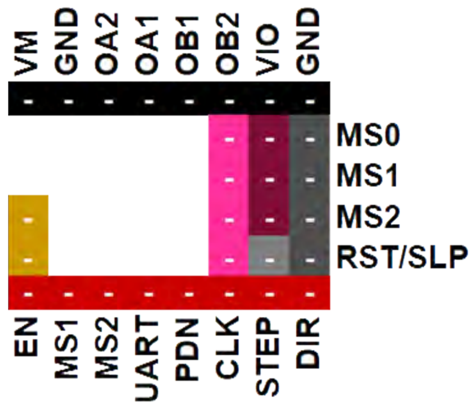
UART Mode

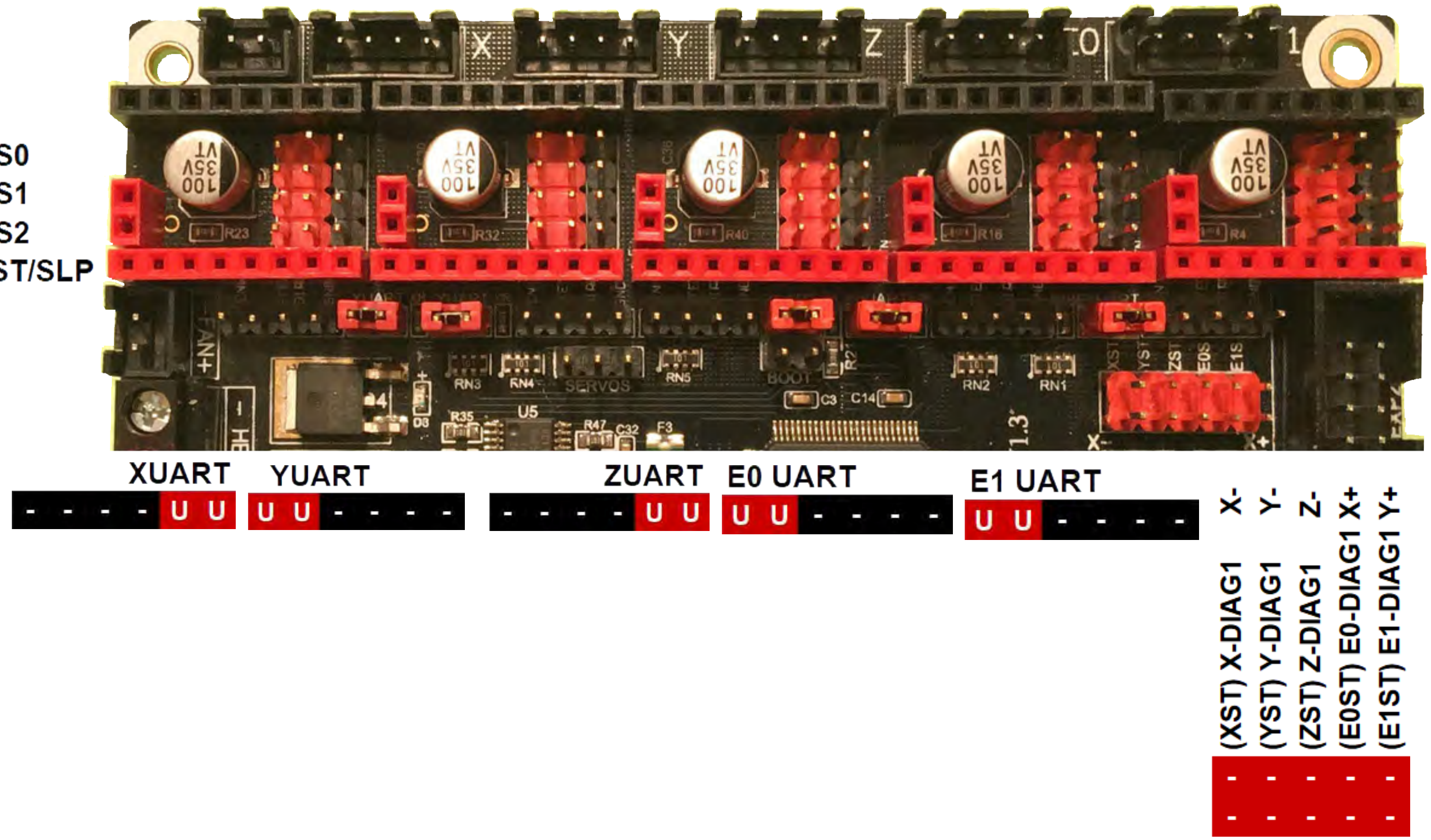
BIQU TMC2208 V3.0

UART Mode

Note: Set the "U" Jumper(s) for UART MODE!

UART



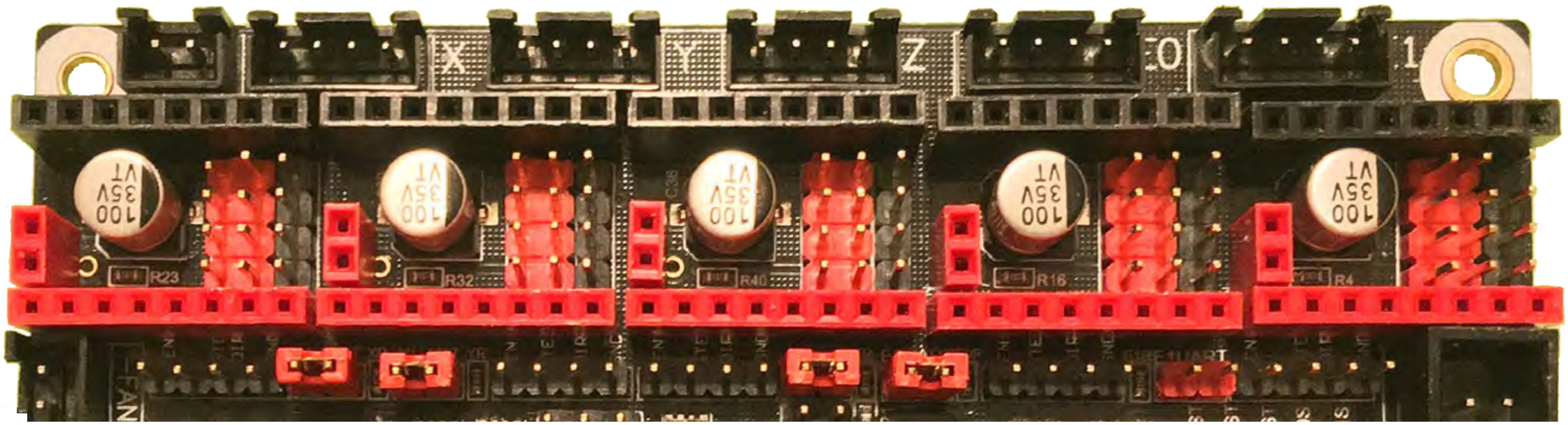


See Appendix D for legend

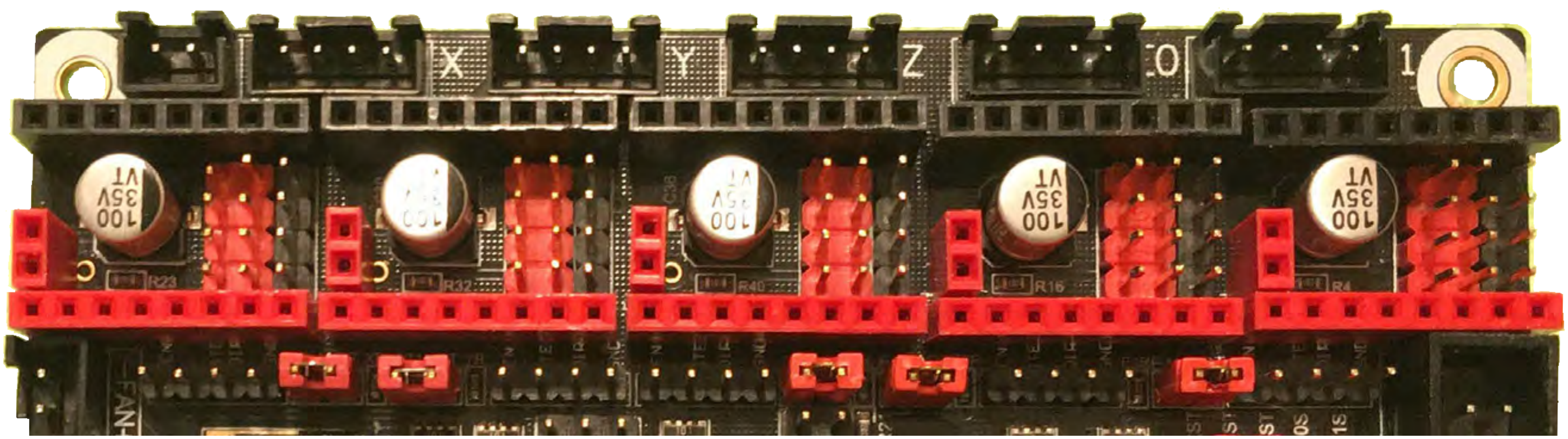
Note: TMC2208 does not have sensor-less homing capability.

Examples of Different UART Configurations

X, Y, Z, and E0 axes configured for UART mode.



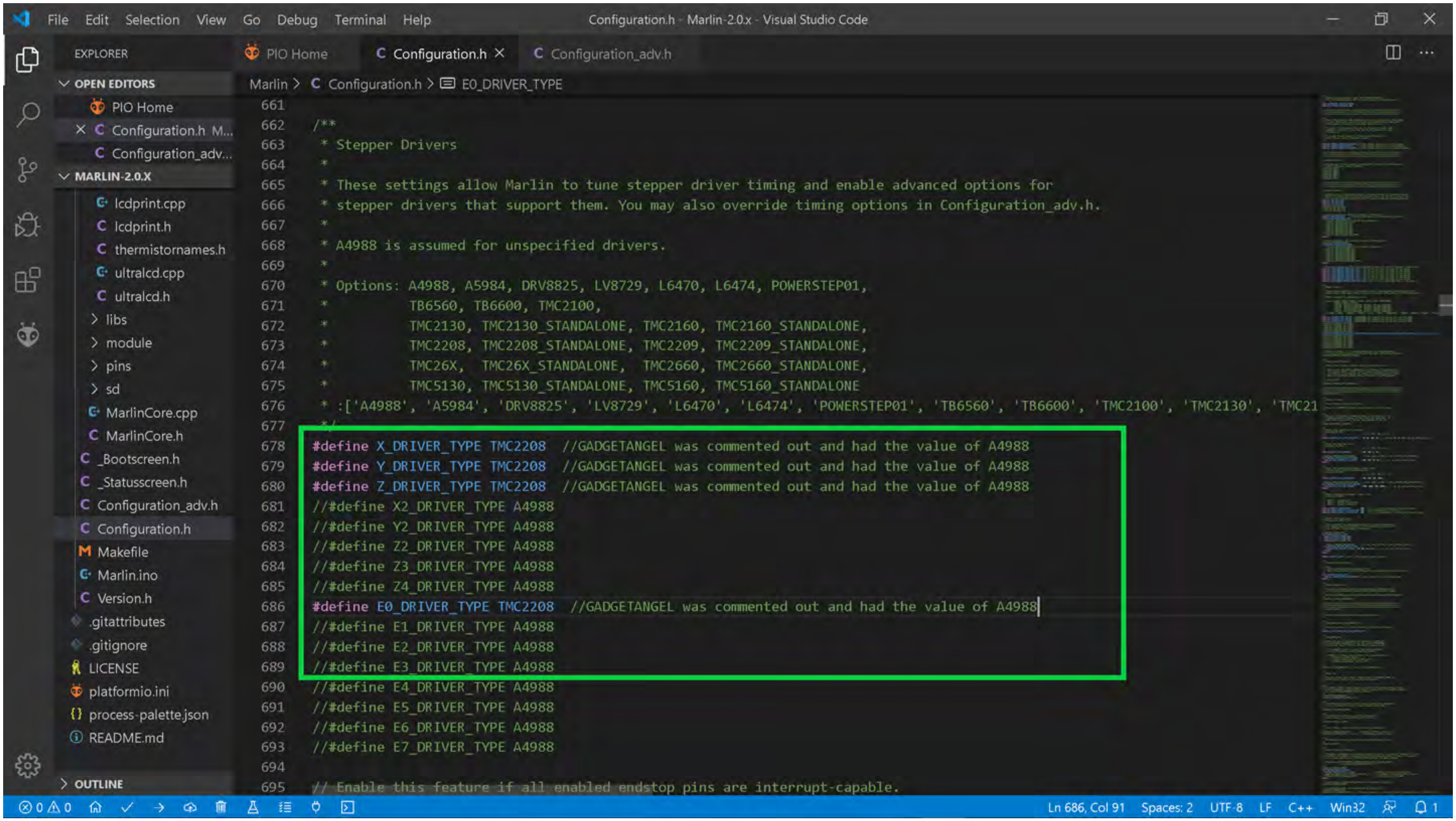
X, Y, Z, E0 and E1 axes configured for UART mode.



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

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

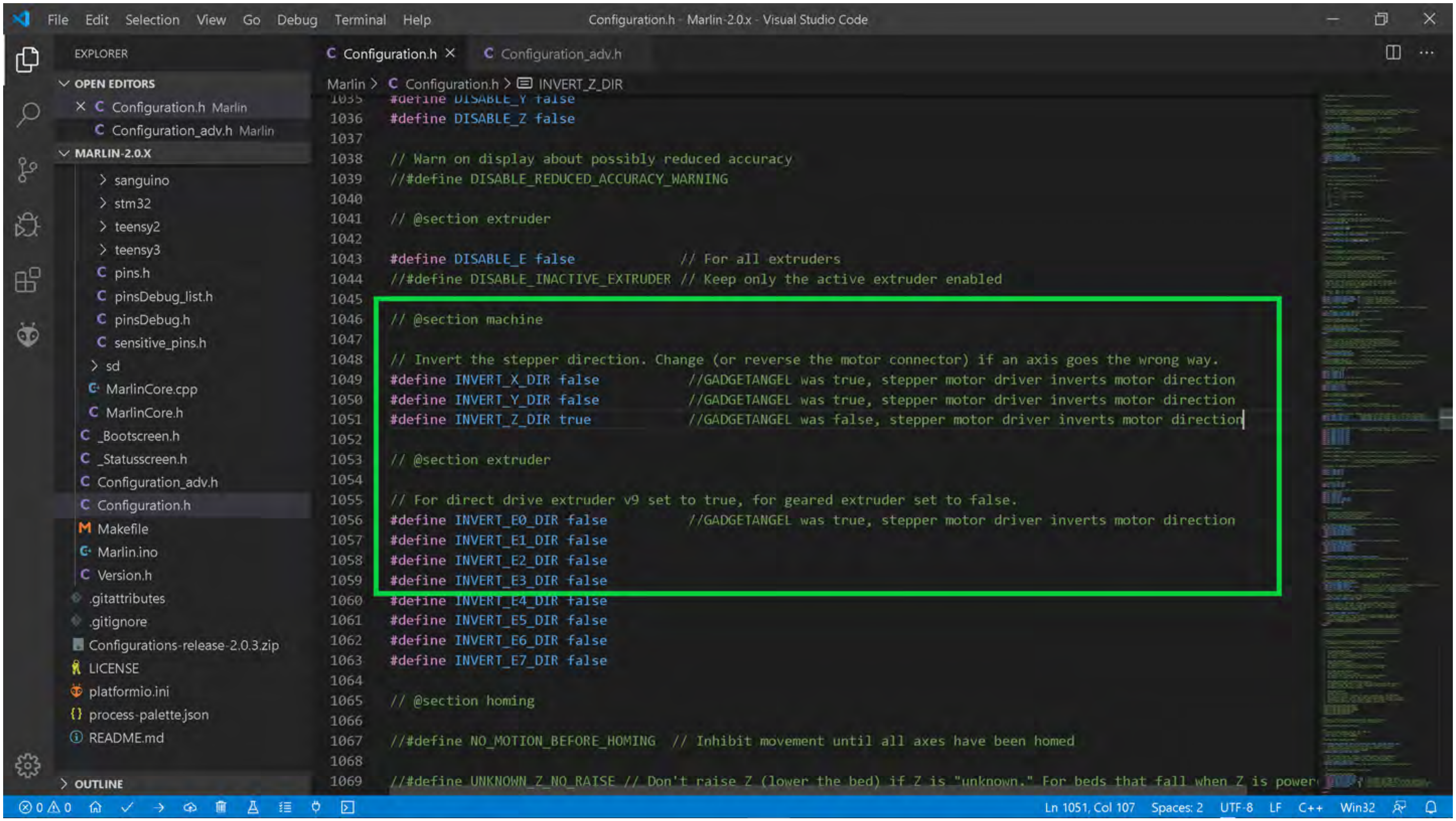
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "//" appear at the beginning of a line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



- Go to the next page.

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

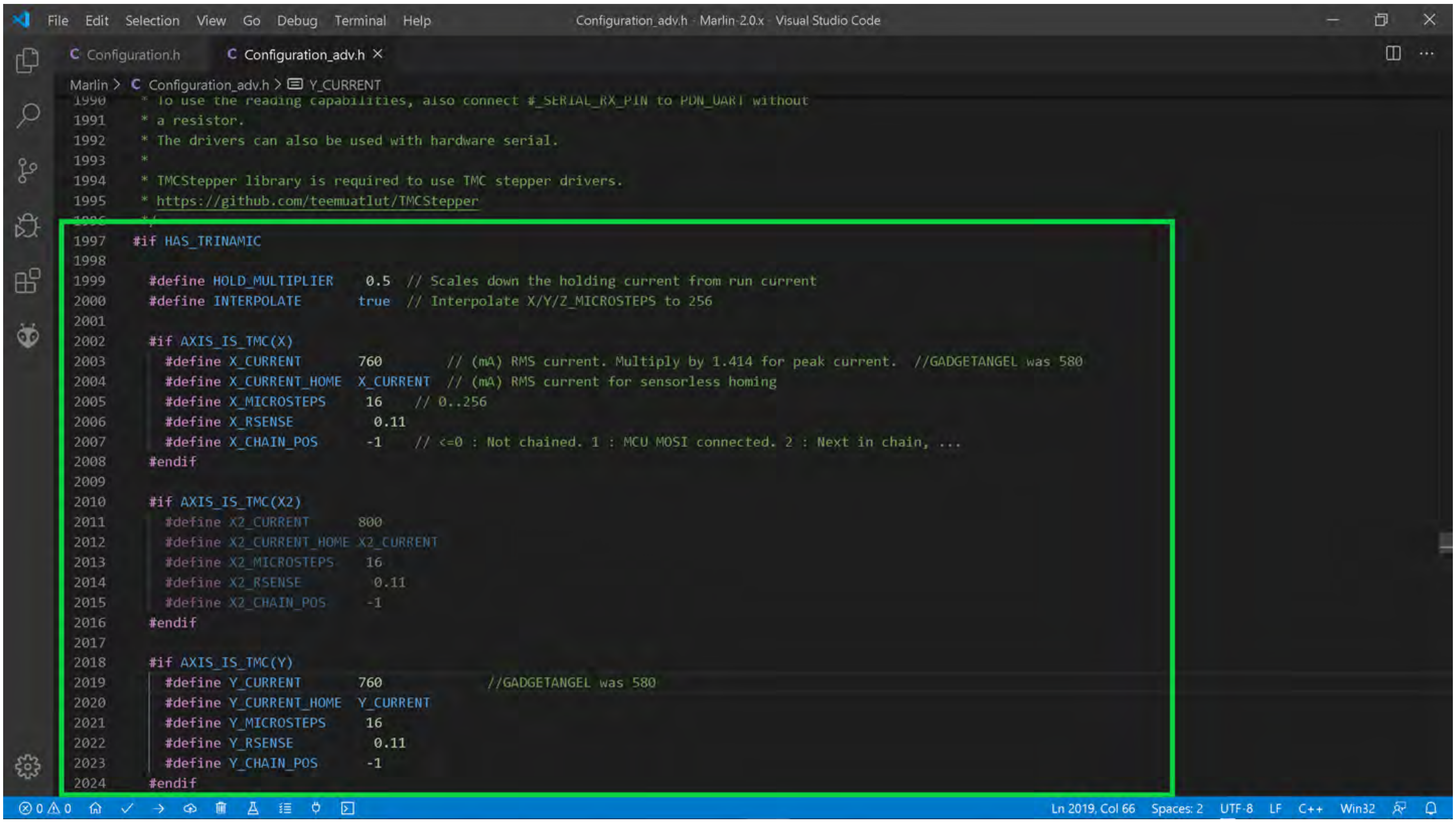
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

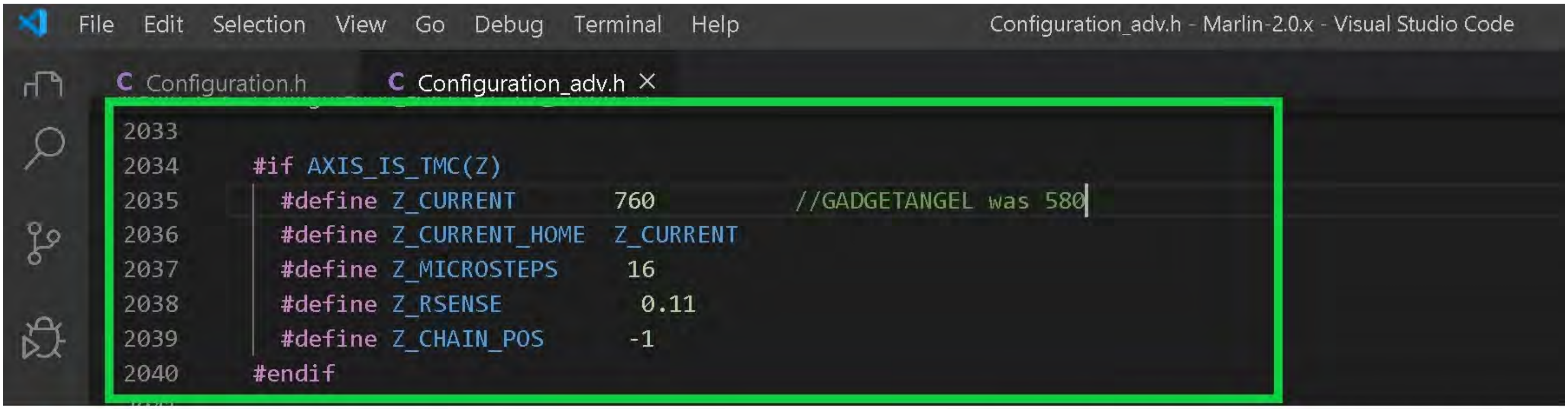
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2208 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "X_CURRENT" and/or "Y_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT" and/or "Y_CURRENT".

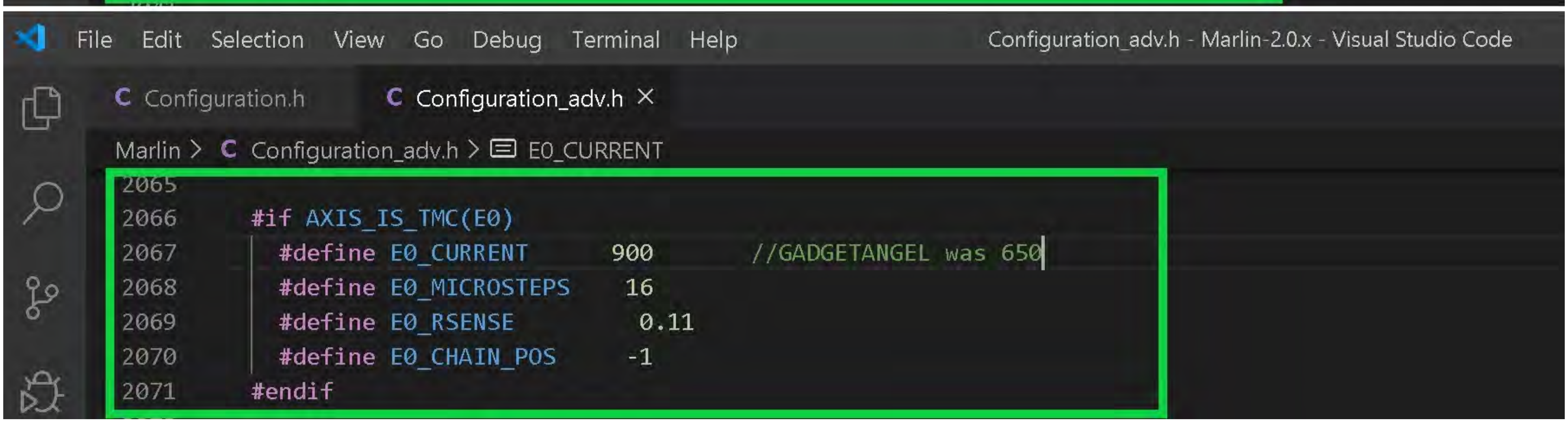


- Go to the next page.

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

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

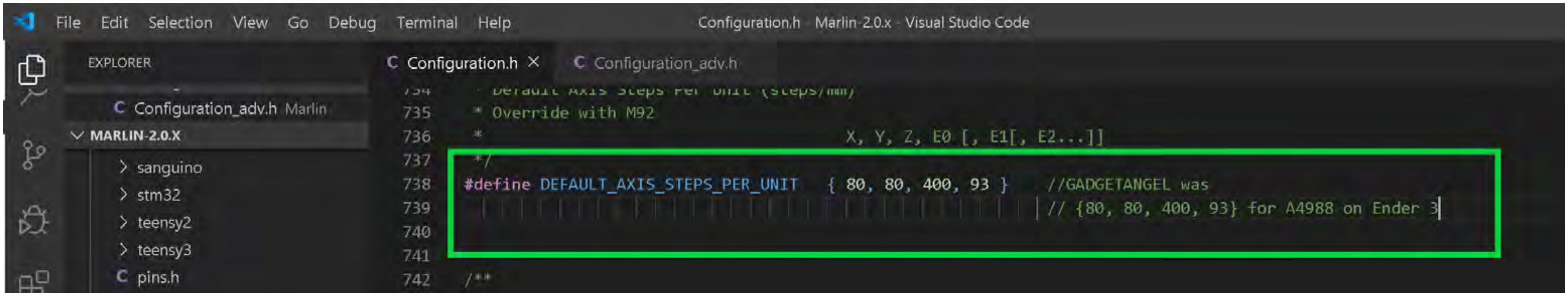




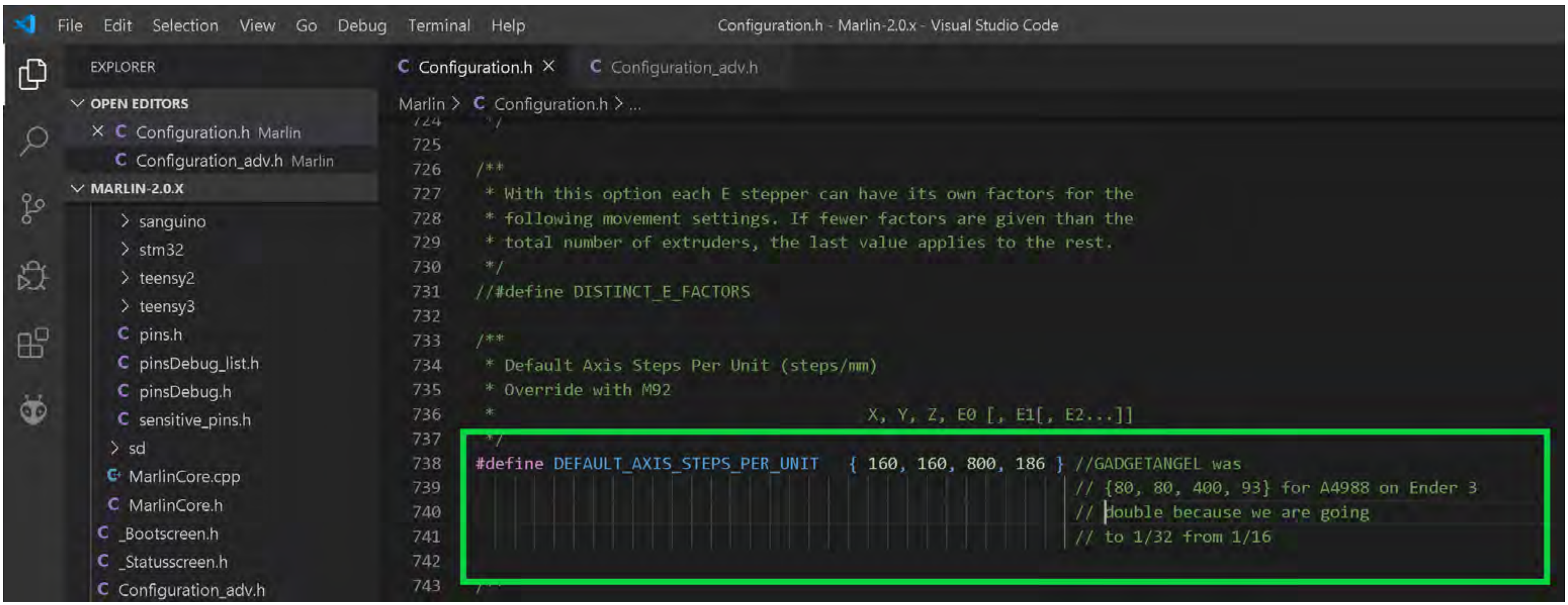
- Go to the next page.

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

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



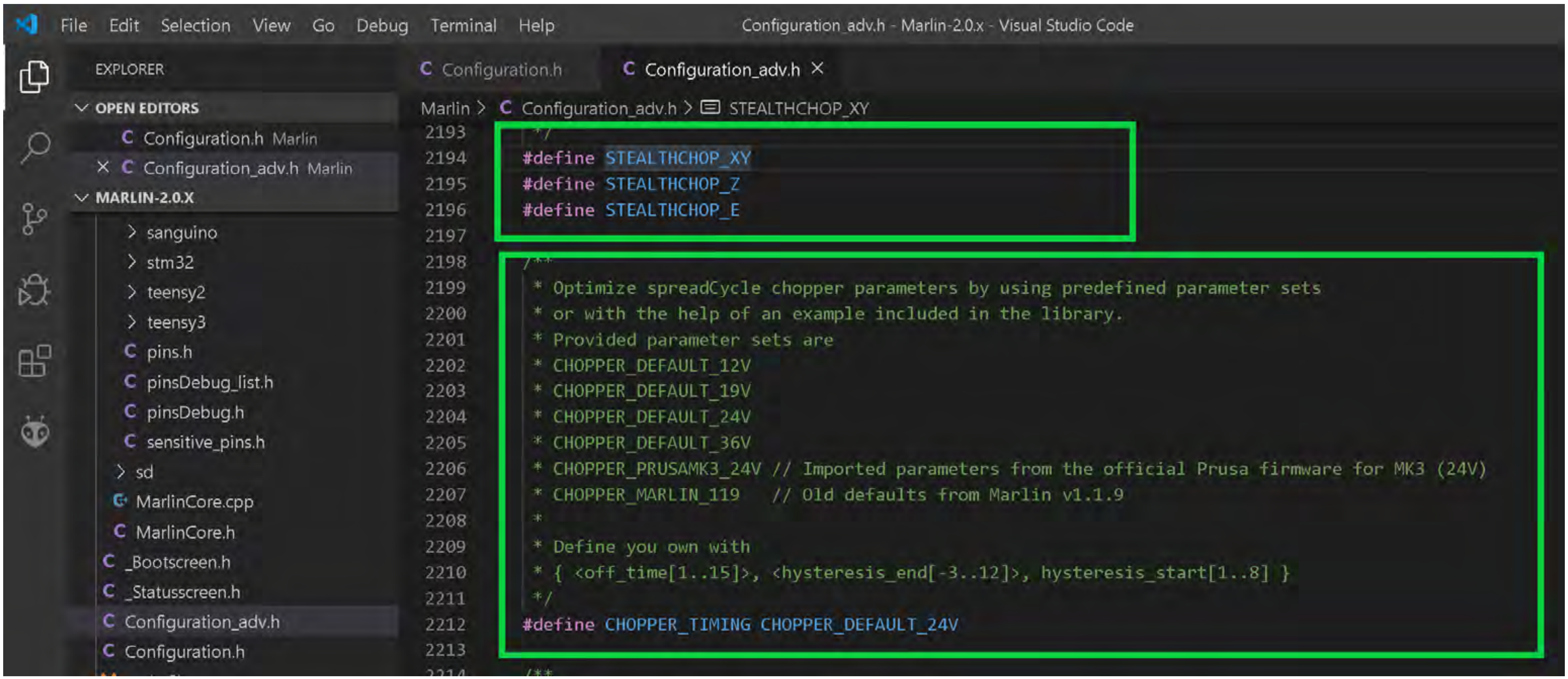
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



- Go to the next page.

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

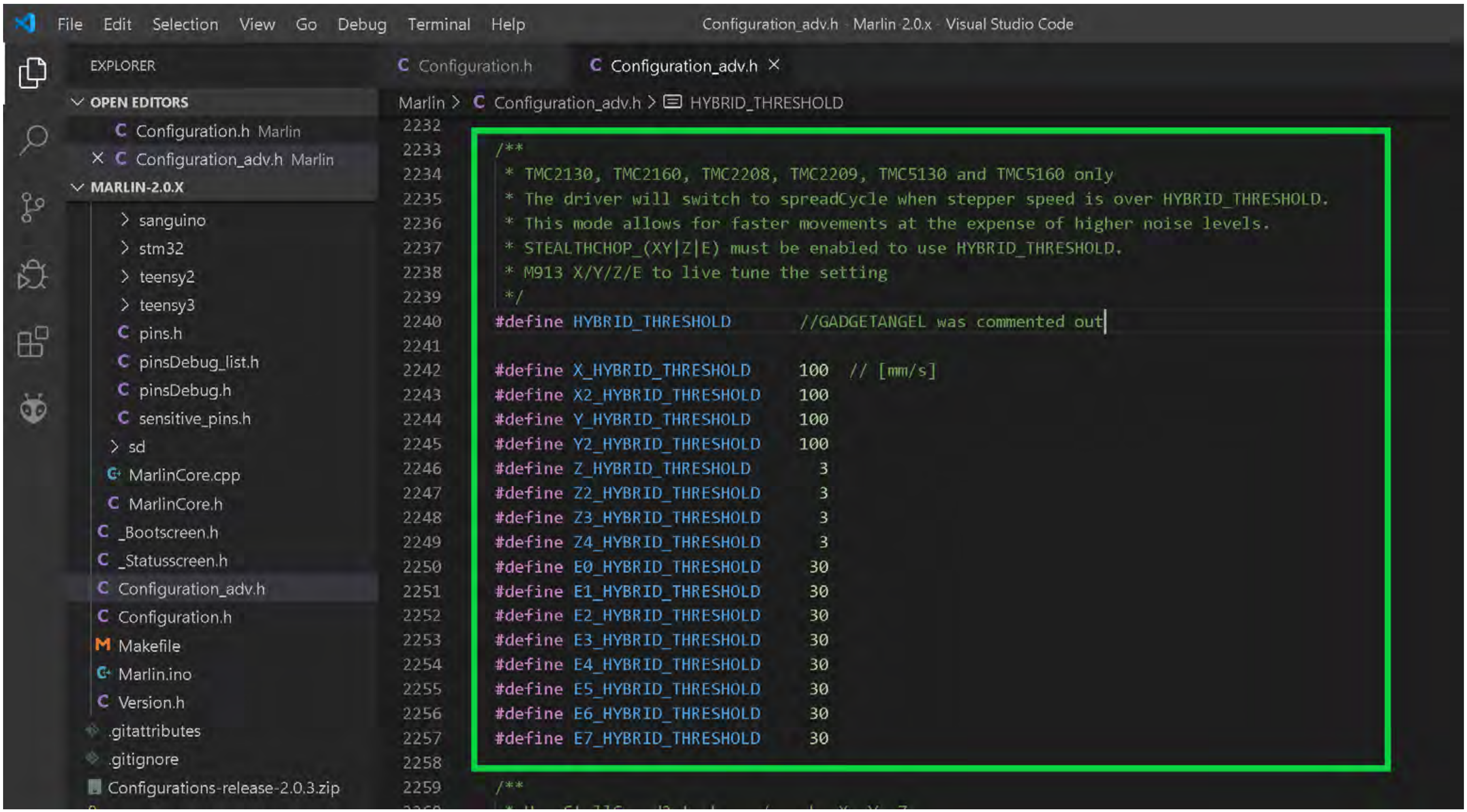
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

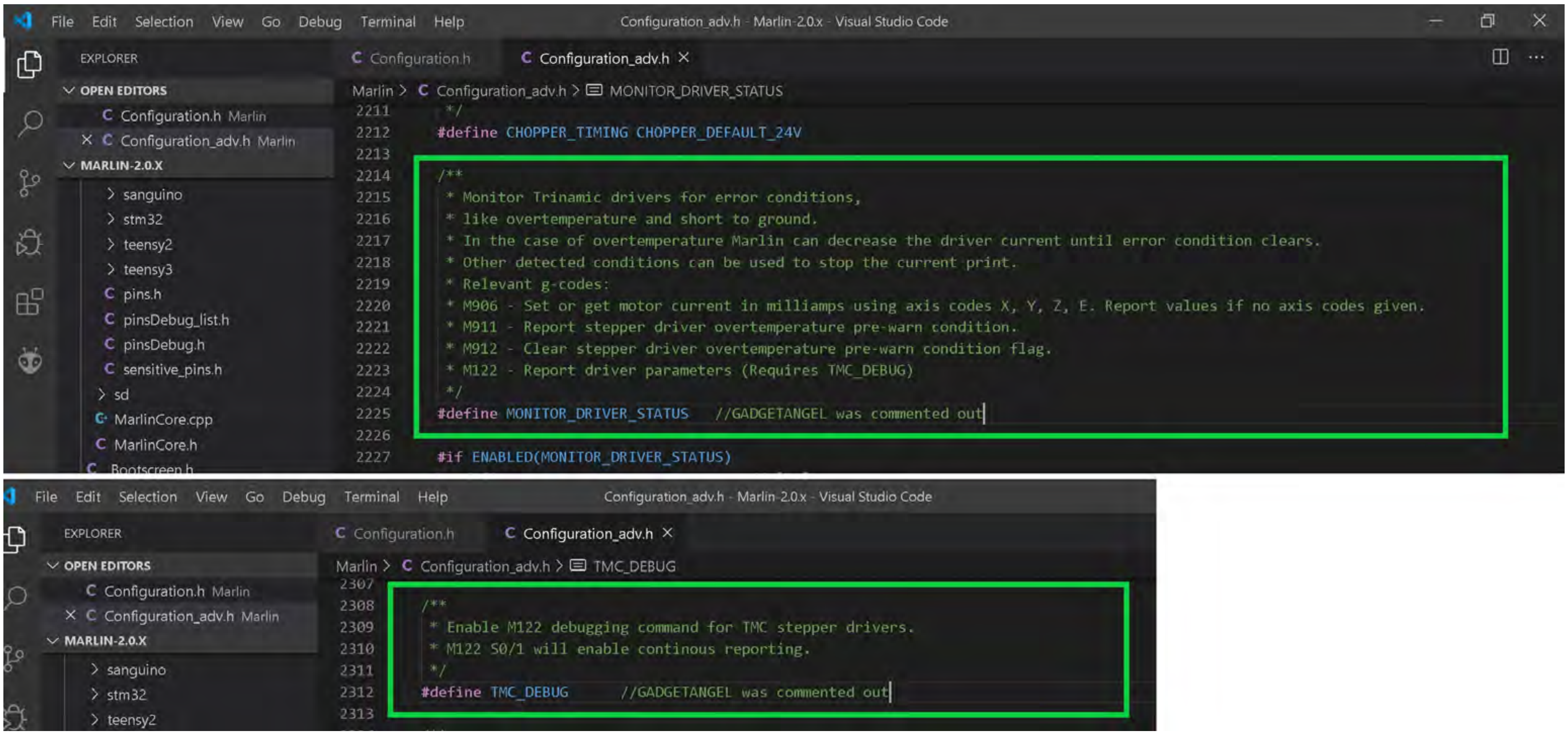
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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



- Go to the next page.

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

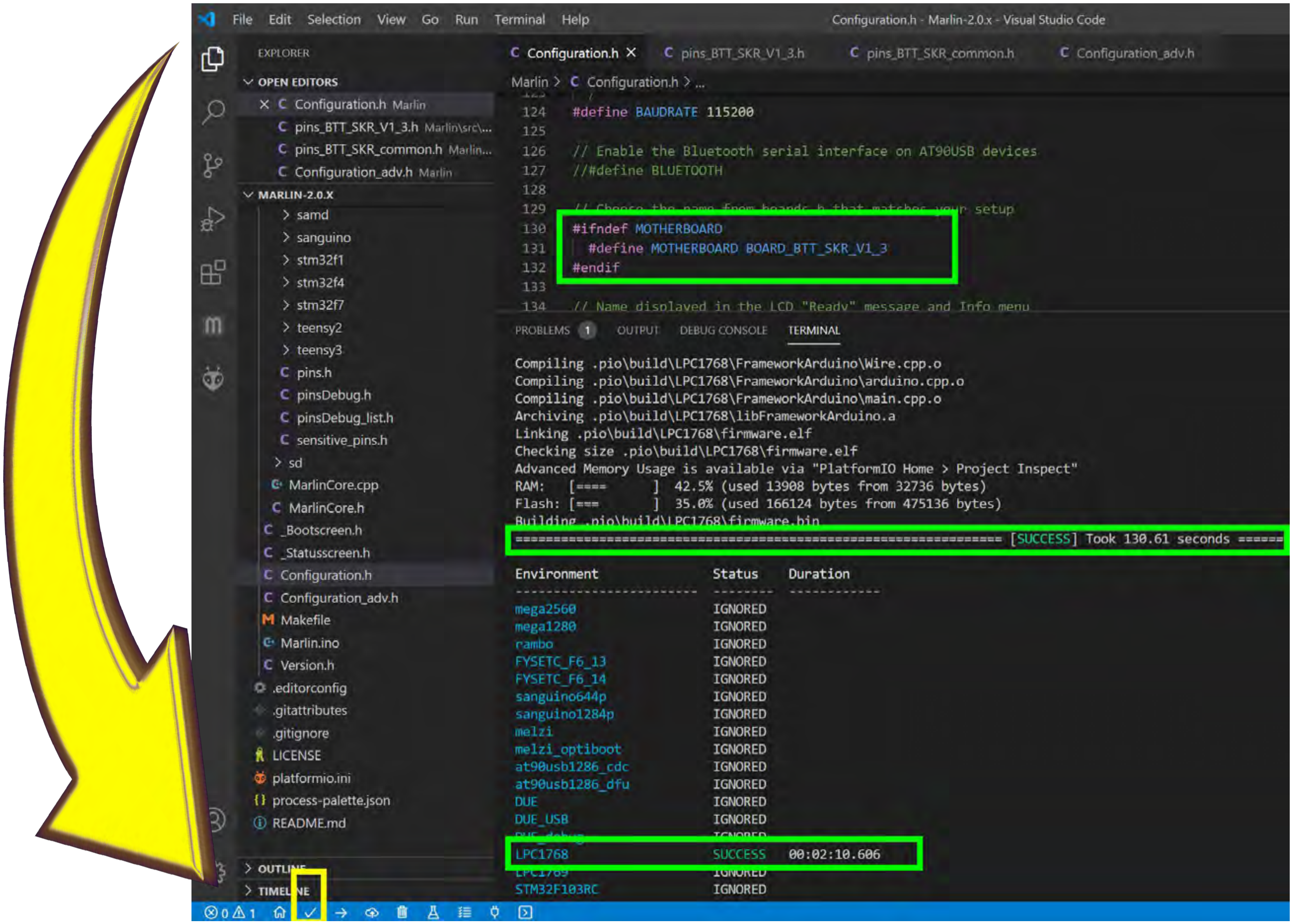
- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR_DRIVER_STATUS" and "TMC_DEBUG". "MONITOR_DRIVER_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC_DEBUG" will enable the M122 G-code command. You can read about these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



- Go to the next page.

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

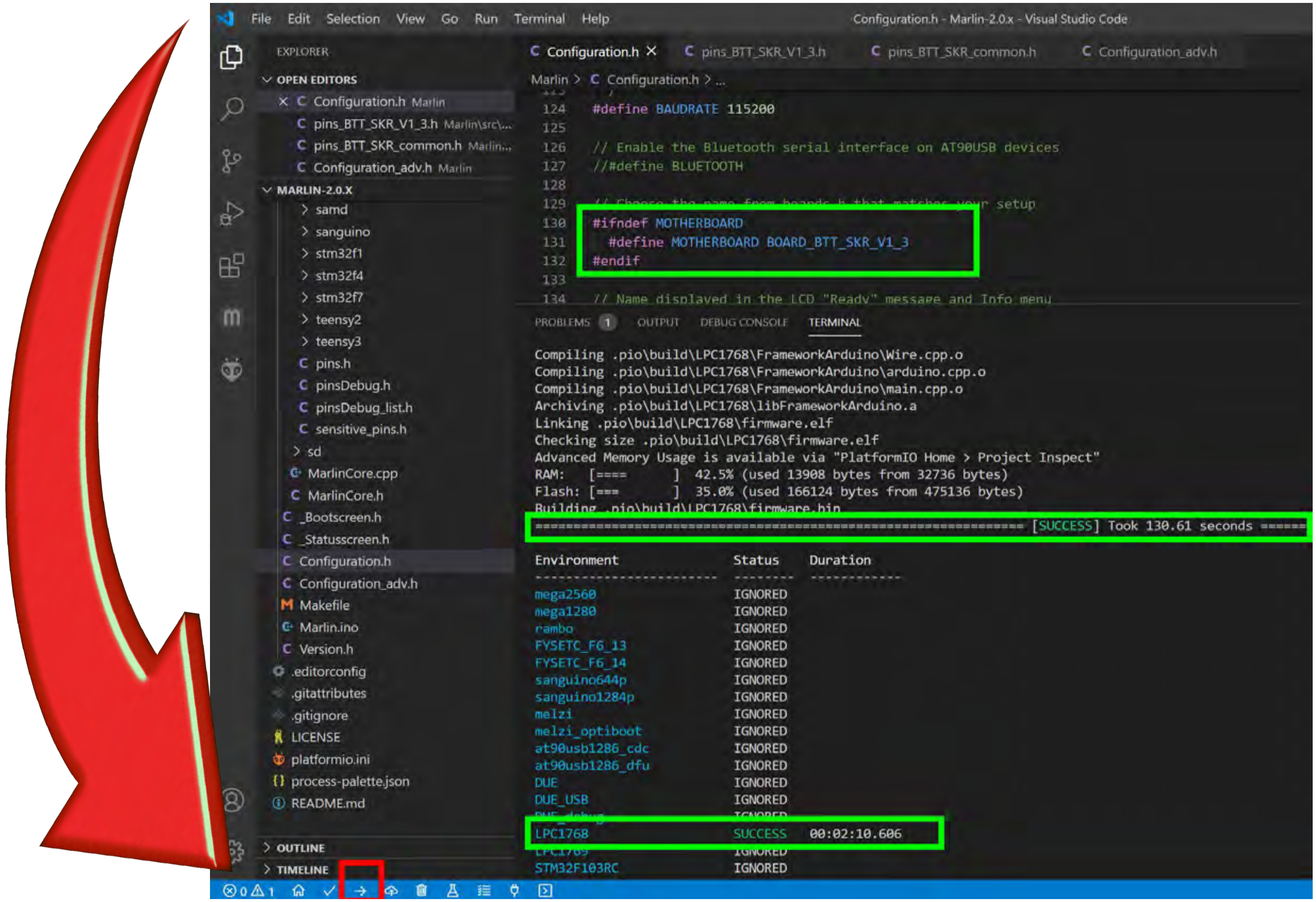
- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



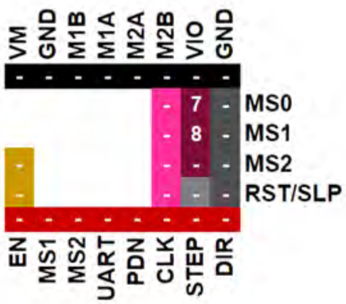
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

FYSETC TMC2208 V1.2

Stand-alone Mode

Stand-alone Mode

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

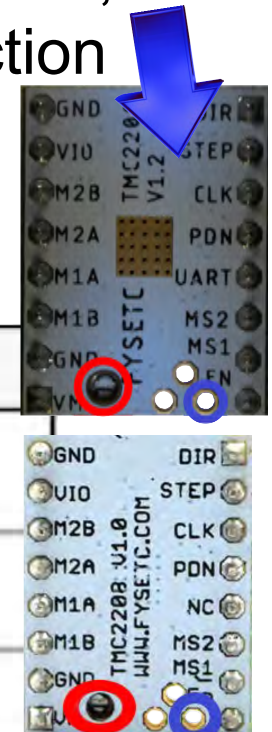
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

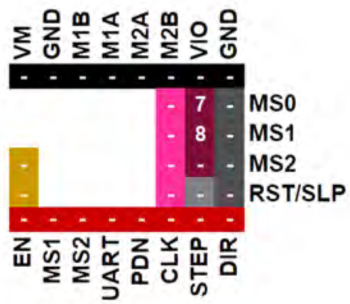
- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

FYSETC TMC2208 V1.2

Stand-alone Mode

Stand-alone Mode

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

STATUS:
Not Yet Tested

See Appendix D for legend

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low → No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High → Jumper Set

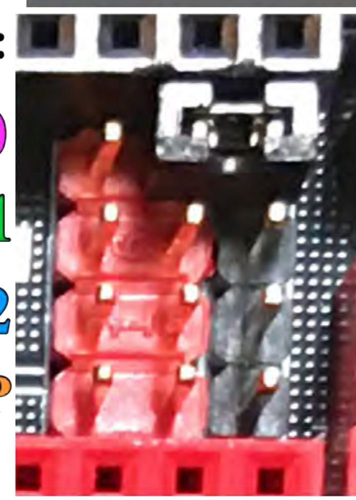
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

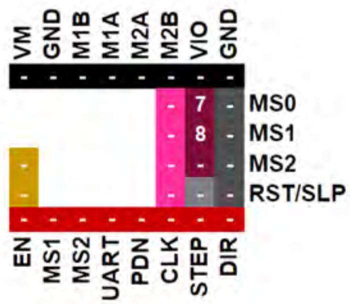
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

FYSETC TMC2208 V1.2

Stand-alone Mode

Stand-alone Mode

base of card



	0	1
8	0	0
2	1	0
4	0	1
16	1	1

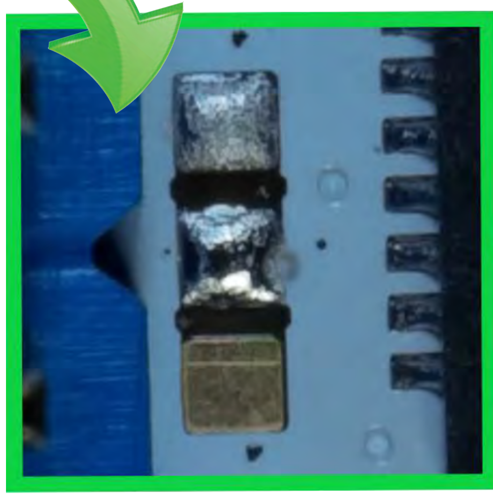
fans/heaters/usb side

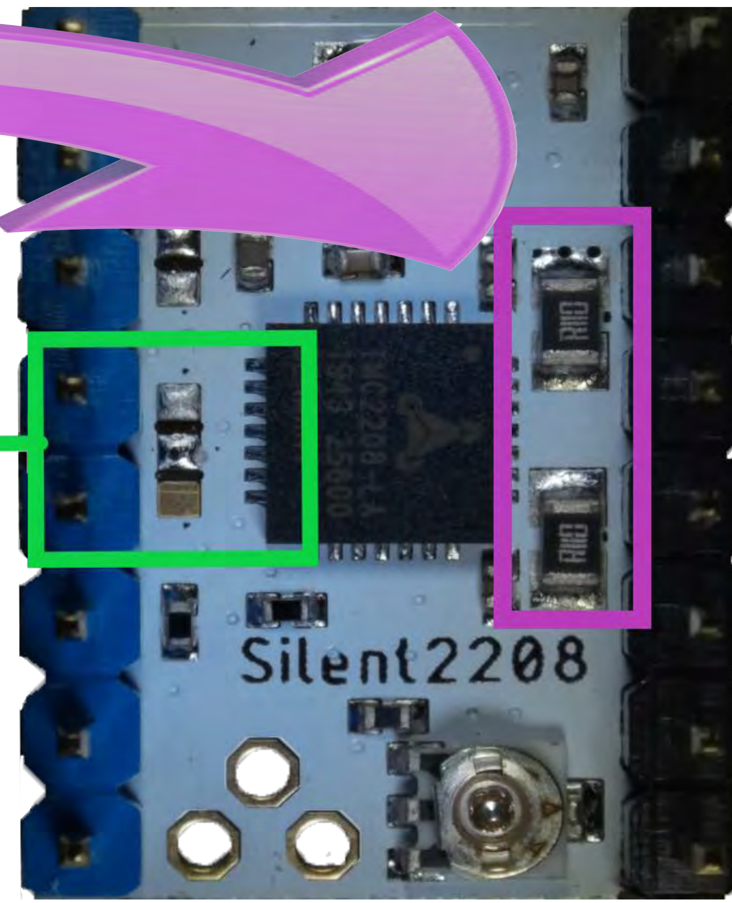
STATUS: Not Yet Tested

See Appendix D for legend

Note: To determine if your driver board is in UART mode, check the bottom of the driver board for three pads, located in the same position as shown in the picture below. There could be two or three pads located at this position (let's call it the "J2" position). To have the driver board in STANDALONE MODE, ALL the pads at "J2" MUST NOT be connected.

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





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

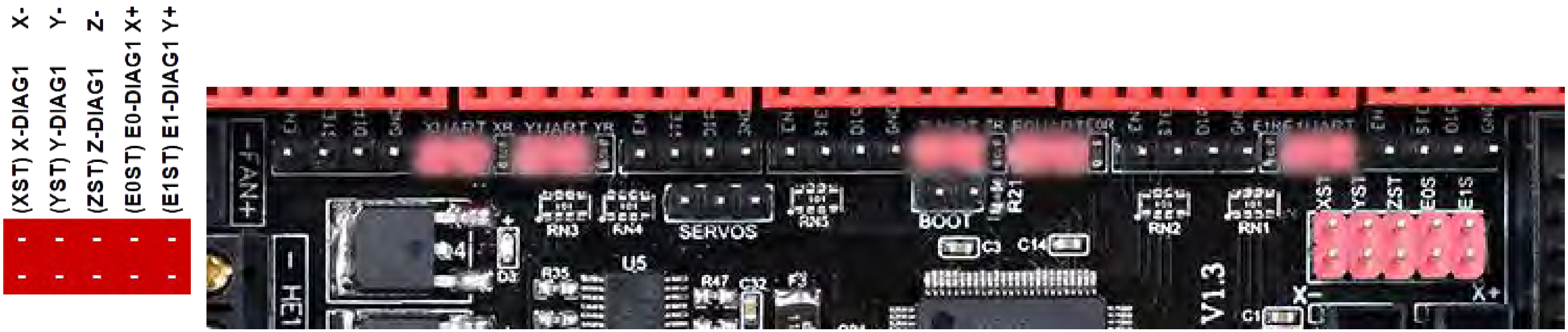
FYSETC TMC2208 V1.2 for Standalone Mode

Stand-alone Mode

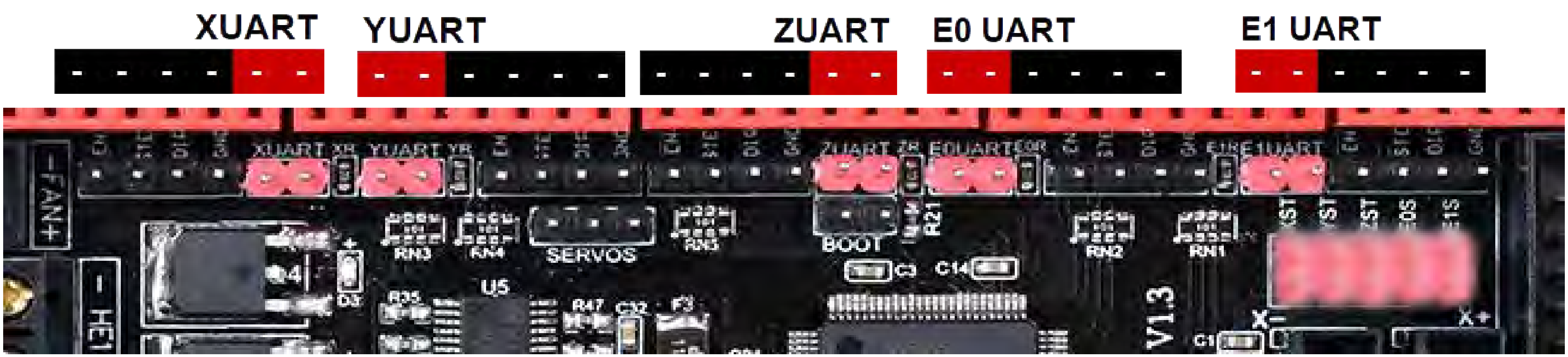
FYSETC TMC2208 V1.2

Stand-alone Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.



Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

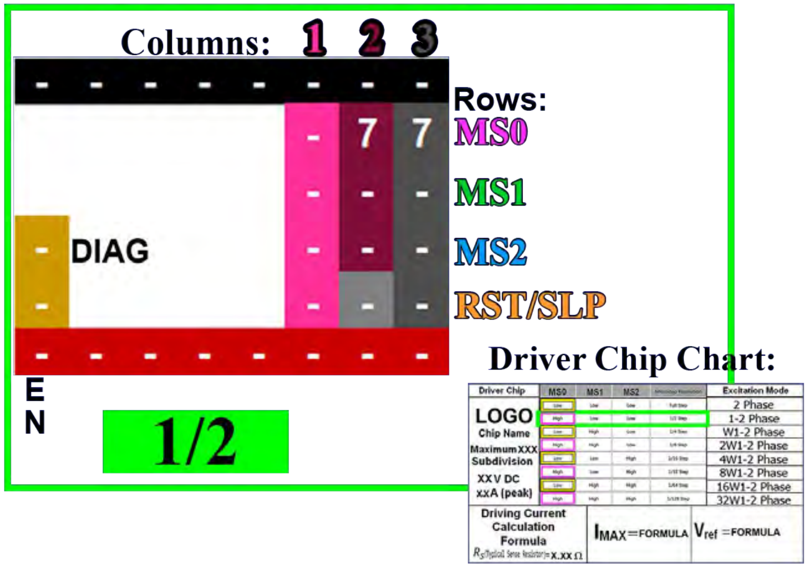


FYSETC TMC2208 V1.2

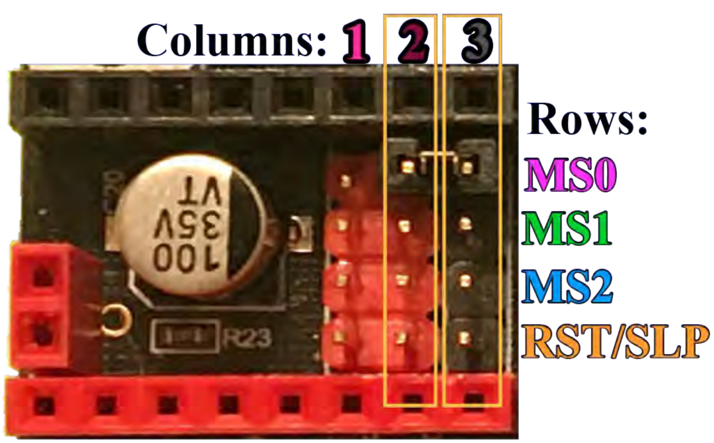
Stand-alone Mode

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

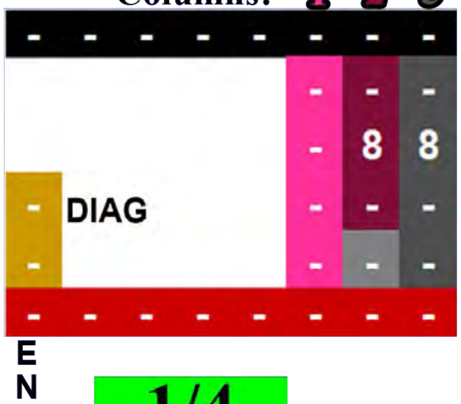
Meaning:

Driver Chip Chart:

Driver Chip Chart	Meaning
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

Driver Socket Representation

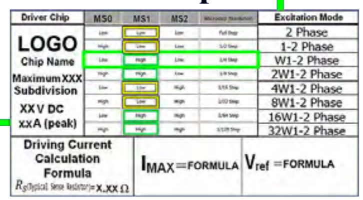
Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

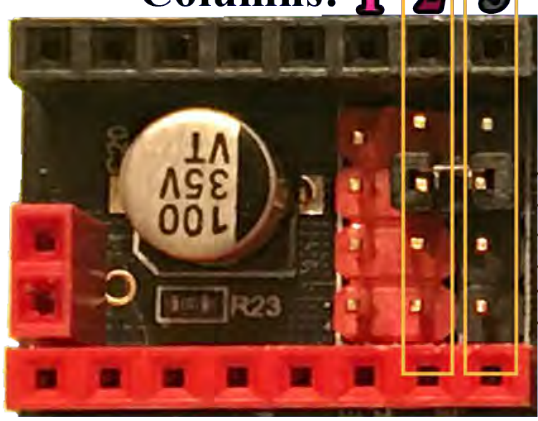
1/4

Driver Chip Chart:



High:

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Meaning:

Driver Chip Chart:		
High	→	set Jumper between column 2 and column 3 on the MS1 row
Low	→	No Jumper Set

FYSETC TMC2208 V1.2

Stand-alone Mode

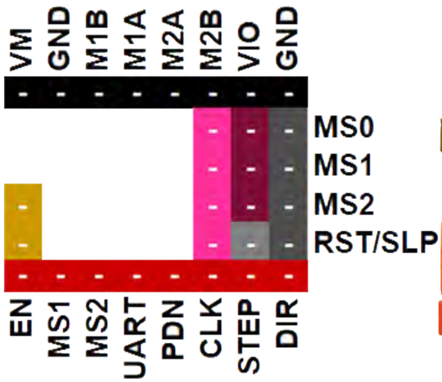
Stand-alone Mode

1 / 8

Interpolation:

1 / 256

StealthChop



See Appendix D for legend

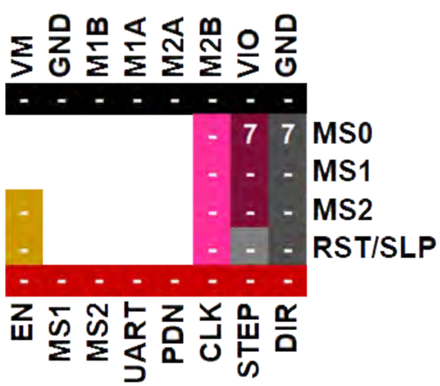
Stand-alone Mode

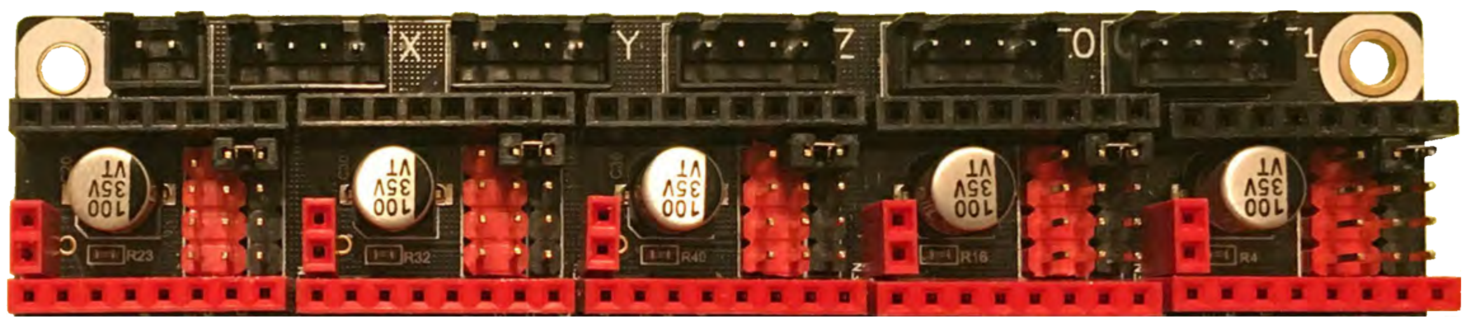
1 / 2

Interpolation:

1 / 256

StealthChop





See Appendix D for legend

Stand-alone Mode

FYSETC TMC2208 V1.2

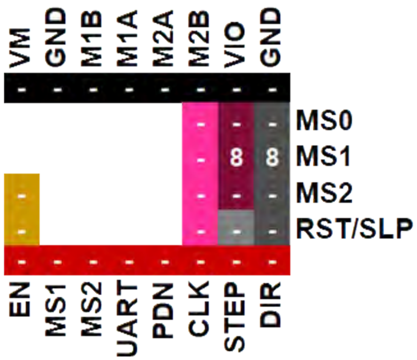
Stand-alone Mode

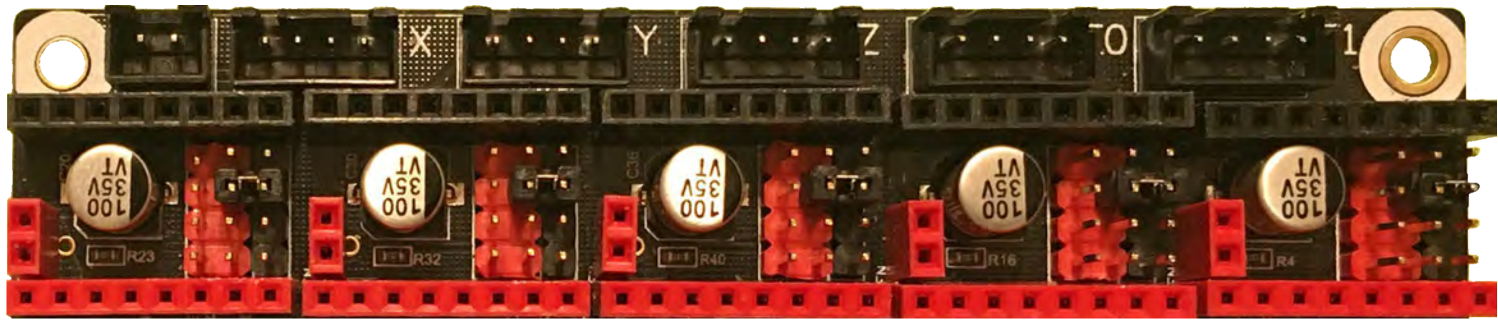
Stand-alone Mode

1 / 4

Interpolation:
1 / 256

StealthChop





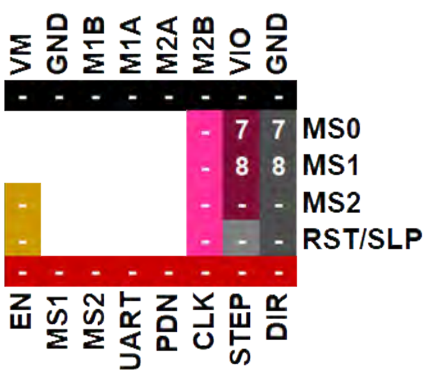
See Appendix D for legend

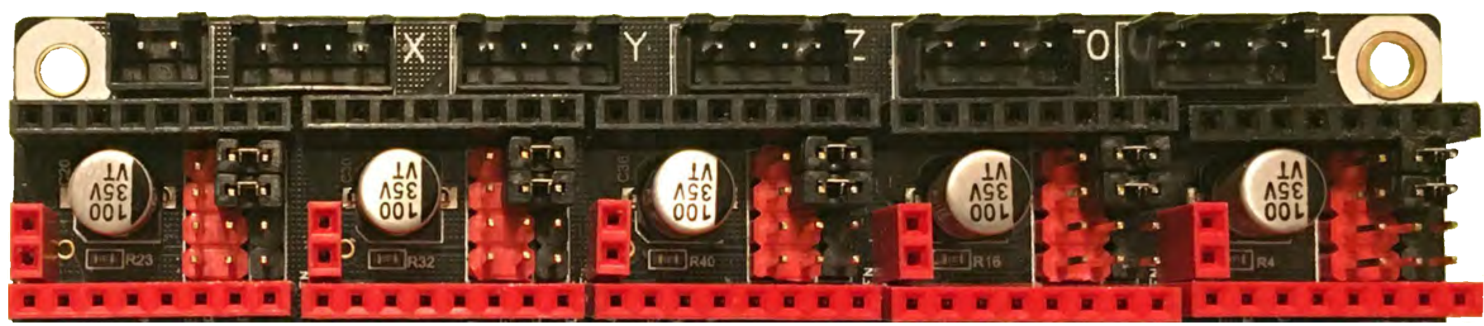
Stand-alone Mode

1 / 16

Interpolation:
1 / 256

StealthChop



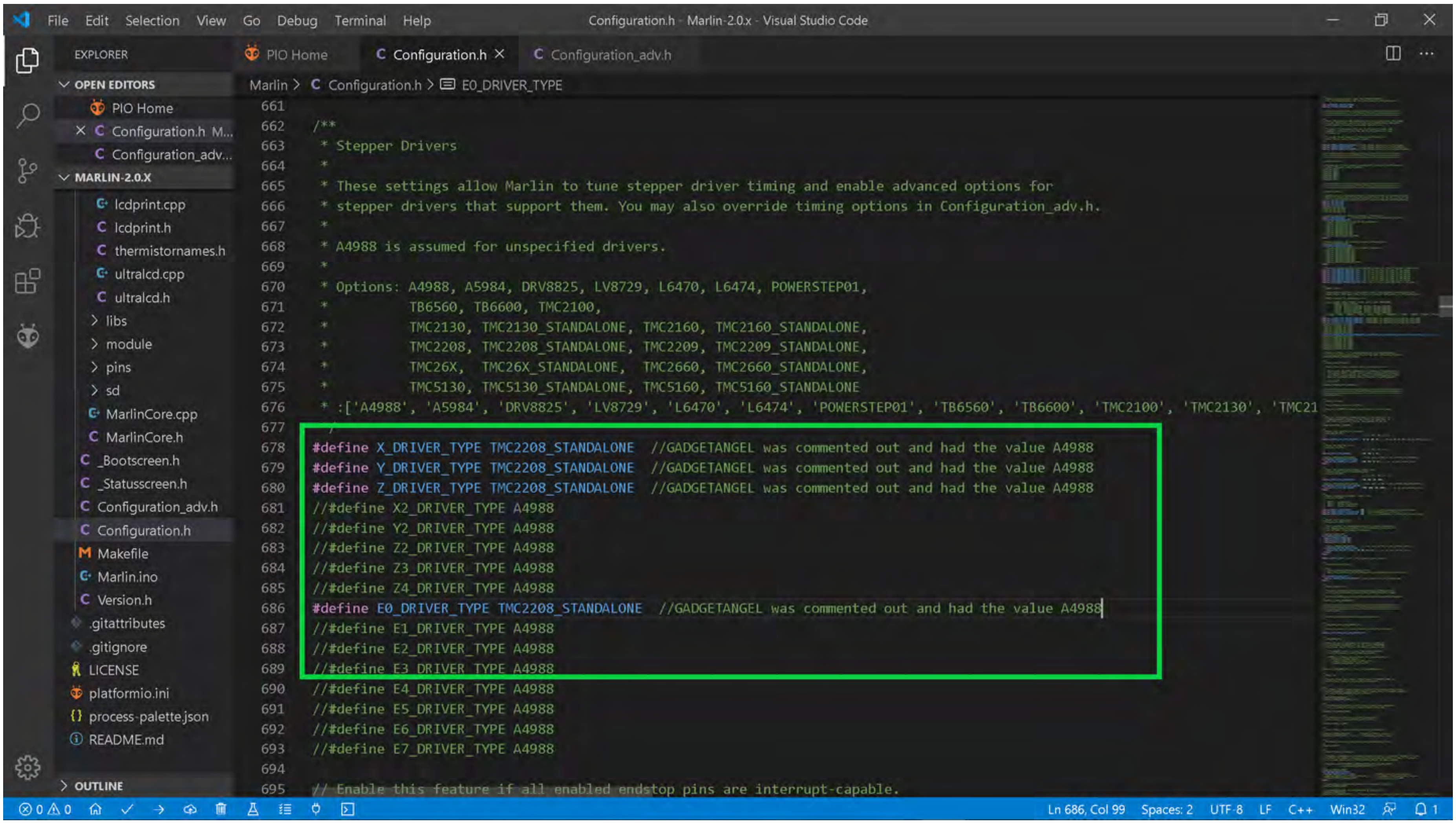


See Appendix D for legend

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

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

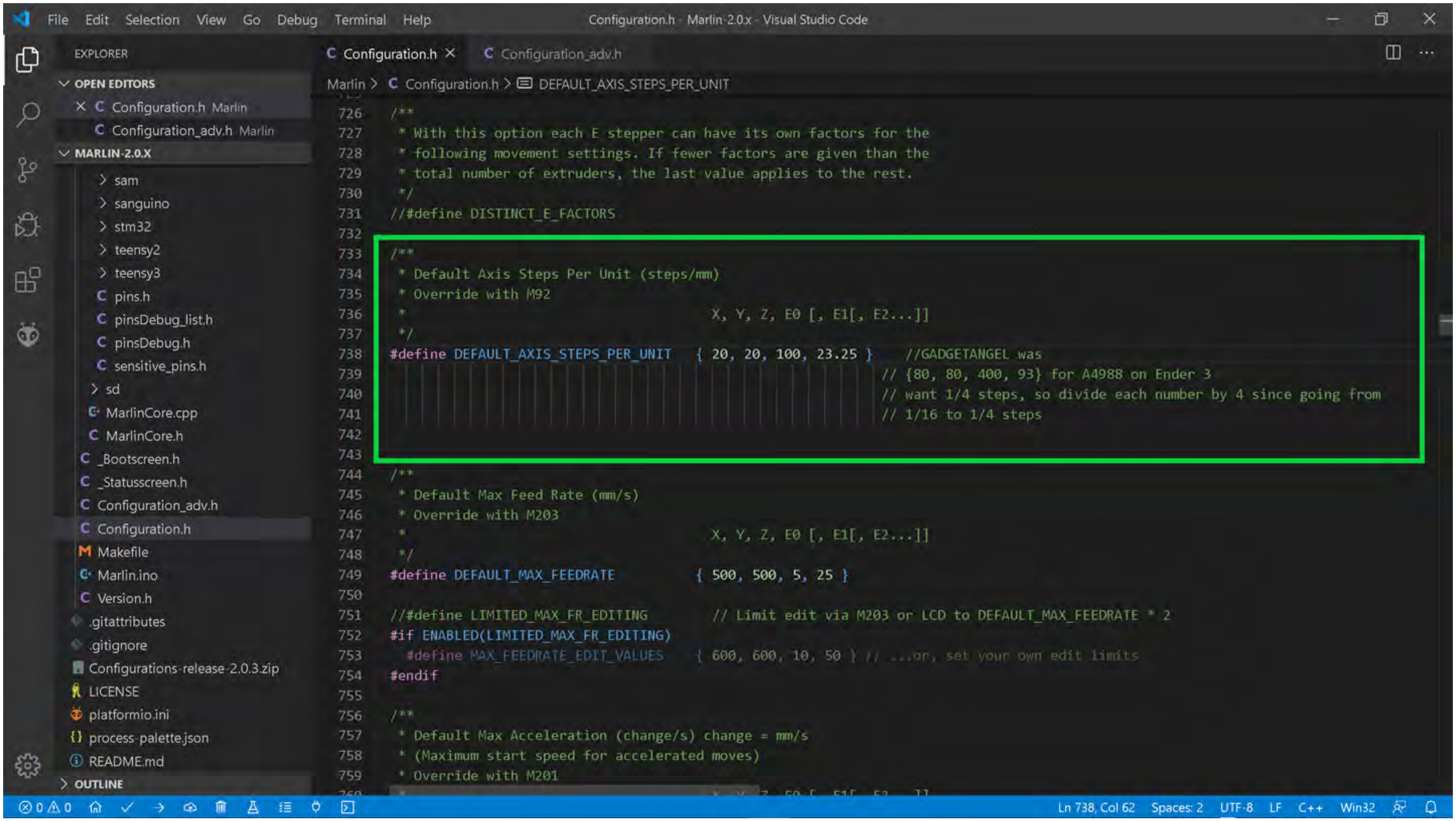
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

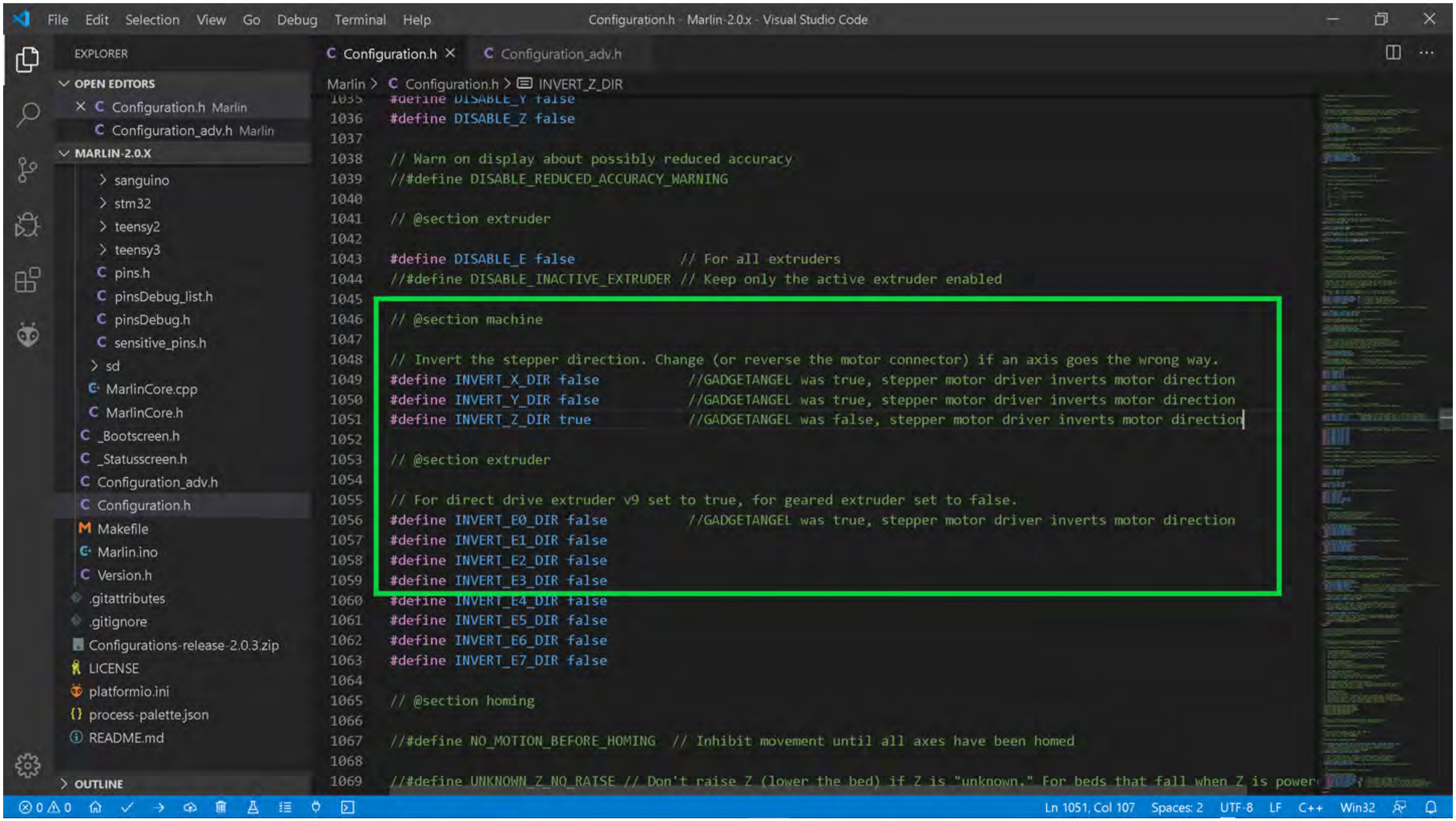
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

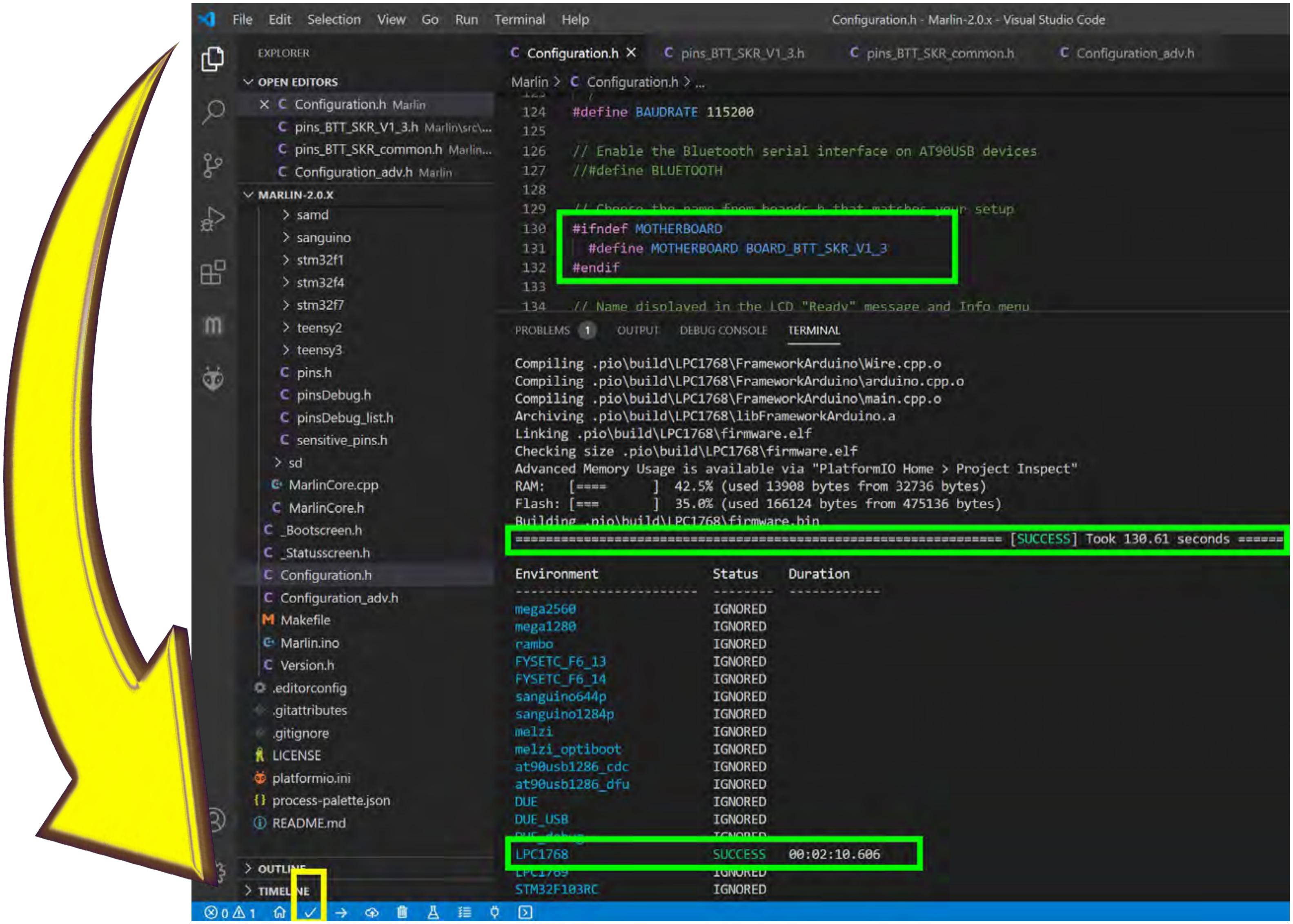
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

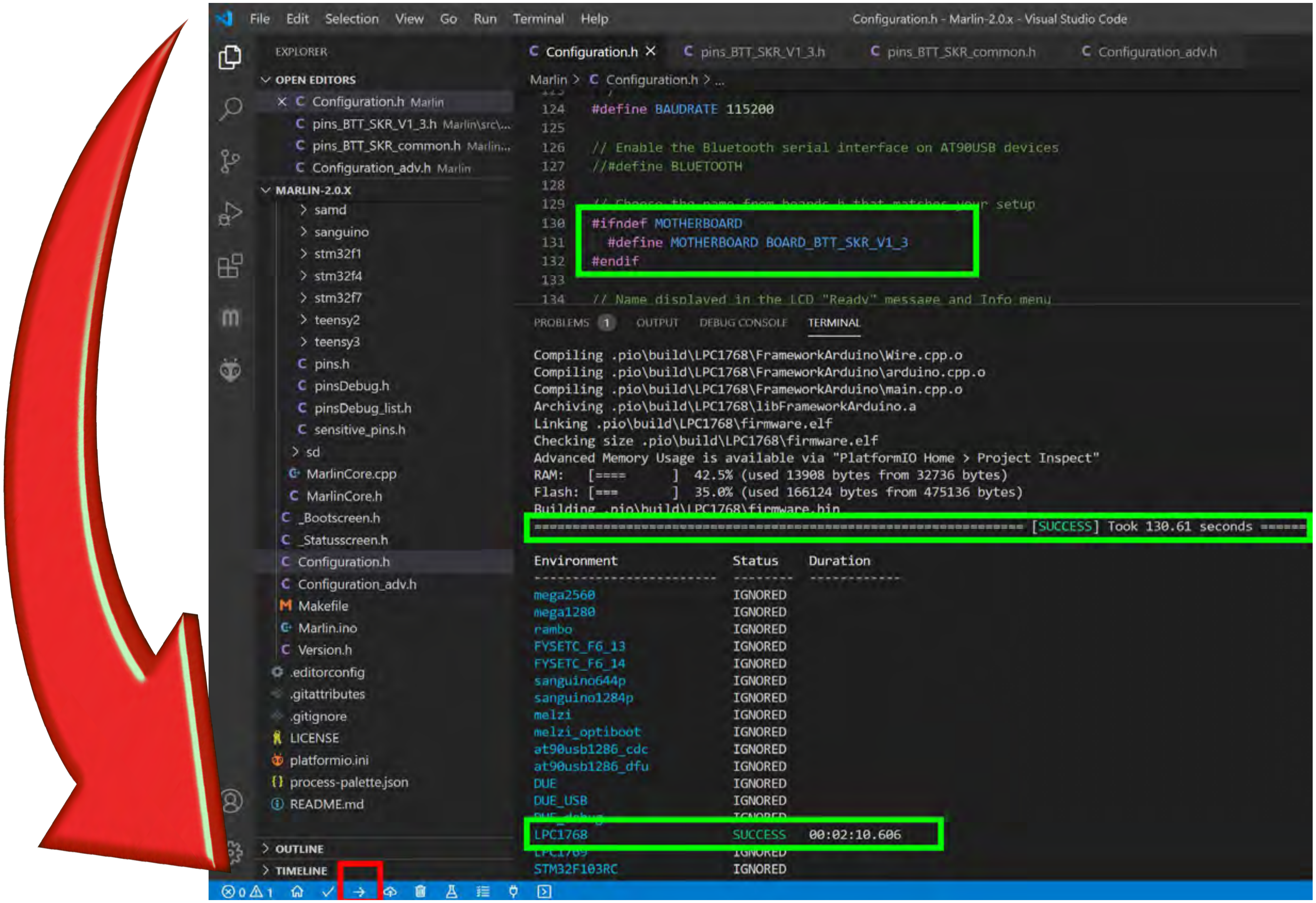
- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



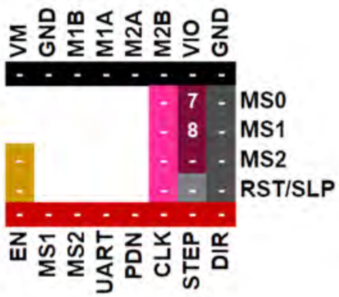
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card



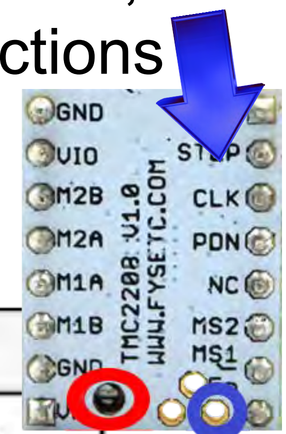
	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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	MS0	MS1	Steps	Interpolation	Mode
FYSETC **TMC2208** OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	1 / 8	1 / 256	spreadCycle
	High	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	1 / 256	spreadCycle
	High	High	1 / 16	1 / 256	spreadCycle
Driving Current Calculation Formula R_S(Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}*0.9286$ See Appendix B #3. Use 50% to 90% as shown below: $I_{MAX}=(V_{ref}*0.9286)*0.90$			$V_{ref}=I_{MAX}*1.0769$ See Appendix B #3. Use 50% to 90% as shown below: $V_{ref}=(I_{MAX}*1.0769)*0.90$	

- **See next page for the LEGEND that belongs to the above Driver Chip Chart.**

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card

	0	1
8	0	0
2	1	0
4	0	1
16	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC x.xA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S(Typical Sense Resistor)= x.xx Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA			

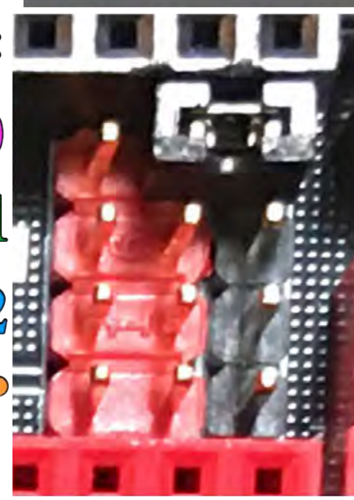
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

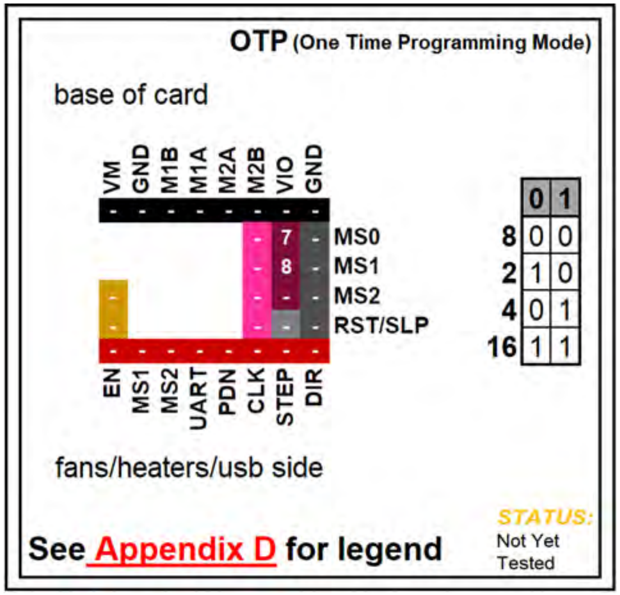
MS2 SET HIGH

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

FYSETC TMC2208 V1.2

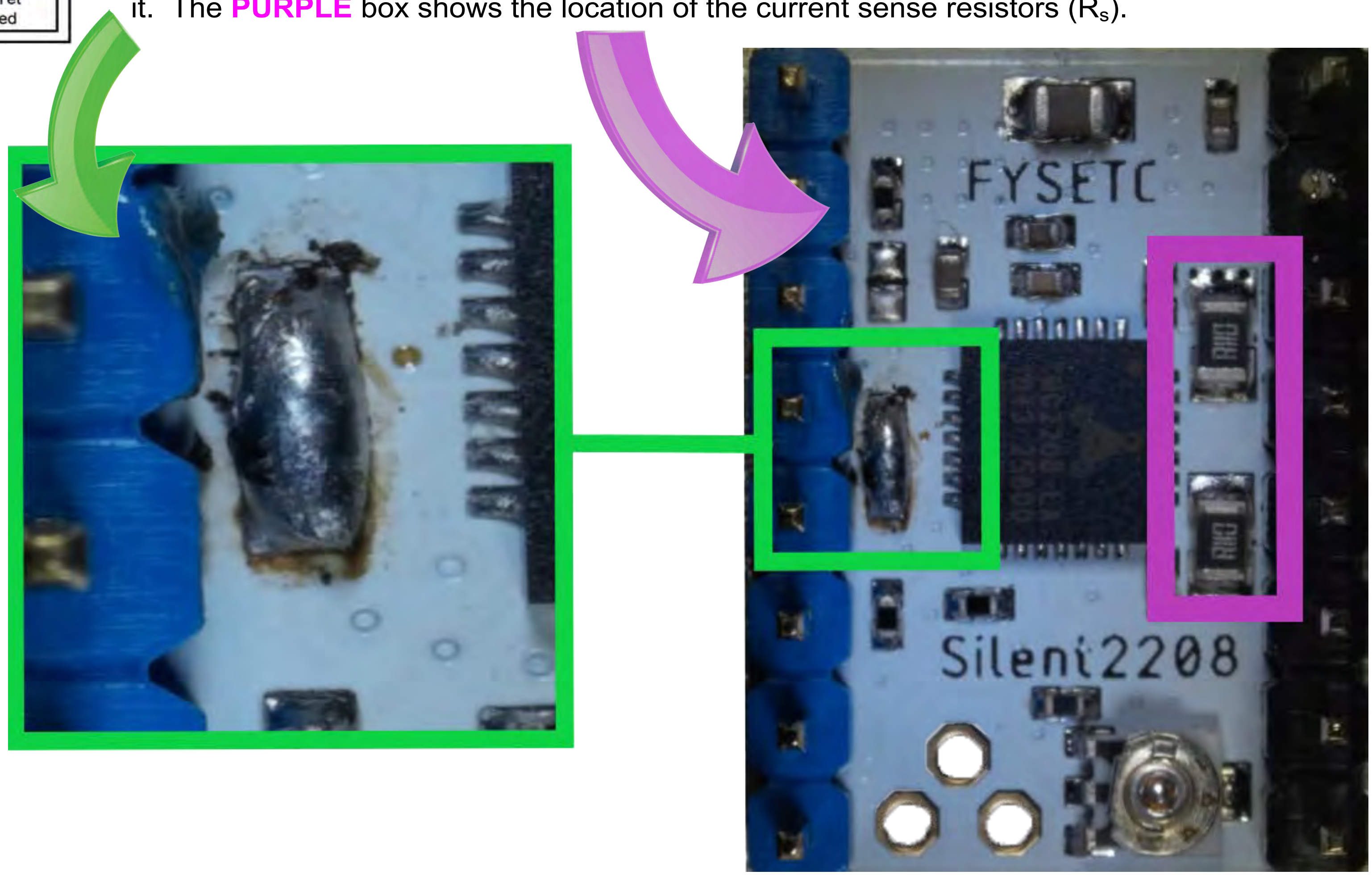
One Time Programming (OTP) Mode



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

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

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

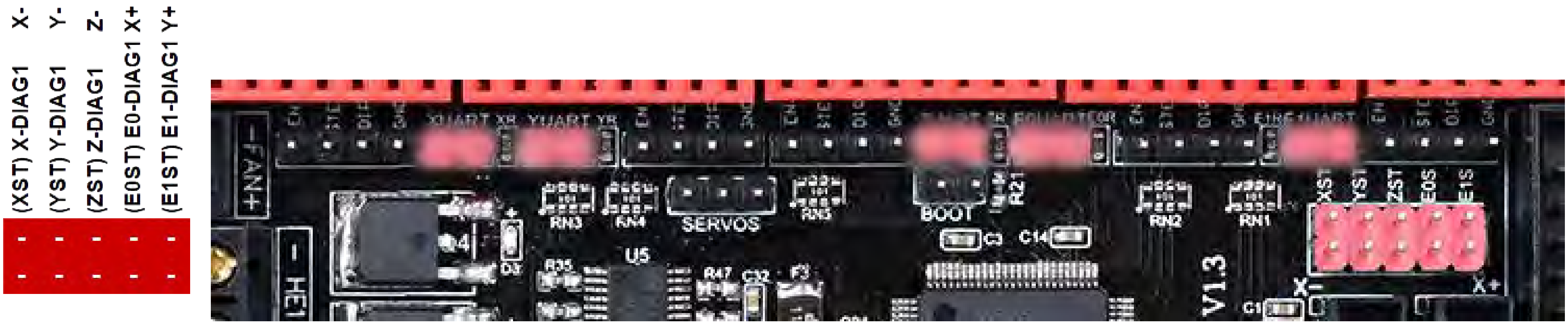


OTP (One Time Programming) Mode

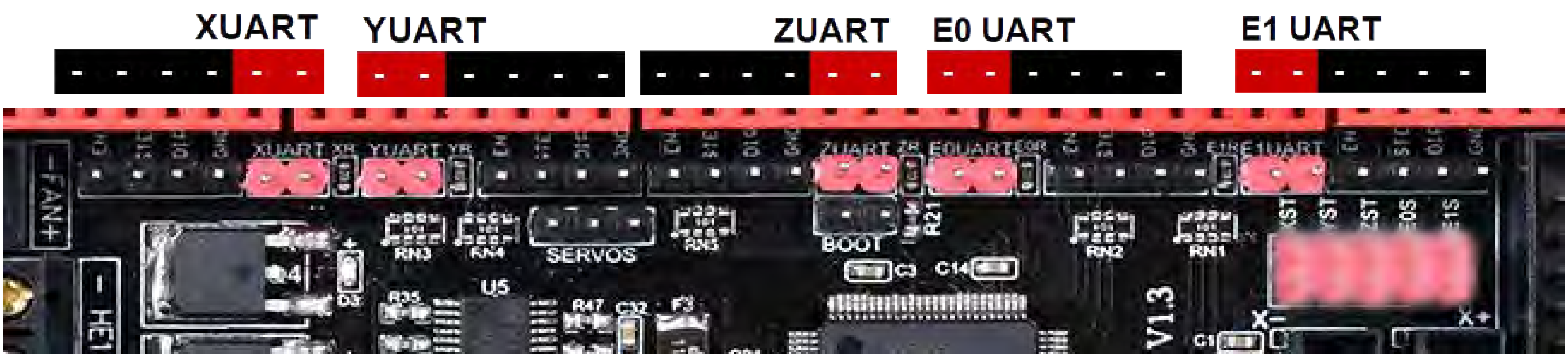
FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation.



Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation.

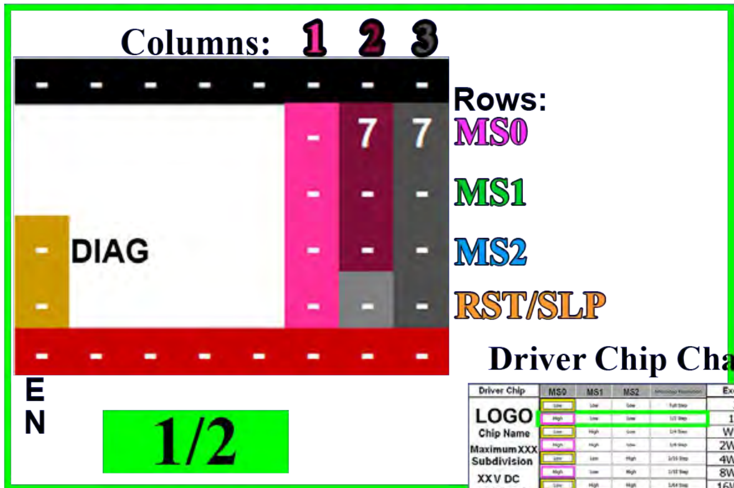


FYSETC TMC2208 V1.2

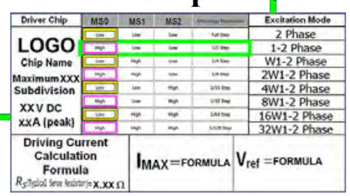
One Time Programming (OTP) Mode

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



Driver Chip Chart:

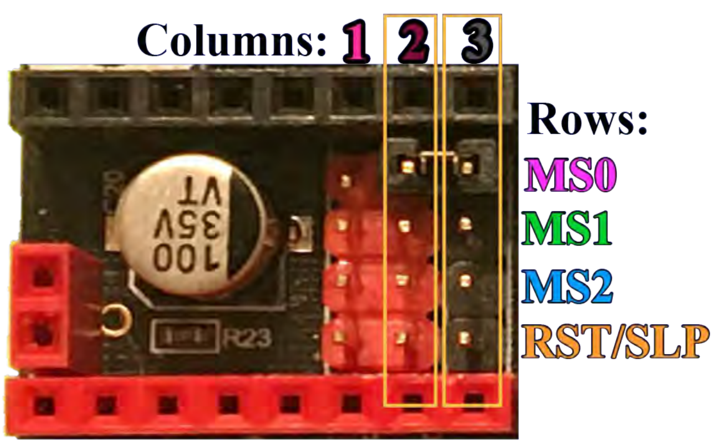


MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

High:

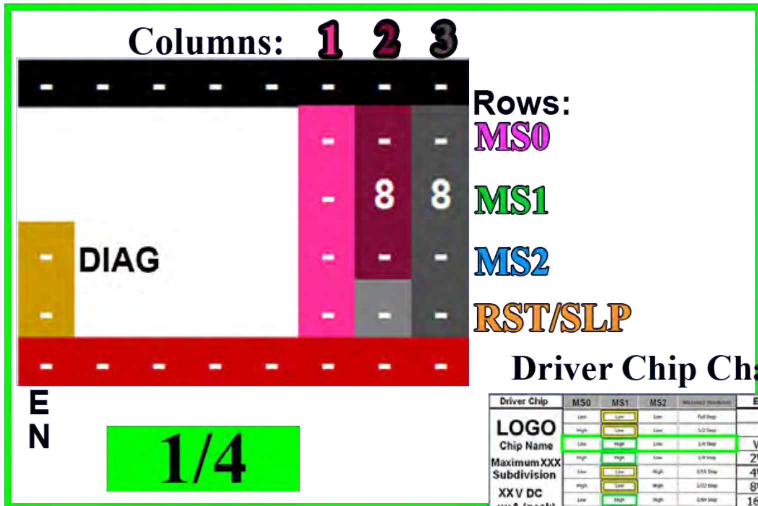


Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	No Jumper Set

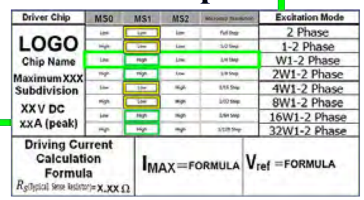
OTP (One Time Programming) Mode

One Time Programming (OTP) Mode

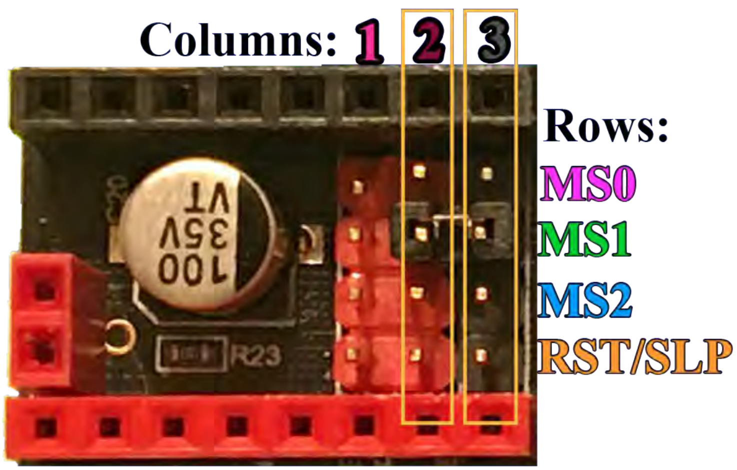
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Meaning:

Driver Chip Chart:		
High	→	set Jumper between column 2 and column 3 on the MS1 row
Low	→	No Jumper Set

OTP (One Time Programming) Mode

FYSETC TMC2208 V1.2

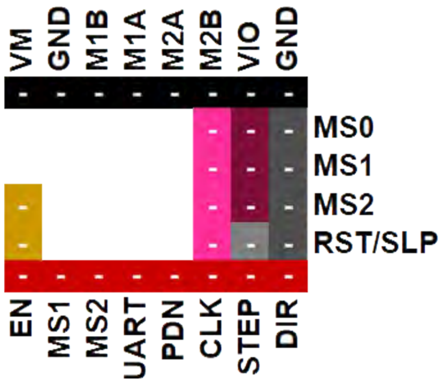
One Time Programming (OTP) Mode

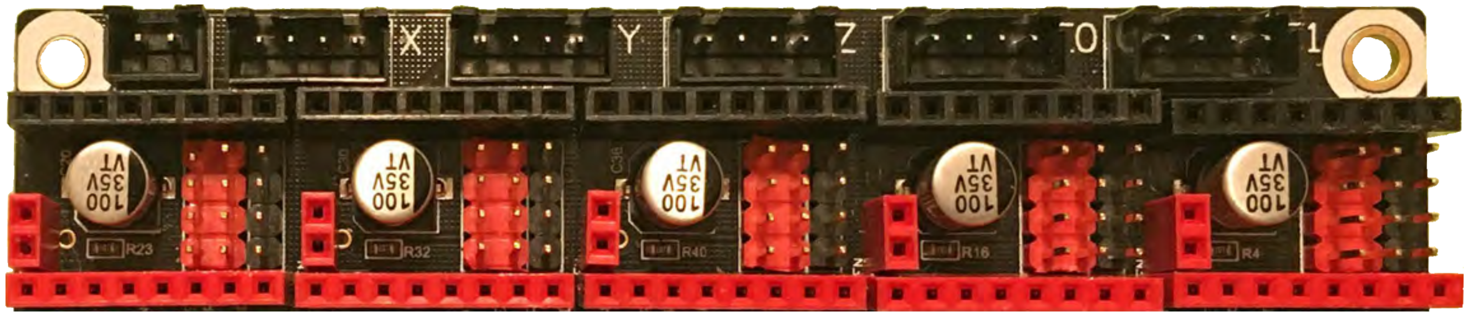
OTP

1 / 8

Interpolation:
1 / 256

SpreadCycle





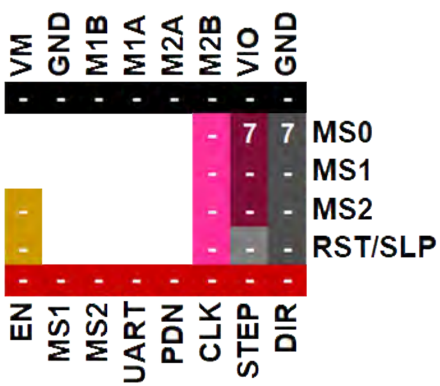
See Appendix D for legend

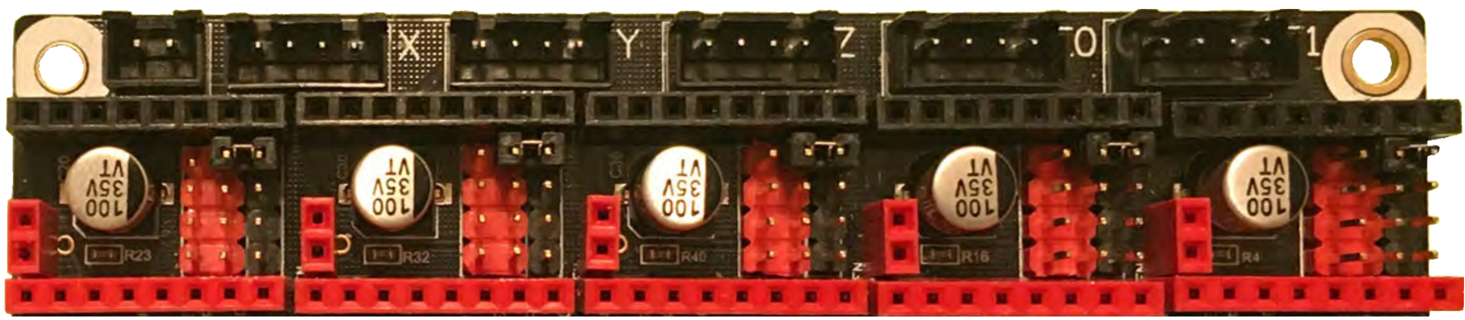
OTP

1 / 2

Interpolation:
1 / 256

SpreadCycle



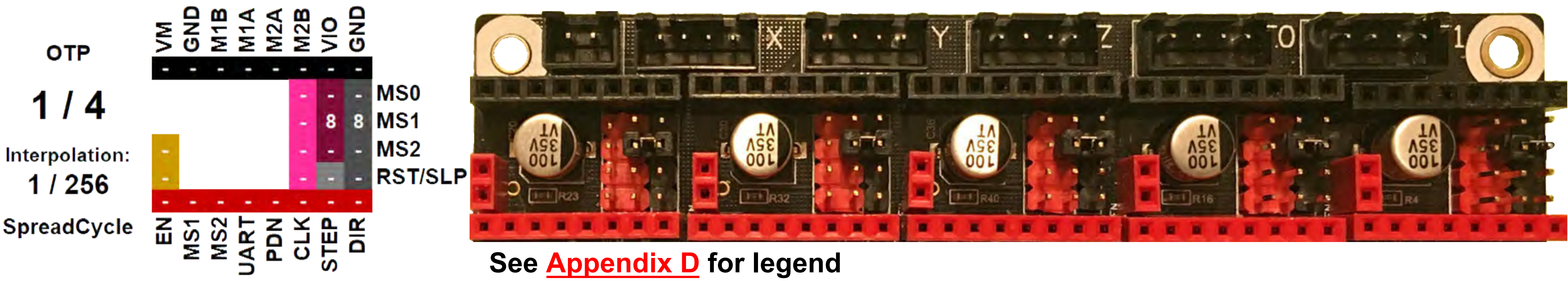


See Appendix D for legend

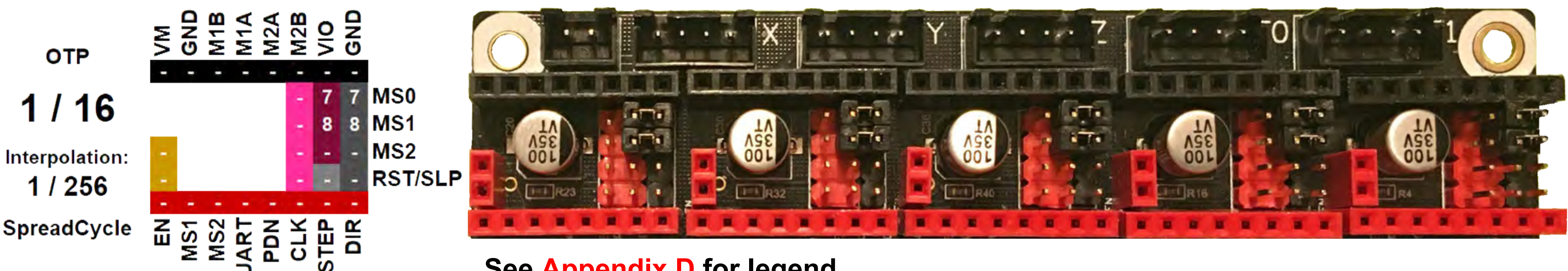
OTP (One Time Programming) Mode

FYSETC TMC2208 V1.2

One Time Programming (OTP) Mode



See Appendix D for legend

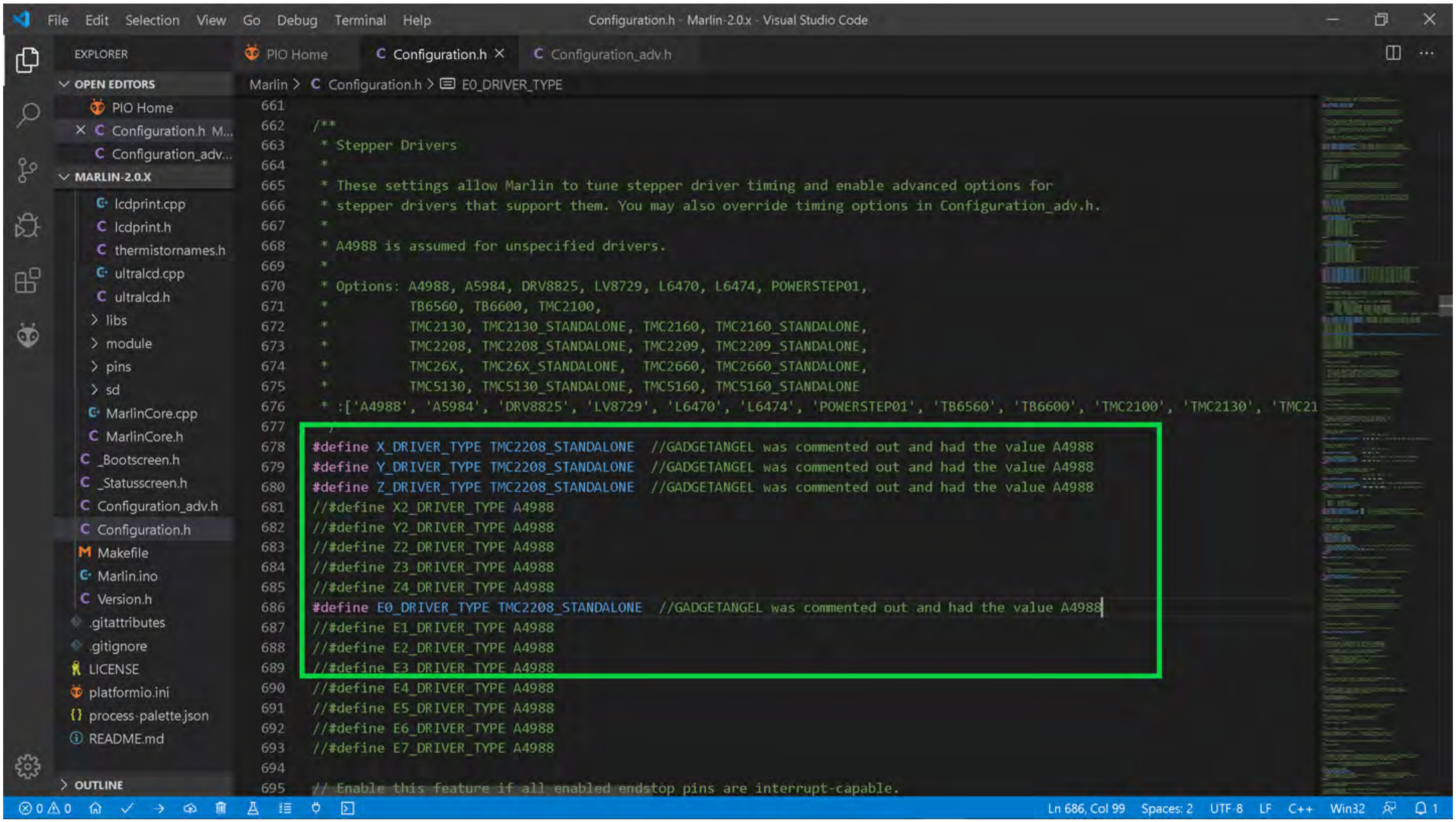


See Appendix D for legend

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

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

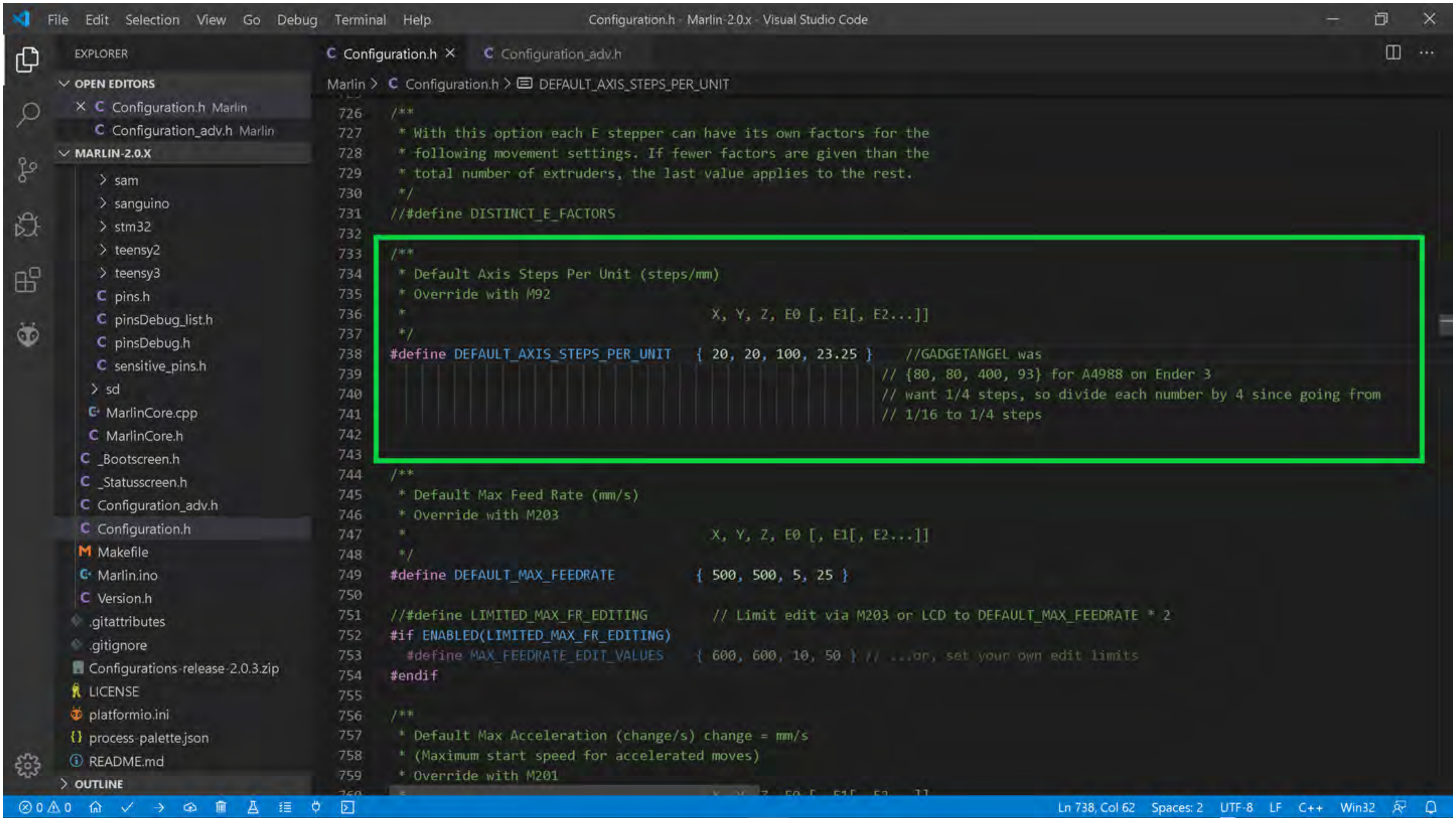
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "//" appear at the beginning of a line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



- Go to the next page.

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

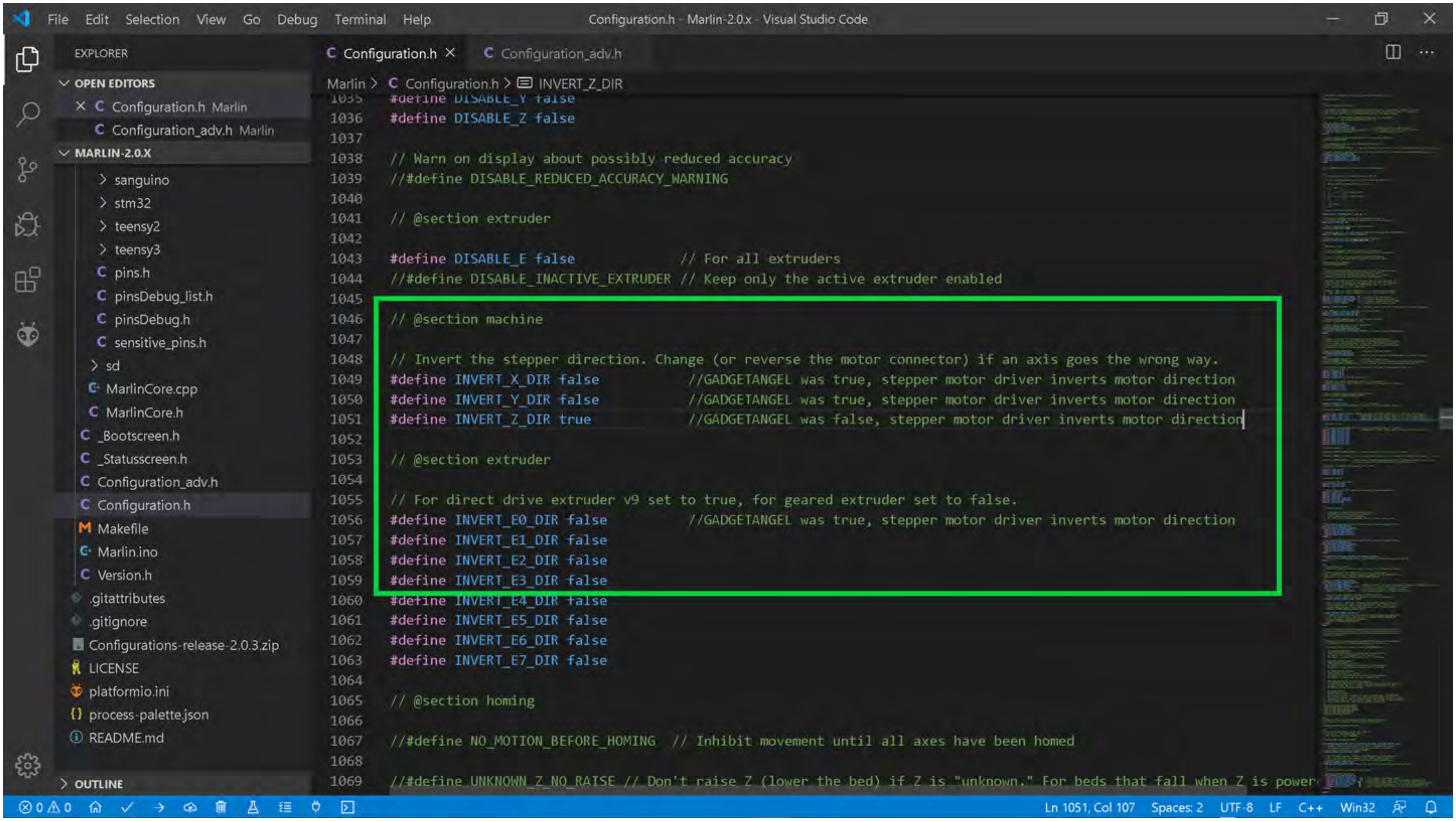
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

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

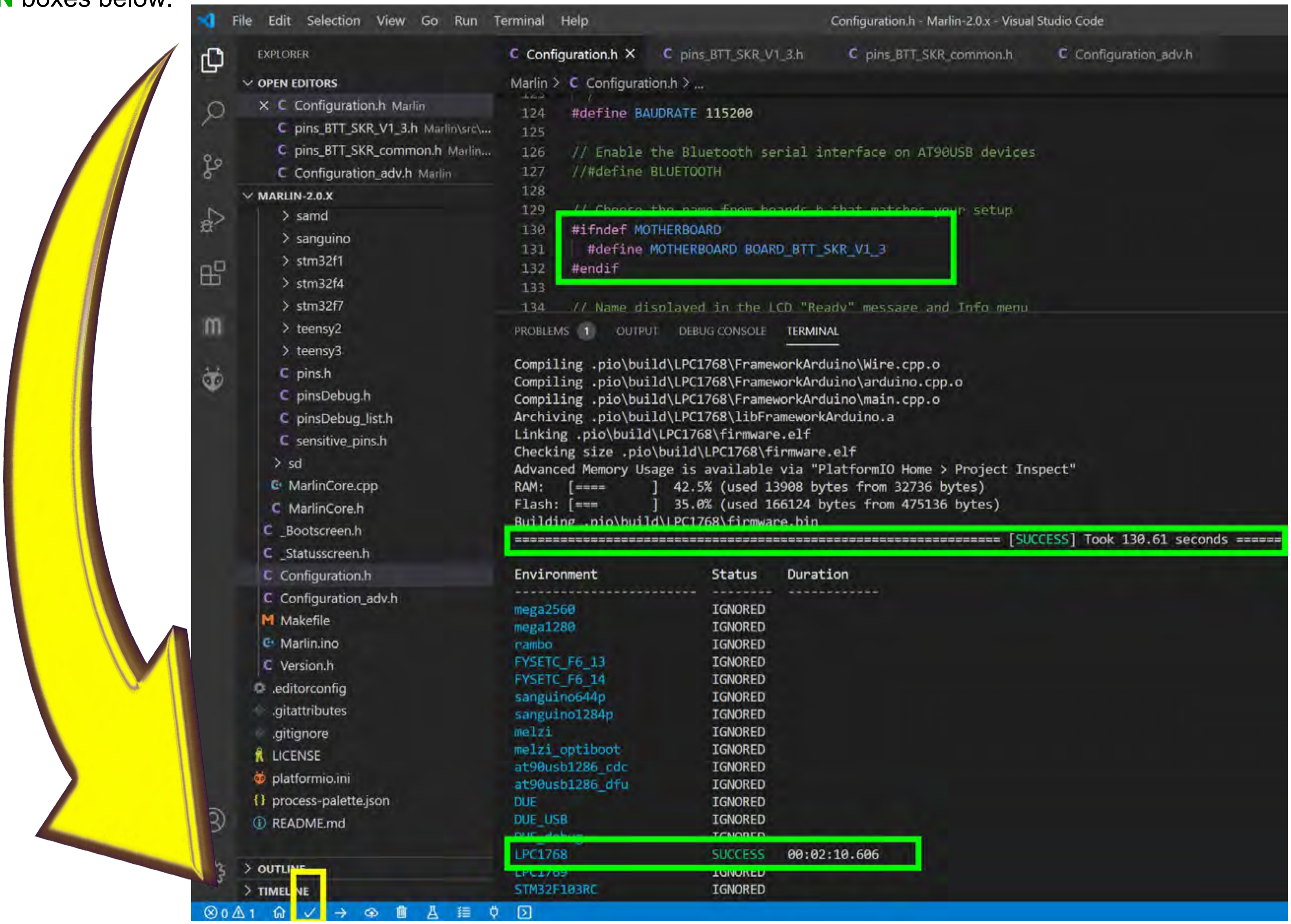
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

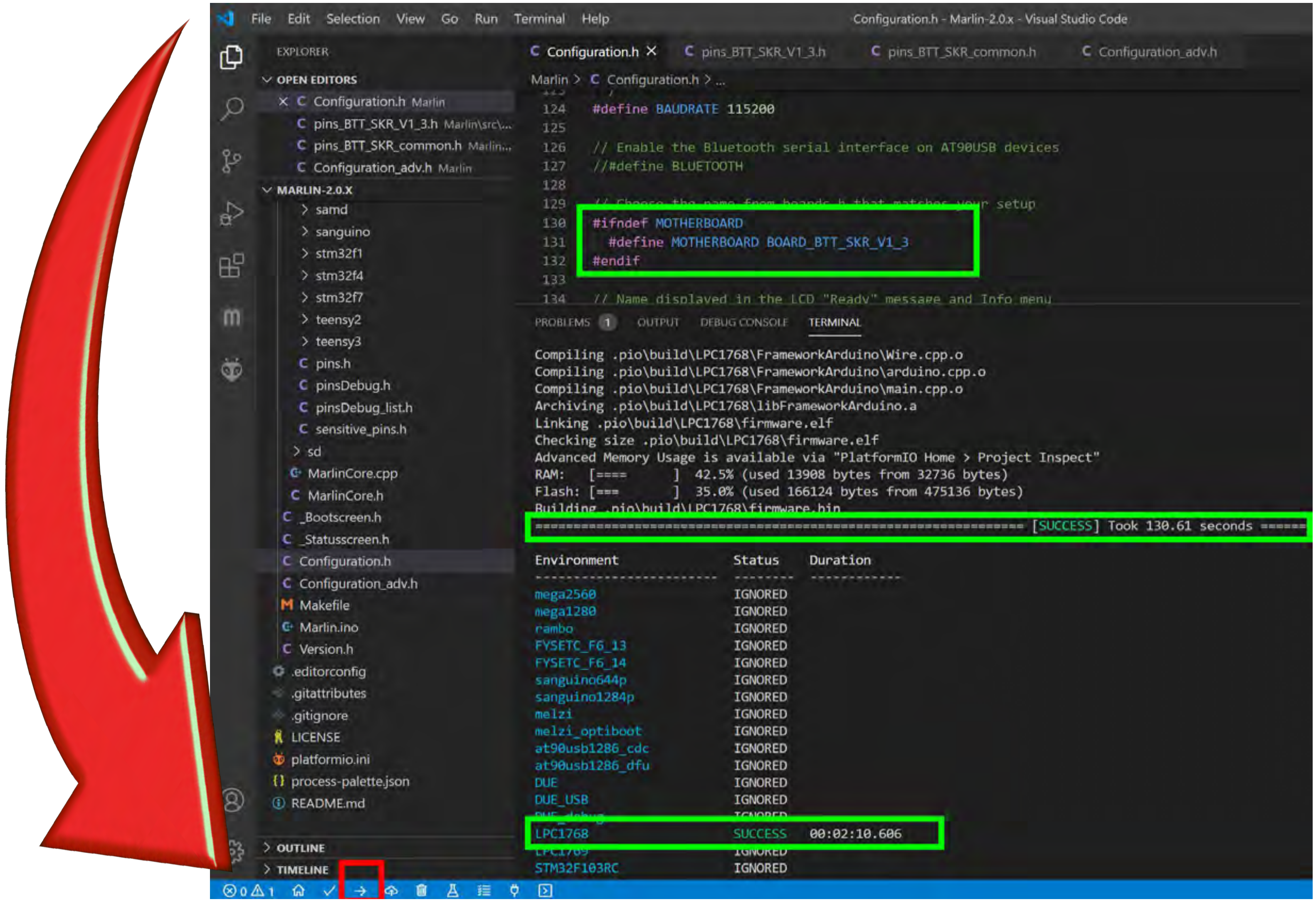
- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



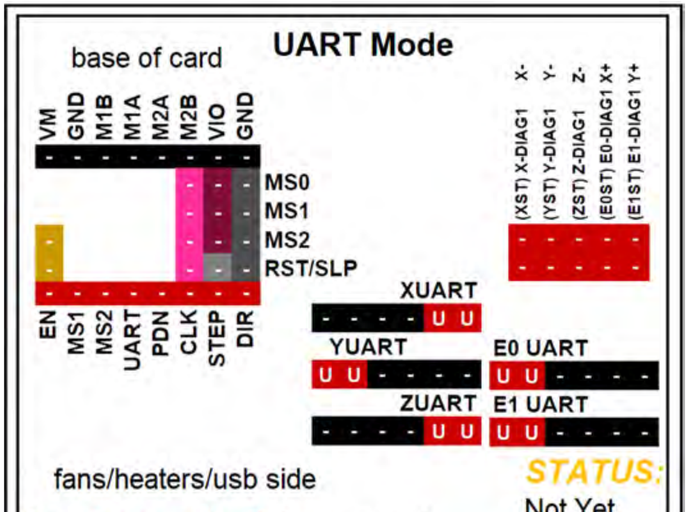
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.



See Appendix D for legend

FYSETC TMC2208 V1.2

UART Mode

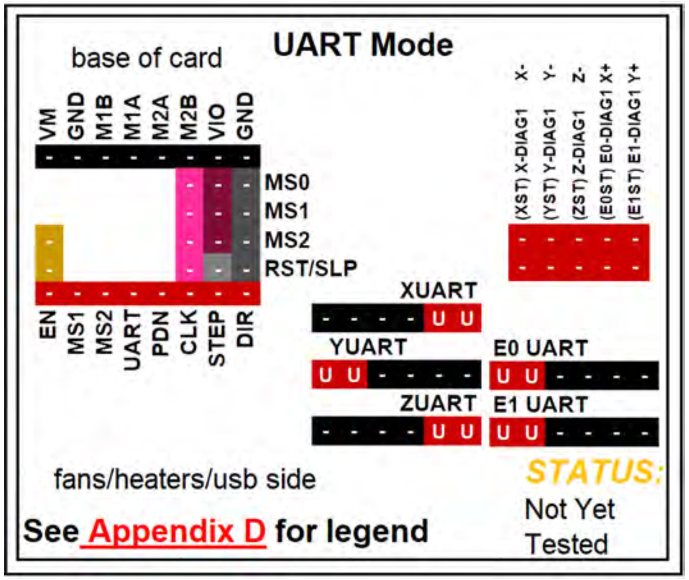
Note: You can use 50% to 90% of the calculated I_{RMS} (I_{MAX}/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

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

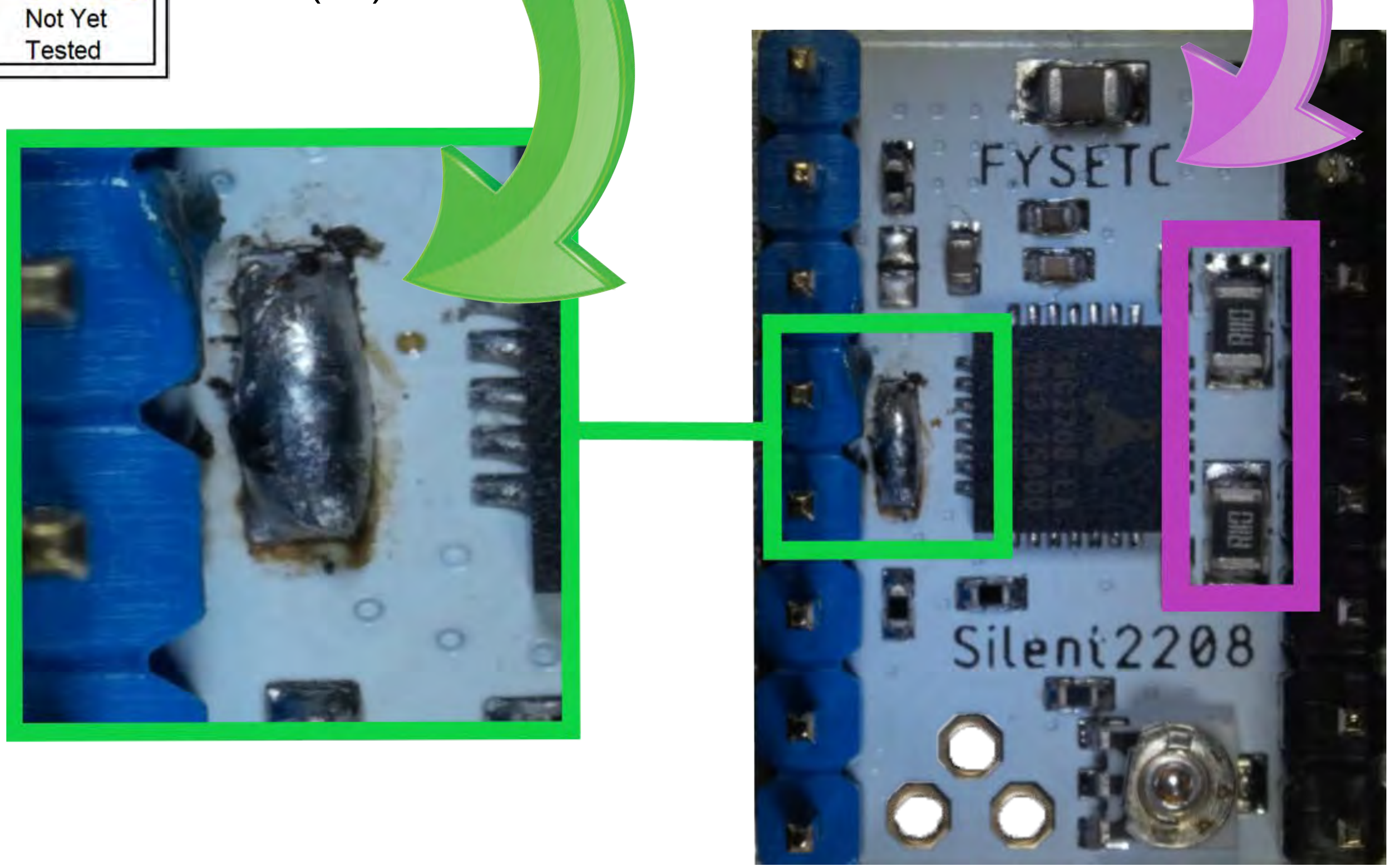
FYSETC TMC2208 V1.2

UART Mode



Important: To ensure that the FYSETC TMC2208 V1.0 or V1.2 is in UART Mode, check to see if all the adjacent pads are soldered together on the bottom of the driver board, as shown in the **GREEN** box. The **PURPLE** box shows the location of the current sense resistors (R_S).

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



FYSETC TMC2208 V1.2 in UART Mode

UART Mode

FYSETC TMC2208 V1.2

UART Mode

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_CURRENT".

R_s = R050 is 0.05 Ohms

R_s = R062 is 0.062 Ohms

R_s = R068 is 0.068 Ohms

R_s = R075 is 0.075 Ohms

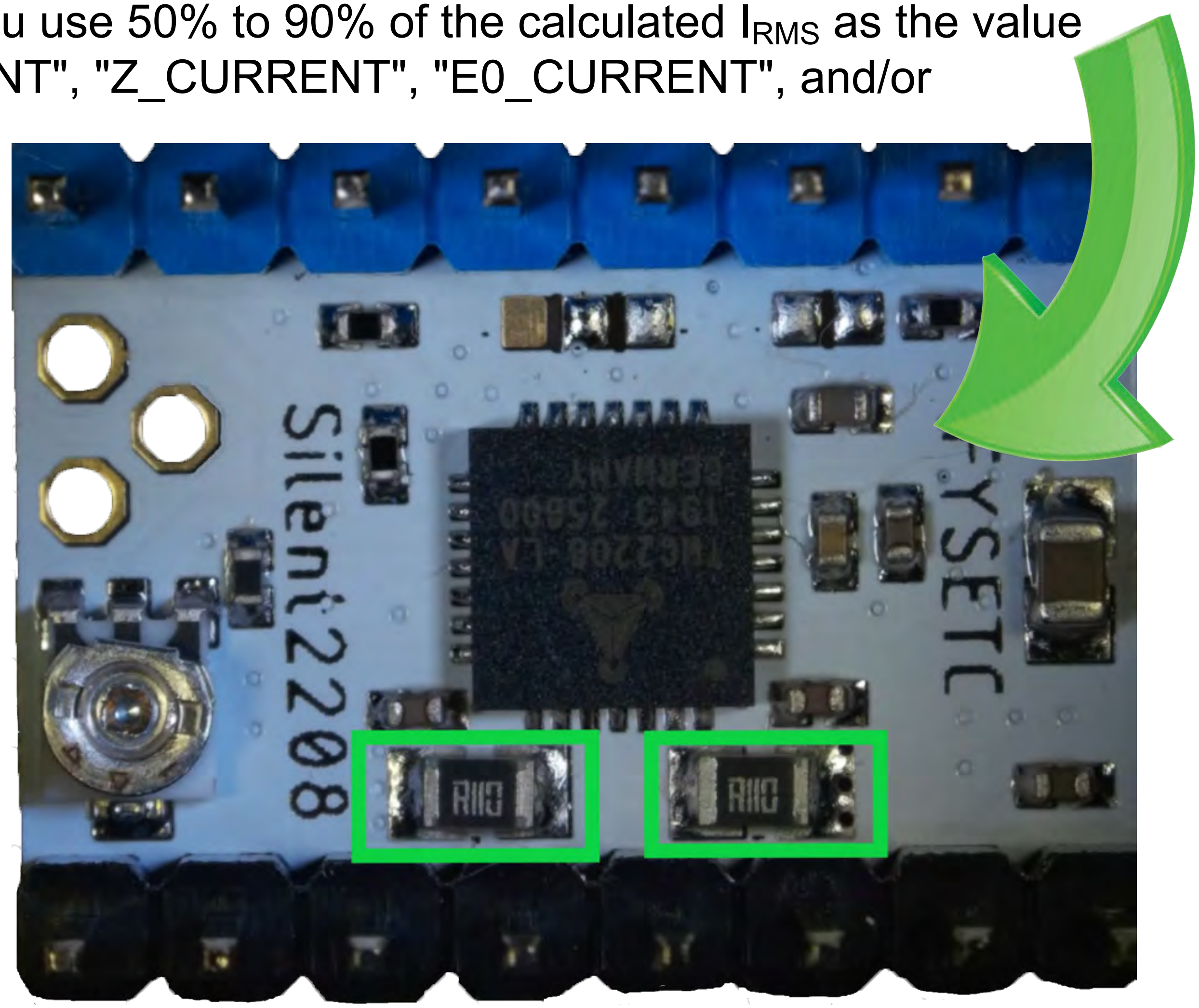
R_s = R100 is 0.1 Ohms

R_s = R110 is 0.11 Ohms

R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

R_s = R220 is 0.22 Ohms



UART Mode

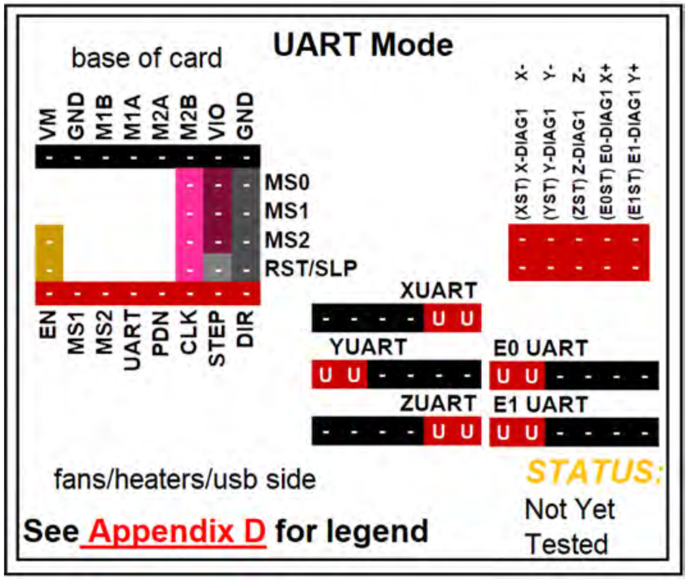
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to ensure proper operation for UART mode.

Note: Set the "U" Jumper(s) for UART MODE!

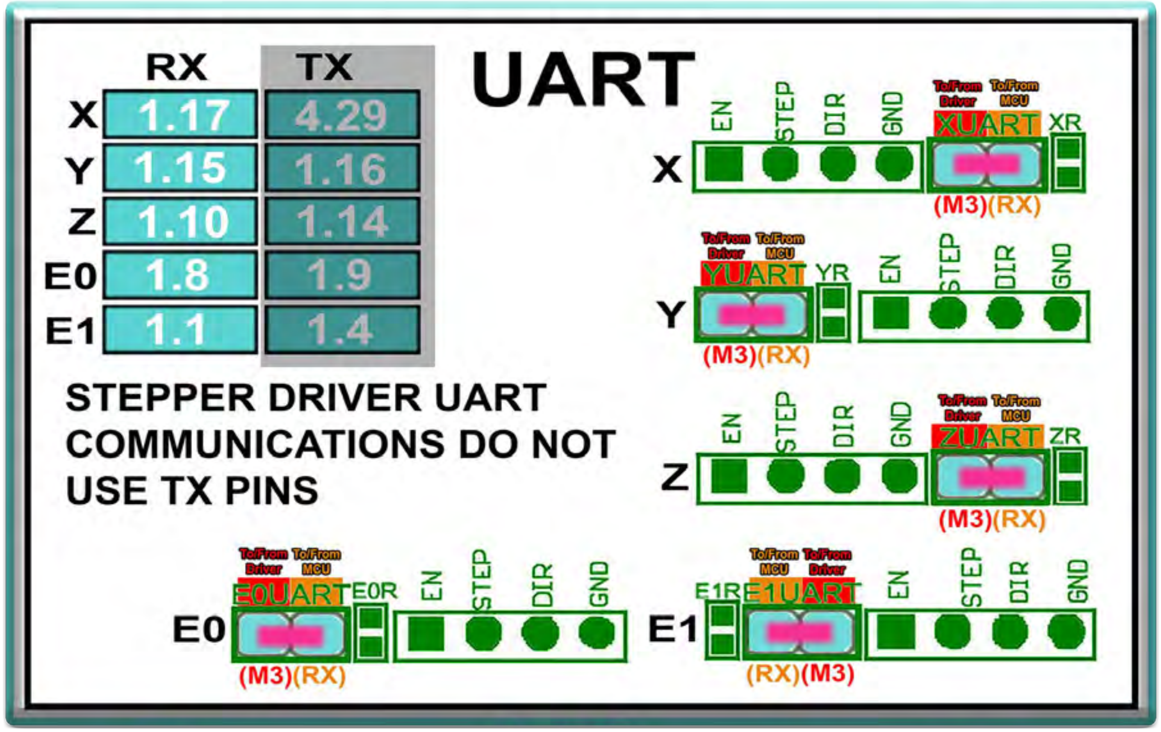
XUART: - - - - U U
YUART: U U - - - -
ZUART: - - - - U U
E0 UART: U U - - - -
E1 UART: U U - - - -

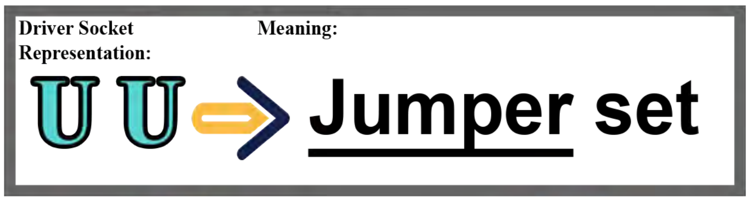
FYSETC TMC2208 V1.2

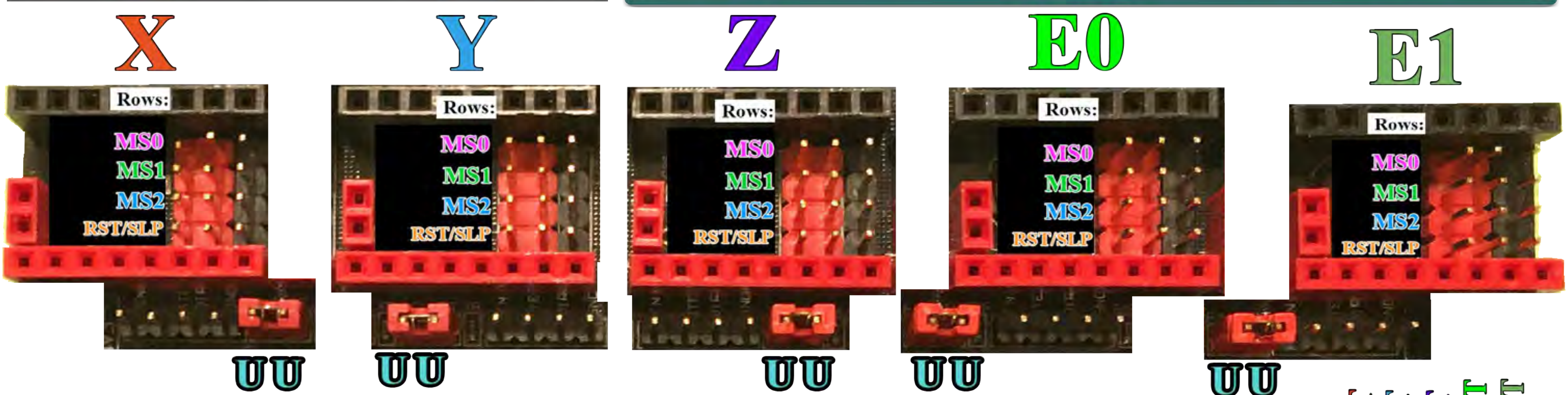
UART Mode



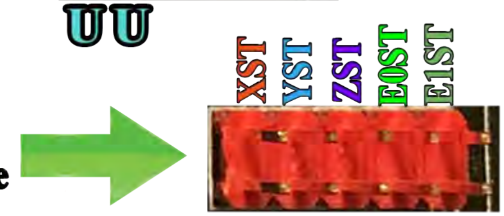
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode





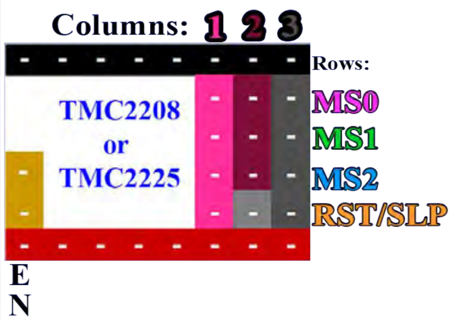


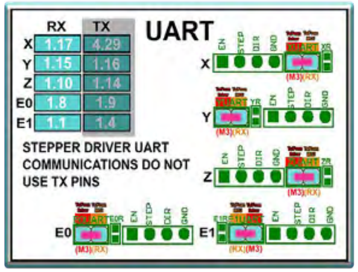
REMOVE 'ST' JUMPER(s) to disable Sensor-less Homing – for drivers without stallGaurd™ feature



SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode

Driver Socket Representation

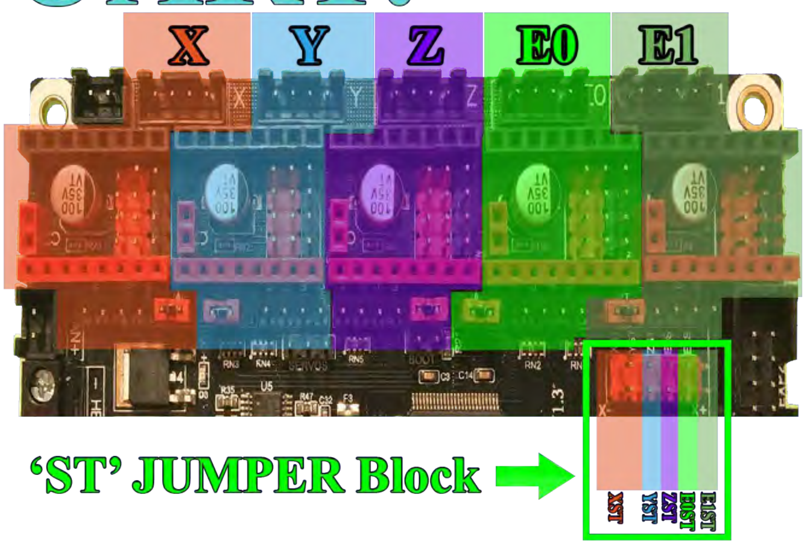




NOTE: SPI PINS must not be set if UART mode is ensbled.

REMOVE 'ST' JUMPER to disable Sensor-less Homing

UART:



'ST' JUMPER Block

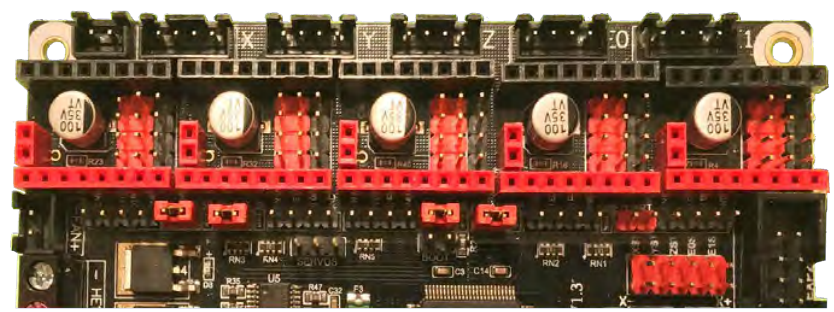
UART Capable Drivers **without** stallGaurd™ feature: TMC2208, and TMC2225

Driver Socket Representation:	Meaning:
U U ⇒	**Jumper Set**

Driver Socket Representation:	Meaning:
U U ⇒	set Jumper so that both U PINS are covered by the Jumper

Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS REMOVED. All other PINS need to be empty for UART mode to work properly.

Here is an example of only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS removed for ALL Axes:



FYSETC TMC2208 V1.2

UART Mode

Information on Sensor-less Homing

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

NOTE: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to ensure proper operation for UART mode.

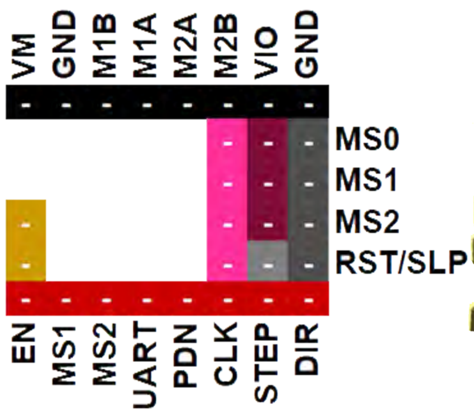
UART Mode

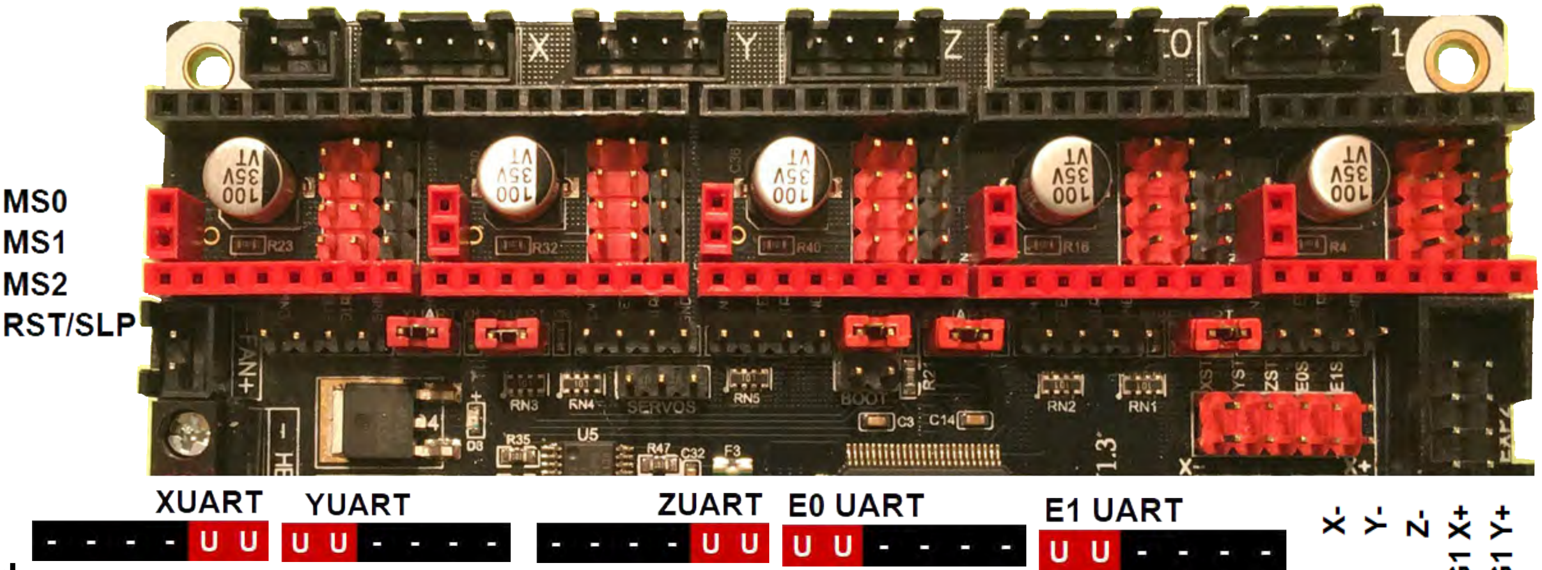
FYSETC TMC2208 V1.2

UART Mode

Note: Set the "U" Jumper(s) for UART MODE!

UART





See Appendix D for legend

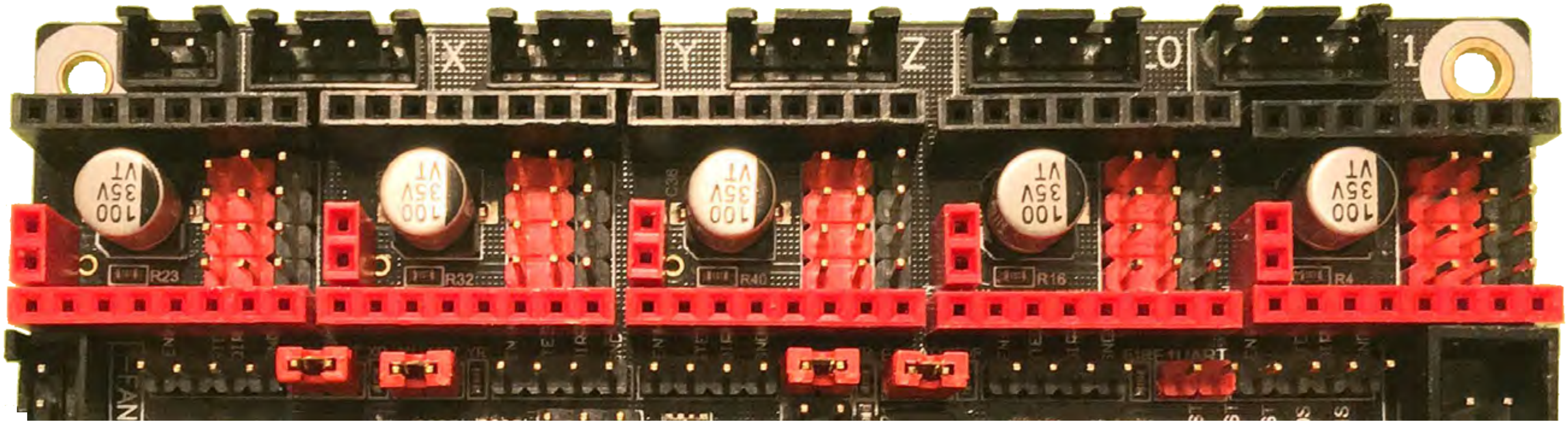
Note: **TMC2208 __does not have__ sensor-less homing capability.**

FYSETC TMC2208 V1.2

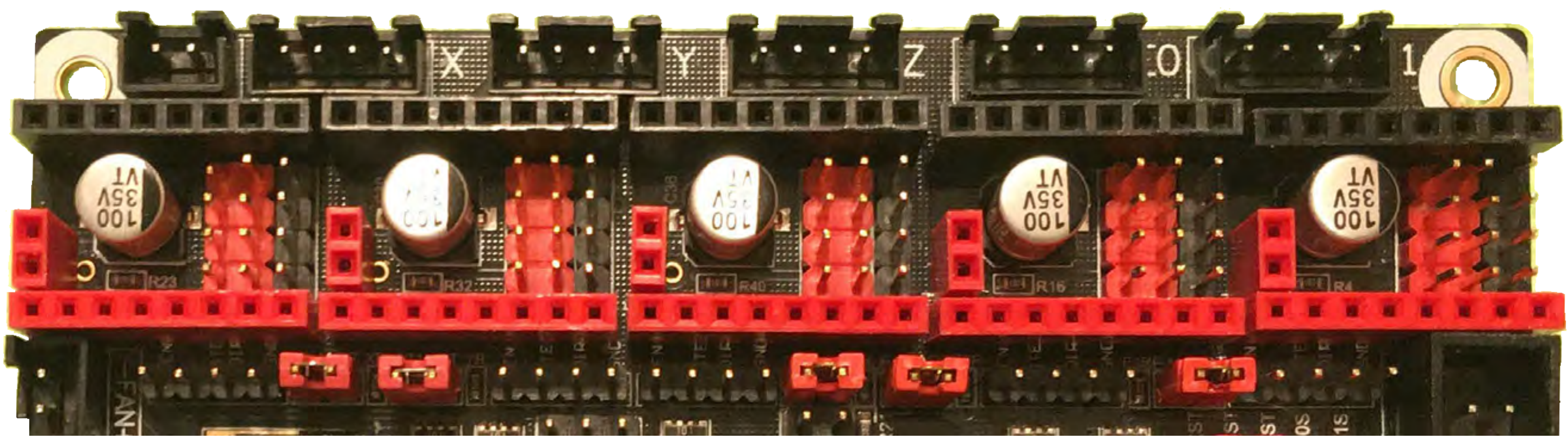
UART Mode

Examples of Different UART Configurations

X, Y, Z, and E0 axes configured for UART mode.



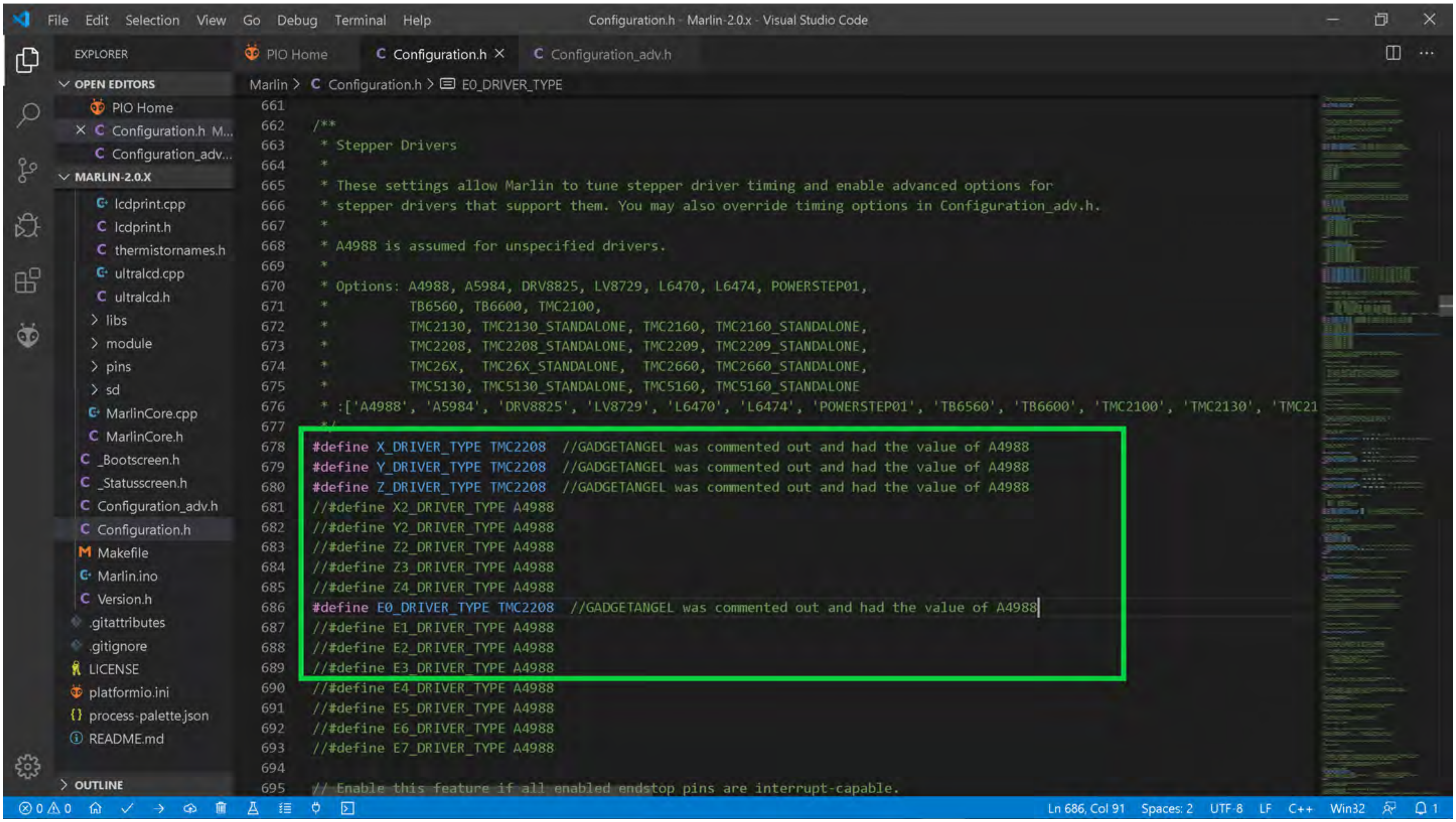
X, Y, Z, E0 and E1 axes configured for UART mode.



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

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

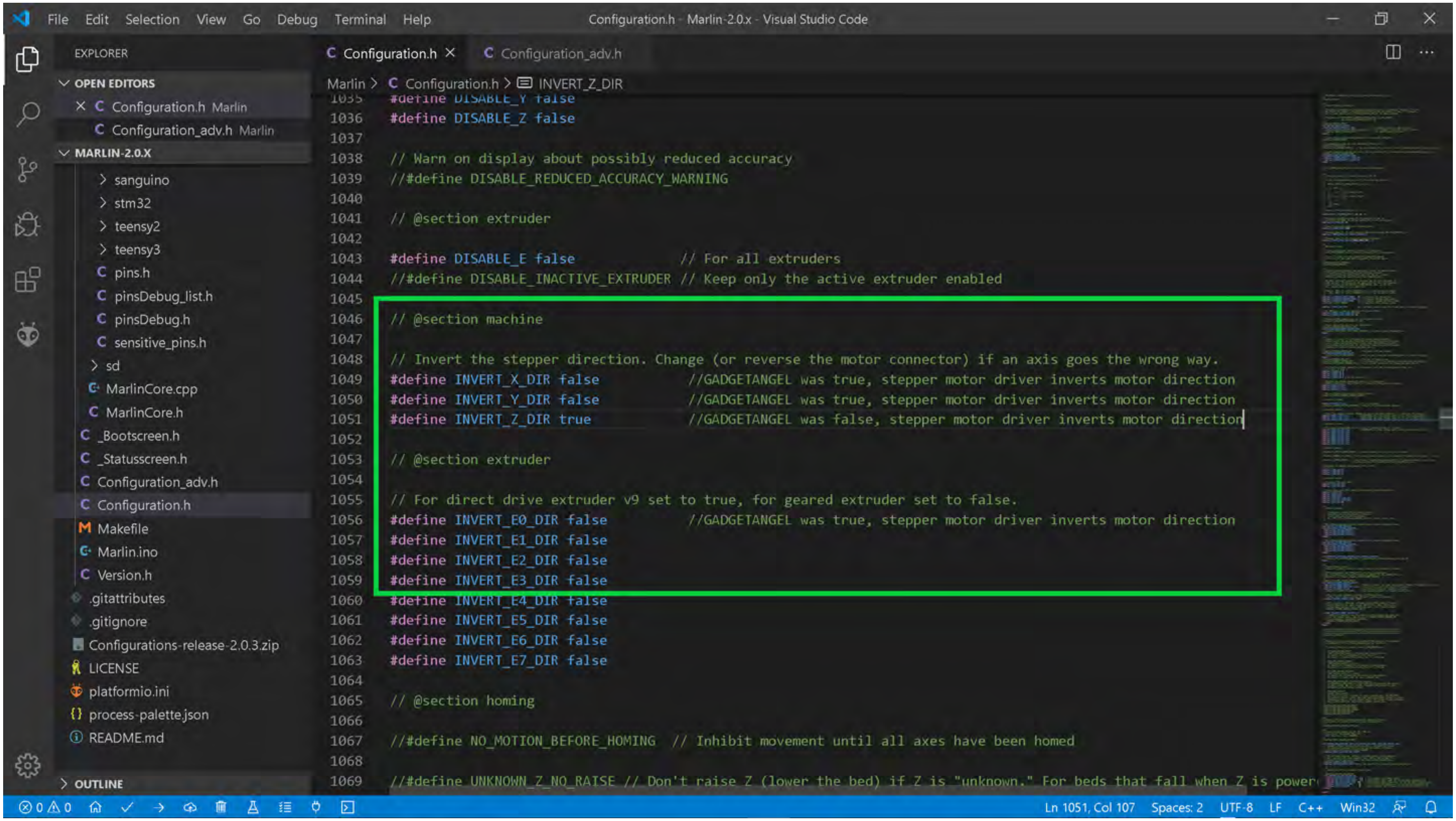
- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "//" appear at the beginning of a line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



- Go to the next page.

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

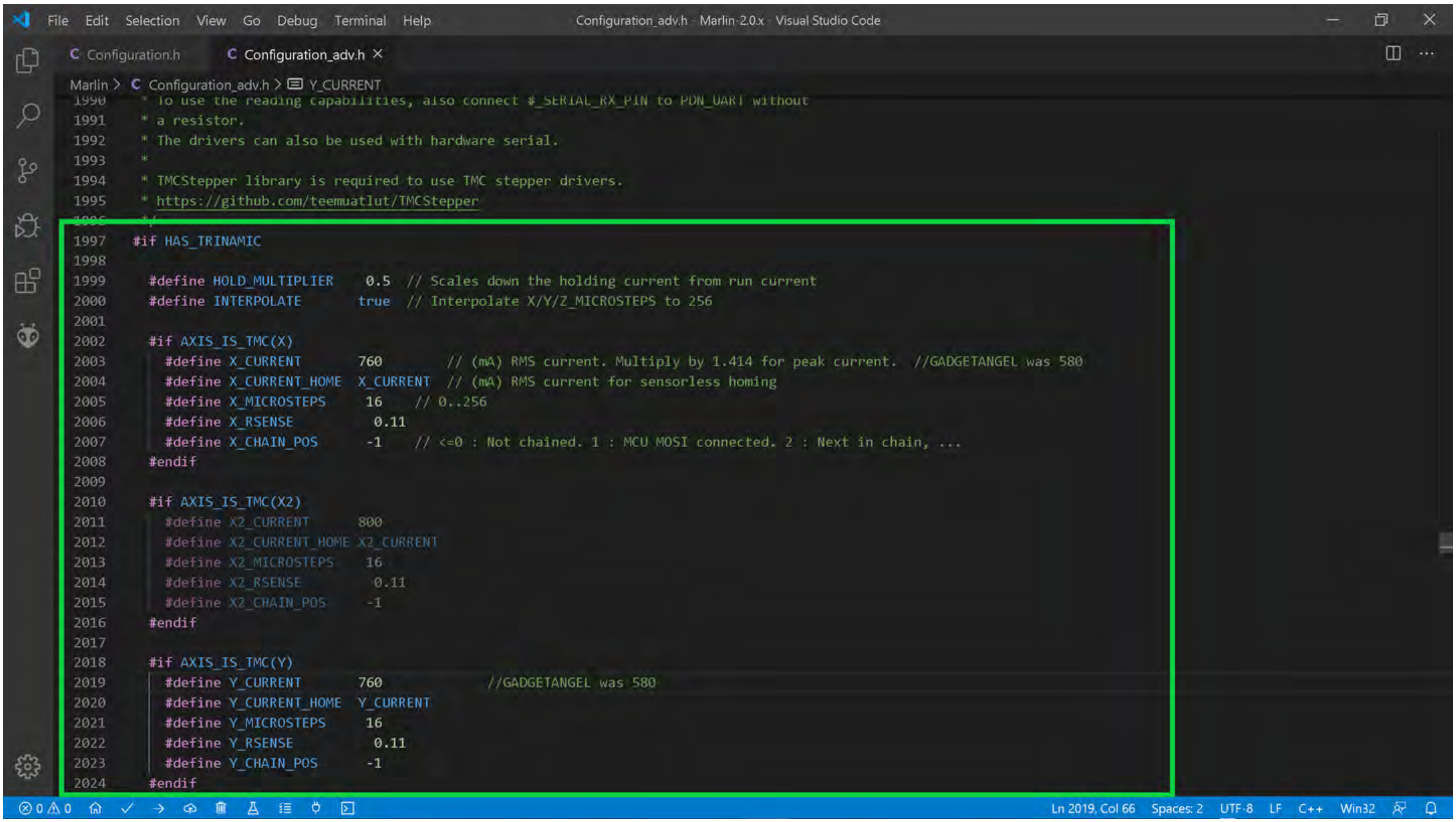
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208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



- Go to the next page.

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

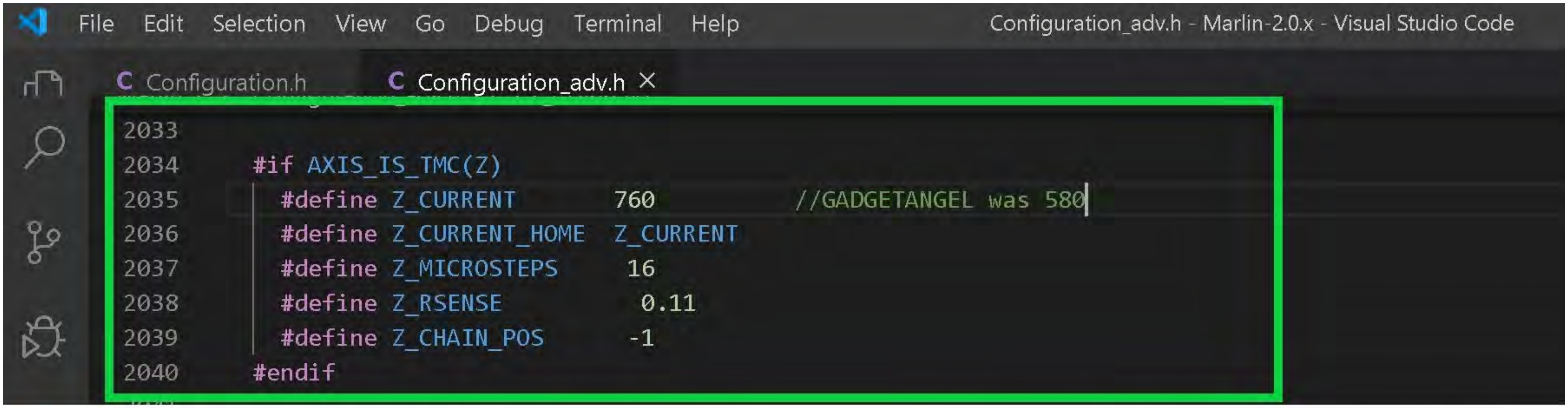
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2208 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you do not want to use V_{ref} as the value for "X_CURRENT" and/or "Y_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT" and/or "Y_CURRENT".

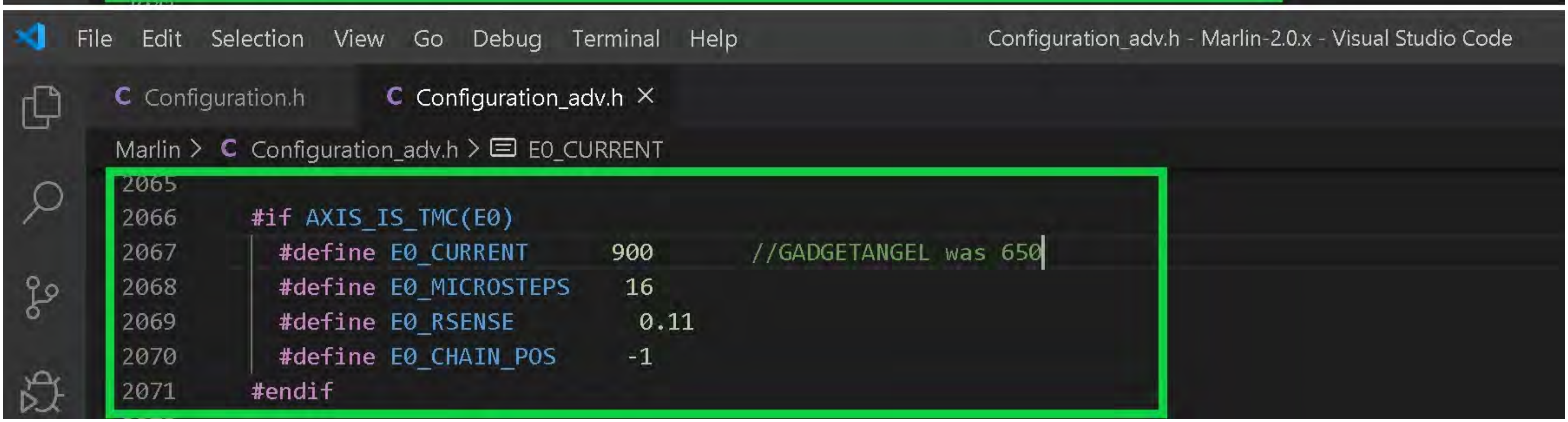


- Go to the next page.

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

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

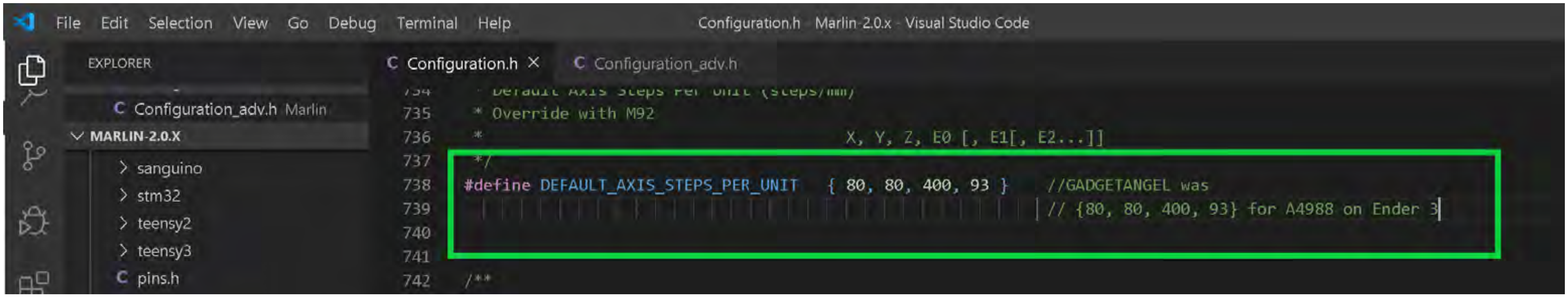




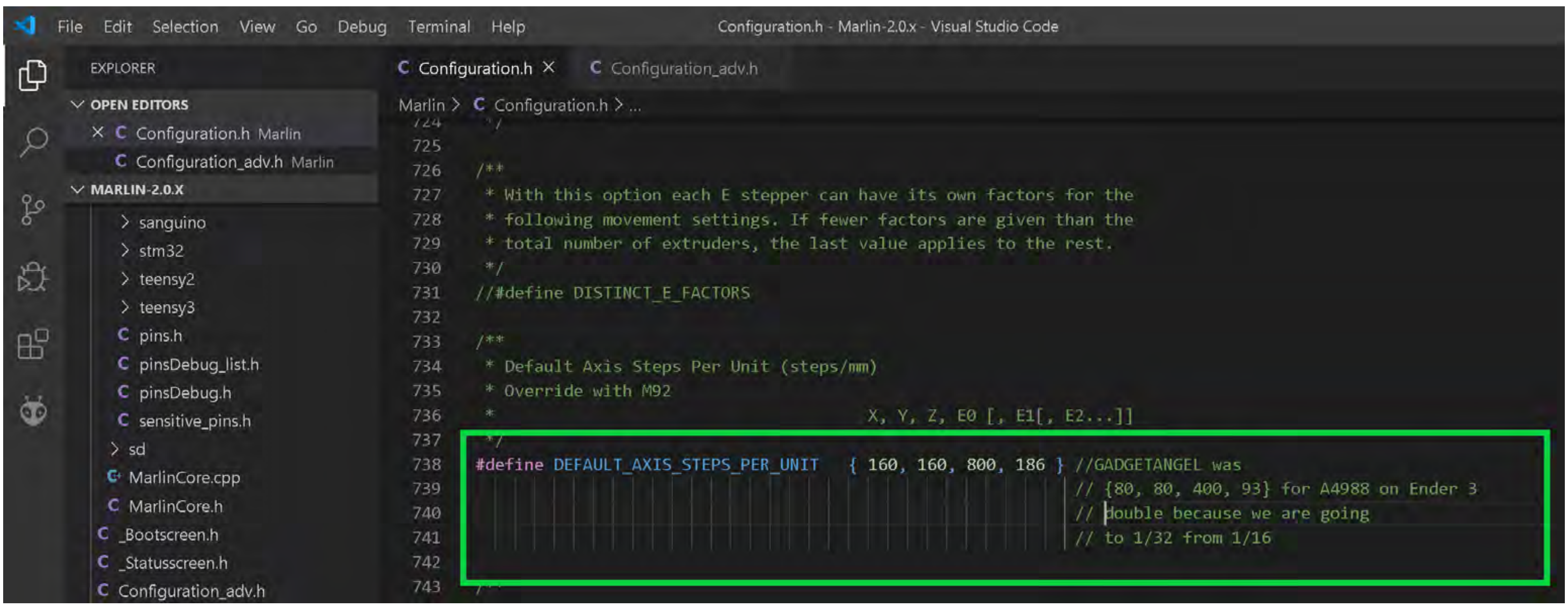
- Go to the next page.

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

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



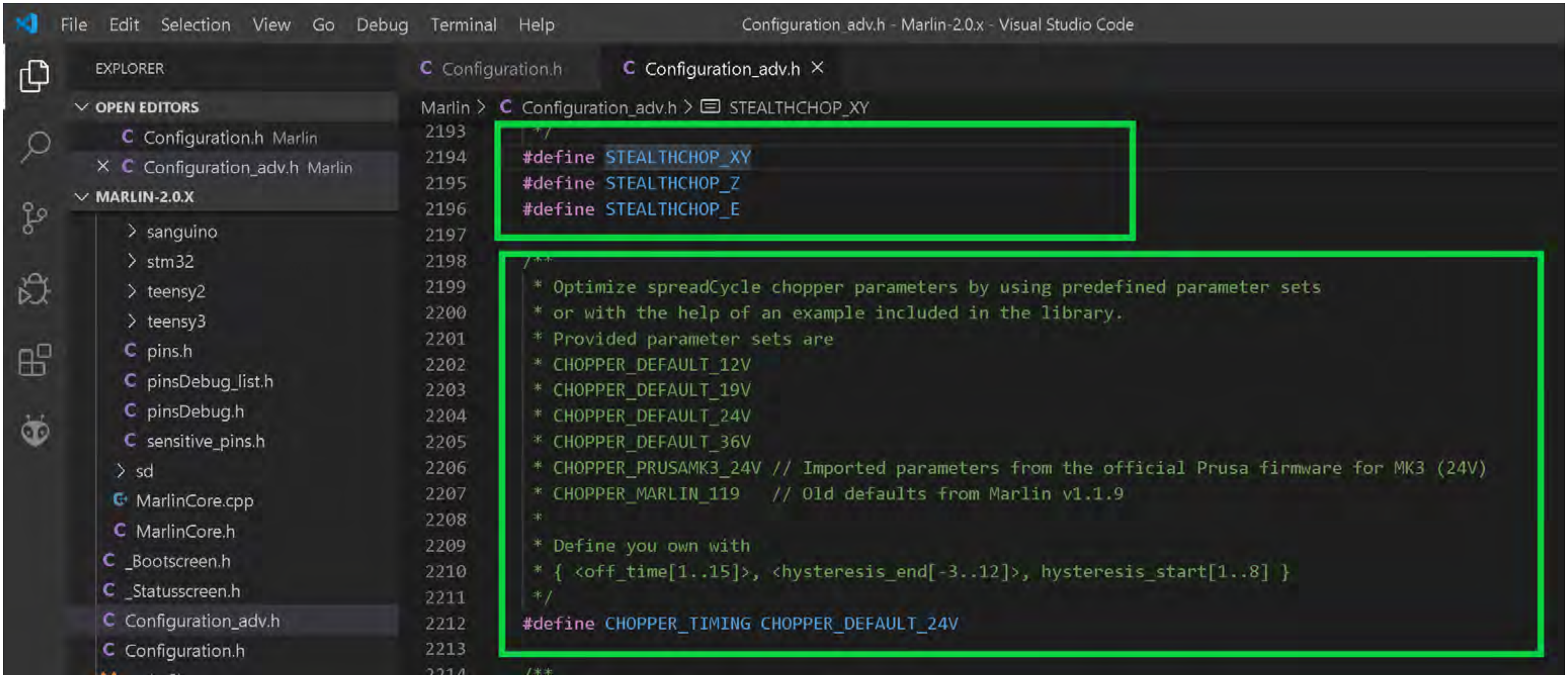
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



- Go to the next page.

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

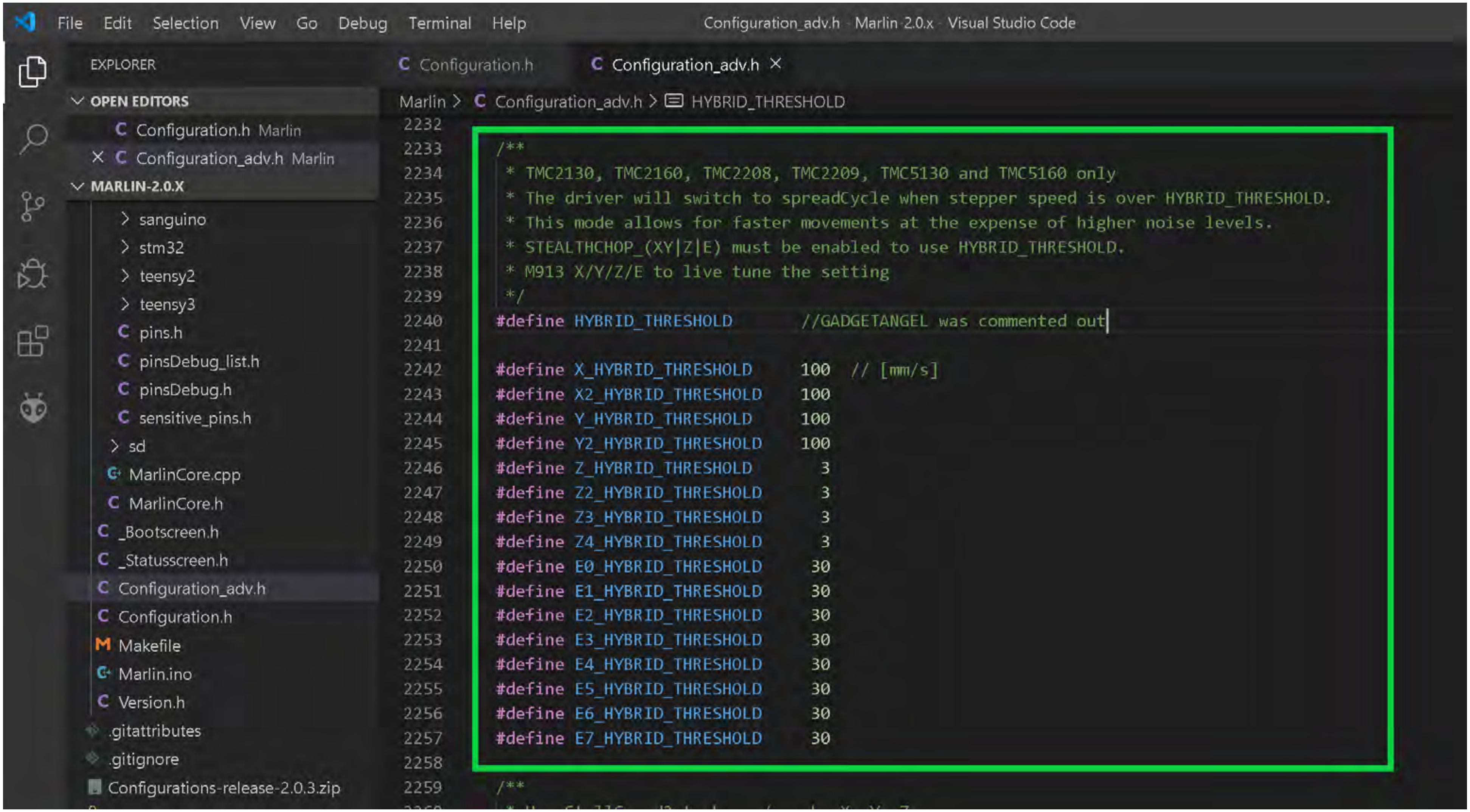
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

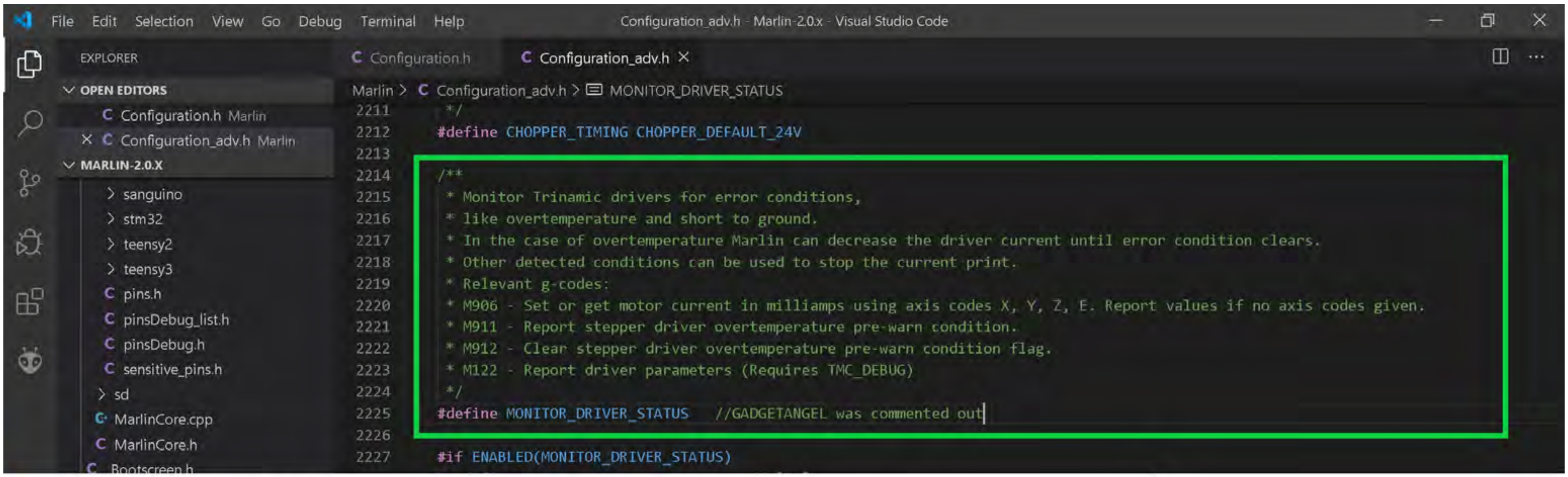
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

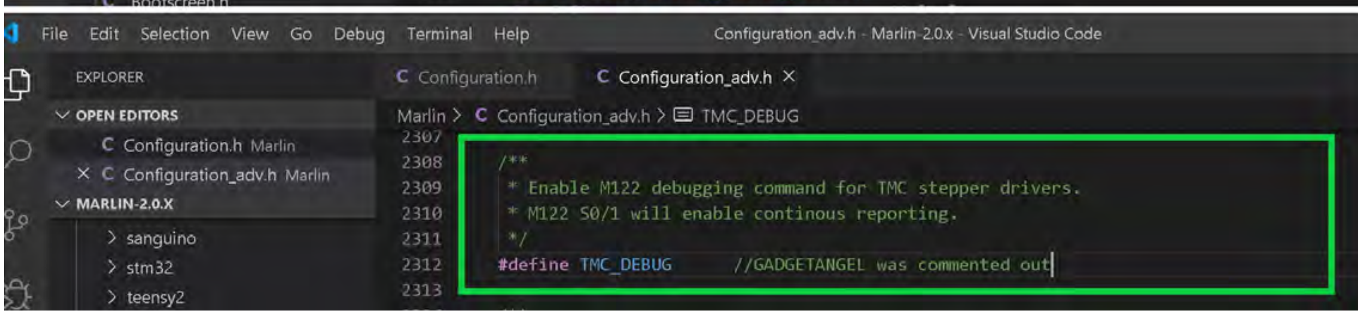


- Go to the next page.

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

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

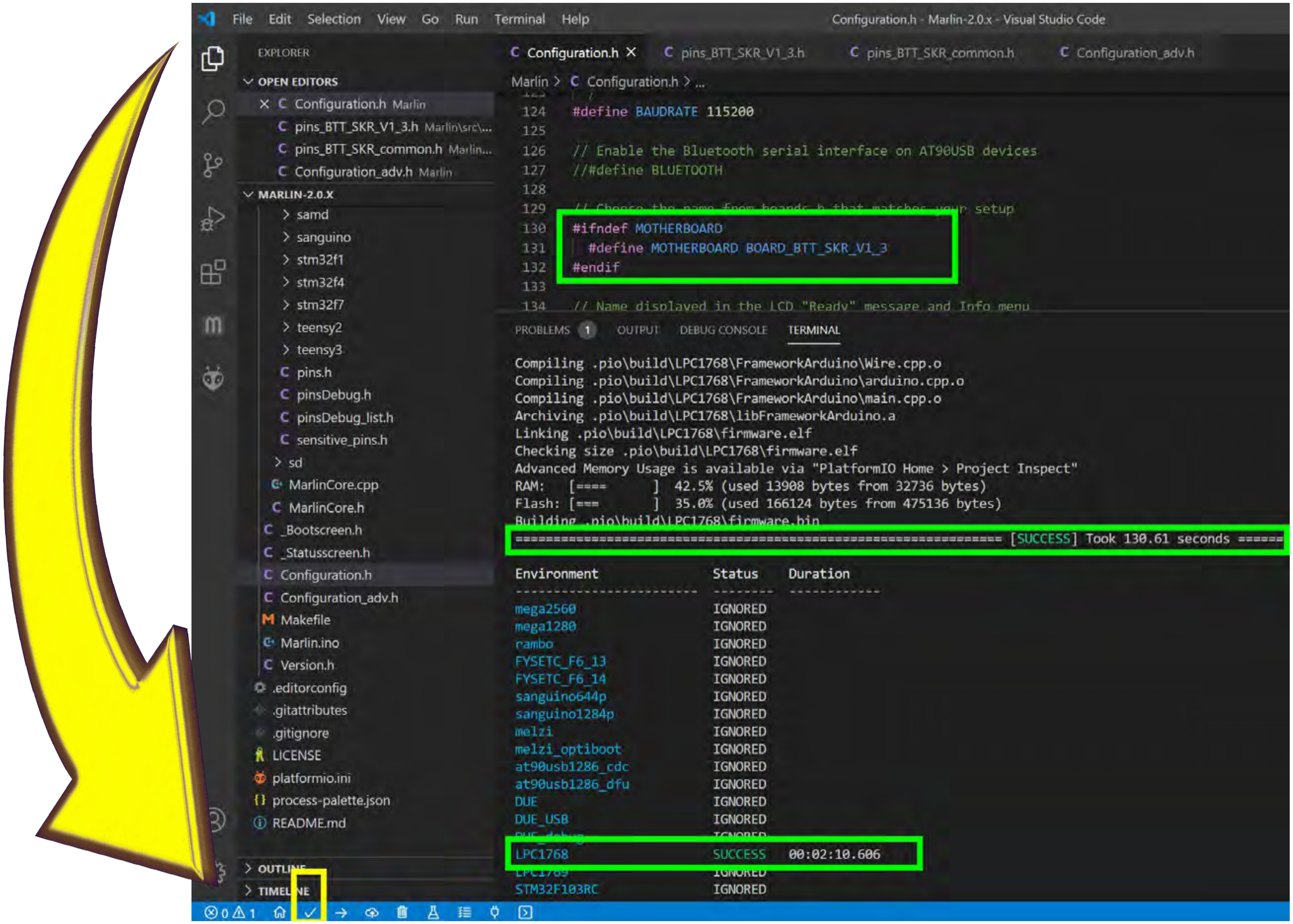




- Go to the next page.

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

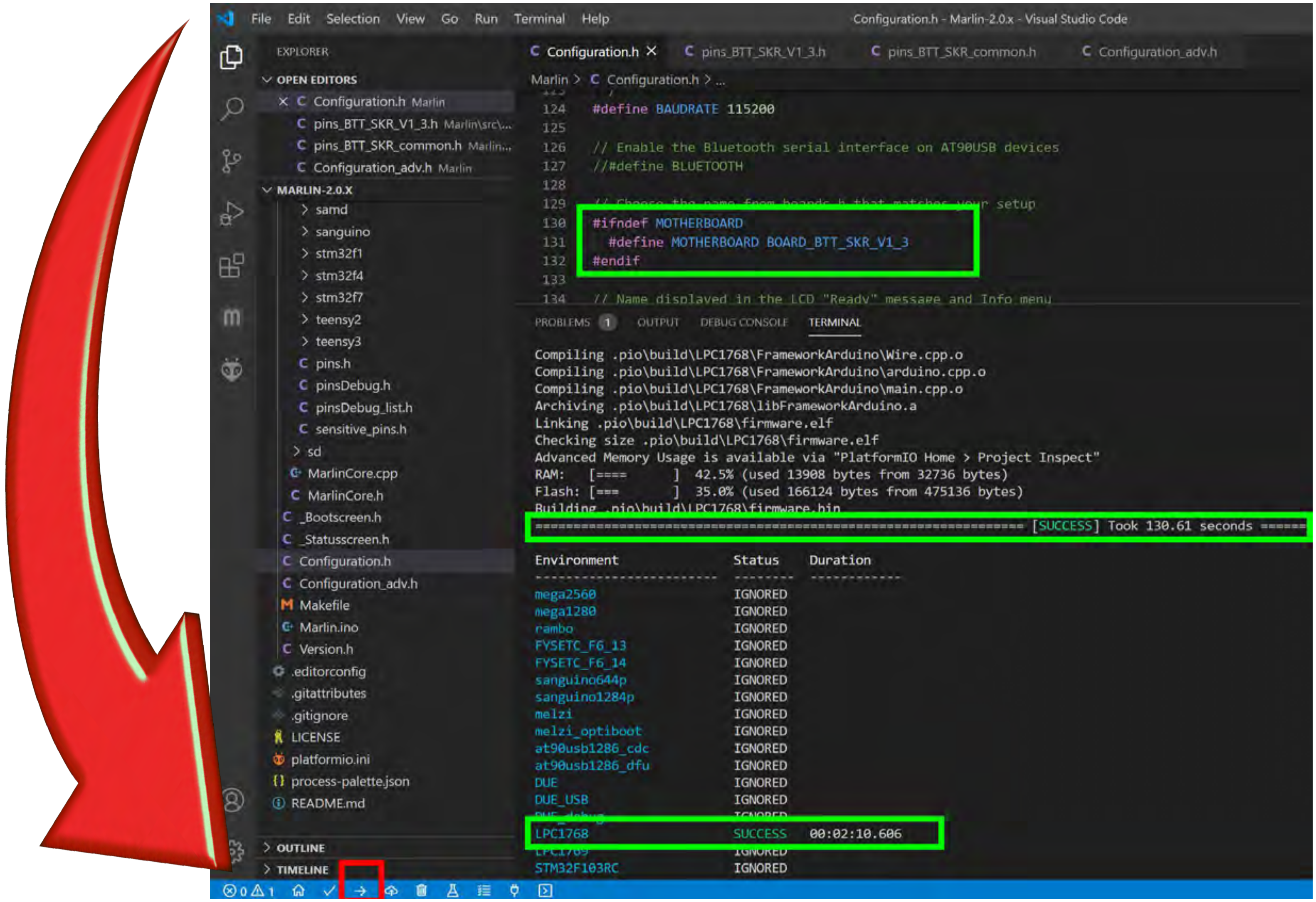
- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



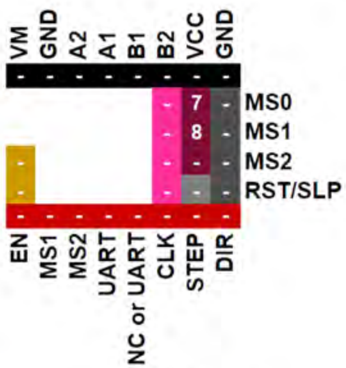
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2225 V1.0

Stand-alone Mode

base of card

Stand-alone Mode



	0	1
4	0	0
8	1	0
16	0	1
32	1	1

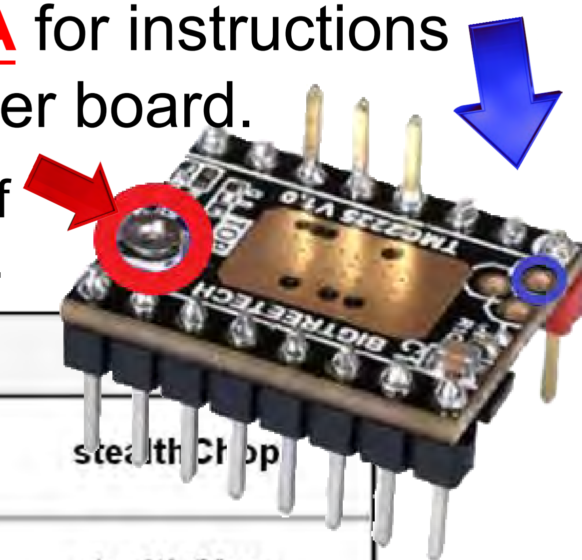
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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	MS0	MS1	Steps	Interpolation	
BIQU® TMC2225 Stand Alone Mode Maximum 32 Subdivision 35V DC 2A (peak)	Low	Low	1 / 4	1 / 256	stealthChop
	High	Low	1 / 8	1 / 256	stealthChop
	Low	High	1 / 16	1 / 256	stealthChop
	High	High	1 / 32	1 / 256	stealthChop

Driving Current Calculation Formula R_S(Typical Sense Resistor)=0.15Ω	$I_{MAX}=V_{ref}*0.7222$ See Appendix B#10.Use 50% to 90% as shown below: $I_{MAX}=(V_{ref}*0.7222)*0.90$	$V_{ref}=I_{MAX}*1.3846$ See Appendix B#10.Use 50% to 90% as shown below: $V_{ref}=(I_{MAX}*1.3846)*0.90$

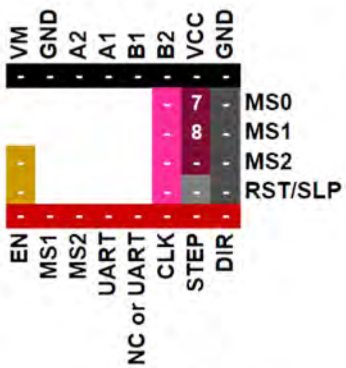
- **See next page for the legend that belongs to the above chart.**

BIQU TMC2225 V1.0

Stand-alone Mode

base of card

Stand-alone Mode



	0	1
4	0	0
8	1	0
16	0	1
32	1	1

fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
X.XA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)=X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇨ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇨ Jumper Set

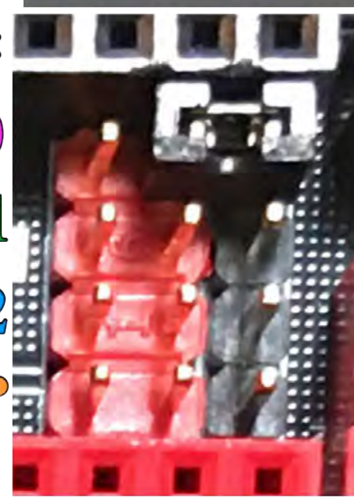
Rows:

MS0

MS1

MS2

RST/SLP



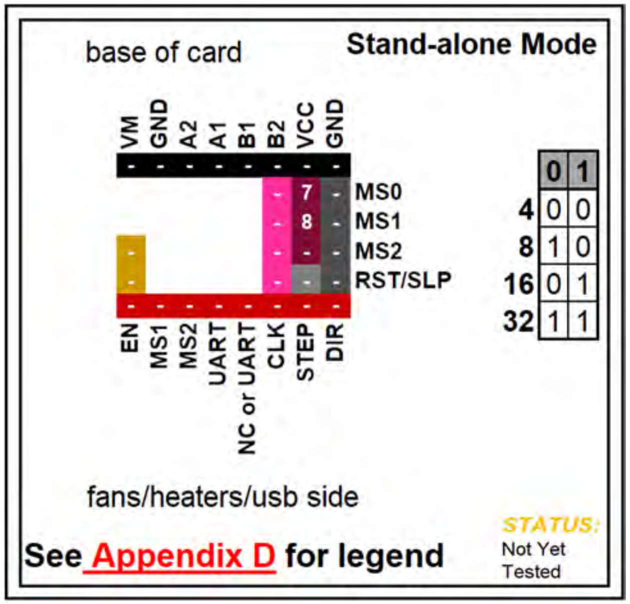
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU TMC2225 V1.0

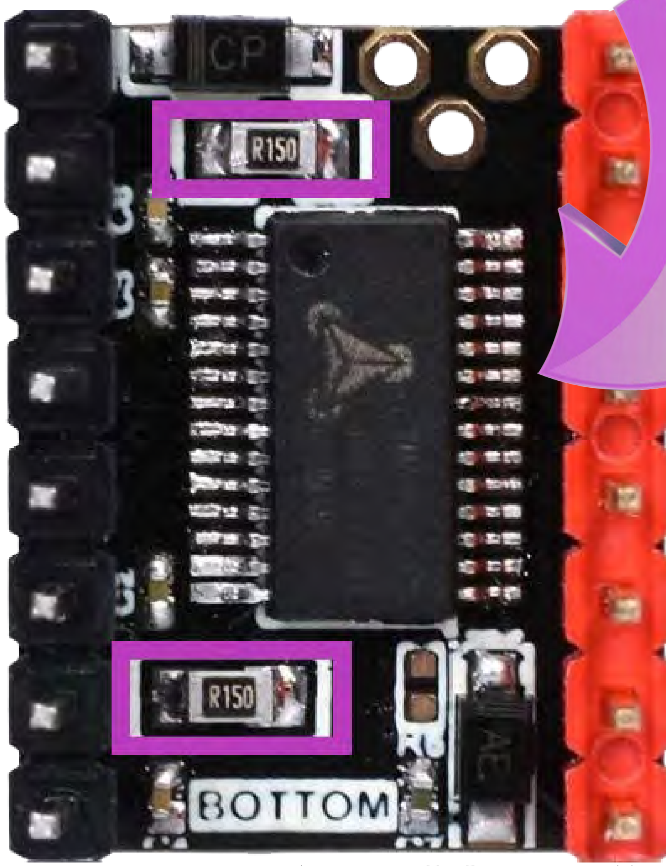
Stand-alone Mode

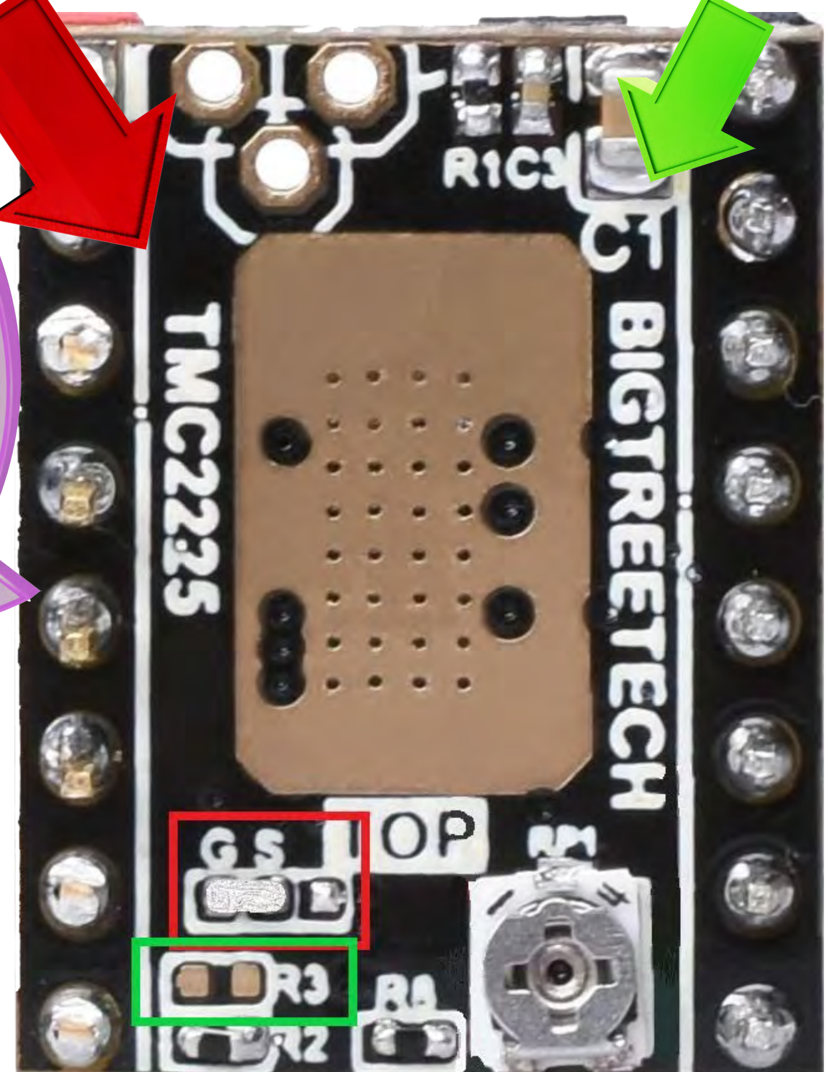


Note: To obtain **stand-alone mode** for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as seen in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for StealthChop as seen in **RED** below.

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

Note: MOST BIQU TMC2225 V1.0 driver boards, when purchased **for UART mode, will have two R3 pads (located on the top of the driver board), which are NOT soldered together.** This indicates the driver board can use the UART pin for the UART single wire interface (if the UART,"U", jumper is in place on the SKR V1.3 board)

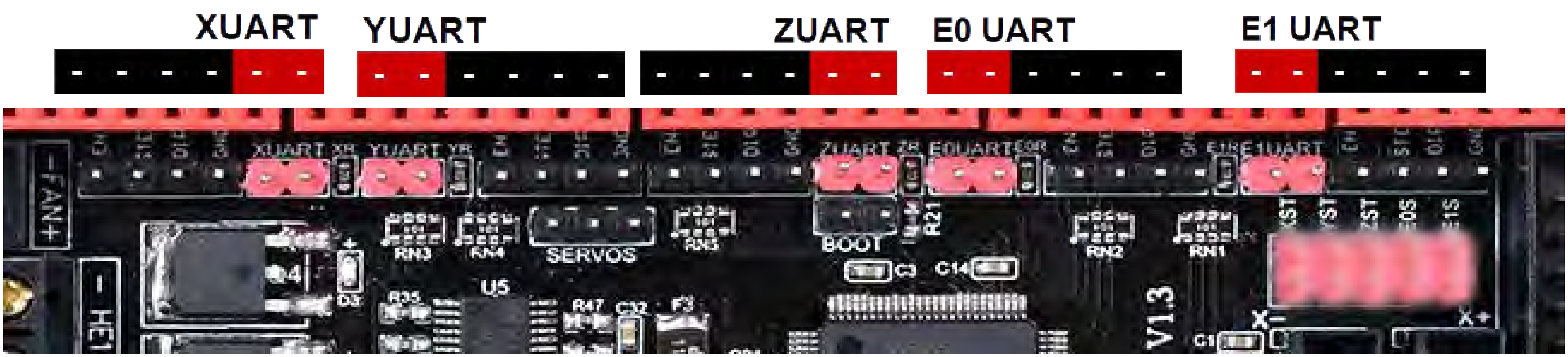




Stand-alone Mode

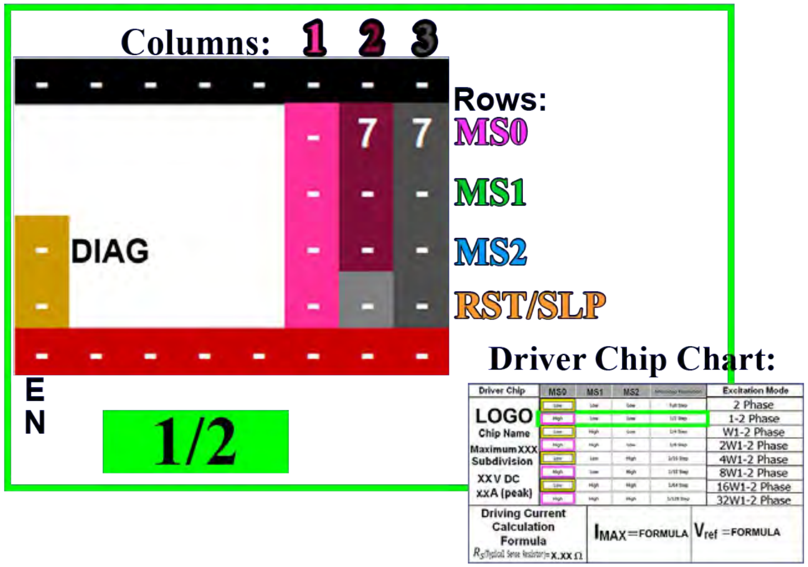
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode.

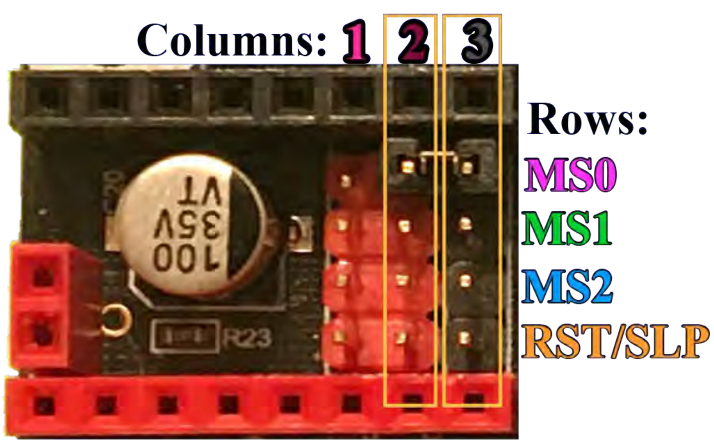


SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Meaning:

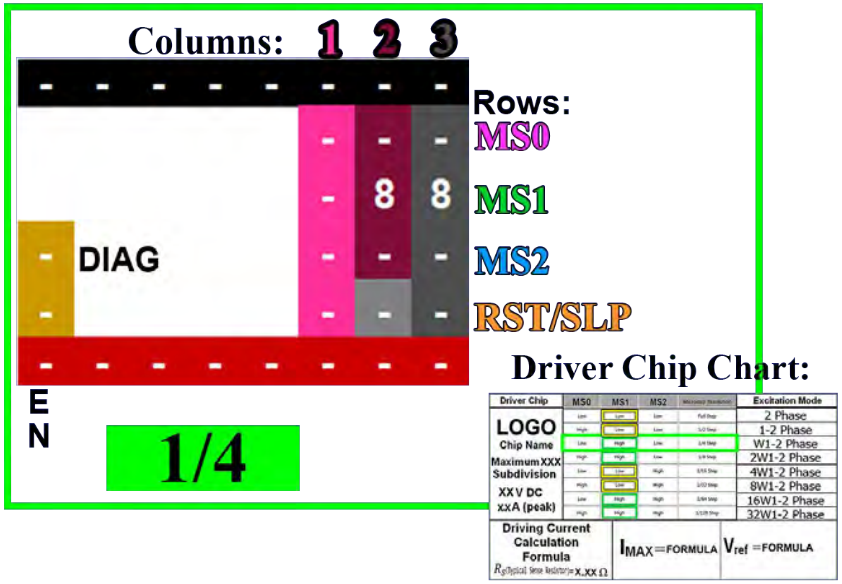
Driver Chip Chart:

Driver Chip Chart	Meaning
High	set Jumper between column 2 and column 3 on the MS0 row
Low	No Jumper Set

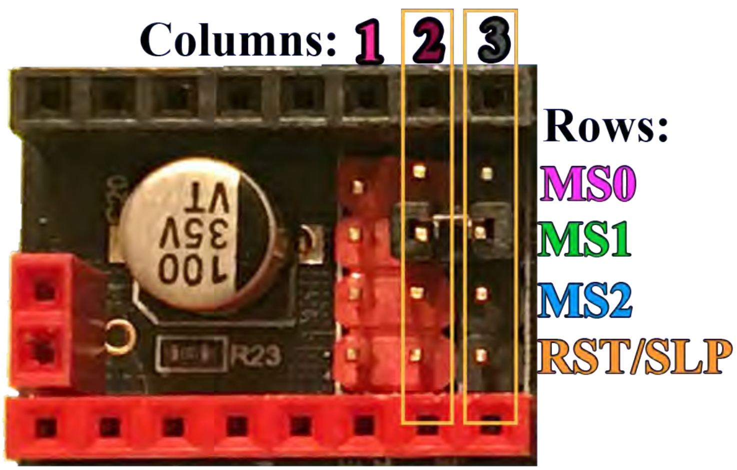
BIQU TMC2225 V1.0

Stand-alone Mode

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	→	Low
8 8	→	High
- -	→	Low

Meaning:

Driver Chip Chart:		Meaning:
High	→	set <u>Jumper</u> between column 2 and column 3 on the MS1 row
Low	→	No <u>Jumper</u> Set

Stand-alone Mode

BIQU TMC2225 V1.0

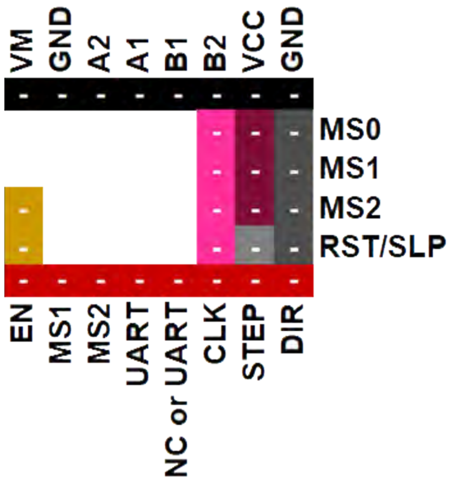
Stand-alone Mode

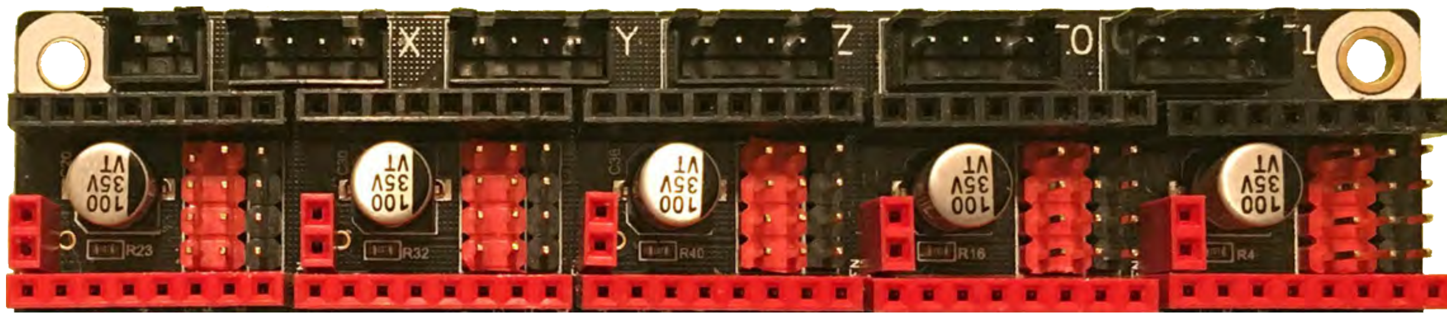
Stand-alone Mode

1 / 4

Interpolation:
1 / 256

StealthChop





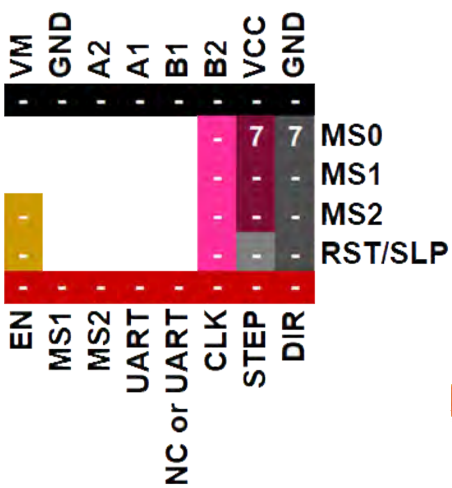
See Appendix D for legend

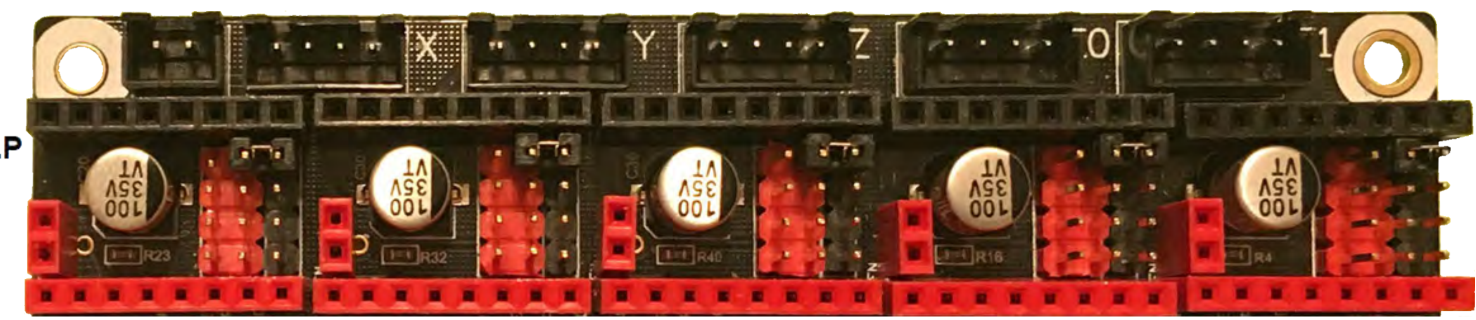
Stand-alone Mode

1 / 8

Interpolation:
1 / 256

StealthChop





See Appendix D for legend

Stand-alone Mode

BIQU TMC2225 V1.0

Stand-alone Mode

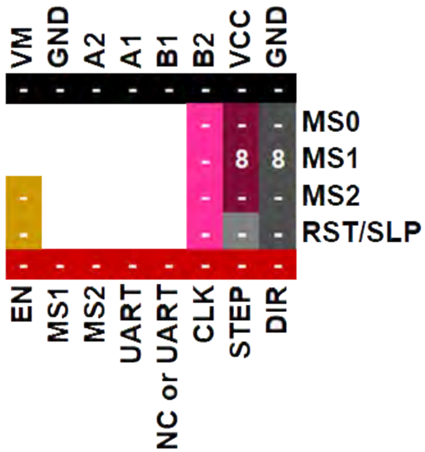
Stand-alone Mode

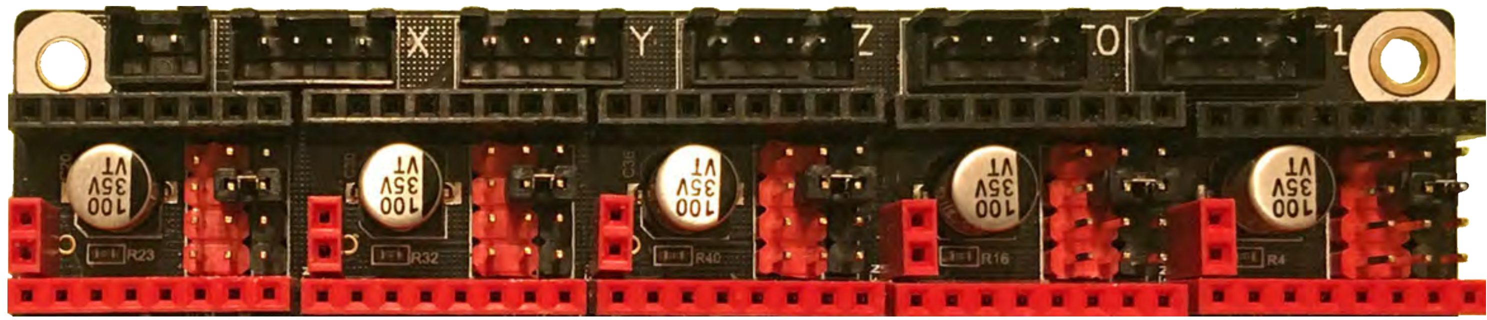
1 / 16

Interpolation:

1 / 256

StealthChop





See Appendix D for legend

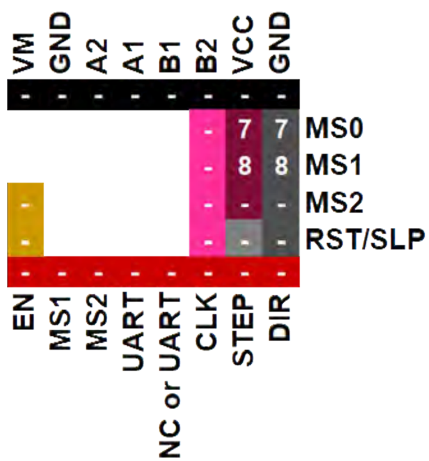
Stand-alone Mode

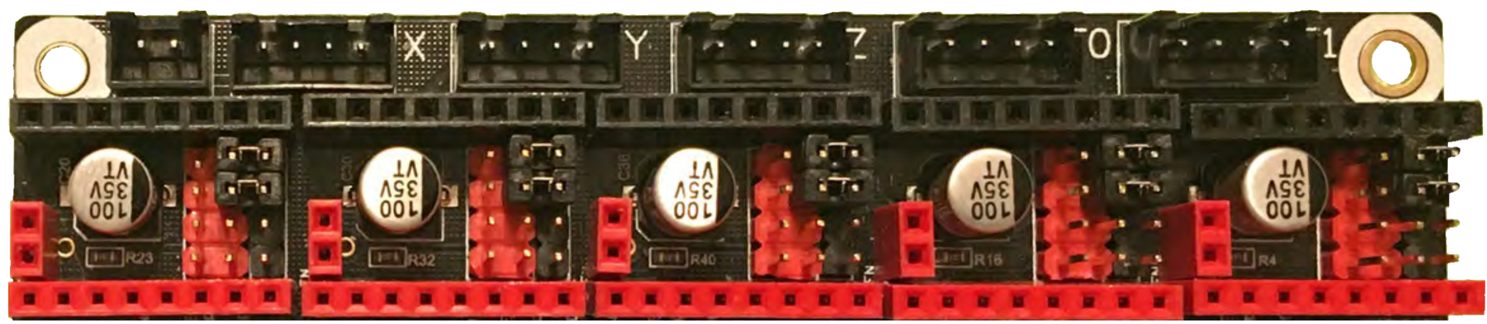
1 / 32

Interpolation:

1 / 256

StealthChop



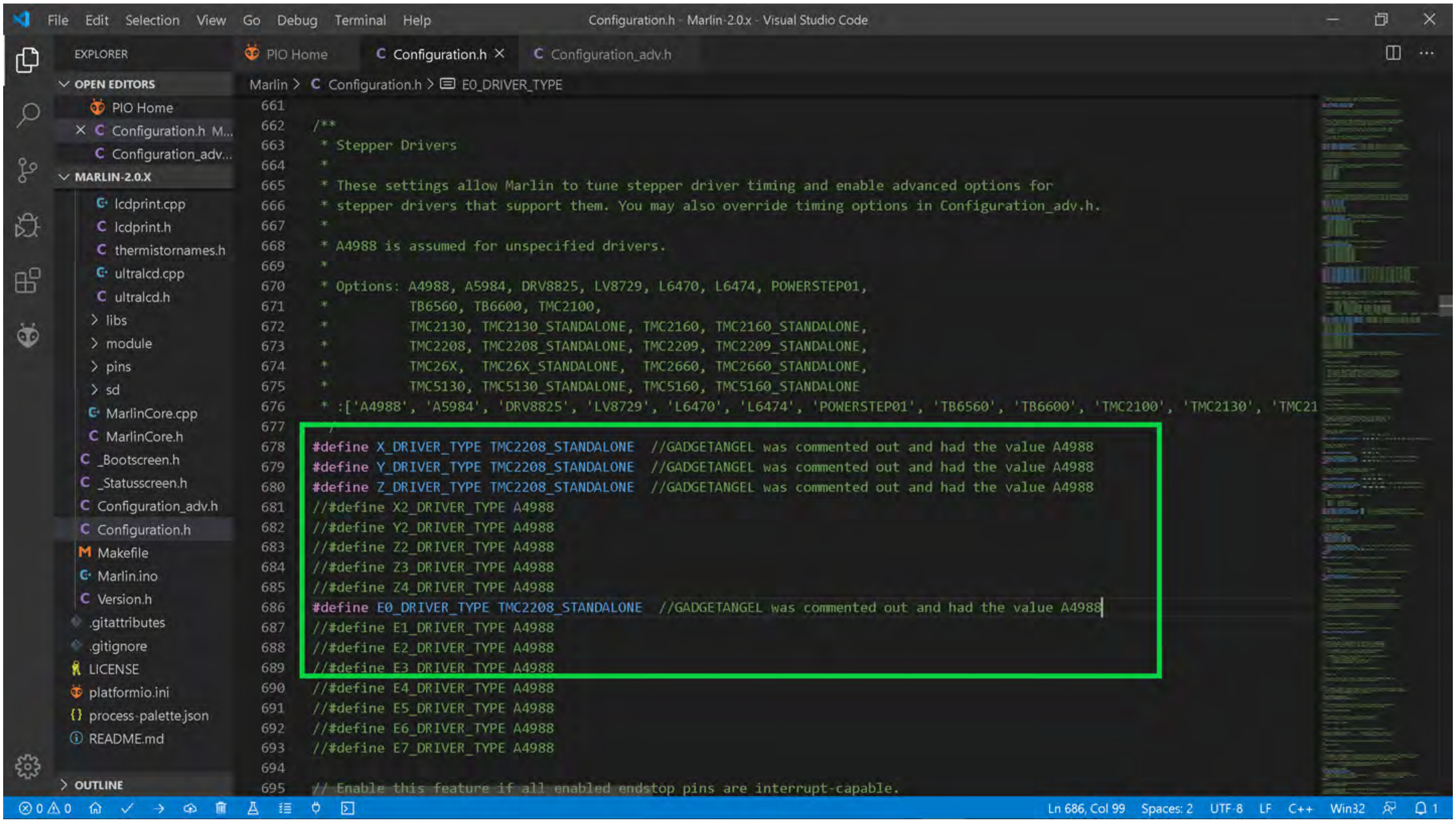


See Appendix D for legend

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

NOTE: **Go to Appendix C,** and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in stand-alone mode.

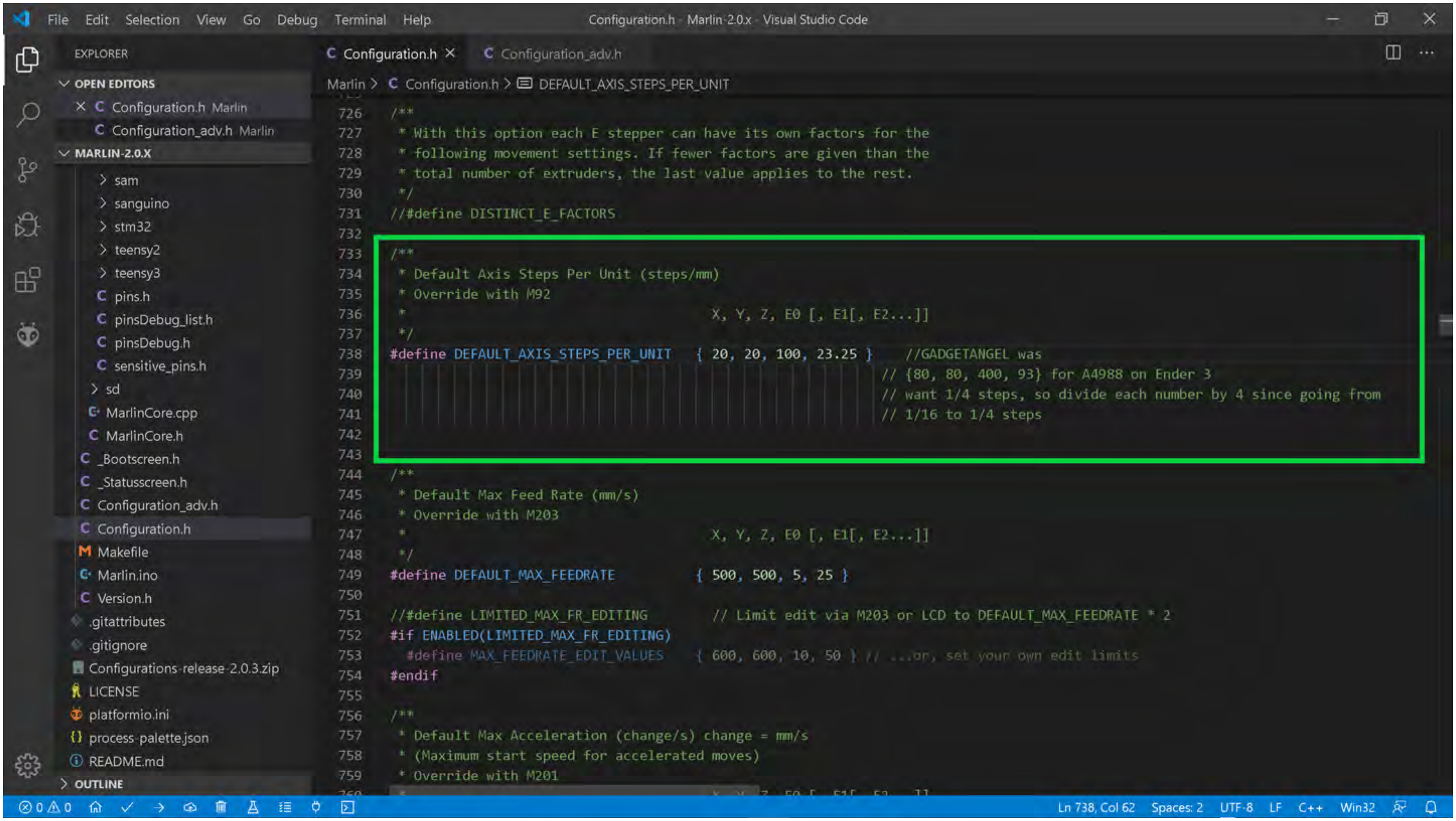
- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2225 drivers in stand-alone mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208_STANDALONE" for the DRIVER_TYPE. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

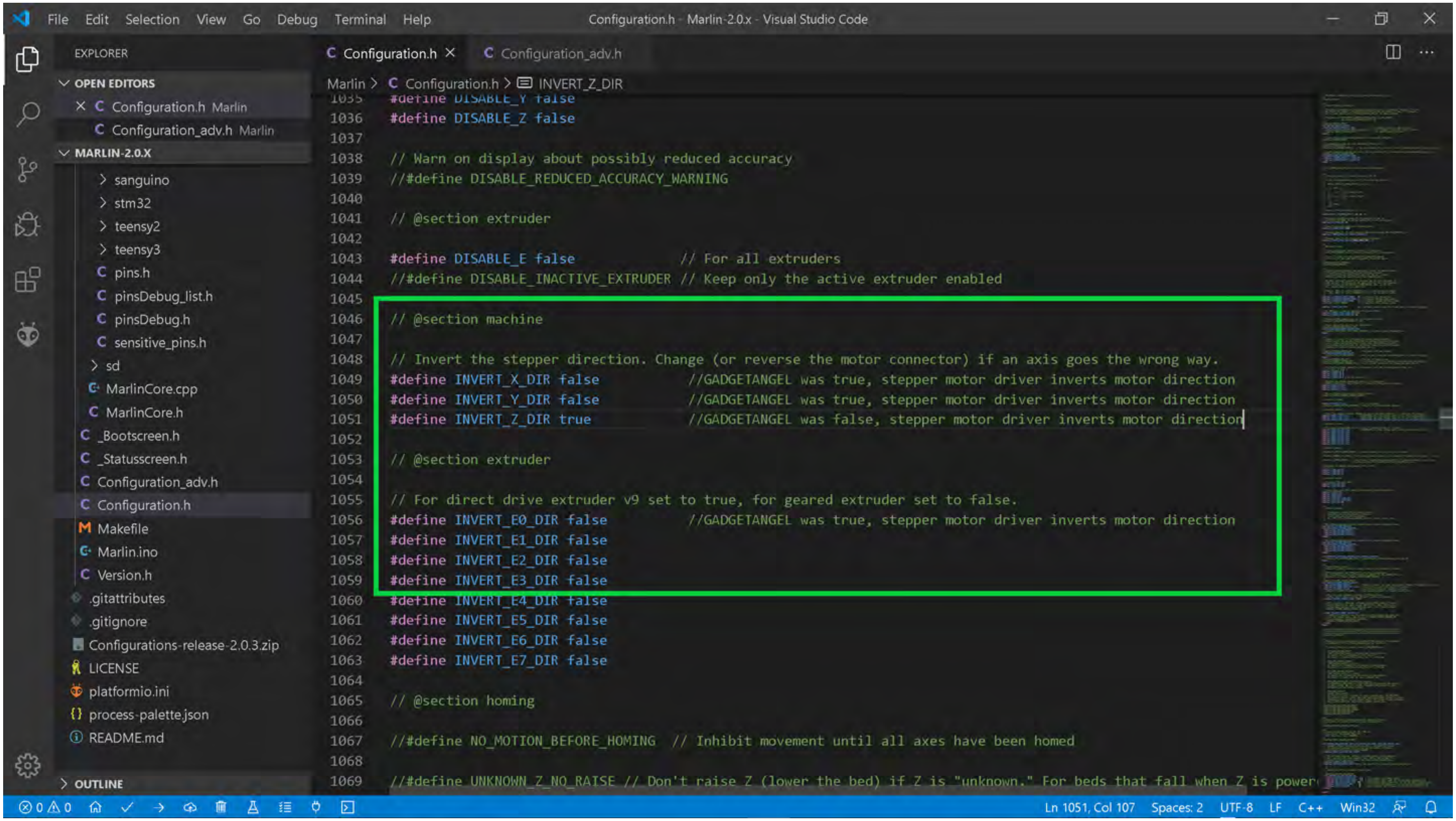
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

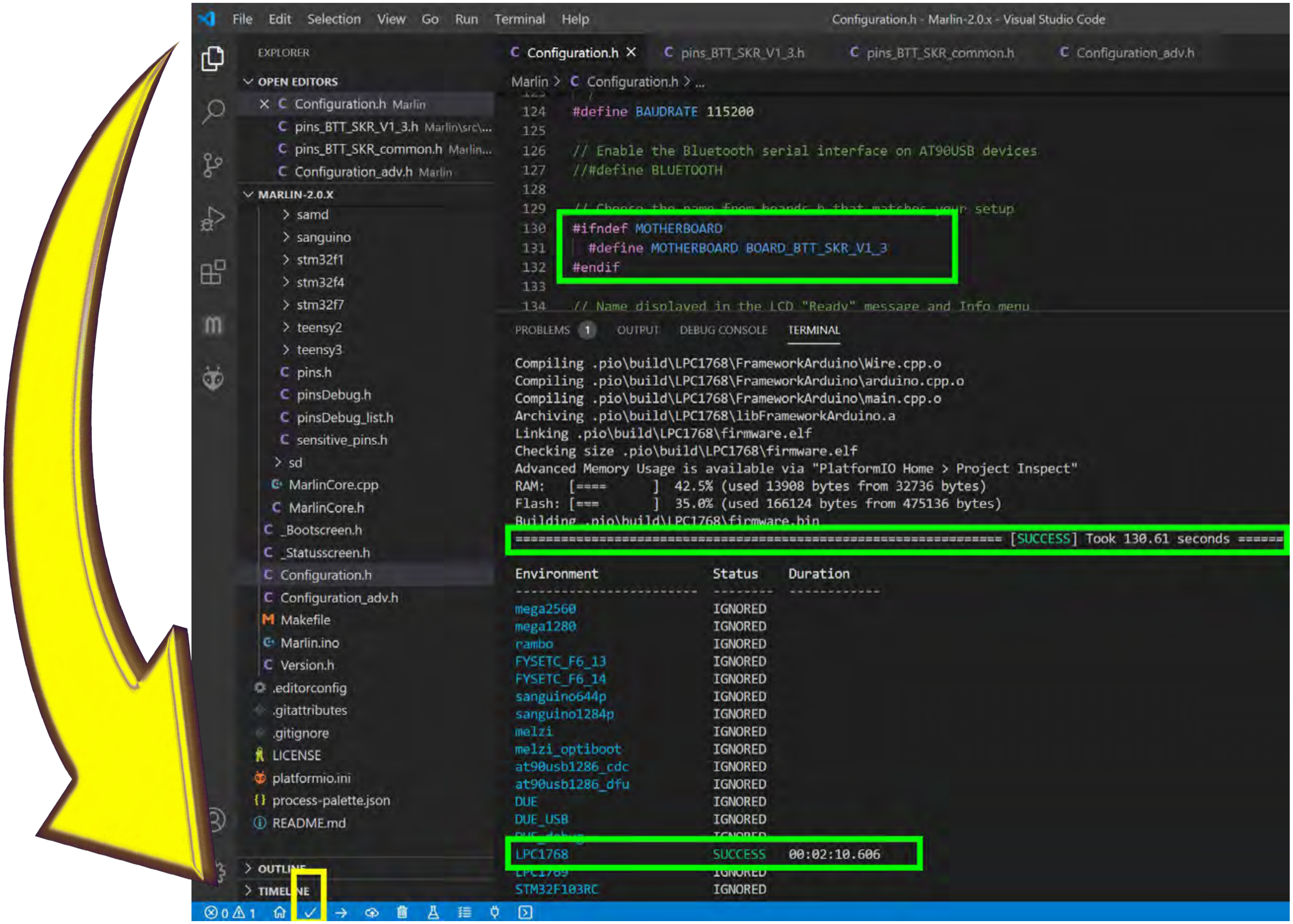
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2225 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

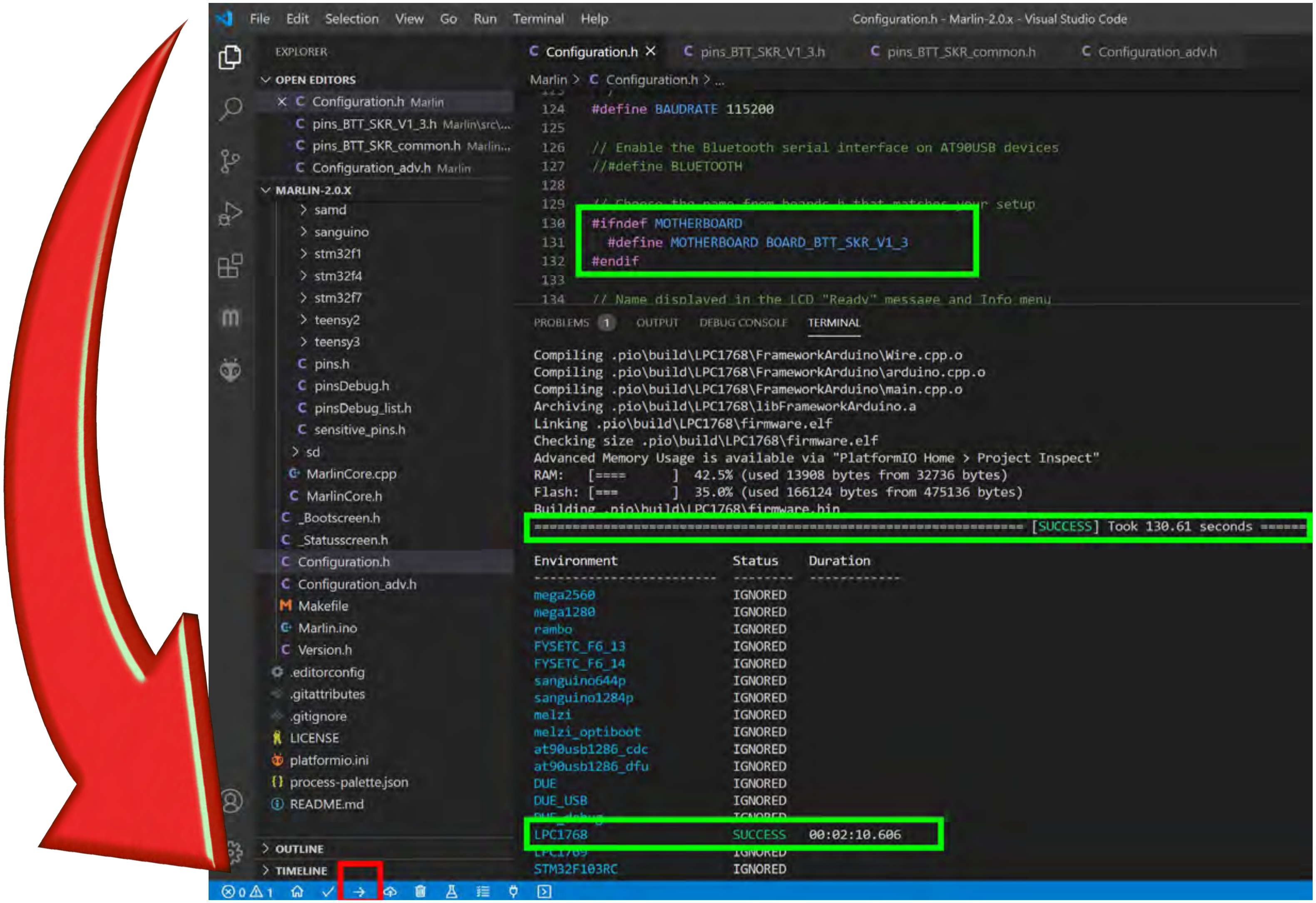
- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2225 V1.0

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card

	0	1
4	0	0
8	1	0
16	0	1
32	1	1

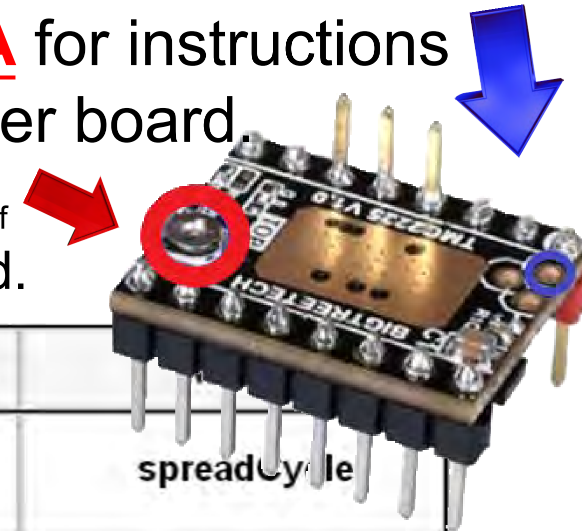
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

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

BIQU TMC2225 V1.0

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

base of card

	0	1
4	0	0
8	1	0
16	0	1
32	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

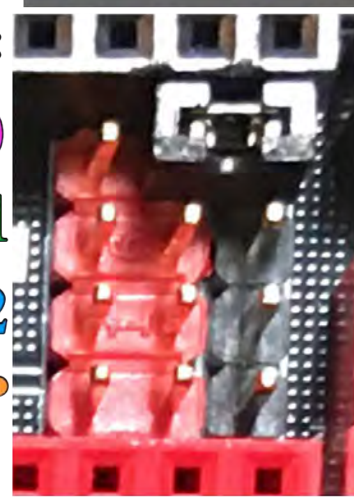
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇒ No Jumper

Rows:
MS0
MS1
MS2
RST/SLP

High ⇒ Jumper Set

Rows:
MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU TMC2225 V1.0

One Time Programming (OTP) Mode

OTP (One Time Programming Mode)

	0	1
4	0	0
8	1	0
16	0	1
32	1	1

STATUS: Not Yet Tested

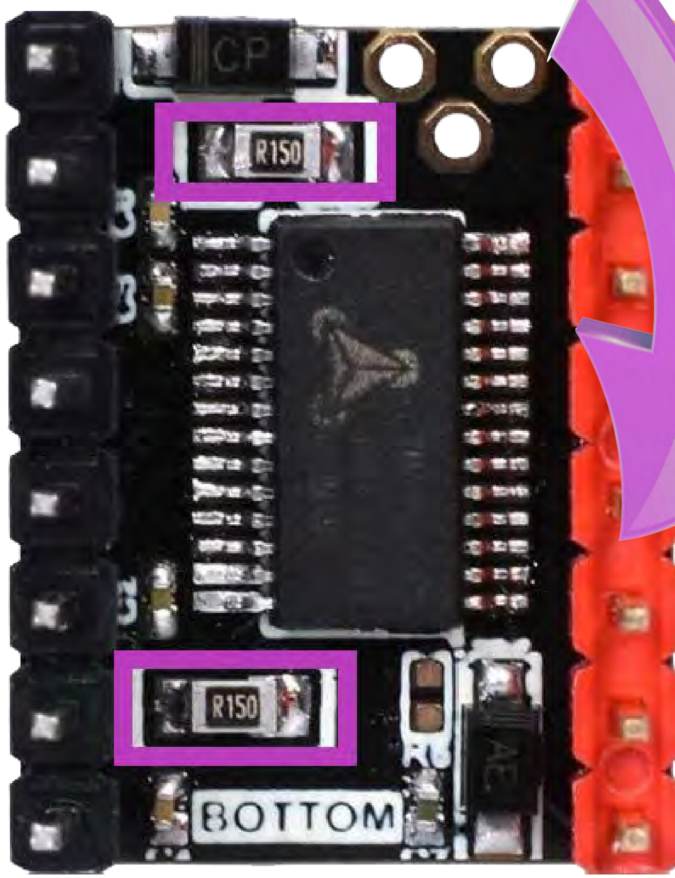
See [Appendix D](#) for legend

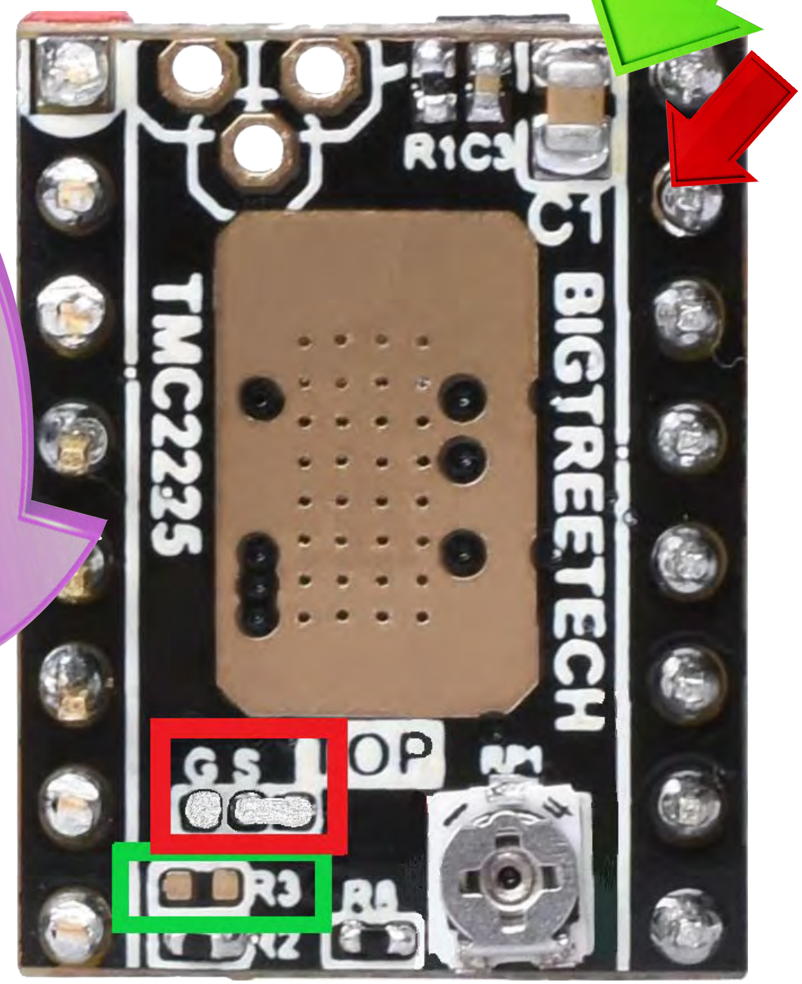
Note: Stand-alone mode by default uses stealthChop, if you want spreadCycle, you MUST use OTP mode. To obtain **One Time Programming (OTP) mode,** for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as sown in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for spread Cycle as seen in **RED** below.

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

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

Note: **MOST BIQU TMC2225 V1.0 driver boards, when purchased for UART mode, will have two R3 pads (located on the top of the driver board), which are NOT soldered together. This indicates the driver board can use the UART pin for the UART single wire interface (if the UART,"U", jumper is in place on the SKR V1.3 board)**



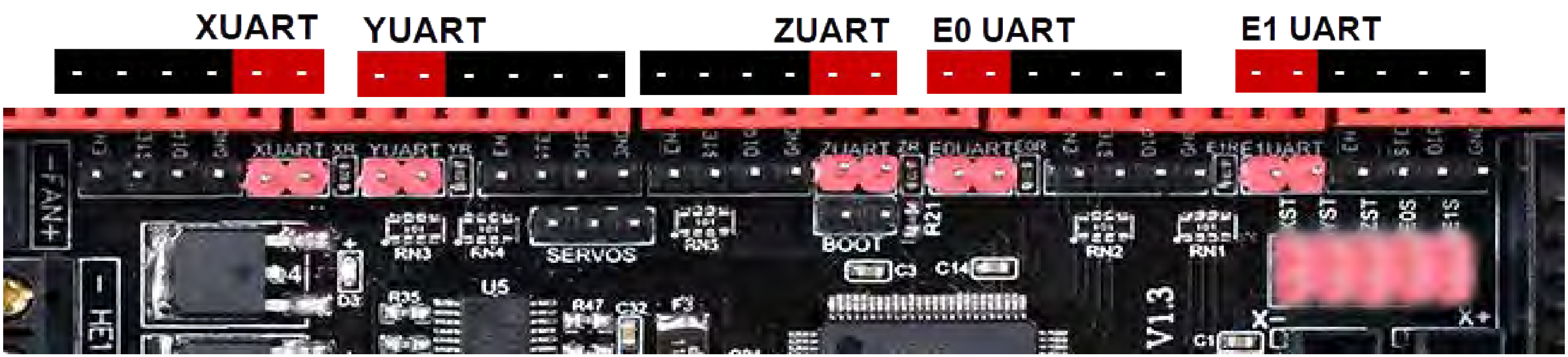


OTP (One Time Programming) Mode

One Time Programming (OTP) Mode

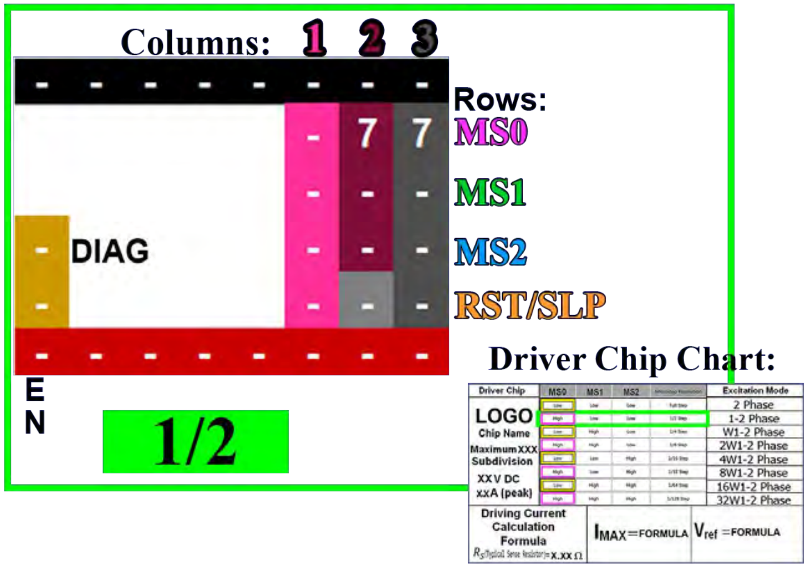
Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation.

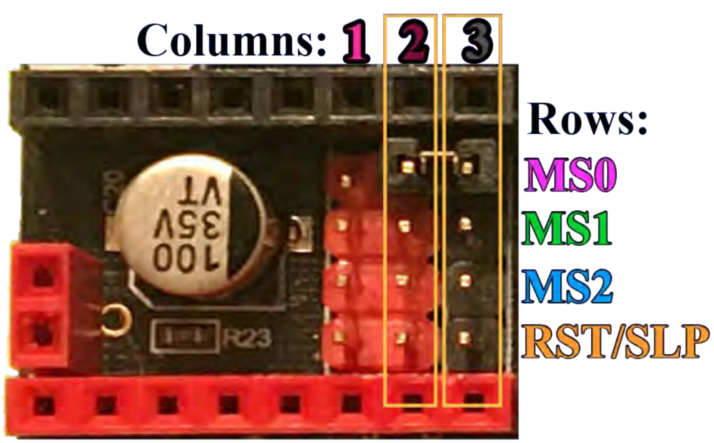


SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Chip Chart: | **Meaning:**

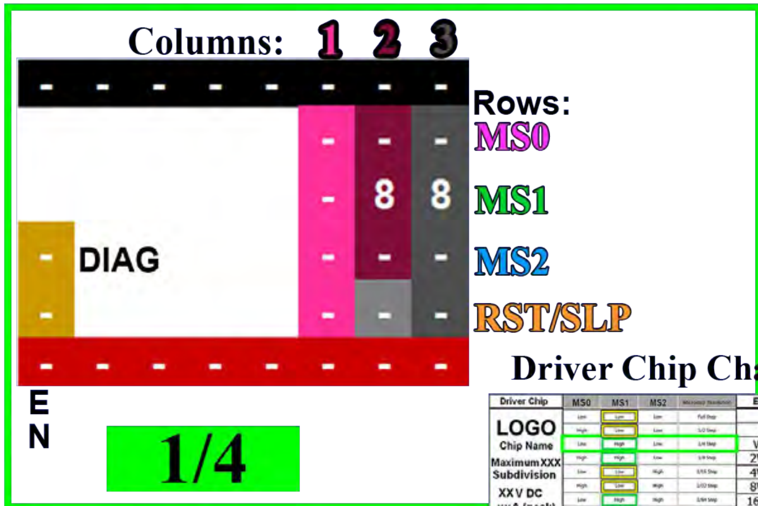
High → set Jumper between column 2 and column 3 on the MS0 row

Low → No Jumper Set

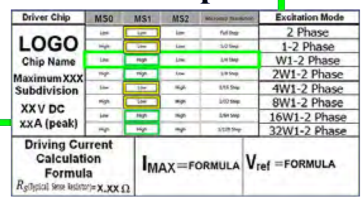
OTP (One Time Programming) Mode

One Time Programming (OTP) Mode

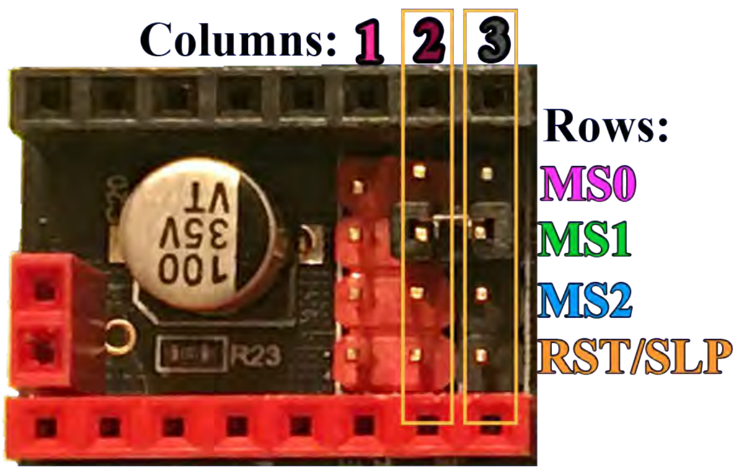
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Socket Representation: 8 8 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Driver Chip Chart:

Meaning:

High → set **Jumper** between column 2 and column 3 on the MS1 row

Low → No **Jumper** Set

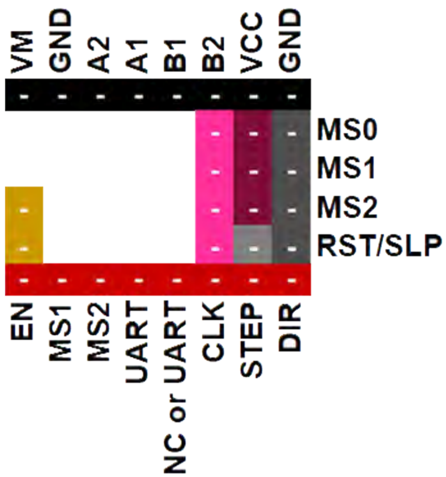
OTP (One Time Programming) Mode

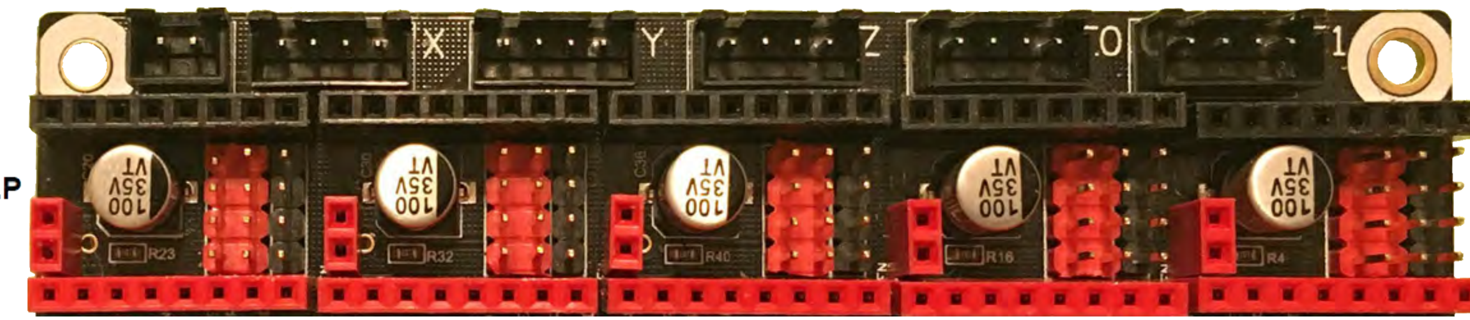
OTP

1 / 4

Interpolation:
1 / 256

SpreadCycle





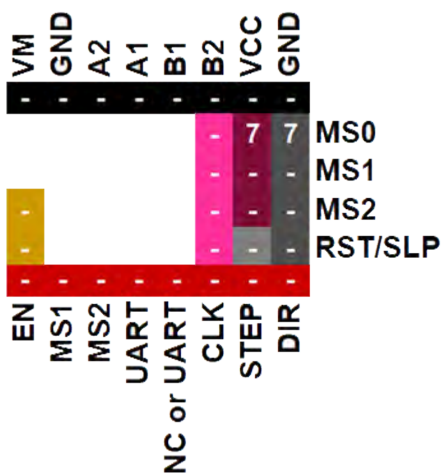
See <u>Appendix D</u> for legend

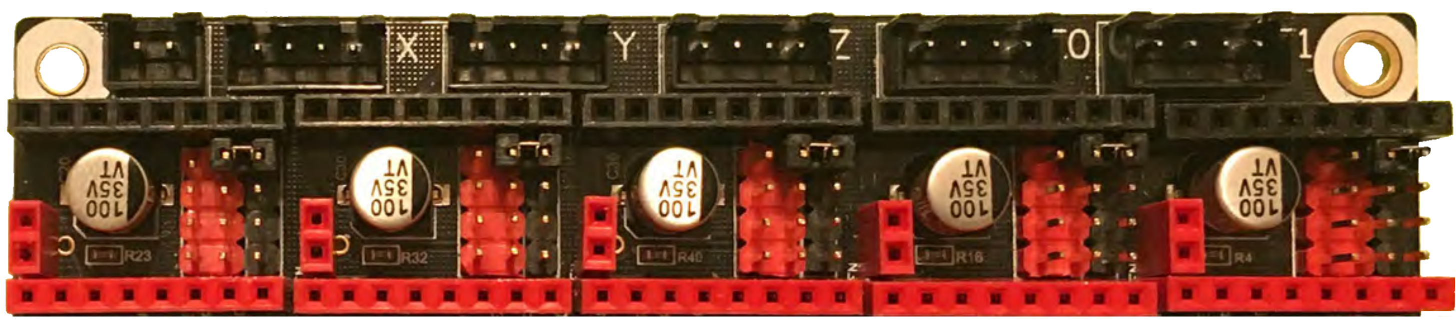
OTP

1 / 8

Interpolation:
1 / 256

SpreadCycle

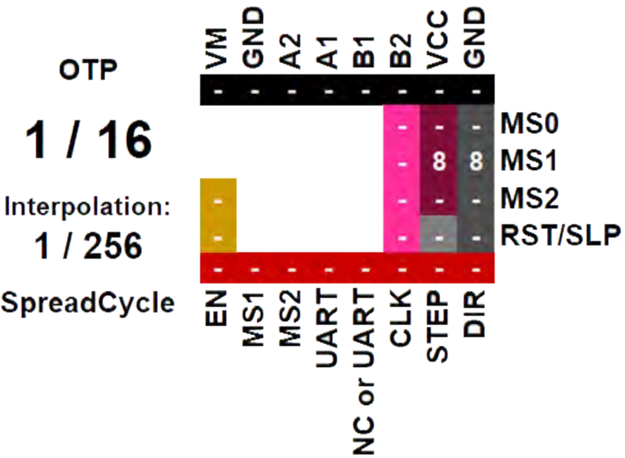


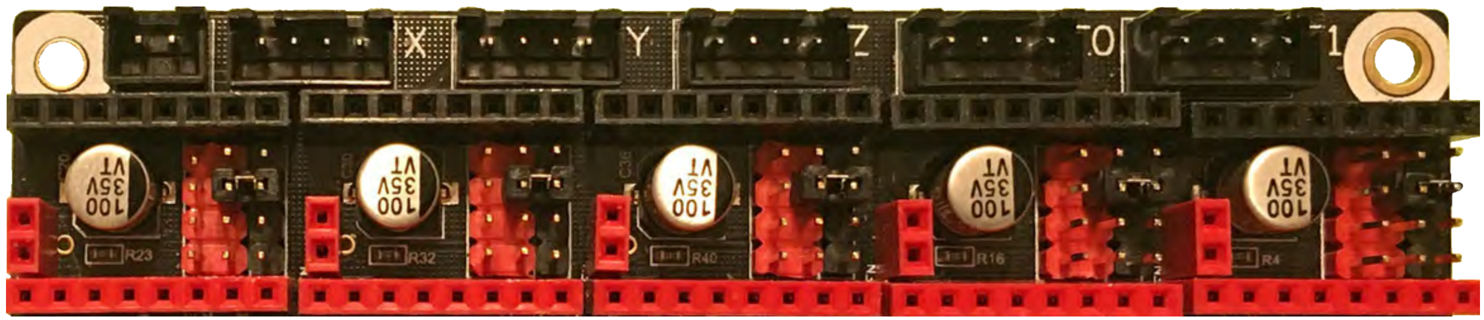


See <u>Appendix D</u> for legend

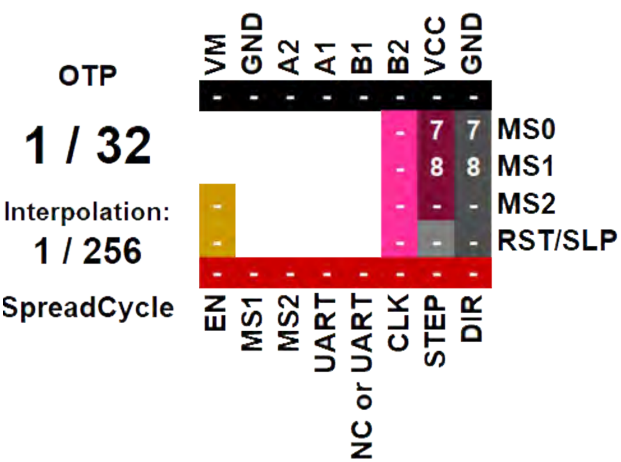
OTP (One Time Programming) Mode

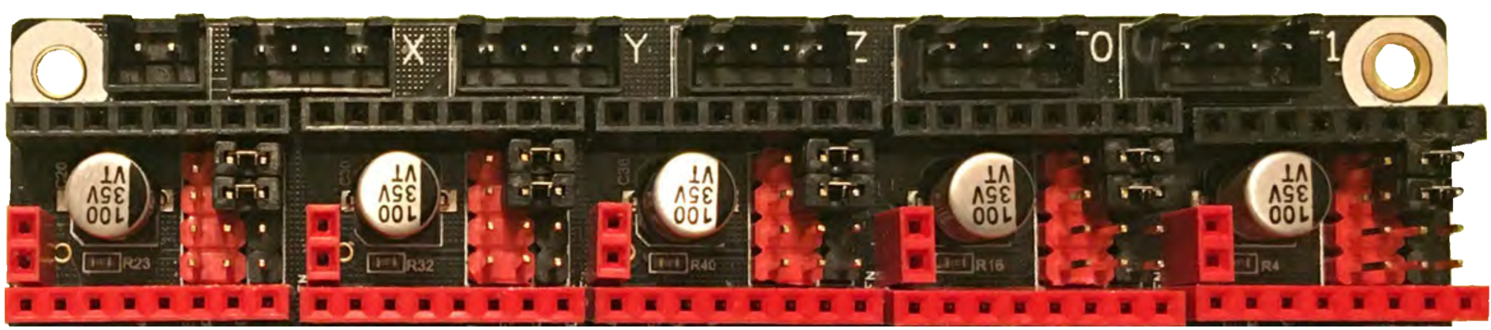
One Time Programming (OTP) Mode





See <u>Appendix D</u> for legend



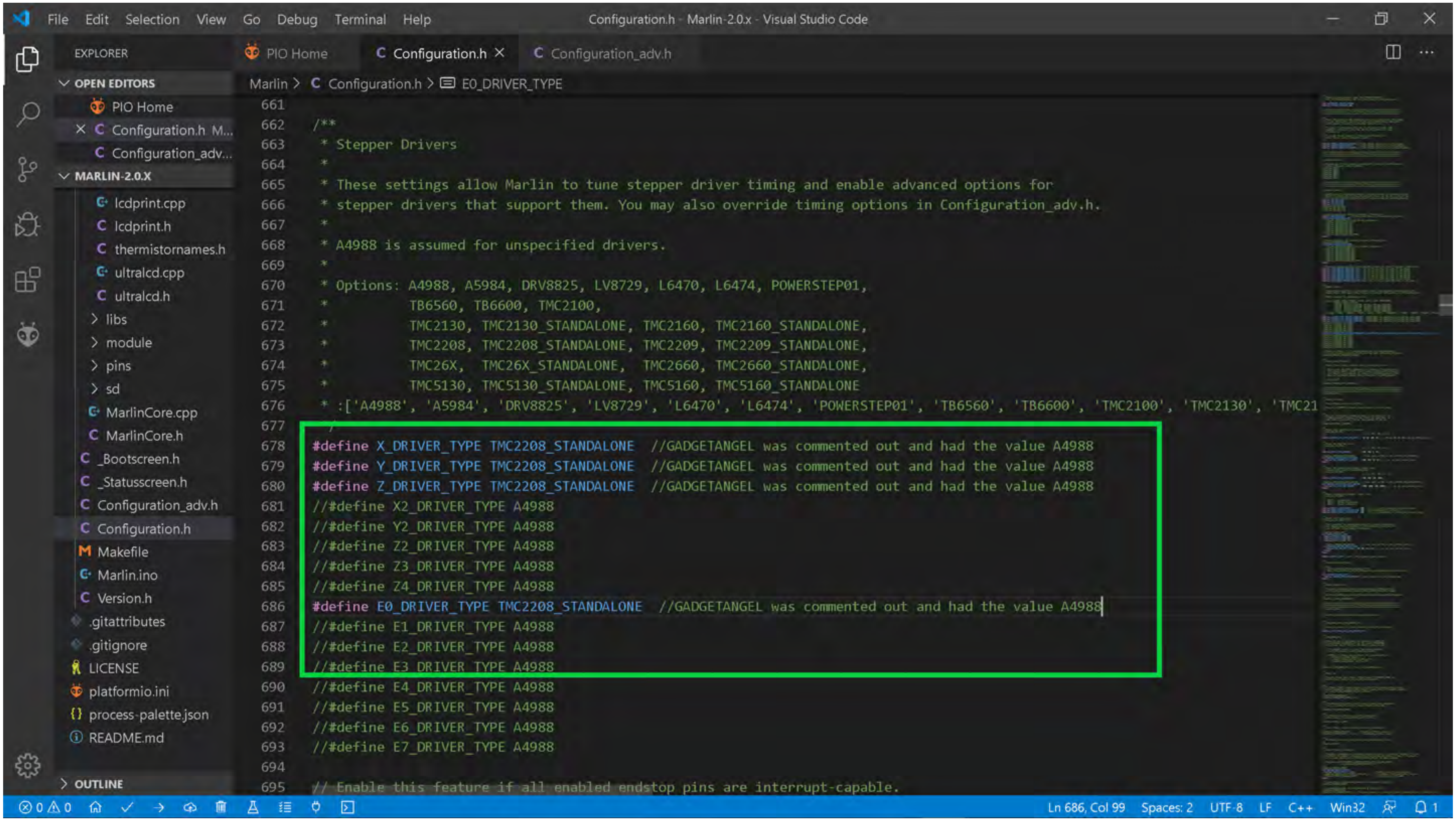


See <u>Appendix D</u> for legend

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

NOTE: **Go to Appendix C**, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in OTP mode.

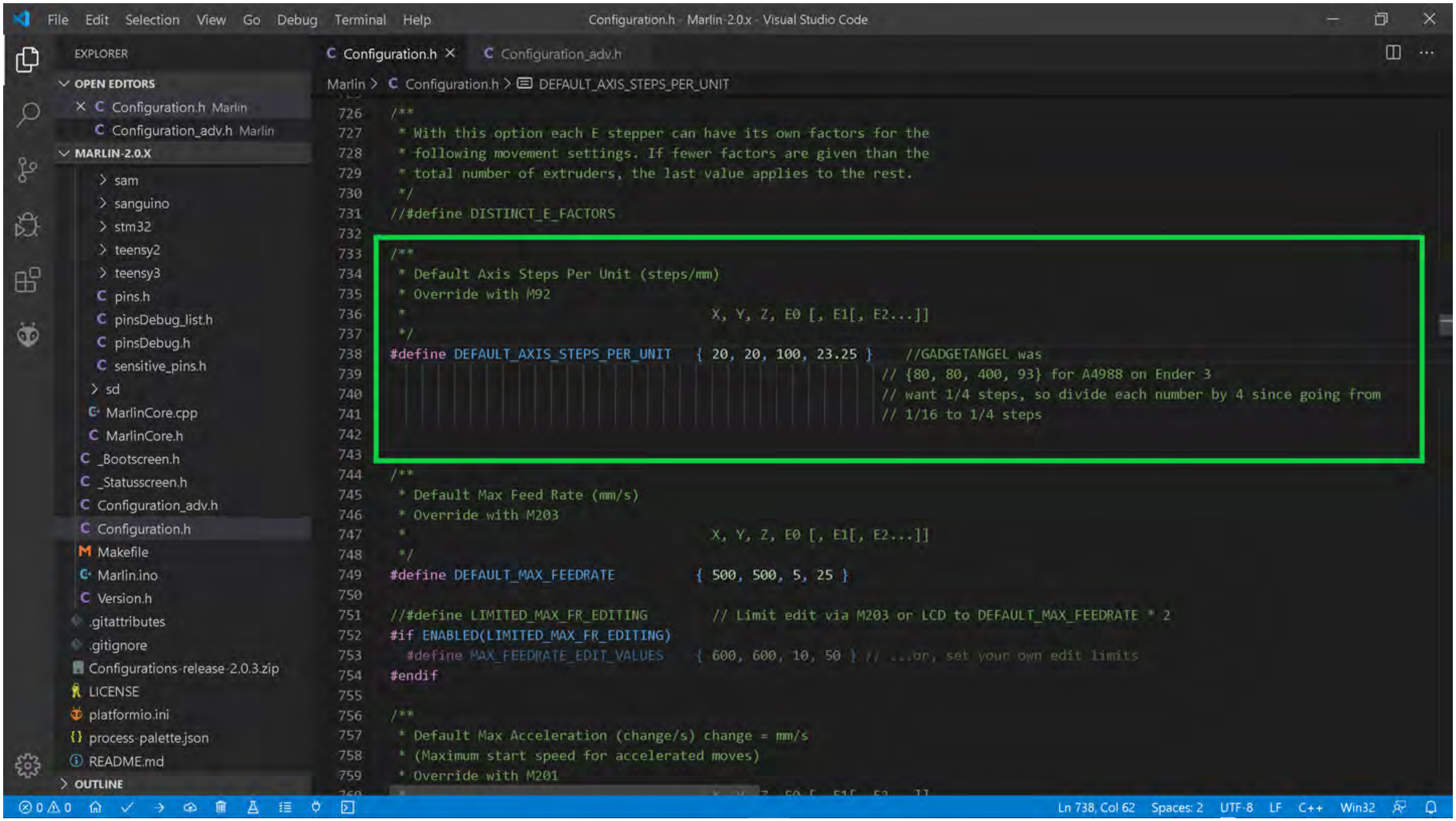
- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU TMC2225 drivers in OTP mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208_STANDALONE" for the DRIVER_TYPE. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

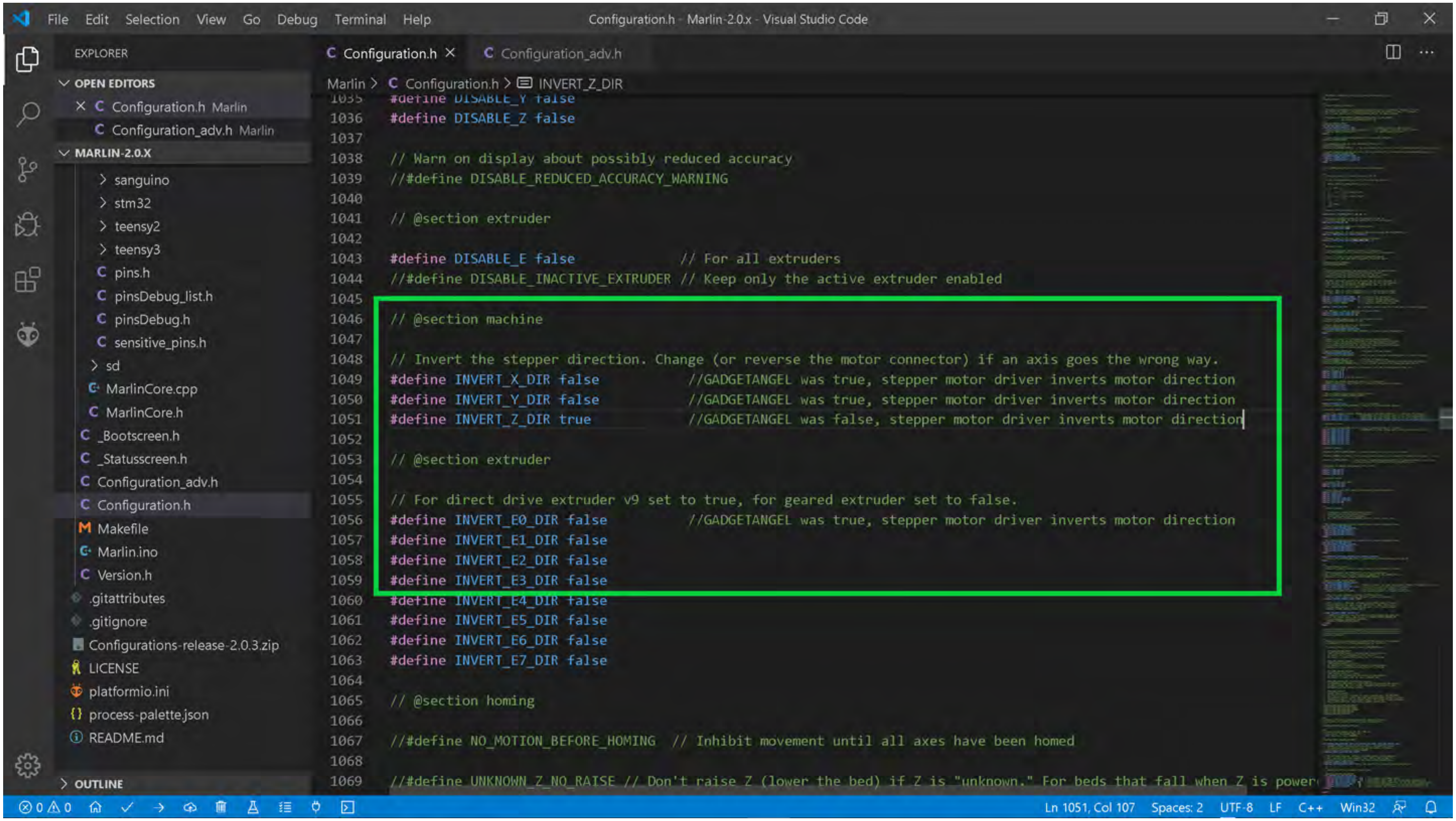
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the **GREEN** box below.



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

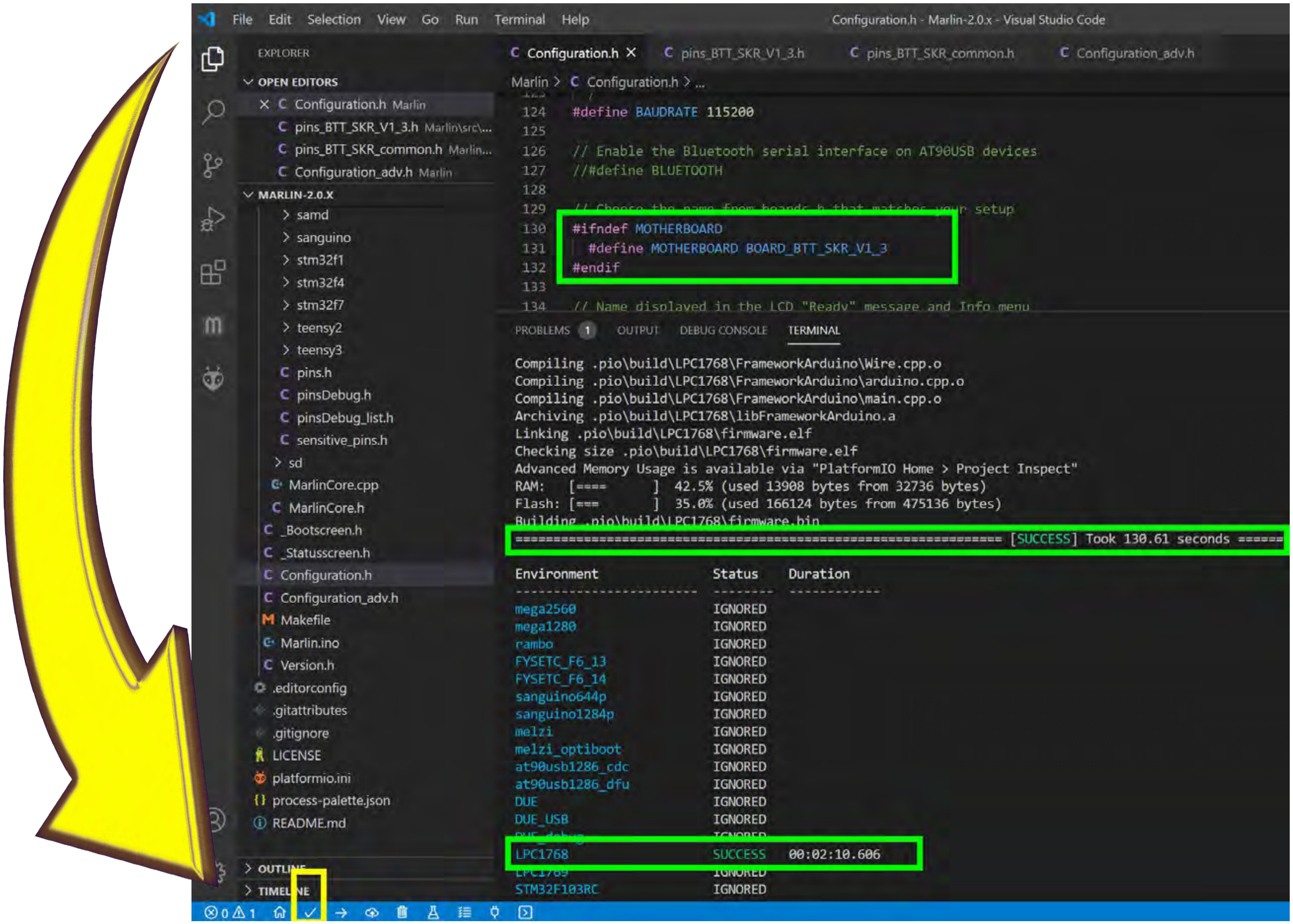
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

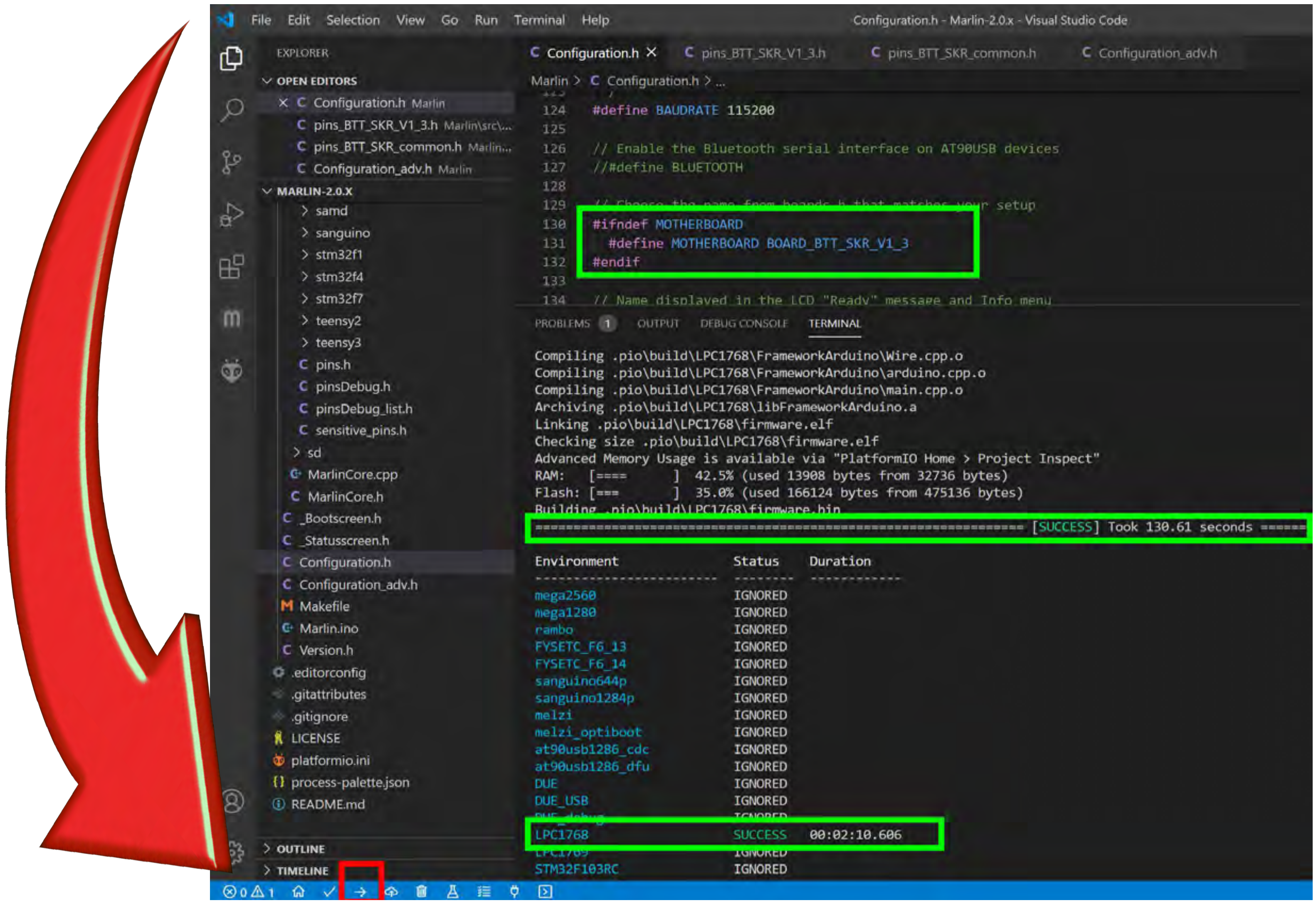
- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



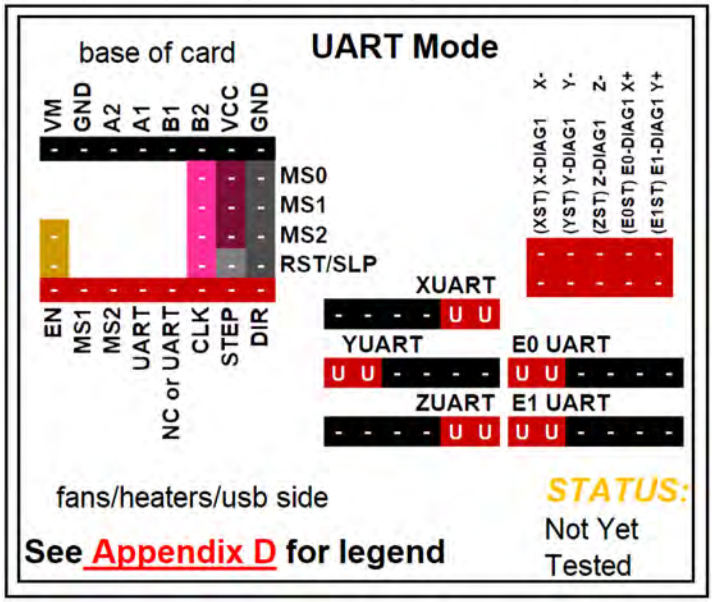
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.



BIQU TMC2225 V1.0

UART Mode

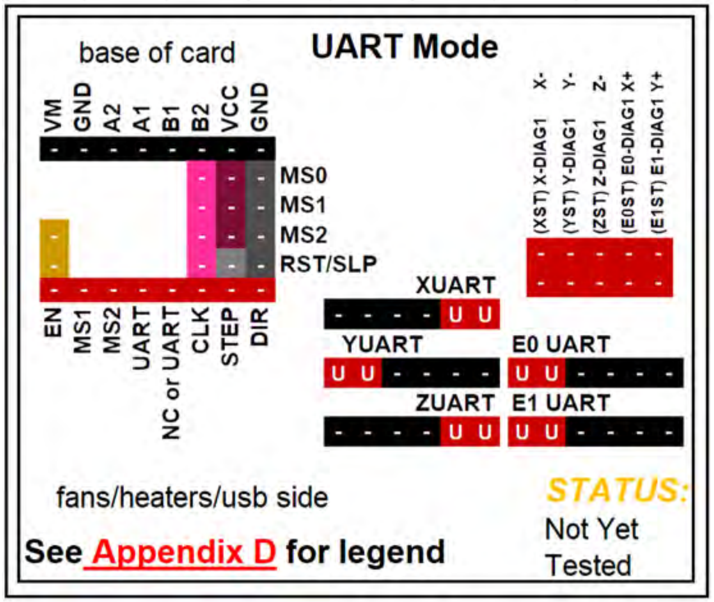
Note: You can use 50% to 90% of the calculated I_{RMS} (I_{MAX}/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

Driver Chip		
BIQU® TMC2225 UART Mode Maximum 256 Subdivision 35V DC 2A (peak)	Steps are set inside of your Firmware	
Driving Current Calculation Formula R_S(Typical Sense Resistor)=0.15Ω	$I_{MAX}=V_{ref}*0.7647$ See Appendix B #7. Use 50% to 90% as shown below: $I_{MAX}=(V_{ref}*0.7647)*0.90$	$V_{ref}=I_{MAX}*1.3077$ See Appendix B #7. Use 50% to 90% as shown below: $V_{ref}=(I_{MAX}*1.3077)*0.90$

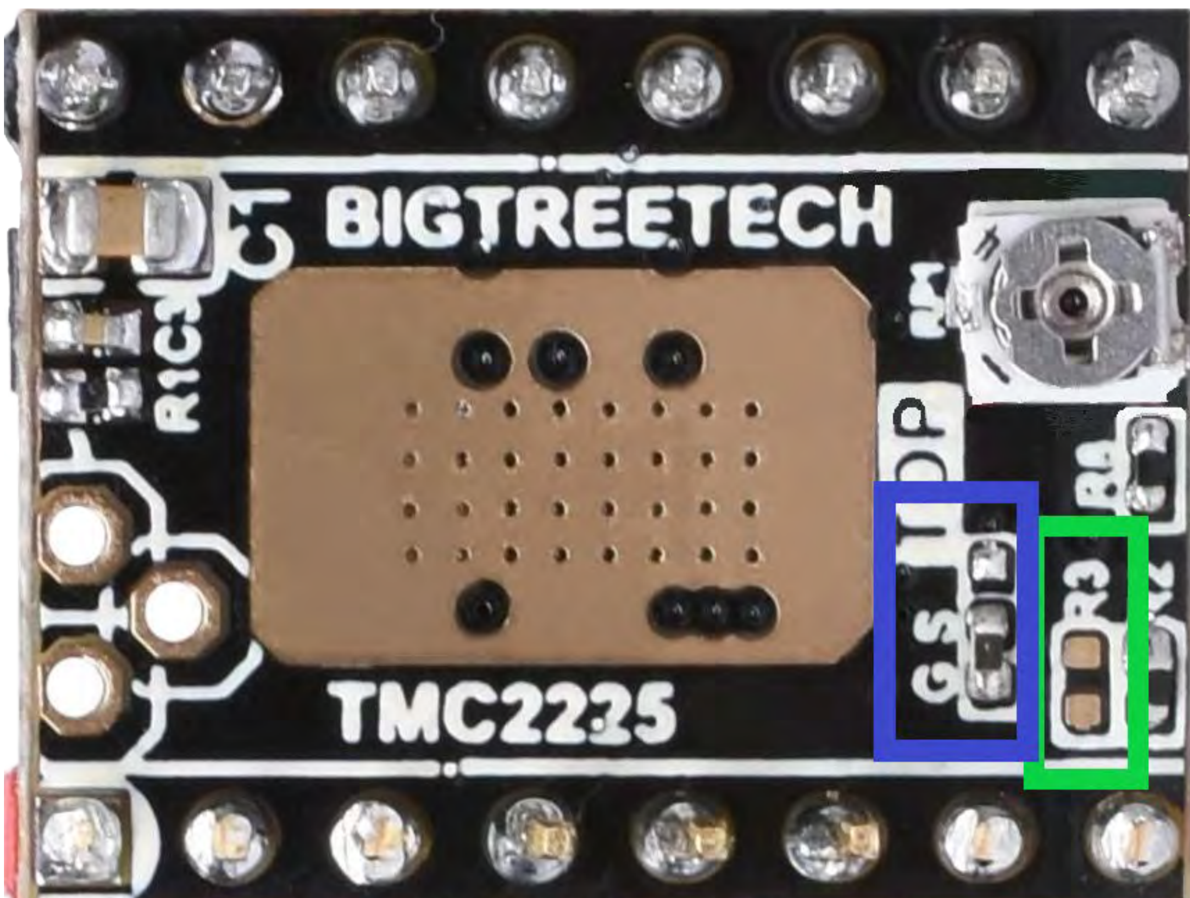
BIQU TMC2225 V1.0

UART Mode



See Appendix D for legend

Important: To ensure that the BIQU TMC2225 V1.0 is in UART Mode, check to see if the two pads located at R3 have a gap between them, as seen in GREEN below. Ensure the "U" jumper is in place on the SKR V1.3 board.



The **BLUE** box show the device has StealthChop capability.

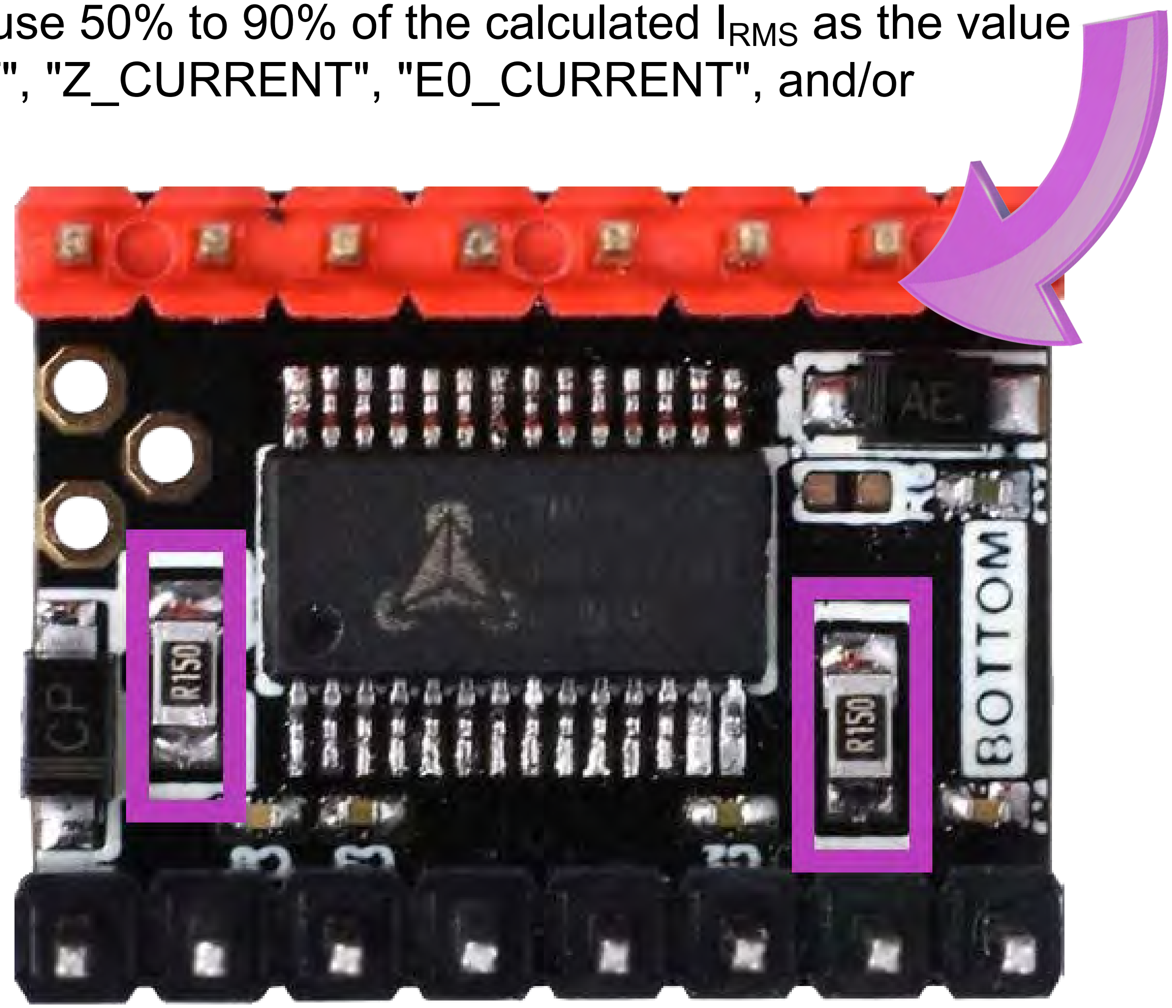
UART Mode

BIQU TMC2225 V1.0

UART Mode

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_CURRENT".

R_s = R050 is 0.05 Ohms
R_s = R062 is 0.062 Ohms
R_s = R068 is 0.068 Ohms
R_s = R075 is 0.075 Ohms
R_s = R100 is 0.1 Ohms
R_s = R110 is 0.11 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms



UART Mode

BIQU TMC2225 V1.0

UART Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to ensure proper operation for UART mode.

Note: Set the "U" Jumper(s) for UART MODE!

XUART: - - - - U U

YUART: U U - - - -

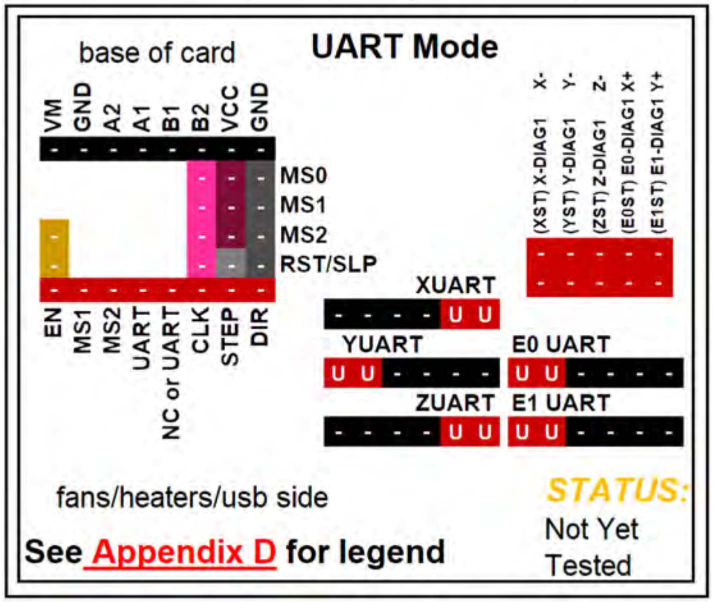
ZUART: - - - - U U

E0 UART: U U - - - -

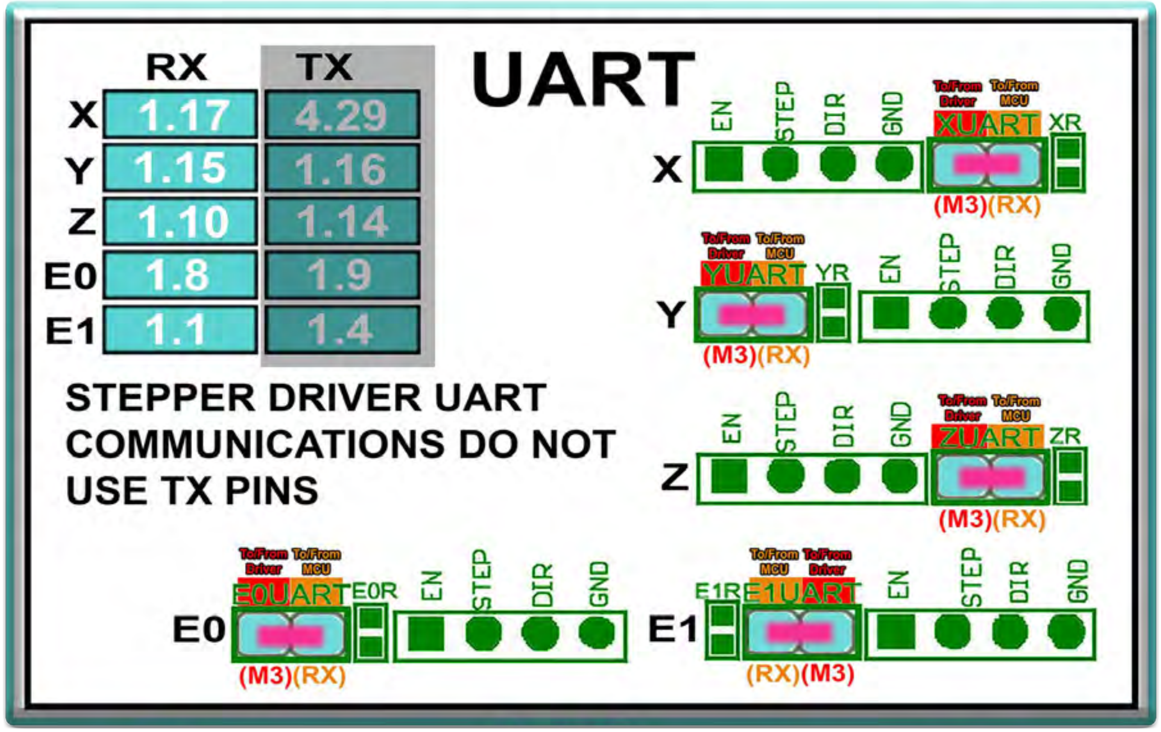
E1 UART: U U - - - -

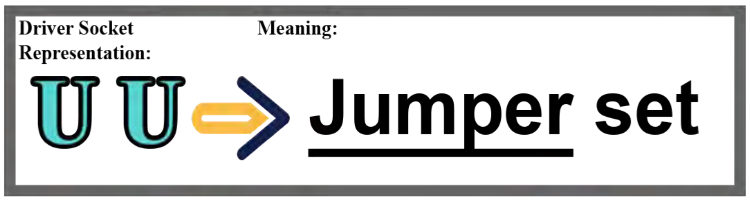
BIQU TMC2225 V1.0

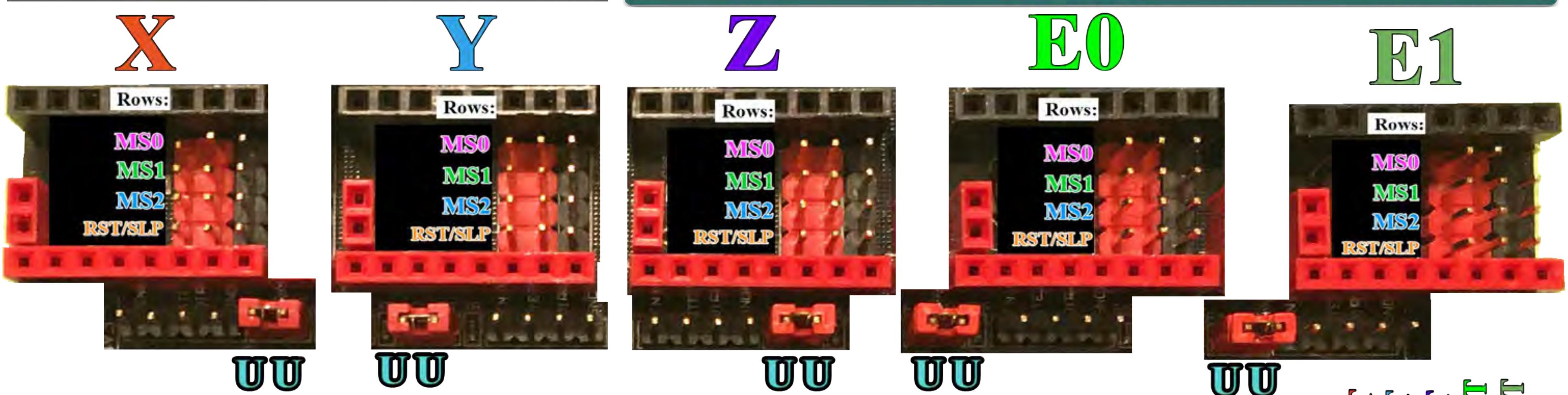
UART Mode



SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode



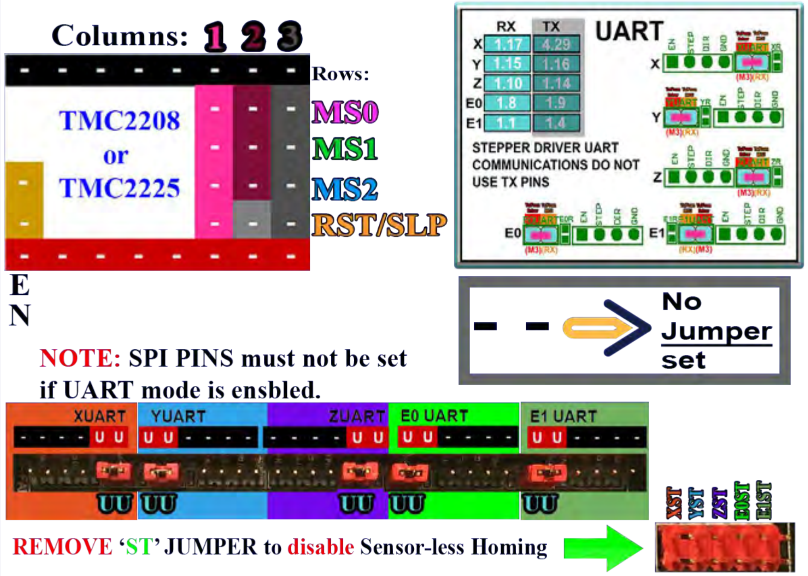


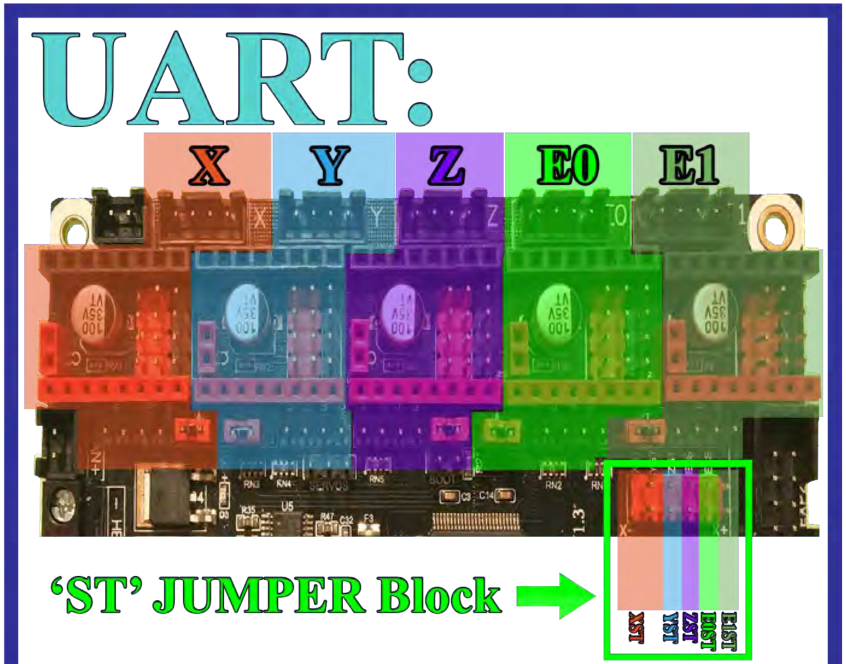


REMOVE 'ST' JUMPER(s) to disable Sensor-less Homing – for drivers without stallGaurd™ feature

SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode

Driver Socket Representation





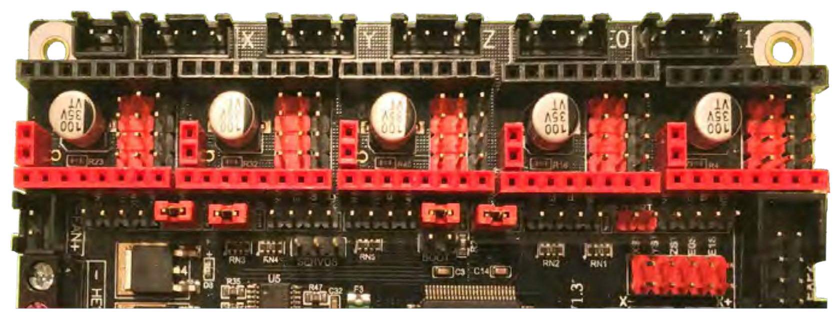
UART Capable Drivers **without** stallGaurd™ feature: TMC2208, and TMC2225

Driver Socket Representation:	Meaning:
U U →	**Jumper Set**

Driver Socket Representation:	Meaning:
U U →	set **Jumper** so that both U PINS are covered by the **Jumper**

Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS REMOVED. All other PINS need to be empty for UART mode to work properly.

Here is an example of only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS removed for ALL Axes:



BIQU TMC2225 V1.0

UART Mode

Information on Sensor-less Homing

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

NOTE: Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.

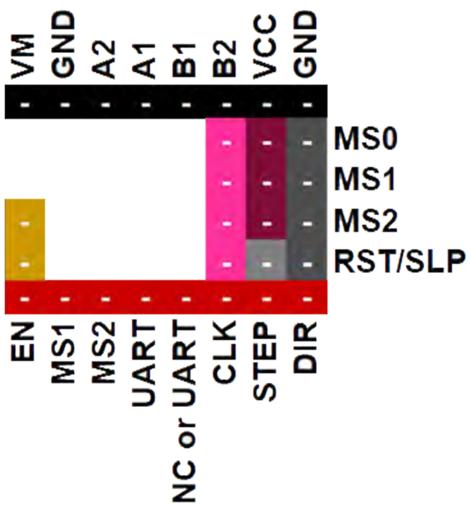
UART Mode

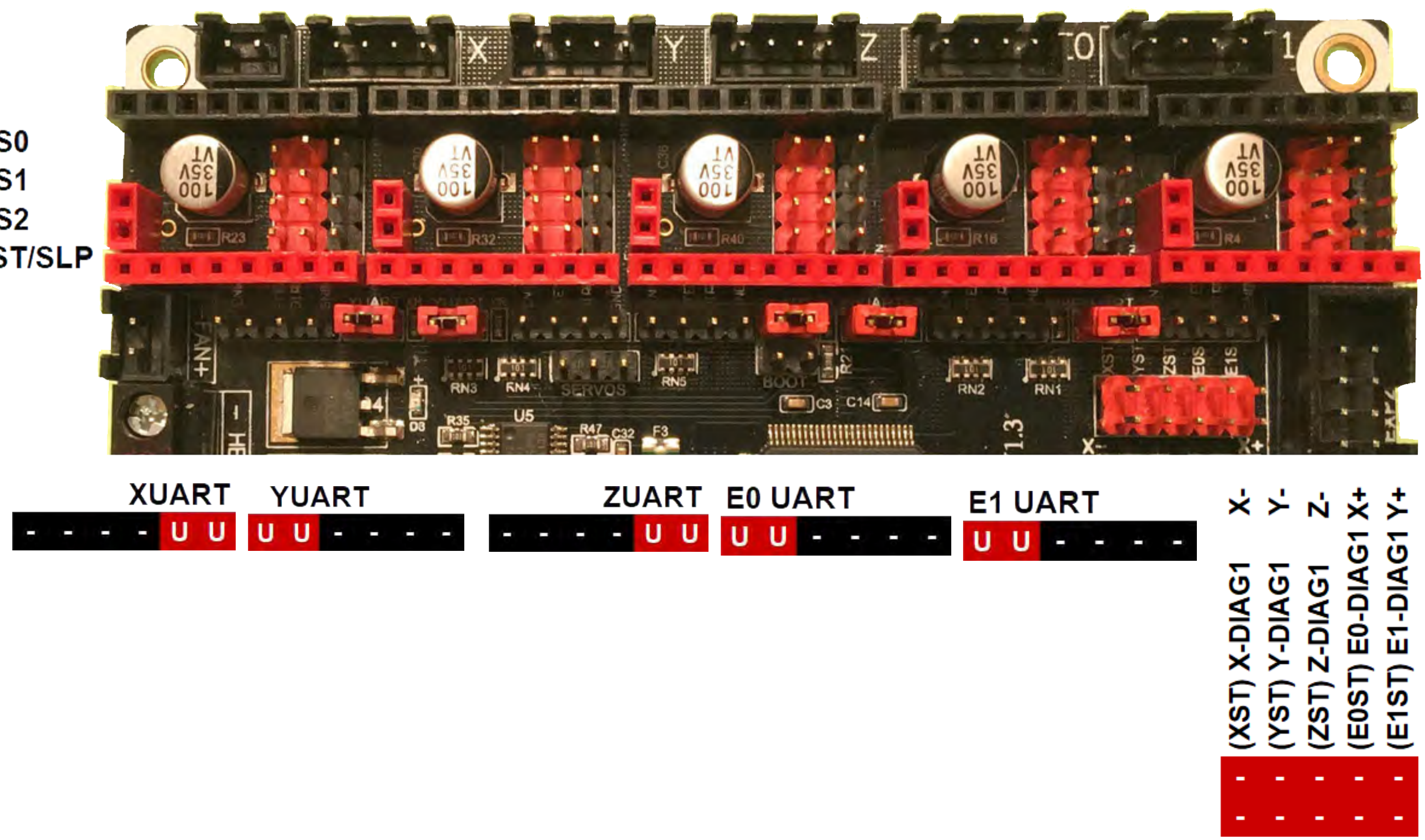
BIQU TMC2225 V1.0

UART Mode

Note: Set the "U" Jumper(s) for UART MODE!

UART





See Appendix D for legend

Note: TMC2225 does not have sensor-less homing capability.

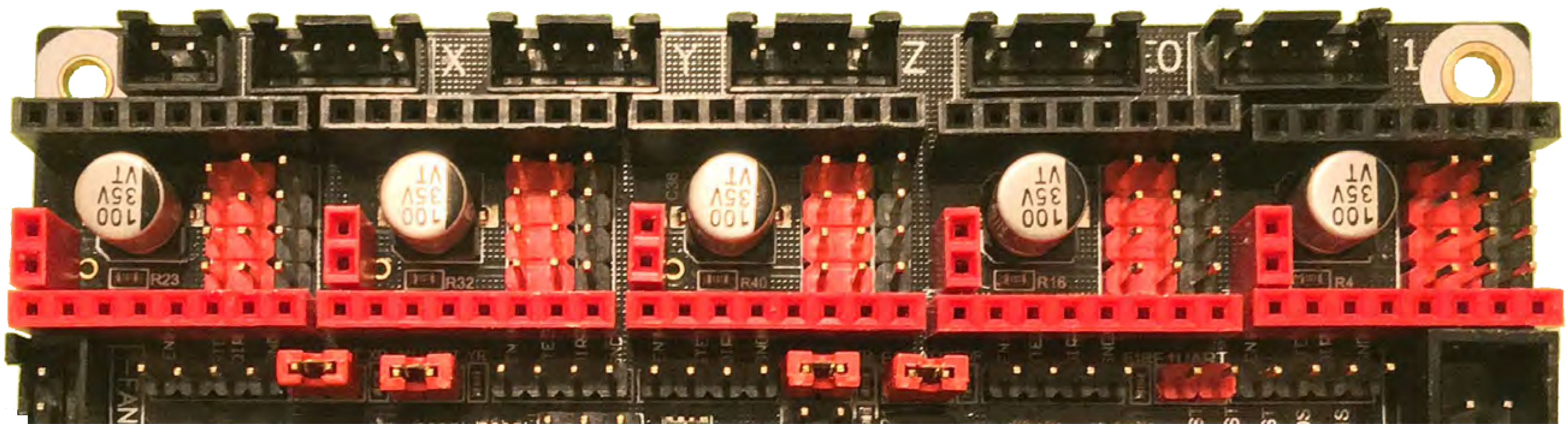
UART Mode

BIQU TMC2225 V1.0

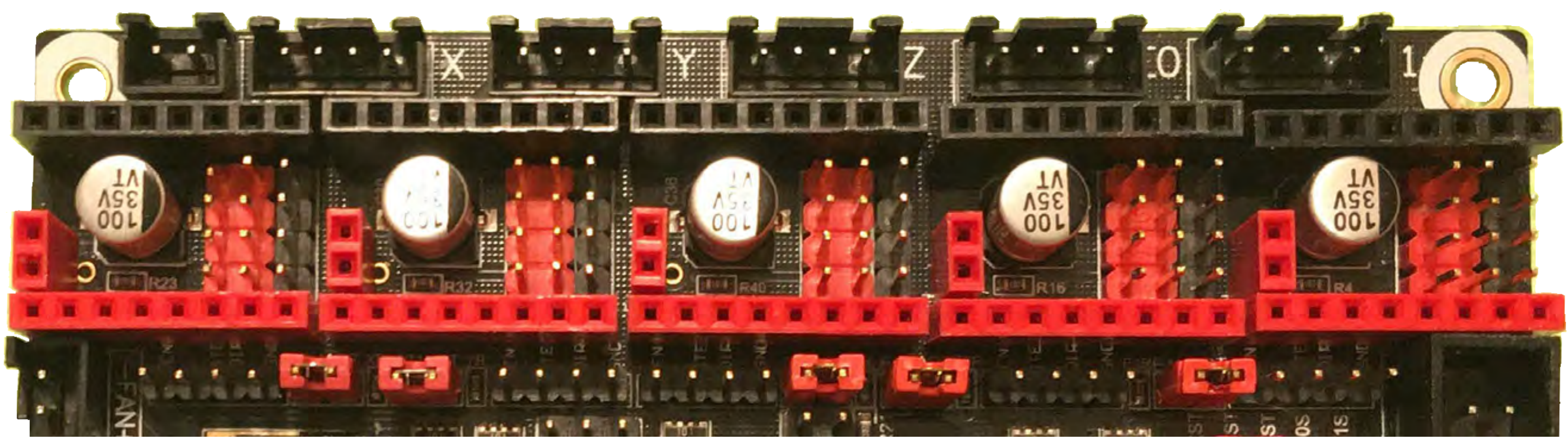
UART Mode

Examples of Different UART Configurations

X, Y, Z, and E0 axes configured for UART mode.



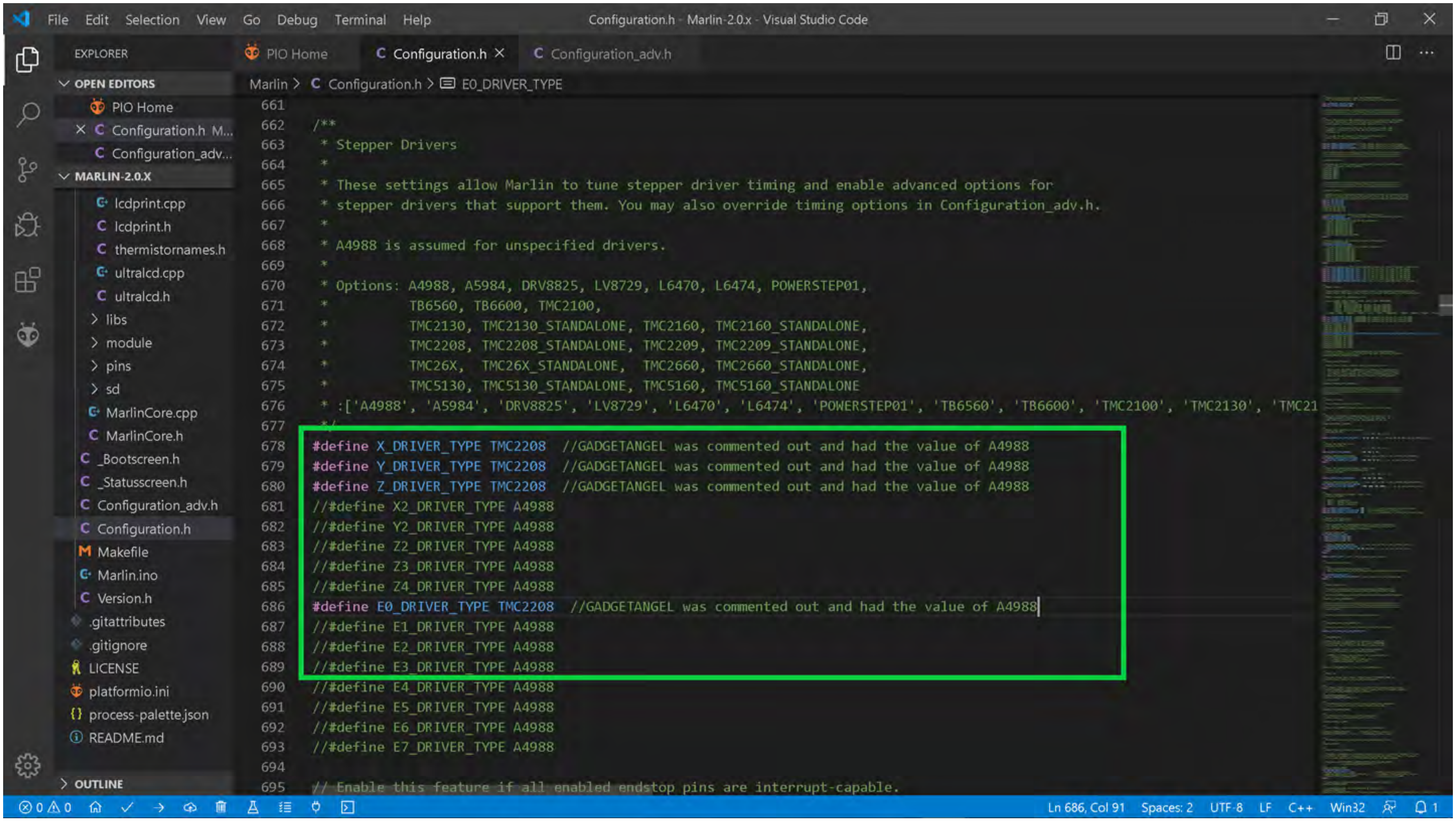
X, Y, Z, E0 and E1 axes configured for UART mode.



The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

NOTE: **Go to Appendix C**, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in UART mode.

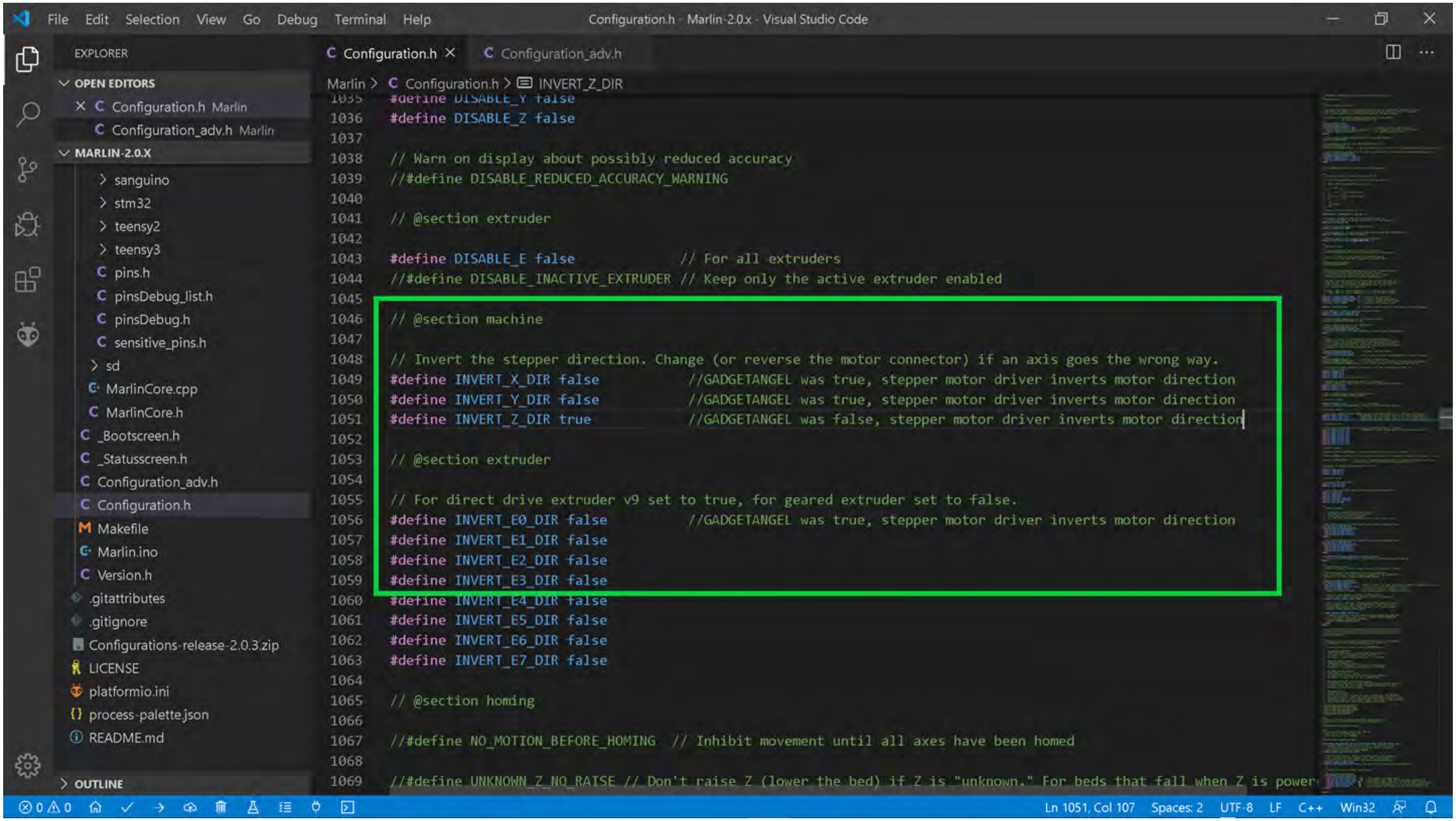
- Change the stepper motor drivers so that Marlin knows you are using TMC2225 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2225 drivers in UART mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208" for the DRIVER_TYPE. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

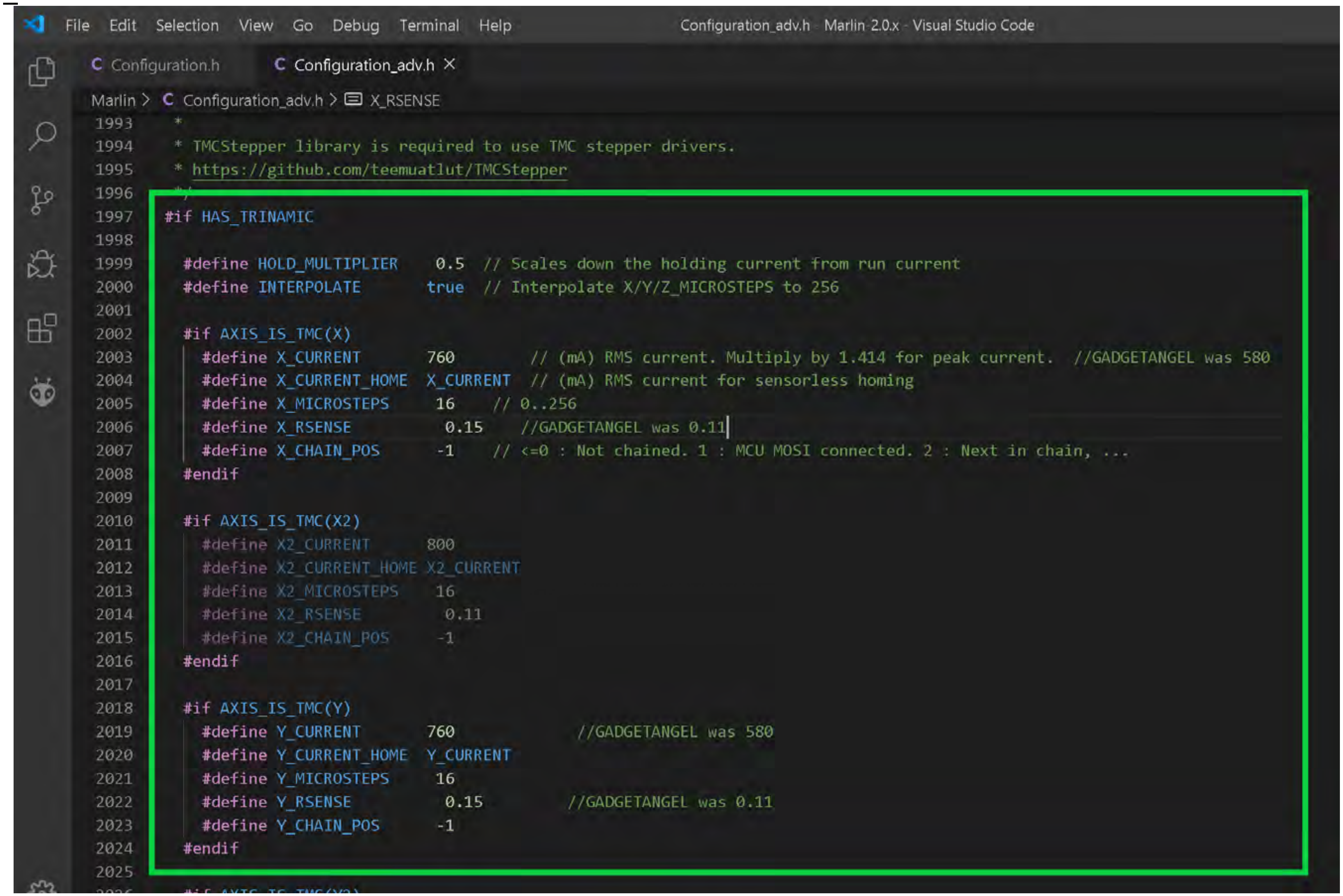
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2225 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

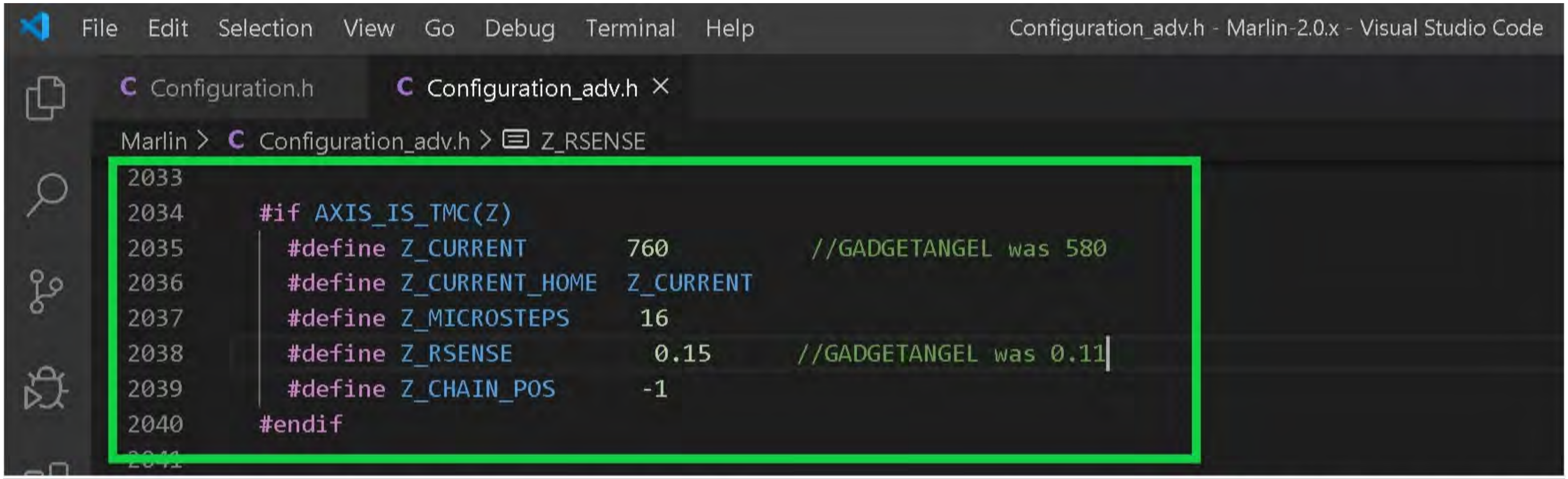
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2225 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.15. Ensure "Y_RSENSE" is set to 0.15.
- If you do not want to use V_{ref} as the value for "X_CURRENT" and/or "Y_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT" and/or "Y_CURRENT".

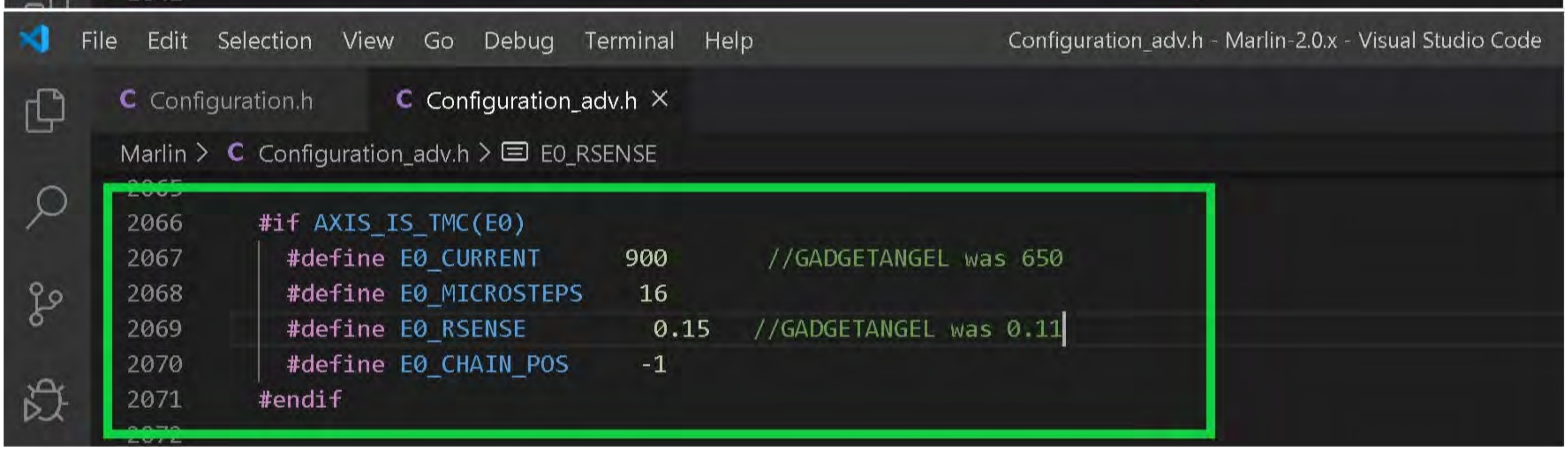


- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the **GREEN** boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z_RSENSE" is set to 0.15. Ensure "E0_RSENSE" is set to 0.15.
- If you do not want to use V_{ref} as the value for "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "Z_CURRENT" and/or "E0_CURRENT".

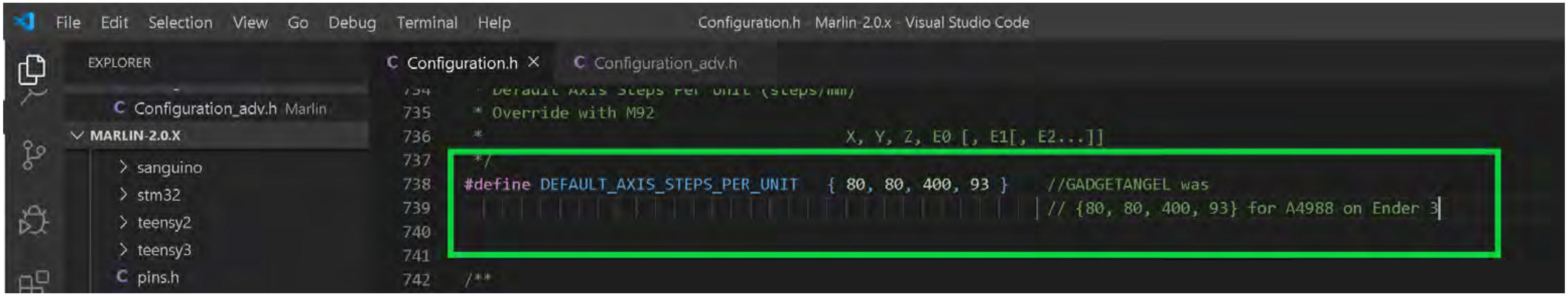




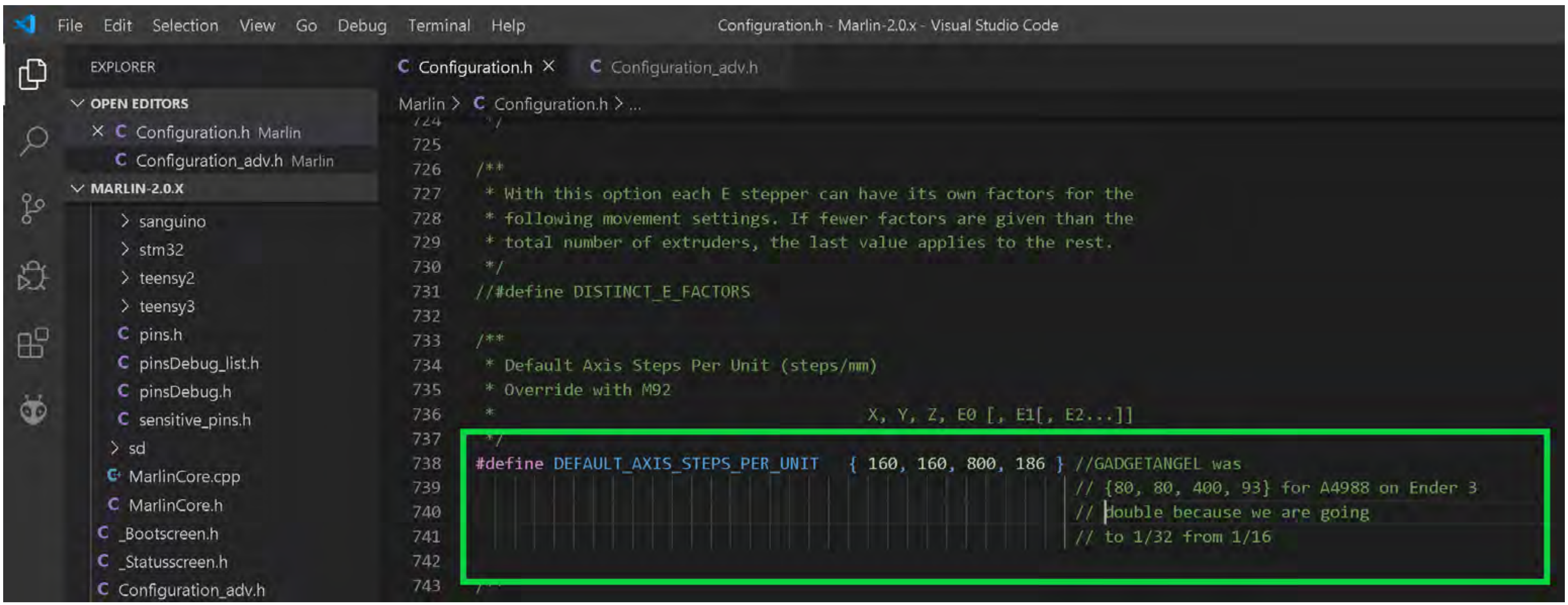
- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

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



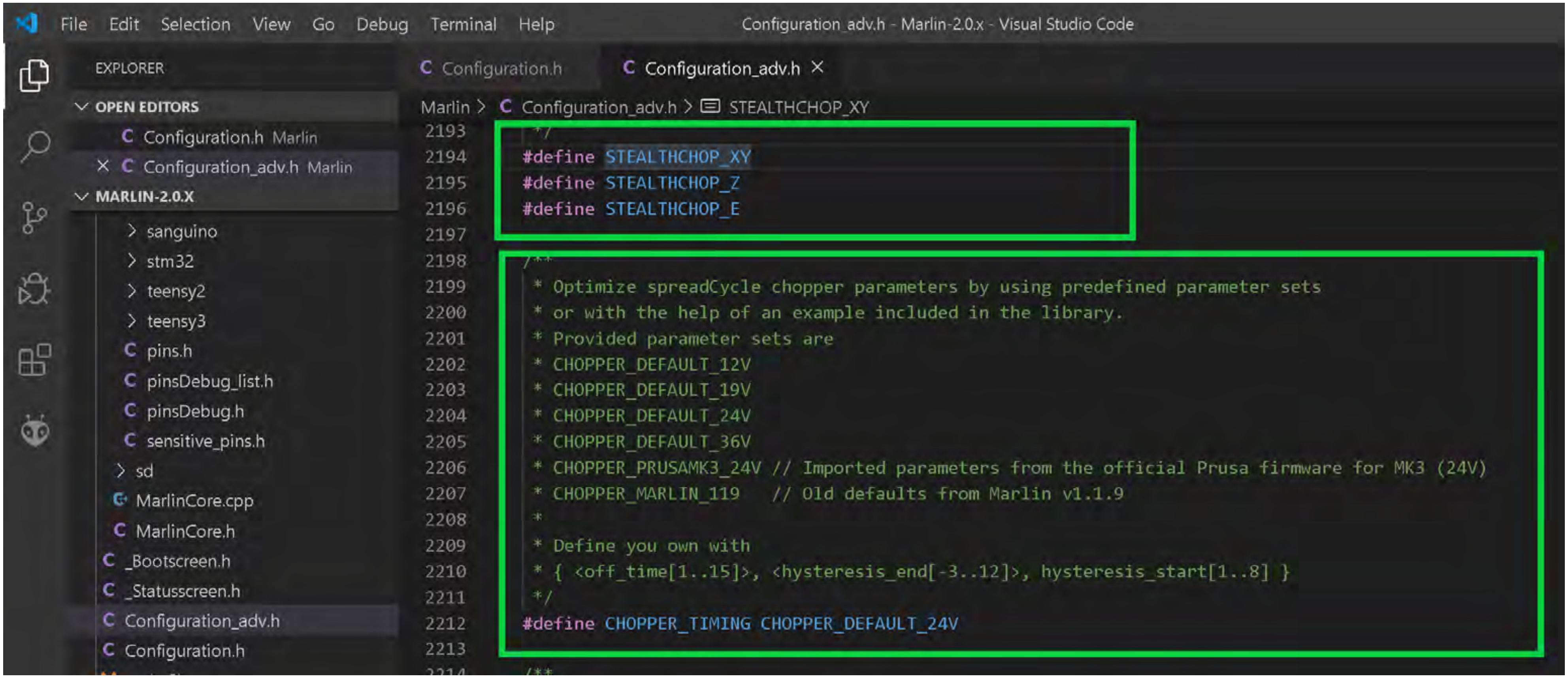
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

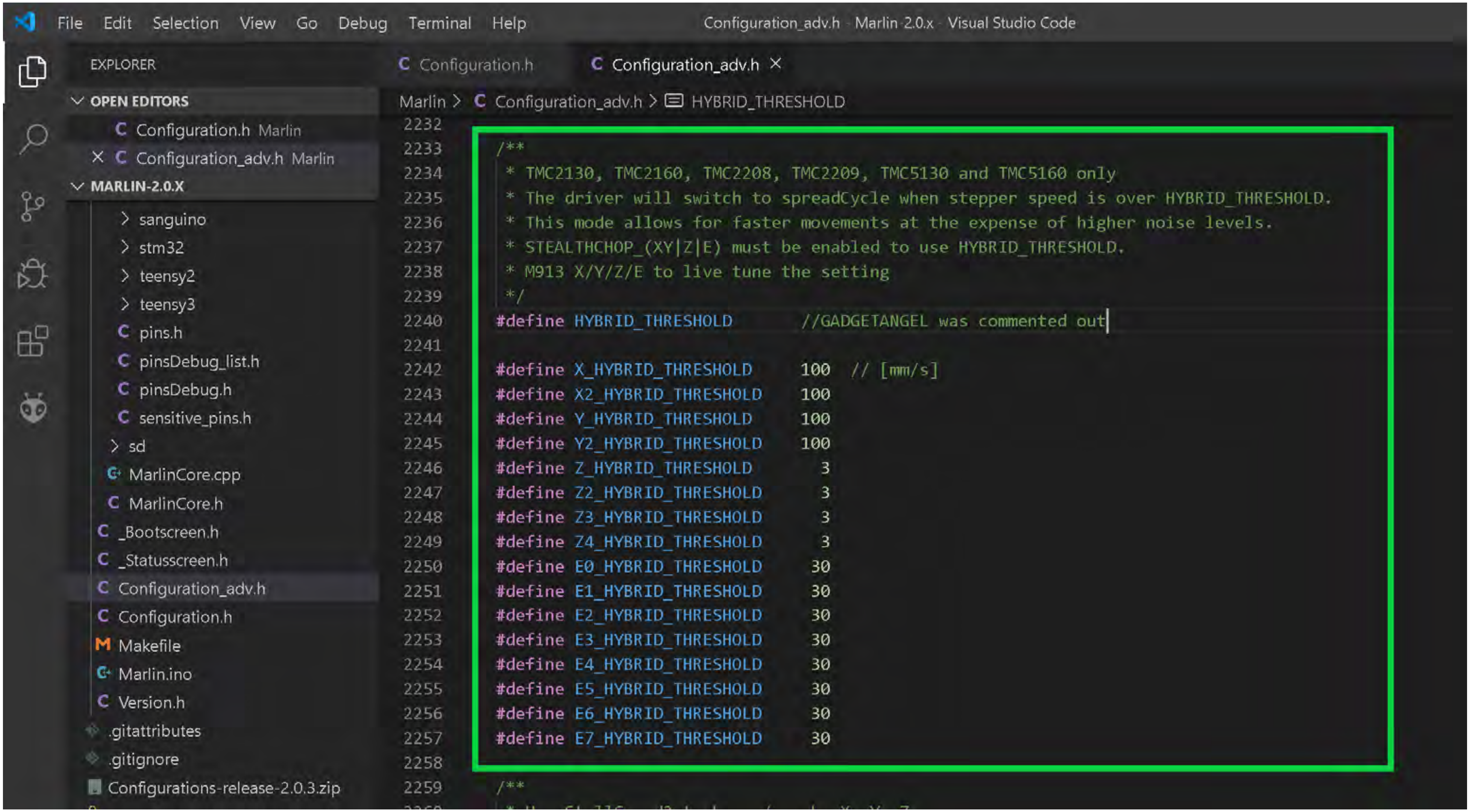
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

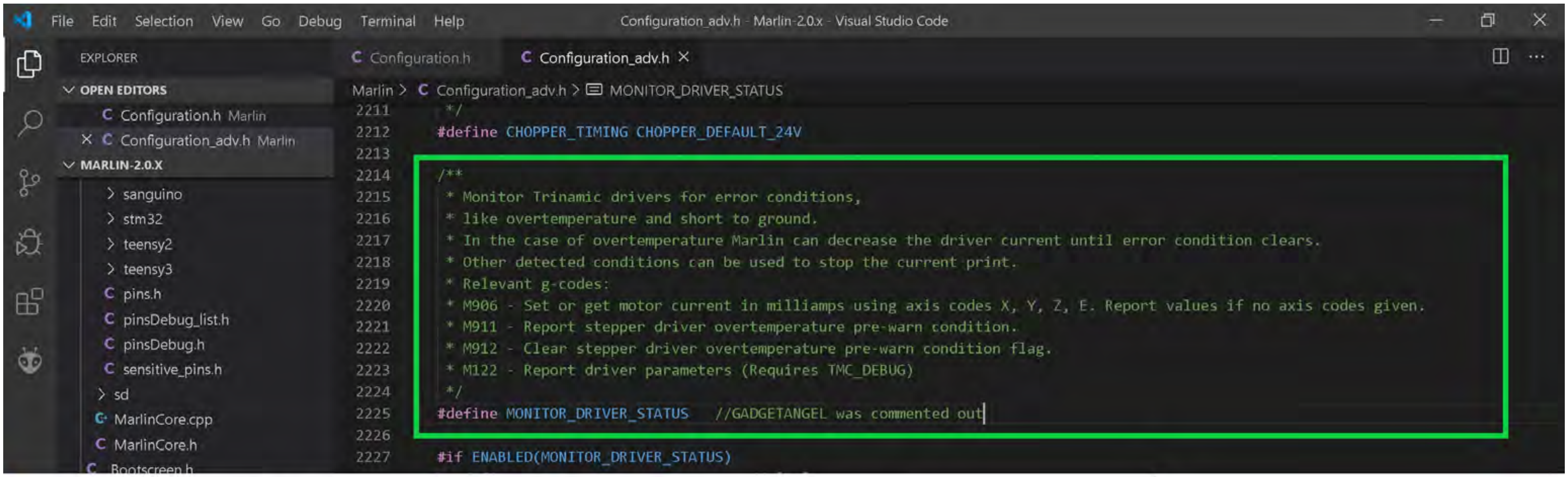
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

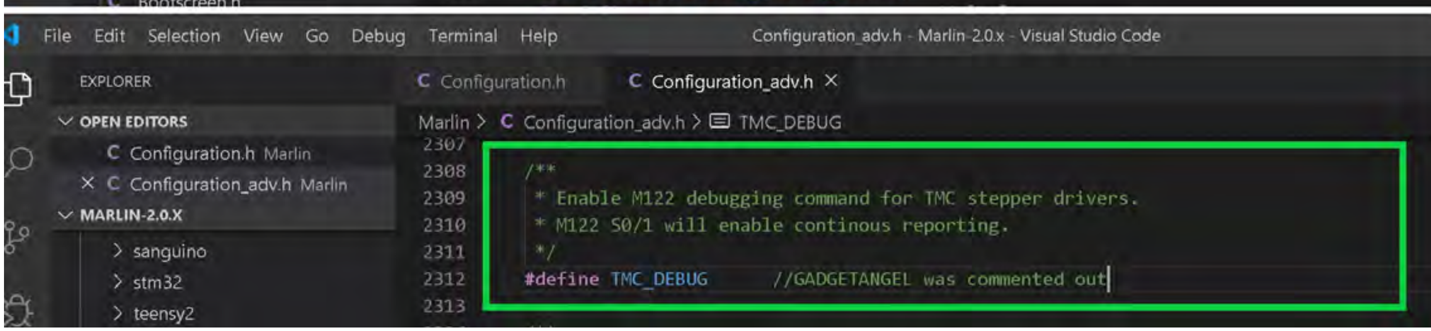


- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

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

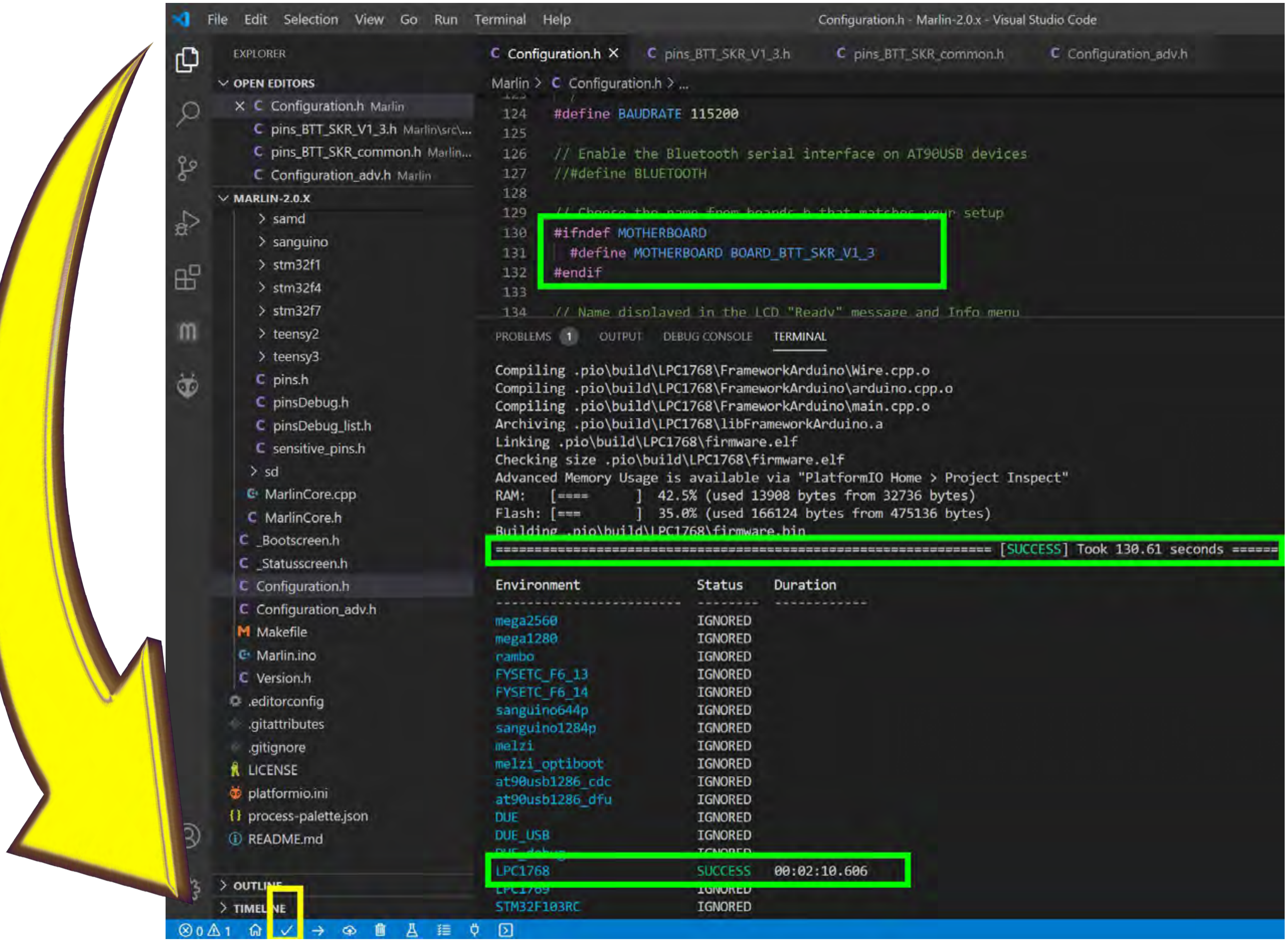




- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

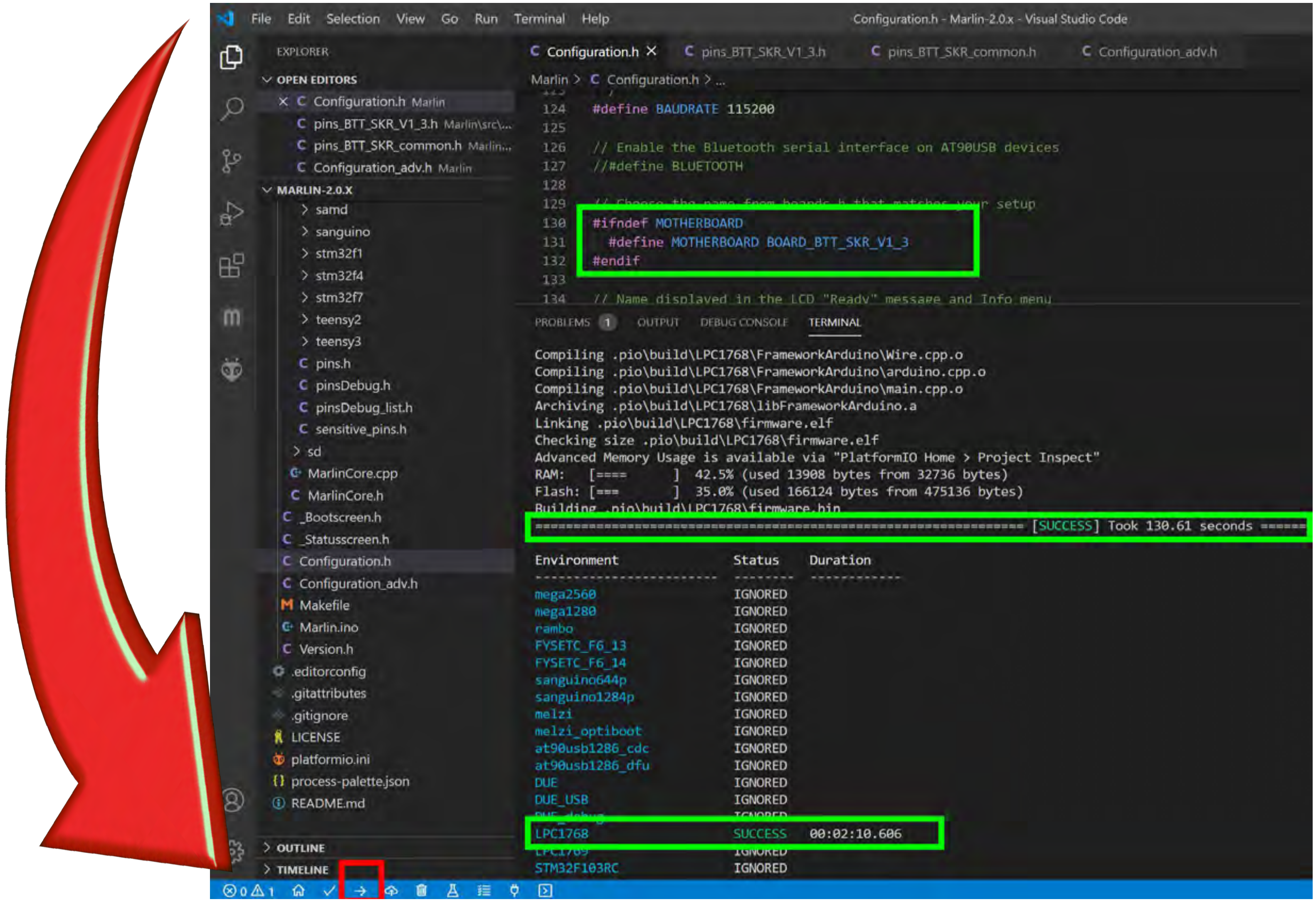
- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

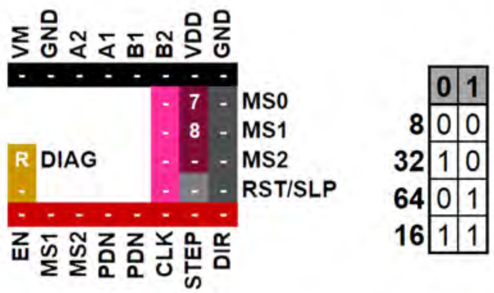
BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

R - Stall Detection

Stand-alone Mode - StealthChop

base of card



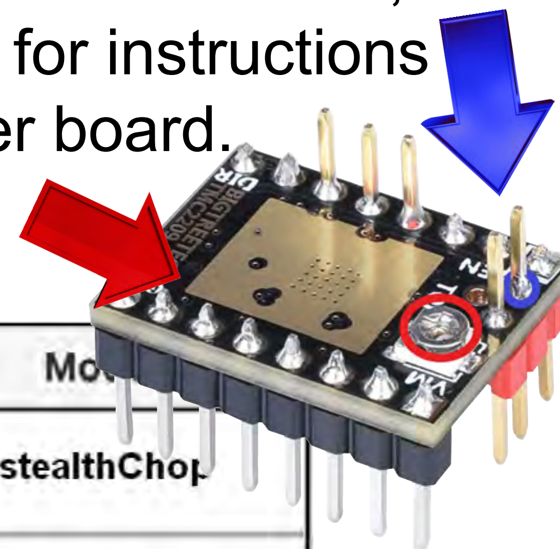
fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

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	MS0	MS1	Steps	Interpolation	Mo
BIQU® TMC2209 Stand Alone Mode. Maximum 64 Subdivision. 28V DC 2.8A (peak)	Low	Low	1 / 8	1 / 256	stealthChop
	High	Low	1 / 32	1 / 256	stealthChop
	Low	High	1 / 64	1 / 256	stealthChop
	High	High	1 / 16	1 / 256	stealthChop

Driving Current Calculation Formula	$I_{MAX} = V_{ref}$	$V_{ref} = I_{MAX}$
R_S (Typical Sense Resistor) = 0.11Ω	See Appendix B #5. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	See Appendix B #5. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

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

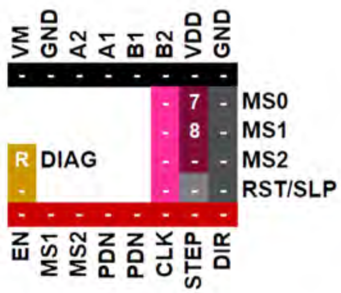
BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

R - Stall Detection

Stand-alone Mode - StealthChop

base of card



	0	1
8	0	0
32	1	0
64	0	1
16	1	1

fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
x.xA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)=X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low ⇒ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High ⇒ Jumper Set

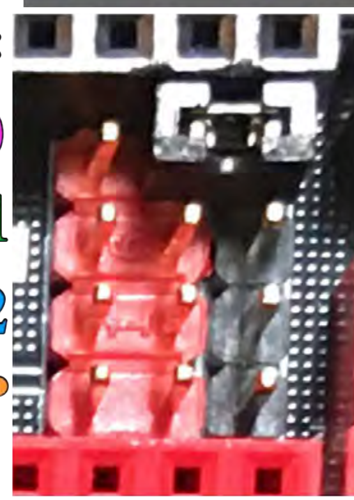
Rows:

MS0

MS1

MS2

RST/SLP



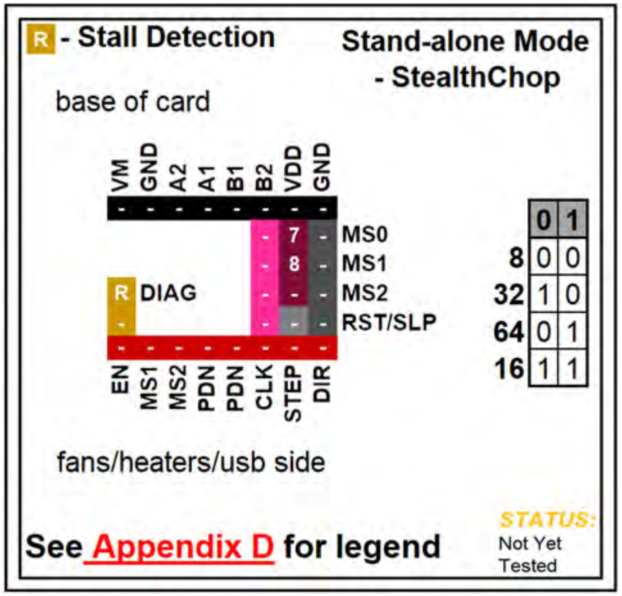
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU TMC2209 V1.2

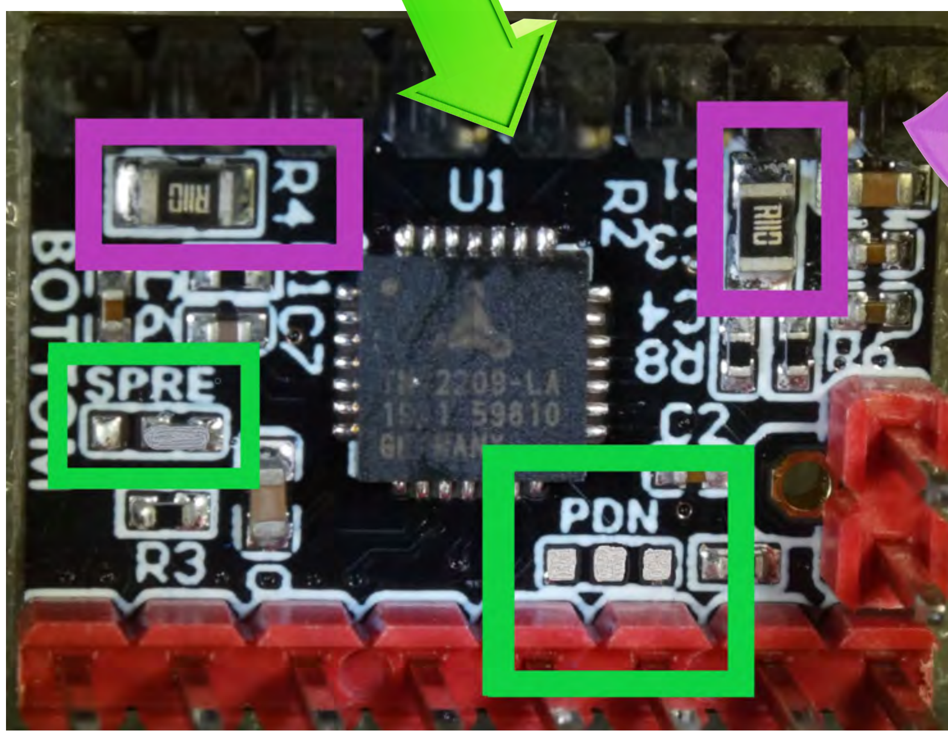
Stand-alone Mode for StealthChop



NOTE: The SPRE jumper is located on the bottom of the driver board. In stand-alone mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board. The **PURPLE boxes** below show the location of the current sense resistors (R_s) for the TMC2209

Note: To switch to stand-alone mode, none of the PDN pads should be bridged. The picture below, as seen in the **GREEN** boxes, shows stand-alone mode with StealthChop.

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.



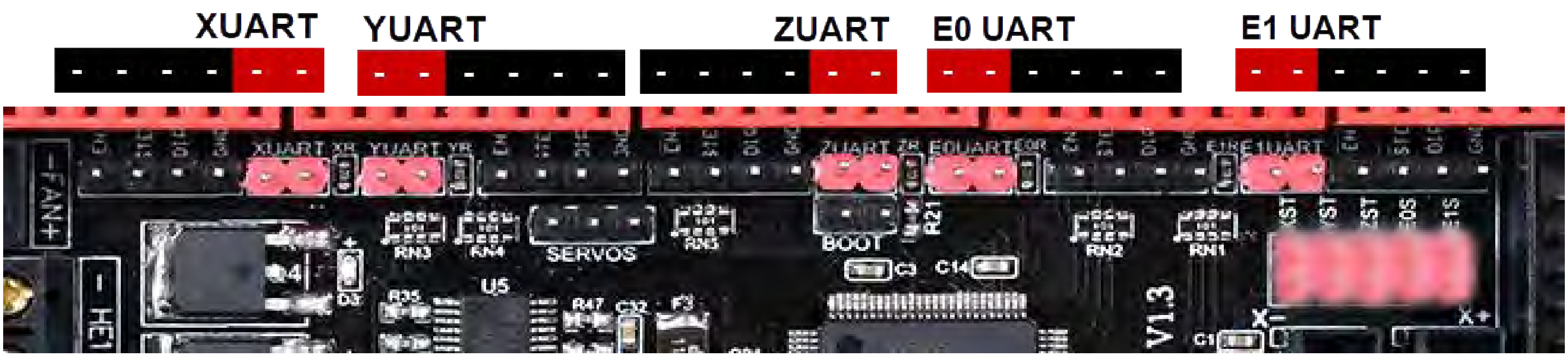
Stand-alone StealthChop Mode

BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode with StealthChop enabled.

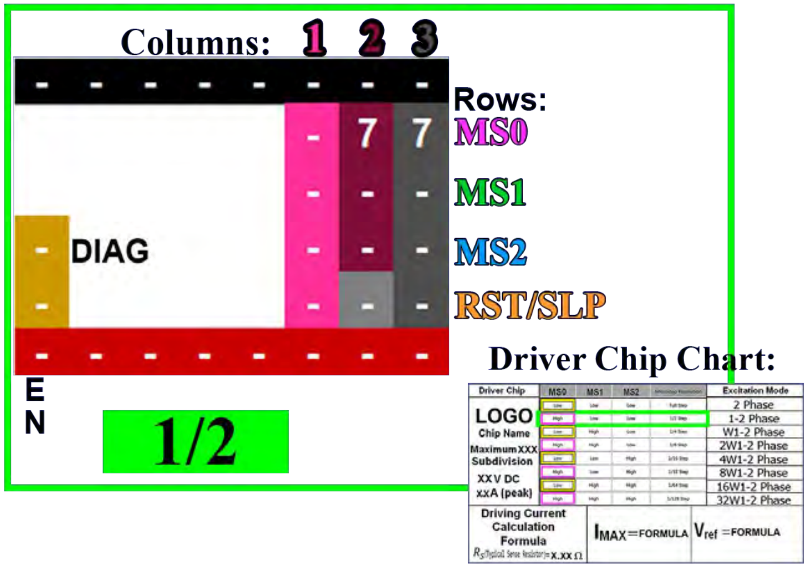
Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode with StealthChop enabled.



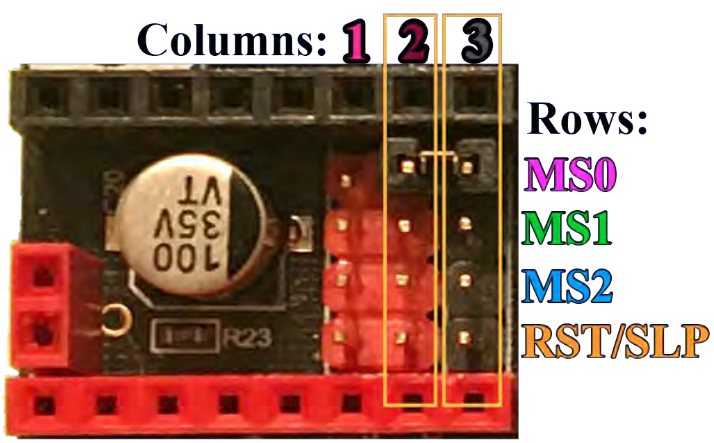
Stand-alone StealthChop Mode

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: **High**

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Chip Chart: **Meaning:**

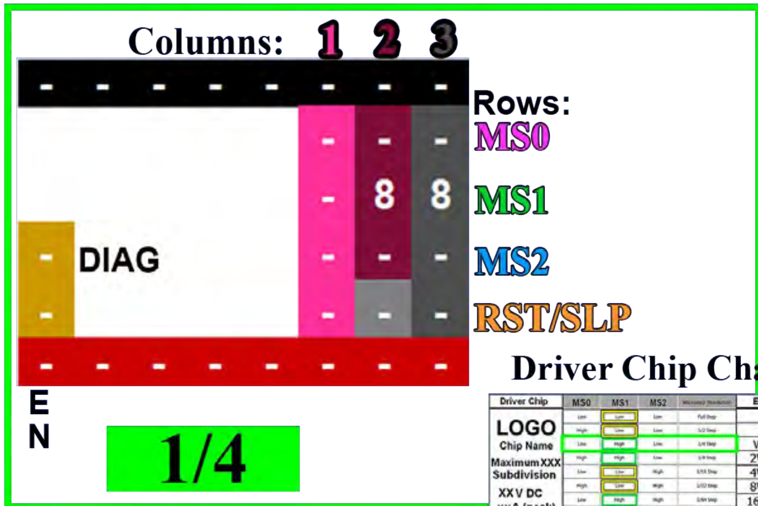
High → **set Jumper between column 2 and column 3 on the MS0 row**

Low → **No Jumper Set**

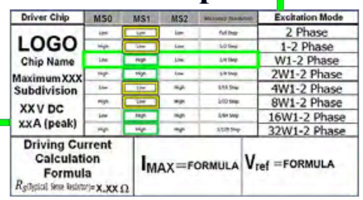
Stand-alone StealthChop Mode

Stand-alone Mode for StealthChop

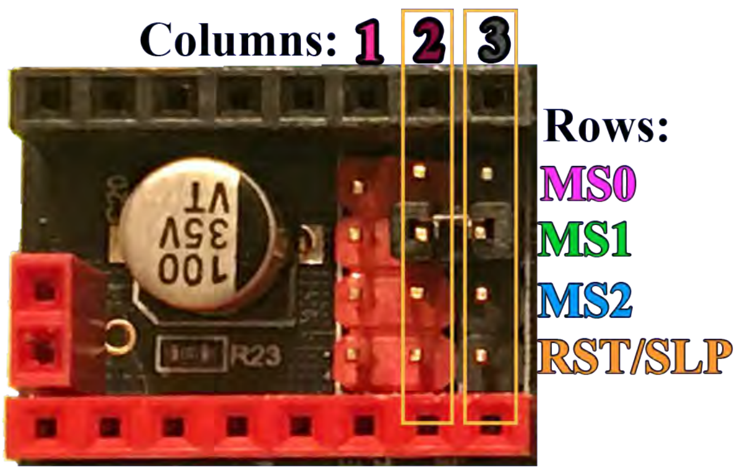
Driver Socket Representation



Driver Chip Chart:



High:



MS1 for Binary State Drivers:

Driver Socket Representation:	Driver Chip Chart:
- -	Low
8 8	High
- -	Low

Driver Chip Chart: / **Meaning:**

High → set <u>Jumper</u> between column 2 and column 3 on the MS1 row

Low → No <u>Jumper</u> Set

Stand-alone StealthChop Mode

BIQU TMC2209 V1.2

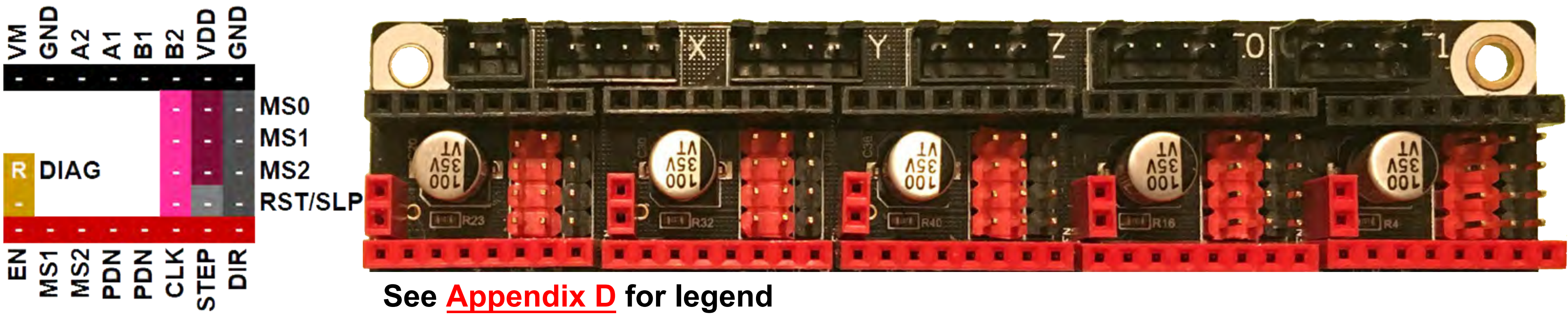
Stand-alone Mode for StealthChop

Stand-alone Mode

1 / 8

Interpolation:
1 / 256

StealthChop



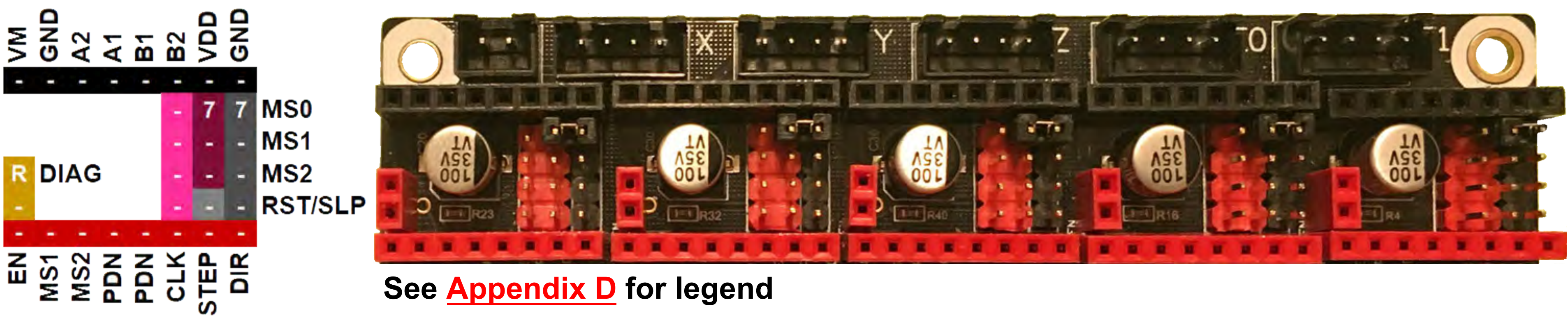
See Appendix D for legend

Stand-alone Mode

1 / 32

Interpolation:
1 / 256

StealthChop



See Appendix D for legend

Stand-alone StealthChop Mode

BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

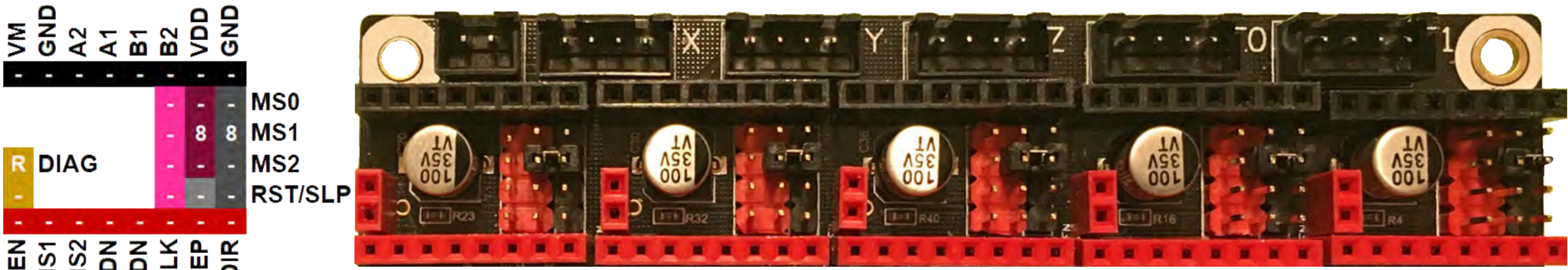
Stand-alone Mode

1 / 64

Interpolation:

1 / 256

StealthChop



See Appendix D for legend

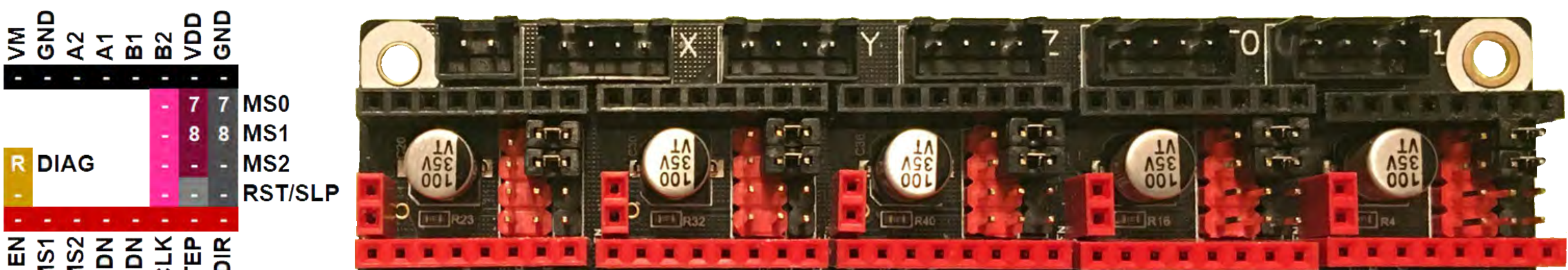
Stand-alone Mode

1 / 16

Interpolation:

1 / 256

StealthChop

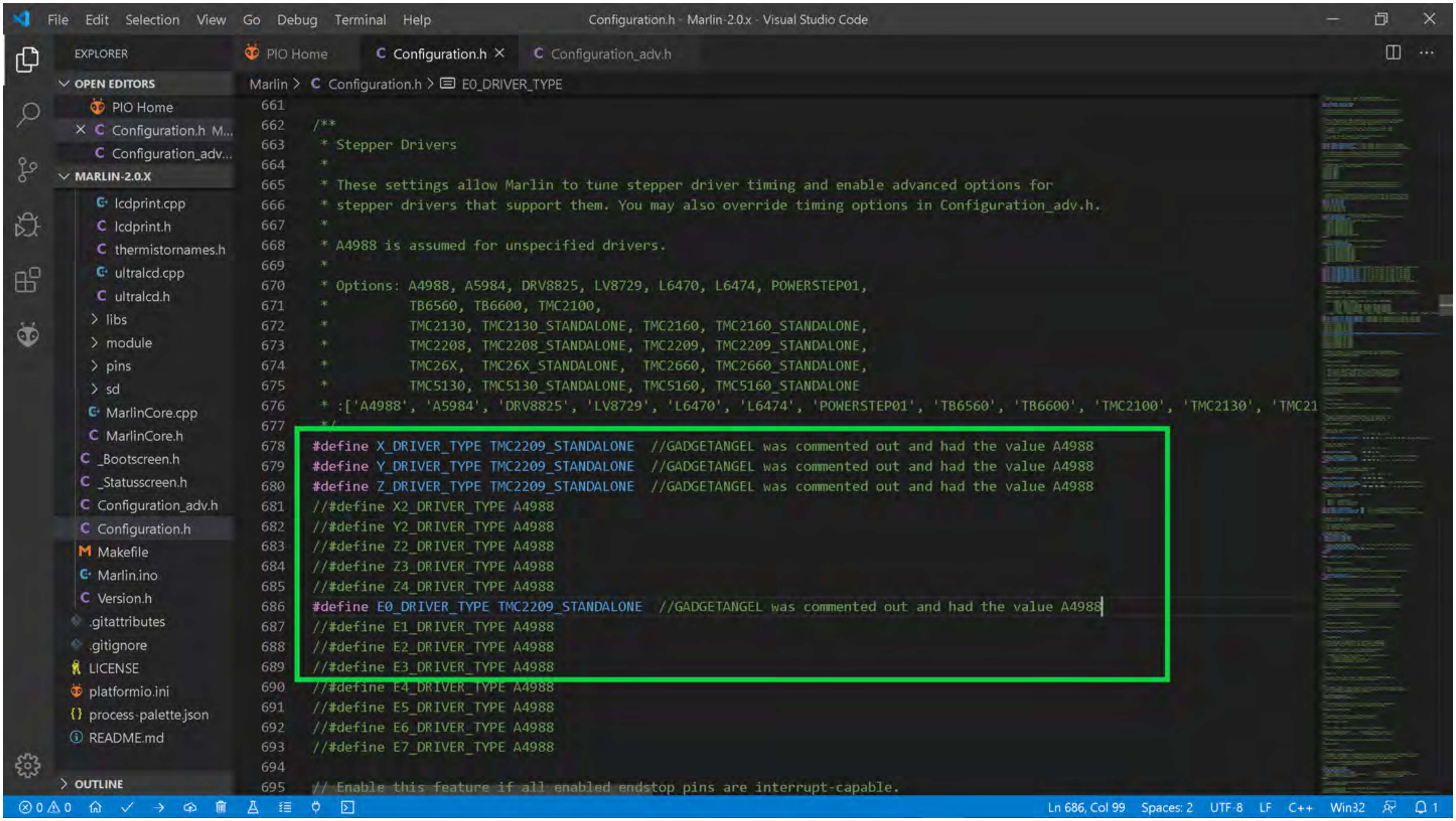


See Appendix D for legend

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

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

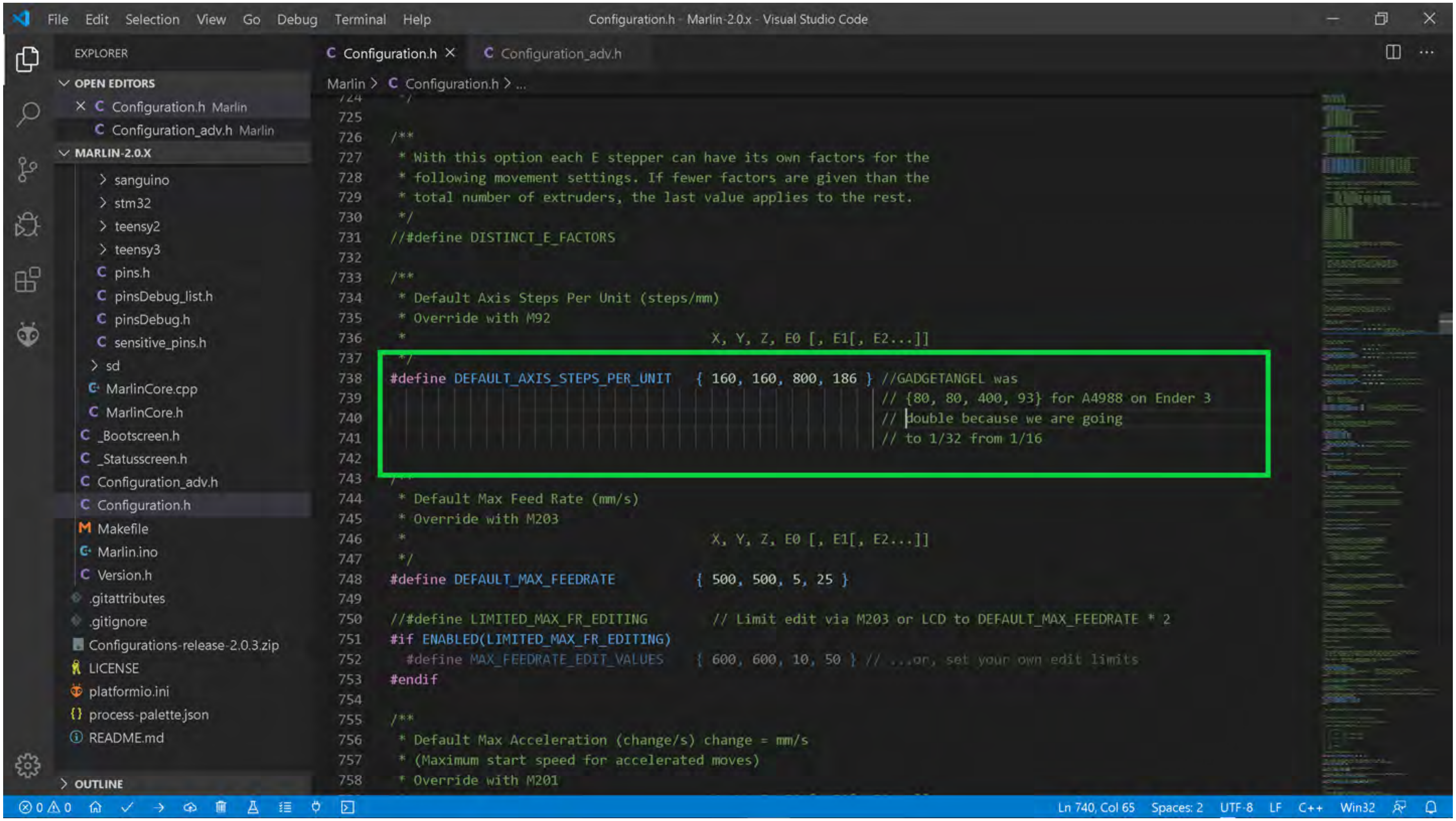
- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

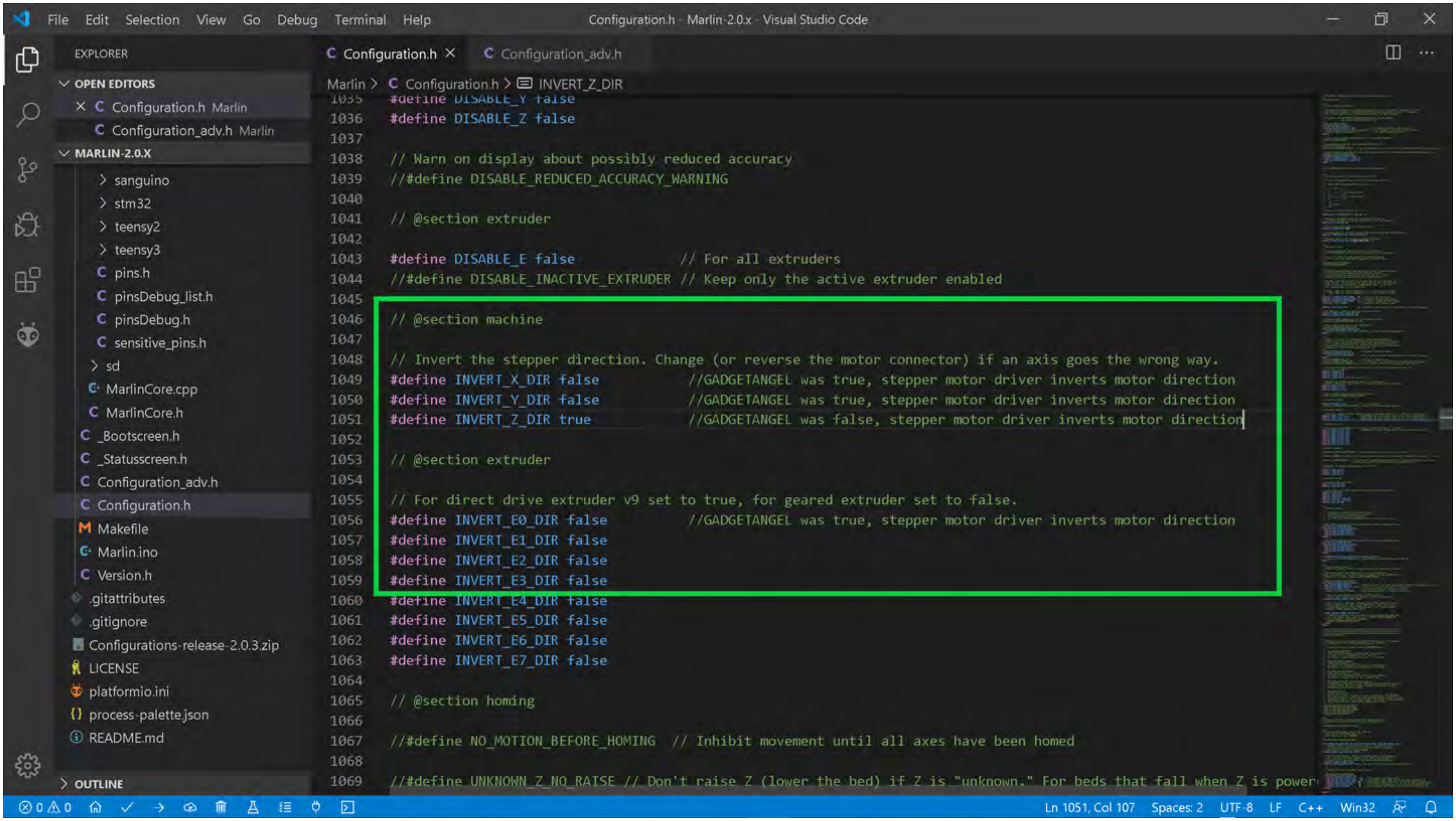
- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



- Go to the next page.

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

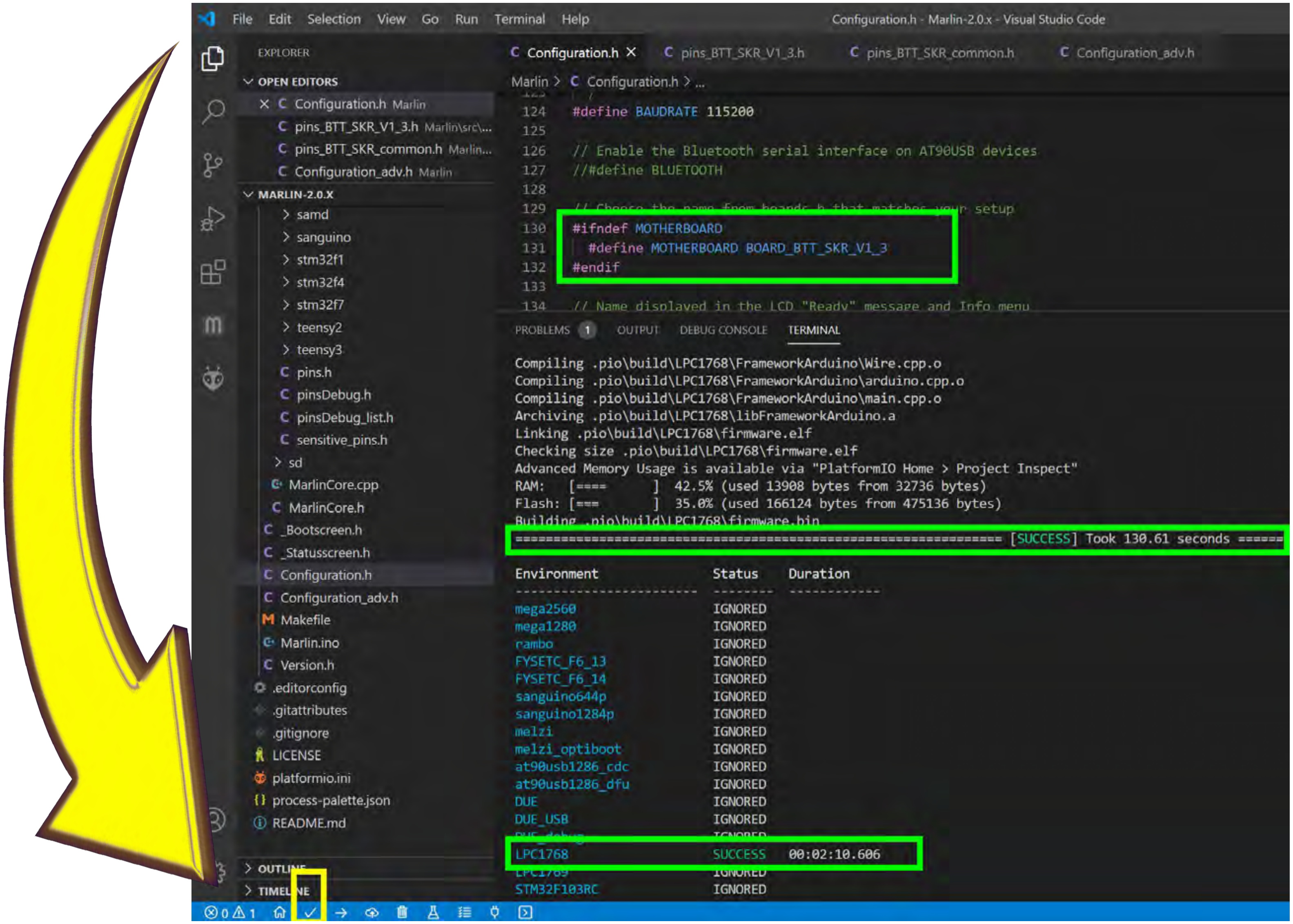
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209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



- Go to the next page.

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

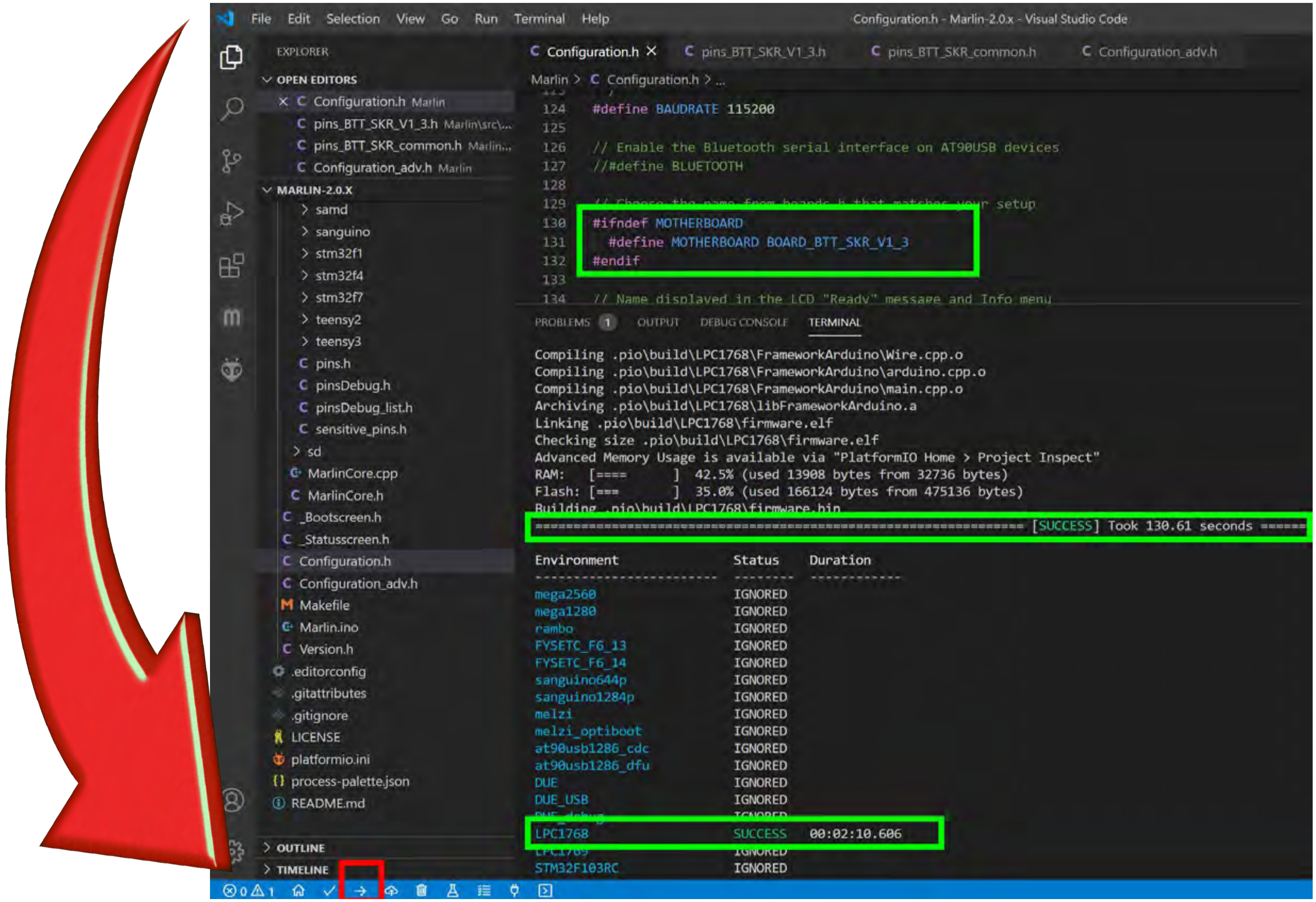
- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand-alone mode for stealthChop. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

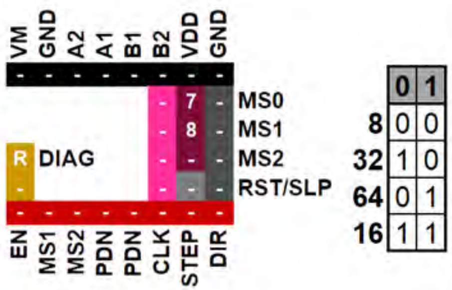
BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

R - Stall Detection

Stand-alone Mode - SpreadCycle

base of card



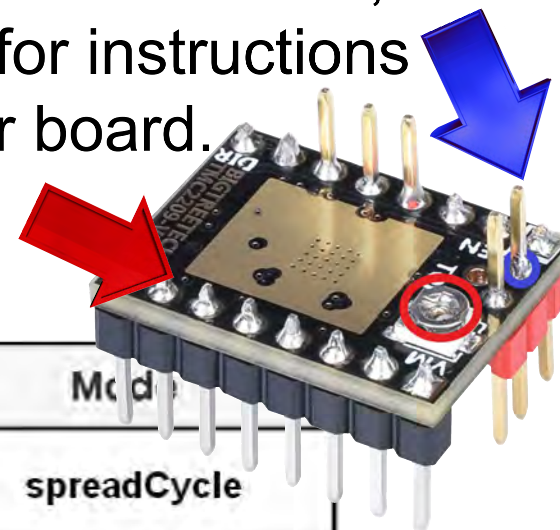
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

Driving Current Calculation Formula	$I_{MAX}=V_{ref}$	$V_{ref}=I_{MAX}$
R_S (Typical Sense Resistor) = 0.11Ω	See Appendix B #5. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$	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.**

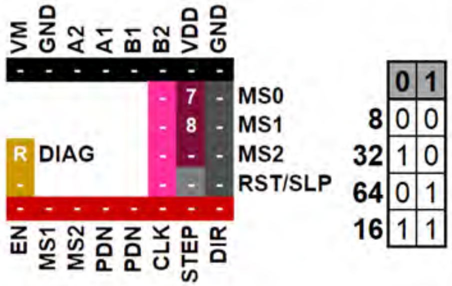
BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

R - Stall Detection

Stand-alone Mode - SpreadCycle

base of card



fans/heaters/usb side

See Appendix D for legend

STATUS: Not Yet Tested

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX	Low	High	Low	1/4 Step	W1-2 Phase
Subdivision	High	High	Low	1/8 Step	2W1-2 Phase
XX V DC	Low	Low	High	1/16 Step	4W1-2 Phase
X.XA (peak)	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)=X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Low → No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

High → Jumper Set

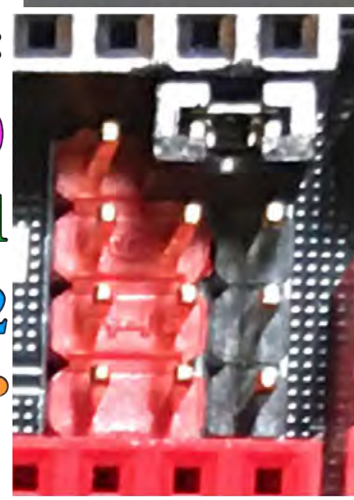
Rows:

MS0

MS1

MS2

RST/SLP



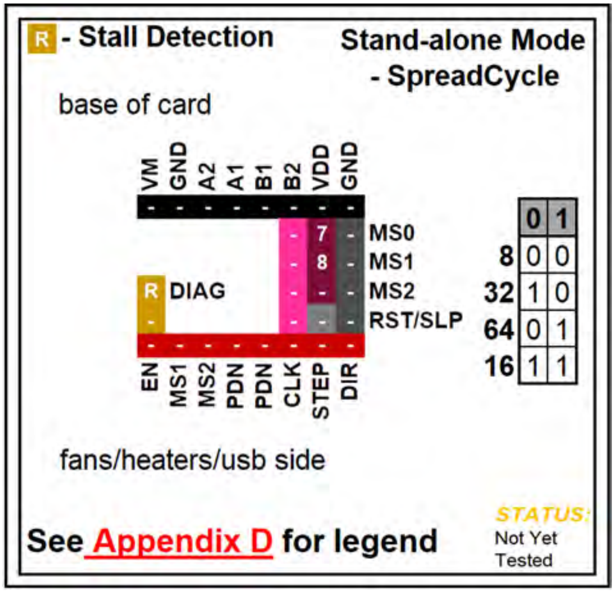
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

BIQU TMC2209 V1.2

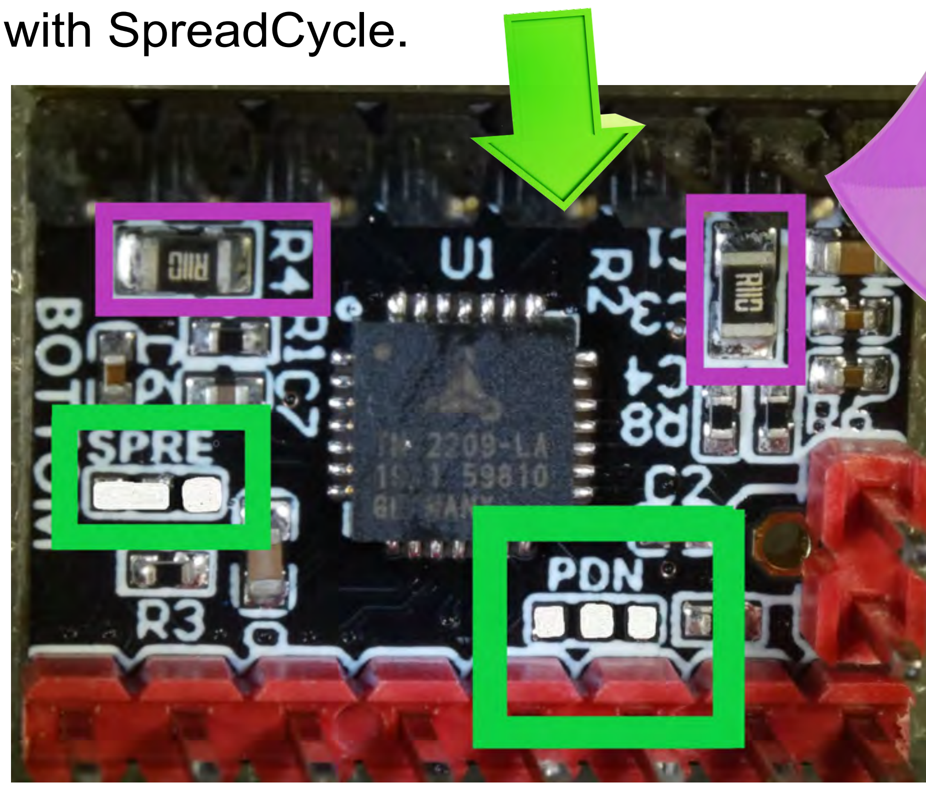
Stand-alone Mode for SpreadCycle



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.

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.

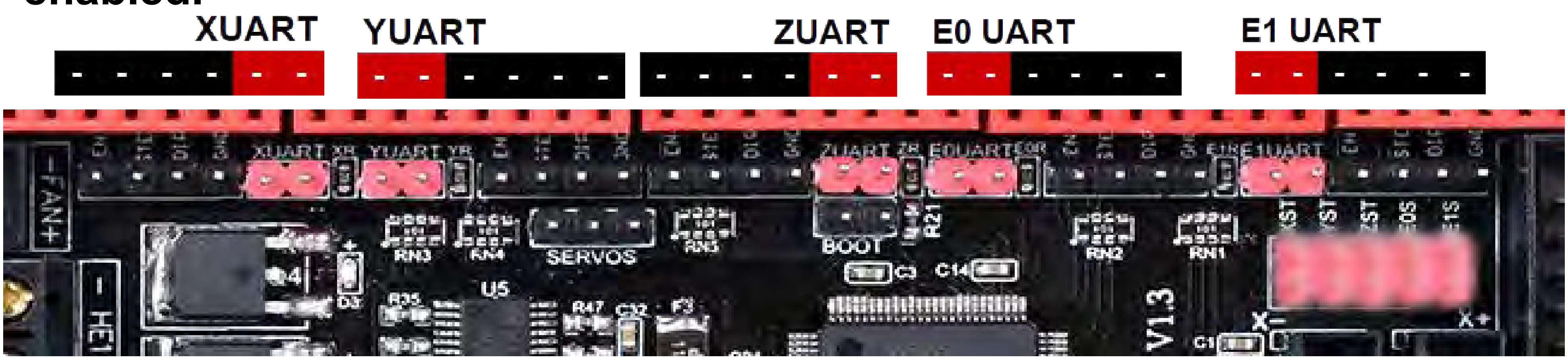


Stand-alone SpreadCycle Mode

Stand-alone Mode for SpreadCycle

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode with spreadCycle enabled.

Note: Ensure the UART jumpers are all empty on the SKR V1.3 board to enable proper operation for stand-alone mode with spreadCycle enabled.



Stand-alone SpreadCycle Mode

Stand-alone Mode for SpreadCycle

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

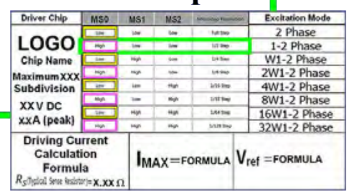
Driver Socket Representation

Columns: 1 2 3

Rows:
MS0
MS1
MS2
RST/SLP

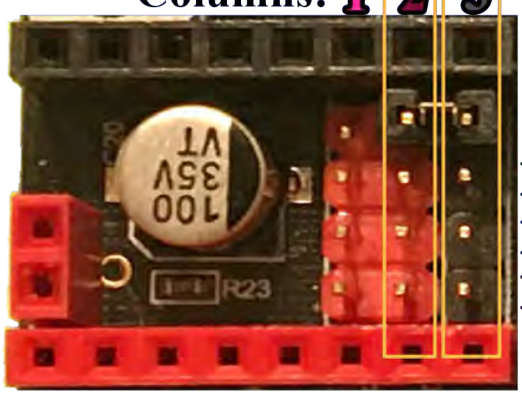
1/2

Driver Chip Chart:



High:

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

Driver Chip Chart:

High → set Jumper between column 2 and column 3 on the MS0 row

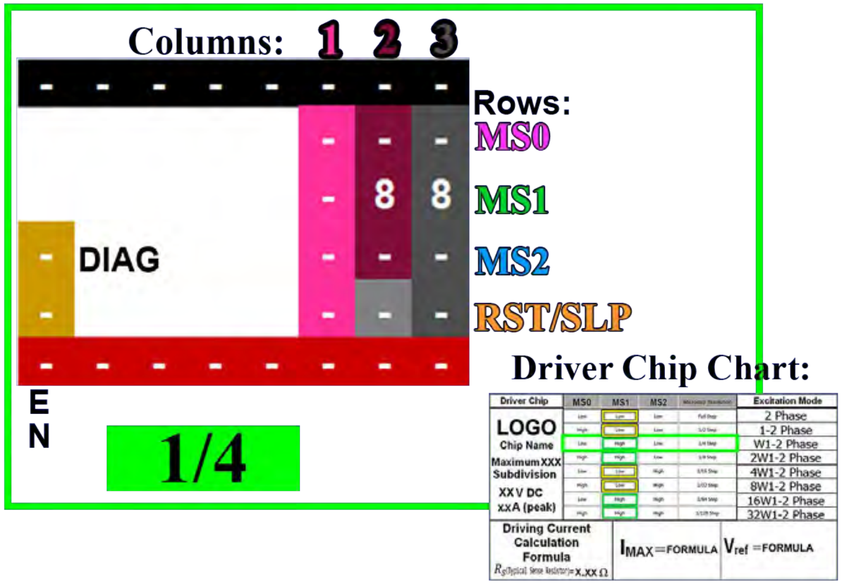
Low → No Jumper Set

Stand-alone SpreadCycle Mode

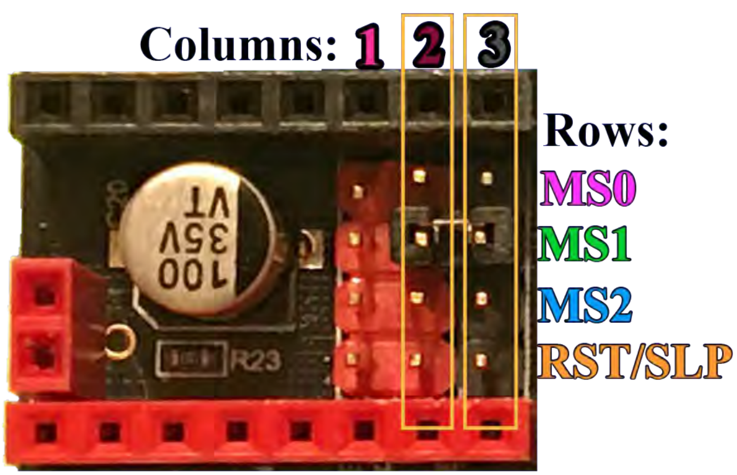
BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

Driver Socket Representation



High:



MS1 for Binary State Drivers:

Driver Socket Representation: - - → **Driver Chip Chart:** Low

Driver Socket Representation: 8 8 → **Driver Chip Chart:** High

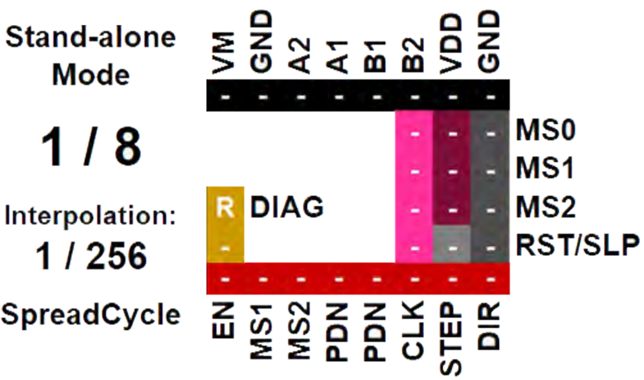
Driver Socket Representation: - - → **Driver Chip Chart:** Low

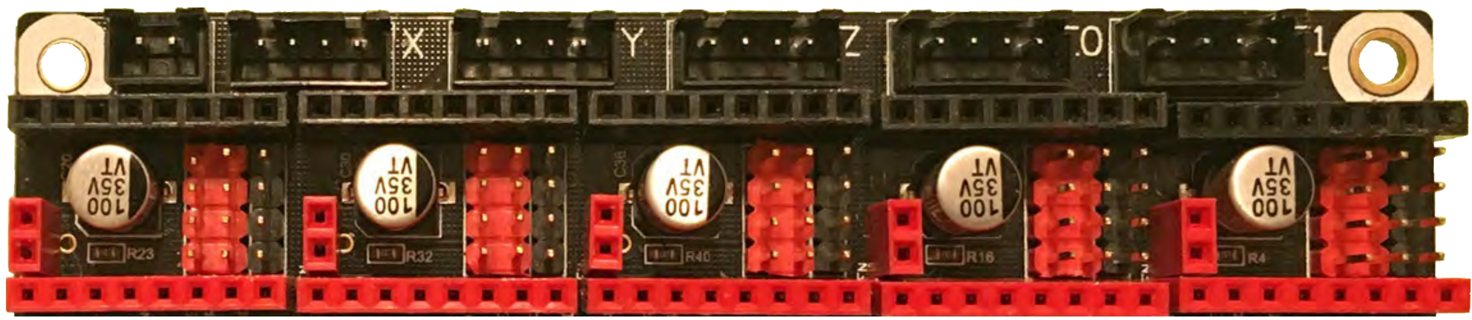
Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS1 row
Low →	No Jumper Set

Stand-alone SpreadCycle Mode

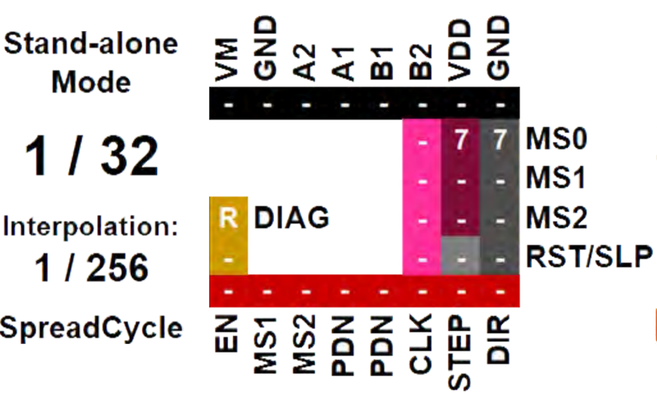
BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle





See Appendix D for legend

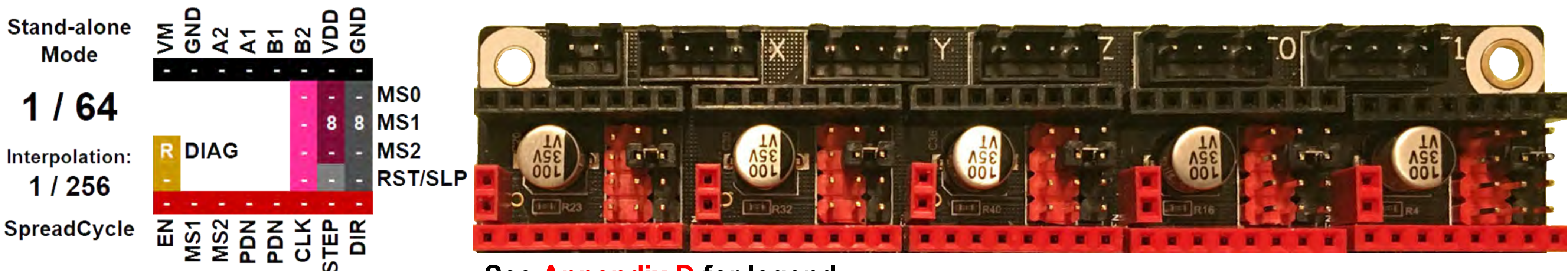


See Appendix D for legend

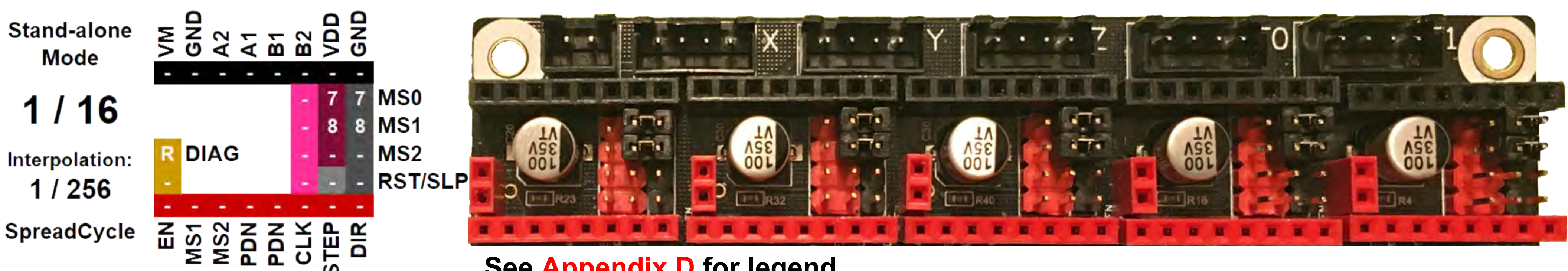
BIQU TMC2209 V1.2

Stand-alone SpreadCycle Mode

Stand-alone Mode for SpreadCycle



See Appendix D for legend

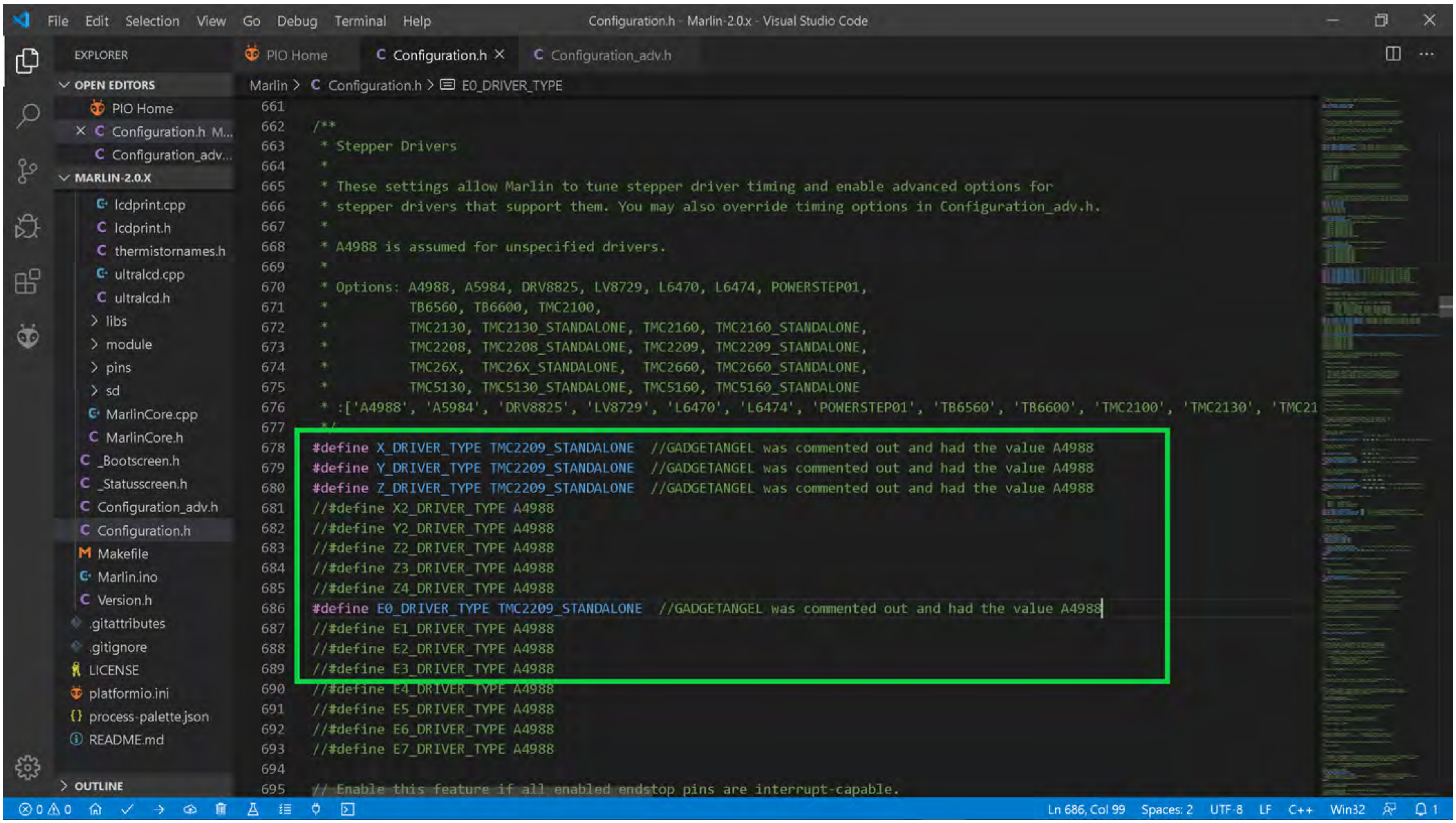


See Appendix D for legend

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.

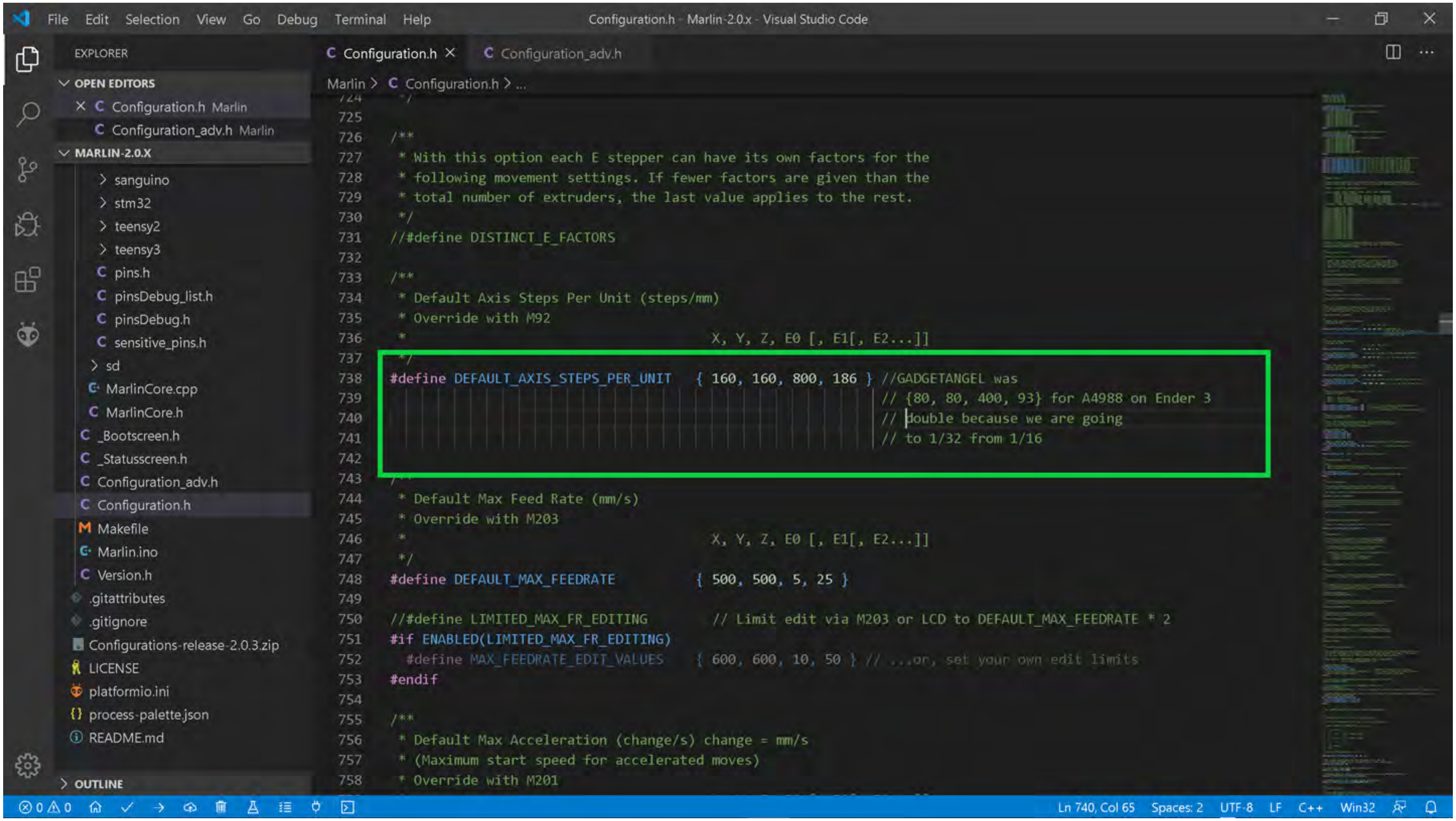
- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand-alone mode. When two "//" appear at the beginning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



- Go to the next page.

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

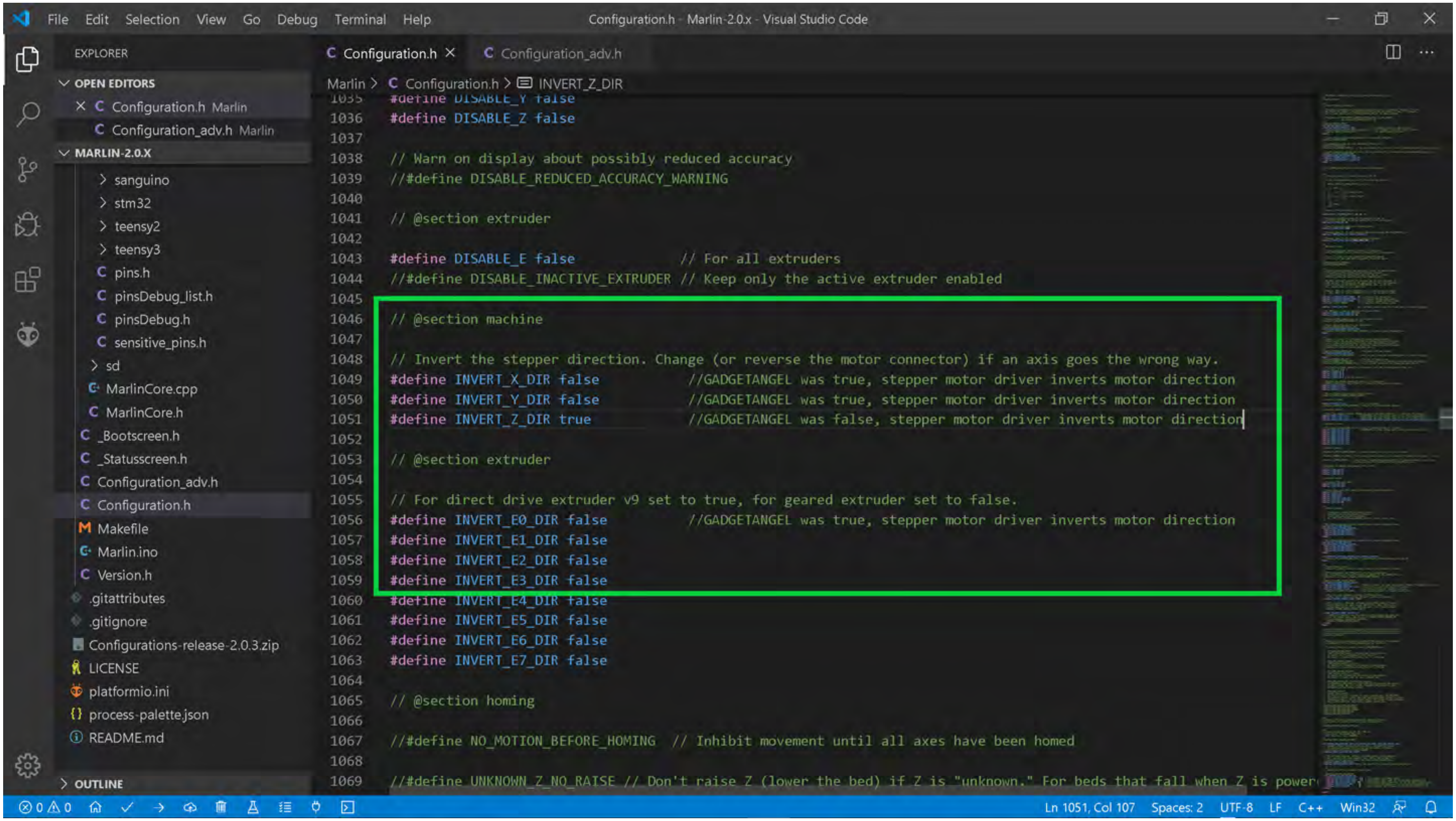
- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust 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.



- Go to the next page.

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

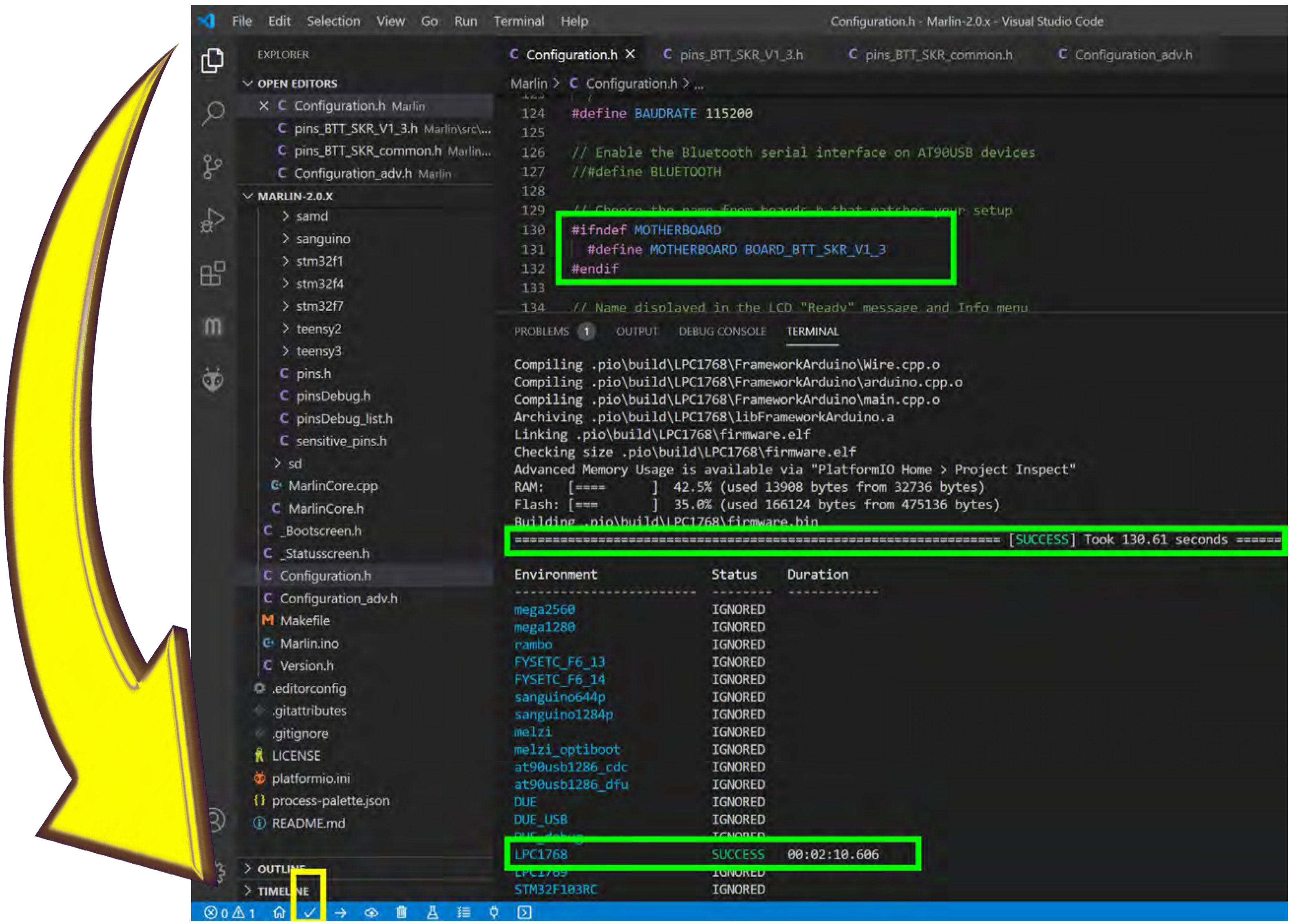
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 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



- Go to the next page.

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

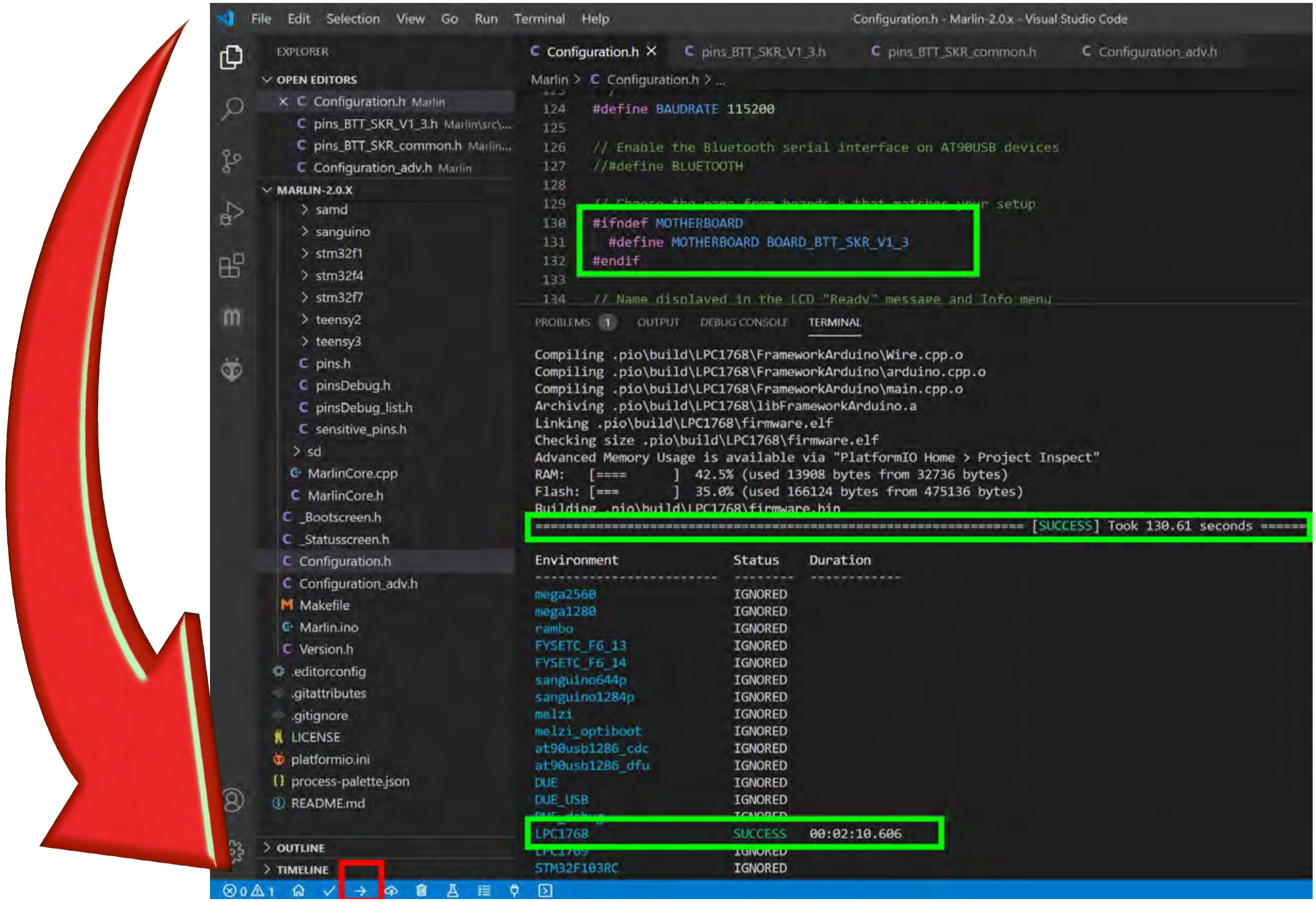
- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand-alone mode for spreadCycle. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

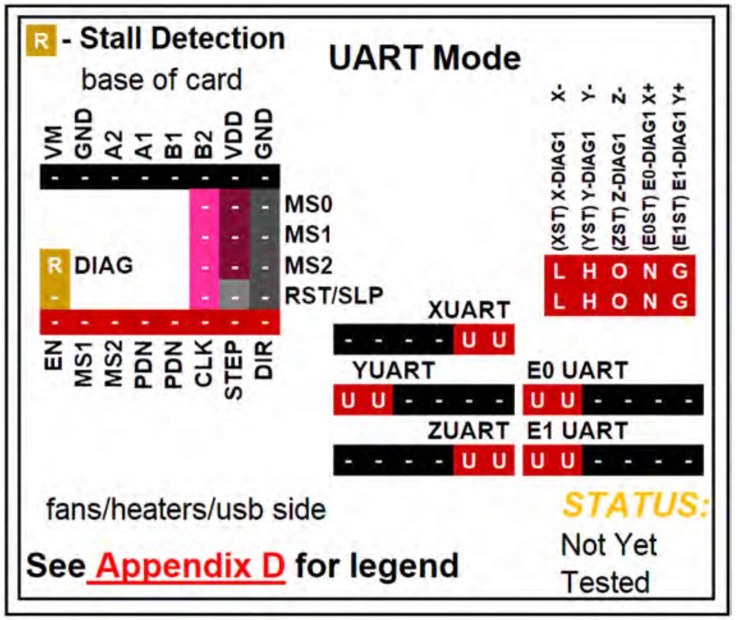
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC2209 V1.2

UART Mode



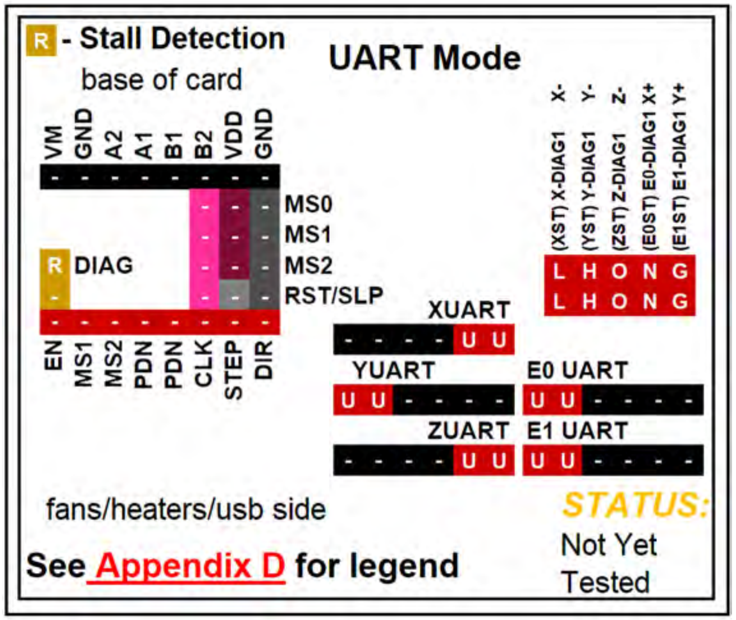
Note: You can use 50% to 90% of the calculated I_{RMS} (I_{MAX}/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

Driver Chip	Steps are set inside of your Firmware	
BIQU® TMC2209 UART Mode Maximum 256 Subdivision 28V DC 2.8A (peak)		
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: $V_{ref}=V_{ref} * 0.90$

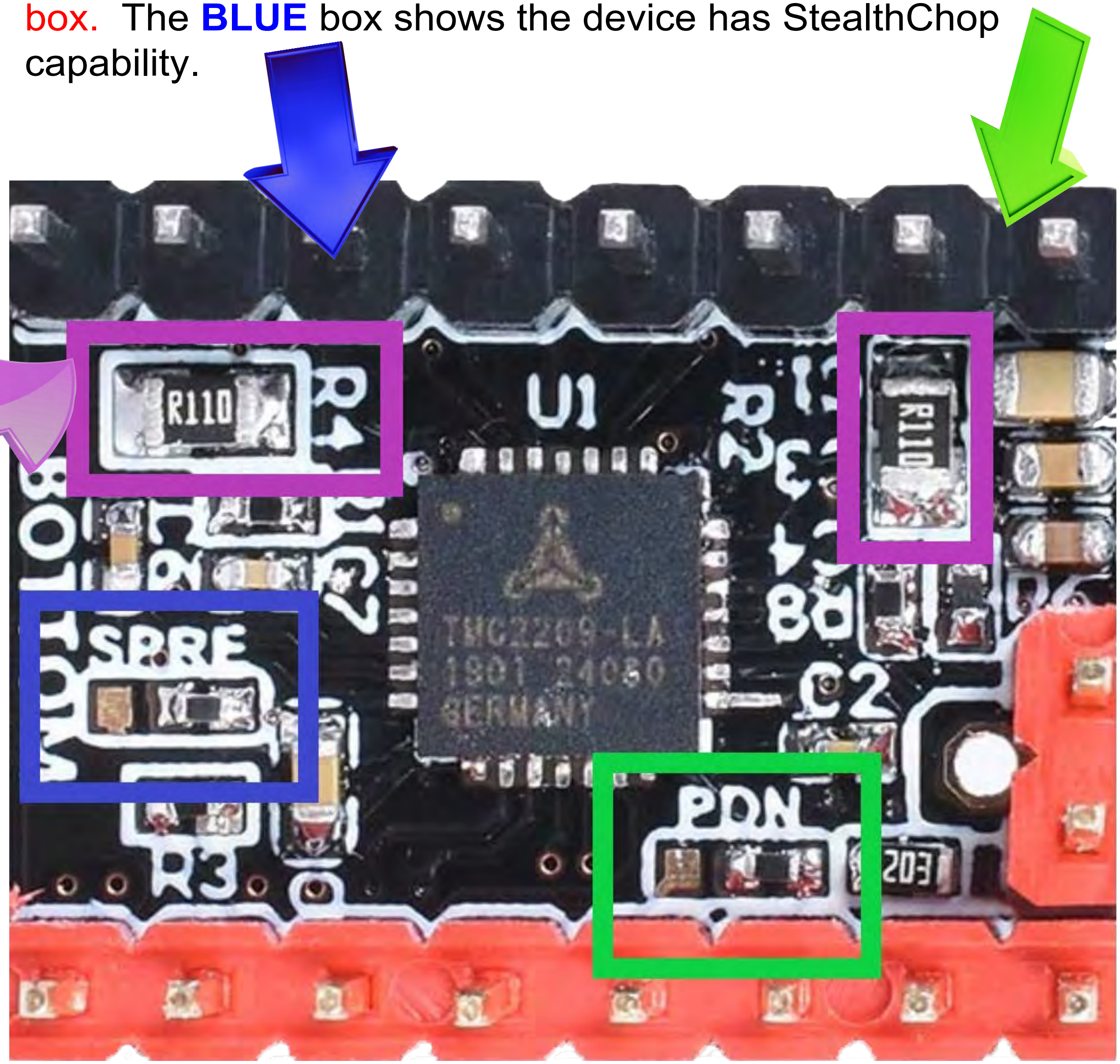
BIQU TMC2209 V1.2

UART Mode



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.

The **PURPLE** boxes show the location of the current sense resistors (R_S).



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

UART 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", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_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

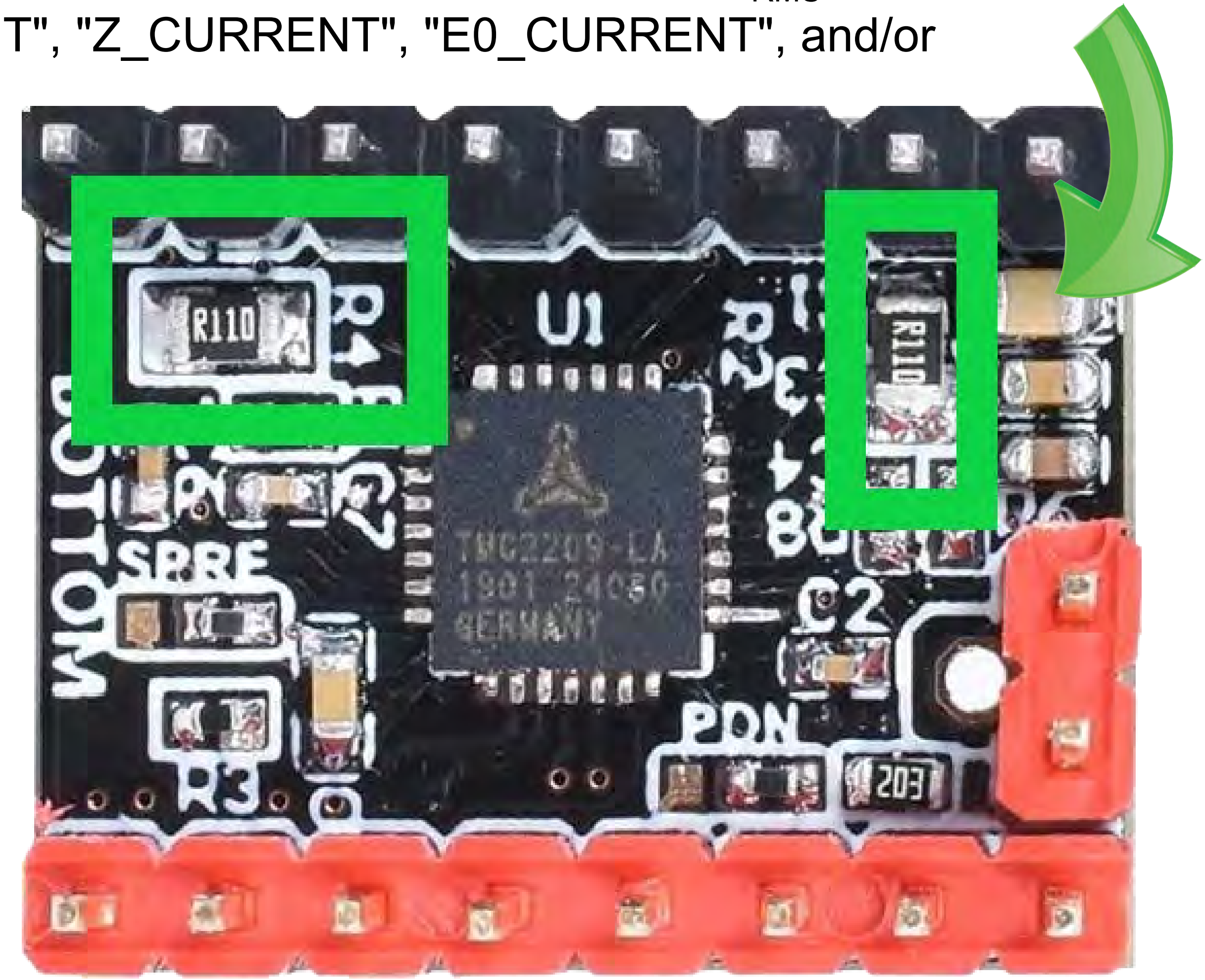
R_s = R100 is 0.1 Ohms

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



BIQU TMC2209 V1.2

UART Mode

UART Mode

R - Stall Detection

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	-	-
L	H	O	-	-

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	N	G
L	H	O	N	G

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
-	-	-	-	-
-	-	-	-	-

Note: If sensor-less homing is not wanted ensure the following JUMPER(s) **are empty**: "L", "H", "O", "N", and "G".

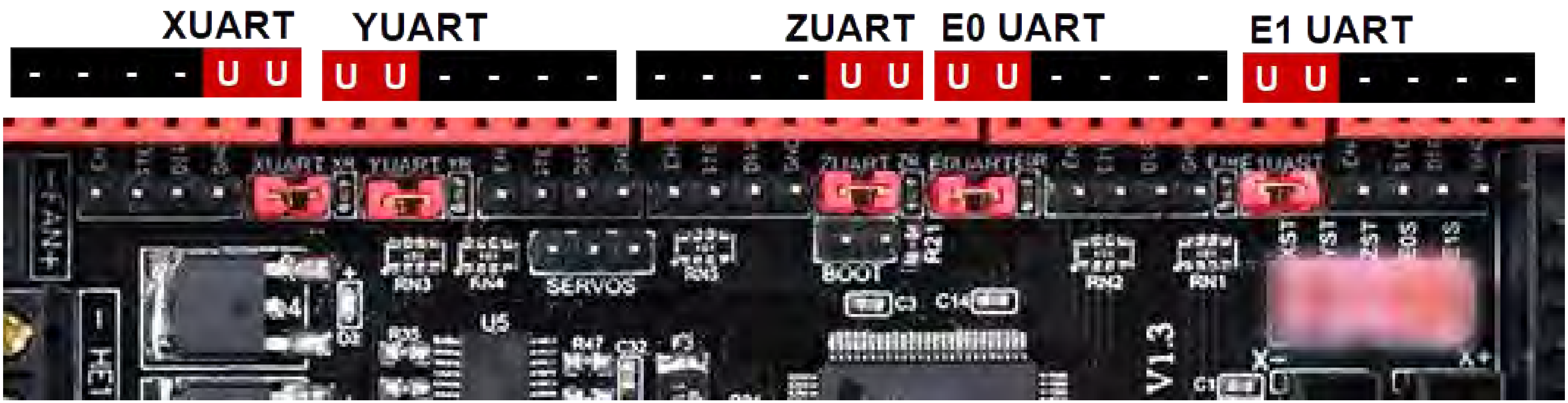
BIQU TMC2209 V1.2

UART Mode

UART Mode

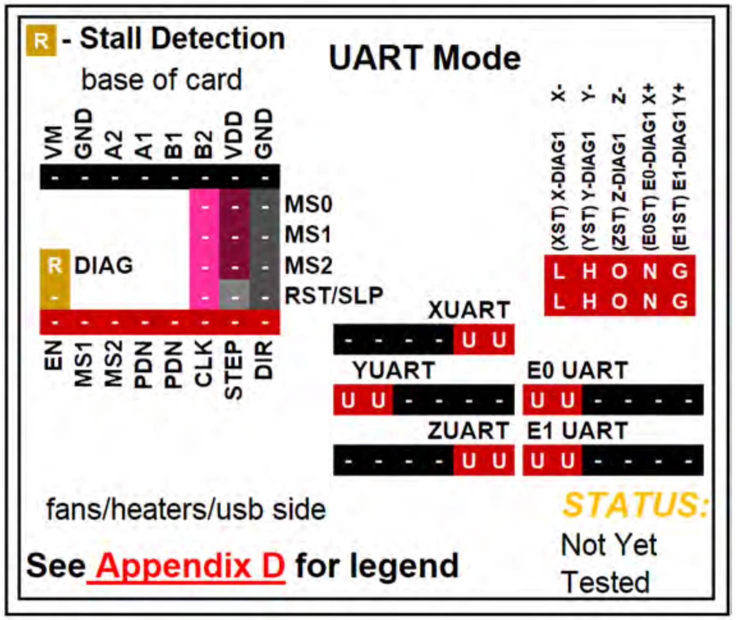
R - Stall Detection

Note: Set the "U" Jumper(s) for UART MODE!

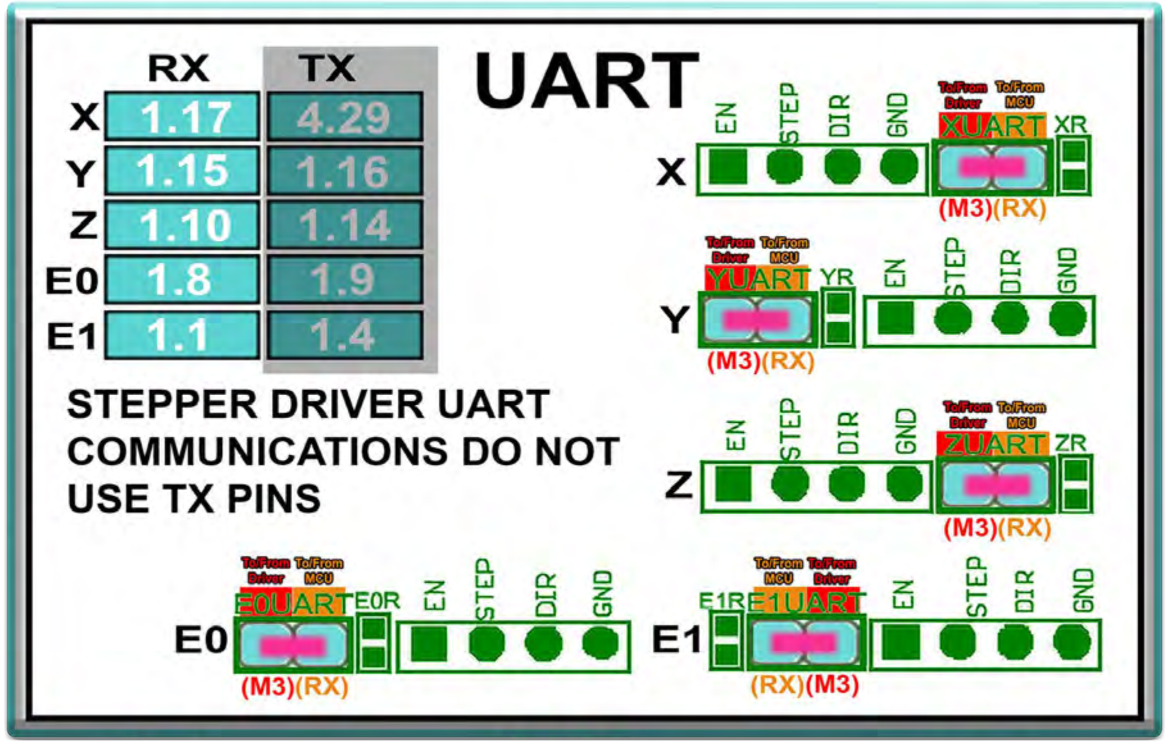


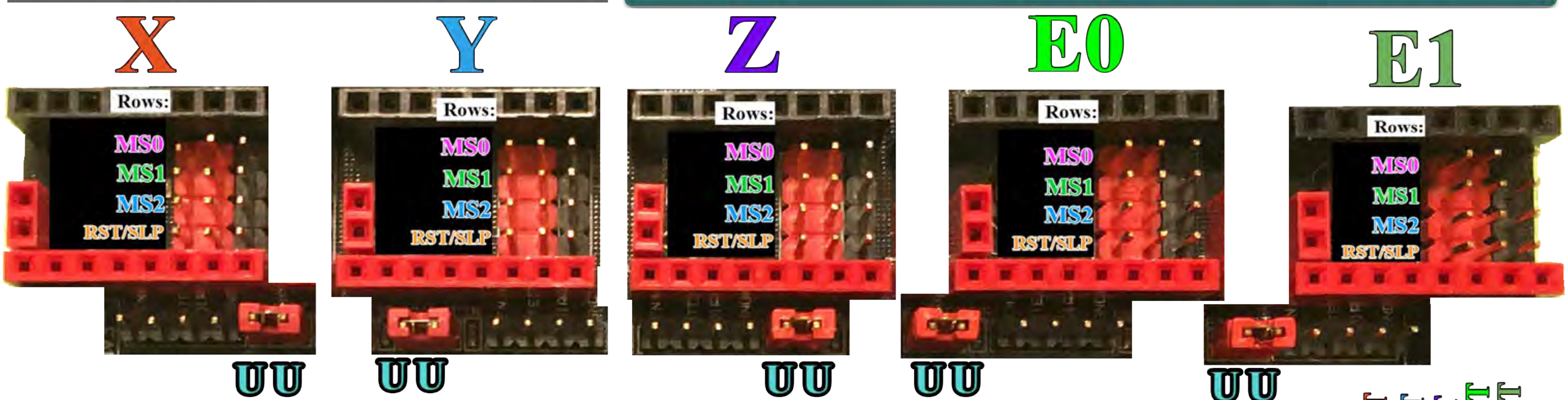
BIQU TMC2209 V1.2

UART Mode

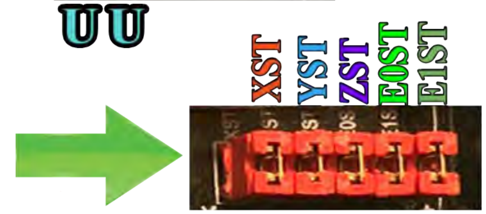


SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode





SET 'ST' JUMPER(s) to enable Sensor-less Homing – for drivers with stallGaurd™ feature

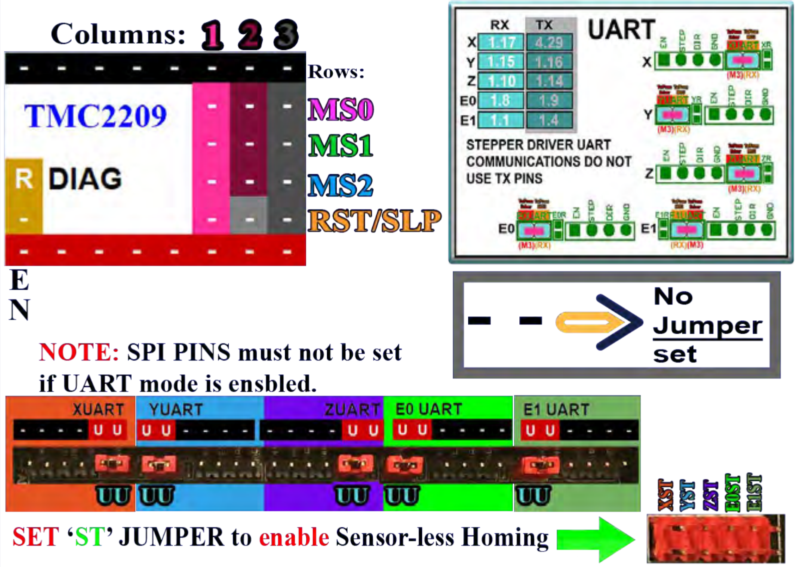


UART Mode

R - Stall Detection

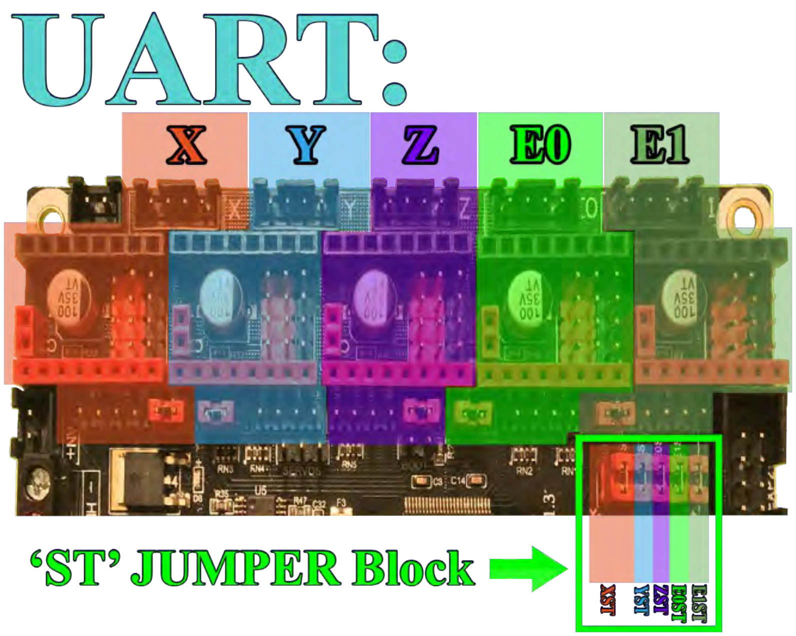
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode

Driver Socket Representation



NOTE: SPI PINS must not be set if UART mode is ensbled.

SET 'ST' JUMPER to enable Sensor-less Homing



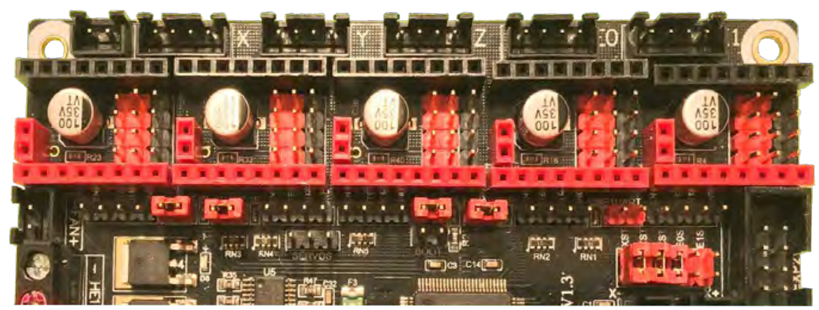
'ST' JUMPER Block

UART Capable Drivers with stallGaurd™ feature: TMC2209

Driver Socket Representation:	Meaning:
U U ⇒	Jumper Set
U U ⇒	set Jumper so that both U PINS are covered by the Jumper

Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS SET. All other PINS need to be empty for UART mode to work properly.

Here is an example of (TMC2209 UART mode) only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS set for X, Y, and Z Axes:



BIQU TMC2209 V1.2

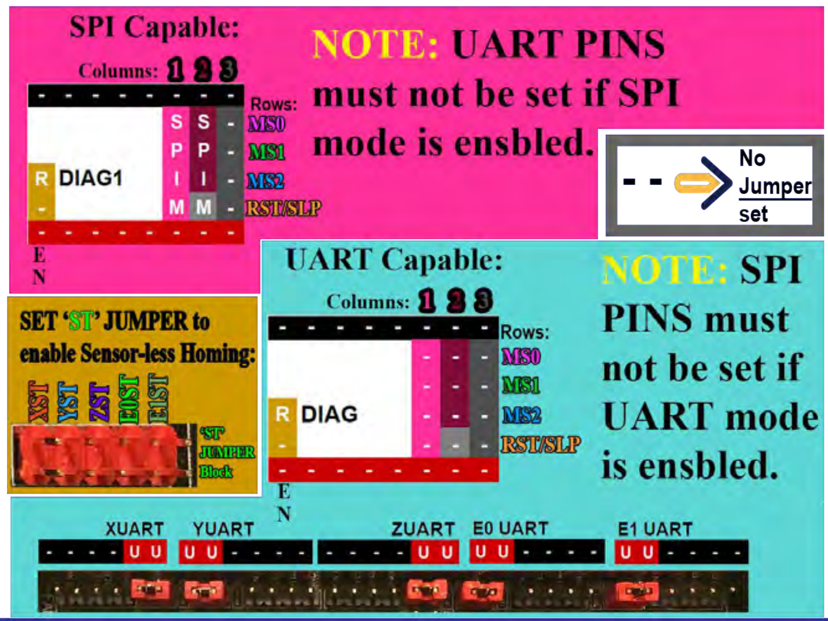
UART Mode

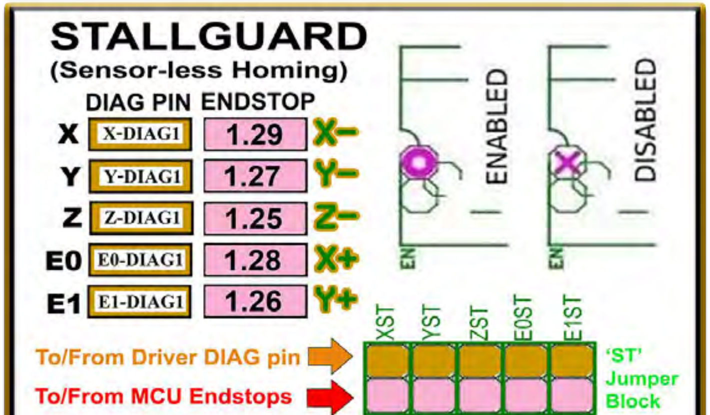
UART Mode

R - Stall Detection

Information on Sensor-less Homing

Driver Socket Representation





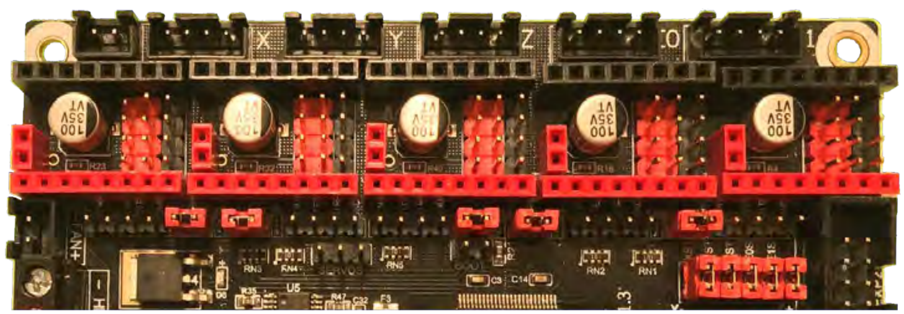
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

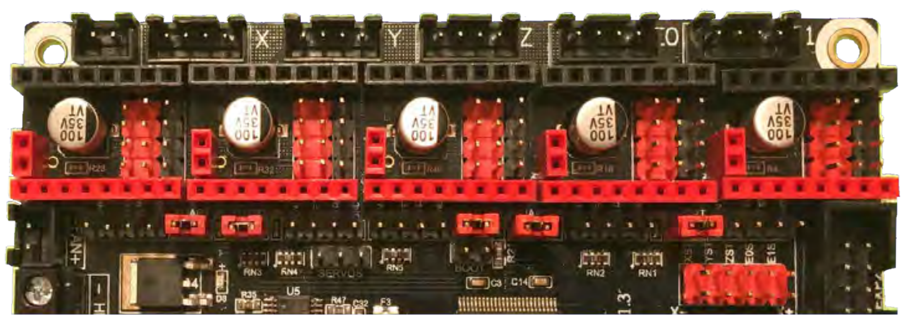
Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161
UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are SET for ALL Axes:



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are REMOVED for ALL Axes:



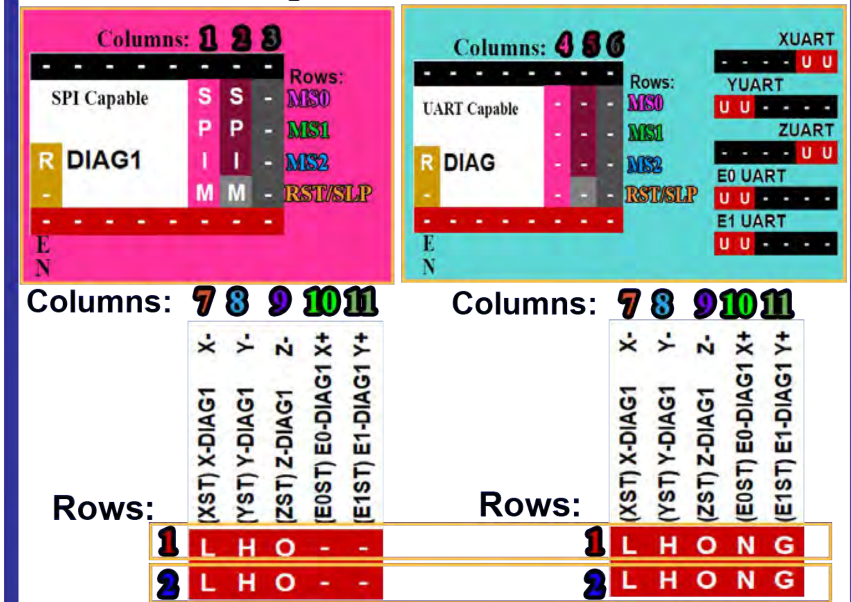
BIQU TMC2209 V1.2

UART Mode

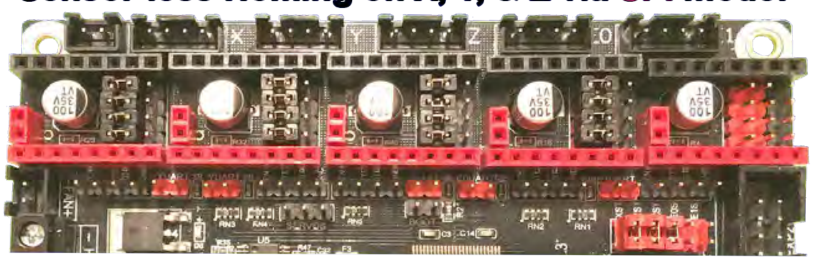
UART Mode

R - Stall Detection

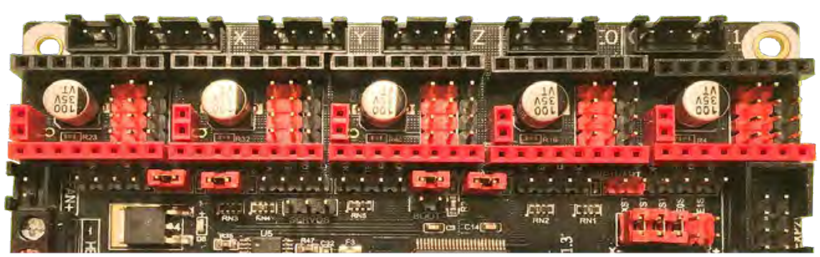
Driver Socket Representation of 'ST' JUMPER Block



Sensor-less Homing on X, Y, & Z via SPI mode:



Sensor-less Homing on X, Y, & Z via UART mode:



Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

L L → X MIN Endstop to enable Sensor-less Homing	H H → Y MIN Endstop to enable Sensor-less Homing
O O → Z MIN Endstop to enable Sensor-less Homing	N N → X MAX Endstop to enable Sensor-less Homing
G G → Y MAX Endstop to enable Sensor-less Homing	- - → No Jumper Set

Driver Socket Representation:		Meaning:
L L	→	set Jumper between row 1 and row 2 in column 7
H H	→	set Jumper between row 1 and row 2 in column 8
O O	→	set Jumper between row 1 and row 2 in column 9
N N	→	set Jumper between row 1 and row 2 in column 10
G G	→	set Jumper between row 1 and row 2 in column 11

UART Mode

UART Mode

R - Stall Detection

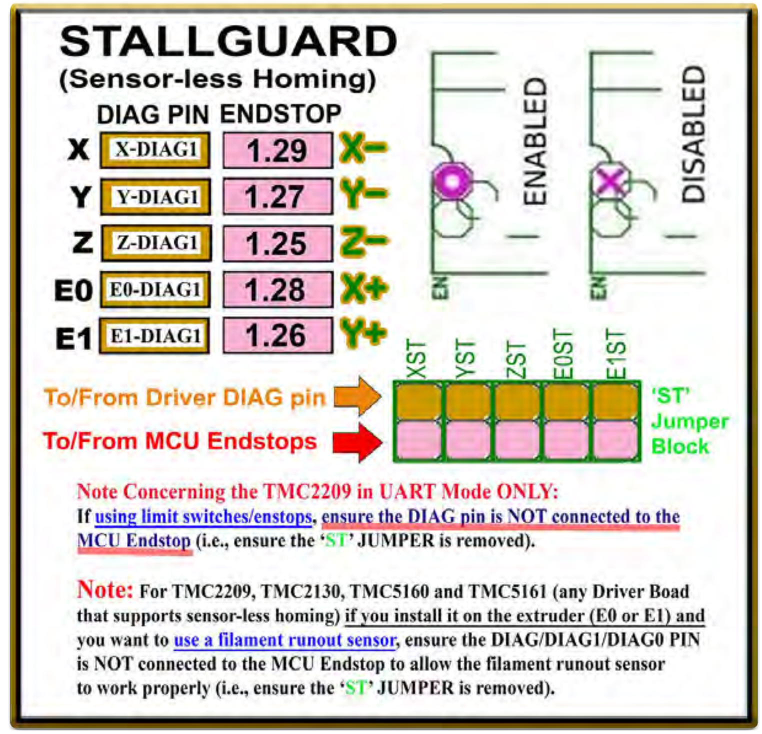
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capabilitiy of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

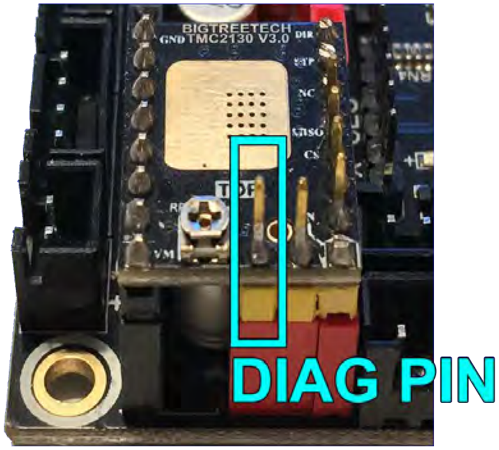
The following DO NOT have Endstops:

- **Extruder Axis (E0 or E1)**
- **Z Axis, if a BLTouch is used to Home instead of a physical endstop**

UART Mode

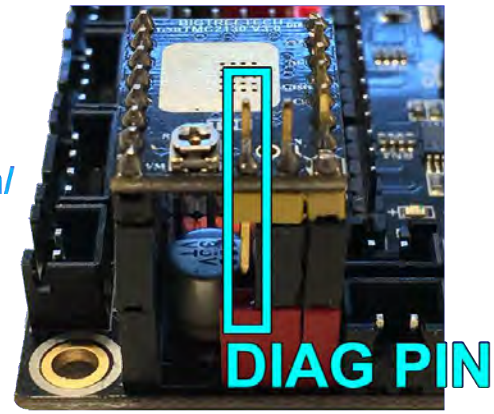
R - Stall Detection

To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.



To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

Link to stackable header pins:
https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P



BIQU TMC2209 V1.2

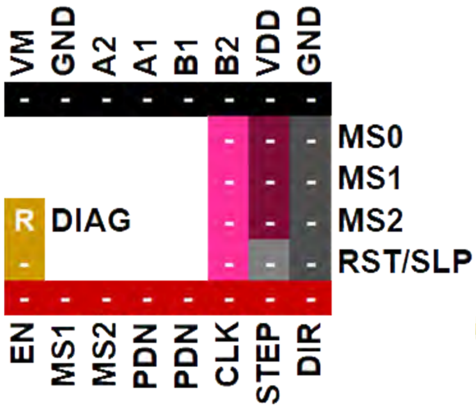
UART Mode

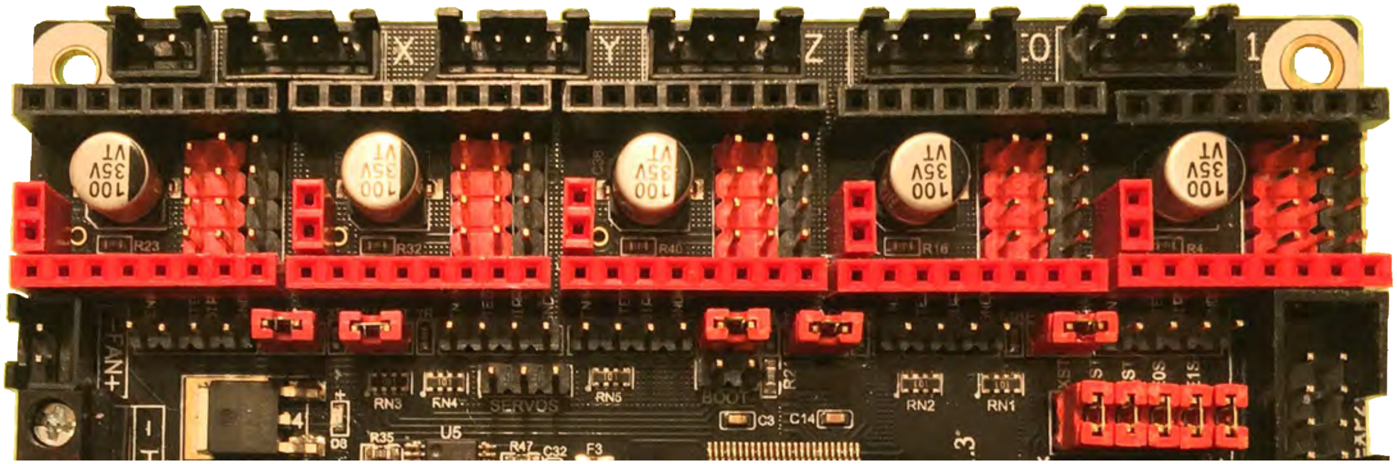
UART Mode

R - Stall Detection

Note: Set the "U" Jumper(s) for UART MODE!

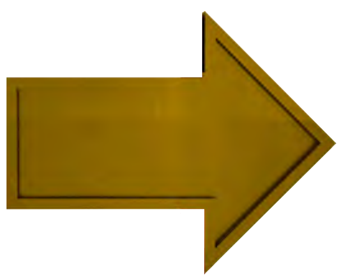
UART

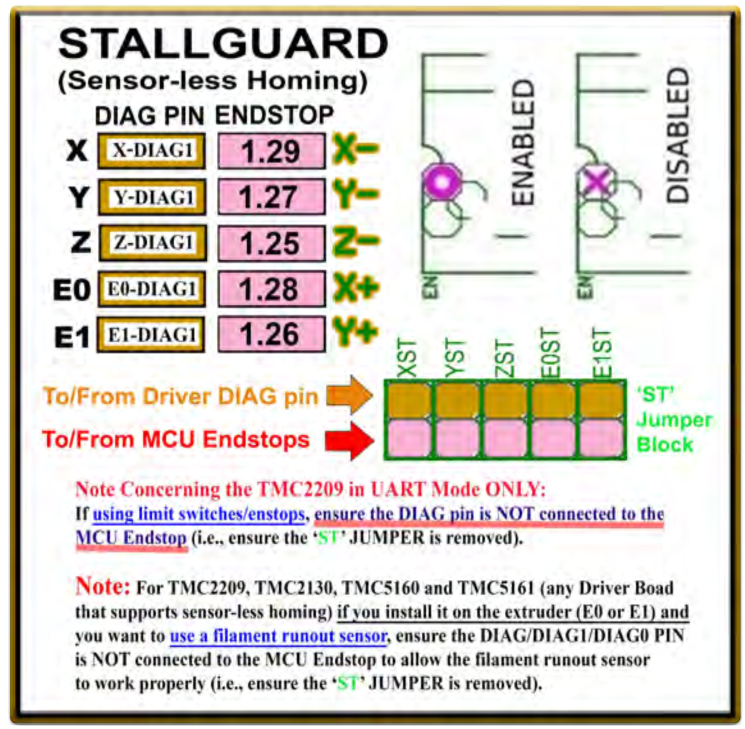




See Appendix D for legend

Note: TMC2209 has sensor-less homing capability





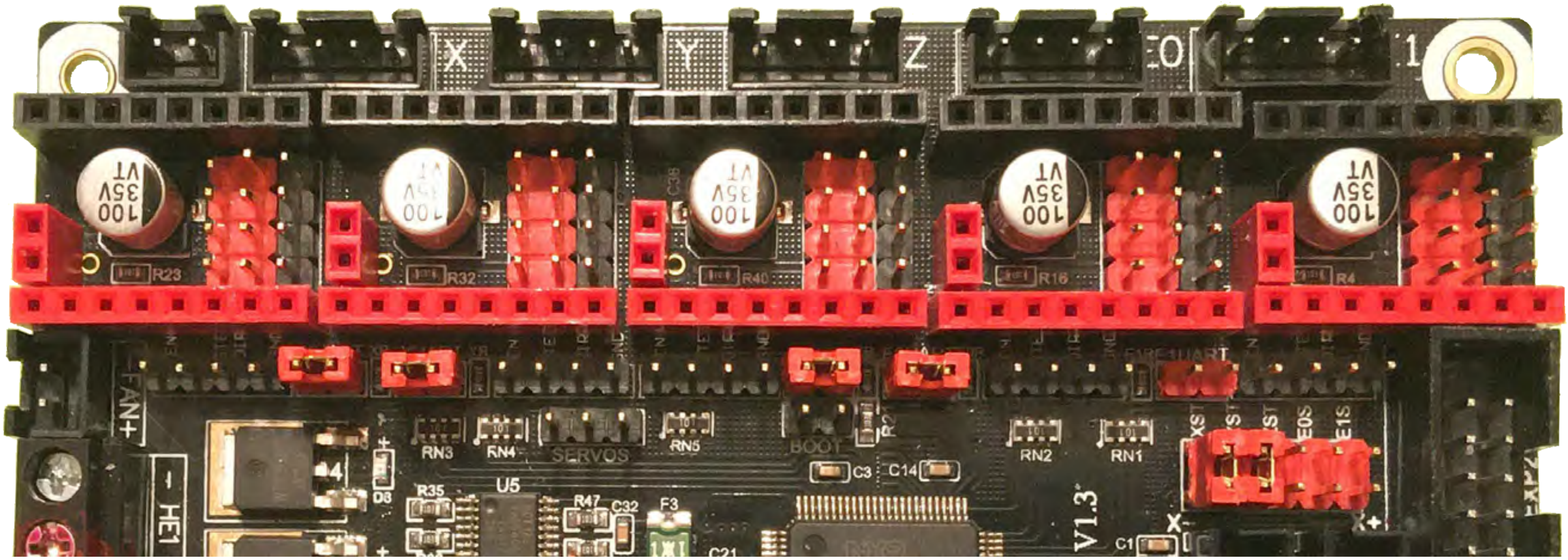
UART Mode

R - Stall Detection

Examples of Different UART Configurations Part 1

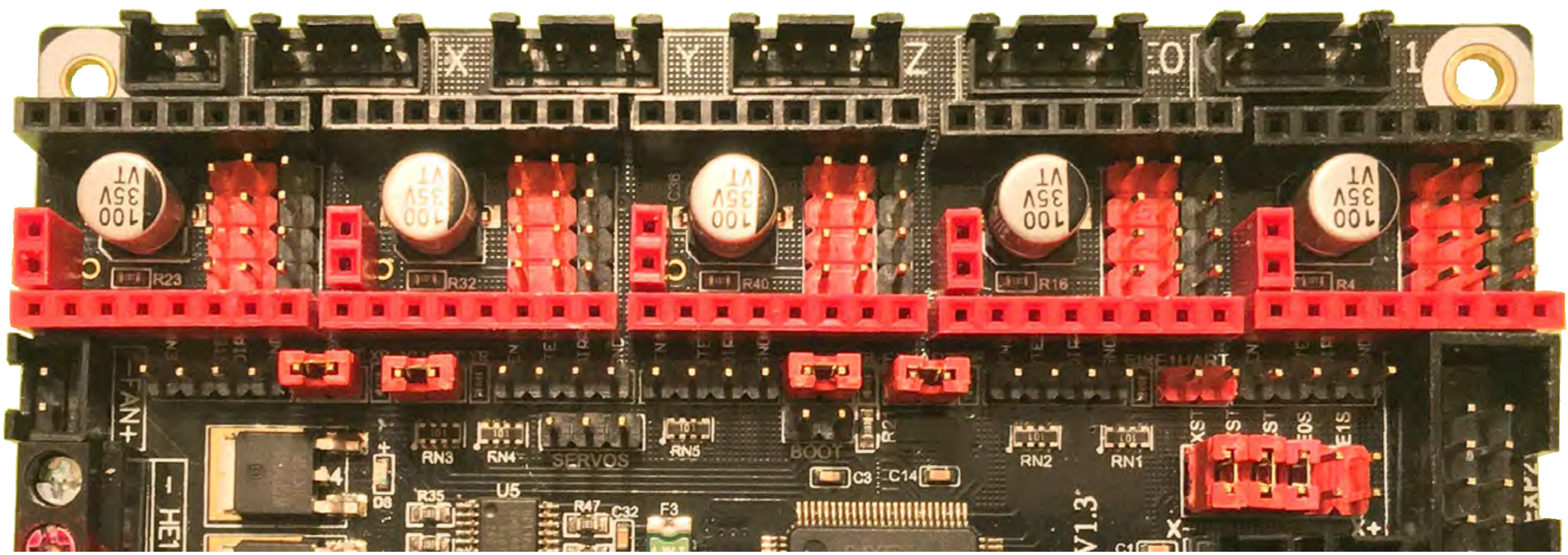
X, Y, Z and E0 axes configured for UART mode.

Sensor-less homing for X and Y axes.



X, Y, Z and E0 axes configured for UART mode.

Sensor-less homing for X, Y and Z axes.

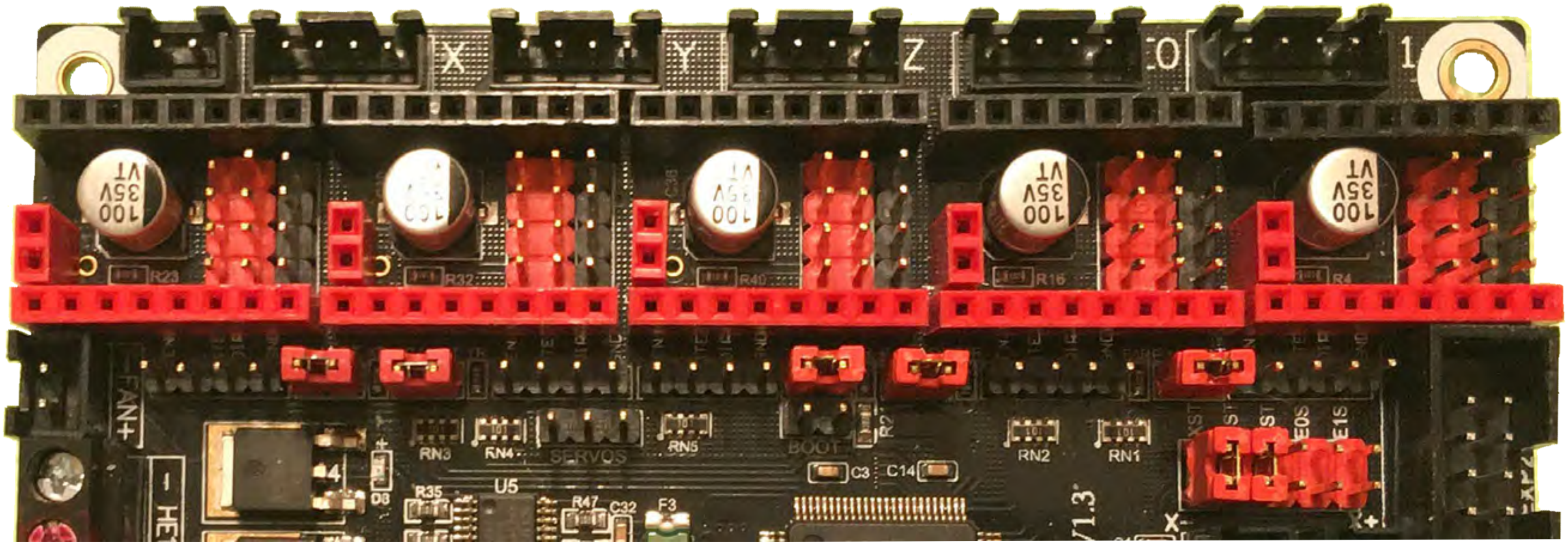


UART Mode
R - Stall Detection

Examples of Different UART Configurations Part 2

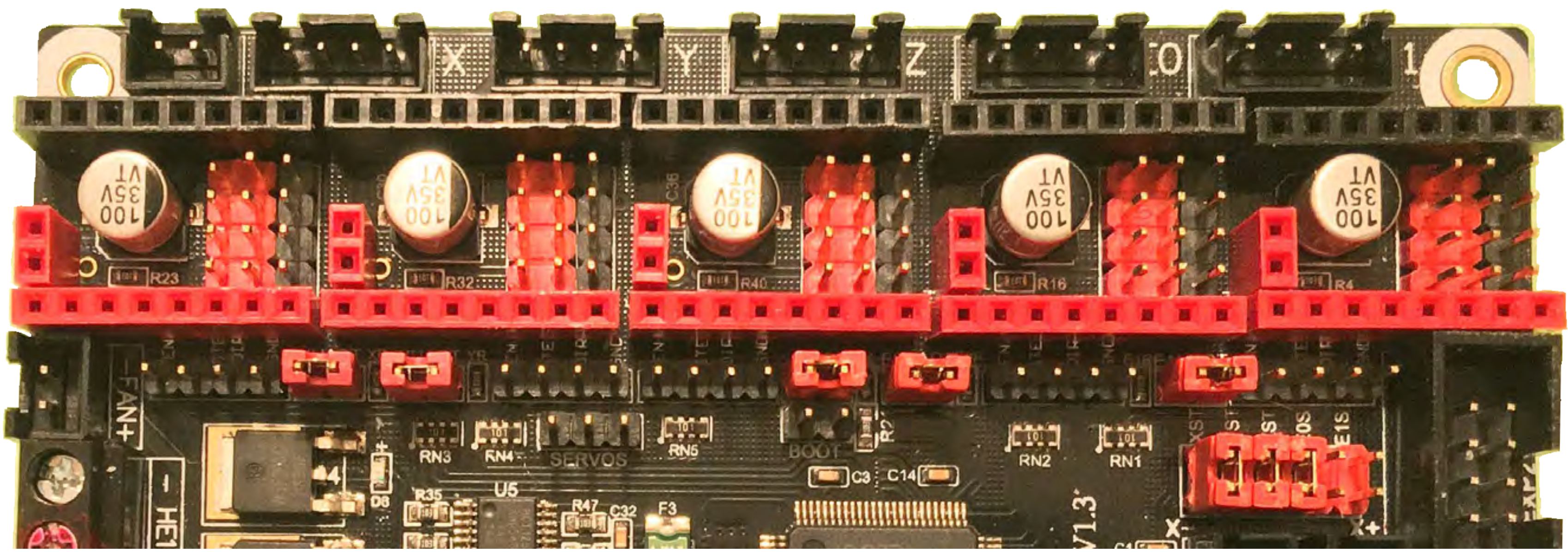
X, Y, Z, E0, and E1 axes configured for UART mode.

Sensor-less homing for X and Y axes.



X, Y, Z, E0 and E1 axes configured for UART mode.

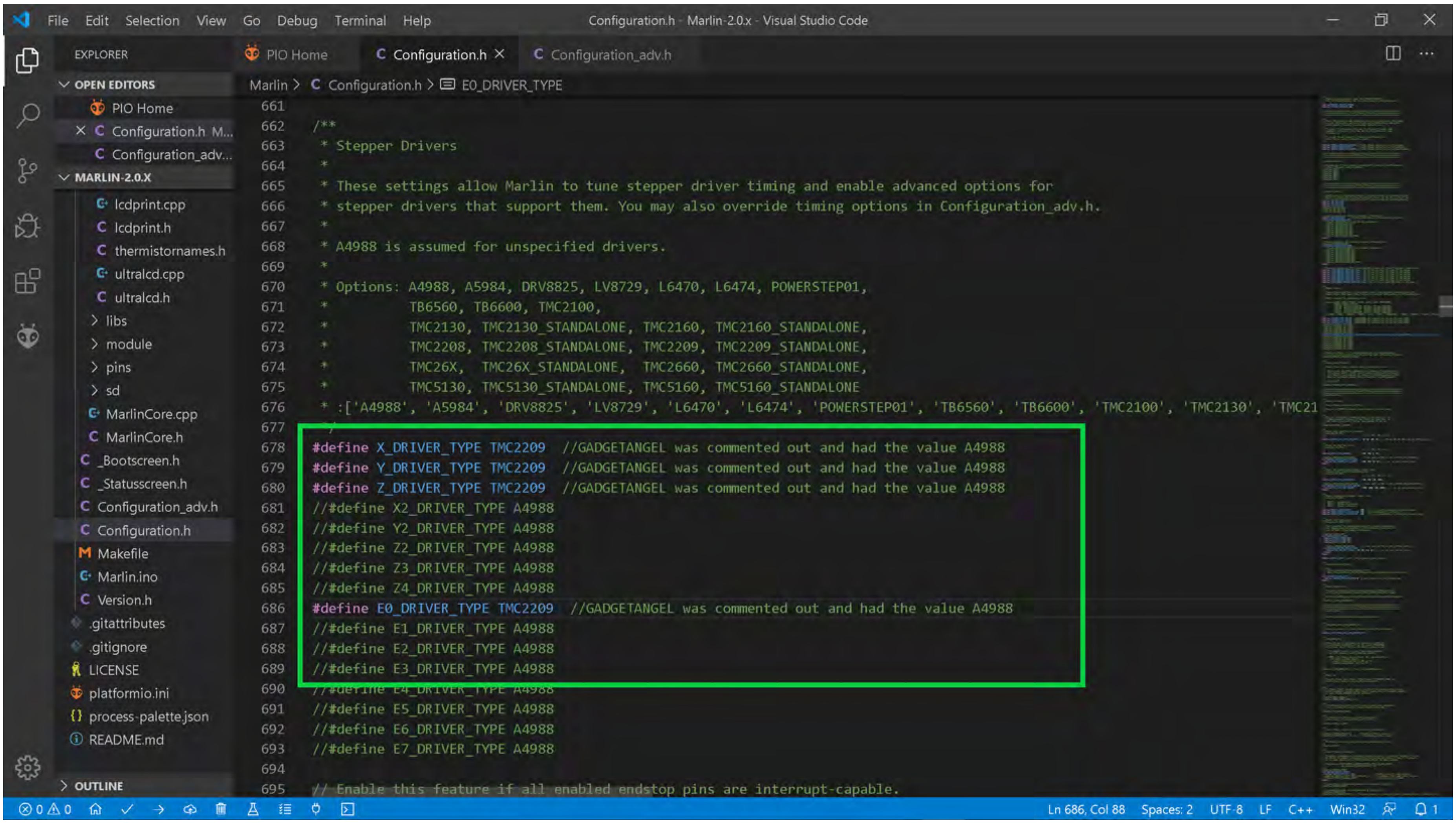
Sensor-less homing for X, Y and Z axes.



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.

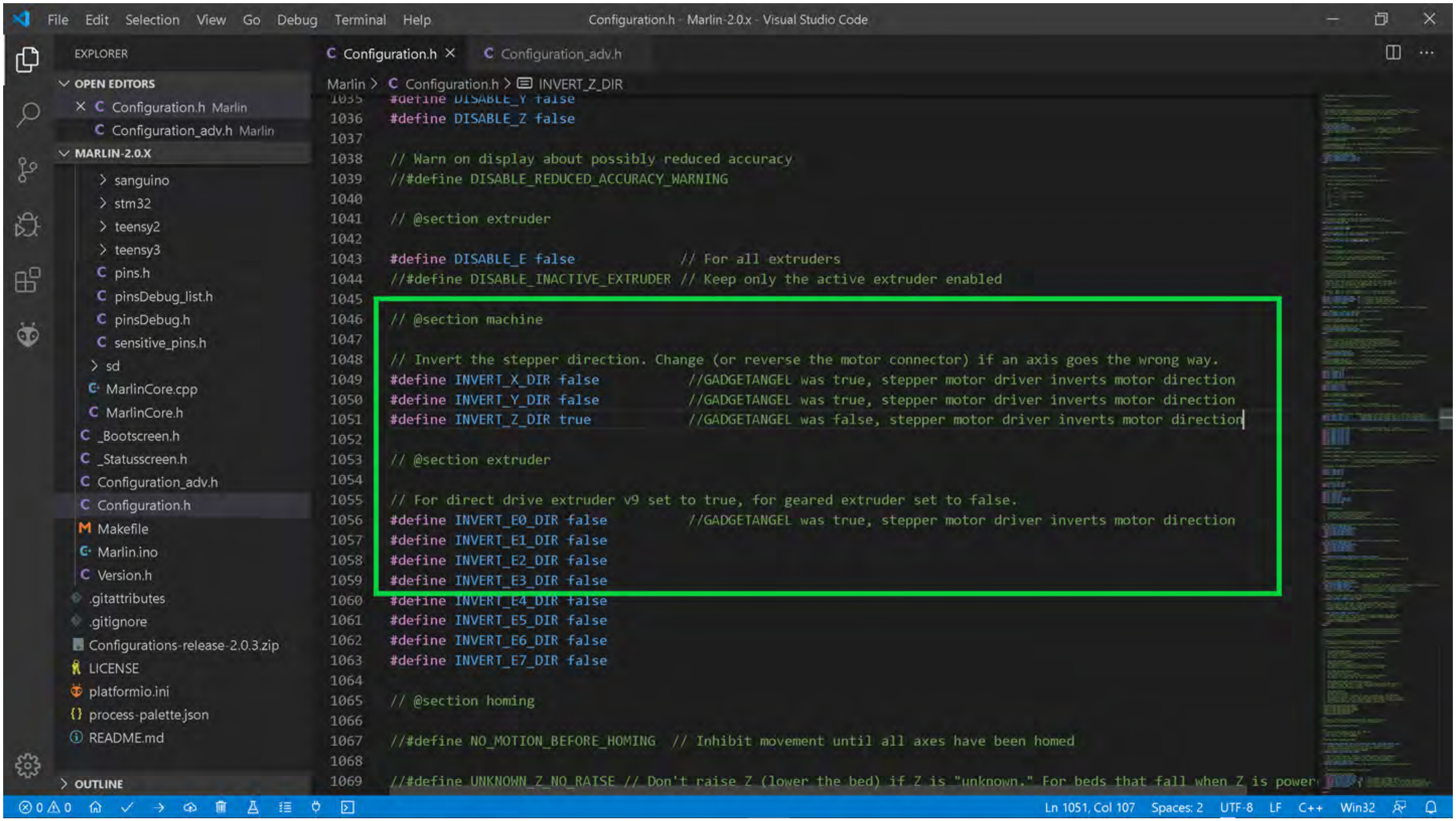
- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in UART mode. When two "//" appear at the beginning of a line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



- Go to the next page.

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

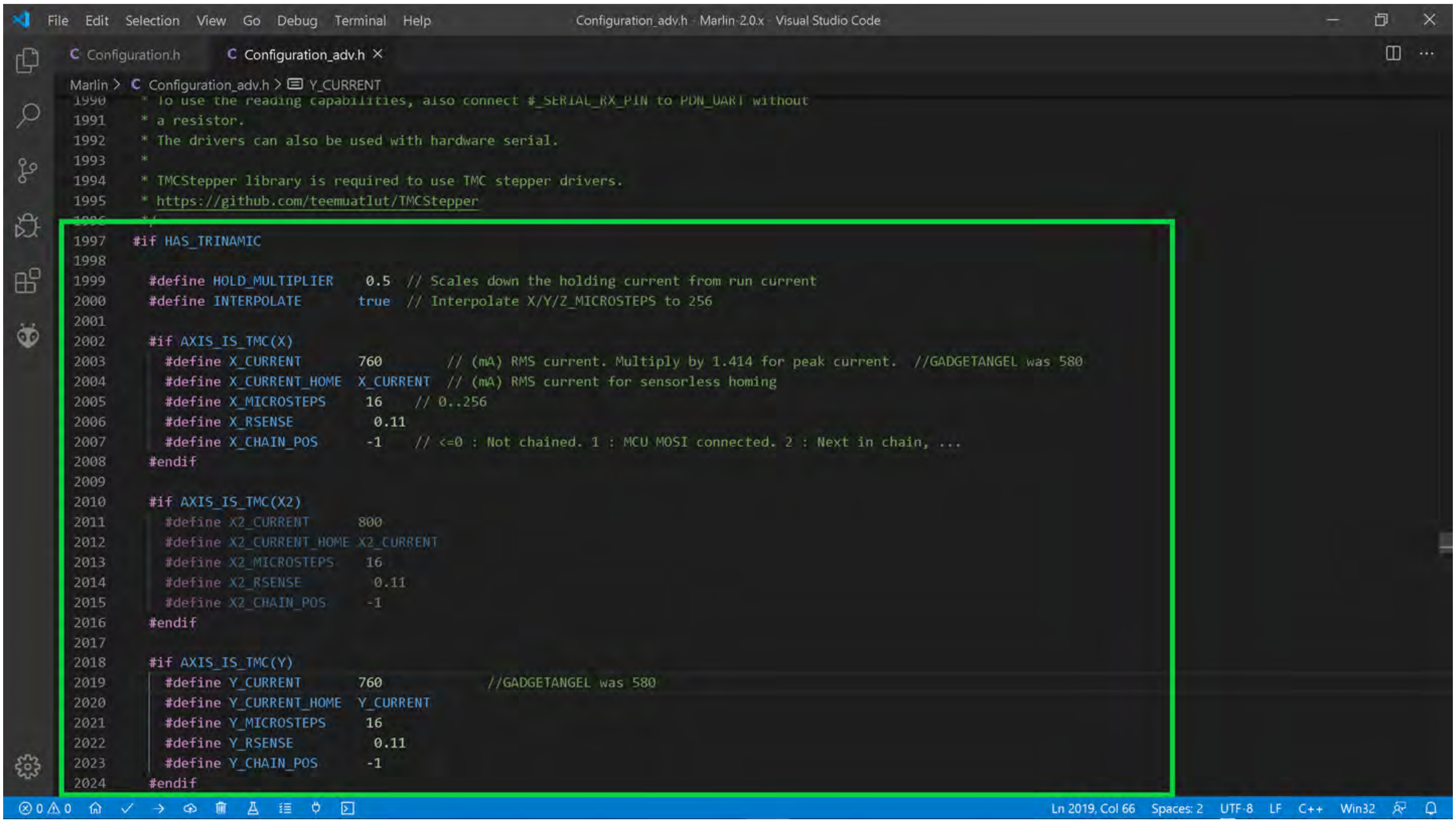
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 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



- Go to the next page.

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

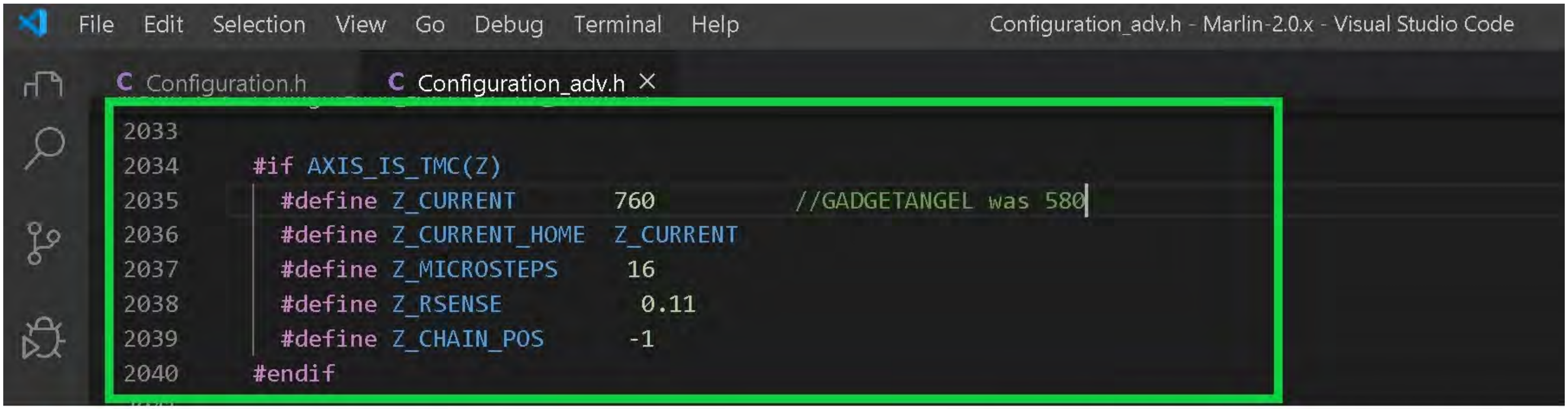
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2209 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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".

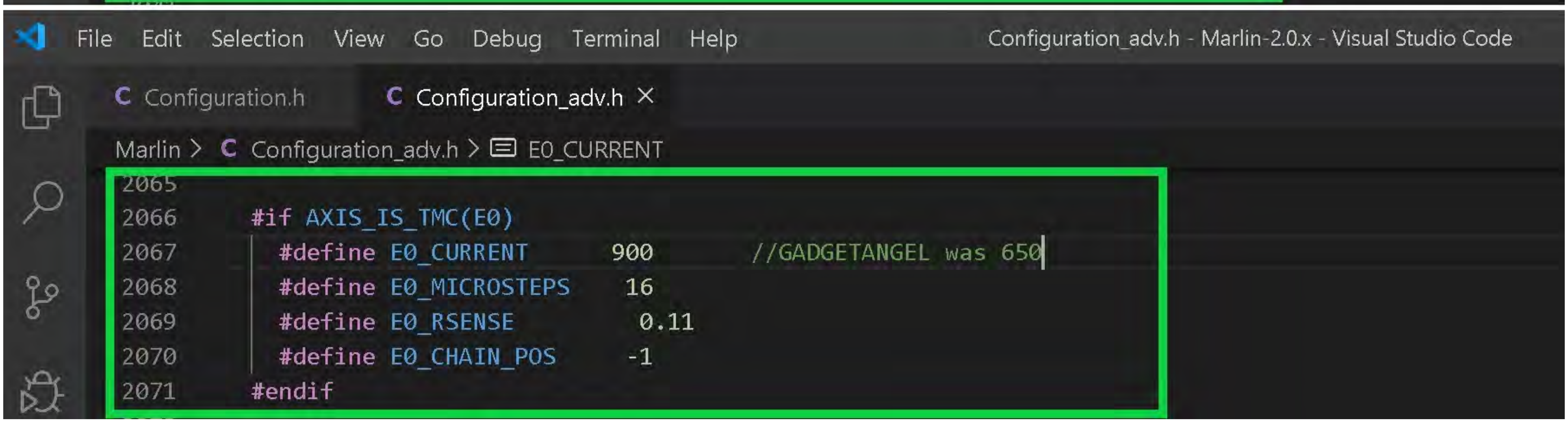


- Go to the next page.

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".

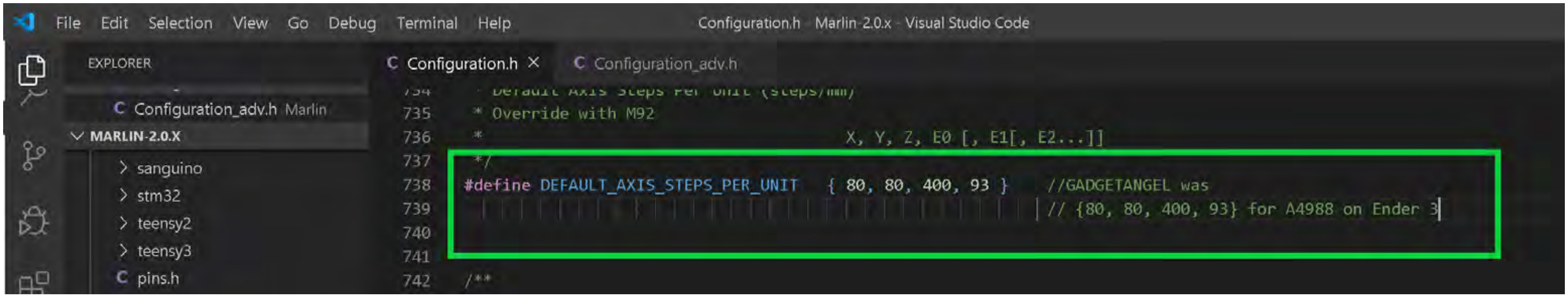




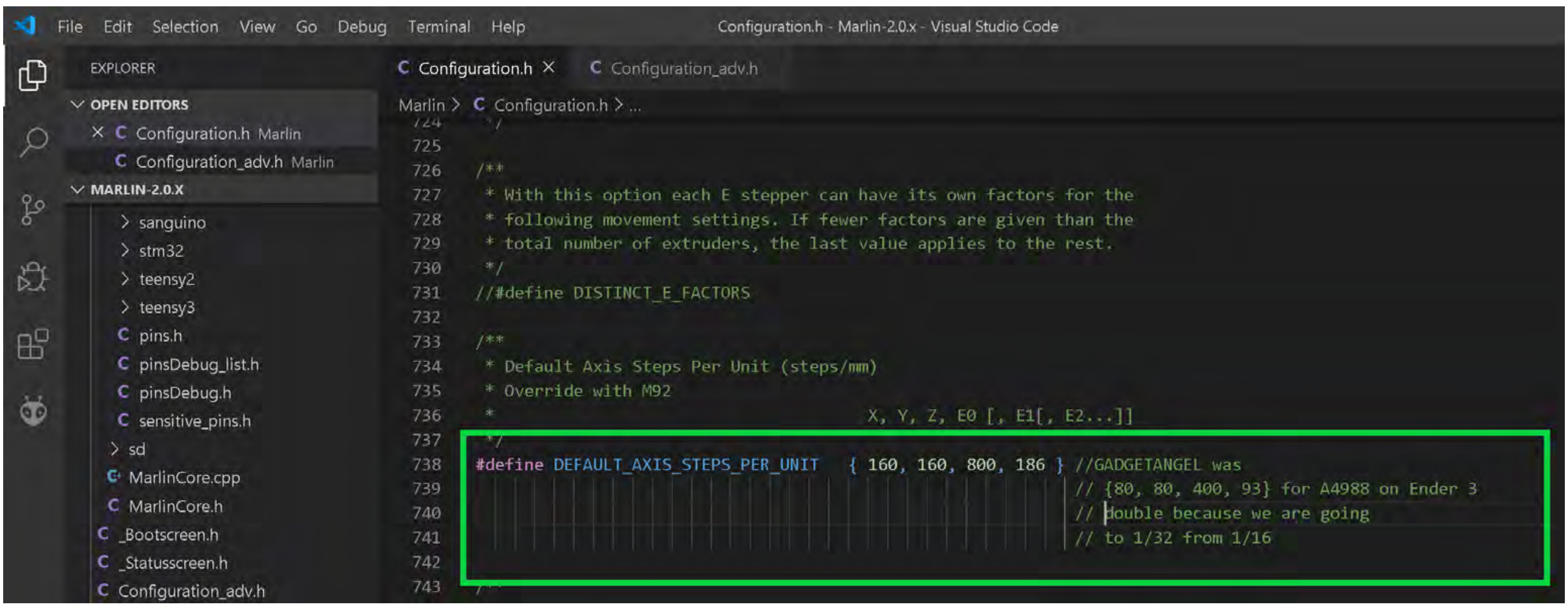
- Go to the next page.

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



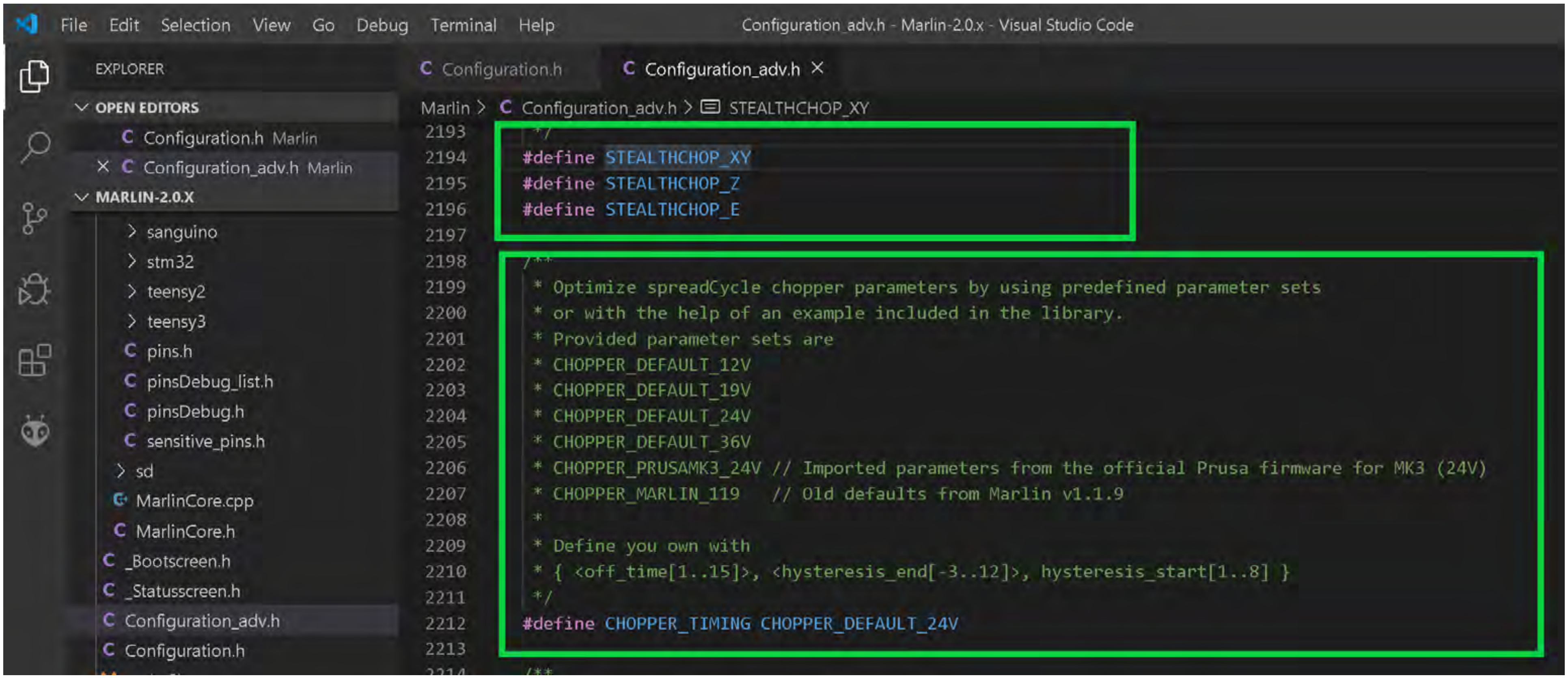
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



- Go to the next page.

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

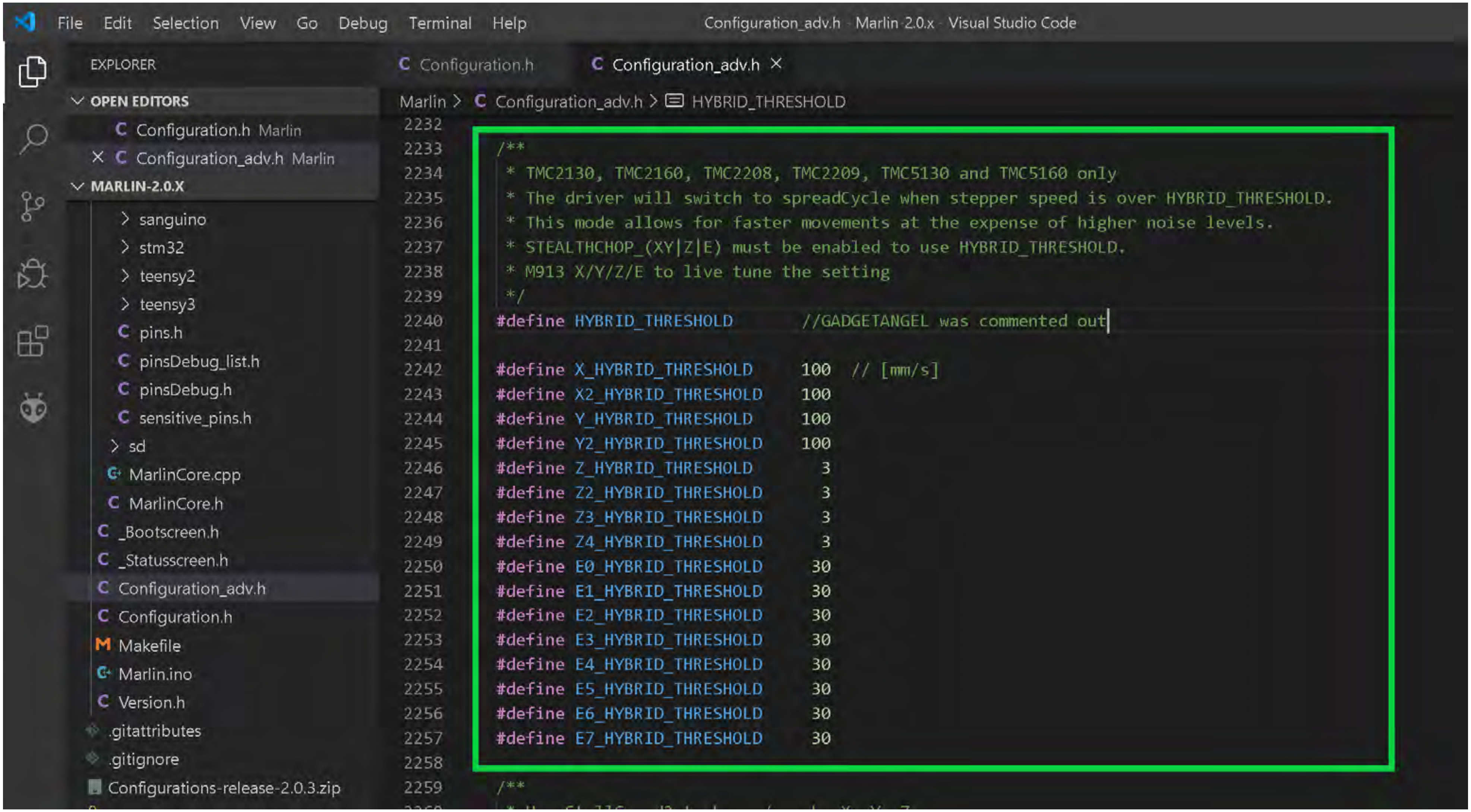
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

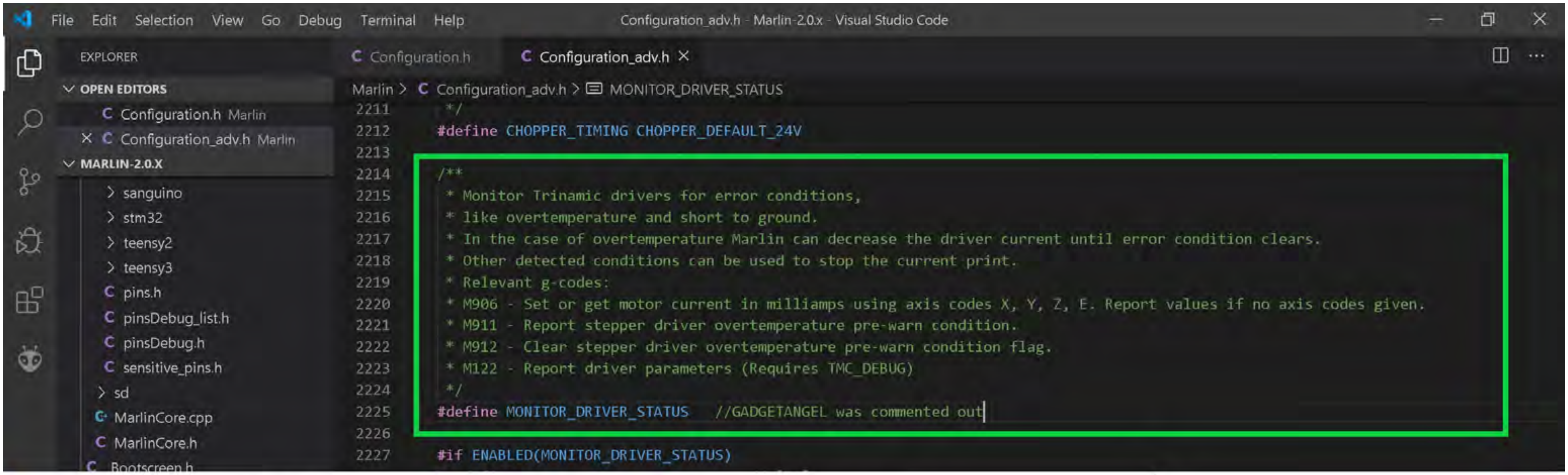
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

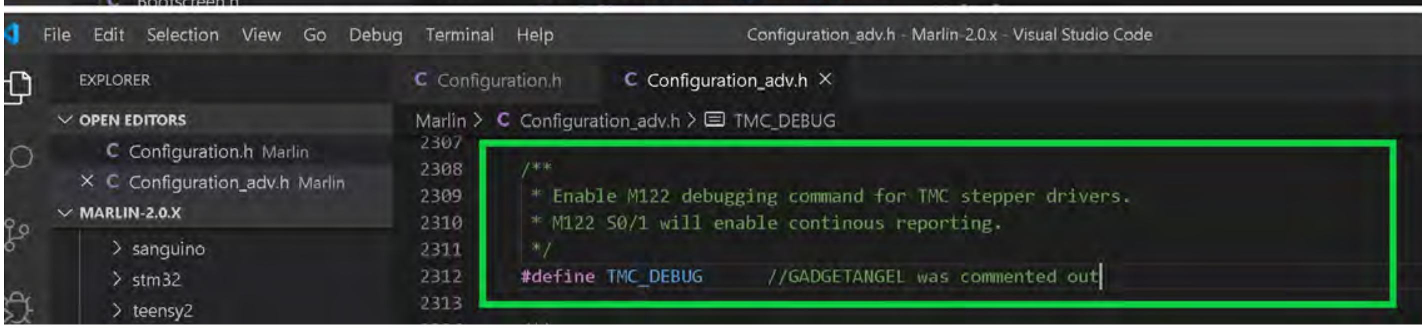


- Go to the next page.

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.**

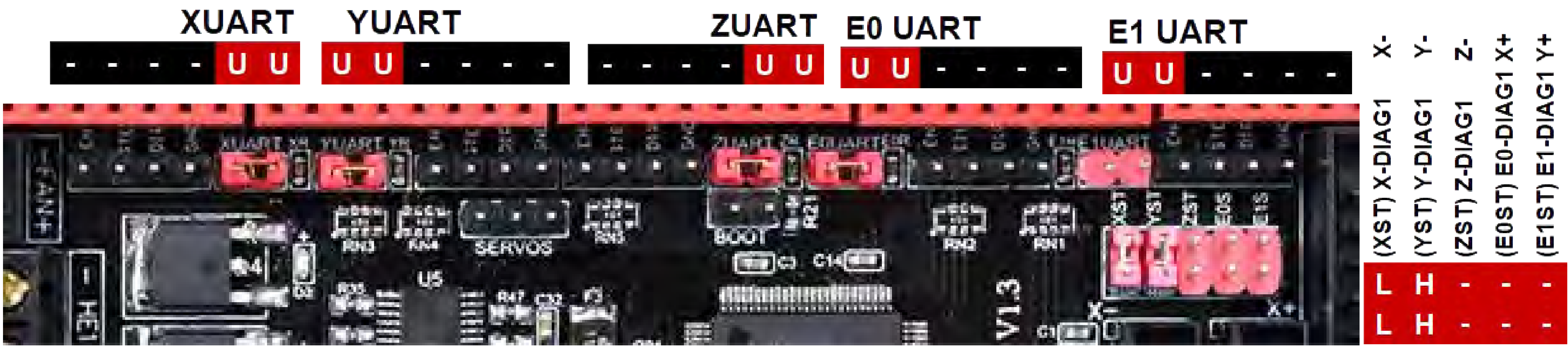




- Go to the next page.

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 DIAG1 pin on the TMC2209 driver has to be plugged into the SKR V1.3 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 TMC2209 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC2209 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC2209 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the following jumpers are in place: are the "L" and "H" jumpers to enable sensor-less homing for X and Y axes. These two jumper will connect my X DIAG1 pin and Y DIAG1 pin to their respective endstops. I will set the UART jumper for X, Y, Z and E0.

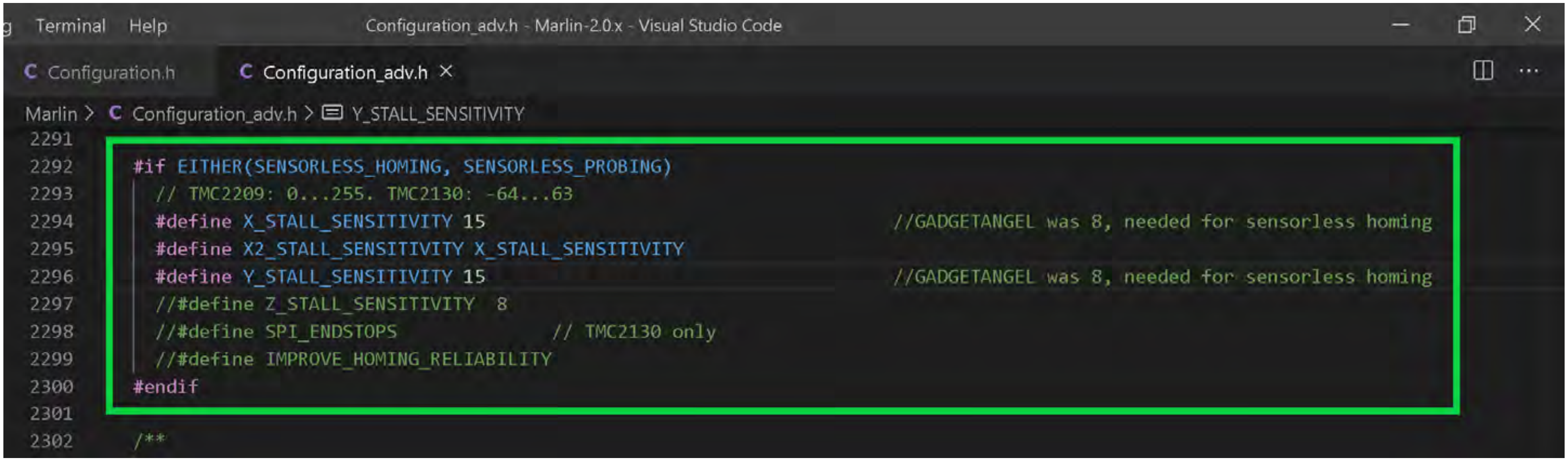


- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

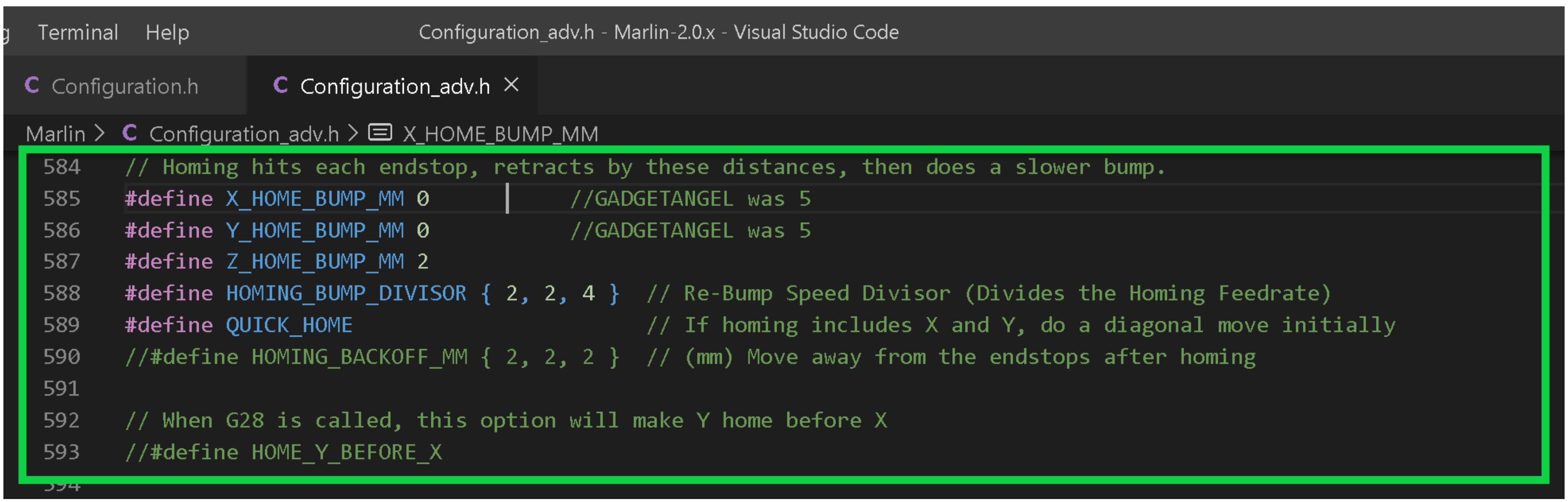
- Go to the next page.

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".



- 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.



- 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 and 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

Configuration.h × Configuration_adv.h

Marlin > Configuration.h > ENDSTOPPULLUPS

```
// Enable pullup for all endstops to prevent a floating state
#define ENDSTOPPULLUPS            //GADGETANGEL this should always be uncommented for an Ender 3
#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  //#define ENDSTOPPULLUP_XMAX
  //#define ENDSTOPPULLUP_YMAX
  //#define ENDSTOPPULLUP_ZMAX
  //#define ENDSTOPPULLUP_XMIN
  //#define ENDSTOPPULLUP_YMIN
  //#define ENDSTOPPULLUP_ZMIN
  //#define ENDSTOPPULLUP_ZMIN_PROBE
#endif
```

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

Configuration.h × Configuration_adv.h

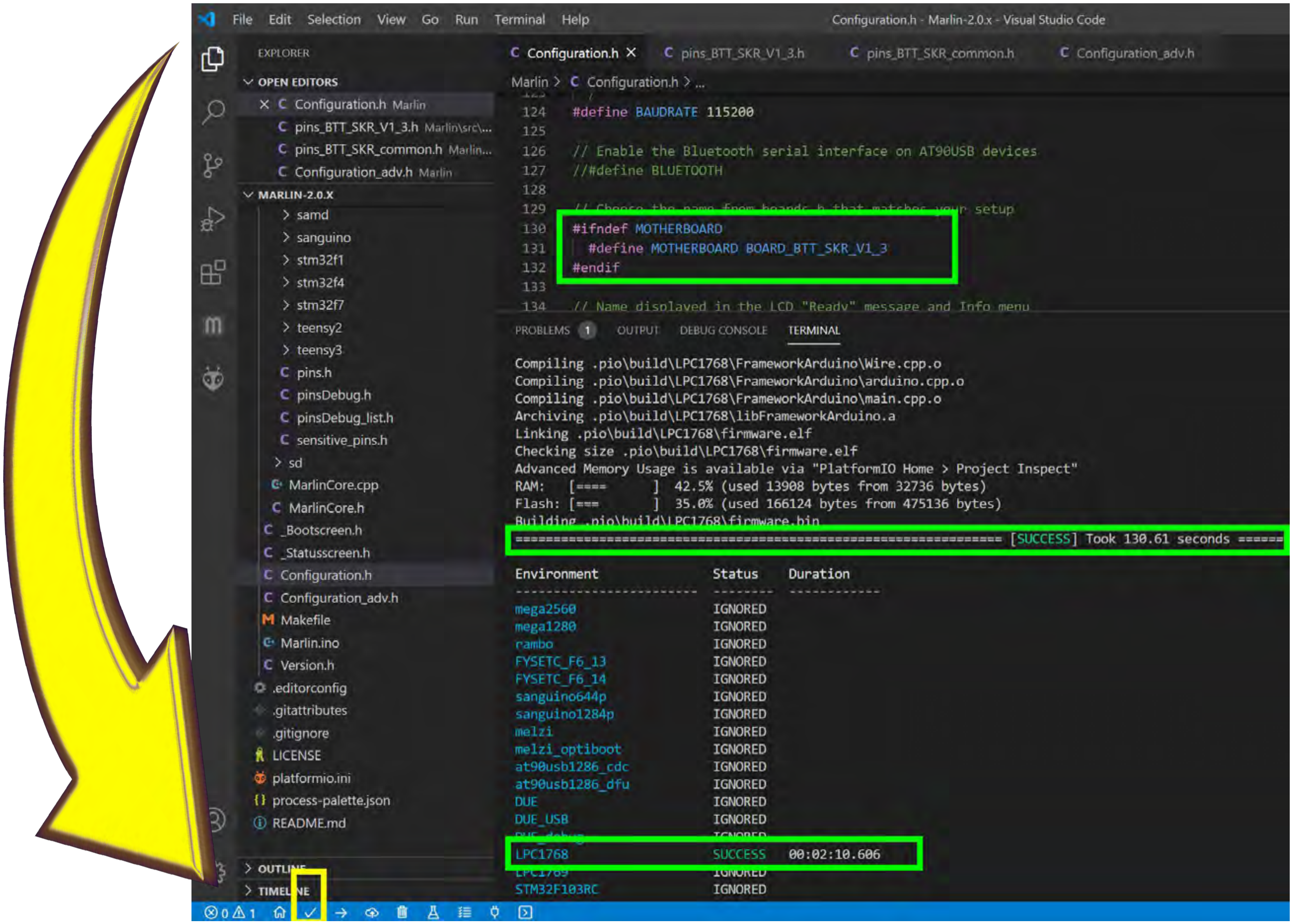
Marlin > Configuration.h > X_MIN_ENDSTOP_INVERTING

```
// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
#define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
```

- Go to the next page.

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

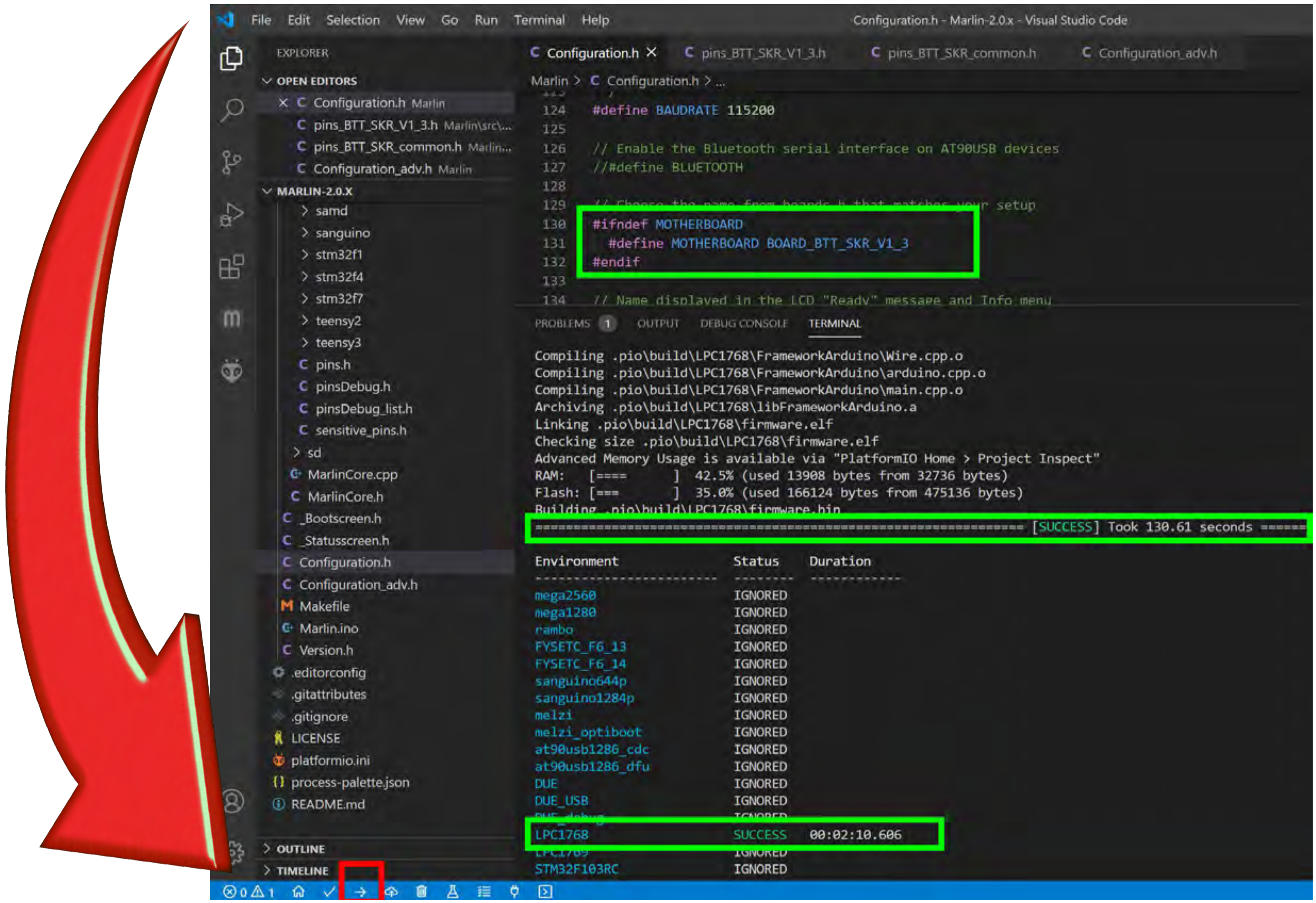
- The end of Marlin setup for BIQU TMC2209 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

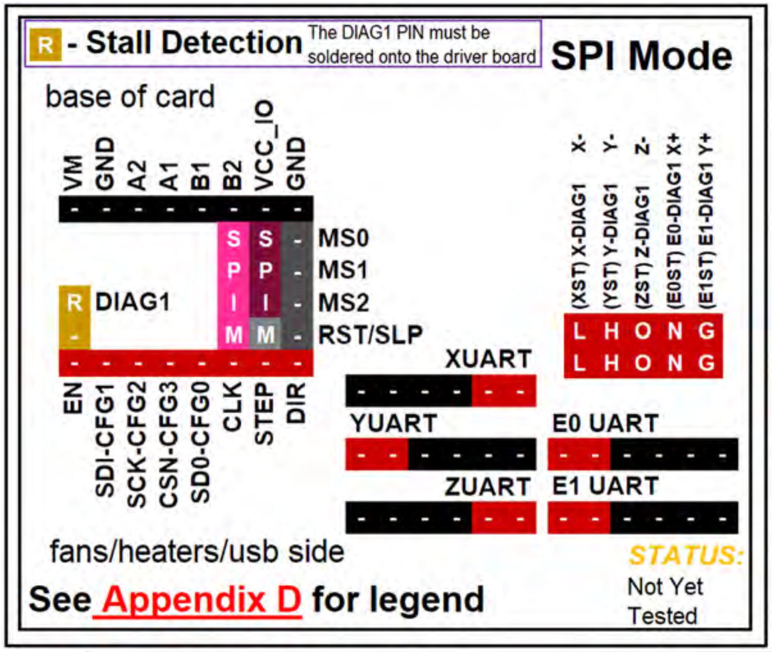
- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

BIQU TMC5160 V1.2

SPI Mode



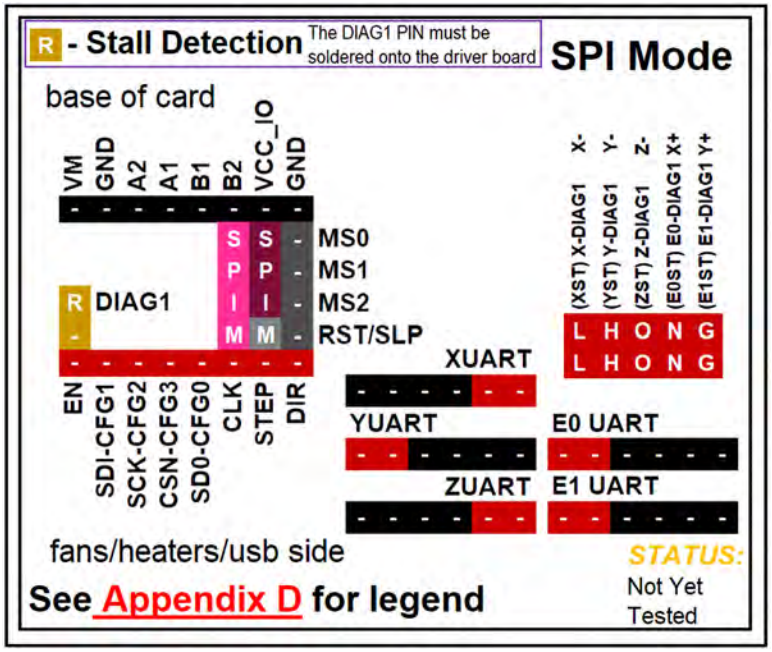
Note: You can use 50% to 90% of the calculated I_{RMS} ($I_{MAX}/1.414$) when tuning ("X_CURRENT", "Y_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

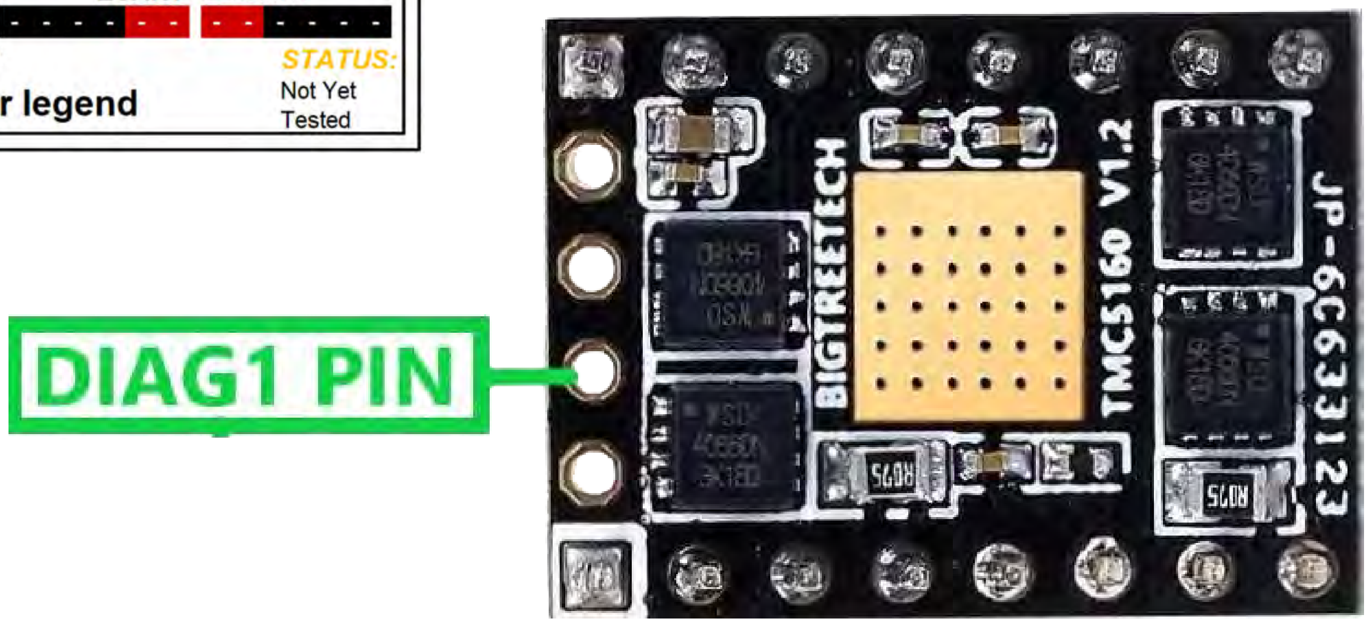
Driver Chip	Steps are set inside of your Firmware	
BIQU® TMC5160 SPI Mode Maximum 256 Subdivision 40V DC 4.3A (peak)		
Driving Current Calculation Formula R_S(Typical Sense Resistor)=0.075 Ω	$I_{MAX}=4.333$ See Appendix B #8. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX}*0.90=3.900$	Current Limit is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

BIQU TMC5160 V1.2

SPI Mode



Note: The TMC5160 V1.2 by default comes in SPI mode. The BIQU TMC5160 does NOT come with a POT or "V_{ref} Test point" location so the I_{RMS} 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", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_CURRENT".

R_s = R050 is 0.05 Ohms

R_s = R062 is 0.062 Ohms

R_s = R068 is 0.068 Ohms

R_s = R075 is 0.075 Ohms

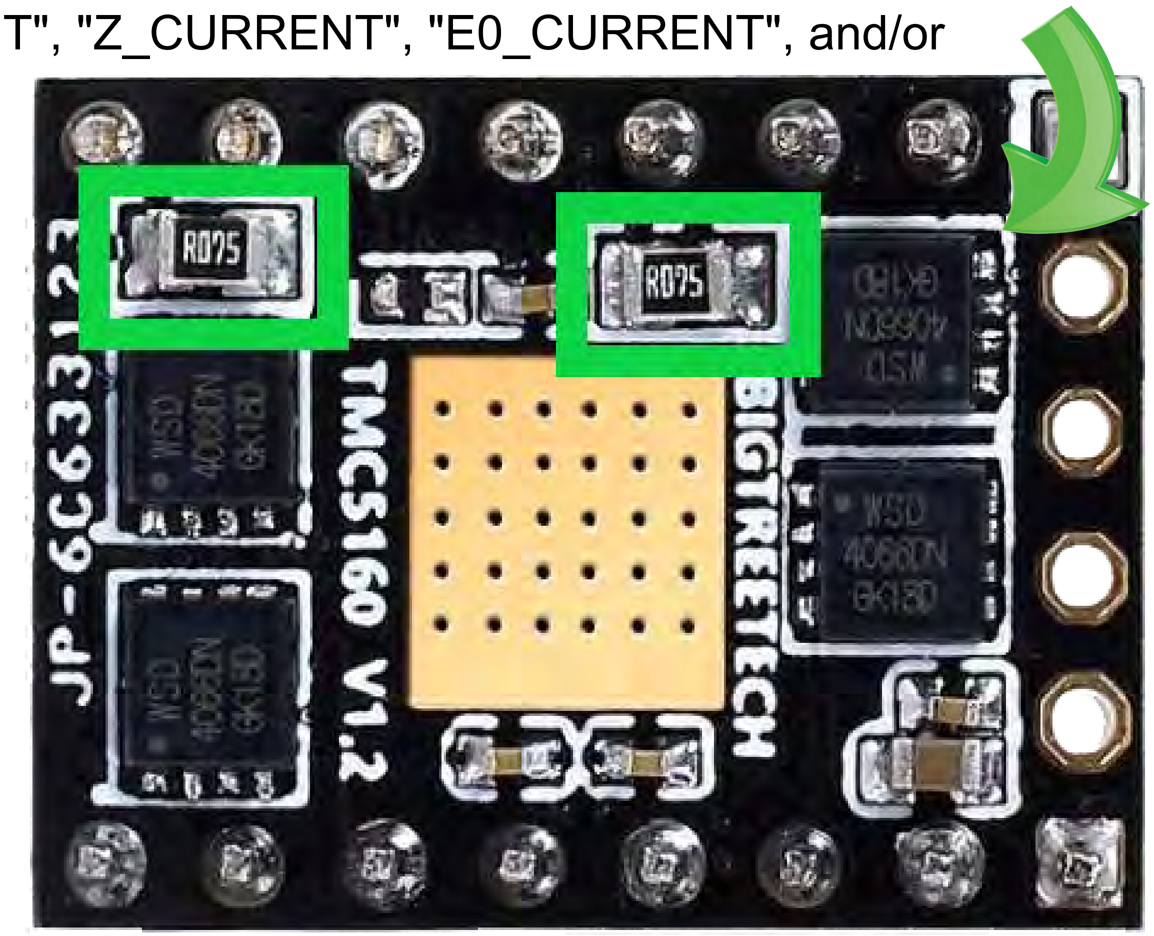
R_s = R100 is 0.1 Ohms

R_s = R110 is 0.11 Ohms

R_s = R150 is 0.15 Ohms

R_s = R200 is 0.2 Ohms

R_s = R220 is 0.22 Ohms



SPI Mode

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	-	-
L	H	O	-	-

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	N	G
L	H	O	N	G

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

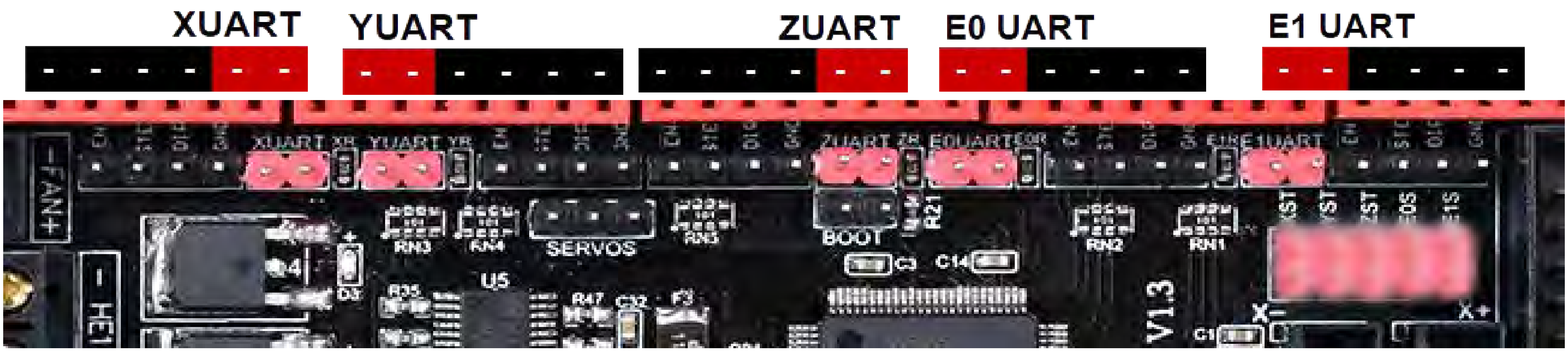
Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
-	-	-	-	-
-	-	-	-	-

Note: If sensor-less homing is not wanted ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".

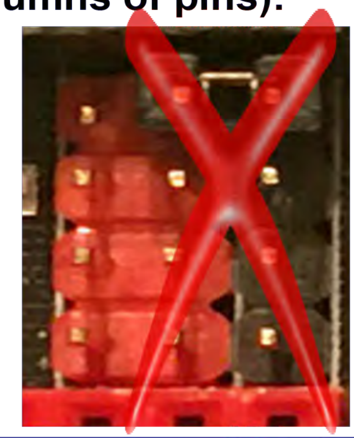
SPI Mode

Note: Ensure that ALL the UART jumpers are EMPTY.

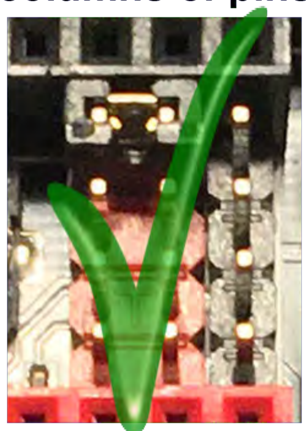


SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

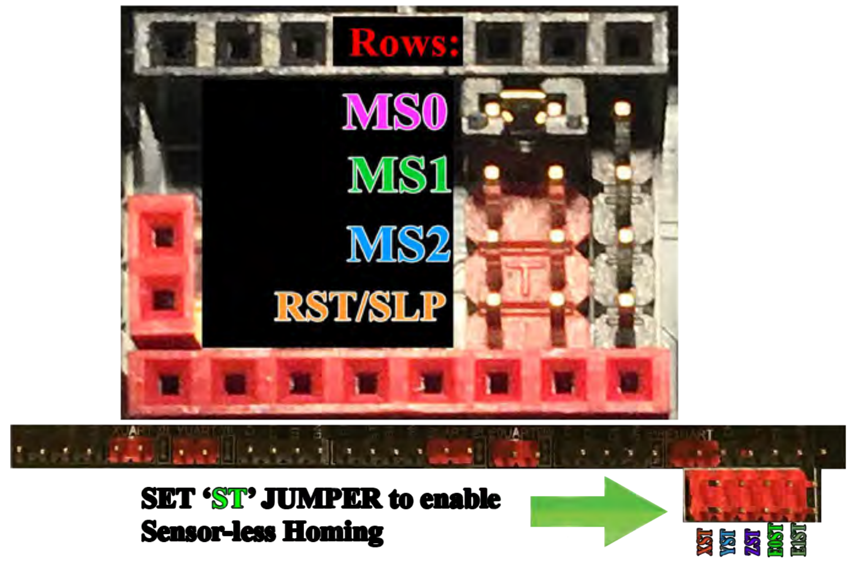
Example of the **WRONG** way to set the "S" jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the "S" jumper for SPI mode (left two columns of pins):

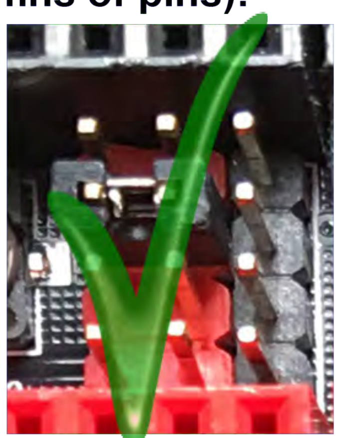


S S ⇒ Jumper set

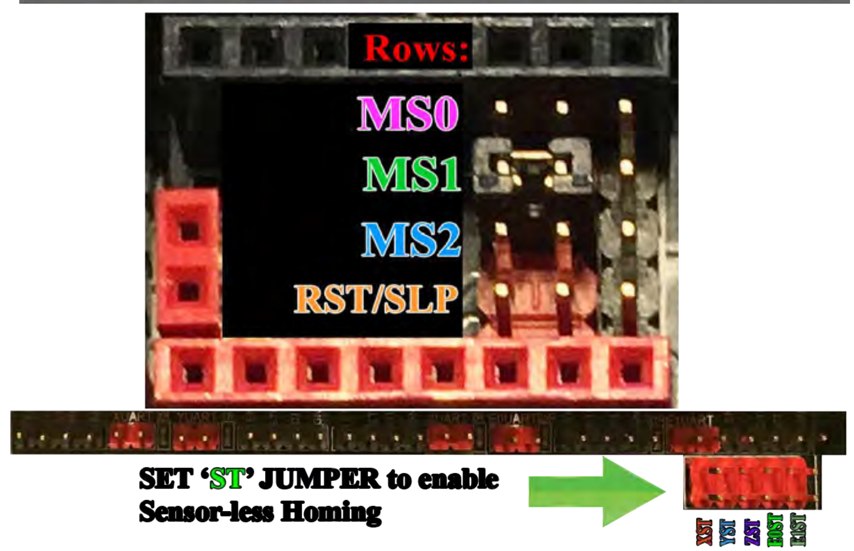


Example of the **WRONG** way to set the "P" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "P" jumper for SPI mode (left two columns of pins):



P P ⇒ Jumper set



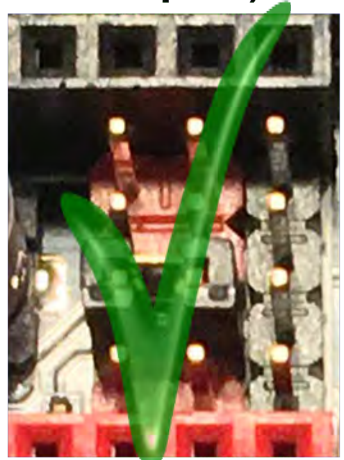
SPI Mode

BIQU TMC5160 V1.2

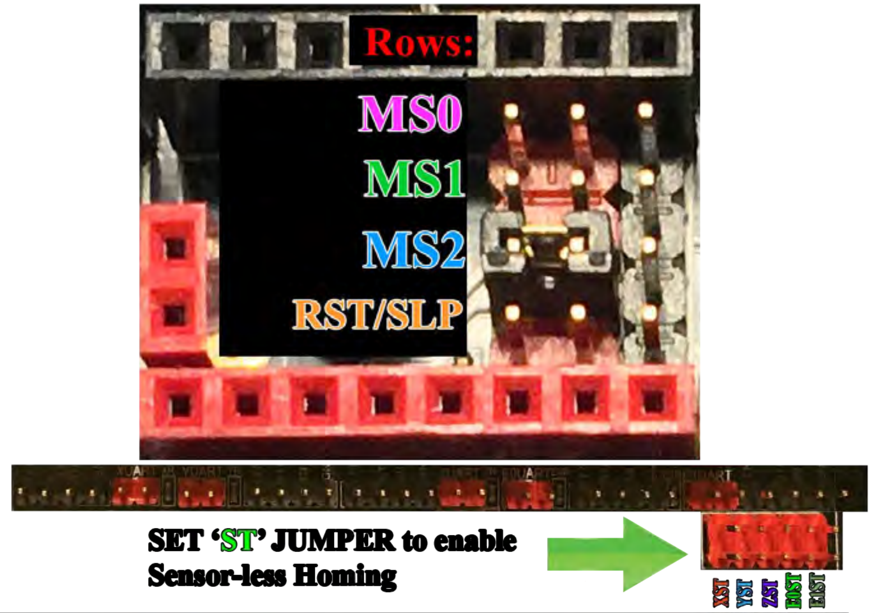
SPI Mode

Example of the **WRONG** way to set the "I" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "I" jumper for SPI mode (left two columns of pins):



I I ⇒ Jumper set

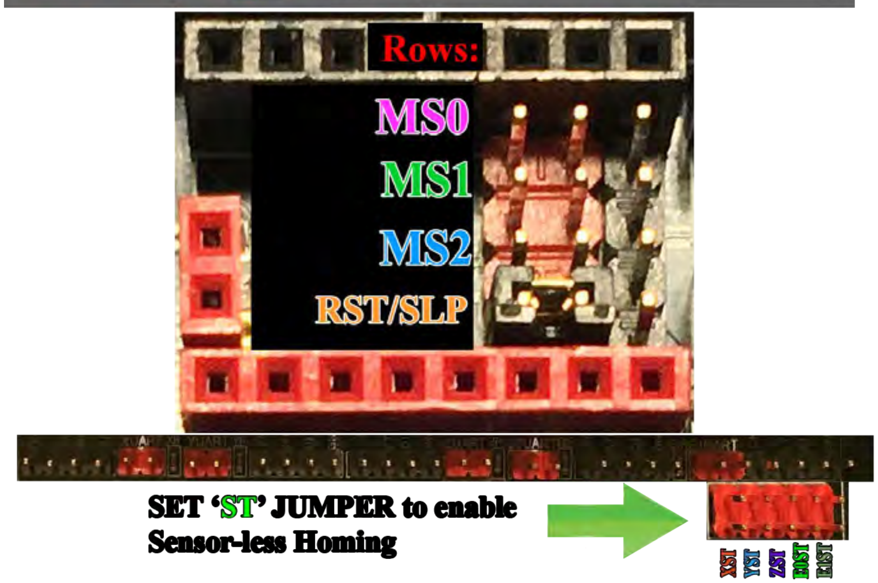


SET 'ST' JUMPER to enable Sensor-less Homing

Example of the **WRONG** way to set the "M" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "M" jumper for SPI mode (left two columns of pins):

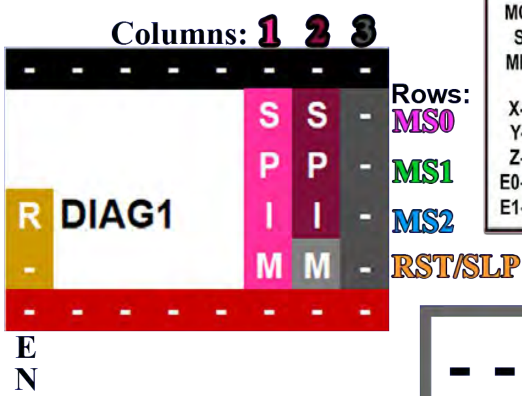
M M ⇒ Jumper set

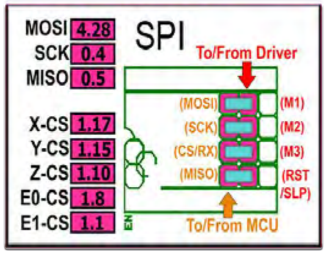


SET 'ST' JUMPER to enable Sensor-less Homing

SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

Driver Socket Representation



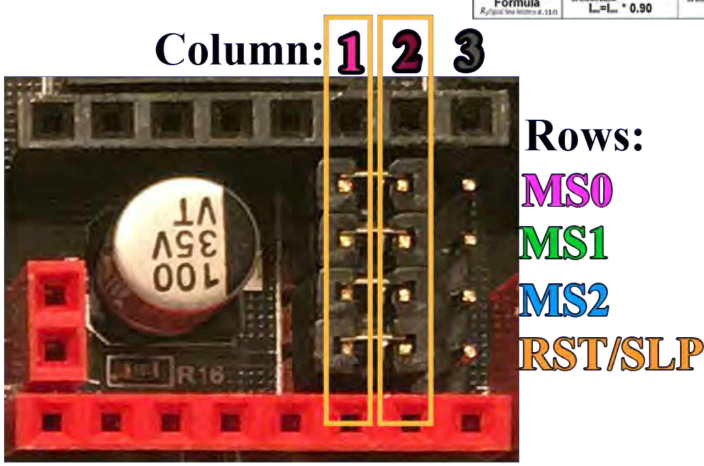


- - ⇒ No Jumper set

NOTE: UART PINS must not be set if SPI mode is ensbled.

SPIM:

Driver Chip Chart:



SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

S S
P P
I I
M M
⇒

Meaning:

Jumpers Set

Driver Socket Representation: S S ⇒ Meaning: **set Jumper between column 1 and column 2 on the MS0 row**

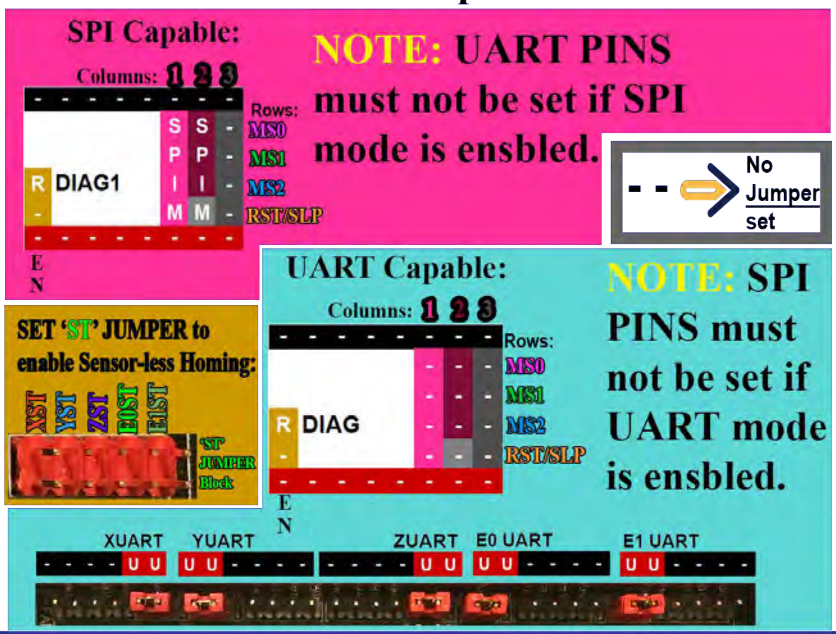
Driver Socket Representation: P P ⇒ Meaning: **set Jumper between column 1 and column 2 on the MS1 row**

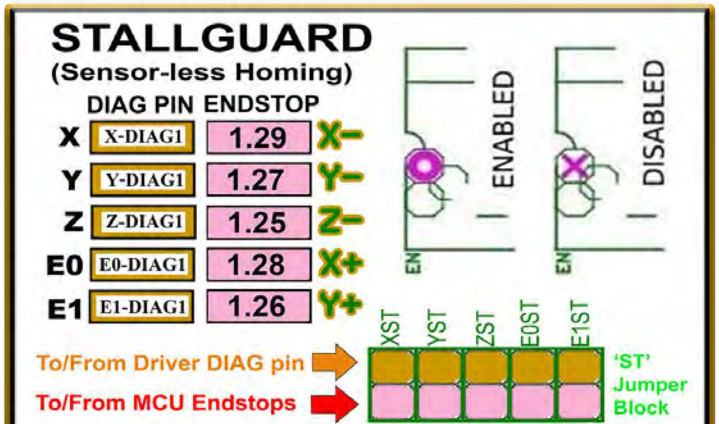
Driver Socket Representation: I I ⇒ Meaning: **set Jumper between column 1 and column 2 on the MS2 row**

Driver Socket Representation: M M ⇒ Meaning: **set Jumper between column 1 and column 2 on the RST/SLP row**

Information on Sensor-less Homing

Driver Socket Representation





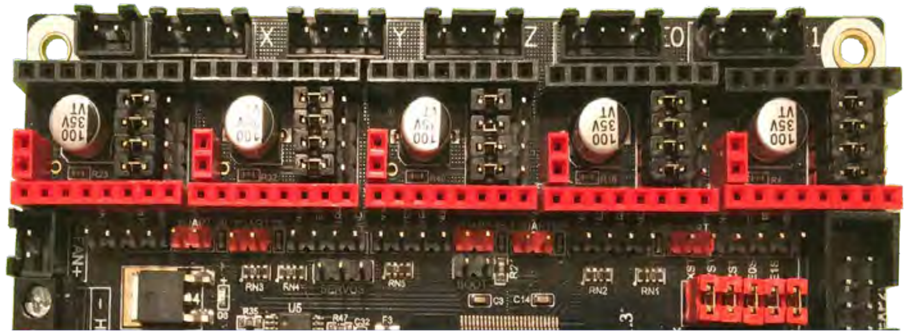
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

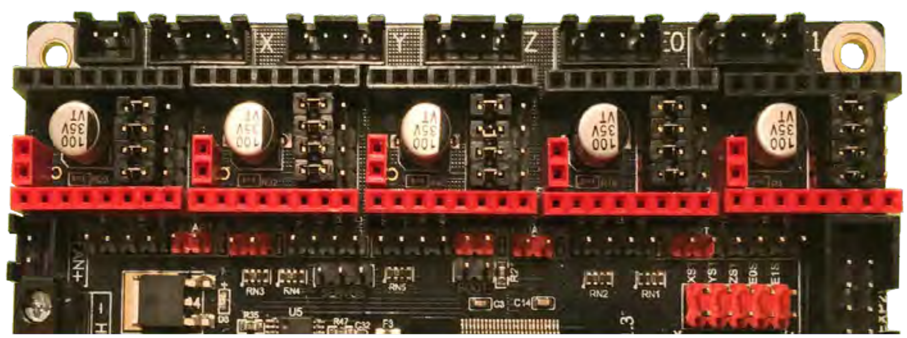
Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161
UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are SET for ALL Axes:



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are REMOVED for ALL Axes:



SPI Mode

BIQU TMC5160 V1.2

SPI Mode

Driver Socket Representation of 'ST' JUMPER Block

SPI Capable — Columns: 1 2 3

	1	2	3	Rows:
	S	S	-	MS0
	P	P	-	MS1
R DIAG1	I	I	-	MS2
-	M	M	-	RST/SLP

E N

UART Capable — Columns: 4 5 6

	4	5	6	Rows:
	-	-	-	MS0
	-	-	-	MS1
R DIAG	-	-	-	MS2
-	-	-	-	RST/SLP

E N

XUART: - - - - U U
YUART: U U - - - -
ZUART: - - - - U U
E0 UART: U U - - - -
E1 UART: U U - - - -

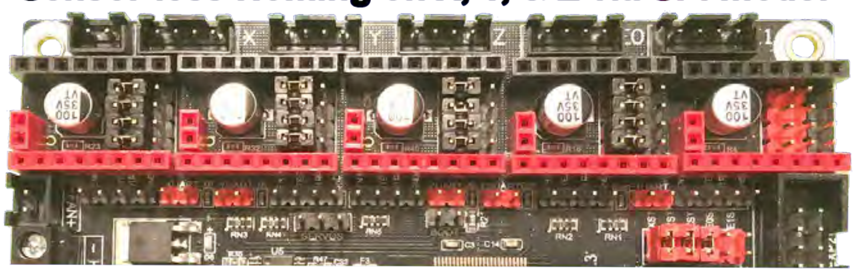
Columns: 7 8 9 10 11

Rows:	7 (XST) X-DIAG1 X-	8 (YST) Y-DIAG1 Y-	9 (ZST) Z-DIAG1 Z-	10 (E0ST) E0-DIAG1 X+	11 (E1ST) E1-DIAG1 Y+
1	L	H	O	-	-
2	L	H	O	-	-

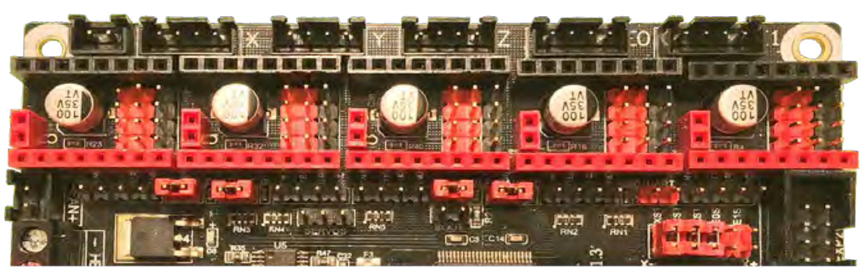
Columns: 7 8 9 10 11

Rows:	7 (XST) X-DIAG1 X-	8 (YST) Y-DIAG1 Y-	9 (ZST) Z-DIAG1 Z-	10 (E0ST) E0-DIAG1 X+	11 (E1ST) E1-DIAG1 Y+
1	L	H	O	N	G
2	L	H	O	N	G

Sensor-less Homing on X, Y, & Z via SPI mode:



Sensor-less Homing on X, Y, & Z via UART mode:



Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

- L L → X MIN Endstop to enable Sensor-less Homing
- H H → Y MIN Endstop to enable Sensor-less Homing
- O O → Z MIN Endstop to enable Sensor-less Homing
- N N → X MAX Endstop to enable Sensor-less Homing
- G G → Y MAX Endstop to enable Sensor-less Homing
- \- - → No Jumper Set

Driver Socket Representation:	Meaning:
L L	set Jumper between row 1 and row 2 in column 7
H H	set Jumper between row 1 and row 2 in column 8
O O	set Jumper between row 1 and row 2 in column 9
N N	set Jumper between row 1 and row 2 in column 10
G G	set Jumper between row 1 and row 2 in column 11

SPI Mode

SPI Mode

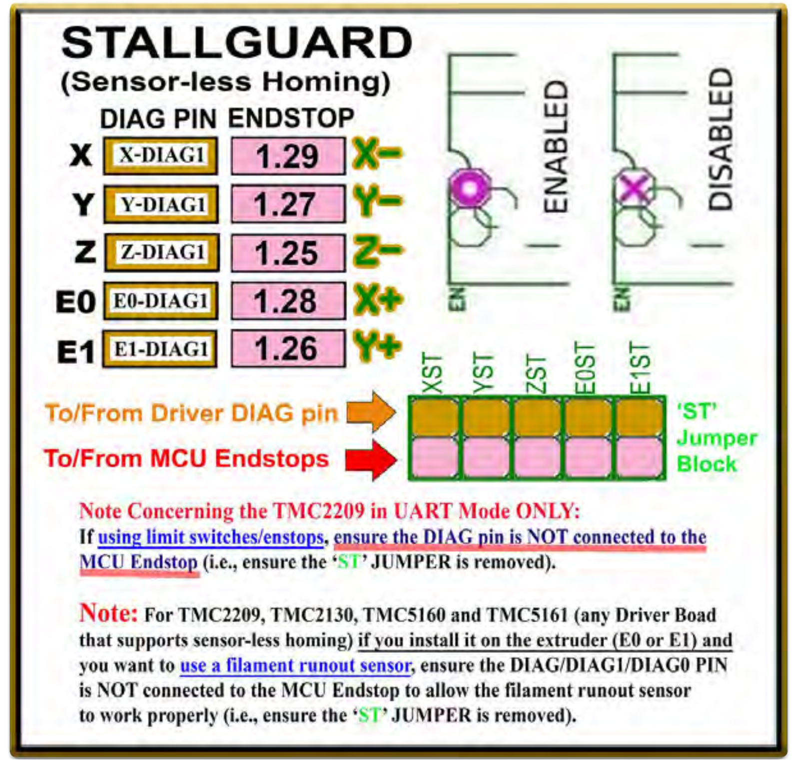
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capabiltiy of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).

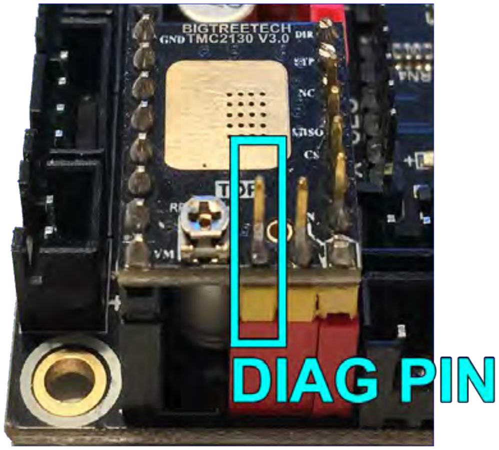


If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

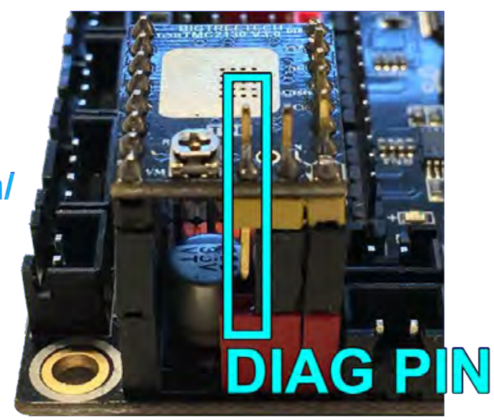
- **Extruder Axis (E0 or E1)**
- **Z Axis, if a BLTouch is used to Home instead of a physical endstop**

To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.



To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

Link to stackable header pins: https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P



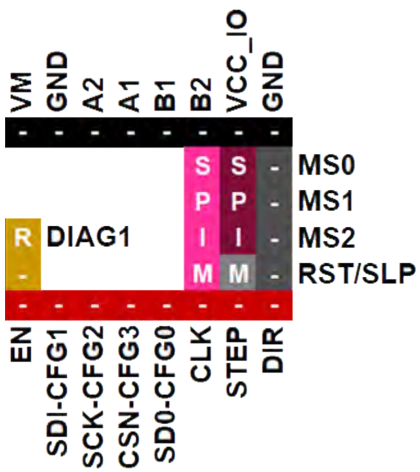
SPI Mode

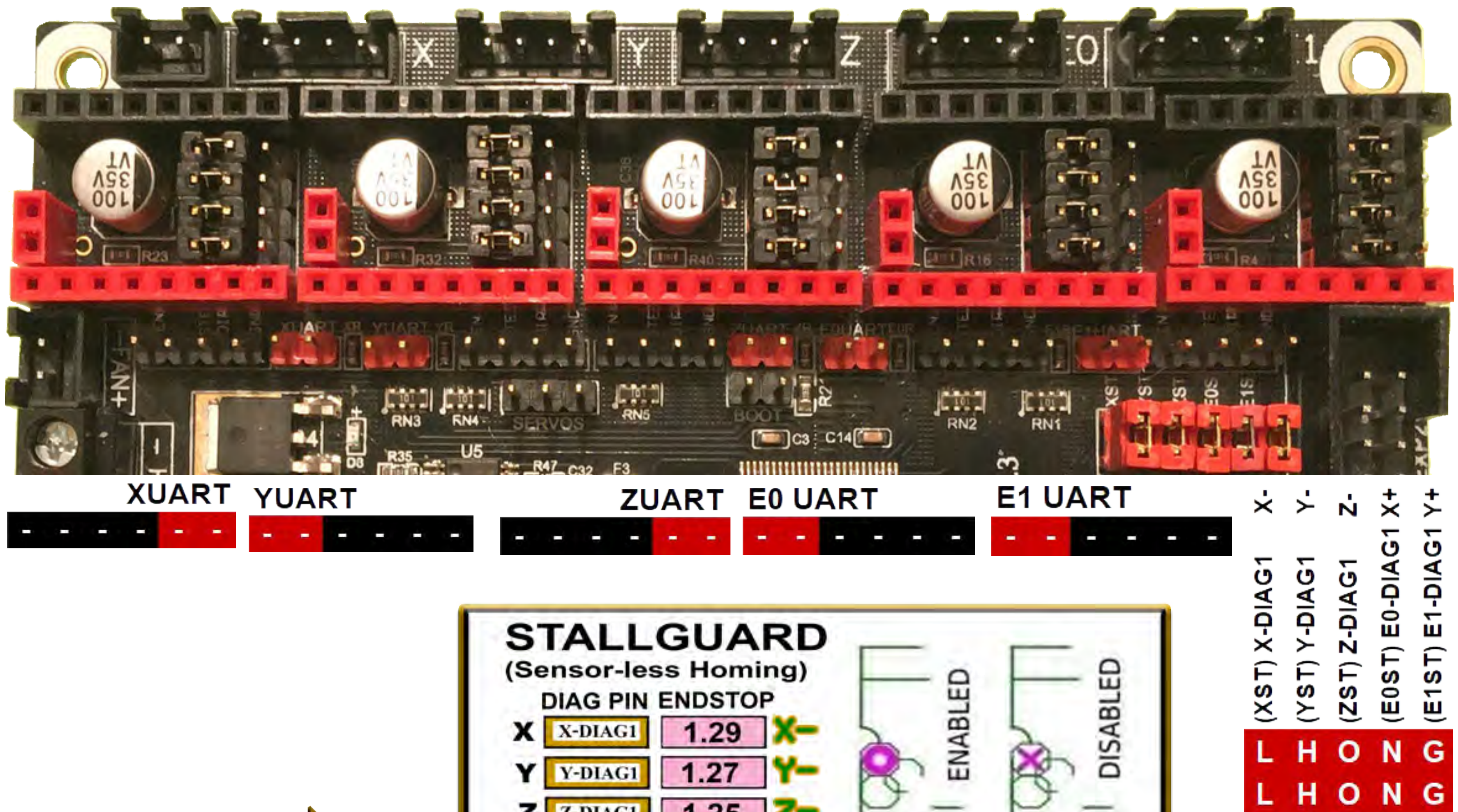
BIQU TMC5160 V1.2

SPI Mode

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

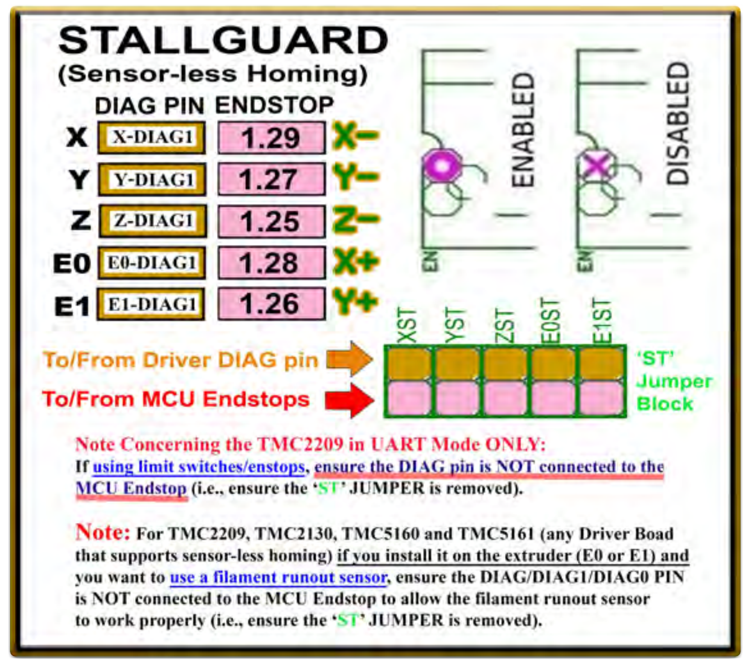
SPI





See Appendix D for legend

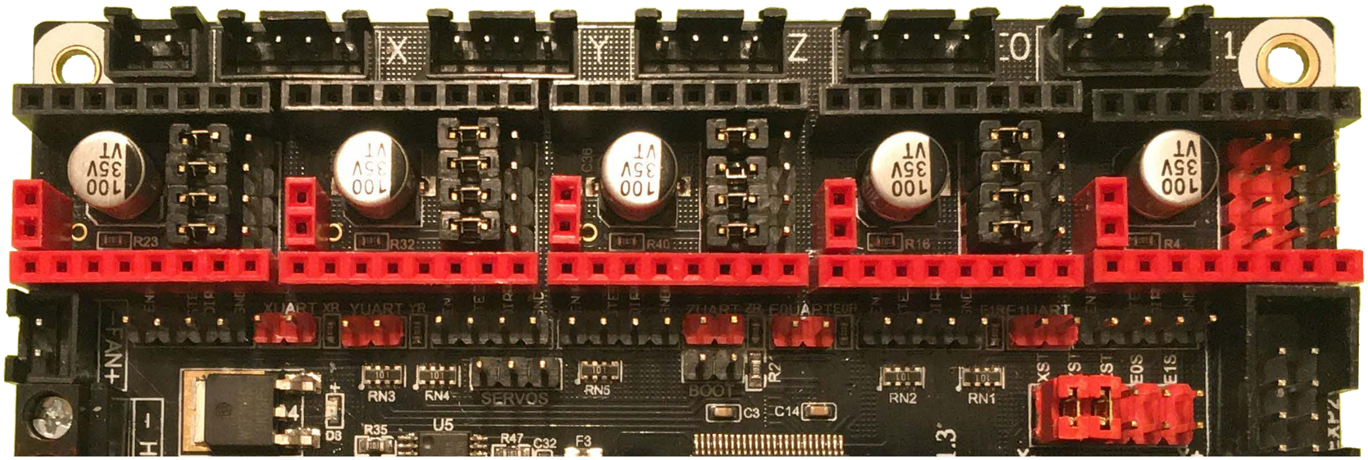
Note: TMC5160 has sensor-less homing capability



Examples of Different SPI Configurations Part 1

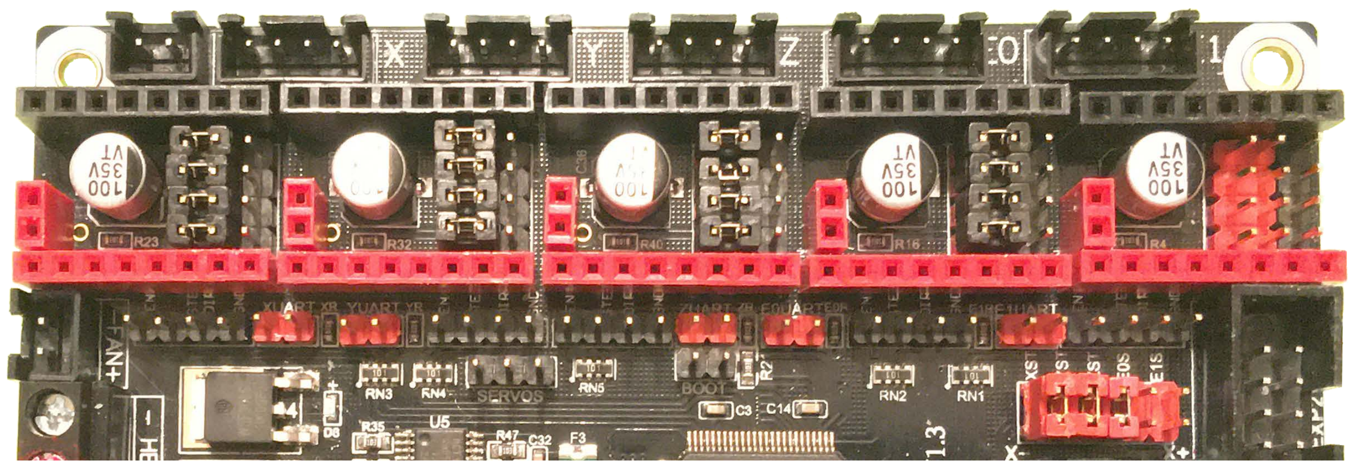
X, Y, Z and E0 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z and E0 axes configured for SPI mode.

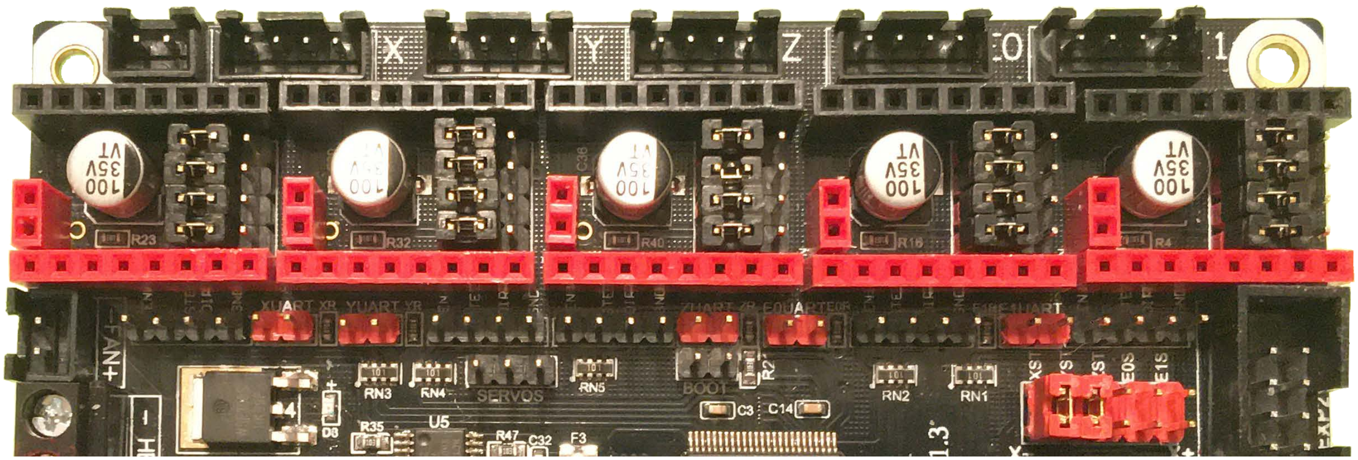
Sensor-less homing for X, Y and Z axes.



Examples of Different SPI Configurations Part 2

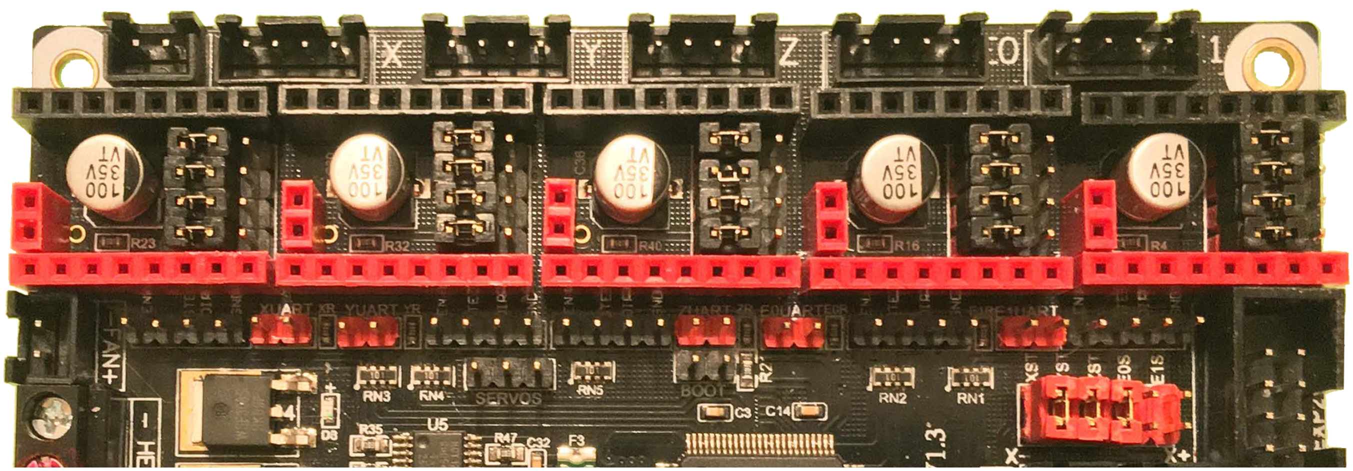
X, Y, Z, E0, and E1 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z, E0 and E1 axes configured for SPI mode.

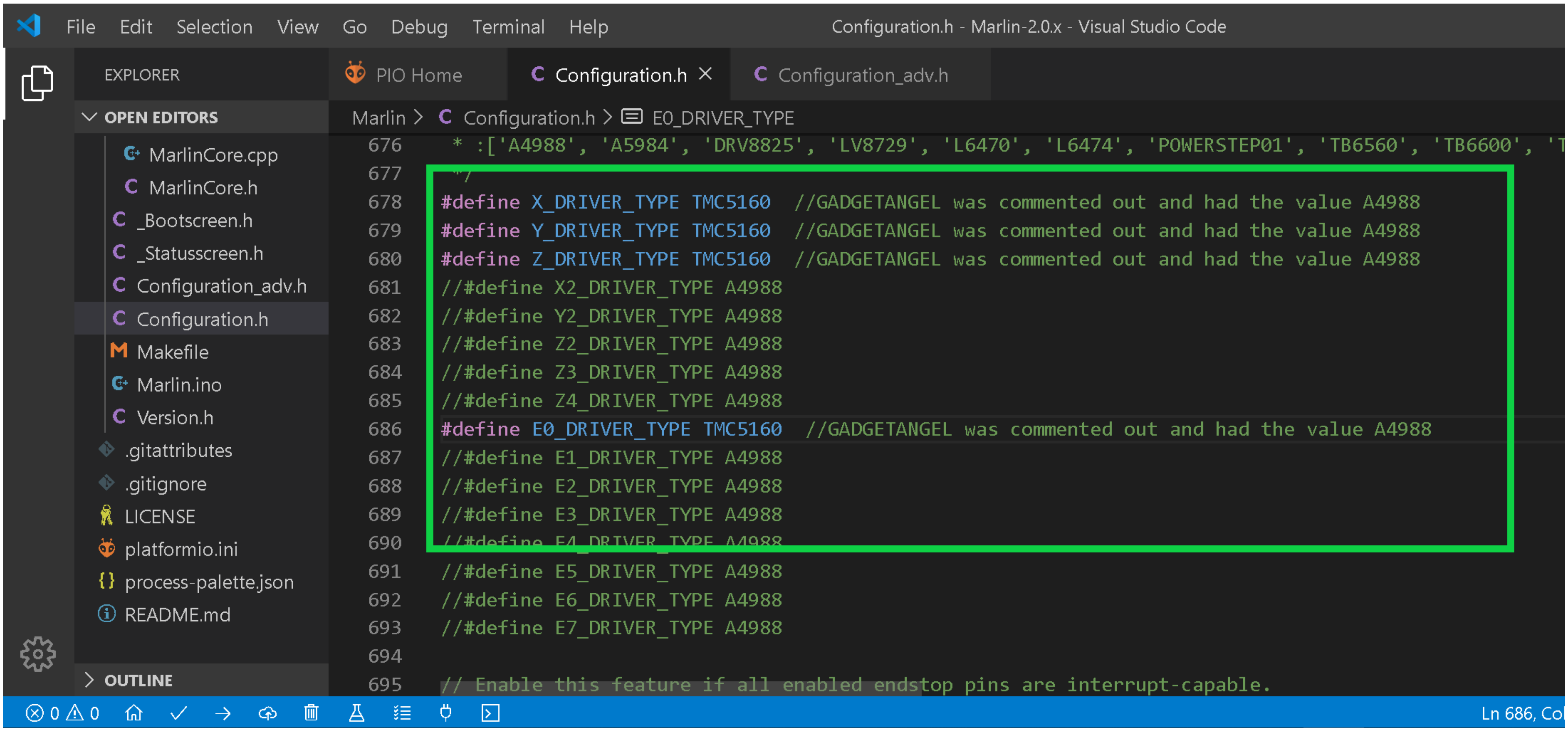
Sensor-less homing for X, Y and Z axes.



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.

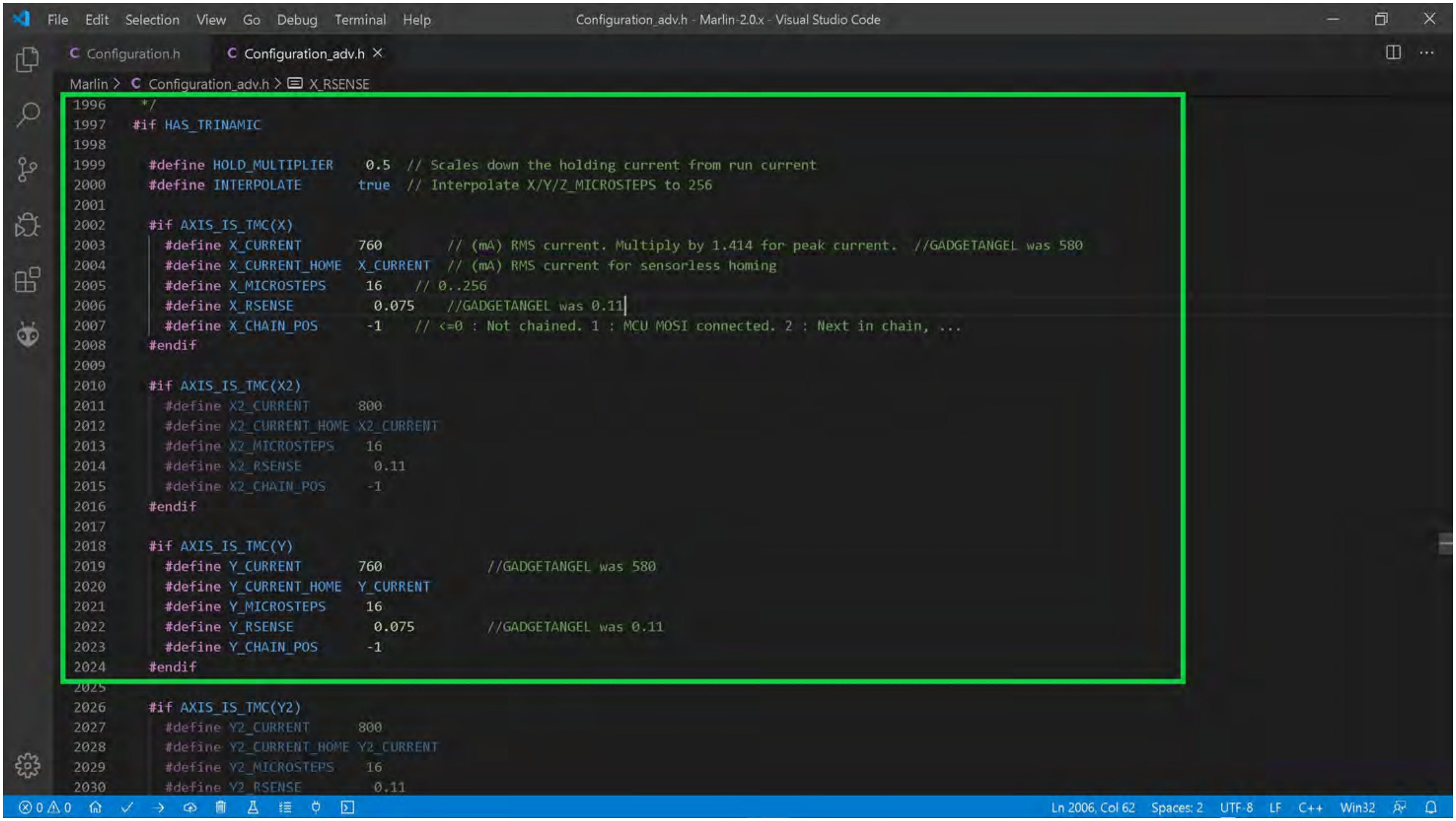
- Change the stepper motor drivers so that Marlin knows you are using TMC5160 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5160 drivers in SPI mode. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

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

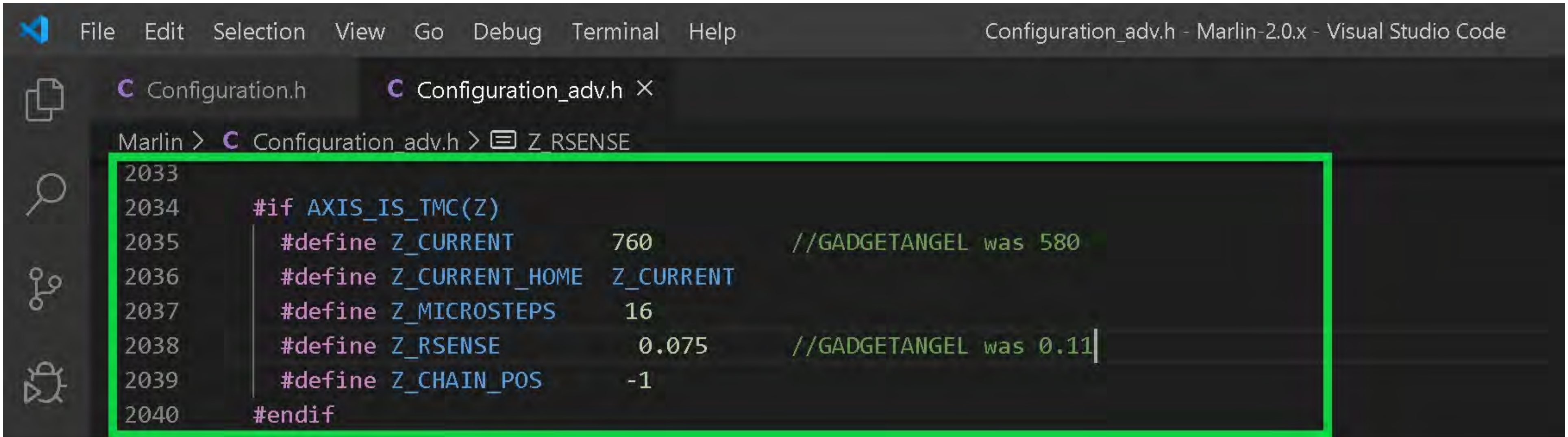
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5160 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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".

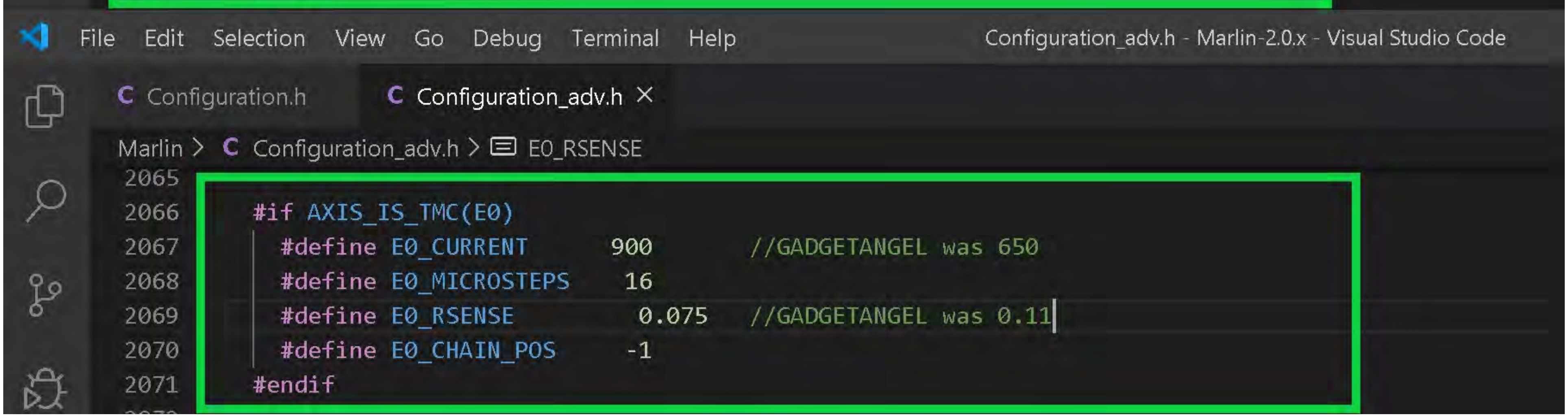


- Go to the next page.

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".

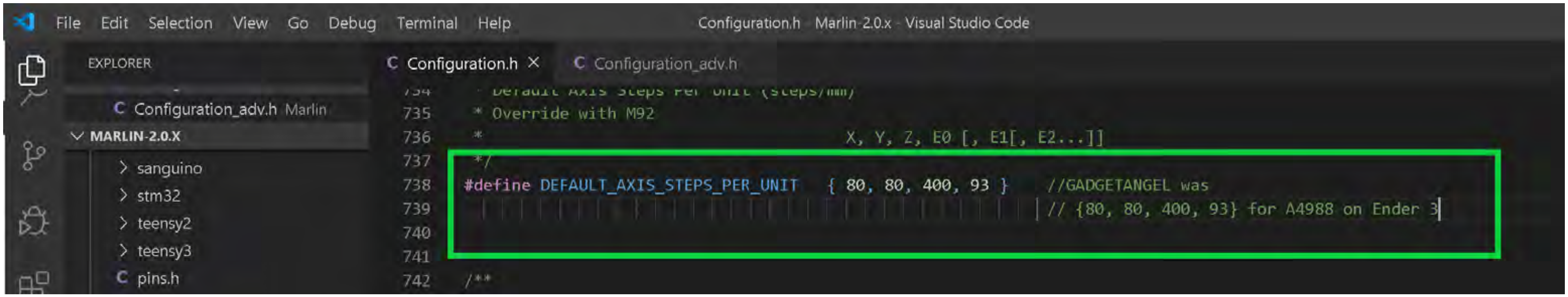




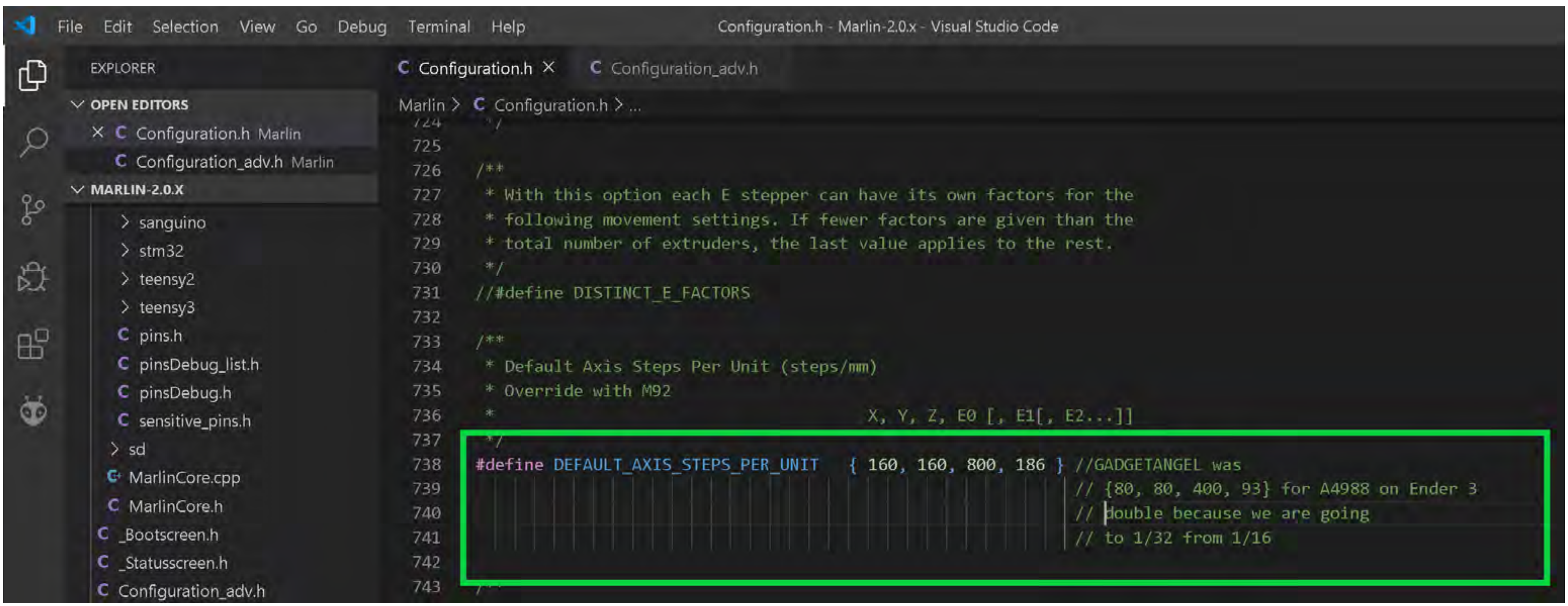
- Go to the next page.

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



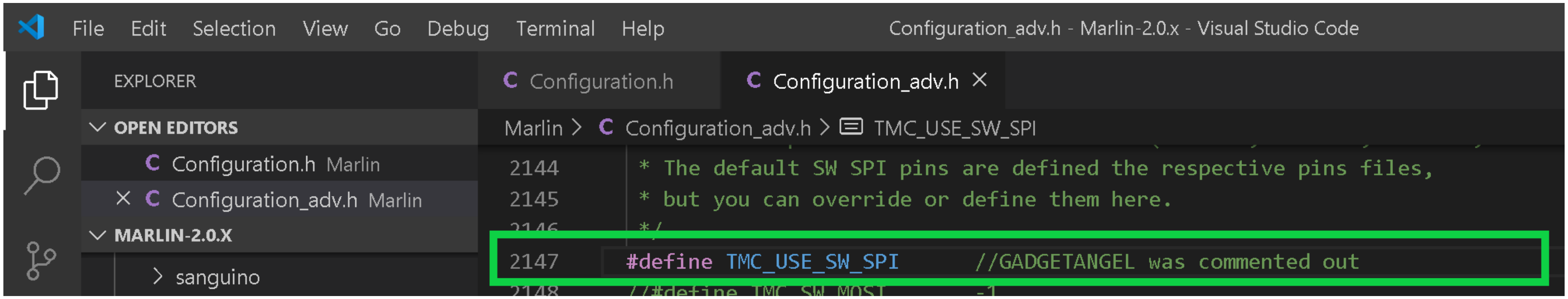
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



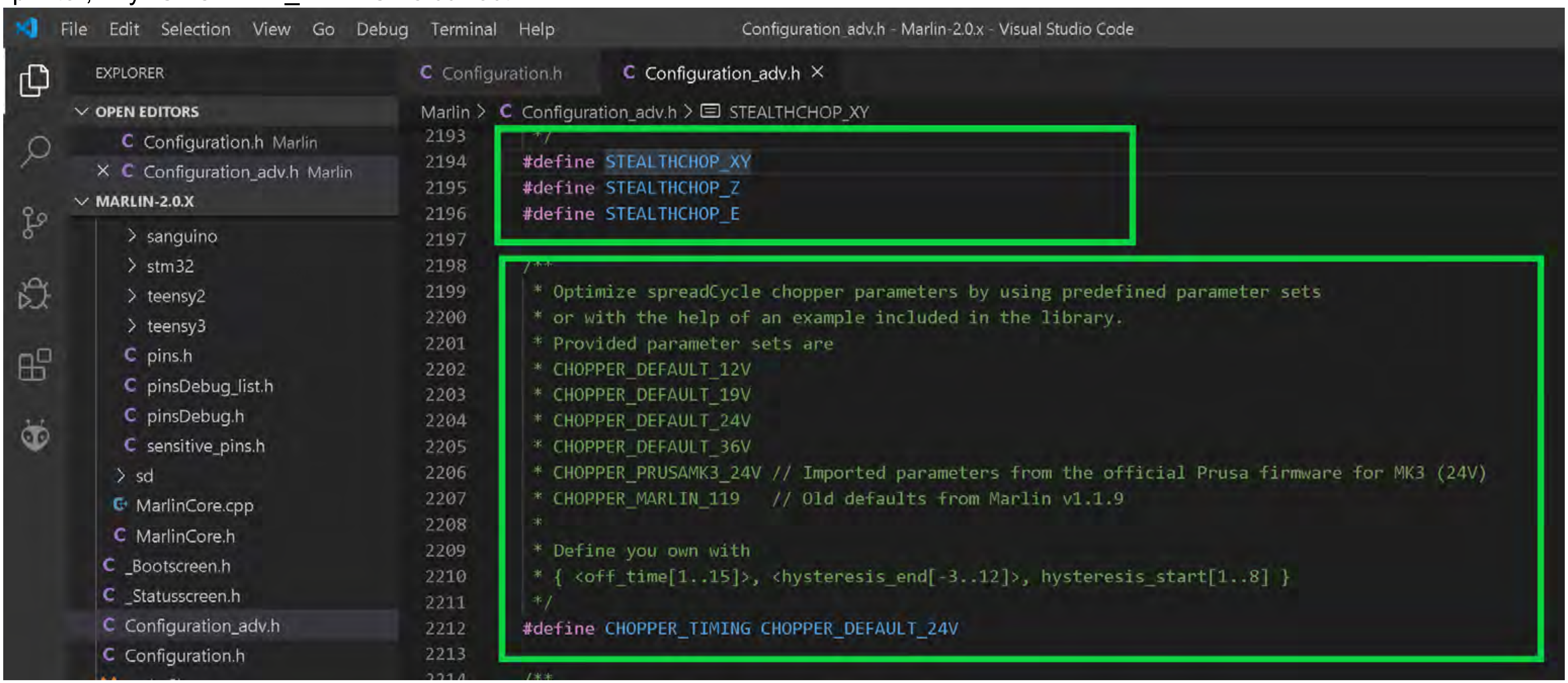
- Go to the next page.

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 V1.3 pins file depends on this variable to define its SPI pins



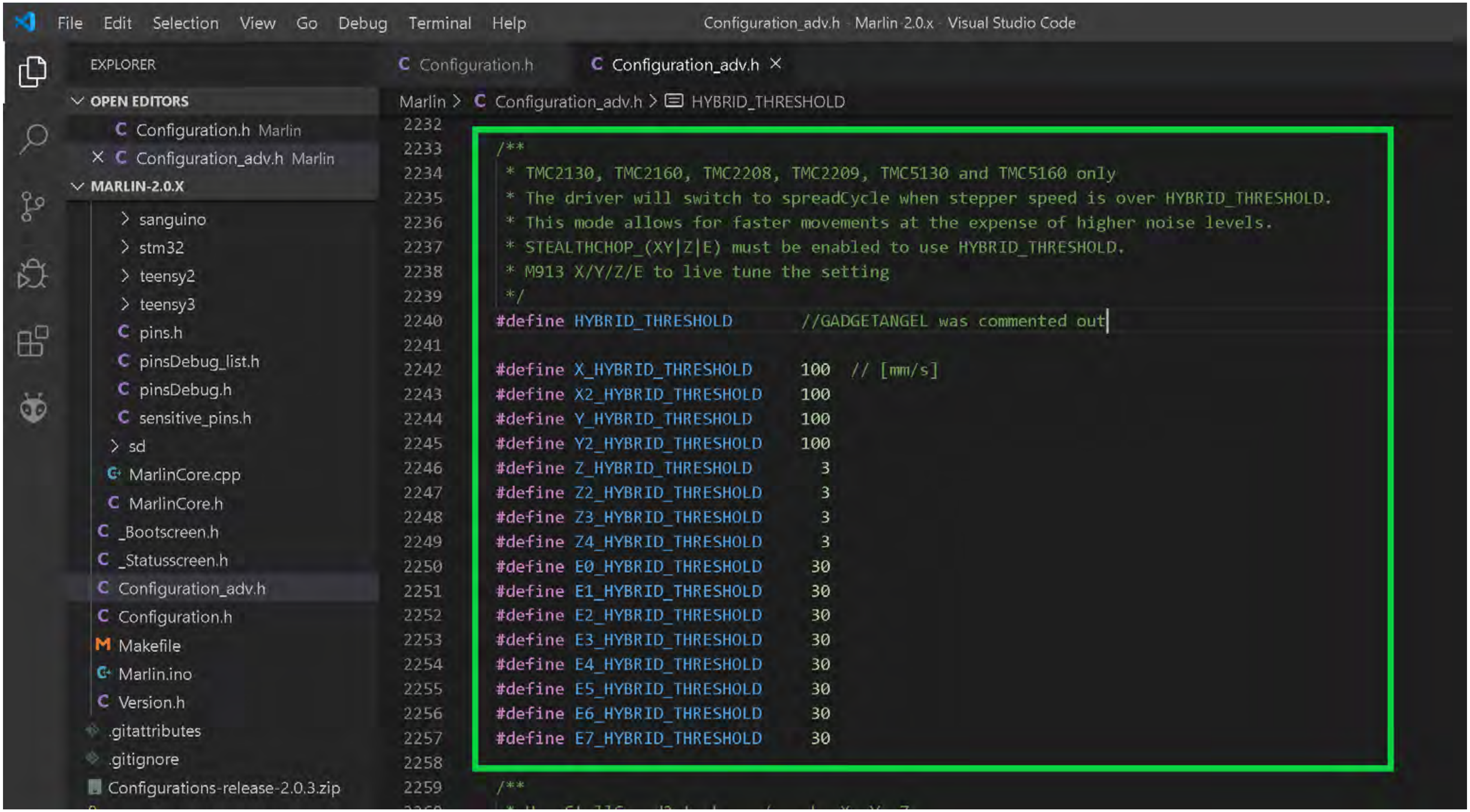
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

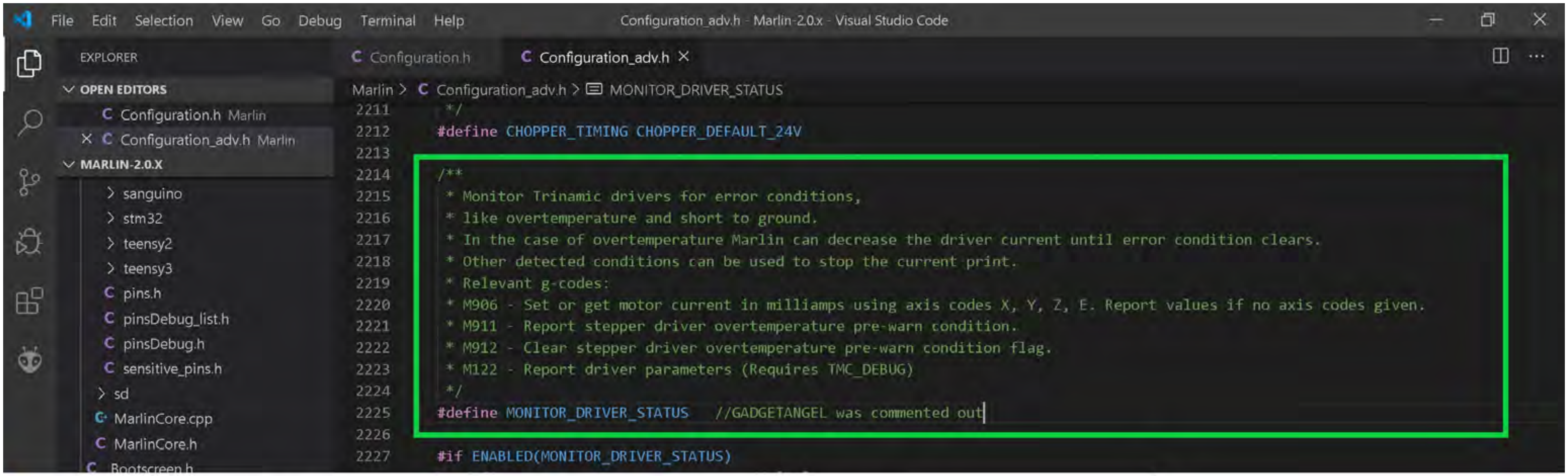
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

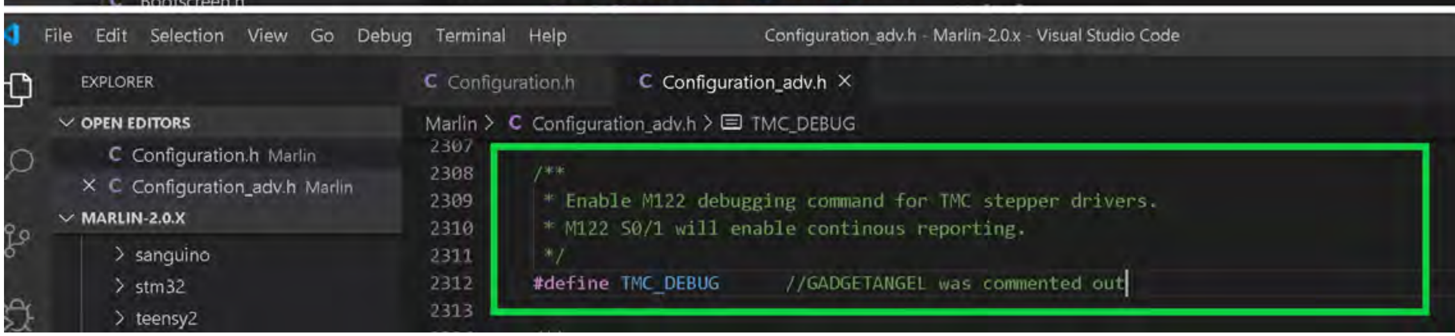


- Go to the next page.

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.**

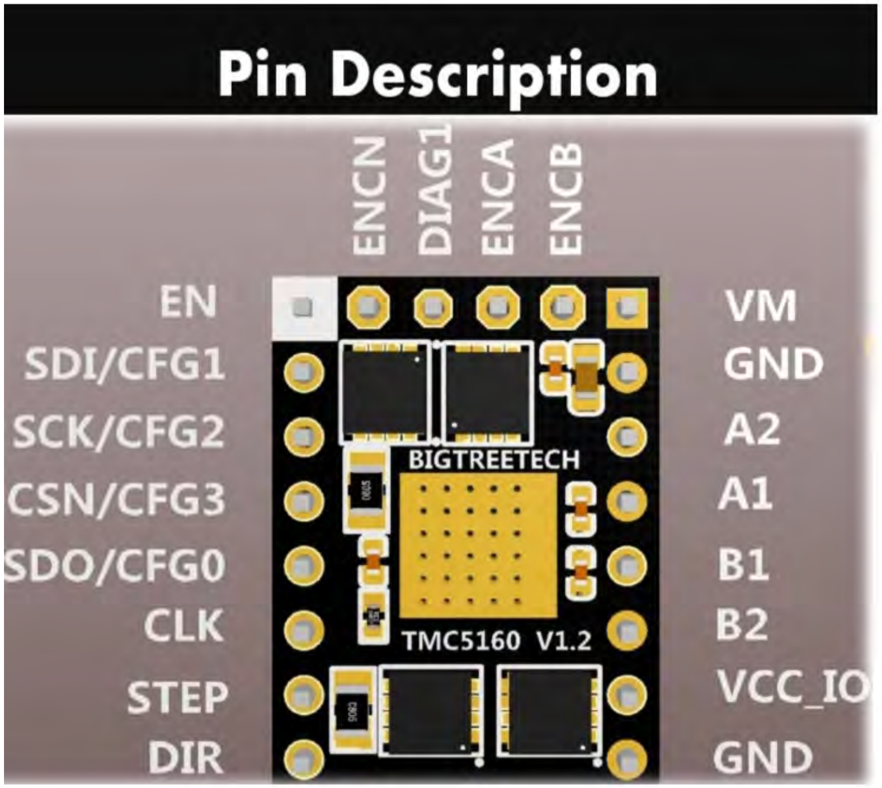


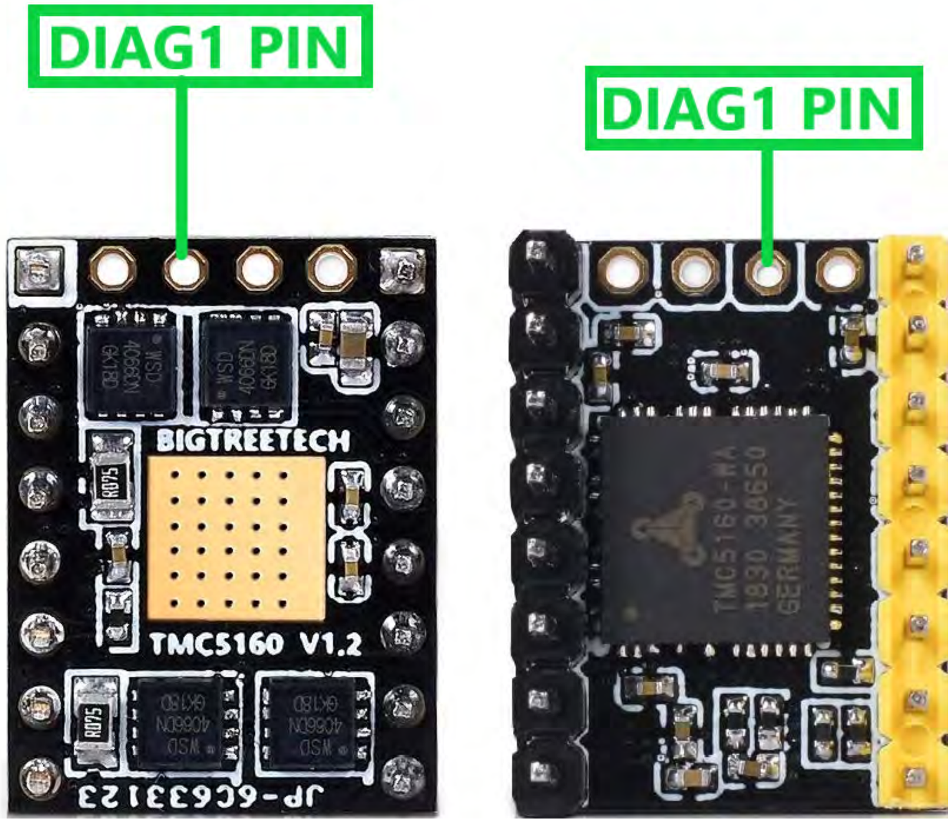


- Go to the next page.

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.



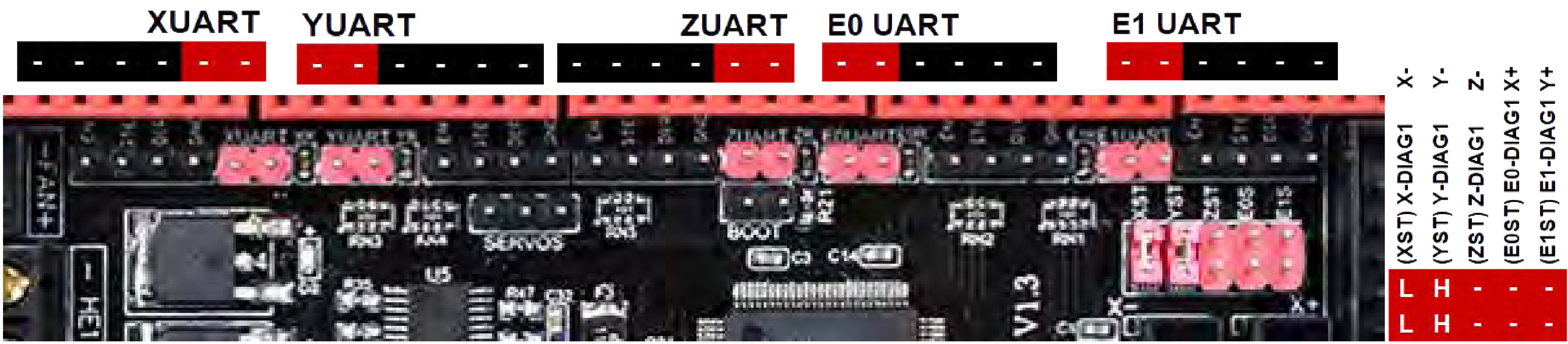


- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5160 into the SKR V1.3 board the DIAG1 pin will be seated in the SKR V1.3 board. Then I can use the XST jumper to connect the DIAG1 pin to the X endstop. I can then, also, use the YST jumper to connect the DIAG1 pin to the Y endstop. Connecting the DIAG1 pins to the endstops enables sensor-less homing to work.

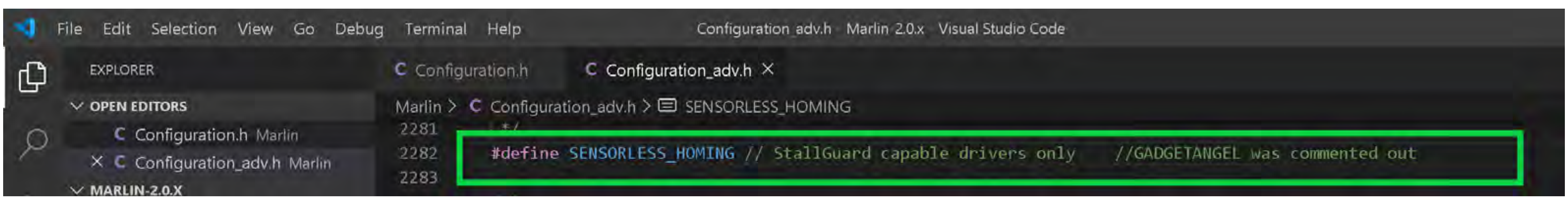
- Go to the next page.

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 V1.3 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 TMC5160 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC560 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC5160 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers that are in place are the "L" and "H" jumpers.



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



- 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 ×

Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY

```
#if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
  // TMC2209: 0...255. TMC2130: -64...63
  #define X_STALL_SENSITIVITY 15                         //GADGETANGEL was 8, needed for sensorless homing
  #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
  #define Y_STALL_SENSITIVITY 15                         //GADGETANGEL was 8, needed for sensorless homing
  //#define Z_STALL_SENSITIVITY  8
  //#define SPI_ENDSTOPS              // TMC2130 only
  //#define IMPROVE_HOMING_RELIABILITY
#endif

/**
```

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

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

C Configuration.h C Configuration_adv.h ×

Marlin > C Configuration_adv.h > X_HOME_BUMP_MM

```
// Homing hits each endstop, retracts by these distances, then does a slower bump.
#define X_HOME_BUMP_MM 0            //GADGETANGEL was 5
#define Y_HOME_BUMP_MM 0            //GADGETANGEL was 5
#define Z_HOME_BUMP_MM 2
#define HOMING_BUMP_DIVISOR { 2, 2, 4 }  // Re-Bump Speed Divisor (Divides the Homing Feedrate)
#define QUICK_HOME                       // If homing includes X and Y, do a diagonal move initially
//#define HOMING_BACKOFF_MM { 2, 2, 2 }  // (mm) Move away from the endstops after homing

// When G28 is called, this option will make Y home before X
//#define HOME_Y_BEFORE_X
```

- 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 × | Configuration_adv.h

Marlin > Configuration.h > ENDSTOPPULLUPS

```
// Enable pullup for all endstops to prevent a floating state
#define ENDSTOPPULLUPS          //GADGETANGEL this should always be uncommented for an Ender 3
#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  //#define ENDSTOPPULLUP_XMAX
  //#define ENDSTOPPULLUP_YMAX
  //#define ENDSTOPPULLUP_ZMAX
  //#define ENDSTOPPULLUP_XMIN
  //#define ENDSTOPPULLUP_YMIN
  //#define ENDSTOPPULLUP_ZMIN
  //#define ENDSTOPPULLUP_ZMIN_PROBE
#endif
```

- 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 × | Configuration_adv.h

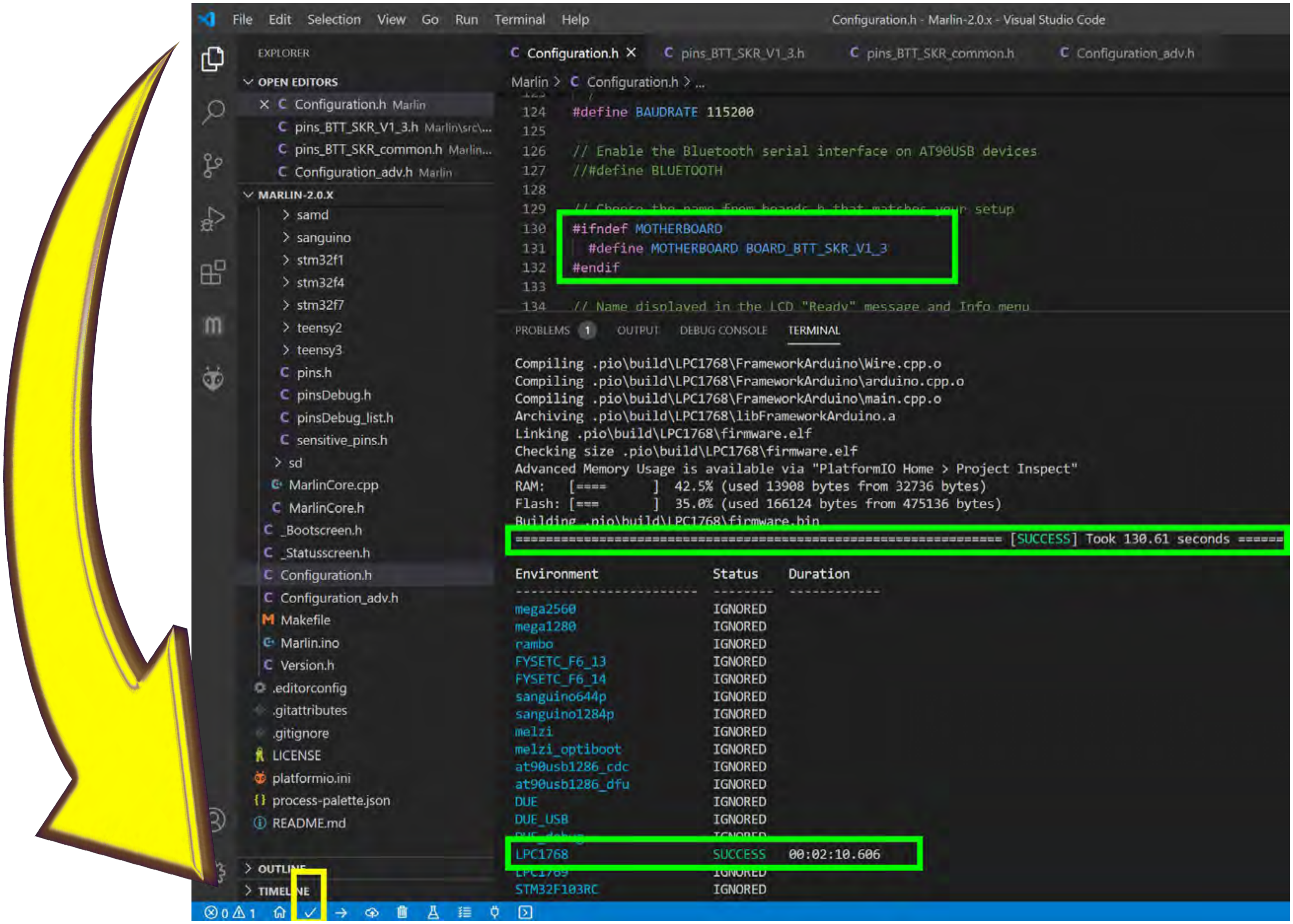
Marlin > Configuration.h > X_MIN_ENDSTOP_INVERTING

```
// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
#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 TMC51
#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 TMC51
#define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
```

- Go to the next page.

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

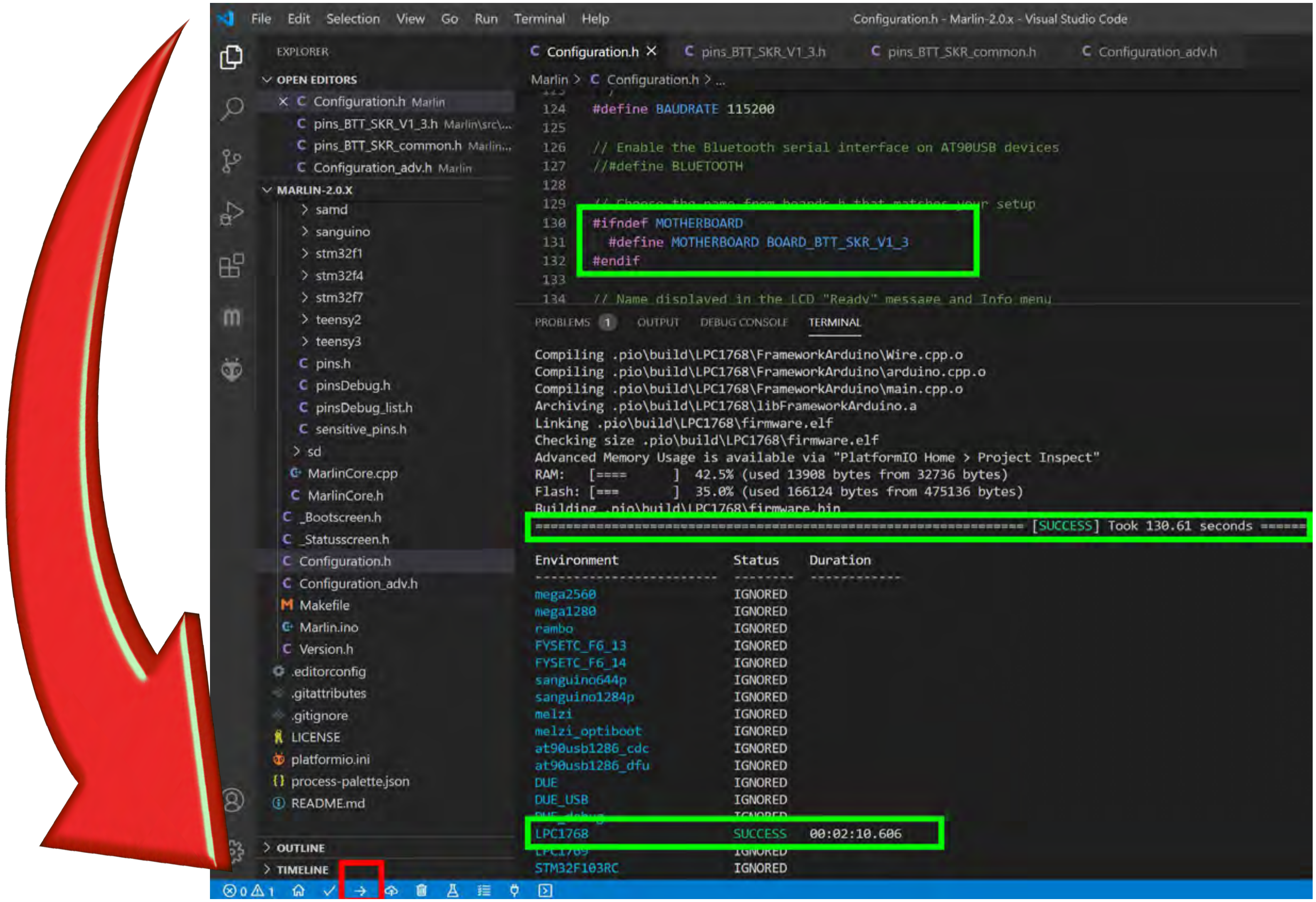
- The end of Marlin setup for BIQU TMC5160 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



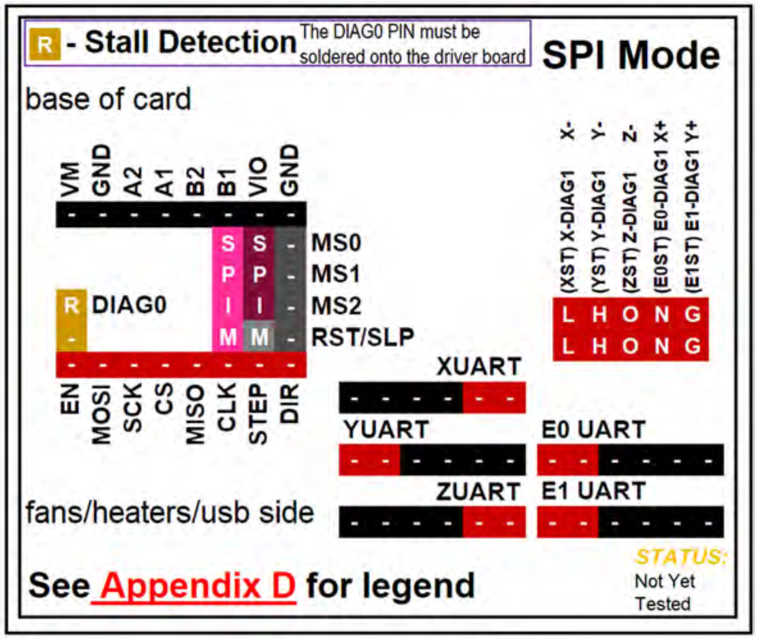
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.



BIQU TMC5161 V1.0

SPI Mode

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

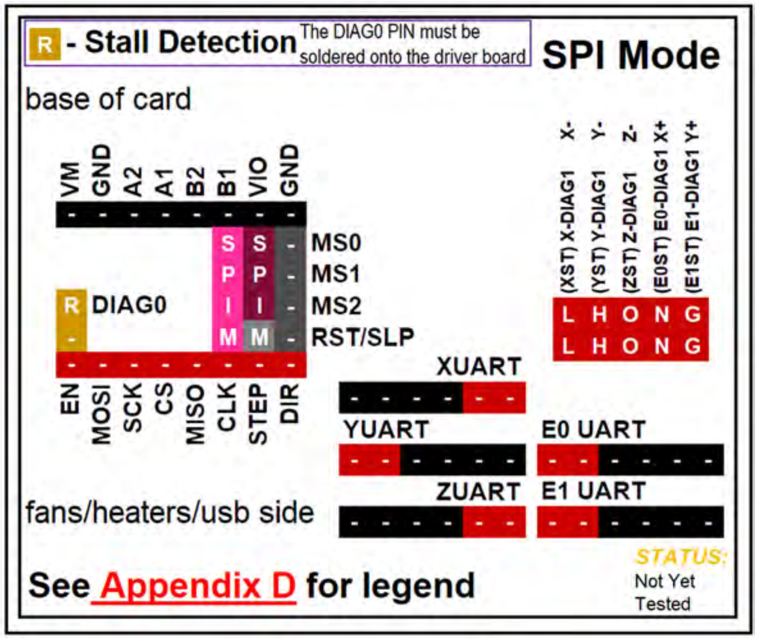
See the next page for further information.

Driver Chip	Steps are set inside of your Firmware
BIQU® TMC5161 SPI Mode Maximum 256 Subdivision 40V DC 5.2A (peak)	

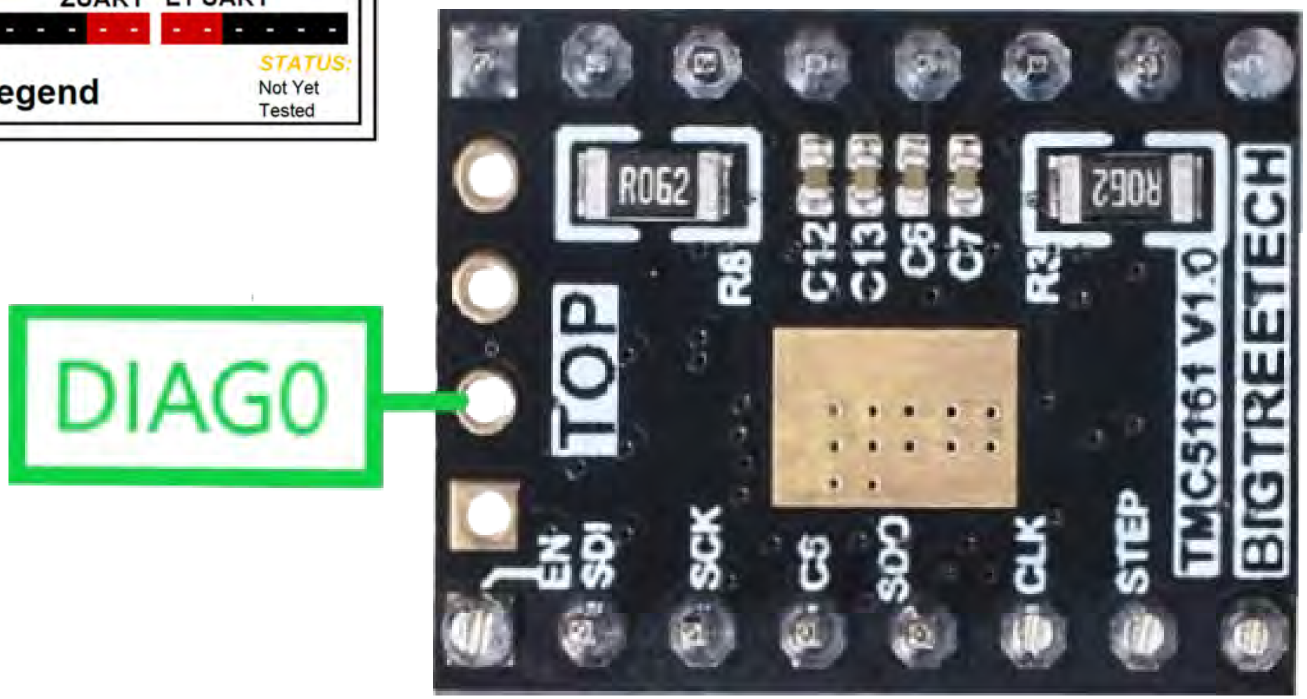
Driving Current Calculation Formula		
R_S(Typical Sense Resistor)=0.062Ω	I_{MAX}=5.2419 See Appendix B #9. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX}*0.90=4.718$	Current Limit is set by the current sense resistors (R_s). Use 50% - 90% of I_{MAX}.

BIQU TMC5161 V1.0

SPI Mode



Note: The TMC5161 V1.0 by default comes in SPI mode. The BIQU TMC5161 does NOT come with a POT or "V$_{ref}$ Test point" location because the IRMS is set inside of the Firmware.



NOTE: BIQU TMC5161 has the ability to do sensor-less homing. By default the DIAG0 pin is NOT soldered onto the driver board. Therefore, for any axis you want sensor-less homing enabled, YOU WILL HAVE to solder on the DIAG0 pin.

SPI Mode

BIQU 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", "E0_RSENSE" and/or "E1_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", "E0_CURRENT" and/or "E1_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", "E0_CURRENT", and/or "E1_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

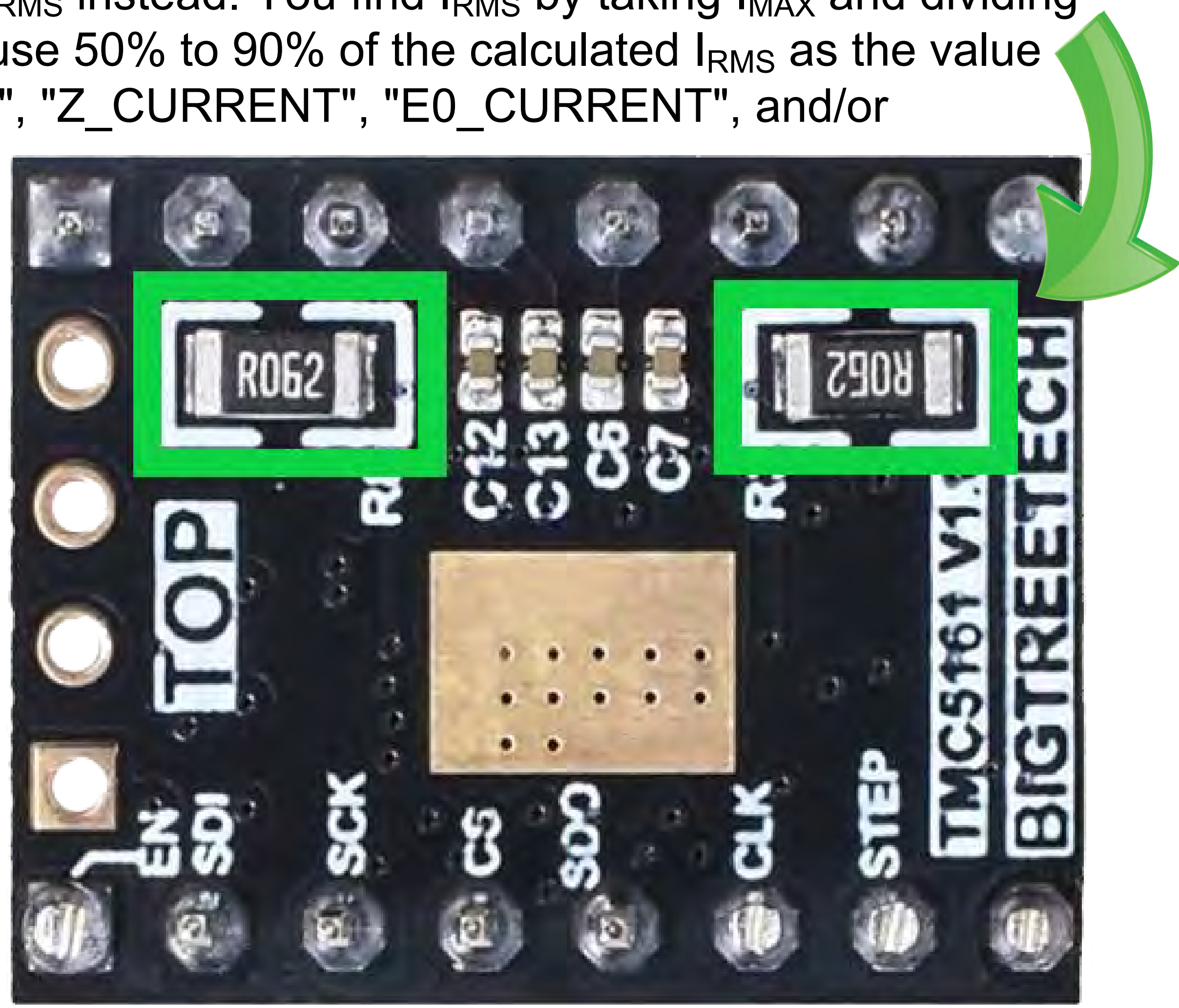
R_s = R100 is 0.1 Ohms

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



BIQU TMC5161 V1.0

SPI Mode

SPI Mode

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	-	-
L	H	O	-	-

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
L	H	O	N	G
L	H	O	N	G

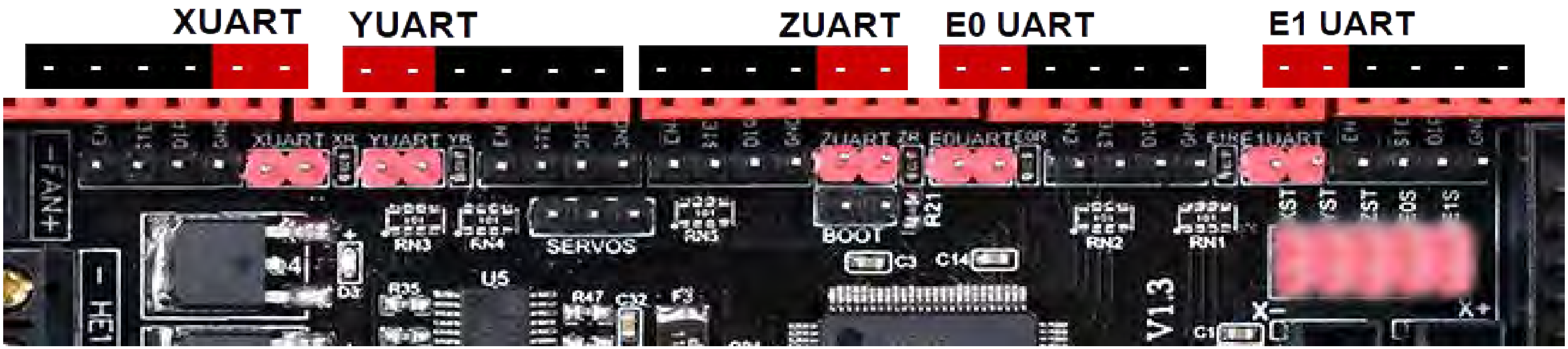
Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.

(XST) X-DIAG1 X-	(YST) Y-DIAG1 Y-	(ZST) Z-DIAG1 Z-	(E0ST) E0-DIAG1 X+	(E1ST) E1-DIAG1 Y+
-	-	-	-	-
-	-	-	-	-

Note: If sensor-less homing is not wanted ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".

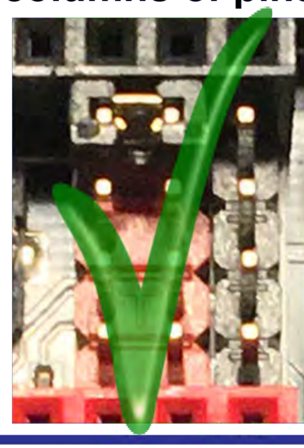
Note: Ensure that **ALL** the UART jumpers are **EMPTY**.



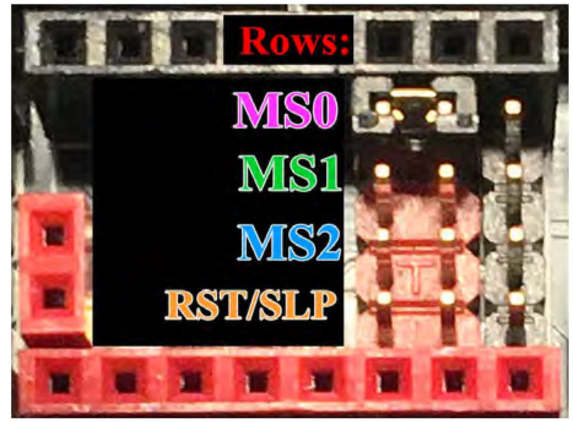
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

Example of the **WRONG** way to set the "S" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "S" jumper for SPI mode (left two columns of pins):



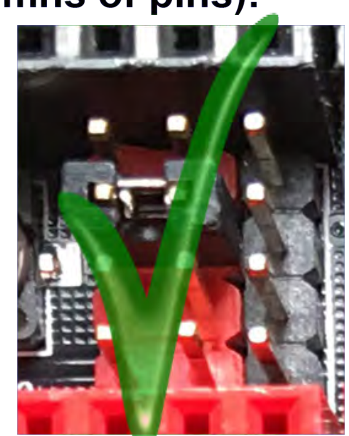
S S ⇒ Jumper set



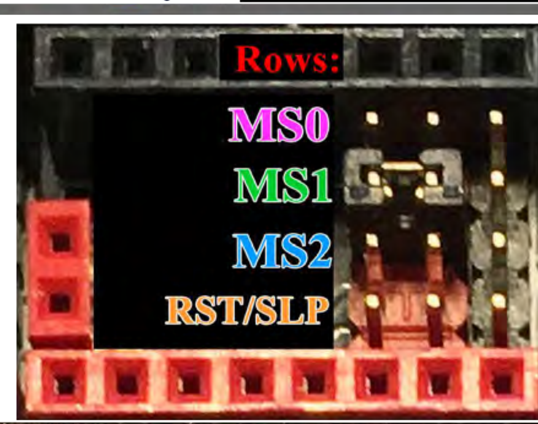
SET 'ST' JUMPER to enable
Sensor-less Homing

Example of the **WRONG** way to set the "P" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "P" jumper for SPI mode (left two columns of pins):



P P ⇒ Jumper set



SET 'ST' JUMPER to enable
Sensor-less Homing

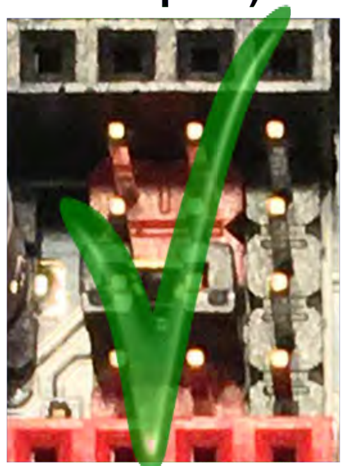
SPI Mode

BIQU TMC5161 V1.0

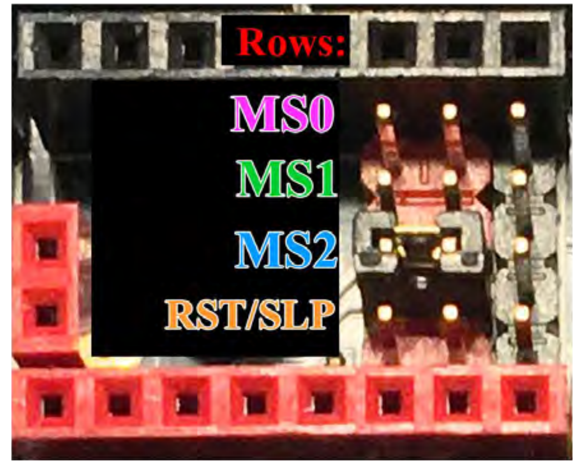
SPI Mode

Example of the **WRONG** way to set the "I" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "I" jumper for SPI mode (left two columns of pins):



I I → Jumper set

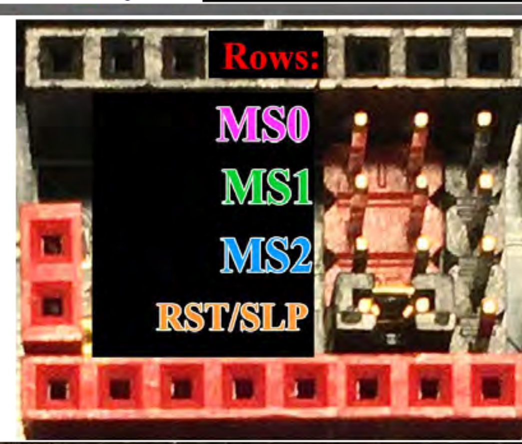


SET 'ST' JUMPER to enable Sensor-less Homing

Example of the **WRONG** way to set the "M" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "M" jumper for SPI mode (left two columns of pins):

M M → Jumper set

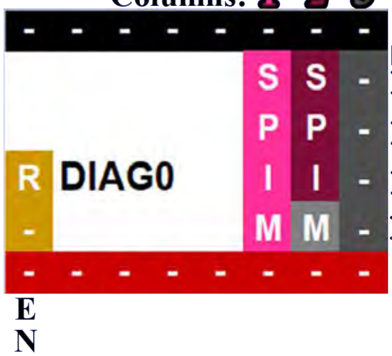


SET 'ST' JUMPER to enable Sensor-less Homing

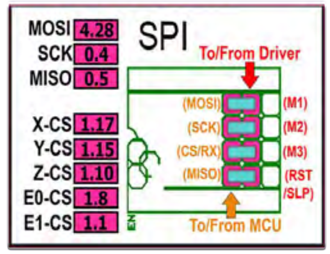
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

Driver Socket Representation

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP



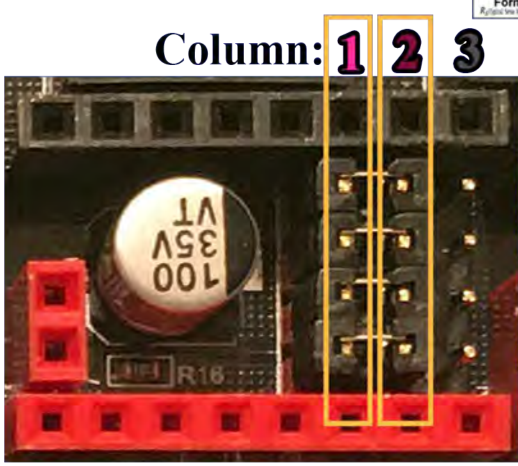
- - → No Jumper set

NOTE: UART PINS must not be set if SPI mode is ensbled.

SPIM:

Driver Chip Chart:

Column: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

S S
P P
I I
M M
→

Meaning:
Jumpers Set

Driver Socket Representation: S S →
Meaning: **set Jumper between column 1 and column 2 on the MS0 row**

Driver Socket Representation: P P →
Meaning: **set Jumper between column 1 and column 2 on the MS1 row**

Driver Socket Representation: I I →
Meaning: **set Jumper between column 1 and column 2 on the MS2 row**

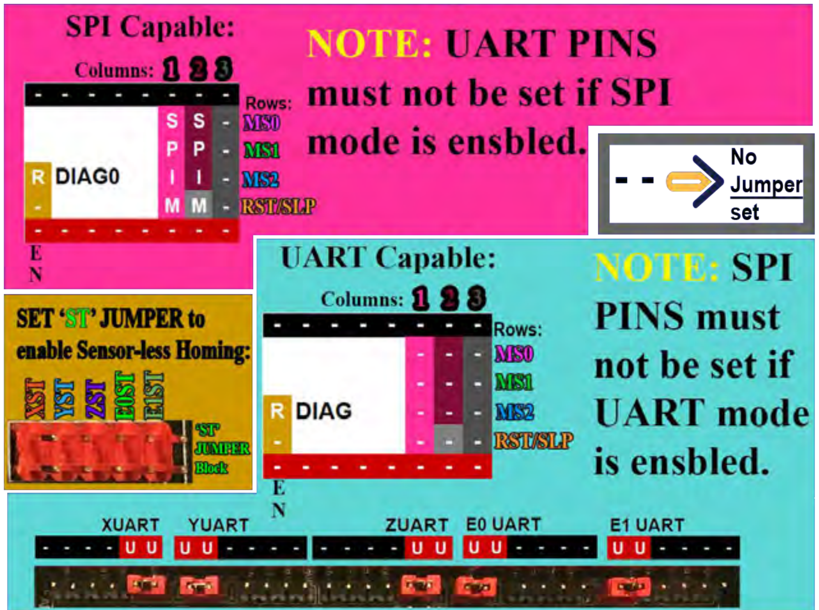
Driver Socket Representation: M M →
Meaning: **set Jumper between column 1 and column 2 on the RST/SLP row**

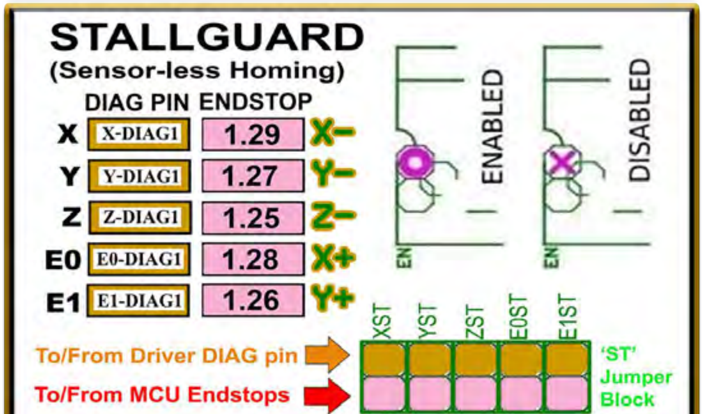
BIQU TMC5161 V1.0

SPI Mode

Information on Sensor-less Homing

Driver Socket Representation





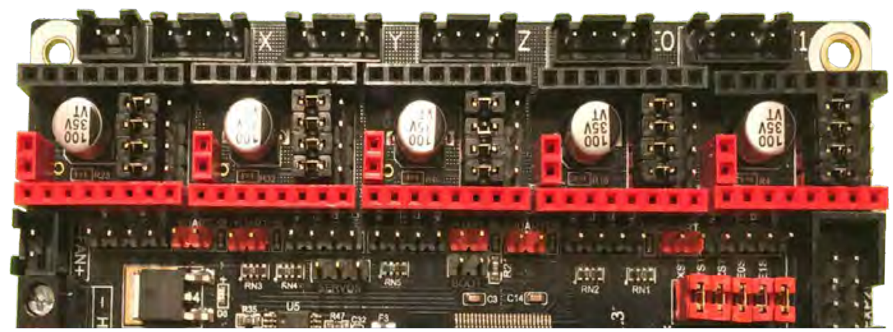
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

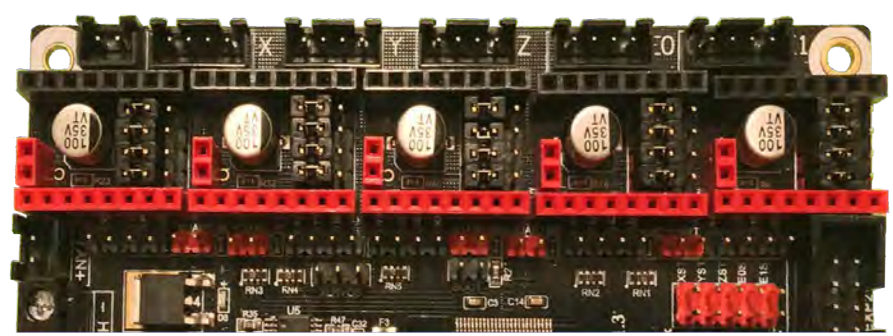
Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161
UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are SET for ALL Axes:



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are REMOVED for ALL Axes:

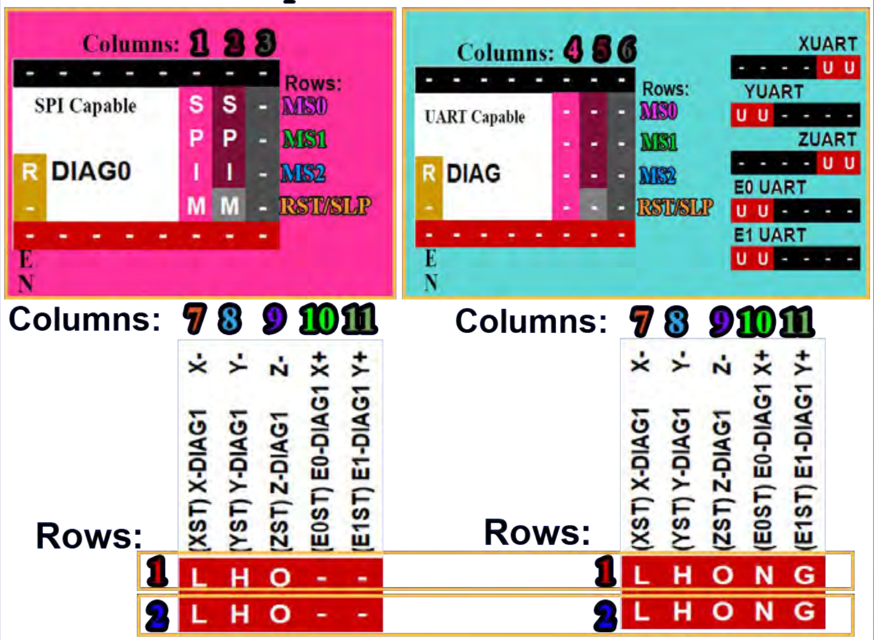


SPI Mode

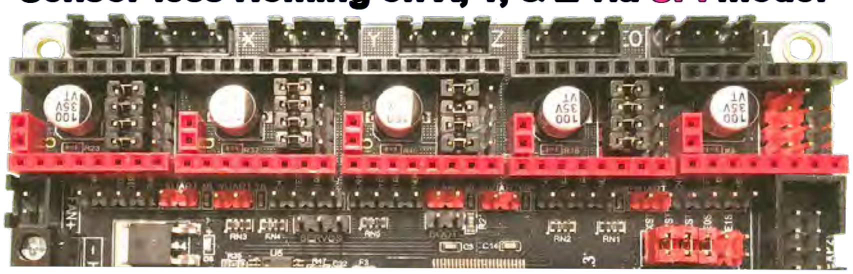
BIQU TMC5161 V1.0

SPI Mode

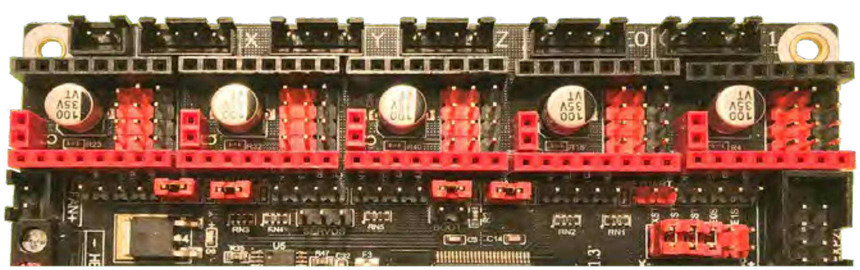
Driver Socket Representation of 'ST' JUMPER Block



Sensor-less Homing on X, Y, & Z via SPI mode:



Sensor-less Homing on X, Y, & Z via UART mode:



Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

- L L → X MIN Endstop to enable Sensor-less Homing
- H H → Y MIN Endstop to enable Sensor-less Homing
- O O → Z MIN Endstop to enable Sensor-less Homing
- N N → X MAX Endstop to enable Sensor-less Homing
- G G → Y MAX Endstop to enable Sensor-less Homing
- \- \- → No Jumper Set

Driver Socket Representation:	Meaning:
L L	set Jumper between row 1 and row 2 in column 7
H H	set Jumper between row 1 and row 2 in column 8
O O	set Jumper between row 1 and row 2 in column 9
N N	set Jumper between row 1 and row 2 in column 10
G G	set Jumper between row 1 and row 2 in column 11

SPI Mode

BIQU TMC5161 V1.0

SPI Mode

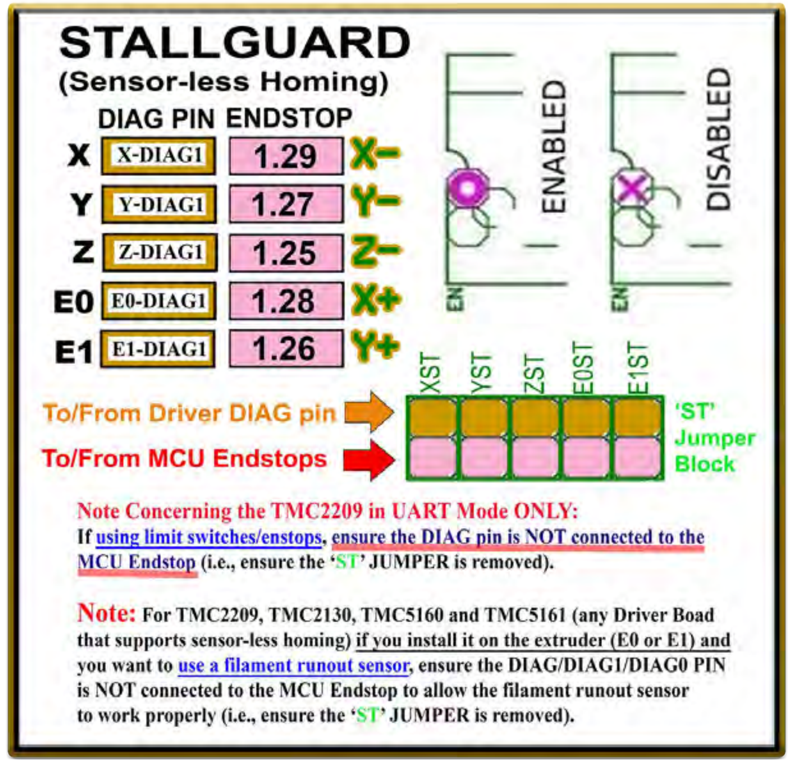
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capabilitiy of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

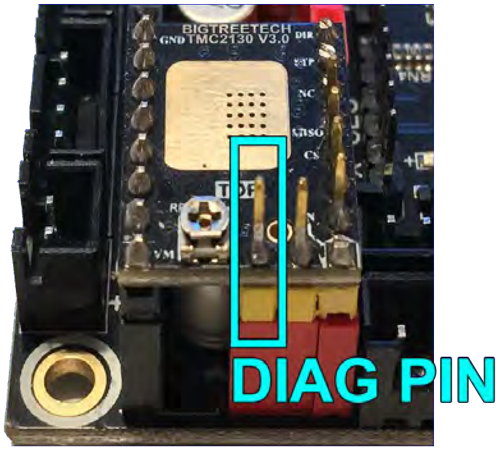
- **Extruder Axis (E0 or E1)**
- **Z Axis, if a BLTouch is used to Home instead of a physical endstop**

SPI Mode

BIQU TMC5161 V1.0

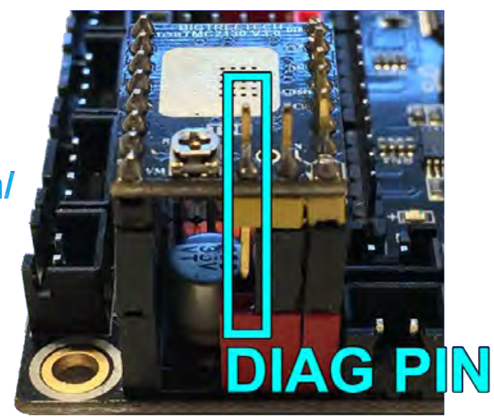
SPI Mode

To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.



To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

Link to stackable header pins: https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P



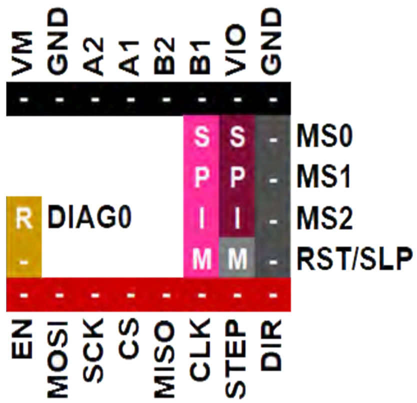
SPI Mode

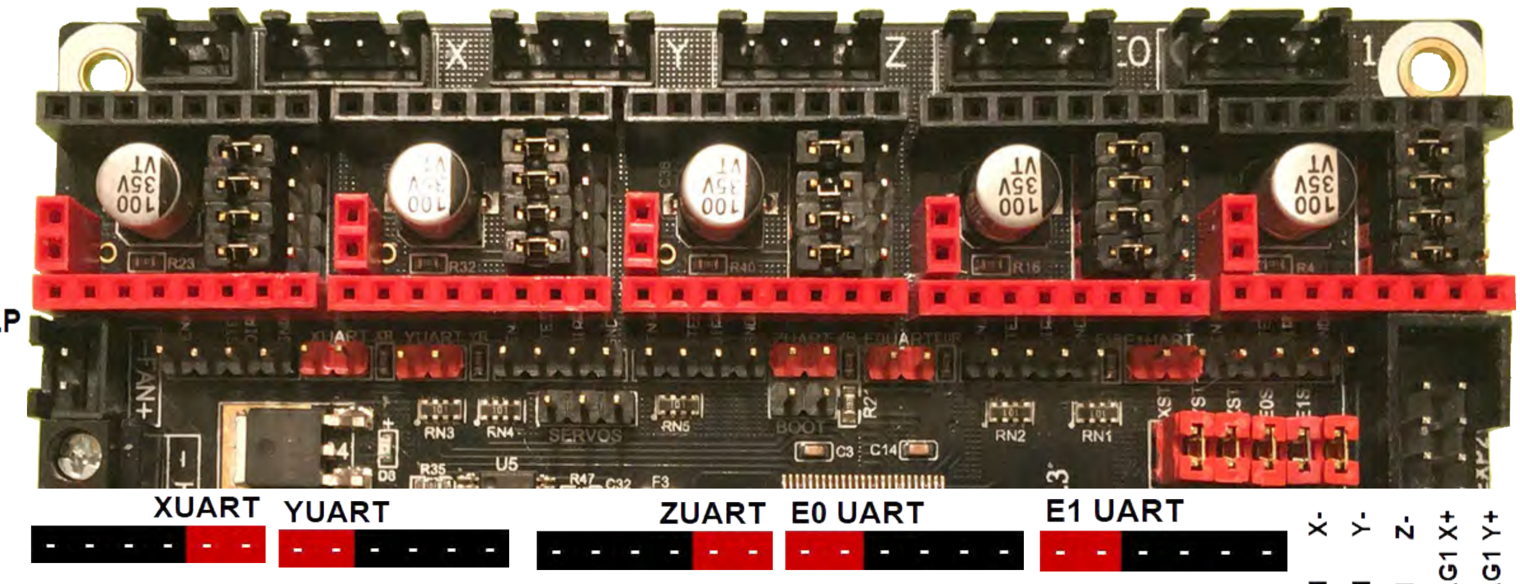
BIQU TMC5161 V1.0

SPI Mode

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

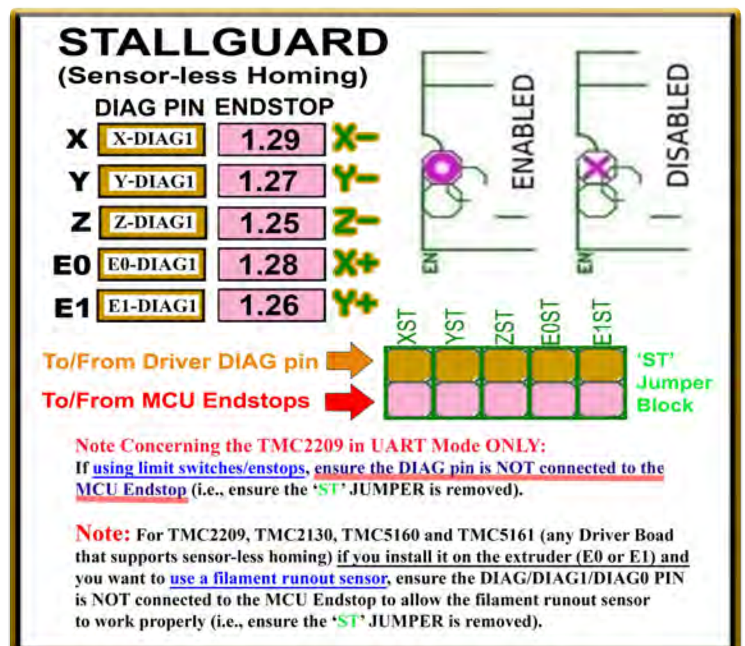
SPI





See <u>Appendix D</u> for legend

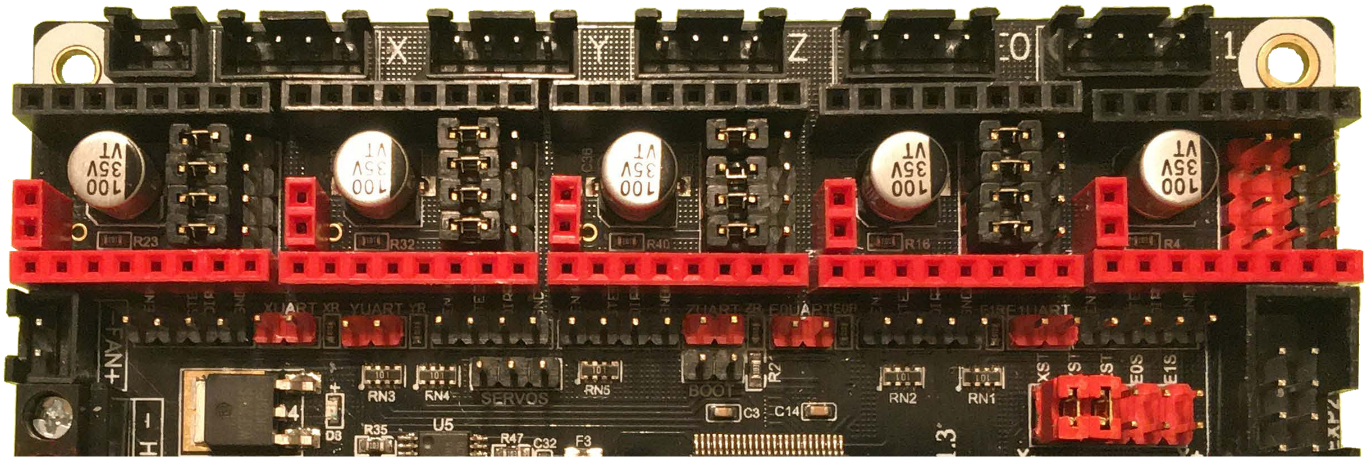
Note: TMC5161 has sensor-less homing capability



Examples of Different SPI Configurations Part 1

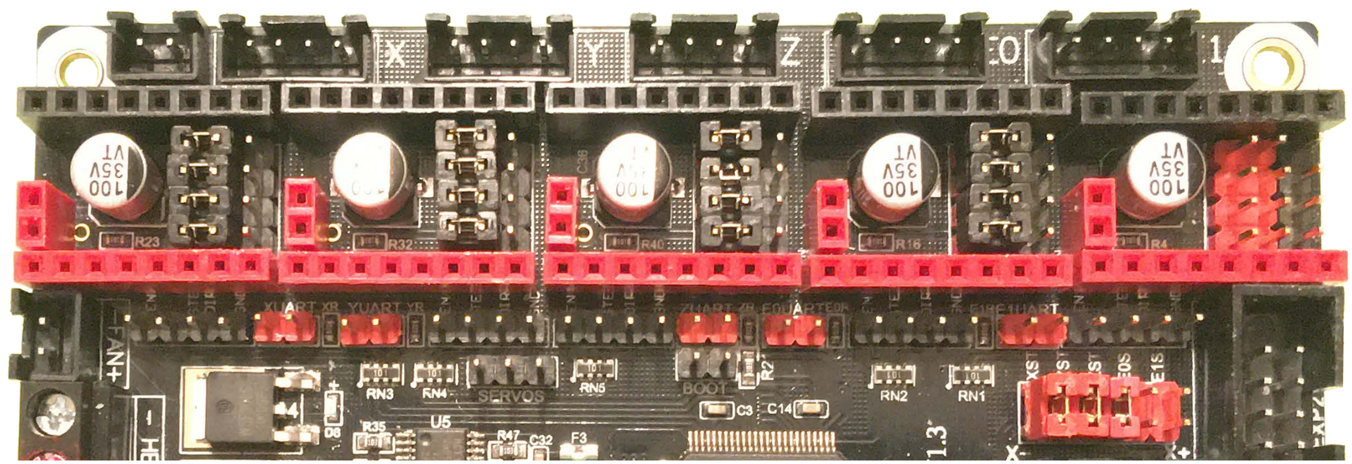
X, Y, Z and E0 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z and E0 axes configured for SPI mode.

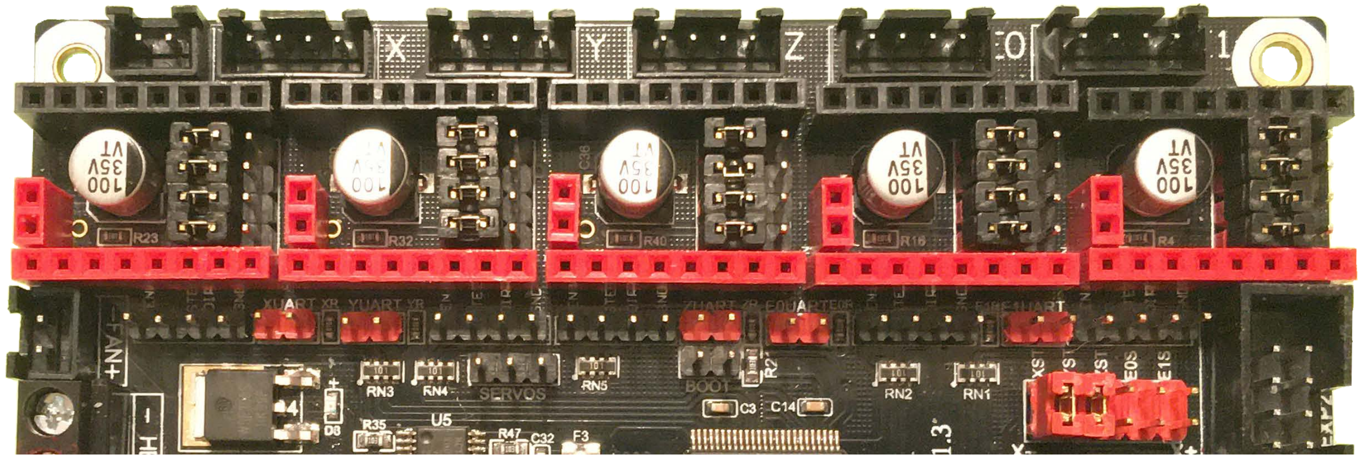
Sensor-less homing for X, Y and Z axes.



Examples of Different SPI Configurations Part 2

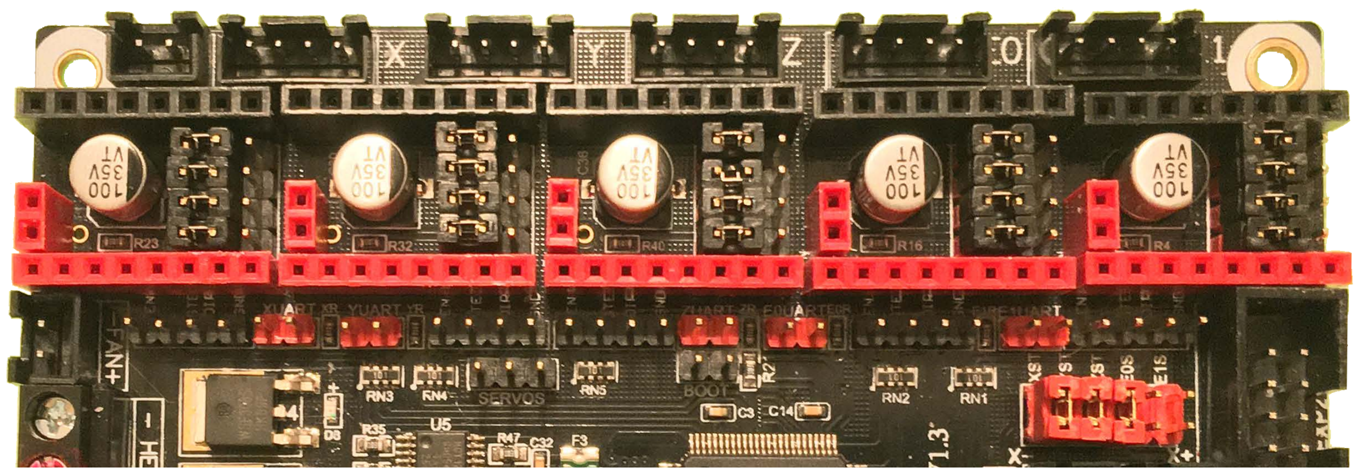
X, Y, Z, E0, and E1 axes configured for SPI mode.

Sensor-less homing for X and Y axes.



X, Y, Z, E0 and E1 axes configured for SPI mode.

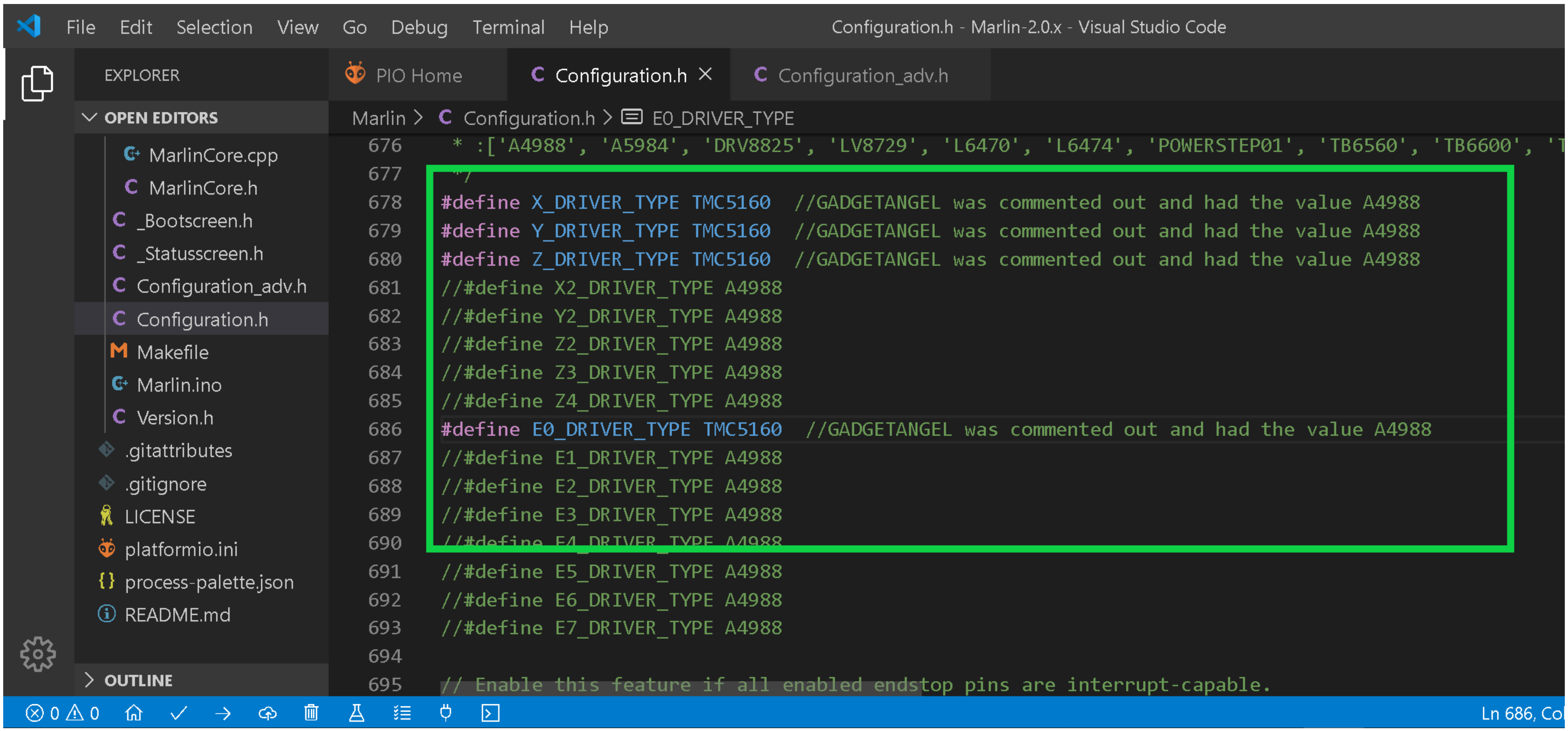
Sensor-less homing for X, Y and Z axes.



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.

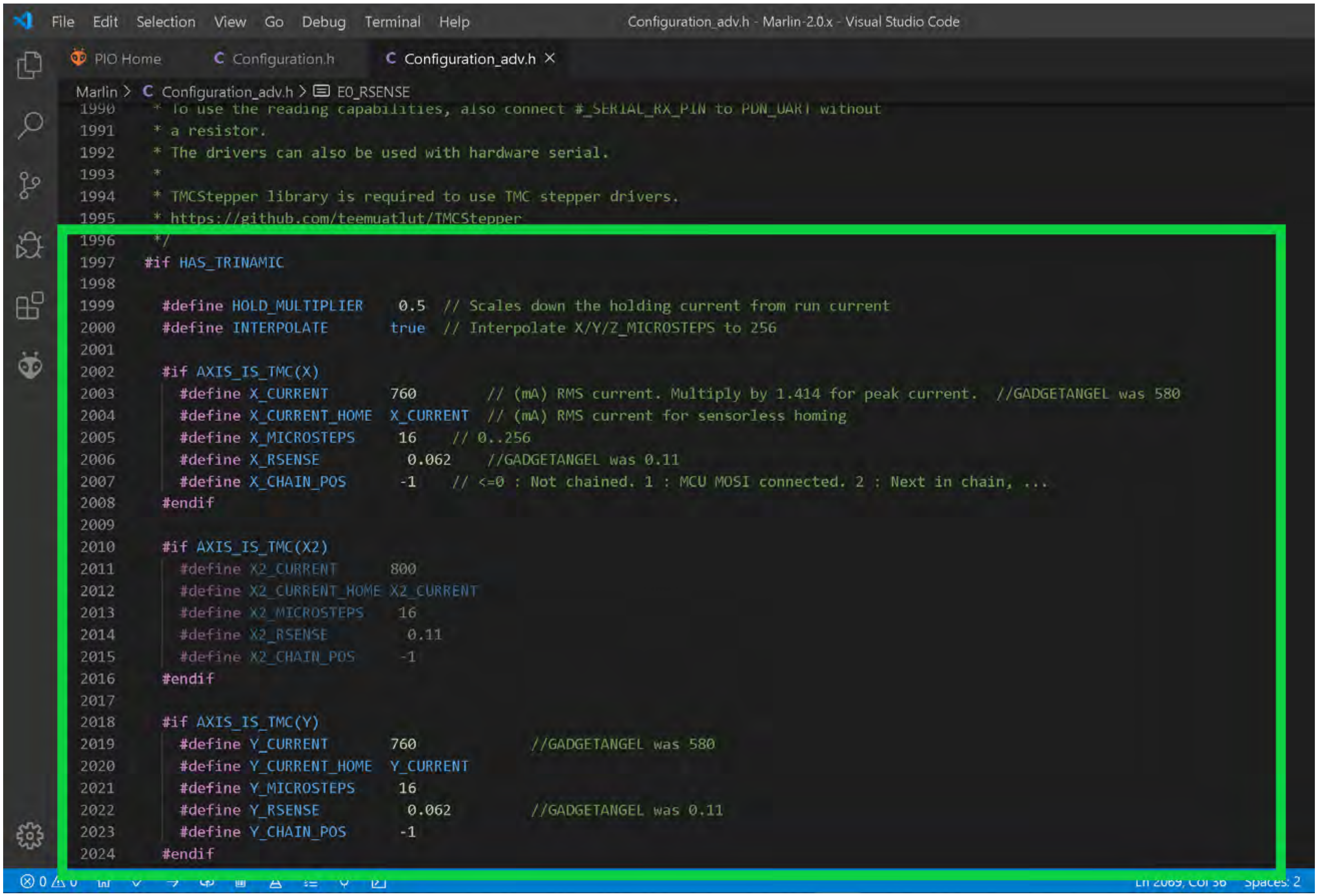
- Change the stepper motor drivers so that Marlin knows you are using TMC5161 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5161 drivers in SPI mode. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

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

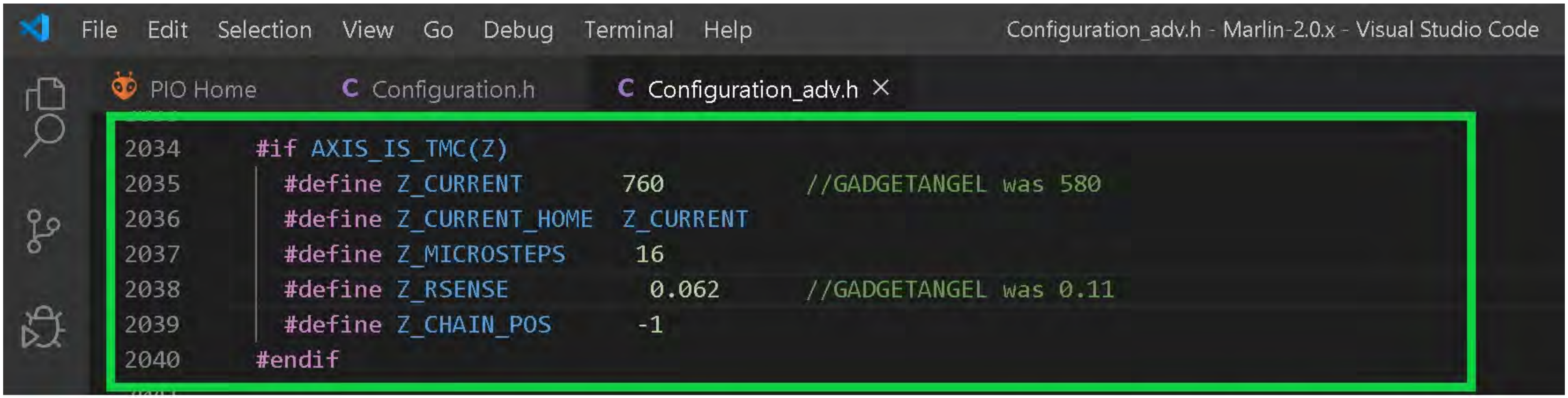
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5161 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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".

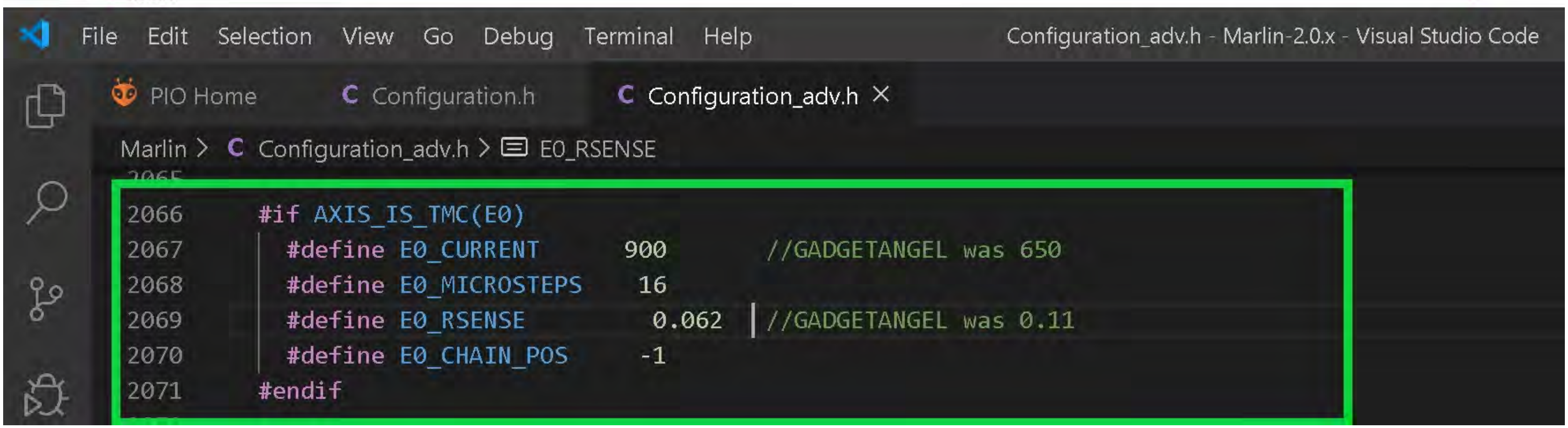


- Go to the next page.

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".

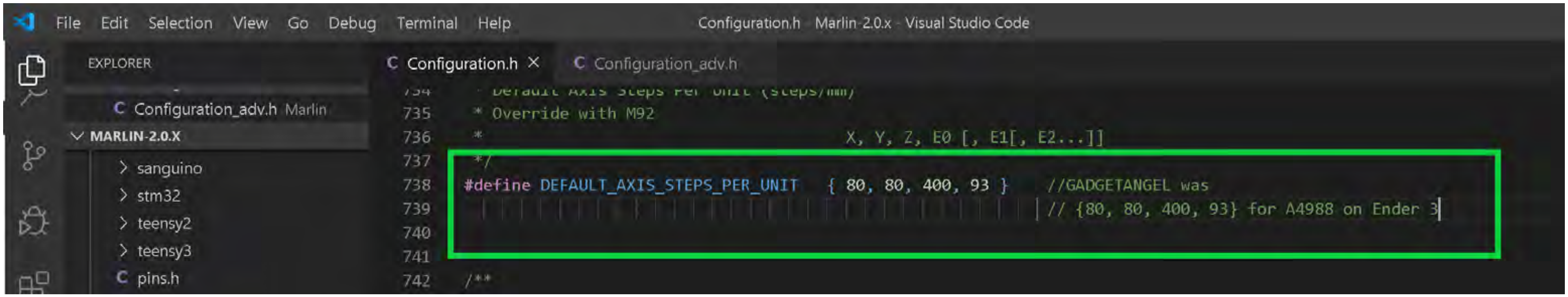




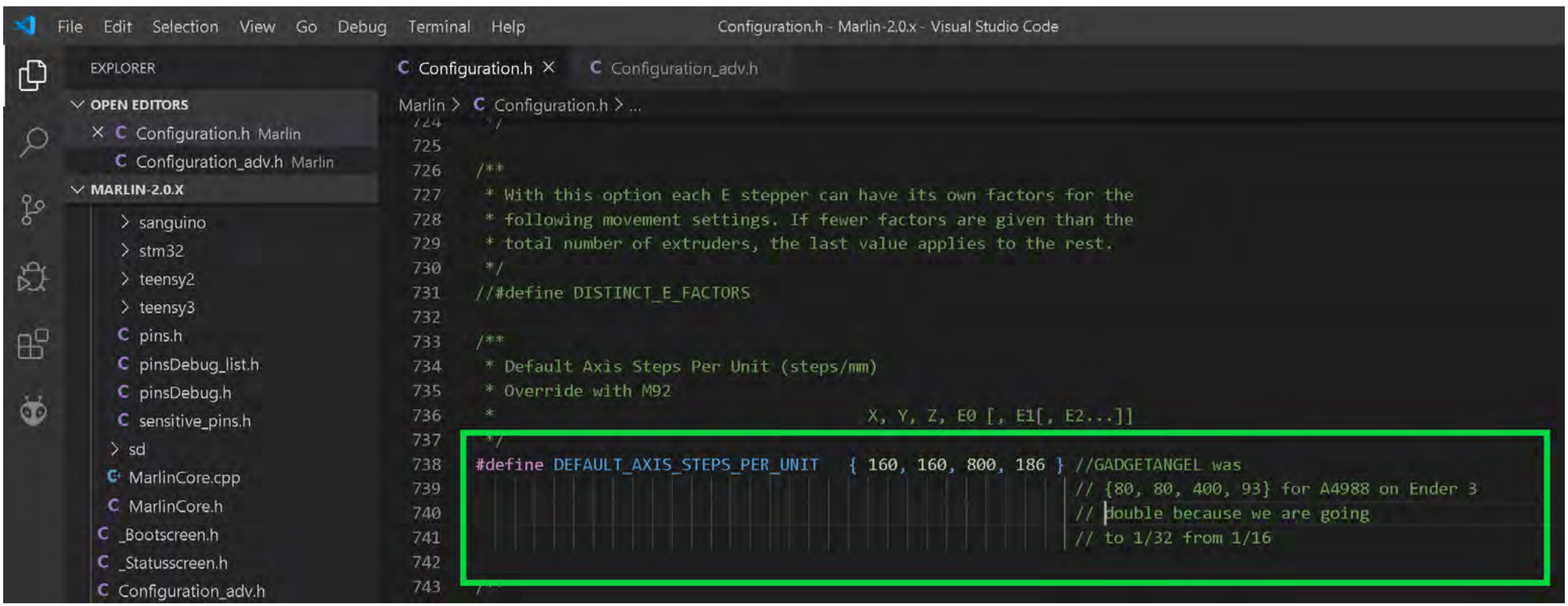
- Go to the next page.

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



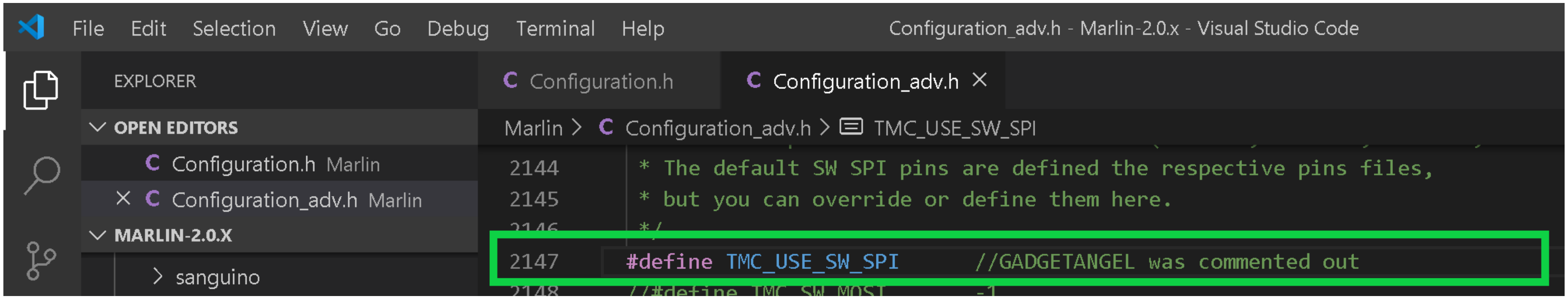
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



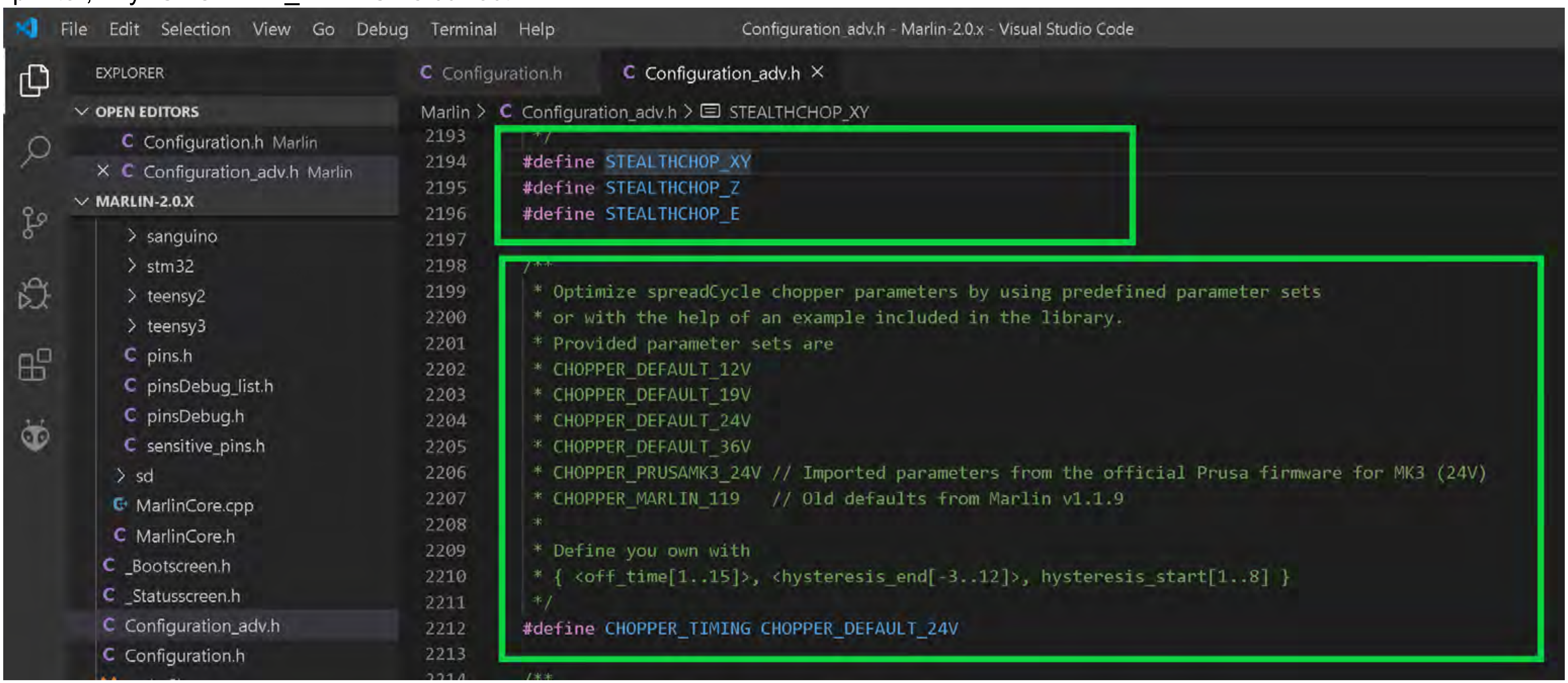
- Go to the next page.

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 V1.3 pins file depends on this variable to define its SPI pins



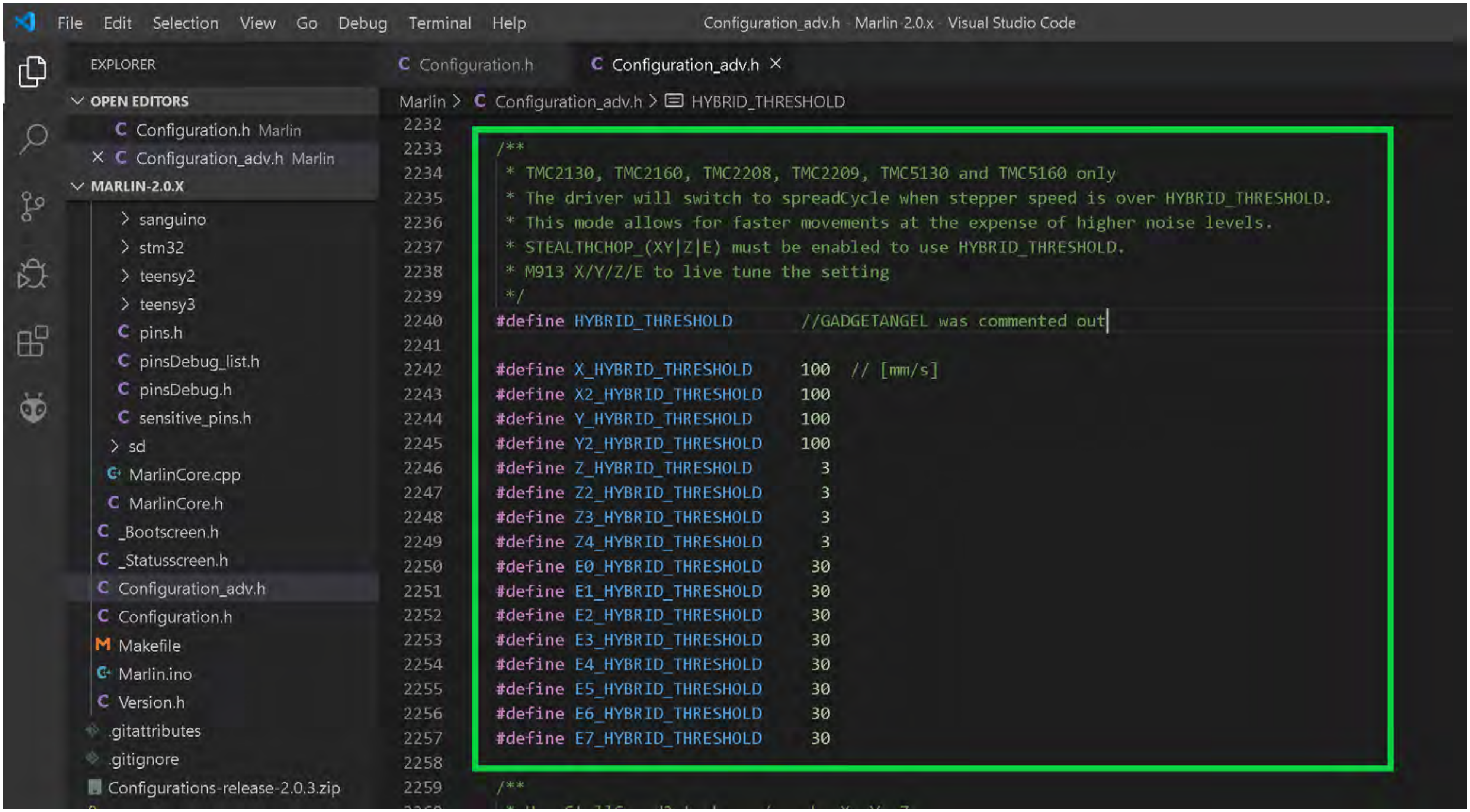
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if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



- Go to the next page.

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

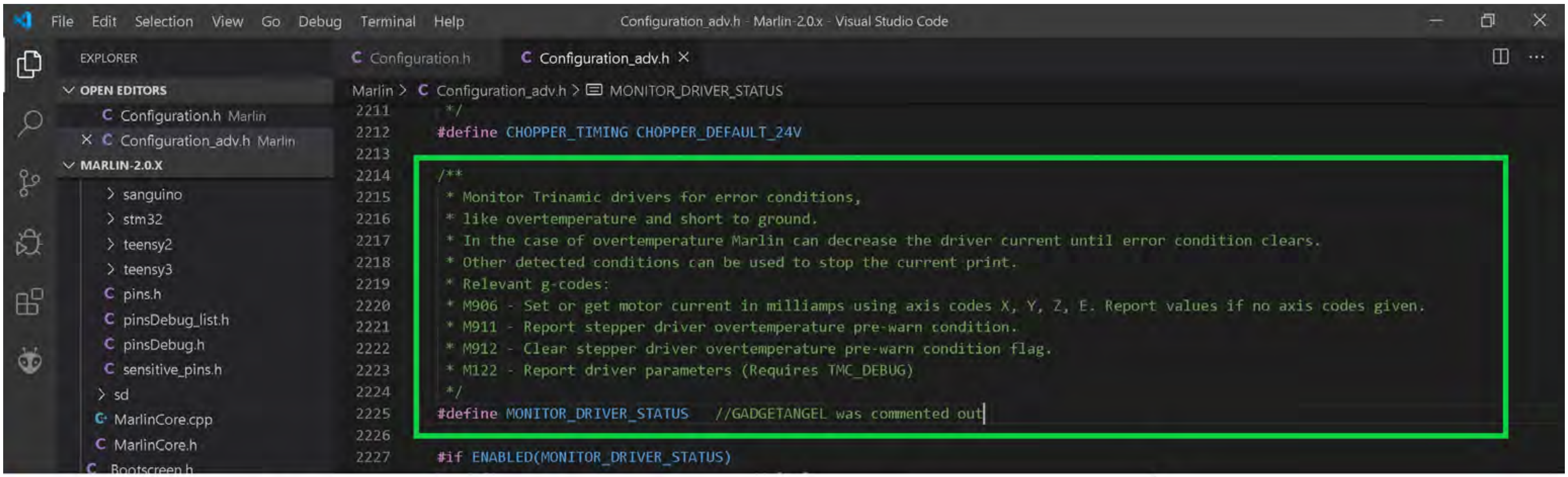
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

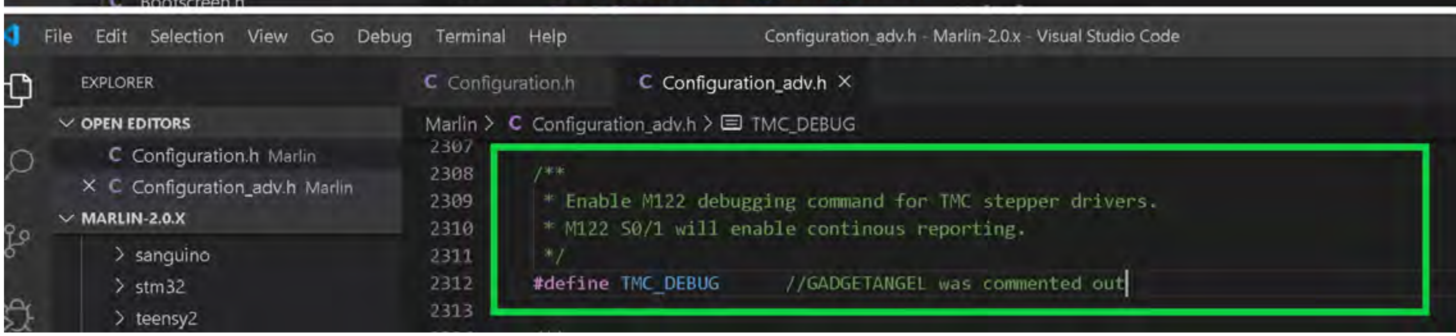


- Go to the next page.

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.**

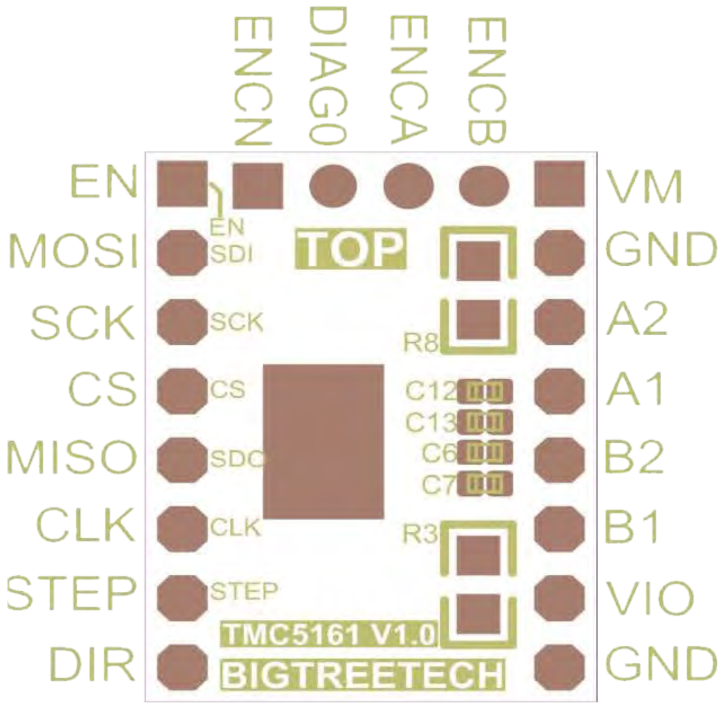


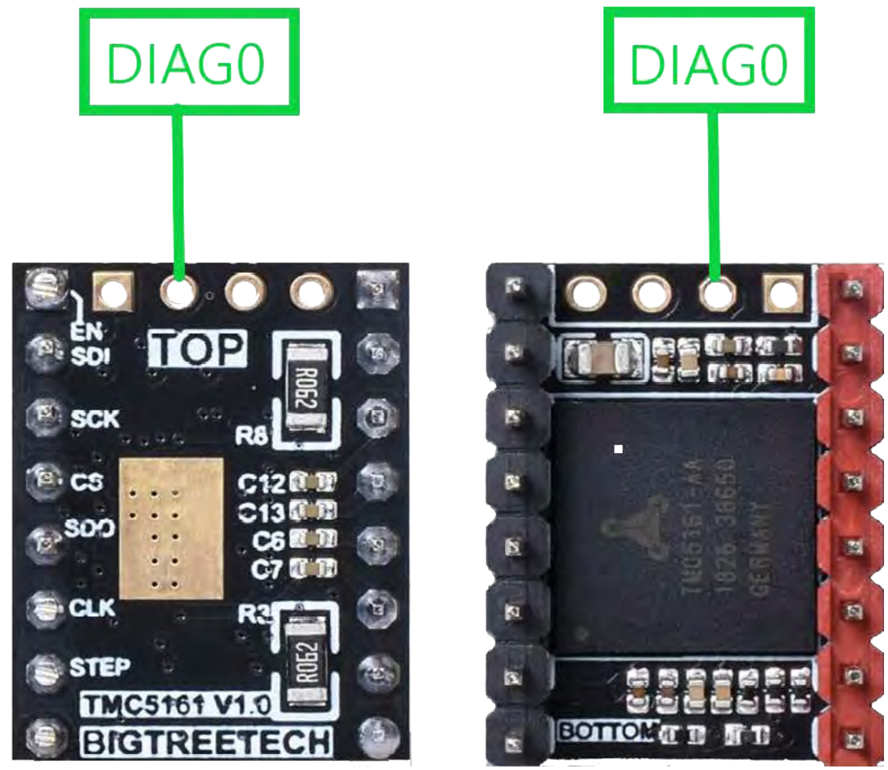


- Go to the next page.

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.



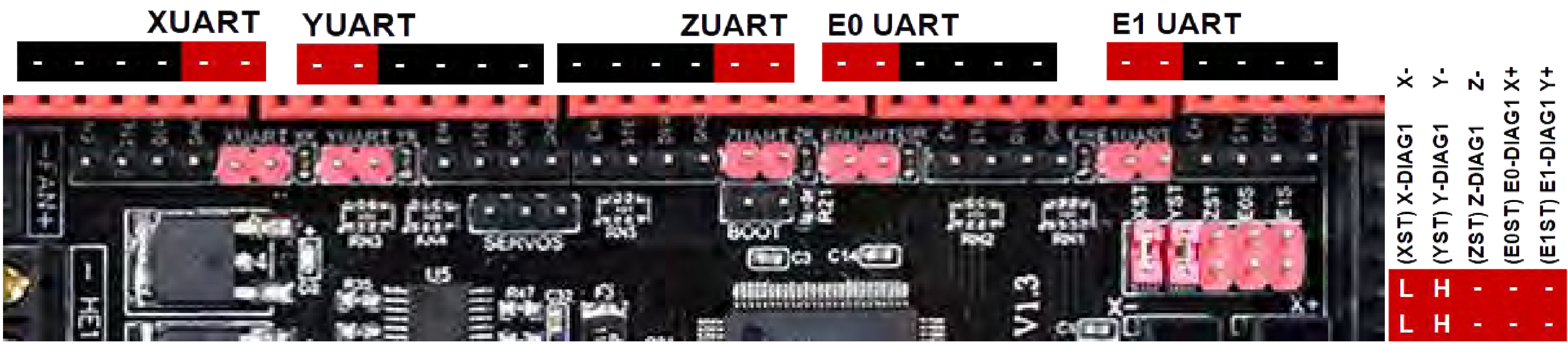


- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR V1.3 board the DIAG0 pin will be seated in the SKR V1.3 board. Then I can use the XST jumper to connect the DIAG0 pin to the X endstop. I can then, also, use the YST jumper to connect the DIAG0 pin to the Y endstop. Connecting the DIAG0 pins to the endstops enables sensor-less homing to work.

- 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 V1.3 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 Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG0 pin of the TMC561 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG0 pin of the TMC5161 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers that are in place are the "L" and "H" jumpers.



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

- 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 ×
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY

#if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
  // TMC2209: 0...255. TMC2130: -64...63
  #define X_STALL_SENSITIVITY 15                //GADGETANGEL was 8, needed for sensorless homing
  #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
  #define Y_STALL_SENSITIVITY 15                //GADGETANGEL was 8, needed for sensorless homing
  //#define Z_STALL_SENSITIVITY  8
  //#define SPI_ENDSTOPS              // TMC2130 only
  //#define IMPROVE_HOMING_RELIABILITY
#endif

/**
```

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

```
g  Terminal  Help          Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h    C Configuration_adv.h ×
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
// Homing hits each endstop, retracts by these distances, then does a slower bump.
#define X_HOME_BUMP_MM 0          //GADGETANGEL was 5
#define Y_HOME_BUMP_MM 0          //GADGETANGEL was 5
#define Z_HOME_BUMP_MM 2
#define HOMING_BUMP_DIVISOR { 2, 2, 4 }  // Re-Bump Speed Divisor (Divides the Homing Feedrate)
#define QUICK_HOME                       // If homing includes X and Y, do a diagonal move initially
//#define HOMING_BACKOFF_MM { 2, 2, 2 }  // (mm) Move away from the endstops after homing

// When G28 is called, this option will make Y home before X
//#define HOME_Y_BEFORE_X
```

- 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 ×    C Configuration_adv.h
Marlin > C Configuration.h > ENDSTOPPULLUPS
// Enable pullup for all endstops to prevent a floating state
#define ENDSTOPPULLUPS          //GADGETANGEL this should always be uncommented for an Ender 3
#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  //#define ENDSTOPPULLUP_XMAX
  //#define ENDSTOPPULLUP_YMAX
  //#define ENDSTOPPULLUP_ZMAX
  //#define ENDSTOPPULLUP_XMIN
  //#define ENDSTOPPULLUP_YMIN
  //#define ENDSTOPPULLUP_ZMIN
  //#define ENDSTOPPULLUP_ZMIN_PROBE
#endif
```

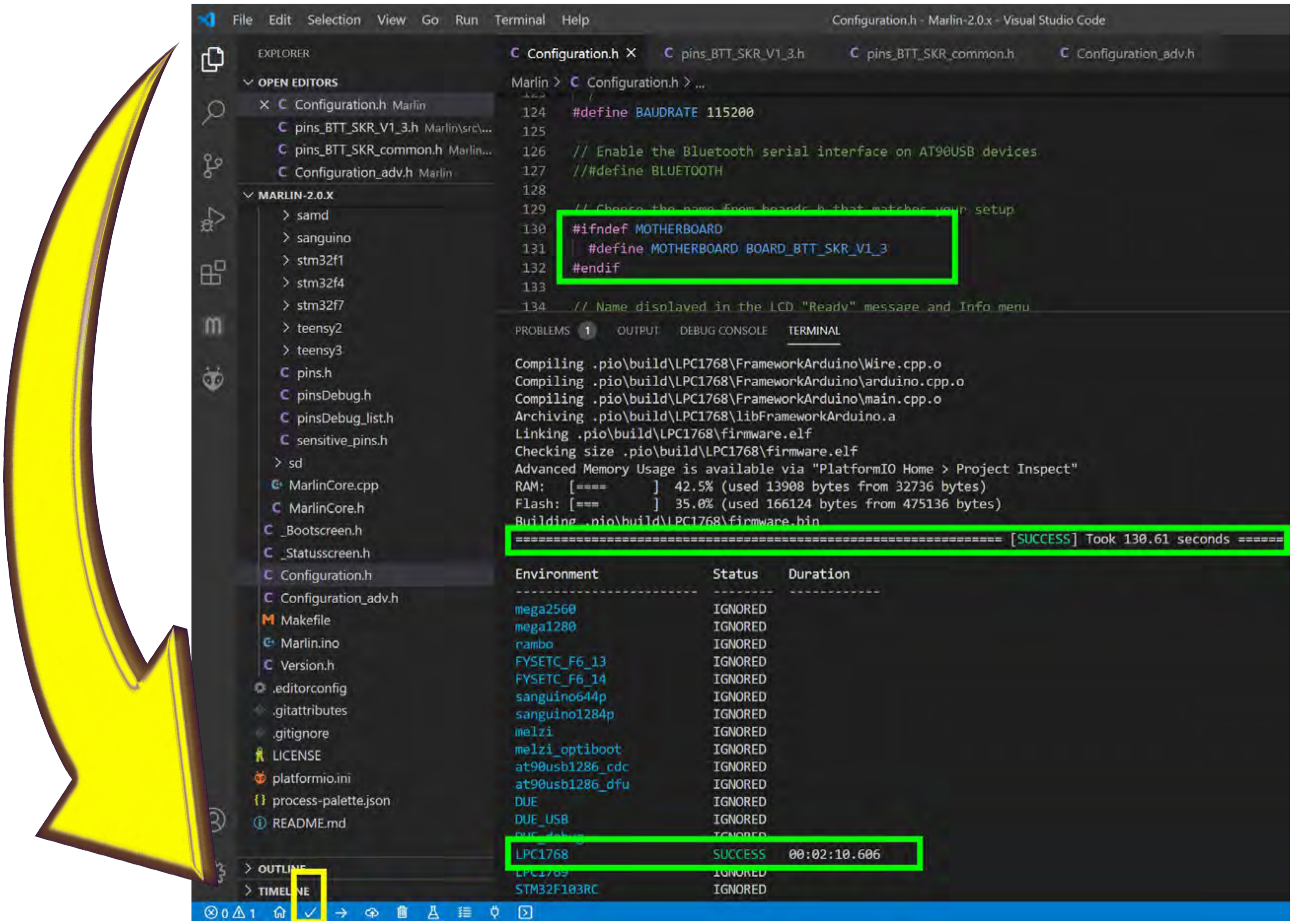
- Next to allow sensor-less homing to work (while using the BIQU TMC5161) we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. 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 ×    C Configuration_adv.h
Marlin > C Configuration.h > X_MIN_ENDSTOP_INVERTING
// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
#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 TMC51
#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 TMC51
#define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
```

- Go to the next page.

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

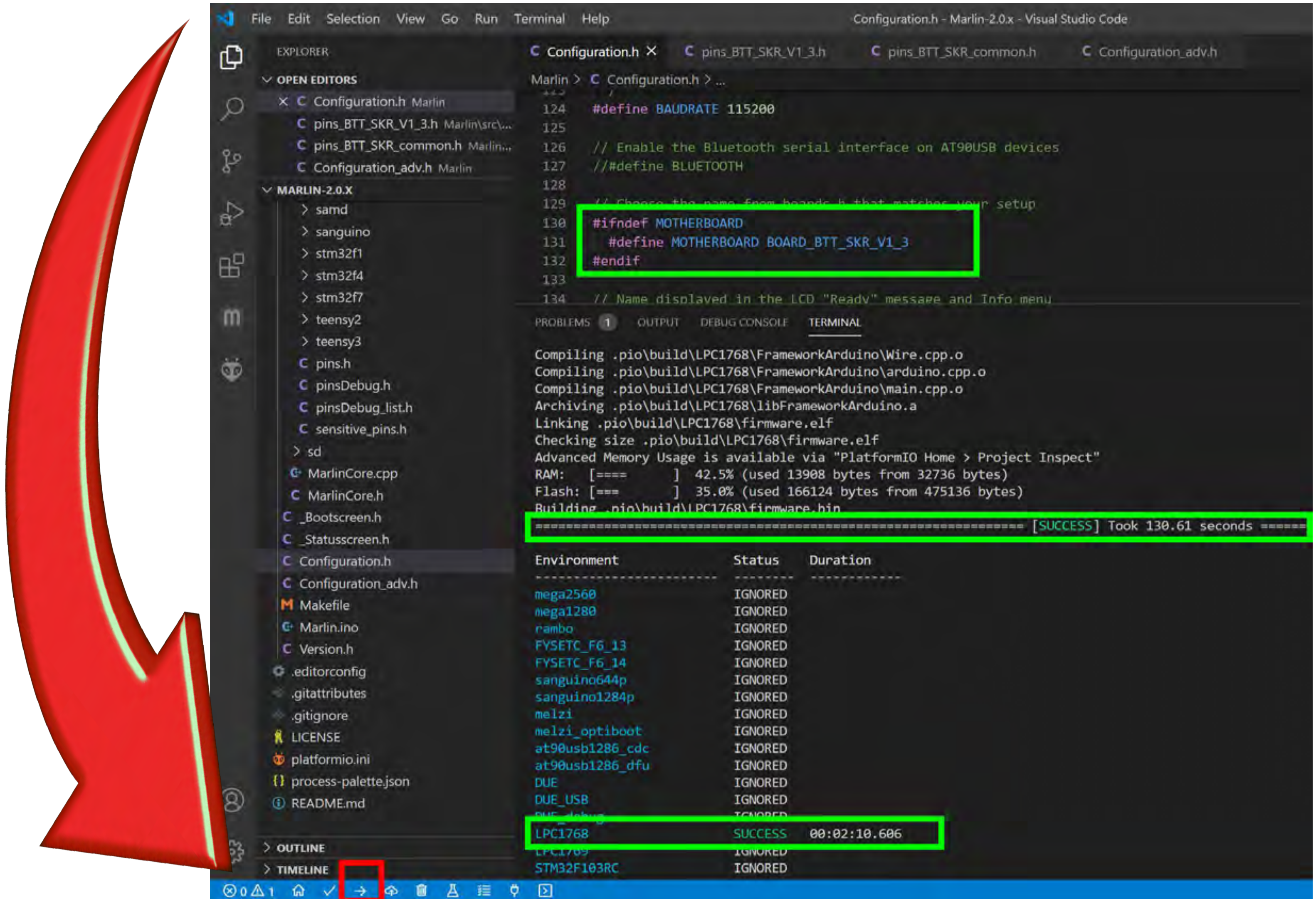
- The end of Marlin setup for BIQU TMC5161 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

- To compile **and** upload the 'firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to **Appendix E** to learn how to find the "firmware.bin" file for the SKR V1.3 board.

How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer[1, 2]

Modern 3D printers usually use NEMA17 motors. The first piece of information you will need is the "Rated Current" of your NEMA17 motor. So, write down your motor's part number and pay attention to the Step Angle, Holding Torque, Rated Current, Voltage, and Inductance.

Use the Step Angle to work out your printer's "DEFAULT_AXIS_STEPS_PER_UNIT" for your firmware with:

1. **http://www.prusaprinters.org/calculator/**

How to Tune Stepper Motor Drivers[2]

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 V1.3 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 V1.3 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 V1.3 board is fine to use. Look up the correct current for your motor part number. If you have motors with no part number, assume they have a max of 1.00 amps (I_{MAX}) to be safe. Look up the proper formula for your stepper motor drivers (as show in this document), and find the voltage (but ONLY use 90% of the calculated V_{ref}) which corresponds with the current you want to set. Use a ceramic screw driver to adjust the POT. A ceramic screw driver is nonconductive and if you slip while making the adjustment to the POT you could short circuit the stepper motor driver board (i.e. KILL the driver board) in the process.

Time Saver tip, but more dangerous: Get slip-on alligator clips for your multimeter. Clamp ground to a 12VDC/24VDC ground (PSU Ground) wire and clamp positive to your plastic handle screwdriver. This way you'll measure the voltage as you adjust ("live adjustment") and don't need three hands.

Note: **See the next page for a diagram of the setup.**

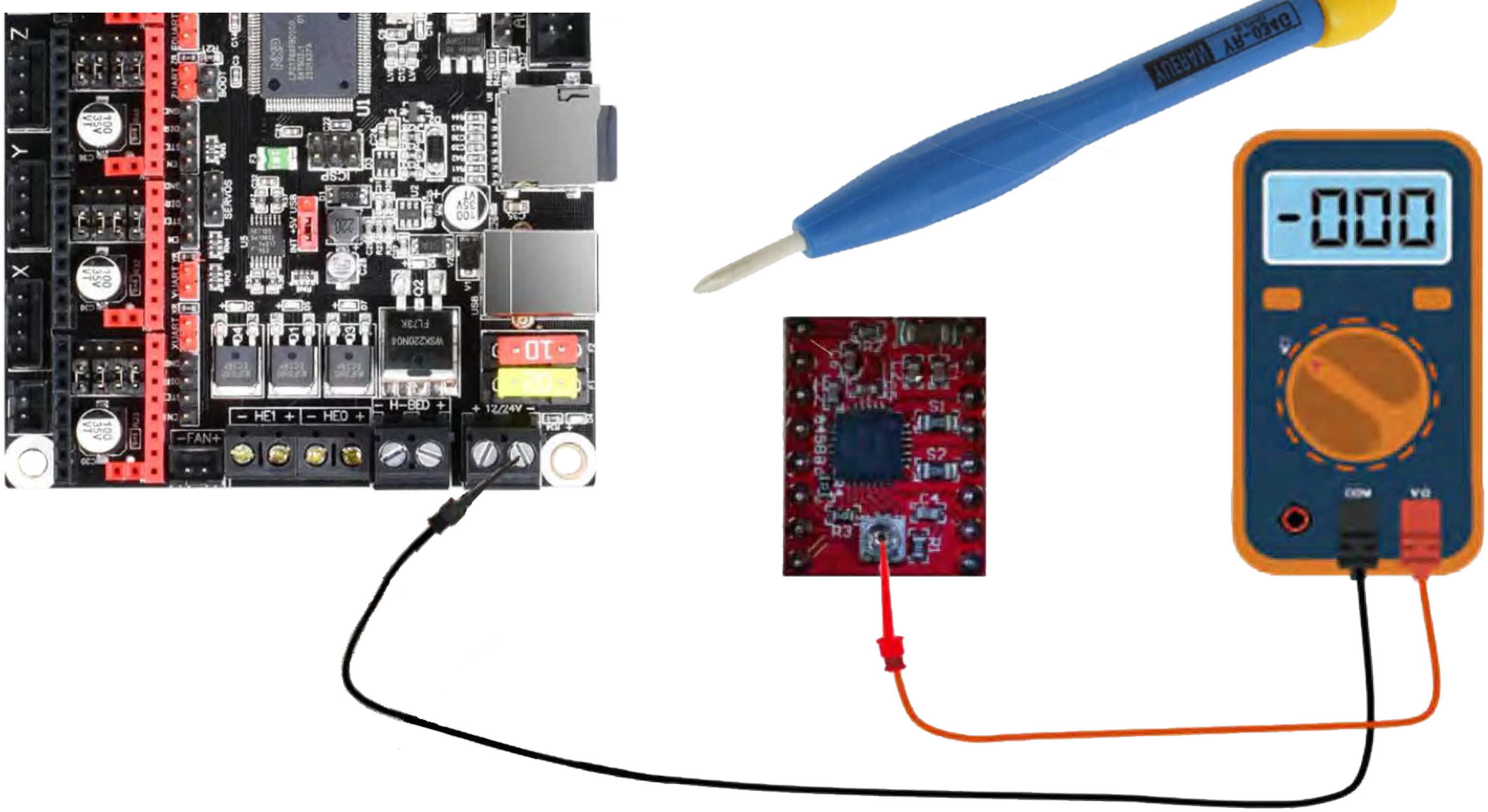
[1] from https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers and
[2] from https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers

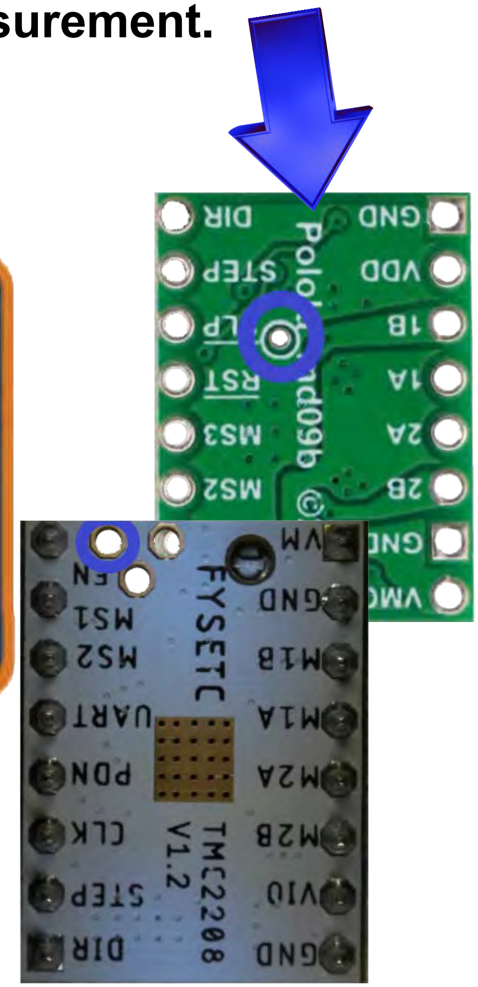
How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer[1, 2]

Note: A ceramic screw driver is non-conductive. If you use a plastic handle screw driver with alligator clips to your multimeter to make a "live adjustment" of V_{ref} (and you slip while making the adjustment to the POT), you could short circuit the stepper motor driver board (i.e. KILL the board) in the process.

Note: Some stepper motor driver boards have a "V_{ref} Test point" location, shown in BLUE. Check bottom or top of your board for a location.

If a "V_{ref} Test point" location is not available, use the potentiometer on the stepper driver board to do the V_{ref} measurement.





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

APPENDIX B

For the TMC drivers what's the difference between stand-alone mode and ("UART" or "SPI ") modes?

All the TMC driver chips **EXCEPT TMC5160 and TMC5161** have a stand-alone mode. **Stand-alone mode** allows you to just drop the driver into your motherboard to replace your A4988 driver. The **OTP mode** is similar, but you use some software to **PERMANENTLY** change the driver's mode of operation. The **UART or SPI modes** allow you to **dynamically** change the driver in your firmware.

How to Calculate V_{ref} for Non-TMC Stepper Motor Divers

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.**

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 for this example 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:

$I_{MAX} = (V_{ref} / 2.5) * (2.46 \cong 2.5),$

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 B

Driving Current Calculation Formulas for TMC Stepper Motor Drivers

1. TMC2100 with R_s = 0.110Ω (110mΩ) :

I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((**320mV** / (**R_s + 20mΩ**)))
= ((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

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 = 0.110Ω (110mΩ) :

I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((**325mV** / (**R_s + 20mΩ**)))
= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20)))
= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5))
= (1 / 1.41) * V_{ref}

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 = 0.110Ω (110mΩ) for Stand-alone Mode:

I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((**325mV** / (**R_s + 30mΩ**)))
= ((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

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 B

Driving Current Calculation Formulas for TMC Stepper Motor Drivers

4. TMC2208 with $R_s = 0.110\Omega$ ($110m\Omega$) for UART Mode:

$I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((\mathbf{325mV} / (\mathbf{R_s} + \mathbf{20m\Omega})\ ))$

$= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20)))$

$= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5))$

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

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

Therefore,

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

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

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

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

$I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((\mathbf{325mV} / (\mathbf{R_s} + \mathbf{20m\Omega})\ ))$

$= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / (110 + 20)))$

$= ((\ 1 / 1.41) * (V_{ref} / 2.5) * (2.5))$

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

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

Therefore,

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

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

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

See next page for other TMC stepper motor drivers

APPENDIX B

Driving Current Calculation Formulas for TMC Stepper Motor Drivers

6. TMC2209 with R_s = 0.110Ω (110mΩ) for UART Mode:

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

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

Therefore,

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

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

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

7. TMC2225 with R_s = 0.150Ω (150mΩ) for UART Mode:

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

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

Therefore,

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

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

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

8. TMC5160 with R_s = 0.075Ω (75mΩ) for SPI Mode:

$I_{RMS} = ((1 / 1.41) * (\mathbf{325mV} / \mathbf{R_s}))$
$= ((1 / 1.41) * (325 / 75))$
$= (1 / 1.41) * 4.33$

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

Therefore,

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

$\mathbf{I_{MAX}}$ **= 4.333 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 B

Driving Current Calculation Formulas for TMC Stepper Motor Drivers

9. TMC5161 with R_s = 0.062Ω (62mΩ) for SPI Mode:

I_{RMS} = ((1 / 1.41) * (**325mV** / **R_s**))
= ((1 / 1.41) * (325 / 62))
= (1/1.41) * 5.24194

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

Therefore,
I_{MAX} * (1 / 1.41) = (1 / 1.41) * 5.24194
I_{MAX} = 5.24194 or 5.2419

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 = 0.150Ω (150mΩ) for Stand-alone Mode:

I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((**325mV** / (**R_s** + **30**mΩ)))
= ((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
Since I_{RMS} = I_{MAX} * (1 / 1.41)

Therefore,
I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref} * 0.7222
I_{MAX} = V_{ref} * 0.7222
V_{ref} = I_{MAX} * 1.3846

APPENDIX C

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

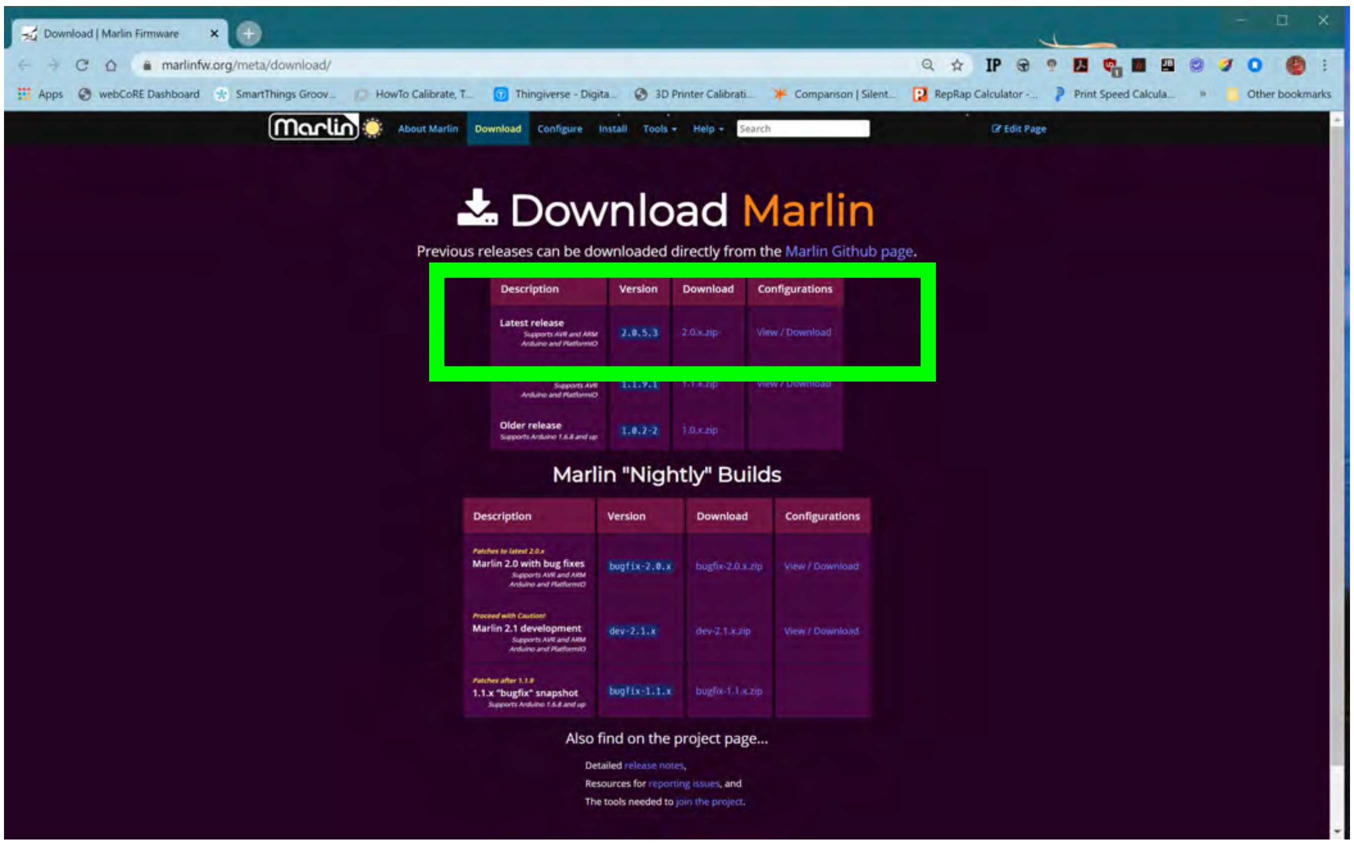
Please follow this guide to install Visual Studio Code with Platformio onto your computer. The link to the instruction are located at: https://marlinfw.org/docs/basics/install_platformio.html

Please refer to the following documents:

- **BIGTREETECH SKR-V1.3 Instruction Manual.pdf**
- **BIGTREETECH SKR-V1.3 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

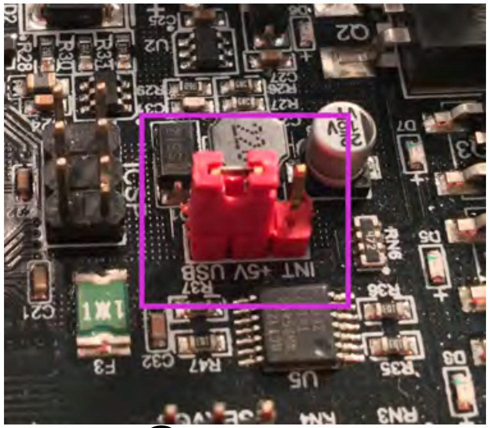


- Go to the next page.

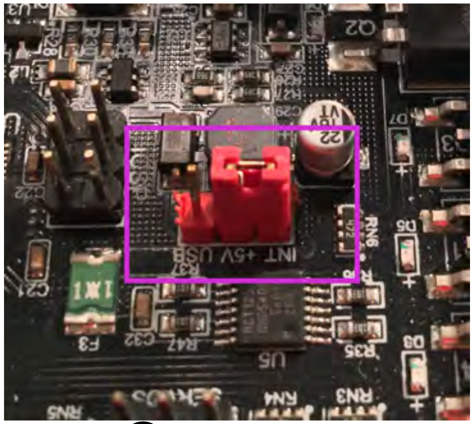
The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

To ensure that any changes I make to my Marlin Firmware are taken as I program it, I do a couple of things to ensure that the parameters are set properly. I will send an 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 V1.3 Instruction manual.pdf", is set to the correct power input source ②. Even though you can use the USB as the source of power when performing a firmware update, I prefer to use the 12V/24V power source, as seen below.



①USB power



②12/24V power

Marlin2.0 Firmware Update Method:

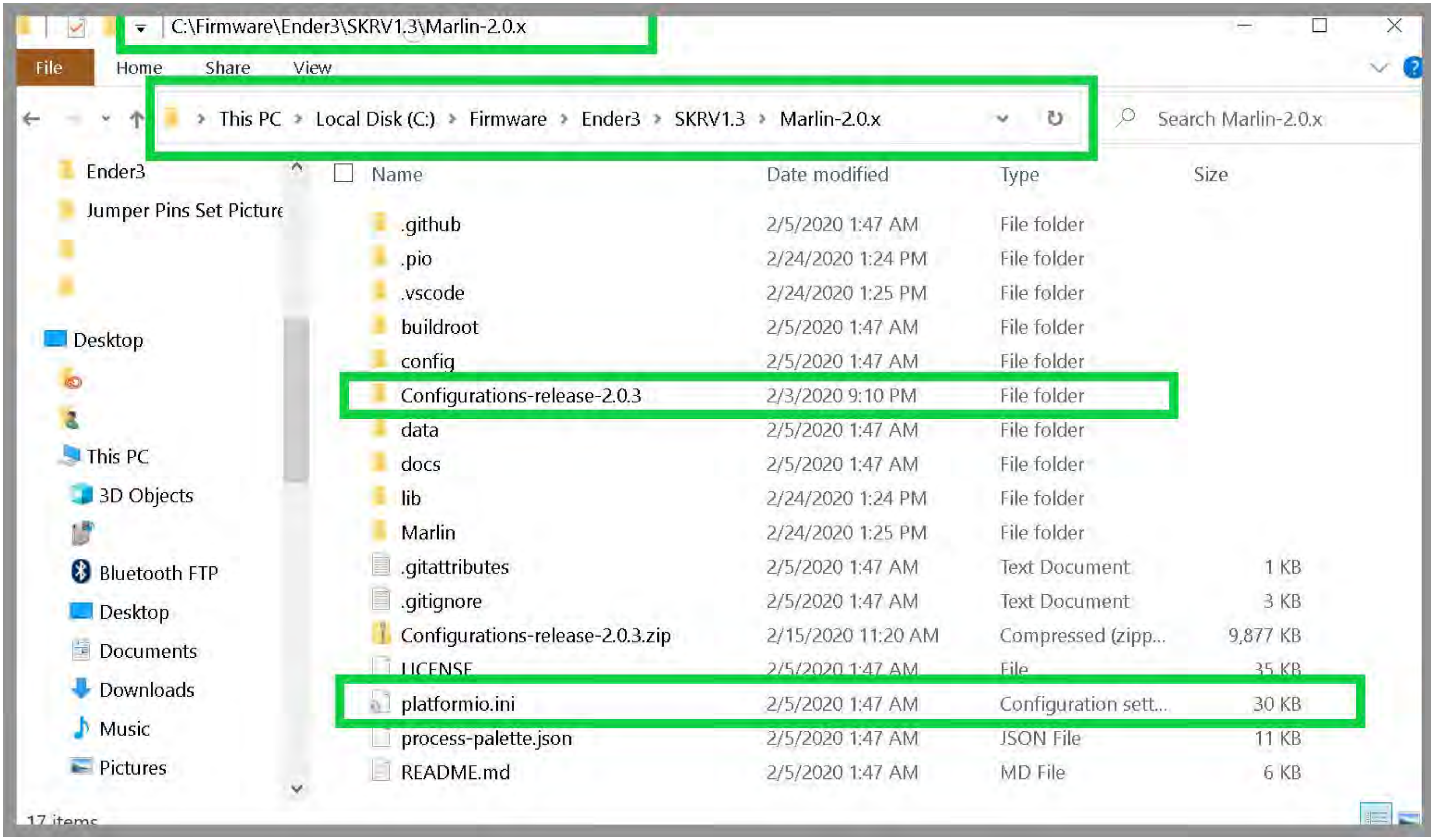
After downloading the files, use Visual Studio Code to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the SD card in the board. Reboot the board; wait for about 10 seconds 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 V1.3 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

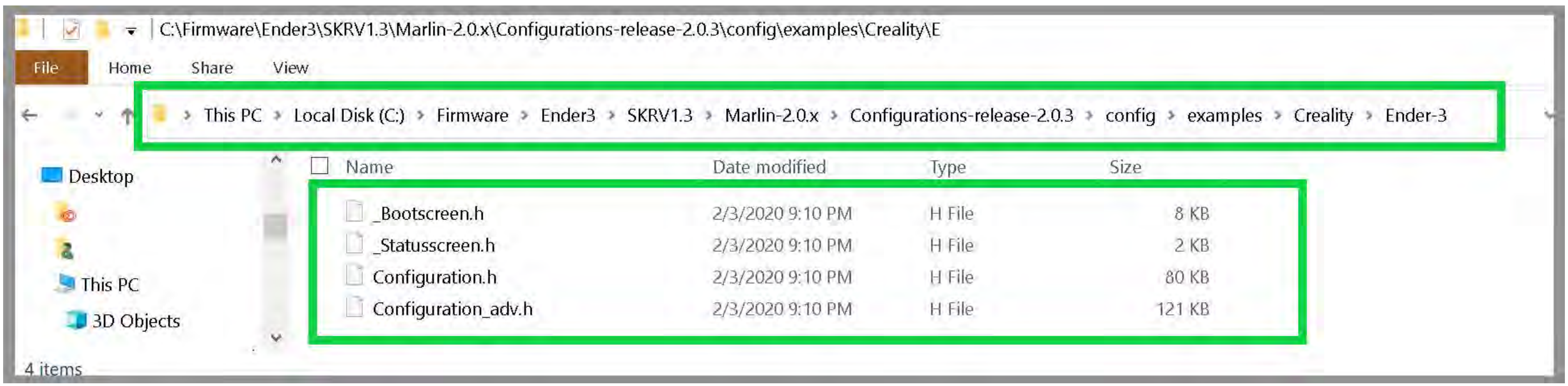


- Go to the next page.

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 uses a release of Marlin firmware which is version 2.0.3. The latest release is version 2.0.5. For version 2.0.5, all files are located in the same sub directories as version 2.0.3. The base code has been changed between these two versions of Marlin. Instead of having to replace all following pictures in this section, just insert "2.0.5" everywhere you see "2.0.3" and select the appropriate default configuration files for your specific printer!

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

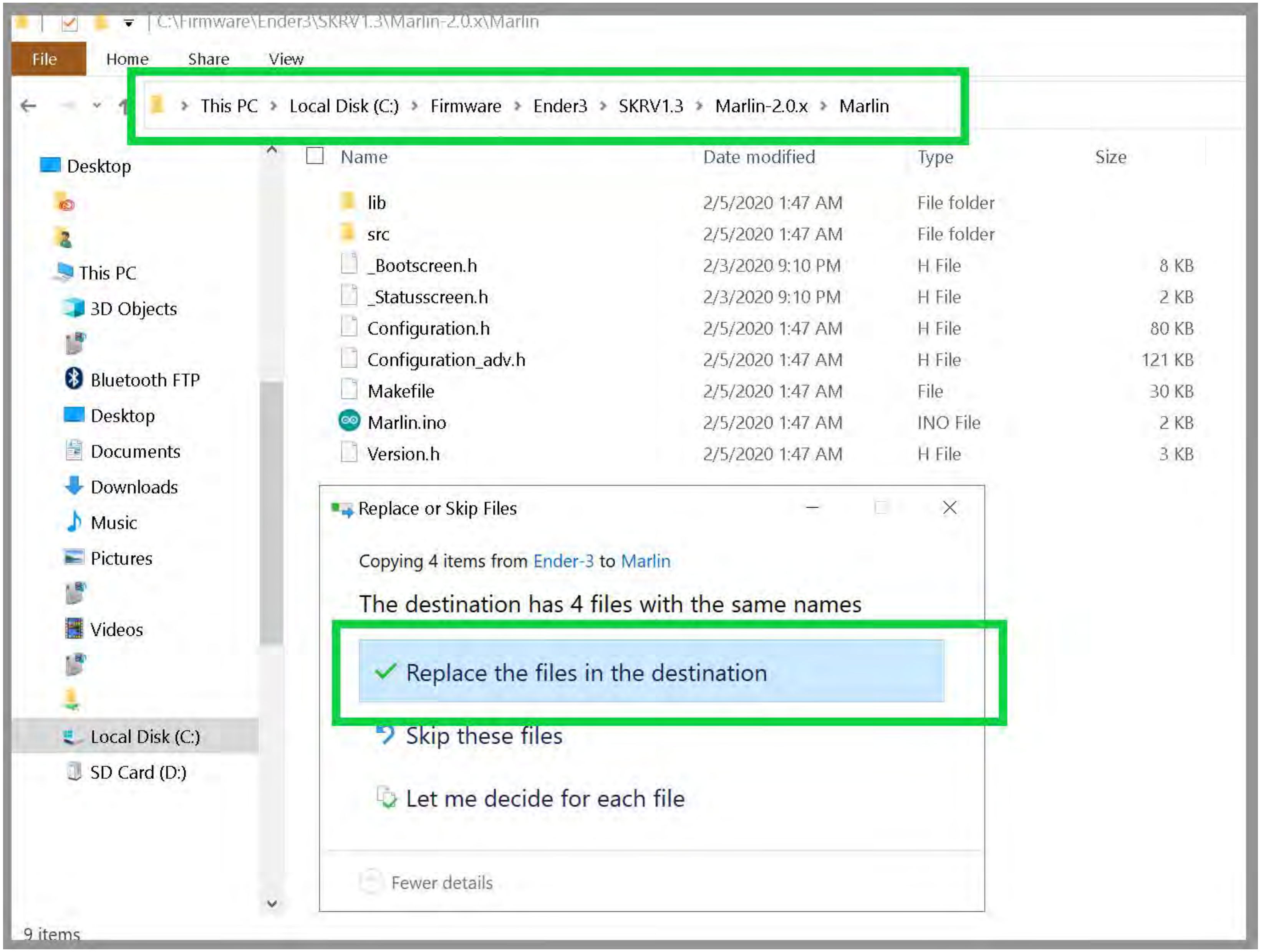


- Go to the next page.

APPENDIX C

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

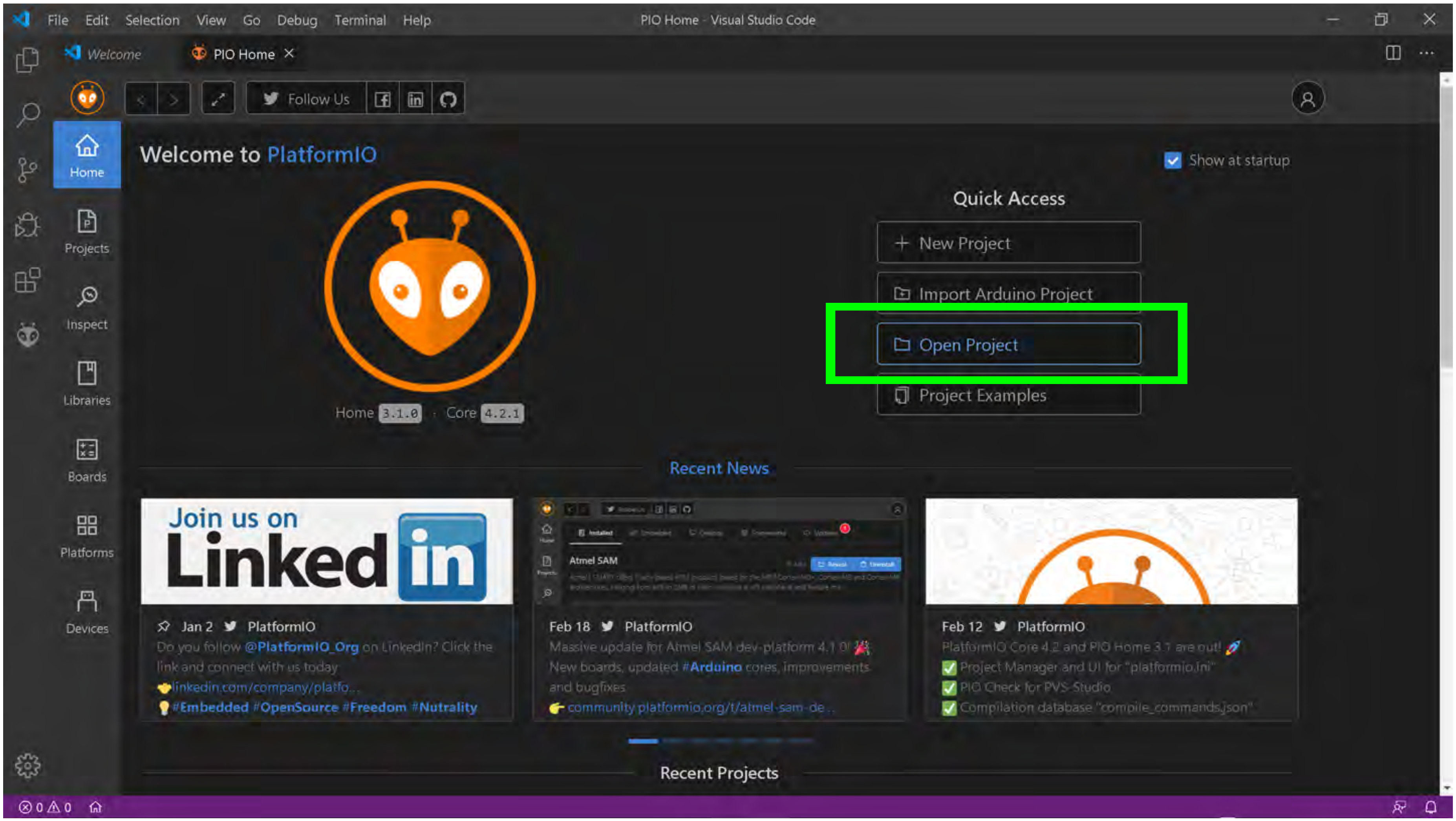
- Copy TO THIS directory so the above files reside in the same directory as Configuation.h and Configuration_adv.h. When prompted allow the files to be overwritten!



- 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".

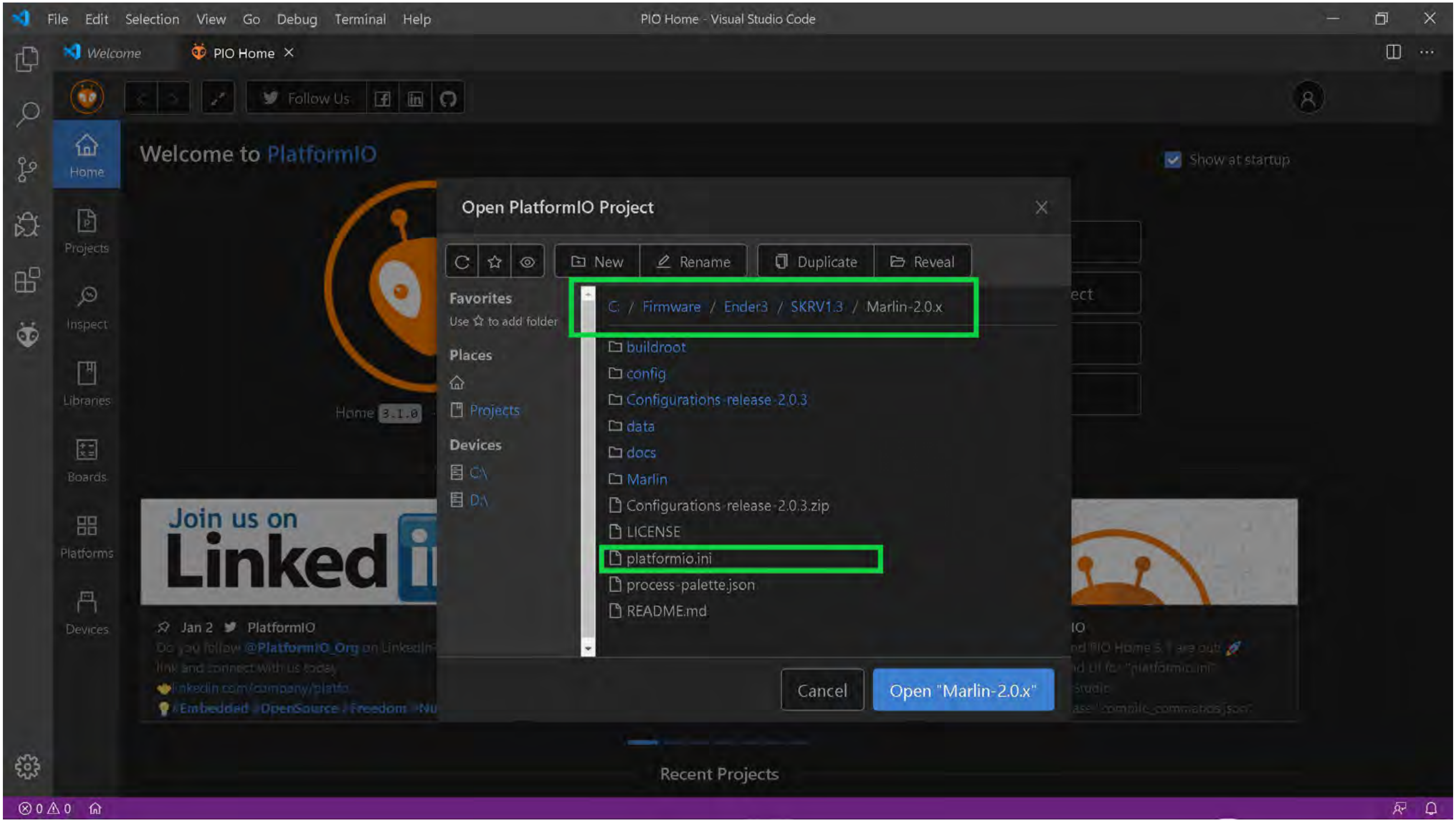


- Go to the next page.

APPENDIX C

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

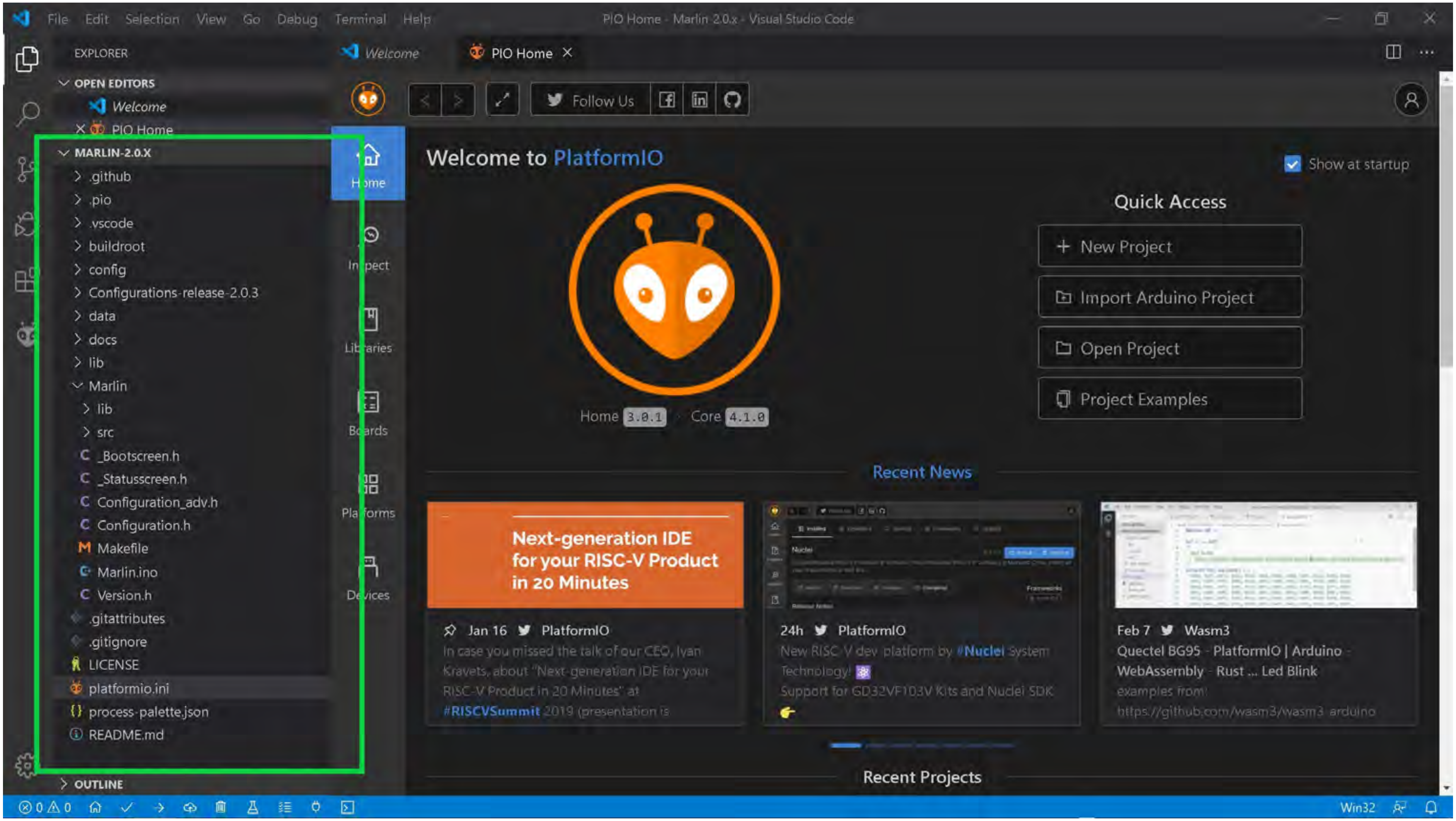
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- 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.

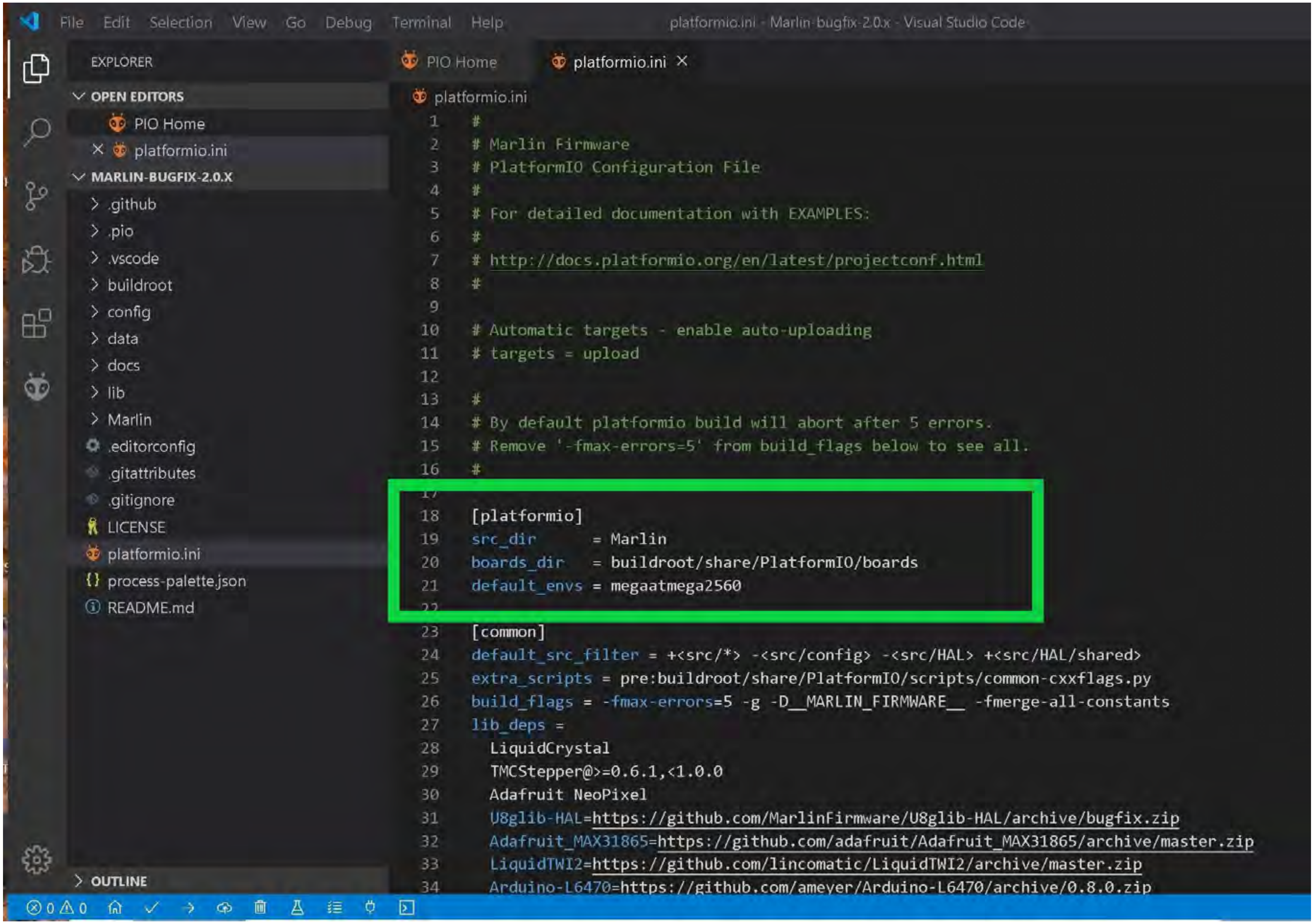


- Go to the next page.

APPENDIX C

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

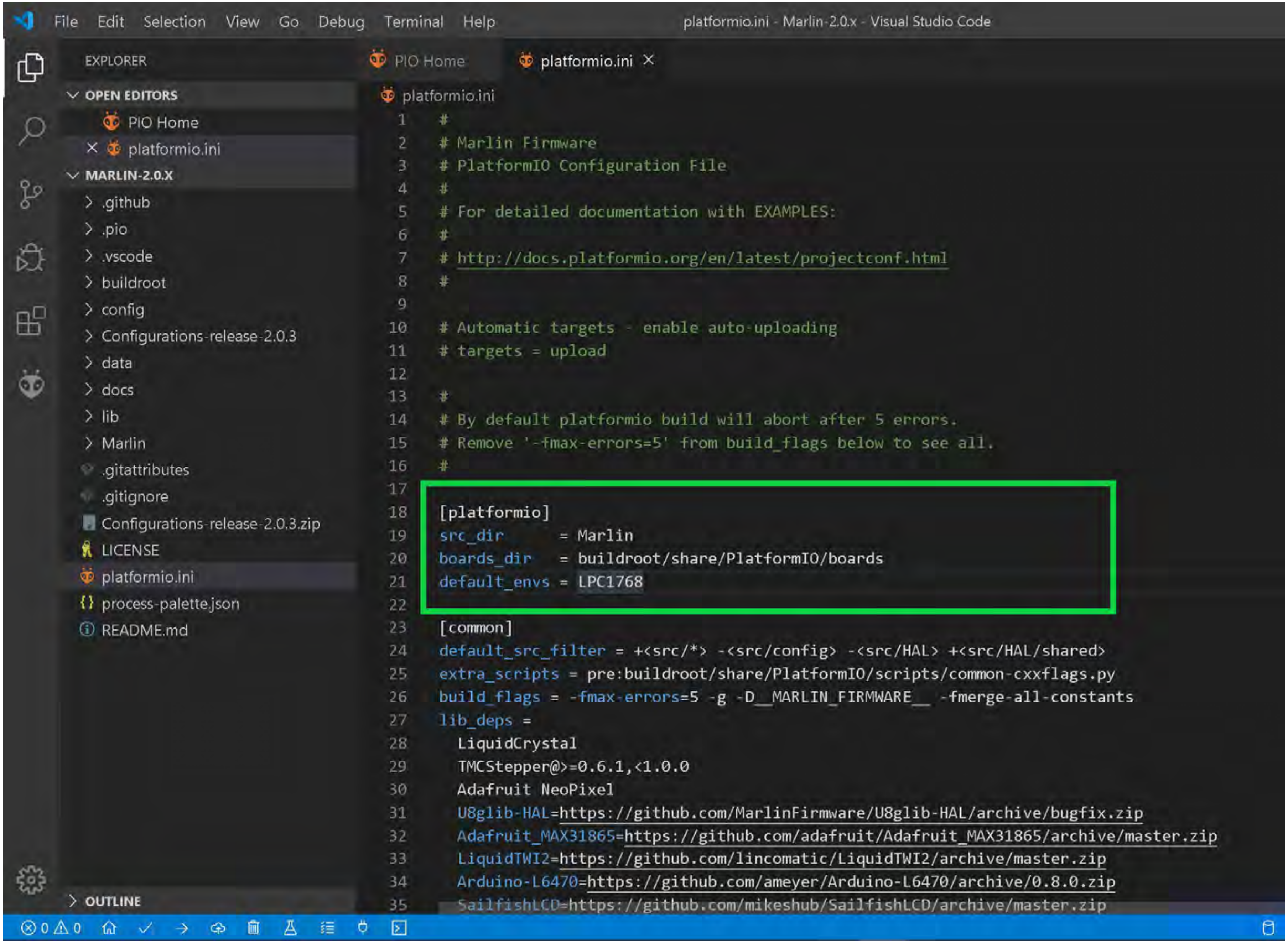
- What you will see when "platformio.ini" is opened up in the VS code editor window



- 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 = LPC1768", as shown below in **GREEN**

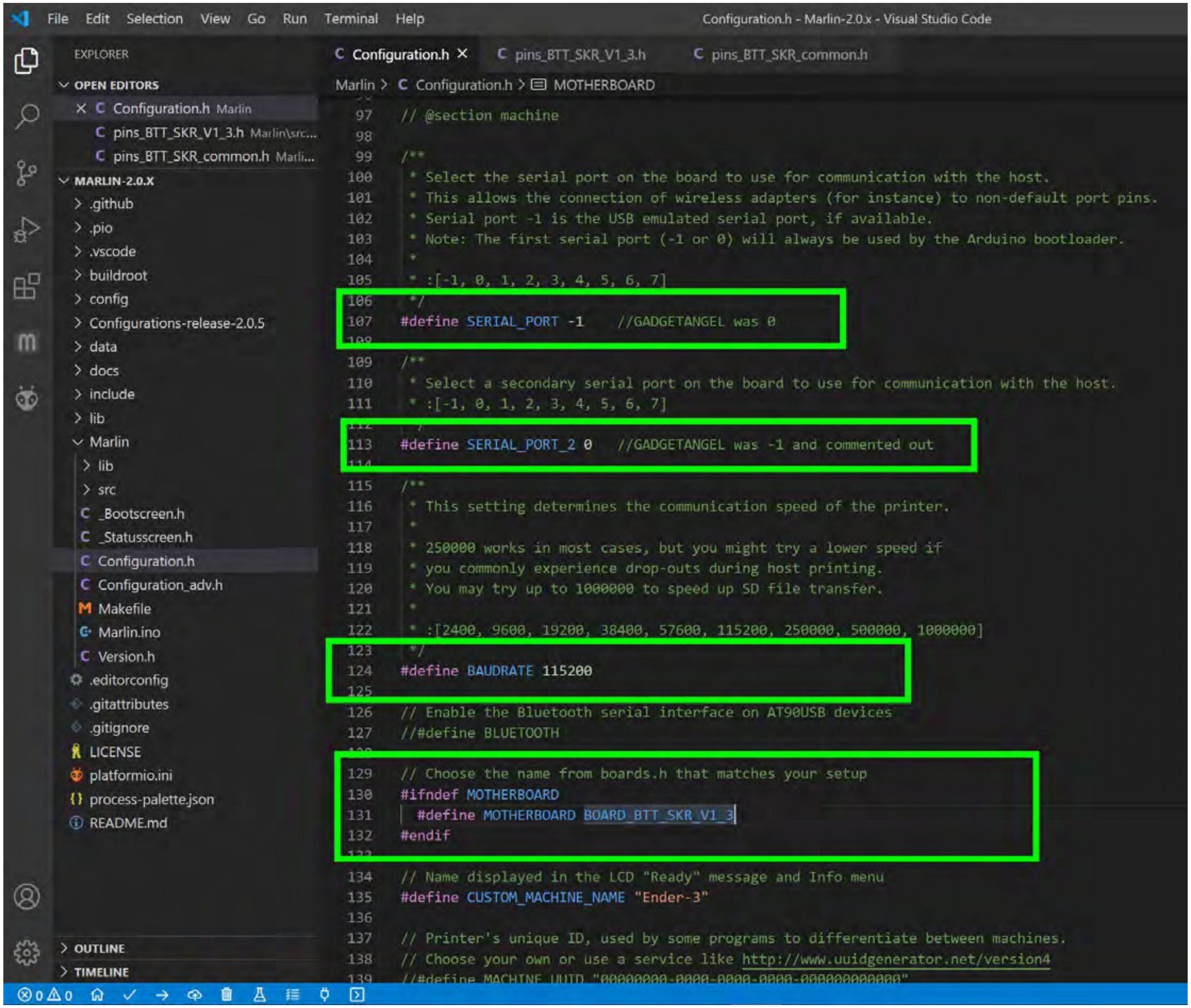


- Go to the next page.

APPENDIX C

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 **4 GREEN** boxes below.

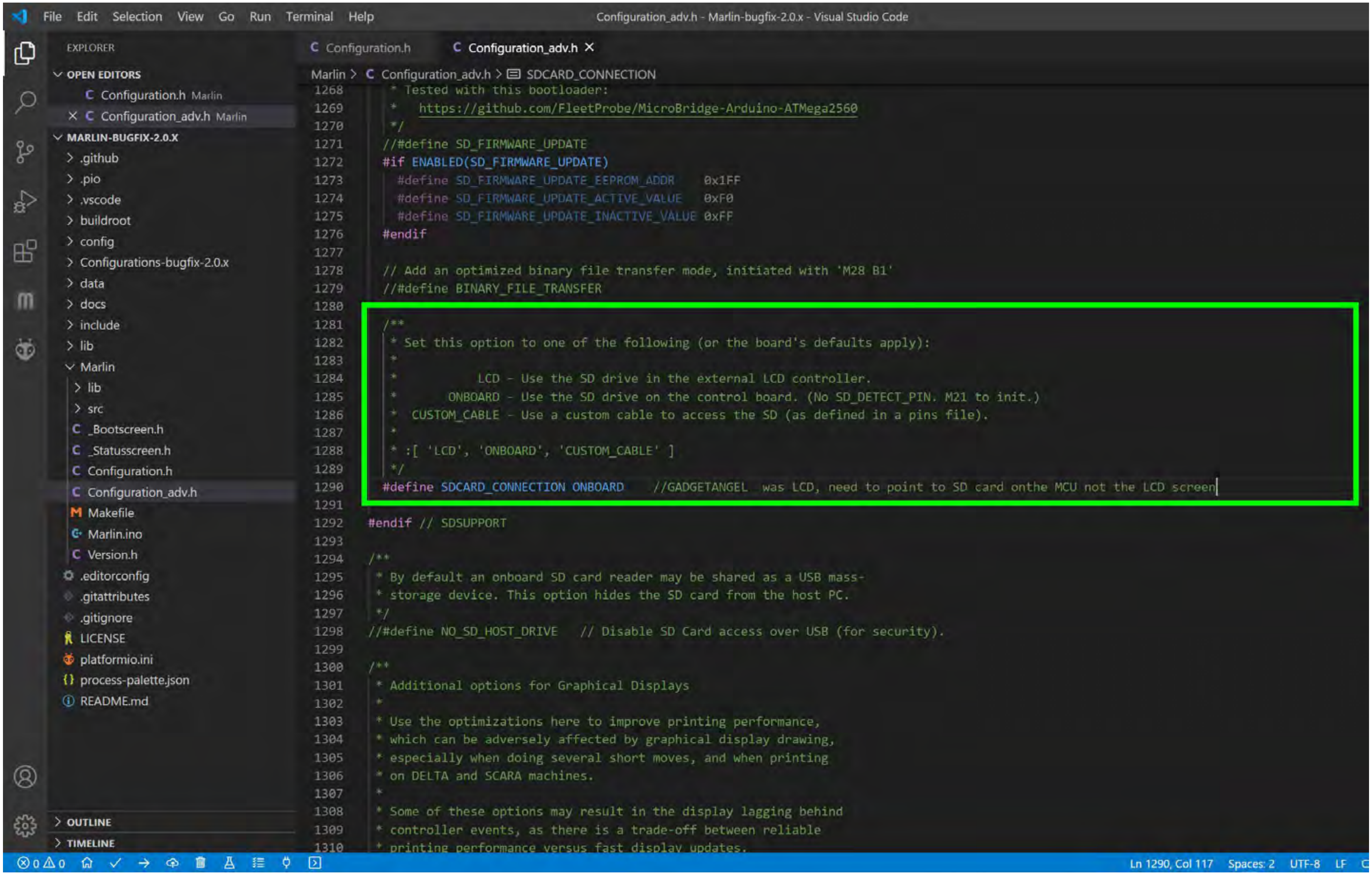


- You can set "BAUDRATE" to "115200" or "250000 ". Either setting will work but I have found that "115200" option works with any LCD that you choose to use.
- Go to the next page.

APPENDIX C

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

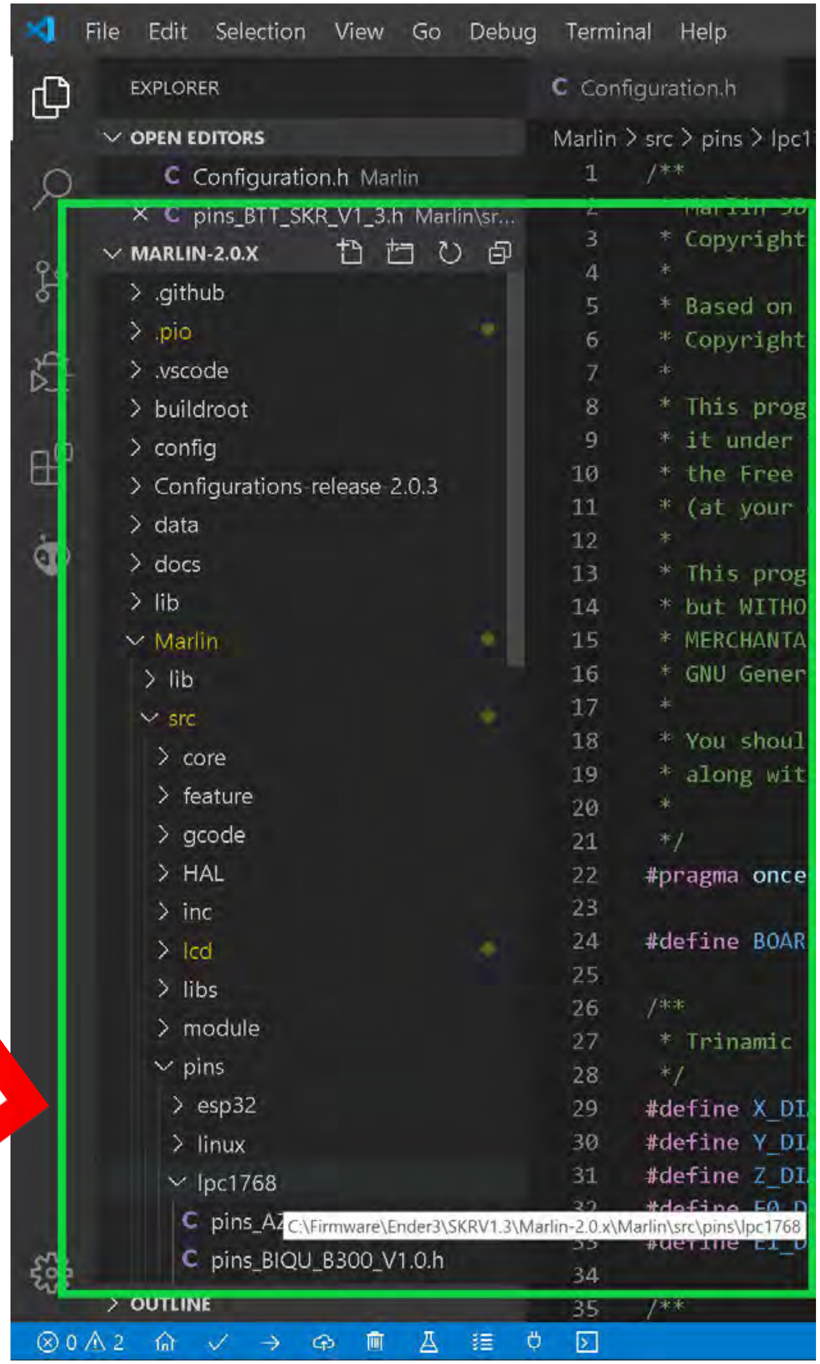
- **In Configuration_adv.h, enable the Marlin variable SDCARD_CONNECTION or that you delete the two forward slashes ("//") from the beginning of the line.**
- **In Configuration_adv.h, change the Marlin variable SDCARD_CONNECTION to ONBOARD**



- Go to the next page.

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

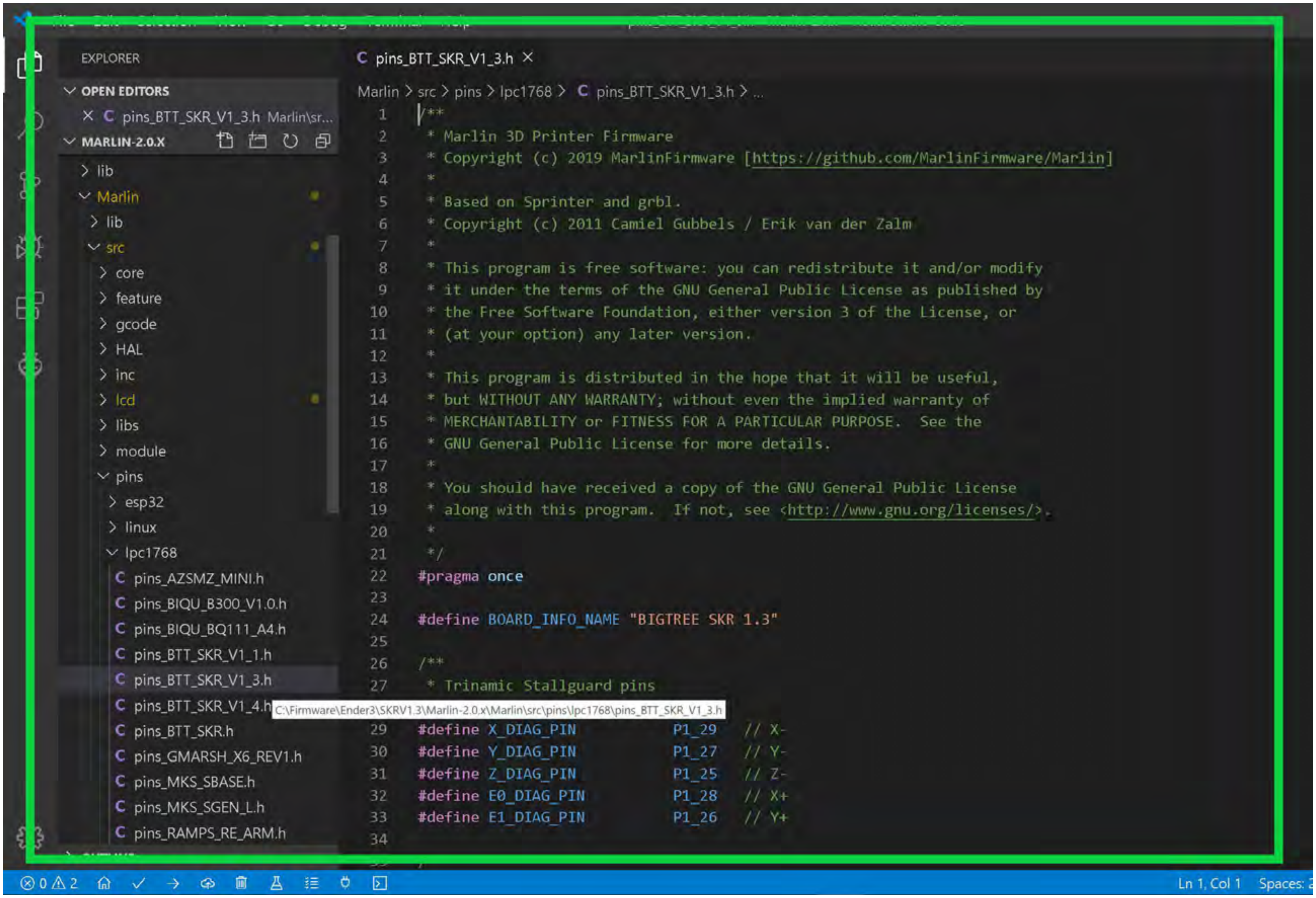
- Time to learn where the SKR V1.3 board's pins file is located: look at the left side and find the **C:\...\Marlin-2.0.X\Marlin\src\pins\lpc1768** subdirectory, as seen in the pictures below. Open the file, "pins_BTT_SKR_V1_3.h", by double clicking on it.

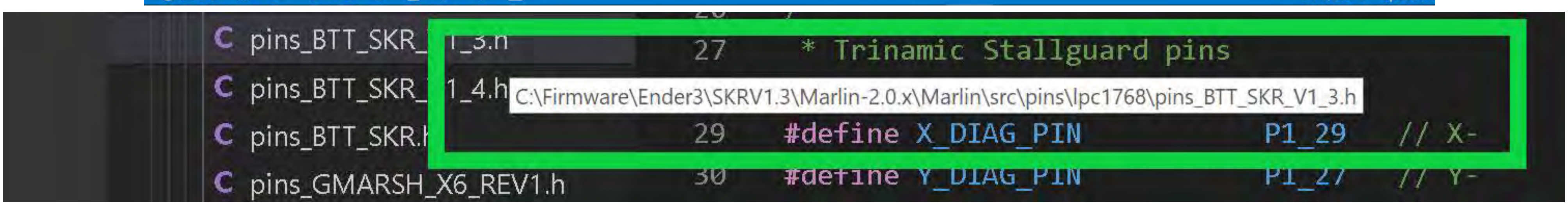


- Go to the next page.

APPENDIX C

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



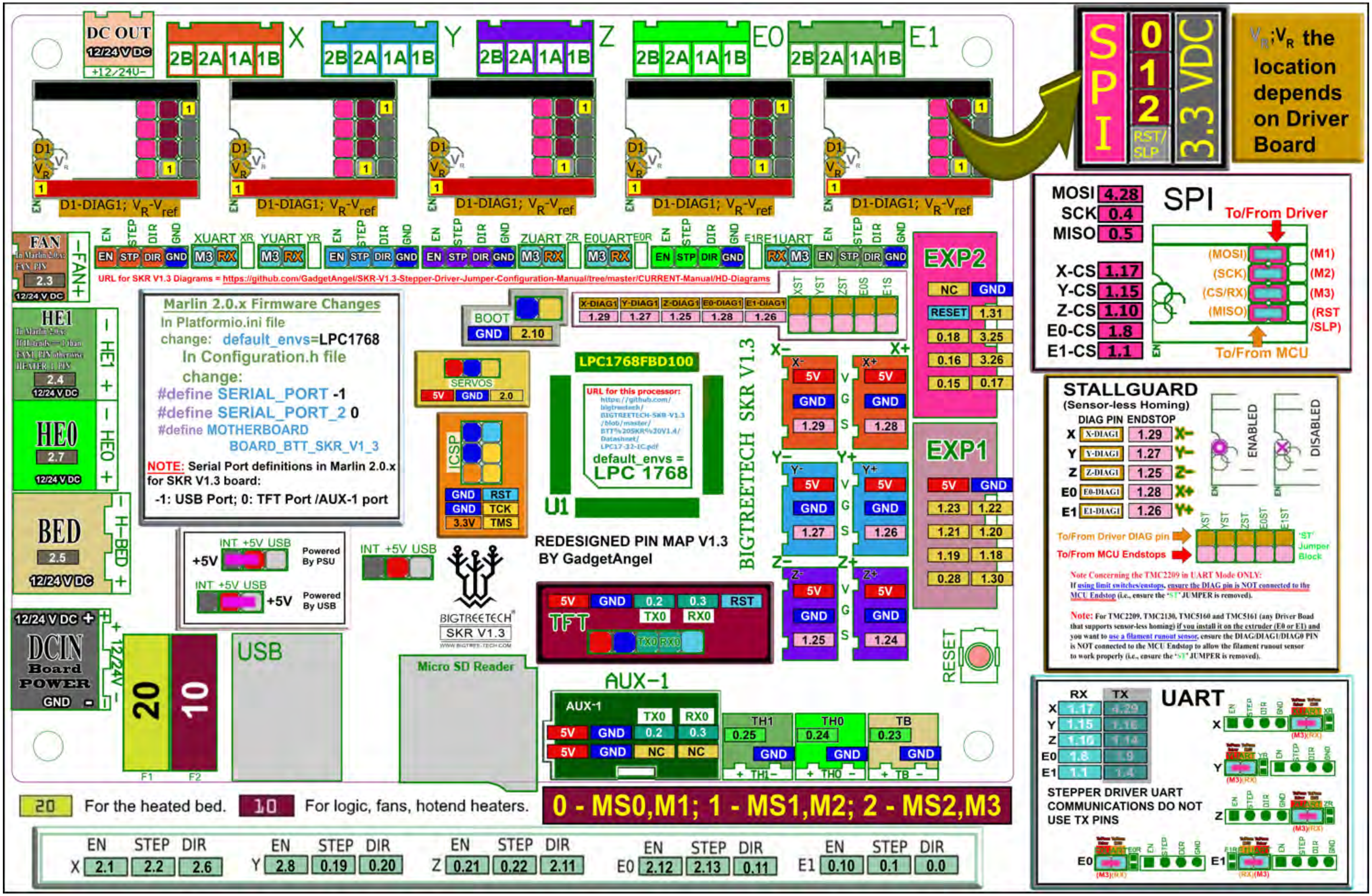


- Go to the next page.

APPENDIX C

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

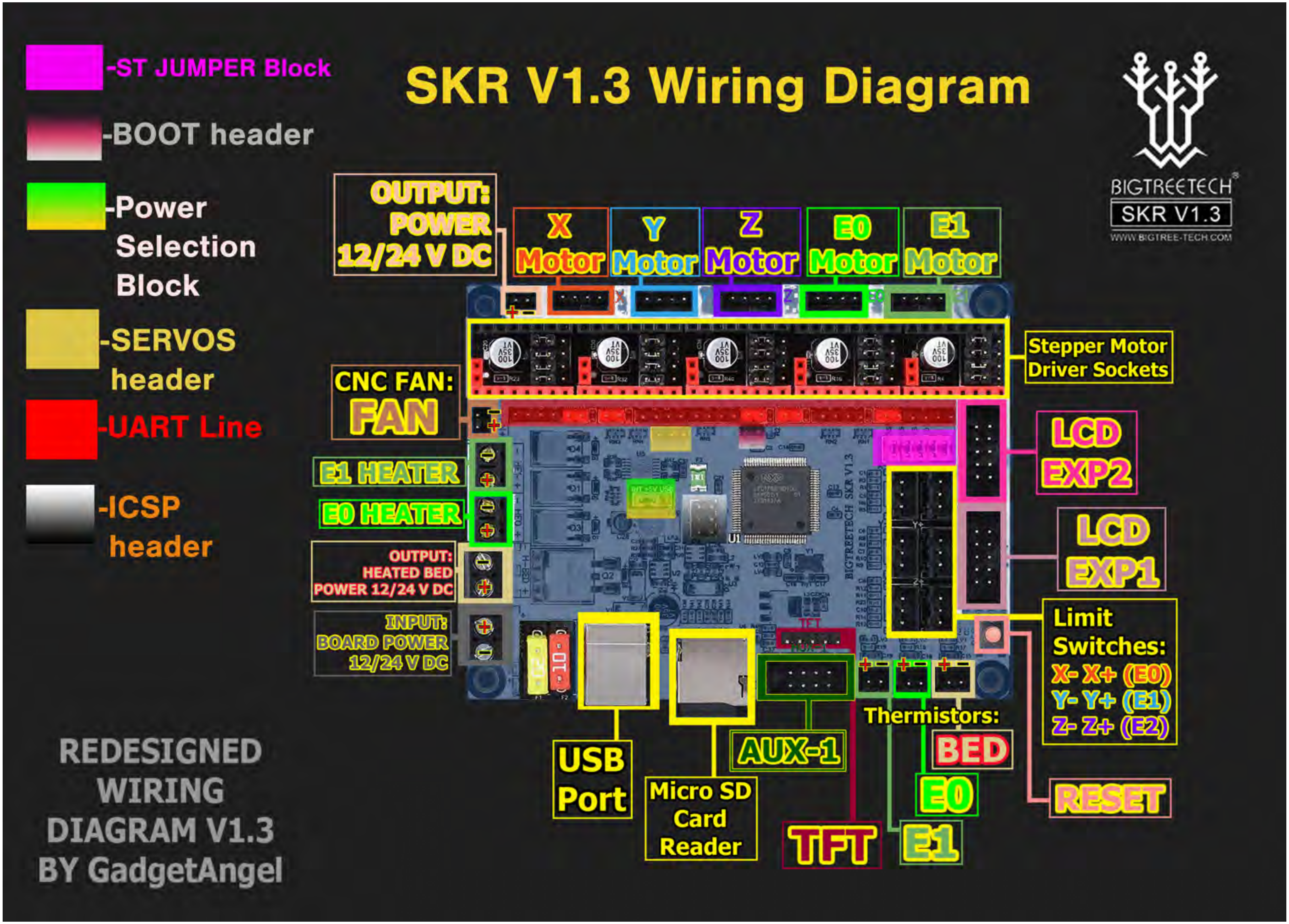
- We want to set the fan on the hot end to automatically turn on when the hot end starts to get hot. In the pins_BTT_SKR_V1_3.h file you will find all the pins that are defined for the board. They should all conform to the SKR V1.3 Pin diagram shown below. See the next page for a functional picture of the SKR V1.3 wiring diagram.



- 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. So, in this example I will use FAN1 port for my hot end cooling fan. In pins_BTT_SKR_V1_3.h file you will not find the FAN1 port defined, but you will see that this pins file refers to pins_BTT_SKR_common.h file. You can also see that there is not a FAN1 port on the SKR V1.3 board. Since I am only using one hot end, I can use HE1 port as FAN1 port. Usually a second hot end heater port can act as a CNC FAN port if a second hot end is not attached to your printer.
- Go to the next page.

APPENDIX C

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

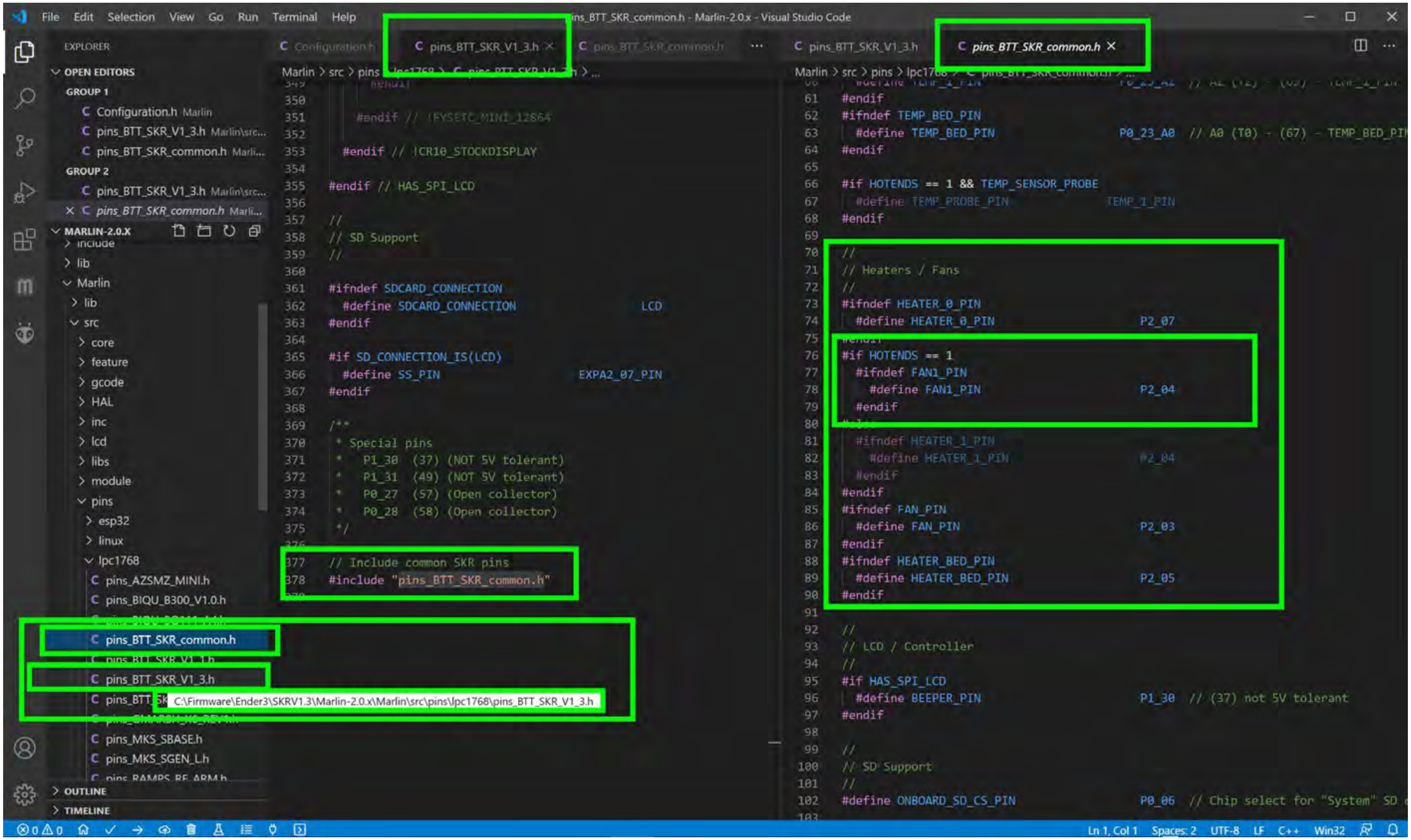


- Go to the next page.

APPENDIX C

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

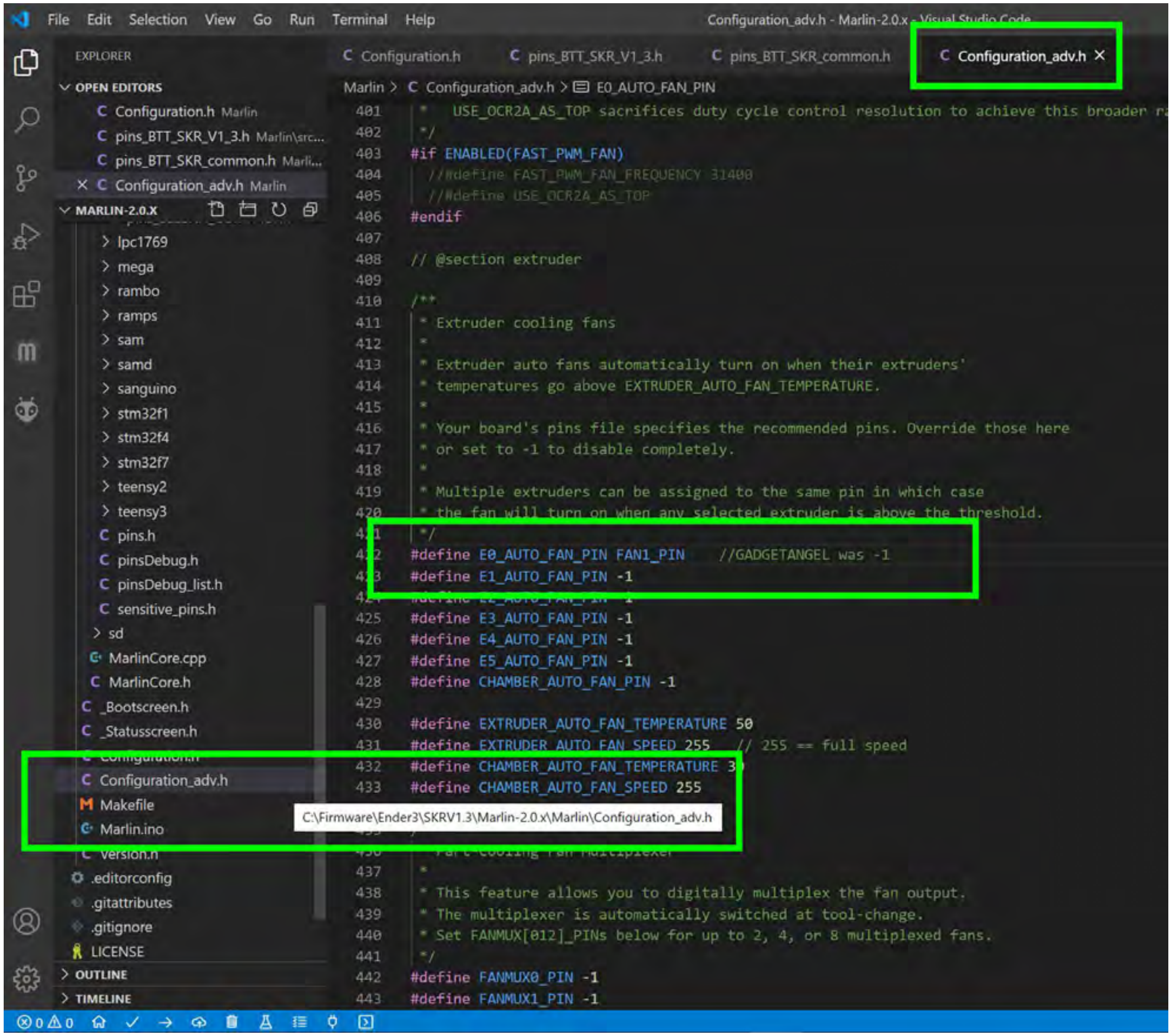
- **I plan on using the main FAN port or main CNC FAN port to control the fan in the electronics box. I prefer to have the electronics box fan to run only when the electronics need cooling. The main CNC FAN port is on "FAN_PIN" or "P2_03", as seen in "pins_BTT_SKR_common.h".**
- **When you go to pins_BTT_SKR_V1_3.h file, you will see that it refers to pins_BTT_SKR_common.h file. In pins_BTT_SKR_common.h file you will find the pin definitions for all the FAN ports available on the SKR V1.3 board. In this example I will use the HE1 port for my hot end cooling fan. In pins_BTT_SKR_common.h file we see FAN1 port (FAN1_PIN) is defined as P2_04. If you look at the SKR V1.3 color pin diagram you will see that HE1 port has 2.4 as its pin number (i.e. "P2_04" in Marlin firmware). So you can choose to copy "P2_04" or "FAN1_PIN".**



- Go to the next page.

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

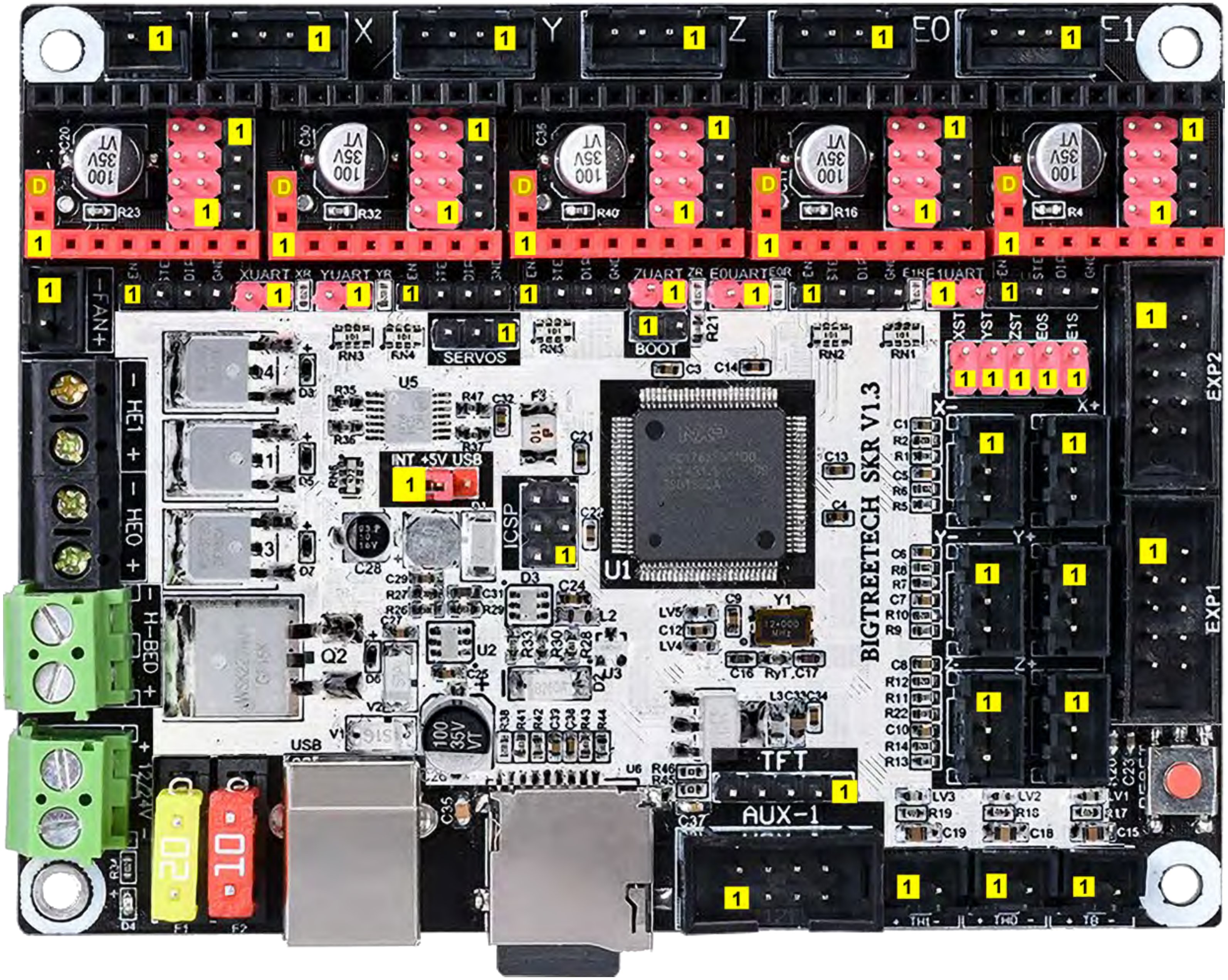
- To set the hot end fan, I will use FAN1_PIN and set "E0_AUTO_FAN_PIN" in the Configuration_adv.h file to FAN1_PIN, as seen in the picture below



- 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 V1.3, go to the next page.**

APPENDIX C - Color PIN 1 Diagram

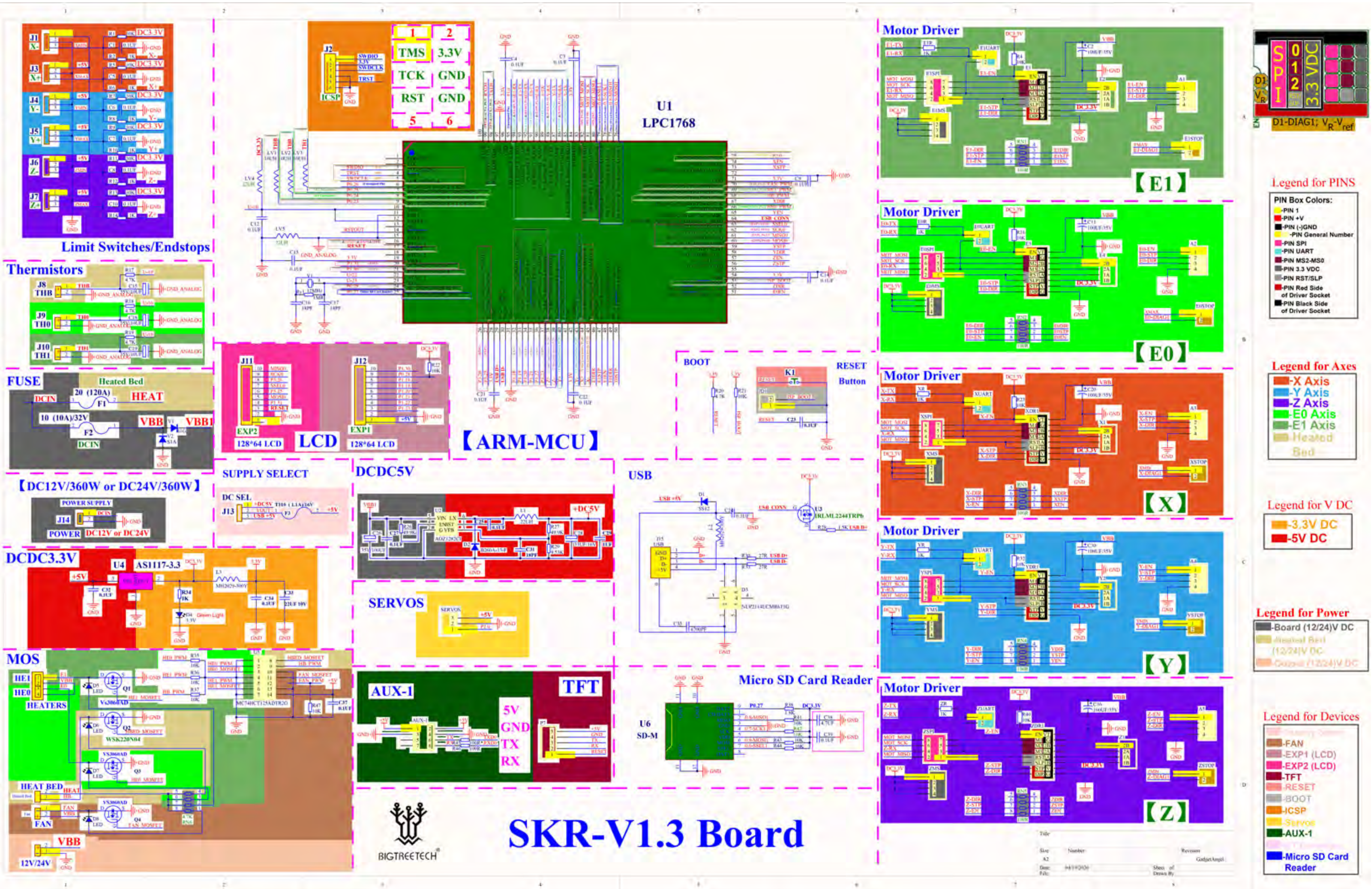


D -DIAG PIN

1 -PIN 1

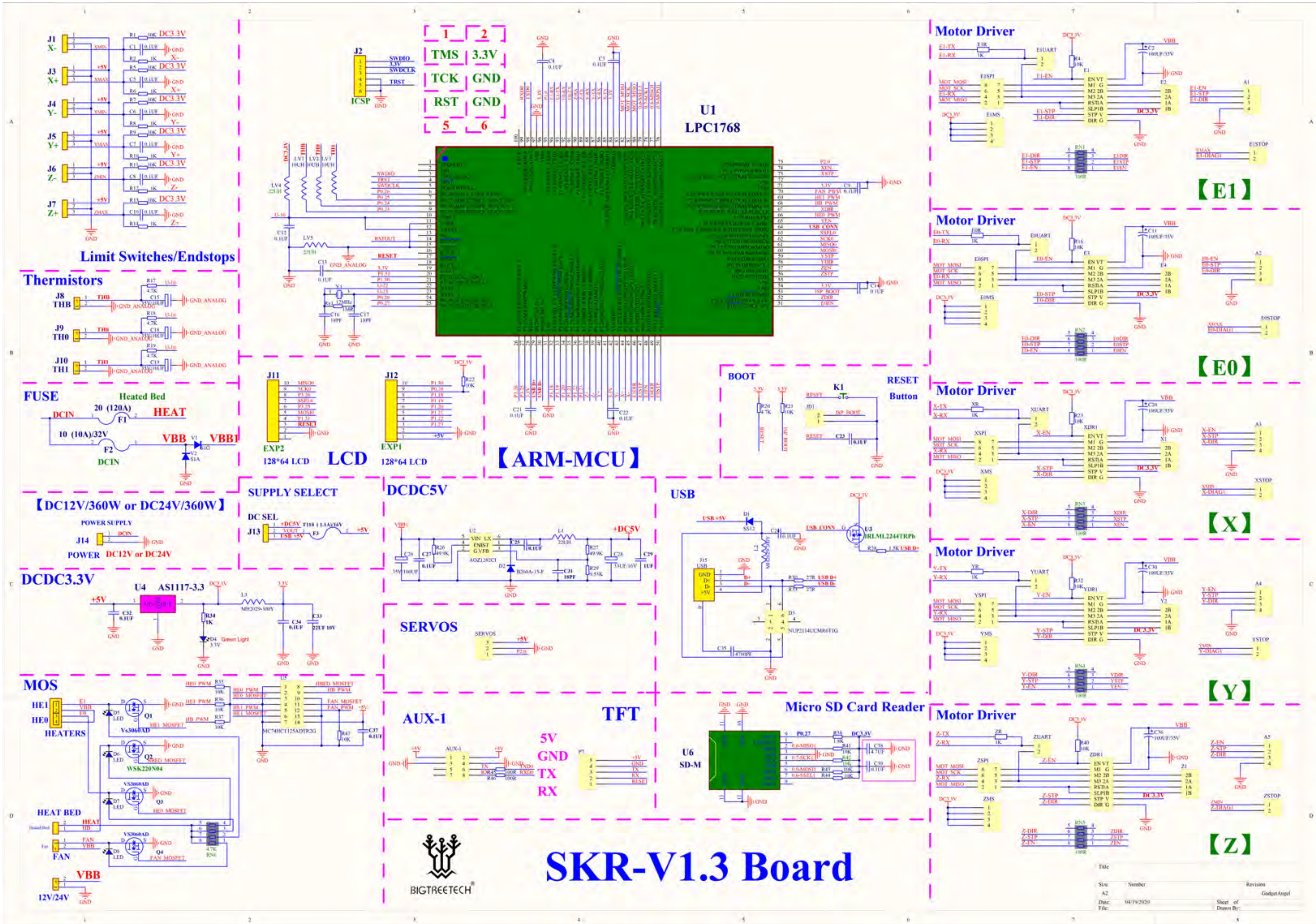
- To see my updated color schematic diagram for SKR V1.3, go to the next page.

APPENDIX C - Color Schematic Diagram



- To see my updated uncolored schematic diagram for SKR V1.3, go to the next page.

APPENDIX C - Uncolored Schematic Diagram



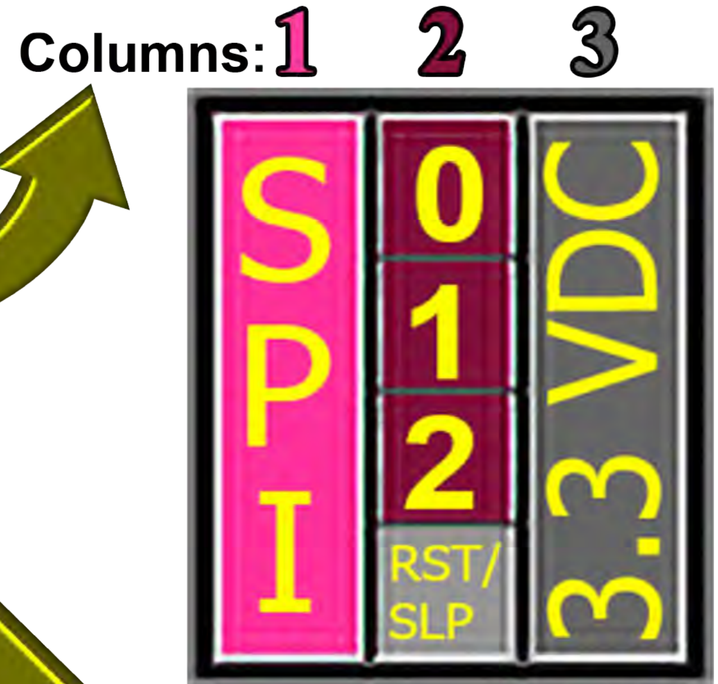
APPENDIX D

Legends for SKR V1.3 Stepper Driver Socket Representations

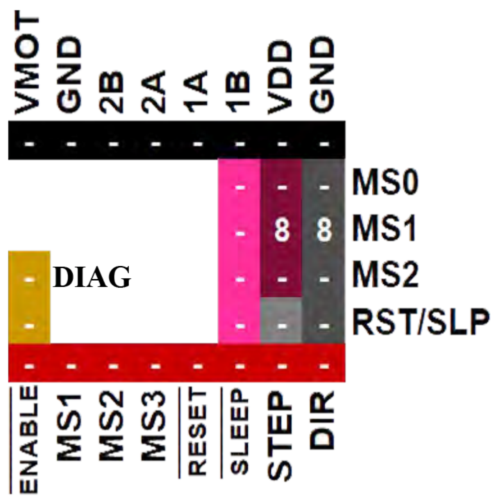
Column 1 is for SPI PINS

Column 2 is for MS0-MS2 PINS & RST/SLP PIN

Column 3 is for 3.3 V DC PINS



NOTICE: PIN labels are relative to stepper driver chip's carrier board, not the SKR V1.3 controller board. Double check driver compatibility before use. Number/Letter pairs denote required jumper(s). The example below indicates that you must place ONE jumper (8) across columns 2 and 3 in the MS1 row.

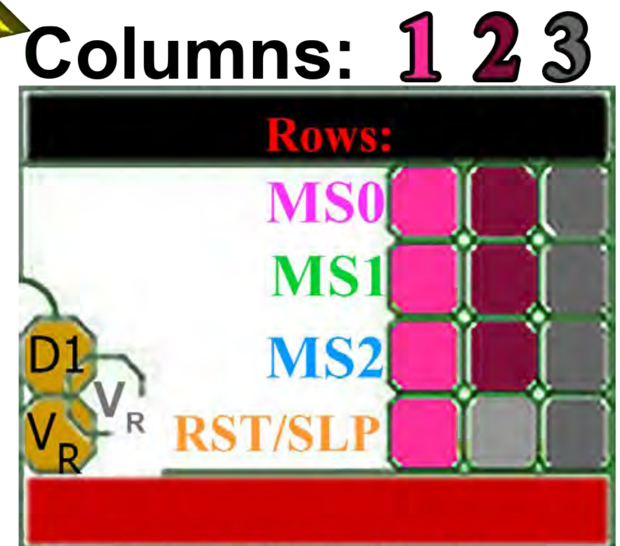


Legend for MS0-MS2 PINS:

0 - MS0,M1		0 - MS1,M1
1 - MS1,M2	OR	1 - MS2,M2
2 - MS2,M3		2 - MS3,M3

Legend for Driver Socket Rep.:

- \- \- means: No Jumper
- 7* - means: DuPont Jumper Cable to GND
- 8* - means: DuPont Jumper Cable to GND
- 7 7 or 8 8 or 9 9 means: Jumper set

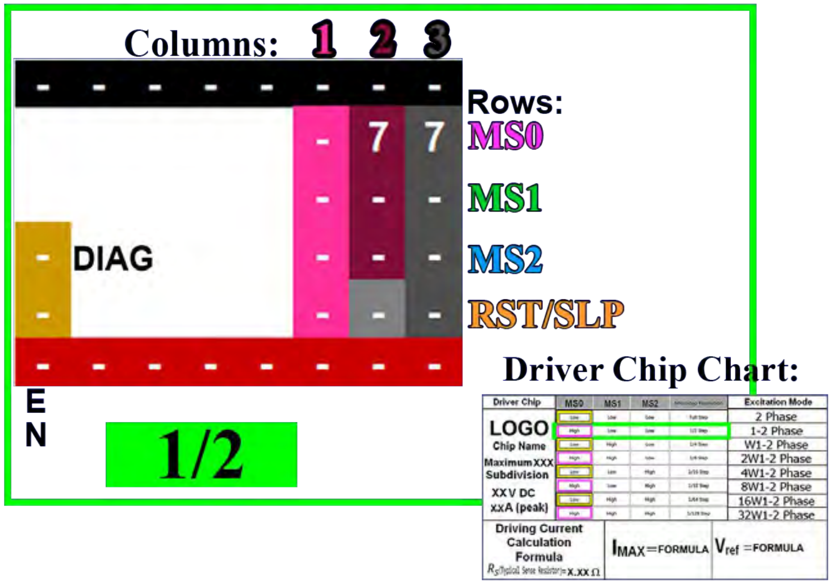


D1-DIAG1; V_R-V_{ref}

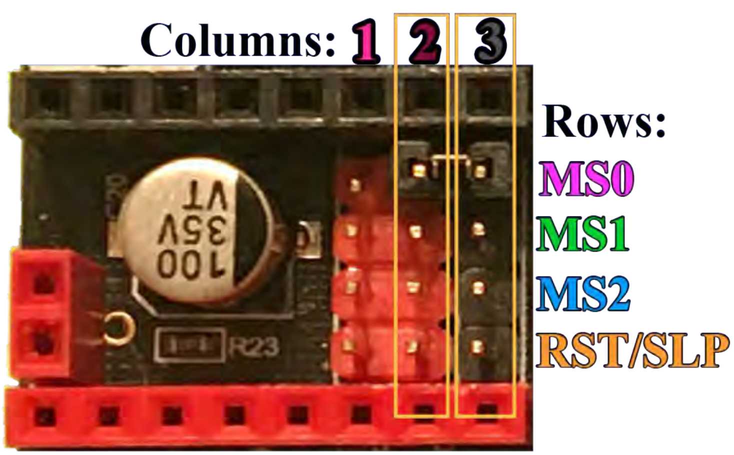
V_R;V_R -location depends on Driver Board

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS0 for Binary State Drivers:

Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: - - → Driver Chip Chart: Low

Meaning:

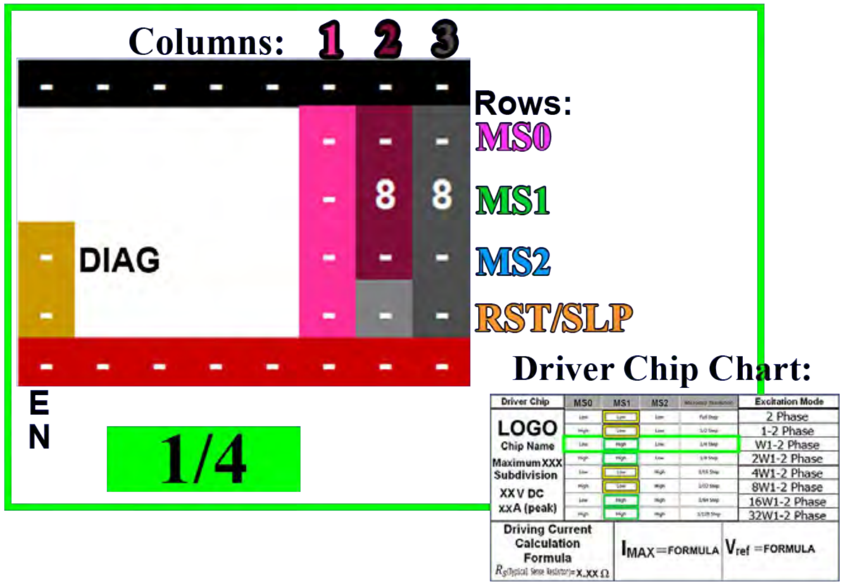
Driver Chip Chart:

High → set <u>Jumper</u> between column 2 and column 3 on the MS0 row

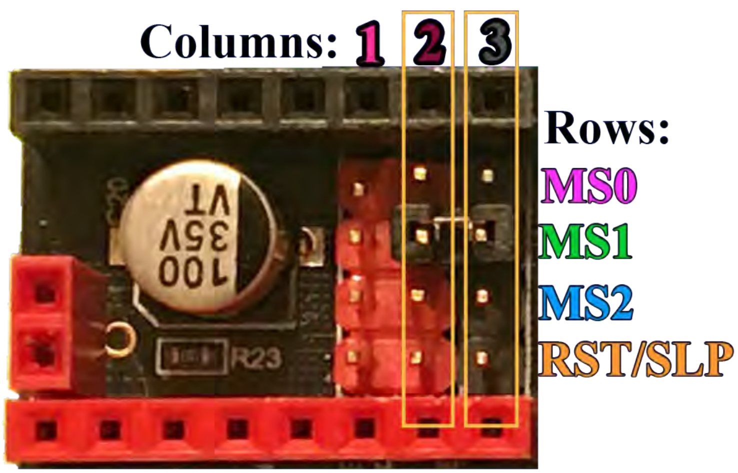
Low → No <u>Jumper</u> Set

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



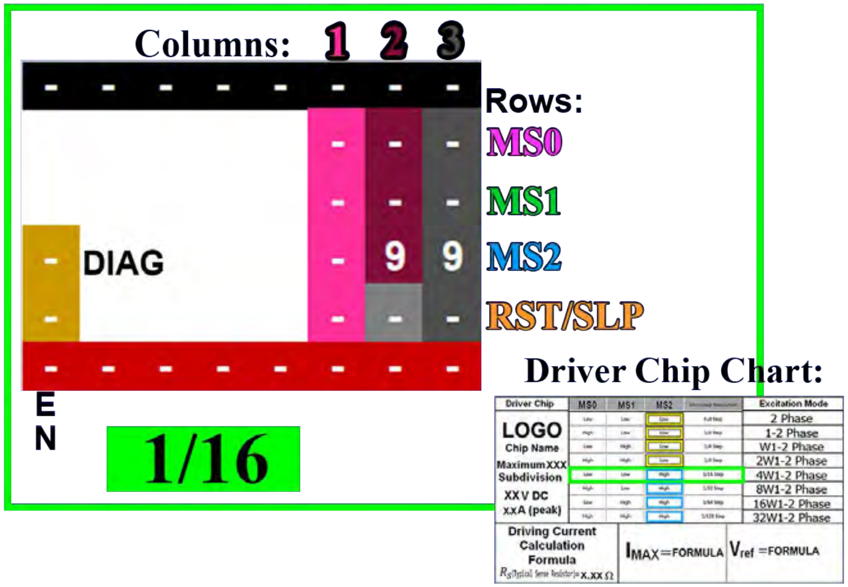
MS1 for Binary State Drivers:

Driver Socket Representation:		Driver Chip Chart:
- -	⇒	Low
8 8	⇒	High
- -	⇒	Low

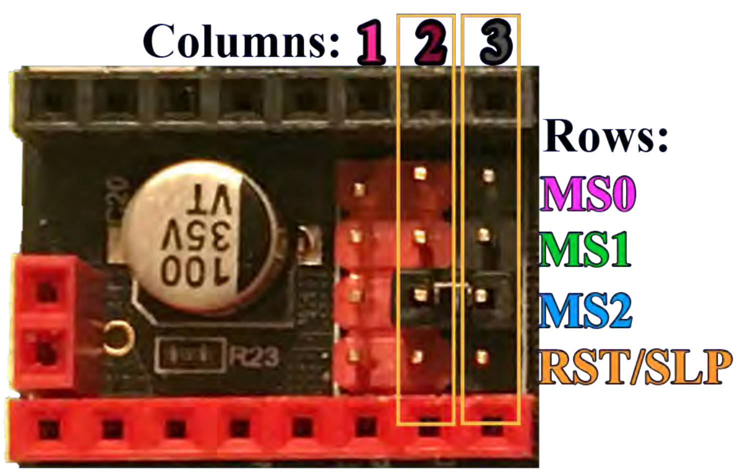
Driver Chip Chart:		Meaning:
High	⇒	set **Jumper** between column 2 and column 3 on the MS1 row
Low	⇒	No **Jumper** Set

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

Driver Socket Representation



High:



MS2 for Binary State Drivers:

Driver Socket Representation: - - → Driver Chip Chart: **Low**

Driver Socket Representation: 9 9 → Driver Chip Chart: **High**

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS2 row
Low →	No Jumper Set

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO Chip Name Maximum XXX Subdivision XX V DC xxA (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase

Driving Current Calculation Formula R_S(Typical Sense Resistor)= X.XX Ω	I_{MAX}=FORMULA	V_{ref} =FORMULA

Low ⇒ No Jumper

Rows:

MS0

MS1

MS2

RST/SLP

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

High ⇒ Jumper Set

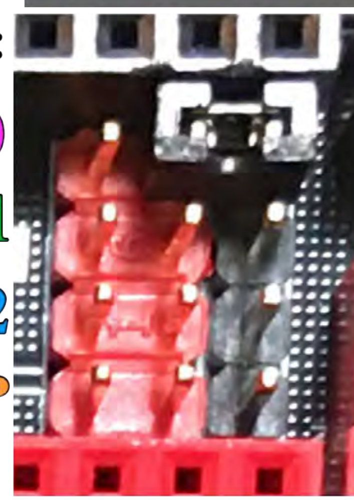
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

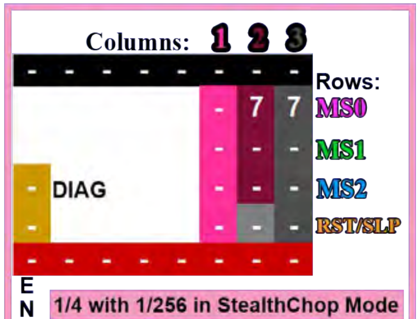
MS1 SET HIGH

MS2 SET HIGH

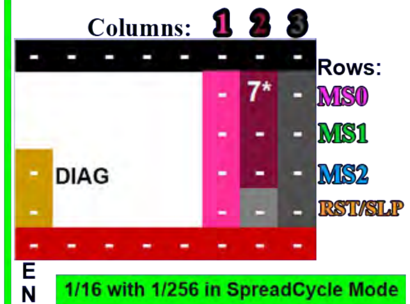
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

Driver Socket Representation



1/4 with 1/256 in StealthChop Mode



1/16 with 1/256 in SpreadCycle Mode

Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX}$ * 0.90			$V_{ref}=I_{MAX}$ See Appendix B. Use 50% to 90% as shown below: $V_{ref}=V_{ref}$ * 0.90	

- - → No Jumper set

MS0 for Tri State Drivers ONLY (TMC2100 & TMC2130):

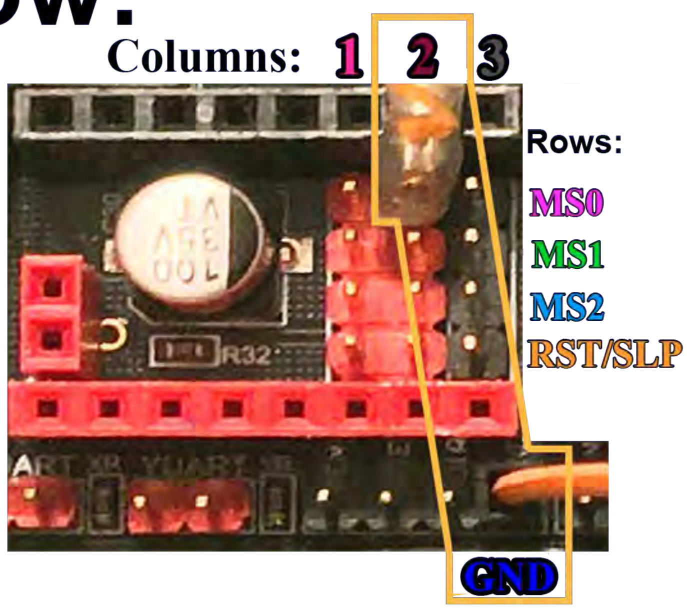
Driver Socket Representation: 7 7 → Driver Chip Chart: High

Driver Socket Representation: 7* - → Driver Chip Chart: Low

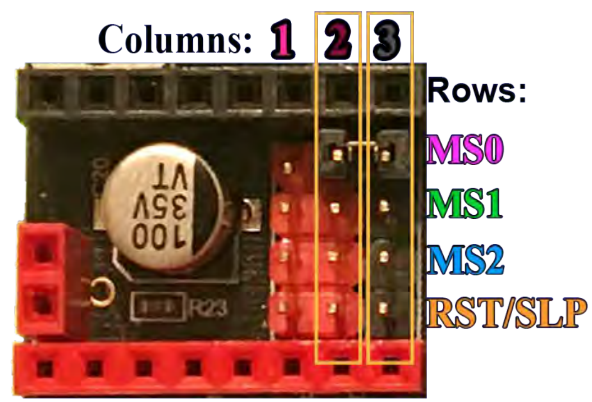
Driver Socket Representation: - - → Driver Chip Chart: Open

Driver Chip Chart:	Meaning:
High →	set Jumper between column 2 and column 3 on the MS0 row
Low →	Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)
Open →	No Jumper set

Low:

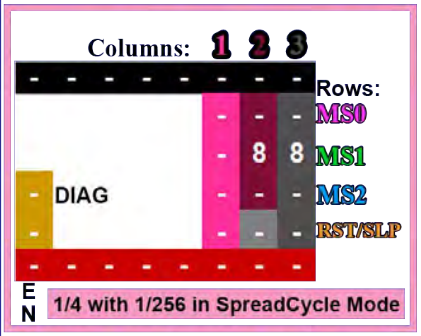


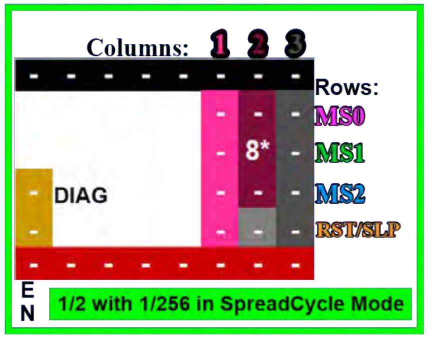
High:



SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

Driver Socket Representation



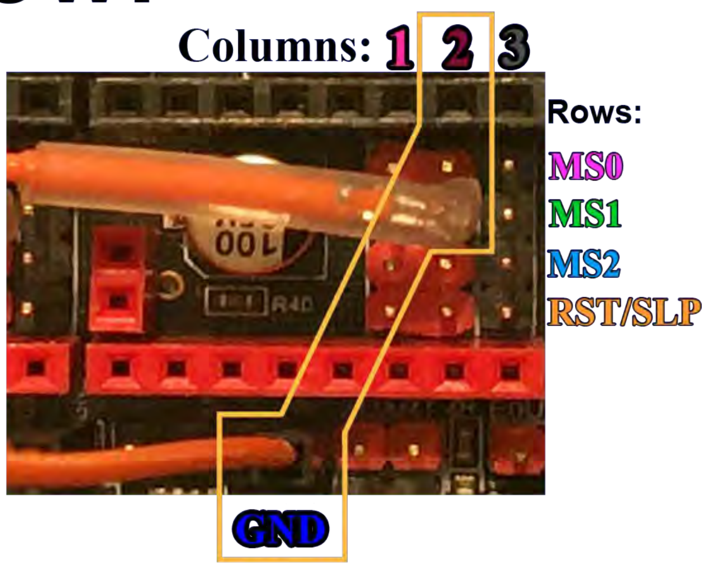


Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop
Driving Current Calculation Formula R_S(Typical Sense Resistor) = 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B #x. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX}$ * 0.90			$V_{ref}=I_{MAX}$ See Appendix B #x. Use 50% to 90% as shown below: $V_{ref}=V_{ref}$ * 0.90	

- - ⇨ No Jumper set

Low:



MS1 for Tri State Drivers ONLY (TMC2100 & TMC2130):

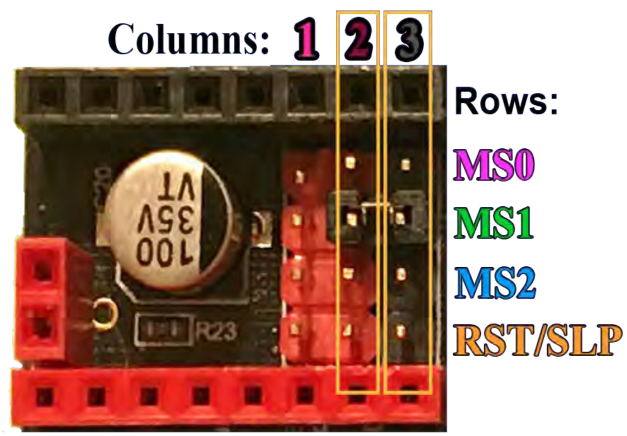
Driver Socket Represntation: 8 8 ⇨ Driver Chip Chart: **High**

Driver Socket Represntation: 8* - ⇨ Driver Chip Chart: **Low**

Driver Socket Represntation: - - ⇨ Driver Chip Chart: **Open**

Driver Chip Chart:	Meaning:
High ⇨	set Jumper between column 2 and column 3 on the MS1 row
Low ⇨	Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)
Open ⇨	No Jumper set

High:



SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers

Driver Chip Chart:

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

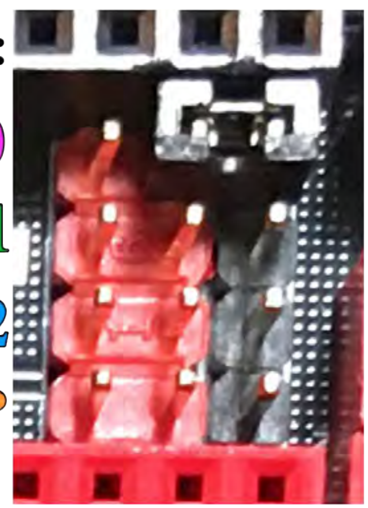
OPEN → No Jumper

Rows:
MS0
MS1
MS2
RST/SLP

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

High → Jumper Set

Rows:
MS0
MS1
MS2
RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

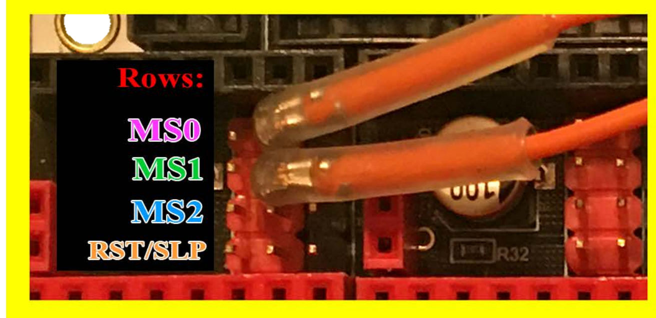
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

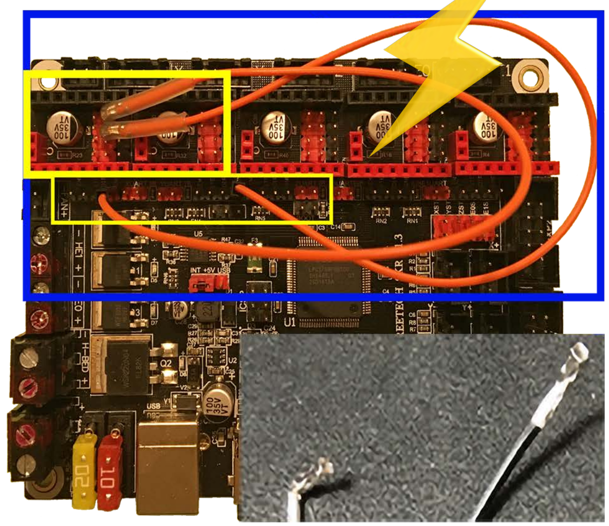
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers

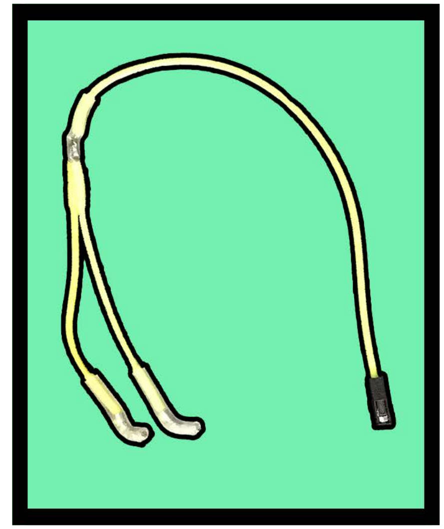
Driver Chip Chart:

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

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







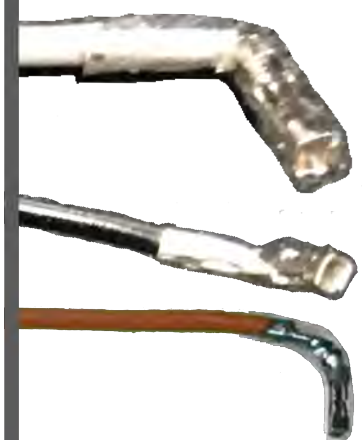
Low → **MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable**

Note: See next page for instructions on how to create a DuPont jumper cable.

How to Create a SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers

Remove DuPont cover from one end of a female to female DuPont jumper wire.

Place a single DuPont pin into exposed female end and bend the connector backwards 90 degrees.

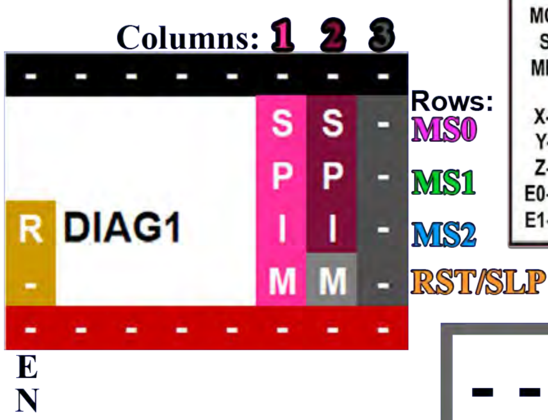


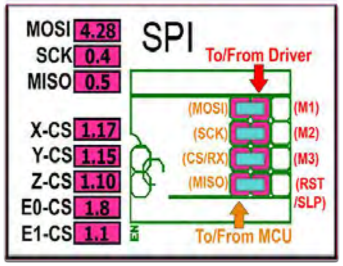
Since bending the connector weakens it, place a bead of solder at the 90 degree bend. I inserted a single DuPont pin inside the female connector to ensure that the connector stays clear of solder. After soldering is done, move the inserted DuPont pin around to ensure a functioning right angle female connector.

Place heat shrink over the exposed female DuPont connector to protect the exposed metal from causing shorts. Picture here is a finished jumper cable with a single DuPont pin inserted.

SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers

Driver Socket Representation





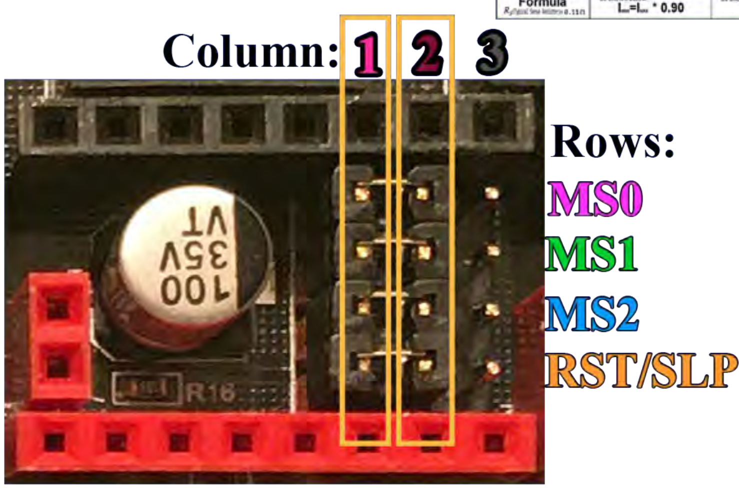
- - → No Jumper set

NOTE: UART PINS must not be set if SPI mode is ensbled.

SPIM:

Driver Chip Chart:

Driver Chip	
BIQU TMCxxxx SPI Mode	Steps are set inside of your Firmware
Driving Current Calculation Formula	$I_{MAX}=V_{ref}$; $V_{ref}=I_{MAX}$



SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

S S
P P
I I
M M

Meaning: Jumpers Set

Driver Socket Representation: S S → **Meaning:** set Jumper between column 1 and column 2 on the MS0 row

Driver Socket Representation: P P → **Meaning:** set Jumper between column 1 and column 2 on the MS1 row

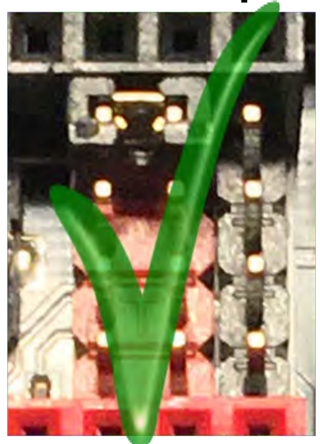
Driver Socket Representation: I I → **Meaning:** set Jumper between column 1 and column 2 on the MS2 row

Driver Socket Representation: M M → **Meaning:** set Jumper between column 1 and column 2 on the RST/SLP row

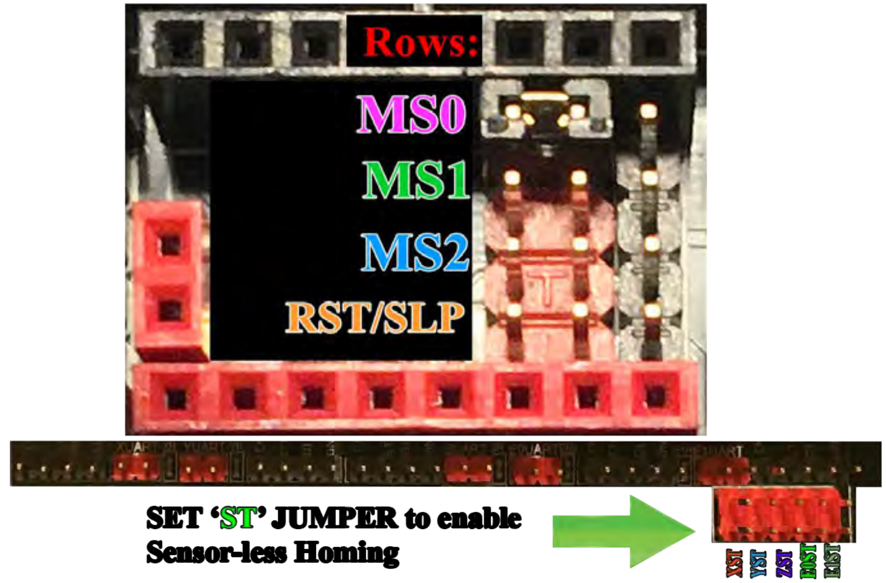
SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers

Example of the **WRONG** way to set the "S" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "S" jumper for SPI mode (left two columns of pins):

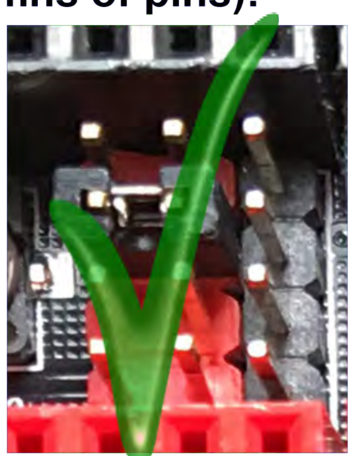


S S ⇒ Jumper set

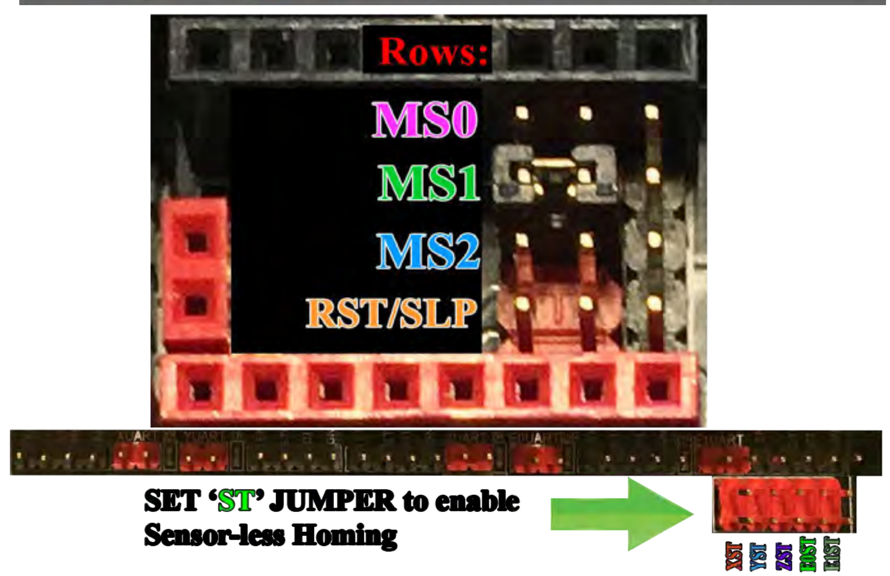


Example of the **WRONG** way to set the "P" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "P" jumper for SPI mode (left two columns of pins):



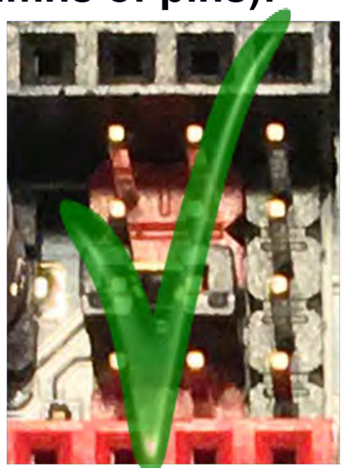
P P ⇒ Jumper set



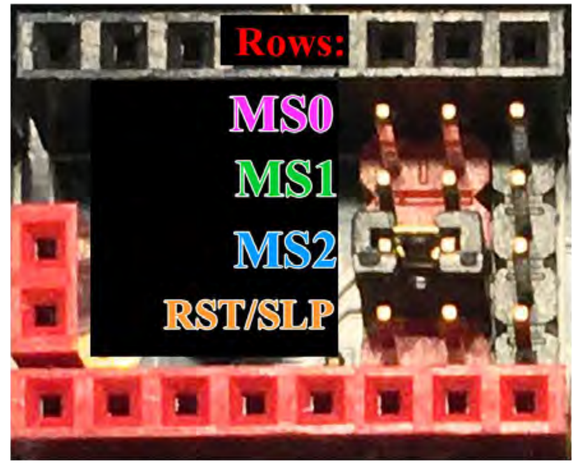
SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers

Example of the **WRONG** way to set the "I" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "I" jumper for SPI mode (left two columns of pins):



I I Jumper set

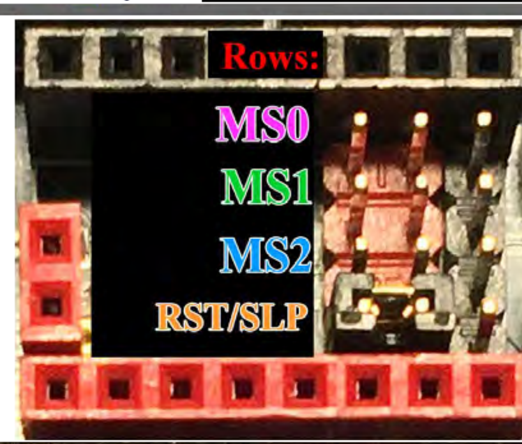


SET 'ST' JUMPER to enable Sensor-less Homing

Example of the **WRONG** way to set the "M" jumper for SPI mode (right two columns of pins):

Example of the **RIGHT** way to set the "M" jumper for SPI mode (left two columns of pins):

M M Jumper set

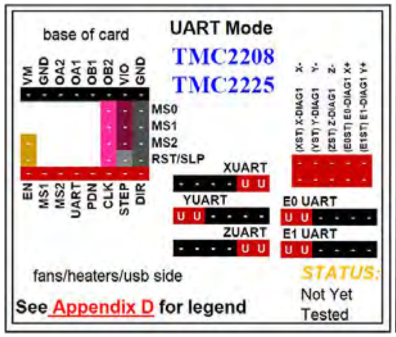


SET 'ST' JUMPER to enable Sensor-less Homing

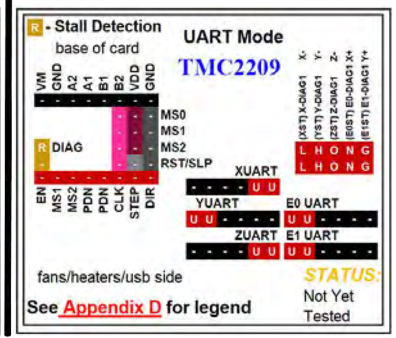
APPENDIX D

SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers

UART without stallGaurd™ feature



UART with stallGaurd™ feature



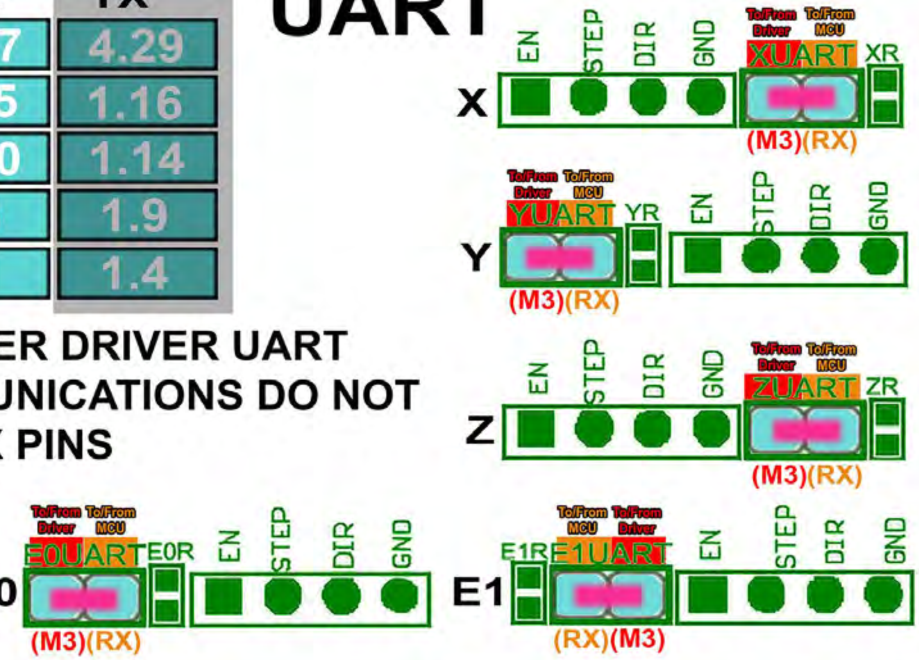
Driver Socket Representation: **Meaning:**

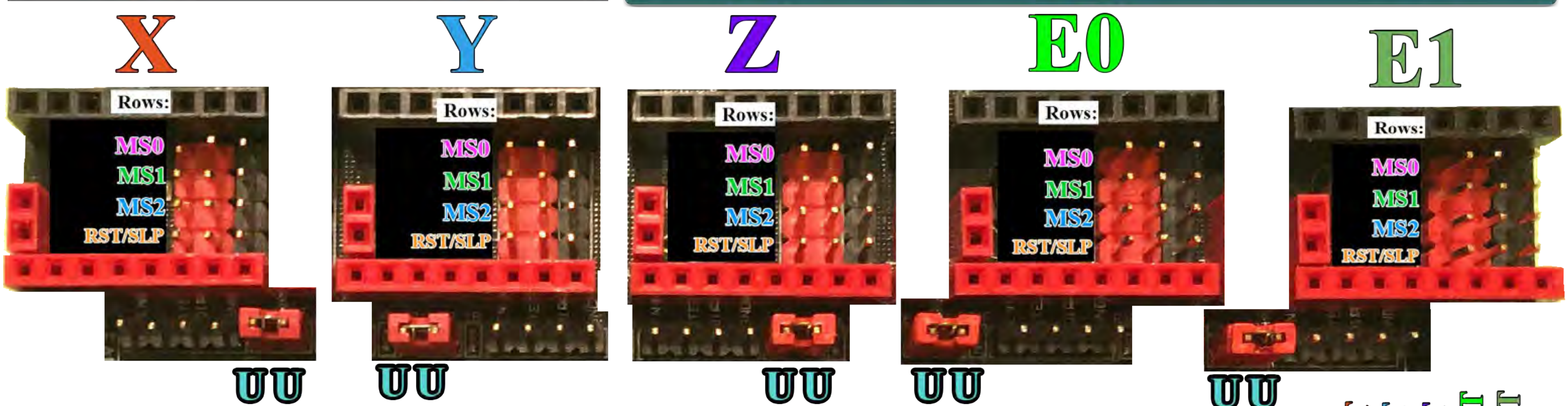
UU ⇒ Jumper set

UART

	RX	TX
X	1.17	4.29
Y	1.15	1.16
Z	1.10	1.14
E0	1.8	1.9
E1	1.1	1.4

STEPPER DRIVER UART COMMUNICATIONS DO NOT USE TX PINS

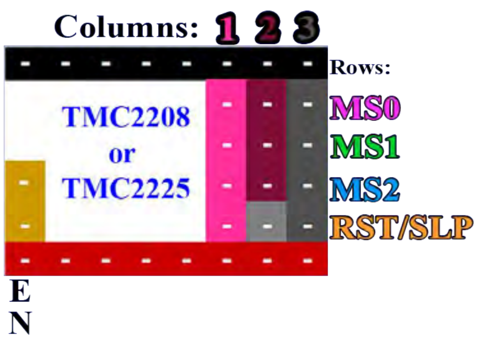


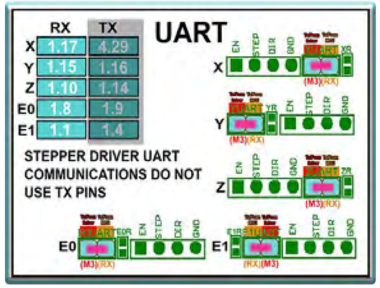


SET 'ST' JUMPER(s) to enable Sensor-less Homing -- for drivers with stallGaurd™ feature

SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers

Driver Socket Representation



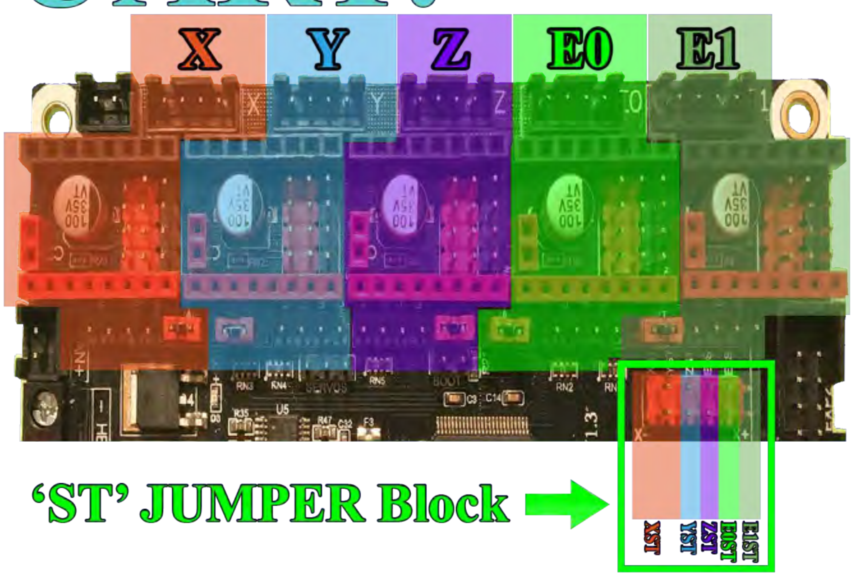


- - → No Jumper set

NOTE: SPI PINS must not be set if UART mode is ensbled.

REMOVE 'ST' JUMPER to disable Sensor-less Homing

UART:



'ST' JUMPER Block

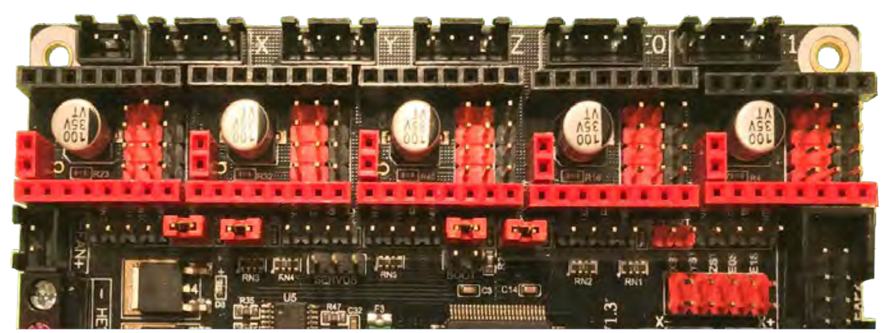
UART Capable Drivers without stallGaurd™ feature: TMC2208, and TMC2225

Driver Socket Representation:	Meaning:
U U →	Jumper Set

Driver Socket Representation:	Meaning:
U U →	set Jumper so that both U PINS are covered by the Jumper

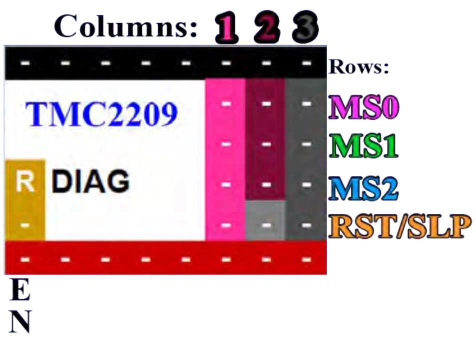
Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS REMOVED. All other PINS need to be empty for UART mode to work properly.

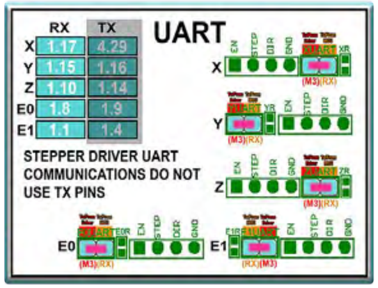
Here is an example of only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS removed for ALL Axes:



SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers

Driver Socket Representation



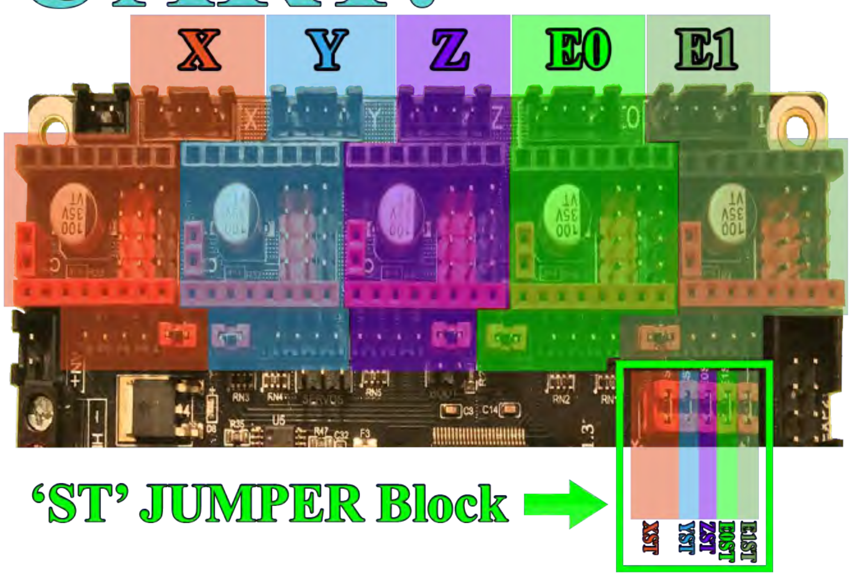


- - → No Jumper set

NOTE: SPI PINS must not be set if UART mode is ensbled.

SET 'ST' JUMPER to enable Sensor-less Homing →

UART:



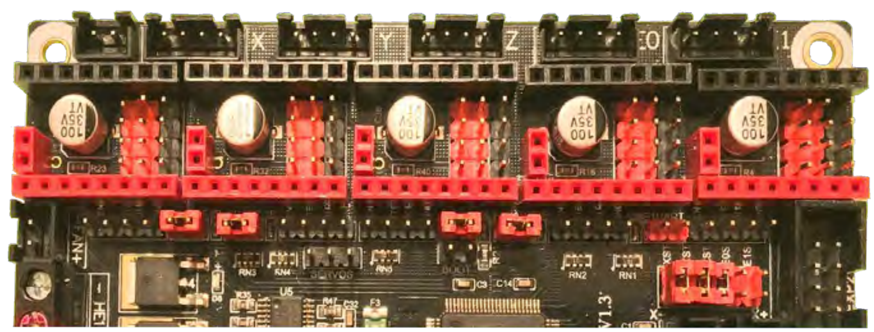
'ST' JUMPER Block →

UART Capable Drivers with stallGaurd™ feature: TMC2209

Driver Socket Representation:	Meaning:
U U →	Jumper Set
U U →	set Jumper so that both U PINS are covered by the Jumper

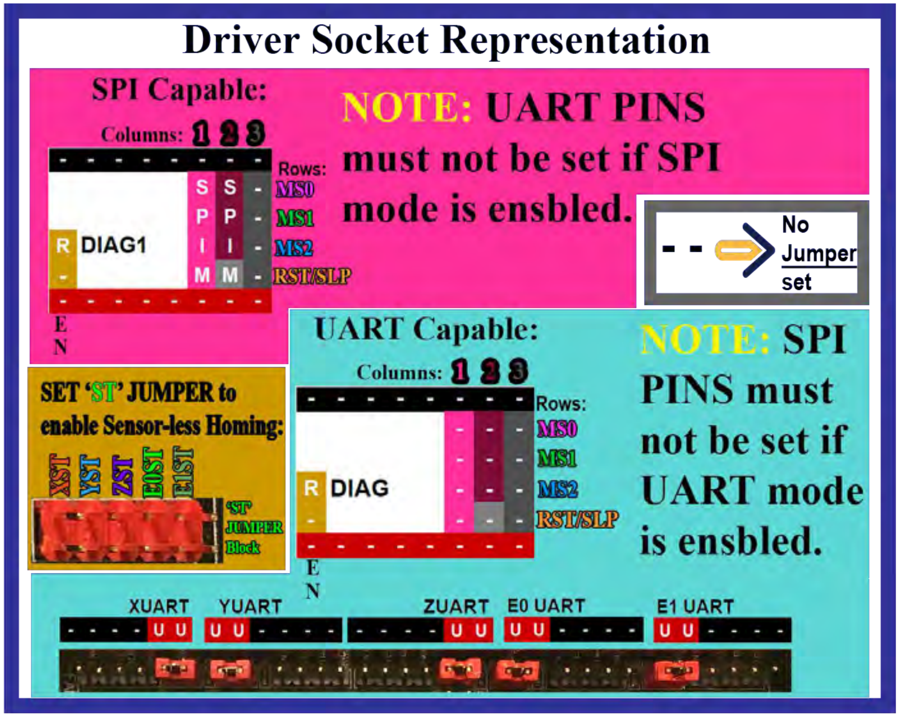
Each Axis has its own UART PINS (UU). You can set as many axes as you need. The UART: picture shows all axes having UART enabled AND all 'ST' JUMPERS SET. All other PINS need to be empty for UART mode to work properly.

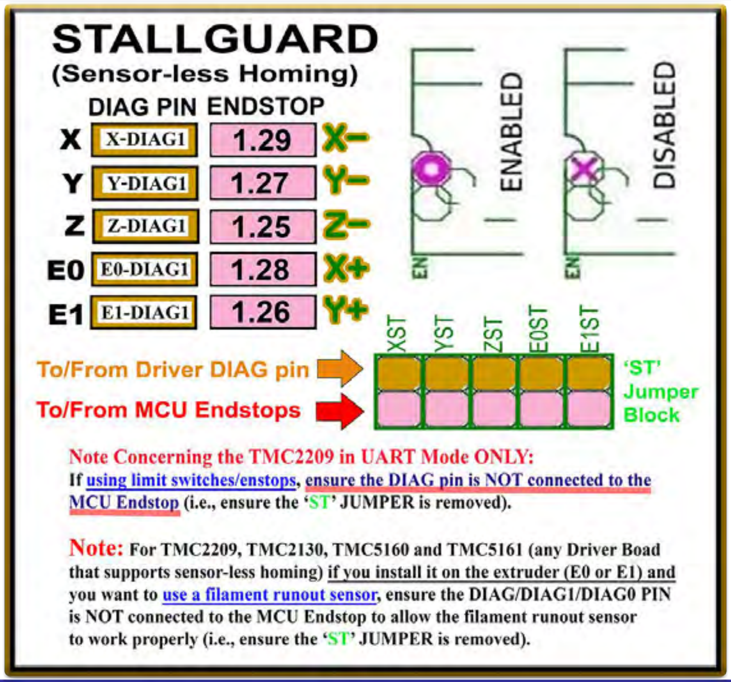
Here is an example of (TMC2209 UART mode) only the X, Y, Z, and E0 axes having UART enabled AND 'ST' JUMPERS set for X, Y, and Z Axes:



APPENDIX D

SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers

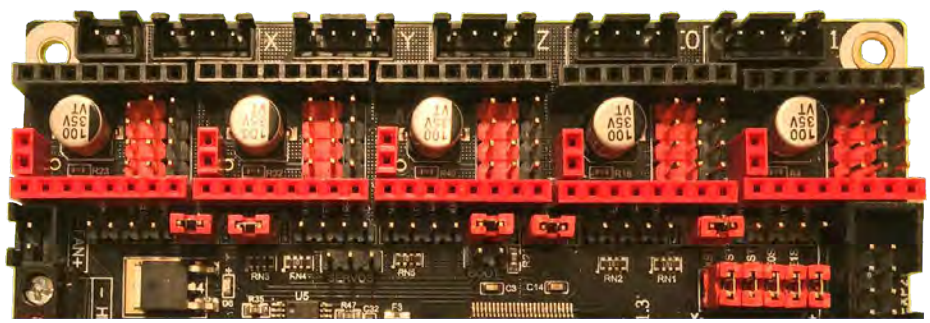




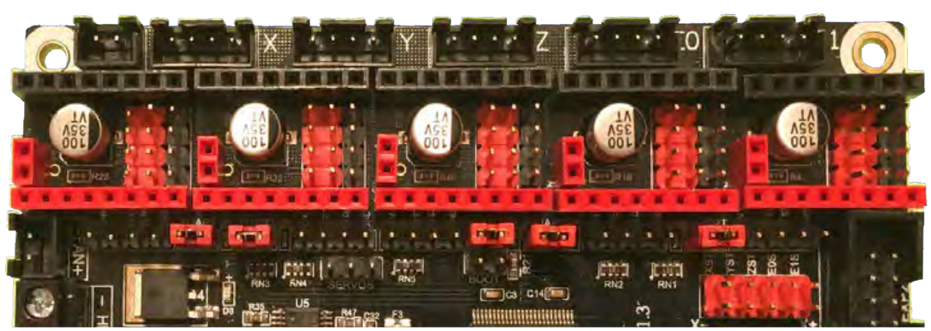
Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161
UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are SET for ALL Axes:

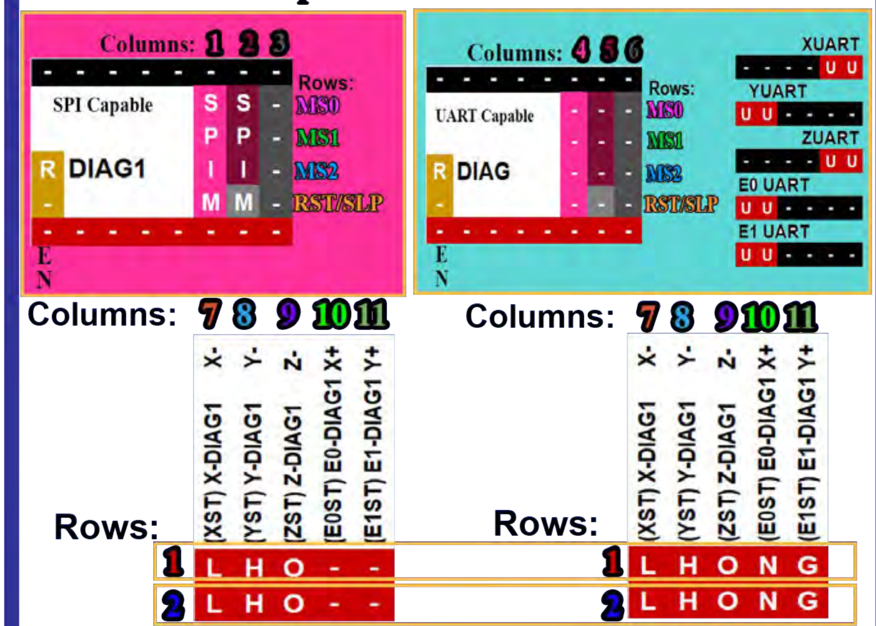


If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERs are REMOVED for ALL Axes:

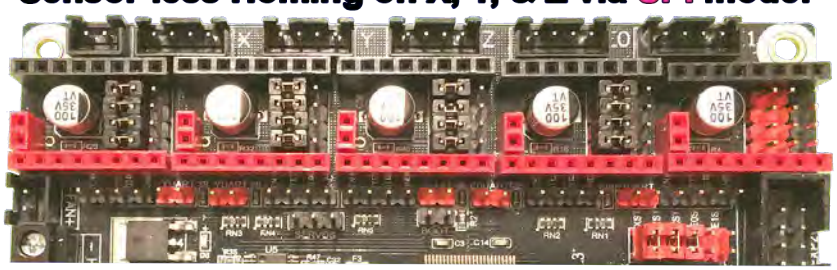


SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers

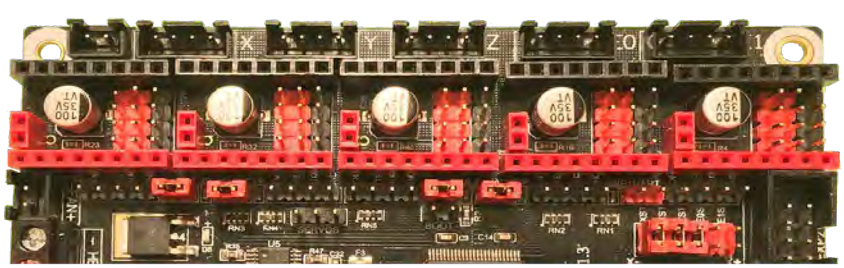
Driver Socket Representation of 'ST' JUMPER Block



Sensor-less Homing on X, Y, & Z via SPI mode:



Sensor-less Homing on X, Y, & Z via UART mode:



Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

- L L → X MIN Endstop to enable Sensor-less Homing
- H H → Y MIN Endstop to enable Sensor-less Homing
- O O → Z MIN Endstop to enable Sensor-less Homing
- N N → X MAX Endstop to enable Sensor-less Homing
- G G → Y MAX Endstop to enable Sensor-less Homing
- \- \- → No Jumper Set

Driver Socket Representation:	Meaning:
L L	set Jumper between row 1 and row 2 in column 7
H H	set Jumper between row 1 and row 2 in column 8
O O	set Jumper between row 1 and row 2 in column 9
N N	set Jumper between row 1 and row 2 in column 10
G G	set Jumper between row 1 and row 2 in column 11

SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers

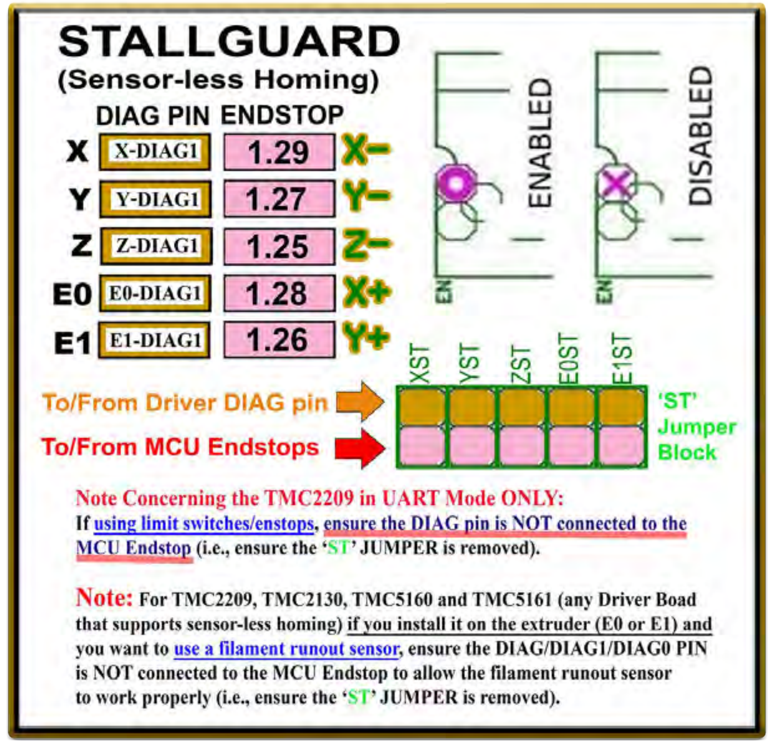
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capabliltiy of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therfore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



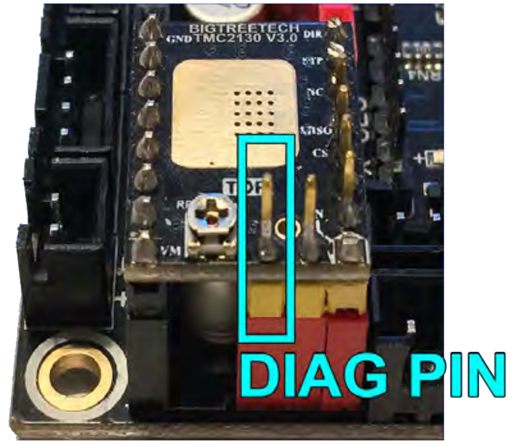
If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

- **Extruder Axis (E0 or E1)**
- **Z Axis, if a BLTouch is used to Home instead of a physical endstop**

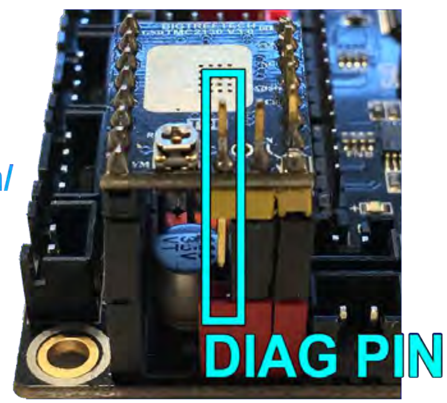
SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers

To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.



To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

Link to stackable header pins:
https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P

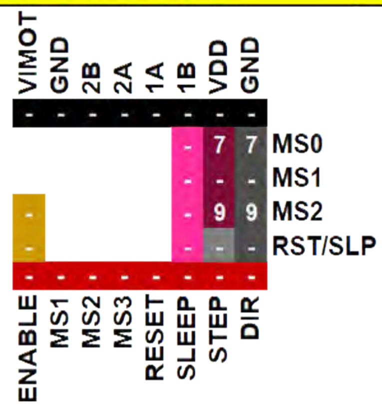


APPENDIX D

Examples of Driver Socket Representations

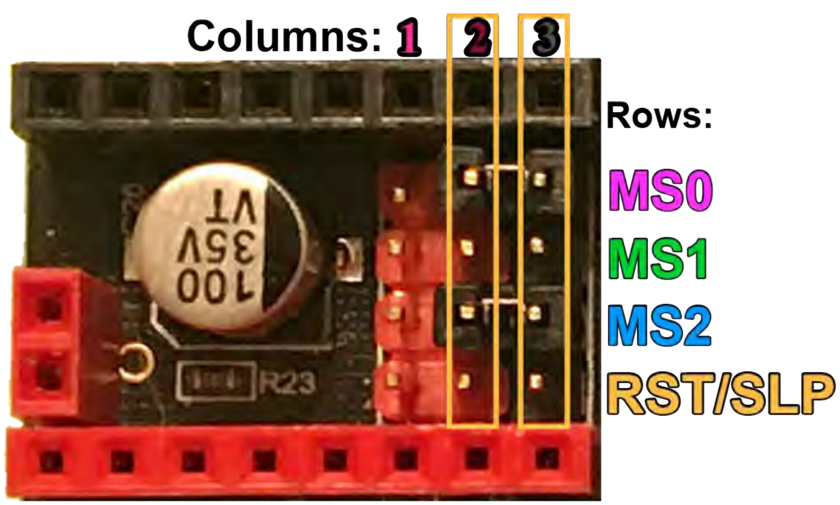
Example 1 (LV8729 Driver Board; Binary State Driver) for SKR V1.3 Driver Socket Representation:

1 / 32



Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Microstep Resolution
BIQU® LV8729	Low	Low	Low	Full Step
	High	Low	Low	1/2 Step
	High	Low	High	1/32 Step



Driver Socket Representation: 7 7 → Driver Chip Chart: High

MS0 PIN:

High → set Jumper between column 2 and 3 on MS0 row

Driver Socket Representation: - - → Driver Chip Chart: Low

MS1 PIN:

Low → No Jumper Set

Driver Socket Representation: 9 9 → Driver Chip Chart: High

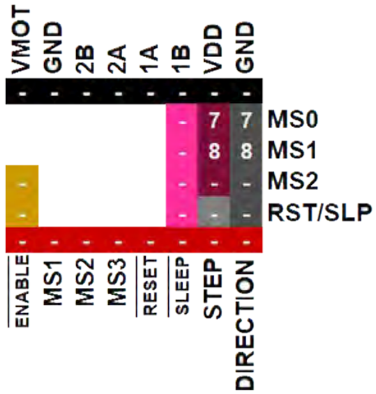
MS2 PIN:

High → set Jumper between column 2 and 3 on MS2 row

APPENDIX D

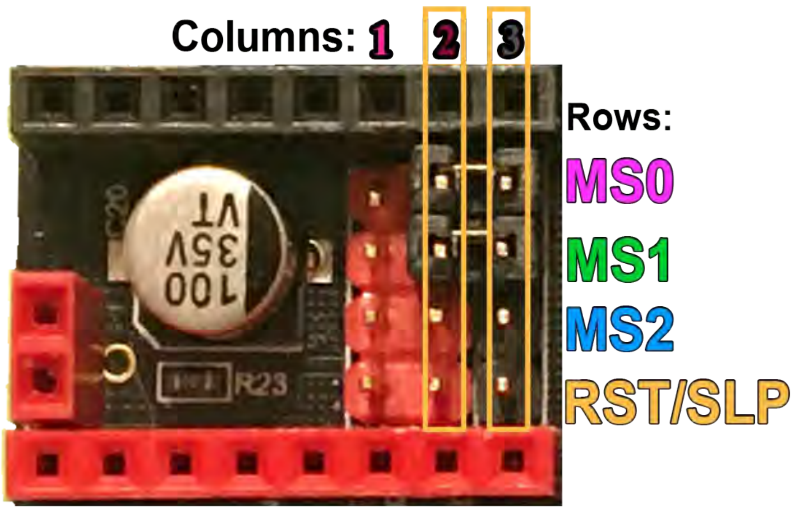
Example 2 (A4988 Driver Board; Binary State Driver) for SKR V1.3 Driver Socket Representation:

1 / 8



Driver Chip Chart:

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



Driver Socket Representation: 7 7 ⇒ Driver Chip Chart: High

MS0 PIN:

High ⇒ set **Jumper** between column 2 and 3 on MS0 row

Driver Socket Reprsentation: 8 8 ⇒ Driver Chip Chart: High

MS1 PIN:

High ⇒ set **Jumper** between column 2 and 3 on MS1 row

Driver Socket Representation: - - ⇒ Driver Chip Chart: Low

MS2 PIN:

Low ⇒ No **Jumper** Set

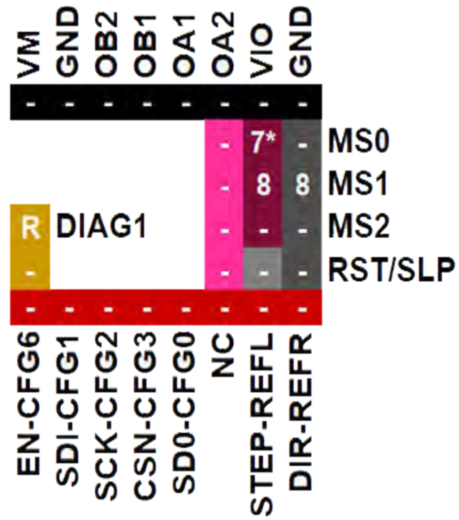
Example 3 (TMC2130 Stand-alone Mode; Tri State Driver) for SKR V1.3 Driver Socket Representation:

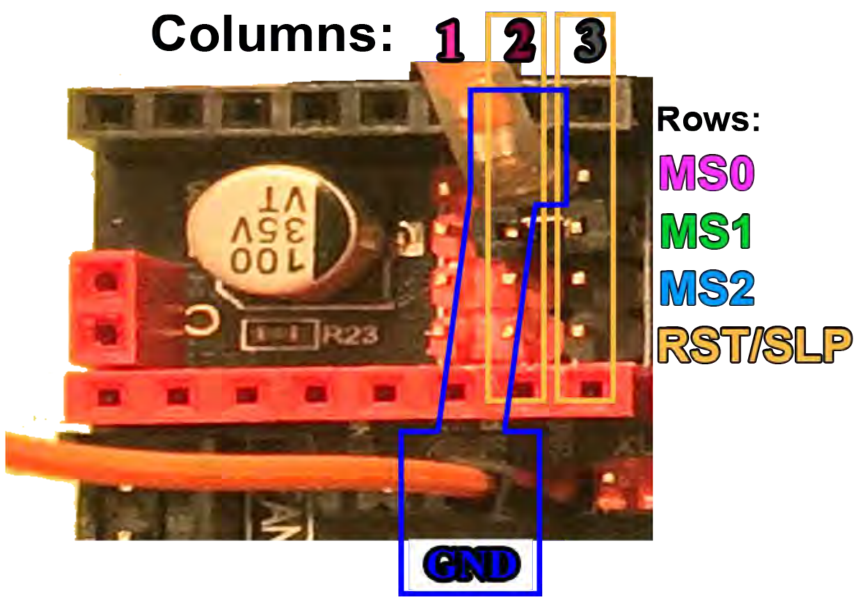
Stand-alone Mode

1 / 4

Interpolation: none

SpreadCycle





Driver Socket Representation: 7* - ⇒ Driver Chip Chart: Low

MS0 PIN:

Low ⇒ Use a **DuPont Jumper Cable** to connect column 2 PIN from the MS0 row to ground (GND)

Driver Socket Representation: 8 8 ⇒ Driver Chip Chart: High

MS1 PIN:

High ⇒ set **Jumper** between column 2 and 3 on MS1 row

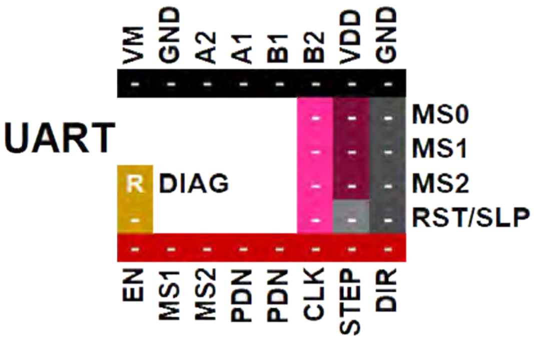
Driver Socket Representation: - - ⇒ Driver Chip Chart: No Jumper Set

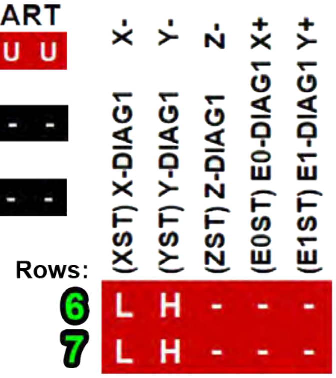
Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
BIQU® TMC2130 Stand Alone Mode	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop

APPENDIX D

Example 4 (TMC2209 UART with Sensor-less Homing) for SKR V1.3 Driver Socket Representation:





- - → No Jumper Set

: → No Jumper Set

UU → Jumper Set

L L → Jumper Set

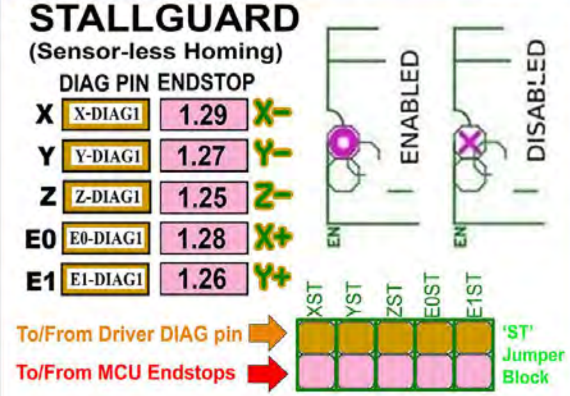
H H → Jumper Set

Driver Socket Representation:	Meaning:
U U	set Jumper between column 4 and 5
L L	set Jumper between rows 6 and 7 & column X
H H	set Jumper between rows 6 and 7 & column Y

STALLGUARD

(Sensor-less Homing)

	DIAG PIN	ENDSTOP	
X	X-DIAG1	1.29	X-
Y	Y-DIAG1	1.27	Y-
Z	Z-DIAG1	1.25	Z-
E0	E0-DIAG1	1.28	X+
E1	E1-DIAG1	1.26	Y+



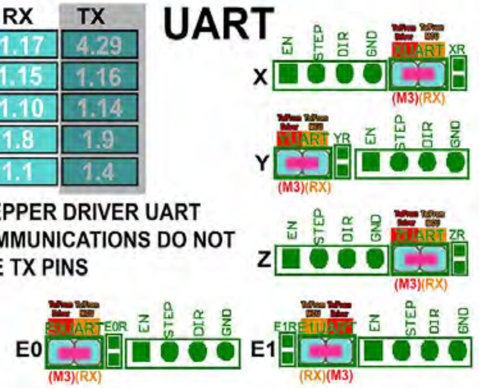
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

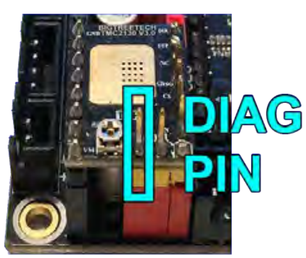
UART

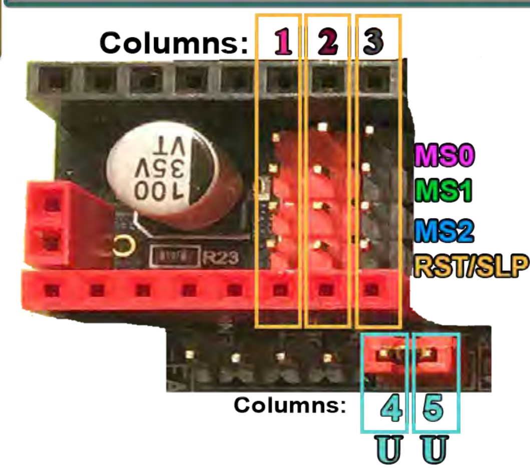
	RX	TX
X	1.17	4.29
Y	1.15	1.16
Z	1.10	1.14
E0	1.8	1.9
E1	1.1	1.4

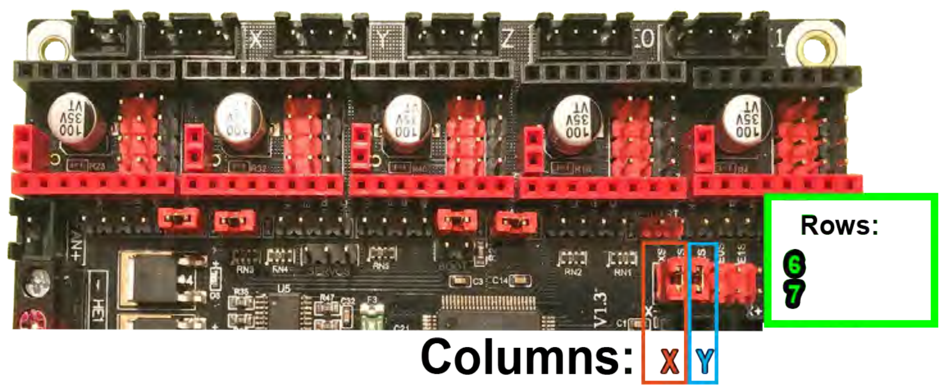
STEPPER DRIVER UART COMMUNICATIONS DO NOT USE TX PINS



To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.

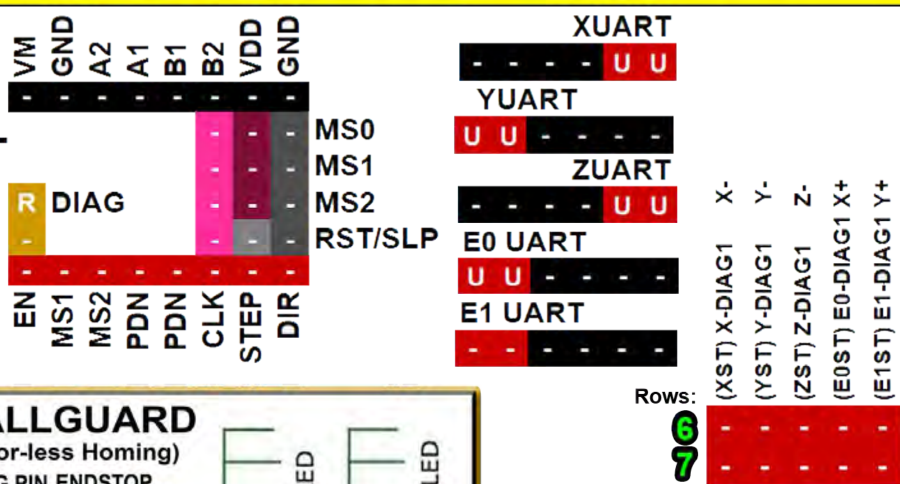


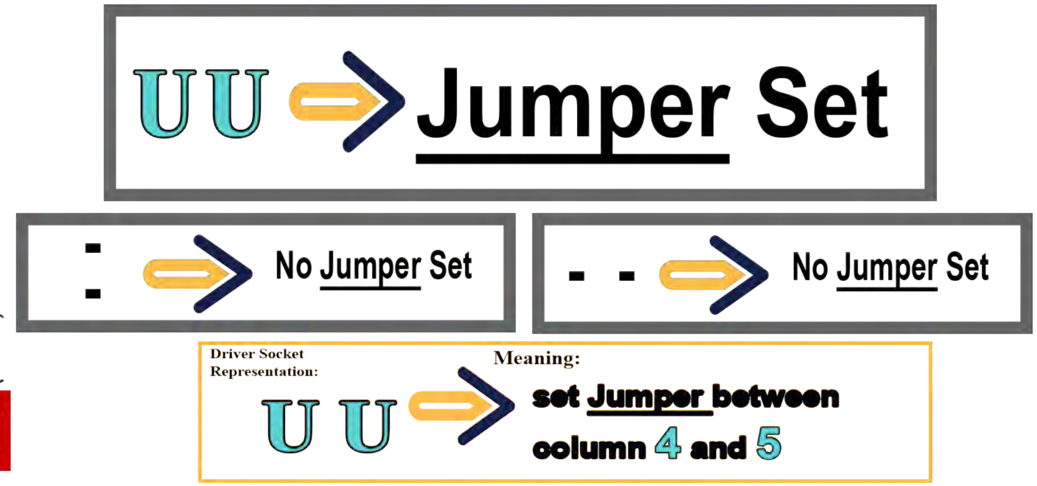




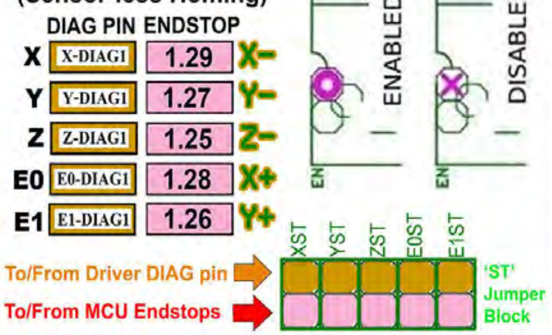
APPENDIX D

Example 5 (TMC2209 UART WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation:





STALLGUARD
(Sensor-less Homing)



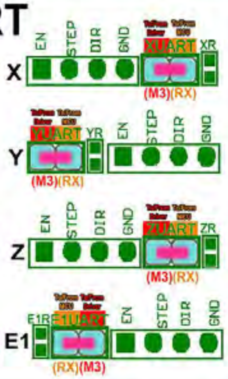
Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

UART

	RX	TX
X	1.17	4.29
Y	1.15	1.16
Z	1.10	1.14
E0	1.8	1.9
E1	1.1	1.4

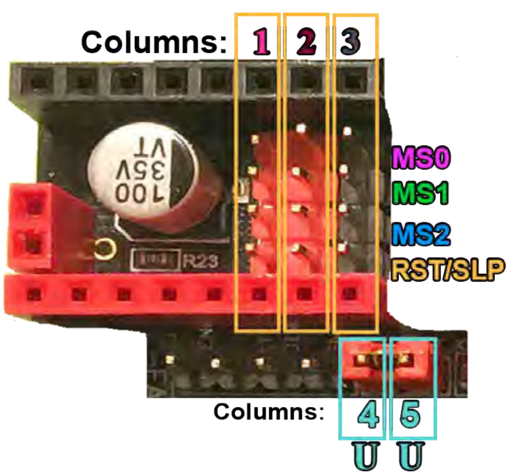
STEPPER DRIVER UART COMMUNICATIONS DO NOT USE TX PINS

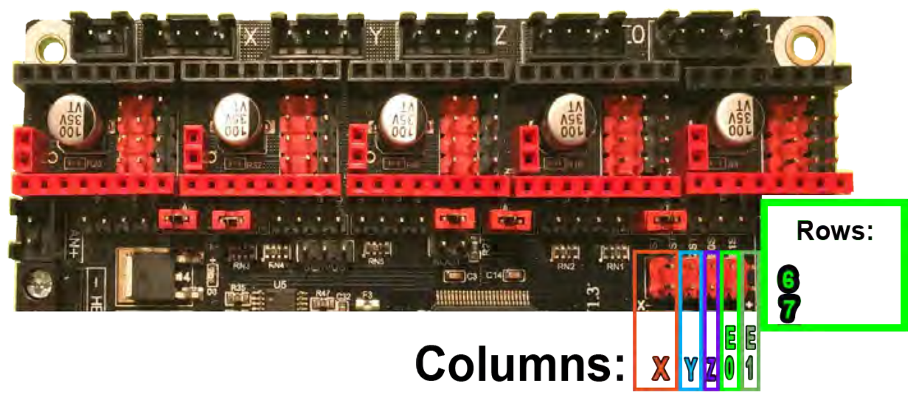


The picture below shows ALL stepper driver sockets are setup for UART mode AND ALL 'ST' JUMPERS have been removed to DISABLE Sensor-less Homing due to the fact that the user wants to use physical endstops/ limit switches on ALL Axes.

To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

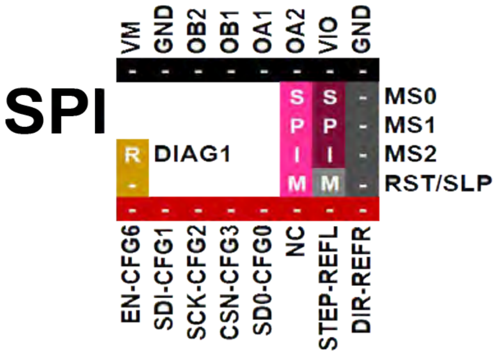
Link to stackable header pins: https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P

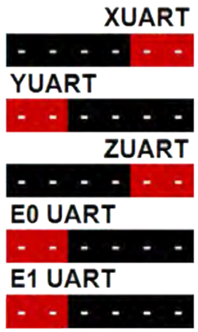




APPENDIX D

Example 6 (TMC2130 SPI with Sensor-less Homing) for SKR V1.3 Driver Socket Representation:





Rows:	X- (XST) X-DIAG1	Y- (YST) Y-DIAG1	Z- (ZST) Z-DIAG1	X+ (E0ST) E0-DIAG1	Y+ (E1ST) E1-DIAG1
6	L	H	-	-	-
7	L	H	-	-	-

- - → No Jumper Set

: → No Jumper Set

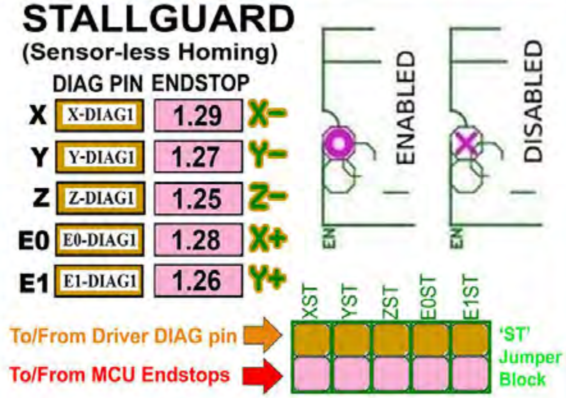
SPIM SPIM → Jumper Set

L L → Jumper Set

H H → Jumper Set

STALLGUARD
(Sensor-less Homing)

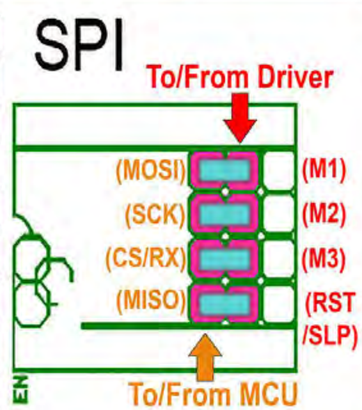
	DIAG PIN	ENDSTOP	
X	X-DIAG1	1.29	X-
Y	Y-DIAG1	1.27	Y-
Z	Z-DIAG1	1.25	Z-
E0	E0-DIAG1	1.28	X+
E1	E1-DIAG1	1.26	Y+



Note Concerning the TMC2209 in UART Mode ONLY:
If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

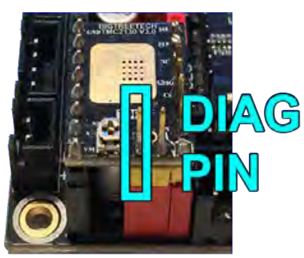
Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

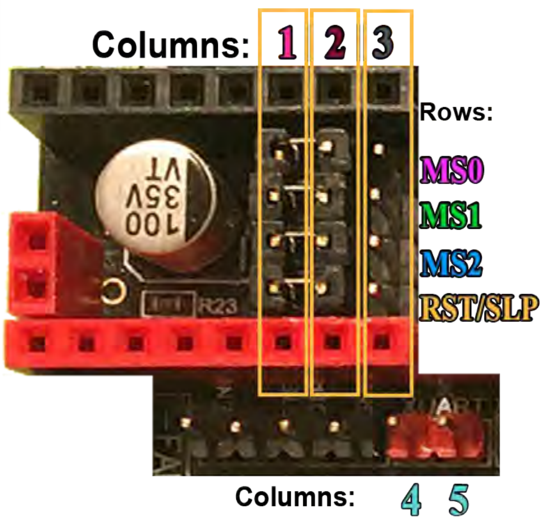
MOSI	4.28
SCK	0.4
MISO	0.5
X-CS	1.17
Y-CS	1.15
Z-CS	1.10
E0-CS	1.8
E1-CS	1.1

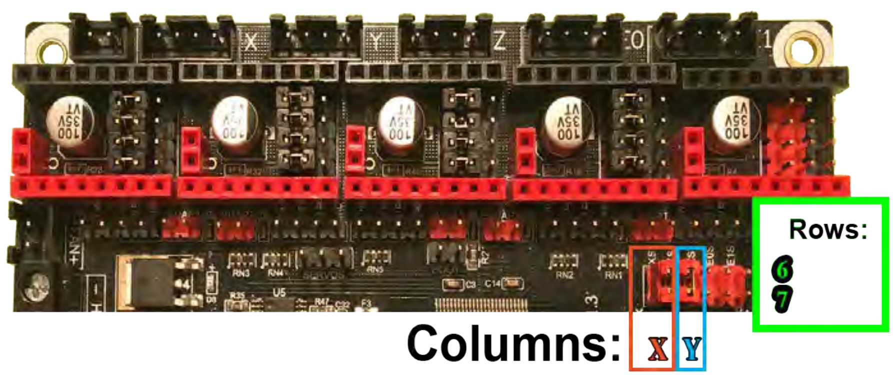


Driver Socket Representation:	Meaning:
SPIM SPIM	set Jumpers between columns 1 and 2 in rows: MS0, MS1, MS2, and RST/SLP
L L	set Jumper between rows 6 and 7 in column X
H H	set Jumper between rows 6 and 7 in column Y

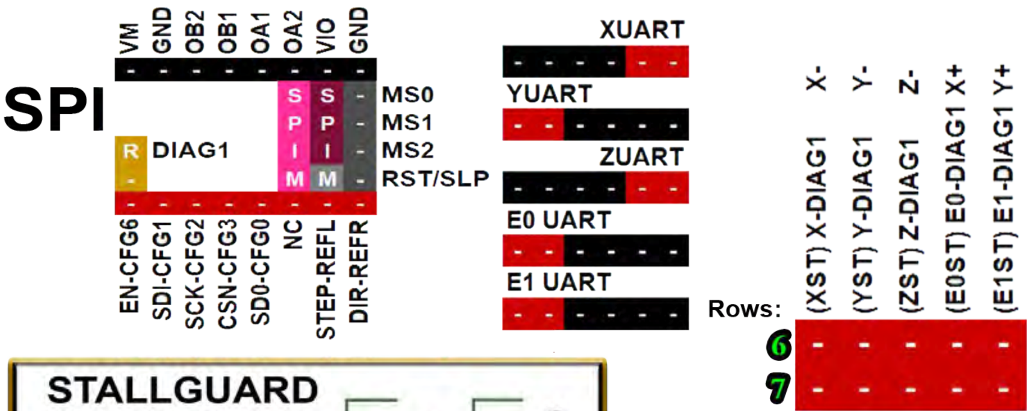
To enable sensor-less homing ensure that the DIAG PIN is plugged into the SKR V1.3 Board AND the 'ST' JUMPER is set in the 'ST' JUMPER Block for the Axis.

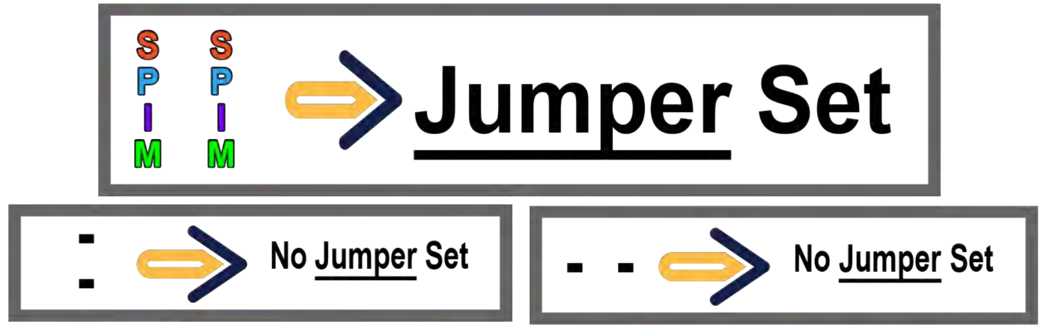






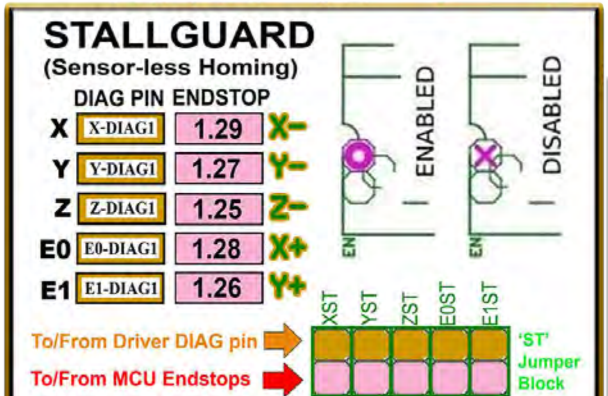
Example 7 (TMC2130 SPI WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation:





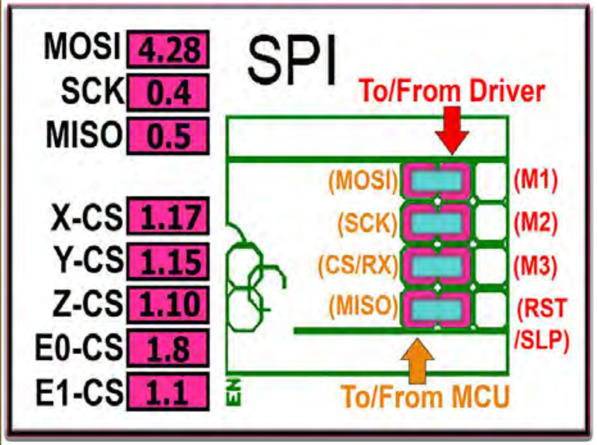
Driver Socket Representation: S P I M S P I M

Meaning: set Jumpers between columns 1 and 2 in rows: MS0, MS1, MS2, and RST/SLP



Note Concerning the TMC2209 in UART Mode ONLY: If using limit switches/enstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

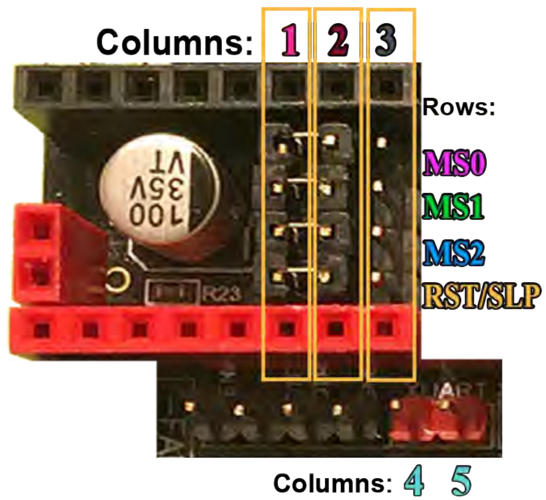
Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Boad that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

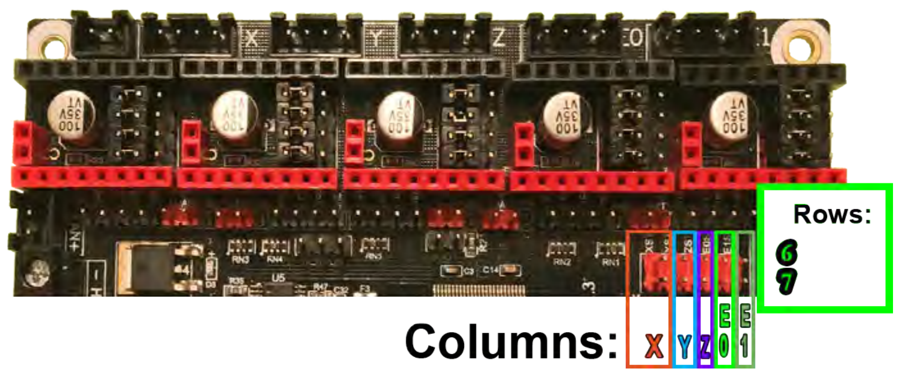


The picture below shows ALL stepper driver sockets are setup for SPI mode AND ALL 'ST' JUMPERS have been removed to DISABLE Sensor-less Homing due to the fact that the user wants to use physical endstops/ limit switches on ALL Axes.

To disable sensor-less homing either ensure that the DIAG PIN is NOT plugged into the SKR V1.3 Board OR remove the 'ST' JUMPER from the 'ST' JUMPER Block.

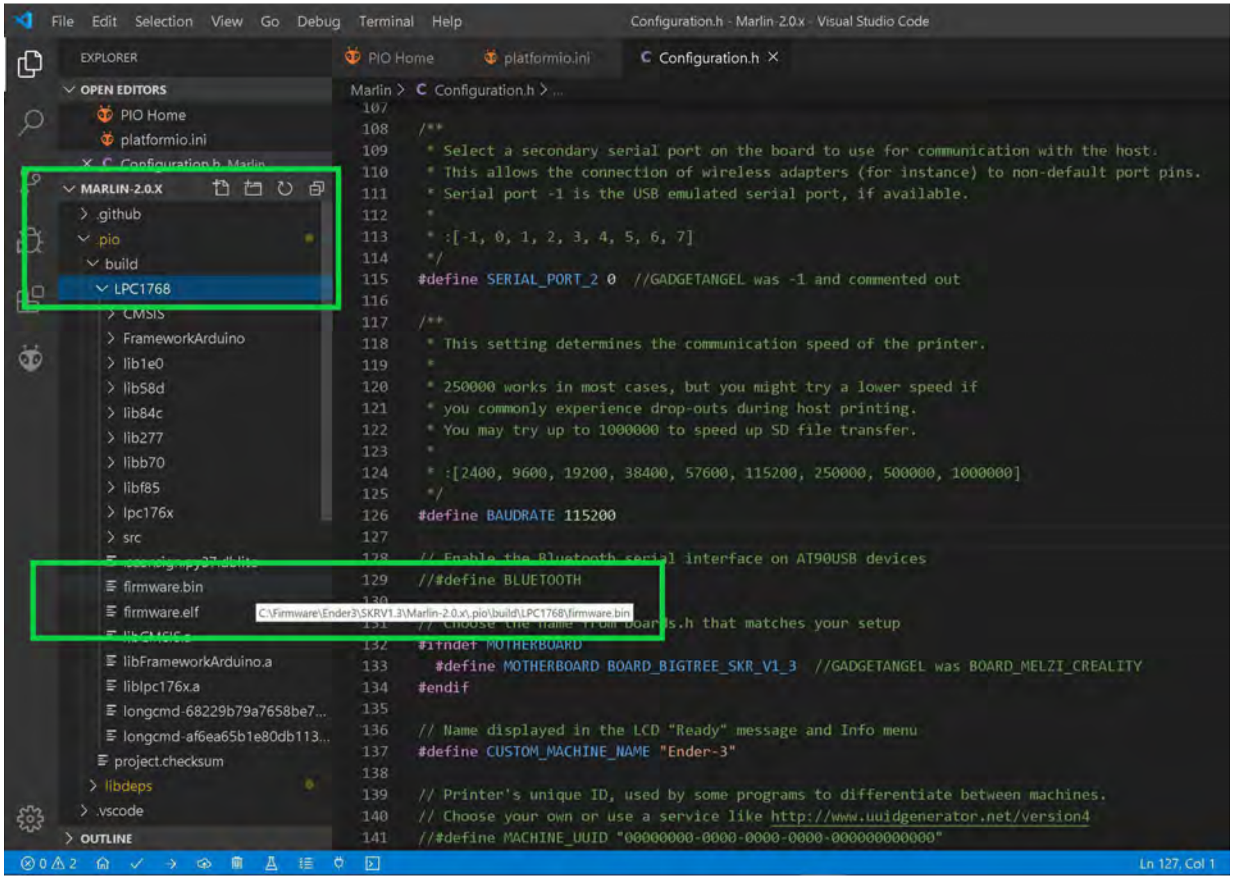
Link to stackable header pins: https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P

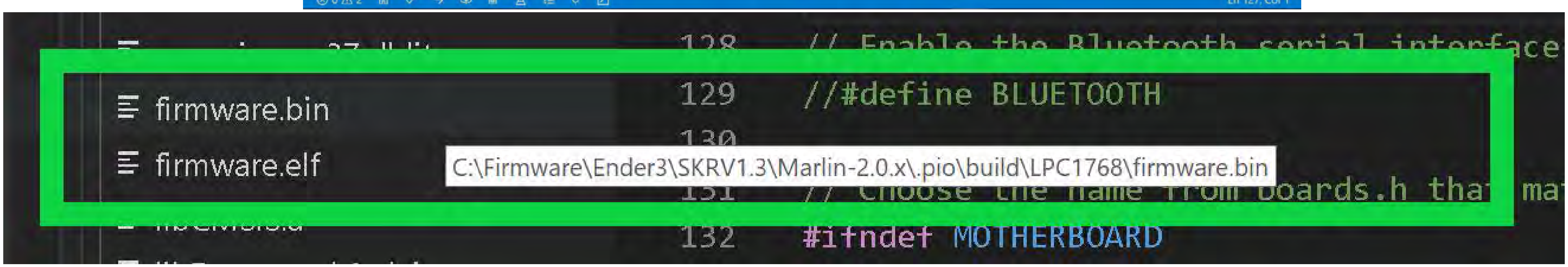




Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

- For the SKR V1.3 board, you can save the "firmware.bin" file to the micro SD card. You can then place the micro SD card into the micro SD card reader of the SKR V1.3 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR V1.3 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\LPC1768\ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file explorer window.

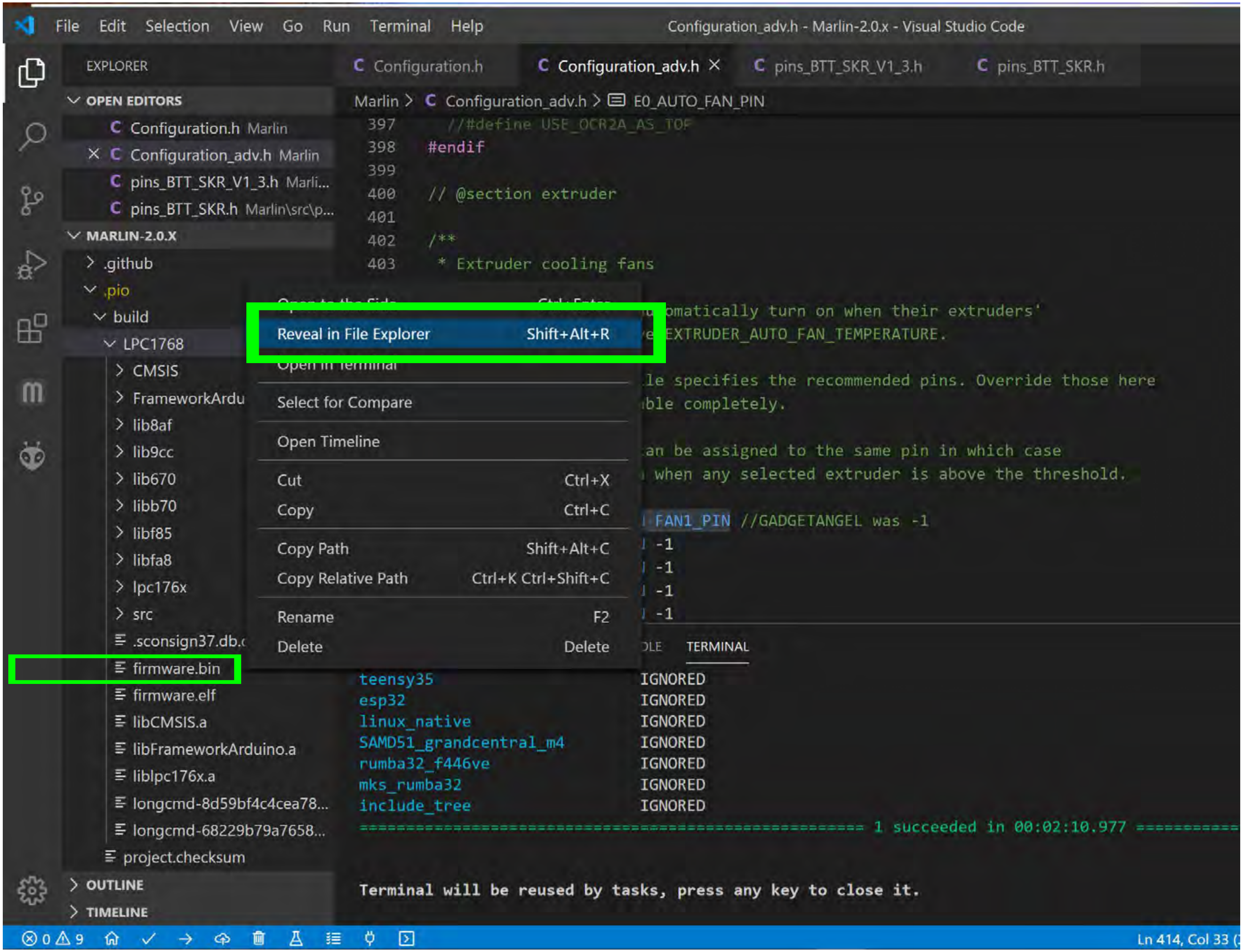




- Go to the next page.

APPENDIX E

Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

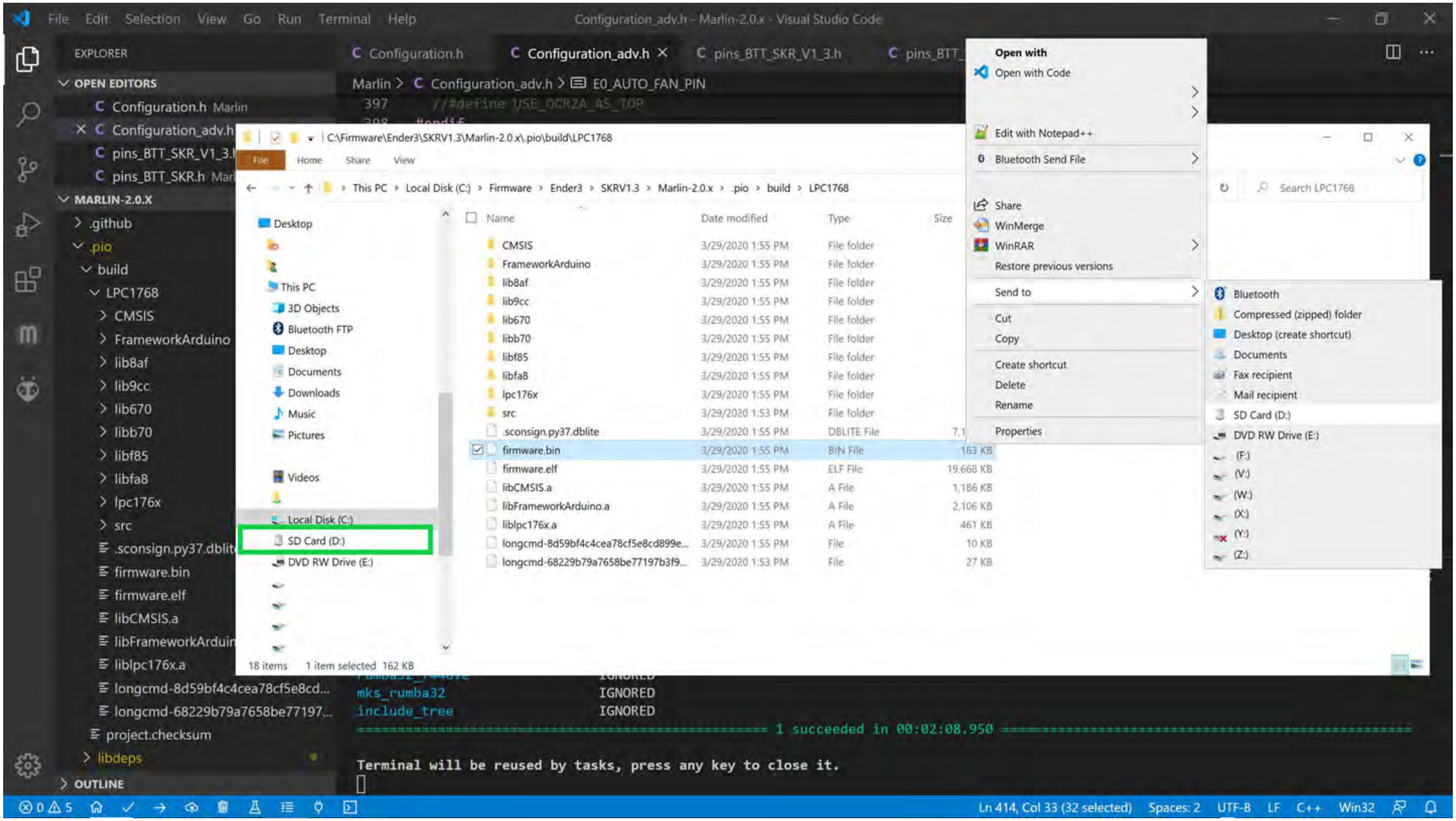


- Go to the next page.

APPENDIX E

Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

- Ensure your micro SD card, which came with your SKR V1.3 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 V1.3 board. Turn on the power to the printer (i.e. the SKR V1.3 board) and the "FIRMWARE.CUR" file will be uploaded to the board. After the upload, the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

APPENDIX F
Links to Reference Material

Marlin Firmware Documentation

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

Information on Stepper Motor Drivers and Micro-stepping

- https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping

- Go to the next page.

APPENDIX F

Links to Reference Material

POLOLU A4988 and BIQU A4988

- YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware
- A4988 Stepper Motor Driver Carrier

DRV8825

- YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware at 667s into the Video
- DRV8825 Stepper Motor Driver Carrier, High Current

- Go to the next page.

APPENDIX F

Links to Reference Material

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

- YouTube Channel Teching Tech: LV8729 guide + comparison with TMC drivers

BIQU LV8729

- BIQU LV8729 Stepper Motor Driver Support 6V-36V With Heatsinks

- Go to the next page.

APPENDIX F

Links to Reference Material

FYSETC LV8729

- FYSETC LV8729 V1.0 Product Information

LERDGE LV8729

- LERDGE LV8729 Stepper Motor Driver Informatin from AliExpress

- Go to the next page.

APPENDIX F

Links to Reference Material

MKS LV8729

- MKS LV8729 - ultra quiet stepper motor driver 1.5A Product Information

FYSETC S6128 V1.1

- FYSETC S6128 V1.1 Product Information

- Go to the next page.

APPENDIX F

Links to Reference Material

FYSETC ST820

- FYSETC ST820 V1.0 Product Information

BIQU ST820

- BIGTREETECH ST820 V1.0 Step Stepper Motor Driver Product Informatin from Alibaba.com
- YouTube Channel Pro3d: First Looks - ST820 NEW Stepper Drivers
- https://github.com/bigtreetech/BIGTREETECH-ST820-V1.0

- Go to the next page.

APPENDIX F

Links to Reference Material

POLOLU ST820 (STSPIN820)

- STSPIN820 Stepper Motor Driver Carrier Product Information

POLOLU MP6500

PW
- YouTube Channel PW: Input signal current to drive an MP6500 Stepper driver board
- Data Sheet for MP6500
- MP6500 Stepper Motor Driver Carrier, (Potentiometer Current Control) Product Information

- Go to the next page.

APPENDIX F
Links to Reference Material

POLOLU TB67S249FTG

- Data sheet for TB67S249FTG Stepper Motor Driver TOSHIBA
- Another Data Sheet on TB67S249FTG Stepper Motor Driver from POLOLU website Pololu Robotics & Electronics
- POLOLU TB67S249FTG Stepper Motor Driver Compact Carrier Product Information Pololu Robotics & Electronics

BIQU S109 and FYSETC S109

- Data sheet TB67S109AFTG for S109 Stepper Motor Drivers TOSHIBA

- Go to the next page.

APPENDIX F

Links to Reference Material

BIQU S109

- BIGTREETECH S109 V1.0 Stepper Motor Driver Product Information

FYSETC S109

- FYSETC S109 V1.1 Product Information

- Go to the next page.

APPENDIX F

Links to Reference Material

Marlin Firmware Documentation Specific to TMC Drivers

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

Information Common to All TMC Drivers

- https://github.com/watterott/SilentStepStick
- Trinamic: The SilentStepStick is a stepper driver board for 2-phase motors, based on the TMC2100, TMC2130, TMC2208, TMC2209 or TMC5160
- Instructable: Install and Configure SilentStepStick in RAMPS

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU TMC2100 and MKS TMC2100

- TMC2100 SILENTSTEPSTICK PIN out
- TMC2100 WIKI site
- YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware stating at 869seonds
- TMC2100 DATASHEET
- Trinamic's Website: TMC2100 DATASHEET
- BIGTREETECH TMC2100 Stepper Motor Driver Product Information

- Go to the next page.

APPENDIX F

Links to Reference Material

BIQU TMC2130

- TMC 2130 Stepper Drivers - Standalone Mode - Chris's Basement
- TMC2130 Guide - stepper motors driver upgrades part 3 - Sensorless homing, starting at 34s
- TMC2130 SILENTSTEPSTICK PINout
- https://github.com/bigtreetech/BIGTREETECH-TMC2130-V3.0/blob/master/TMC2130-V3.0RM.pdf
- Trinamic's TMC2130 DATASHEET
- Trinamic 2130 Stepper Driver Setup - Marlin - Chris's Basement
- TMC2130 Guide - stepper motors driver upgrades part 3 - Sensorless homing, starting at 555s

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

- TMC2208 UART on BigTreeTech/BIQU SKR V1.1, V1.3, and V1.4 Controllers
- Trinamic's TMC2202, TMC22082 & TMC2224 family Datasheet
- TMC2208 SILENTSTEPSTICK PINout
- YouTube Channel Da Hai Zhu: TMC2208 3 Ways - Part 1
- TMC2208 guide - Stepper driver upgrades part 2
- TMC2208 guide - Stepper driver upgrades part 2, starting at 419s

- Go to the next page.

Links to Reference Material

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

- YouTube Channel Da Hai Zhu: TMC2208 3 ways - Part 2
- YouTube Channel Keith Young: TMC2208
- YouTube Channel Da Hai Zhu: TMC2208 3 Ways - Part 3
- YouTube Channel Teching Tech: TMC2208 guide - Stepper driver upgrades part 2, starting at 510s

BIQU TMC2208 V3.0

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

FYSETC TMC2208 V1.2

- FYSETC TMC2208 Product Information

- Go to the next page.

APPENDIX F

Links to Reference Material

Information Common to TMC2208 and BIQU TMC2225

- Instructable: TMC2208 UART on BigTreeTech/BIQU SKR V1.1, V1.3, and V1.4 Controllers

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
- TMC2225 Datasheet

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU TMC2209 V1.2

- YouTube Channel Teching Tech: TMC2209 and TMC 5160: Guide for MKS Gen L and SKR V1.3
- TMC2209 Stepper Drivers - Bigtreetech - SKR 1.3 - Install - Chris's Basement
- TMC2209 SILENTSTEPSTICK PINout
- https://github.com/bigtreetech/BIGTREETECH-TMC2209-V1.2/blob/master/manual/TMC2209-V1.2-manual.pdf
- Trinamic's TMC2209 Datasheet

BIQU TMC5160 V1.2

- YouTube Channel Teching Tech: TMC2209 and TMC 5160: Guide for MKS Gen L and SKR V1.3
- TMC5160 SILENTSTEPSTICK PINout
- Trinamic's TMC5160 / TMC5160A DATASHEET
- https://github.com/bigtreetech/BIGTREETECH-TMC5160-V1.0/blob/master/manual/BIGTREETECH%20TMC5160-V1.0%20manual.pdf
- BIGTREETECH TMC5160 V1.2 Stepper Motor Driver SPI Mode, Product Information

- Go to the next page.

APPENDIX F

Links to Reference Material

BIQU TMC5161 V1.0

- https://github.com/bigtreetech/BIGTREETECH-TMC5161-V1.0/blob/master/TMC5161%20v1.0%20mamual.pdf
- Trinamic's TMC5161A DATASHEET

SKR V1.3 Board

- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/BigtreeTech%20SKR%201.3%20Case%20model
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/Wiring%20diagram
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/firmware
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/hardware
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/issues
- https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/issues?q=is%3Aissue+is%3Aclosed

• Go to the next page.

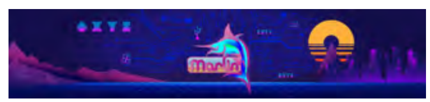
APPENDIX F

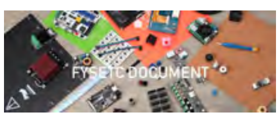
Links to Reference Material

Facebook Groups

- BIGTREETECH SKR PRO User Group 
- BIGTREETECH 

- Bigtreetech / Biqu SKR 32 Bit 3D printer board owners group 
- MarlinFirmware 

- Marlin Firmware for 3D Printers User Group 

- FYSETC 

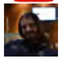
- The Institute of 3D Printing 

- Go to the next page.

APPENDIX F

Links to Reference Material

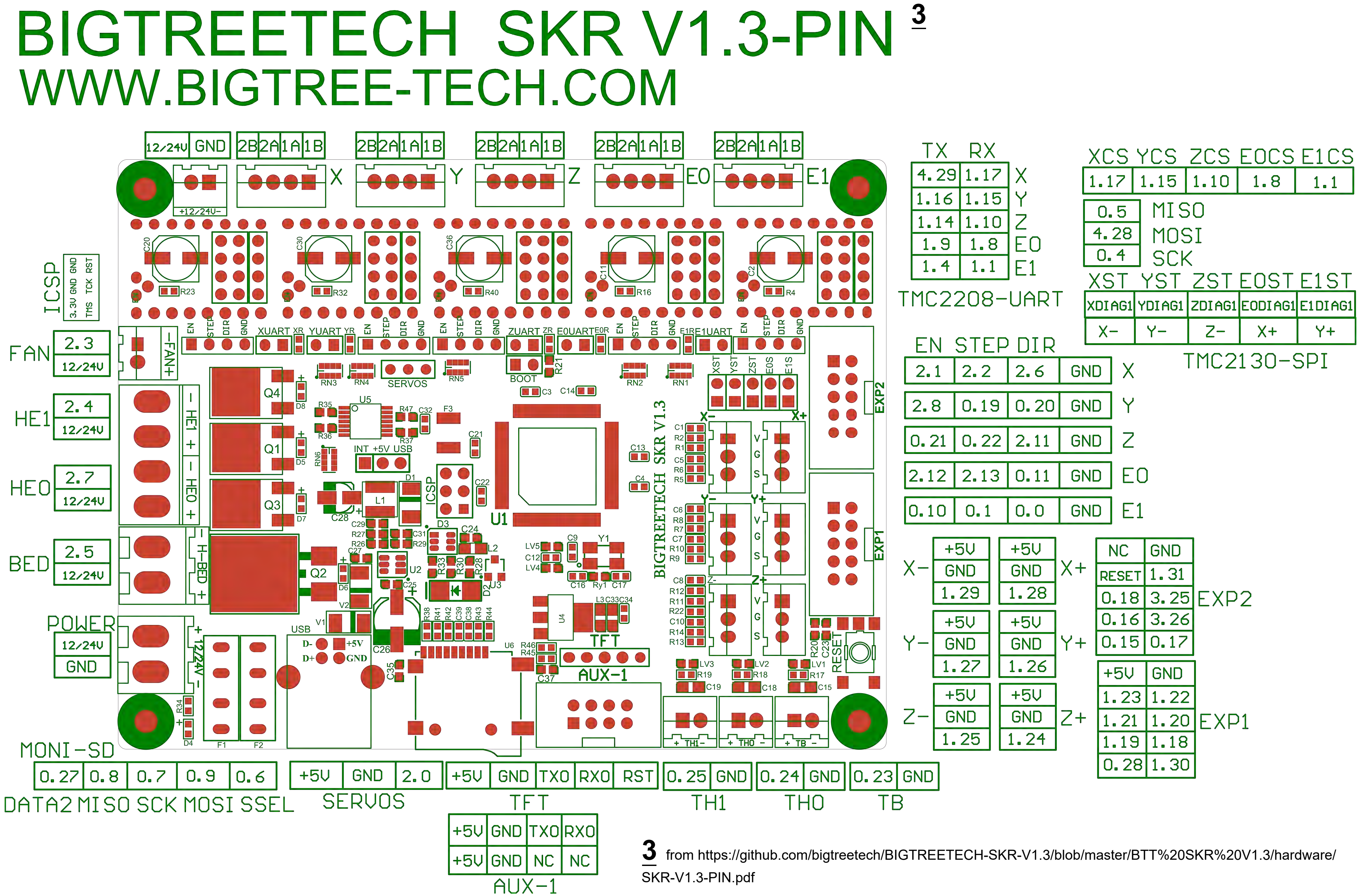
Miscellaneous Information



- YouTube Channel Stelios Liver: Vref formula for common step motor drivers
- Stepper wiring WIKI
- Watterott CONFIGURATOR FOR TMC220X
- FYSETC's TMC2208/#to-run-the-program
- https://marlinfw.org/meta/gcode/
- https://blog.prusaprinters.org/calculator/
- https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers
- https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers
- https://marlinfw.org/docs/basics/install_platformio.html
- https://marlinfw.org/meta/download/
- https://www.pronterface.com/
- Sensorless Homing
- Watterott's Silentstepstick Comparison
- https://reprap.org/wiki/Category:Reference
- https://reprap.org/wiki/Calibration
- How does one use the PT100 Amp Board with a 3.3v controller?
- ▪Stepper Motor Wiring Conventions
- ▪BLtouch probe installation on SKR 32-bit board

Original PIN Diagram:

BIGTREETECH Reference Material

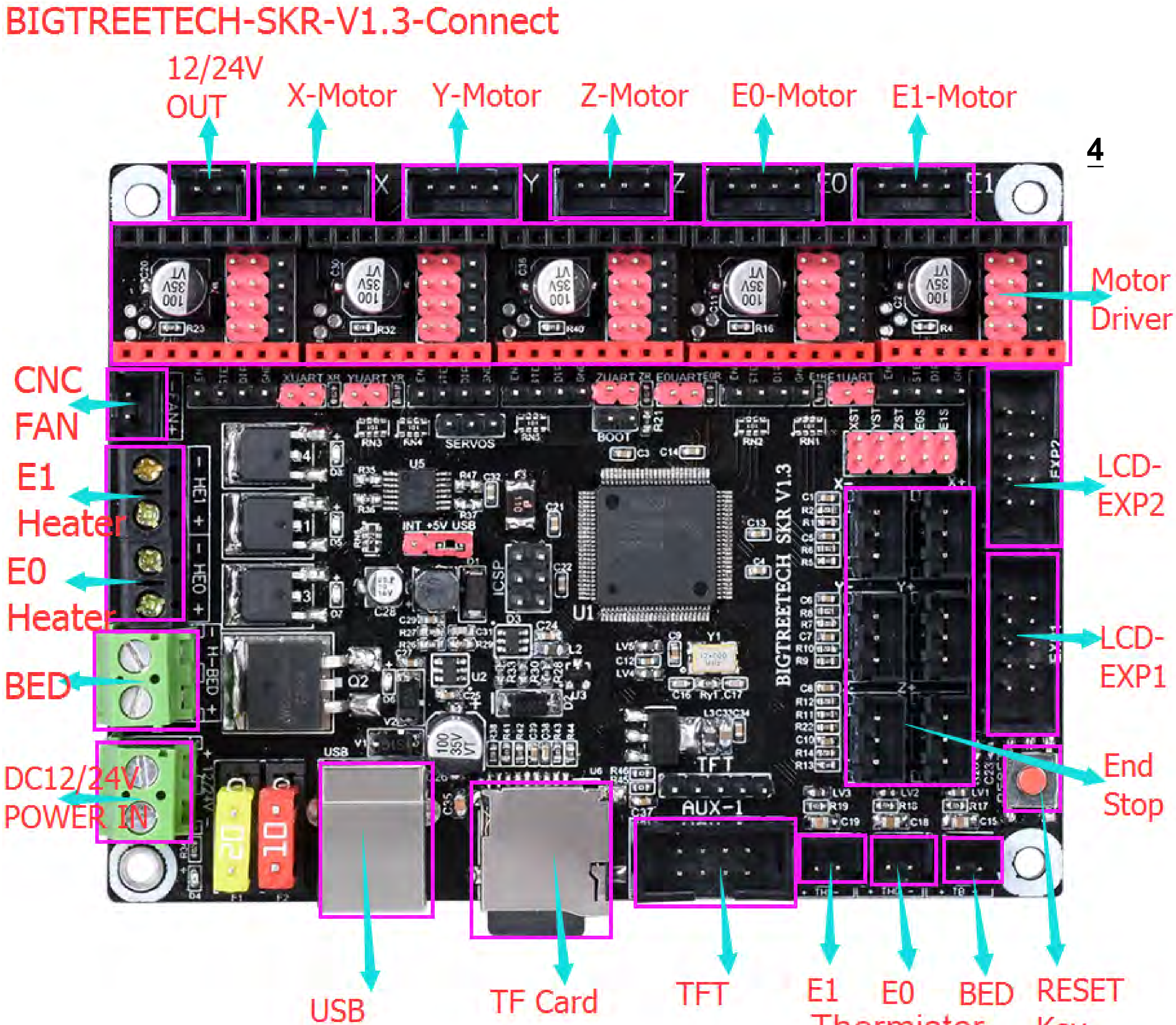


3 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKR-V1.3-PIN.pdf

- Go to the next page.

APPENDIX G
BIGTREETECH Reference Material

Original Wiring Diagram 1:



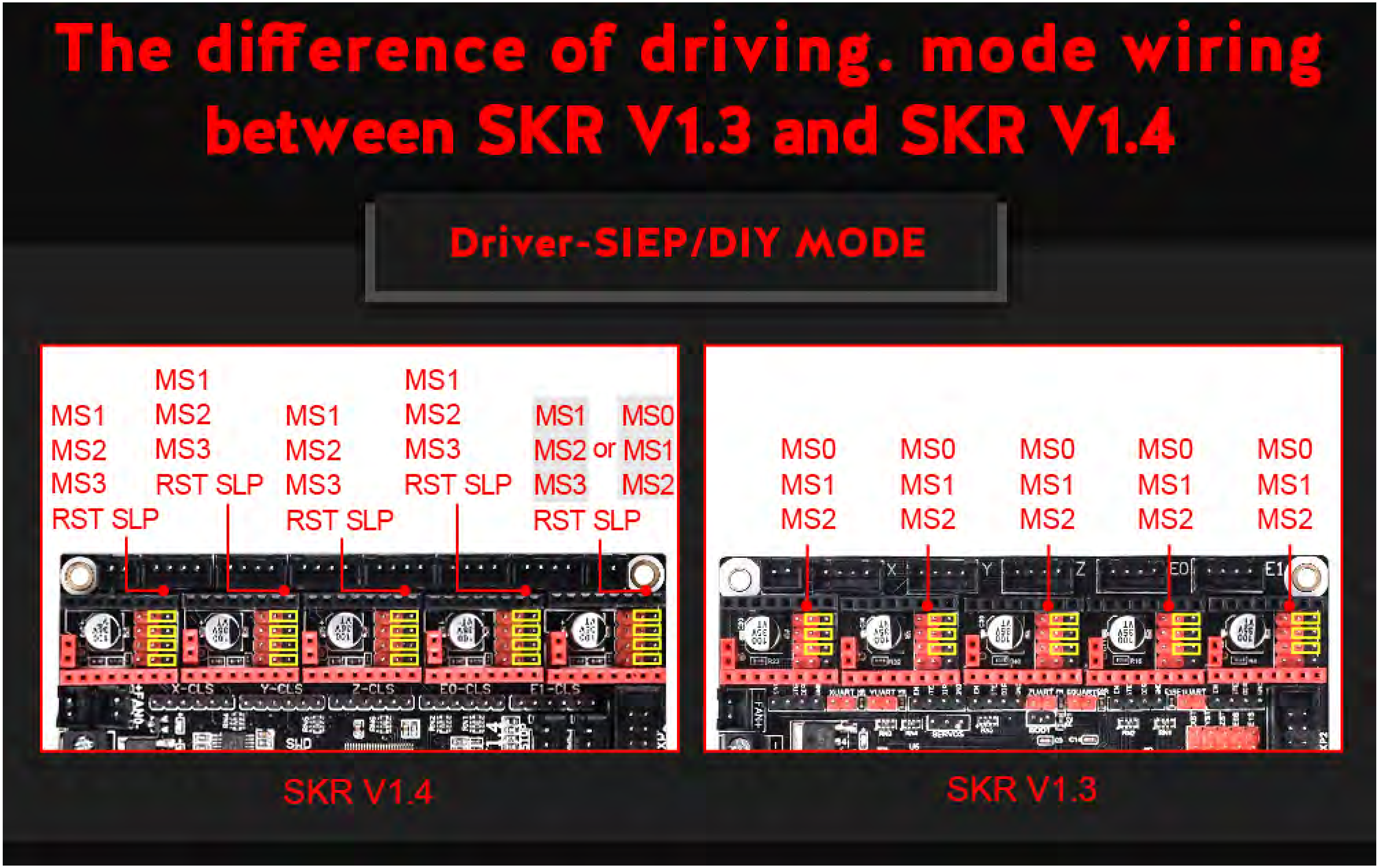
4

4 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/Wiring%20diagram/SKR%20V1.3%20Wiring%20diagram.jpg

- Go to the next page.

BIGTREETECH Reference Material

Original Wiring Diagram 2 for STEP/DIR Mode: 5



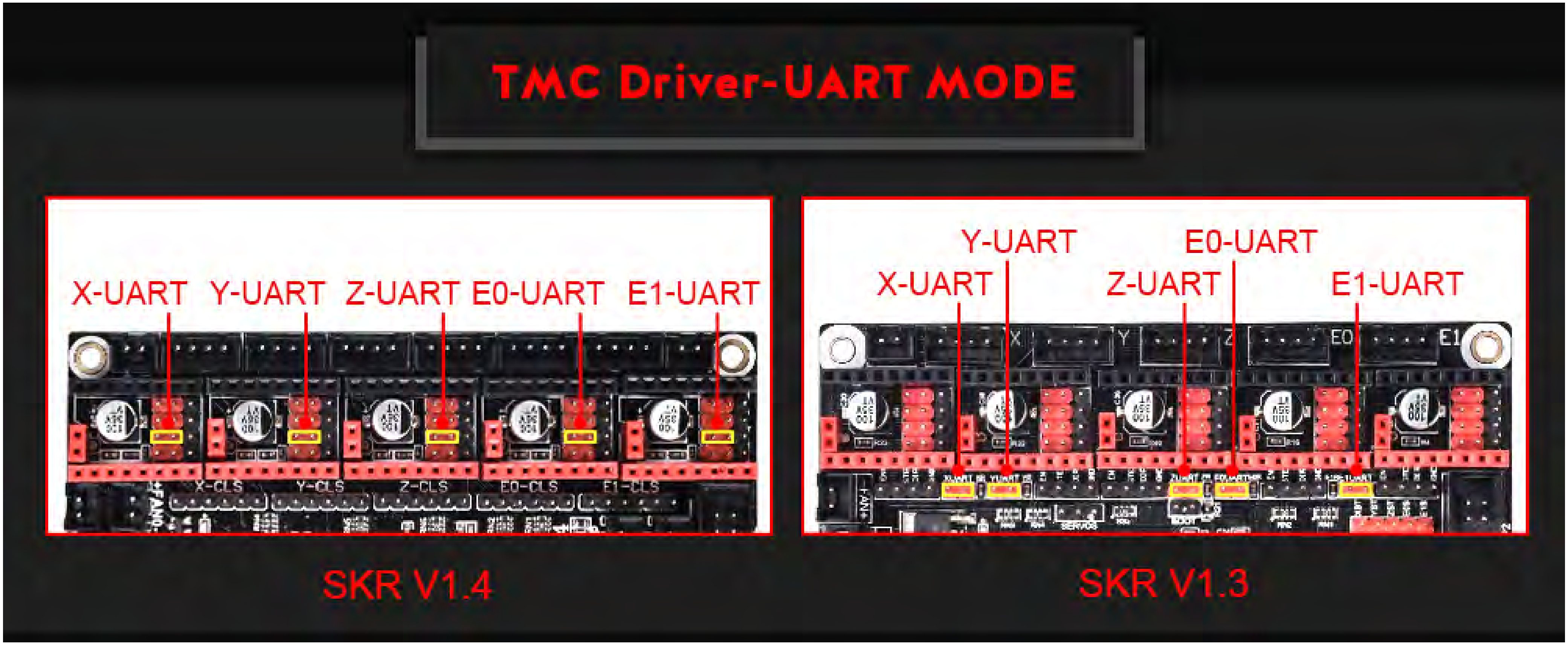
5 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/STEP%20DIR.jpg

- Go to the next page.

APPENDIX G
BIGTREETECH Reference Material

Original Wiring Diagram 3 for UART Mode:

6



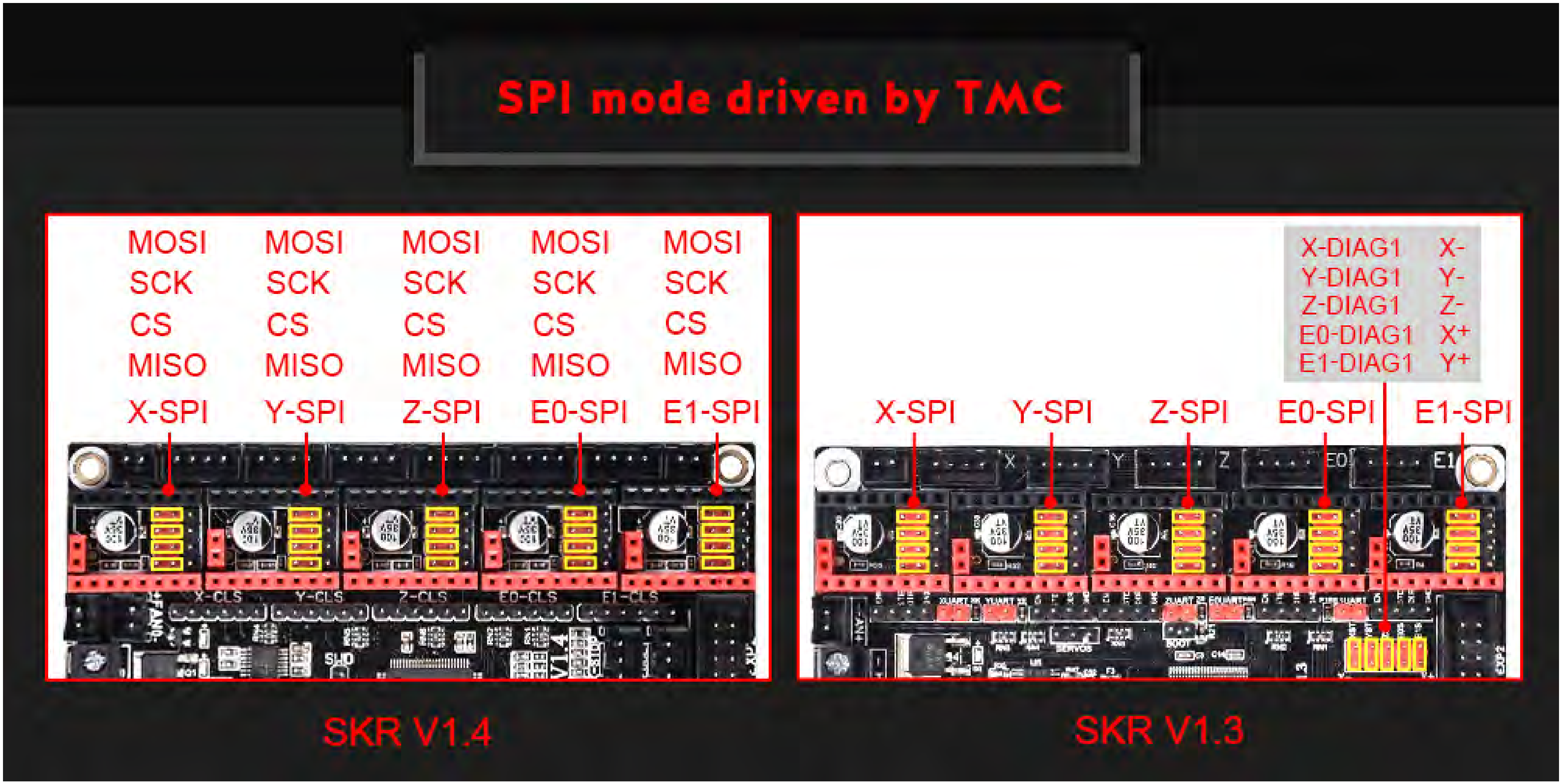
6 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/UART.jpg

- Go to the next page.

APPENDIX G
BIGTREETECH Reference Material

Original Wiring Diagram 4 for SPI Mode:

7



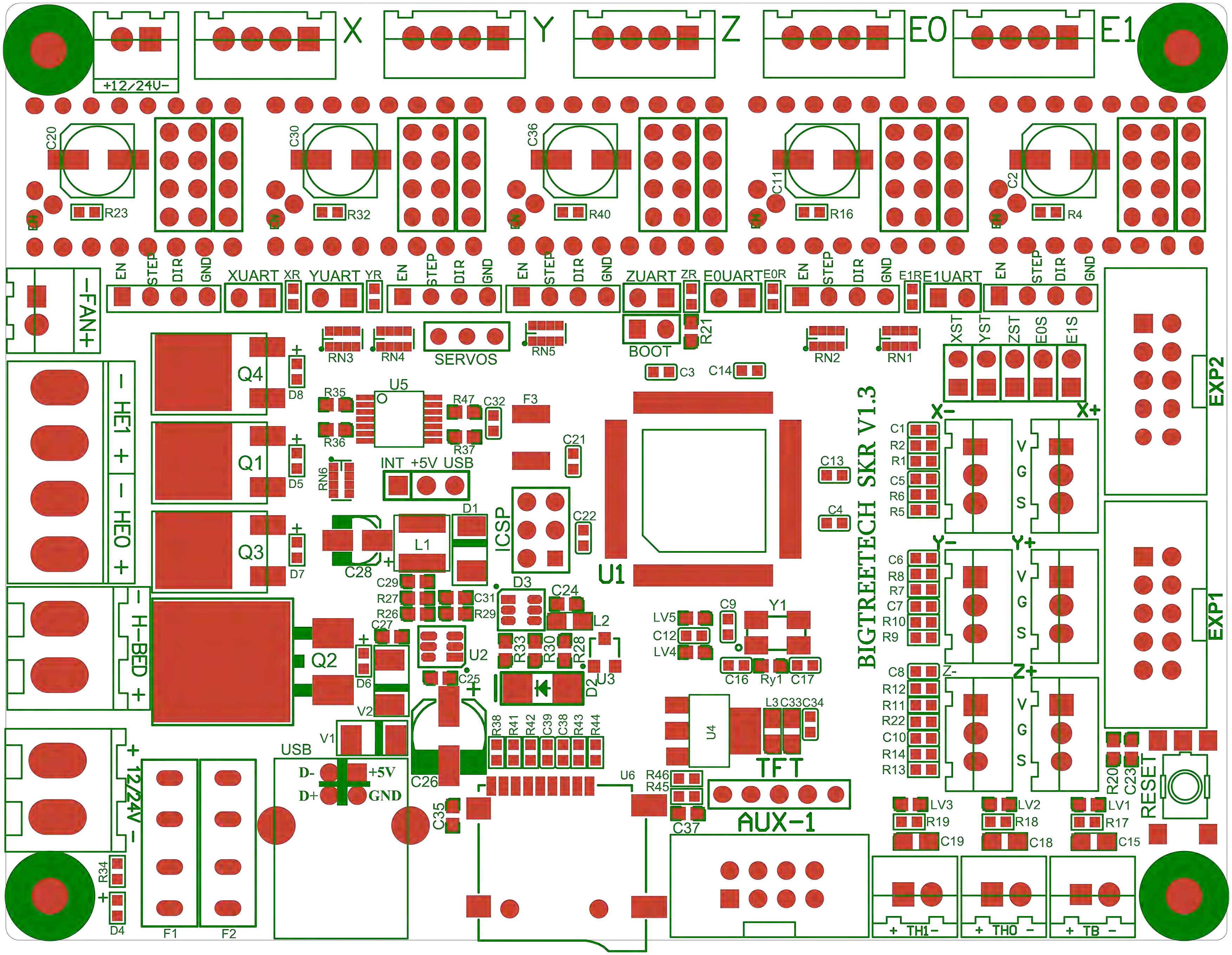
7 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/SPI.jpg

- Go to the next page.

APPENDIX G
BIGTREETECH Reference Material

Original PIN 1 Diagram:

8



8 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKR-V1.3-TOP.pdf

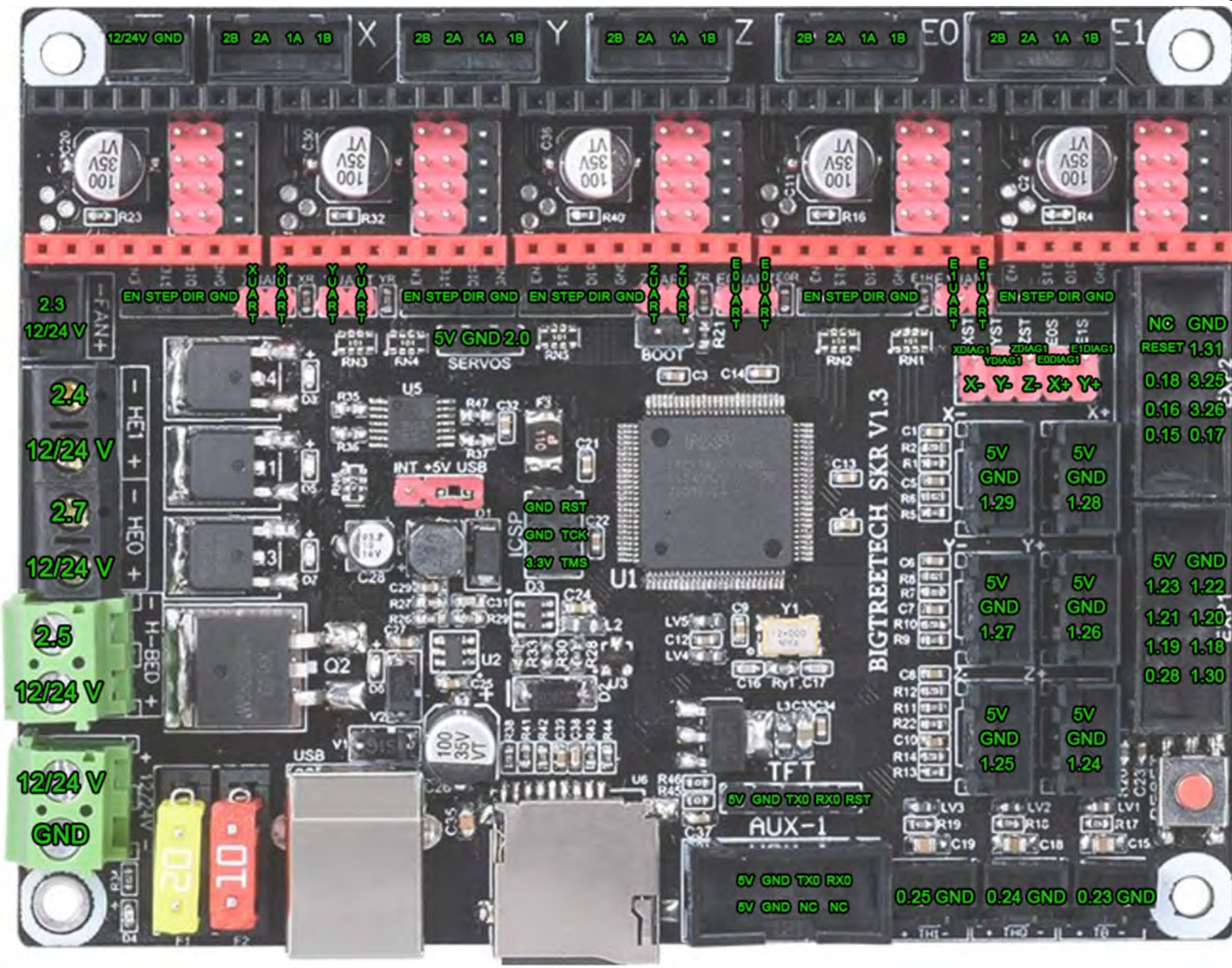
- Go to the next page.

APPENDIX G

BIGTREETECH Reference Material

Additional Original Reference Material for PIN 1 Diagram:

9

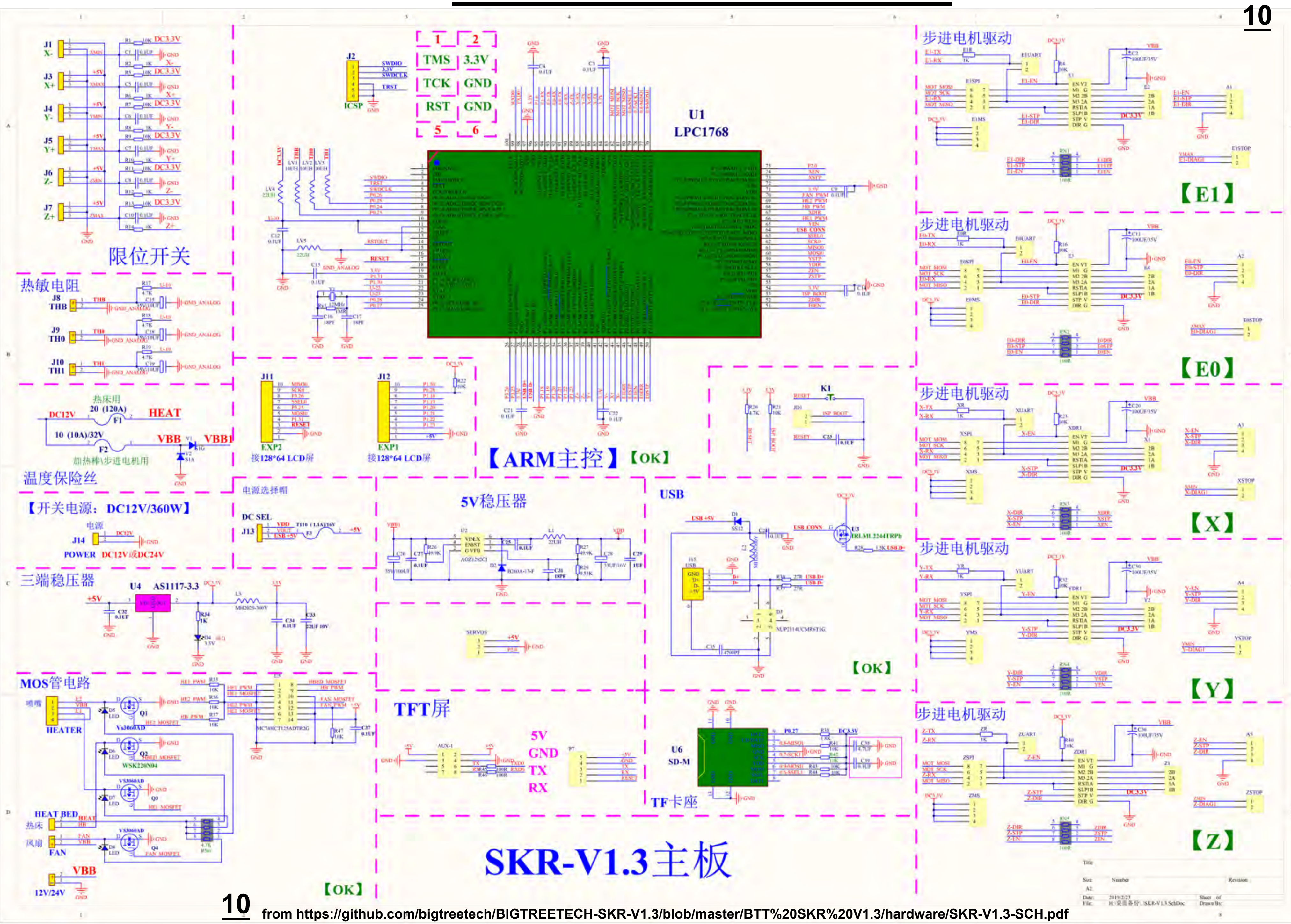


9 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKRpinoutpic.jpg

Original Schematic Diagram:

BIGTREETECH Reference Material

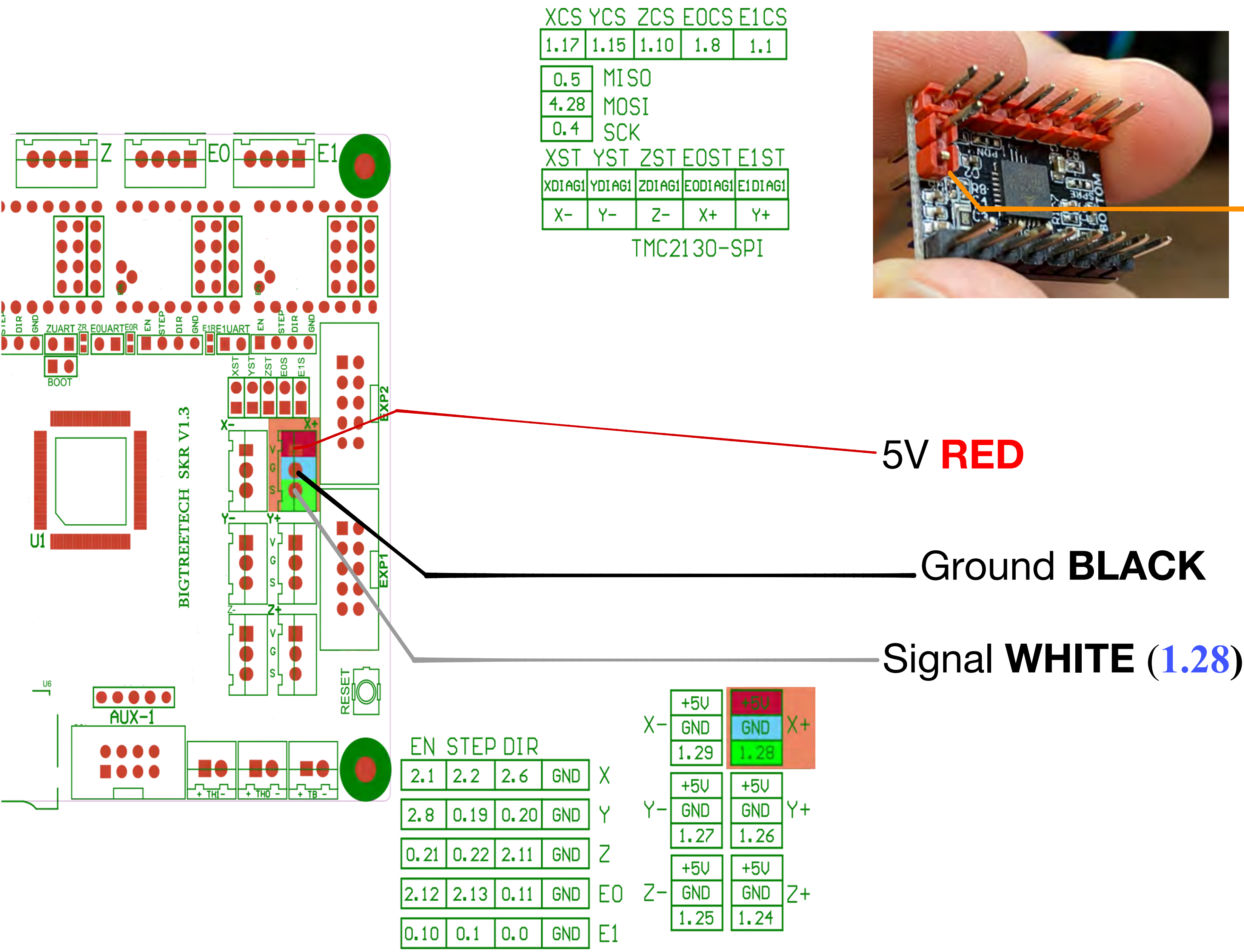
10



10 from https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKR-V1.3-SCH.pdf

APPENDIX H

Filament Runout Sensor Wired to Limit Switch {X-, Y-, Z-, X+, Y+, or Z+}



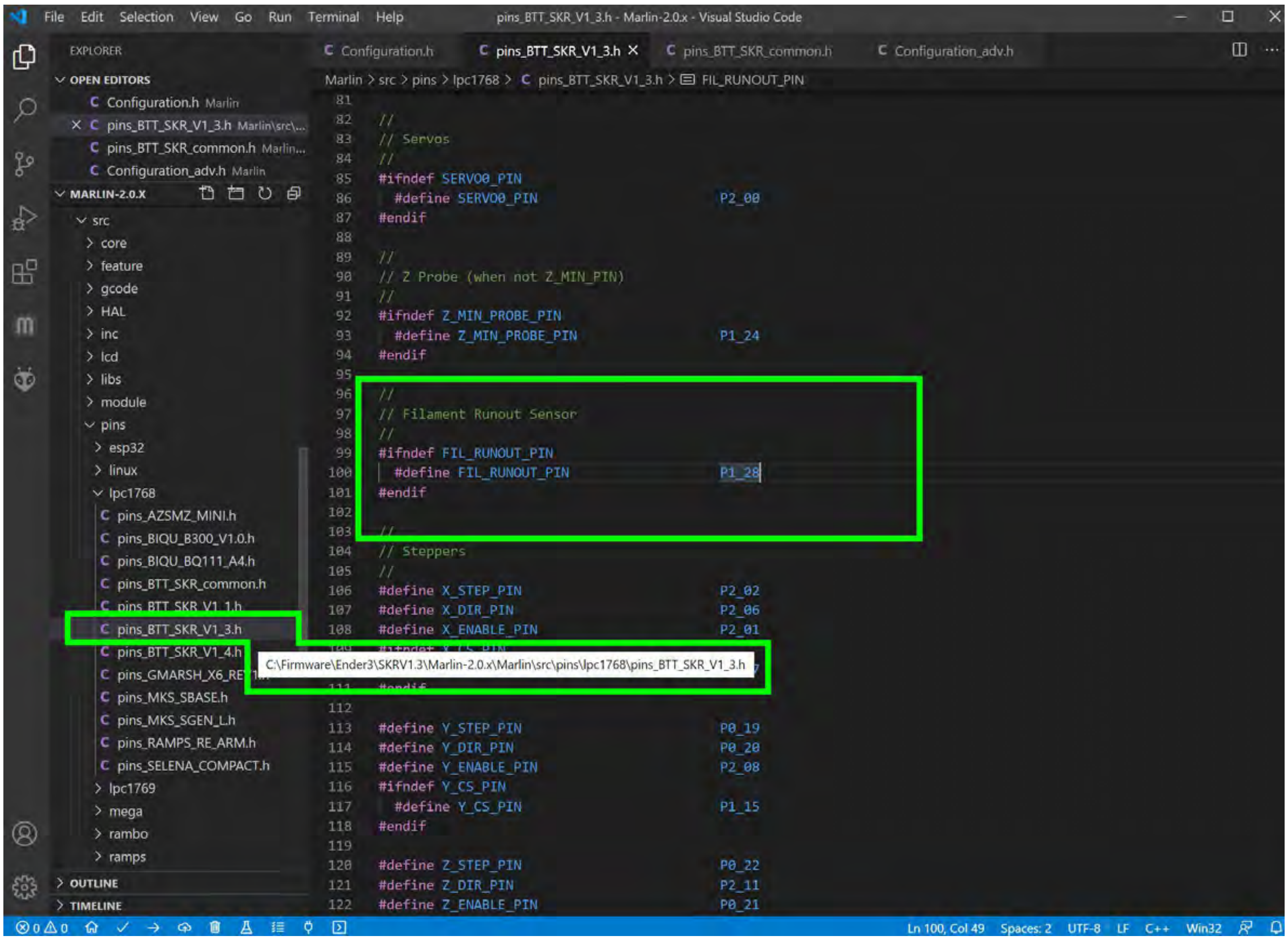
The filament runout sensor will not work with the diagnostic PIN (DIAG/DIAG0/DIAG1) of the extruded stepper (or other Limit switches/ Endstops) so you MUST cut it off or use a soldering iron to heat up the PIN and slide it up out of the way.

- Go to the next page.

Marlin 2.0.x Setup for Filament Runout Sensor Connected to X+ Endstop Connector

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**

- **Ensure the FIL_RUNOUT_PIN is defined in pins_BTT_SKR_V1_3.h file. The pins file for SKR V1.3 board is located in "...\Marlin\src\pins\lpc1768\ subdirectory.**
- **FIL_RUNOUT_PIN should be equal to P1_28**

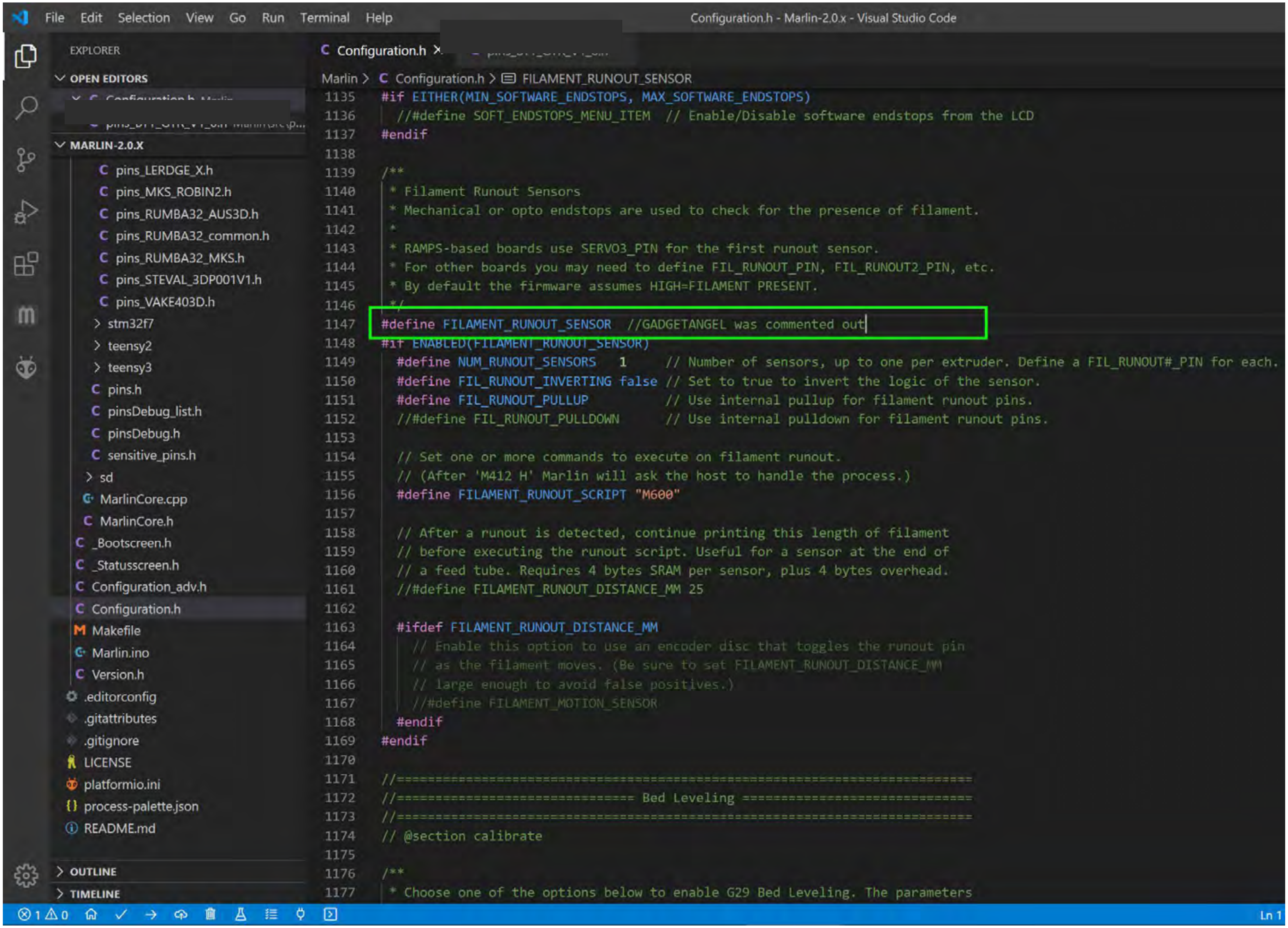


- Go to the next page.

APPENDIX H

Marlin 2.0.x Setup for Filament Runout Sensor

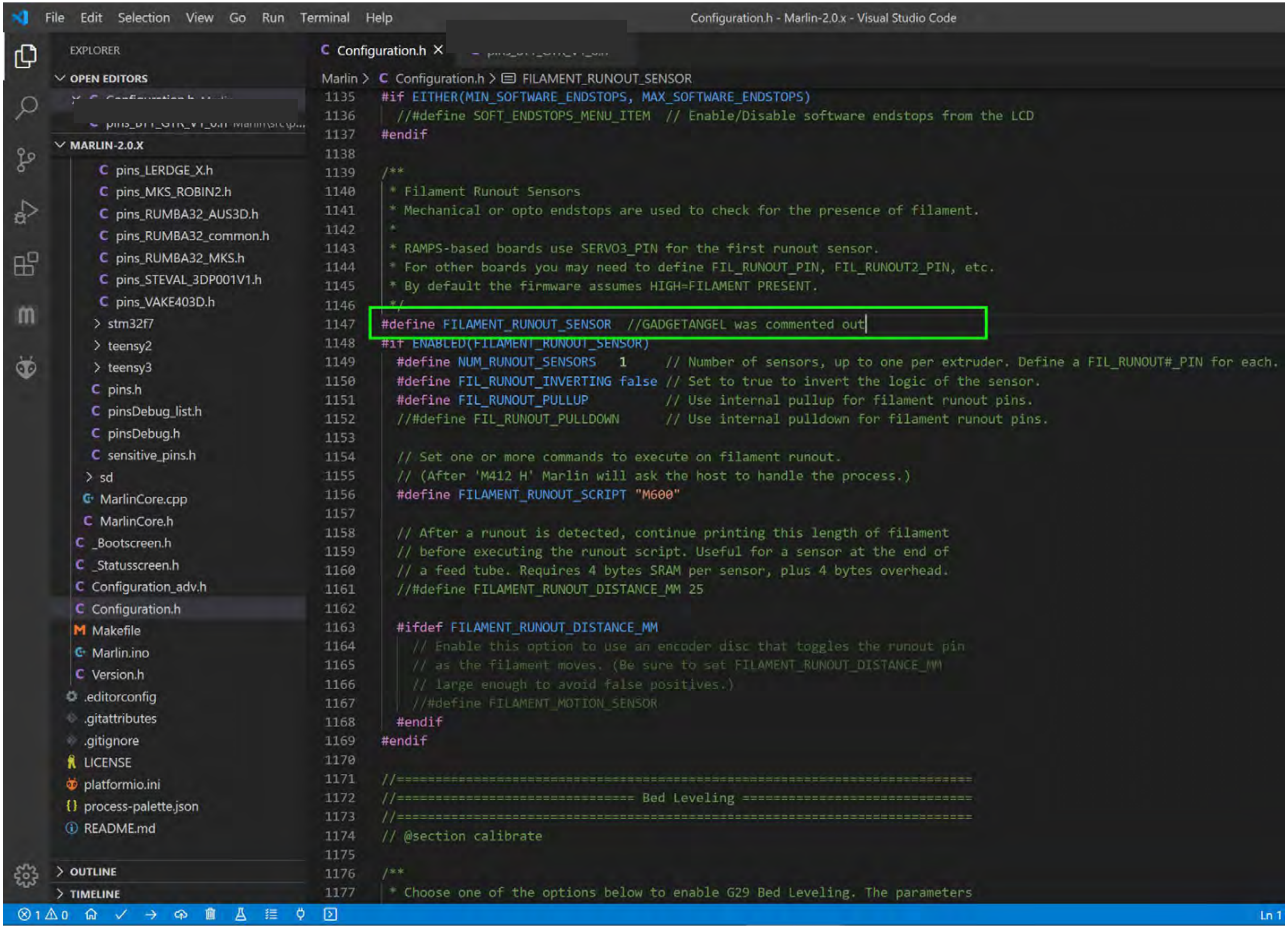
- **In Configuration.h file remove the forward slashes in front of #define FILAMENT_RUNOUT_SENSOR line**



- Go to the next page.

Marlin 2.0.x Setup for Filament Runout Sensor

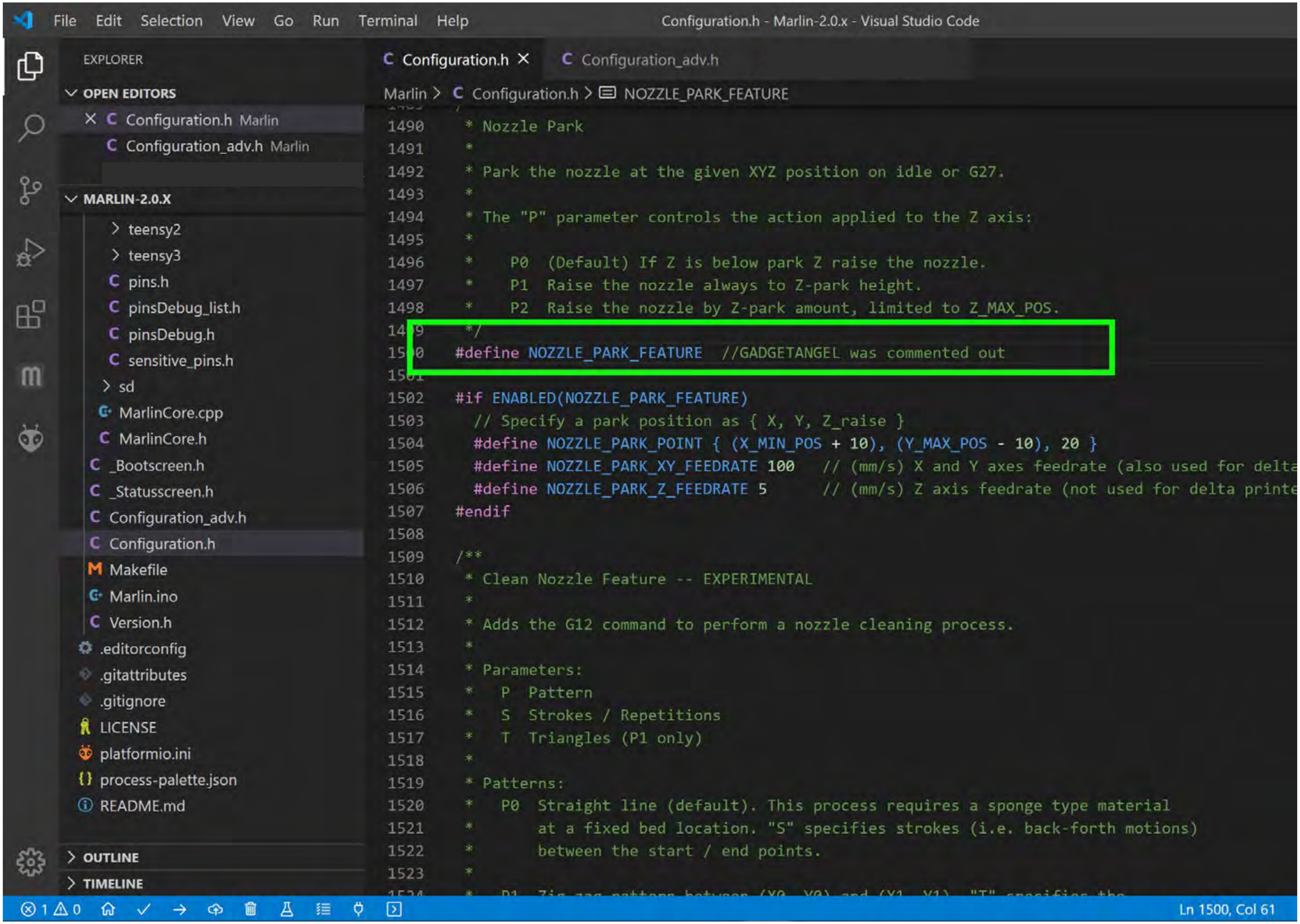
- **In Configuration.h file remove the forward slashes in front of #define FILAMENT_RUNOUT_SENSOR line**



- Go to the next page.

Marlin 2.0.x Setup for Filament Runout Sensor

- **In Configuration.h file remove the forward slashes in front of #define NOZZLE_PARK_FEATURE line**

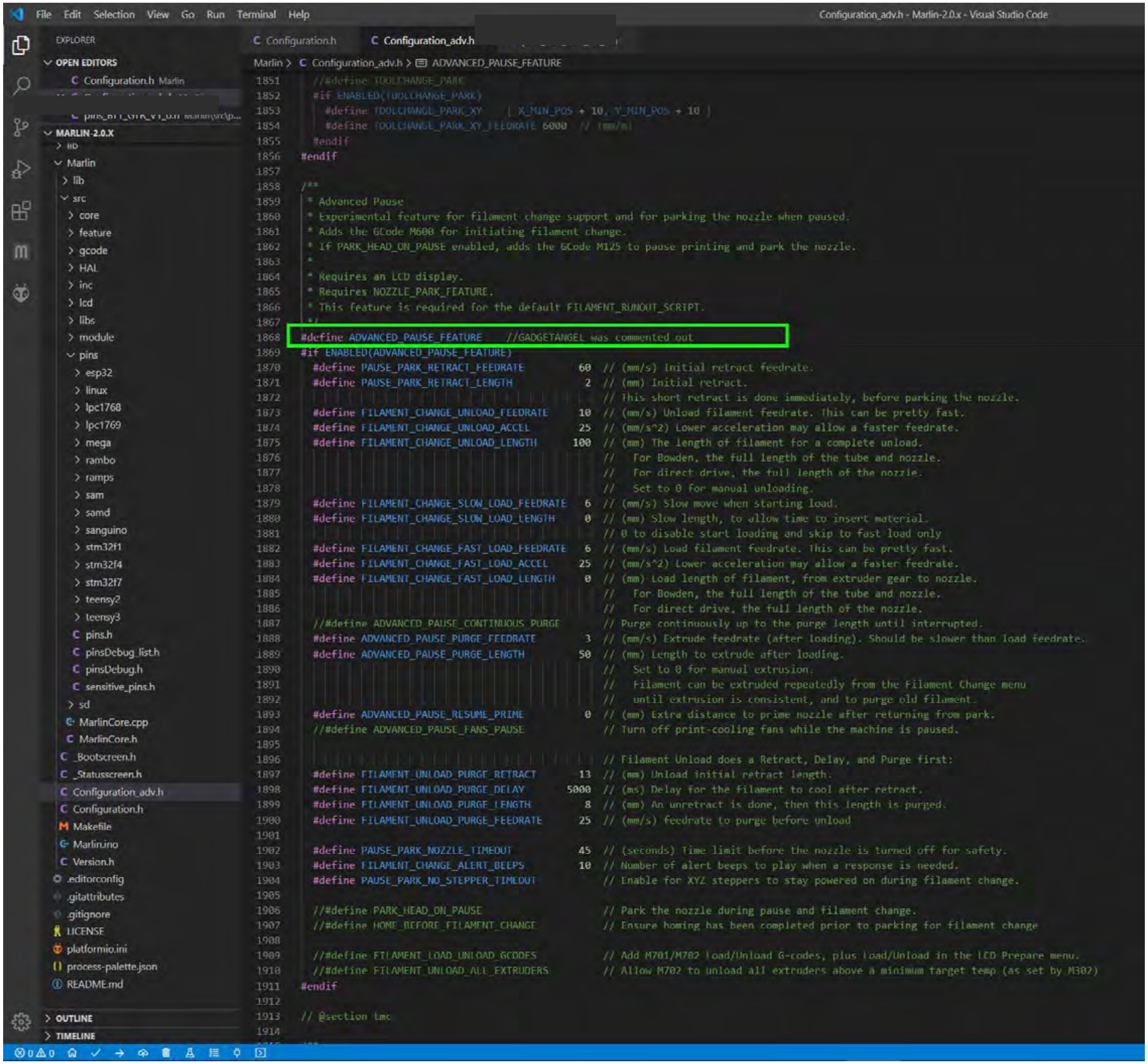


- Go to the next page.

APPENDIX H

Marlin 2.0.x Setup for Filament Runout Sensor

- **In Configuration_adv.h file remove the forward slashes before #define ADVANCED_PAUSE_FEATURE**



- Go to the next page.

Marlin 2.0.x Setup for Filament Runout Sensor

- **keep the Marlin settings the same (as above) until the filament sensor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the filament runout sensor!!**
- **Use a spare strip of filament and feed it through the filament runout sensor. Start a print, and then pull the filament strip out of the filament runout sensor.**
- **If filament runout sensor is working, the printer will have paused and if this was not a test, you would then load in new filament.**

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

Connecting SKR V1.3 with Pi 3B+/4 by 3 wires, (2 wires which are serial connection and 1 wire is ground [GND]). With 3 wires for serial connections made you no longer need the use of USB cable. *1

- **Raspberry Pi 3B+/4B and SKR V1.3 both use 3.3 V logic.**
- **The idea is to connect (see Wiring Diagram below) the TX and RX from the Raspberry Pi to the ARM Processor's RX and TX on the SKR V1.3 board.**

Therefore, do the following:

- **Connect TXD0 (Raspberry Pi's TX) from Raspberry Pi 3B+/4B connector TO the UART0_RX (RX0) from SKR V1.3 board's TFT or AUX-1 connectors.**
- **Connect RXD0 (Raspberry Pi's RX) from Raspberry Pi 3B+/4B connector TO the UART0_TX (TX0) from SKR V1.3 board's TFT or AUX-1 connectors.**
- **Connect GND from Raspberry Pi 3B+/4B Connector TO GND from SKR 1.3 board's TFT or AUX-1 connectors.**

DO NOT connect the Raspberry Pi 5V and 3.3V PINS to the SKR V1.3 board. Connecting the 5V and 3.3V PINS would bypass the fuses on the Raspberry Pi board and could cause issues to BOTH the Raspberry Pi board and the SKR V1.3 board. In fact, I recommend powering the Raspberry Pi from an independent and separate 5V power supply but connect the GND on the Raspberry Pi power supply to the GND on the SKR V1.3 board.

- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the areas on the Raspberry Pi board which are of interest when wanting to connect up the Raspberry Pi to the SKR V1.3 board. This is an overview. The diagram indicates the location of the TXD0 and RXD0 PINS on the Raspberry Pi.

Pi 3B+/4B @ GPIO connector: PINS used: TXD0, RXD0 and GND (8-10-6):

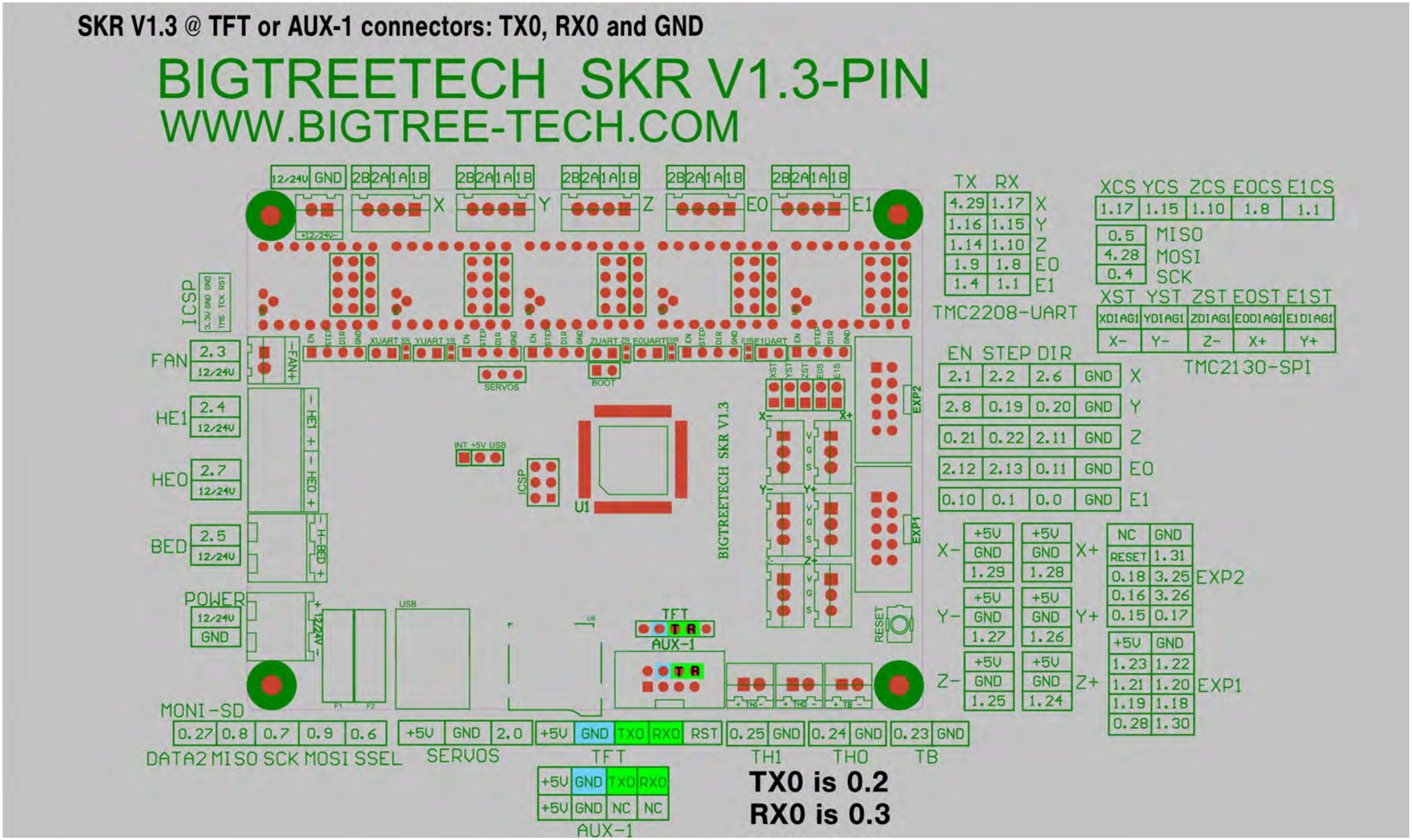
		Pin no.			
DC Power	3.3V	1	2	5V	DC Power
SDA1, I^2C	GPIO 2	3	4	5V	DC Power
SCL1, I^2C	GPIO 3	5	6	GND	
GPIO_GCLK	GPIO 4	7	8	GPIO 14	TXD0
	GND	9	10	GPIO 15	RXD0
GPIO_GEN0	GPIO 17	11	12	GPIO 18	GPIO_GEN1
GPIO_GEN2	GPIO 27	13	14	GND	
GPIO_GEN3	GPIO 22	15	16	GPIO 23	GPIO_GEN4
DC Power	3.3V	17	18	GPIO 24	GPIO_GEN5
SPI_MOSI	GPIO 10	19	20	GND	
SPI_MISO	GPIO 9	21	22	GPIO 25	GPIO_GEN6
SPI_CLK	GPIO 11	23	24	GPIO 8	SPI_CE0_N
	GND	25	26	GPIO 7	SPI_CE1_N
I^2C ID EEPROM	DNC	27	28	DNC	I^2C ID EEPROM
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	
	GND	39	40	GPIO 21	

- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the Raspberry Pi to the SKR V1.3 board. This is an overview. The diagram indicates which PINS are involved.

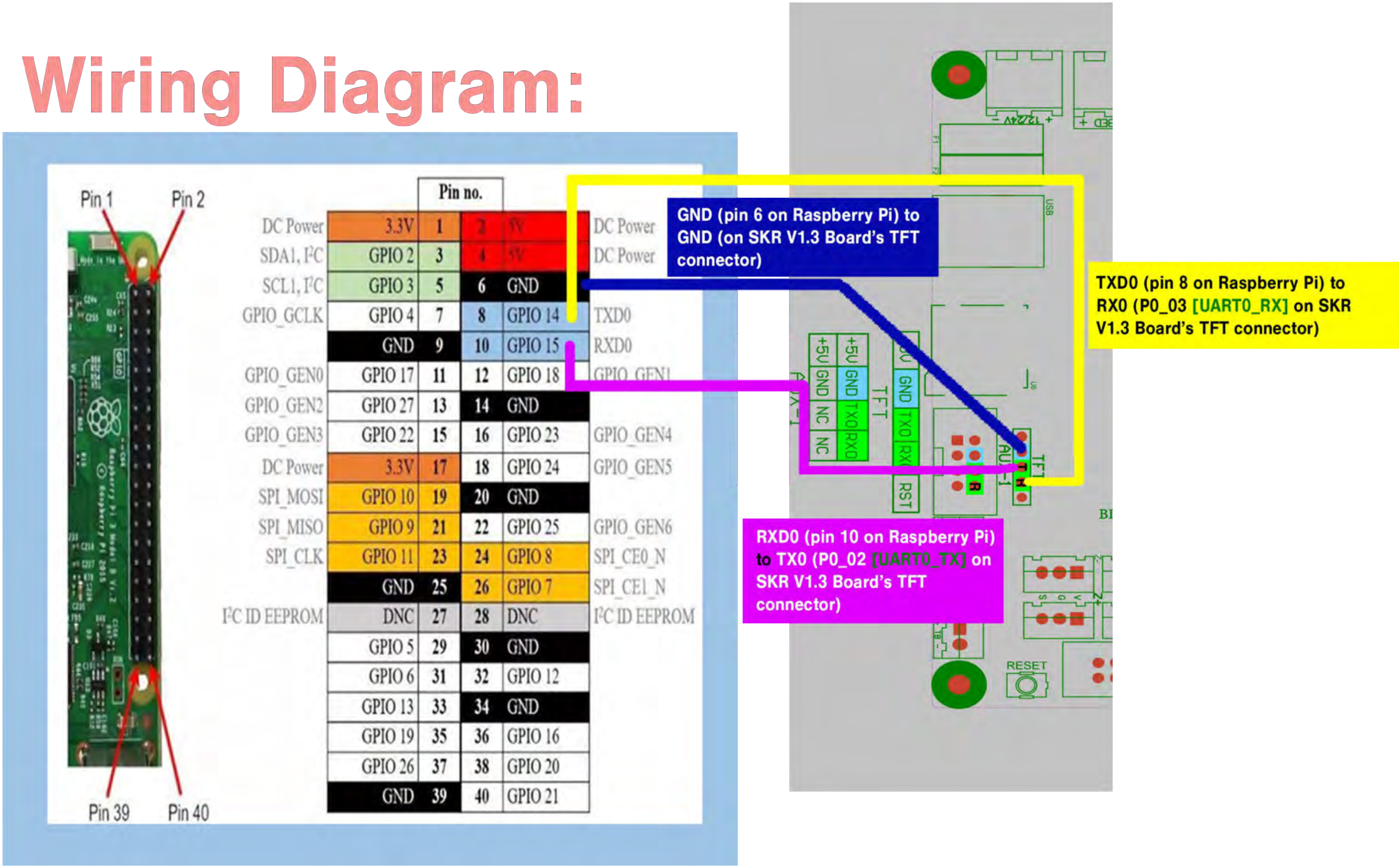


- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the shows the interconnection between the Raspberry Pi and the SKR V1.3 board. This diagram indicates which PINS are involved and how to connect them together.



- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

You also have to change some settings in the Raspberry pi 3B+/4B operating system.

Raspberry Pi 3B+/4B Changes that need to be made:

Login to your Raspberry Pi.

Swapping ports used by GPIO and Bluetooth

The first thing to change in the serial connection is to swap the ports used by the GPIO pins and the internal Bluetooth chip. We need to add a line in the boot config file on the boot partition.

Log in to the Pi with SSH on the IP address used.

Type in sudo nano /boot/config.txt

Move the cursor to the end of the file by cursor and add:

dtoverlay=pi3-miniuart-bt

save the file and exit the editor by control+O, hit enter then hit control+X

Disabling the serial console

Moving to another config file, where part of the code must be deleted to disable serial console.

Type in sudo nano /boot/cmdline.txt

Look for following string (text) and delete it

console=serial0,115200

Note: This baud rate must be the same speed set in Marlin and in Octoprint!

save the file and exit the editor by control+O, hit enter then hit control+X

Rebooting RPi

For all changes to take effect, please reboot your Raspberry Pi

Type in Sudo reboot

Adding the serial port in Octoprint

Last part of the configuration is in the Octoprint web interface. Open your browser and type either "octoprint.local" or the IP address of the Pi. You might be greeted with the welcome wizard, please go through it first.

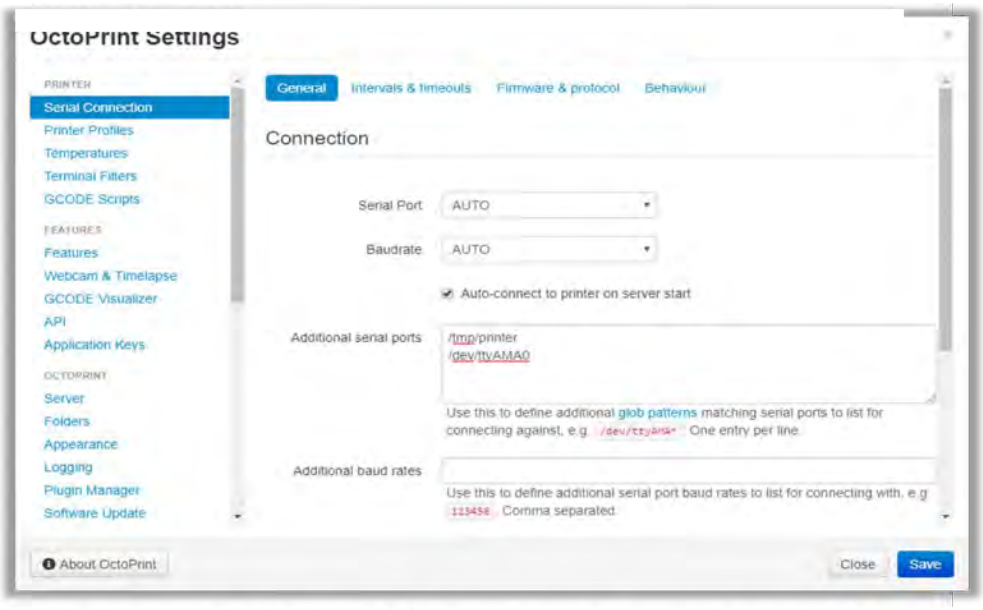
As soon as you arrive at the home screen, open "Settings" (top right), head to "Serial Connection", then "Additional serial ports" and insert following:

/dev/ttyAMA0

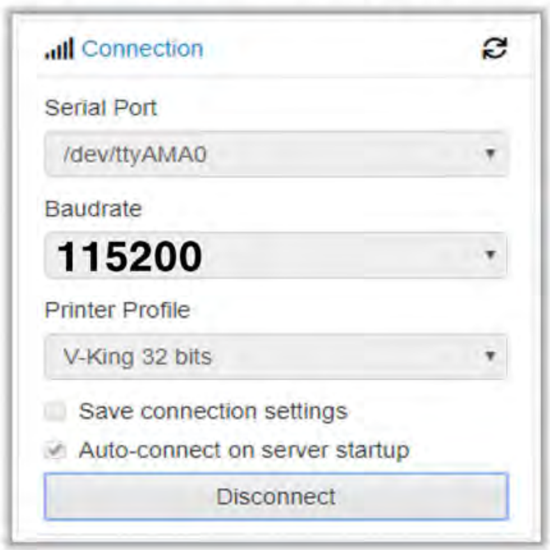
- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable



Save the change and reboot OctoPrint. After reboot, select the new port and connect to your printer. Making connection might take a few seconds longer then with USB.



Note: This baud rate must be the same speed set in Marlin and for the console speed on the Raspberry Pi!

***1 This information was taken from BIGTREETECH SKRV1.3 Guide 2019 - 6 .PDF**

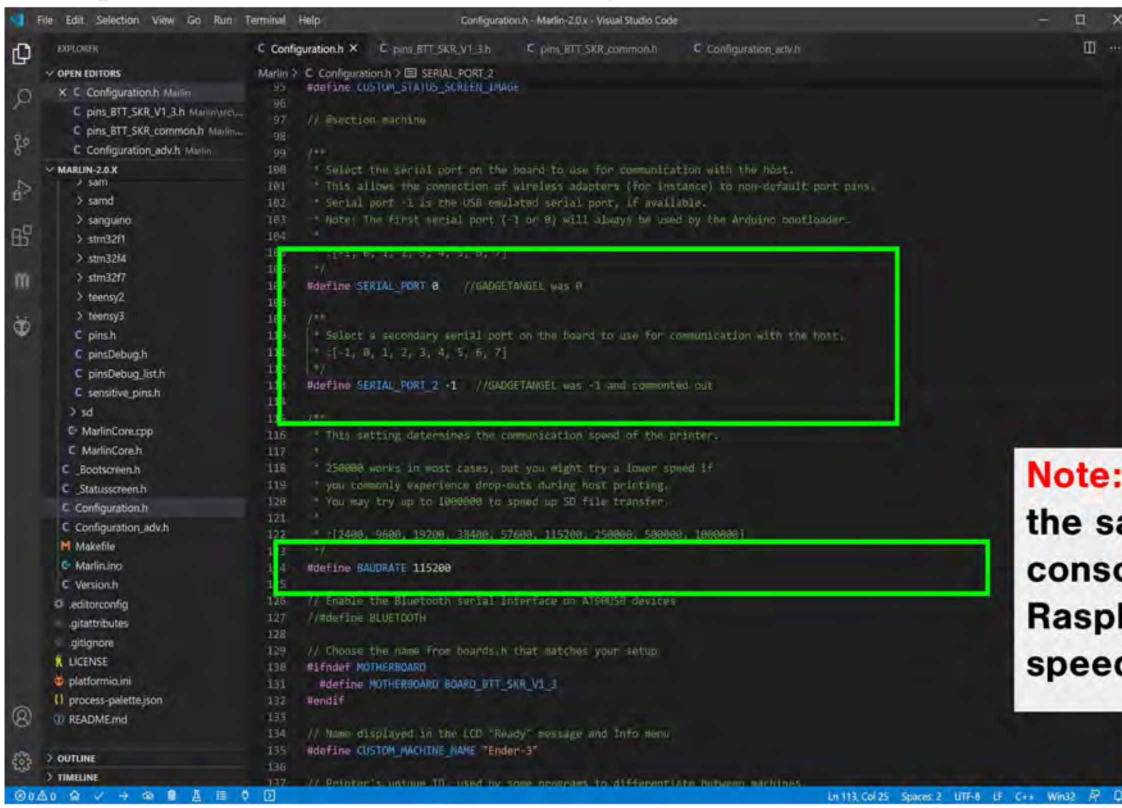
- Go to the next page.

APPENDIX I

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**

    - **SKR V1.3 board has only 2 numbers that can be used for SERIAL_PORT or SERIAL_PORT_2 Marlin variables: -1 for USB port; 0 for TFT or AUX-1 port.**
    - **The SERIAL_PORT in Marlin is the one you want to use to get ALL errors reported. The SERIAL_PORT_2 is the secondary port. The secondary port will only receive errors on the connections the seondaary port sends out on.**
    - **In Configuration.h set SERIAL_PORT to 0 and SERIAL_PORT_2 to -1 using these setting Marlin will use the TX0/RX0 on the TFT connector to communicate with Octoprint. The USB port (-1) will receive secondary information.**



Note: This baud rate must be the same speed set for the console speed on the Raspberry Pi and the same speed set in Octoprint!

Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

- **keep the Marlin settings the same (as above) until the serial communication with the Raspberry Pi is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the serial connection with the Raspberry Pi!!**

Connecting SKR V1.3 with BLTouch

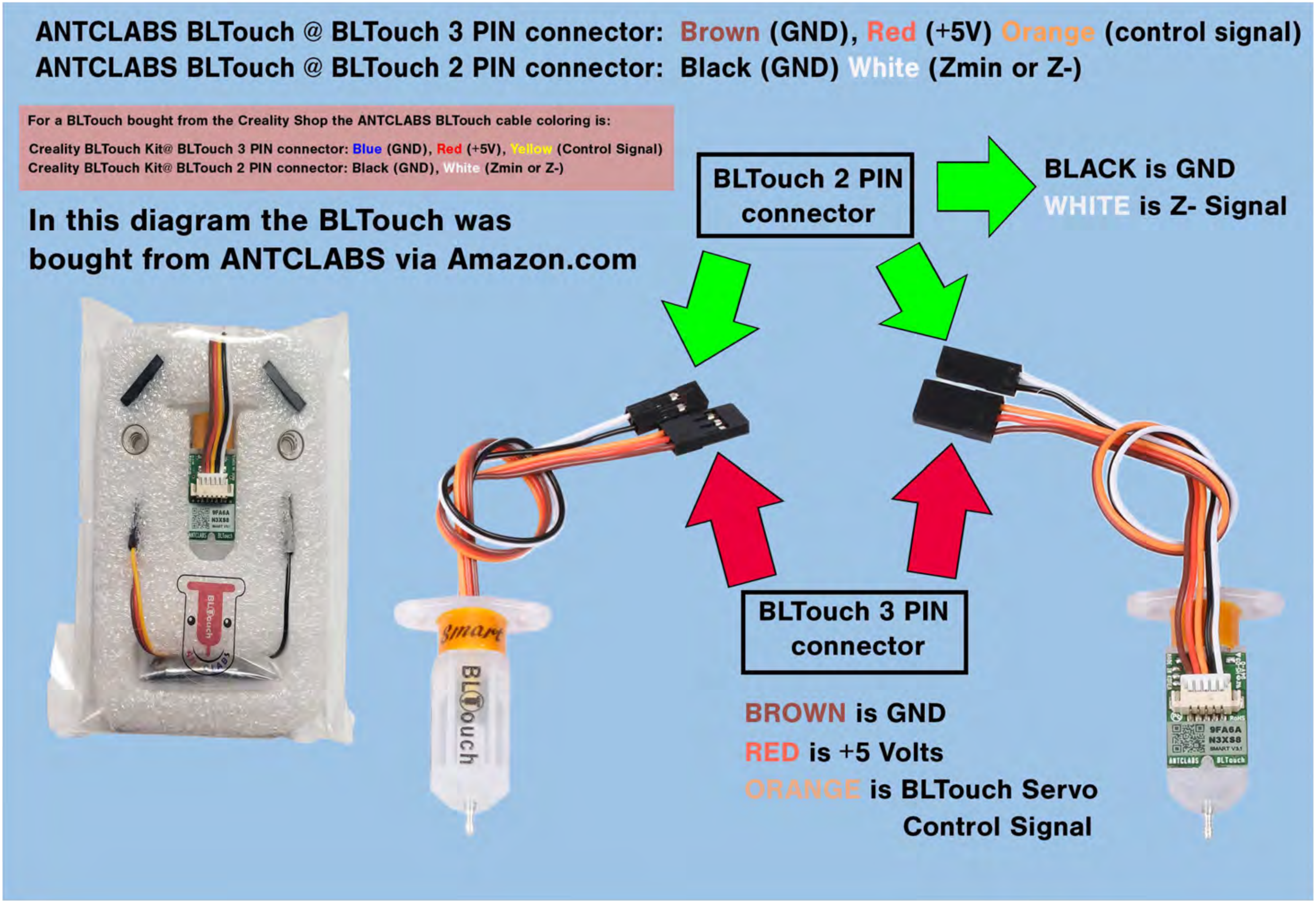
To install a BLTouch to your printer requires three steps:

- **Mount the BLTouch to your printer's print head. Search Thingiverse for a BLTouch mount for your specific printer and print head.**
- **Wiring the BLTouch to the MCU board (SKR V1.3 board).**
- **Marlin 2.0.x Firmware setup.**

- Go to the next page.

APPENDIX J
Connecting SKR V1.3 with BLTouch

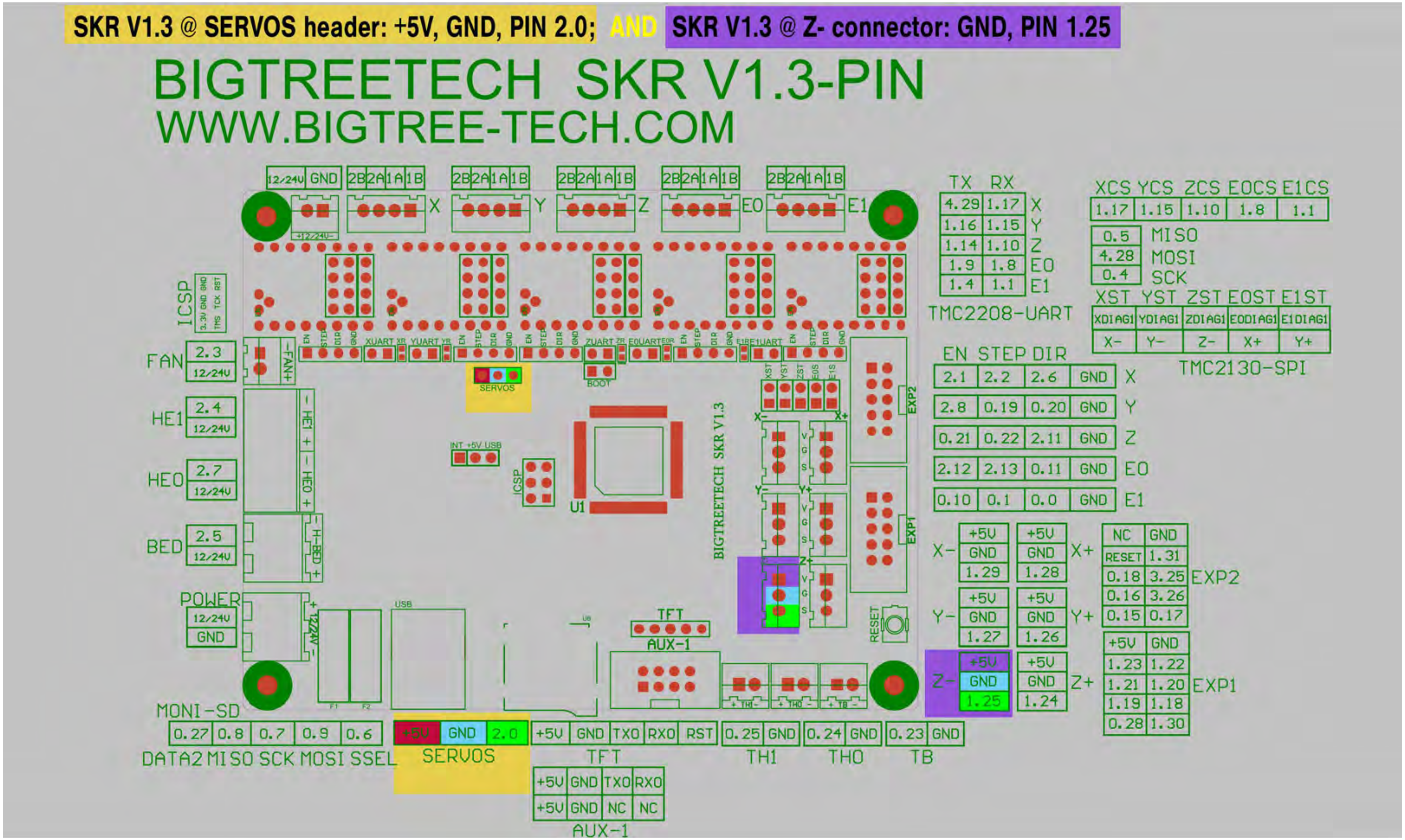
The diagram below shows the areas on the BLTouch Sensor which are of interest when wanting it up to the SKR V1.3 board. This is an overview. The diagram indicates the location of the PINS on the BLTouch that will be used when connecting it up to the SKR V1.3 board.



- Go to the next page.

APPENDIX J
Connecting SKR V1.3 with BLTouch

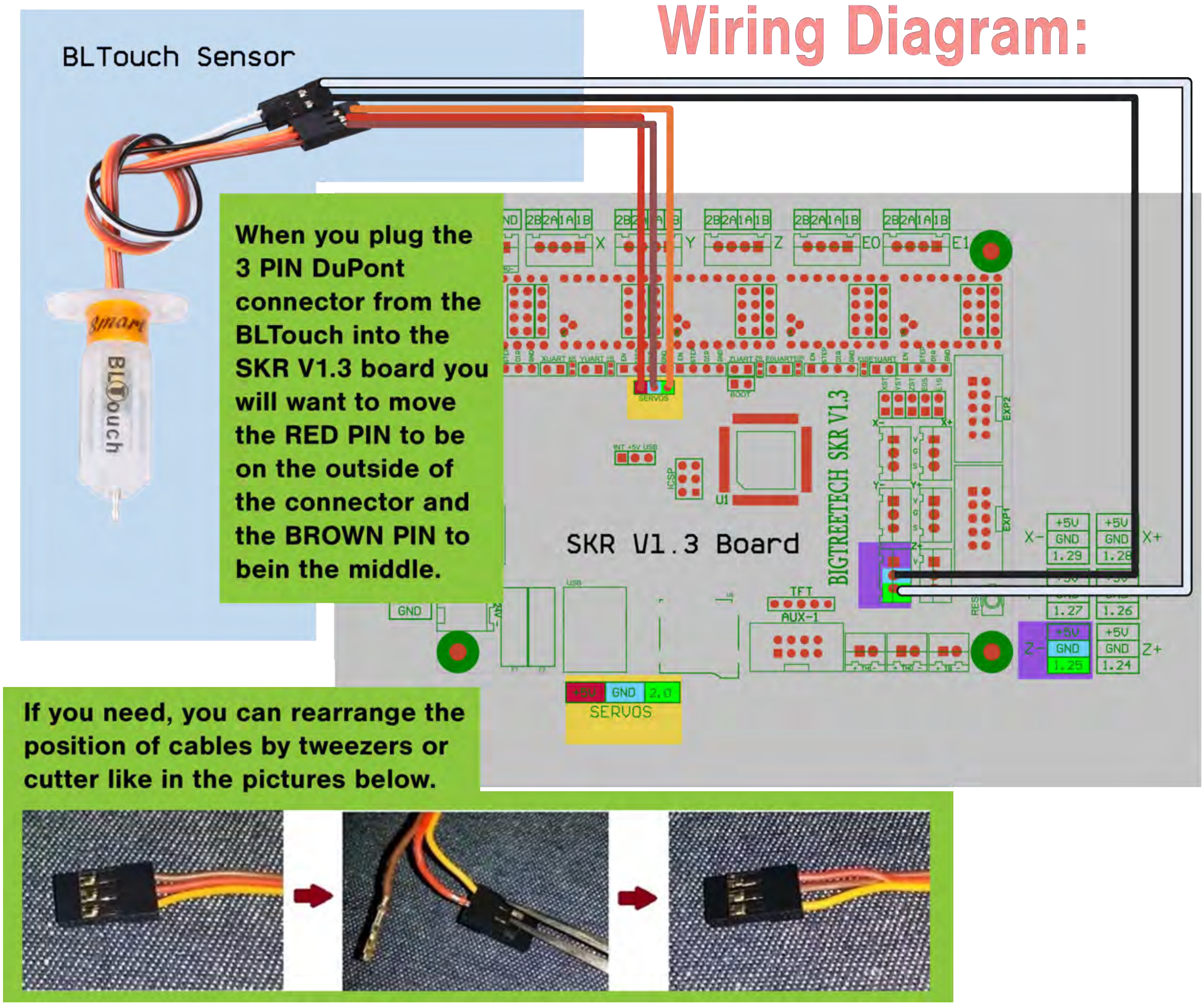
The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the BLTouch to the SKR V1.3 board. This is an overview. The diagram indicates which PINS are involved.



- Go to the next page.

APPENDIX J
Connecting SKR V1.3 with BLTouch

The diagram below shows the shows the interconnection between the BLTouch and the SKR V1.3 board. This diagram indicates which PINS are involved and how to connect them together.

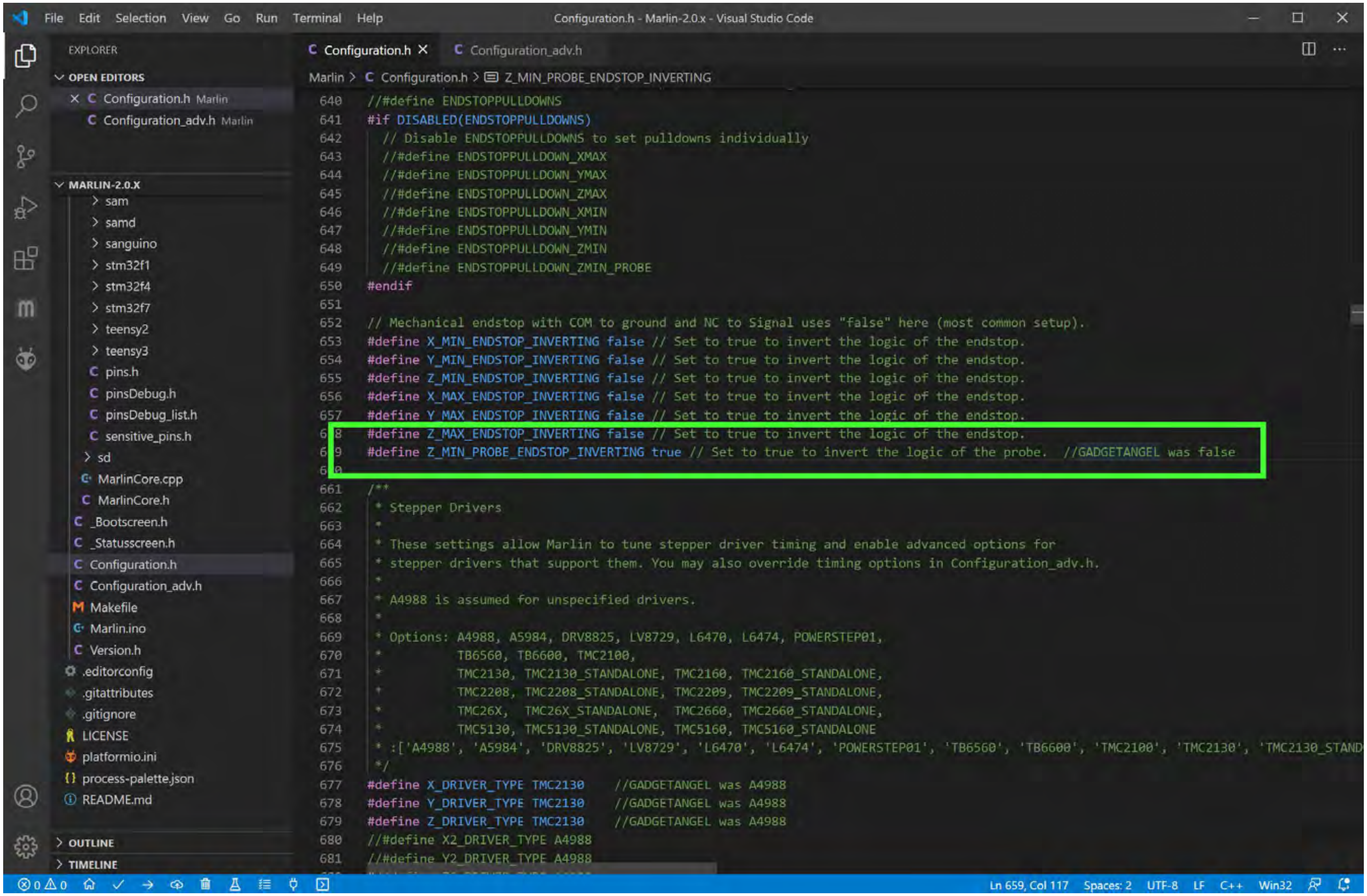


- Go to the
- next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**

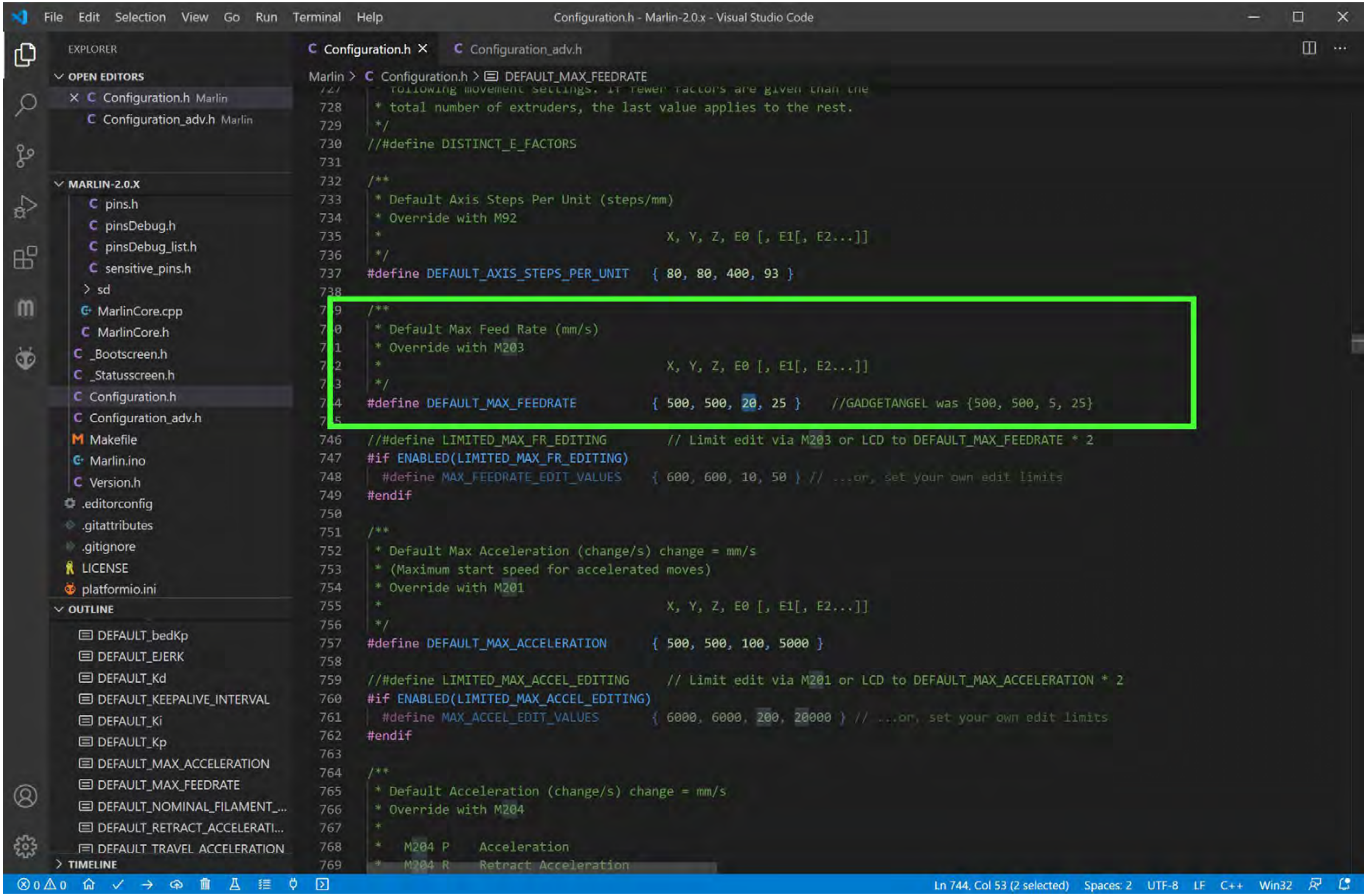
- **change the Marlin variable Z_MIN_PROBE_ENDSTOP_INVERTING to true**



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **change the Z value of the Marlin variable DEFAULT_MAX_FEEDRATE to 20**

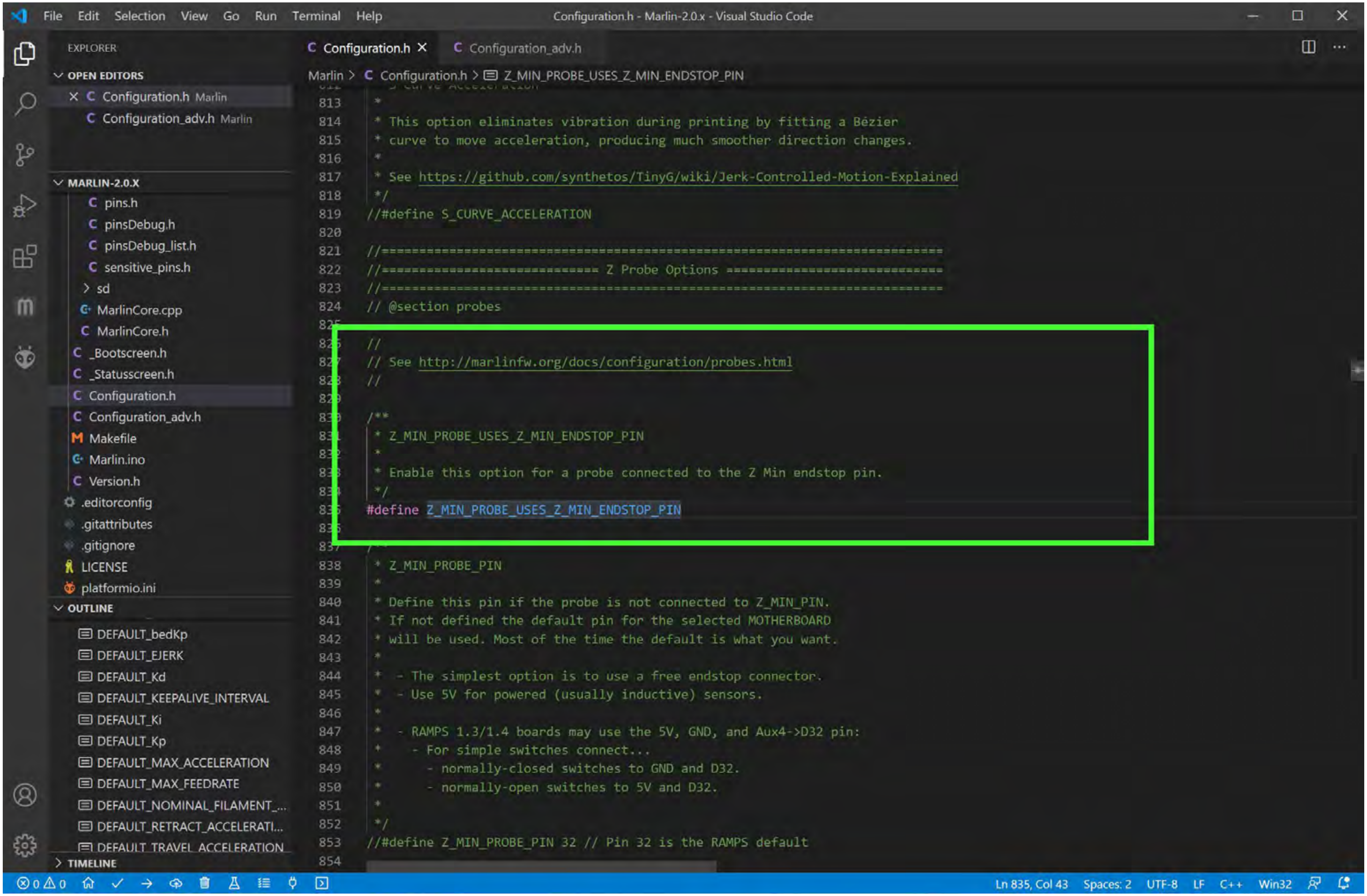


- Go to the next page.

APPENDIX J

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **ensure the Marlin variable Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN is enabled or that you delete the two forward slashes ("//") from the beginning of the line.**

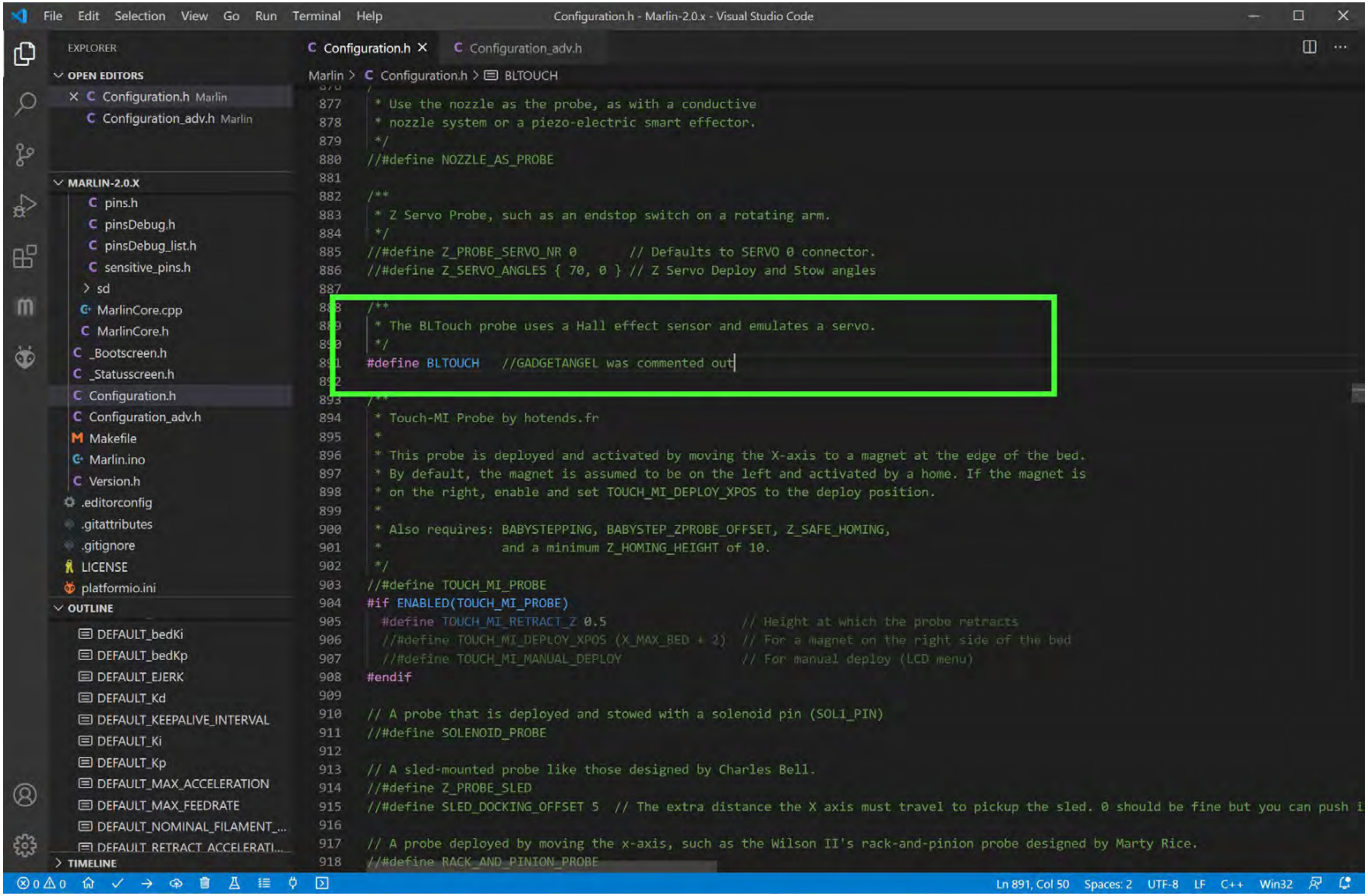


- Go to the next page.

APPENDIX J

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **enable the Marlin variable BLTOUCH or that you delete the two forward slashes ("//") from the beginning of the line.**



- Go to the next page.

APPENDIX J

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

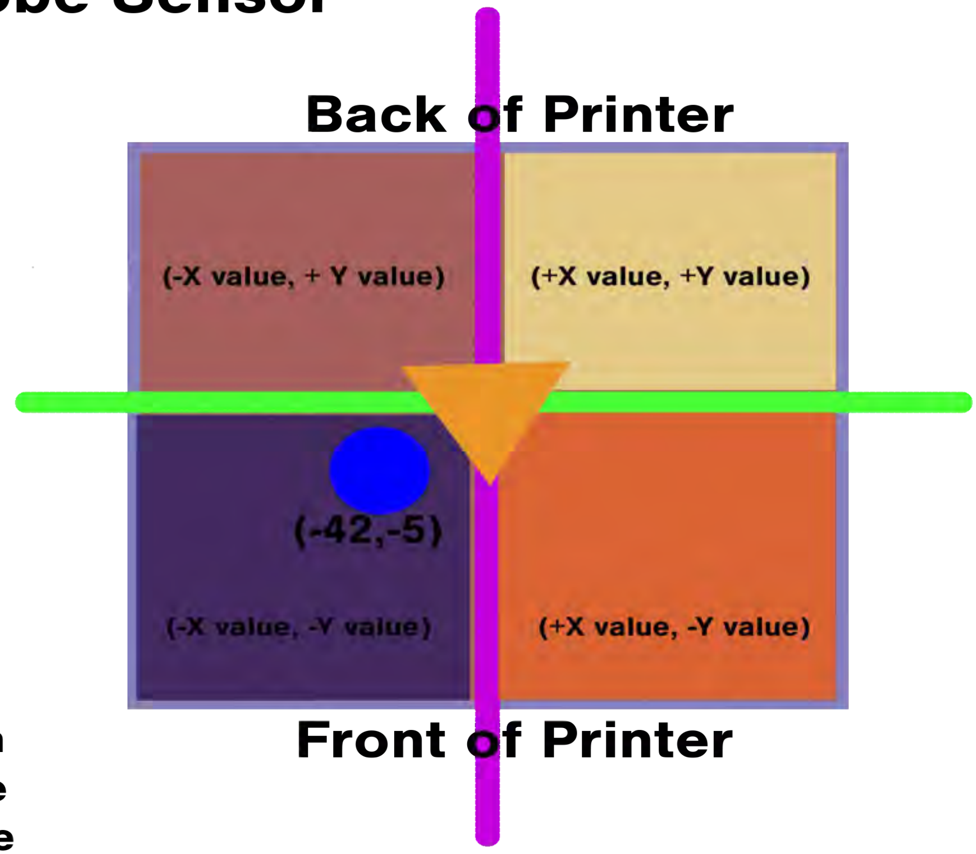
Understanding Marlin Firmware's NOZZLE_TO_PROBE_OFFSET Setting

- NOZZLE - BLTouch Probe Sensor - Printer Bed

The BLTouch Probe Sensor can be located in one of four quadrants. The Nozzle is at the center.

The diagram on this page shows the X and Y values for each quadrant.

If the BLTouch is located in FRONT of the Green Line or in FRONT of the NOZZLE then the Y value for the Probe-to-Nozzle-Offiset will be a negative (-) value. If the BLtouch is located BEHIND the Green Line or BEHIND the NOZZLE then the Y value for the Probe-to-Nozzle-Offset will be a positive (+) value. If the BLTouch is located to the LEFT of the Pink Line or to the LEFT of the NOZZLE then the X value for the Probe-to-Nozzle-Offset will be a negative (-) value. If the BLTouch is located to the RIGHT of the Pink Line or to the RIGHT of the NOZZLE then the X value for the Probe-to-Nozzle-Offset will be a positive (+) value.

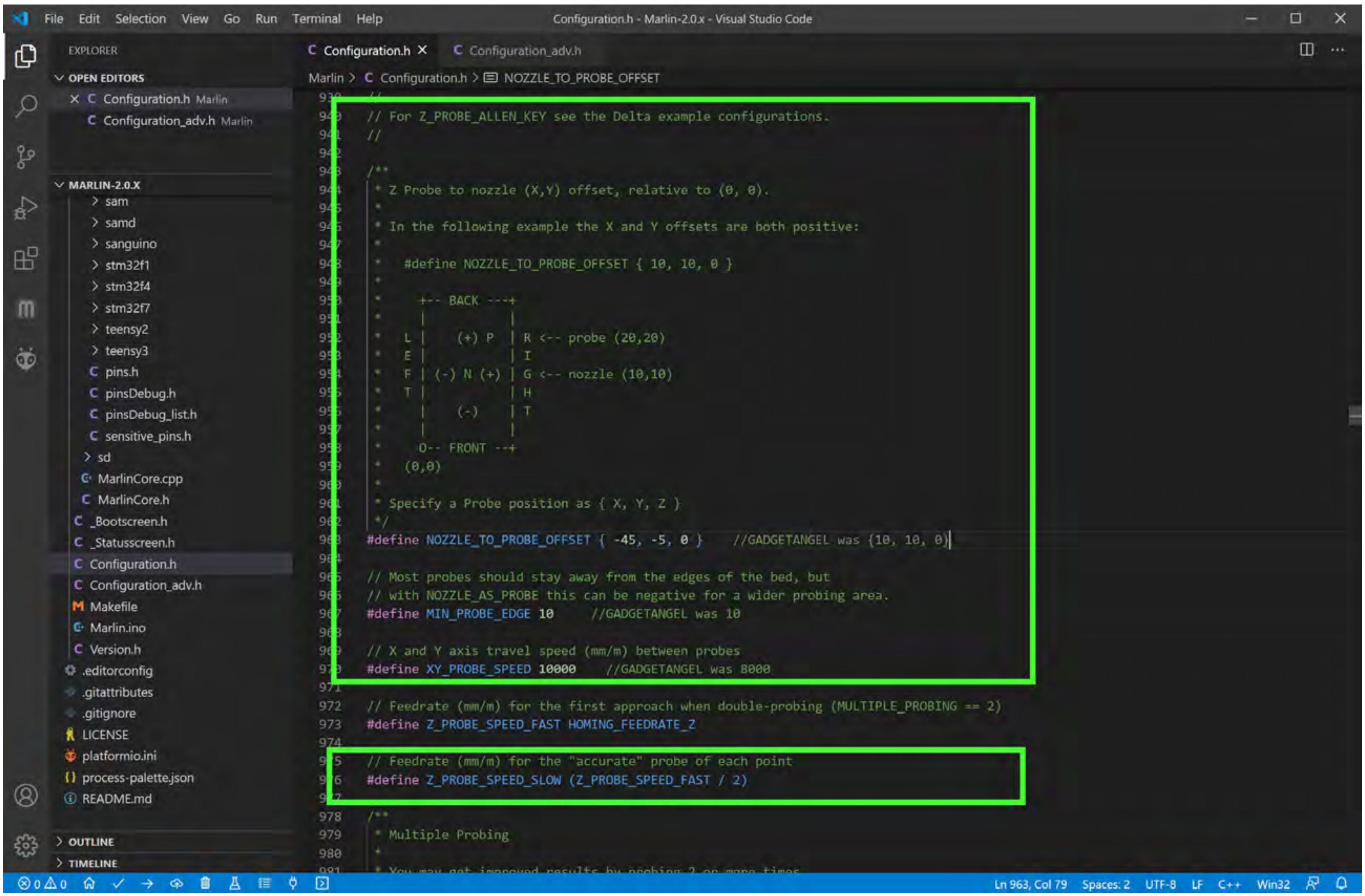


In the above diagram the BLTouch will have a -X value and -Y value for the NOZZLE_TO_PROBE_OFFSET values. In the above example, the Probe-to-Nozzle-Offset values are -42 for X and -5 for Y.

- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

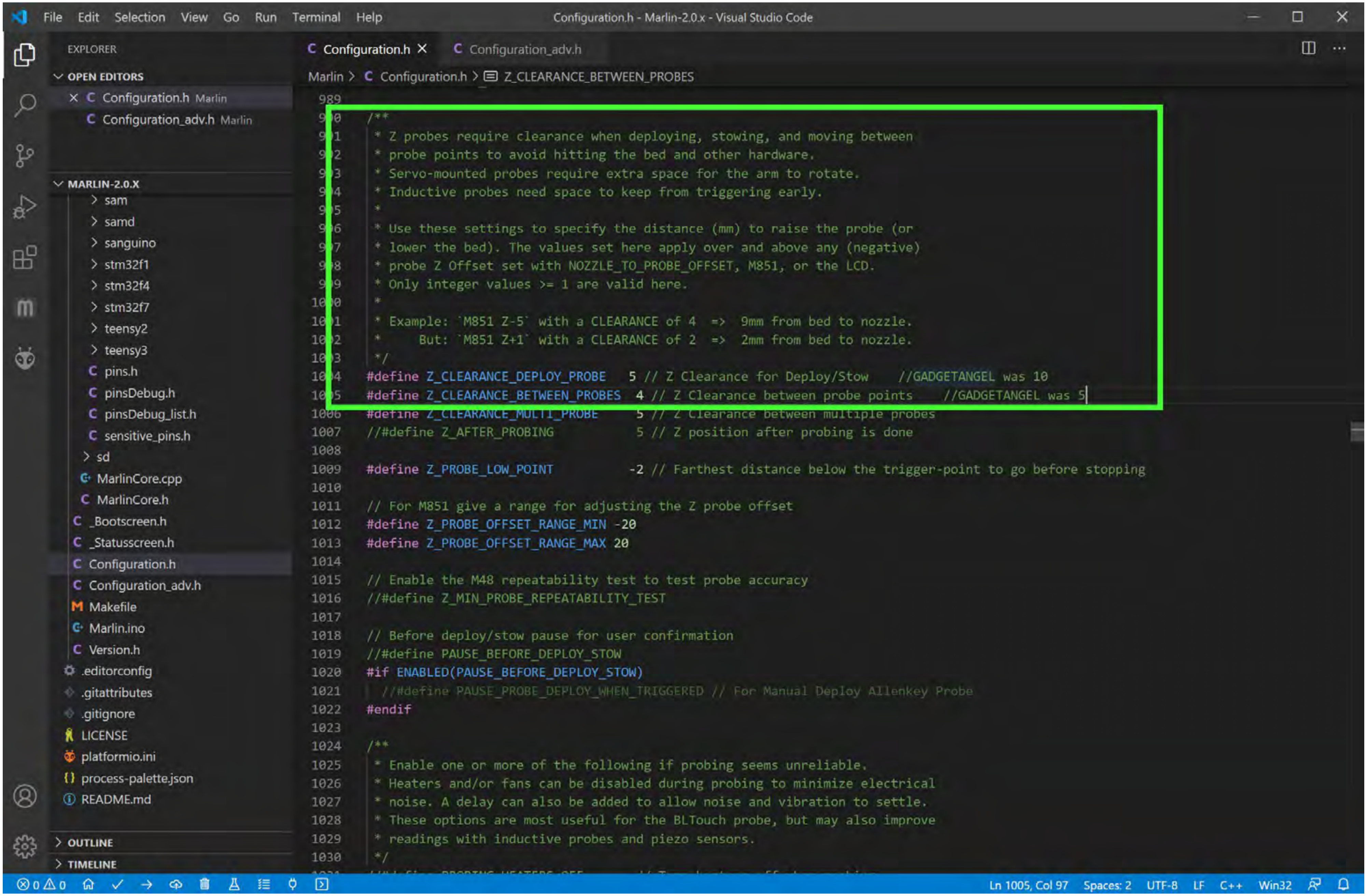
- change the X and Y values of the Marlin variable NOZZLE_TO_PROBE_OFFSET to match your measurements from your 3D printer. In the example shown from the previous page, the X value is equal to -42 and the Y value is equal to -5.
- set the Marlin variable MIN_PROBE_EDGE is set for 10
- set the Marlin variable XY_PROBE_SPEED to 10000
- ensure the Marlin variable Z_PROBE_SPEED_SLOW is equal to Z_PROBE_SPEED_FAST / 2



- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

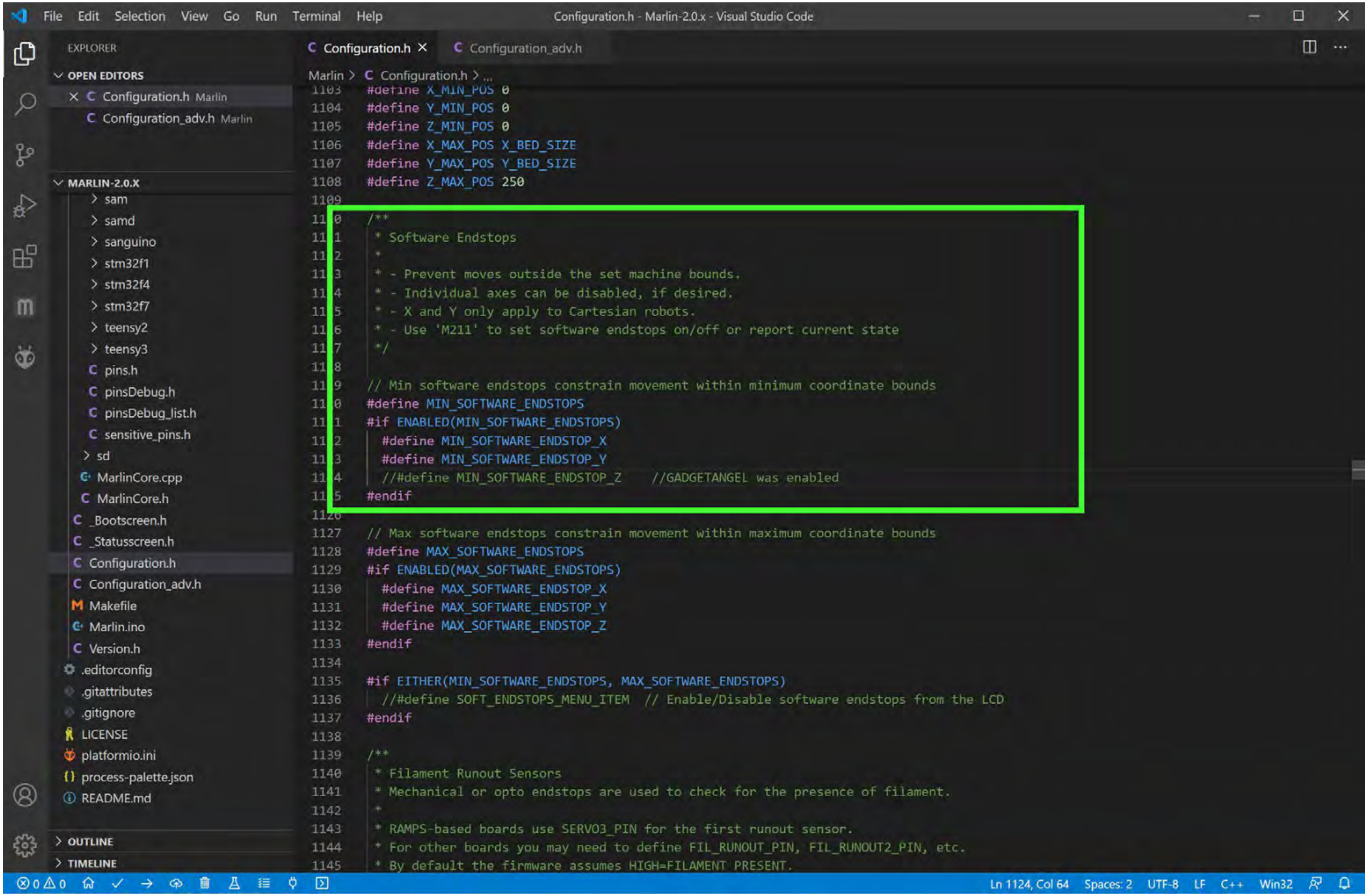
- **change the Marlin variable Z_CLEARANCE_DEPLOY_PROBE to 5**
- **change the Marlin variable Z_CLEARANCE_BETWEEN_PROBES to 4**



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

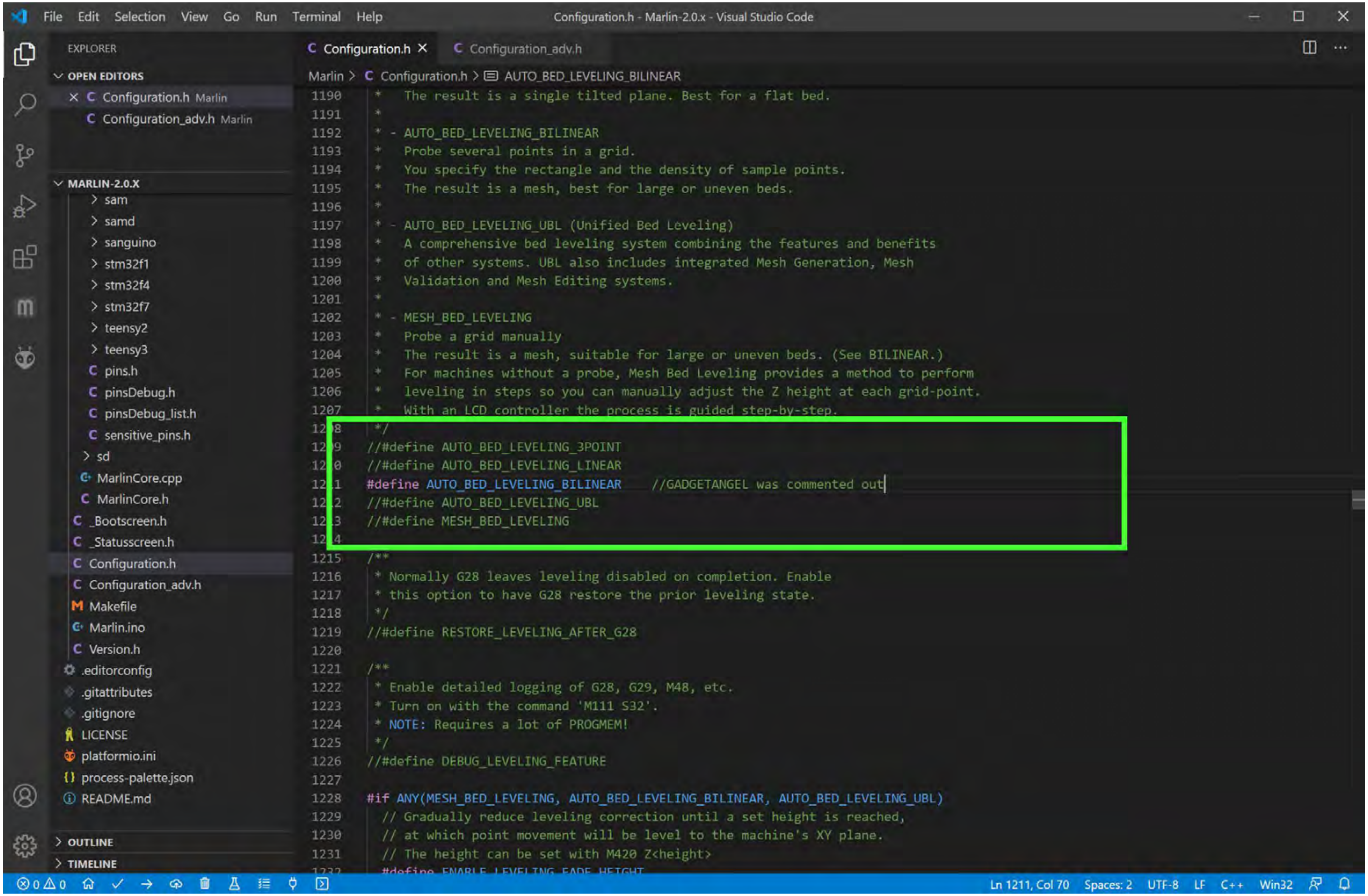
- **disable** the Marlin variable **MIN_SOFTWARE_ENDSTOP_Z** or that you comment out the line by **inserting two forward slashes** ('//') at the beginning of the line.



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

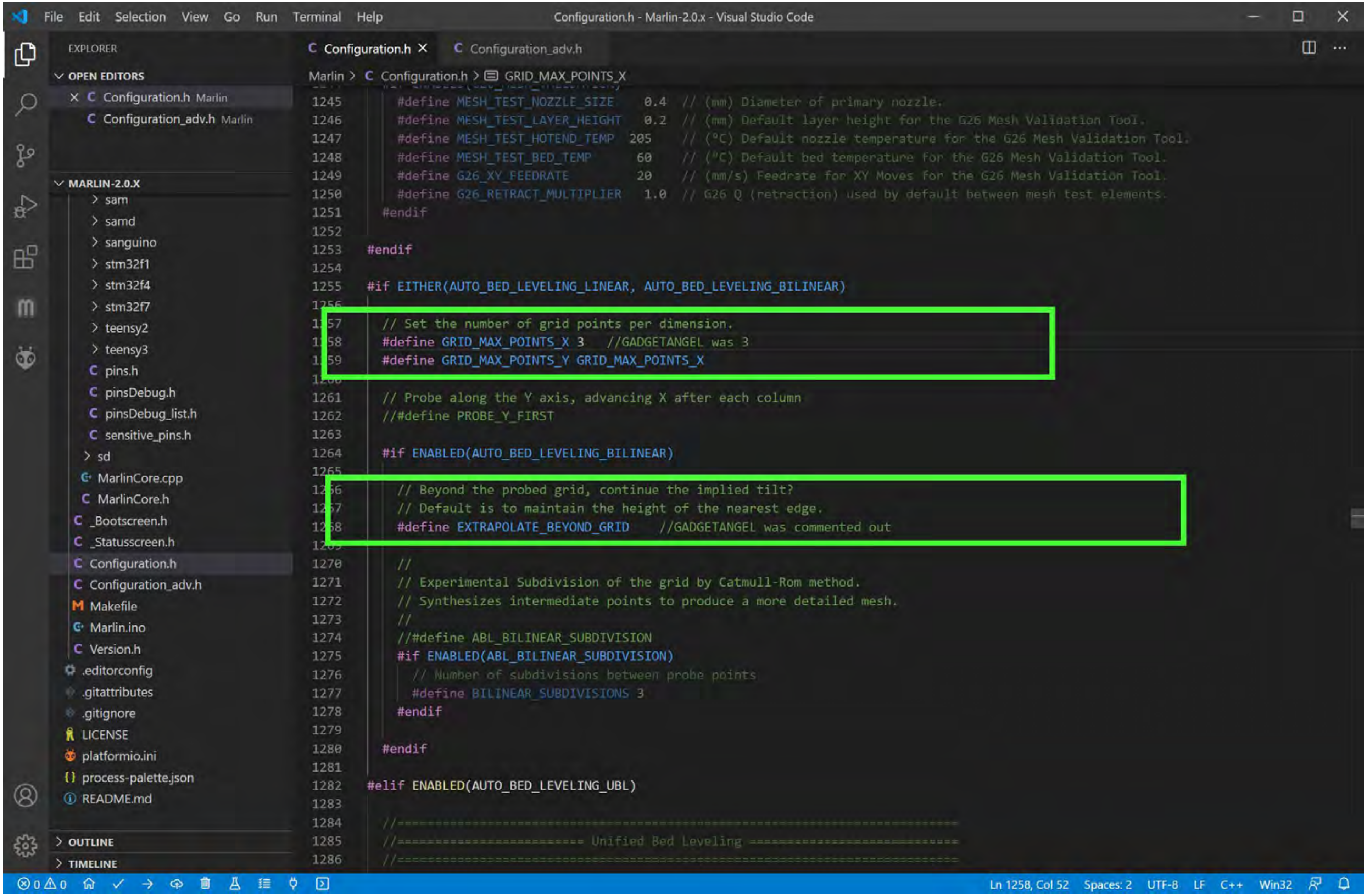
- **enable** the Marlin variable **AUTO_BED_LEVELING_BILINEAR** or that you **delete the two forward slashes** ("//") from the beginning of the line.



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

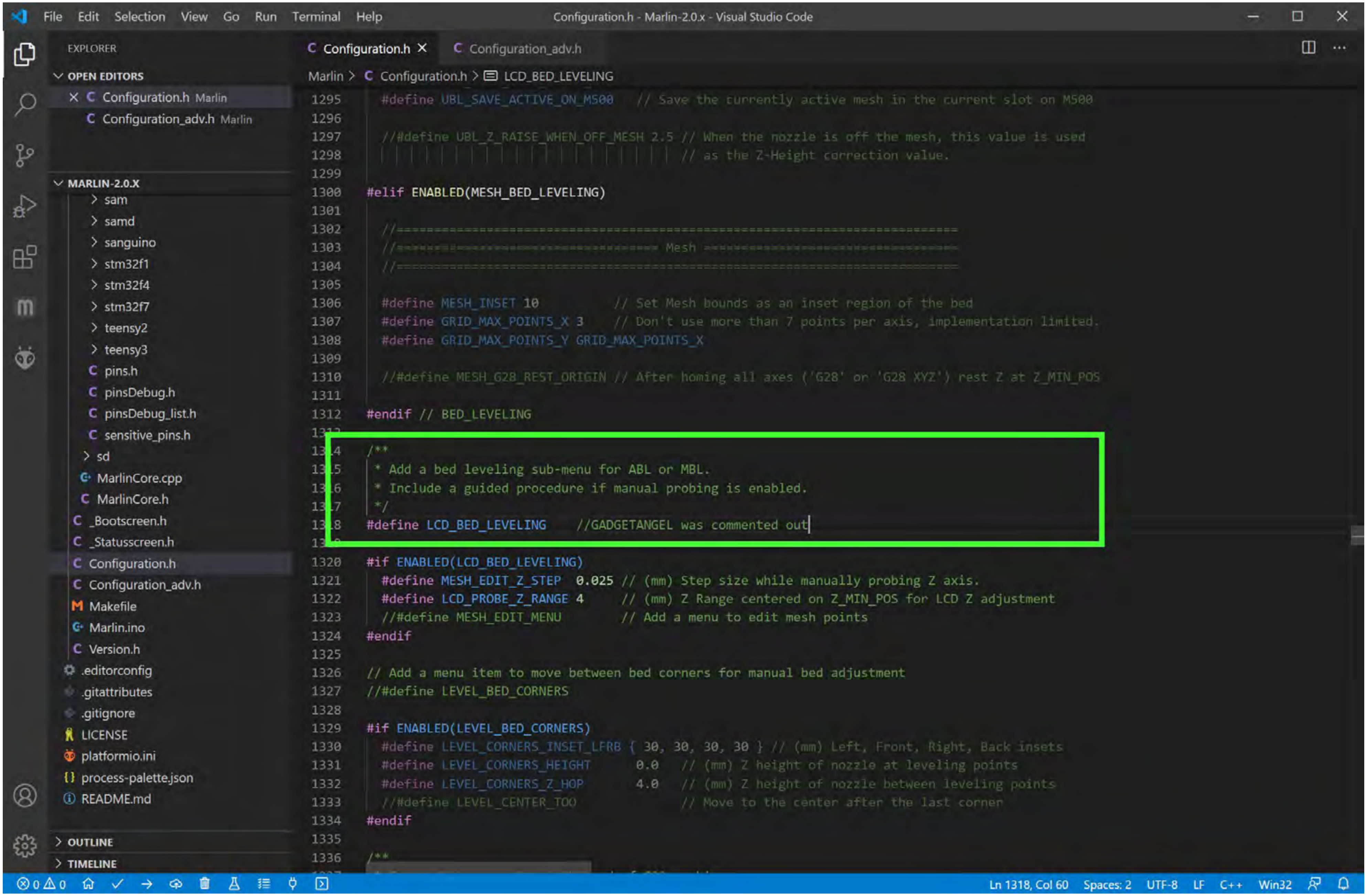
- **ensure the Marlin variable GRID_MAX_POINTS_X is set to 3**
- **enable the Marlin variable EXTRAPOLATE_BEYOND_GRID or that you delete the two forward slashes ("//") from the beginning of the line.**



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **enable the Marlin variable LCD_BED_LEVELING or that you delete the two forward slashes ("//") from the beginning of the line.**

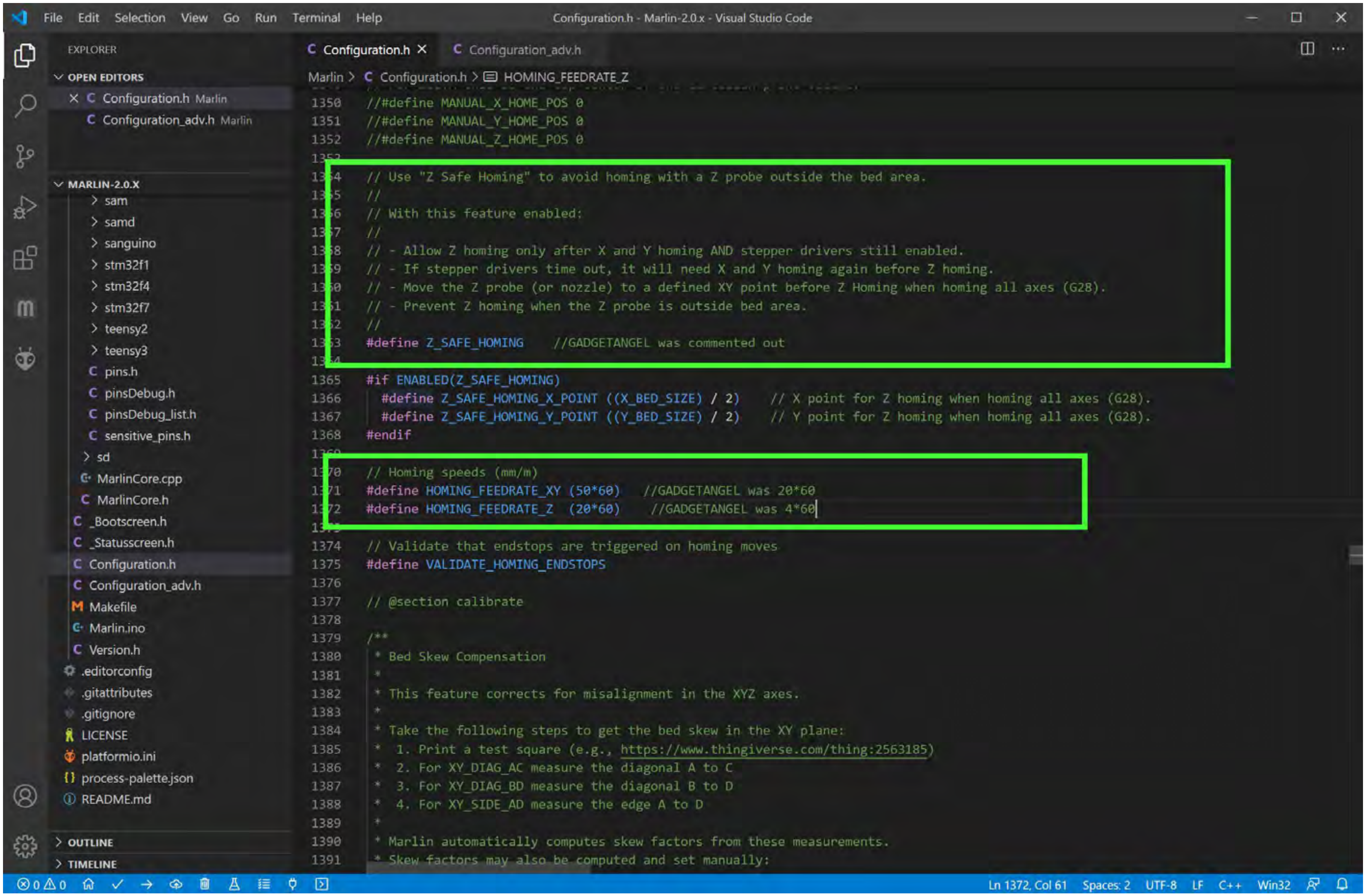


- Go to the next page.

APPENDIX J

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

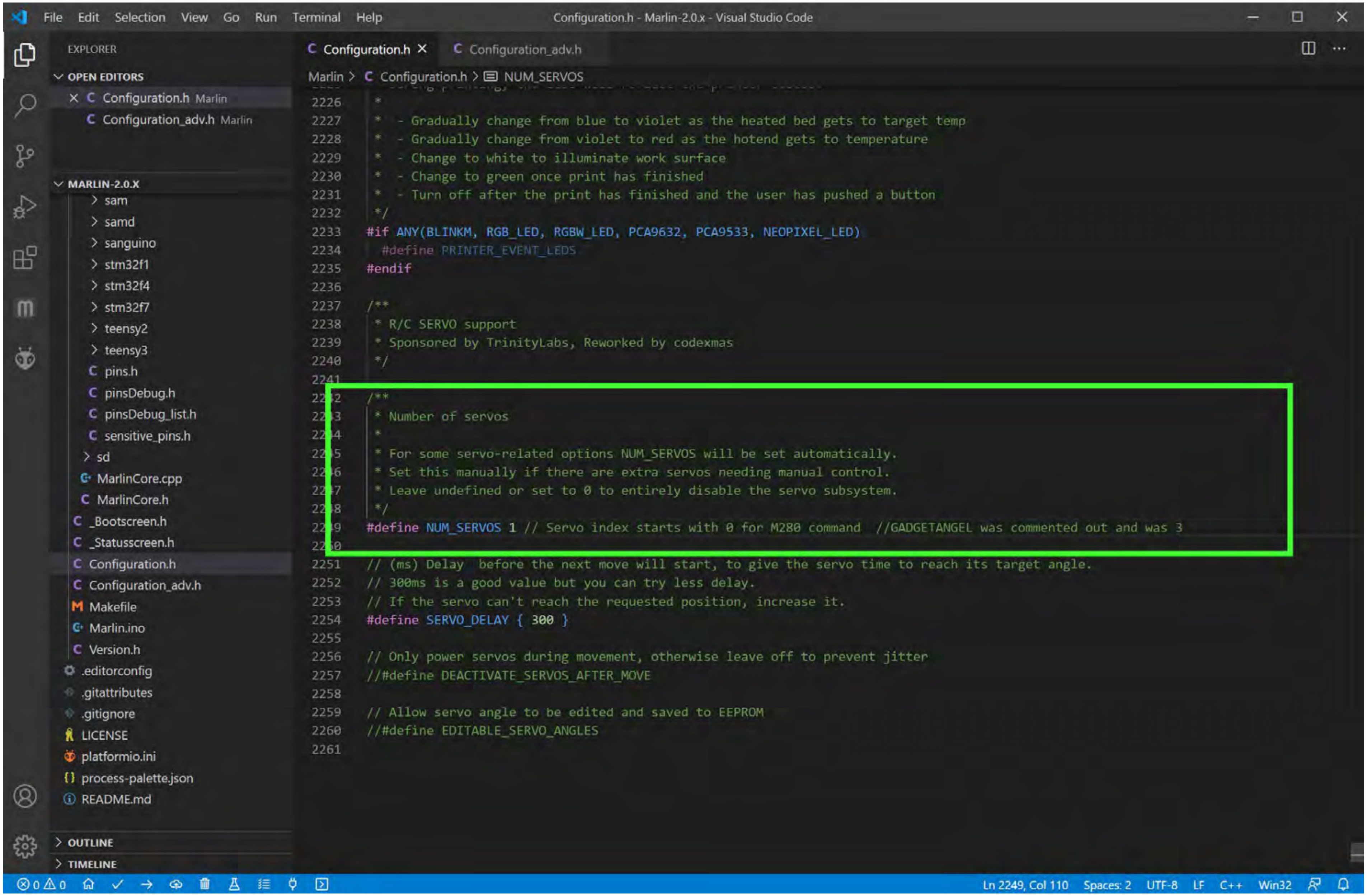
- **enable the Marlin variable Z_SAFE_HOMING or that you delete the two forward slashes ("//") from the beginning of the line.**
- **set the Marlin variable HOMING_FEEDRATE_XY to 50*60**
- **set the Marlin variable HOMING_FEEDRATE_Z to 20*60**



- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

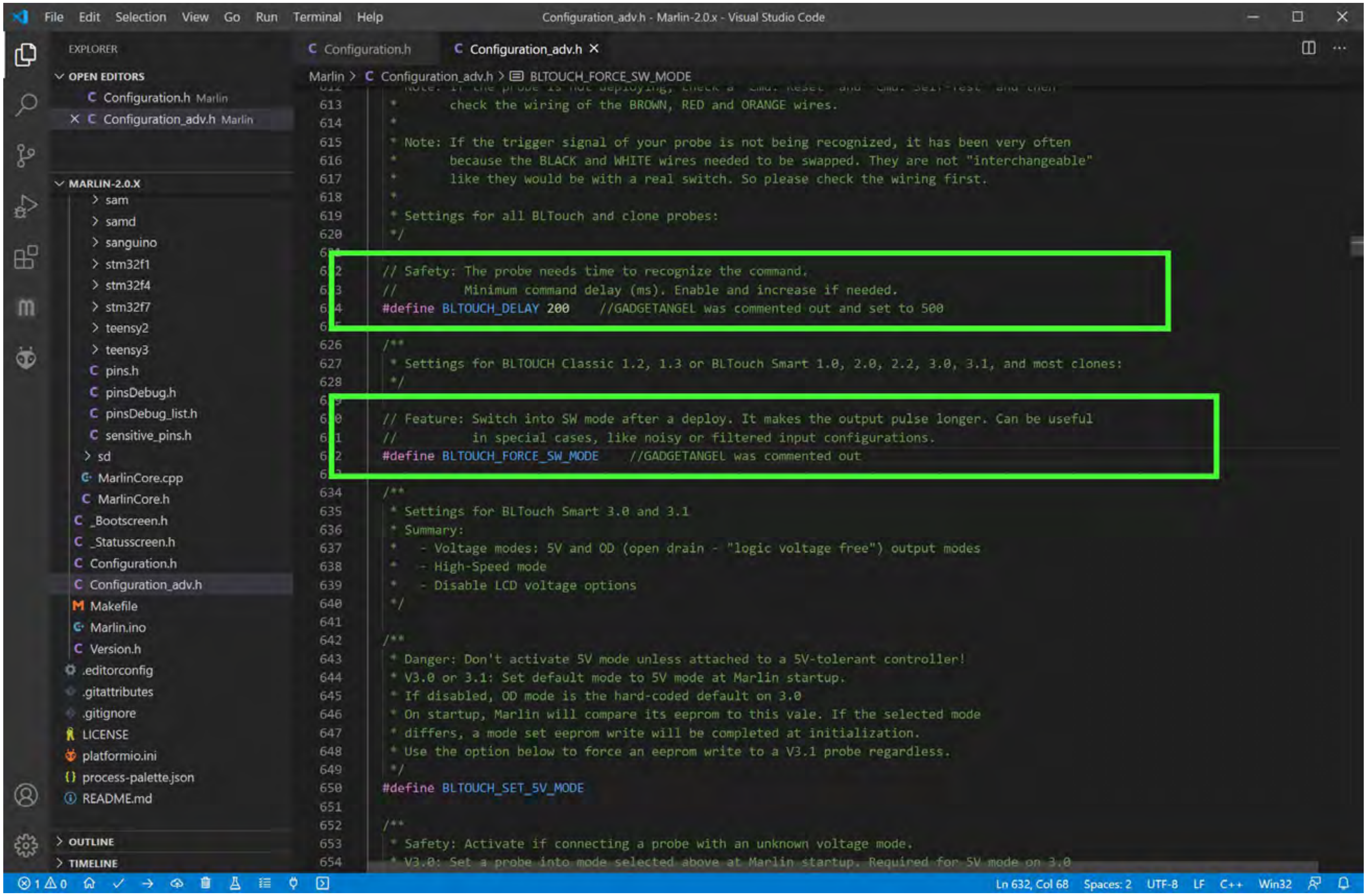
- **enable the Marlin variable NUM_SERVOS or that you delete the two forward slashes ("//") from the beginning of the line.**
- **change the Marlin variable NUM_SERVOS to 1**



- Go to the next page.

APPENDIX J
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **All the following changes are made to the Configuration_adv.h file**
- **enable the Marlin variable BLTOUCH_DELAY or that you delete the two forward slashes ("//") from the beginning of the line.**
- **change the Marlin variable BLTOUCH_DELAY to 200**
- **enable the Marlin variable BLTOUCH_FORCE_SW_MODE or that you delete the two forward slashes ("//") from the beginning of the line.**

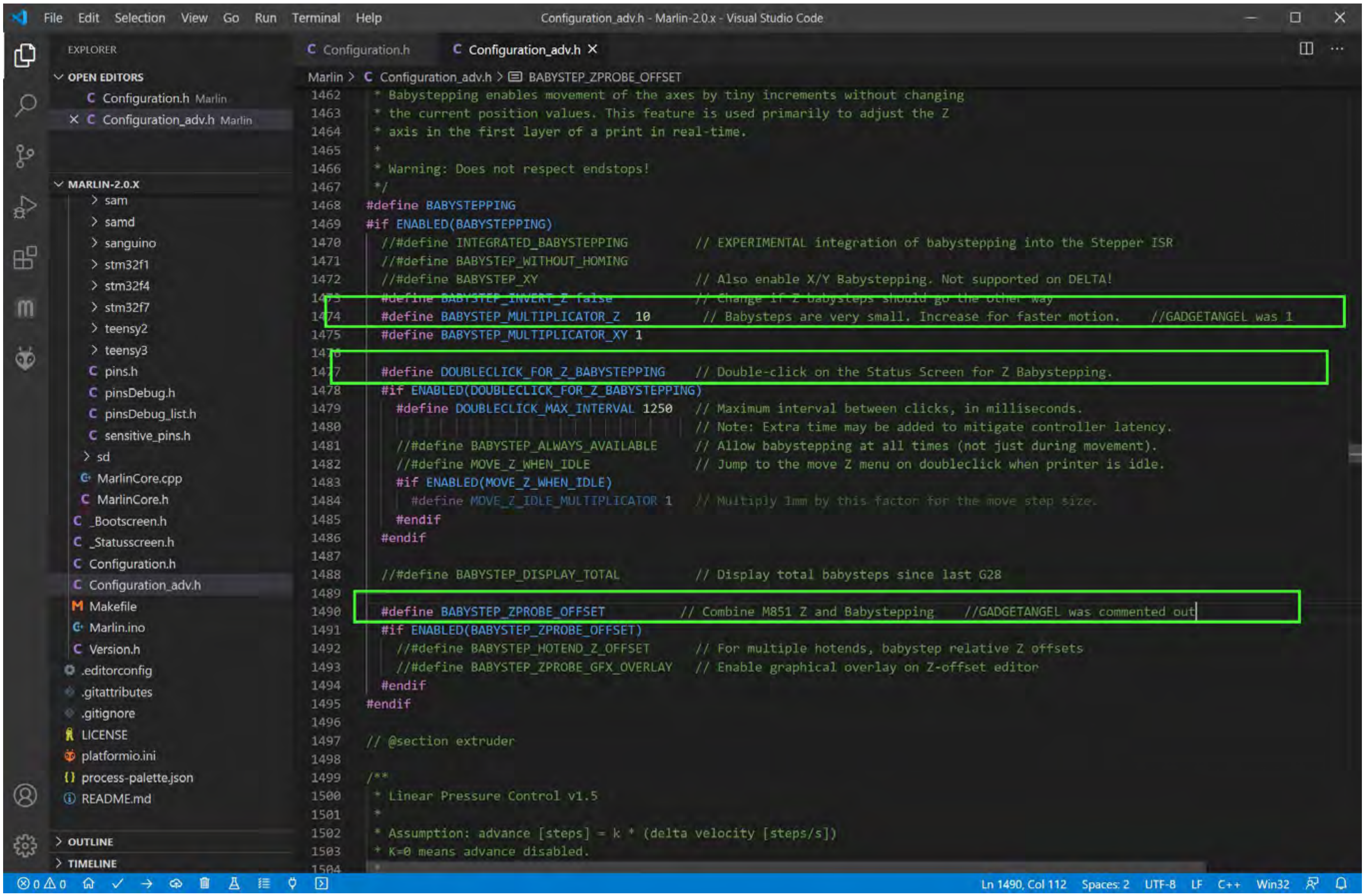


- Go to the next page.

APPENDIX J

Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **change the Marlin variable BABYSTEP_MULTIPLICATOR_Z to 10**
- **ensure the Marlin variable DOUBLECLICK_FOR_Z_BABYSTEPPING is enabled or that you delete the two forward slashes ("//") from the beginning of the line.**
- **enable the Marlin variable BABYSTEP_ZPROBE_OFFSET or that you delete the two forward slashes ("//") from the beginning of the line.**



- Go to the next page.

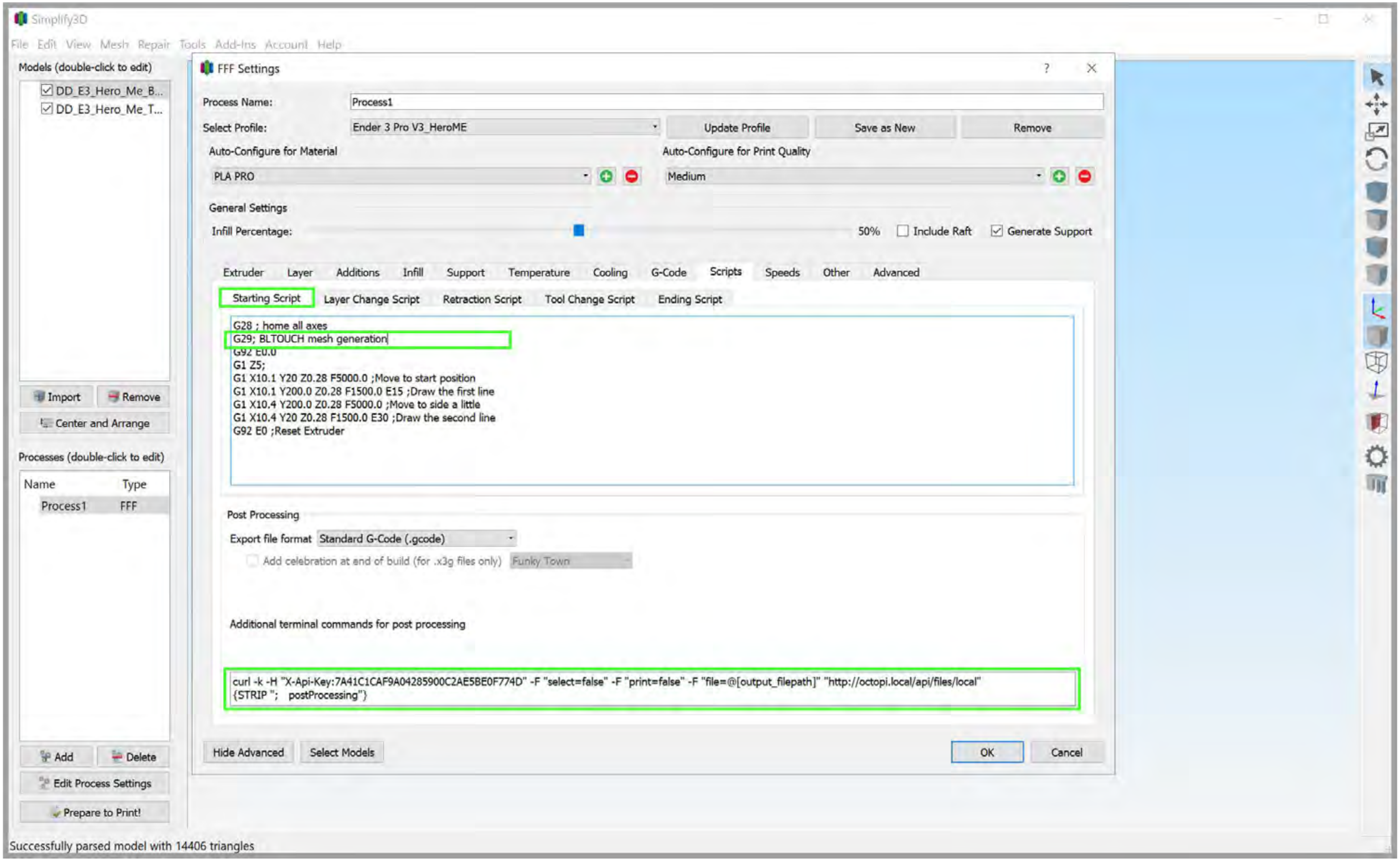
APPENDIX J
Testing the BLTouch Setup

- **Now that the BLTouch is wired to the SKR V1.3 board and the Marlin Firmware is setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.**
- **we need to set the BLTouch Z probe offset on the 3D Printer.**
- **Raise the 3D Printer's Z Axis up high enough so that you can get your hand under the BLTouch Probe.**
- **Try to HOME the 3D Printer from the Printer's LCD screen.**
- **As the Z Axis drops place your hand under the BLTouch so that the BLTouch Probe will hit your hand. If your 3D Printer's Z Axis raises up after the BLTouch Probe hits your hand than the BLTouch is working properly. If the Z Axis still drops, after the BLTouch hits your hand, then turn the power off your 3D printer to stop the Z Axis from hitting the bed. To correct the problem, double check your wiring of the BLTouch to the SKR V1.3 board.**
- **If the BLTouch Z Axis rises after hitting your hand, then your BLTouch is wired up to the SKR V1.3 board correctly.**
- **Connect to the Printer via Pronterface. Send a G28 command to HOME all axes. After the Printer finishes homing all axes. Send a G29 command to tell the 3D Printer to probe the bed for ABL (automatic bed leveling). Does you printer probe the bed? If so it's time to set the Z OFFSET.**
- **Before setting the Z OFFSET, lets change our SLICER settings to add the G29 command to the start G-Code script in your SLICER Settings. I use Simply3D Slicer.**
- **see the next page for my Simply3D Slicer setting for the start G-Code script.**

- Go to the next page.

APPENDIX J
Testing the BLTouch Setup

- **change the starting G-Code script for your slicer. You need to find the appropriate settings page in your slicer that is equivalent to the below picture, which shows the Starting G-Code script from Simplify3D.**
- **Add G29; to the starting G-Code script to your slicer. It is placed right after the G28; command.**
- **The line located in the box labeled "Additional terminal commands for post processing" is a command to allow me to send files to my 3D printer via Octoprint. This information is optional. This command will cause the generated .stl file from the slicer to be placed in a Octoprint watched folder on the Raspberry Pi.**



- Go to the next page.

APPENDIX J
Testing the BLTouch Setup

- **With your SLICER setup to do a G-29 command automatically at the start of a 3D print job, we need a 3D object that is big but only 1 layer thick. On Thingiverse there is an object called "Bedlevel test cross (For ABL, BLTouch, 3DTouch etc)" by mdieli. You can find it here at this URL: https://www.thingiverse.com/thing:3846283**
- **Import the object into your slicer. Use a layer thickness of 0.2mm. And send the slicer generated file to your 3D Printer to print the object. Your 3D Printer should perform an ABL (Automatic Bed Leveling) Bed Probe before it starts printing the object.**
- **While the object is printing, go to the MOTION menu on your 3D Printer's LCD display and change the Z OFFSET. Turn the knob to the LEFT will place the Nozzle closer to the printer bed. Turn the knob to the RIGHT will raise the Nozzle away from the printer bed.**
- **Adjust the Z OFFSET with the knob until you are satisfied with your first layer.**
- **Save your Z OFFSET value by going to your 3D Printer's CONTROL menu and selecting "Store Settings" to save the Z OFFSET to the EEPROM of the 3D Printer. Once saved to EEPROM, the Z OFFSET will be saved in between reboots of your 3D Printer.**

- Go to the next page.

- **keep the Marlin settings the same (as above) until the BLTouch is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the BLTouch!!**

APPENDIX K

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

To install a PT100 temperature sensor to your 3D printer requires four steps:

- **Decide which of the two techniques you will use to connect the PT100 amplifier board to the MCU board (SKR V1.3 board).**
- **Wire up the PT100 sensor to PT100 amplifier board.**
- **Wire up the PT100 amplifier board to the MCU board (SKR V1.3 board). If you chose to connect up the PT100 with Technique #1 then you MUST use Method #1 (Digital PWR) to power your amplifier board. If you chose to connect up the PT100 with Technique #2 than you have two methods to select from to power your amplifier board: Method #1 (Digital PWR) or Method #2 (Analog PWR).**
- **Marlin 2.0.x Firmware setup.**

There are two techniques you can use to wire up the PT100 amplifier board to the MCU:

- **Technique #1: Attach the PT100 amplifier board to the TFT Port by using TX0 PIN. With this technique you lose the ability to use the TFT. The advantage is that you do not have to perform hardware hacks on the SKR V1.3 board. ONLY use Method #1 (Digital PWR) to power your amplifier board.**
- **Technique #2[*1]: Attach the PT100 amplifier board to a thermistor port (TH0 or TH1). The disadvantage of this method is that you must perform hardware hacks to change the thermistor port (TH0) PIN to an analog input PIN. This requires removing a resistor and capacitor from the chosen thermistor port (TH0). The advantage of this method is that you CAN use the TFT because you will not be using TX0 PIN with this technique.**

***1: https://www.thingiverse.com/thing:3603432**

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Technique #2*1 (continued): If you chose to power the amplifier board by using Method #2 (Analog PWR) you get the advantage of noise immunity. Another advantage is that there are very few changes needed in Marlin 2.0.x Firmware. With Technique #2 you have to decide which method you will use to power your amplifier board. There are two methods to select from: Method #1 (Digital PWR), or Method #2 (Analog PWR). Method #2 is the preferred method with this technique due to noise immunity.

Technique #2 &Method #2 is the preferred and recommended option to connect your SKR V1.3 board up to a PT100 Sensor (with a PT100 amplifier board).

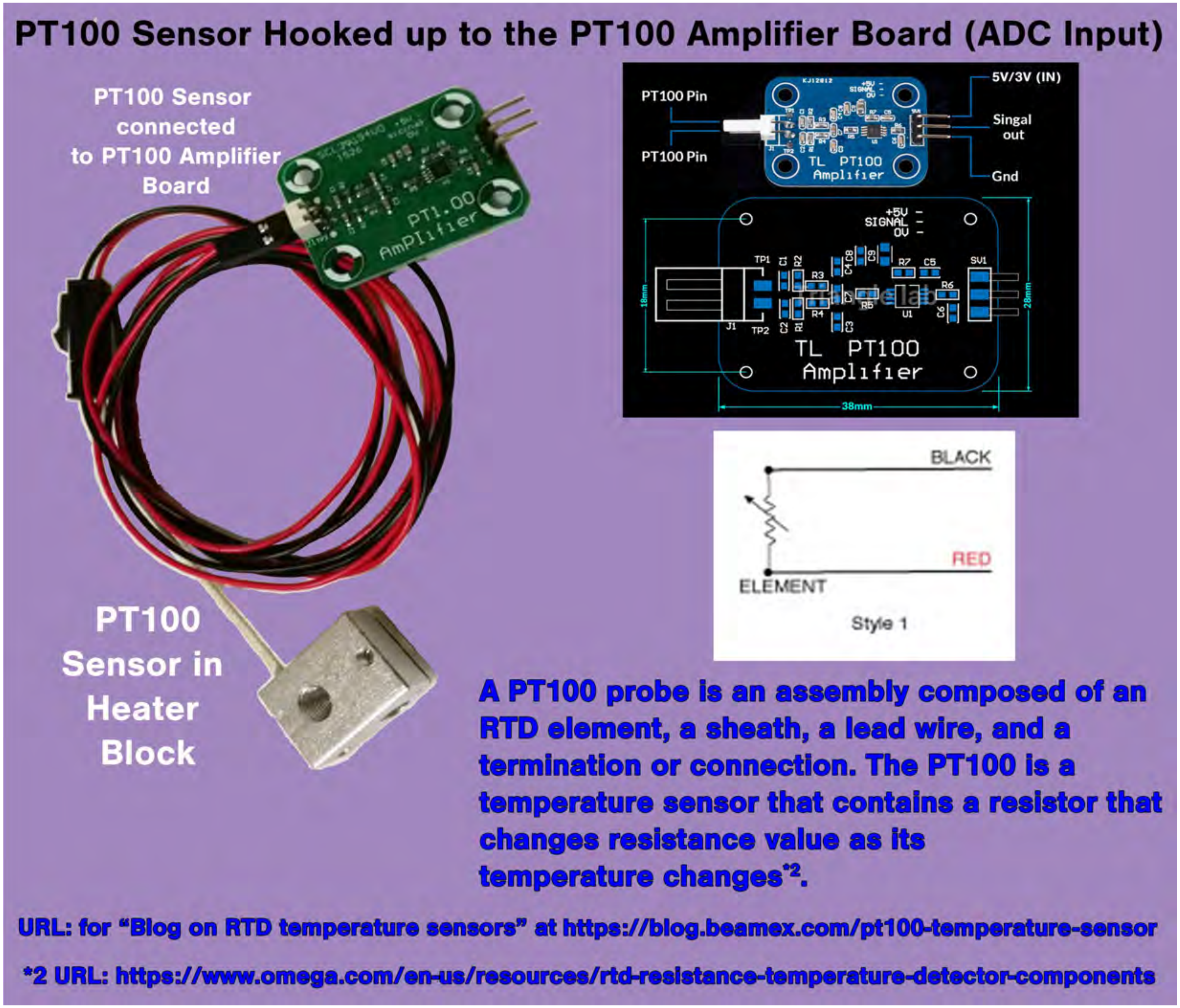
The diagram on the next page shows you the PT100 amplifier board attached to to a PT100 (RTD) temperature sensor.

It also indicates that the amplifier board will be using the ADC input on the MCU not SPI inputs. SPI inputs uses digital I/O lines while the ADC input uses an analog I/O line.

- Go to the next page.

APPENDIX K

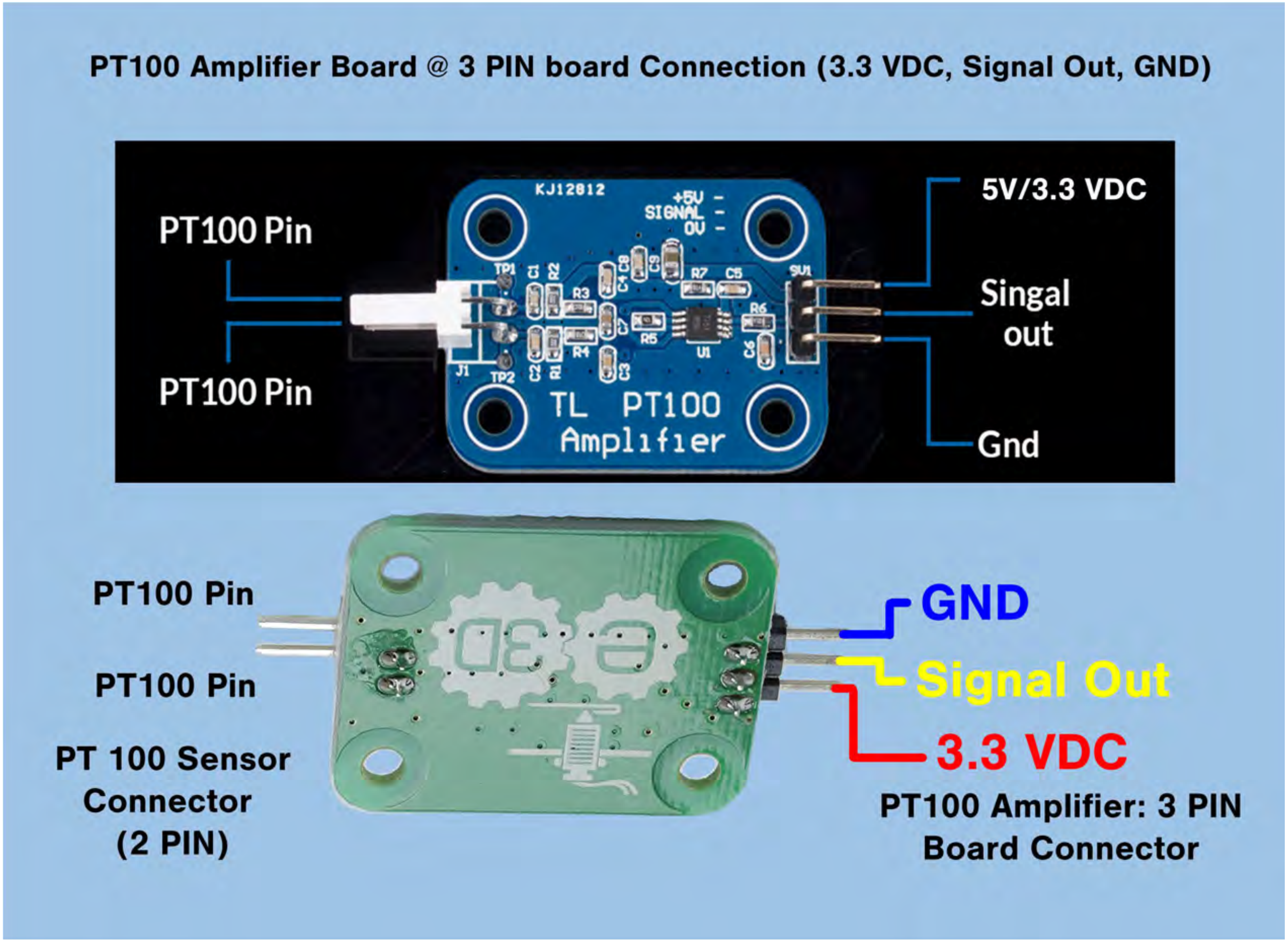
Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor



- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

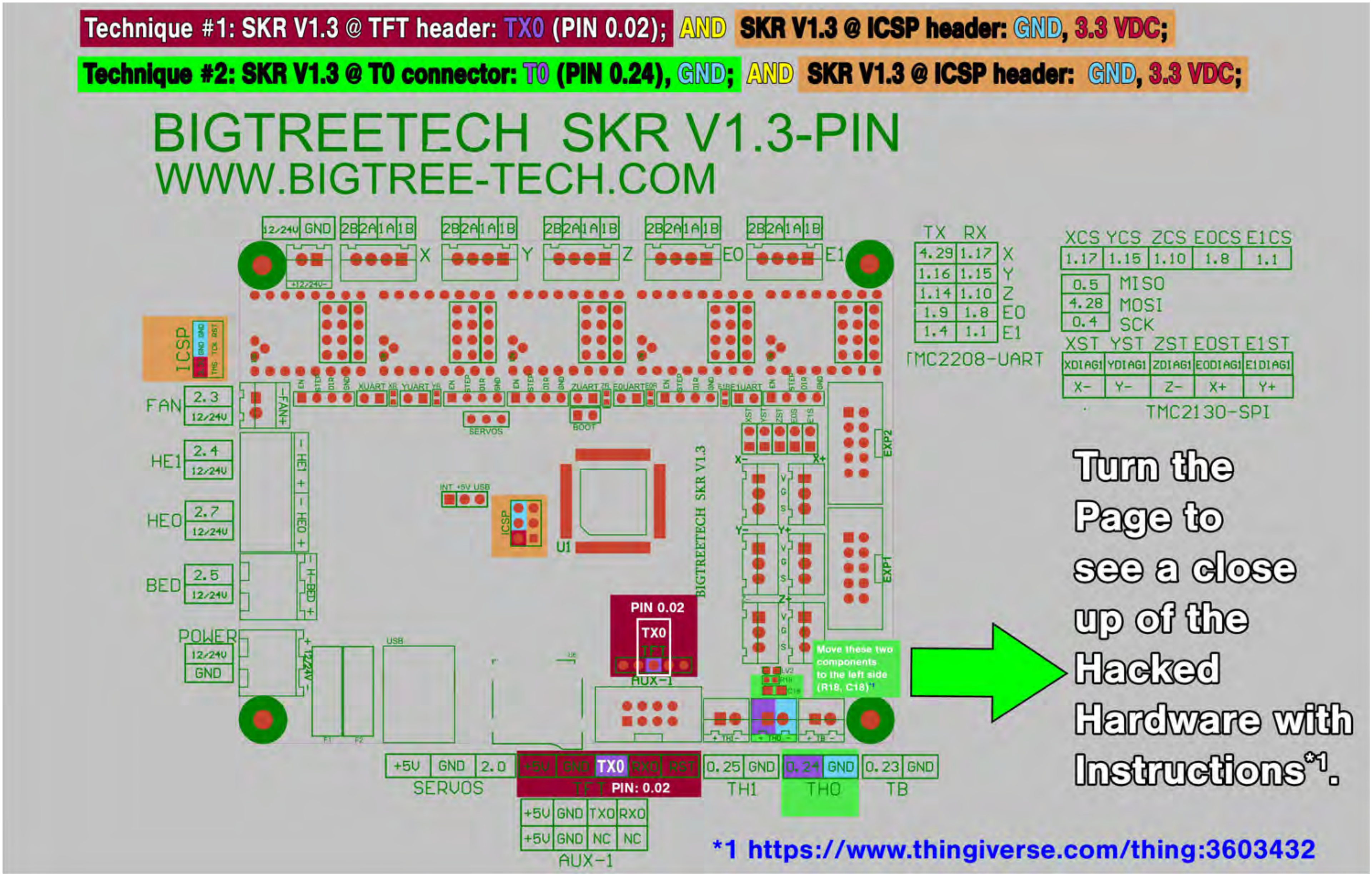
The diagram below shows the locations for +5V/3.3VDC, GND, and Signal Out (which is the ADC output signal that carries the information about the temperature). The diagram below also shows where the PT100 sensor pins are located. The board on top is a PT100 amplifier board made by Trianglelab while the board on the bottom of the picture is an amplifier board made by E3D. Each board's silk screen (labels) show the location for each OUTPUT PIN.



- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the PT100 temperature sensor with the PT100 amplifier board. This is an overview. The diagram indicates which PINS are involved in Technique #1 and Technique #2.

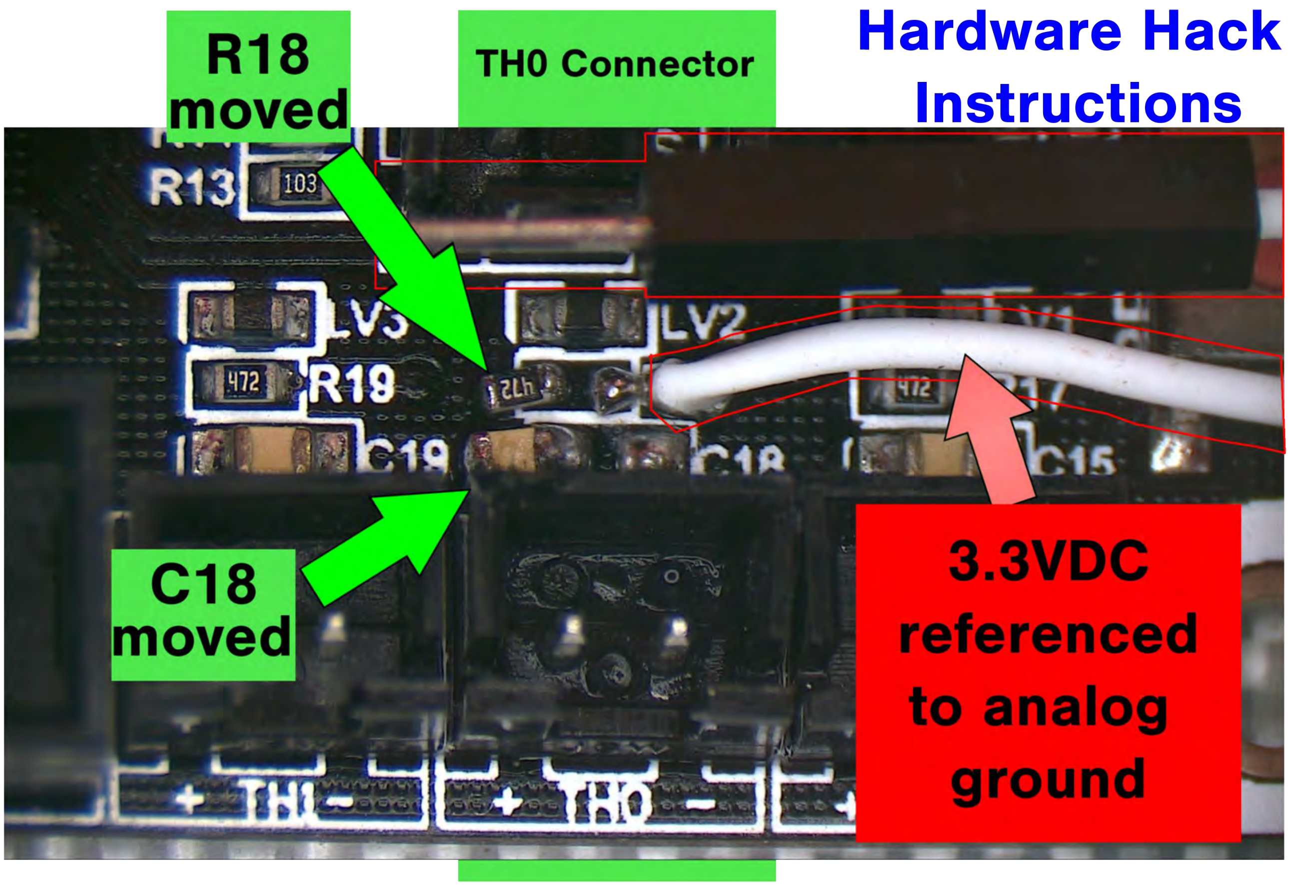


- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Instructions on how to perform the SKR V1.3 hardware hack to obtain an ADC input on a Thermistor port:

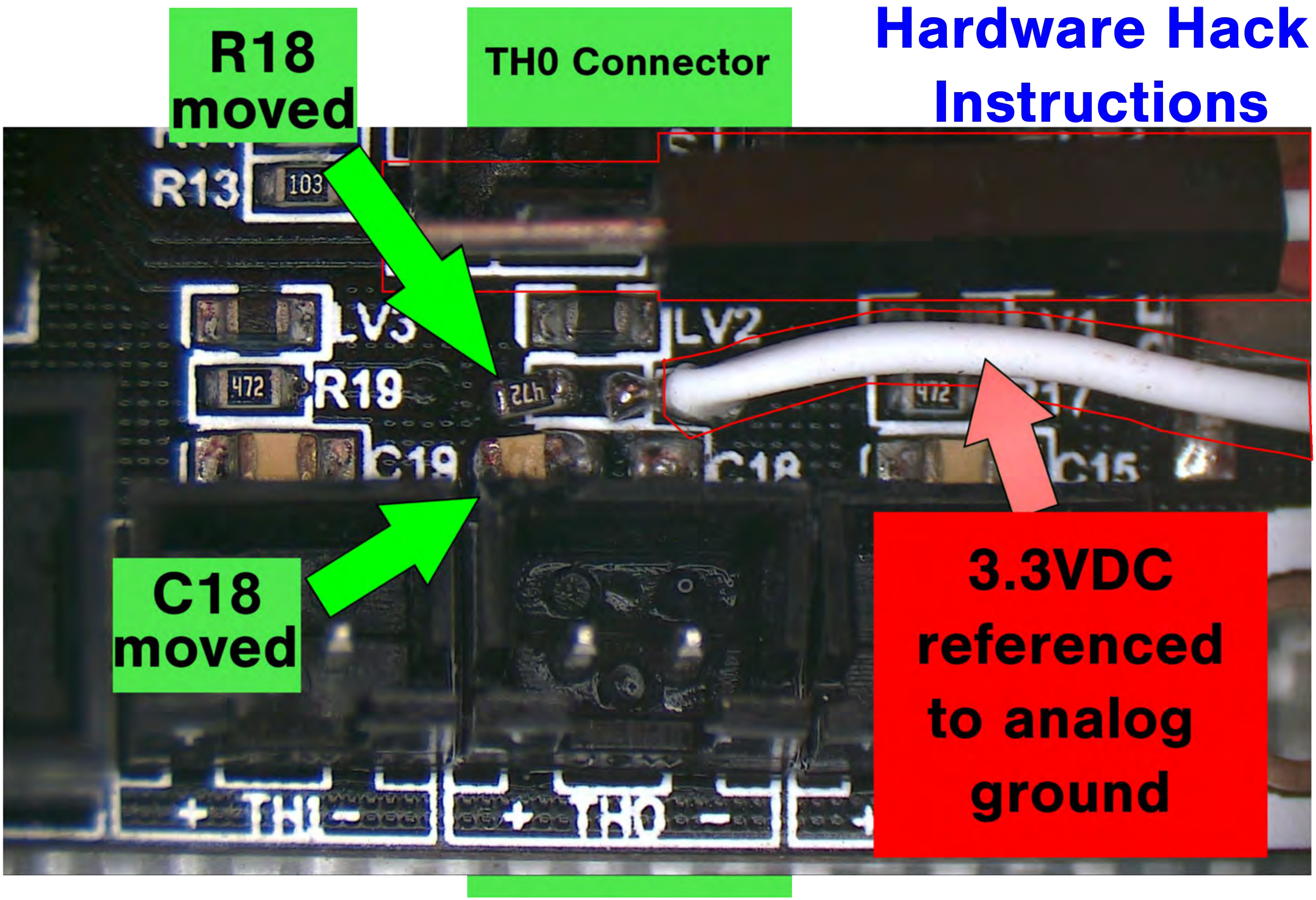
Remove R18 and C18 or shift to left-hand solder pad and leave right-hand solder pad empty. If you don't have a soldering iron you can cut them (R18, C18) with wire cutters, but that will not allow you to revert the hardware back if you want to use TH0 for a thermistor port*1. Notice the board is oriented so the TH0 label is not upside-down.



The next page shows the above diagram again (full page size) for your convenience.

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor



- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Now you have to decide on how you want to power the PT100 amplifier board. Where are you going to get the 3.3 VDC and ground for the PT100 amplifier board's +V and 0V PINS? There are two methods to choose from. Method #1 does not require you to perform another hardware hack, but the power supplied to the PT100 amplifier board will only be conditioned from your 3D Printer's PSU and not conditioned for analog ADC input. While Method #2 requires you to perform another small hardware hack, but the power will be conditioned and filtered by the electronics (for analog ADC input)found on the MCU board. Method #2 is the preferred method for noise immunity.

If you want to use Method #2 for powering the PT100 amplifier board do the following:

1. **Obtain a male DuPont Jumper cable. Cut off one end of the Jumper cable and expose the copper wire on that end. Solder the copper end of the male DuPont Jumper cable to the R18's solder pad located on the right-hand side (the board is rotated so that the TH0 labeling is NOT upside-down).**

Now its time to supply power to the PT100 amplifier board. The following pages will describe the two methods available to select from. Please chose ONE method.

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Instructions for Suppling Power to the PT100 Amplifier Board:

INTRODUCTION:

If the analog ground (GND_ANALOG on schematic) and digital ground (GND on schematic) are interconnected in any way, a ground loop is created which introduces noise into the ground circuit which causes errant temperature readings.

Therefore, there are two ways to hook up ground and the 3.3VDC for the amplifier board. Choose ONE method ONLY. Do NOT inter-mix them!

Method #1 (Digital PWR) uses digital ground and a 3.3VDC that is referenced to its digital ground. While Method #2 (Analog PWR) uses analog ground and a 3.3VDC that is referenced to its analog ground.

NOTICE: Do not pair an analog ground to a digital referenced 3.3VDC. Do not pair a digital ground to an analog referenced 3.3VDC!!

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Method #1 for Powering the PT100 Amplifier Board (Digital PWR):

If you remove or move R18 and C18 and DO NOT solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. **the ICSP header provides you access to the 3.3VDC that is reference to the digital ground (GND on the schematic). Therefore, use the ICSP's 3.3VDC PIN to power the amplifier board. Connect the 3.3VDC input of the PT100 amplifier board (+V) to ICSP header's 3.3VDC PIN.**
2. **the ground located on the ICSP header is one possible digital ground (GND on schematic) available on the MCU. Connect the ground from the ICSP header to the PT100 amplifier board's ground PIN (0V).**
3. **DO NOT use the ground from the TH0 connector (GND_ANALOG on the schematic – this is analog ground you want digital ground!) for this method.**
4. **if you use this method (Method #1) refer to:**
   1. **Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header.**
   2. **Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Method #2 for Powering PT100 Amplifier Board (Analog PWR):

If you remove or move R18 and C18 and solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

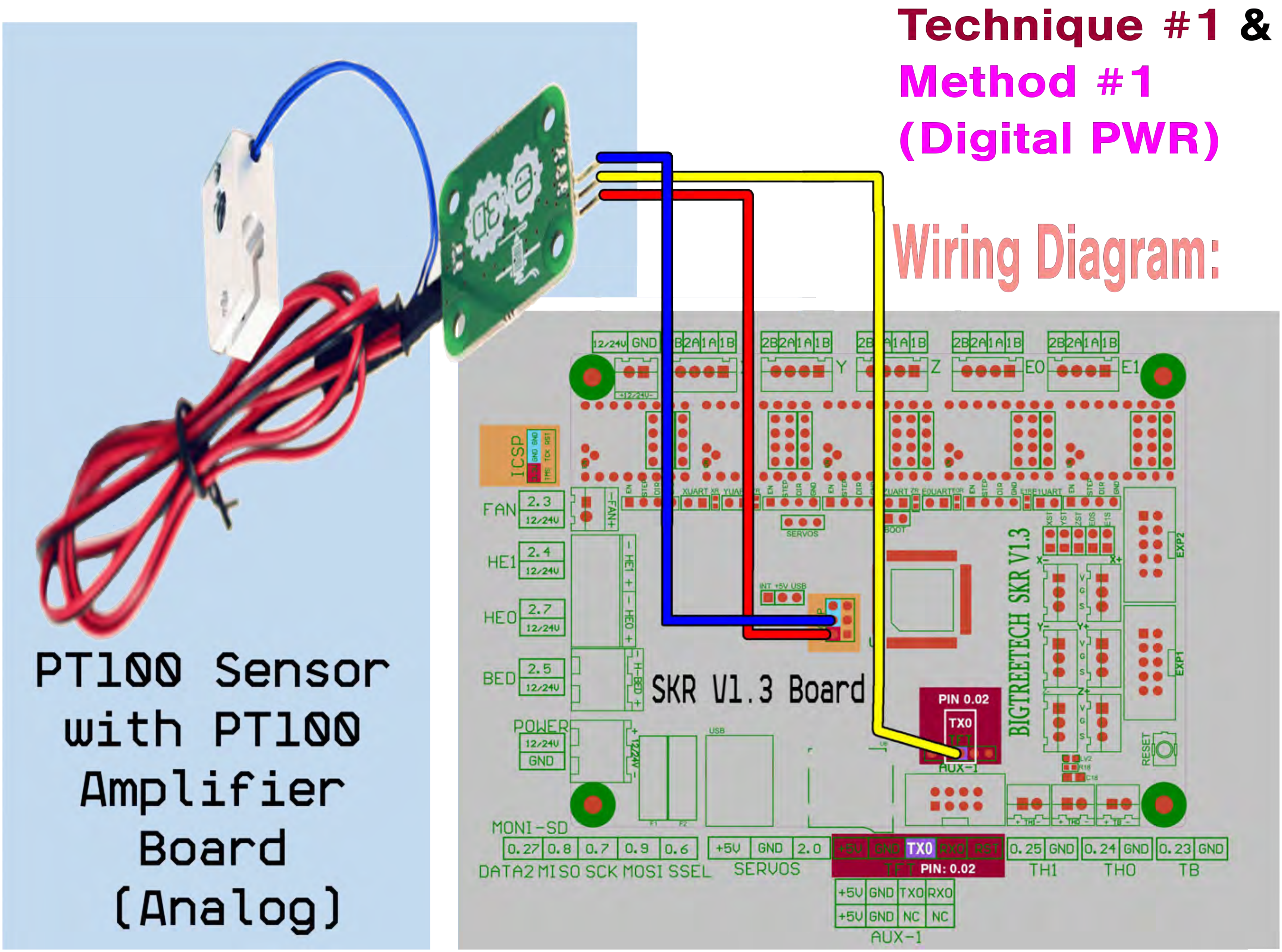
1. **the male DuPont Jumper cable now provides you access to the 3.3VDC that is reference to the analog ground (GND_ANALOG on schematic). Therefore, use the male DuPont Jumper cable to power the PT100 amplifier board. Place the male DuPont Jumper cable onto the 3.3VDC input of the PT100 amplifier board (+V).**
2. **the ground located on the TH0 connector is your analog ground (GND_ANALOG on schematic). Connect the TH0 connector's ground to the PT100 amplifier board's ground PIN (0V).**
3. **DO NOT use the ground from the ICSP header (GND on the schematic – this is digital ground you want analog ground!) for this method.**
4. **if you use this method (Method #2) refer to:**
    1. **Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

       This is the preferred and recommend way to connect up the PT100 amplifier board.

- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

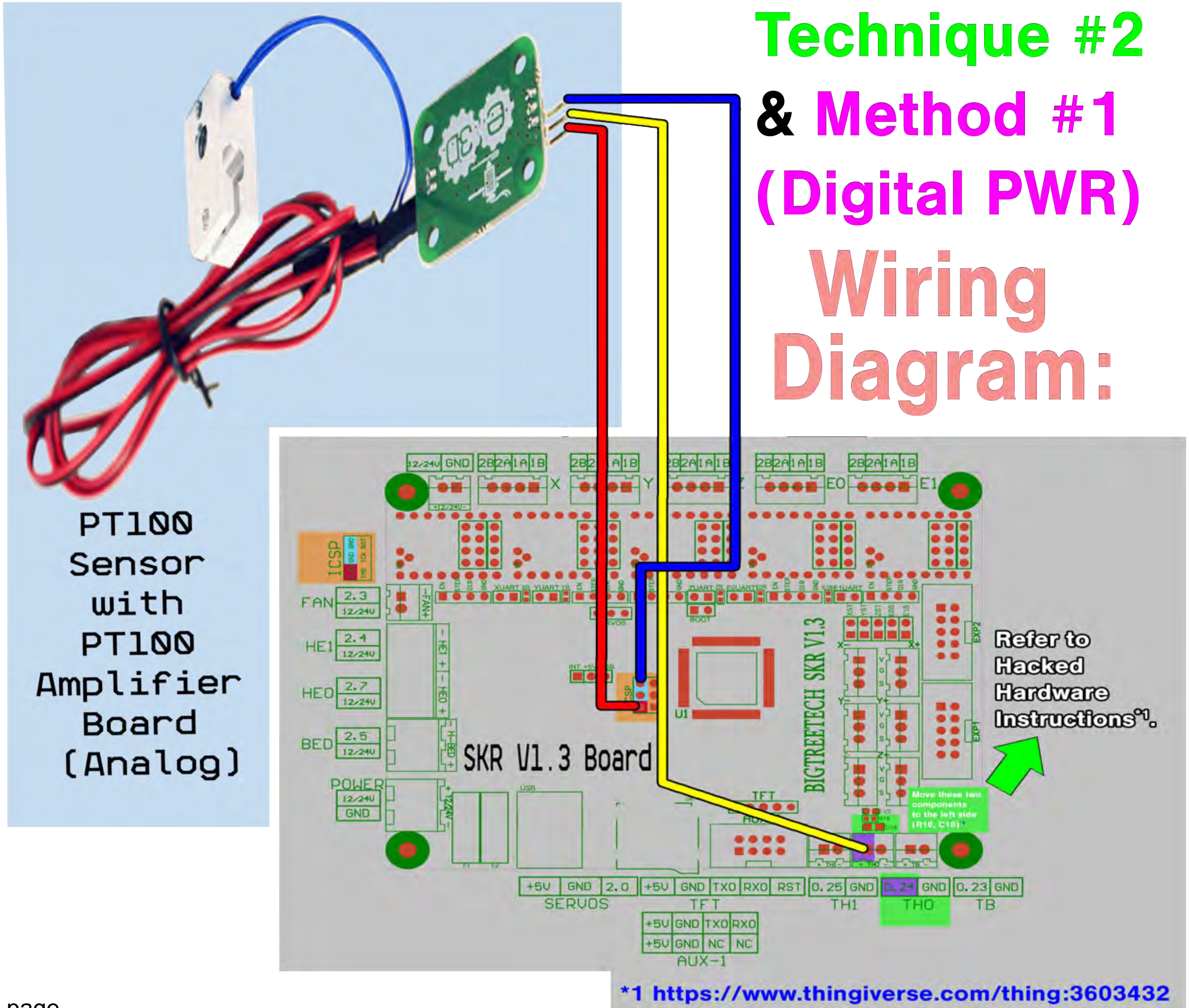
Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header:



- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector:



- Go to the next page.

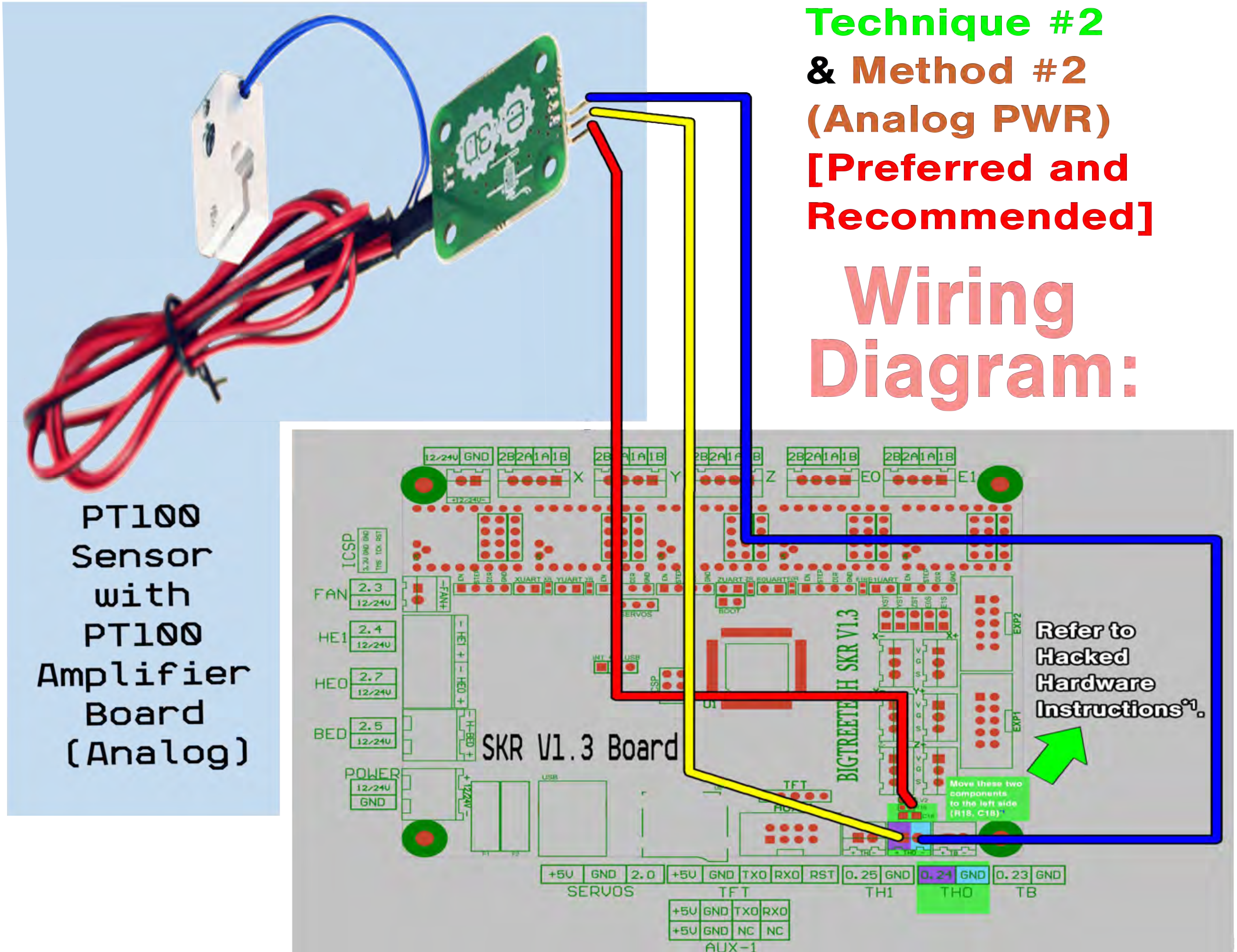
Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your PT100 to the TH0 connector:

This is the preferred and recommend way to connect up the amplifier board.

Technique #2 & Method #2 (Analog PWR) [Preferred and Recommended]

Wiring Diagram:



*1 https://www.thingiverse.com/thing:3603432

- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

If you are using Technique #2 goto the pages titled "Marlin 2.0x Firmware Setup via Technique #2".

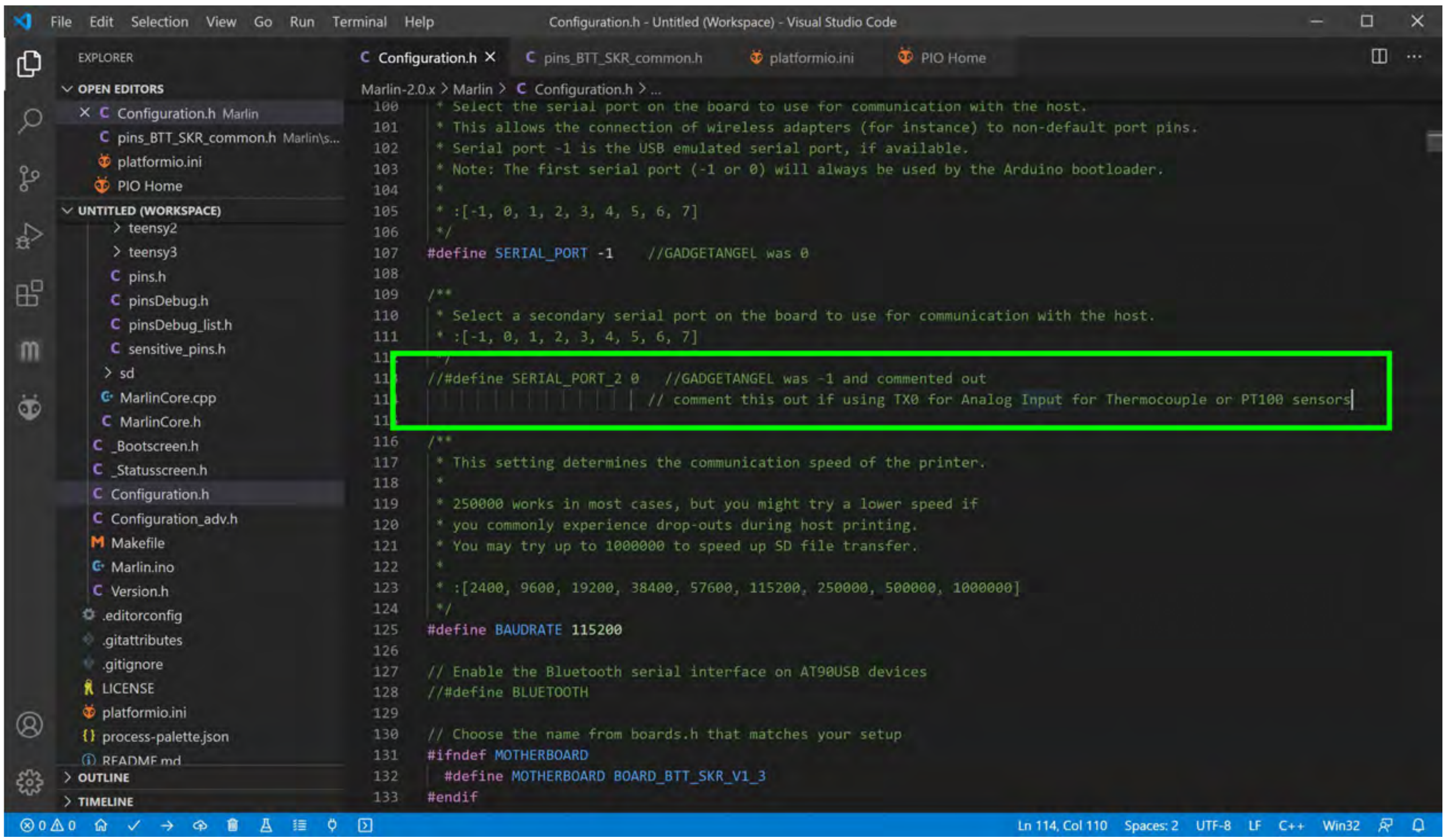
If you are using Technique #1 goto the next page.

- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #1

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**
- **disable the Marlin variable SERIRAL_PORT_2 or that you comment out the line by inserting two forward slashes ('//') at the beginning of the line.**

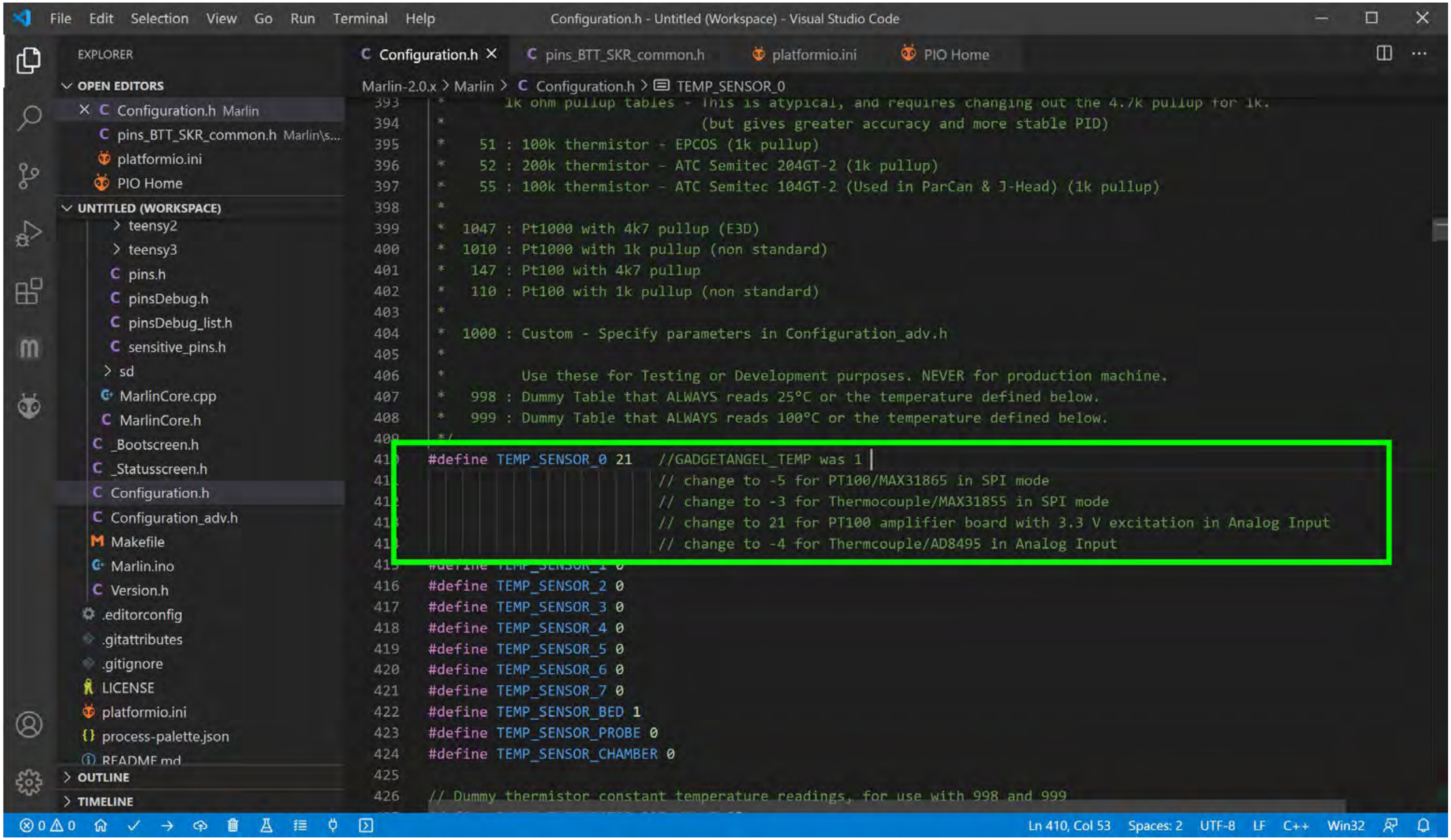


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #1

- **change the Marlin variable TEMP_SENSOR_0 to 21**

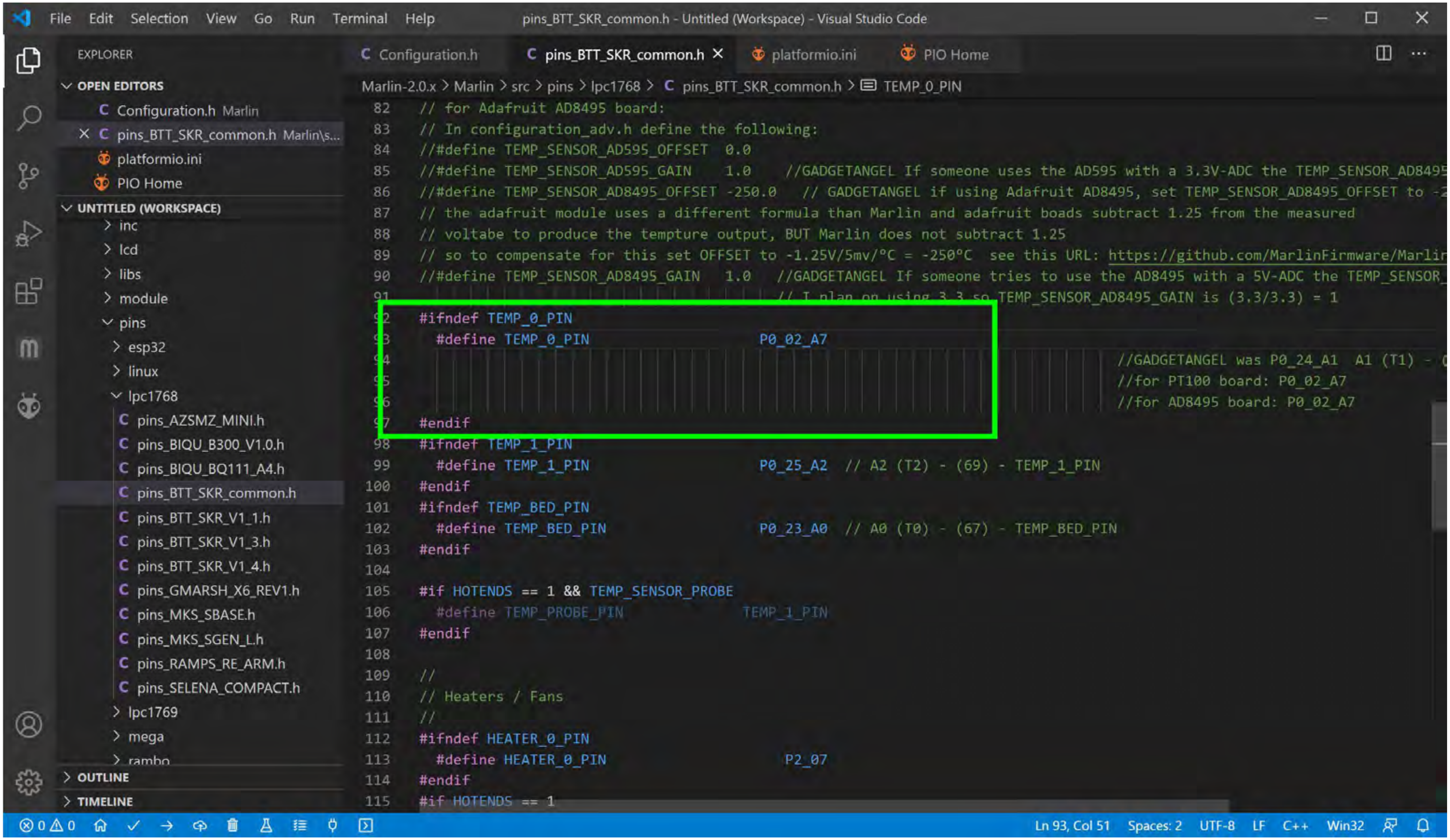


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #1

- **change the Marlin variable TEMP_0_PIN to P0_02_A7**

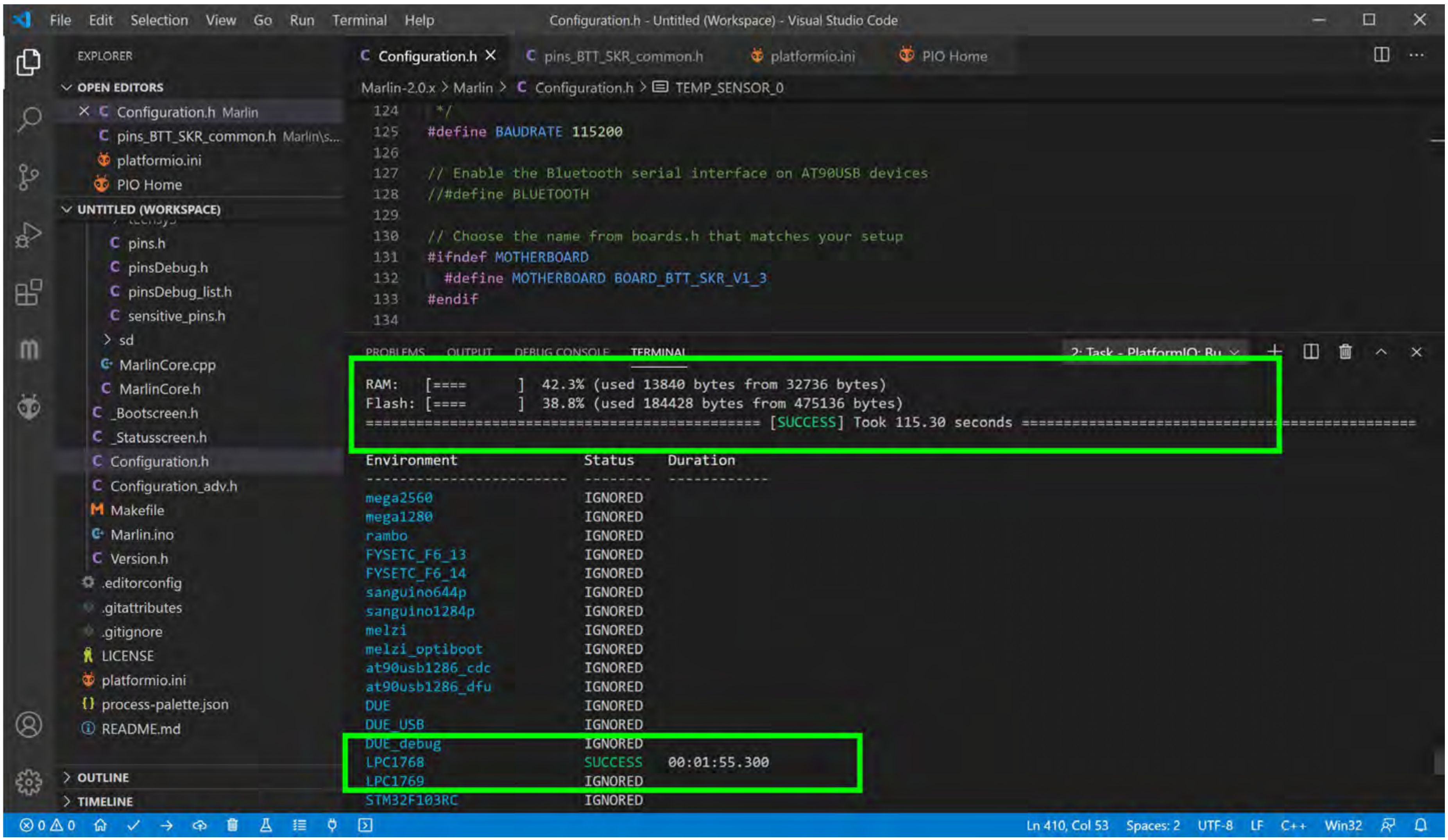


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #1

- **Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.**



- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

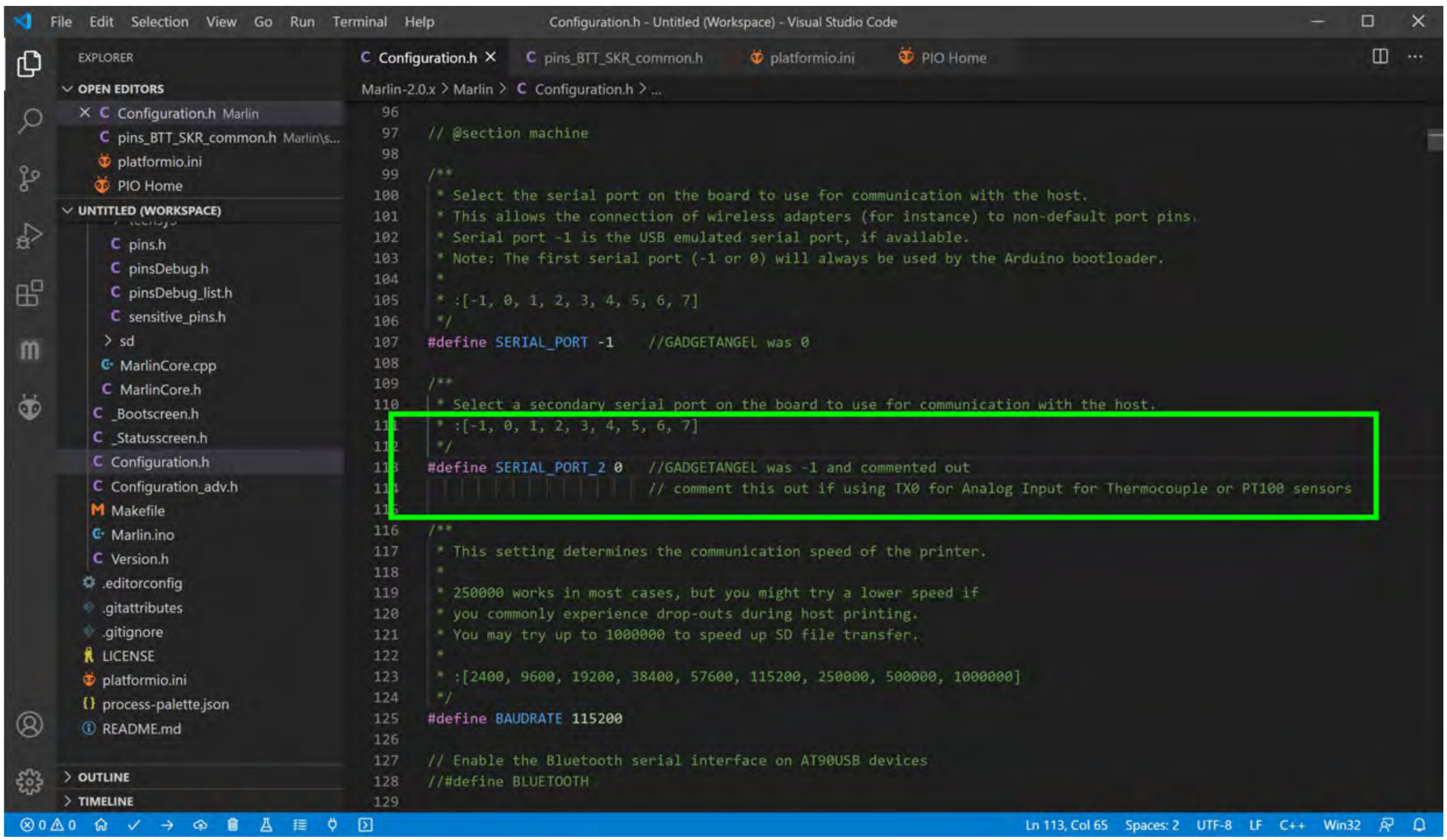
Marlin 2.0.x Firmware Setup via Technique #1

- **keep the Marlin settings the same (as above) until the PT100 senor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the PT100 amplifier board and PT100 sensor using Technique #1!!**

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #2

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**
- **if disabled pleanse enable the Marlin variable SERIRAL_PORT_2 or that you delete the two forward slashes ("//") from the beginning of the line.**

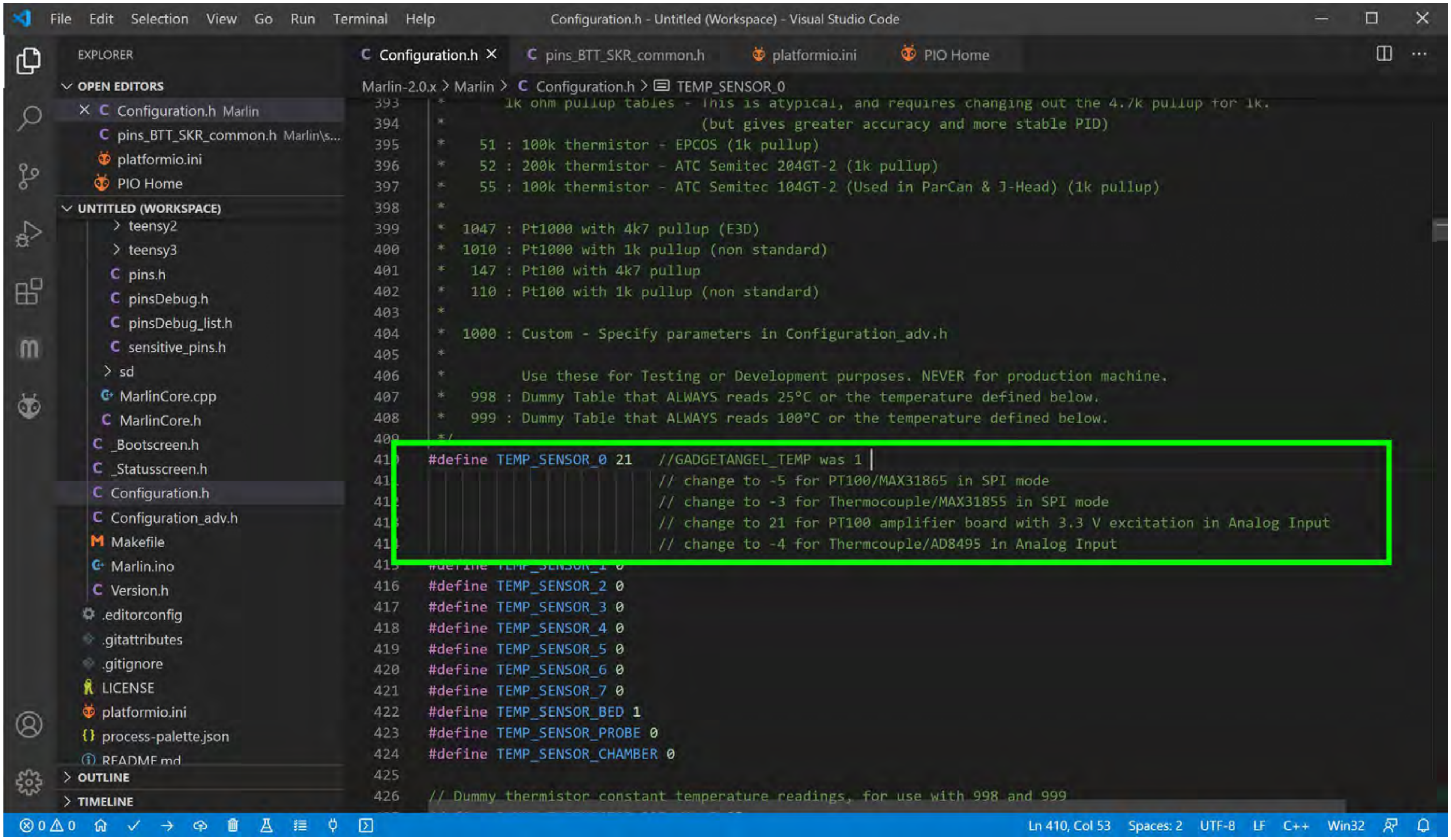


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #2

- **change the Marlin variable TEMP_SENSOR_0 to 21**

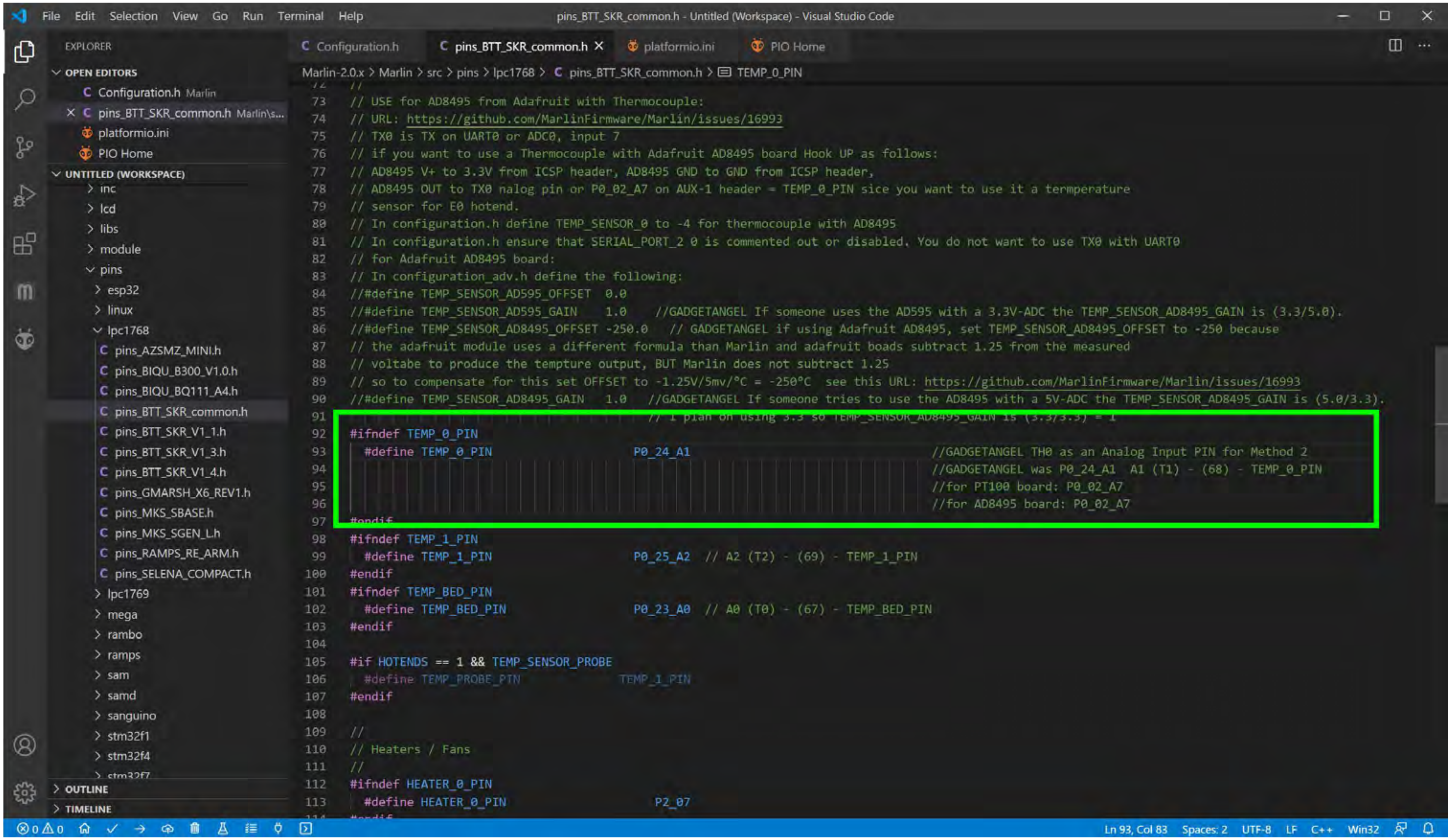


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #2

- **change the Marlin variable TEMP_0_PIN to P0_24_A1**

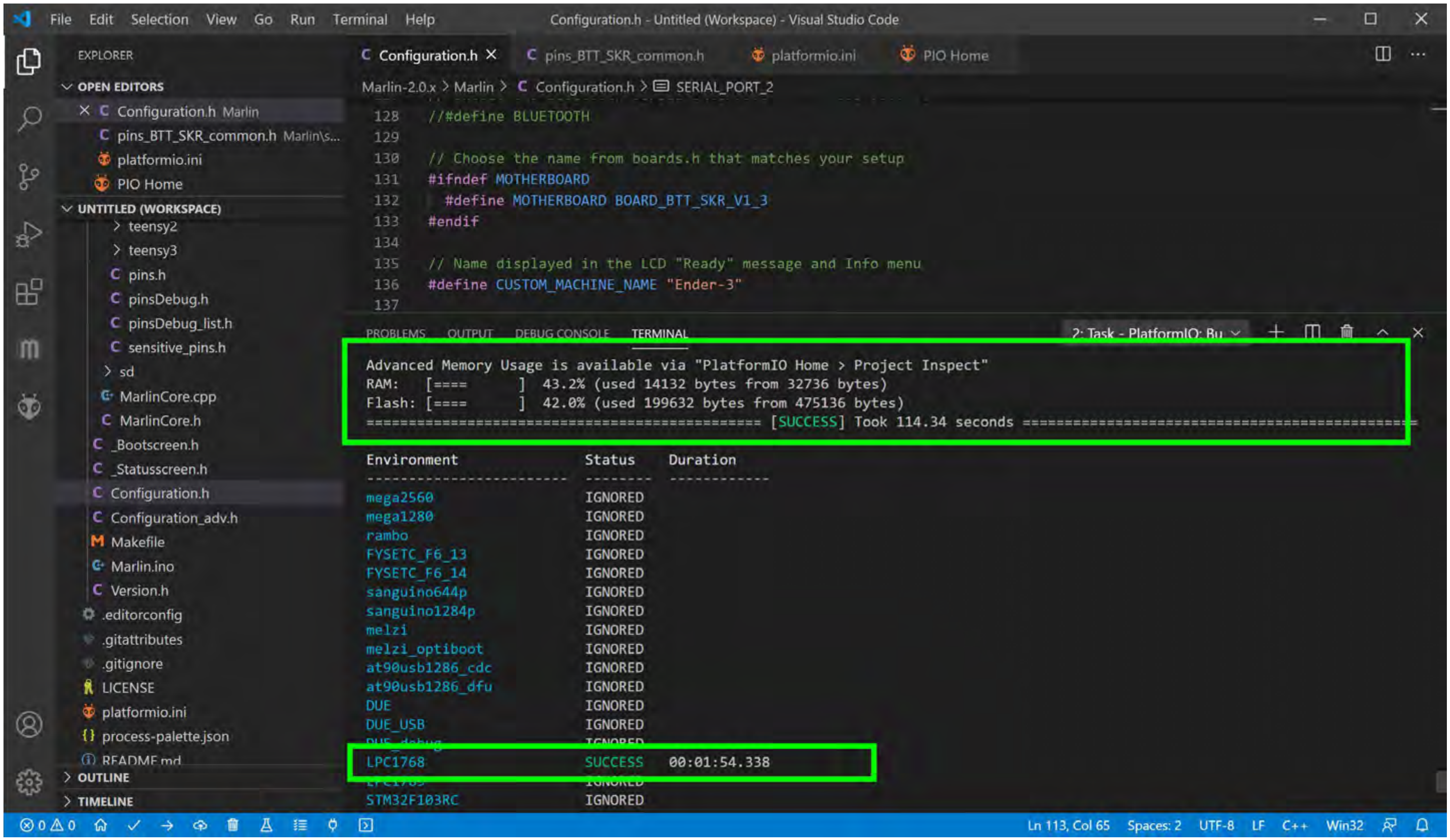


- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

Marlin 2.0.x Firmware Setup via Technique #2

- **Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.**



- Go to the next page.

Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Marlin 2.0.x Firmware Setup via Technique #2

- **keep the Marlin settings the same (as above) until the PT100 senor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the PT100 amplifier board and PT100 sensor using Technique #2!!**

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

To install a K-Type Thermocouple temperature sensor to your 3D printer requires four steps:

- **Decide which of the two techniques you will use to connect the K-Type thermocouple amplifier board (AD8495) to the MCU board (SKR V1.3 board).**
- **Wire up the K-Type thermocouple Sensor to the Adafruit's AD8495 Amplifier board.**
- **Wire up the Adafruit's AD8495 amplifier board to the MCU board (SKR V1.3 board). If you chose to connect up the K-Type thermocouple with Technique #1 then you MUST use Method #1 (Digital PWR) to power your amplifier board. If you chose to connect up the K-Type thermocouple with Technique #2 than you have two methods to select from to power your amplifier board: Method #1 (Digital PWR) or Method #2 (Analog PWR).**
- **Marlin 2.0.x Firmware setup.**

There are two techniques you can use to wire up the Adafruit's AD8495 Amplifier board to the MCU:

- **Technique #1: Attach the AD8495 amplifier board to the TFT Port by using TX0 PIN. With this technique you lose the ability to use the TFT. The advantage is that you do not have to perform hardware hacks on the SKR V1.3 board. ONLY use Method #1 (Digital PWR) to power your amplifier board.**
- **Technique #2[*1]: Attach Adafruit's AD8495 amplifier board to a thermistor port (TH0 or TH1). The disadvantage of this method is that you must perform hardware hacks to change the thermistor port (TH0) PIN to an analog input PIN. This requires removing a resistor and capacitor from the chosen thermistor port (TH0). The advantage of this method is that you CAN use the TFT because you will not be using TX0 PIN with this technique.**

***1: https://www.thingiverse.com/thing:3603432**

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

If you chose to power the amplifier board by using Method #2 (Analog PWR) you get the advantage of noise immunity. Another advantage is that there are very few changes needed in Marlin 2.0.x Firmware. With Technique #2 you have to decide which method you will use to power your amplifier board. There are two methods to select from: Method #1 (Digital PWR), or Method #2 (Analog PWR). Method #2 is the preferred method with this technique due to noise immunity.

Technique #2 & Method #2 is the preferred and recommended option to connect your SKR V1.3 board up to a K-Type Thermocouple Sensor (with AD8495 amplifier board).

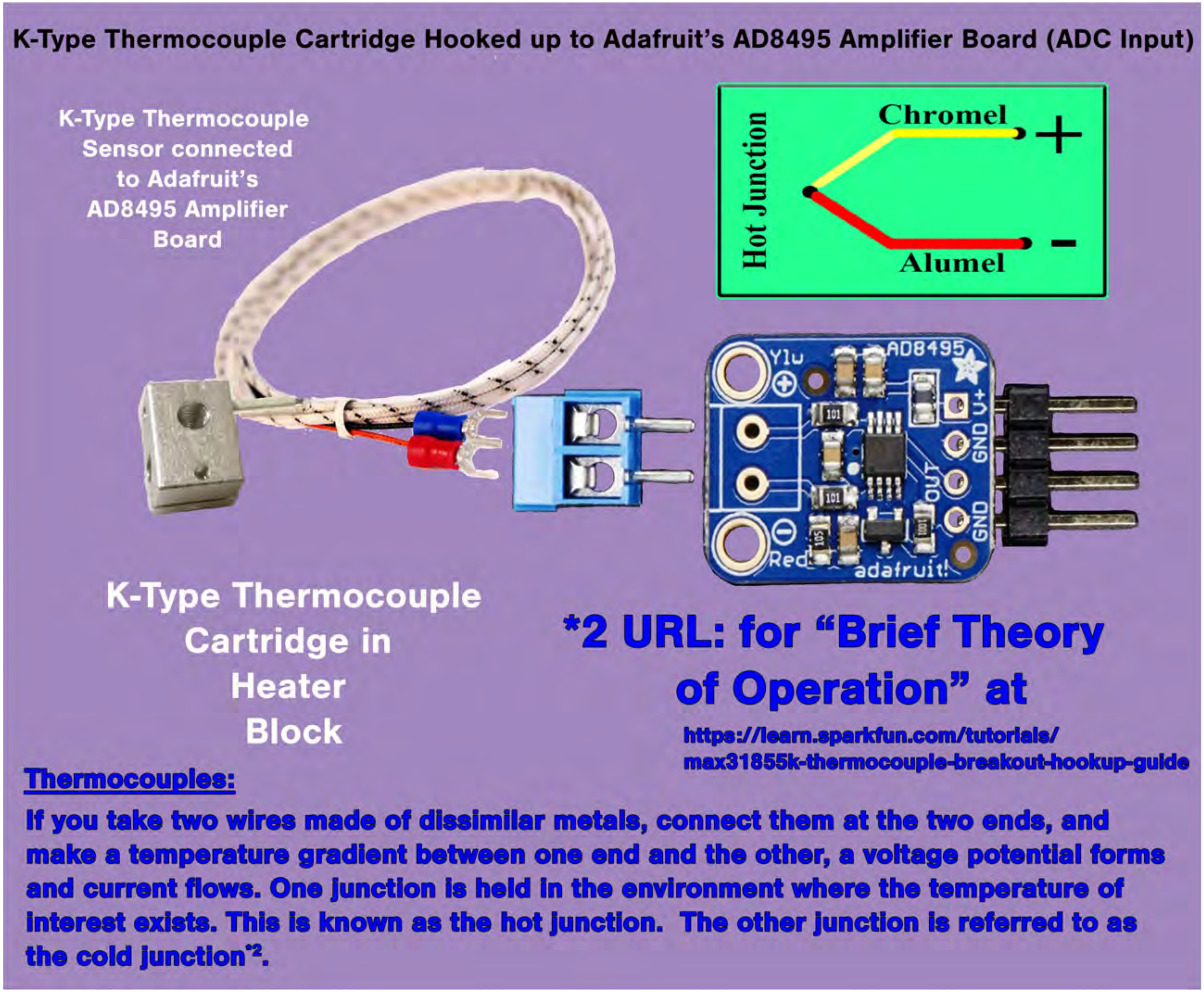
The diagram on the next page shows you the AD8495 amplifier board attached to to a K-Type Thermocouple temperature sensor.

It also indicates that the amplifier board will be using the ADC input on the MCU not SPI inputs. SPI inputs uses digital I/O lines while the ADC input uses an analog I/O line.

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

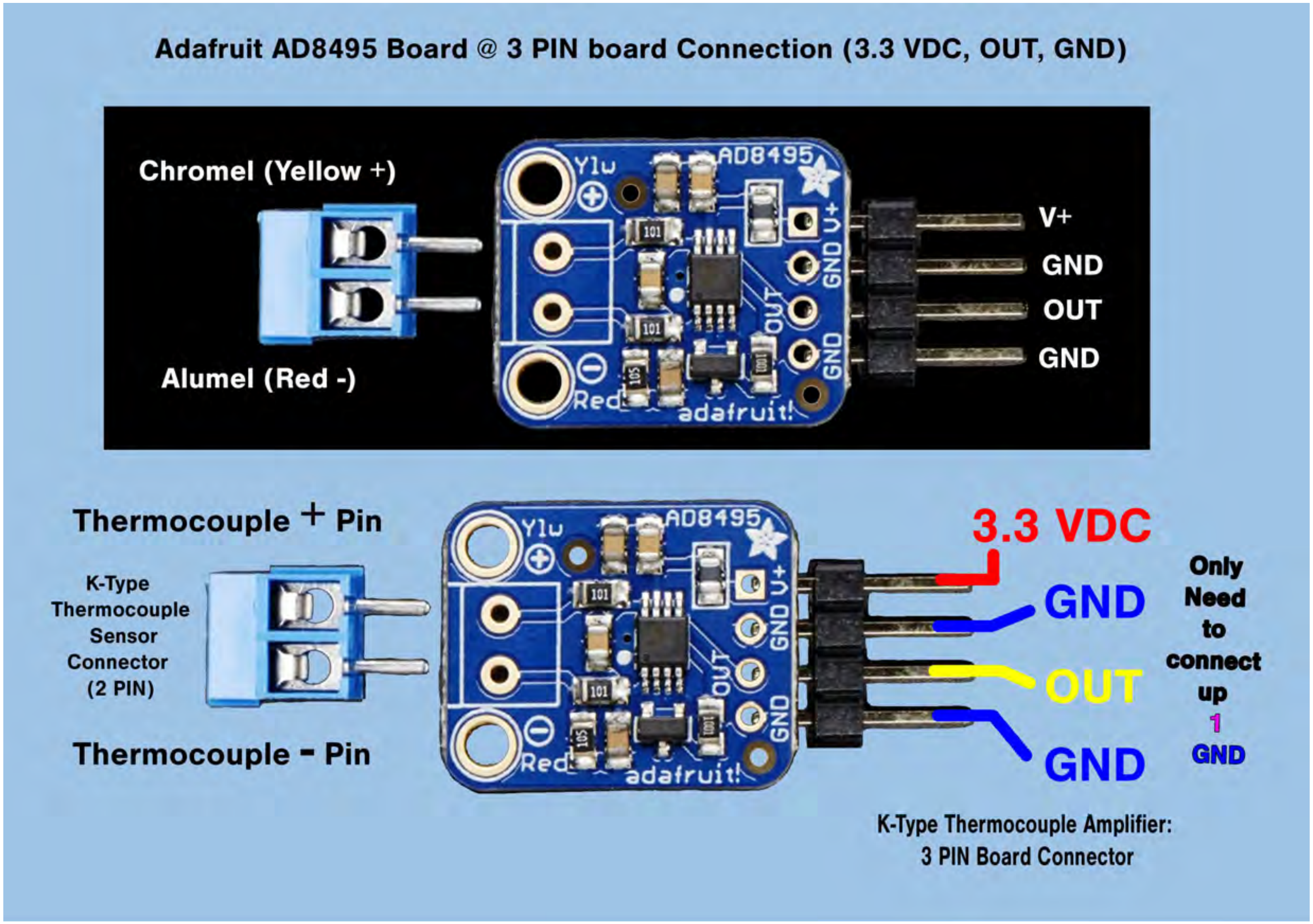


- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

The diagram below shows the locations for +V, GND, and OUT (which is the ADC output signal that carries the information about the temperature). The diagram below also shows where the K-Type Thermocouple sensor pins are located. The Chromel side of the thermocouple is usually indicated with a YELLOW(+) or BLUE(+) end. The Alumenl side of the thermocouple is usually indicated with a RED(-) end. The board on top and bottom is the AD8495 amplifier board made by Adafruit. The board's silk screen (labels) show the location for each OUTPUT PIN.

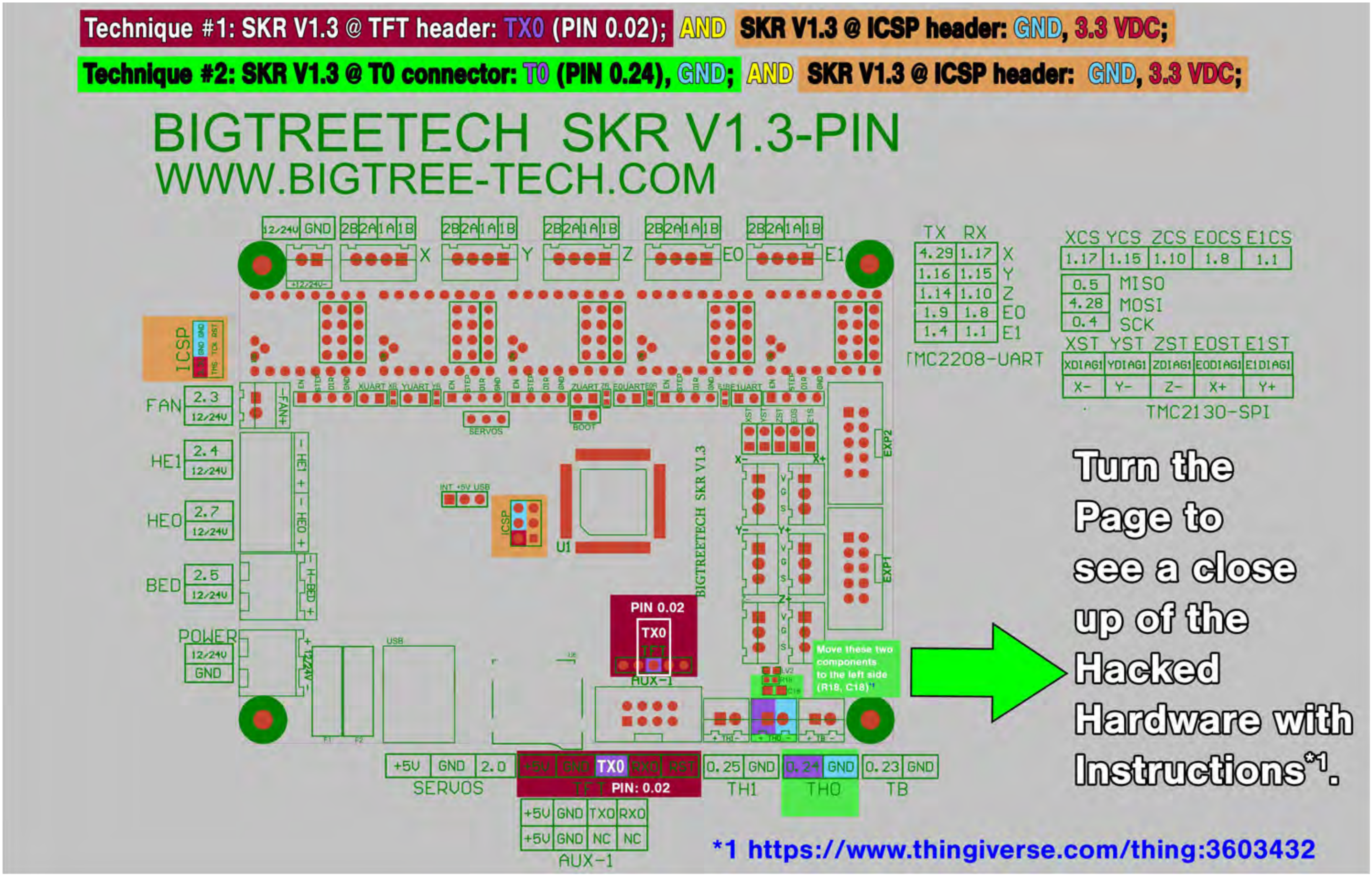


- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the K-Type Thermocouple temperature sensor with the AD8495 amplifier board. This is an overview. The diagram indicates which PINS are involved in Technique #1 and Technique #2.



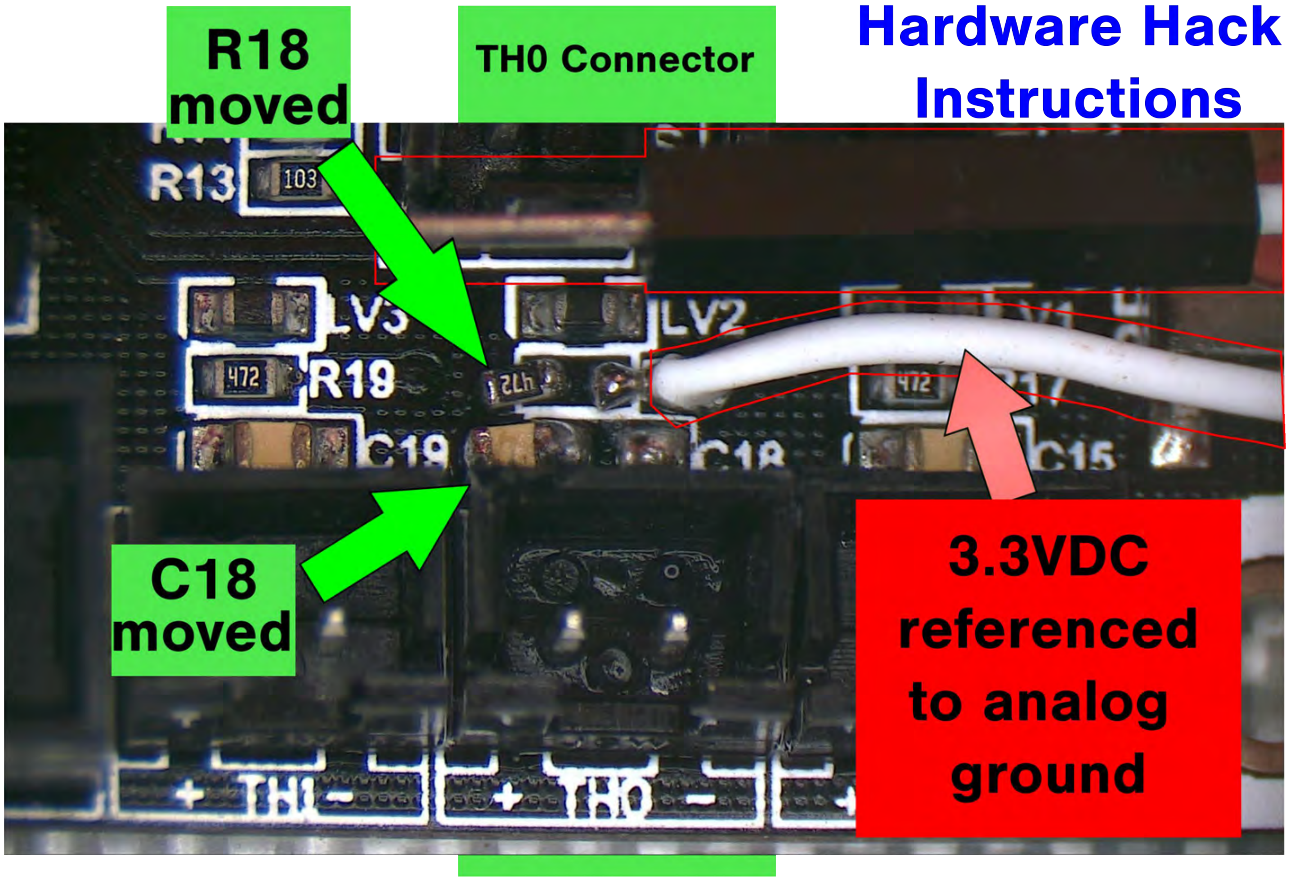
- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Instructions on how to perform the SKR V1.3 hardware hack to obtain an ADC input on a Thermistor port:

Remove R18 and C18 or shift to left-hand solder pad and leave right-hand solder pad empty. If you don't have a soldering iron you can cut them (R18, C18) with wire cutters, but that will not allow you to revert the hardware back if you want to use TH0 for a thermistor port*1. Notice the board is oriented so the TH0 label is not upside-down.

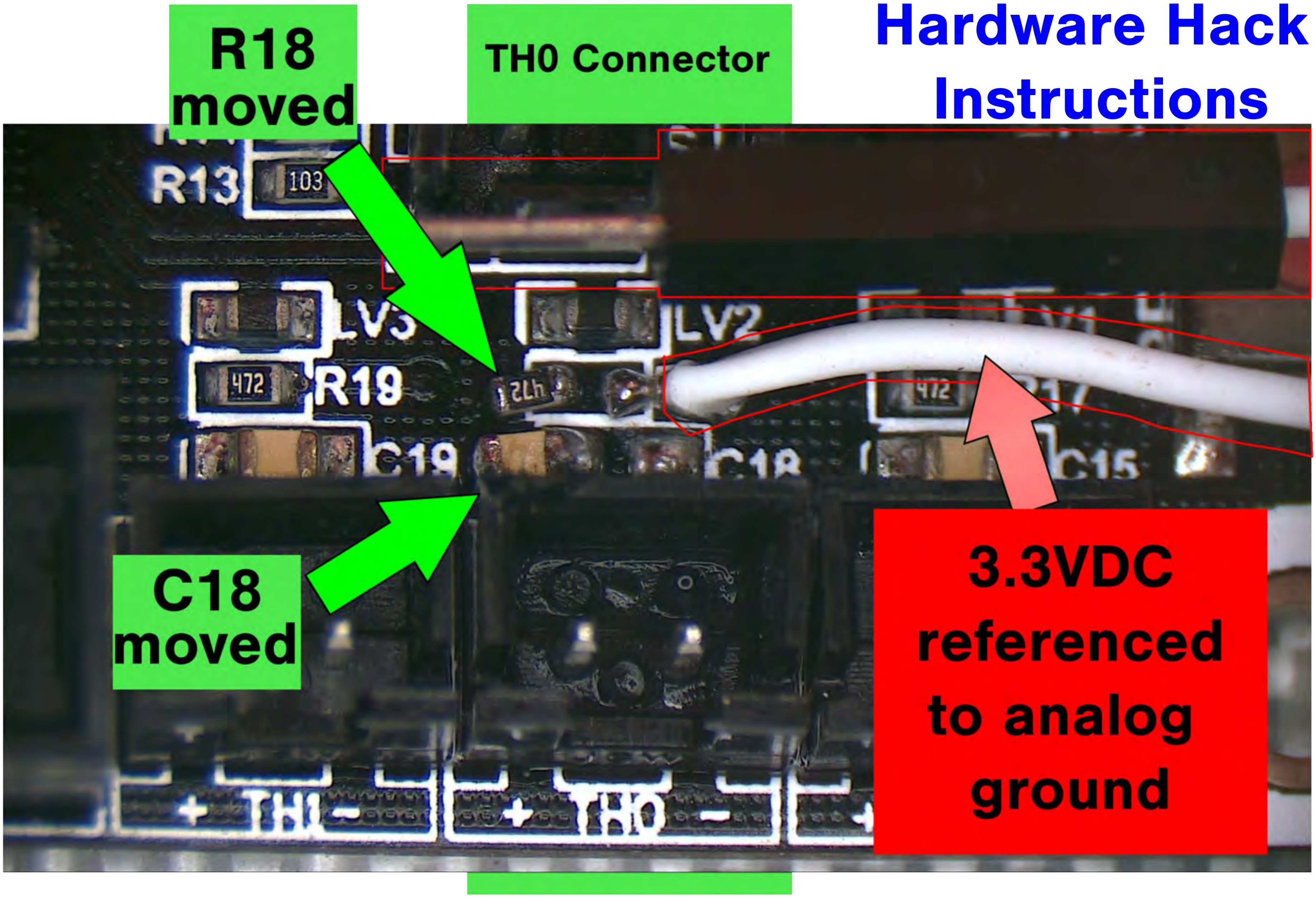


The next page shows the above diagram again (full page size) for your convenience.

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board



- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Now you have to decide on how you want to power the AD8495 amplifier board. Where are you going to get the 3.3 VDC and ground for the AD8495 amplifier board's +V and 0V PINS? There are two methods to choose from. Method #1 does not require you to perform another hardware hack, but the power supplied to the AD8495 amplifier board will only be conditioned from your 3D Printer's PSU and not conditioned for analog ADC input. While Method #2 requires you to perform another small hardware hack, but the power will be conditioned and filtered by the electronics (for analog ADC input) found on the MCU board. Method #2 is the preferred method for noise immunity.

If you want to use Method #2 for powering the AD8495 amplifier board do the following:

1. **Obtain a male DuPont Jumper cable. Cut off one end of the Jumper cable and expose the copper wire on that end. Solder the copper end of the male DuPont Jumper cable to the R18's solder pad located on the right-hand side (the board is rotated so that the TH0 labeling is NOT upside-down).**

Now its time to supply power to the AD8495 amplifier board. The following pages will describe the two methods available to select from. Please chose ONE method.

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Instructions for Suppling Power to the AD8495 Amplifier Board:

INTRODUCTION:

If the analog ground (GND_ANALOG on schematic) and digital ground (GND on schematic) are interconnected in any way, a ground loop is created which introduces noise into the ground circuit which causes errant temperature readings.

Therefore, there are two ways to hook up ground and the 3.3VDC for the amplifier board. Choose ONE method ONLY. Do NOT inter-mix them!

Method #1 (Digital PWR) uses digital ground and a 3.3VDC that is referenced to its digital ground. While Method #2 (Analog PWR) uses analog ground and a 3.3VDC that is referenced to its analog ground.

NOTICE: Do not pair an analog ground to a digital referenced 3.3VDC. Do not pair a digital ground to an analog referenced 3.3VDC!!

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Method #1 for Powering the AD8495 Amplifier Board (Digital PWR):

If you remove or move R18 and C18 and DO NOT solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. **the ICSP header provides you access to the 3.3VDC that is reference to the digital ground (GND on the schematic). Therefore, use the ICSP's 3.3VDC PIN to power the amplifier board. Connect the 3.3VDC input of the AD8495 amplifier board (+V) to ICSP header's 3.3VDC PIN.**
2. **the ground located on the ICSP header is one possible digital ground (GND on schematic) available on the MCU. Connect the ground from the ICSP header to the AD8495 amplifier board's ground PIN (0V).**
3. **DO NOT use the ground from the TH0 connector (GND_ANALOG on the schematic – this is analog ground you want digital ground!) for this method.**
4. **if you use this method (Method #1) refer to:**
    1. **Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header.**
    2. **Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Method #2 for Powering AD8495 Amplifier Board (Analog PWR):

If you remove or move R18 and C18 and solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. **the male DuPont Jumper cable now provides you access to the 3.3VDC that is reference to the analog ground (GND_ANALOG on schematic). Therefore, use the male DuPont Jumper cable to power the AD8495 amplifier board. Place the male DuPont Jumper cable onto the 3.3VDC input of the AD8495 amplifier board (+V).**
2. **the ground located on the TH0 connector is your analog ground (GND_ANALOG on schematic). Connect the TH0 connector's ground to the AD8495 amplifier board's ground PIN (0V).**
3. **DO NOT use the ground from the ICSP header (GND on the schematic – this is digital ground you want analog ground!) for this method.**
4. **if you use this method (Method #2) refer to:**
   1. **Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

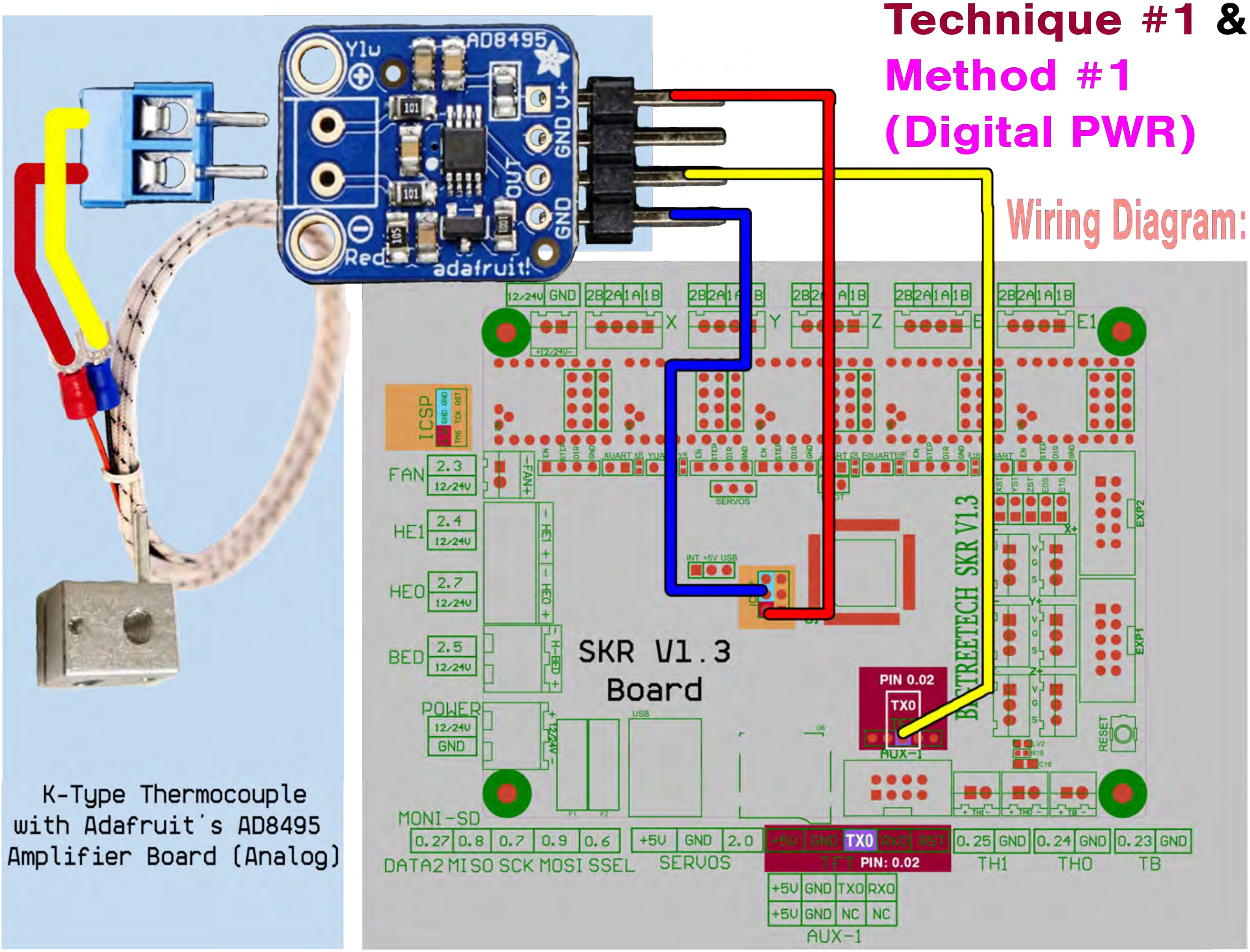
      This is the preferred and recommend way to connect up the AD8495 amplifier board.

- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your K-Type Thermcouple to the TFT header:

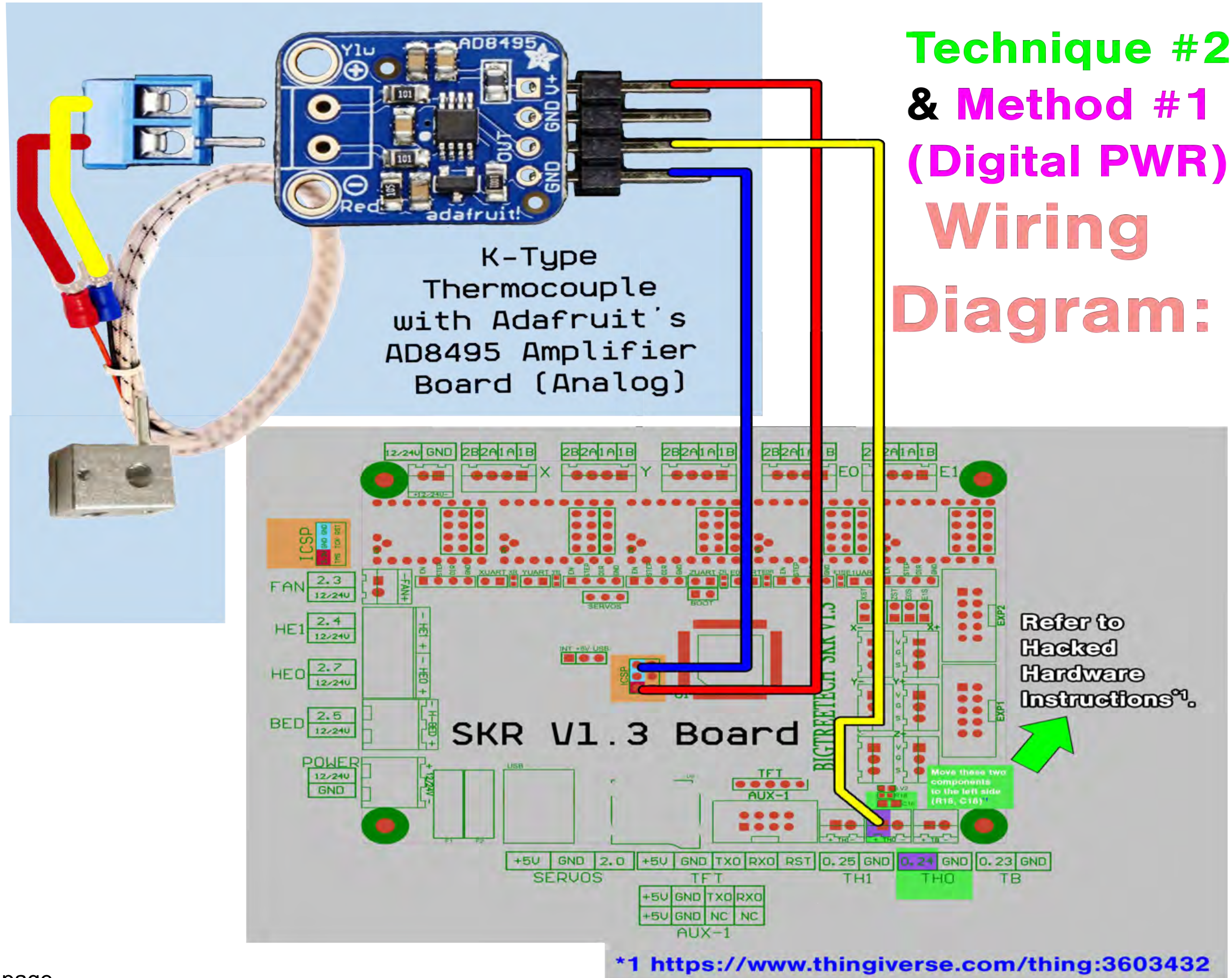


- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your K-Type Thermocouple to the TH0 connector:



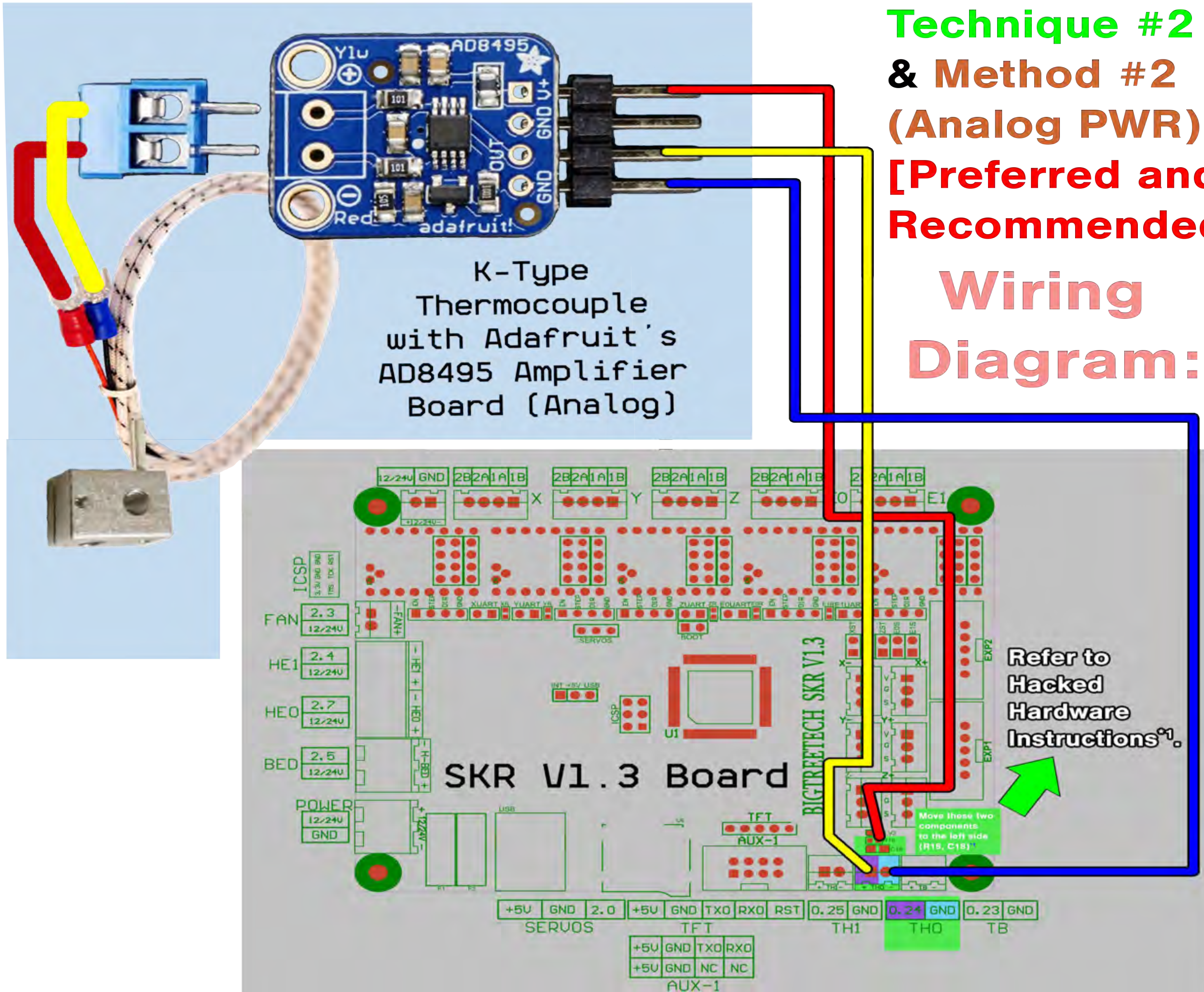
- Go to the next page.

APPENDIX L

Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your K-Type Thermocouple to the TH0 connector:

This is the preferred and recommend way to connect up the amplifier board.



Technique #2 & Method #2 (Analog PWR) [Preferred and Recommended] Wiring Diagram:

*1 https://www.thingiverse.com/thing:3603432

- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

If you are using Technique #2 goto the pages titled "Marlin 2.0x Firmware Setup via Technique #2".

If you are using Technique #1 goto the next page.

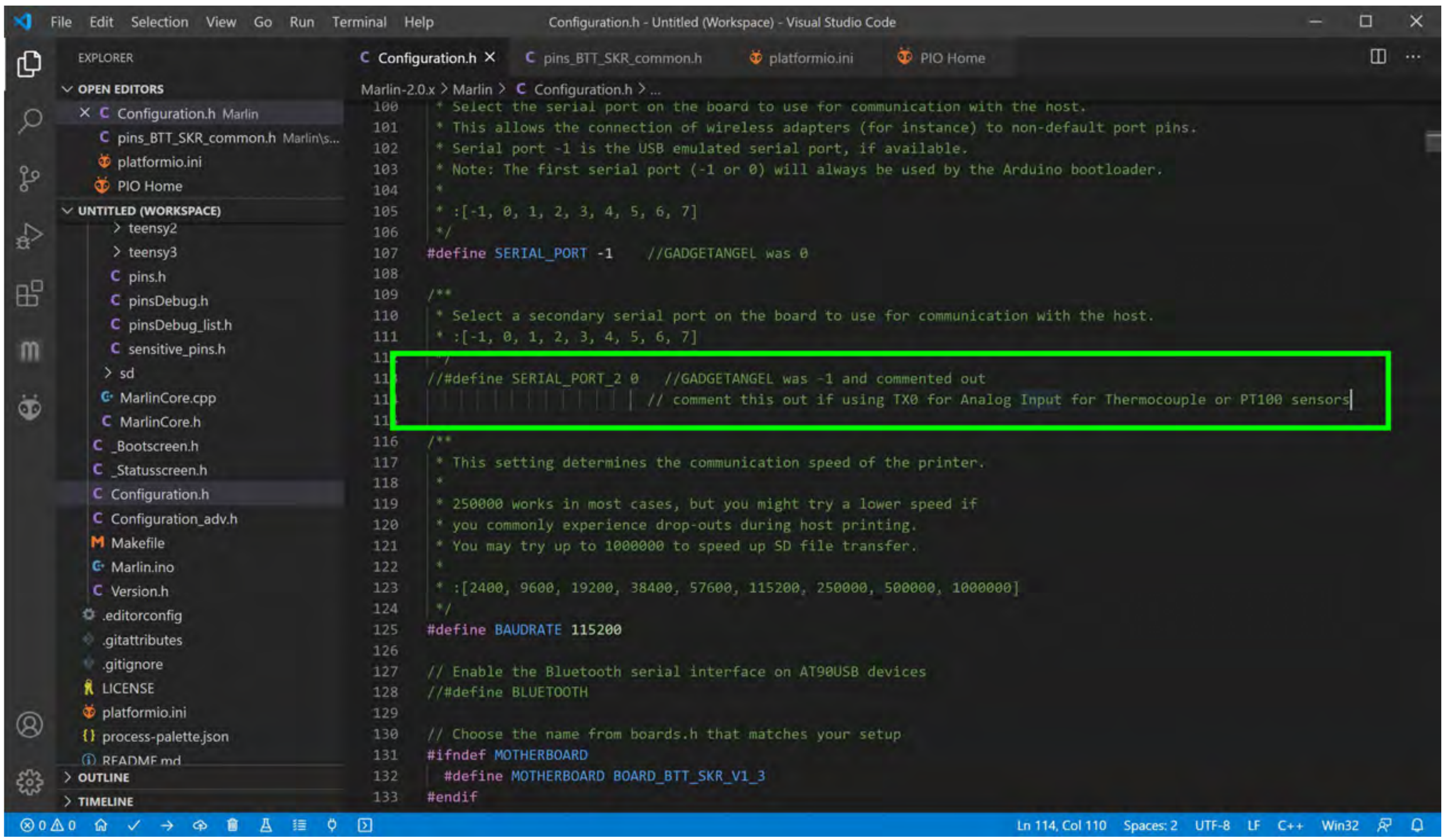
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

- **Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual**
- **disable the Marlin variable SERIRAL_PORT_2 or that you comment out the line by inserting two forward slashes ('//') at the beginning of the line.**



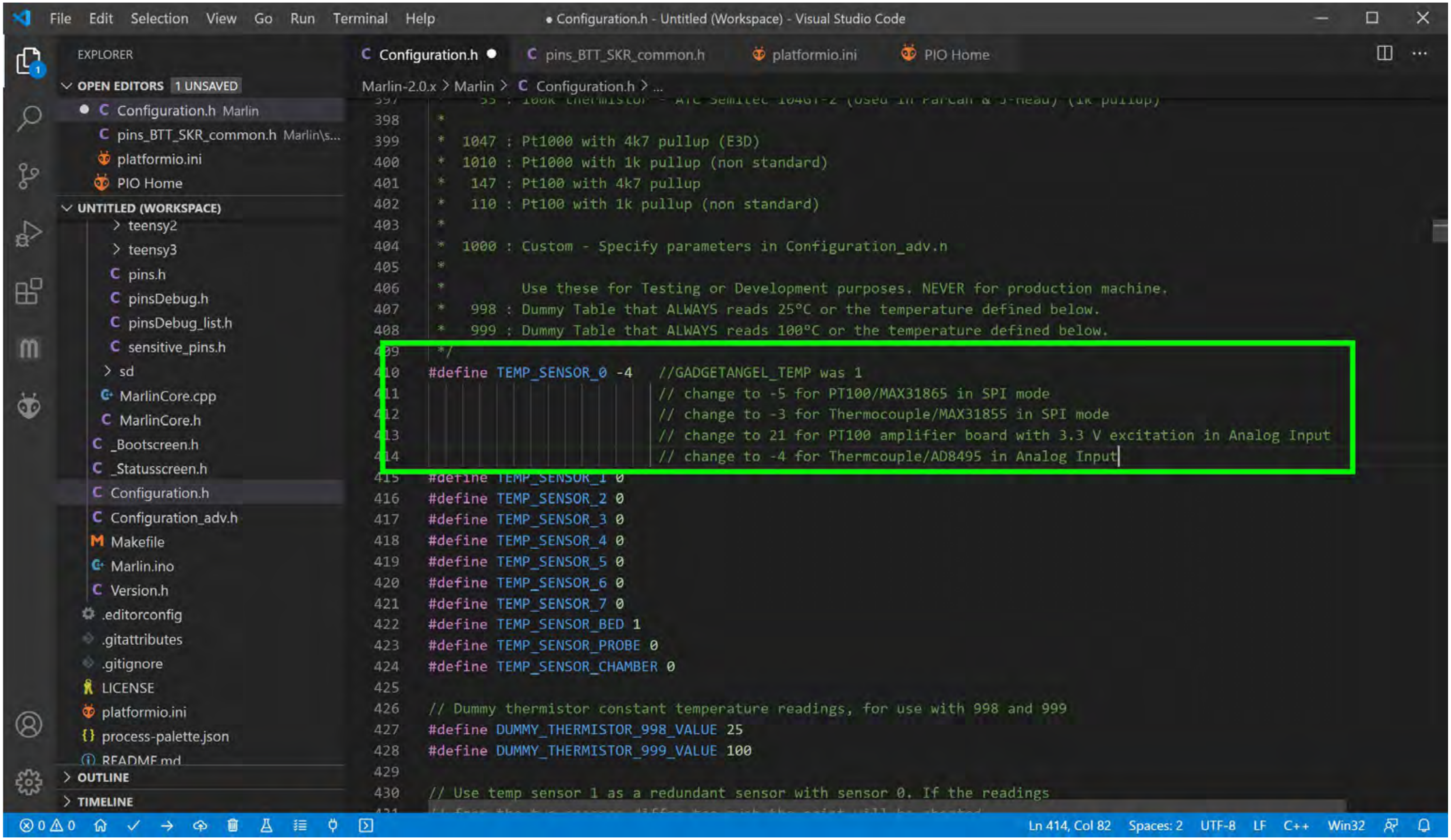
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

- **change the Marlin variable TEMP_SENSOR_0 to -4**



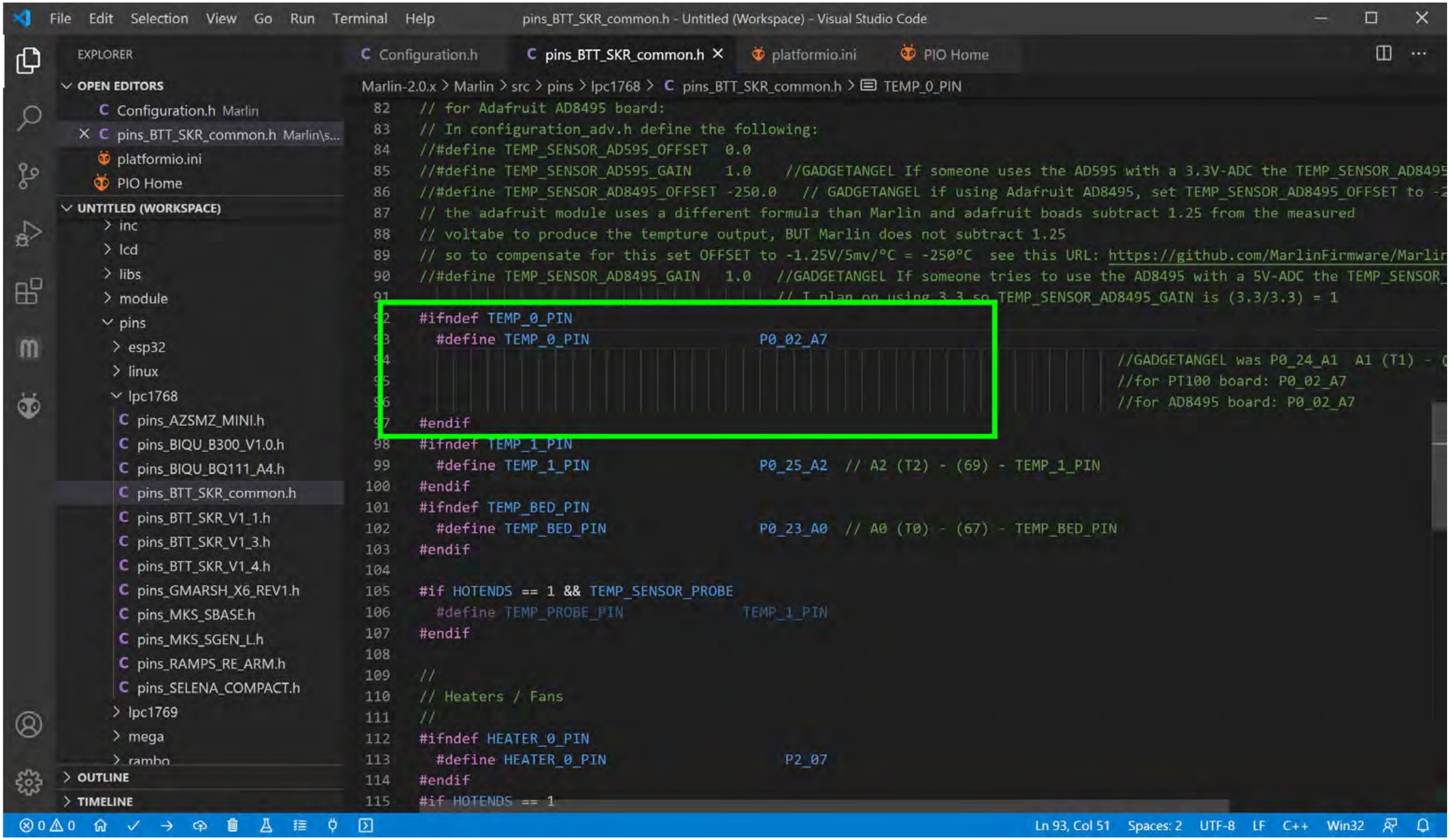
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

- **change the Marlin variable TEMP_0_PIN to P0_02_A7**



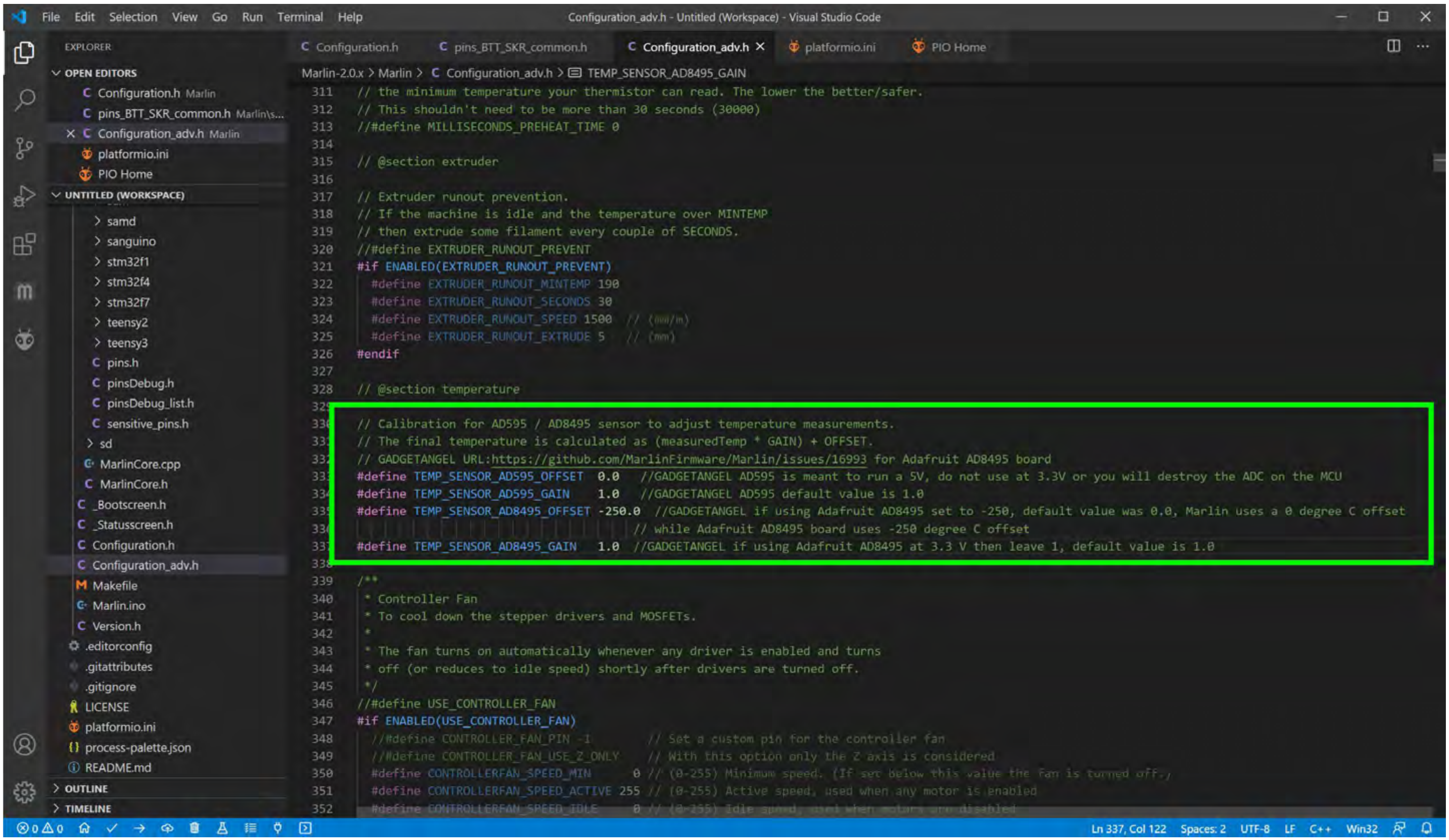
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

- **change the Marlin variable TEMP_SENSOR_AD8495_OFFSET to -250.0**
- **ensure the Marlin variable TEMP_SENSOR_AD8495_GAIN is set to 1.0**



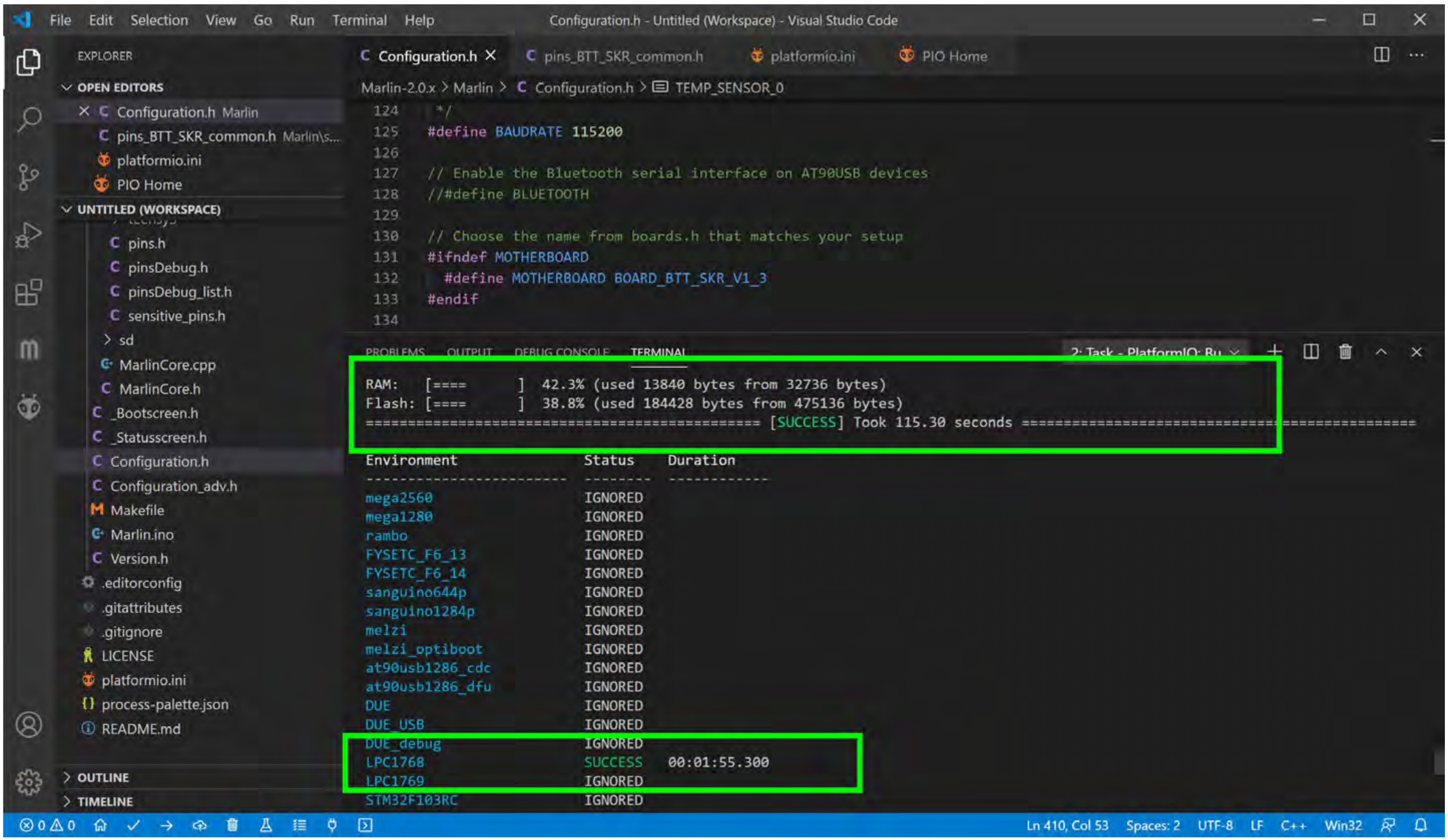
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

- **Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.**



- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #1

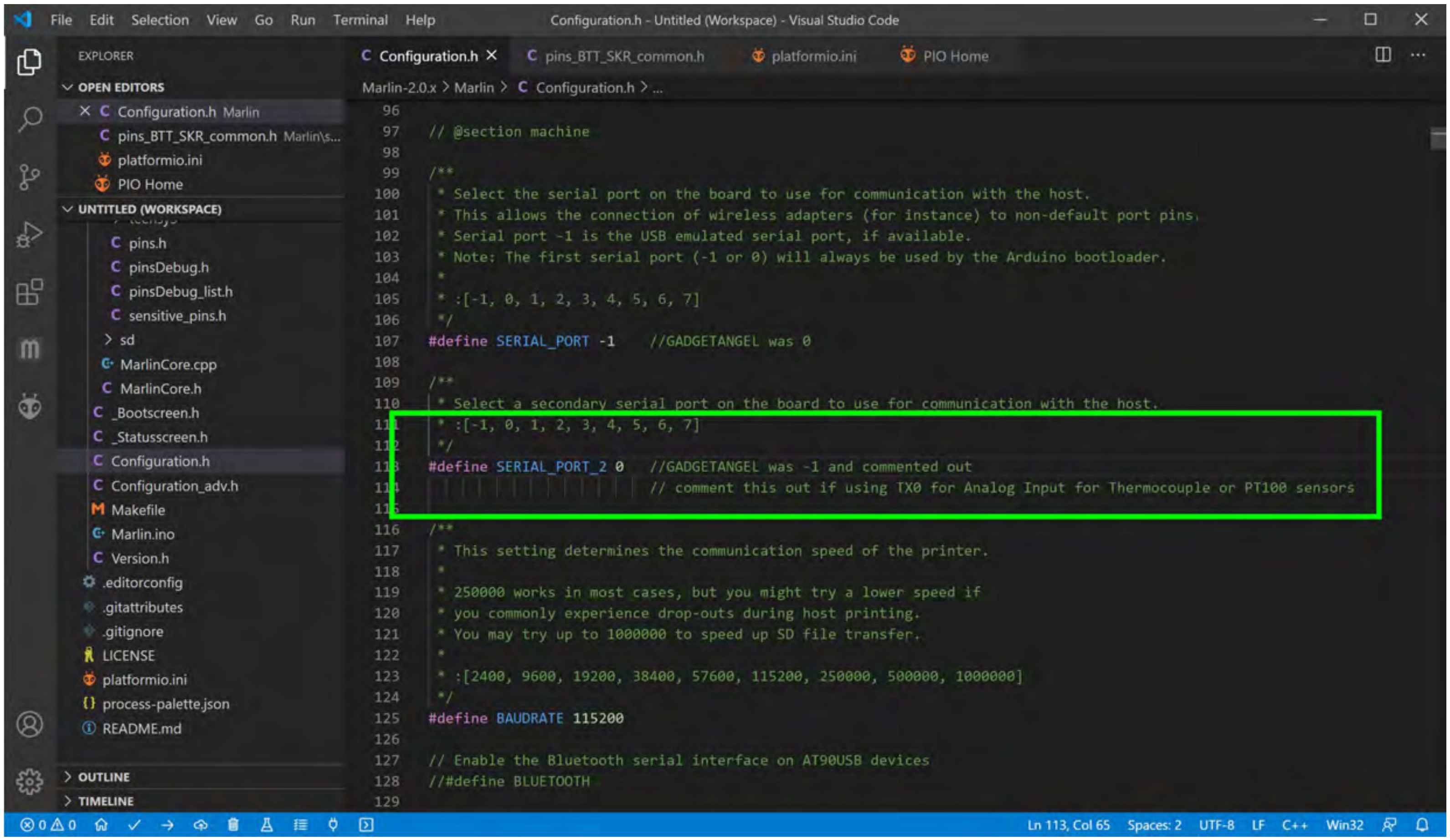
- **keep the Marlin settings the same (as above) until the K-Type thermocouple sensor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the AD8495 amplifier board and the K-Type thermocouple sensor using Technique #1!!**

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #2

- Perform the Marlin Firmware steps found in the manual called "SKR V1.3 Stepper Driver Configuration.pdf " in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual
- if disabled pleanse enable the Marlin variable SERIRAL_PORT_2 or that you delete the two forward slashes ("//") from the beginning of the line.



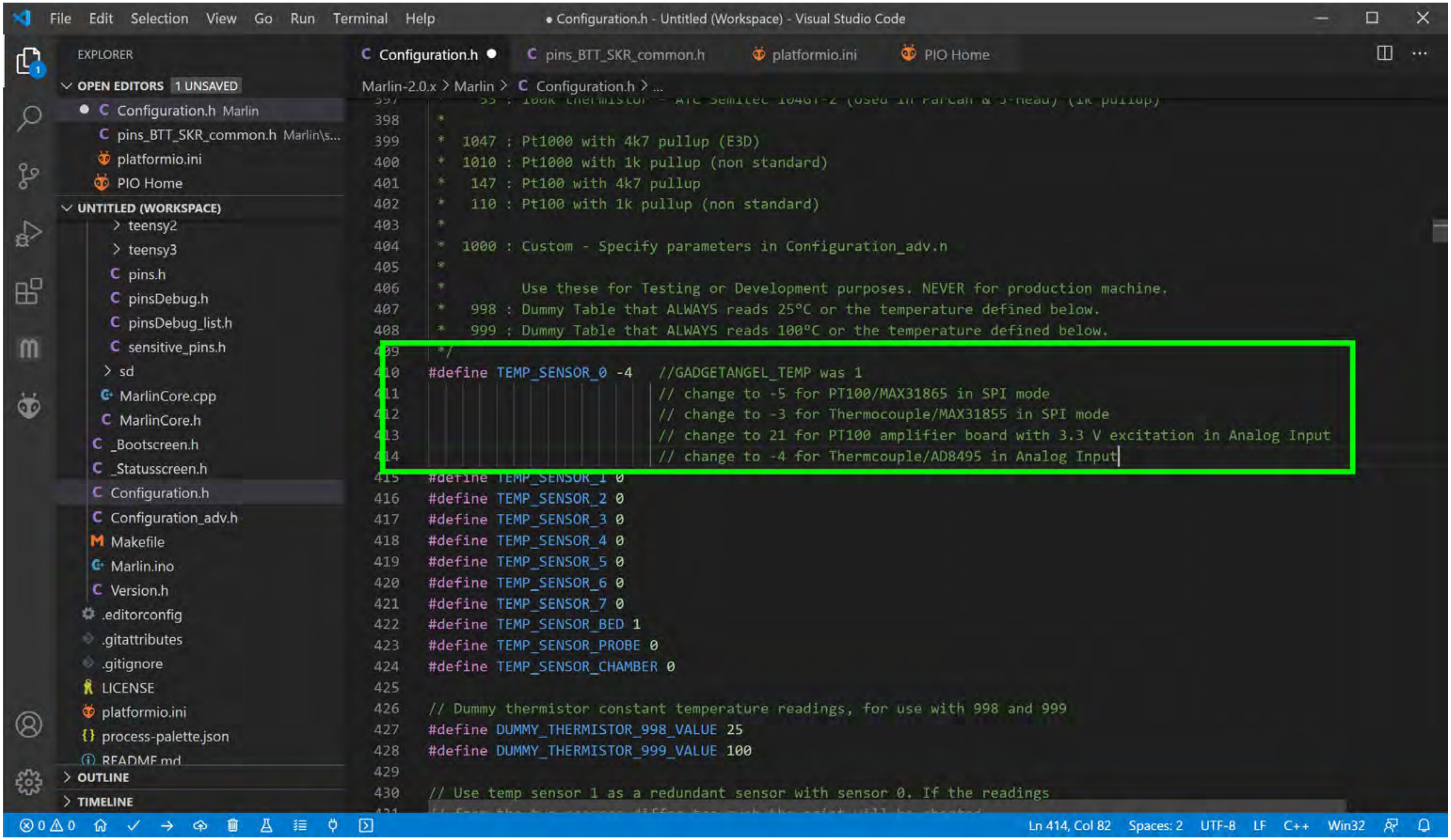
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #2

- **change the Marlin variable TEMP_SENSOR_0 to -4**



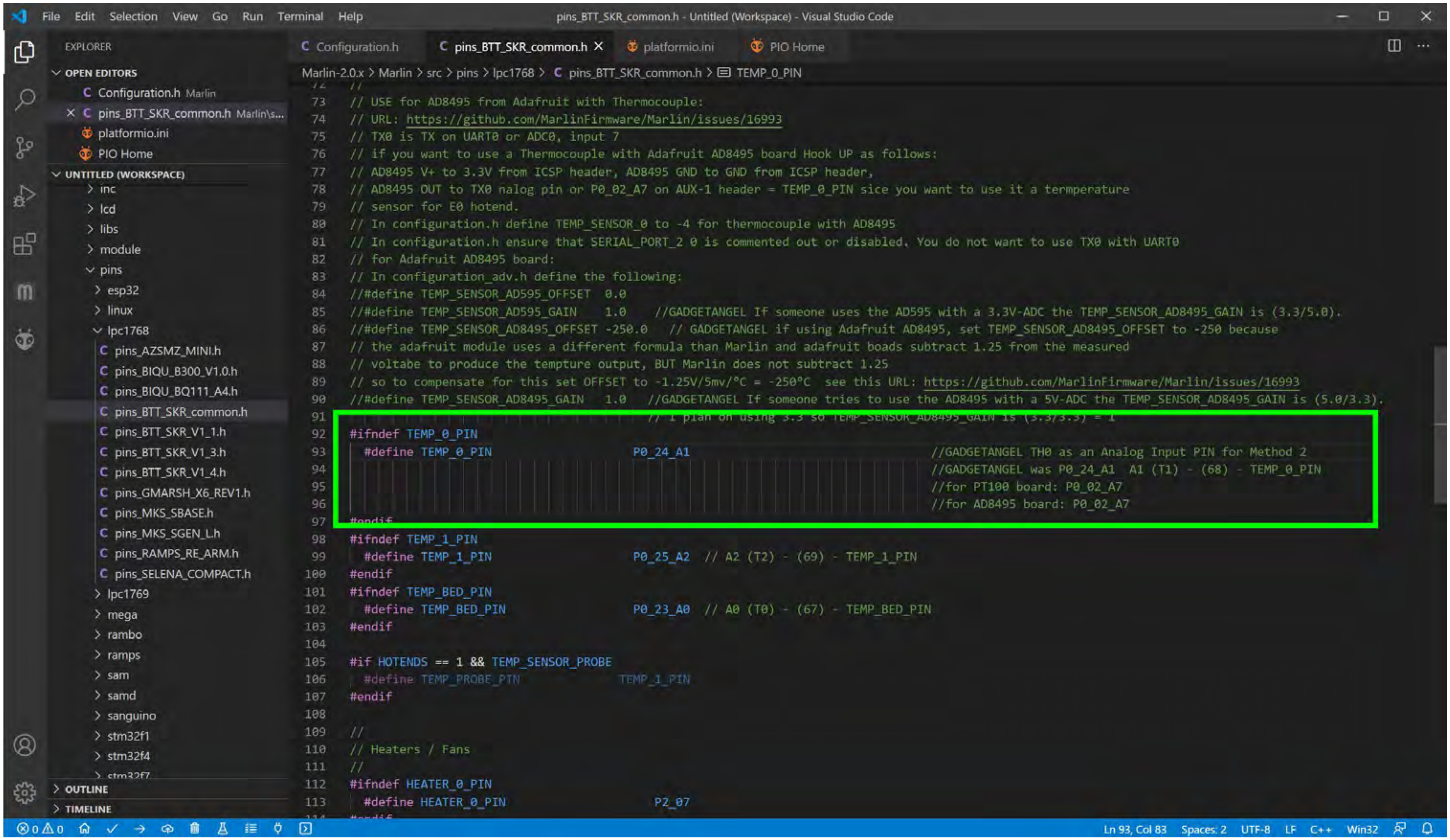
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #2

- **change the Marlin variable TEMP_0_PIN to P0_24_A1**



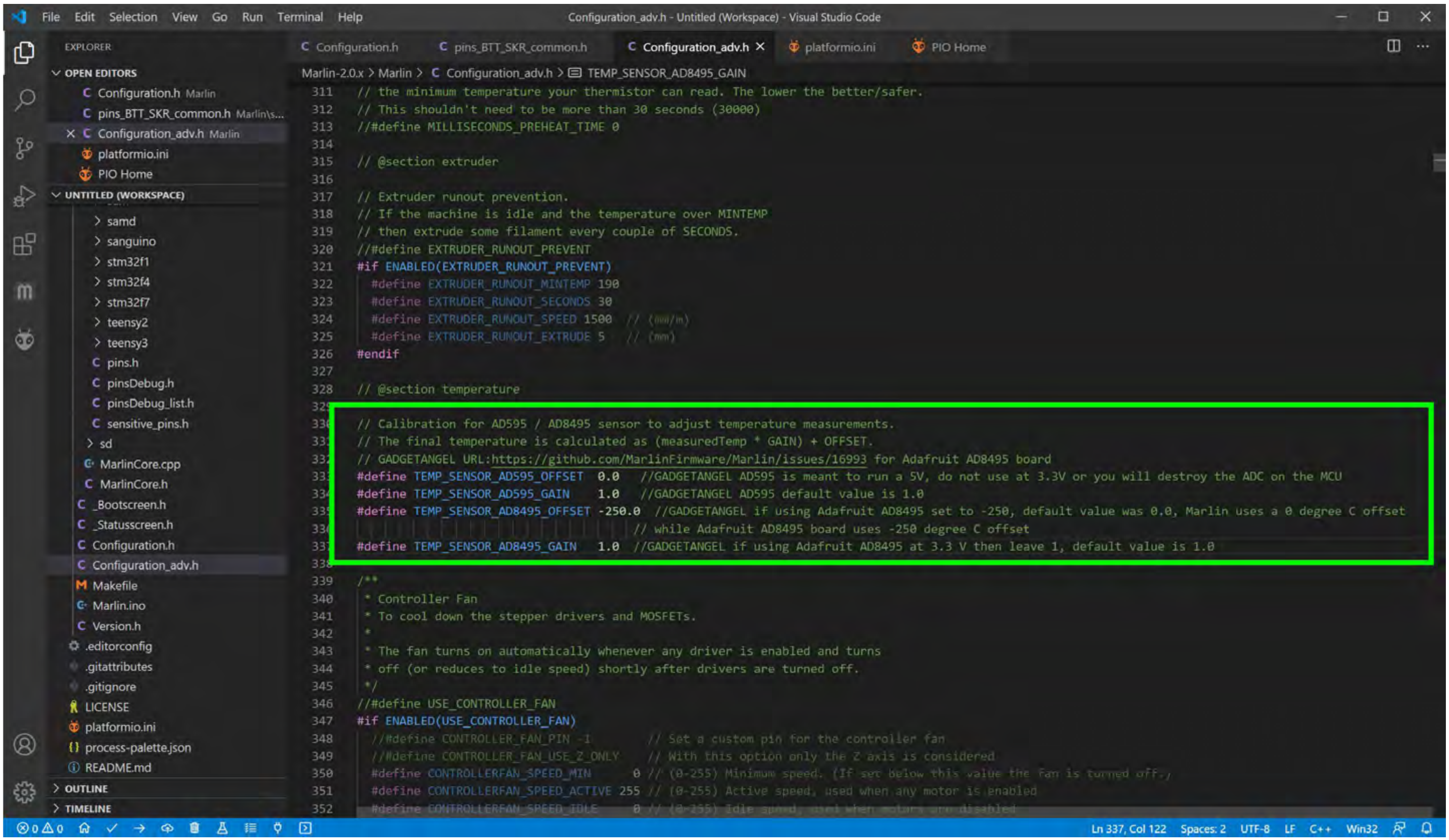
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #2

- **change the Marlin variable TEMP_SENSOR_AD8495_OFFSET to -250.0**
- **ensure the Marlin variable TEMP_SENSOR_AD8495_GAIN is set to 1.0**



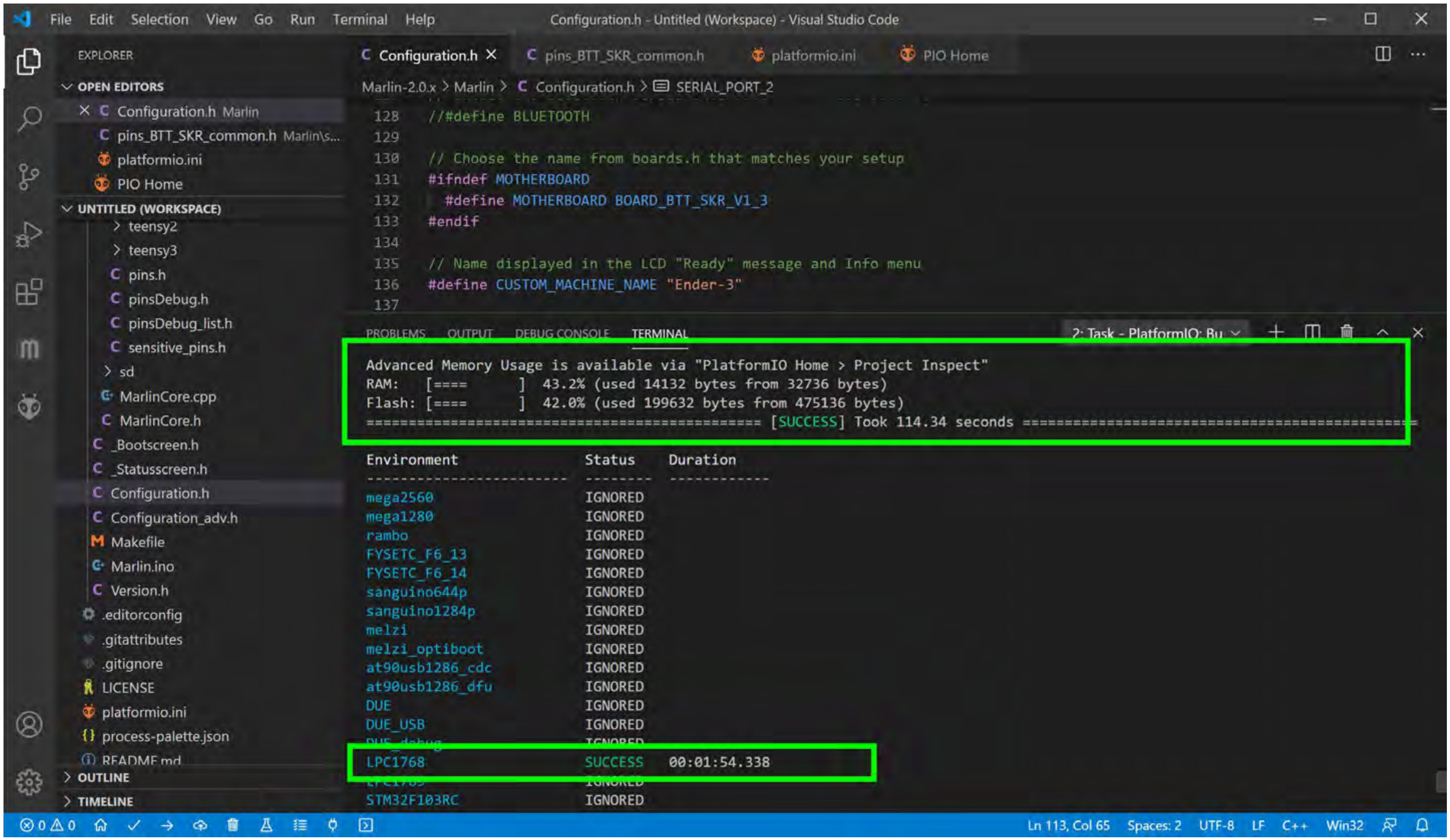
- Go to the next page.

APPENDIX L

Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Marlin 2.0.x Firmware Setup via Technique #2

- **Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.**



- Go to the next page.

Marlin 2.0.x Firmware Setup via Technique #2

- **keep the Marlin settings the same (as above) until the K-Type thermocouple sensor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the AD8495 amplifier board and the K-Type thermocouple sensor using Technique #2!!**