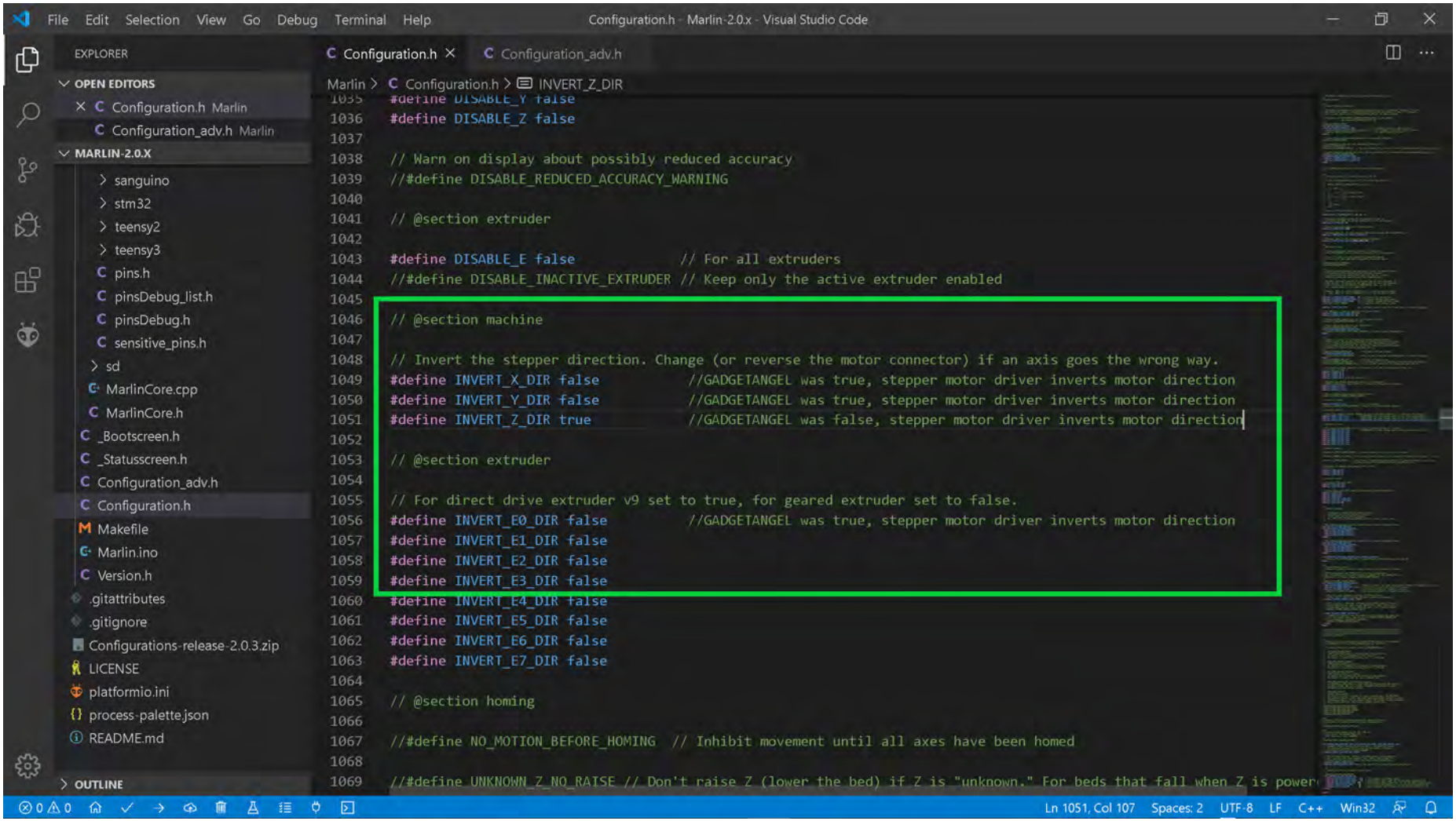
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

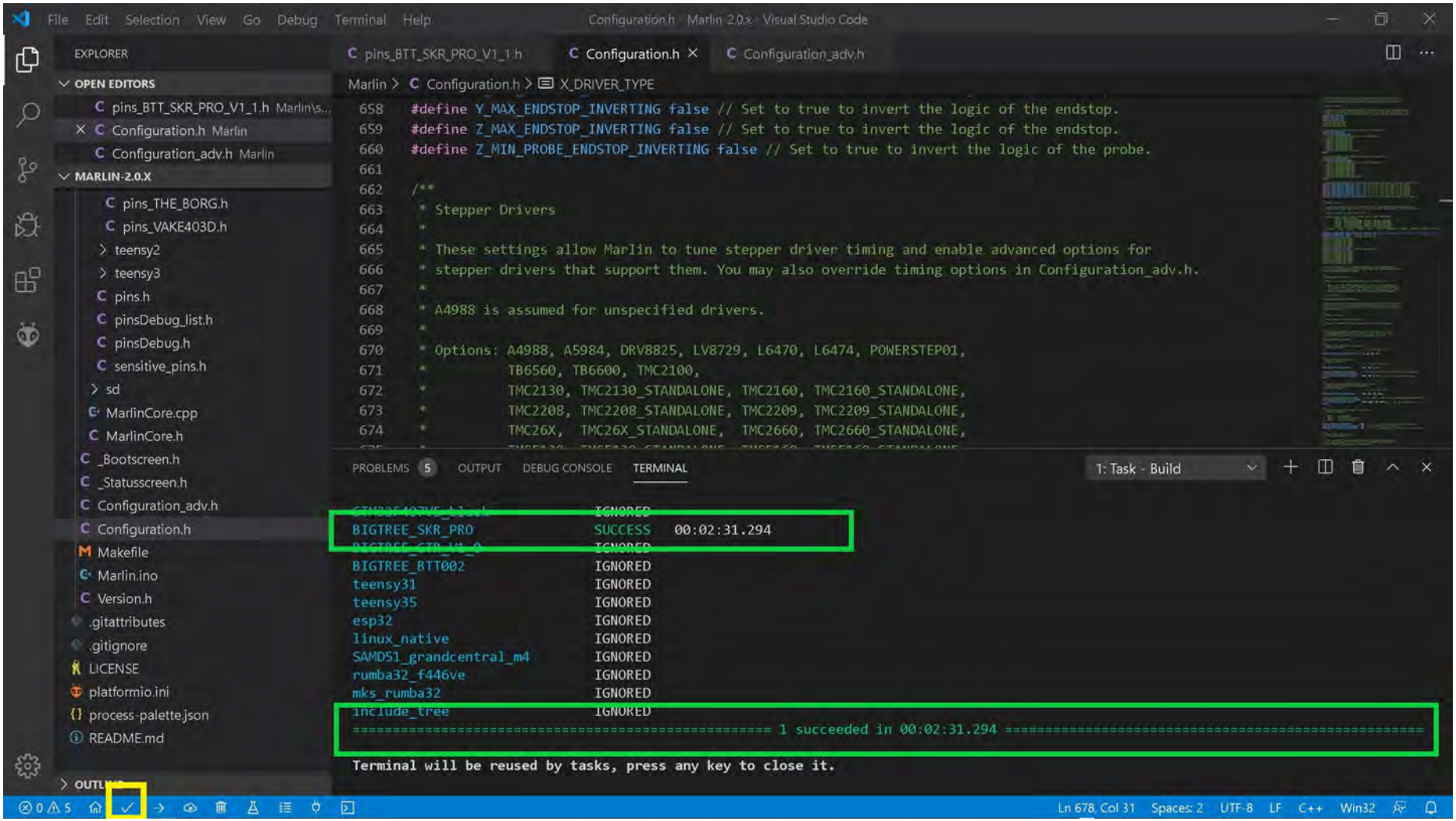
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 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

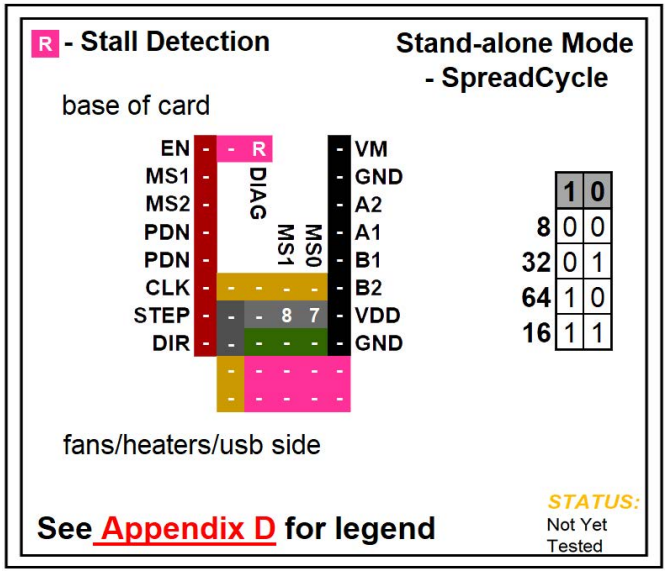
- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand-alone mode for stealthChop. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



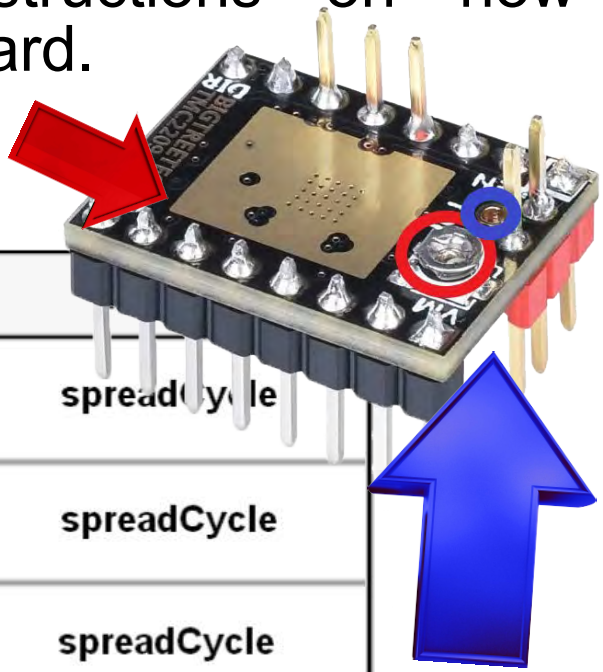
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle



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



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

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

Driving Current Calculation Formula		
R_S(Typical Sense Resistor)= 0.11Ω	$I_{MAX}=V_{ref}$ See Appendix B #5. Use 50% to 90% as shown below: $I_{MAX}=I_{MAX} * 0.90$	$V_{ref}=I_{MAX}$ See Appendix B #5. Use 50% to 90% as shown below: $V_{ref}=V_{ref} * 0.90$

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

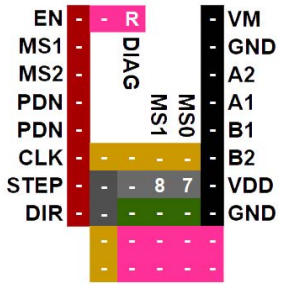
BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

R - Stall Detection

Stand-alone Mode - SpreadCycle

base of card



	1	0
8	0	0
32	0	1
64	1	0
16	1	1

fans/heaters/usb side

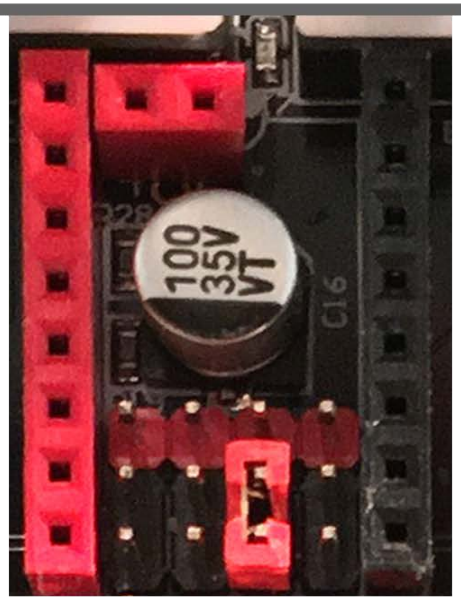
See Appendix D for legend

STATUS: Not Yet Tested

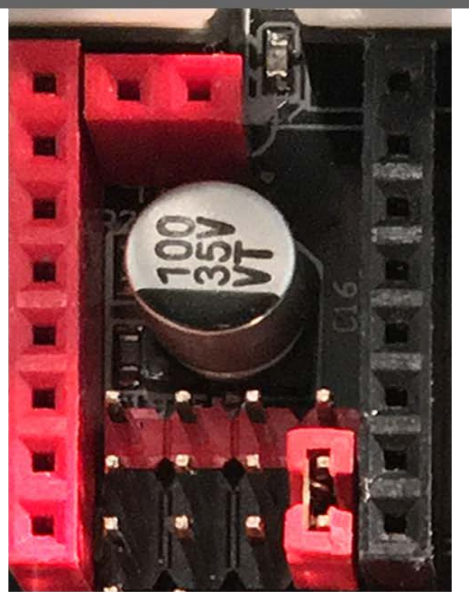
SKR PRO V1.1 LEGEND for Binary State Stepper Drivers

Low ⇒ set Jumper between rows 2 and 3

Row:
1
2
3

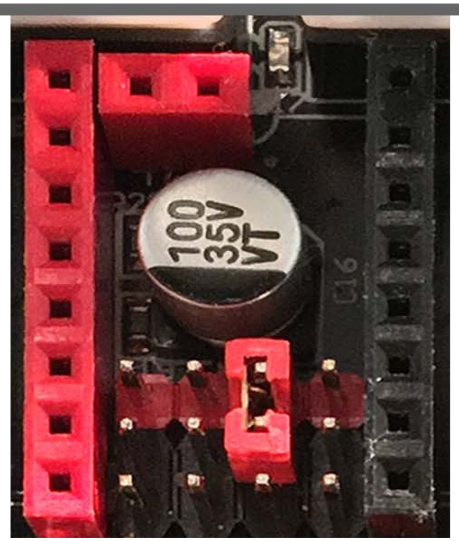


Row:
1
2
3

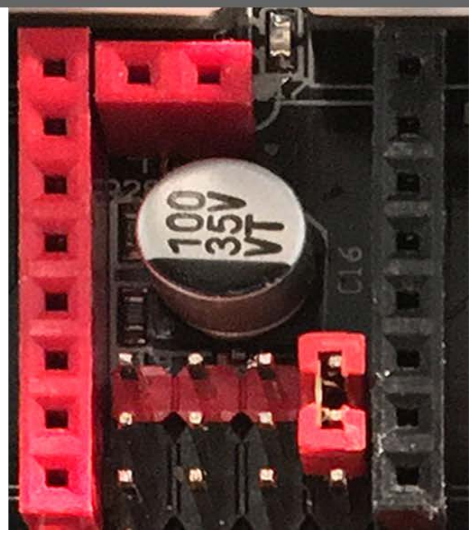


High ⇒ set Jumper between rows 1 and 2

Row:
1
2
3



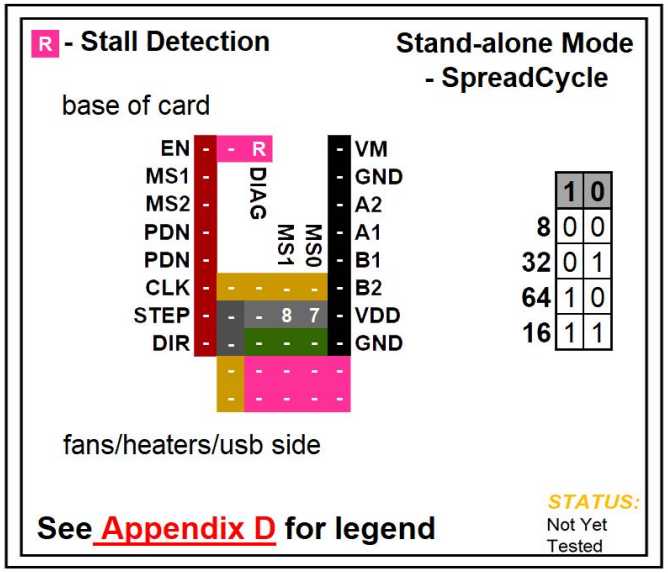
Row:
1
2
3



This is a Jumper:

BIQU TMC2209 V1.2

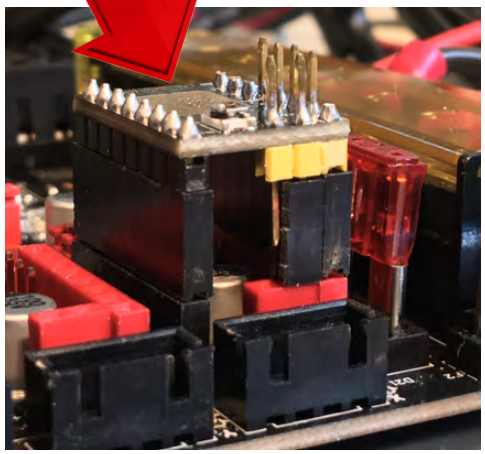
Stand-alone Mode for SpreadCycle



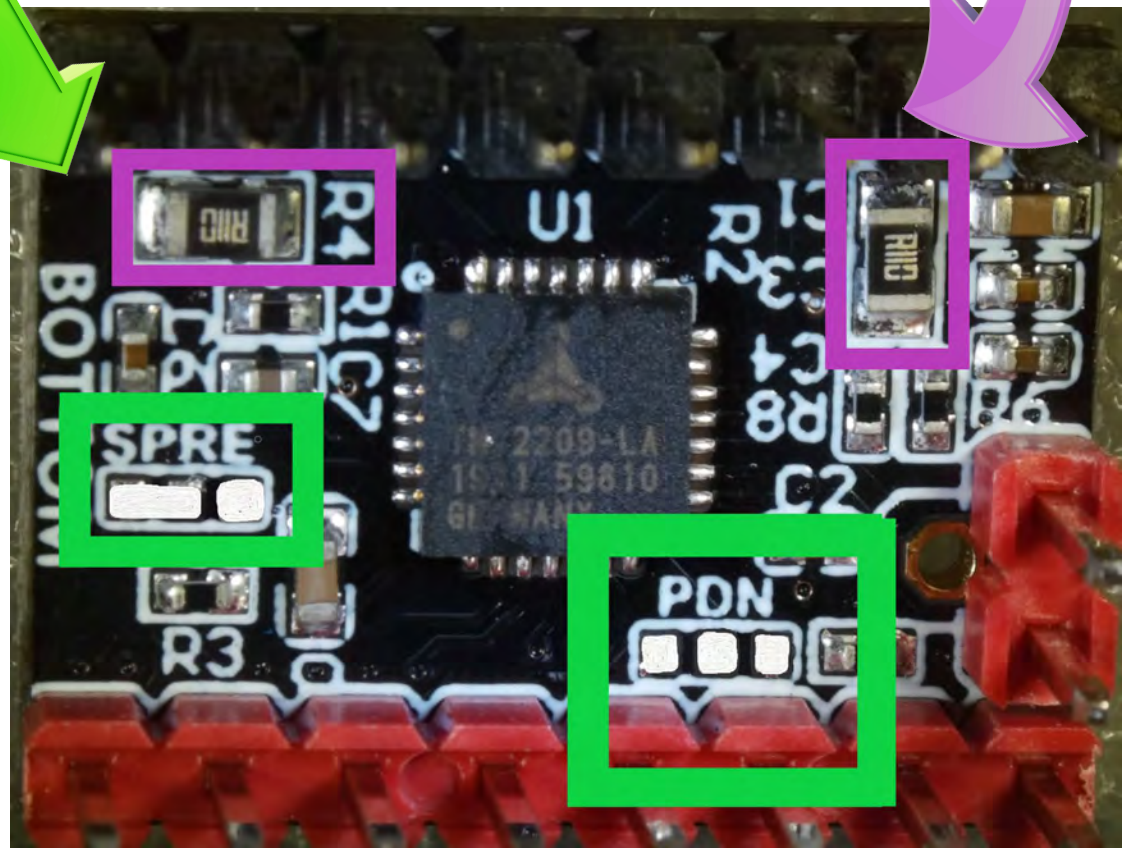
NOTE: The SPRE jumper is located on the bottom of the driver board. In Standalone Mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board. The **PURPLE boxes** below show the location of the current sense resistors (R_s) for the TMC2209.

Note: To switch to stand-alone mode, none of the PDN pads should be bridged. The picture below, as seen in the **GREEN** boxes, shows stand-alone mode with SpreadCycle.

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



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



BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

Stand-alone SpreadCycle Mode

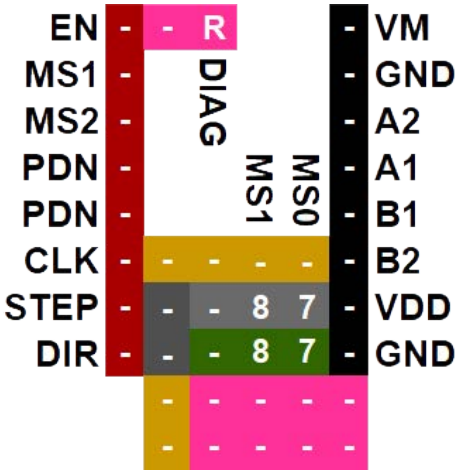
Stand-alone Mode

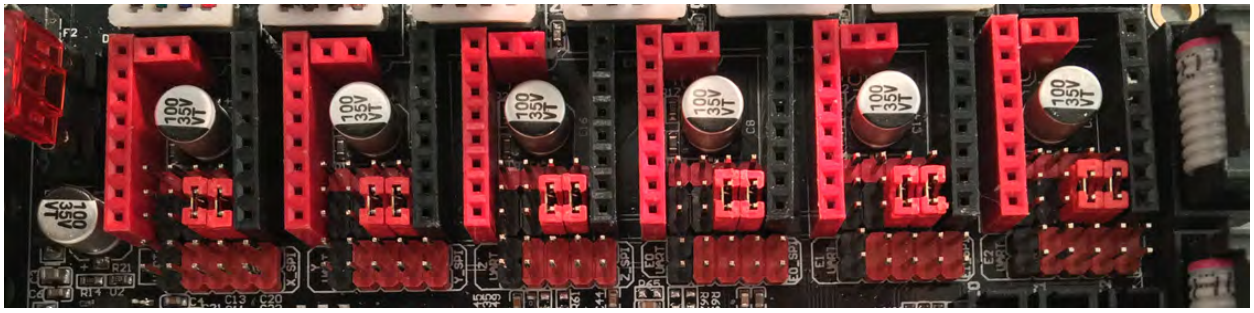
1 / 8

Interpolation:

1 / 256

SpreadCycle





See Appendix D for legend

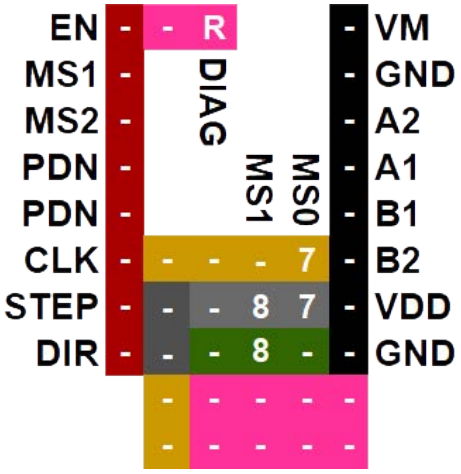
Stand-alone Mode

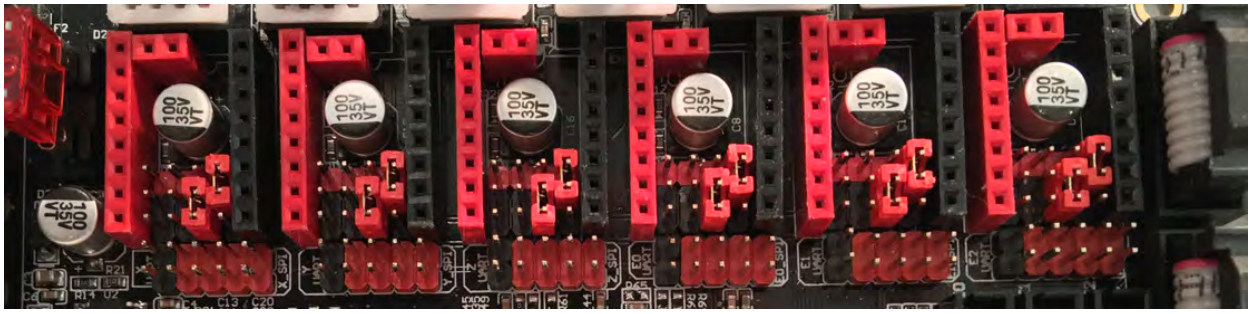
1 / 32

Interpolation:

1 / 256

SpreadCycle





See Appendix D for legend

BIQU TMC2209 V1.2

Stand-alone Mode for SpreadCycle

Stand-alone SpreadCycle Mode

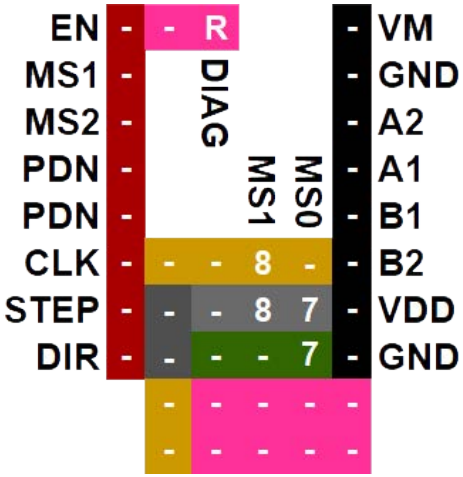
Stand-alone Mode

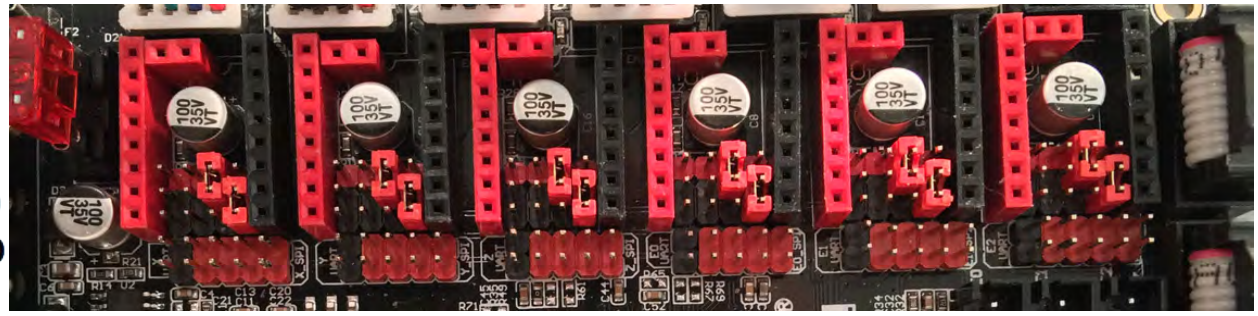
1 / 64

Interpolation:

1 / 256

SpreadCycle





See Appendix D for legend

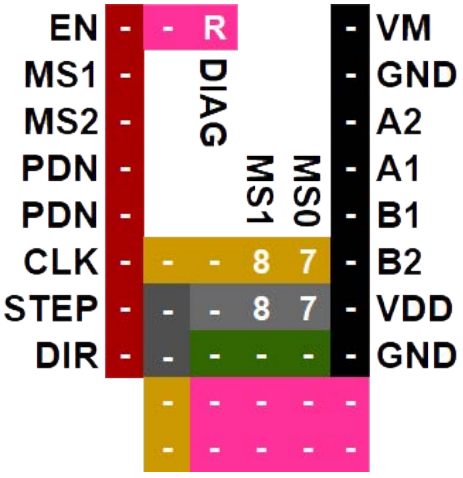
Stand-alone Mode

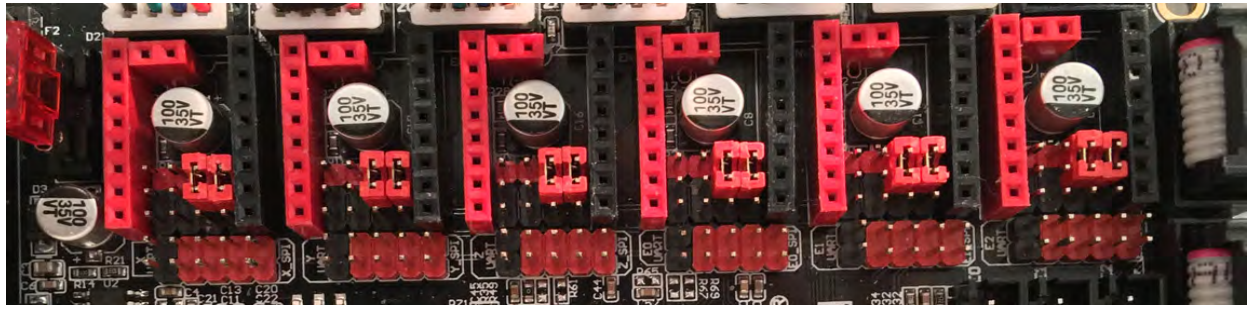
1 / 16

Interpolation:

1 / 256

SpreadCycle



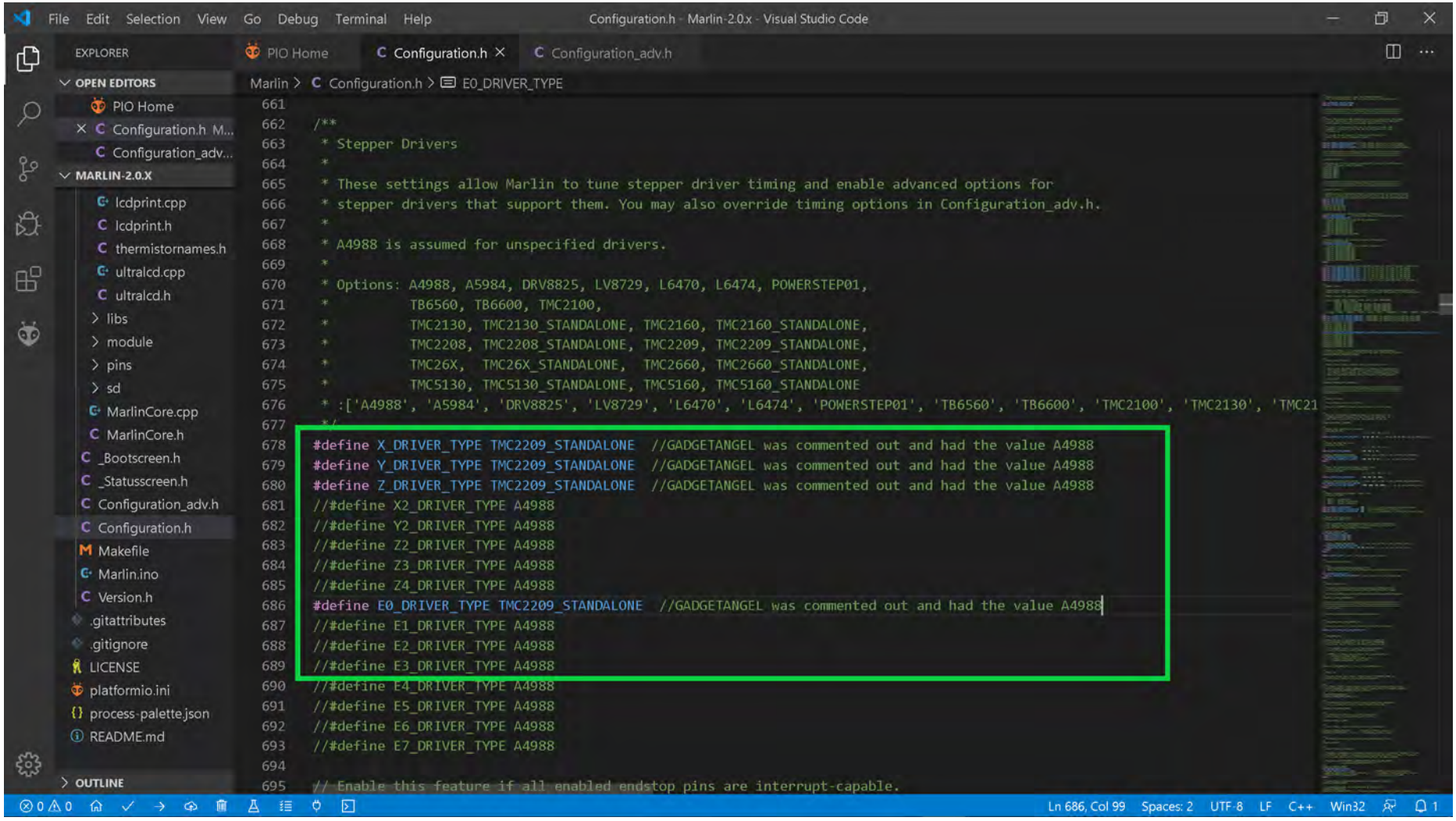


See Appendix D for legend

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.

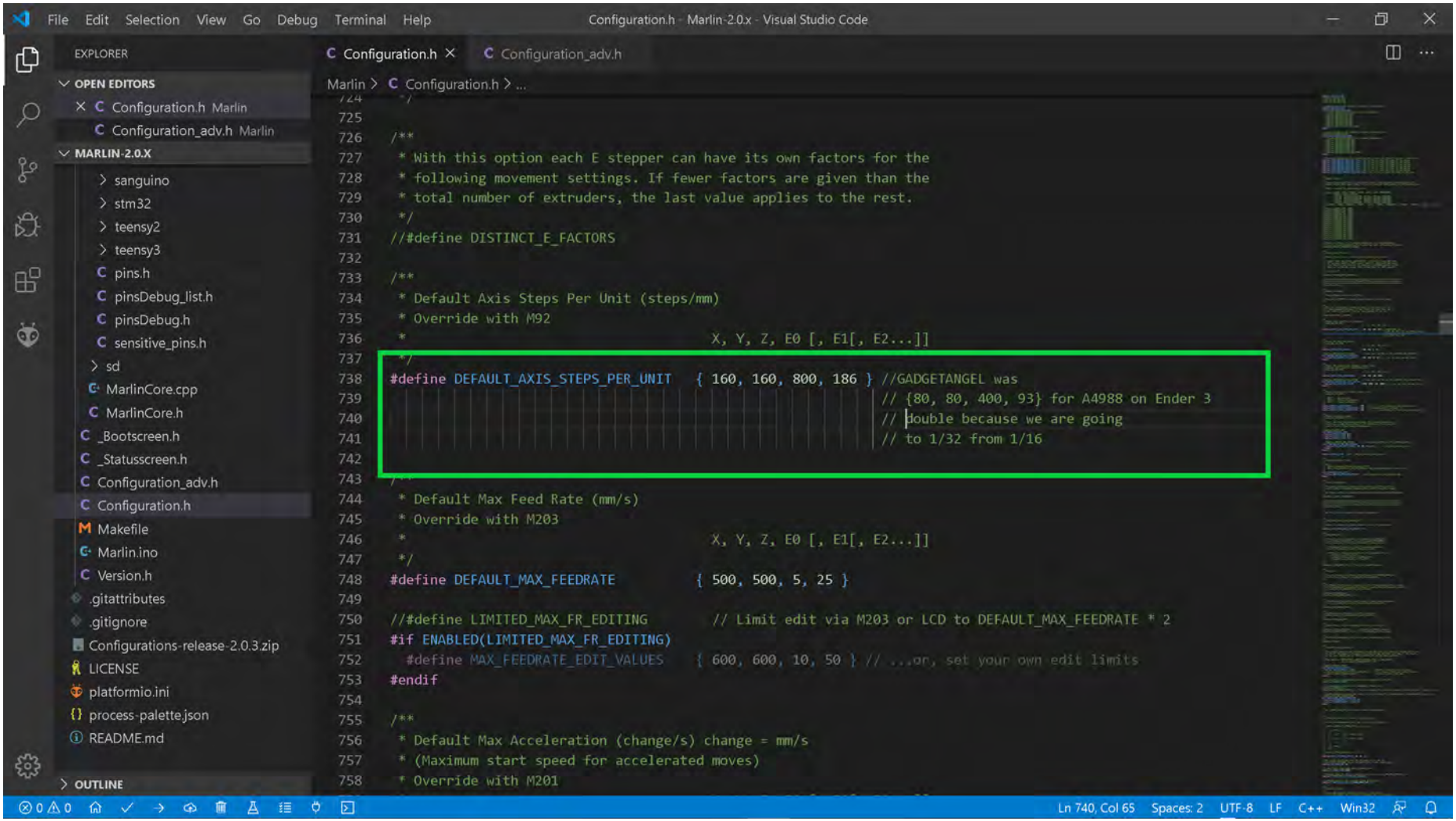
- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand-alone mode. When two "//" appear at the beginning 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

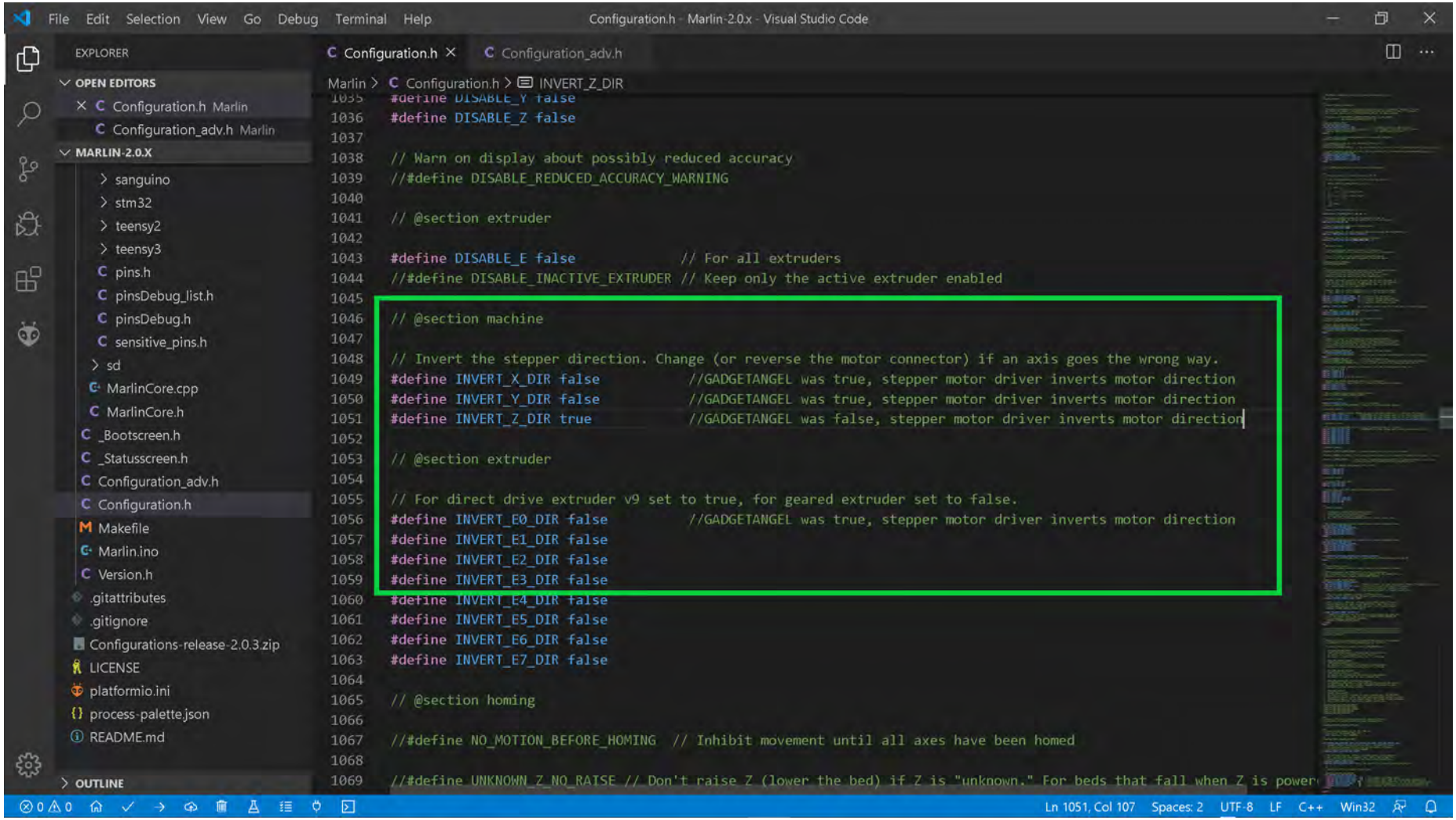
- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust 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

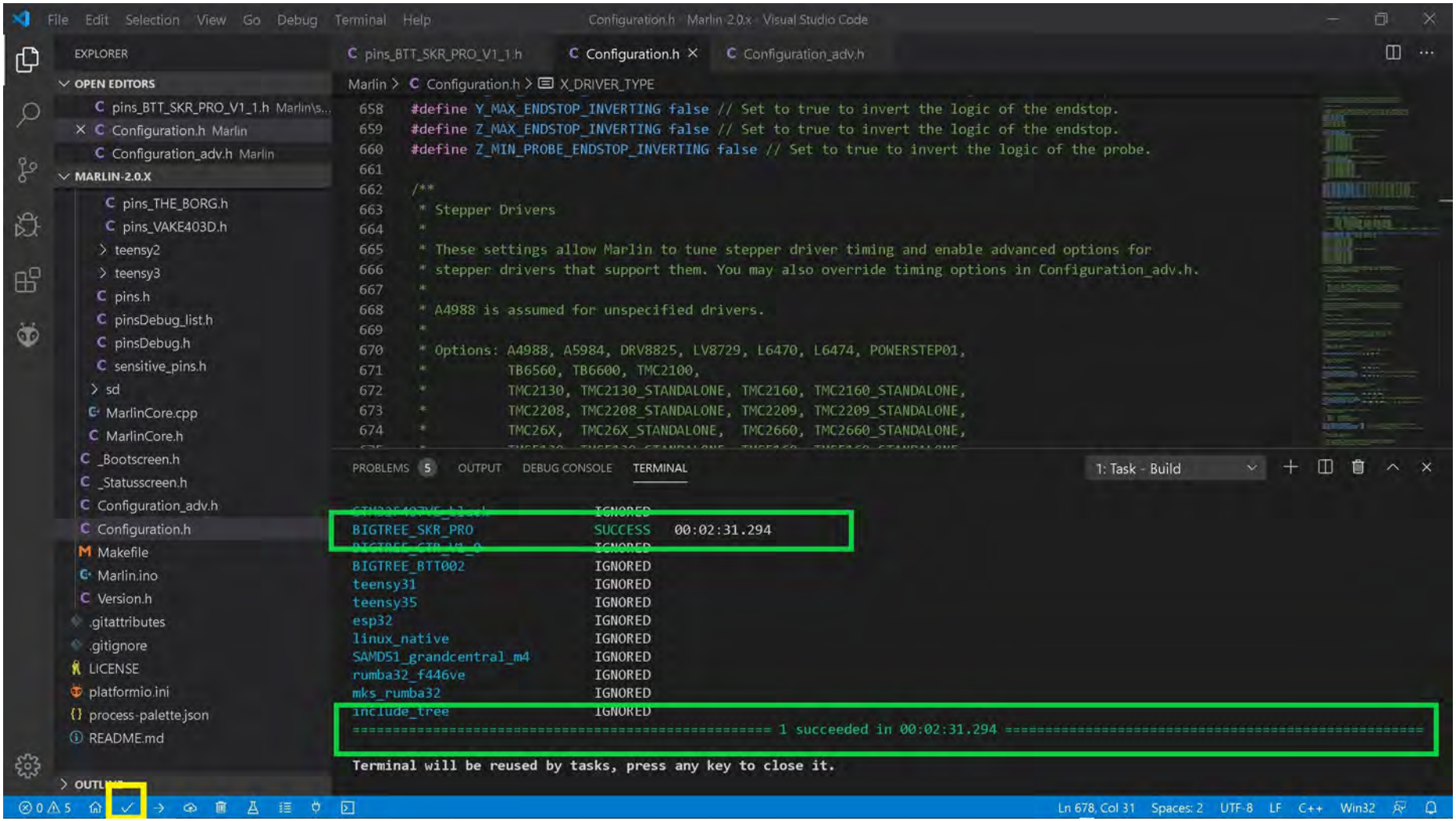
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 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.



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

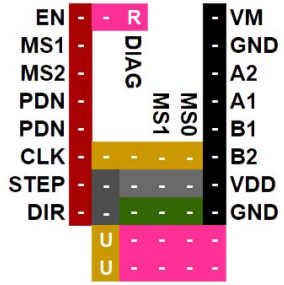
BIQU TMC2209 V1.2

UART Mode

R - Stall Detection

UART Mode

base of card



fans/heaters/usb side

See Appendix D for legend

STATUS:
Not Yet Tested

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

See the next page for further information.

Driver Chip	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

R - Stall Detection

UART Mode

base of card

EN			R		VM
MS1			DIAG		GND
MS2					A2
PDN				MS1	A1
PDN				MS0	B1
CLK					B2
STEP					VDD
DIR					GND
	U				
	U				

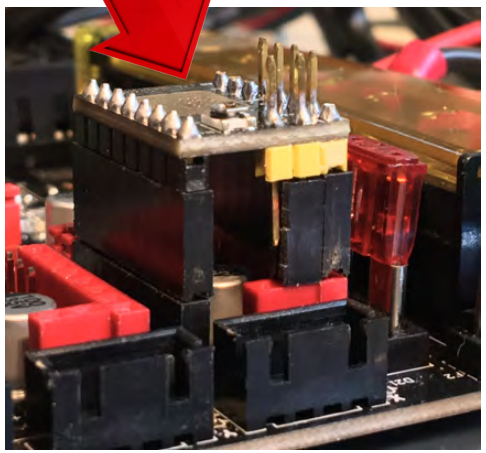
fans/heaters/usb side

STATUS: Not Yet Tested

See Appendix D for legend

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

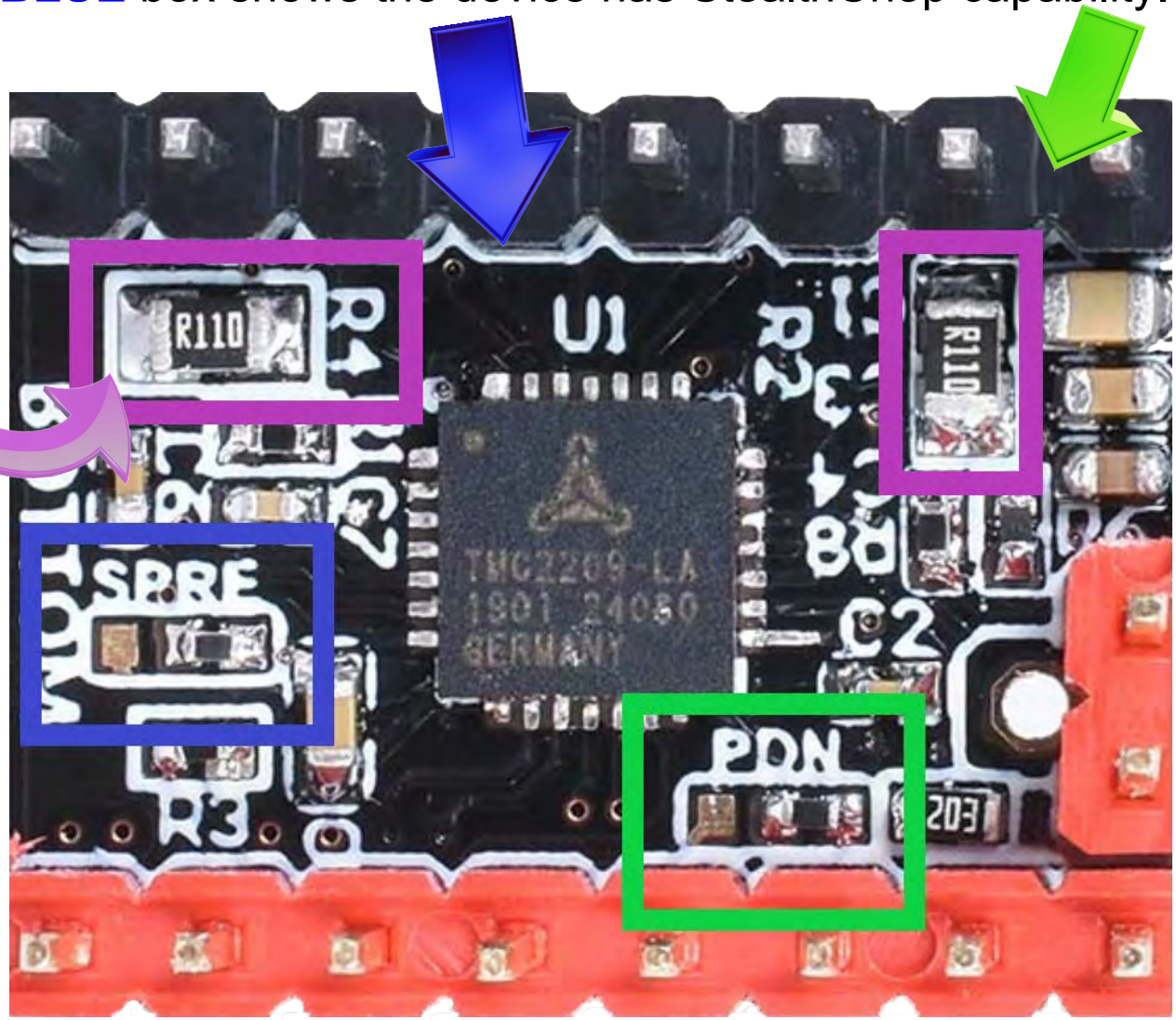
Note: When the stall-guard function is not used, the stall-guard pin ("R") of the TMC2209 must be removed (desoldered) or use long pin header risers so that the "R" pin does not connect to the SKR PRO V1.1 board



The PURPLE boxes show the location of the current sense resistors (R_S).

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

The BLUE box shows the device has StealthChop capability.



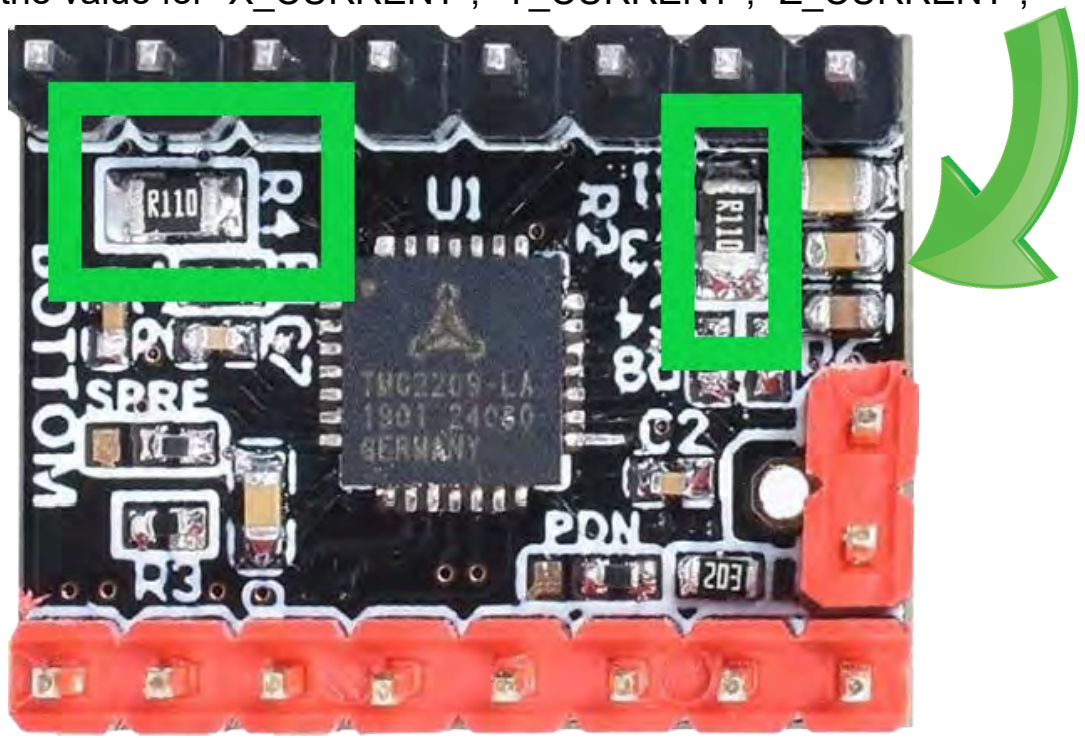
BIQU TMC2209 V1.2

UART Mode

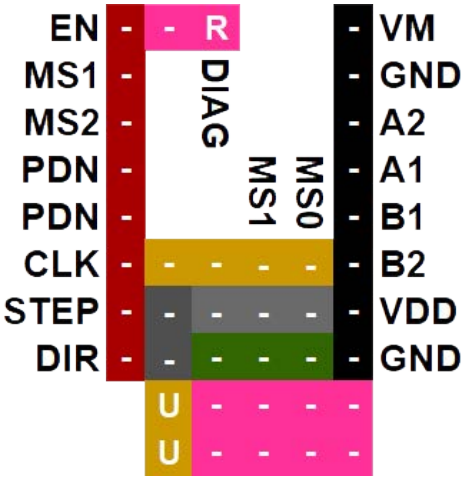
UART Mode
R - Stall Detection

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

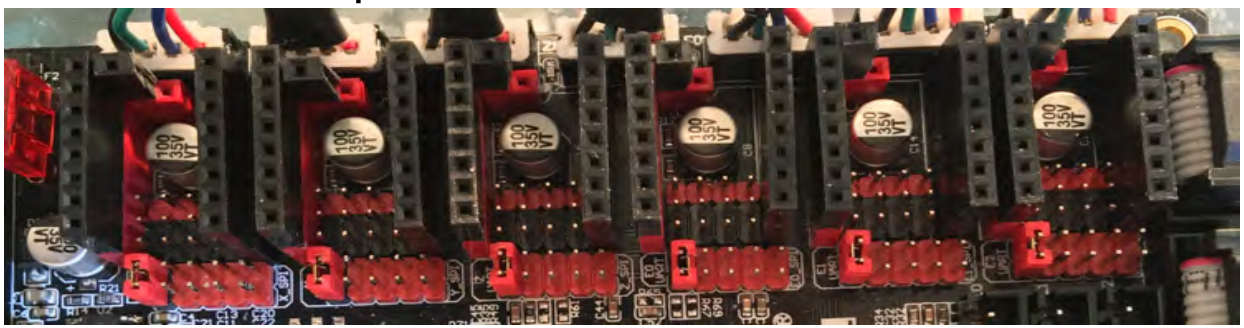
R_s = R050 is 0.05 Ohms
R_s = R062 is 0.062 Ohms
R_s = R068 is 0.068 Ohms
R_s = R075 is 0.075 Ohms
R_s = R100 is 0.1 Ohms
R_s = R110 is 0.11 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms



UART



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

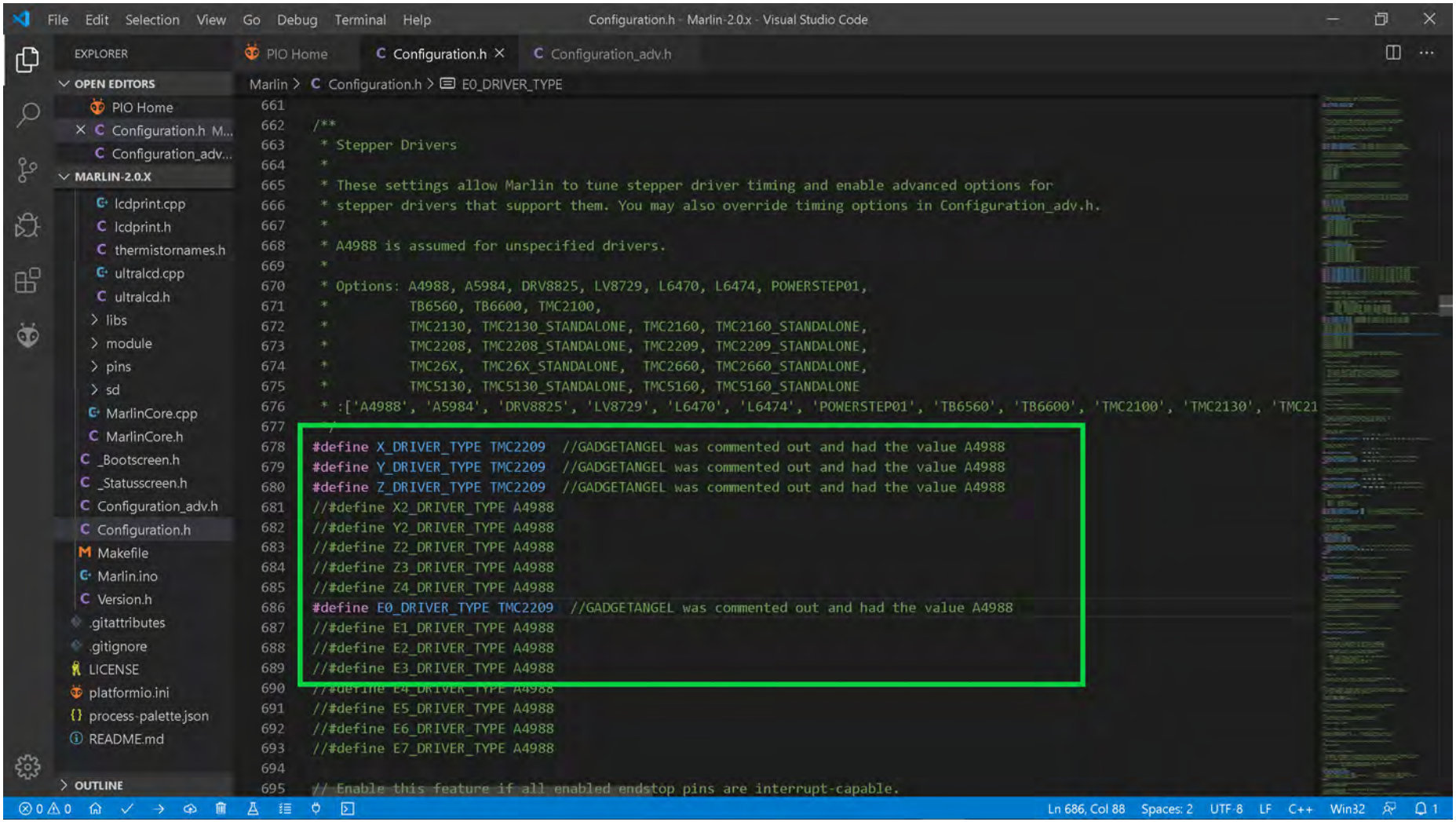


See Appendix D for legend

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.

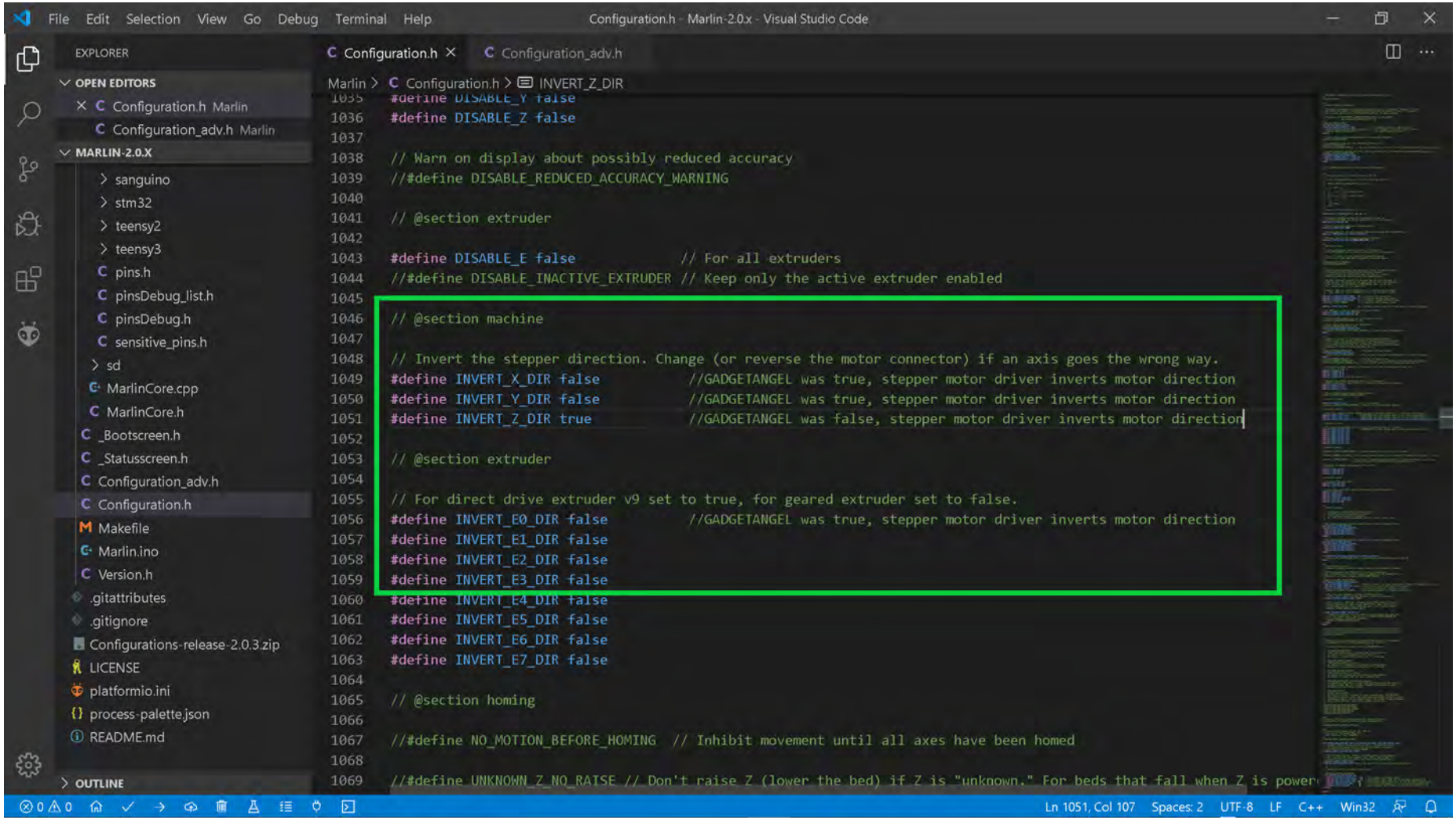
- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in UART mode. When two "//" appear at the beginning of a line 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

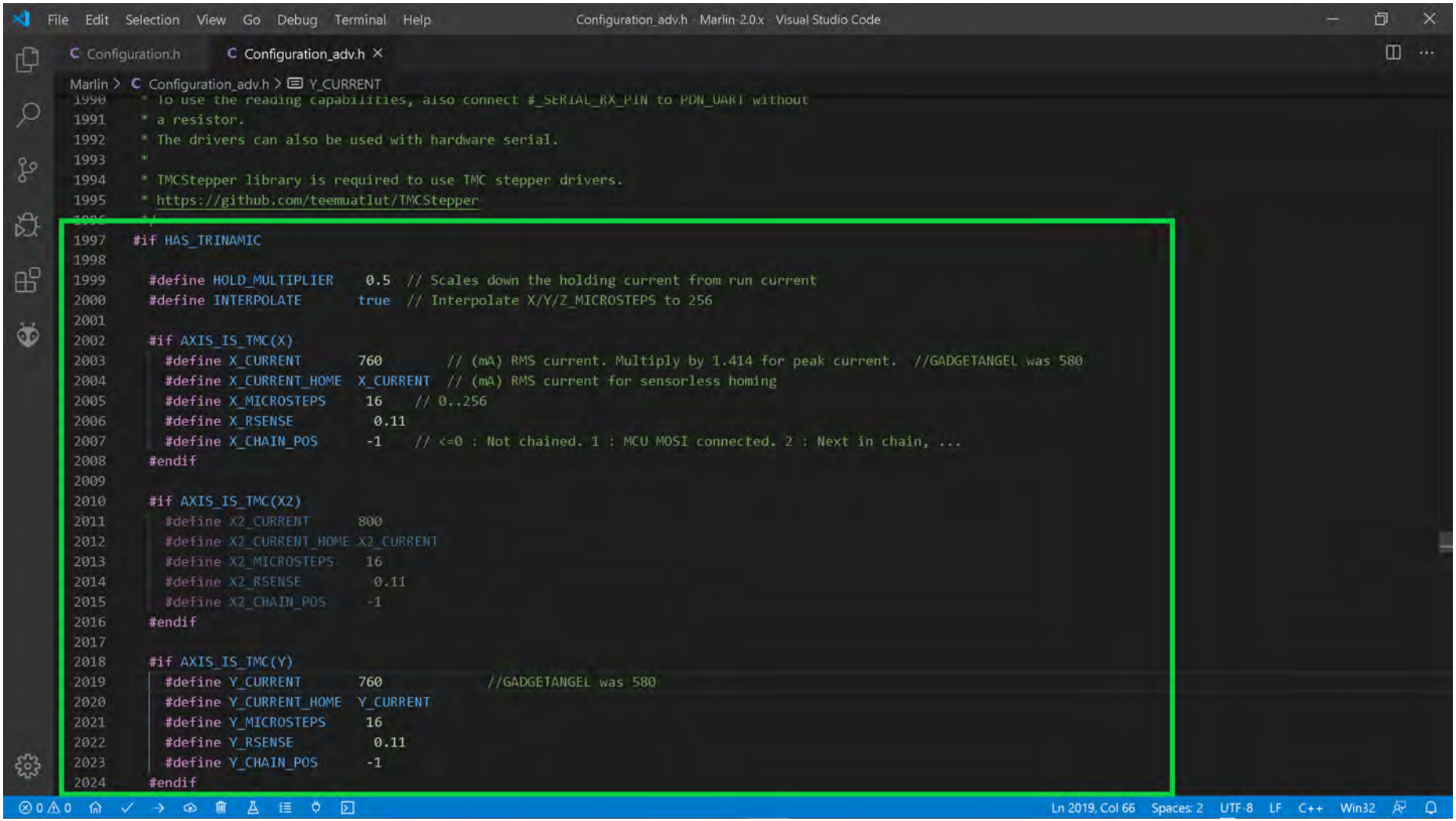
- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 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

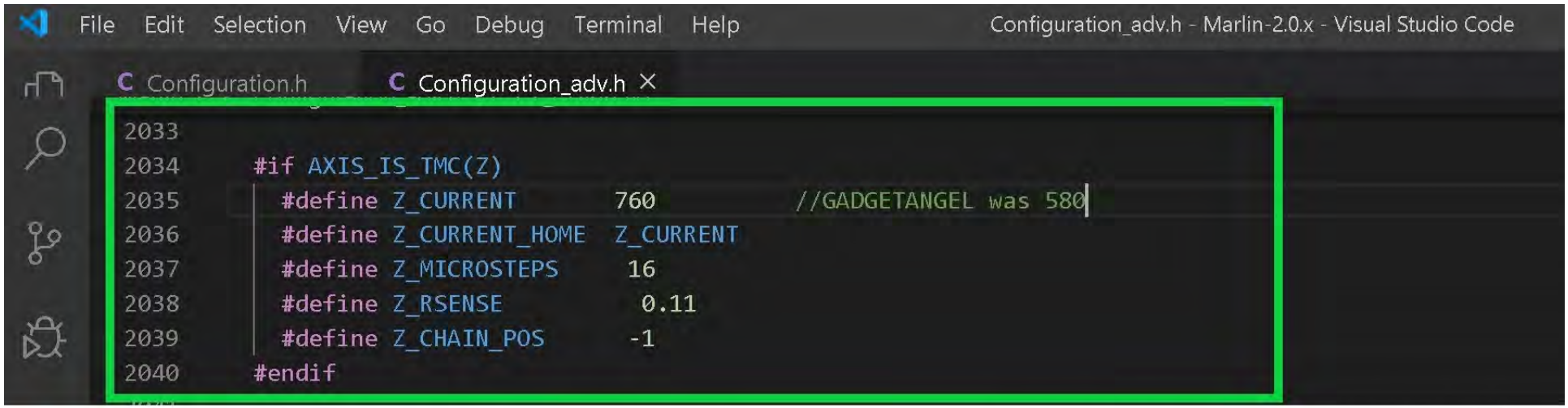
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2209 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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".

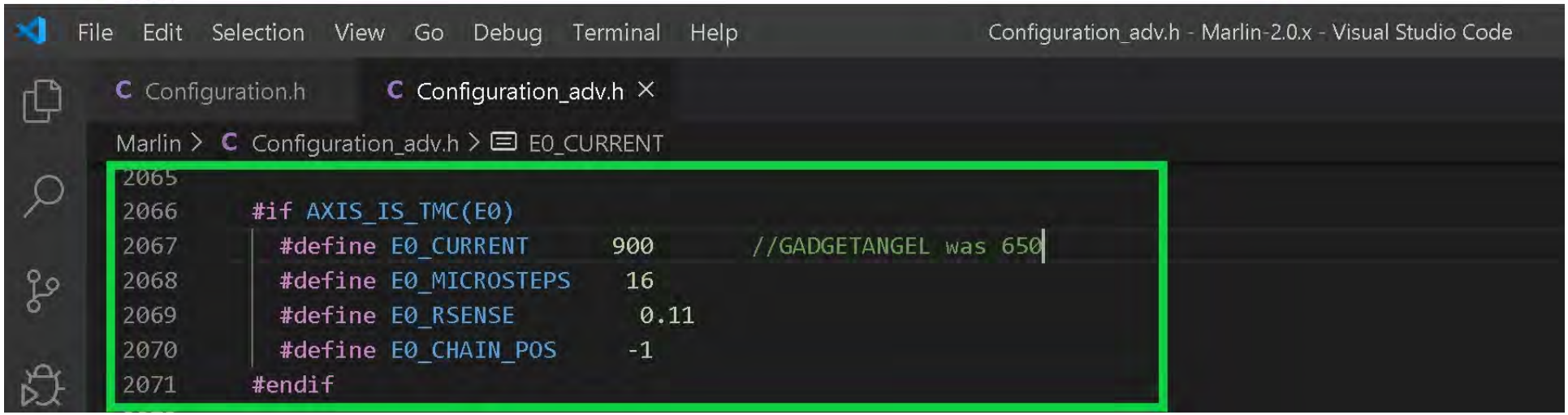


- Go to the next page.

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".

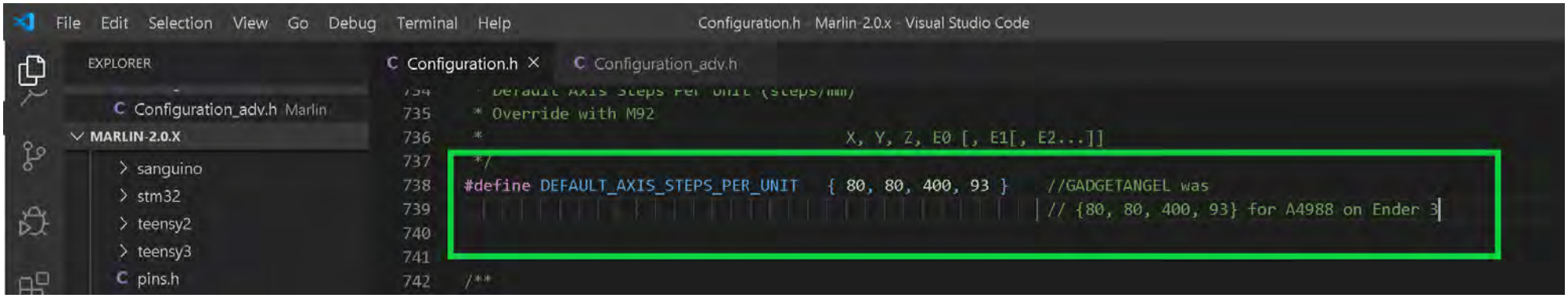




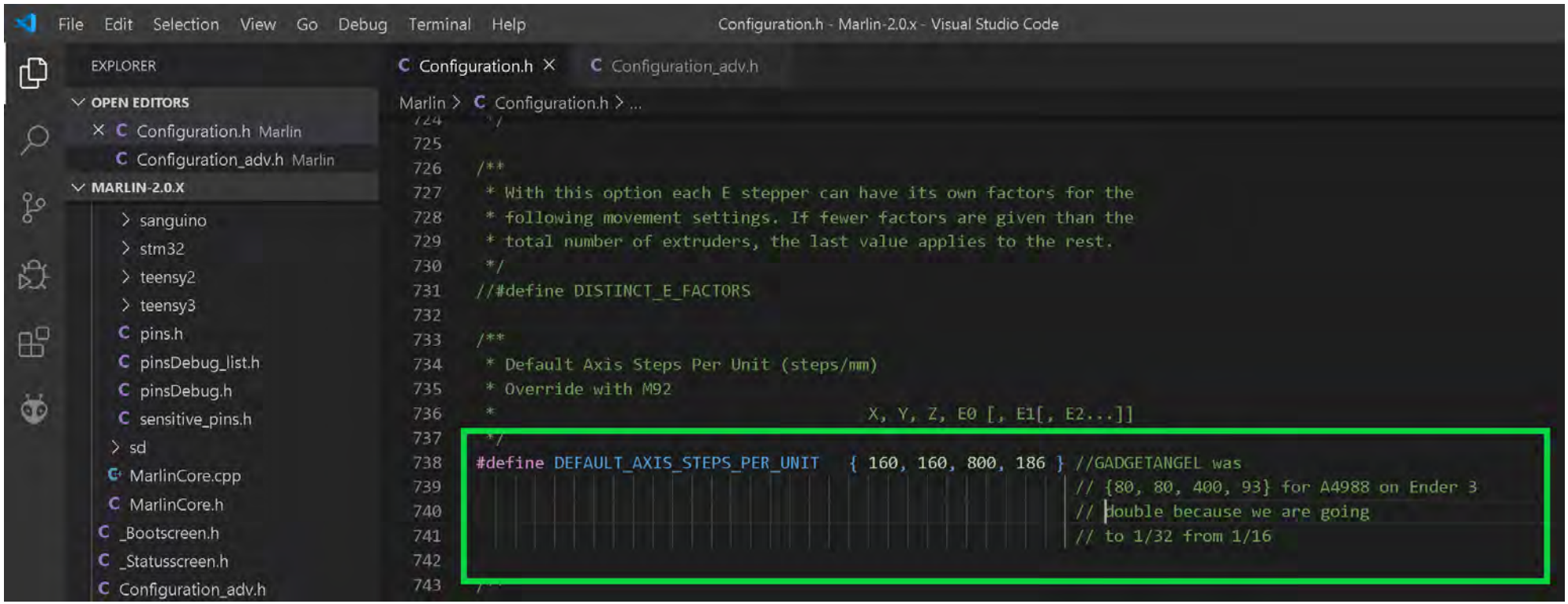
- Go to the next page.

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



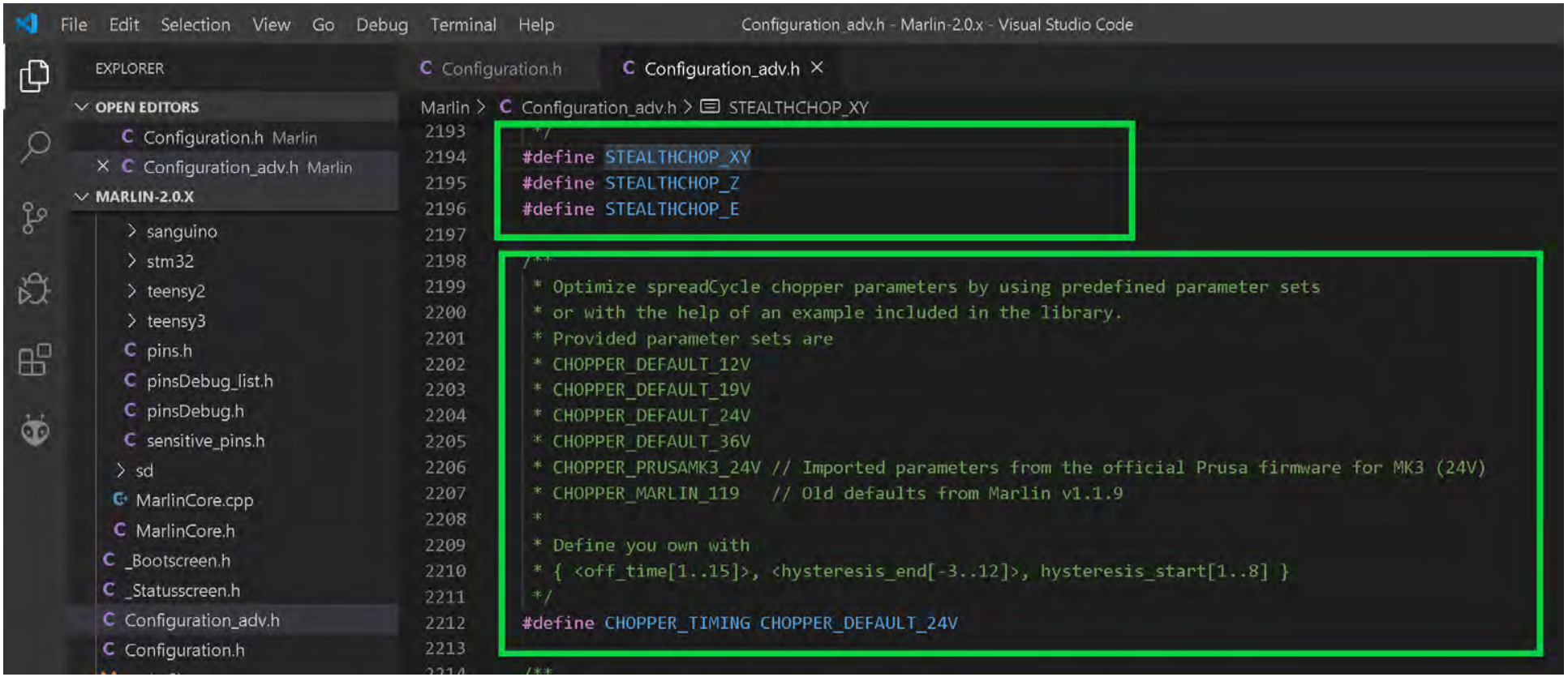
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



- Go to the next page.

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

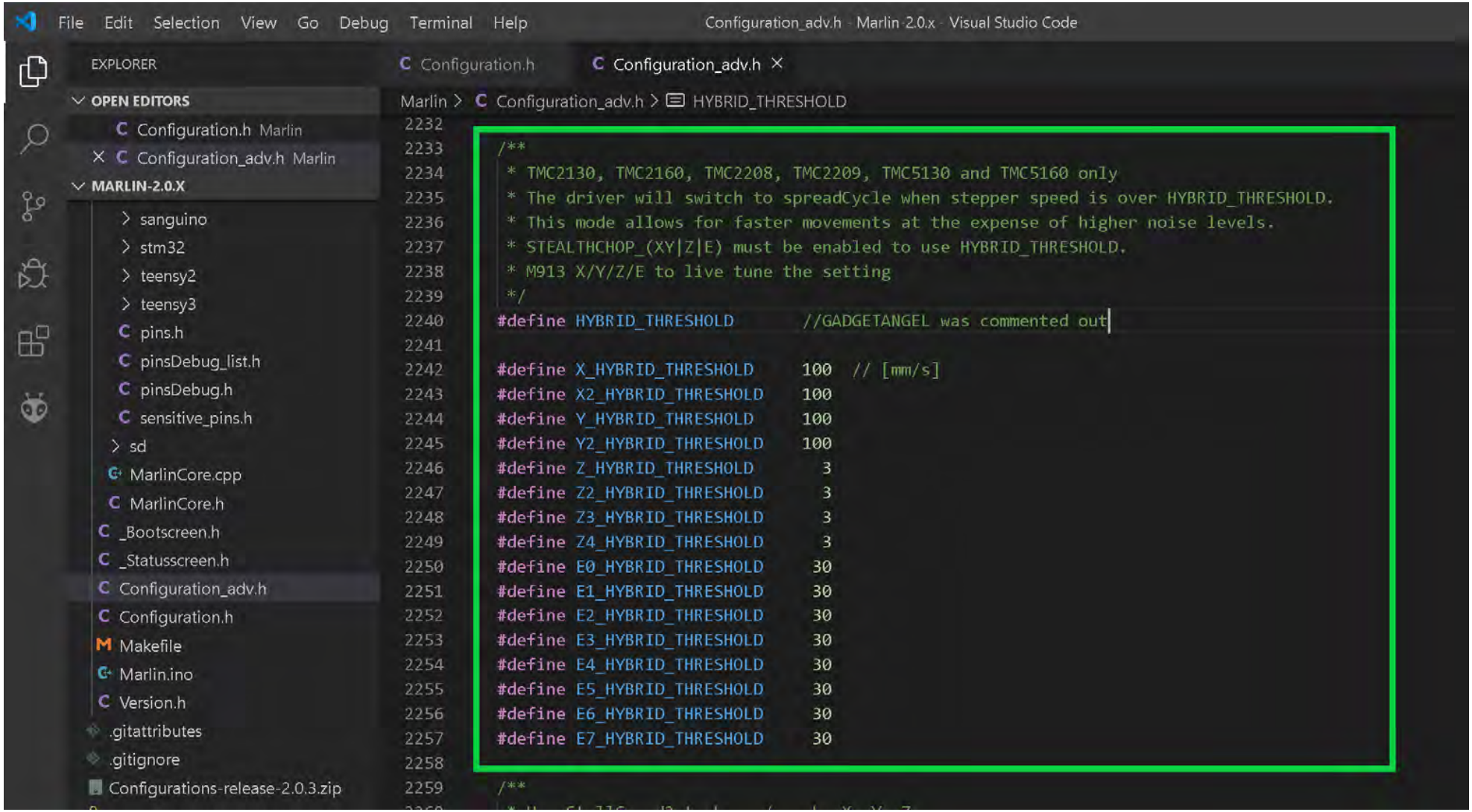
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if 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

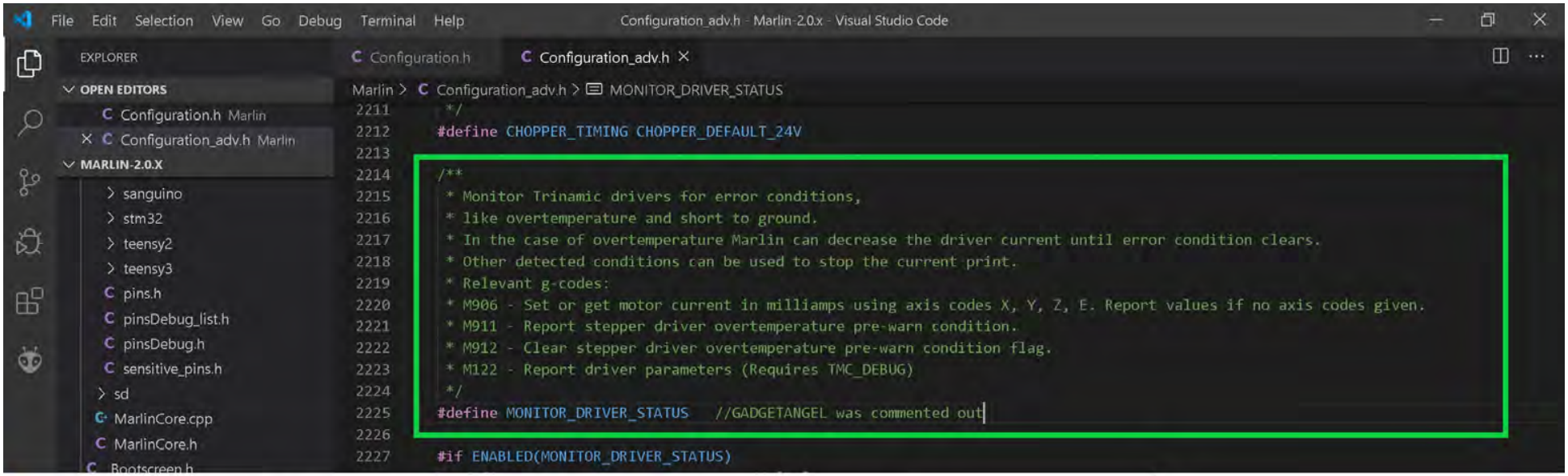
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

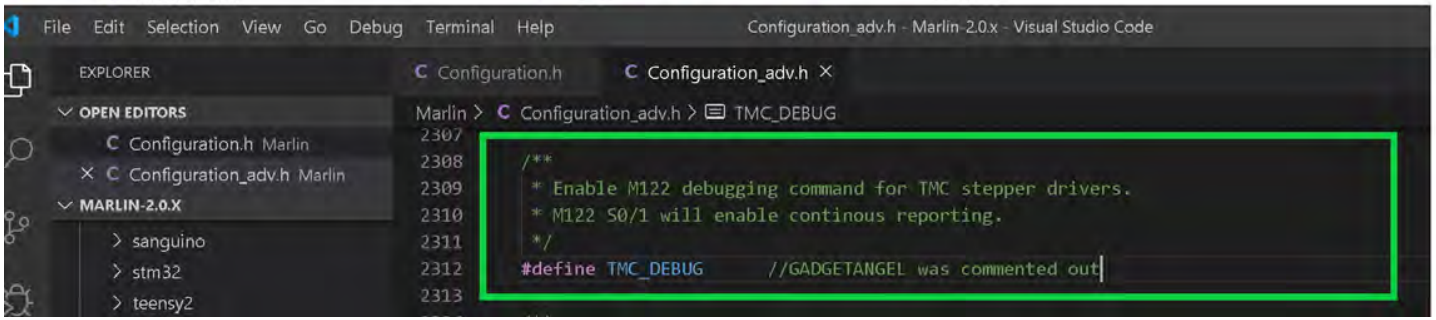


- Go to the next page.

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.**

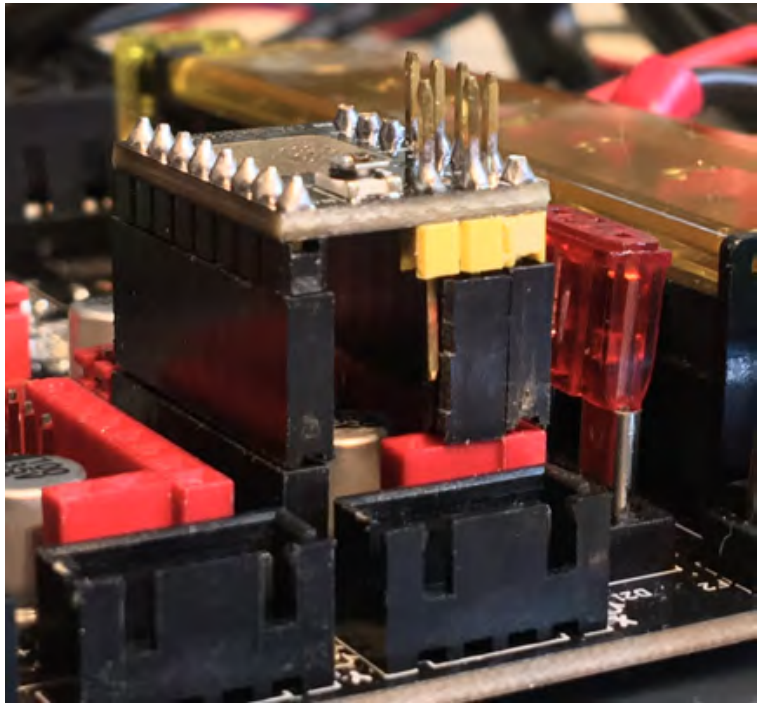




- Go to the next page.

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 DIAG pin on the TMC2209 driver has to be plugged into the SKR PRO V1.1 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG pin on the Z axis' TMC2209 is NOT connected to the board. I plan to plug my Z axis TMC2209 into my SKR PRO V1.1 board by using long stackable header pin risers, as seen in the picture below.

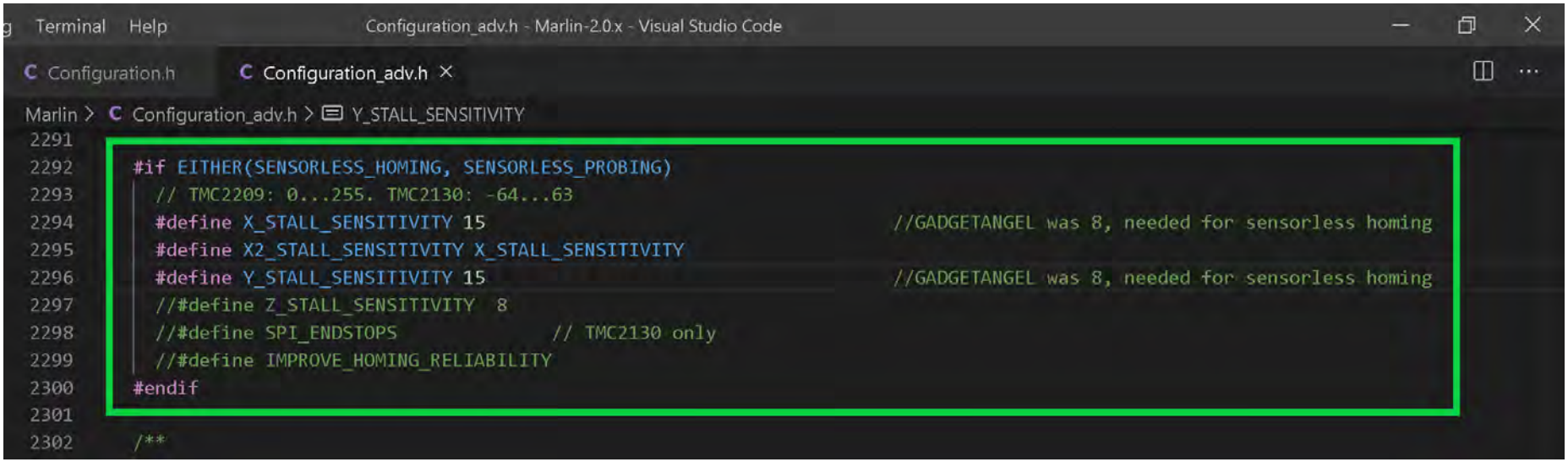


- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

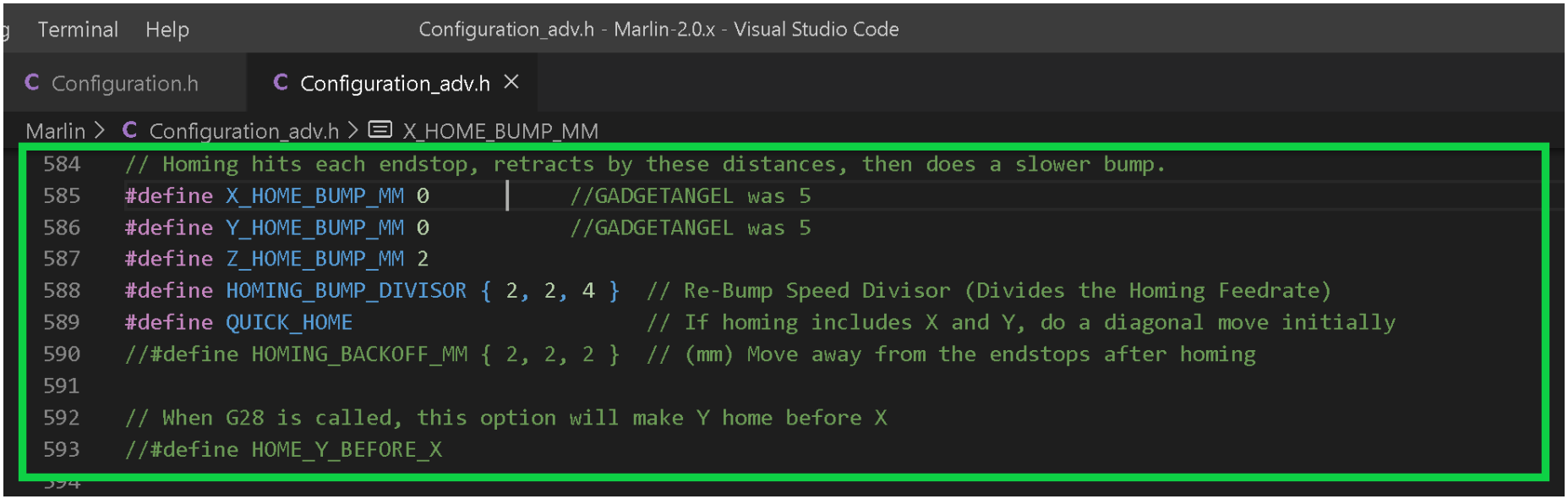
- Go to the next page.

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".



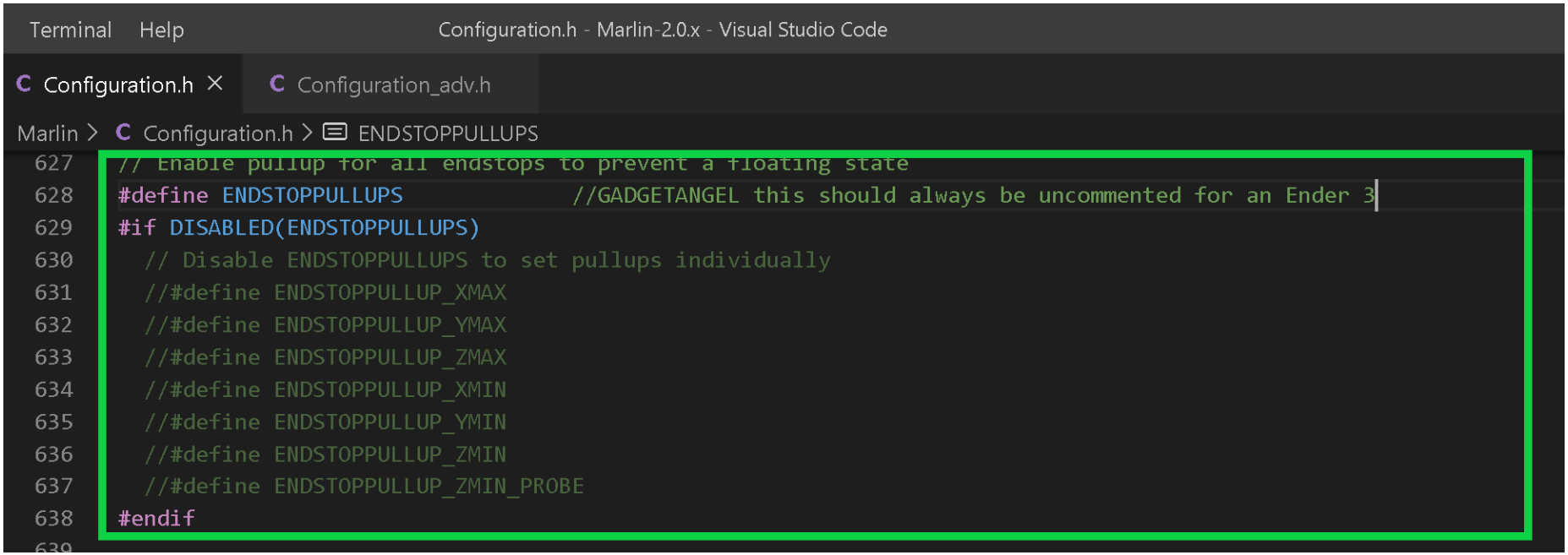
- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.

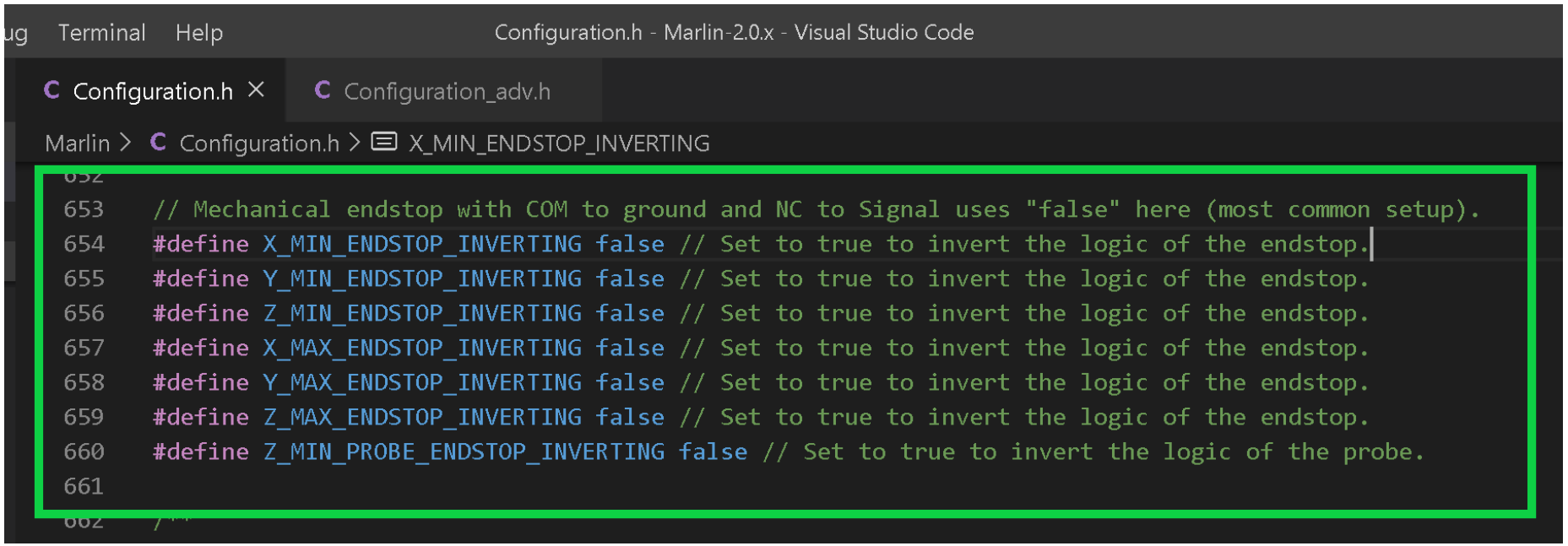


- Go to the next page.

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.

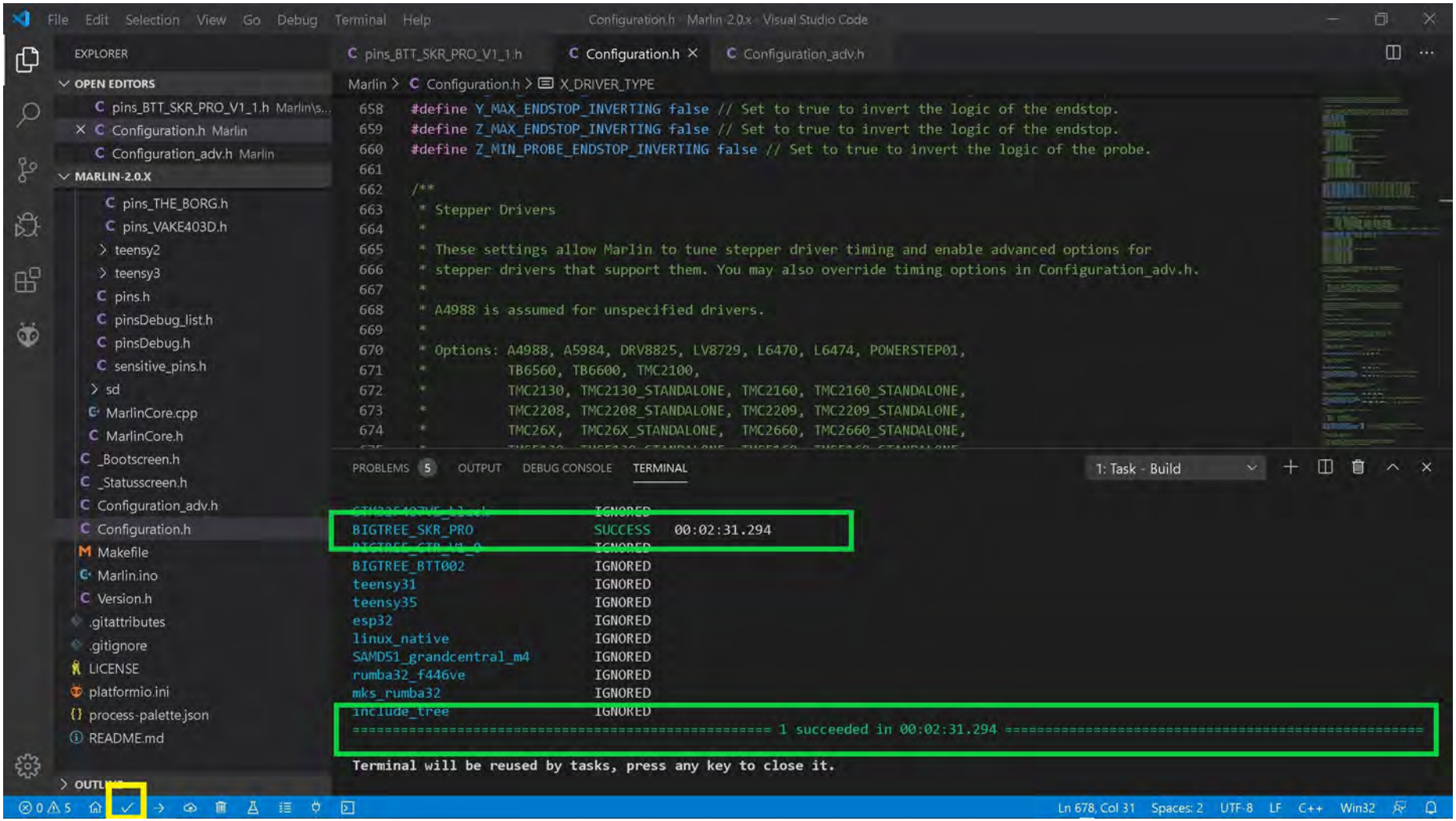




- Go to the next page.

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.



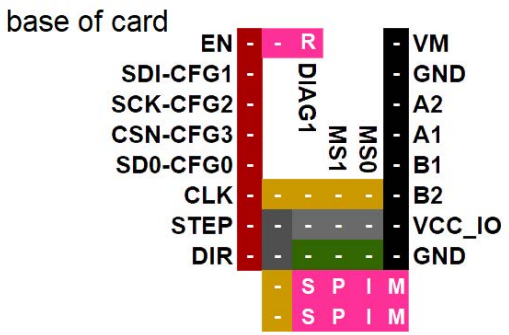
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

BIQU TMC5160 V1.2

SPI Mode

R - Stall Detection
The DIAG1 PIN must be soldered onto the driver board

SPI Mode



See Appendix D for legend

STATUS:
Not Yet Tested

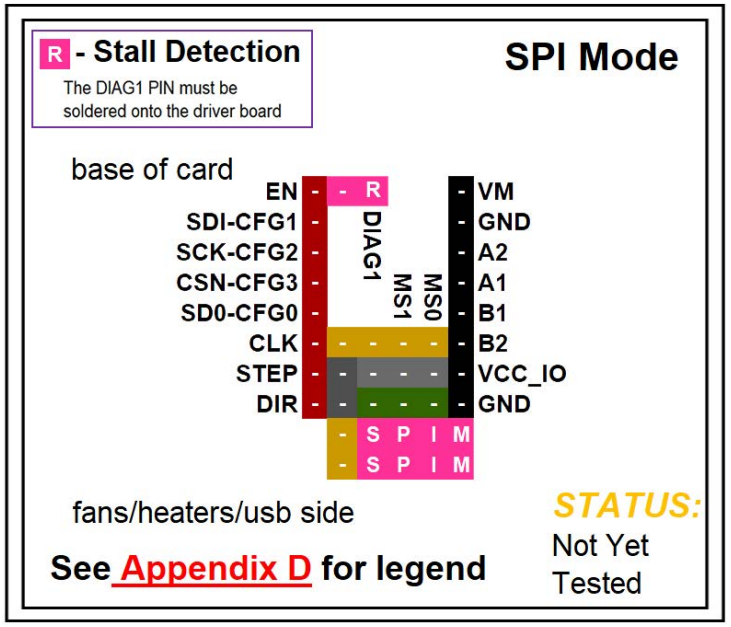
Note: You can use 50% to 90% of the calculated I_{RMS} (I_{MAX}/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

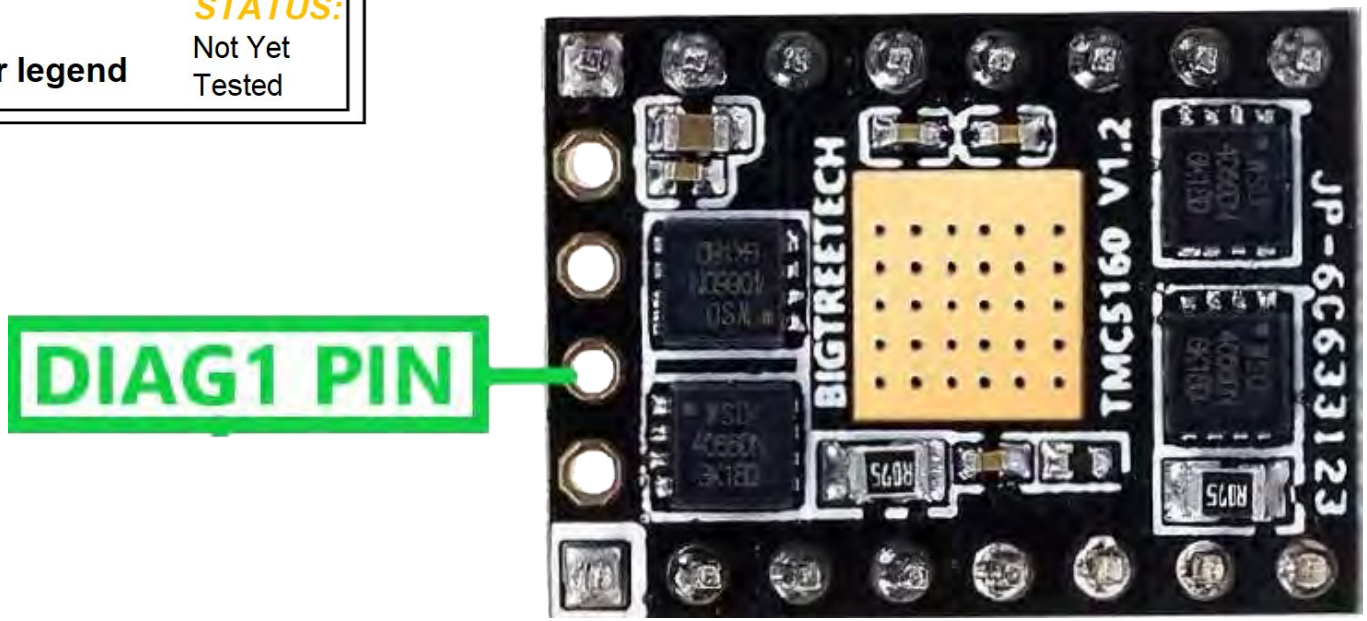
Driver Chip	Steps are set inside of your Firmware	
BIQU® TMC5160 SPI Mode Maximum 256 Subdivision 40V DC 4.3A (peak)		
Driving Current Calculation Formula R_S(Typical Sense Resistor)=0.075 Ω	**I_{MAX}=4.333** See Appendix B #8. Use 50% to 90% as shown below: **I_{MAX}=I_{MAX} * 0.90=3.900**	Current Limit 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 because 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.

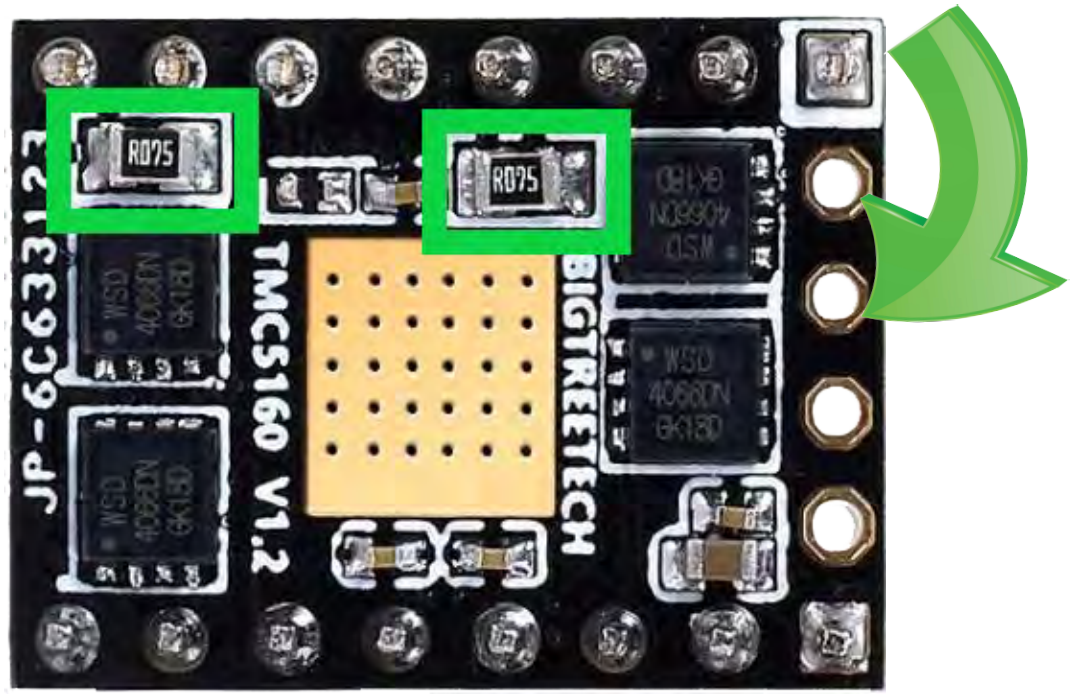
SPI Mode

BIQU TMC5160 V1.2

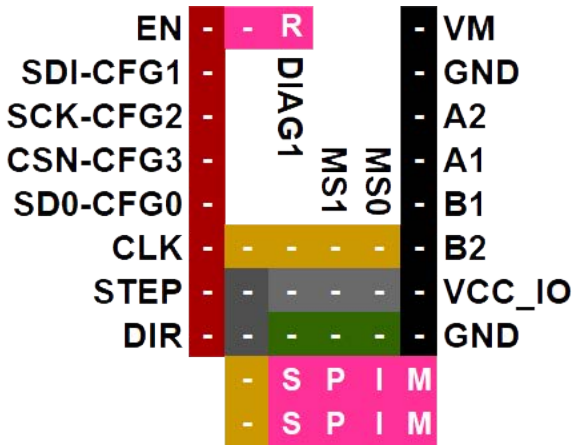
SPI Mode

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" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

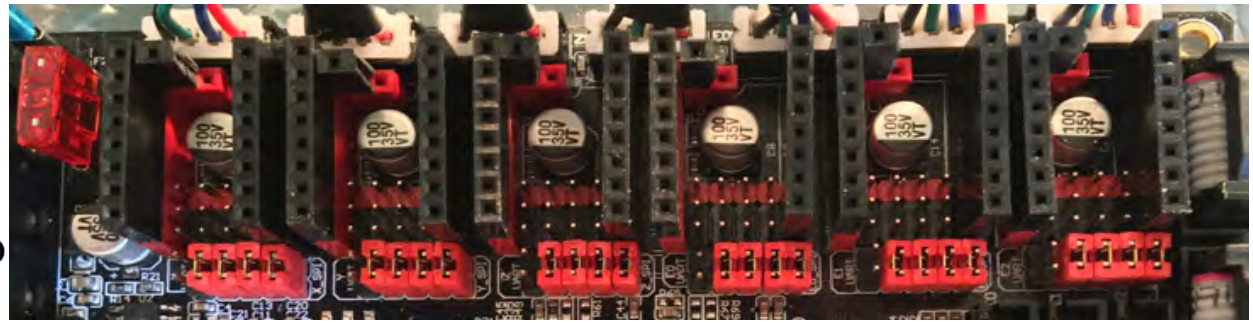
R_s = R050 is 0.05 Ohms
R_s = R062 is 0.062 Ohms
R_s = R068 is 0.068 Ohms
R_s = R075 is 0.075 Ohms
R_s = R100 is 0.1 Ohms
R_s = R110 is 0.11 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms



SPI



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

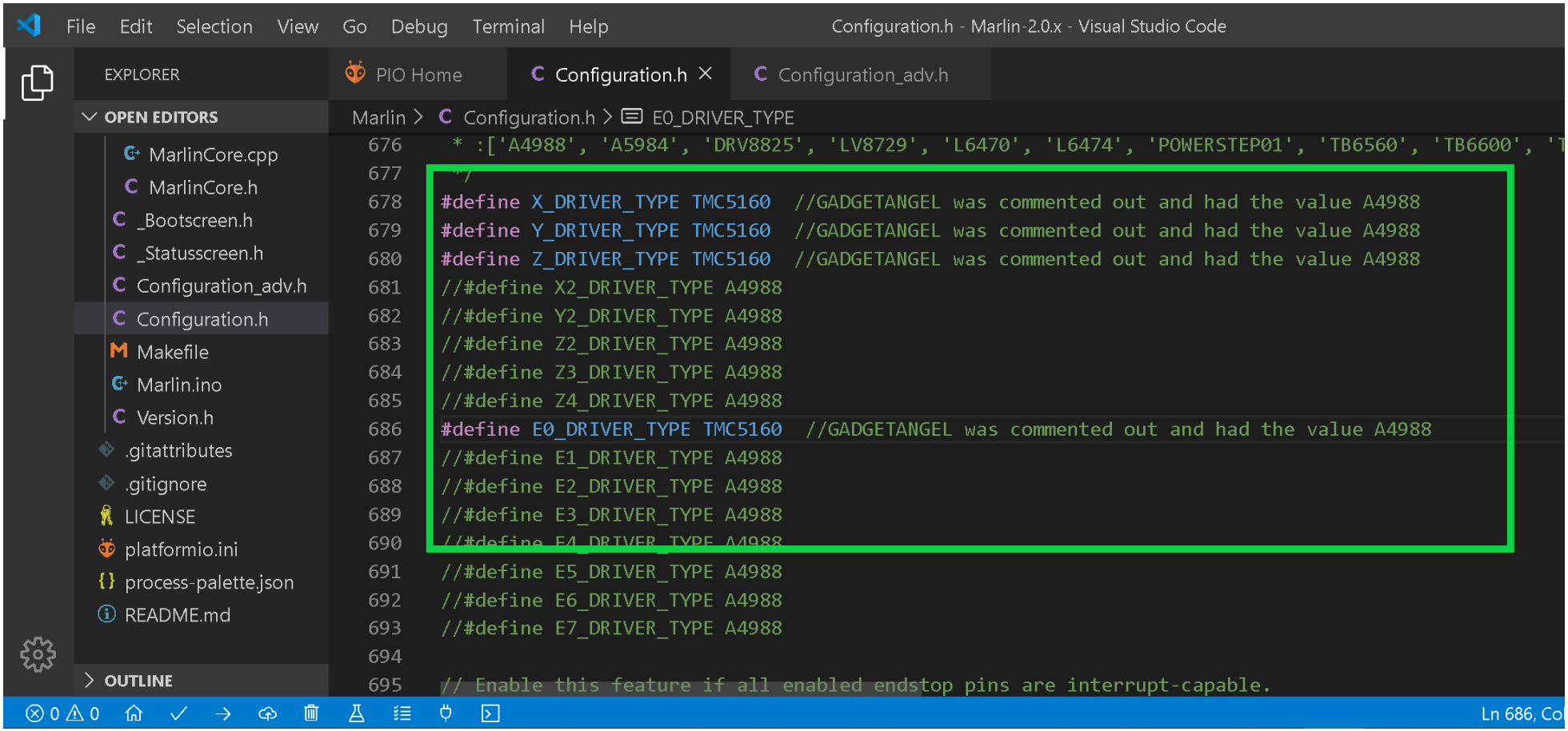


See Appendix D for legend

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.

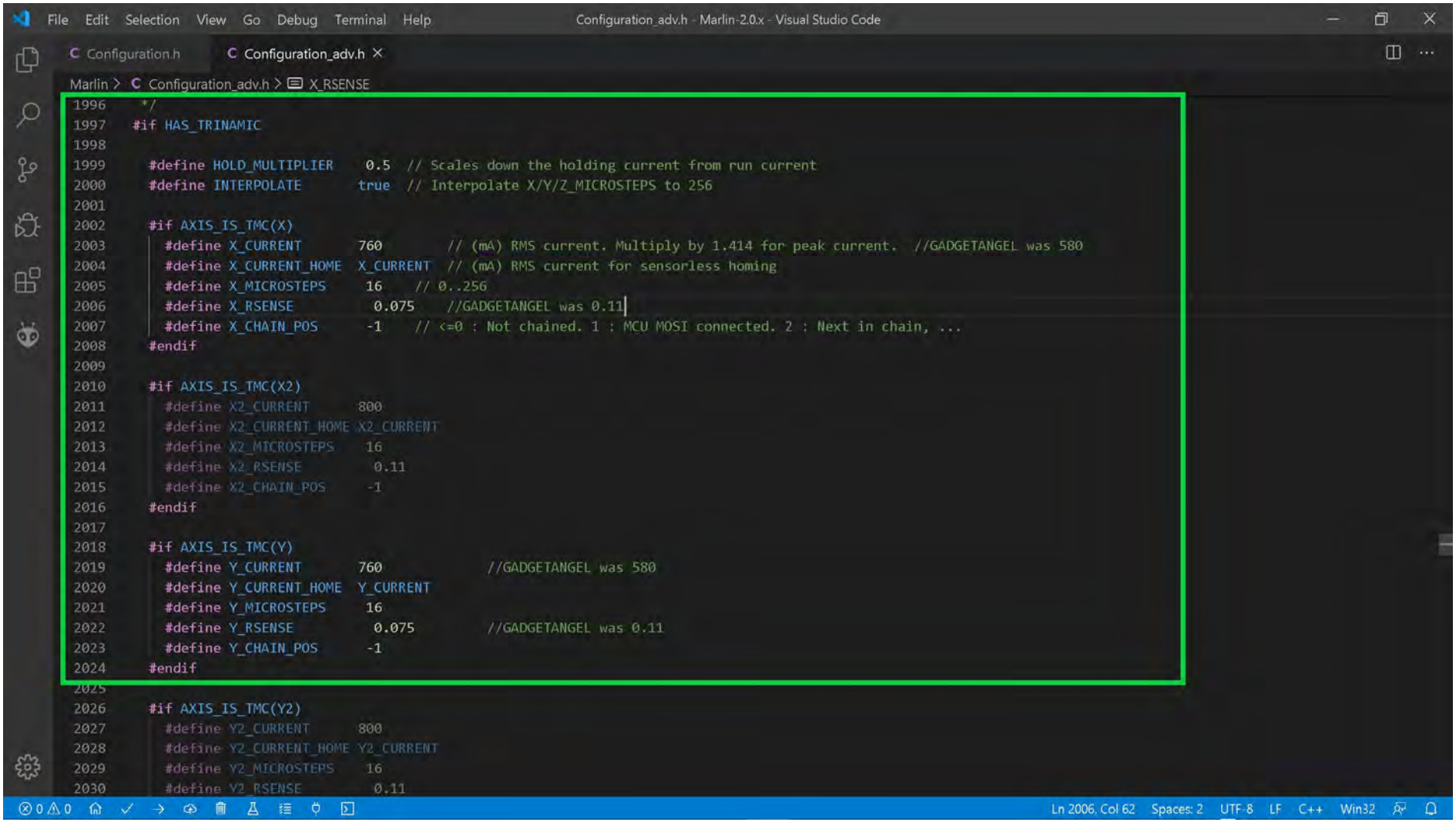
- Change the stepper motor drivers so that Marlin knows you are using TMC5160 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5160 drivers in SPI mode. When two "//" appear at the beginning of a line 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

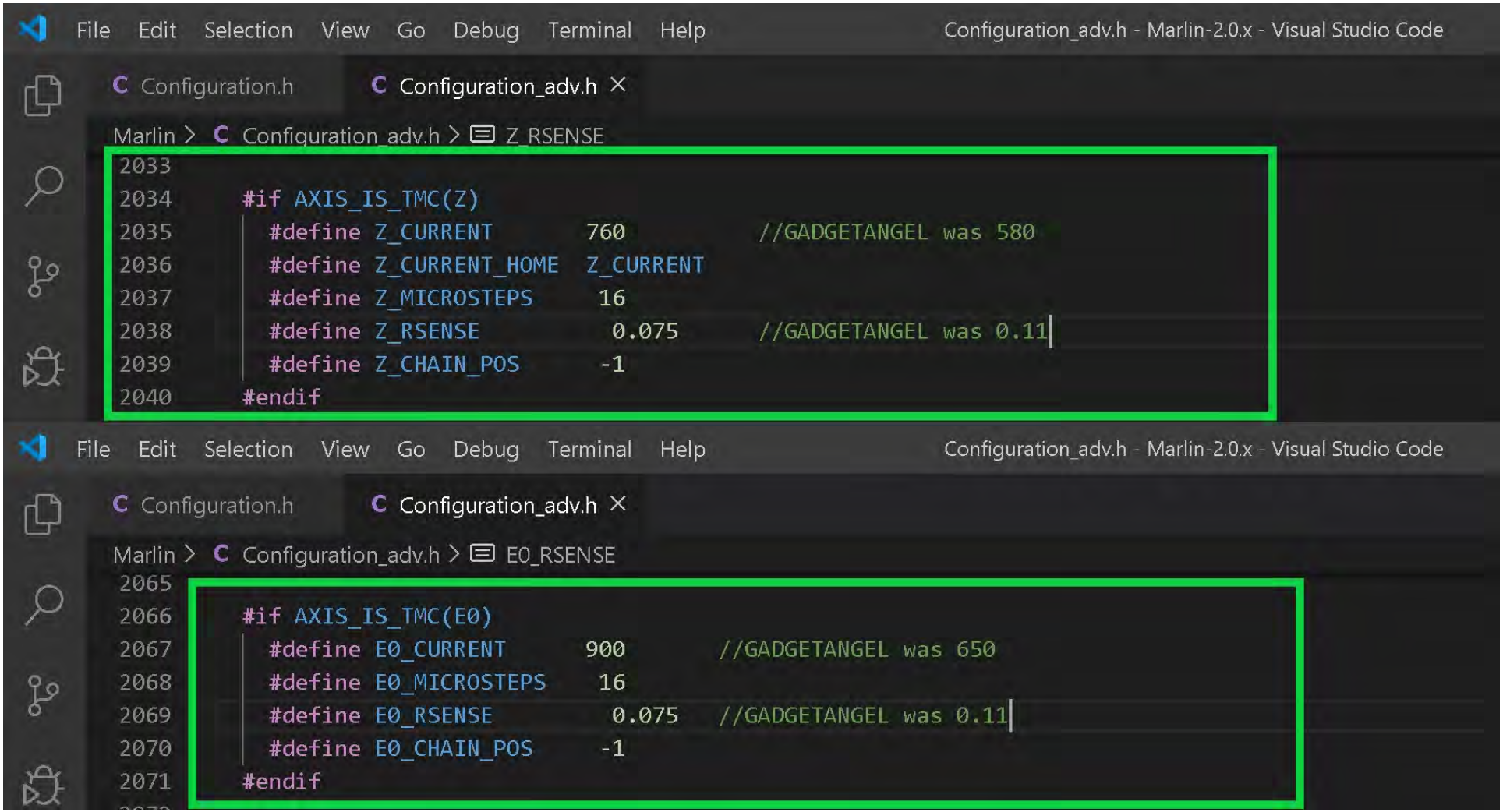
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5160 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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

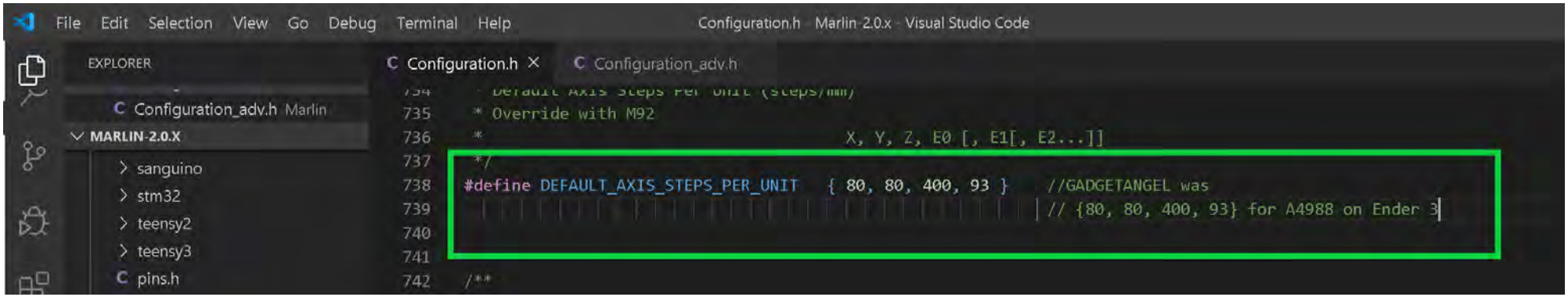
- Now, I am setting the V_{ref} for Z-Axis and the extruder, as seen in the **GREEN** boxes below. I changed the "Z_CURRENT" to be the calculated V_{ref} for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0_CURRENT" to be the calculated V_{ref} for my Extruder, which is 900mV on the Ender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



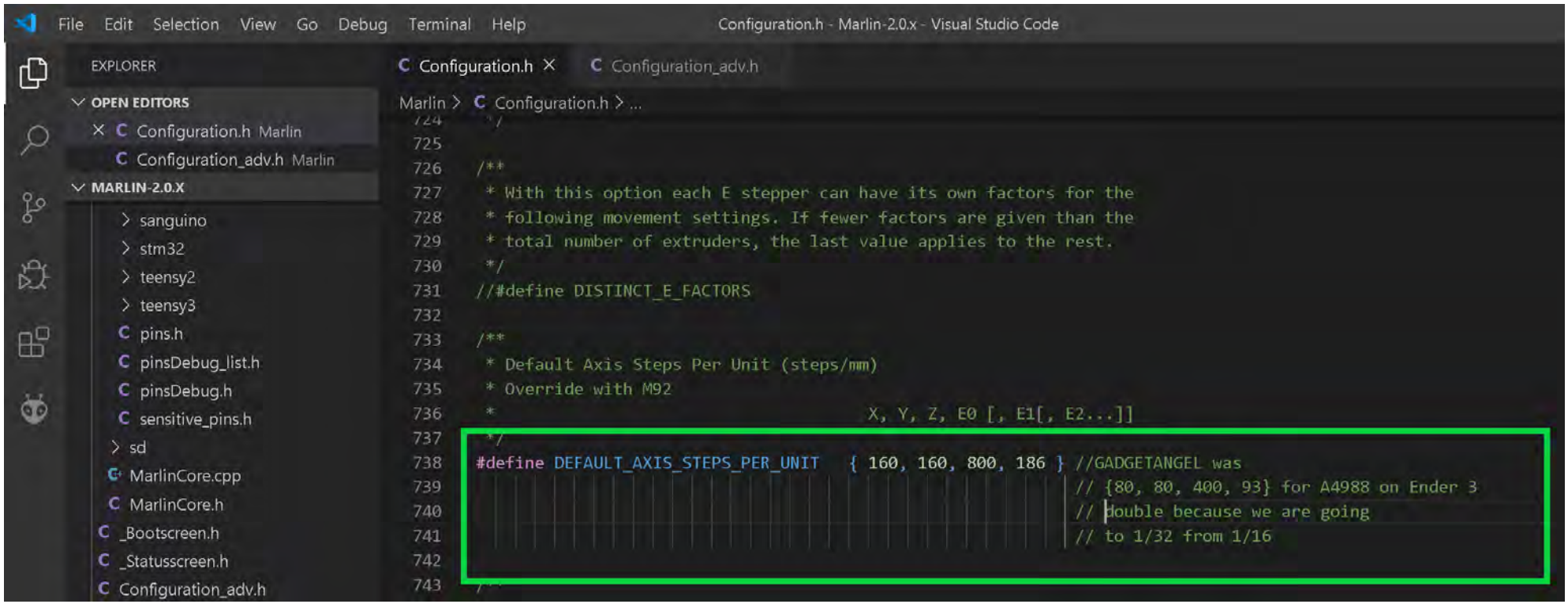
- Go to the next page.

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



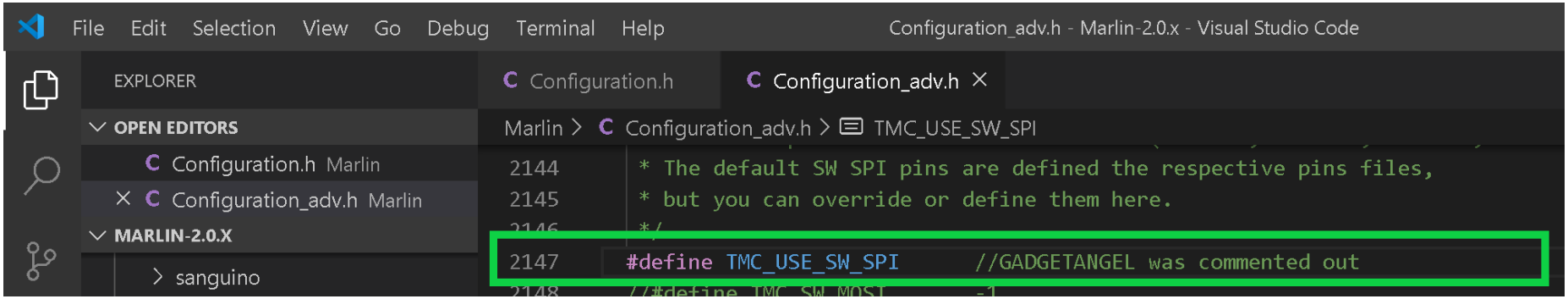
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



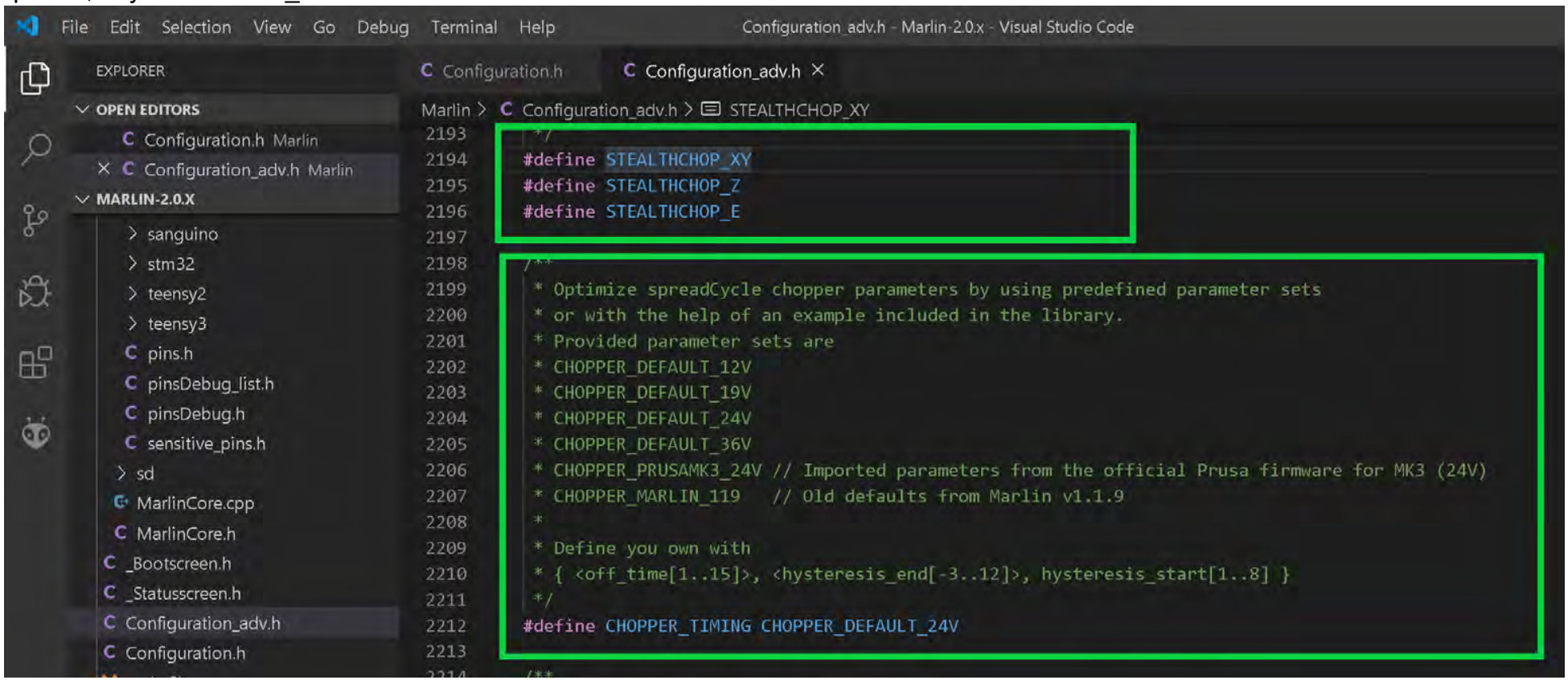
- Go to the next page.

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 PRO V1.1 pins file depends on this variable to define its SPI pins



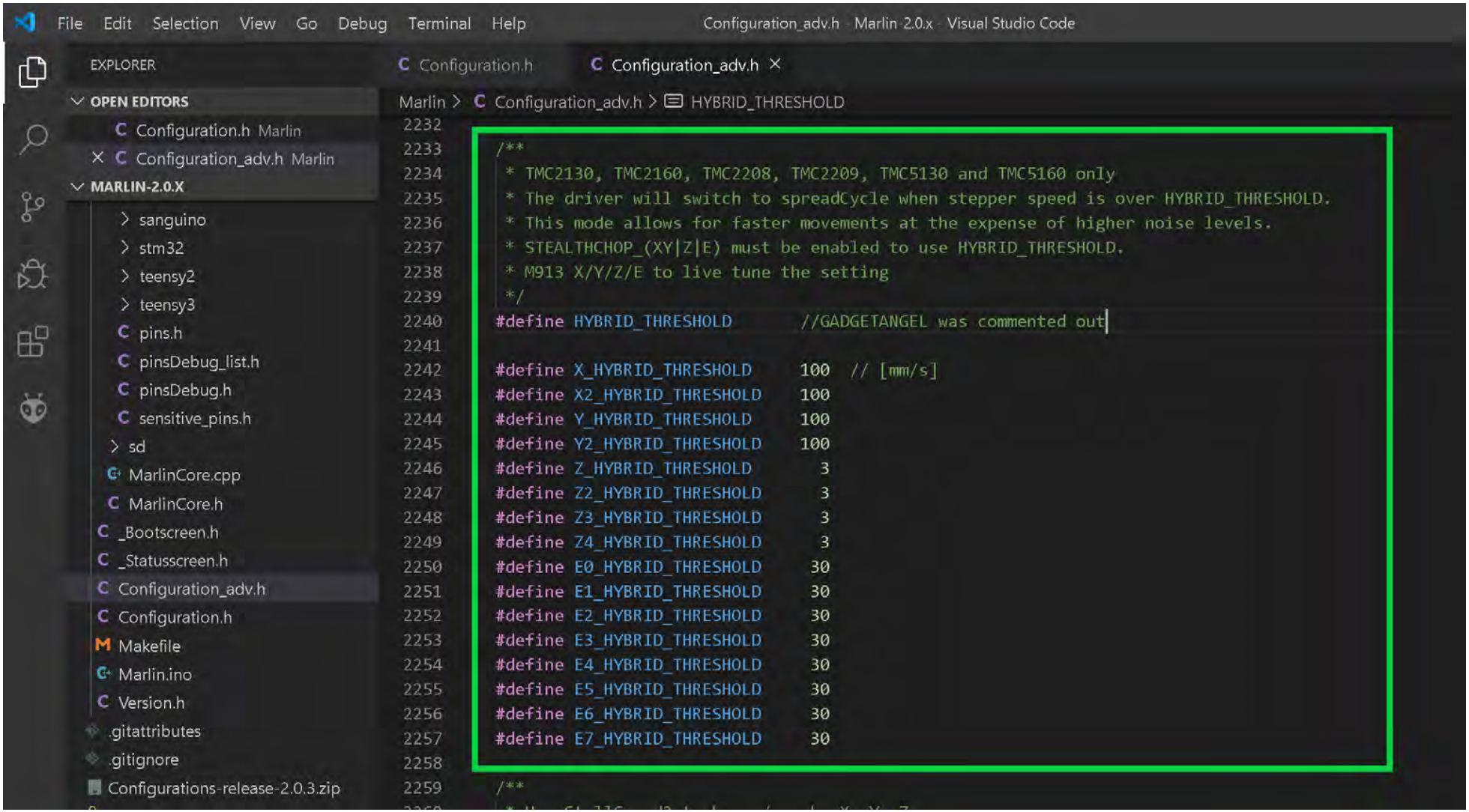
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if 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

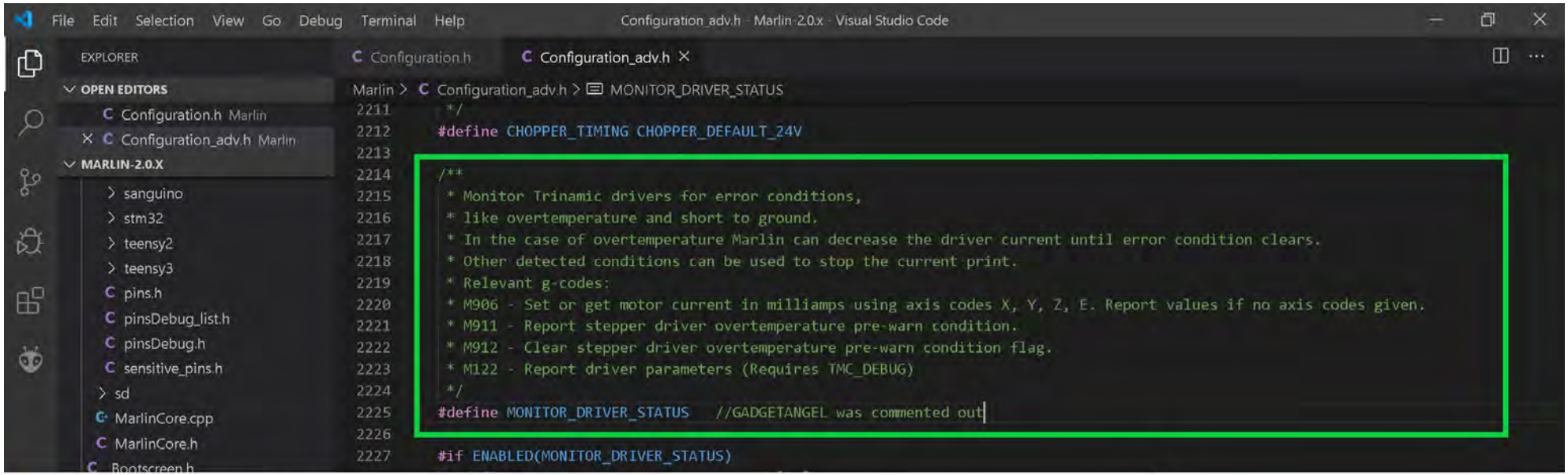
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

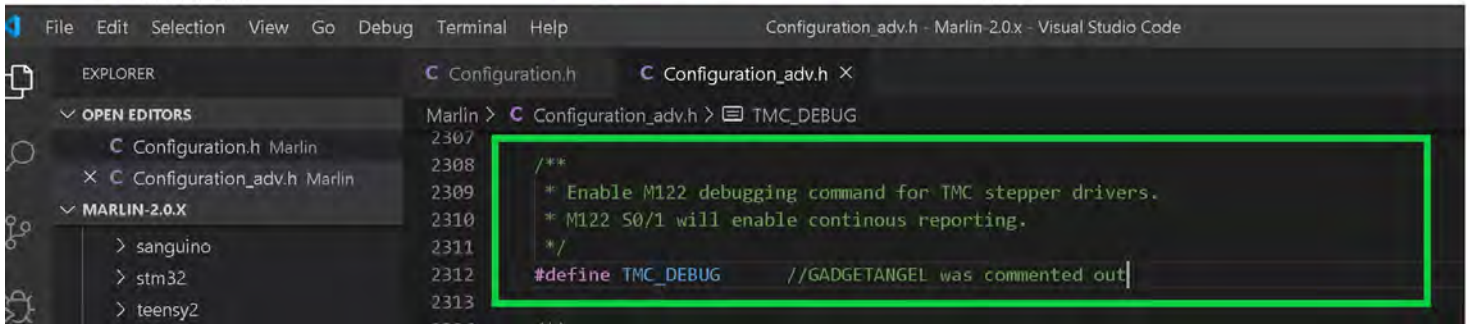


- Go to the next page.

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.**

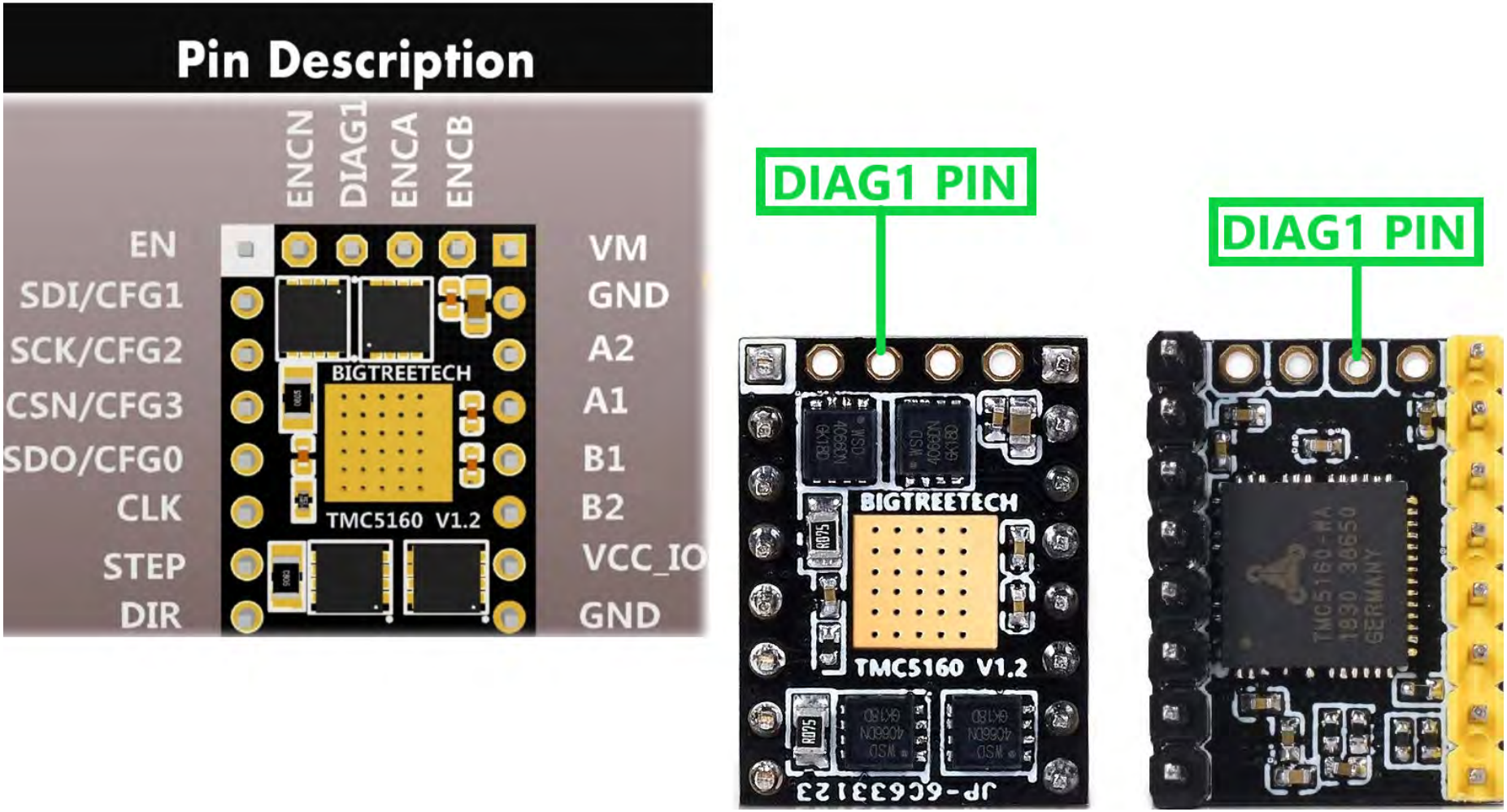




- Go to the next page.

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.

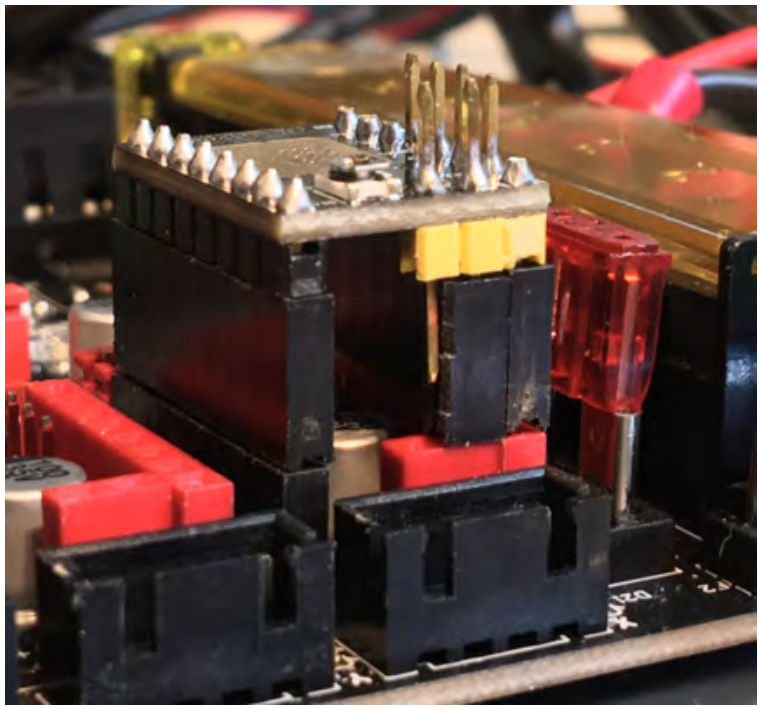


- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5160 into the SKR PRO V1.1 board the DIAG1 pin will be seated in the SKR PRO V1.1 board.

- Go to the next page.

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

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

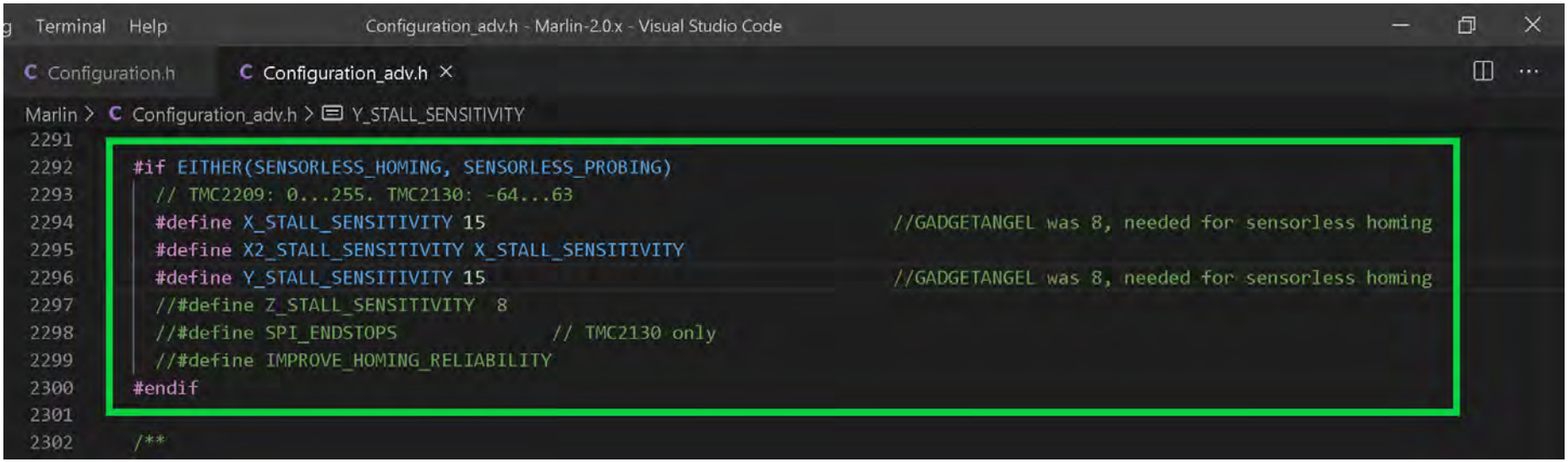


- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"

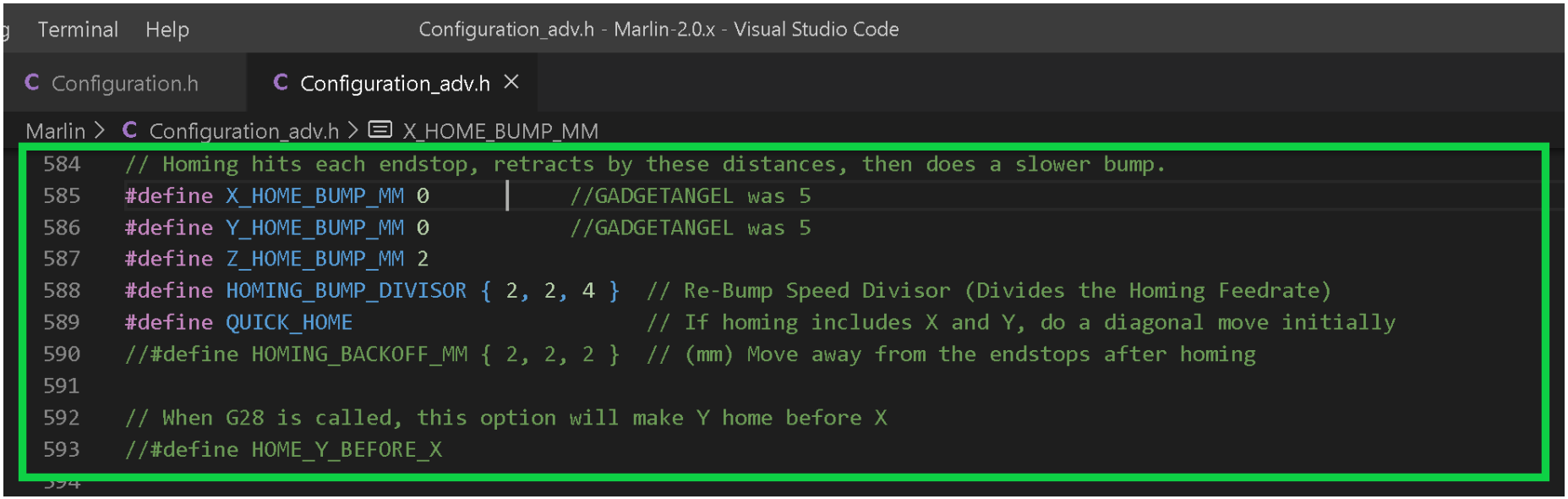
- Go to the next page.

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".



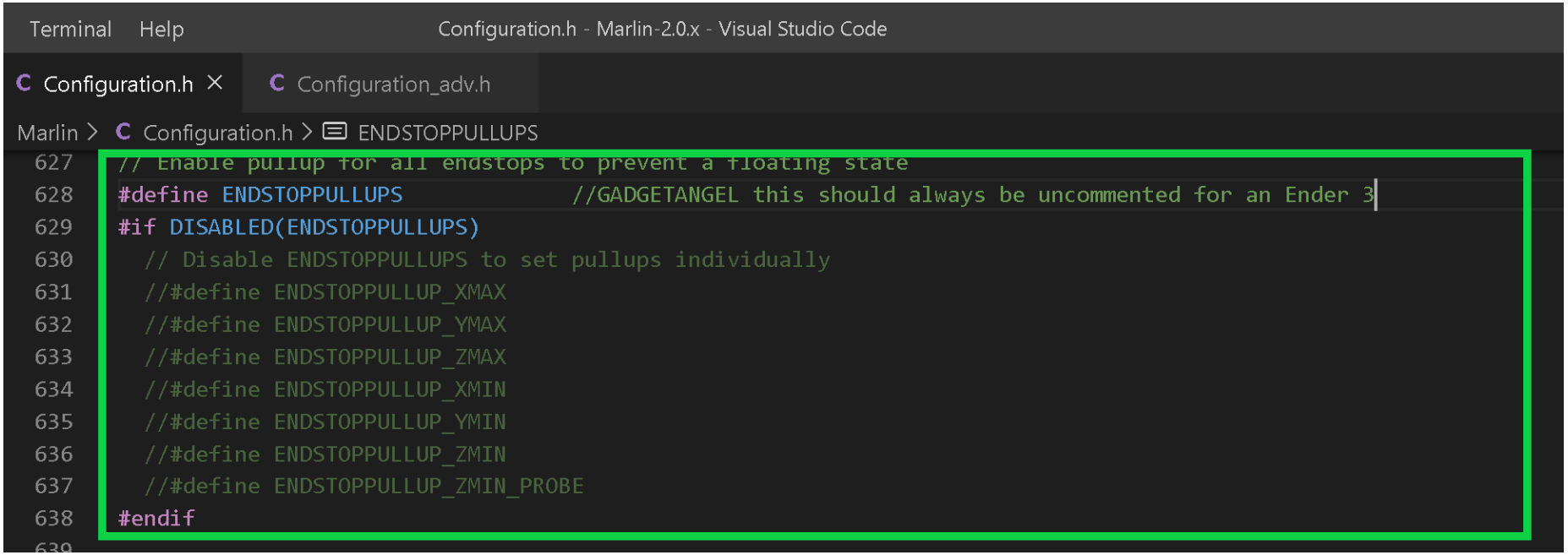
- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.



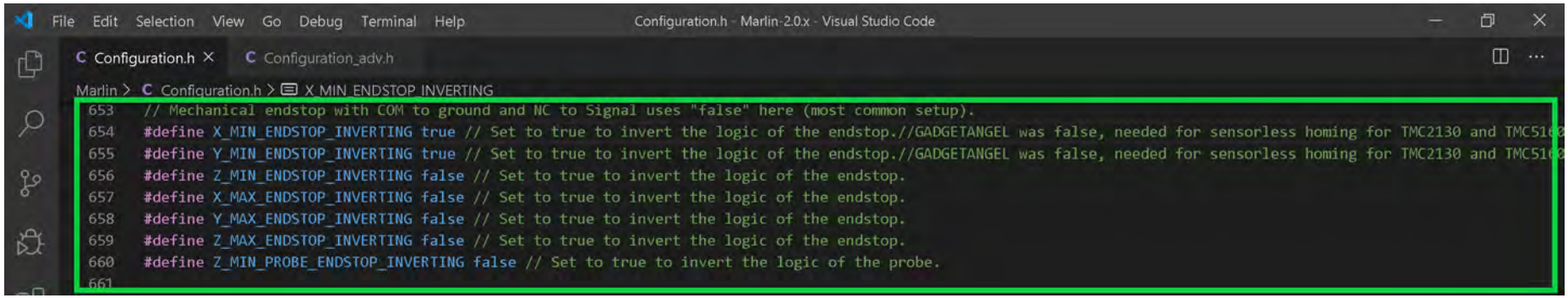
- Go to the next page.

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

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



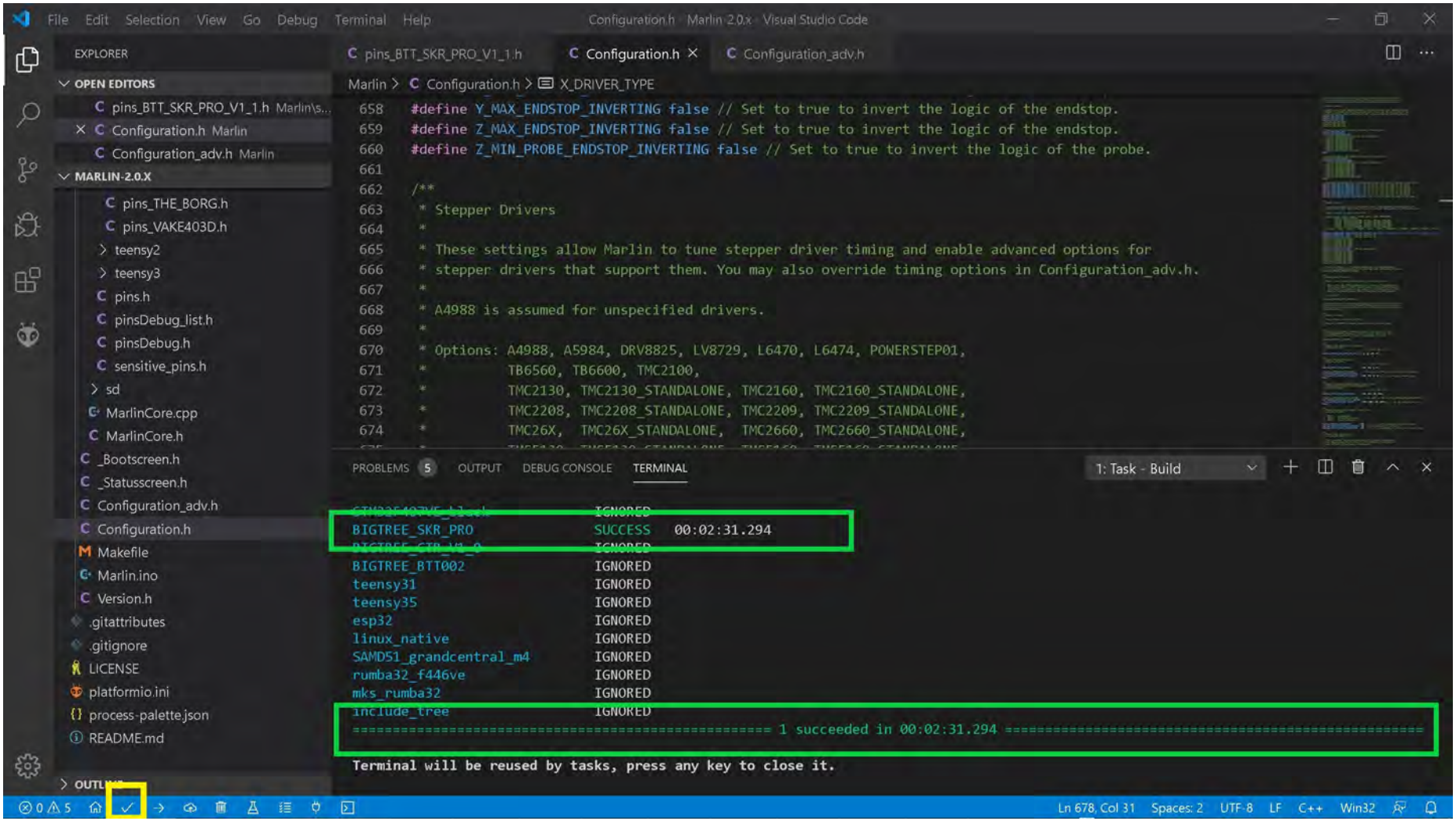
- Next to allow sensor-less homing to work (while using the BIQU TMC5160) we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true.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



- 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.



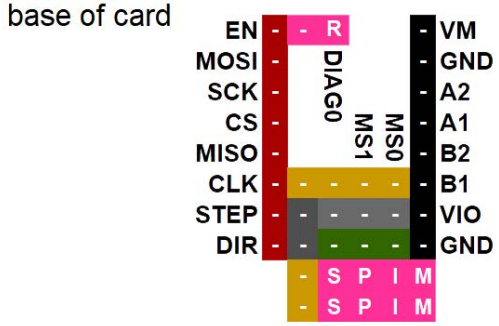
- For the SKR PRO V1.1 board you must save the "firmware.bin" file to the micro SD card then place the micro SD card into the micro SD card reader of the SKR PRO V1.1 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR PRO MCU. To locate the "firmware.bin"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

BIQU TMC5161 V1.0

SPI Mode

R - Stall Detection
The DIAG0 PIN must be soldered onto the driver board

SPI Mode



See Appendix D for legend

STATUS: Not Yet Tested

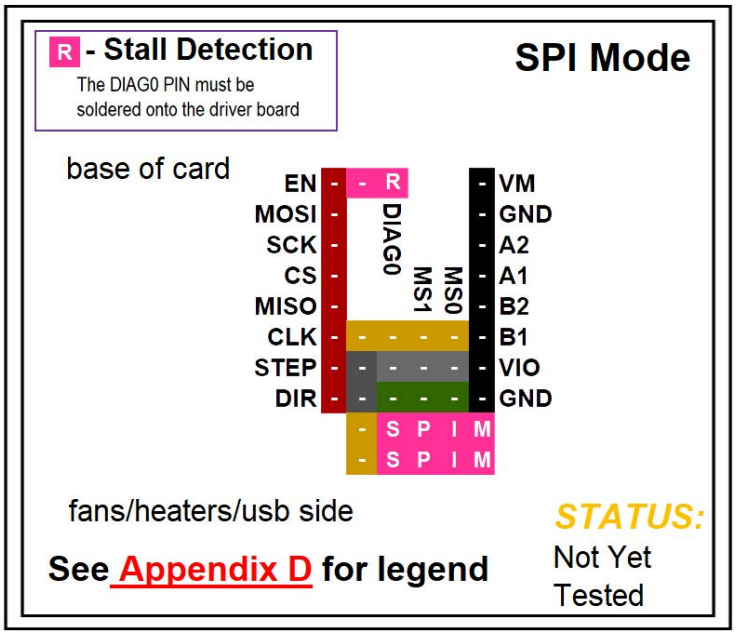
Note: You can use 50% to 90% of the calculated I_{RMS} (I_{MAX}/1.414) when tuning ("X_CURRENT", "Y_CURRENT", etc. the stepper motor driver in the firmware.

See the next page for further information.

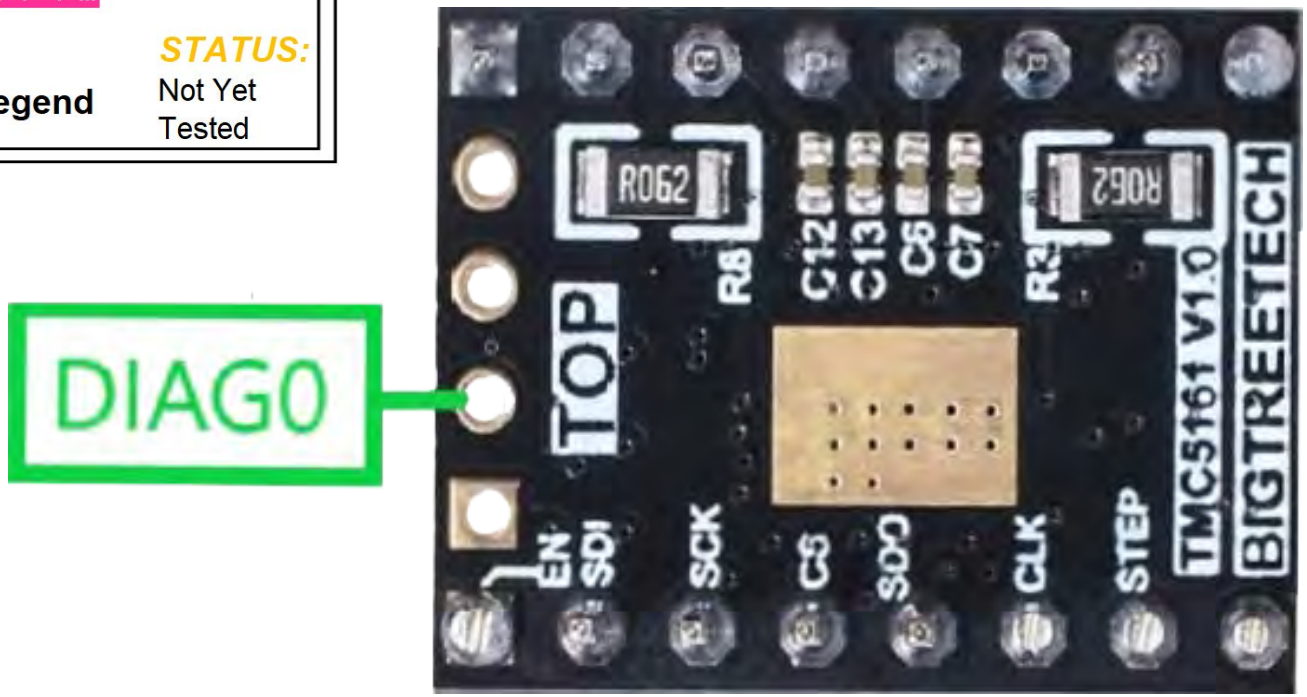
Driver Chip	Steps are set inside of your Firmware	
BIQU® TMC5161 SPI Mode Maximum 256 Subdivision 40V DC 5.2A (peak)		
Driving Current Calculation Formula R_S(Typical Sense Resistor)=0.062Ω	**I_{MAX}=5.2419** See Appendix B #9. Use 50% to 90% as shown below: **I_{MAX}=I_{MAX} * 0.90=4.718**	Current Limit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

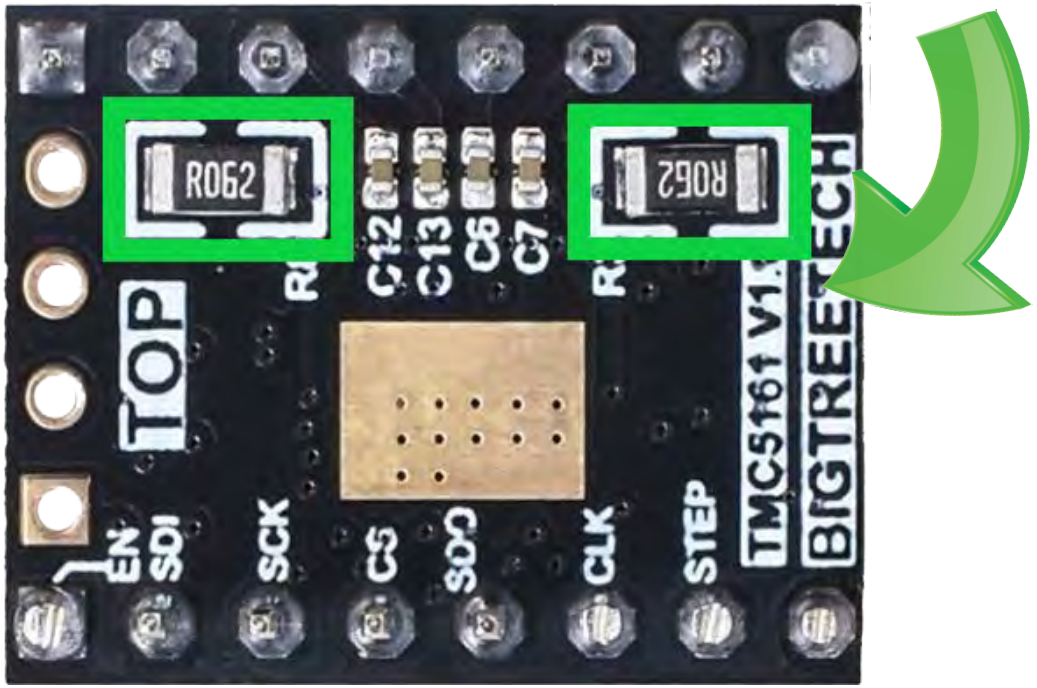
SPI Mode

BIQU TMC5161 V1.0

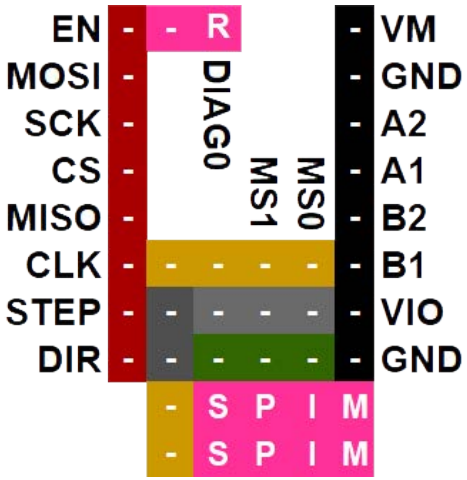
SPI Mode

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" and/or "E0_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT" and/or "E0_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", and/or "E0_CURRENT".

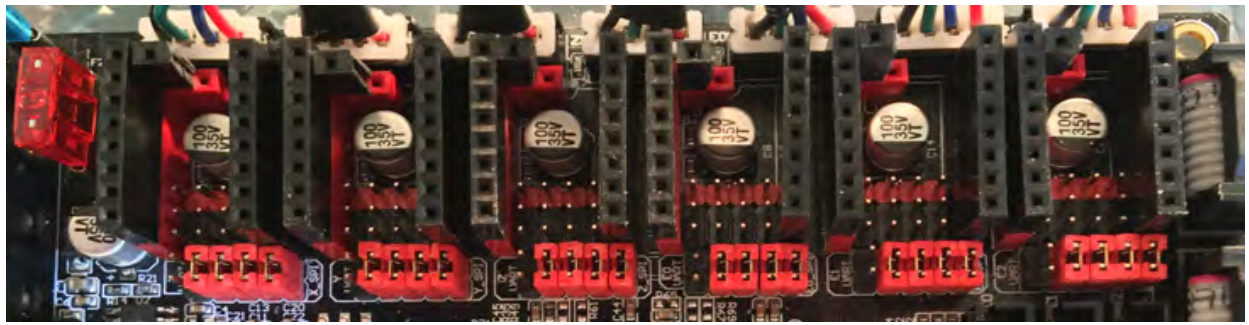
R_s = R050 is 0.05 Ohms
R_s = R062 is 0.062 Ohms
R_s = R068 is 0.068 Ohms
R_s = R075 is 0.075 Ohms
R_s = R100 is 0.1 Ohms
R_s = R110 is 0.11 Ohms
R_s = R150 is 0.15 Ohms
R_s = R200 is 0.2 Ohms
R_s = R220 is 0.22 Ohms



SPI



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

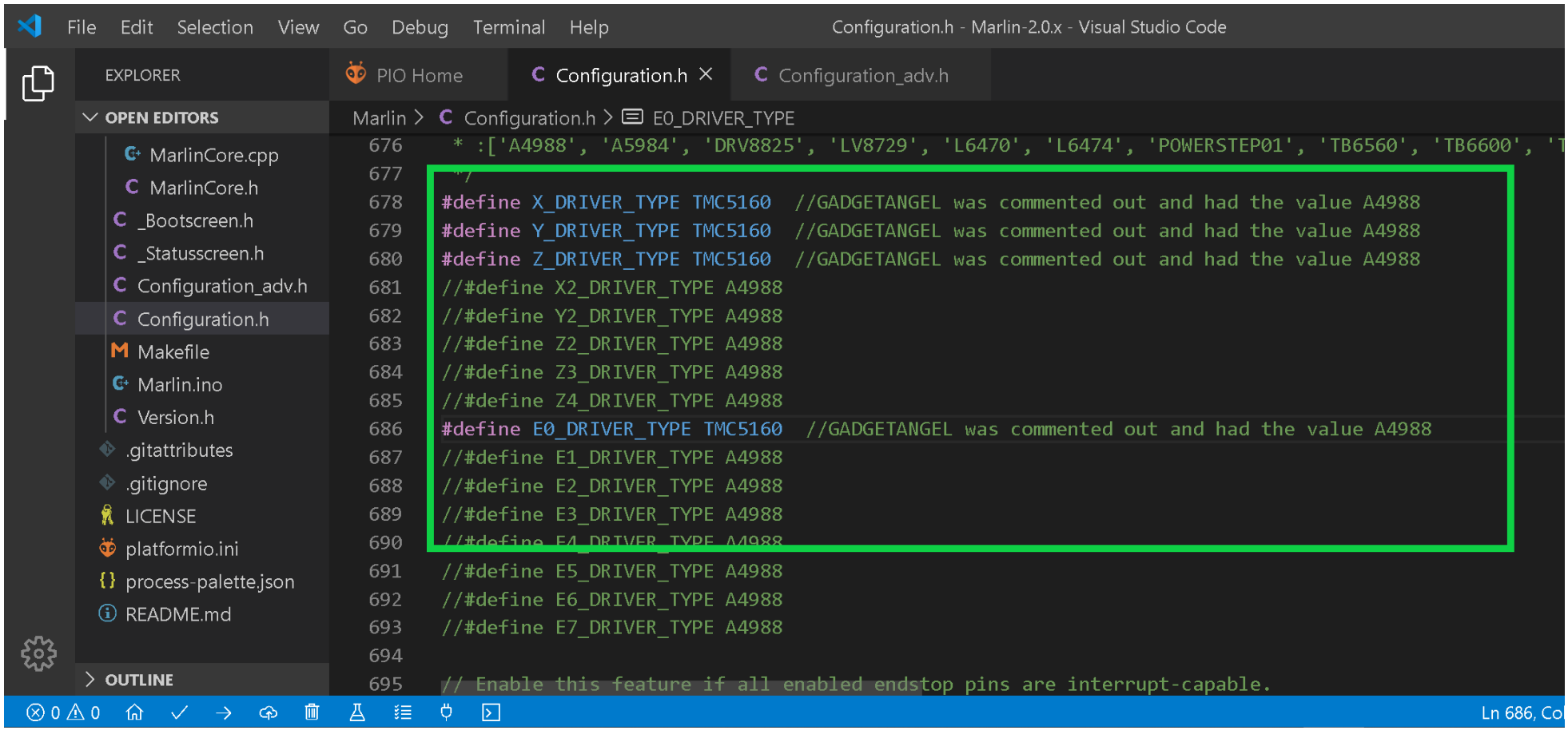


See Appendix D for legend

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

NOTE: **Go to Appendix C**, and then come back here for the changes to Marlin for BIQU TMC5161 V1.0 stepper motor drivers in SPI mode.

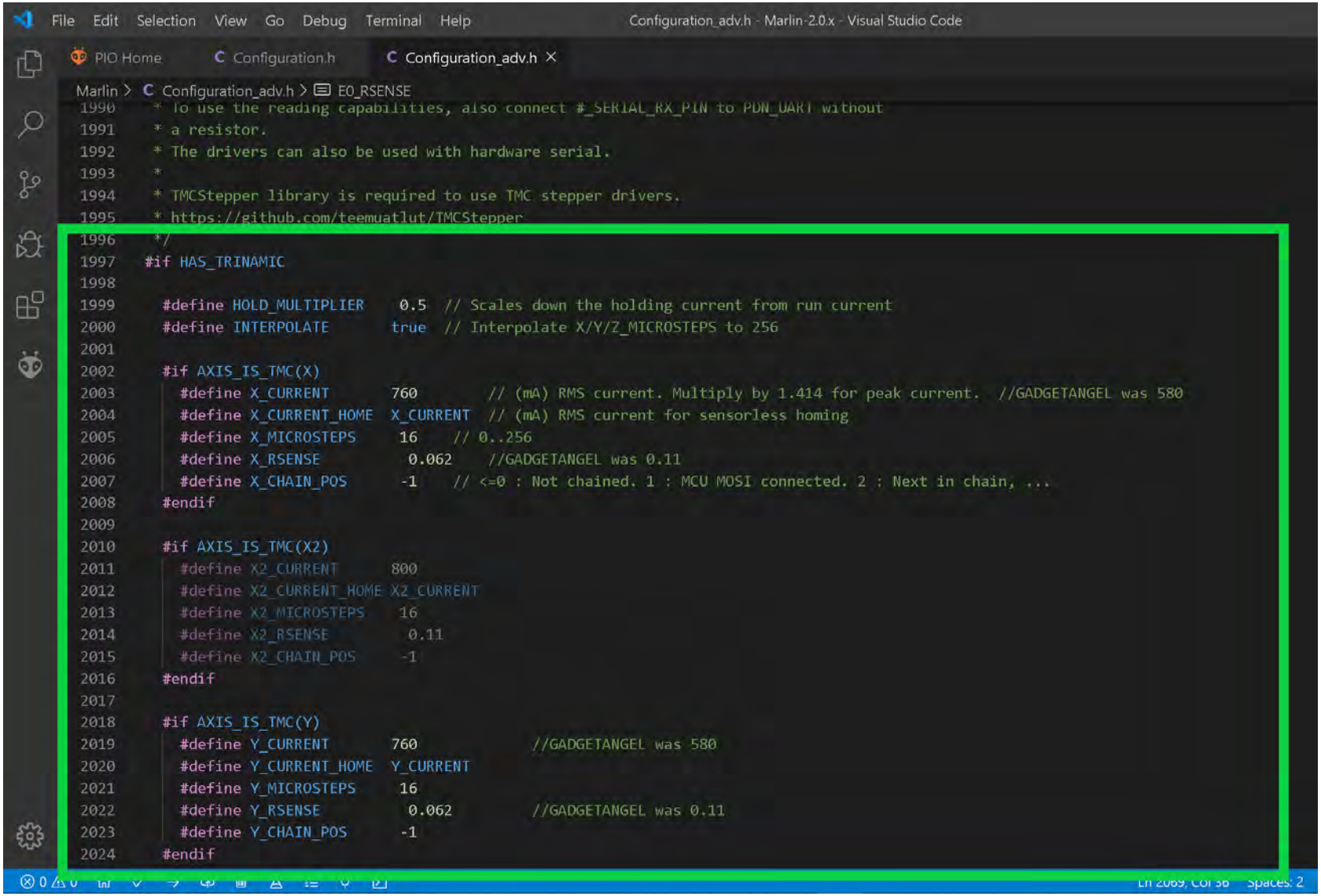
- Change the stepper motor drivers so that Marlin knows you are using TMC5161 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5161 drivers in SPI mode. When two "//" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



- Go to the next page.

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

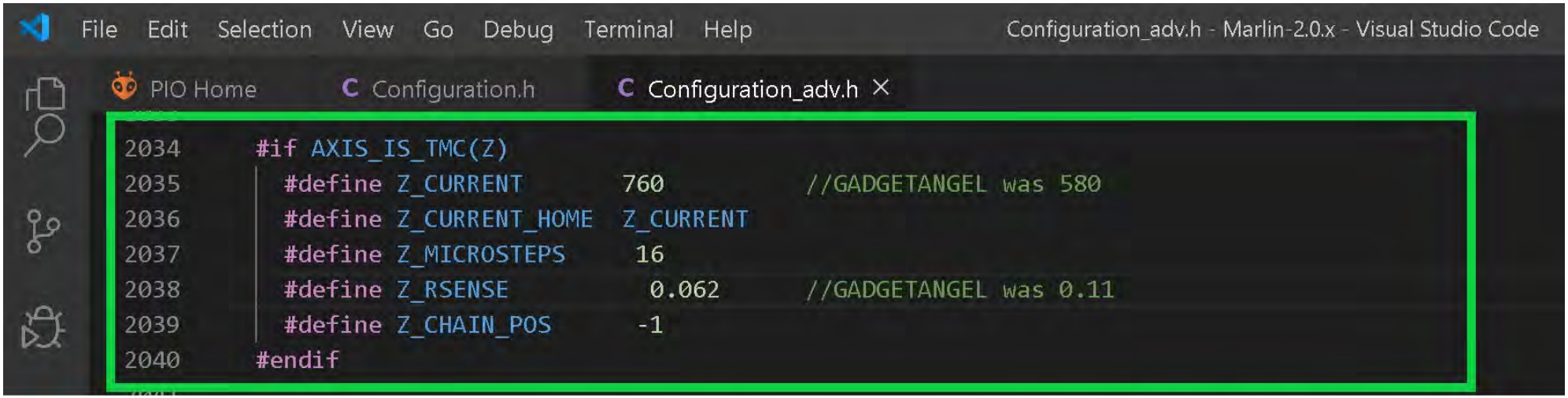
- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC5161 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my 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".

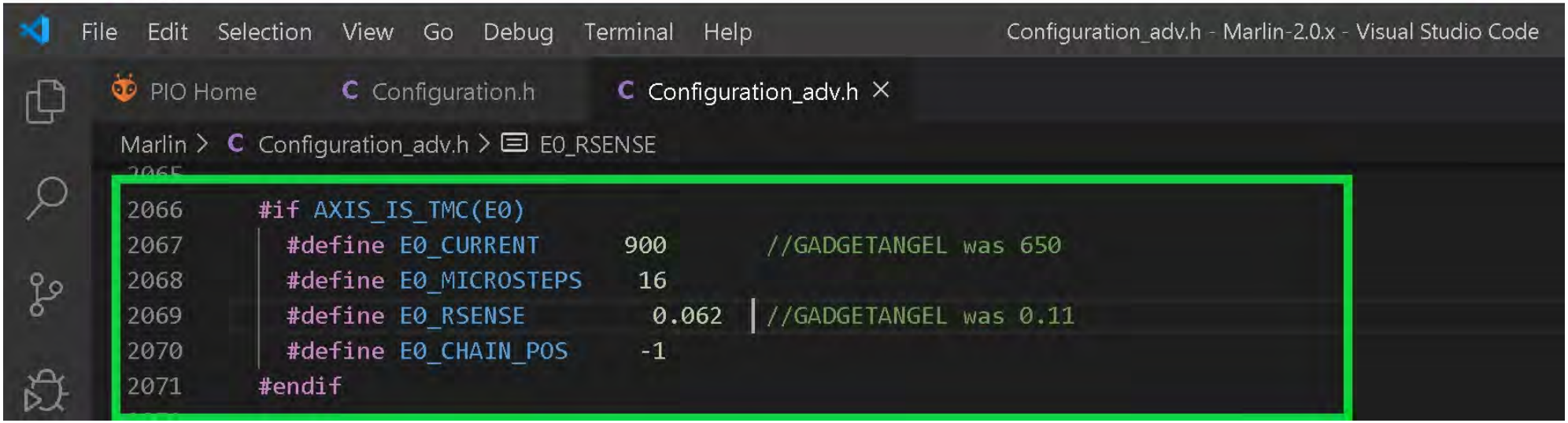


- Go to the next page.

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".

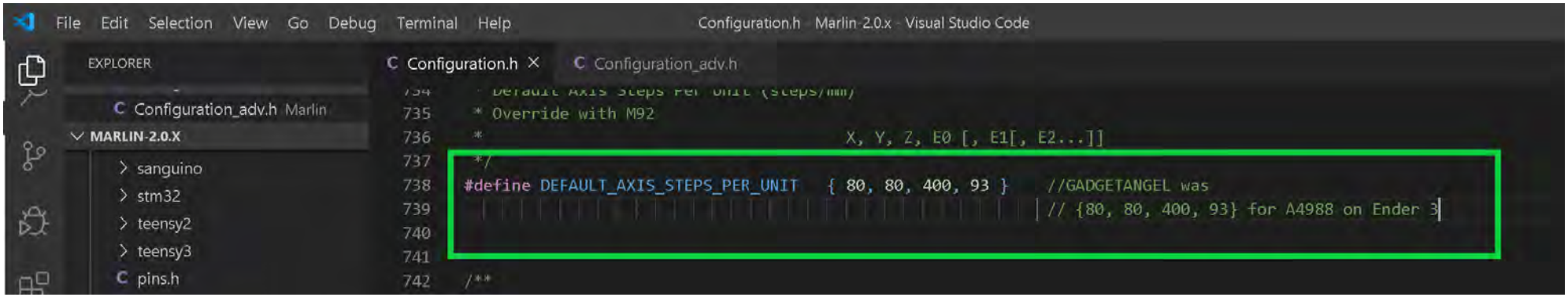




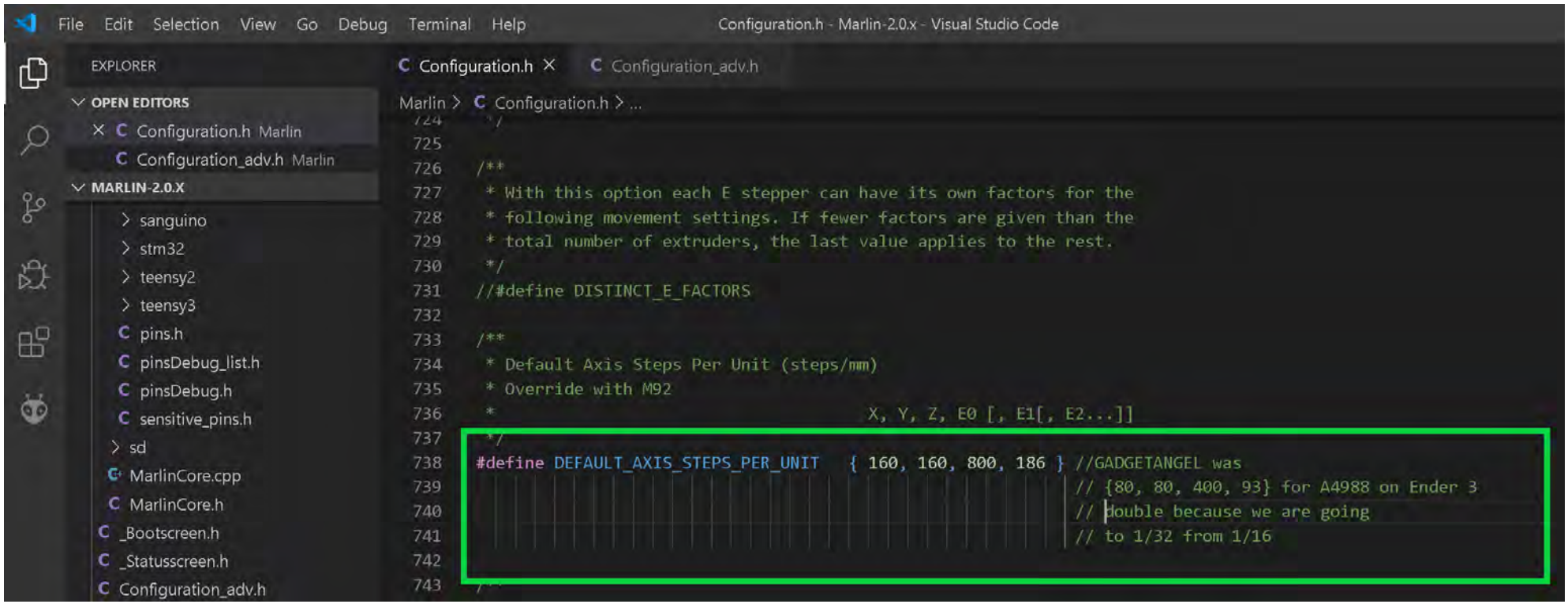
- Go to the next page.

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



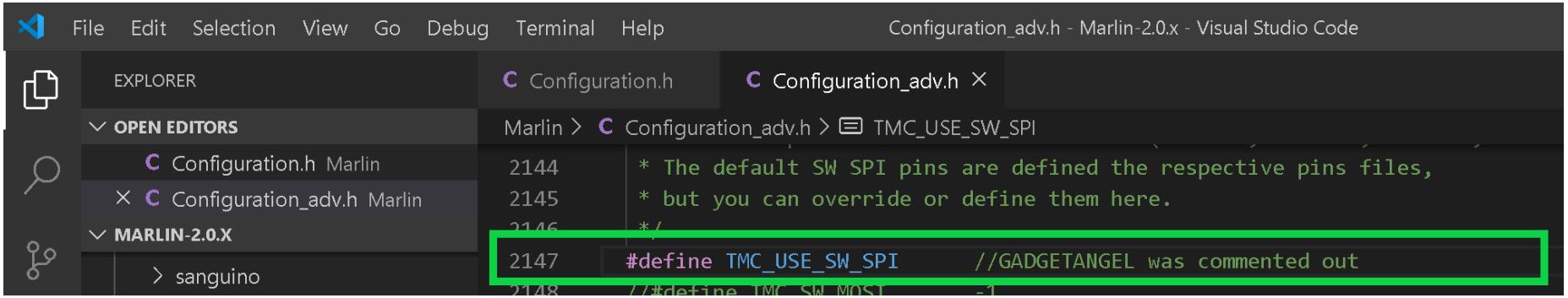
- FOR EAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box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



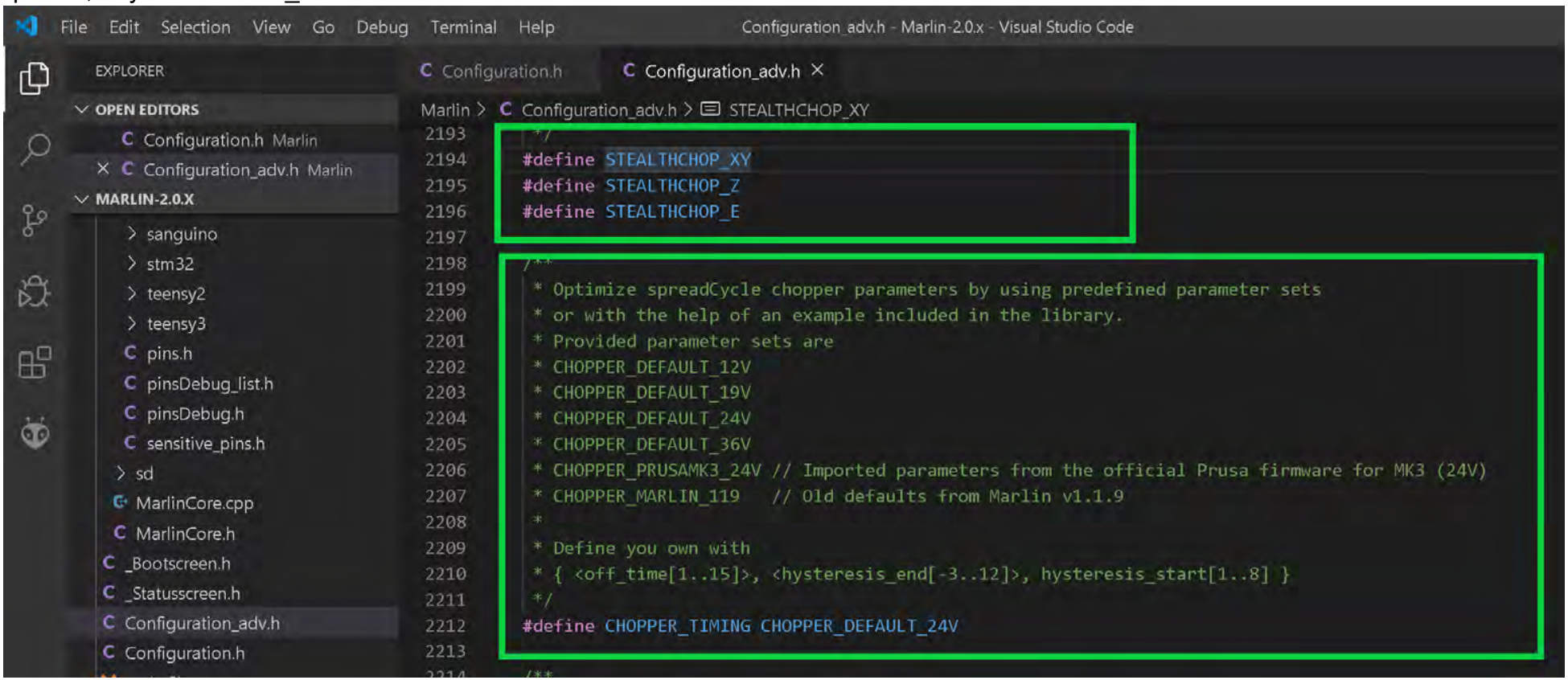
- Go to the next page.

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 PRO V1.1 pins file depends on this variable to define its SPI pins



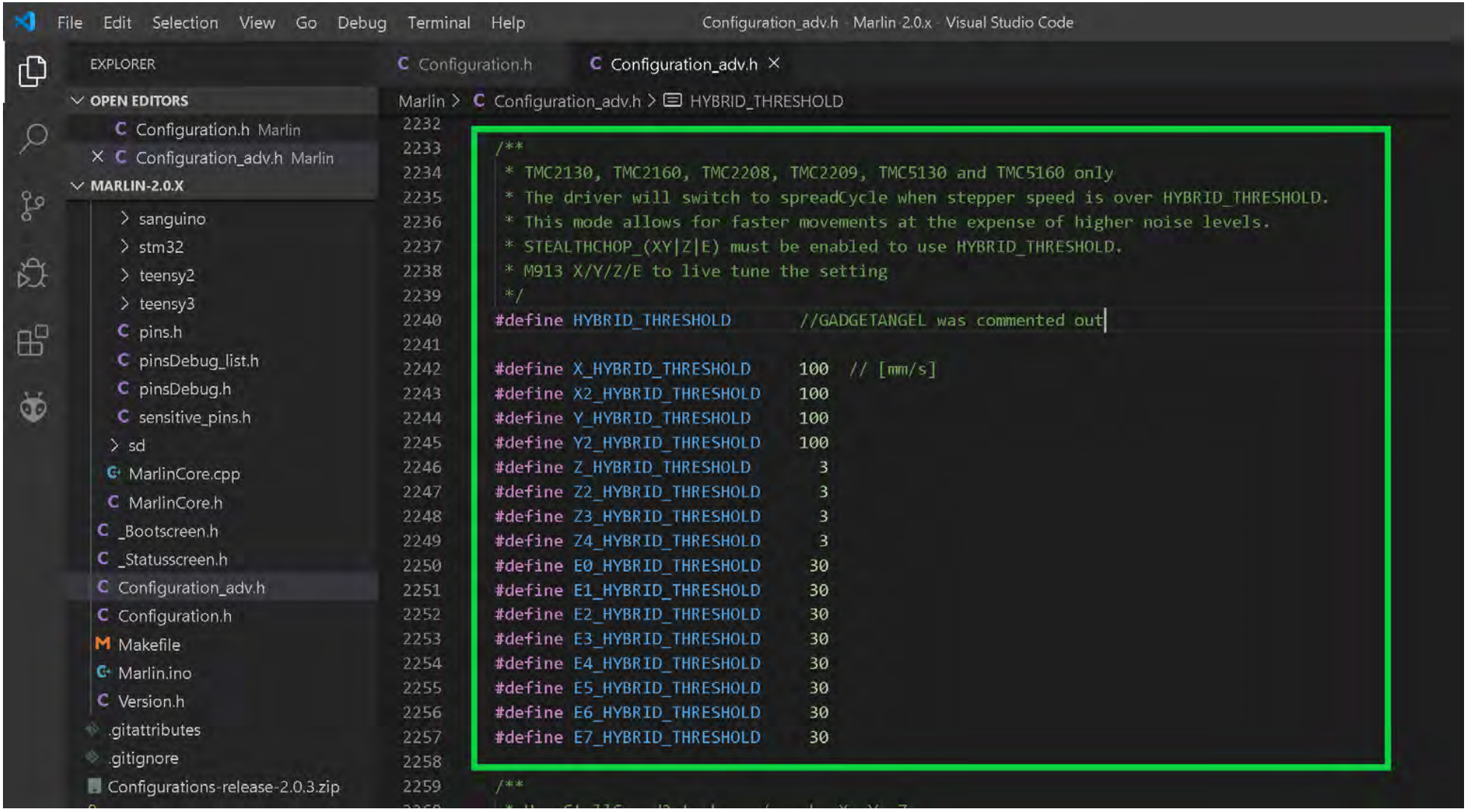
- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP_XY", "STEALTHCHOP_Z" and "STEALTHCHOP_E"}. You also want to check to see if 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

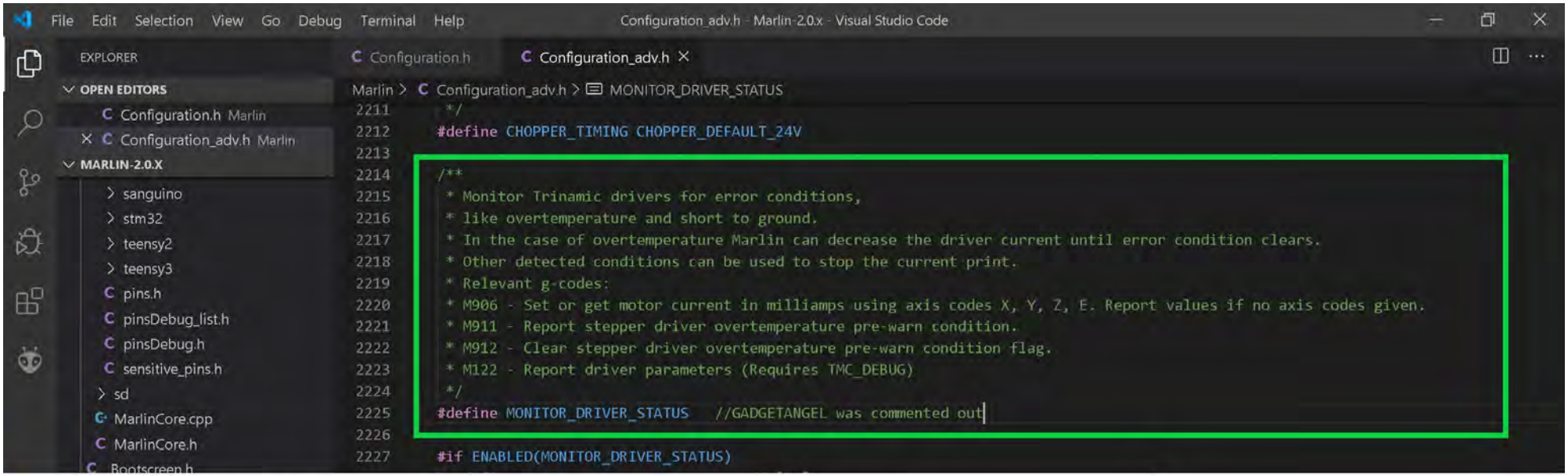
- Now you either enable "HYBRID_THRESHOLD" or disable it. By default, it is disabled. "HYBRID_THERSHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments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

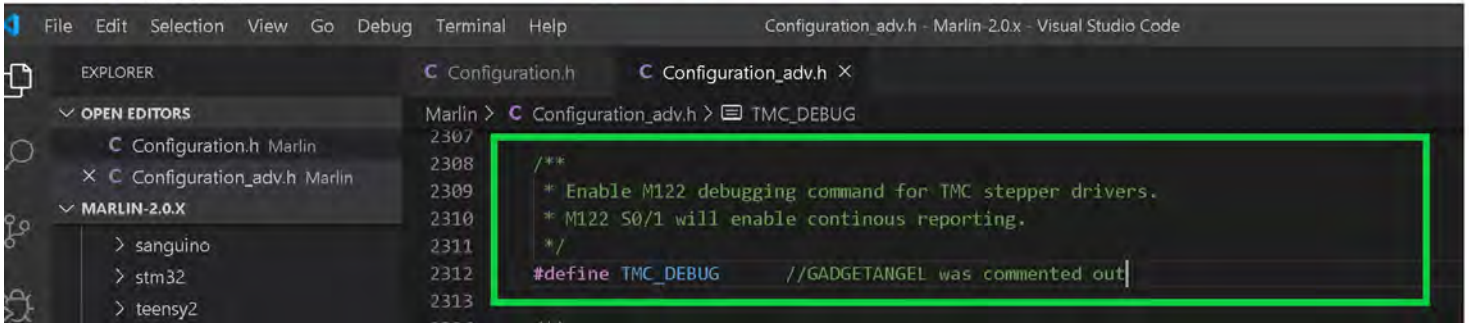


- Go to the next page.

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.**

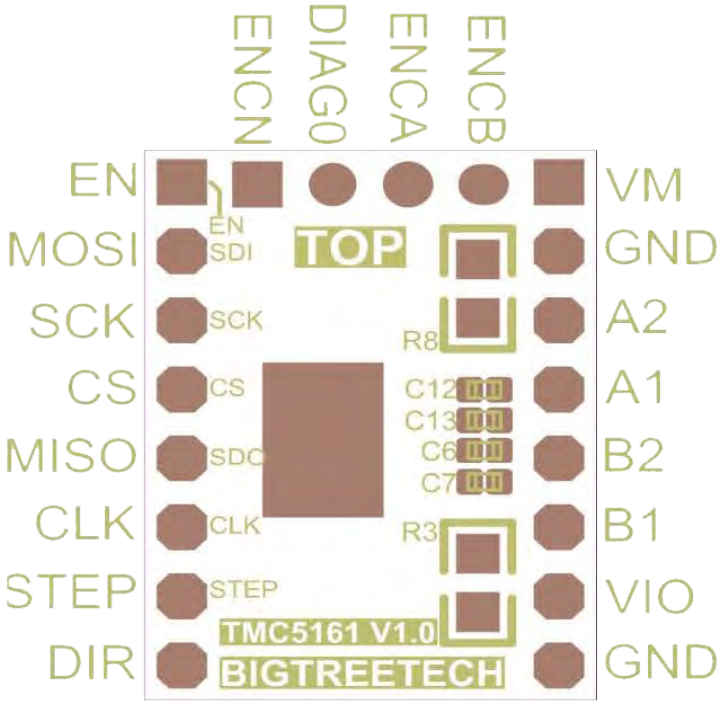


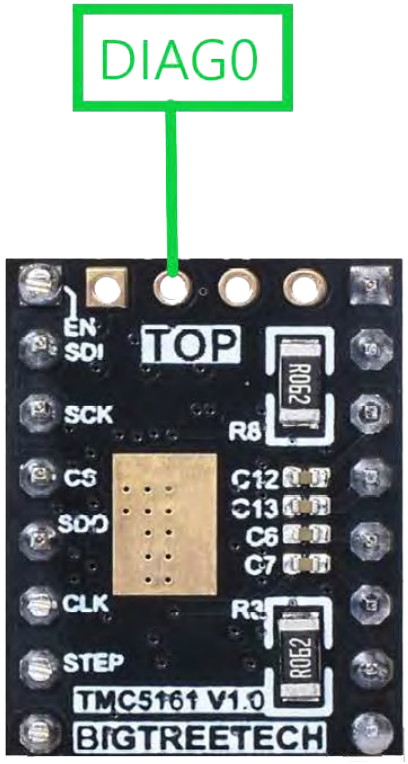


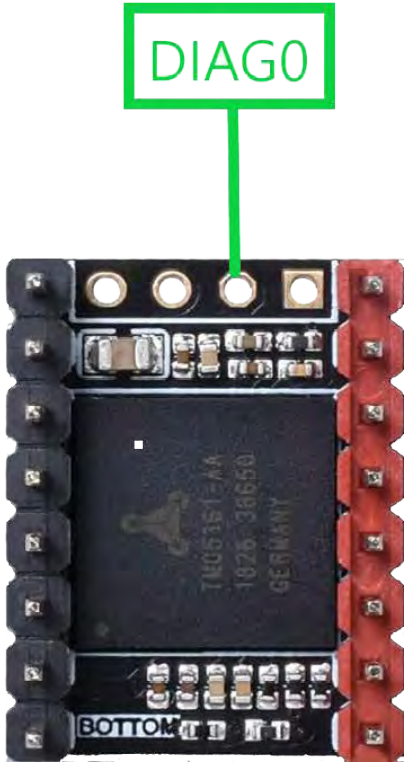
- Go to the next page.

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.





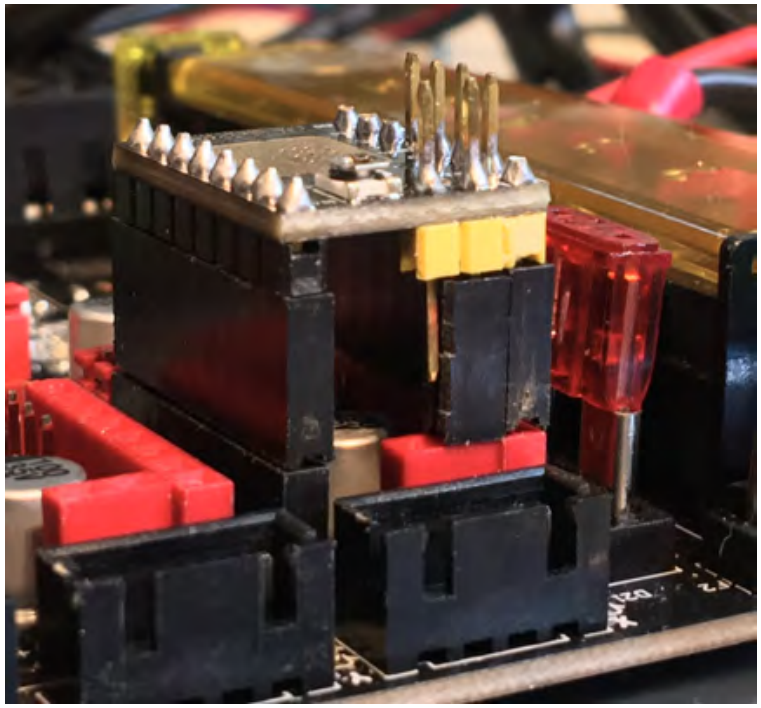


- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR PRO V1.1 board the DIAG0 pin will be seated in the SKR PRO V1.1 board.

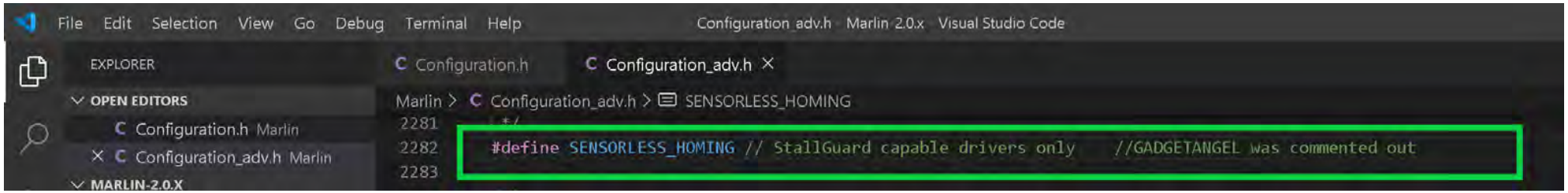
- Go to the next page.

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

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



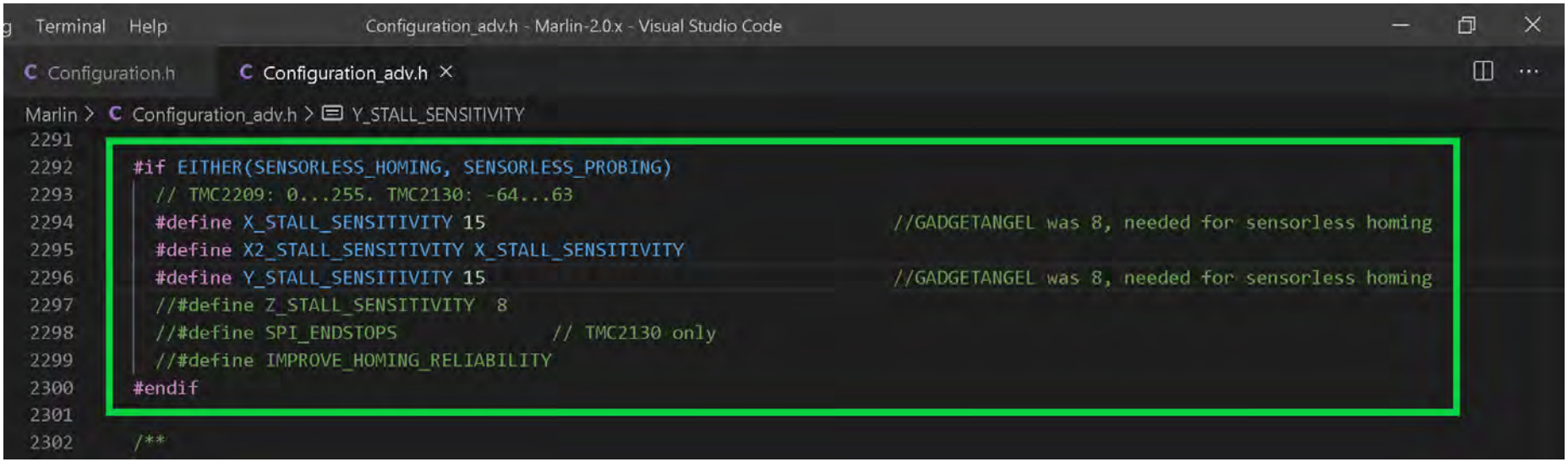
- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS_HOMING"



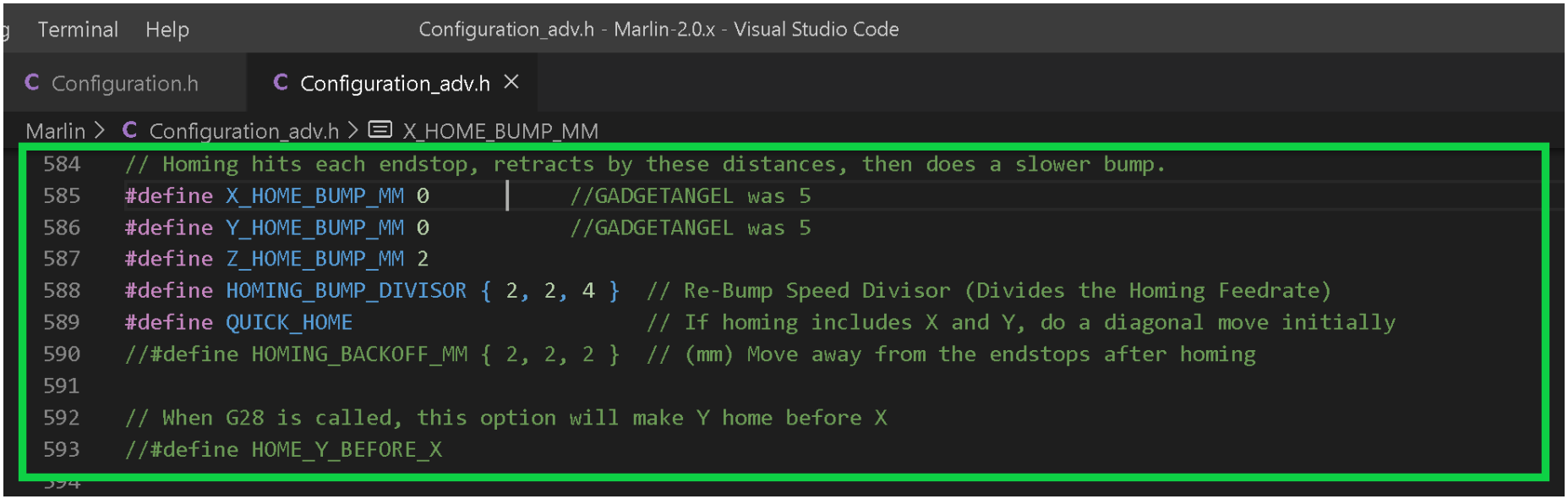
- Go to the next page.

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".



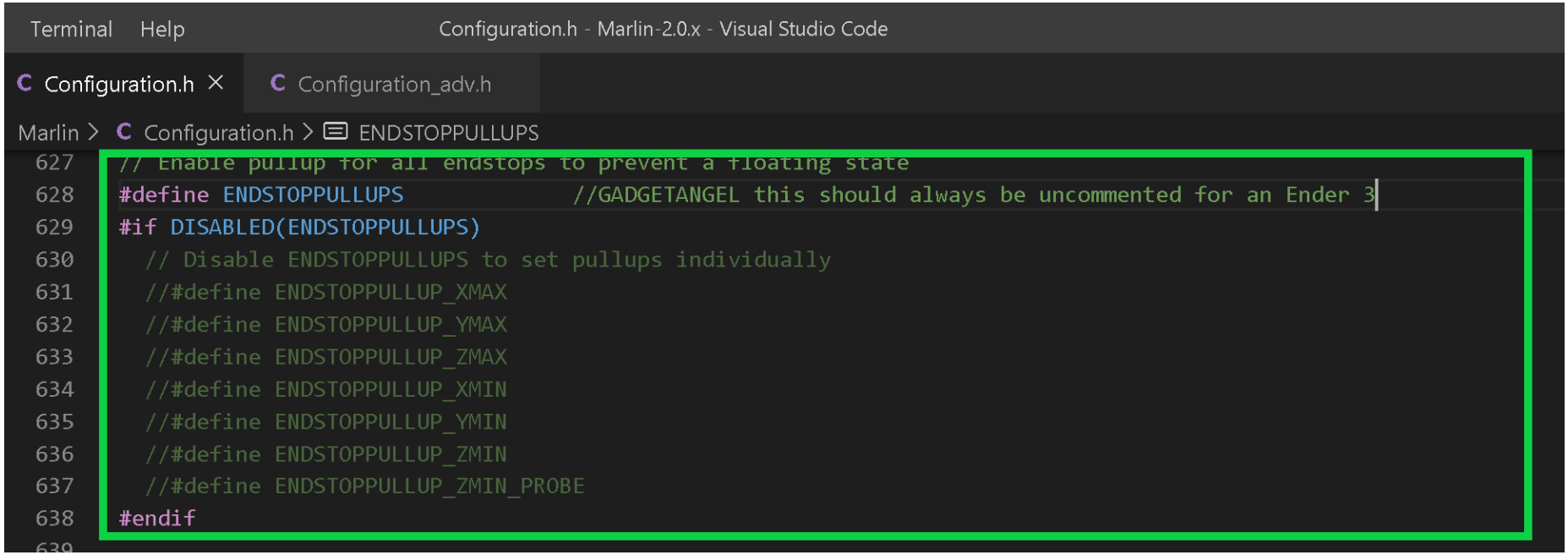
- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X_HOME_BUMP_MM" to 0 and "Y_HOME_BUMP_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z_HOME_BUMP_MM" to 0.



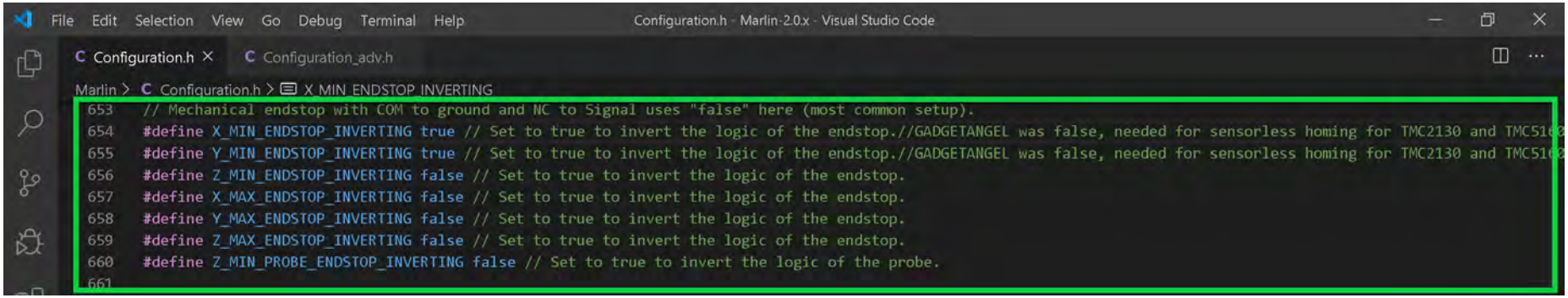
- Go to the next page.

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

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



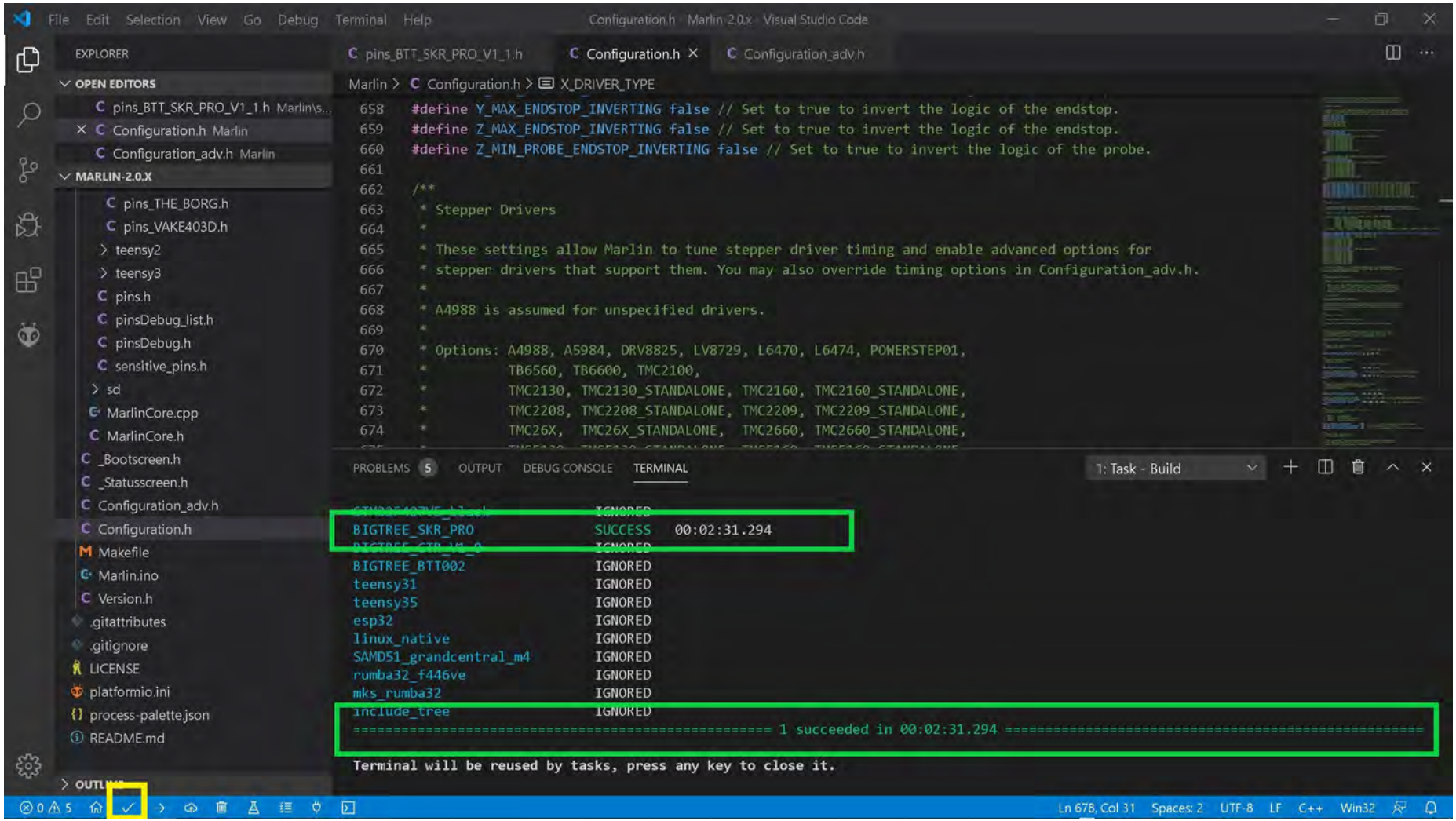
- Next to allow sensor-less homing to work (while using the BIQU TMC5161) we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z_MIN_ENDSTOP_INVERTING" set to false.



- 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.



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

How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer[1, 2]

Modern 3D printers usually use NEMA17 motors. The first piece of information you will need is the "Rated Current" of your NEMA17 motor. So, write down your motor's part number and pay attention to the Step Angle, Holding Torque, Rated Current, Voltage, and Inductance.

Use the Step Angle to work out your printer's "DEFAULT_AXIS_STEPS_PER_UNIT" for your firmware with:

1. **http://www.prusaprinters.org/calculator/**

How to Tune Stepper Motor 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 PRO V1.1 board.
2. When done, turn power off the printer, plug in the stepper motor cables turn power back on your printer and test motor movement.

NOTE: Don't tune stepper motor drivers with the motors plugged in, if you accidentally set current too high you can fry the motor or the stepper motor driver.

NOTE: Don't plug or unplug stepper motors with the power on the printer (i.e. power on the SKR PRO V1.1 board)

Measure DC voltage between the stepper motor driver's trimpot (POT) or "V_{ref} Test point" and your PSU's (12VDC/24VDC) ground. The ground at the PSU connector to the SKR PRO board is fine to use. Look up the correct current for your motor part number. If you have motors with no part number, assume they have a max of 1.00 amps (I_{MAX}) to be safe. Look up the proper formula for your stepper motor drivers (as show in this document), and find the voltage (but ONLY use 90% of the calculated V_{ref}) which corresponds with the current you want to set. Use a ceramic screw driver to adjust the POT. A ceramic screw driver is nonconductive and if you slip while making the adjustment to the POT you could short circuit the stepper motor driver board (i.e. KILL the driver board) in the process.

Time Saver tip, but more dangerous: Get slip-on alligator clips for your multimeter. Clamp ground to a 12VDC/24VDC ground (PSU Ground) wire and clamp positive to your plastic handle screwdriver. This way you'll measure the voltage as you adjust ("live adjustment") and don't need three hands.

Note: **See the next page for a diagram of the setup.**

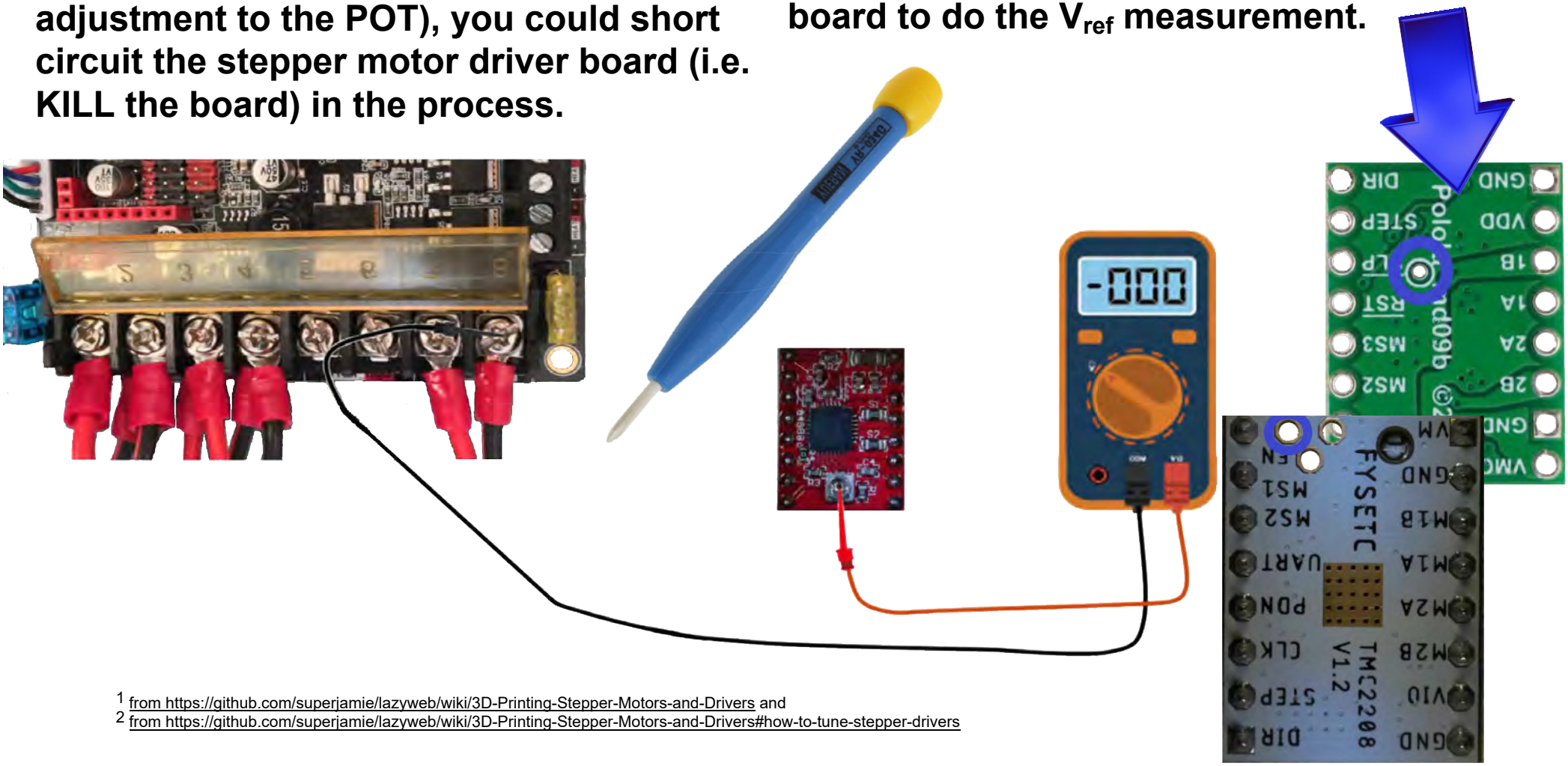
[1] from https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers and
[2] from https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers

How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer[1, 2]

Note: **A ceramic screw driver is non-conductive. If you use a plastic handle screw driver with alligator clips to your multimeter to make a "live adjustment" of V_{ref} (and you slip while making the adjustment to the POT), you could short circuit the stepper motor driver board (i.e. KILL the board) in the process.**

Note: **Some stepper motor driver boards have a "V_{ref} Test point" location, shown in BLUE. Check bottom or top of your board for a location.**

If a "V_{ref} Test point" location is not available, use the potentiometer on the stepper driver board to do the V_{ref} measurement.



[1] from https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers 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.**

APPENDIX B

How to Calculate V_{ref} for TMC Stepper Motor Drivers

My machine is an Ender 3.The X, Y, and Z stepper motors have a "Rated Current" of 0.84 Amps, while E (extruder) stepper motor "Rated Current" is 1 Amp.

We use the I_{MAX} formula and use Algebra to find the V_{ref} formula. The I_{MAX} formula for each TMC Driver is listed on the following pages. But we will use TMC2100 drivers so we will use the below equation:

TMC2100 Stand-alone Mode, (with $R_s = 110m\Omega$):

$$I_{RMS} = ((V_{ref} / 2.5) * (1 / 1.41) * ((320mV / (R_s + 20m\Omega)))$$

Since $I_{MAX} = (I_{RMS} * 1.41)$ is a known value then the above equation can be written as follows:

$$(I_{MAX} * (1 / 1.41)) = ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / (110 + 20)))$$

Since (1 / 1.41) is on both sides of the Algebra equation they cancel each other out leaving the equation as follows:

$$I_{MAX} = (V_{ref} / 2.5) * (2.46 \cong 2 .5),$$

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) :

$I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((\mathbf{320mV} / (\mathbf{R_s} + \mathbf{20m\Omega})))$
$= ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / 110 + 20)))$
$= ((1 / 1.41) * (V_{ref} / 2.5) * (2.46))$
$= (1 / 1.41) * V_{ref} * 0.99$
$= (1 / 1.41) * V_{ref} * 1$
Since $I_{RMS} = I_{MAX} * (1 / 1.41)$
Therefore,
$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$
$\mathbf{I_{MAX} = V_{ref}}$
$\mathbf{V_{ref} = I_{MAX}}$

2. TMC2130 with R_s = 0.110Ω (110mΩ) :

$I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((\mathbf{325mV} / (\mathbf{R_s} + \mathbf{20m\Omega})))$
$= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20)))$
$= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5))$
$= (1 / 1.41) * V_{ref}$
Since $I_{RMS} = I_{MAX} * (1 / 1.41)$
Therefore,
$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$
$\mathbf{I_{MAX} = V_{ref}}$
$\mathbf{V_{ref} = I_{MAX}}$

3. TMC2208 with R_s = 0.110Ω (110mΩ) for Stand-alone Mode:

$I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((\mathbf{325mV} / (\mathbf{R_s} + \mathbf{30m\Omega})))$
$= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 30)))$
$= ((1 / 1.41) * (V_{ref} / 2.50) * (2.32143))$
$= (1 / 1.41) * V_{ref} * 0.928572$
Since $I_{RMS} = I_{MAX} * (1 / 1.41)$
Therefore,
$I_{MAX} * (1 / 1.41) = (1/1.41) * V_{ref} * 0.928572$
$\mathbf{I_{MAX} = V_{ref} * 0.9286}$
$\mathbf{V_{ref} = I_{MAX} * 1.0769}$

- 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 = 0.110Ω (110mΩ) for UART Mode:

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} * 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}

5. TMC2209 with R_s = 0.110Ω (110mΩ) for Stand-alone Mode:

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} * 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}

- 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) * ((**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} * 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}

7. TMC2225 with R_s = 0.150Ω (150mΩ) for UART Mode:

I_{RMS} = ((1 / 1.41) * (V_{ref} / 2.5) * ((**325mV** / (**R_s + 20mΩ**)))
= ((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
I_{MAX} = V_{ref} * 0.7647
V_{ref} = I_{MAX} * 1.3077

8. TMC5160 with R_s = 0.075Ω (75mΩ) for SPI Mode:

I_{RMS} = ((1 / 1.41) * (**325mV** / **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
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 + **30mΩ**)))
= ((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-PRO-V1.1 User Manual.pdf**
- **BIGTREETECH SKR-PRO-V1.1 Guide.pdf**

This example will use the Creality Ender 3 printer. Select the appropriate default configuration files for your specific printer!

- Download the latest release of Marlin from here: **https://marlinfw.org/meta/download/**
- Unzip the latest release of Marlin onto your hard drive
- Also, download the latest release of the Marlin Configuration files and then unzip the Marlin Configuration files so they reside in the same subdirectory as the Marlin files, see the pictures below for how I organize my file structure for Marlin

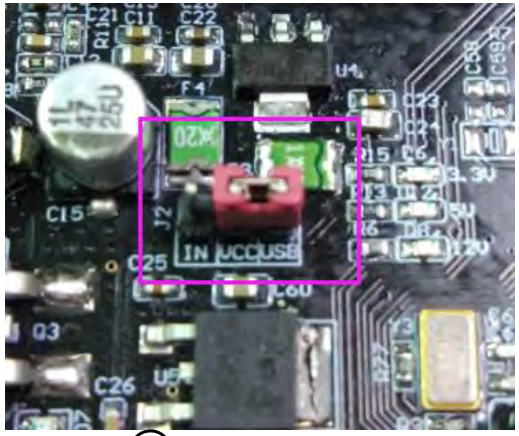
- Go to the next page.

APPENDIX C

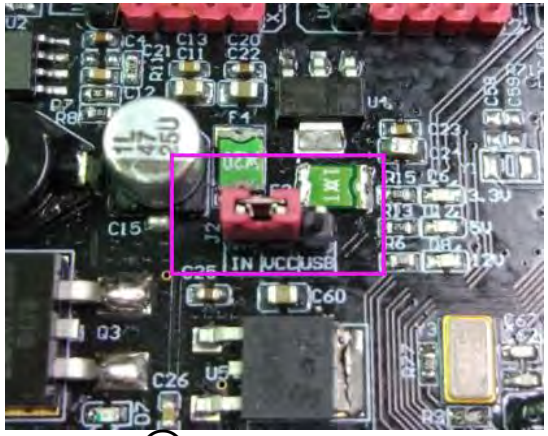
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

To ensure that any changes I make to my Marlin Firmware are permanent, I do a couple of things. I will send the G-code M503 by using the **Pronterface software** **BEFORE I change the Marlin Firmware** so I can **write down all the calibrations that I have previously set**.

I ensure that the "Power selection", as defined in "SKR PRO V1.1 user manual v1.1.pdf" is set to the correct power input source ②. Even though you can use the USB as the source of power, I prefer to use the 12V/24V DC power source, as shown below.



①USB power



②12/24V power

Marlin2.0 Firmware Update Method:

After downloading the files, use Visual Studio Code (VScode) to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the micro SD card into the board. Reboot the board; wait for about 10 seconds before doing anything else with the board. I then use the **Pronterface software**, to send the following "G-code" commands:

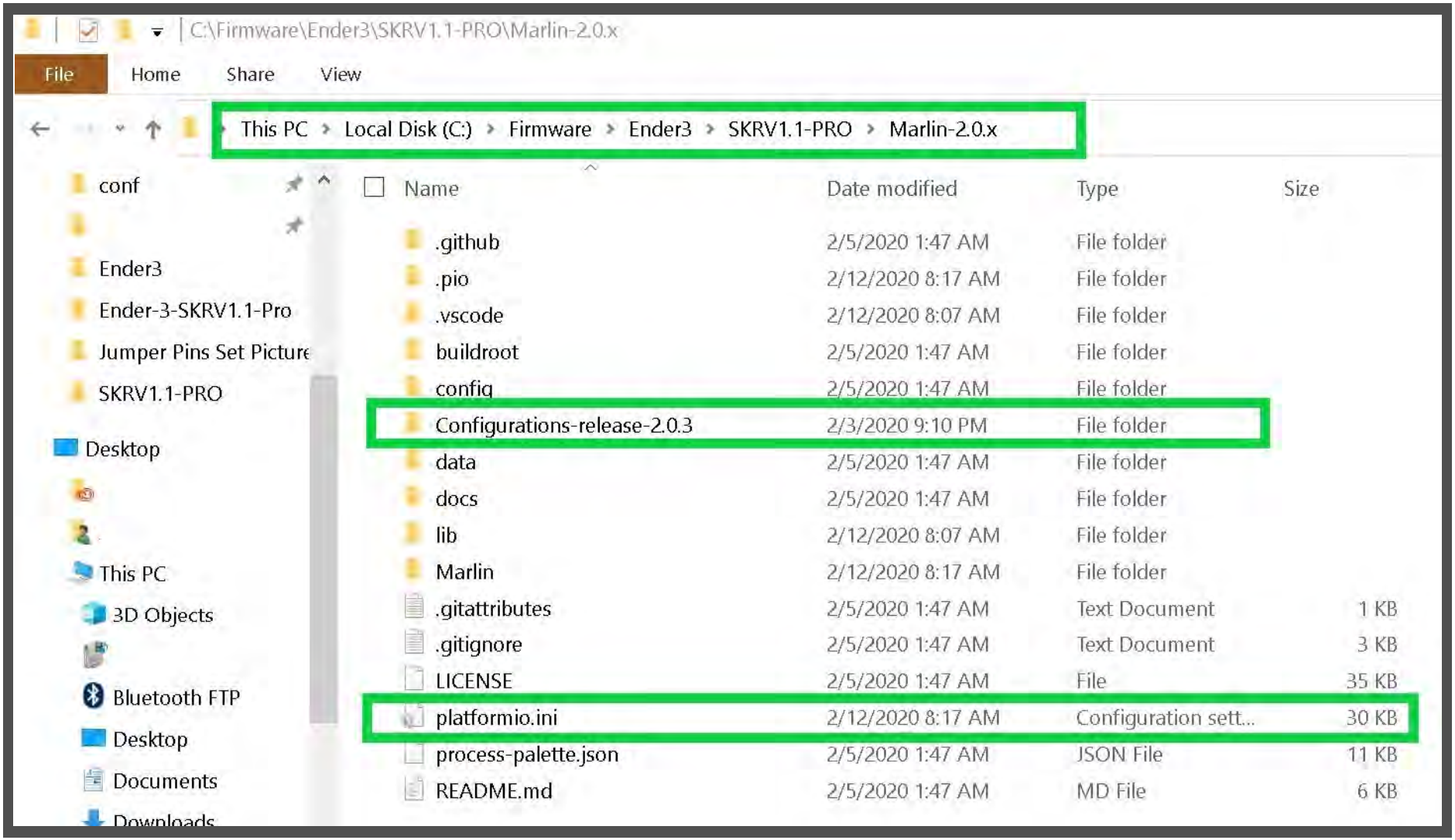
- M502
- M500
- M504

M502 reset all configurable settings to their factory defaults. When you follow a M502 by a M500 the M500 will also reset settings in EEPROM. The M504 command will validate the contents of the EEPROM to ensure that the EEPROM settings have been changed to the factory defaults. If the reset does not show the correct settings, find your compiled "firmware.bin" file and copy it again to the SD card, then plug the micro SD card into the SKR PRO board. Reboot the board, wait for about 10 seconds and check the settings again.

After uploading new firmware, you will need to calibrate your 3D printer again. Please see the following document for instructions on how to calibrate your 3D printer: How to Calibrate your 3D printer

APPENDIX C

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



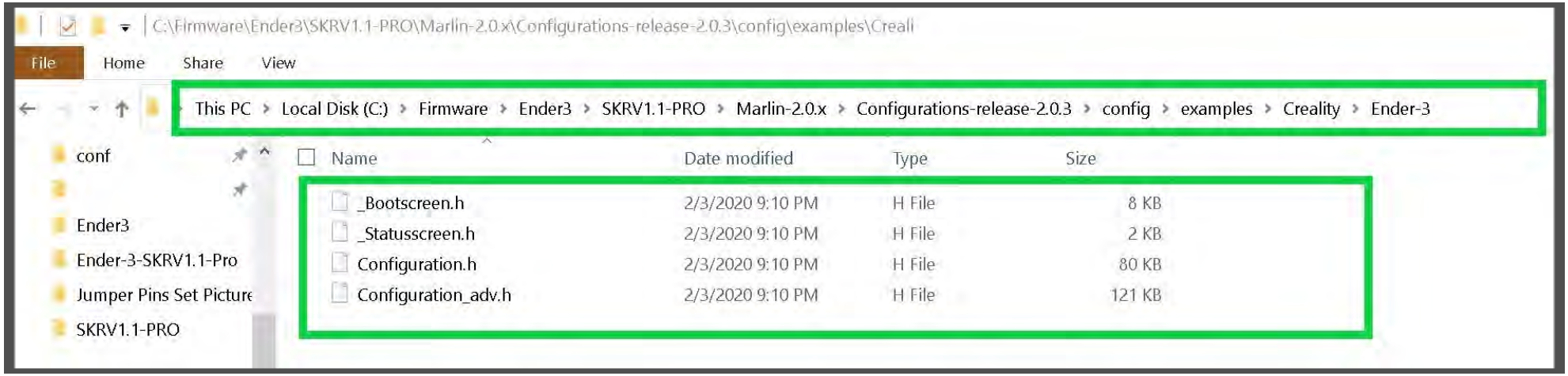
- Go to the next page.

APPENDIX C

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

NOTE: **This example will use the Creality Ender 3 printer and using the latest release of Marlin firmware which is version 2.0.3. Select the appropriate default configuration files for your specific printer!**

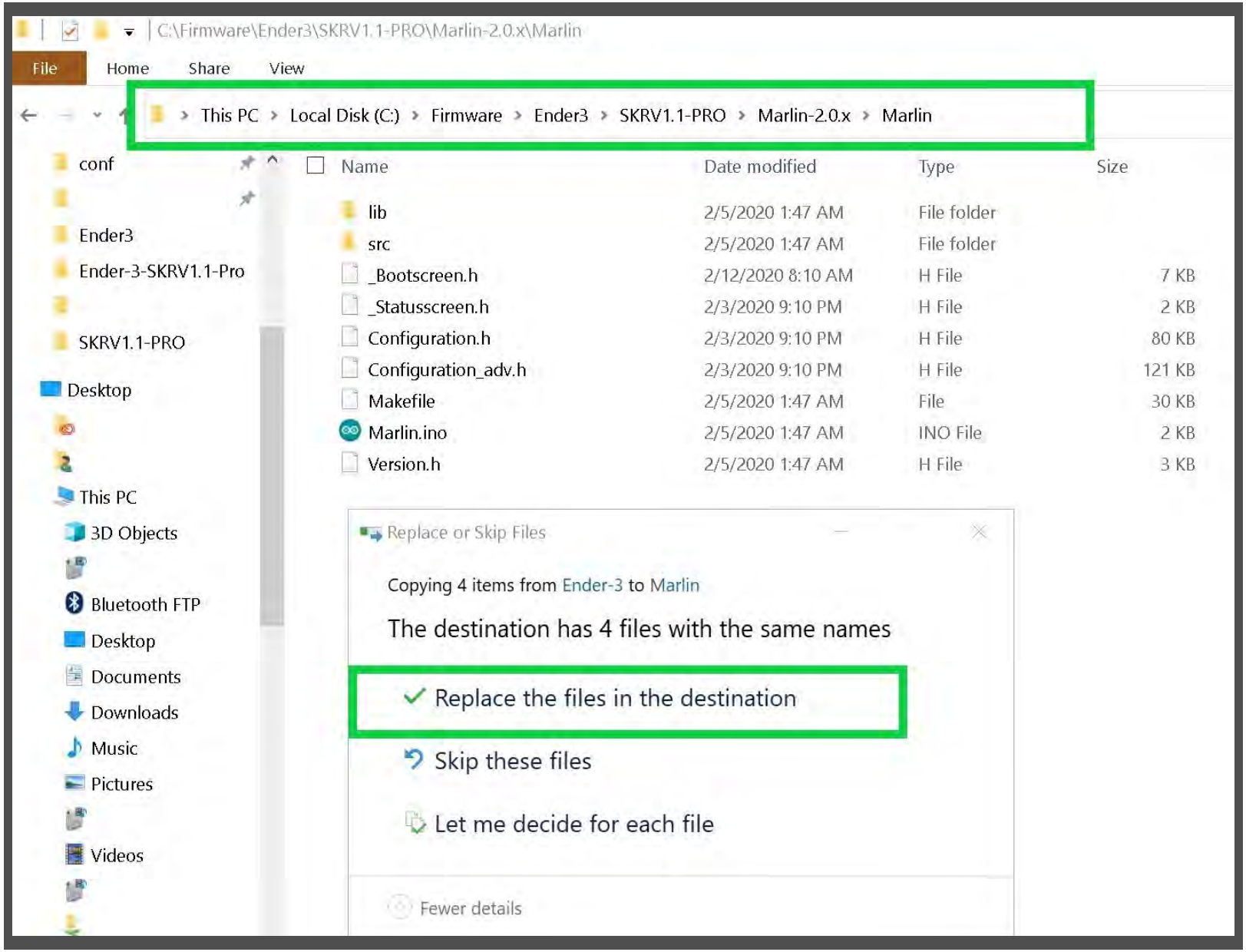
- Copy the below default configuration files (for me Ender 3 files, for you it could be another printer) to the directory where Marlin's Configuration.h and Configuration_adv.h reside. See picture below for which files you will copy. See the next page to see where to place the files.



- Go to the next page.

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

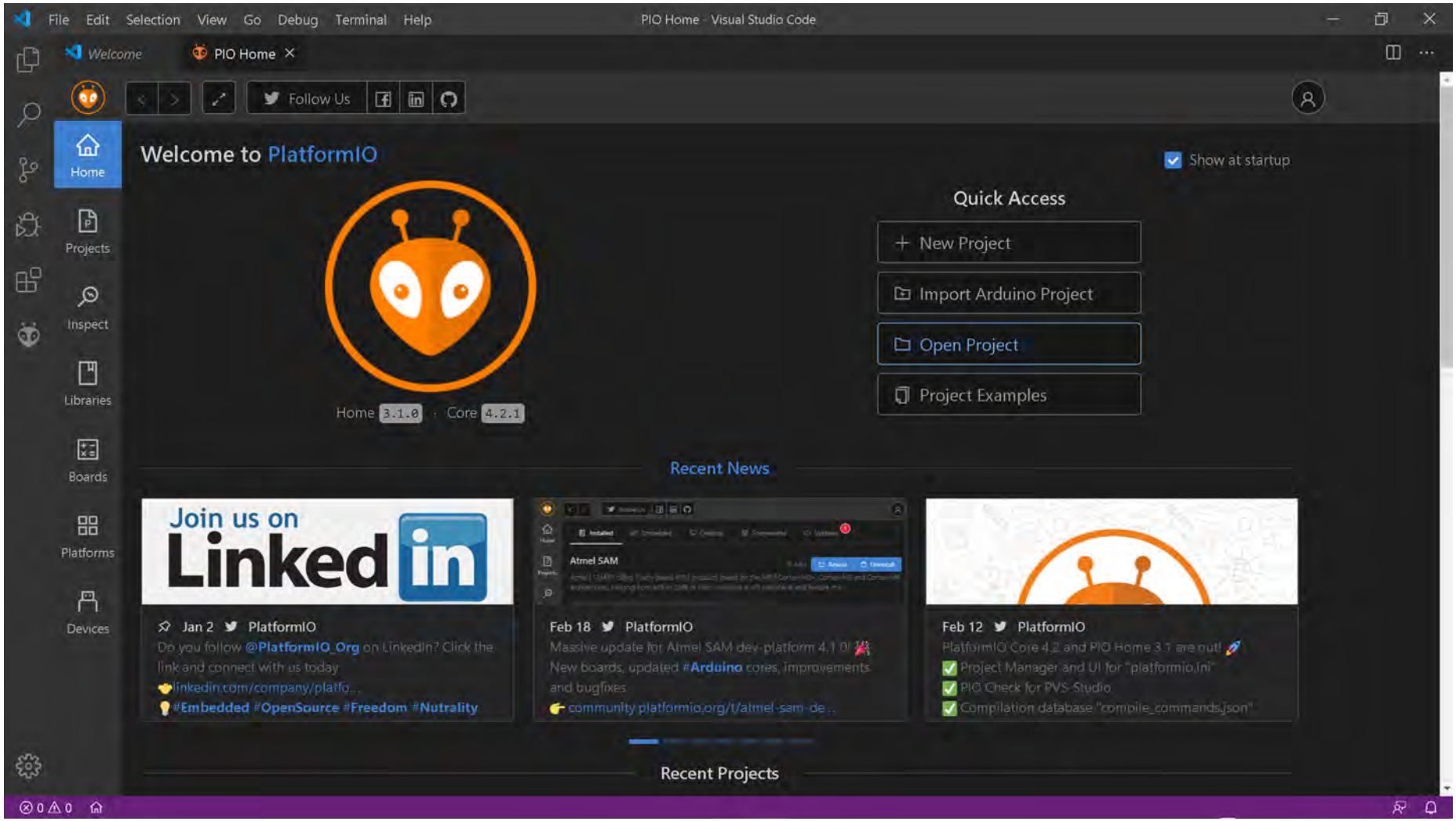
- Copy TO THIS directory so the above files reside in the same directory as Configuation.h and Configuration_adv.h. When prompted allow the files to be overwritten!



- 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".

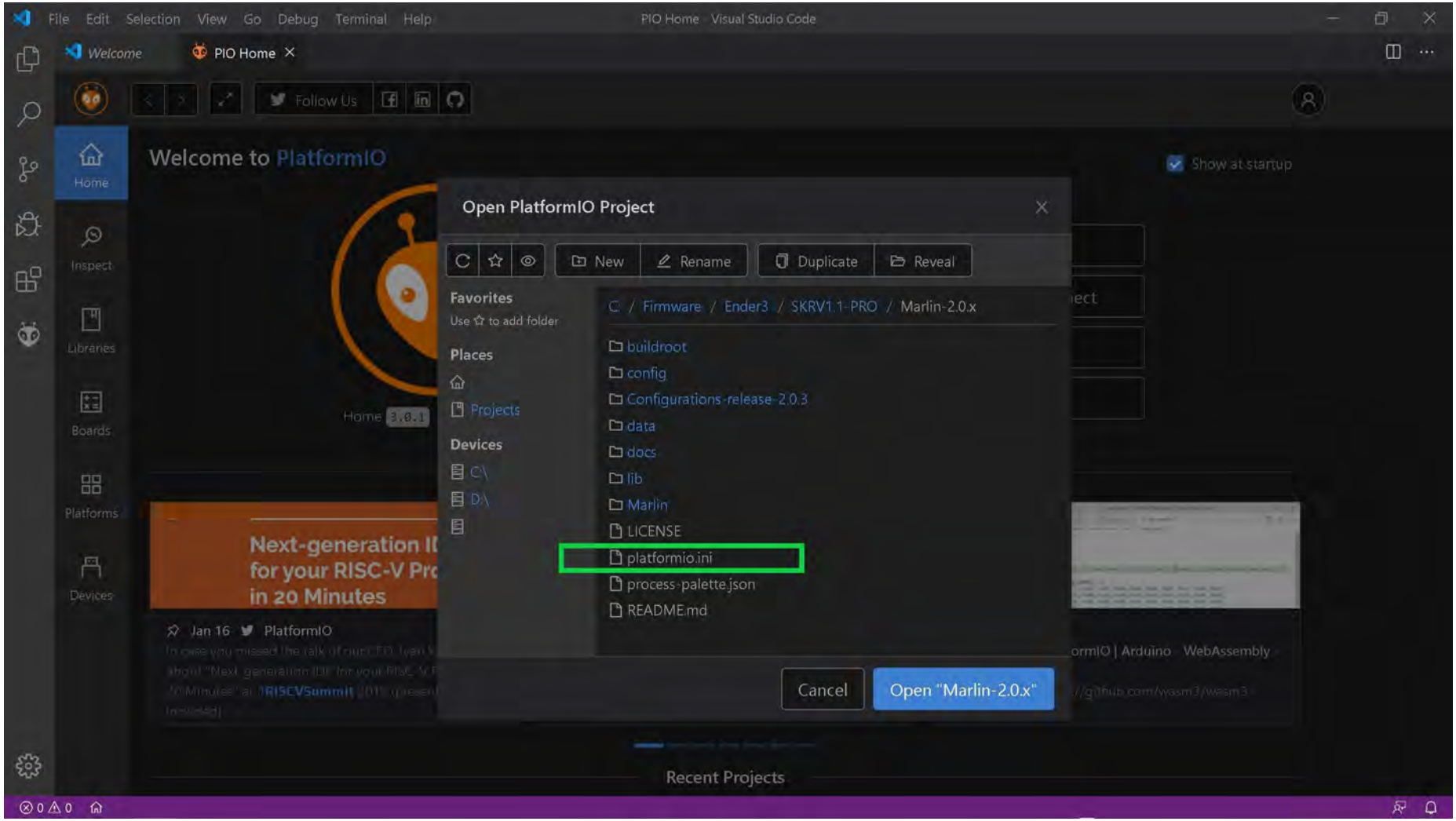


- Go to the next page.

APPENDIX C

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

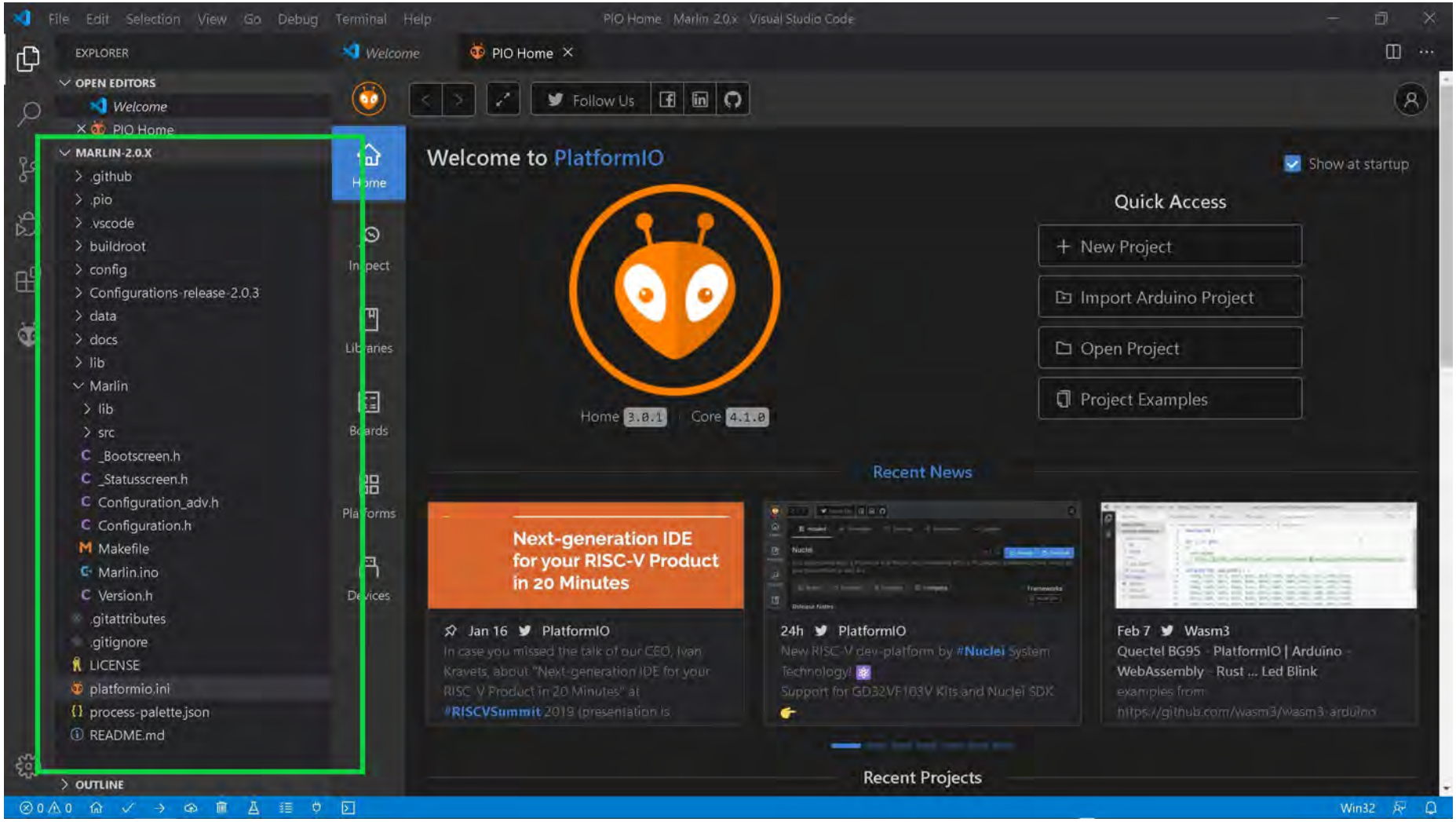
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- Go to the next page.

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

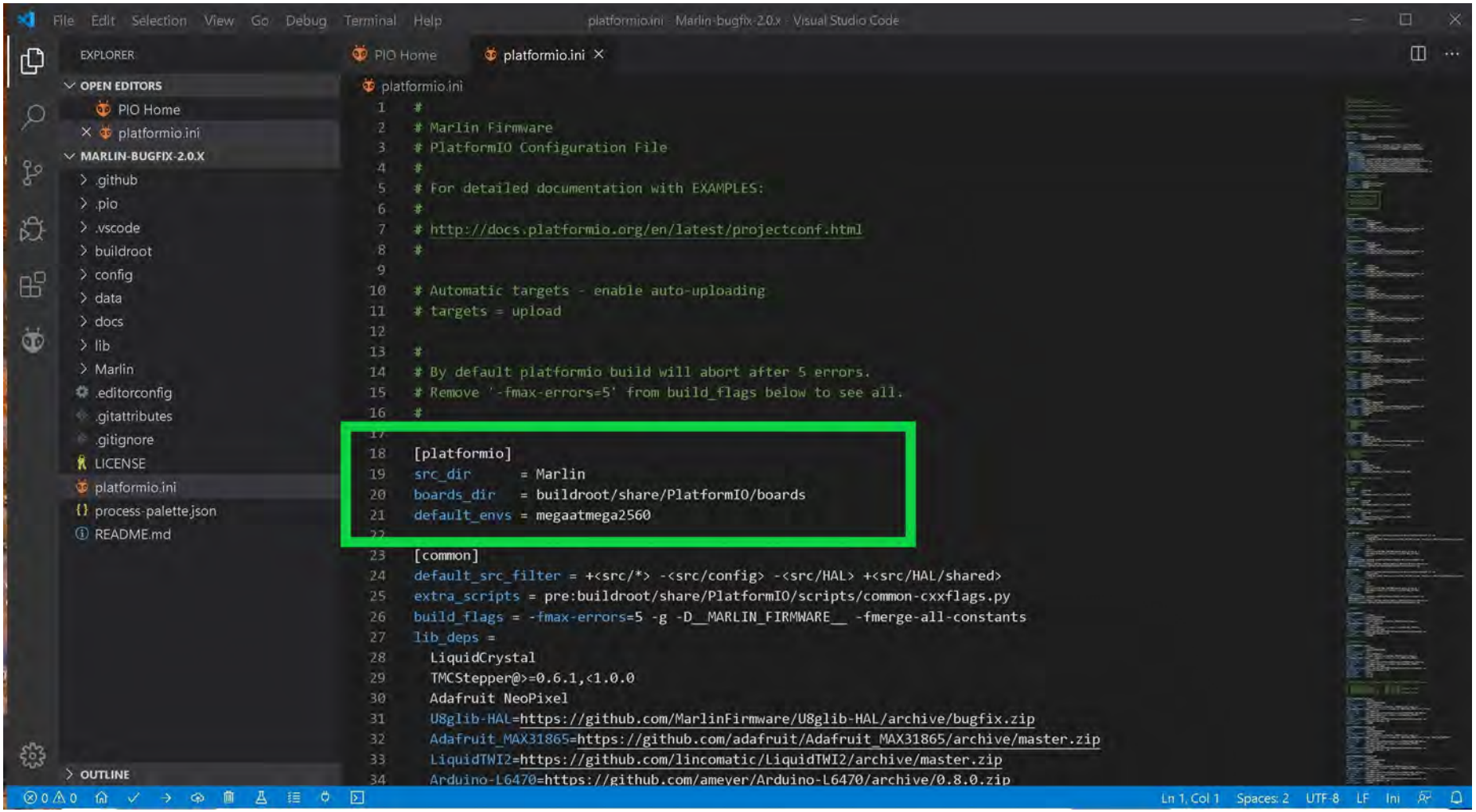
- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



- Go to the next page.

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

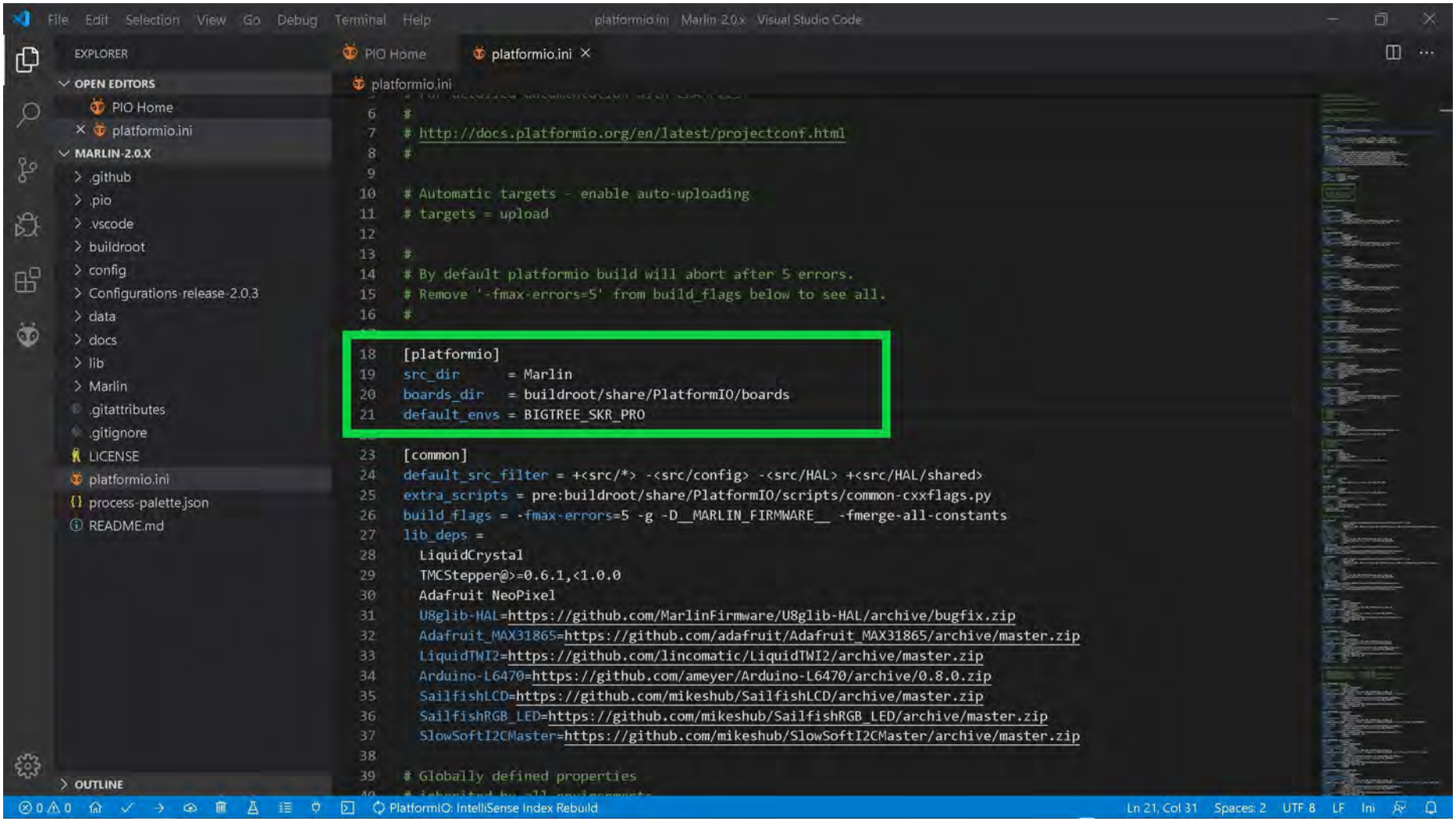


- Go to the next page.

APPENDIX C

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

- Edit the "default_envs = megaatmega2560" line and make "default_envs = BIGTREE_SKR_PRO", as shown below in **GREEN**

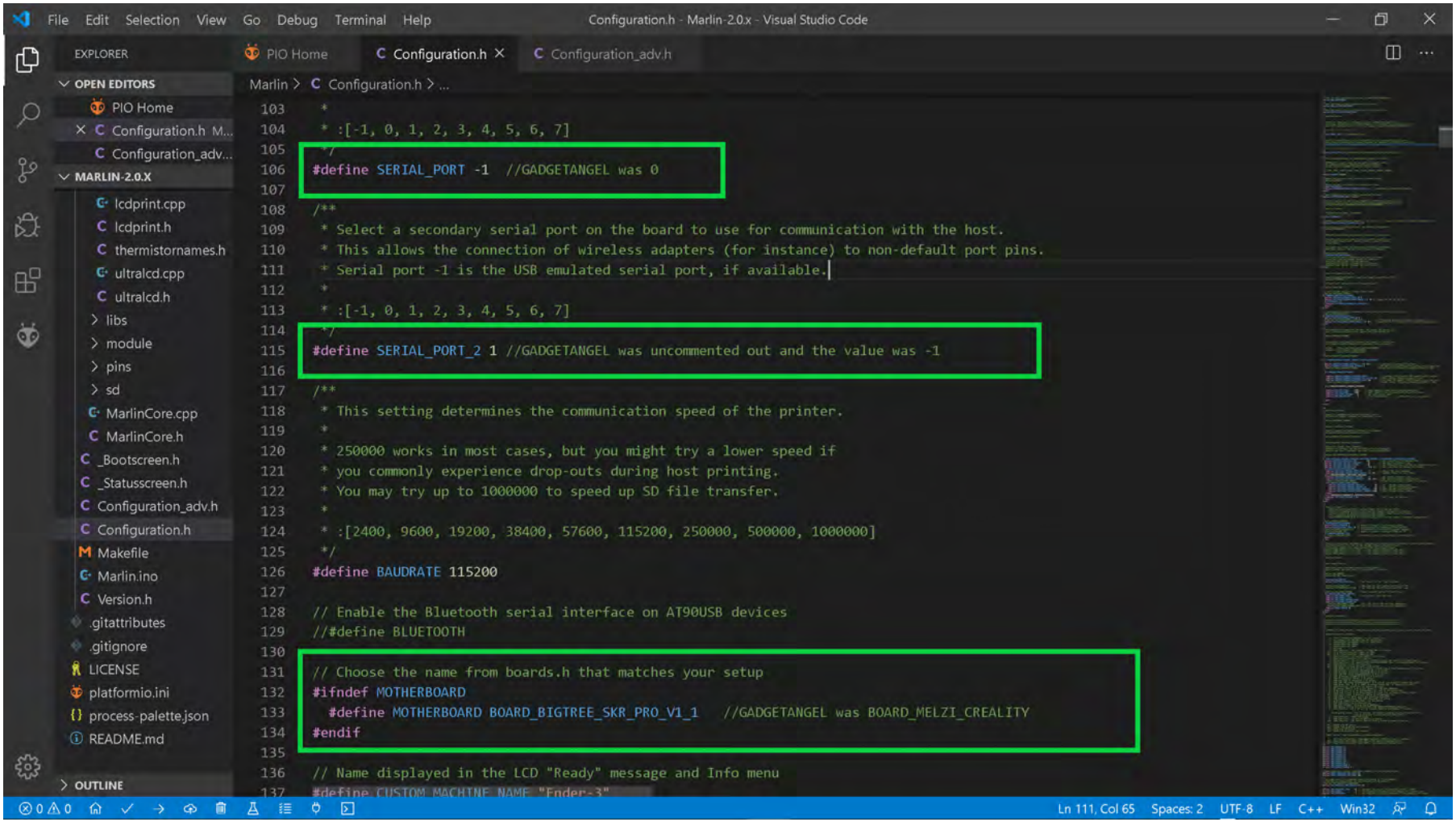


- Go to the next page.

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 **3 GREEN** boxes below.

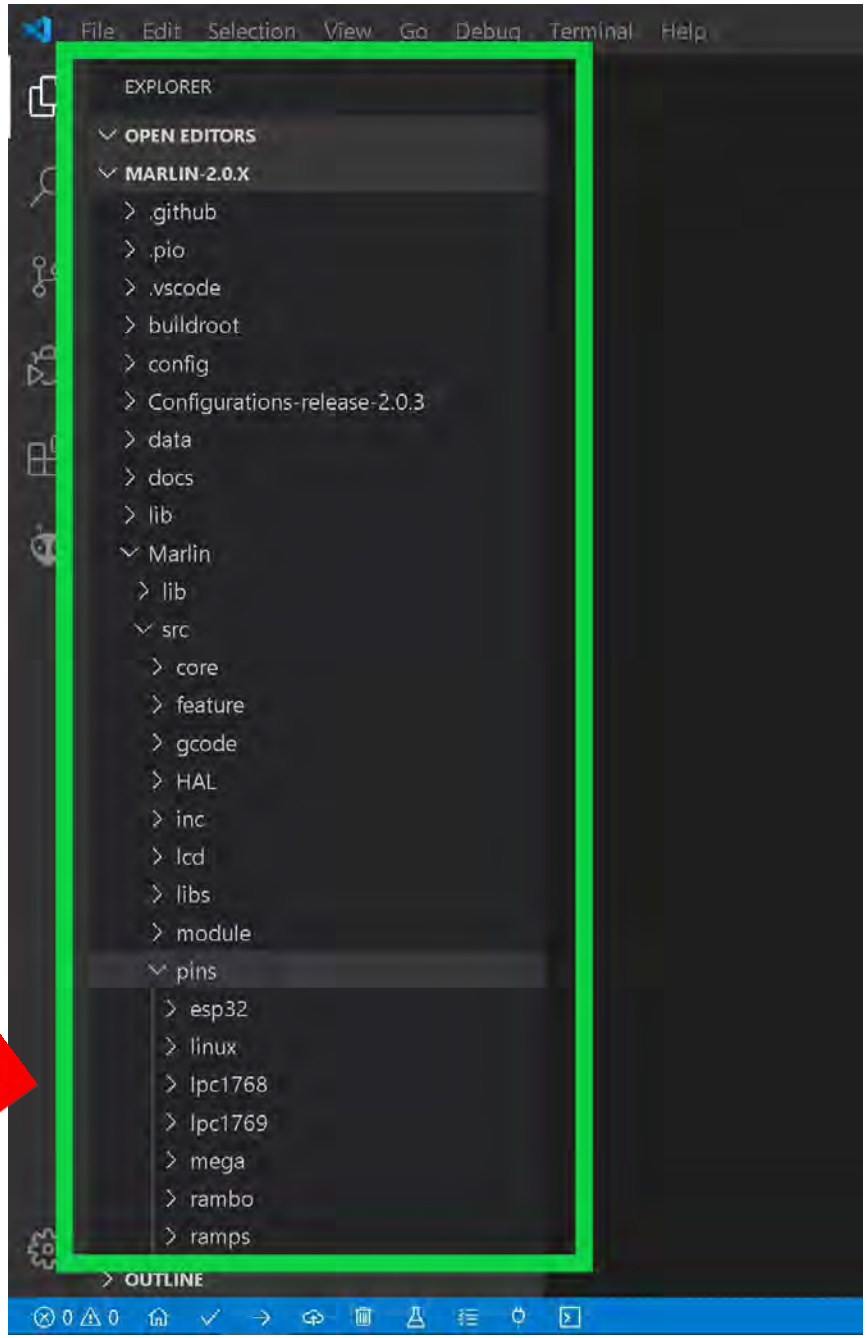


- You can set "BAUDRATE" to "115200" or "250000 ". Either setting will work but I have found that "115200" option works with any LCD that you choose to use.

- Go to the next page.

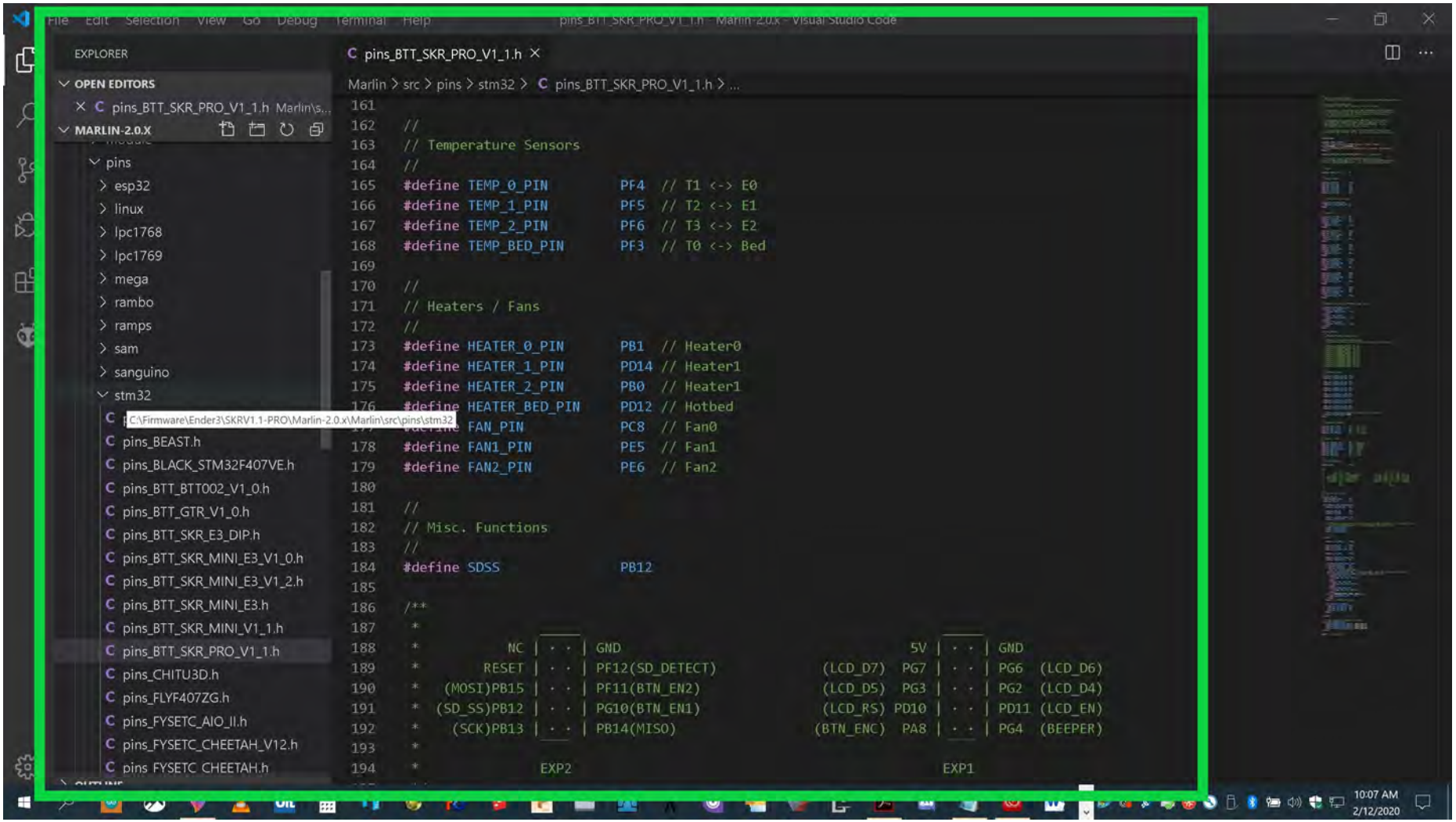
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers

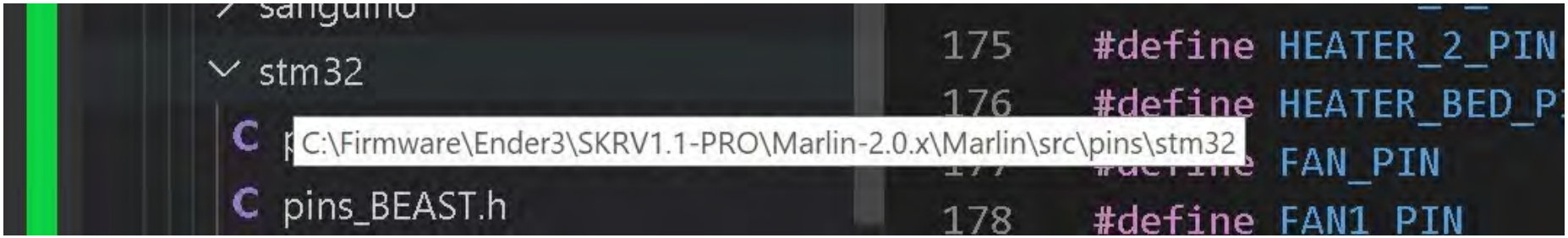
- Time to learn where the SKR PRO V1.1 board's pins file is located: look at the left side and find the Marlin-2.0.X/Marlin/src/pins/stm32 subdirectory, as seen in the pictures below. Open the file, pins_BTT_SKR_PRO_V1_1.h, by double clicking on it.



- Go to the next page.

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

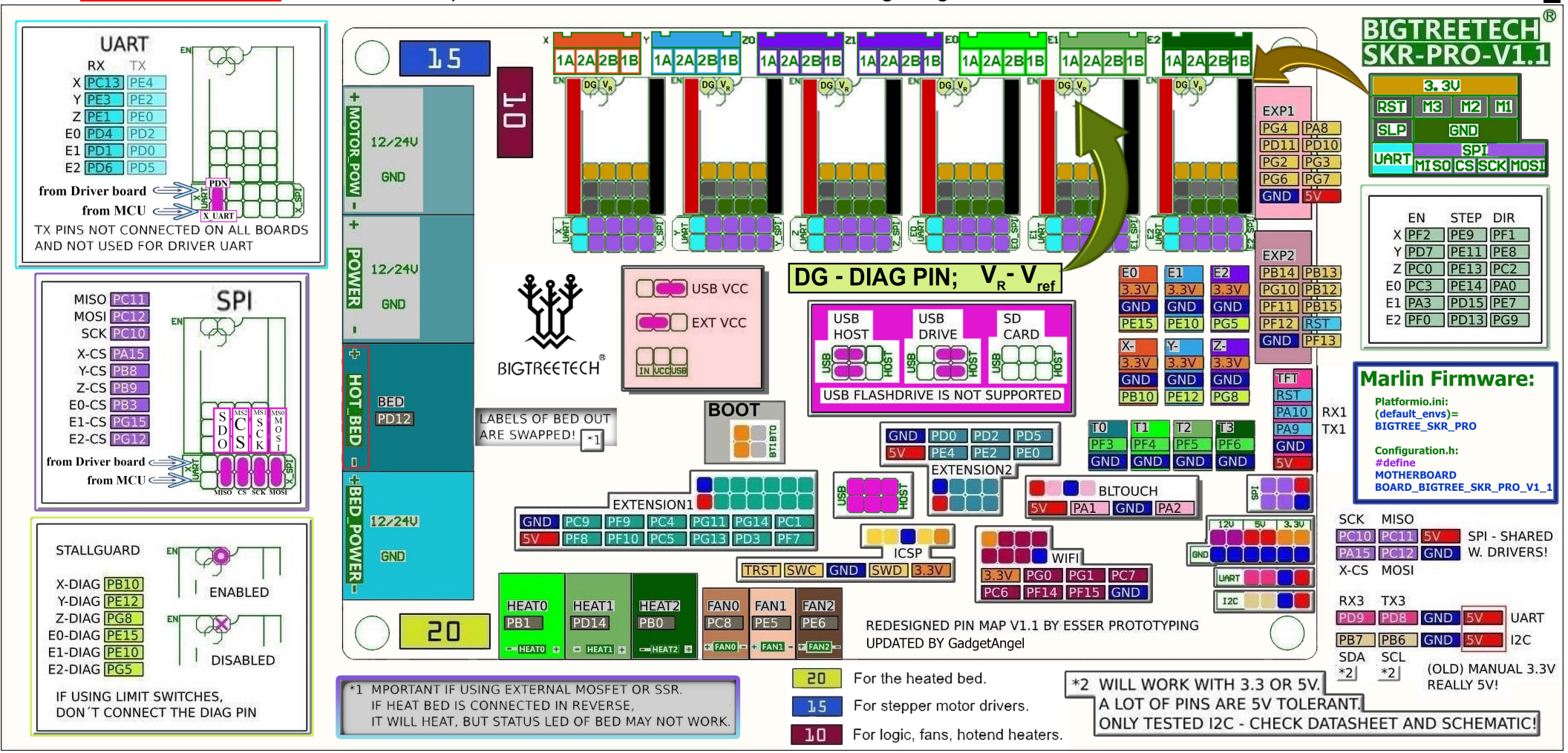




- Go to the next page.

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

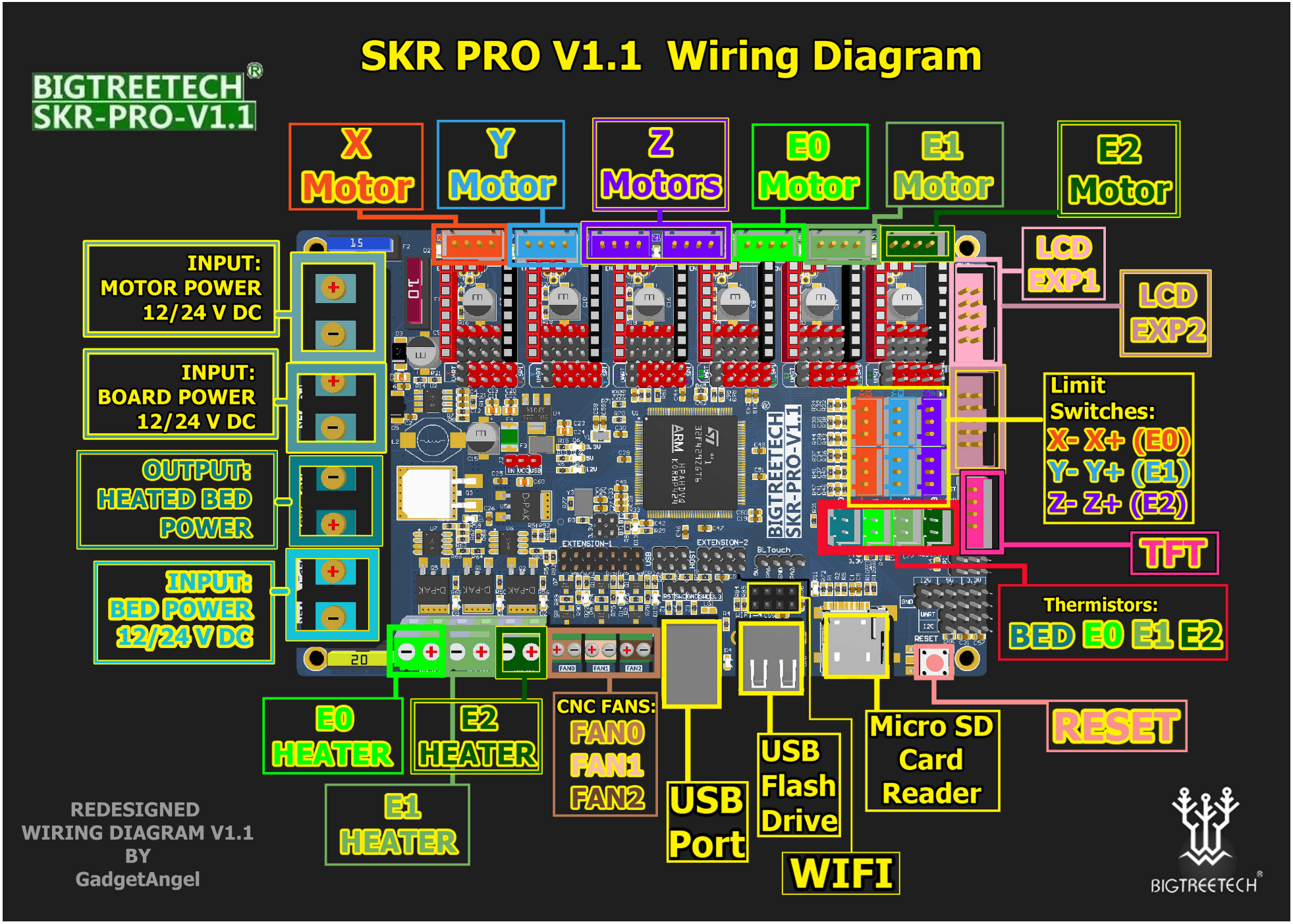
- We want to set the fan on the hot end to automatically turn on when the hot end starts to get hot. In the pins_BTT_SKR_PRO_V1_1.h file you will find all the pins that are defined for the board. They should all conform to the SKR PRO V1.1 Pin diagram shown below. See the next page for a functional picture of the SKR PRO V1.1 Color Wiring Diagram. [3]



- We want to copy and paste the Marlin name or the actual pin number of where you hooked up the fan for your hot end and use that to set "E0_AUTO_FAN_PIN" in the Configuration_adv.h file. I am connecting FAN1 port to my electronics case fan. I am connecting FAN0 port to my part or print cooling fan and connecting FAN2 port to my dual 5015 hot end cooling fans. But we are only interested in how to set the hot end cooling fan up. So, in this example I will use FAN2 port for my hot end cooling fan. In pins_BTT_SKR_PRO_V1_1.h file we see FAN2 port is defined as PE6. So you can choose to copy "PE6" or "FAN2_PIN". Go to next page.

[3] Color PIN Diagram is done by **Thomas White** and updated by **GadgetAngel**

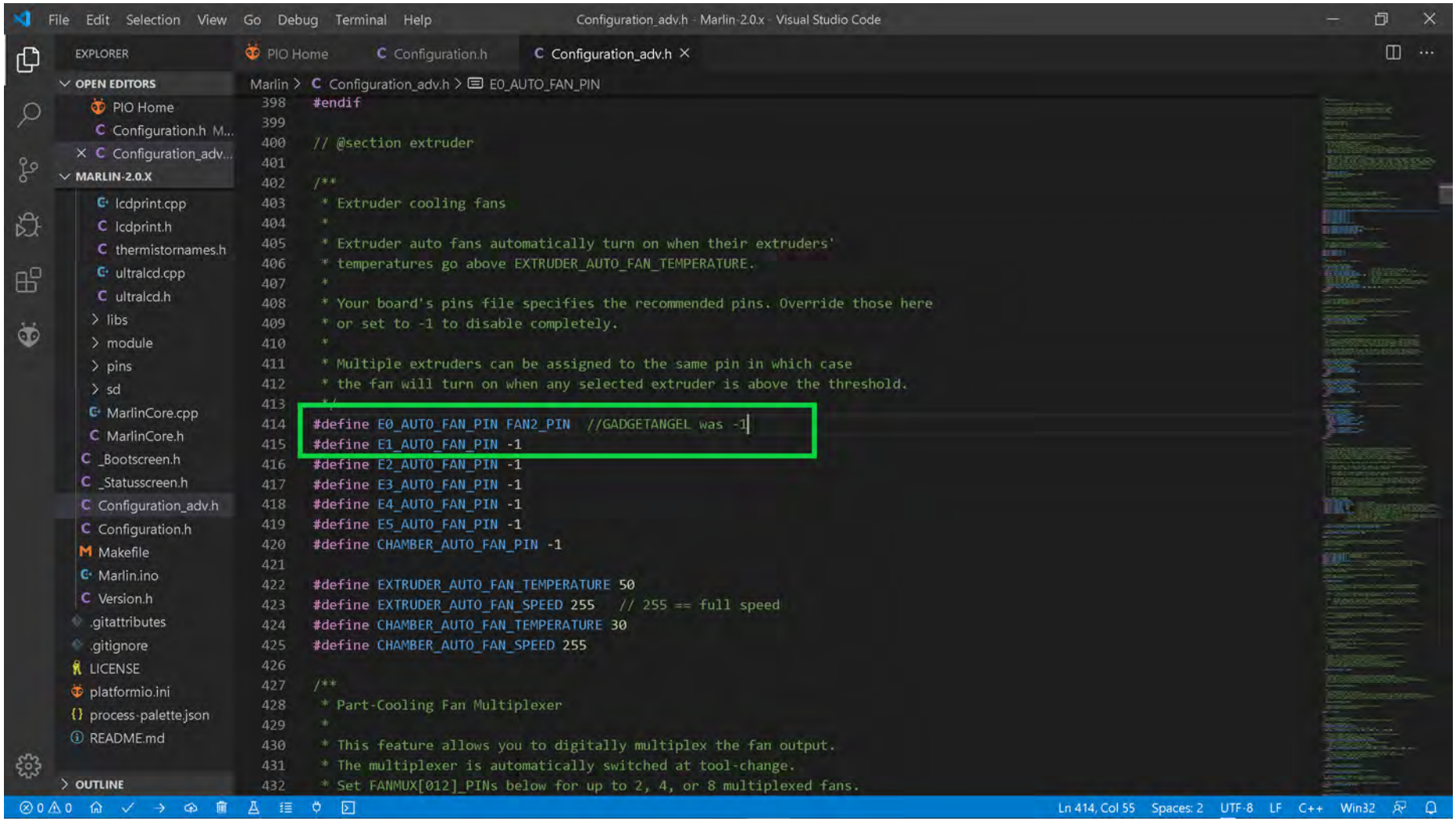
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers



- **Go to the next page.**

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

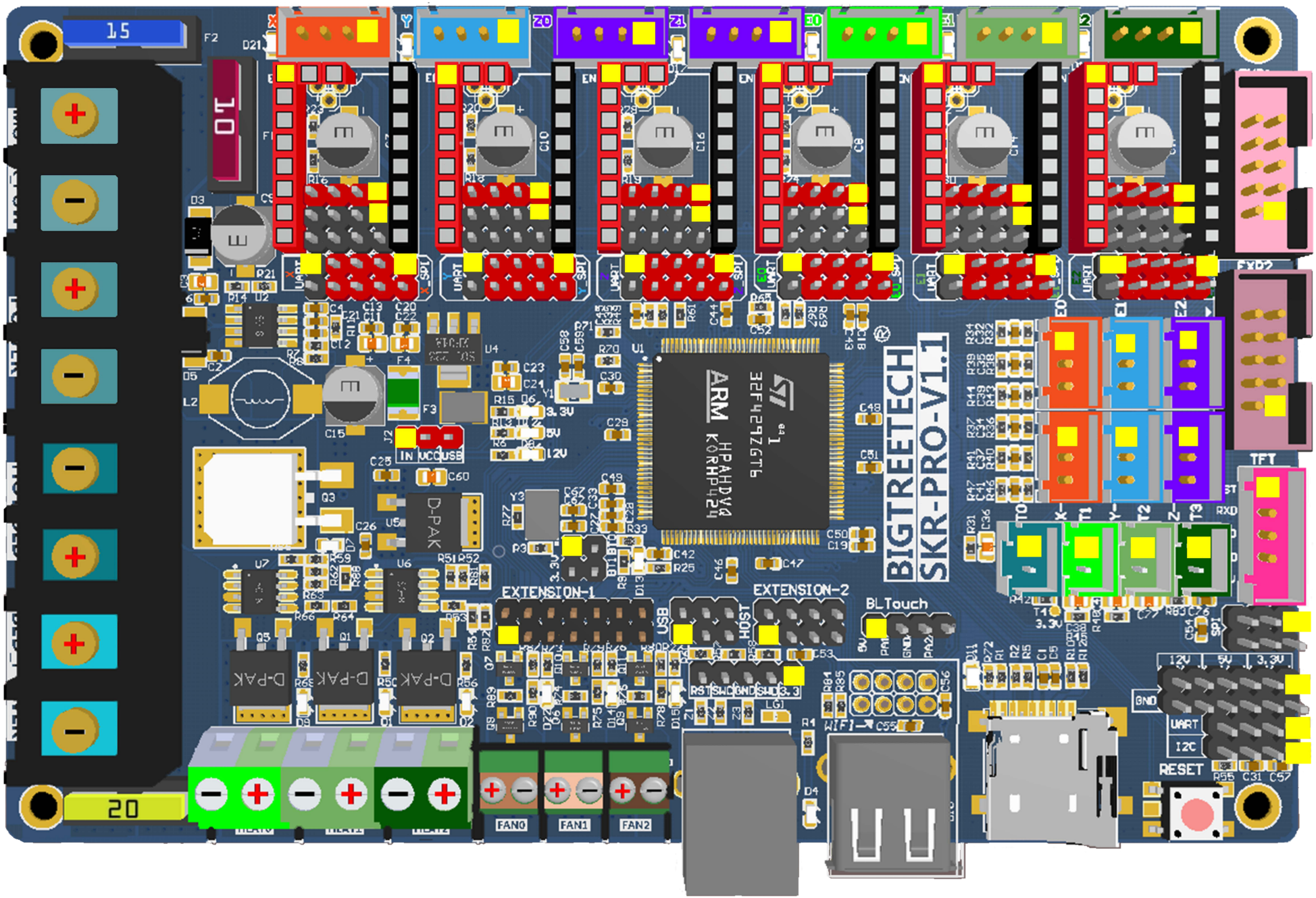
- To set the hot end fan, I will use FAN2_PIN and set "E0_AUTO_FAN_PIN" in the Configuration_adv.h file to FAN2_PIN, as seen in the picture below



- To see more Marlin setup for the latest release, **please refer to the stepper motor driver section of this document for the stepper motor driver of your choice.**

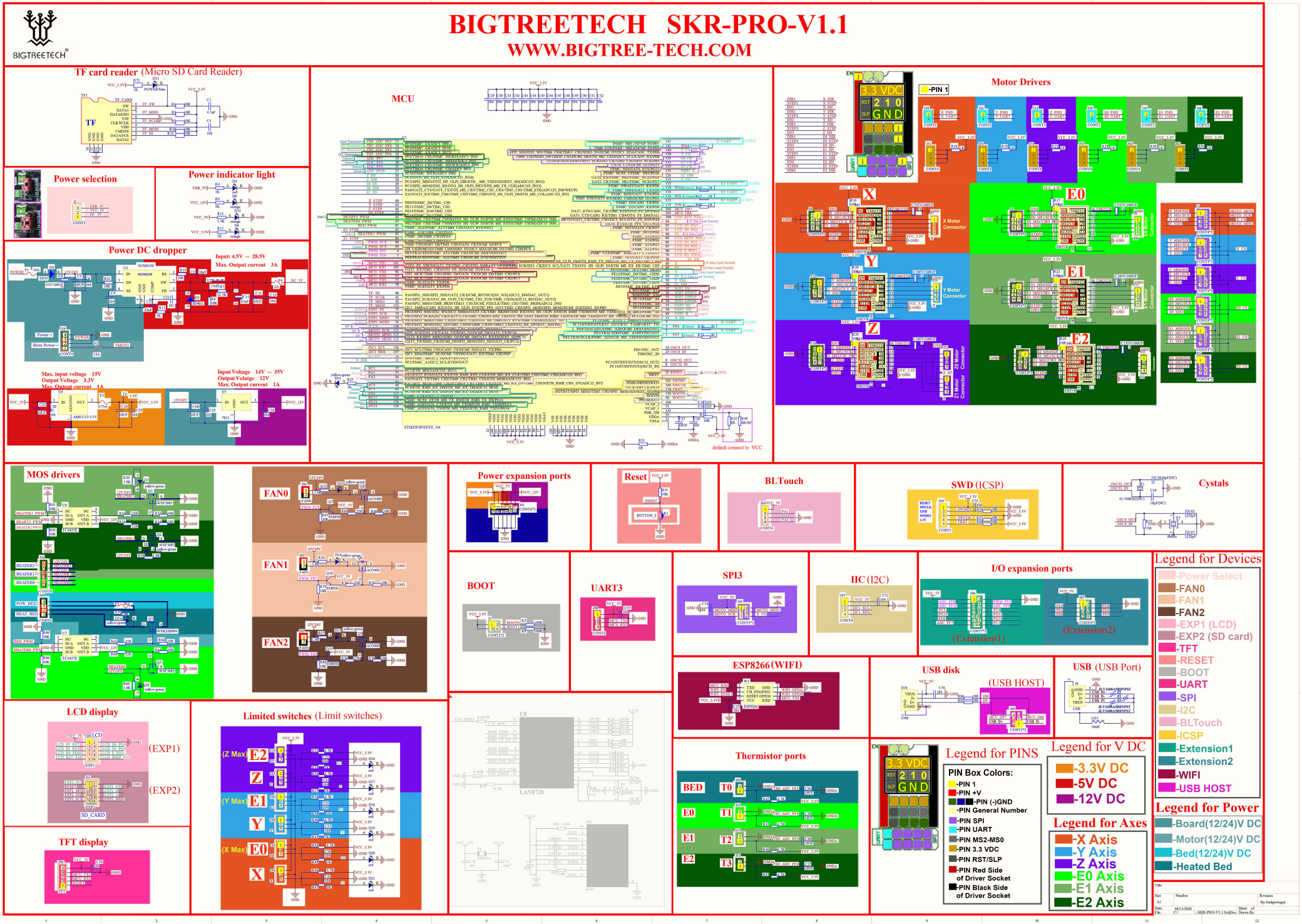
- **To see my updated PIN 1 diagram for SKR PRO V1.1, go to the next page.**

APPENDIX C - Color PIN 1 Diagram



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

APPENDIX C - Color Schematic Diagram



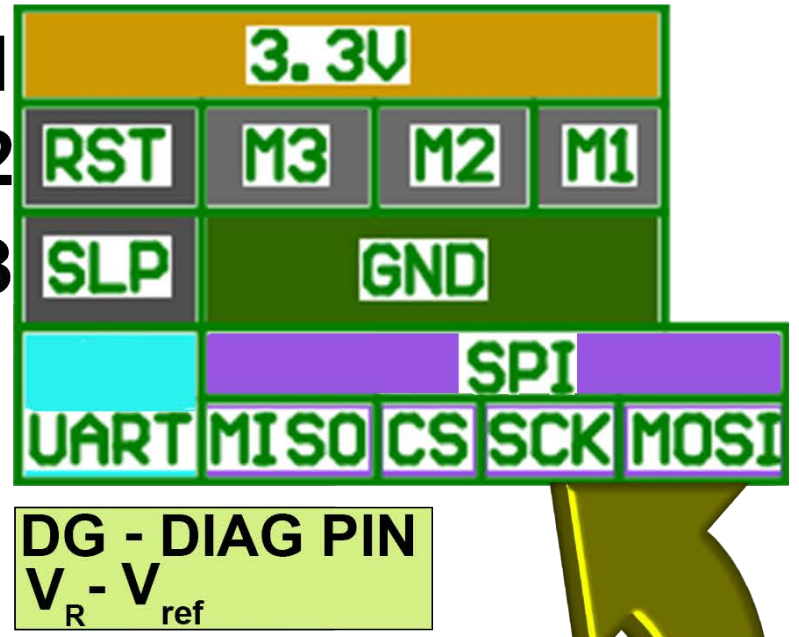
APPENDIX D

Legends for SKR PRO V1.1 Stepper Driver Socket Representations

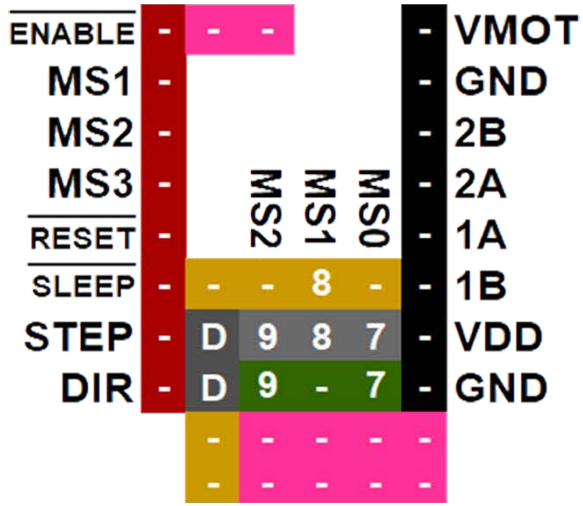
Row 1 is for 3.3 VDC PINS

Row 2 is for RST and MS2-MS0 PINS

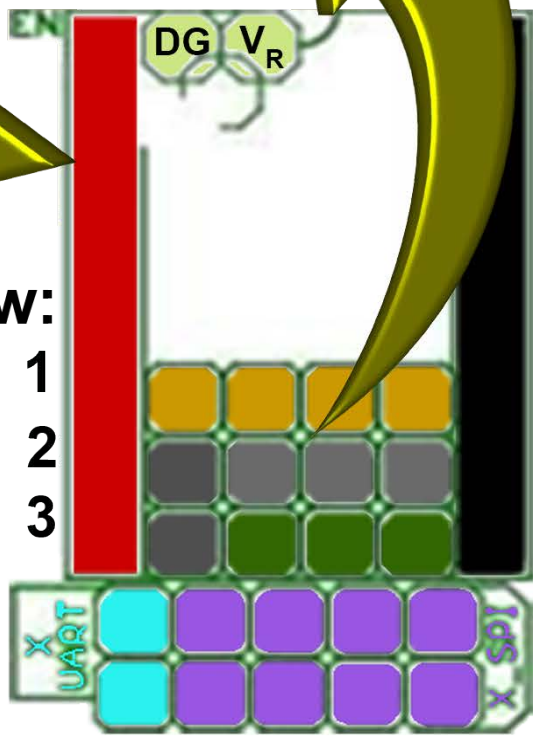
Row 3 is for SLP and GND PINS



NOTICE: PIN labels are relative to stepper driver chip's carrier board, not the SKR PRO V1.1 controller board. **Double check driver compatibility before use**. Numbered pairs denote **required jumper(s)**. The example below indicates that you must place ONE jumper (8) across row 1 and 2 in MS1 column; ONE jumper (D) across row 2 and 3 in RST/SLP column; ONE jumper (9) across row 2and 3 in MS2 column; and ONE jumper (7) across row 2 and 3 in MS0 column.



DG - DIAG PIN
V_R - V_{ref}



Legend for MS0-MS2 PINS:

2 - MS2
1 - MS1
0 - MS0

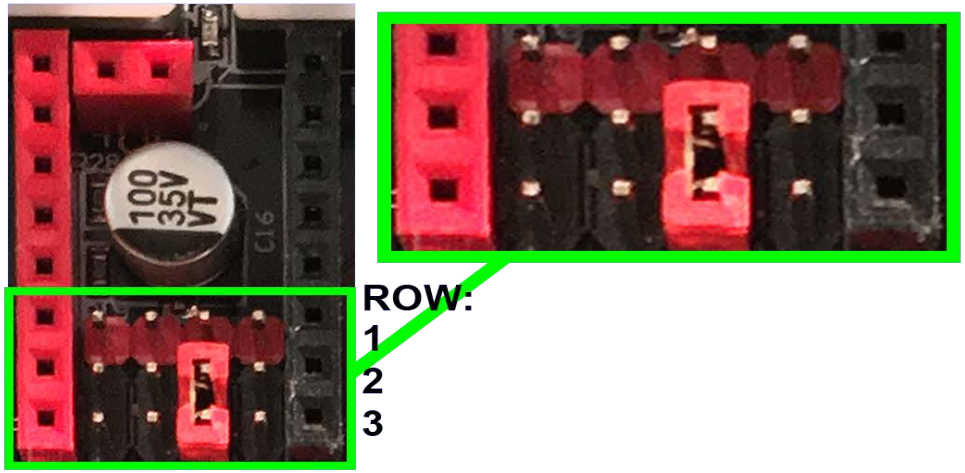
Legend for Socket:

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

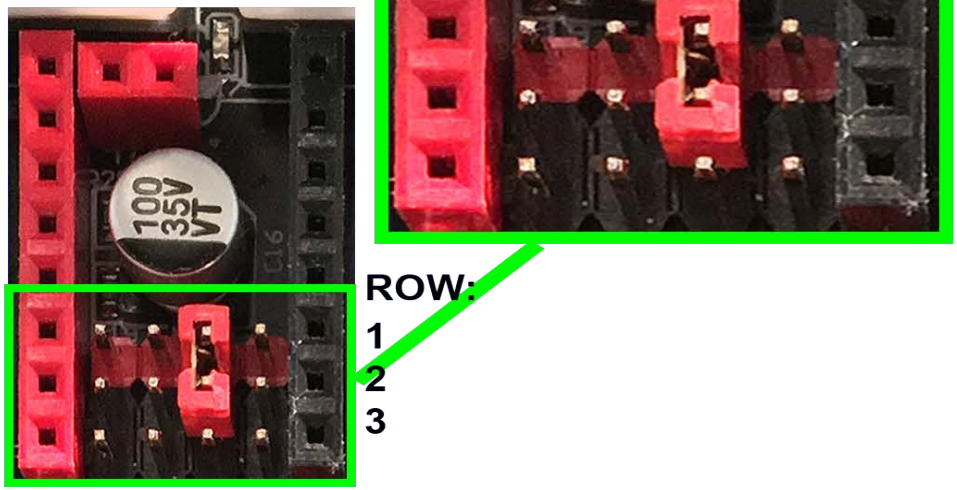
- Go to the next page.

Legends for SKR PRO V1.1 Stepper Driver Socket Representations

Low:



High:



9
9 ⇒ **Jumper Set**

Low ⇒ set Jumper between rows 2 and 3

High ⇒ set Jumper between rows 1 and 2

8
8 ⇒ **Jumper Set**

Low ⇒ set Jumper between rows 2 and 3

High ⇒ set Jumper between rows 1 and 2

7
7 ⇒ **Jumper Set**

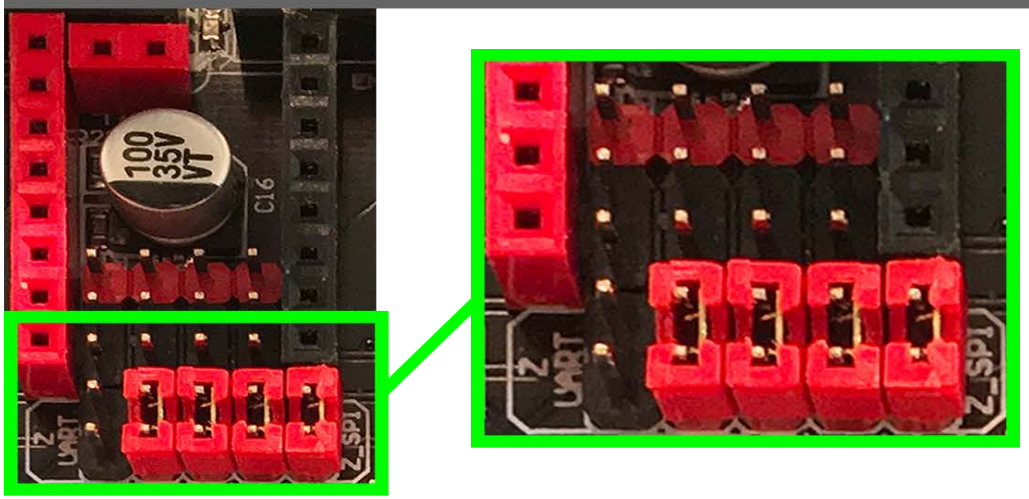
Low ⇒ set Jumper between rows 2 and 3

High ⇒ set Jumper between rows 1 and 2

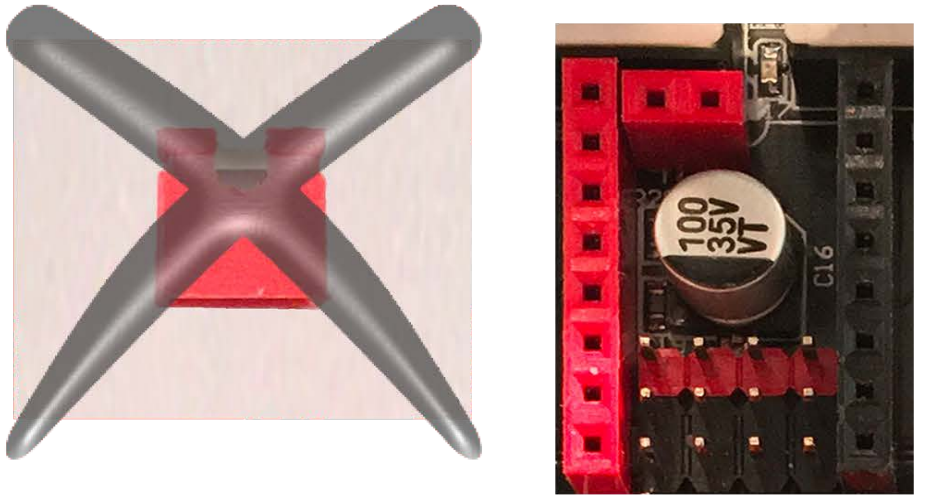
- Go to the next page.

Legends for SKR PRO V1.1 Stepper Driver Socket Representations

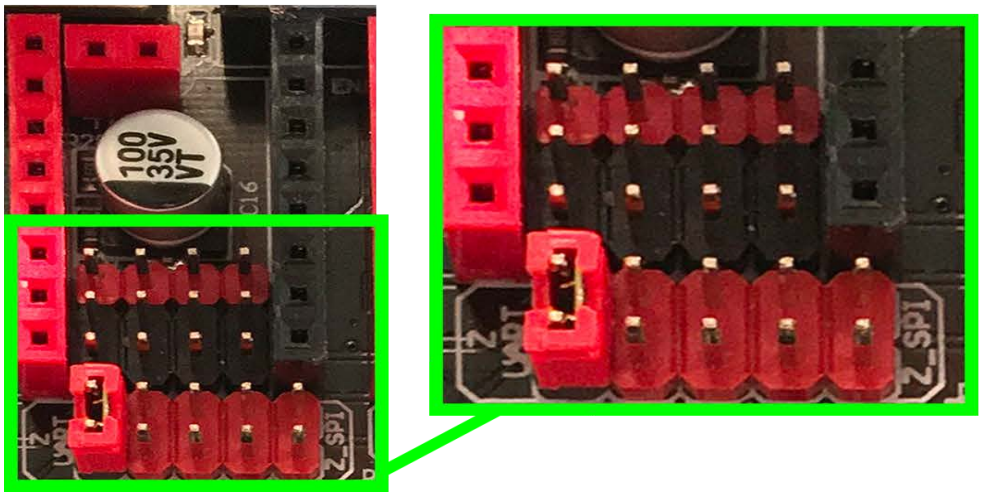
SPIM ⇒ Jumper Set



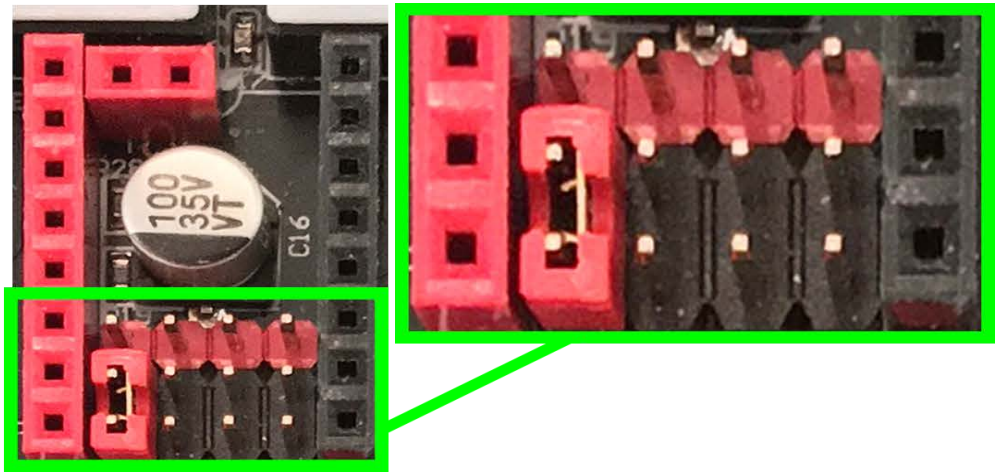
- ⇒ No Jumper



U U ⇒ Jumper Set



D D ⇒ Jumper Set

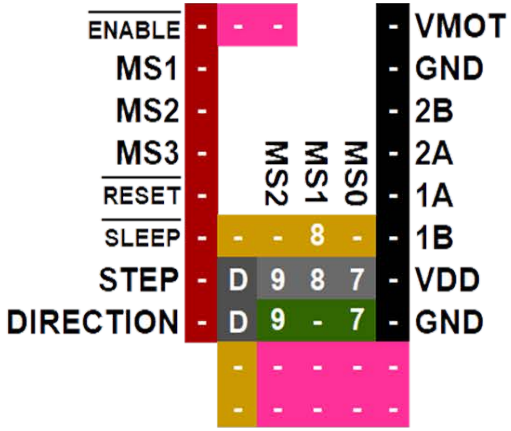


- Go to the next page.

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

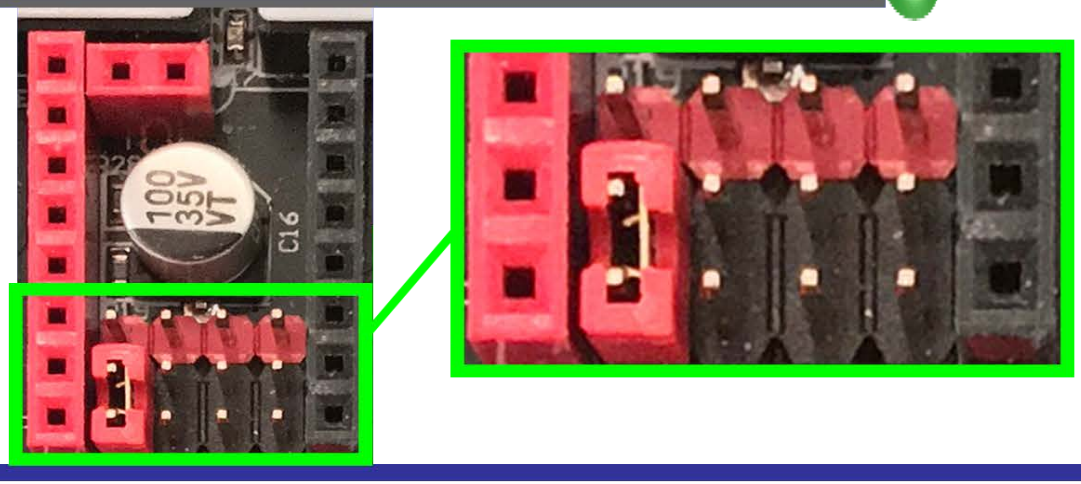
Driver Socket Representation for D Jumper:

1 / 4



The Right Way to Set the D Jumper:

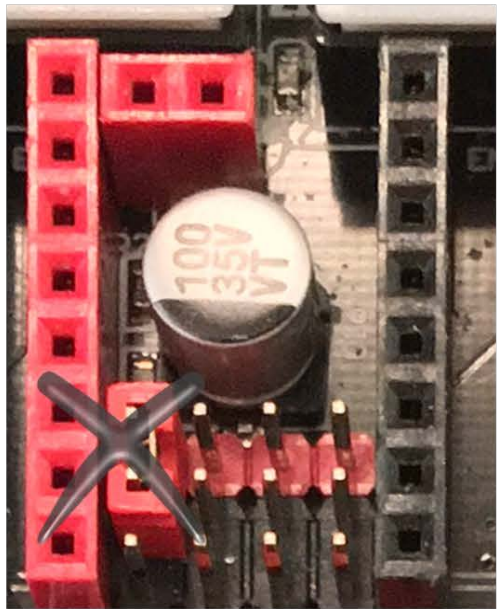
D D ⇒ Jumper Set



The Wrong Way to Set the D Jumper:

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

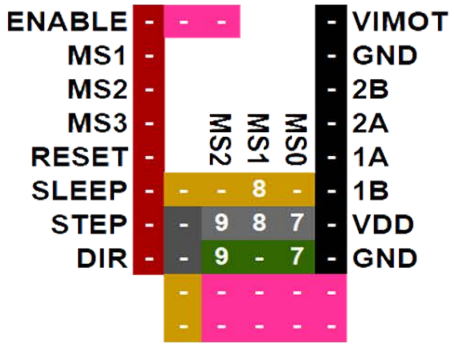
ROW:
1
2
3



- Go to the next page.

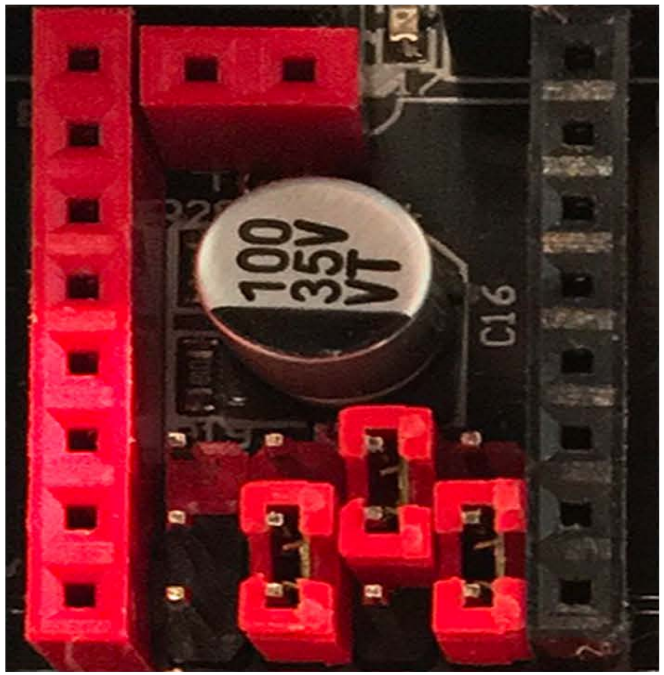
Examples for Stepper Driver Socket Representations

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



1 / 4

M S 2 | M S 1 | M S 0



Row:
1
2
3

9 / 9 ⇒ Jumper Set

MS2 PIN:

Low ⇒ set Jumper between rows 2 and 3

8 / 8 ⇒ Jumper Set

MS1 PIN:

High ⇒ set Jumper between rows 1 and 2

7 / 7 ⇒ Jumper Set

MS0 PIN:

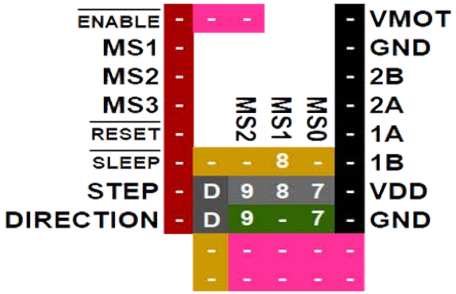
Low ⇒ set Jumper between rows 2 and 3

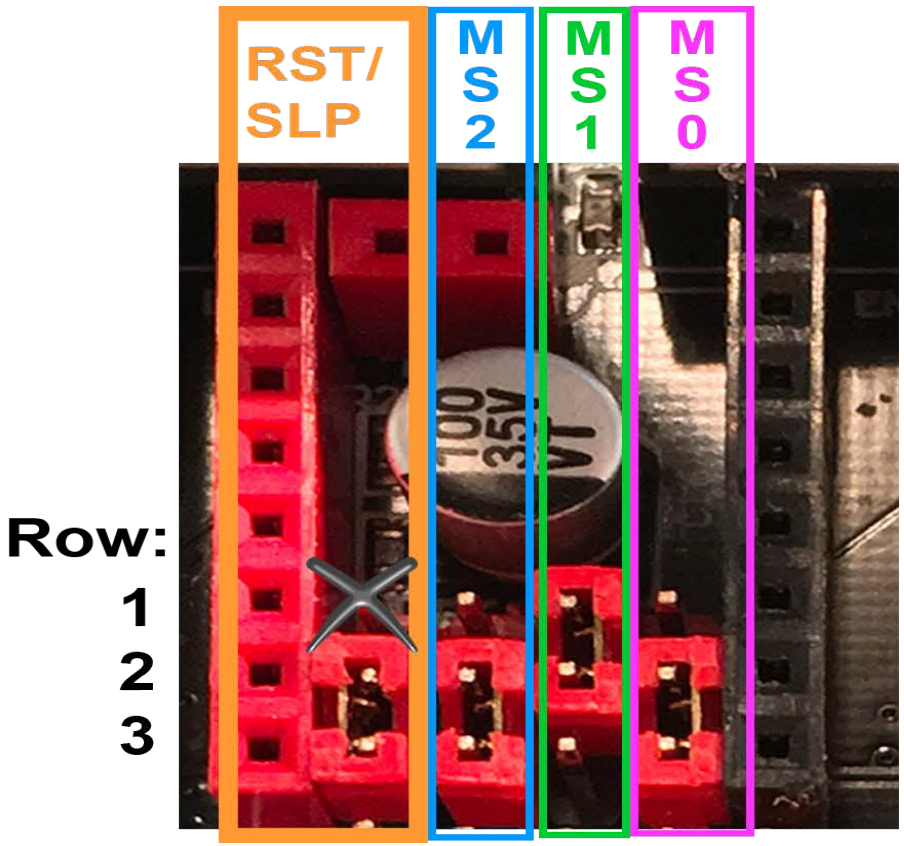
- Go to the next page.

Examples for Stepper Driver Socket Representations

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

1 / 4





RST
SLP ⇨ Jumper Set

RST & SLP PINS:

RST/SLP ⇨ set Jumper between rows 2 and 3

9
9 ⇨ Jumper Set

MS2 PIN:

Low ⇨ set Jumper between rows 2 and 3

8
8 ⇨ Jumper Set

MS1 PIN:

High ⇨ set Jumper between rows 1 and 2

7
7 ⇨ Jumper Set

MS0 PIN:

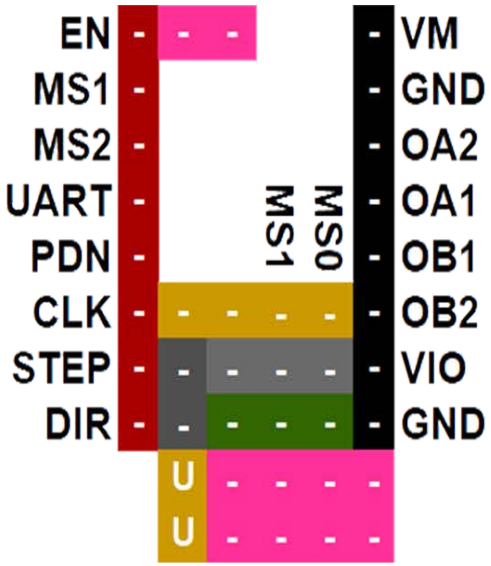
Low ⇨ set Jumper between rows 2 and 3

- Go to the next page.

Examples for Stepper Driver Socket Representations

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

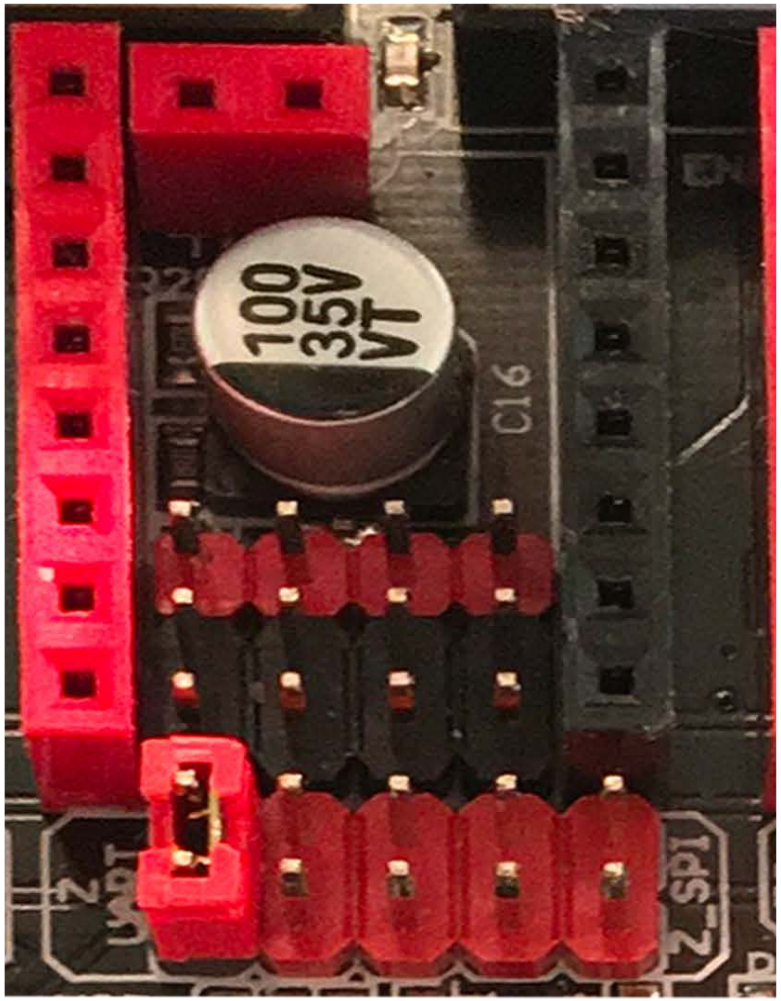
UART



RST/SLP MS2 MS1 MS0

Row:

1
2
3
4
5



U
U ⇨ Jumper Set

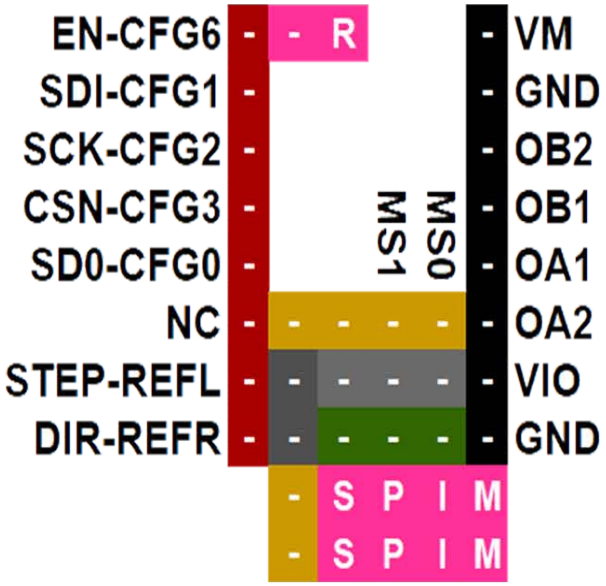
U ⇨ set Jumper between rows 4 and 5

- Go to the next page.

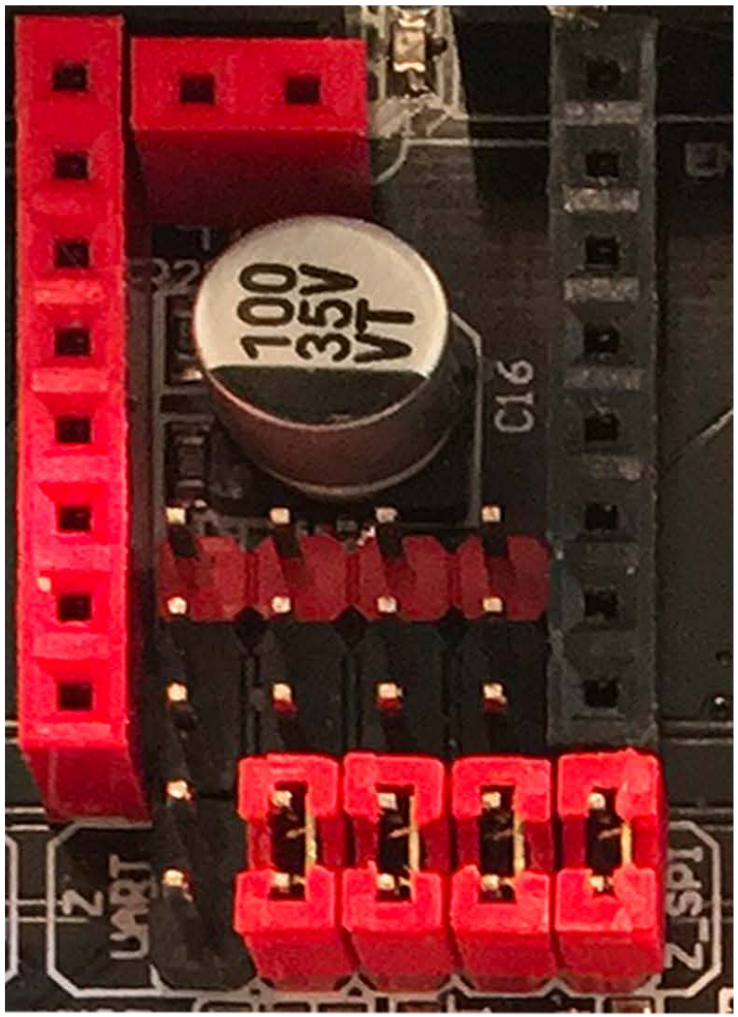
Examples for Stepper Driver Socket Representations

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

SPI



RST/SLP MS2 MS1 MS0



Row:
1
2
3
4
5

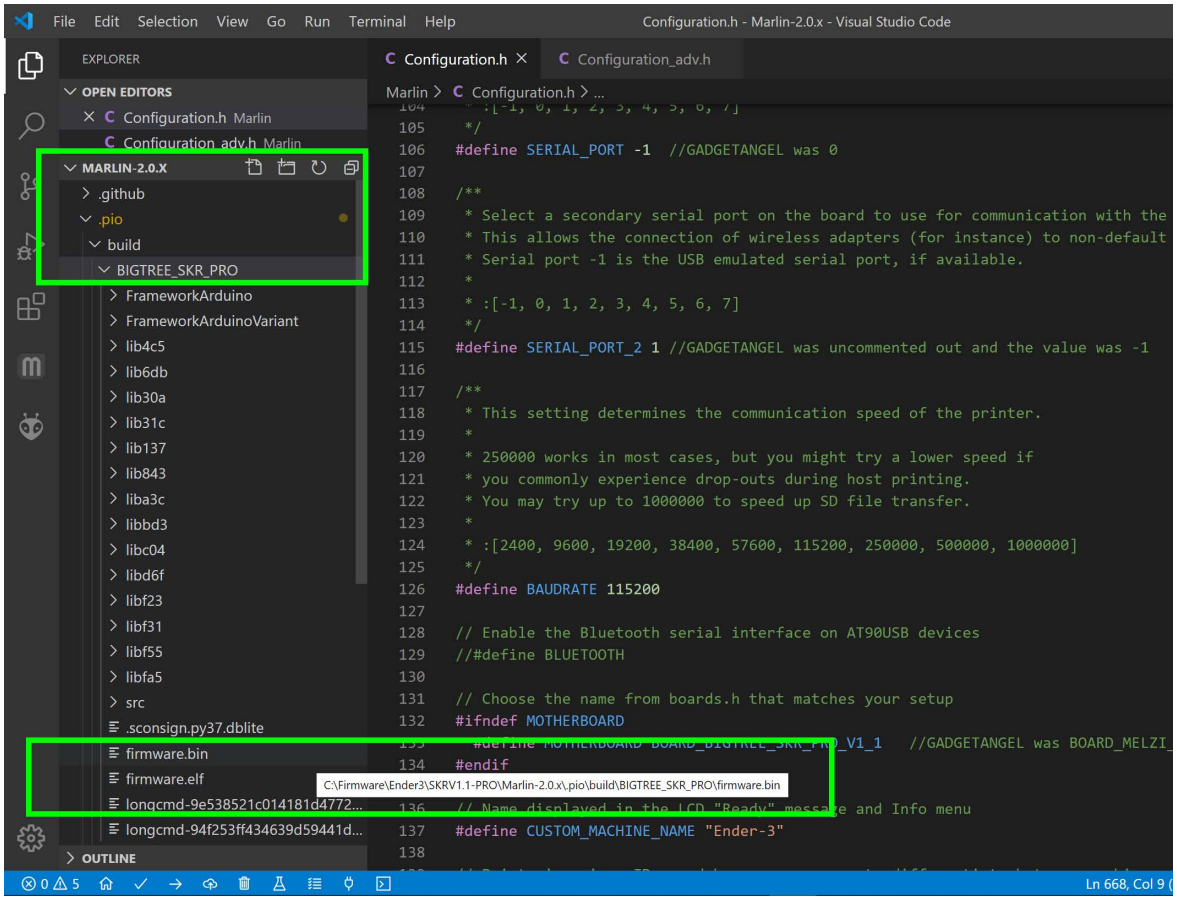
SPIM ⇨ Jumper Set

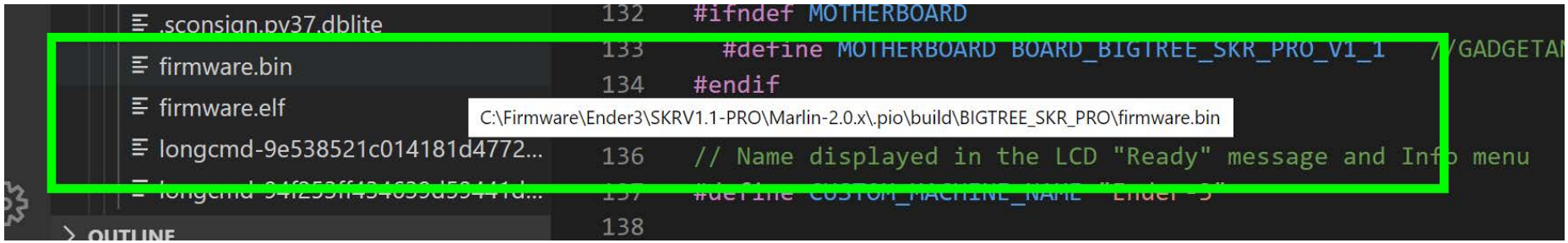
SPIM
SPIM ⇨ set Jumpers between rows 4 and 5

APPENDIX E

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

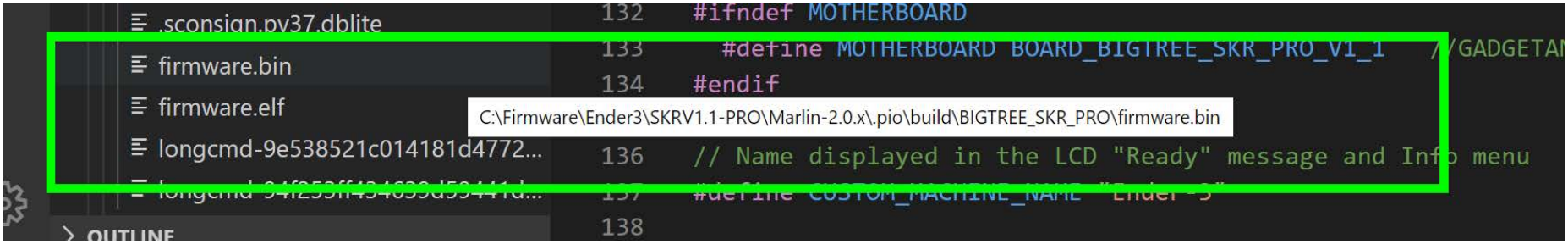
- The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X\.pio\build\BIGTREE_SKR_PRO\ and right clicking on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Window's machine open a file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

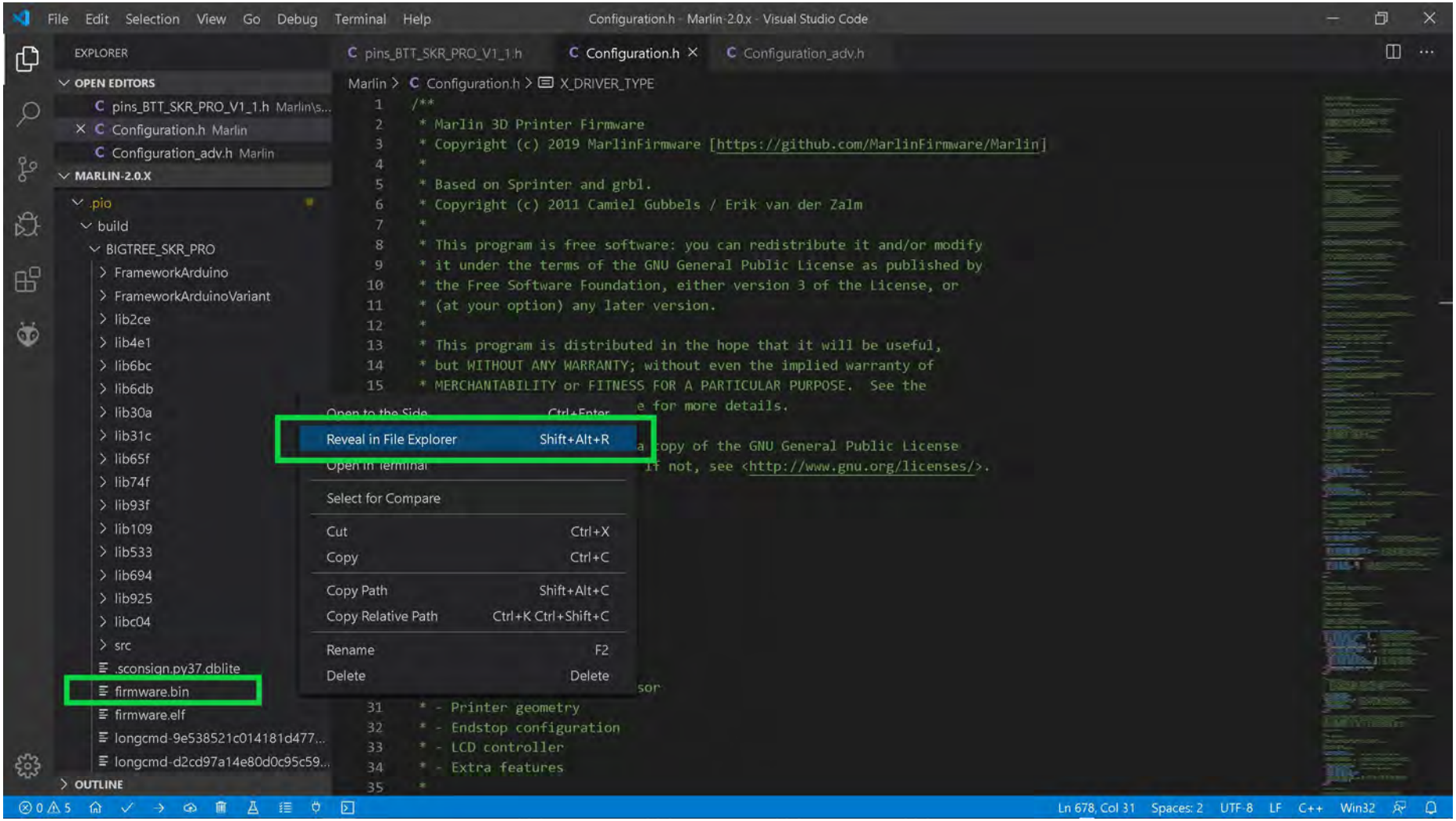




- Go to the next page.

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

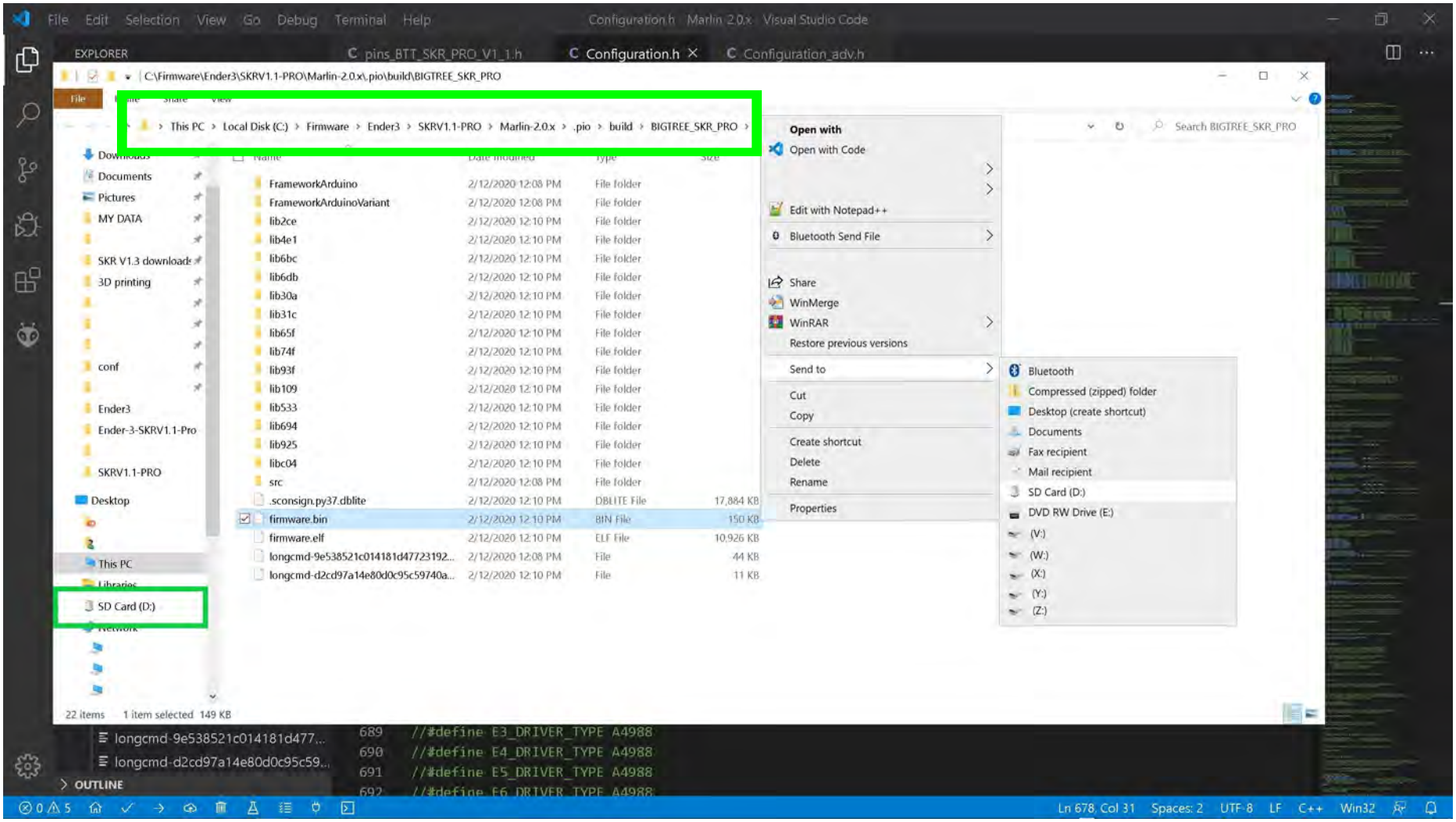




- Go to the next page.

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

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



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

APPENDIX F
Links to Reference Material

Marlin Firmware Documentation

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

Information on Stepper Motor Drivers and Micro-stepping

- https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping

- Go to the next page.

APPENDIX F
Links to Reference Material

POLOLU A4988 and BIQU A4988

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

DRV8825

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

- Go to the next page.

APPENDIX F
Links to Reference Material

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

- https://www.youtube.com/watch?v=lDljLaXWg4E

BIQU LV8729

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

- Go to the next page.

APPENDIX F
Links to Reference Material

FYSETC LV8729

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

LERDGE LV8729

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

- Go to the next page.

APPENDIX F
Links to Reference Material

MKS LV8729

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

FYSETC S6128 V1.1

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

- Go to the next page.

APPENDIX F
Links to Reference Material

FYSETC ST820

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

BIQU ST820

- https://bigtreetech.en.alibaba.com/product/60873031205-806048513/BIGTREETECH_ST820_V1_0_Step_Stepper_Motor_Driver_Microstepping_Kit_MKS_GEN_V1_4_MKS_GEN_L_SKR_V1_1_Ramps_1_4.html
- https://www.youtube.com/watch?v=HNciZ31GAro

- https://github.com/bigtreetech/BIGTREETECH-ST820-V1.0

- Go to the next page.

APPENDIX F
Links to Reference Material

POLOLU ST820 (STSPIN820)

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

POLOLU MP6500

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

- Go to the next page.

APPENDIX F
Links to Reference Material

POLOLU TB67S249FTG

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

BIQU S109 and FYSETC S109

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

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU S109

- https://www.biqu.equipment/products/bigtreetech-s109-v1-0-stepper-motor-driver

FYSETC S109

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

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

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU TMC2100 and MKS TMC2100

- https://learn.watterott.com/silentstepstick/pinconfig/tmc2100/
- https://reprap.org/wiki/TMC2100
- https://www.youtube.com/watch?v=H41hIXdB6js&t=869s
- https://wiki.fysetc.com/TMC2100/
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2100_datasheet_Rev1.10.pdf
- https://www.biqu.equipment/products/biqu-bigtreetech-tmc2100-2208-2130-v1-0-stepstick-mks-stepper-motor-driver-ultra-silent

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU TMC2130

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

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

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

- Go to the next page.

APPENDIX F
Links to Reference Material

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

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

BIQU TMC2208 V3.0

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

FYSETC TMC2208 V1.2

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

- Go to the next page.

APPENDIX F
Links to Reference Material

Information Common to TMC2208 and BIQU TMC2225

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

BIQU TMC2225 V1.0

- https://github.com/bigtreetech/BIGTREETECH-TMC2225-V1.0
- https://github.com/bigtreetech/BIGTREETECH-TMC2225-V1.0/blob/master/TMC2225%20V1.0%20manual.pdf
- http://img.ekeic.com/ekeic/9981564407175.pdf

- Go to the next page.

APPENDIX F
Links to Reference Material

BIQU TMC2209 V1.2

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

BIQU TMC5160 V1.2

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

- Go to the next page.

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
- https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC5161A-datasheet_Rev1.05.pdf

SKR PRO V1.1 Board

- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/Schematic
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/Virtual%20COM%20Port%20driver
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/firmware
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/manual
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/models
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/tree/master/picture
- https://github.com/bigtreetech/BIGTREETECH-SKR-PRO-V1.1/issues

- Go to the next page.

APPENDIX F
Links to Reference Material

Facebook Groups

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

- Go to the next page.

APPENDIX F
Links to Reference Material

Miscellaneous Information

- https://youtu.be/8wk1eIugv5A
- https://reprap.org/wiki/Stepper_wiring
- https://learn.watterott.com/silentstepstick/configurator/
- https://wiki.fysetc.com/TMC2208/#to-run-the-program
- https://marlinfw.org/meta/gcode/
- https://blog.prusaprinters.org/calculator/
- https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers
- https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers
- https://marlinfw.org/docs/basics/install_platformio.html
- https://marlinfw.org/meta/download/
- https://www.pronterface.com/
- https://www.klipper3d.org/Sensorless_Homing.html
- https://learn.watterott.com/silentstepstick/comparison/
- https://reprap.org/wiki/Category:Reference
- https://reprap.org/wiki/Calibration