

Stepper Driver Configurations for SKR V1.3 Board



By
@GadgetAngel



Based on Work By @rflulling

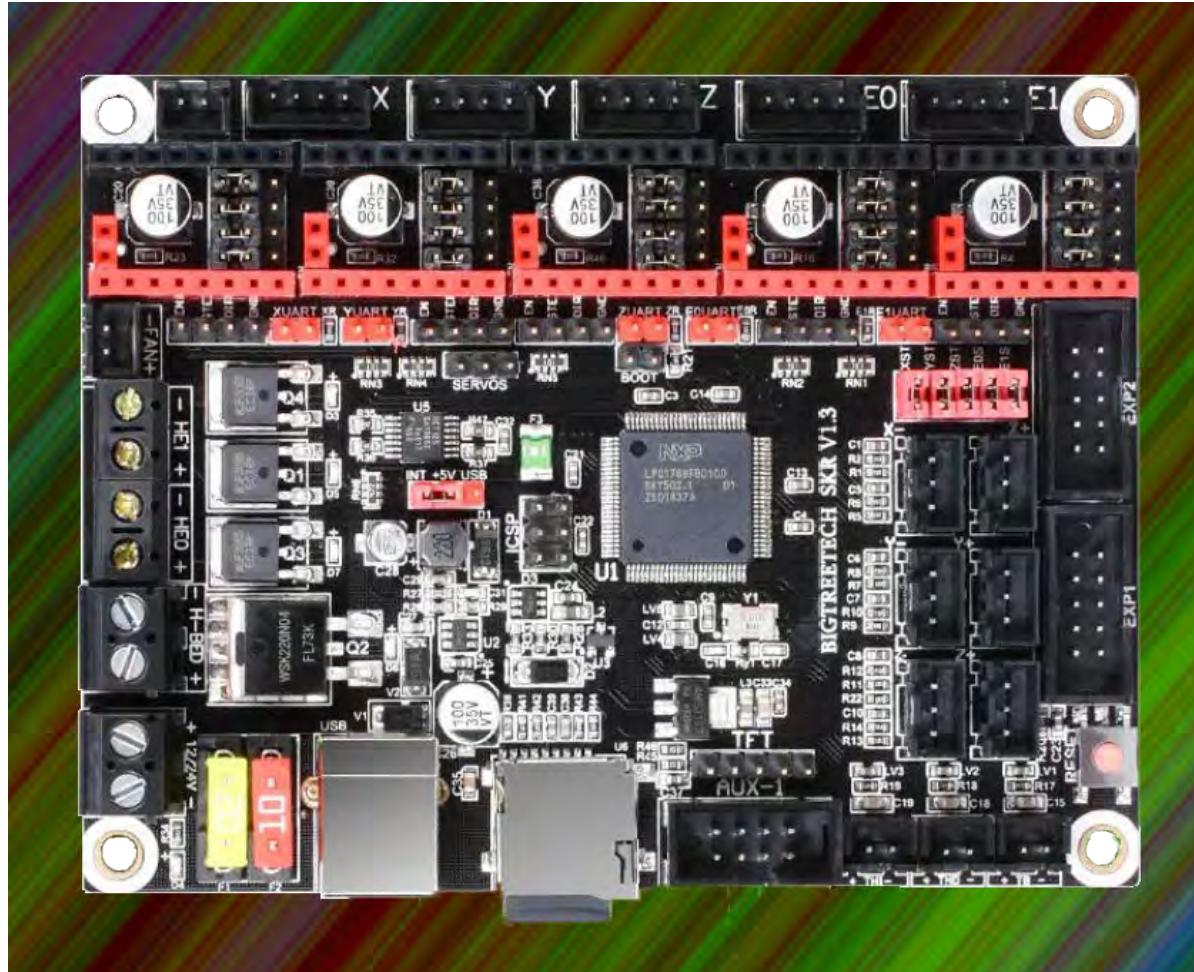


Table of Contents

Preface	1
What is a stepper motor?	2
Stall Detection and Sensor-less Homing	3
Requirements Needed to Make Sensor-less Homing Work	3
POLOLU A4988	4
Driver Chip Chart for POLOLU A4988	4
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	5
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	8
The (latest release of) Marlin Setup for POLOLU A4988 Drivers	13
BIQU A4988	16
Driver Chip Chart for BIQU 4988.....	16
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	17
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	20
The (latest release of) Marlin Setup for BIQU A4988 Drivers	25
DRV8825	28
Driver Chip Chart for POLOLU DRV8825	28
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	29
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	32
The (latest release of) Marlin Setup for DRV8825 Drivers	38
BIQU LV8729	42
Driver Chip Chart for BIQU LV8729	42
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	43
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	46
The (latest release of) Marlin Setup for BIQU LV8729 Drivers	52
FYSETC LV8729	57
Driver Chip Chart for FYSETC LV8729	57

Table of Contents

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	58
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	61
The (latest release of) Marlin Setup for FYSETC LV8729 Drivers.....	67
LERDGE LV8729	72
Driver Chip Chart for LERDGE LV8729	72
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	73
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	76
The (latest release of) Marlin Setup for LERDGE LV8729 Drivers	82
MKS LV8729	87
Driver Chip Chart for MKS LV8729	87
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	88
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	91
The (latest release of) Marlin Setup for MKS LV8729 Drivers	97
FYSETC S6128 V1.1.....	102
Driver Chip Chart for FYSETC S6128	102
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	103
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	106
The (latest release of) Marlin Setup for FYSETC S6128 V1.1 Drivers.....	112
FYSETC ST820.....	117
Driver Chip Chart for FYSETC ST820	117
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	118
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	121
The (latest release of) Marlin Setup for FYSETC ST820 Drivers.....	127
BIQU ST820	133
Driver Chip Chart for BIQU ST820	133
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	134

Table of Contents

SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	137
The (latest release of) Marlin Setup for BIQU ST820 Drivers	143
POLOLU ST820 (STSPIN820)	149
Driver Chip Chart for POLOLU ST820	149
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	150
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	153
The (latest release of) Marlin Setup for POLOLU ST820 (STSPIN820) Drivers.....	159
POLOLU MP6500	165
Driver Chip Chart for POLOLU MP6500	165
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	166
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	169
The (latest release of) Marlin Setup for POLOLU MP6500 Drivers	173
POLOLU TB67S249FTG.....	178
Driver Chip Chart for POLOLU TB67S249FTG	178
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	179
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	182
The (latest release of) Marlin Setup for POLOLU TB67S249FTG Drivers	188
BIQU S109	193
Driver Chip Chart for BIQU S109	193
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	194
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	197
The (latest release of) Marlin Setup for BIQU S109 Drivers	203
FYSETC S109	208
Driver Chip Chart for FYSETC S109	208
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	209
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	212

Table of Contents

The (latest release of) Marlin Setup for FYSETC S109 Drivers	218
BIQU TMC2100 Stand-alone Mode	223
Driver Chip Chart for Biqu TMC2100 in Stand-alone Mode.....	223
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers	224
How to Create aa SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers	226
SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers.....	229
Additional Equipment Needed for Low Low (STEP or FULL) Configuration.....	231
The (latest release of) Marlin Setup for Biqu TMC2100 Drivers in Stand-alone Mode.....	239
MKS TMC2100 Stand-alone Mode	244
Driver Chip Chart for MKS TMC2100 in Stand-alone Mode.....	244
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers	245
How to Create a SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers	247
SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers.....	250
Additional Equipment Needed for Low Low (STEP or FULL) Configuration.....	252
The (latest release of) Marlin Setup for MKS TMC2100 Drivers in Stand-alone Mode.....	260
BIQU TMC2130 Stand-alone Mode	265
Driver Chip Chart for Biqu TMC2130 in Stand-alone Mode.....	265
SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers	266
How to Create a SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers	268
SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers.....	271
Additional Equipment Needed for Low Low (STEP or FULL) Configuration.....	273
The (latest release of) Marlin Setup for Biqu TMC2130 Drivers in Stand-alone Mode.....	281
BIQU TMC2130 SPI Mode	286
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode.....	291
Information on Sensor-less Homing	294
Examples of Different SPI Configurations	299
The (latest release of) Marlin Setup for Biqu TMC2130 Drivers in SPI Mode	301

Table of Contents

BIQU TMC2208 V3.0 Stand-alone Mode	314
Driver Chip Chart for BIQUE TMC2208 in Stand-alone Mode.....	314
SKR V1.3 LEGEND of Driver Chip for Binary State Stepper Drivers	315
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	318
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in Stand-alone Mode	322
BIQU TMC2208 V3.0 One Time Programming (OTP) Mode.....	327
Driver Chip Chart for BIQUE TMC2208 in OTP Mode	327
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	328
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	331
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode	335
BIQU TMC2208 V3.0 UART Mode.....	340
Driver Chip Chart for BIQUE TMC2208 in UART Mode	340
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode	344
Examples of Different UART Configurations	348
The (latest release of) Marlin Setup for BIQUE TMC2208 V3.0 Drivers in UART Mode.....	349
FYSETC TMC2208 V1.2 Stand-alone Mode	359
Driver Chip Chart for FYSETC TMC2208 in Stand-alone Mode.....	359
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	360
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	363
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode	367
FYSETC TMC2208 V1.2 One Time Programming (OTP) Mode.....	372
Driver Chip Chart for FYSETC TMC2208 in OTP Mode.....	372
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	373
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	376
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode	380
FYSETC TMC2208 V1.2 UART Mode	385

Table of Contents

Driver Chip Chart for FYSETC TMC2208 in UART Mode	385
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode	389
Examples of Different UART Configurations	393
The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode.....	394
BIQU TMC2225 V1.0 Stand-alone Mode	404
Driver Chip Chart for BIQU TMC2225 in Stand-alone Mode.....	404
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	405
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	408
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode	412
BIQU TMC2225 V1.0 One Time Programming (OTP) Mode.....	417
Driver Chip Chart for BIQU TMC2225 in OTP Mode	417
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	418
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	421
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode	425
BIQU TMC2225 V1.0 UART Mode.....	430
Driver Chip Chart for BIQU TMC2225 in UART Mode	430
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode	434
Examples of Different UART Configurations	438
The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode.....	439
BIQU TMC2209 V1.2 Stand-alone Mode for StealthChop.....	449
Driver Chip Chart for BIQU TMC2209 in Stand-alone Mode (StealthChop)	449
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	450
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	453
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop	457
BIQU TMC2209 V1.2 Stand-alone Mode for SpreadCycle	462
Driver Chip Chart for BIQU TMC2209 in Stand-alone Mode (SpreadCycle).....	462

Table of Contents

SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers.....	463
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	466
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle.....	470
BIQU TMC2209 V1.2 UART Mode.....	475
Driver Chip Chart for BIQU TMC2209 in UART Mode	475
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode	480
Information on Sensor-less Homing	482
Examples of Different UART Configuratin	487
The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode.....	489
BIQU TMC5160 V1.2 SPI Mode.....	502
Driver Chip Chart for BIQU TMC5160 in SPI Mode.....	502
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode.....	507
Information on Sensor-less Homing	510
Examples of Different SPI Configurations	515
The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode	517
BIQU TMC5161 V1.0 SPI Mode.....	530
Driver Chip Chart for BIQU TMC5161 in SPI Mode.....	530
SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode.....	535
Information on Sensor-less Homing	538
Examples of Different SPI Configurations	543
The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode	545
APPENDIX A.....	558
How to adjust the V_{ref} on a Stepper Motor Driver board using the Potentiometer.....	558
APPENDIX B.....	560
For the TMC drivers what's the difference between stand-alone mode and ("UART" or "SPI ") modes?.....	560
How to Calculate V_{ref} for Non-TMC Stepper Motor Divers	560

Table of Contents

How to Calculate V_{ref} for TMC Stepper Motor Drivers.....	561
Driving Current Calculation Formulas for TMC Stepper Motor Drivers	562
#1 TMC2100 with R _s = 0.110Ω (110mΩ)	562
#2 TMC2130 with R _s = 0.110Ω (110mΩ)	562
#3 TMC2208 with R _s = 0.110Ω (110mΩ) for Stand-alone Mode	562
#4 TMC2208 with R _s = 0.110Ω (110mΩ) for UART Mode.....	563
#5 TMC2209 with R _s = 0.110Ω (110mΩ) for Stand-alone Mode	563
#6 TMC2209 with R _s = 0.110Ω (110mΩ) for UART Mode.....	564
#7 TMC2225 with R _s = 0.150Ω (150mΩ) for UART Mode.....	564
#8 TMC5160 with R _s = 0.075Ω (75mΩ) for SPI Mode.....	564
#9 TMC5161 with R _s = 0.062Ω (62mΩ) for SPI Mode.....	565
#10 TMC2225 with R _s = 0.150Ω (150mΩ) for Stand-alone Mode	565
APPENDIX C.....	566
The (Latest Release of) Marlin Setup That Is Common to ALL Stepper Motor Drivers	566
Link to <u>BIGTREETECH SKR V1.3 Instruction Manual.pdf</u>	566
Link to <u>BIGTREETECH SKR V1.3 Guide.pdf</u>	566
Link to <u>Download Marlin 2.0.x Firmware Website</u>	566
Link to <u>Pronterface Software</u>	567
Link to <u>How to Calibrate your 3D Printer</u>	567
SKR V1.3 Color PIN Diagram	579
SKR V1.3 Color Wiring Diagram	580
SKR V1.3 Color <u>PIN 1</u> Diagram	583
SKR V1.3 Color Schematic Diagram	584
SKR V1.3 Uncolored Schematic Diagram.....	585
APPENDIX D.....	586
Legends for SKR V1.3 Stepper Driver Socket Representations	586
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers	587
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers	590
SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers.....	591

Table of Contents

SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers.....	593
<i>How to Create a SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers.</i>	595
SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers	596
SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers	599
SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers	602
Examples for Stepper Driver Socket Representations	606
Example 1 (LV8729 Driver Board; Binary State Driver) for SKR V1.3 Driver Socket Representation.....	606
Example 2 (A4988 Driver Board; Binary State Driver) for SKR V1.3 Driver Socket Representation.....	607
Example 3 (TMC2130 Driver in Stand-alone Mode; Tri State Driver) for SKR V1.3 Driver Socket Representation	608
Example 4 (TMC2209 UART with Sensor-less Homing) for SKR V1.3 Driver Socket Representation	609
Example 5 (TMC2209 UART WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation.....	610
Example 6 (TMC2130 SPI with Sensor-less Homing) for SKR V1.3 Driver Socket Representation	611
Example 7 (TMC2130 SPI WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation.....	612
APPENDIX E.....	613
Location Of “firmware.bin” File from the Marlin Compilation for SKR V1.3 Board	613
APPENDIX F	616
Links to Reference Material	616
Marlin Firmware Documentation.....	616
Information on Stepper Motor Drivers and Micro-stepping	616
POLOLU A4988 and BIQU A4988.....	617
DRV8825.....	617
BIQU LV8729, FYSETC LV8729, LERDGE LV8729, and MKS LV8729	618
BIQU LV8729.....	618
FYSETC LV8729.....	619
LERDGE LV8729.....	619
MKS LV8729.....	620
FYSETC S6128 V1.1	620
FYSETC ST820.....	621

Table of Contents

BIQU ST820	621
POLOLU ST820 (STSPIN820)	622
POLOLU MP6500	622
POLOLU TB67S249FTG	623
BIQU S109 and FYSETC S109	623
BIQU S109	624
FYSETC S109	624
Marlin Firmware Documentation Specific to TMC Drivers	625
Information Common to All TMC Drivers	625
BIQU TMC2100 and MKS TMC2100	626
BIQU TMC2130	627
Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2	627
Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2 (Continued)	628
BIQU TMC2208 V3.0	628
FYSETC TMC2208 V1.2	628
Information Common to TMC2208 and BIQU TMC2225	629
BIQU TMC2225 V1.0	629
BIQU TMC2209 V1.2	630
BIQU TMC5160 V1.2	630
BIQU TMC5161 V1.0	631
SKR V1.3 Board	631
Facebook Groups	632
Miscellaneous Information	633
APPENDIX G	634
BIGTREETECH Reference Material	634
Original PIN Diagram	634
Original Wiring Diagram 1	635

Table of Contents

Original Wiring Diagram 2 for STEP/DIR Mode	636
Original Wiring Diagram 3 for UART Mode	637
Original Wiring Diagram 4 for SPI Mode	638
Original <u>PIN 1</u> Diagram	639
Additional Original Reference Material for <u>PIN 1</u> Diagram	640
Original Schematic Diagram	641
APPENDIX H.....	642
Filament Runout Sensor Wired to Limit Switch {X-, Y-, Z-, X+, Y+, or Z+}	642
Marlin 2.0.x Setup for Filament Runout Sensor Connected to X+ Endstop Connector.....	643
Marlin 2.0.x Setup for Filament Runout Sensor	644
APPENDIX I	649
Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable	649
Wiring Diagram for Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable	652
Marlin 2.0.x Setup for Communicating with Raspberry Pi via Serial Connection on SKR V1.3 Board	655
APPENDIX J	657
Connecting SKR V1.3 with BLTouch.....	657
Wiring Diagram for Connecting SKR V1.3 with BLTouch.....	660
Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board	661
Understanding Marlin Firmware's NOZZLE_TO_PROBE_OFFSET Setting	665
APPENDIX K.....	680
Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor.....	680
Instructions on How to Perform the SKR V1.3 Hardware Hack to Obtain an ADC Input on a Thermistor Port.....	685
Instructions for Suppling Power to the PT100 Amplifier Board	688
Method #1 for Powering the PT100 Amplifier Board (Digital PWR).....	689
Method #2 for Powering the PT100 Amplifier Board (Analog PWR).....	690
Technique #1 & Method #1 (Digital PWR) Wiring Diagram for Connecting Up You PT100 to the TFT Header	691
Technique #2 & Method #1 (Digital PWR) Wiring Diagram for Connecting Up Your PT100 to the TH0 Connector	692

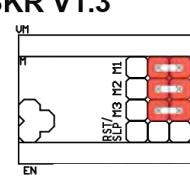
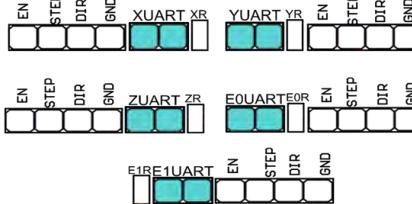
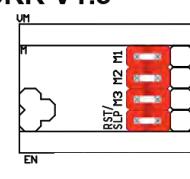
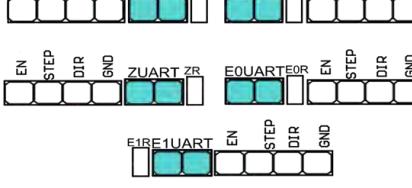
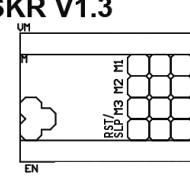
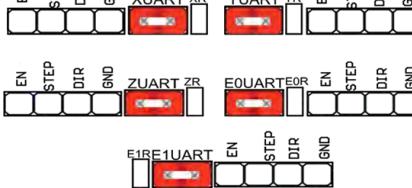
Table of Contents

Technique #2 & Method #2 (Analog PWR) Wiring Diagram for Connecting Up Your PT100 to the TH0 Connector.....	693
Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board	695
Marlin 2.0.x Firmware Setup via Technique #1	695
Marlin 2.0.x Firmware Setup via Technique #2	700
APPENDIX L.....	705
 Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board.....	705
Instructions on How to Perform the SKR V1.3 Hardware Hack to Obtain an ADC Input on a Thermistor Port.....	710
 Instructions for Suppling Power to the AD8495 Amplifier Board.....	713
Method #1 for Powering the AD8495 Amplifier Board (Digital PWR)	714
Method #2 for Powering AD8495 Amplifier Board (Analog PWR)	715
 Technique #1 & Method #1 (Digital PWR) Wiring Diagram for Connecting Up Your K-Type Thermocouple to the TFT.....	716
 Technique #2 & Method #1 (Digital PWR) Wiring Diagram for Connecting Up Your K-Type Thermocouple to the TH0 Connector.....	717
 Technique #2 & Method #2 (Analog PWR) Wiring Diagram for Connecting Up Your K-Type Thermocouple to the TH0 Connector	718
 Marline 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board.....	719
Marlin 2.0.x Firmware Setup via Technique #1	720
Marlin 2.0.x Firmware Setup via Technique #2	726

Preface

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

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

<p>STAND-ALONE NORMAL MODE</p> 	 <p>M1, M2 and M3 rows are used to set the desired micro-steps for your 3D Printer's stepper motors which are controlled by the installed stepper motor driver boards or drivers. You set M1, M2 and M3 rows by setting JUMPERS on the MCU board. See the driver's Chip Charts (Tables) located in this manual for the different M1, M2, and M3 settings to obtain the desired micro-stepping for this mode. The RST/SLP row is not used for this mode on the SKR V1.3 board.</p> <p> - this means a JUMPER <u>maybe</u> present</p> <p>No JUMPERS on the UART-LINE per Axis.</p>
<p>SPI MODE</p> 	 <p>SPI mode is a software controlled mode.</p> <p> - this means a JUMPER <u>MUST</u> be present</p> <p>ONLY FOUR [4] (M1, M2, M3, and RST/SLP rows) SPI JUMPERS.</p> <p>No JUMPERS on the UART-LINE per Axis.</p>
<p>UART MODE</p> 	 <p>UART mode is a software controlled mode.</p> <p> - this means a JUMPER <u>MUST</u> be present</p> <p>No (M1, M2, M3, and RST/SLP rows) JUMPERS.</p> <p>ONE [1] JUMPER on the UART-LINE per Axis.</p> <p>Up to FOUR [4] JUMPERS are possible.</p>

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

What is a Stepper motor?

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

Properties

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

Micro stepping

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

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

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

Stall Detection and Sensor-less Homing

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

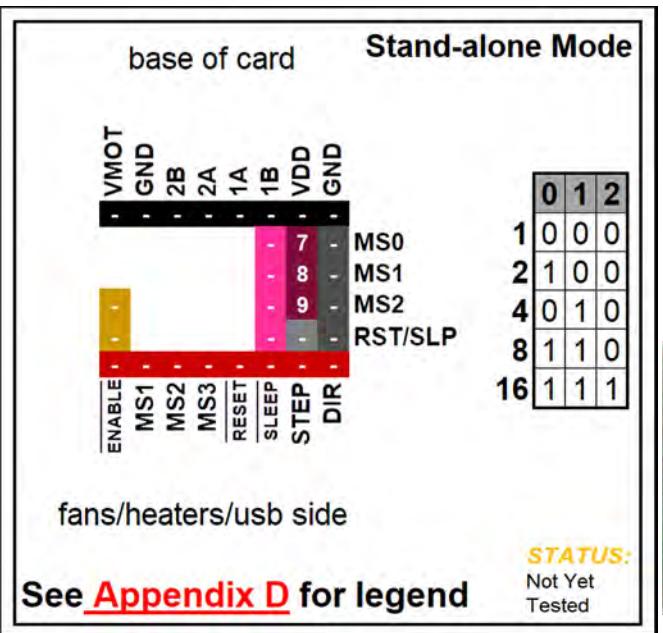
URL2: https://www.klipper3d.org/Sensorless_Homing.html#:~:text=Homing%20and%20Tuning-,Sensorless%20Homing,the%20stepper%20motor%20lose%20steps.

Overview

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

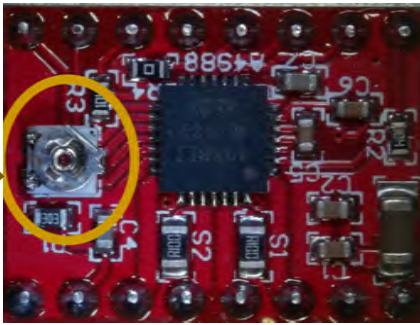
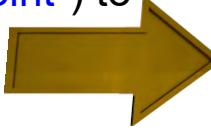
Requirements Needed to Make Sensor-less Homing Work:

- #1: You **MUST** be using a stepper driver board that has the stallGuard™ feature like the following: TMC2130 in SPI mode, TMC2209 in UART mode, TMC5160 in SPI mode and TMC5161 in SPI mode. **NOTE: The TMC2208 and TMC2225 DO NOT have stallGuard™ feature and CAN NOT be used for Senor-less Homing.**
- #2: A DIAG (DIAG/DIAG0/DIAG1) PIN **MUST** be installed on the stepper driver board or driver.
- #3: **A way to connect the DIAG PIN to the MCU Endstops or Limit switches.** Most 32-bit MCU boards do this by electrical trace in the MCU PCB, others do it with a JUMPER Block and non-32-bit MCU boards will require external DuPont Jumper cables to be used.
- #4: **Marlin Firmware** or other Firmware that can enable Sensor-less Homing on the 3D Printer. I know for Marlin >2.0.x Firmware, the sensor-less homing by default is disabled and **MUST** be enabled (un-commented) by the user.
- #5: **Physical removal of Endstops/Limit switches** **MAYBE needed** to make Sensor-less Homing work properly on the Axis. If using TMC2209 driver and you **WANT** sensor-less homing then the physical enstops **MUST** be REMOVED. If using TMC2209 driver and you **DO NOT WANT** sensor-less homing then remove the 'ST' JUMPER for the Axis from the 'ST' JUMPER block.
- #6: **If using a filament runout sensor,** remove the DIAG PIN from the driver board that is installed in the Extruder socket (E0 and/or E1 and/or E2).

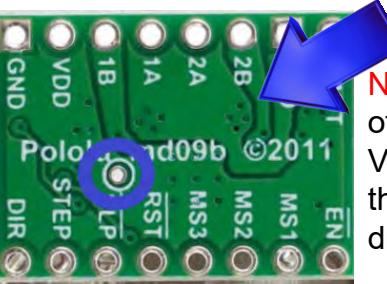


POLOLU A4988

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



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



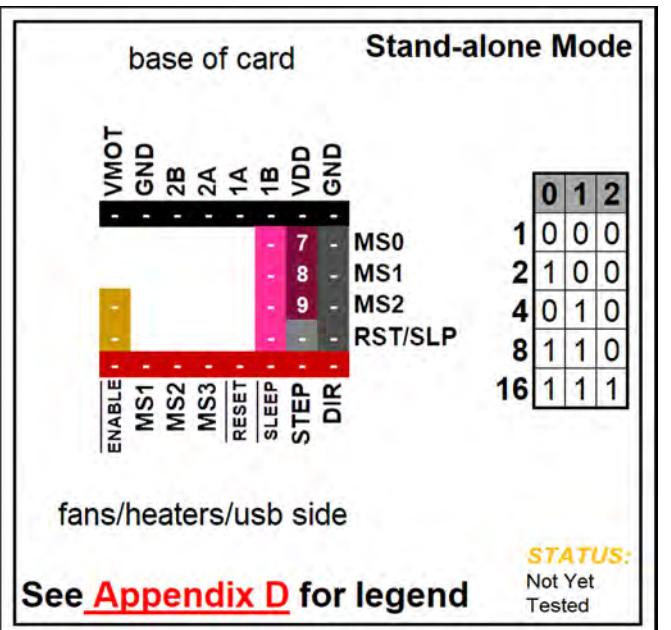
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

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

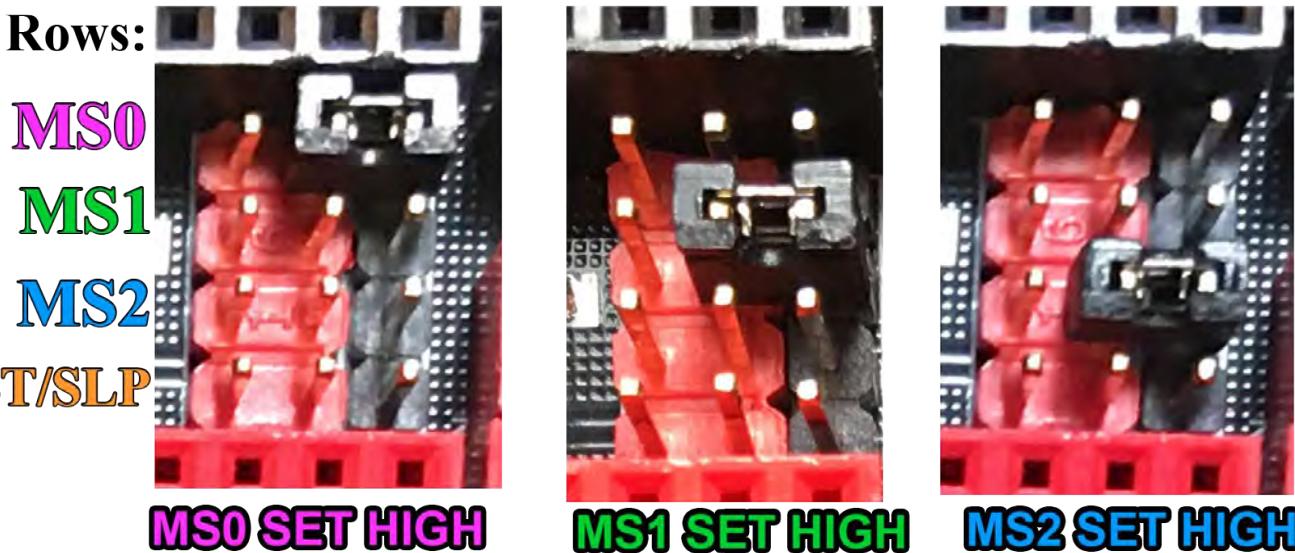
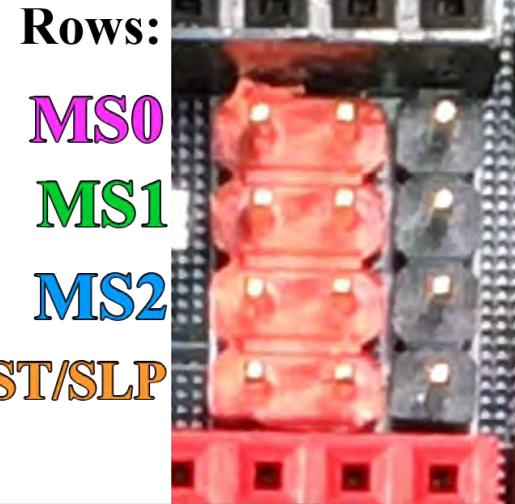
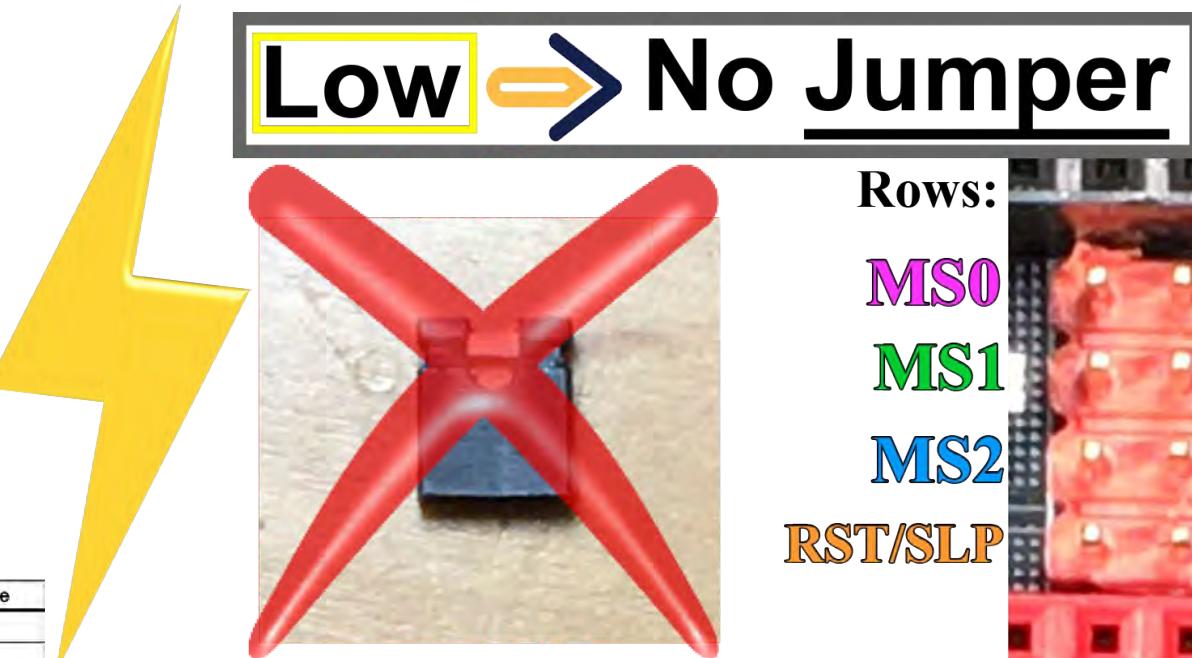
Note: See this video about current sense resistors (R_s) and their possible locations: <https://youtu.be/8wk1elugv5A>

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

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



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



Driver Chip Chart:

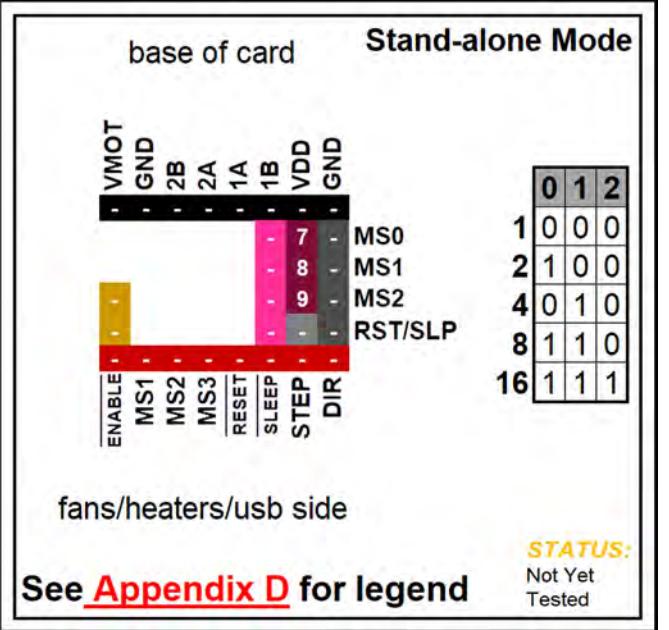
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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

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

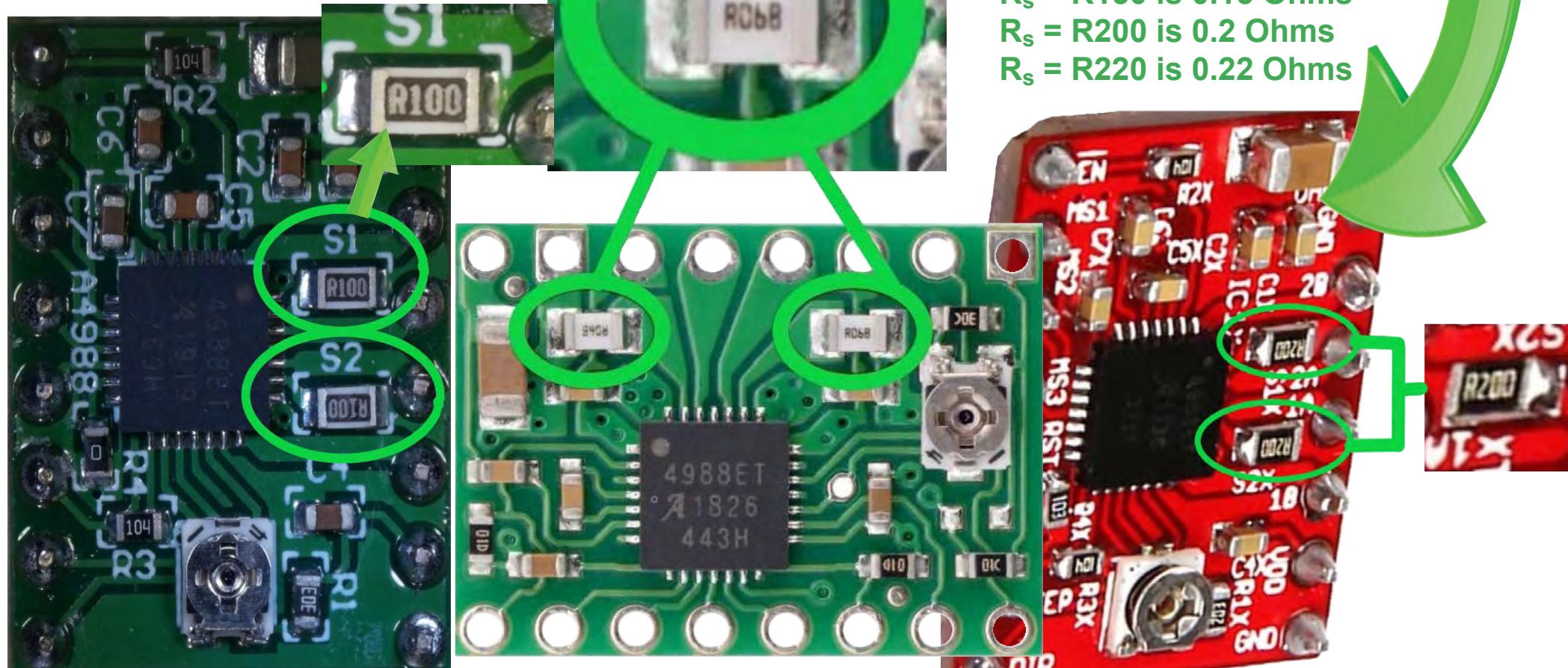
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

POLOLU A4988

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



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

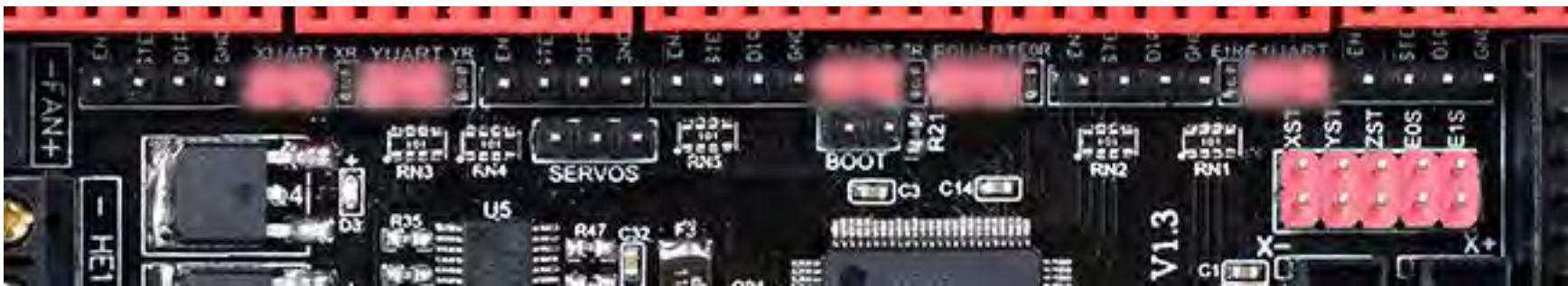


POLOLU A4988

Stand-alone Mode

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

X-	Y-
(XST) X-DIAG1	(YST) Y-DIAG1
(ZST) Z-DIAG1	(E0ST) E0-DIAG1 X+
(E1ST) E1-DIAG1 Y+	



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

XUART YUART



ZUART E0 UART



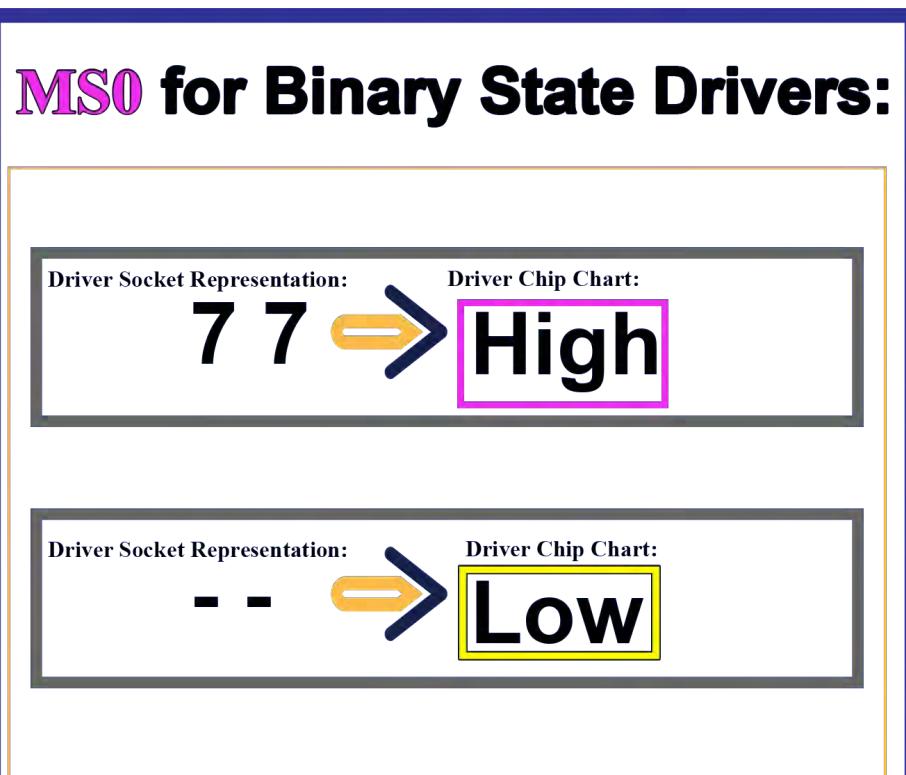
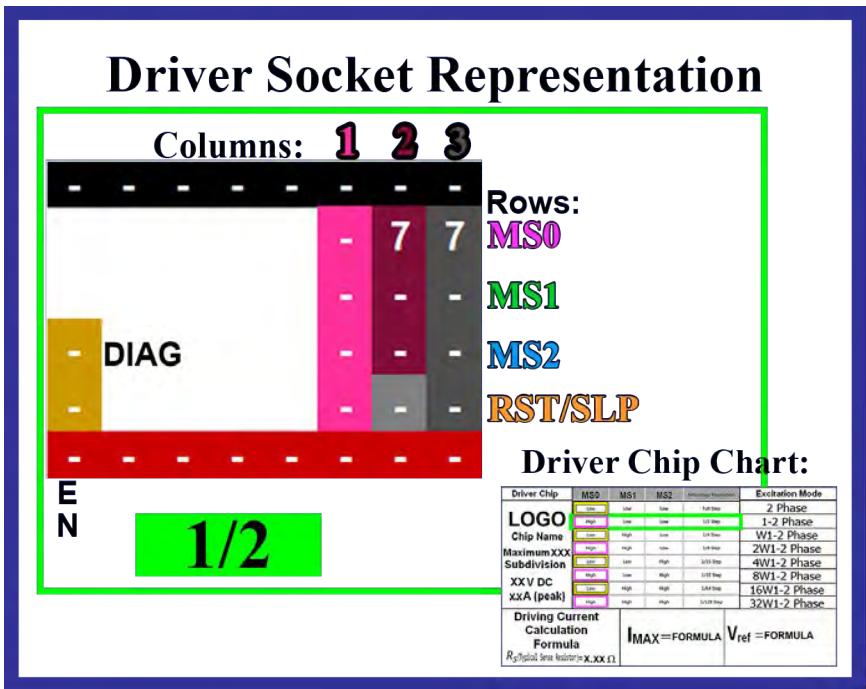
E1 UART



Stand-alone Mode

POLOLU A4988

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



Meaning:

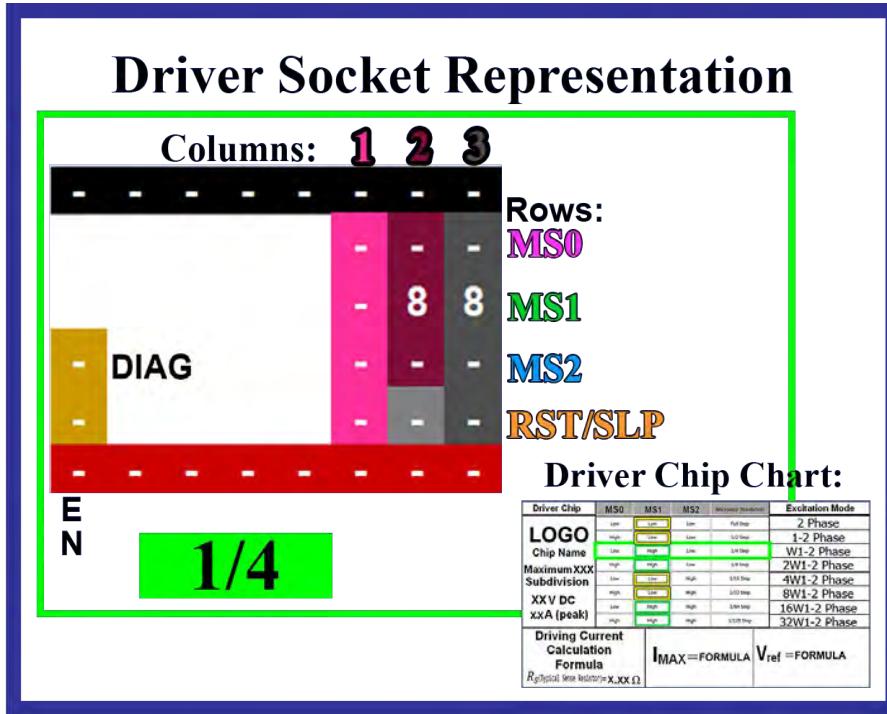
Driver Chip Chart:

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

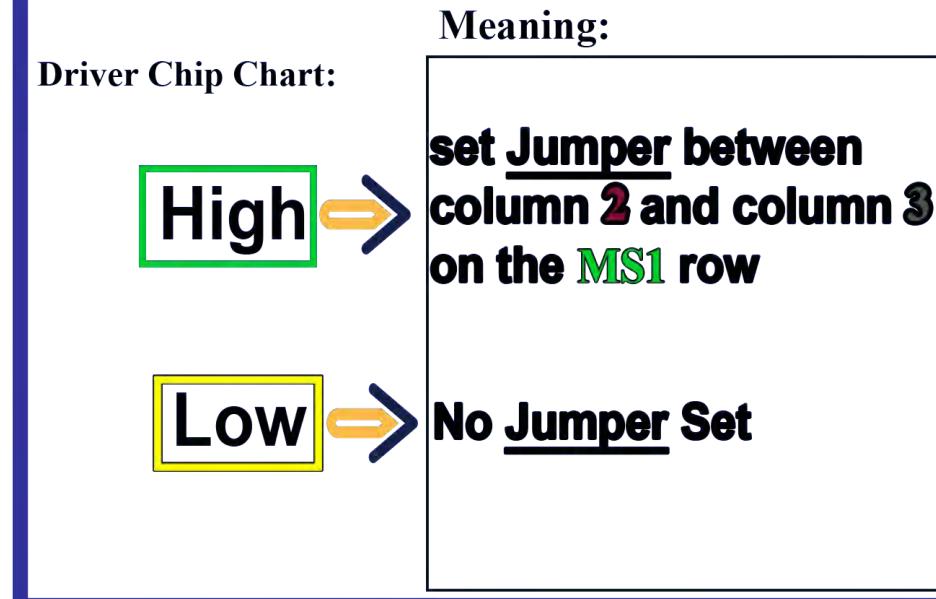
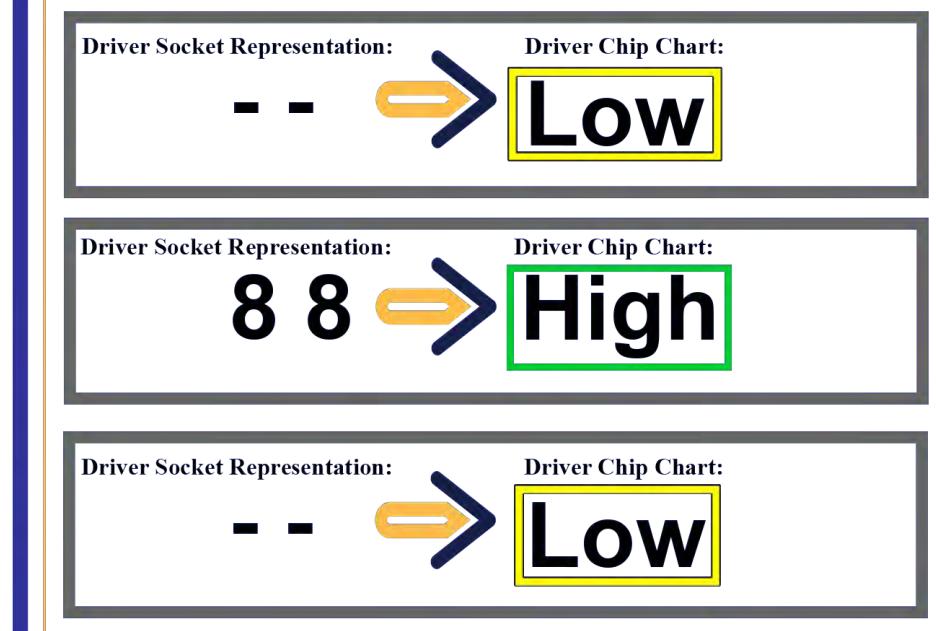
Low → **No Jumper Set**

Stand-alone Mode

POLOLU A4988



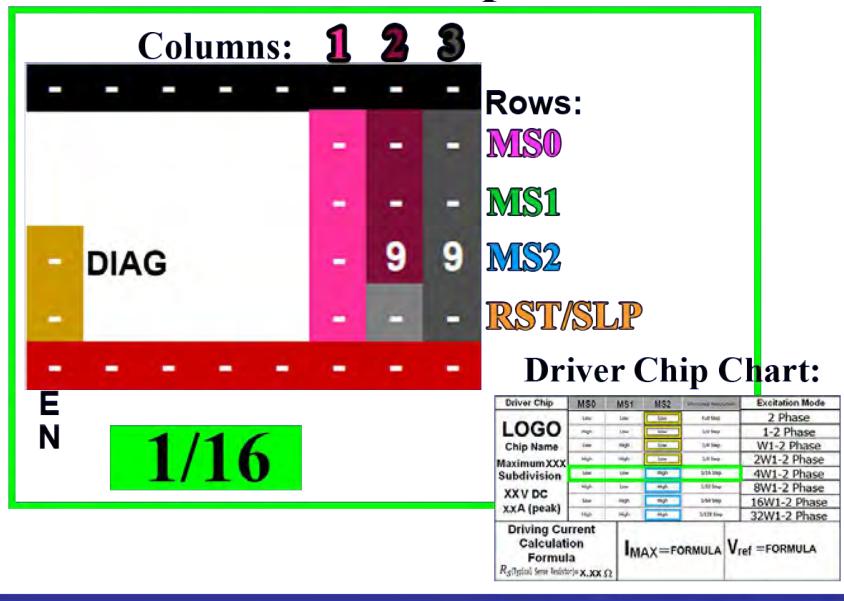
MS1 for Binary State Drivers:



Stand-alone Mode

POLOLU A4988

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

High →

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

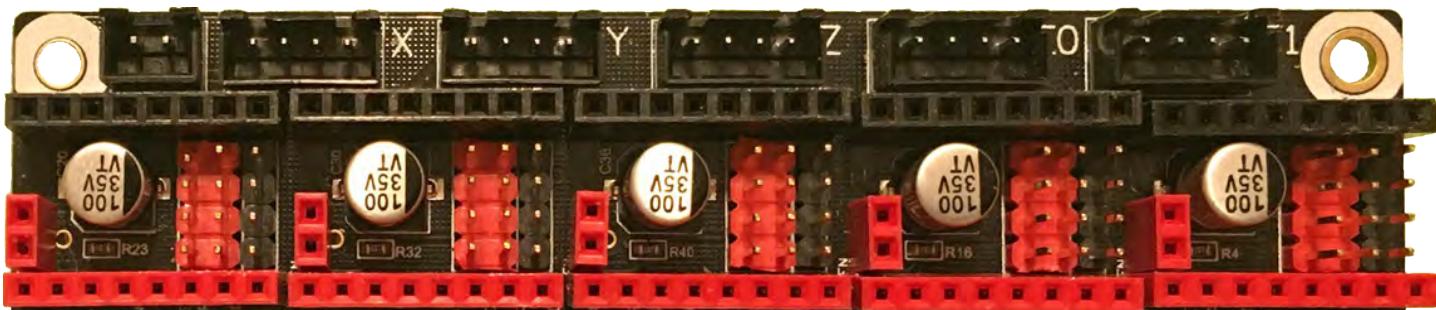
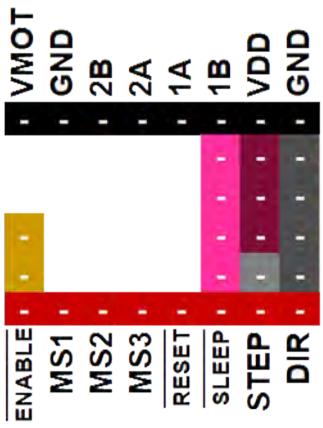
Low →

No Jumper Set

Stand-alone Mode

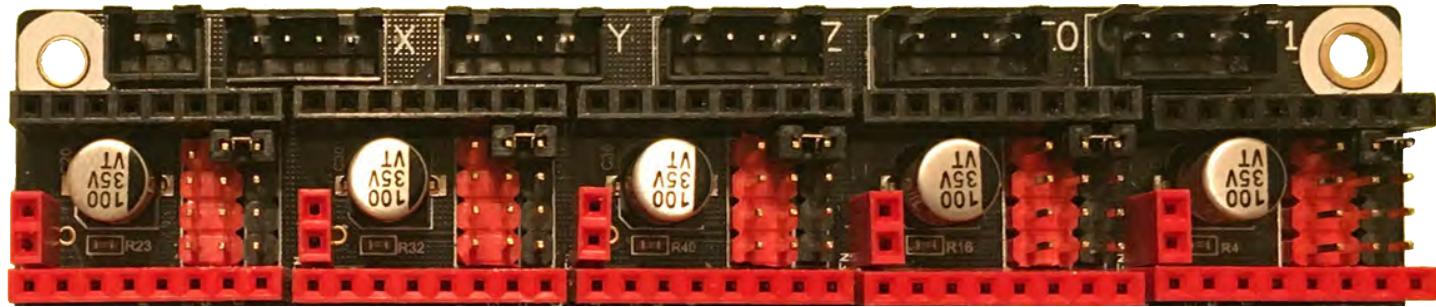
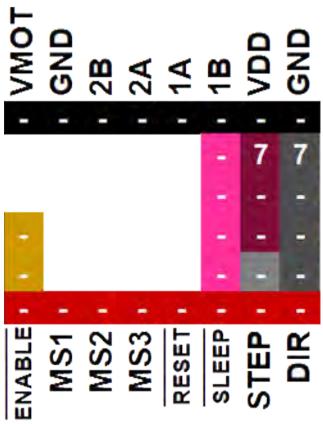
POLOLU A4988

STEP



See [Appendix D](#) for legend

1 / 2

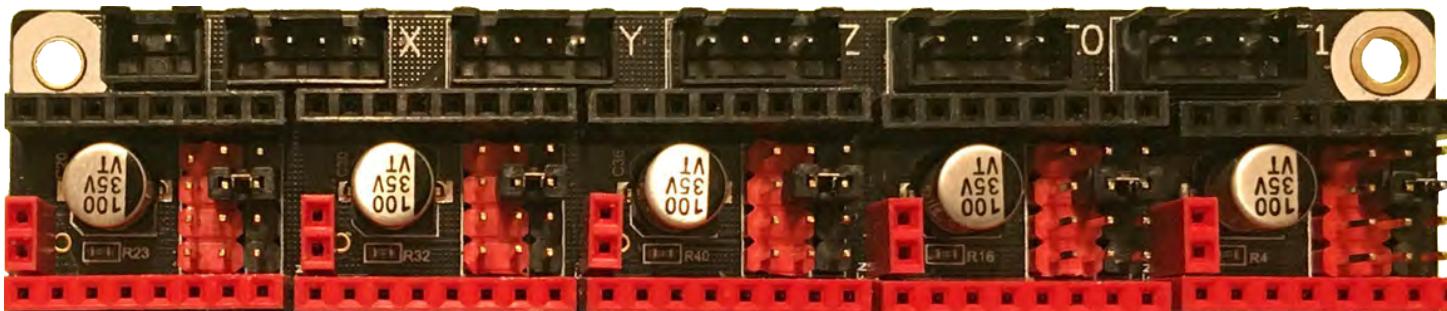
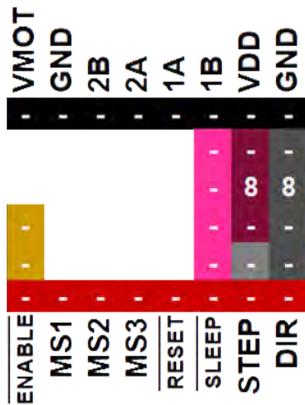


See [Appendix D](#) for legend

Stand-alone Mode

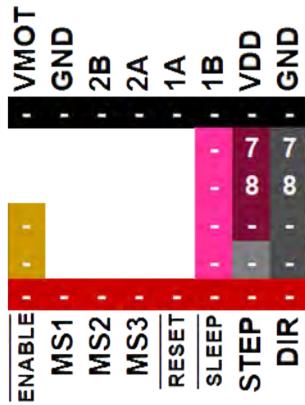
POLOLU A4988

1 / 4



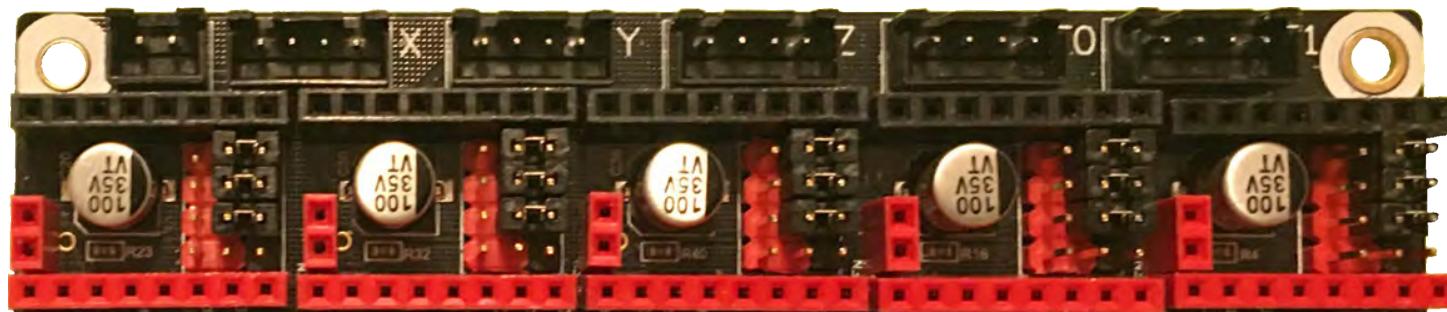
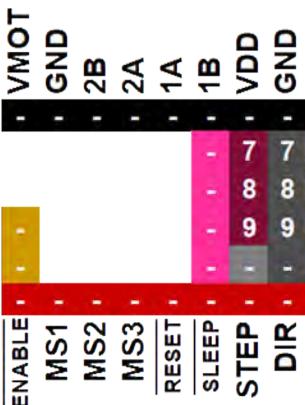
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16

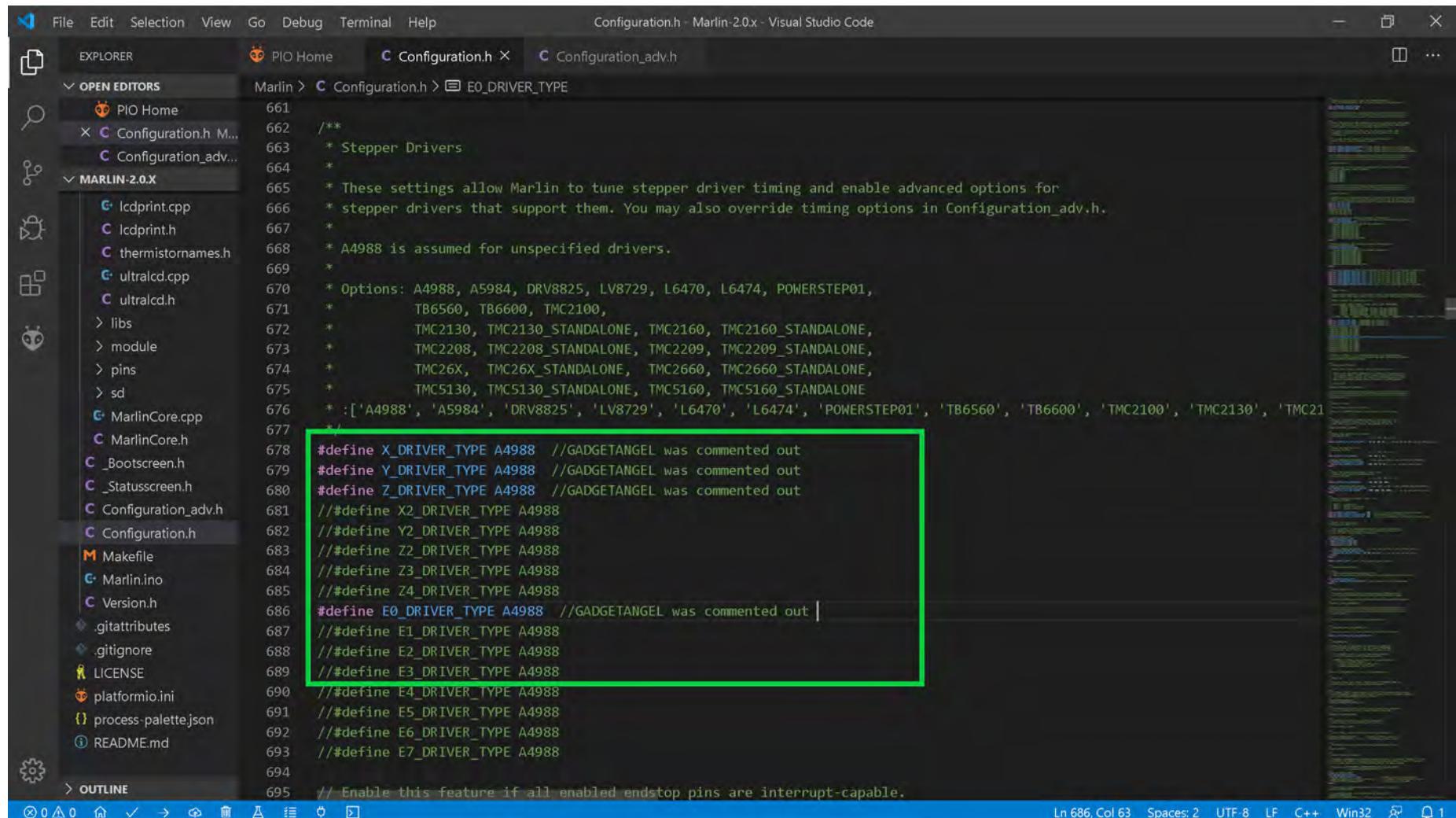


See [Appendix D](#) for legend

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

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

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



```

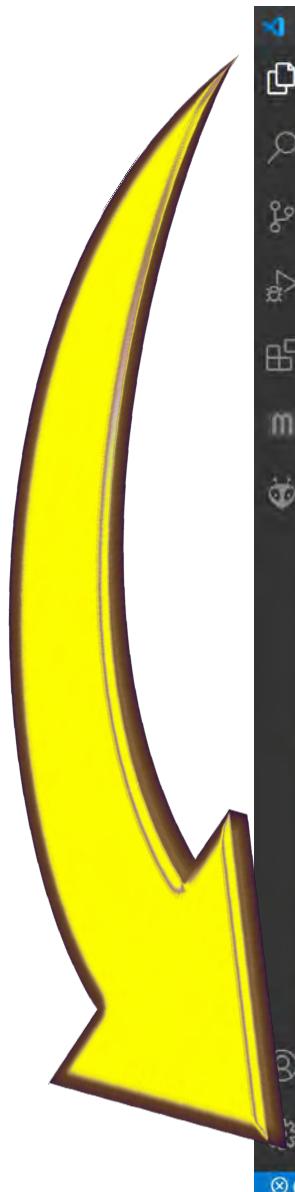
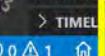
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
676 */
677 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
678 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 #define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out |
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693
694 // Enable this feature if all enabled endstop pins are interrupt-capable.

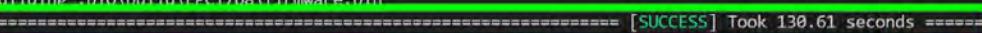
```

- Go to the next page.

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

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

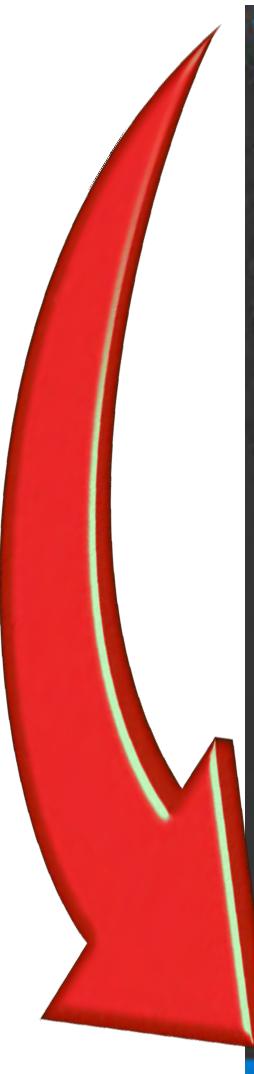


Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	DUE	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

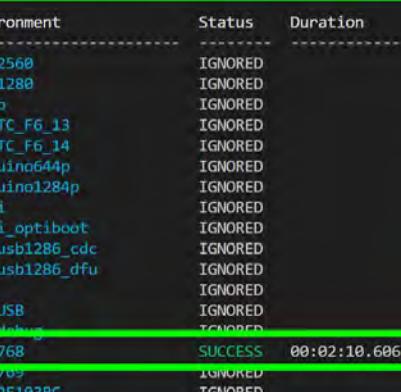
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

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







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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src\pins\pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Marlin\src\pins\pins_BTT_SKR_common.h Configuration_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

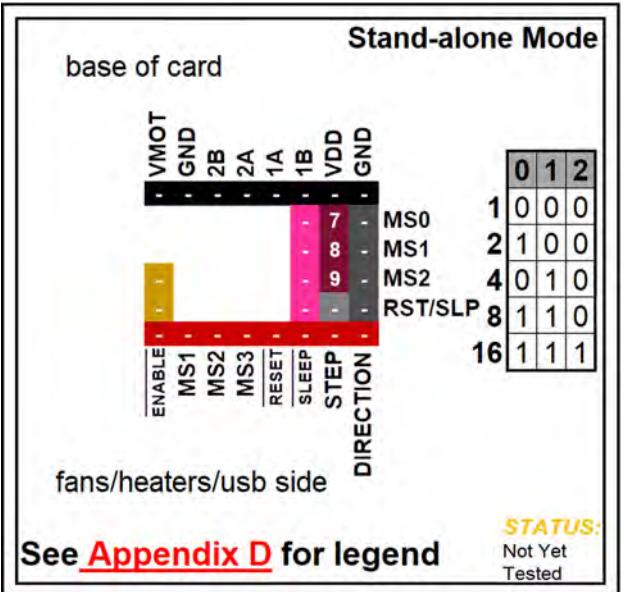
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

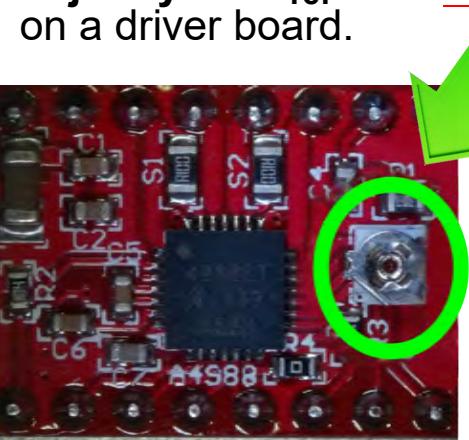
> OUTLINE > TIMELINE

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



BIQU A4988

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

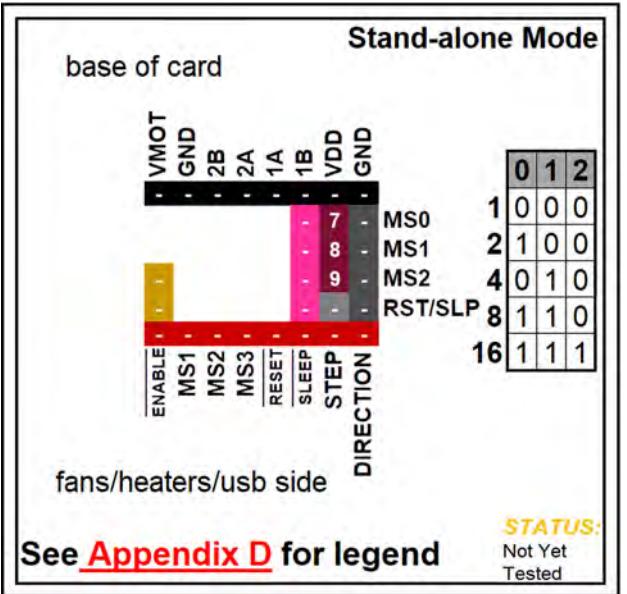


Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

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

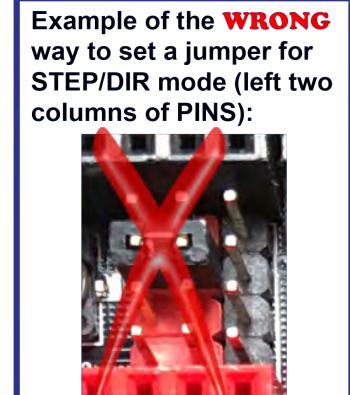
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
BIQU® A4988 Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	Low	Full step	2 Phase
	High	Low	Low	Half step	1-2 Phase
	Low	High	Low	Quarter step	W1-2 Phase
	High	High	Low	Eighth step	2W1-2 Phase
	High	High	High	Sixteenth step	4W1-2 Phase
Driving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

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



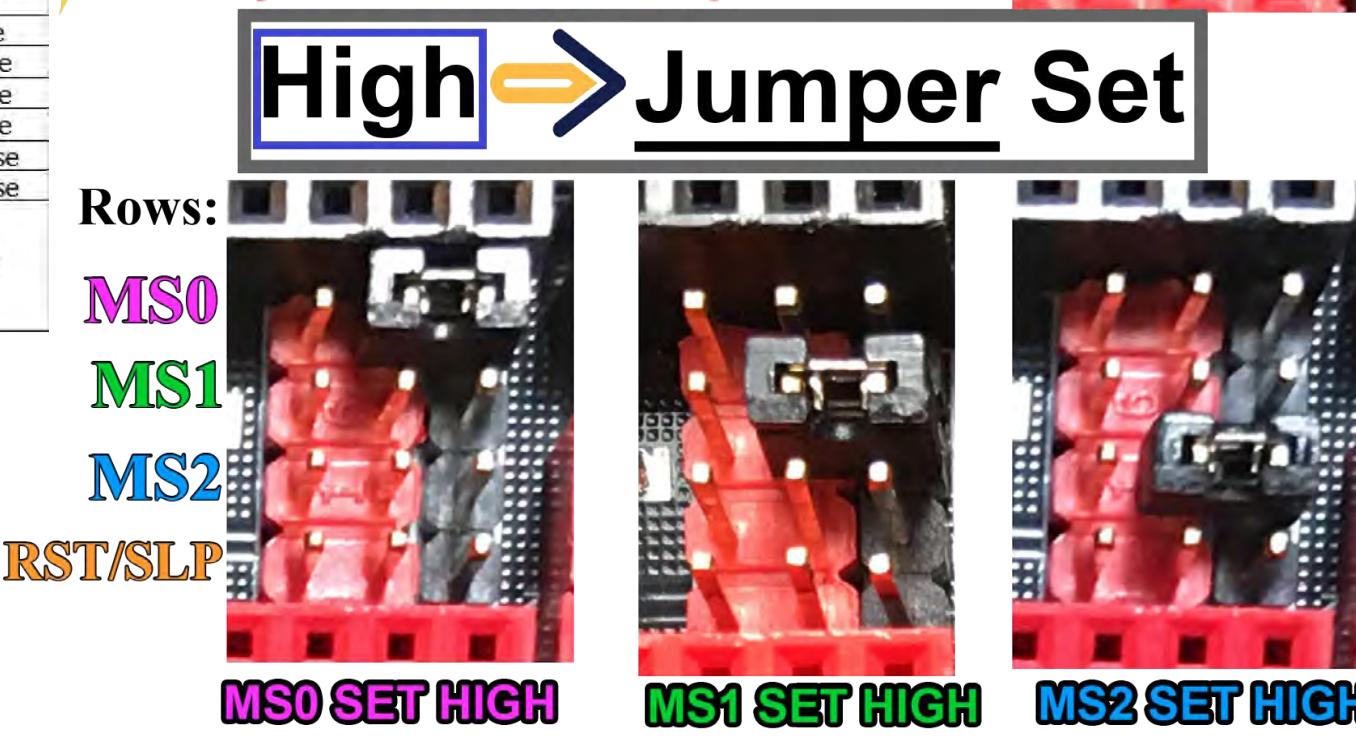
Driver Chip Chart:

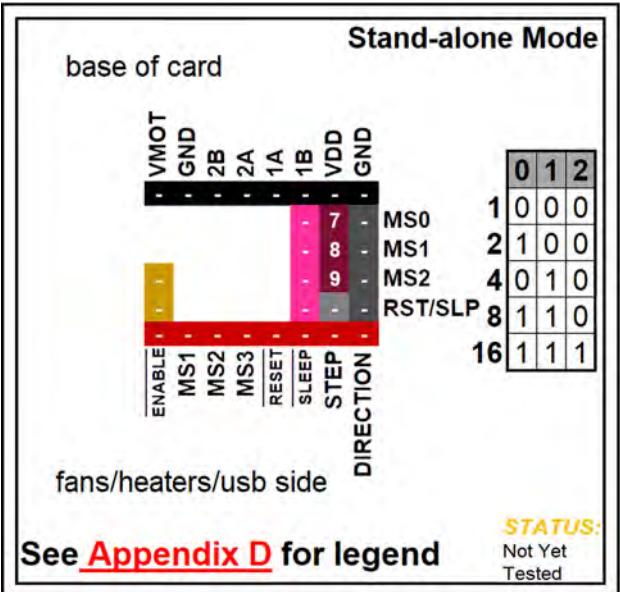
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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



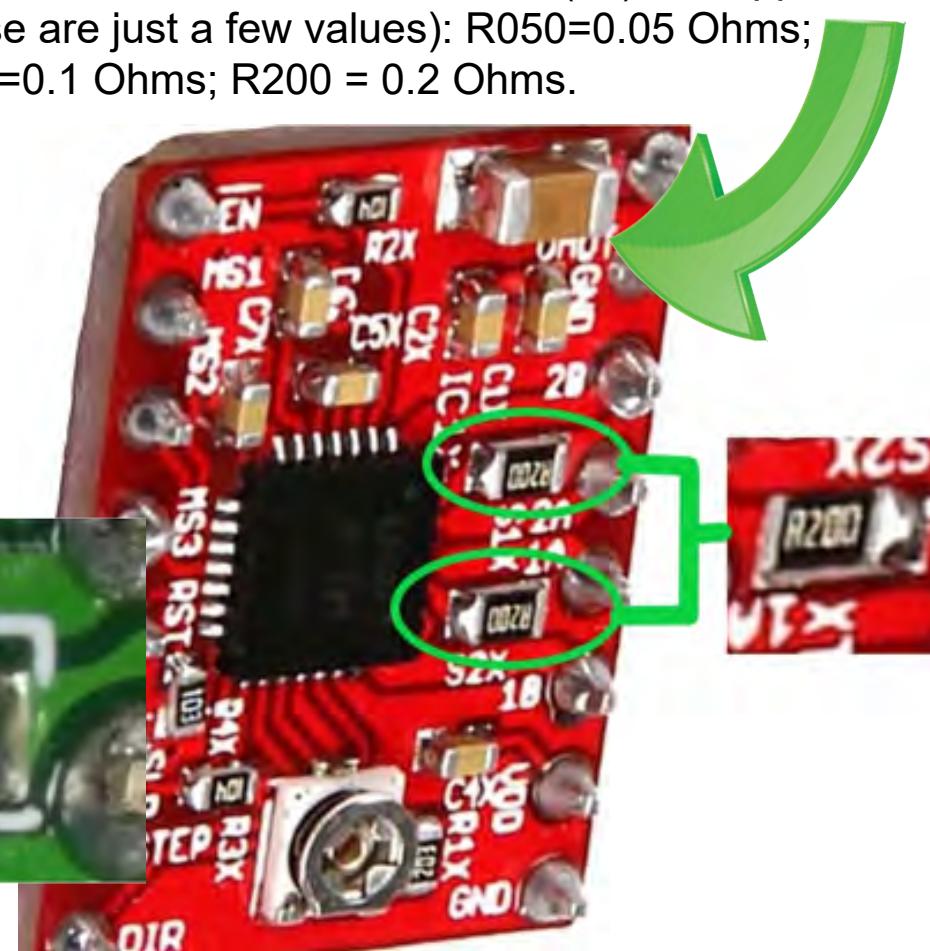
BINU A4988

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



Biqu A4988

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

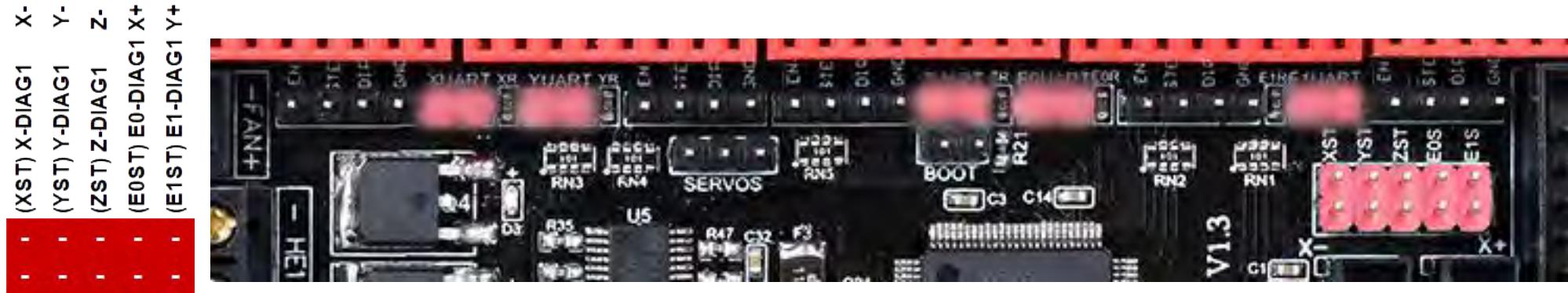


$R_s = R050$ is 0.05 Ohms; $R_s = R068$ is 0.068 Ohms
 $R_s = R100$ is 0.1 Ohms; $R_s = R150$ is 0.15 Ohms
 $R_s = R200$ is 0.2 Ohms; $R_s = R220$ is 0.22 Ohms

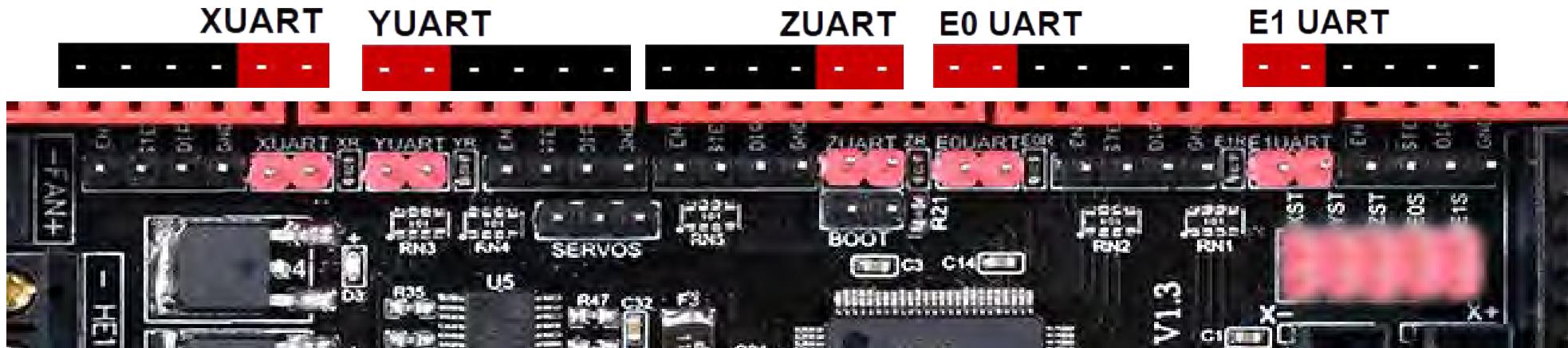
Stand-alone Mode

BIQU A4988

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

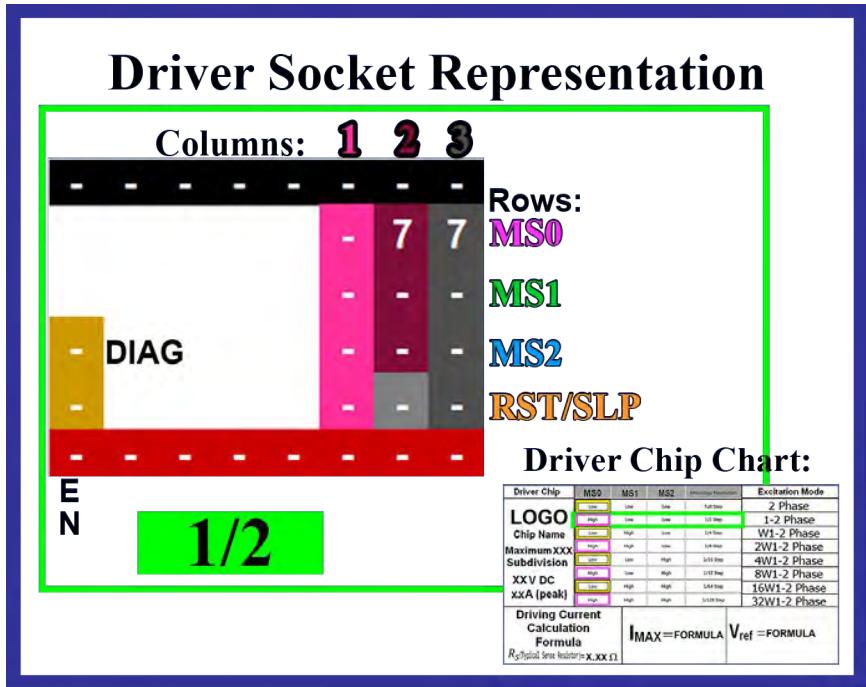


Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



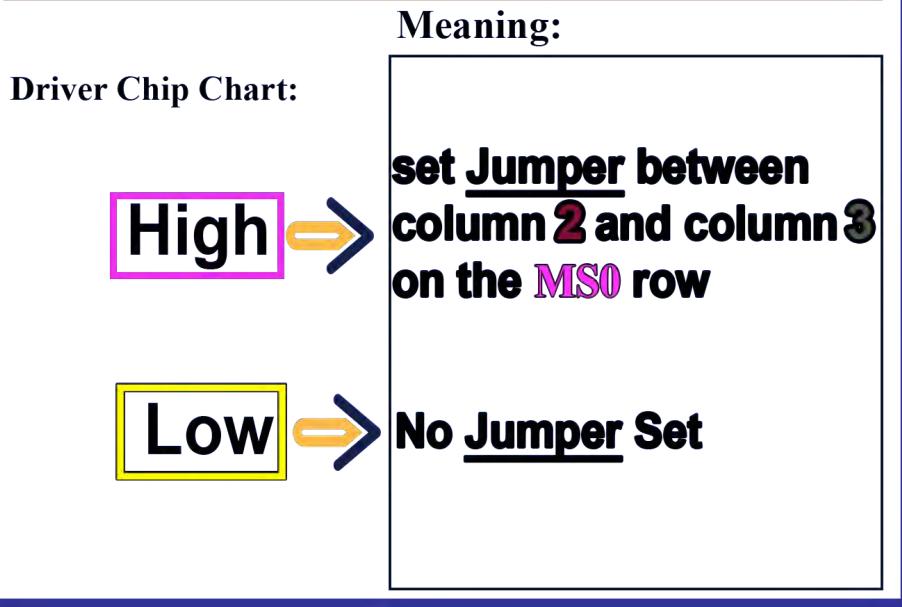
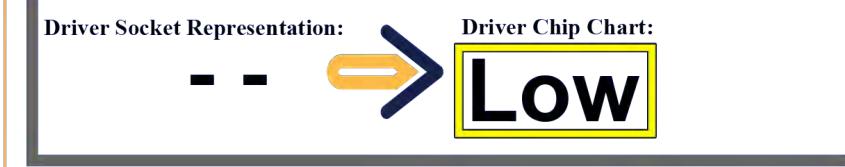
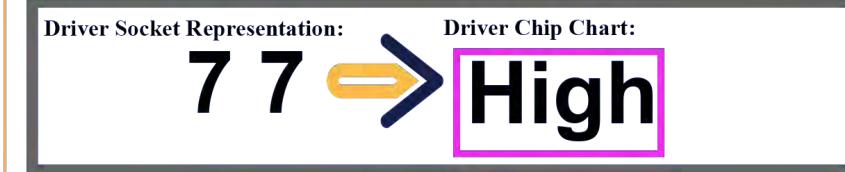
Stand-alone Mode

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



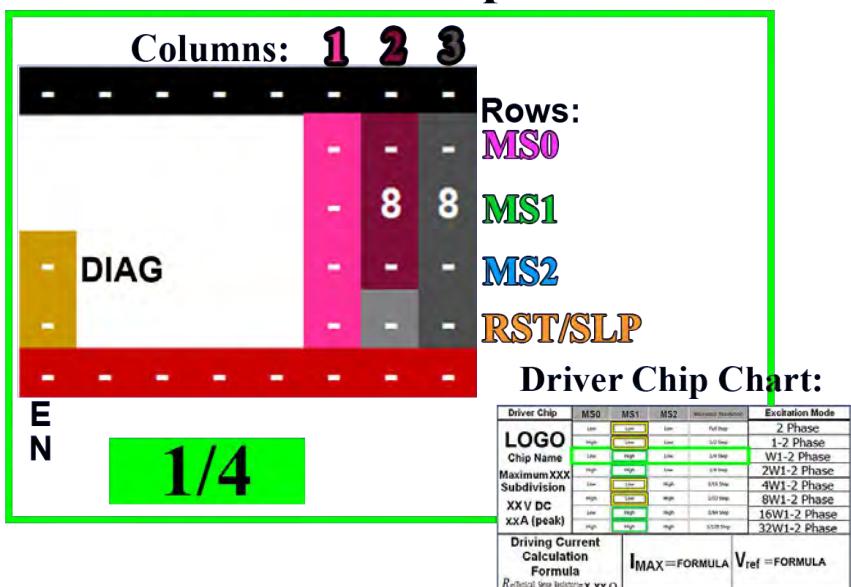
BIQU A4988

MS0 for Binary State Drivers:



Stand-alone Mode

Driver Socket Representation



High:



BINU A4988

MS1 for Binary State Drivers:

Driver Socket Representation: Driver Chip Chart:

Low

Driver Socket Representation: Driver Chip Chart:

High

Driver Socket Representation: Driver Chip Chart:

Low

Meaning:

Driver Chip Chart:

High

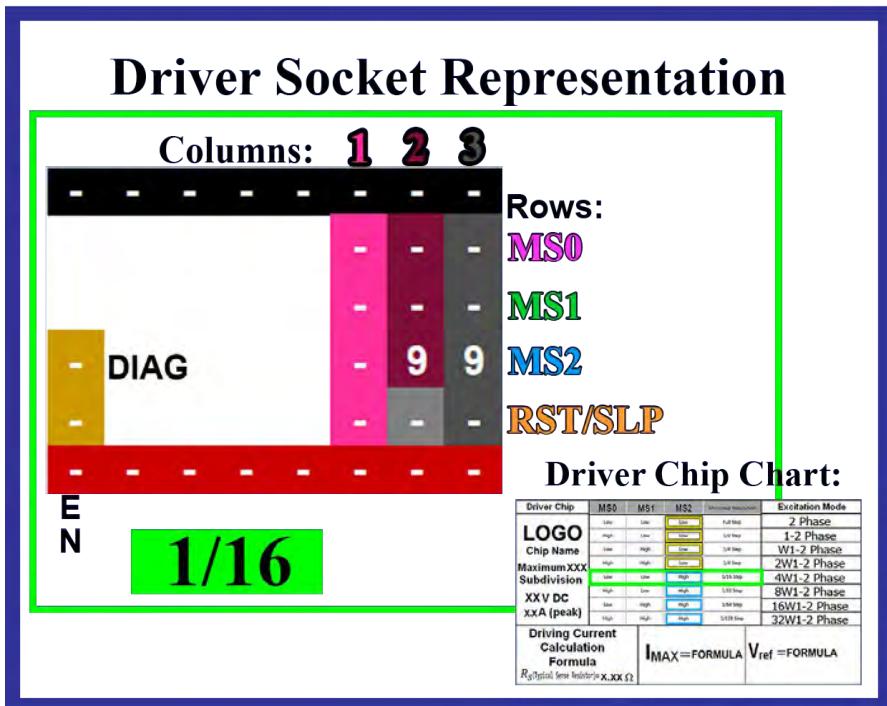
set Jumper between column 2 and column 3 on the MS1 row

Low

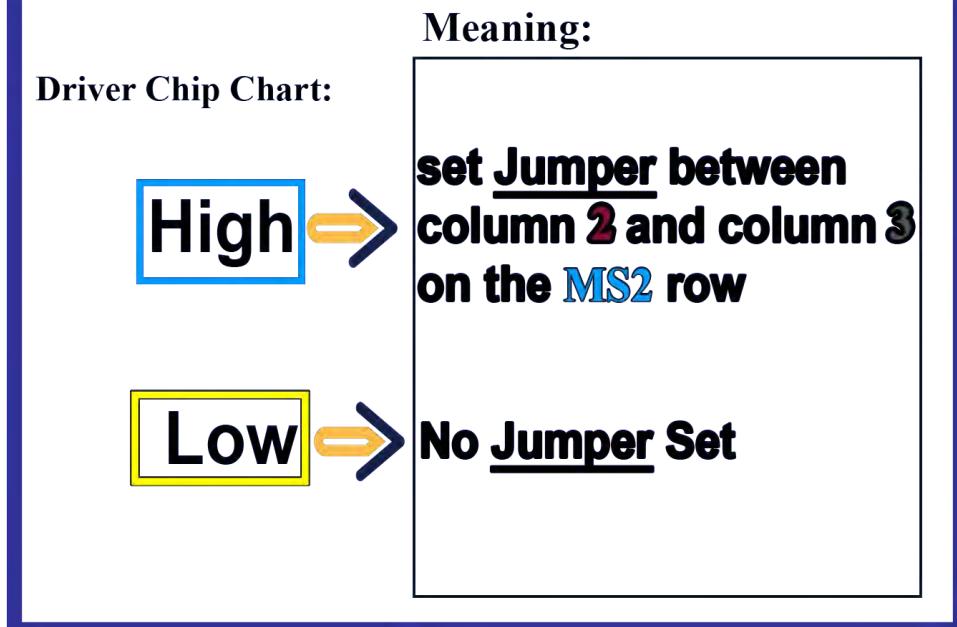
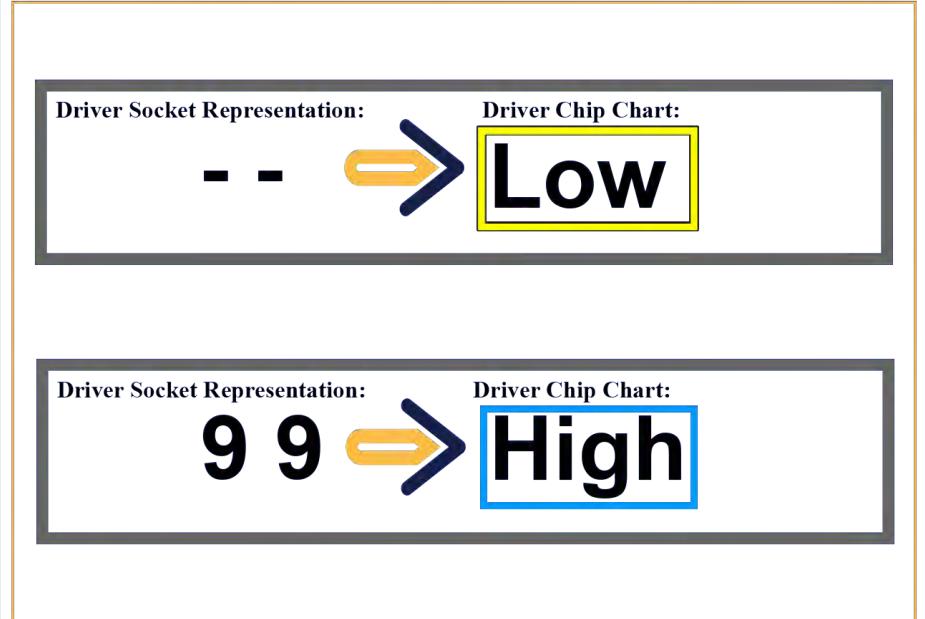
No Jumper Set

Stand-alone Mode

BINU A4988



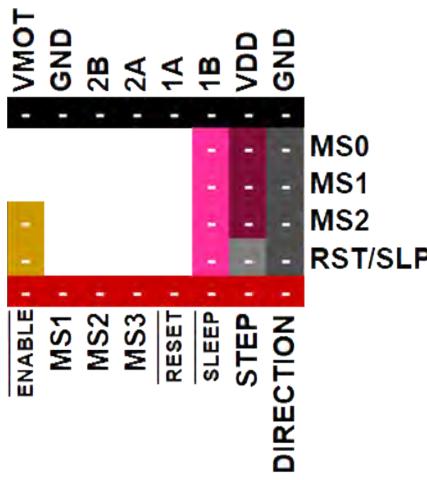
MS2 for Binary State Drivers:



Stand-alone Mode

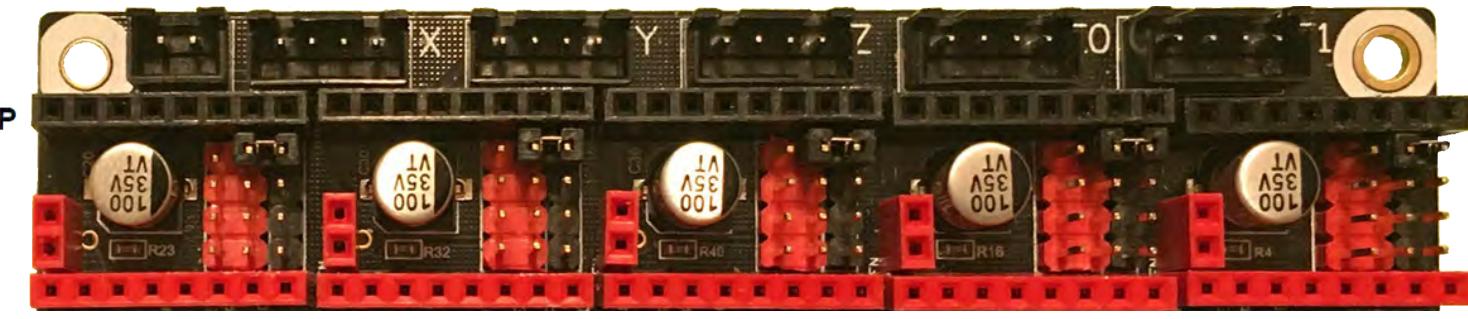
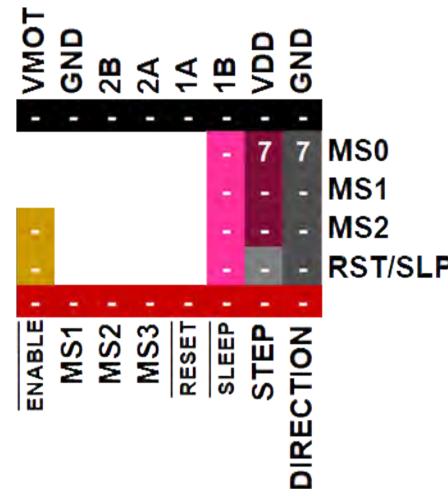
BIQU A4988

STEP



See [Appendix D](#) for legend

1 / 2



See [Appendix D](#) for legend

Stand-alone Mode

BIQU A4988

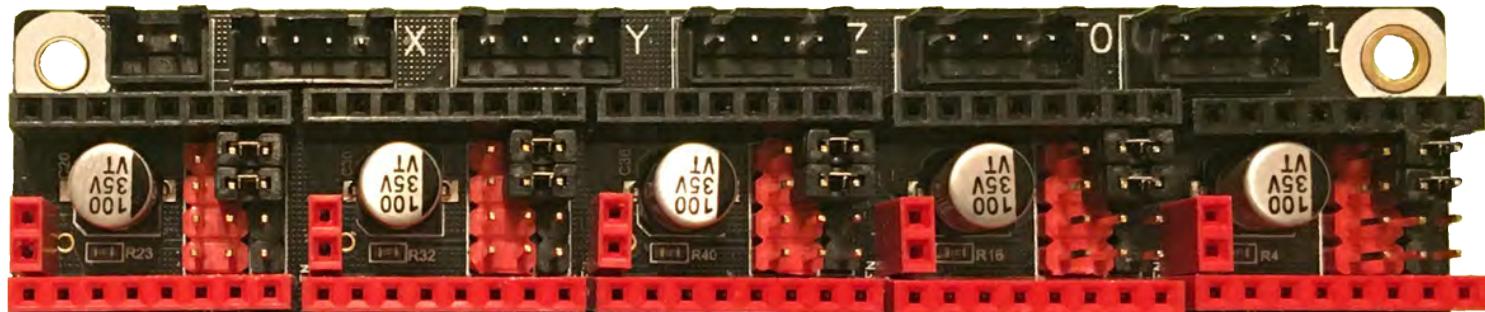
1 / 4

VMOT	GND	2B	2A	1A	1B	VDD	GND
ENABLE							
MS0							
MS1							
MS2							
RST/SLP							
DIRECTION							

See [Appendix D](#) for legend

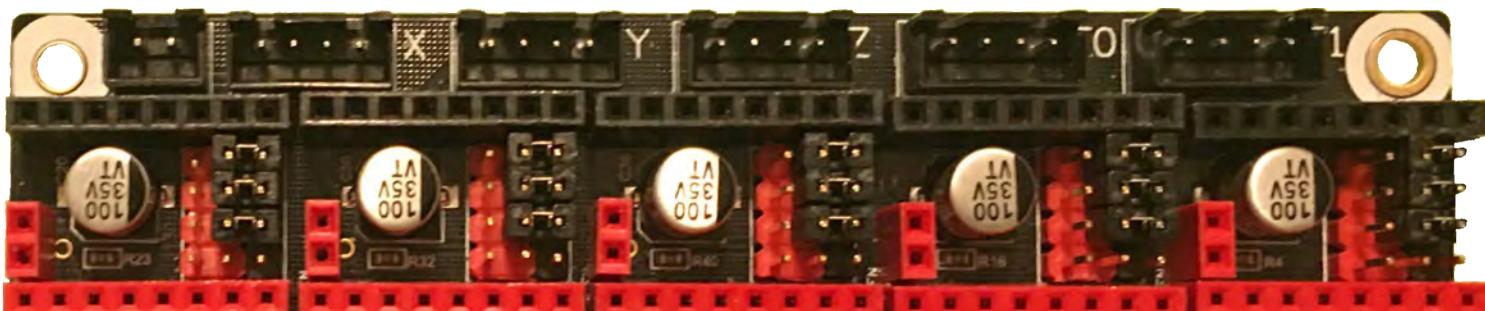
1 / 8

VMOT	GND	2B	2A	1A	1B	VDD	GND
ENABLE							
MS1							
MS2							
MS3							
RESET							
SLEEP							
STEP							
DIRECTION							

See [Appendix D](#) for legend

1 / 16

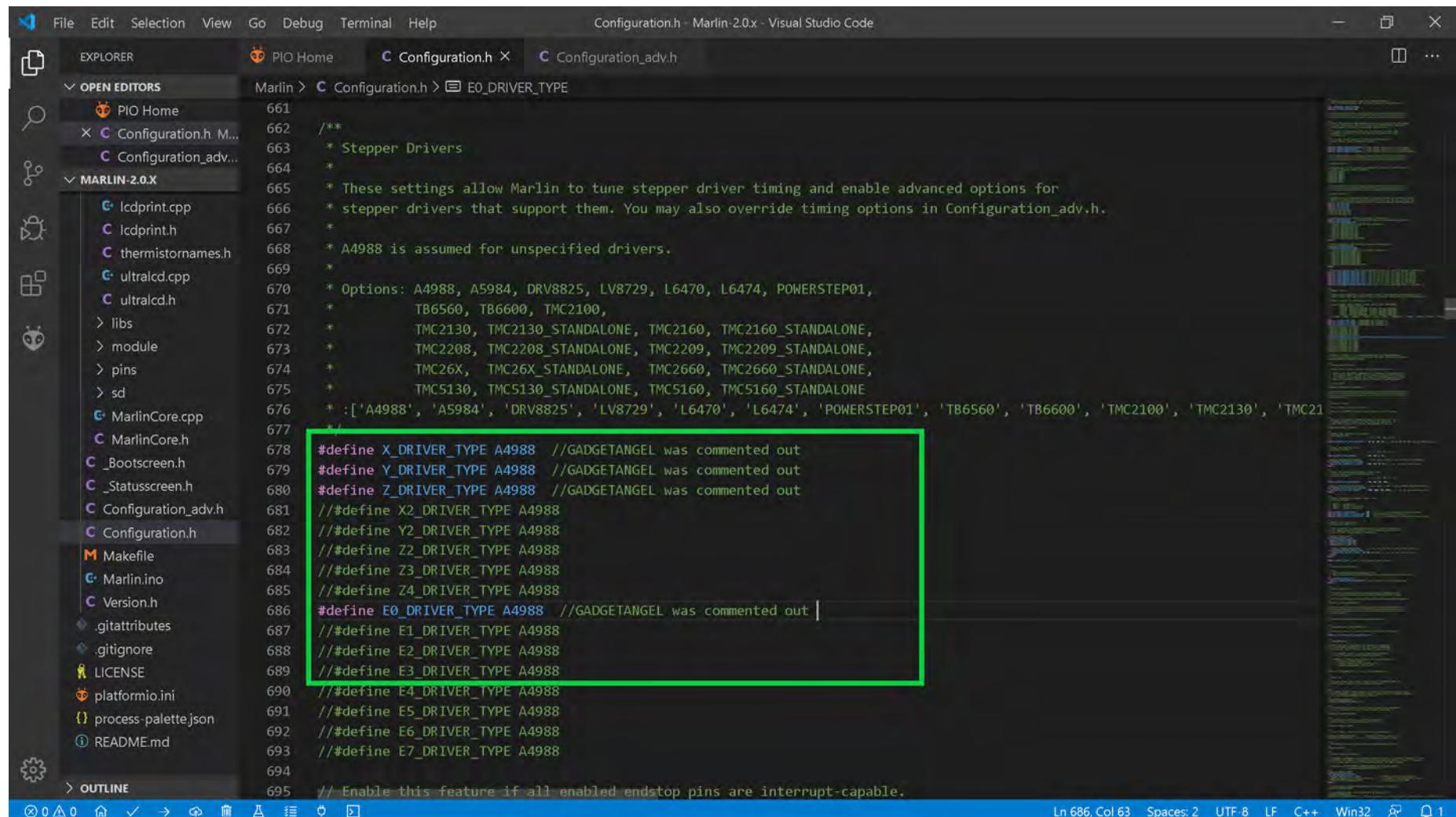
VMOT	GND	2B	2A	1A	1B	VDD	GND
ENABLE							
MS1							
MS2							
MS3							
RESET							
SLEEP							
STEP							
DIRECTION							

See [Appendix D](#) for legend

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

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

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



```

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

EXPLORER PIO Home Configuration.h Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE

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

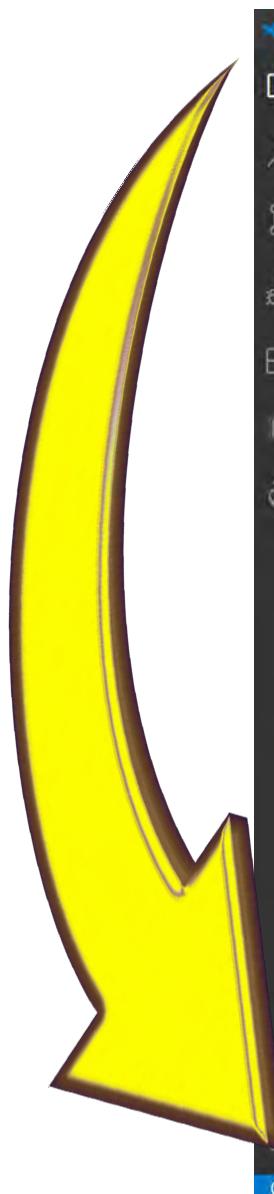
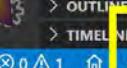
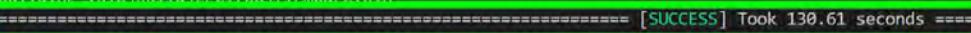
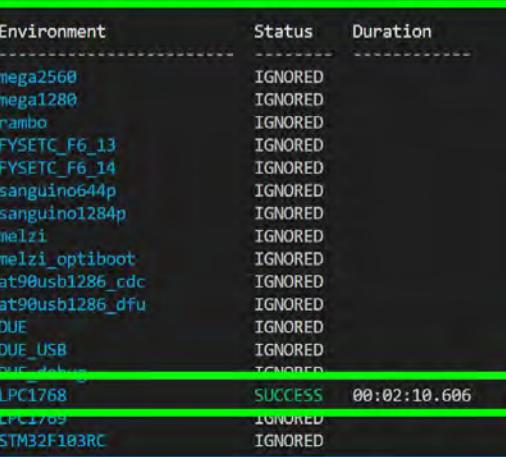
```

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

- Go to the next page.

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

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

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

EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
[SUCCESS] Took 130.61 seconds =====

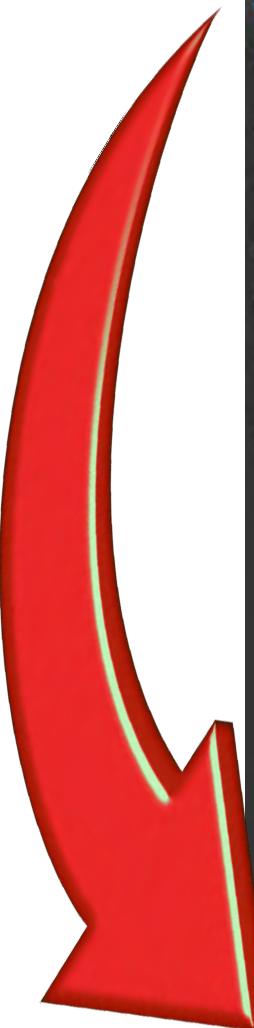
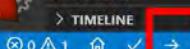
```

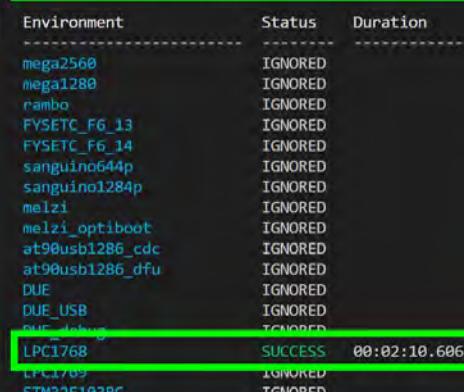
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC1768	SUCCESS	00:02:10.606
STM32F103RC	IGNORED	

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

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

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

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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

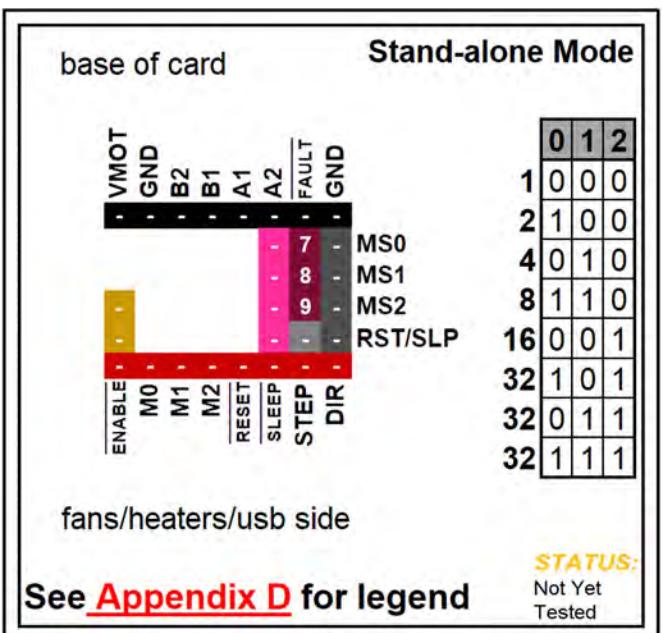
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1709	IGNORED	
STM32F103RC	IGNORED	

> OUTLINE > TIMELINE

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

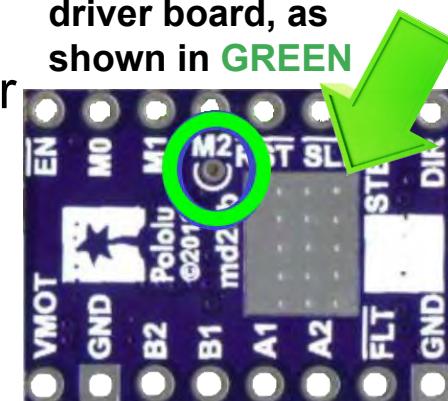


DRV8825

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

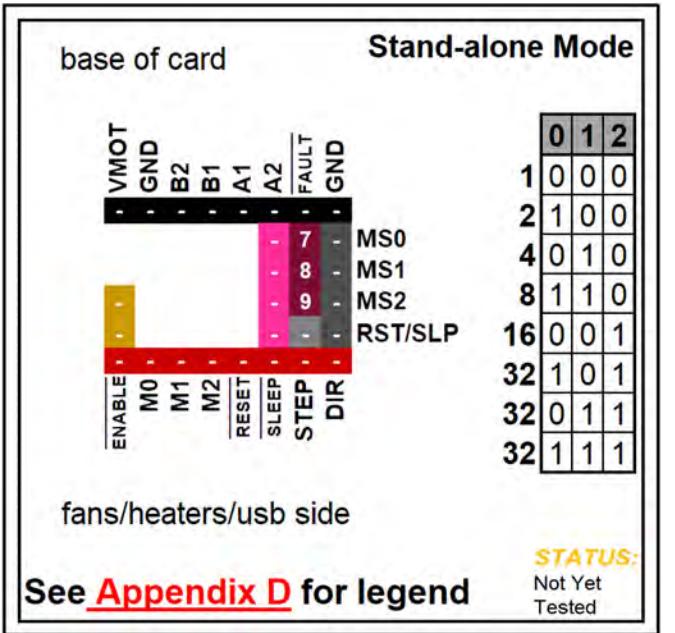
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board

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



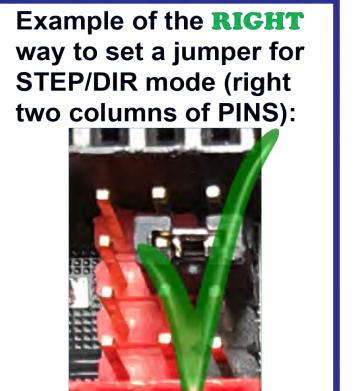
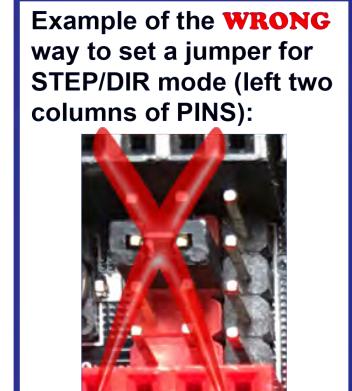
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
Pololu DRV8825 Maximum 32 Subdivision 45V DC 2.2A (peak)	Low	Low	Low	Full step	2 Phase
	High	Low	Low	Half step	1-2 Phase
	Low	High	Low	1/4 step	W1-2 Phase
	High	High	Low	1/8 step	2W1-2 Phase
	Low	Low	High	1/16 step	4W1-2 Phase
	High	Low	High	1/32 step	8W1-2 Phase
	Low	High	High	1/32 step	8W1-2 Phase
	High	High	High	1/32 step	8W1-2 Phase
Driving Current Calculation Formula		$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$	
R_S (Typical Sense Resistor) = 0.1Ω					

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



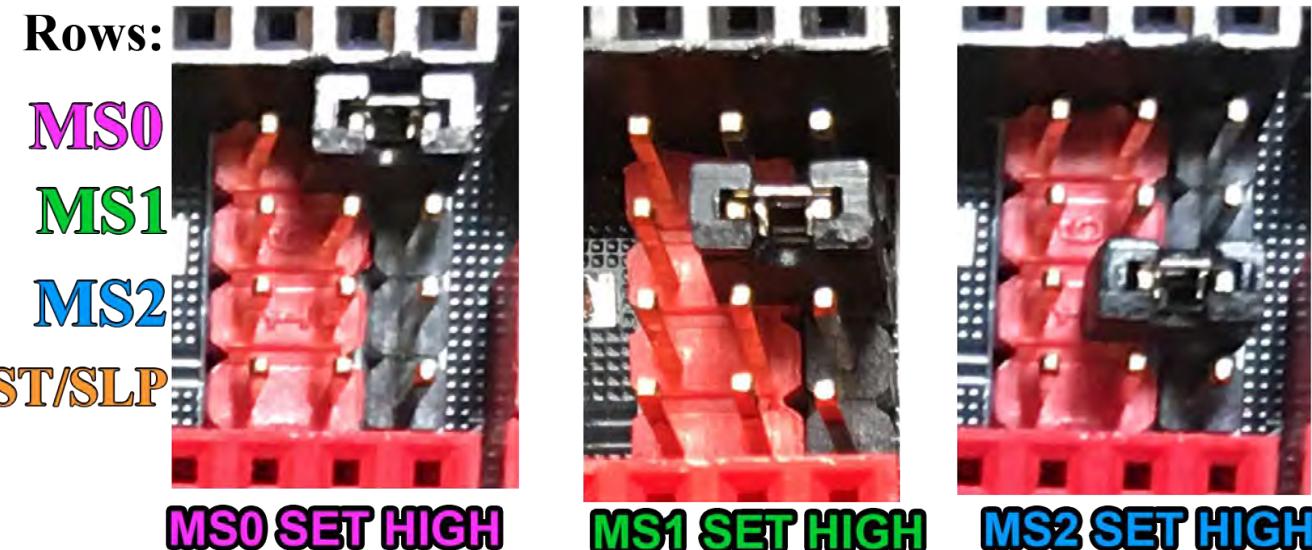
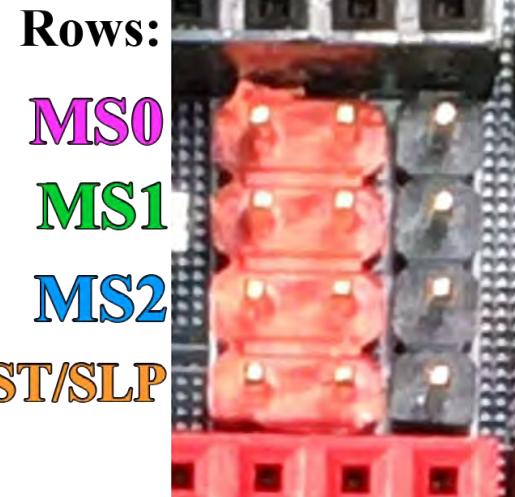
Driver Chip Chart:

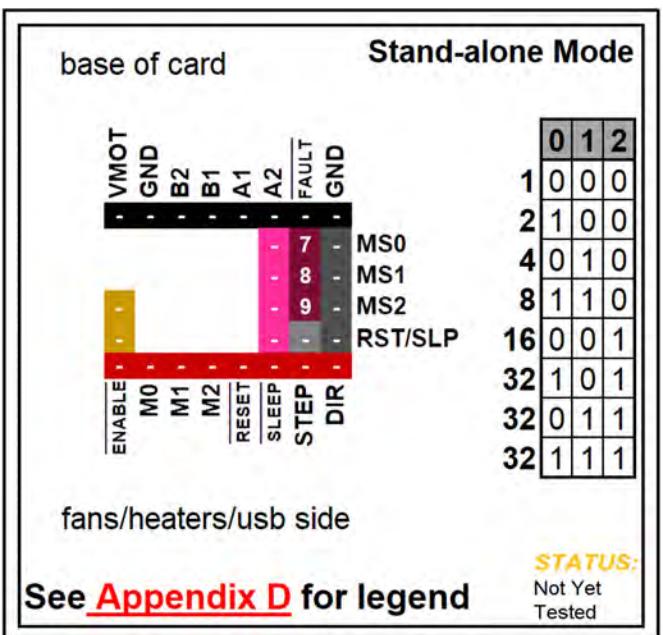
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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



DRV8825

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

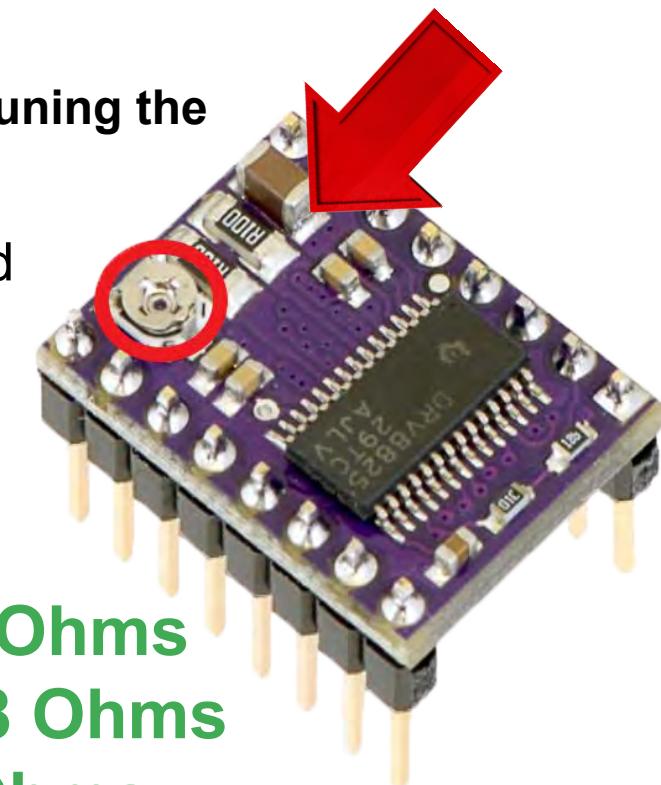




DRV8825

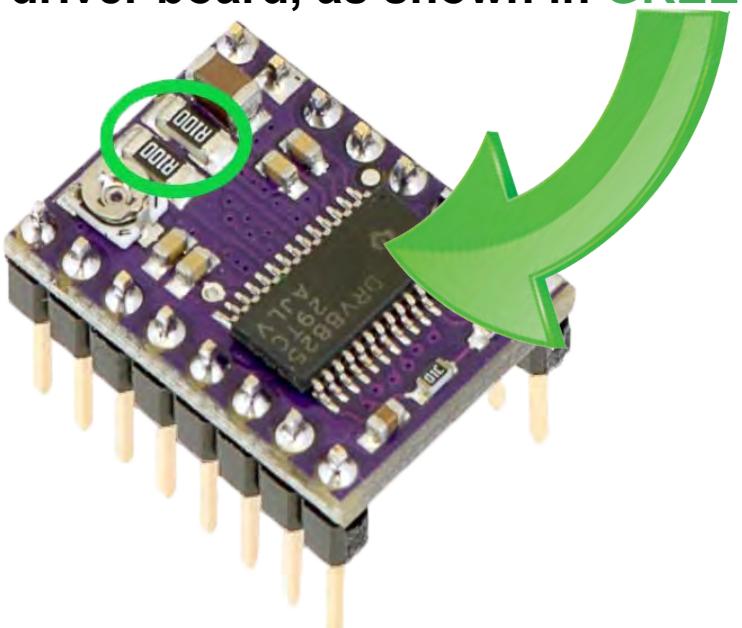
NOTE: Use the potentiometer (POT) on the top of the board (or the board's "V_{ref} Test point") to adjust your V_{ref}. See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.

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



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

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

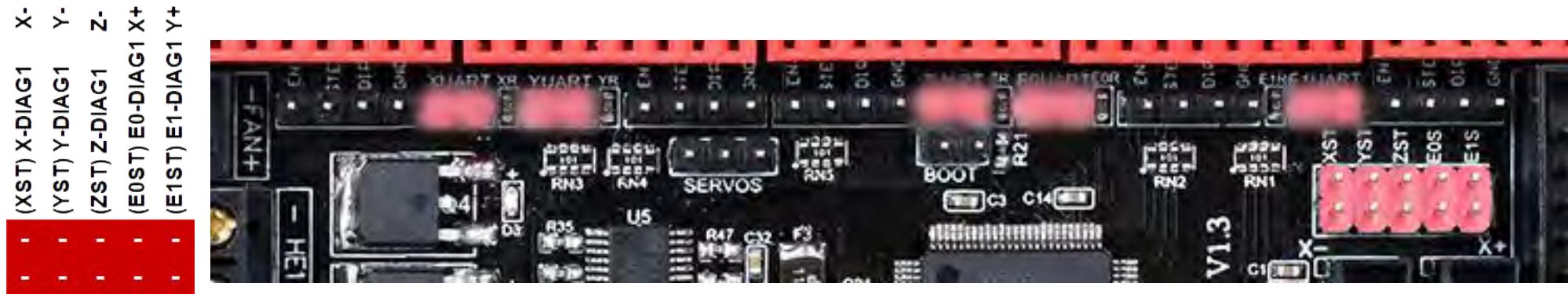


- R_s = R050 is 0.05 Ohms
- R_s = R068 is 0.068 Ohms
- R_s = R100 is 0.1 Ohms
- R_s = R150 is 0.15 Ohms
- R_s = R200 is 0.2 Ohms
- R_s = R220 is 0.22 Ohms

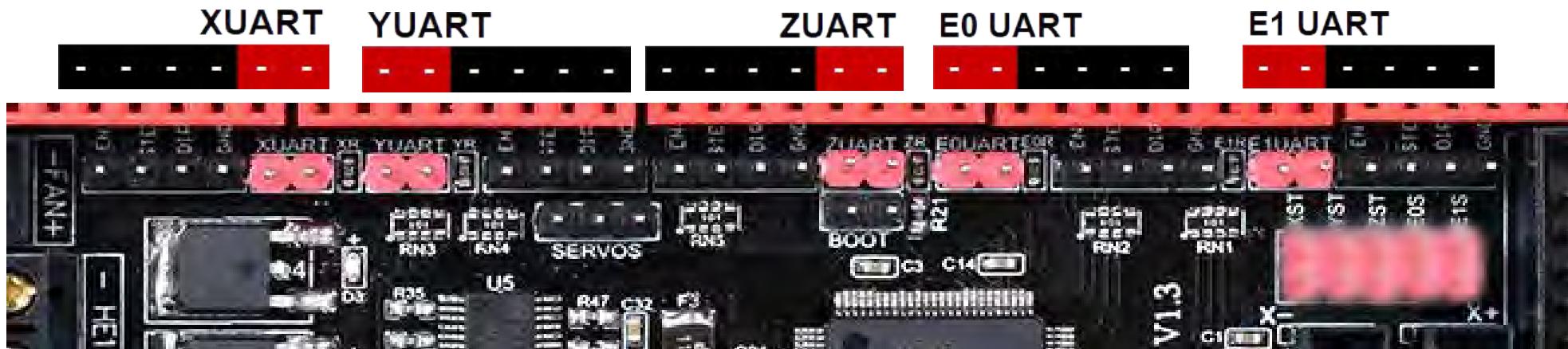
Stand-alone Mode

DRV8825

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

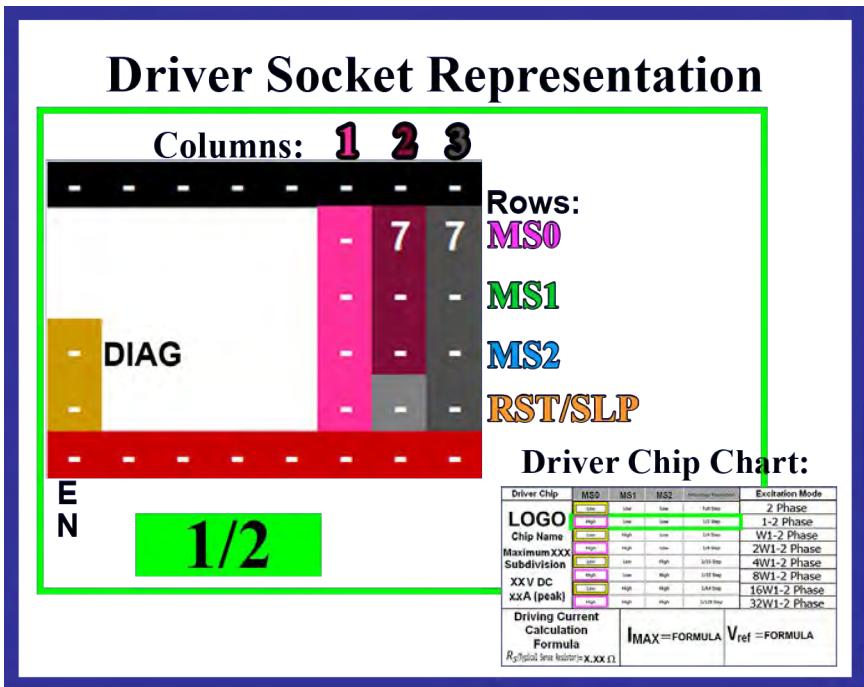


Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



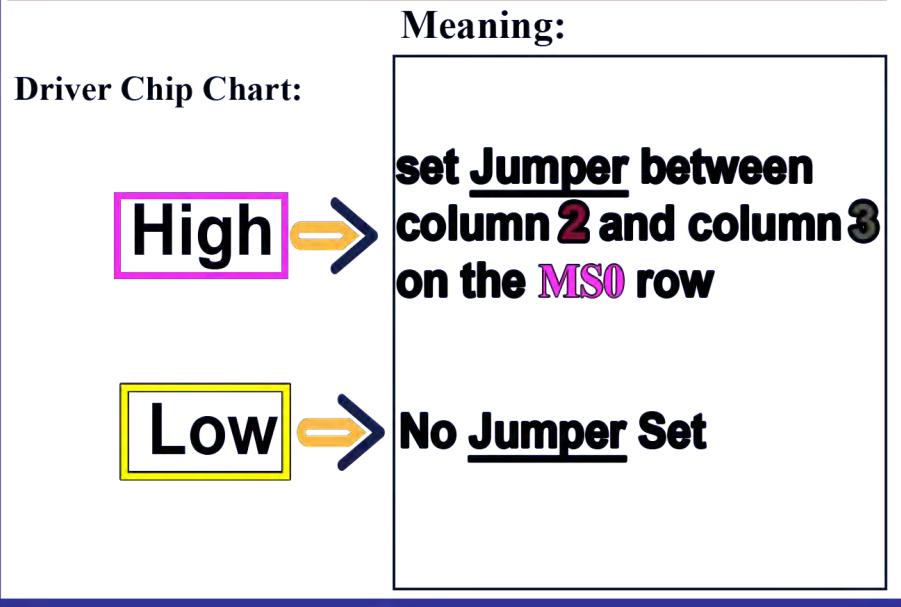
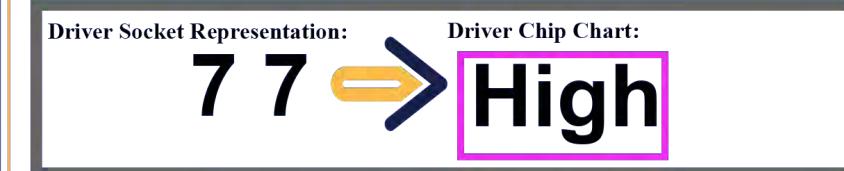
Stand-alone Mode

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

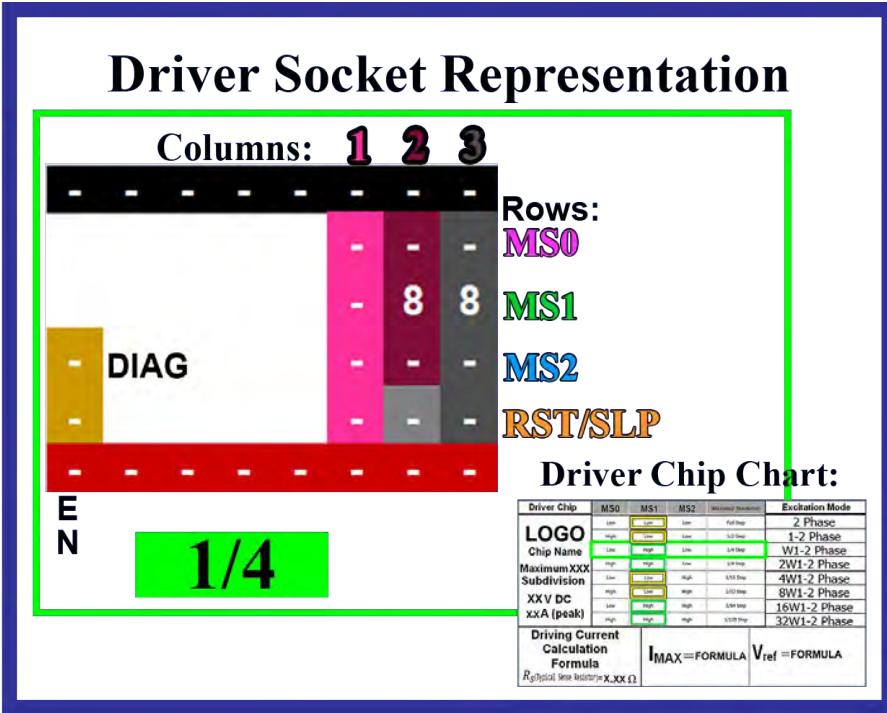


DRV8825

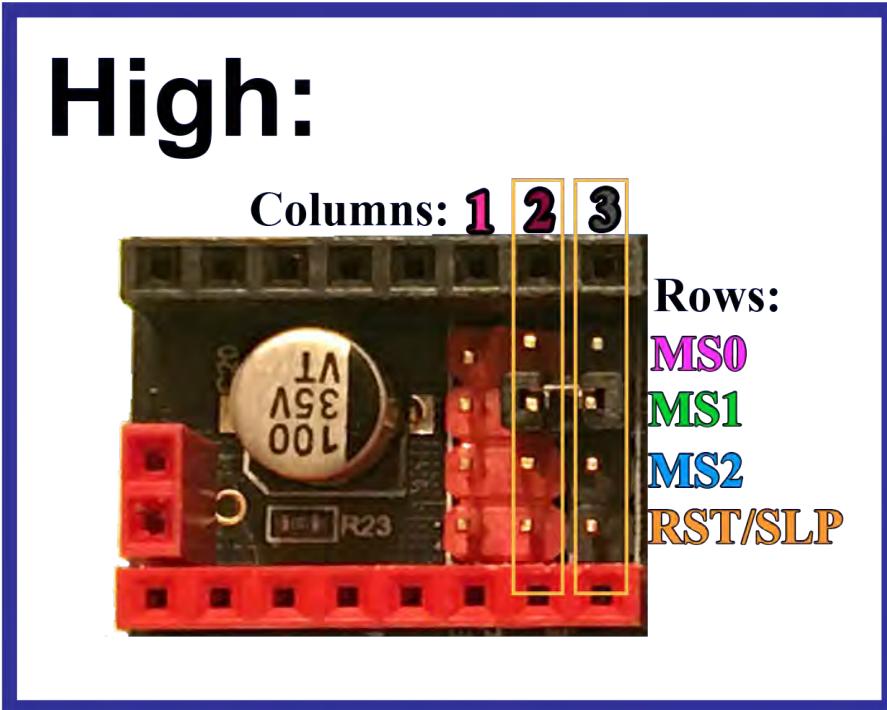
MS0 for Binary State Drivers:



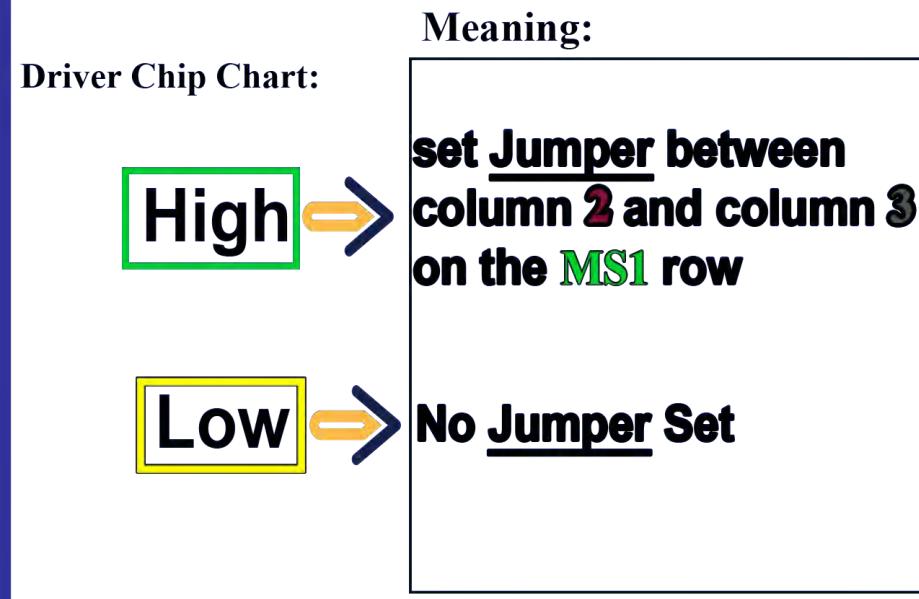
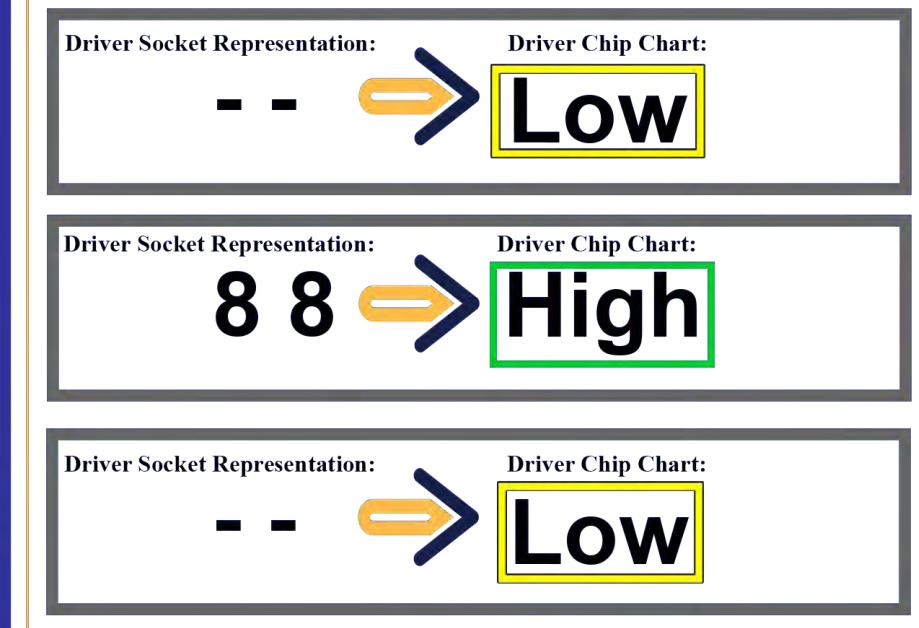
Stand-alone Mode



DRV8825

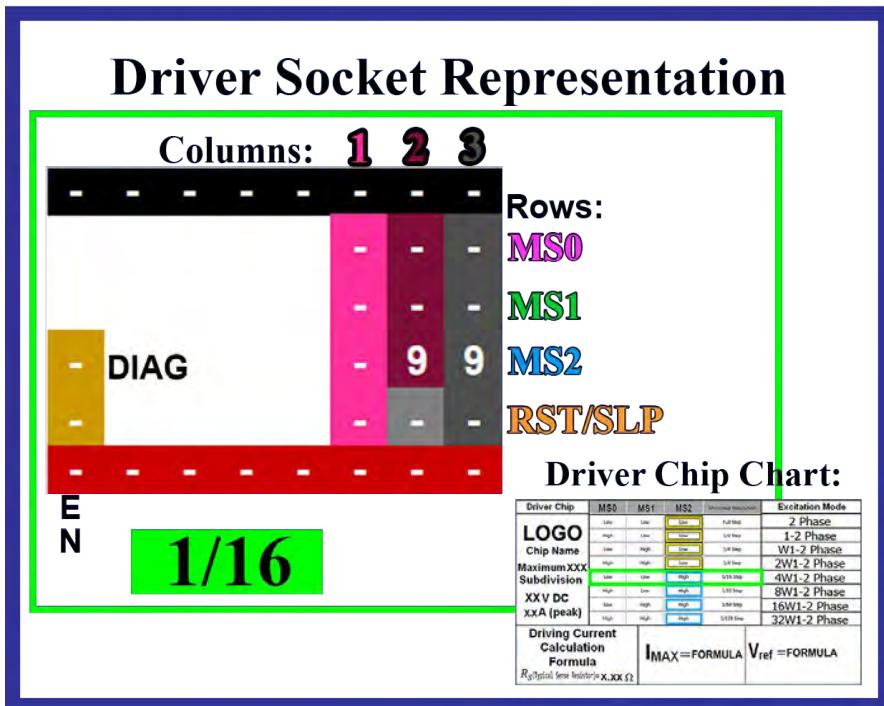


MS1 for Binary State Drivers:

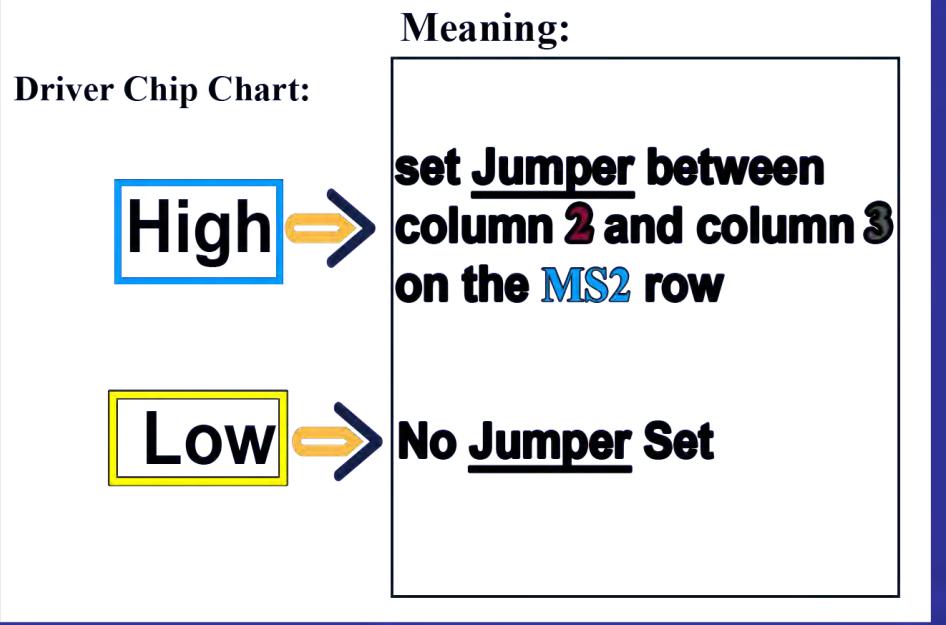
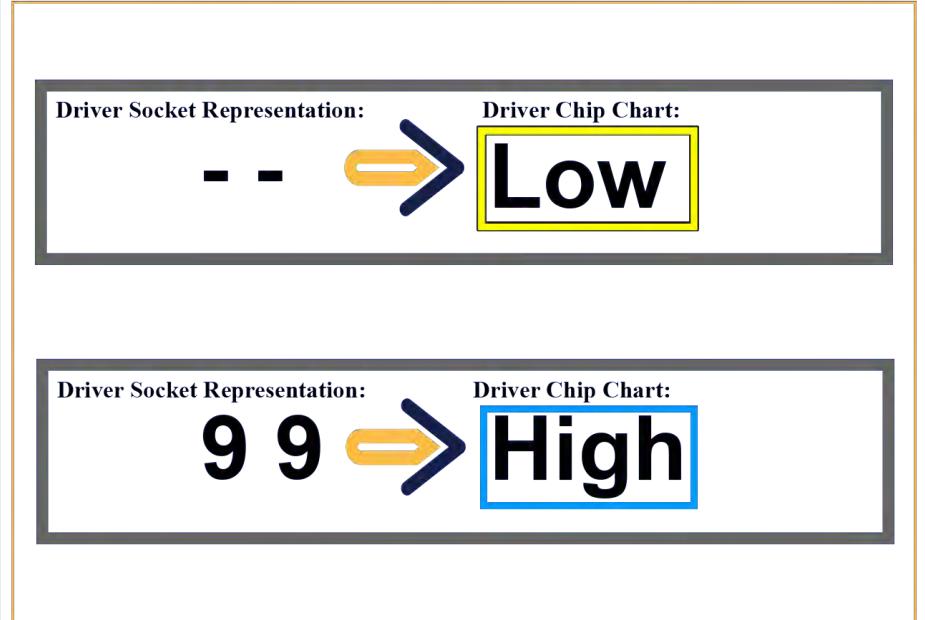


Stand-alone Mode

DRV8825



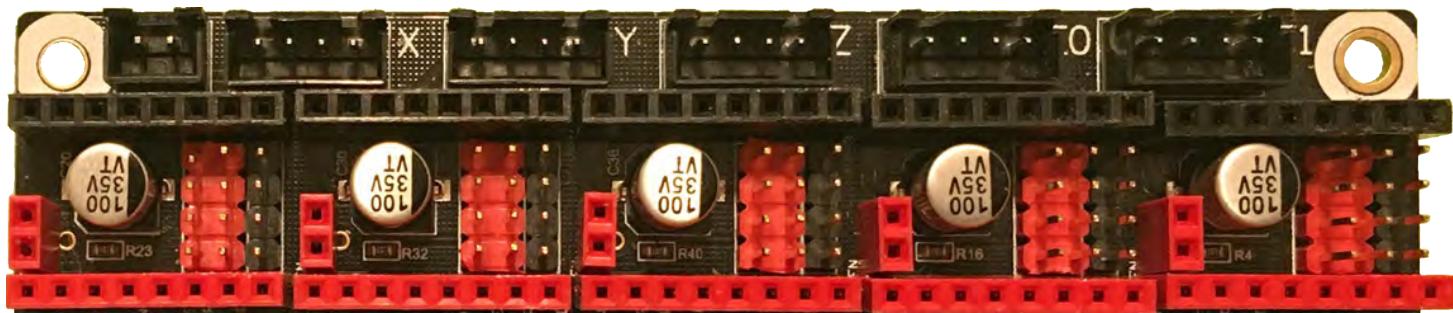
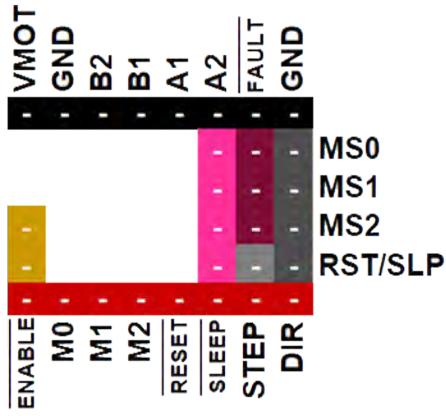
MS2 for Binary State Drivers:



Stand-alone Mode

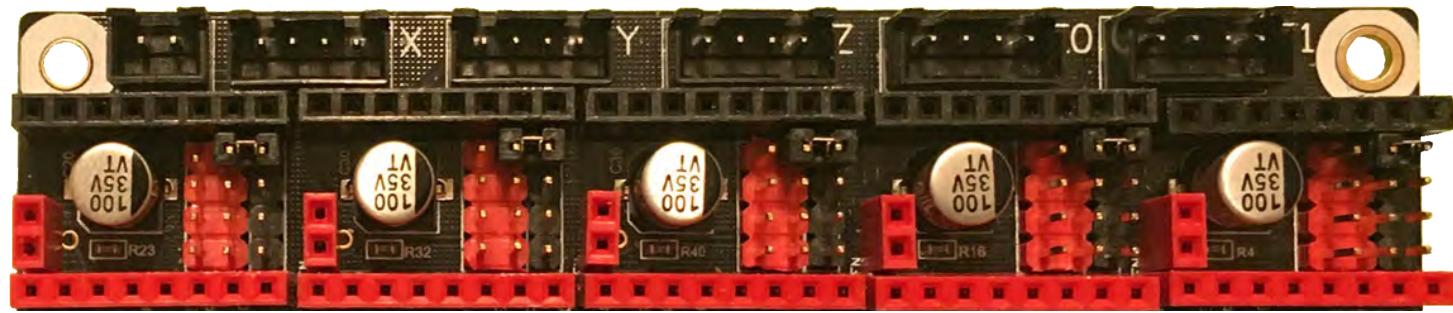
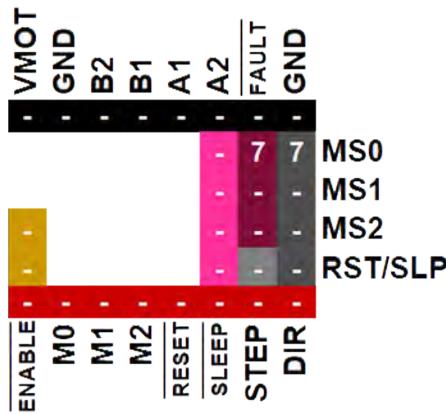
DRV8825

STEP



See [Appendix D](#) for legend

1 / 2

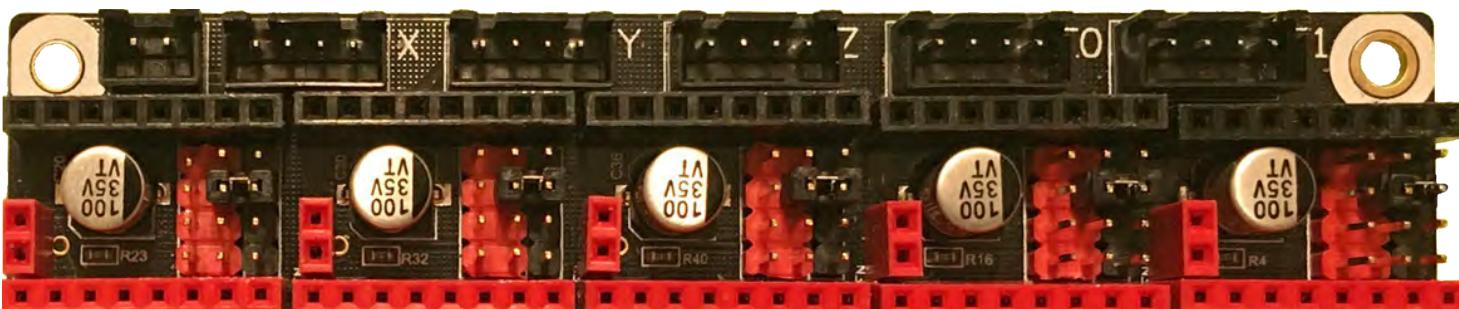
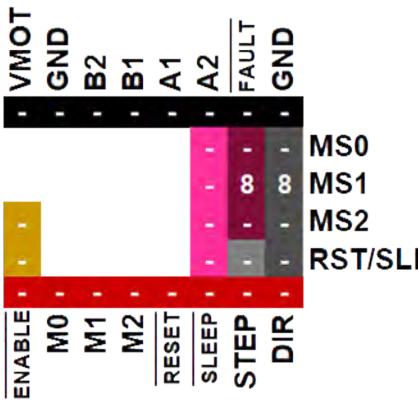


See [Appendix D](#) for legend

Stand-alone Mode

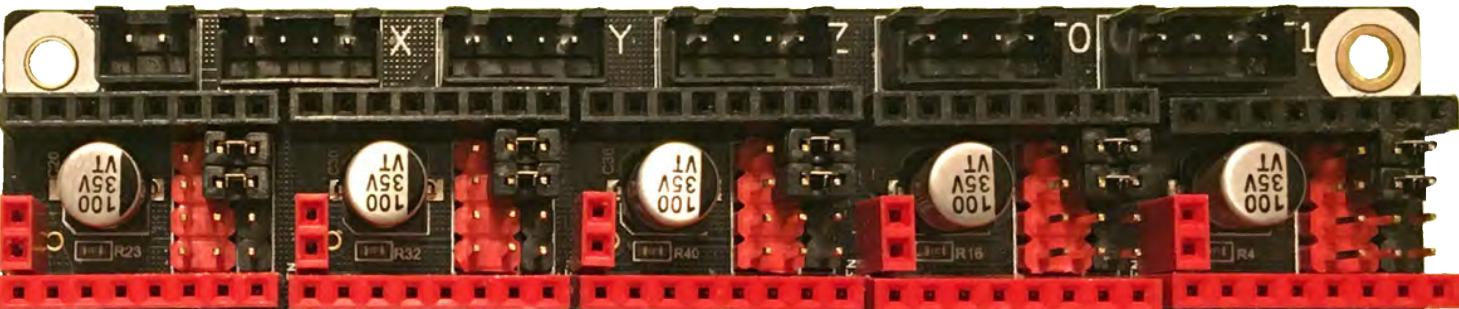
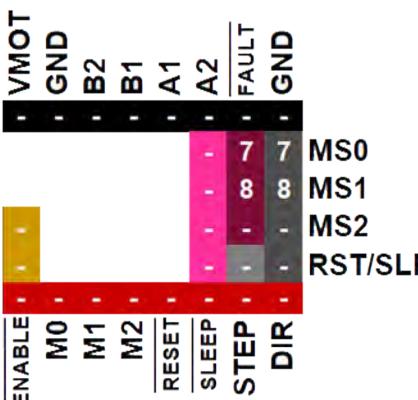
DRV8825

1 / 4



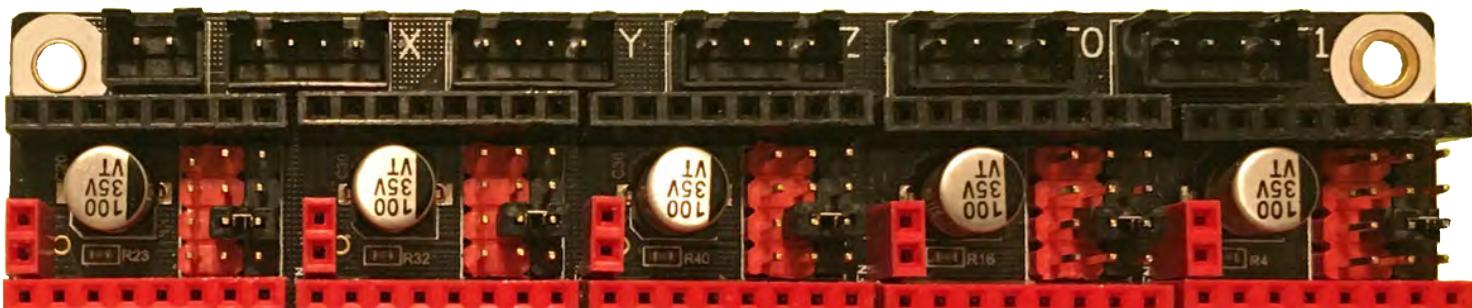
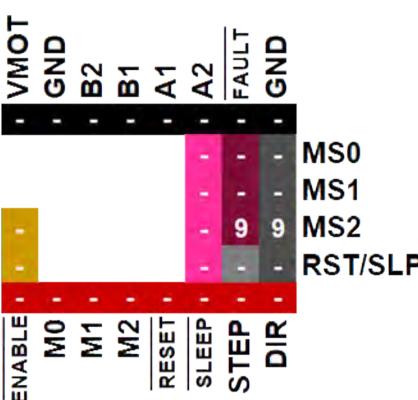
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



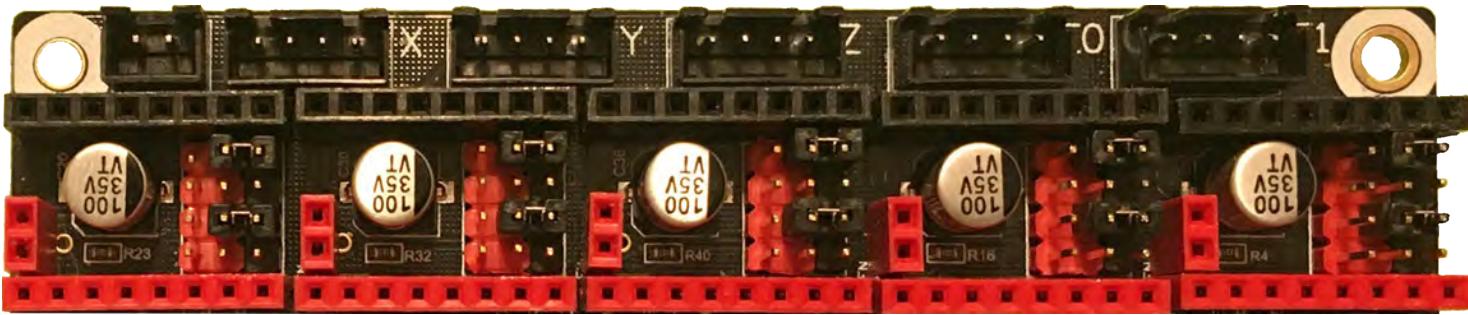
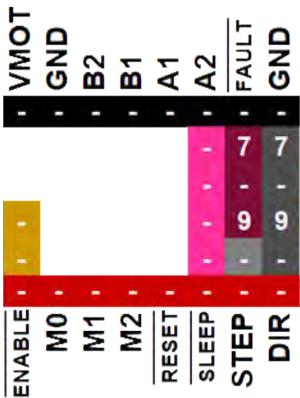
See [Appendix D](#) for legend

Stand-alone Mode

DRV8825

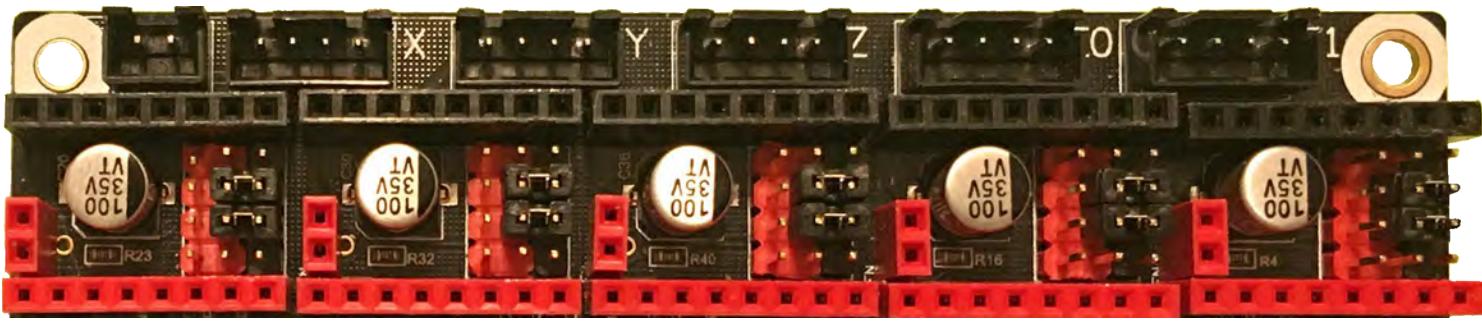
Note: All THREE of these settings will work for 1/32, choose your preference!!

1 / 32



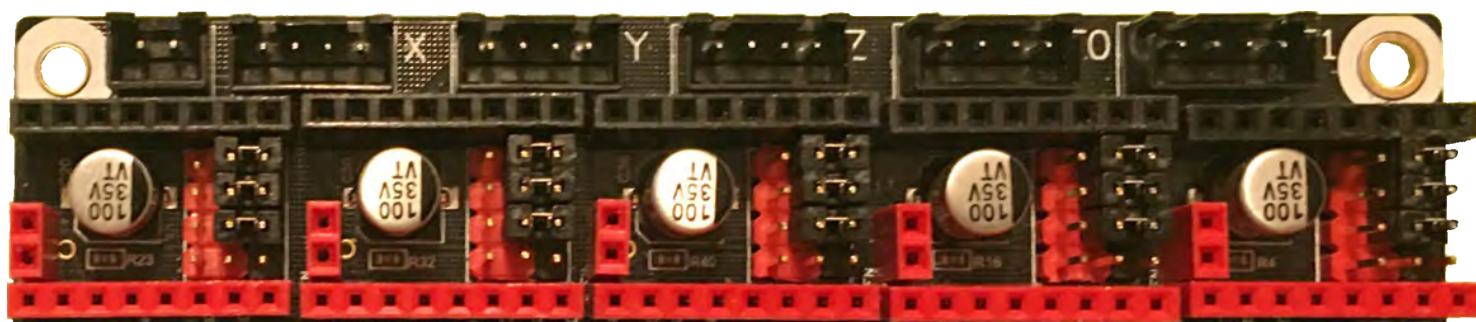
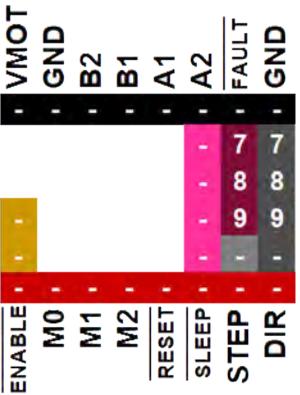
See [Appendix D](#) for legend

1 / 32



See [Appendix D](#) for legend

1 / 32

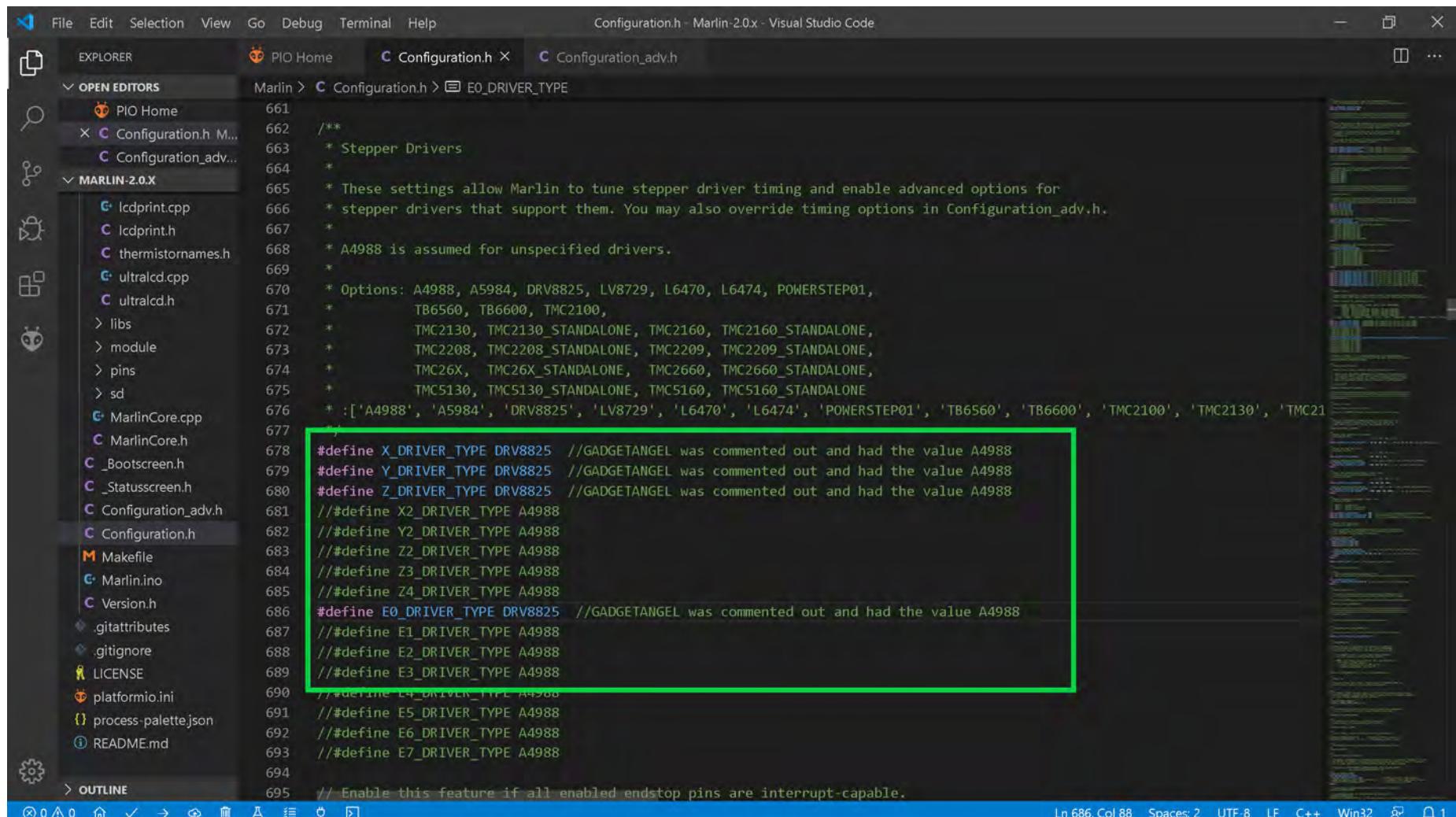


See [Appendix D](#) for legend

The (latest release of) Marlin Setup for DRV8825 Drivers

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

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



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

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

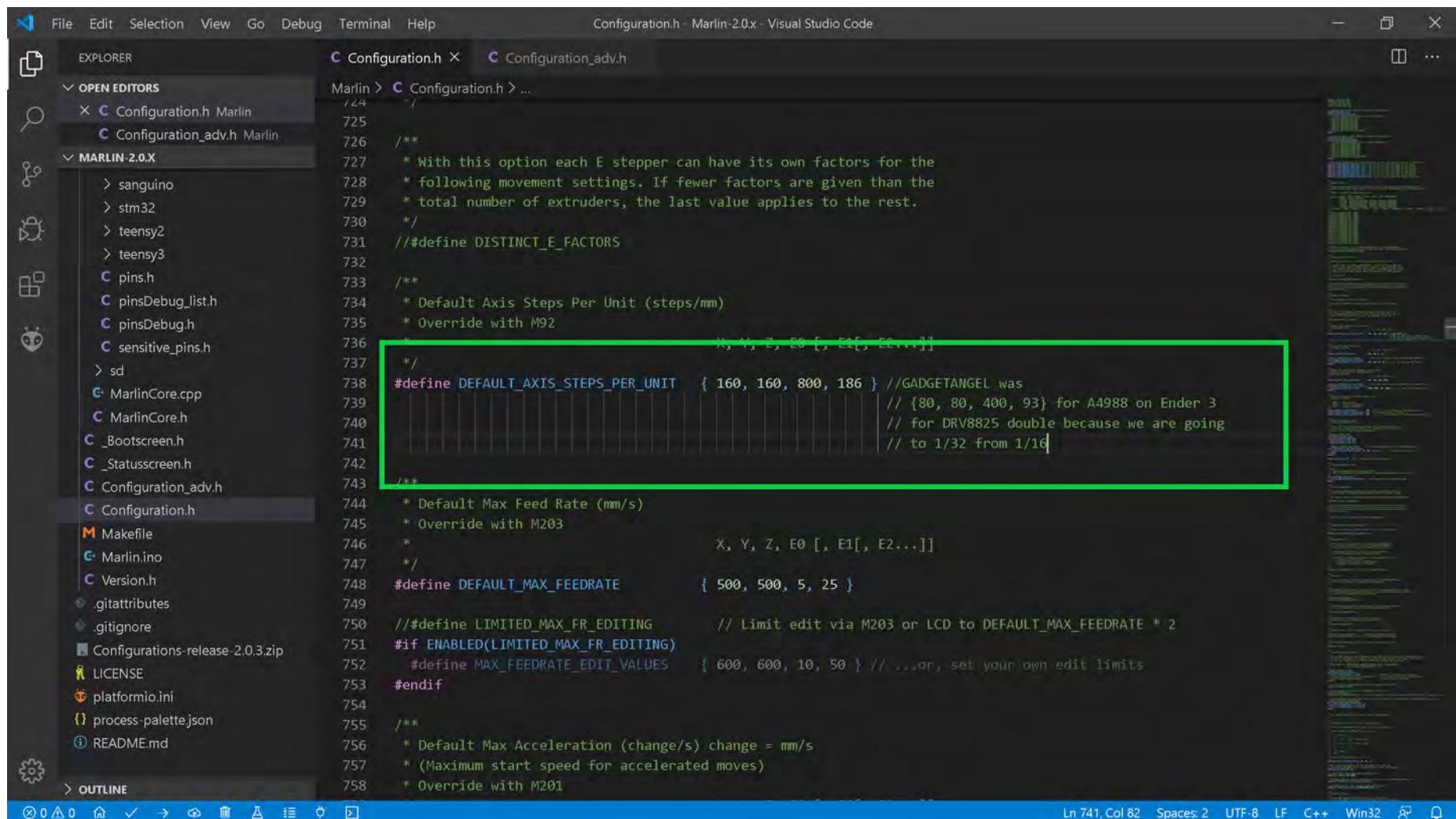
```

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

- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

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



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

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

The code editor's status bar at the bottom indicates: Ln 741, Col 82, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

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

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

- File Explorer:** Shows the project structure under "MARLIN-2.0.X".
- Editor:** Displays the "Configuration.h" file with the following code snippet highlighted:

```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif
```
- Terminal:** Shows the build logs:

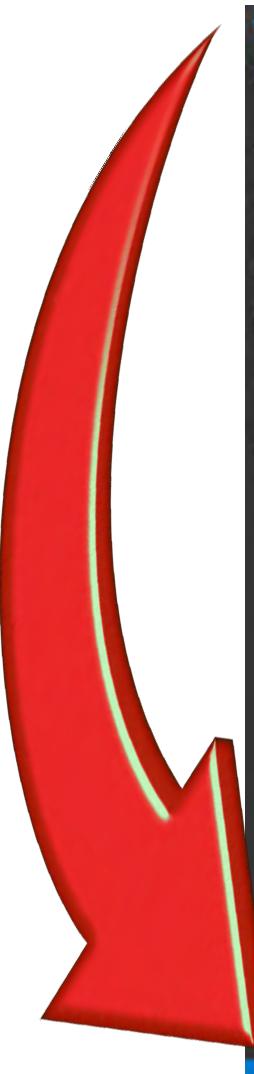
```
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
===== [SUCCESS] Took 130.61 seconds =====
```
- Build Results:** A table showing build environments and their status:

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
esp_dfu	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

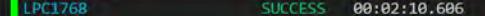
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for DRV8825 Drivers

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







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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

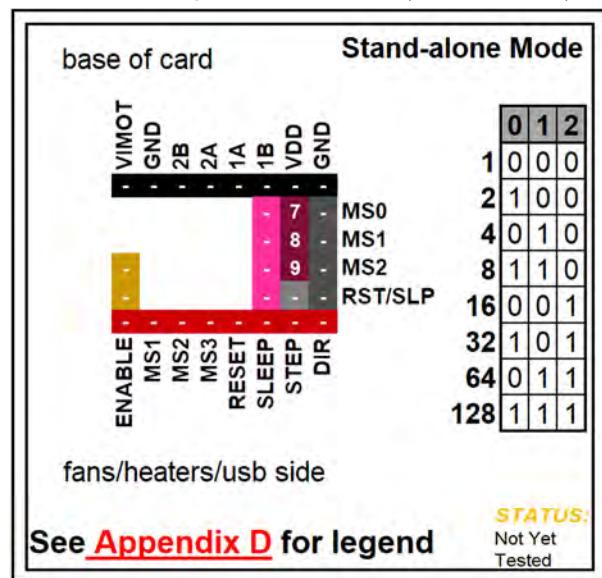
#define BAUDRATE 115200 // Enable the Bluetooth serial interface on AT90USB devices // #define BLUETOOTH // Choose the name from boards.h that matches your setup #ifndef MOTHERBOARD #define MOTHERBOARD BOARD_BTT_SKR_V1_3 #endif // Name displayed in the LCD "Ready" message and Info menu

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o Archiving .pio\build\LPC1768\libFrameworkArduino.a Linking .pio\build\LPC1768\firmware.elf Checking size .pio\build\LPC1768\firmware.elf Advanced Memory Usage is available via "PlatformIO Home > Project Inspect" RAM: [====] 42.5% (used 13908 bytes from 32736 bytes) Flash: [====] 35.0% (used 166124 bytes from 475136 bytes) Building .pio\build\LPC1768\firmware.bin [SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

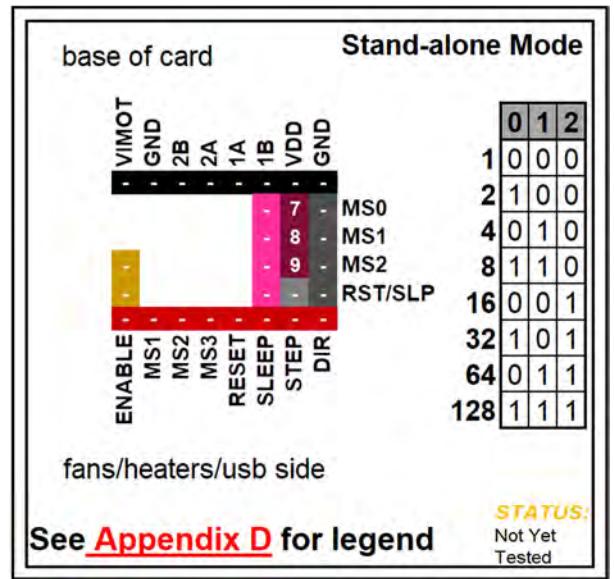
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3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

BIQU LV8729

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

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

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



Driver Chip Chart:

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

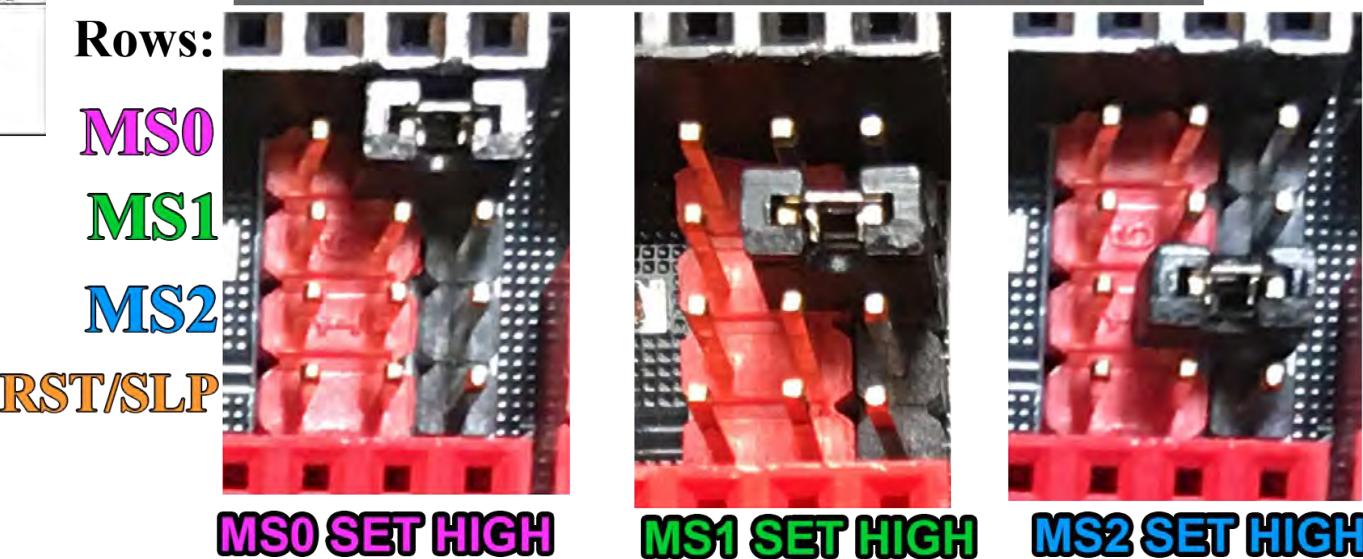
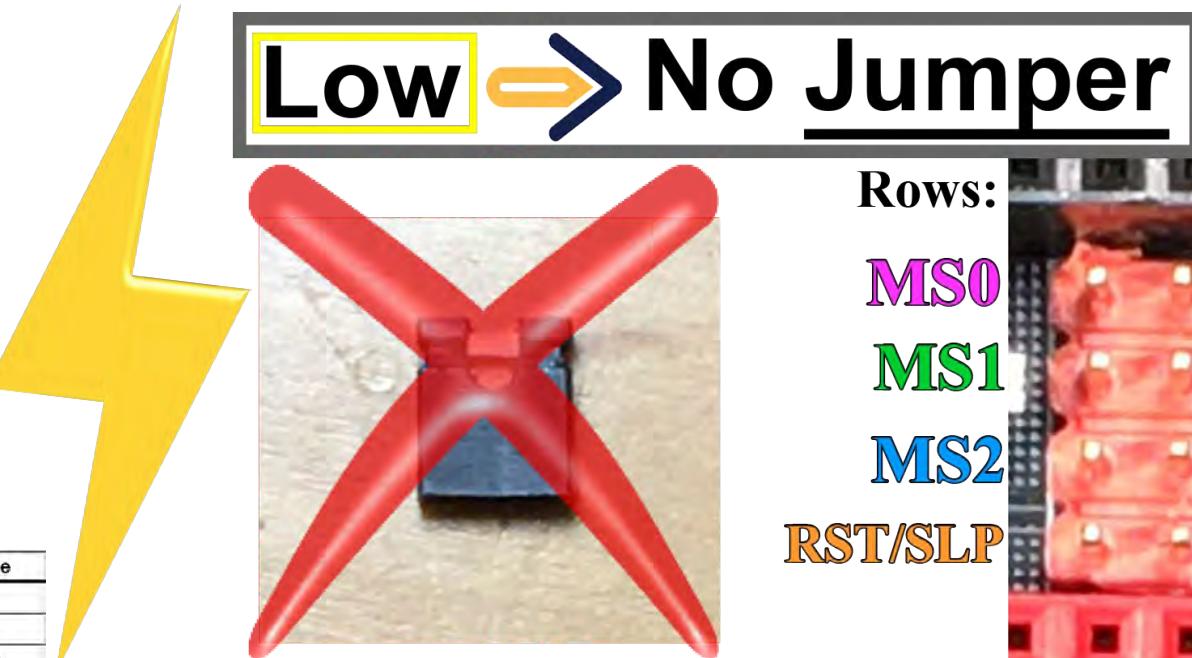
Driving Current Calculation Formula $I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

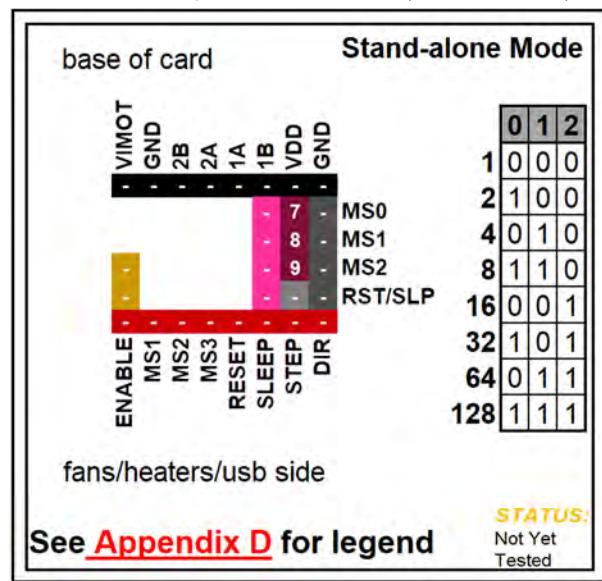
$R_S(\text{Typical Sense Resistor}) = X.XX \Omega$



Biqu LV8729

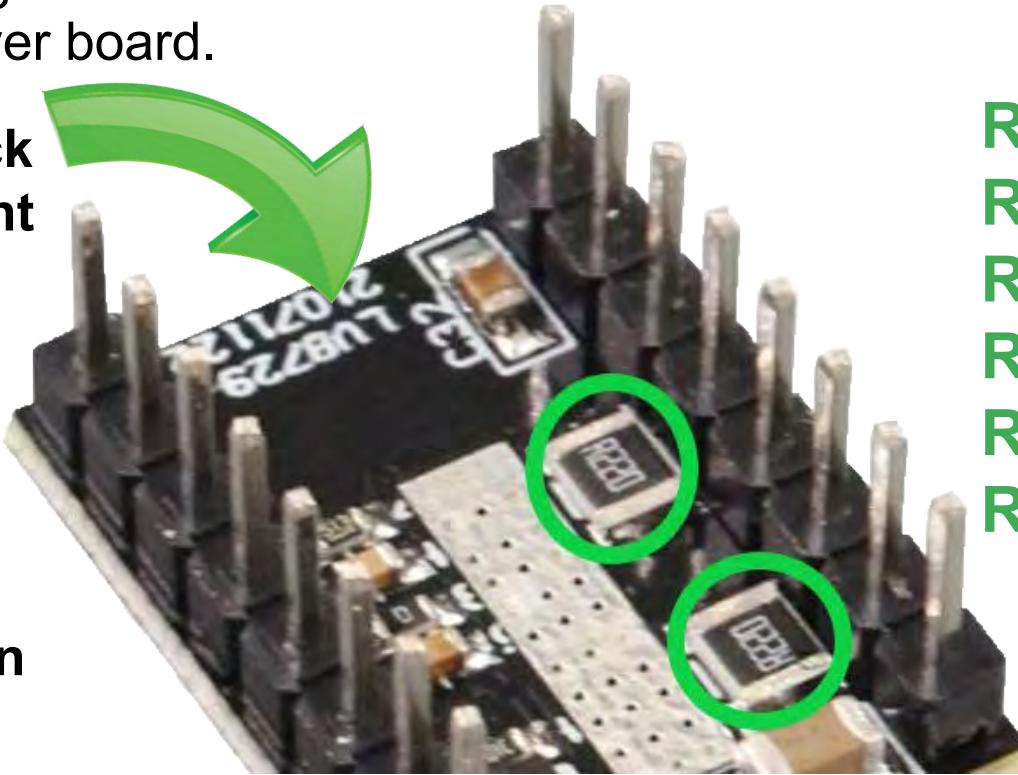
SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers





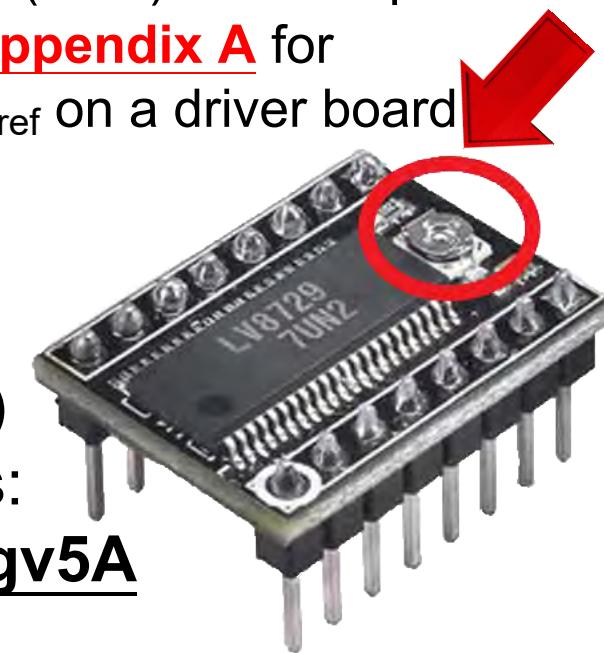
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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



BIQU LV8729

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



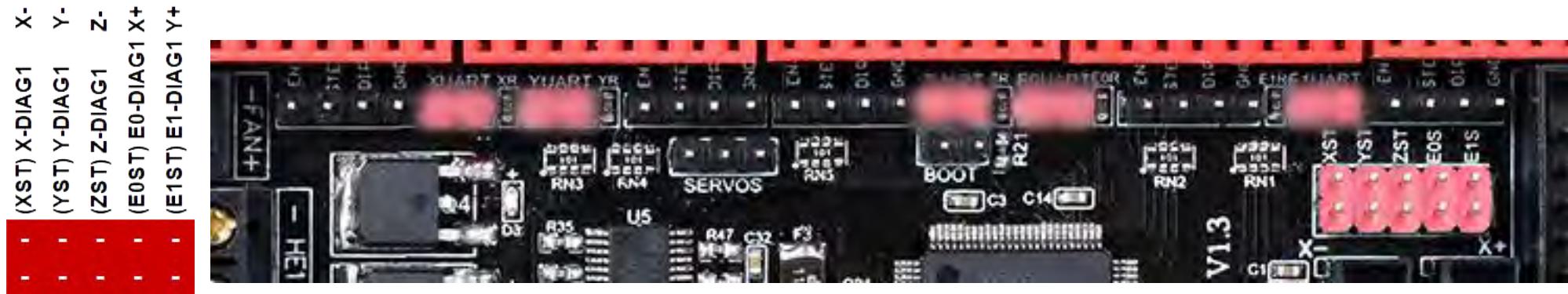
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

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

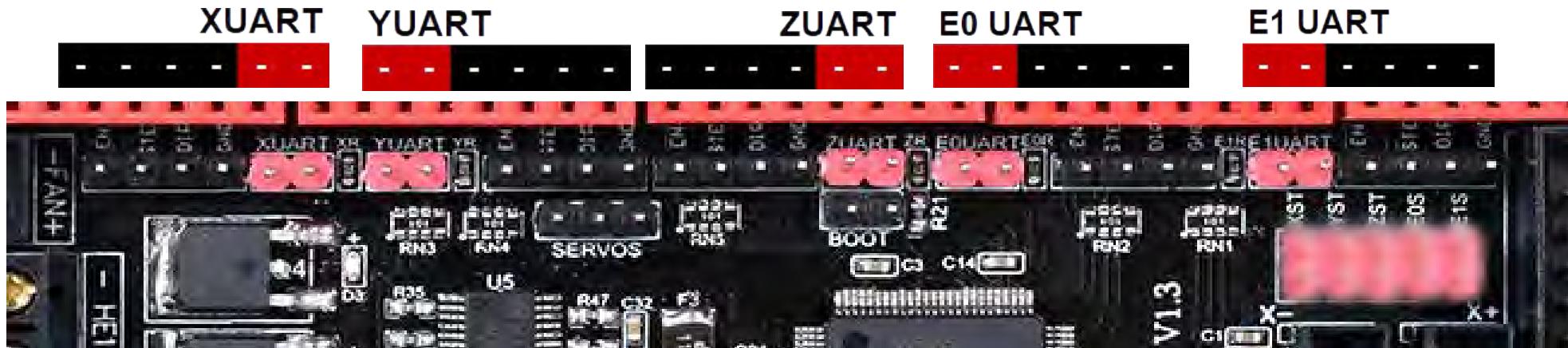
Stand-alone Mode

BIQU LV8729

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



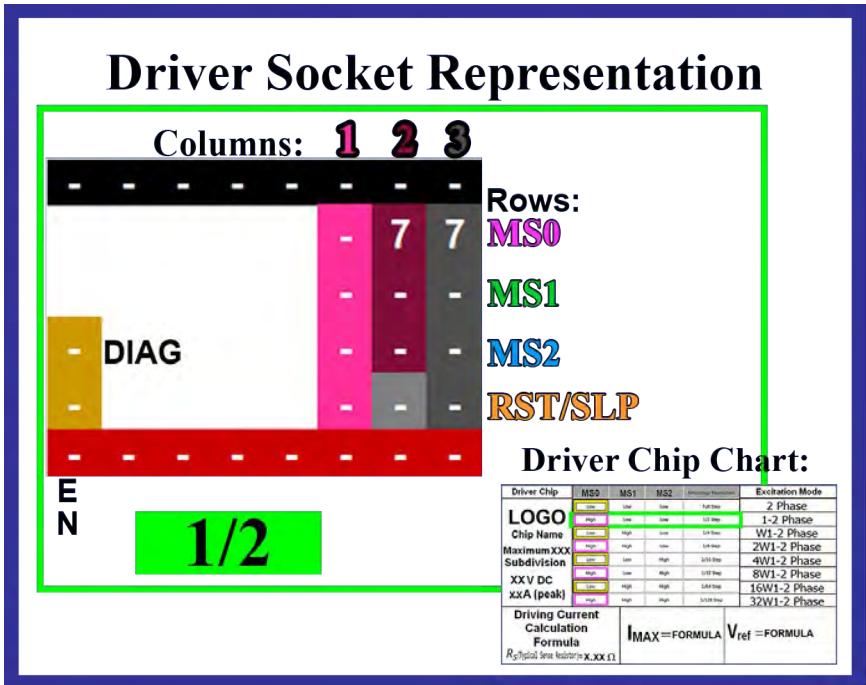
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



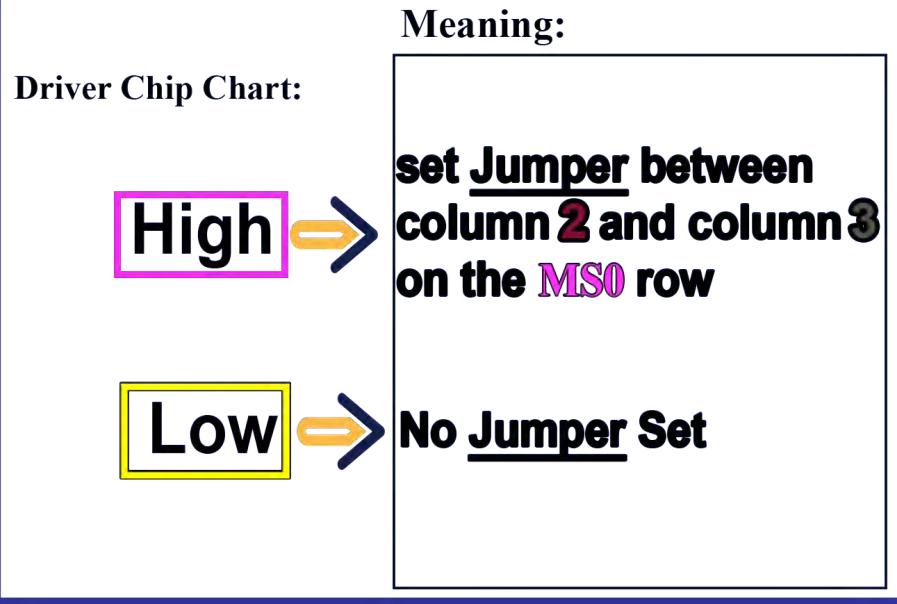
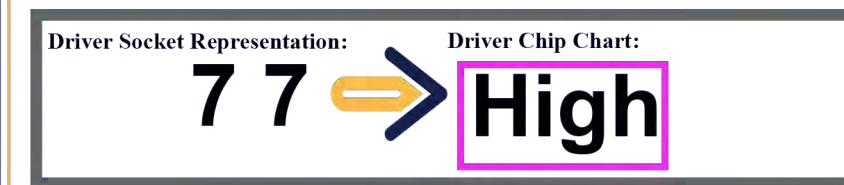
Stand-alone Mode

BIQU LV8729

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

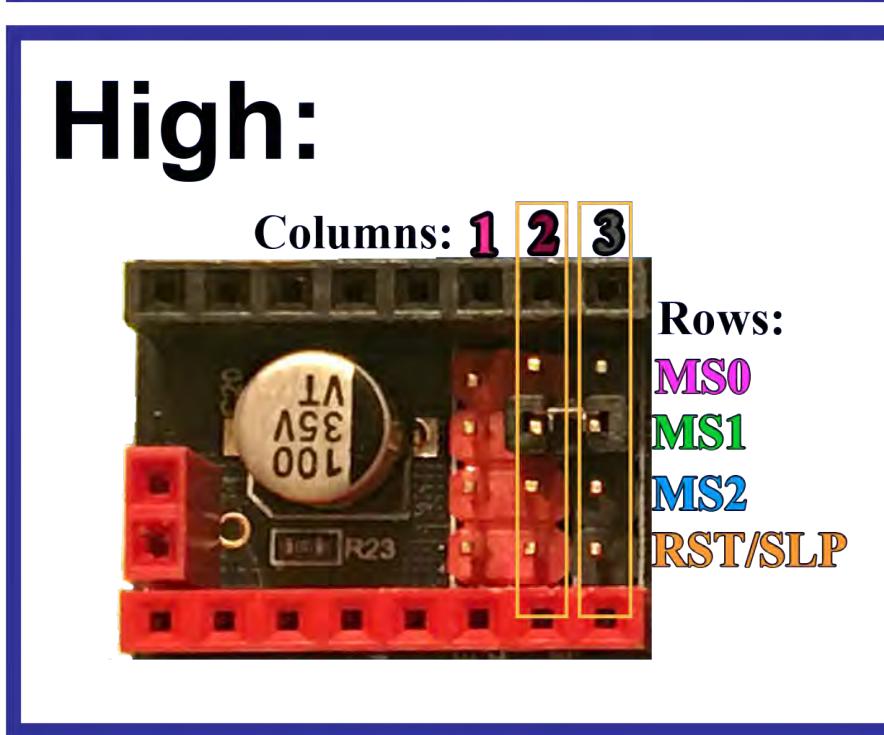
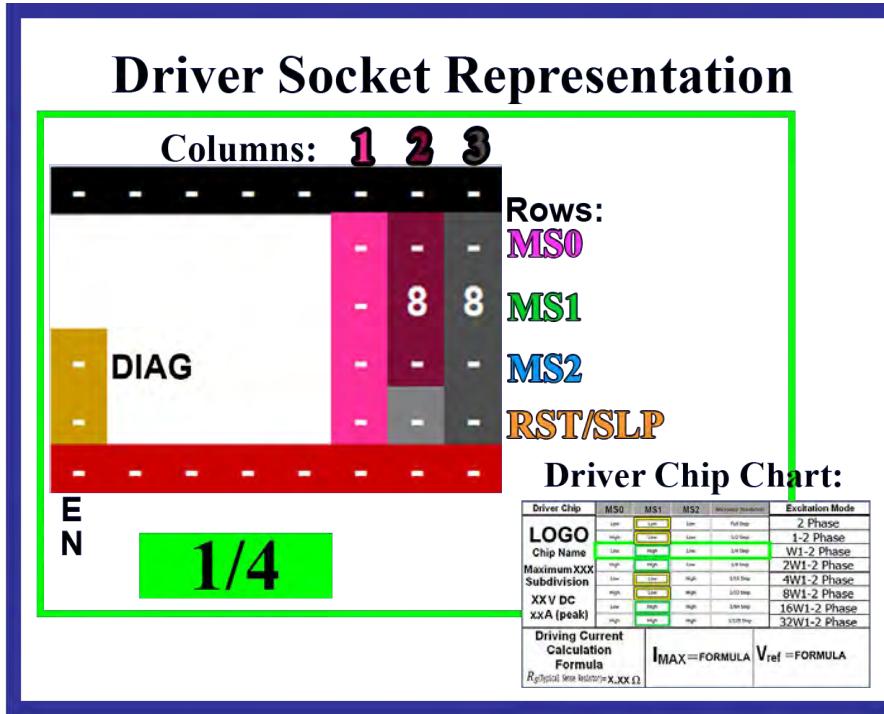


MS0 for Binary State Drivers:

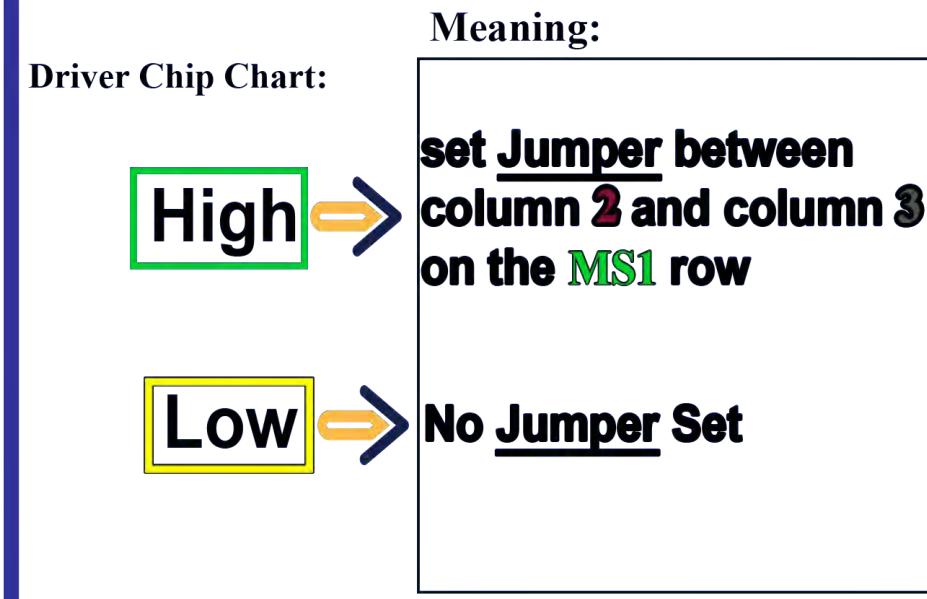
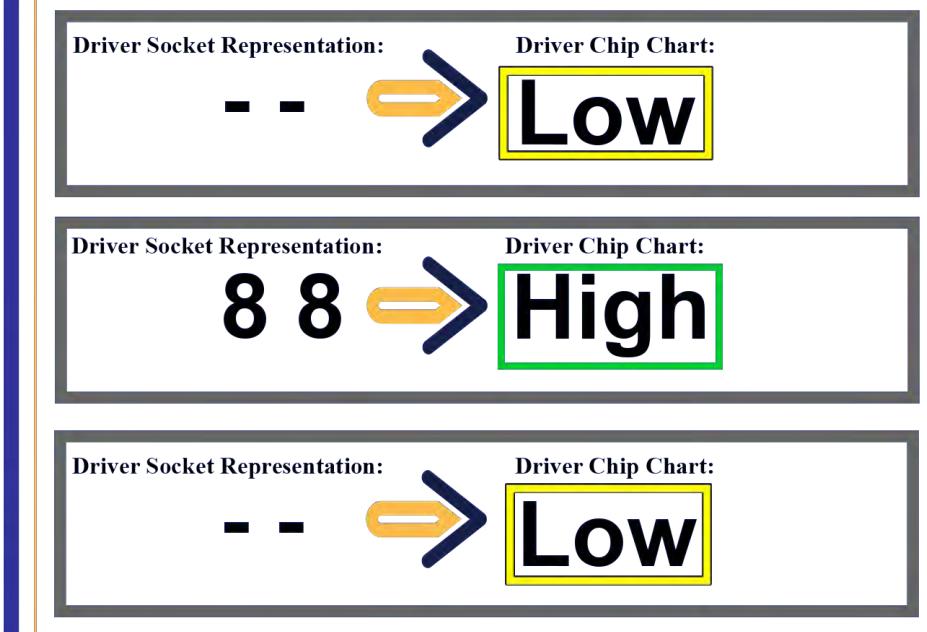


Stand-alone Mode

BIQU LV8729

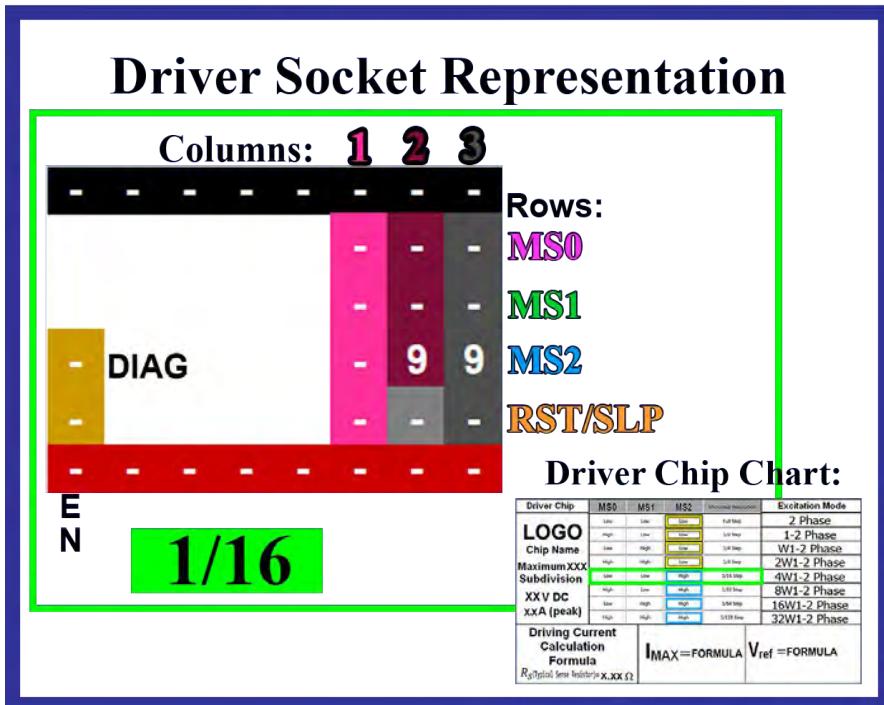


MS1 for Binary State Drivers:

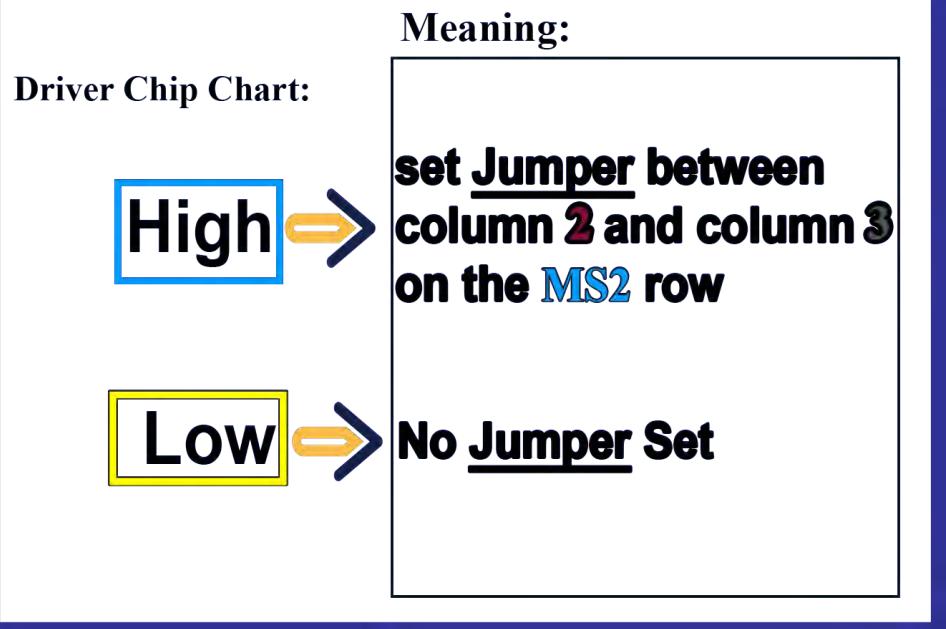
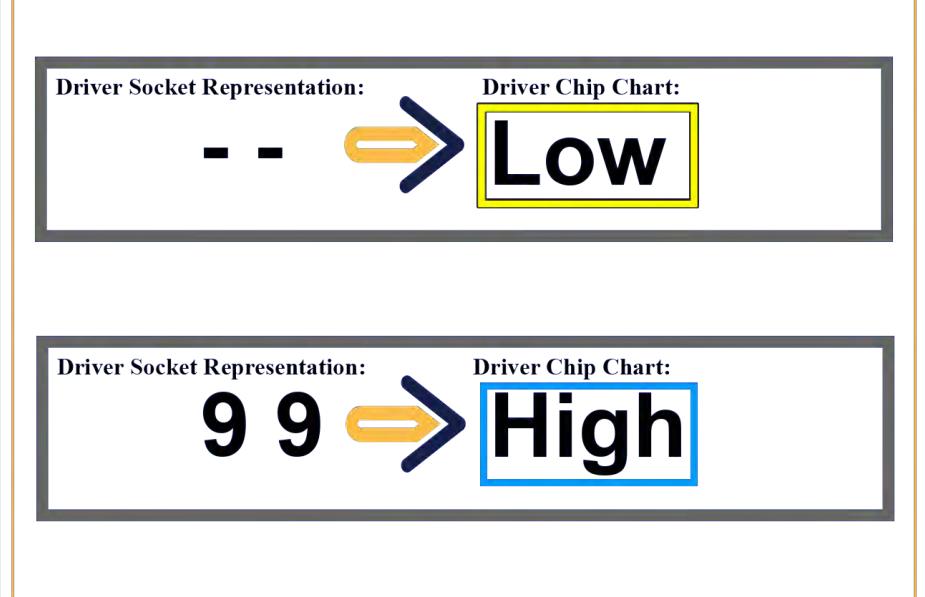


Stand-alone Mode

BIQU LV8729



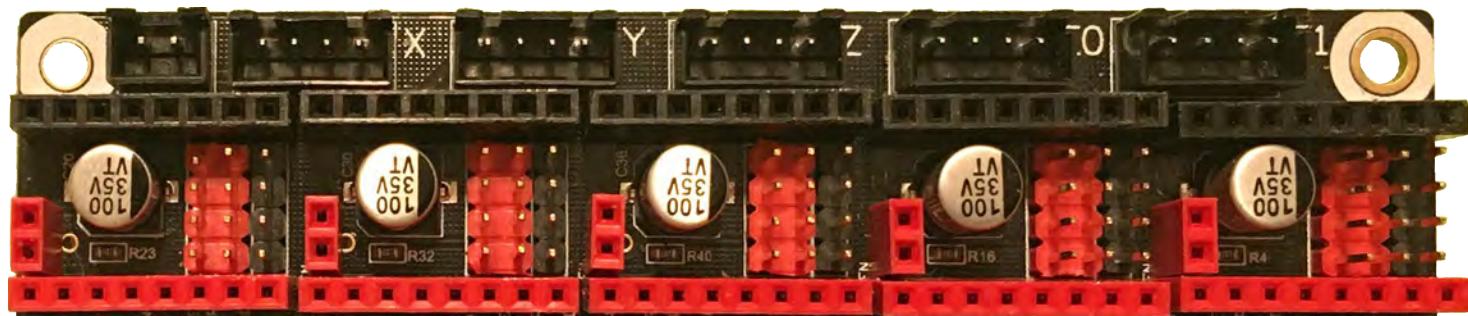
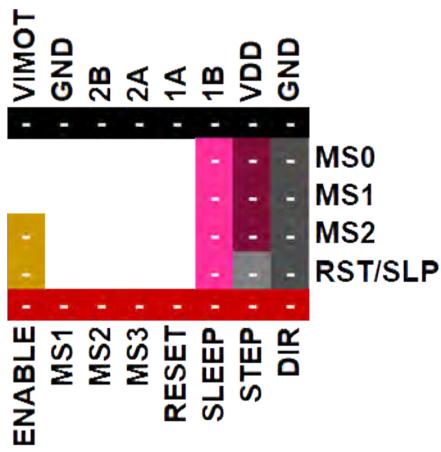
MS2 for Binary State Drivers:



Stand-alone Mode

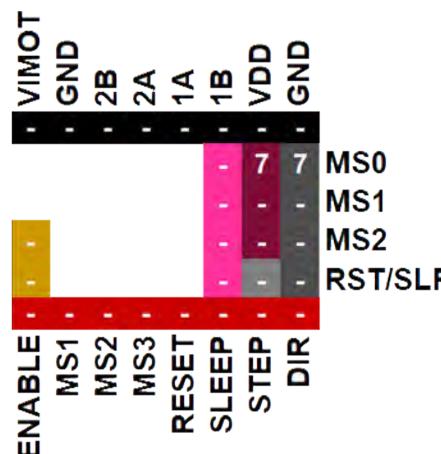
BIQU LV8729

STEP



See [Appendix D](#) for legend

1 / 2

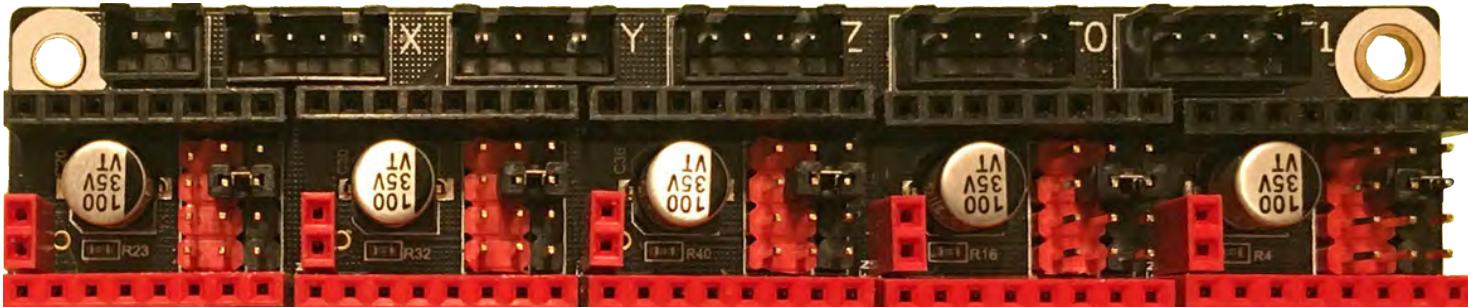
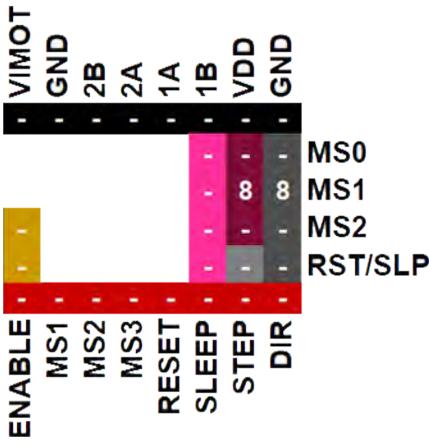


See [Appendix D](#) for legend

Stand-alone Mode

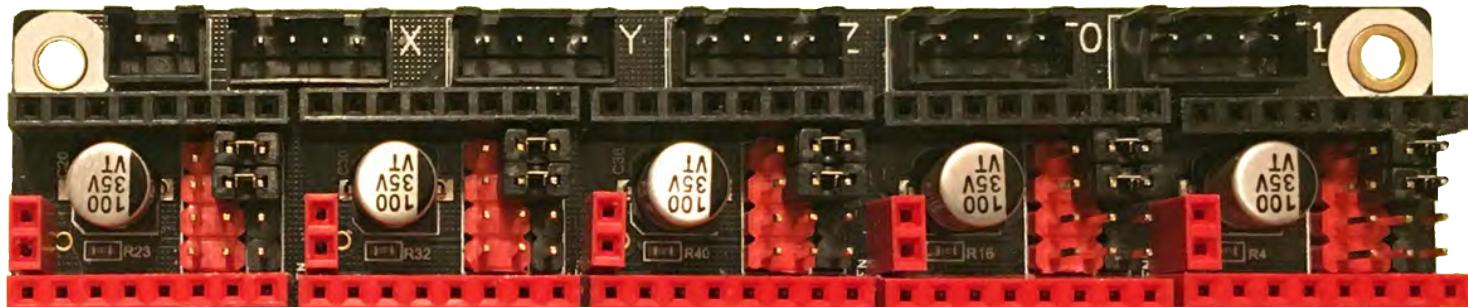
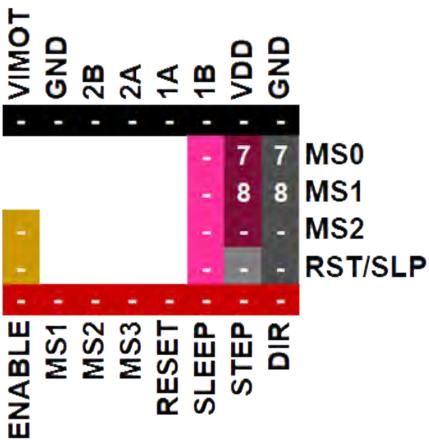
BIQU LV8729

1 / 4



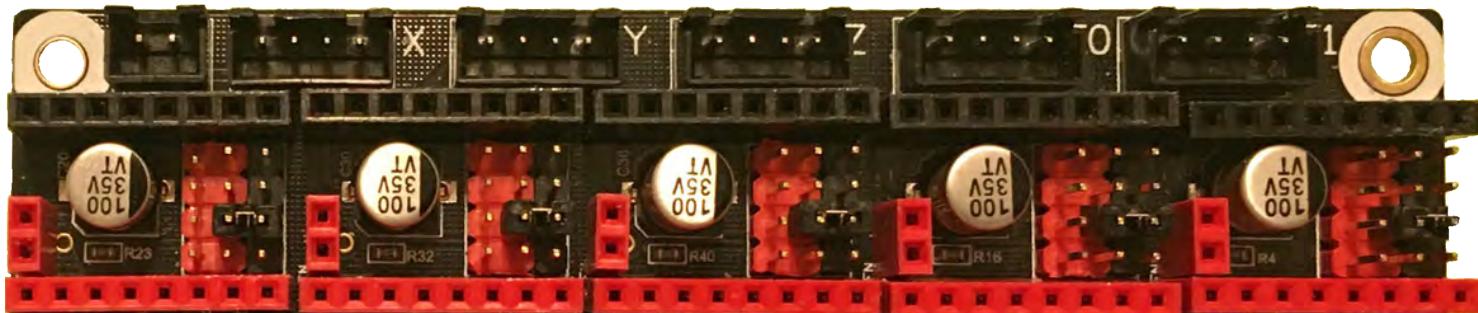
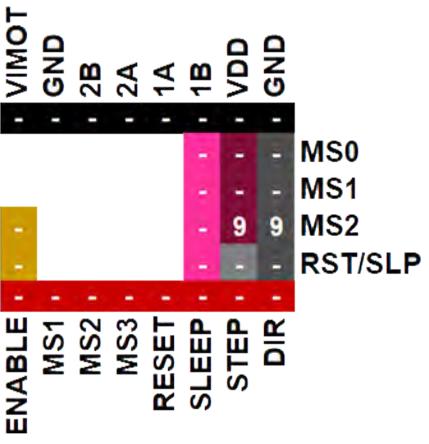
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16

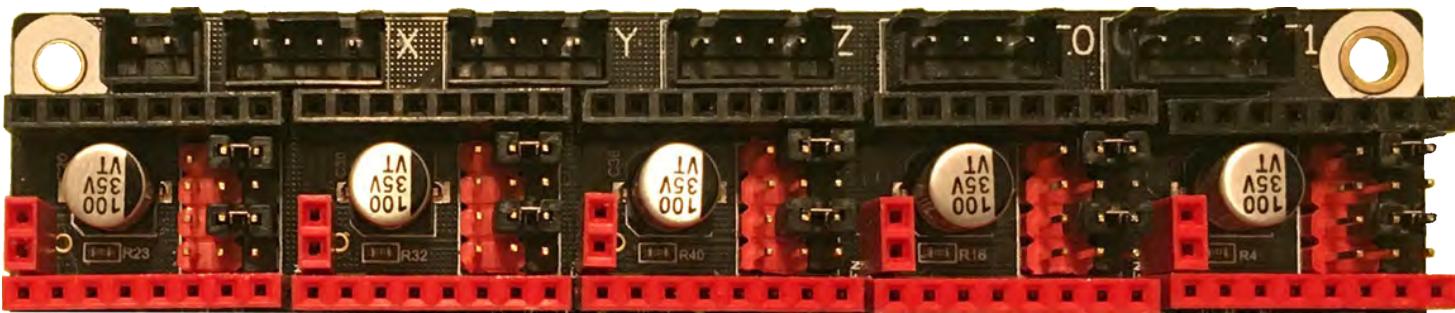
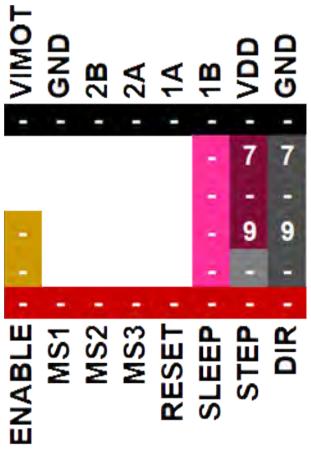


See [Appendix D](#) for legend

Stand-alone Mode

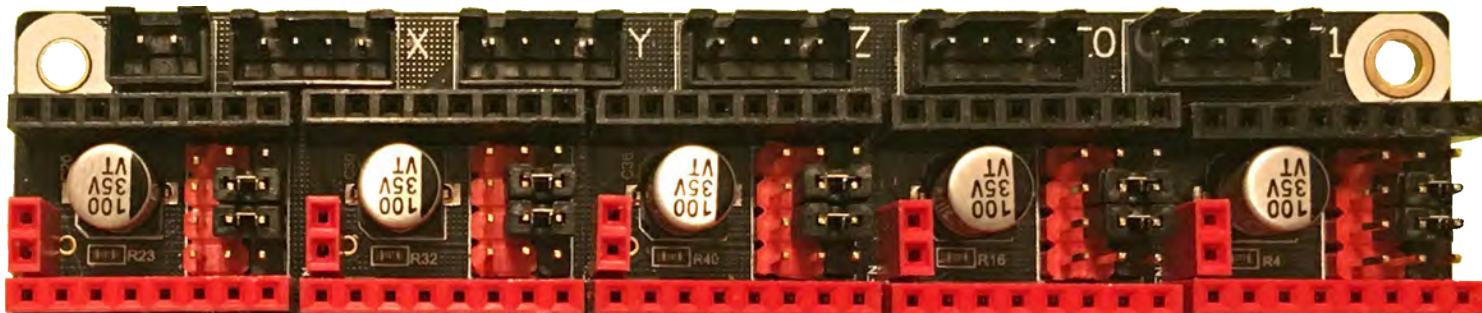
BIQU LV8729

1 / 32



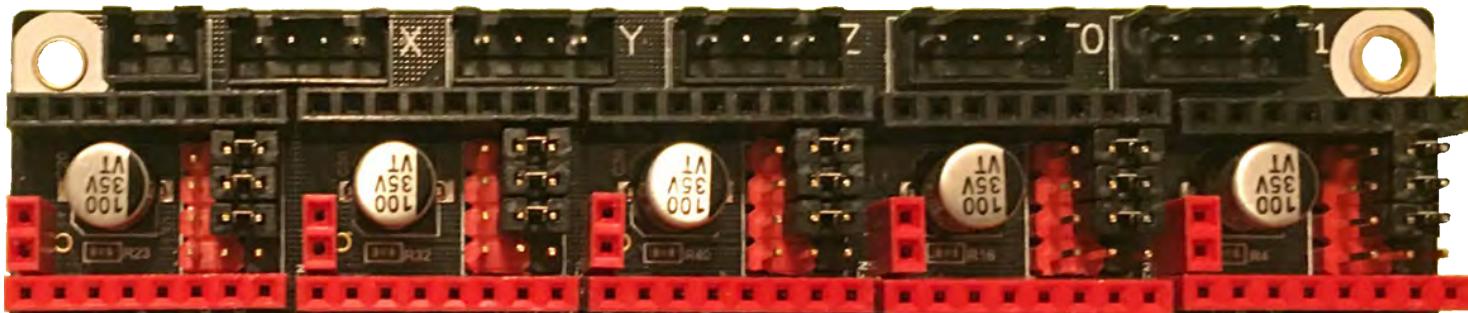
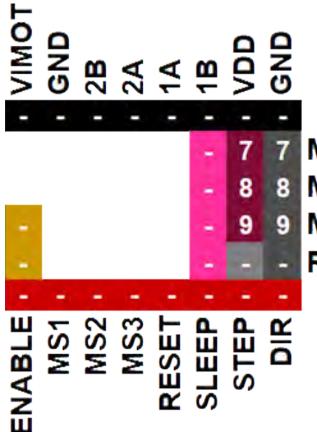
See [Appendix D](#) for legend

1 / 64



See [Appendix D](#) for legend

1 / 128

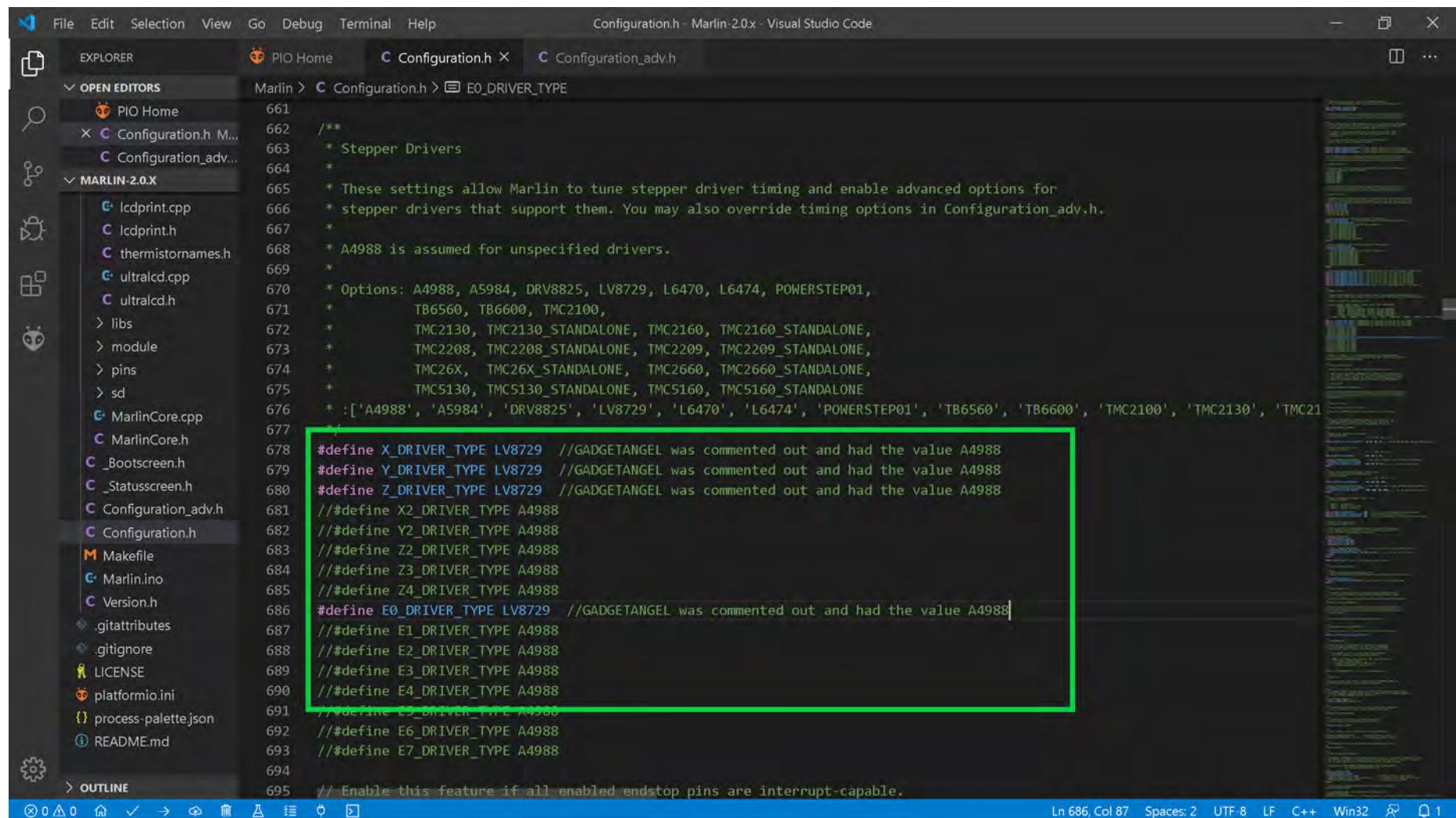


See [Appendix D](#) for legend

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

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

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



```

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

EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
MARLIN-2.0.X
  LCDprint.cpp
  LCDprint.h
  thermistornames.h
  ultralcd.cpp
  ultralcd.h
  > libs
  > module
  > pins
  > sd
  MarlinCore.cpp
  MarlinCore.h
  Bootscreen.h
  Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md

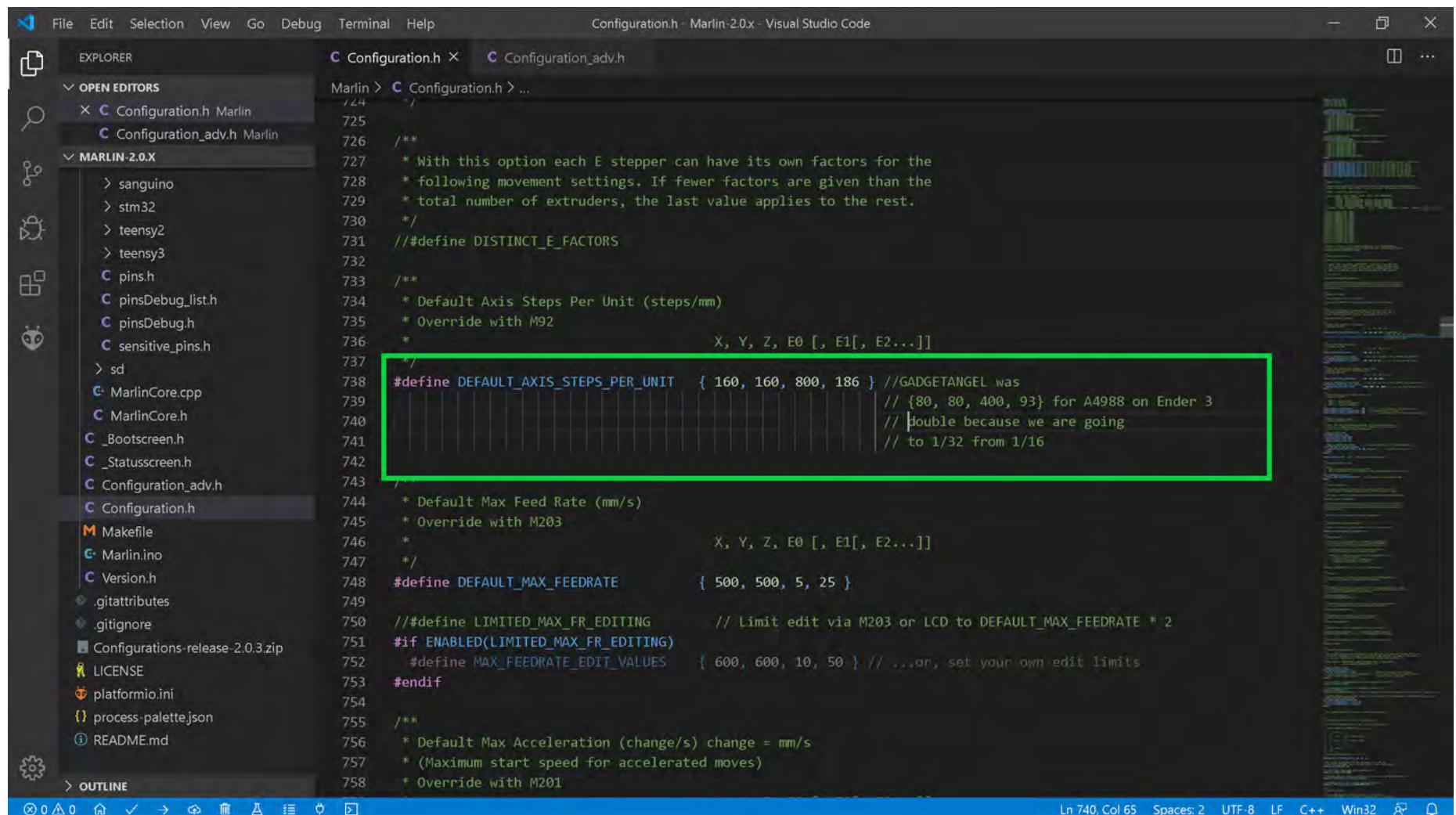
  661
  662 /**
  663 * Stepper Drivers
  664 *
  665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  667 *
  668 * A4988 is assumed for unspecified drivers.
  669 *
  670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
  671 *           TB6560, TB6600, TMC2100,
  672 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
  673 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
  674 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
  675 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
  676 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2
  677 */
  678 #define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
  679 #define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
  680 #define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
  681 //#define X2_DRIVER_TYPE A4988
  682 //#define Y2_DRIVER_TYPE A4988
  683 //#define Z2_DRIVER_TYPE A4988
  684 //#define Z3_DRIVER_TYPE A4988
  685 //#define Z4_DRIVER_TYPE A4988
  686 #define E0_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
  687 //#define E1_DRIVER_TYPE A4988
  688 //#define E2_DRIVER_TYPE A4988
  689 //#define E3_DRIVER_TYPE A4988
  690 //#define E4_DRIVER_TYPE A4988
  691 //#define E5_DRIVER_TYPE A4988
  692 //#define E6_DRIVER_TYPE A4988
  693 //#define E7_DRIVER_TYPE A4988
  694
  695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



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

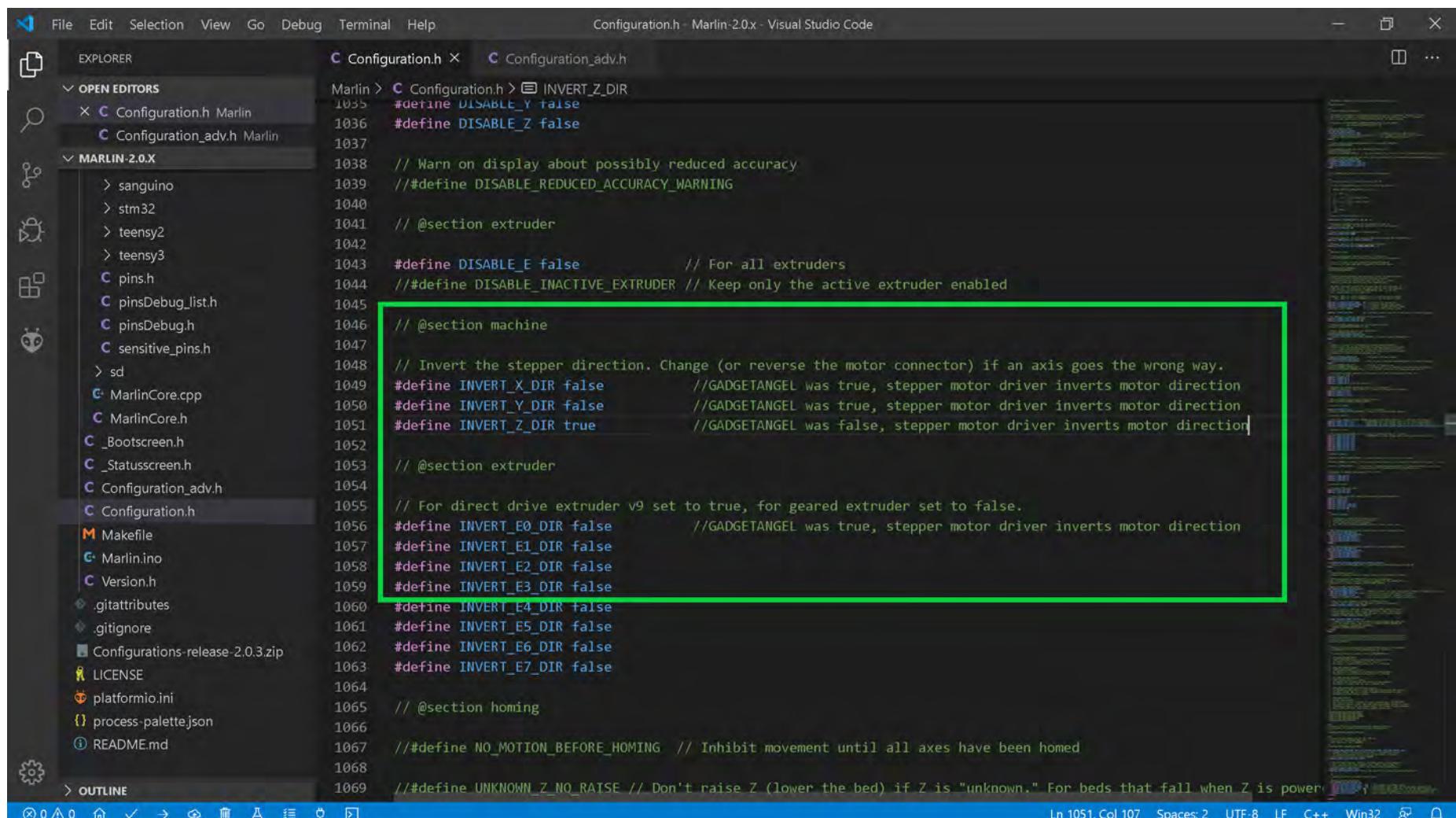
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



```

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

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  C pins.h
  C pinsDebug_list.h
  C pinsDebug.h
  C sensitive_pins.h
  > sd
  G MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C _Statusscreen.h
  C Configuration_adv.h
  C Configuration.h
  M Makefile
  G Marlin.ino
  C Version.h
  .gitattributes
  .gitignore
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
Ln 1051, Col 107  Spaces: 2  UTF-8  LF  C++  Win32  ⚡  🔍

Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

```

- Go to the next page.

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

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

```

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

EXPLORER
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin

MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md

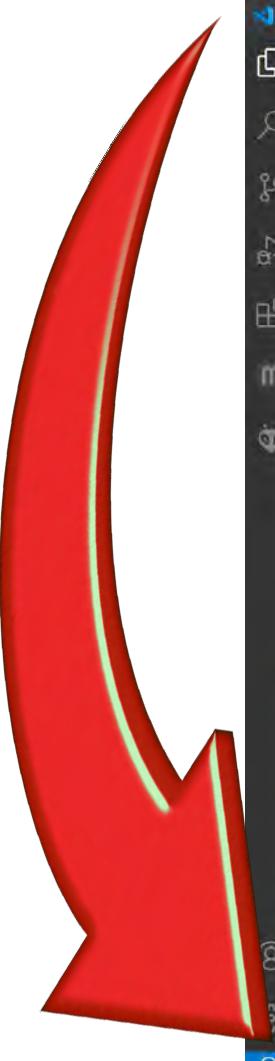
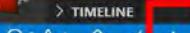
TERMINAL
Marlin > Configuration.h > ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
[SUCCESS] Took 130.61 seconds =====
Environment Status Duration
----- -----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu DUE
DUE_USB IGNORED
DUE_DFU IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

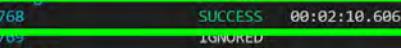
```

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

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

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

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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug.list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h _Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

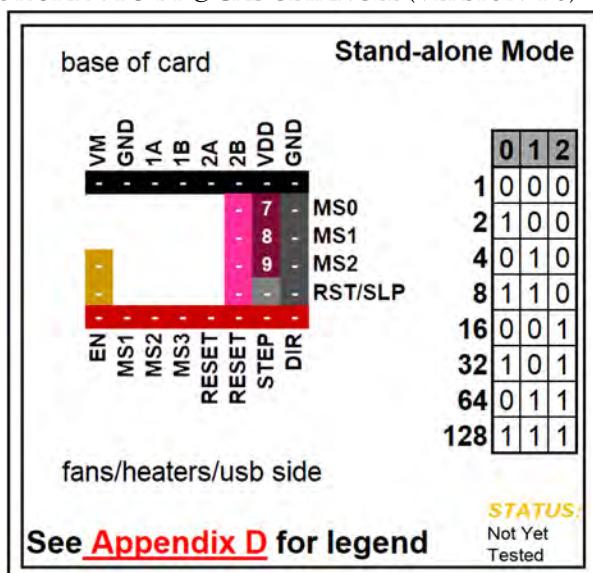
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o Archiving .pio\build\LPC1768\libFrameworkArduino.a Linking .pio\build\LPC1768\firmware.elf Checking size .pio\build\LPC1768\firmware.elf Advanced Memory Usage is available via "PlatformIO Home > Project Inspect" RAM: [=====] 42.5% (used 13988 bytes from 32736 bytes) Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes) Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUINO_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

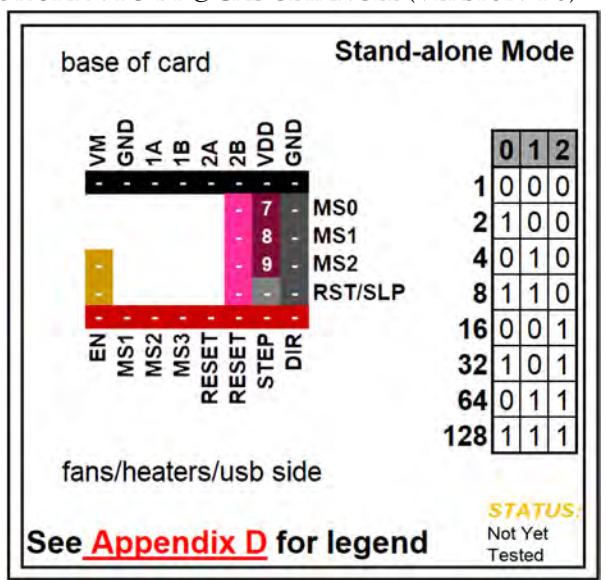
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3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

FYSETC LV8729

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

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

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

**Driver Chip Chart:**

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

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

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

FYSETC LV8729**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Low → **No Jumper**



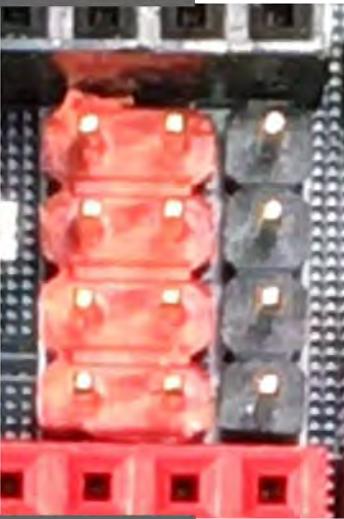
Rows:

MS0

MS1

MS2

RST/SLP



High → **Jumper Set**

Rows:

MS0

MS1

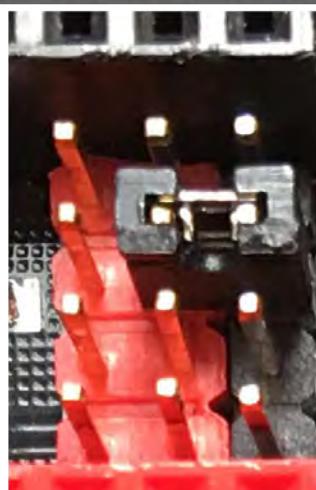
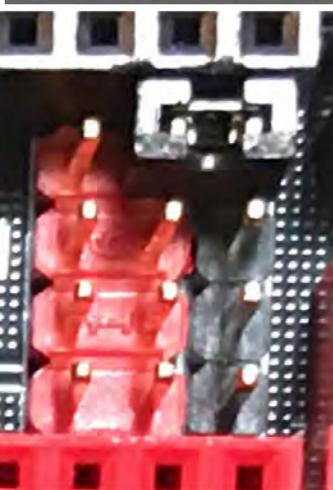
MS2

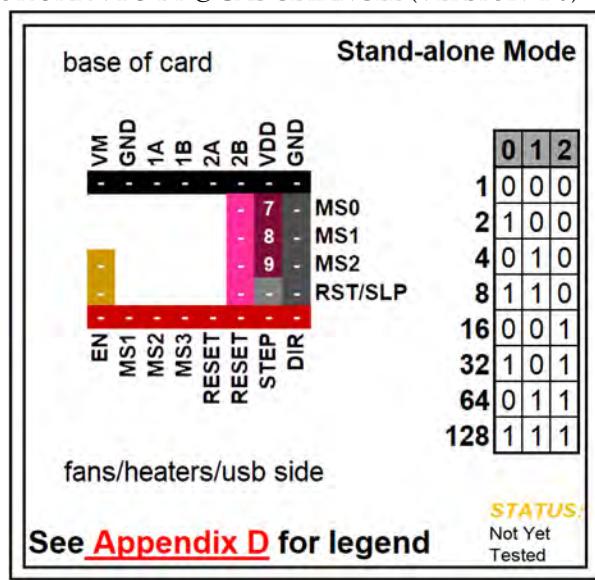
RST/SLP

MS0 SET HIGH

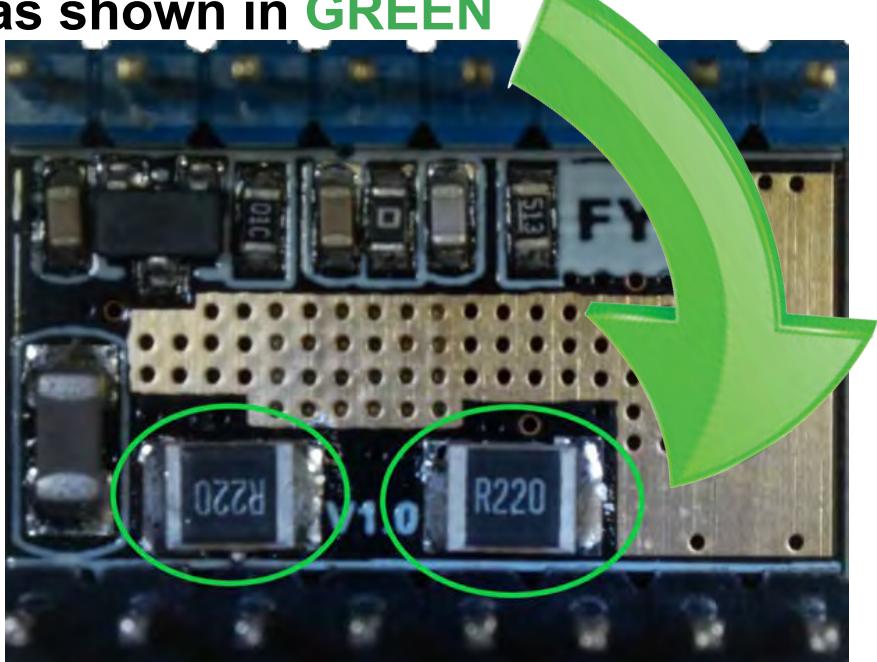
MS1 SET HIGH

MS2 SET HIGH





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

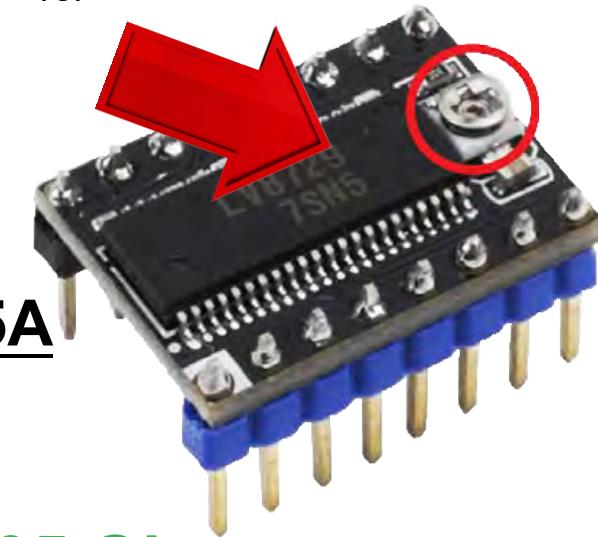


FYSETC LV8729

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

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

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



$R_s = R050$ is 0.05 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

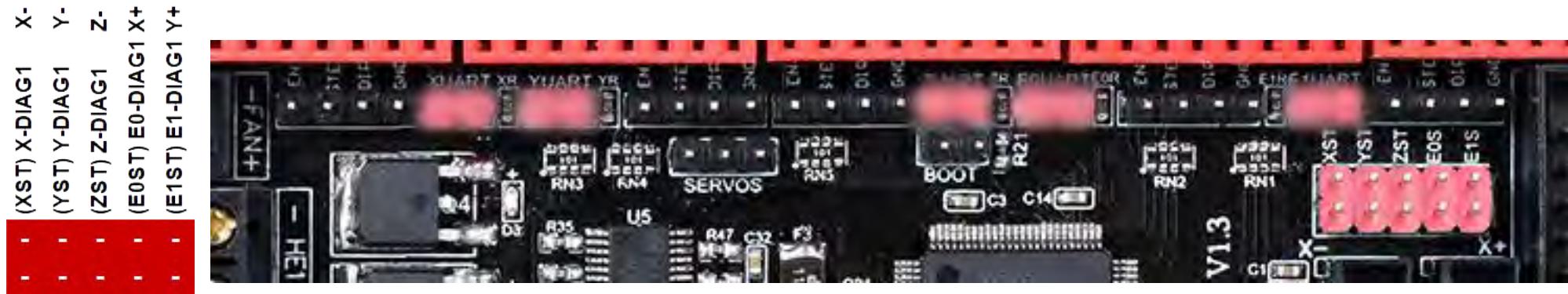
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

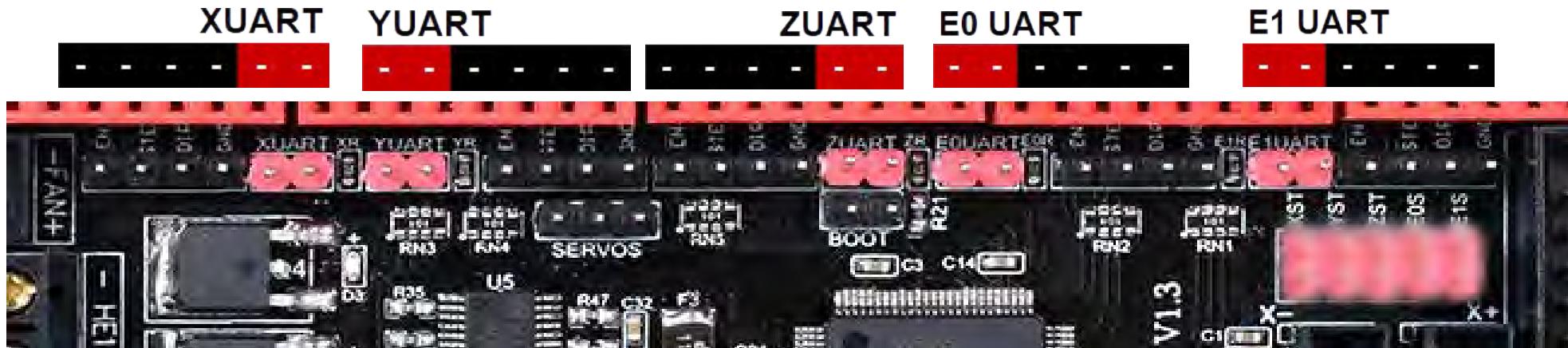
Stand-alone Mode

FYSETC LV8729

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



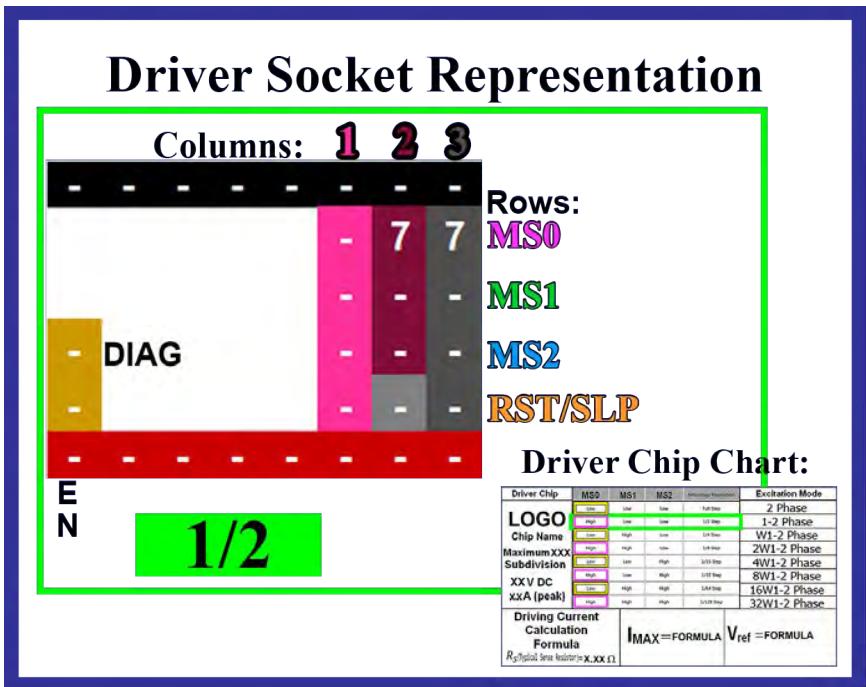
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



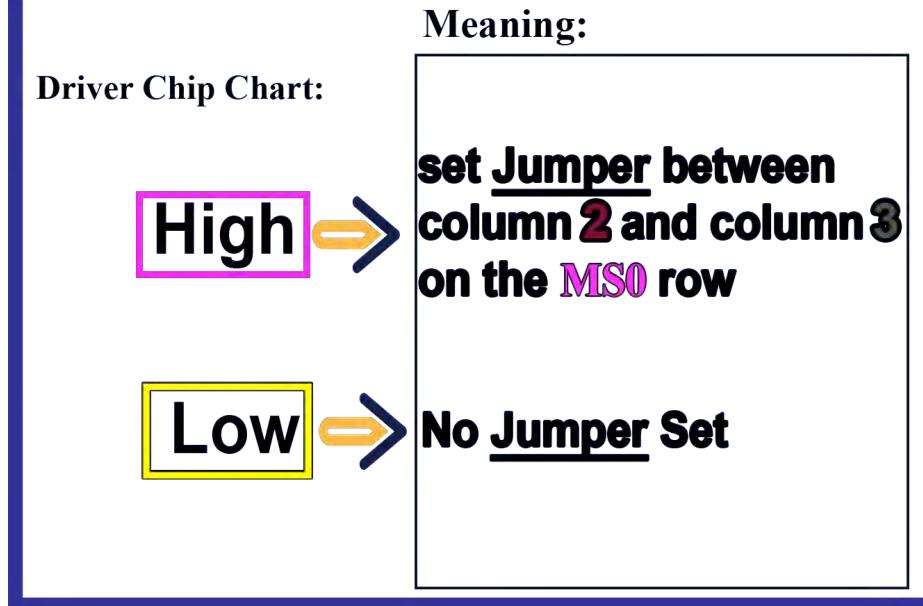
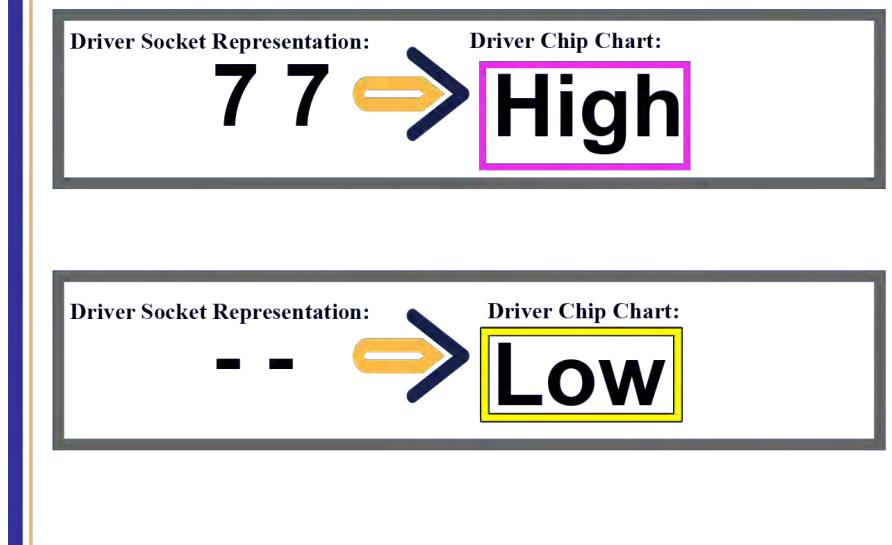
Stand-alone Mode

FYSETC LV8729

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

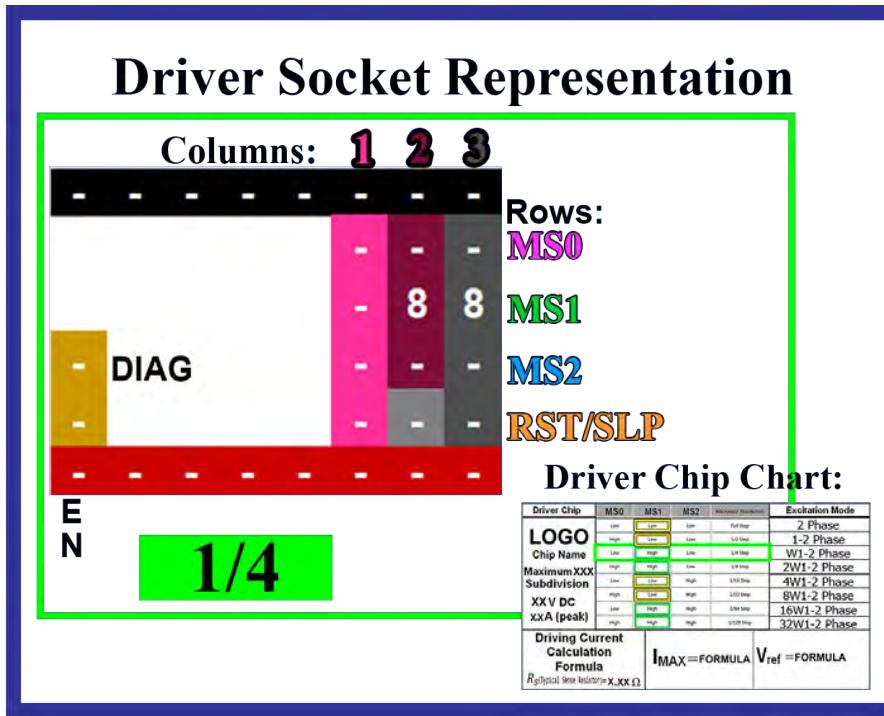


MS0 for Binary State Drivers:

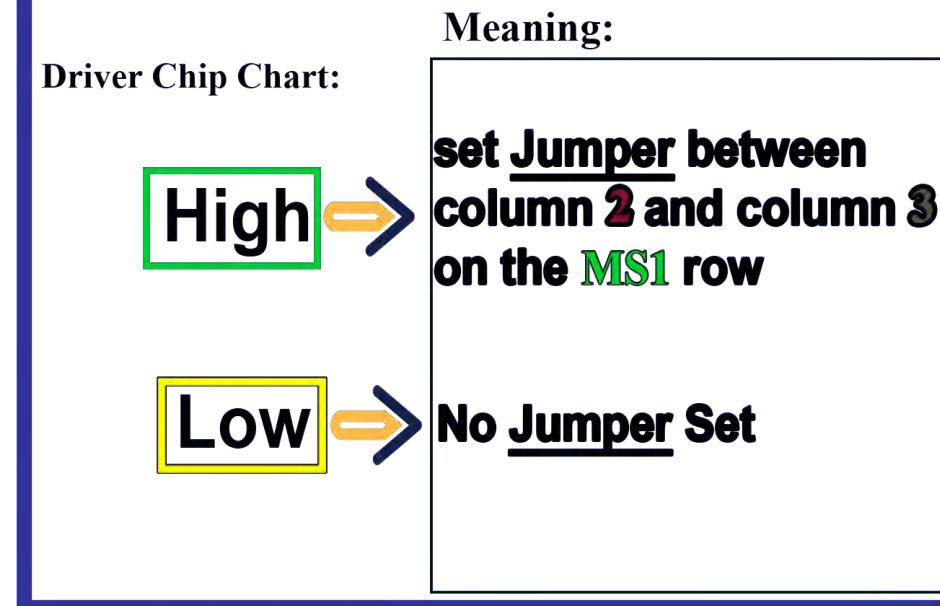
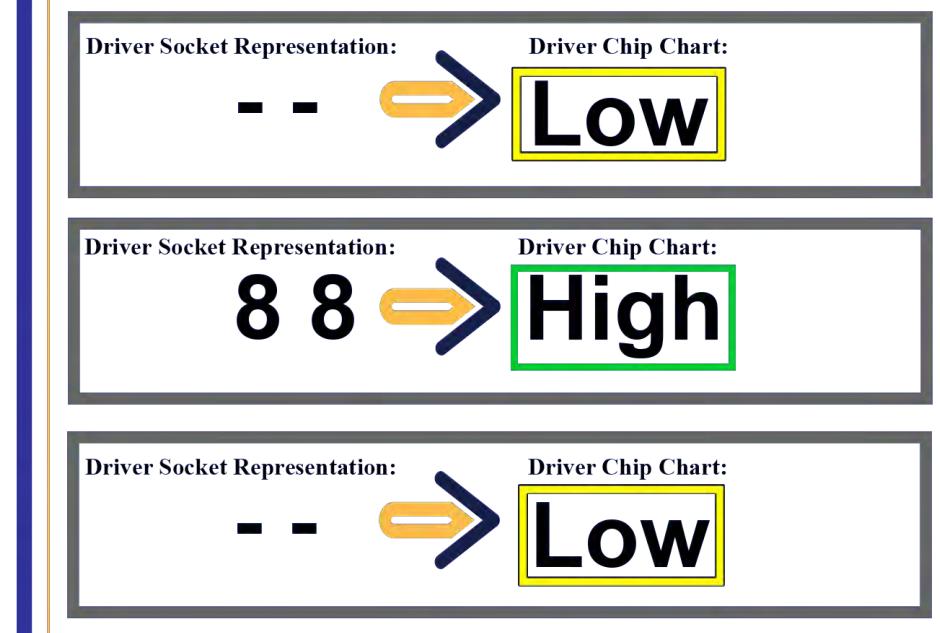


Stand-alone Mode

FYSETC LV8729



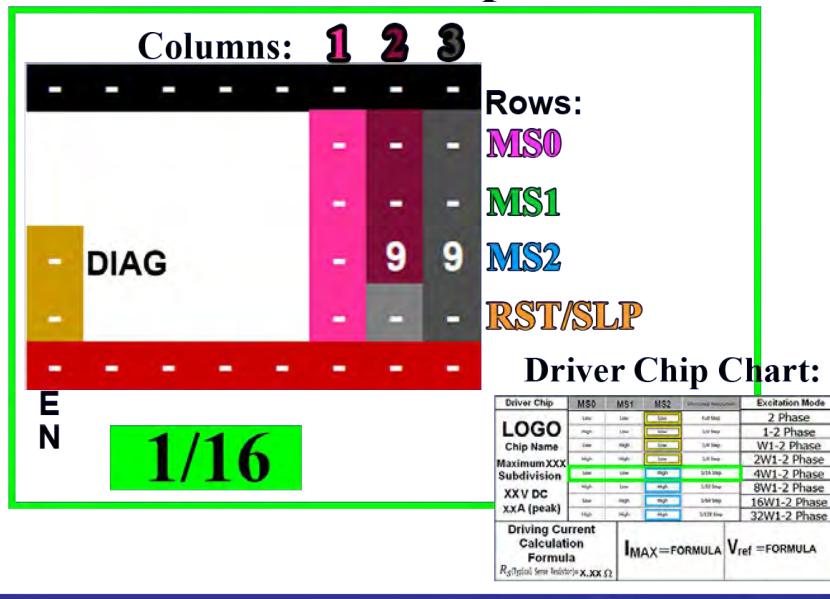
MS1 for Binary State Drivers:



Stand-alone Mode

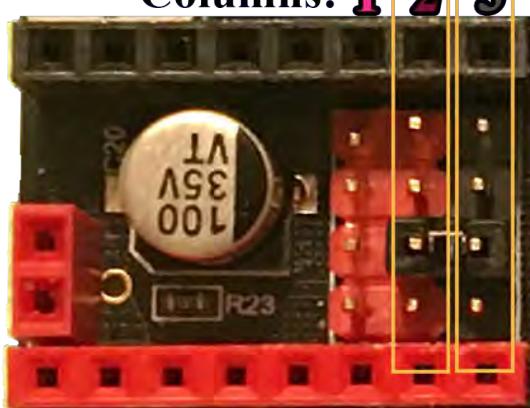
FYSETC LV8729

Driver Socket Representation



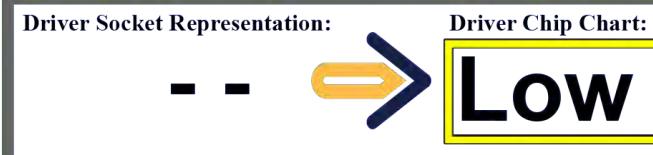
High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

High →

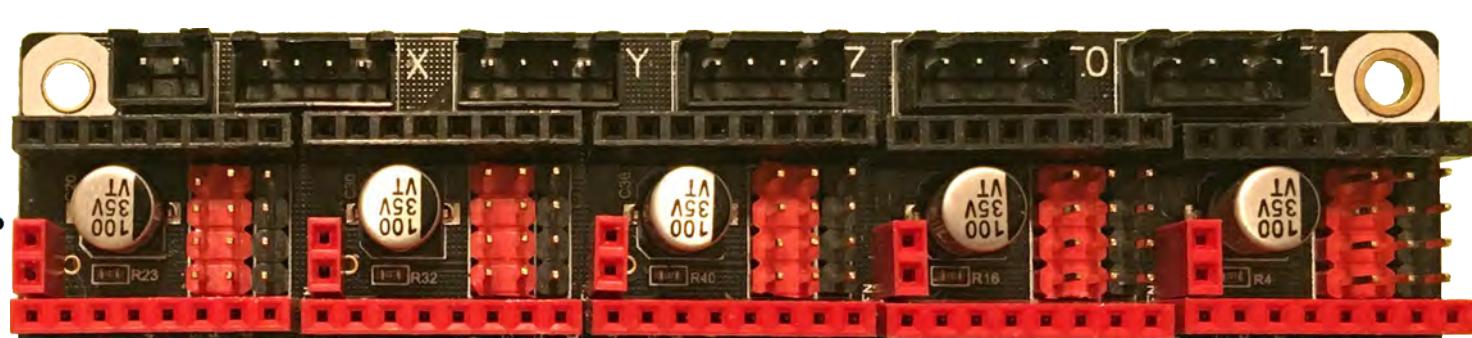
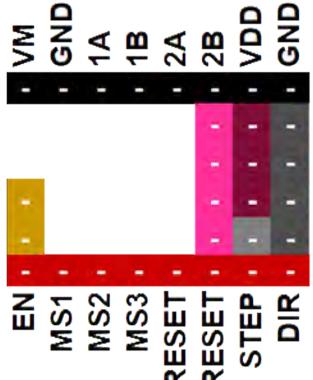
set Jumper between column 2 and column 3 on the MS2 row

Low →

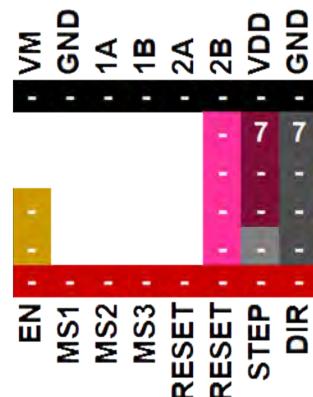
No Jumper Set

Stand-alone Mode

FYSETC LV8729



See [Appendix D](#) for legend

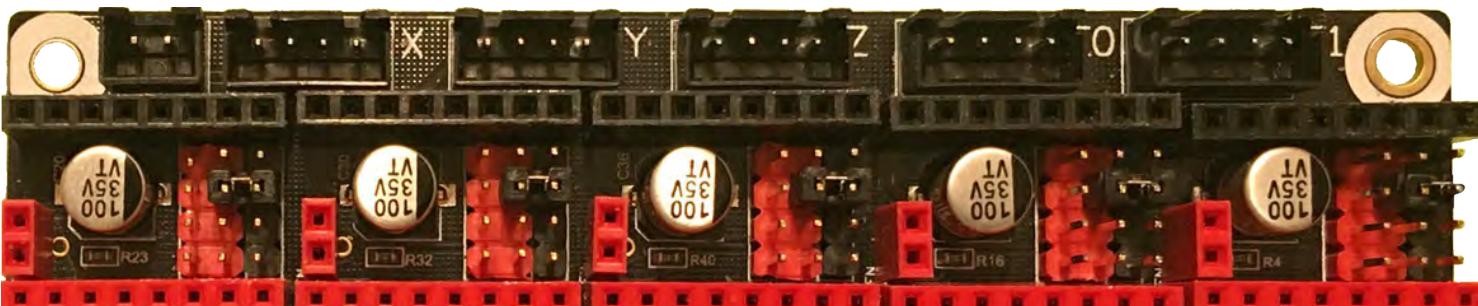
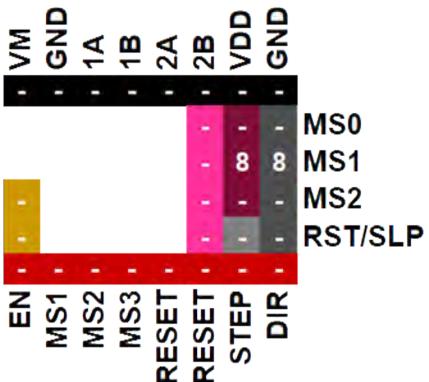


See [Appendix D](#) for legend

Stand-alone Mode

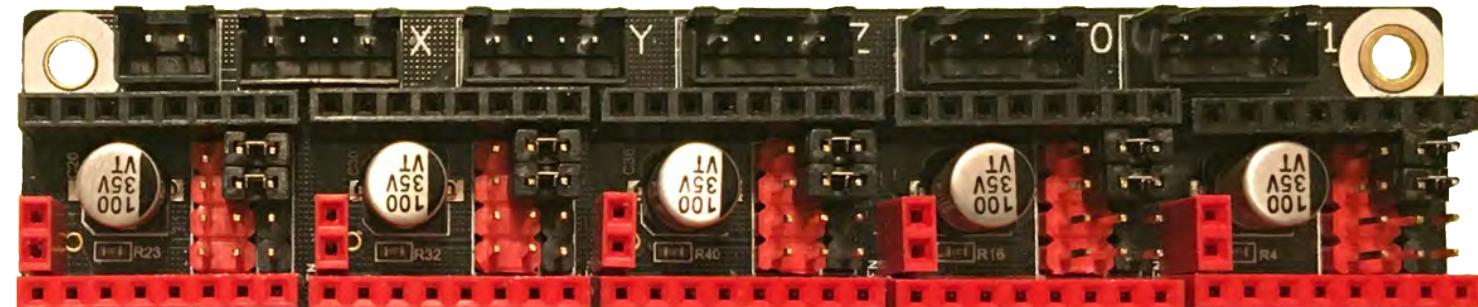
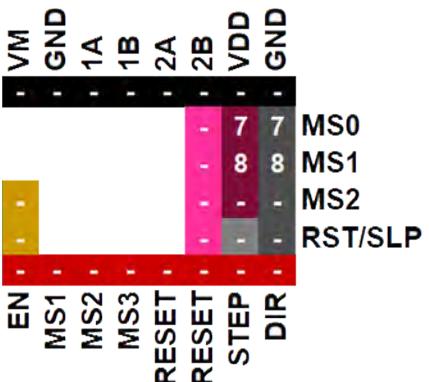
FYSETC LV8729

1 / 4



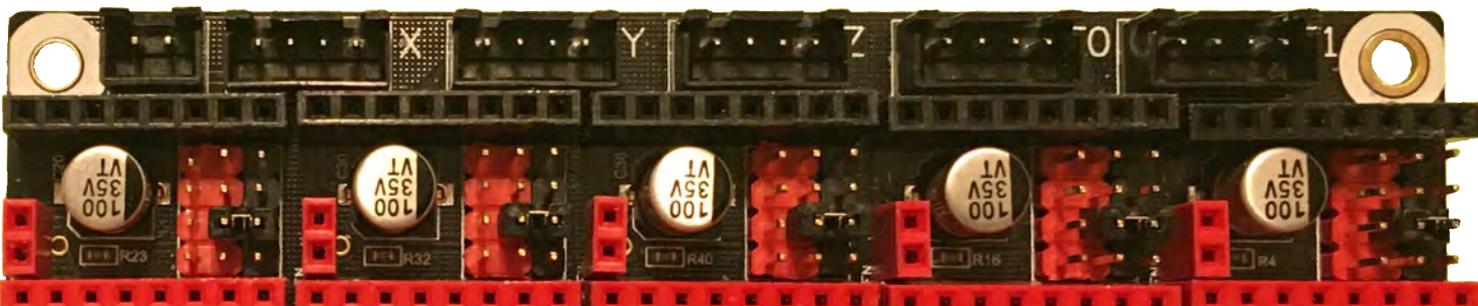
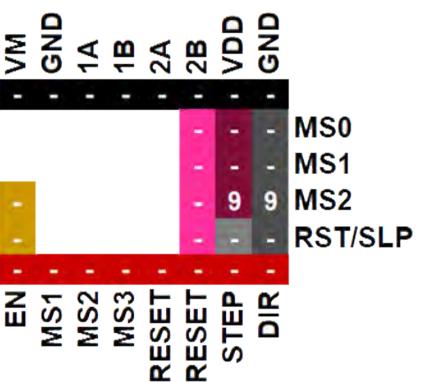
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16

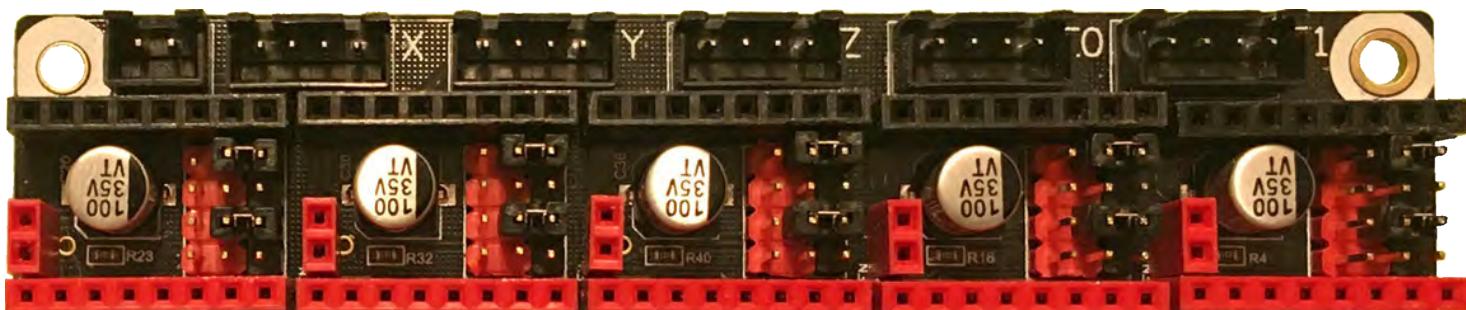
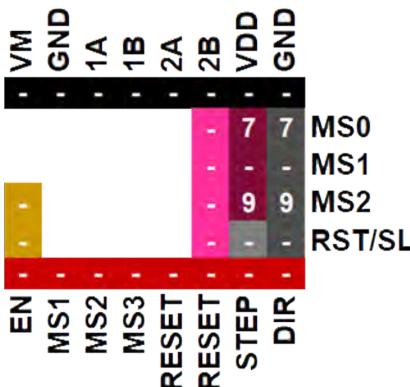


See [Appendix D](#) for legend

Stand-alone Mode

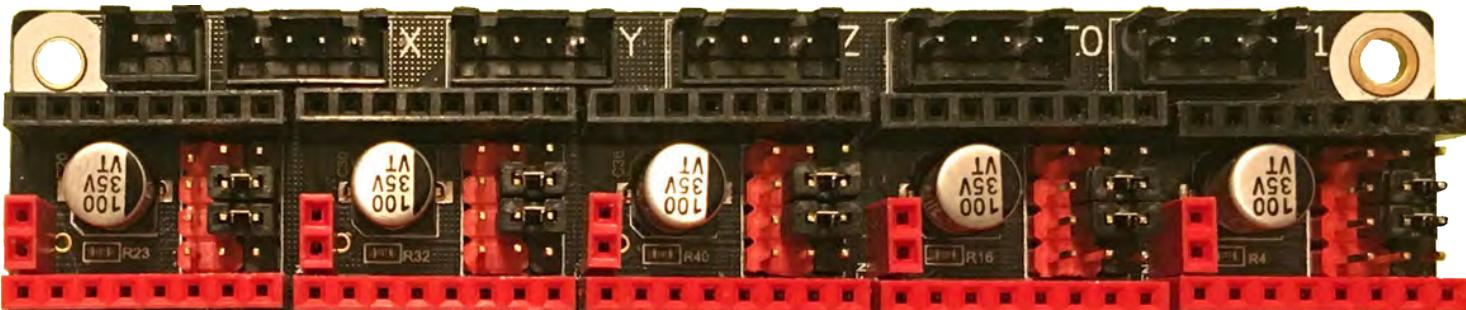
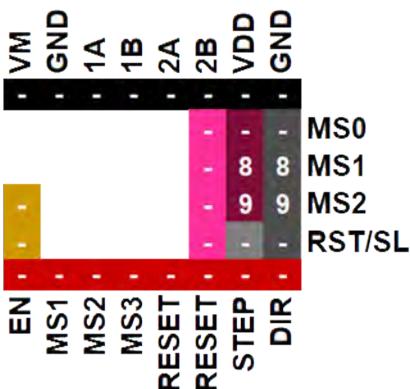
FYSETC LV8729

1 / 32



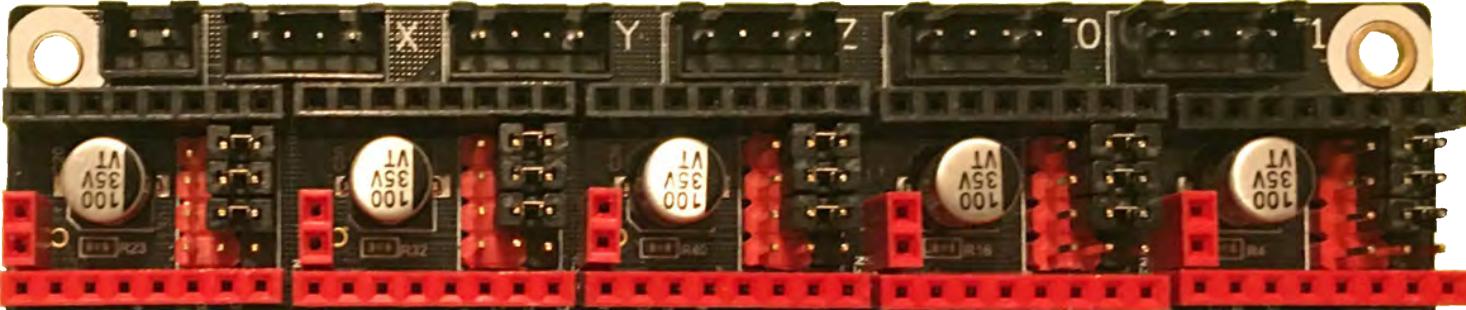
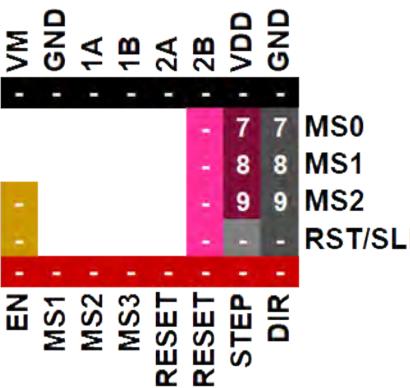
See [Appendix D](#) for legend

1 / 64



See [Appendix D](#) for legend

1 / 128



See [Appendix D](#) for legend

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

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

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

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

```

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

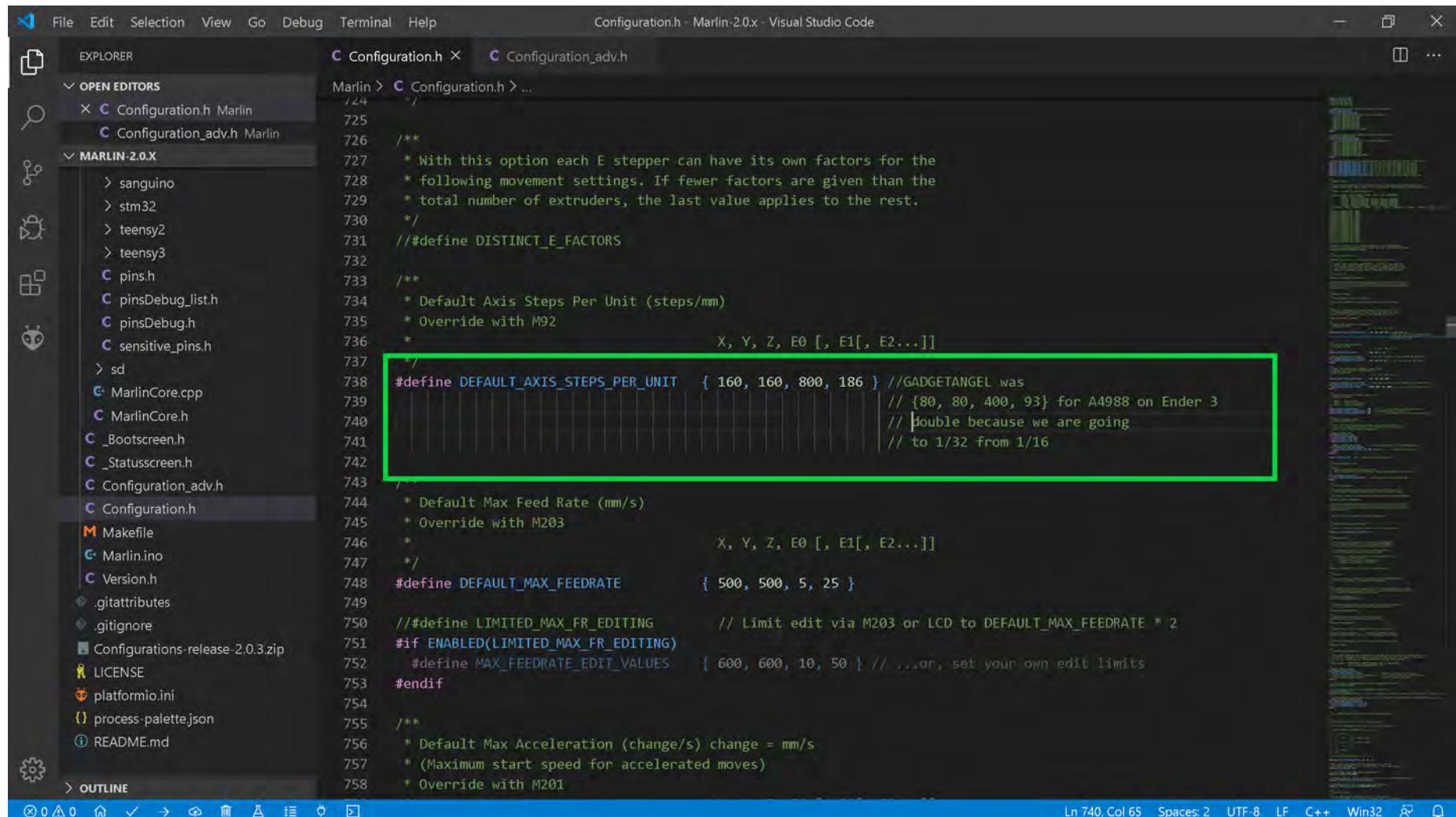
```

The lines from 678 to 693 are highlighted with a green box, indicating they are the specific changes needed for the FYSETC LV8729 drivers.

- Go to the next page.

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

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



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

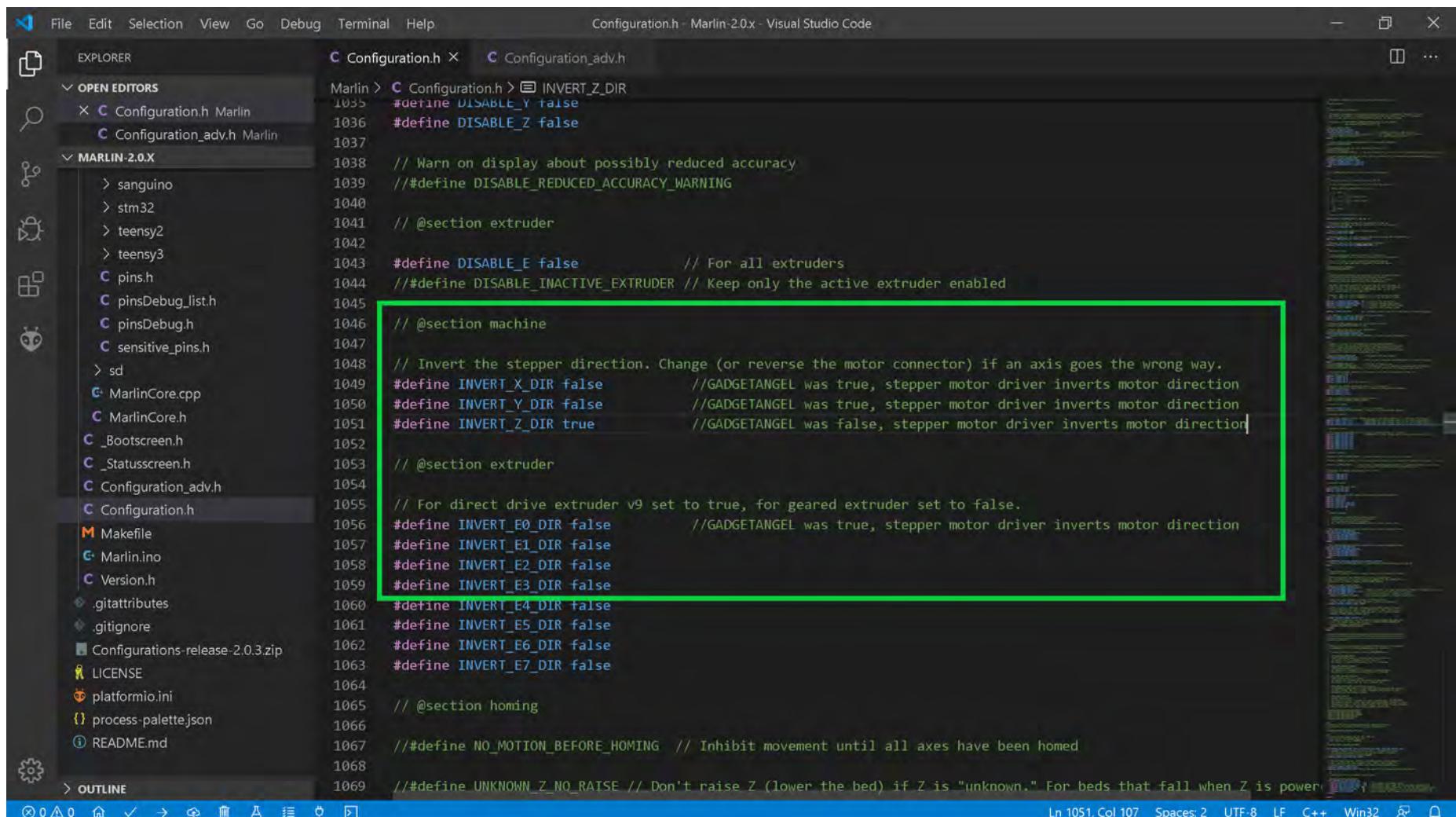
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays C++ code for Marlin 2.0.x. A green rectangular box highlights a section of the code where the value of the macro `INVERT_Z_DIR` is being modified. The code snippet is as follows:

```

Marlin > C Configuration.h > #define INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

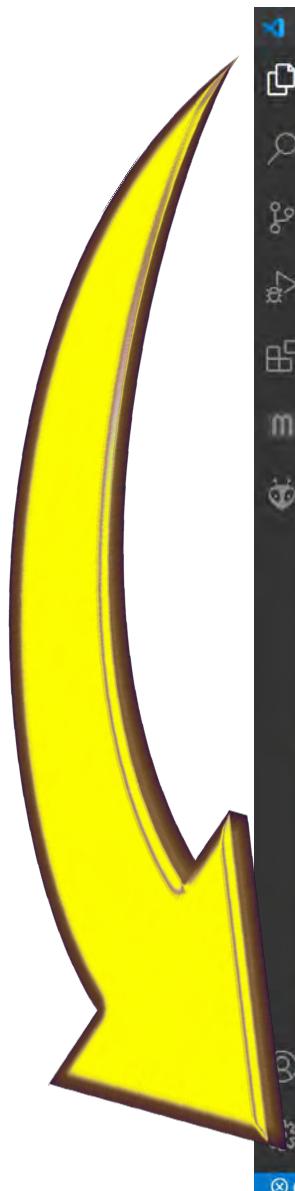
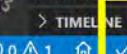
```

The green box covers lines 1049 through 1051, where the value of `INVERT_Z_DIR` is being changed from `false` to `true`.

- Go to the next page.

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

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

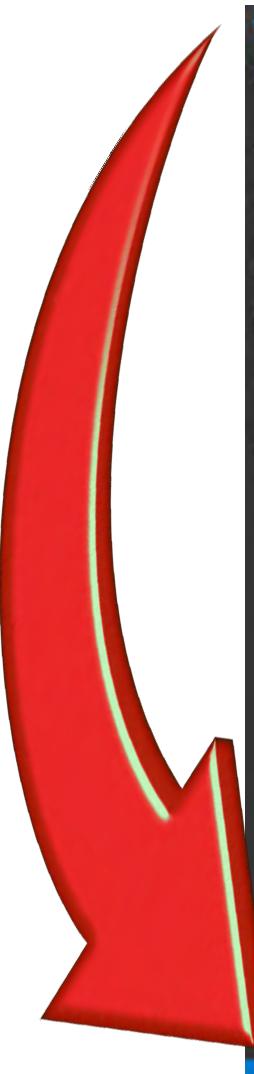


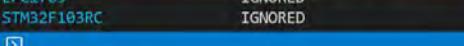
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	DUE	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

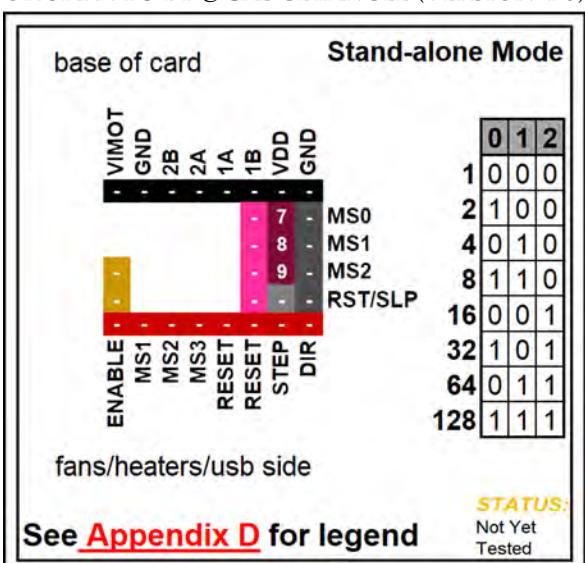
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1709	IGNORED	
STM32F103RC	IGNORED	

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



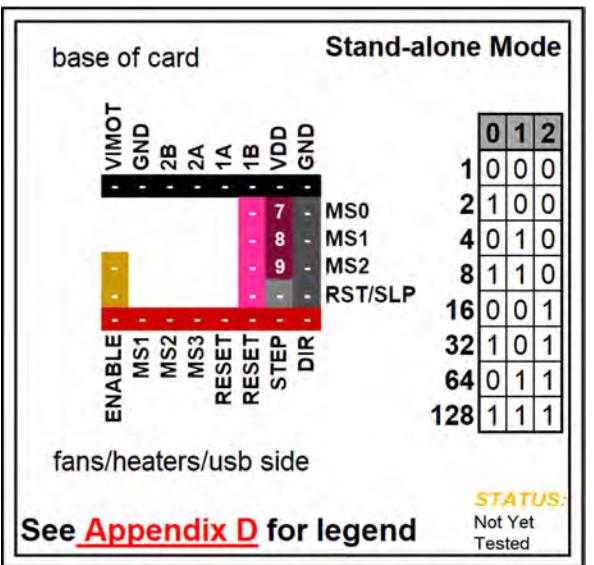
See [Appendix D](#) for legend

LERDGE LV8729

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

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

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



Driver Chip Chart:

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

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

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

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

LERDGE LV8729

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

Low ➡ **No Jumper**



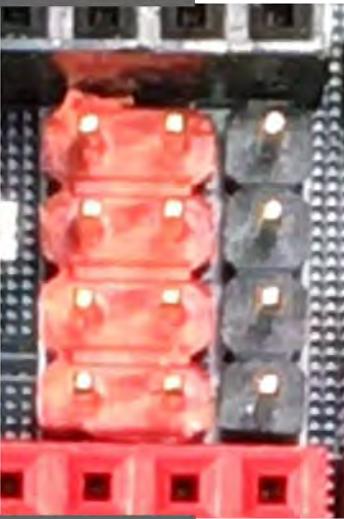
Rows:

MS0

MS1

MS2

RST/SLP



High ➡ **Jumper Set**

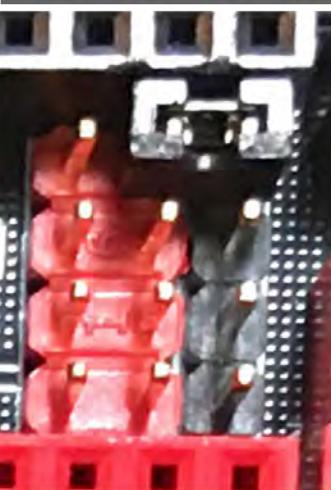
Rows:

MS0

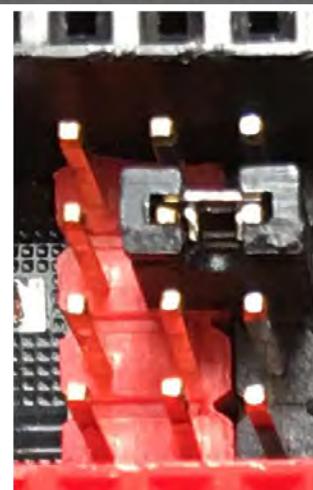
MS1

MS2

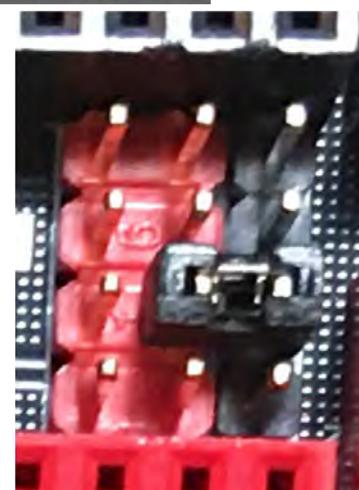
RST/SLP



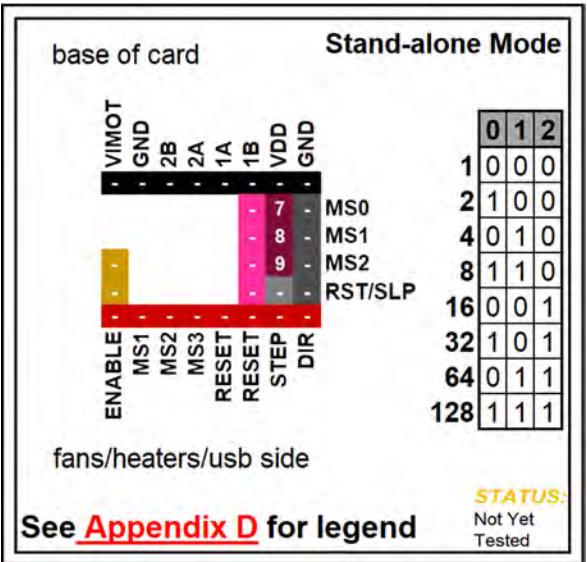
MS0 SET HIGH



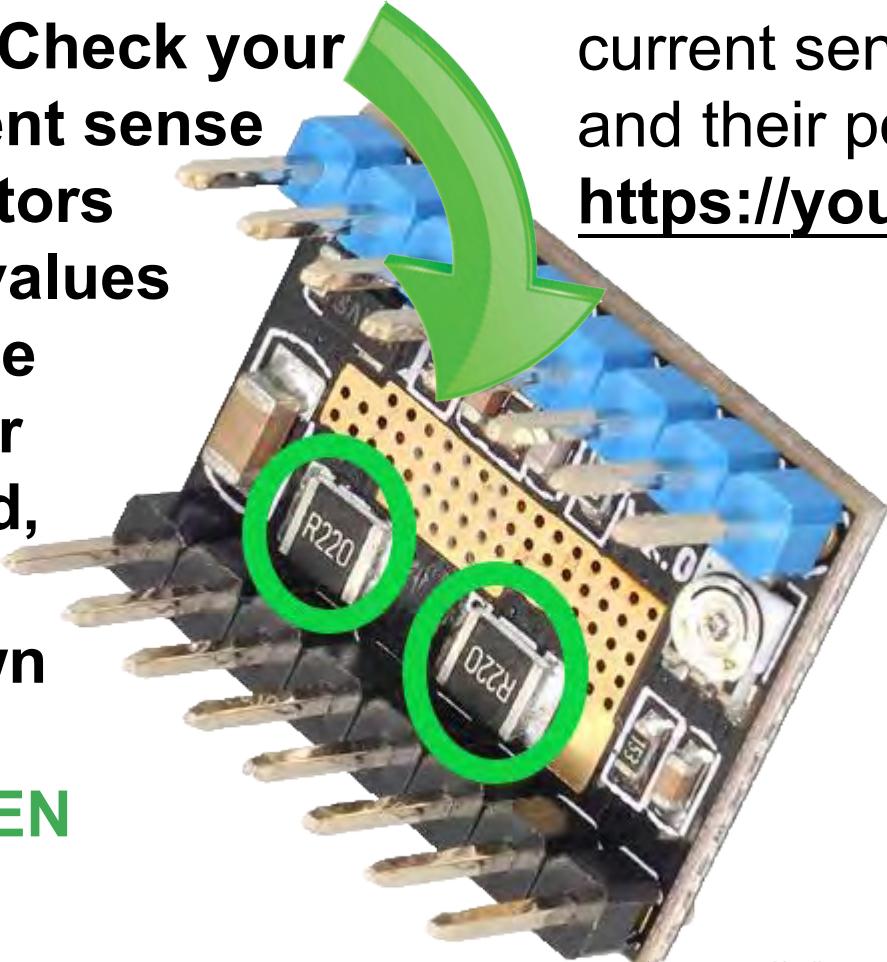
MS1 SET HIGH



MS2 SET HIGH



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

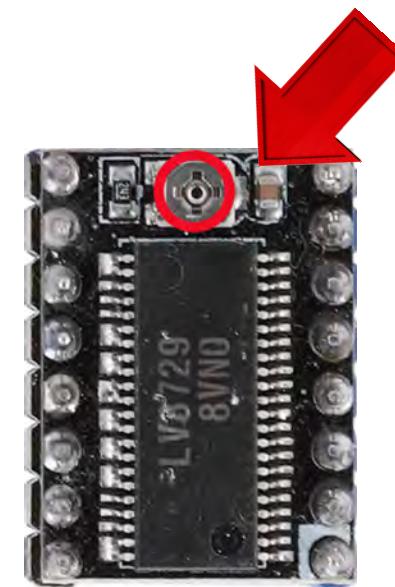


LERDGE LV8729

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

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

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



$R_s = R050$ is 0.05 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

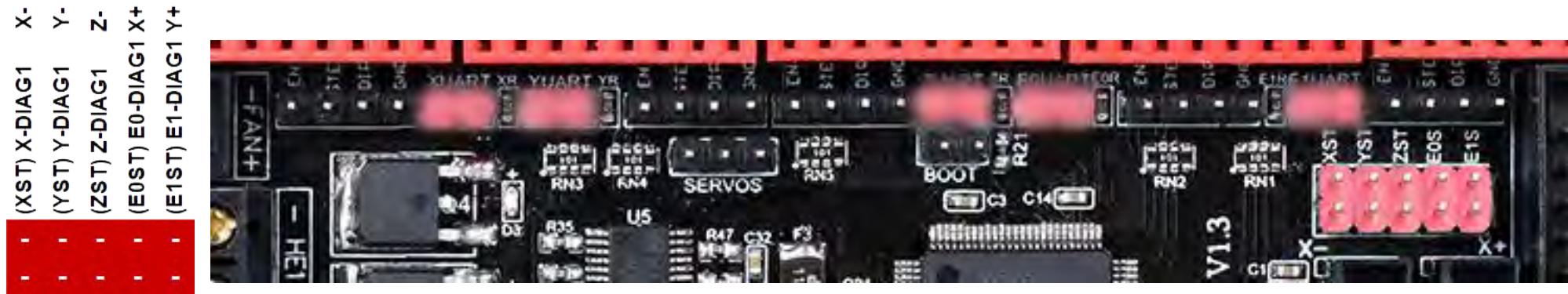
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

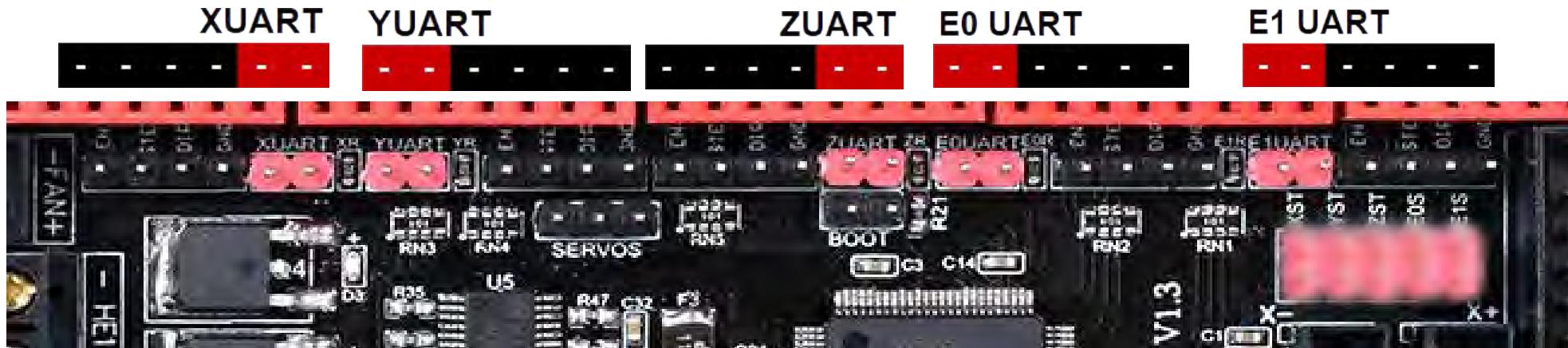
Stand-alone Mode

LERDGE LV8729

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



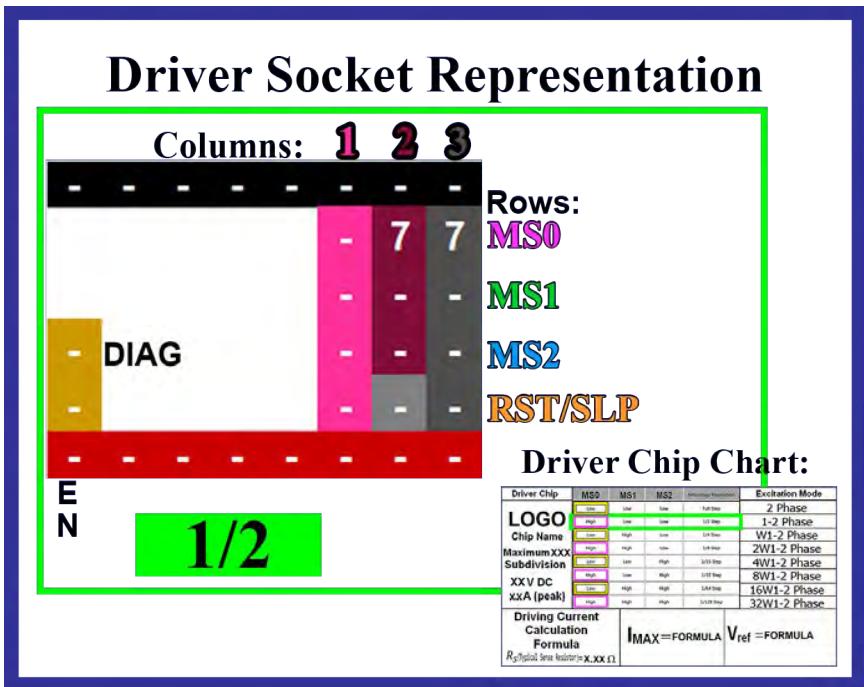
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



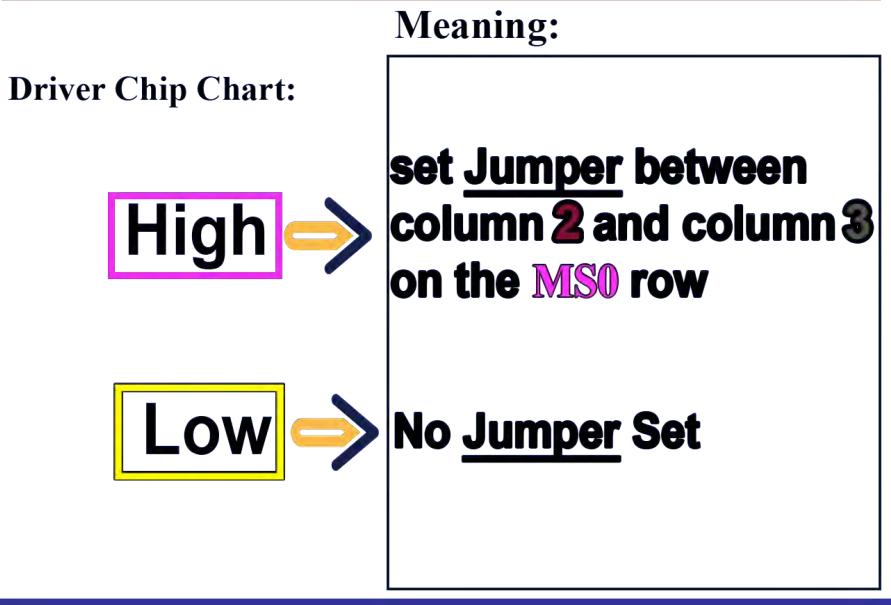
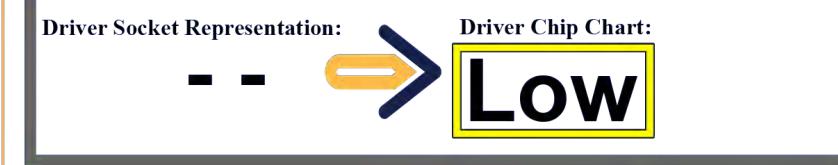
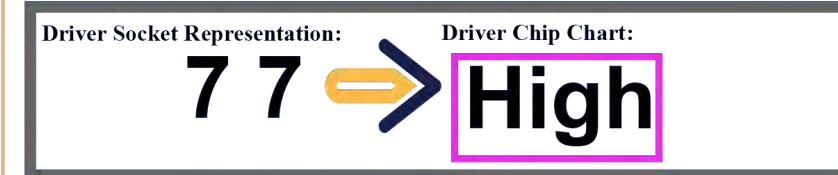
Stand-alone Mode

LERDGE LV8729

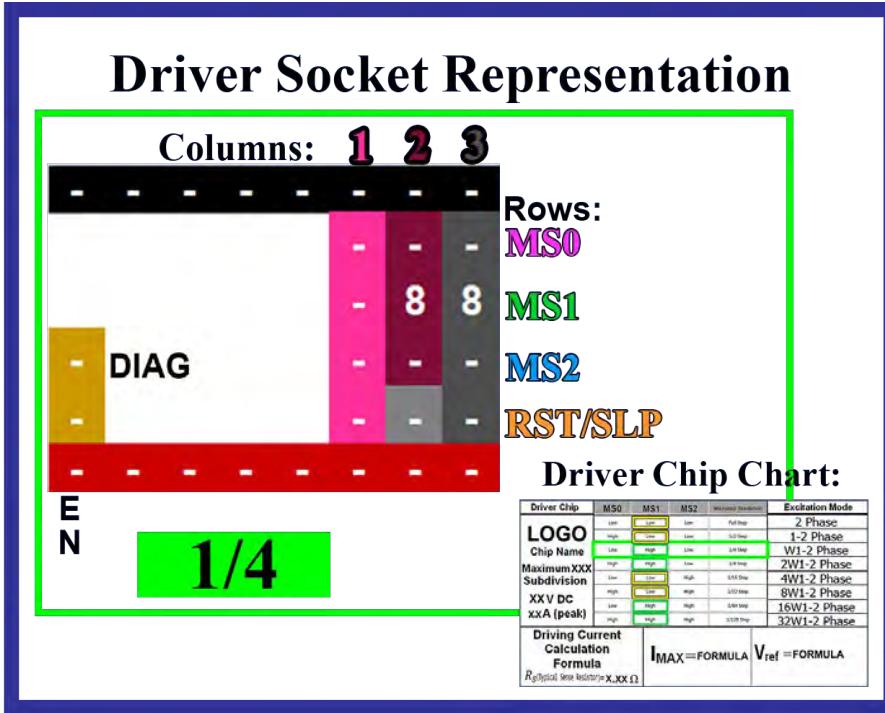
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers



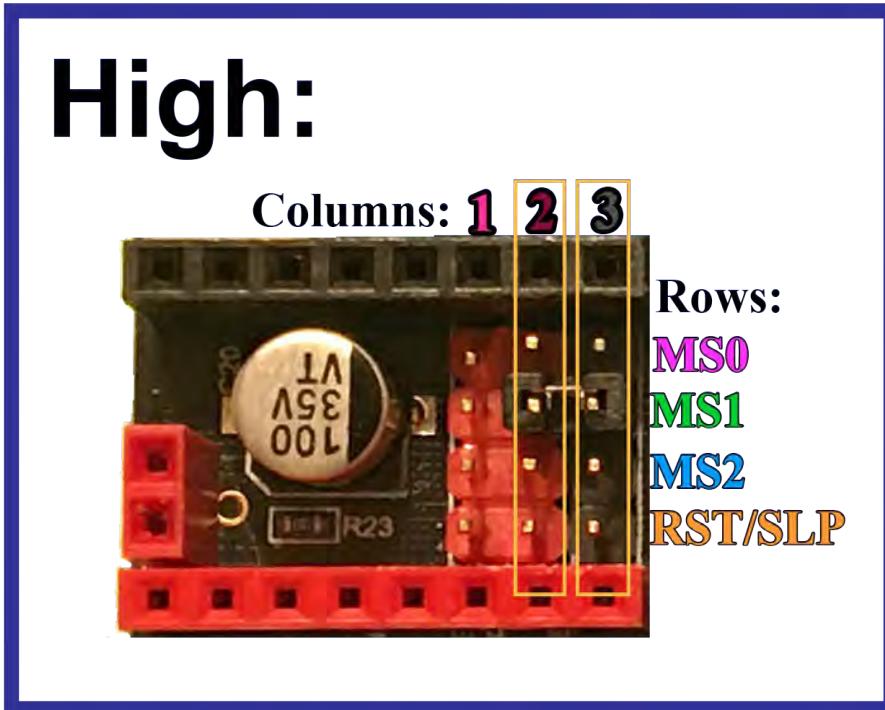
MS0 for Binary State Drivers:



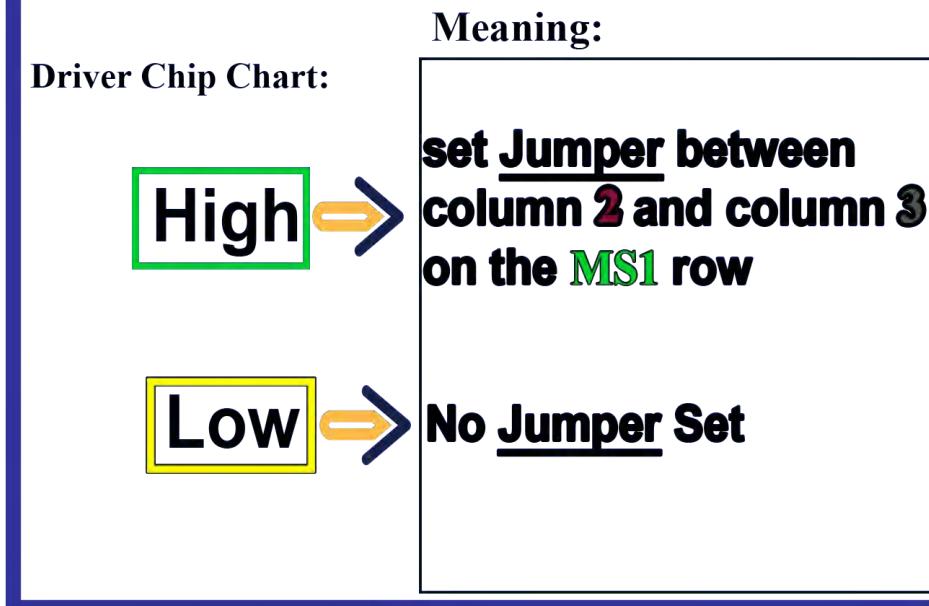
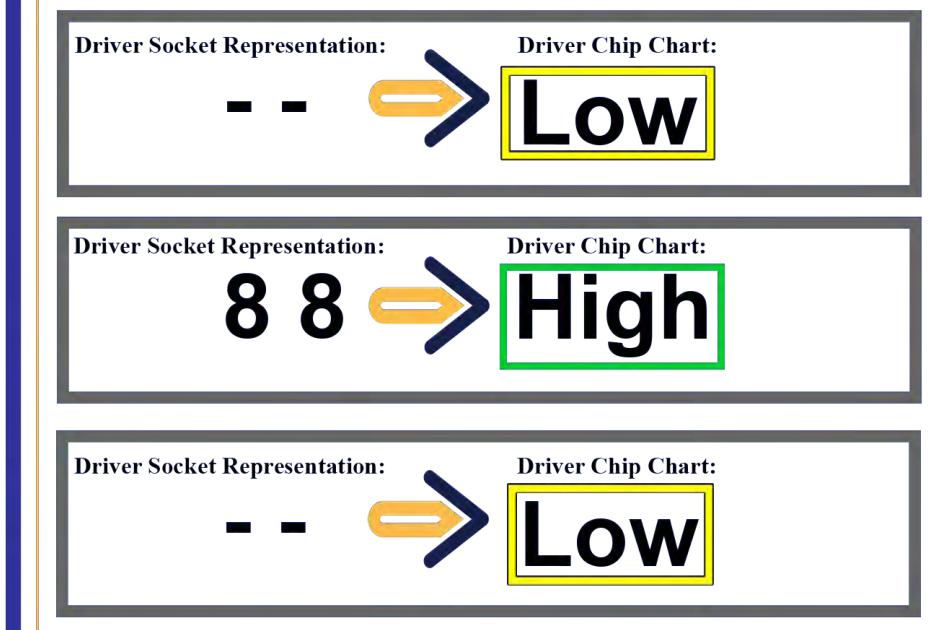
Stand-alone Mode



LERDGE LV8729



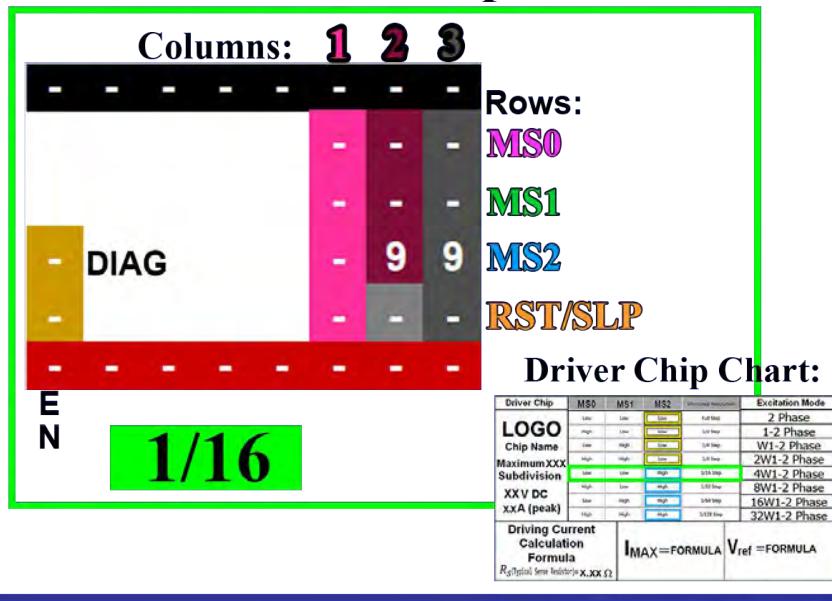
MS1 for Binary State Drivers:



Stand-alone Mode

LERDGE LV8729

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

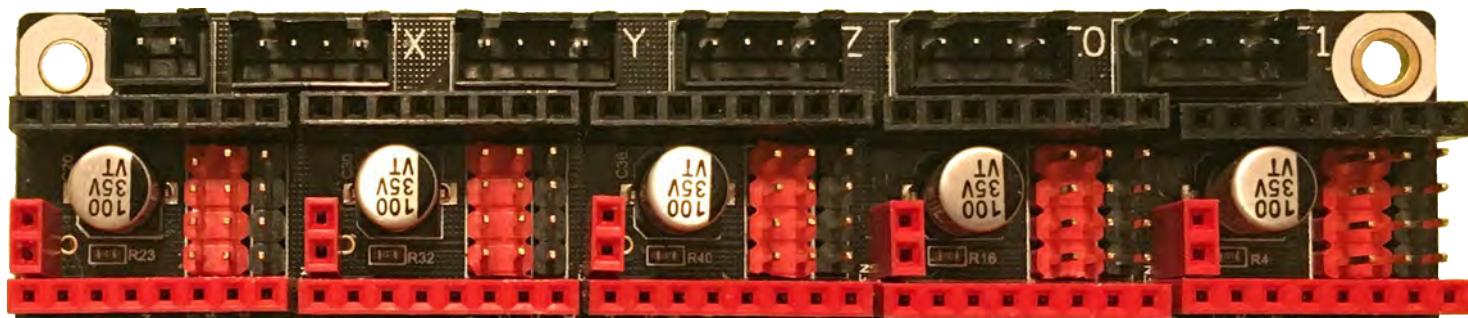
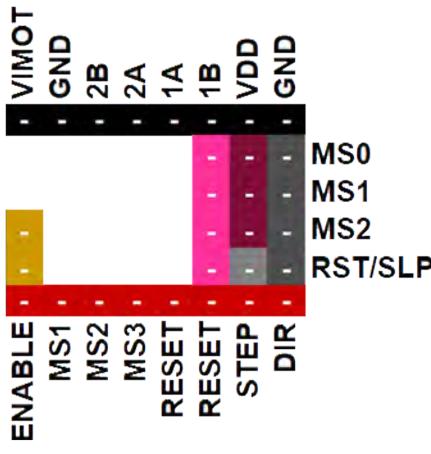
High → **set Jumper between column 2 and column 3 on the MS2 row**

Low → **No Jumper Set**

Stand-alone Mode

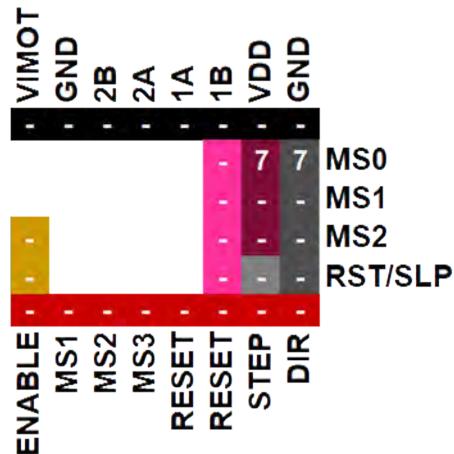
LERDGE LV8729

STEP



See [Appendix D](#) for legend

1 / 2

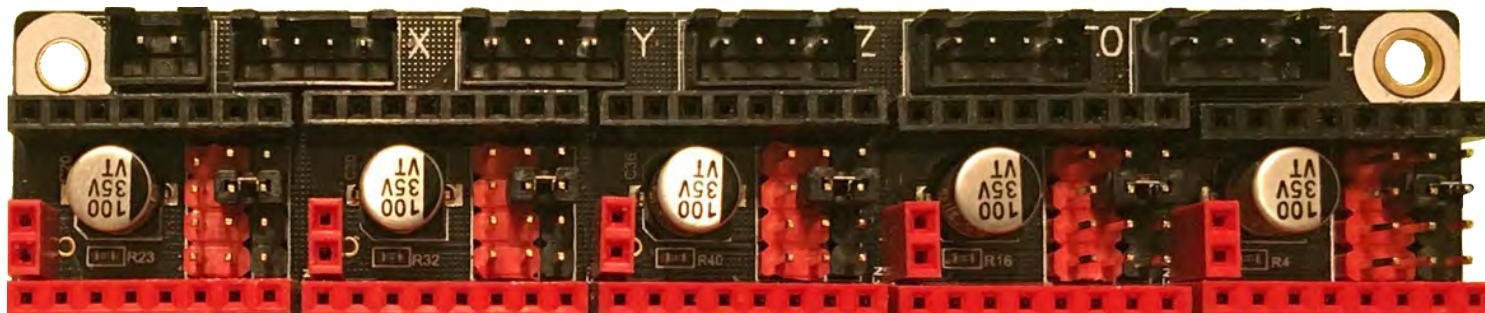
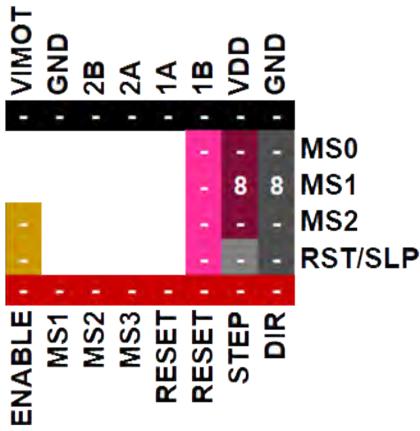


See [Appendix D](#) for legend

Stand-alone Mode

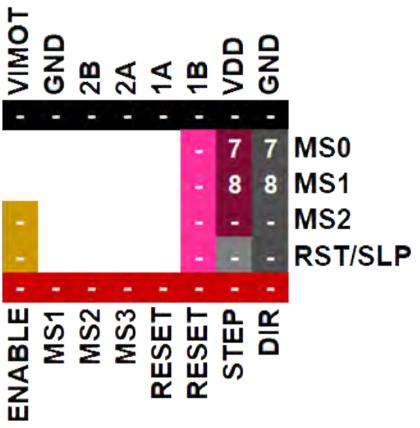
LERDGE LV8729

1 / 4



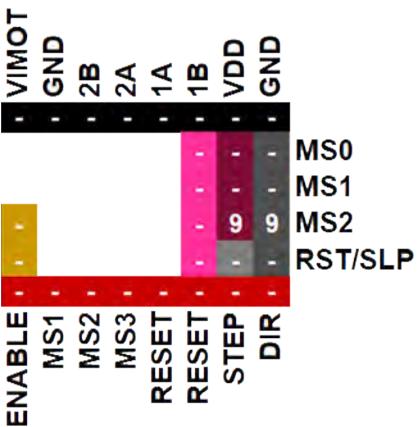
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



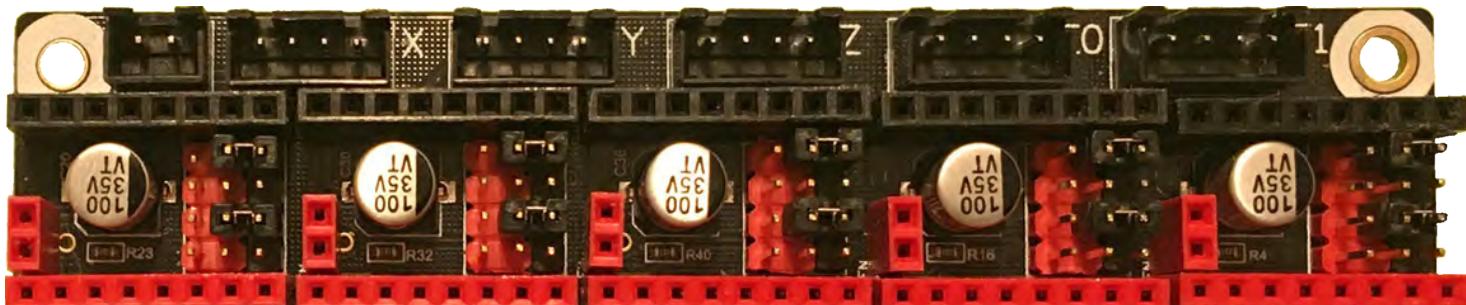
See [Appendix D](#) for legend

Stand-alone Mode

LERDGE LV8729

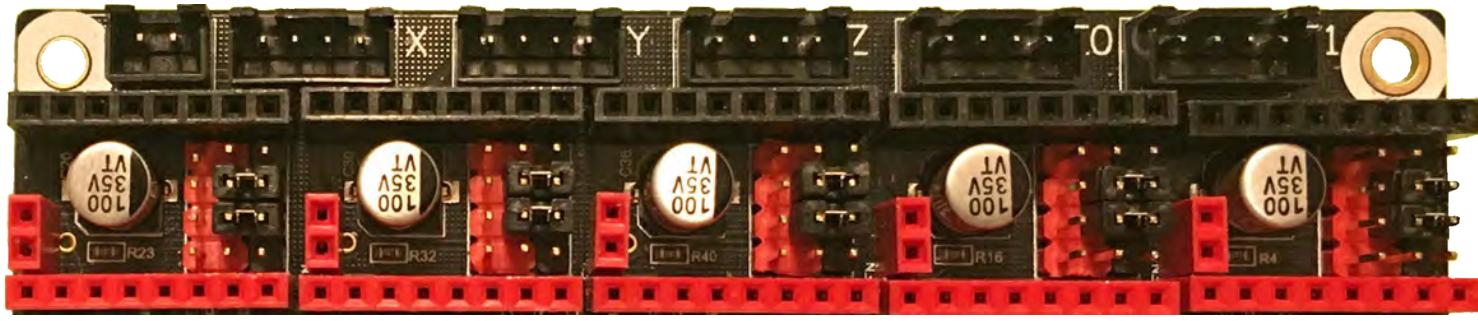
1 / 32

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



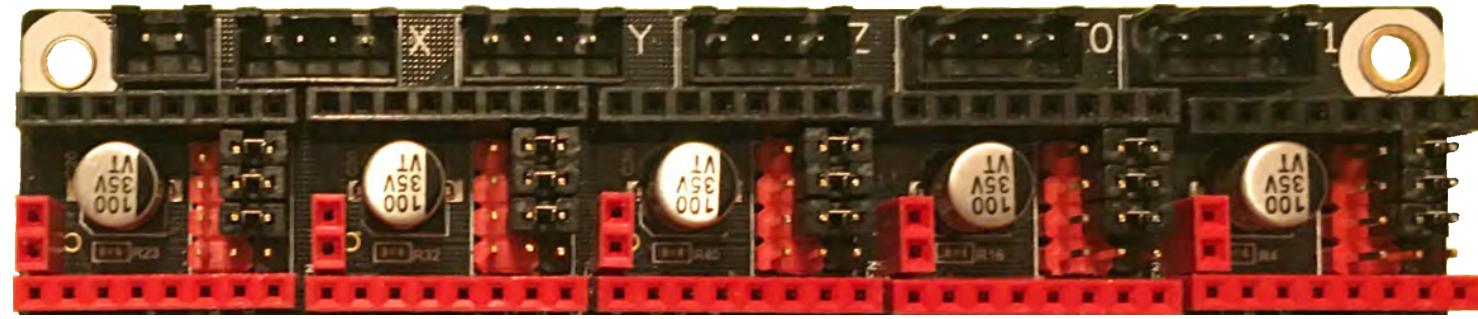
1 / 64

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



1 / 128

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



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

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

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

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

```

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

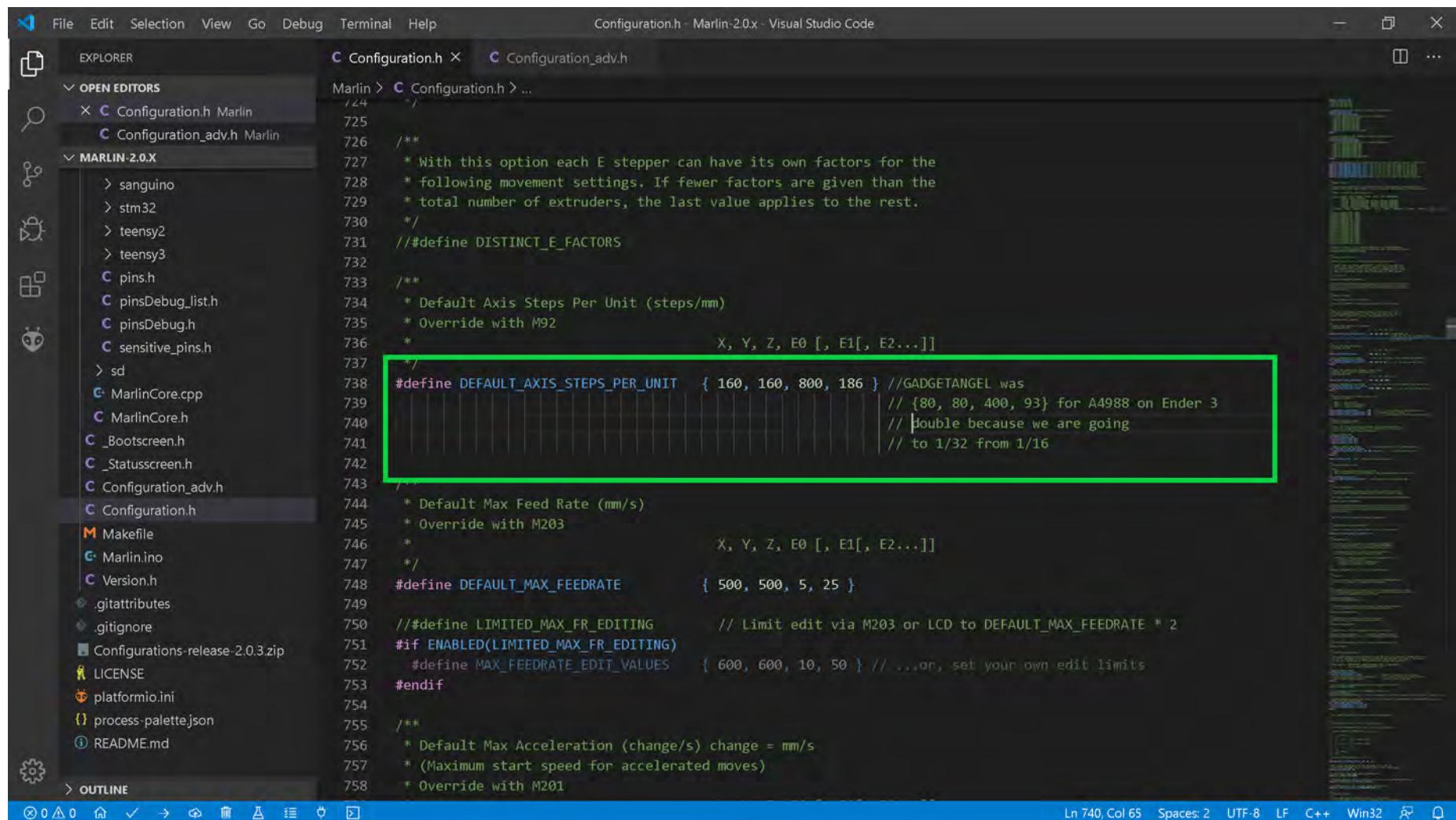
```

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

- Go to the next page.

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

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



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

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

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

- Go to the next page.

[The \(latest release of\) Marlin Setup for LERDGE LV8729 Drivers](#)

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

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

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):** Shows the project structure with files like Configuration.h, Configuration_adv.h, sanguino, stm32, teensy2, teensy3, pins.h, pinsDebug_list.h, pinsDebug.h, sensitive_pins.h, sd, MarlinCore.cpp, MarlinCore.h, _Bootscreen.h, _Statusscreen.h, Configuration_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, Configurations-release-2.0.3.zip, LICENSE, platformio.ini, process-palette.json, README.md.
- Right Sidebar:** Shows a vertical stack of code snippets or preview windows.
- Code Editor:** The Configuration.h file is open, showing C/C++ code. A green box highlights the following section:

```
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is power
```
- Bottom Status Bar:** Line 1051, Col 107, Spaces: 2, UTF-8, LF, C++, Win32.

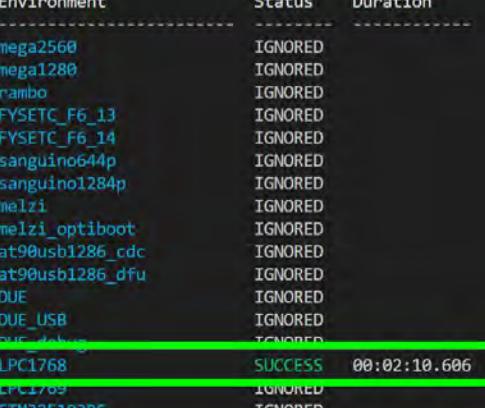
- Go to the next page.

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

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







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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

Marlin > Configuration.h ...

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131   #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

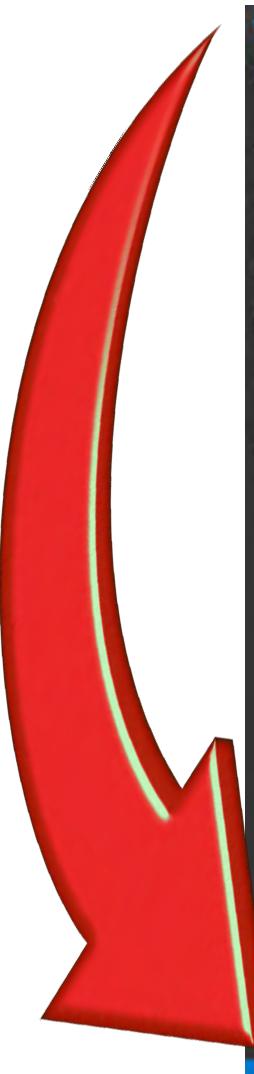
```

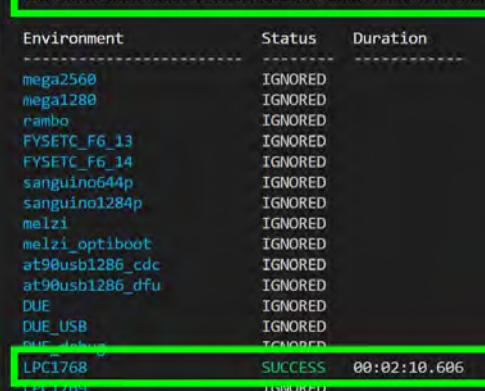
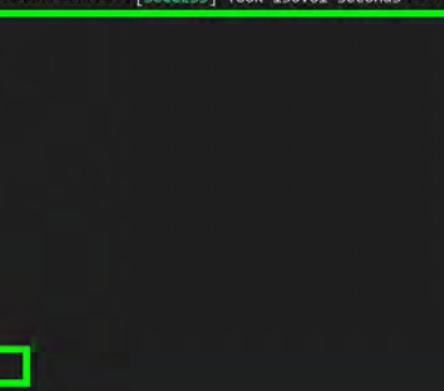
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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



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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

Configuration.h

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

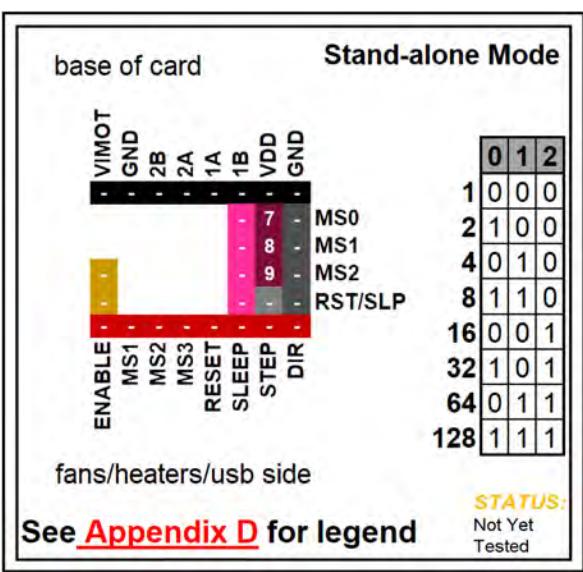
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1709	IGNORED	
STM32F103RC	IGNORED	

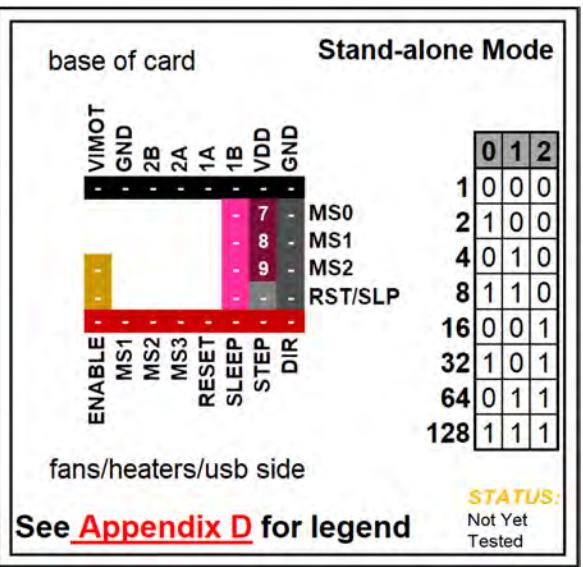
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3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

MKS LV8729

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

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

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



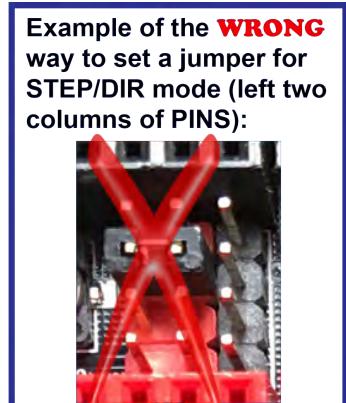
Driver Chip Chart:

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

Driving Current Calculation Formula

$I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

$R_s(\text{Typical Sense Resistor}) = X.XX \Omega$



MKS LV8729

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

Low ➡ **No Jumper**



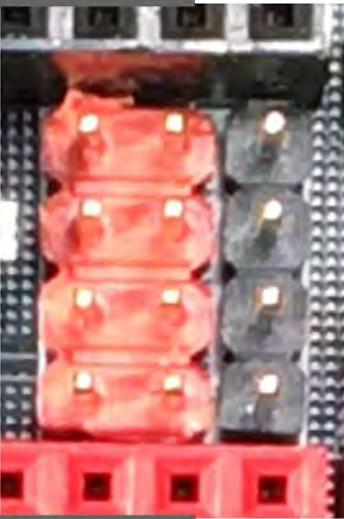
Rows:

MS0

MS1

MS2

RST/SLP



High ➡ **Jumper Set**

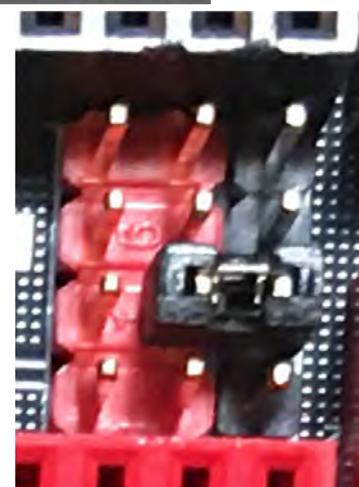
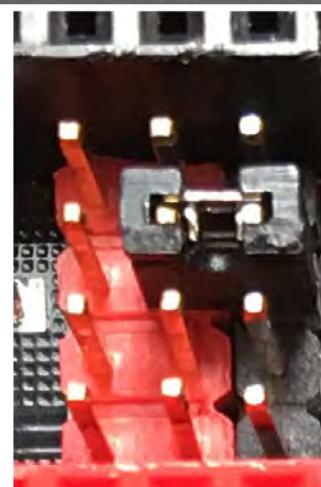
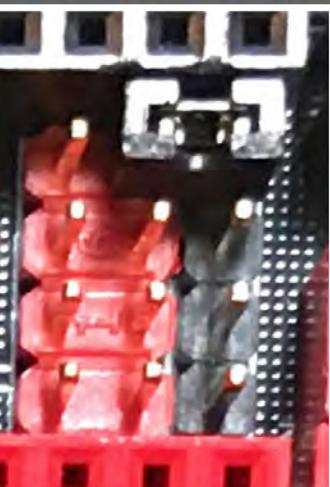
Rows:

MS0

MS1

MS2

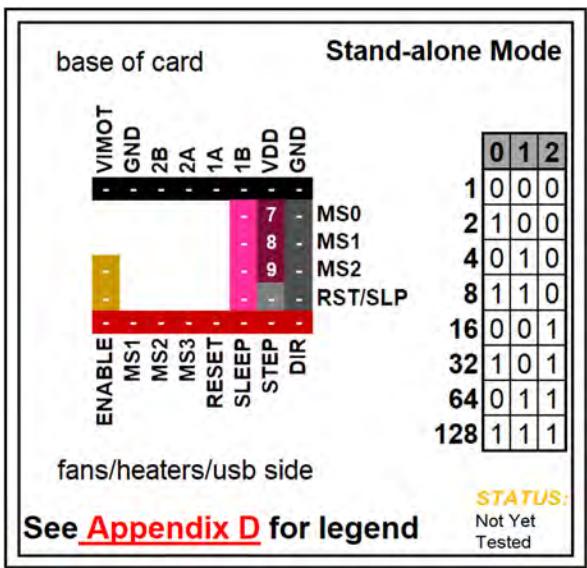
RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH



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



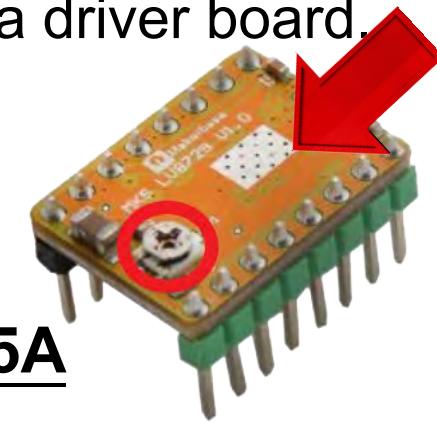
MKS LV8729

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

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

Note: See this video about current sense resistors (R_s) and their possible locations:

<https://youtu.be/8wk1elugv5A>

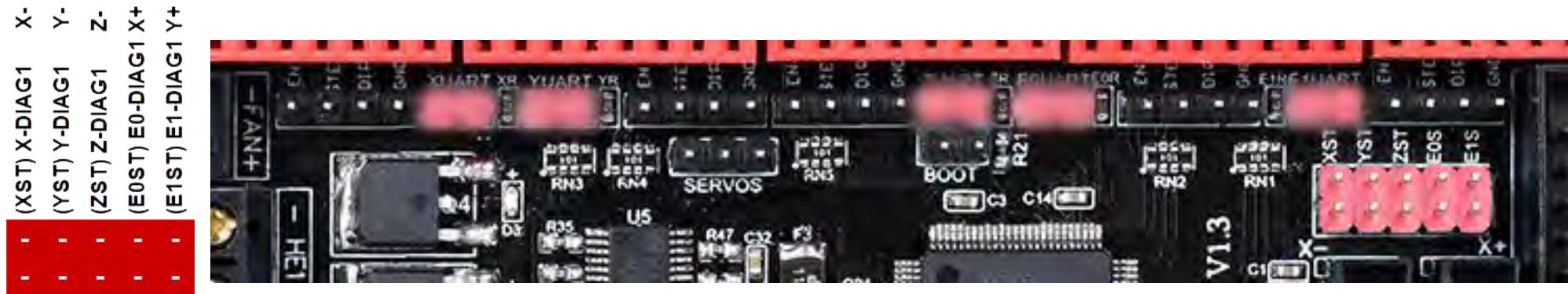


- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

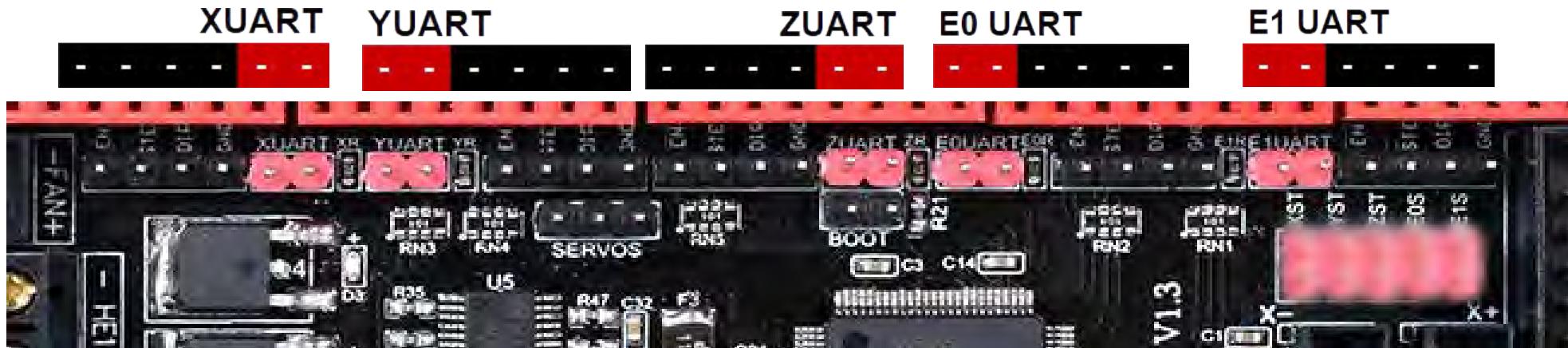
Stand-alone Mode

MKS LV8729

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



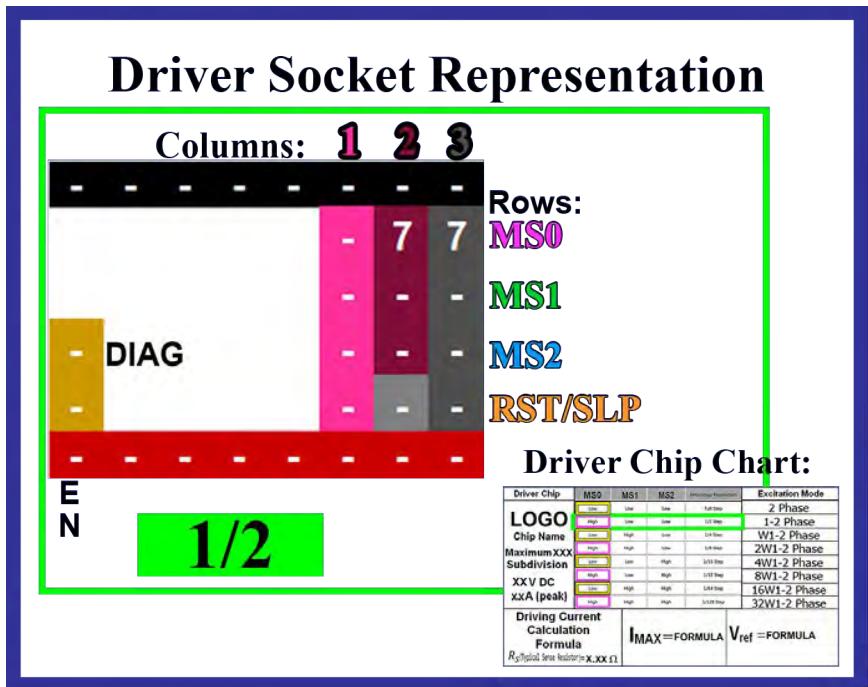
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



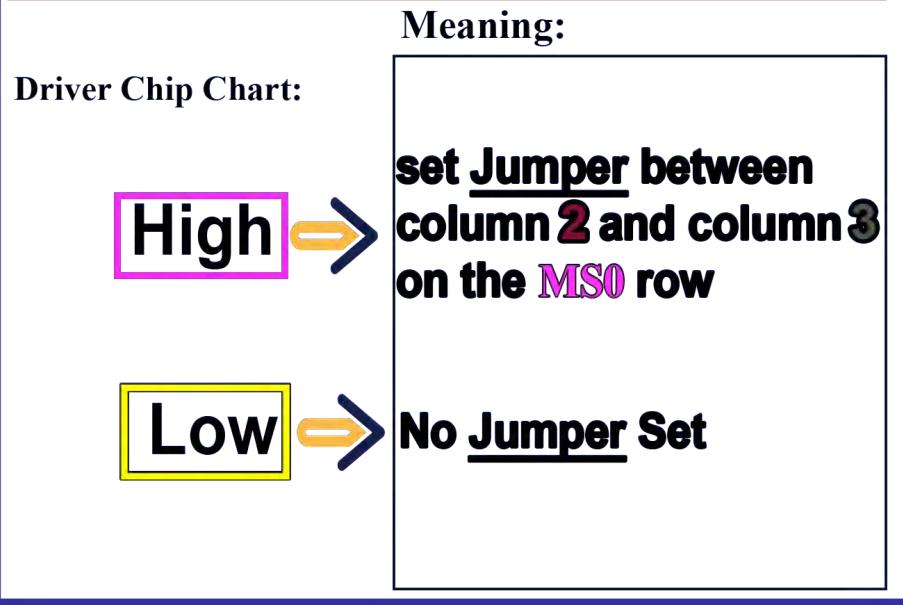
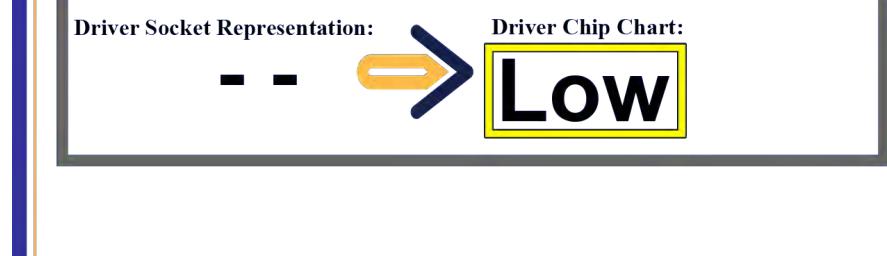
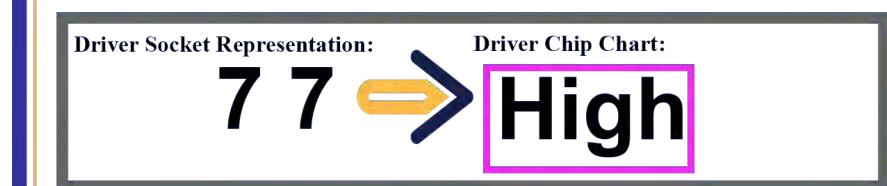
Stand-alone Mode

MKS LV8729

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

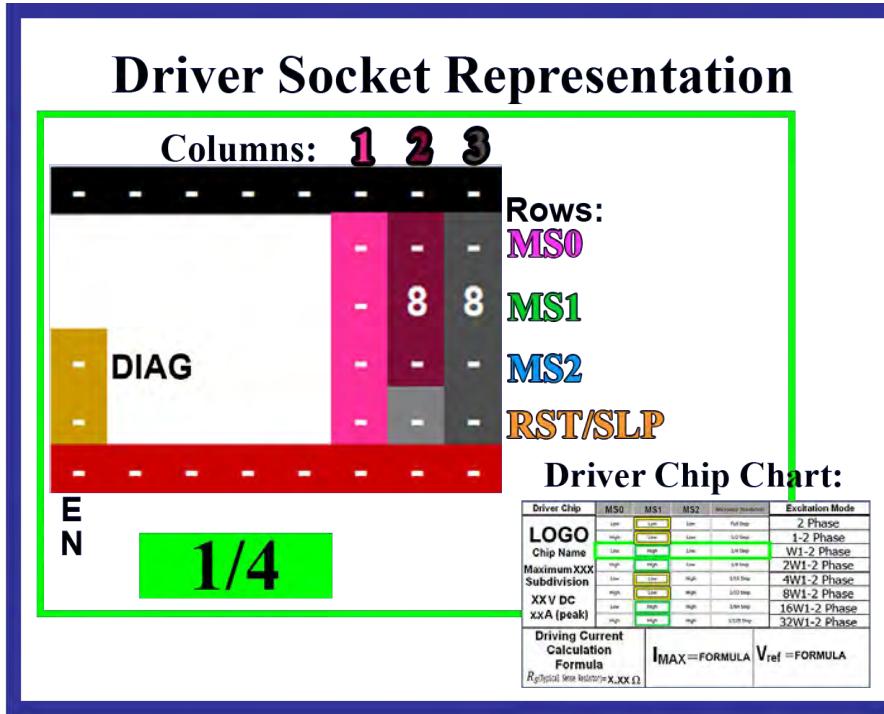


MS0 for Binary State Drivers:

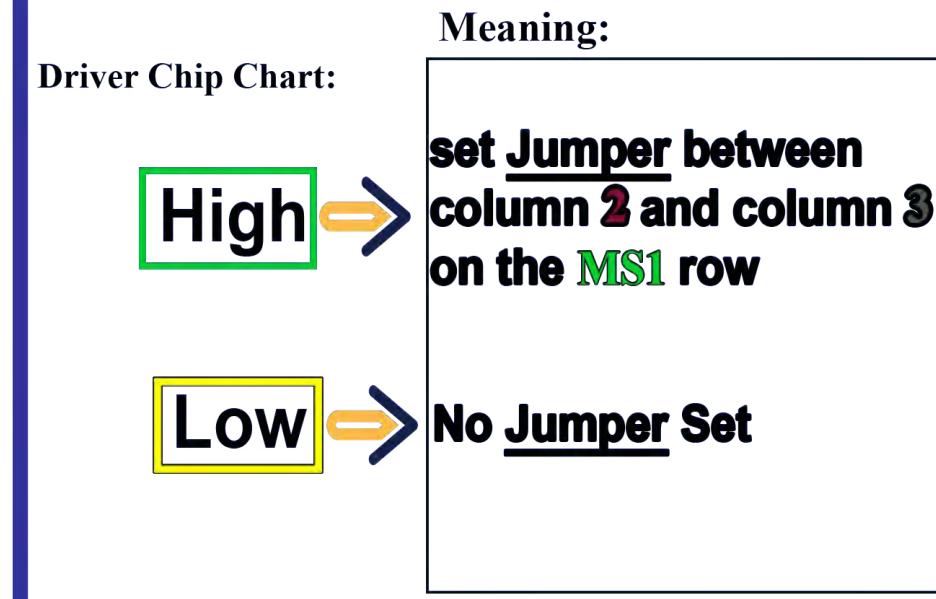
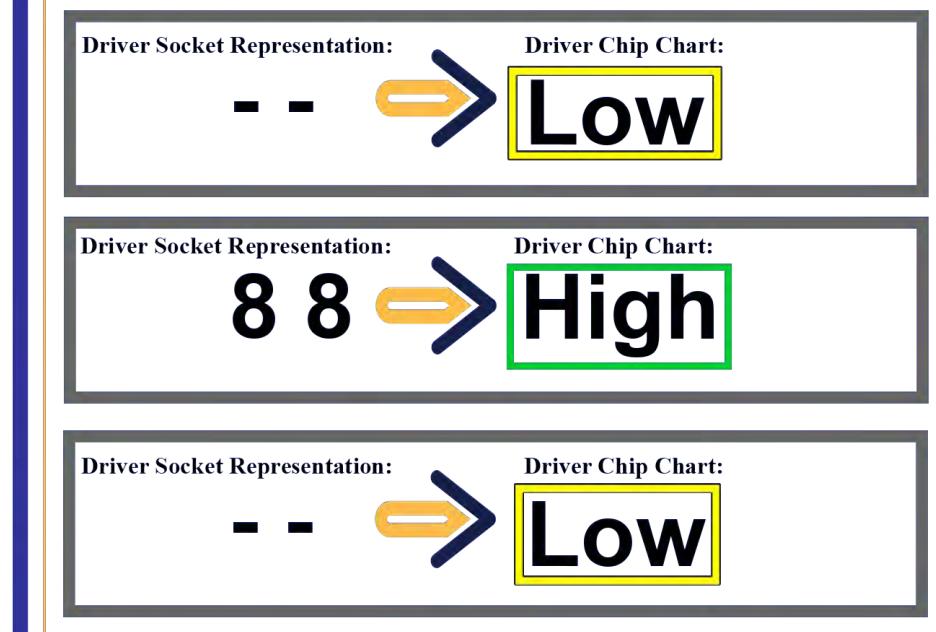


Stand-alone Mode

MKS LV8729



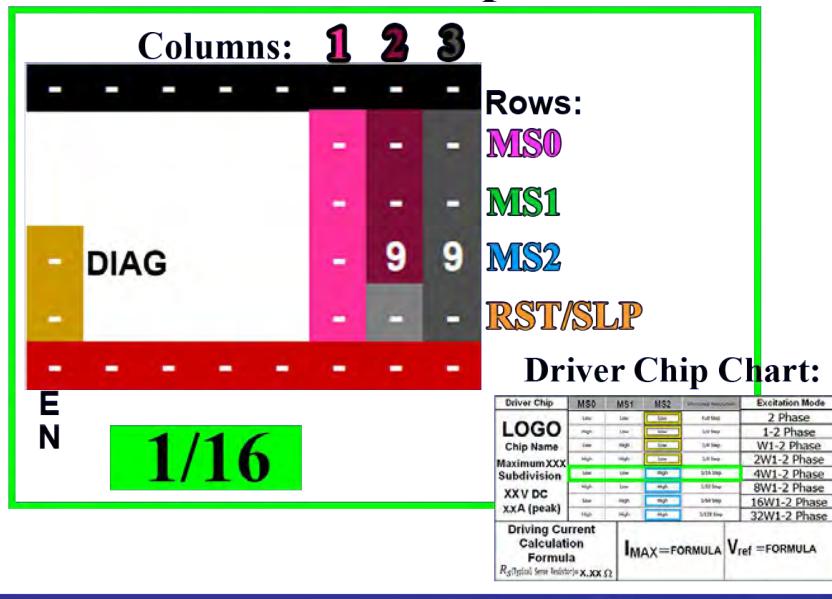
MS1 for Binary State Drivers:



Stand-alone Mode

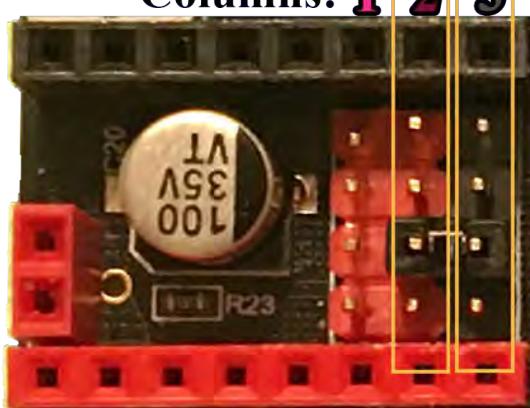
MKS LV8729

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

High →

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

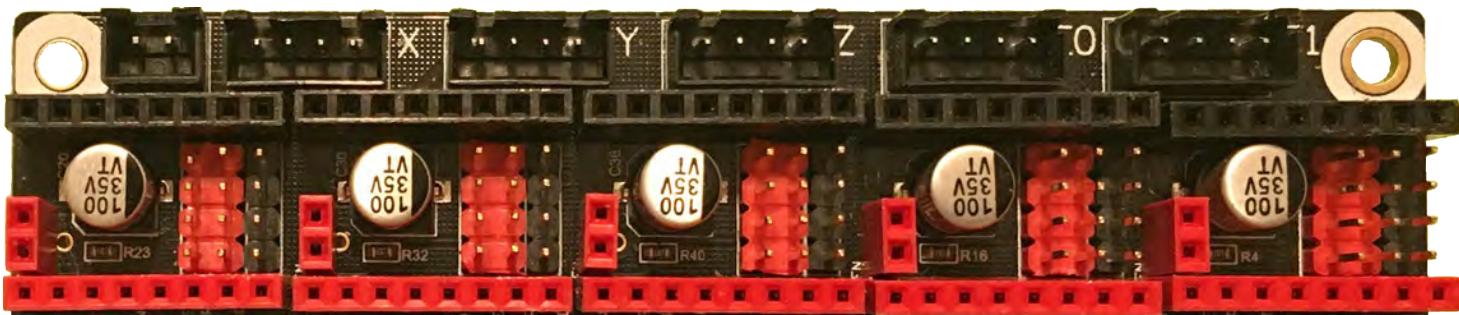
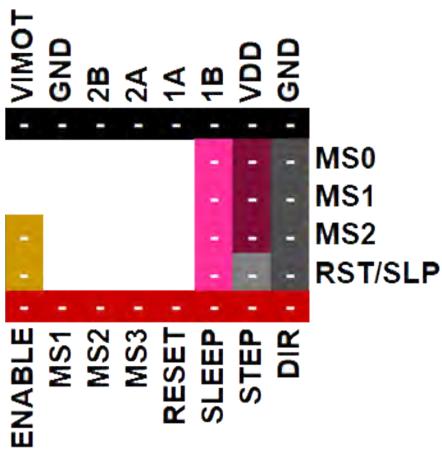
Low →

No Jumper Set

Stand-alone Mode

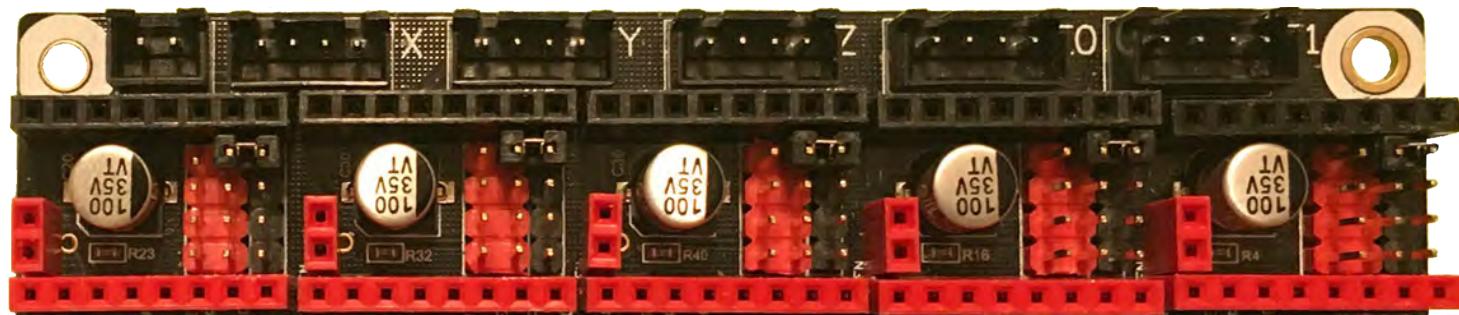
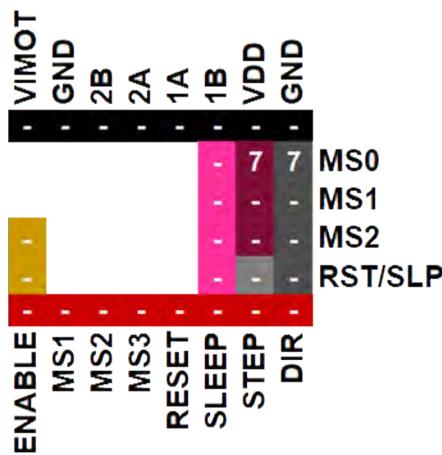
MKS LV8729

STEP



See [Appendix D](#) for legend

1 / 2

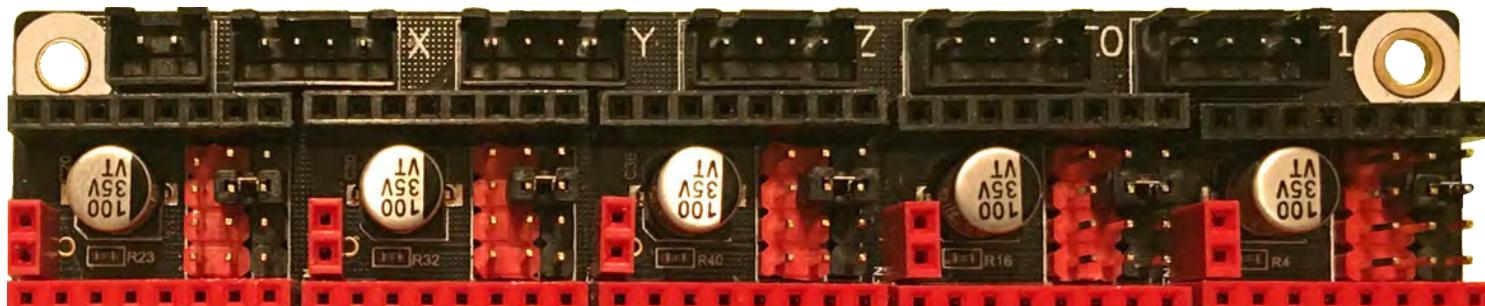
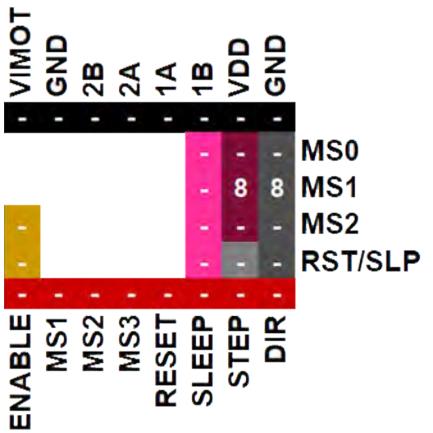


See [Appendix D](#) for legend

Stand-alone Mode

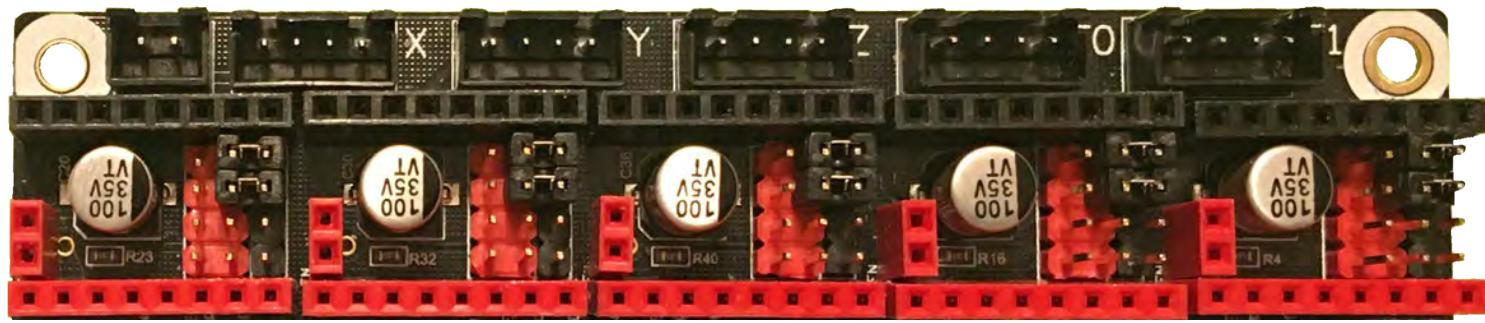
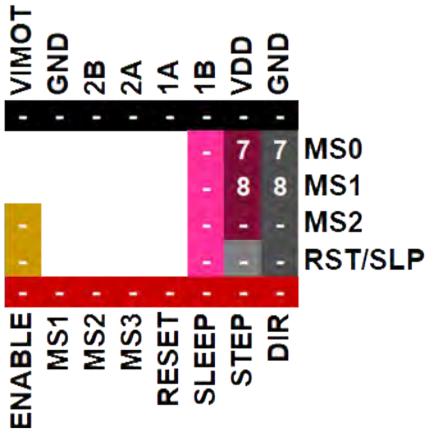
MKS LV8729

1 / 4



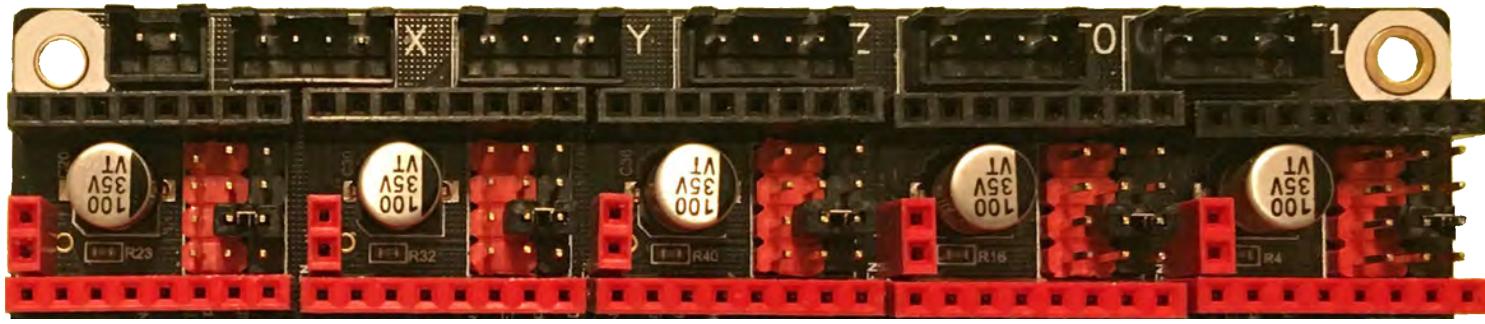
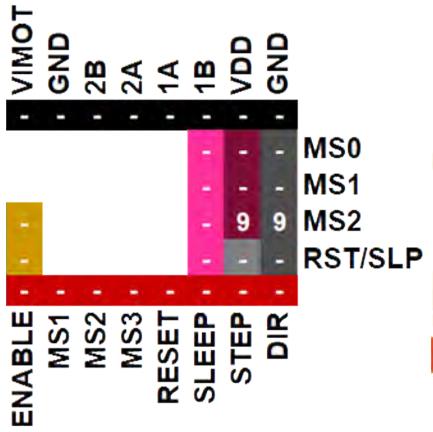
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



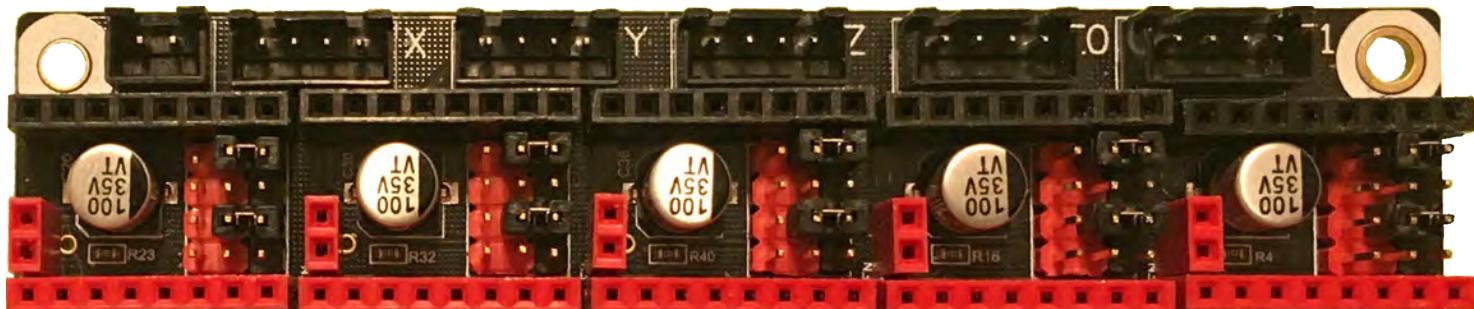
See [Appendix D](#) for legend

Stand-alone Mode

MKS LV8729

1 / 32

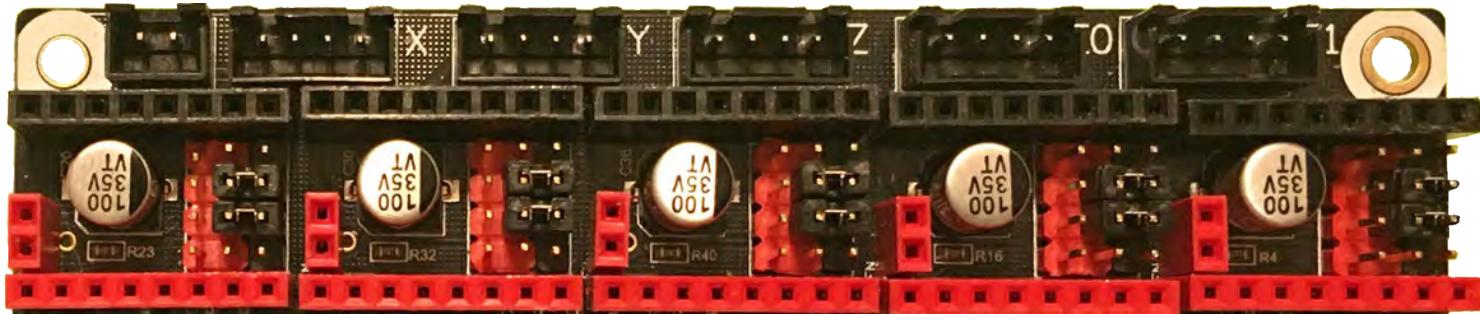
VIMOT	-	GND	-	2B	-	2A	-	1A	-	1B	VDD	GND
ENABLE	-	-	GND	-	-	2B	-	2A	-	1A	VDD	GND
MS0	-	-	-	MS1	-	-	MS2	-	-	MS3	-	-
MS1	-	-	-	MS2	-	-	MS3	-	-	RESET	-	-
MS2	-	-	-	RESET	-	-	SLEEP	-	-	SLEEP	-	-
RST/SLP	-	-	-	SLEEP	-	-	STEP	-	-	STEP	-	-
STEP	-	-	-	STEP	-	-	DIR	-	-	DIR	-	-
DIR	-	-	-	DIR	-	-	-	-	-	-	-	-



See [Appendix D](#) for legend

1 / 64

VIMOT	-	GND	-	2B	-	2A	-	1A	-	1B	VDD	GND
ENABLE	-	-	GND	-	-	2B	-	2A	-	1A	VDD	GND
MS0	-	-	-	MS1	-	-	MS2	-	-	MS3	-	-
MS1	-	-	-	MS2	-	-	MS3	-	-	RESET	-	-
MS2	-	-	-	RESET	-	-	SLEEP	-	-	SLEEP	-	-
RST/SLP	-	-	-	SLEEP	-	-	STEP	-	-	STEP	-	-
STEP	-	-	-	STEP	-	-	DIR	-	-	DIR	-	-
DIR	-	-	-	DIR	-	-	-	-	-	-	-	-



See [Appendix D](#) for legend

1 / 128

VIMOT	-	GND	-	2B	-	2A	-	1A	-	1B	VDD	GND
ENABLE	-	-	GND	-	-	2B	-	2A	-	1A	VDD	GND
MS0	-	-	-	MS1	-	-	MS2	-	-	MS3	-	-
MS1	-	-	-	MS2	-	-	MS3	-	-	RESET	-	-
MS2	-	-	-	RESET	-	-	SLEEP	-	-	SLEEP	-	-
RST/SLP	-	-	-	SLEEP	-	-	STEP	-	-	STEP	-	-
STEP	-	-	-	STEP	-	-	DIR	-	-	DIR	-	-
DIR	-	-	-	DIR	-	-	-	-	-	-	-	-

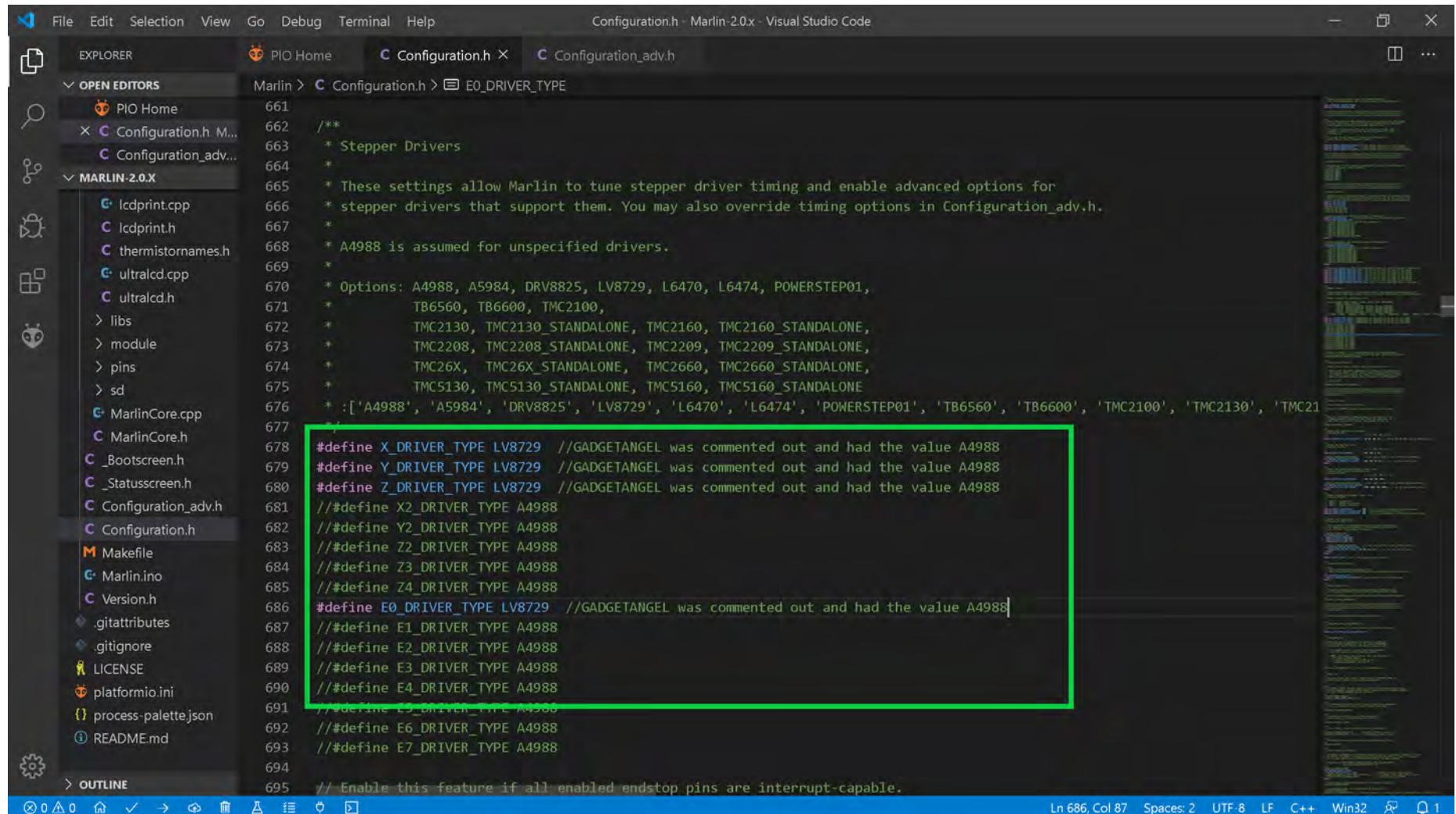


See [Appendix D](#) for legend

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

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

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



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

```

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

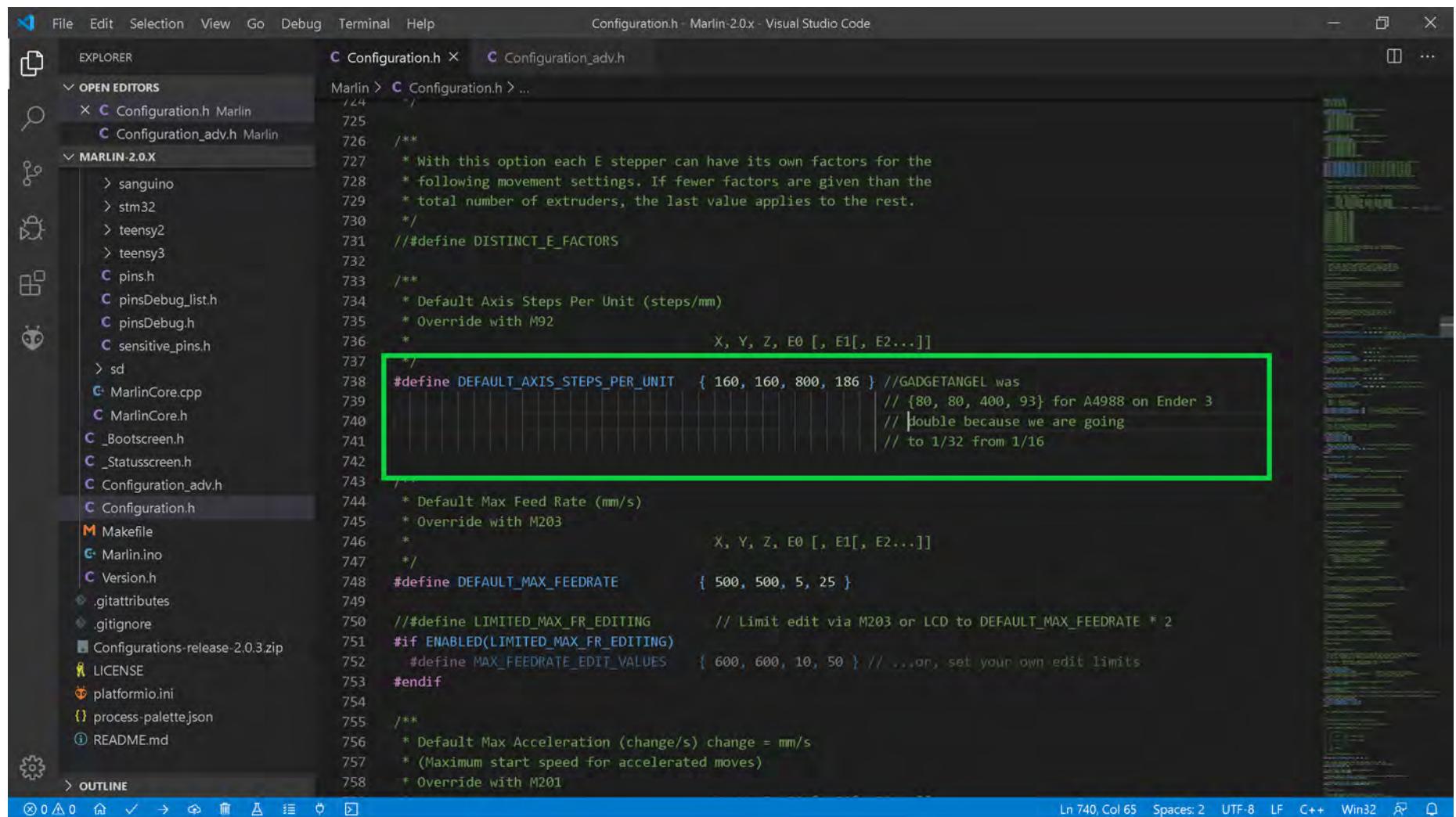
```

A green rectangular box highlights the driver type definitions for axes X, Y, Z, and E0. The code editor status bar at the bottom right shows: Ln 686, Col 87, Spaces: 2, UTF-8, LF, C++, Win32, 87, 1.

- Go to the next page.

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

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



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

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

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

- Go to the next page.

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

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

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

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X". The "Configuration.h" file is selected.
- Right Side (Code Editor):** Displays the content of Configuration.h. A green box highlights the section starting at line 1035, which defines the `INVERT_Z_DIR` macro. The code shows various configurations for stepper motor direction inversion based on GCODE TANGO settings.

```
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GCODE TANGO was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GCODE TANGO was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GCODE TANGO was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GCODE TANGO was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```

- Go to the next page.

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

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

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

EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

OPEN EDITORS Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src... pins_BTT_SKR_common.h Marlin\src... Configuration_adv.h Marlin

MARLIN-2.0.X

- > samd
- > sanguino
- > stm32f1
- > stm32f4
- > stm32f7
- > teensy2
- > teensy3
- C pins.h
- C pinsDebug.h
- C pinsDebug.list
- C sensitive_pins.h
- > sd
- E MarlinCore.cpp
- C MarlinCore.h
- C _Bootscreen.h
- C _Statusscreen.h
- C Configuration.h
- C Configuration_adv.h
- M Makefile
- E Marlin.ino
- C Version.h
- .editorconfig
- .gitattributes
- .gitignore
- LICENSE
- platformio.ini
- process-palette.json
- README.md

OUTLINE TIMELINE

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 //#define BLUETOOTH
128
129 // Choose the name from below to best match your setup
130 #ifndef MOTHERBOARD
131   #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin
 ====== [SUCCESS] Took 130.61 seconds =====

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

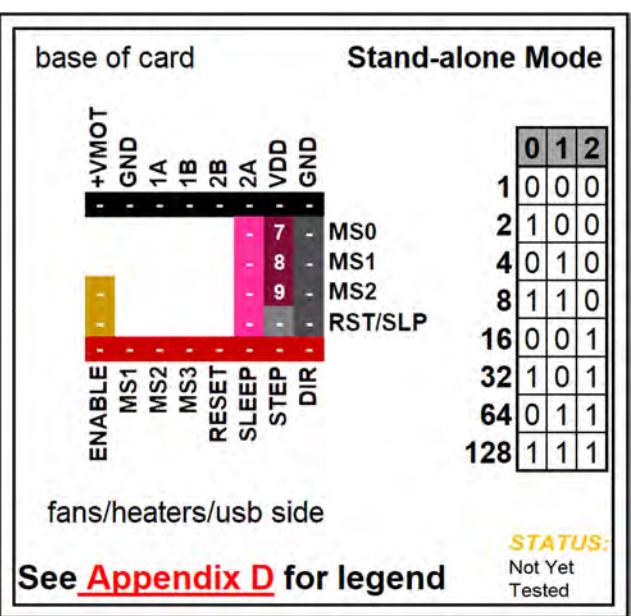
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

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

The screenshot shows the Visual Studio Code interface for Marlin 2.0.x setup. The Explorer sidebar on the left lists files and folders related to Marlin 2.0.x, including Configuration.h, pins_BTT_SKR_V1_3.h, pins_BTT_SKR_common.h, and Configuration_adv.h. The main editor window displays Configuration.h with code for defining BAUDRATE and MOTHERBOARD. A green rectangular box highlights the line "#define MOTHERBOARD BOARD_BTT_SKR_V1_3". The bottom status bar shows the build process: "Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o" through "Building .pio\build\LPC1768\firmware.bin". The status bar also indicates "[SUCCESS] Took 130.61 seconds". A green rectangular box highlights the "LPC1768" entry in the "Environment" list, which shows "SUCCESS" and a duration of "00:02:10.606". The bottom toolbar features several icons, including a red box highlighting the "Upload" icon (a right-pointing arrow).

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

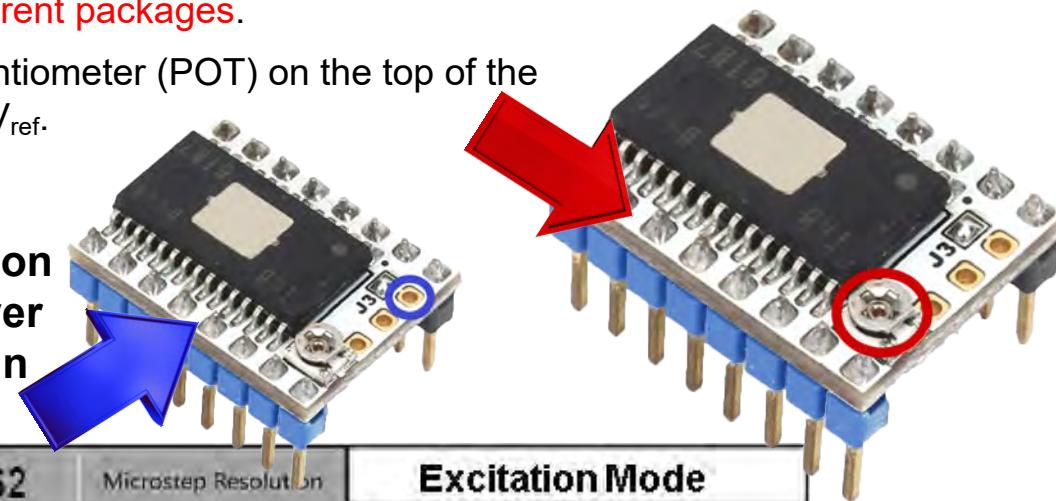


FYSETC S6128 V1.1

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

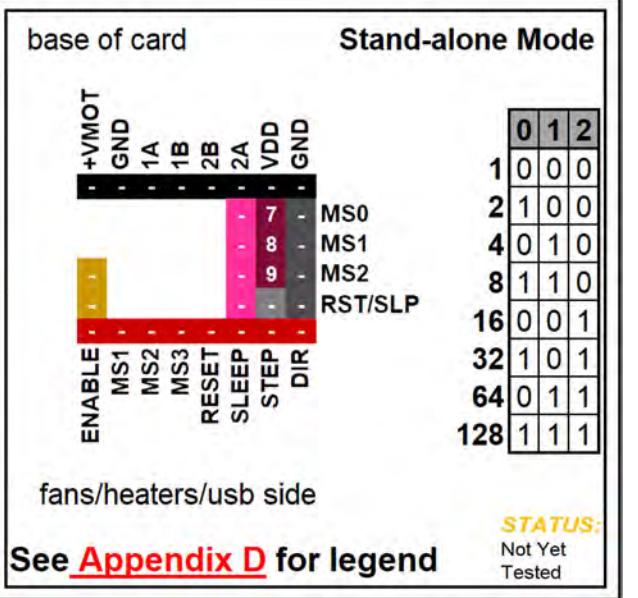
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

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



Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
FYSETC SureStepr SD6128 Maximum 128 Subdivision 35V DC 2.2A (peak)	Low	Low	Low	Full Step	2 Phase
	High	Low	Low	1/2 Step	1-2 Phase
	Low	High	Low	1/4 Step	W1-2 Phase
	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.1Ω	$I_{MAX} = \frac{V_{ref}}{5 * R_S}$		$V_{ref} = 5 * I_{MAX} * R_S$		

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

**Driver Chip Chart:**

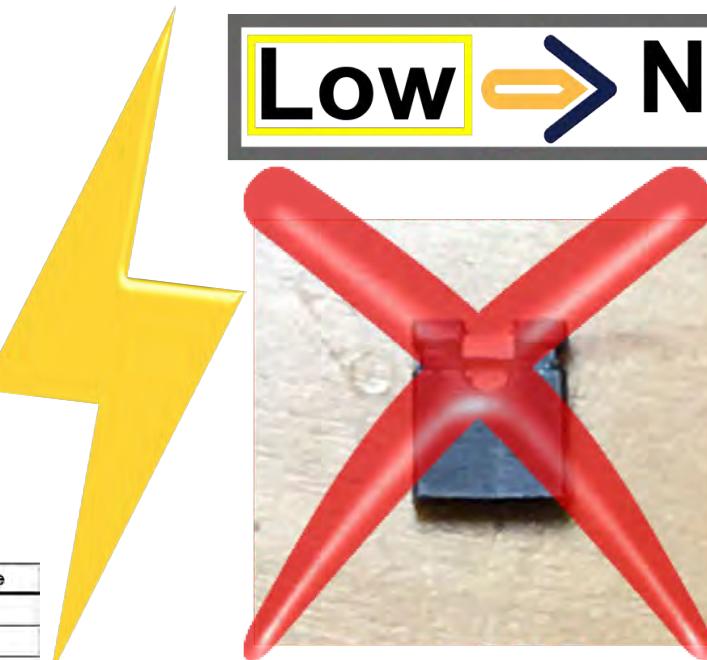
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XXV DC xxA (peak)	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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

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

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

FYSETC S6128 V1.1**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Low ➔ No Jumper



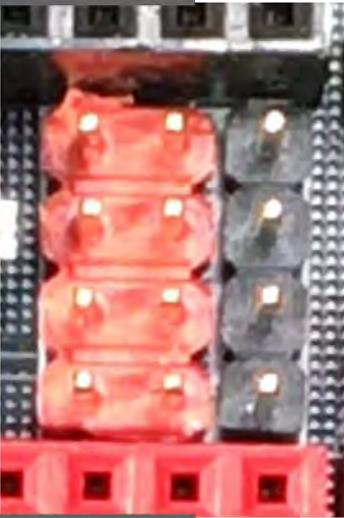
Rows:

MS0

MS1

MS2

RST/SLP



High ➔ Jumper Set

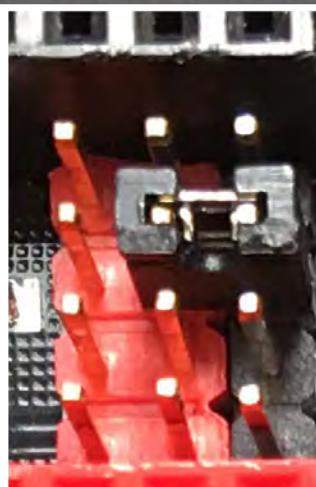
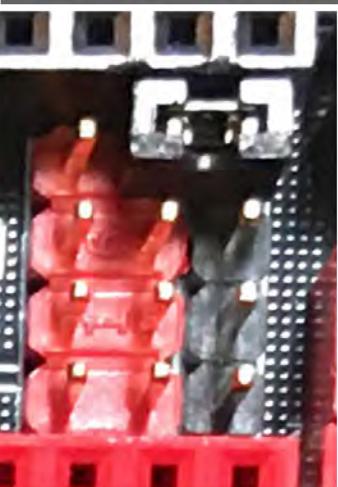
Rows:

MS0

MS1

MS2

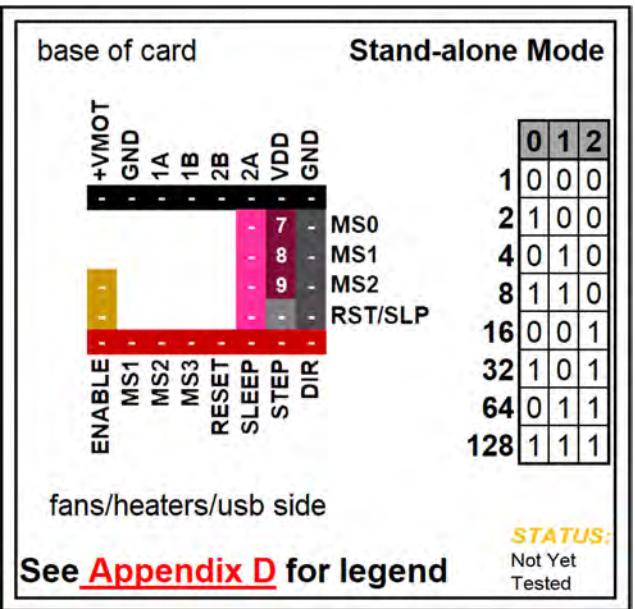
RST/SLP



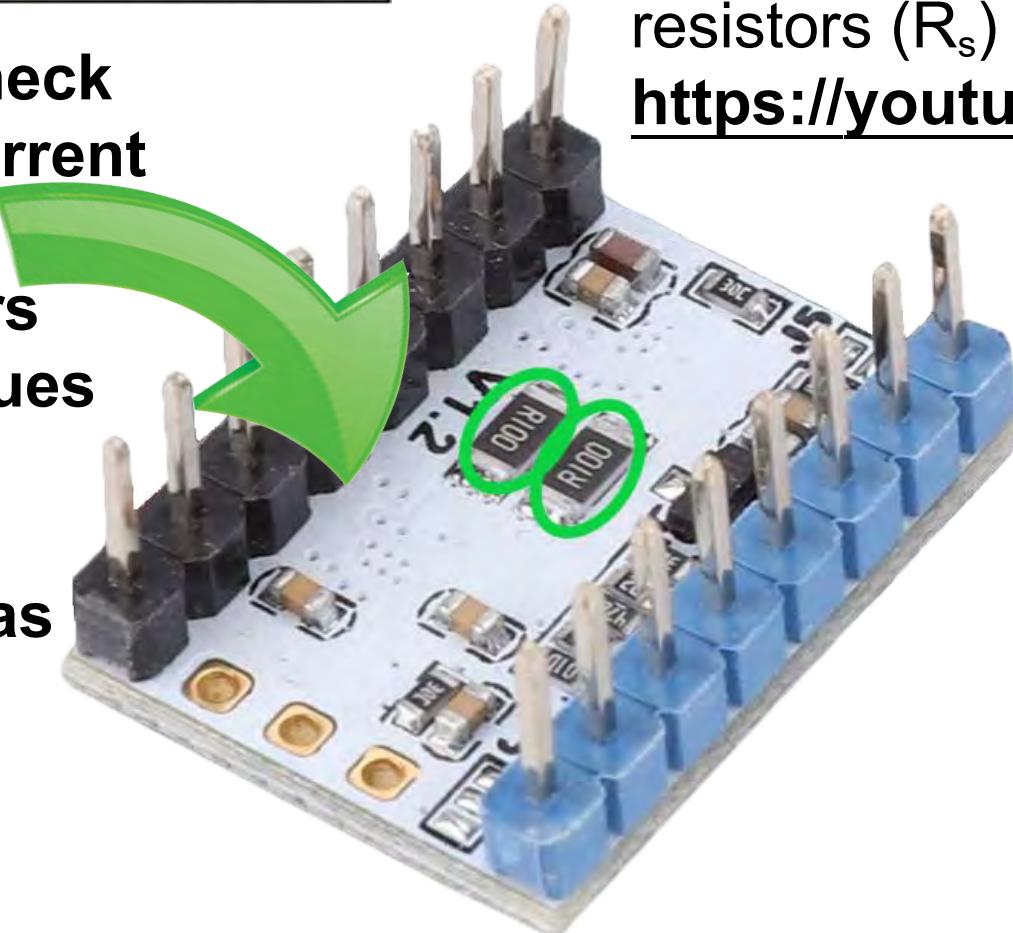
MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH



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



FYSETC S6128 V1.1

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

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

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

$R_s = R050$ is 0.05 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

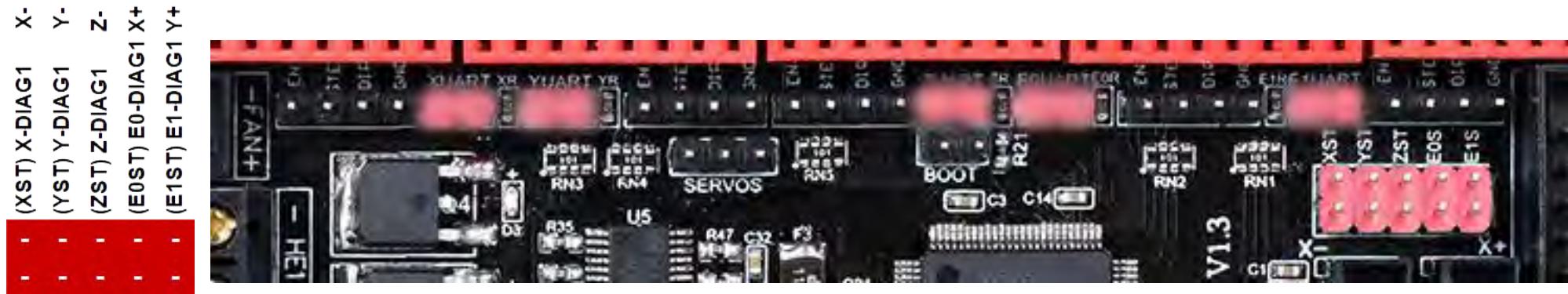
$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

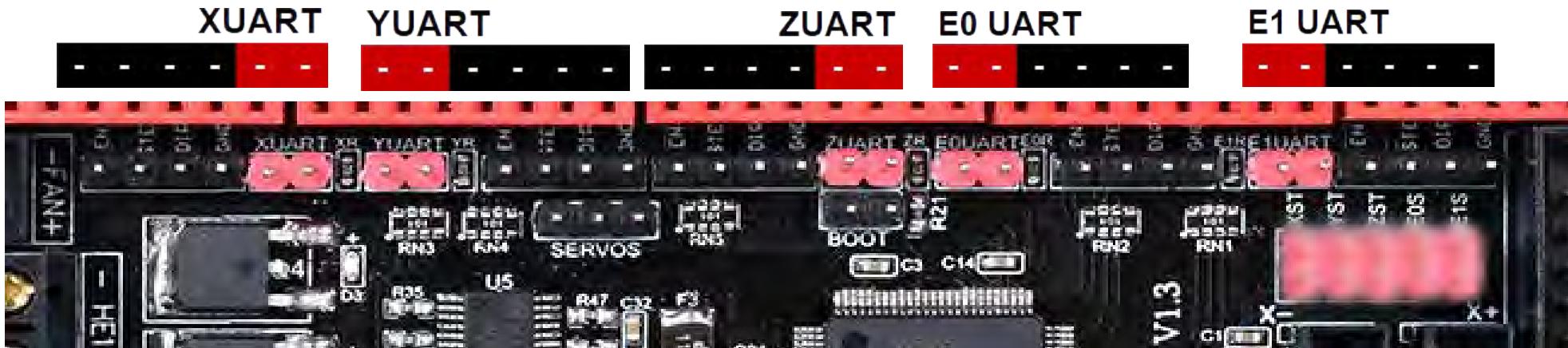
Stand-alone Mode

FYSETC S6128 V1.1

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



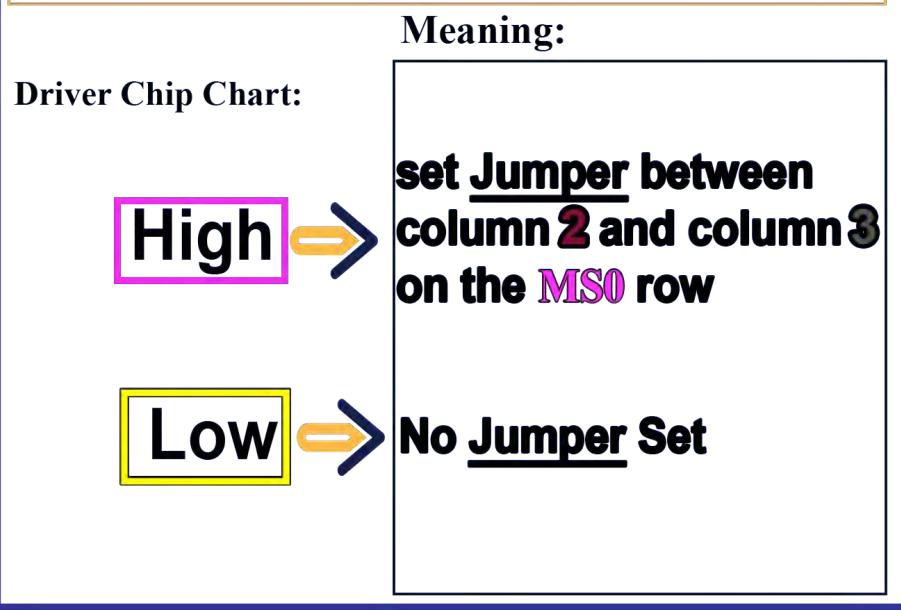
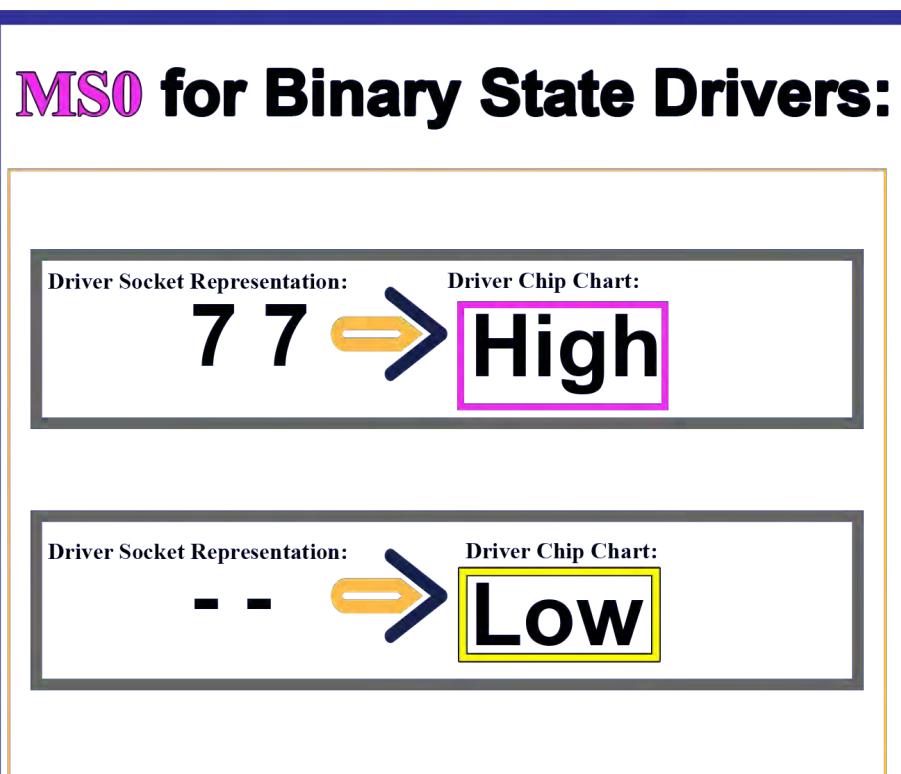
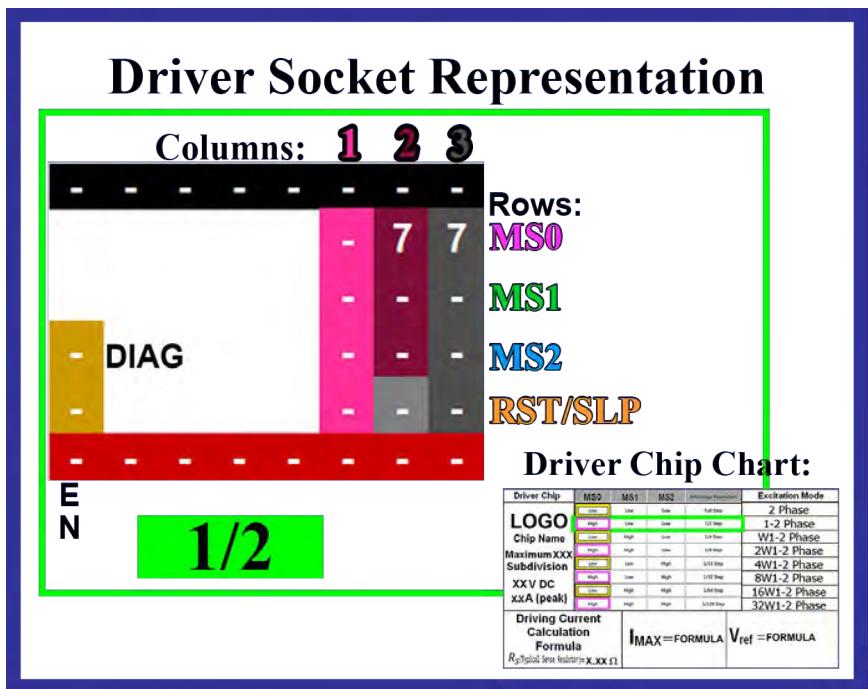
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



Stand-alone Mode

FYSETC S6128 V1.1

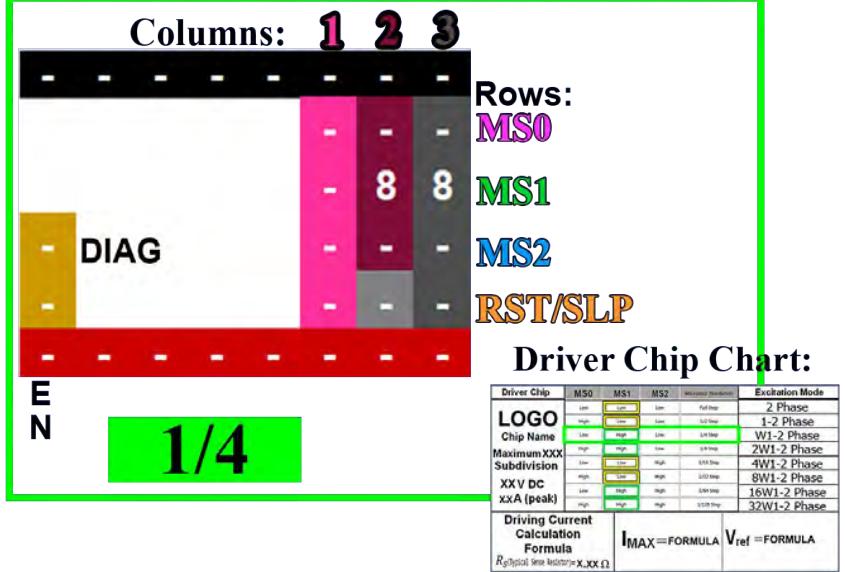
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers



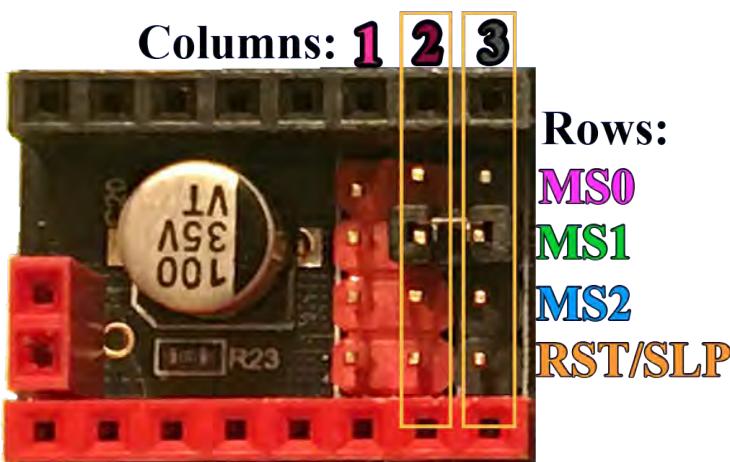
Stand-alone Mode

FYSETC S6128 V1.1

Driver Socket Representation



High:



MS1 for Binary State Drivers:

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

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

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

Meaning:

Driver Chip Chart:

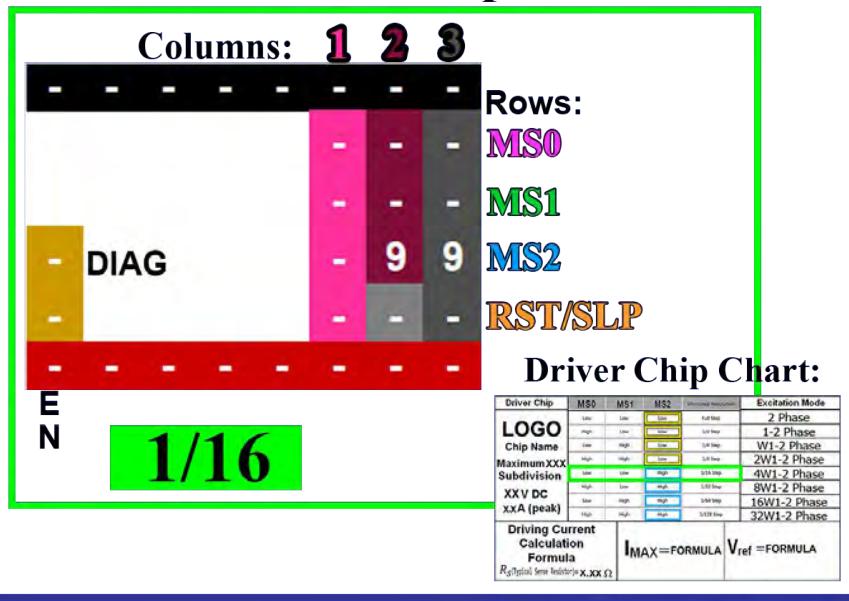
High → set Jumper between column 2 and column 3 on the MS1 row

Low → No Jumper Set

Stand-alone Mode

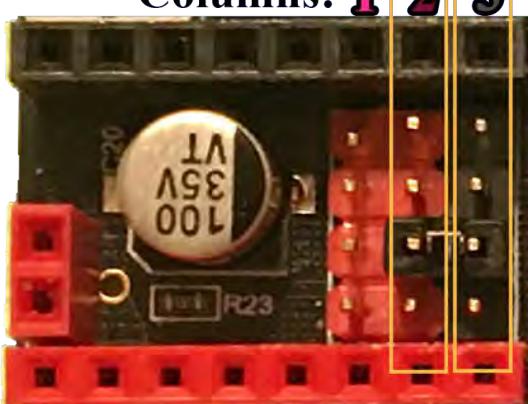
FYSETC S6128 V1.1

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:

Driver Socket Representation:

- -

Driver Chip Chart:

LOW

Driver Socket Representation:

9 9

Driver Chip Chart:

High

Meaning:

Driver Chip Chart:

High

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

Low

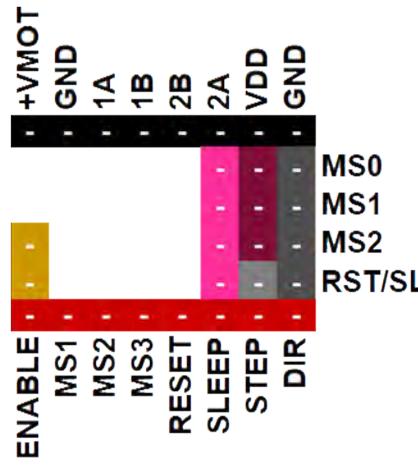
No Jumper Set

Stand-alone Mode

FYSETC S6128 V1.1

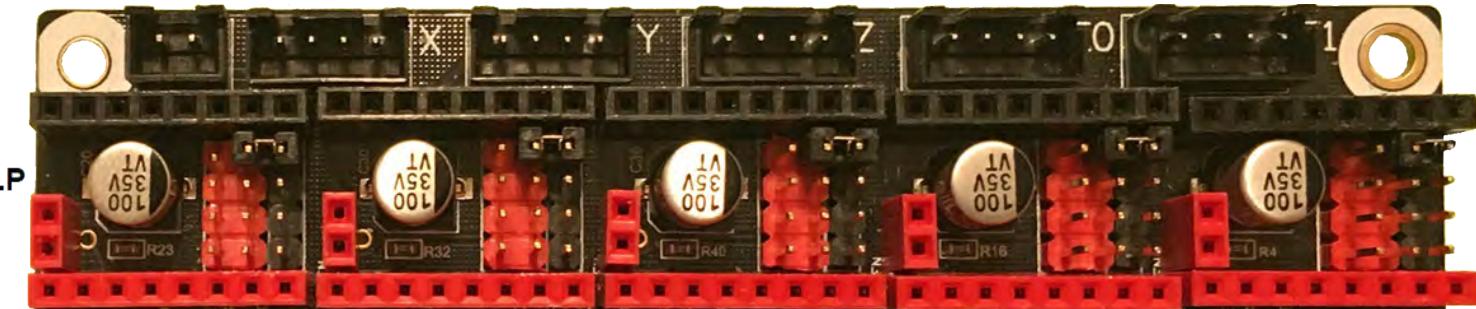
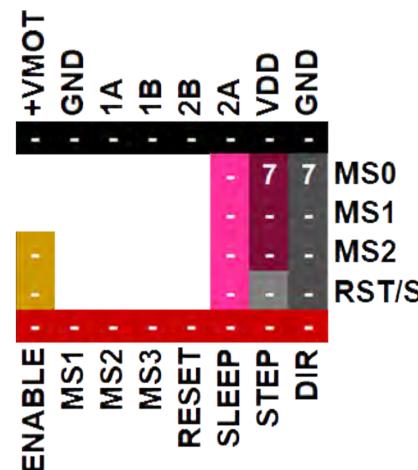
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

STEP



See [Appendix D](#) for legend

1 / 2



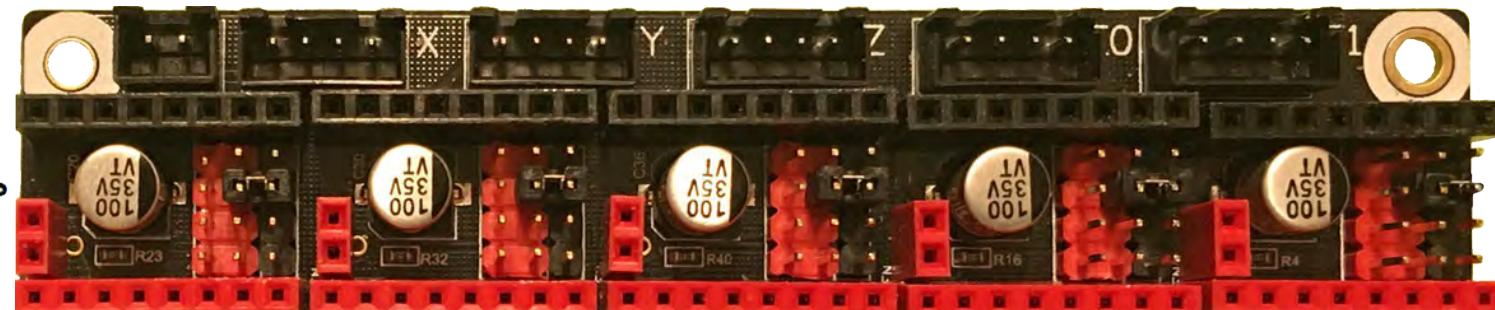
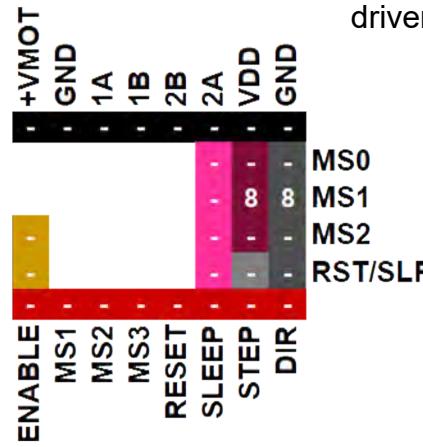
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC S6128 V1.1

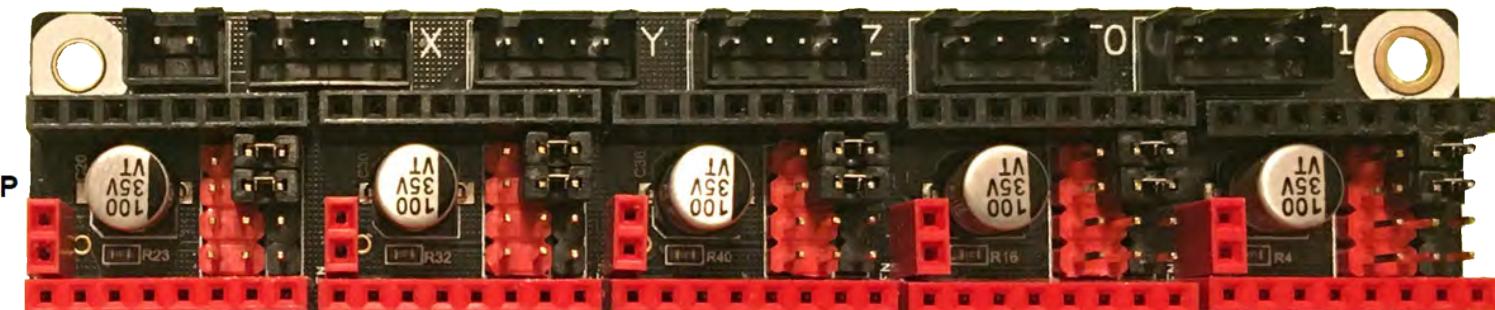
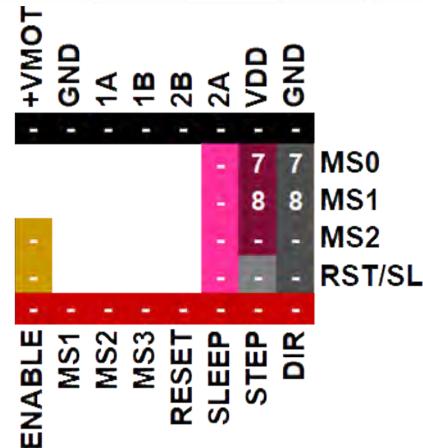
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 4



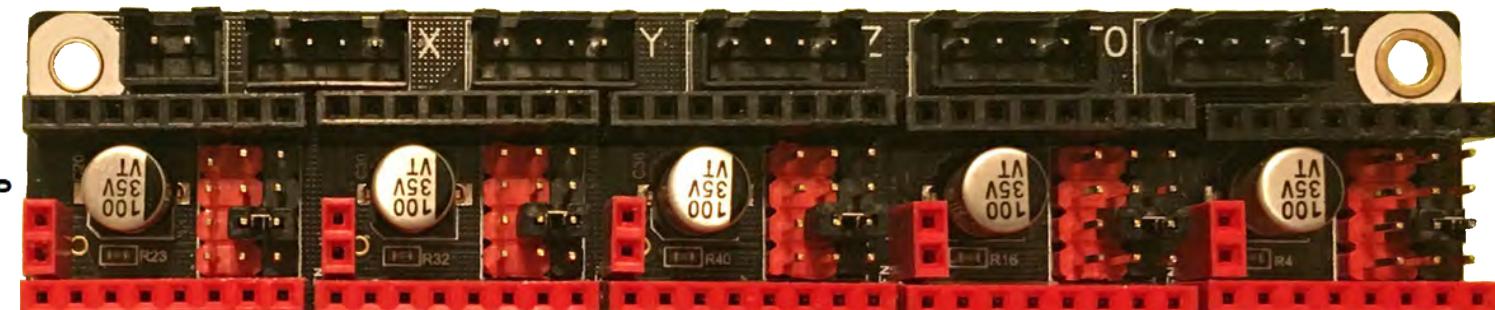
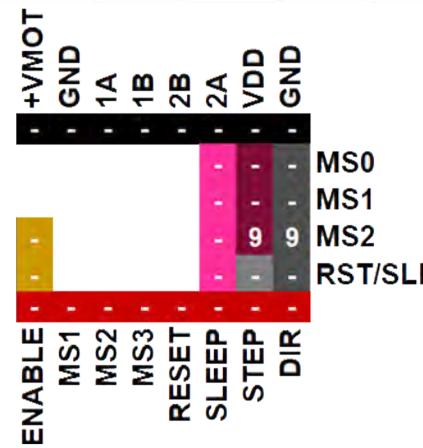
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



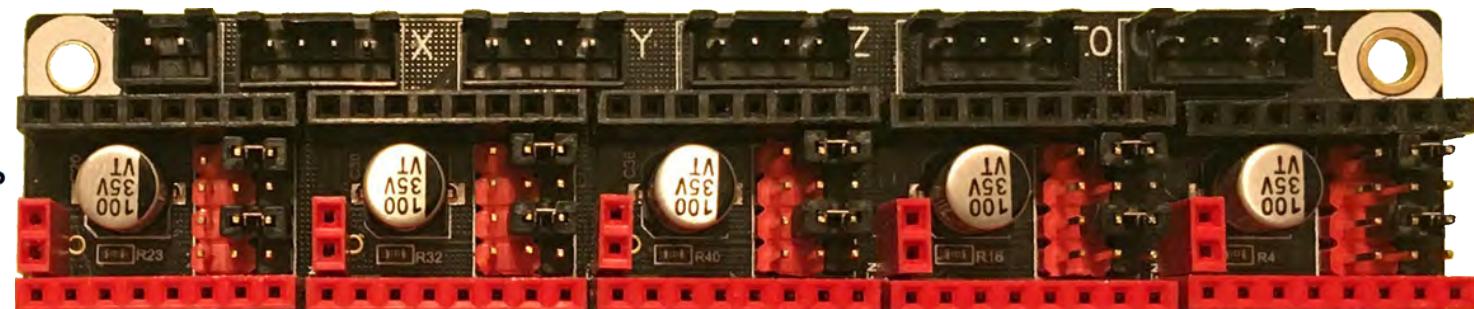
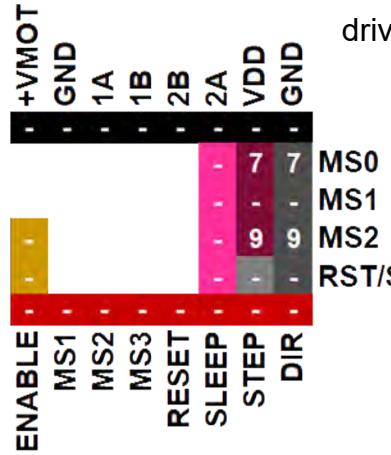
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC S6128 V1.1

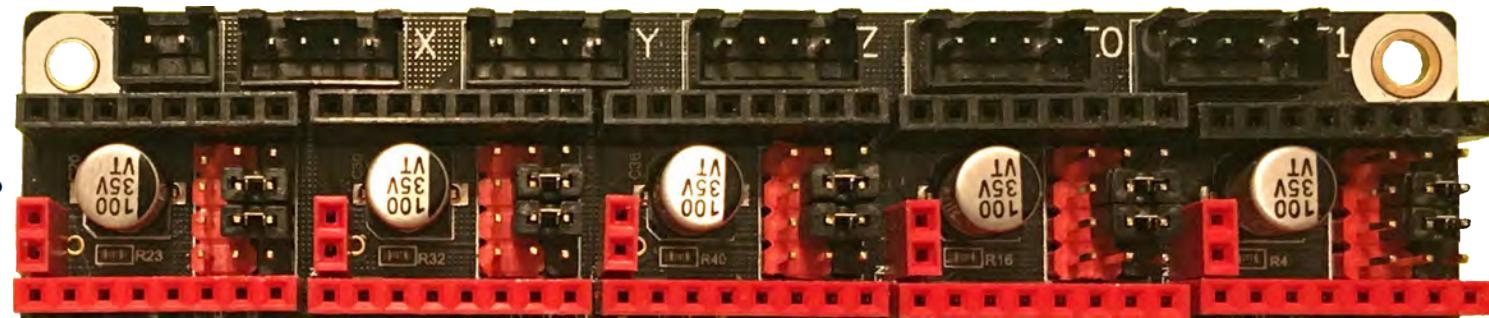
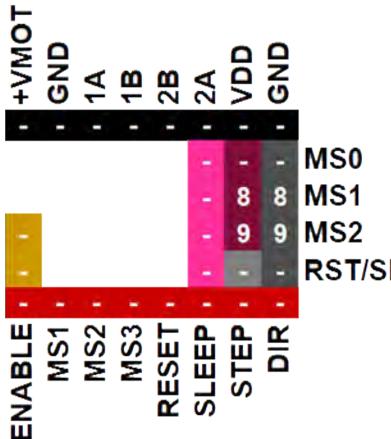
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 32



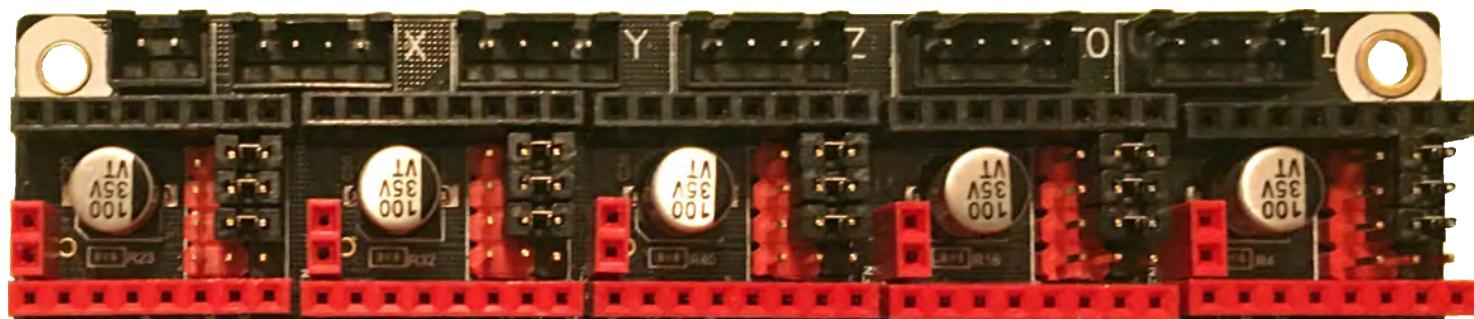
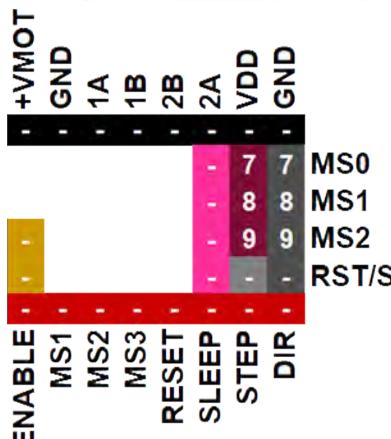
See [Appendix D](#) for legend

1 / 64



See [Appendix D](#) for legend

1 / 128

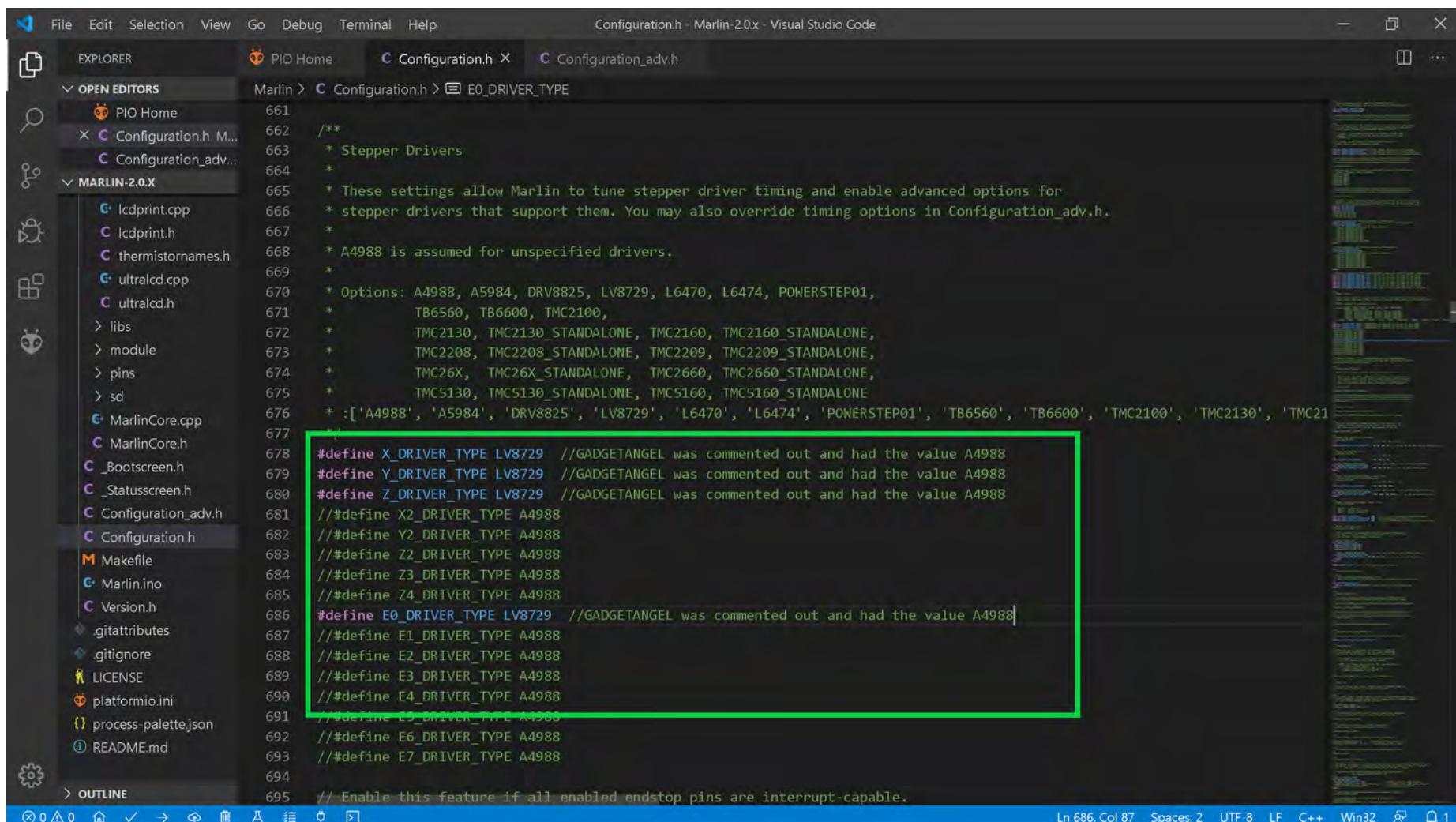


See [Appendix D](#) for legend

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

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

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



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 *           T86560, T86600, TMC2100,
671 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'T86560', 'T86600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130', 'TMC5160']
676 */
677
#define X_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE LV8729 //GADGETANGEL was commented out and had the value A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

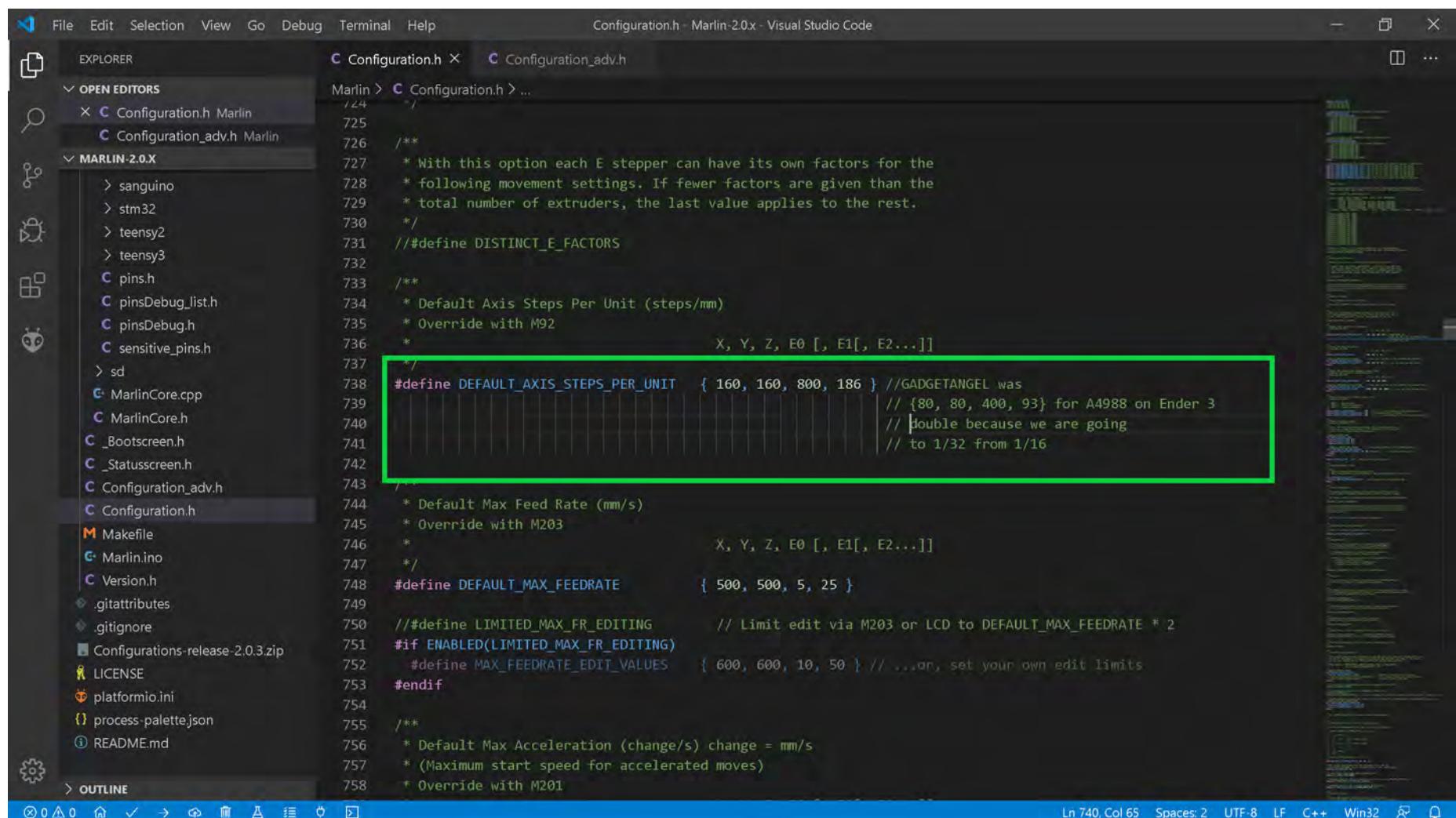
```

A green rectangular box highlights the driver type definitions for axes X, Y, Z, and E0, which were previously commented out (indicated by the double slash prefix). The code editor status bar at the bottom right shows: Ln 686, Col 87, Spaces: 2, UTF-8, LF, C++, Win32, 1 tab, 1 line.

- Go to the next page.

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

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



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

```

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

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [, E1[, E2...]]
 */

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

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

#ifndef LIMITED_MAX_FR_EDITING
#define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif

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

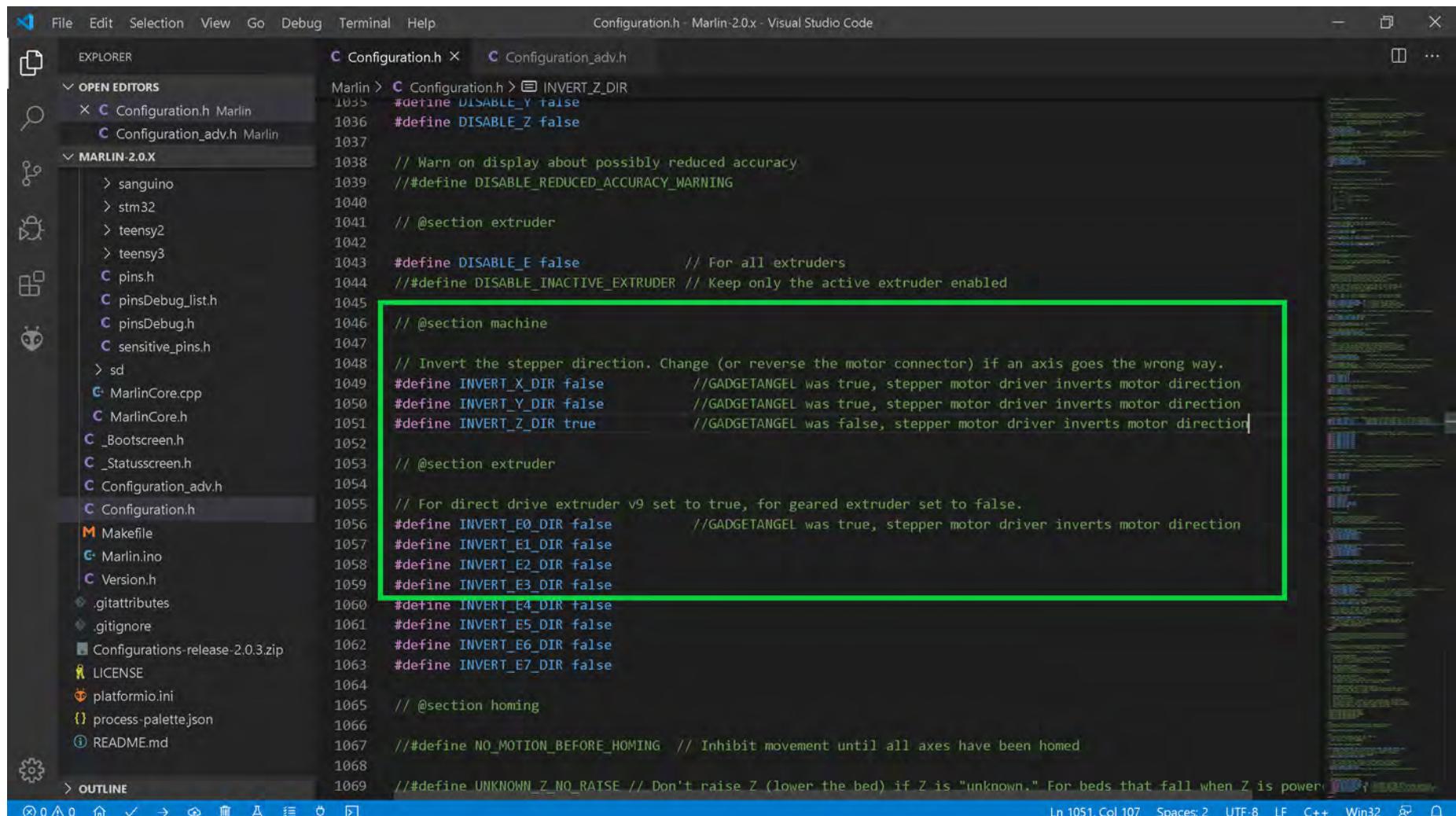
```

The line `#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 }` is highlighted with a green rectangular box. The status bar at the bottom of the code editor shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

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

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



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

```

File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  1035 #define DISABLE_Y false
  1036 #define DISABLE_Z false
  1037
  1038 // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  1041 // @section extruder
  1042
  1043 #define DISABLE_E false          // For all extruders
  1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1045
  1046 // @section machine
  1047
  1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1052
  1053 // @section extruder
  1054
  1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
  1056 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1057 #define INVERT_E1_DIR false
  1058 #define INVERT_E2_DIR false
  1059 #define INVERT_E3_DIR false
  1060 #define INVERT_E4_DIR false
  1061 #define INVERT_E5_DIR false
  1062 #define INVERT_E6_DIR false
  1063 #define INVERT_E7_DIR false
  1064
  1065 // @section homing
  1066
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

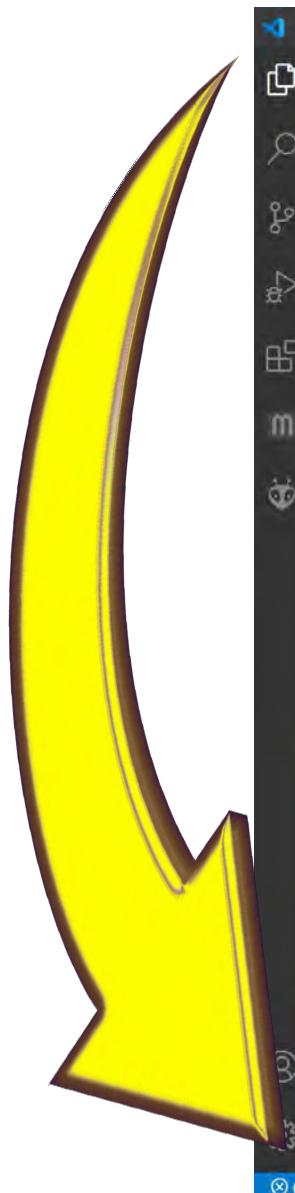
```

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

- Go to the next page.

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

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






Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	DUE	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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

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

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin
MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
TIMELINE

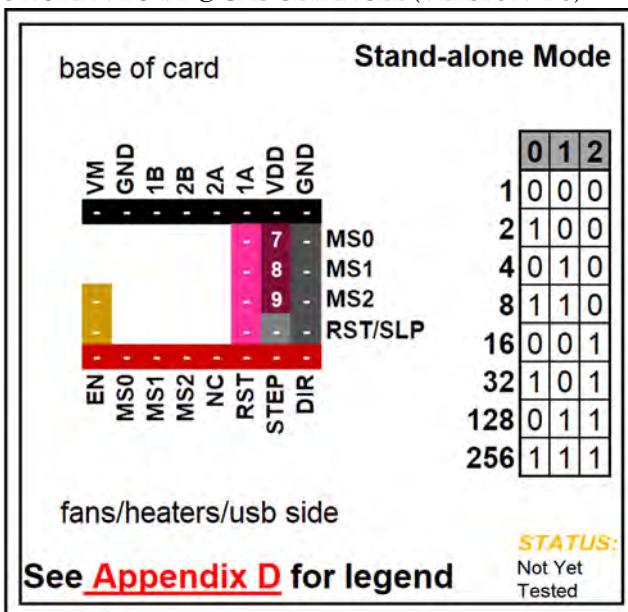
```

```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds
Environment Status Duration
----- -----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_LVDS IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

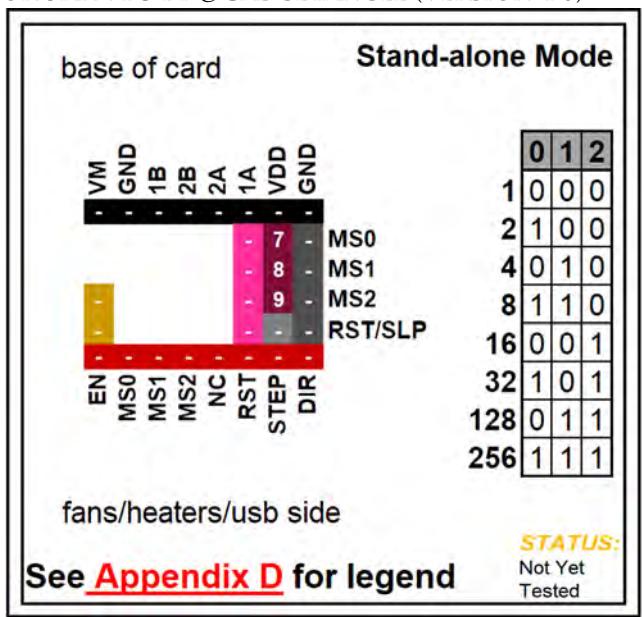
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3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

FYSETC ST820

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

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

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

**Driver Chip Chart:**

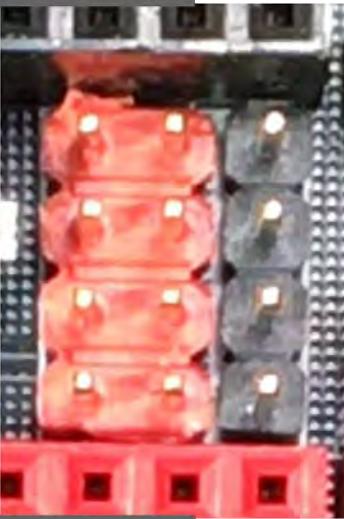
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XXV DC xxA (peak)	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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

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

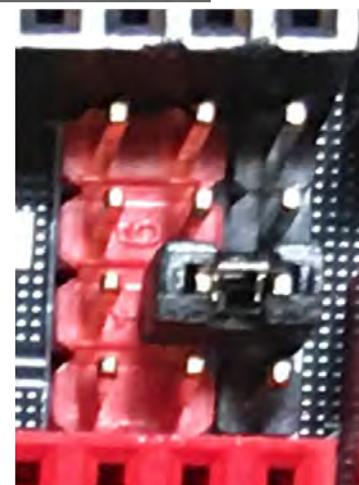
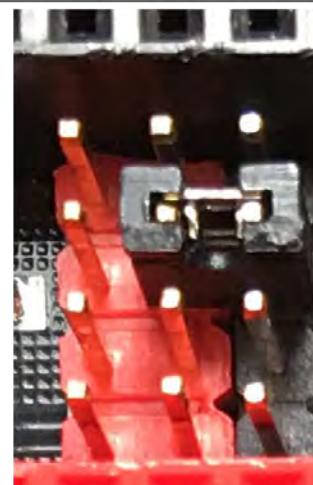
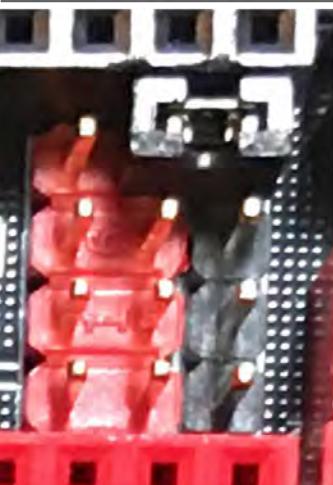
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

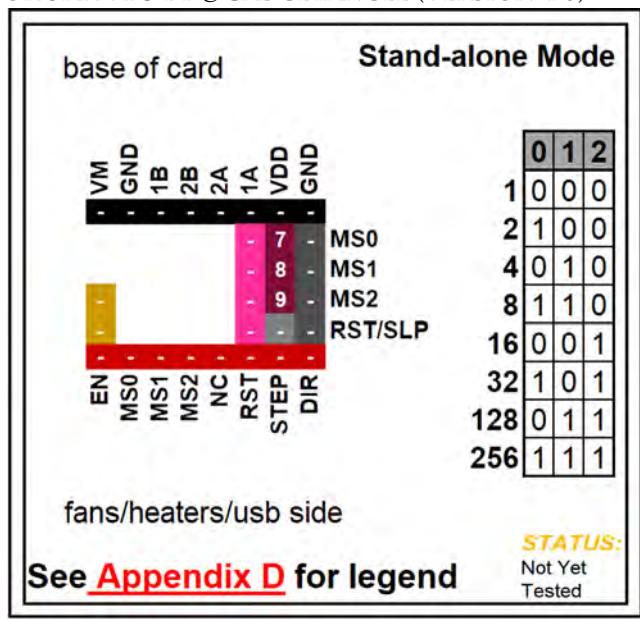
FYSETC ST820**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Low **No Jumper**

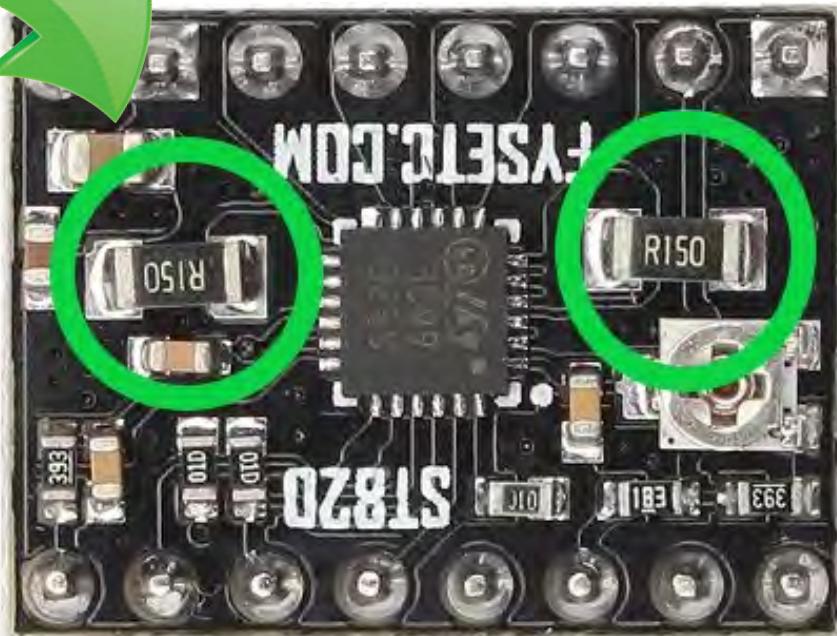
**Rows:****MS0****MS1****MS2****RST/SLP**

High **Jumper Set**

Rows:**MS0****MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**



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



FYSETC ST820

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

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



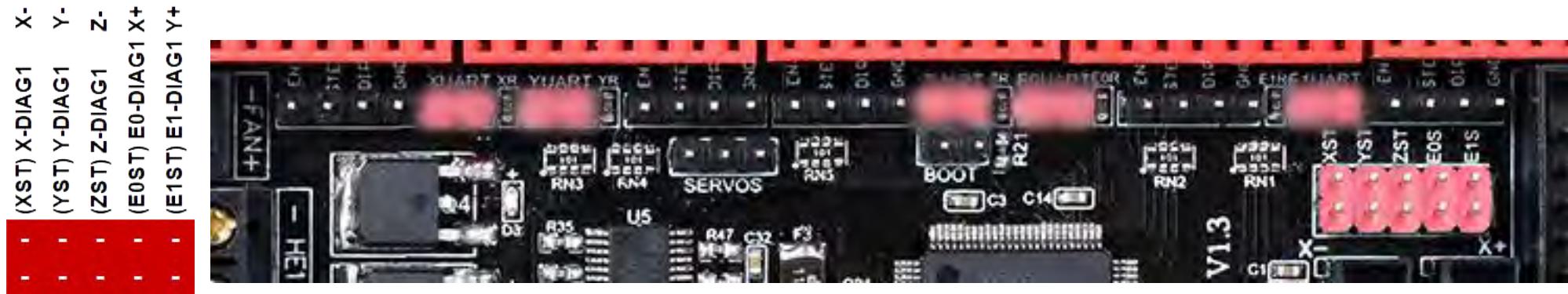
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

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

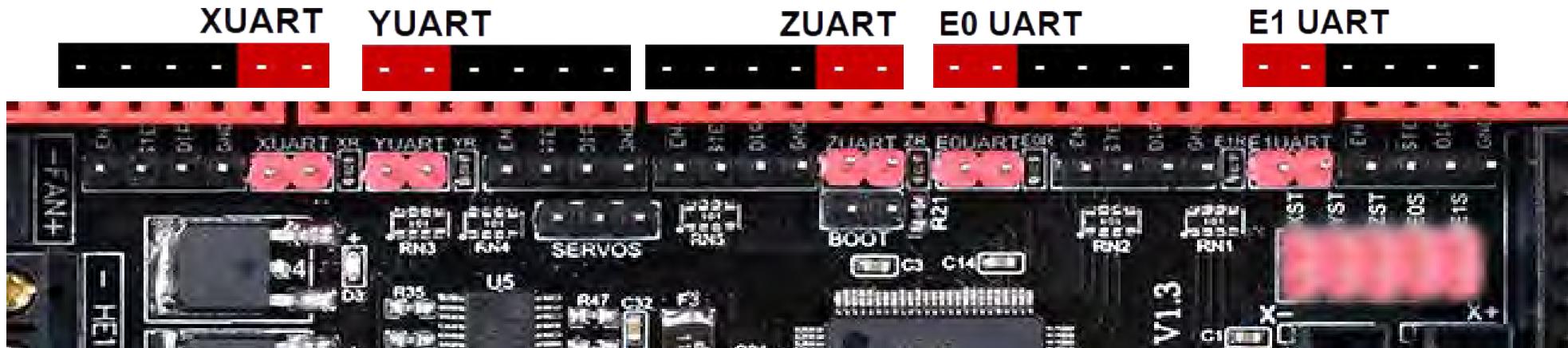
Stand-alone Mode

FYSETC ST820

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



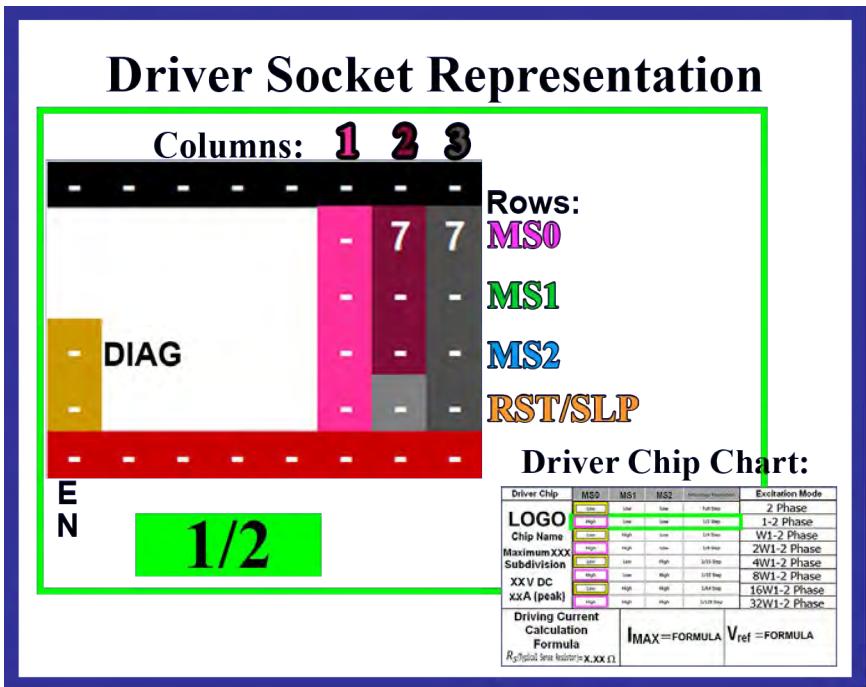
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



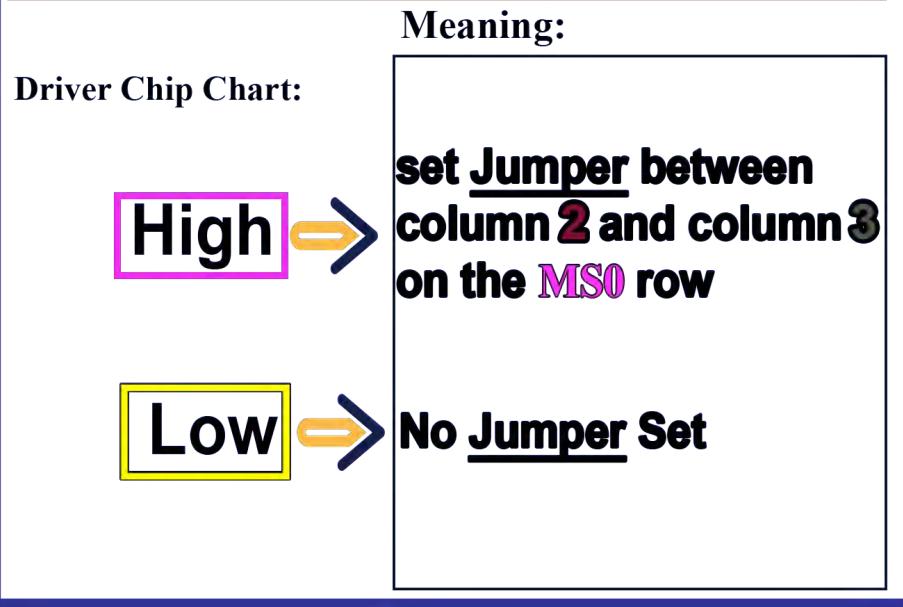
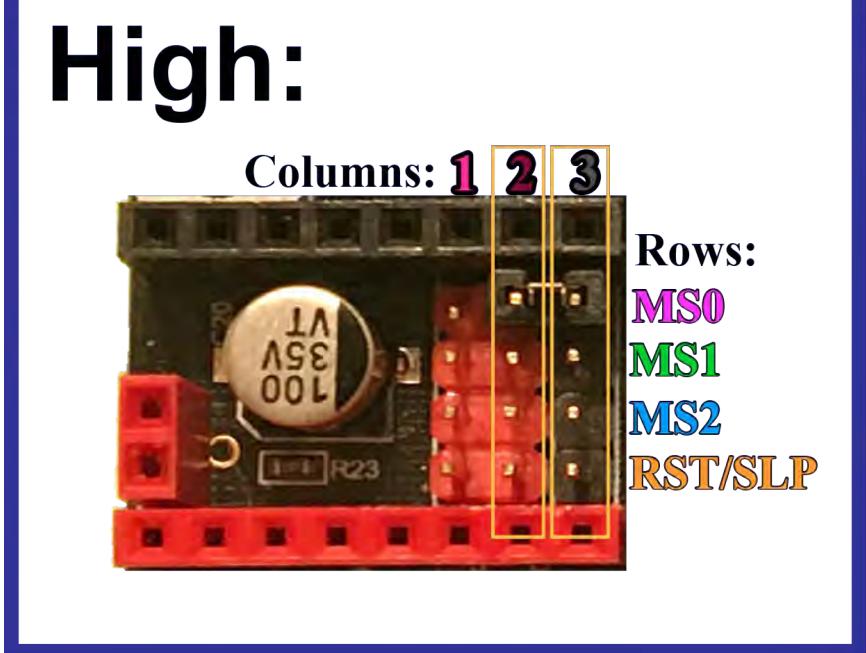
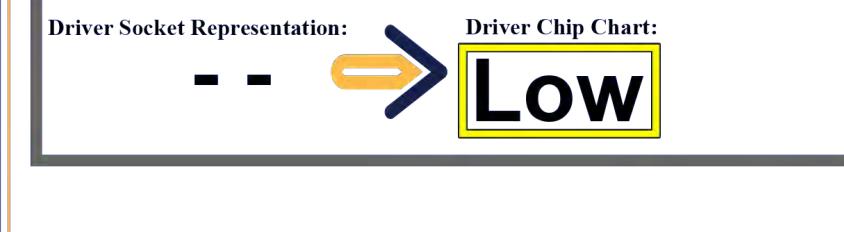
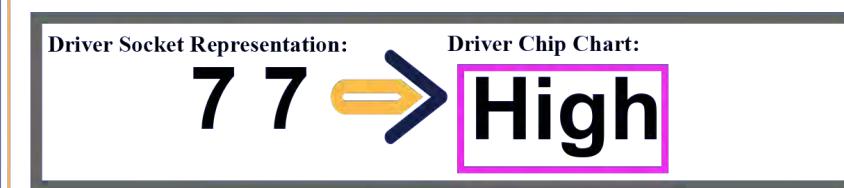
Stand-alone Mode

FYSETC ST820

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

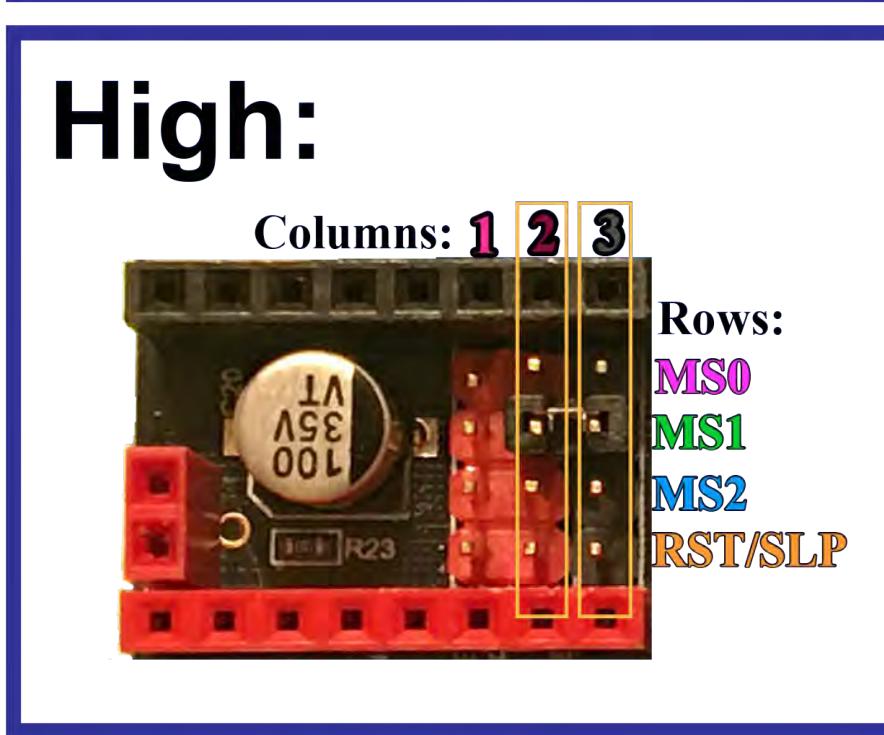
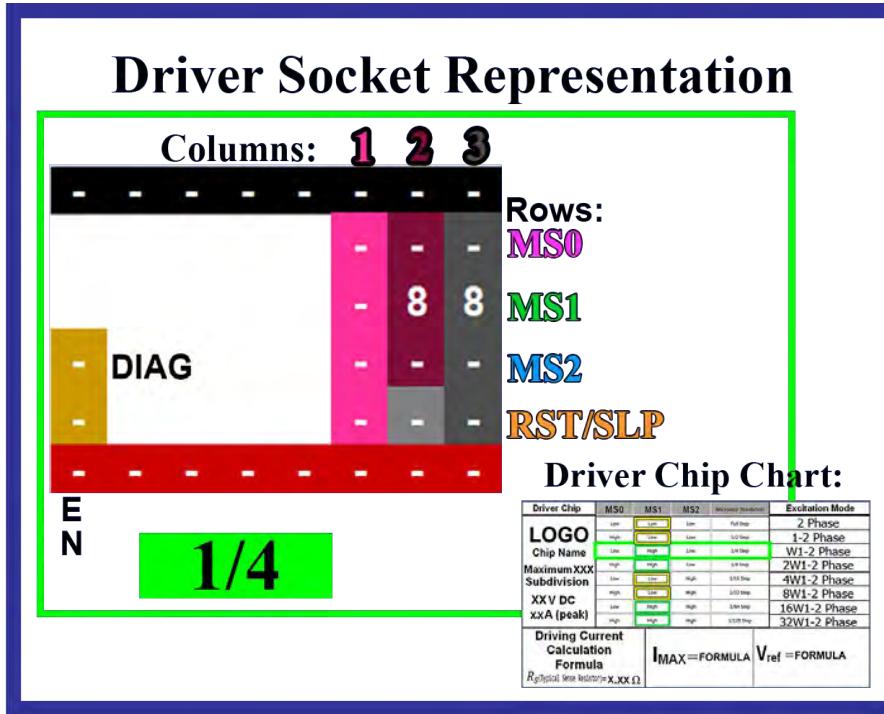


MS0 for Binary State Drivers:

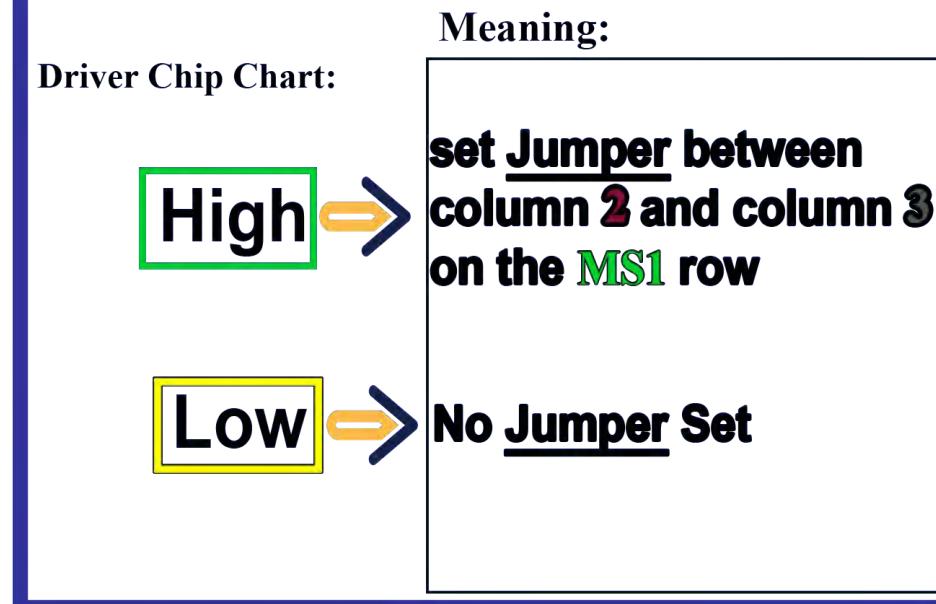
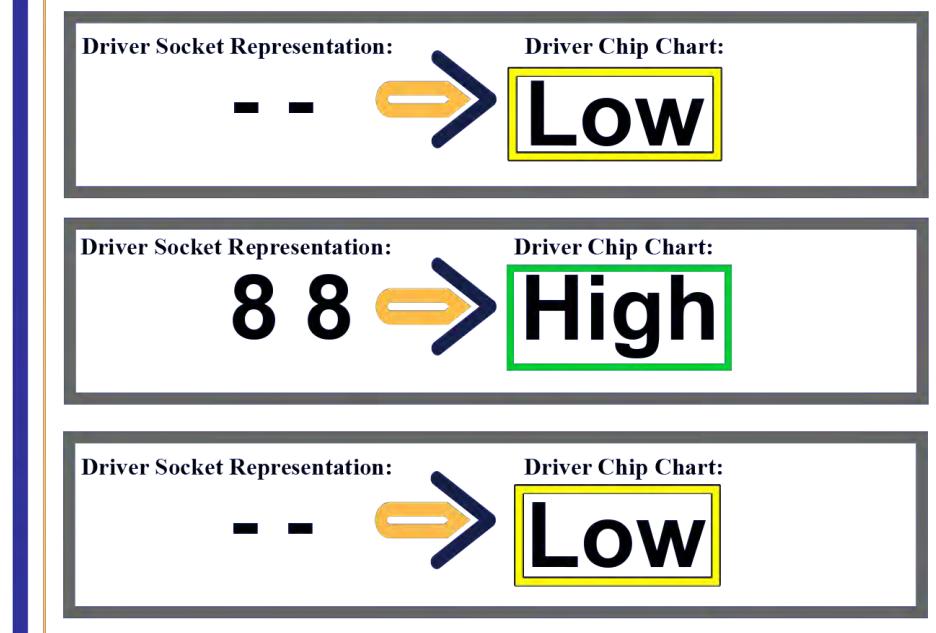


Stand-alone Mode

FYSETC ST820



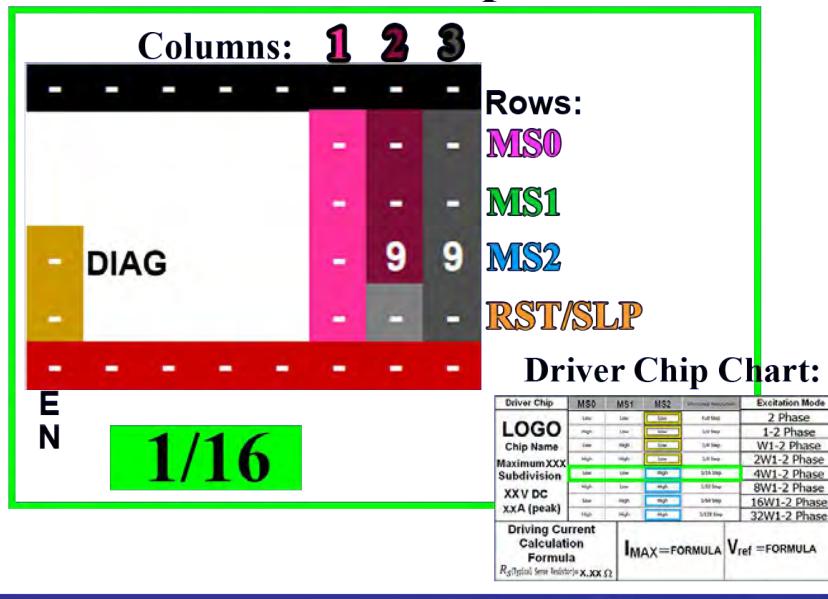
MS1 for Binary State Drivers:



Stand-alone Mode

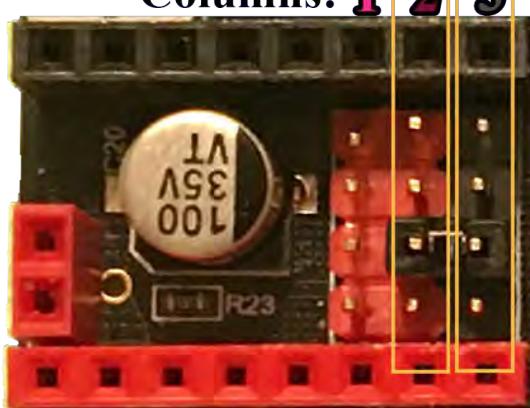
FYSETC ST820

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

High →

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

Low →

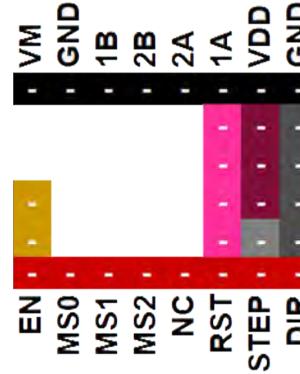
No Jumper Set

Stand-alone Mode

FYSETC ST820

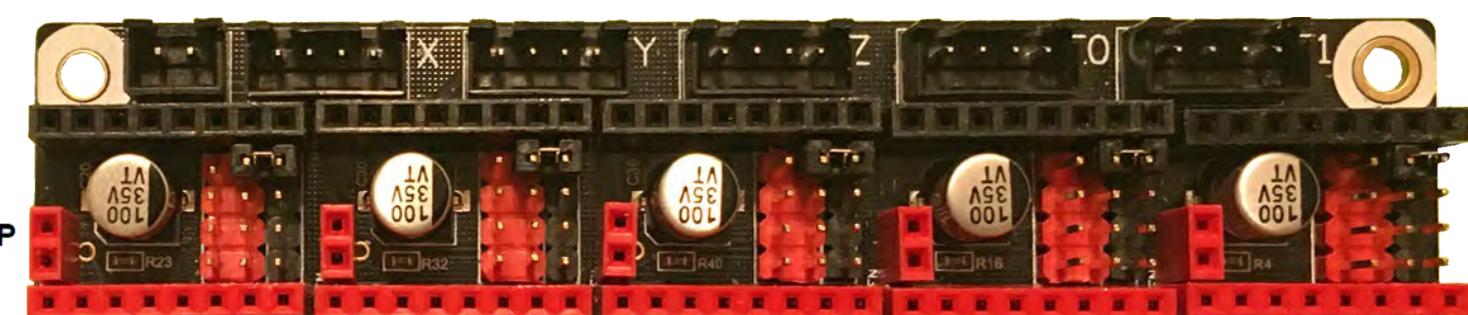
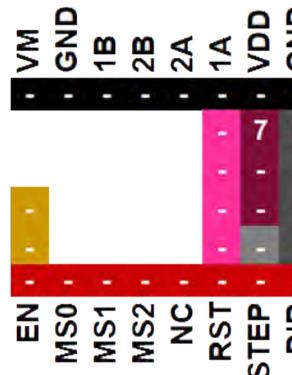
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

STEP



See [Appendix D](#) for legend

1 / 2



See [Appendix D](#) for legend

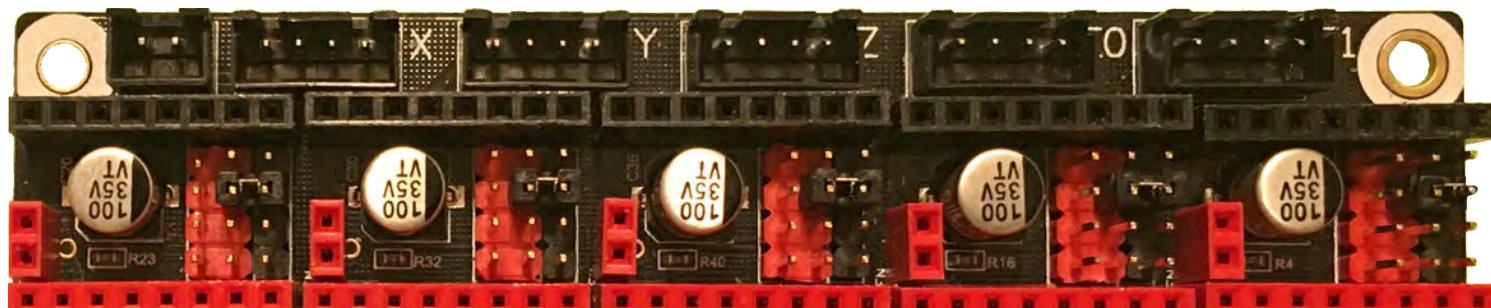
Stand-alone Mode

FYSETC ST820

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

1 / 4

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



See [Appendix D](#) for legend

1 / 8

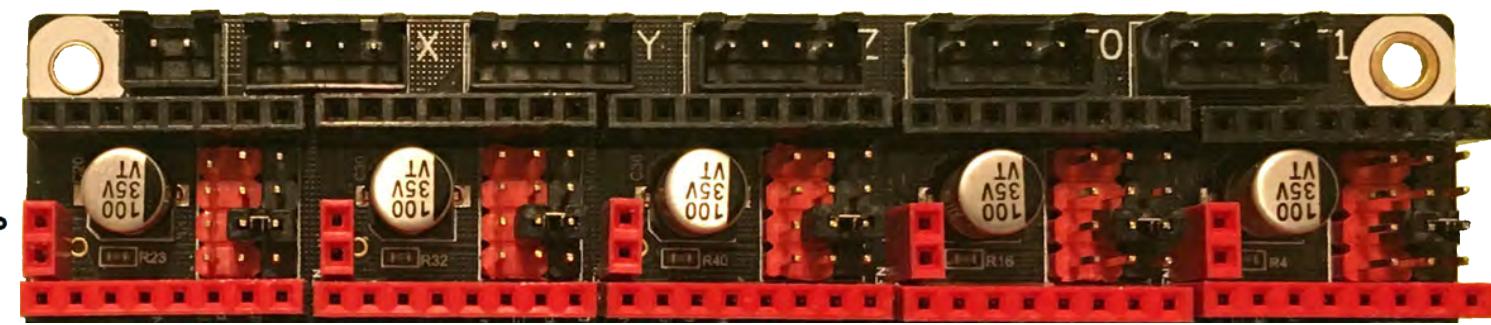
	VM	GND		1B	2B	2A	1A	VDD	GND
	EN	MS0	MS1	MS2	NC	RST	STEP	DIR	
		7	7						



See [Appendix D](#) for legend

1 / 16

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



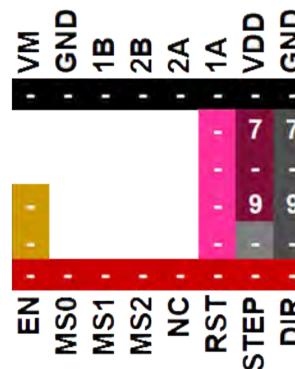
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC ST820

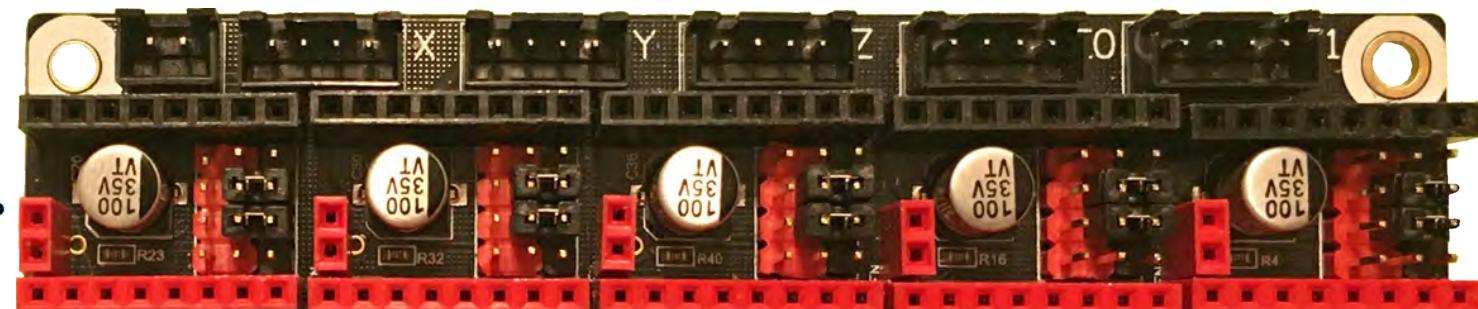
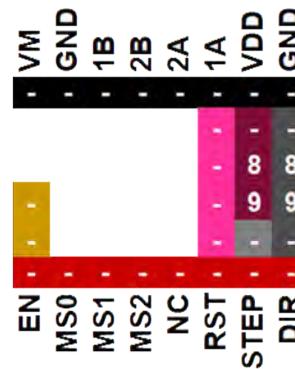
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 32



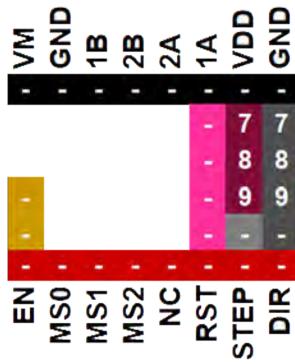
See [Appendix D](#) for legend

1 / 128



See [Appendix D](#) for legend

1 / 256



See [Appendix D](#) for legend

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

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

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

```

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

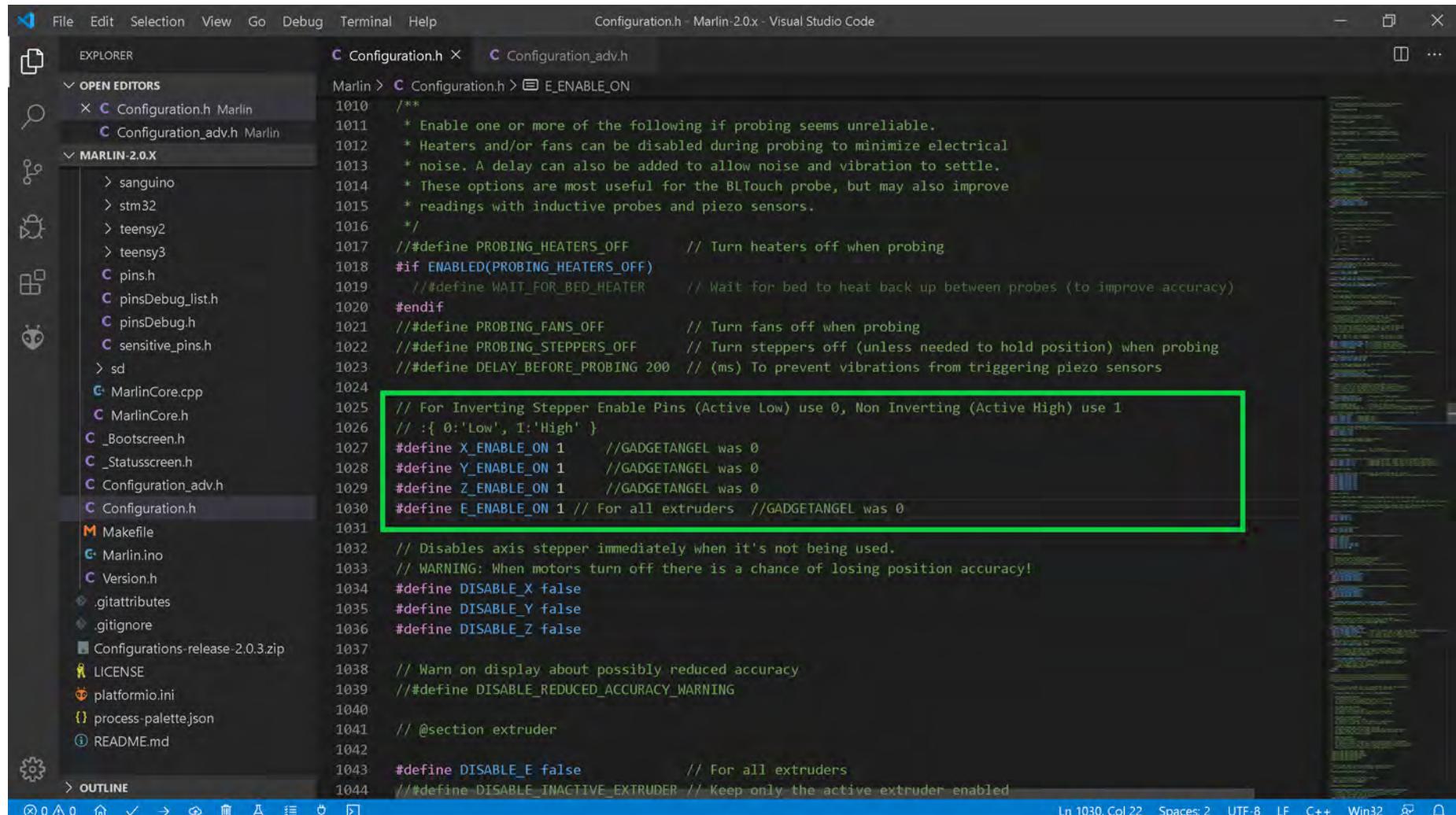
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h ...
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support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

```

- Go to the next page.

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

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



```

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

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E_ENABLE_ON
Marlin > Configuration.h > Configuration.h
Marlin > Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  pins.h
  pinsDebug_list.h
  pinsDebug.h
  sensitive_pins.h
  > sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
Ln 1030, Col 22 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌂

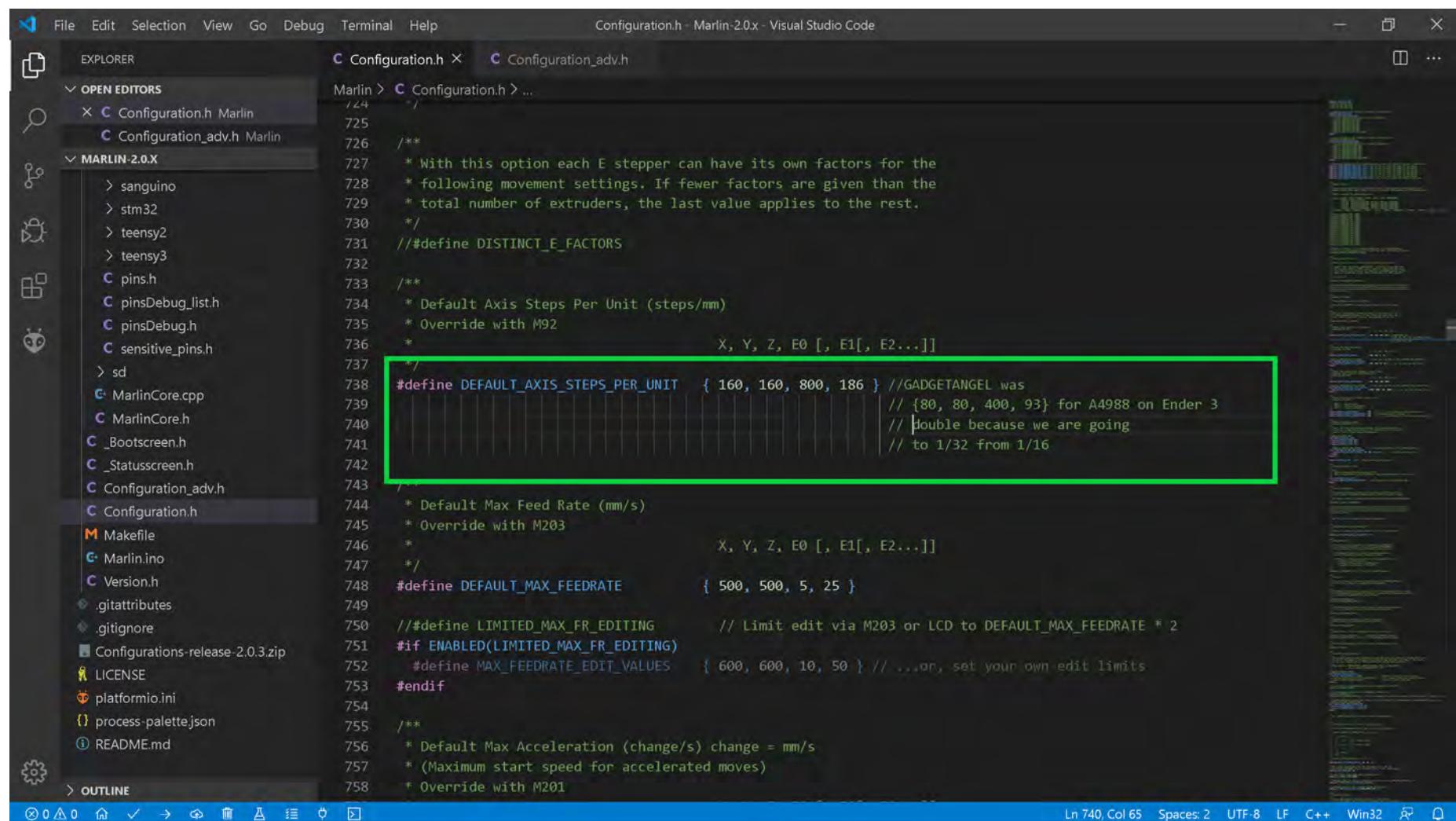
1010 /**
1011 * Enable one or more of the following if probing seems unreliable.
1012 * Heaters and/or fans can be disabled during probing to minimize electrical
1013 * noise. A delay can also be added to allow noise and vibration to settle.
1014 * These options are most useful for the BLTouch probe, but may also improve
1015 * readings with inductive probes and piezo sensors.
1016 */
1017 // #define PROBING_HEATERS_OFF // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019   // #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 // #define PROBING_FANS_OFF // Turn fans off when probing
1022 // #define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1023 // #define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // : { 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1 // GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1 // GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1 // GADGETANGEL was 0
1030 #define E_ENABLE_ON 1 // For all extruders // GADGETANGEL was 0
1031
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled

```

- Go to the next page.

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

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



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

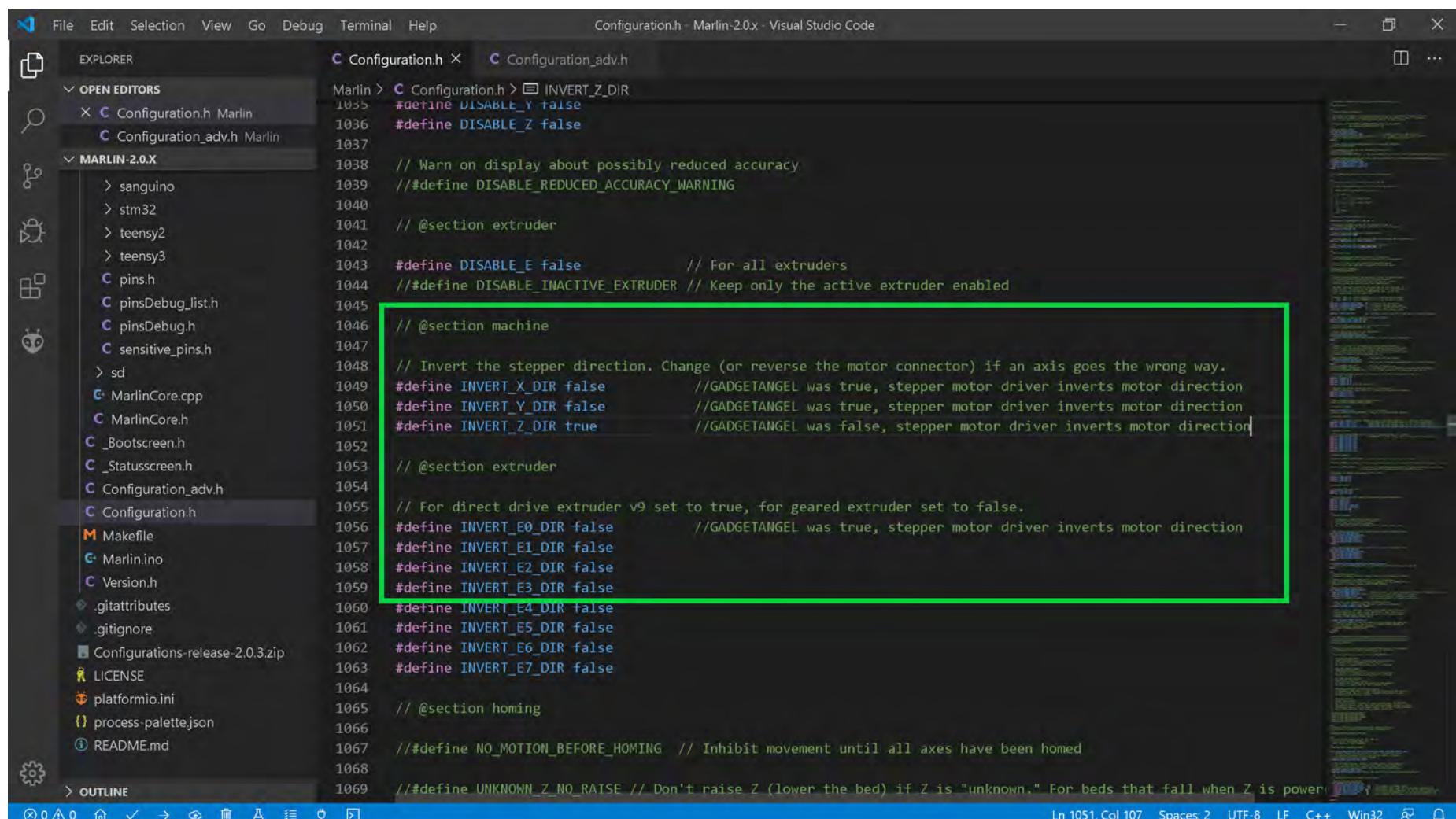
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



The screenshot shows the Visual Studio Code interface with the file 'Configuration.h' open. The code editor displays the Marlin 2.0.x configuration header. A green rectangular box highlights a specific section of the code where the INVERT_Z_DIR macro is defined. The code snippet is as follows:

```

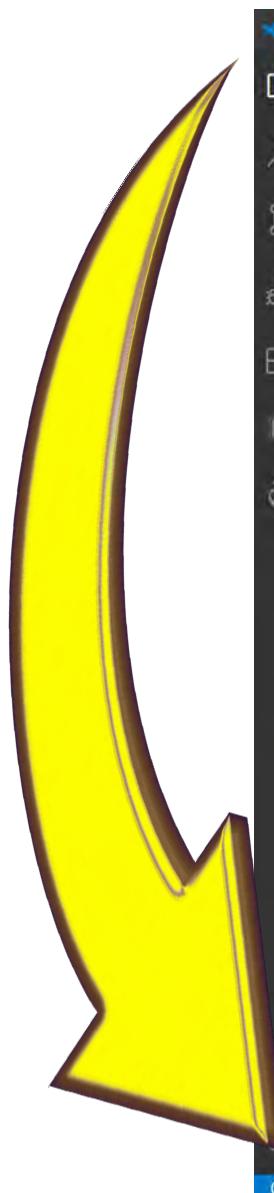
Marlin > C Configuration.h > #define INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X

- samd
- sanguino
- stm32f1
- stm32f4
- stm32f7
- teensy2
- teensy3
- pins.h
- pinsDebug.h
- pinsDebug_list.h
- sensitive_pins.h
- sd
- MarlinCore.cpp
- MarlinCore.h
- _Bootscreen.h
- _Statusscreen.h
- Configuration.h
- Configuration_adv.h
- Makefile
- Marlin.ino
- Version.h
- .editorconfig
- .gitattributes
- .gitignore
- LICENSE
- platformio.ini
- process-palette.json
- README.md

TERMINAL

```
#define BAUDRATE 115200
// Enable the Bluetooth serial interface on AT90USB devices
//#define BLUETOOTH
// Choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif
// Name displayed in the LCD "Ready" message and Info menu
=====
[SUCCESS] Took 130.61 seconds =====
```

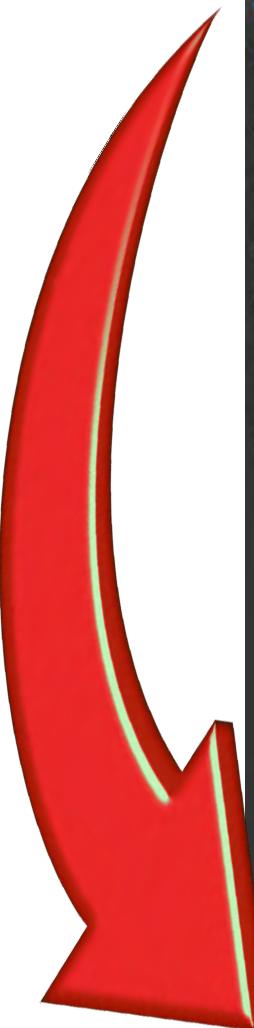
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	DUE	
DUE	IGNORED	
DUE_USB	IGNORED	
PCB_DALY	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

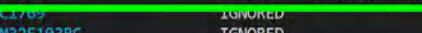
OUTLINE TIMELINE

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

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

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

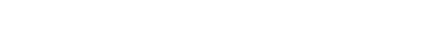




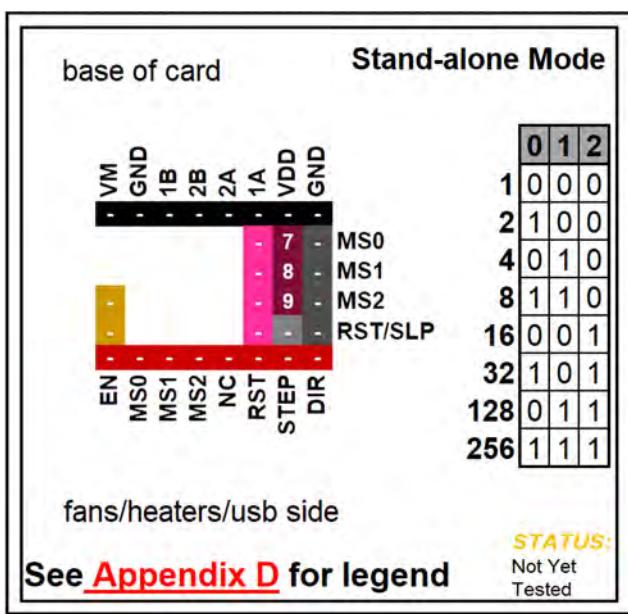









<img alt="A green rectangular box highlights the 'LPC1768' entry in the 'Environment

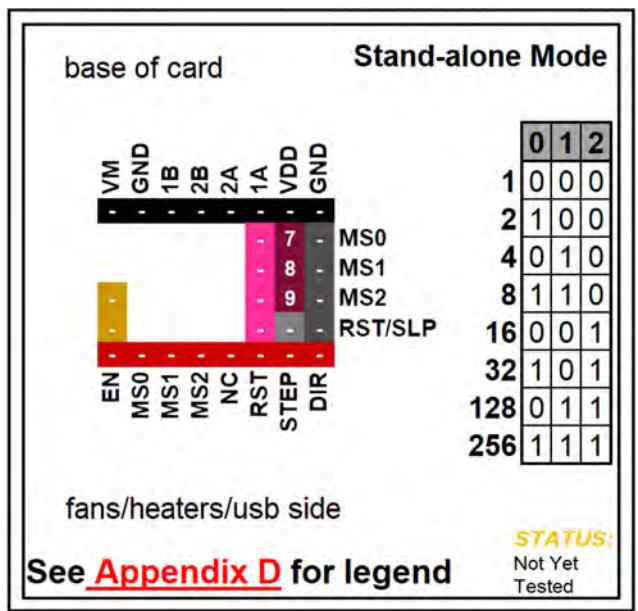


BIQU ST820

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

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

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



BIQU ST820

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



Low ➡ **No Jumper**



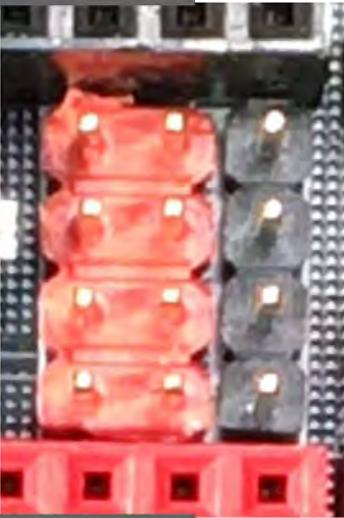
Rows:

MS0

MS1

MS2

RST/SLP



High ➡ **Jumper Set**

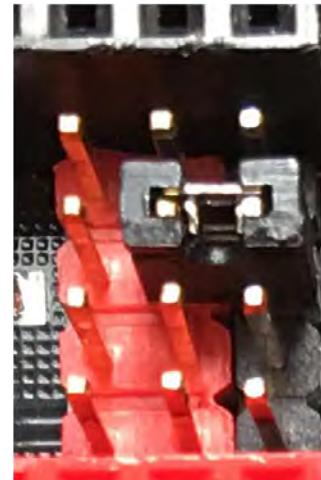
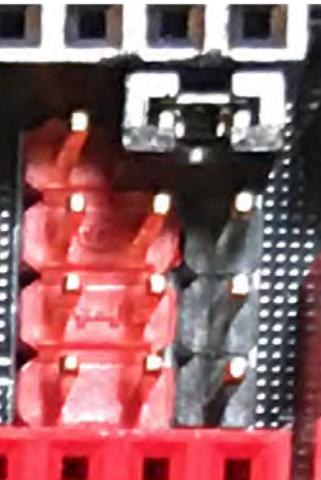
Rows:

MS0

MS1

MS2

RST/SLP



MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

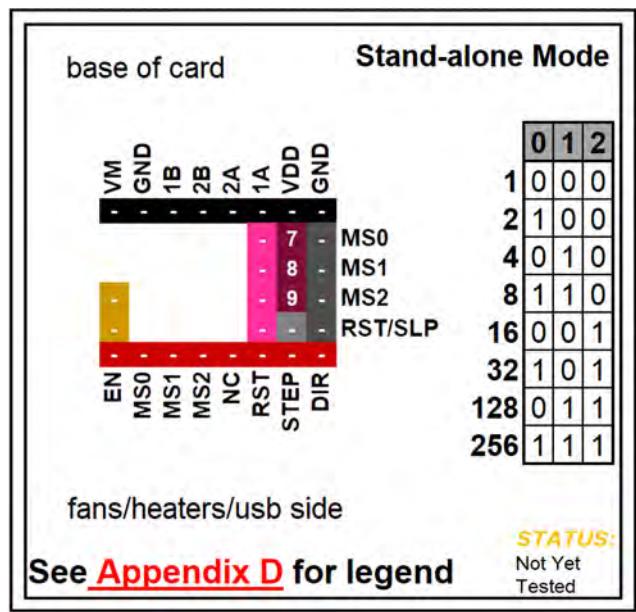
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XXV DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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

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



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





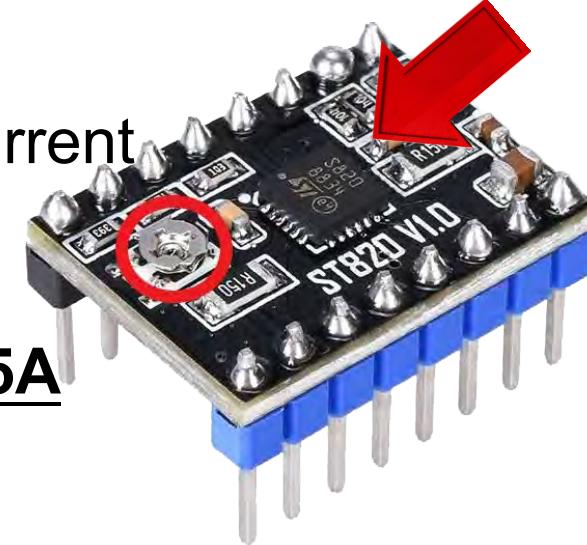
Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



BIQU ST820

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

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



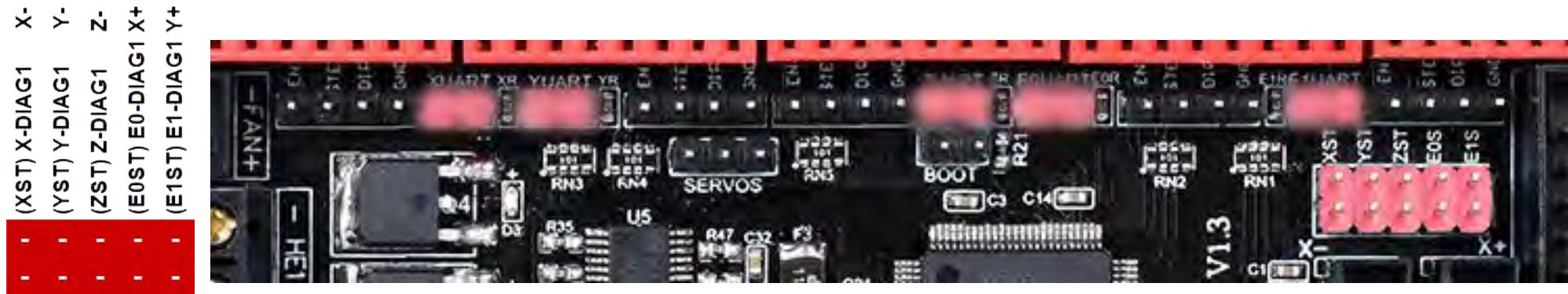
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

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

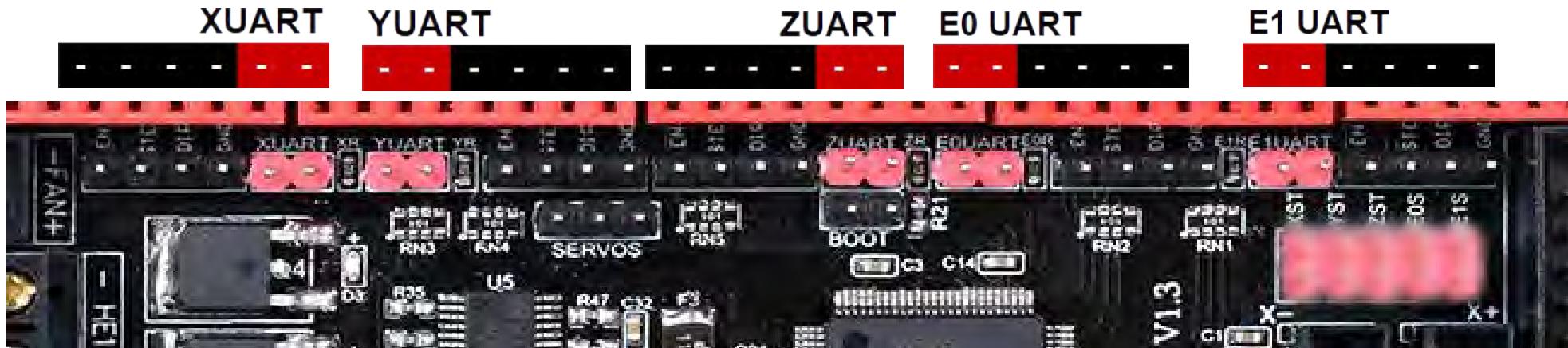
Stand-alone Mode

BIQU ST820

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



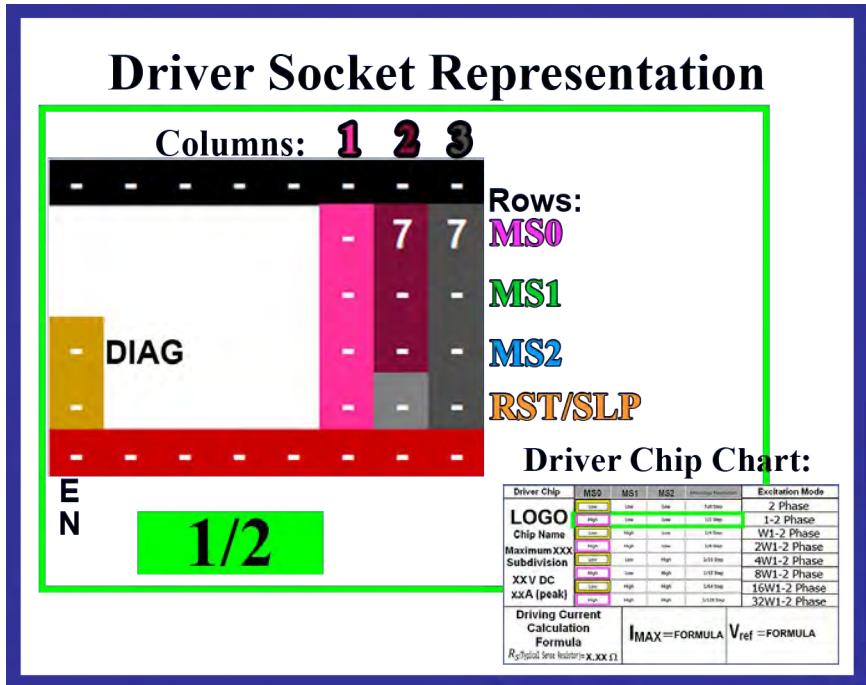
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



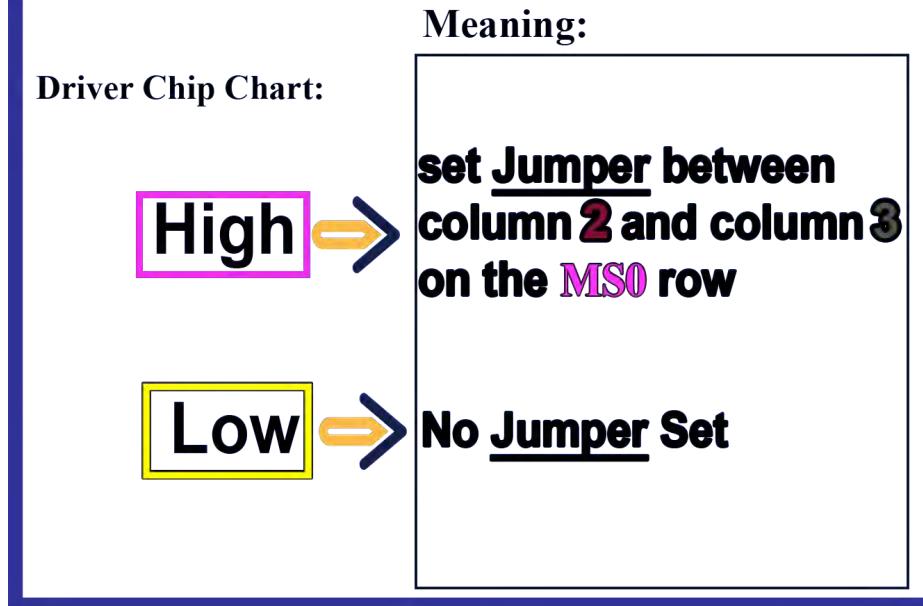
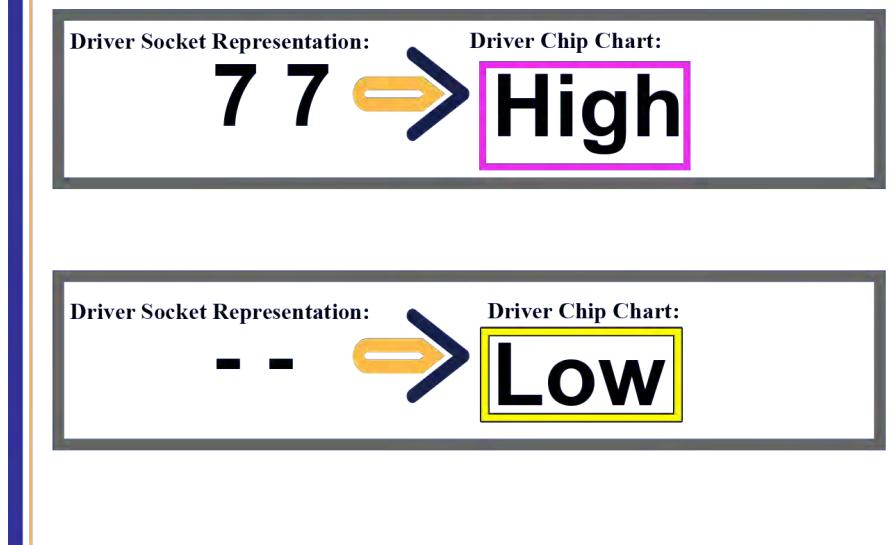
Stand-alone Mode

BIQU ST820

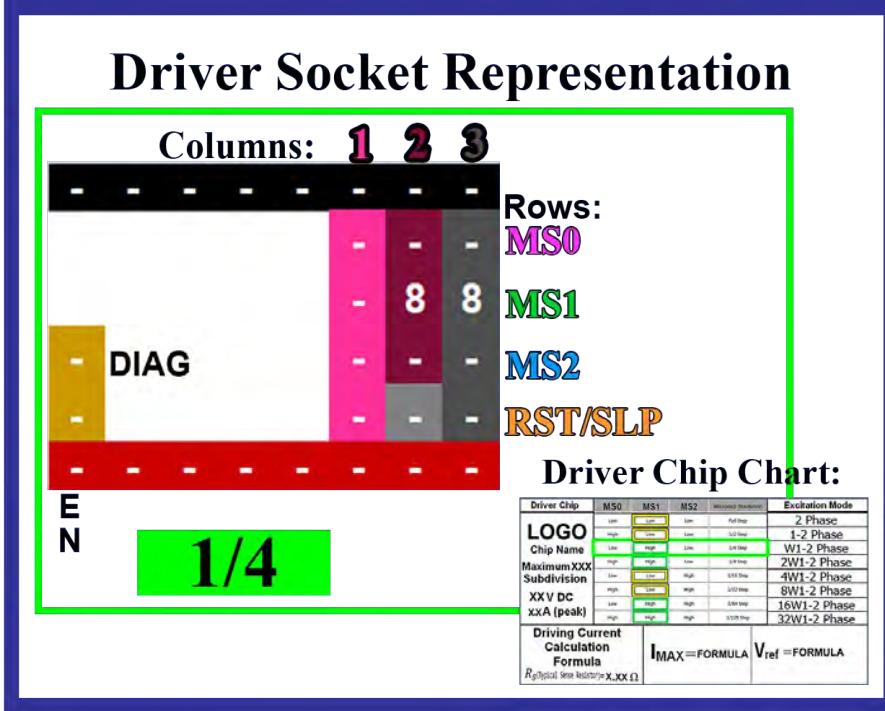
SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers



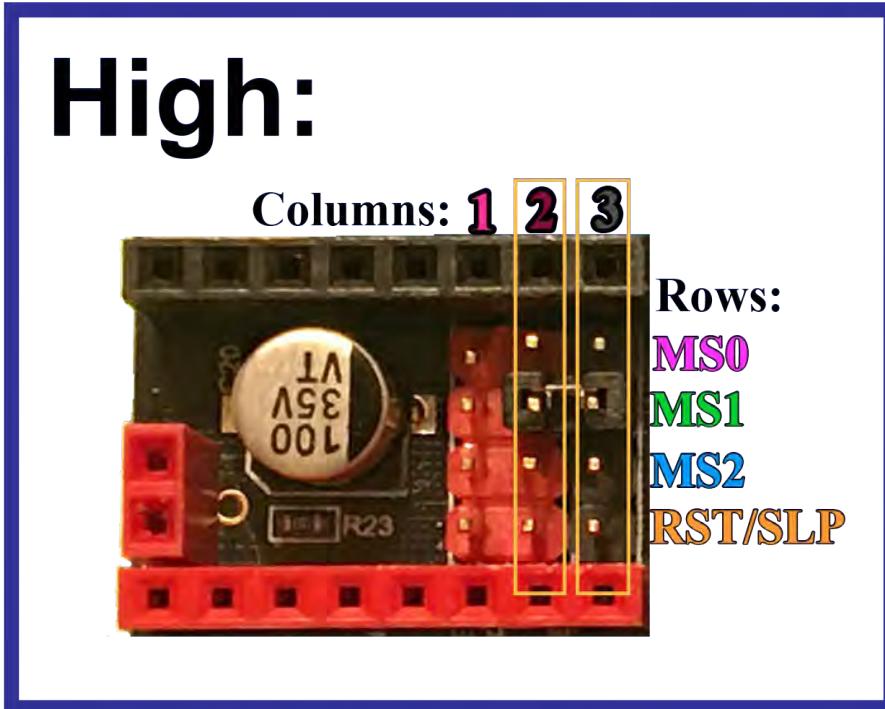
MS0 for Binary State Drivers:



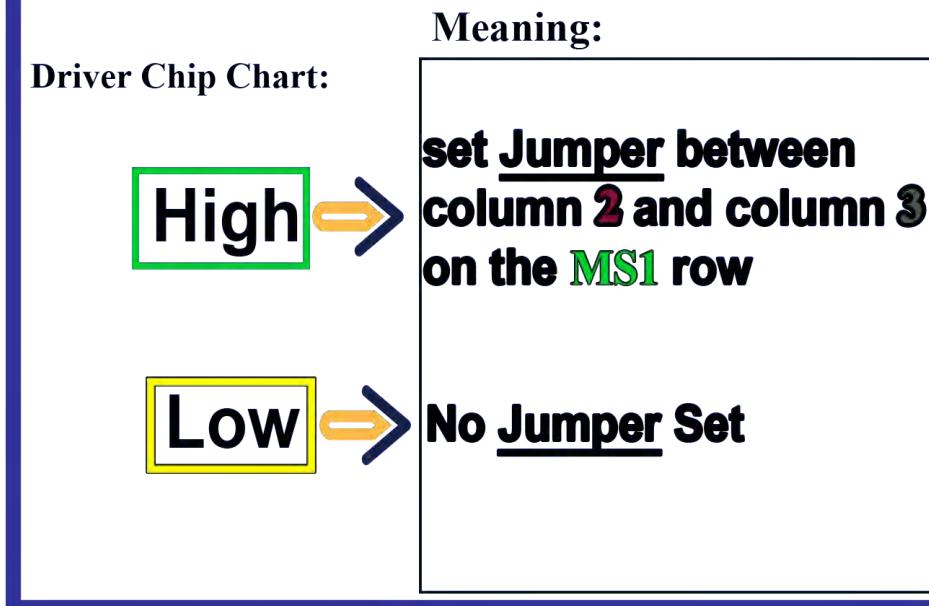
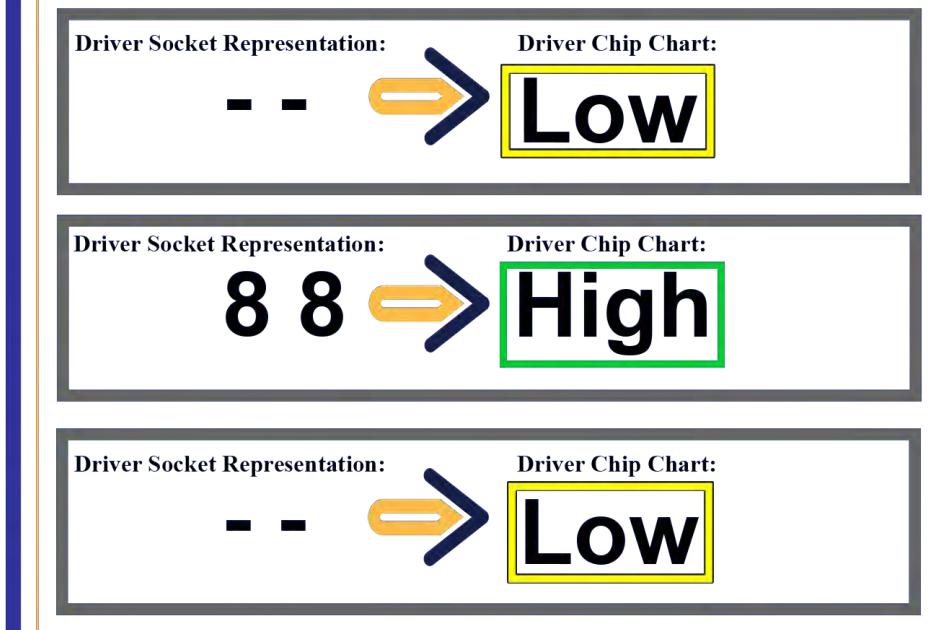
Stand-alone Mode



BIQU ST820



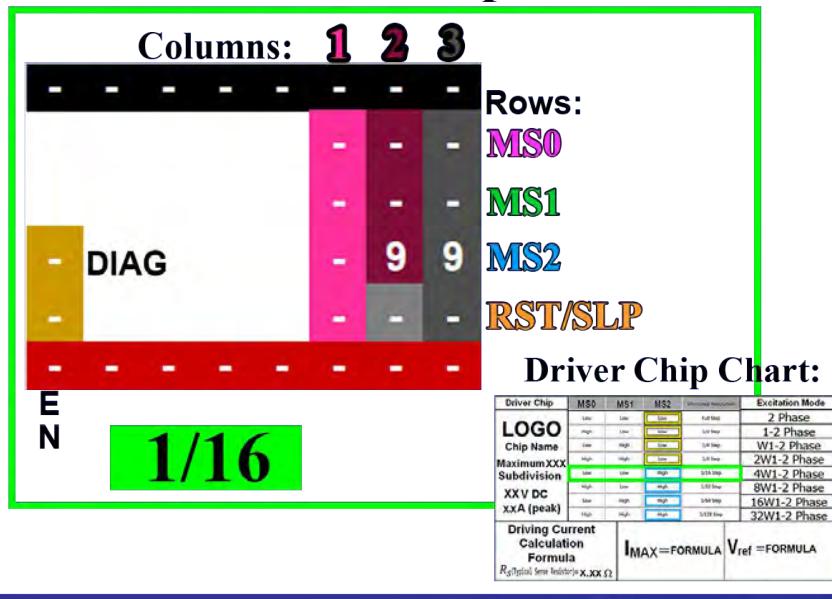
MS1 for Binary State Drivers:



Stand-alone Mode

BIQU ST820

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:
Driver Chip Chart:

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

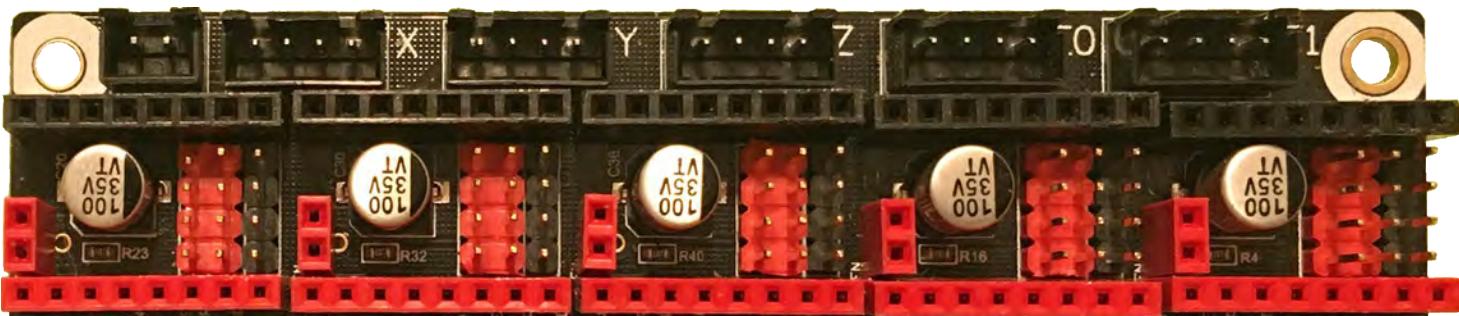
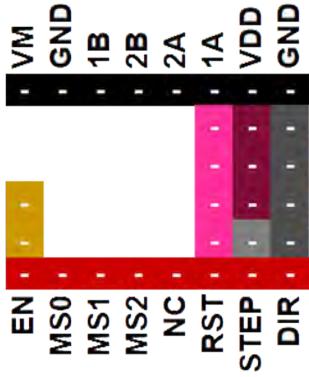
Low → **No Jumper Set**

Stand-alone Mode

BIQU ST820

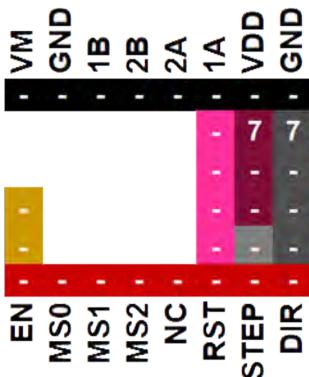
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

STEP



See [Appendix D](#) for legend

1 / 2



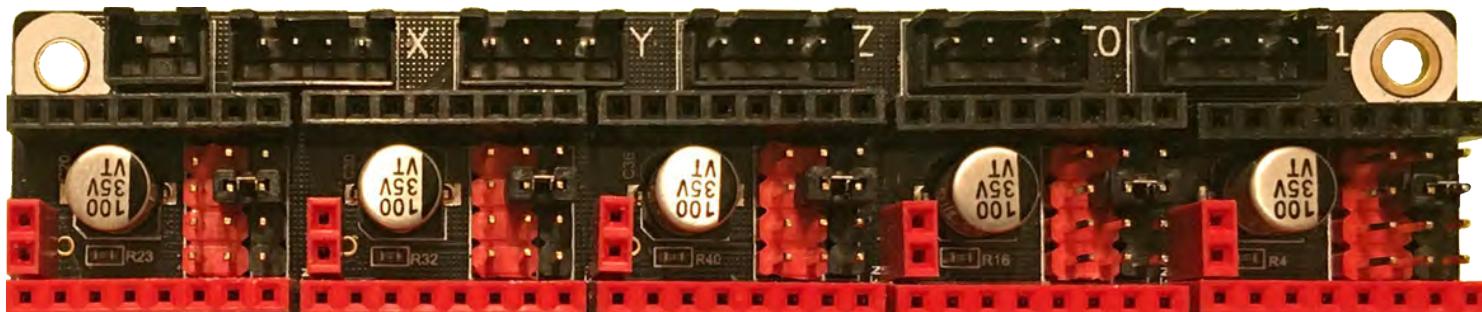
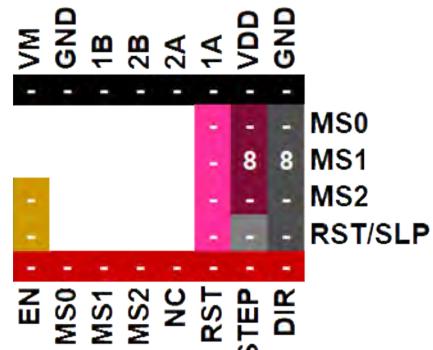
See [Appendix D](#) for legend

Stand-alone Mode

BIQU ST820

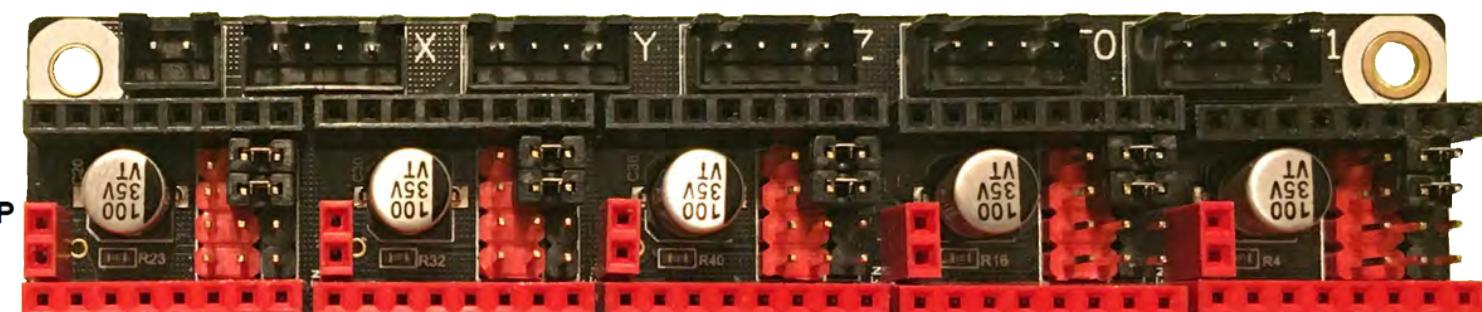
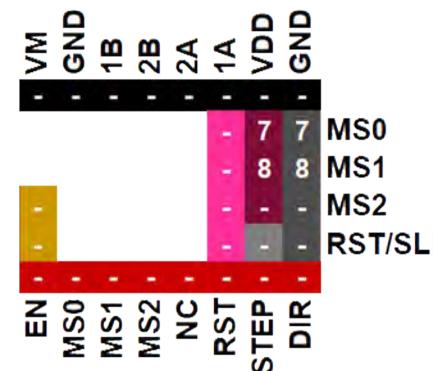
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 4



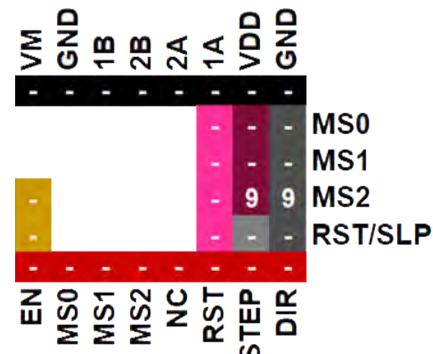
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



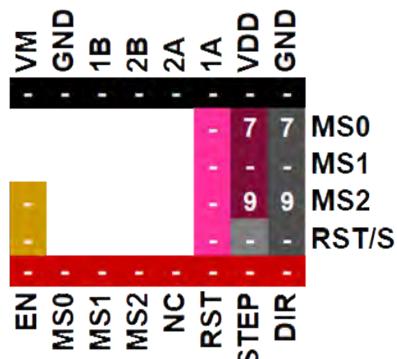
See [Appendix D](#) for legend

Stand-alone Mode

BIQU ST820

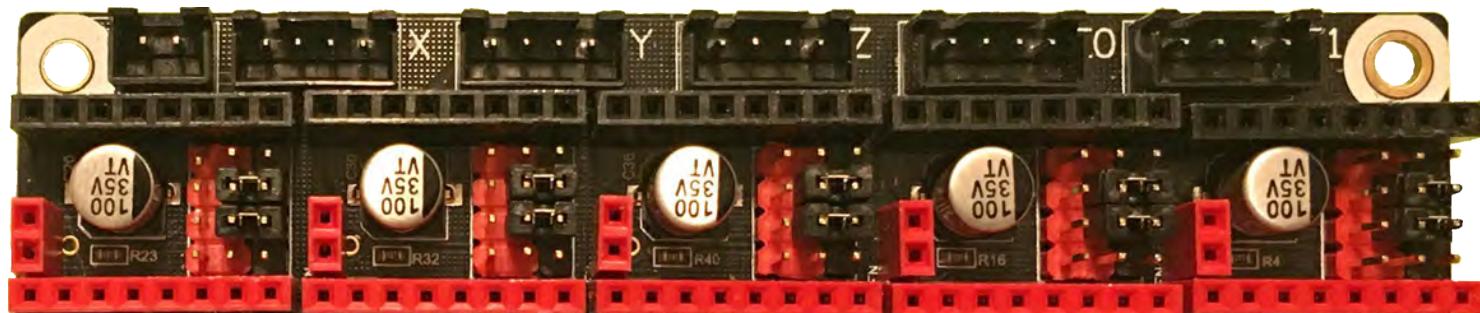
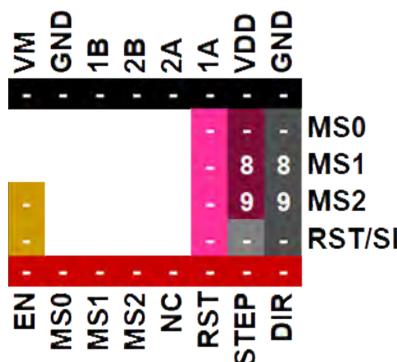
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 32



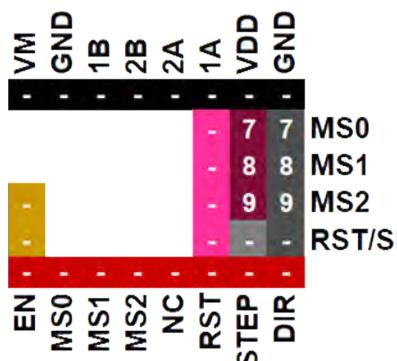
See [Appendix D](#) for legend

1 / 128



See [Appendix D](#) for legend

1 / 256

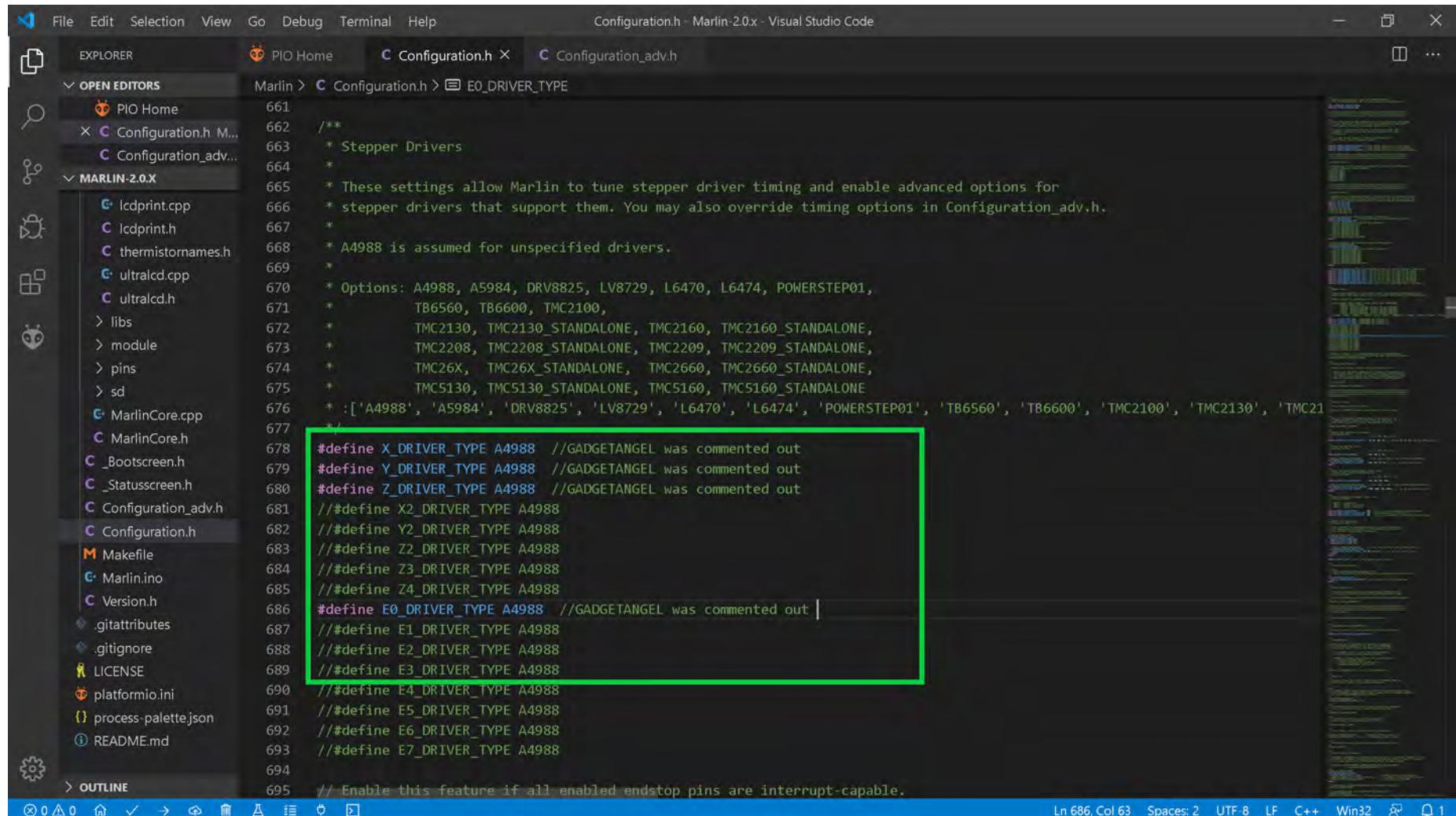


See [Appendix D](#) for legend

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

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

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



```

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

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

```

- Go to the next page.

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

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

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

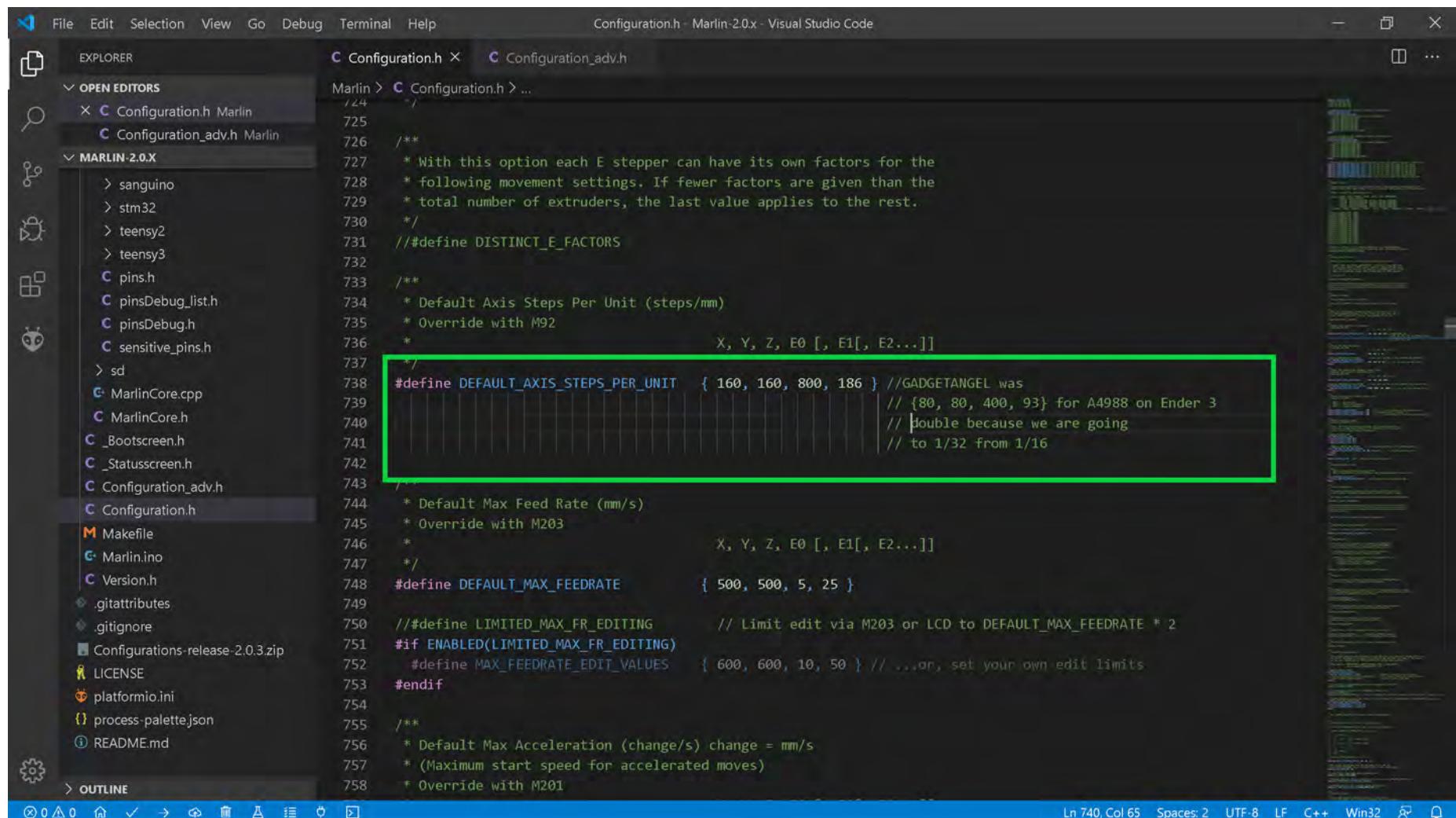
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
  > sanguino
  > stm32
  > teensy2
  > teensy3
  C pins.h
  C pinsDebug_list.h
  C pinsDebug.h
  C sensitive_pins.h
  > sd
  C MarlinCore.cpp
  C MarlinCore.h
  C _Bootscreen.h
  C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

Marlin > Configuration.h > E_ENABLE_ON
1010 /**
1011 * Enable one or more of the following if probing seems unreliable.
1012 * Heaters and/or fans can be disabled during probing to minimize electrical
1013 * noise. A delay can also be added to allow noise and vibration to settle.
1014 * These options are most useful for the BLTouch probe, but may also improve
1015 * readings with inductive probes and piezo sensors.
1016 */
1017 // #define PROBING_HEATERS_OFF // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019 // #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 // #define PROBING_FANS_OFF // Turn fans off when probing
1022 // #define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1023 // #define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // :{ 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1 // GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1 // GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1 // GADGETANGEL was 0
1030 #define E_ENABLE_ON 1 // For all extruders // GADGETANGEL was 0
1031
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
```

- Go to the next page.

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

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



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

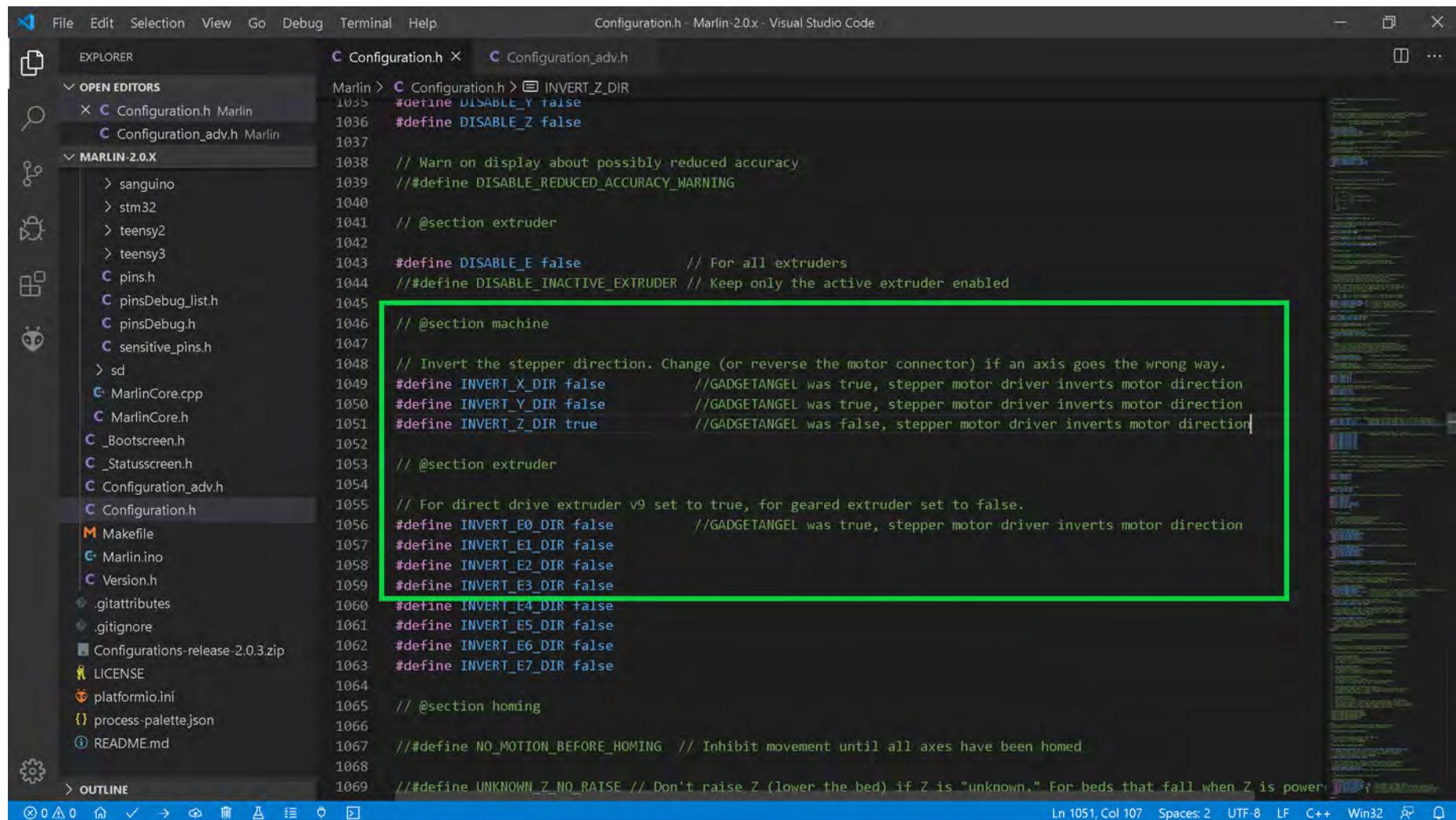
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Marlin > Configuration.h > Configuration_adv.h
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

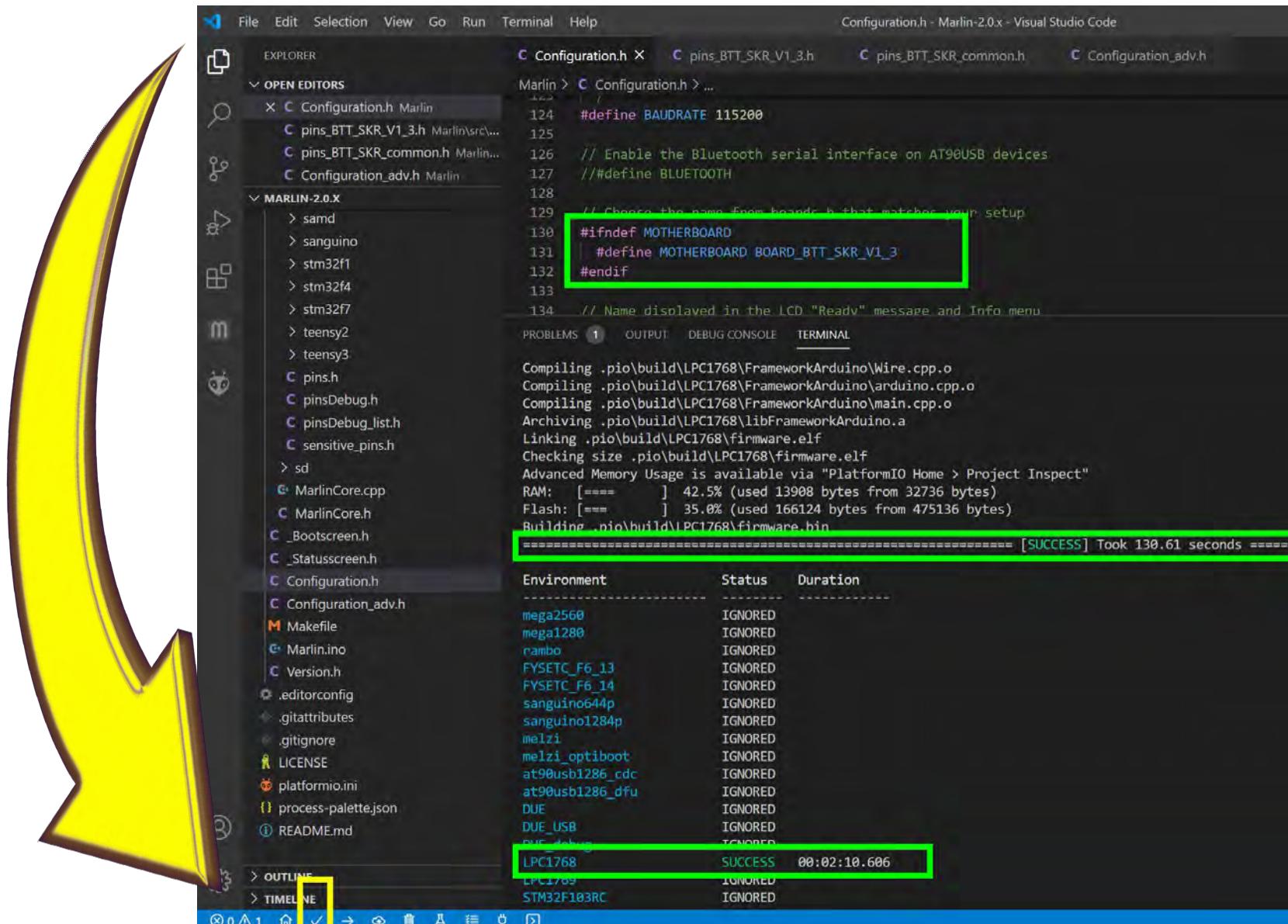
```

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

- Go to the next page.

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

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



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

EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132#endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
===== [SUCCESS] Took 130.61 seconds =====

```

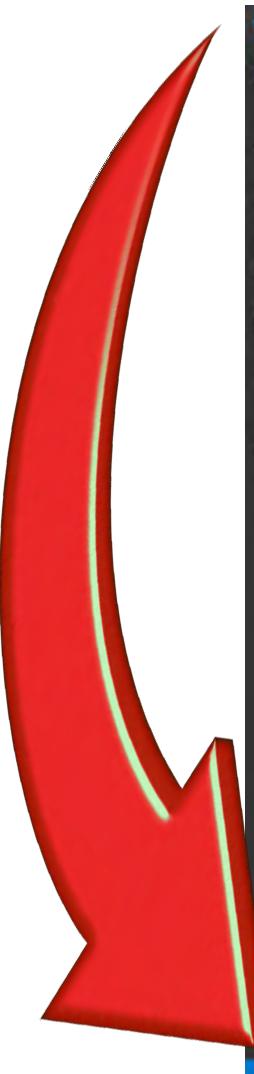
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	DUE	
DUE	IGNORED	
DUE_USB	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

OUTLINE TIMELINE

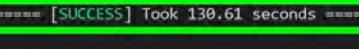
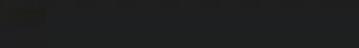
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

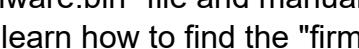
- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

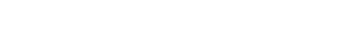


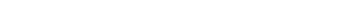




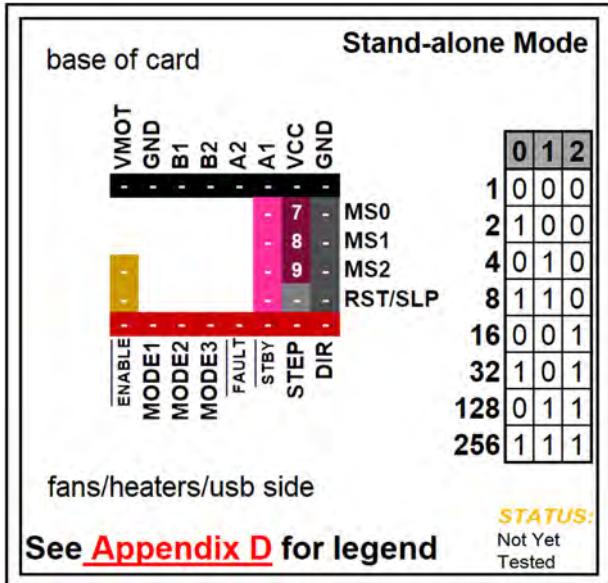




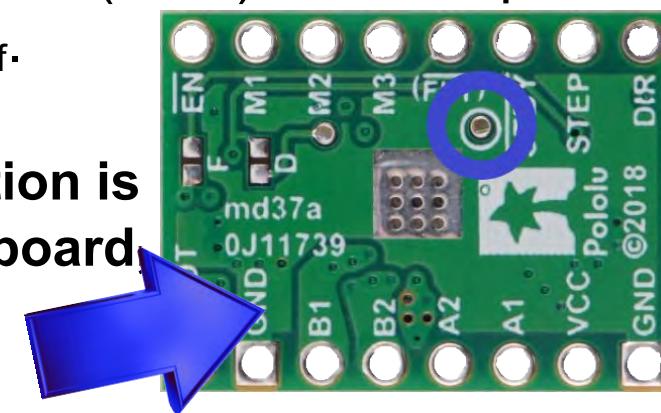
<img alt="A green rectangular box highlights the 'LPC1768' entry in the 'Environment' table in the

POLOLU ST820 (STSPIN820)



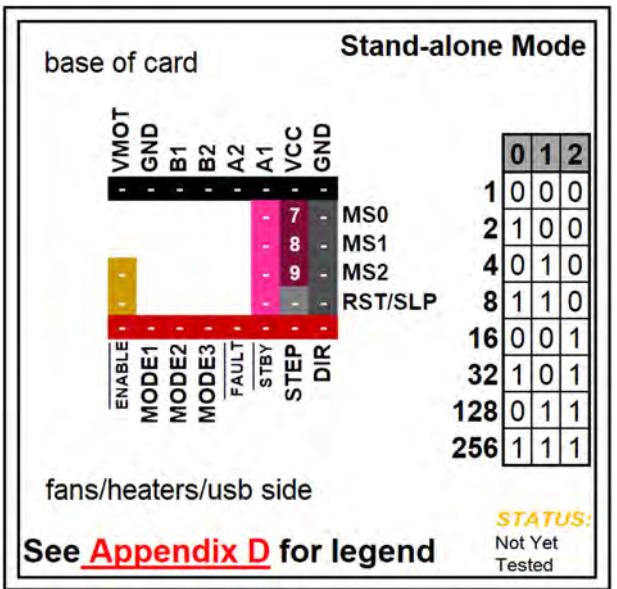
NOTE: Use the potentiometer (POT) on the top of the board to adjust your V_{ref} .

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



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

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



Driver Chip Chart:

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

Driving Current Calculation Formula

$I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

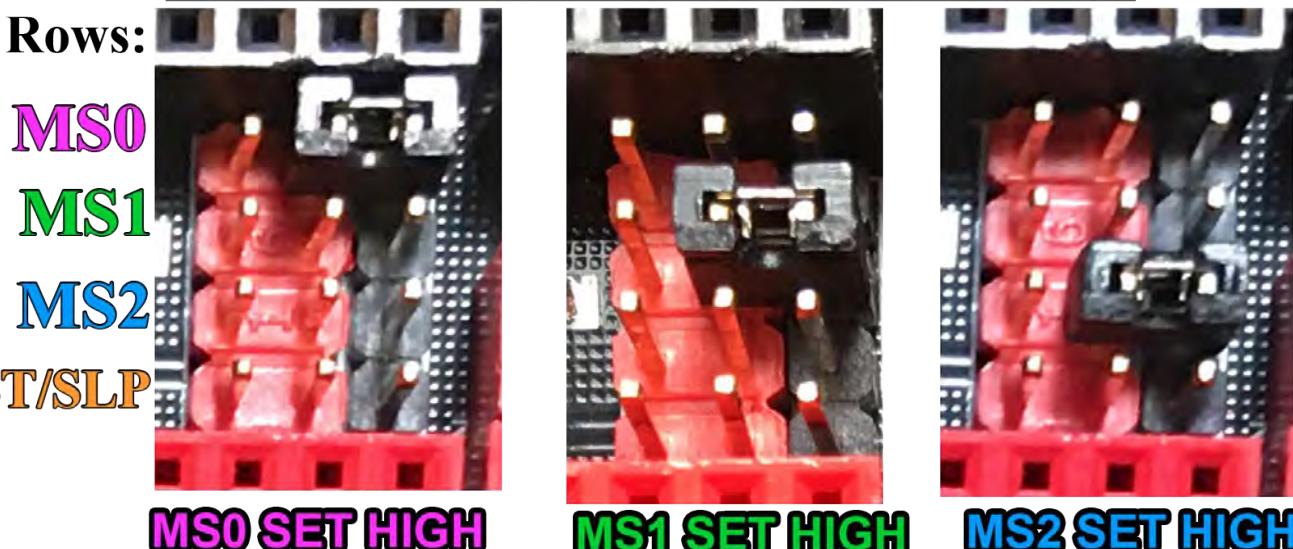
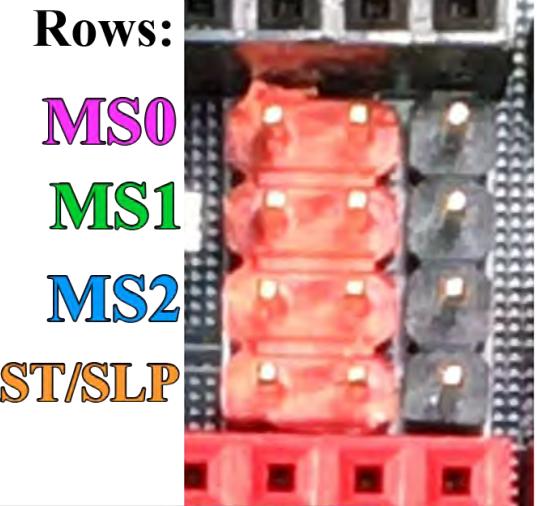
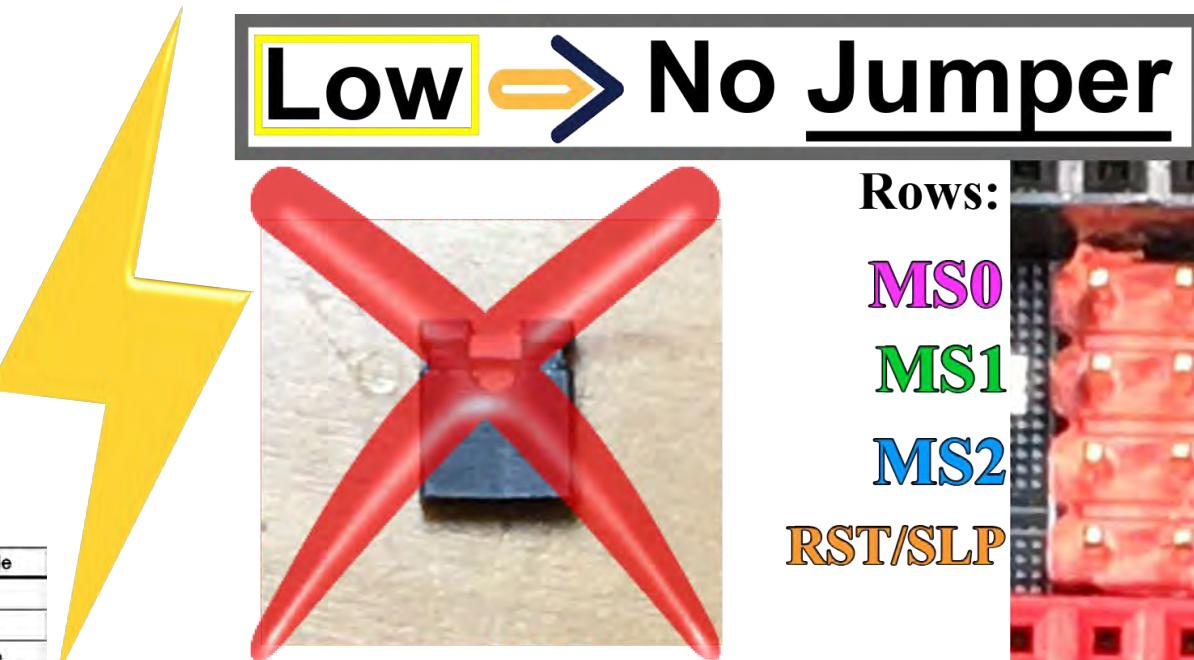
$R_S(\text{Typical Sense Resistor}) = X.XX \Omega$

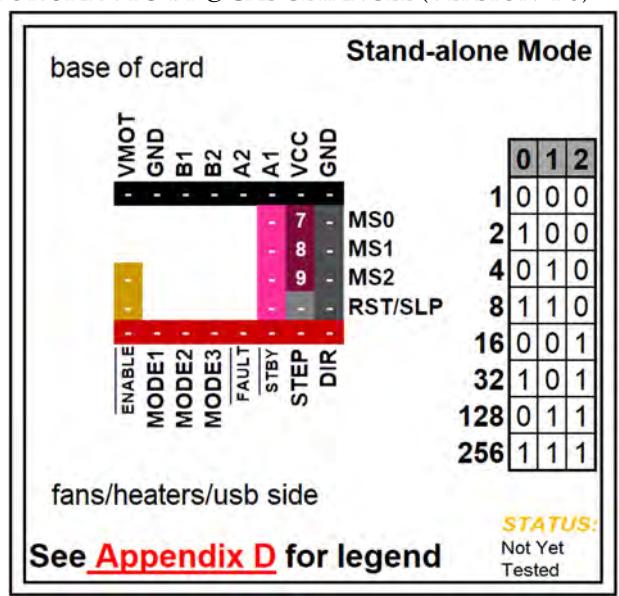
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

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

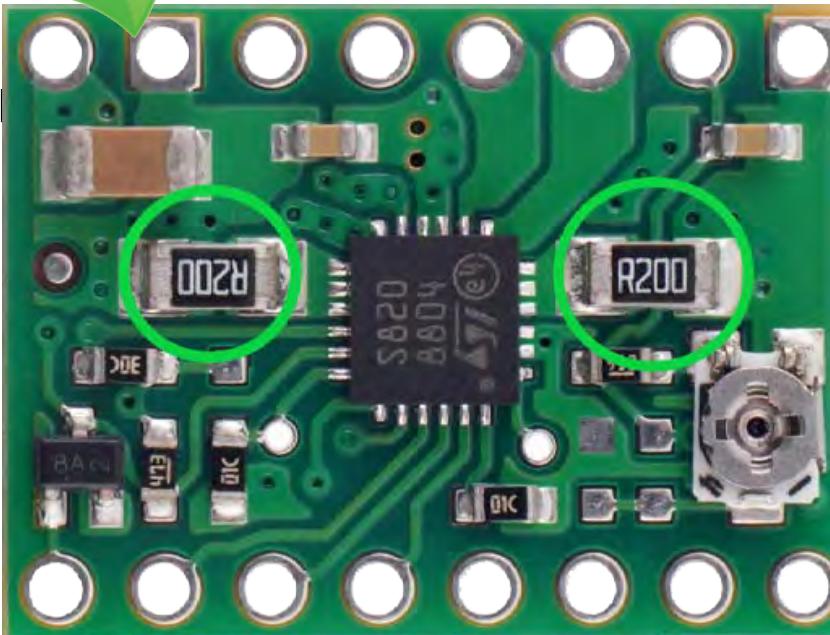
POLOLU ST820 (STSPIN820)

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





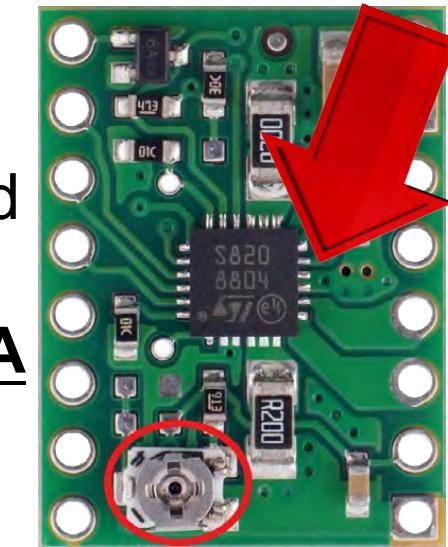
Note: Check your current sense resistors (R_s) values on the driver board, as shown in GREEN



POLOLU ST820 (STSPIN820)

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

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



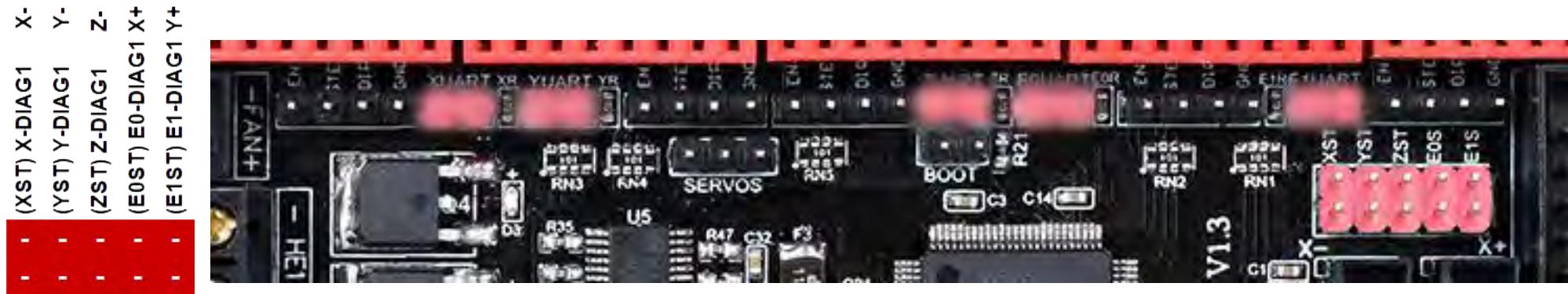
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

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

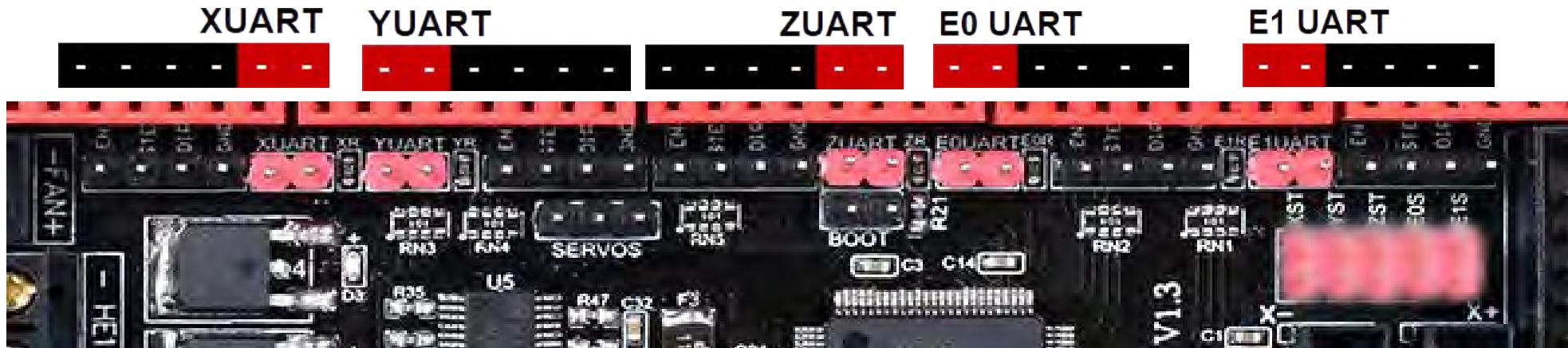
Stand-alone Mode

POLOLU ST820 (STSPIN820)

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



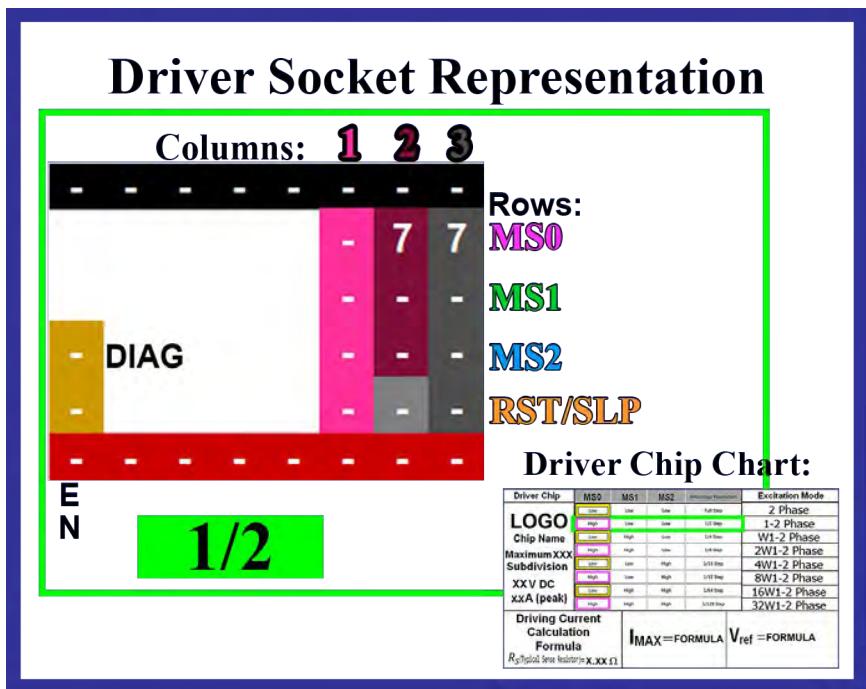
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



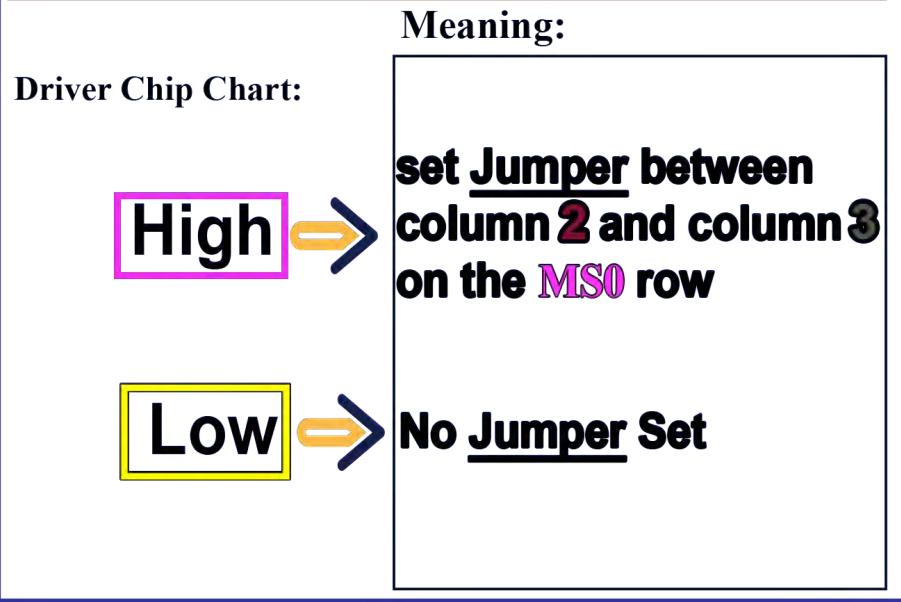
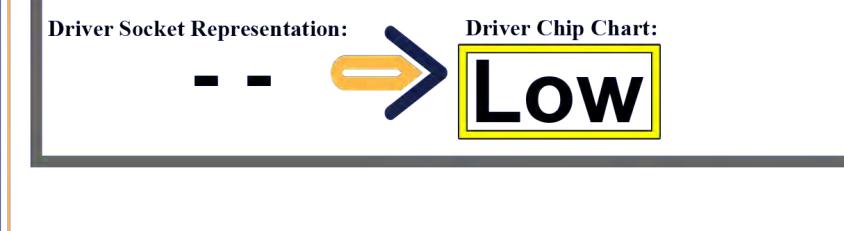
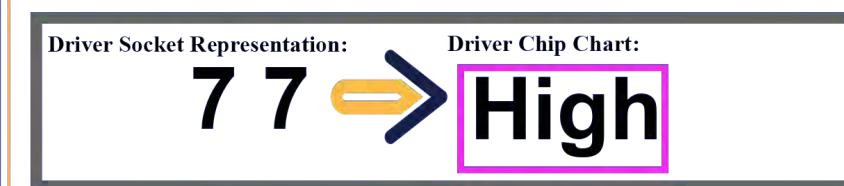
Stand-alone Mode

POLOLU ST820 (STSPIN820)

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



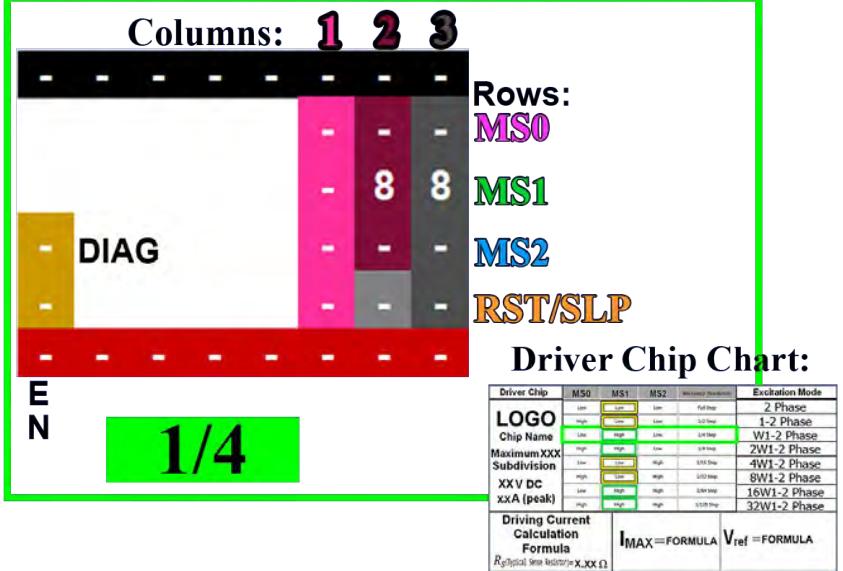
MS0 for Binary State Drivers:



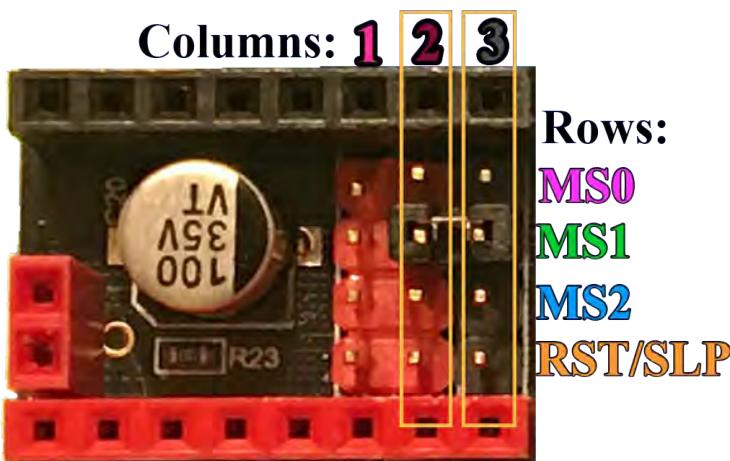
Stand-alone Mode

POLOLU ST820 (STSPIN820)

Driver Socket Representation



High:



MS1 for Binary State Drivers:

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

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

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

Meaning:

Driver Chip Chart:

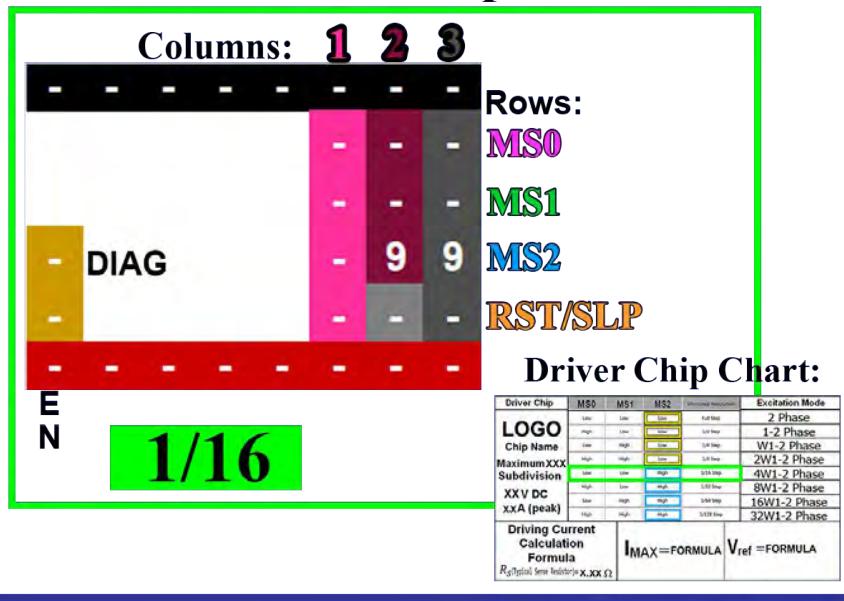
High → set Jumper between column 2 and column 3 on the MS1 row

Low → No Jumper Set

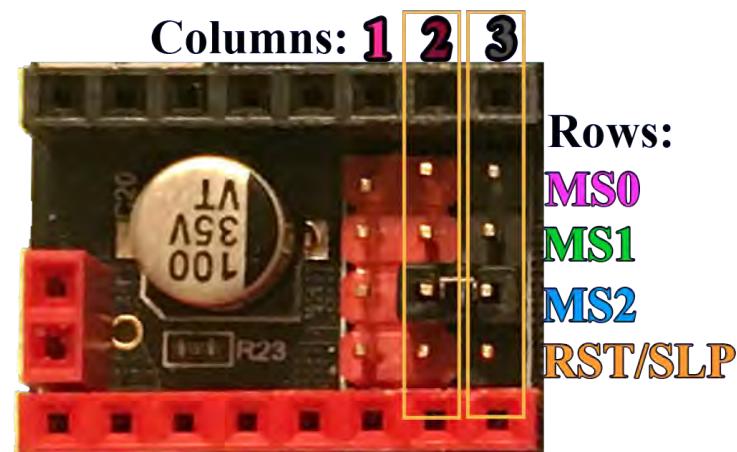
Stand-alone Mode

POLOLU ST820 (STSPIN820)

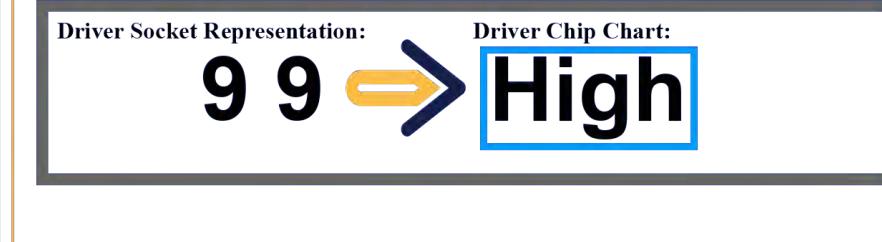
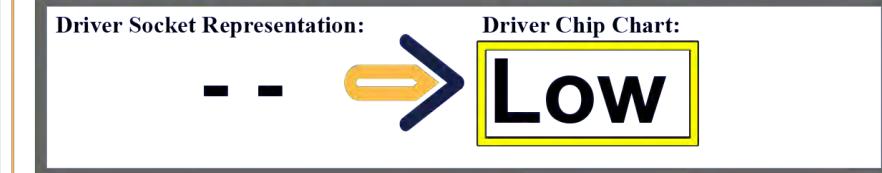
Driver Socket Representation



High:



MS2 for Binary State Drivers:



Meaning:

Driver Chip Chart:

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

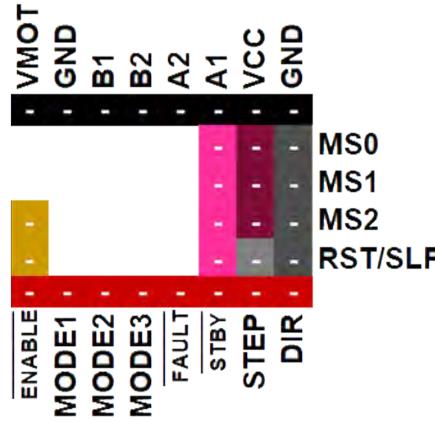
Low → **No Jumper Set**

Stand-alone Mode

POLOLU ST820 (STSPIN820)

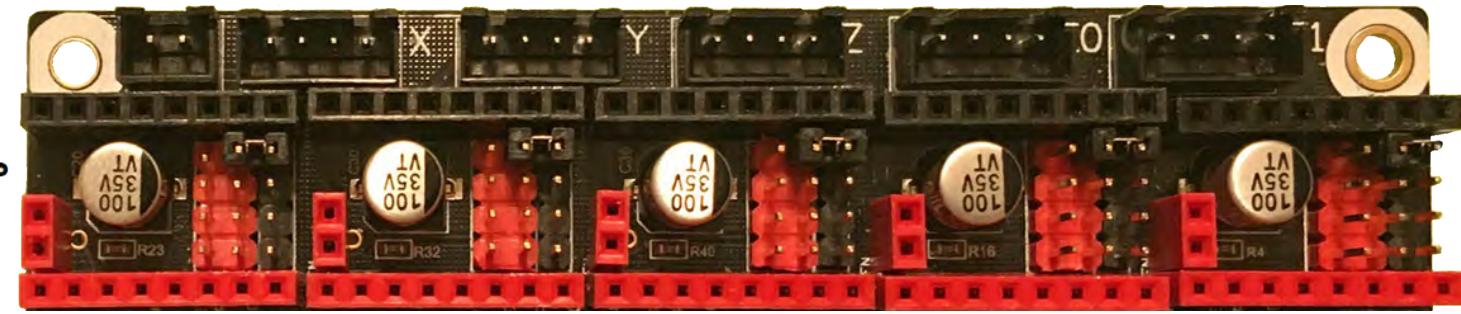
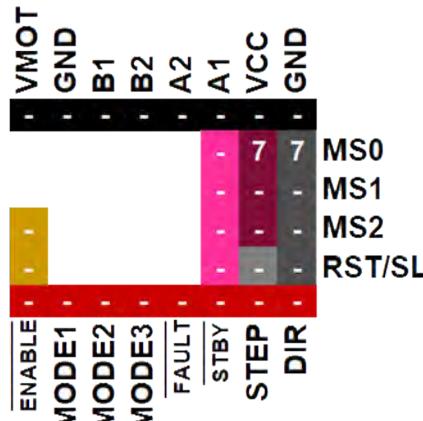
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

STEP



See [Appendix D](#) for legend

1 / 2



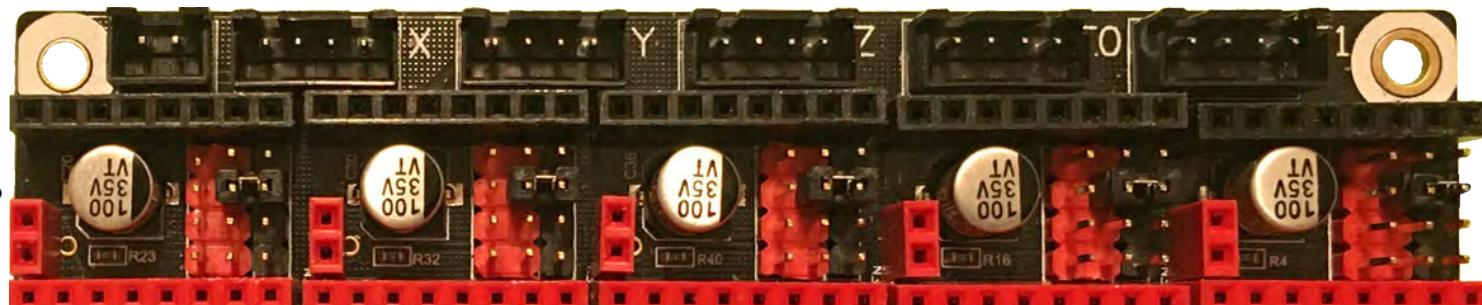
See [Appendix D](#) for legend

Stand-alone Mode

POLOLU ST820 (STSPIN820)

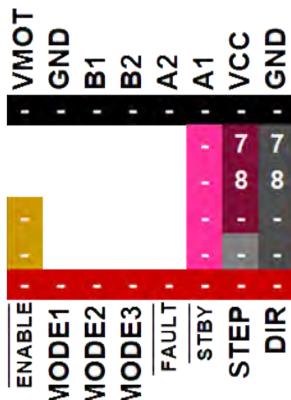
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 4



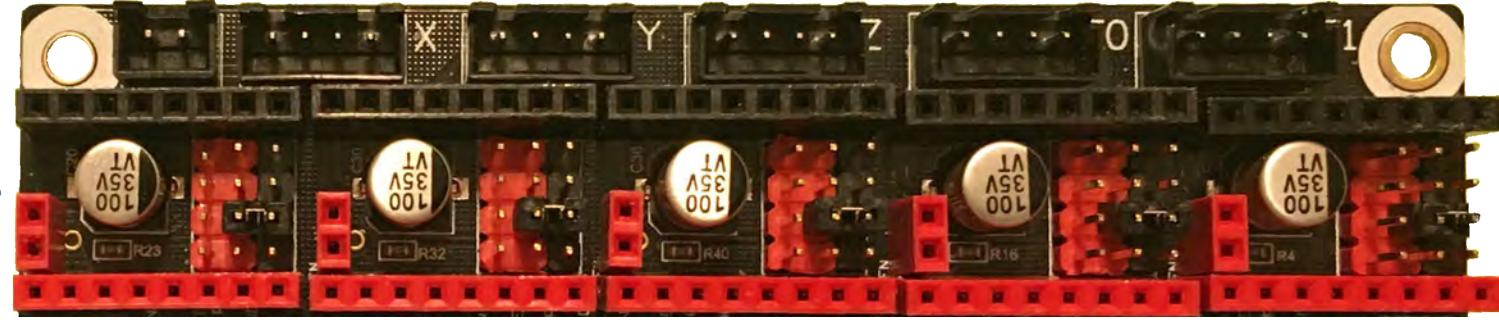
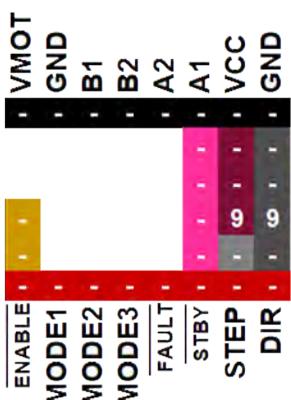
See [Appendix D](#) for legend

1 / 8



See [Appendix D](#) for legend

1 / 16



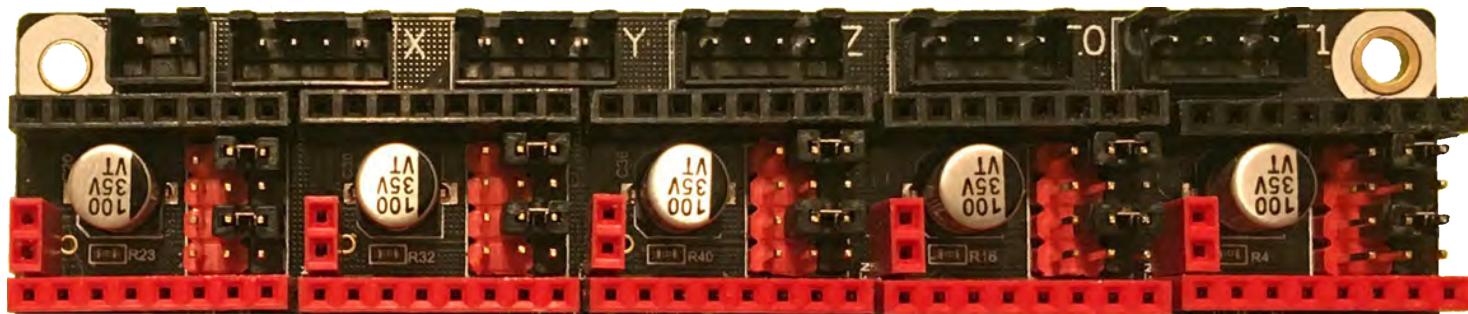
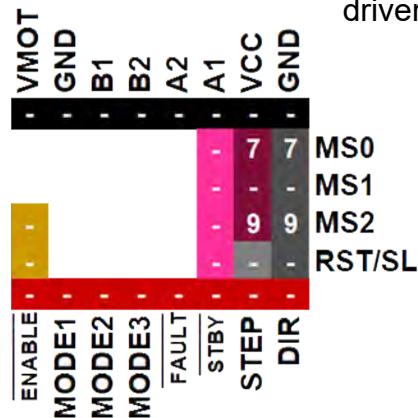
See [Appendix D](#) for legend

Stand-alone Mode

POLOLU ST820 (STSPIN820)

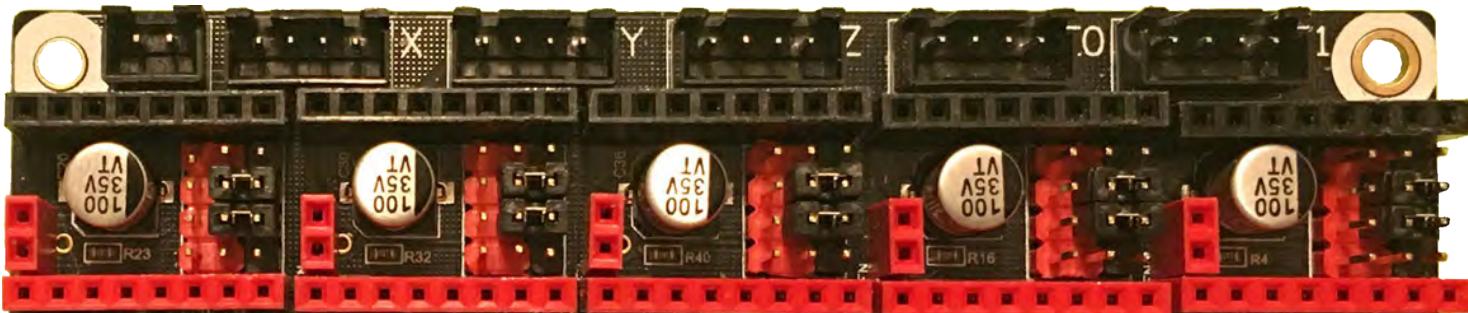
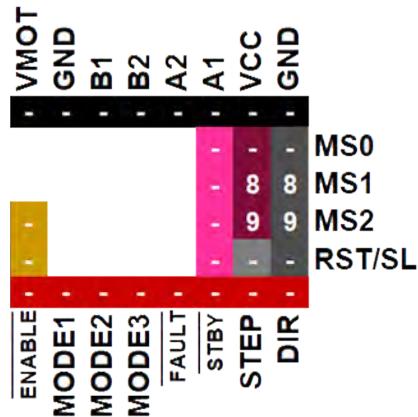
Important: This driver has special requirements in the Configuration and Configuration_adv.h. Also, this driver requires constant cooling the moment any motor is used, or it will switch on and off.

1 / 32



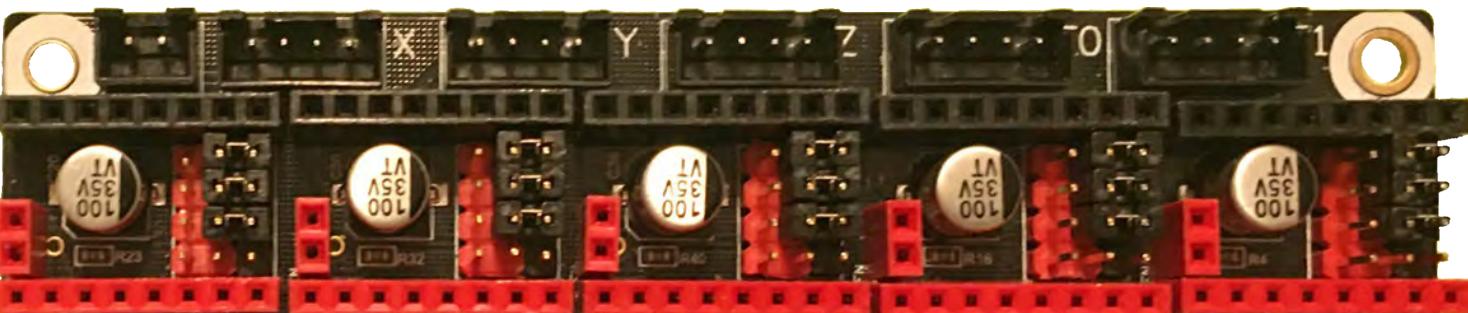
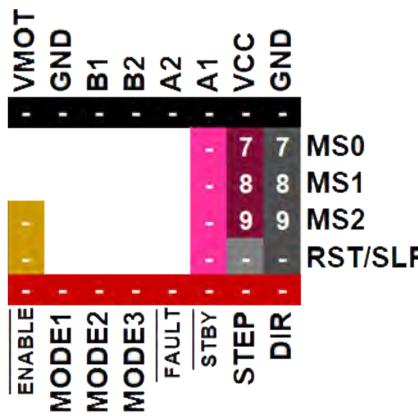
See [Appendix D](#) for legend

1 / 128



See [Appendix D](#) for legend

1 / 256

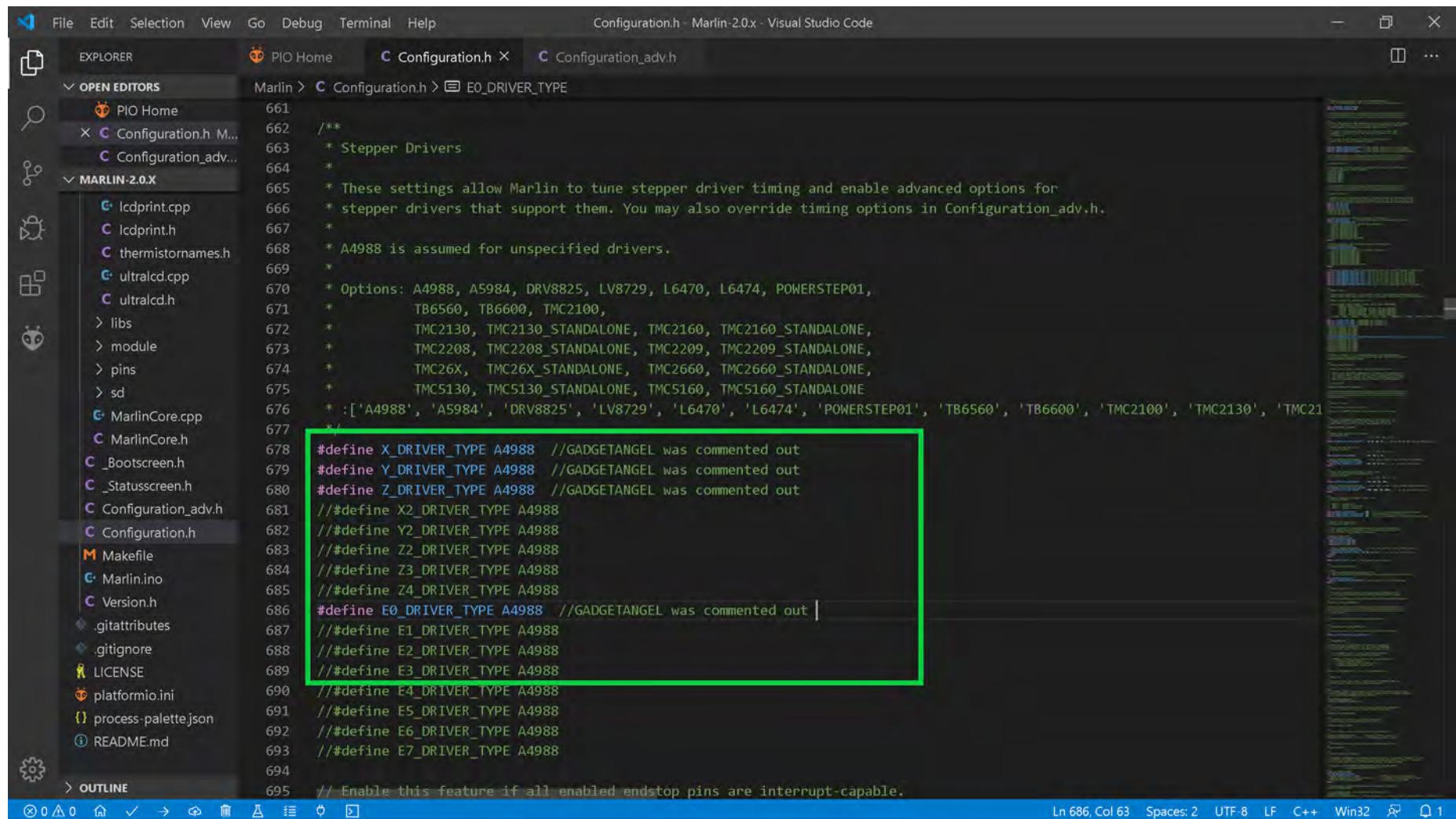


See [Appendix D](#) for legend

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

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

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



```

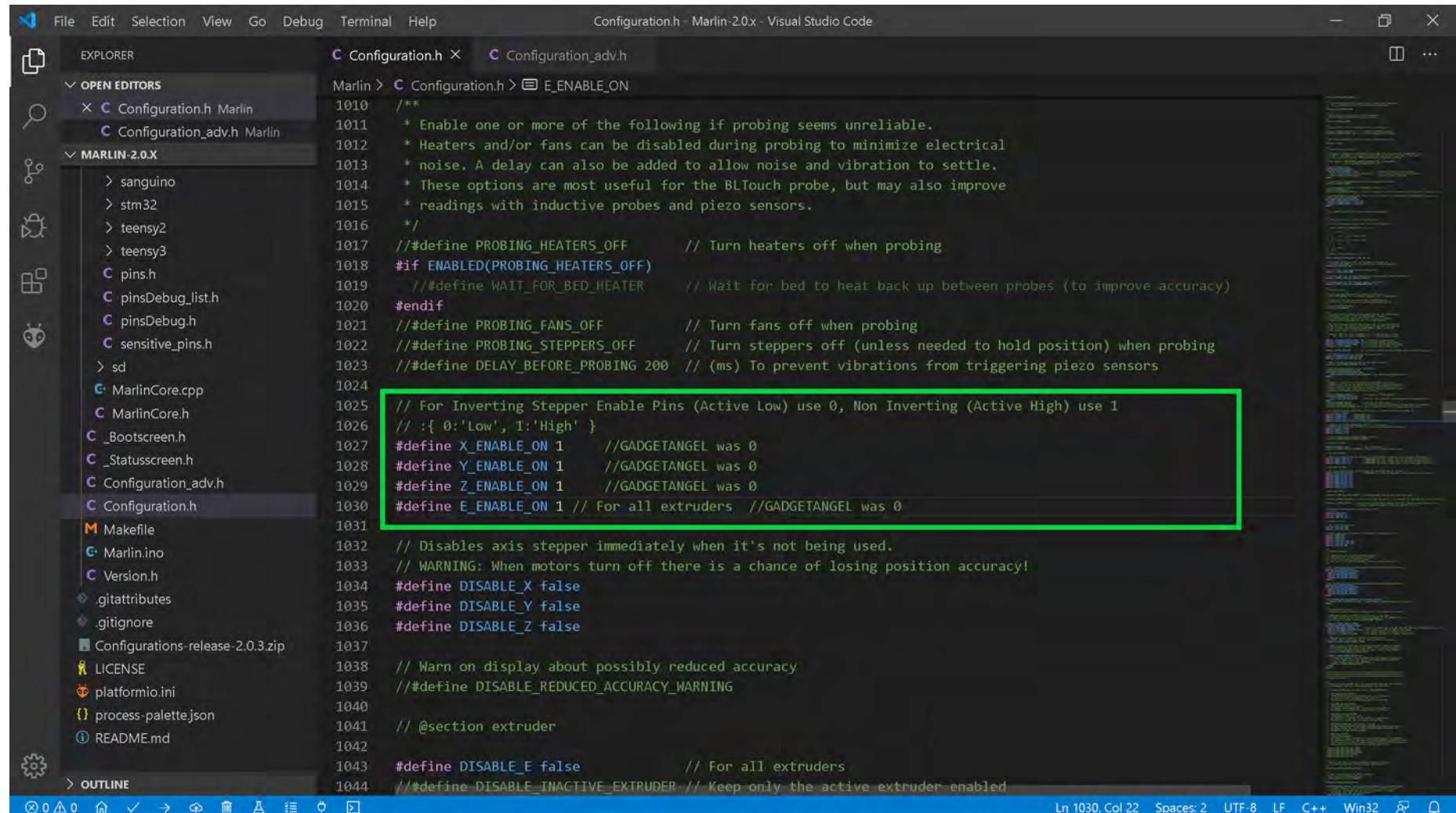
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
MARLIN-2.0.X
Lcdprint.cpp
Lcdprint.h
thermistornames.h
ultralcd.cpp
ultralcd.h
> libs
> module
> pins
> sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
676 */
677
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



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

```

File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h Configuration_adv.h
Marlin > Configuration.h > E_ENABLE_ON
1010 /**
1011 * Enable one or more of the following if probing seems unreliable.
1012 * Heaters and/or fans can be disabled during probing to minimize electrical
1013 * noise. A delay can also be added to allow noise and vibration to settle.
1014 * These options are most useful for the BLTouch probe, but may also improve
1015 * readings with inductive probes and piezo sensors.
1016 */
1017 //#define PROBING_HEATERS_OFF // Turn heaters off when probing
1018 #if ENABLED(PROBING_HEATERS_OFF)
1019   // #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to improve accuracy)
1020 #endif
1021 //#define PROBING_FANS_OFF // Turn fans off when probing
1022 //#define PROBING_STEPPERS_OFF // Turn steppers off (unless needed to hold position) when probing
1023 //#define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering piezo sensors
1024
1025 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
1026 // :{ 0:'Low', 1:'High' }
1027 #define X_ENABLE_ON 1 //GADGETANGEL was 0
1028 #define Y_ENABLE_ON 1 //GADGETANGEL was 0
1029 #define Z_ENABLE_ON 1 //GADGETANGEL was 0
1030 #define E_ENABLE_ON 1 // For all extruders //GADGETANGEL was 0
1031
1032 // Disables axis stepper immediately when it's not being used.
1033 // WARNING: When motors turn off there is a chance of losing position accuracy!
1034 #define DISABLE_X false
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 //#define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 //#define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled

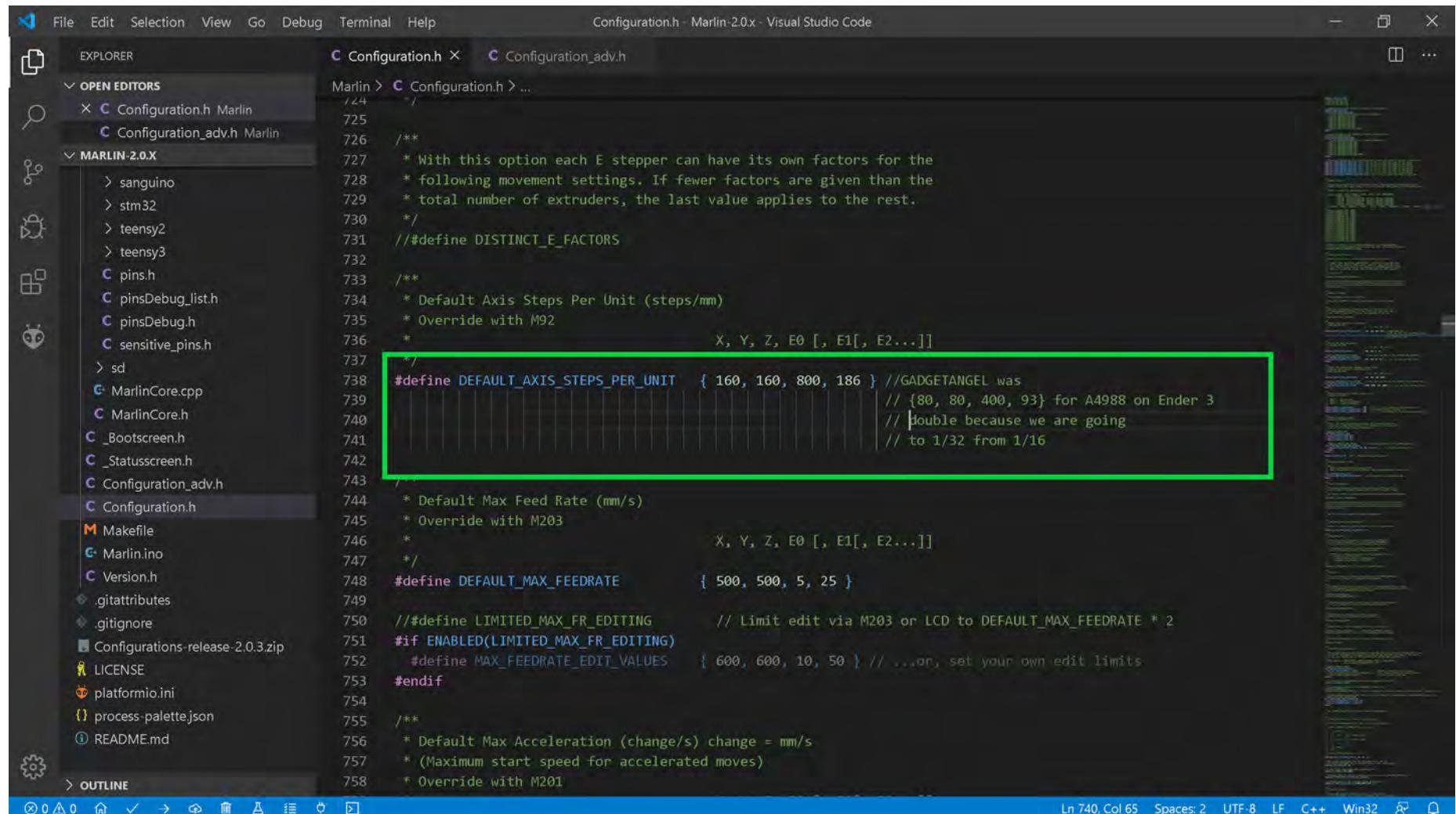
```

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

- Go to the next page.

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

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



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

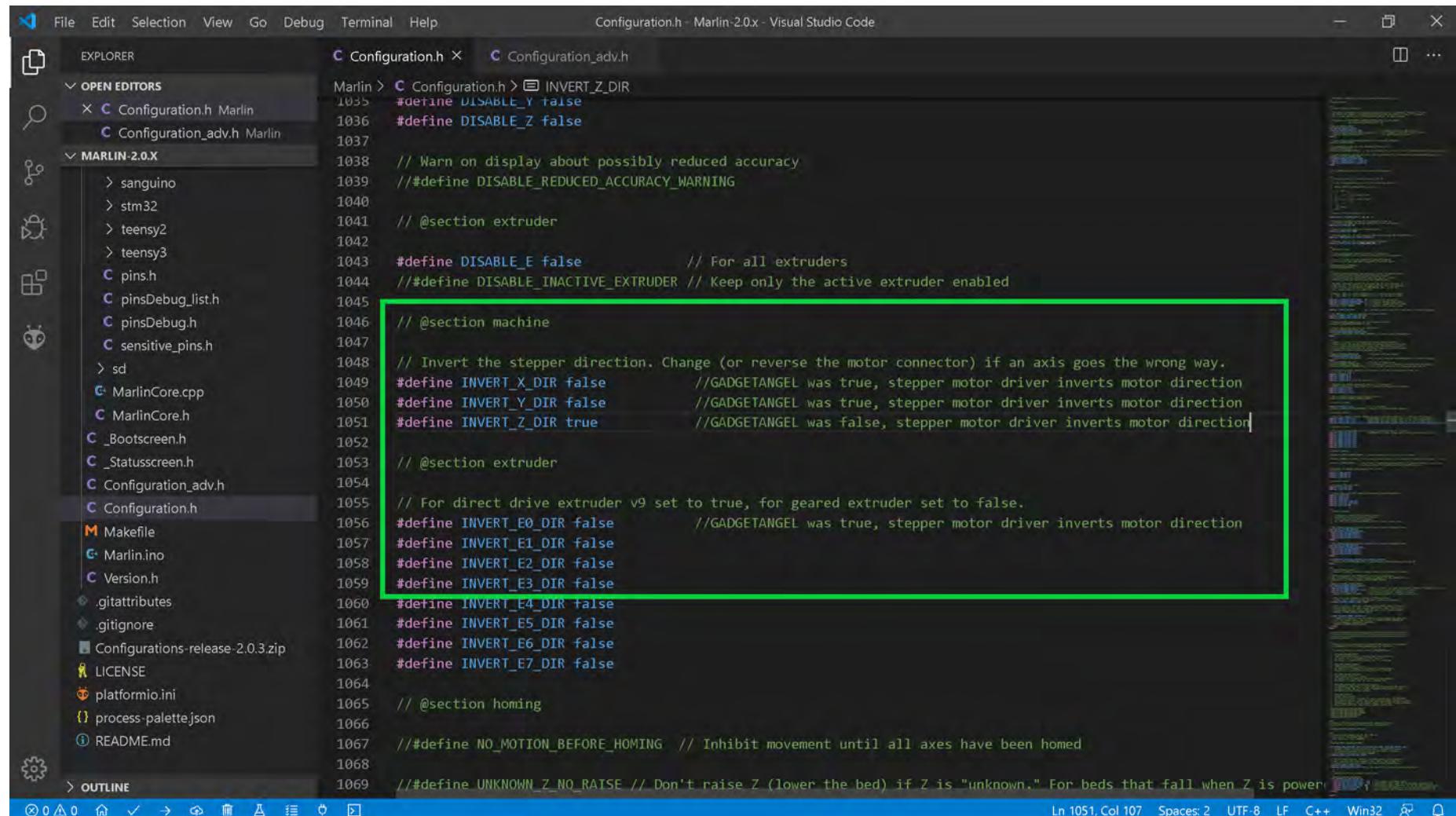
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 1036 #define DISABLE_Z false
  1037
  1038 // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  1041 // @section extruder
  1042
  1043 #define DISABLE_E false          // For all extruders
  1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1045
  1046 // @section machine
  1047
  1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1052
  1053 // @section extruder
  1054
  1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
  1056 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1057 #define INVERT_E1_DIR false
  1058 #define INVERT_E2_DIR false
  1059 #define INVERT_E3_DIR false
  1060 #define INVERT_E4_DIR false
  1061 #define INVERT_E5_DIR false
  1062 #define INVERT_E6_DIR false
  1063 #define INVERT_E7_DIR false
  1064
  1065 // @section homing
  1066
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  
```

The code block shows the Configuration.h file from Marlin 2.0.x. A green rectangular box highlights the following lines of code:

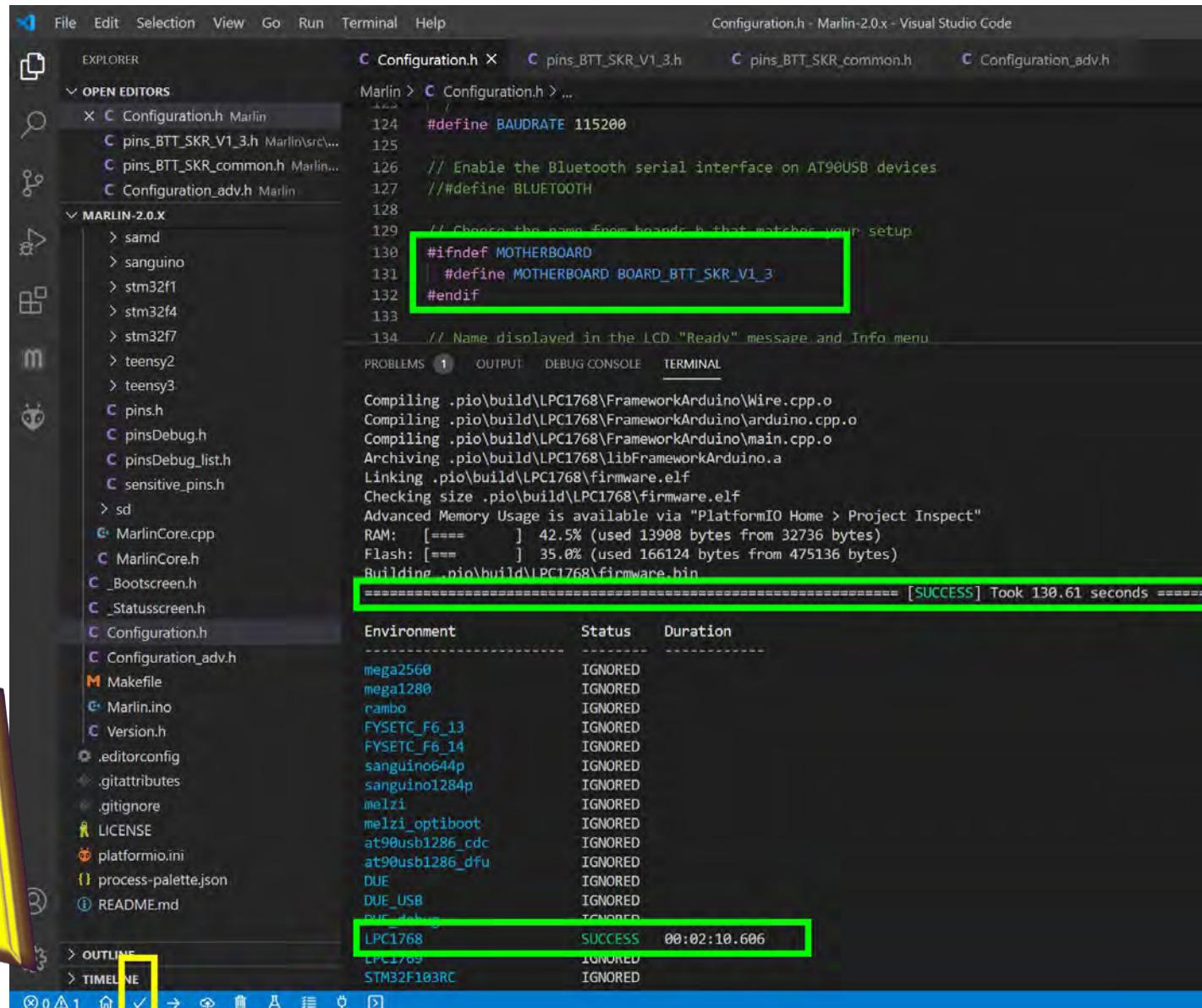
```

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

- Go to the next page.

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

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



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

- EXPLORER:** Shows files in the project structure, including Configuration.h, pins_BTT_SKR_V1_3.h, pins_BTT_SKR_common.h, Configuration_adv.h, and various Marlin source files like MarlinCore.cpp and MarlinCore.h.
- EDITOR:** Displays the Configuration.h file with the following code snippet highlighted in green:

```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif
```
- TERMINAL:** Shows the compilation process and success message:

```
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
[SUCCESS] Took 130.61 seconds =====
```
- PROBLEMS:** Shows one error (1).
- STATUS BAR:** Shows build status: **LPC1768 SUCCESS 00:02:10.606**.

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

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

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

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

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin
MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
TIMELINE

```

```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```

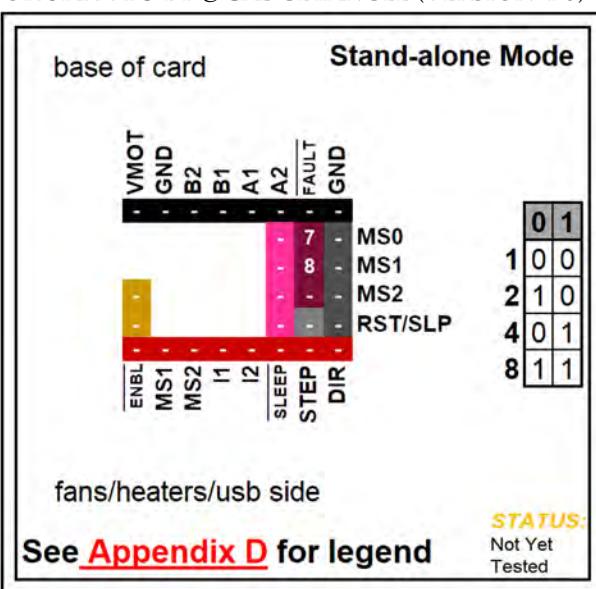
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin

```

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET_3D	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

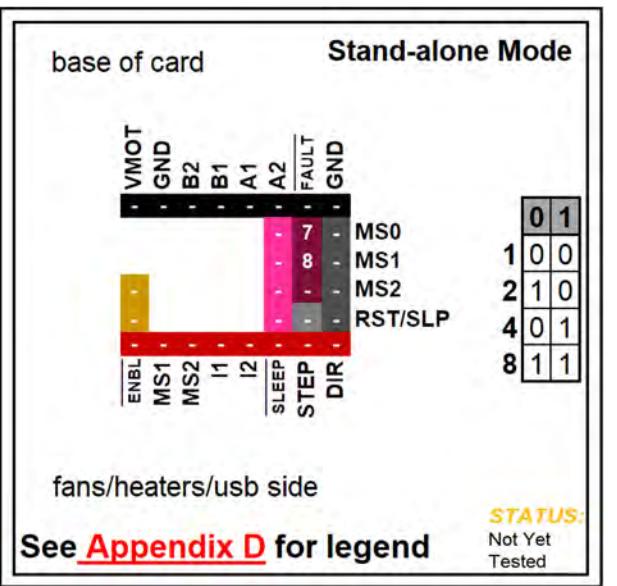


POLOLU MP6500

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

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

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

**Driver Chip Chart:**

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

Driving Current Calculation Formula $I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

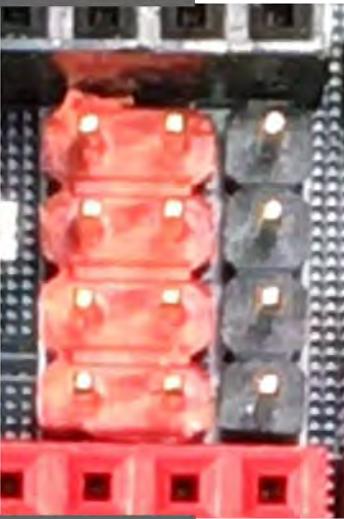
$R_S(\text{Typical Sense Resistor}) = X.XX \Omega$

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

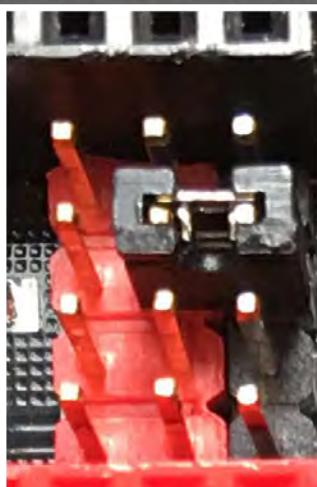
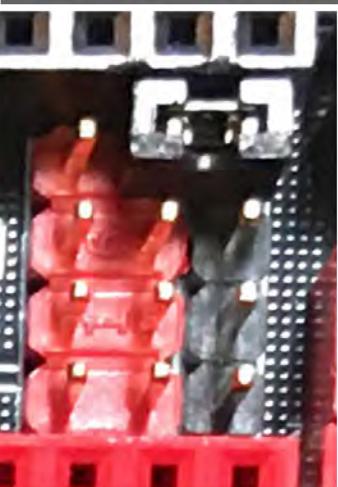
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

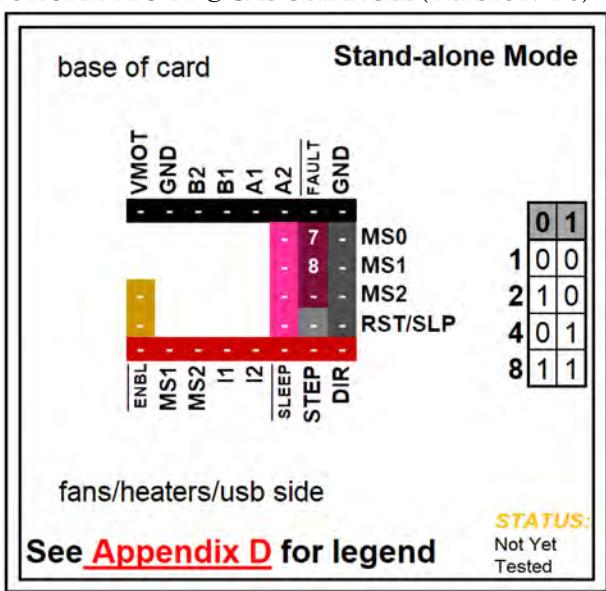
POLOLU MP6500**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Low **No Jumper**

**Rows:****MS0****MS1****MS2****RST/SLP**

High **Jumper Set**

Rows:**MS0****MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

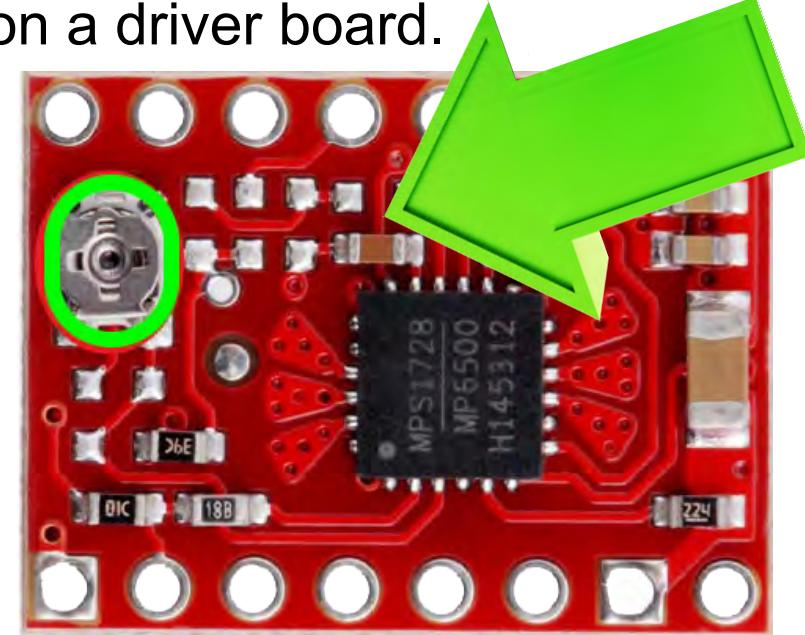
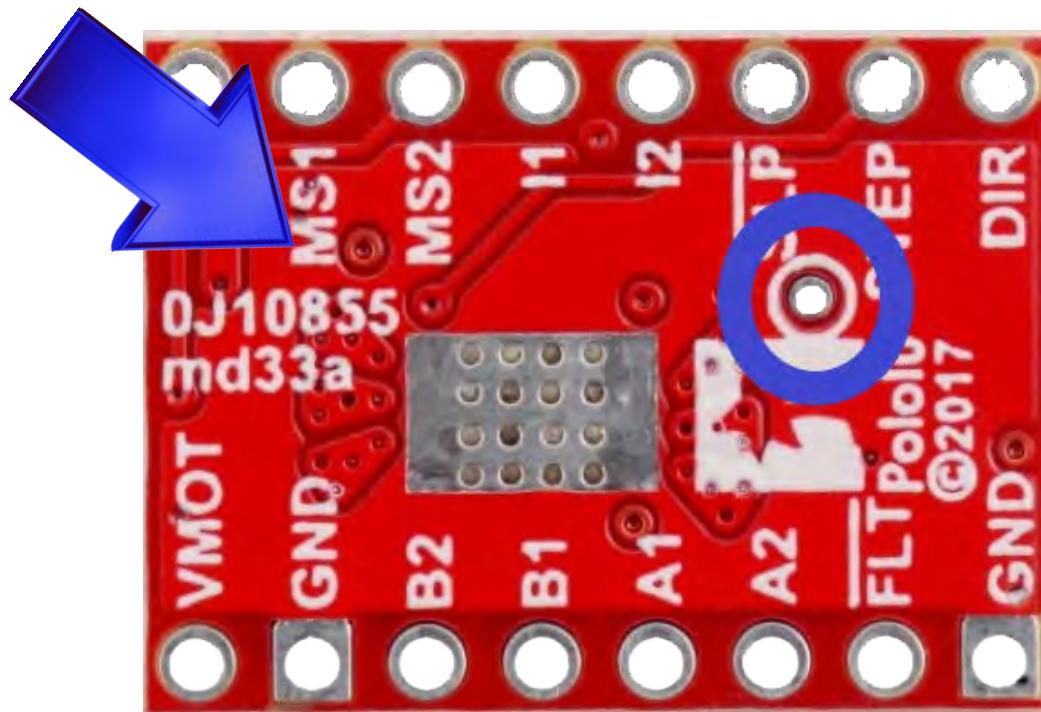


POLOLU MP6500

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

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

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

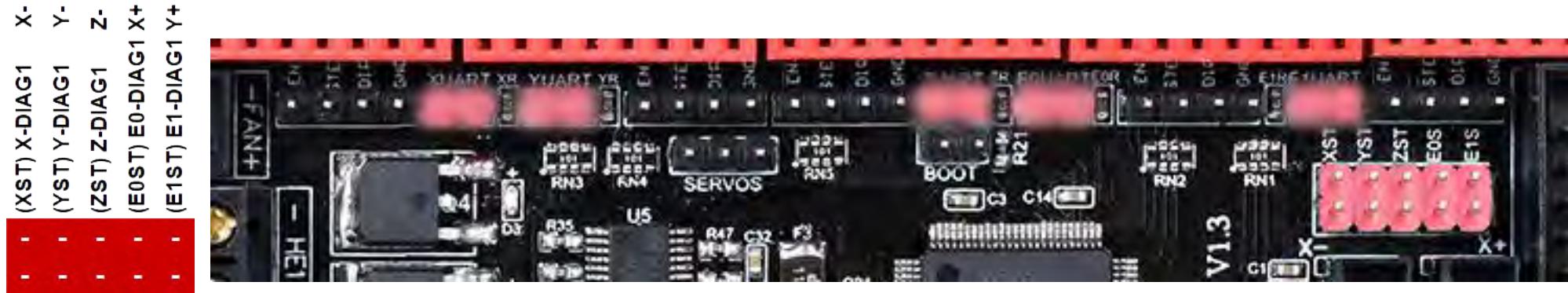


Note: MP6500 driver board does not use external current sense resistors (R_s).

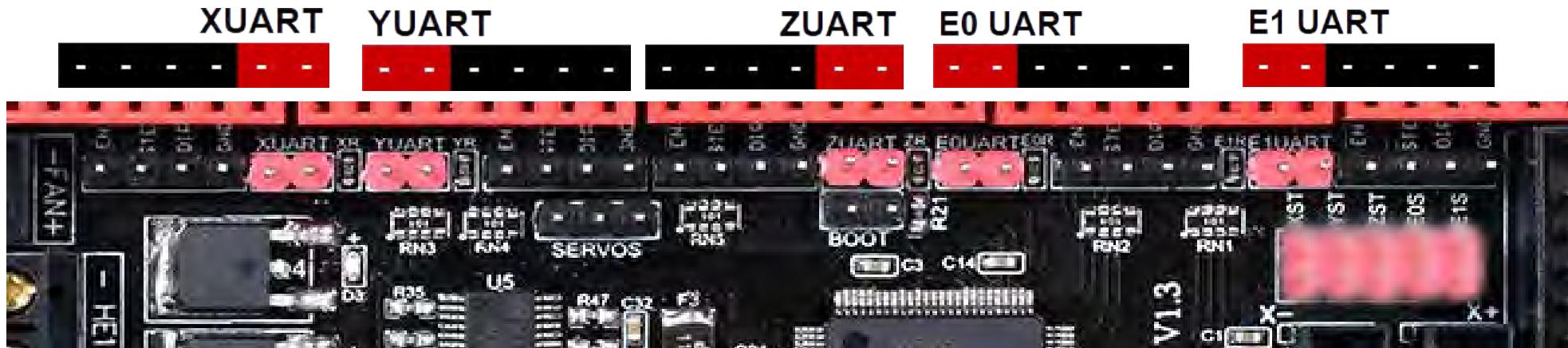
Stand-alone Mode

POLOLU MP6500

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



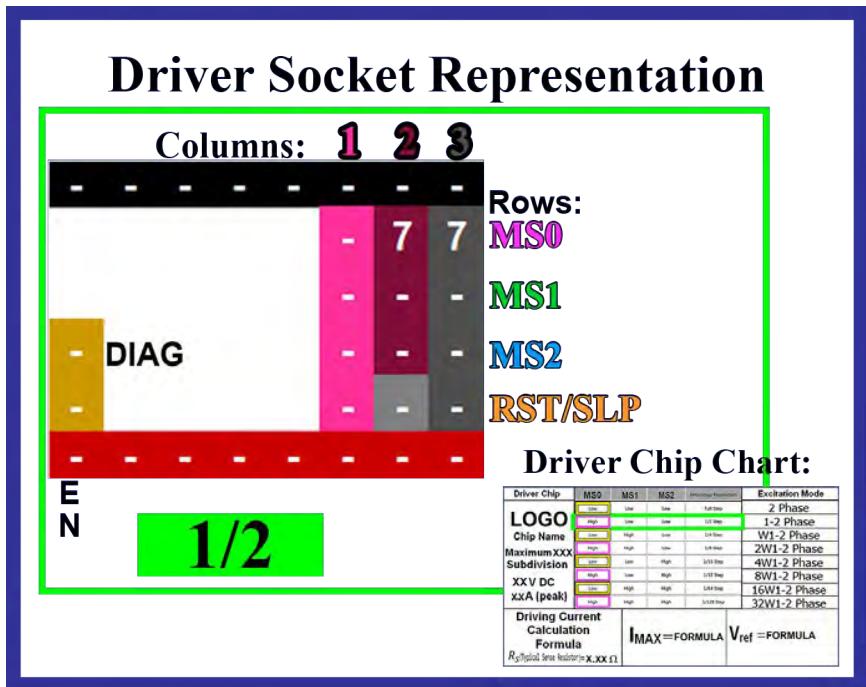
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



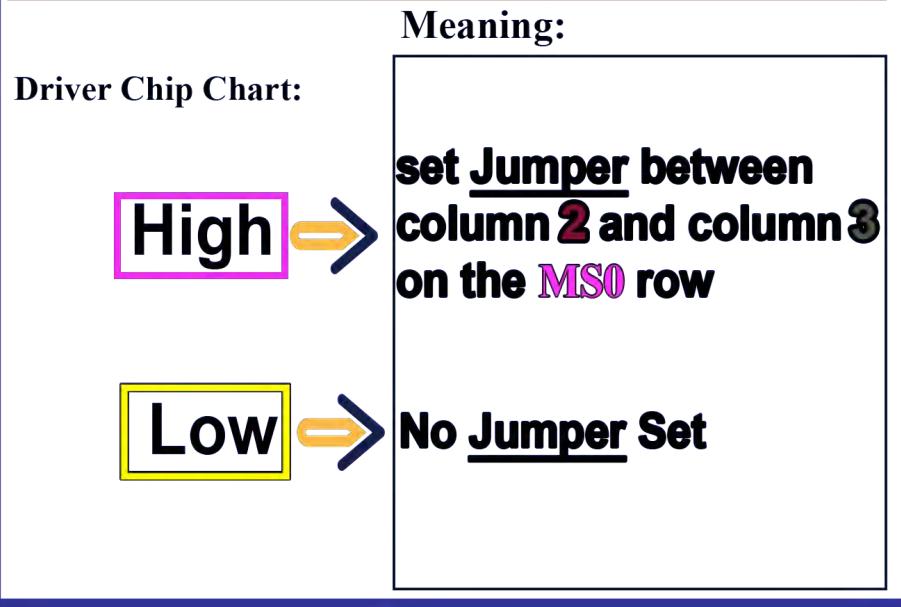
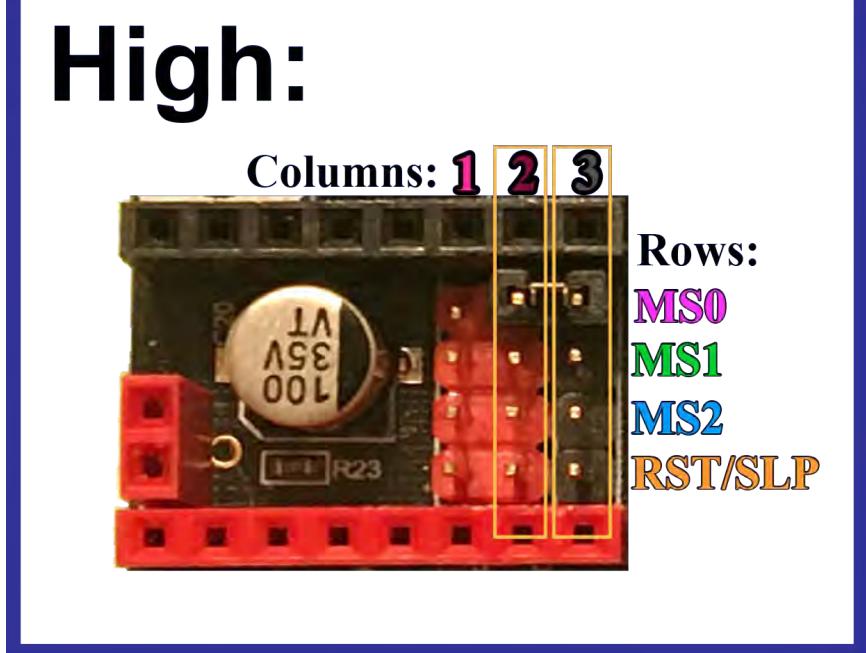
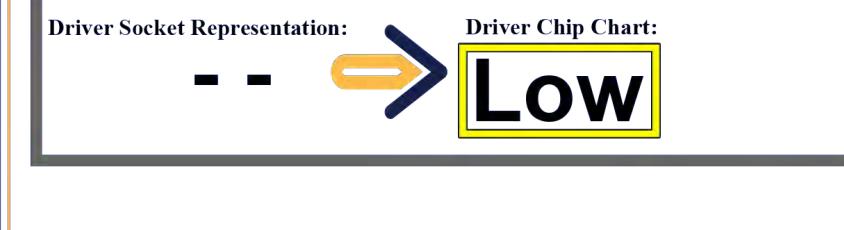
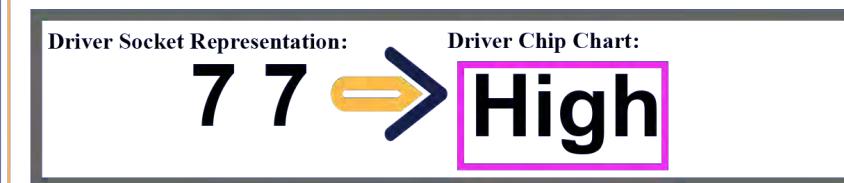
Stand-alone Mode

POLOLU MP6500

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

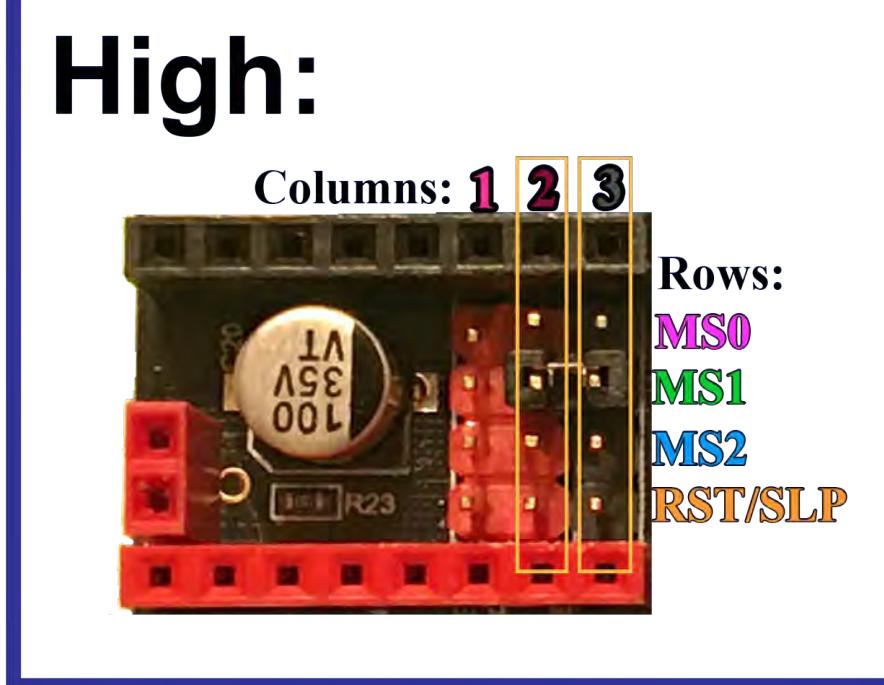
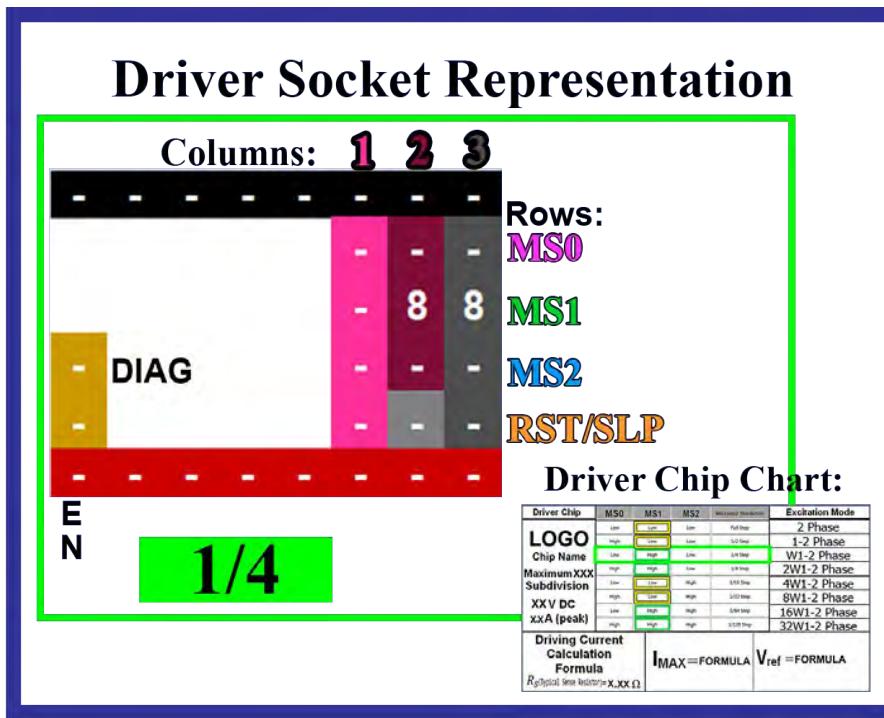


MS0 for Binary State Drivers:

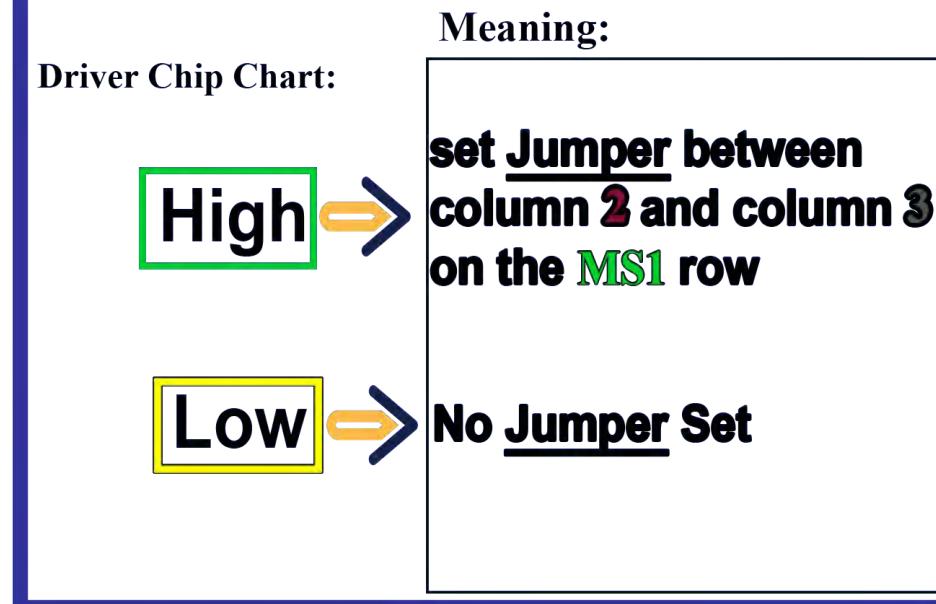
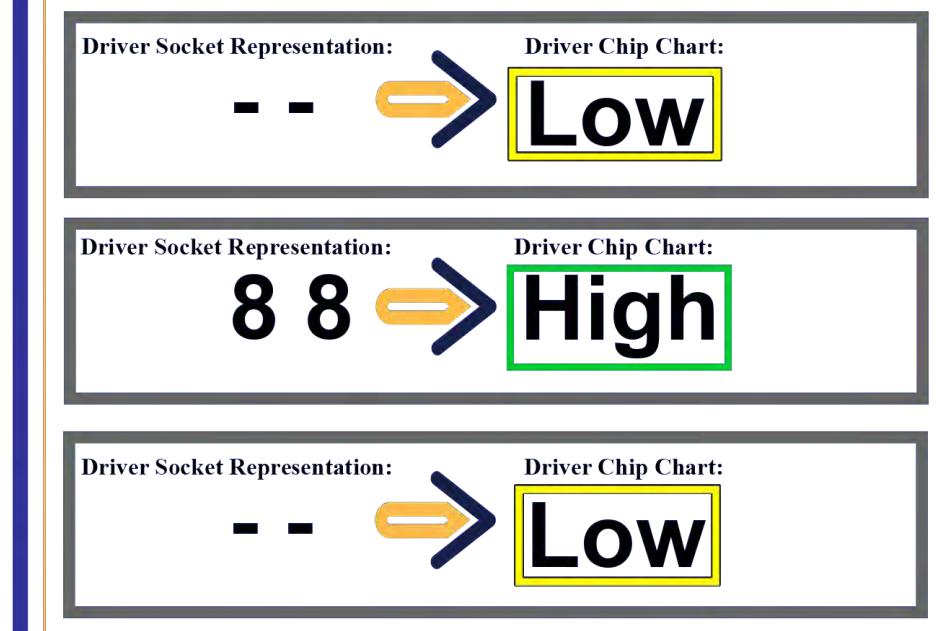


Stand-alone Mode

POLOLU MP6500



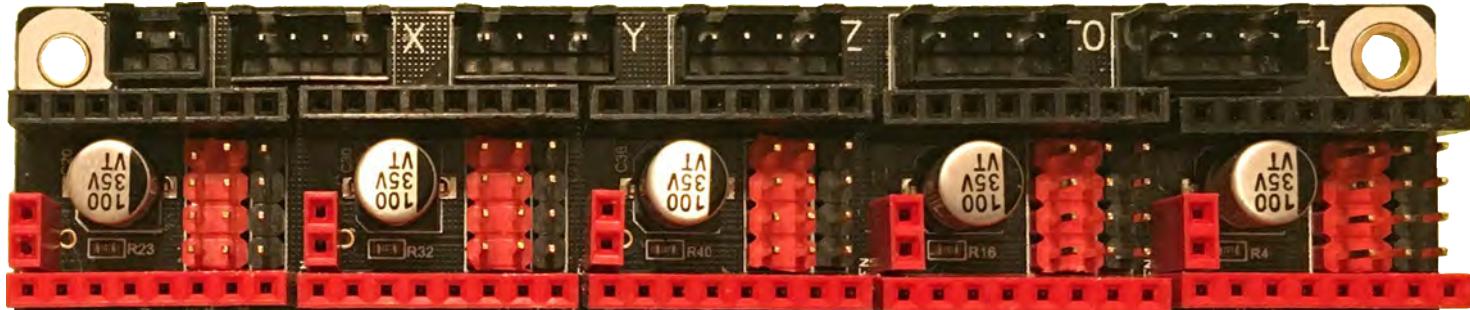
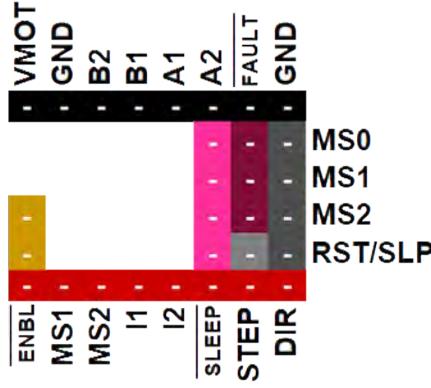
MS1 for Binary State Drivers:



Stand-alone Mode

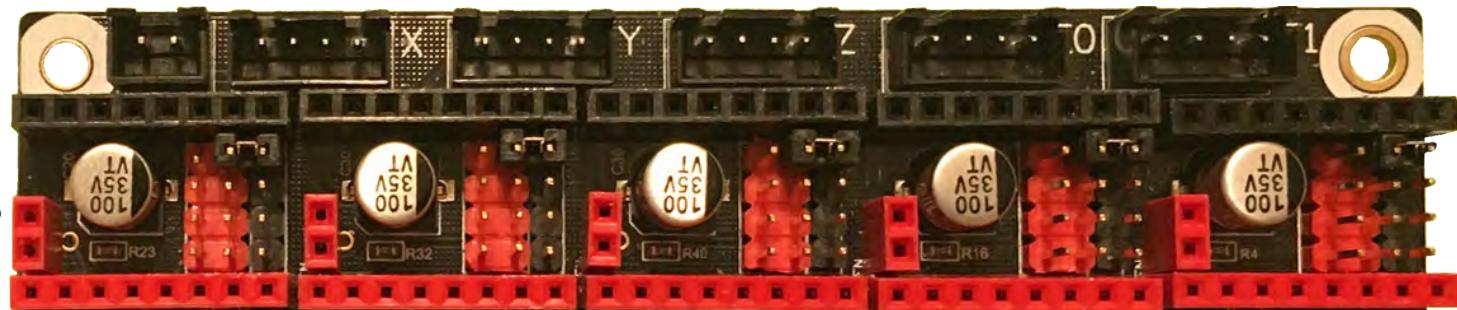
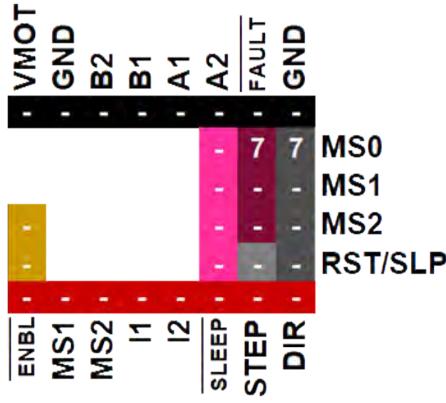
POLOLU MP6500

STEP



See [Appendix D](#) for legend

1 / 2

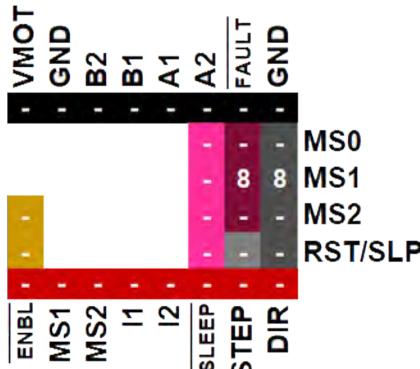


See [Appendix D](#) for legend

Stand-alone Mode

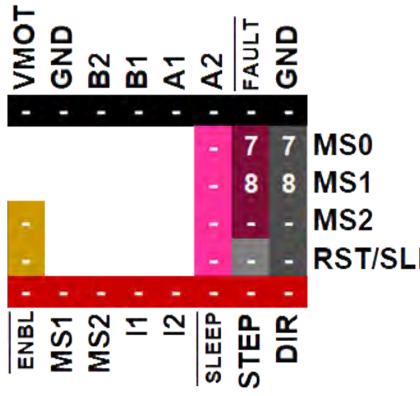
POLOLU MP6500

1 / 4



See [Appendix D](#) for legend

1 / 8

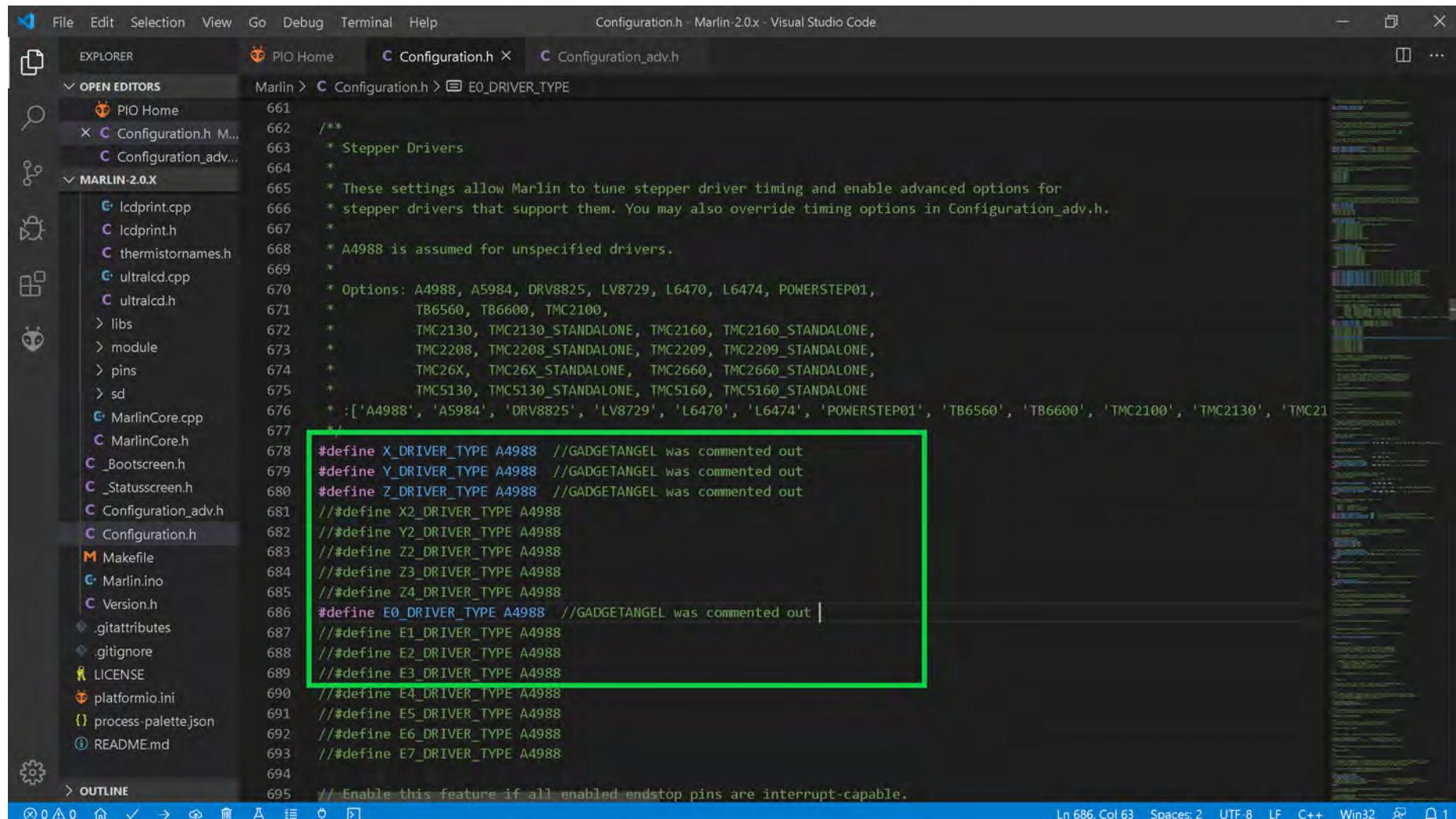


See [Appendix D](#) for legend

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

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

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



```

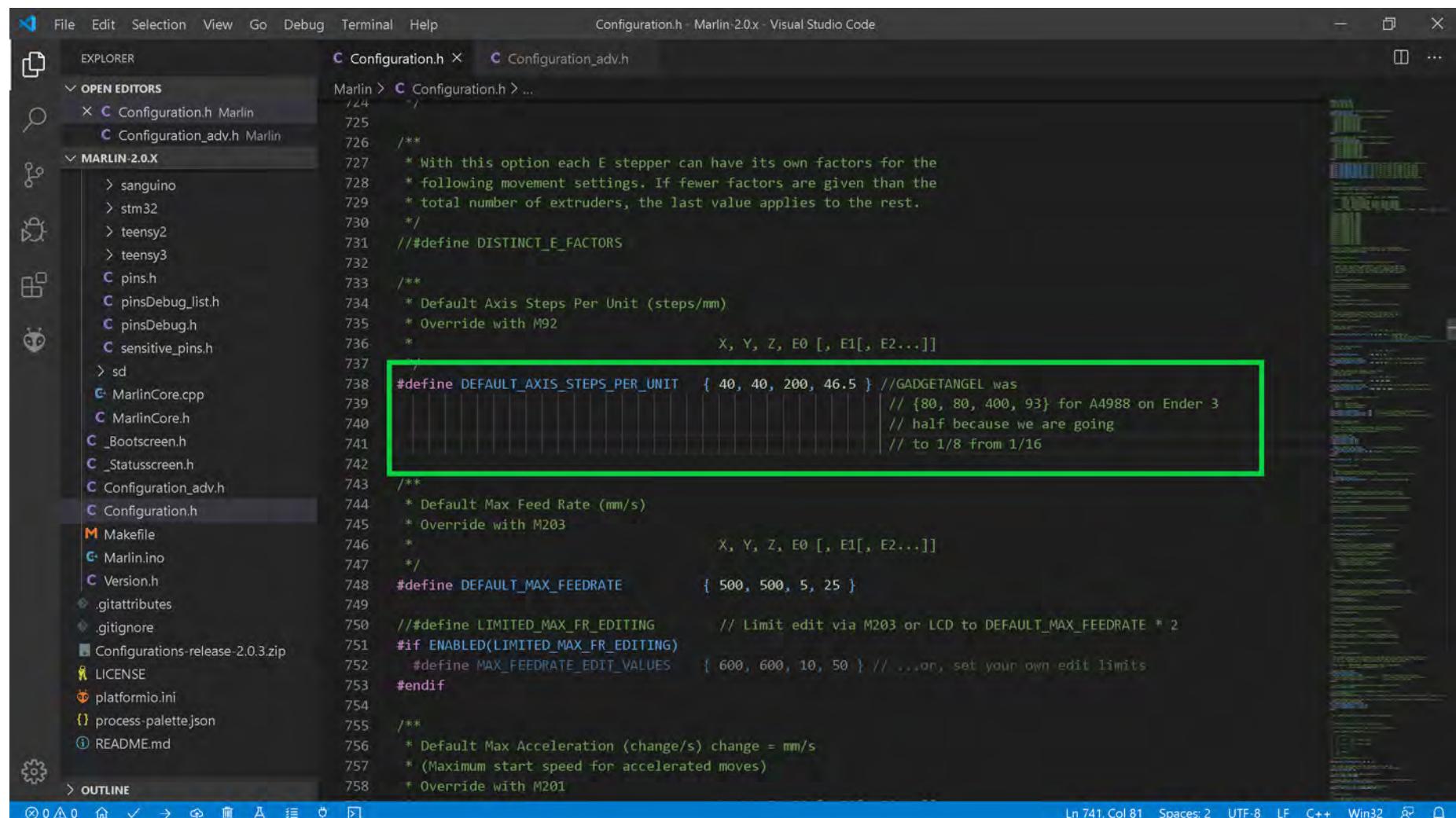
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
678 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 //##define X2_DRIVER_TYPE A4988
681 //##define Y2_DRIVER_TYPE A4988
682 //##define Z2_DRIVER_TYPE A4988
683 //##define Z3_DRIVER_TYPE A4988
684 //##define Z4_DRIVER_TYPE A4988
685 //##define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out |
686 //##define E1_DRIVER_TYPE A4988
687 //##define E2_DRIVER_TYPE A4988
688 //##define E3_DRIVER_TYPE A4988
689 //##define E4_DRIVER_TYPE A4988
690 //##define E5_DRIVER_TYPE A4988
691 //##define E6_DRIVER_TYPE A4988
692 //##define E7_DRIVER_TYPE A4988
693 //##define E8_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.
Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1

```

- Go to the next page.

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

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



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

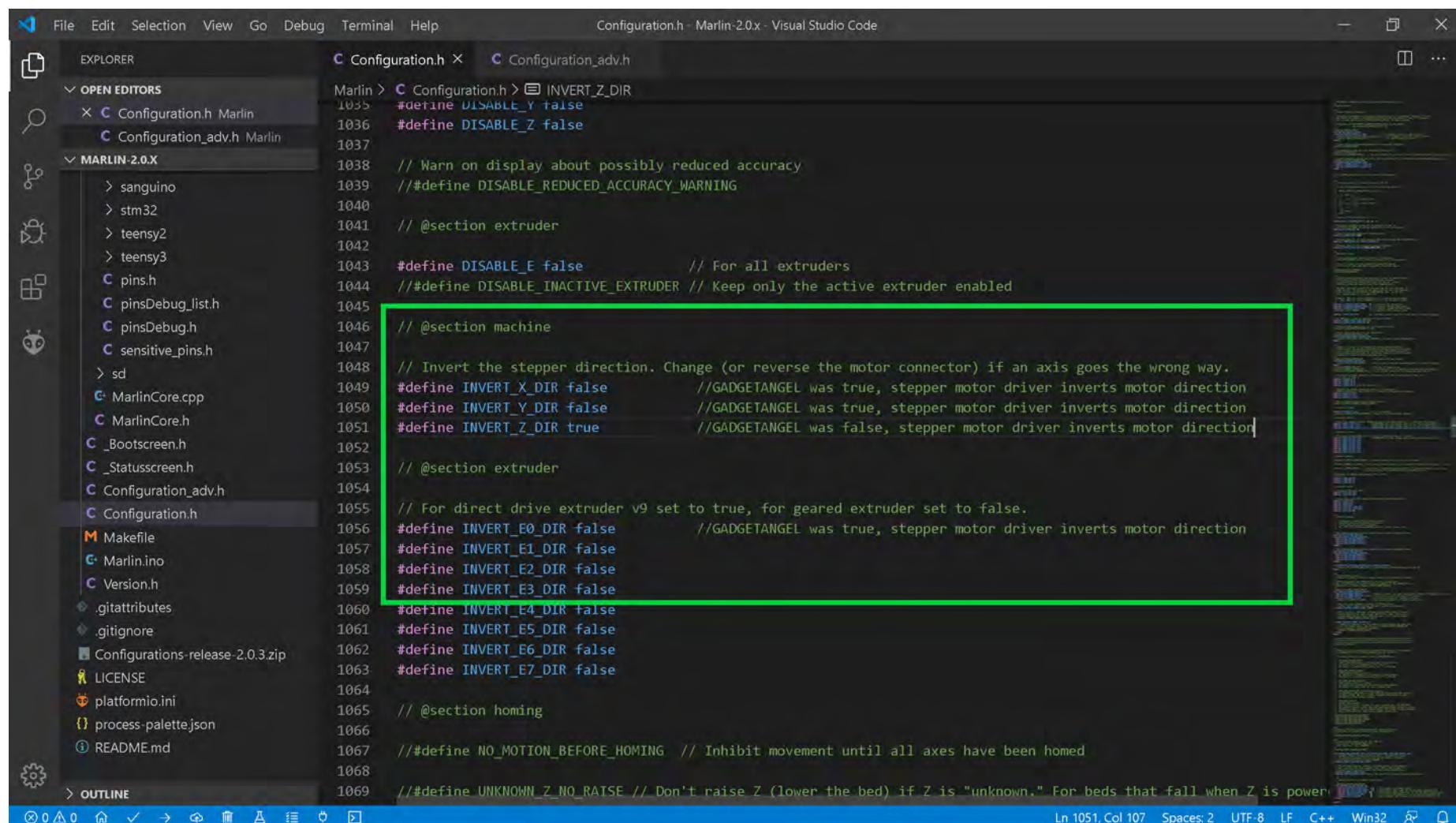
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 40, 40, 200, 46.5 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// half because we are going
// to 1/8 from 1/16
```

The code editor's status bar at the bottom right indicates: Ln 741, Col 81, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

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

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



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet of Marlin 2.0.x configuration code:

```

Marlin > C Configuration.h > #define INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

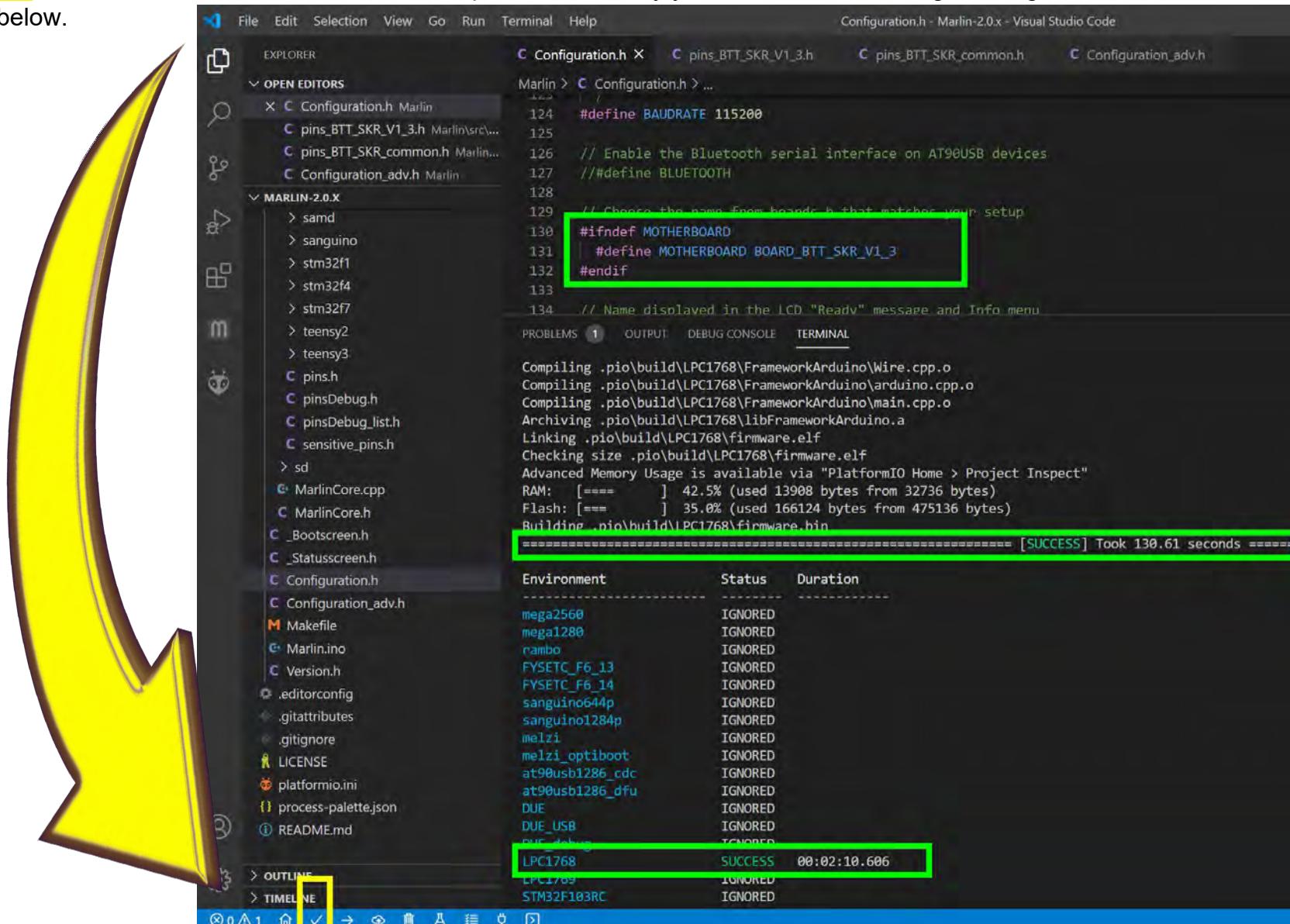
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that this setting should be changed from "false" to "true" if using a POLOLU MP6500 driver.

- Go to the next page.

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

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



A screenshot of the Visual Studio Code interface. On the left is the Explorer sidebar with project files like Configuration.h, pins_BTT_SKR_V1_3.h, pins_BTT_SKR_common.h, and Configuration_adv.h. Below the sidebar are buttons for OUTLINE and TIMELINE, with the OUTLINE button highlighted in yellow. The main editor area shows code for Configuration.h, specifically defining the motherboard as BOARD_BTT_SKR_V1_3. The status bar at the bottom shows the build output: "===== [SUCCESS] Took 130.61 seconds =====". A green box highlights the success message. A large yellow arrow points downwards from the top-left towards the OUTLINE button.

```

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

EXPLORER
OPEN EDITORS
Configuration.h Marlin
pins_BTT_SKR_V1_3.h Marlin\src...
pins_BTT_SKR_common.h Marlin...
Configuration_adv.h Marlin

MARLIN-2.0.X
samd
sanguino
stm32f1
stm32f4
stm32f7
teensy2
teensy3
pins.h
pinsDebug.h
pinsDebug.list.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration.h
Configuration_adv.h
Makefile
Marlin.ino
Version.h
.editorconfig
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

OUTLINE
TIMELINE

```

```

Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the name from board to match your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

Environment Status Duration
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_DFU IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

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

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

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

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

EXPLORER OPEN EDITORS Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

Configuration.h

```

Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

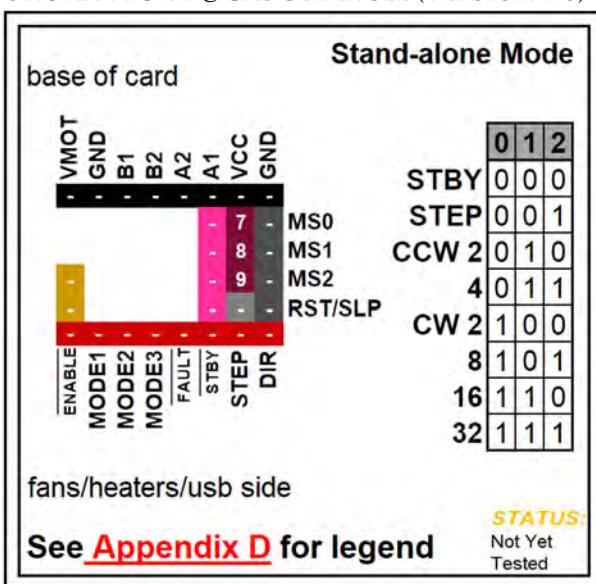
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET_3D	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

OUTLINE TIMELINE

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

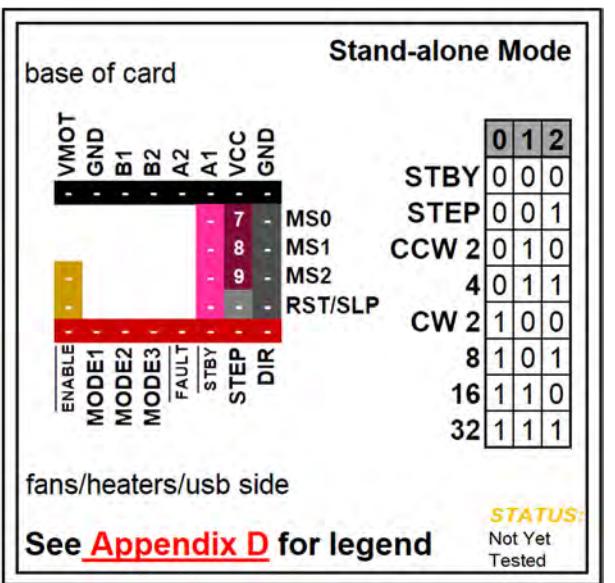


POLOLU TB67S249FTG

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

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

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



Driver Chip Chart:

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

Driving Current Calculation Formula

$I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

$R_S(\text{Typical Sense Resistor}) = X.XX \Omega$

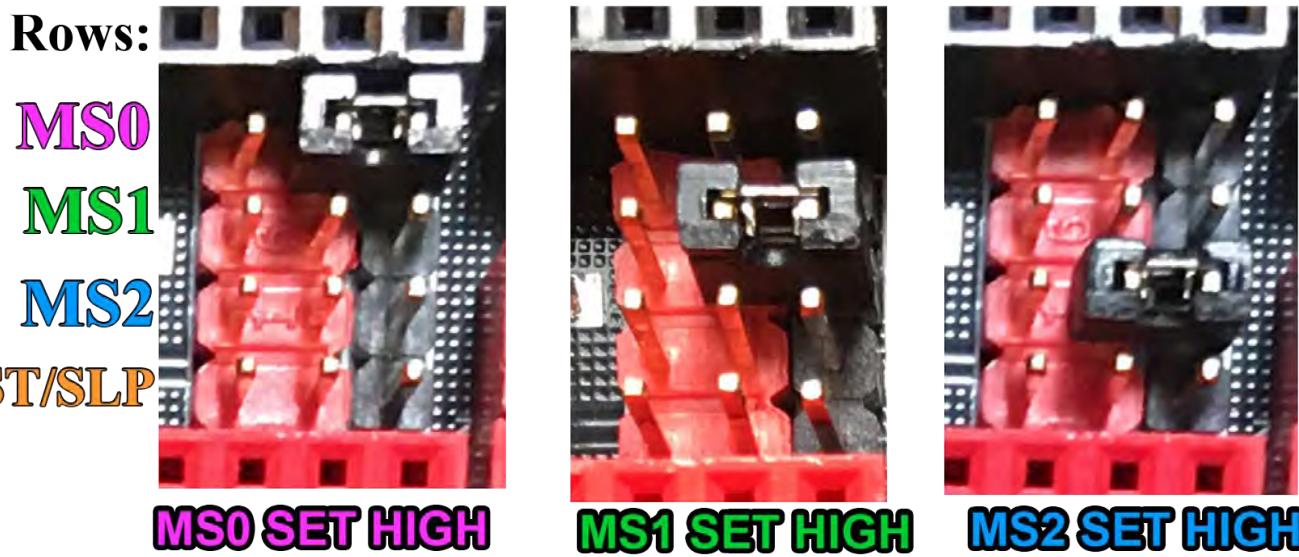
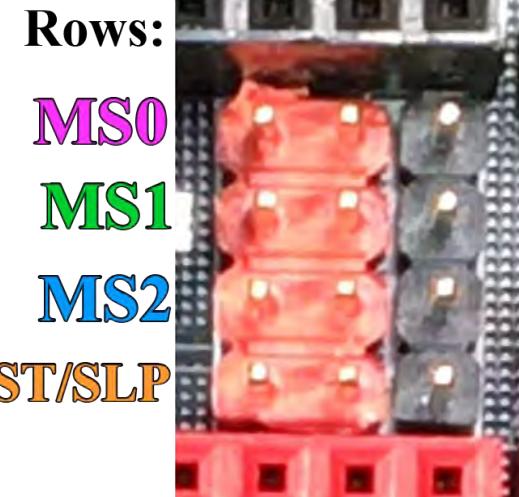
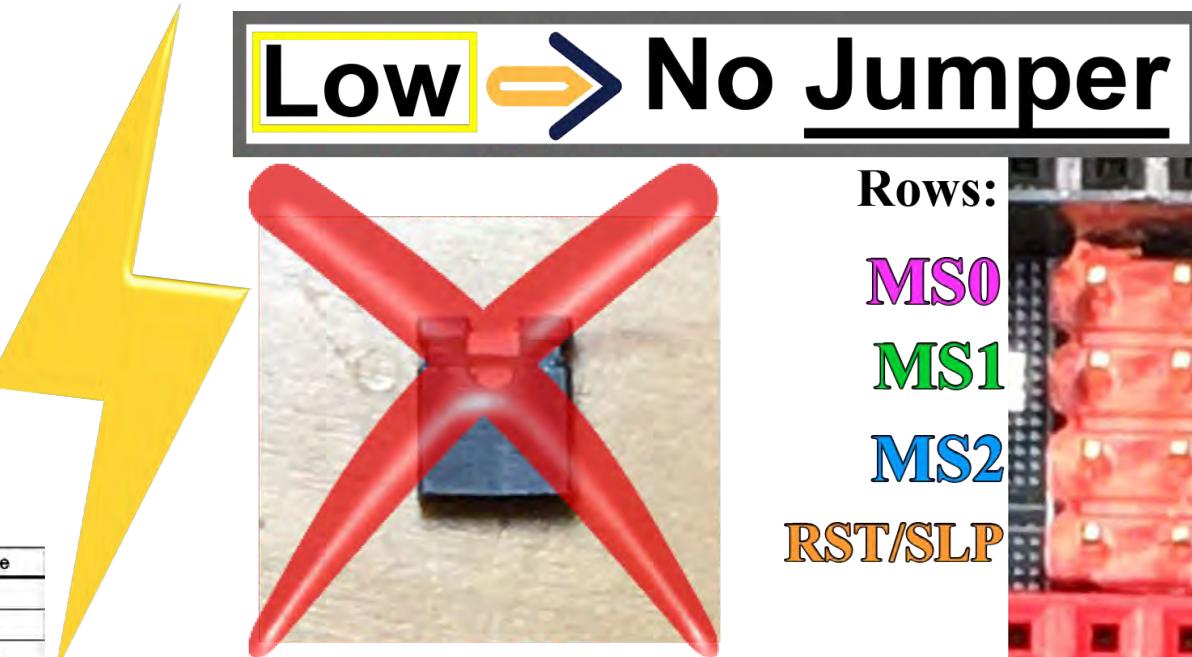
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

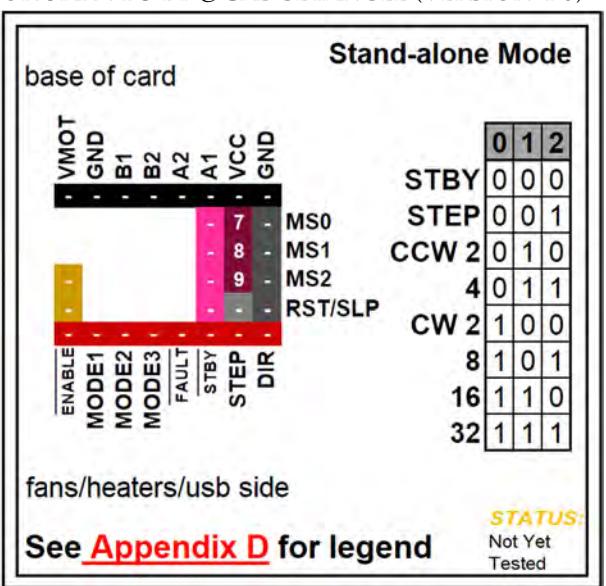


Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):



POLOLU TB67S249FTG SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers



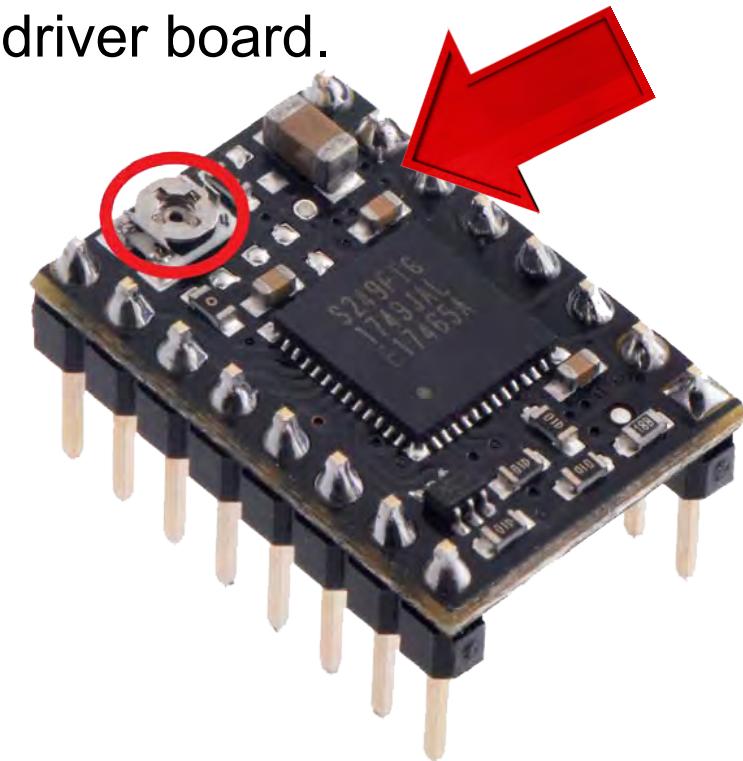
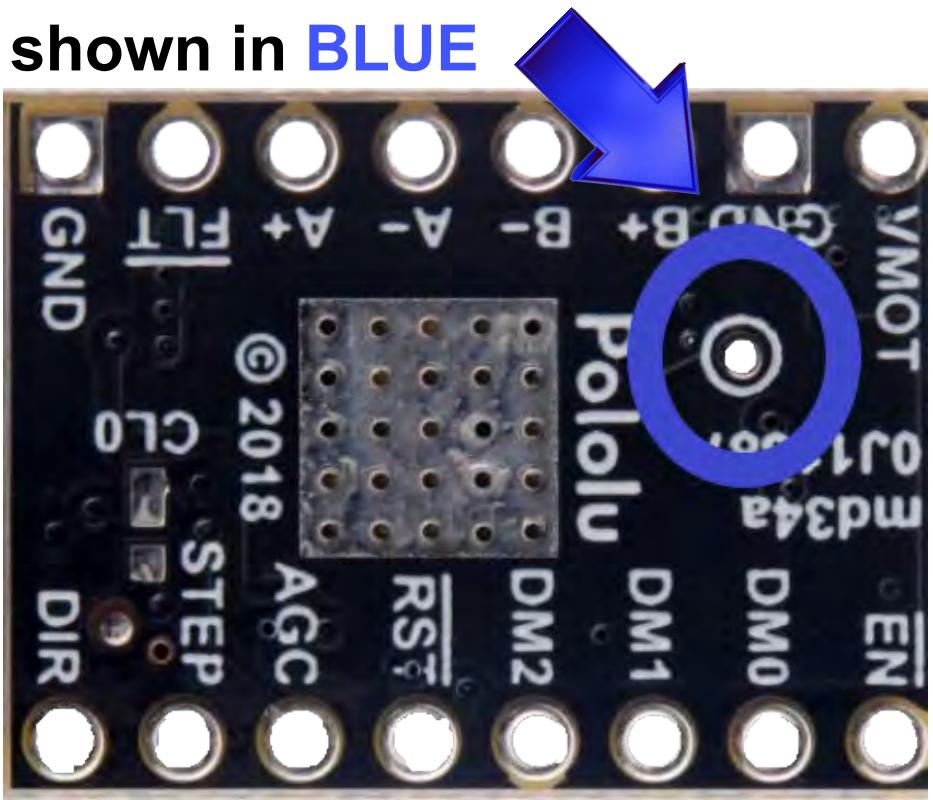


POLOLU TB67S249FTG

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

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

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

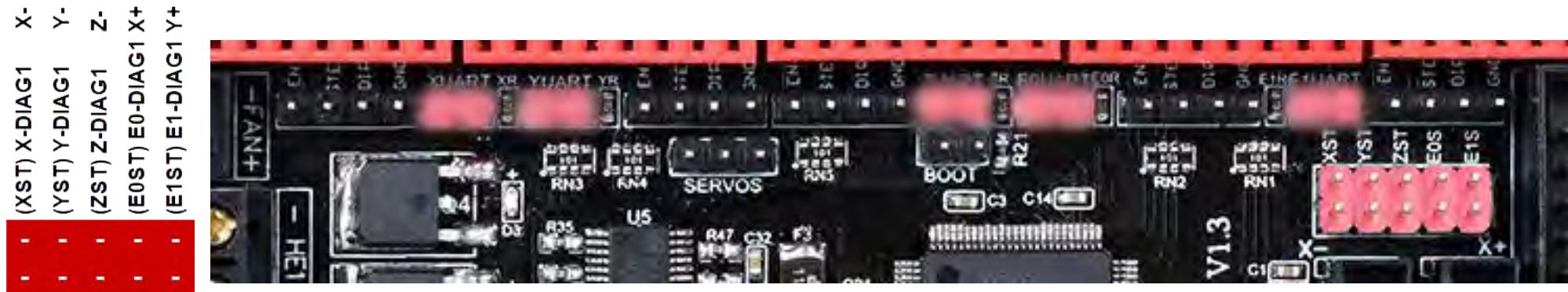


Note: TB67S249FTG driver board does not use external current sense resistors (R_s).

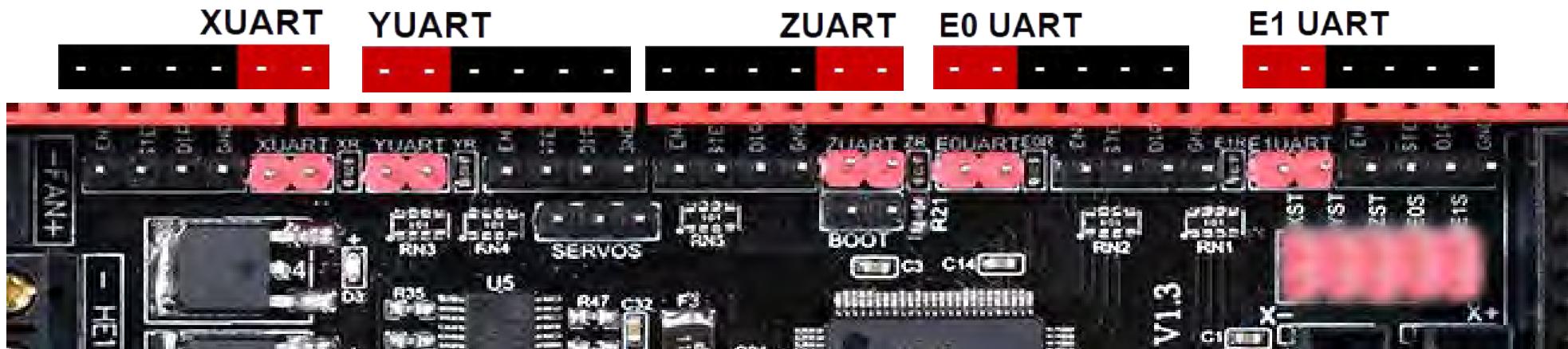
Stand-alone Mode

POLOLU TB67S249FTG

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



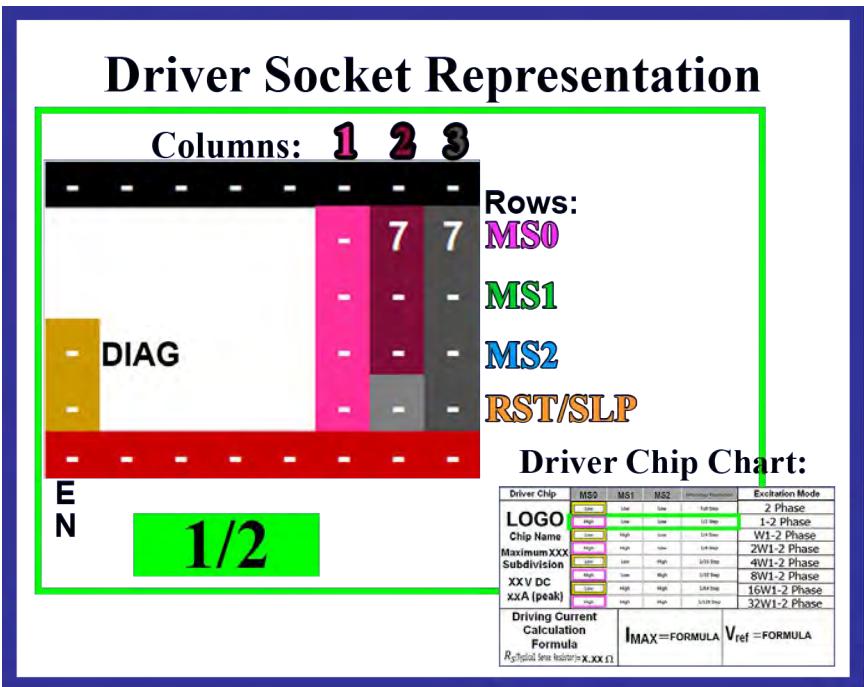
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



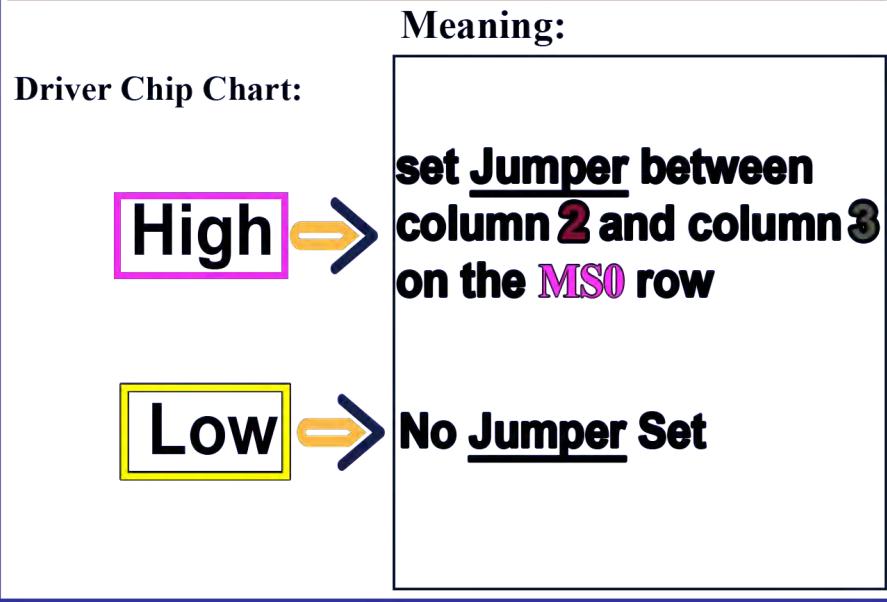
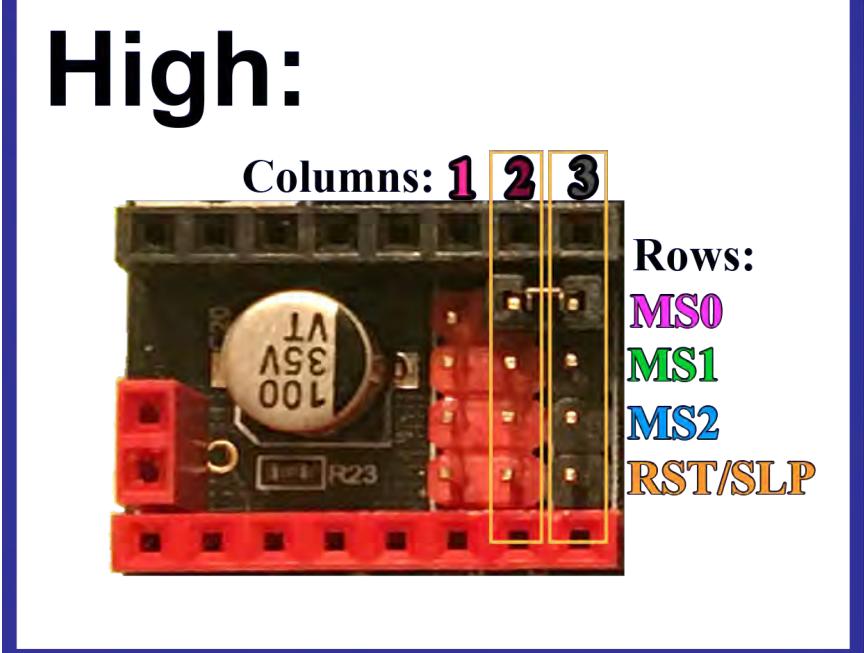
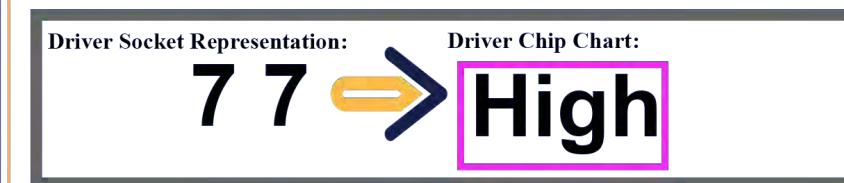
Stand-alone Mode

POLOLU TB67S249FTG

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

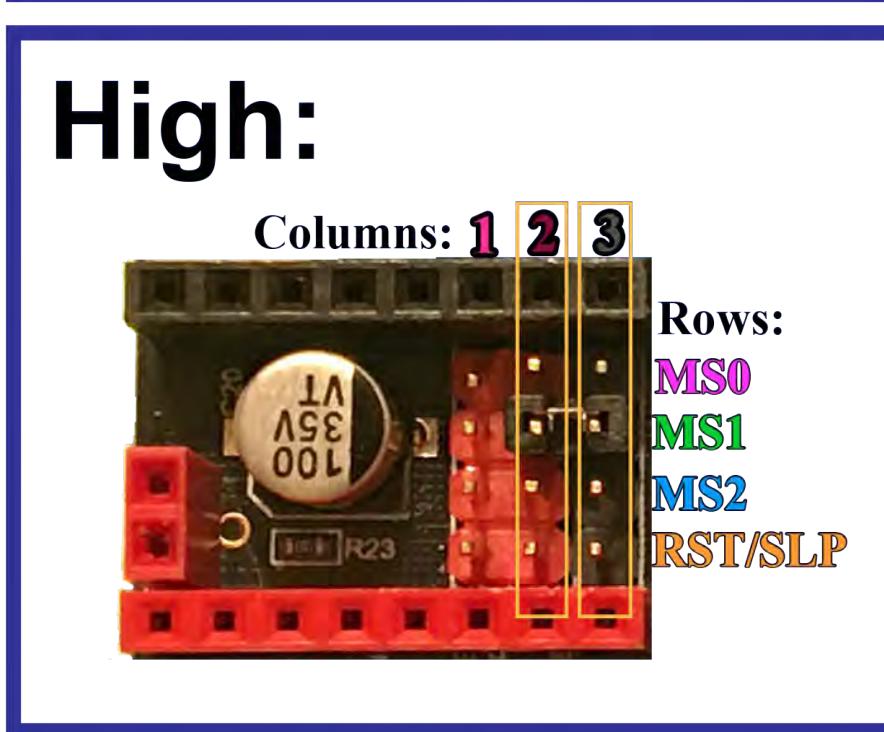
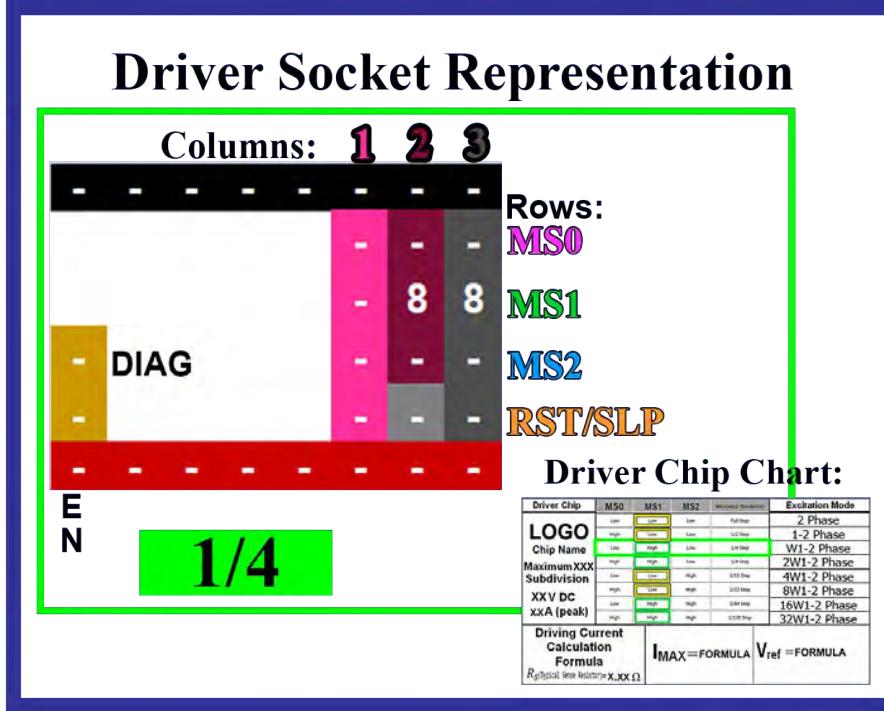


MS0 for Binary State Drivers:

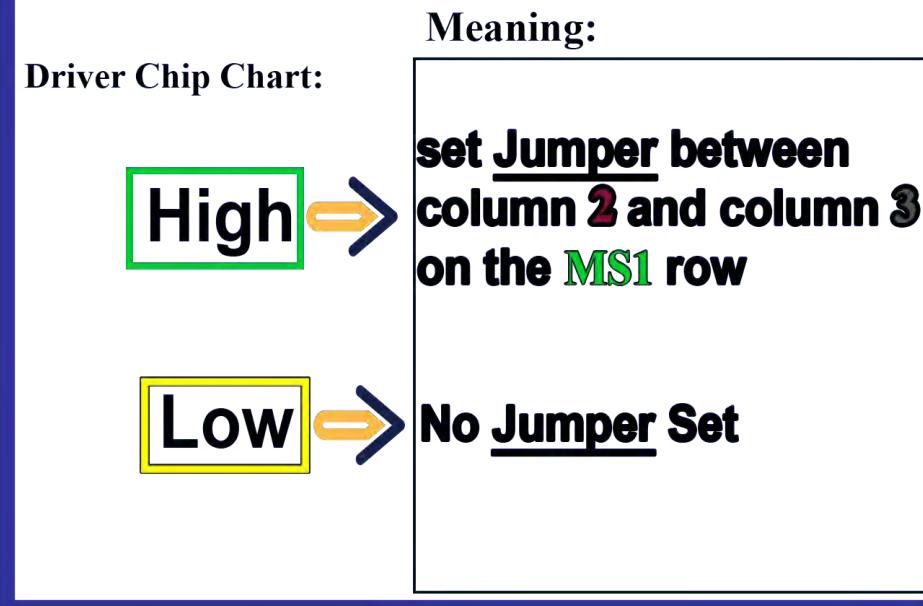
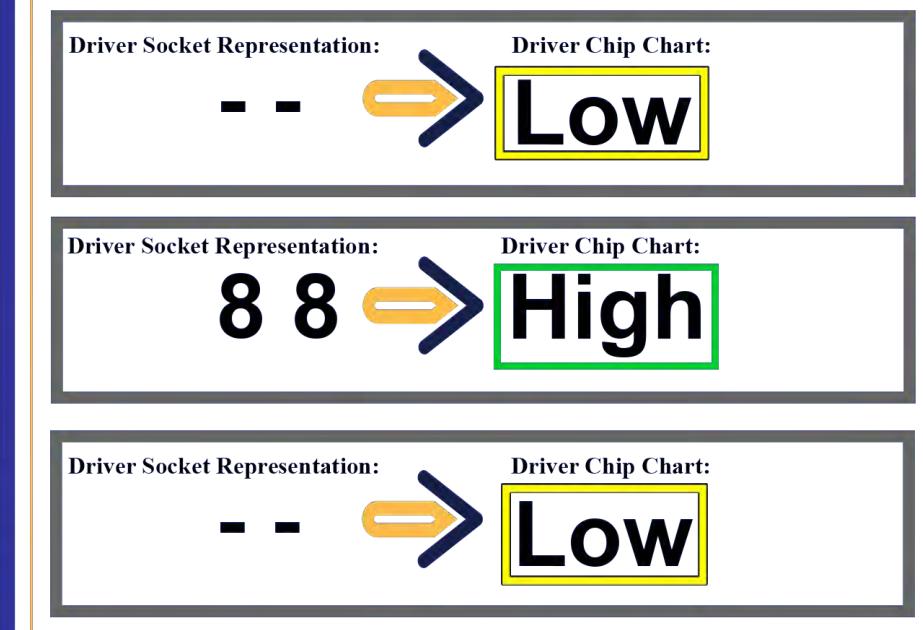


Stand-alone Mode

POLOLU TB67S249FTG



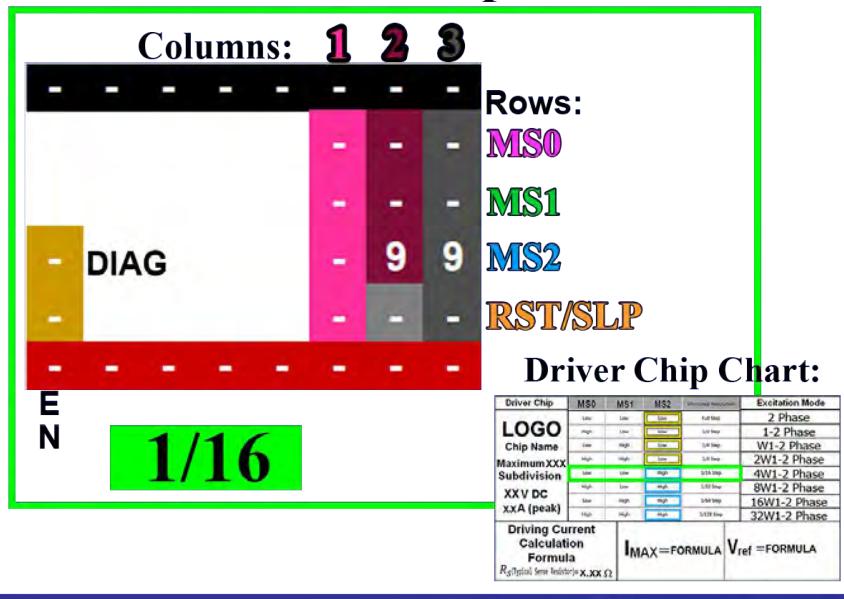
MS1 for Binary State Drivers:



Stand-alone Mode

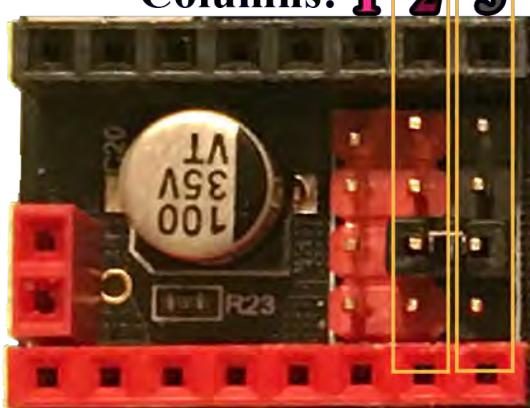
POLOLU TB67S249FTG

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**, **MS1**, **MS2**, **RST/SLP**

MS2 for Binary State Drivers:

Driver Socket Representation:

- -

Driver Chip Chart:

LOW

Driver Socket Representation:

9 9

Driver Chip Chart:

High

Meaning:

Driver Chip Chart:

High

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

Low

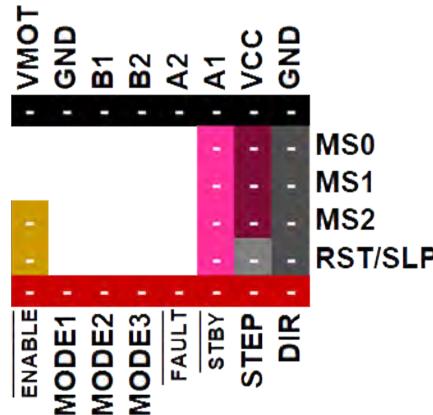
No Jumper Set

Stand-alone Mode

POLOLU TB67S249FTG

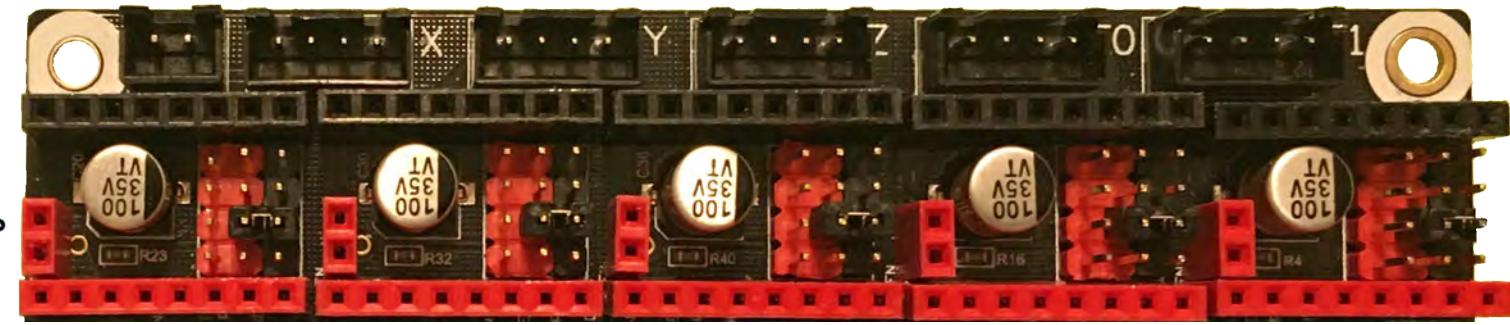
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

**STAND
BY**



See [Appendix D](#) for legend

STEP



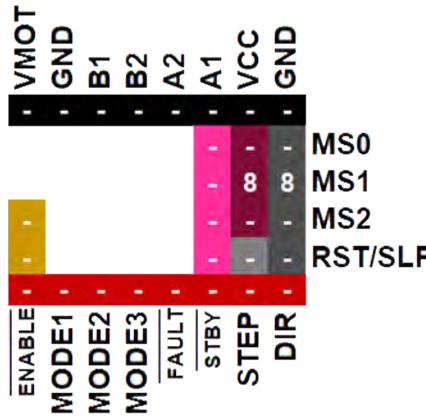
See [Appendix D](#) for legend

Stand-alone Mode

POLOLU TB67S249FTG

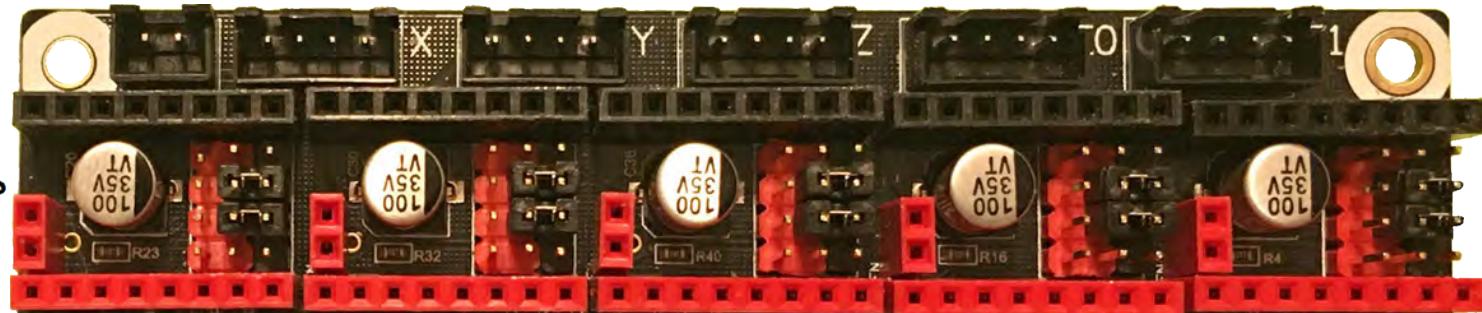
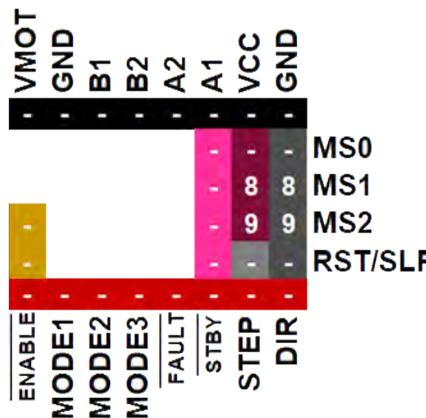
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

CCW
1 / 2



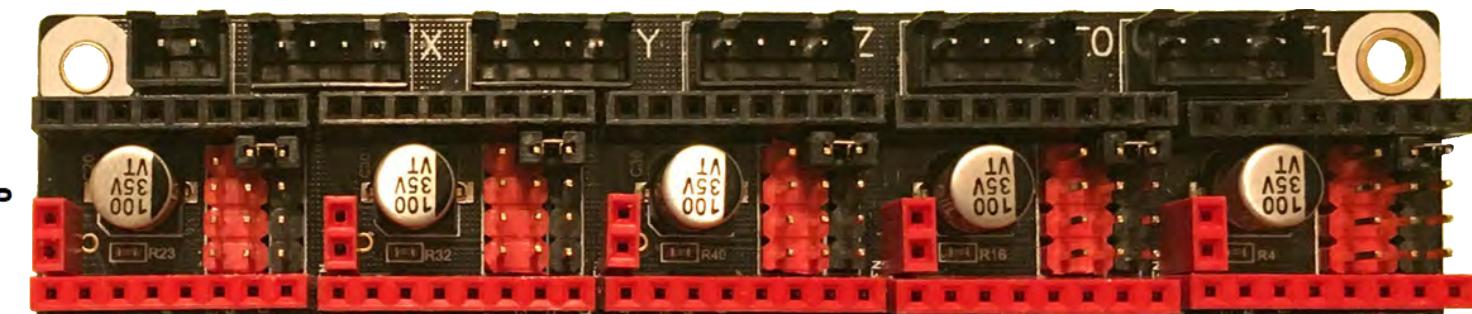
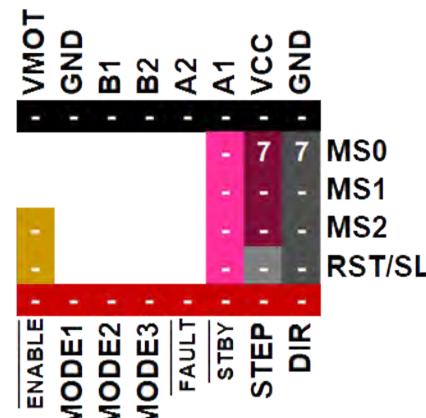
See [Appendix D](#) for legend

1 / 4



See [Appendix D](#) for legend

CW
1 / 2



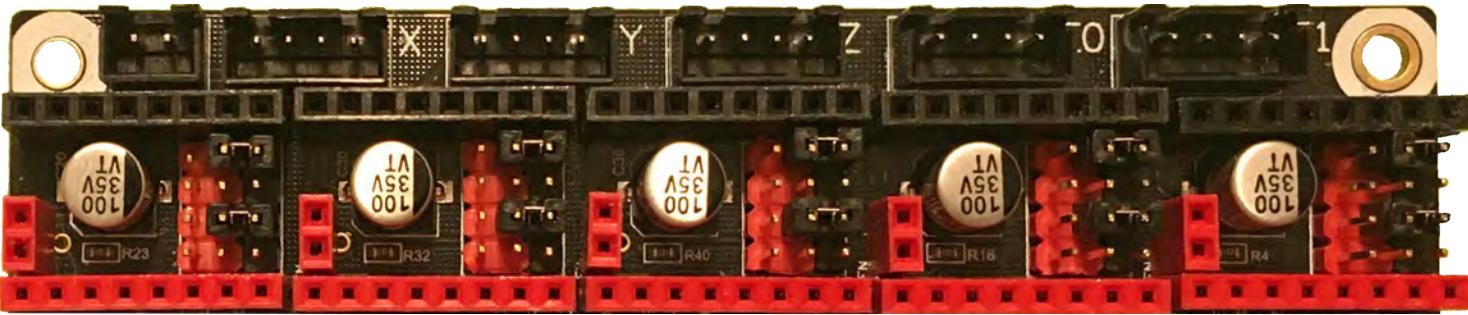
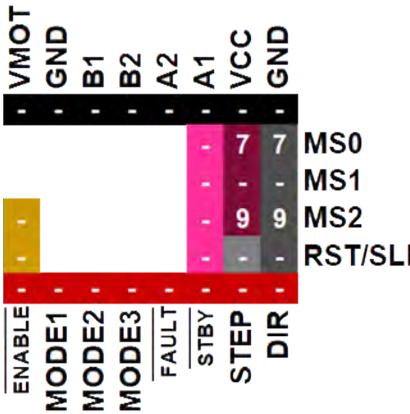
See [Appendix D](#) for legend

Stand-alone Mode

POLOLU TB67S249FTG

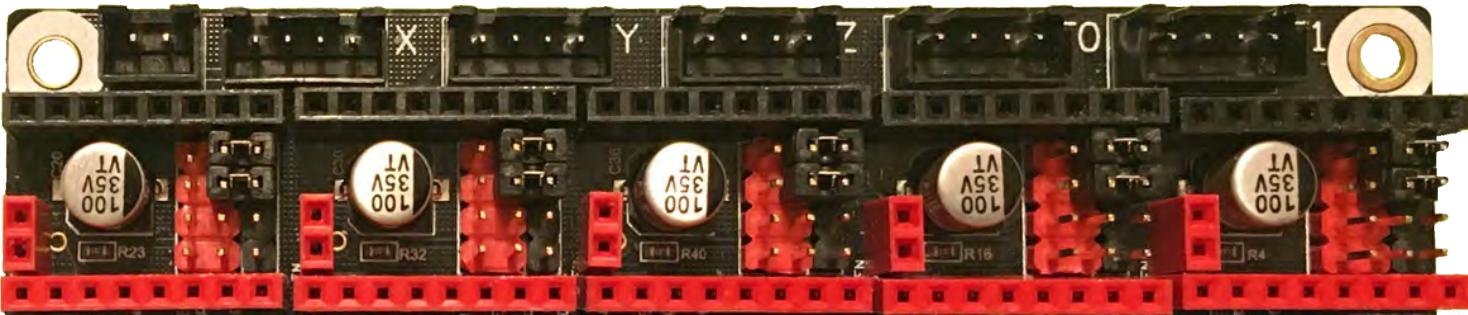
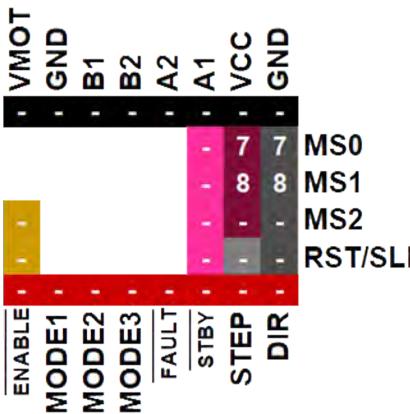
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

1 / 8



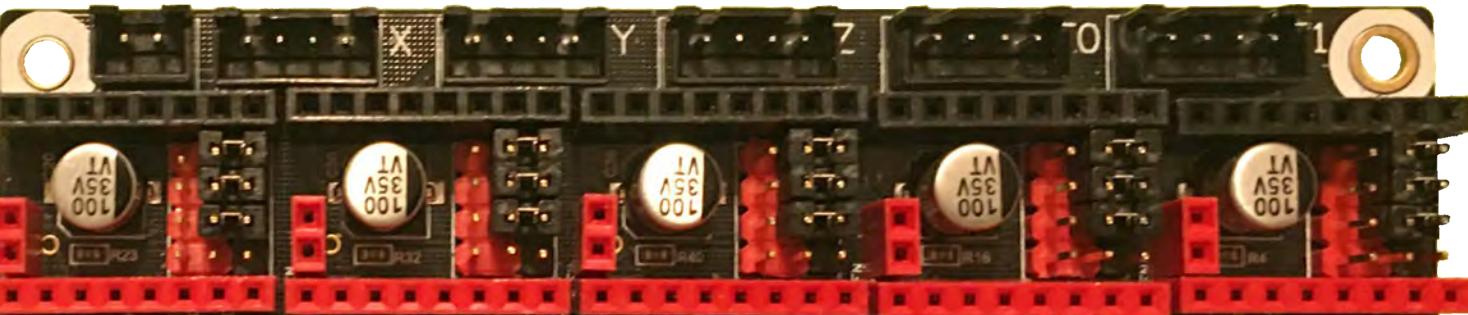
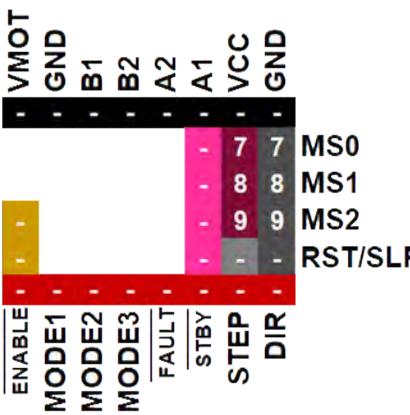
See [Appendix D](#) for legend

1 / 16



See [Appendix D](#) for legend

1 / 32



See [Appendix D](#) for legend

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

NOTE: [Go to Appendix C](#), and then come back here for the changes to Marlin for POLOLU TB67S249FTG stepper motor drivers.

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

The screenshot shows the Visual Studio Code interface with the Marlin 2.0.x repository open. The left sidebar displays the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X'. The main editor area shows the 'Configuration.h' file, specifically the section for stepper driver configuration. A green box highlights the line '#define E0_DRIVER_TYPE A4988'. The status bar at the bottom indicates the current line (Ln 686), column (Col 63), and other settings like spaces, tabs, and file type.

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

EXPLORER PIO Home Configuration.h Configuration_adv.h

OPEN EDITORS
  PIO Home
  Configuration.h M...
  Configuration_adv.h
MARLIN-2.0.X
  Lcdprint.cpp
  Lcdprint.h
  thermistornames.h
  ultralcd.cpp
  ultralcd.h
  > libs
  > module
  > pins
  > sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md

OUTLINE

Marlin > Configuration.h > E0_DRIVER_TYPE

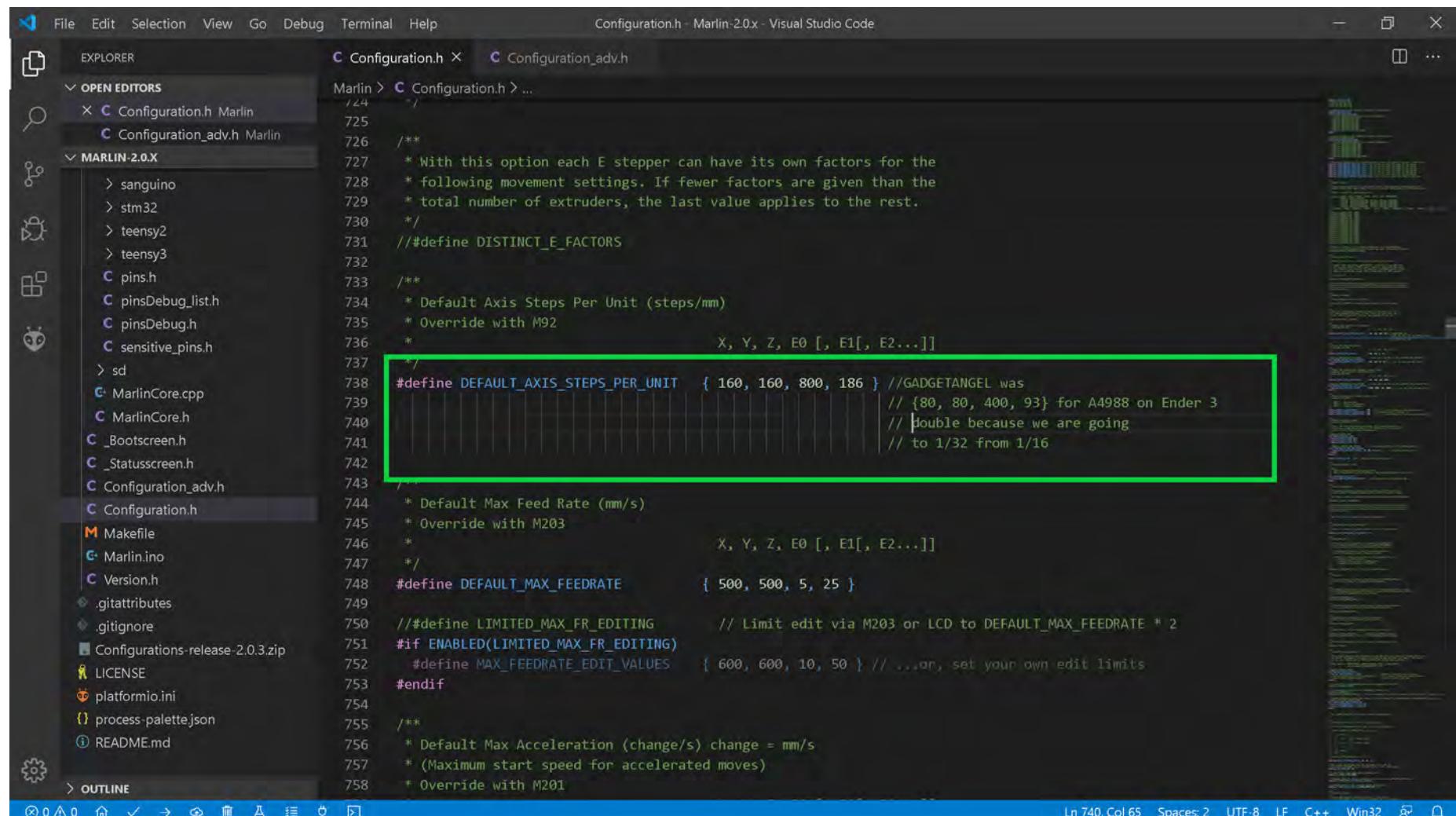
661
662 /**
663 * Stepper Drivers
664 *
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670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

Ln 686 Col 63 Spaces: 2 UTF-8 LF C++ Win32
```

- Go to the next page.

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

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



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

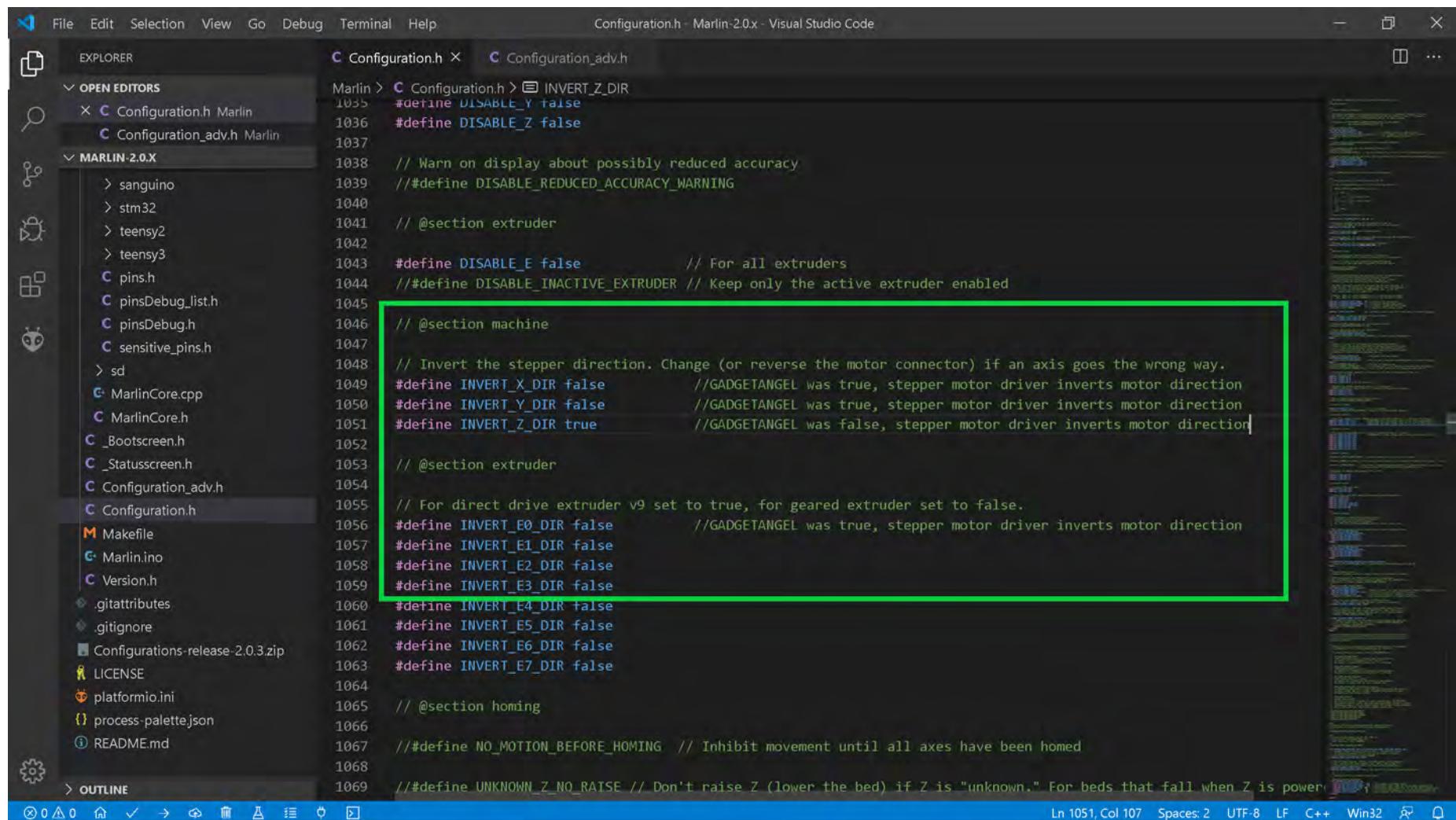
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// double because we are going
// to 1/32 from 1/16
```

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

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

C Configuration.h x C Configuration_adv.h
Marlin > C Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 //##define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 //##define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         //GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       //GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 //##define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 //##define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

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

- Go to the next page.

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

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

```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER OPEN EDITORS Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
===== [SUCCESS] Took 130.61 seconds =====
Environment Status Duration
-----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_debug IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

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

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

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

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

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin
MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
TIMELINE

```

```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

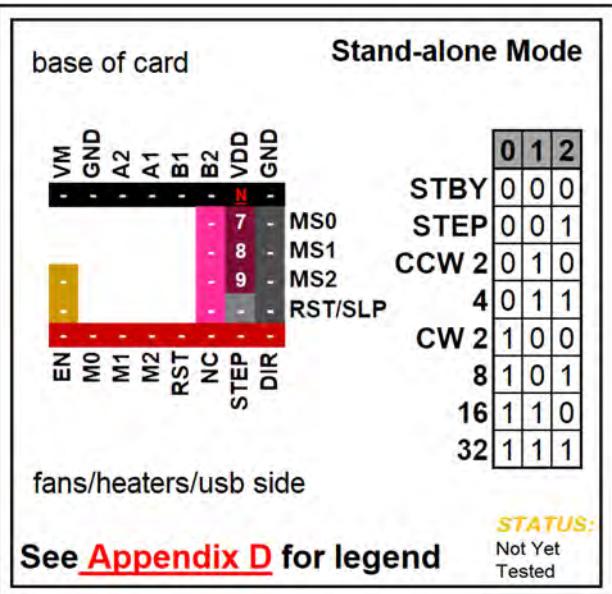
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET_3D	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

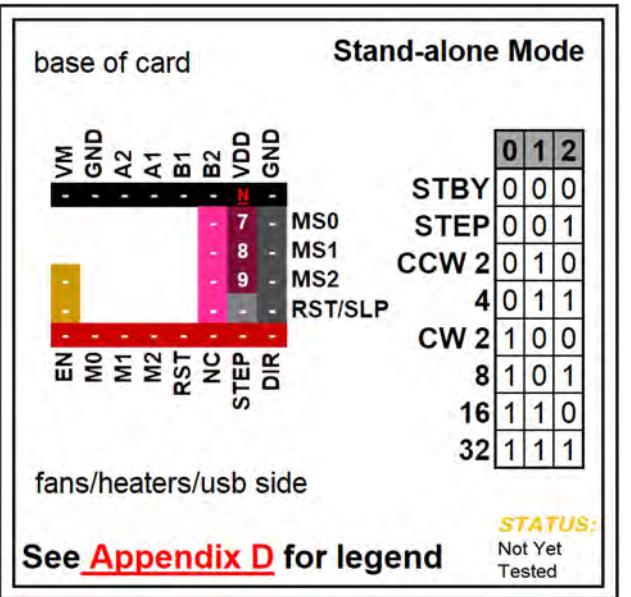
BIQU S109

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

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

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

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

**Driver Chip Chart:**

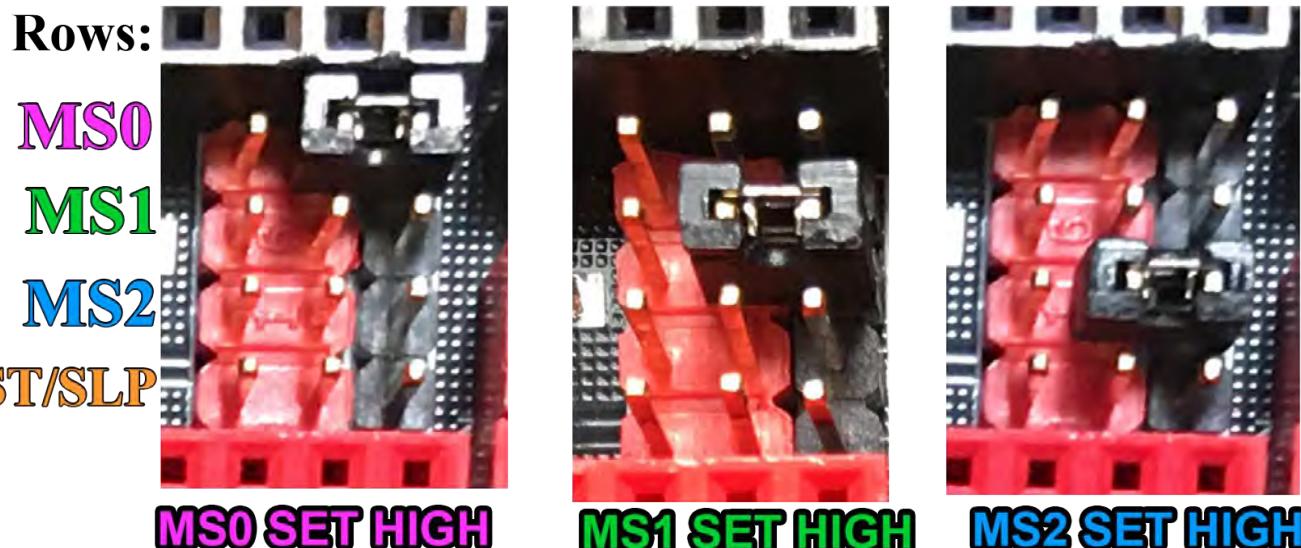
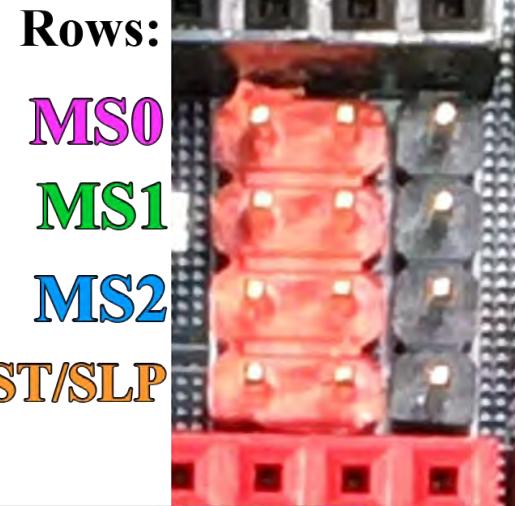
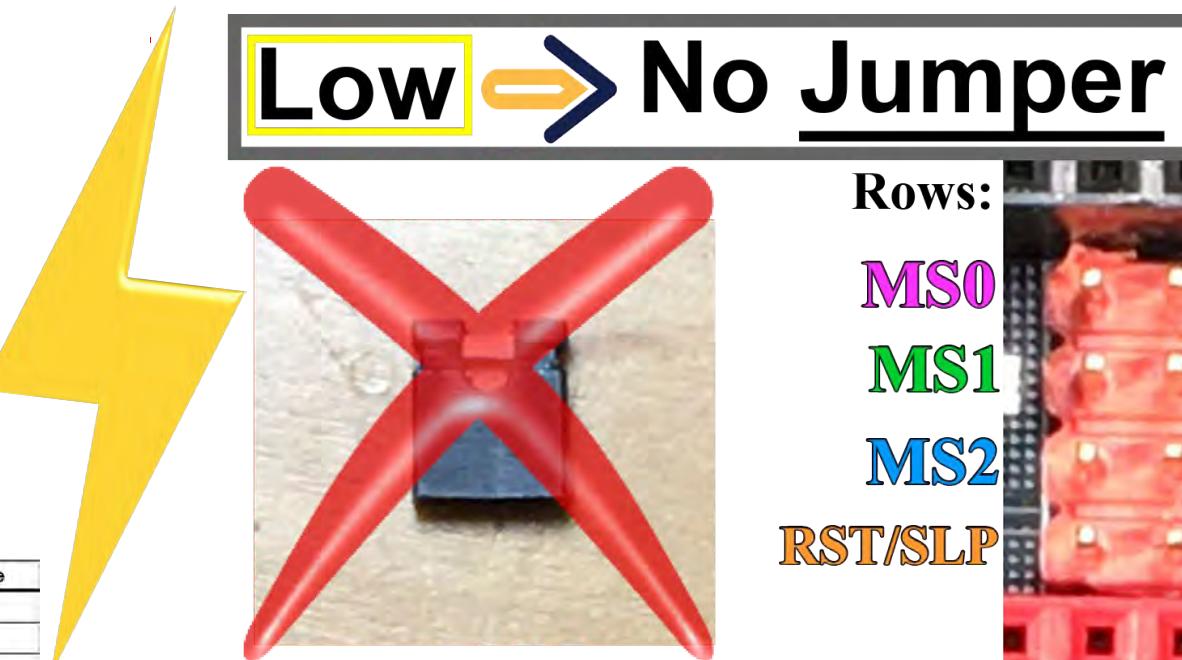
Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XX V DC	High	High	Low	1/8 Step	2W1-2 Phase
xxA (peak)	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32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

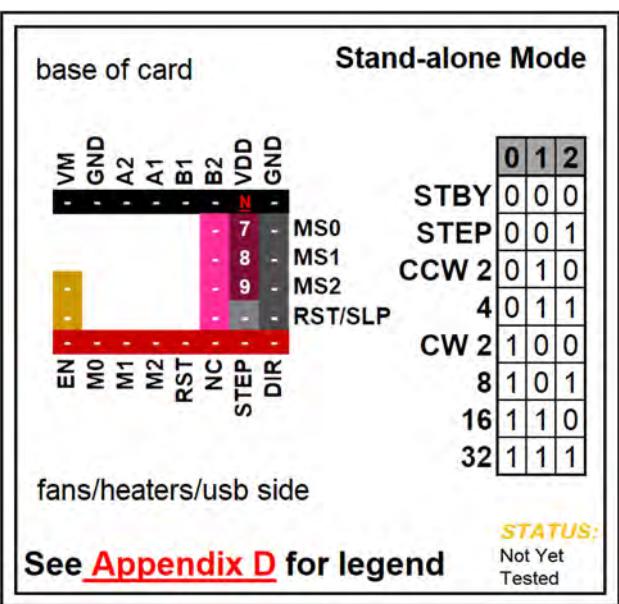
Driving Current Calculation Formula $I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

$R_S(\text{Typical Sense Resistor}) = X.XX \Omega$

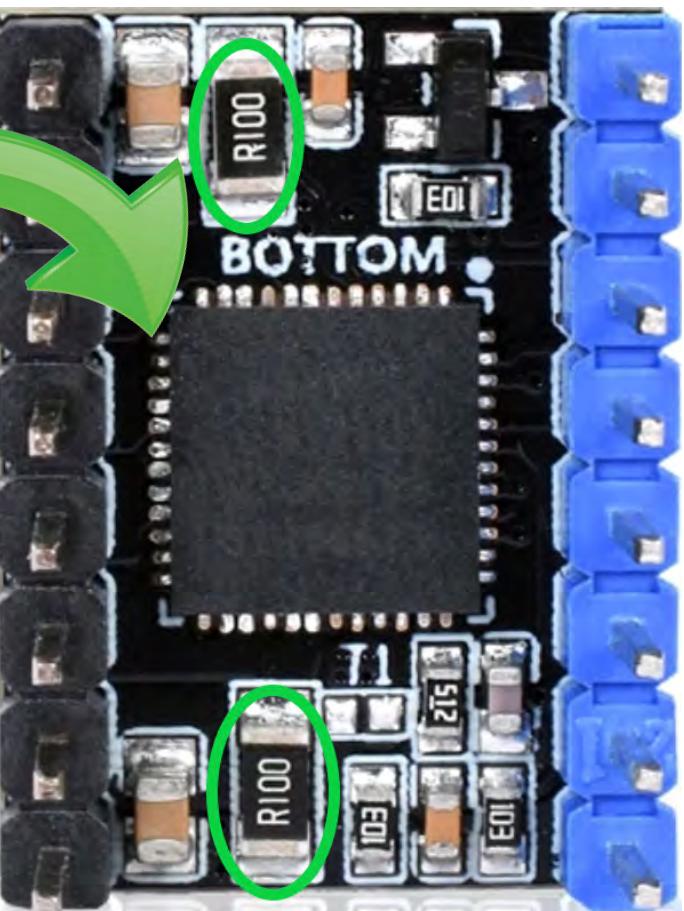
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

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

BINU S109**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**



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



BIQU S109

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

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



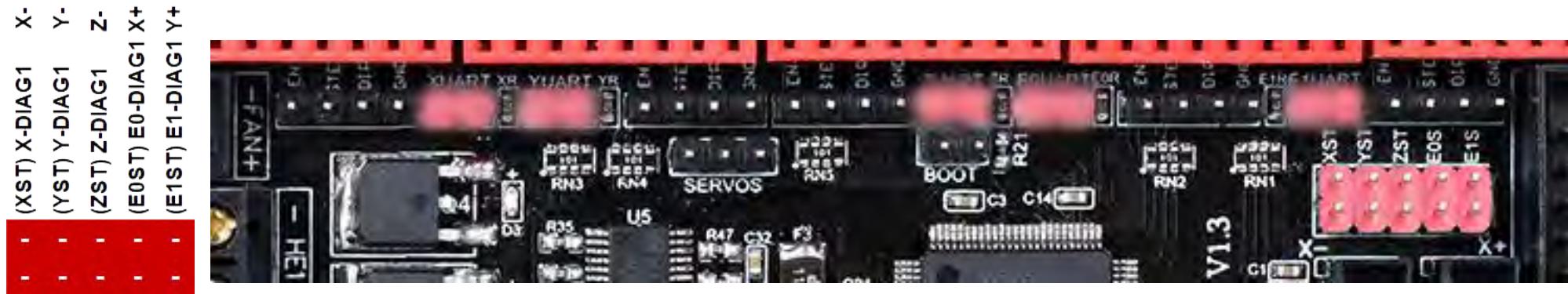
- $R_s = R050$ is 0.05 Ohms
- $R_s = R068$ is 0.068 Ohms
- $R_s = R100$ is 0.1 Ohms
- $R_s = R150$ is 0.15 Ohms
- $R_s = R200$ is 0.2 Ohms
- $R_s = R220$ is 0.22 Ohms

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

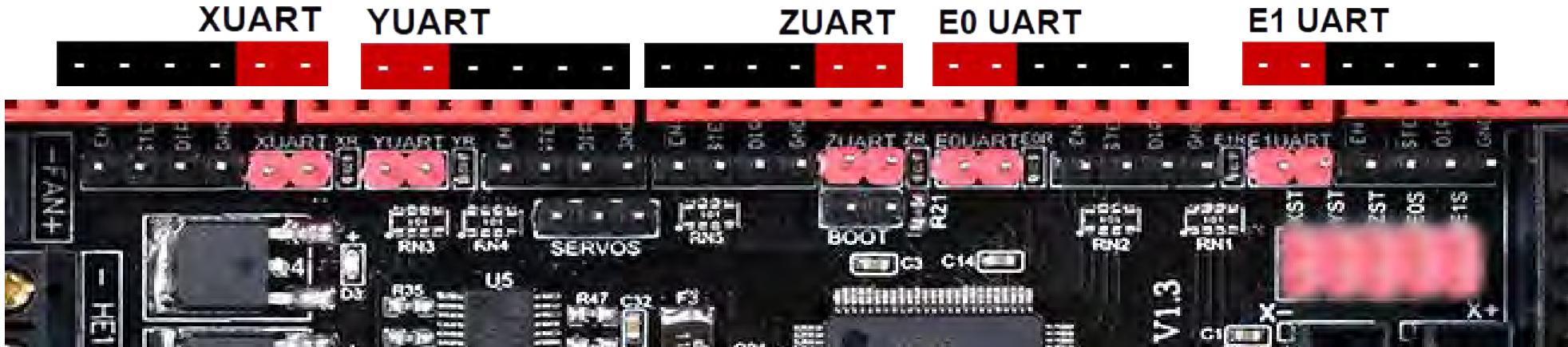
Stand-alone Mode

BIQU S109

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



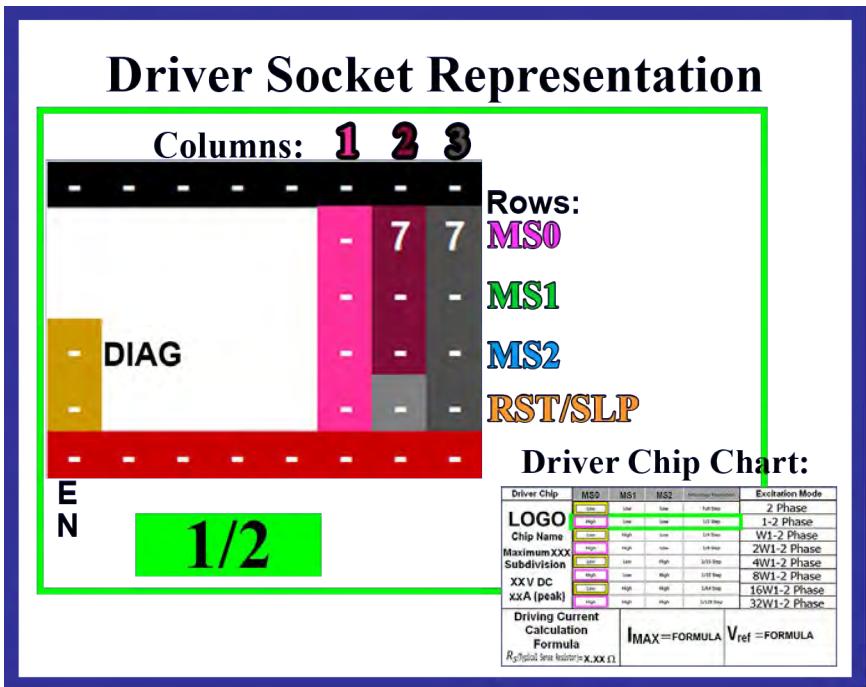
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



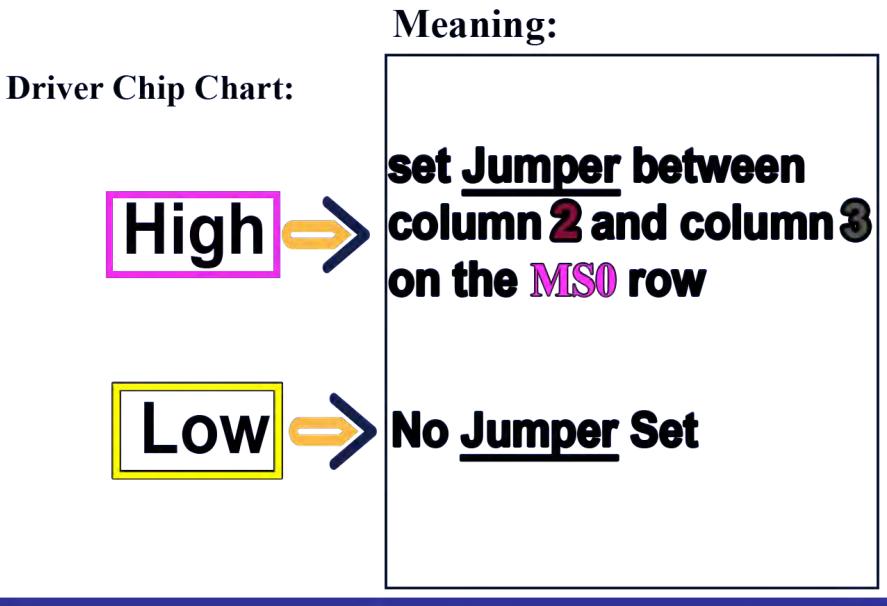
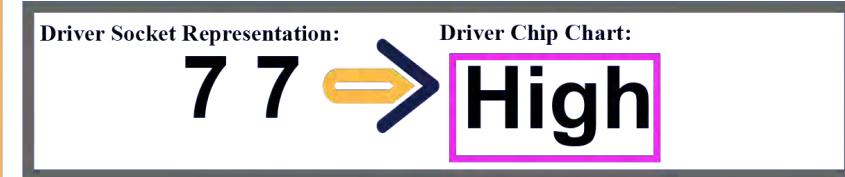
Stand-alone Mode

BINU S109

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

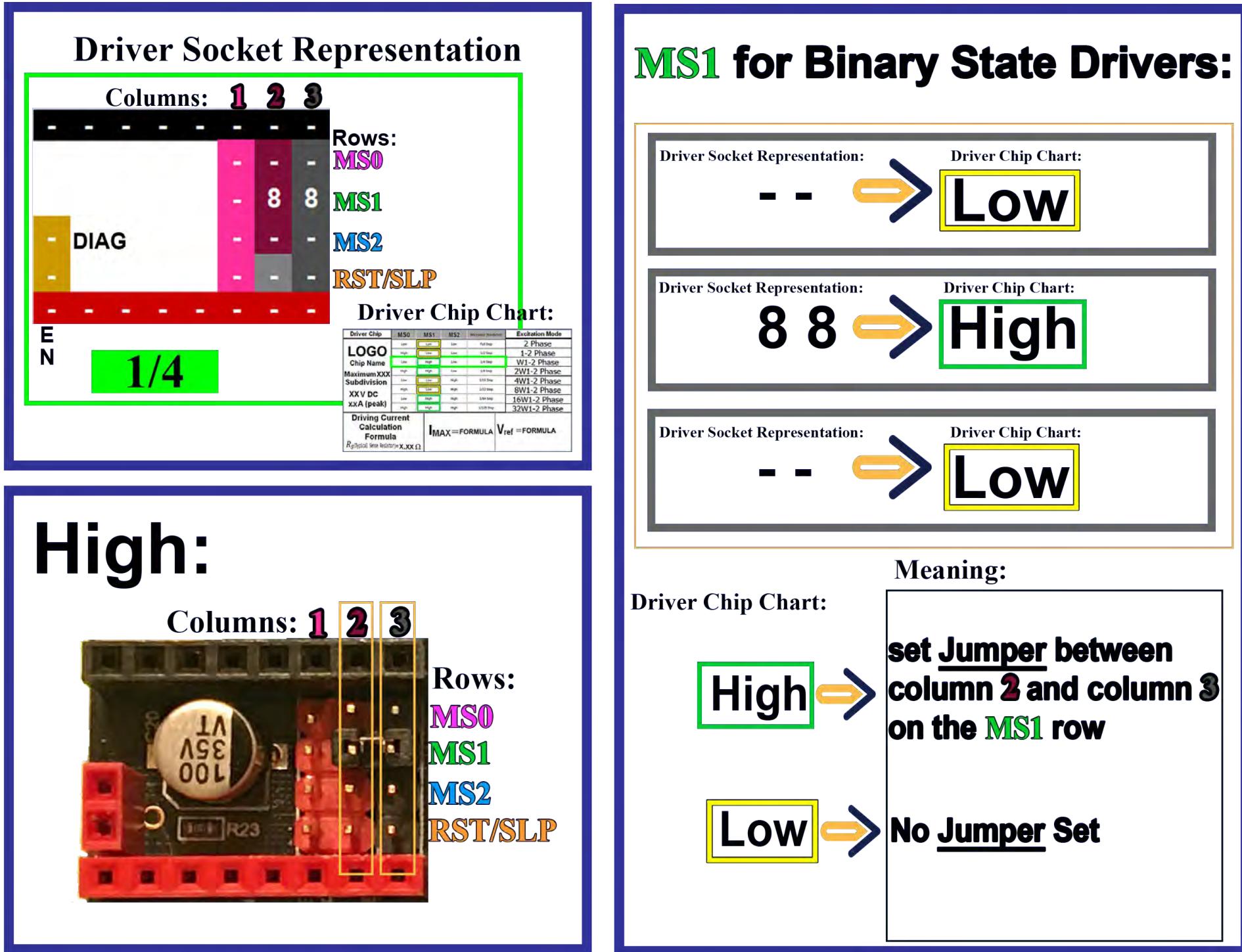
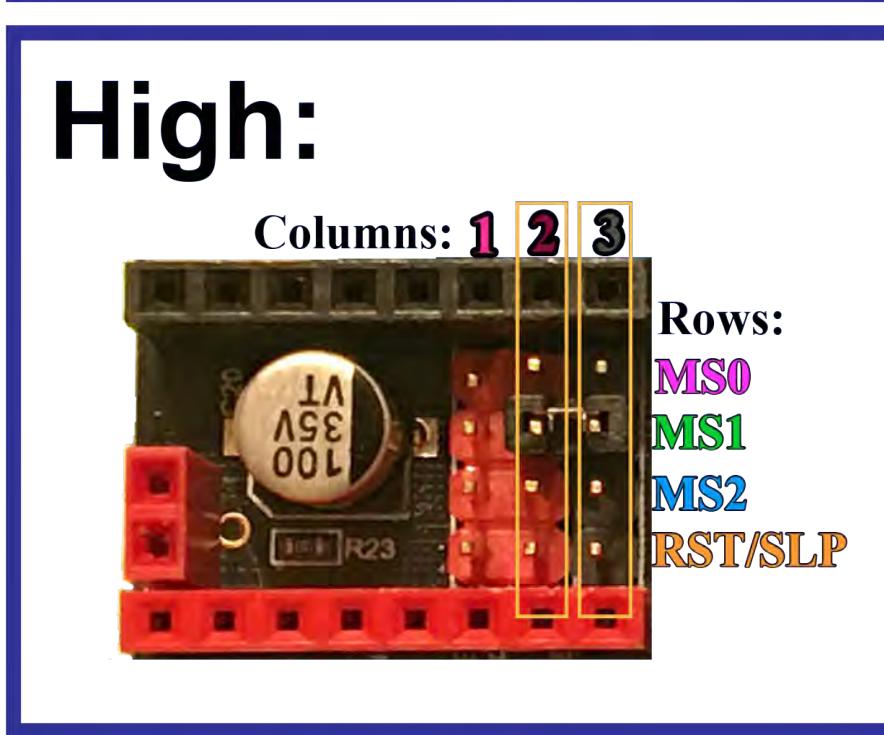
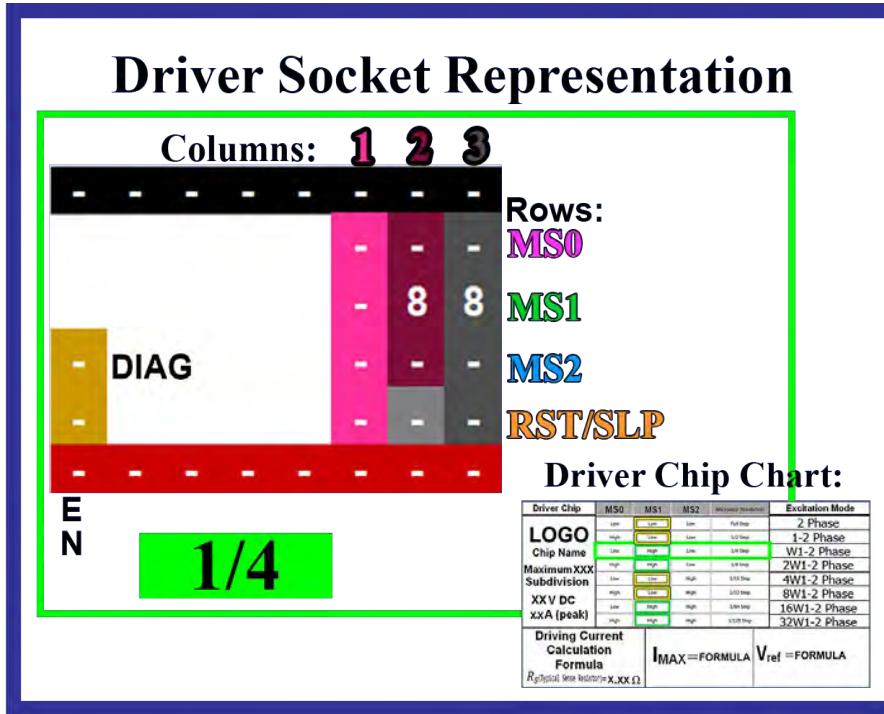


MS0 for Binary State Drivers:



Stand-alone Mode

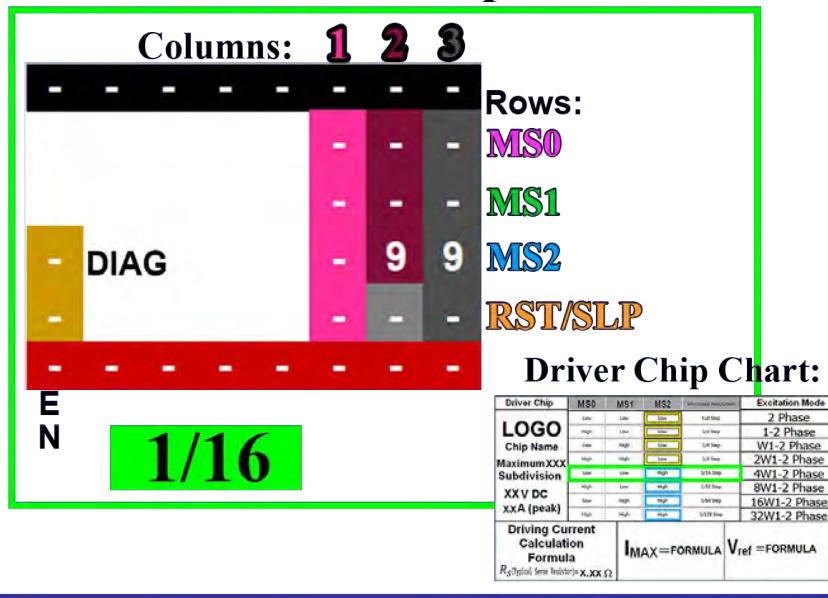
BQU S109



Stand-alone Mode

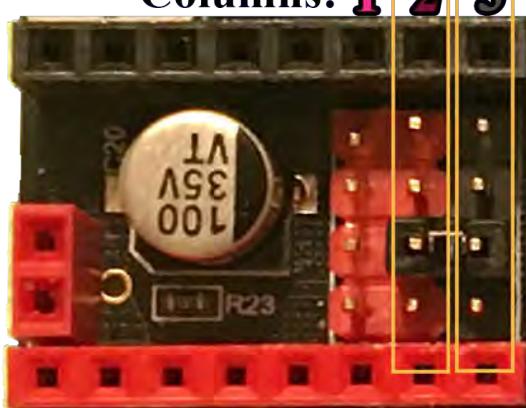
Biqu S109

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:
Driver Chip Chart:

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

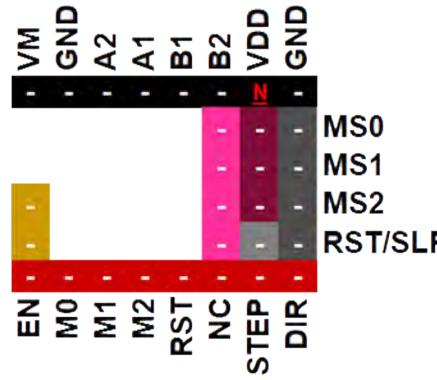
Low → **No Jumper Set**

Stand-alone Mode

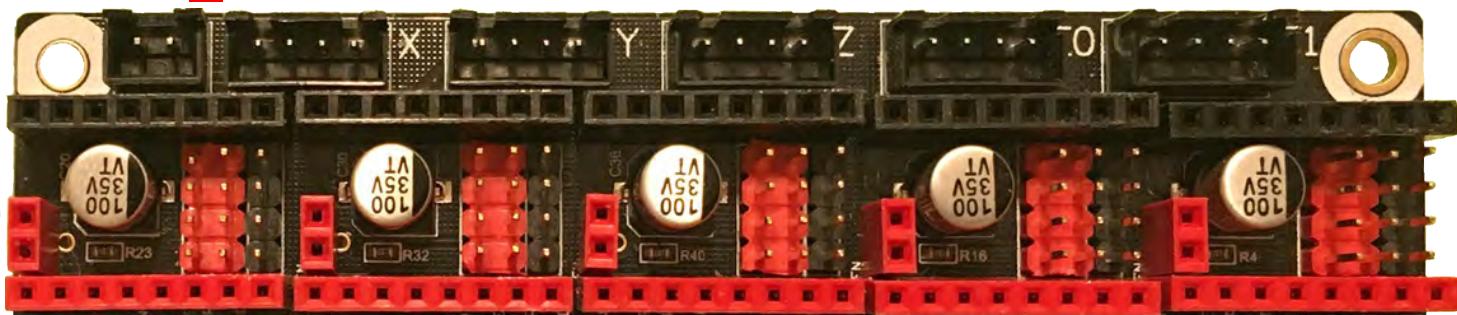
BIQU S109

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

**STAND
BY**

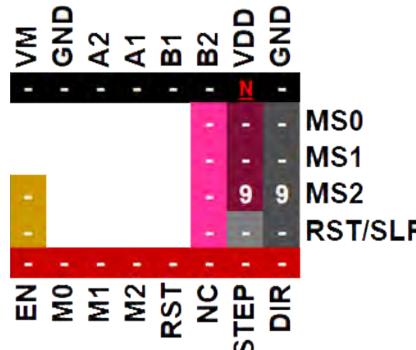


Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



See [Appendix D](#) for legend

STEP



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



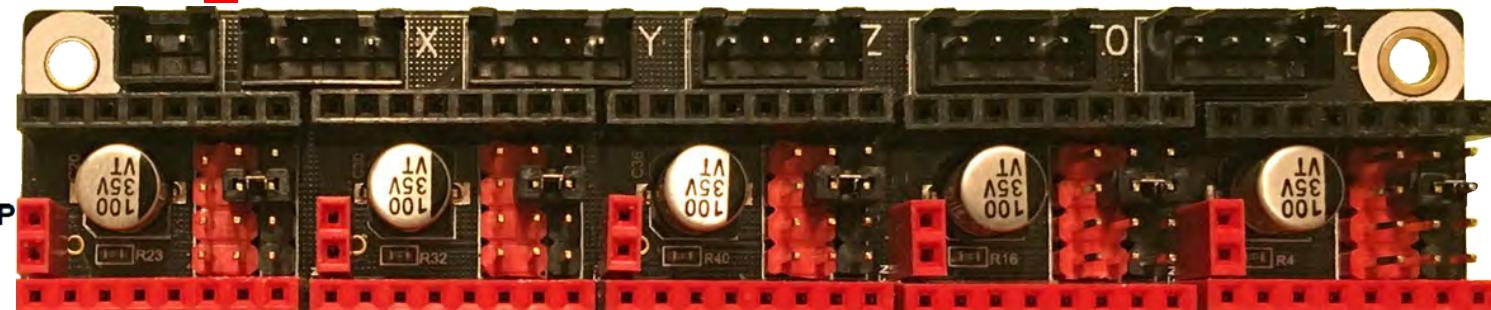
See [Appendix D](#) for legend

Stand-alone Mode

BINU S109

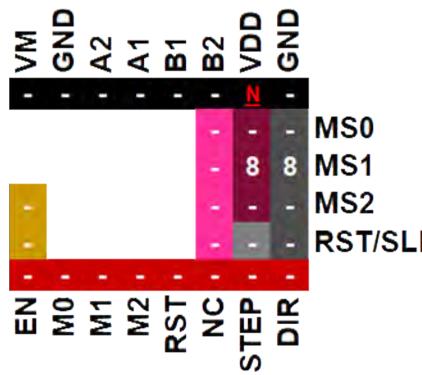
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

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

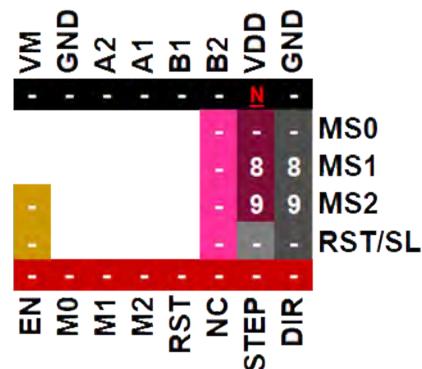


See [Appendix D](#) for legend

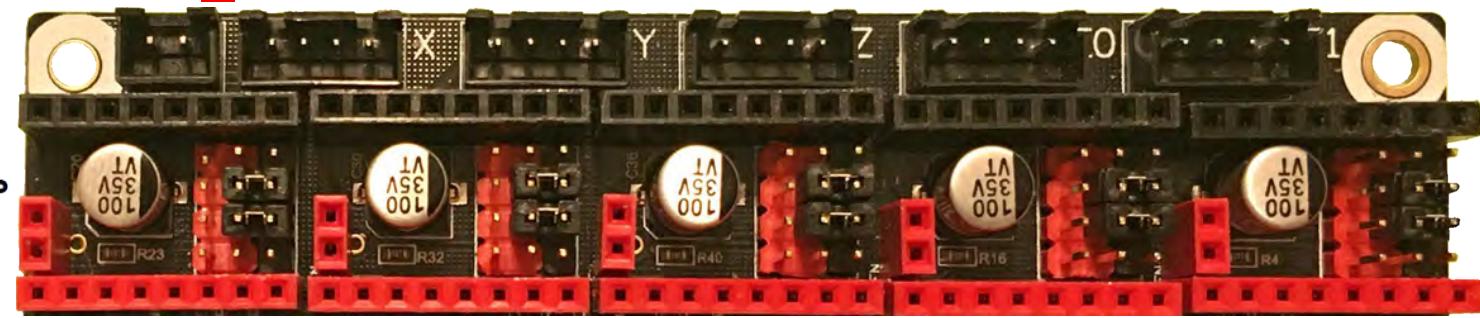
CCW
1 / 2



1 / 4

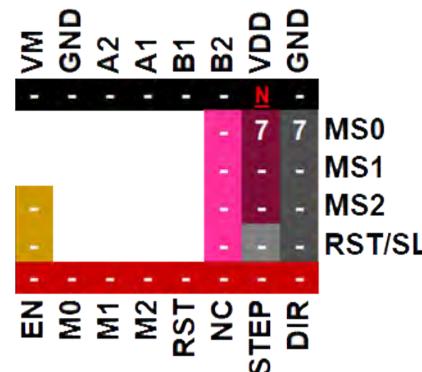


Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

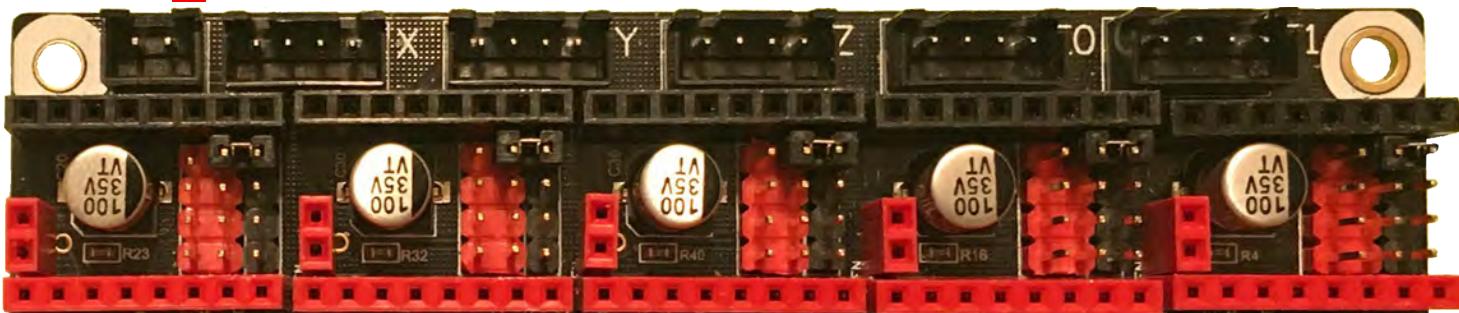


See [Appendix D](#) for legend

CW
1 / 2



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



See [Appendix D](#) for legend

Stand-alone Mode

BQU S109

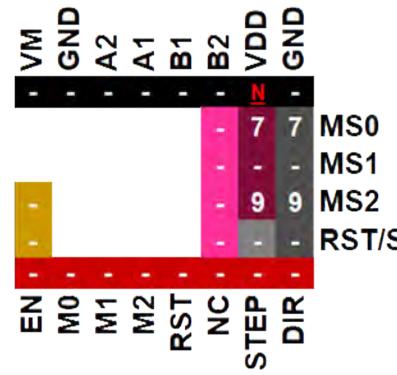
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

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

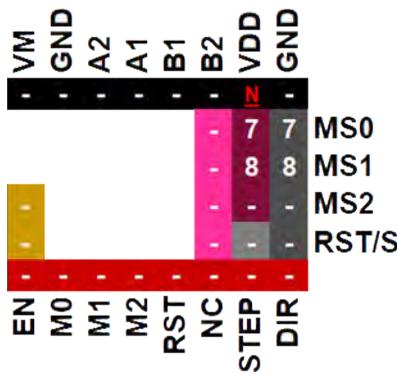


See [Appendix D](#) for legend

1 / 8



1 / 16

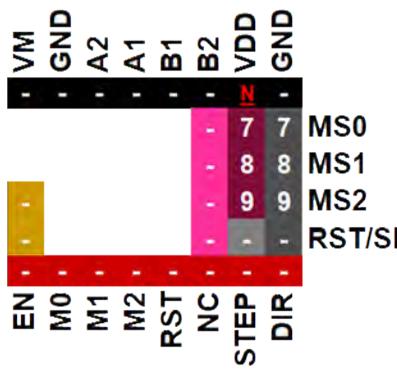


Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



See [Appendix D](#) for legend

1 / 32



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

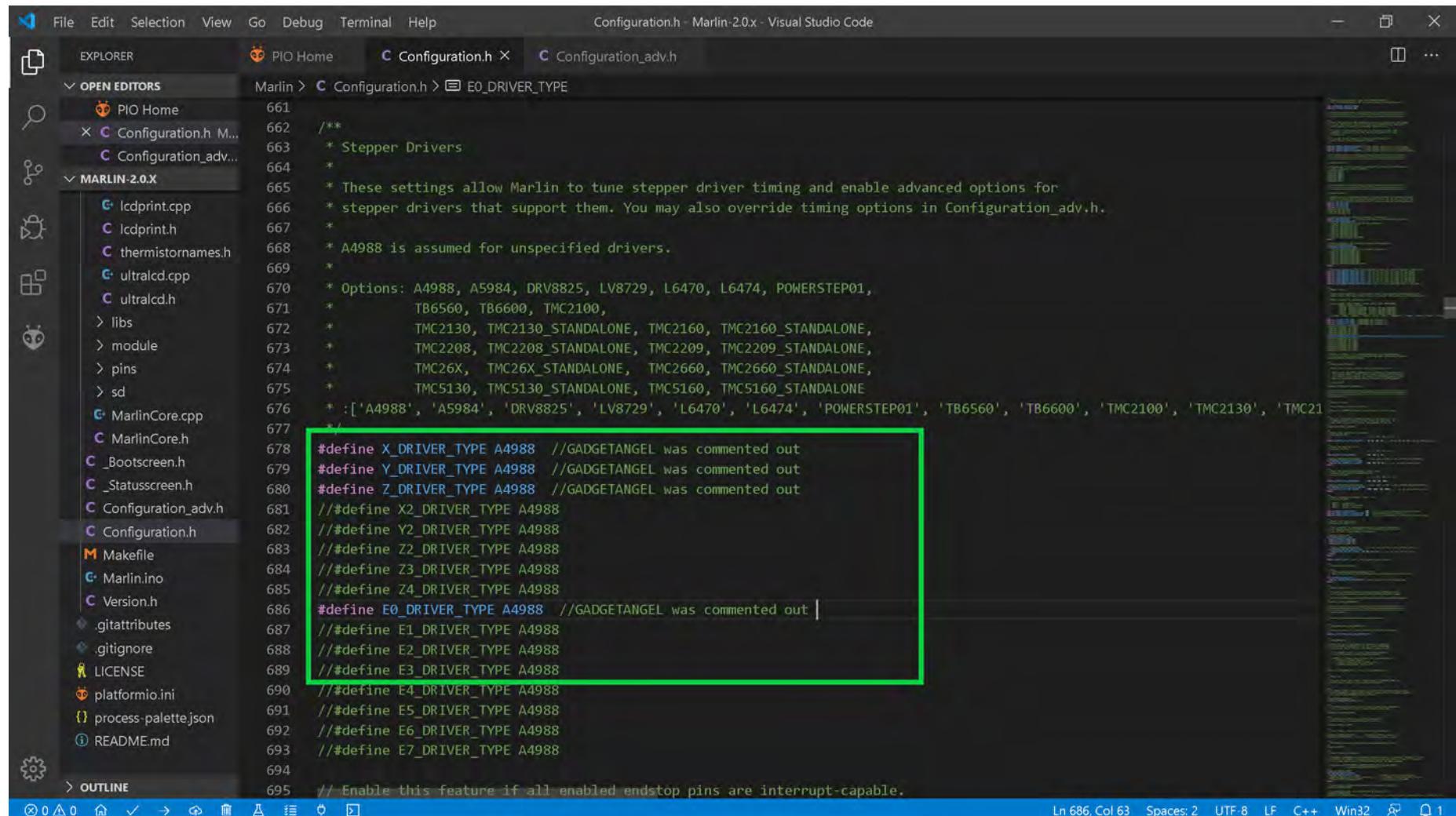


See [Appendix D](#) for legend

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

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

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



```

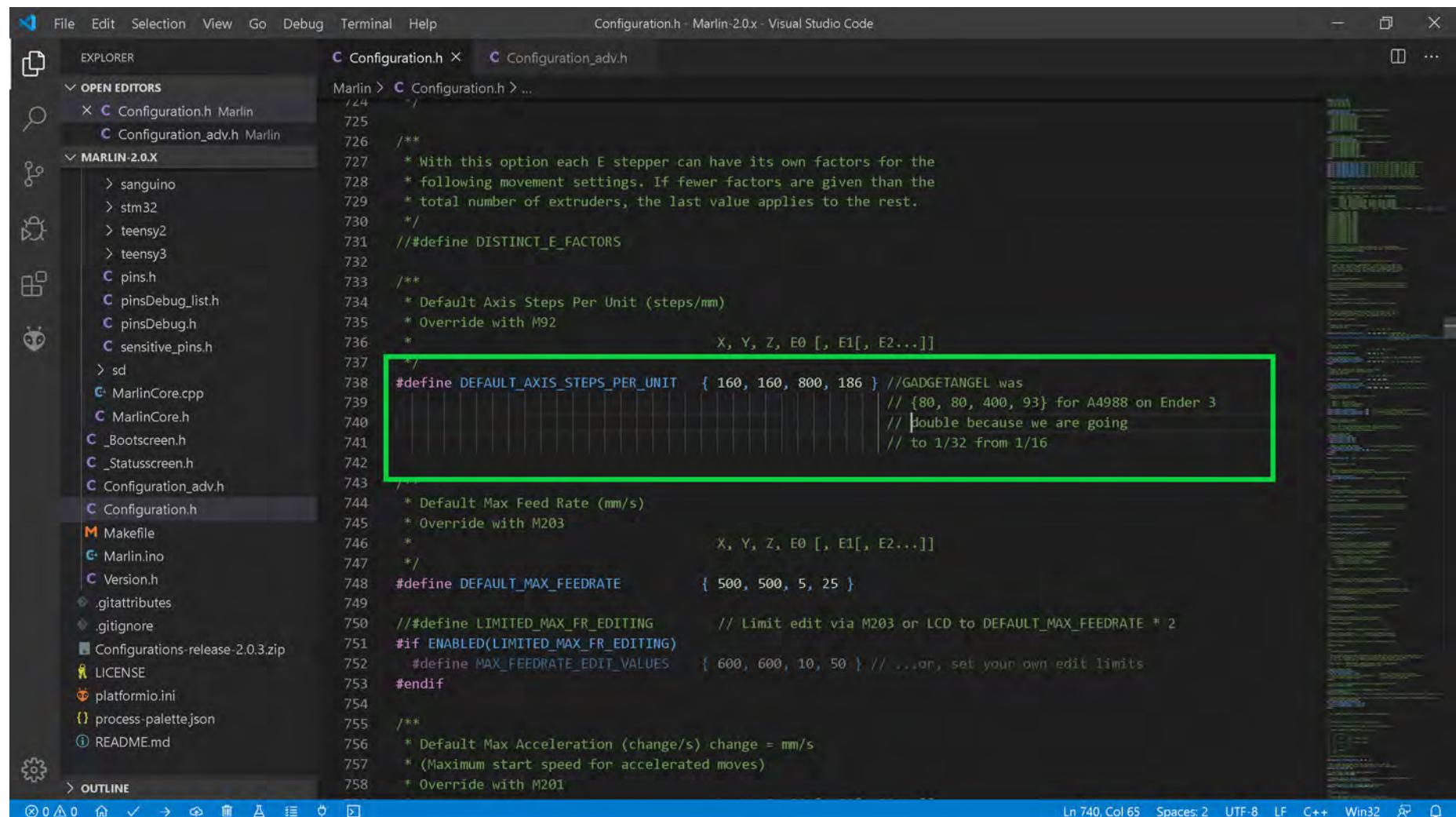
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home Configuration.h M... Configuration_adv.h
MARLIN-2.0.X
Lcdprint.cpp Lcdprint.h thermistornames.h
ultralcd.cpp ultralcd.h
libs module pins sd
MarlinCore.cpp MarlinCore.h _Bootscreen.h _Statusscreen.h
Configuration_adv.h Configuration.h Makefile Marlin.ino Version.h
.gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md
> OUTLINE
Ln 686, Col 63 Spaces: 2 UTF-8 LF C++ Win32 1
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2
676
677
678 #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
679 #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
680 #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out |
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



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

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

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

- Go to the next page.

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

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

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin 1035 #define DISABLE_Y false
  Configuration_adv.h Marlin 1036 #define DISABLE_Z false
  1037
  1038 // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  1041 // @section extruder
  1042
  1043 #define DISABLE_E false          // For all extruders
  1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1045
  1046 // @section machine
  1047
  1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1052
  1053 // @section extruder
  1054
  1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
  1056 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1057 #define INVERT_E1_DIR false
  1058 #define INVERT_E2_DIR false
  1059 #define INVERT_E3_DIR false
  1060 #define INVERT_E4_DIR false
  1061 #define INVERT_E5_DIR false
  1062 #define INVERT_E6_DIR false
  1063 #define INVERT_E7_DIR false
  1064
  1065 // @section homing
  1066
  1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1068
  1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- Go to the next page.

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

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

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

File Edit Selection View Go Run Terminal Help

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src... pins_BTT_SKR_common.h Marlin... Configuration_adv.h Marlin

samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h _Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

OUTLINE TIMELINE

Configuration.h Configuration.h ...

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132#endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin

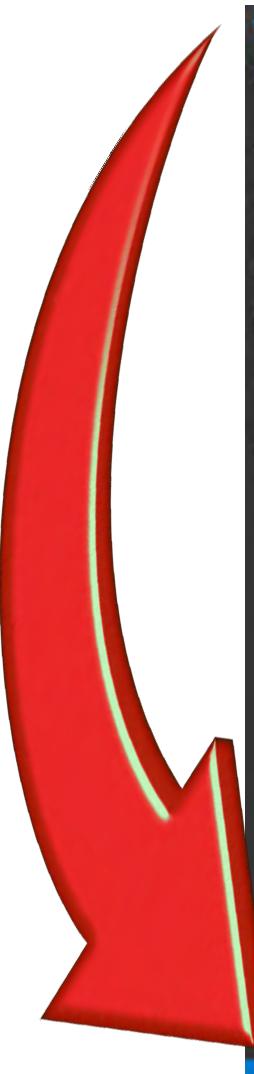
===== [SUCCESS] Took 130.61 seconds =====

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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




Screenshot of Visual Studio Code showing the Marlin setup process:

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

- File Explorer:** Shows the project structure under "OPEN EDITORS" and "MARLIN-2.0.X".
- Code Editor:** Displays the "Configuration.h" file with the following code snippet highlighted by a green box:

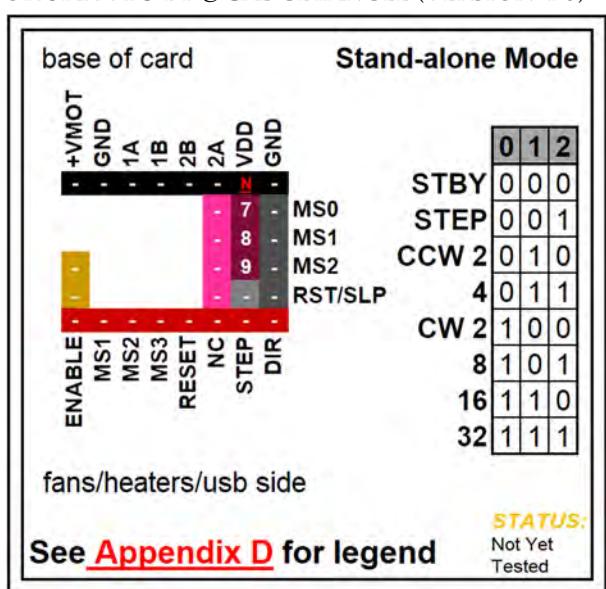

```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif
```
- Terminal:** Shows the compilation process and success message:


```
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
```

[SUCCESS] Took 130.61 seconds
- Environment Table:** Lists various boards and their status:

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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



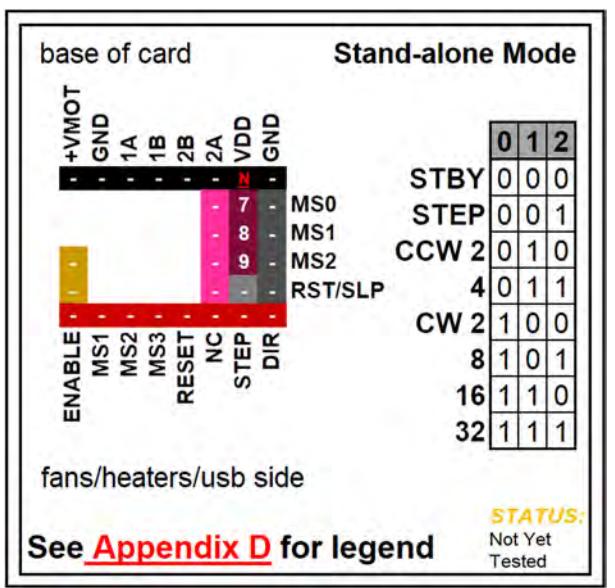
FYSETC S109

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

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

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

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



Driver Chip Chart:

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

Driving Current Calculation Formula $I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$
 $R_s(\text{Typical Sense Resistor}) = X.XX \Omega$

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

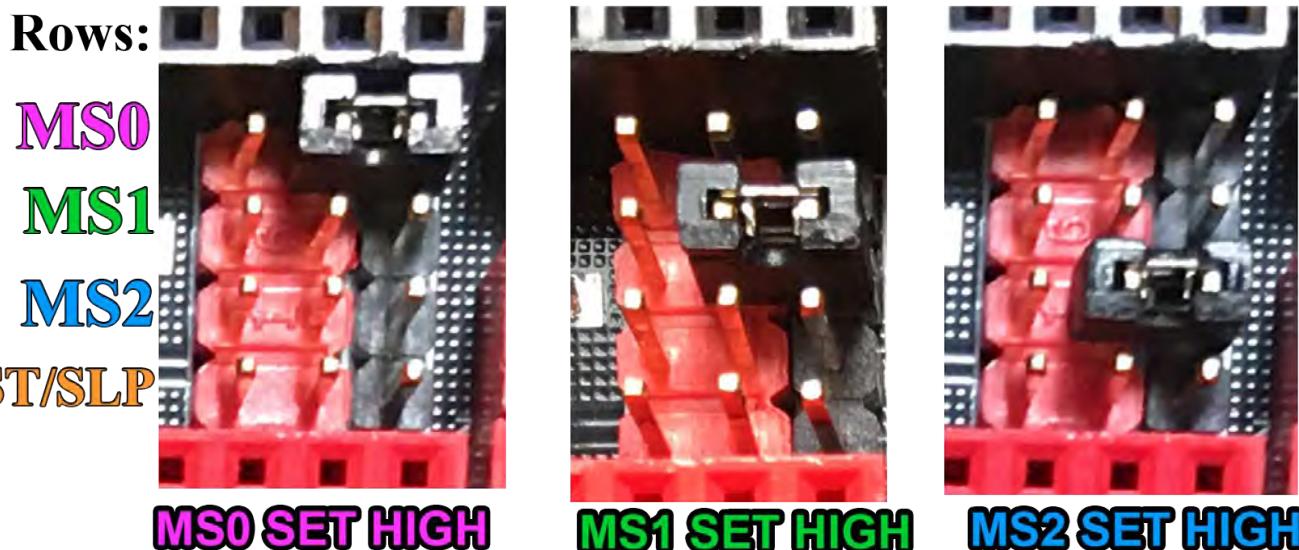
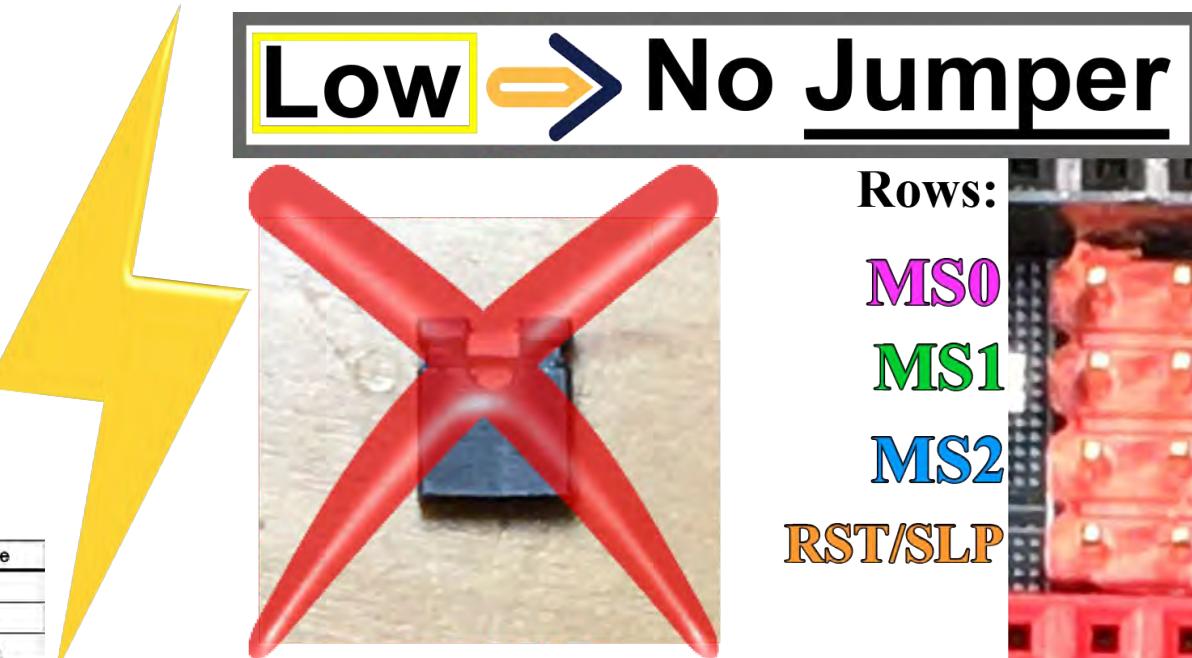


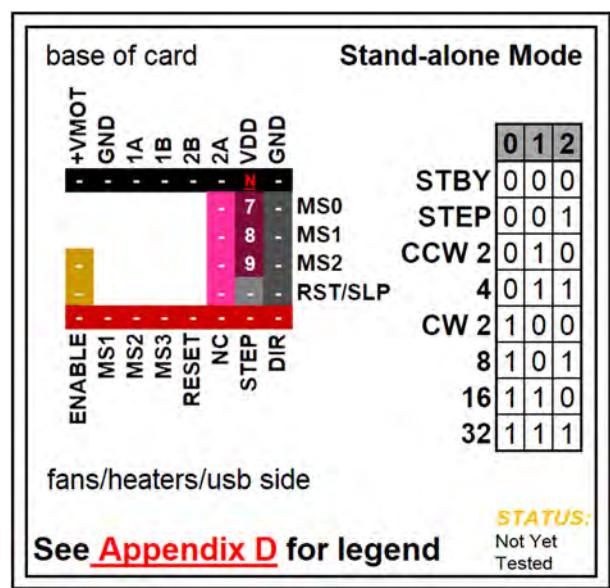
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):



FYSETC S109

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



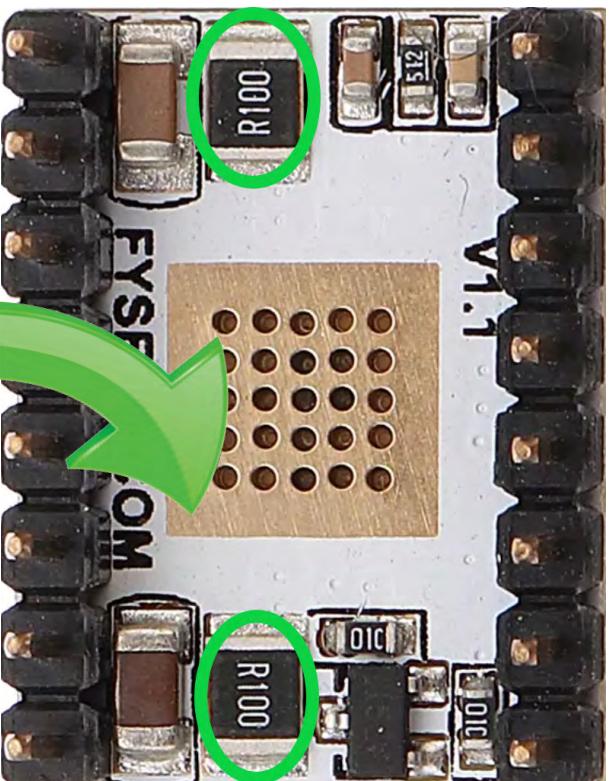


FYSETC S109

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

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

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



$R_s = R050$ is 0.05 Ohms

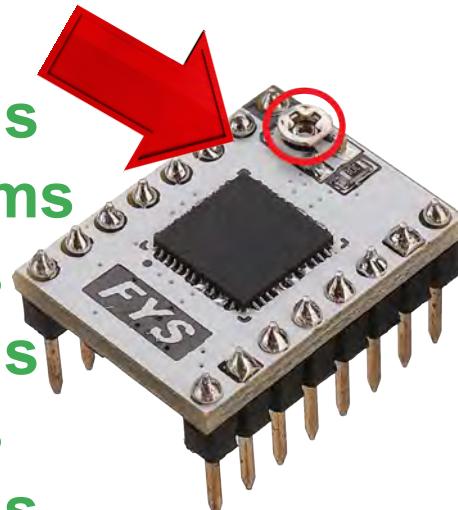
$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

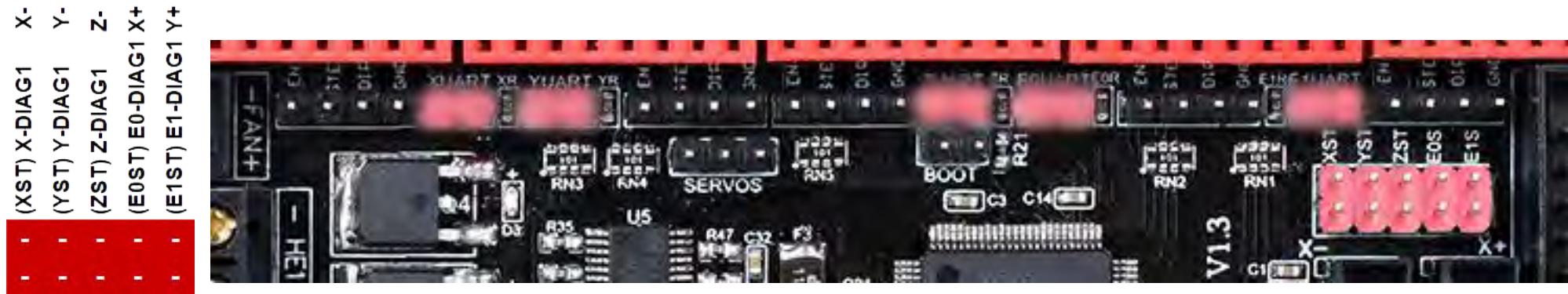


Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

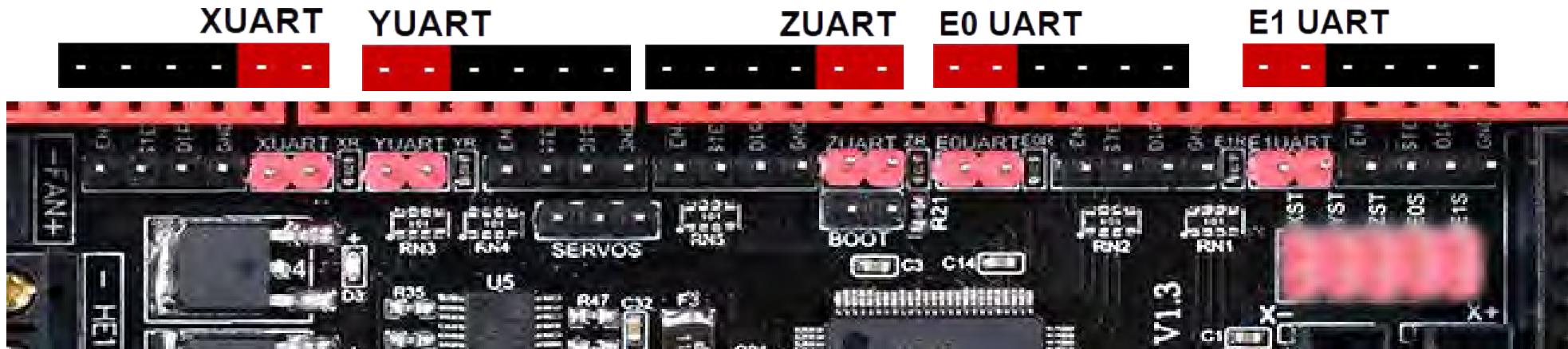
Stand-alone Mode

FYSETC S109

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



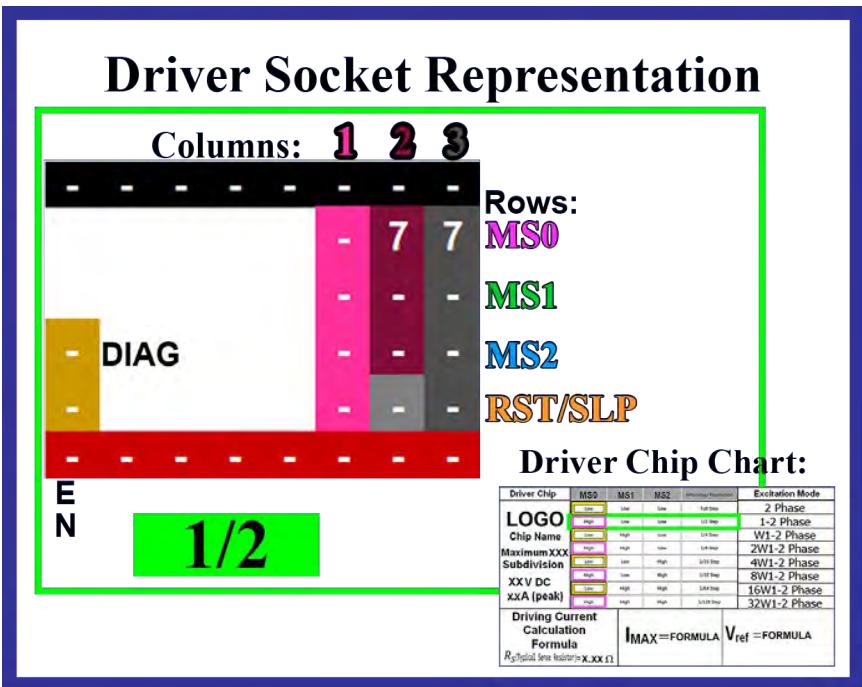
Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



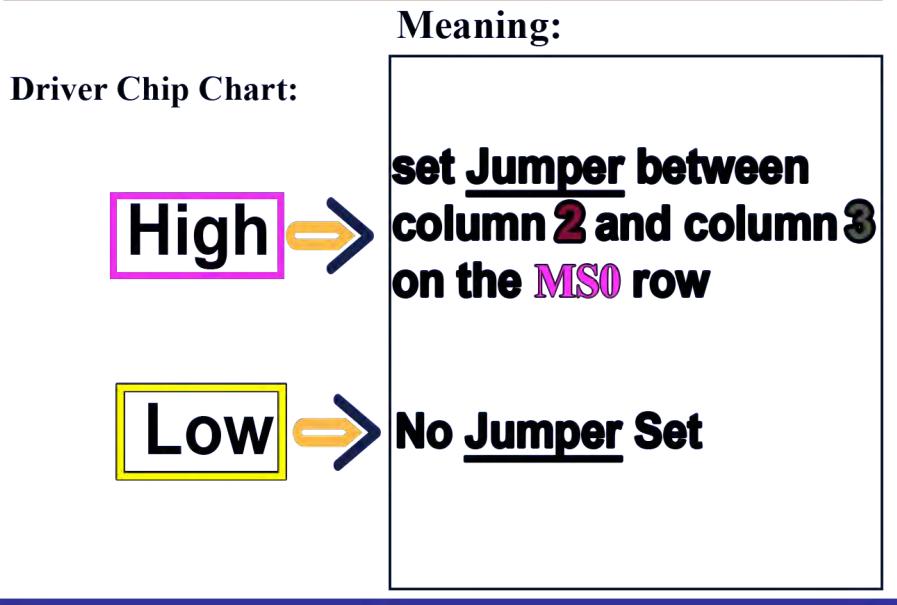
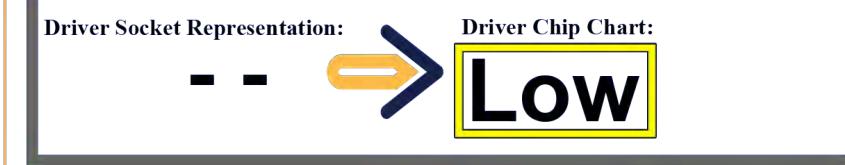
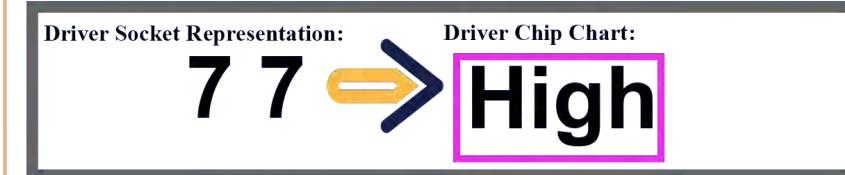
Stand-alone Mode

FYSETC S109

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

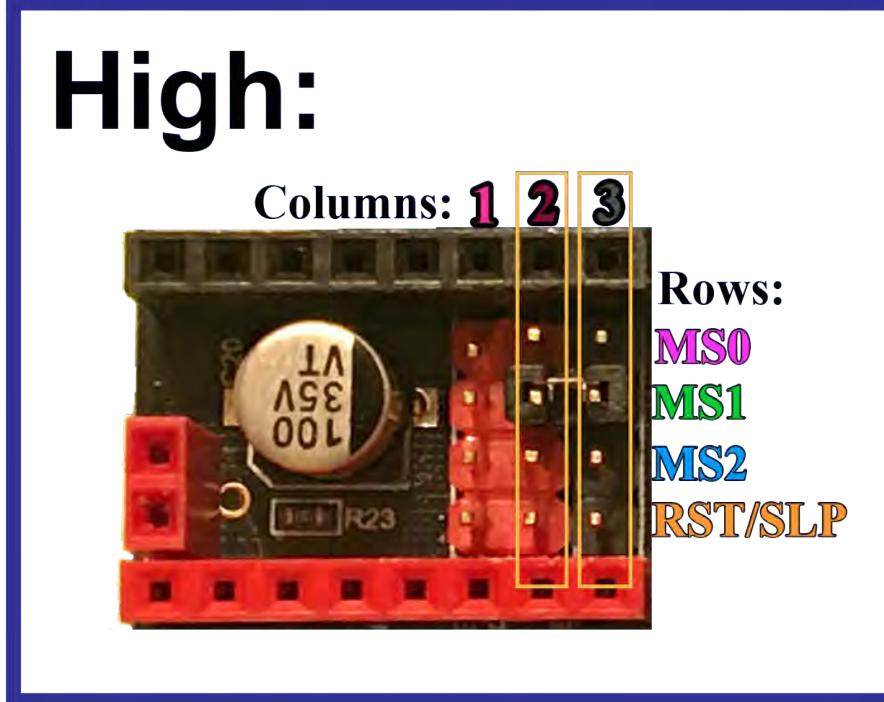
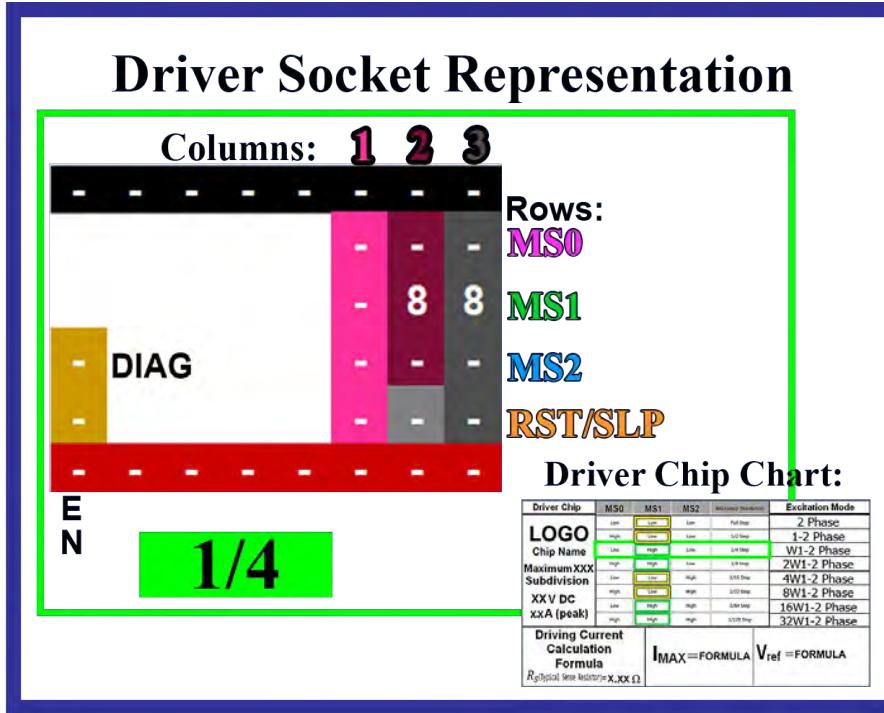


MS0 for Binary State Drivers:

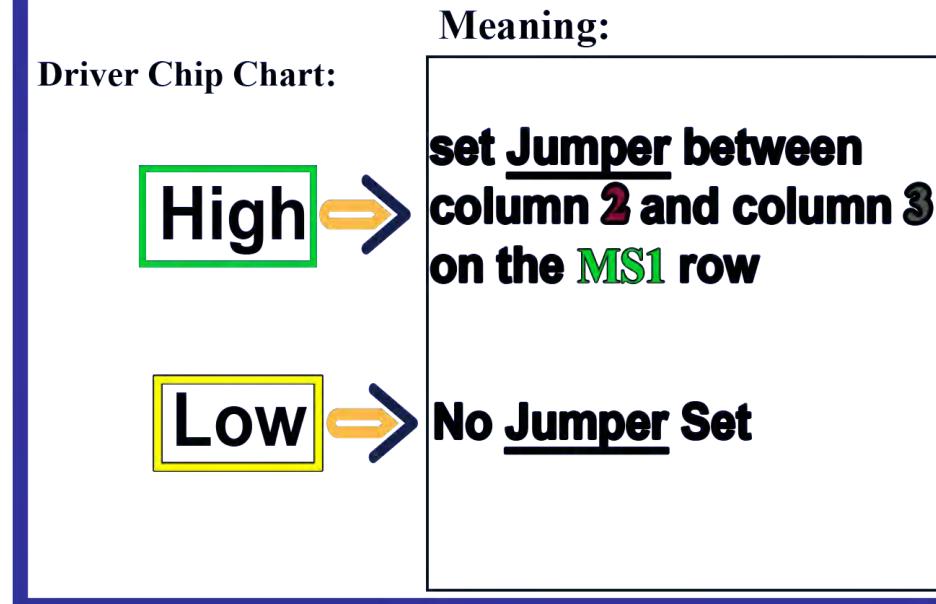
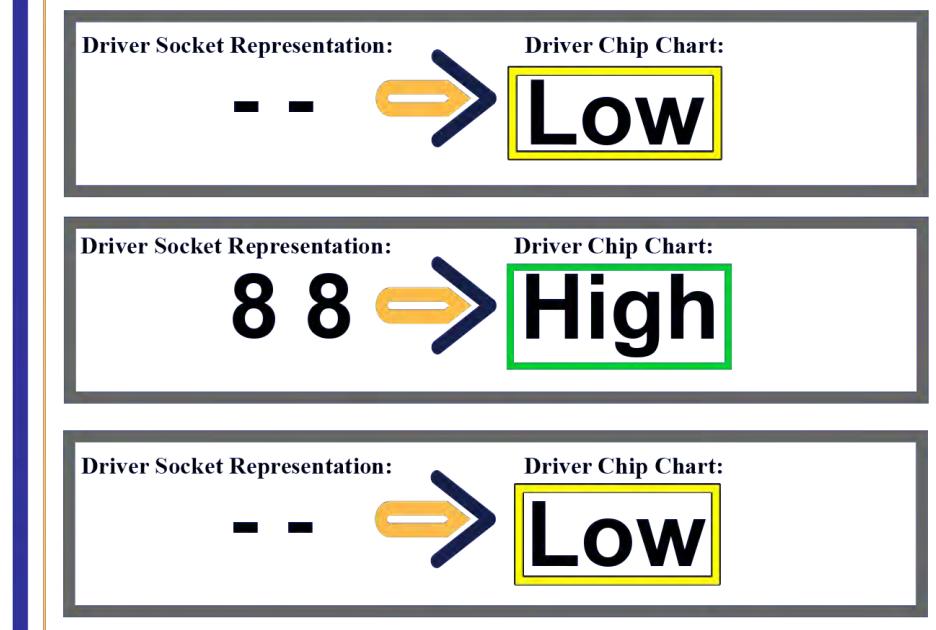


Stand-alone Mode

FYSETC S109



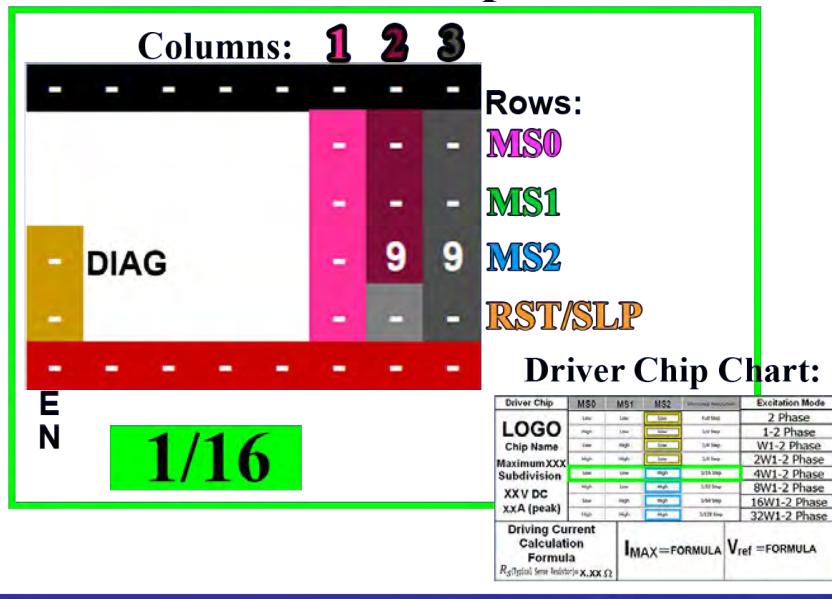
MS1 for Binary State Drivers:



Stand-alone Mode

FYSETC S109

Driver Socket Representation



High:

Columns: **1 2 3**



Rows: **MS0**
MS1
MS2
RST/SLP

MS2 for Binary State Drivers:



Meaning:
Driver Chip Chart:

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

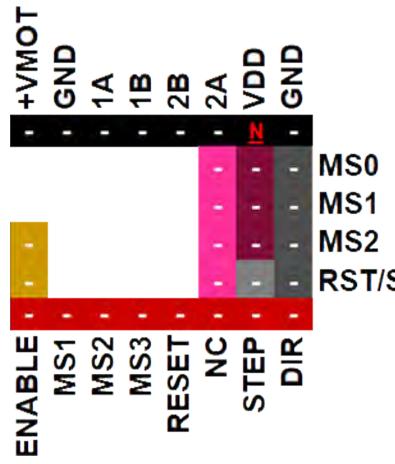
Low → **No Jumper Set**

Stand-alone Mode

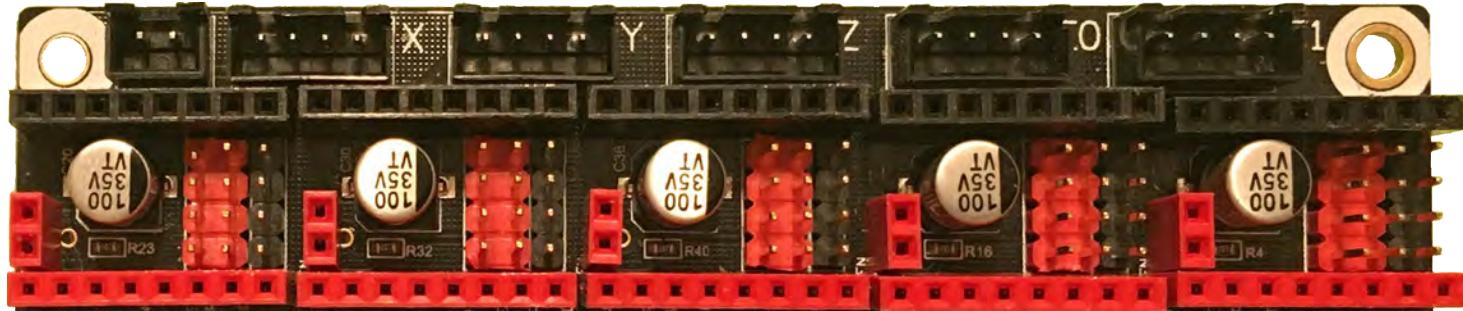
FYSETC S109

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

**STAND
BY**

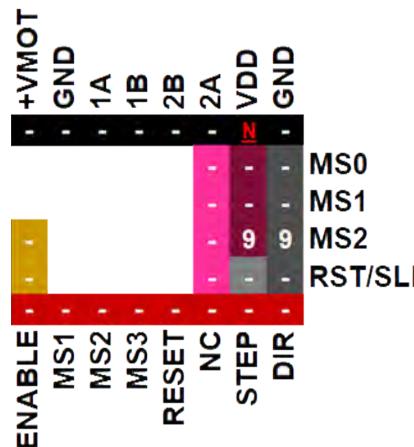


Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!

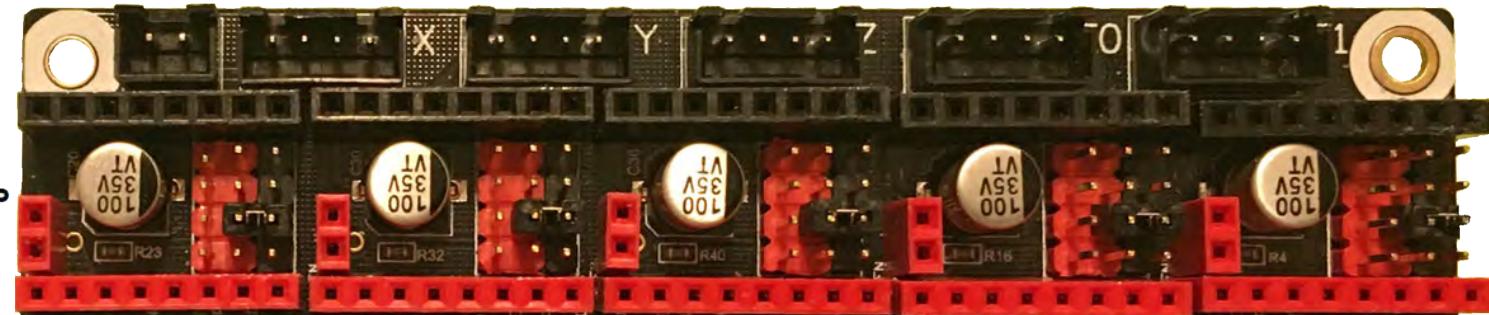


See [Appendix D](#) for legend

STEP



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



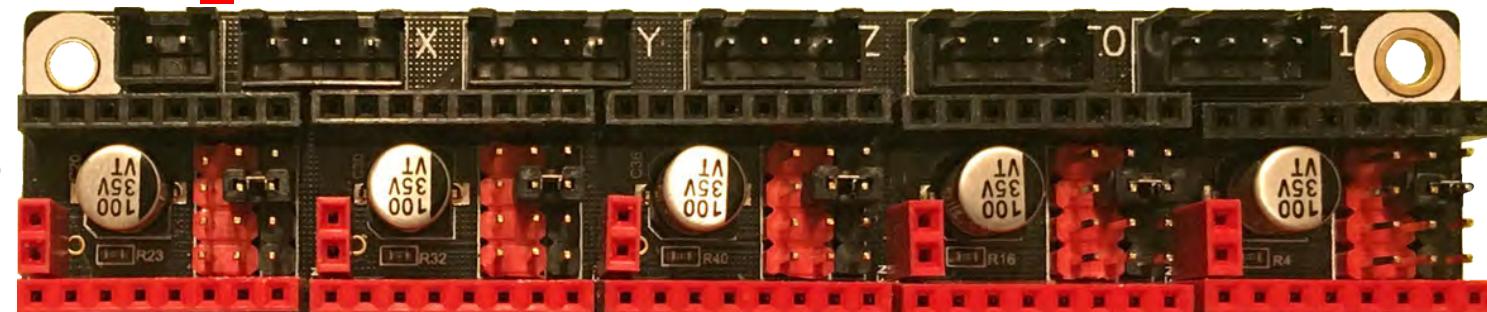
See [Appendix D](#) for legend

Stand-alone Mode

FYSETC S109

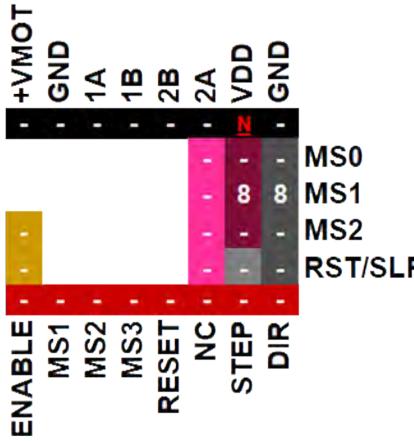
Note: 'Circular half step ("b")' means "clockwise (CW) motor direction with half step resolution". 'Non-circular half step ("a")' means "counterclockwise (CCW) motor direction with half step resolution".

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

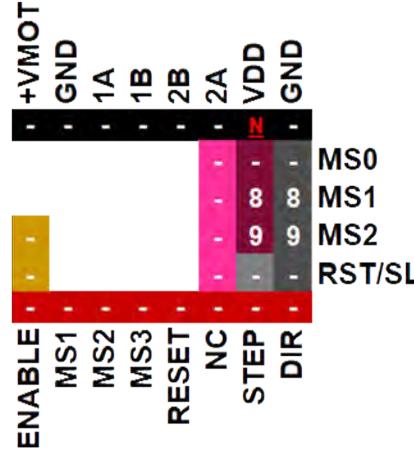


See [Appendix D](#) for legend

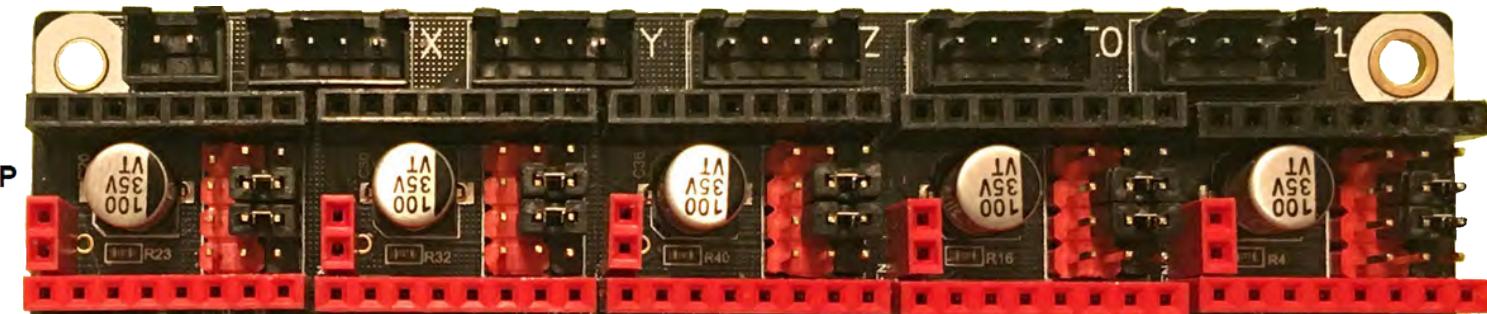
CCW
1 / 2



1 / 4

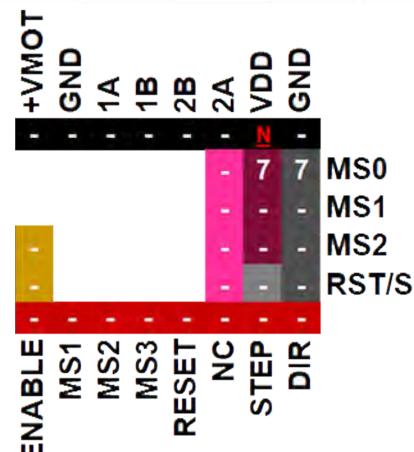


Note: N, on a 3.3 V MCU, like SKR V1.3 **MUST** be disconnected!

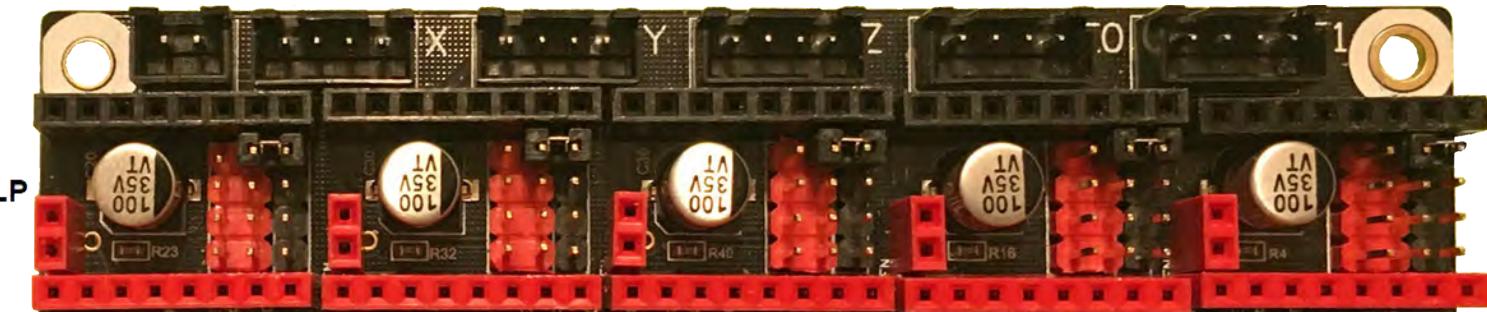


See [Appendix D](#) for legend

CW
1 / 2



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



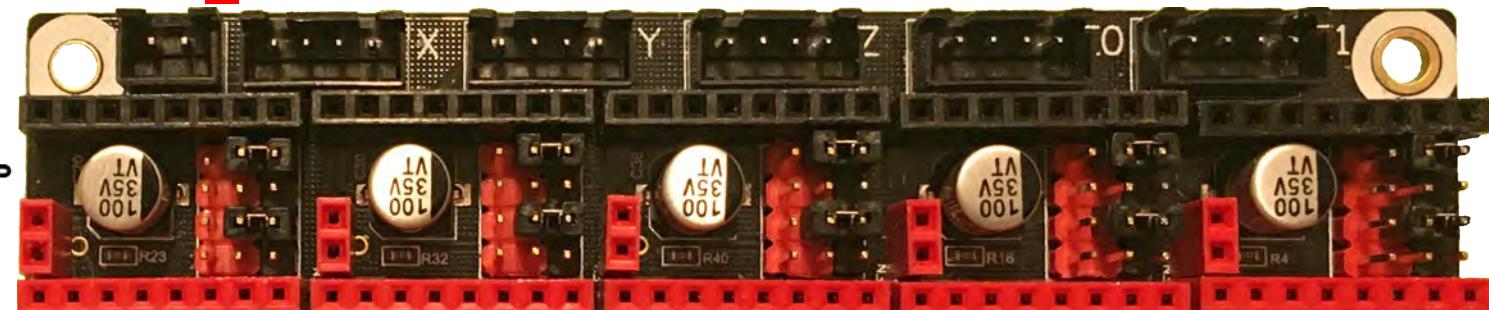
See [Appendix D](#) for legend

Stand-alone Mode

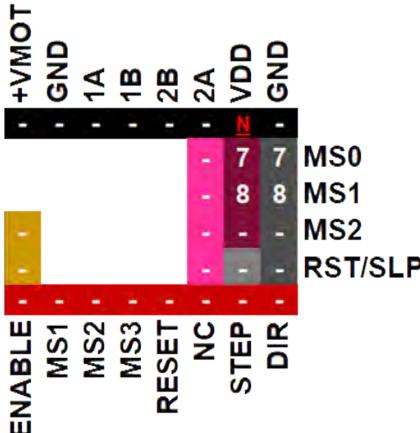
FYSETC S109

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

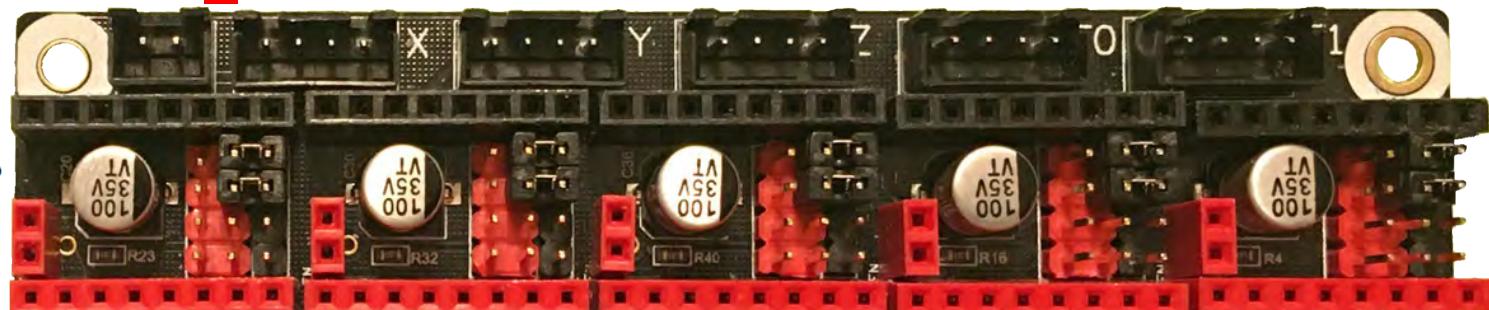
Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



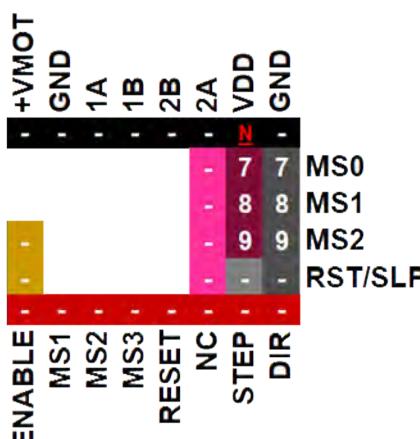
See [Appendix D](#) for legend



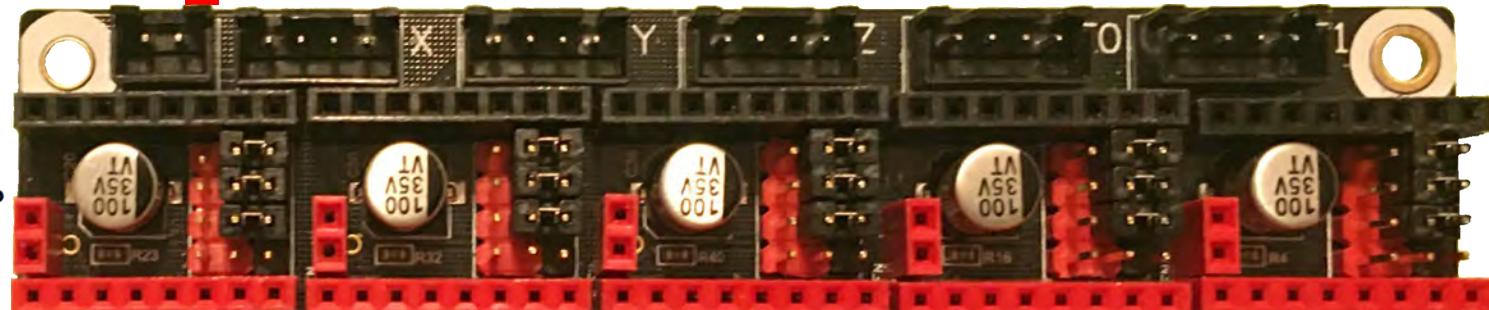
Note: N, on a 3.3 V MCU, like SKR V1.3, **MUST** be disconnected!



See [Appendix D](#) for legend



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

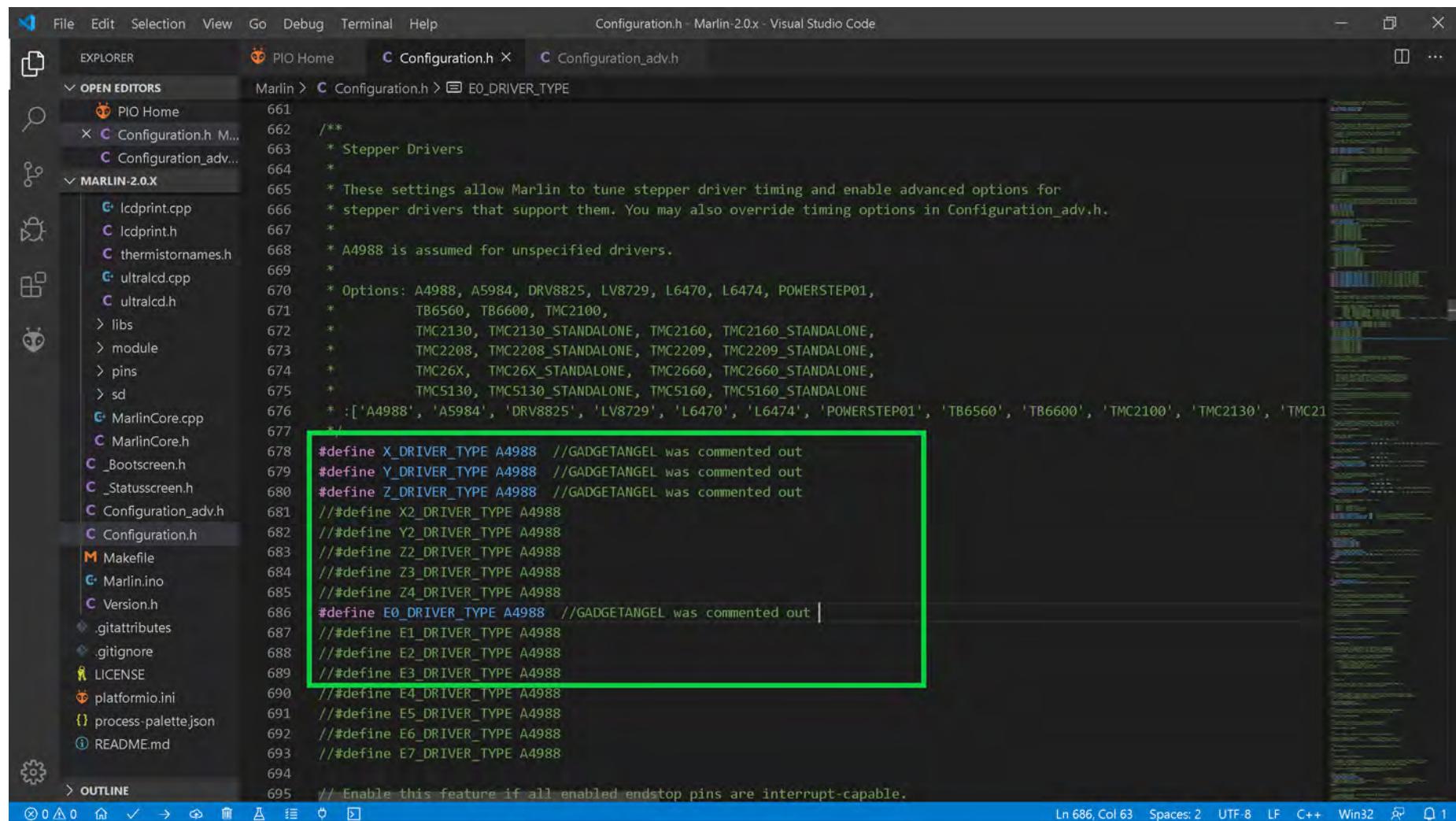


See [Appendix D](#) for legend

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

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

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



```

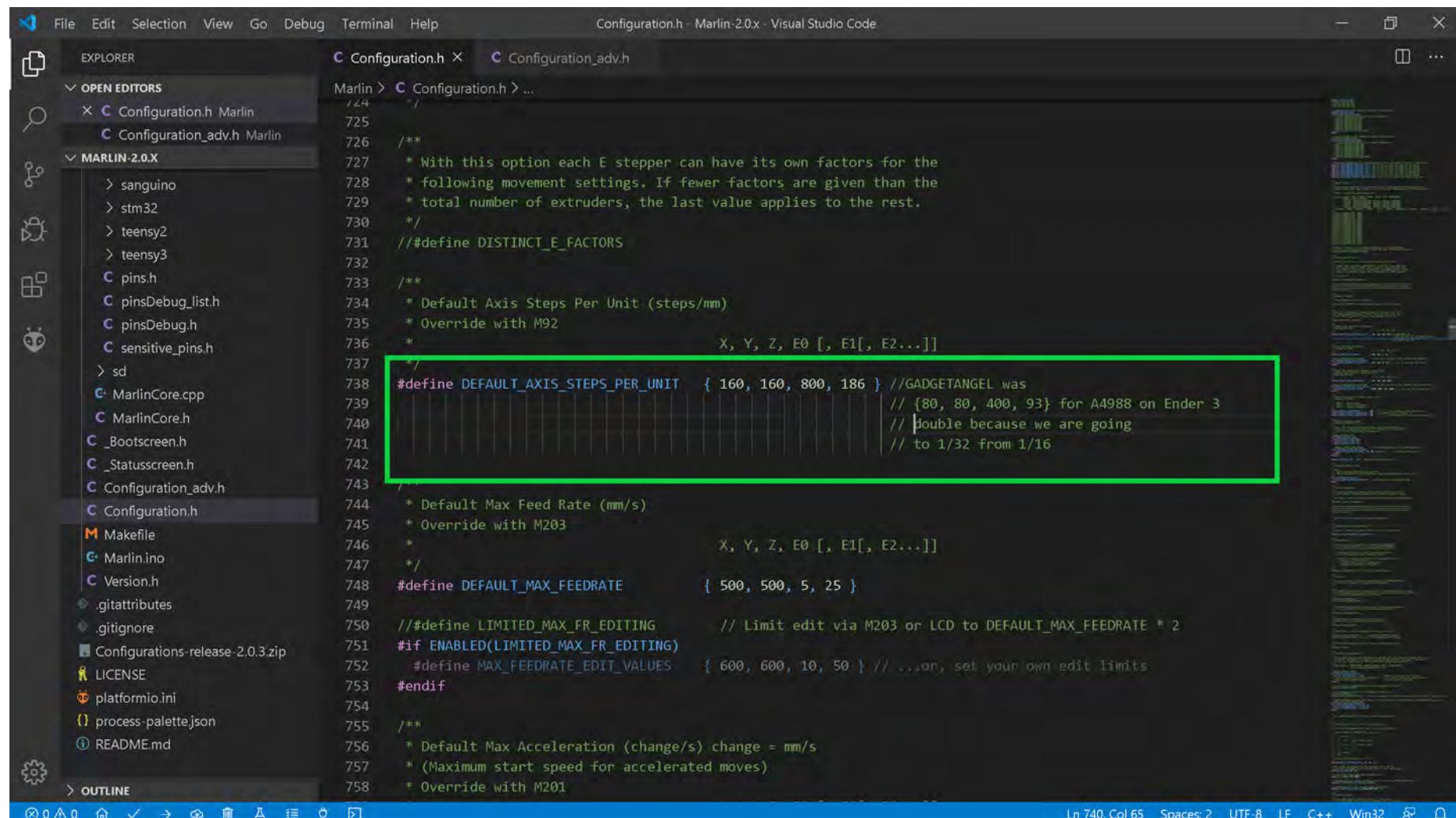
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
  PIO Home
  Configuration.h M...
  Configuration_adv...
MARLIN-2.0.X
  Lcdprint.cpp
  Lcdprint.h
  thermistornames.h
  ultralcd.cpp
  ultralcd.h
  > libs
  > module
  > pins
  > sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
Ln 686 Col 63 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1
  #define X_DRIVER_TYPE A4988 //GADGETANGEL was commented out
  #define Y_DRIVER_TYPE A4988 //GADGETANGEL was commented out
  #define Z_DRIVER_TYPE A4988 //GADGETANGEL was commented out
  //#define X2_DRIVER_TYPE A4988
  //#define Y2_DRIVER_TYPE A4988
  //#define Z2_DRIVER_TYPE A4988
  //#define Z3_DRIVER_TYPE A4988
  //#define Z4_DRIVER_TYPE A4988
  #define E0_DRIVER_TYPE A4988 //GADGETANGEL was commented out
  //#define E1_DRIVER_TYPE A4988
  //#define E2_DRIVER_TYPE A4988
  //#define E3_DRIVER_TYPE A4988
  //#define E4_DRIVER_TYPE A4988
  //#define E5_DRIVER_TYPE A4988
  //#define E6_DRIVER_TYPE A4988
  //#define E7_DRIVER_TYPE A4988
  // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



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

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

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

- Go to the next page.

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

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

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

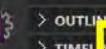
- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Sidebar:** EXPLORER, OPEN EDITORS, MARLIN-2.0.X, OUTLINE.
- Open Editors:** Configuration.h (Marlin), Configuration_adv.h.
- Code Editor:** The Configuration.h file is open, showing C code. A green box highlights the following section:

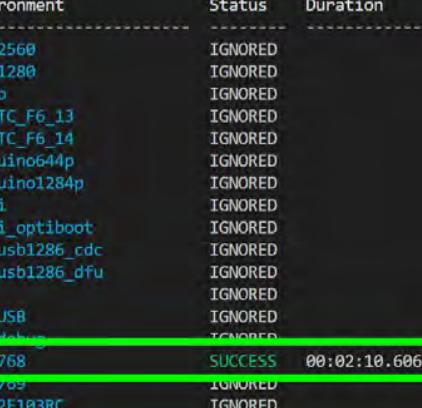
```
#define INVERT_X_DIR false          //GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Y_DIR false          //GADGETANGEL was true, stepper motor driver inverts motor direction
#define INVERT_Z_DIR true           //GADGETANGEL was false, stepper motor driver inverts motor direction
```
- Right Panel:** Shows the full codebase of the Marlin repository.

- Go to the next page.

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

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

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

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

```

124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
126 // #define BLUETOOTH
127
128 // Choose the name from boards.h that matches your setup
129 #ifndef MOTHERBOARD
130 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
131 #endif
132
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

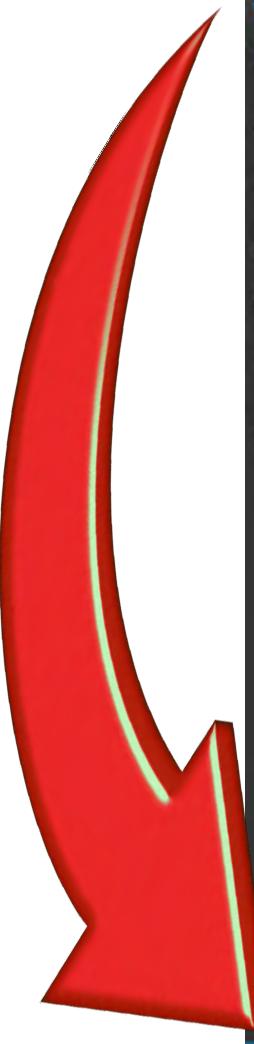
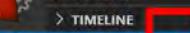
```

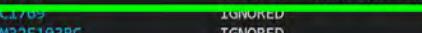
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

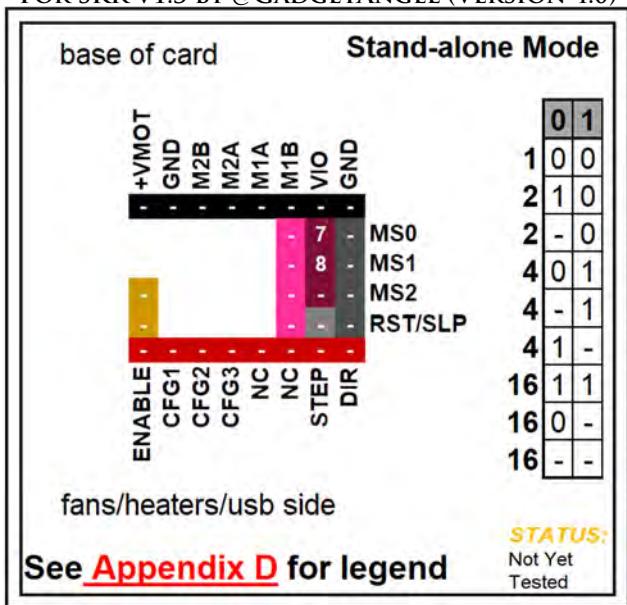
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

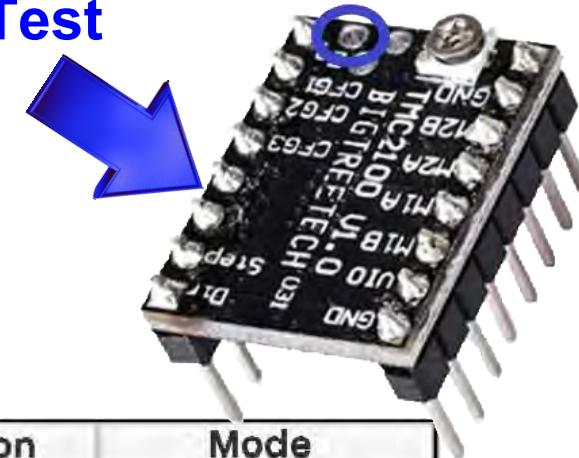
[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1709	IGNORED	
STM32F103RC	IGNORED	

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

**BIQU TMC2100**Stand-alone Mode

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

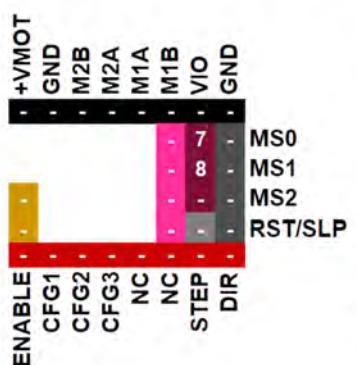


Driver Chip	MS0	MS1	Steps	Interpolation	Mode
BIQU® TMC2100 Stand Alone Mode Maximum 16 Subdivision 46V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4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

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

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

base of card

Stand-alone Mode

fans/heaters/usb side

See [Appendix D](#) for legend**Driver Chip Chart:**

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

Driving Current Calculation Formula
 $R_s \text{ (typical Sense Resistor)} = 0.11\Omega$

$I_{MAX} = V_{ref}$

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

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

$V_{ref} = I_{MAX}$

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

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

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

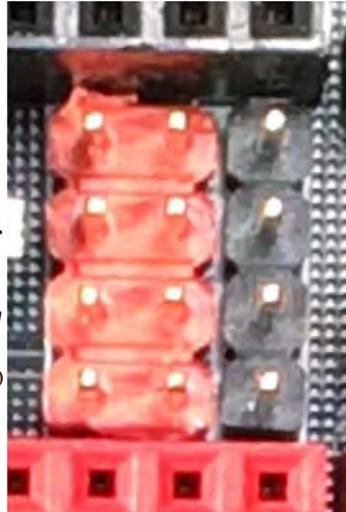
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

BIGU TMC2100**Stand-alone Mode****SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1**

OPEN ➔ **No Jumper**

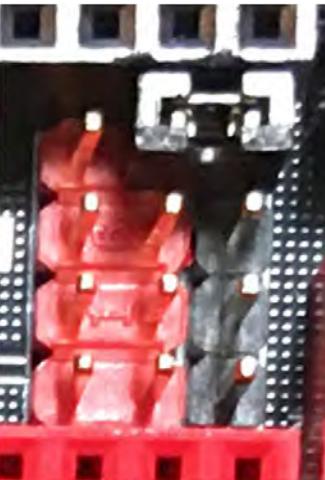
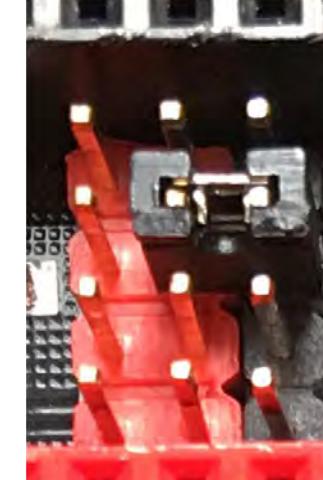
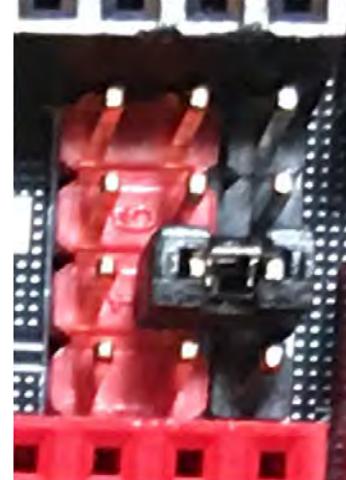


Rows:

MS0**MS1****MS2****RST/SLP**

High ➔ **Jumper Set**

Rows:

MS0**MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

Driver Chip Chart:

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

Driving Current Calculation Formula

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

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

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

$V_{ref} = I_{MAX}$

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

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

R_S (typical Sense Resistor) = 0.11Ω

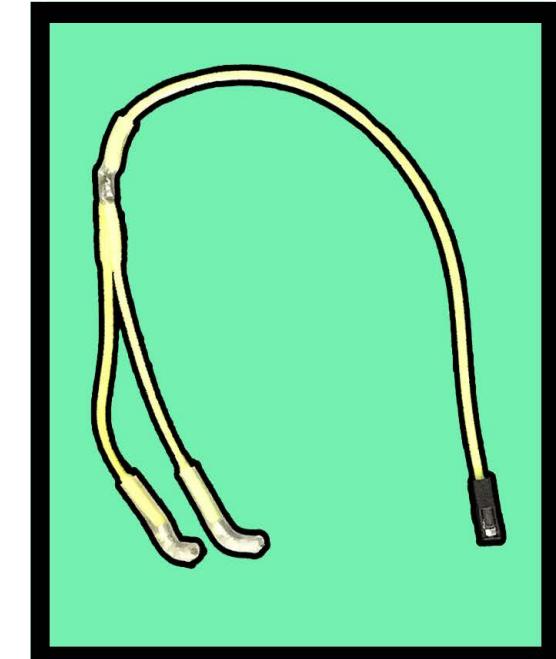
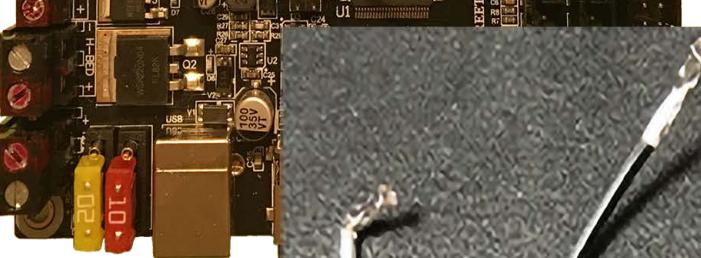
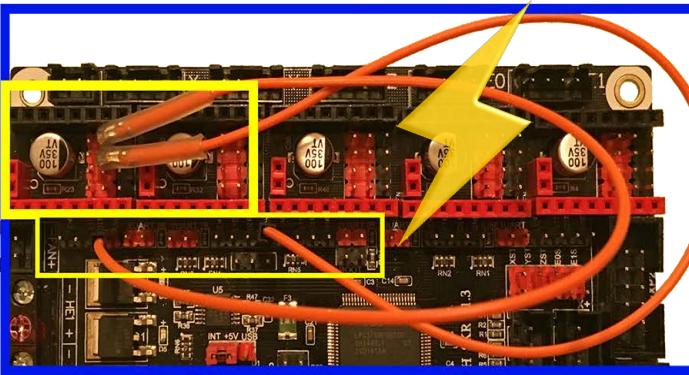
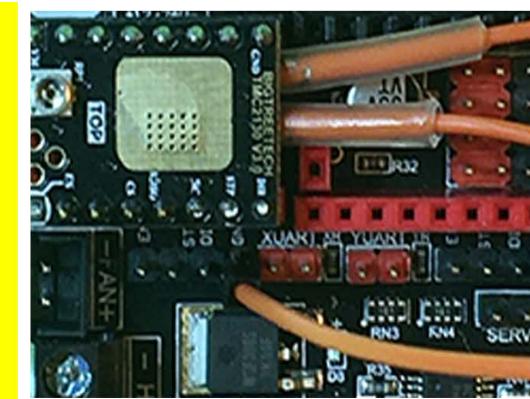
Low ➔
MS0 or MS1
connected to
Ground(GND)
via DuPont
Jumper Cable

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

BIGU TMC2100

Stand-alone Mode

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

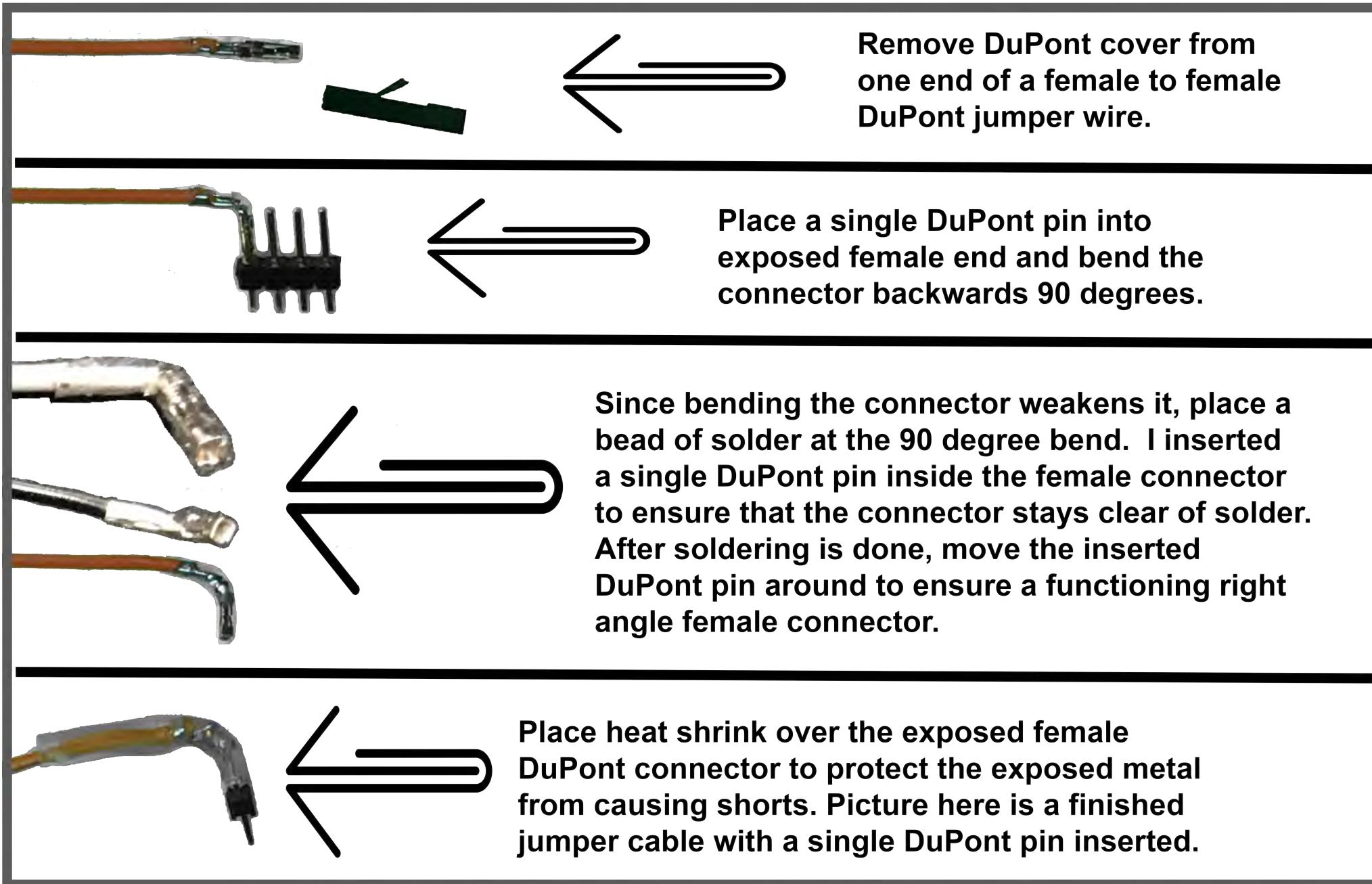


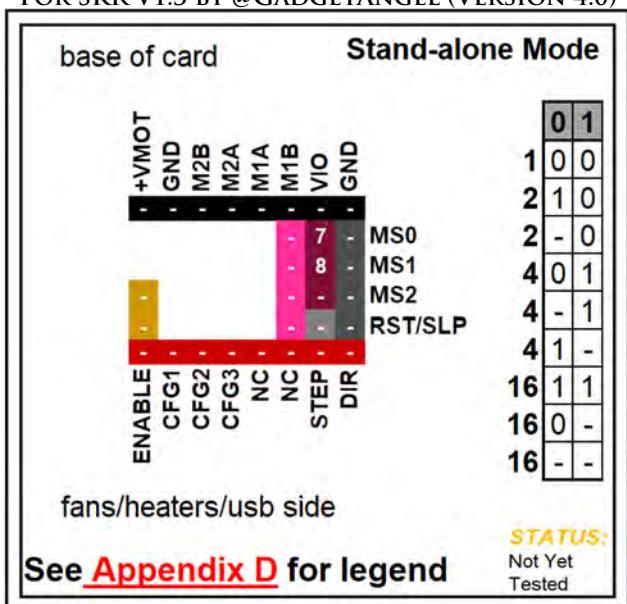
Stand-alone Mode

BIGU TMC2100

Stand-alone Mode

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





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

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

BIQU TMC2100

Stand-alone Mode

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

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



$R_s = R050$ is 0.05 Ohms

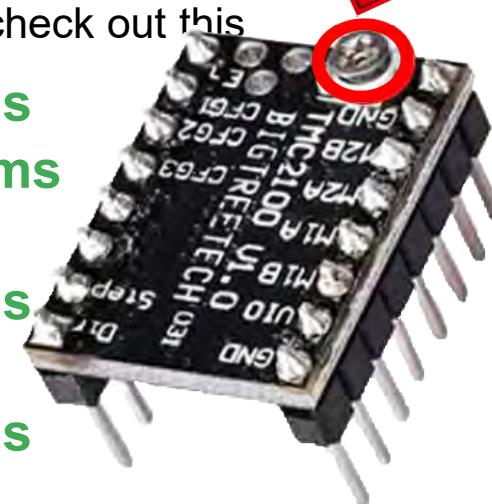
$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms



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

Stand-alone Mode

BIQU TMC2100

Stand-alone Mode

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

X- (XST) X-DIAG1
Y- (YST) Y-DIAG1
Z- (ZST) Z-DIAG1
(E0ST) E0-DIAG1 X+
(E1ST) E1-DIAG1 Y+



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

XUART

YUART

ZUART

E0 UART

E1 UART



Stand-alone Mode

BIGU TMC2100

Stand-alone Mode

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

Driver Socket Representation

Columns: 1 2 3
Rows: MS0
MS1
MS2
RST/SLP

DIAG

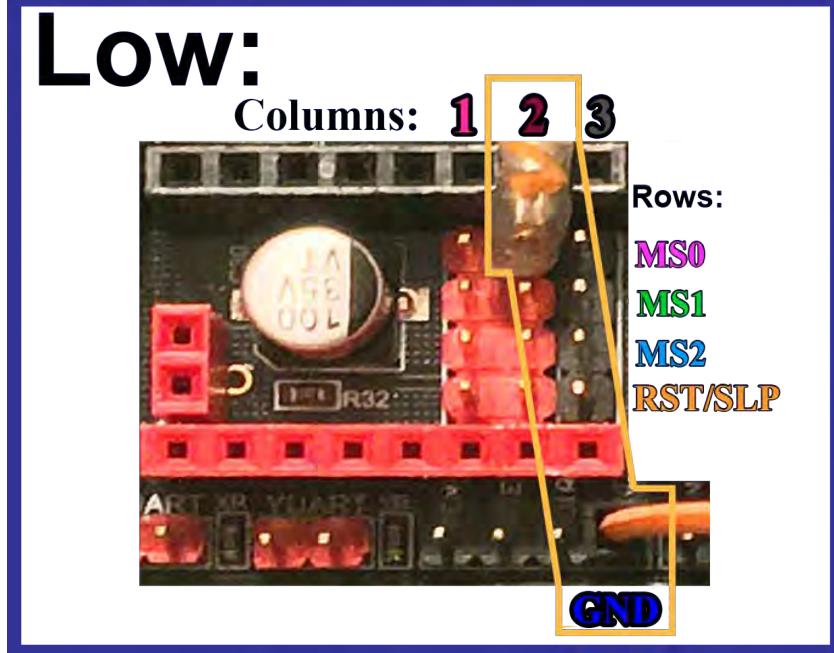
E N 1/4 with 1/256 in StealthChop Mode

Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode	Low	Low	1	NONE	spreadCycle
	High	Low	1/2	NONE	spreadCycle
	OPEN	Low	1/2	1/256	spreadCycle
	LOW	High	1/4	NONE	spreadCycle
	OPEN	High	1/4	1/256	spreadCycle
	High	OPEN	1/4	1/256	stealthChop
High	High	1/16	NONE	spreadCycle	
LOW	OPEN	1/16	1/256	spreadCycle	
OPEN	OPEN	1/16	1/256	stealthChop	

Driving Current Calculation Formula: $I_{MAX} = V_{ref}$ See Appendix B. $V_{ref} = I_{MAX}$ See Appendix B. $I_{MAX} = I_{MAX} * 0.90$ See Appendix B. $V_{ref} = V_{ref} * 0.90$

No Jumper set



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

Driver Socket Representation: 7 7 → High

Driver Chip Chart: High

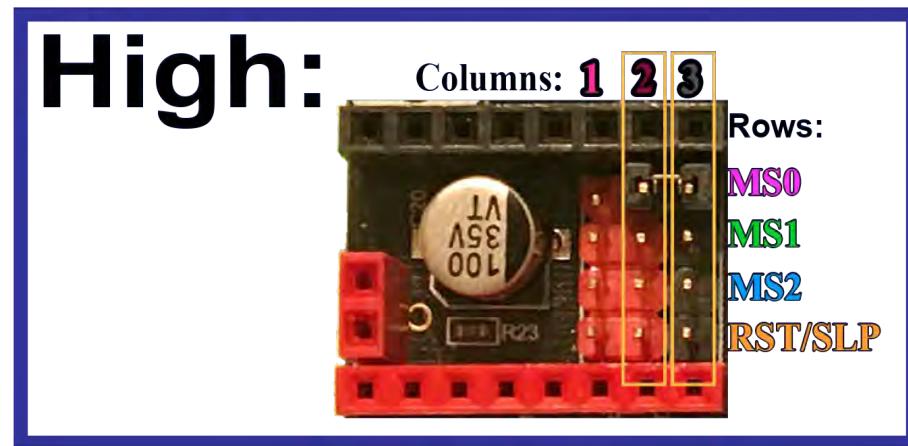
Driver Socket Representation: 7* - → Low

Driver Chip Chart: Low

Driver Socket Representation: - - → Open

Driver Chip Chart: Open

Meaning:
set Jumper between column 2 and column 3 on the MS0 row
Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)
No Jumper set

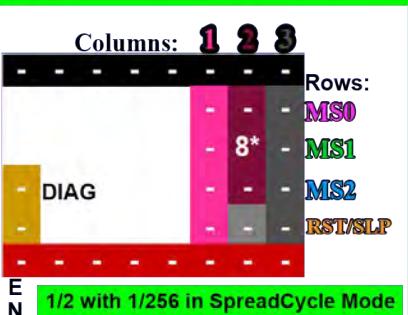
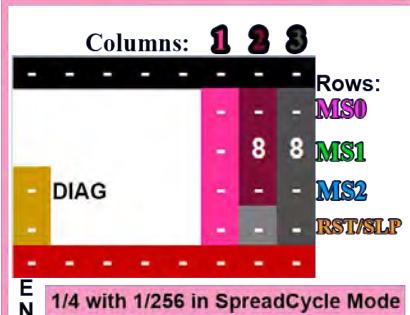


Stand-alone Mode

BIGU TMC2100

Stand-alone Mode

Driver Socket Representation

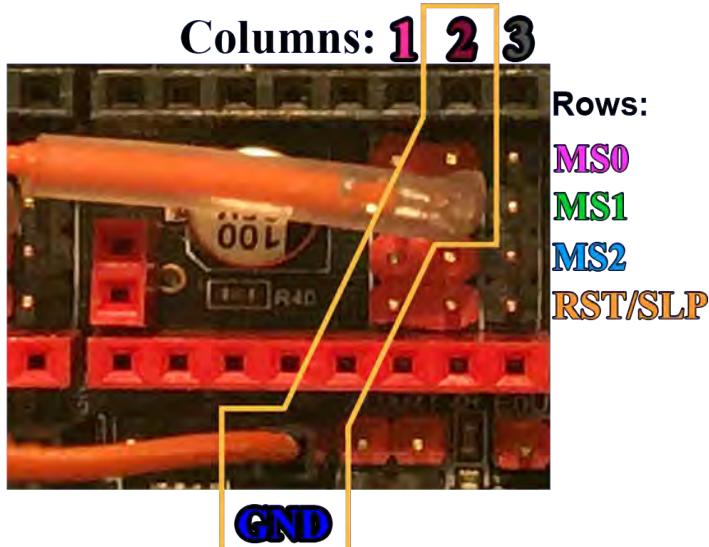


Driver Chip Chart:

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



LOW:



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



Driver Chip Chart:

High

Low

Open

Meaning:

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

Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)

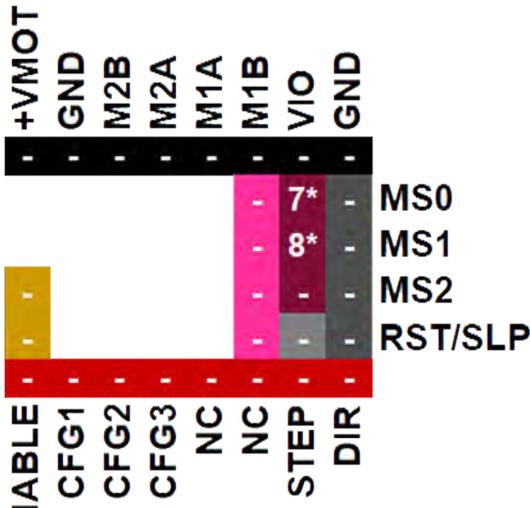
No Jumper set

High:



Stand-alone Mode

Stand-alone
Mode



STEP

Interpolation:
none

SpreadCycle

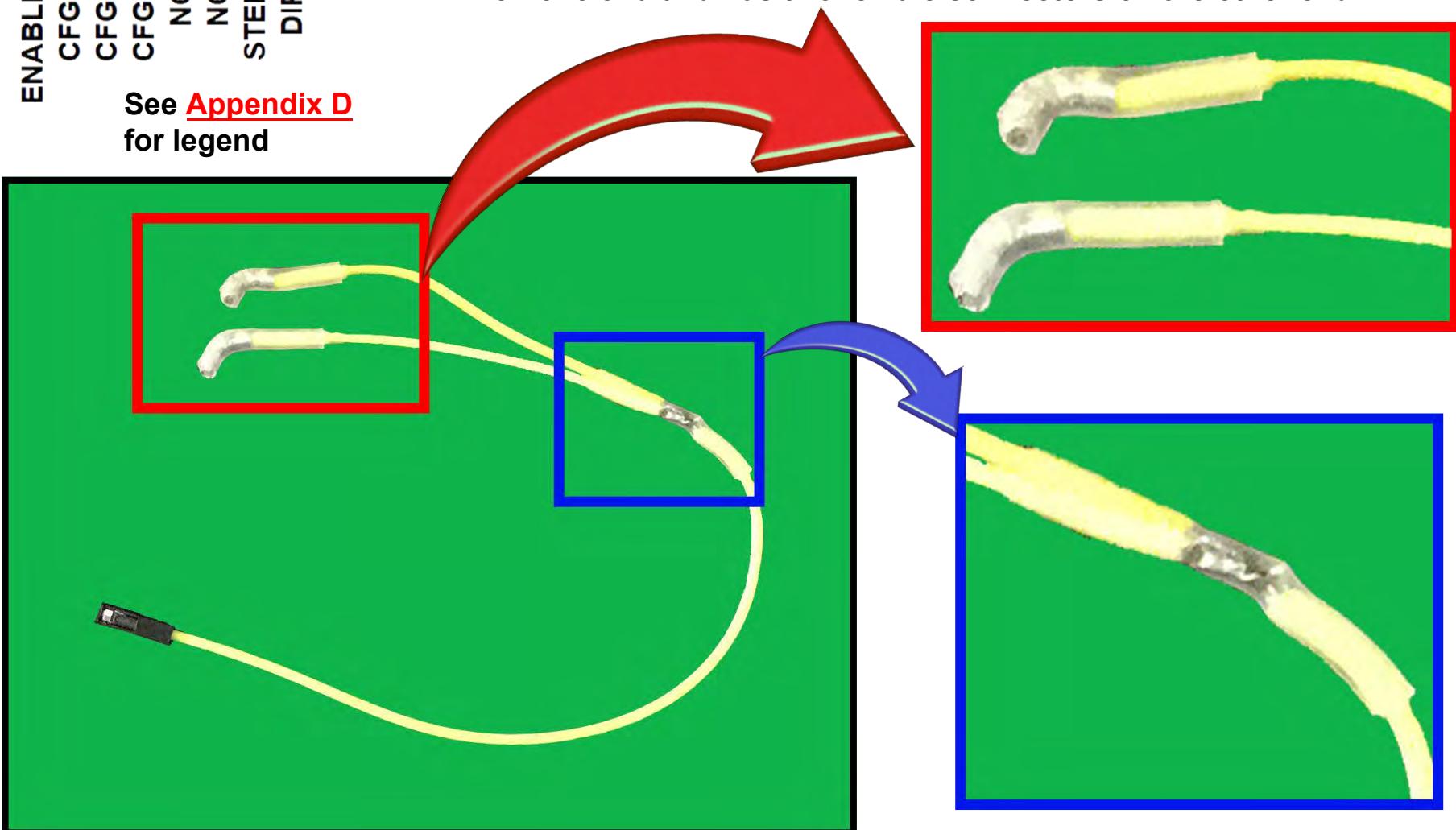
See [Appendix D](#)
for legend

BIQU TMC2100

Stand-alone Mode

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

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



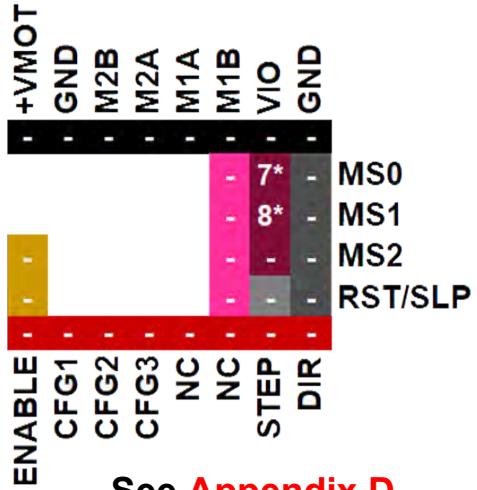
Stand-alone Mode

Stand-alone
Mode

STEP

Interpolation:
none

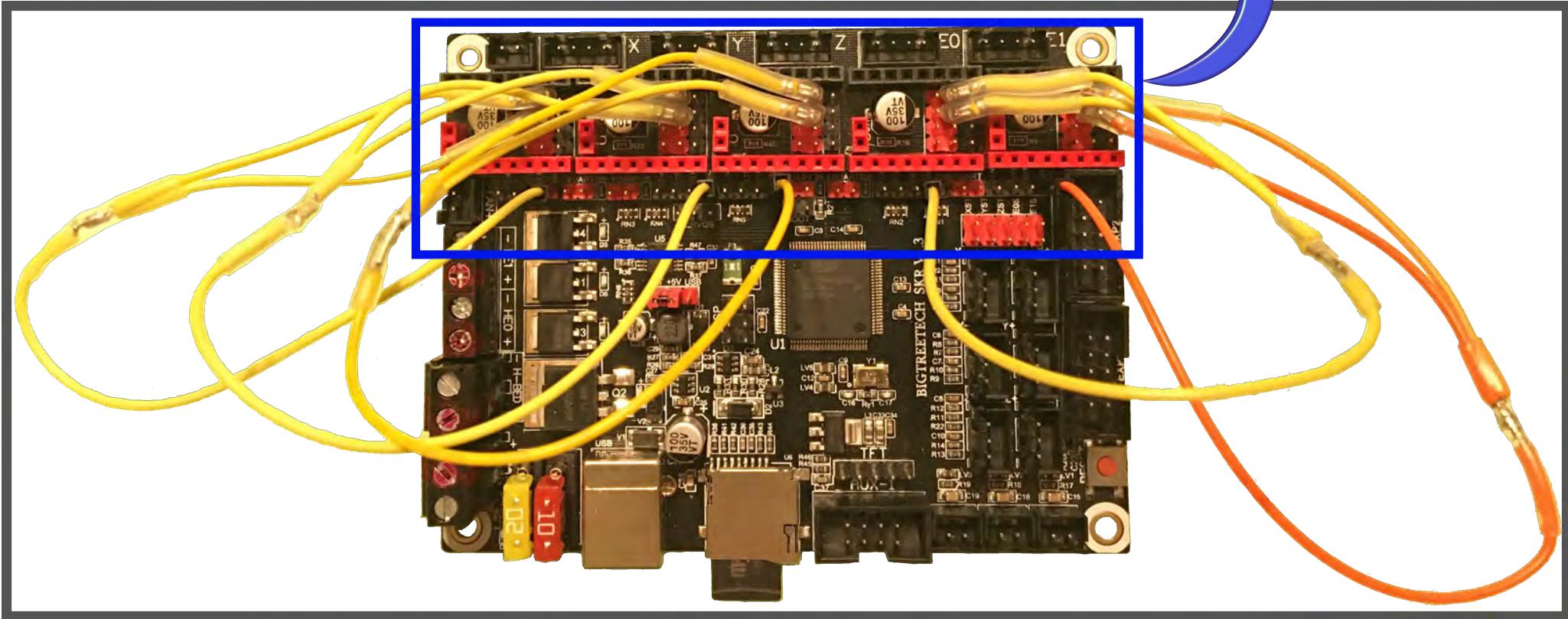
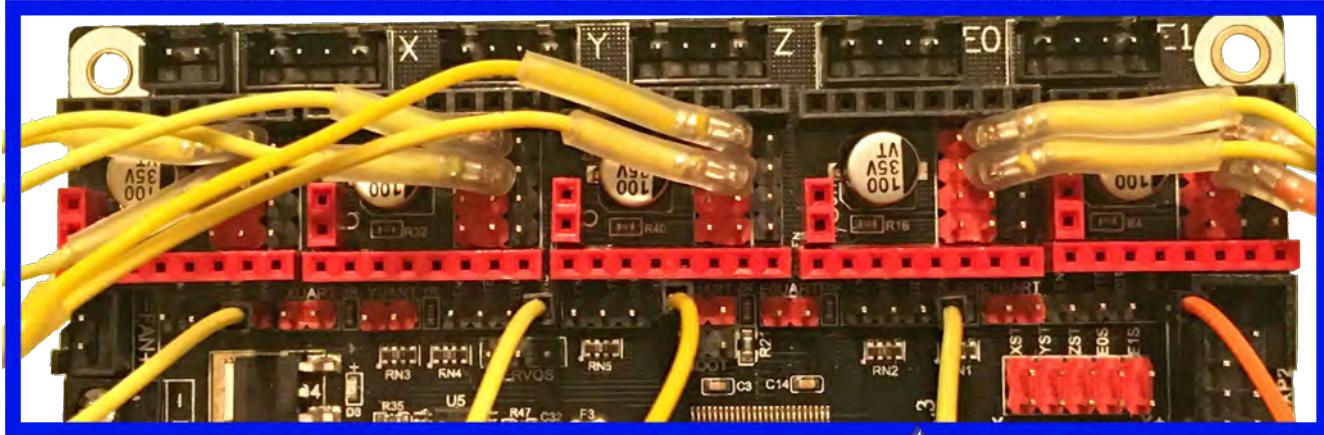
SpreadCycle



See [Appendix D](#)
for legend

BIQU TMC2100

Stand-alone Mode

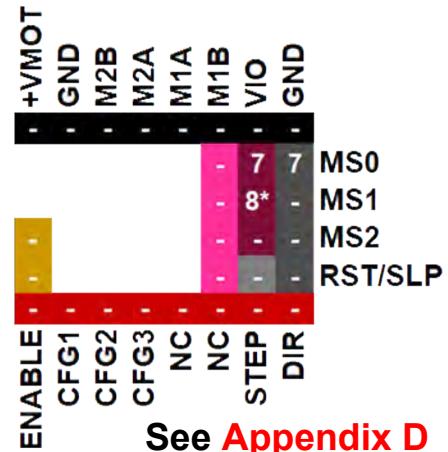


Stand-alone Mode

Stand-alone
Mode

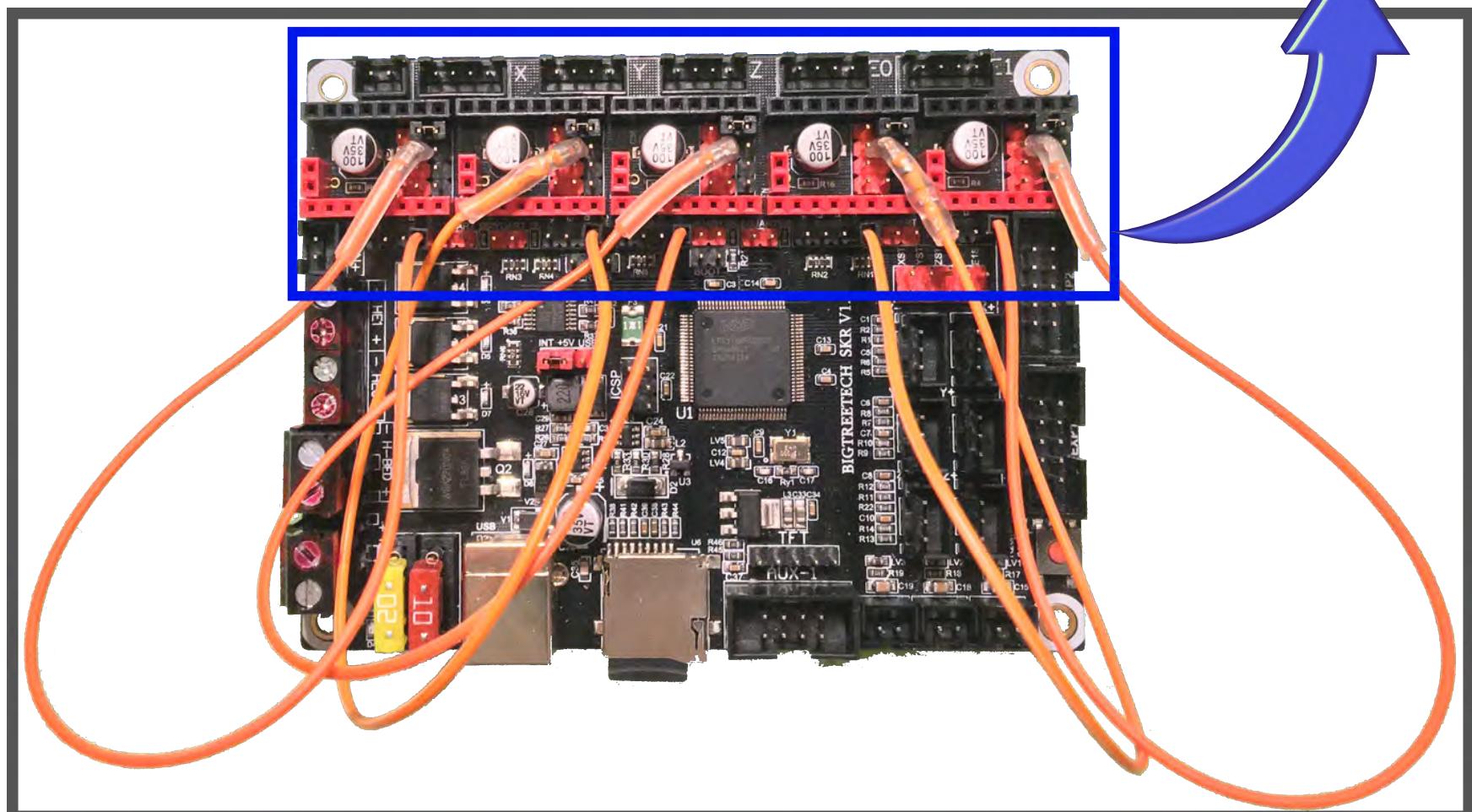
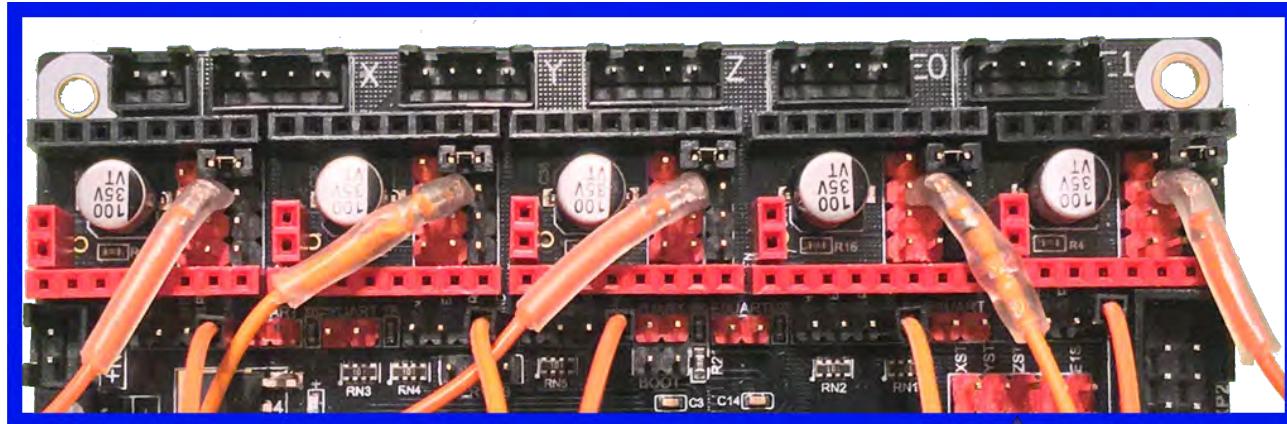
1 / 2

Interpolation:
none
SpreadCycle



BIQU TMC2100

Stand-alone Mode



Stand-alone Mode

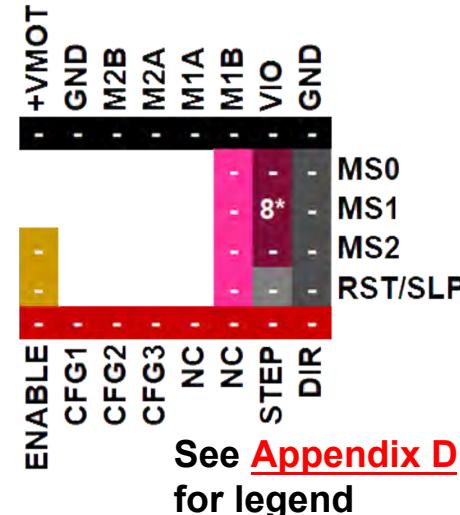
Stand-alone
Mode

1 / 2

Interpolation:

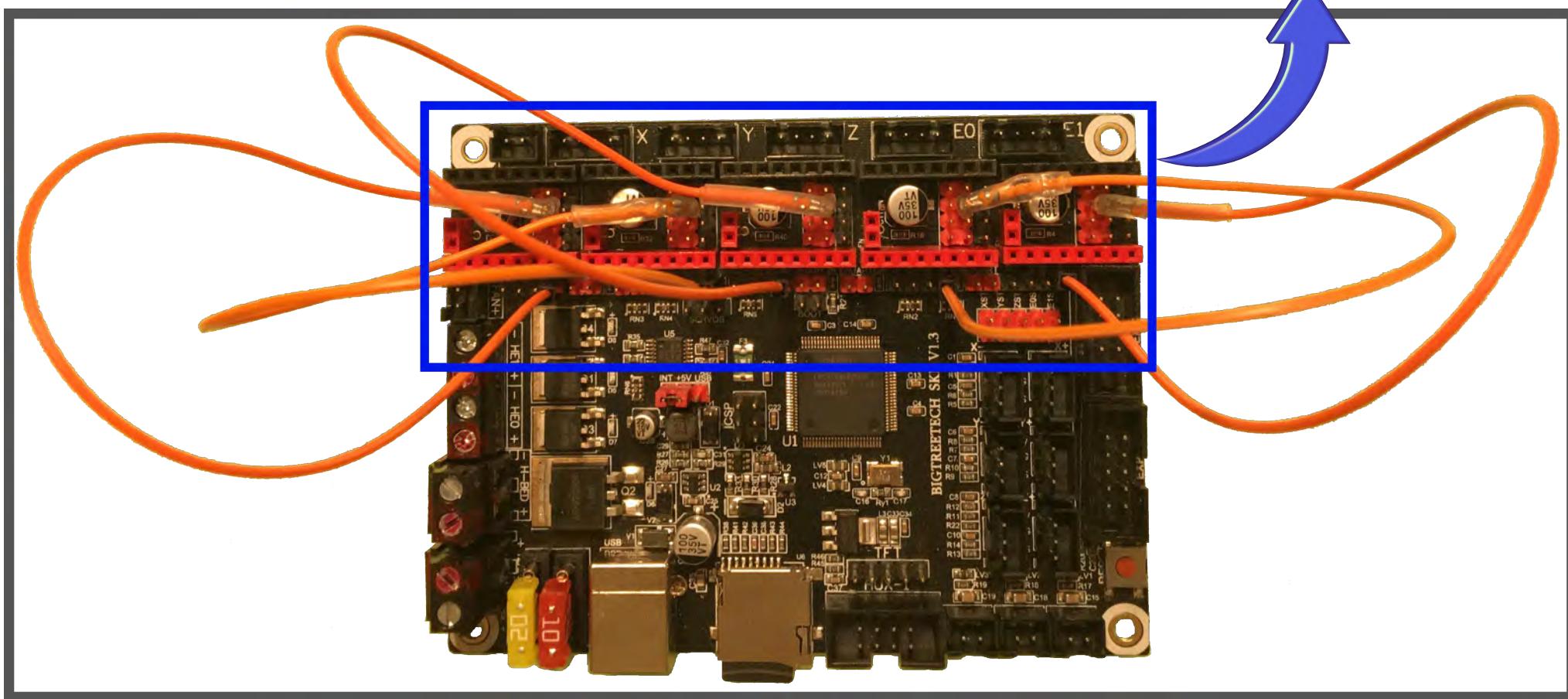
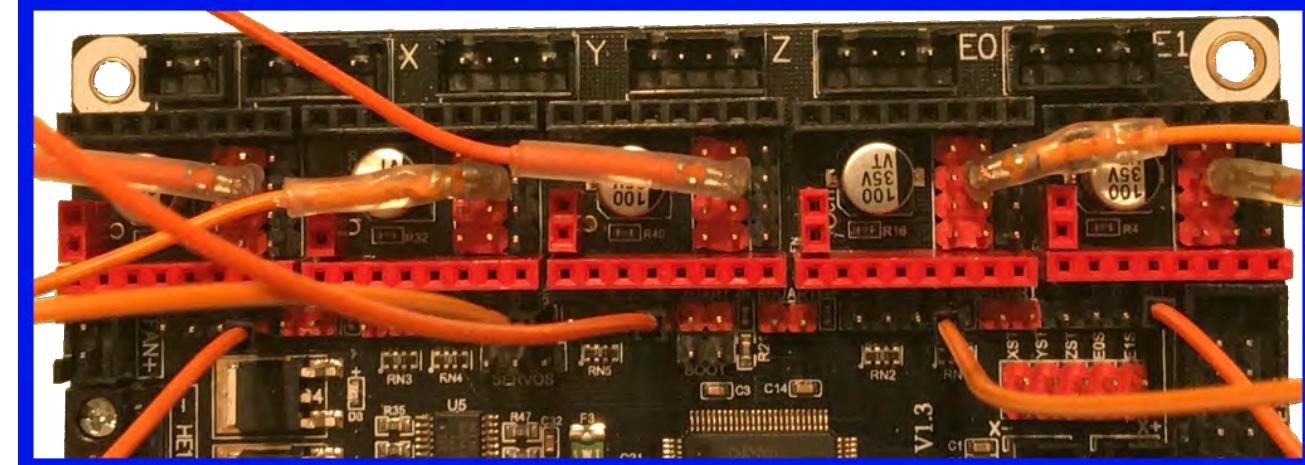
1 / 256

SpreadCycle



BIQU TMC2100

Stand-alone Mode



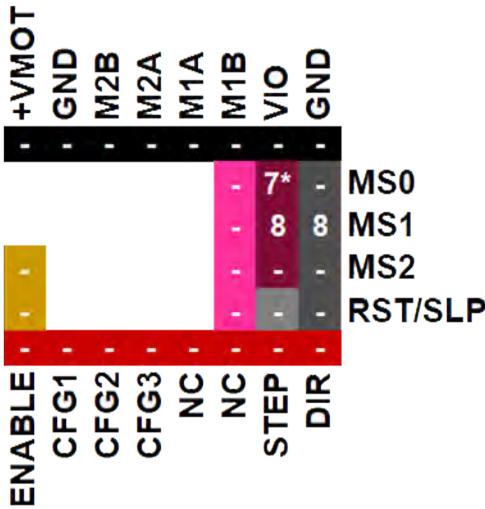
Stand-alone Mode

Stand-alone
Mode

1 / 4

Interpolation:
none

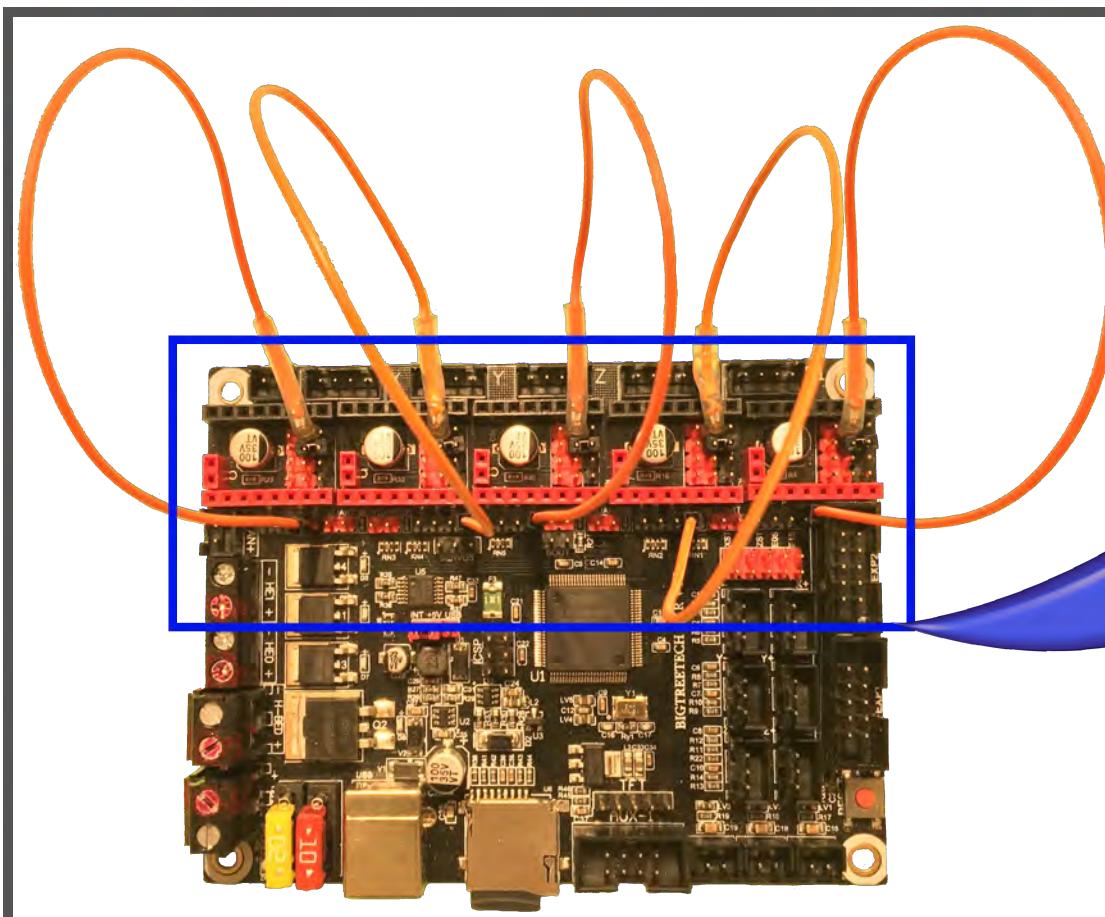
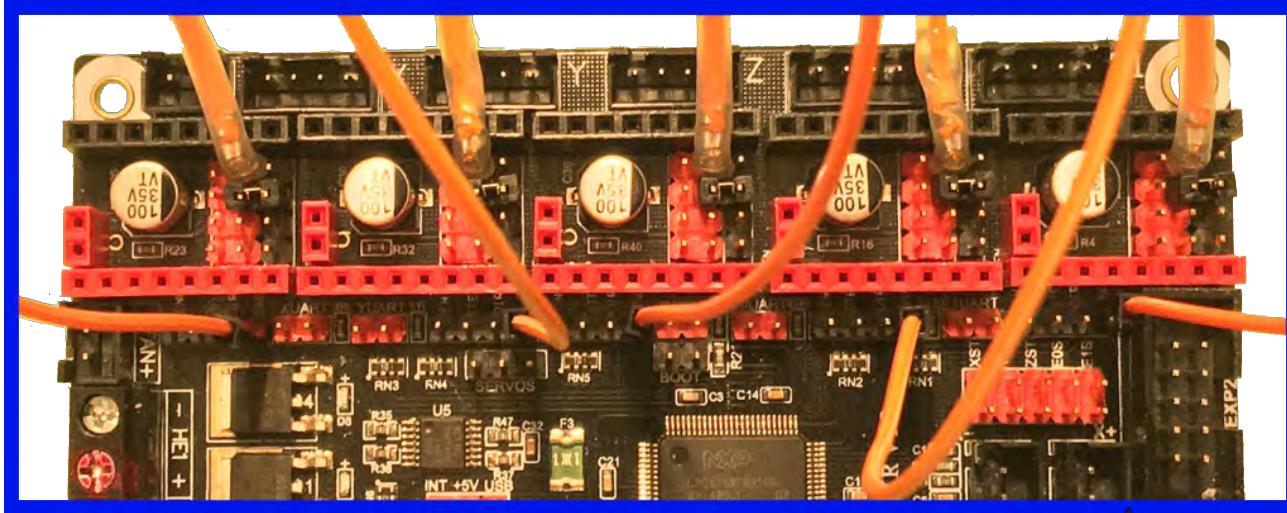
SpreadCycle



See [Appendix D](#)
for legend

BIGTREETECH BIQU TMC2100

Stand-alone Mode

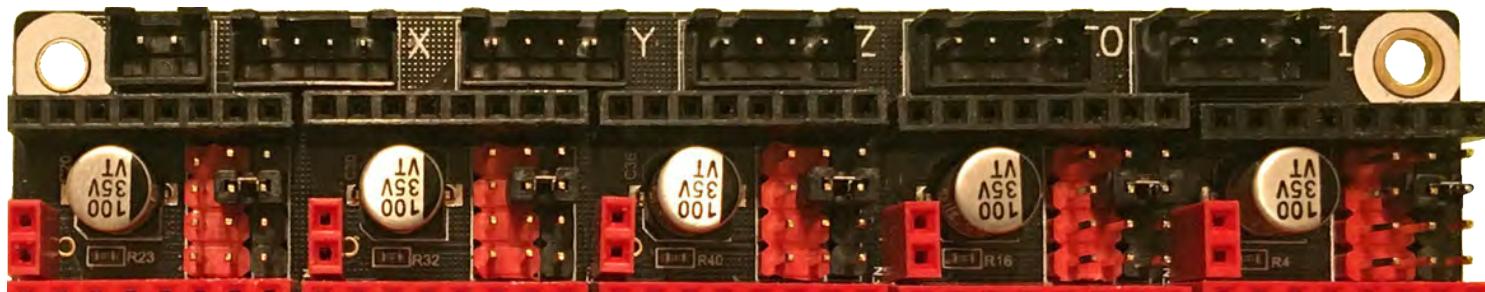


Stand-alone Mode

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

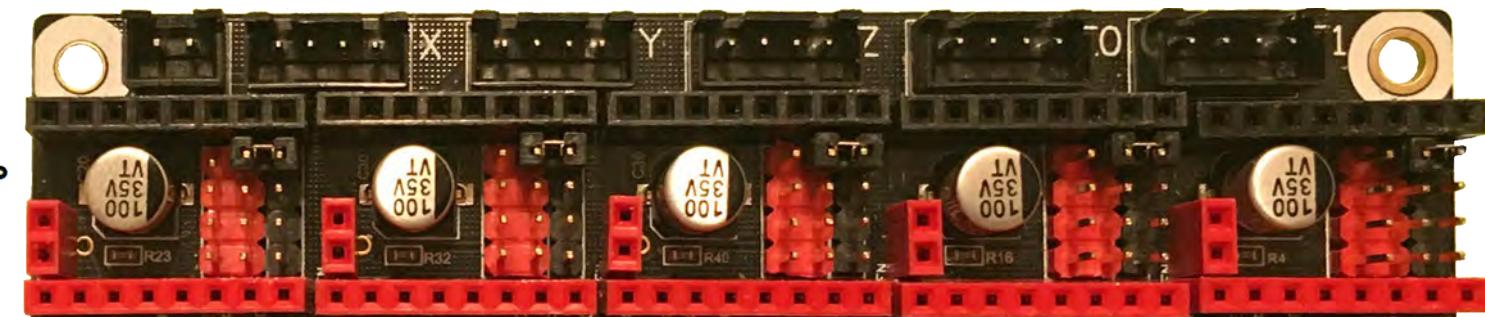
BIGU TMC2100

Stand-alone Mode



See [Appendix D](#) for legend

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



See [Appendix D](#) for legend

Stand-alone Mode	+VMOT	GND	M2B	M2A	M1A	M1B	VIO	GND
1 / 16								
Interpolation:	none							
SpreadCycle	ENABLE	CFG1	CFG2	CFG3	NC	NC	STEP	DIR



See [Appendix D](#) for legend

Stand-alone Mode

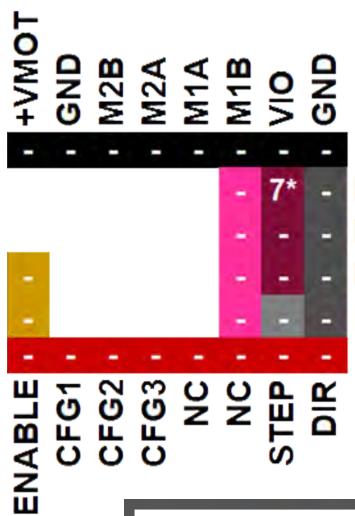
Stand-alone
Mode

1 / 16

Interpolation:

1 / 256

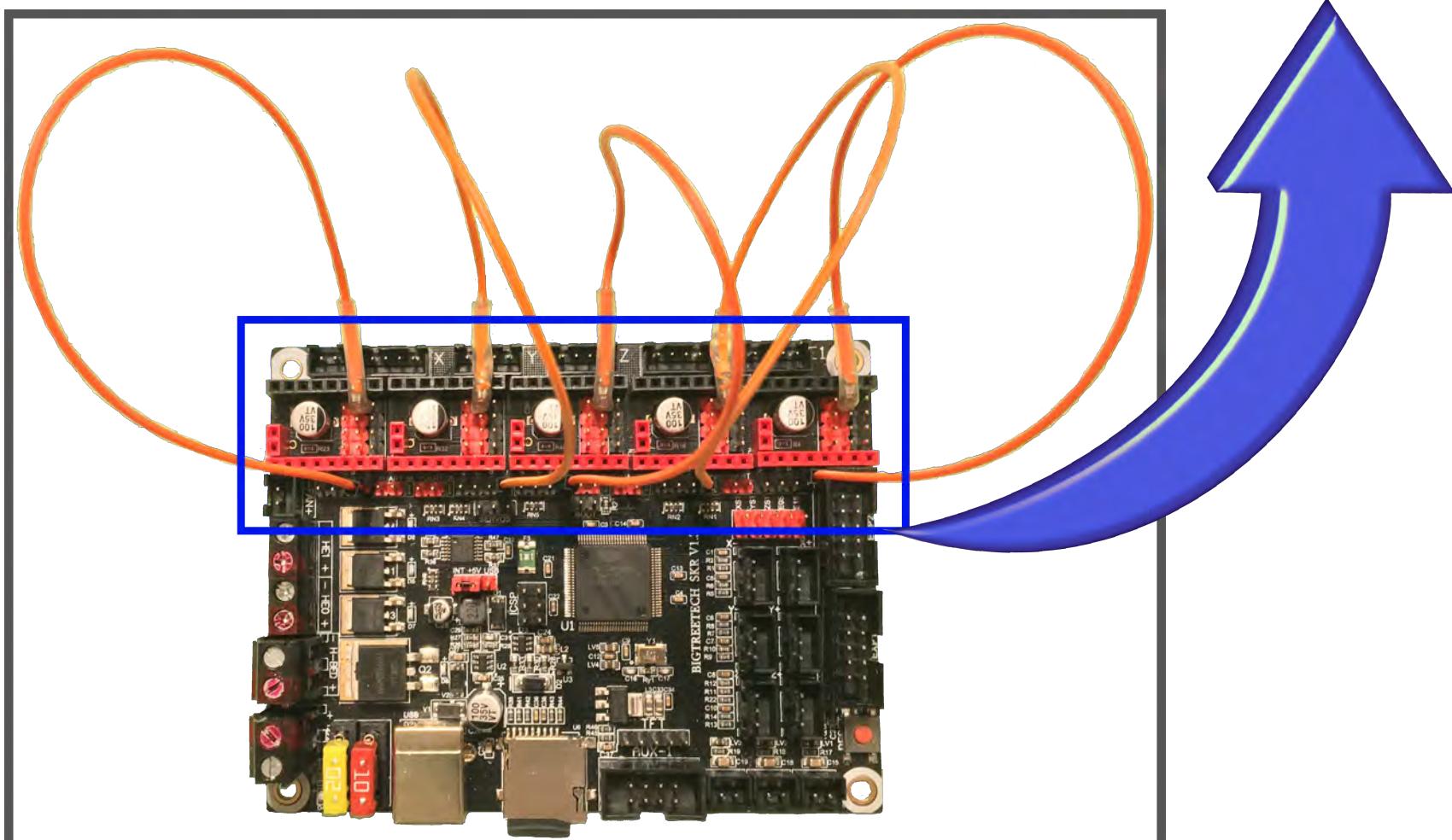
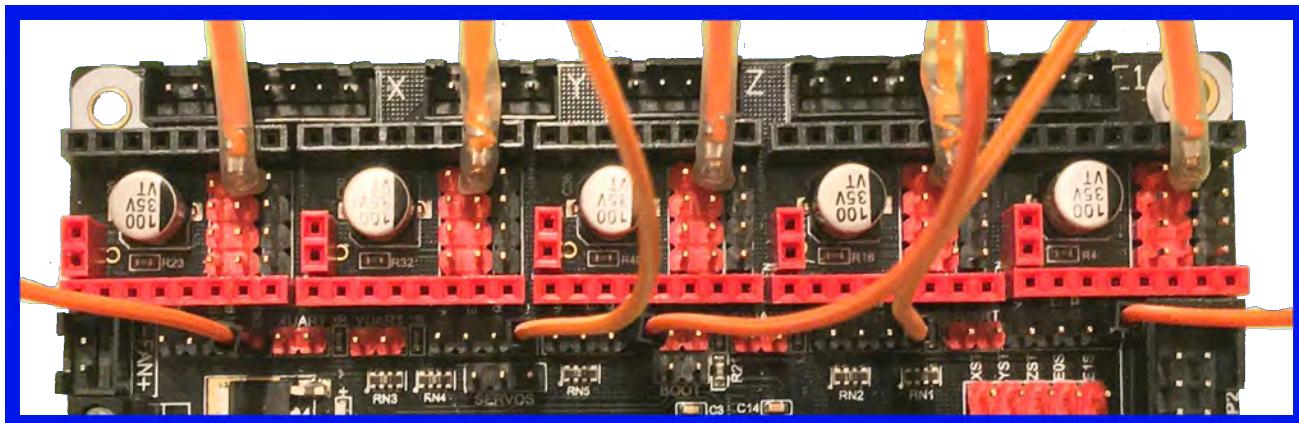
SpreadCycle



See [Appendix D](#)
for legend

BIQU TMC2100

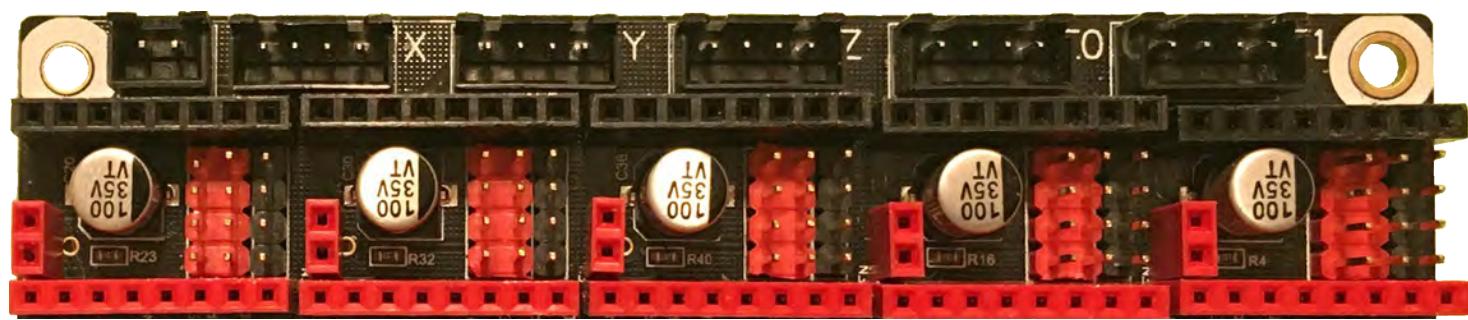
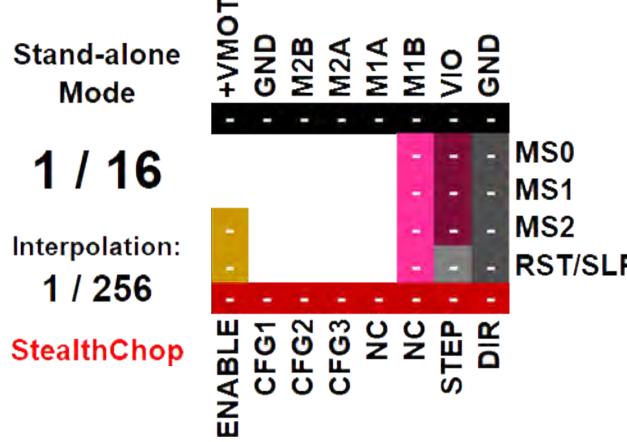
Stand-alone Mode



Stand-alone Mode

BIGU TMC2100

Stand-alone Mode



See [Appendix D](#) for legend

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

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

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

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

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code
- Explorer Bar (Left):** Shows the project structure under MARLIN-2.0.X, including files like LCDprint.cpp, thermistornames.h, and Configuration.h.
- Editor Area (Center):** Displays the content of Configuration.h. A green box highlights the section for E0_DRIVER_TYPE, which contains comments about stepper driver timing and options like A4988, A5984, DRV8825, etc. The code also includes sections for X, Y, Z, and E drivers.
- Right Side:** Shows the GitHub Pull Requests sidebar with several open pull requests from various contributors.

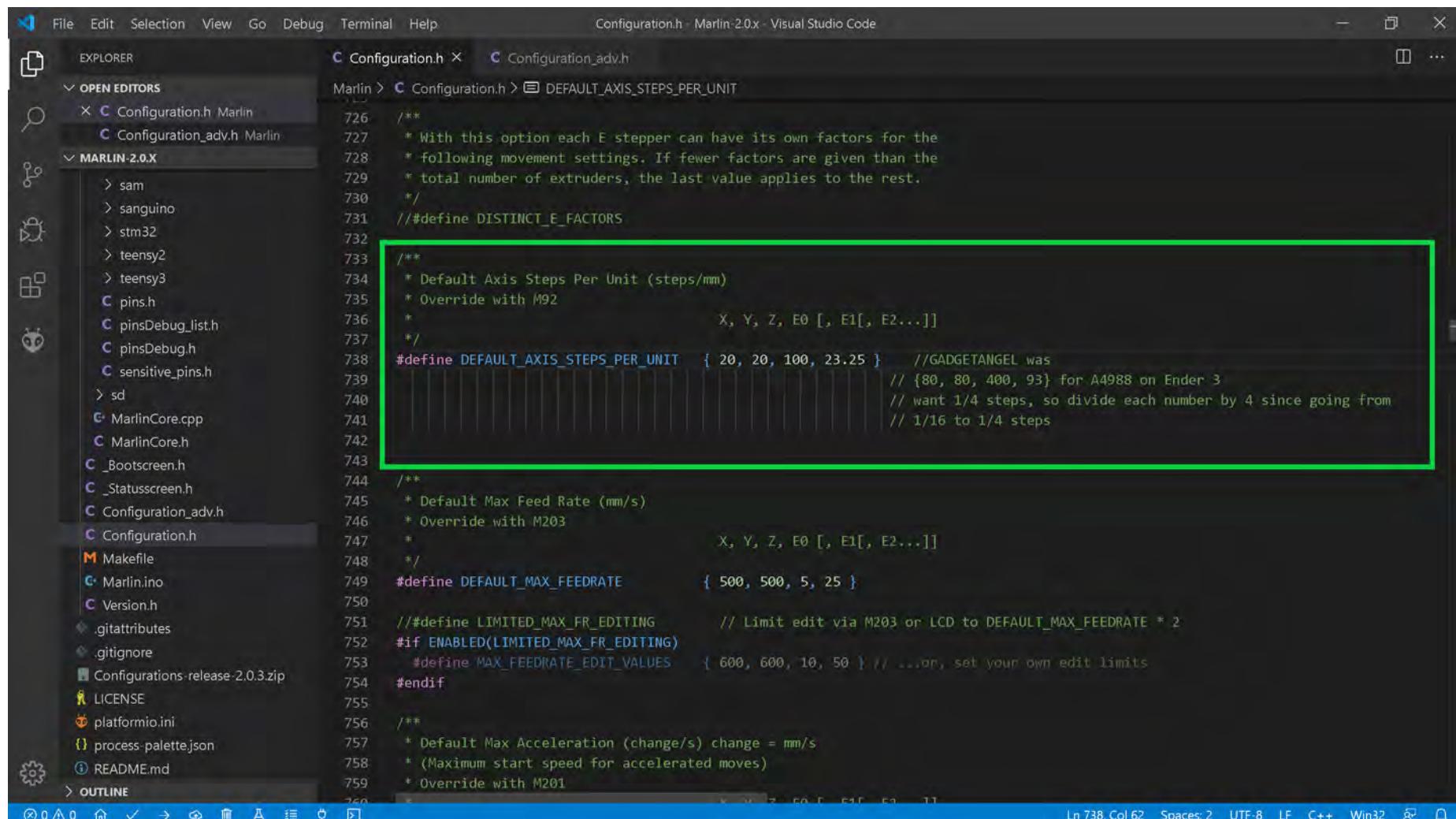
```
Marlin > C Configuration.h > E0_DRIVER_TYPE

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

- Go to the next page.

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

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



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

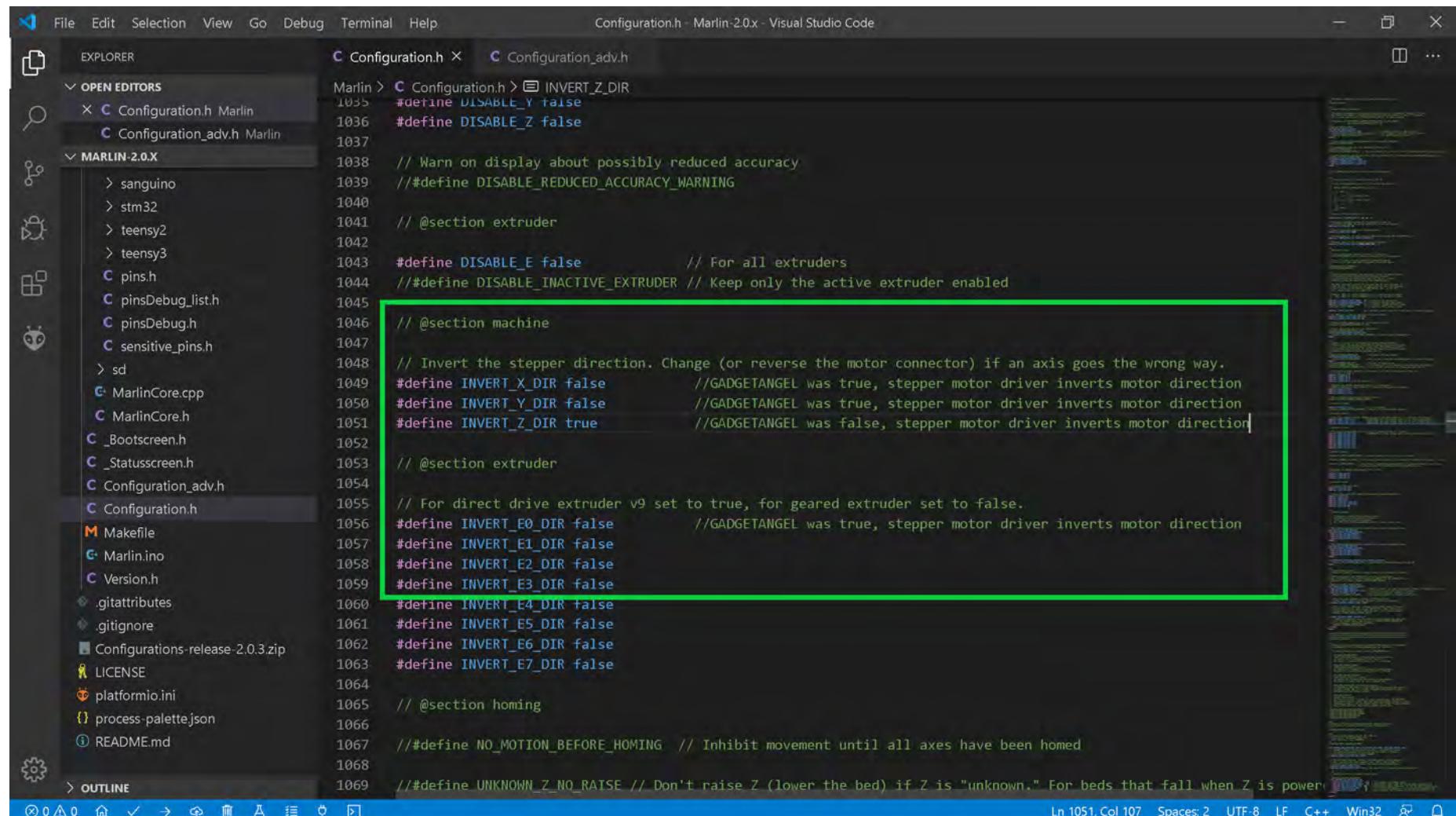
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps
```

The code editor's status bar at the bottom indicates 'Ln 738, Col 62'. The left sidebar shows the project structure with files like Configuration.h, Configuration_adv.h, and various Marlin 2.0.x source files.

- Go to the next page.

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

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



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a dark theme. On the left, the Explorer sidebar shows various Marlin files and folders. The main code area displays the following configuration snippet:

```

1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 //##define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 //##define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        //GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         //GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       //GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 //##define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 //##define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered on

```

A green rectangular box highlights the section of code from line 1048 to line 1059, which defines the inverting of stepper directions for extruders. This is the specific part of the code being discussed in the text above.

- Go to the next page.

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

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



Visual Studio Code Screenshot:

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

- File Explorer:** Shows files like Configuration.h, pins_BTT_SKR_V1_3.h, pins_BTT_SKR_common.h, and Configuration_adv.h.
- Editor:** Displays the Configuration.h file with the following code snippet highlighted in green:


```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif
```
- Terminal:** Shows the compilation process and success message:

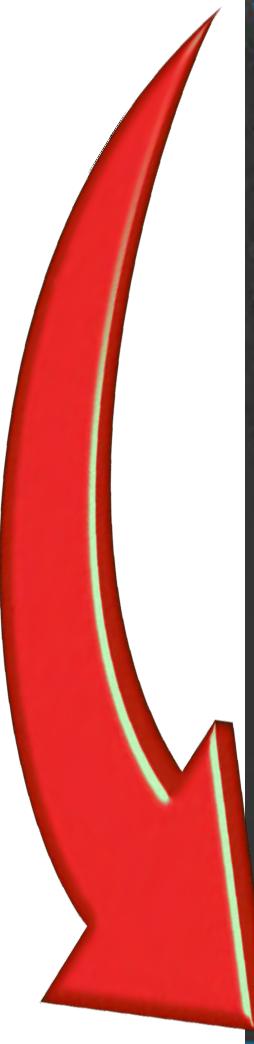

```
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
=====
===== [SUCCESS] Took 130.61 seconds =====
```
- Bottom Status Bar:** Shows build status for various boards:

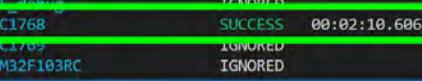
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
ramps	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

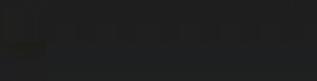
- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

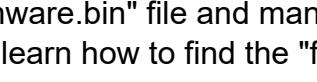




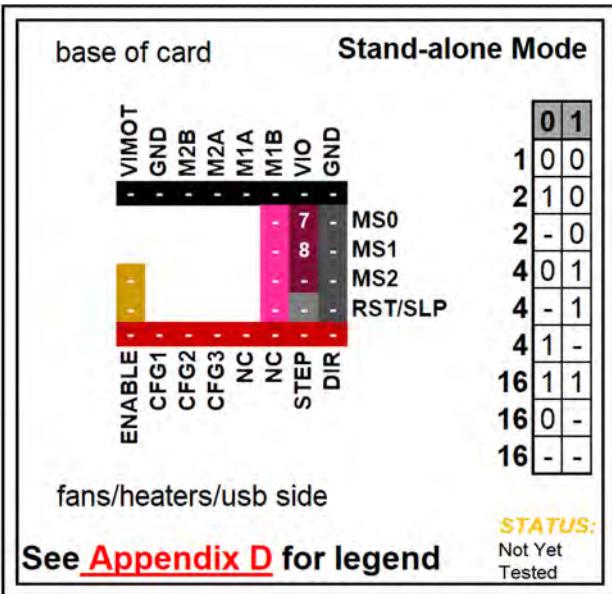








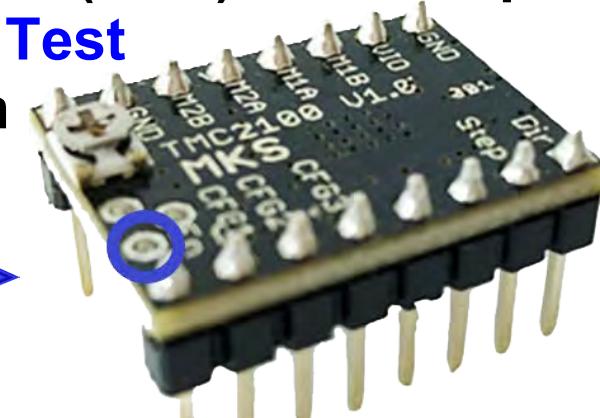


MKS TMC2100

Stand-alone Mode

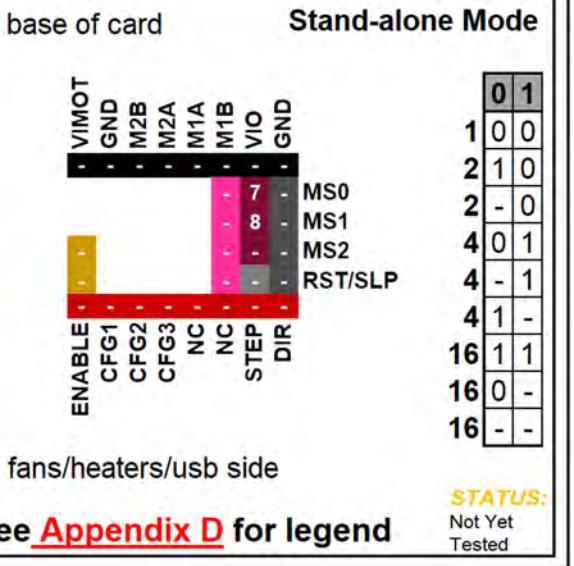
NOTE: Use the potentiometer (POT) on the top of the board or use the "V_{ref} Test point" location, as shown in BLUE, to adjust your driver board's V_{ref}



Driver Chip	MS0	MS1	Steps	Interpolation	Mode
Makerbase TMC2100 Stand Alone Mode	Low	Low	1	NONE	spreadCycle
	High	Low	1 / 2	NONE	spreadCycle
	OPEN	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	NONE	spreadCycle
	OPEN	High	1 / 4	1 / 256	spreadCycle
	High	OPEN	1 / 4	1 / 256	stealthChop
	High	High	1 / 16	NONE	spreadCycle
	Low	OPEN	1 / 16	1 / 256	spreadCycle
	OPEN	OPEN	1 / 16	1 / 256	stealthChop

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

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

**Driver Chip Chart:**

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO	Low	Low	1	NONE	spreadCycle
TMCxxxx	High	Low	1/2	NONE	spreadCycle
Stand Alone Mode	OPEN	Low	1/2	1/256	spreadCycle
Maximum 16 Subdivision	Low	High	1/4	NONE	spreadCycle
46V DC	OPEN	High	1/4	1/256	spreadCycle
2.5A (peak)	High	OPEN	1/4	1/256	stealthChop
Driving Current Calculation Formula	High	High	1/16	NONE	spreadCycle
R_s (typical Sense Resistor)= 0.11Ω	Low	OPEN	1/16	1/256	spreadCycle
	OPEN	OPEN	1/16	1/256	stealthChop

Driving Current Calculation Formula
 $I_{MAX}=V_{ref}$
See Appendix B #x. Use 50% to 90% as shown below:
 $I_{MAX}=I_{MAX} * 0.90$

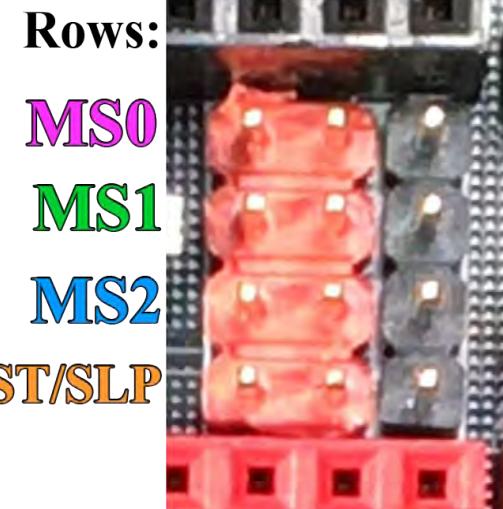
Driving Current Calculation Formula
 $V_{ref}=I_{MAX}$
See Appendix B #x. Use 50% to 90% as shown below:
 $V_{ref}=V_{ref} * 0.90$

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

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

MKS TMC2100**Stand-alone Mode****SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1**

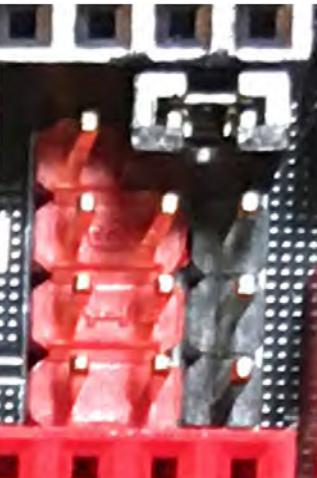
OPEN ➡ **No Jumper**



High ➡ **Jumper Set**

Rows:

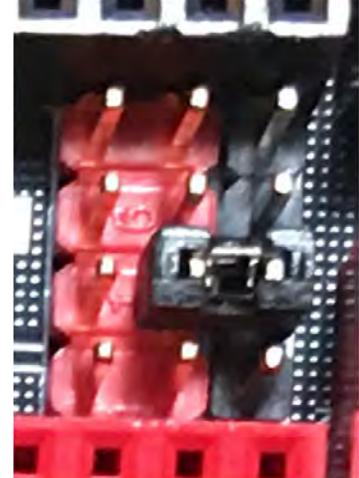
MS0



MS1



MS2



RST/SLP

MS0 SET HIGH

MS1 SET HIGH

MS2 SET HIGH

Driver Chip Chart:

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

Driving Current Calculation Formula

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

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

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

$V_{ref} = I_{MAX}$

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

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

R_S (typical Sense Resistor) = 0.11Ω

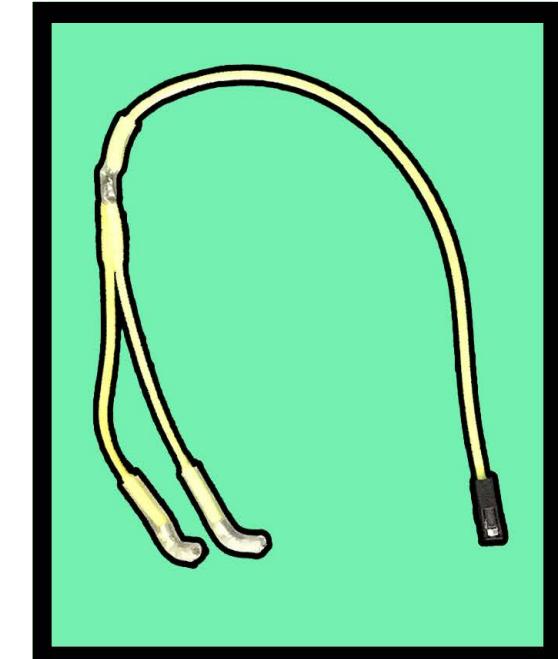
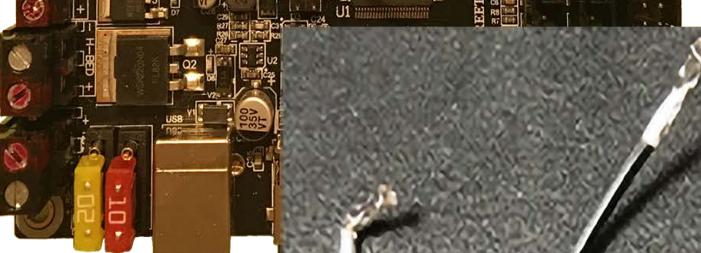
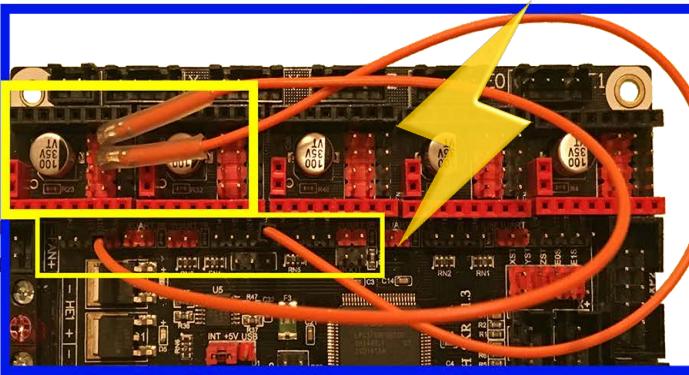
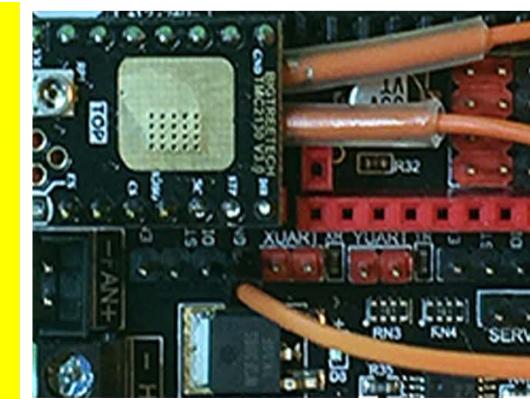
Low ➔
MS0 or MS1
connected to
Ground(GND)
via DuPont
Jumper Cable

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

MKS TMC2100

Stand-alone Mode

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

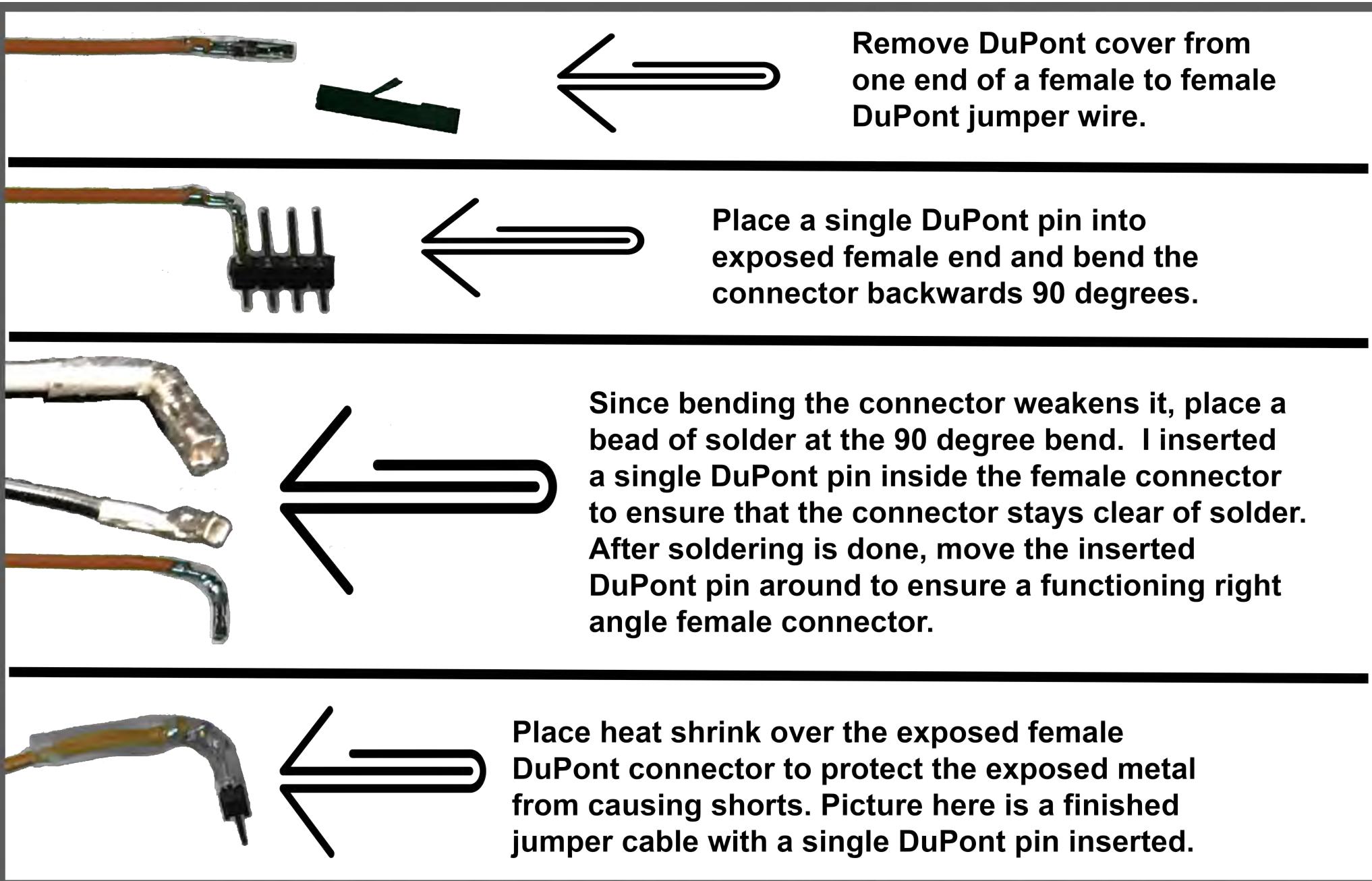


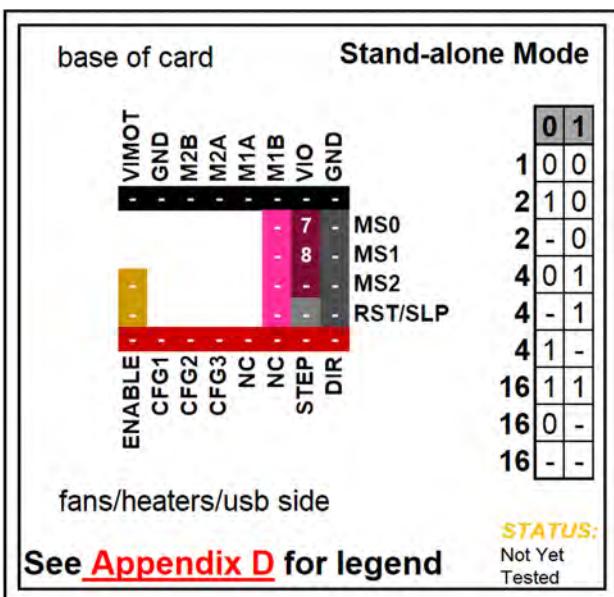
Stand-alone Mode

MKS TMC2100

Stand-alone Mode

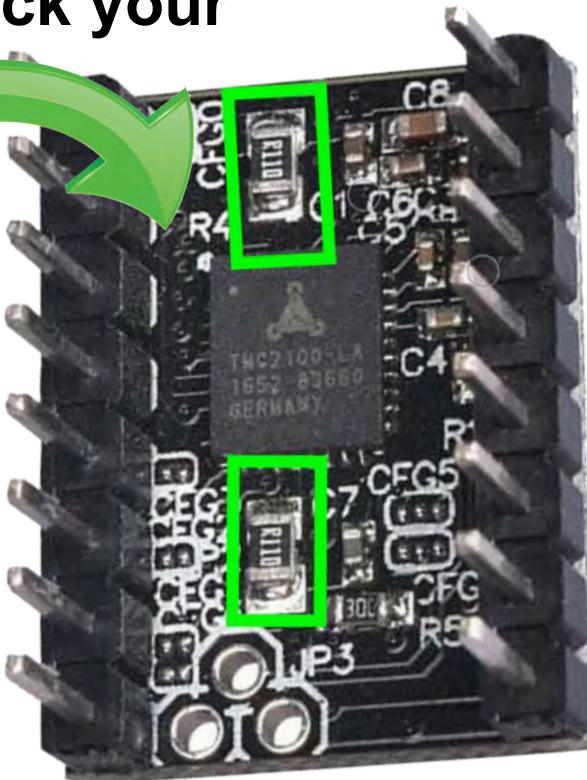
How to Create a SKR V1.3 Dupont Jumper Cable to Use with Tri State Drivers





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

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

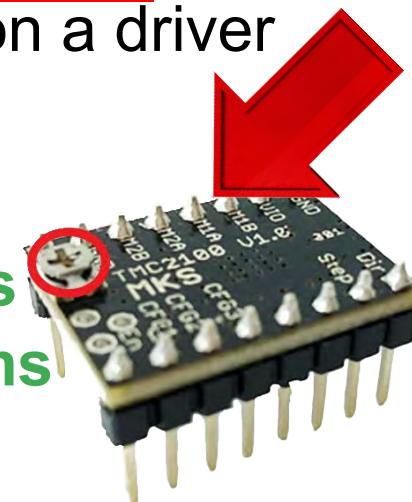


MKS TMC2100

Stand-alone Mode

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

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



$R_s = R050$ is 0.05 Ohms

$R_s = R068$ is 0.068 Ohms

$R_s = R100$ is 0.1 Ohms

$R_s = R150$ is 0.15 Ohms

$R_s = R200$ is 0.2 Ohms

$R_s = R220$ is 0.22 Ohms

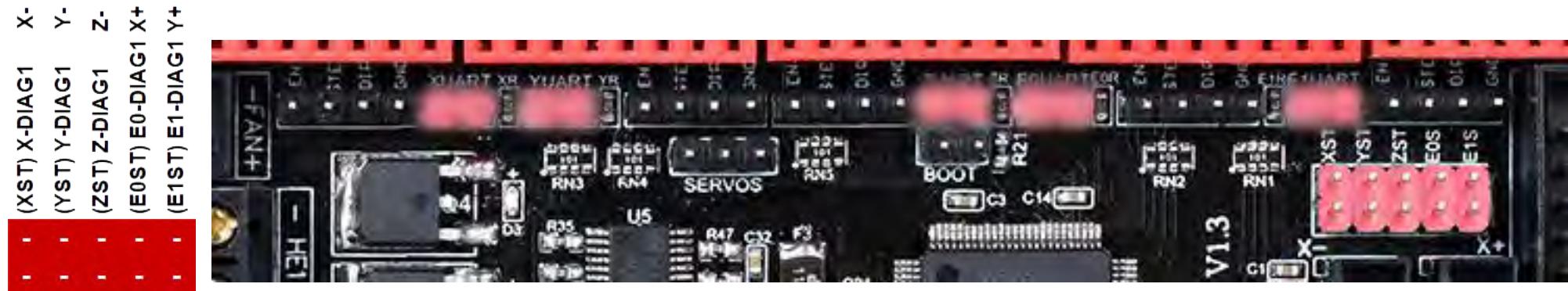
Note: See this video about current sense resistors (R_s) and their possible locations:
<https://youtu.be/8wk1elugv5A>

Stand-alone Mode

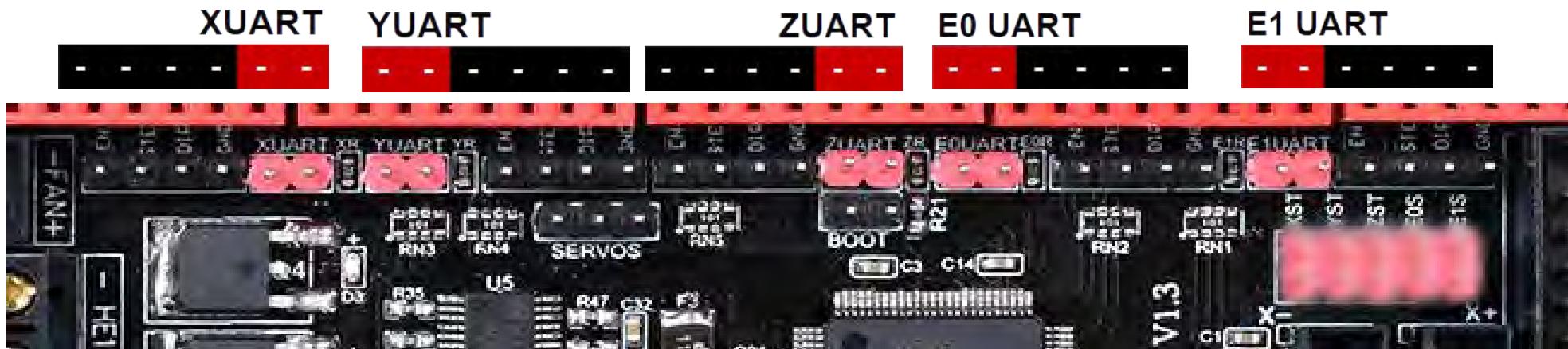
MKS TMC2100

Stand-alone Mode

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



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



Stand-alone Mode

MKS TMC2100

Stand-alone Mode

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

Driver Socket Representation

Columns: 1 2 3 Rows: MS0 MS1 MS2 RST/SLP

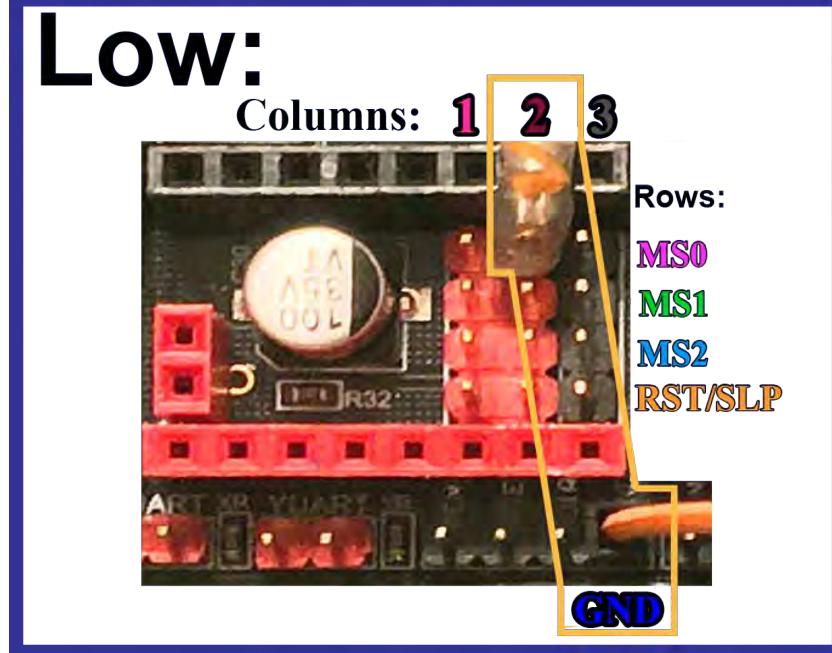
Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode Maximum 16 Subdivision 48V DC 2.5A (peak)	Low	Low	1	NONE	spreadCycle
	High	Low	1/2	NONE	spreadCycle
	OPEN	Low	1/2	1/256	spreadCycle
	LOW	High	1/4	NONE	spreadCycle
	OPEN	High	1/4	1/256	spreadCycle
	High	OPEN	1/4	1/256	stealthChop
	High	High	1/16	NONE	spreadCycle
	LOW	OPEN	1/16	1/256	spreadCycle

Driving Current Calculation Formula: $I_{MAX} = V_{ref}$ $V_{ref} = I_{MAX}$

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

No Jumper set



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

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

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

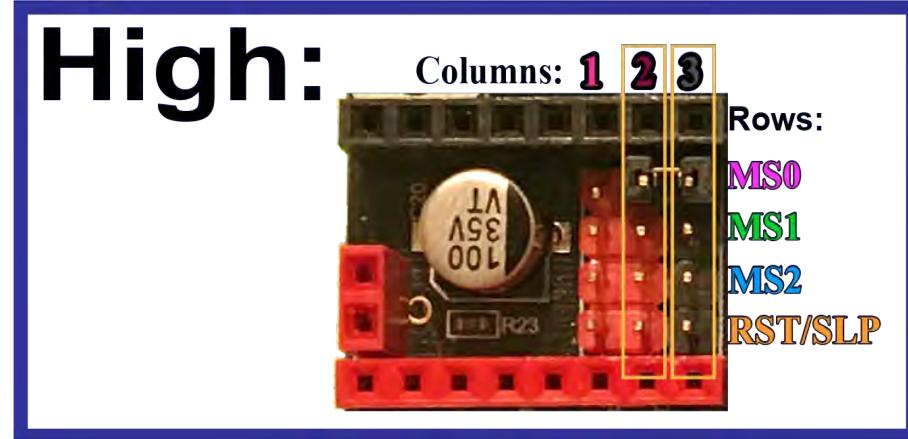
Driver Socket Representation: - - → **Driver Chip Chart:** Open

Meaning:

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

Low → Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)

Open → No Jumper set

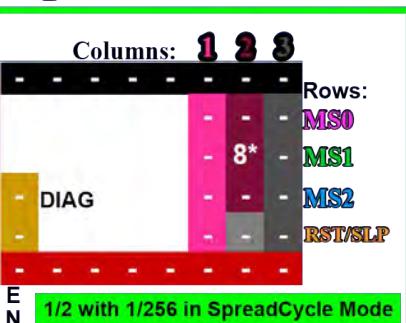
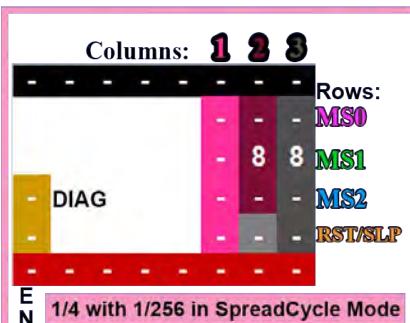


Stand-alone Mode

MKS TMC2100

Stand-alone Mode

Driver Socket Representation

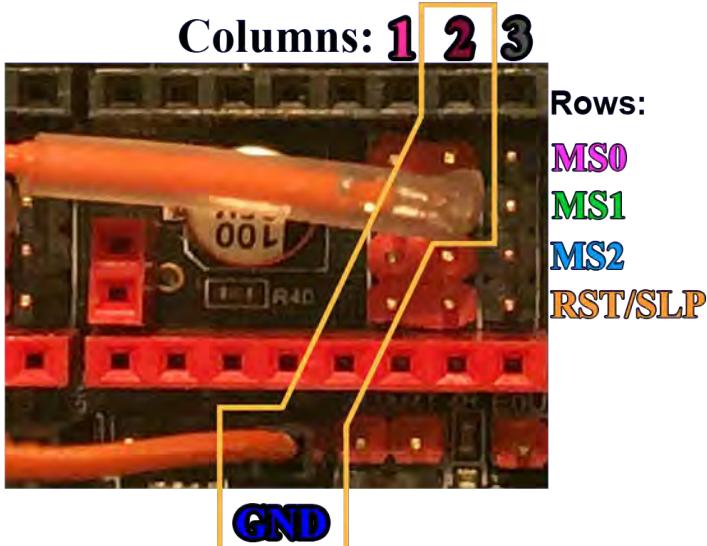


Driver Chip Chart:

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



LOW:



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



Driver Chip Chart:

High

Low

Open

Meaning:

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

Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)

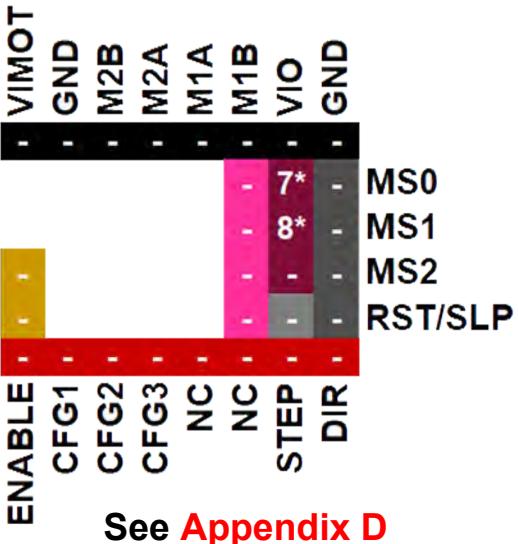
No Jumper set

High:



Stand-alone Mode

Stand-alone
Mode

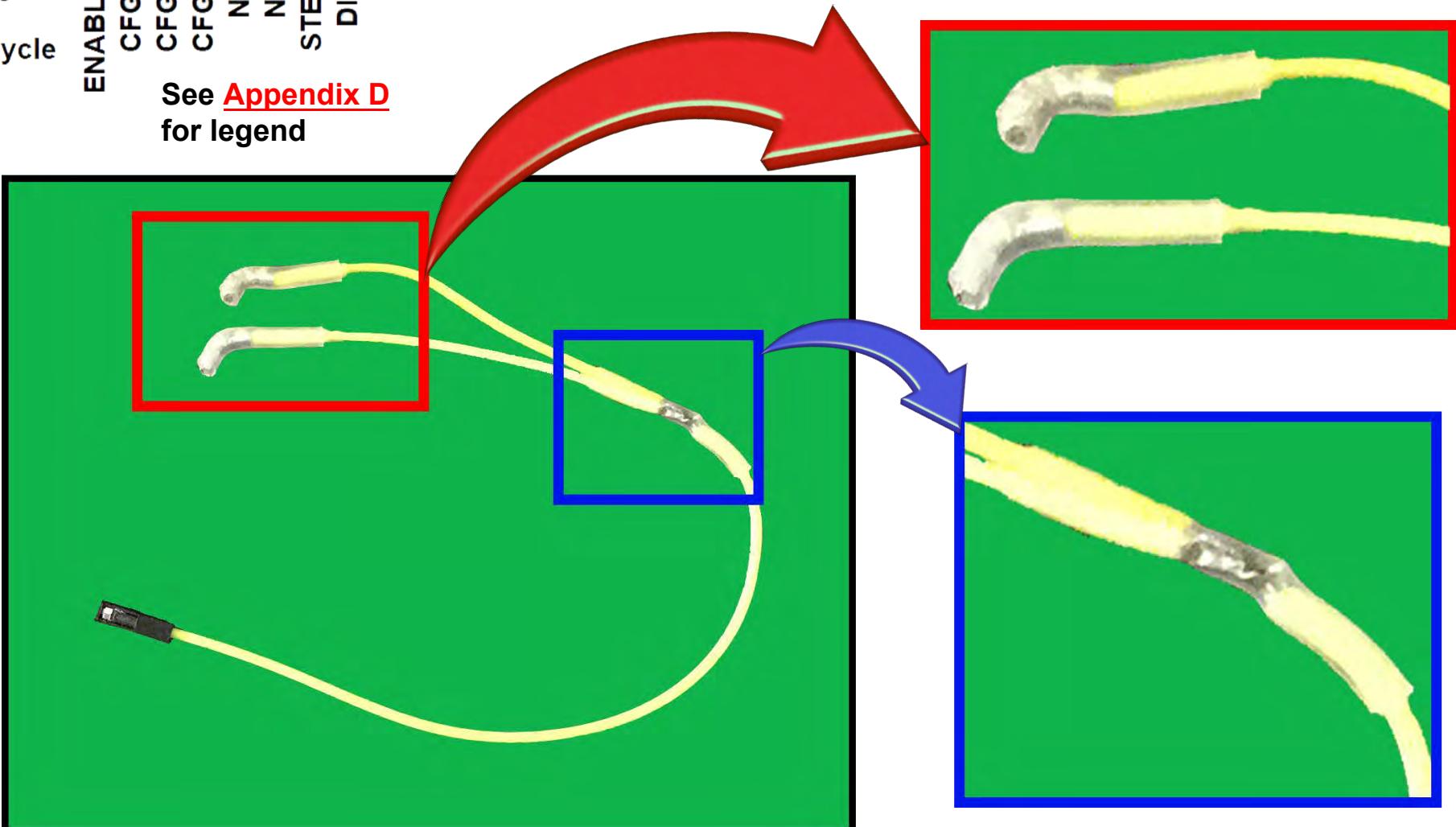


MKS TMC2100

Stand-alone Mode

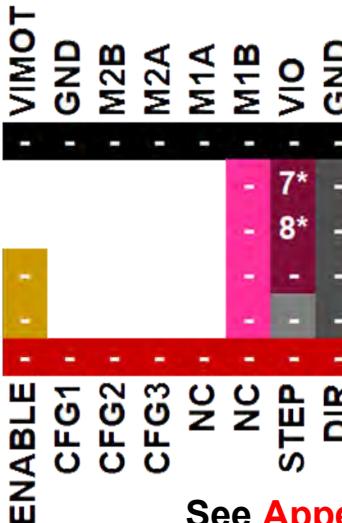
Additional Equipment Needed for Low Low (STEP or FULL) Configuration

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



Stand-alone Mode

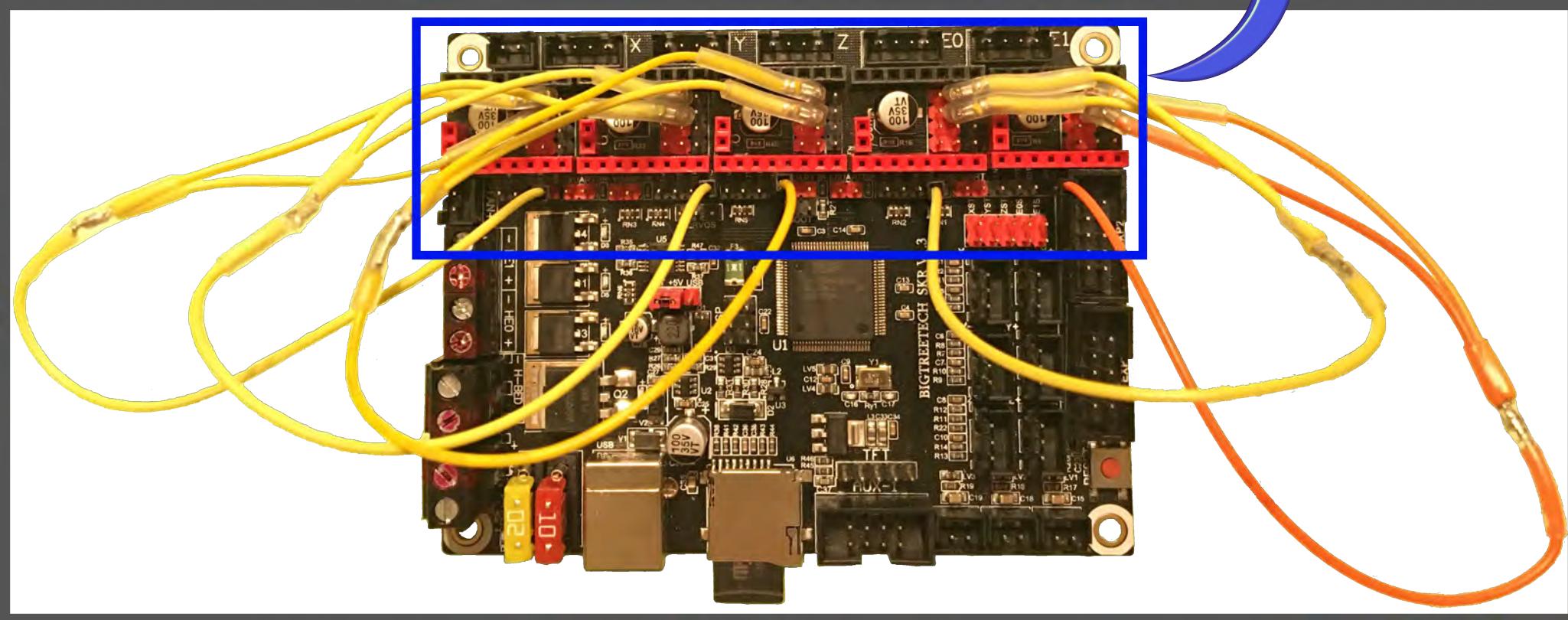
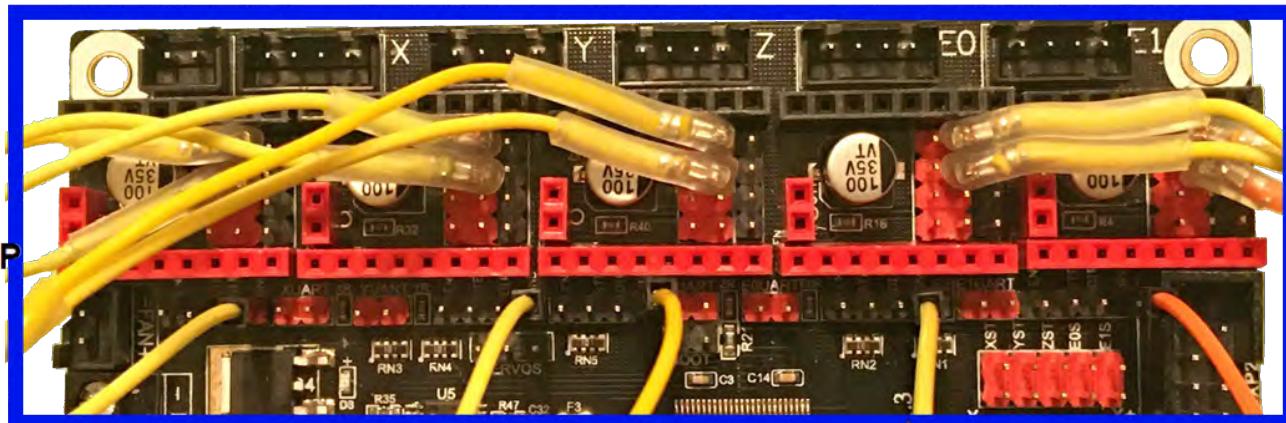
Stand-alone
Mode



See [Appendix D](#)
for legend

MKS TMC2100

Stand-alone Mode

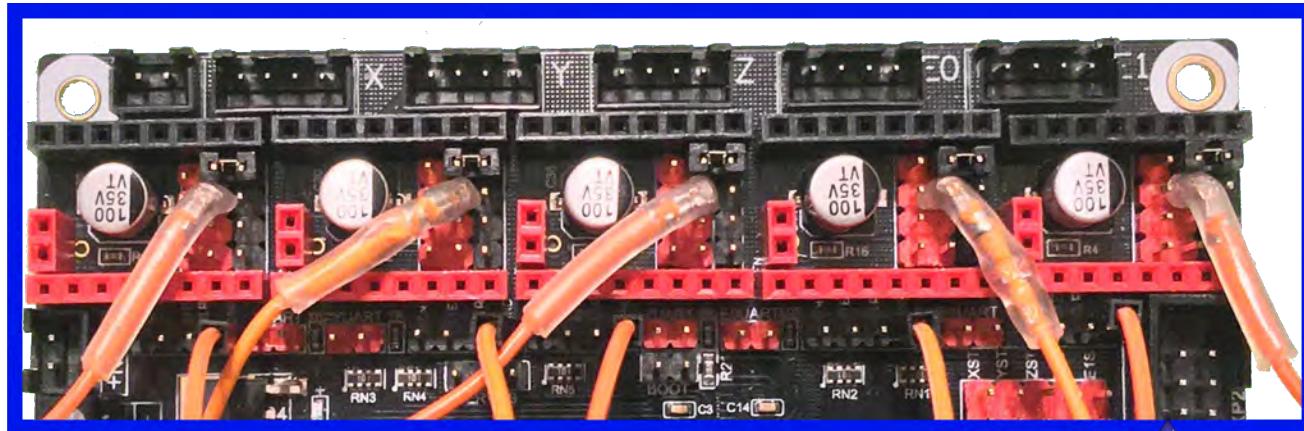


Stand-alone Mode

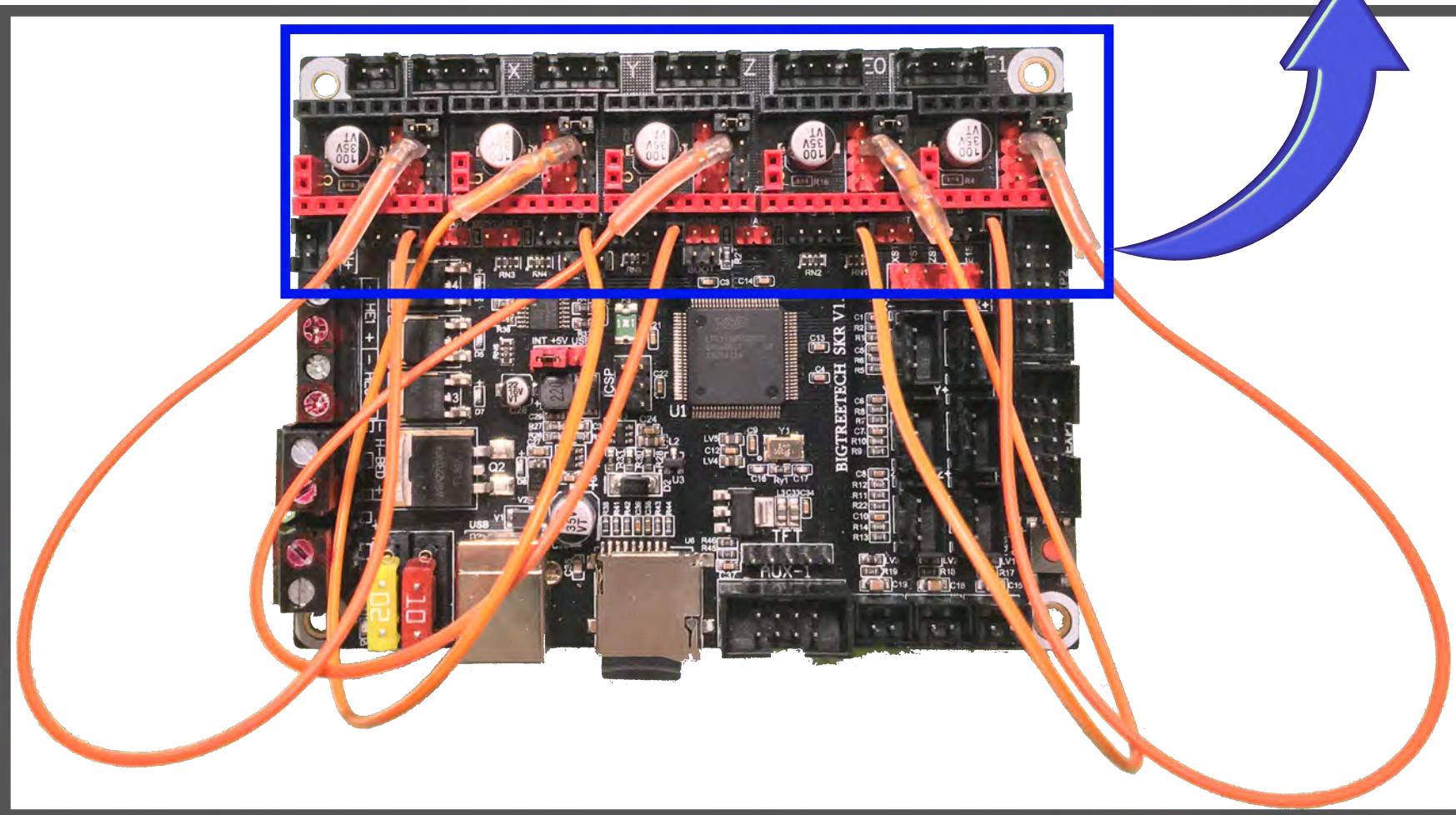
Stand-alone Mode	VIMOT	GND	M2B	M2A	M1A	M1B	VIO	GND
1 / 2					7	7	MS0	
Interpolation:					8*	-	MS1	
none						-	MS2	
SpreadCycle	ENABLE		CFG1	CFG2	CFG3	NC	NC	STEP DIR

MKS TMC2100

Stand-alone Mode



See [Appendix D](#)
for legend



Stand-alone Mode

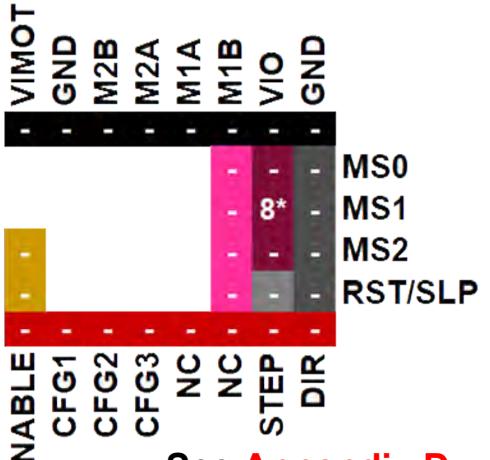
Stand-alone
Mode

1 / 2

Interpolation:

1 / 256

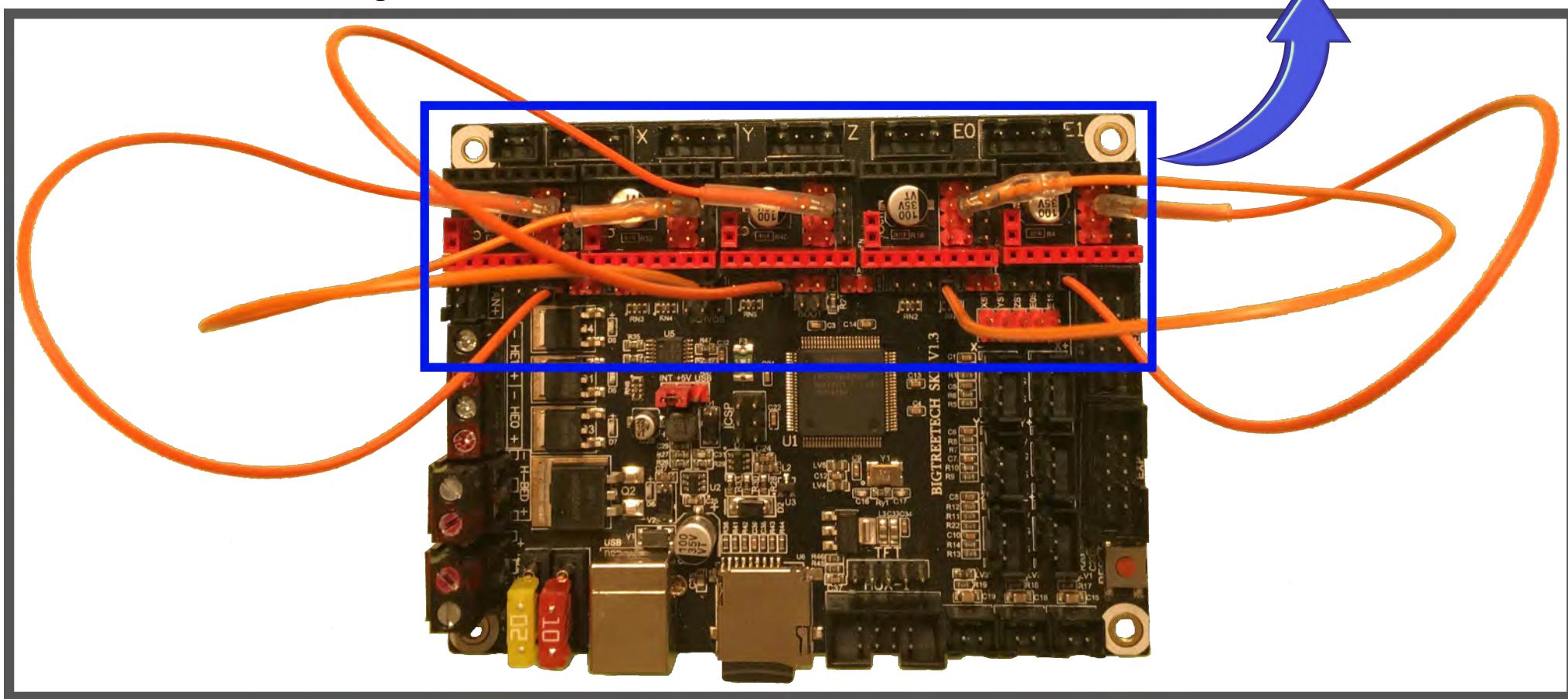
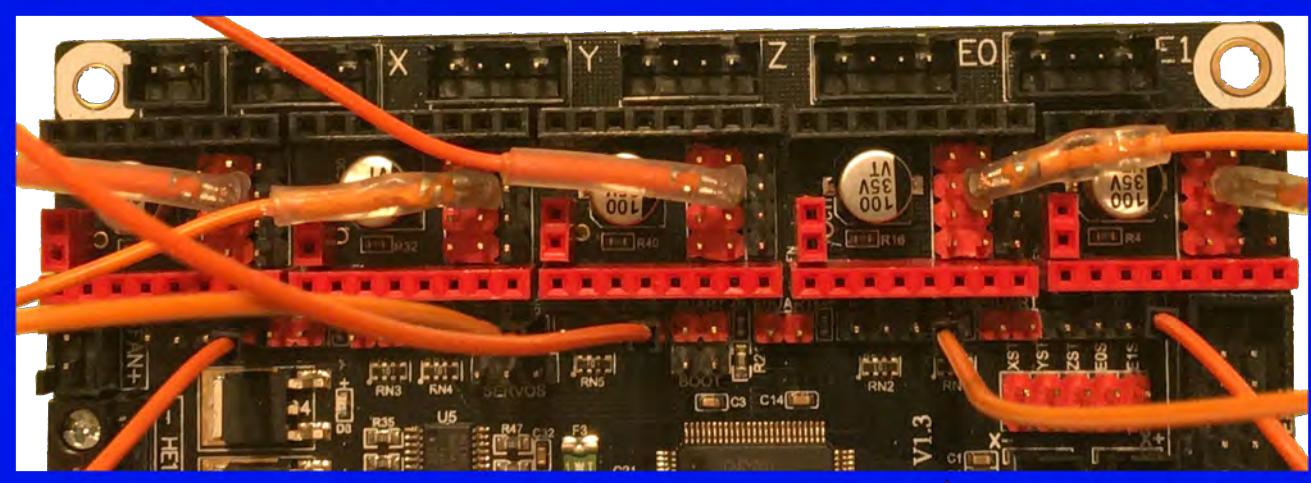
SpreadCycle



See [Appendix D](#)
for legend

MKS TMC2100

Stand-alone Mode



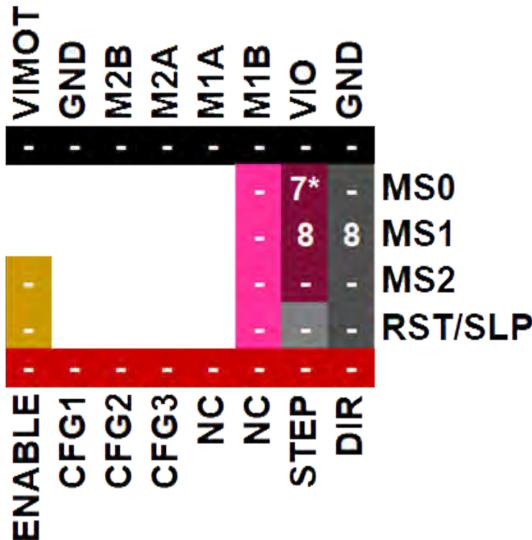
Stand-alone Mode

Stand-alone
Mode

1 / 4

Interpolation:
none

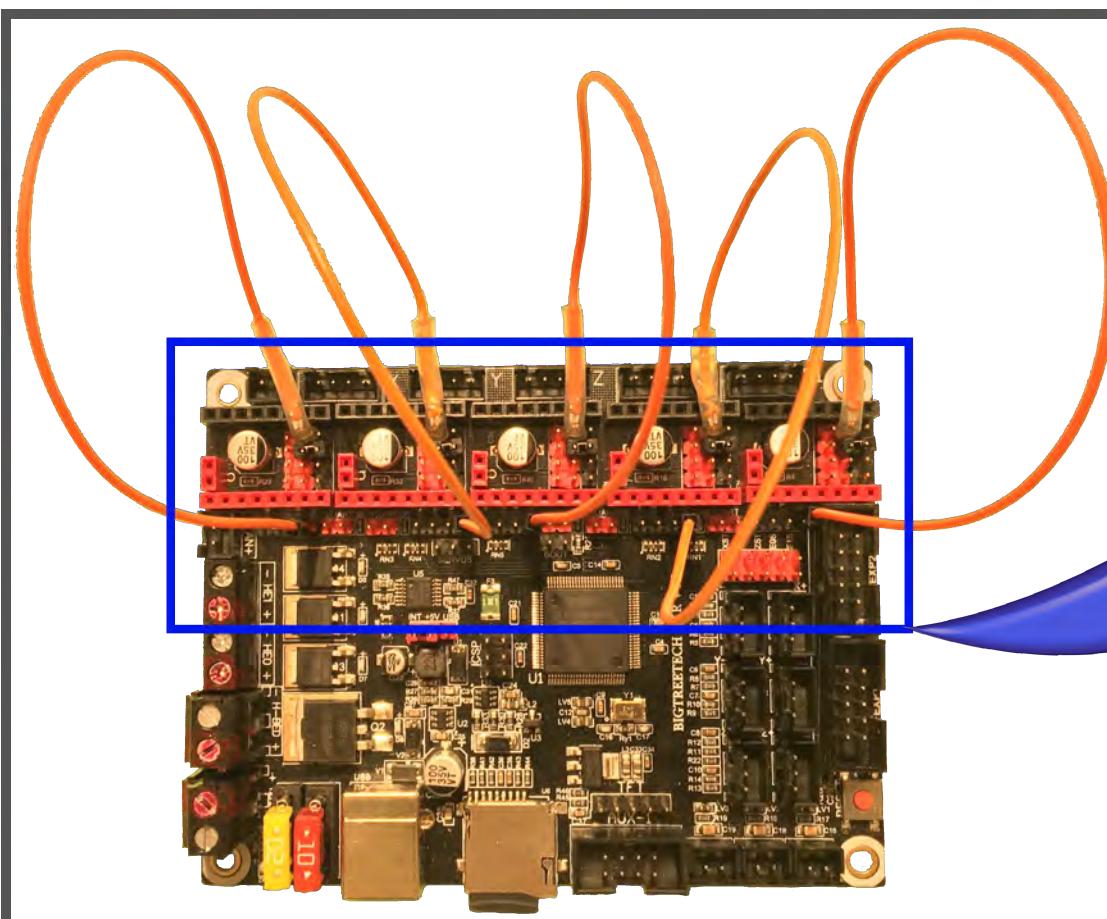
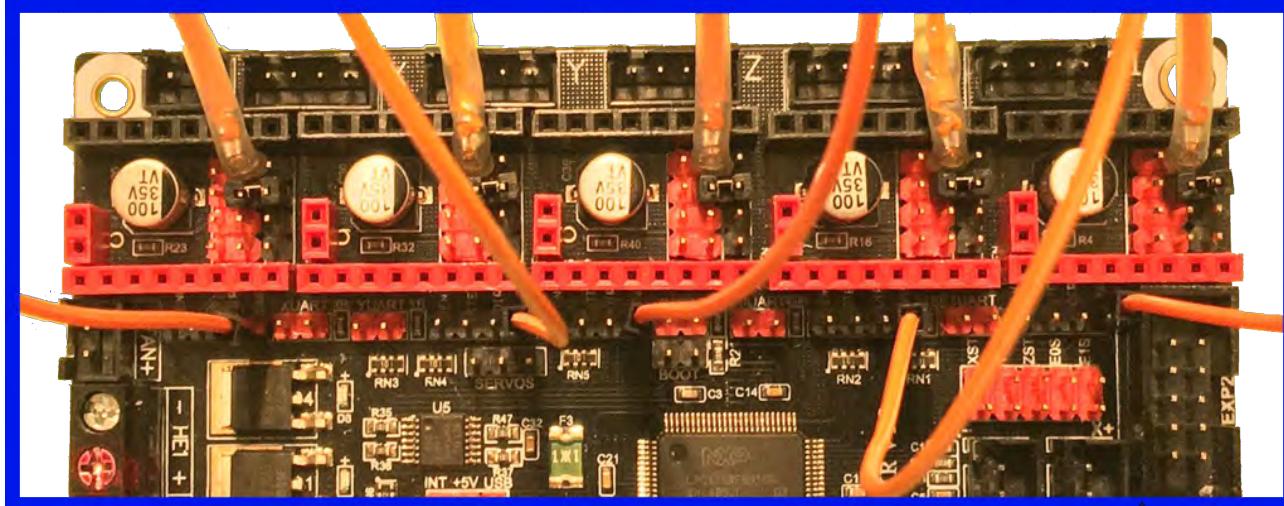
SpreadCycle



See [Appendix D](#)
for legend

MKS TMC2100

Stand-alone Mode

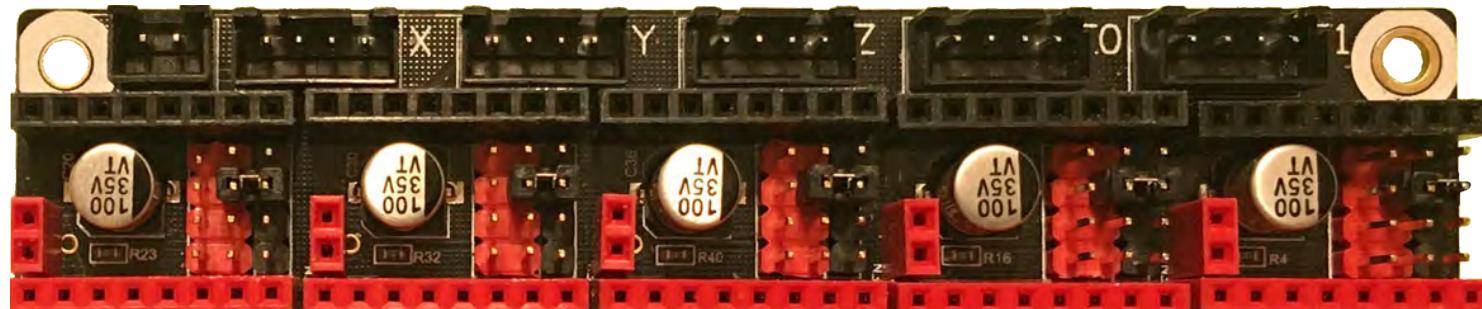


Stand-alone Mode

MKS TMC2100

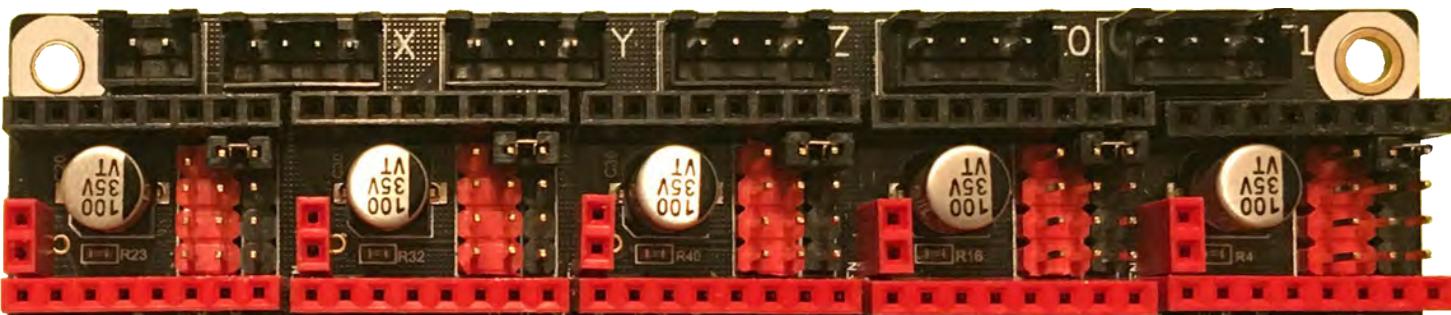
Stand-alone Mode

Stand-alone Mode	VIMOT	GND	M2B	M2A	M1A	M1B	VIO	GND
1 / 4							MS0	
Interpolation:					MS1			
1 / 256					MS2			
SpreadCycle	ENABLE	CFG1	CFG2	CFG3	NC	NC	STEP	DIR



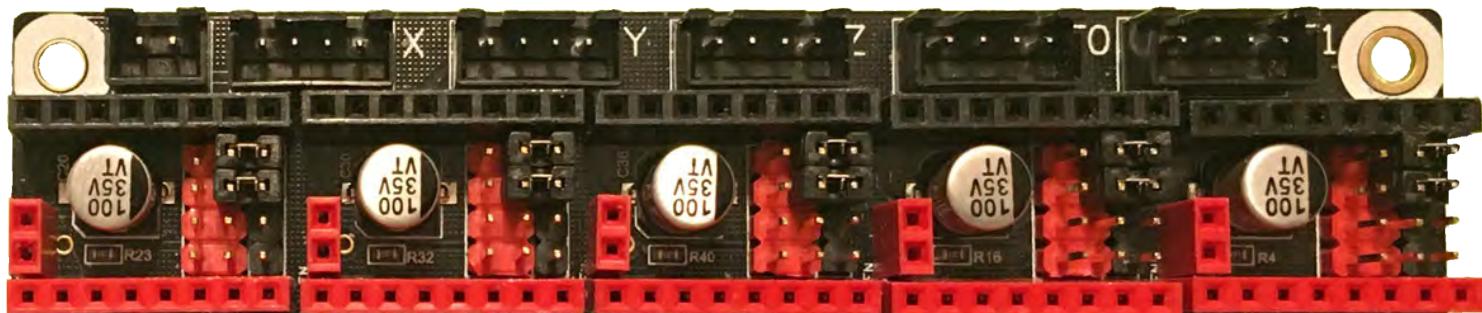
See [Appendix D](#) for legend

Stand-alone Mode	VIMOT	GND	M2B	M2A	M1A	M1B	VIO	GND
1 / 4							MS0	
Interpolation:					MS1			
1 / 256					MS2			
StealthChop	ENABLE	CFG1	CFG2	CFG3	NC	NC	STEP	DIR



See [Appendix D](#) for legend

Stand-alone Mode	VIMOT	GND	M2B	M2A	M1A	M1B	VIO	GND
1 / 16							MS0	
Interpolation:					MS1			
none					MS2			
SpreadCycle	ENABLE	CFG1	CFG2	CFG3	NC	NC	STEP	DIR



See [Appendix D](#) for legend

Stand-alone Mode

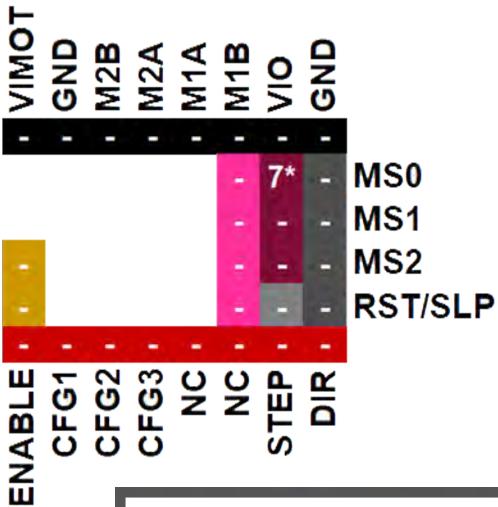
Stand-alone
Mode

1 / 16

Interpolation:

1 / 256

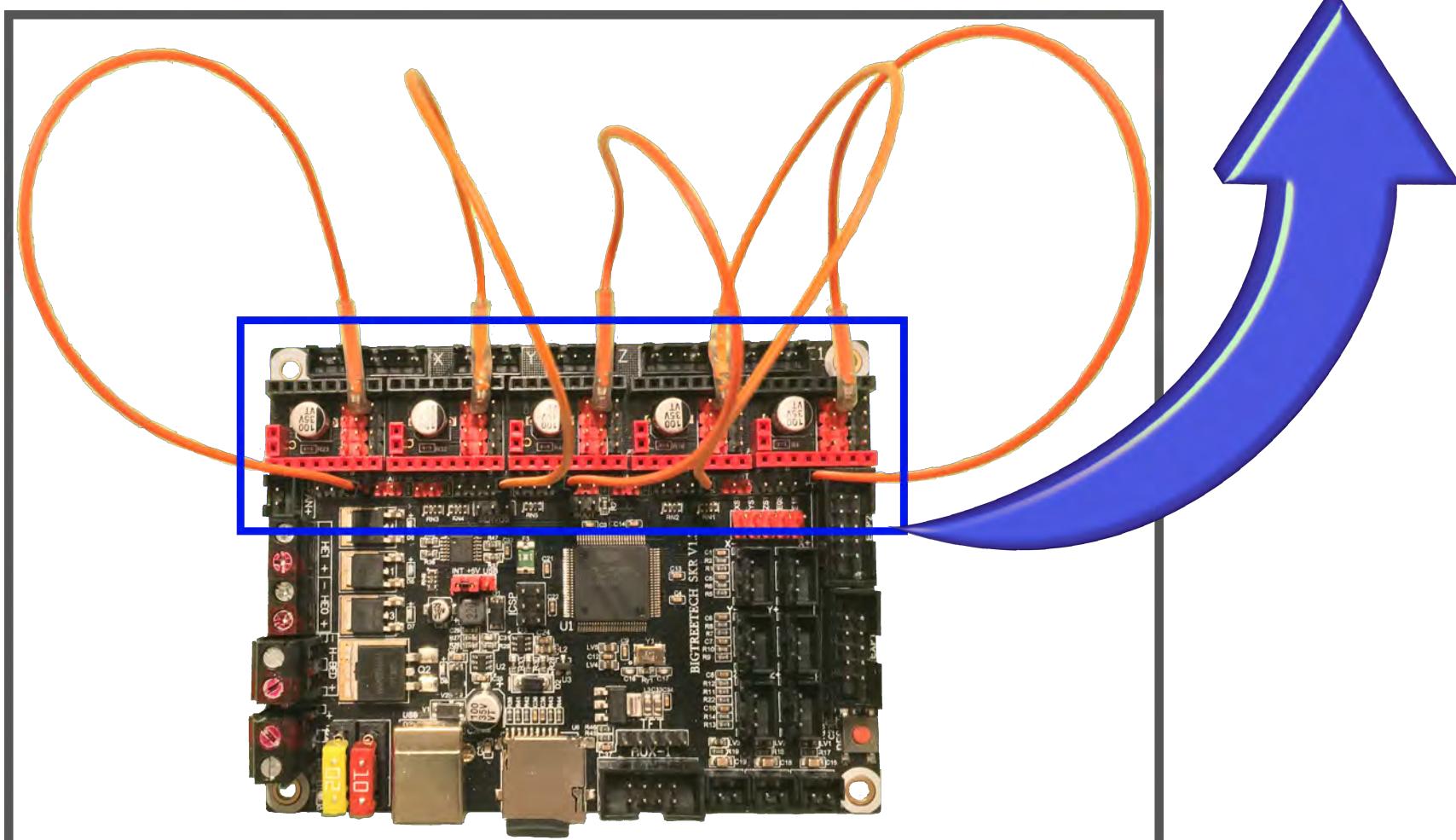
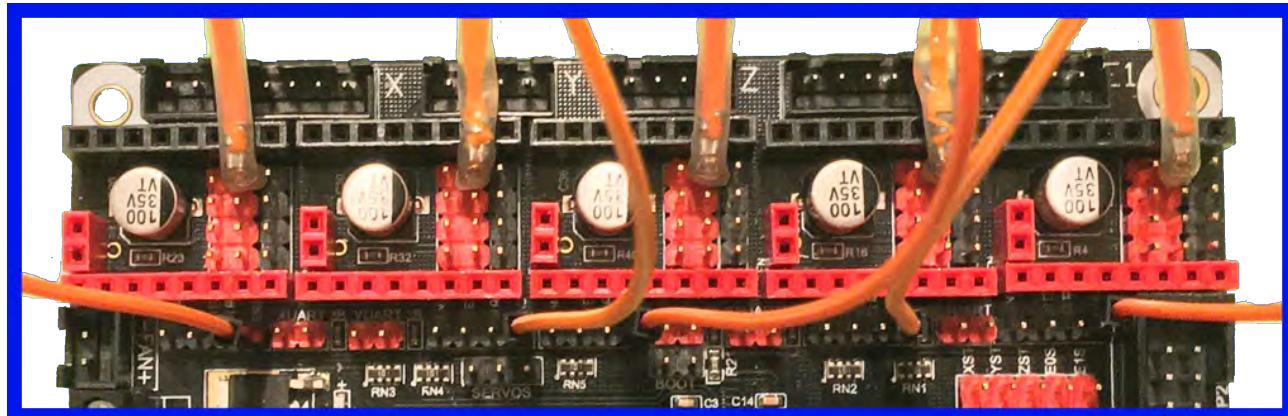
SpreadCycle



See [Appendix D](#)
for legend

MKS TMC2100

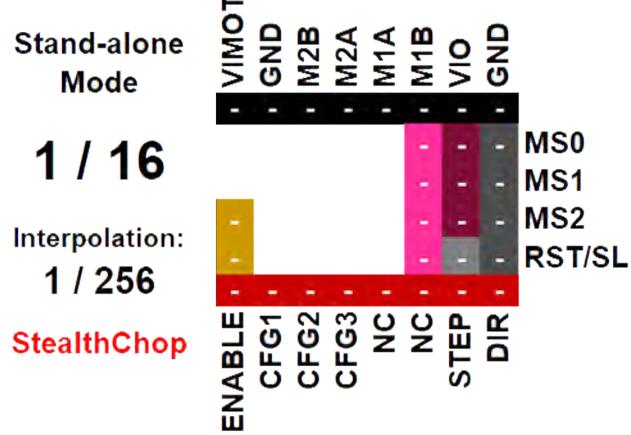
Stand-alone Mode



Stand-alone Mode

MKS TMC2100

Stand-alone Mode

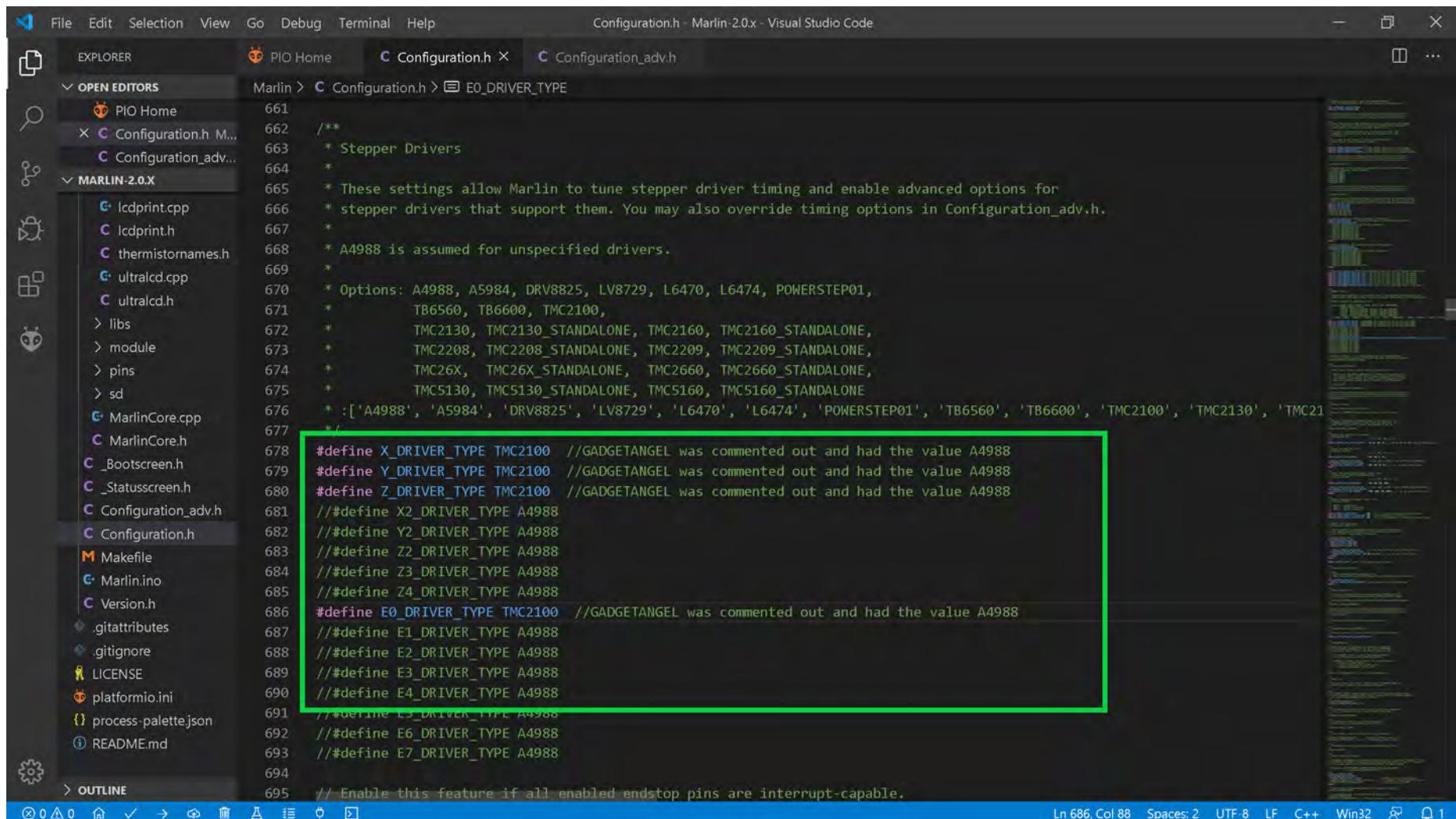


See [Appendix D](#) for legend

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

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

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



```

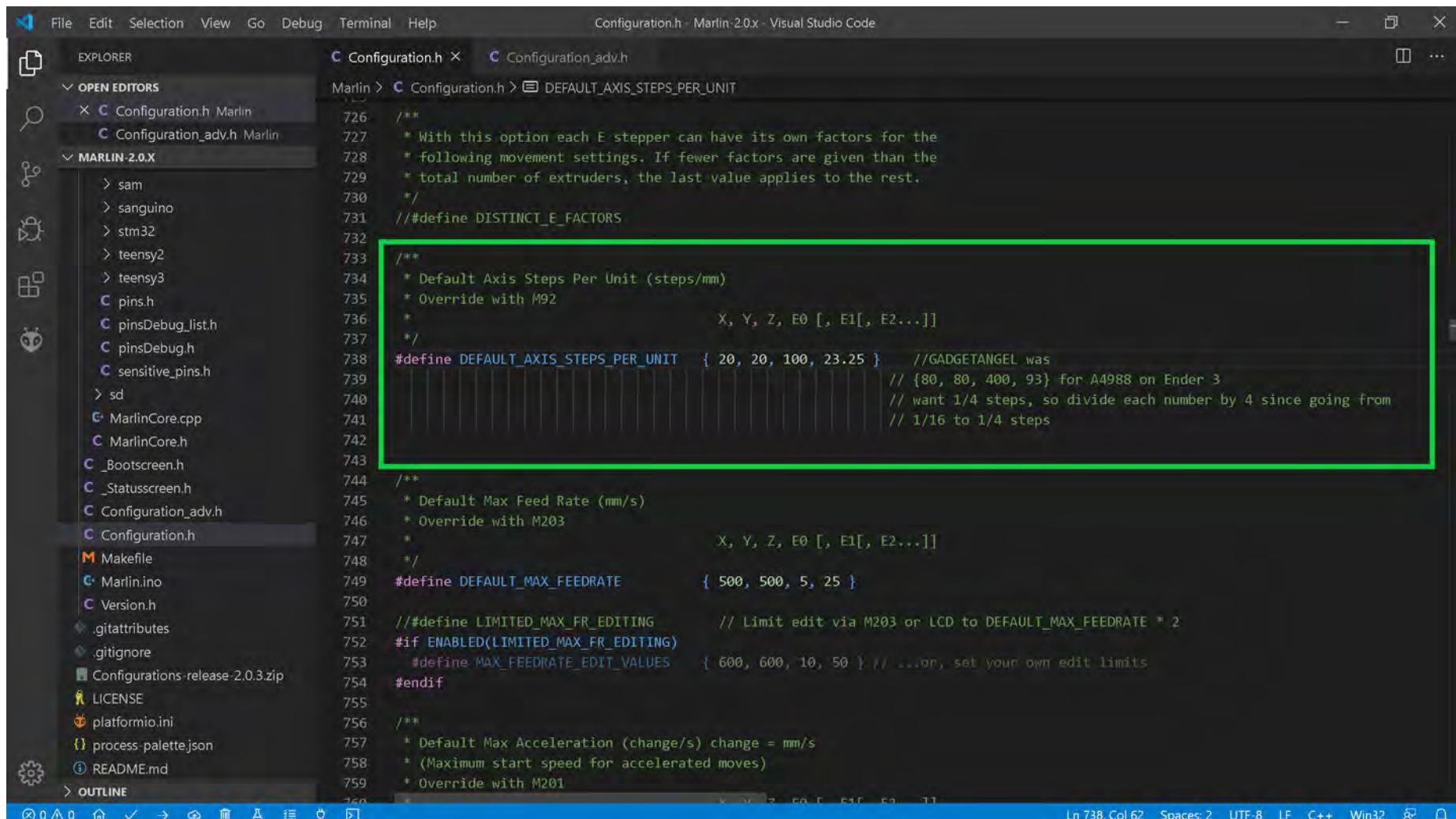
File Edit Selection View Go Debug Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2660']
676 */
677 #define X_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 //#define E0_DRIVER_TYPE TMC2100 //GADGETANGEL was commented out and had the value A4988
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693 // Enable this feature if all enabled endstop pins are interrupt-capable.
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

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

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



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

```

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

732 /**
733 * Default Axis Steps Per Unit (steps/mm)
734 * Override with M92
735 *
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742
743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

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

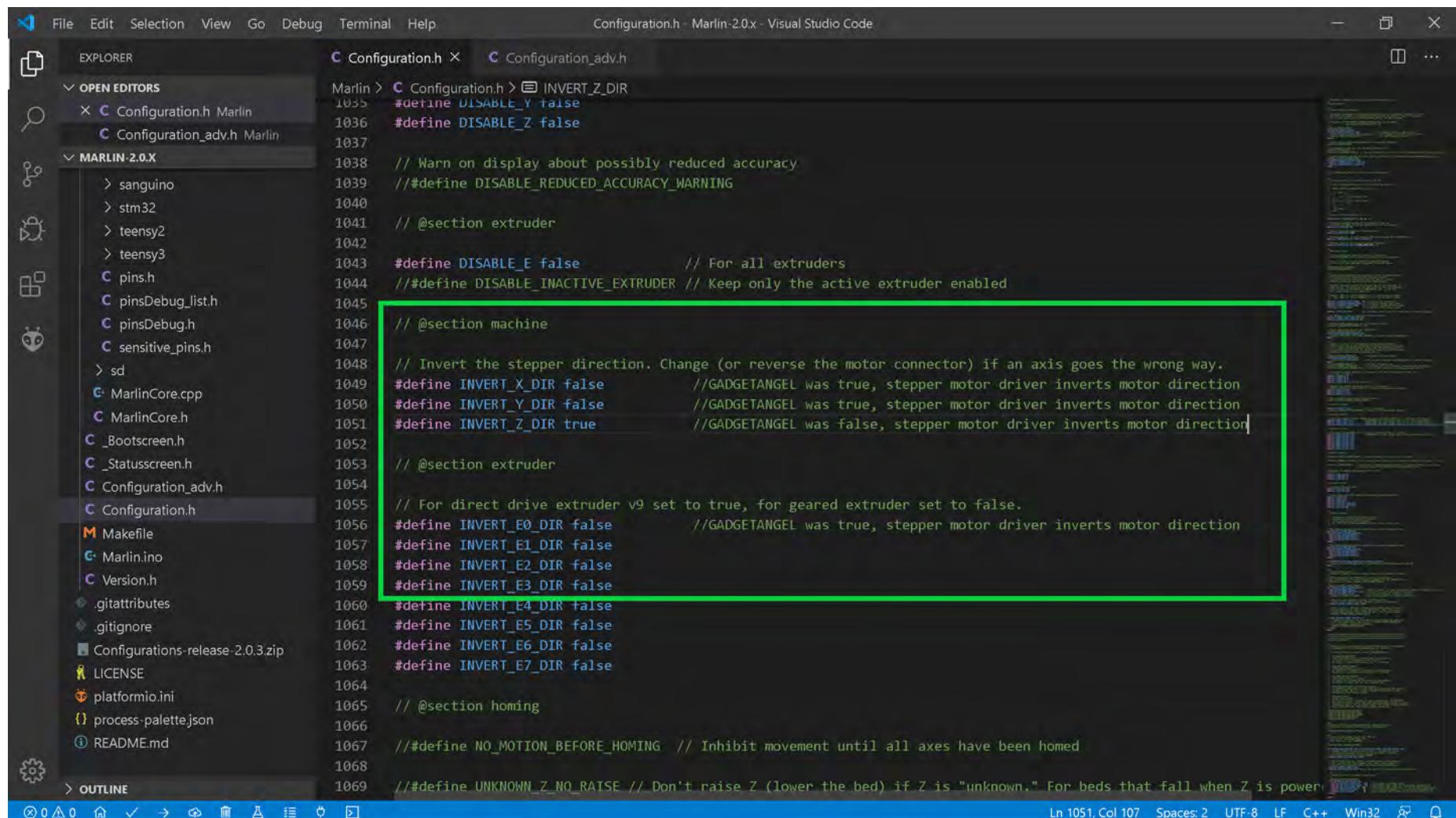
```

The code editor status bar at the bottom indicates: Ln 738, Col 62, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

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

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



```

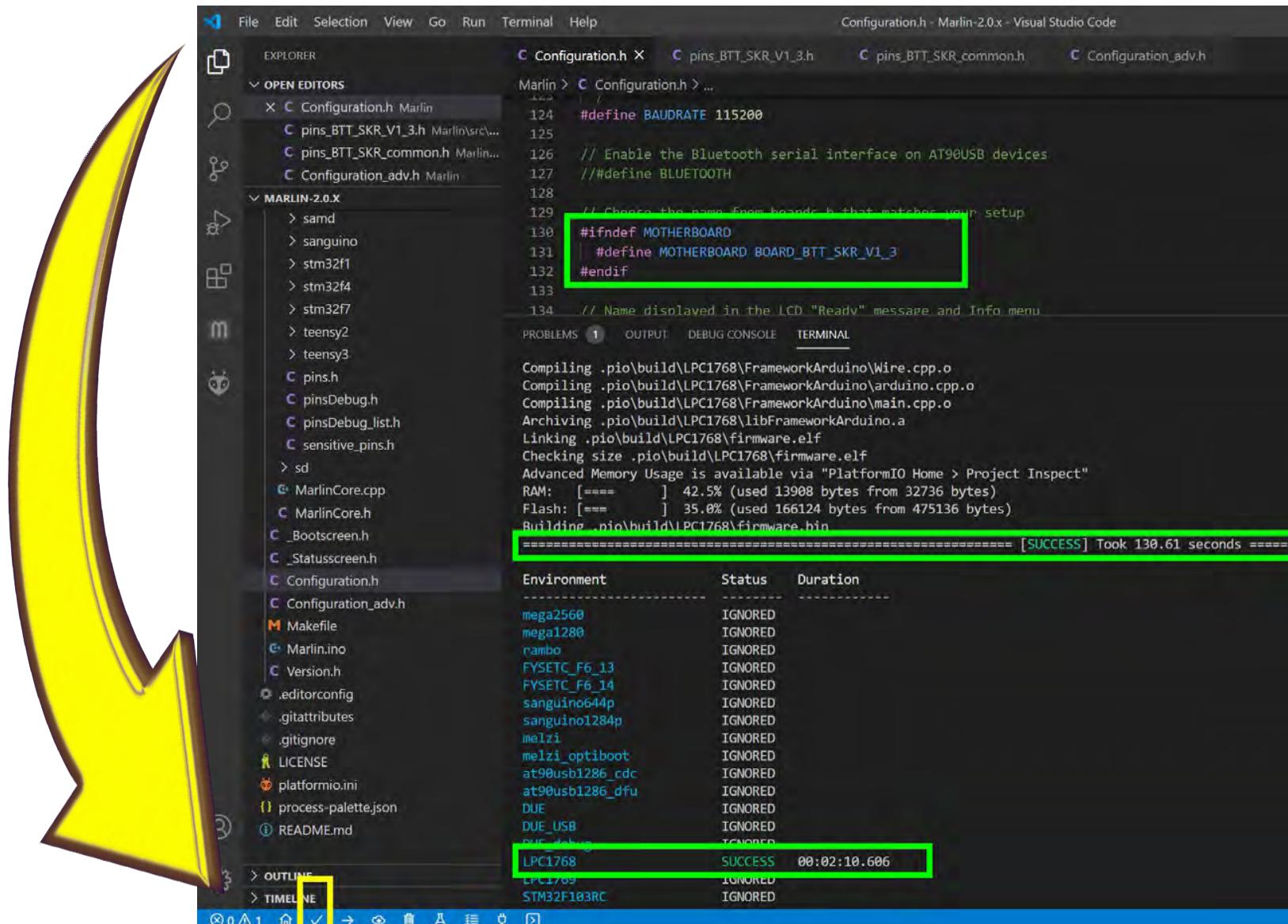
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  pins.h
  pinsDebug_list.h
  pinsDebug.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palettejson
  README.md
  OUTLINE
  1035 #define DISABLE_Y false
  1036 #define DISABLE_Z false
  1037
  // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  // @section extruder
  1041
  1042
  1043 #define DISABLE_E false          // For all extruders
  1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1045
  // @section machine
  1046
  // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1048 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1049 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1051
  // @section extruder
  1052
  // For direct drive extruder v9 set to true, for geared extruder set to false.
  1055 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1056 #define INVERT_E1_DIR false
  1057 #define INVERT_E2_DIR false
  1058 #define INVERT_E3_DIR false
  1059 #define INVERT_E4_DIR false
  1060 #define INVERT_E5_DIR false
  1061 #define INVERT_E6_DIR false
  1062 #define INVERT_E7_DIR false
  1063
  // @section homing
  1065 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1066
  1067 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  1068
  1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

C Configuration.h X C pins_BTT_SKR_V1_3.h C pins_BTT_SKR_common.h C Configuration_adv.h
Marlin > C Configuration.h > ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

```

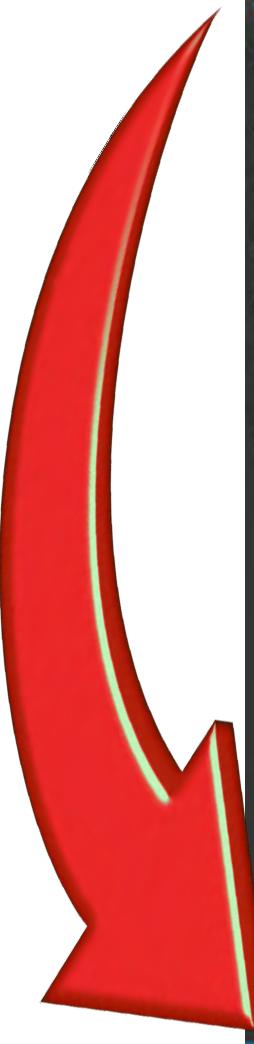
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
ramps	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

OUTLINE TIMELINE

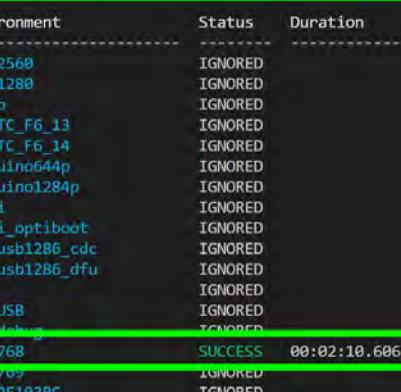
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

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

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







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

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

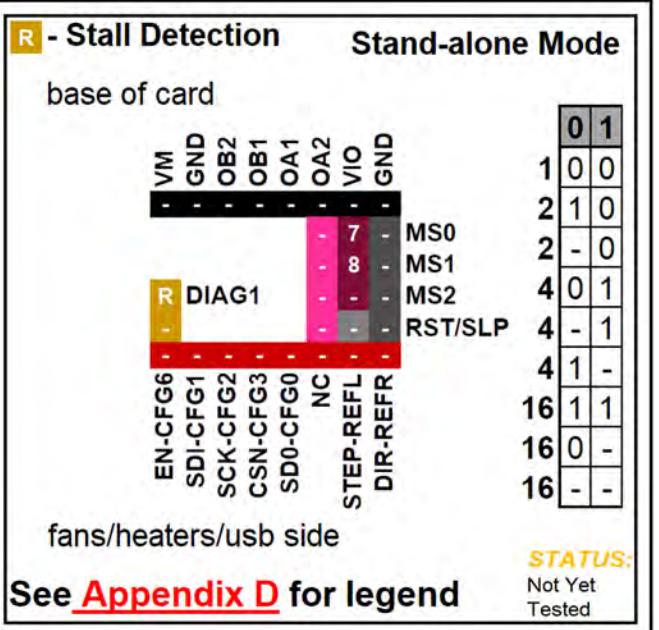
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1709	IGNORED	
STM32F103RC	IGNORED	

> OUTLINE > TIMELINE

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

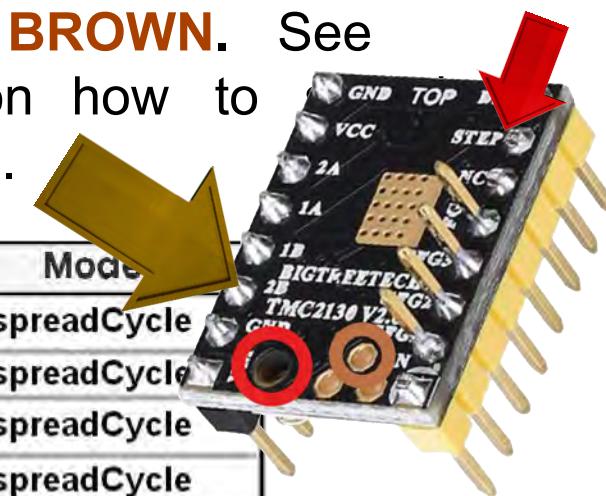


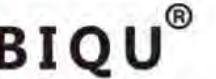
BIQU TMC2130

Stand-alone Mode

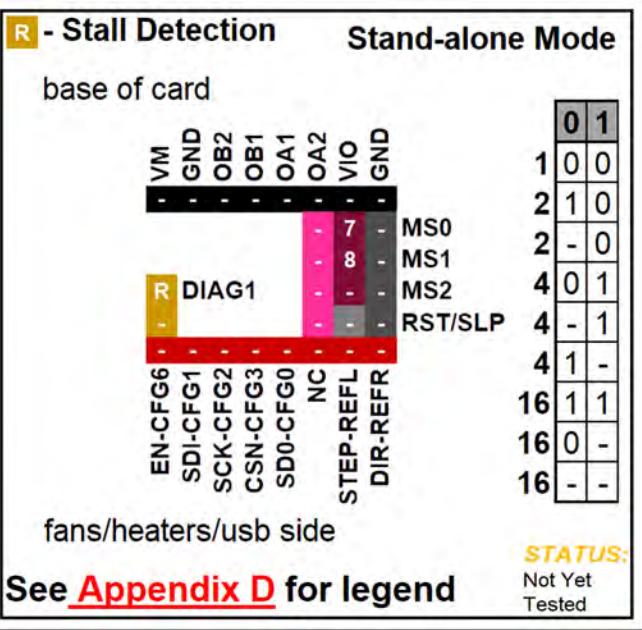
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

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



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

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

**Driver Chip Chart:**

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

Driving Current Calculation Formula
 R_s (typical Sense Resistor)= 0.11Ω

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

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

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

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

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

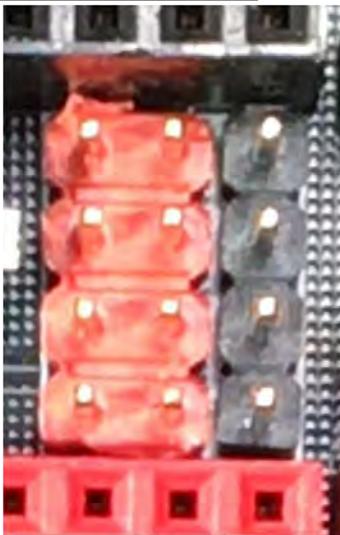
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers - PART 1**BIQU TMC2130**
Stand-alone Mode

OPEN → **No Jumper**

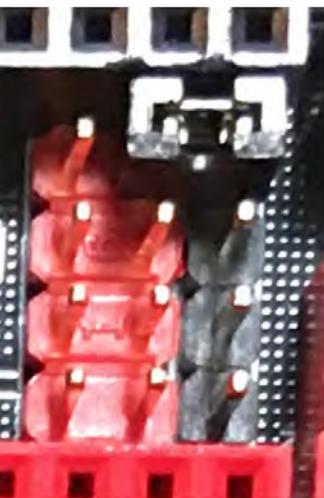
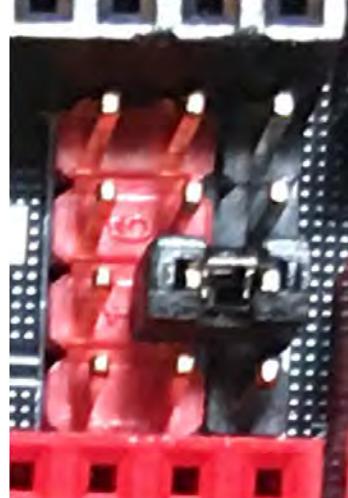


Rows:

MS0**MS1****MS2****RST/SLP**

High → **Jumper Set**

Rows:

MS0**MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

BIQU TMC2130

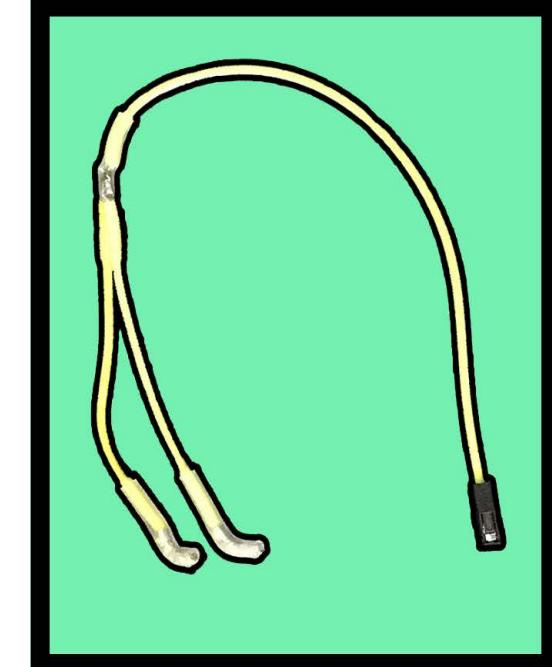
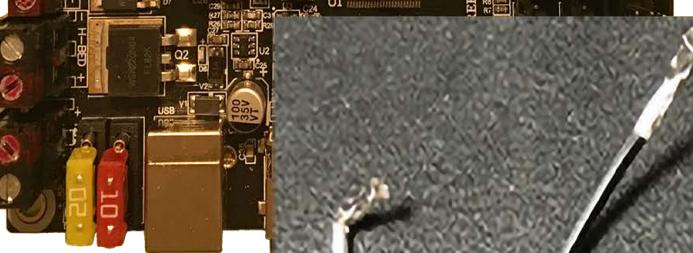
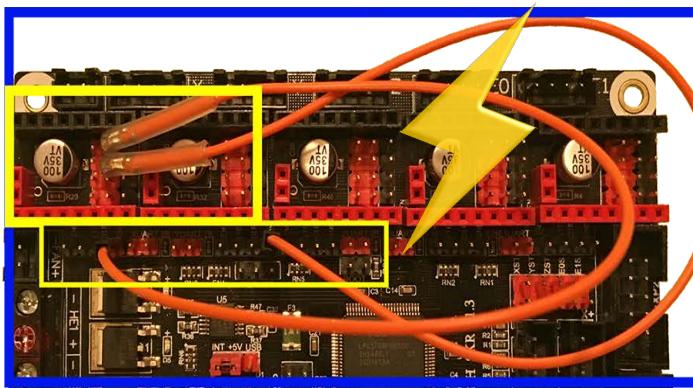
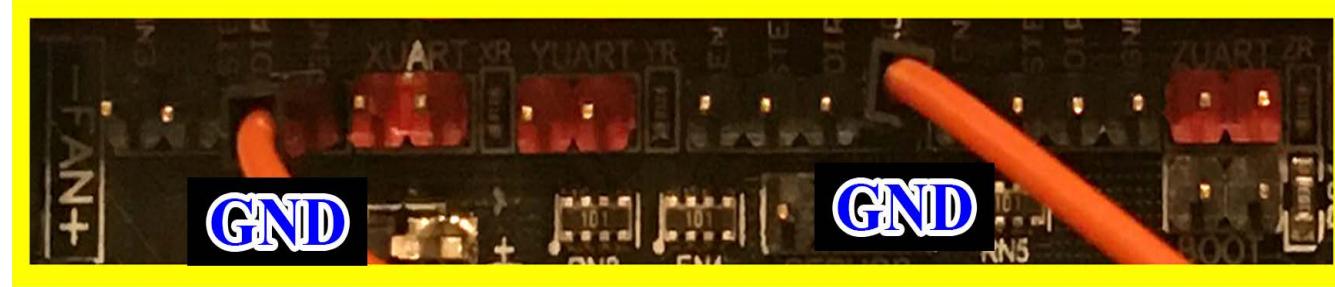
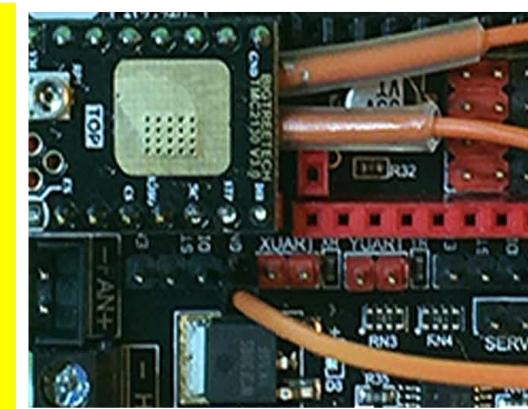
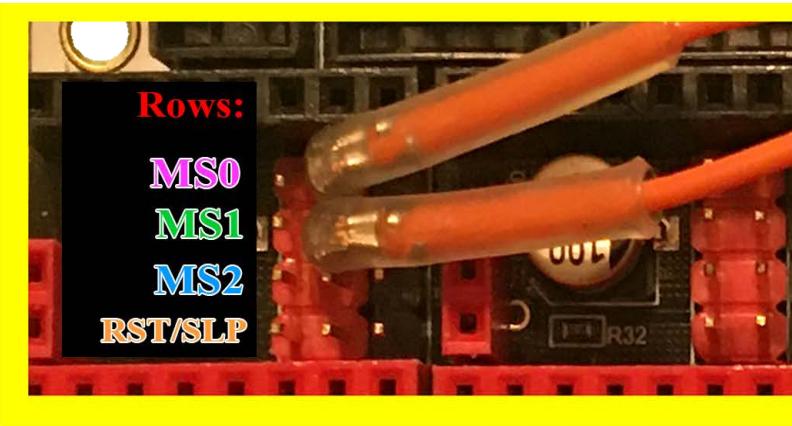
Stand-alone Mode

Driver Chip Chart:

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

Driving Current Calculation Formula	$I_{MAX} = V_{ref}$	$V_{ref} = I_{MAX}$
	See Appendix B #x. Use 50% to 90% as shown below: $I_{MAX} = I_{MAX} * 0.90$	See Appendix B #x. Use 50% to 90% as shown below: $V_{ref} = V_{ref} * 0.90$

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

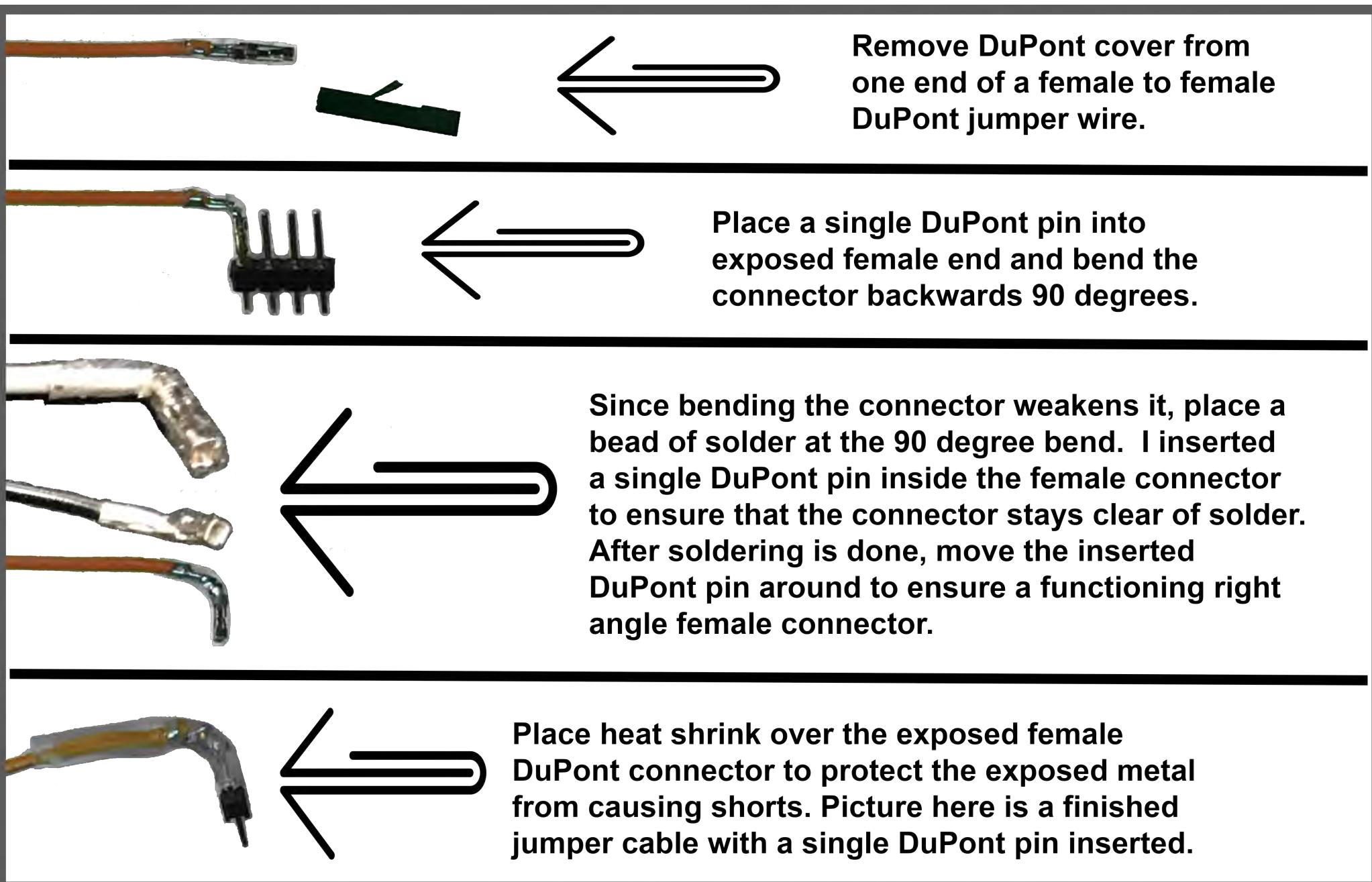


Low ➔ MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable

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

Stand-alone Mode

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

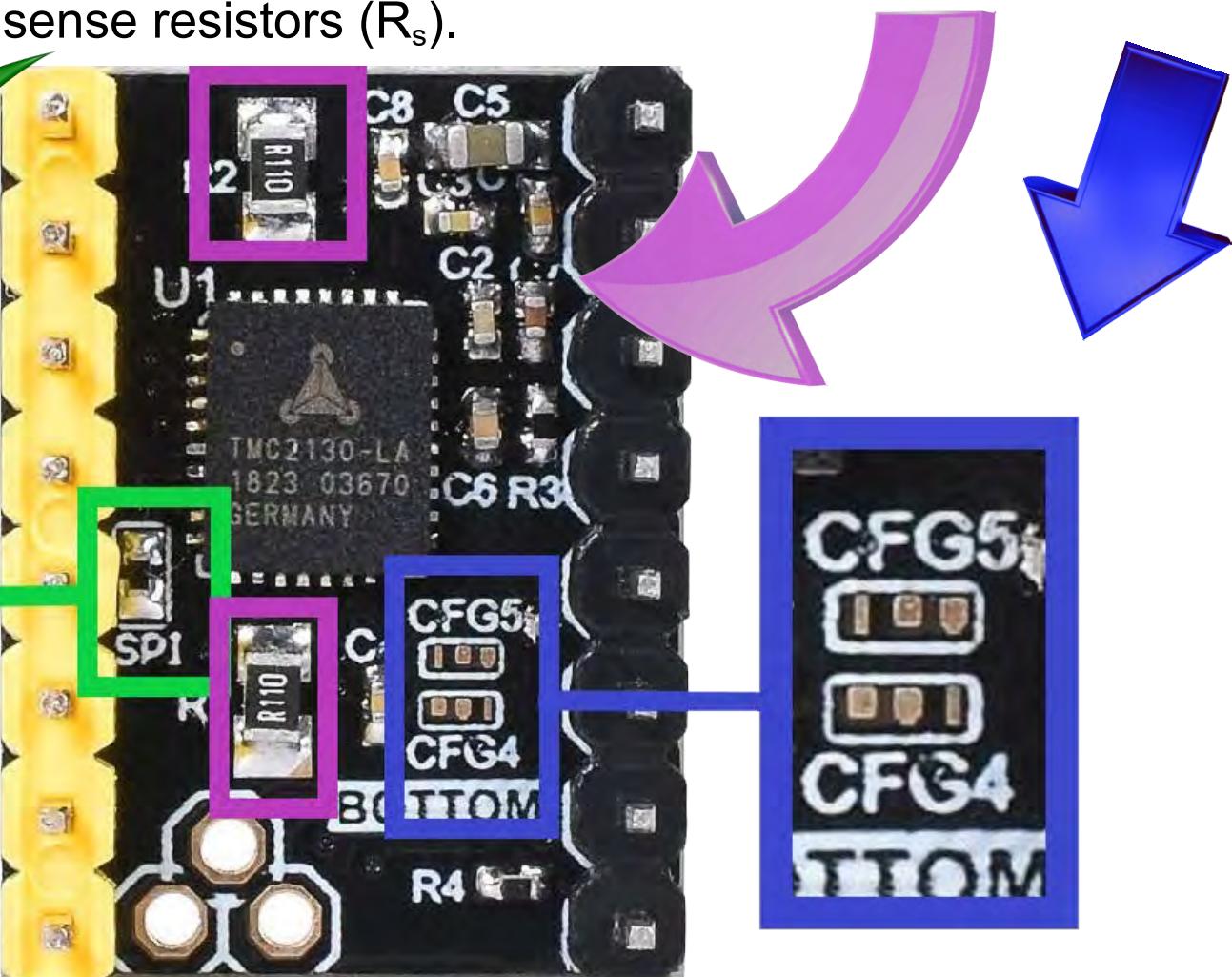
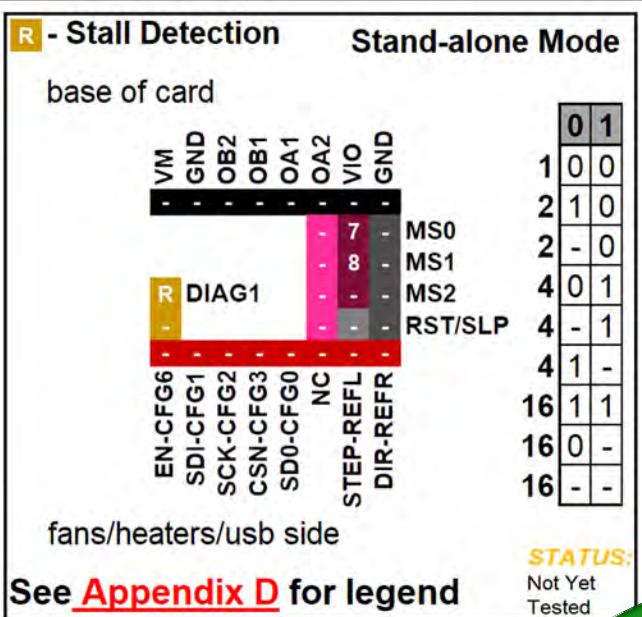


BIQU TMC2130

Stand-alone Mode

To place the BIQU TMC2130 into Stand-alone mode:

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

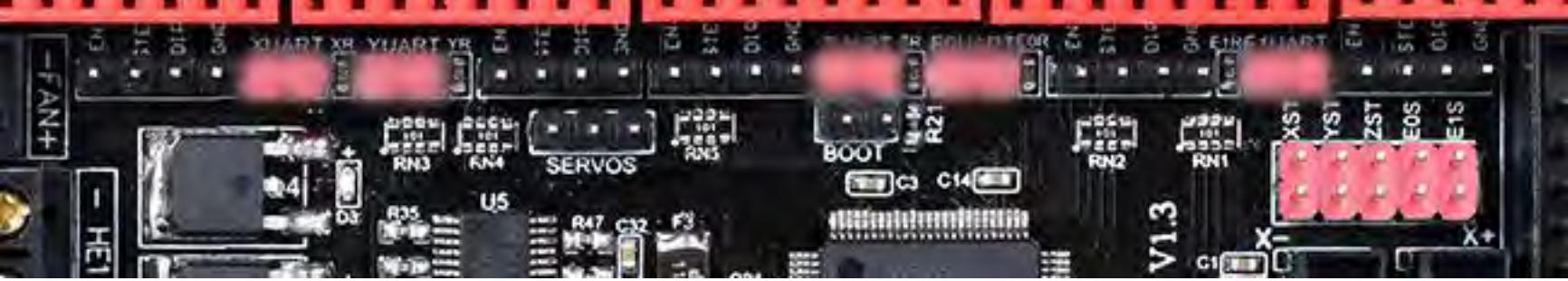


BIQU TMC2130Stand-alone Mode

Stand-alone Mode

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

X- (XST) X-DIAG1
Y- (YST) Y-DIAG1
Z- (ZST) Z-DIAG1
(E0ST) E0-DIAG1 X+
(E1ST) E1-DIAG1 Y+



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

XUART



YUART



ZUART



E0 UART



E1 UART



Stand-alone Mode

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

Driver Socket Representation

Columns: 1 2 3 Rows: MS0 MS1 MS2 RST/SLP

Driver Chip Chart:

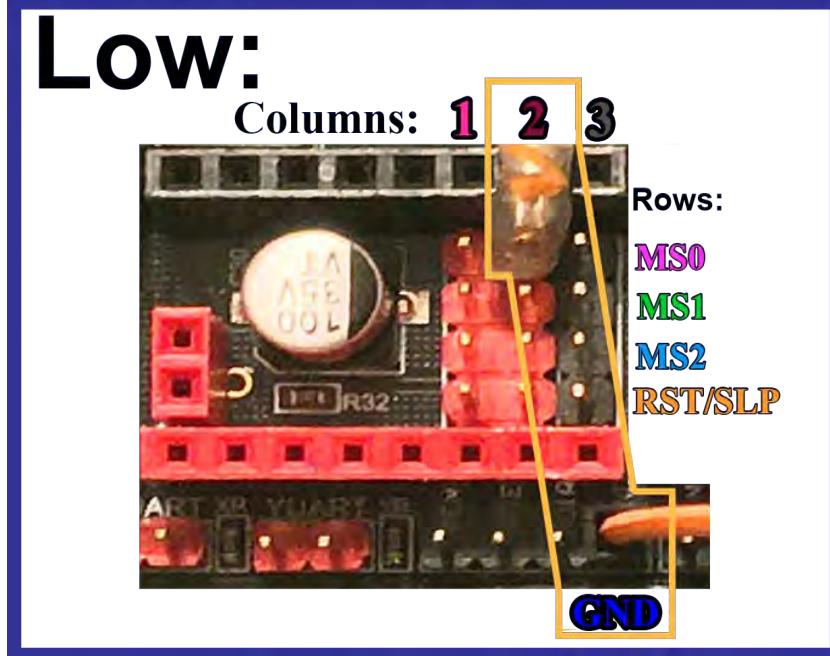
Driver Chip	MS0	MS1	Steps	Interpolation	Mode
LOGO TMC21XX Stand Alone Mode	Low	Low	1	NONE	spreadCycle
	High	Low	1/2	NONE	spreadCycle
	OPEN	Low	1/2	1/256	spreadCycle
	LOW	High	1/4	NONE	spreadCycle
	OPEN	High	1/4	1/256	spreadCycle
	High	OPEN	1/4	1/256	stealthChop
	High	High	1/16	NONE	spreadCycle
	LOW	OPEN	1/16	1/256	spreadCycle
OPEN	OPEN	1/16	1/256	stealthChop	

Driving Current Calculation Formula: $I_{MAX} = V_{ref}$ $V_{ref} = I_{MAX}$

See Appendix B $\text{R}_{\text{on}} = 0.11 \Omega$ See Appendix B $\text{R}_{\text{on}} = 0.05 \Omega$

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

No Jumper set



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

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

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

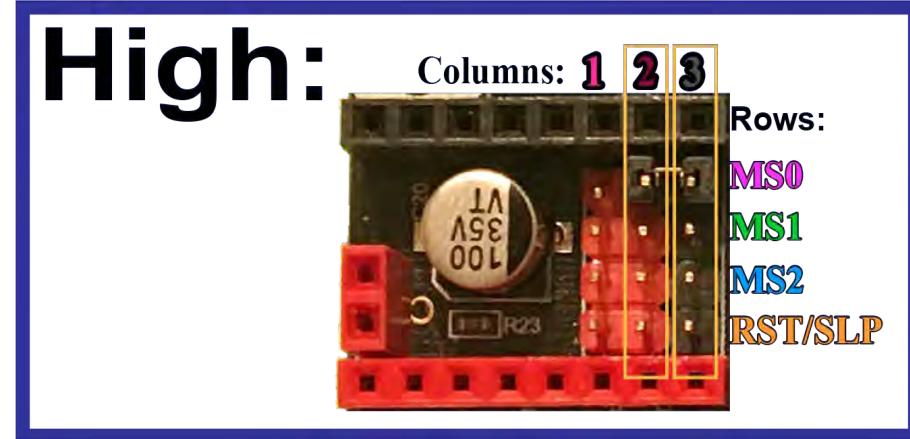
Driver Socket Representation: - - → **Driver Chip Chart:** Open

Meaning:

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

Low → Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)

Open → No Jumper set



BIQU TMC2130

Stand-alone Mode

Stand-alone Mode

Driver Socket Representation

Columns: 1 2 3

Rows: MS0 MS1 MS2 RST/SLP

E N 1/4 with 1/256 in SpreadCycle Mode

Columns: 1 2 3

Rows: MS0 MS1 MS2 RST/SLP

E N 1/2 with 1/256 in SpreadCycle Mode

Driver Chip Chart:

Driver Chip	MS0	MS1	Steps	Interpolation	Mode
TMC21XX Stand Alone Mode	Low	Low	1	NONE	spreadCycle
	High	Low	1/2	NONE	spreadCycle
	OPEN	Low	1/2	1/256	spreadCycle
	Low	High	1/4	NONE	spreadCycle
	OPEN	High	1/4	1/256	spreadCycle
	High	OPEN	1/4	1/256	stealthChop
	High	High	1/16	NONE	spreadCycle
	Low	OPEN	1/16	1/256	spreadCycle

Driving Current Calculation Formula

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

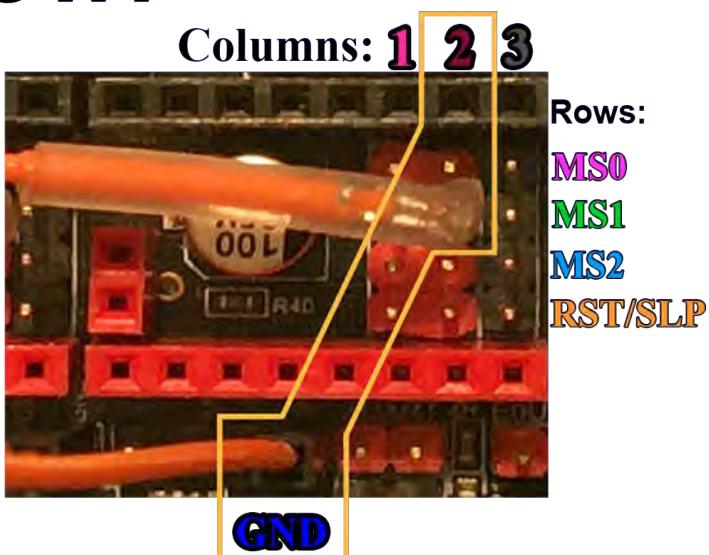
See Appendix B Use 50% to 90% as shown below:

$$I_{max} = I_{MAX} * 0.90 \quad V_{max} = V_{ref} * 0.90$$

Appendix B: Subdivision 46V DC 2.5A (peak) $R_{SPLITTER} = 0.11\Omega$

No Jumper set

LOW:



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



Driver Chip Chart:

High →

Low →

Open →

Meaning:

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

Use a DuPont Jumper Cable to connect column 2 PIN from the MS1 row to ground (GND)

No Jumper set

High:



Stand-alone Mode

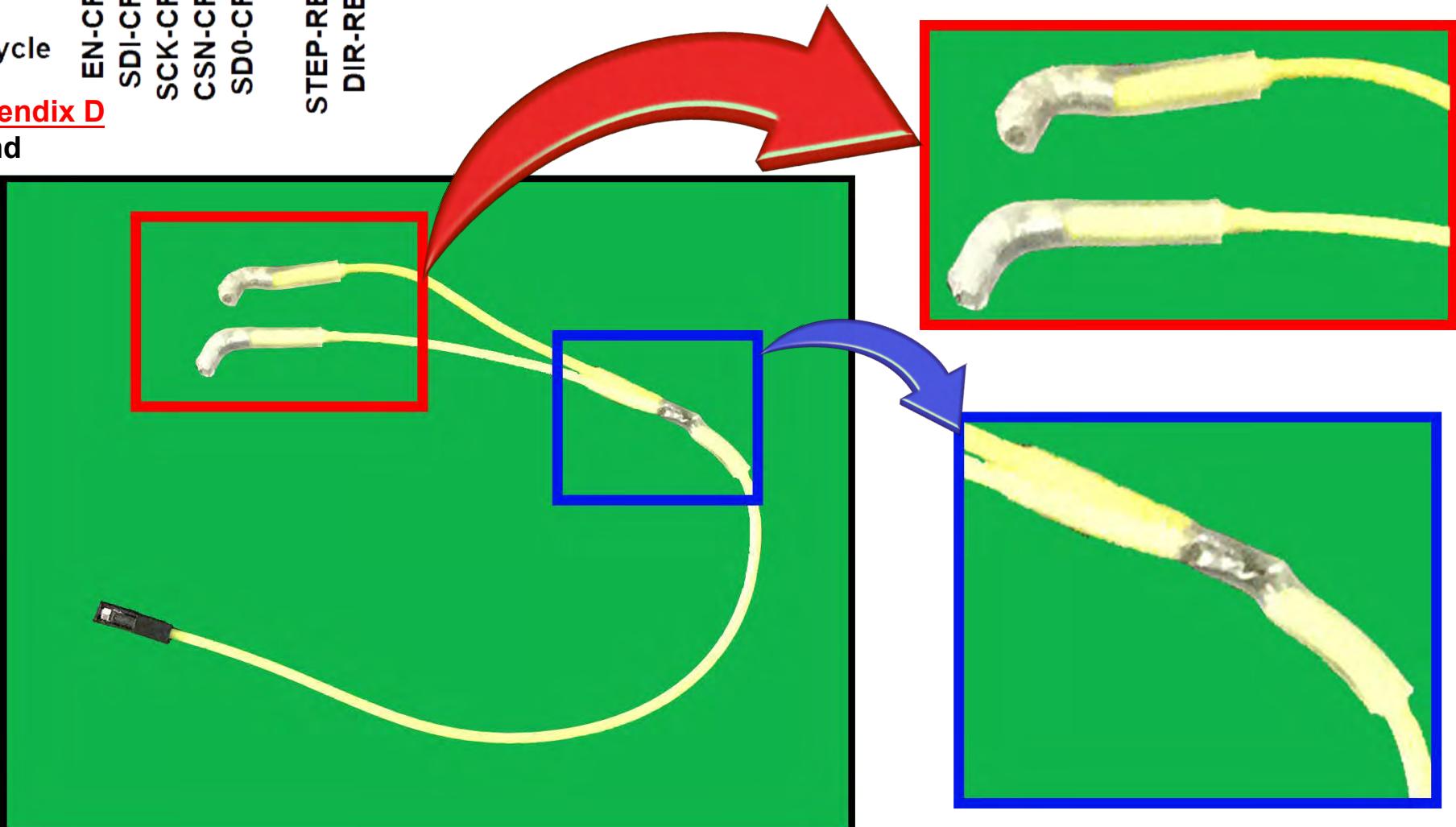
Stand-alone Mode	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
	-	-	-	-	-	-	-	-
STEP	R	DIAG1			7*		MS0	
	-	-			8*	-	MS1	
					MS2			
Interpolation:							RST/SLP	
none								
SpreadCycle	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
See Appendix D for legend								

BIQU TMC2130

Stand-alone Mode

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

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

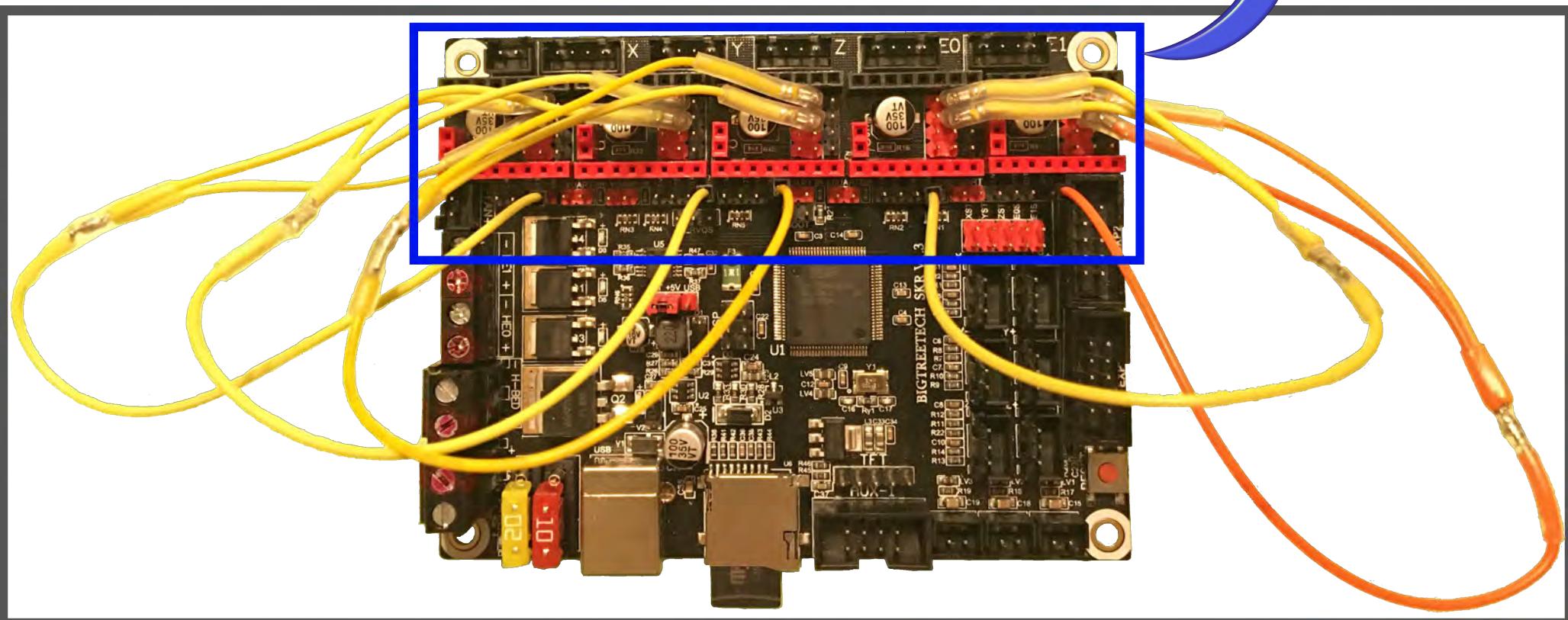
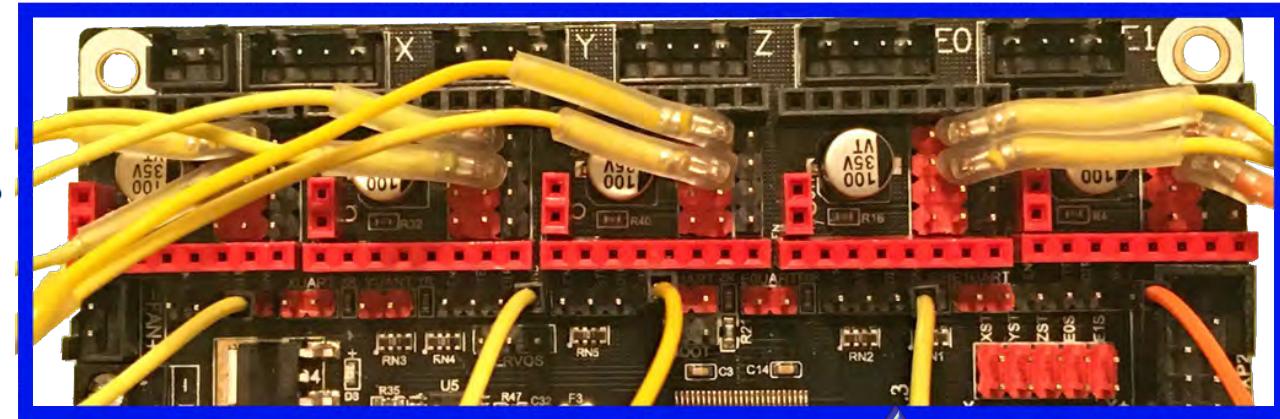


BIQU TMC2130

Stand-alone Mode

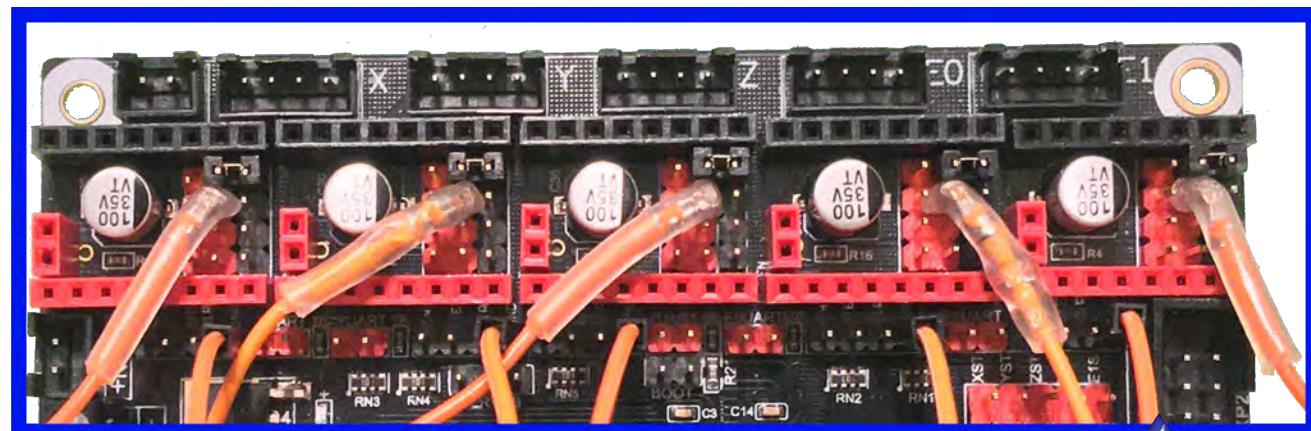
Stand-alone	-	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
Mode	-	-	-	-	-	-	-	-	-
STEP	R	DIAG1	-	-	-	7*	-	MS0	-
Interpolation:	-	-	-	-	-	8*	-	MS1	-
none	-	-	-	-	-	-	-	MS2	-
SpreadCycle	-	-	NC	-	SD0-CFG0	SD0-CFG1	SD0-CFG2	SD0-CFG3	SD0-CFG4
	-	-	-	-	STEP-REFL	DIR-REFR	STEP-REFL	DIR-REFR	DIR-REFR

See [Appendix D](#) for legend

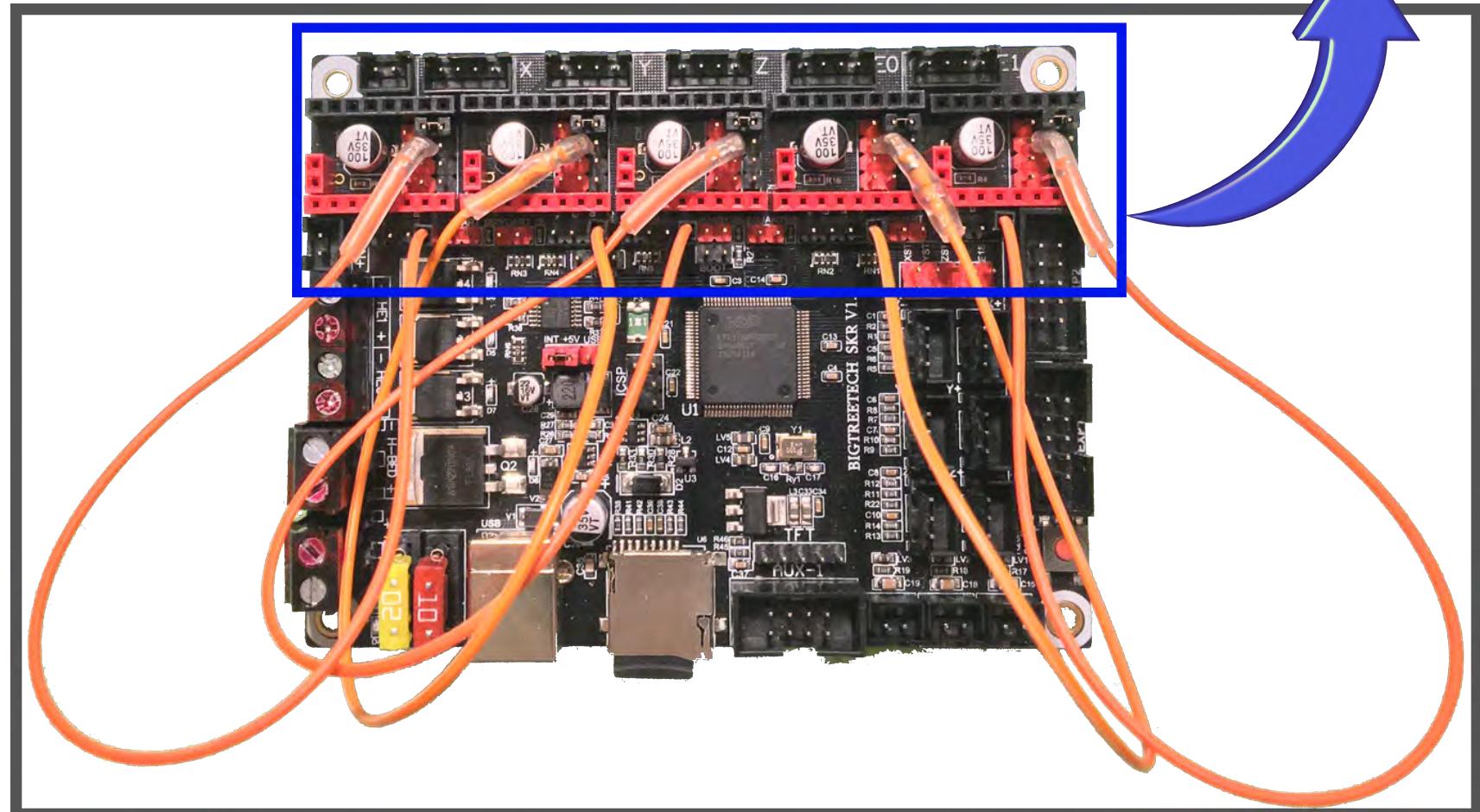


BIQU TMC2130Stand-alone Mode**Stand-alone Mode**

Stand-alone Mode	VM	GND	OB2	OB1	OA1	OA2	V/I/O	GND
	-	-	-	-	7	7	MS0	
	R	DIAG1			8*	-	MS1	
							MS2	
							RST/SLP	
1 / 2								
Interpolation:	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
none								
SpreadCycle								



See [Appendix D](#)
for legend



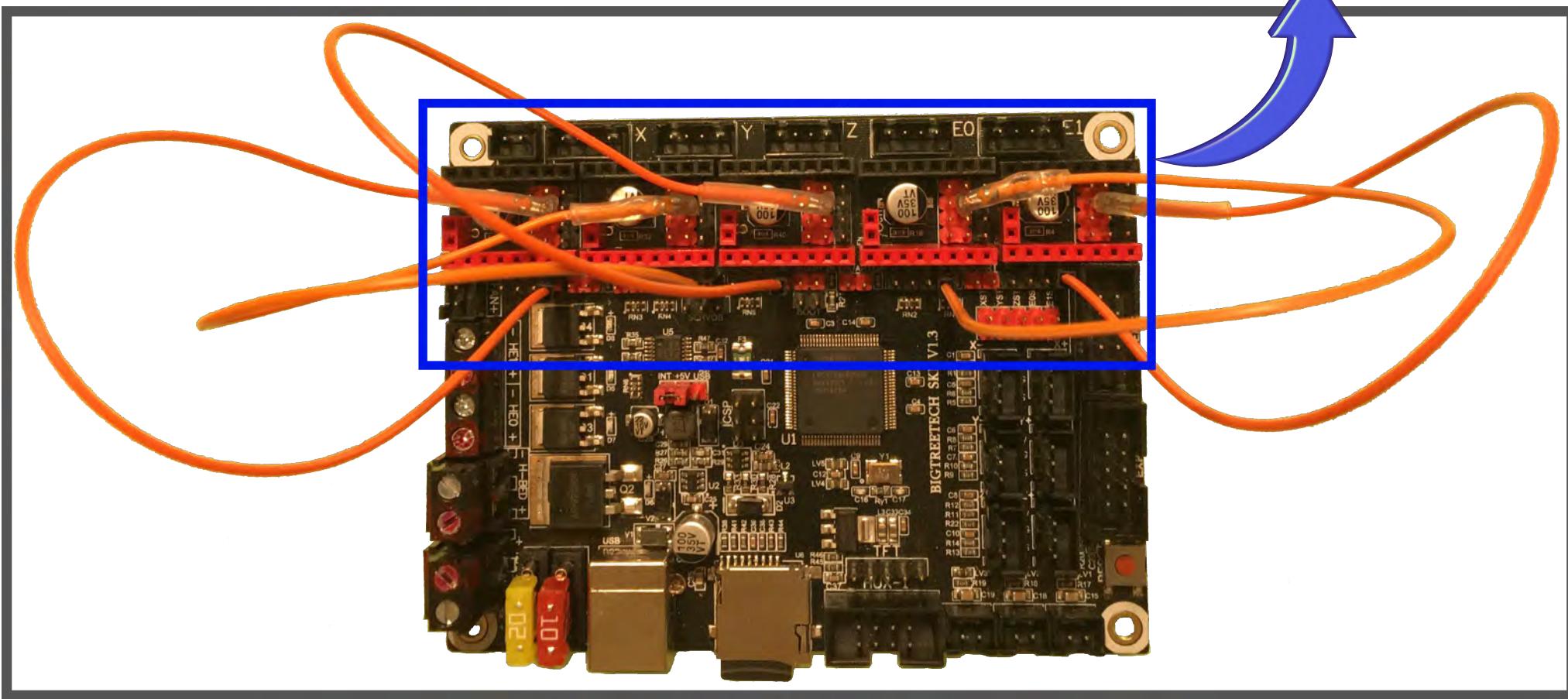
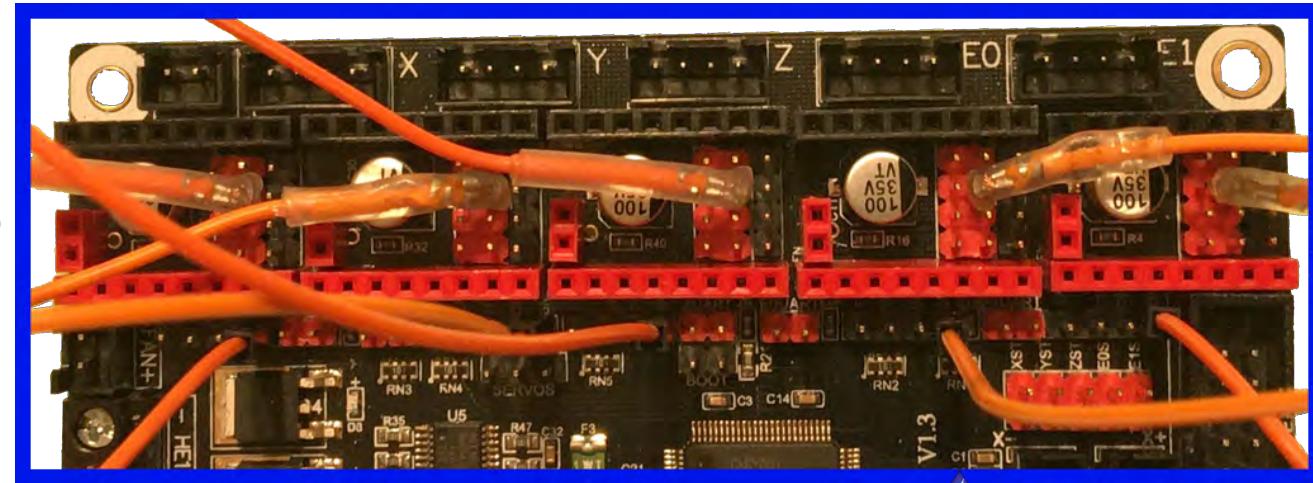
BIQU TMC2130

Stand-alone Mode

Stand-alone Mode

Stand-alone Mode	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
1 / 2	R	DIAG1			MS0			
Interpolation:	EN-CFG6		8*		MS1			
1 / 256	SDI-CFG1				MS2			
SpreadCycle	SCK-CFG2							
	CSN-CFG3							
	SD0-CFG0	NC						
	STEP-REFL							
	DIR-REFR							

See [Appendix D](#)
for legend

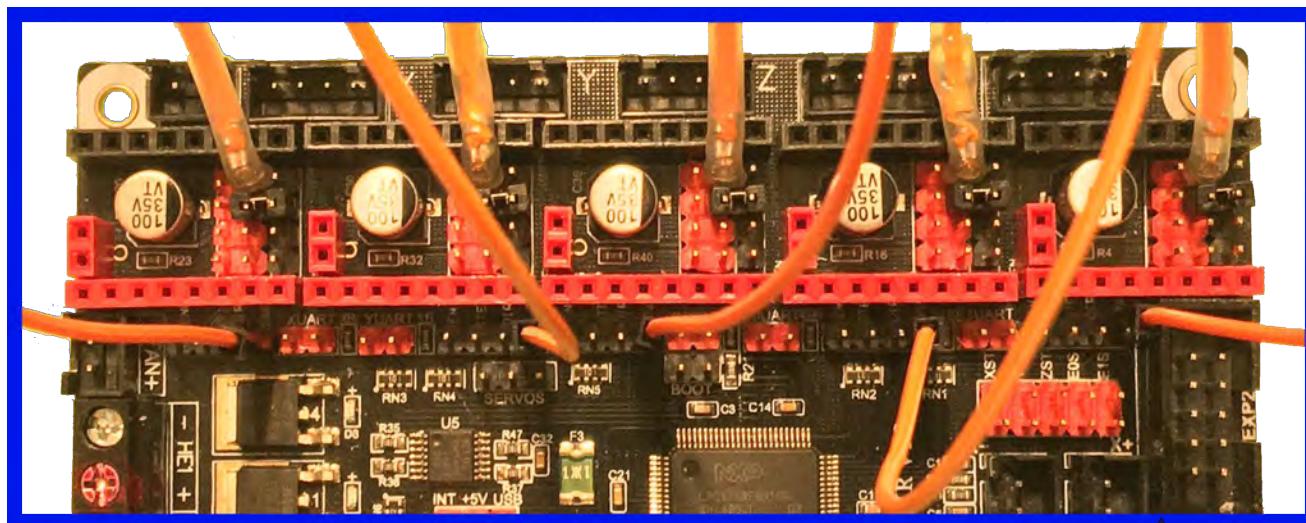


BIQU TMC2130

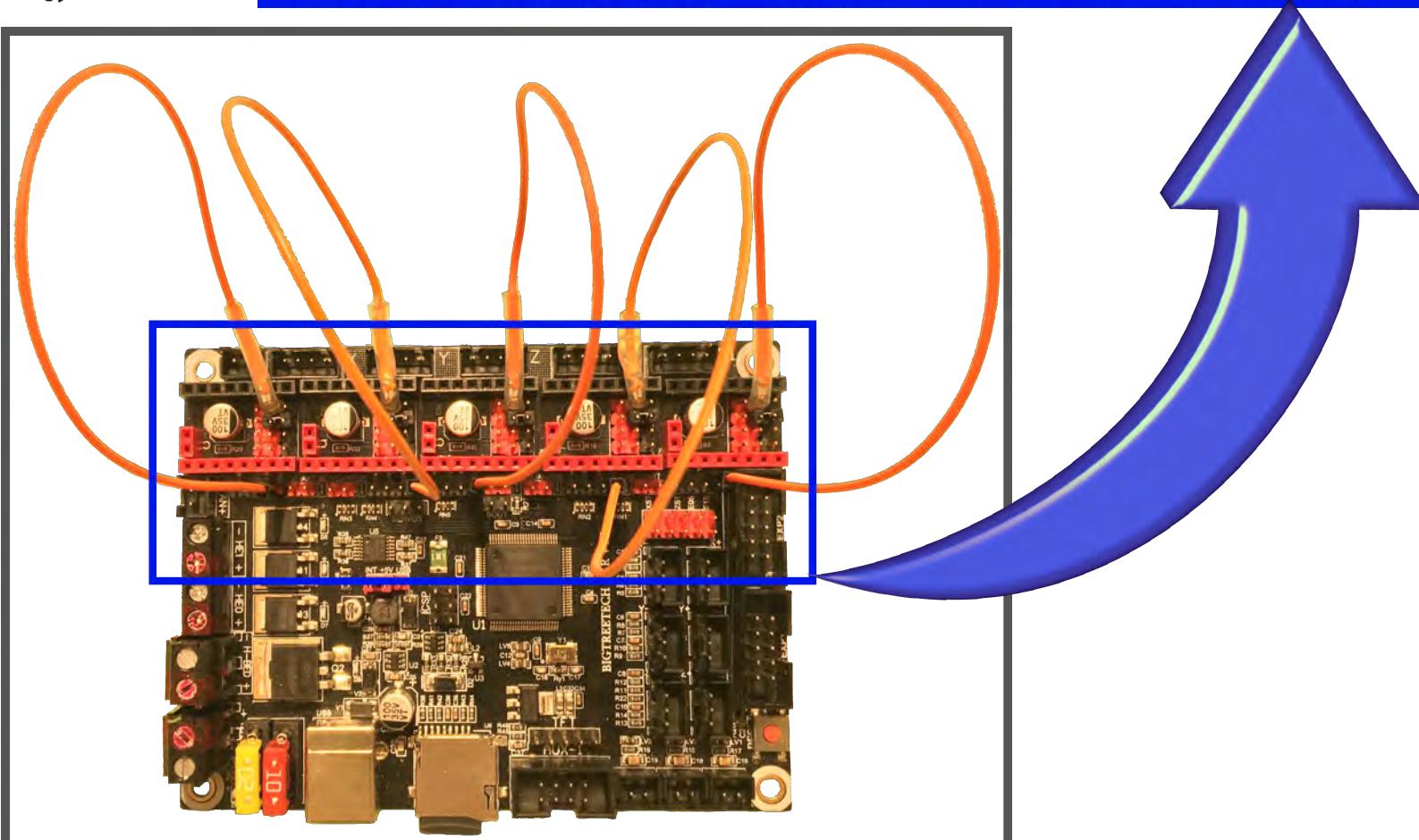
Stand-alone Mode

Stand-alone Mode

Stand-alone Mode	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
1 / 4	R	DIAG1	-	-	7*	-	MS0	-
			-	-	8	8	MS1	-
			-	-			MS2	-
			-	-			RST/SLP	-
Interpolation:	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
none								
SpreadCycle								



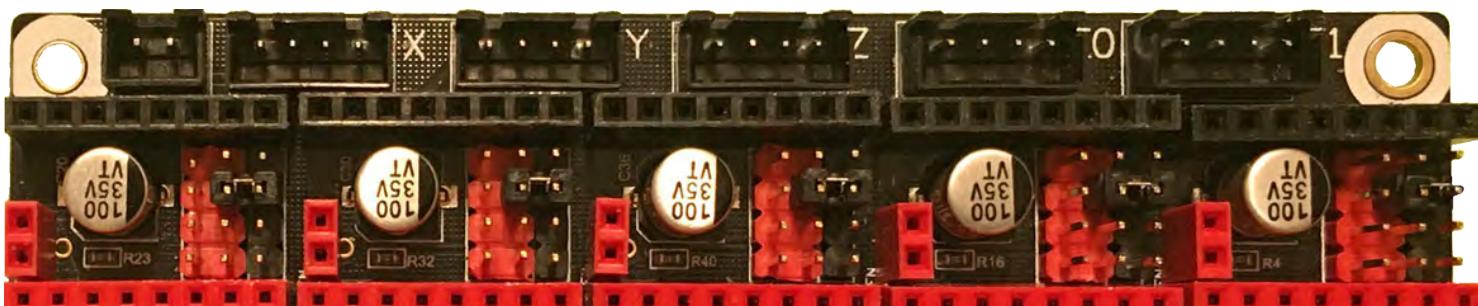
See [Appendix D](#)
for legend



BIQU TMC2130Stand-alone Mode**Stand-alone Mode****1 / 4**Interpolation:
1 / 256

SpreadCycle

	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
Stand-alone Mode	-	-	-	-	-	-	-	
1 / 4	R	DIAG1	-	-	-	-	-	
Interpolation:	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
SpreadCycle	-	-	-	-	-	-	-	-

See [Appendix D](#) for legend

Stand-alone Mode

1 / 4Interpolation:
1 / 256

StealthChop

	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
Stand-alone Mode	-	-	-	-	-	-	-	
1 / 4	R	DIAG1	-	-	-	-	-	
Interpolation:	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
SpreadCycle	-	-	-	-	-	-	-	-

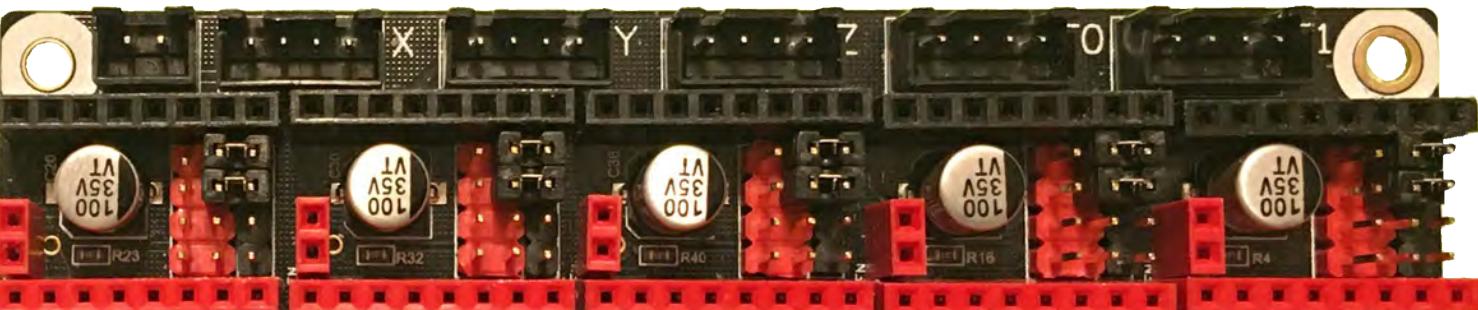
See [Appendix D](#) for legend

Stand-alone Mode

1 / 16Interpolation:
none

SpreadCycle

	VM	GND	OB2	OB1	OA1	OA2	VIO	GND
Stand-alone Mode	-	-	-	-	-	-	-	
1 / 16	R	DIAG1	-	-	-	-	-	
Interpolation:	EN-CFG6	SDI-CFG1	SCK-CFG2	CSN-CFG3	SD0-CFG0	NC	STEP-REFL	DIR-REFR
SpreadCycle	-	-	-	-	-	-	-	-

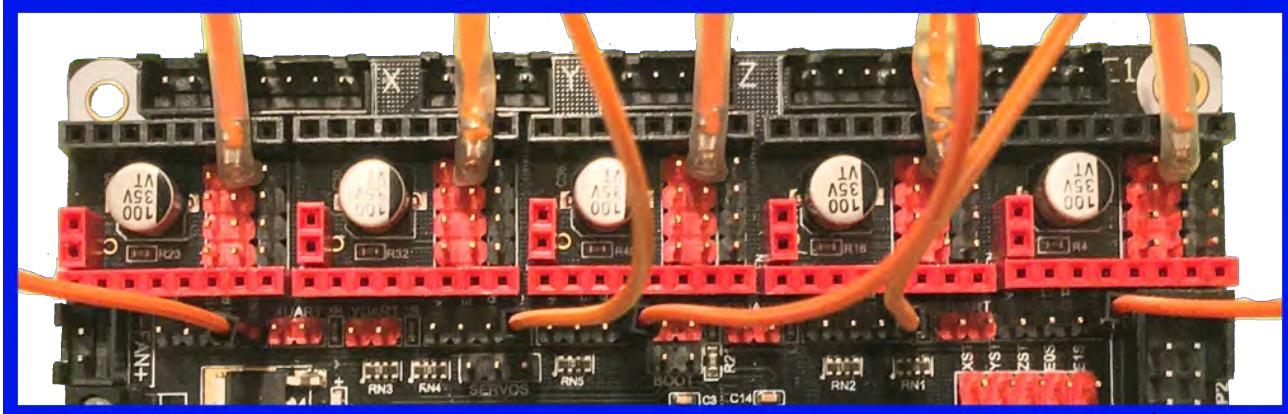
See [Appendix D](#) for legend

Stand-alone Mode

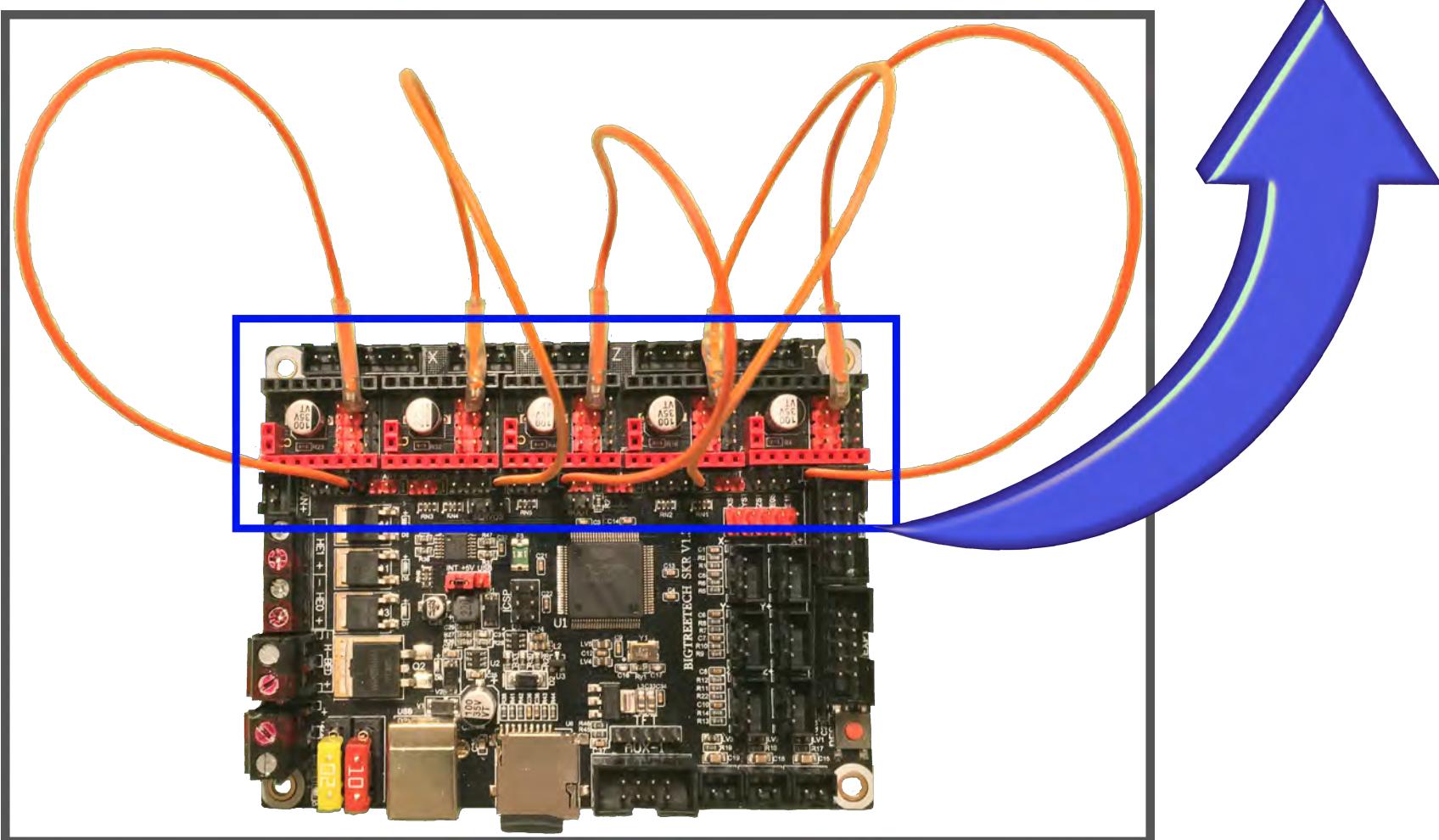
Stand-alone Mode	VM	GND				
	-	-	OB2	OB1	OA1	OA2
			-	-	VIO	GND
1 / 16	R	DIAG1	MS0	7*	-	-
Interpolation: 1 / 256	EN-CFG6	SDI-CFG1	MS1	-	-	-
SpreadCycle	SCK-CFG2	CSN-CFG3	MS2	-	-	-
	SD0-CFG0	NC	RST/SLP	-	-	-
			STEP-REFL	-	-	-
			DIR-REFR	-	-	-

BIQU TMC2130

Stand-alone Mode



See [Appendix D](#)
for legend



BIQU TMC2130

Stand-alone Mode

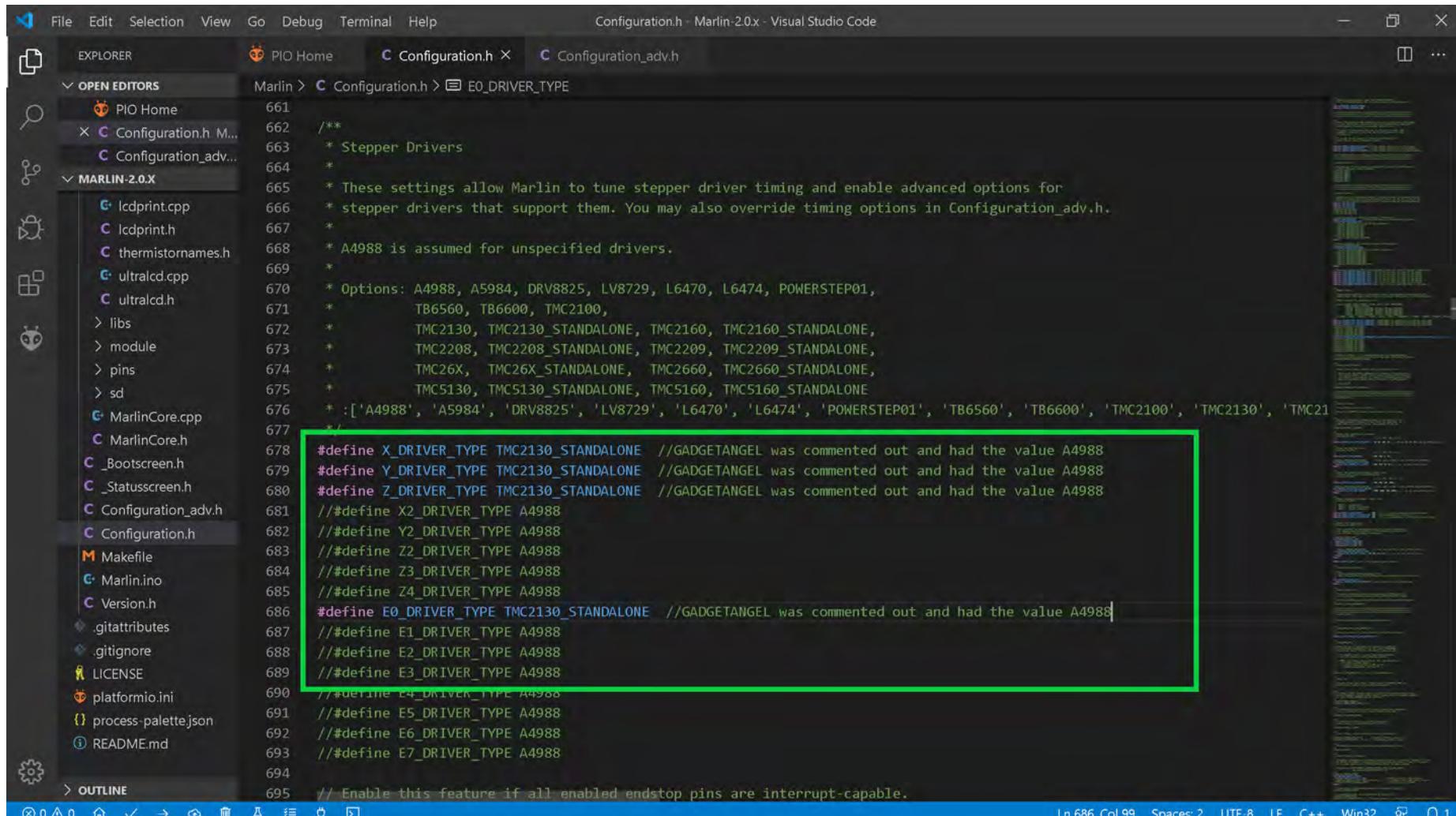
Stand-alone Mode



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

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2130 stepper motor drivers in stand-alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2130 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in stand-alone mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

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

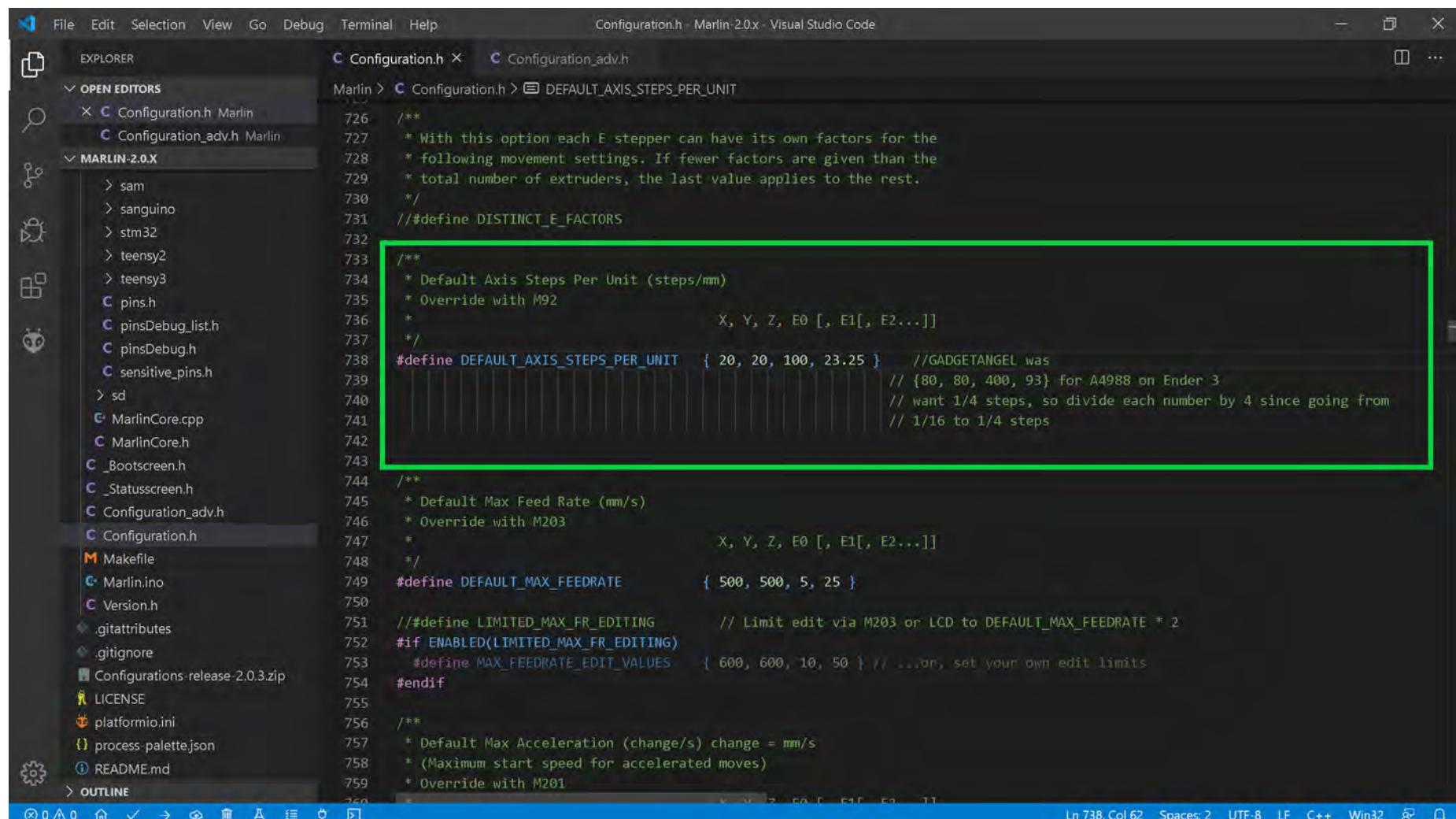
```

A green rectangular box highlights the driver configuration section starting from line 678, specifically the lines defining `X_DRIVER_TYPE`, `Y_DRIVER_TYPE`, `Z_DRIVER_TYPE`, and `E0_DRIVER_TYPE`. These lines were previously commented out with double slashes (//).

- Go to the next page.

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

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to BIQU TMC2130 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.



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

```

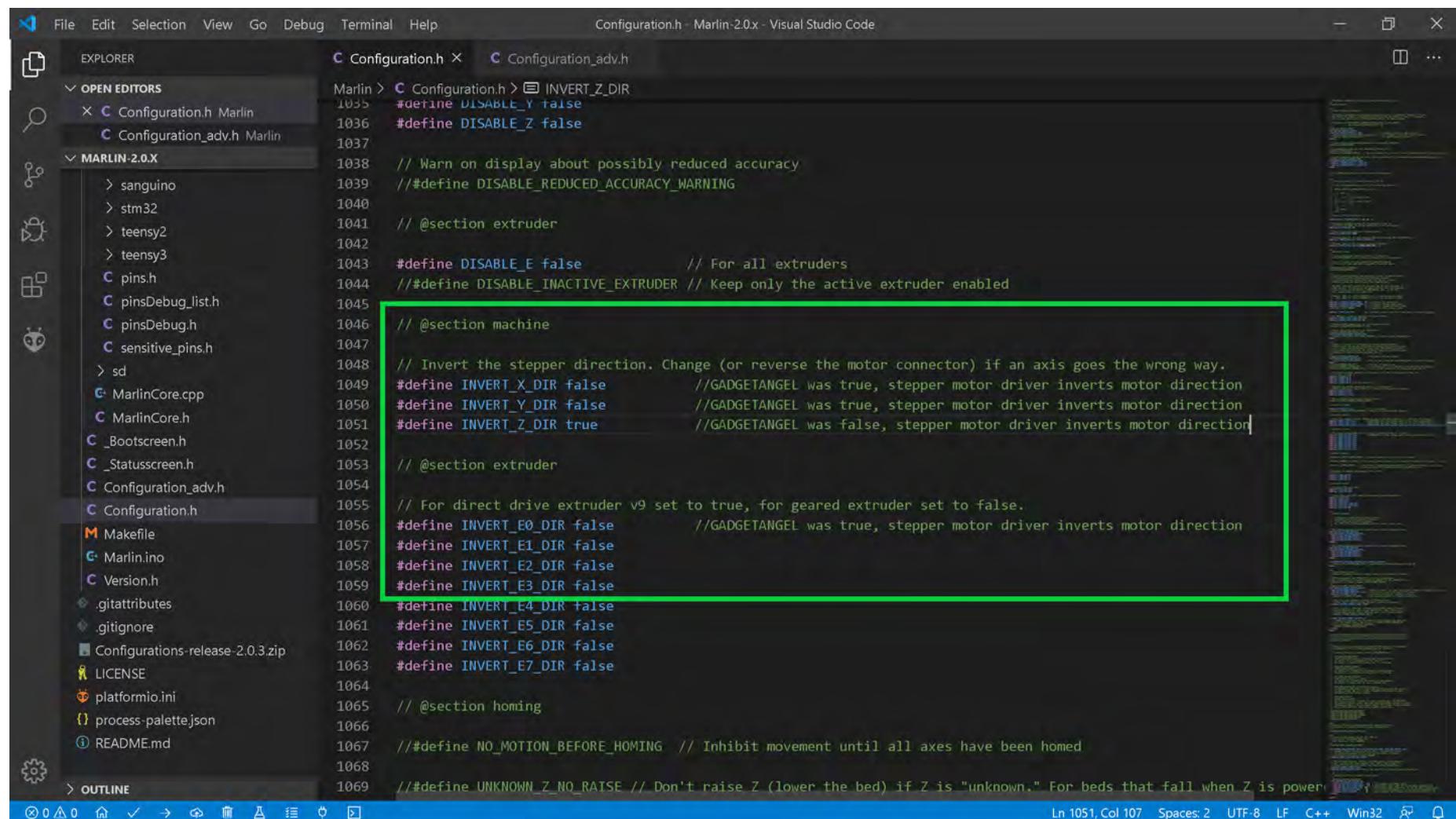
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732
733 /**
734 * Default Axis Steps Per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

```

- Go to the next page.

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

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as show in the **GREEN** box below



```

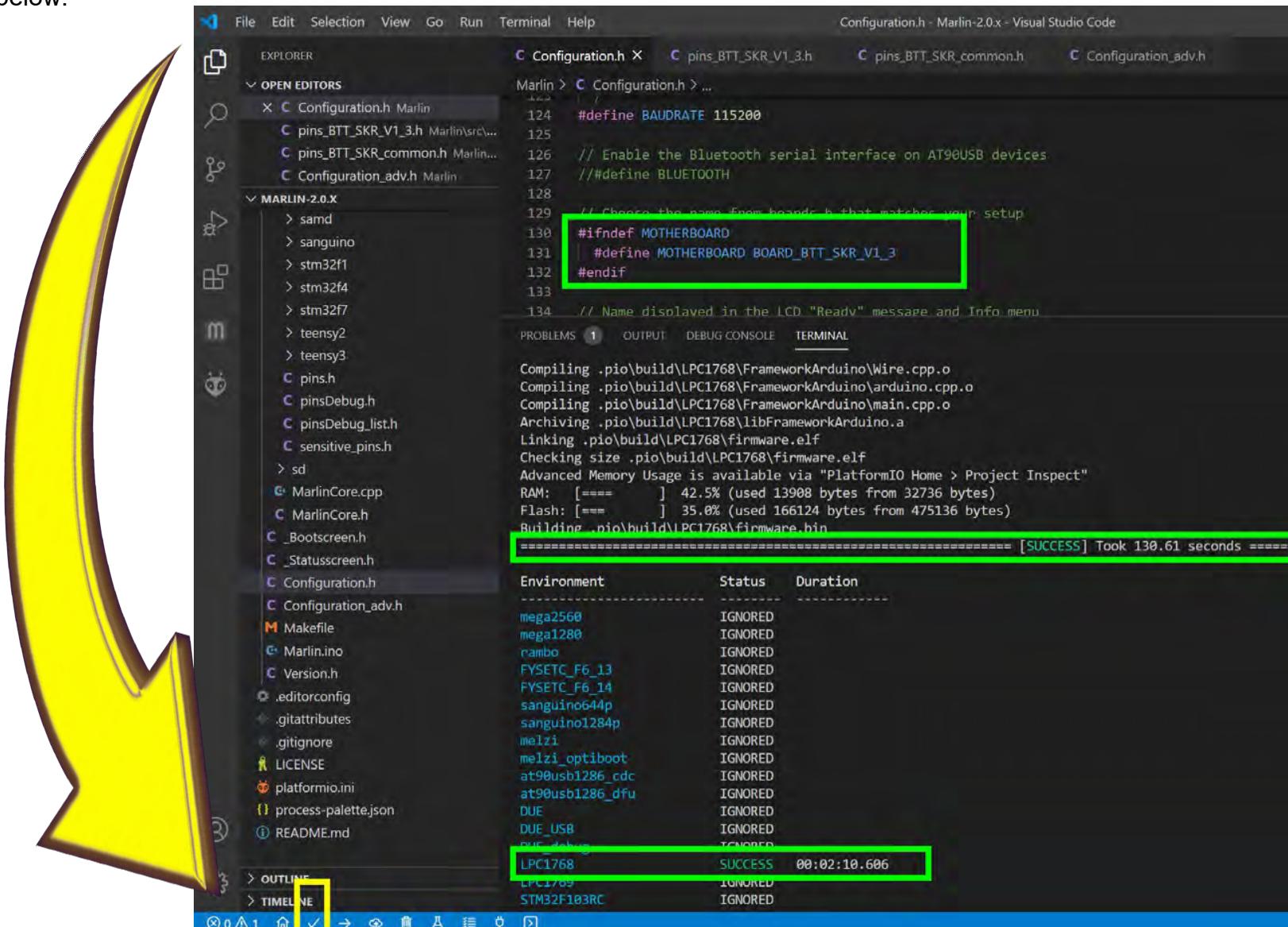
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
  Configuration.h Marlin
  Configuration_adv.h Marlin
MARLIN-2.0.X
  sanguino
  stm32
  teensy2
  teensy3
  pins.h
  pinsDebug_list.h
  pinsDebug.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  Configurations-release-2.0.3.zip
  LICENSE
  platformio.ini
  process-palettejson
  README.md
  OUTLINE
  1035 #define DISABLE_Y false
  1036 #define DISABLE_Z false
  1037
  // Warn on display about possibly reduced accuracy
  1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
  1040
  // @section extruder
  1041
  1042
  1043 #define DISABLE_E false          // For all extruders
  1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
  1045
  1046 // @section machine
  1047
  // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
  1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
  1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
  1052
  // @section extruder
  1053
  // For direct drive extruder v9 set to true, for geared extruder set to false.
  1055 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
  1056 #define INVERT_E1_DIR false
  1057 #define INVERT_E2_DIR false
  1058 #define INVERT_E3_DIR false
  1059 #define INVERT_E4_DIR false
  1060 #define INVERT_E5_DIR false
  1061 #define INVERT_E6_DIR false
  1062 #define INVERT_E7_DIR false
  1063
  // @section homing
  1065 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
  1066
  1067 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
  1068
  1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

```

124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
126 // #define BLUETOOTH
127
128 // Choose the name from boards.h that matches your setup
129 #ifndef MOTHERBOARD
130 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
131 #endif
132
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

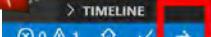
```

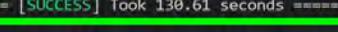
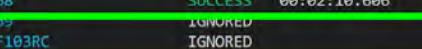
Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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

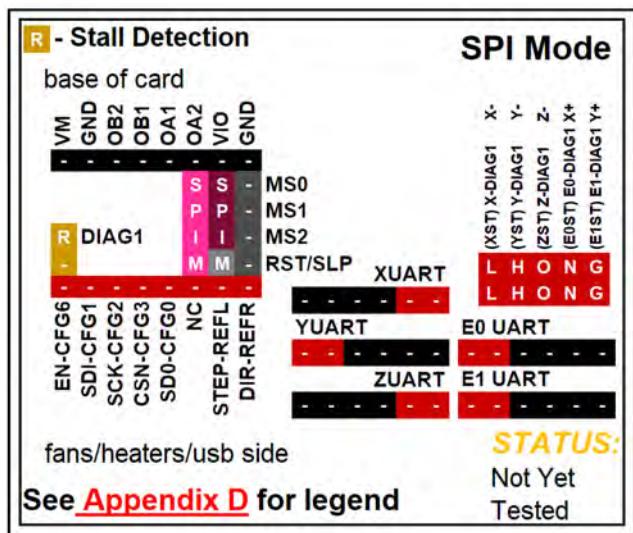




```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin
MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
TIMELINE
  Configuration.h ...
  124 #define BAUDRATE 115200
  125
  126 // Enable the Bluetooth serial interface on AT90USB devices
  127 // #define BLUETOOTH
  128
  129 // Choose the name from boards.h that matches your setup
  130 #ifndef MOTHERBOARD
  131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
  132 #endif
  133
  134 // Name displayed in the LCD "Ready" message and Info menu
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds
Environment Status Duration
-----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
SUE_L103 IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

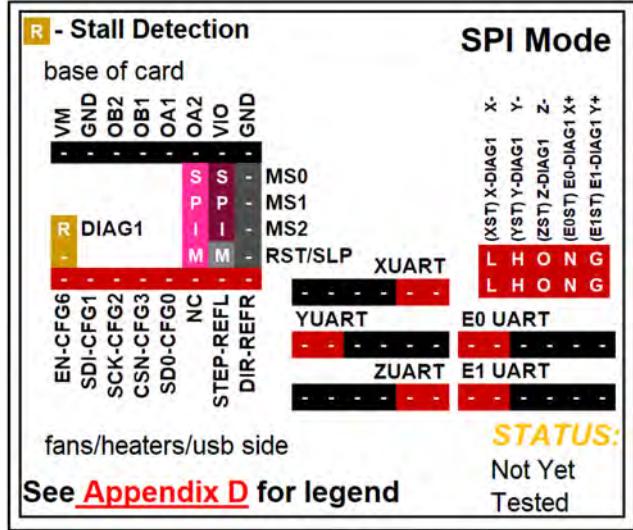
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3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

**BIQU TMC2130**SPI Mode

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

See the next page for further information.

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

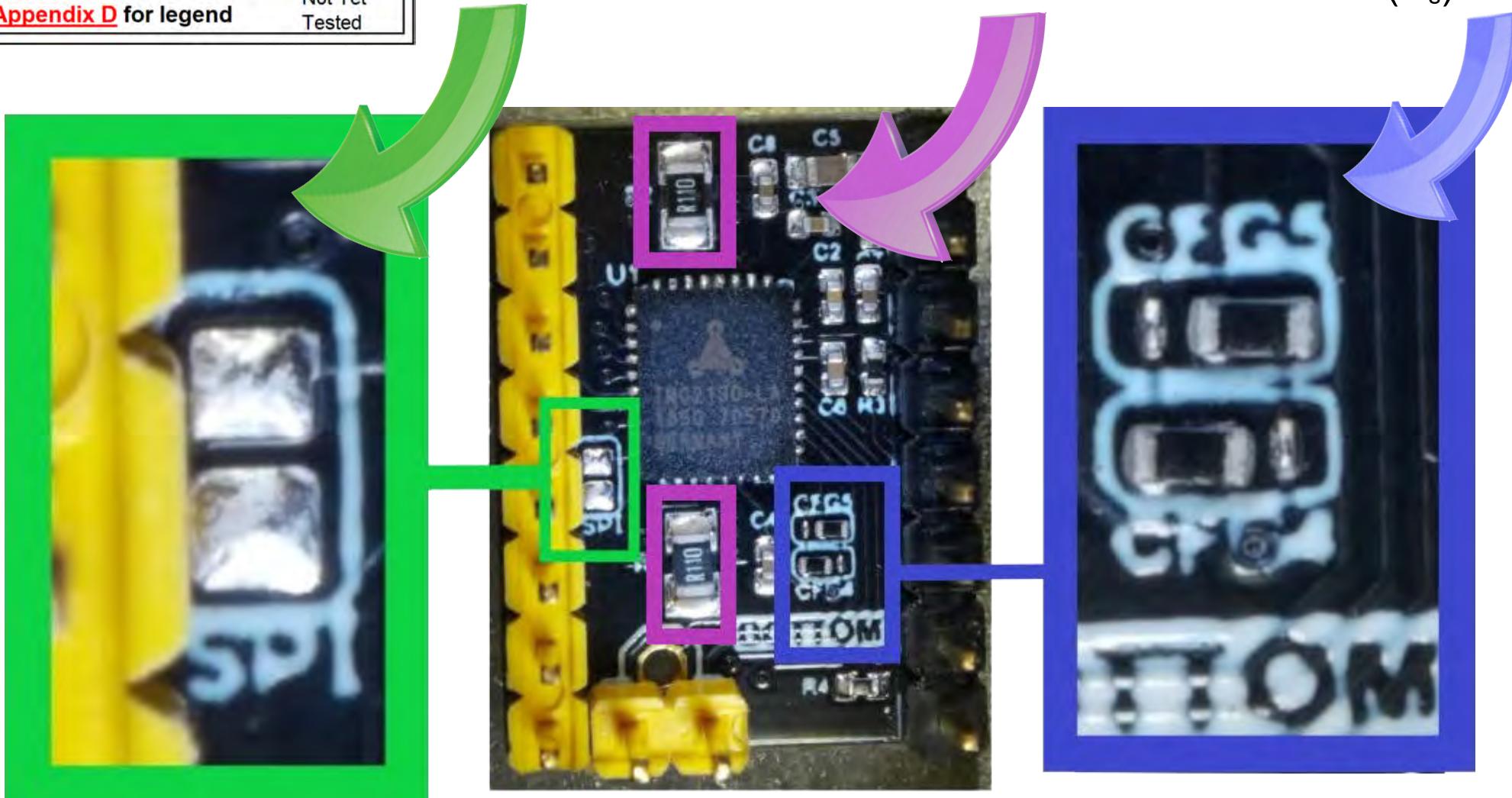


BIQU TMC2130

SPI Mode

To place the BIQU TMC2130 into SPI Mode:

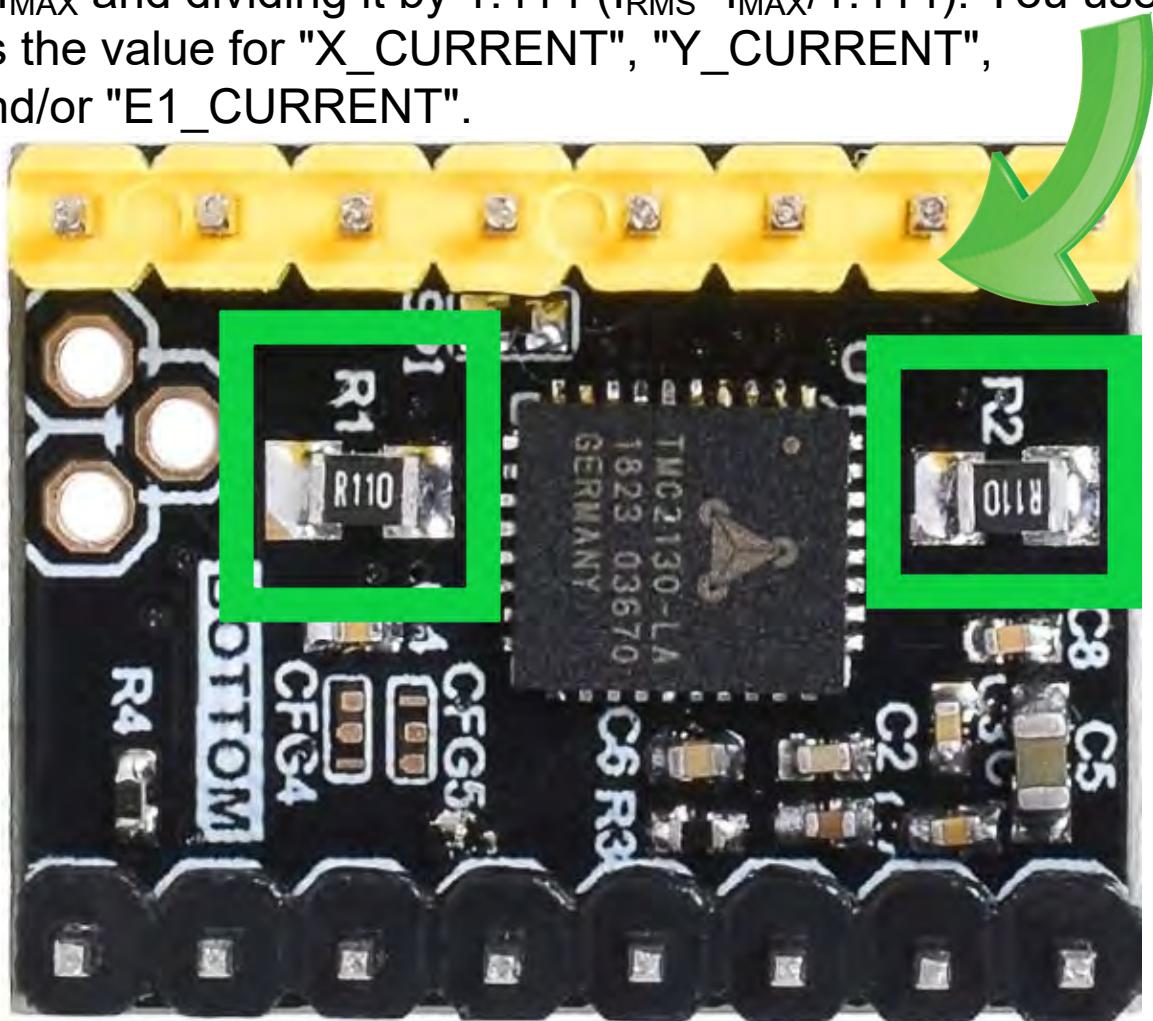
The SPI Jumper must have a gap between the two SPI pads, on the bottom of the driver board, as shown in the **GREEN** box below. Ensure that at CFG4 location and CFG5 location the correct two pads are soldered together to form a bridge, as shown in the **BLUE** box. The **PURPLE** box shows the location of the current sense resistors (R_s).



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

Note: The location of the current sense resistors are shown in **GREEN**. Use the current sense resistor's value in the Marlin Firmware ("X_RSENSE", "Y_RSENSE", "Z_RSENSE", "E0_RSENSE" and/or "E1_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use V_{ref} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", "E0_CURRENT" and/or "E1_CURRENT", you should use I_{RMS} instead. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS} = I_{MAX}/1.414$). You use 50% to 90% of the calculated I_{RMS} as the value for "X_CURRENT", "Y_CURRENT", "Z_CURRENT", "E0_CURRENT", and/or "E1_CURRENT".

- $R_s = R_{050}$ is 0.05 Ohms**
- $R_s = R_{062}$ is 0.062 Ohms**
- $R_s = R_{068}$ is 0.068 Ohms**
- $R_s = R_{075}$ is 0.075 Ohms**
- $R_s = R_{100}$ is 0.1 Ohms**
- $R_s = R_{110}$ is 0.11 Ohms**
- $R_s = R_{150}$ is 0.15 Ohms**
- $R_s = R_{200}$ is 0.2 Ohms**
- $R_s = R_{220}$ is 0.22 Ohms**



SPI Mode

R - Stall Detection

X-	(XST) X-DIAG1	X-	(XST) X-DIAG1
Y-	(YST) Y-DIAG1	Y-	(YST) Y-DIAG1
Z-	(ZST) Z-DIAG1	Z-	(ZST) Z-DIAG1
+X	(E0ST) E0-DIAG1	+X	(E0ST) E0-DIAG1
+Y	(E1ST) E1-DIAG1	+Y	(E1ST) E1-DIAG1

BIQU TMC2130

SPI Mode

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.



(XST)	X-DIAG1	X-
(YST)	Y-DIAG1	Y-
(ZST)	Z-DIAG1	Z-
(E0ST)	E0-DIAG1	X+
(E1ST)	E1-DIAG1	Y+

Note: If sensor-less homing is **not wanted** ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".



SPI Mode

R - Stall Detection

BIQU TMC2130
SPI Mode

Note: Ensure that **ALL** the UART jumpers are **EMPTY**.



SPI Mode**R - Stall Detection****BIGU TMC2130**
SPI Mode**SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode**

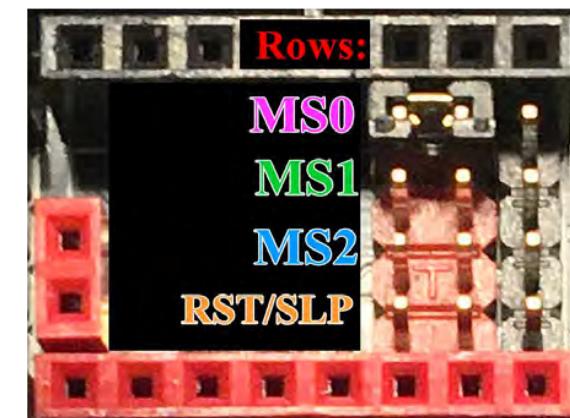
Example of the **WRONG** way to set the “**S**” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “**S**” jumper for SPI mode (left two columns of pins):



S S → Jumper set



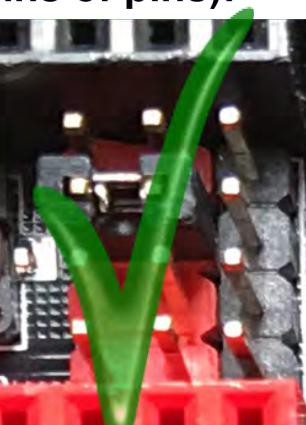
SET ‘**ST**’ JUMPER to enable Sensor-less Homing



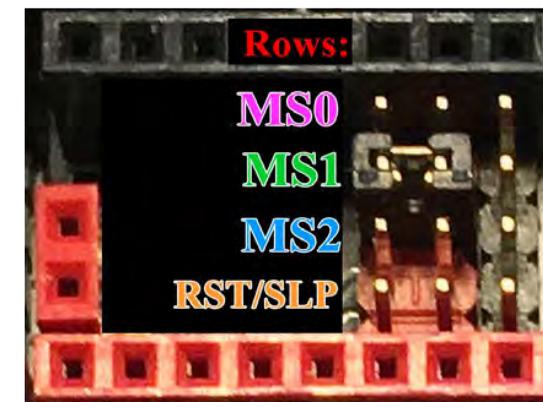
Example of the **WRONG** way to set the “**P**” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “**P**” jumper for SPI mode (left two columns of pins):



P P → Jumper set



SET ‘**ST**’ JUMPER to enable Sensor-less Homing

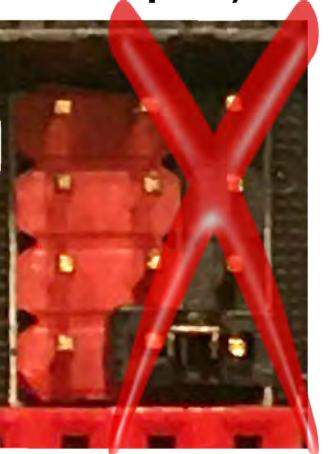


SPI Mode**R - Stall Detection**

Example of the **WRONG** way to set the “I” jumper for SPI mode (right two columns of pins):



Example of the **WRONG** way to set the “M” jumper for SPI mode (right two columns of pins):

**BIQU TMC2130**SPI Mode

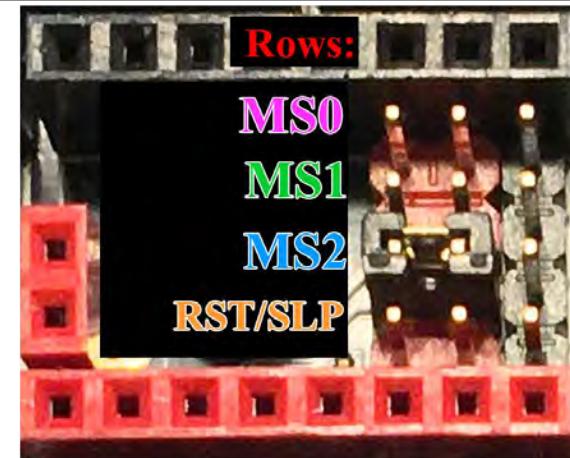
Example of the **RIGHT** way to set the “I” jumper for SPI mode (left two columns of pins):



Example of the **RIGHT** way to set the “M” jumper for SPI mode (left two columns of pins):



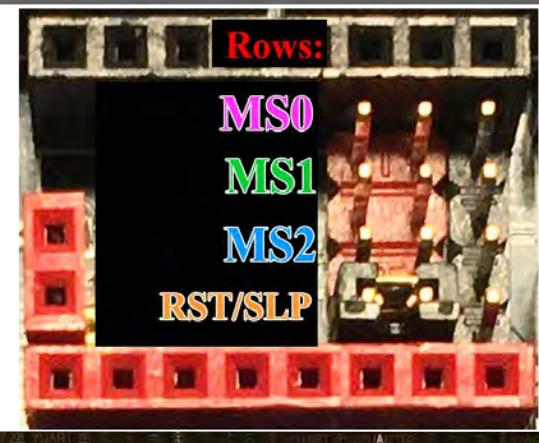
I **I** **→ Jumper set**



SET ‘**ST**’ JUMPER to enable Sensor-less Homing

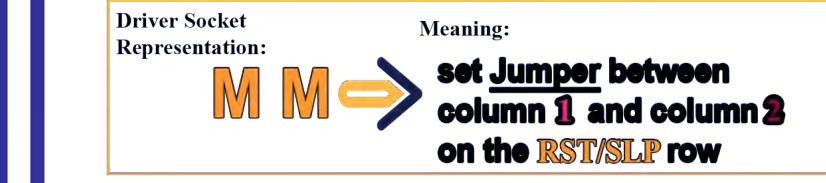
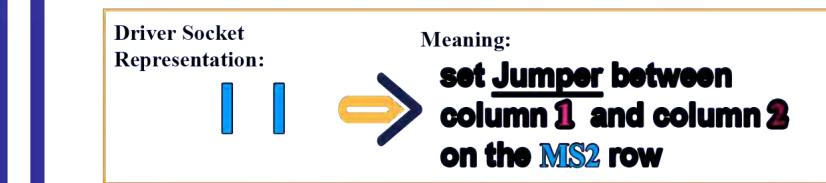
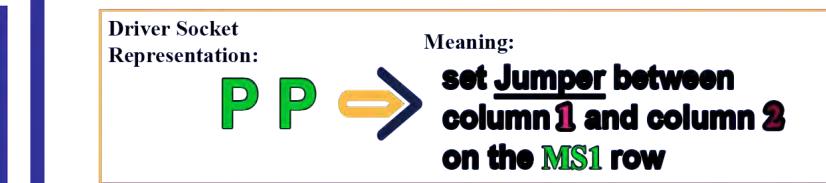
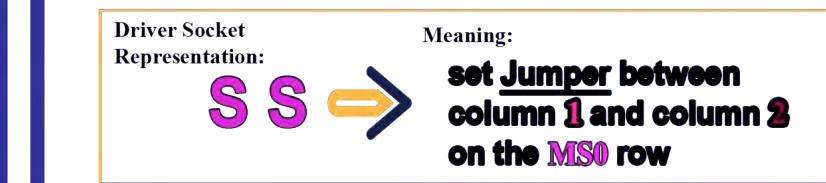
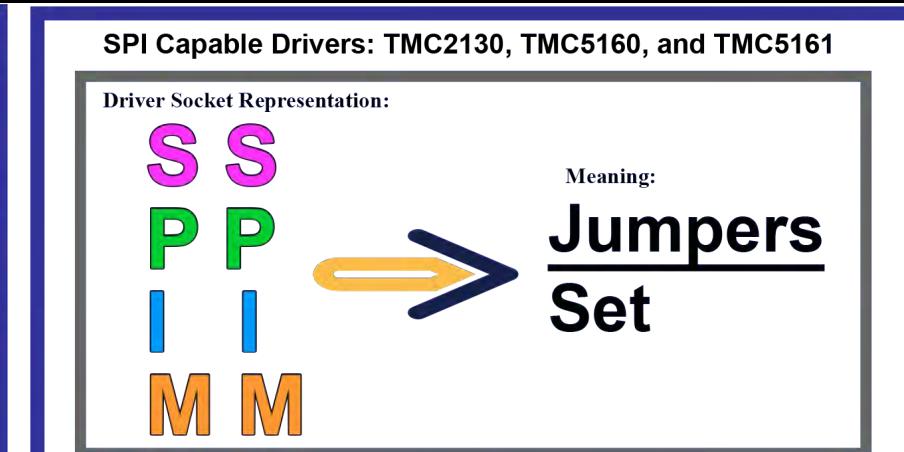
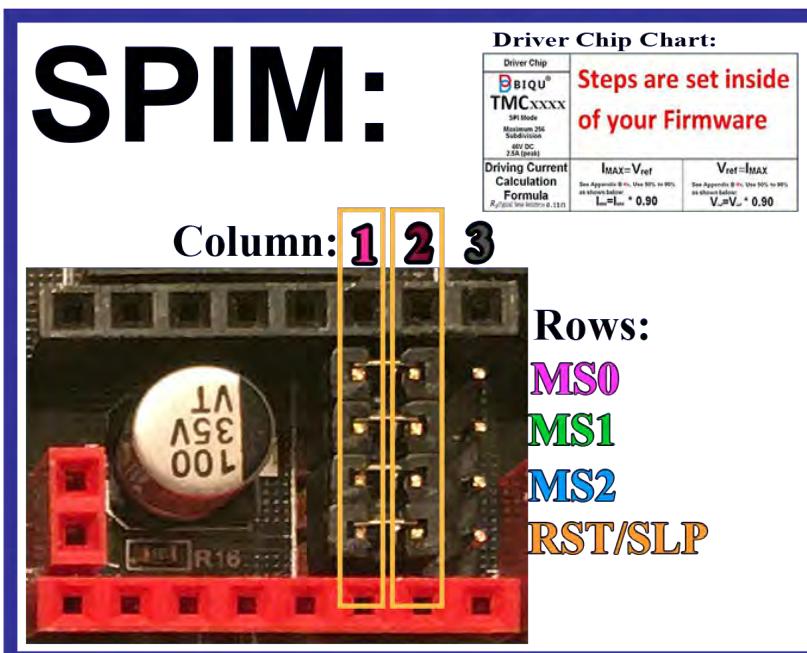
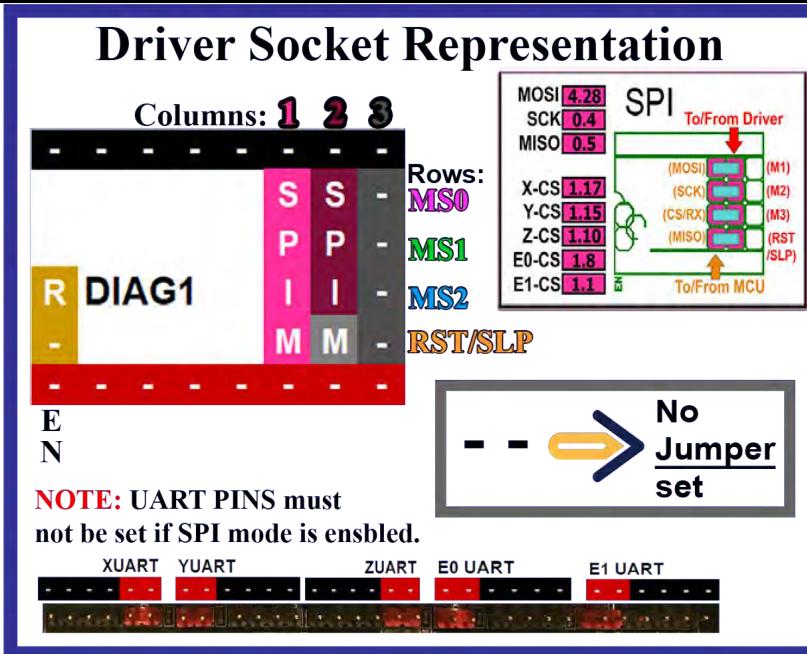


M **M** **→ Jumper set**



SET ‘**ST**’ JUMPER to enable Sensor-less Homing



SPI Mode**R - Stall Detection****BIQU TMC2130**SPI Mode**SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode**

SPI Mode**R - Stall Detection****BQU TMC2130****SPI Mode****Information on Sensor-less Homing****Driver Socket Representation****SPI Capable:**

Columns: 1 2 3



**NOTE: UART PINS
must not be set if SPI
mode is enabled.**



**SET 'ST' JUMPER to
enable Sensor-less Homing:**



XUART YUART ZUART E0 UART E1 UART

UART Capable:

Columns: 1 2 3



**NOTE: SPI
PINS must
not be set if
UART mode
is enabled.**

STALLGUARD

(Sensor-less Homing)

DIAG PIN ENDSTOP

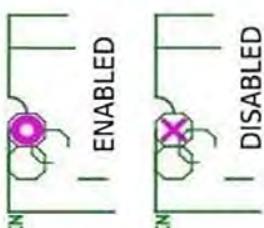
X X-DIAG1 1.29 X-

Y Y-DIAG1 1.27 Y-

Z Z-DIAG1 1.25 Z-

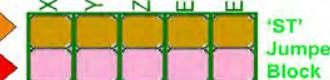
E0 E0-DIAG1 1.28 X+

E1 E1-DIAG1 1.26 Y+



To/From Driver DIAG pin

To/From MCU Endstops



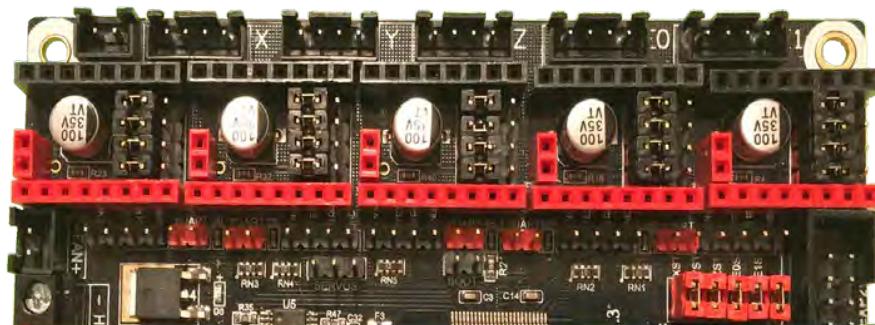
Note Concerning the TMC2209 in UART Mode ONLY:

If using limit switches/endstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

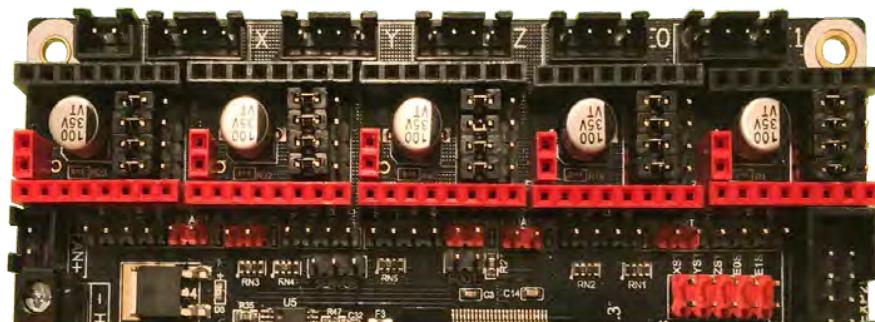
Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

Sensor-less Homing Capable Drivers:**SPI Capable Drivers:** TMC2130, TMC5160 & TMC5161**UART Capable Drivers:** TMC2209

If you **want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block.** The example below shows **ALL Axes have UART set and the 'ST' JUMPERS are SET for ALL Axes:**

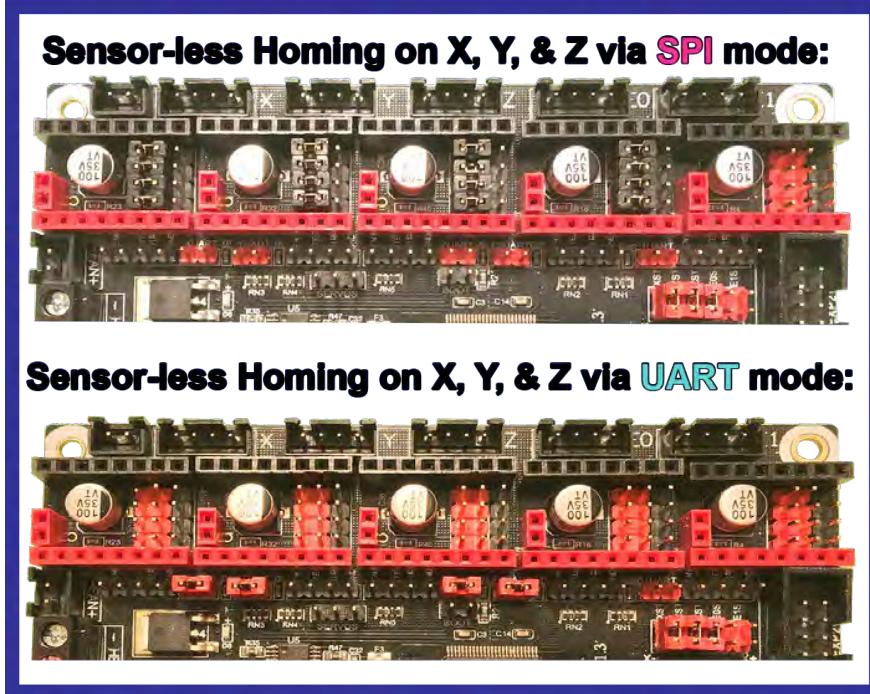
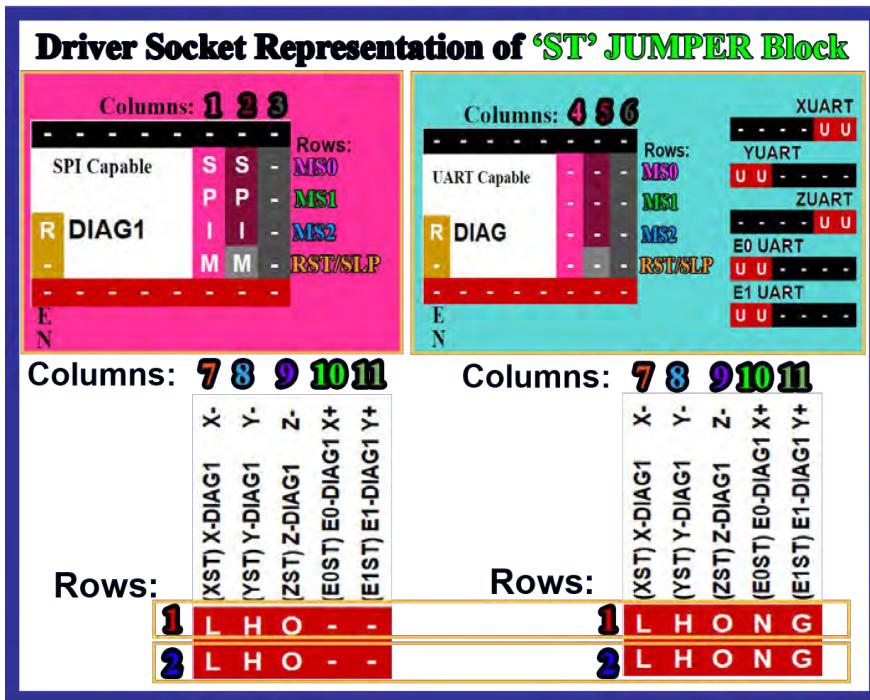


If you **do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block.** The example below shows **ALL Axes have UART set and the 'ST' JUMPERS are REMOVED for ALL Axes:**



SPI Mode

R - Stall Detection



Biqu TMC2130

SPI Mode

Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

X MIN Endstop to enable Sensor-less Homing

Y MIN Endstop to enable Sensor-less Homing

Z MIN Endstop to enable Sensor-less Homing

X MAX Endstop to enable Sensor-less Homing

Y MAX Endstop to enable Sensor-less Homing

No Jumper Set

Driver Socket Representation: Meaning: set Jumper between row 1 and row 2 in column 7

Driver Socket Representation: Meaning: set Jumper between row 1 and row 2 in column 8

Driver Socket Representation: Meaning: set Jumper between row 1 and row 2 in column 9

Driver Socket Representation: Meaning: set Jumper between row 1 and row 2 in column 10

Driver Socket Representation: Meaning: set Jumper between row 1 and row 2 in column 11

SPI Mode

R - Stall Detection

Use the '**ST**' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The '**ST**' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capability of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a '**ST**' JUMPER in the '**ST**' Jumper Block for that Axis.

The way you ensure the DIAG PIN is **NOT** connected to the MCU Endstop for the Axis is by ensuring the corresponding '**ST**' JUMPER is removed from the '**ST**' Jumper Block for that particular Axis.

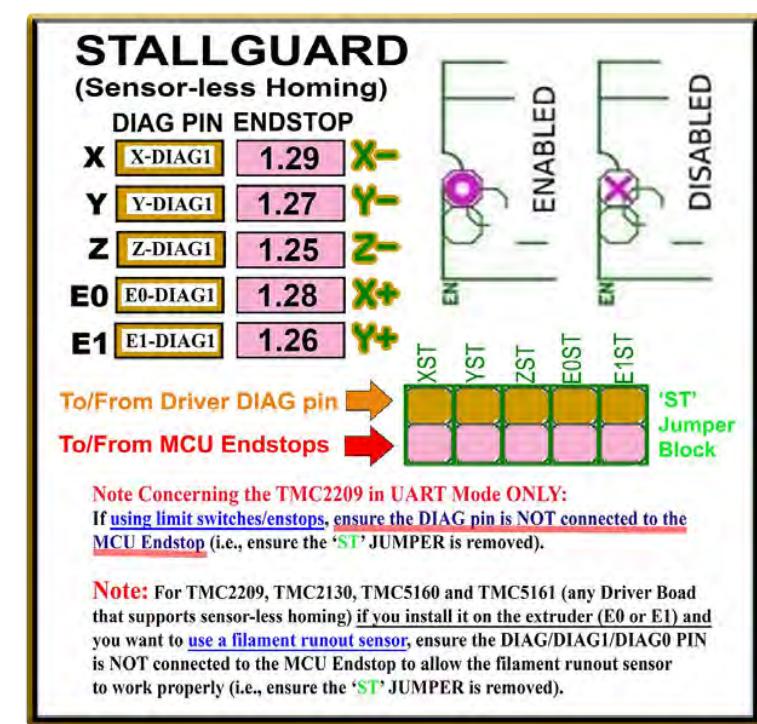
NOTE: The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode **AND** you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is **NOT** connected to the MCU Endstop (i.e., remove the '**ST**' JUMPER in the '**ST**' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location **AND** you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is **NOT** connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the '**ST**' JUMPER is removed from the '**ST**' Jumper Block for that axis).

BQU TMC2130

SPI Mode



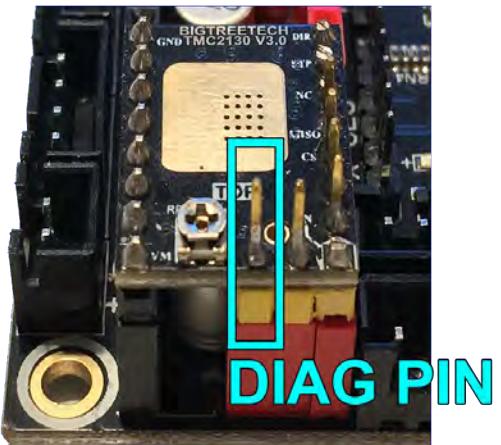
If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode **AND** you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the '**ST**' JUMPER is removed from the '**ST**' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the '**ST**' JUMPER is removed from the '**ST**' Jumper Block.

The following DO NOT have Endstops:

- Extruder Axis (E0 or E1)
- Z Axis, if a BLTouch is used to Home instead of a physical endstop

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

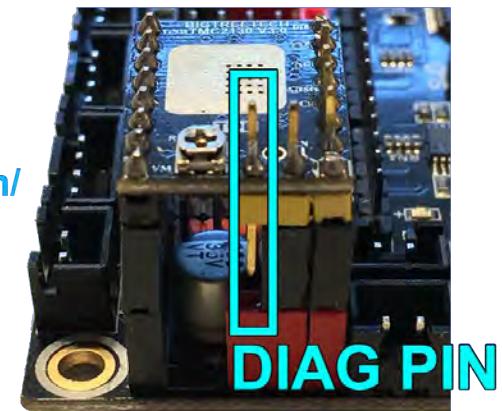
To enable sensor-less homing ensure that the **DIAG PIN is plugged into the SKR V1.3 Board **AND** the '**ST**' JUMPER is set in the '**ST**' JUMPER Block for the Axis.**



To disable sensor-less homing either ensure that the **DIAG PIN is **NOT** plugged into the SKR V1.3 Board **OR** remove the '**ST**' JUMPER from the '**ST**' JUMPER Block.**

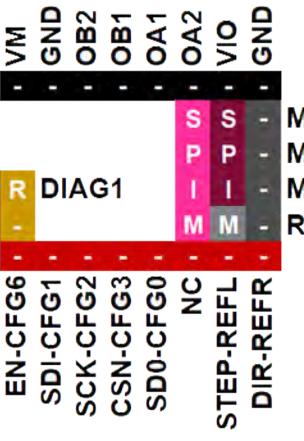
Link to stackable header pins:

<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>

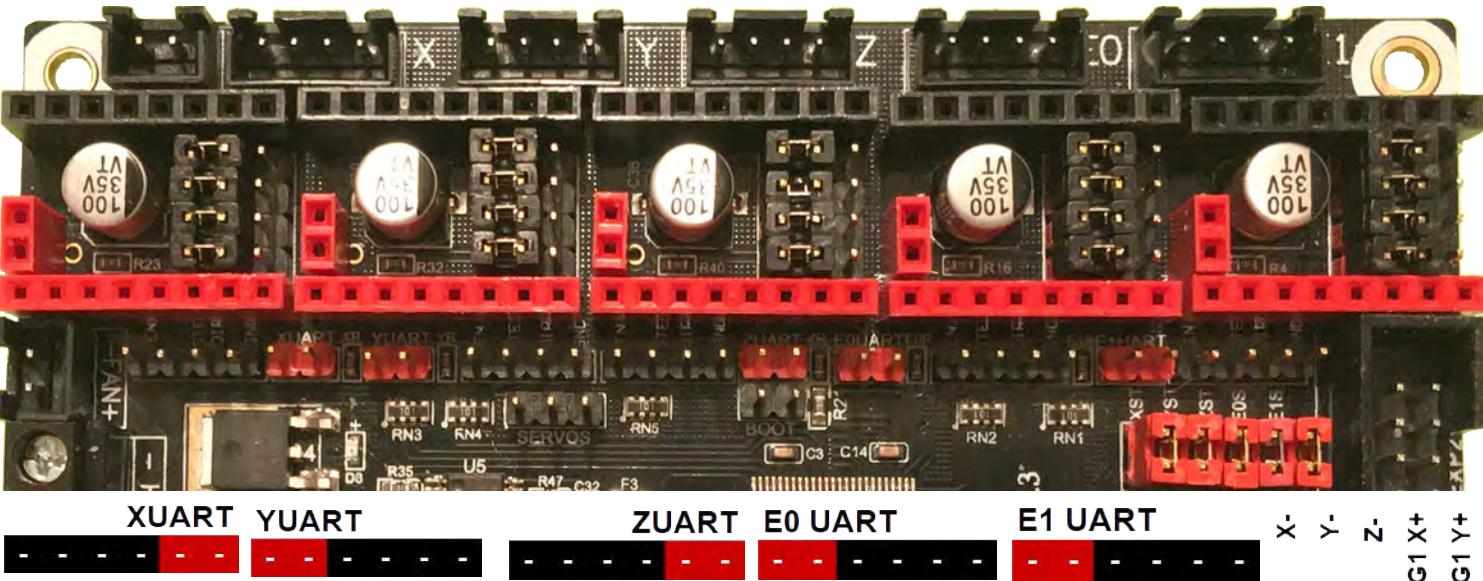


SPI Mode**R - Stall Detection****BIQU TMC2130**SPI Mode

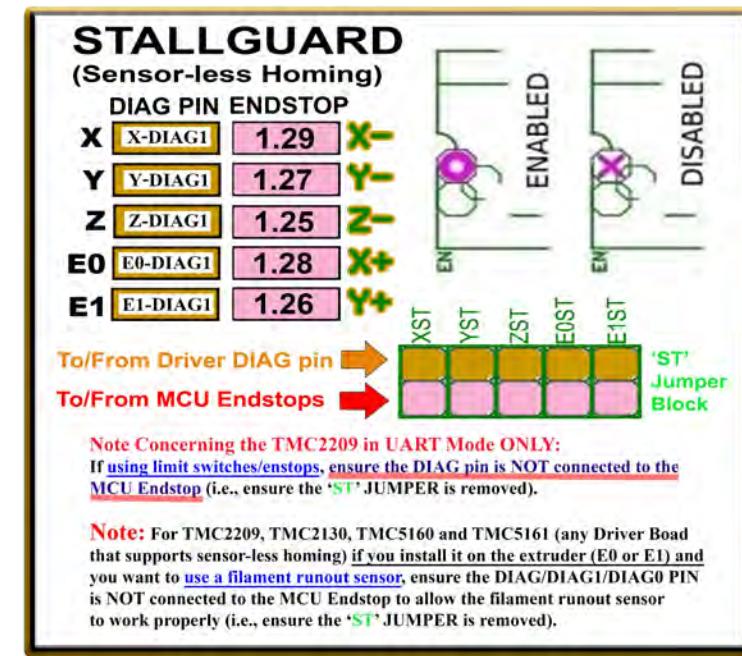
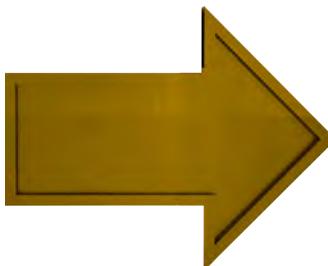
Note: Set JUMPERS "S", "P", "I", and "M" on the board!!



See [Appendix D](#) for legend



Note: TMC2130 has sensor-less homing capability



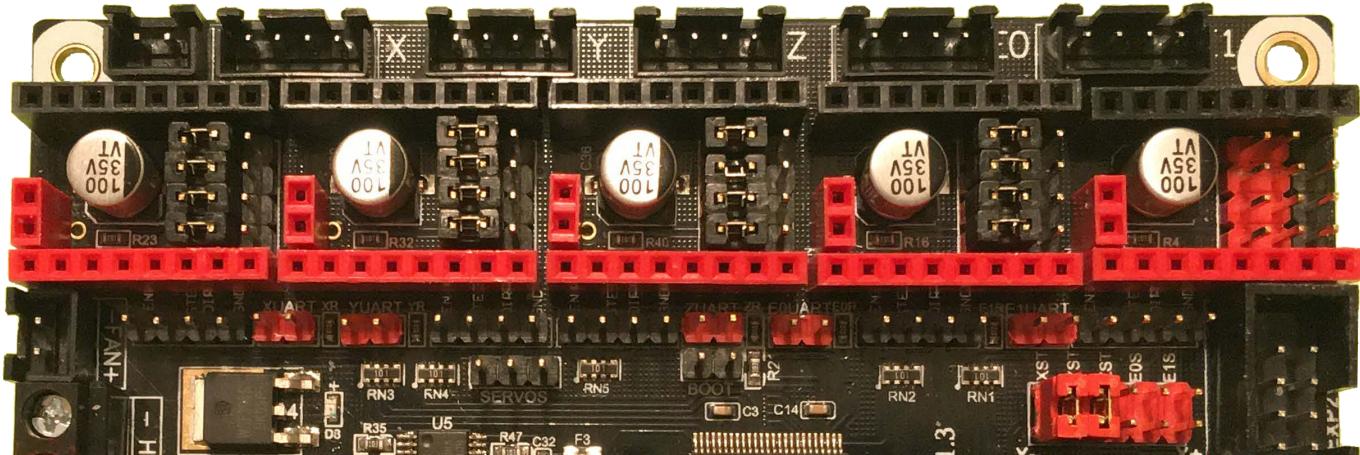
(XST) X-DIAG1	(YST) Y-DIAG1	(ZST) Z-DIAG1	(E0ST) E0-DIAG1	(E1ST) E1-DIAG1
L H O N G	L H O N G	L H O N G	L H O N G	L H O N G

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

Examples of Different SPI Configurations Part 1

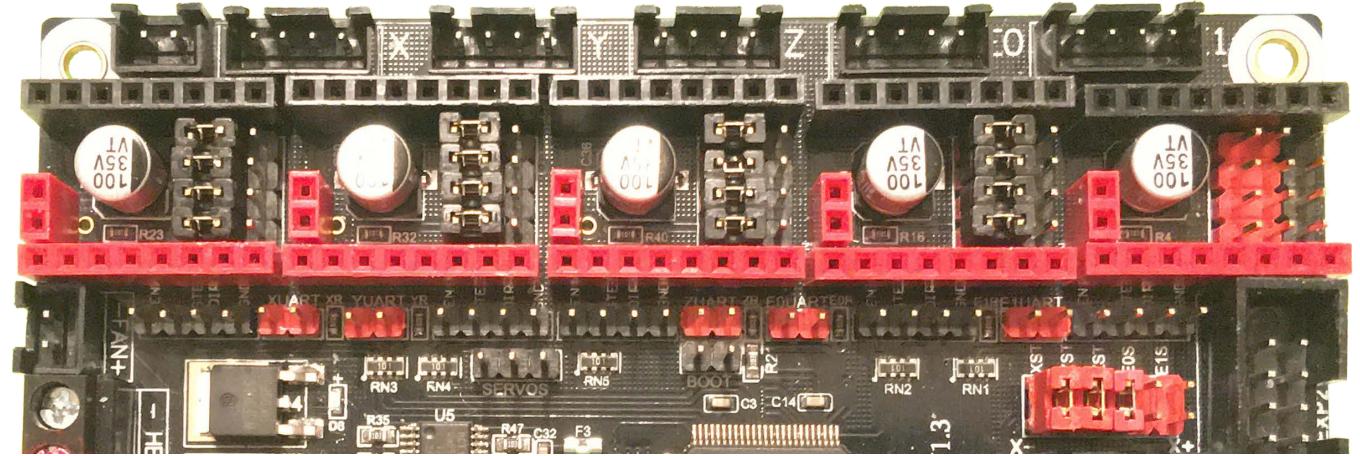
**X, Y, Z and E0 axes
configured for SPI mode.**

**Sensor-less homing for X
and Y axes.**



**X, Y, Z and E0 axes
configured for SPI mode.**

**Sensor-less homing for X,
Y and Z axes.**

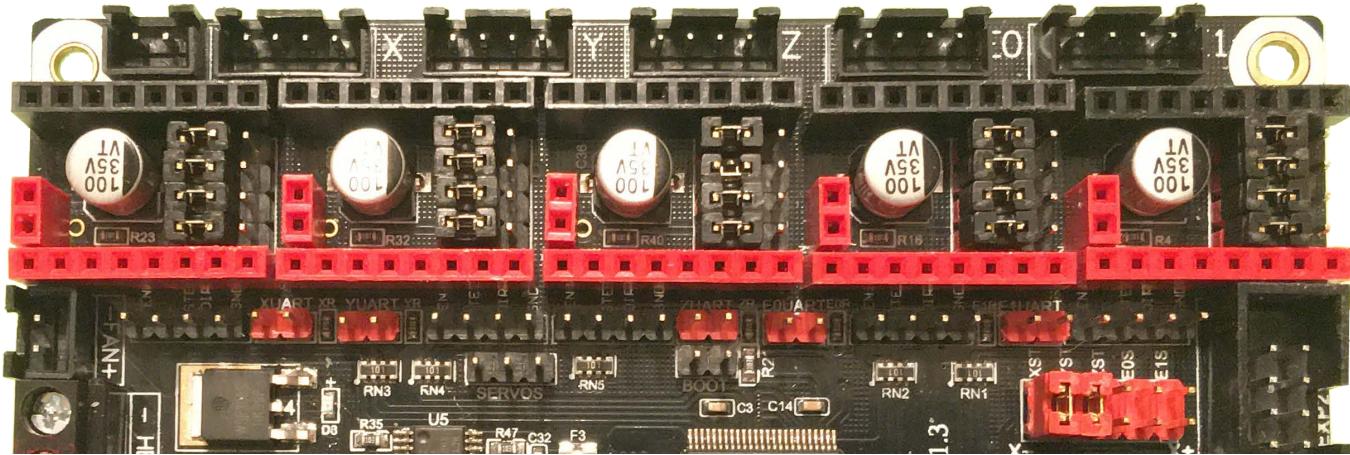


SPI Mode**R - Stall Detection****BIGU TMC2130**SPI Mode

Examples of Different SPI Configurations Part 2

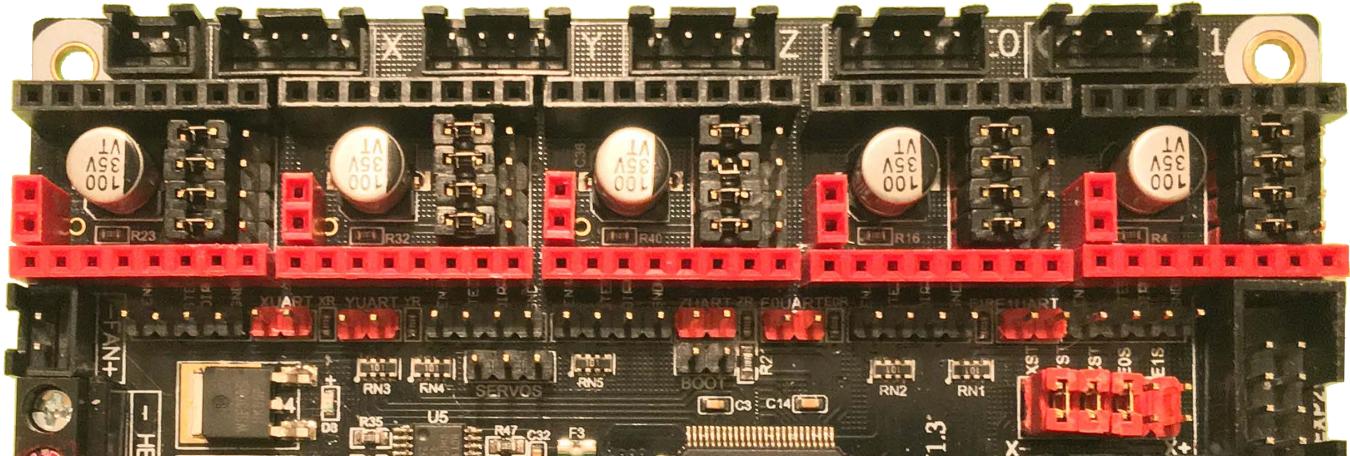
**X, Y, Z, E0, and E1 axes
configured for SPI mode.**

**Sensor-less homing for X
and Y axes.**



**X, Y, Z, E0 and E1 axes
configured for SPI mode.**

**Sensor-less homing for X,
Y and Z axes.**



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

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2130 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2130 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2130 drivers in SPI mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

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

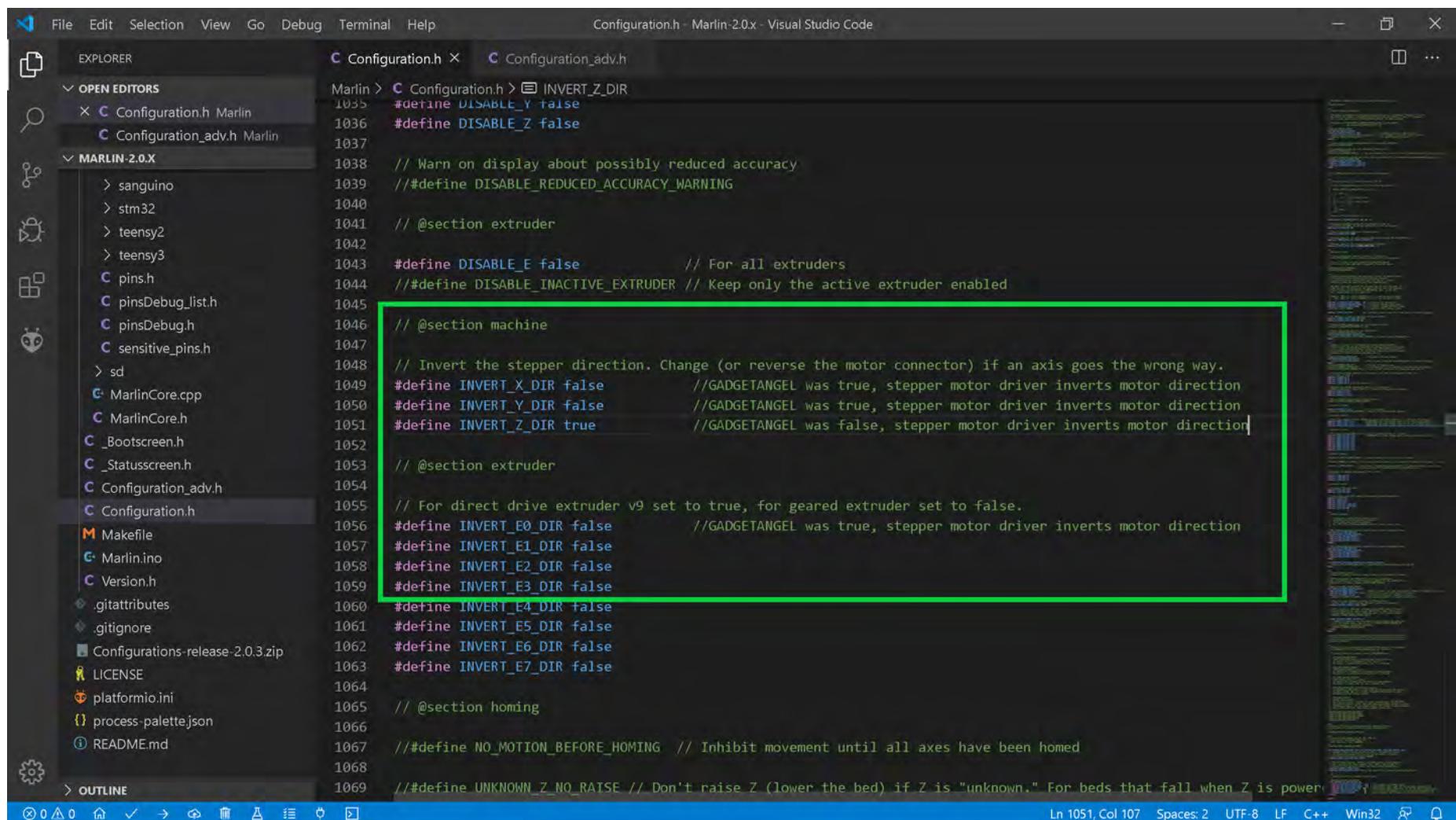
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home 661 /**
  * Stepper Drivers
  *
  * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  *
  * A4988 is assumed for unspecified drivers.
  *
  * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
  *          TB6560, TB6600, TMC2100,
  *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
  *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
  *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
  *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
  *          :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
  */
#define X_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define E0_DRIVER_TYPE TMC2130 //GADGETANGEL was commented out and had the value A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
// Enable this feature if all enabled endstop pins are interrupt-capable.

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

- Go to the next page.

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

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2130 drivers, I must invert the stepper motor direction because the TMC2130 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2130 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



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

```

Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false           // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false        // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true         // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false       // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

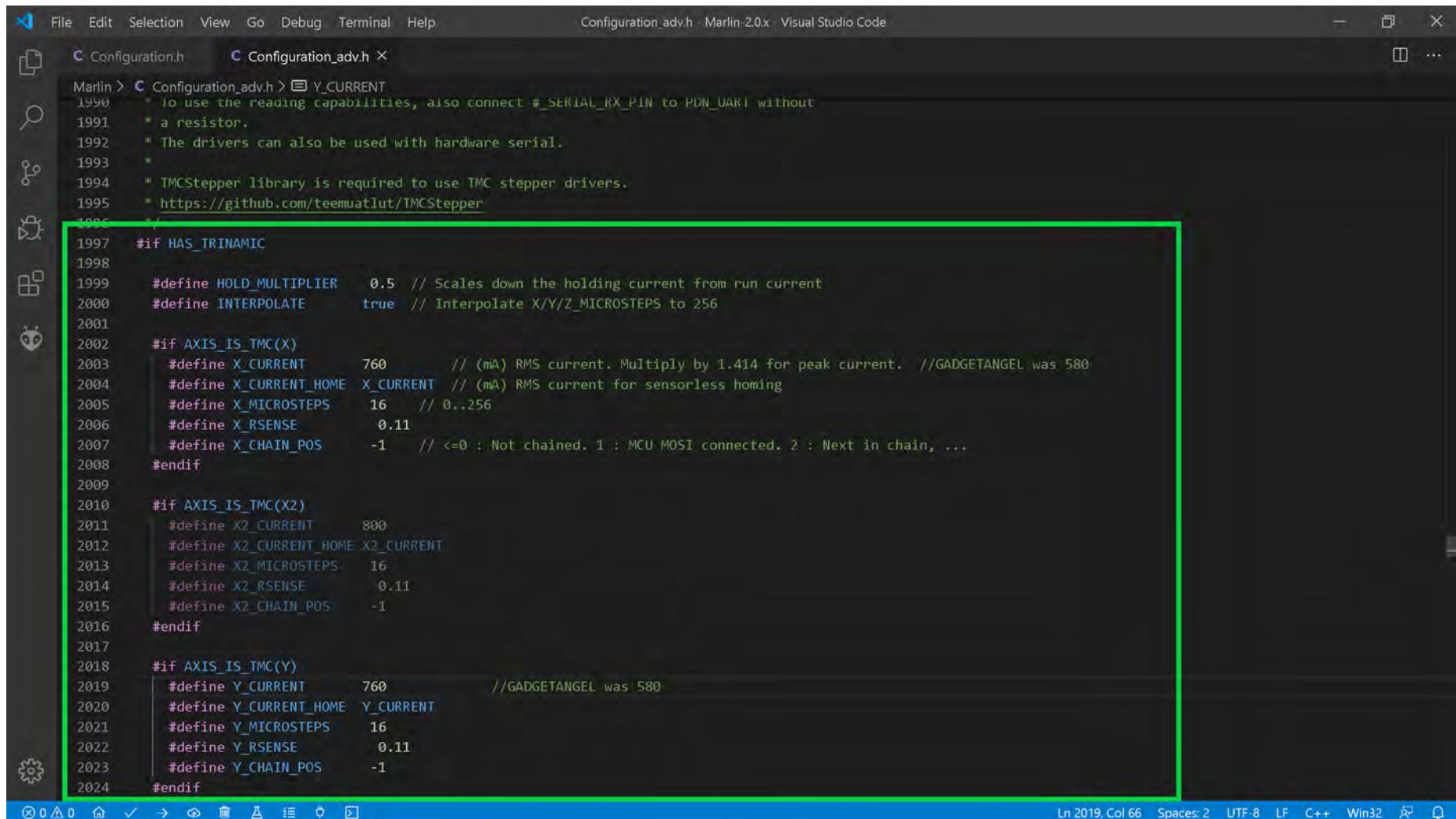
```

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

- Go to the next page.

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

- Next you want to set your V_{ref} in the Marlin firmware for each axis that has the TMC2130 driver, as seen in the **GREEN** box below. I changed the "X_CURRENT" to be the calculated V_{ref} for my X-Axis, which is 760mV for an Ender 3. I changed the "Y_CURRENT" to be the calculated V_{ref} for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X_RSENSE" is set to 0.11. Ensure "Y_RSENSE" is set to 0.11.
- If you **do not want to use V_{ref}** as the value for "X_CURRENT" and/or "Y_CURRENT", you should **use I_{RMS} instead**. You find I_{RMS} by taking I_{MAX} and dividing it by 1.414 ($I_{RMS}=I_{MAX}/1.414$). You use **50% to 90% of the calculated I_{RMS}** as the value for "X_CURRENT" and/or "Y_CURRENT".



```

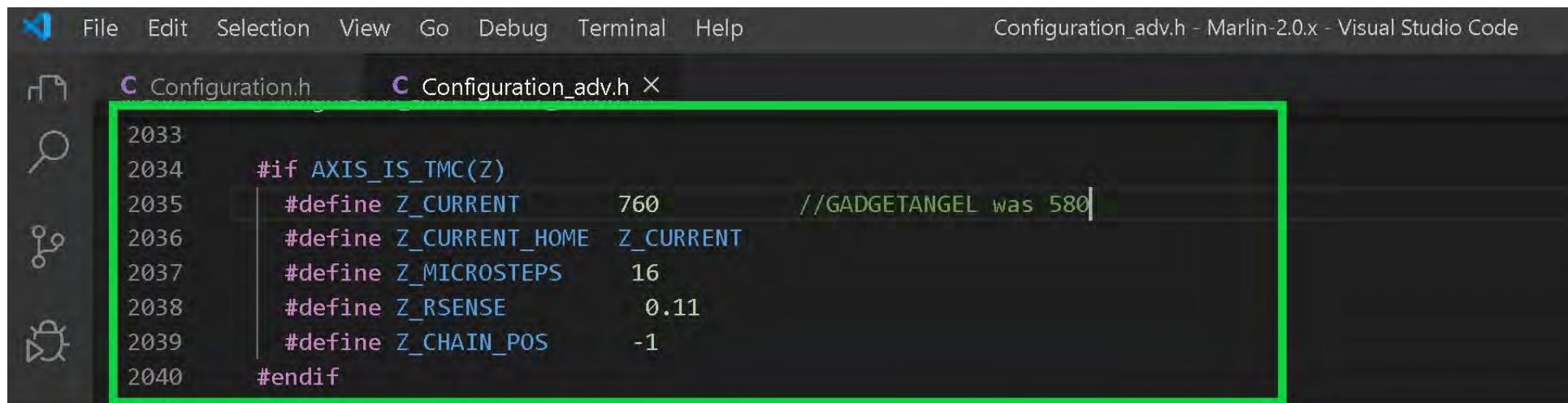
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008#endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016#endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.11
2023 #define Y_CHAIN_POS -1
2024#endif
Ln 2019, Col 66 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

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

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



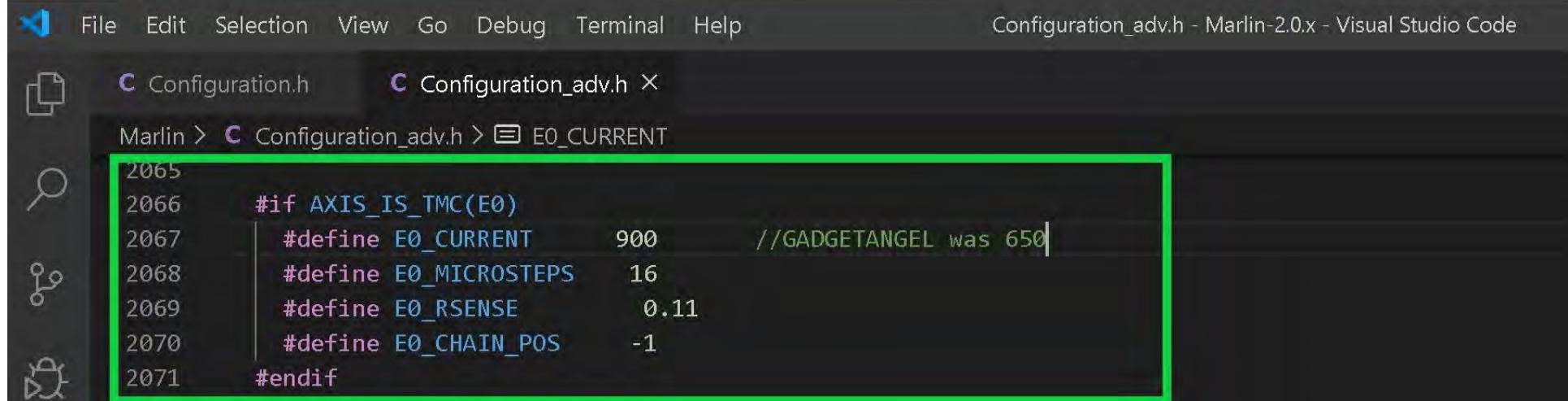
```

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

Configuration.h Configuration_adv.h X

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

```



```

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

Marlin > Configuration_adv.h > E0_CURRENT

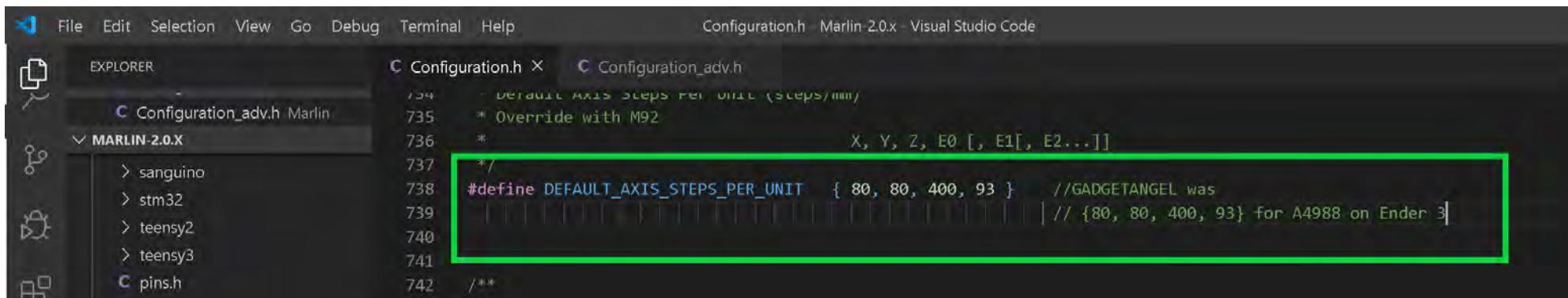
2065
2066 #if AXIS_IS_TMC(E0)
2067   #define E0_CURRENT      900          //GADGETANGEL was 650
2068   #define E0_MICROSTEPS   16
2069   #define E0_RSENSE        0.11
2070   #define E0_CHAIN_POS     -1
2071 #endif

```

- Go to the next page.

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

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT_AXIS_STEPS_PER_UNIT" to reflect your changes



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

EXPLORER Configuration.h X Configuration_adv.h

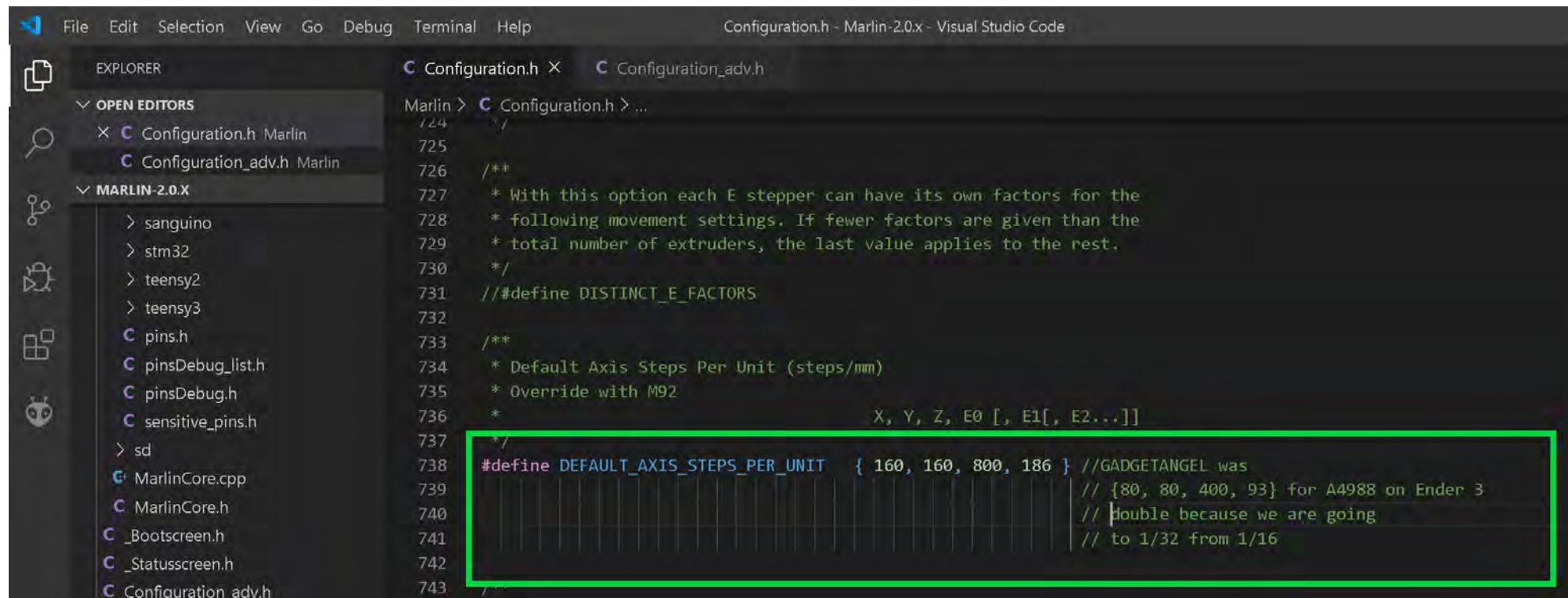
MARLIN-2.0.X

```

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

```

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



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

EXPLORER Configuration.h X Configuration_adv.h

MARLIN-2.0.X

```

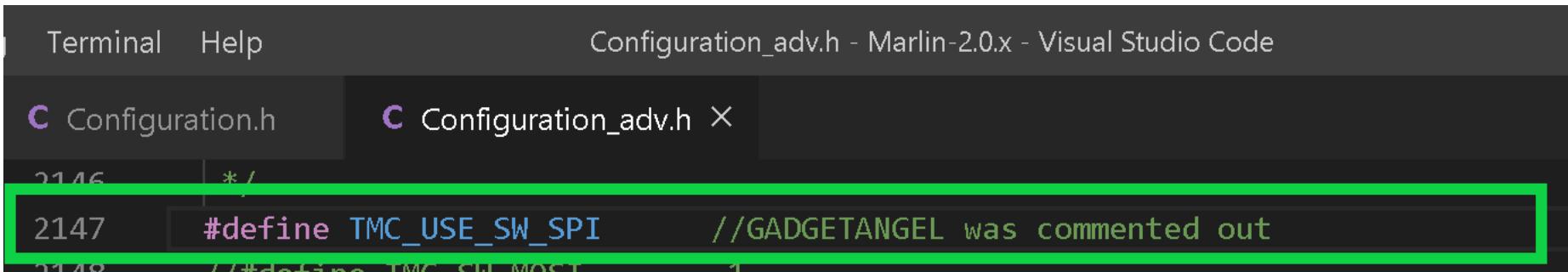
 724 */
 725
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 //#define DISTINCT_E_FACTORS
 732
 733 /**
 734 * Default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

```

- Go to the next page.

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

- We need to uncomment out the "TMC_USE_SW_SPI" because the SKR V1.3 pins file depends on this variable to define its SPI pins

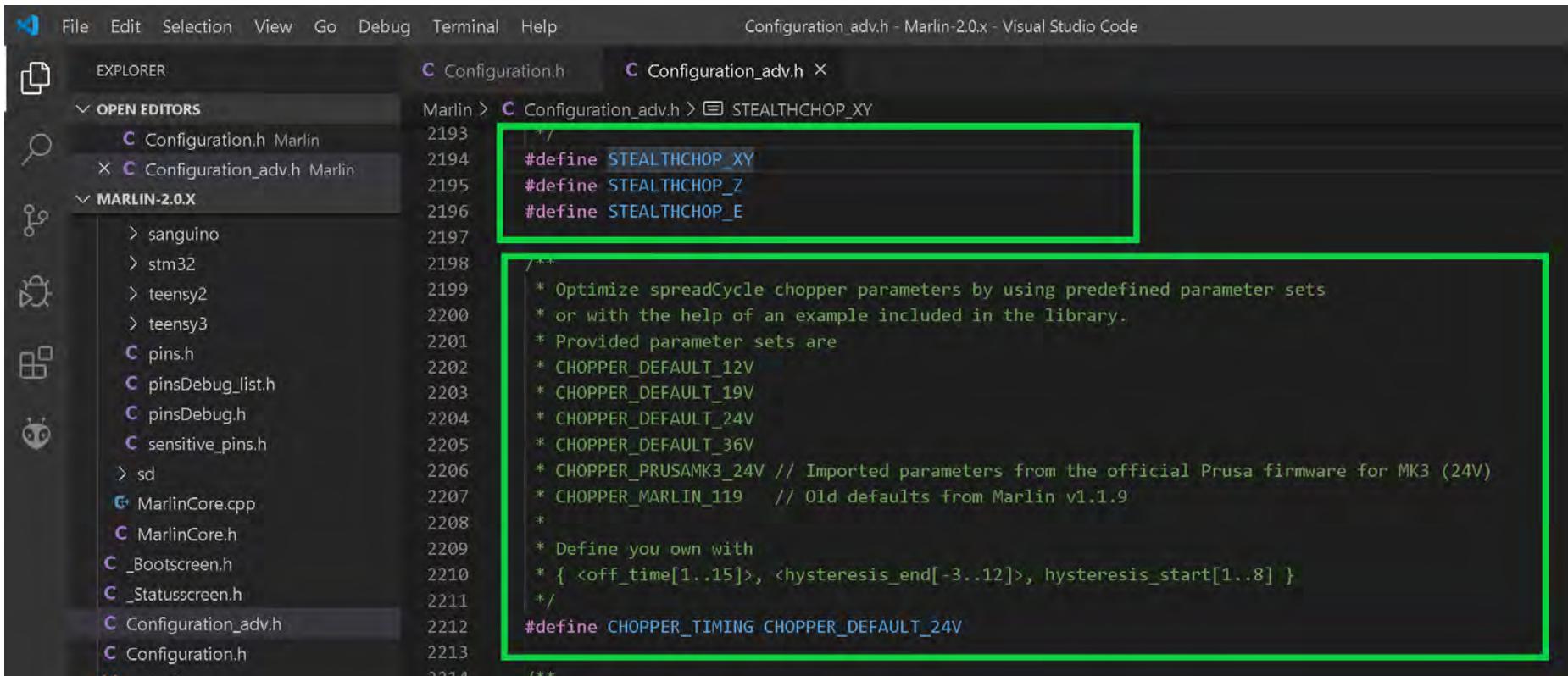


Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

C Configuration.h C Configuration_adv.h X

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

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



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

EXPLORER Configuration.h Configuration_adv.h X

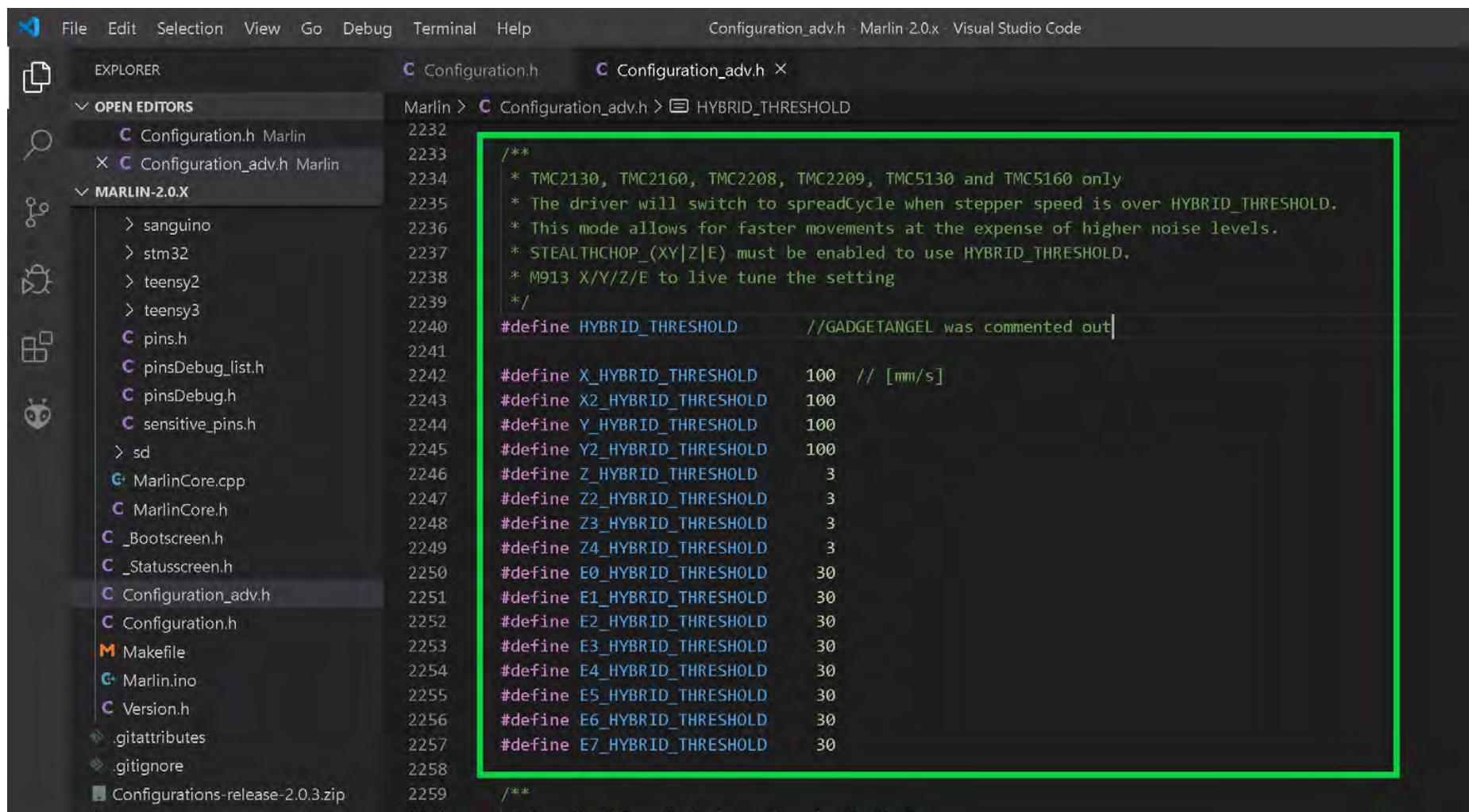
Marlin > C Configuration_adv.h > STEALTHCHOP_XY

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

- Go to the next page.

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

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



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

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
- Explorer:** Shows the project structure under MARLIN-2.0.X, including files like sanguino, stm32, teensy2, teensy3, pins.h, pinsDebug_list.h, pinsDebug.h, sensitive_pins.h, sd, MarlinCore.cpp, MarlinCore.h, _Bootscreen.h, _Statusscreen.h, Configuration_adv.h (which is the active editor), Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, and Configurations-release-2.0.3.zip.
- Editor:** The Configuration_adv.h file is open at line 2232. The code block for HYBRID_THRESHOLD is highlighted with a green border. The line containing the define is currently commented out with //.

```

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

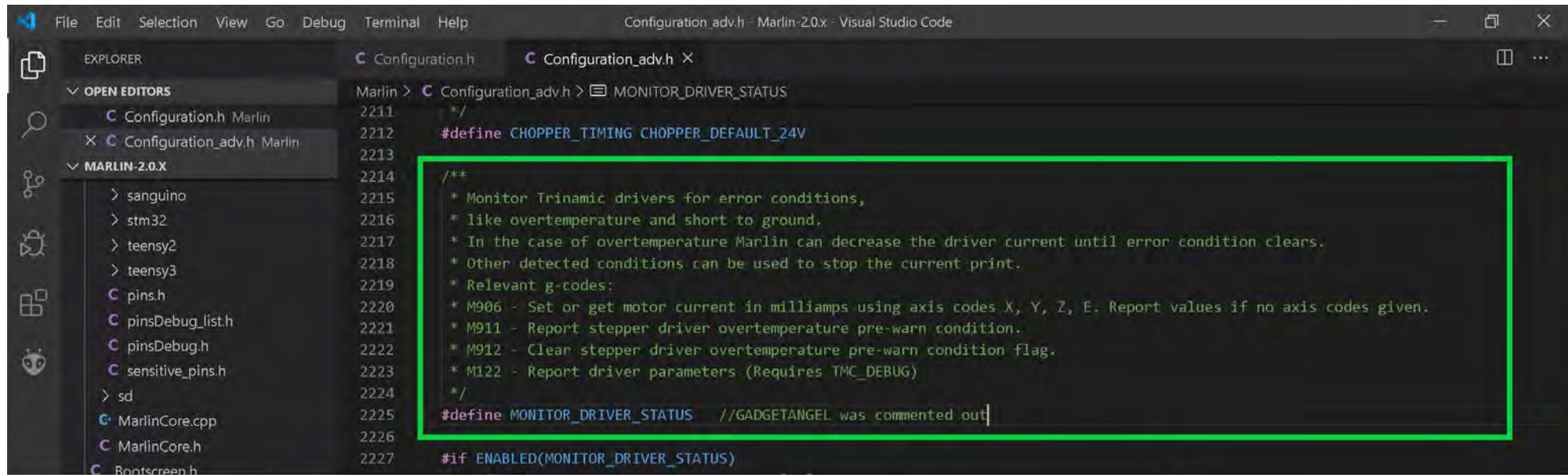
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

```

- Go to the next page.

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

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



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

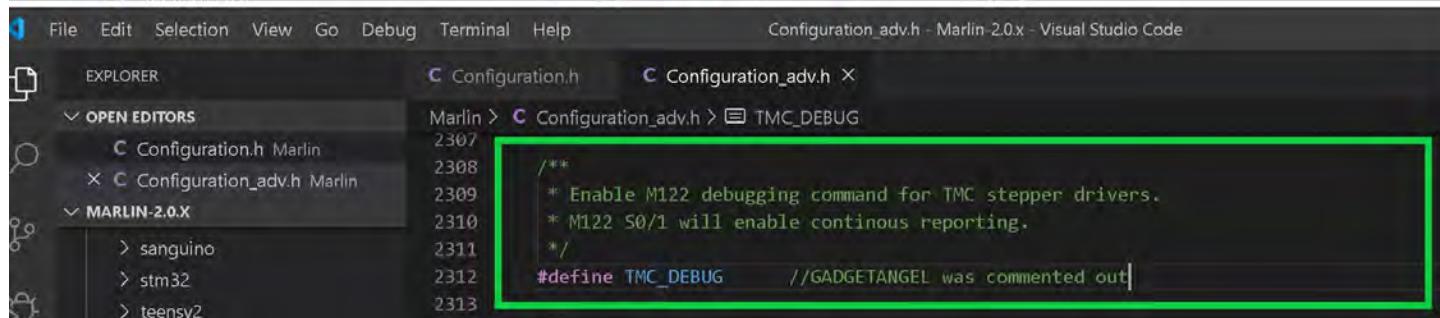
EXPLORER Configuration.h Configuration_adv.h

OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS

```

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

```



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

EXPLORER Configuration.h Configuration_adv.h

OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG

```

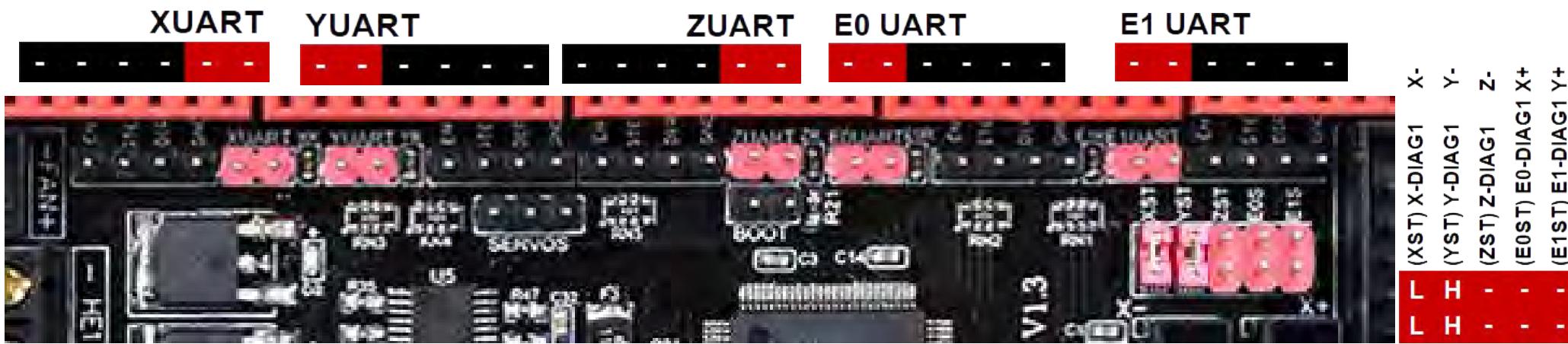
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

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

- This next section covers sensor-less homing which is available for the TMC2130 in SPI mode. I want to enable it so I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC2130 driver has to be plugged into the SKR V1.3 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC2130 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC2130 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC2130 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers are in place: the "L" and "H" jumpers.



Screenshot of a Visual Studio Code editor showing the `Configuration_adv.h` file for Marlin 2.0.x. The code includes a comment for sensorless homing:

```

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

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

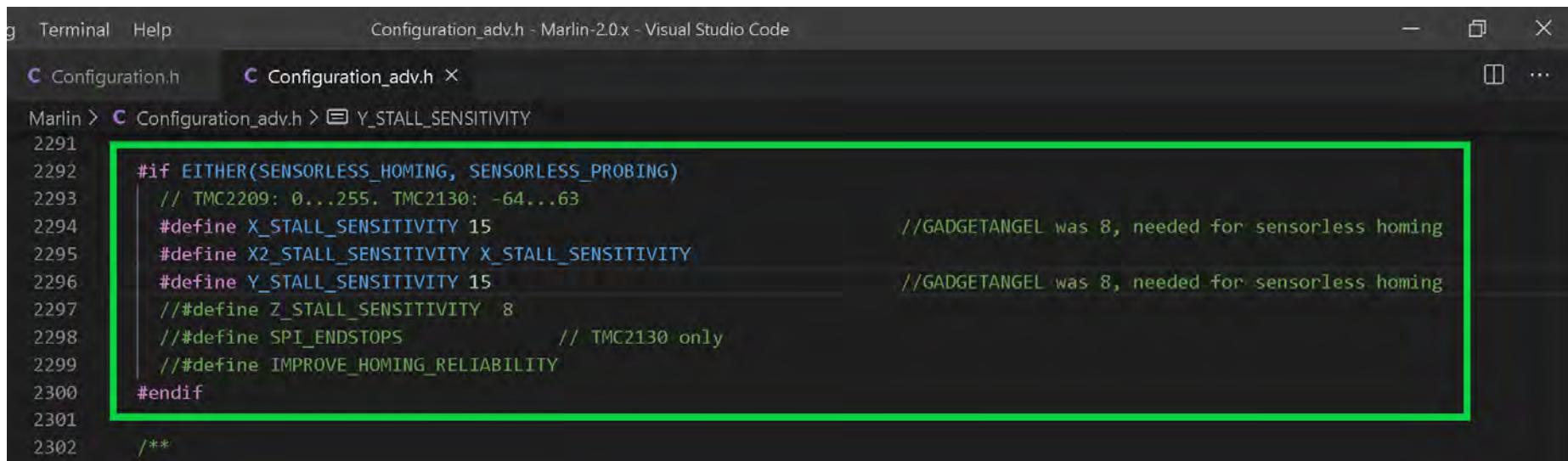
2281 */
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out
2283

```

- Go to the next page.

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

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

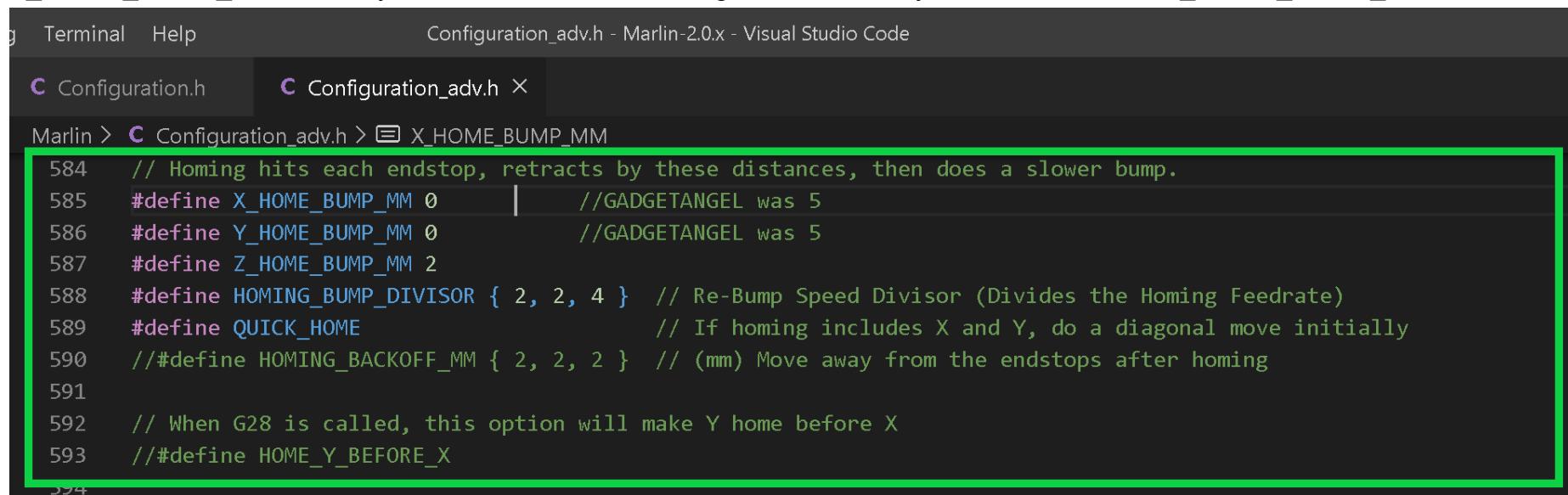


```

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

```

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



```

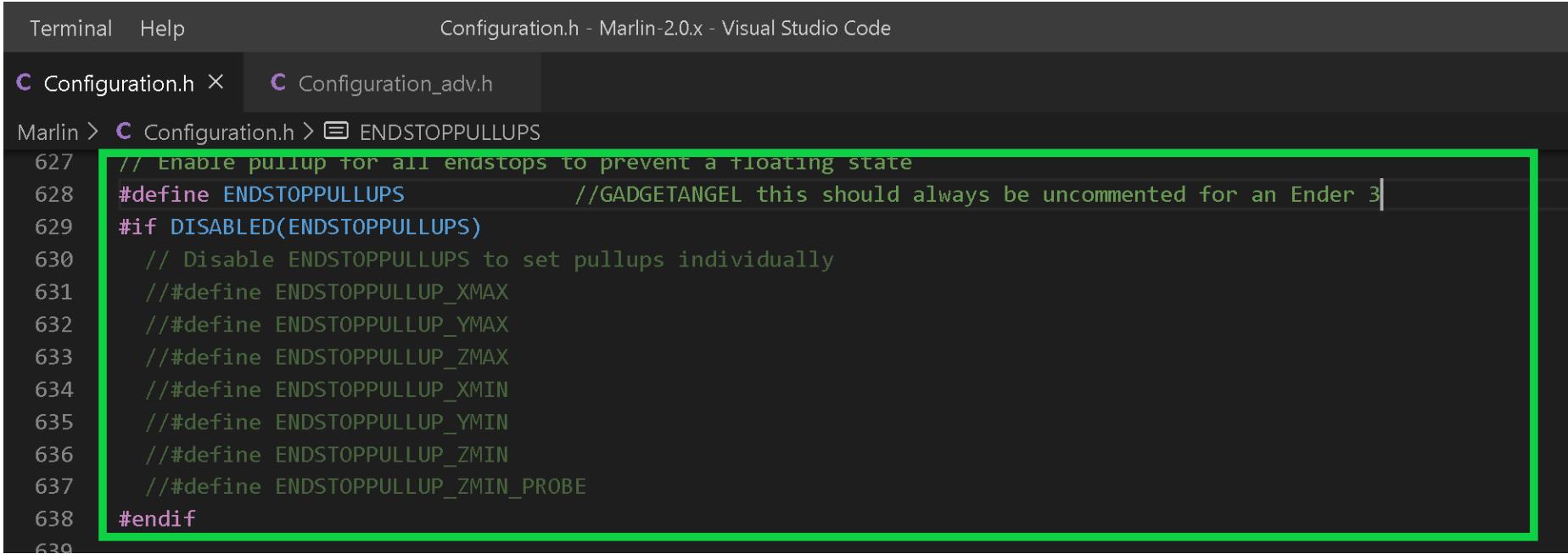
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586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

```

- Go to the next page.

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

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



```

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

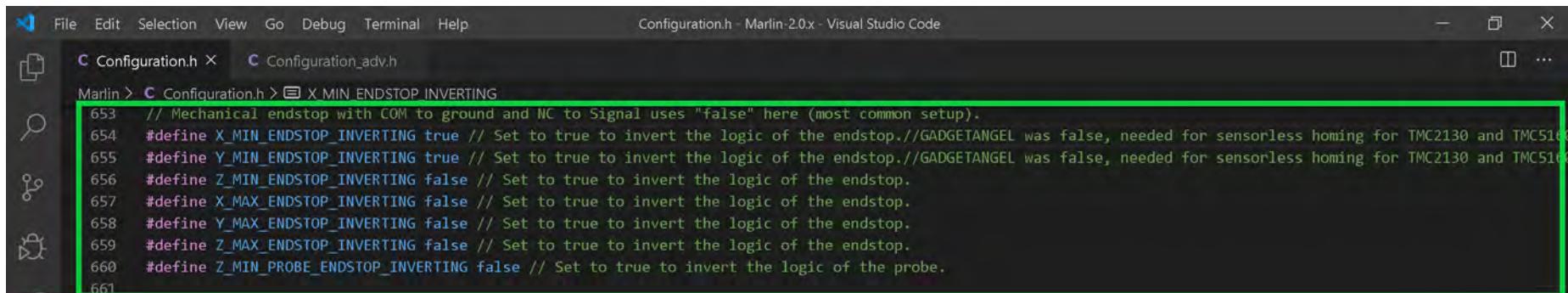
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > ENDSTOPPULLUPS

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

```

- Next to allow sensor-less homing to work (while using the BIQU TMC2130) we need to change our end stop logic. Therefore I set "X_MIN_ENDSTOP_INVERTING" to true and "Y_MIN_ENSTOP_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z_MIN_ENDSTOP_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z_MIN_ENDSTOP_INVERTING" set to false.



```

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

C Configuration.h X C Configuration_adv.h

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

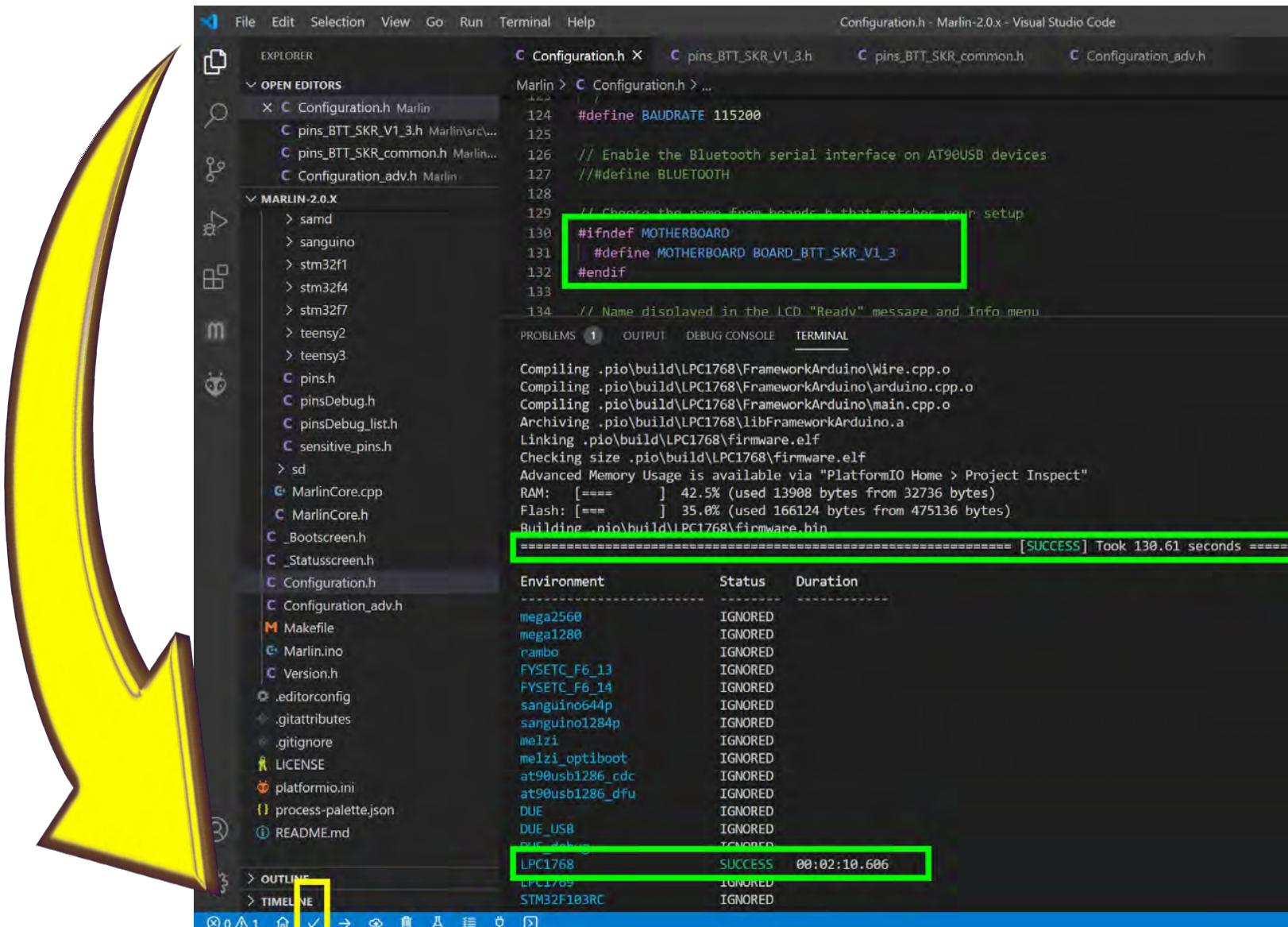
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5100
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5100
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661

```

- Go to the next page.

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

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



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

EXPLORER OPEN EDITORS MARLIN-2.0.X

Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src... pins_BTT_SKR_common.h Marlin... Configuration_adv.h Marlin

samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h _Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

OUTLINE TIMELINE

```

124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
126 // #define BLUETOOTH
127
128 // Choose the name from boards.h that matches your setup
129 #ifndef MOTHERBOARD
130 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
131 #endif
132
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
 Archiving .pio\build\LPC1768\libFrameworkArduino.a
 Linking .pio\build\LPC1768\firmware.elf
 Checking size .pio\build\LPC1768\firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
 Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
 Building .pio\build\LPC1768\firmware.bin
 ====== [SUCCESS] Took 130.61 seconds =====

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUE_debug	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

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

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

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

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

EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h

OPEN EDITORS Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src... pins_BTT_SKR_common.h Marlin... Configuration_adv.h Marlin

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug_list.h sensitive_pins.h sd MarlinCore.cpp MarlinCore.h _Bootscreen.h Statusscreen.h Configuration.h Configuration_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

Configuration.h

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

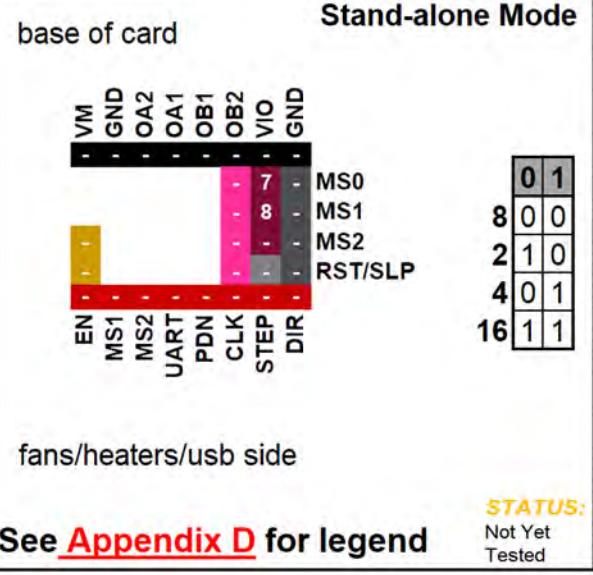
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET_3D	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

OUTLINE TIMELINE

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

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

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

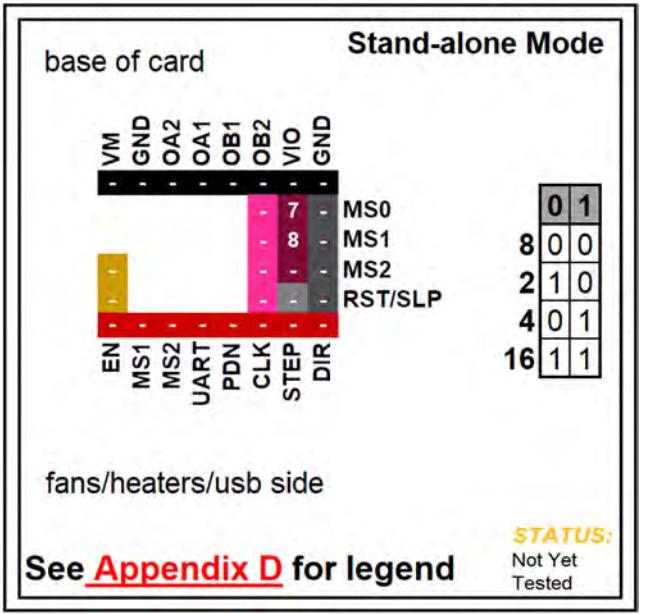
NOTE: Use the potentiometer (POT) on the top of the board (or use the board's " V_{ref} Test point" to set your V_{ref} . See [Appendix A](#) for instructions on how to set the V_{ref} on a driver board.



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

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

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

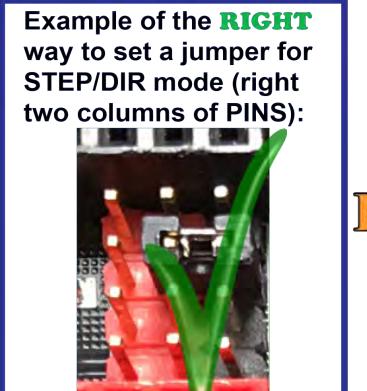
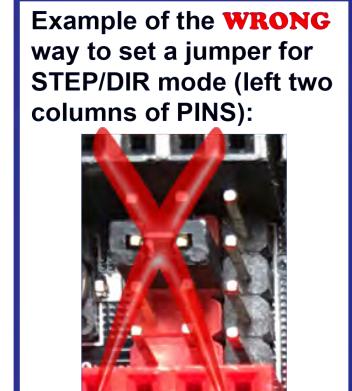
**Driver Chip Chart:**

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

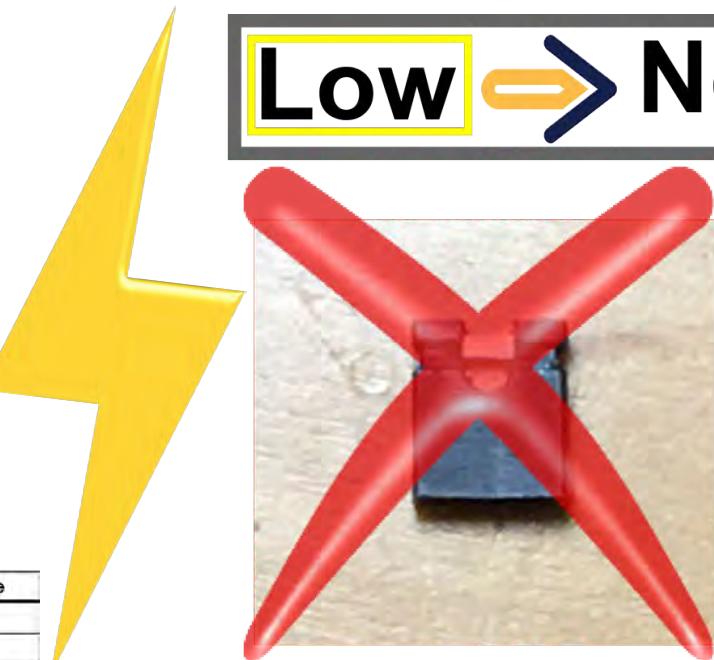
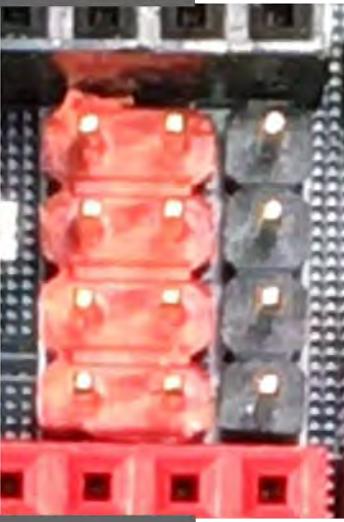
Driving Current Calculation Formula

$I_{MAX} = \text{FORMULA}$ $V_{ref} = \text{FORMULA}$

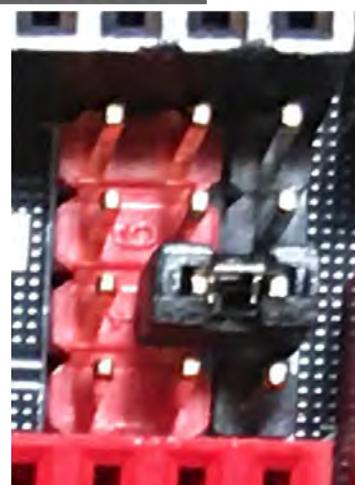
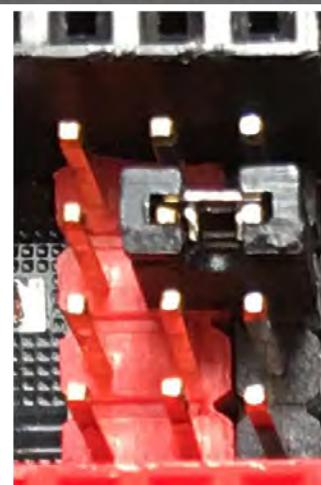
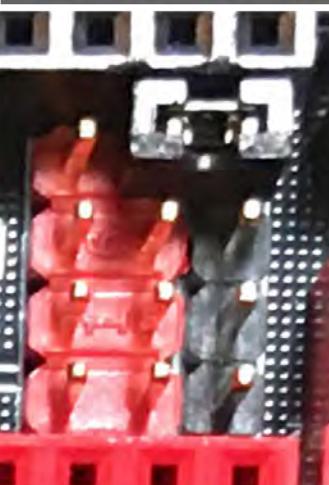
$R_S(\text{typical Sense Resistor}) = X.XX \Omega$

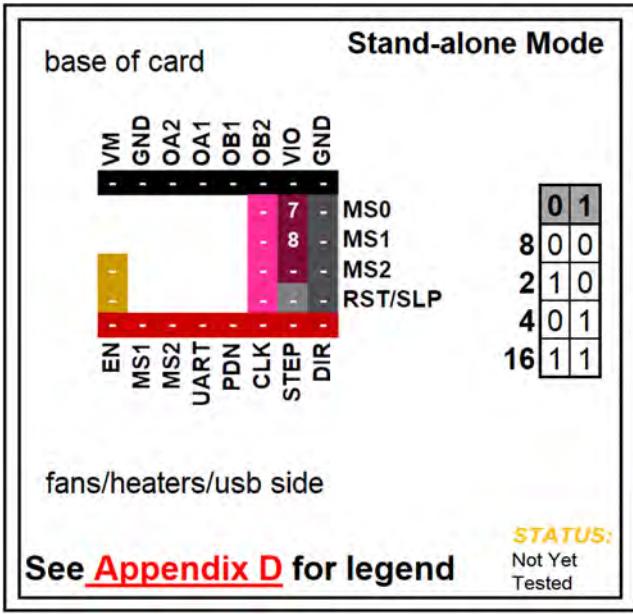
**BIQU TMC2208 V3.0****Stand-alone Mode****SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Low → **No Jumper**

**Rows:****MS0****MS1****MS2****RST/SLP**

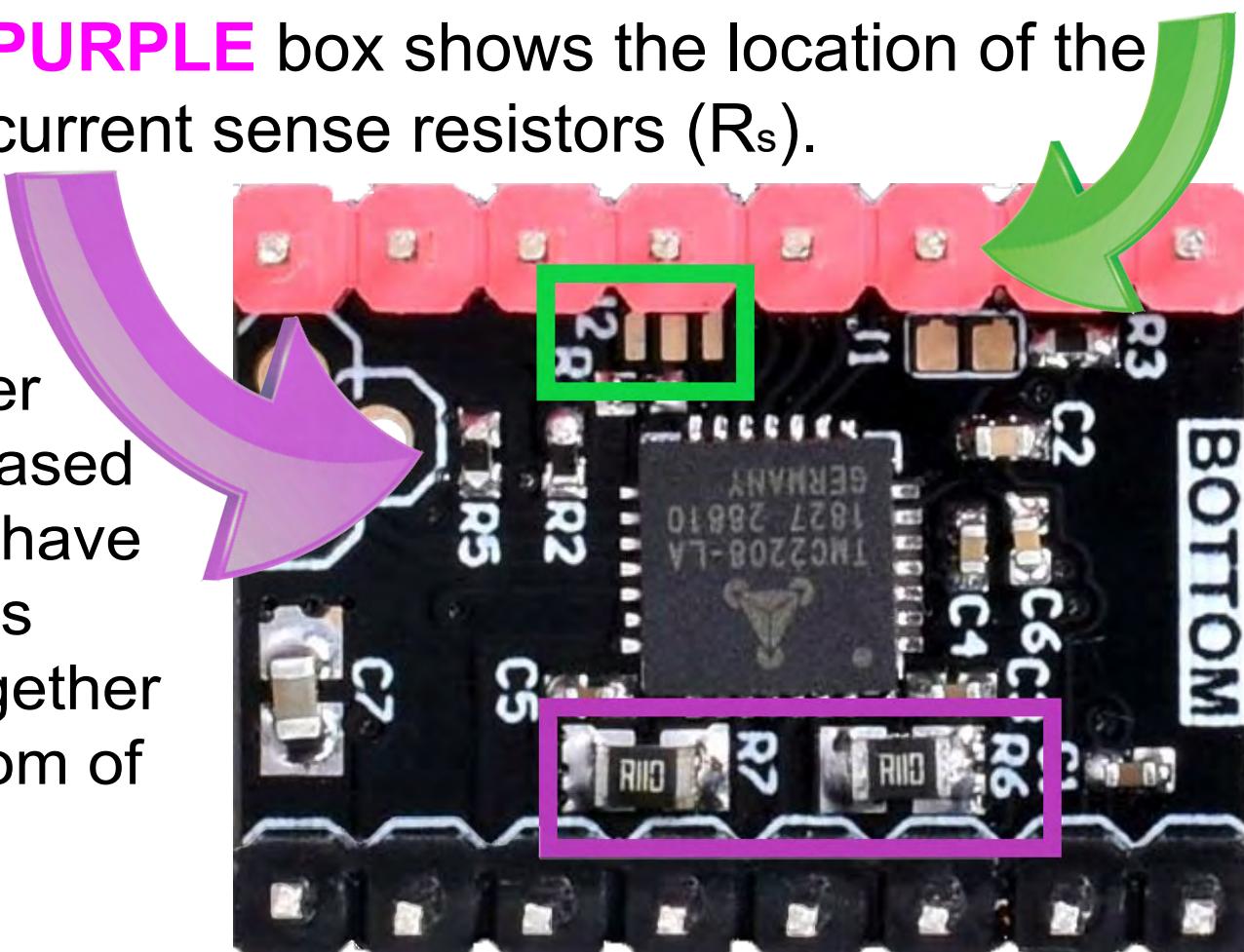
High → **Jumper Set**

Rows:**MS0****MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

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

Note: The three pads (J2), on the bottom of the BIQU TMC2208 V3.0 driver boards, **MUST NOT** be connected. Again, a gap MUST be in place between all three J2 pads to obtain stand-alone mode for the TMC2208 V3.0, as seen in the **GREEN** box below. The **PURPLE** box shows the location of the current sense resistors (R_s).

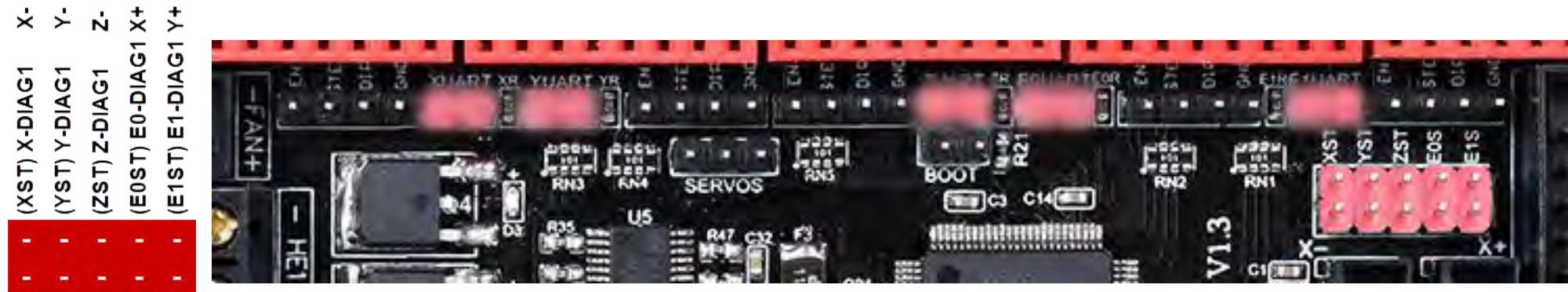
Note: MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).



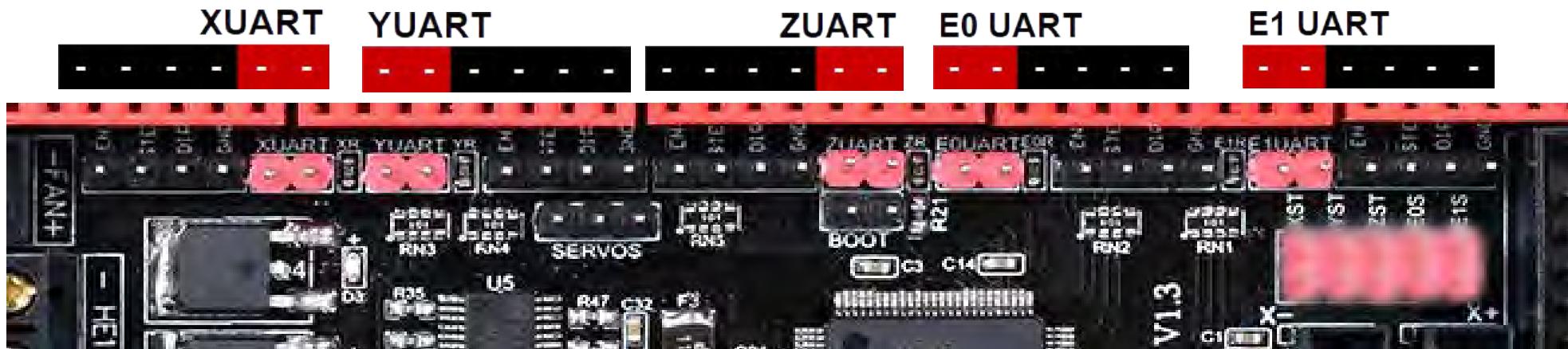
Stand-alone Mode

Stand-alone Mode

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



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



Stand-alone Mode

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

Driver Socket Representation

Columns: **1 2 3**Rows:
MS0**MS1****MS2****RST/SLP****E****1/2**

Driver Chip Chart:

Driver Chip	MS0	MS1	MS2	Excitation Mode
LOGO	Low	Low	Low	1-1 Phase
Chip Name	High	Low	Low	W1-2 Phase
Maximum XXX	Low	High	Low	2W1-2 Phase
Subdivision	Low	Low	High	4W1-2 Phase
XXV DC	High	Low	High	SW1-2 Phase
XXA (peak)	Low	High	High	16W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} = \text{FORMULA}$		$V_{ref} = \text{FORMULA}$	
R_g (typical) Series Resistor (in Ω)				

MS0 for Binary State Drivers:

Driver Socket Representation:

7 7

Driver Chip Chart:

High

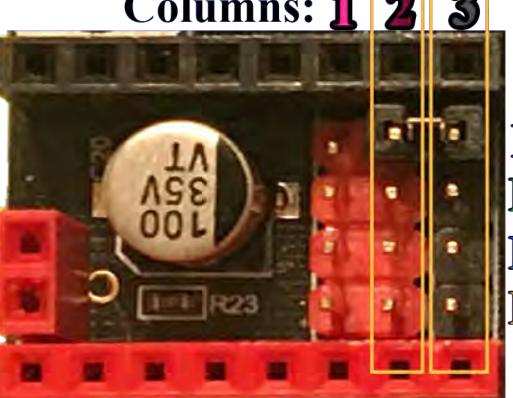
Driver Socket Representation:

--

Driver Chip Chart:

Low

High:

Columns: **1 2 3**Rows:
MS0**MS1****MS2****RST/SLP**

Meaning:

Driver Chip Chart:

High

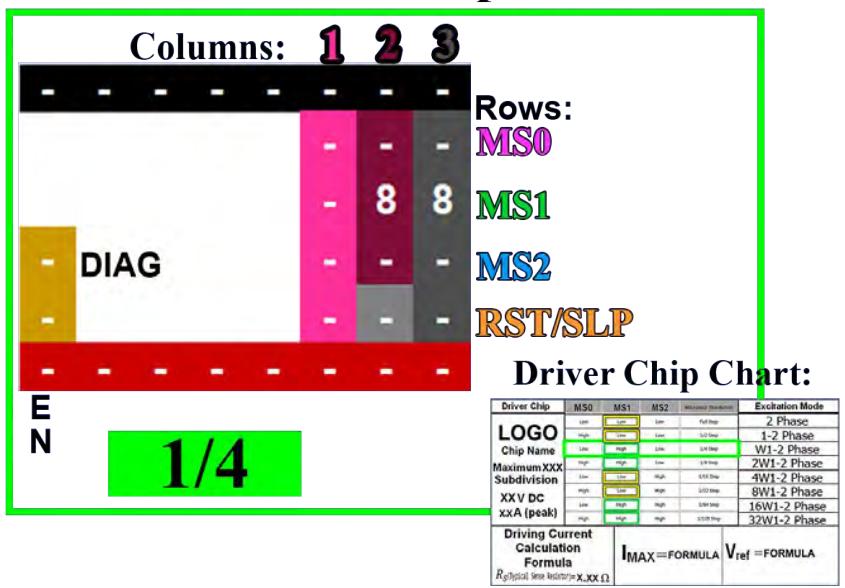
set Jumper between column 2 and column 3 on the MS0 row

Low

No Jumper Set

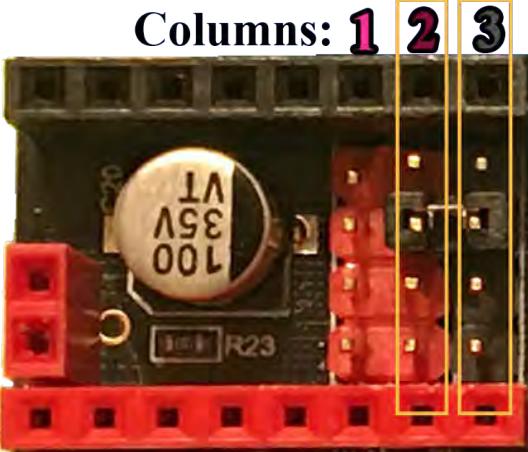
Stand-alone Mode

Driver Socket Representation



High:

Columns: 1 2 3



Rows:
MS0
MS1
MS2
RST/SLP

MS1 for Binary State Drivers:

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

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

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

Meaning:

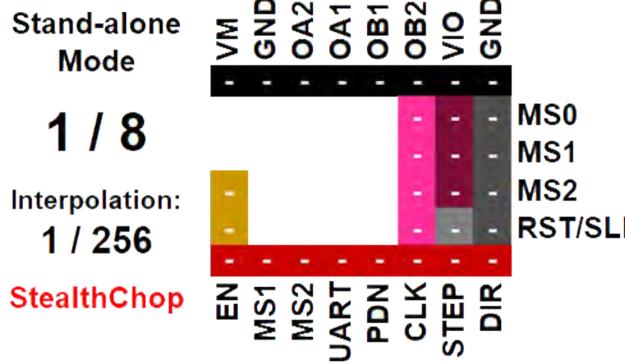
Driver Chip Chart:

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

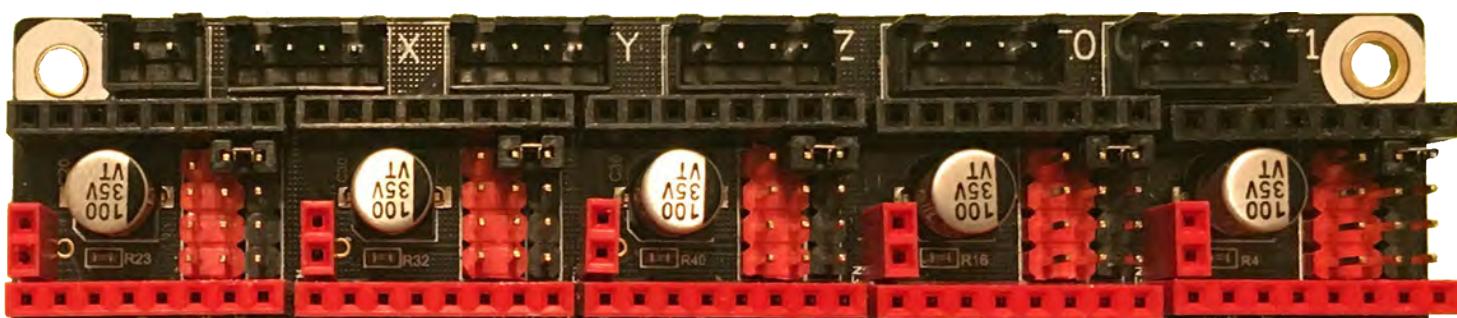
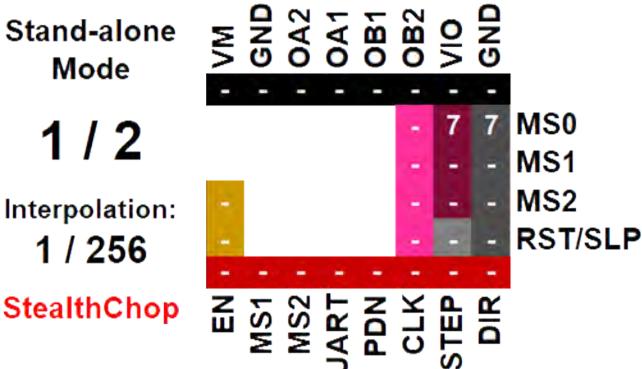
Low → No Jumper Set

Stand-alone Mode

Stand-alone Mode



See [Appendix D](#) for legend



See [Appendix D](#) for legend

Stand-alone Mode

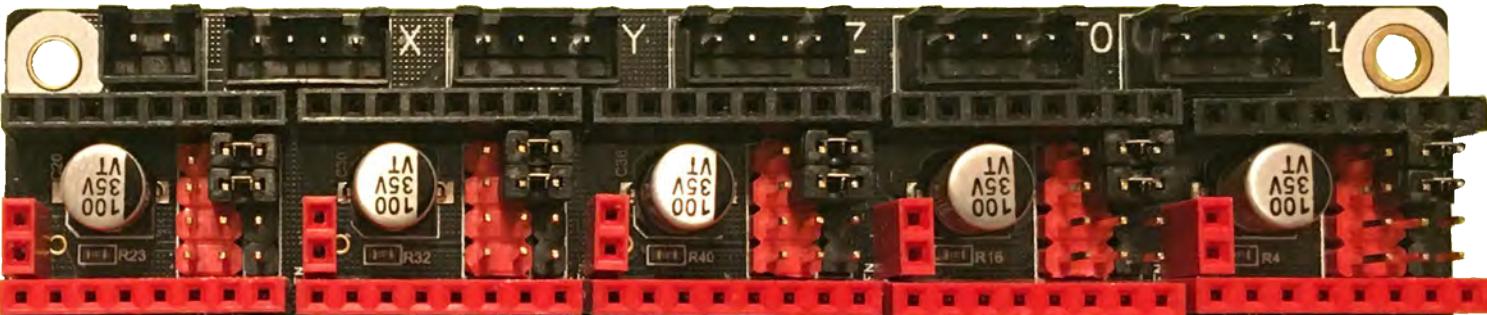
Stand-alone Mode

Stand-alone Mode	VM	GND	OA2	OA1	OB1	OB2	VIO	GND
1 / 4	-	-	-	-	-	-	8	8
Interpolation:	-	-	-	-	-	-	-	-
1 / 256	-	-	-	-	-	-	-	-
StealthChop	EN	MS1	MS2	UART	PDN	CLK	STEP	DIR



See [Appendix D](#) for legend

Stand-alone Mode	VM	GND	OA2	OA1	OB1	OB2	VIO	GND
1 / 16	-	-	-	-	-	-	7	7
Interpolation:	-	-	-	-	-	-	-	-
1 / 256	-	-	-	-	-	-	-	-
StealthChop	EN	MS1	MS2	UART	PDN	CLK	STEP	DIR

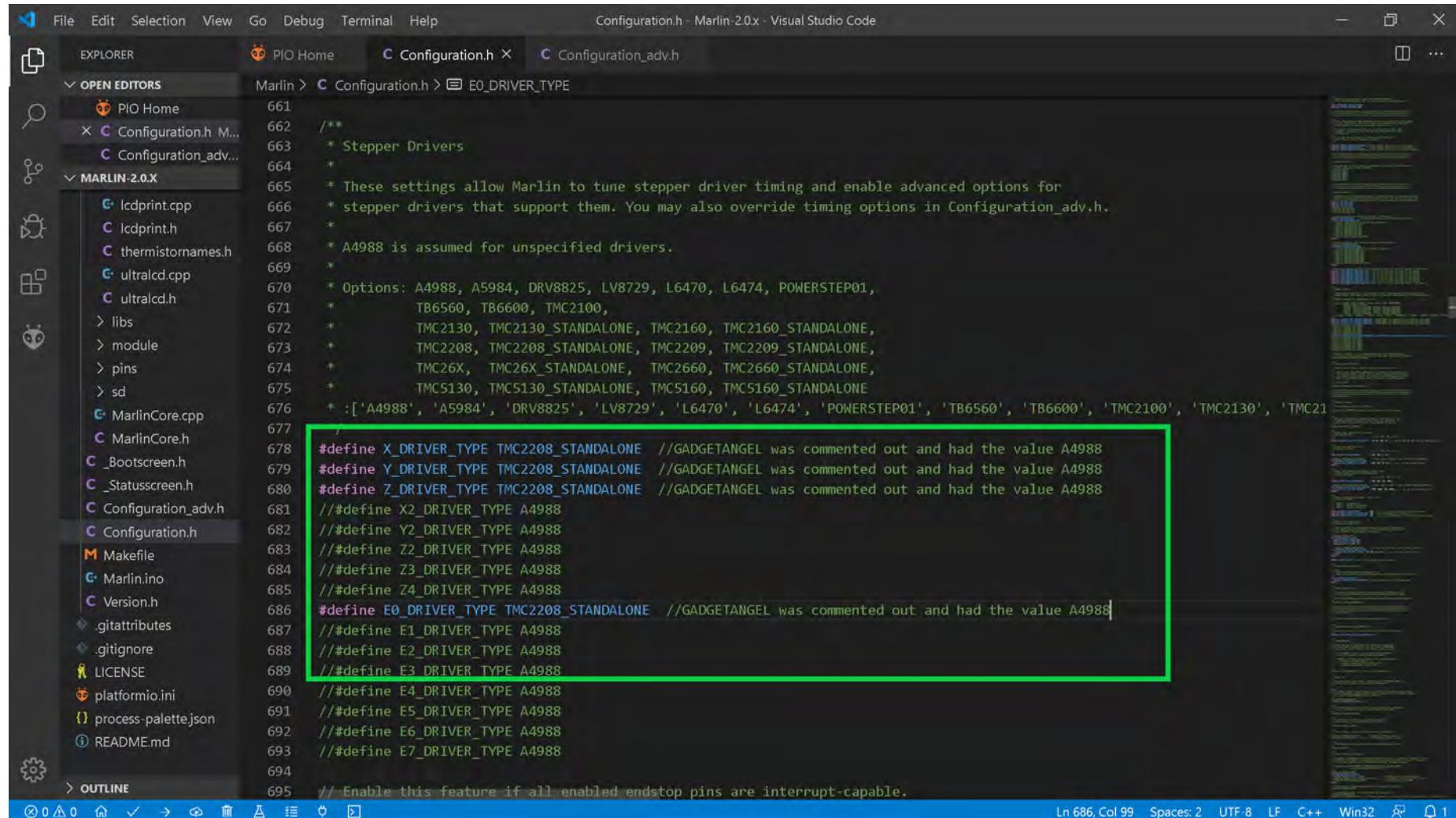


See [Appendix D](#) for legend

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

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2208 stepper motor drivers in stand-alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand-alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h X Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE

661 /**
662 * Stepper Drivers
663 *
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC2660']
677 */
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand-alone Mode

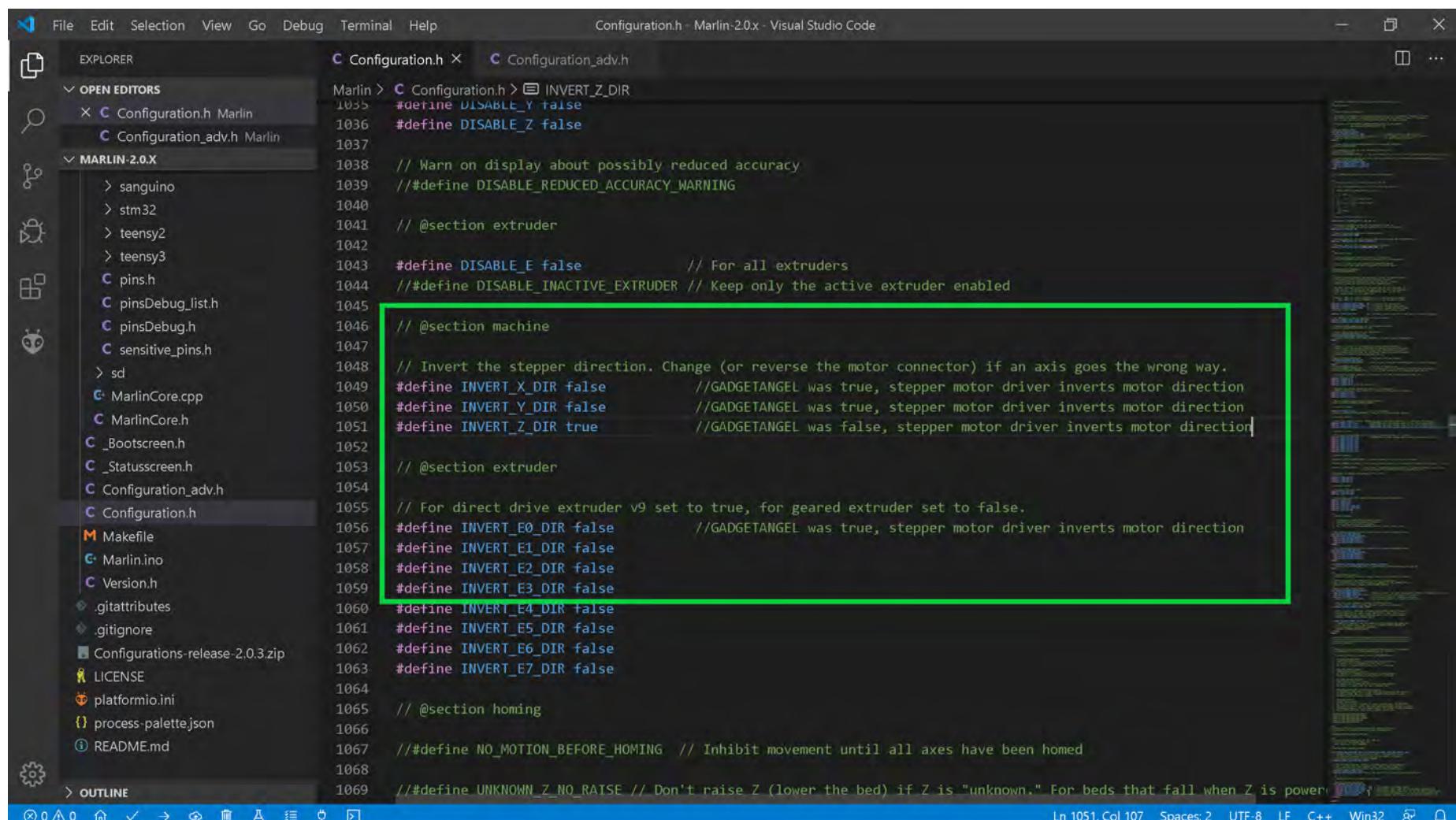
- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h DEFAULT_AXIS_STEPS_PER_UNIT
sam
sanguino
stm32
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
Configurations-release-2.0.3.zip
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 738, Col 62 Spaces: 2 UTF-8 LF C++ Win32
20 20 100 23.25
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 //#define DISTINCT_E_FACTORS
 732 /**
 733 * Default Axis Steps Per Unit (steps/mm)
 734 * Override with M92
 735 *
 736 X, Y, Z, E0 [, E1[, E2...]]
 737 */
 738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
 739 // {80, 80, 400, 93} for A4988 on Ender 3
 740 // want 1/4 steps, so divide each number by 4 since going from
 741 // 1/16 to 1/4 steps
 742 /**
 743 * Default Max Feed Rate (mm/s)
 744 * Override with M203
 745 *
 746 X, Y, Z, E0 [, E1[, E2...]]
 747 */
 748 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
 749
 750 //#define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
 751 #if ENABLED(LIMITED_MAX_FR_EDITING)
 752 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
 753 #endif
 754 /**
 755 * Default Max Acceleration (change/s) change = mm/s
 756 * (Maximum start speed for accelerated moves)
 757 * Override with M201
 758 */
 759
 760 /**
 761 * Default Max Retract Speed (mm/s)
 762 * (Maximum speed for retract moves)
 763 * Override with M202
 764 */
 765
 766 /**
 767 * Default Max Travel Speed (mm/s)
 768 * (Maximum speed for travel moves)
 769 * Override with M203
 770 */
 771
 772 /**
 773 * Default Max Feedrate for G1 (mm/s)
 774 * (Maximum feedrate for G1 moves)
 775 * Override with M203
 776 */
 777
 778 /**
 779 * Default Max Acceleration for G1 (change/s)
 780 * (Maximum acceleration for G1 moves)
 781 * Override with M201
 782 */
 783
 784 /**
 785 * Default Max Retract Speed for G1 (mm/s)
 786 * (Maximum retract speed for G1 moves)
 787 * Override with M202
 788 */
 789
 790 /**
 791 * Default Max Travel Speed for G1 (mm/s)
 792 * (Maximum travel speed for G1 moves)
 793 * Override with M203
 794 */
 795
 796 /**
 797 * Default Max Acceleration for G2/G3 (change/s)
 798 * (Maximum acceleration for G2/G3 moves)
 799 * Override with M201
 800 */
 801
 802 /**
 803 * Default Max Retract Speed for G2/G3 (mm/s)
 804 * (Maximum retract speed for G2/G3 moves)
 805 * Override with M202
 806 */
 807
 808 /**
 809 * Default Max Travel Speed for G2/G3 (mm/s)
 810 * (Maximum travel speed for G2/G3 moves)
 811 * Override with M203
 812 */
 813
 814 /**
 815 * Default Max Feedrate for G2/G3 (mm/s)
 816 * (Maximum feedrate for G2/G3 moves)
 817 * Override with M203
 818 */
 819
 820 /**
 821 * Default Max Acceleration for G4 (change/s)
 822 * (Maximum acceleration for G4 moves)
 823 * Override with M201
 824 */
 825
 826 /**
 827 * Default Max Retract Speed for G4 (mm/s)
 828 * (Maximum retract speed for G4 moves)
 829 * Override with M202
 830 */
 831
 832 /**
 833 * Default Max Travel Speed for G4 (mm/s)
 834 * (Maximum travel speed for G4 moves)
 835 * Override with M203
 836 */
 837
 838 /**
 839 * Default Max Feedrate for G4 (mm/s)
 840 * (Maximum feedrate for G4 moves)
 841 * Override with M203
 842 */
 843
 844 /**
 845 * Default Max Acceleration for G5 (change/s)
 846 * (Maximum acceleration for G5 moves)
 847 * Override with M201
 848 */
 849
 850 /**
 851 * Default Max Retract Speed for G5 (mm/s)
 852 * (Maximum retract speed for G5 moves)
 853 * Override with M202
 854 */
 855
 856 /**
 857 * Default Max Travel Speed for G5 (mm/s)
 858 * (Maximum travel speed for G5 moves)
 859 * Override with M203
 860 */
 861
 862 /**
 863 * Default Max Feedrate for G5 (mm/s)
 864 * (Maximum feedrate for G5 moves)
 865 * Override with M203
 866 */
 867
 868 /**
 869 * Default Max Acceleration for G6 (change/s)
 870 * (Maximum acceleration for G6 moves)
 871 * Override with M201
 872 */
 873
 874 /**
 875 * Default Max Retract Speed for G6 (mm/s)
 876 * (Maximum retract speed for G6 moves)
 877 * Override with M202
 878 */
 879
 880 /**
 881 * Default Max Travel Speed for G6 (mm/s)
 882 * (Maximum travel speed for G6 moves)
 883 * Override with M203
 884 */
 885
 886 /**
 887 * Default Max Feedrate for G6 (mm/s)
 888 * (Maximum feedrate for G6 moves)
 889 * Override with M203
 890 */
 891
 892 /**
 893 * Default Max Acceleration for G7 (change/s)
 894 * (Maximum acceleration for G7 moves)
 895 * Override with M201
 896 */
 897
 898 /**
 899 * Default Max Retract Speed for G7 (mm/s)
 900 * (Maximum retract speed for G7 moves)
 901 * Override with M202
 902 */
 903
 904 /**
 905 * Default Max Travel Speed for G7 (mm/s)
 906 * (Maximum travel speed for G7 moves)
 907 * Override with M203
 908 */
 909
 910 /**
 911 * Default Max Feedrate for G7 (mm/s)
 912 * (Maximum feedrate for G7 moves)
 913 * Override with M203
 914 */
 915
 916 /**
 917 * Default Max Acceleration for G8 (change/s)
 918 * (Maximum acceleration for G8 moves)
 919 * Override with M201
 920 */
 921
 922 /**
 923 * Default Max Retract Speed for G8 (mm/s)
 924 * (Maximum retract speed for G8 moves)
 925 * Override with M202
 926 */
 927
 928 /**
 929 * Default Max Travel Speed for G8 (mm/s)
 930 * (Maximum travel speed for G8 moves)
 931 * Override with M203
 932 */
 933
 934 /**
 935 * Default Max Feedrate for G8 (mm/s)
 936 * (Maximum feedrate for G8 moves)
 937 * Override with M203
 938 */
 939
 940 /**
 941 * Default Max Acceleration for G9 (change/s)
 942 * (Maximum acceleration for G9 moves)
 943 * Override with M201
 944 */
 945
 946 /**
 947 * Default Max Retract Speed for G9 (mm/s)
 948 * (Maximum retract speed for G9 moves)
 949 * Override with M202
 950 */
 951
 952 /**
 953 * Default Max Travel Speed for G9 (mm/s)
 954 * (Maximum travel speed for G9 moves)
 955 * Override with M203
 956 */
 957
 958 /**
 959 * Default Max Feedrate for G9 (mm/s)
 960 * (Maximum feedrate for G9 moves)
 961 * Override with M203
 962 */
 963
 964 /**
 965 * Default Max Acceleration for G10 (change/s)
 966 * (Maximum acceleration for G10 moves)
 967 * Override with M201
 968 */
 969
 970 /**
 971 * Default Max Retract Speed for G10 (mm/s)
 972 * (Maximum retract speed for G10 moves)
 973 * Override with M202
 974 */
 975
 976 /**
 977 * Default Max Travel Speed for G10 (mm/s)
 978 * (Maximum travel speed for G10 moves)
 979 * Override with M203
 980 */
 981
 982 /**
 983 * Default Max Feedrate for G10 (mm/s)
 984 * (Maximum feedrate for G10 moves)
 985 * Override with M203
 986 */
 987
 988 /**
 989 * Default Max Acceleration for G11 (change/s)
 990 * (Maximum acceleration for G11 moves)
 991 * Override with M201
 992 */
 993
 994 /**
 995 * Default Max Retract Speed for G11 (mm/s)
 996 * (Maximum retract speed for G11 moves)
 997 * Override with M202
 998 */
 999
 1000 /**
 1001 * Default Max Travel Speed for G11 (mm/s)
 1002 * (Maximum travel speed for G11 moves)
 1003 * Override with M203
 1004 */
 1005
 1006 /**
 1007 * Default Max Feedrate for G11 (mm/s)
 1008 * (Maximum feedrate for G11 moves)
 1009 * Override with M203
 1010 */
 1011
 1012 /**
 1013 * Default Max Acceleration for G12 (change/s)
 1014 * (Maximum acceleration for G12 moves)
 1015 * Override with M201
 1016 */
 1017
 1018 /**
 1019 * Default Max Retract Speed for G12 (mm/s)
 1020 * (Maximum retract speed for G12 moves)
 1021 * Override with M202
 1022 */
 1023
 1024 /**
 1025 * Default Max Travel Speed for G12 (mm/s)
 1026 * (Maximum travel speed for G12 moves)
 1027 * Override with M203
 1028 */
 1029
 1030 /**
 1031 * Default Max Feedrate for G12 (mm/s)
 1032 * (Maximum feedrate for G12 moves)
 1033 * Override with M203
 1034 */
 1035
 1036 /**
 1037 * Default Max Acceleration for G13 (change/s)
 1038 * (Maximum acceleration for G13 moves)
 1039 * Override with M201
 1040 */
 1041
 1042 /**
 1043 * Default Max Retract Speed for G13 (mm/s)
 1044 * (Maximum retract speed for G13 moves)
 1045 * Override with M202
 1046 */
 1047
 1048 /**
 1049 * Default Max Travel Speed for G13 (mm/s)
 1050 * (Maximum travel speed for G13 moves)
 1051 * Override with M203
 1052 */
 1053
 1054 /**
 1055 * Default Max Feedrate for G13 (mm/s)
 1056 * (Maximum feedrate for G13 moves)
 1057 * Override with M203
 1058 */
 1059
 1060 /**
 1061 * Default Max Acceleration for G14 (change/s)
 1062 * (Maximum acceleration for G14 moves)
 1063 * Override with M201
 1064 */
 1065
 1066 /**
 1067 * Default Max Retract Speed for G14 (mm/s)
 1068 * (Maximum retract speed for G14 moves)
 1069 * Override with M202
 1070 */
 1071
 1072 /**
 1073 * Default Max Travel Speed for G14 (mm/s)
 1074 * (Maximum travel speed for G14 moves)
 1075 * Override with M203
 1076 */
 1077
 1078 /**
 1079 * Default Max Feedrate for G14 (mm/s)
 1080 * (Maximum feedrate for G14 moves)
 1081 * Override with M203
 1082 */
 1083
 1084 /**
 1085 * Default Max Acceleration for G15 (change/s)
 1086 * (Maximum acceleration for G15 moves)
 1087 * Override with M201
 1088 */
 1089
 1090 /**
 1091 * Default Max Retract Speed for G15 (mm/s)
 1092 * (Maximum retract speed for G15 moves)
 1093 * Override with M202
 1094 */
 1095
 1096 /**
 1097 * Default Max Travel Speed for G15 (mm/s)
 1098 * (Maximum travel speed for G15 moves)
 1099 * Override with M203
 1100 */
 1101
 1102 /**
 1103 * Default Max Feedrate for G15 (mm/s)
 1104 * (Maximum feedrate for G15 moves)
 1105 * Override with M203
 1106 */
 1107
 1108 /**
 1109 * Default Max Acceleration for G16 (change/s)
 1110 * (Maximum acceleration for G16 moves)
 1111 * Override with M201
 1112 */
 1113
 1114 /**
 1115 * Default Max Retract Speed for G16 (mm/s)
 1116 * (Maximum retract speed for G16 moves)
 1117 * Override with M202
 1118 */
 1119
 1120 /**
 1121 * Default Max Travel Speed for G16 (mm/s)
 1122 * (Maximum travel speed for G16 moves)
 1123 * Override with M203
 1124 */
 1125
 1126 /**
 1127 * Default Max Feedrate for G16 (mm/s)
 1128 * (Maximum feedrate for G16 moves)
 1129 * Override with M203
 1130 */
 1131
 1132 /**
 1133 * Default Max Acceleration for G17 (change/s)
 1134 * (Maximum acceleration for G17 moves)
 1135 * Override with M201
 1136 */
 1137
 1138 /**
 1139 * Default Max Retract Speed for G17 (mm/s)
 1140 * (Maximum retract speed for G17 moves)
 1141 * Override with M202
 1142 */
 1143
 1144 /**
 1145 * Default Max Travel Speed for G17 (mm/s)
 1146 * (Maximum travel speed for G17 moves)
 1147 * Override with M203
 1148 */
 1149
 1150 /**
 1151 * Default Max Feedrate for G17 (mm/s)
 1152 * (Maximum feedrate for G17 moves)
 1153 * Override with M203
 1154 */
 1155
 1156 /**
 1157 * Default Max Acceleration for G18 (change/s)
 1158 * (Maximum acceleration for G18 moves)
 1159 * Override with M201
 1160 */
 1161
 1162 /**
 1163 * Default Max Retract Speed for G18 (mm/s)
 1164 * (Maximum retract speed for G18 moves)
 1165 * Override with M202
 1166 */
 1167
 1168 /**
 1169 * Default Max Travel Speed for G18 (mm/s)
 1170 * (Maximum travel speed for G18 moves)
 1171 * Override with M203
 1172 */
 1173
 1174 /**
 1175 * Default Max Feedrate for G18 (mm/s)
 1176 * (Maximum feedrate for G18 moves)
 1177 * Override with M203
 1178 */
 1179
 1180 /**
 1181 * Default Max Acceleration for G19 (change/s)
 1182 * (Maximum acceleration for G19 moves)
 1183 * Override with M201
 1184 */
 1185
 1186 /**
 1187 * Default Max Retract Speed for G19 (mm/s)
 1188 * (Maximum retract speed for G19 moves)
 1189 * Override with M202
 1190 */
 1191
 1192 /**
 1193 * Default Max Travel Speed for G19 (mm/s)
 1194 * (Maximum travel speed for G19 moves)
 1195 * Override with M203
 1196 */
 1197
 1198 /**
 1199 * Default Max Feedrate for G19 (mm/s)
 1200 * (Maximum feedrate for G19 moves)
 1201 * Override with M203
 1202 */
 1203
 1204 /**
 1205 * Default Max Acceleration for G20 (change/s)
 1206 * (Maximum acceleration for G20 moves)
 1207 * Override with M201
 1208 */
 1209
 1210 /**
 1211 * Default Max Retract Speed for G20 (mm/s)
 1212 * (Maximum retract speed for G20 moves)
 1213 * Override with M202
 1214 */
 1215
 1216 /**
 1217 * Default Max Travel Speed for G20 (mm/s)
 1218 * (Maximum travel speed for G20 moves)
 1219 * Override with M203
 1220 */
 1221
 1222 /**
 1223 * Default Max Feedrate for G20 (mm/s)
 1224 * (Maximum feedrate for G20 moves)
 1225 * Override with M203
 1226 */
 1227
 1228 /**
 1229 * Default Max Acceleration for G21 (change/s)
 1230 * (Maximum acceleration for G21 moves)
 1231 * Override with M201
 1232 */
 1233
 1234 /**
 1235 * Default Max Retract Speed for G21 (mm/s)
 1236 * (Maximum retract speed for G21 moves)
 1237 * Override with M202
 1238 */
 1239
 1240 /**
 1241 * Default Max Travel Speed for G21 (mm/s)
 1242 * (Maximum travel speed for G21 moves)
 1243 * Override with M203
 1244 */
 1245
 1246 /**
 1247 * Default Max Feedrate for G21 (mm/s)
 1248 * (Maximum feedrate for G21 moves)
 1249 * Override with M203
 1250 */
 1251
 1252 /**
 1253 * Default Max Acceleration for G22 (change/s)
 1254 * (Maximum acceleration for G22 moves)
 1255 * Override with M201
 1256 */
 1257
 1258 /**
 1259 * Default Max Retract Speed for G22 (mm/s)
 1260 * (Maximum retract speed for G22 moves)
 1261 * Override with M202
 1262 */
 1263
 1264 /**
 1265 * Default Max Travel Speed for G22 (mm/s)
 1266 * (Maximum travel speed for G22 moves)
 1267 * Override with M203
 1268 */
 1269
 1270 /**
 1271 * Default Max Feedrate for G22 (mm/s)
 1272 * (Maximum feedrate for G22 moves)
 1273 * Override with M203
 1274 */
 1275
 1276 /**
 1277 * Default Max Acceleration for G23 (change/s)
 1278 * (Maximum acceleration for G23 moves)
 1279 * Override with M201
 1280 */
 1281
 1282 /**
 1283 * Default Max Retract Speed for G23 (mm/s)
 1284 * (Maximum retract speed for G23 moves)
 1285 * Override with M202
 1286 */
 1287
 1288 /**
 1289 * Default Max Travel Speed for G23 (mm/s)
 1290 * (Maximum travel speed for G23 moves)
 1291 * Override with M203
 1292 */
 1293
 1294 /**
 1295 * Default Max Feedrate for G23 (mm/s)
 1296 * (Maximum feedrate for G23 moves)
 1297 * Override with M203
 1298 */
 1299
 1300 /**
 1301 * Default Max Acceleration for G24 (change/s)
 1302 * (Maximum acceleration for G24 moves)
 1303 * Override with M201
 1304 */
 1305
 1306 /**
 1307 * Default Max Retract Speed for G24 (mm/s)
 1308 * (Maximum retract speed for G24 moves)
 1309 * Override with M202
 1310 */
 1311
 1312 /**
 1313 * Default Max Travel Speed for G24 (mm/s)
 1314 * (Maximum travel speed for G24 moves)
 1315 * Override with M203
 1316 */
 1317
 1318 /**
 1319 * Default Max Feedrate for G24 (mm/s)
 1320 * (Maximum feedrate for G24 moves)
 1321 * Override with M203
 1322 */
 1323
 1324 /**
 1325 * Default Max Acceleration for G25 (change/s)
 1326 * (Maximum acceleration for G25 moves)
 1327 * Override with M201
 1328 */
 1329
 1330 /**
 1331 * Default Max Retract Speed for G25 (mm/s)
 1332 * (Maximum retract speed for G25 moves)
 1333 * Override with M202
 1334 */
 1335
 1336 /**
 1337 * Default Max Travel Speed for G25 (mm/s)
 1338 * (Maximum travel speed for G25 moves)
 1339 * Override with M203
 1340 */
 1341
 1342 /**
 1343 * Default Max Feedrate for G25 (mm/s)
 1344 * (Maximum feedrate for G25 moves)
 1345 * Override with M203
 1346 */
 1347
 1348 /**
 1349 * Default Max Acceleration for G26 (change/s)
 1350 * (Maximum acceleration for G26 moves)
 1351 * Override with M201
 1352 */
 1353
 1354 /**
 1355 * Default Max Retract Speed for G26 (mm/s)
 1356 * (Maximum retract speed for G26 moves)
 1357 * Override with M202
 1358 */
 1359
 1360 /**
 1361 * Default Max Travel Speed for G26 (mm/s)
 1362 * (Maximum travel speed for G26 moves)
 1363 * Override with M203
 1364 */
 1365
 1366 /**
 1367 * Default Max Feedrate for G26 (mm/s)
 1368 * (Maximum feedrate for G26 moves)
 1369 * Override with M203
 1370 */
 1371
 1372 /**
 1373 * Default Max Acceleration for G27 (change/s)
 1374 * (Maximum acceleration for G27 moves)
 1375 * Override with M201
 1376 */
 1377
 1378 /**
 1379 * Default Max Retract Speed for G27 (mm/s)
 1380 * (Maximum retract speed for G27 moves)
 1381 * Override with M202
 1382 */
 1383
 1384 /**
 1385 * Default Max Travel Speed for G27 (mm/s)
 1386 * (Maximum travel speed for G27 moves)
 1387 * Override with M203
 1388 */
 1389
 1390 /**
 1391 * Default Max Feedrate for G27 (mm/s)
 1392 * (Maximum feedrate for G27 moves)
 1393 * Override with M203
 1394 */
 1395
 1396 /**
 1397 * Default Max Acceleration for G28 (change/s)
 1398 * (Maximum acceleration for G28 moves)
 1399 * Override with M201
 1400 */
 1401
 1402 /**
 1403 * Default Max Retract Speed for G28 (mm/s)
 1404 * (Maximum retract speed for G28 moves)
 1405 * Override with M202
 1406 */
 1407
 1408 /**
 1409 * Default Max Travel Speed for G28 (mm/s)
 1410 * (Maximum travel speed for G28 moves)
 1411 * Override with M203
 1412 */
 1413
 1414 /**
 1415 * Default Max Feedrate for G28 (mm/s)
 1416 * (Maximum feedrate for G28 moves)
 1417 * Override with M203
 1418 */
 1419
 1420 /**
 1421 * Default Max Acceleration for G29 (change/s)
 1422 * (Maximum acceleration for G29 moves)
 1423 * Override with M201
 1424 */
 1425
 1426 /**
 1427 * Default Max Retract Speed for G29 (mm/s)
 1428 * (Maximum retract speed for G29 moves)
 1429 * Override with M202
 1430 */
 1431
 1432 /**
 1433 * Default Max Travel Speed for G29 (mm/s)
 1434 * (Maximum travel speed for G29 moves)
 1435 * Override with M203
 1436 */
 1437
 1438 /**
 1439 * Default Max Feedrate for G29 (mm/s)
 1440 * (Maximum feedrate for G29 moves)
 1441 * Override with M203
 1442 */
 1443
 1444 /**
 1445 * Default Max Acceleration for G30 (change/s)
 1446 * (Maximum acceleration for G30 moves)
 1447 * Override with M201
 1448 */
 1449
 1450 /**
 1451 * Default Max Retract Speed for G30 (mm/s)
 1452 * (Maximum retract speed for G30 moves)
 1453 * Override with M202
 1454 */
 1455
 1456 /**
 1457 * Default Max Travel Speed for G30 (mm/s)
 1458 * (Maximum travel speed for G30 moves)
 1459 * Override with M203
 1460 */
 1461
 1462 /**
 1463 * Default Max Feedrate for G30 (mm/s)
 1464 * (Maximum feedrate for G30 moves)
 1465 * Override with M203
 1466 */
 1467
 1468 /**
 1469 * Default Max Acceleration for G31 (change/s)
 1470 * (Maximum acceleration for G31 moves)
 1471 * Override with M201
 1472 */
 1473
 1474 /**
 1475 * Default Max Retract Speed for G31 (mm/s)
 1476 * (Maximum retract speed for G31 moves)
 1477 * Override with M202
 1478 */
 1479
 1480 /**
 1481 * Default Max Travel Speed for G31 (mm/s)
 1482 * (Maximum travel speed for G31 moves)
 1483 * Override with M203
 1484 */
 1485
 1486 /**
 1487 * Default Max Feedrate for G31 (mm/s)
 1488 * (Maximum feedrate for G31 moves)
 1489 * Override with M203
 1490 */
 1491
 1492 /**
 1493 * Default Max Acceleration for G32 (change/s)
 1494 * (Maximum acceleration for G32 moves)
 1495 * Override with M201
 1496 */
 1497
 1498 /**
 1499 * Default Max Retract Speed for G32 (mm/s)
 1500 * (Maximum retract speed for G32 moves)
 1501 * Override with M202
 1502 */
 1503
 1504 /**
 1505 * Default Max Travel Speed for G32 (mm/s)
 1506 * (Maximum travel speed for G32 moves)
 1507 * Override with M203
 1508 */
 1509
 1510 /**
 1511 * Default Max Feedrate for G32 (mm/s)
 1512 * (Maximum feedrate for G32 moves)
 1513 * Override with M203
 1514 */
 1515
 1516 /**
 1517 * Default Max Acceleration for G33 (change/s)
 1518 * (Maximum acceleration for G33 moves)
 1519 * Override with M201
 1520 */
 1521
 1522 /**
 1523 * Default Max Retract Speed for G33 (mm/s)
 1524 * (Maximum retract speed for G33 moves)
 1525 * Override with M202
 1526 */
 1527
 1528 /**
 1529 * Default Max Travel Speed for G33 (mm/s)
 1530 * (Maximum travel speed for G33 moves)
 1531 * Override with M203
 1532 */
 1533
 1534 /**
 1535 * Default Max Feedrate for G33 (mm/s)
 1536 * (Maximum feedrate for G33 moves)
 1537 * Override with M203
 1538 */
 1539
 1540 /**
 1541 * Default Max Acceleration for G34 (change/s)
 1542 * (Maximum acceleration for G34 moves)
 1543 * Override with M201
 1544 */
 1545
 1546 /**
 1547 * Default Max Retract Speed for G34 (mm/s)
 1548 * (Maximum retract speed for G34 moves)
 1549 * Override with M202
 1550 */
 1551
 1552 /**
 1553 * Default Max Travel Speed for G34 (mm/s)
 1554 * (Maximum travel speed for G34 moves)
 1555 * Override with M203
 1556 */
 1557
 1558 /**
 1559 * Default Max Feedrate for G34 (mm/s)
 1560 * (Maximum feedrate for G34 moves)
 1561 * Override with M203
 1562 */
 1563
 1564 /**
 1565 * Default Max Acceleration for G35 (change/s)
 1566 * (Maximum acceleration for G35 moves)
 1567 * Override with M201
 1568 */
 1569
 1570 /**
 1571 * Default Max Retract Speed for G35 (mm/s)
 1572 * (Maximum retract speed for G35 moves)
 1573 * Override with M202
 1574 */
 1575
 1576 /**
 1577 * Default Max Travel Speed for G35 (mm/s)
 1578 * (Maximum travel speed for G35 moves)
 1579 * Override with M203
 1580 */
 1581
 1582 /**
 1583 * Default Max Feedrate for G35 (mm/s)
 1584 * (Maximum feedrate for G35 moves)
 1585 * Override with M203
 1586 */
 1587
 1588 /**
 1589 * Default Max Acceleration for G36 (change/s)
 1590 * (Maximum acceleration for G36 moves)
 1591 * Override with M201
 1592 */
 1593
 1594 /**
 1595 * Default Max Retract Speed for G36 (mm/s)
 1596 * (Maximum retract speed for G36 moves)
 1597 * Override with M202
 1598 */
 1599
 1600 /**
 1601 * Default Max Travel Speed for G36 (mm/s)
 1602 * (Maximum travel speed for G36 moves)
 1603 * Override with M203
 1604 */
 1605
 1606 /**
 1607 * Default Max Feedrate for G36 (mm/s)
 1608 * (Maximum feedrate for G36 moves)
 1609 * Override with M203
 1610 */
 1611
 1612 /**
 1613 * Default Max Acceleration for G37 (change/s)
 1614 * (Maximum acceleration for G37 moves)
 1615 * Override with M201
 1616 */
 1617
 1618 /**
 1619 * Default Max Retract Speed for G37 (mm/s)
 1620 * (Maximum retract speed for G37 moves)
 1621 * Override with M202
 1622 */
 1623
 1624 /**
 1625 * Default Max Travel Speed for G37 (mm/s)
 1626 * (Maximum travel speed for G37 moves)
 1627 * Override with M203
 1628 */
 1629
 1630 /**
 1631 * Default Max Feedrate for G37 (mm/s)
 1632 * (Maximum feedrate for G37 moves)
 1633 * Override with M203
 1634 */
 1635
 1636 /**
 1637 * Default Max Acceleration for G38 (change/s)
 1638 * (Maximum acceleration for G38 moves)
 1639 * Override with M201
 1640 */
 1641
 1642 /**
 1643 * Default Max Retract Speed for G38 (mm/s)
 1644 * (Maximum retract speed for G38 moves)
 1645 * Override with M202
 1646 */
 1647
 1648 /**
 1649 * Default Max Travel Speed for G38 (mm/s)
 1650 * (Maximum travel speed for G38 moves)
 1651 * Override with M203
 1652 */
 1653
 1654 /**
 1655 * Default Max Feedrate for G38 (mm/s)
 1656 * (Maximum feedrate for G38 moves)
 1657 * Override with M203
 1658 */
 1659
 1660 /**
 1661 * Default Max Acceleration for G39 (change/s)
 1662 * (Maximum acceleration for G39 moves)
 1663 * Override with M201
 1664 */
 1665
 1666 /**
 1667 * Default Max Retract Speed for G39 (mm/s)
 1668 * (Maximum retract speed for G39 moves)
 1669 * Override with M202
 1670 */
 1671
 1672 /**
 1673 * Default Max Travel Speed for G39 (mm/s)
 1674 * (Maximum travel speed for G39 moves)
 1675 * Override with M203
 1676 */
 1677
 1678 /**
 1679 * Default Max Feedrate for G39 (mm/s)
 1680 * (Maximum feedrate for G39 moves)
 1681 * Override with M203
 1682 */
 1683
 1684 /**
 1685 * Default Max Acceleration for G40 (change/s)
 1686 * (Maximum acceleration for G40 moves)
 1687 * Override with M201
 1688 */
 1689
 1690 /**
 1691 * Default Max Retract Speed for G40 (mm/s)
 1692 * (Maximum retract speed for G40 moves)
 1693 * Override with M202
 1694 */
 1695
 1696 /**
 1697 * Default Max Travel Speed for G40 (mm/s)
 1698 * (Maximum travel speed for G40 moves)
 1699 * Override with M203
 1700 */
 1701
 1702 /**
 1703 * Default Max Feedrate for G40 (mm/s)
 1704 * (Maximum feedrate for G40 moves)
 1705 * Override with M203
 1706 */
 1707
 1708 /**
 1709 * Default Max Acceleration for G41 (change/s)
 1710 * (Maximum acceleration for G41 moves)
 1711 * Override with M201
 1712 */
 1713
 1714 /**
 1715 * Default Max Retract Speed for G41 (mm/s)
 1716 * (Maximum retract speed for G41 moves)
 1717 * Override with M202
 1718 */
 1719
 1720 /**
 1721 * Default Max Travel Speed for G41 (mm/s)
 1722 * (Maximum travel speed for G41 moves)
 1723 * Override with M203
 1724 */
 1725
 1726 /**
 1727 * Default Max Feedrate for G41 (mm/s)
 1728 * (Maximum feedrate for G41 moves)
 1729 * Override with M203
 1730 */
 1731
 1732 /**
 17
```

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand-alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor has a dark theme. On the left is the Explorer sidebar showing project files like 'Configuration.h' and 'Configuration_adv.h'. The main area displays the following code:

```

1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false          // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false      // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true       // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false     // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

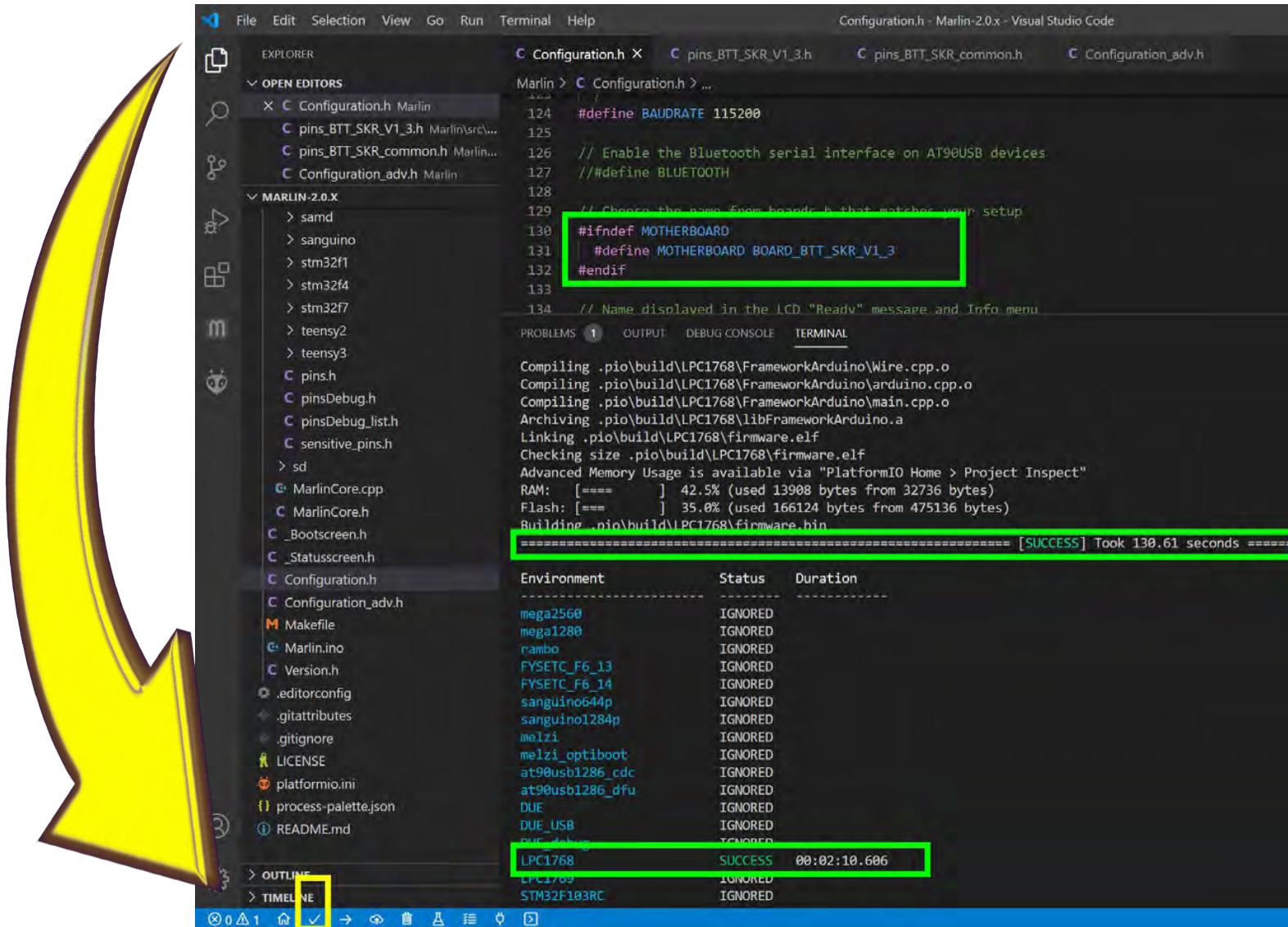
```

A green rectangular box highlights the section of code from line 1049 to line 1051, specifically the definitions for `INVERT_X_DIR`, `INVERT_Y_DIR`, and `INVERT_Z_DIR`. The status bar at the bottom right shows 'Ln 1051, Col 107'.

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand-alone Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in Stand-alone Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

Configuration.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
  Configuration.h Marlin
  pins_BTT_SKR_V1_3.h Marlin\src...
  pins_BTT_SKR_common.h Marlin...
  Configuration_adv.h Marlin
MARLIN-2.0.X
  samd
  sanguino
  stm32f1
  stm32f4
  stm32f7
  teensy2
  teensy3
  pins.h
  pinsDebug.h
  pinsDebug_list.h
  sensitive_pins.h
  sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  Statusscreen.h
  Configuration.h
  Configuration_adv.h
  Makefile
  Marlin.ino
  Version.h
  .editorconfig
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md
OUTLINE
TIMELINE

```

```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

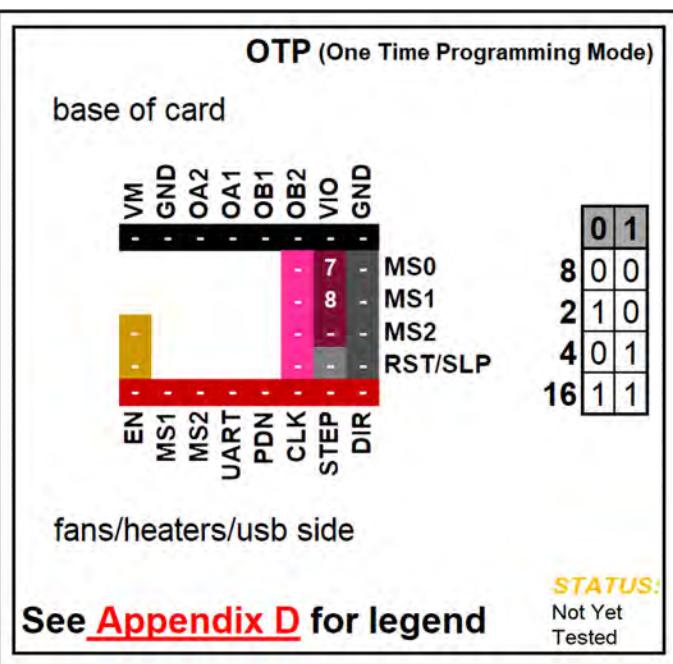
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

Environment	Status	Duration
mega2560	IGNORED	
mega1280	IGNORED	
rambo	IGNORED	
FYSETC_F6_13	IGNORED	
FYSETC_F6_14	IGNORED	
sanguino644p	IGNORED	
sanguino1284p	IGNORED	
melzi	IGNORED	
melzi_optiboot	IGNORED	
at90usb1286_cdc	IGNORED	
at90usb1286_dfu	IGNORED	
DUE	IGNORED	
DUE_USB	IGNORED	
DUET_3D	IGNORED	
LPC1768	SUCCESS	00:02:10.606
LPC1769	IGNORED	
STM32F103RC	IGNORED	

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.



BIQU TMC2208 V3.0

One Time Programming (OTP) Mode

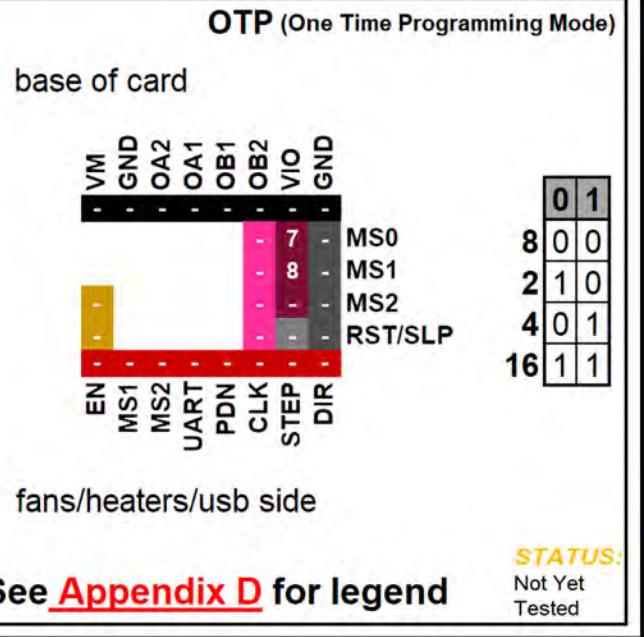
Note: Use 90% of the calculated V_{ref} when tuning the stepper driver board.

Note: Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's " **V_{ref} Test point**" location, as shown in **BLUE**, to set your V_{ref} . See [Appendix A](#) for instruction on how to set the V_{ref} on a driver board.



Driver Chip	MS0	MS1	Steps	Interpolation	Modulation
BIQU® TMC2208 OTP Mode Maximum 16 Subdivision 35V DC 2A (peak)	Low	Low	1 / 8	1 / 256	spreadCycle
	High	Low	1 / 2	1 / 256	spreadCycle
	Low	High	1 / 4	1 / 256	spreadCycle
	High	High	1 / 16	1 / 256	spreadCycle
Driving Current Calculation Formula R_S (Typical Sense Resistor) = 0.11Ω	$I_{MAX} = V_{ref} * 0.9286$ See Appendix B #3. Use 50% to 90% as shown below:			$V_{ref} = I_{MAX} * 1.0769$ See Appendix B #3. Use 50% to 90% as shown below:	
	$I_{MAX} = (V_{ref} * 0.9286) * 0.90$			$V_{ref} = (I_{MAX} * 1.0769) * 0.90$	

- See next page for the LEGEND that belongs to the above Driver Chip Chart.

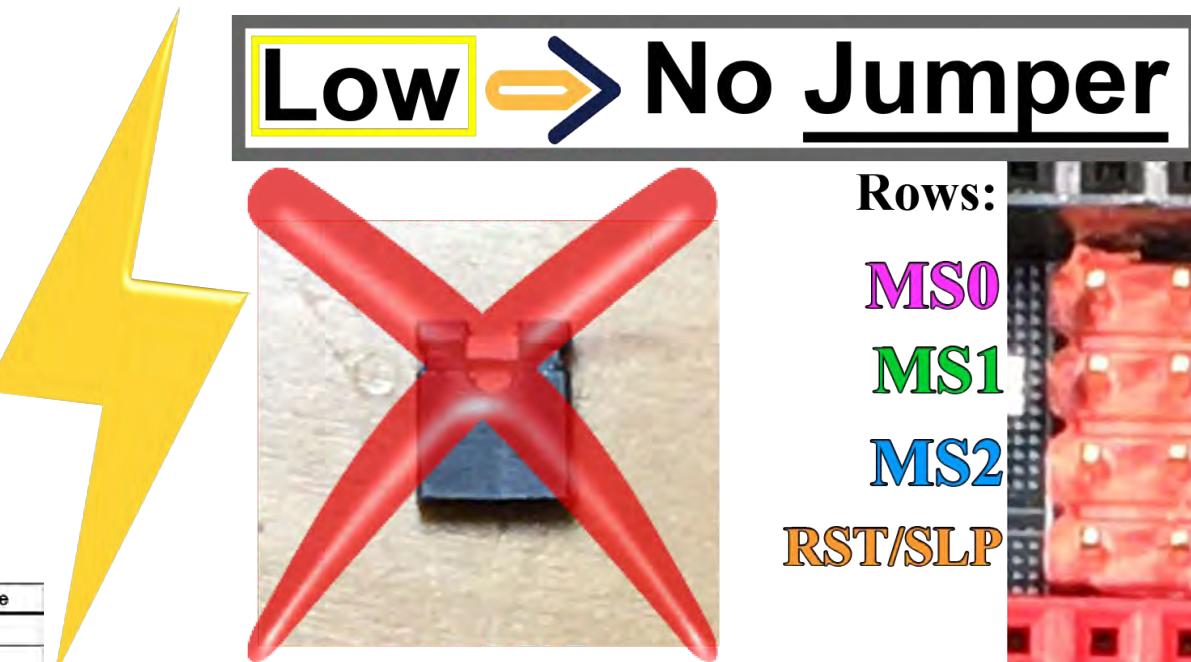
BIQU TMC2208 V3.0One Time Programming (OTP) Mode**Driver Chip Chart:**

Driver Chip	MS0	MS1	MS2	Microstep Resolution	Excitation Mode
LOGO	Low	Low	Low	Full Step	2 Phase
Chip Name	High	Low	Low	1/2 Step	1-2 Phase
Maximum XXX Subdivision	Low	High	Low	1/4 Step	W1-2 Phase
XXV DC xxA (peak)	High	High	Low	1/8 Step	2W1-2 Phase
	Low	Low	High	1/16 Step	4W1-2 Phase
	High	Low	High	1/32 Step	8W1-2 Phase
	Low	High	High	1/64 Step	16W1-2 Phase
	High	High	High	1/128 Step	32W1-2 Phase
Driving Current Calculation Formula	$I_{MAX} = \text{FORMULA}$		$V_{ref} = \text{FORMULA}$		
R_S (Typical Sense Resistor) = x.XX Ω					

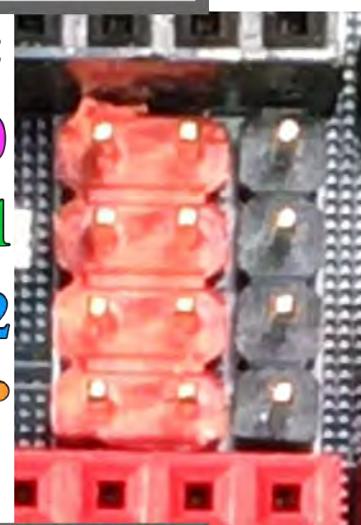
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



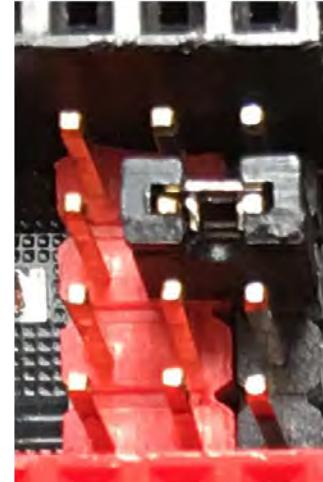
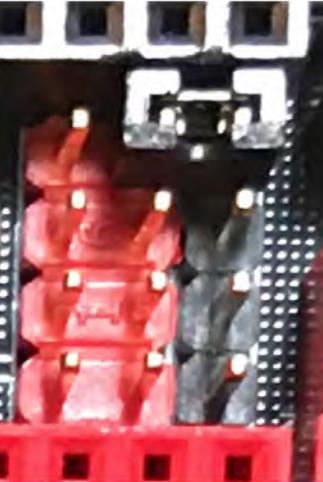
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

**SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

Rows:

MS0**MS1****MS2****RST/SLP**

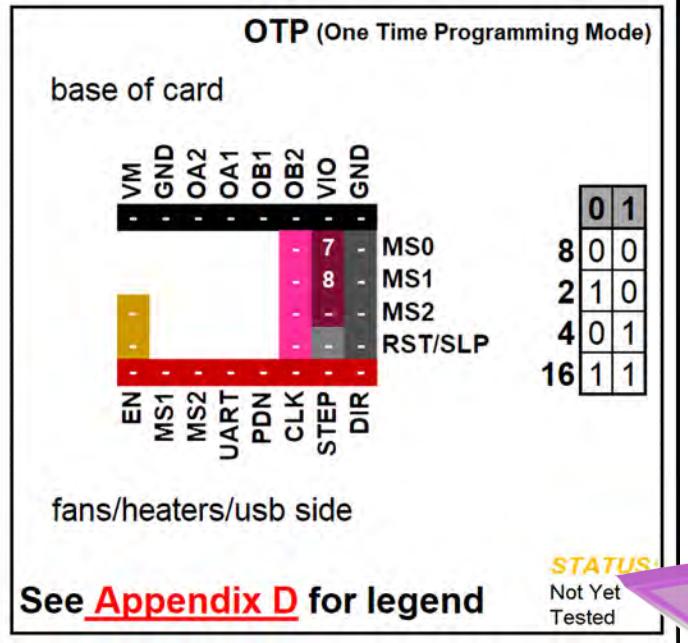
Rows:

MS0**MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

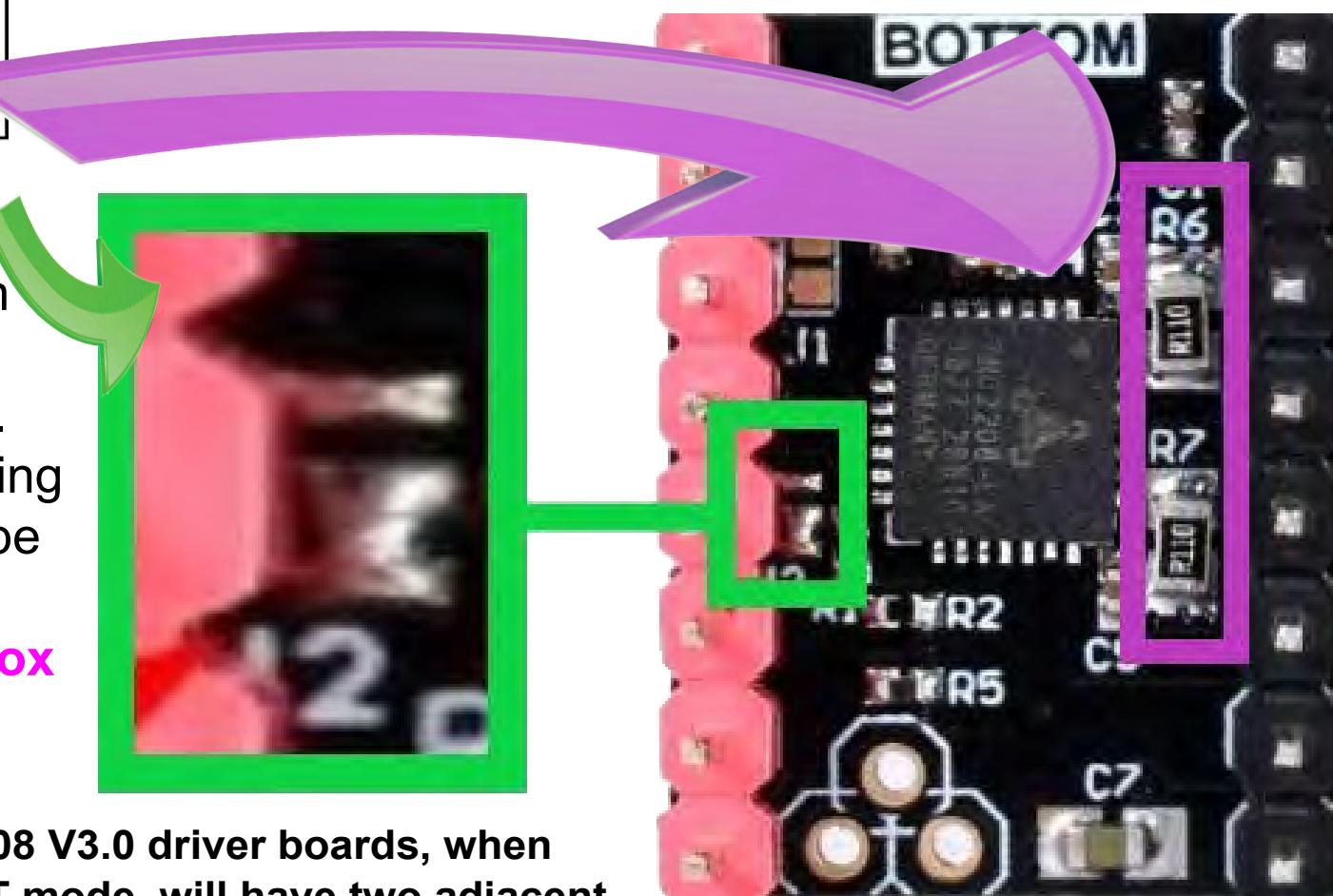
BIQU TMC2208 V3.0One Time Programming (OTP) Mode

NOTE: Stand-alone Mode by default uses StealthChop, if you want SpreadCycle, you **MUST** use OTP mode. See TMC220x Configurator for One-Time-Programming Information: [TMC220x Configurator](#).

Important: To place BIQU TMC2208 V3.0 into OTP mode, at J2, you **must solder the top two pads together OR solder the bottom two pads together** on the bottom of the driver board.



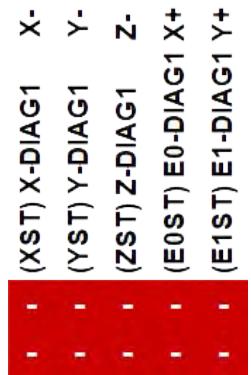
As an example, the picture shows the TOP two pads soldered together, as shown in the **GREEN box**, so that OTP mode can be obtained. To do One-Time-Programming (OTP), the TMC2208 must be placed in UART mode to program it. The **PURPLE box** shows the location of the current sense resistors (R_s).



MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).

OTP (One Time Programming) ModeOne Time Programming (OTP) Mode

Note: Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation.



Note: Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation.

XUART

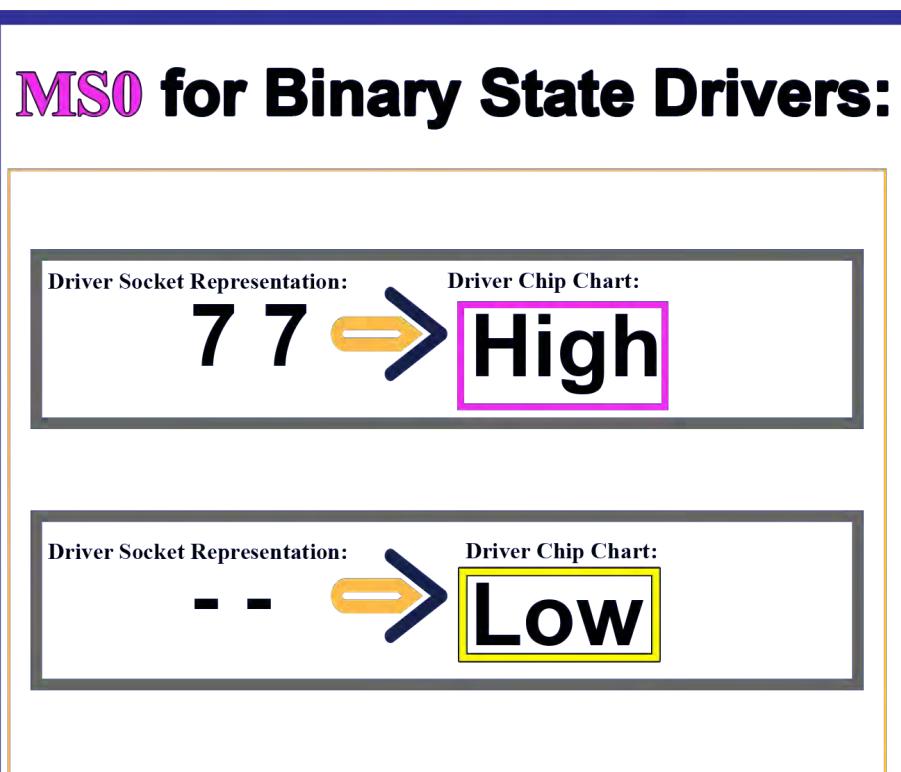
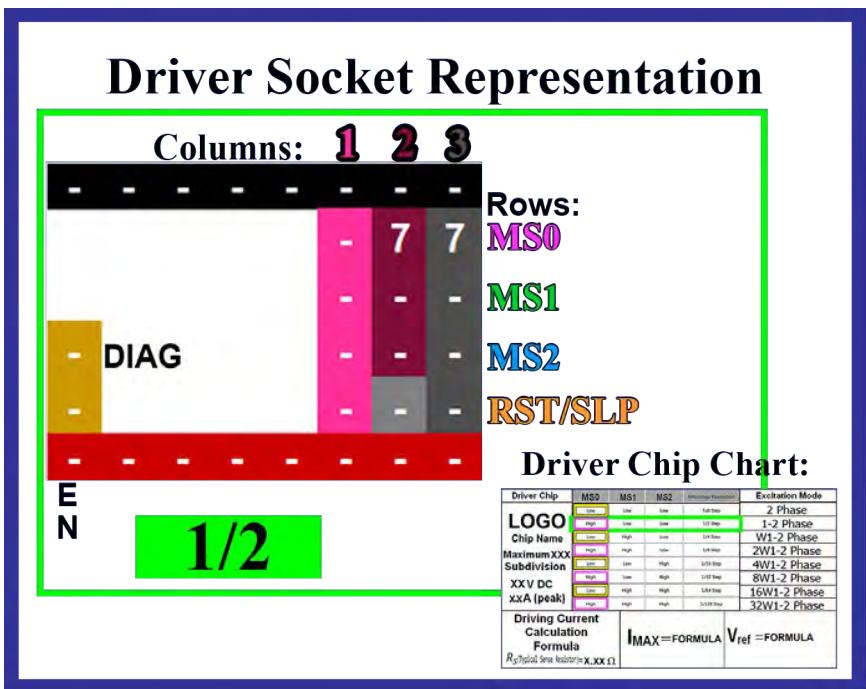
YUART

ZUART

E0 UART

E1 UART



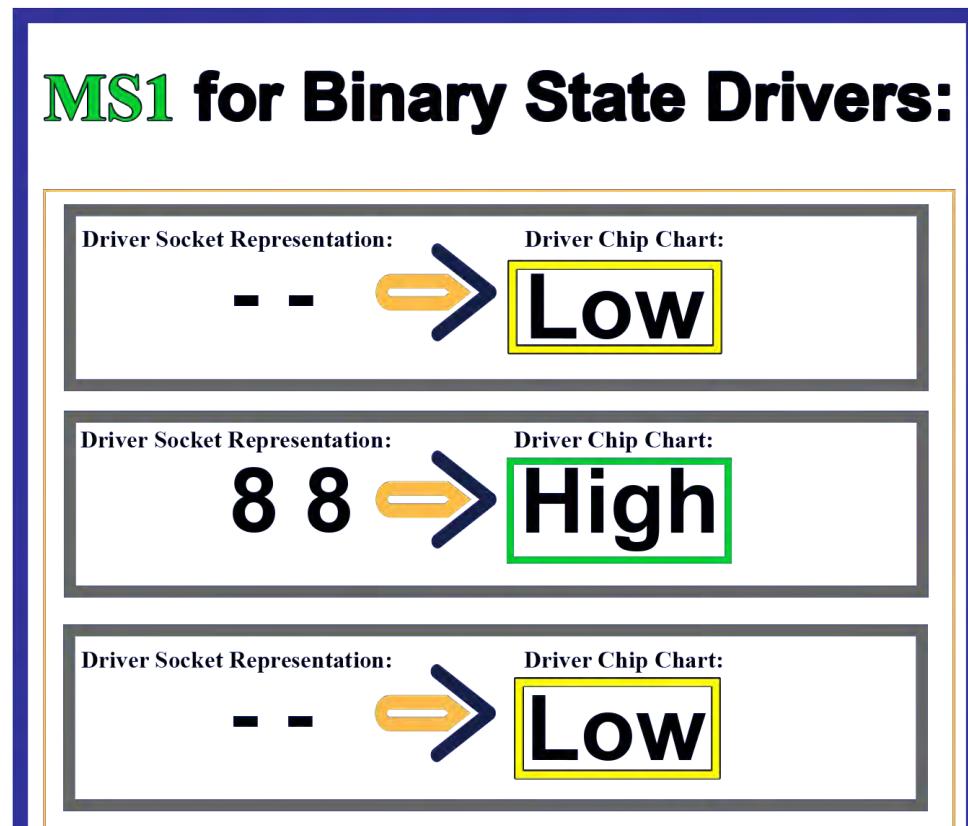
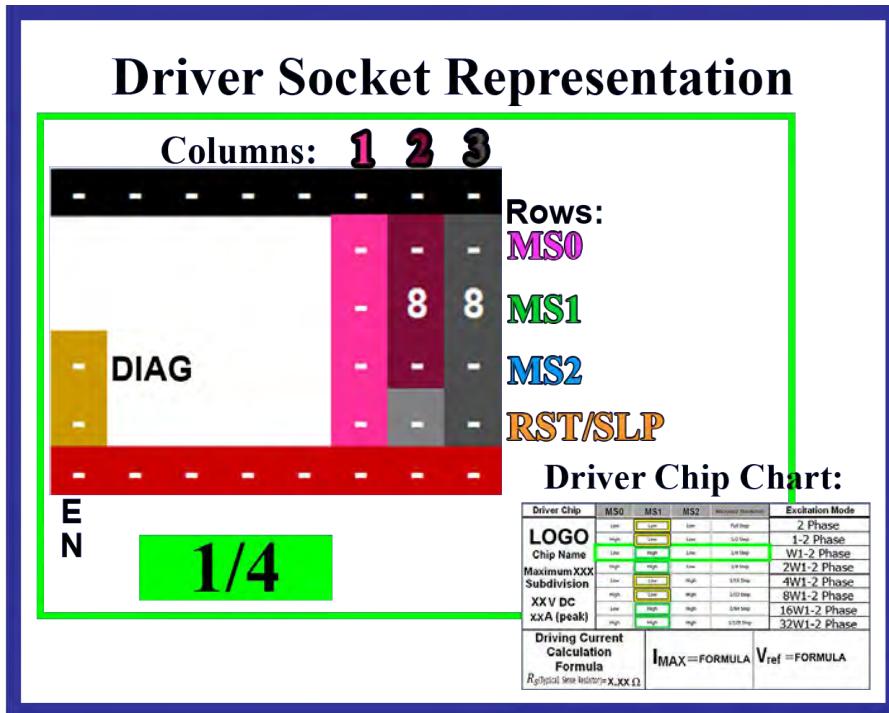
OTP (One Time Programming) Mode**One Time Programming (OTP) Mode****SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers**

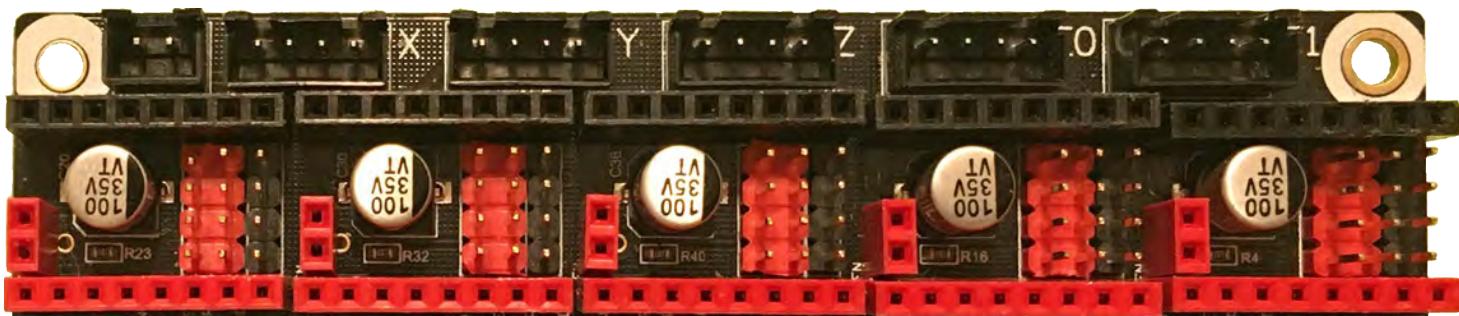
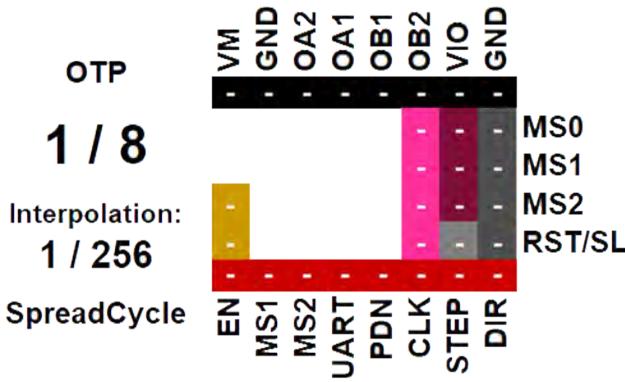
Meaning:

Driver Chip Chart:

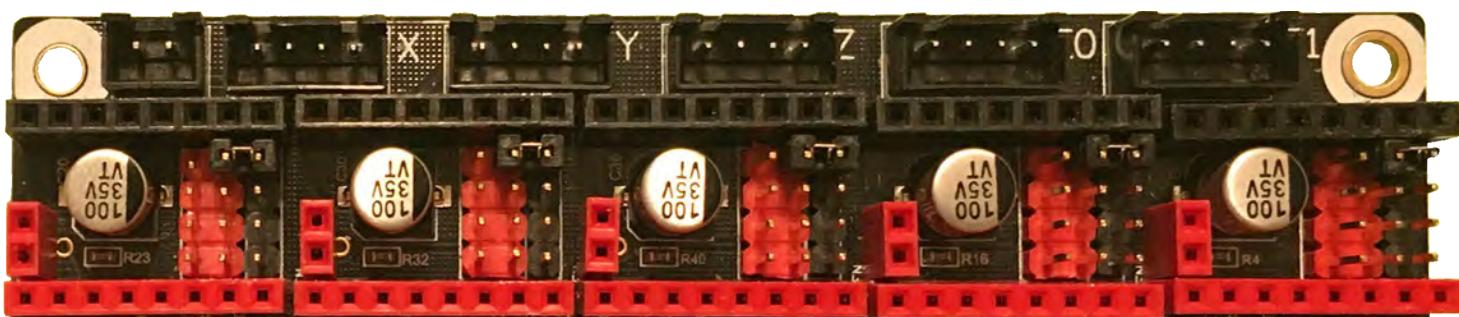
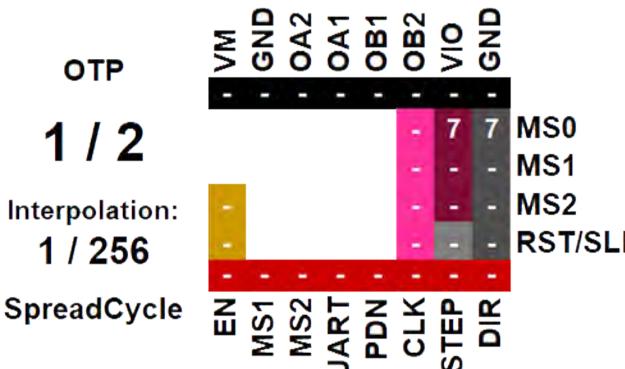
High → **set Jumper between column 2 and column 3 on the MS0 row**

Low → **No Jumper Set**

OTP (One Time Programming) ModeOne Time Programming (OTP) Mode

OTP (One Time Programming) Mode**One Time Programming (OTP) Mode**

See [Appendix D](#) for legend



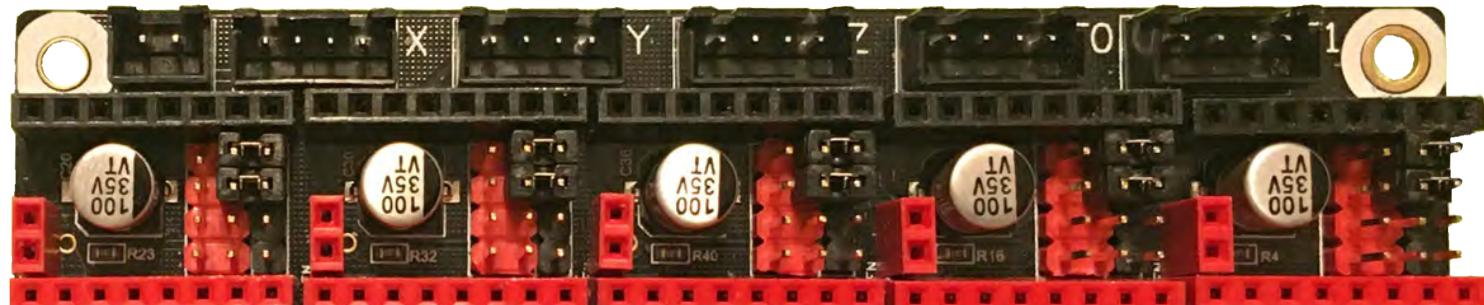
See [Appendix D](#) for legend

OTP (One Time Programming) Mode**One Time Programming (OTP) Mode**

OTP	VM	GND	OA2	OA1	OB1	OB2	VIO	GND
1 / 4	-	-	-	-	-	-	-	-
Interpolation:	-	-	-	-	-	-	-	-
1 / 256	-	-	-	-	-	-	-	-
SpreadCycle	EN	MS1	MS2	UART	PDN	CLK	STEP	DIR



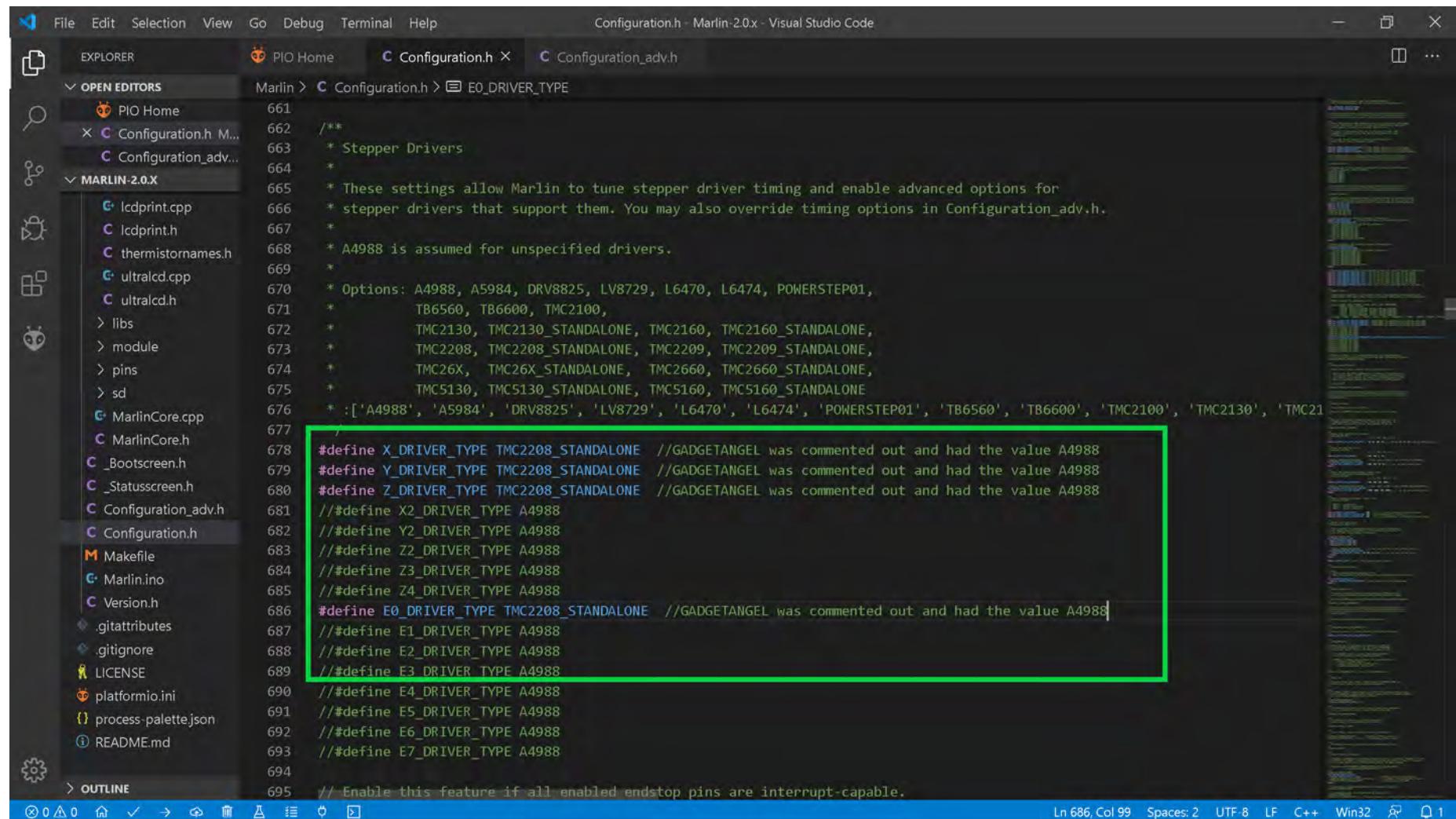
OTP	VM	GND	OA2	OA1	OB1	OB2	VIO	GND
1 / 16	-	-	-	-	-	-	-	-
Interpolation:	-	-	-	-	-	-	-	-
1 / 256	-	-	-	-	-	-	-	-
SpreadCycle	EN	MS1	MS2	UART	PDN	CLK	STEP	DIR



The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

NOTE: Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2208 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
  PIO Home
  Configuration.h M...
  Configuration_adv...
MARLIN-2.0.X
  Lcdprint.cpp
  Lcdprint.h
  thermistornames.h
  ultralcd.cpp
  ultralcd.h
  > libs
  > module
  > pins
  > sd
  MarlinCore.cpp
  MarlinCore.h
  _Bootscreen.h
  _Statusscreen.h
  Configuration_adv.h
  Configuration.h
  Makefile
  Marlin.ino
  Version.h
  .gitattributes
  .gitignore
  LICENSE
  platformio.ini
  process-palette.json
  README.md

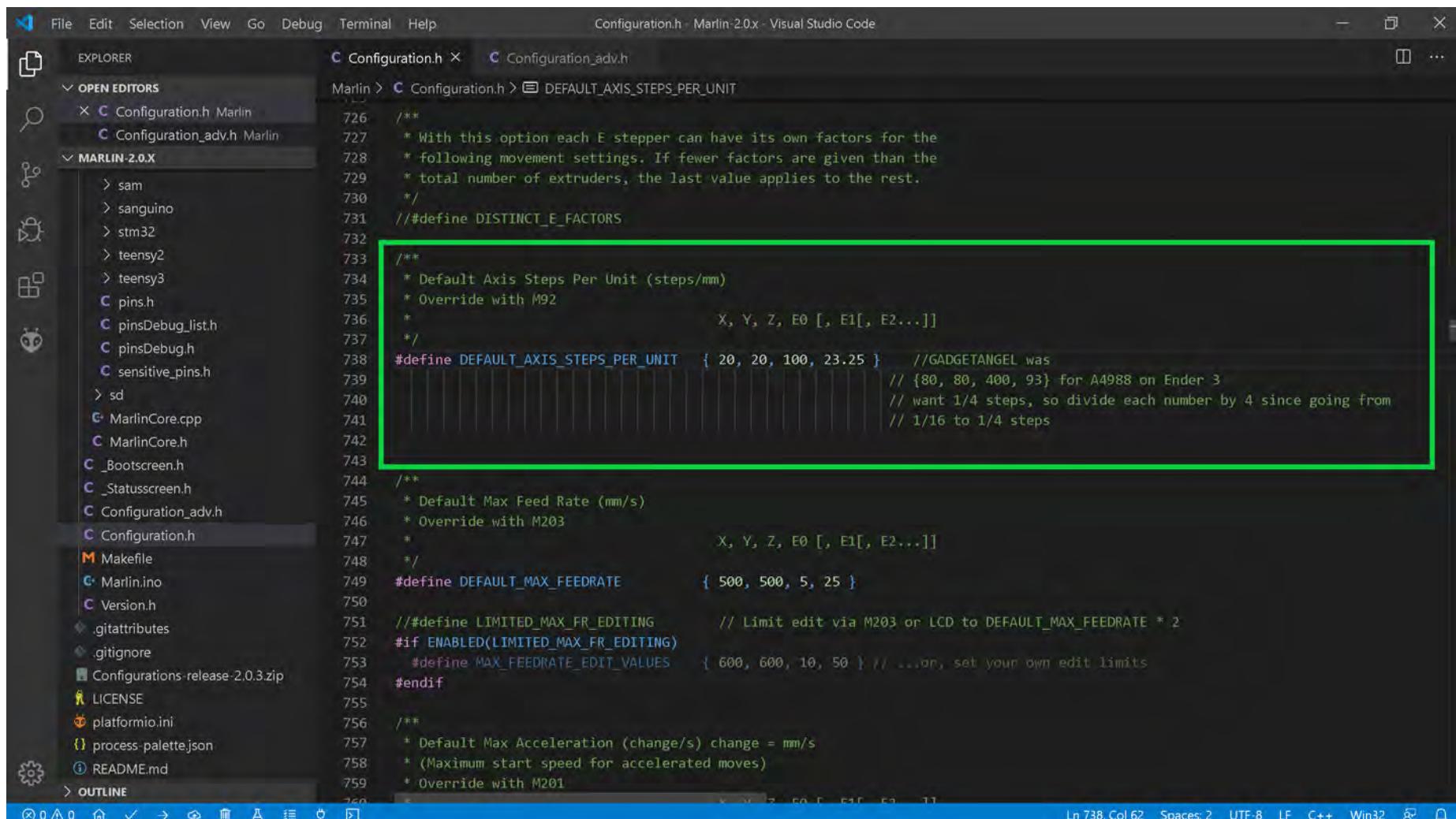
  661 /**
  662 * Stepper Drivers
  663 *
  664 */
  665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
  666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
  667 *
  668 * A4988 is assumed for unspecified drivers.
  669 *
  670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
  671 *           TB6560, TB6600, TMC2100,
  672 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
  673 *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
  674 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
  675 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
  676 *           :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
  677 */
  678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  681 //#define X2_DRIVER_TYPE A4988
  682 //#define Y2_DRIVER_TYPE A4988
  683 //#define Z2_DRIVER_TYPE A4988
  684 //#define Z3_DRIVER_TYPE A4988
  685 //#define Z4_DRIVER_TYPE A4988
  686 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
  687 //#define E1_DRIVER_TYPE A4988
  688 //#define E2_DRIVER_TYPE A4988
  689 //#define E3_DRIVER_TYPE A4988
  690 //#define E4_DRIVER_TYPE A4988
  691 //#define E5_DRIVER_TYPE A4988
  692 //#define E6_DRIVER_TYPE A4988
  693 //#define E7_DRIVER_TYPE A4988
  694
  695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT_AXIS_STEPS_PER_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT_AXIS_STEPS_PER_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following snippet:

```

    /**
     * With this option each E stepper can have its own factors for the
     * following movement settings. If fewer factors are given than the
     * total number of extruders, the last value applies to the rest.
     */
#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *
 *          X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 *
 *          X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

#ifndef LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
#if ENABLED(LIMITED_MAX_FR_EDITING)
#define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif
```

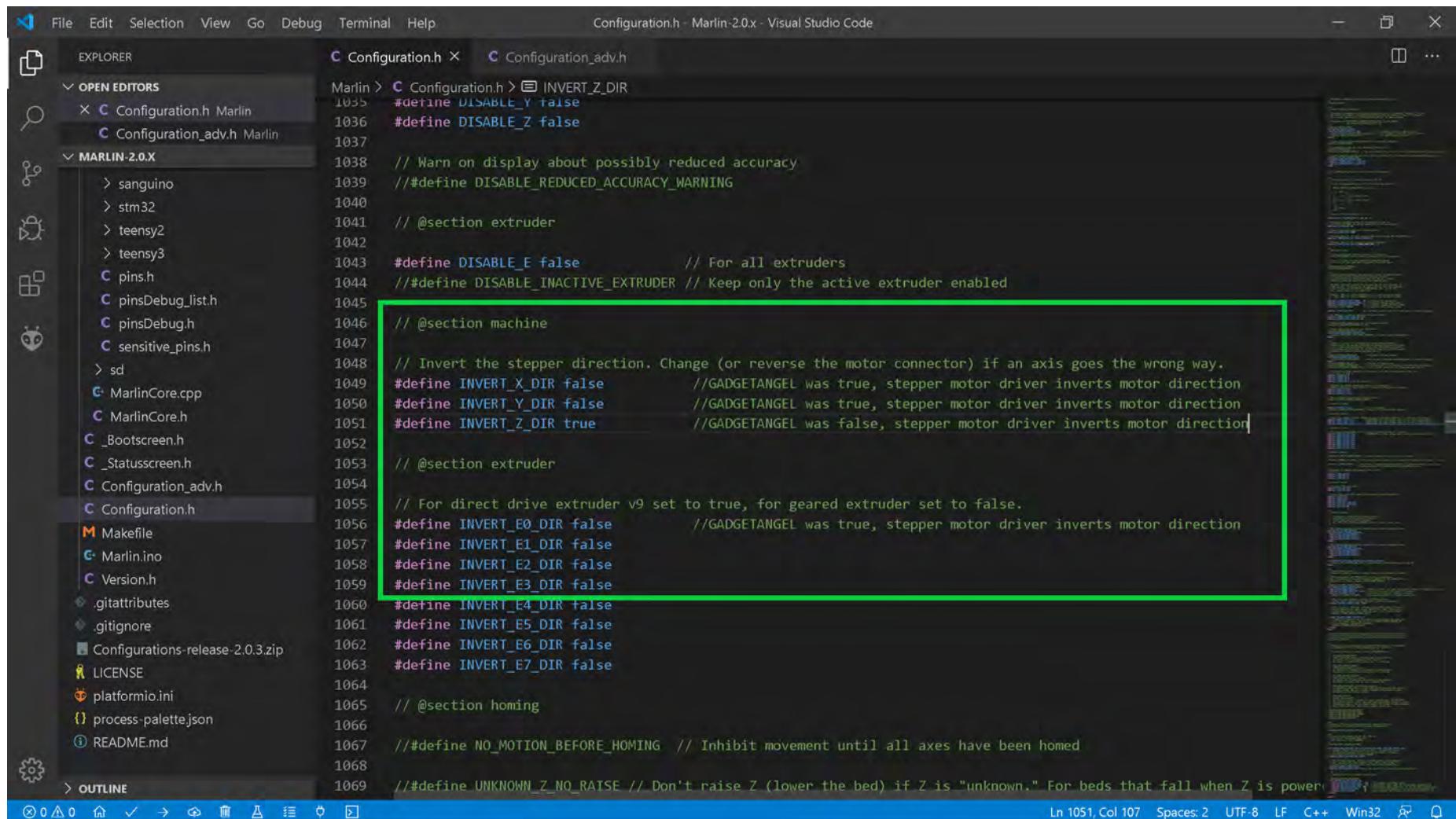
```

A green rectangular box highlights the line `#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 }`. The status bar at the bottom right indicates 'Ln 738, Col 62'.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet:

```

Marlin > C Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

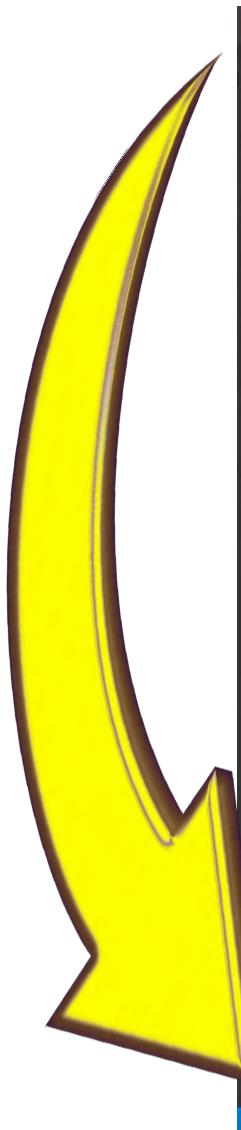
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`, indicating that this setting was false and needs to be changed to true for the BIQU TMC2208 V3.0 Drivers.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src... pins\_BTT\_SKR\_common.h Marlin\src... Configuration\_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the name below to that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[=====] [SUCCESS] Took 130.61 seconds =====

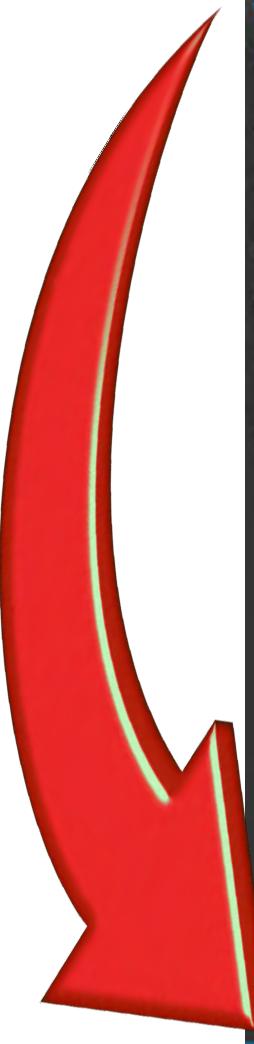
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| ramps           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino44p     | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUE_dfu         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

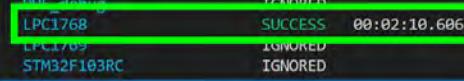
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in One Time Programming (OTP) Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.







File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug\_list.h sensitive\_pins.h sd MarlinCore.cpp MarlinCore.h \_Bootscreen.h Statusscreen.h Configuration.h Configuration\_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

Configuration.h

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

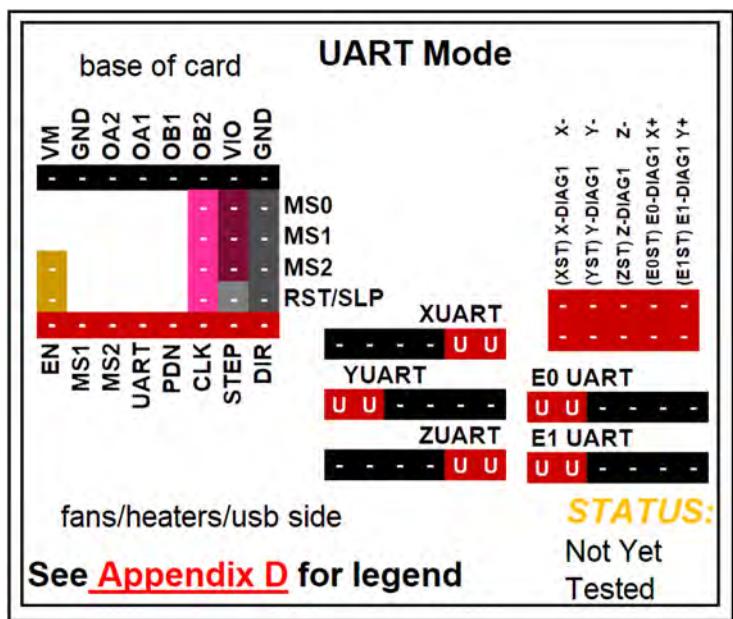
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET_3D         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

> OUTLINE > TIMELINE

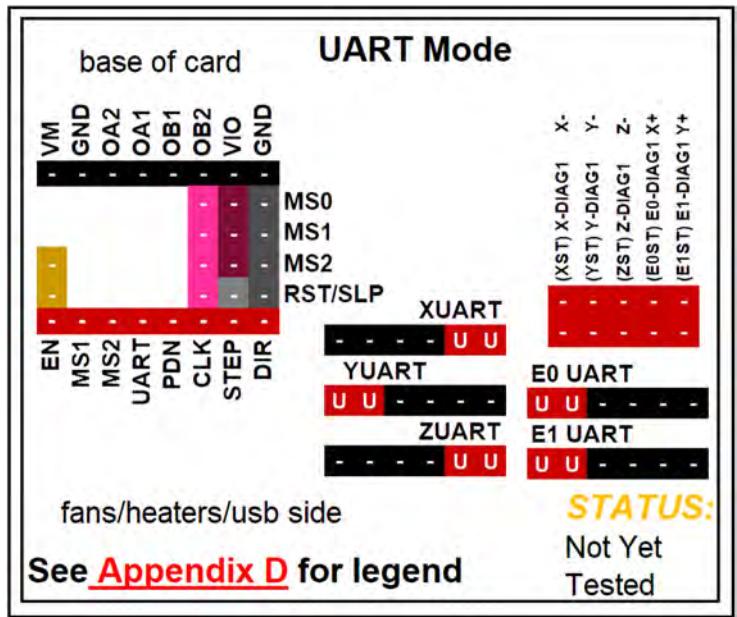
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC2208 V3.0****UART Mode**

**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

|                                                                                                                                                                                          |                                                                                                        |                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Driver Chip</b><br><br><b>TMC2208</b><br>UART Mode<br>Maximum 256 Subdivision<br>35V DC<br>2A (peak) | <b>Steps are set inside<br/>of your Firmware</b>                                                       |                                                                                                        |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.11Ω                                                                                                     | $I_{MAX} = V_{ref}$<br>See Appendix B #4. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90$ | $V_{ref} = I_{MAX}$<br>See Appendix B #4. Use 50% to 90% as shown below:<br>$V_{ref} = V_{ref} * 0.90$ |

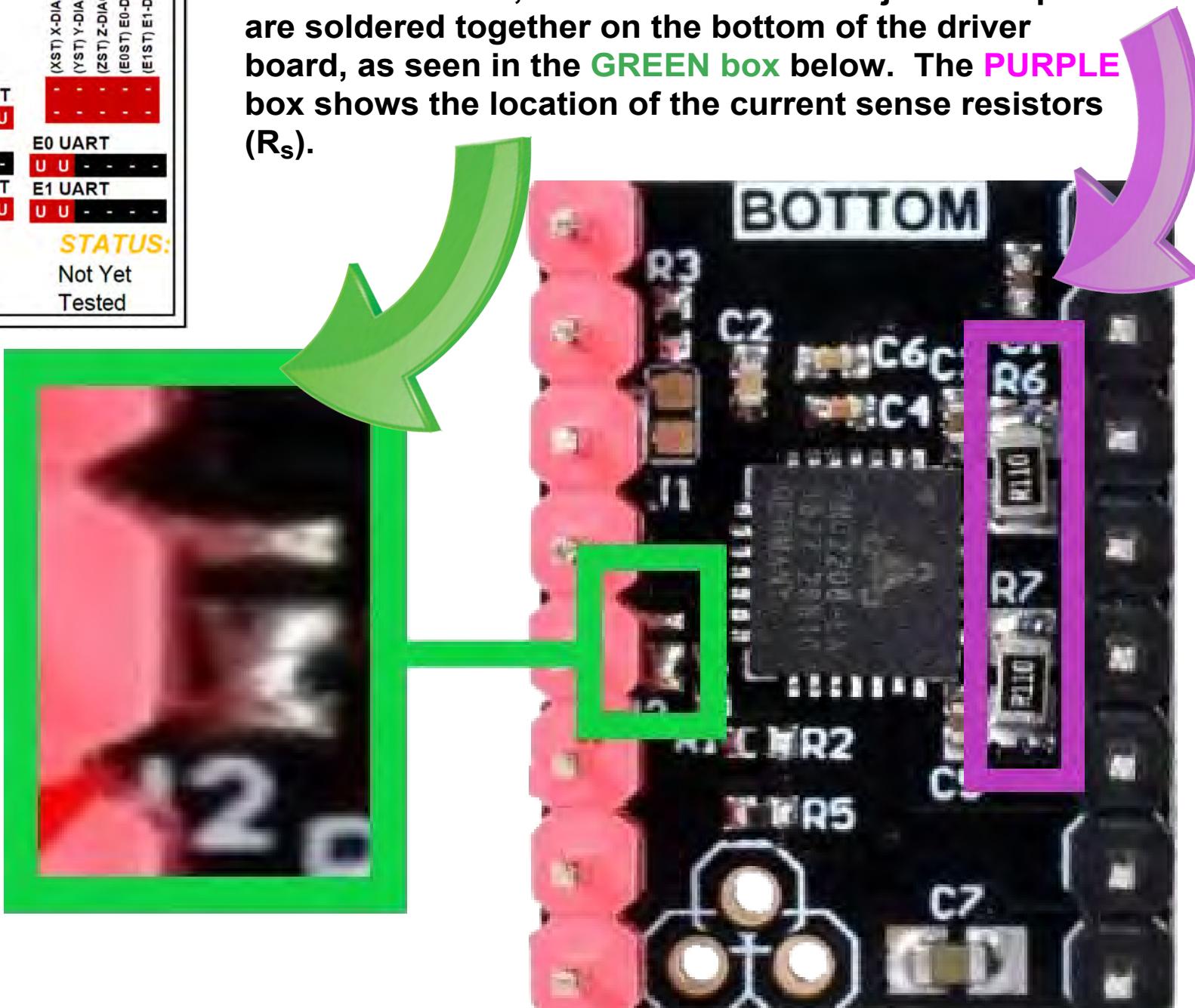


**MOST BIQU TMC2208 V3.0 driver boards, when purchased for UART mode, will have two adjacent J2 pads already soldered together (located on the bottom of the driver board).**

## BIGU TMC2208 V3.0

### UART Mode

**Important:** To ensure that the BIGU TMC2208 V3.0 is in UART Mode, check to see if two adjacent J2 pads are soldered together on the bottom of the driver board, as seen in the **GREEN box** below. The **PURPLE box** shows the location of the current sense resistors ( $R_s$ ).



BIQU TMC2208 V3.0UART Mode

# UART Mode

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

$R_s = R_{050}$  is 0.05 Ohms

$R_s = R_{062}$  is 0.062 Ohms

$R_s = R_{068}$  is 0.068 Ohms

$R_s = R_{075}$  is 0.075 Ohms

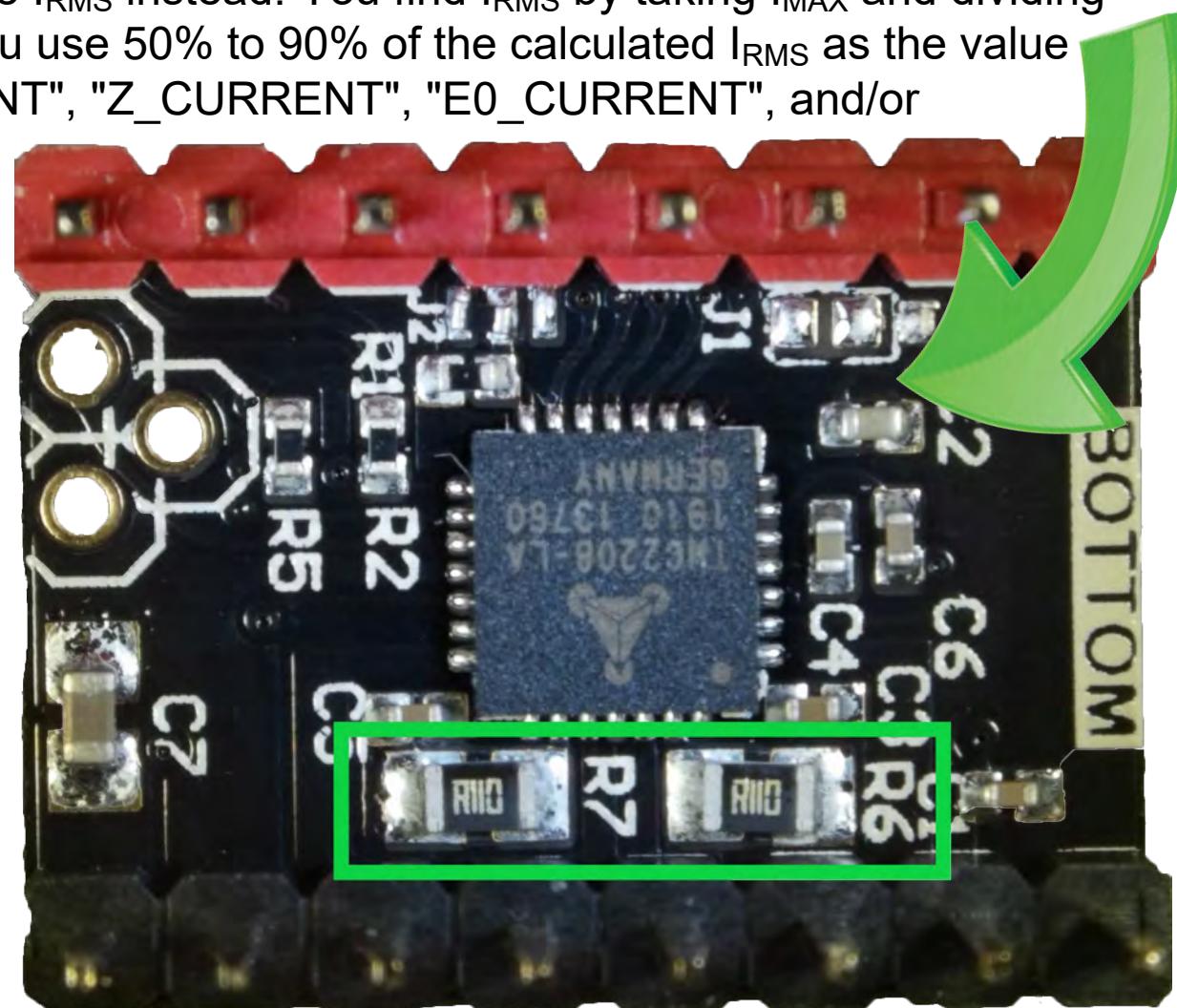
$R_s = R_{100}$  is 0.1 Ohms

$R_s = R_{110}$  is 0.11 Ohms

$R_s = R_{150}$  is 0.15 Ohms

$R_s = R_{200}$  is 0.2 Ohms

$R_s = R_{220}$  is 0.22 Ohms



BIQU TMC2208 V3.0UART Mode

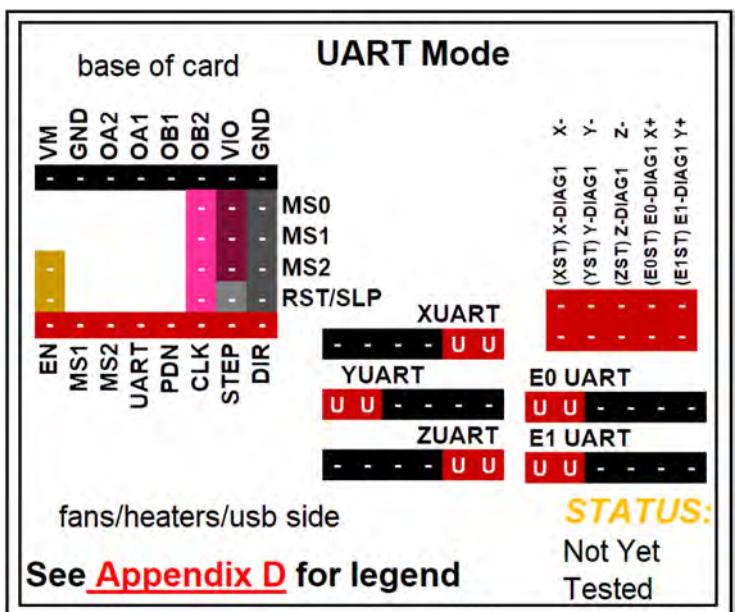
# UART Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.



**Note:** Set the "U" Jumper(s) for UART MODE!



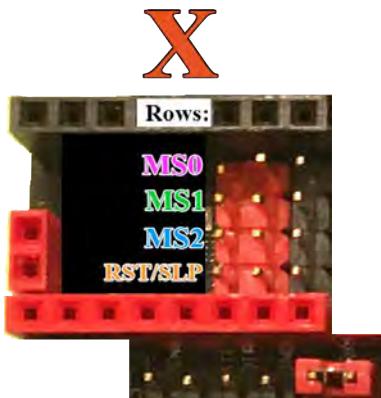


See [Appendix D](#) for legend

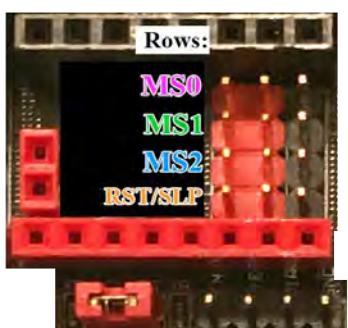
## Driver Socket Representation:

**Meaning:**

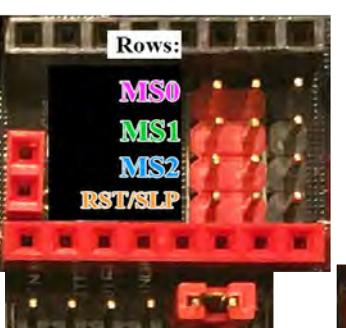
# U U ➔ Jumper set



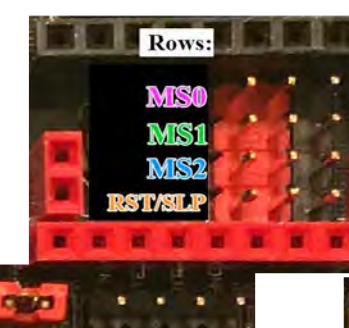
UU



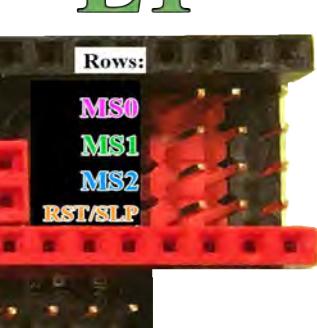
UU



UU



11



EAST

BIQU TMC2208 V3.0

## UART Mode

**SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode**

|    | RX   | TX   |
|----|------|------|
| X  | 1.17 | 4.29 |
| Y  | 1.15 | 1.16 |
| Z  | 1.10 | 1.14 |
| E0 | 1.8  | 1.9  |
| E1 | 1.1  | 1.4  |

# STEPPER DRIVER UART COMMUNICATIONS DO NOT USE TX PINS



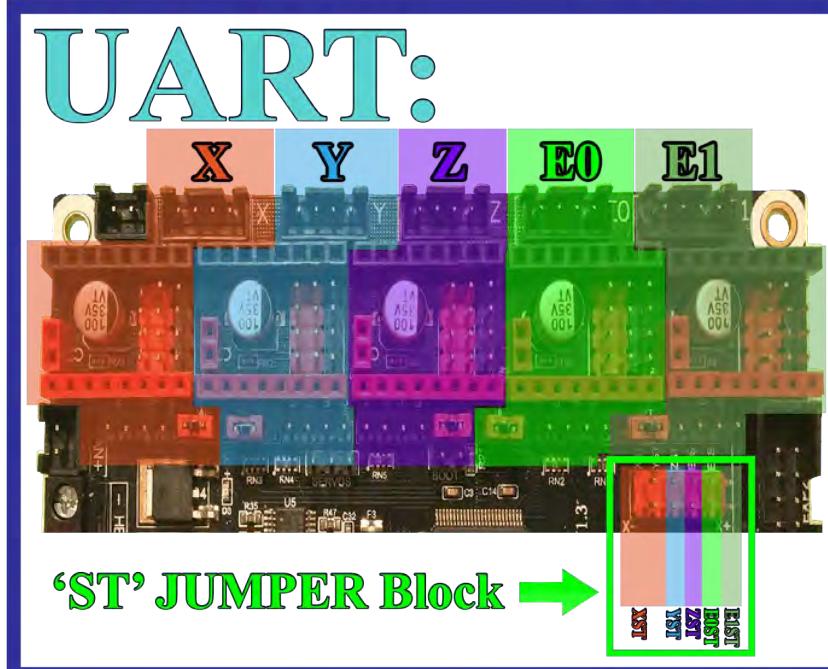
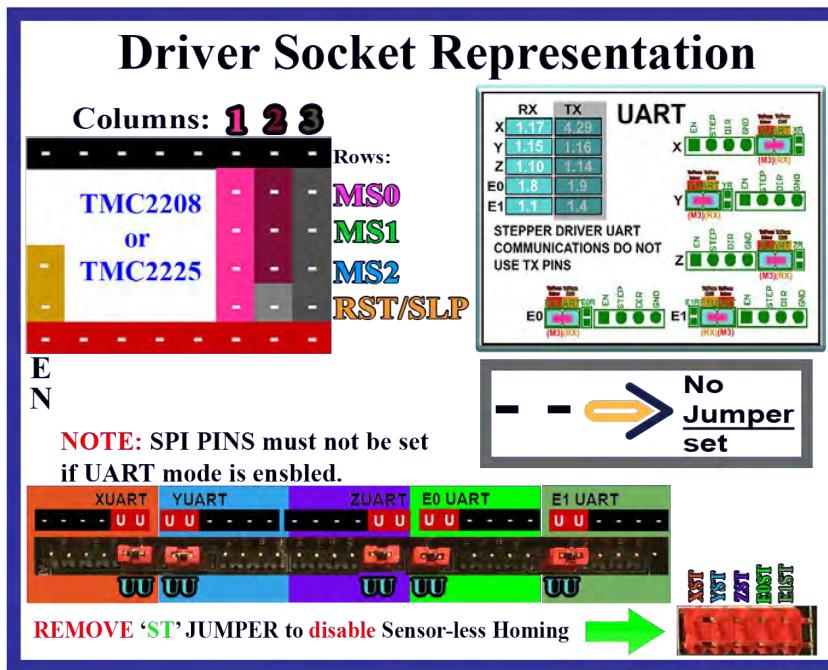
**REMOVE ‘ST’ JUMPER(s) to disable Sensor-less Homing – for drivers without stallGaurd™ feature**



**BIQU TMC2208 V3.0****UART Mode**

# UART Mode

## SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode



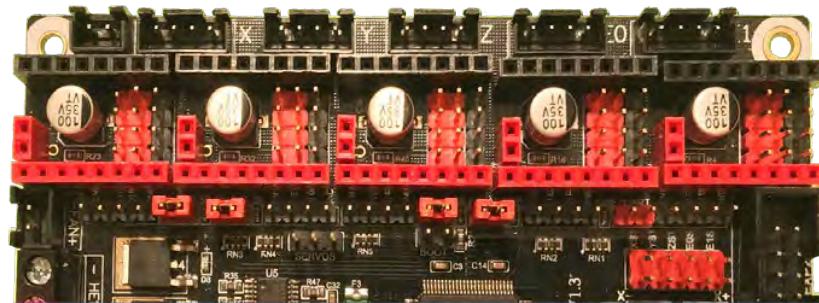
UART Capable Drivers **without** stallGuard™ feature: TMC2208, and TMC2225

**Driver Socket Representation:** **UU** → **Jumper Set**

**Meaning:** set Jumper so that both **U PINS** are covered by the **Jumper**

**Each Axis has its own UART PINS (UU).** You can set as many axes as you need. The **UART:** picture shows all axes having UART enabled **AND** all **'ST' JUMPERS REMOVED.** All other PINS need to be empty for UART mode to work properly.

Here is an example of only the X, Y, Z, and E0 axes having UART enabled **AND** **'ST' JUMPERS removed** for ALL Axes:



## UART Mode

BIQU TMC2208 V3.0

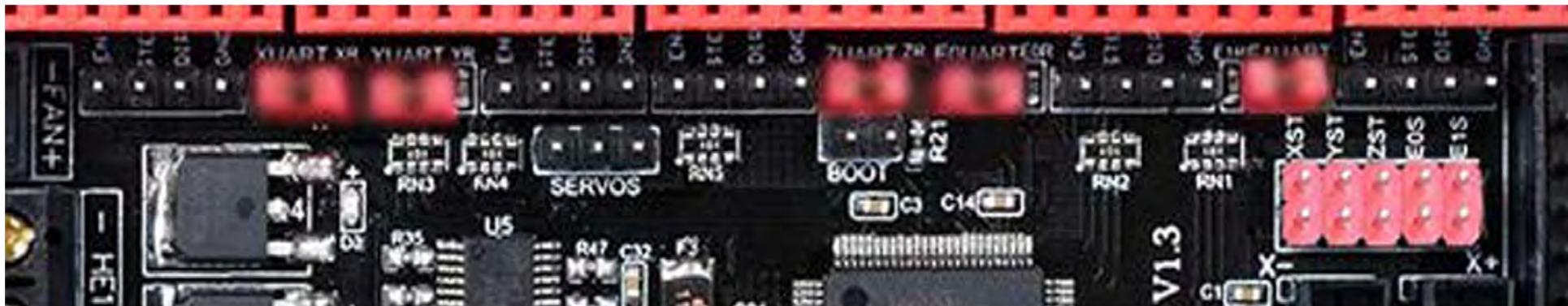
UART Mode

### Information on Sensor-less Homing

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing.

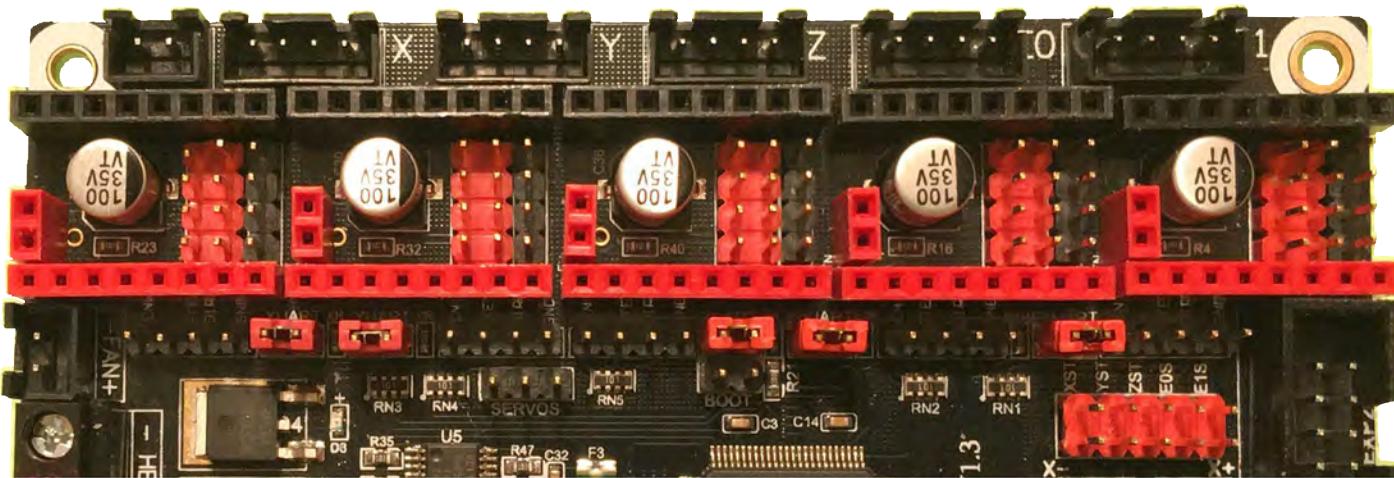
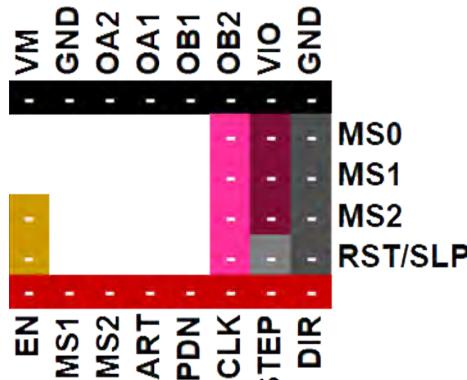
Please read the PREFACE to this manual on “Stall detection and Sensor-less Homing”.

**NOTE:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.

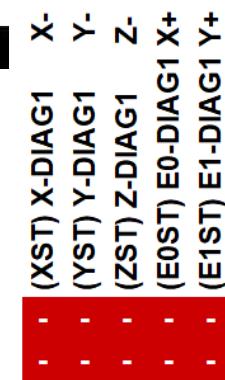


**BIQU TMC2208 V3.0****UART Mode**

**Note:** Set the "U" Jumper(s) for UART MODE!

**UART**

See [Appendix D](#) for legend



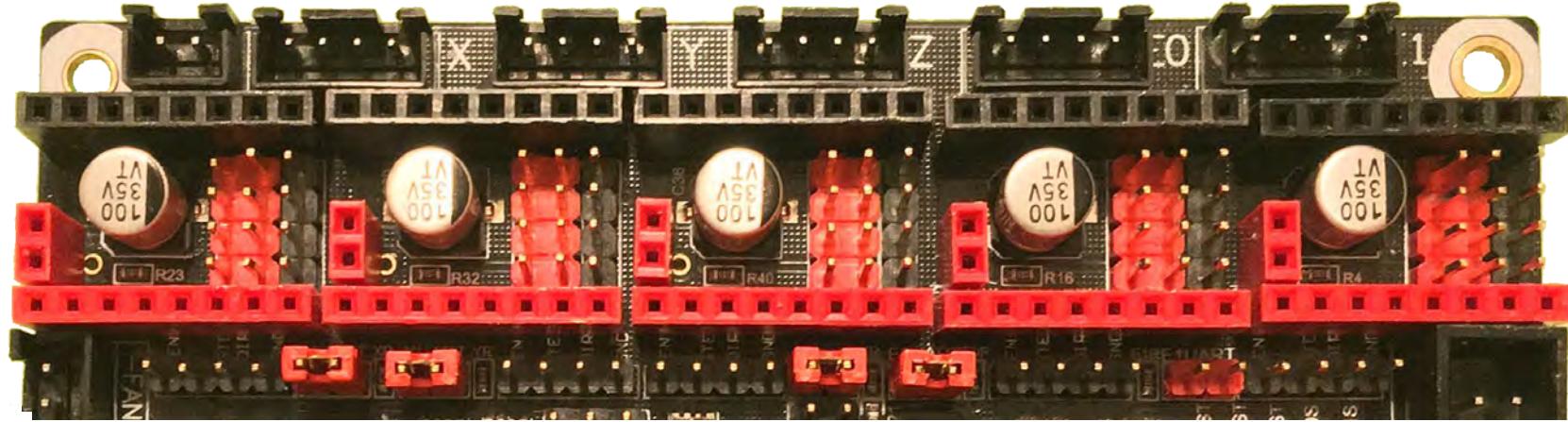
**Note:** TMC2208 does not have sensor-less homing capability.

# BIQU TMC2208 V3.0

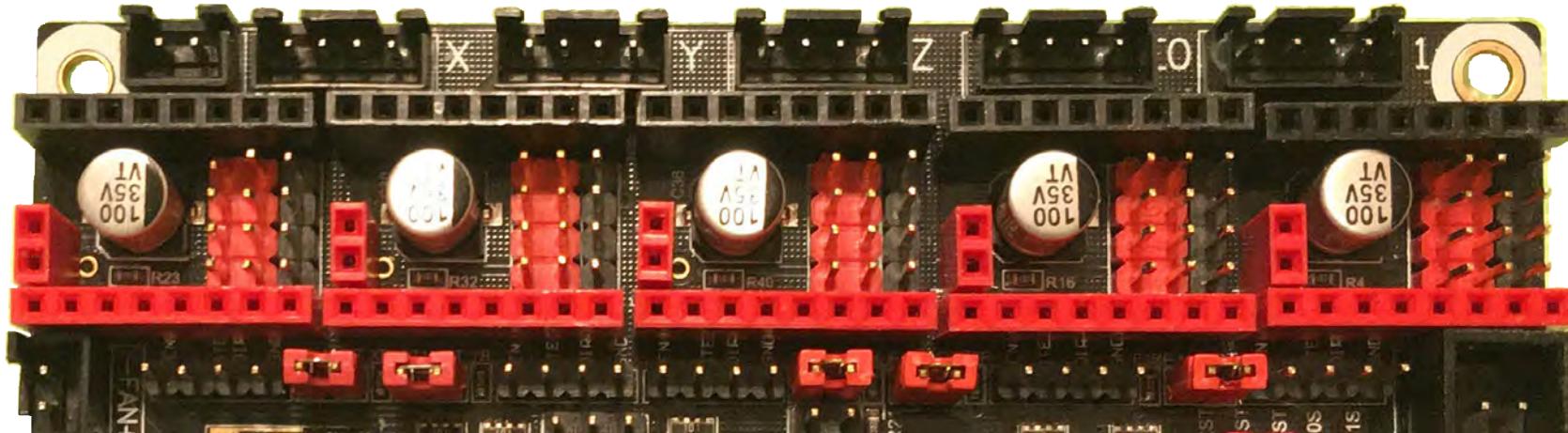
## UART Mode

### Examples of Different UART Configurations

X, Y, Z,  
and E0 axes  
configured for  
UART mode.



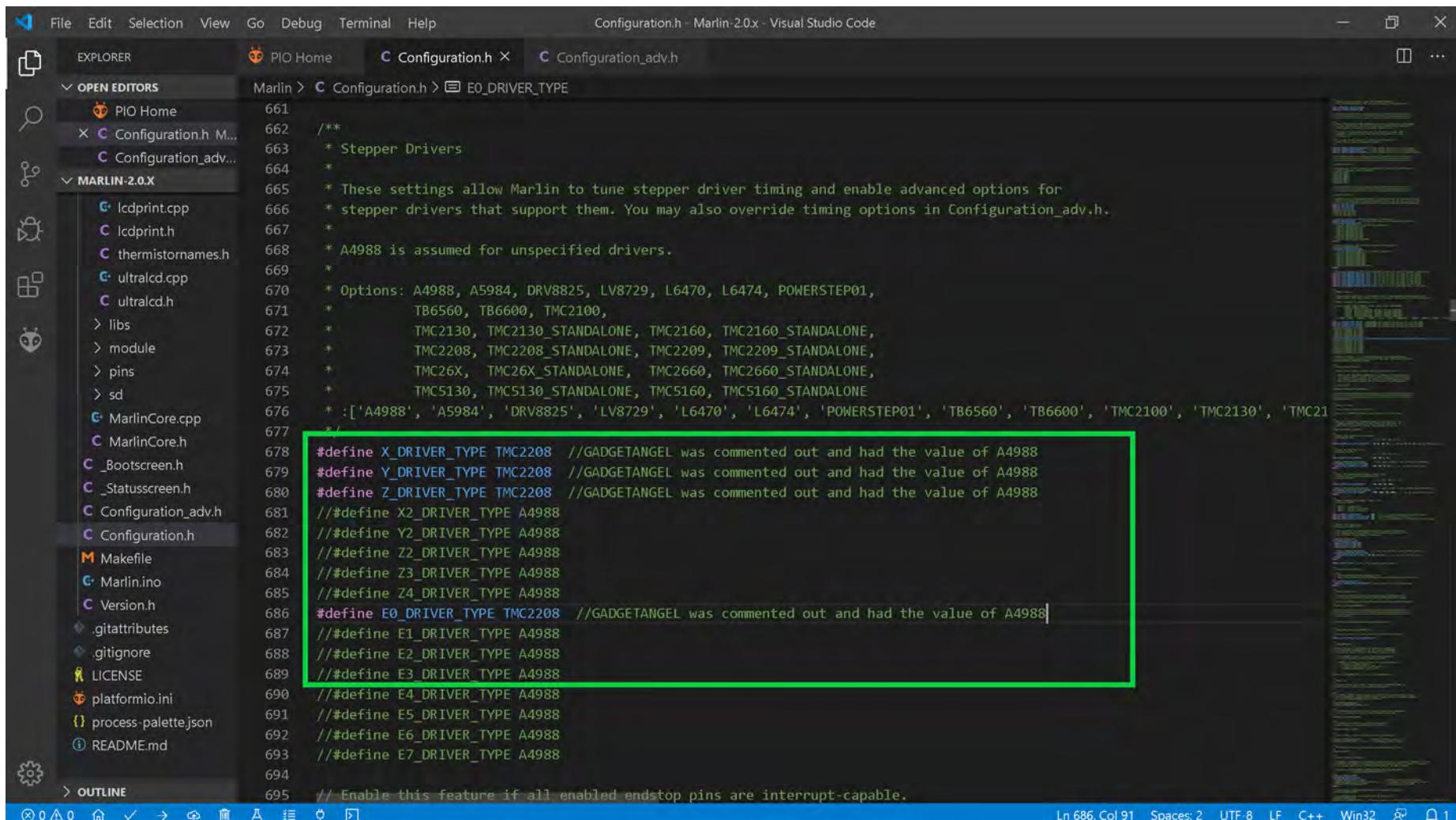
X, Y, Z, E0  
and E1 axes  
configured for  
UART mode.



## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2208 V3.0 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
Marlin > Configuration.h M... Configuration_adv.h
 661
 662 /**
 663 * Stepper Drivers
 664 *
 665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 667 *
 668 * A4988 is assumed for unspecified drivers.
 669 *
 670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 671 * TB6560, TB6600, TMC2100,
 672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2160', 'TMC2208', 'TMC2209', 'TMC2660', 'TMC5130', 'TMC5160']
 677 */
#define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
 //#define X2_DRIVER_TYPE A4988
 //#define Y2_DRIVER_TYPE A4988
 //#define Z2_DRIVER_TYPE A4988
 //#define Z3_DRIVER_TYPE A4988
 //#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
 //#define E1_DRIVER_TYPE A4988
 //#define E2_DRIVER_TYPE A4988
 //#define E3_DRIVER_TYPE A4988
 //#define E4_DRIVER_TYPE A4988
 //#define E5_DRIVER_TYPE A4988
 //#define E6_DRIVER_TYPE A4988
 //#define E7_DRIVER_TYPE A4988
 690
 691
 692
 693
 694
 695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037 // Warn on display about possibly reduced accuracy
1038 // #define DISABLE_REDUCED_ACCURACY_WARNING
1039 // @section extruder
1040
1041 #define DISABLE_E false // For all extruders
1042 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1043
1044 // @section machine
1045
1046 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1047 // GADGETANGEL was true, stepper motor driver inverts motor direction
1048 // GADGETANGEL was true, stepper motor driver inverts motor direction
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

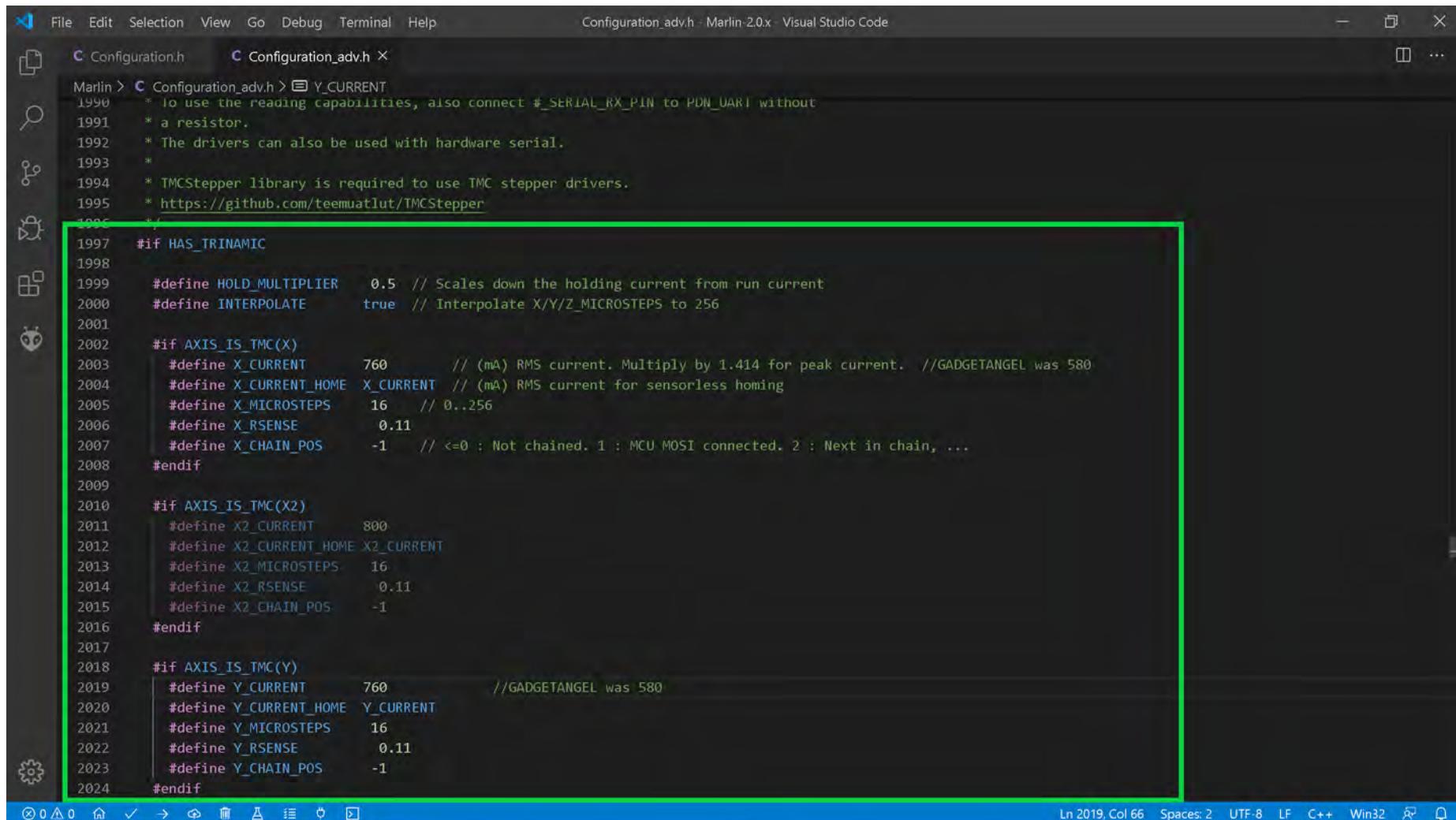
```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC2208 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.11. Ensure "Y\_RSENSE" is set to 0.11.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif

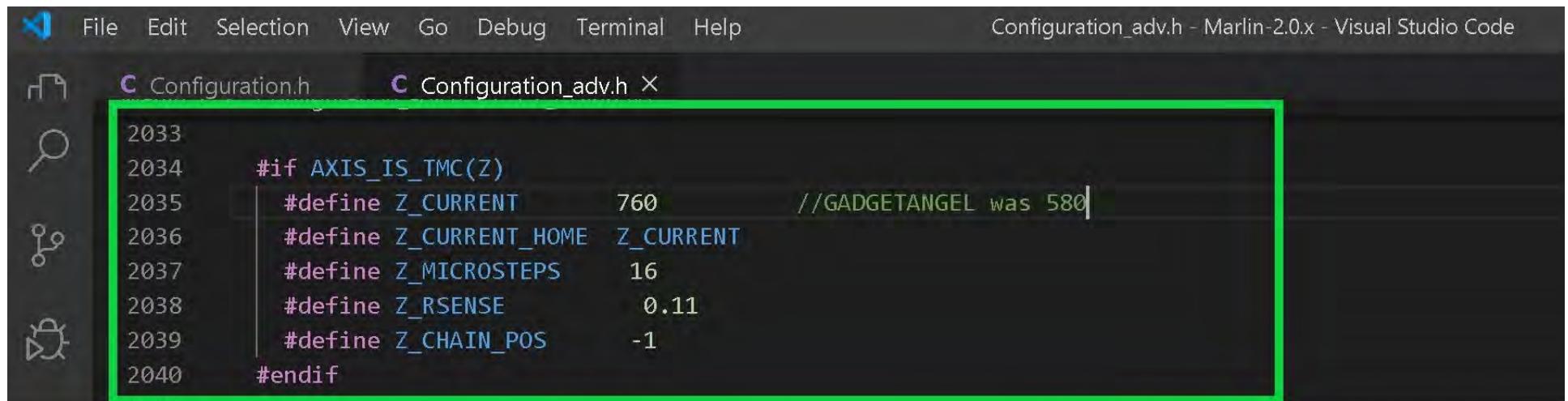
```

Ln 2019, Col 66 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.11. Ensure "E0\_RSENSE" is set to 0.11.
- If you do not want to use  $V_{ref}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



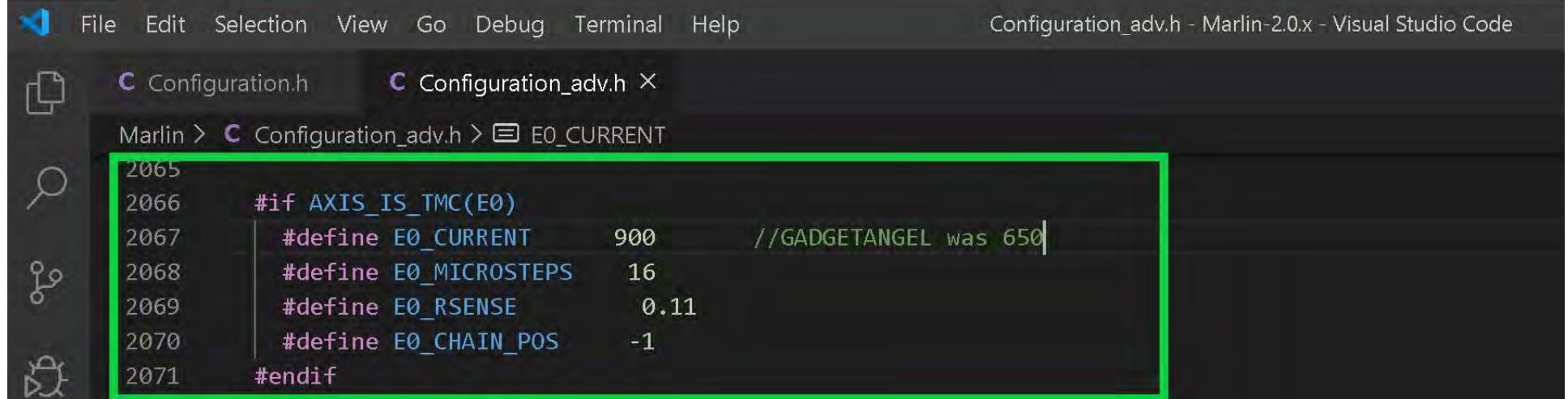
```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035 #define Z_CURRENT 760 //GADGETANGEL was 580
2036 #define Z_CURRENT_HOME Z_CURRENT
2037 #define Z_MICROSTEPS 16
2038 #define Z_RSENSE 0.11
2039 #define Z_CHAIN_POS -1
2040 #endif

```



```

File Edit Selection View Go Debug Terminal Help
Configuration.h Configuration_adv.h X

Marlin > Configuration_adv.h > E0_CURRENT

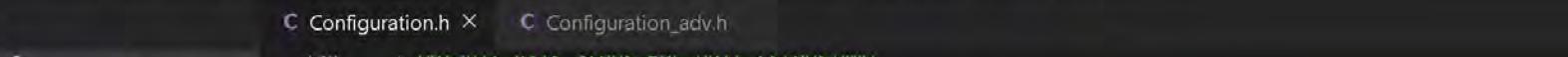
2065
2066 #if AXIS_IS_TMC(E0)
2067 #define E0_CURRENT 900 //GADGETANGEL was 650
2068 #define E0_MICROSTEPS 16
2069 #define E0_RSENSE 0.11
2070 #define E0_CHAIN_POS -1
2071 #endif

```

- Go to the next page.

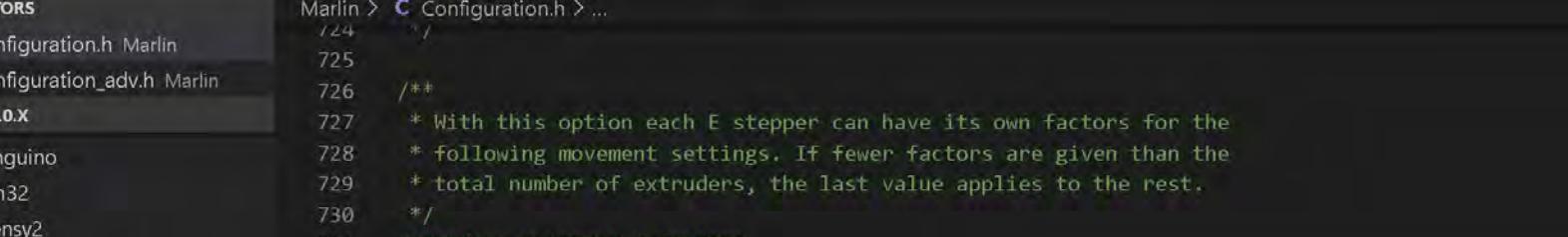
The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes



```
Configuration.h Configuration_adv.h
734 // DEFAULT_AXIS_STEPS_PER_UNIT (steps/mm)
735 * Override with M92
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740
741
742 /**
```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT"** anytime our STEPS are **NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration_adv.h

Marlin > Configuration.h > ...
724 /*
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732 /**
733 *
734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */


```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h ×
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
Configuration.h Marlin Configuration_adv.h Marlin
Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
> sanguino
> stm32
> teensy2
> teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
> sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default, it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):** Shows the project structure under MARLIN-2.0.X, including files like sanguino, stm32, teensy2, teensy3, pins.h, pinsDebug\_list.h, pinsDebug.h, sensitive\_pins.h, sd, MarlinCore.cpp, MarlinCore.h, \_Bootscreen.h, \_Statusscreen.h, Configuration\_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, and .gitignore.
- Central Area:** The file Configuration\_adv.h is open, showing code related to HYBRID\_THRESHOLD. A green box highlights the section of code defining various hybrid thresholds.

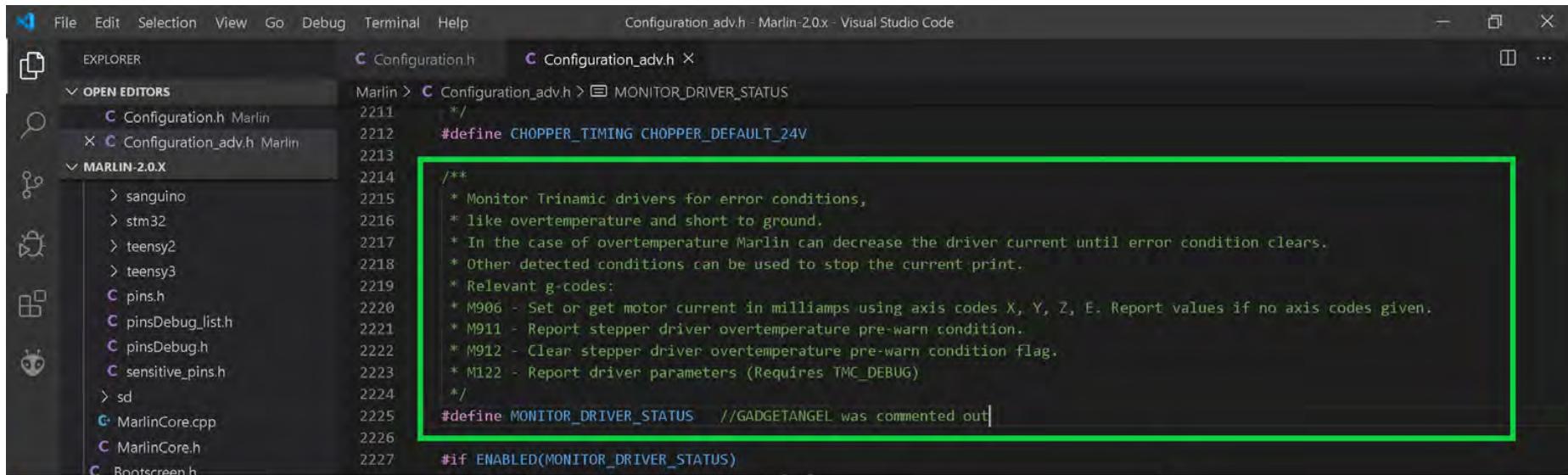
```
/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //GADGETANGEL was commented out

#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30
```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line.](#)



File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

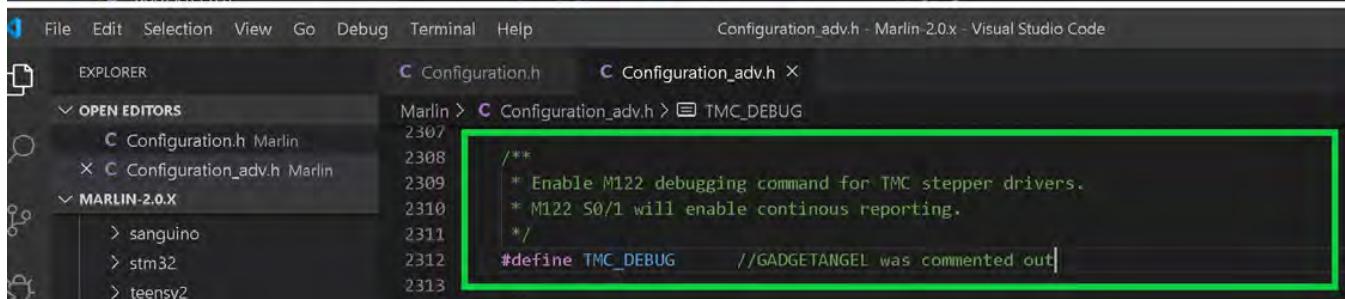
EXPLORER Configuration.h Configuration\_adv.h

OPEN EDITORS Marlin > Configuration\_adv.h > MONITOR\_DRIVER\_STATUS

```

2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration\_adv.h

OPEN EDITORS Marlin > Configuration\_adv.h > TMC\_DEBUG

```

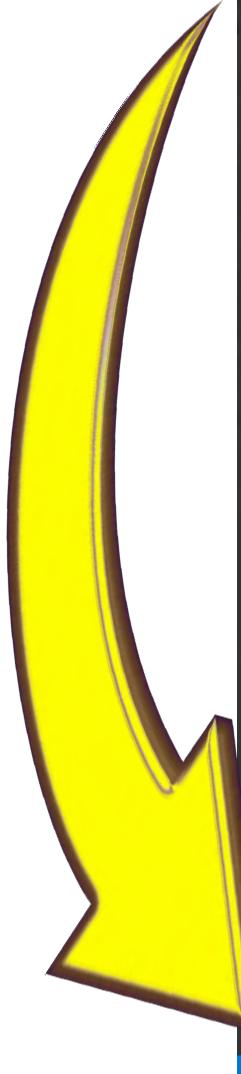
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- The end of Marlin setup for BIQU TMC2208 V3.0 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



Configuration.h - Marlin-2.0.x - Visual Studio Code

**EXPLORER**

- OPEN EDITORS
  - Configuration.h Marlin
  - pins\_BTT\_SKR\_V1\_3.h Marlin\src\pins\
  - pins\_BTT\_SKR\_common.h Marlin\src\pins\
  - Configuration\_adv.h Marlin
- MARLIN-2.0.X
  - samd
  - sanguino
  - stm32f1
  - stm32f4
  - stm32f7
  - teensy2
  - teensy3
  - pins.h
  - pinsDebug.h
  - pinsDebug.list.h
  - sensitive\_pins.h
  - sd
  - MarlinCore.cpp
  - MarlinCore.h
  - \_Bootscreen.h
  - \_Statusscreen.h
  - Configuration.h
  - Configuration\_adv.h
  - Makefile
  - Marlin.ino
  - Version.h
  - .editorconfig
  - .gitattributes
  - .gitignore
  - LICENSE
  - platformio.ini
  - process-palette.json
  - README.md

**EDITOR**

```

C Configuration.h X C pins_BTT_SKR_V1_3.h C pins_BTT_SKR_common.h C Configuration_adv.h
Marlin > C Configuration.h > ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 | #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

**PROBLEMS** 1 OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

```

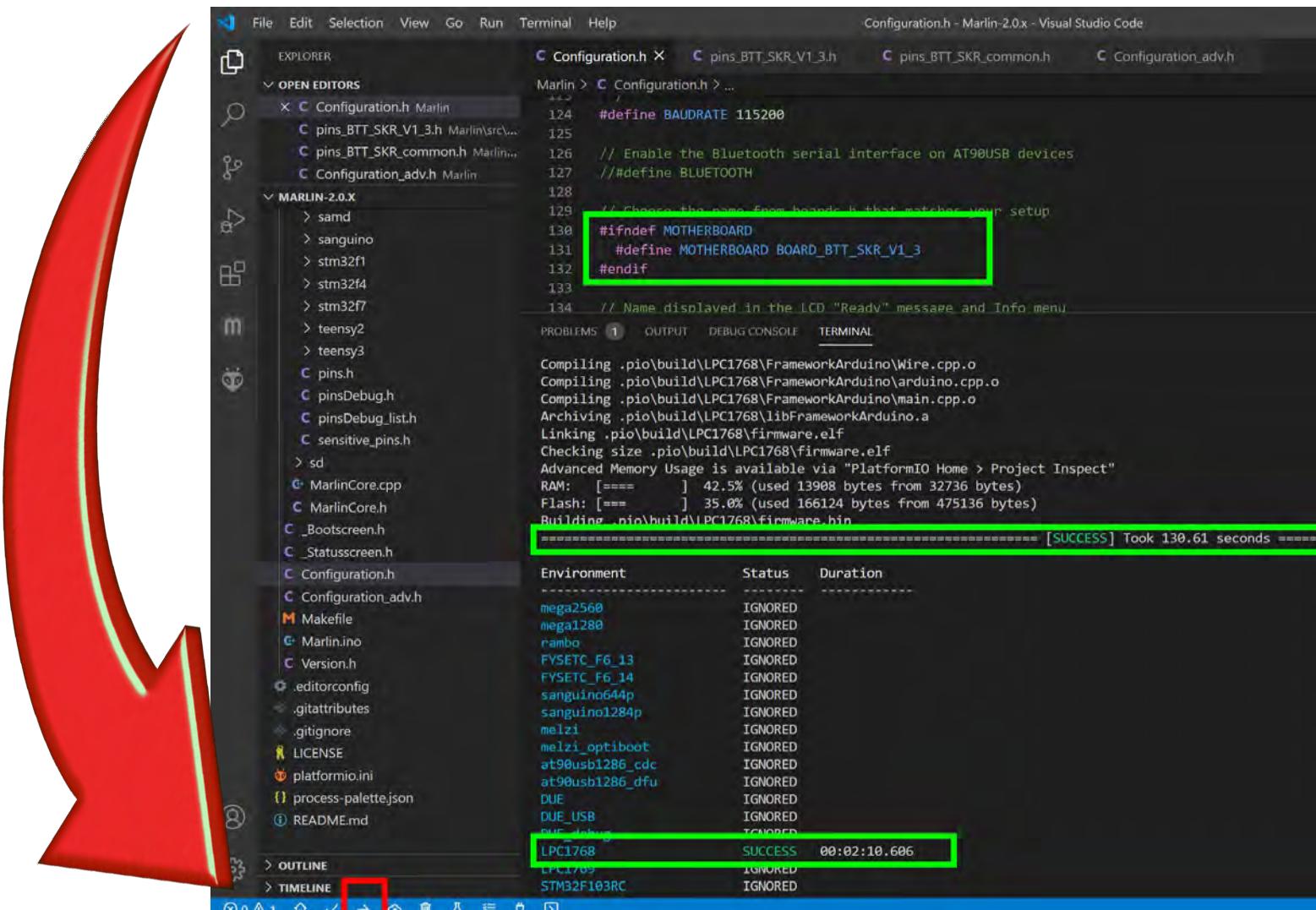
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET_duet3      | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

**OUTLINE** **TIMELINE**

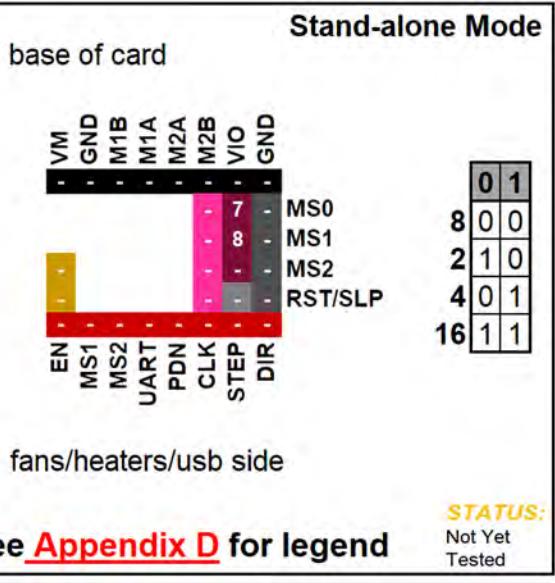
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2208 V3.0 Drivers in UART Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

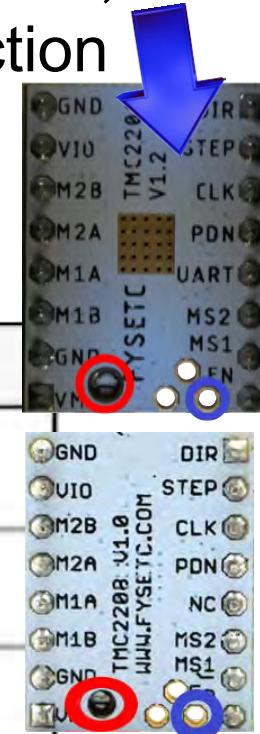


- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**FYSETC TMC2208 V1.2**Stand-alone Mode

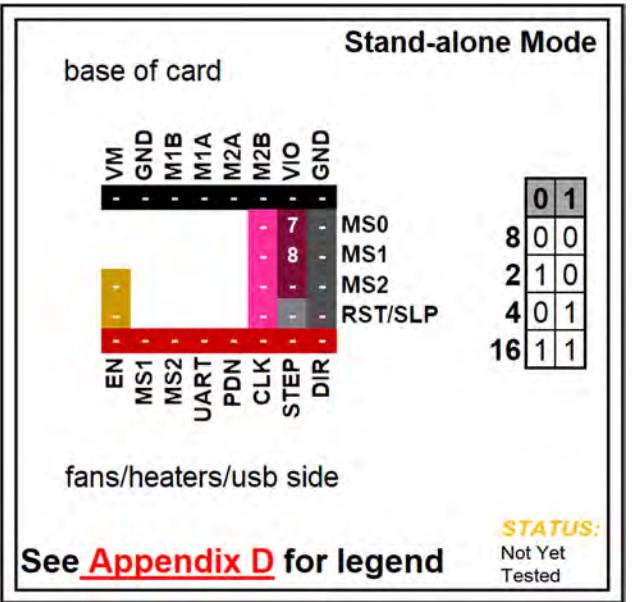
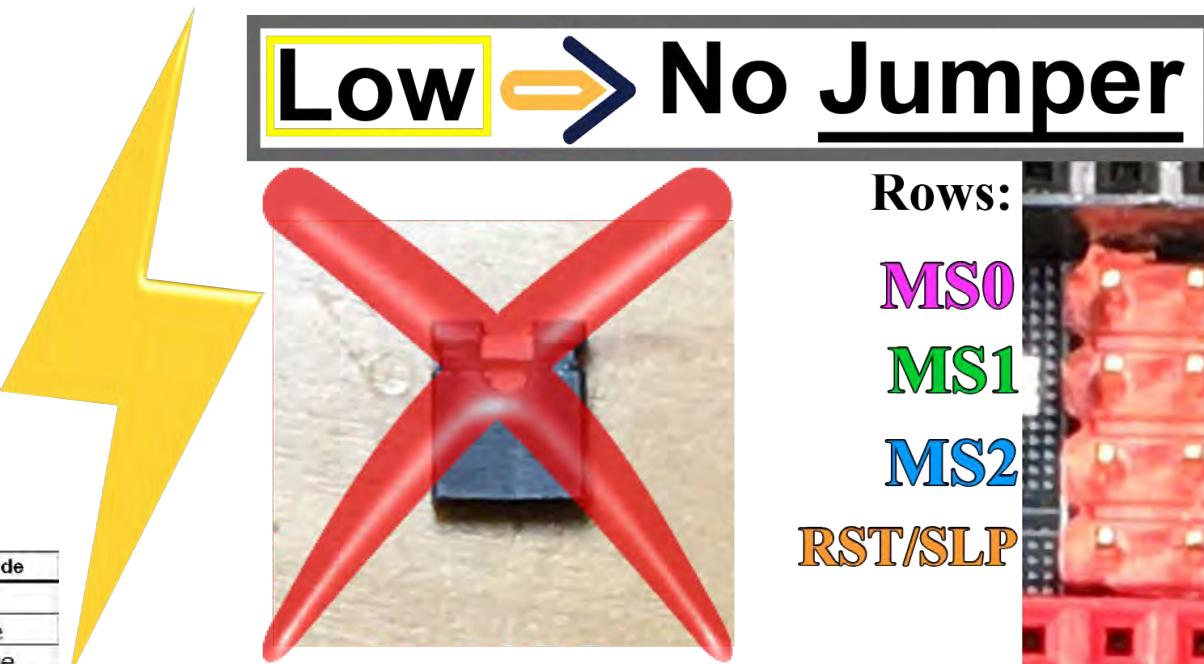
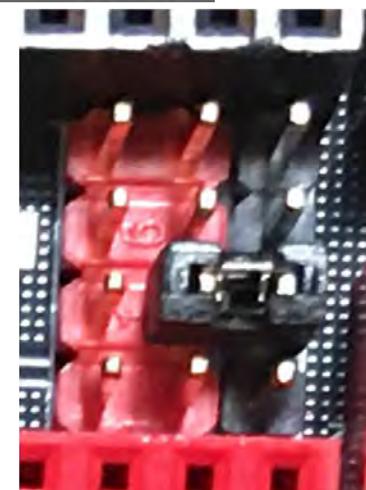
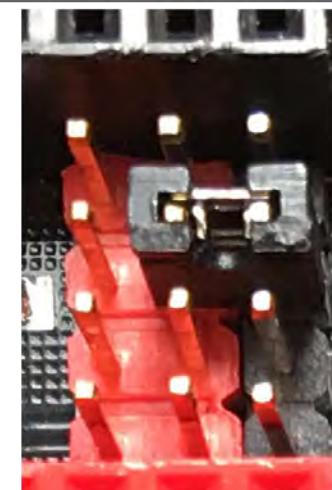
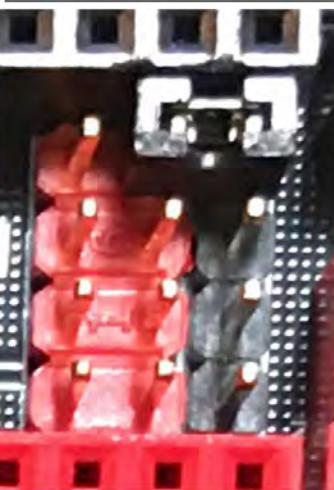
**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See **Appendix A** for instruction on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.



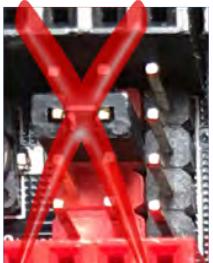
| Driver Chip                                                                                          | MS0                                                                                                                        | MS1  | Steps  | Interpolation                                                                                                              | Mode        |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|------|--------|----------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>FYSETC</b><br><b>TMC2208</b><br>Stand Alone Mode<br>Maximum 16 Subdivision<br>35V DC<br>2A (peak) | Low                                                                                                                        | Low  | 1 / 8  | 1 / 256                                                                                                                    | stealthChop |
|                                                                                                      | High                                                                                                                       | Low  | 1 / 2  | 1 / 256                                                                                                                    | stealthChop |
|                                                                                                      | Low                                                                                                                        | High | 1 / 4  | 1 / 256                                                                                                                    | stealthChop |
|                                                                                                      | High                                                                                                                       | High | 1 / 16 | 1 / 256                                                                                                                    | stealthChop |
| <b>Driving Current Calculation Formula</b><br><br>$R_S$ (Typical Sense Resistor) = 0.11Ω             | $I_{MAX} = V_{ref} * 0.9286$<br>See Appendix B #3. Use 50% to 90% as shown below:<br>$I_{MAX} = (V_{ref} * 0.9286) * 0.90$ |      |        | $V_{ref} = I_{MAX} * 1.0769$<br>See Appendix B #3. Use 50% to 90% as shown below:<br>$V_{ref} = (I_{MAX} * 1.0769) * 0.90$ |             |

- See next page for the LEGEND that belongs to the above Driver Chip Chart.

**FYSETC TMC2208 V1.2****Stand-alone Mode****SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers****Rows:****MS0****MS1****MS2****RST/SLP****Rows:****MS0****MS1****MS2****RST/SLP****Driver Chip Chart:**

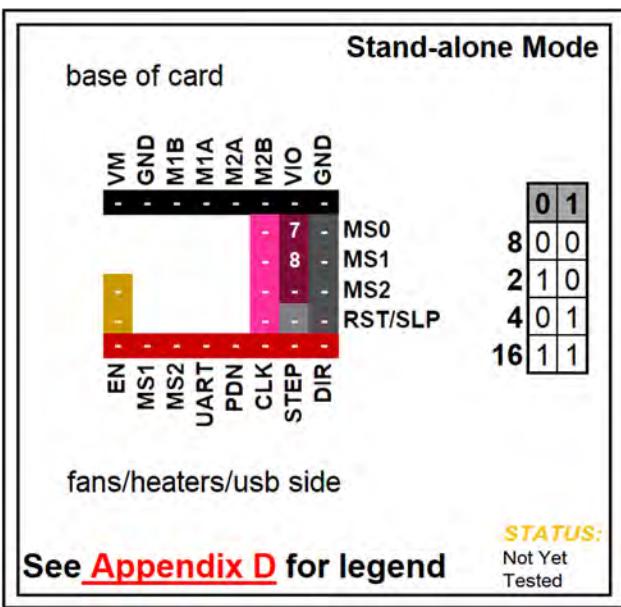
| Driver Chip                           | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|---------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b>                           | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
| Chip Name                             | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision               | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
| XX V DC xxA (peak)                    | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                       | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                       | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                       | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                       | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula   | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |
| $R_S$ (Typical Sense Resistor)=X.XX Ω |                            |      |                            |                      |                 |

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):





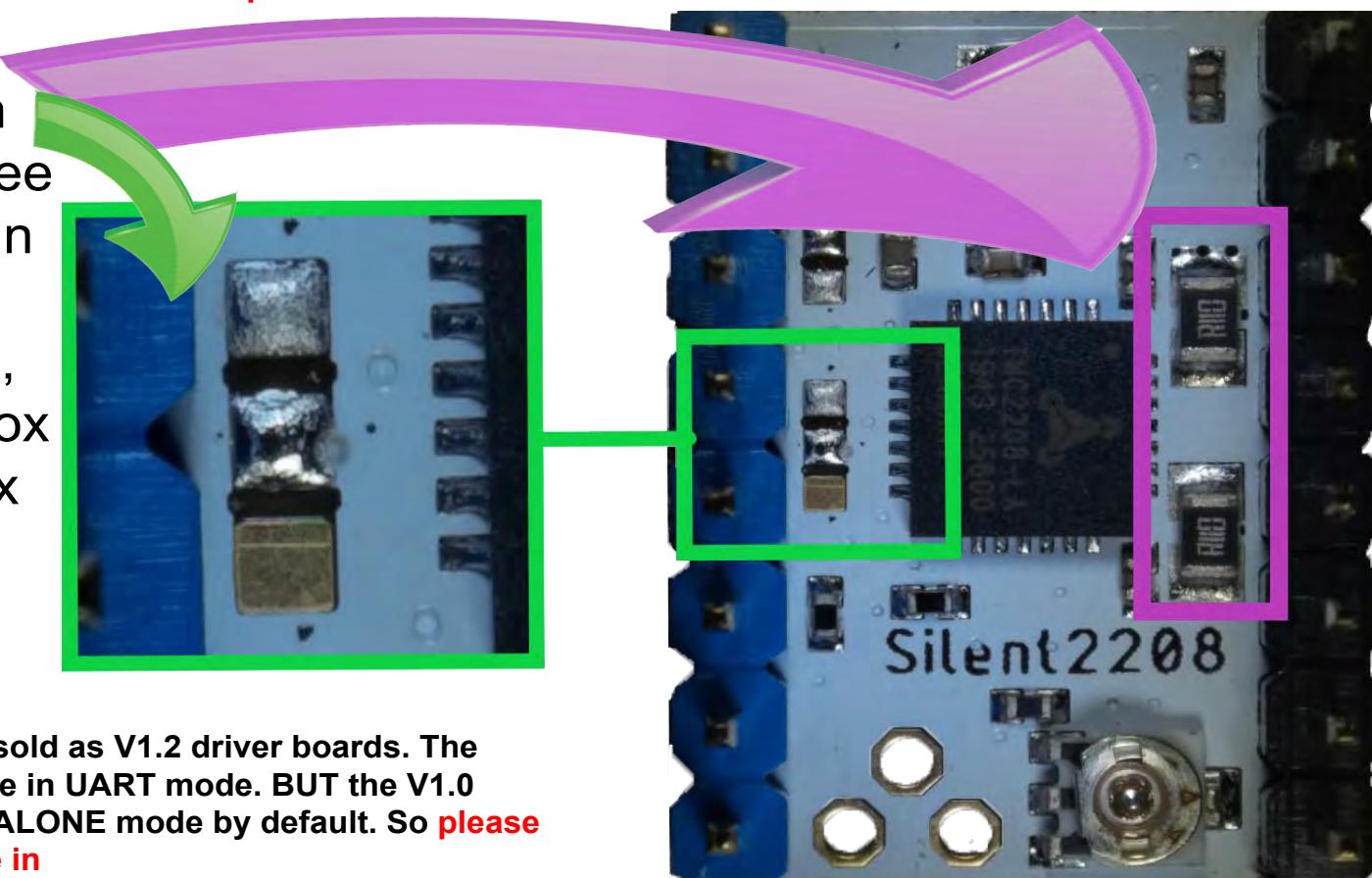
Again, a gap MUST be in place between one or all (three or two) "J2" pads to obtain Standalone Mode for the FYSETC TMC2208 V1.X, as seen in the **GREEN** box below. The **PURPLE** box shows the location of the current sense resistors ( $R_s$ ).

MOST FYSETC TMC2208 boards are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. So **please check your boards to ensure they are in the correct mode!!**

## FYSETC TMC2208 V1.2

### Stand-alone Mode

**Note:** To determine if your driver board is in UART mode, check the bottom of the driver board for three pads, located in the same position as shown in the picture below. There could be two or three pads located at this position (let's call it the "J2" position). To have the driver board in **STANDALONE MODE**, **ALL the pads at "J2" MUST NOT be connected.**

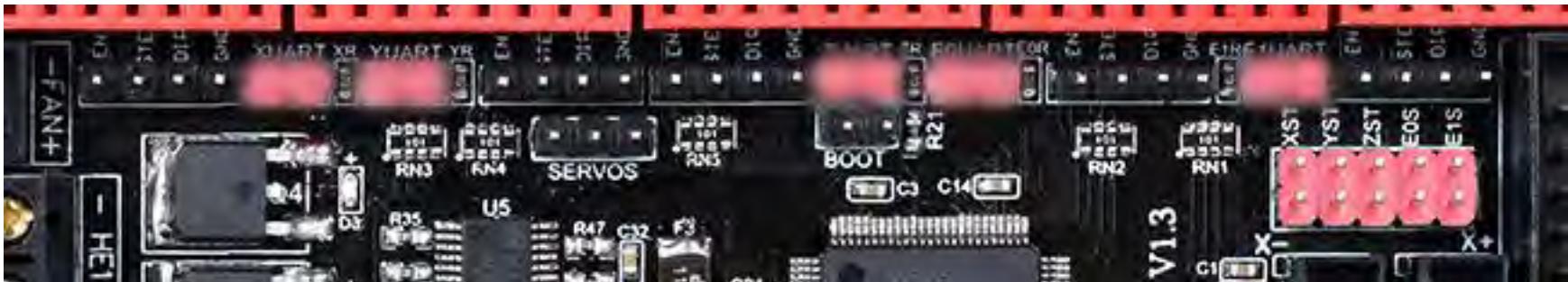
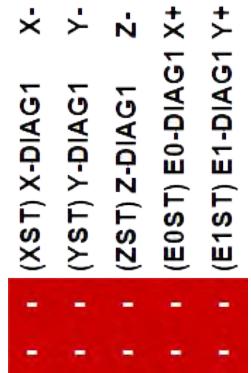


**FYSETC TMC2208 V1.2 for Standalone Mode**

FYSETC TMC2208 V1.2Stand-alone Mode

# Stand-alone Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.

XUART



YUART



ZUART



E0 UART



E1 UART



# Stand-alone Mode

## FYSETC TMC2208 V1.2

### Stand-alone Mode

#### **SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers**

##### Driver Socket Representation

Columns: **1 2 3**

Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

- DIAG

E  
N

**1/2**

##### Driver Chip Chart:

| Driver Chip                                        | MS0                        | MS1  | MS2                        | Excitation Mode |
|----------------------------------------------------|----------------------------|------|----------------------------|-----------------|
| LOGO                                               | Low                        | Low  | Low                        | 1-Phase         |
| Chip Name                                          | High                       | Low  | Low                        | W1-2 Phase      |
| Maximum XXX                                        | Low                        | High | Low                        | 2W1-2 Phase     |
| Subdivision                                        | Low                        | Low  | High                       | 4W1-2 Phase     |
| XXV DC                                             | Low                        | High | High                       | SW1-2 Phase     |
| XXA (peak)                                         | High                       | High | High                       | 16W1-2 Phase    |
| Driving Current Calculation Formula                | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                 |
| $R_g$ (typical) Series Resistor (max XX $\Omega$ ) |                            |      |                            |                 |

#### **MS0 for Binary State Drivers:**

Driver Socket Representation:

**7 7**

Driver Chip Chart:

**High**

Driver Socket Representation:

**--**

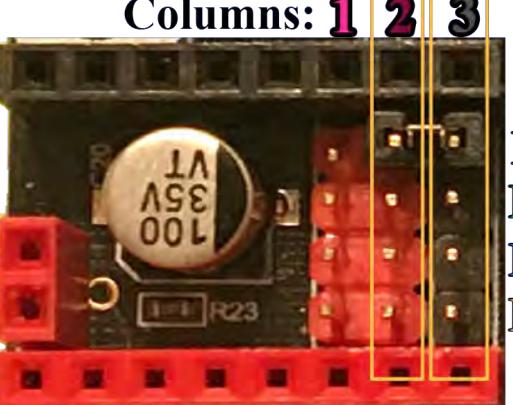
Driver Chip Chart:

**Low**

#### **High:**

Columns: **1 2 3**

Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**



#### Meaning:

Driver Chip Chart:

**High**

**set Jumper between column 2 and column 3 on the MS0 row**

**Low**

**No Jumper Set**

# Stand-alone Mode

## FYSETC TMC2208 V1.2

### Stand-alone Mode

#### Driver Socket Representation

Columns: **1 2 3**

Rows:  
**MS0**  
**MS1**

**MS2**

**RST/SLP**

**DIAG**

**1/4**

#### Driver Chip Chart:

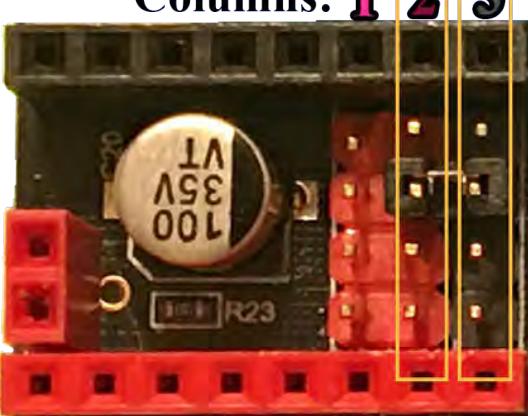
| Driver Chip | MS0  | MS1  | MS2  | Min current (threshold) | Excitation Mode |
|-------------|------|------|------|-------------------------|-----------------|
| LOGO        | Low  | Low  | Low  | 1.5 Step                | 2 Phase         |
| Chip Name   | Low  | High | Low  | 1.5 Step                | 1-2 Phase       |
| Maximum XXX | High | High | Low  | 1.5 Step                | W1-2 Phase      |
| Subdivision | Low  | High | High | 1.5 Step                | 2W1-2 Phase     |
| XXV DC      | High | High | High | 1.5 Step                | 4W1-2 Phase     |
| xxA (peak)  | Low  | High | High | 1.5 Step                | 8W1-2 Phase     |
|             | High | High | High | 1.5 Step                | 16W1-2 Phase    |
|             | High | High | High | 1.5 Step                | 32W1-2 Phase    |

Driving Current  
Calculation  
Formula  
 $I_{MAX} = \text{FORMULA}$  |  $V_{ref} = \text{FORMULA}$

#### High:

Columns: **1 2 3**

Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**



#### MS1 for Binary State Drivers:

Driver Socket Representation:

- -

Driver Chip Chart:

**Low**

Driver Socket Representation:

**8 8**

Driver Chip Chart:

**High**

Driver Socket Representation:

- -

Driver Chip Chart:

**Low**

#### Meaning:

Driver Chip Chart:

**High**

set Jumper between  
column **2** and column **3**  
on the **MS1** row

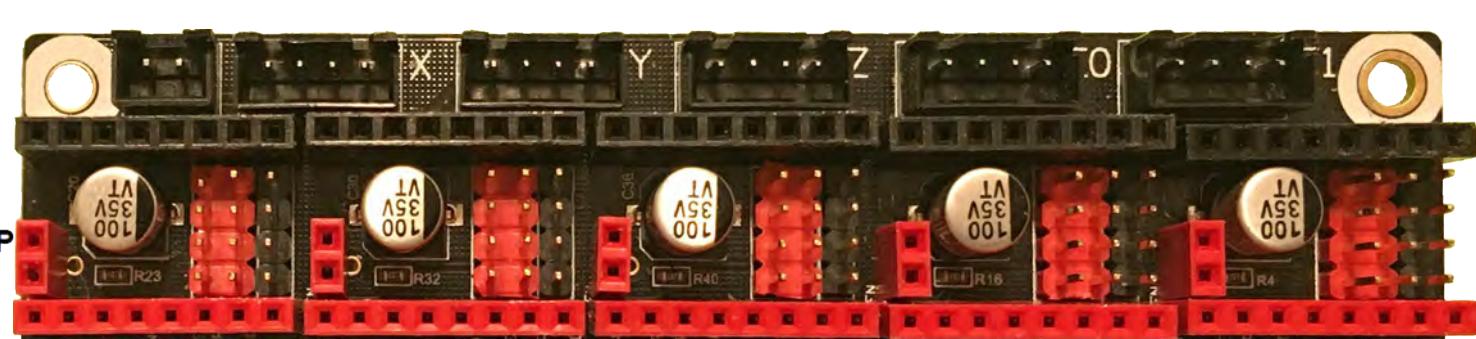
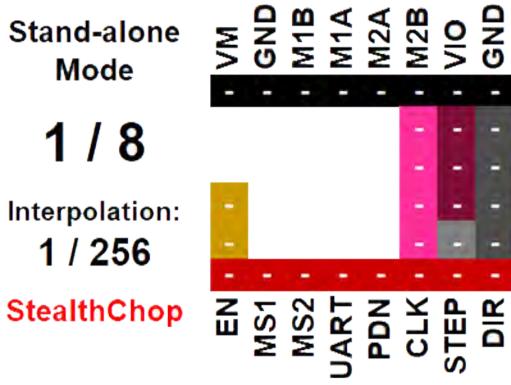
**Low**

No Jumper Set

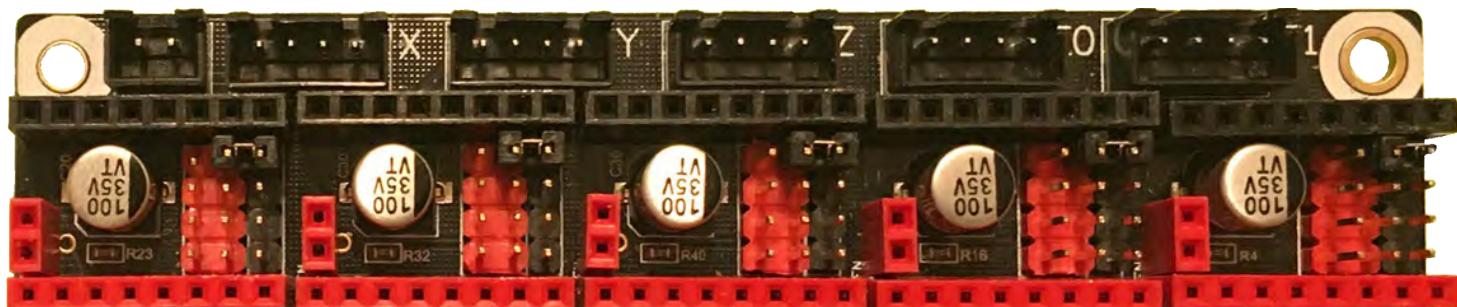
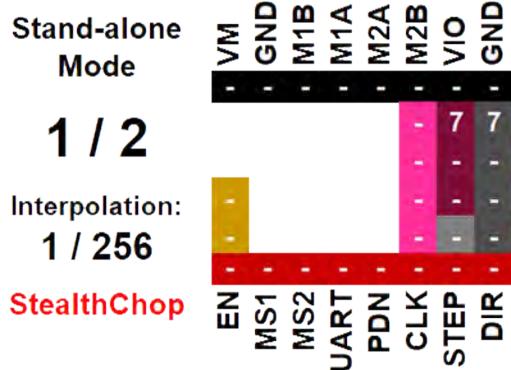
# Stand-alone Mode

## FYSETC TMC2208 V1.2

### Stand-alone Mode



See [Appendix D](#) for legend



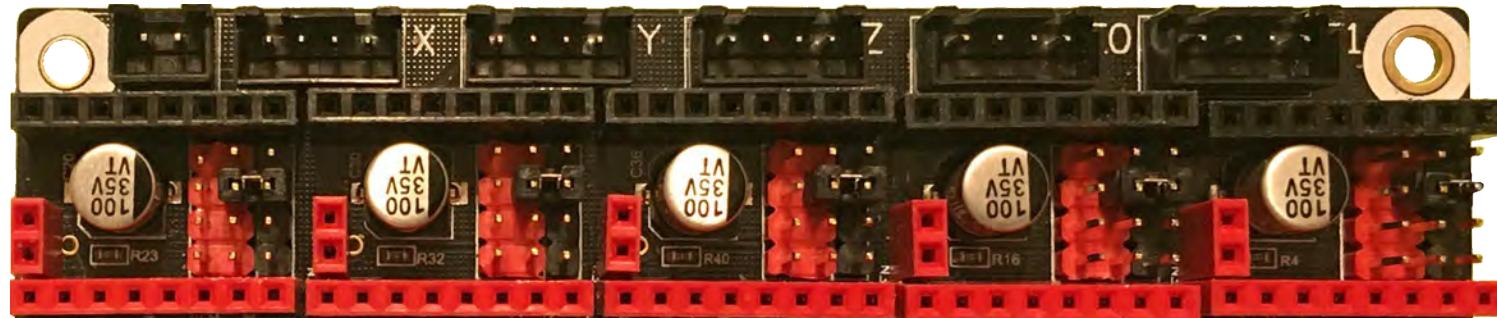
See [Appendix D](#) for legend

# Stand-alone Mode

## FYSETC TMC2208 V1.2

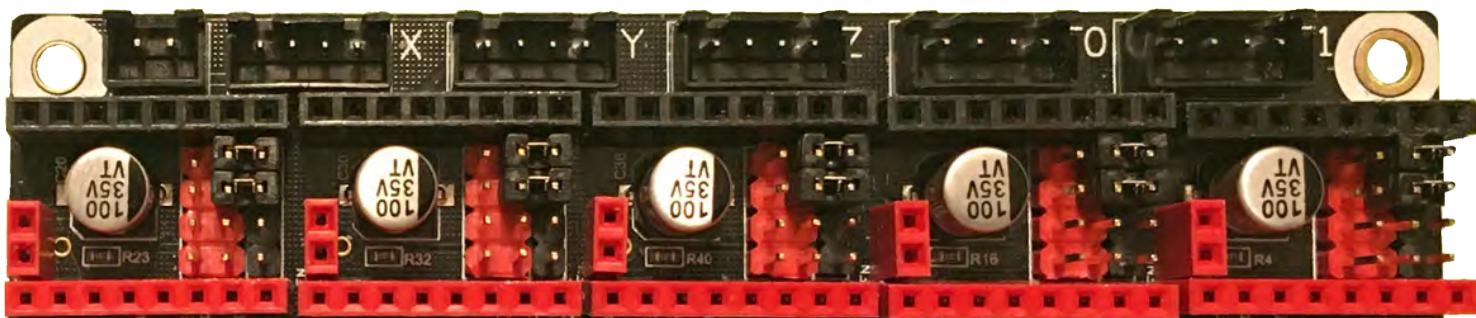
### Stand-alone Mode

| Stand-alone Mode | VM | GND | M1B | M1A  | M2A | M2B | VIO  | GND |
|------------------|----|-----|-----|------|-----|-----|------|-----|
| <b>1 / 4</b>     | -  | -   | -   | -    | -   | -   | MS0  | -   |
| Interpolation:   | 1  | /   | 256 | -    | -   | -   | MS1  | -   |
|                  | -  | -   | -   | -    | -   | -   | MS2  | -   |
| StealthChop      | EN | MS1 | MS2 | UART | PDN | CLK | STEP | DIR |



See [Appendix D](#) for legend

| Stand-alone Mode | VM | GND | M1B | M1A  | M2A | M2B | VIO  | GND |
|------------------|----|-----|-----|------|-----|-----|------|-----|
| <b>1 / 16</b>    | -  | -   | -   | -    | -   | -   | MS0  | -   |
| Interpolation:   | 1  | /   | 256 | -    | -   | -   | MS1  | -   |
|                  | -  | -   | -   | -    | -   | -   | MS2  | -   |
| StealthChop      | EN | MS1 | MS2 | UART | PDN | CLK | STEP | DIR |

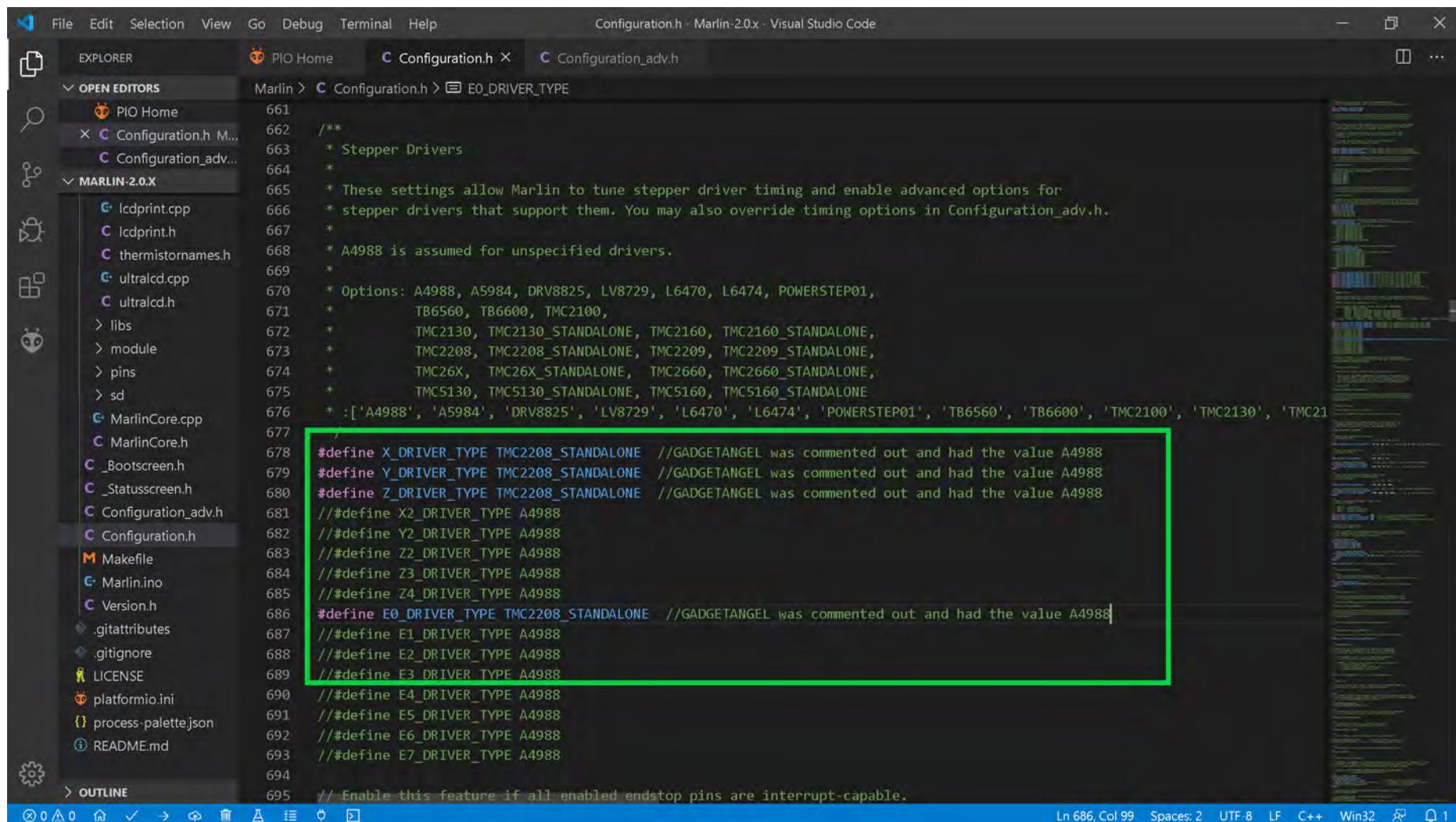


See [Appendix D](#) for legend

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for FYSETC TMC2208 V1.2 stepper motor drivers in stand-alone mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in stand-alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following configuration for stepper drivers:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE,
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208',
676 * TMC26X', 'TMC2660', 'TMC5130', 'TMC5160'],
677 */
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
681 //##define X2_DRIVER_TYPE A4988
682 //##define Y2_DRIVER_TYPE A4988
683 //##define Z2_DRIVER_TYPE A4988
684 //##define Z3_DRIVER_TYPE A4988
685 //##define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
687 //##define E1_DRIVER_TYPE A4988
688 //##define E2_DRIVER_TYPE A4988
689 //##define E3_DRIVER_TYPE A4988
690 //##define E4_DRIVER_TYPE A4988
691 //##define E5_DRIVER_TYPE A4988
692 //##define E6_DRIVER_TYPE A4988
693 //##define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

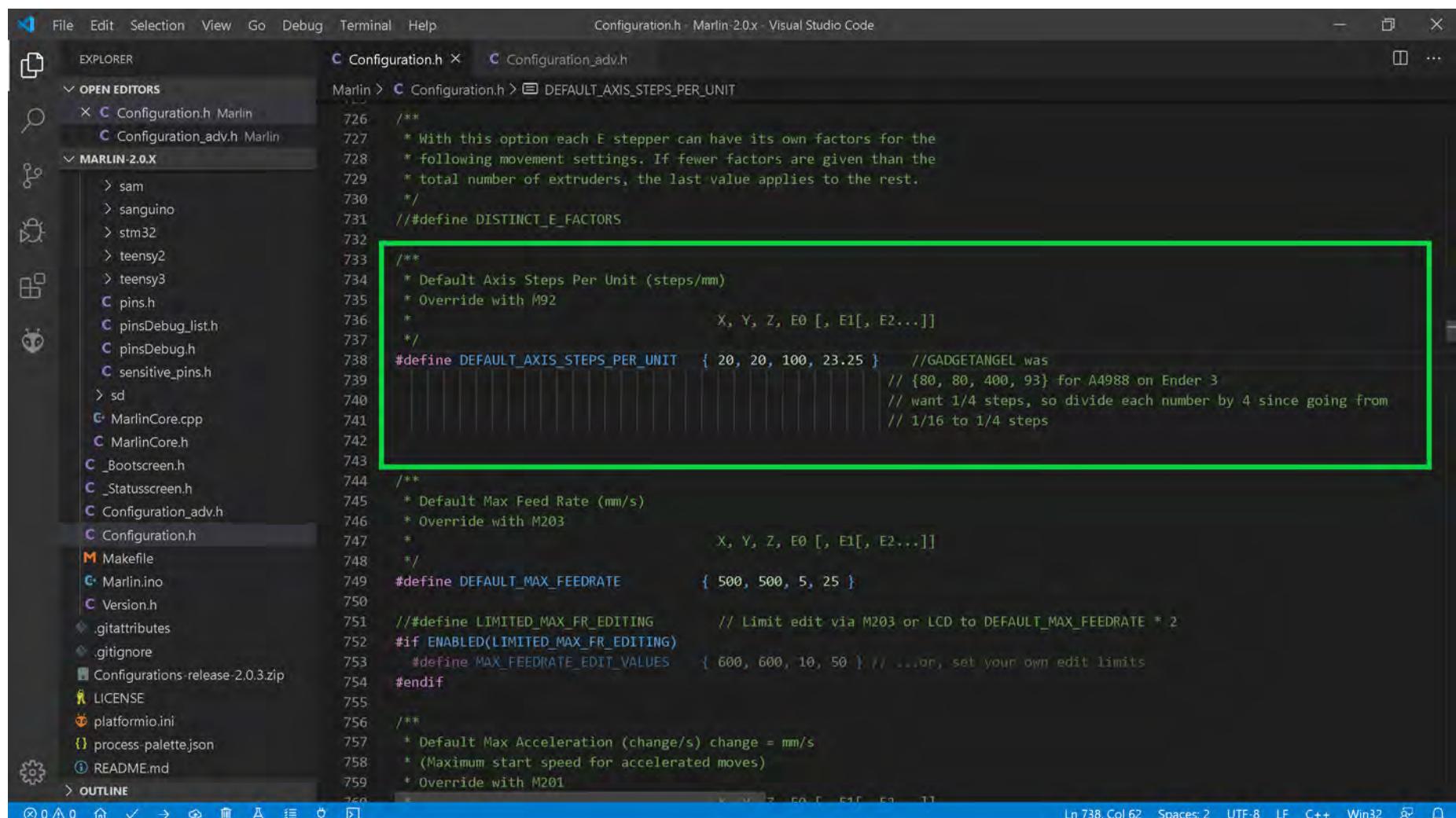
```

A green rectangular box highlights the driver type definitions for the X, Y, Z, and E0 axes, specifically the lines starting with `#define`. These lines were previously commented out with double slashes (//).

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x configuration header. A green rectangular box highlights the following line of code:

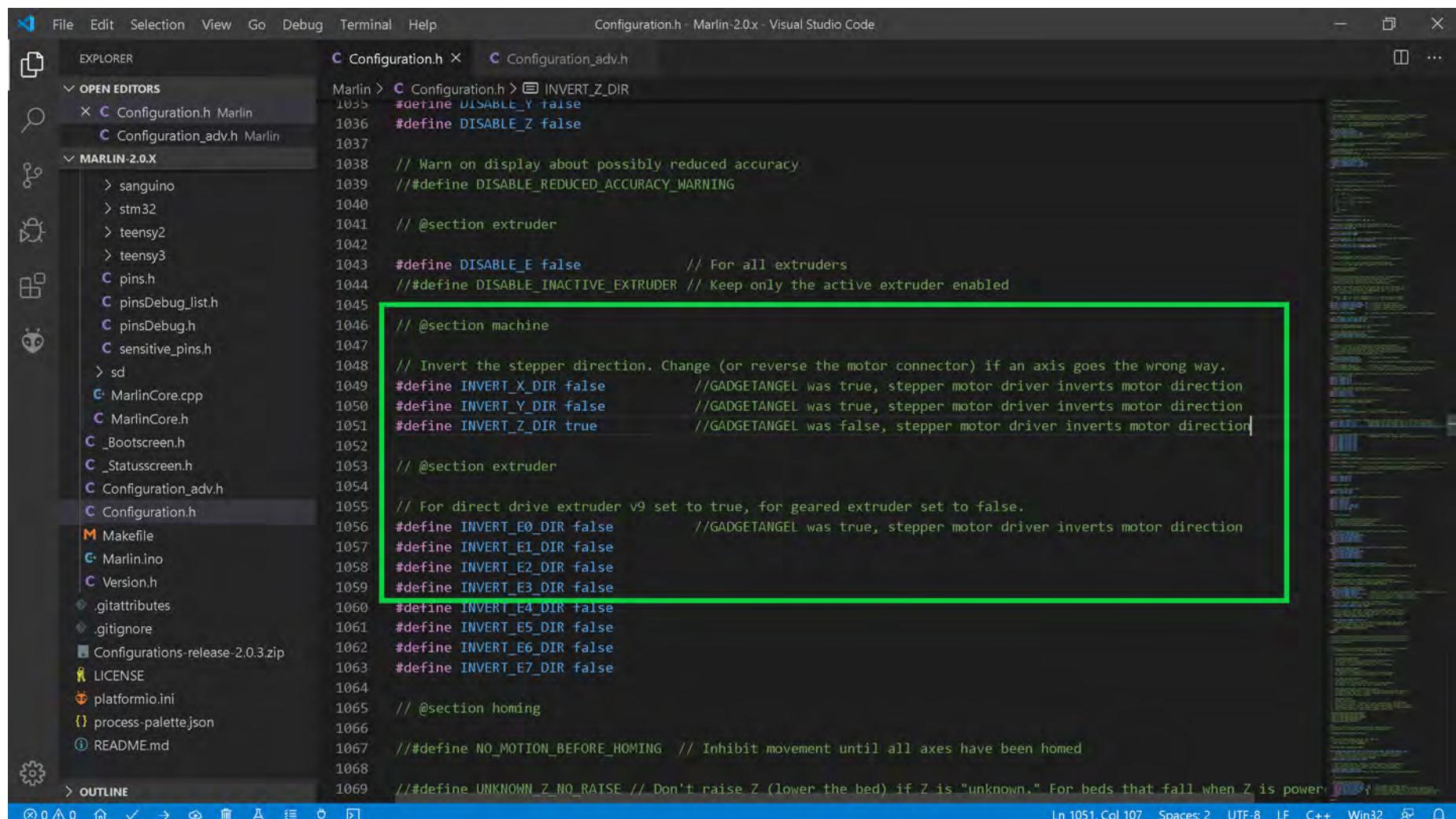
```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// want 1/4 steps, so divide each number by 4 since going from
// 1/16 to 1/4 steps
```

The code editor's status bar at the bottom right indicates: Ln 738, Col 62, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code is part of the Marlin 2.0.x repository. A green rectangular box highlights a section of the code where the `INVERT_X_DIR`, `INVERT_Y_DIR`, and `INVERT_Z_DIR` macros are defined. The original code in the image shows `INVERT_X_DIR` and `INVERT_Y_DIR` as `false` and `INVERT_Z_DIR` as `true`. Below this, there is a note explaining the context: `// GADGETANGEL was true, stepper motor driver inverts motor direction`.

```

1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 //#define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X

```

124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
126 // #define BLUETOOTH
127
128 // Choose the name from boards.h that matches your setup
129 #ifndef MOTHERBOARD
130 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
131
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin  
 ====== [SUCCESS] Took 130.61 seconds =====

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET            | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in Stand-alone Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

Configuration.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE

```

```

Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

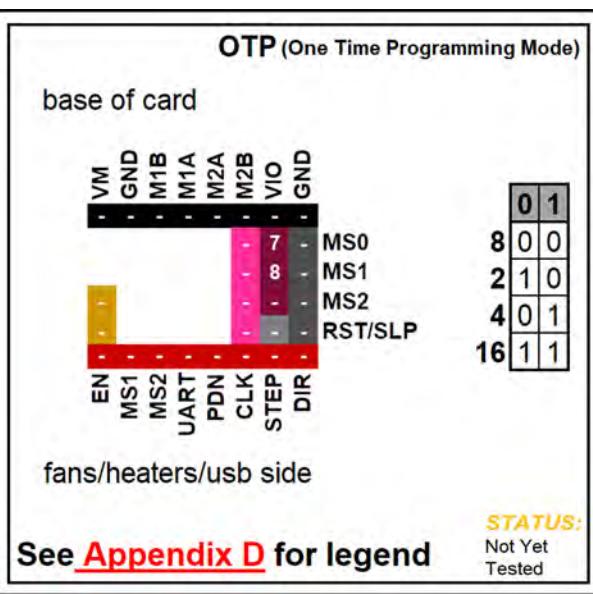
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET_3D         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

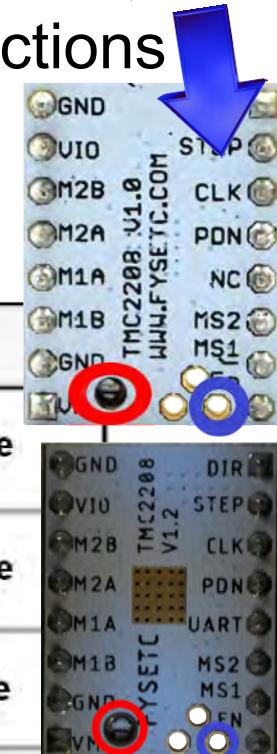


## FYSETC TMC2208 V1.2

### One Time Programming (OTP) Mode

**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See [Appendix A](#) for instructions on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.

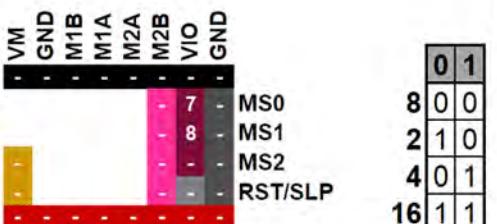


| Driver Chip                                                                                  | MS0                                                                                                                        | MS1  | Steps  | Interpolation                                                                                                              | Mode        |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|------|--------|----------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>FYSETC</b><br><b>TMC2208</b><br>OTP Mode<br>Maximum 16 Subdivision<br>35V DC<br>2A (peak) | Low                                                                                                                        | Low  | 1 / 8  | 1 / 256                                                                                                                    | spreadCycle |
|                                                                                              | High                                                                                                                       | Low  | 1 / 2  | 1 / 256                                                                                                                    | spreadCycle |
|                                                                                              | Low                                                                                                                        | High | 1 / 4  | 1 / 256                                                                                                                    | spreadCycle |
|                                                                                              | High                                                                                                                       | High | 1 / 16 | 1 / 256                                                                                                                    | spreadCycle |
| <b>Driving Current Calculation Formula</b><br><br>$R_S$ (Typical Sense Resistor) = 0.11Ω     | $I_{MAX} = V_{ref} * 0.9286$<br>See Appendix B #3. Use 50% to 90% as shown below:<br>$I_{MAX} = (V_{ref} * 0.9286) * 0.90$ |      |        | $V_{ref} = I_{MAX} * 1.0769$<br>See Appendix B #3. Use 50% to 90% as shown below:<br>$V_{ref} = (I_{MAX} * 1.0769) * 0.90$ |             |

- See next page for the LEGEND that belongs to the above Driver Chip Chart.

**OTP (One Time Programming Mode)**

base of card

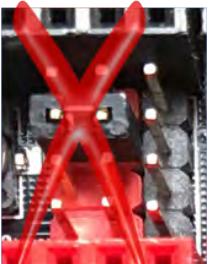


fans/heaters/usb side

See [Appendix D](#) for legend**Driver Chip Chart:**

| Driver Chip                            | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|----------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b>                            | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
| Chip Name                              | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision                | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
| XX V DC xxA (peak)                     | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                        | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                        | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                        | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                        | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula    | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |
| $R_S$ (Typical Sense Resistor)= x.xx Ω |                            |      |                            |                      |                 |

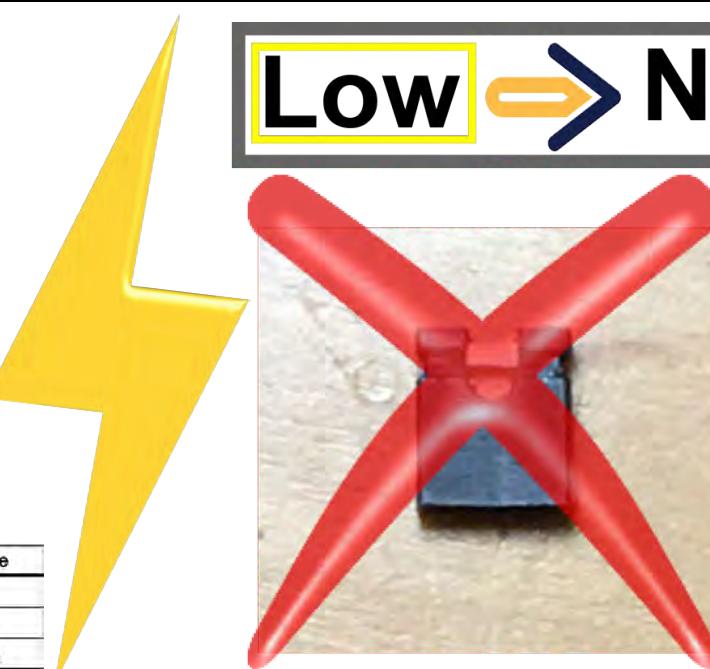
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

**FYSETC TMC2208 V1.2****One Time Programming (OTP) Mode****SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

**Low** → No Jumper

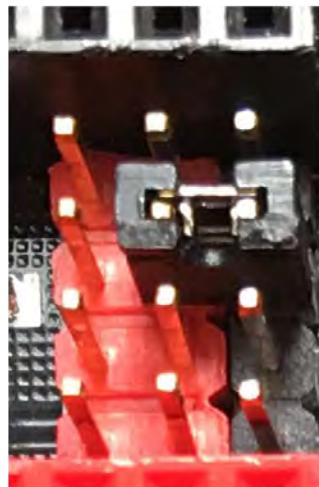
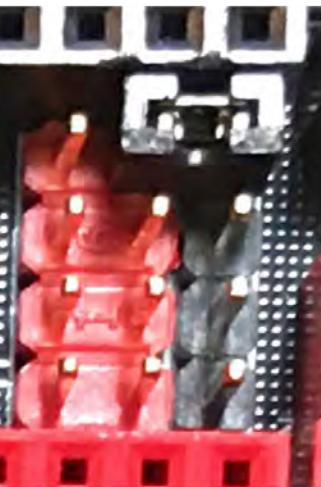


Rows:

**MS0****MS1****MS2****RST/SLP**

**High** → Jumper Set

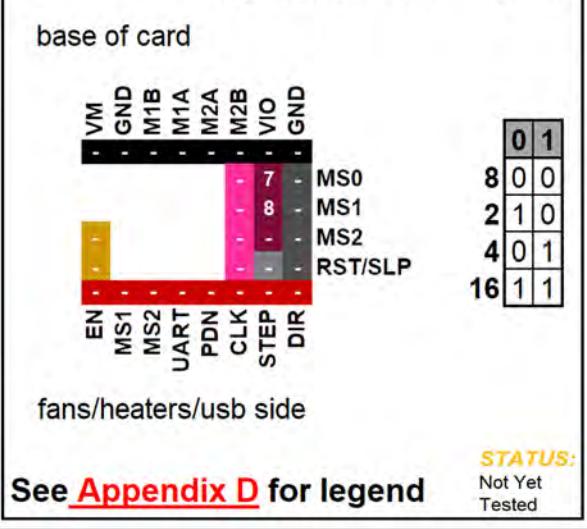
Rows:

**MS0****MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**

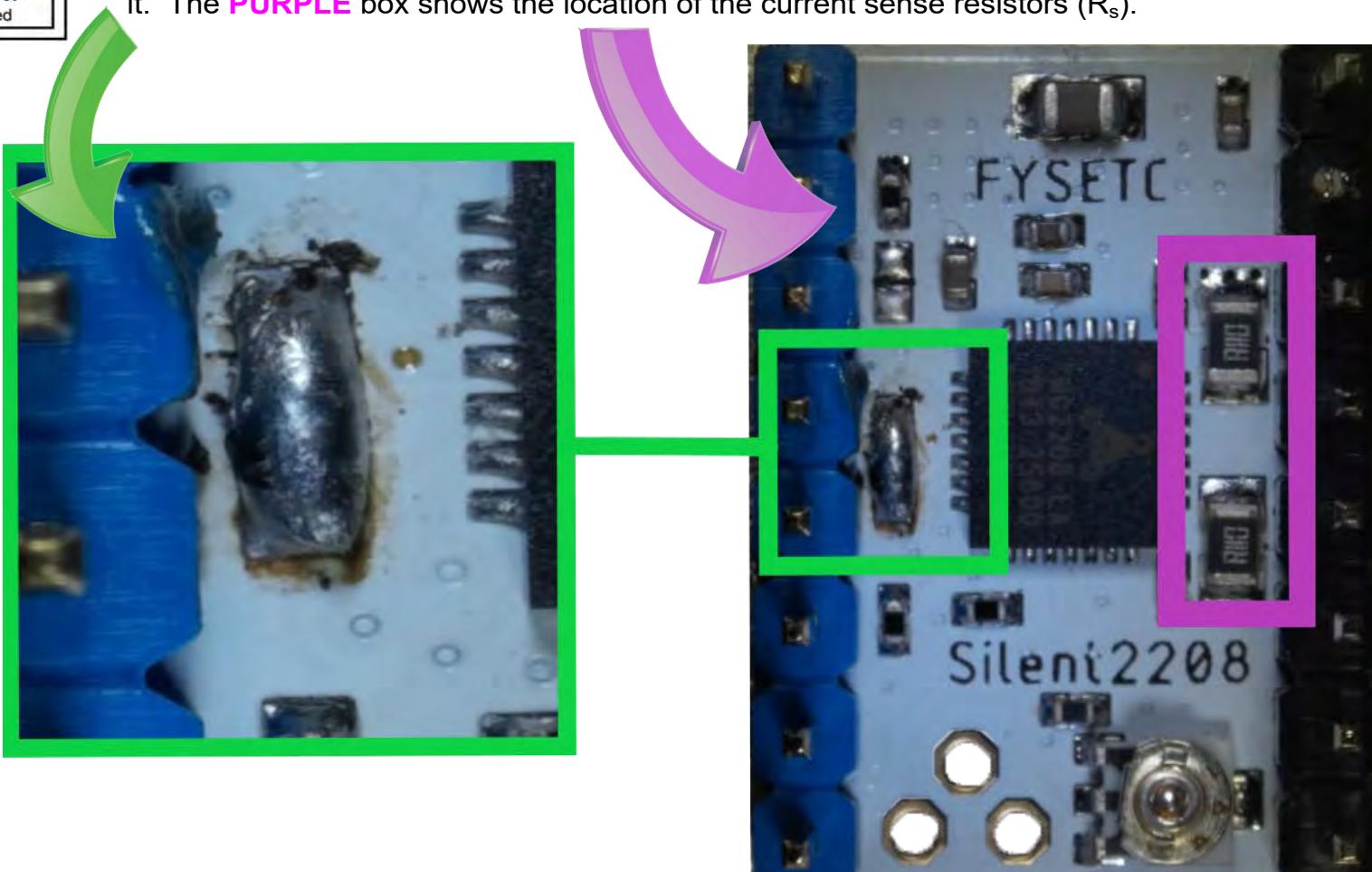
**FYSETC TMC2208 V1.2****One Time Programming (OTP) Mode**

**NOTE:** Stand-alone Mode by default uses StealthChop, if you **want SpreadCycle**, you **MUST** use **OTP mode**. Here are the directions for running the TMC220x Configurator: <https://wiki.fysetc.com/TMC2208/#to-run-the-program>. See TMC220x Configurator for One-Time-Programming Information: [TMC220x Configurator](#).

**Important:** To place FYSETC TMC2208 V1.0 or V1.2 **into OTP Mode** you **must solder all adjacent pads together** on the bottom of the driver board. As an example, the picture (V1.2) below shows all the pads soldered together, as shown in **GREEN**. To do One-Time-Programming (OTP), the TMC2208 must be placed in UART mode to program it. The **PURPLE** box shows the location of the current sense resistors ( $R_s$ ).

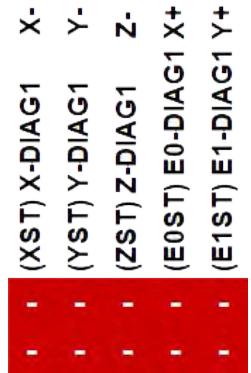


MOST FYSETC TMC2208 boards are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in STANDALONE mode by default. So **please check your boards to ensure they are in the correct mode you desire!** To be able to program the Chip for OTP mode, the FYSETC TMC2208 board **MUST** be set in **UART mode (all three pads bridged together)**!



**FYSETC TMC2208 V1.2****One Time Programming (OTP) Mode****OTP (One Time Programming) Mode**

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation.



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation.

XUART

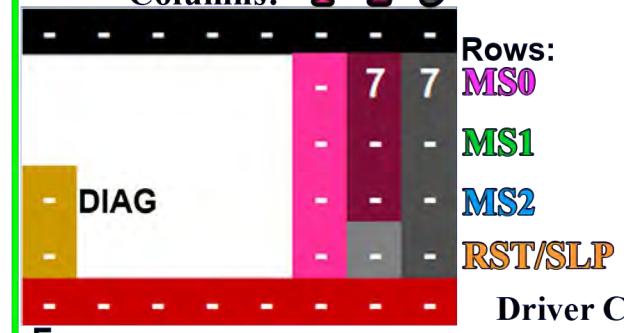
YUART

ZUART

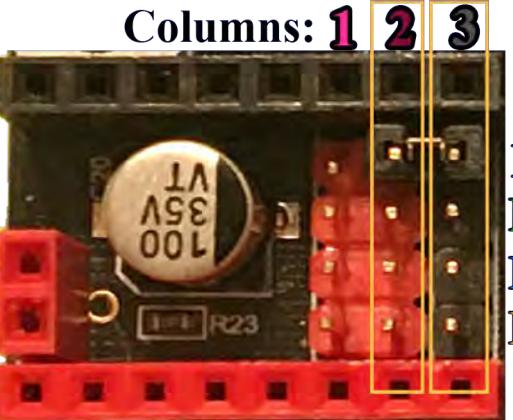
E0 UART

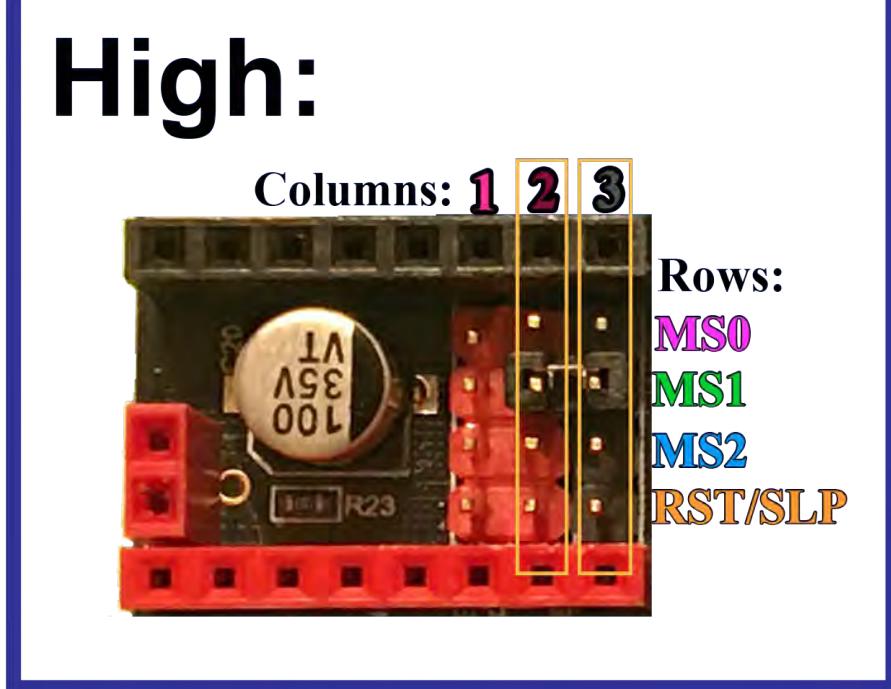
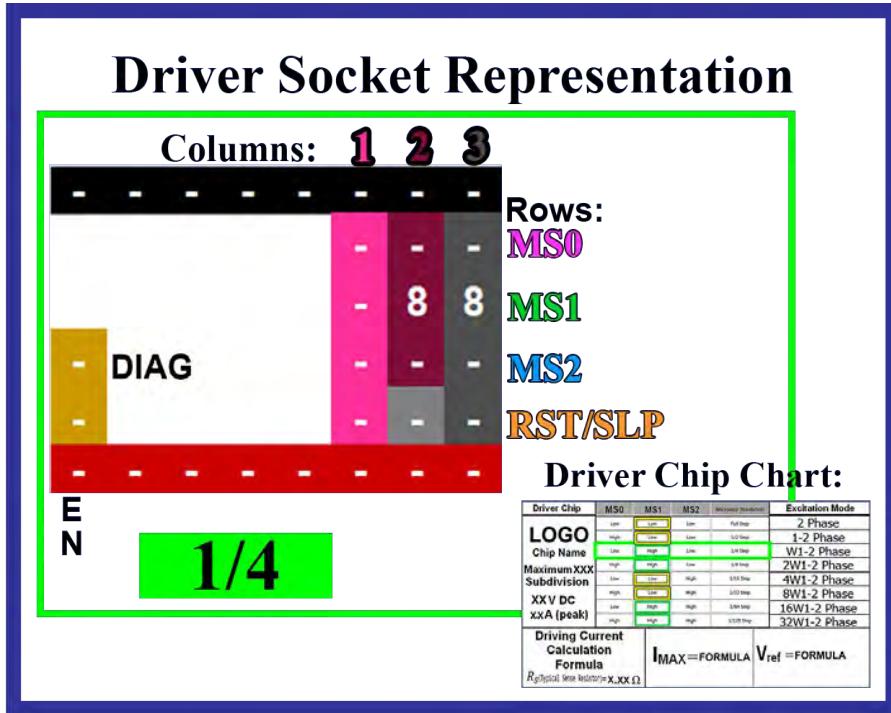
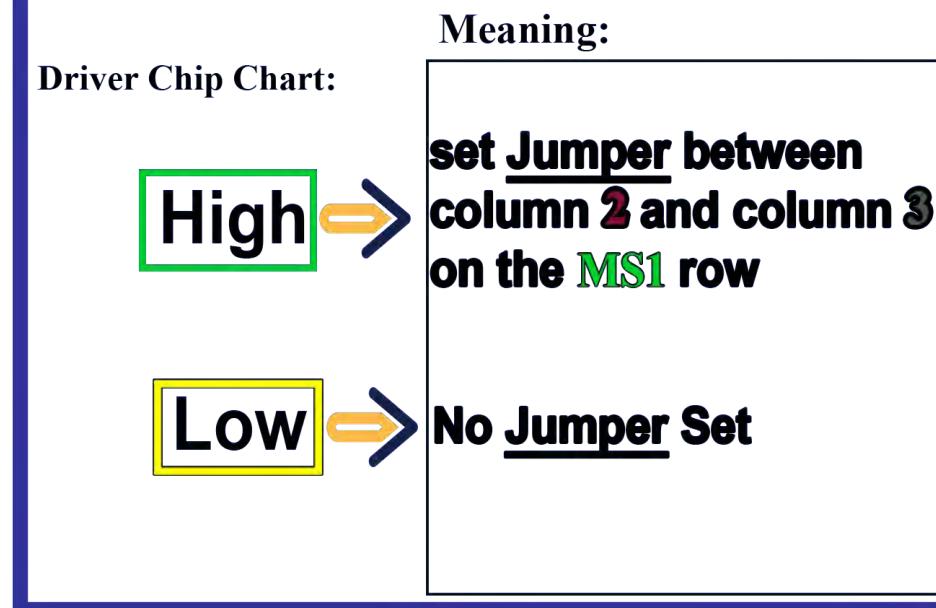
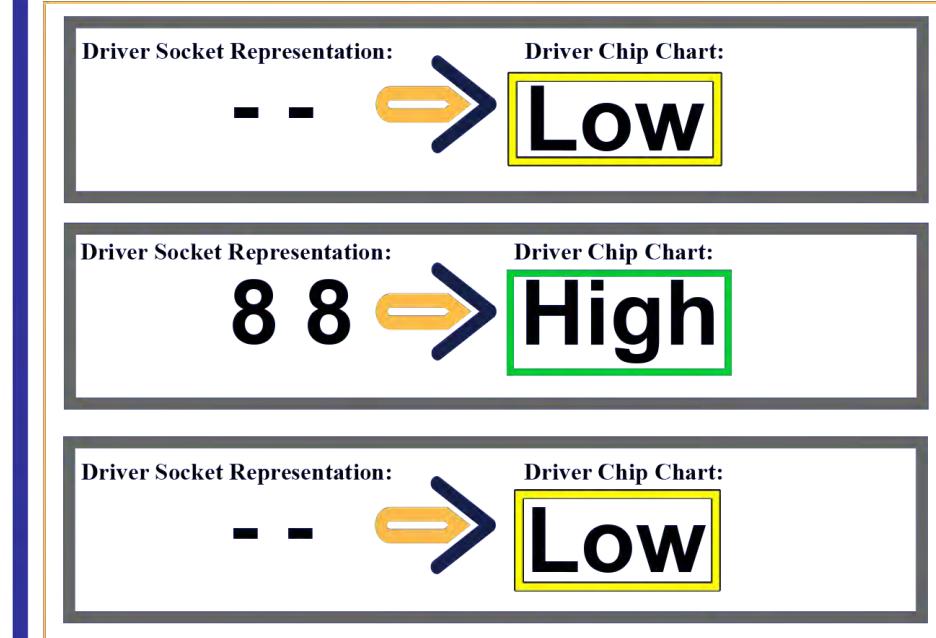
E1 UART



**FYSETC TMC2208 V1.2****One Time Programming (OTP) Mode****OTP (One Time Programming) Mode****SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers****Driver Socket Representation**Columns: **1 2 3****1/2****Driver Chip Chart:**

| Driver Chip                              | MS0                        | MS1  | MS2                        | Excitation Mode |
|------------------------------------------|----------------------------|------|----------------------------|-----------------|
| LOGO                                     | Low                        | Low  | Low                        | 1-Phase         |
| Chip Name                                | High                       | Low  | Low                        | W1-2 Phase      |
| Maximum XXX                              | Low                        | High | Low                        | 2W1-2 Phase     |
| Subdivision                              | Low                        | Low  | High                       | 4W1-2 Phase     |
| XXV DC                                   | High                       | Low  | High                       | SW1-2 Phase     |
| XXA (peak)                               | Low                        | High | High                       | 16W1-2 Phase    |
| Driving Current Calculation Formula      | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                 |
| $R_g$ (typical Series Resistor) $\Omega$ |                            |      |                            |                 |

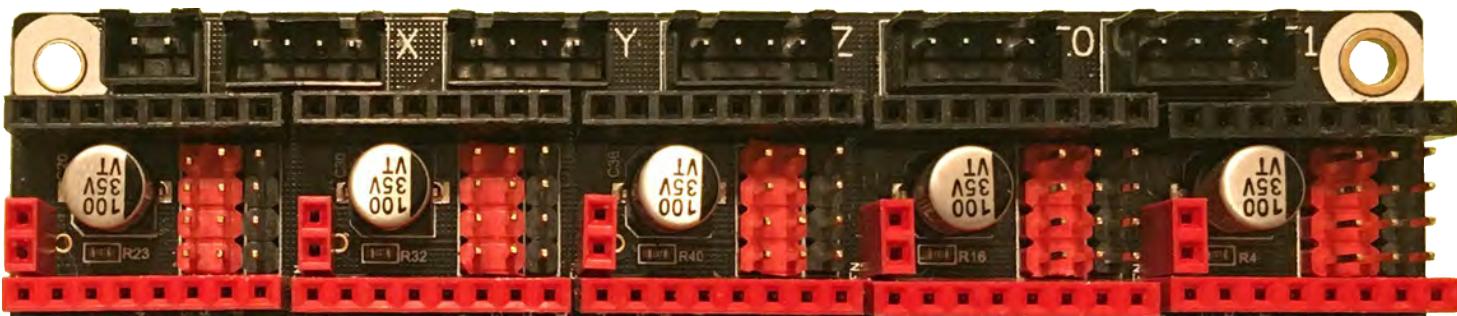
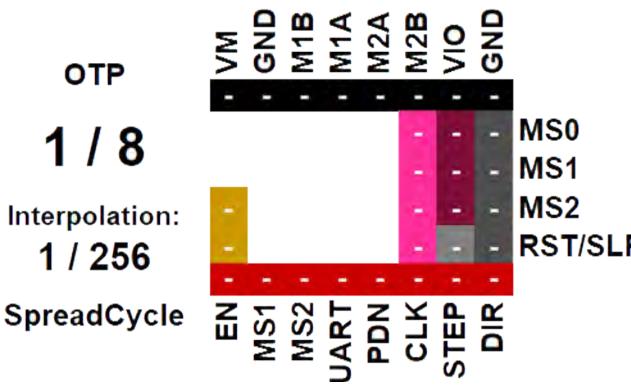
**MS0 for Binary State Drivers:****Driver Socket Representation:****7 7****Driver Chip Chart:****High****Driver Socket Representation:****--****Driver Chip Chart:****Low****High:**Columns: **1 2 3****Rows:**
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**
**Meaning:****Driver Chip Chart:****High**
**set Jumper between column 2 and column 3 on the MS0 row**
**Low**
**No Jumper Set**

**FYSETC TMC2208 V1.2**One Time Programming (OTP) Mode**OTP (One Time Programming) Mode****MS1 for Binary State Drivers:**

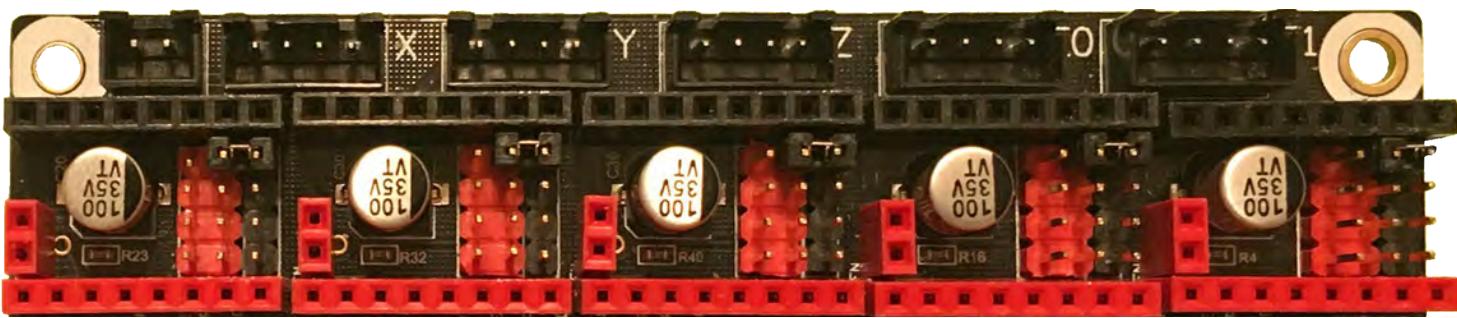
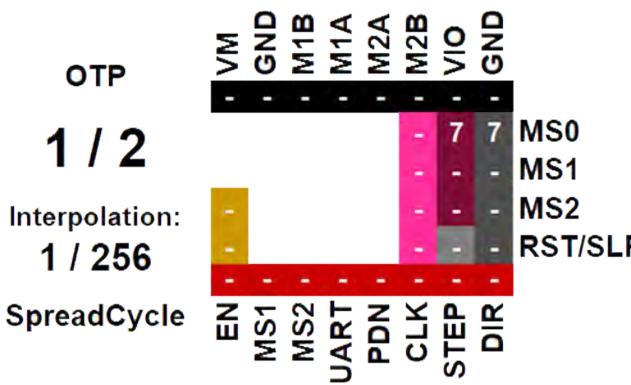
# FYSETC TMC2208 V1.2

## One Time Programming (OTP) Mode

### OTP (One Time Programming) Mode



See [Appendix D](#) for legend

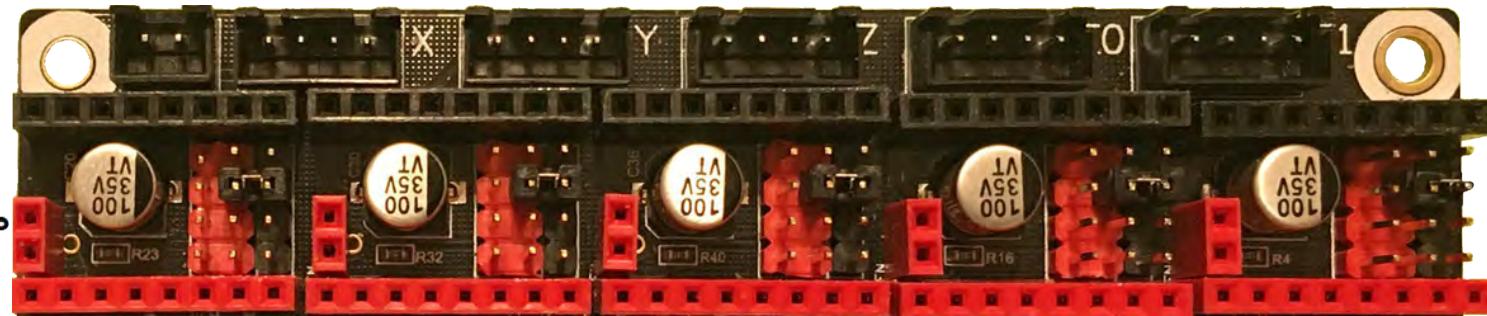
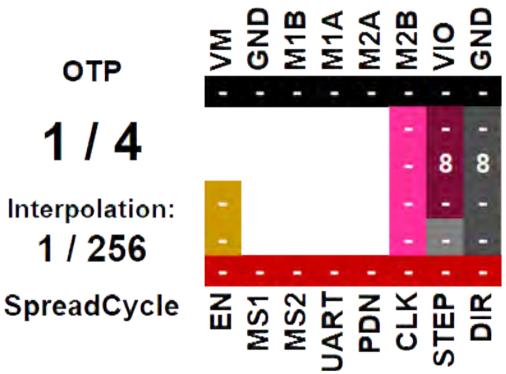


See [Appendix D](#) for legend

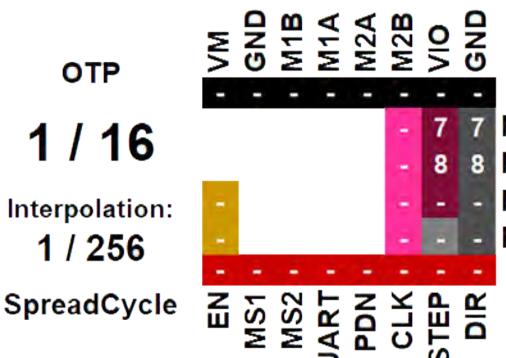
# FYSETC TMC2208 V1.2

## One Time Programming (OTP) Mode

### OTP (One Time Programming) Mode



See [Appendix D](#) for legend

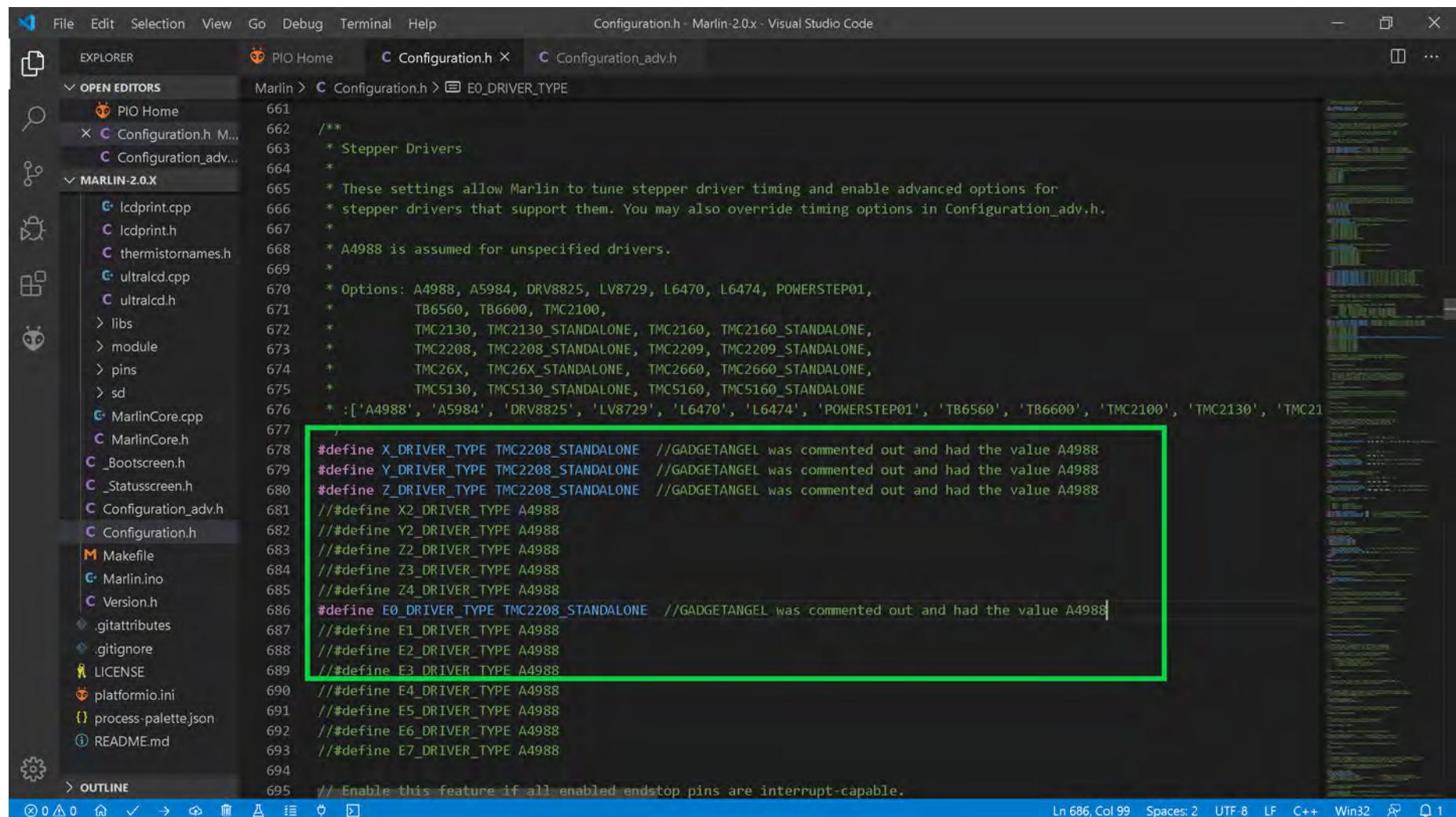


See [Appendix D](#) for legend

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for FYSETC TMC2208 V1.2 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in OTP mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

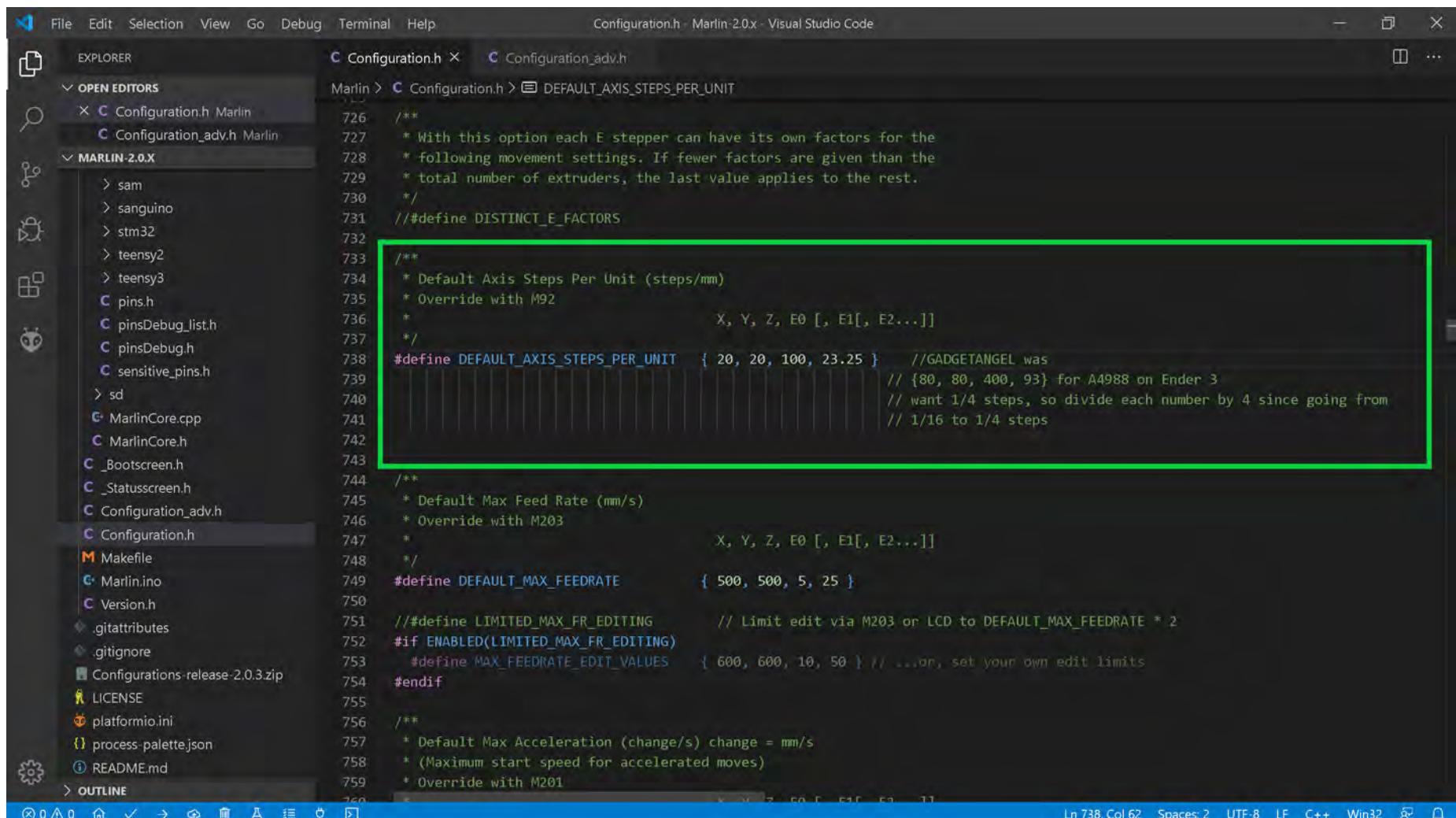
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin 2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
PIO Home 661 /**
X Configuration.h M... 662 */
C Configuration.h 663 * Stepper Drivers
C Configuration_adv.h 664 *
MARLIN-2.0.X 665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
Lcdprint.cpp 666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
Lcdprint.h 667 *
thermistornames.h 668 * A4988 is assumed for unspecified drivers.
ultralcd.cpp 669 *
ultralcd.h 670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
> libs 671 * TB6560, TB6600, TMC2100,
> module 672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
> pins 673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
> sd 674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
MarlinCore.cpp 675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
MarlinCore.h 676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
Configuration.h 677 */
#define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
##define X2_DRIVER_TYPE A4988
##define Y2_DRIVER_TYPE A4988
##define Z2_DRIVER_TYPE A4988
##define Z3_DRIVER_TYPE A4988
##define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
##define E1_DRIVER_TYPE A4988
##define E2_DRIVER_TYPE A4988
##define E3_DRIVER_TYPE A4988
##define E4_DRIVER_TYPE A4988
##define E5_DRIVER_TYPE A4988
##define E6_DRIVER_TYPE A4988
##define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to FYSETC TMC2208 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the Marlin 2.0.x configuration file. A green rectangular box highlights the following code block:

```

726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS

732 /**
733 * Default Axis Steps Per Unit (steps/mm)
734 * Override with M92
735 *
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742

743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
754 #endif

755 /**
756 * Default Max Acceleration (change/s) change = mm/s
757 * (Maximum start speed for accelerated moves)
758 * Override with M201
759 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
 × Configuration.h Marlin 1035 #define DISABLE_Y false
 Configuration_adv.h Marlin 1036 #define DISABLE_Z false
 1037
 1038 // Warn on display about possibly reduced accuracy
 1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
 1040
 1041 // @section extruder
 1042
 1043 #define DISABLE_E false // For all extruders
 1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
 1045
 // @section machine
 1046
 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
 1047 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1048 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1049 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
 1050
 // @section extruder
 1051
 // For direct drive extruder v9 set to true, for geared extruder set to false.
 1052 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1053 #define INVERT_E1_DIR false
 1054 #define INVERT_E2_DIR false
 1055 #define INVERT_E3_DIR false
 1056 #define INVERT_E4_DIR false
 1057 #define INVERT_E5_DIR false
 1058 #define INVERT_E6_DIR false
 1059 #define INVERT_E7_DIR false
 1060
 // @section homing
 1061
 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
 1062
 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
 1063
 1064
 1065
 1066
 1067
 1068
 1069

```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



**Configuration.h - Marlin-2.0.x - Visual Studio Code**

File Edit Selection View Go Run Terminal Help

EXPLORER Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src... pins\_BTT\_SKR\_common.h Marlin\src... Configuration\_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the name below to that matching your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[=====] [SUCCESS] Took 130.61 seconds =====

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| ramps           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino44p     | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUINO           | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in One Time Programming (OTP) Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

Configuration.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE

```

```

Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

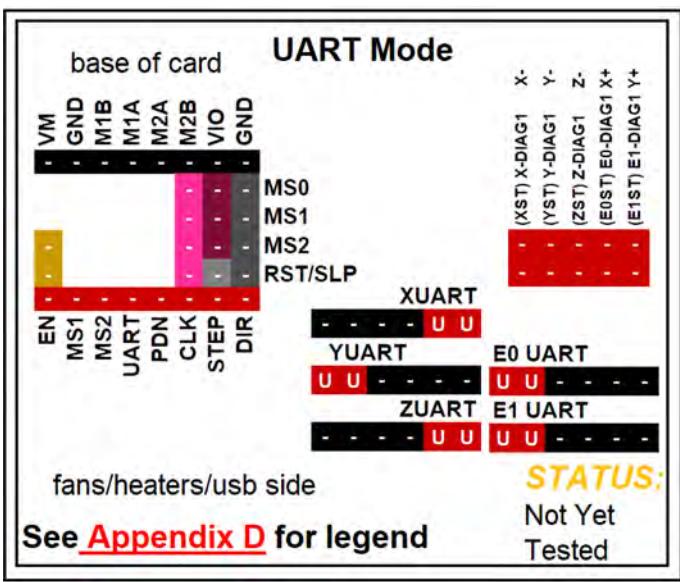
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET_3D         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

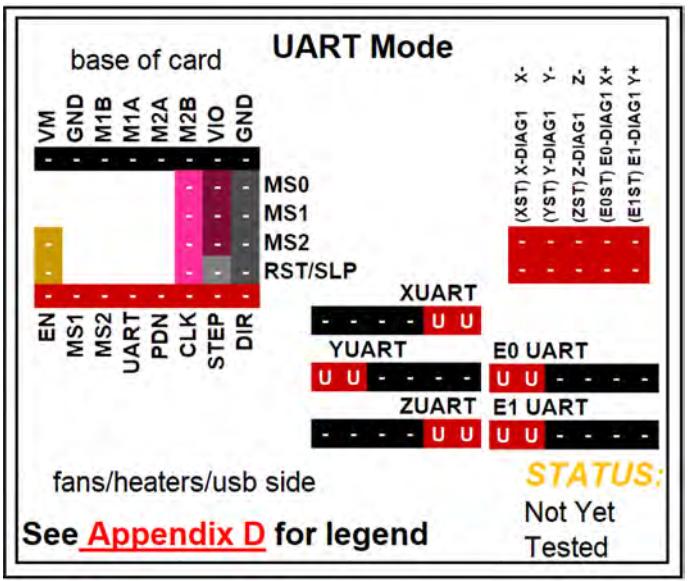
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**FYSETC TMC2208 V1.2****UART Mode**

**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

| Driver Chip                                                                                    | Steps are set inside<br>of your Firmware                                                               |                                                                                                        |
|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>FYSETC</b><br><b>TMC2208</b><br>UART Mode<br>Maximum 256 Subdivision<br>35V DC<br>2A (peak) |                                                                                                        |                                                                                                        |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.11Ω           | $I_{MAX} = V_{ref}$<br>See Appendix B #4. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90$ | $V_{ref} = I_{MAX}$<br>See Appendix B #4. Use 50% to 90% as shown below:<br>$V_{ref} = V_{ref} * 0.90$ |



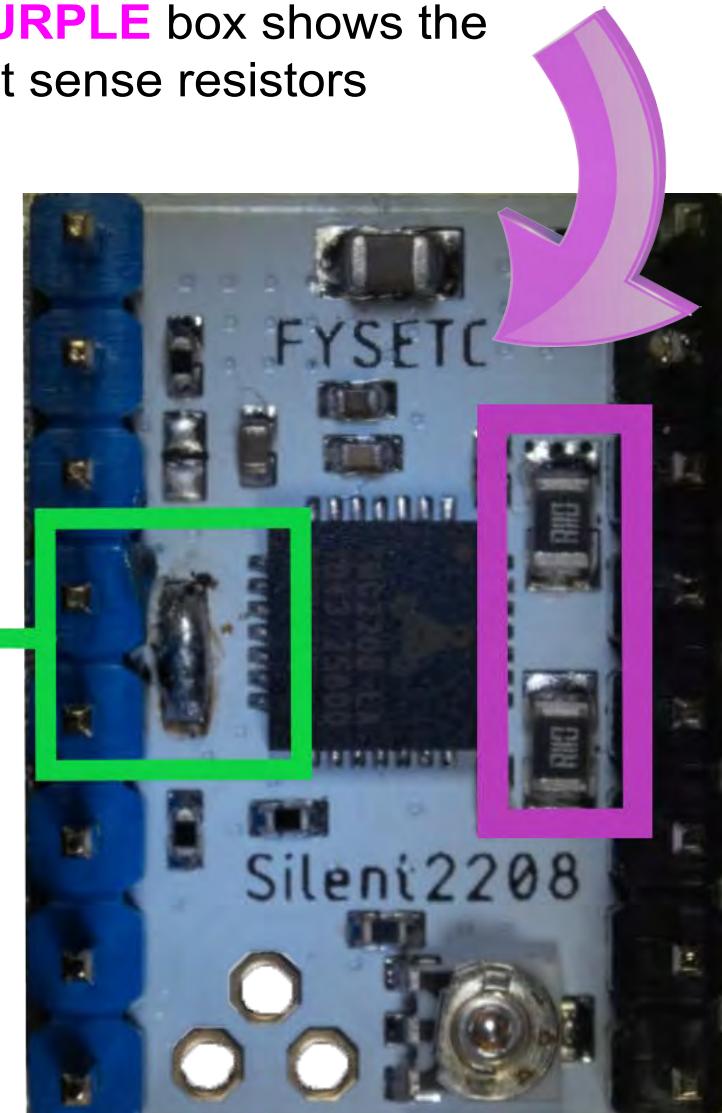
MOST FYSETC TMC2208 boards are sold as V1.2 driver boards. The V1.2 driver board **might be** setup to be in UART mode. BUT the V1.0 driver board is setup to be in stand-alone mode by default. **So please check your boards to ensure they are in the correct mode you desire!**



## FYSETC TMC2208 V1.2

### UART Mode

**Important:** To ensure that the FYSETC TMC2208 V1.0 or V1.2 is in UART Mode, check to see if all the adjacent pads are soldered together on the bottom of the driver board, as shown in the **GREEN** box. The **PURPLE** box shows the location of the current sense resistors ( $R_s$ ).



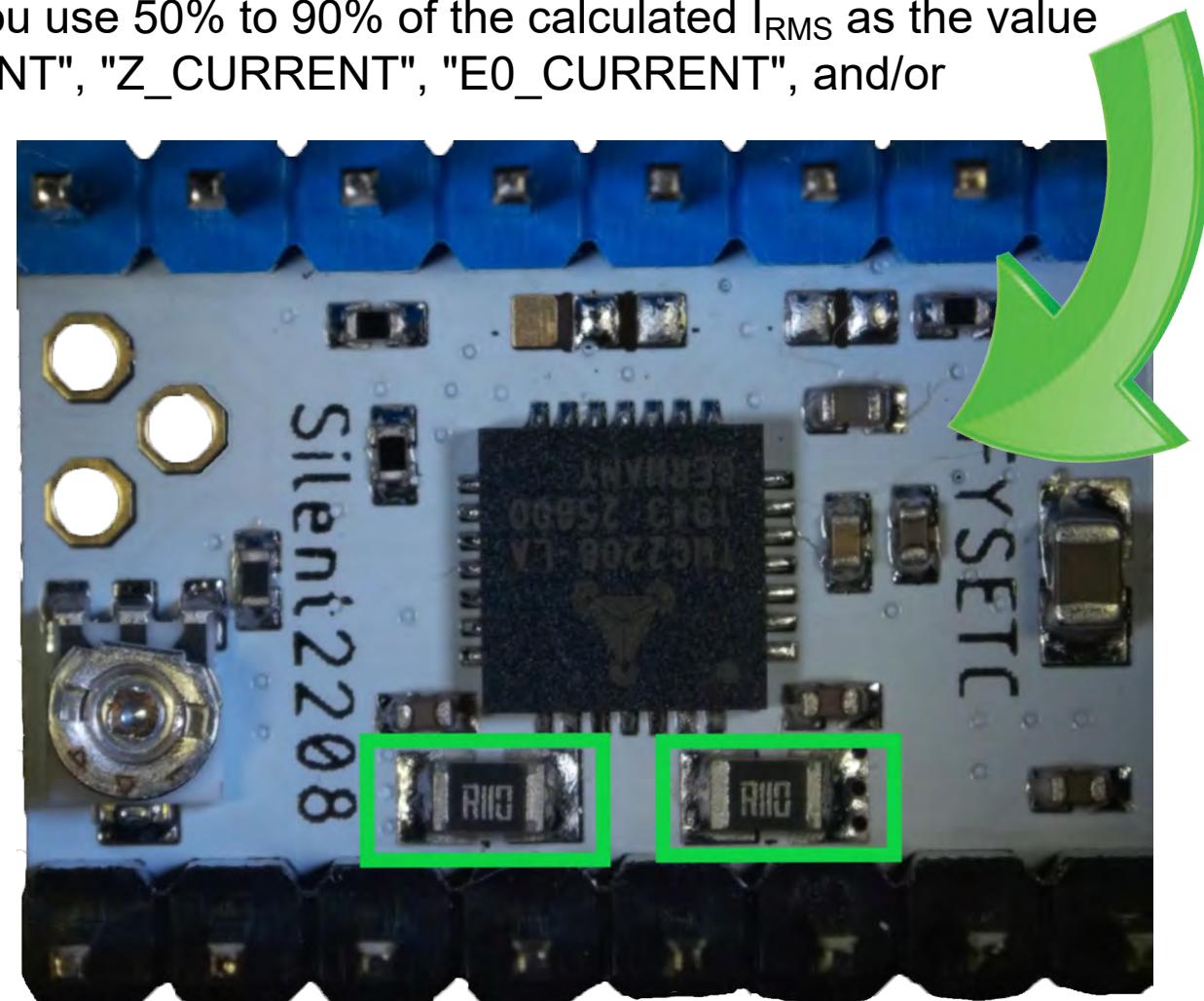
**FYSETC TMC2208 V1.2 in UART Mode**

# UART Mode

## UART Mode

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

$R_s = R_{050}$  is 0.05 Ohms  
 $R_s = R_{062}$  is 0.062 Ohms  
 $R_s = R_{068}$  is 0.068 Ohms  
 $R_s = R_{075}$  is 0.075 Ohms  
 $R_s = R_{100}$  is 0.1 Ohms  
 $R_s = R_{110}$  is 0.11 Ohms  
 $R_s = R_{150}$  is 0.15 Ohms  
 $R_s = R_{200}$  is 0.2 Ohms  
 $R_s = R_{220}$  is 0.22 Ohms



FYSETC TMC2208 V1.2UART Mode

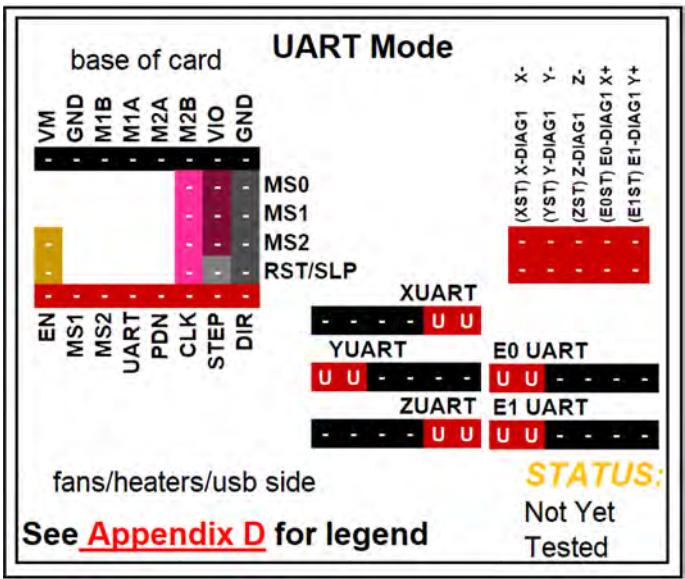
# UART Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.

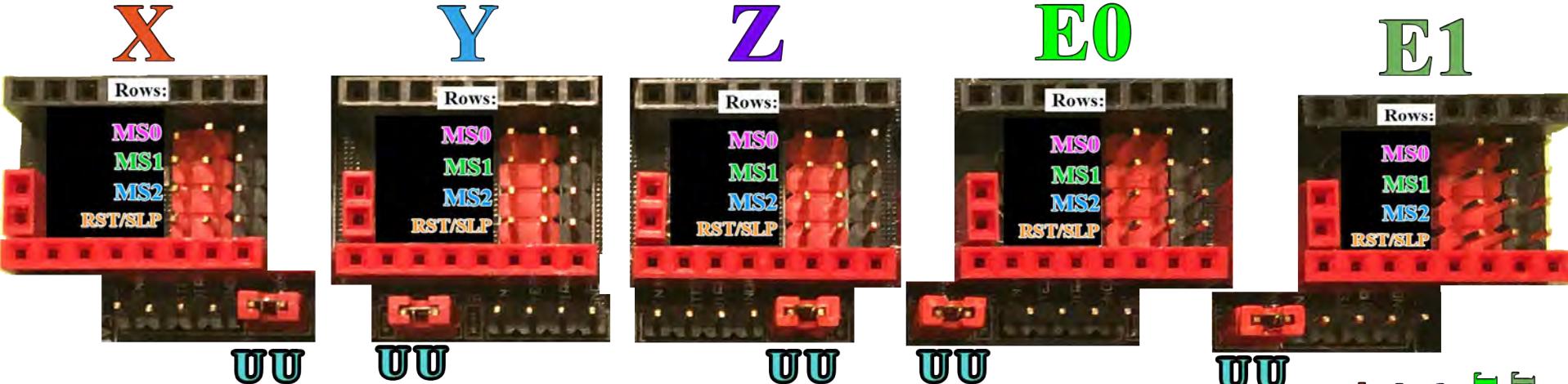


**Note:** Set the "U" Jumper(s) for UART MODE!





Driver Socket Representation: **UU** Meaning: **Jumper set**



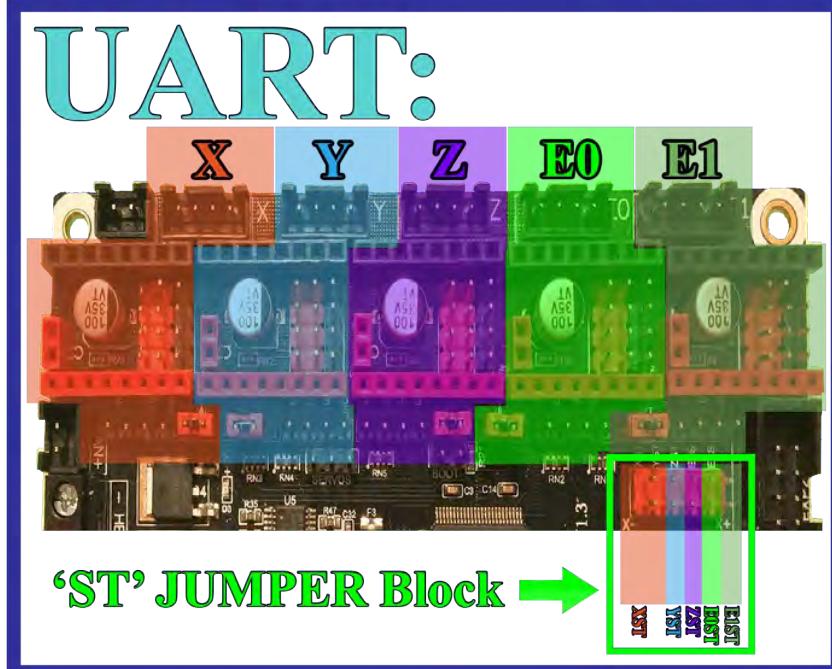
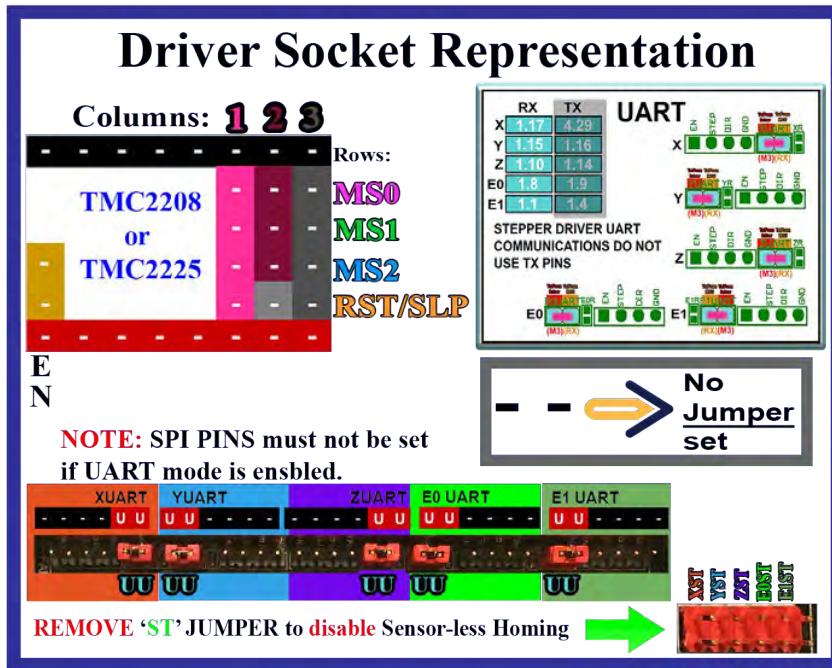
**REMOVE 'ST' JUMPER(s) to disable Sensor-less Homing – for drivers without stallGuard™ feature**



FYSETC TMC2208 V1.2UART Mode

# UART Mode

## SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode



UART Capable Drivers **without** stallGuard™ feature: TMC2208, and TMC2225

Driver Socket Representation: 

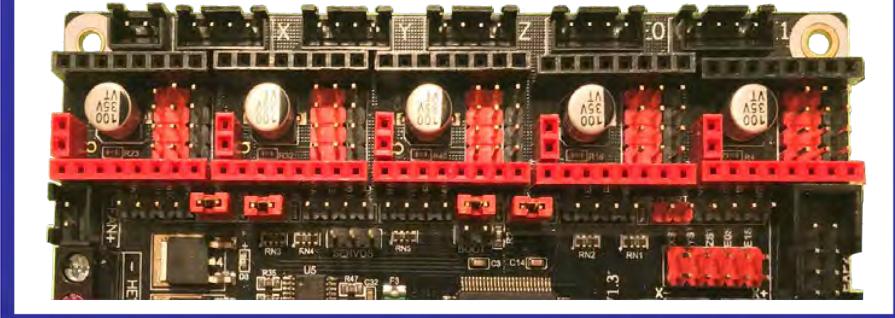
Meaning:

Driver Socket Representation: 

Meaning: **set Jumper so that both U PINS are covered by the Jumper**

Each Axis has its own **UART PINS (UU)**. You can set as many axes as you need. The **UART**: picture shows all axes having **UART enabled AND all 'ST' JUMPERS REMOVED**. All other PINS need to be empty for **UART mode to work properly**.

Here is an example of only the X, Y, Z, and E0 axes having **UART enabled AND 'ST' JUMPERS removed** for ALL Axes:



## UART Mode

### UART Mode

#### Information on Sensor-less Homing

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing.

Please read the PREFACE to this manual on “Stall detection and Sensor-less Homing”.

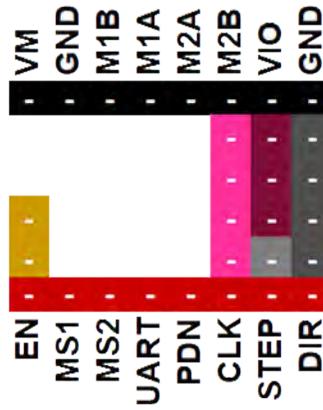
**NOTE:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.



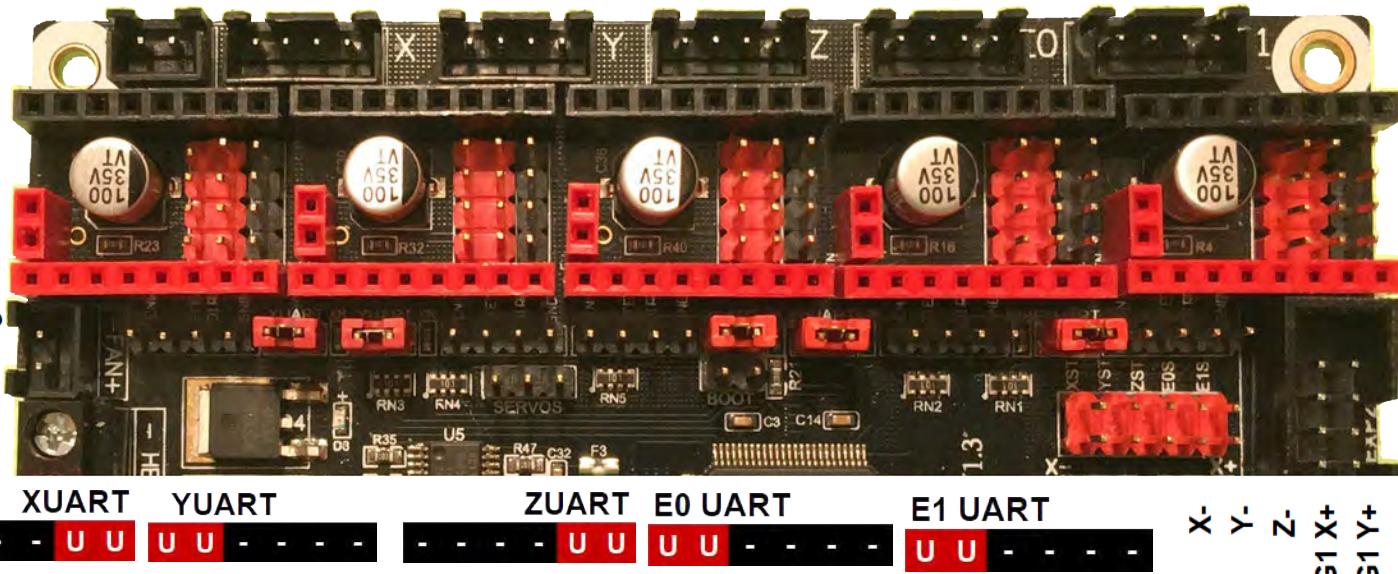
# UART Mode

**Note:** Set the "U" Jumper(s) for UART MODE!

UART



See [Appendix D](#) for legend



**Note:** TMC2208 does not have sensor-less homing capability.

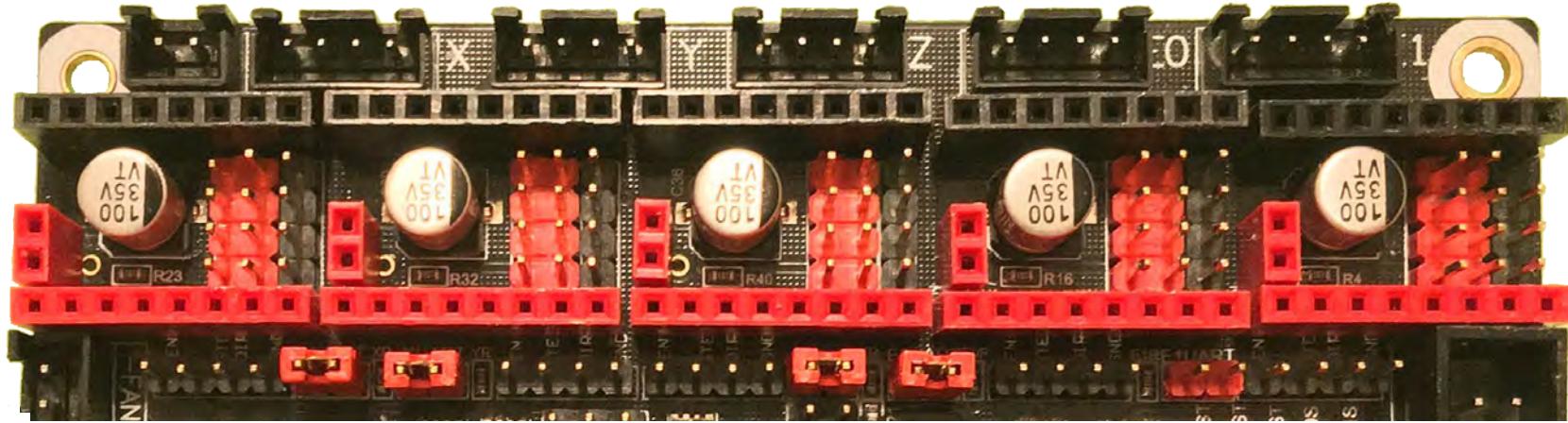


# UART Mode

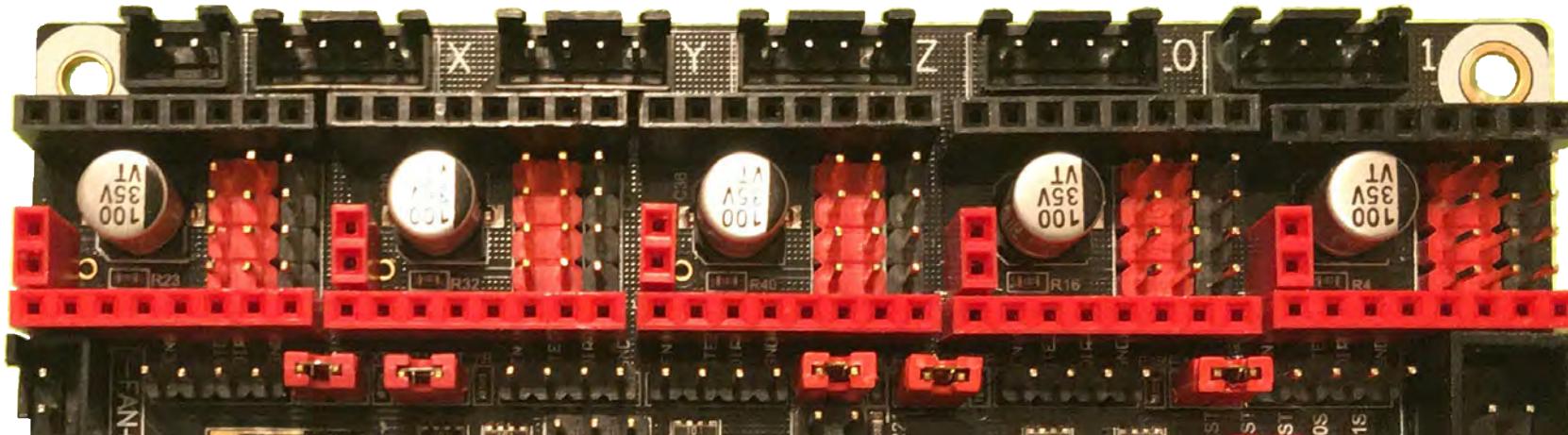
## UART Mode

### Examples of Different UART Configurations

X, Y, Z,  
and E0 axes  
configured for  
UART mode.



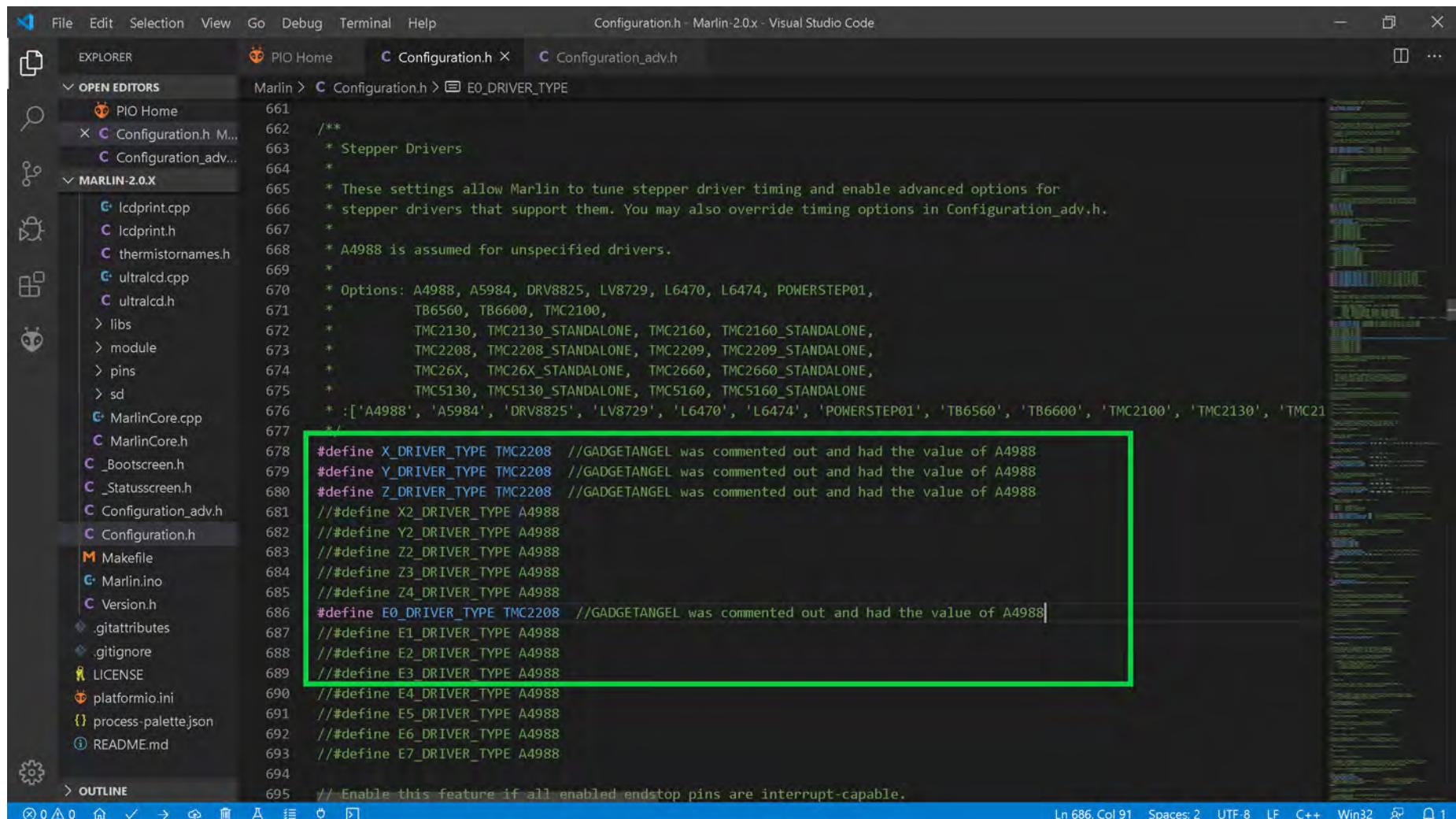
X, Y, Z, E0  
and E1 axes  
configured for  
UART mode.



## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for FYSETC TMC2208 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2208 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2208 drivers in UART mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

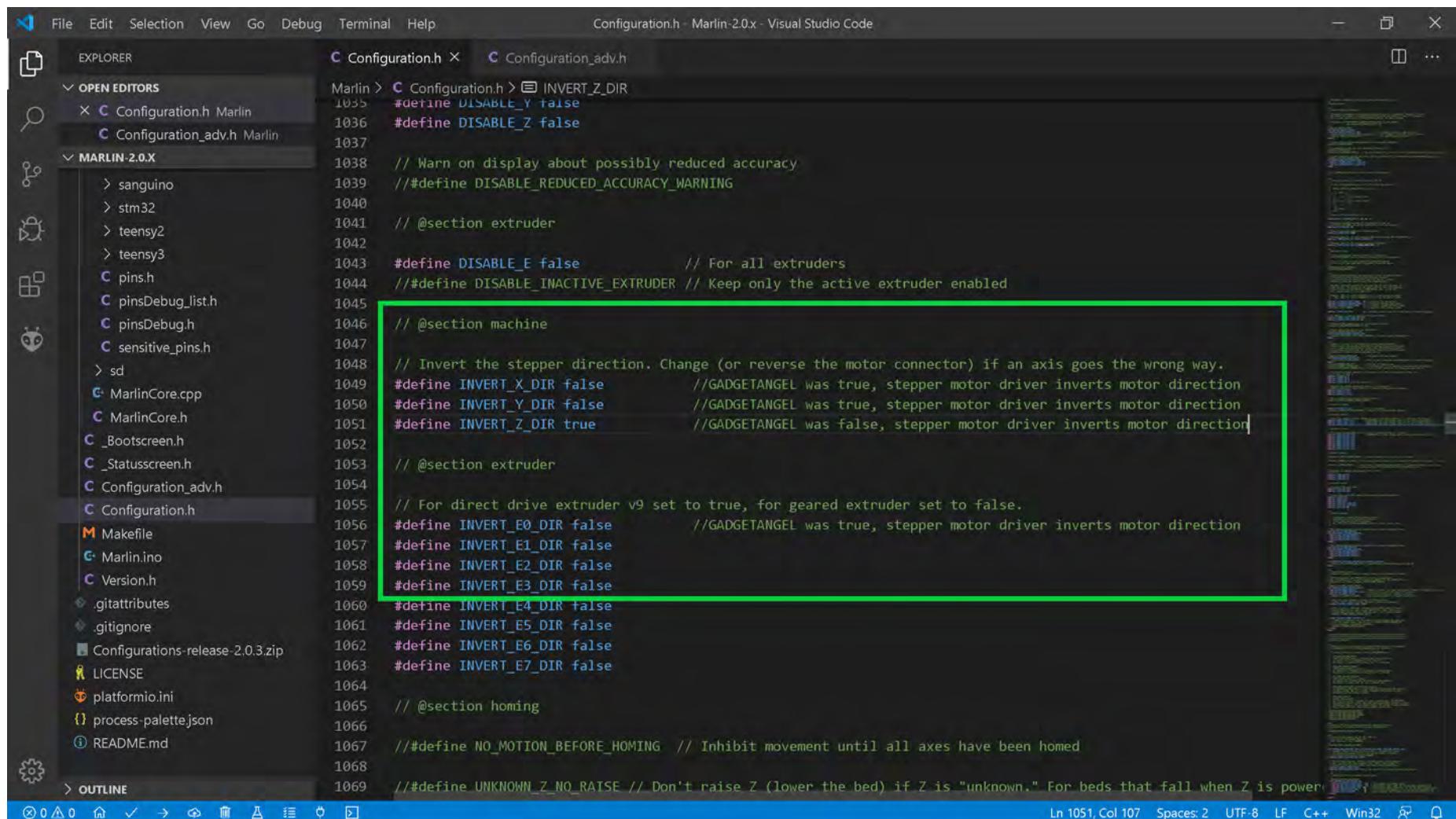
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
 PIO Home
 Configuration.h M...
 Configuration_adv...
MARLIN-2.0.X
 LCDprint.cpp
 LCDprint.h
 thermistornames.h
 ultralcd.cpp
 ultralcd.h
 libs
 module
 pins
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration_adv.h
 Configuration.h
 Makefile
 Marlin.ino
 Version.h
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
 OUTLINE
Ln 686, Col 91 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌂ 1
661 /**
662 * Stepper Drivers
663 *
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 *
668 * A4988 is assumed for unspecified drivers.
669 *
670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
671 * TB6560, TB6600, TMC2100,
672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
677 */
678 #define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
679 #define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
680 #define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2208 drivers, I must invert the stepper motor direction because the TMC2208 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2208 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



```

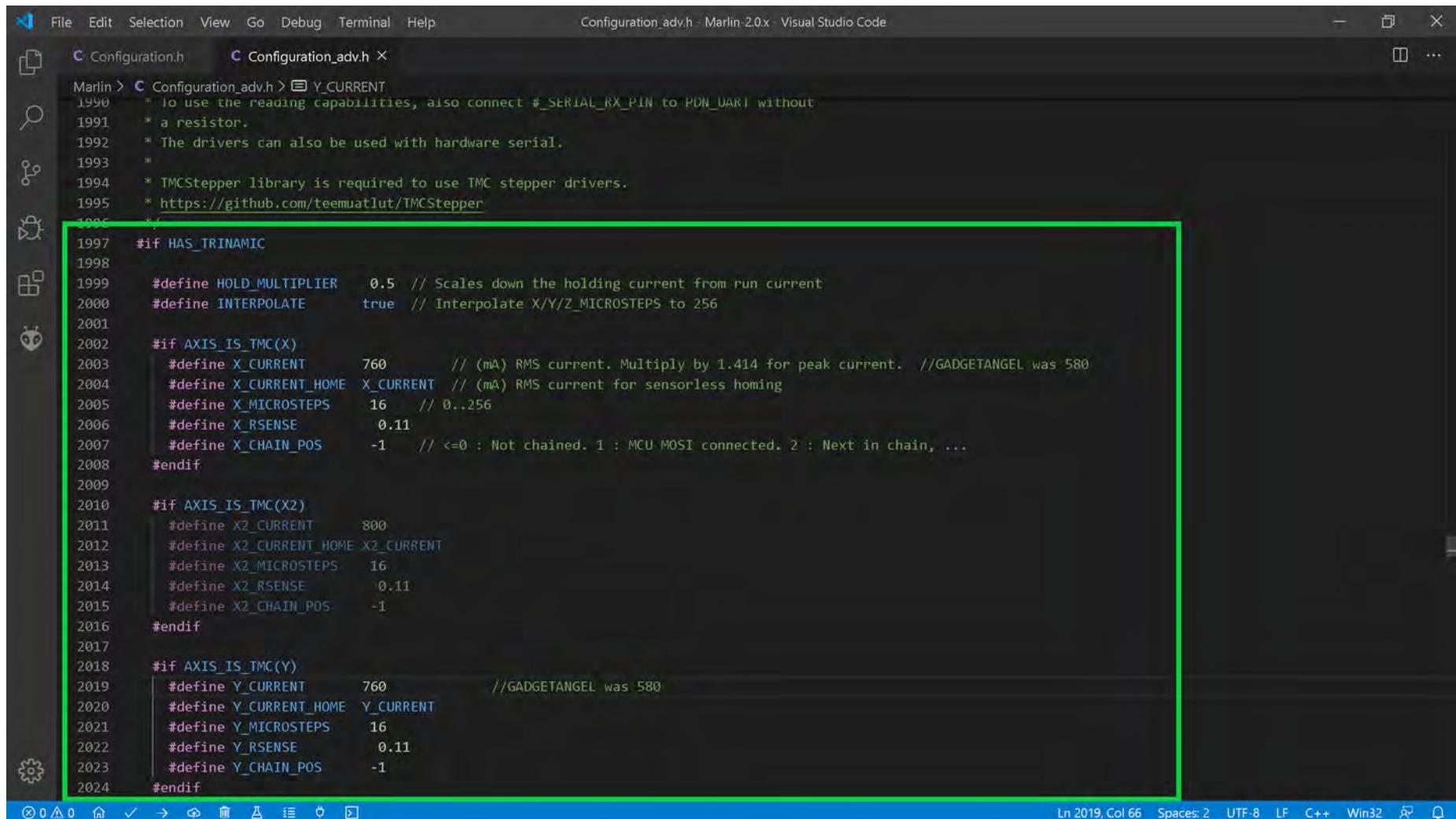
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RATSE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC2208 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.11. Ensure "Y\_RSENSE" is set to 0.11.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif

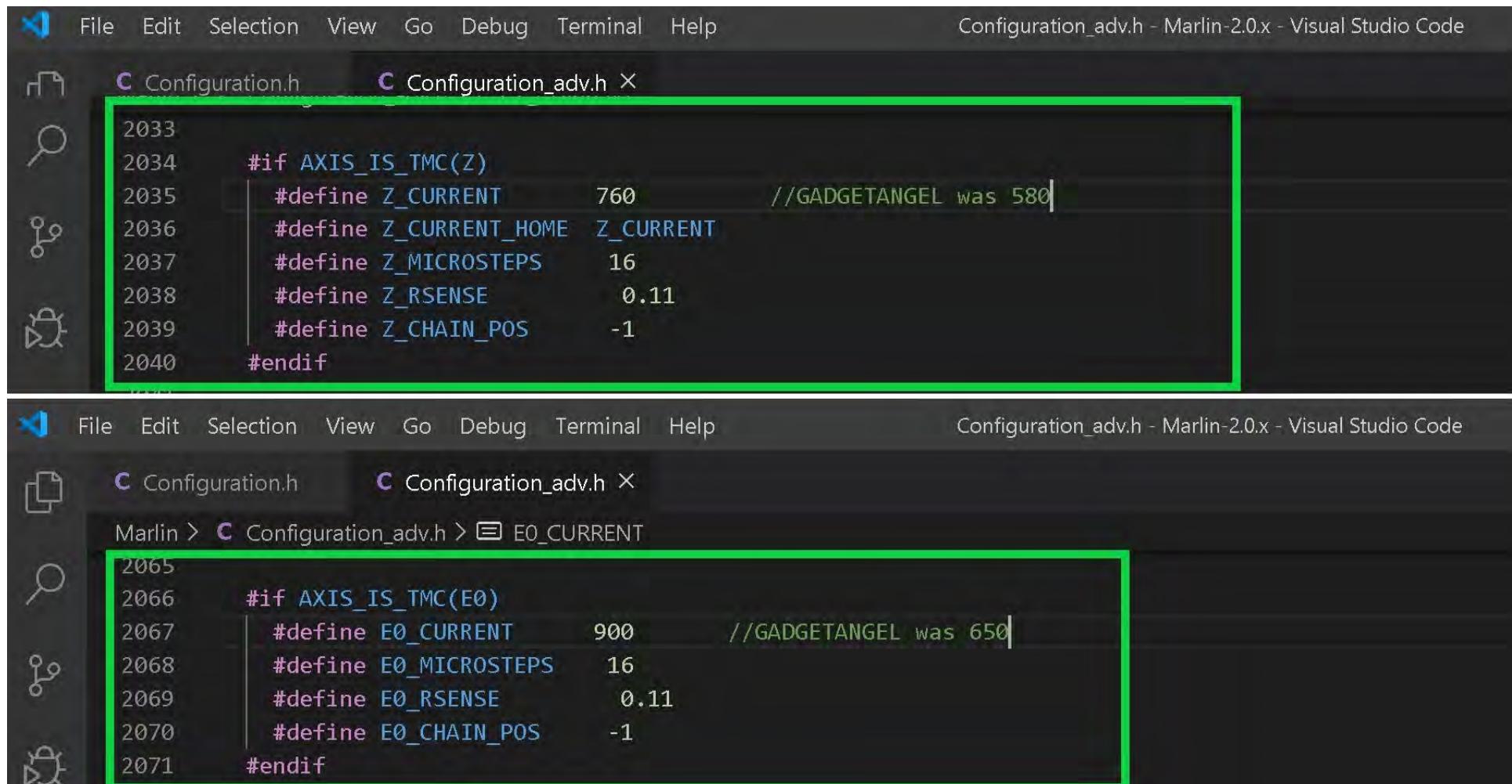
```

Ln 2019, Col 66 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the **GREEN** boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.11. Ensure "E0\_RSENSE" is set to 0.11.
- If you **do not want to use  $V_{ref}$**  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h X
2033
2034 #if AXIS_IS_TMC(Z)
2035 #define Z_CURRENT 760 //GADGETANGEL was 580
2036 #define Z_CURRENT_HOME Z_CURRENT
2037 #define Z_MICROSTEPS 16
2038 #define Z_RSENSE 0.11
2039 #define Z_CHAIN_POS -1
2040 #endif

```

```

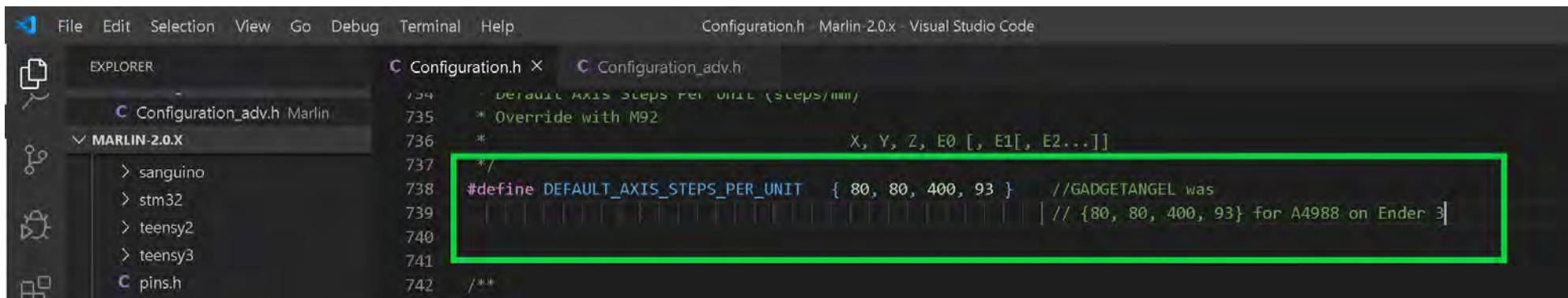
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h X
Marlin > Configuration_adv.h > E0_CURRENT
2065
2066 #if AXIS_IS_TMC(E0)
2067 #define E0_CURRENT 900 //GADGETANGEL was 650
2068 #define E0_MICROSTEPS 16
2069 #define E0_RSENSE 0.11
2070 #define E0_CHAIN_POS -1
2071 #endif

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes



File Edit Selection View Go Debug Terminal Help Configuration.h Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration\_adv.h

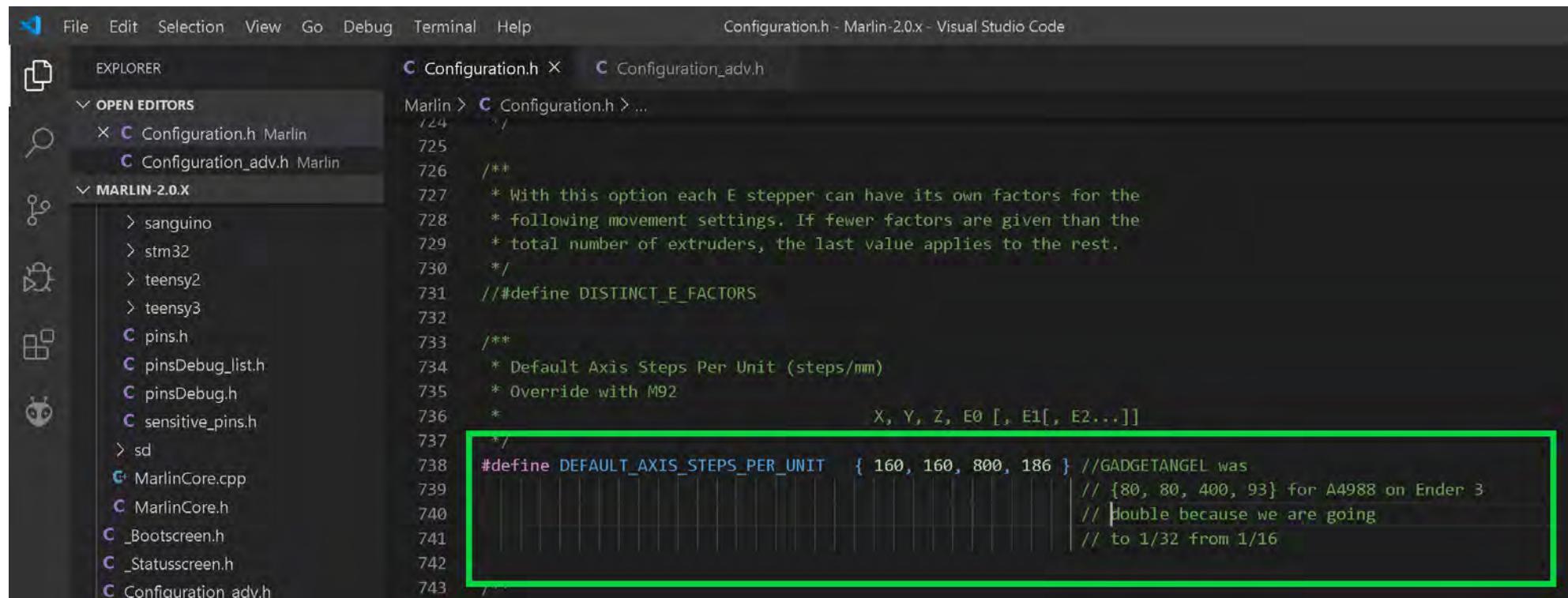
MARLIN-2.0.X

```

 734 * Default Axis Steps Per Unit (steps/mm)
 735 * Override with M92
 736 * X, Y, Z, E0 [, E1[, E2...]]
 737 */
 738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
 739 // {80, 80, 400, 93} for A4988 on Ender 3
 740
 741 /**
 742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration\_adv.h

MARLIN-2.0.X

```

 724 */
 725
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 // #define DISTINCT_E_FACTORS
 732
 733 /**
 734 * Default Axis Steps Per Unit (steps/mm)
 735 * Override with M92
 736 *
 737 */
 738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
 739 // {80, 80, 400, 93} for A4988 on Ender 3
 740 // Double because we are going
 741 // to 1/32 from 1/16
 742
 743 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
Configuration.h Marlin Configuration_adv.h Marlin
X C Configuration_adv.h Marlin
MARLIN-2.0.X
sanguino
stm32
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h

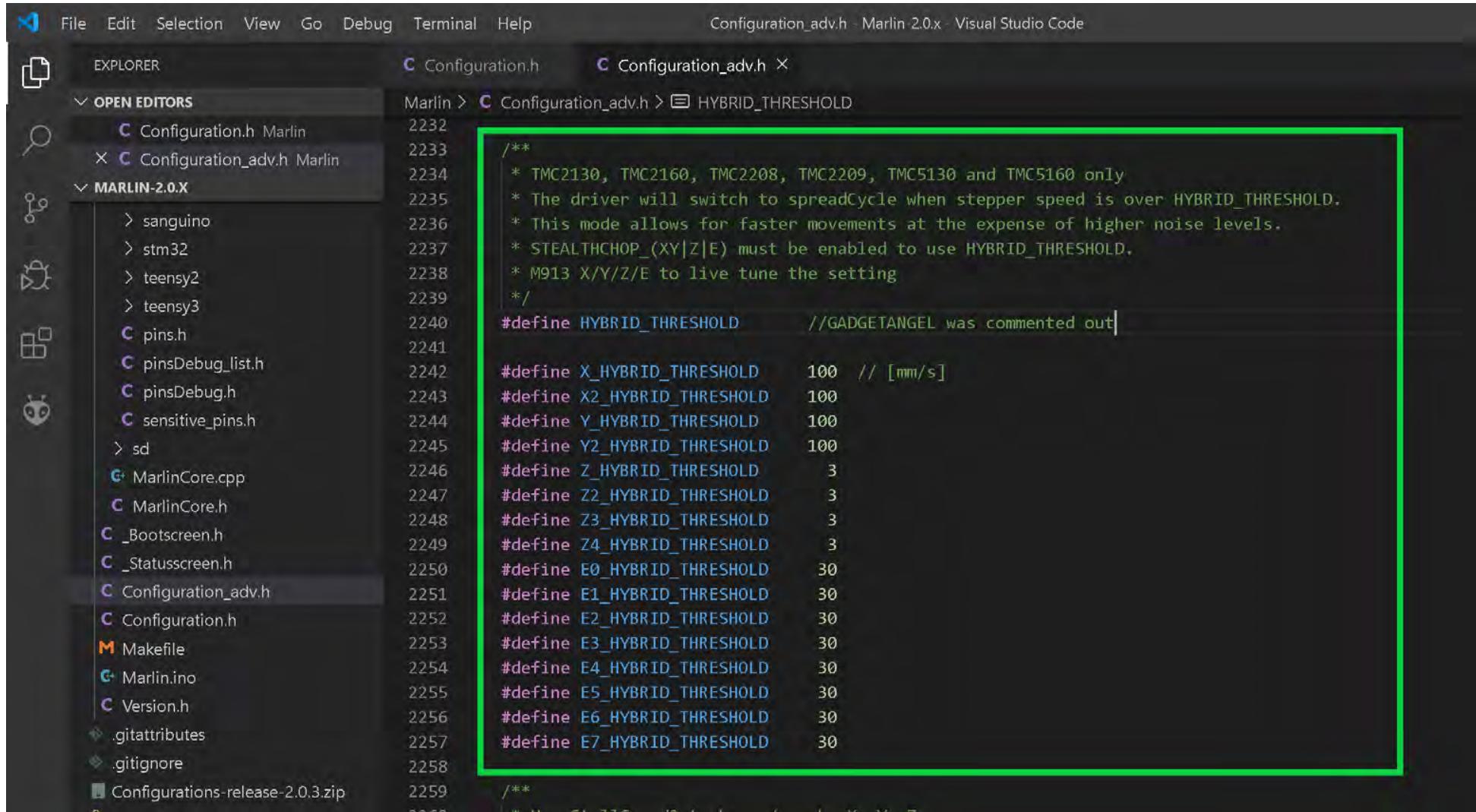
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):**
  - OPEN EDITORS: Configuration.h Marlin, Configuration\_adv.h Marlin
  - MARLIN-2.0.X:
    - > sanguino
    - > stm32
    - > teensy2
    - > teensy3
    - C pins.h
    - C pinsDebug\_list.h
    - C pinsDebug.h
    - C sensitive\_pins.h
    - > sd
    - C MarlinCore.cpp
    - C MarlinCore.h
    - C \_Bootscreen.h
    - C \_Statusscreen.h
    - C Configuration\_adv.h
    - C Configuration.h
    - M Makefile
    - C Marlin.ino
    - C Version.h
  - .gitattributes
  - .gitignore
  - Configurations-release-2.0.3.zip
- Central Area (Editor):** Shows the content of Configuration\_adv.h, specifically the HYBRID\_THRESHOLD section. The code is as follows:
 

```

2232 /**
2233 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
2234 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
2235 * This mode allows for faster movements at the expense of higher noise levels.
2236 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
2237 * M913 X/Y/Z/E to live tune the setting
2238 */
2239 #define HYBRID_THRESHOLD //GADGETANGEL was commented out
2240
2241 #define X_HYBRID_THRESHOLD 100 // [mm/s]
2242 #define X2_HYBRID_THRESHOLD 100
2243 #define Y_HYBRID_THRESHOLD 100
2244 #define Y2_HYBRID_THRESHOLD 100
2245 #define Z_HYBRID_THRESHOLD 3
2246 #define Z2_HYBRID_THRESHOLD 3
2247 #define Z3_HYBRID_THRESHOLD 3
2248 #define Z4_HYBRID_THRESHOLD 3
2249 #define E0_HYBRID_THRESHOLD 30
2250 #define E1_HYBRID_THRESHOLD 30
2251 #define E2_HYBRID_THRESHOLD 30
2252 #define E3_HYBRID_THRESHOLD 30
2253 #define E4_HYBRID_THRESHOLD 30
2254 #define E5_HYBRID_THRESHOLD 30
2255 #define E6_HYBRID_THRESHOLD 30
2256 #define E7_HYBRID_THRESHOLD 30
2257
2258 /**
2259 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line](#).

Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h X
Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```

Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h X
Marlin > Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- The end of Marlin setup for FYSETC TMC2208 V1.2 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



**Configuration.h - Marlin-2.0.x - Visual Studio Code**

**EXPLORER**

- OPEN EDITORS
  - Configuration.h Marlin
  - pins\_BTT\_SKR\_V1\_3.h Marlin\src...
  - pins\_BTT\_SKR\_common.h Marlin...
  - Configuration\_adv.h Marlin
- MARLIN-2.0.X
  - samd
  - sanguino
  - stm32f1
  - stm32f4
  - stm32f7
  - teensy2
  - teensy3
  - pins.h
  - pinsDebug.h
  - pinsDebug.list.h
  - sensitive\_pins.h
  - sd
  - MarlinCore.cpp
  - MarlinCore.h
  - \_Bootscreen.h
  - \_Statusscreen.h
  - Configuration.h
  - Configuration\_adv.h
  - Makefile
  - Marlin.ino
  - Version.h
  - .editorconfig
  - .gitattributes
  - .gitignore
  - LICENSE
  - platformio.ini
  - process-palette.json
  - README.md

**TERMINAL**

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
126 // #define BLUETOOTH
128
129 // Choose the correct BOARD in that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libframeworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====
Environment Status Duration
----- -----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino64p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_dfu IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

**OUTLINE** **TIMELINE**

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

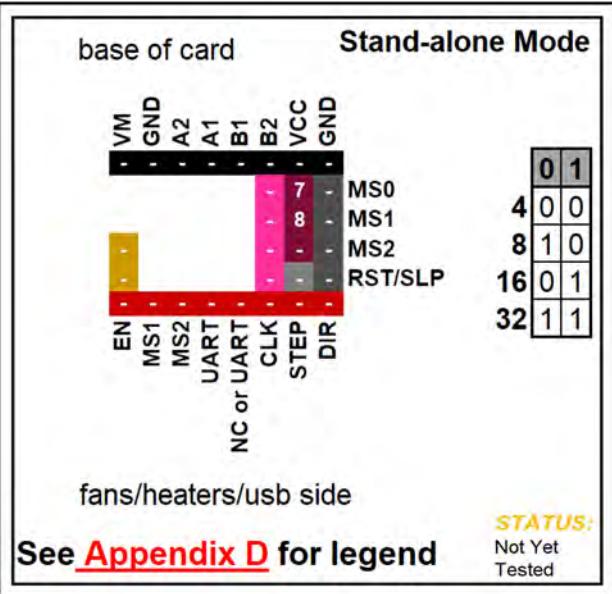
## The (latest release of) Marlin Setup for FYSETC TMC2208 V1.2 Drivers in UART Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

The screenshot shows the Visual Studio Code interface for Marlin 2.0.x setup. The Explorer sidebar on the left lists files and folders related to Marlin 2.0.x, including Configuration.h, pins\_BTT\_SKR\_V1\_3.h, pins\_BTT\_SKR\_common.h, and Configuration\_adv.h. The main editor window displays Configuration.h with code for baud rate and motherboard definitions. The bottom status bar shows the build progress: "Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o" and so on, ending with "[SUCCESS] Took 130.61 seconds". The bottom toolbar includes icons for Save, Undo, Redo, and the Upload button, which is highlighted by a red box. A table in the bottom right shows build environments and their status.

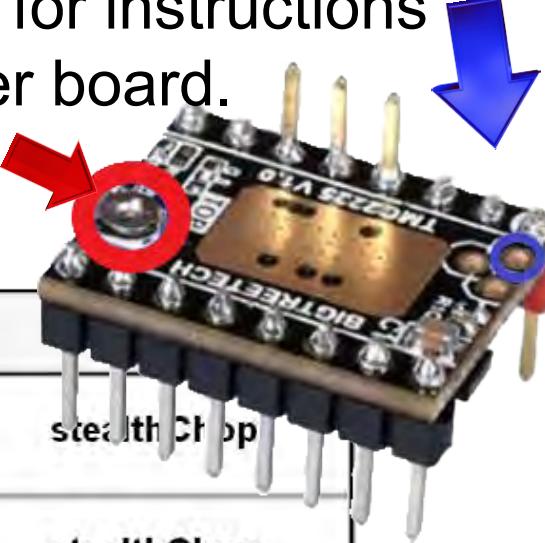
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| SUE_L103        | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC2225 V1.0****Stand-alone Mode**

**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See **Appendix A** for instructions on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.

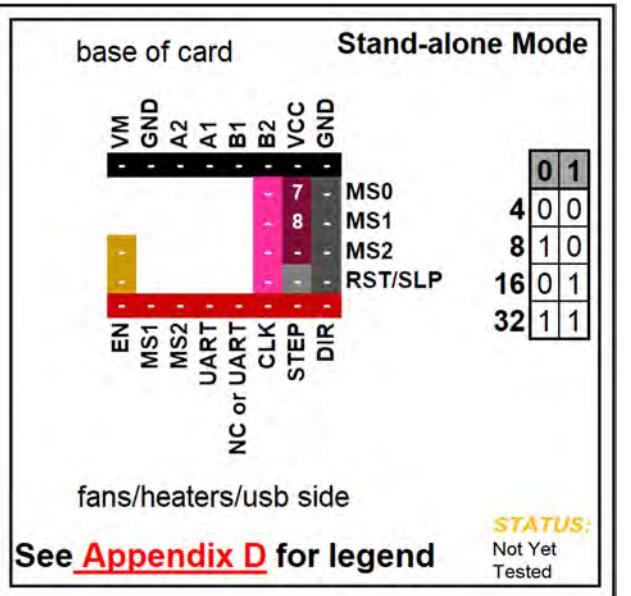


| Driver Chip                                                                                                     | MS0         | MS1         | Steps         | Interpolation  |                    |
|-----------------------------------------------------------------------------------------------------------------|-------------|-------------|---------------|----------------|--------------------|
| <b>BIQU®<br/>TMC2225</b><br><small>Stand Alone Mode<br/>Maximum 32 Subdivision<br/>35V DC<br/>2A (peak)</small> | <b>Low</b>  | <b>Low</b>  | <b>1 / 4</b>  | <b>1 / 256</b> | <b>stealthChop</b> |
|                                                                                                                 | <b>High</b> | <b>Low</b>  | <b>1 / 8</b>  | <b>1 / 256</b> | <b>stealthChop</b> |
|                                                                                                                 | <b>Low</b>  | <b>High</b> | <b>1 / 16</b> | <b>1 / 256</b> | <b>stealthChop</b> |
|                                                                                                                 | <b>High</b> | <b>High</b> | <b>1 / 32</b> | <b>1 / 256</b> | <b>stealthChop</b> |

|                                                                                                          |                                                                                                                                                            |                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Driving Current Calculation Formula</b><br><small>R<sub>S</sub>(Typical Sense Resistor)=0.15Ω</small> | <b>I<sub>MAX</sub>=V<sub>ref</sub>*0.7222</b><br>See Appendix B#10. Use 50% to 90% as shown below:<br><b>I<sub>MAX</sub>=(V<sub>ref</sub>*0.7222)*0.90</b> | <b>V<sub>ref</sub>=I<sub>MAX</sub>*1.3846</b><br>See Appendix B#10. Use 50% to 90% as shown below:<br><b>V<sub>ref</sub>=(I<sub>MAX</sub>*1.3846)*0.90</b> |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

- See next page for the legend that belongs to the above chart.

**Driver Chip Chart:**

| Driver Chip             | MS0  | MS1  | MS2  | Microstep Resolution | Excitation Mode |
|-------------------------|------|------|------|----------------------|-----------------|
| <b>LOGO</b>             | Low  | Low  | Low  | Full Step            | 2 Phase         |
| Chip Name               | High | Low  | Low  | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision | Low  | High | Low  | 1/4 Step             | W1-2 Phase      |
| XXV DC xxA (peak)       | High | High | Low  | 1/8 Step             | 2W1-2 Phase     |
|                         | Low  | Low  | High | 1/16 Step            | 4W1-2 Phase     |
|                         | High | Low  | High | 1/32 Step            | 8W1-2 Phase     |
|                         | Low  | High | High | 1/64 Step            | 16W1-2 Phase    |
|                         | High | High | High | 1/128 Step           | 32W1-2 Phase    |

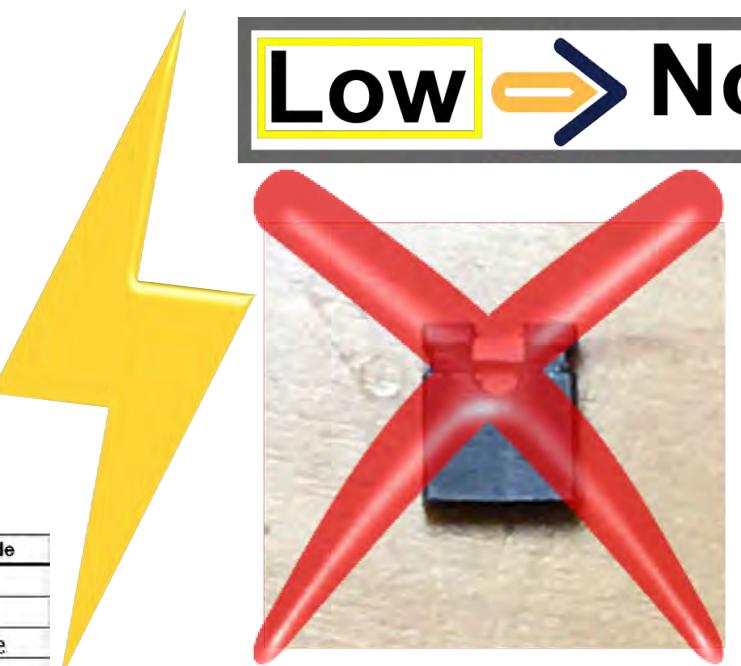
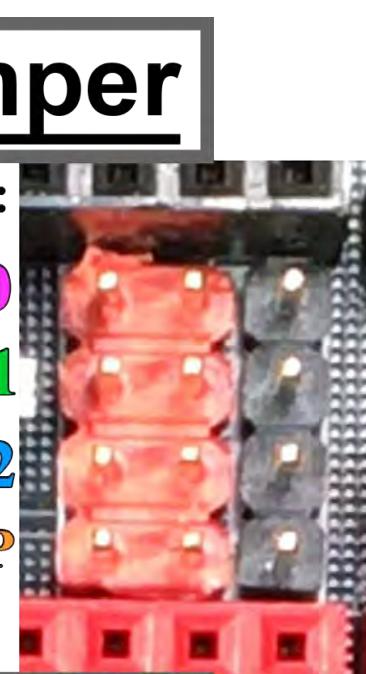
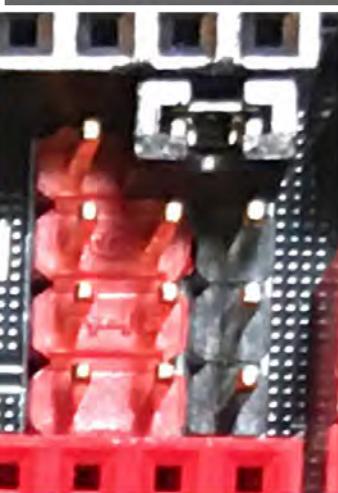
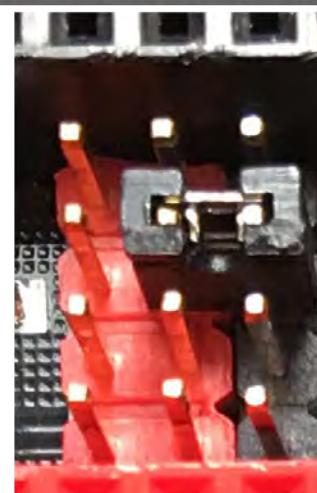
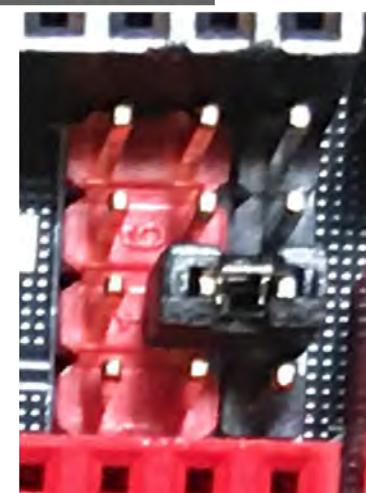
**Driving Current Calculation Formula**  $I_{MAX} = \text{FORMULA}$   $V_{ref} = \text{FORMULA}$

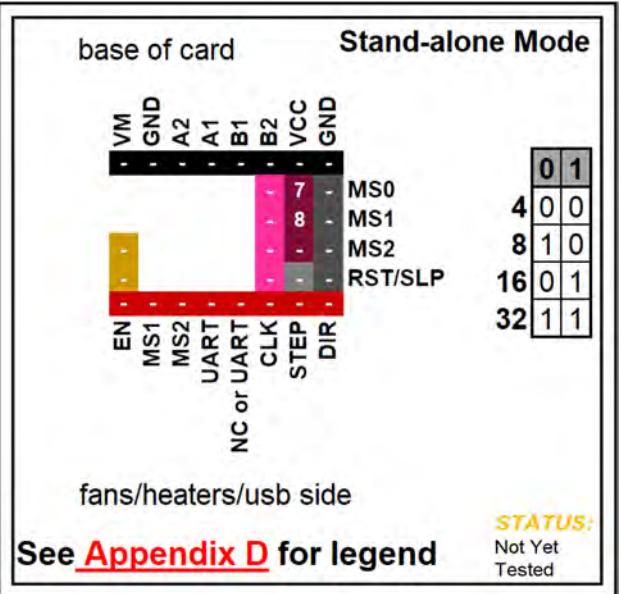
$R_S(\text{Typical Sense Resistor}) = x.xx \Omega$

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

**BIQU TMC2225 V1.0****Stand-alone Mode****SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers****Rows:****MS0****MS1****MS2****RST/SLP****Rows:****MS0****MS1****MS2****RST/SLP****MS0 SET HIGH****MS1 SET HIGH****MS2 SET HIGH**



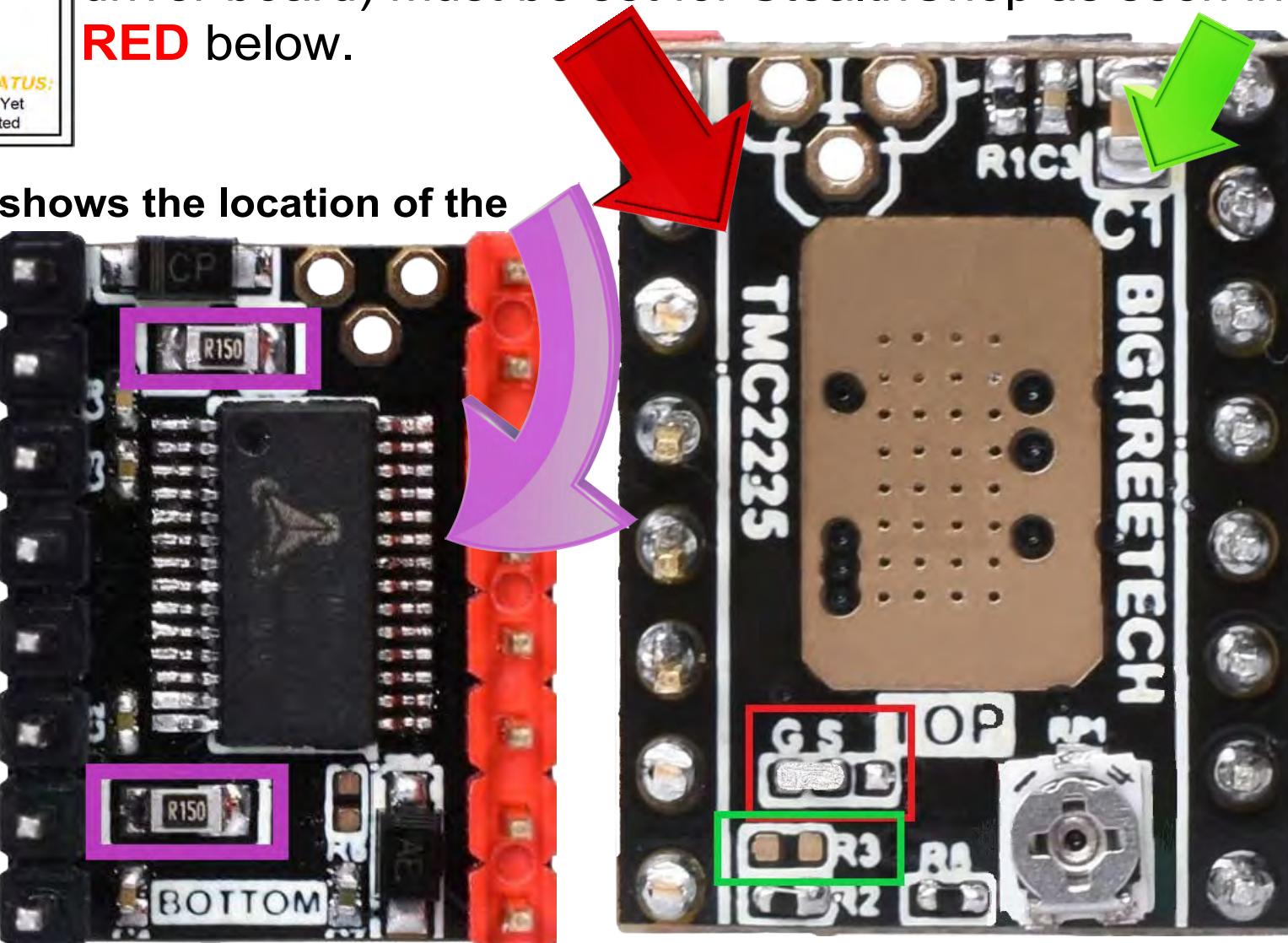
The **PURPLE** boxes shows the location of the current sense resistors ( $R_s$ ).

**Note:** MOST BIQU TMC2225 V1.0 driver boards, when purchased for **UART mode**, will have two **R3** pads (located on the top of the driver board), which are **NOT soldered together**. This indicates the driver board can use the **UART** pin for the **UART single wire interface** (if the **UART, "U"**, jumper is in place on the **SKR V1.3** board)

## BIQU TMC2225 V1.0

### Stand-alone Mode

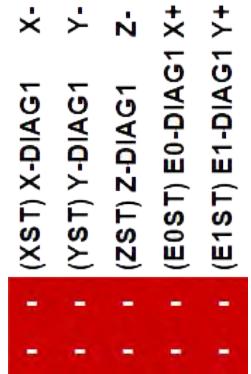
**Note:** To obtain **stand-alone mode** for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as seen in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for **StealthChop** as seen in **RED** below.



BIQU TMC2225 V1.0Stand-alone Mode

# Stand-alone Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode.

XUART



YUART



ZUART



E0 UART



E1 UART



# Stand-alone Mode

## BIQU TMC2225 V1.0

### Stand-alone Mode

## SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers

### Driver Socket Representation

Columns: **1 2 3**



**1/2**

### Driver Chip Chart:

| Driver Chip                                    | MS0                        | MS1  | MS2                        | Excitation Mode |
|------------------------------------------------|----------------------------|------|----------------------------|-----------------|
| LOGO                                           | Low                        | Low  | Low                        | 1-1 Phase       |
| Chip Name                                      | High                       | Low  | Low                        | W1-1 Phase      |
| Maximum XXX                                    | Low                        | High | Low                        | 2W1-2 Phase     |
| Subdivision                                    | High                       | Low  | High                       | 4W1-2 Phase     |
| XXV DC                                         | Low                        | High | High                       | SW1-2 Phase     |
| XXA (peak)                                     | High                       | High | High                       | 16W1-2 Phase    |
| Driving Current Calculation Formula            | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                 |
| $R_g$ (typical) Series Resistor (in $\Omega$ ) |                            |      |                            |                 |

## MS0 for Binary State Drivers:

Driver Socket Representation:

**7 7**

Driver Chip Chart:

**High**

Driver Socket Representation:

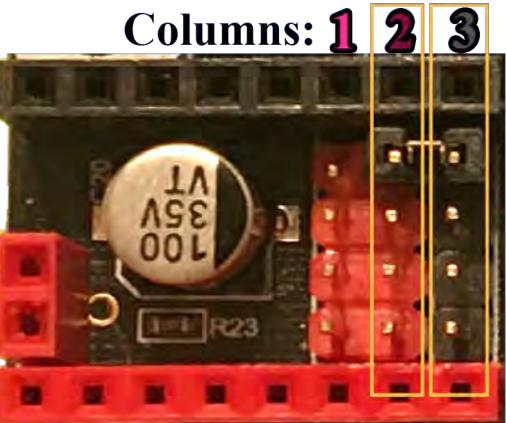
**--**

Driver Chip Chart:

**Low**

## High:

Columns: **1 2 3**



Rows:

**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

### Meaning:

Driver Chip Chart:

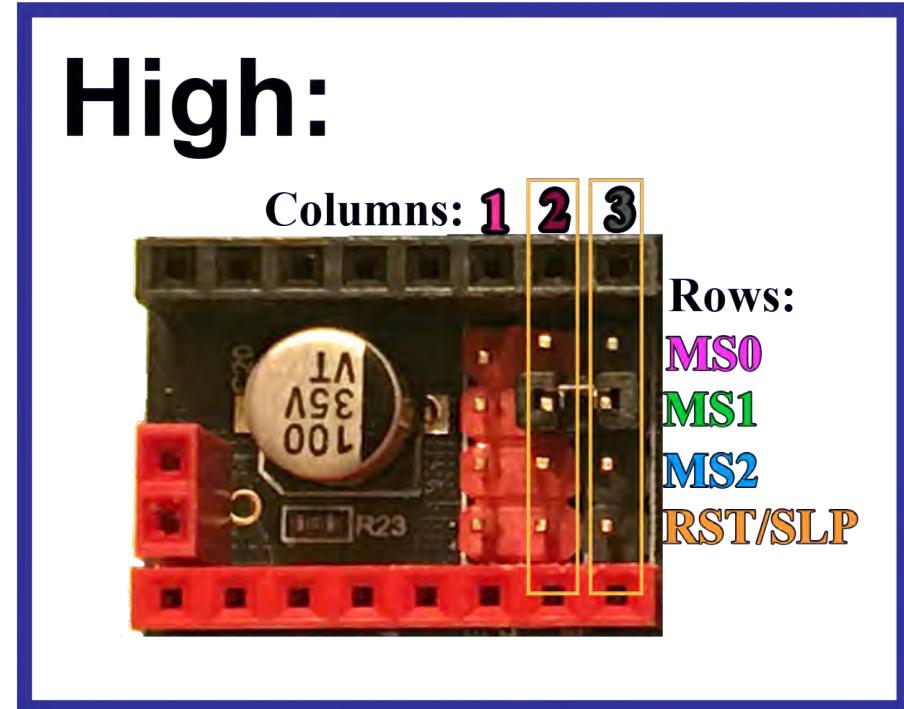
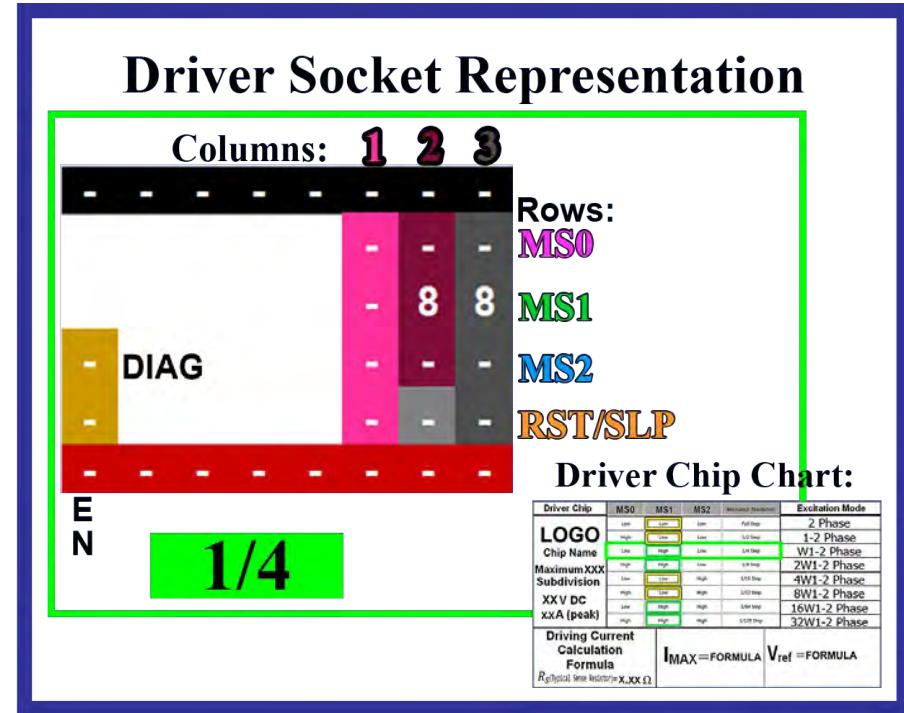
**High**

**set Jumper between column 2 and column 3 on the MS0 row**

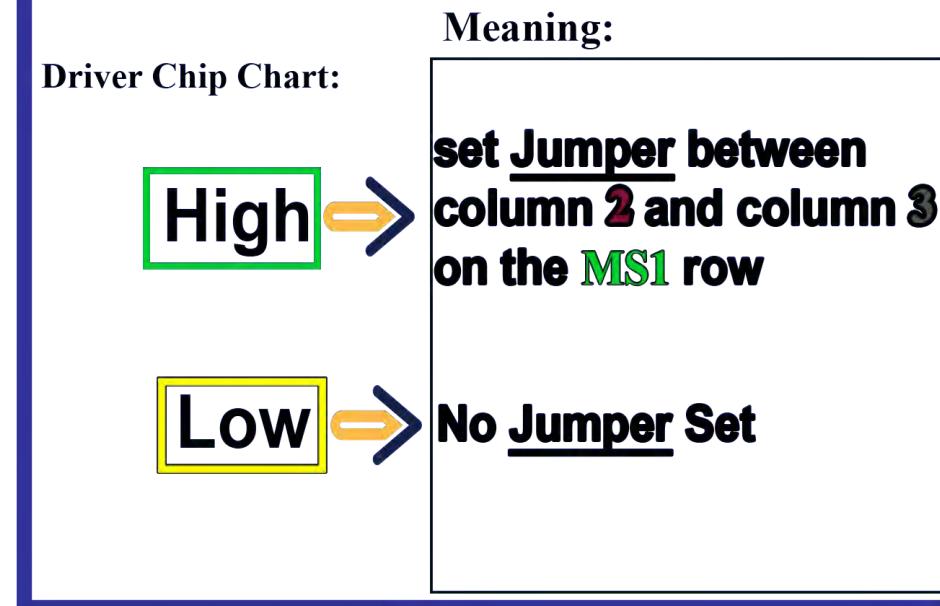
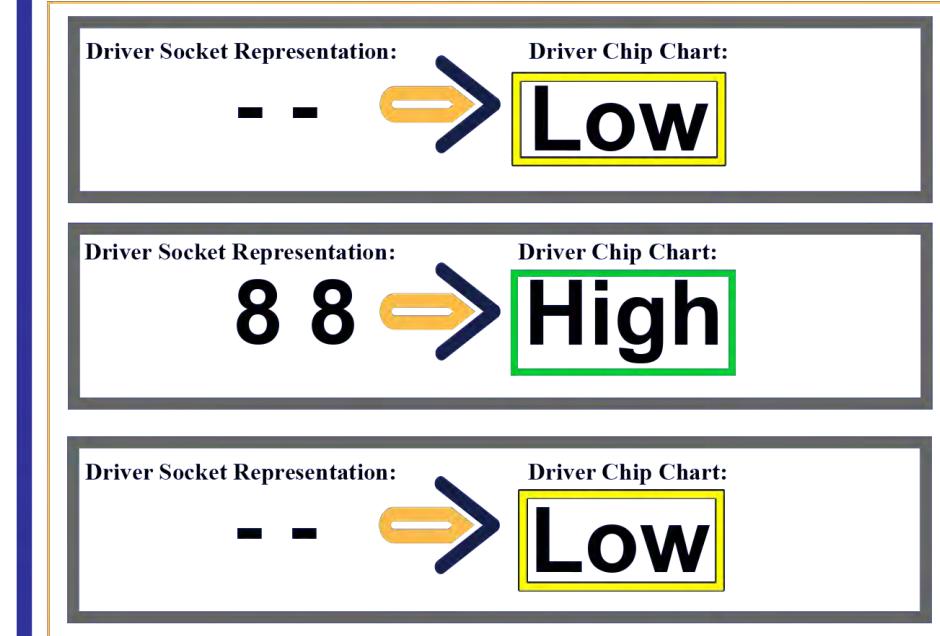
**Low**

**No Jumper Set**

# Stand-alone Mode



## MS1 for Binary State Drivers:



# Stand-alone Mode

## BIQU TMC2225 V1.0

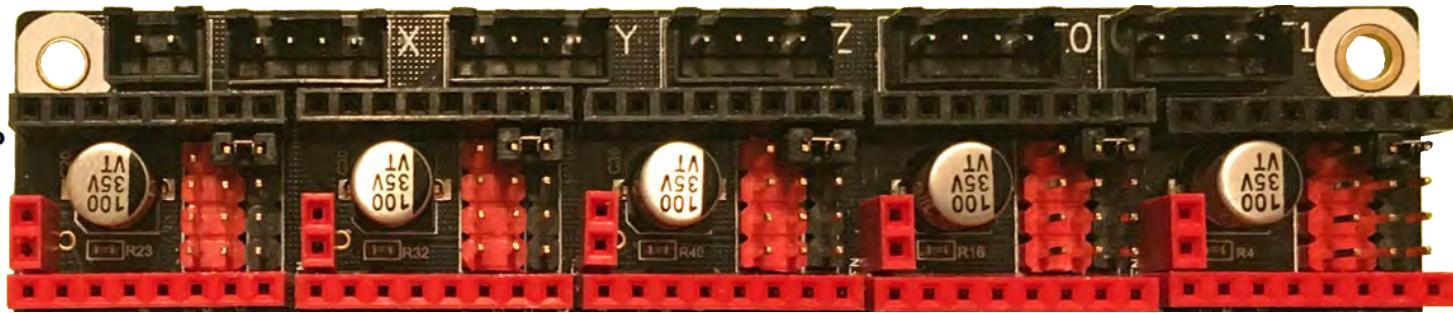
### Stand-alone Mode

| Stand-alone Mode   | VM | GND | A2  | A1   | B1         | B2  | VCC  | GND |
|--------------------|----|-----|-----|------|------------|-----|------|-----|
| <b>1 / 4</b>       |    |     |     |      |            |     | MS0  |     |
| Interpolation:     |    |     |     |      |            |     | MS1  |     |
| <b>1 / 256</b>     |    |     |     |      |            |     | MS2  |     |
| <b>StealthChop</b> | EN | MS1 | MS2 | UART | NC or UART | CLK | STEP | DIR |



See [Appendix D](#) for legend

| Stand-alone Mode   | VM | GND | A2  | A1   | B1         | B2  | VCC  | GND |
|--------------------|----|-----|-----|------|------------|-----|------|-----|
| <b>1 / 8</b>       |    |     |     |      |            |     | MS0  |     |
| Interpolation:     |    |     |     |      |            |     | MS1  |     |
| <b>1 / 256</b>     |    |     |     |      |            |     | MS2  |     |
| <b>StealthChop</b> | EN | MS1 | MS2 | UART | NC or UART | CLK | STEP | DIR |



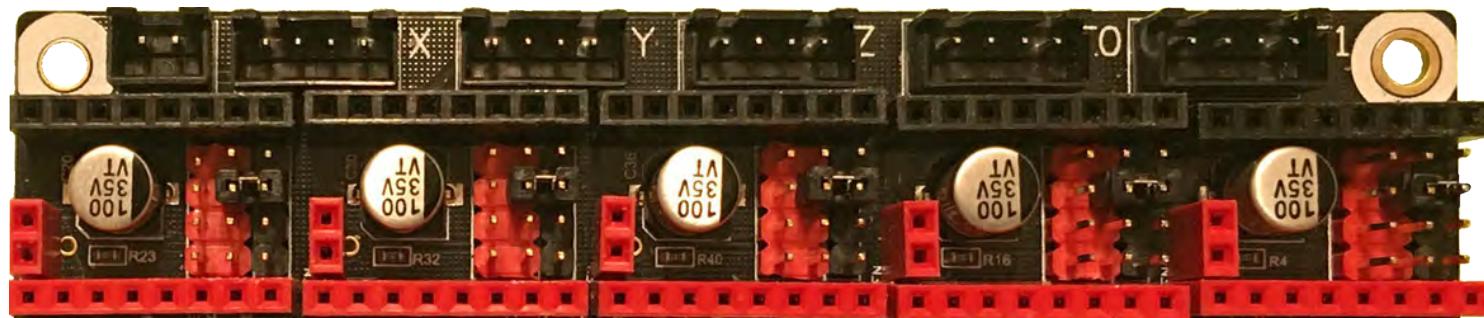
See [Appendix D](#) for legend

# Stand-alone Mode

## BIQU TMC2225 V1.0

### Stand-alone Mode

| Stand-alone Mode | VM | GND | A2  | A1   | B1         | B2  | VCC  | GND |
|------------------|----|-----|-----|------|------------|-----|------|-----|
| <b>1 / 16</b>    |    |     |     |      |            |     | MS0  |     |
| Interpolation:   |    |     |     |      |            |     | MS1  |     |
| <b>1 / 256</b>   |    |     |     |      |            |     | MS2  |     |
| StealthChop      | EN | MS1 | MS2 | UART | NC or UART | CLK | STEP | DIR |



See [Appendix D](#) for legend

| Stand-alone Mode | VM | GND | A2  | A1   | B1         | B2  | VCC  | GND |
|------------------|----|-----|-----|------|------------|-----|------|-----|
| <b>1 / 32</b>    |    |     |     |      |            |     | MS0  |     |
| Interpolation:   |    |     |     |      |            |     | MS1  |     |
| <b>1 / 256</b>   |    |     |     |      |            |     | MS2  |     |
| StealthChop      | EN | MS1 | MS2 | UART | NC or UART | CLK | STEP | DIR |

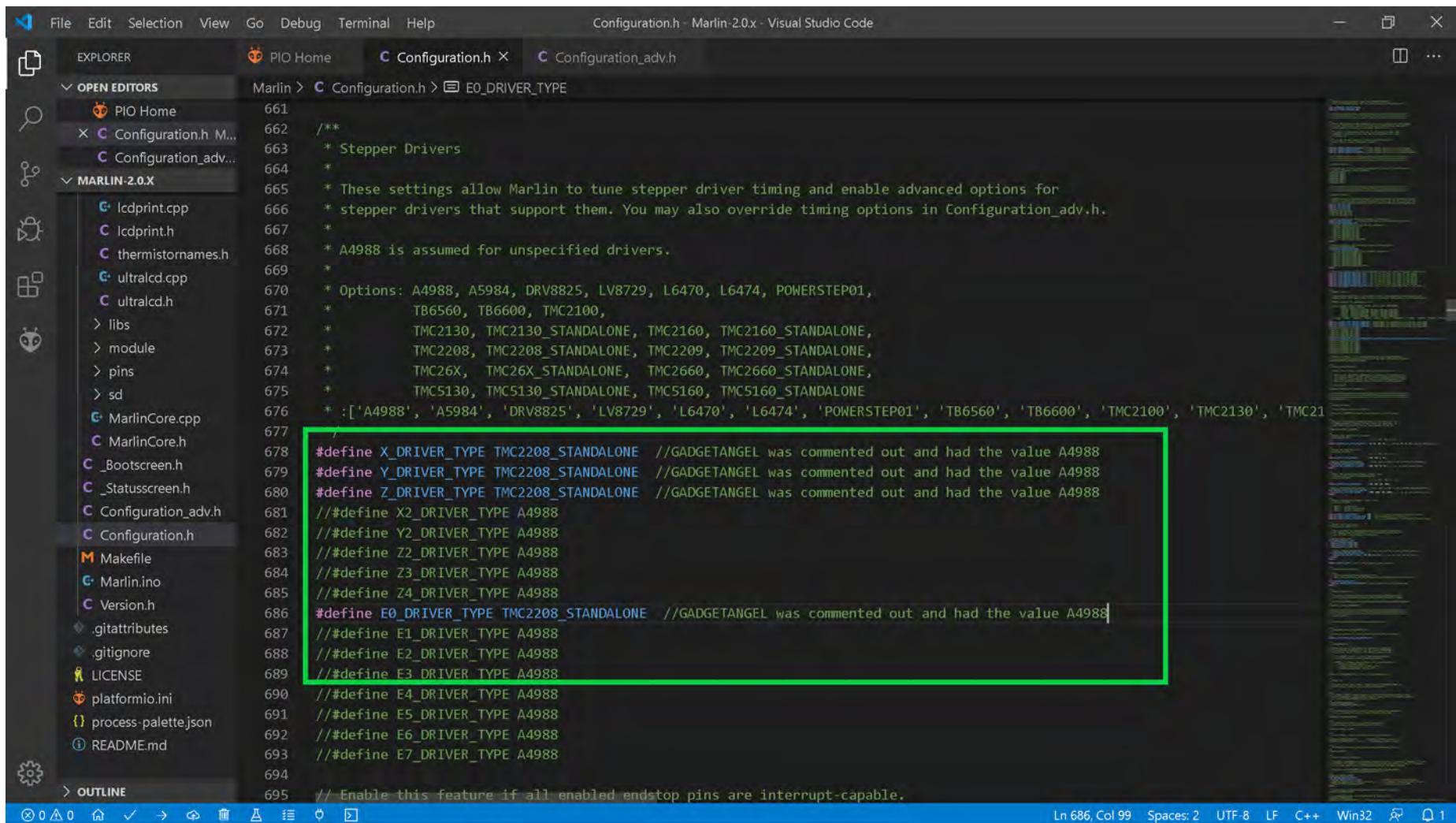


See [Appendix D](#) for legend

# The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in stand-alone mode.

- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2225 drivers in stand-alone mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208\_STANDALONE" for the DRIVER\_TYPE. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following relevant snippet:

```

661 //**
662 * Stepper Drivers
663 */
664 */
665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
667 */
668 */
669 */
670 */
671 */
672 */
673 */
674 */
675 */
676 */
677 */
678 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
681 //##define X2_DRIVER_TYPE A4988
682 //##define Y2_DRIVER_TYPE A4988
683 //##define Z2_DRIVER_TYPE A4988
684 //##define Z3_DRIVER_TYPE A4988
685 //##define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
687 //##define E1_DRIVER_TYPE A4988
688 //##define E2_DRIVER_TYPE A4988
689 //##define E3_DRIVER_TYPE A4988
690 //##define E4_DRIVER_TYPE A4988
691 //##define E5_DRIVER_TYPE A4988
692 //##define E6_DRIVER_TYPE A4988
693 //##define E7_DRIVER_TYPE A4988
694 */
695 */

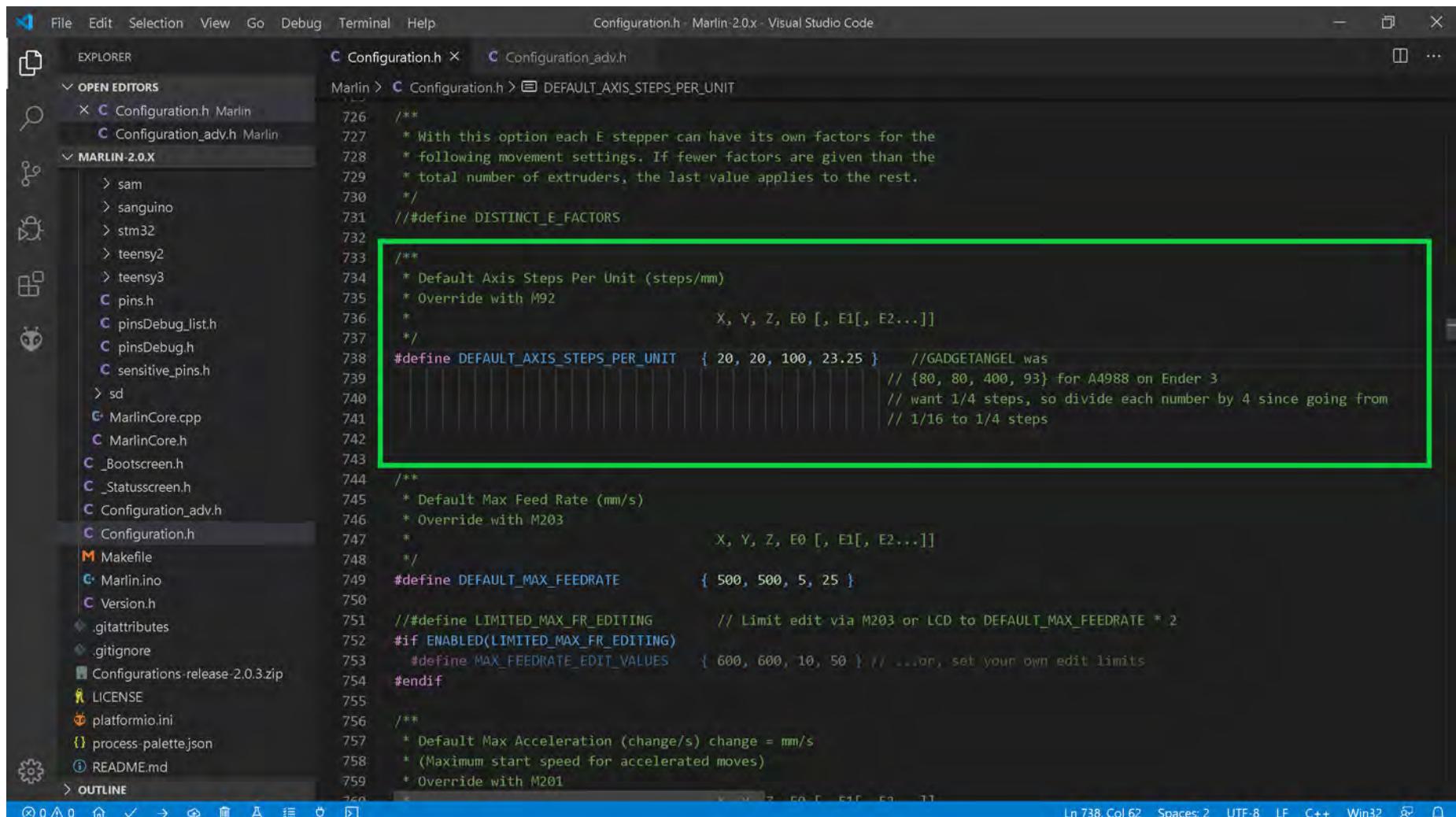
```

A green rectangular box highlights the lines defining the driver type for all axes (X, Y, Z, E0-E7) as 'TMC2208\_STANDALONE'. The original values ('A4988') are visible within the comments.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the Marlin 2.0.x configuration file. A green rectangular box highlights the following code block:

```

726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS

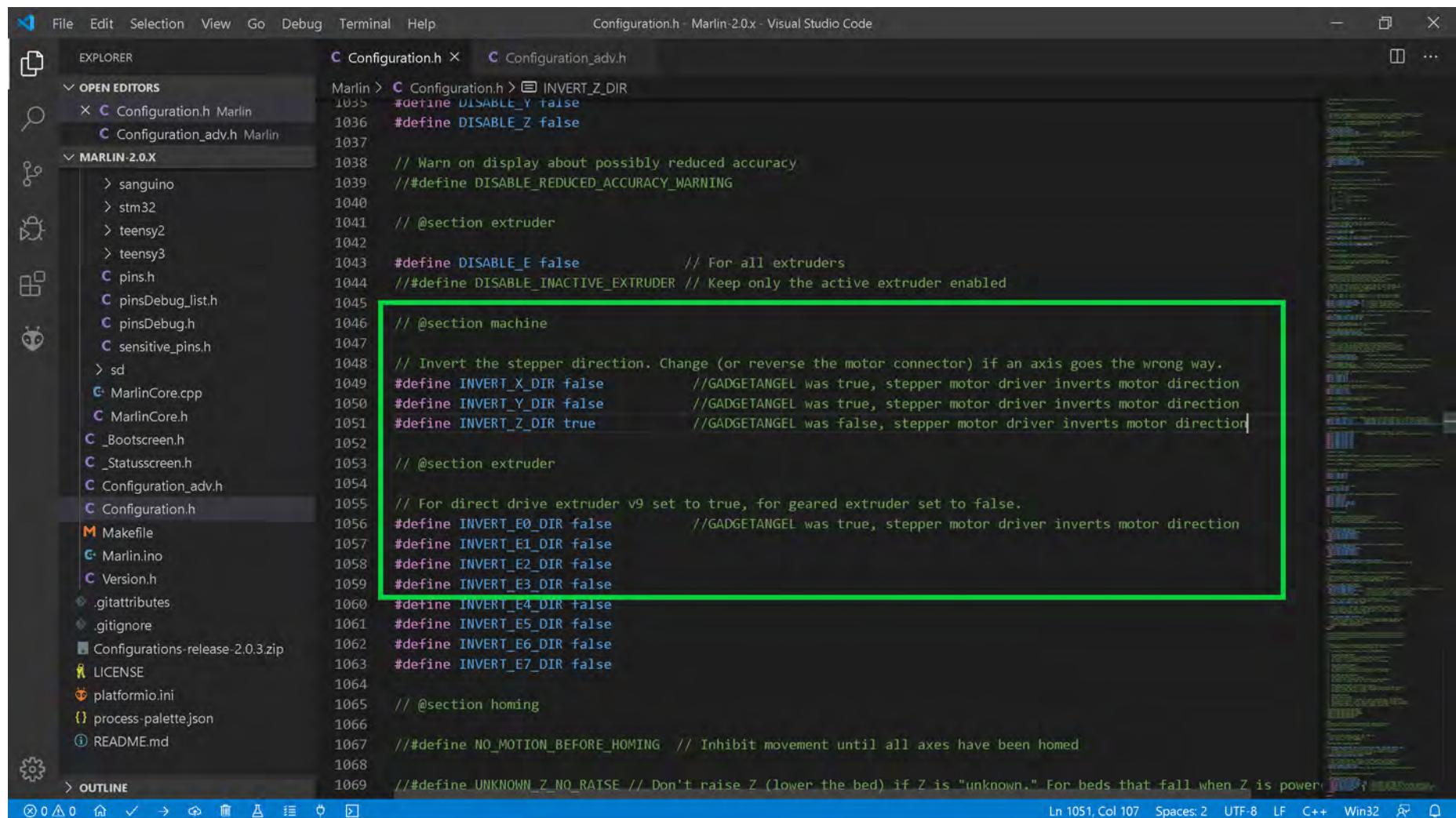
732 /**
733 * Default Axis Steps Per Unit (steps/mm)
734 * Override with M92
735 *
736 * X, Y, Z, E0 [, E1[, E2...]]
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // want 1/4 steps, so divide each number by 4 since going from
741 // 1/16 to 1/4 steps
742
743 /**
744 * Default Max Feed Rate (mm/s)
745 * Override with M203
746 *
747 * X, Y, Z, E0 [, E1[, E2...]]
748 */
749 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }
750
751 // #define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
752 #if ENABLED(LIMITED_MAX_FR_EDITING)
753 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ... or, set your own edit limits
754 #endif
755
756 /**
757 * Default Max Acceleration (change/s) change = mm/s
758 * (Maximum start speed for accelerated moves)
759 * Override with M201
760 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2225 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code is part of the Marlin 2.0.x repository. A green rectangular box highlights a section of the code where the `INVERT_Z_DIR` macro is defined. The code snippet is as follows:

```

1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

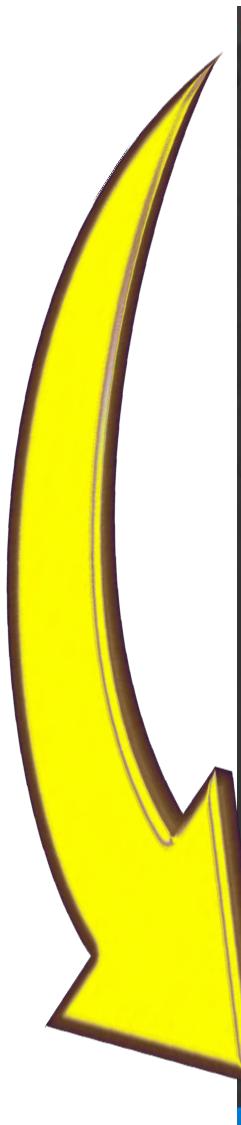
```

The green box covers lines 1049 through 1053, specifically the definitions for `INVERT_X_DIR`, `INVERT_Y_DIR`, and `INVERT_Z_DIR`. The original code had `false` for `X` and `Y`, and `true` for `Z`. The green box highlights the changes made by GADGETANGEL to invert the `Z` direction.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in stand-alone mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



**Configuration.h - Marlin-2.0.x - Visual Studio Code**

**EXPLORER**

- OPEN EDITORS
  - Configuration.h Marlin
  - pins\_BTT\_SKR\_V1\_3.h Marlin\src...
  - pins\_BTT\_SKR\_common.h Marlin...
  - Configuration\_adv.h Marlin
- MARLIN-2.0.X
  - samd
  - sanguino
  - stm32f1
  - stm32f4
  - stm32f7
  - teensy2
  - teensy3
  - pins.h
  - pinsDebug.h
  - pinsDebug\_list.h
  - sensitive\_pins.h
  - sd
  - MarlinCore.cpp
  - MarlinCore.h
  - \_Bootscreen.h
  - \_Statusscreen.h
  - Configuration.h
  - Configuration\_adv.h
  - Makefile
  - Marlin.ino
  - Version.h
  - .editorconfig
  - .gitattributes
  - .gitignore
  - LICENSE
  - platformio.ini
  - process-palette.json
  - README.md

**TERMINAL**

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h > ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the name below to that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

**PROBLEMS** 1 OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| ramps           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino44p     | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| PIC16F          | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in Stand-alone Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

**Configuration.h - Marlin-2.0.x - Visual Studio Code**

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE

```

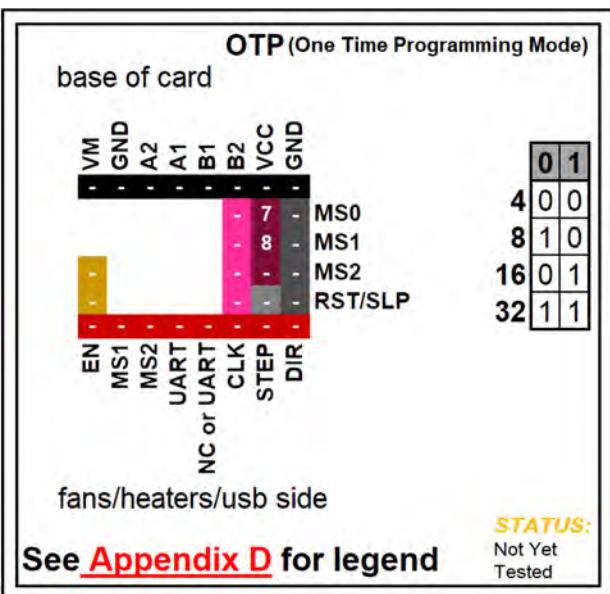
```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds
Environment Status Duration

mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino644p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_LVDS IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC1769 IGNORED
STM32F103RC IGNORED

```

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

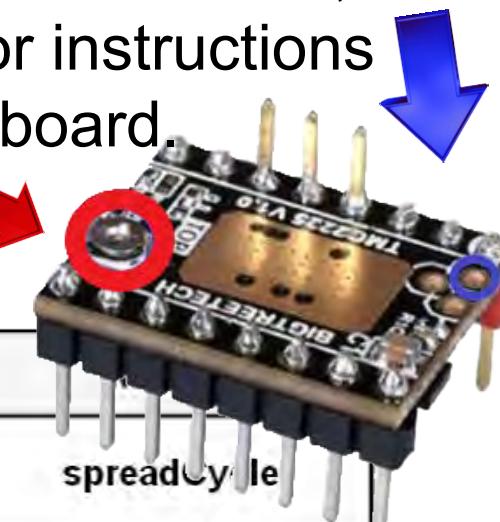


## BIQU TMC2225 V1.0

### One Time Programming (OTP) Mode

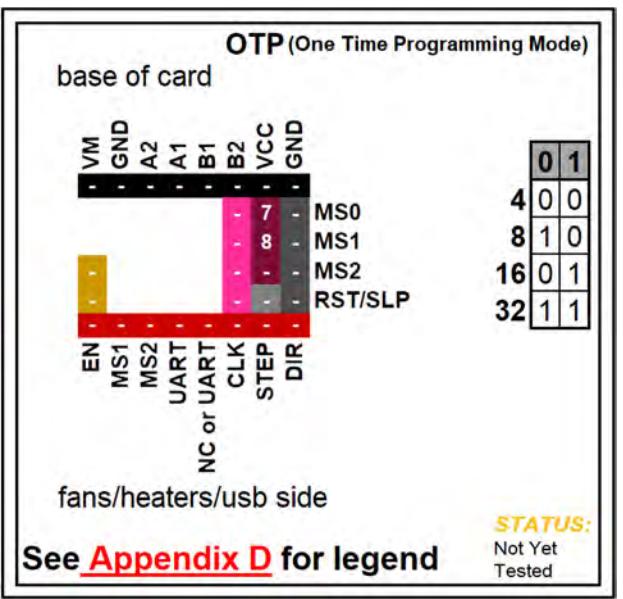
**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See [Appendix A](#) for instructions on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.



| Driver Chip                                                                                              | MS0                                                                                                | MS1  | Steps  | Interpolation                                                                                      |             |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|------|--------|----------------------------------------------------------------------------------------------------|-------------|
| <b>BIQU®<br/>TMC2225</b><br><small>OTP Mode<br/>Maximum 32 Subdivision<br/>35V DC<br/>2A (peak)</small>  | Low                                                                                                | Low  | 1 / 4  | 1 / 256                                                                                            | spreadCycle |
|                                                                                                          | High                                                                                               | Low  | 1 / 8  | 1 / 256                                                                                            | spreadCycle |
|                                                                                                          | Low                                                                                                | High | 1 / 16 | 1 / 256                                                                                            | spreadCycle |
|                                                                                                          | High                                                                                               | High | 1 / 32 | 1 / 256                                                                                            | spreadCycle |
| <b>Driving Current Calculation Formula</b><br><small>R<sub>S</sub>(Typical Sense Resistor)=0.15Ω</small> | <b>I<sub>MAX</sub>=V<sub>ref</sub>*0.7222</b><br>See Appendix B#10. Use 50% to 90% as shown below: |      |        | <b>V<sub>ref</sub>=I<sub>MAX</sub>*1.3846</b><br>See Appendix B#10. Use 50% to 90% as shown below: |             |
|                                                                                                          | <b>I<sub>MAX</sub>=(V<sub>ref</sub>*0.7222)*0.90</b>                                               |      |        | <b>V<sub>ref</sub>=(I<sub>MAX</sub>*1.3846)*0.90</b>                                               |             |

- See next page for the legend that belongs to the above chart.



### Driver Chip Chart:

| Driver Chip                                  | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|----------------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b>                                  | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
| Chip Name                                    | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision                      | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
| XXV DC xxA (peak)                            | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                              | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                              | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                              | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                              | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula          | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |
| $R_S$ (typical Sense Resistor)=x.XX $\Omega$ |                            |      |                            |                      |                 |

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

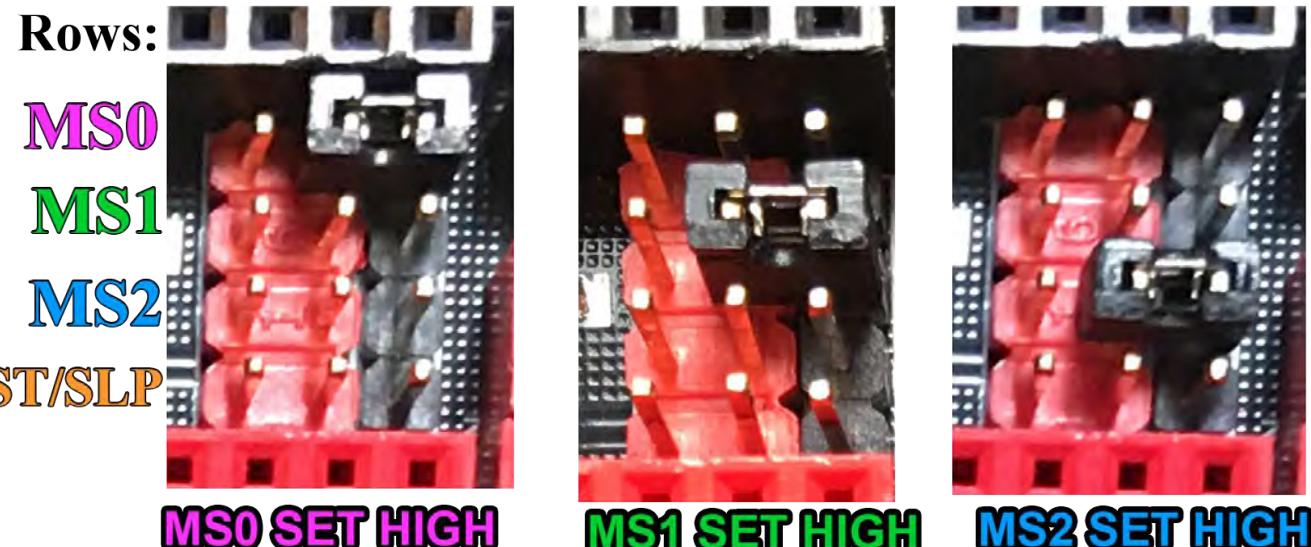
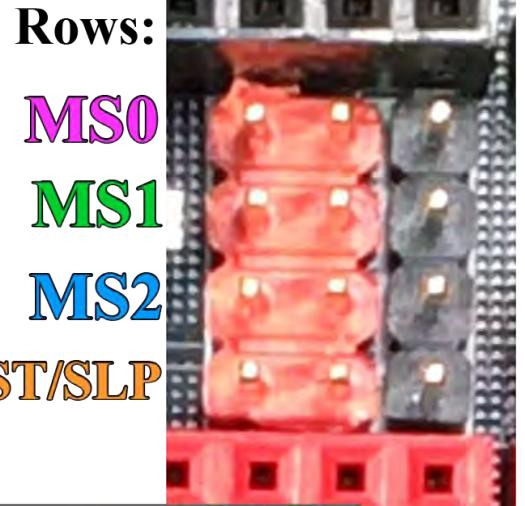
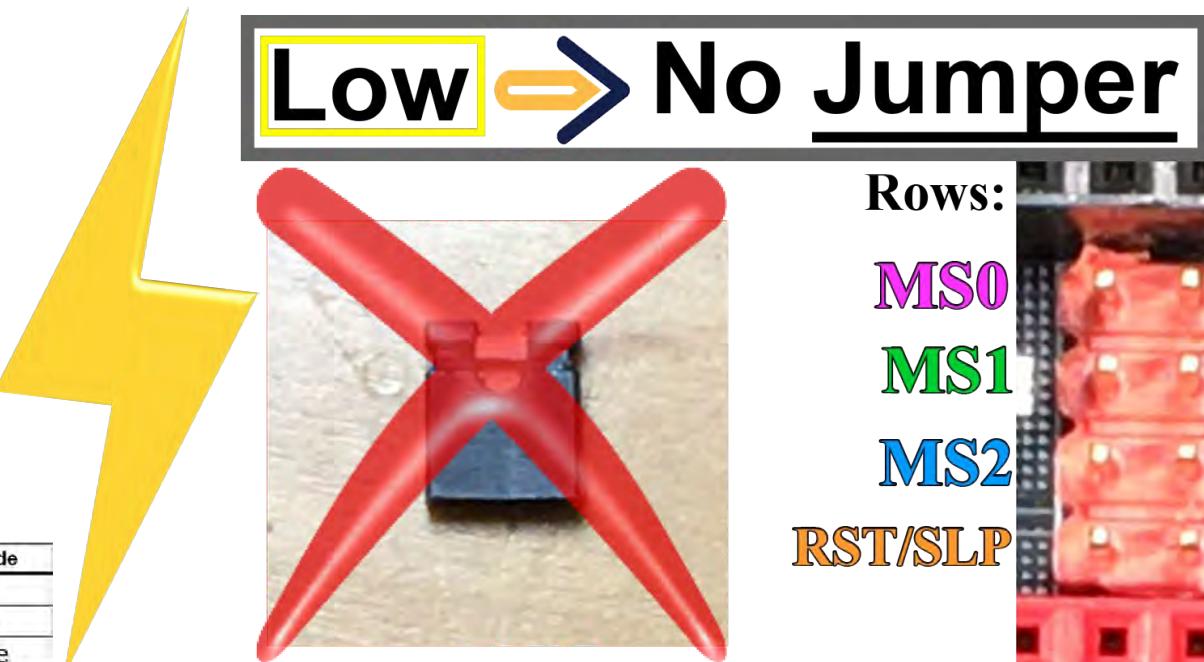


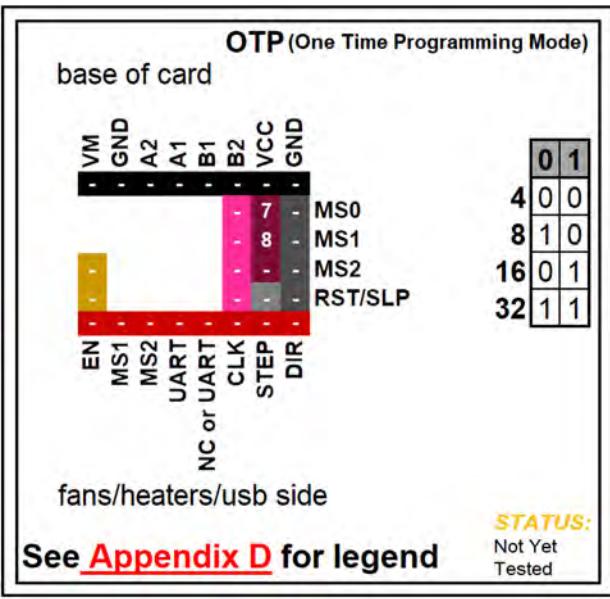
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):



### BIQU TMC2225 V1.0 One Time Programming (OTP) Mode

### SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers



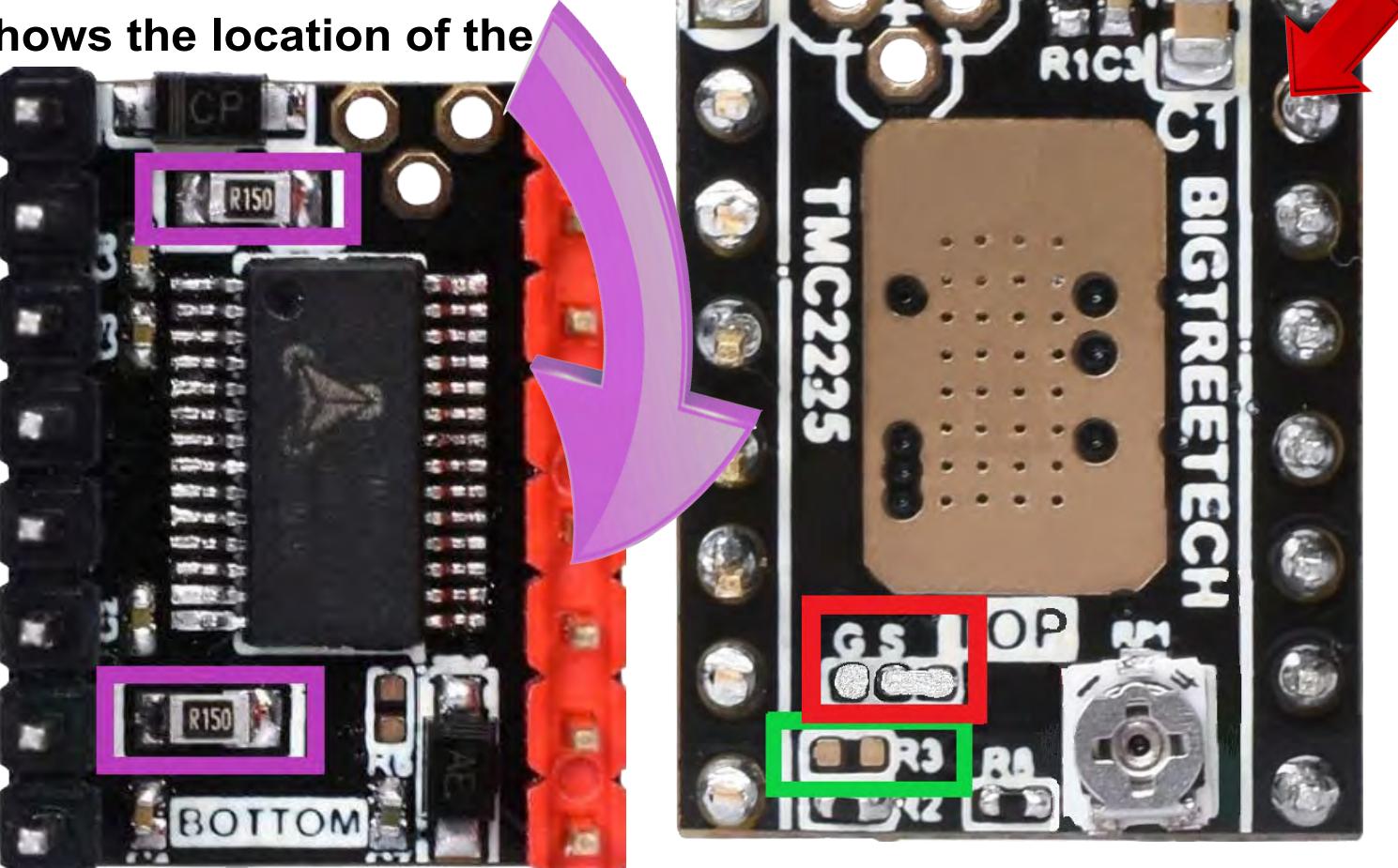


**Note:** Stand-alone mode by default uses stealthChop, if you want spreadCycle, you MUST use OTP mode. To obtain **One Time Programming (OTP) mode**, for the BIQU TMC2225 V1.0, the two pads located at R3 must have a gap between them, as seen in **GREEN** below, and the two pads at "G S" (located on the top of the driver board) must be set for spread Cycle as seen in **RED** below.

To do One-Time-Programming (OTP), the TMC2225 must be placed in UART mode. See [TMC220x Configurator](#) for One-Time-Programming Information: [TMC220x Configurator](#).

The **PURPLE** boxes shows the location of the current sense resistors ( $R_s$ ).

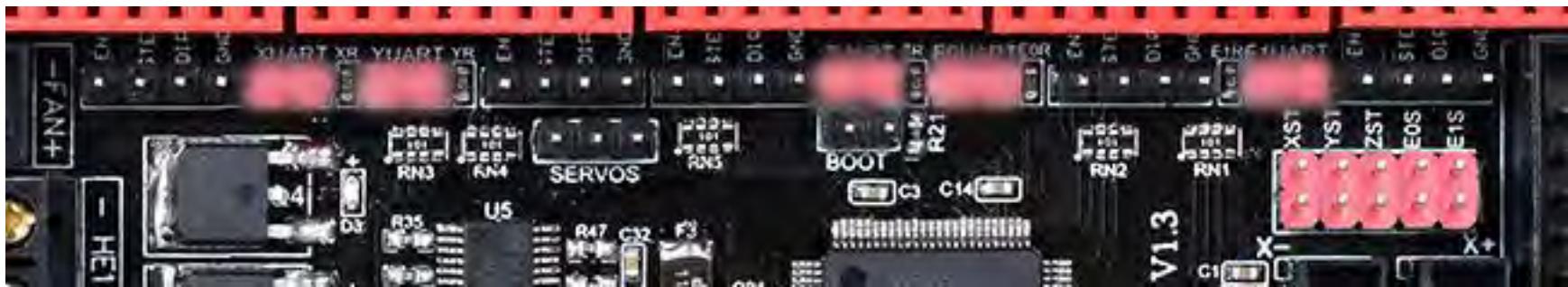
**Note:** MOST BIQU TMC2225 V1.0 driver boards, when purchased for UART mode, will have two R3 pads (located on the top of the driver board), which are NOT soldered together. This indicates the driver board can use the UART pin for the UART single wire interface (if the UART,"U", jumper is in place on the SKR V1.3 board)



## OTP (One Time Programming) Mode One Time Programming (OTP) Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation.

|                    |               |               |                    |
|--------------------|---------------|---------------|--------------------|
| X-                 | Y-            | Z-            | X+                 |
| (XST) X-DIAG1      | (YST) Y-DIAG1 | (ZST) Z-DIAG1 | (E0ST) E0-DIAG1 X+ |
| - - - - -          | - - - - -     | - - - - -     | - - - - -          |
| (E1ST) E1-DIAG1 Y+ |               |               |                    |



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation.

XUART

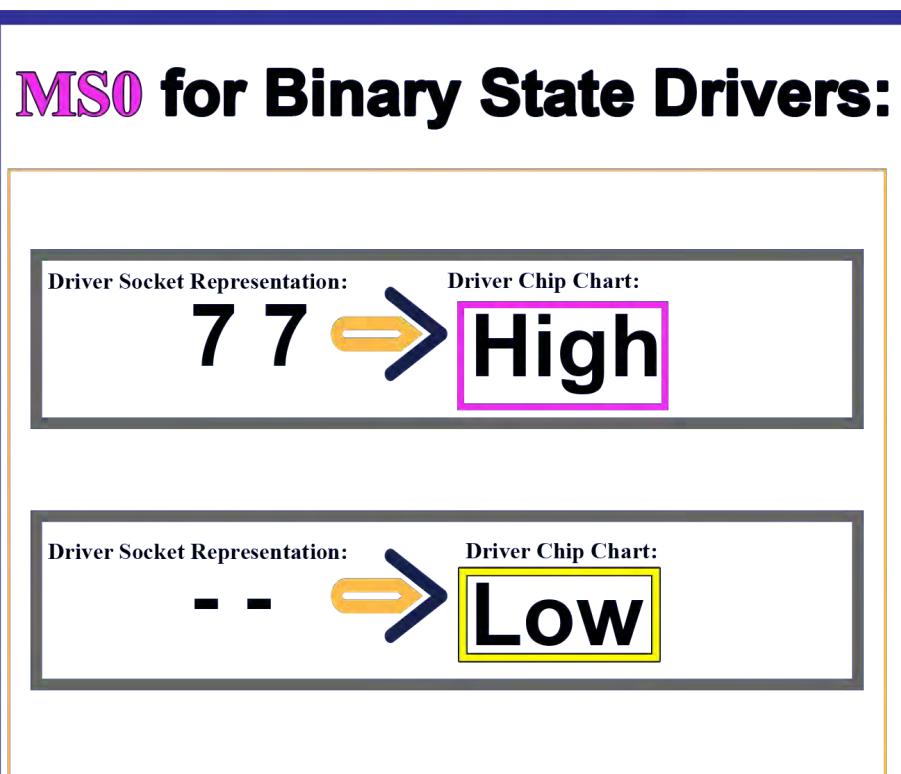
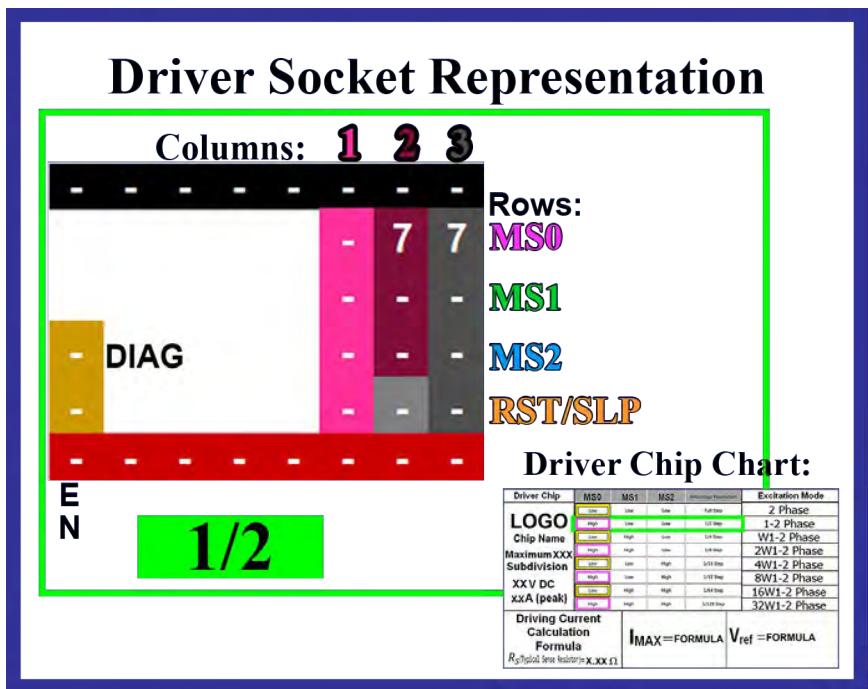
YUART

ZUART

E0 UART

E1 UART



**OTP (One Time Programming) Mode** One Time Programming (OTP) Mode**SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers**

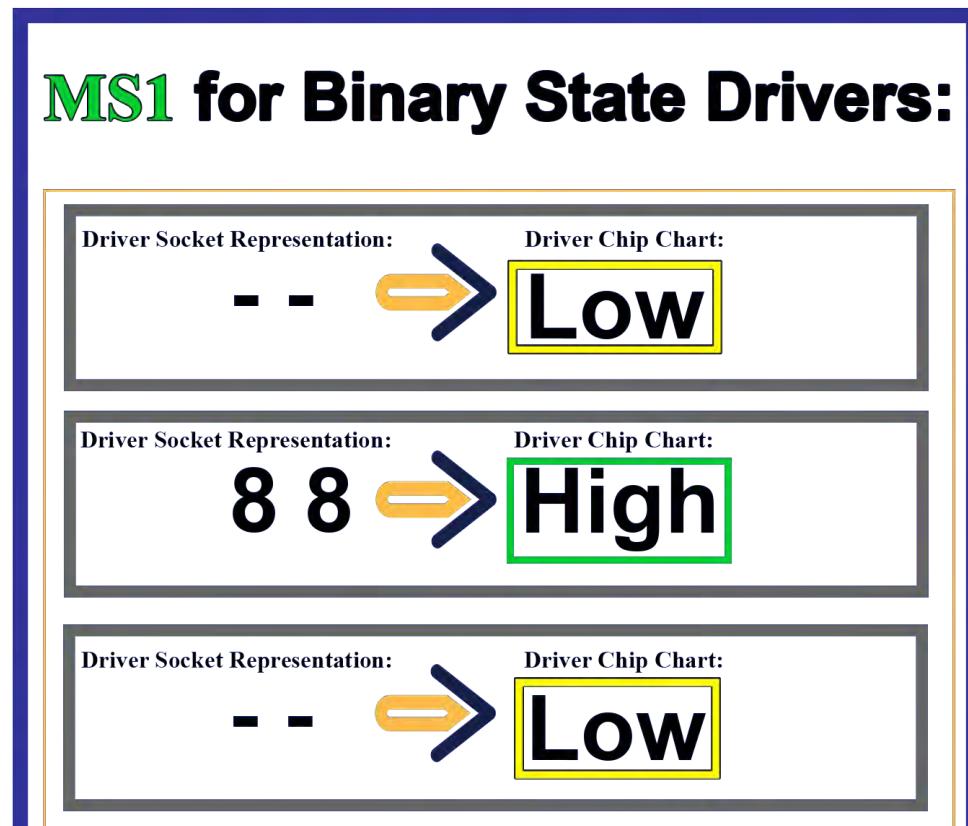
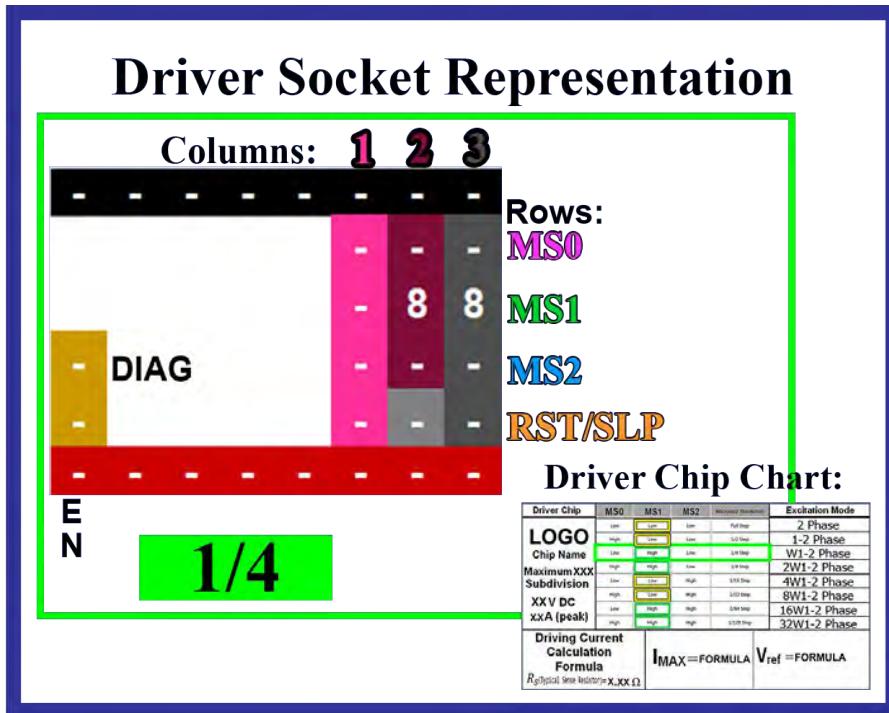
**Meaning:**

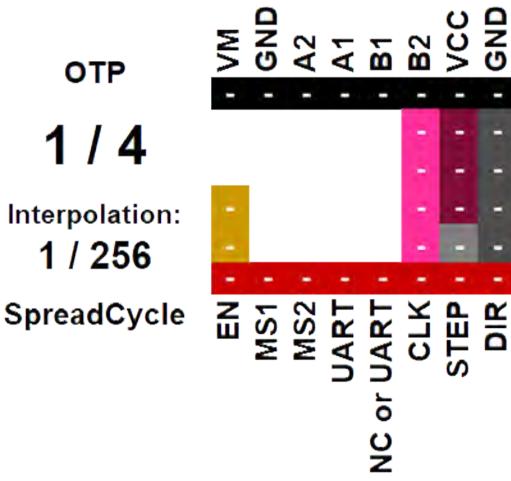
**Driver Chip Chart:**

**High** → **set Jumper between column 2 and column 3 on the MS0 row**

**Low** → **No Jumper Set**

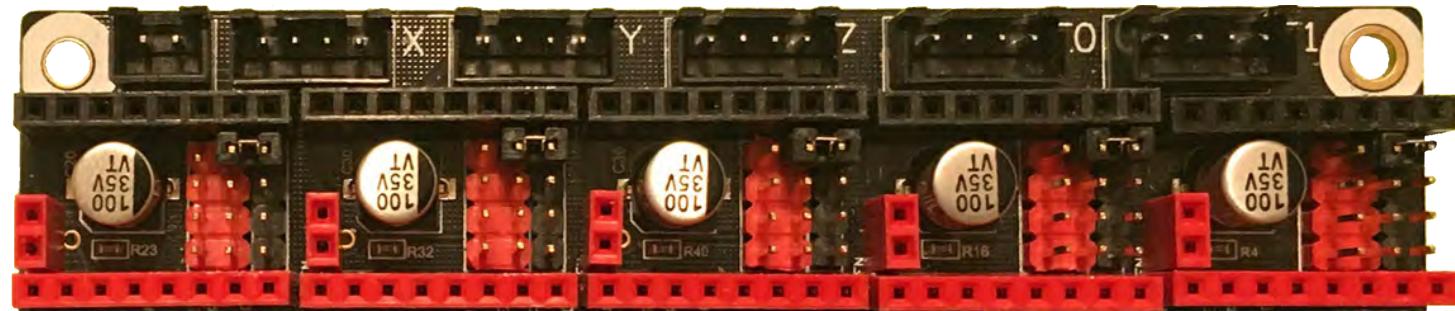
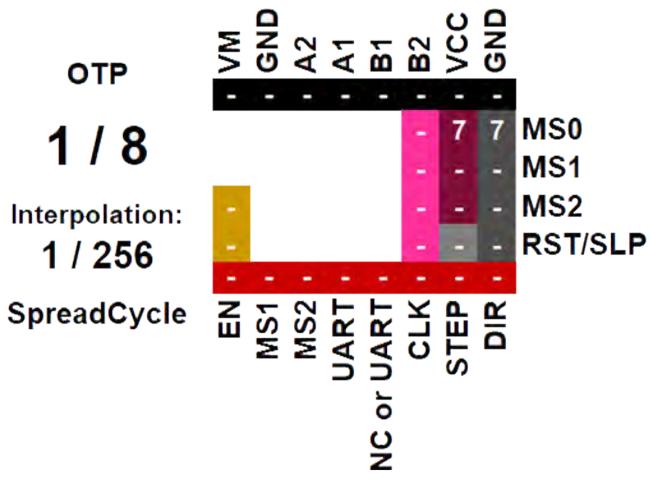
## OTP (One Time Programming) Mode One Time Programming (OTP) Mode



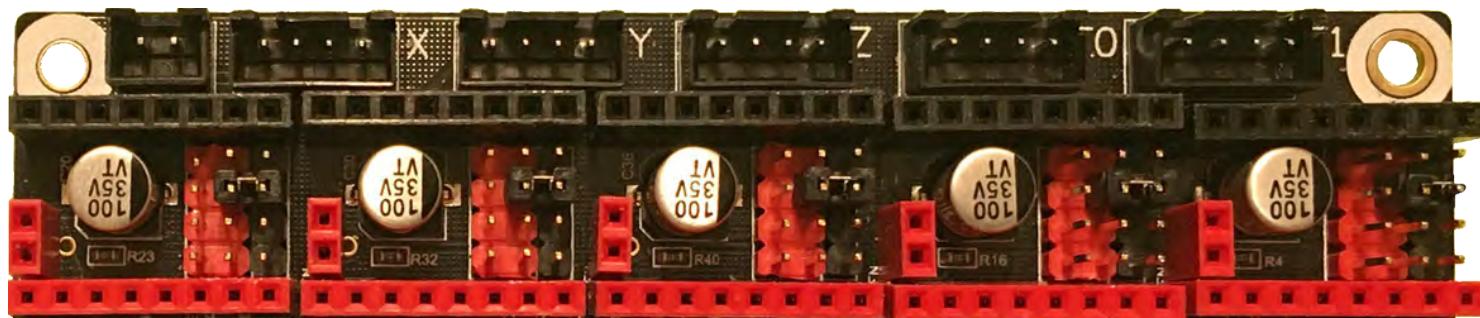
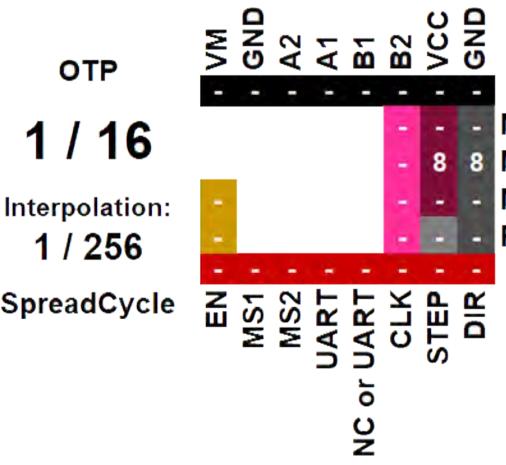
**BIQU TMC2225 V1.0****OTP (One Time Programming) Mode** [One Time Programming \(OTP\) Mode](#)

See [Appendix D](#) for legend

---

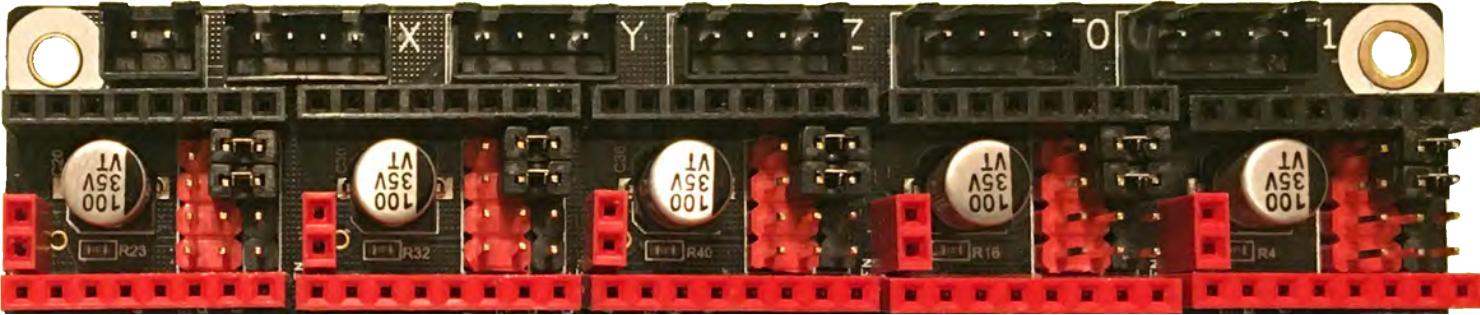
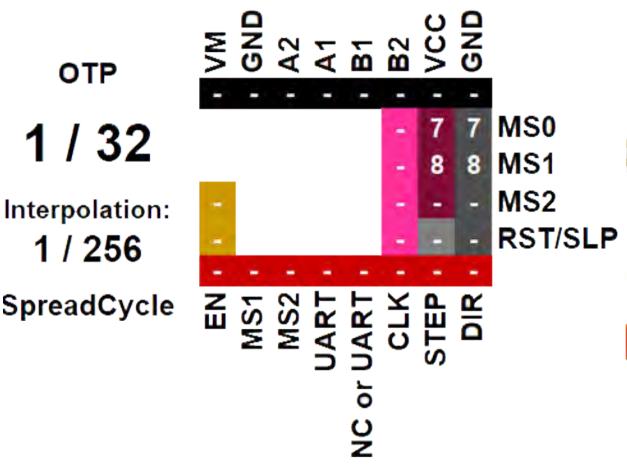


See [Appendix D](#) for legend

**BIQU TMC2225 V1.0****OTP (One Time Programming) Mode** **One Time Programming (OTP) Mode**

See [Appendix D](#) for legend

---

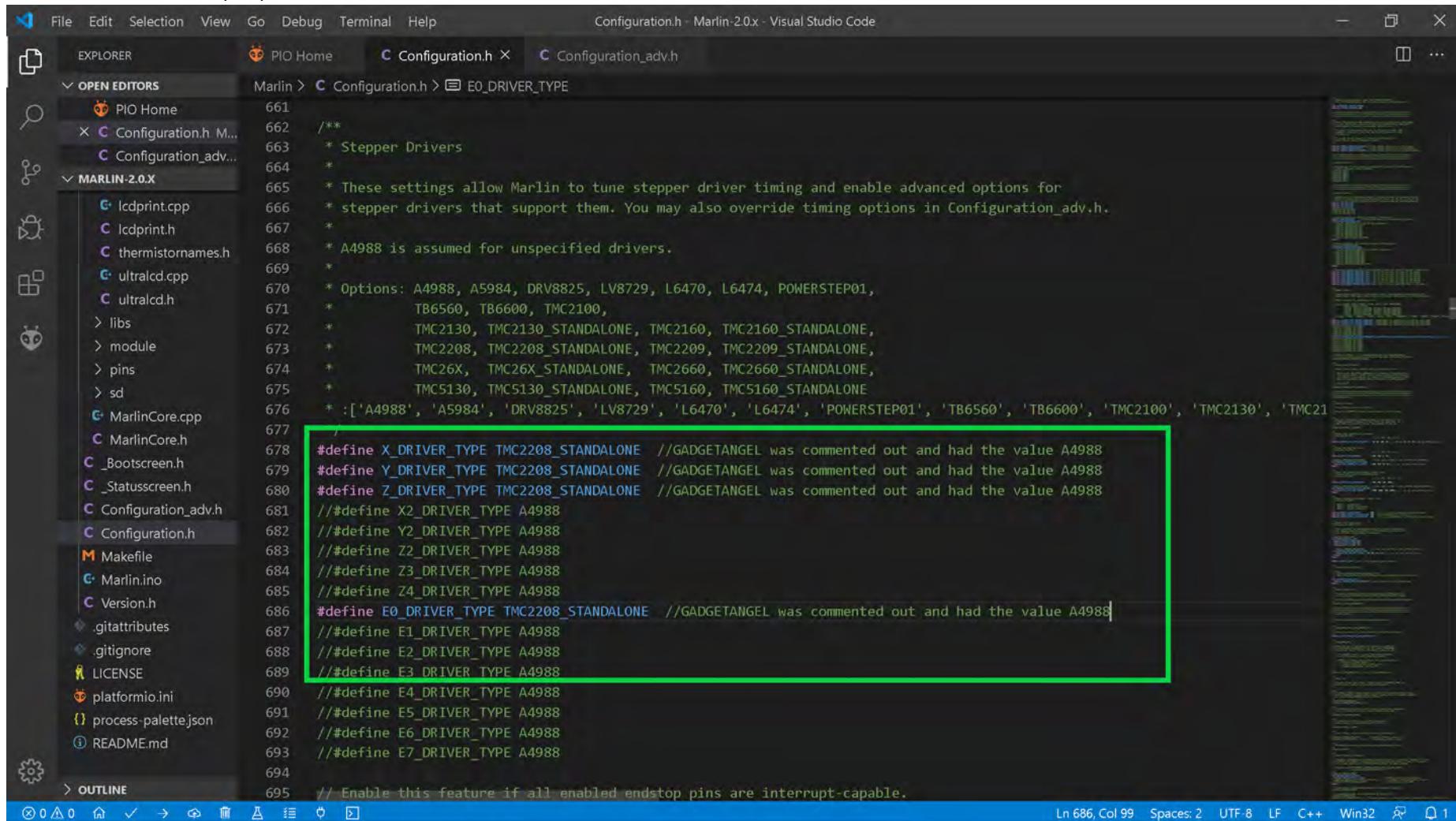


See [Appendix D](#) for legend

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in OTP mode.

- Change the stepper motor drivers so that Marlin knows you are using BIQU TMC2225 drivers in OTP mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use BIQU TMC2225 drivers in OTP mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208\_STANDALONE" for the DRIVER\_TYPE. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
 661 /**
 662 * Stepper Drivers
 663 *
 664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 666 * A4988 is assumed for unspecified drivers.
 667 *
 668 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 669 * TB6560, TB6600, TMC2100,
 670 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 671 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 672 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 673 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 674 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2
 675 */
 676 #define X_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
 677 #define Y_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
 678 #define Z_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
 679 //##define X2_DRIVER_TYPE A4988
 680 //##define Y2_DRIVER_TYPE A4988
 681 //##define Z2_DRIVER_TYPE A4988
 682 //##define Z3_DRIVER_TYPE A4988
 683 //##define Z4_DRIVER_TYPE A4988
 684 #define E0_DRIVER_TYPE TMC2208_STANDALONE //GADGETANGEL was commented out and had the value A4988
 685 //##define E1_DRIVER_TYPE A4988
 686 //##define E2_DRIVER_TYPE A4988
 687 //##define E3_DRIVER_TYPE A4988
 688 //##define E4_DRIVER_TYPE A4988
 689 //##define E5_DRIVER_TYPE A4988
 690 //##define E6_DRIVER_TYPE A4988
 691 //##define E7_DRIVER_TYPE A4988
 692 //##define E8_DRIVER_TYPE A4988
 693 //##define E9_DRIVER_TYPE A4988
 694 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- Since I desire to use 1/4 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2225 (which are exactly like the TMC2208) stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/4 stepping. So we are cutting our STEPS by one quarter. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {20, 20, 100, 23.25}, as seen in the GREEN box below.

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin Configuration.h DEFAULT_AXIS_STEPS_PER_UNIT
sam
sanguino
stm32
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
Configurations-release-2.0.3.zip
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
Ln 738, Col 62 Spaces: 2 UTF-8 LF C++ Win32

```

Configuration.h - Marlin-2.0.x - Visual Studio Code

```

266 /**
267 * With this option each E stepper can have its own factors for the
268 * following movement settings. If fewer factors are given than the
269 * total number of extruders, the last value applies to the rest.
270 */
271 //#define DISTINCT_E_FACTORS

272 /**
273 * Default Axis Steps Per Unit (steps/mm)
274 * Override with M92
275 *
276 * X, Y, Z, E0 [, E1[, E2...]]
277 */
278 #define DEFAULT_AXIS_STEPS_PER_UNIT { 20, 20, 100, 23.25 } //GADGETANGEL was
279 // {80, 80, 400, 93} for A4988 on Ender 3
280 // want 1/4 steps, so divide each number by 4 since going from
281 // 1/16 to 1/4 steps

282 /**
283 * Default Max Feed Rate (mm/s)
284 * Override with M203
285 *
286 * X, Y, Z, E0 [, E1[, E2...]]
287 */
288 #define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

289 //define LIMITED_MAX_FR_EDITING // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
290 #if ENABLED(LIMITED_MAX_FR_EDITING)
291 #define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
292 #endif

293 /**
294 * Default Max Acceleration (change/s) change = mm/s
295 * (Maximum start speed for accelerated moves)
296 * Override with M201
297 */
298
```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below

The screenshot shows the Visual Studio Code interface with the following details:

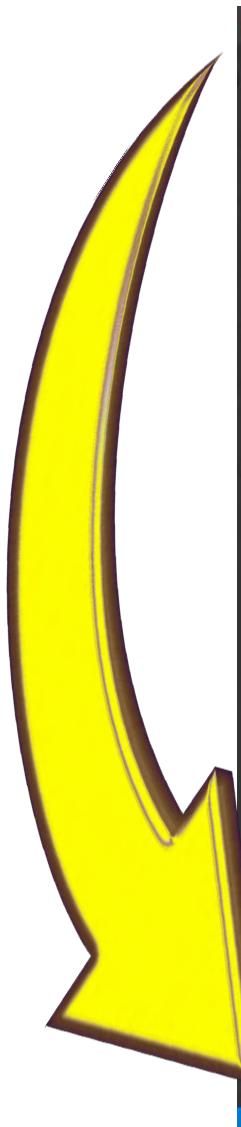
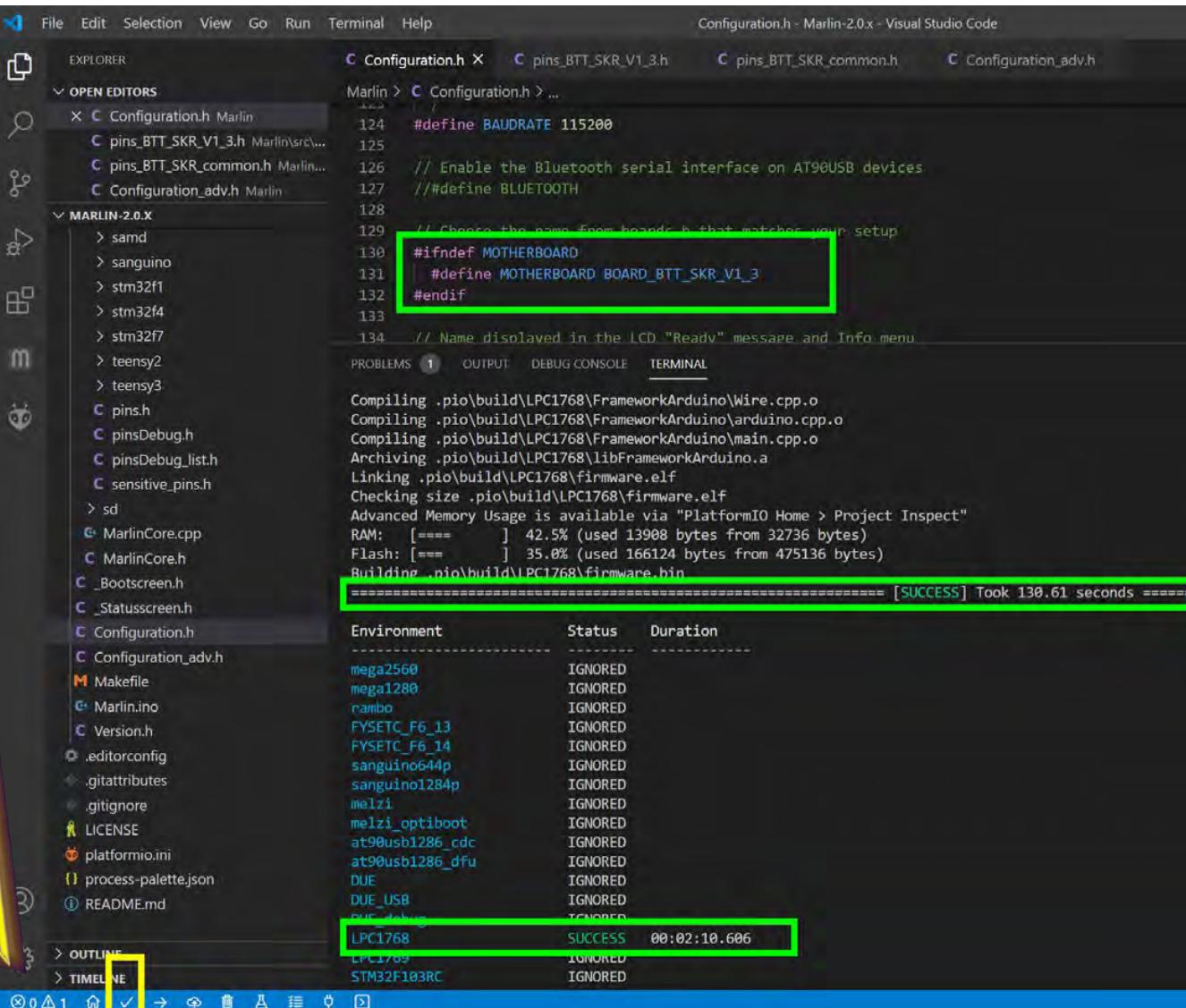
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar (EXPLORER):** Shows the project structure under MARLIN-2.0.X, including files like Configuration.h, Configuration\_adv.h, sanguino, stm32, teensy2, teensy3, pins.h, pinsDebug\_list.h, pinsDebug.h, sensitive\_pins.h, sd, MarlinCore.cpp, MarlinCore.h, \_Bootscreen.h, \_Statusscreen.h, Configuration\_adv.h, Configuration.h, Makefile, Marlin.ino, Version.h, .gitattributes, .gitignore, Configurations-release-2.0.3.zip, LICENSE, platformio.ini, process-palette.json, README.md, and OUTLINE.
- Code Editor:** The Configuration.h file is open. A specific section of the code is highlighted with a green box:

```
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
```
- Bottom Status Bar:** Ln 1051, Col 107, Spaces: 2, UFT-8, LF, C++, Win32.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in OTP mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif

// Name displayed in the LCD "Ready" message and Info menu

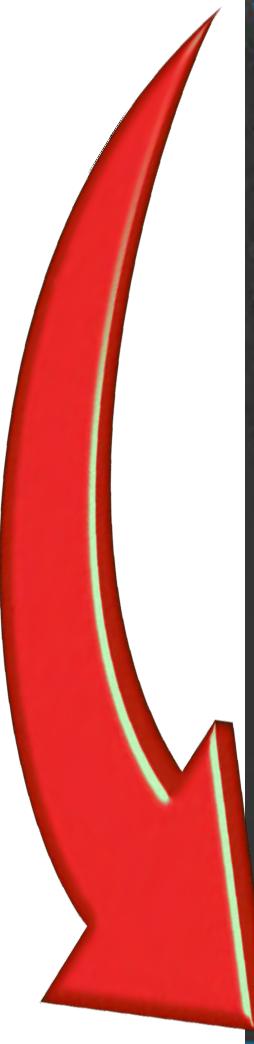
===== [SUCCESS] Took 130.61 seconds =====

Environment Status Duration
----- -----
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguinob44p IGNORED
sanguinol284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
TC3588E IGNORED
LPC1768 SUCCESS 00:02:10.606
LPC4709 IGNORED
STM32F103RC IGNORED
```

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in One Time Programming (OTP) Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.







File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src\pins\pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Marlin\src\pins\pins\_BTT\_SKR\_common.h Configuration\_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

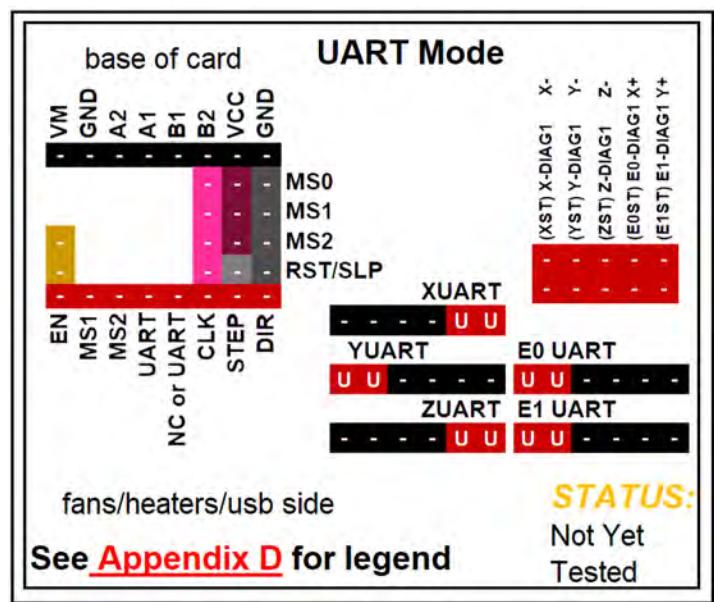
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUET_3D         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

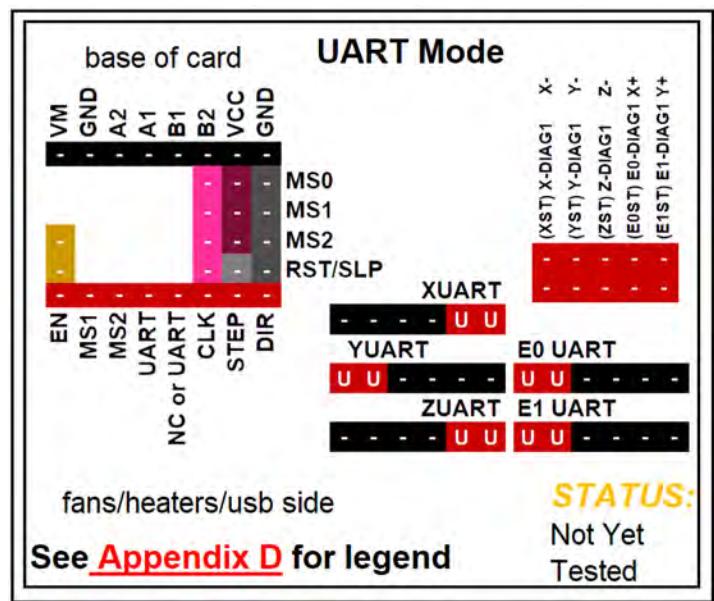
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC2225 V1.0**UART Mode

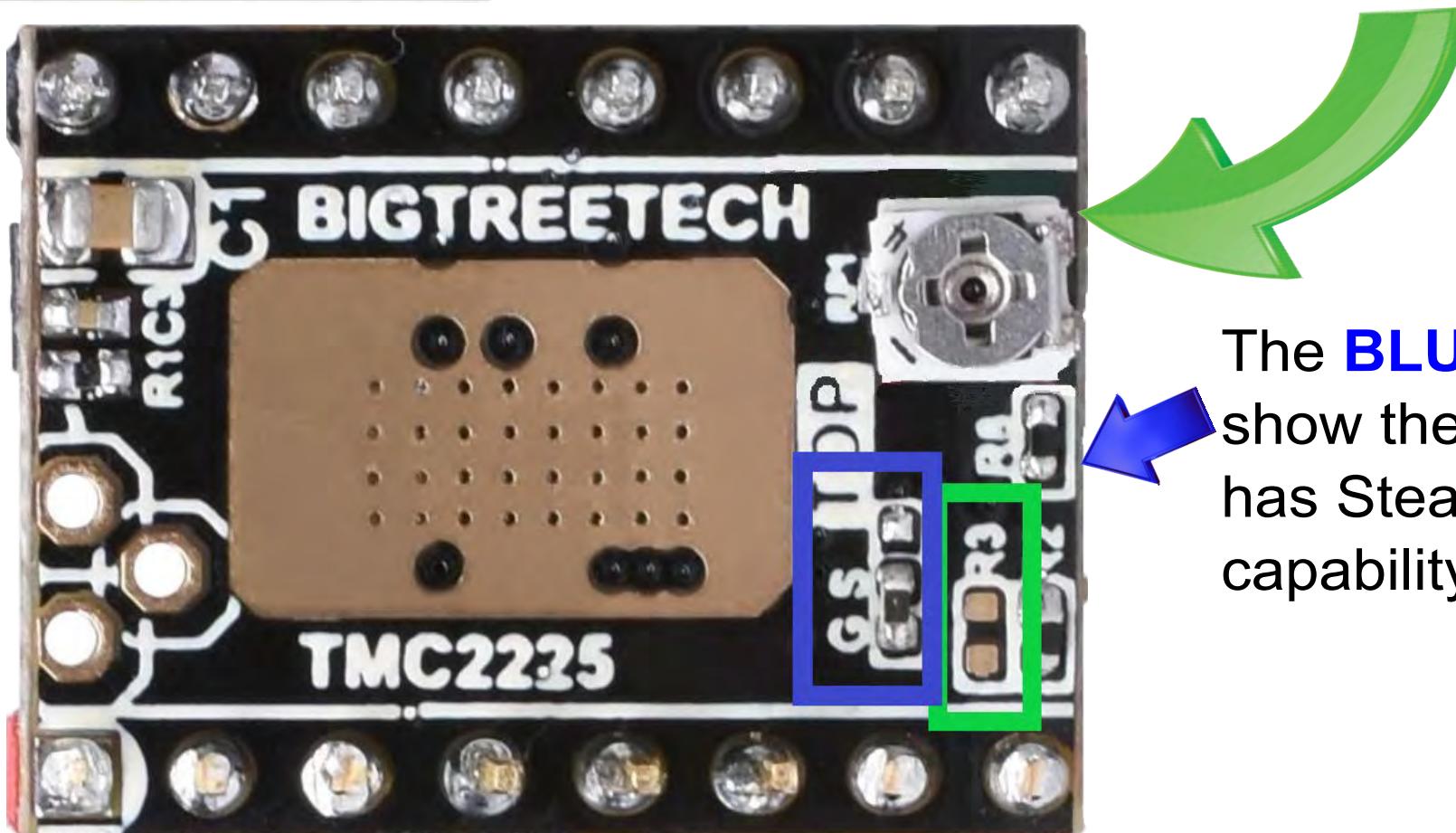
**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

|                                                                                                                                                                                          |                                                                                                                            |                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Driver Chip</b><br><br><b>TMC2225</b><br>UART Mode<br>Maximum 256 Subdivision<br>35V DC<br>2A (peak) | <b>Steps are set inside<br/>of your Firmware</b>                                                                           |                                                                                                                            |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.15Ω                                                                                                     | $I_{MAX} = V_{ref} * 0.7647$<br>See Appendix B #7. Use 50% to 90% as shown below:<br>$I_{MAX} = (V_{ref} * 0.7647) * 0.90$ | $V_{ref} = I_{MAX} * 1.3077$<br>See Appendix B #7. Use 50% to 90% as shown below:<br>$V_{ref} = (I_{MAX} * 1.3077) * 0.90$ |

**BIQU TMC2225 V1.0**UART Mode

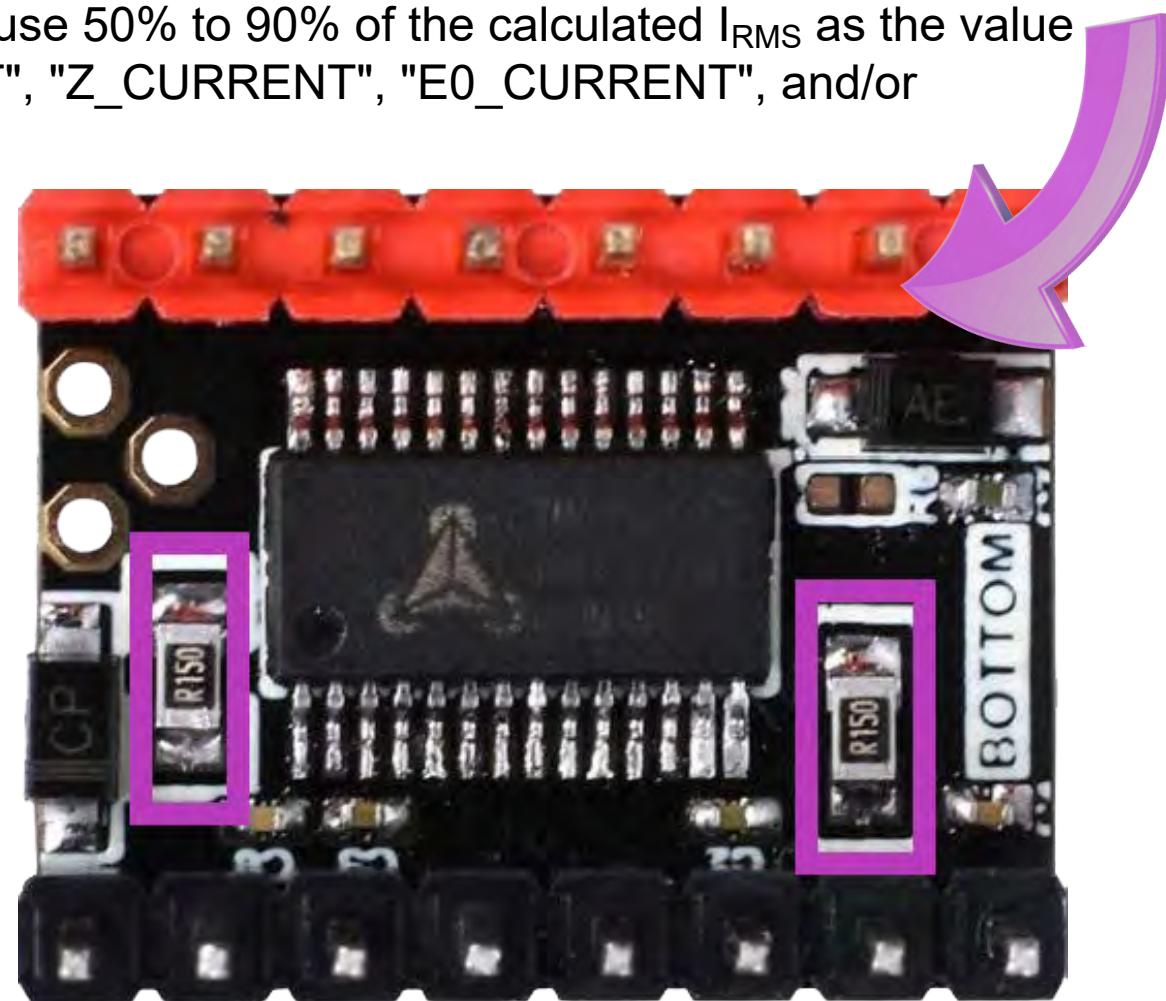
**Important:** To ensure that the BIQU TMC2225 V1.0 is in UART Mode, check to see if the two pads located at R3 have a gap between them, as seen in **GREEN** below. Ensure the "U" jumper is in place on the SKR V1.3 board.



**BIQU TMC2225 V1.0**UART Mode**UART Mode**

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE" so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

- $R_s = R050$  is 0.05 Ohms**
- $R_s = R062$  is 0.062 Ohms**
- $R_s = R068$  is 0.068 Ohms**
- $R_s = R075$  is 0.075 Ohms**
- $R_s = R100$  is 0.1 Ohms**
- $R_s = R110$  is 0.11 Ohms**
- $R_s = R150$  is 0.15 Ohms**
- $R_s = R200$  is 0.2 Ohms**
- $R_s = R220$  is 0.22 Ohms**



**BIQU TMC2225 V1.0**UART Mode

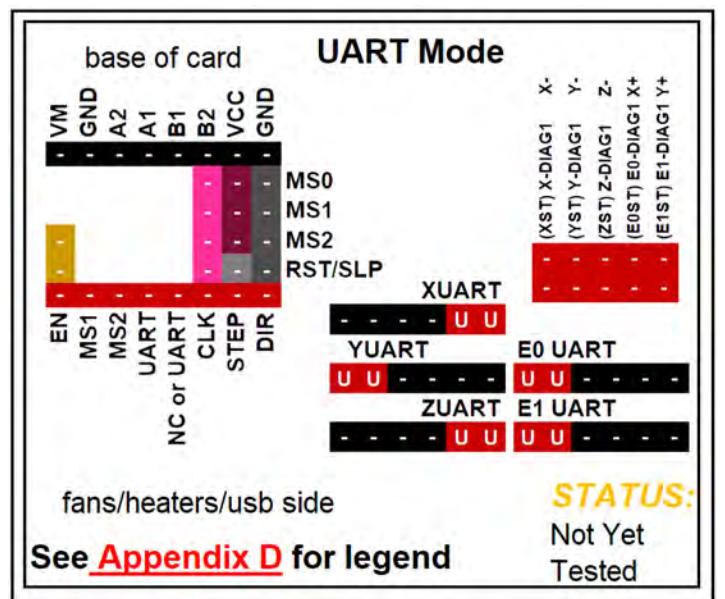
# UART Mode

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.

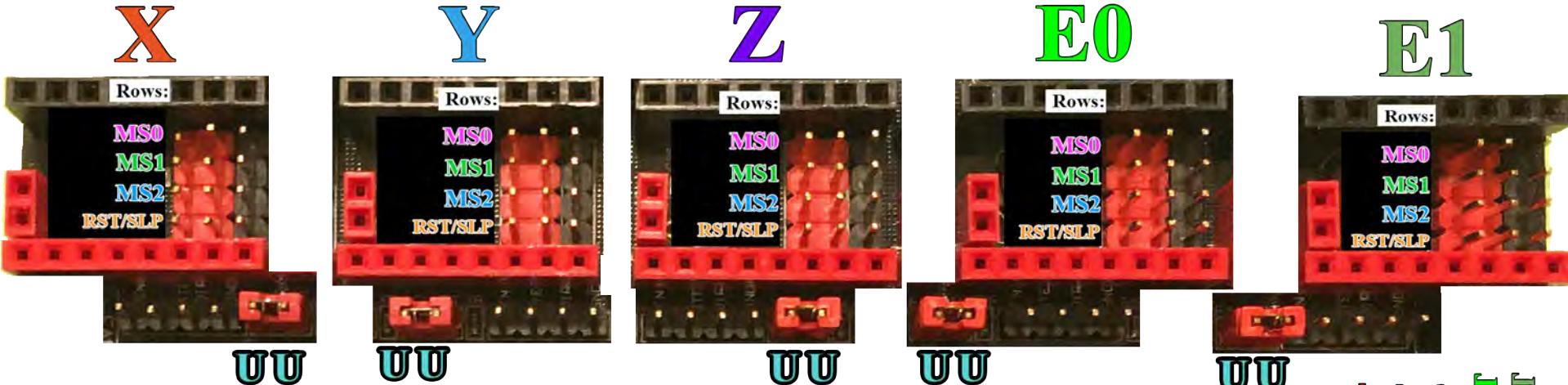


**Note:** Set the "U" Jumper(s) for UART MODE!



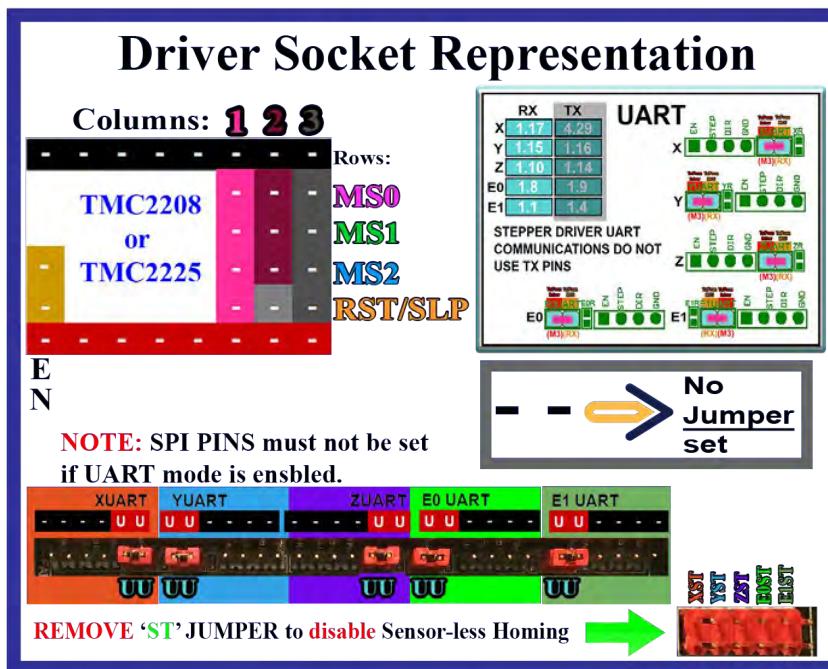


Driver Socket Representation: **UU** → Jumper set



**REMOVE 'ST' JUMPER(s) to disable Sensor-less Homing – for drivers without stallGuard™ feature**



**BIQU TMC2225 V1.0**UART Mode**UART Mode****SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode**

UART Capable Drivers **without** stallGuard™ feature: TMC2208, and TMC2225

Driver Socket Representation:



Meaning:

**Jumper Set**

Driver Socket Representation:

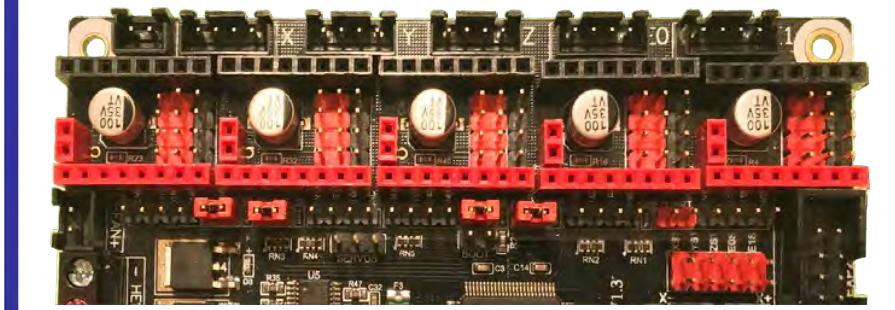
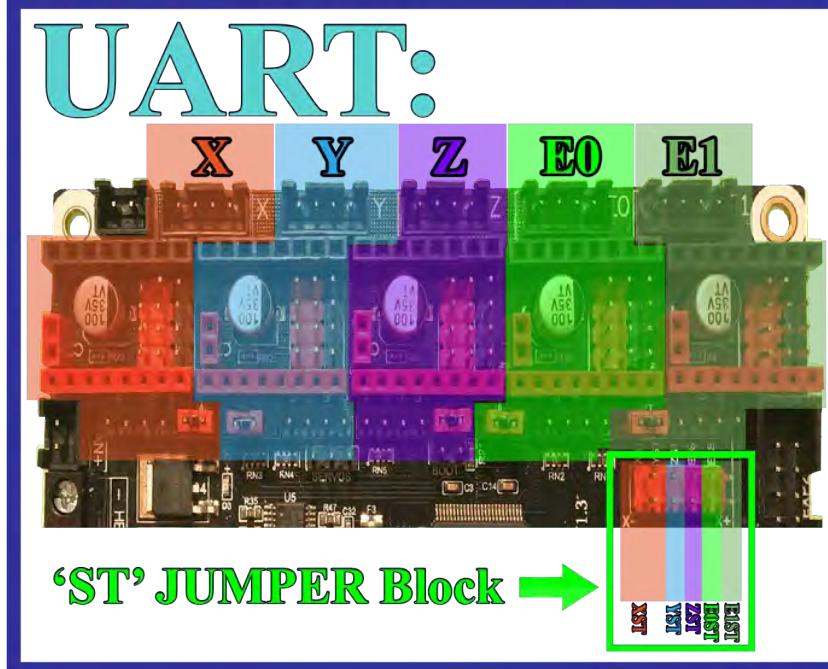


Meaning:

**set Jumper so that both U PINS are covered by the Jumper**

**Each Axis has its own UART PINS (UU).** You can set as many axes as you need. The **UART:** picture shows all axes having UART enabled **AND** all **'ST' JUMPERS REMOVED.** All other PINS need to be empty for UART mode to work properly.

Here is an example of only the X, Y, Z, and E0 axes having UART enabled **AND** **'ST' JUMPERS removed** for ALL Axes:



# UART Mode

BIQU TMC2225 V1.0

UART Mode

## Information on Sensor-less Homing

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing.

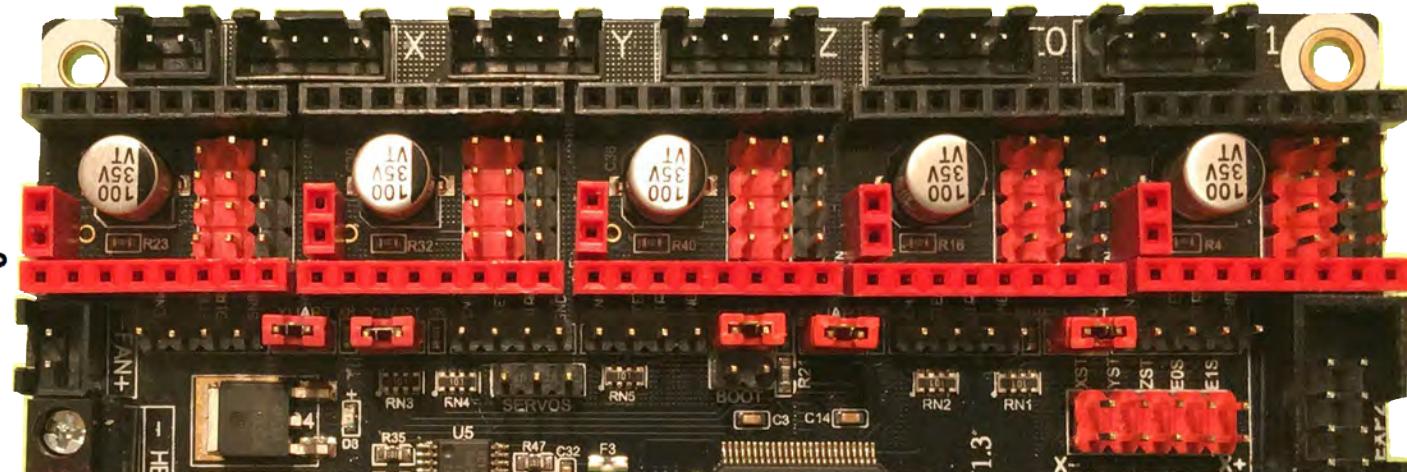
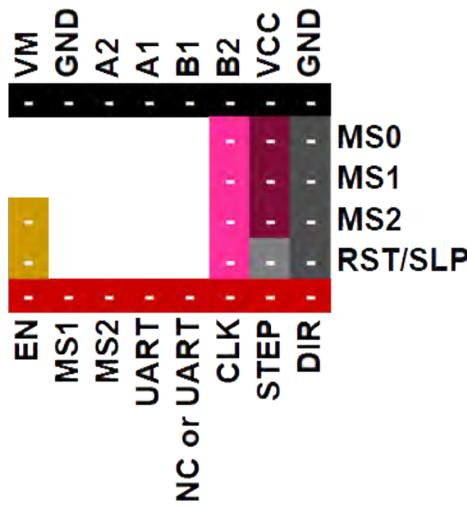
Please read the PREFACE to this manual on “Stall detection and Sensor-less Homing”.

**NOTE:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to ensure proper operation for UART mode.



**BIQU TMC2225 V1.0**UART Mode**UART Mode**

**Note:** Set the "U" Jumper(s) for UART MODE!

**UART**

See [Appendix D](#) for legend

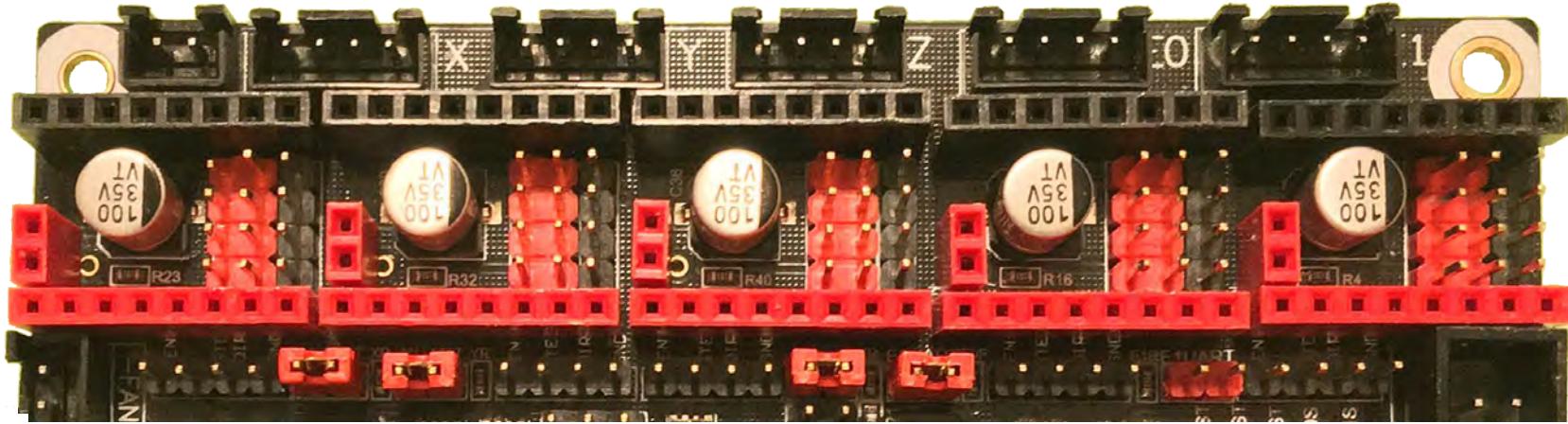


**Note:** TMC2225 does not have sensor-less homing capability.

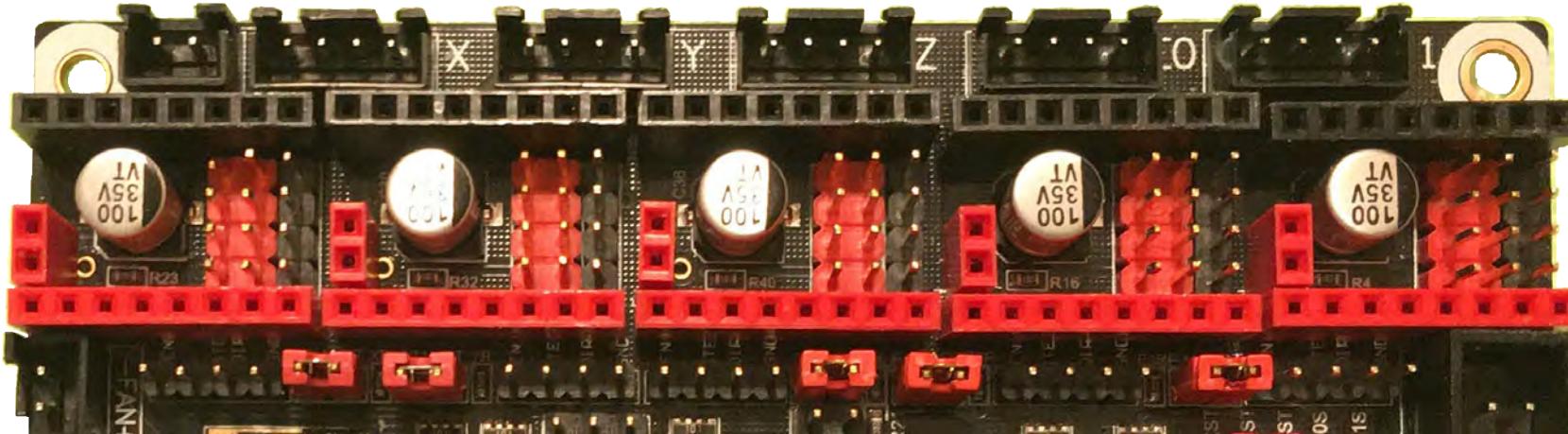
**BIQU TMC2225 V1.0**  
UART Mode

## Examples of Different UART Configurations

X, Y, Z,  
and E0 axes  
configured for  
UART mode.



X, Y, Z, E0  
and E1 axes  
configured for  
UART mode.



## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2225 V1.0 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2225 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2225 drivers in UART mode. Since Marlin does not have an option for TMC2225 drivers we will use "TMC2208" for the DRIVER\_TYPE. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

```

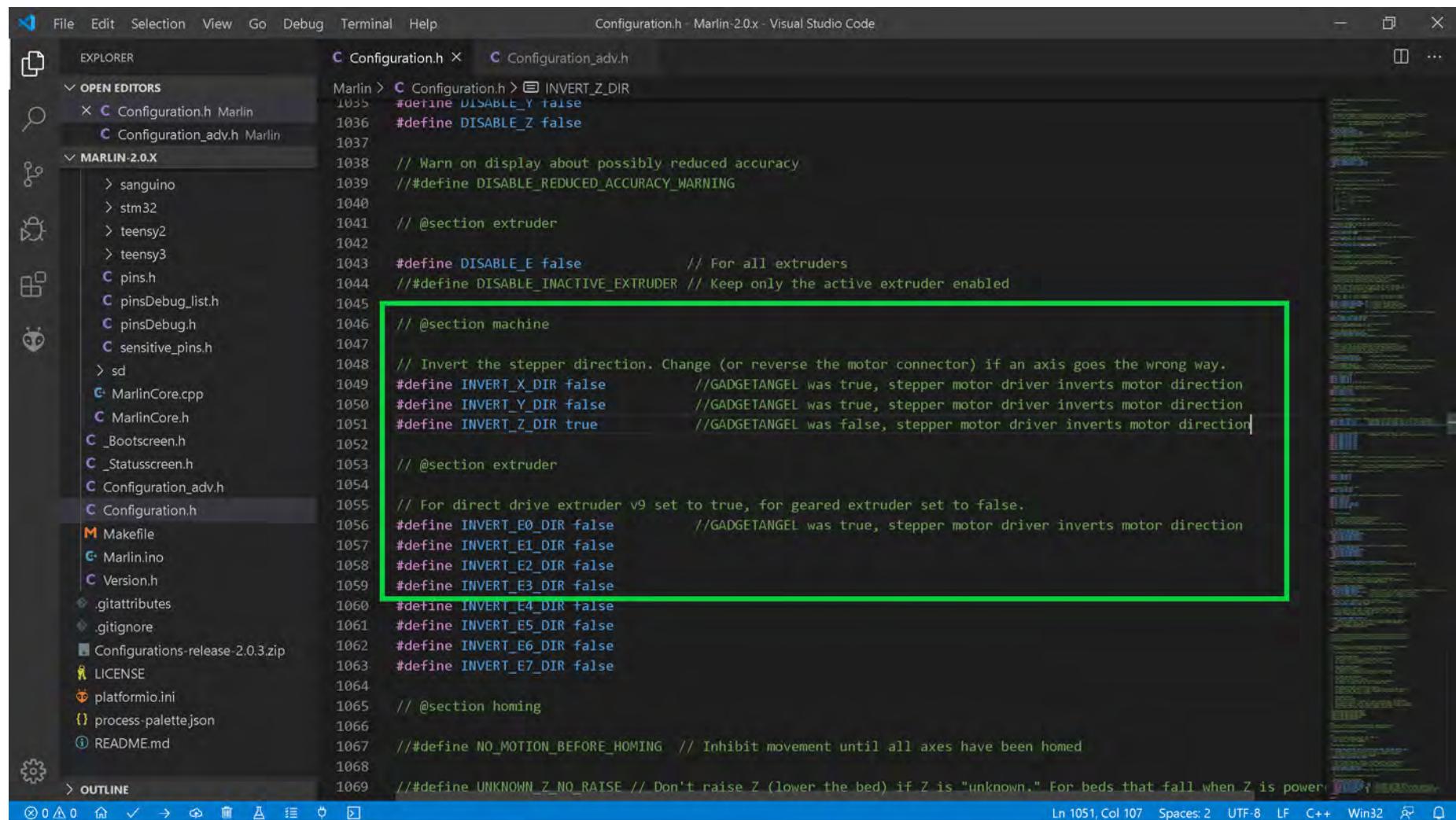
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
 PIO Home
 Configuration.h M...
 Configuration_adv.h
MARLIN-2.0.X
 Lcdprint.cpp
 Lcdprint.h
 thermistornames.h
 ultralcd.cpp
 ultralcd.h
 libs
 module
 pins
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration_adv.h
 Configuration.h
 Makefile
 Marlin.ino
 Version.h
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
 OUTLINE
Ln 686, Col 91 Spaces: 2 UTF-8 LF C++ Win32 ⌂ 1
 661 /**
 662 * Stepper Drivers
 663 *
 664 */
 665 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 666 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 667 *
 668 * A4988 is assumed for unspecified drivers.
 669 *
 670 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 671 * TB6560, TB6600, TMC2100,
 672 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 673 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 674 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 675 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC2660', 'TMC5130', 'TMC5160']
 677 */
#define X_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Y_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
#define Z_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208 //GADGETANGEL was commented out and had the value of A4988
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988
 694
 695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2225 drivers, I must invert the stepper motor direction because the TMC2225 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2225 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
Configuration.h Marlin 1035 #define DISABLE_Y false
Configuration_adv.h Marlin 1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

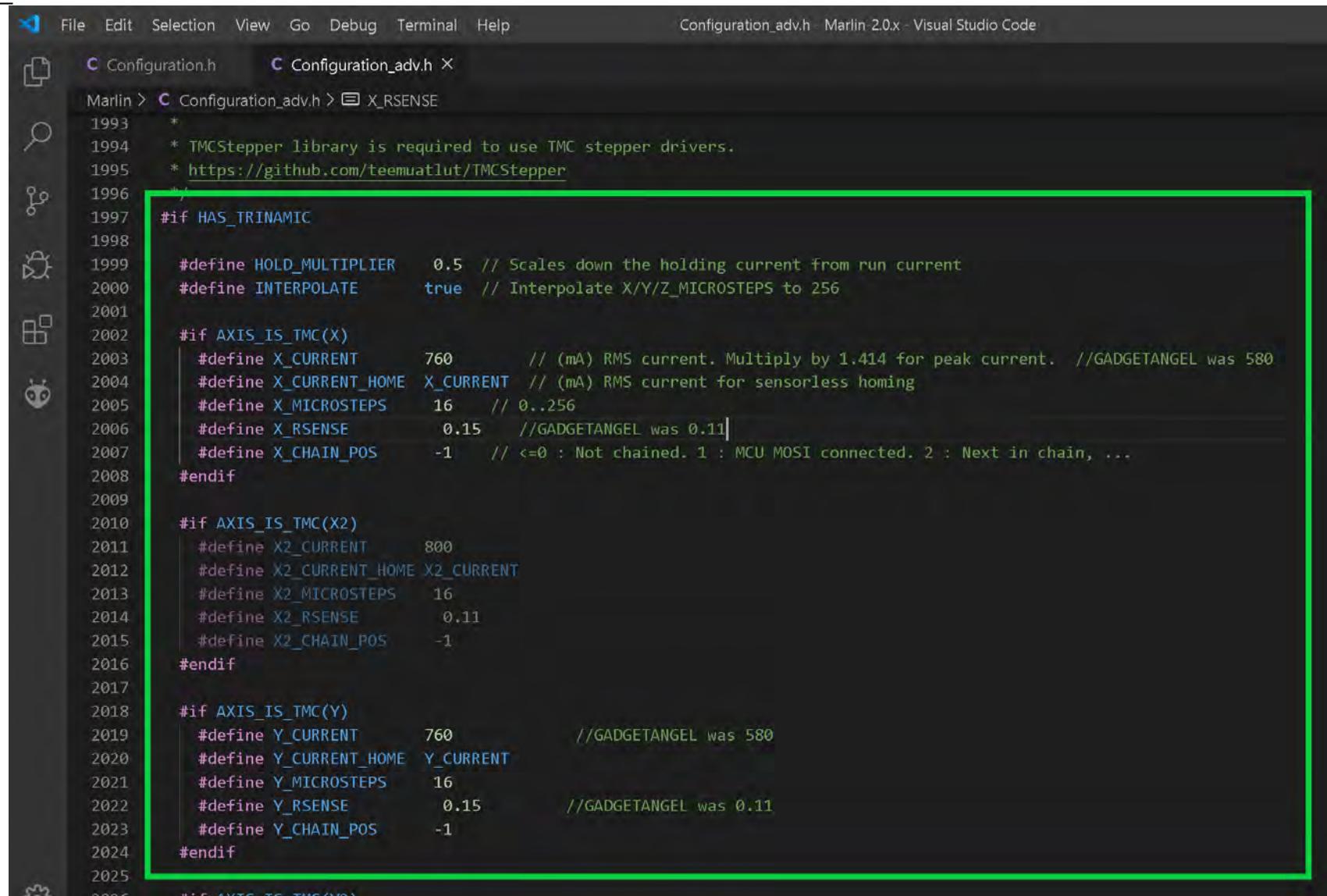
```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC2225 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.15. Ensure "Y\_RSENSE" is set to 0.15.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".



```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin 2.0.x - Visual Studio Code

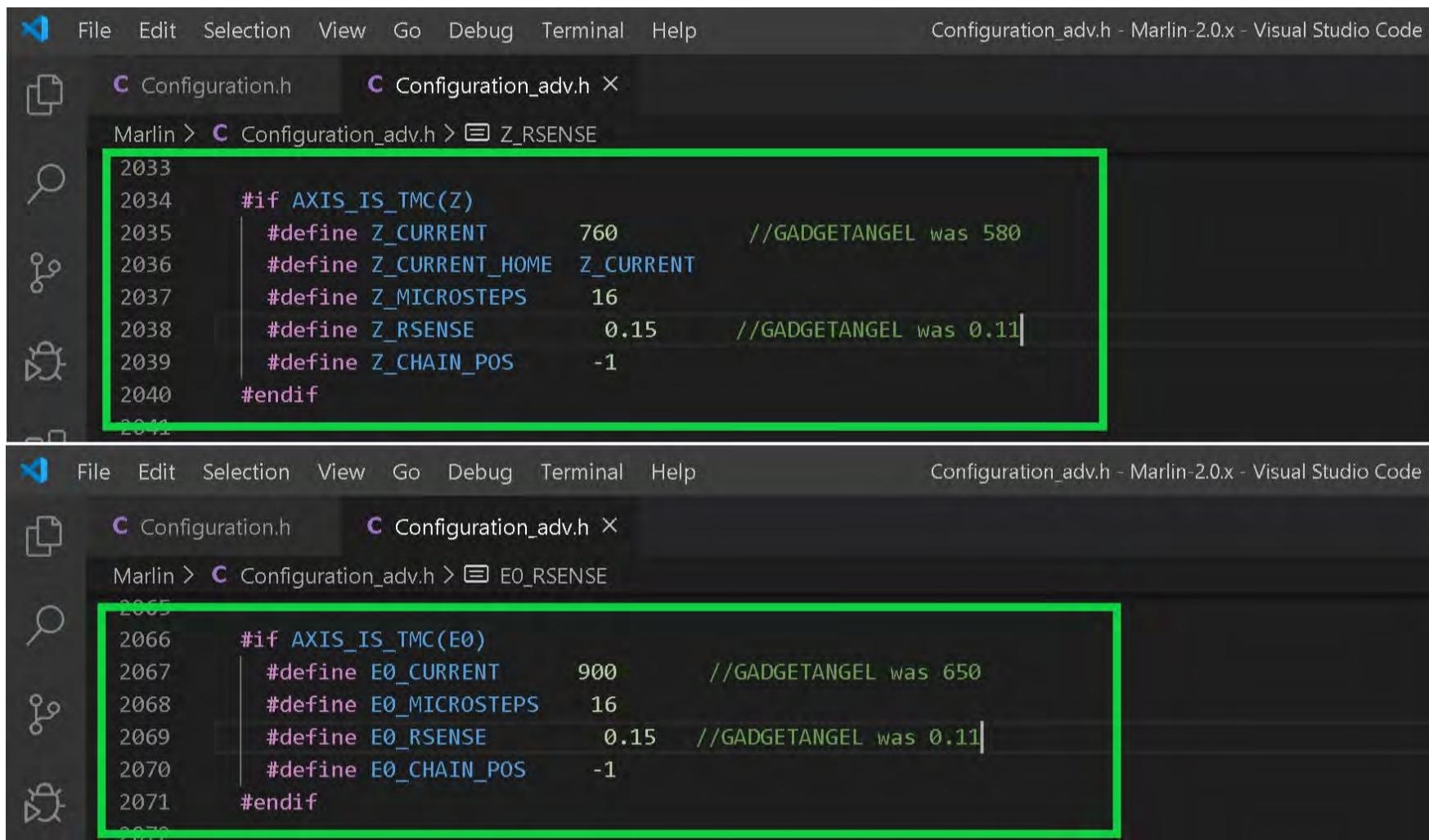
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_RSENSE
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.15 //GADGETANGEL was 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.15 //GADGETANGEL was 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif
2025
2026 //Lc 10/10/2019

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.15. Ensure "E0\_RSENSE" is set to 0.15.
- If you do not want to use  $V_{ref}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



```

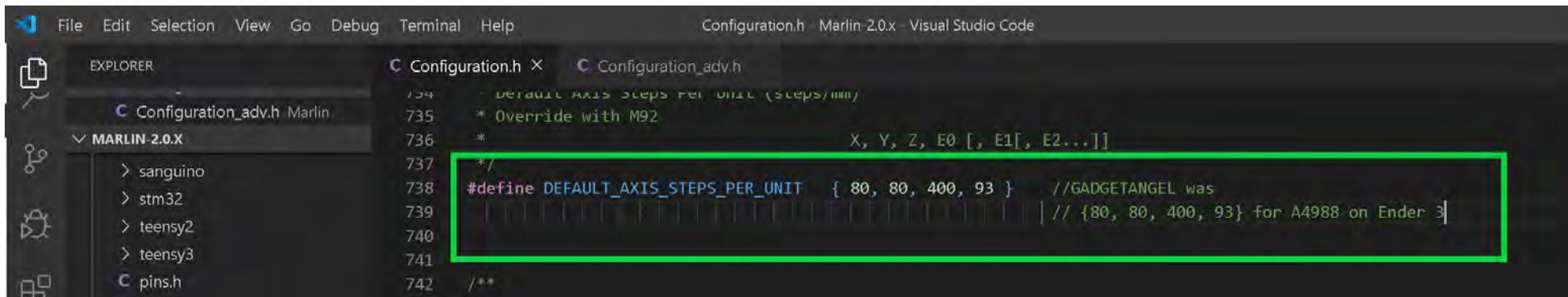
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z_RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035 | #define Z_CURRENT 760 //GADGETANGEL was 580
2036 | #define Z_CURRENT_HOME Z_CURRENT
2037 | #define Z_MICROSTEPS 16
2038 | #define Z_RSENSE 0.15 //GADGETANGEL was 0.11
2039 | #define Z_CHAIN_POS -1
2040 #endif
2041
2042
2043
2044
2045
2046 #if AXIS_IS_TMC(E0)
2047 | #define E0_CURRENT 900 //GADGETANGEL was 650
2048 | #define E0_MICROSTEPS 16
2049 | #define E0_RSENSE 0.15 //GADGETANGEL was 0.11
2050 | #define E0_CHAIN_POS -1
2051 #endif
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes



File Edit Selection View Go Debug Terminal Help Configuration.h Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration\_adv.h

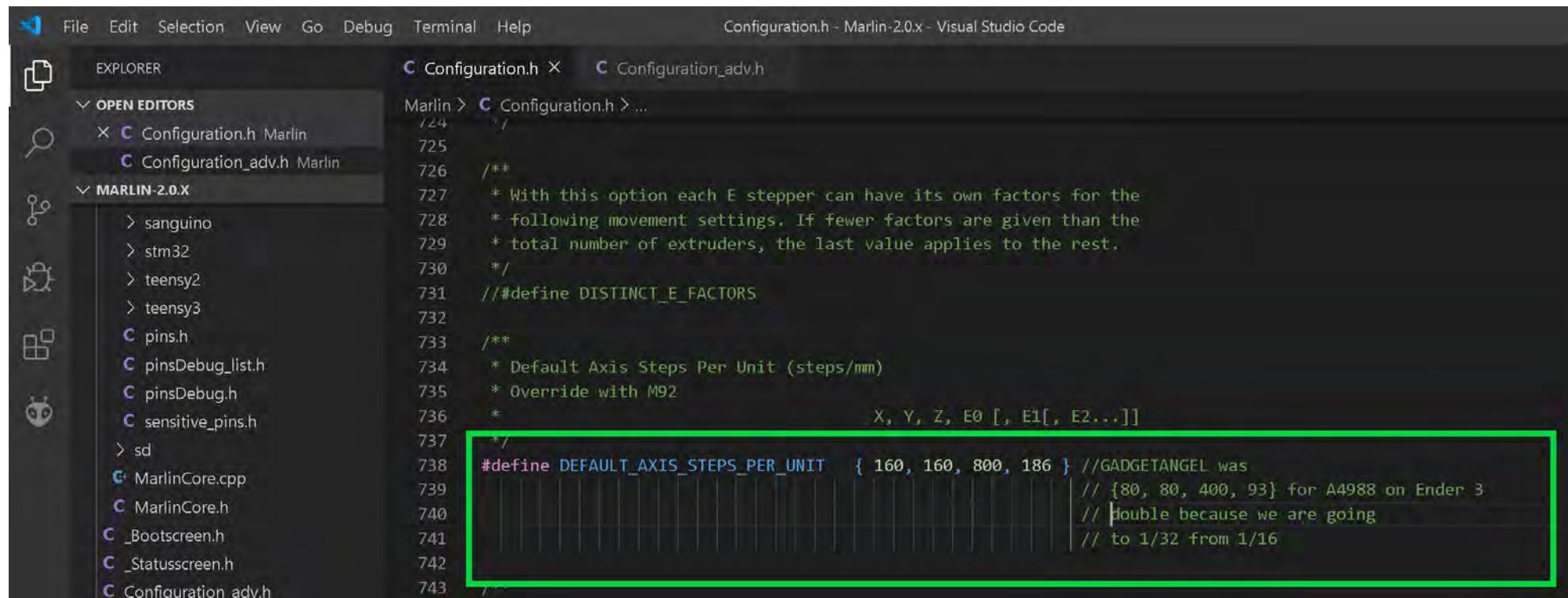
MARLIN-2.0.X

```

 734 * Default Axis Steps Per Unit (steps/mm)
 735 * Override with M92
 736 * X, Y, Z, E0 [, E1[, E2...]]
 737 */
 738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
 739 // {80, 80, 400, 93} for A4988 on Ender 3
 740
 741 /**
 742 */

```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration\_adv.h

OPEN EDITORS Configuration.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X

```

 724 */
 725
 726 /**
 727 * With this option each E stepper can have its own factors for the
 728 * following movement settings. If fewer factors are given than the
 729 * total number of extruders, the last value applies to the rest.
 730 */
 731 //#define DISTINCT_E_FACTORS
 732
 733 /**
 734 * Default Axis Steps Per Unit (steps/mm)
 735 * Override with M92
 736 *
 737 */
 738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
 739 // {80, 80, 400, 93} for A4988 on Ender 3
 740 // Double because we are going
 741 // to 1/32 from 1/16
 742
 743 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I **want stealthChop enabled** so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays the following configuration:

```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
 > sanguino
 > stm32
 > teensy2
 > teensy3
 < pins.h
 < pinsDebug_list.h
 < pinsDebug.h
 < sensitive_pins.h
 > sd
 < MarlinCore.cpp
 < MarlinCore.h
 < Bootscreen.h
 < Statusscreen.h
 Configuration_adv.h
 Configuration.h

Marlin > Configuration_adv.h > STEALTHCHOP_XY
2193 /**
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213 /**

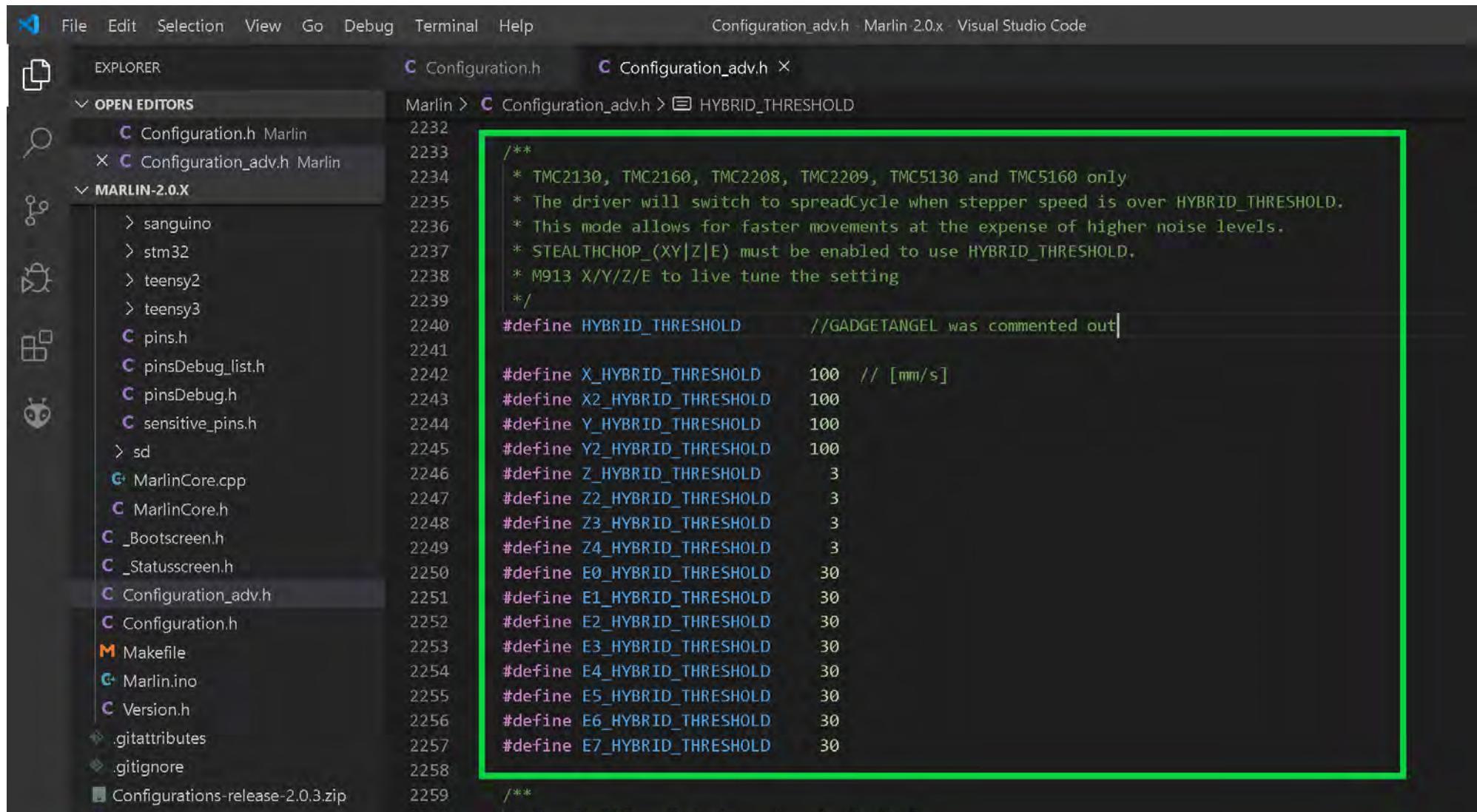
```

The code editor has two sections highlighted with green boxes. The top section highlights the definitions for `STEALTHCHOP_XY`, `STEALTHCHOP_Z`, and `STEALTHCHOP_E`. The bottom section highlights the comments and the definition for `CHOPPER_TIMING`.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default, it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):**
  - OPEN EDITORS: Configuration.h Marlin, Configuration\_adv.h Marlin
  - MARLIN-2.0.X:
    - > sanguino
    - > stm32
    - > teensy2
    - > teensy3
    - C pins.h
    - C pinsDebug\_list.h
    - C pinsDebug.h
    - C sensitive\_pins.h
    - > sd
    - G MarlinCore.cpp
    - C MarlinCore.h
    - C \_Bootscreen.h
    - C \_Statusscreen.h
    - C Configuration\_adv.h
    - C Configuration.h
    - M Makefile
    - C Marlin.ino
    - C Version.h
  - .gitattributes
  - .gitignore
  - Configurations-release-2.0.3.zip
- Right Editor Area:** Shows the content of Configuration\_adv.h. A specific section is highlighted with a green border:
 

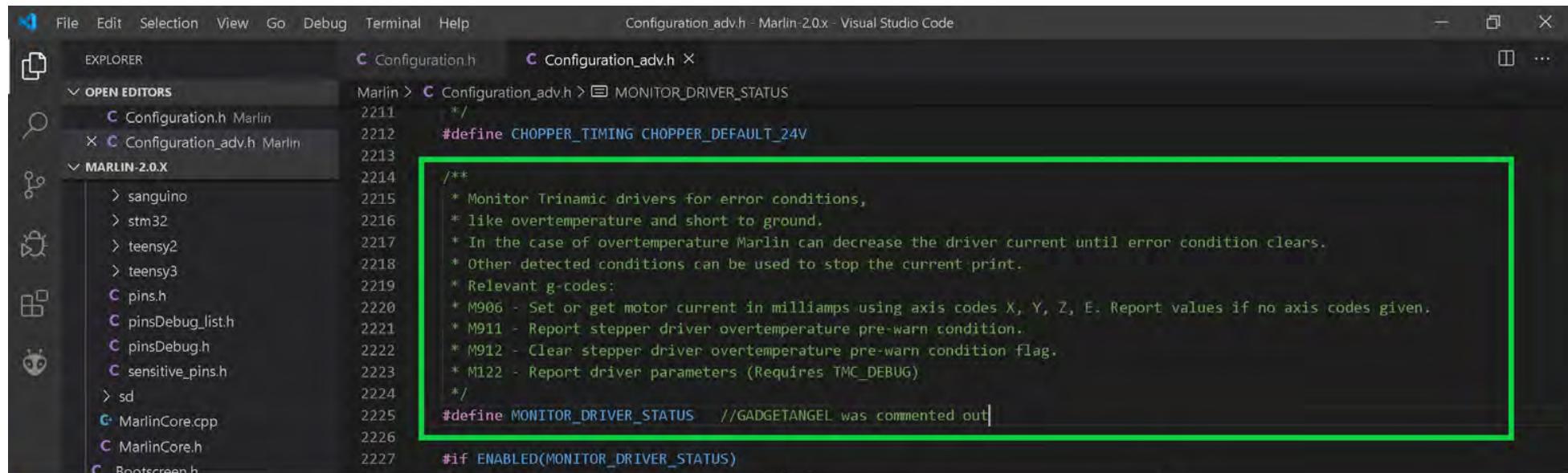
```
/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //GADGETANGEL was commented out

#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30
```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line.](#)



File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0x - Visual Studio Code

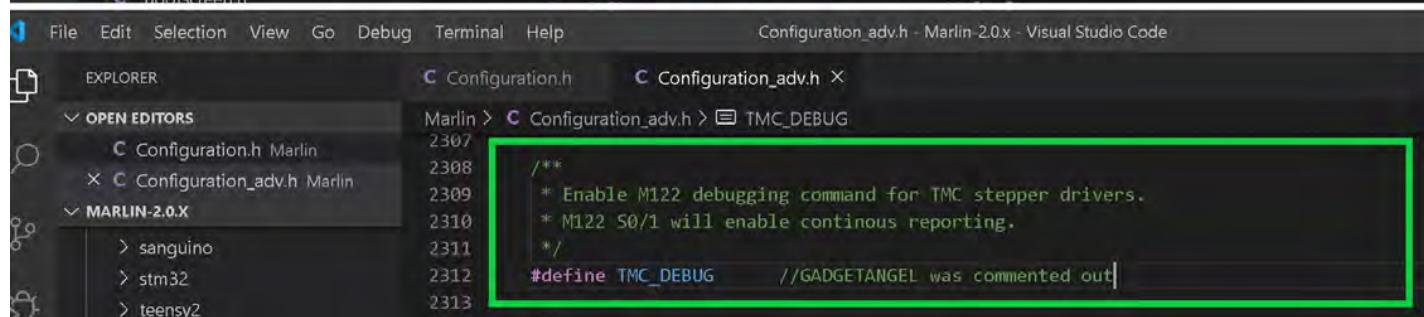
EXPLORER Configuration.h Configuration\_adv.h

Marlin > Configuration.h > MONITOR\_DRIVER\_STATUS

```

2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration\_adv.h

Marlin > Configuration\_adv.h > TMC\_DEBUG

```

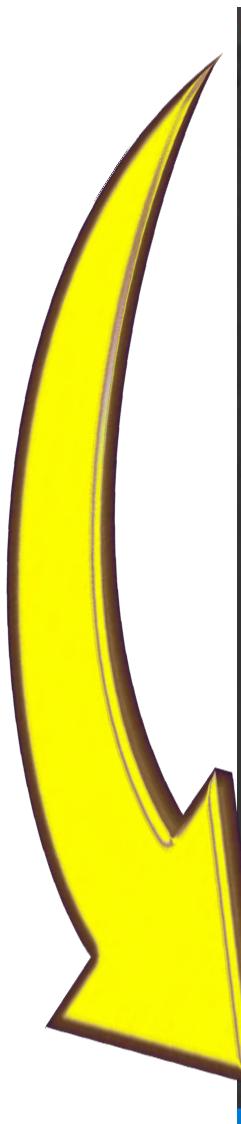
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- The end of Marlin setup for BIQU TMC2225 V1.0 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



**Configuration.h - Marlin-2.0.x - Visual Studio Code**

File Edit Selection View Go Run Terminal Help

EXPLORER Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src... pins\_BTT\_SKR\_common.h Marlin\src... Configuration\_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the name below to that matching your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[=====] [SUCCESS] Took 130.61 seconds =====

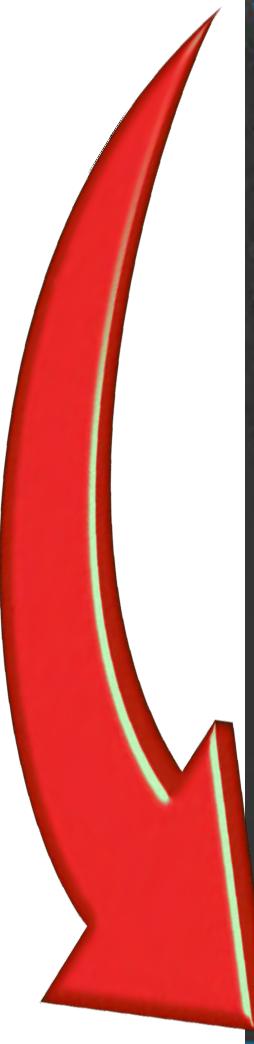
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| ramps           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino&44p    | IGNORED |              |
| sanguino&1284p  | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUE_dfu         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

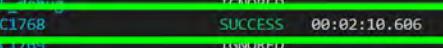
- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2225 V1.0 Drivers in UART Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.







File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

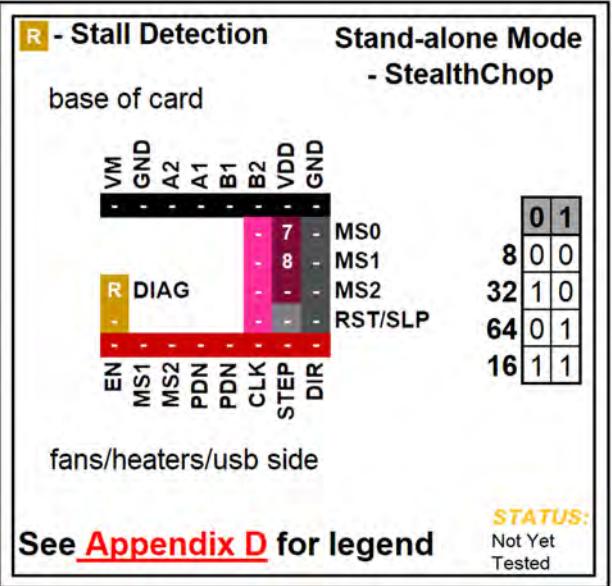
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| SUE_L103        | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

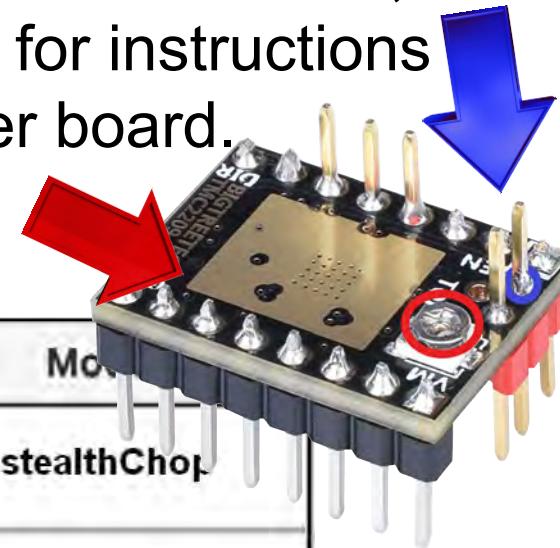


## BIQU TMC2209 V1.2

### Stand-alone Mode for StealthChop

**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See [Appendix A](#) for instructions on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.

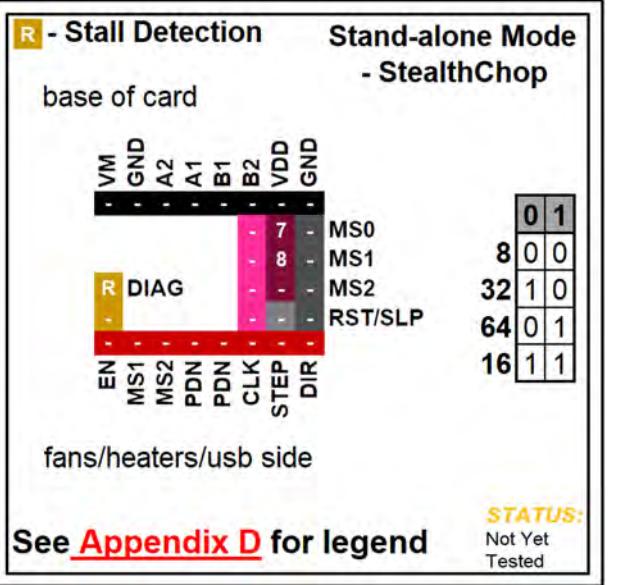


| Driver Chip                                                                                                       | MS0  | MS1  | Steps  | Interpolation | Mode        |
|-------------------------------------------------------------------------------------------------------------------|------|------|--------|---------------|-------------|
| <b>BIQU®<br/>TMC2209</b><br><small>Stand Alone Mode<br/>Maximum 64 Subdivision<br/>28V DC<br/>2.8A (peak)</small> | Low  | Low  | 1 / 8  | 1 / 256       | stealthChop |
|                                                                                                                   | High | Low  | 1 / 32 | 1 / 256       | stealthChop |
|                                                                                                                   | Low  | High | 1 / 64 | 1 / 256       | stealthChop |
|                                                                                                                   | High | High | 1 / 16 | 1 / 256       | stealthChop |

| Driving Current Calculation Formula    | $I_{MAX} = V_{ref}$<br>See Appendix B #5. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90$ | $V_{ref} = I_{MAX}$<br>See Appendix B #5. Use 50% to 90% as shown below:<br>$V_{ref} = V_{ref} * 0.90$ |
|----------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| $R_S$ (Typical Sense Resistor) = 0.11Ω |                                                                                                        |                                                                                                        |

- See next page for the legend that belongs to the above chart.

**Driver Chip Chart:**

| Driver Chip                                        | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|----------------------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b>                                        | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
| Chip Name                                          | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision                            | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
| XXV DC xxA (peak)                                  | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                                    | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                                    | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                                    | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                                    | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula                | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |
| $R_S(\text{typical Sense Resistor}) = x.xx \Omega$ |                            |      |                            |                      |                 |

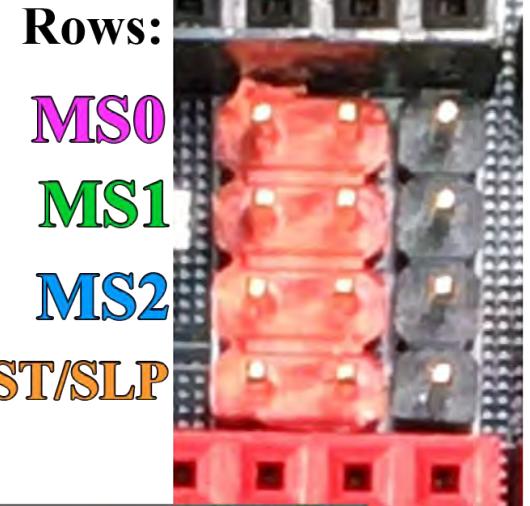
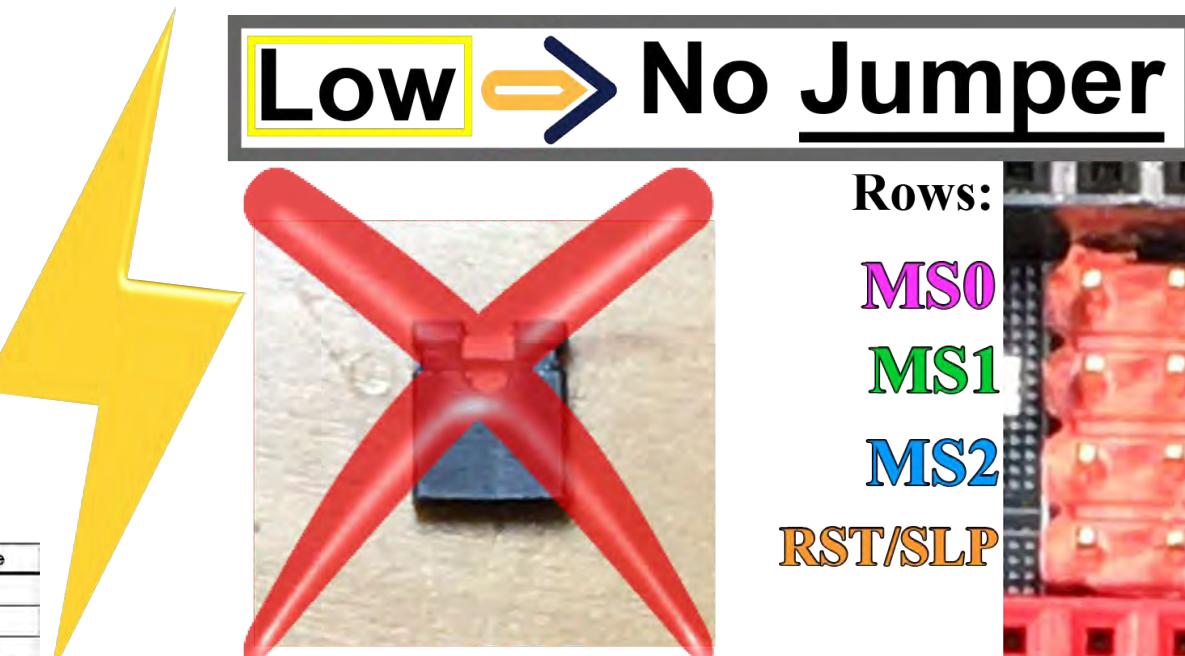
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):

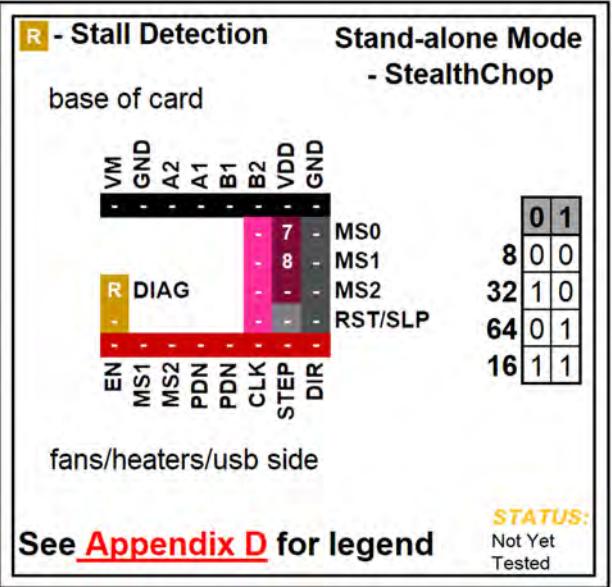
Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

# BIQU TMC2209 V1.2

Stand-alone Mode for StealthChop

## SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers





## BIQU TMC2209 V1.2

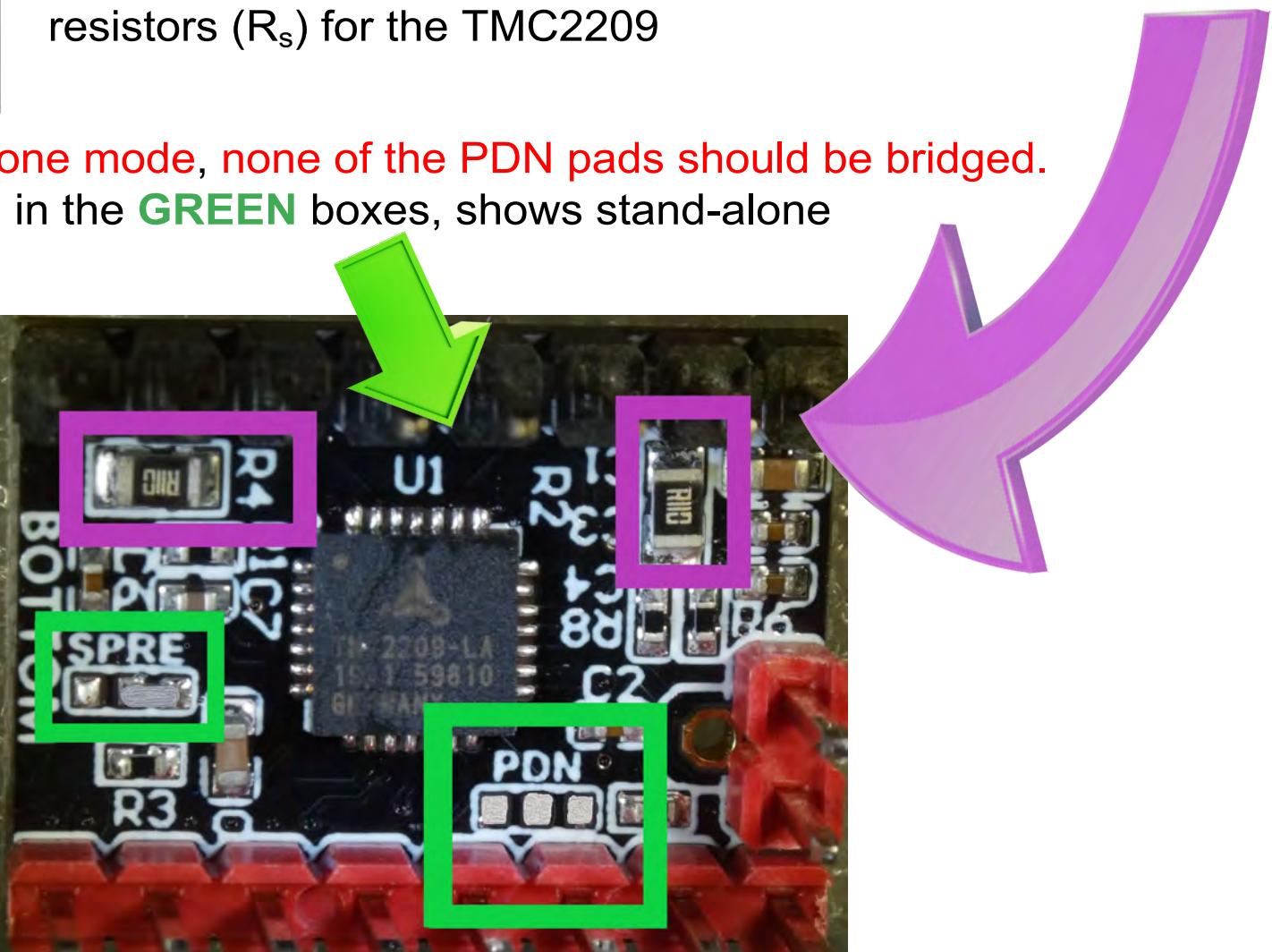
### Stand-alone Mode for StealthChop

**NOTE:** The SPRE jumper is located on the bottom of the driver board. In stand-alone mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board. The **PURPLE boxes** below show the location of the current sense resistors ( $R_s$ ) for the TMC2209

Note: To switch to stand-alone mode, none of the PDN pads should be bridged.

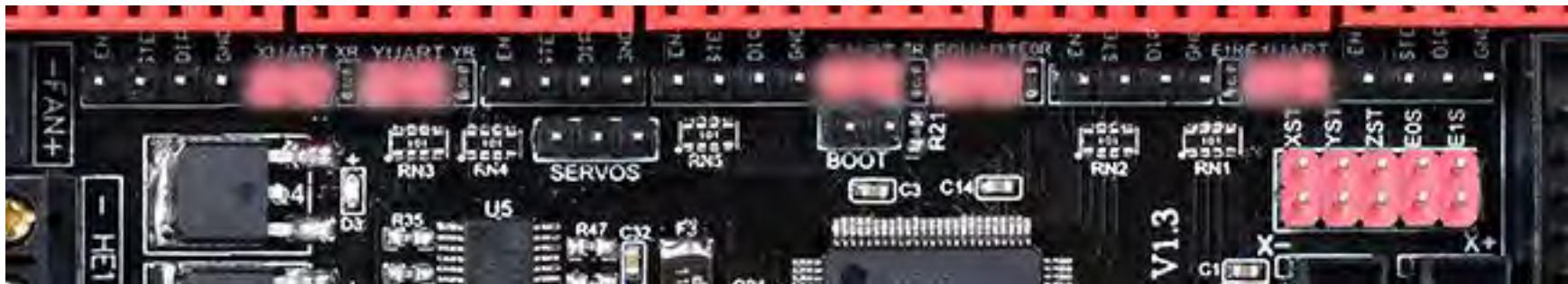
The picture below, as seen in the **GREEN** boxes, shows stand-alone mode with StealthChop.

MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.



**BIQU TMC2209 V1.2****Stand-alone Mode for StealthChop****Stand-alone StealthChop Mode**

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode with StealthChop enabled.



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode with StealthChop enabled.

XUART

YUART

ZUART

E0 UART

E1 UART



**BIQU TMC2209 V1.2****Stand-alone Mode for StealthChop****Stand-alone StealthChop Mode****SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers****Driver Socket Representation**Columns: **1 2 3**Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

DIAG

E  
N**1/2****Driver Chip Chart:**

| Driver Chip                                    | MS0                        | MS1  | MS2                        | Excitation Mode |
|------------------------------------------------|----------------------------|------|----------------------------|-----------------|
| LOGO                                           | Low                        | Low  | Low                        | 1-1 Phase       |
| Chip Name                                      | High                       | Low  | Low                        | W1-1 Phase      |
| Maximum XXX                                    | Low                        | High | Low                        | 2WL-2 Phase     |
| Subdivision                                    | Low                        | Low  | High                       | 4WL-2 Phase     |
| XXV DC                                         | High                       | Low  | High                       | SW1-2 Phase     |
| XXA (peak)                                     | Low                        | High | High                       | 16WL-2 Phase    |
| Driving Current Calculation Formula            | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                 |
| $R_g$ (typical) Series Resistor (in $\Omega$ ) |                            |      |                            |                 |

**MS0 for Binary State Drivers:**

Driver Socket Representation:

**7 7**

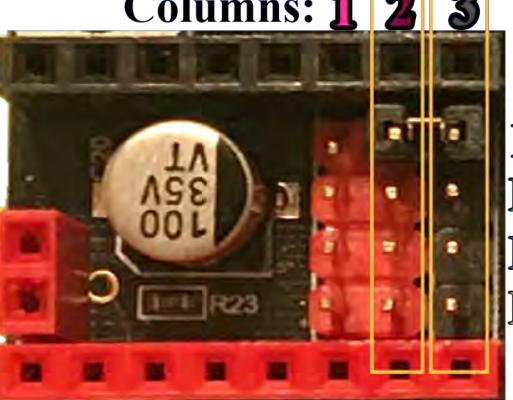
Driver Chip Chart:

**High**

Driver Socket Representation:

**--**

Driver Chip Chart:

**Low****High:**Columns: **1 2 3**Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP****Meaning:**

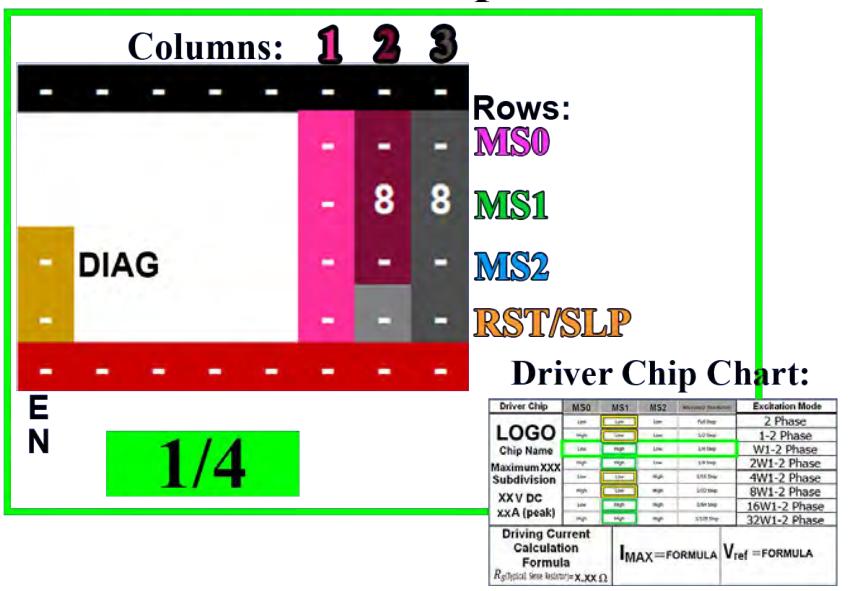
Driver Chip Chart:

**High****set Jumper between column 2 and column 3 on the MS0 row****Low****No Jumper Set**

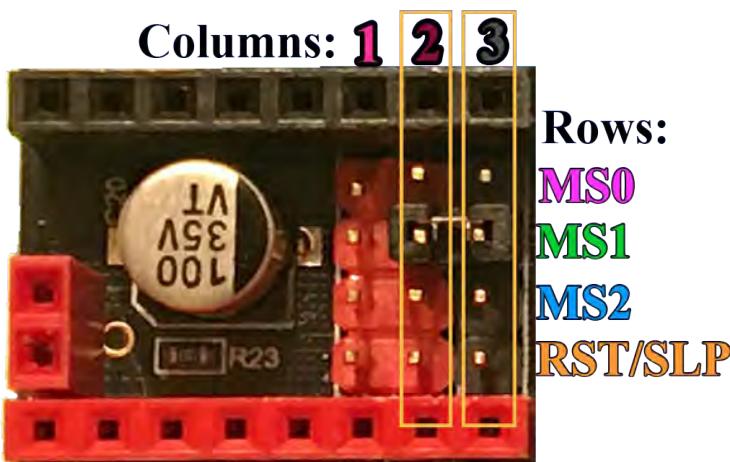
# Stand-alone StealthChop Mode

## Stand-alone Mode for StealthChop

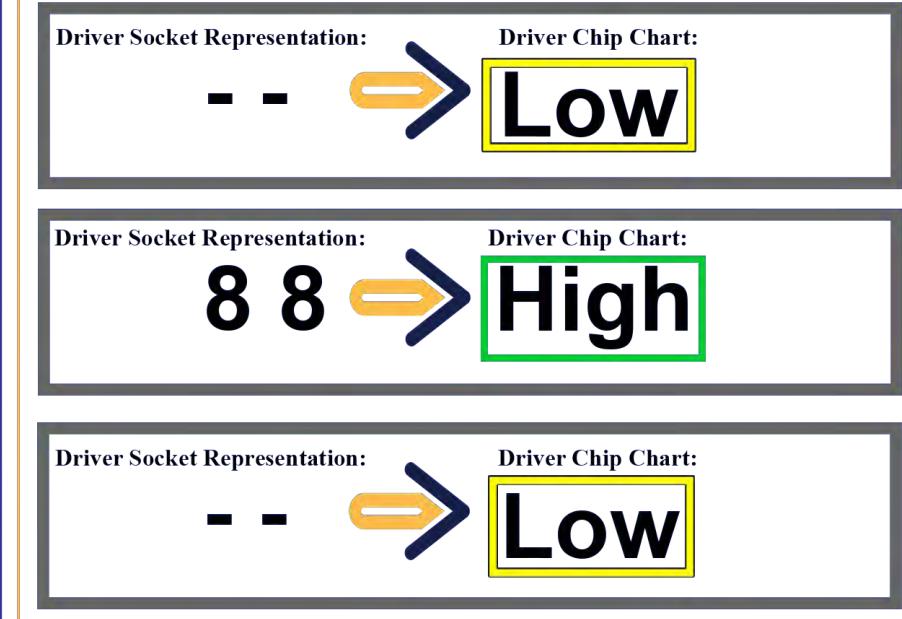
### Driver Socket Representation



### High:



### MS1 for Binary State Drivers:

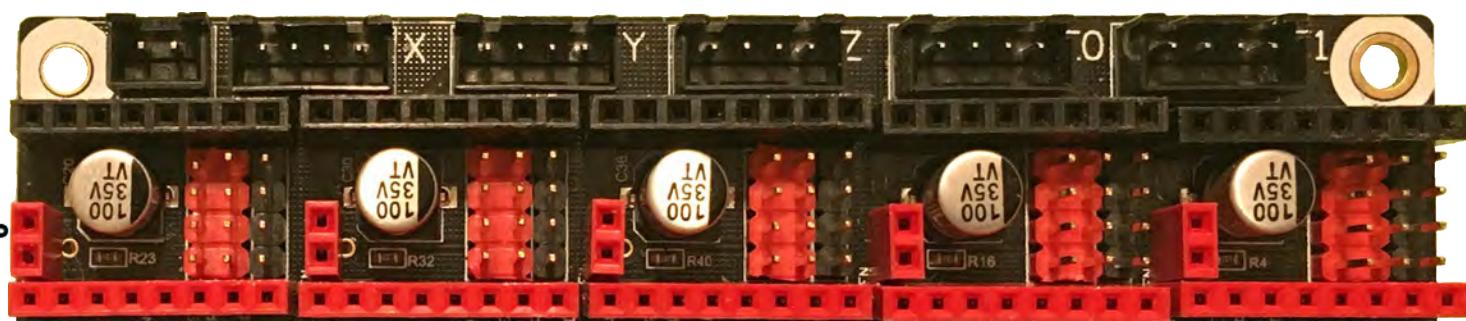
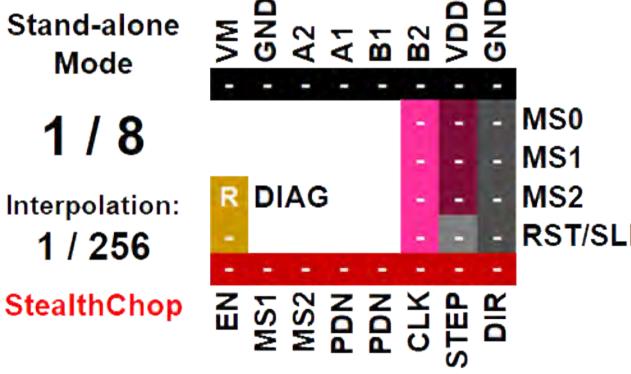
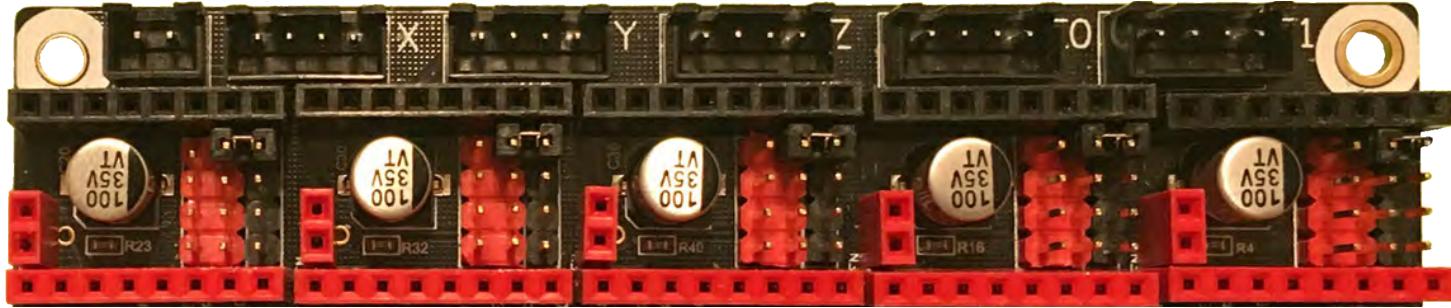
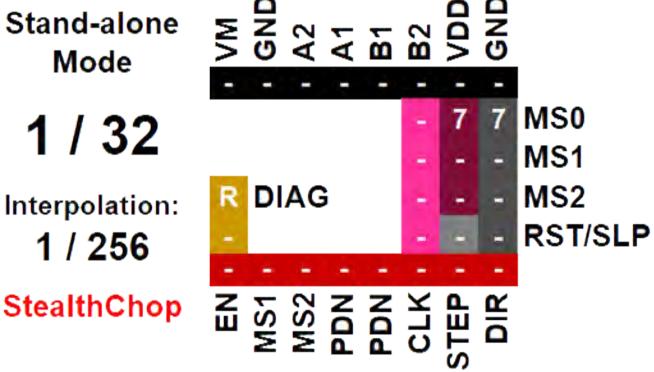


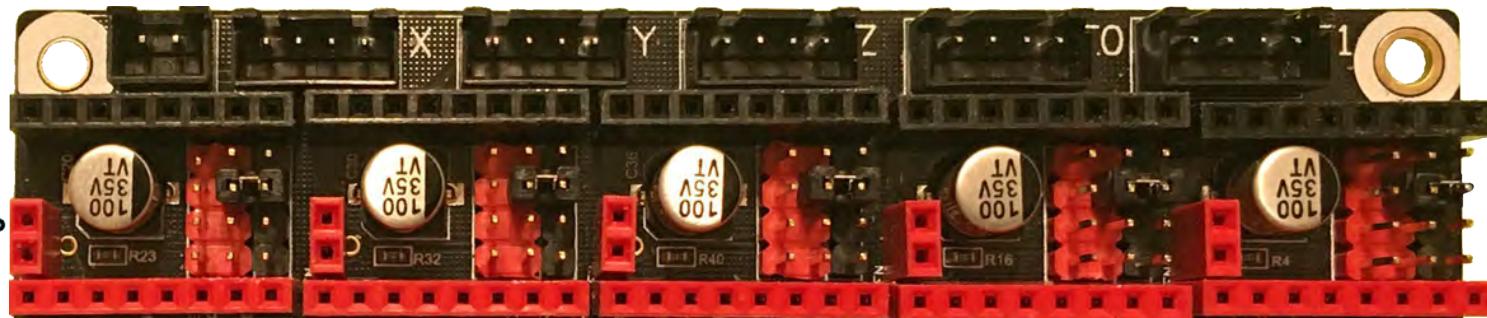
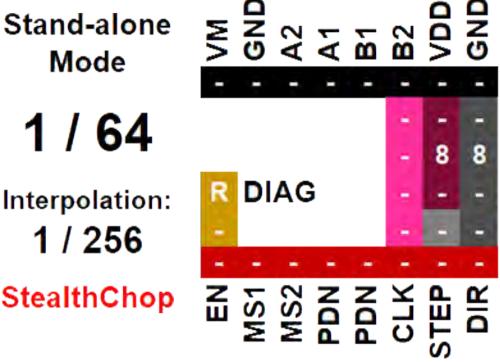
Meaning:

Driver Chip Chart:

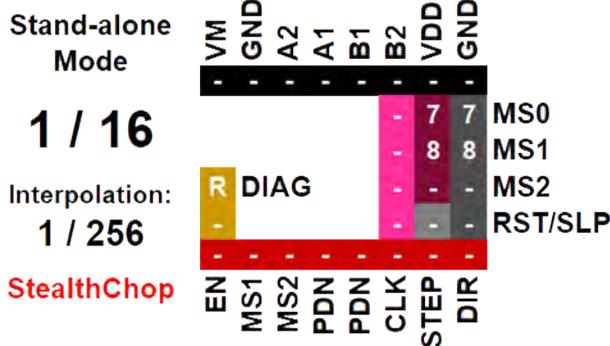
**High** → set Jumper between column 2 and column 3 on the MS1 row

**Low** → No Jumper Set

**BIQU TMC2209 V1.2****Stand-alone Mode for StealthChop****Stand-alone StealthChop Mode**See [Appendix D](#) for legendSee [Appendix D](#) for legend

**BIQU TMC2209 V1.2****Stand-alone Mode for StealthChop****Stand-alone StealthChop Mode**

See [Appendix D](#) for legend

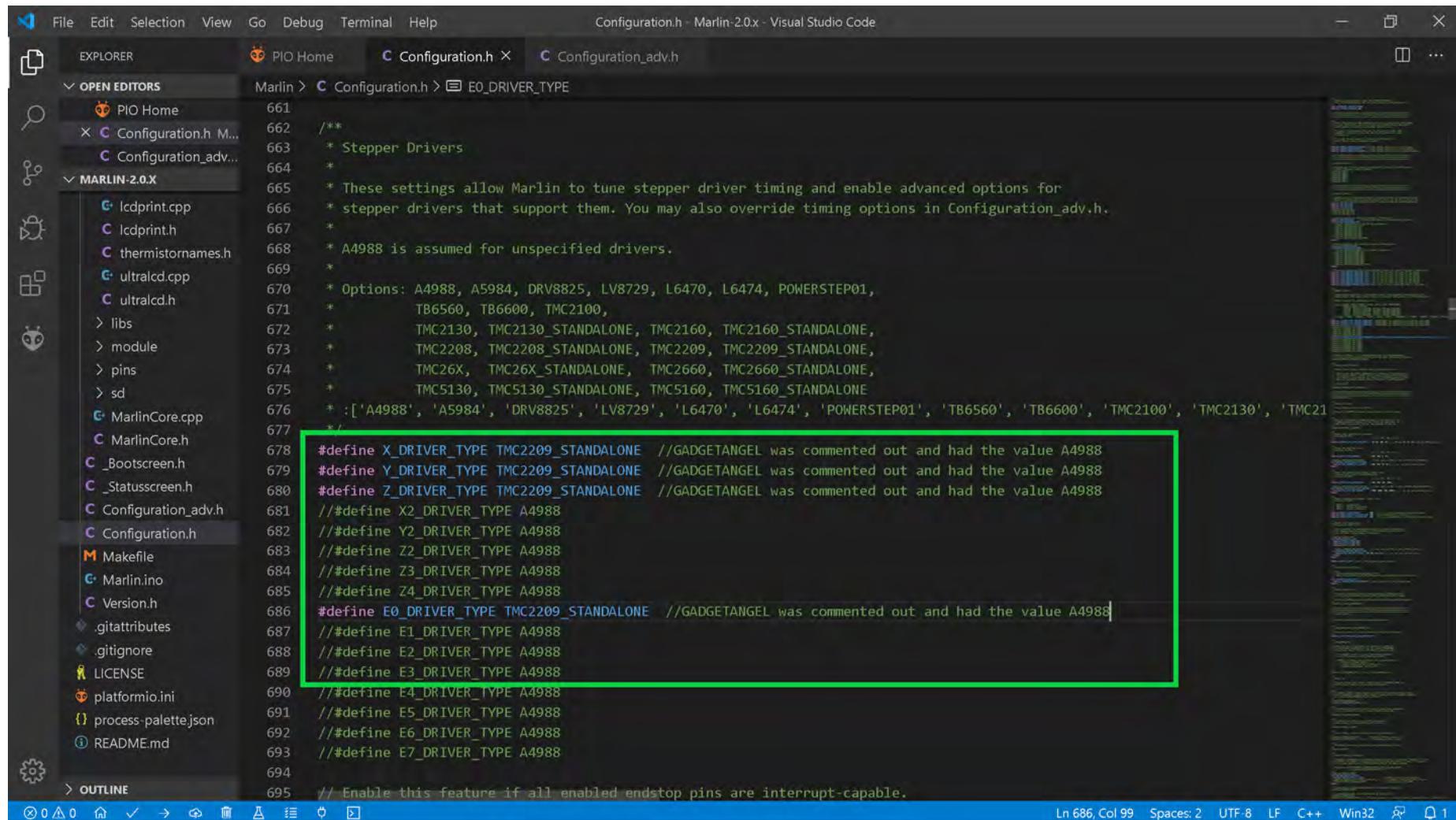


See [Appendix D](#) for legend

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2209 V1.2 stepper motor drivers in stand-alone mode for stealthChop.

- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in stand-alone mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in stand-alone mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following configuration for stepper drivers:

```

661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 //#define E0_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693 //#define E8_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

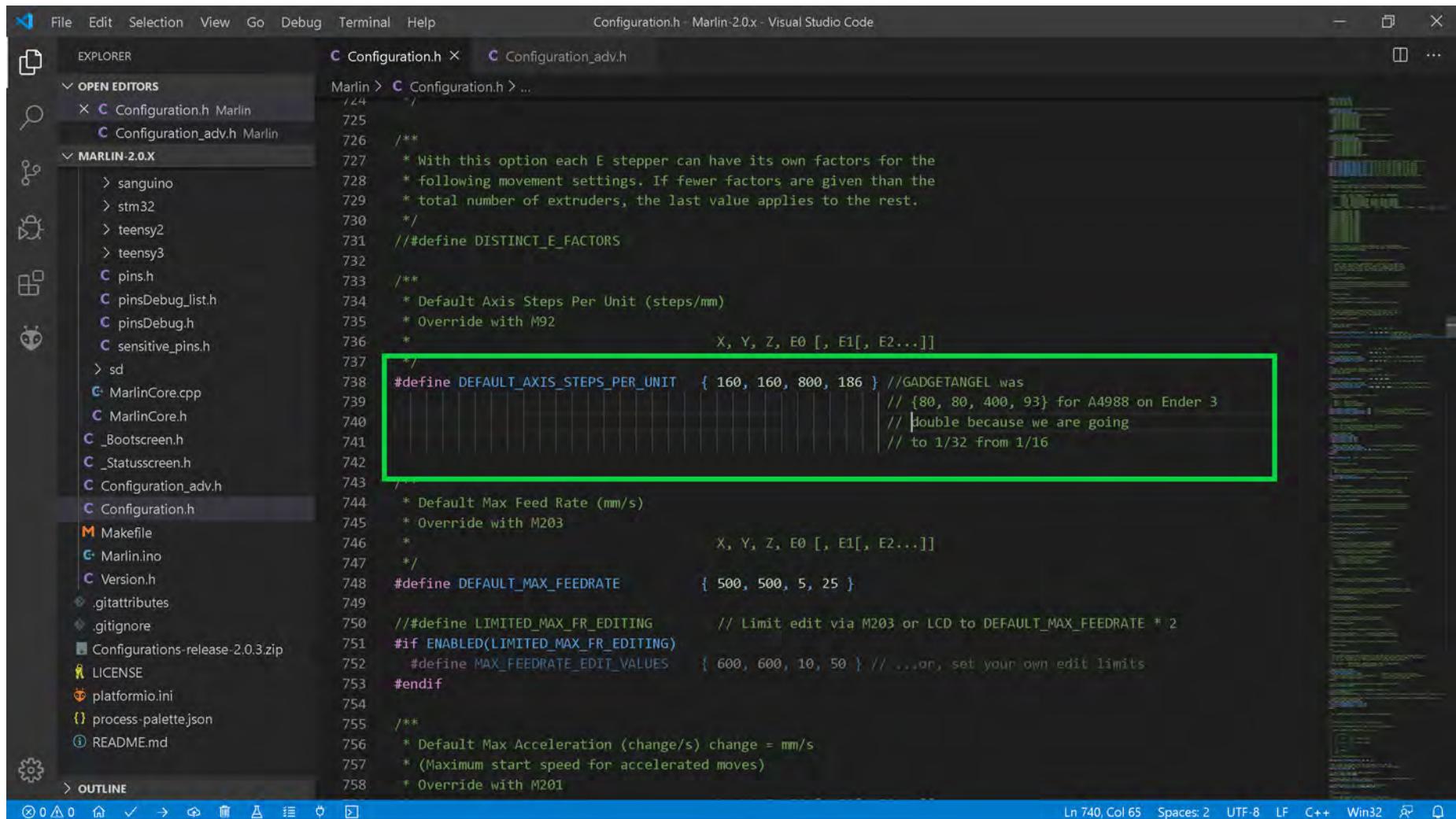
```

A green rectangular box highlights the driver type definitions for X, Y, Z, and E0 axes, specifically the lines starting with '#define' followed by the driver type and a comment about GADGETANGEL's previous configuration.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following C++ code snippet:

```

 ...
 * With this option each E stepper can have its own factors for the
 * following movement settings. If fewer factors are given than the
 * total number of extruders, the last value applies to the rest.
 */
//#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [, E1[, E2...]]
 */

#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16

/*
 * Default Max Feed Rate (mm/s)
 * Override with M203
 * X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

#ifndef LIMITED_MAX_FR_EDITING
#define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif

/*
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 */

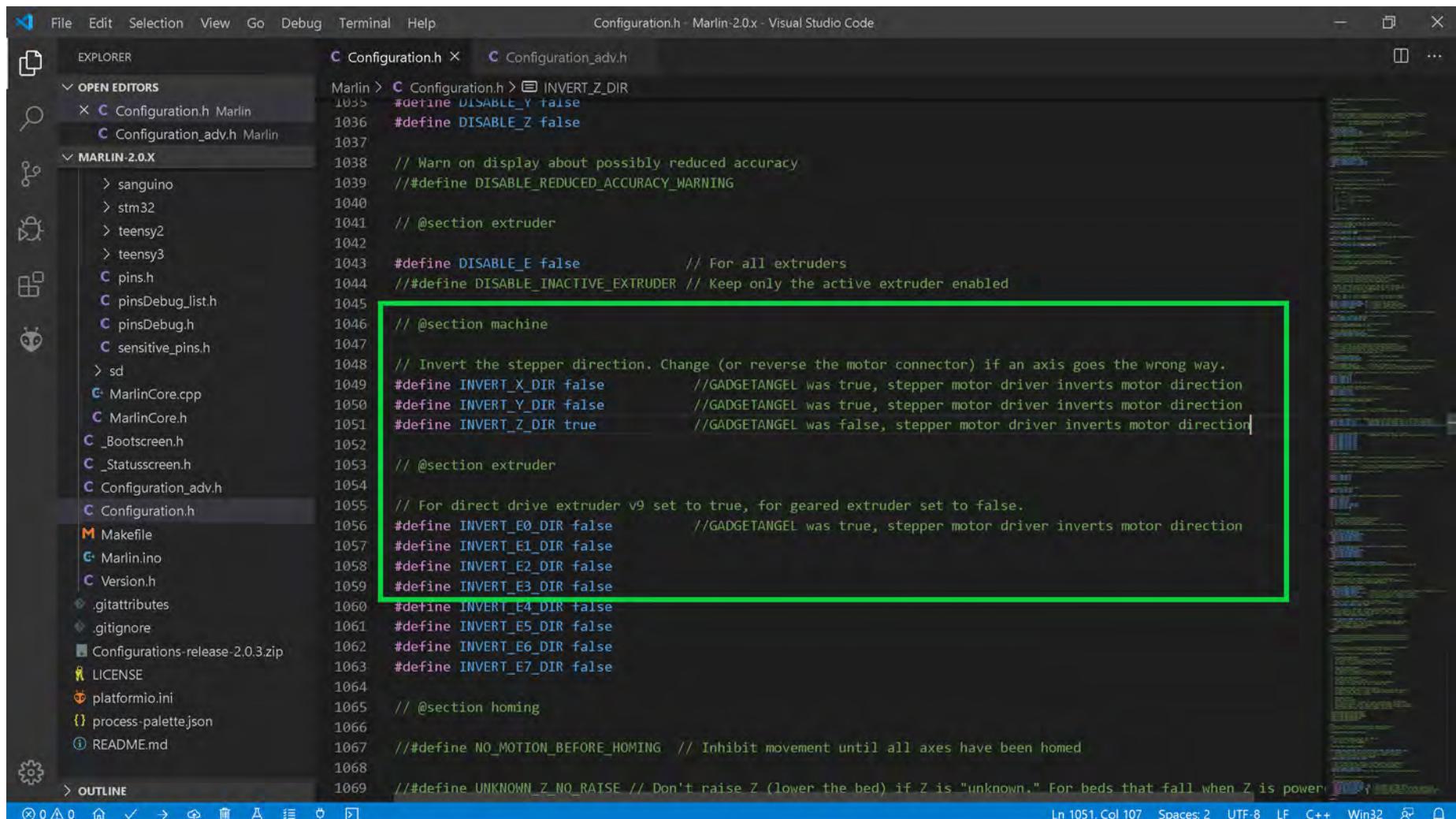
```

The line `#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 }` is highlighted with a green rectangular box. The status bar at the bottom of the code editor shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the following snippet of C++ code:

```

Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered

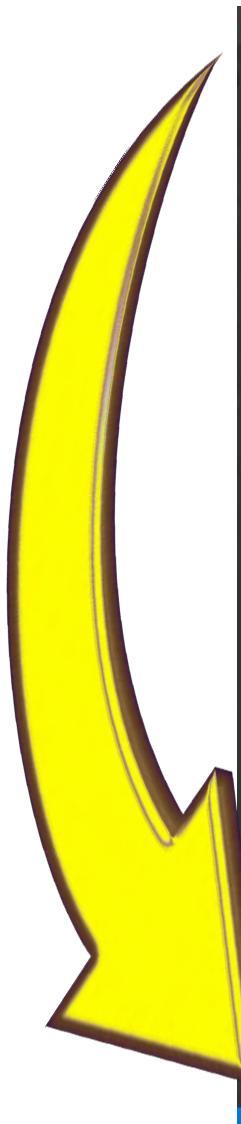
```

A green rectangular box highlights the line `#define INVERT_Z_DIR true`. The status bar at the bottom right of the code editor shows the line number as 1051, column 107, and the file type as C++.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

- The end of Marlin setup for BIQU TMC2209 V1.2 drivers in stand-alone mode for stealthChop. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

OPEN EDITORS Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src... pins\_BTT\_SKR\_common.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X samd sanguino stm32f1 stm32f4 stm32f7 teensy2 teensy3 pins.h pinsDebug.h pinsDebug.listh sensitive\_pins.h sd MarlinCore.cpp MarlinCore.h \_Bootscreen.h \_Statusscreen.h Configuration.h Configuration\_adv.h Makefile Marlin.ino Version.h .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palette.json README.md

TERMINAL

```

124 #define BAUDRATE 115200
125 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Change the board identifier here if needed in your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin  
 ======[SUCCESS] Took 130.61 seconds=====

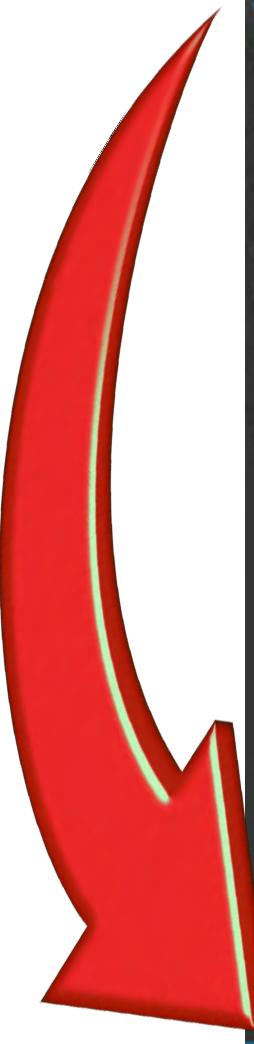
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino64p     | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUE_debug       | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

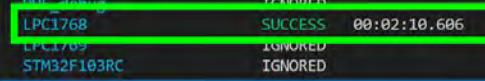
OUTLINE TIMELINE

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for StealthChop

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.



File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

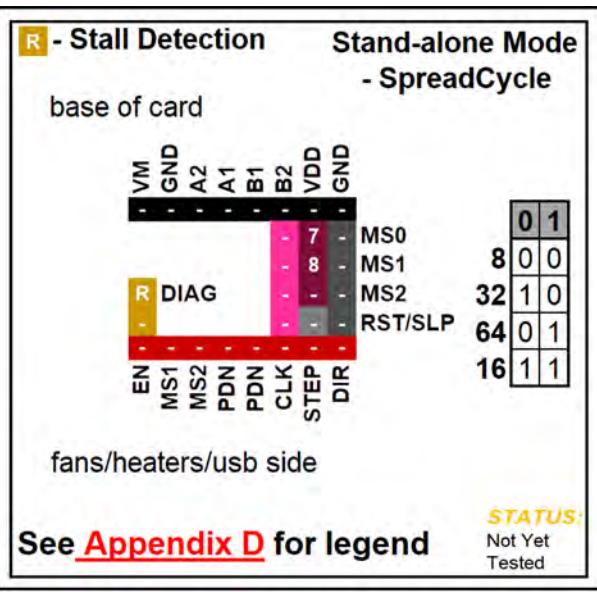
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| SUE_L103        | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

> OUTLINE > TIMELINE

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

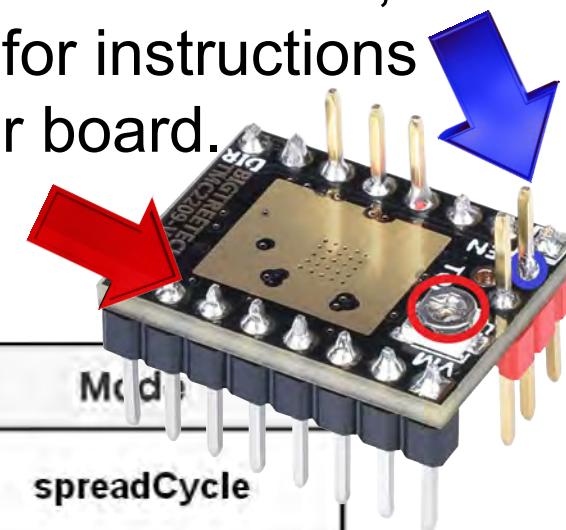


## BIQU TMC2209 V1.2

### Stand-alone Mode for SpreadCycle

**Note:** Use the potentiometer (POT) on the top of the board, as shown in **RED**; or use the board's "**V<sub>ref</sub> Test point**" location, as shown in **BLUE**, to set your V<sub>ref</sub>. See [Appendix A](#) for instructions on how to set the V<sub>ref</sub> on a driver board.

**Note:** Use 90% of the calculated V<sub>ref</sub> when tuning the stepper driver board.

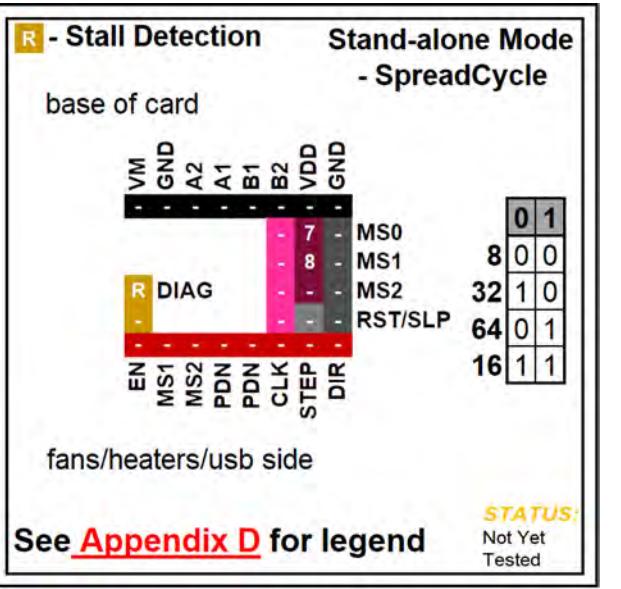


| Driver Chip                                                                                                       | MS0  | MS1  | Steps  | Interpolation | Mode        |
|-------------------------------------------------------------------------------------------------------------------|------|------|--------|---------------|-------------|
| <b>BIQU®<br/>TMC2209</b><br><small>Stand Alone Mode<br/>Maximum 64 Subdivision<br/>28V DC<br/>2.8A (peak)</small> | Low  | Low  | 1 / 8  | 1 / 256       | spreadCycle |
|                                                                                                                   | High | Low  | 1 / 32 | 1 / 256       | spreadCycle |
|                                                                                                                   | Low  | High | 1 / 64 | 1 / 256       | spreadCycle |
|                                                                                                                   | High | High | 1 / 16 | 1 / 256       | spreadCycle |

| Driving Current Calculation Formula    | $I_{MAX} = V_{ref}$<br>See Appendix B #5. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90$ | $V_{ref} = I_{MAX}$<br>See Appendix B #5. Use 50% to 90% as shown below:<br>$V_{ref} = V_{ref} * 0.90$ |
|----------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| $R_S$ (Typical Sense Resistor) = 0.11Ω |                                                                                                        |                                                                                                        |

- See next page for the legend that belongs to the above chart.

**Driver Chip Chart:**

| Driver Chip                             | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|-----------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b>                             | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
| Chip Name                               | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
| Maximum XXX Subdivision                 | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
| XXV DC xxA (peak)                       | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                         | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                         | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                         | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                         | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula     | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |
| $R_s$ (Typical Sense Resistor) = x.xx Ω |                            |      |                            |                      |                 |

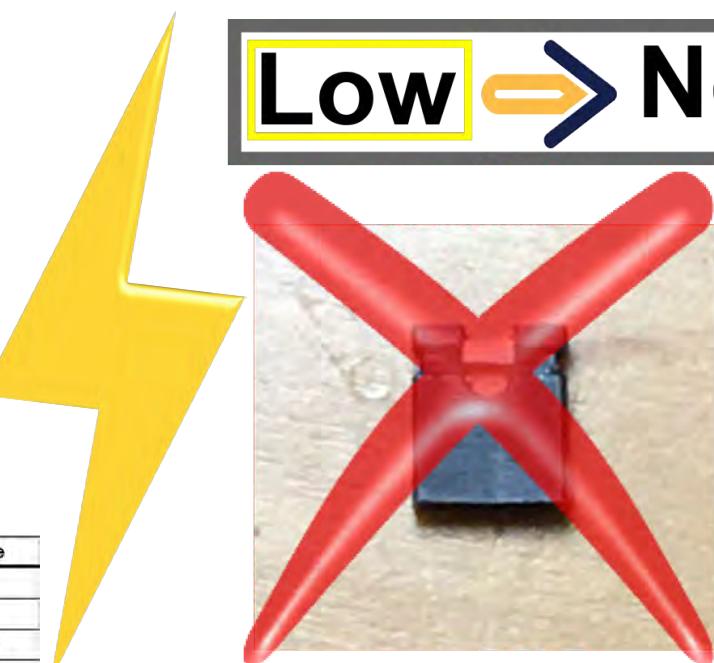
Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):

**BIQU TMC2209 V1.2****Stand-alone Mode for SpreadCycle****SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers**

**Low** → **No Jumper**



**High** → **Jumper Set**

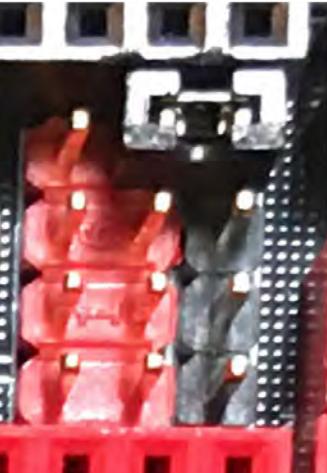
Rows:

**MS0**

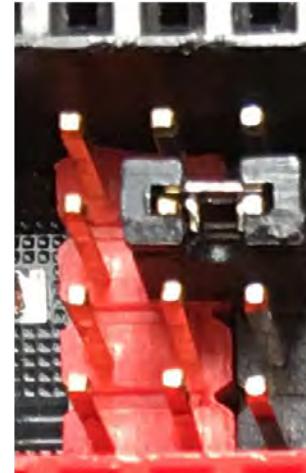
**MS1**

**MS2**

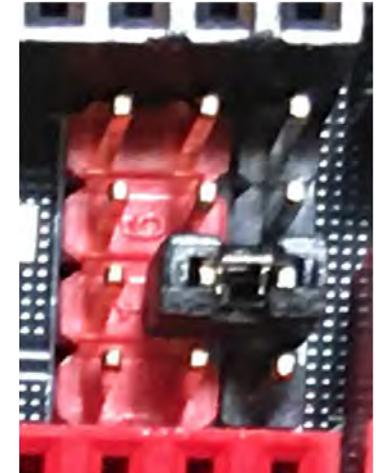
**RST/SLP**



**MS0 SET HIGH**



**MS1 SET HIGH**



**MS2 SET HIGH**

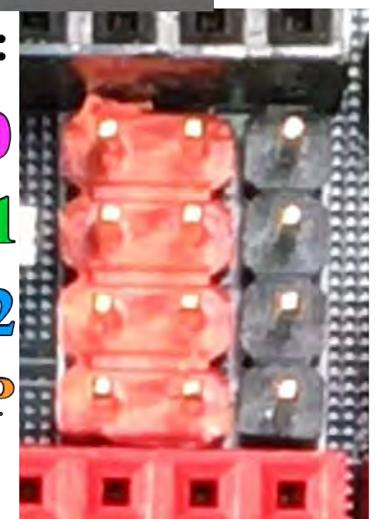
Rows:

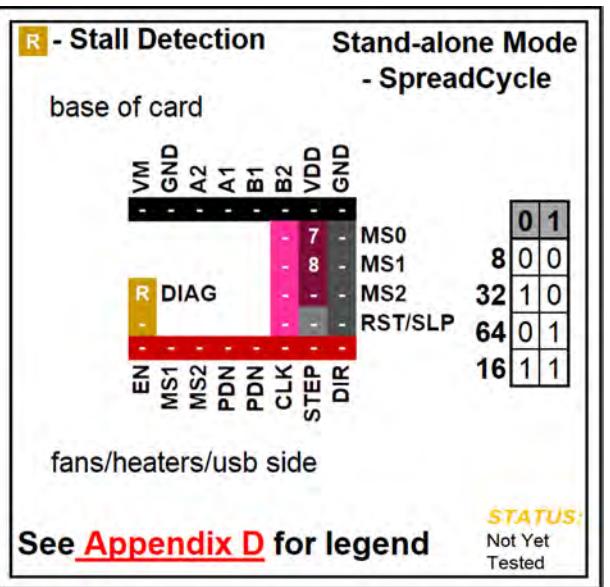
**MS0**

**MS1**

**MS2**

**RST/SLP**





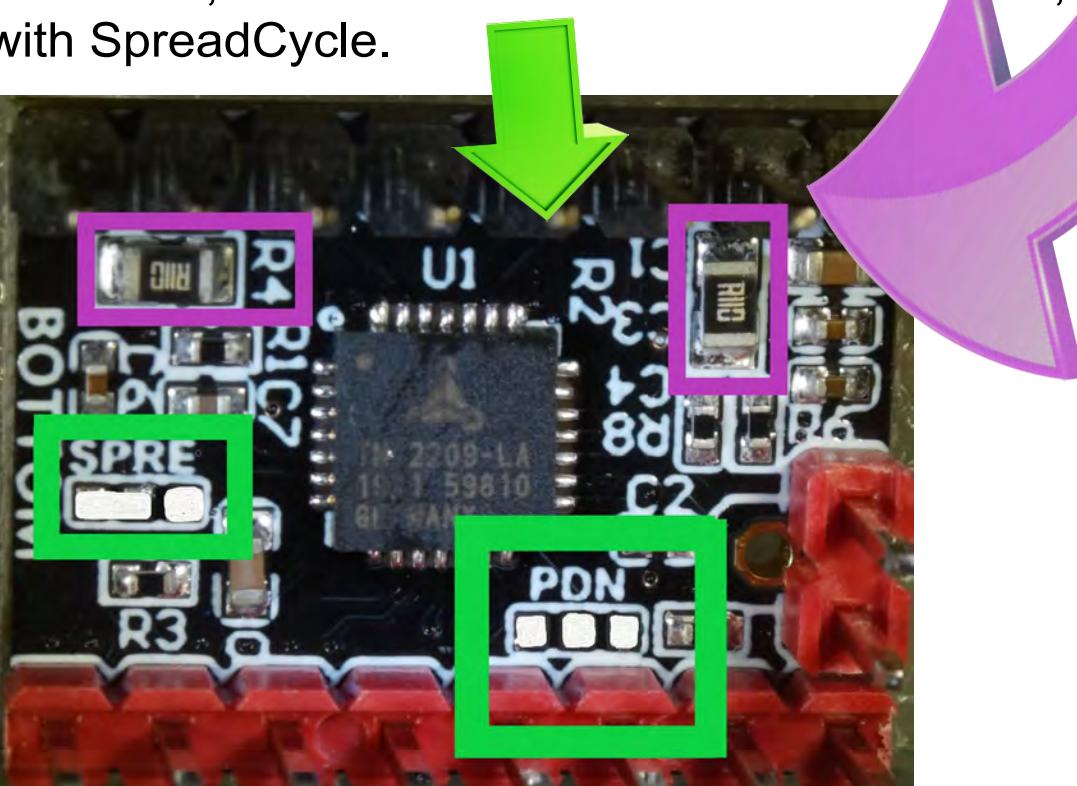
Note: To switch to stand-alone mode, none of the PDN pads should be bridged. The picture below, as seen in the GREEN boxes, shows stand-alone mode with SpreadCycle.

MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.

## BIQU TMC2209 V1.2

### Stand-alone Mode for SpreadCycle

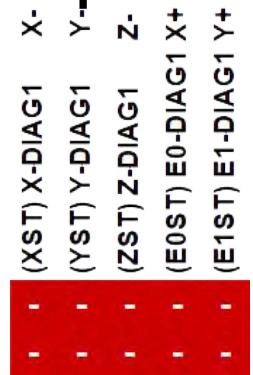
**NOTE:** The SPRE jumper is located on the bottom of the driver board. In Standalone Mode, the default setting is wired for StealthChop; i.e. the SPRE jumper is set to GND. To switch to Standalone with SpreadCycle, one needs to change the SPRE jumper on the bottom of the driver board. The **PURPLE boxes** below show the location of the current sense resistors ( $R_s$ ) for the TMC2209.



# Stand-alone SpreadCycle Mode

## Stand-alone Mode for SpreadCycle

**Note:** Ensure that XST, YST, ZST, E0S, and E1S jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode with spreadCycle enabled.



**Note:** Ensure the UART jumpers are **all empty** on the SKR V1.3 board to enable proper operation for stand-alone mode with spreadCycle enabled.

XUART



YUART



ZUART



E0 UART



E1 UART



**Stand-alone SpreadCycle Mode****SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers****Driver Socket Representation**Columns: **1 2 3**Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

DIAG

E  
N**1/2****Driver Chip Chart:**

| Driver Chip                                        | MS0                        | MS1 | MS2                        | Excitation Mode |
|----------------------------------------------------|----------------------------|-----|----------------------------|-----------------|
| LOGO                                               | Low                        | Low | Low                        | 1-Phase         |
| Chip Name                                          |                            |     |                            | W1-2 Phase      |
| Maximum XXX                                        |                            |     |                            | 2W1-2 Phase     |
| Subdivision                                        |                            |     |                            | 4W1-2 Phase     |
| XXV DC                                             |                            |     |                            | SW1-2 Phase     |
| XXA (peak)                                         |                            |     |                            | 16W1-2 Phase    |
|                                                    |                            |     |                            | 32W1-2 Phase    |
| Driving Current Calculation Formula                | $I_{MAX} = \text{FORMULA}$ |     | $V_{ref} = \text{FORMULA}$ |                 |
| $R_g$ (typical) Series Resistor (max XX $\Omega$ ) |                            |     |                            |                 |

**MS0 for Binary State Drivers:**

Driver Socket Representation:

**7 7**

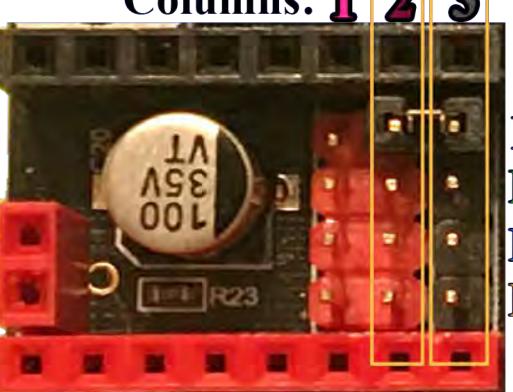
Driver Chip Chart:

**High**

Driver Socket Representation:

**--**

Driver Chip Chart:

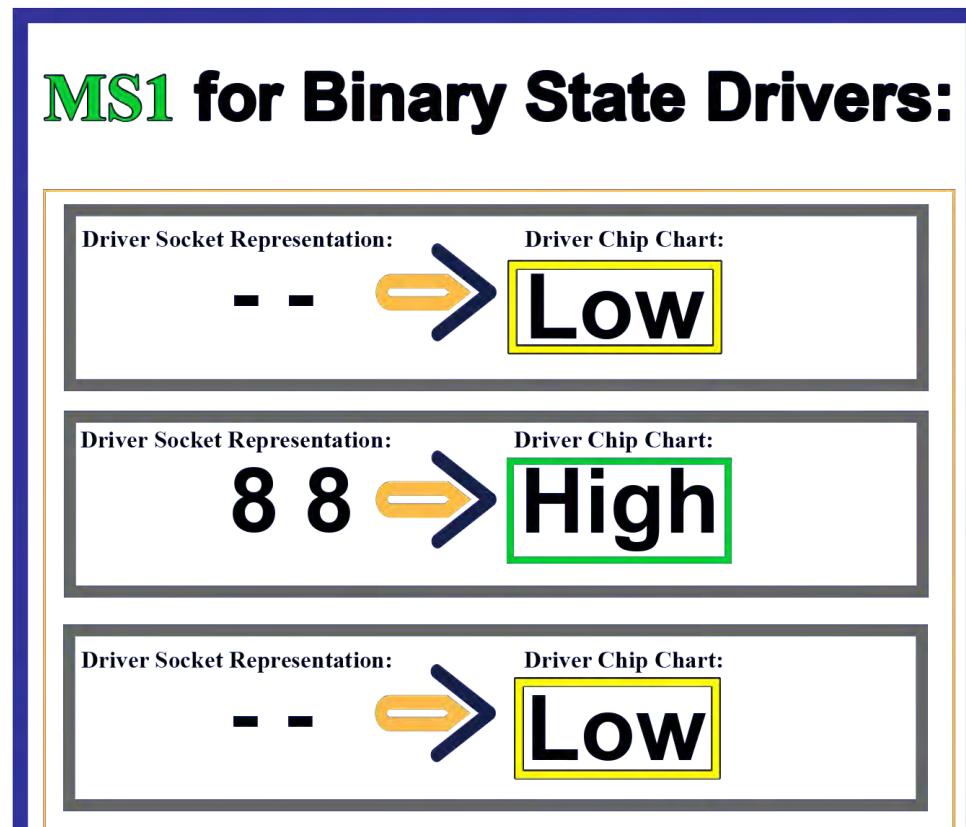
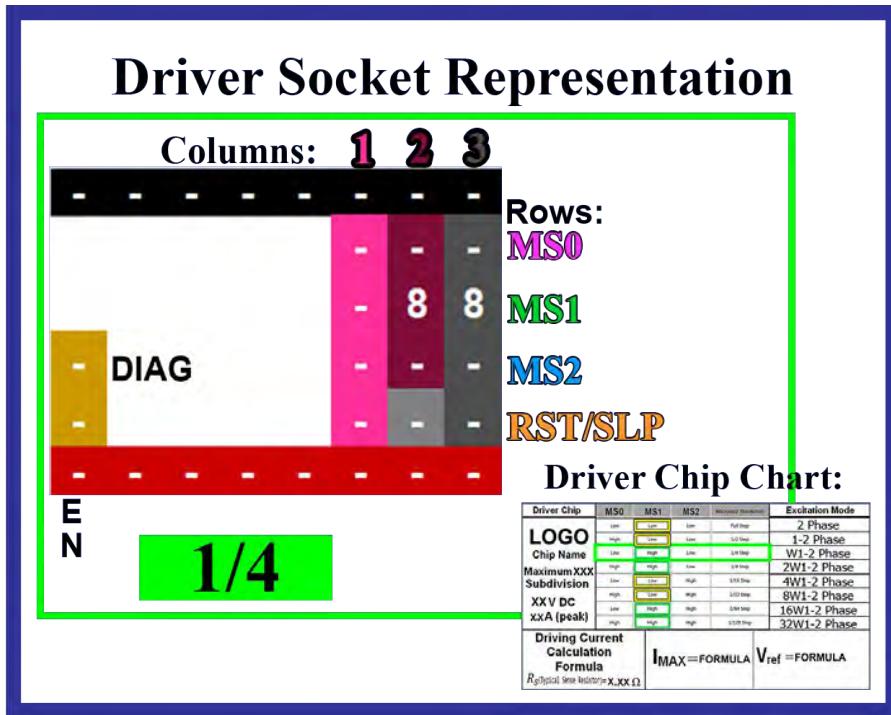
**Low****High:**Columns: **1 2 3**Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP****Meaning:**

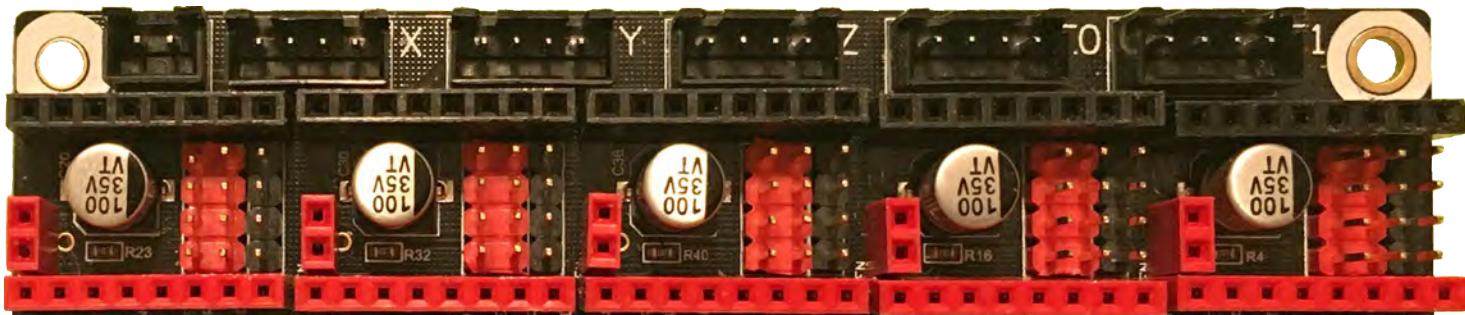
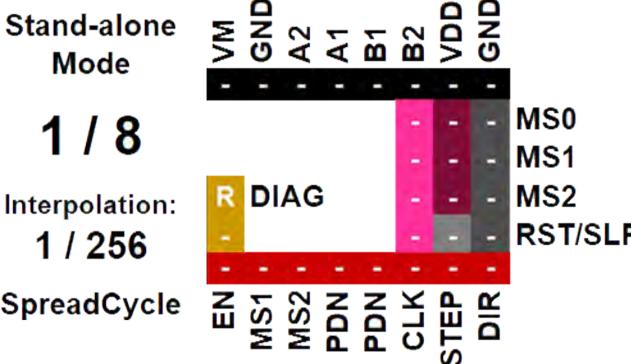
Driver Chip Chart:

**High**set Jumper between column **2** and column **3** on the **MS0** row**Low**No Jumper Set

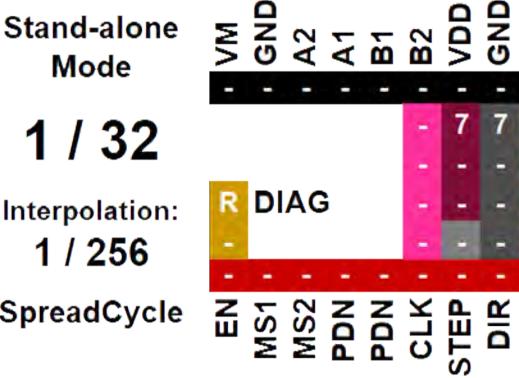
# Stand-alone SpreadCycle Mode

## Stand-alone Mode for SpreadCycle

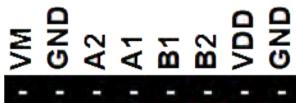


**BIQU TMC2209 V1.2****Stand-alone Mode for SpreadCycle****Stand-alone SpreadCycle Mode**

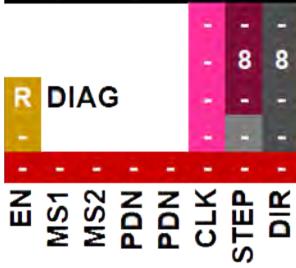
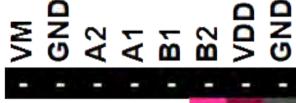
See [Appendix D](#) for legend



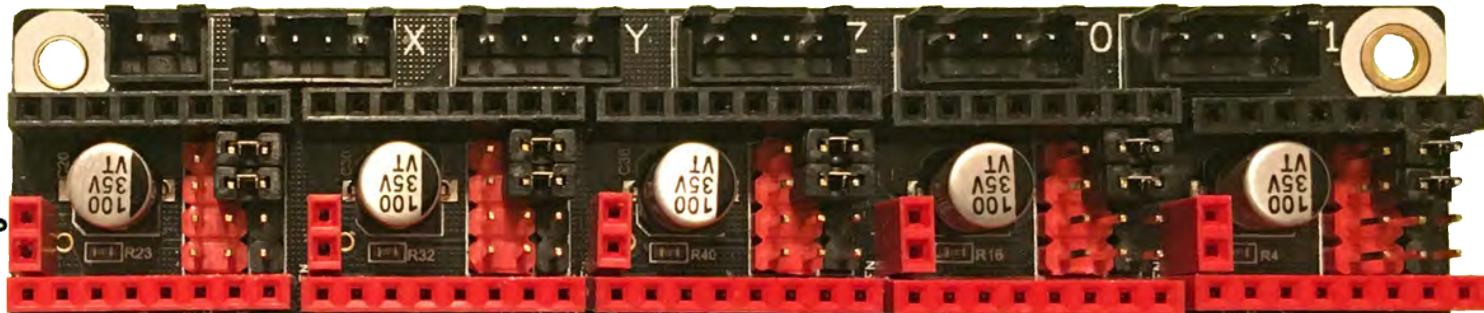
See [Appendix D](#) for legend

**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 legendStand-alone  
Mode**1 / 16**Interpolation:  
**1 / 256**

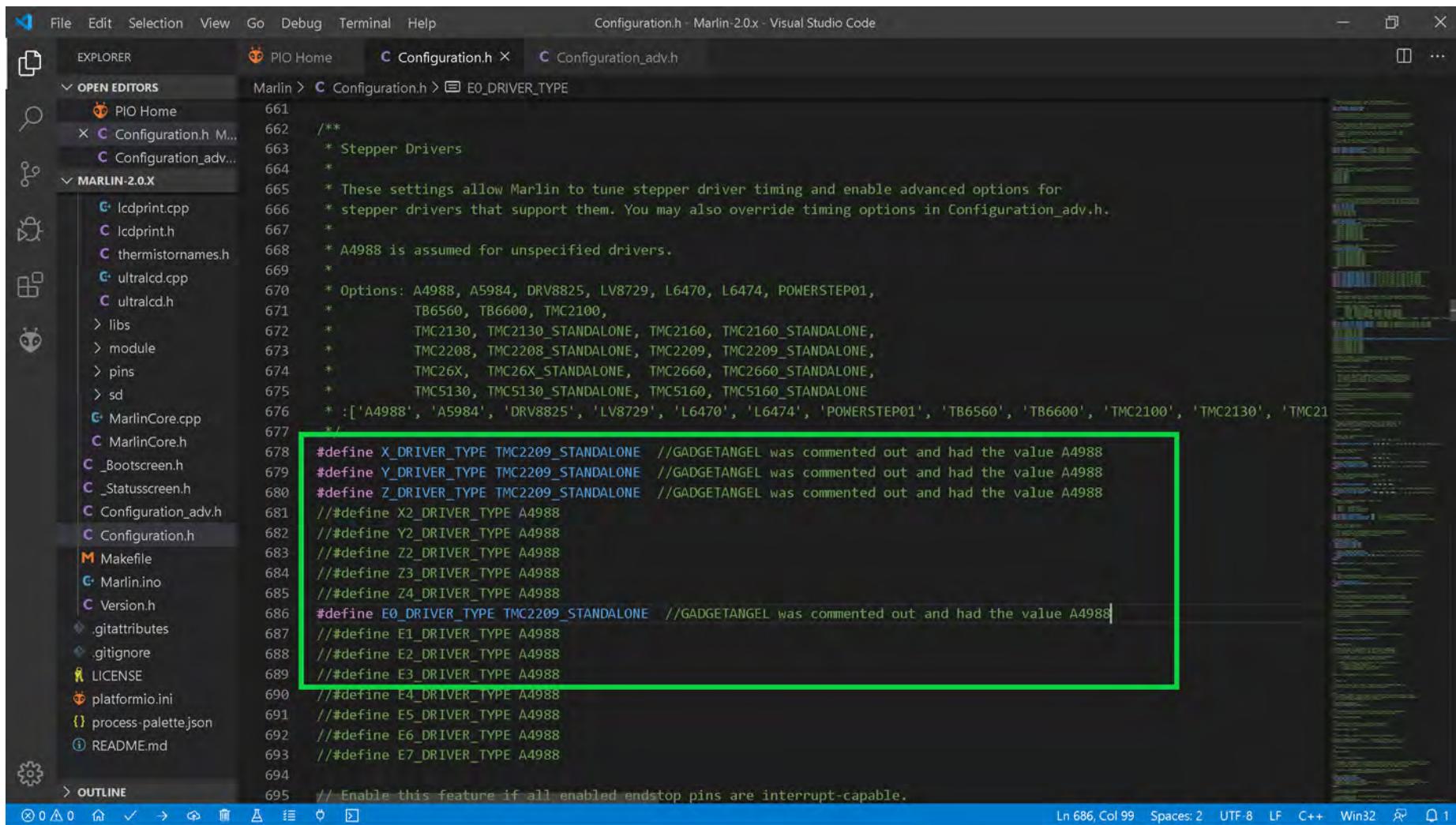
SpreadCycle

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 ("//").



```

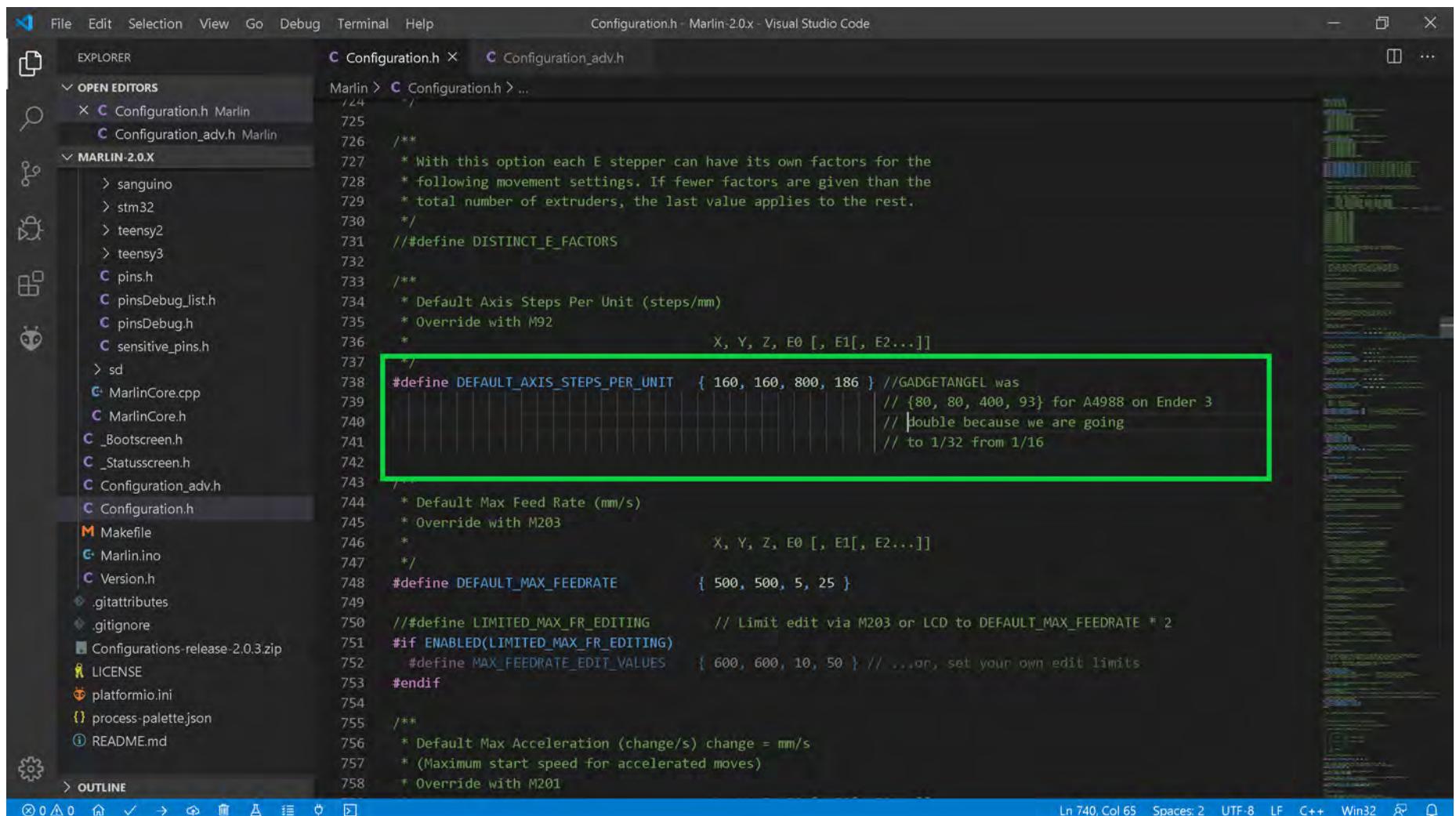
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
 PIO Home Configuration.h M...
 Configuration_adv.h
MARLIN-2.0.X
 LCDprint.cpp
 LCDprint.h
 thermistornames.h
 ultralcd.cpp
 ultralcd.h
 libs
 module
 pins
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration_adv.h
 Configuration.h
 Makefile
 Marlin.ino
 Version.h
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
Ln 686, Col 99 Spaces: 2 UTF-8 LF C++ Win32 1
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC2209', 'TMC26X', 'TMC5130']
676 */
677 #define X_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
678 #define Y_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
679 #define Z_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
680 //#define X2_DRIVER_TYPE A4988
681 //#define Y2_DRIVER_TYPE A4988
682 //#define Z2_DRIVER_TYPE A4988
683 //#define Z3_DRIVER_TYPE A4988
684 //#define Z4_DRIVER_TYPE A4988
685 #define E0_DRIVER_TYPE TMC2209_STANDALONE //GADGETANGEL was commented out and had the value A4988
686 //#define E1_DRIVER_TYPE A4988
687 //#define E2_DRIVER_TYPE A4988
688 //#define E3_DRIVER_TYPE A4988
689 //#define E4_DRIVER_TYPE A4988
690 //#define E5_DRIVER_TYPE A4988
691 //#define E6_DRIVER_TYPE A4988
692 //#define E7_DRIVER_TYPE A4988
693
694 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle

- Since I desire to use 1/32 stepping, and we are changing from A4988 stepper motor drivers on the Ender 3 to TMC2209 stepper motor drivers for each axis and the extruder stepper motor driver, we will be going from 1/16 stepping to 1/32 stepping. So we are doubling our STEPS. Therefore, we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the GREEN box below.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following C++ code snippet:

```

 ...
 * With this option each E stepper can have its own factors for the
 * following movement settings. If fewer factors are given than the
 * total number of extruders, the last value applies to the rest.
 */
//#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [, E1[, E2...]]
 */

#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
// {80, 80, 400, 93} for A4988 on Ender 3
// Double because we are going
// to 1/32 from 1/16

/*
 * Default Max Feed Rate (mm/s)
 * Override with M203
 * X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE { 500, 500, 5, 25 }

#ifndef LIMITED_MAX_FR_EDITING
#define MAX_FEEDRATE_EDIT_VALUES { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif

/*
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
*/

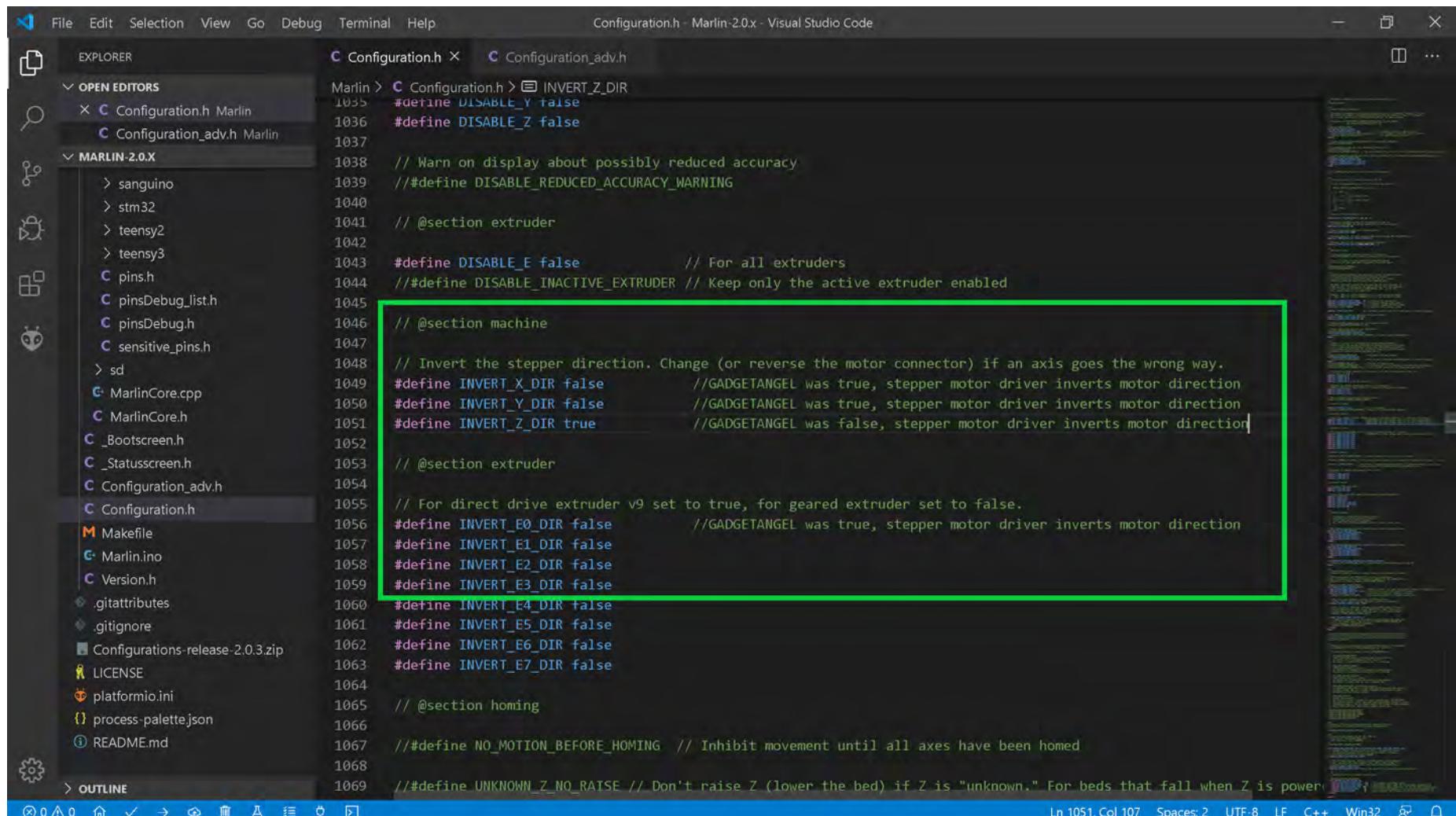
```

The line `#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 }` is highlighted with a green rectangular box. The status bar at the bottom of the code editor shows: Ln 740, Col 65, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU 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



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
 Configuration.h Marlin 1035 #define DISABLE_Y false
 Configuration_adv.h Marlin 1036 #define DISABLE_Z false
 1037
 1038 // Warn on display about possibly reduced accuracy
 1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
 1040
 1041 // @section extruder
 1042
 1043 #define DISABLE_E false // For all extruders
 1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
 1045
 // @section machine
 1046
 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
 1047 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1048 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1049 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
 1050
 // @section extruder
 1051
 // For direct drive extruder v9 set to true, for geared extruder set to false.
 1052 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
 1053 #define INVERT_E1_DIR false
 1054 #define INVERT_E2_DIR false
 1055 #define INVERT_E3_DIR false
 1056 #define INVERT_E4_DIR false
 1057 #define INVERT_E5_DIR false
 1058 #define INVERT_E6_DIR false
 1059 #define INVERT_E7_DIR false
 1060
 // @section homing
 1061
 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
 1062
 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered
 1063
 1064
 1065
 1066
 1067
 1068
 1069

```

Ln 1051, Col 107 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for 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.

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS Configuration.h pins\_BTT\_SKR\_V1\_3.h pins\_BTT\_SKR\_common.h Configuration\_adv.h

MARLIN-2.0.X Configuration.h Marlin pins\_BTT\_SKR\_V1\_3.h Marlin\src... pins\_BTT\_SKR\_common.h Marlin... Configuration\_adv.h Marlin

```

124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
Archiving .pio\build\LPC1768\libFrameworkArduino.a  
Linking .pio\build\LPC1768\firmware.elf  
Checking size .pio\build\LPC1768\firmware.elf  
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)  
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
Building .pio\build\LPC1768\firmware.bin  
===== [SUCCESS] Took 130.61 seconds =====

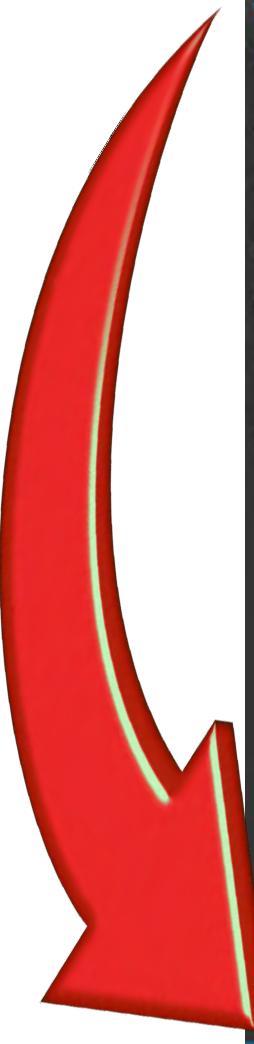
| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino64p     | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| esp8266         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

OUTLINE TIMELINE

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in Stand-alone Mode for SpreadCycle

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.







**Configuration.h - Marlin-2.0.x - Visual Studio Code**

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE

```

```

Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

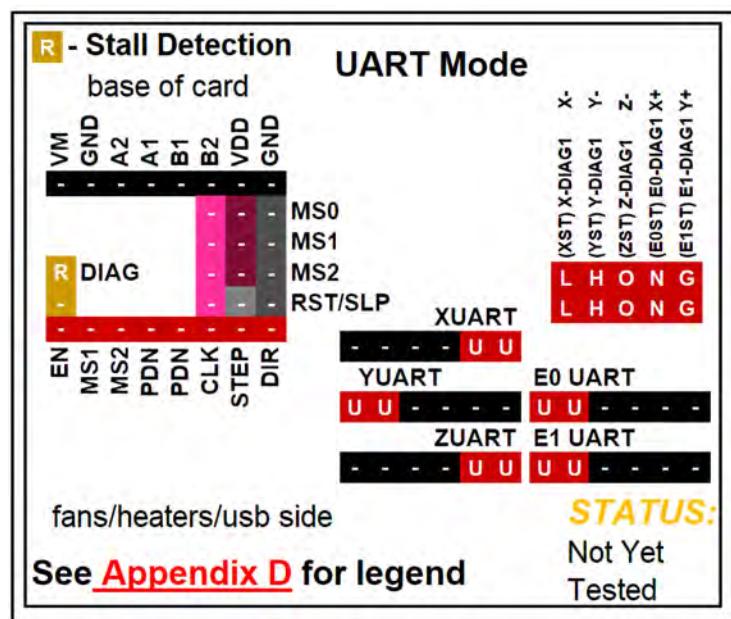
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| SUE_L103        | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

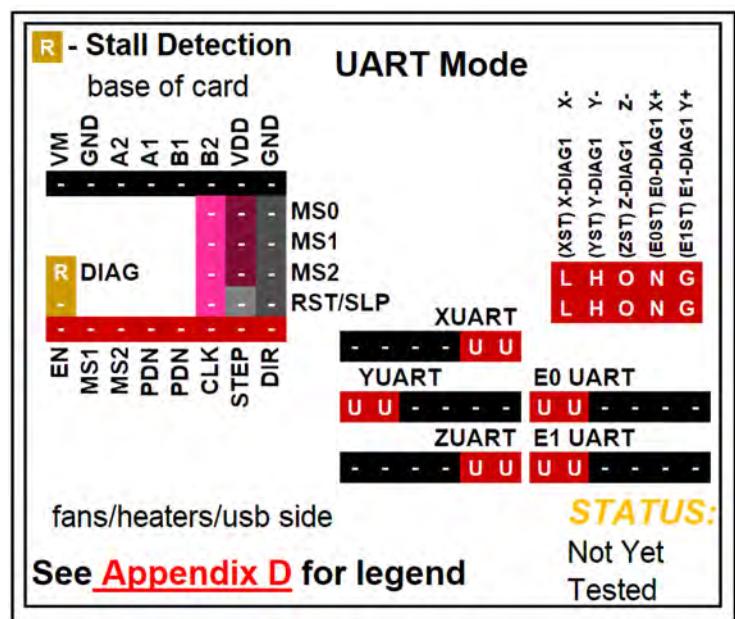
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC2209 V1.2****UART Mode**

**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

| Driver Chip                                                                                     | Steps are set inside<br>of your Firmware                                                               |                                                                                                        |
|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>BIQU®</b><br><b>TMC2209</b><br>UART Mode<br>Maximum 256 Subdivision<br>28V DC<br>2.8A (peak) |                                                                                                        |                                                                                                        |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.11Ω            | $I_{MAX} = V_{ref}$<br>See Appendix B #6. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90$ | $V_{ref} = I_{MAX}$<br>See Appendix B #6. Use 50% to 90% as shown below:<br>$V_{ref} = V_{ref} * 0.90$ |



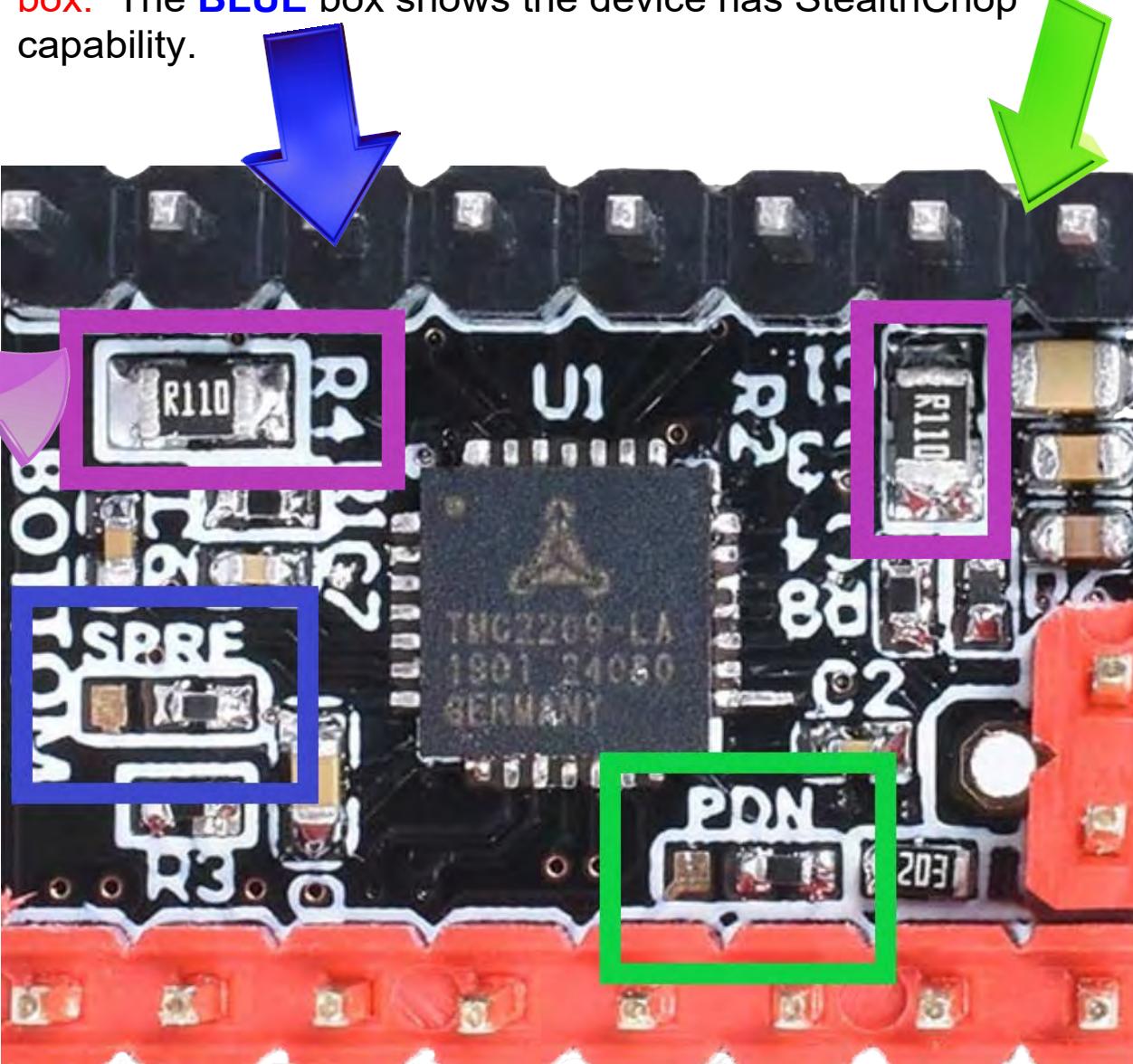
The **PURPLE** boxes show the location of the current sense resistors ( $R_s$ ).

**MOST BIQU TMC2209 V1.2 driver boards, when purchased for UART mode, will have the correct PDN pads already soldered together, located on the bottom of the driver board.**

## BIQU TMC2209 V1.2

### UART Mode

**Note:** To ensure your TMC2209 is in UART mode, look on the bottom of the driver board for the PDN pads. Two of the three pads should be bridged together. **If a bridge exists then the device is in UART Mode, as seen in the GREEN box.** The **BLUE** box shows the device has StealthChop capability.



**UART Mode****R - Stall Detection**

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE" so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

$R_s = R050$  is 0.05 Ohms

$R_s = R062$  is 0.062 Ohms

$R_s = R068$  is 0.068 Ohms

$R_s = R075$  is 0.075 Ohms

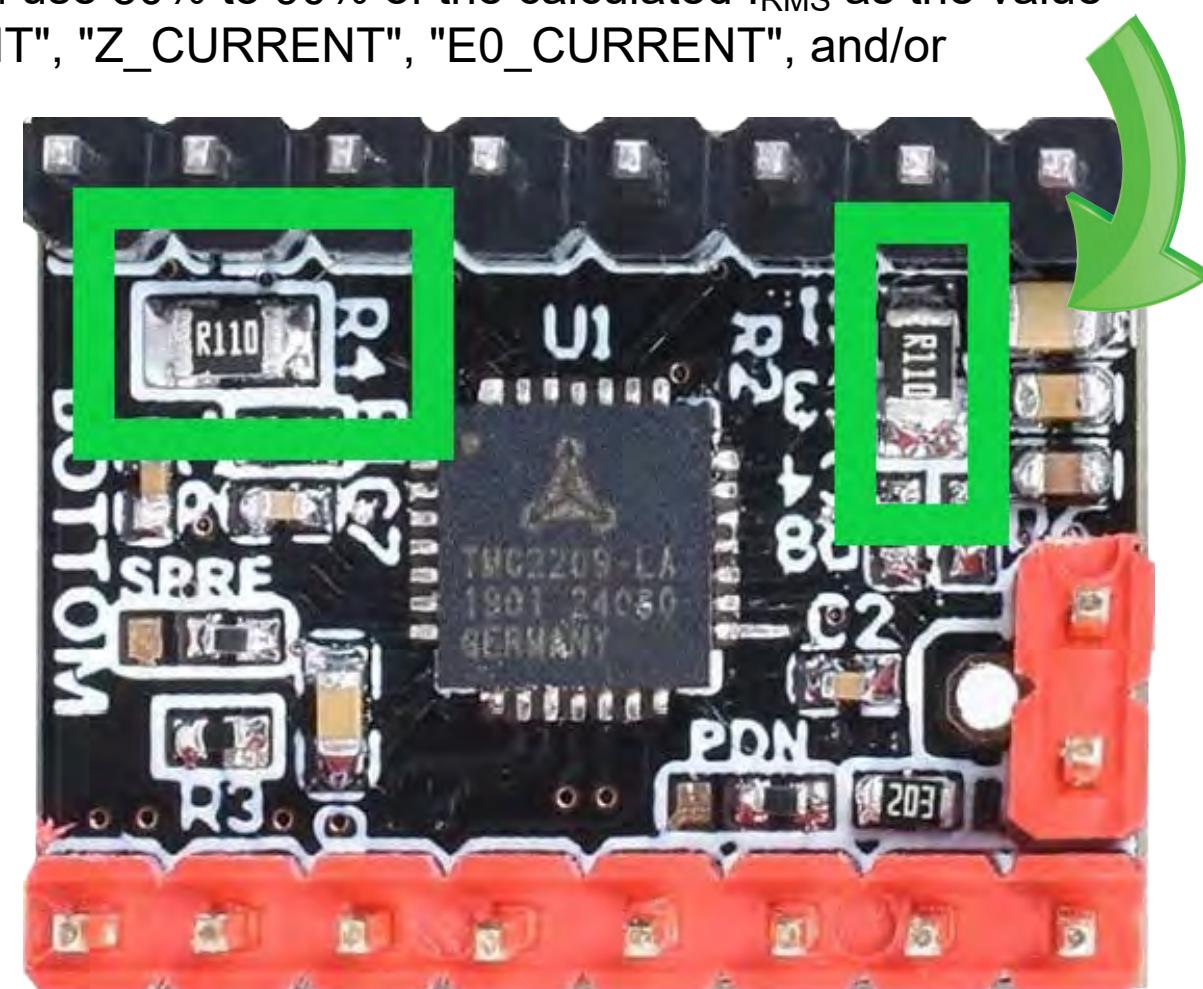
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

$R_s = R150$  is 0.15 Ohms

$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



BIQU TMC2209 V1.2

## UART Mode

## UART Mode

## R - Stall Detection

|                 |    |                 |    |                 |    |                 |    |                 |    |
|-----------------|----|-----------------|----|-----------------|----|-----------------|----|-----------------|----|
| L               | H  | O               | -  | -               | L  | H               | O  | N               | G  |
| L               | H  | O               | -  | -               | L  | H               | O  | N               | G  |
| (XST) X-DIAG1   | X- |
| (YST) Y-DIAG1   | Y- |
| (ZST) Z-DIAG1   | Z- |
| (E0ST) E0-DIAG1 | X+ |
| (E1ST) E1-DIAG1 | Y+ |

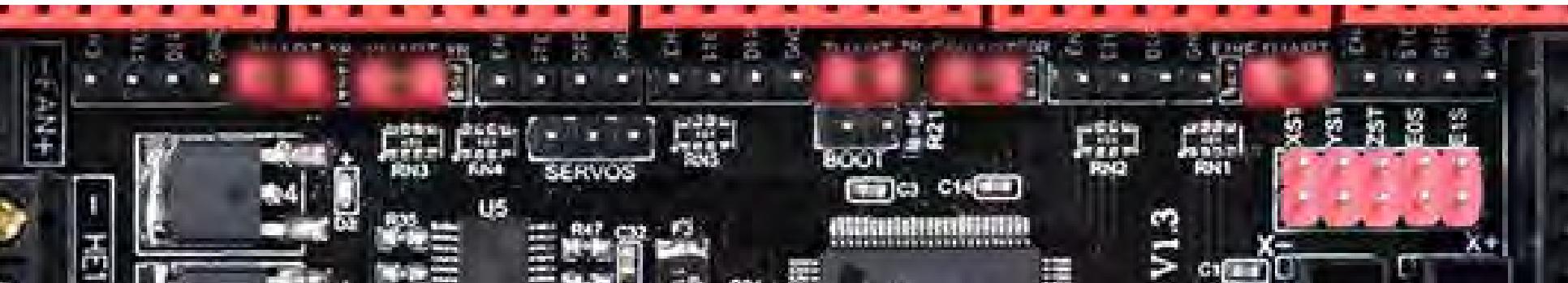
**Note:** If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis, "H" for Y axis, and "O" for Z axis) on the board.

**Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.**



|          |          |    |
|----------|----------|----|
| (X S T)  | X-DIAG1  | X- |
| (Y S T)  | Y-DIAG1  | Y- |
| (Z S T)  | Z-DIAG1  | Z- |
| (E0 S T) | E0-DIAG1 | X+ |
| (E1 S T) | E1-DIAG1 | Y+ |

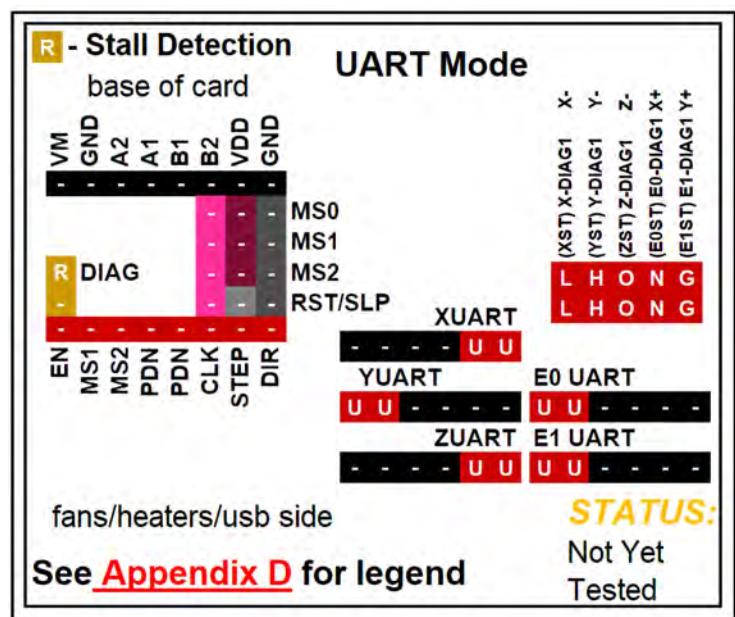
**Note:** If sensor-less homing is **not wanted** ensure the following JUMPER(s) **are empty**: "L", "H", "O", "N", and "G".



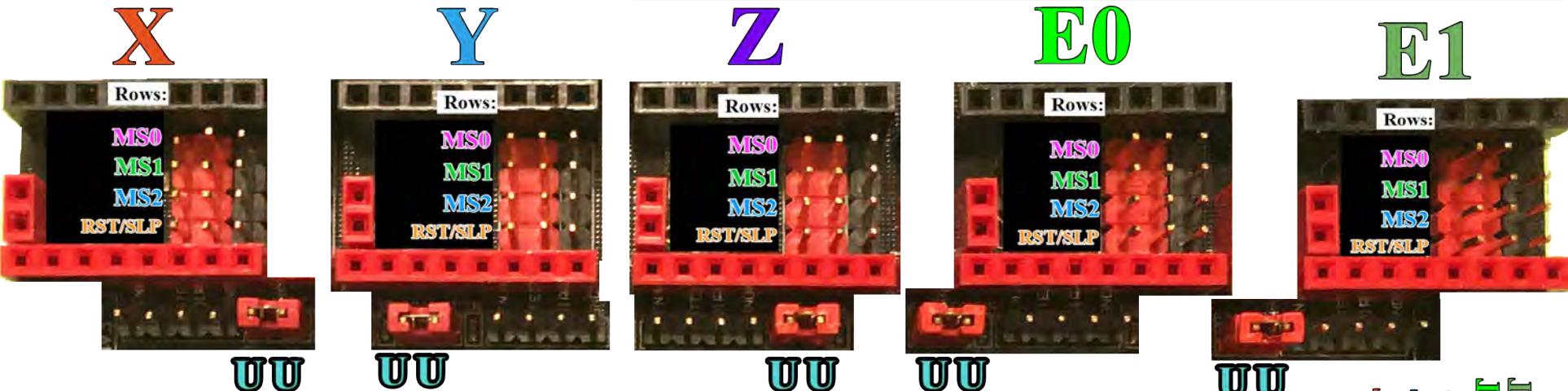
**UART Mode****R** - Stall DetectionUART Mode

# Note: Set the "U" Jumper(s) for UART MODE!





Driver Socket Representation: **UU** → Jumper set



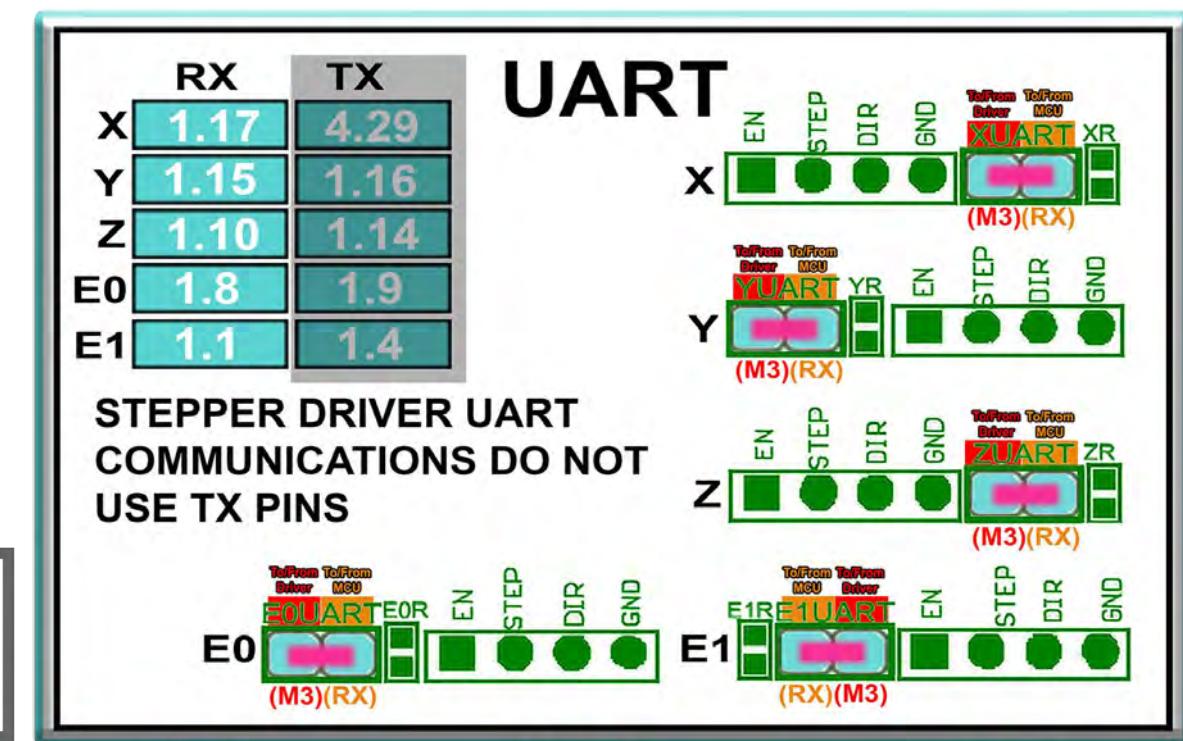
**SET 'ST' JUMPER(s) to enable Sensor-less Homing – for drivers with stallGuard™ feature**

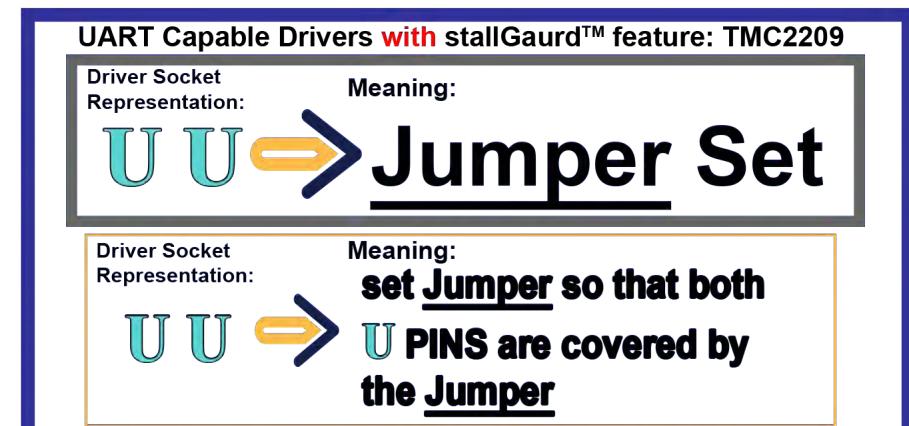
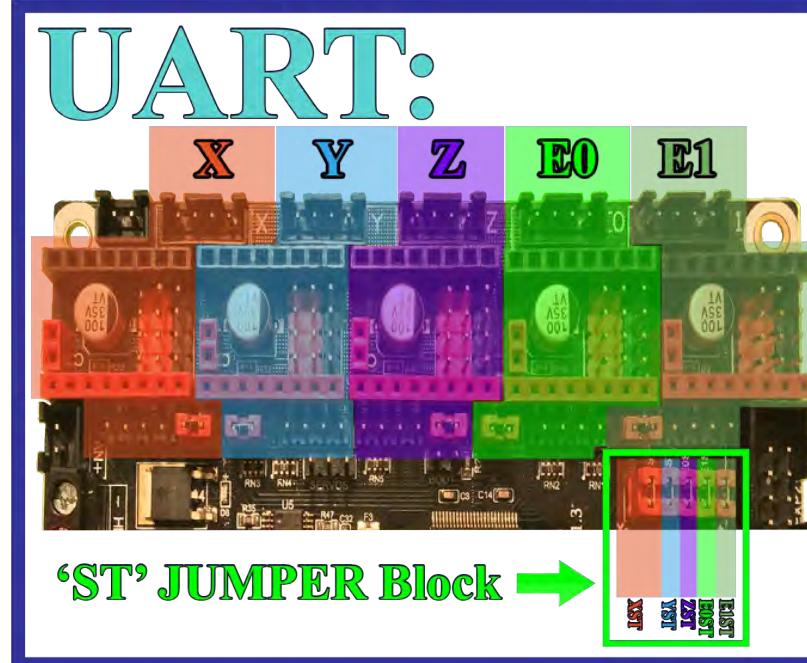
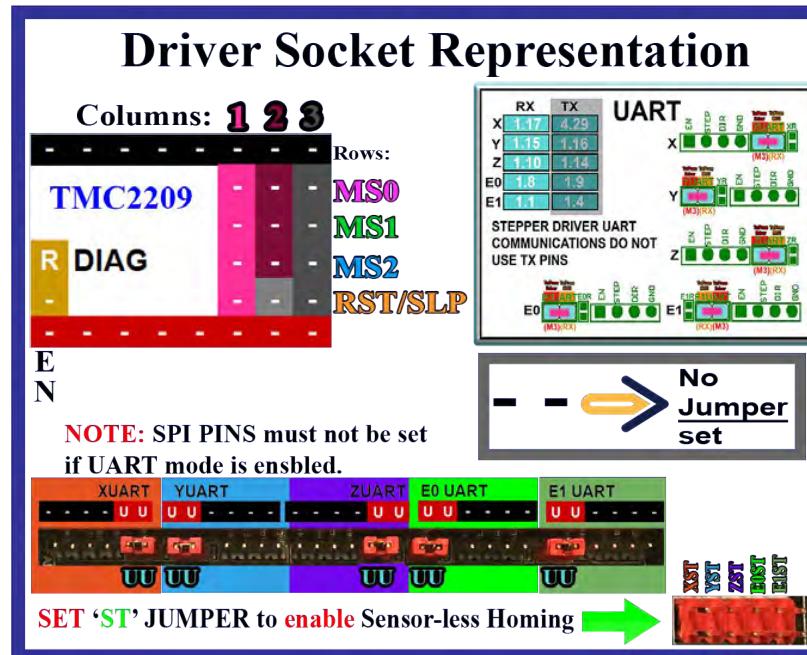


## BIQU TMC2209 V1.2

### UART Mode

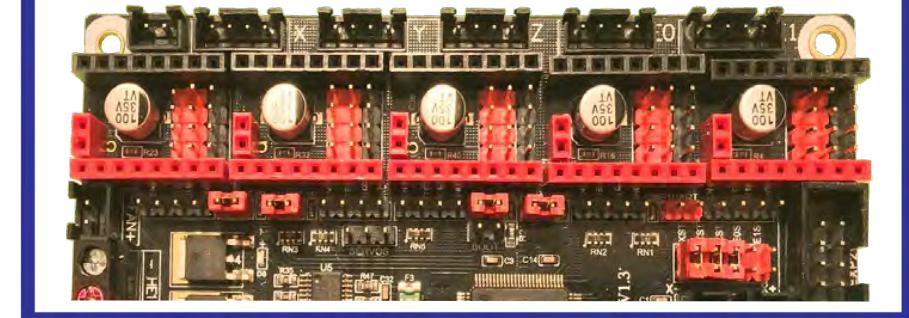
**SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode**



**BIQU TMC2209 V1.2****UART Mode****UART Mode****R - Stall Detection****SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in UART Mode**

**Each Axis has its own UART PINS (UU).** You can set as many axes as you need. The **UART:** picture shows all axes having UART enabled **AND** all **'ST' JUMPERS SET.** All other PINS need to be empty for UART mode to work properly.

Here is an example of (TMC2209 UART mode) only the X, Y, Z, and E0 axes having UART enabled **AND** **'ST' JUMPERS set for X, Y, and Z Axes:**



**BIQU TMC2209 V1.2****UART Mode****UART Mode****R - Stall Detection****Information on Sensor-less Homing****Driver Socket Representation****SPI Capable:****Columns: 1 2 3**

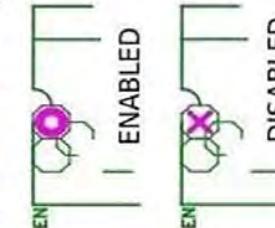
**NOTE: UART PINS must not be set if SPI mode is enabled.**



**SET 'ST' JUMPER to enable Sensor-less Homing:**

**UART Capable:****Columns: 1 2 3****Rows:****MS0****MS1****MS2****RST/SLP****E****N**

**NOTE: SPI PINS must not be set if UART mode is enabled.**

**XUART YUART****ZUART E0 UART****E1 UART****U U****U U****U U****U U****U U****U U****STALLGUARD (Sensor-less Homing)****DIAG PIN ENDSTOP****X X-DIAG1 1.29 X-****Y Y-DIAG1 1.27 Y-****Z Z-DIAG1 1.25 Z-****E0 E0-DIAG1 1.28 X+****E1 E1-DIAG1 1.26 Y+**

**To/From Driver DIAG pin** → 'ST' Jumper Block

**To/From MCU Endstops** → 'ST' Jumper Block

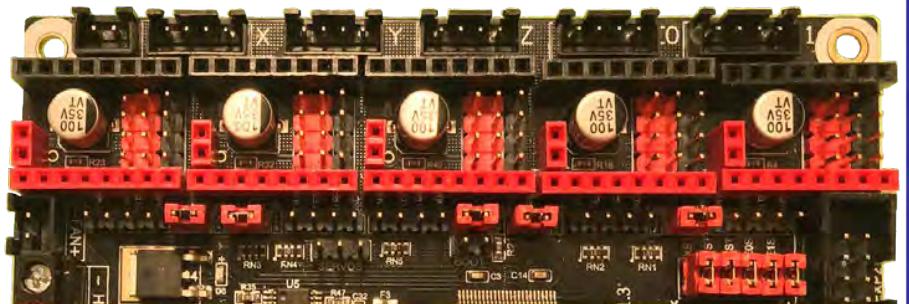
**Note Concerning the TMC2209 in UART Mode ONLY:**

If using limit switches/endstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

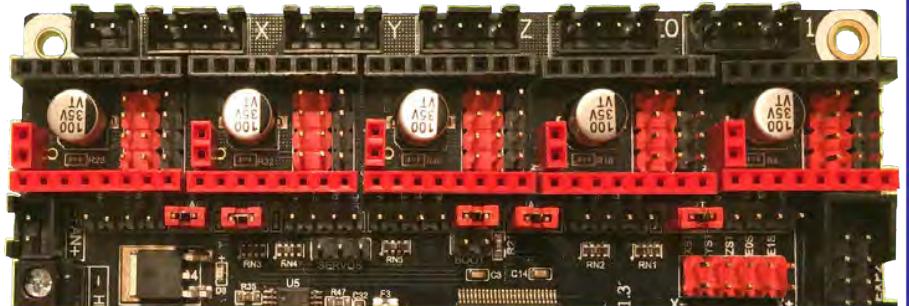
**Note:** For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

**Sensor-less Homing Capable Drivers:****SPI Capable Drivers: TMC2130, TMC5160 & TMC5161****UART Capable Drivers: TMC2209**

**If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are SET for ALL Axes:**



**If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are REMOVED for ALL Axes:**



**BIQU TMC2209 V1.2****UART Mode****UART Mode****R - Stall Detection****Driver Socket Representation of 'ST' JUMPER Block**

Columns: 7 8 9 10 11

Columns: 7 8 9 10 11

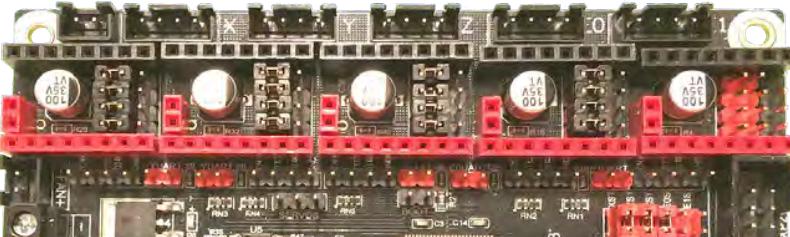
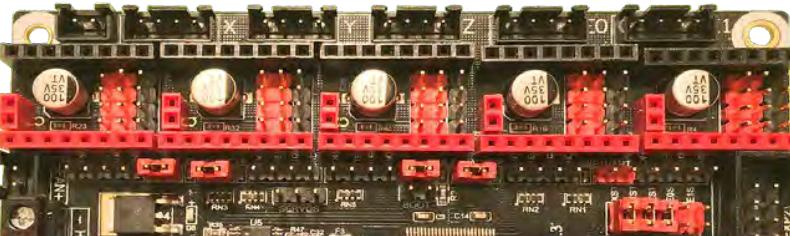
|    |               |        |               |        |               |    |                 |        |                 |
|----|---------------|--------|---------------|--------|---------------|----|-----------------|--------|-----------------|
| X- | (XST) X-DIAG1 | Y-     | (YST) Y-DIAG1 | Z-     | (ZST) Z-DIAG1 | X+ | (E0ST) E0-DIAG1 | Y+     | (E1ST) E1-DIAG1 |
|    | [E0ST]        | [E1ST] |               | [E0ST] | [E1ST]        |    | [E0ST]          | [E1ST] |                 |

Rows:

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | L | H | O | - |
| 2 | L | H | O | - |

Rows:

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | L | H | O | N | G |
| 2 | L | H | O | N | G |

**Sensor-less Homing on X, Y, & Z via SPI mode:****Sensor-less Homing on X, Y, & Z via UART mode:****Sensor-less Homing Capable Drivers:**

SPI Capable Drivers: TMC2130, TMC5160 &amp; TMC5161

UART Capable Drivers: TMC2209



Driver  
Socket  
Representation:  
 Meaning:  
**set Jumper between row 1 and row 2 in column 7**

Driver  
Socket  
Representation:  
 Meaning:  
**set Jumper between row 1 and row 2 in column 8**

Driver  
Socket  
Representation:  
 Meaning:  
**set Jumper between row 1 and row 2 in column 9**

Driver  
Socket  
Representation:  
 Meaning:  
**set Jumper between row 1 and row 2 in column 10**

Driver  
Socket  
Representation:  
 Meaning:  
**set Jumper between row 1 and row 2 in column 11**

# BIQU TMC2209 V1.2

## UART Mode

### UART Mode

#### R - Stall Detection

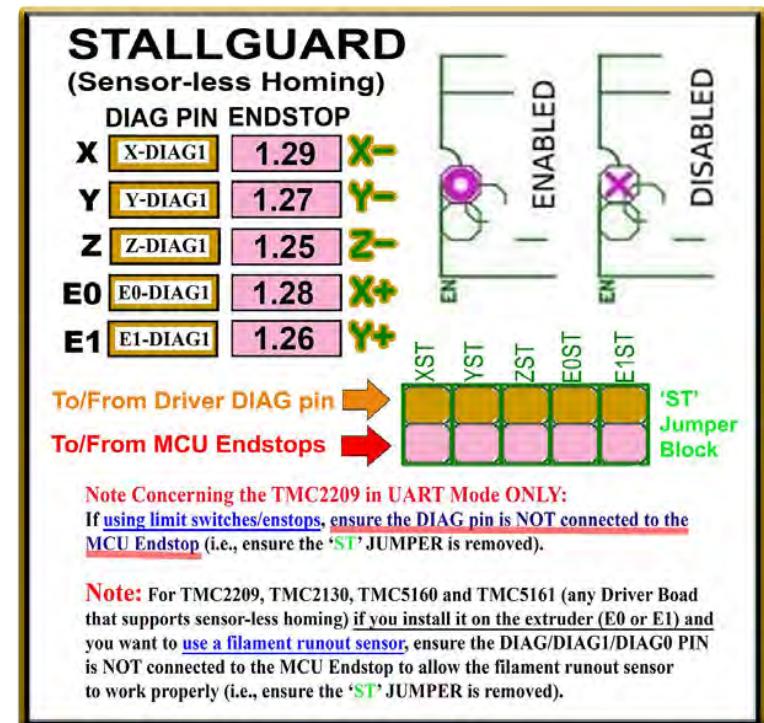
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capability of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

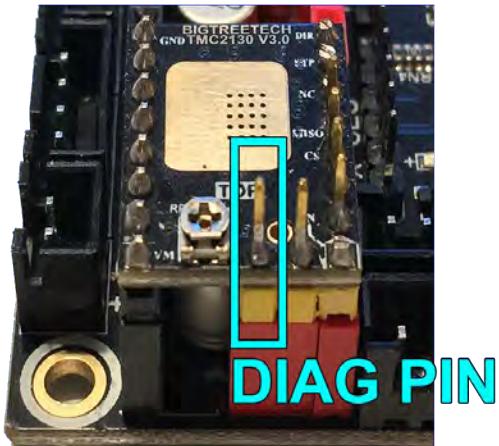
- Extruder Axis (E0 or E1)
- Z Axis, if a BLTouch is used to Home instead of a physical endstop

## UART Mode

## R - Stall Detection

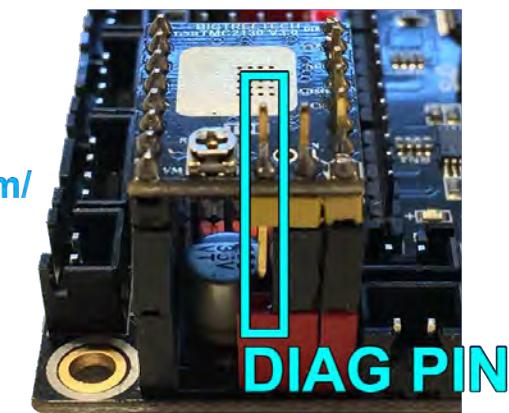
To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board AND** the '**ST'** **JUMPER** is set in the '**'ST' JUMPER Block** for the **Axis.**





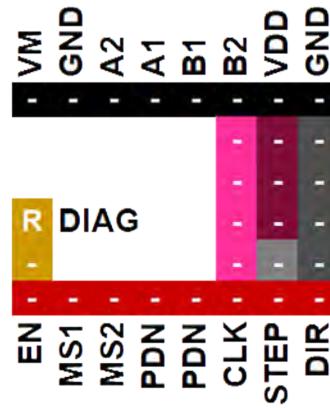
To **disable** sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the **SKR V1.3 Board** **OR** remove the '**ST**' JUMPER from the '**ST**' JUMPER Block.

**Link to stackable  
header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>**



**BIQU TMC2209 V1.2**UART Mode**UART Mode****R - Stall Detection**

**Note:** Set the "U" Jumper(s) for UART MODE!

**UART**

MS0  
MS1  
MS2  
RST/SLP

EN

MS1

MS2

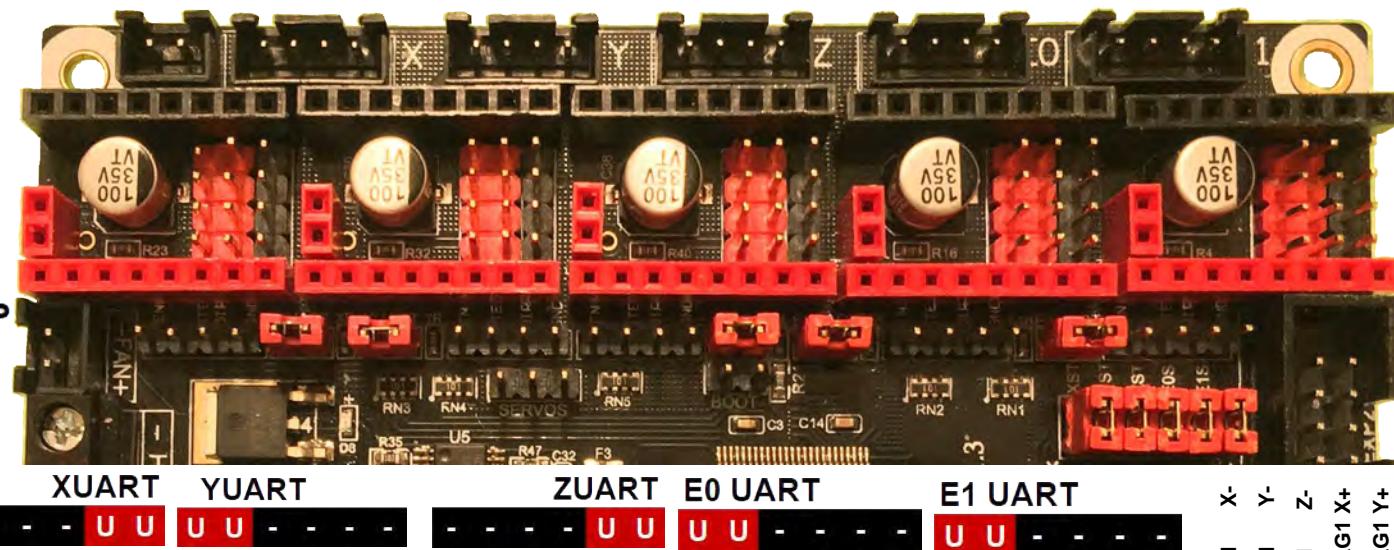
PDN

PDN

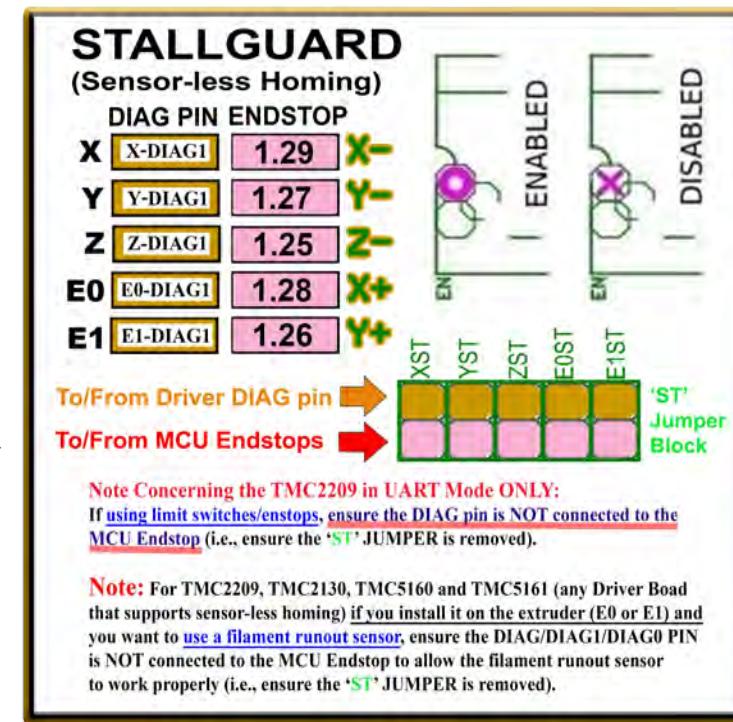
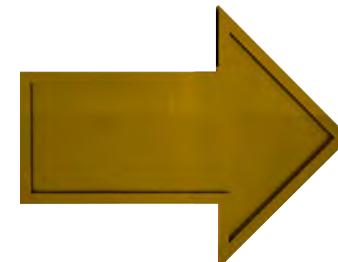
CLK

STEP

DIR

See [Appendix D](#) for legend

**Note: TMC2209 has sensor-less homing capability**



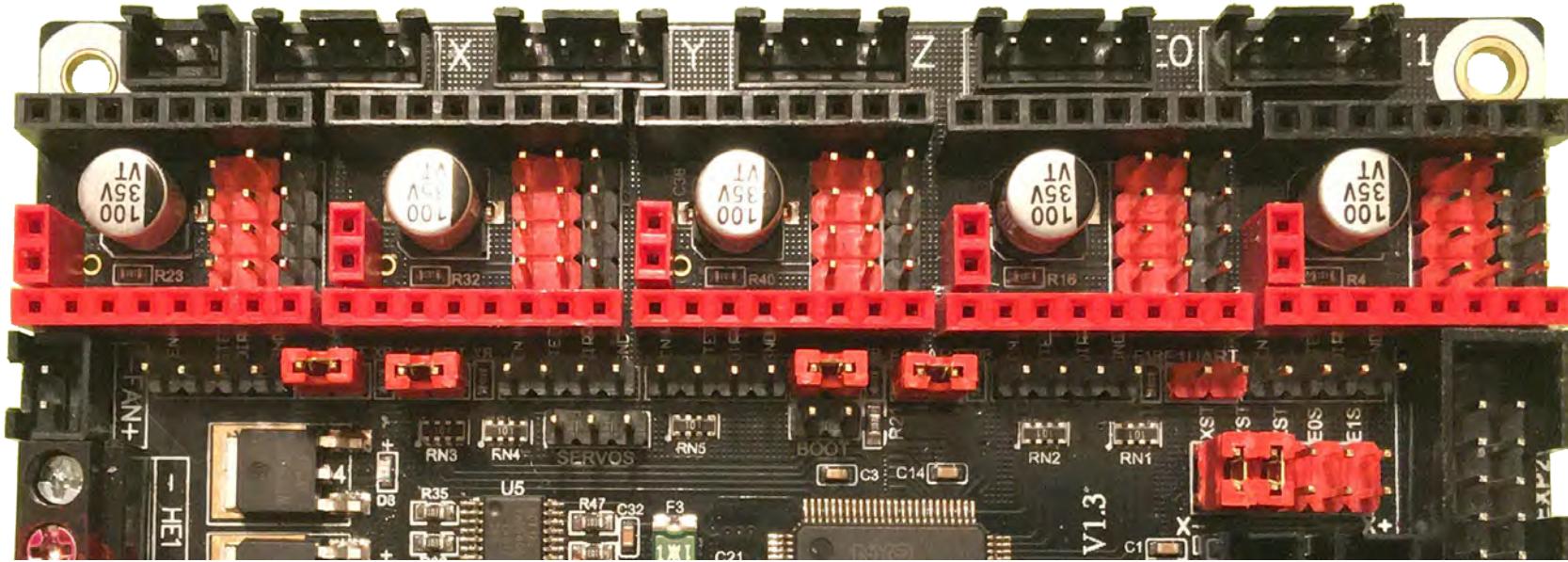
|        |          |   |
|--------|----------|---|
| (XST)  | X-DIAG1  | L |
| (YST)  | Y-DIAG1  | H |
| (ZST)  | Z-DIAG1  | H |
| (E0ST) | E0-DIAG1 | G |
| (E1ST) | E1-DIAG1 | G |

**UART Mode****R - Stall Detection**UART Mode

## Examples of Different UART Configurations Part 1

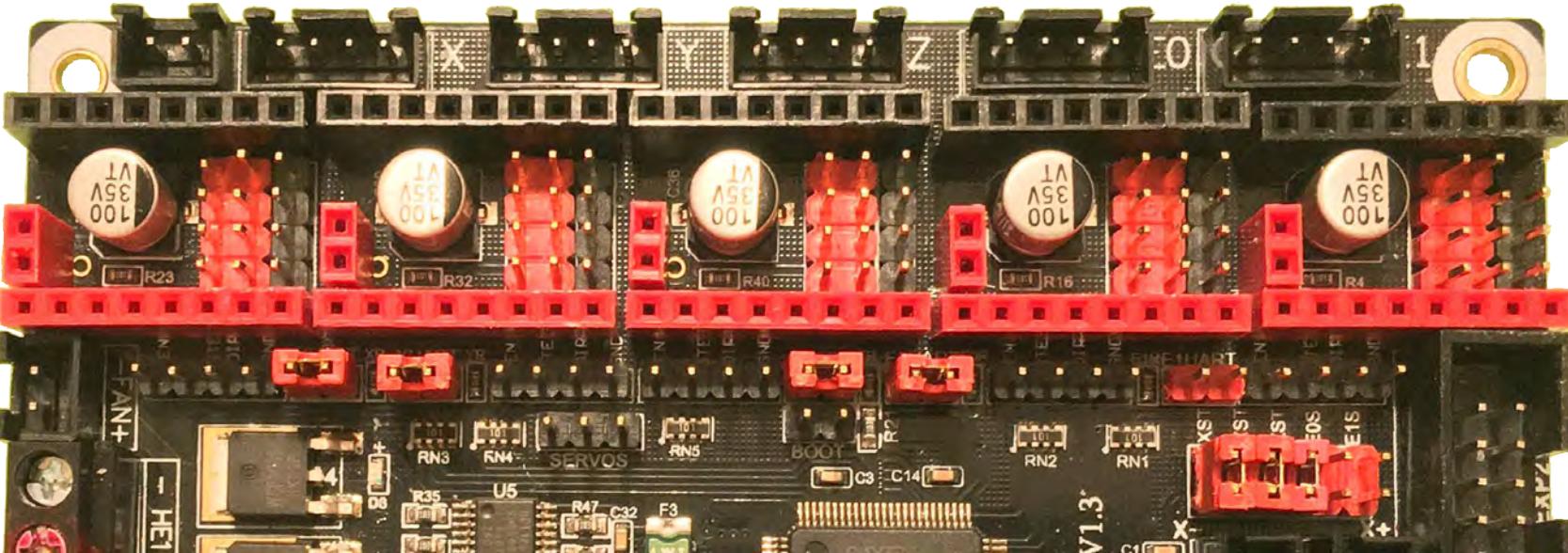
**X, Y, Z and E0  
axes configured  
for UART mode.**

**Sensor-less  
homing for X  
and Y axes.**



**X, Y, Z and E0  
axes configured  
for UART mode.**

**Sensor-less  
homing for X,  
Y and Z axes.**



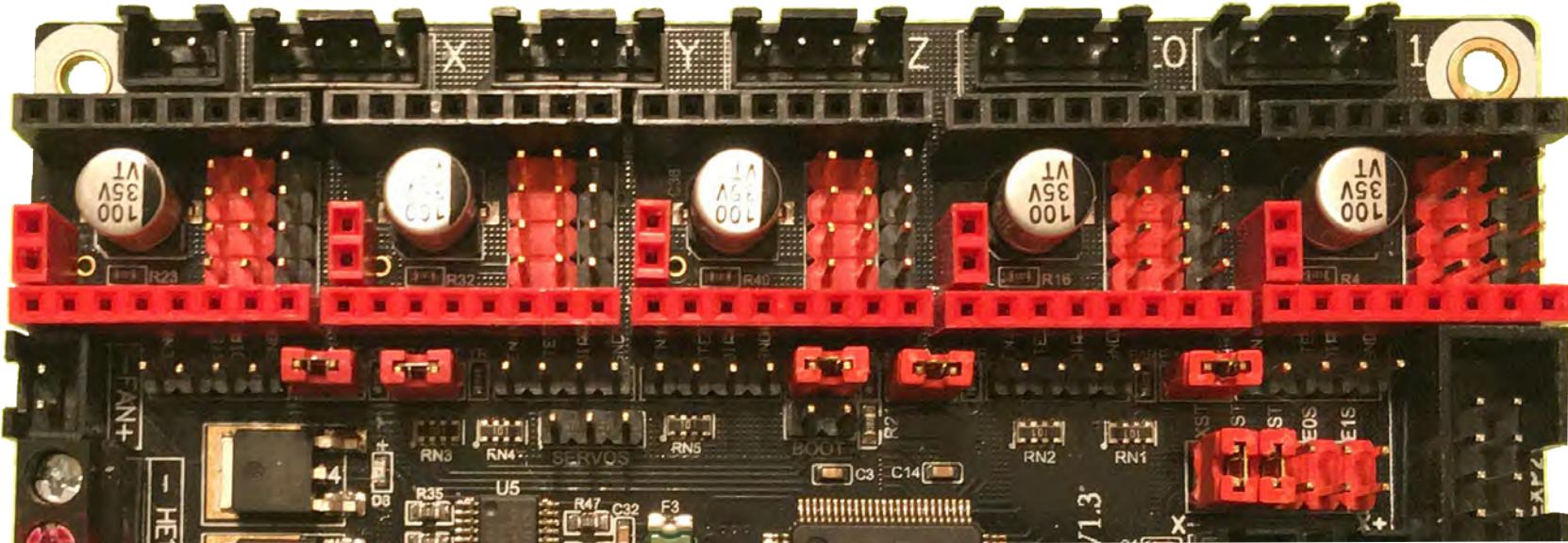
**UART Mode****R** - Stall Detection

X, Y, Z, E0, and  
E1 axes  
configured for  
UART mode.

Sensor-less  
homing for X  
and Y axes.

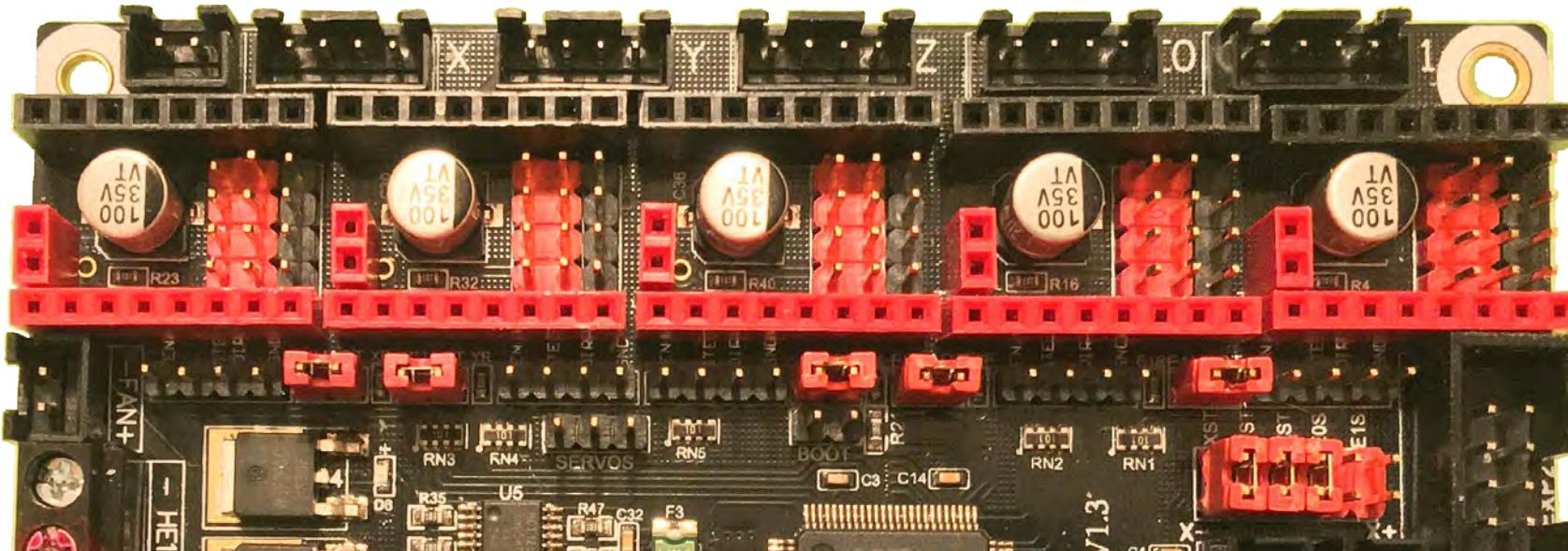
UART Mode

## Examples of Different UART Configurations Part 2



X, Y, Z, E0 and  
E1 axes  
configured for  
UART mode.

Sensor-less  
homing for X,  
Y and Z axes.



## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC2209 V1.2 stepper motor drivers in UART mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC2209 drivers in UART mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC2209 drivers in UART mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
661 /**
662 * Stepper Drivers
663 *
664 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
665 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
666 *
667 * A4988 is assumed for unspecified drivers.
668 *
669 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
670 * TB6560, TB6600, TMC2100,
671 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
672 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
673 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
674 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
675 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2208', 'TMC26X', 'TMC5130']
676 */
677
#define X_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
#define Y_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
#define Z_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define Z3_DRIVER_TYPE A4988
#define Z4_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2209 //GADGETANGEL was commented out and had the value A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988
#define E5_DRIVER_TYPE A4988
#define E6_DRIVER_TYPE A4988
#define E7_DRIVER_TYPE A4988
690 // Enable this feature if all enabled endstop pins are interrupt-capable.
691
692
693
694
695

```

Ln 686, Col 88 Spaces: 2 UTF 8 LF C++ Win32 1

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Since the A4988 driver is what my Ender 3 used, but, now I want to use TMC2209 drivers, I must invert the stepper motor direction because the TMC2209 driver will turn the motors in the opposite direction than the A4988 driver's motor direction. So if the axis' setting you will be using the TMC2209 driver on was "true" change it to "false", as shown in the **GREEN** box below. If the setting was "false", now set it to "true", as shown in the **GREEN** box below

```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > INVERT_Z_DIR
1035 #define DISABLE_Y false
1036 #define DISABLE_Z false
1037
1038 // Warn on display about possibly reduced accuracy
1039 // #define DISABLE_REDUCED_ACCURACY_WARNING
1040
1041 // @section extruder
1042
1043 #define DISABLE_E false // For all extruders
1044 // #define DISABLE_INACTIVE_EXTRUDER // Keep only the active extruder enabled
1045
1046 // @section machine
1047
1048 // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1049 #define INVERT_X_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1050 #define INVERT_Y_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1051 #define INVERT_Z_DIR true // GADGETANGEL was false, stepper motor driver inverts motor direction
1052
1053 // @section extruder
1054
1055 // For direct drive extruder v9 set to true, for geared extruder set to false.
1056 #define INVERT_E0_DIR false // GADGETANGEL was true, stepper motor driver inverts motor direction
1057 #define INVERT_E1_DIR false
1058 #define INVERT_E2_DIR false
1059 #define INVERT_E3_DIR false
1060 #define INVERT_E4_DIR false
1061 #define INVERT_E5_DIR false
1062 #define INVERT_E6_DIR false
1063 #define INVERT_E7_DIR false
1064
1065 // @section homing
1066
1067 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed
1068
1069 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For beds that fall when Z is powered up

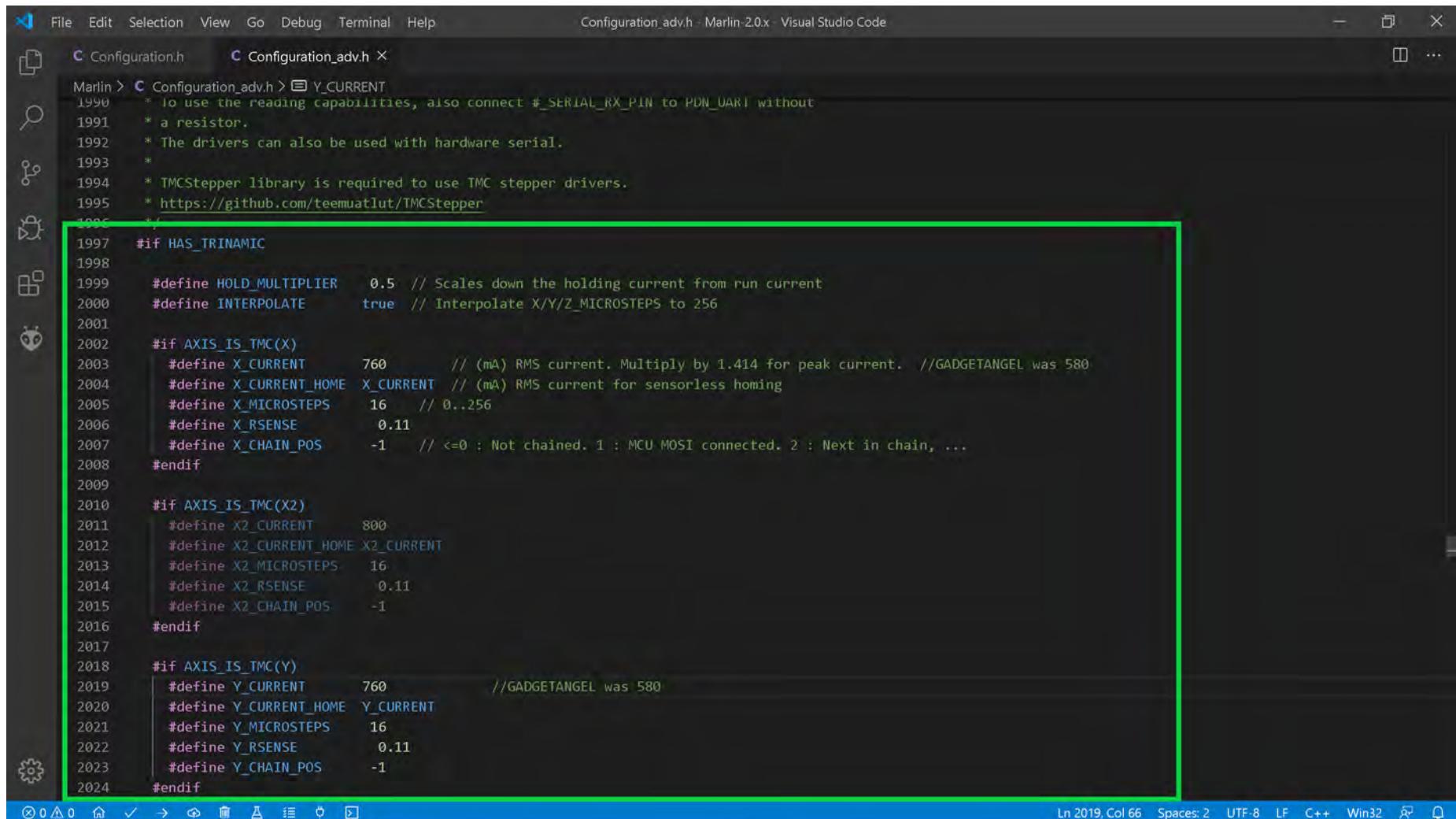
```

Ln 1051, Col 107 Spaces:2 UTF-8 LF C++ Win32

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC2209 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.11. Ensure "Y\_RSENSE" is set to 0.11.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".



```

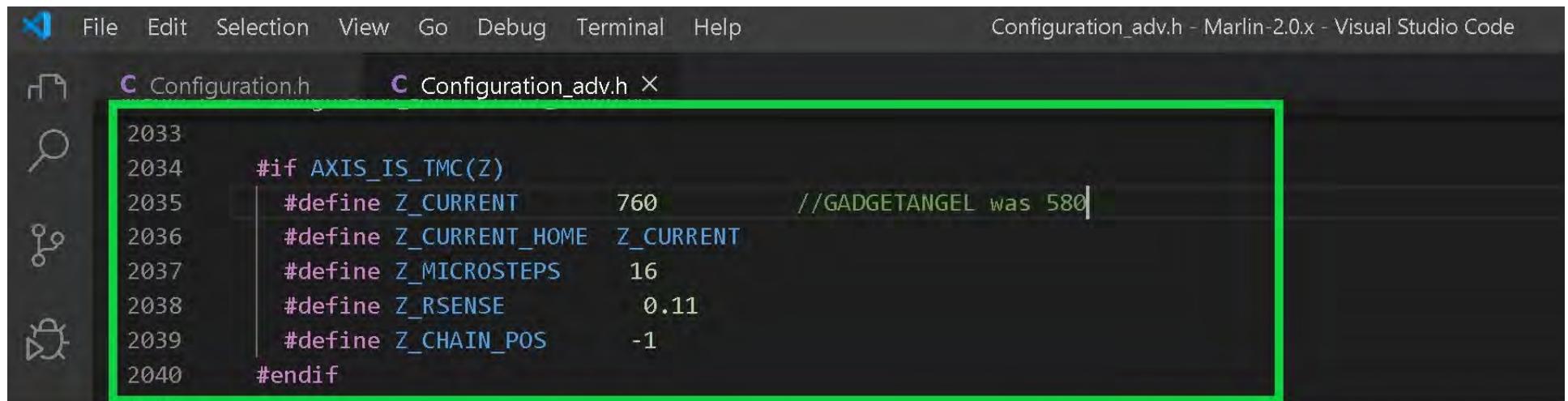
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Y_CURRENT
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOSI connected. 2 : Next in chain, ...
2008#endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016#endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.11
2023 #define Y_CHAIN_POS -1
2024#endif
Ln 2019, Col 66 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.11. Ensure "E0\_RSENSE" is set to 0.11.
- If you do not want to use  $V_{ref}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



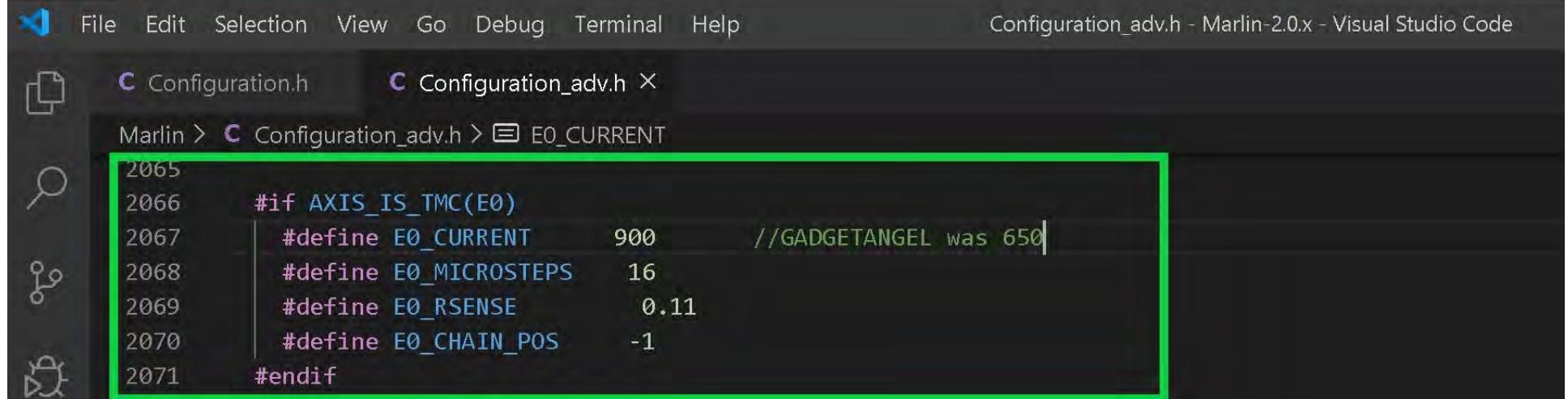
```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

Configuration.h Configuration_adv.h X

2033
2034 #if AXIS_IS_TMC(Z)
2035 #define Z_CURRENT 760 //GADGETANGEL was 580
2036 #define Z_CURRENT_HOME Z_CURRENT
2037 #define Z_MICROSTEPS 16
2038 #define Z_RSENSE 0.11
2039 #define Z_CHAIN_POS -1
2040 #endif

```



```

File Edit Selection View Go Debug Terminal Help
Configuration.h Configuration_adv.h X

Marlin > Configuration_adv.h > E0_CURRENT

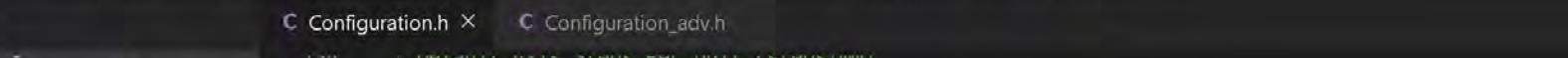
2065
2066 #if AXIS_IS_TMC(E0)
2067 #define E0_CURRENT 900 //GADGETANGEL was 650
2068 #define E0_MICROSTEPS 16
2069 #define E0_RSENSE 0.11
2070 #define E0_CHAIN_POS -1
2071 #endif

```

- Go to the next page.

[The \(latest release of\) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode](#)

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes

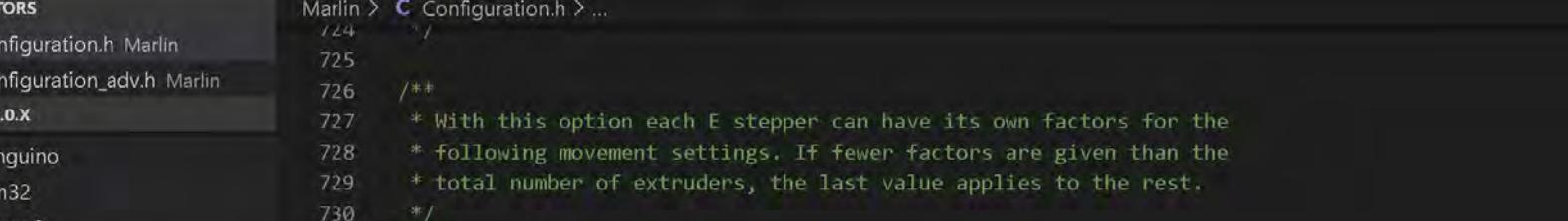


The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h Marlin 2.0.X - Visual Studio Code
- Explorer View:** Shows the project structure under MARLIN-2.0.X:
  - > sanguino
  - > stm32
  - > teensy2
  - > teensy3
  - C pins.h
- Code Editor:** The Configuration.h file is open. A green box highlights the following code block:

```
734 * DEFAULT_AXIS_STEPS_PER_UNIT (steps/mm)
735 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
739 // { 80, 80, 400, 93 } for A4988 on Ender 3
740
741
742 /**
```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT"** anytime our STEPS are NOT 1/16. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```
Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

EXPLORER Configuration.h X Configuration_adv.h

OPEN EDITORS Marlin > Configuration.h > ...
 Configuration.h Marlin
 Configuration_adv.h Marlin

MARLIN-2.0.X
 sanguino
 stm32
 teensy2
 teensy3
 pins.h
 pinsDebug_list.h
 pinsDebug.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration_adv.h

124 /*
125
126 /**
127 * With this option each E stepper can have its own factors for the
128 * following movement settings. If fewer factors are given than the
129 * total number of extruders, the last value applies to the rest.
130 */
131 // #define DISTINCT_E_FACTORS
132 /**
133 * Default Axis Steps Per Unit (steps/mm)
134 * Override with M92
135 *
136 * X, Y, Z, E0 [, E1[, E2...]]
137 */
138 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // GADGETANGEL was
139 // {80, 80, 400, 93} for A4988 on Ender 3
140 // Double because we are going
141 // to 1/32 from 1/16
142
143 */


```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want stealthChop enabled so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h X
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
MARLIN-2.0.X
 > sanguino
 > stm32
 > teensy2
 > teensy3
 C pins.h
 C pinsDebug_list.h
 C pinsDebug.h
 C sensitive_pins.h
 > sd
 G MarlinCore.cpp
 C MarlinCore.h
 C _Bootscreen.h
 C _Statusscreen.h
 C Configuration_adv.h
 C Configuration.h

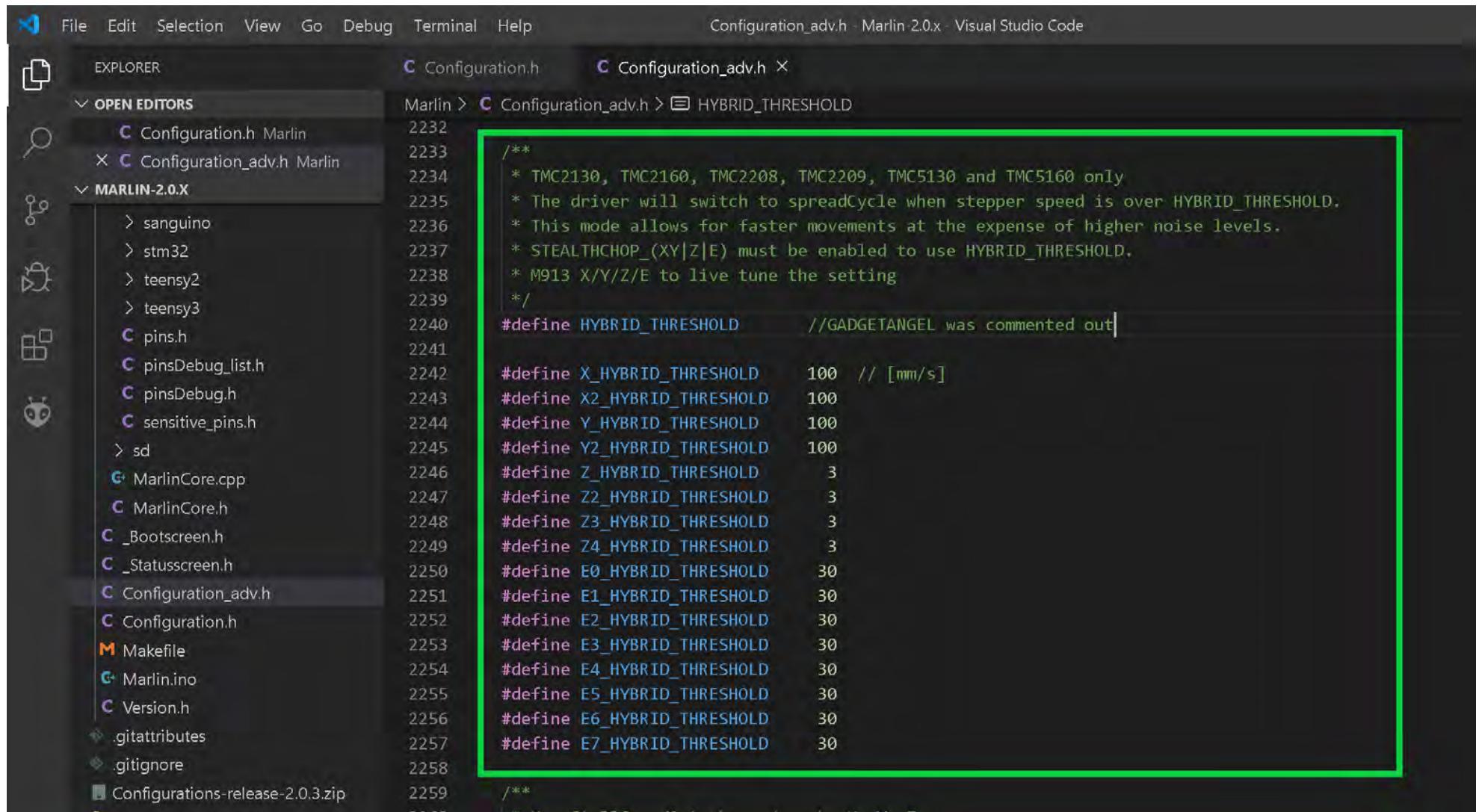
Marlin > C Configuration_adv.h > STEALTHCHOP_XY
2193 /**
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default, it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Explorer:** Shows the project structure under 'OPEN EDITORS' and 'MARLIN-2.0.X' (including files like Configuration.h, Configuration\_adv.h, pins.h, etc.)
- Editor:** Displays the content of Configuration\_adv.h with a green box highlighting the HYBRID\_THRESHOLD section.

```

/*
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //GADGETANGEL was commented out

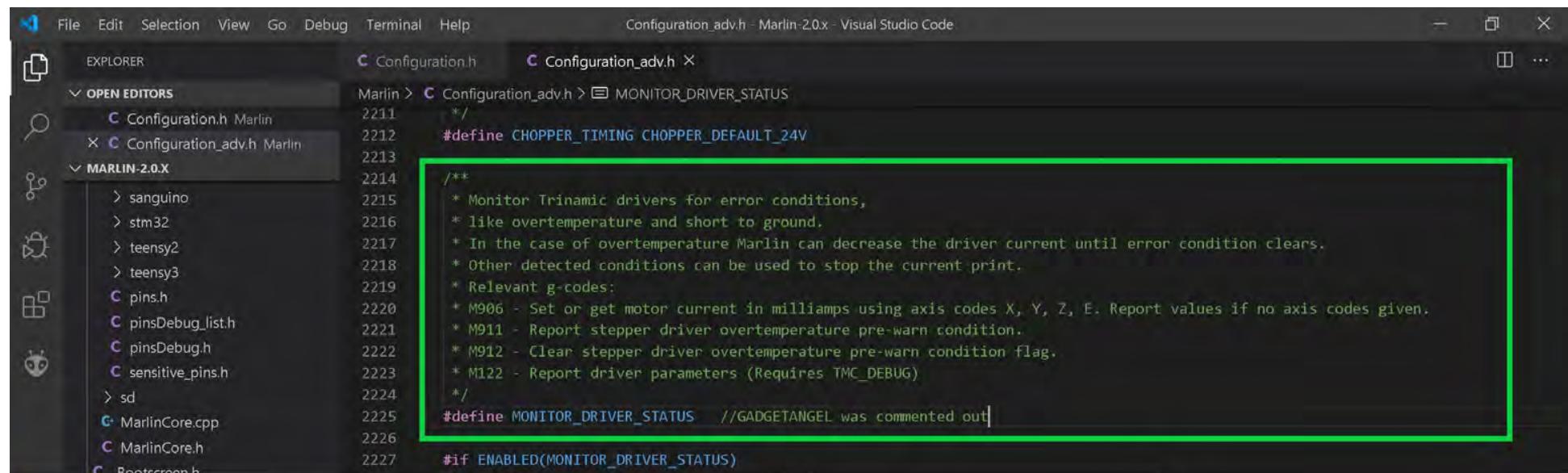
#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line.](#)



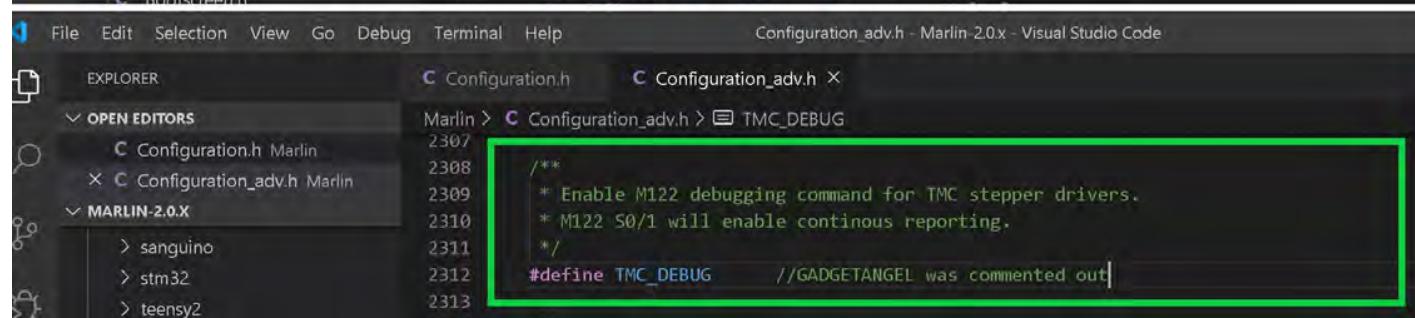
File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > MONITOR_DRIVER_STATUS
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in milliamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```



File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

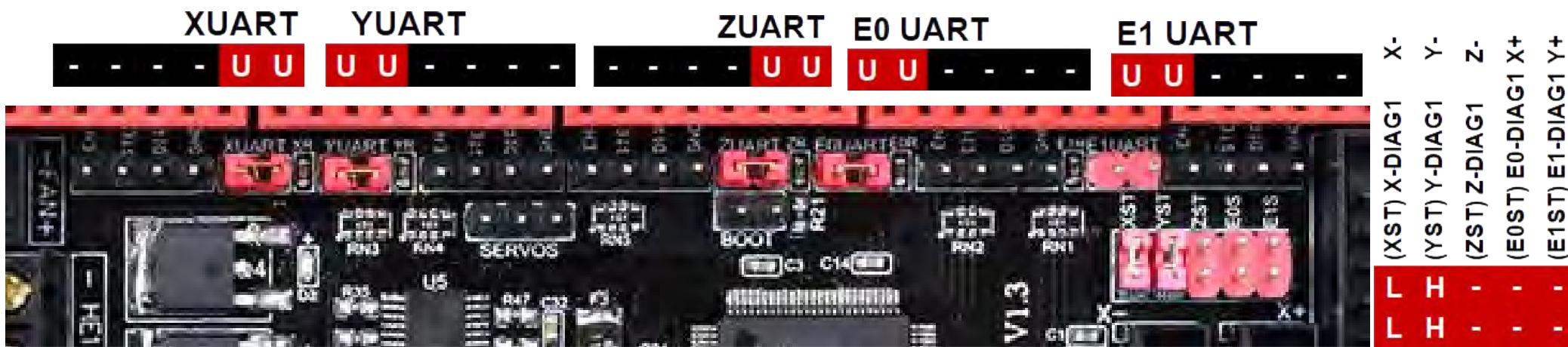
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > TMC_DEBUG
2307
2308 /**
2309 * Enable M122 debugging command for TMC stepper drivers.
2310 * M122 S0/1 will enable continuous reporting.
2311 */
2312 #define TMC_DEBUG //GADGETANGEL was commented out
2313

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- This next section covers sensor-less homing which is available for the TMC2209 in UART mode. I want to enable it so I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC2209 driver has to be plugged into the SKR V1.3 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC2209 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC2209 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC2209 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the following jumpers are in place: are the "L" and "H" jumpers to enable sensor-less homing for X and Y axes. These two jumper will connect my X DIAG1 pin and Y DIAG1 pin to their respective endstops. I will set the UART jumper for X, Y, Z and E0.



- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS\_HOMING"

```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h Marlin 2.0.x Visual Studio Code

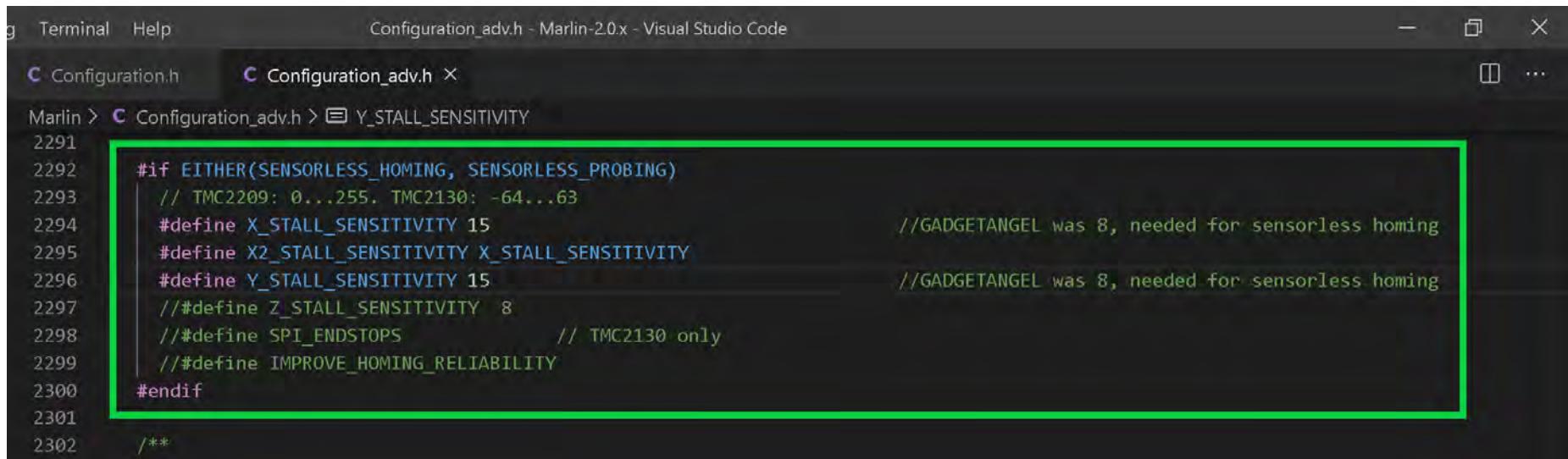
EXPLORER Configuration.h Configuration_adv.h X
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING
2281 */
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out
2283

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Next we set the "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X\_STALL\_SENSITIVITY" and the "Y\_STALL\_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z\_STALL\_SENSITIVITY".

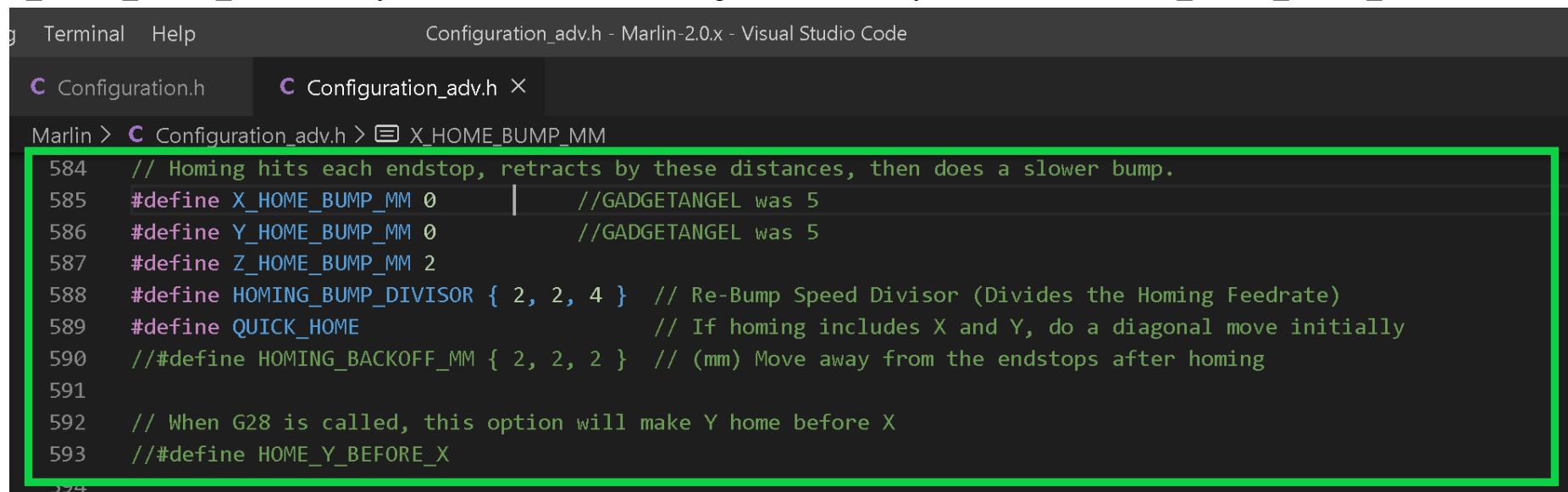


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X\_HOME\_BUMP\_MM" to 0 and "Y\_HOME\_BUMP\_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z\_HOME\_BUMP\_MM" to 0.



```

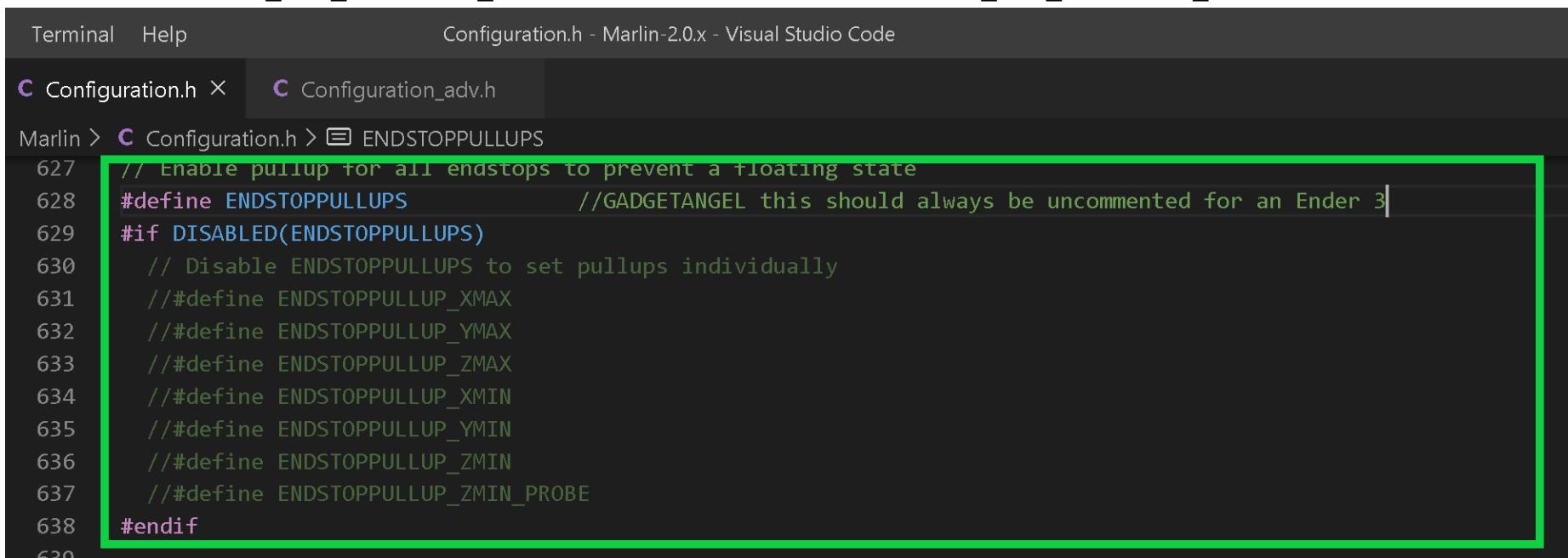
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL was 5
587 #define Z_HOME_BUMP_MM 2
588 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
589 #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
590 //">#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
591
592 // When G28 is called, this option will make Y home before X
593 //">#define HOME_Y_BEFORE_X
594

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default. I also want to check to see how our MIN\_ENDSTOP\_INVERTINGs are set the right way. For an Ender 3 using TMC2209 drivers the "X\_MIN\_ENDSTOP\_INVERTING" should be false, the "Y\_MIN\_ENDSTOP\_INVERTING" should be false, and the "Z\_MIN\_ENDSTOP\_INVERTING" should be false.

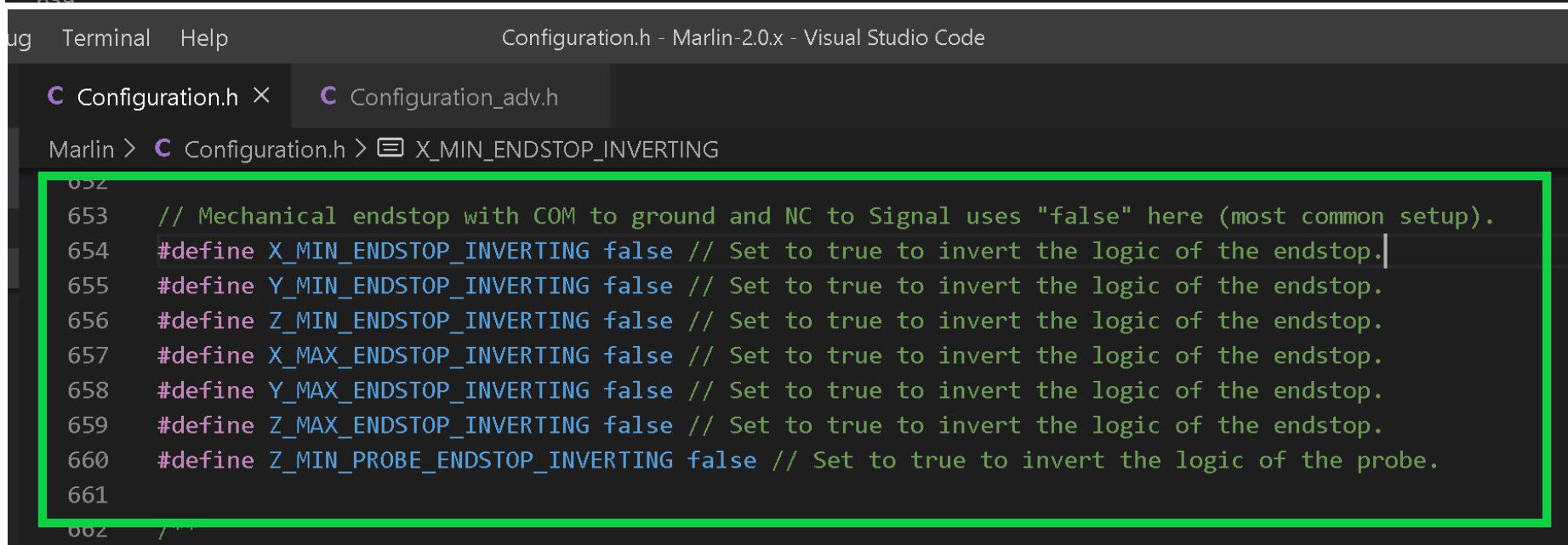


```

Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > ENDSTOPPULLUPS
627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 //#define ENDSTOPPULLUP_XMAX
632 //#define ENDSTOPPULLUP_YMAX
633 //#define ENDSTOPPULLUP_ZMAX
634 //#define ENDSTOPPULLUP_XMIN
635 //#define ENDSTOPPULLUP_YMIN
636 //#define ENDSTOPPULLUP_ZMIN
637 //#define ENDSTOPPULLUP_ZMIN_PROBE
638 #endif
639

```



```

Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h X C Configuration_adv.h

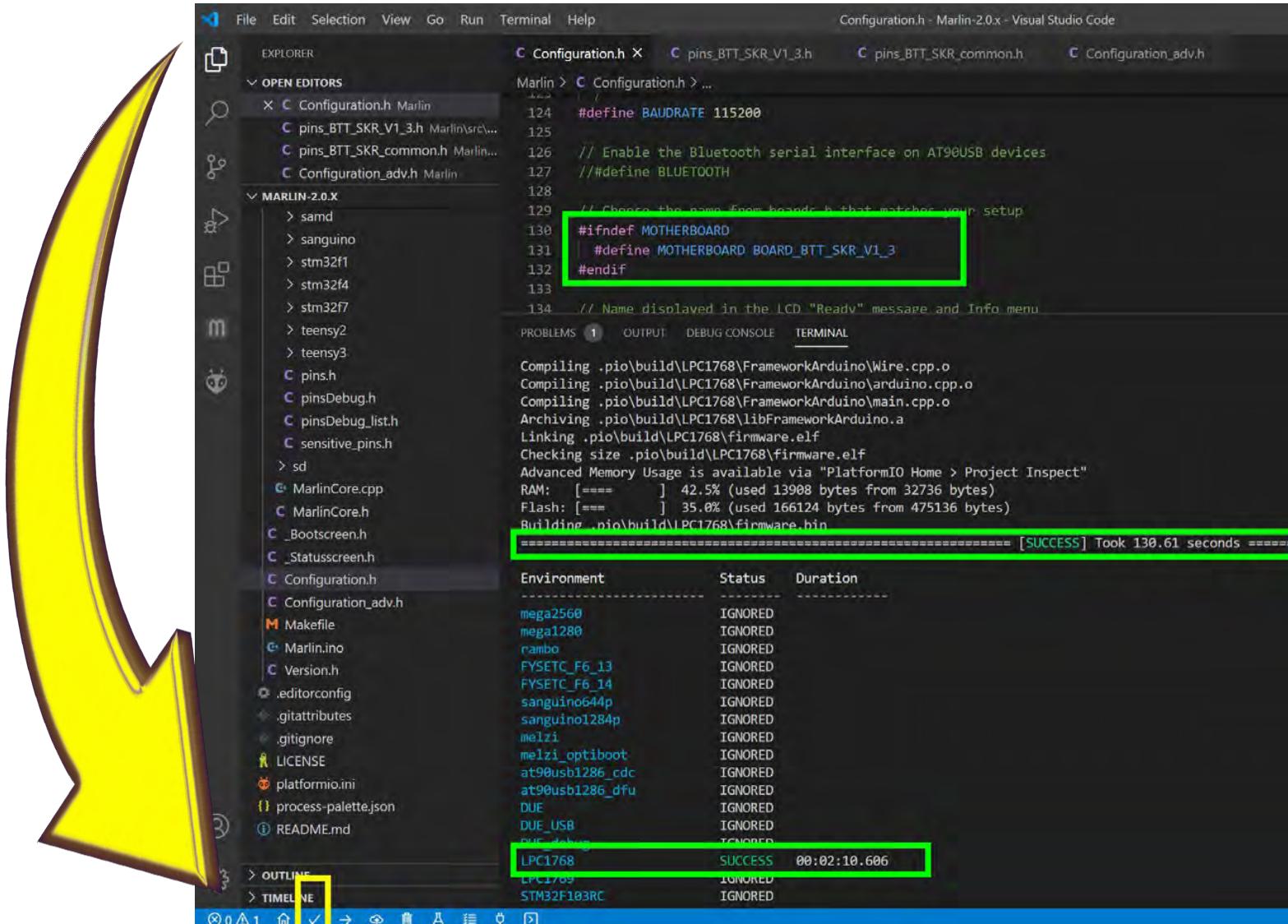
Marlin > C Configuration.h > X_MIN_ENDSTOP_INVERTING
652
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
655 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661
662 /**

```

- Go to the next page.

**The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode**

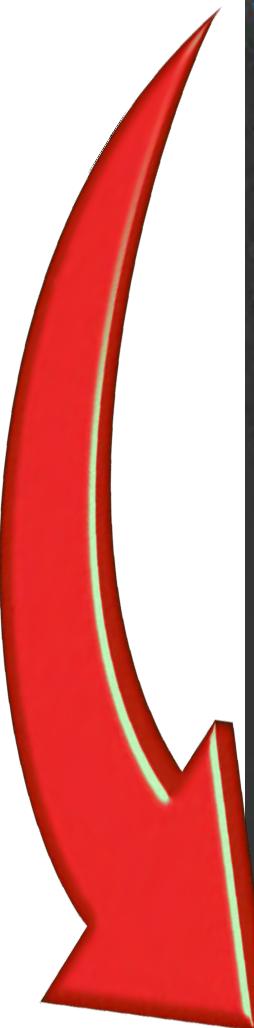
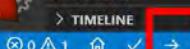
- The end of Marlin setup for BIQU TMC2209 drivers in UART mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.

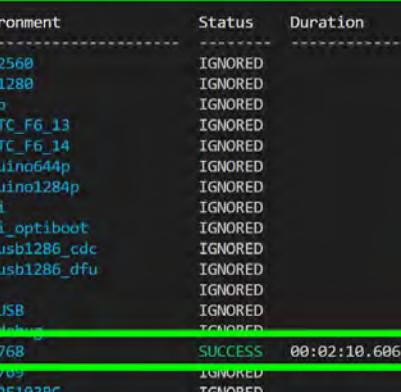


- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC2209 V1.2 Drivers in UART Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

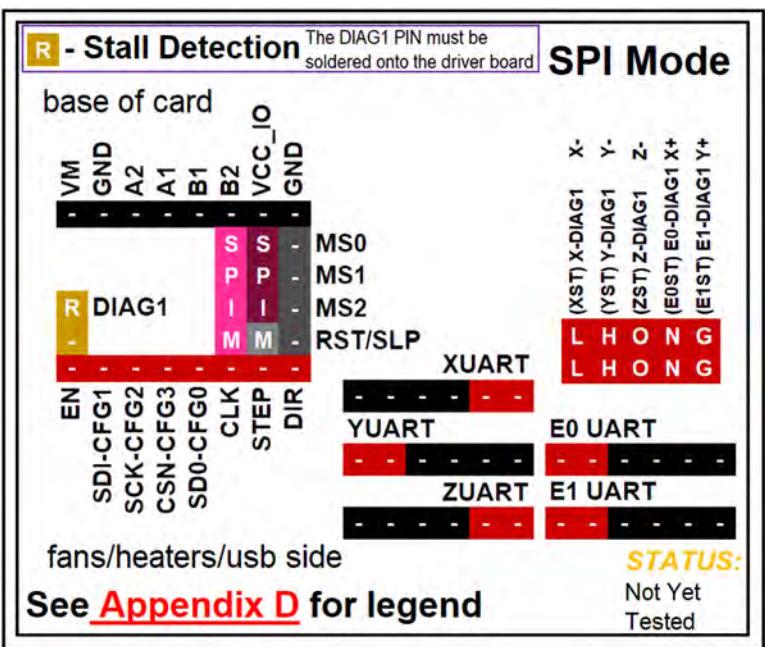
Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o  
 Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o  
 Archiving .pio\build\LPC1768\libFrameworkArduino.a  
 Linking .pio\build\LPC1768\firmware.elf  
 Checking size .pio\build\LPC1768\firmware.elf  
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"  
 RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)  
 Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)  
 Building .pio\build\LPC1768\firmware.bin

[SUCCESS] Took 130.61 seconds

| Environment     | Status         | Duration            |
|-----------------|----------------|---------------------|
| mega2560        | IGNORED        |                     |
| mega1280        | IGNORED        |                     |
| rambo           | IGNORED        |                     |
| FYSETC_F6_13    | IGNORED        |                     |
| FYSETC_F6_14    | IGNORED        |                     |
| sanguino644p    | IGNORED        |                     |
| sanguino1284p   | IGNORED        |                     |
| melzi           | IGNORED        |                     |
| melzi_optiboot  | IGNORED        |                     |
| at90usb1286_cdc | IGNORED        |                     |
| at90usb1286_dfu | IGNORED        |                     |
| DUE             | IGNORED        |                     |
| DUE_USB         | IGNORED        |                     |
| SUE_L103        | IGNORED        |                     |
| <b>LPC1768</b>  | <b>SUCCESS</b> | <b>00:02:10.606</b> |
| LPC1709         | IGNORED        |                     |
| STM32F103RC     | IGNORED        |                     |

> OUTLINE > TIMELINE

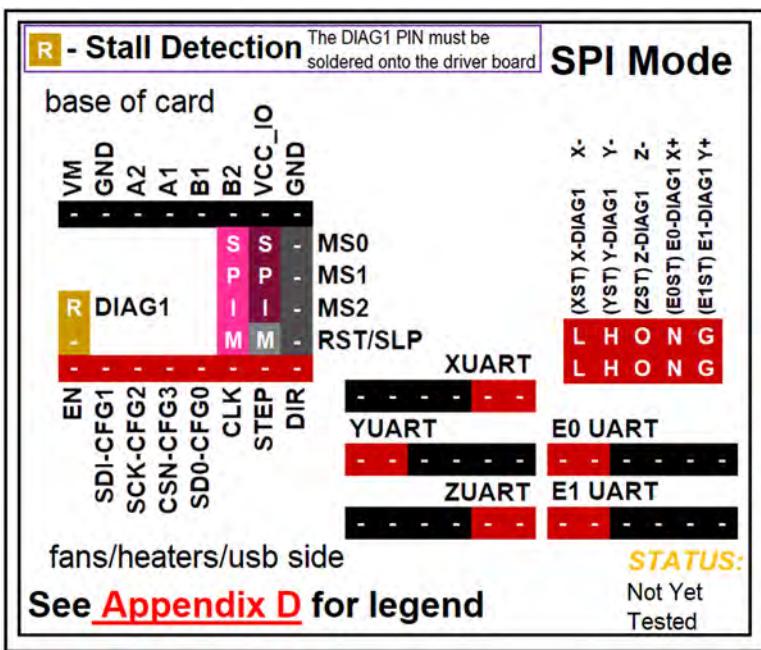
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC5160 V1.2**SPI Mode

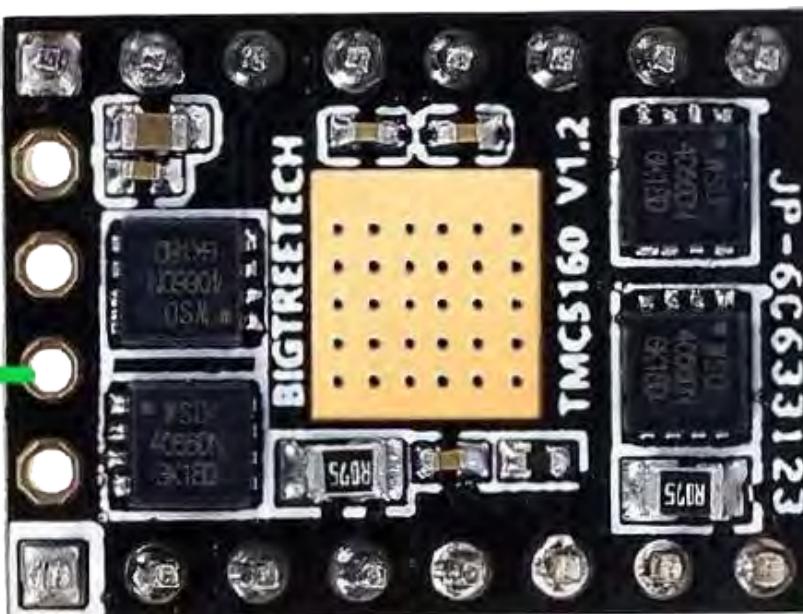
**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

| Driver Chip                                                                              | Steps are set inside<br>of your Firmware                                                                     |                                                                                             |
|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>BIQU®<br/>TMC5160</b><br>SPI Mode<br>Maximum 256 Subdivision<br>40V DC<br>4.3A (peak) |                                                                                                              |                                                                                             |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.075 Ω   | $I_{MAX} = 4.333$<br>See Appendix B #8. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90 = 3.900$ | Current Limit is set by the current sense resistors ( $R_s$ ). Use 50% - 90% of $I_{MAX}$ . |

**BIQU TMC5160 V1.2**SPI Mode

**Note: The TMC5160 V1.2 by default comes in SPI mode.** The BIQU TMC5160 does NOT come with a POT or "V<sub>ref</sub> Test point" location so the I<sub>RMS</sub> 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.

**BIQU TMC5160 V1.2**SPI Mode**SPI Mode**

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

$R_s = R050$  is 0.05 Ohms

$R_s = R062$  is 0.062 Ohms

$R_s = R068$  is 0.068 Ohms

$R_s = R075$  is 0.075 Ohms

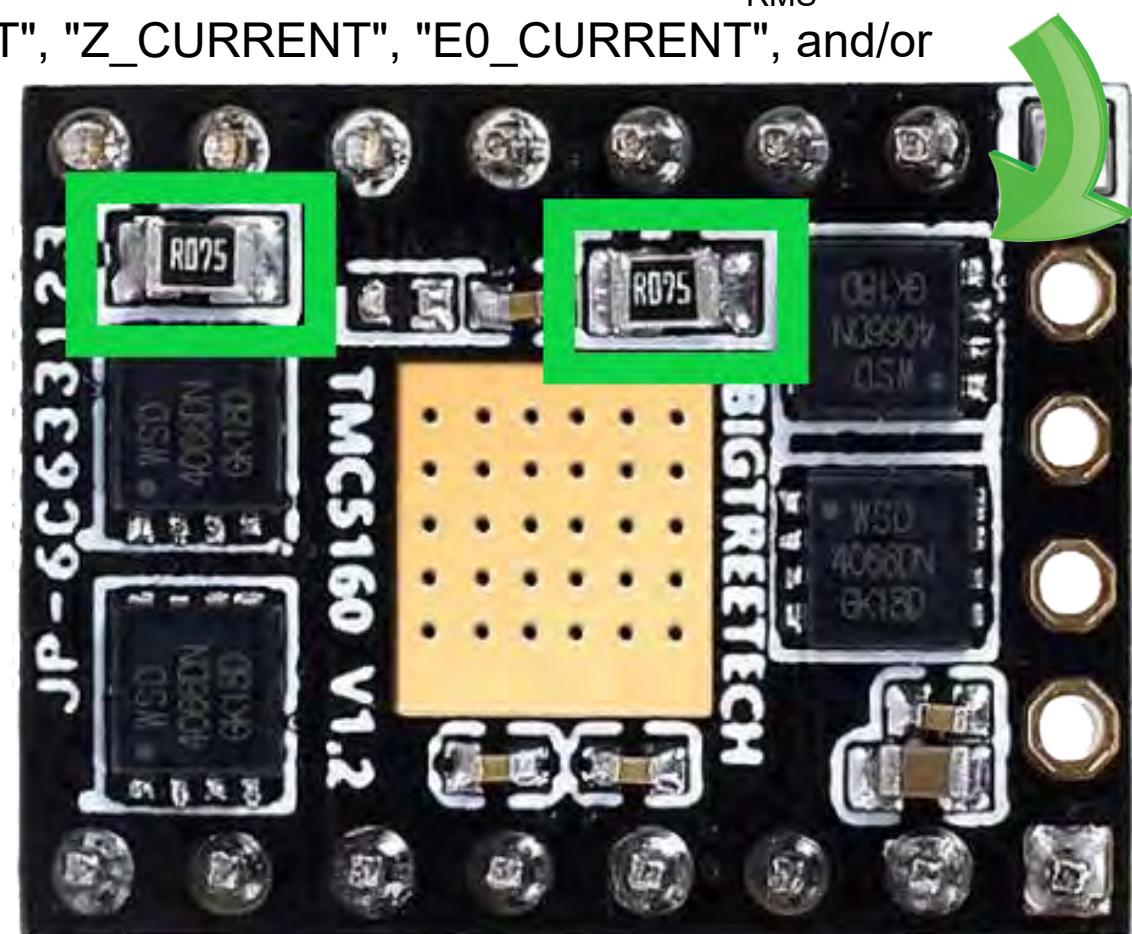
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

$R_s = R150$  is 0.15 Ohms

$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



# SPI Mode

## BIQU TMC5160 V1.2

### SPI Mode

|    |         |    |               |    |               |    |                   |                    |
|----|---------|----|---------------|----|---------------|----|-------------------|--------------------|
| X- | X-DIAG1 | X- | (XST) Y-DIAG1 | Y- | (YST) Z-DIAG1 | Z- | (ZST) E0-DIAG1 X+ | (E0ST) E1-DIAG1 Y+ |
| L  | H       | O  | -             | -  | L             | H  | O                 | G                  |
| L  | H       | O  | -             | -  | L             | H  | O                 | G                  |
|    |         |    |               |    |               |    |                   |                    |

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.



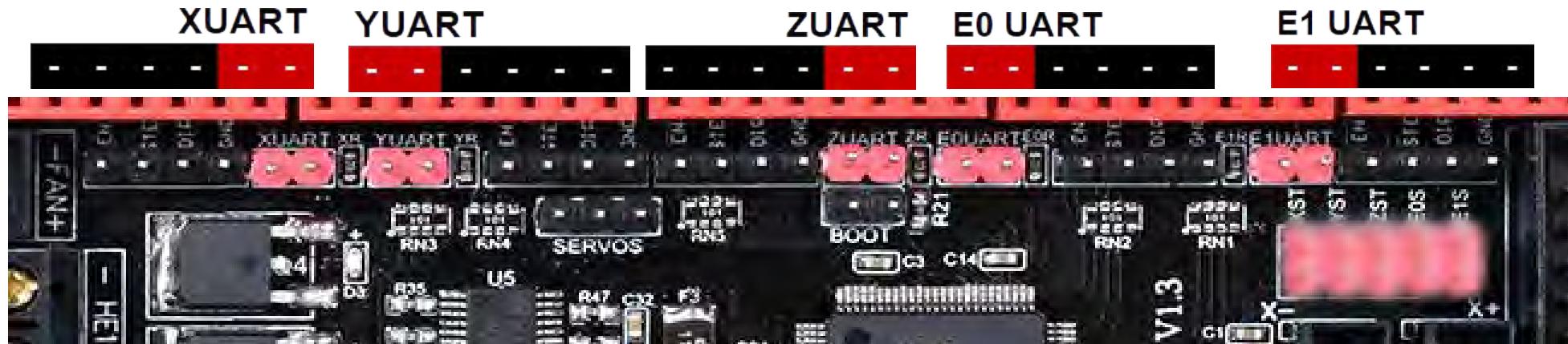
|    |         |    |               |    |               |    |                   |                    |
|----|---------|----|---------------|----|---------------|----|-------------------|--------------------|
| X- | X-DIAG1 | X- | (XST) Y-DIAG1 | Y- | (YST) Z-DIAG1 | Z- | (ZST) E0-DIAG1 X+ | (E0ST) E1-DIAG1 Y+ |
|    |         |    |               |    |               |    |                   |                    |
|    |         |    |               |    |               |    |                   |                    |
|    |         |    |               |    |               |    |                   |                    |
|    |         |    |               |    |               |    |                   |                    |

Note: If sensor-less homing is **not wanted** ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".



**BIQU TMC5160 V1.2**SPI Mode**SPI Mode**

**Note:** Ensure that **ALL** the UART jumpers are **EMPTY**.

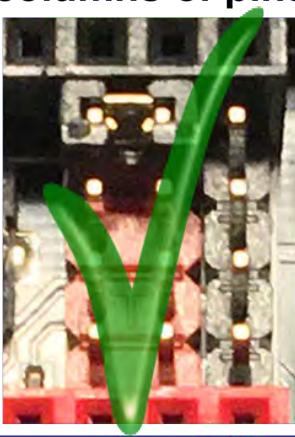


**BIQU TMC5160 V1.2**SPI Mode**SPI Mode****SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode**

Example of the **WRONG** way to set the “**S**” jumper for SPI mode (right two columns of pins):



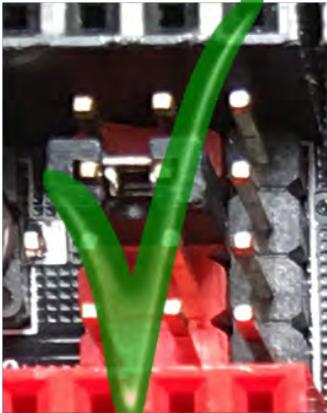
Example of the **RIGHT** way to set the “**S**” jumper for SPI mode (left two columns of pins):



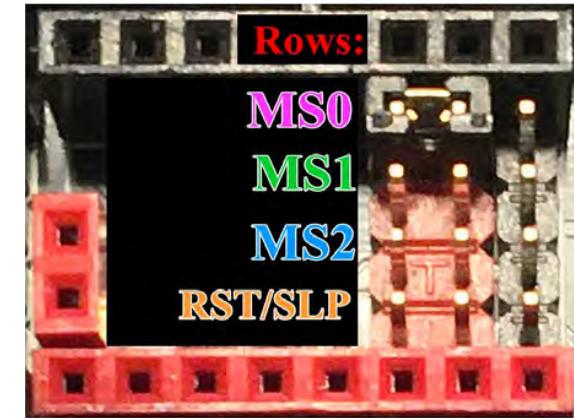
Example of the **WRONG** way to set the “**P**” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “**P**” jumper for SPI mode (left two columns of pins):



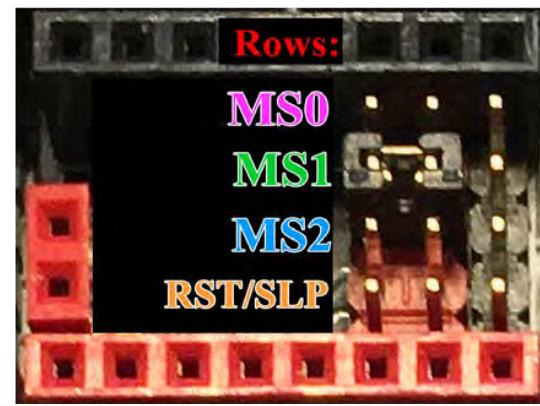
## **S S → Jumper set**



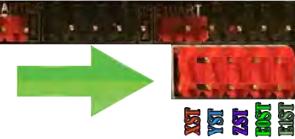
SET ‘**ST**’ JUMPER to enable Sensor-less Homing



## **P P → Jumper set**



SET ‘**ST**’ JUMPER to enable Sensor-less Homing



# SPI Mode

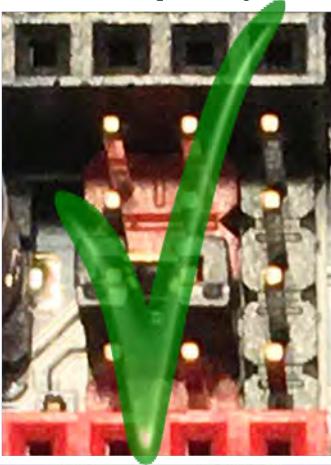
Example of the **WRONG** way to set the “I” jumper for SPI mode (right two columns of pins):



Example of the **WRONG** way to set the “M” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “I” jumper for SPI mode (left two columns of pins):

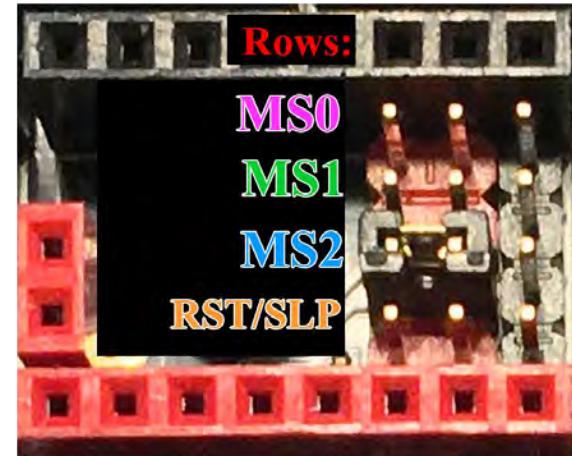


Example of the **RIGHT** way to set the “M” jumper for SPI mode (left two columns of pins):

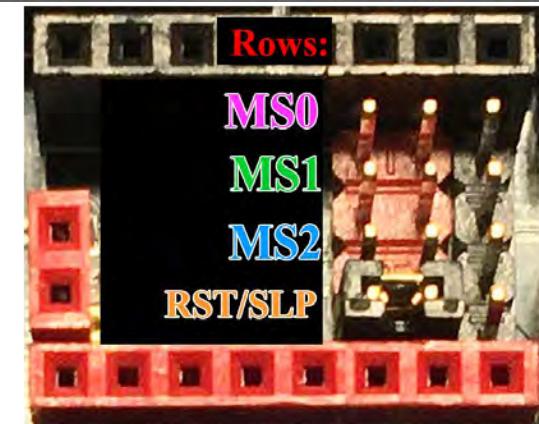


## SPI Mode

I | I → **Jumper set**



M | M → **Jumper set**

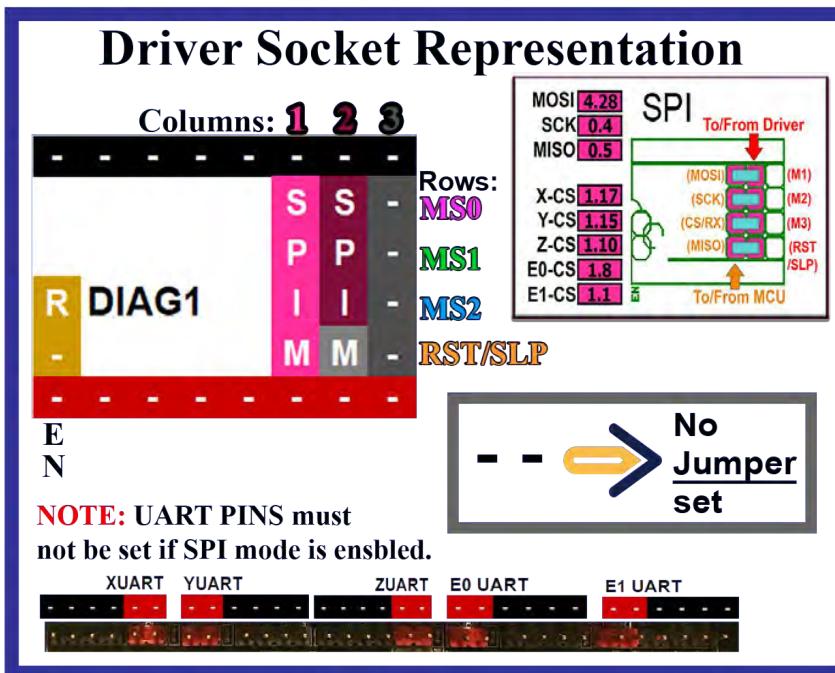


# SPI Mode

## BIQU TMC5160 V1.2

### SPI Mode

## SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode



SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

**S S**  
**P P**  
**I I**  
**M M**

Meaning:

**Jumpers Set**

Driver Socket Representation:

**S S**

Meaning:

**set Jumper between column 1 and column 2 on the MS0 row**

Driver Socket Representation:

**P P**

Meaning:

**set Jumper between column 1 and column 2 on the MS1 row**

Driver Socket Representation:

**I I**

Meaning:

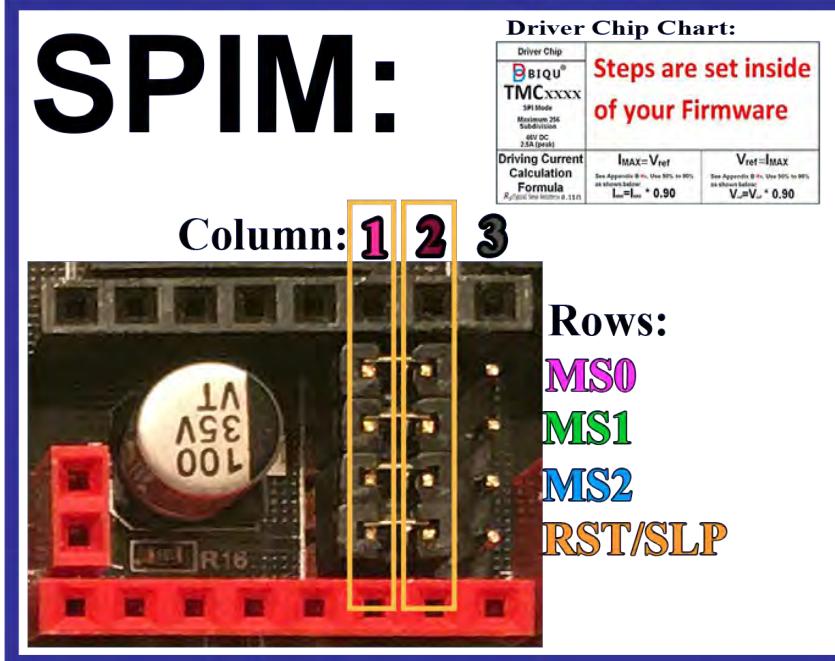
**set Jumper between column 1 and column 2 on the MS2 row**

Driver Socket Representation:

**M M**

Meaning:

**set Jumper between column 1 and column 2 on the RST/SLP row**



# SPI Mode

## BIQU TMC5160 V1.2

### SPI Mode

## Information on Sensor-less Homing

### Driver Socket Representation

SPI Capable:

Columns: 1 2 3

Rows: MS0  
MS1  
MS2  
RST/SLP

R DIAG1  
E N

**NOTE: UART PINS must not be set if SPI mode is enabled.**



SET 'ST' JUMPER to enable Sensor-less Homing:



XUART YUART ZUART E0 UART E1 UART

U U U U U U U

**NOTE: SPI PINS must not be set if UART mode is enabled.**

UART Capable:

Columns: 1 2 3

Rows: MS0  
MS1  
MS2  
RST/SLP

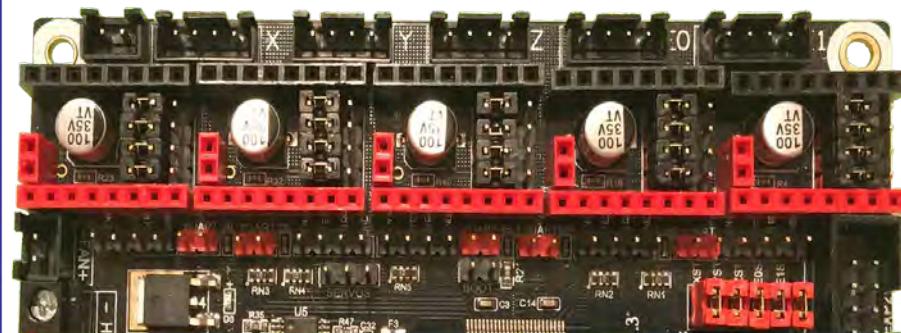
R DIAG E N

### Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are SET for ALL Axes:



### STALLGUARD (Sensor-less Homing)

DIAG PIN ENDSTOP

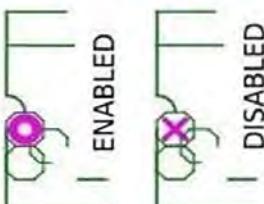
X X-DIAG1 1.29 X-

Y Y-DIAG1 1.27 Y-

Z Z-DIAG1 1.25 Z-

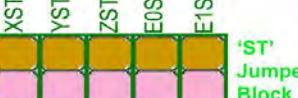
E0 E0-DIAG1 1.28 X+

E1 E1-DIAG1 1.26 Y+



To/From Driver DIAG pin

To/From MCU Endstops

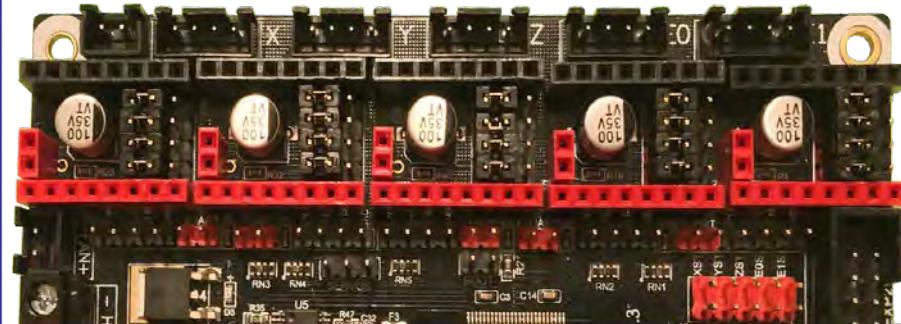


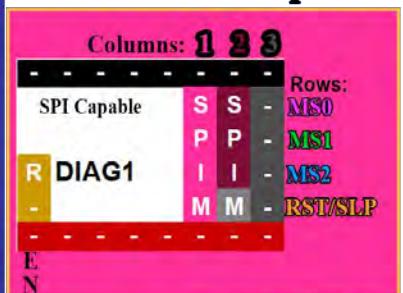
Note Concerning the TMC2209 in UART Mode ONLY:

If using limit switches/endstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

**Note:** For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

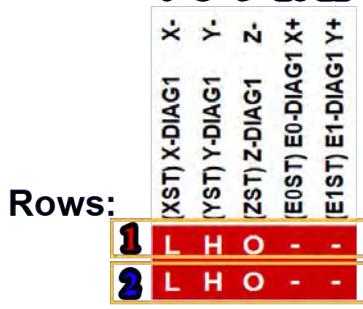
If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are REMOVED for ALL Axes:



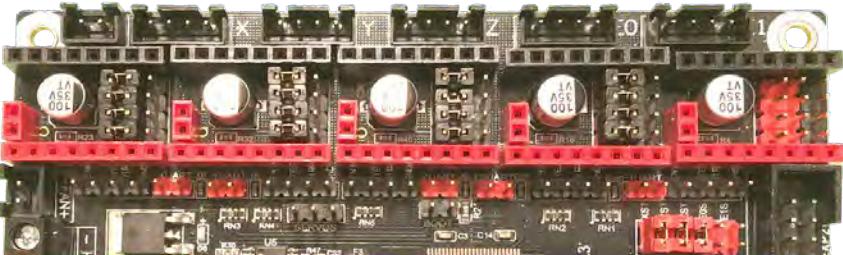
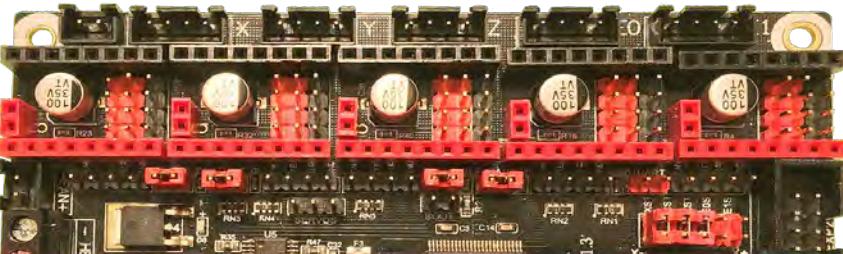
**BIQU TMC5160 V1.2**SPI Mode**Driver Socket Representation of 'ST' JUMPER Block**

Columns: 7 8 9 10 11

Columns: 7 8 9 10 11

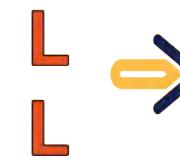


Rows:

**Sensor-less Homing on X, Y, & Z via SPI mode:****Sensor-less Homing on X, Y, & Z via UART mode:****Sensor-less Homing Capable Drivers:****SPI Capable Drivers:** TMC2130, TMC5160 & TMC5161**UART Capable Drivers:** TMC2209

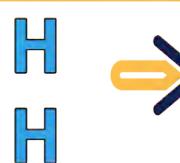
Driver Socket Representation:

Meaning:  
**set Jumper between row 1 and row 2 in column 7**



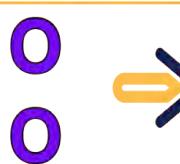
Driver Socket Representation:

Meaning:  
**set Jumper between row 1 and row 2 in column 8**



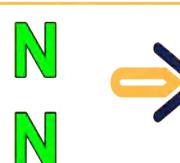
Driver Socket Representation:

Meaning:  
**set Jumper between row 1 and row 2 in column 9**



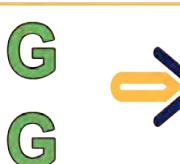
Driver Socket Representation:

Meaning:  
**set Jumper between row 1 and row 2 in column 10**



Driver Socket Representation:

Meaning:  
**set Jumper between row 1 and row 2 in column 11**



**BIQU TMC5160 V1.2**SPI Mode**SPI Mode**

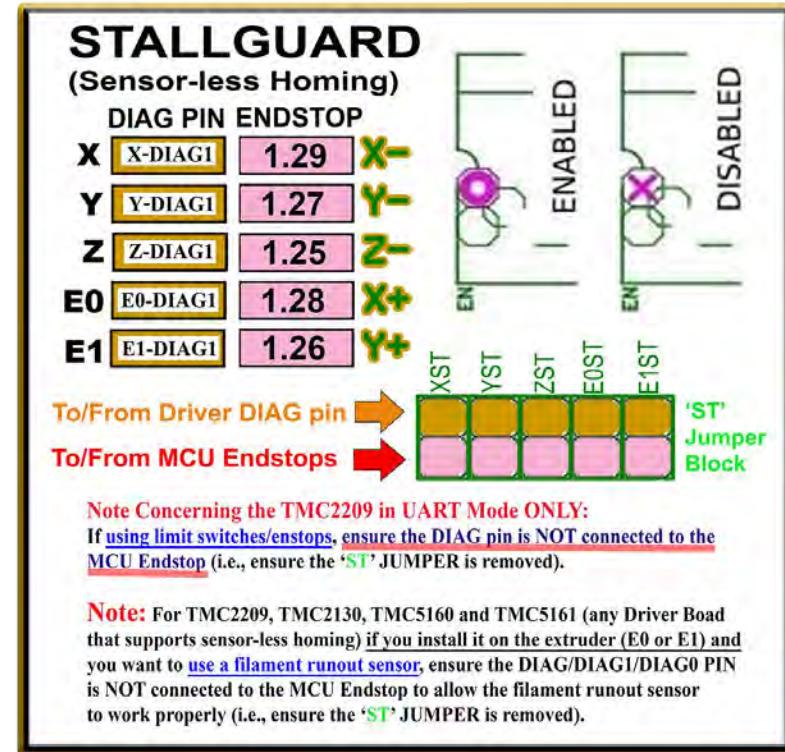
**Use the 'ST' Jumper Block**, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capability of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

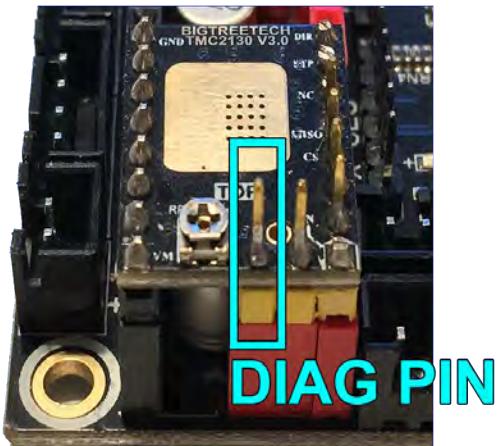
- Extruder Axis (E0 or E1)
- Z Axis, if a BLTouch is used to Home instead of a physical endstop

# SPI Mode

## BIQU TMC5160 V1.2

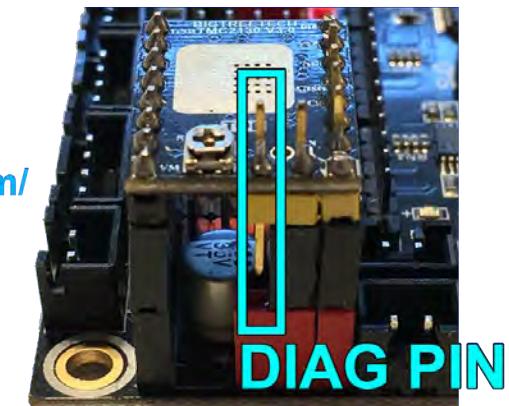
### SPI Mode

To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board** **AND** the '**ST'** **JUMPER** is set in the '**'ST' JUMPER Block** for the **Axis**.



To **disable** sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the **SKR V1.3 Board** **OR** remove the '**ST**' **JUMPER** from the '**'ST' JUMPER Block**.

Link to stackable header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>



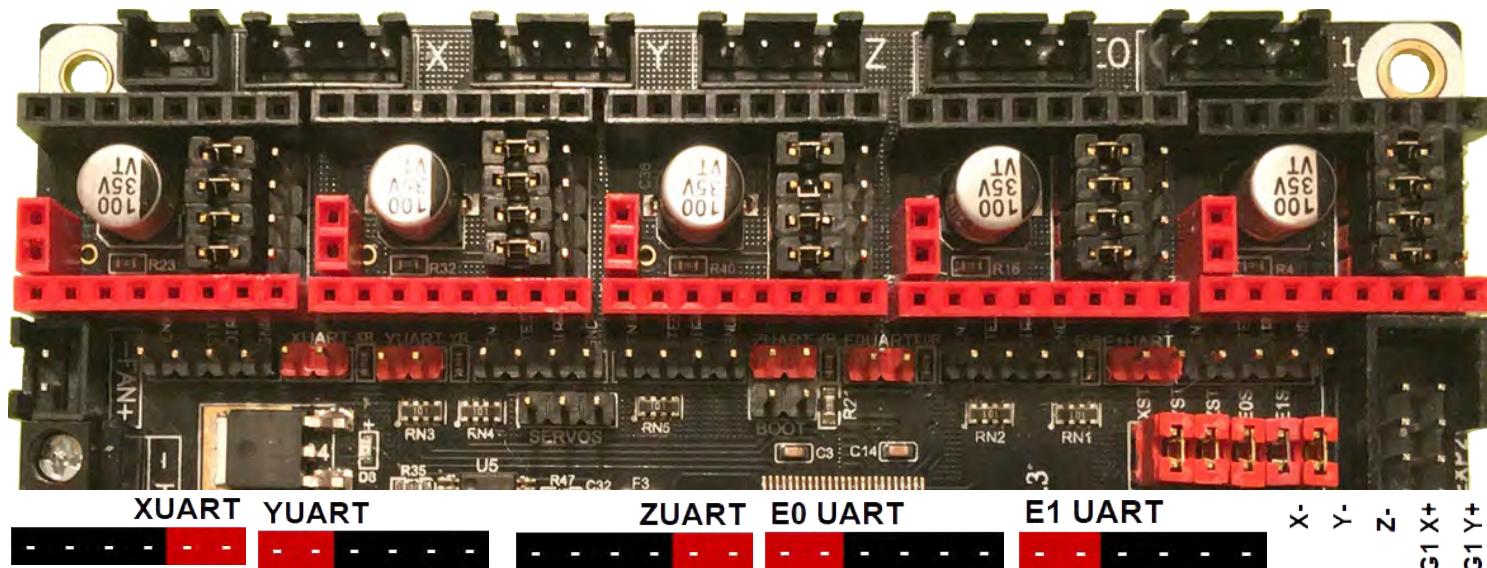
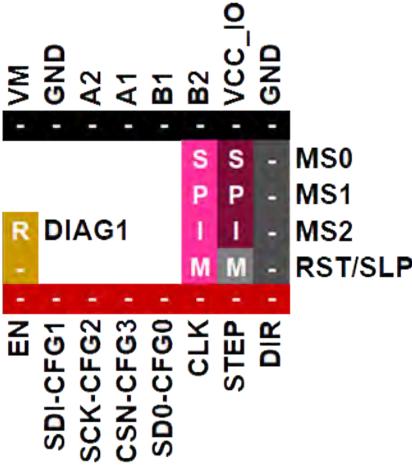
# SPI Mode

## BIQU TMC5160 V1.2

### SPI Mode

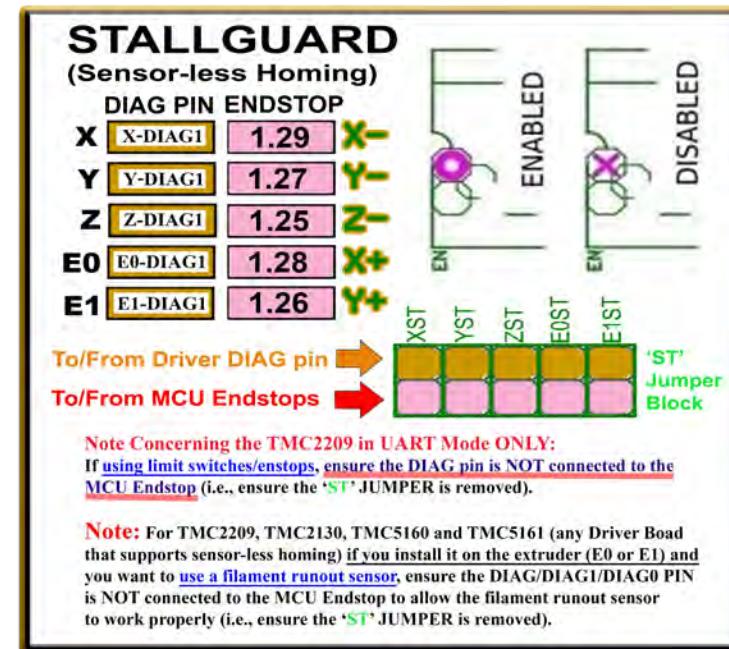
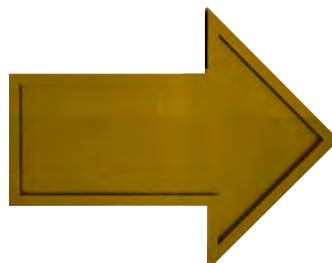
**Note:** Set JUMPERS "S", "P", "I", and "M" on the board!!

SPI



See [Appendix D](#) for legend

**Note:** TMC5160 has sensor-less homing capability



|        |          |    |
|--------|----------|----|
| (XST)  | X-DIAG1  | X- |
| (YST)  | Y-DIAG1  | Y- |
| (ZST)  | Z-DIAG1  | Z- |
| (E0ST) | E0-DIAG1 | X+ |
| (E1ST) | E1-DIAG1 | Y+ |

|   |   |   |   |   |
|---|---|---|---|---|
| L | H | O | N | G |
| L | H | O | N | G |

# SPI Mode

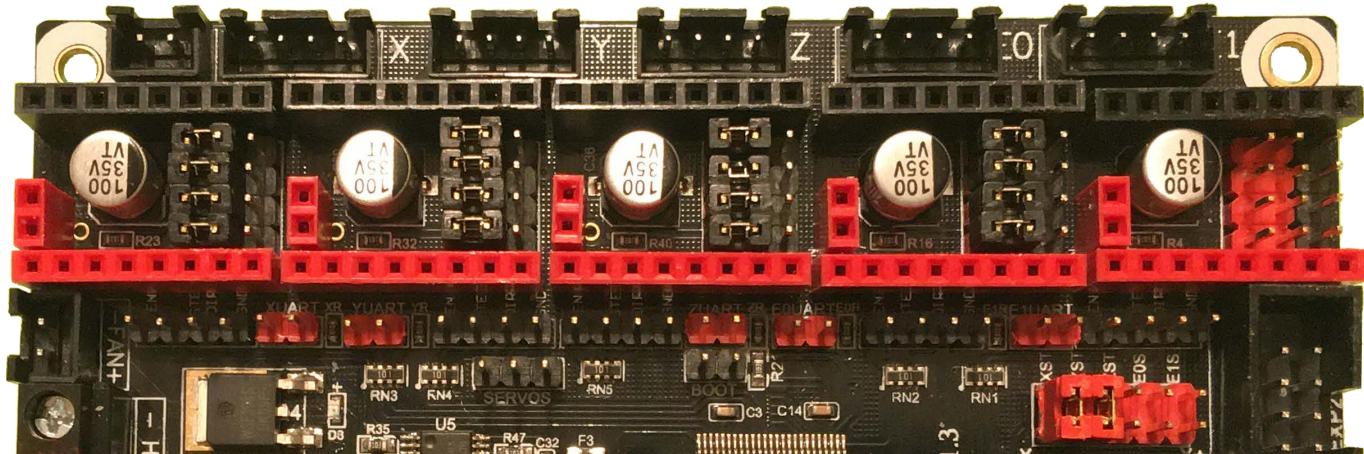
## BIQU TMC5160 V1.2

### SPI Mode

## Examples of Different SPI Configurations Part 1

X, Y, Z and E0 axes  
configured for SPI mode.

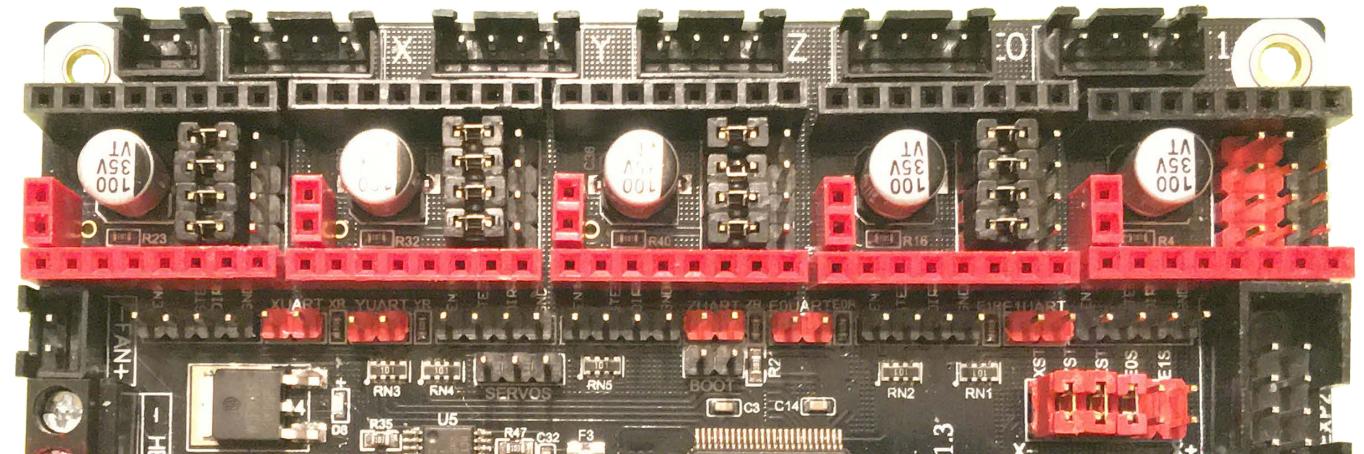
Sensor-less homing for X  
and Y axes.



---

X, Y, Z and E0 axes  
configured for SPI mode.

Sensor-less homing for X,  
Y and Z axes.



# SPI Mode

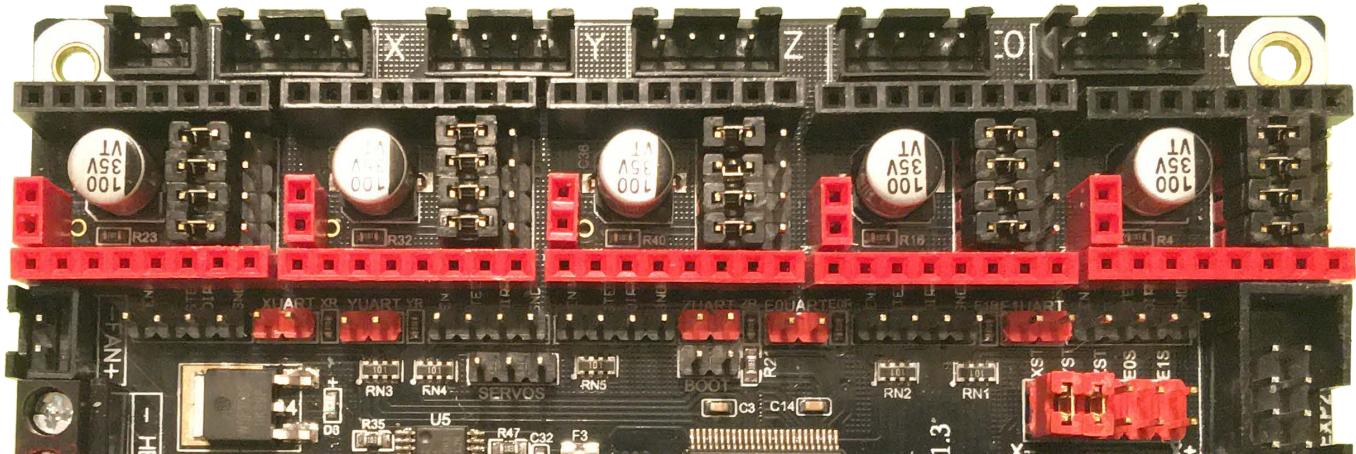
## BIQU TMC5160 V1.2

### SPI Mode

## Examples of Different SPI Configurations Part 2

X, Y, Z, E0, and E1 axes  
configured for SPI mode.

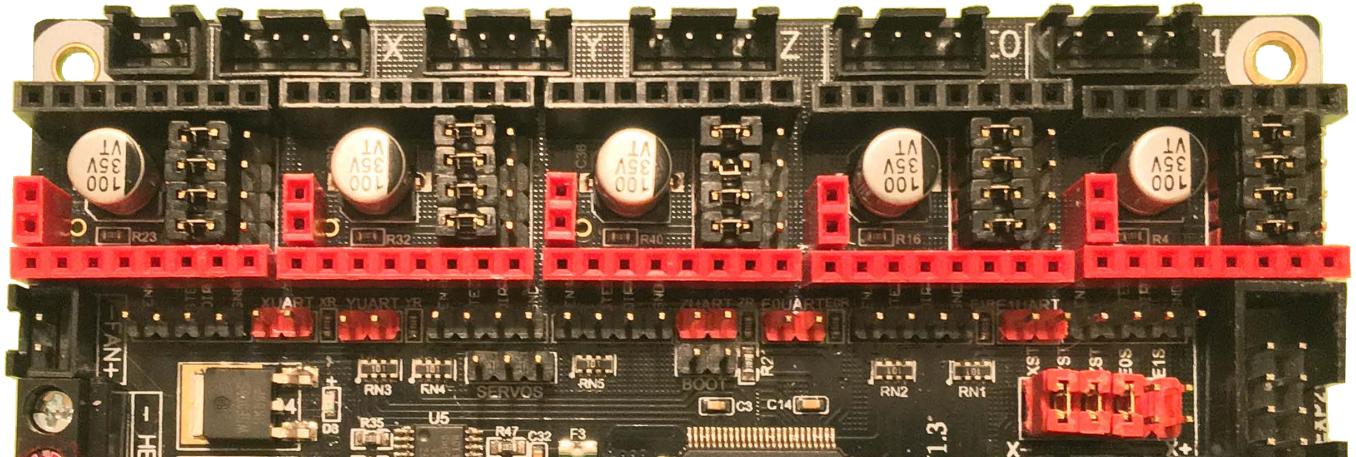
Sensor-less homing for X  
and Y axes.



---

X, Y, Z, E0 and E1 axes  
configured for SPI mode.

Sensor-less homing for X,  
Y and Z axes.



## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

**NOTE:** [Go to Appendix C](#), and then come back here for the changes to Marlin for BIQU TMC5160 V1.2 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC5160 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5160 drivers in SPI mode. When two "/" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").

```

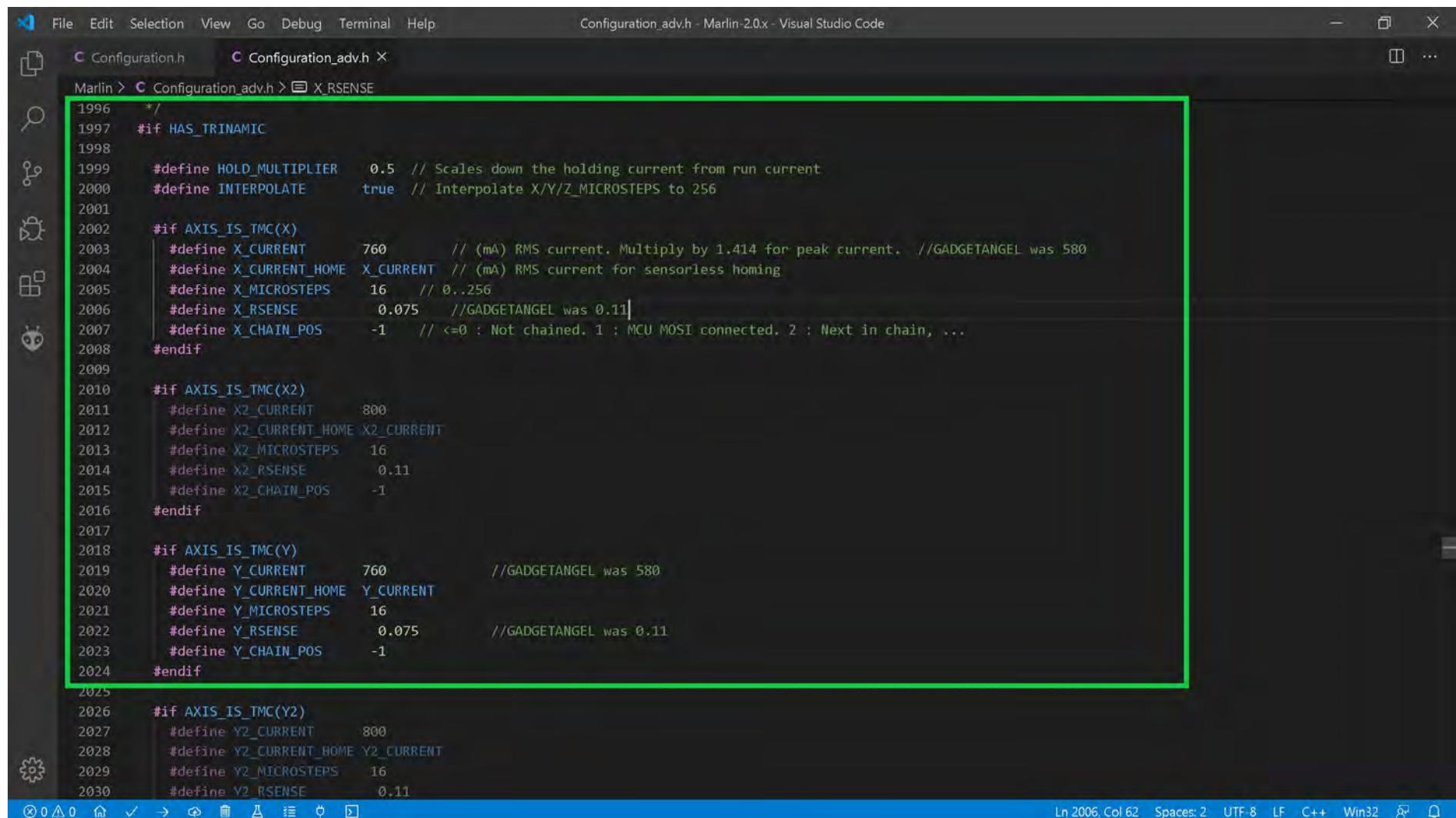
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER PIO Home Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > E0_DRIVER_TYPE
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
> OUTLINE
Ln 686, Col 1
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'T
677 "
678 #define X_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC5160 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.075. Ensure "Y\_RSENSE" is set to 0.075.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".



```

File Edit Selection View Go Debug Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Marlin > C Configuration_adv.h > X.RSENSE
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.075 //GADGETANGEL was 0.11]
2007 #define X_CHAIN_POS -1 // <=0 : Not chained. 1 : MCU MOST connected. 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.075 //GADGETANGEL was 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif
2025
2026 #if AXIS_IS_TMC(Y2)
2027 #define Y2_CURRENT 800
2028 #define Y2_CURRENT_HOME Y2_CURRENT
2029 #define Y2_MICROSTEPS 16
2030 #define Y2_RSENSE 0.11
2031
2032 #endif

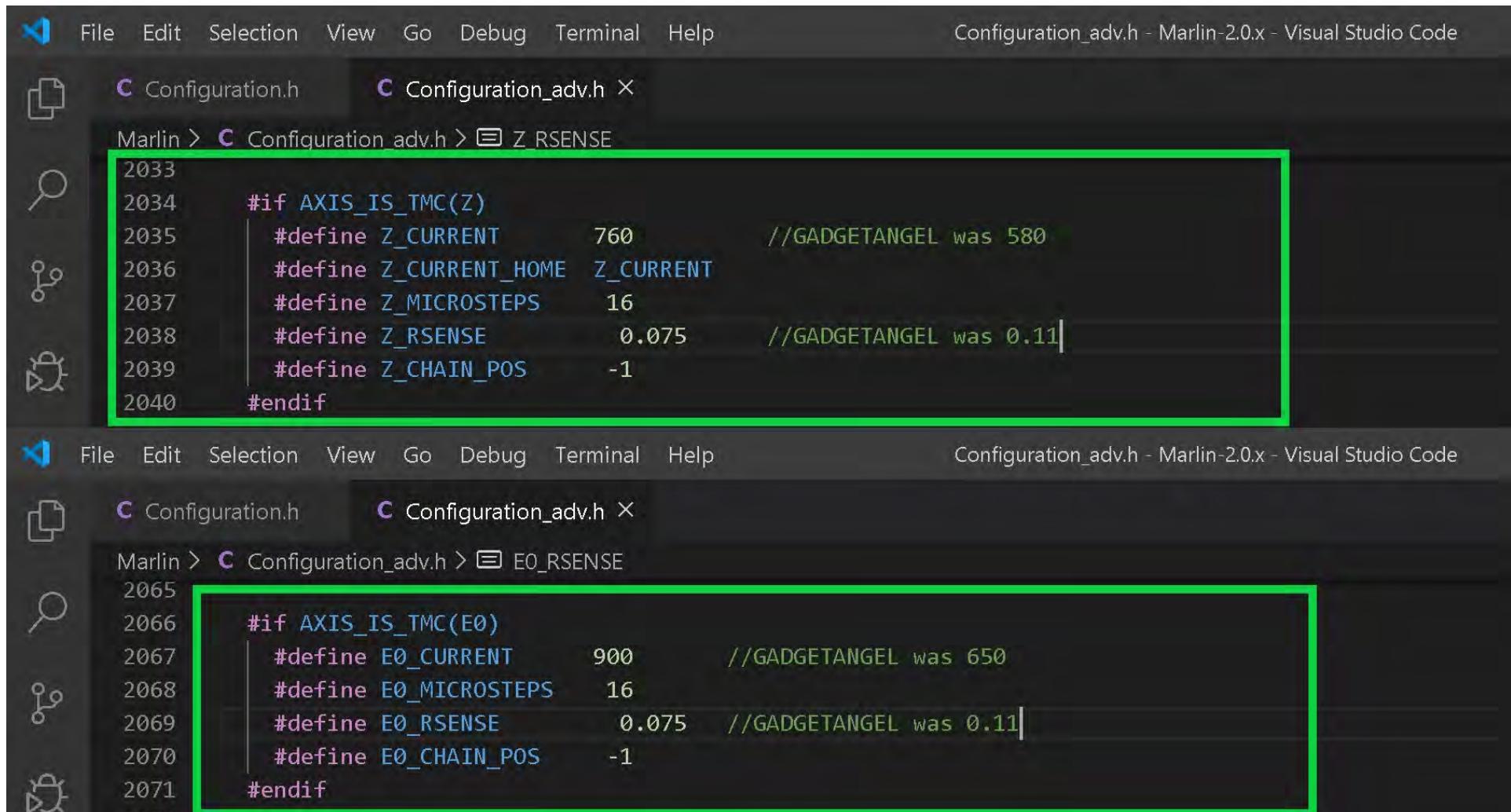
```

Ln 2006, Col 62 Spaces: 2 UTF-8 LF C++ Win32 ⌂ ⌂

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the GREEN boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.075. Ensure "E0\_RSENSE" is set to 0.075.
- If you do not want to use  $V_{ref}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



```

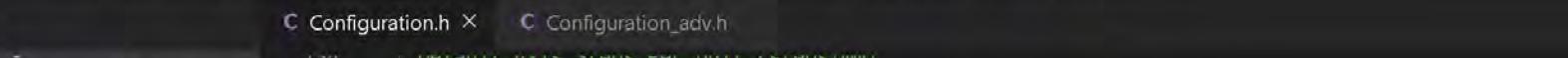
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > Z RSENSE
2033
2034 #if AXIS_IS_TMC(Z)
2035 #define Z_CURRENT 760 //GADGETANGEL was 580
2036 #define Z_CURRENT_HOME Z_CURRENT
2037 #define Z_MICROSTEPS 16
2038 #define Z_RSENSE 0.075 //GADGETANGEL was 0.11
2039 #define Z_CHAIN_POS -1
2040 #endif
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
Configuration.h Configuration_adv.h
Marlin > Configuration_adv.h > E0_RSENSE
2065
2066 #if AXIS_IS_TMC(E0)
2067 #define E0_CURRENT 900 //GADGETANGEL was 650
2068 #define E0_MICROSTEPS 16
2069 #define E0_RSENSE 0.075 //GADGETANGEL was 0.11
2070 #define E0_CHAIN_POS -1
2071 #endif

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes

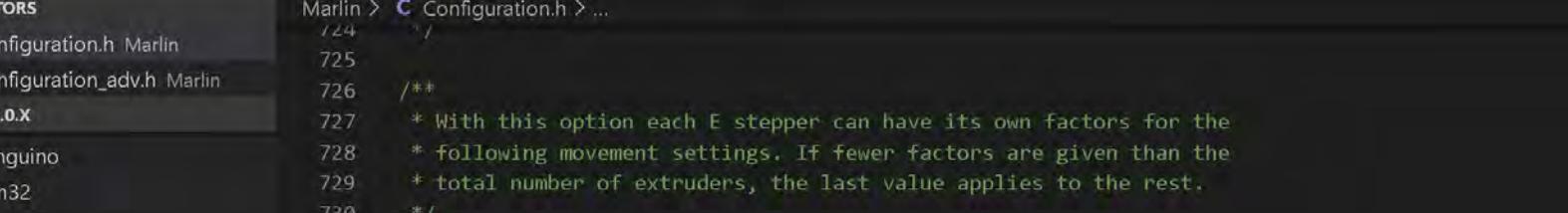


The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h Marlin 2.0.x Visual Studio Code
- Explorer View:** Shows the project structure under MARLIN-2.0.X:
  - > sanguino
  - > stm32
  - > teensy2
  - > teensy3
  - C pins.h
- Editor View:** The Configuration.h file is open. A specific line of code is highlighted with a green box:

```
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
 // {80, 80, 400, 93} for A4988 on Ender 3|
```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT"** anytime our STEPS are **NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code
- Left Sidebar:** Explorer, Open Editors, MARLIN-2.0.X (expanded), showing files like Configuration.h, Configuration\_adv.h, pins.h, pinsDebug\_list.h, pinsDebug.h, sensitive\_pins.h, MarlinCore.cpp, MarlinCore.h, \_Bootscreen.h, \_Statusscreen.h, and Configuration\_adv.h.
- Central Area:** The code editor displays Configuration.h with the following content:

```
124 /*
125
126 /**
127 * With this option each E stepper can have its own factors for the
128 * following movement settings. If fewer factors are given than the
129 * total number of extruders, the last value applies to the rest.
130 */
131 //##define DISTINCT_E_FACTORS
132 /**
133 * Default Axis Steps Per Unit (steps/mm)
134 * Override with M92
135 *
136 * X, Y, Z, E0 [, E1[, E2...]]
137 */
138 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } //GADGETANGEL was
139 // {80, 80, 400, 93} for A4988 on Ender 3
140 // Double because we are going
141 // to 1/32 from 1/16
142
143 */
```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- We need to uncomment out the "TMC\_USE\_SW\_SPI" because the SKR V1.3 pins file depends on this variable to define its SPI pins

File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration\_adv.h ×

OPEN EDITORS Configuration.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X sanguino

```
Marlin > Configuration_adv.h > TMC_USE_SW_SPI
2144 * The default SW SPI pins are defined the respective pins files,
2145 * but you can override or define them here.
2146 */
2147 #define TMC_USE_SW_SPI //GADGETANGEL was commented out
2148 //#define TMC_SW_MOST -1
```

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I **want stealthChop enabled** so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

File Edit Selection View Go Debug Terminal Help Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration\_adv.h ×

OPEN EDITORS Configuration.h Marlin Configuration\_adv.h Marlin

MARLIN-2.0.X sanguino stm32 teensy2 teensy3 pins.h pinsDebug\_list.h pinsDebug.h sensitive\_pins.h sd MarlinCore.cpp MarlinCore.h \_Bootscreen.h \_Statusscreen.h Configuration\_adv.h Configuration.h

```
Marlin > Configuration_adv.h > STEALTHCHOP_XY
2193 */
2194 #define STEALTHCHOP_XY
2195 #define STEALTHCHOP_Z
2196 #define STEALTHCHOP_E
2197
2198 /**
2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
2200 * or with the help of an example included in the library.
2201 * Provided parameter sets are
2202 * CHOPPER_DEFAULT_12V
2203 * CHOPPER_DEFAULT_19V
2204 * CHOPPER_DEFAULT_24V
2205 * CHOPPER_DEFAULT_36V
2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
2208 *
2209 * Define your own with
2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213 /**
```

- Go to the next page.

[The \(latest release of\) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode](#)

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default, it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration\_adv.h - Marlin 2.0.x - Visual Studio Code
- Left Sidebar (EXPLORER):** Shows a tree view of files and folders under "OPEN EDITORS" and "MARLIN-2.0.X". The "Configuration\_adv.h" file is selected in the "MARLIN-2.0.X" section.
- Right Editor Area:** Displays the content of the "Configuration\_adv.h" file. A green box highlights the "HYBRID\_THRESHOLD" section.

```
Marlin > Configuration_adv.h > HYBRID_THRESHOLD

2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259

 /**
 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
 * This mode allows for faster movements at the expense of higher noise levels.
 * STEALTHCHOP_(XYZ|E) must be enabled to use HYBRID_THRESHOLD.
 * M913 X/Y/Z/E to live tune the setting
 */
#define HYBRID_THRESHOLD //GADGETANGEL was commented out

#define X_HYBRID_THRESHOLD 100 // [mm/s]
#define X2_HYBRID_THRESHOLD 100
#define Y_HYBRID_THRESHOLD 100
#define Y2_HYBRID_THRESHOLD 100
#define Z_HYBRID_THRESHOLD 3
#define Z2_HYBRID_THRESHOLD 3
#define Z3_HYBRID_THRESHOLD 3
#define Z4_HYBRID_THRESHOLD 3
#define E0_HYBRID_THRESHOLD 30
#define E1_HYBRID_THRESHOLD 30
#define E2_HYBRID_THRESHOLD 30
#define E3_HYBRID_THRESHOLD 30
#define E4_HYBRID_THRESHOLD 30
#define E5_HYBRID_THRESHOLD 30
#define E6_HYBRID_THRESHOLD 30
#define E7_HYBRID_THRESHOLD 30

 /**
```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line](#).

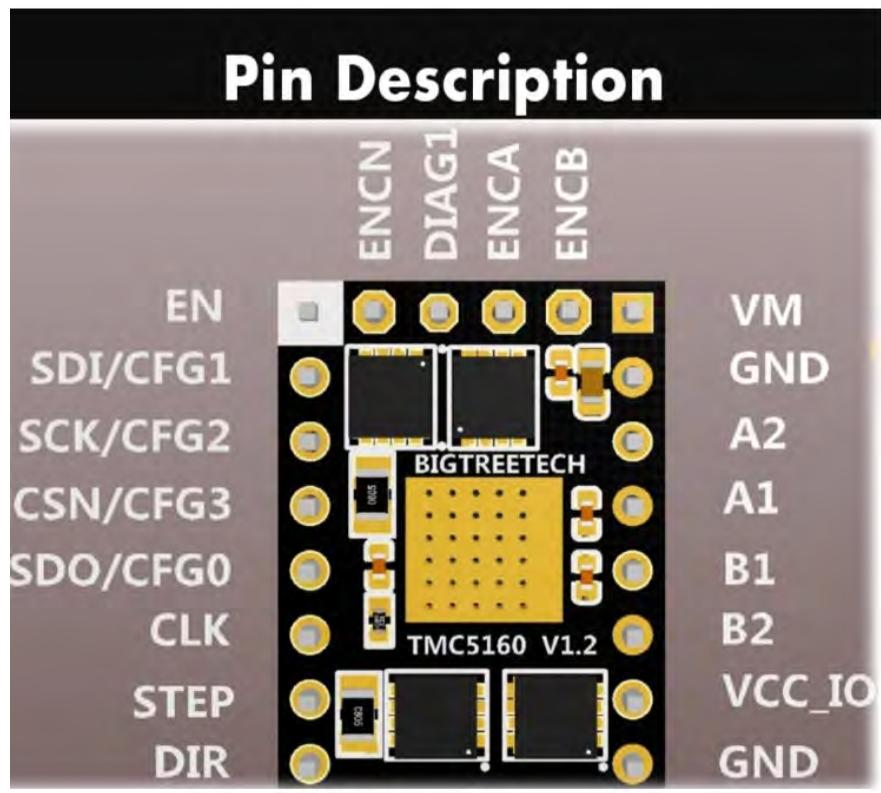
**Configuration\_adv.h - Marlin-2.0.x - Visual Studio Code**

```

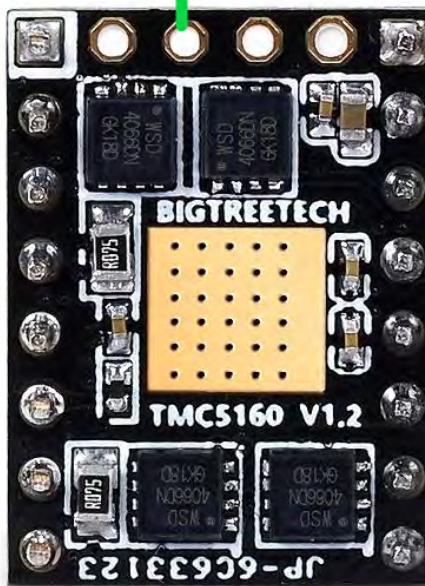
File Edit Selection View Go Debug Terminal Help Configuration.h Configuration_adv.h X
EXPLORER Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
OPEN EDITORS 2211 */
 2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
 2213
 2214 /**
 2215 * Monitor Trinamic drivers for error conditions,
 2216 * like overtemperature and short to ground.
 2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
 2218 * Other detected conditions can be used to stop the current print.
 2219 * Relevant g-codes:
 2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
 2221 * M911 - Report stepper driver overtemperature pre-warn condition.
 2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
 2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
 2224 */
 2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
 2226
 2227 #if ENABLED(MONITOR_DRIVER_STATUS)
 2228
 2229
 2230
 2231
 2232
 2233
 2234
 2235
 2236
 2237
 2238
 2239
 2240
 2241
 2242
 2243
 2244
 2245
 2246
 2247
 2248
 2249
 2250
 2251
 2252
 2253
 2254
 2255
 2256
 2257
 2258
 2259
 2260
 2261
 2262
 2263
 2264
 2265
 2266
 2267
 2268
 2269
 2270
 2271
 2272
 2273
 2274
 2275
 2276
 2277
 2278
 2279
 2280
 2281
 2282
 2283
 2284
 2285
 2286
 2287
 2288
 2289
 2290
 2291
 2292
 2293
 2294
 2295
 2296
 2297
 2298
 2299
 2300
 2301
 2302
 2303
 2304
 2305
 2306
 2307
 2308
 2309
 2310
 2311
 2312
 2313
 2314
 2315
 2316
 2317
 2318
 2319
 2320
 2321
 2322
 2323
 2324
 2325
 2326
 2327
 2328
 2329
 2330
 2331
 2332
 2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344
 2345
 2346
 2347
 2348
 2349
 2350
 2351
 2352
 2353
 2354
 2355
 2356
 2357
 2358
 2359
 2360
 2361
 2362
 2363
 2364
 2365
 2366
 2367
 2368
 2369
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377
 2378
 2379
 2380
 2381
 2382
 2383
 2384
 2385
 2386
 2387
 2388
 2389
 2390
 2391
 2392
 2393
 2394
 2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410
 2411
 2412
 2413
 2414
 2415
 2416
 2417
 2418
 2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452
 2453
 2454
 2455
 2456
 2457
 2458
 2459
 2460
 2461
 2462
 2463
 2464
 2465
 2466
 2467
 2468
 2469
 2470
 2471
 2472
 2473
 2474
 2475
 2476
 2477
 2478
 2479
 2480
 2481
 2482
 2483
 2484
 2485
 2486
 2487
 2488
 2489
 2490
 2491
 2492
 2493
 2494
 2495
 2496
 2497
 2498
 2499
 2500
 2501
 2502
 2503
 2504
 2505
 2506
 2507
 2508
 2509
 2510
 2511
 2512
 2513
 2514
 2515
 2516
 2517
 2518
 2519
 2520
 2521
 2522
 2523
 2524
 2525
 2526
 2527
 2528
 2529
 2530
 2531
 2532
 2533
 2534
 2535
 2536
 2537
 2538
 2539
 2540
 2541
 2542
 2543
 2544
 2545
 2546
 2547
 2548
 2549
 2550
 2551
 2552
 2553
 2554
 2555
 2556
 2557
 2558
 2559
 2560
 2561
 2562
 2563
 2564
 2565
 2566
 2567
 2568
 2569
 2570
 2571
 2572
 2573
 2574
 2575
 2576
 2577
 2578
 2579
 2580
 2581
 2582
 2583
 2584
 2585
 2586
 2587
 2588
 2589
 2590
 2591
 2592
 2593
 2594
 2595
 2596
 2597
 2598
 2599
 2600
 2601
 2602
 2603
 2604
 2605
 2606
 2607
 2608
 2609
 2610
 2611
 2612
 2613
 2614
 2615
 2616
 2617
 2618
 2619
 2620
 2621
 2622
 2623
 2624
 2625
 2626
 2627
 2628
 2629
 2630
 2631
 2632
 2633
 2634
 2635
 2636
 2637
 2638
 2639
 2640
 2641
 2642
 2643
 2644
 2645
 2646
 2647
 2648
 2649
 2650
 2651
 2652
 2653
 2654
 2655
 2656
 2657
 2658
 2659
 2660
 2661
 2662
 2663
 2664
 2665
 2666
 2667
 2668
 2669
 2670
 2671
 2672
 2673
 2674
 2675
 2676
 2677
 2678
 2679
 2680
 2681
 2682
 2683
 2684
 2685
 2686
 2687
 2688
 2689
 2690
 2691
 2692
 2693
 2694
 2695
 2696
 2697
 2698
 2699
 2700
 2701
 2702
 2703
 2704
 2705
 2706
 2707
 2708
 2709
 2710
 2711
 2712
 2713
 2714
 2715
 2716
 2717
 2718
 2719
 2720
 2721
 2722
 2723
 2724
 2725
 2726
 2727
 2728
 2729
 2730
 2731
 2732
 2733
 2734
 2735
 2736
 2737
 2738
 2739
 2740
 2741
 2742
 2743
 2744
 2745
 2746
 2747
 2748
 2749
 2750
 2751
 2752
 2753
 2754
 2755
 2756
 2757
 2758
 2759
 2760
 2761
 2762
 2763
 2764
 2765
 2766
 2767
 2768
 2769
 2770
 2771
 2772
 2773
 2774
 2775
 2776
 2777
 2778
 2779
 2780
 2781
 2782
 2783
 2784
 2785
 2786
 2787
 2788
 2789
 2790
 2791
 2792
 2793
 2794
 2795
 2796
 2797
 2798
 2799
 2800
 2801
 2802
 2803
 2804
 2805
 2806
 2807
 2808
 2809
 2810
 2811
 2812
 2813
 2814
 2815
 2816
 2817
 2818
 2819
 2820
 2821
 2822
 2823
 2824
 2825
 2826
 2827
 2828
 2829
 2830
 2831
 2832
 2833
 2834
 2835
 2836
 2837
 2838
 2839
 2840
 2841
 2842
 2843
 2844
 2845
 2846
 2847
 2848
 2849
 2850
 2851
 2852
 2853
 2854
 2855
 2856
 2857
 2858
 2859
 2860
 2861
 2862
 2863
 2864
 2865
 2866
 2867
 2868
 2869
 2870
 2871
 2872
 2873
 2874
 2875
 2876
 2877
 2878
 2879
 2880
 2881
 2882
 2883
 2884
 2885
 2886
 2887
 2888
 2889
 2890
 2891
 2892
 2893
 2894
 2895
 2896
 2897
 2898
 2899
 2900
 2901
 2902
 2903
 2904
 2905
 2906
 2907
 2908
 2909
 2910
 2911
 2912
 2913
 2914
 2915
 2916
 2917
 2918
 2919
 2920
 2921
 2922
 2923
 2924
 2925
 2926
 2927
 2928
 2929
 2930
 2931
 2932
 2933
 2934
 2935
 2936
 2937
 2938
 2939
 2940
 2941
 2942
 2943
 2944
 2945
 2946
 2947
 2948
 2949
 2950
 2951
 2952
 2953
 2954
 2955
 2956
 2957
 2958
 2959
 2960
 2961
 2962
 2963
 2964
 2965
 2966
 2967
 2968
 2969
 2970
 2971
 2972
 2973
 2974
 2975
 2976
 2977
 2978
 2979
 2980
 2981
 2982
 2983
 2984
 2985
 2986
 2987
 2988
 2989
 2990
 2991
 2992
 2993
 2994
 2995
 2996
 2997
 2998
 2999
 3000
 3001
 3002
 3003
 3004
 3005
 3006
 3007
 3008
 3009
 3010
 3011
 3012
 3013
 3014
 3015
 3016
 3017
 3018
 3019
 3020
 3021
 3022
 3023
 3024
 3025
 3026
 3027
 3028
 3029
 3030
 3031
 3032
 3033
 3034
 3035
 3036
 3037
 3038
 3039
 3040
 3041
 3042
 3043
 3044
 3045
 3046
 3047
 3048
 3049
 3050
 3051
 3052
 3053
 3054
 3055
 3056
 3057
 3058
 3059
 3060
 3061
 3062
 3063
 3064
 3065
 3066
 3067
 3068
 3069
 3070
 3071
 3072
 3073
 3074
 3075
 3076
 3077
 3078
 3079
 3080
 3081
 3082
 3083
 3084
 3085
 3086
 3087
 3088
 3089
 3090
 3091
 3092
 3093
 3094
 3095
 3096
 3097
 3098
 3099
 3100
 3101
 3102
 3103
 3104
 3105
 3106
 3107
 3108
 3109
 3110
 3111
 3112
 3113
 3114
 3115
 3116
 3117
 3118
 3119
 3120
 3121
 3122
 3123
 3124
 3125
 3126
 3127
 3128
 3129
 3130
 3131
 3132
 3133
 3134
 3135
 3136
 3137
 3138
 3139
 3140
 3141
 3142
 3143
 3144
 3145
 3146
 3147
 3148
 3149
 3150
 3151
 3152
 3153
 3154
 3155
 3156
 3157
 3158
 3159
 3160
 3161
 3162
 3163
 3164
 3165
 3166
 3167
 3168
 3169
 3170
 3171
 3172
 3173
 3174
 3175
 3176
 3177
 3178
 3179
 3180
 3181
 3182
 3183
 3184
 3185
 3186
 3187
 3188
 3189
 3190
 3191
 3192
 3193
 3194
 3195
 3196
 3197
 3198
 3199
 3200
 3201
 3202
 3203
 3204
 3205
 3206
 3207
 3208
 3209
 3210
 3211
 3212
 3213
 3214
 3215
 3216
 3217
 3218
 3219
 3220
 3221
 3222
 3223
 3224
 3225
 3226
 3227
 3228
 3229
 3230
 3231
 3232
 3233
 3234
 3235
 3236
 3237
 3238
 3239
 3240
 3241
 3242
 3243
 3244
 3245
 3246
 3247
 3248
 3249
 3250
 3251
 3252
 3253
 3254
 3255
 3256
 3257
 3258
 3259
 3260
 3261
 3262
 3263
 3264
 3265
 3266
 3267
 3268
 3269
 3270
 3271
 3272
 3273
 3274
 3275
 3276
 3277
 3278
 3279
 3280
 3281
 3282
 3283
 3284
 3285
 3286
 3287
 3288
 3289
 3290
 3291
 3292
 3293
 3294
 3295
 3296
 3297
 3298
 3299
 3300
 3301
 3302
 3303
 3304
 3305
 3306
 3307
 3308
 3309
 3310
 3311
 3312
 3313
 3314
 3315
 3316
 3317
 3318
 3319
 3320
 3321
 3322
 3323
 3324
 3325
 3326
 3327
 3328
 3329
 3330
 3331
 3332
 3333
 3334
 3335
 3336
 3337
 3338
 3339
 3340
 3341
 3342
 3343
 3344
 3345
 3346
 3347
 3348
 3349
 3350
 3351
 3352
 3353
 3354
 3355
 3356
 3357
 3358
 3359
 3360
 3361
 3362
 3363
 3364
 3365
 3366
 3367
 3368
 3369
 3370
 3371
 3372
 3373
 3374
 3375
 3376
 3377
 3378
 3379
 3380
 3381
 3382
 3383
 3384
 3385
 3386
 3387
 3388
 3389
 3390
 3391
 3392
 3393
 3394
 3395
 3396
 3397
 3398
 3399
 3400
 3401
 3402
 3403
 3404
 3405
 3406
 3407
 3408
 3409
 3410
 3411
 3412
 3413
 3414
 3415
 3416
 3417
 3418
 3419
 3420
 3421
 3422
 3423
 3424
 3425
 3426
 3427
 3428
 3429
 3430
 3431
 3432
 3433
 3434
 3435
 3436
 3437
 3438
 3439
 3440
 3441
 3442
 3443
 3444
 3445
 3446
 3447
 3448
 3449
 3450
 3451
 3452
 3453
 3454
 3455
 3456
 3457
 3458
 3459
 3460
 3461
 3462
 3463
 3464
 3465
 3466
 3467
 3468
 3469
 3470
 3471
 3472
 3473
 3474
 3475
 3476
 3477
 3478
 3479
 3480
 3481
 3482
 3483
 3484
 3485
 3486
 3487
 3488
 3489
 3490
 3491
 3492
 3493
 3494
 3495
 3496
 3497
 3498
 3499
 3500
 3501
 3502
 3503
 3504
 3505
 3506
 3507
 3508
 3509
 3510
 3511
 3512
 3513
 3514
 3515
 3516
 3517
 3518
 3519
 3520
 3521
 3522
 3523
 3524
 3525
 3526
 3527
 3528
 3529
 3530
 3531
 3532
 3533
 3534
 3535
 3536
 3537
 3538
 3539
 3540
 3541
 3542
 3543
 3544
 3545
 3546
 3547
 3548
 3549
 3550
 3551
 3552
 3553
 3554
 3555
 3556
 3557
 3558
 3559
 3560
 3561
 3562
 3563
 3564
 3565
 3566
 3567
 3568
 3569
 3570
 3571
 3572
 3573
 3574
 3575
 3576
 3577
 3578
 3579
 3580
 3581
 3582
 3583
 3584
 3585
 3586
 3587
 3588
 3589
 3590
 3591
 3592
 3593
 3594
 3595
 3596
 3597
 3598
 3599
 3600
 3601
 3602
 3603
 3604
 3605
 3606
 3607
 3608
 3609
 3610
 3611
 3612
 3613
 3614
 3615
 3616
 3617
 3618
 3619
 3620
 3621
 3622
 3623
 3624
 3625
 3626
 3627
 3628
 3629
 3630
 3631
 3632
 3633
 3634
 3635
 3636
 3637
 3638
 3639
 3640
 3641
 3642
 3643
 3644
 3645
 3646
 3647
 3648
 3649
 3650
 3651
 3652
 3653
 3654
 3655
 3656
 3657
 3658
 3659
 3660
 3661
 3662
 3663
 3664
 3665
 3666
 3667
 3668
 3669
 3670
 3671
 3672
 3673
 3674
 3675
 3676
 3677
 3678
 3679
 3680
 3681
 3682
 3683
 3684
 3685
 3686
 3687
 3688
 3689
 3690
 3691
 3692
 3693
 3694
 3695
 3696
 3697
 3698
 3699
 3700
 3701
 3702
 3703
 3704
 3705
 3706
 3707
 3708
 3709
 3710
 3711
 3712
 3713
 3714
 3715
 3716
 3717
 3718
 3719
 3720
 3721
 3722
 3723
 3724
 3725
 3726
 3727
 3728
 3729
 3730
 3731
 3732
 3733
 3734
 3735
 3736
 3737
 3738
 3739
 3740
 3741
 3742
 3743
 3744
 3745
 3746
 3747
 3748
 3749
 3750
 3751
 37
```

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- This next section covers sensor-less homing which is available for the TMC5160 in SPI mode. I want to enable it BUT for the TMC5160 I first have to solder on the DIAG1 pin onto each TMC5160 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG1 pin for two TMC5160 drivers. Here is a picture of the TMC5160 V1.2 pin-out.



**DIAG1 PIN**



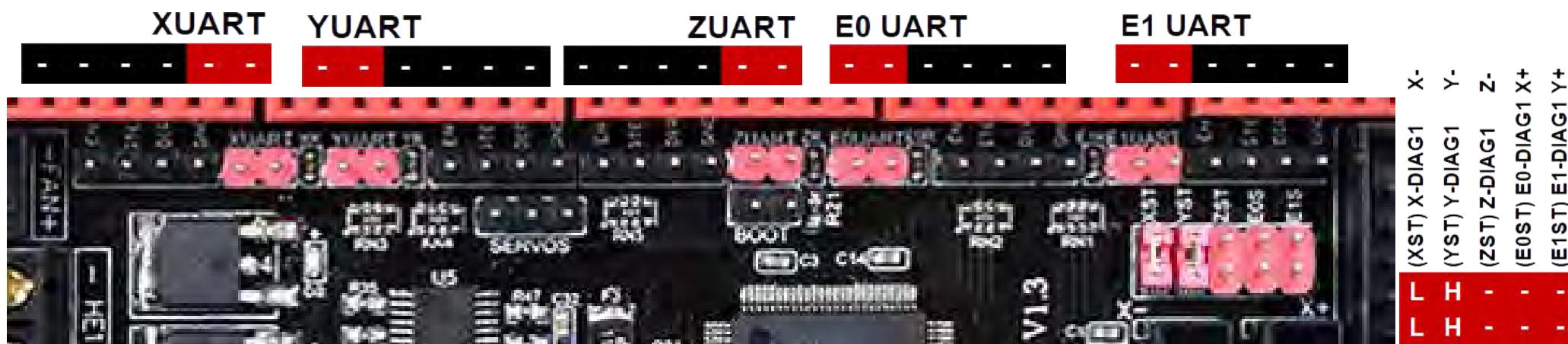
**DIAG1 PIN**



- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5160 into the SKR V1.3 board the DIAG1 pin will be seated in the SKR V1.3 board. Then I can use the XST jumper to connect the DIAG1 pin to the X endstop. I can then, also, use the YST jumper to connect the DIAG1 pin to the Y endstop. Connecting the DIAG1 pins to the endstops enables sensor-less homing to work.
- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG1 pin on the TMC5160 driver has to be plugged into the SKR V1.3 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG1 pin on the Z axis TMC5160 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG1 pin of the TMC5160 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG1 pin of the TMC5160 for the Y axis. Therefore my jumpers for XST, YST, ZST, E0S, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers that are in place are the "L" and "H" jumpers.



- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS\_HOMING"

```

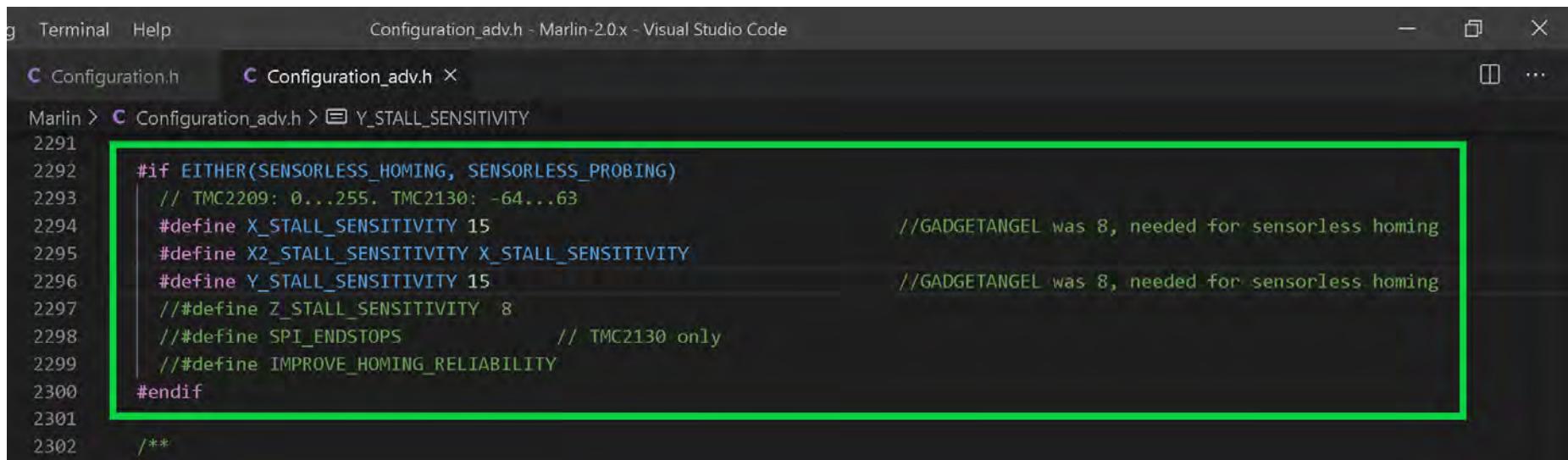
File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h X
OPEN EDITORS Configuration.h Marlin Configuration_adv.h Marlin
Marlin > Configuration_adv.h > SENSORLESS_HOMING
2281 */
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out
2283

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Next we set the "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X\_STALL\_SENSITIVITY" and the "Y\_STALL\_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z\_STALL\_SENSITIVITY".

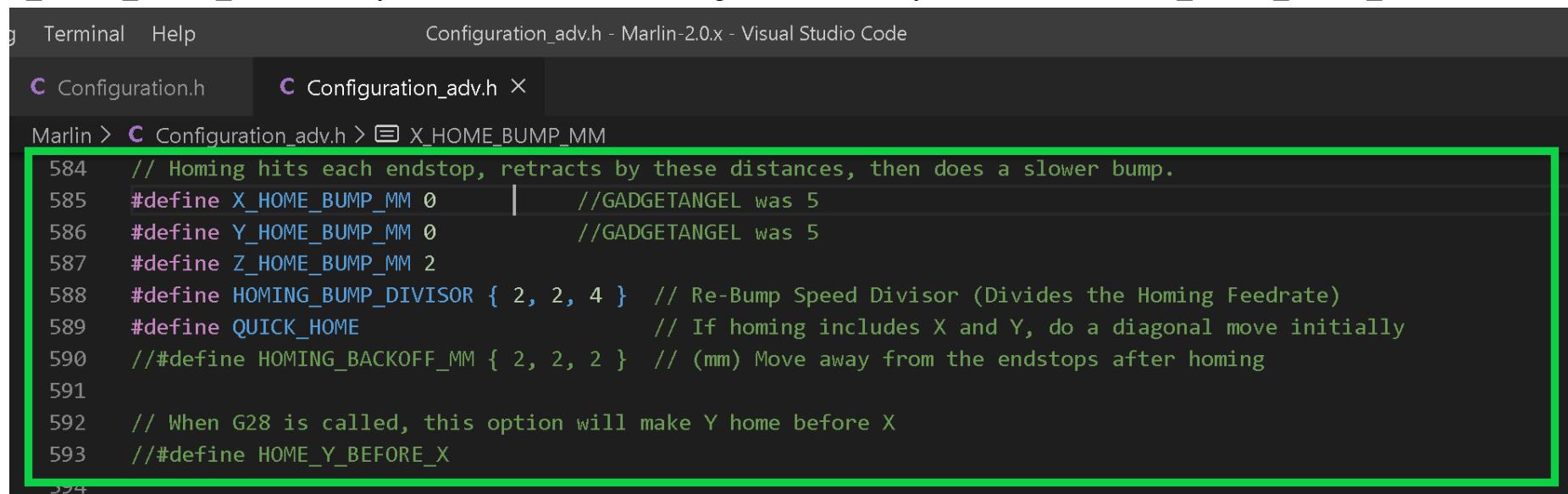


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X\_HOME\_BUMP\_MM" to 0 and "Y\_HOME\_BUMP\_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z\_HOME\_BUMP\_MM" to 0.



```

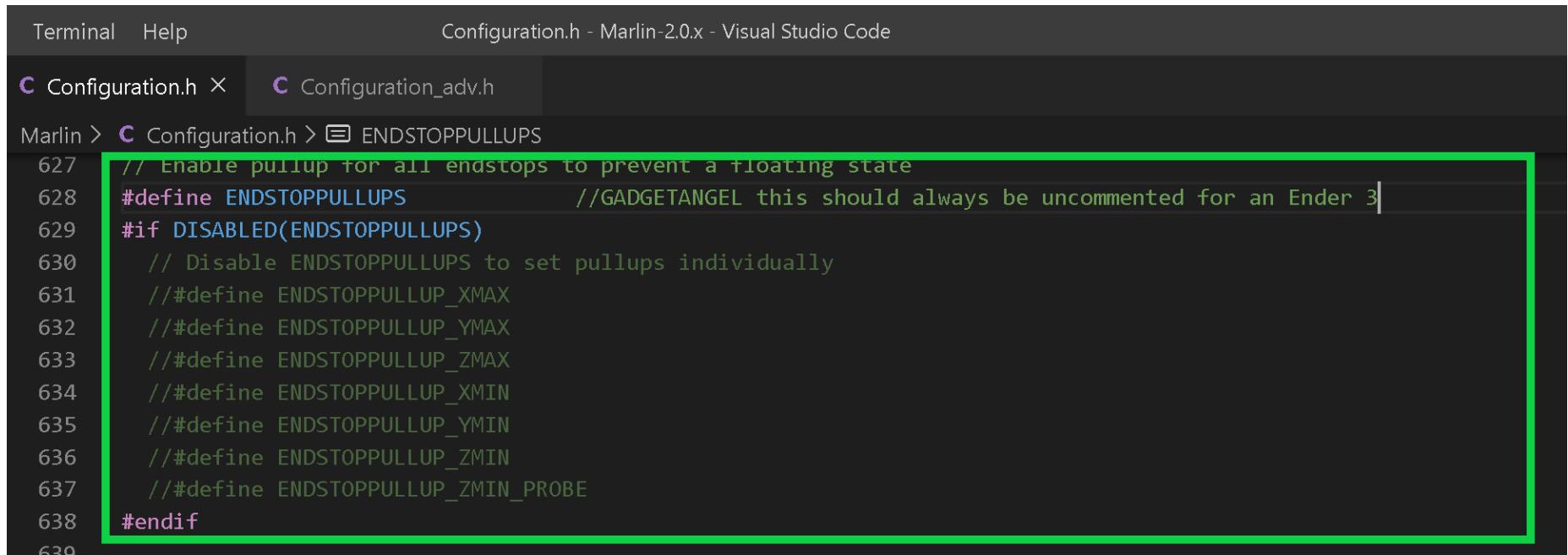
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL was 5
587 #define Z_HOME_BUMP_MM 2
588 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
589 #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
590 //">#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
591
592 // When G28 is called, this option will make Y home before X
593 //">#define HOME_Y_BEFORE_X
594

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default.



Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

Configuration.h X Configuration\_adv.h

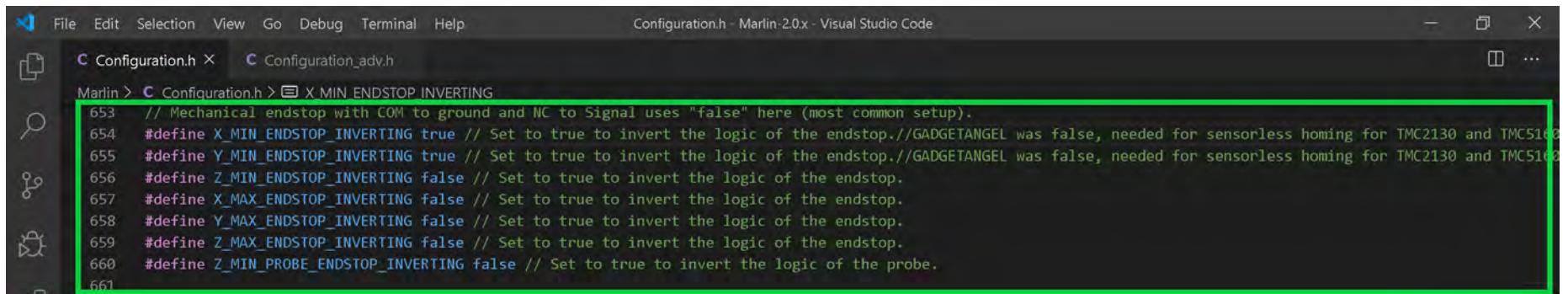
Marlin > Configuration.h > ENDSTOPPULLUPS

```

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 //#define ENDSTOPPULLUP_XMAX
632 //#define ENDSTOPPULLUP_YMAX
633 //#define ENDSTOPPULLUP_ZMAX
634 //#define ENDSTOPPULLUP_XMIN
635 //#define ENDSTOPPULLUP_YMIN
636 //#define ENDSTOPPULLUP_ZMIN
637 //#define ENDSTOPPULLUP_ZMIN_PROBE
638 #endif
639

```

- Next to allow sensor-less homing to work (while using the BIQU TMC5160) we need to change our end stop logic. Therefore I set "X\_MIN\_ENDSTOP\_INVERTING" to true and "Y\_MIN\_ENSTOP\_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z\_MIN\_ENDSTOP\_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z\_MIN\_ENDSTOP\_INVERTING" set to false.



File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

Configuration.h X Configuration\_adv.h

Marlin > Configuration.h > X MIN ENDSTOP INVERTING

```

653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5160
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- The end of Marlin setup for BIQU TMC5160 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



Configuration.h - Marlin-2.0.x - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER OPEN EDITORS MARLIN-2.0.X

```

Configuration.h X pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
Marlin > Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [=====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
===== [SUCCESS] Took 130.61 seconds =====

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| r3mbo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| ESP8266         | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC4709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5160 V1.2 Drivers in SPI Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

**Configuration.h - Marlin-2.0.x - Visual Studio Code**

```

File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h Configuration_adv.h
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE

```

```

C Configuration.h X C Configuration.h ...
Marlin > C Configuration.h ...
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 // #define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132 #endif
133
134 // Name displayed in the LCD "Ready" message and Info menu

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

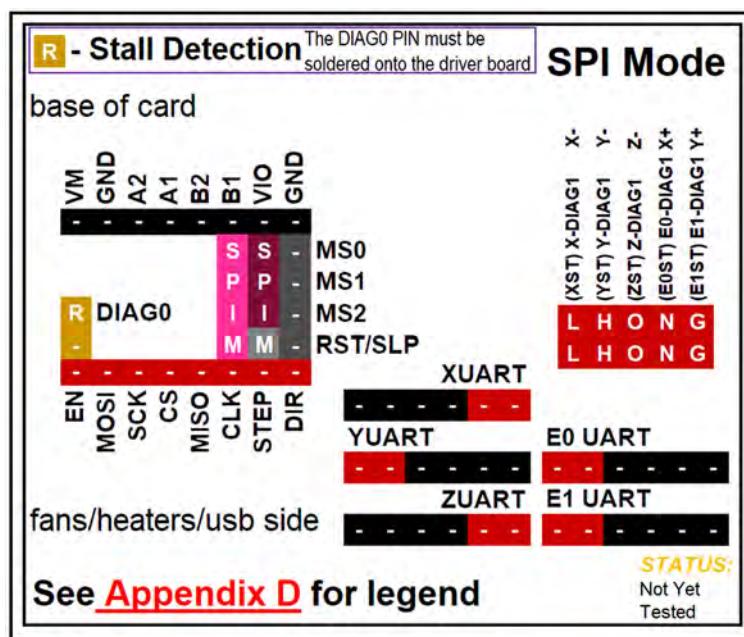
```

Compiling .pio\build\LPC1768\FrameworkArduino\Wire.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\arduino.cpp.o
Compiling .pio\build\LPC1768\FrameworkArduino\main.cpp.o
Archiving .pio\build\LPC1768\libFrameworkArduino.a
Linking .pio\build\LPC1768\firmware.elf
Checking size .pio\build\LPC1768\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 42.5% (used 13908 bytes from 32736 bytes)
Flash: [====] 35.0% (used 166124 bytes from 475136 bytes)
Building .pio\build\LPC1768\firmware.bin
[SUCCESS] Took 130.61 seconds

```

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguino1284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| SUE_L103        | IGNORED |              |
| LPC1768         | SUCCESS | 00:02:10.606 |
| LPC1709         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

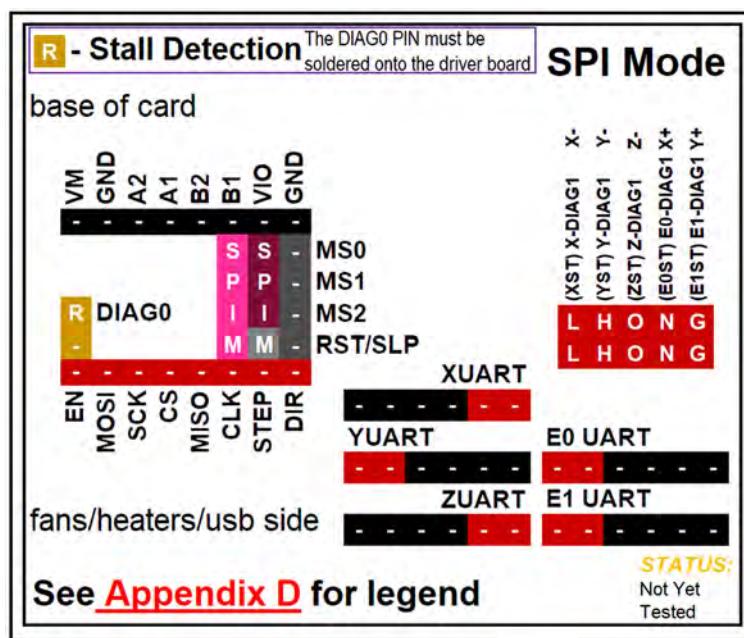
- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

**BIQU TMC5161 V1.0**SPI Mode

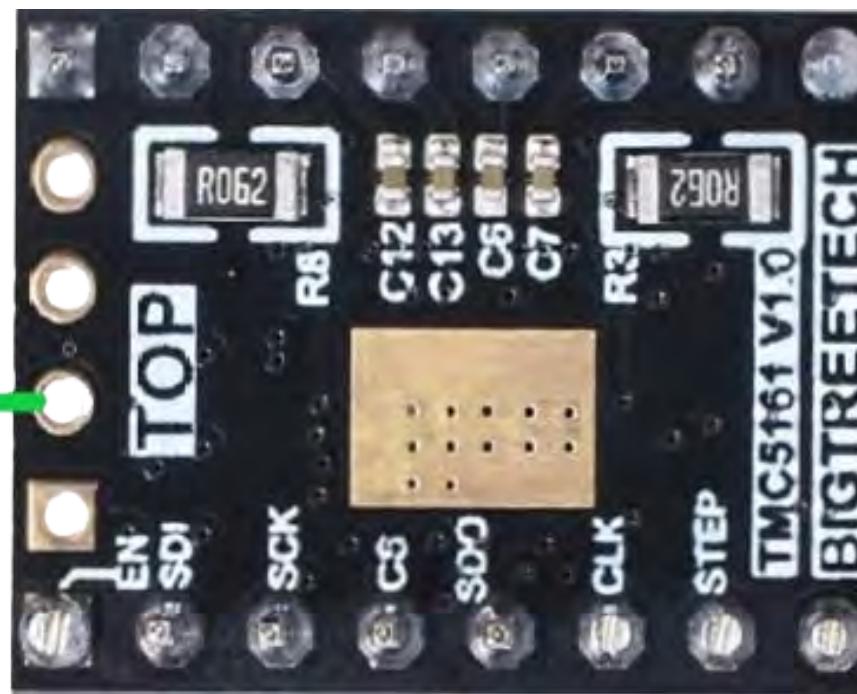
**Note:** You can use 50% to 90% of the calculated  $I_{RMS}$  ( $I_{MAX}/1.414$ ) when tuning ("X\_CURRENT", "Y\_CURRENT", etc.) the stepper motor driver in the firmware.

See the next page for further information.

|                                                                                                          |                                                                                                               |                                                                                             |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Driver Chip</b><br><br><b>TMC5161</b><br>SPI Mode<br>Maximum 256 Subdivision<br>40V DC<br>5.2A (peak) | <b>Steps are set inside<br/>of your Firmware</b>                                                              |                                                                                             |
| <b>Driving Current Calculation Formula</b><br>$R_S$ (Typical Sense Resistor) = 0.062Ω                    | $I_{MAX} = 5.2419$<br>See Appendix B #9. Use 50% to 90% as shown below:<br>$I_{MAX} = I_{MAX} * 0.90 = 4.718$ | 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<sub>ref</sub> Test point" location because the IRMS is set inside of the Firmware.



**NOTE:** BIQU TMC5161 has the ability to do sensor-less homing. By default the DIAG0 pin is **NOT** soldered onto the driver board. Therefore, for any axis you want sensor-less homing enabled, YOU WILL HAVE to solder on the DIAG0 pin.

# SPI Mode

## BIQU TMC5161 V1.0 SPI Mode

**Note:** The location of the current sense resistors are shown in **GREEN**. Use the current sense resistors' value in the Marlin Firmware ("X\_RSENSE", "Y\_RSENSE", "Z\_RSENSE", "E0\_RSENSE" and/or "E1\_RSENSE") so that the appropriate current limit can be sent to the driver board. If you do not want to use  $V_{ref}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT" and/or "E1\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS} = I_{MAX}/1.414$ ). You use 50% to 90% of the calculated  $I_{RMS}$  as the value for "X\_CURRENT", "Y\_CURRENT", "Z\_CURRENT", "E0\_CURRENT", and/or "E1\_CURRENT".

$R_s = R050$  is 0.05 Ohms

$R_s = R062$  is 0.062 Ohms

$R_s = R068$  is 0.068 Ohms

$R_s = R075$  is 0.075 Ohms

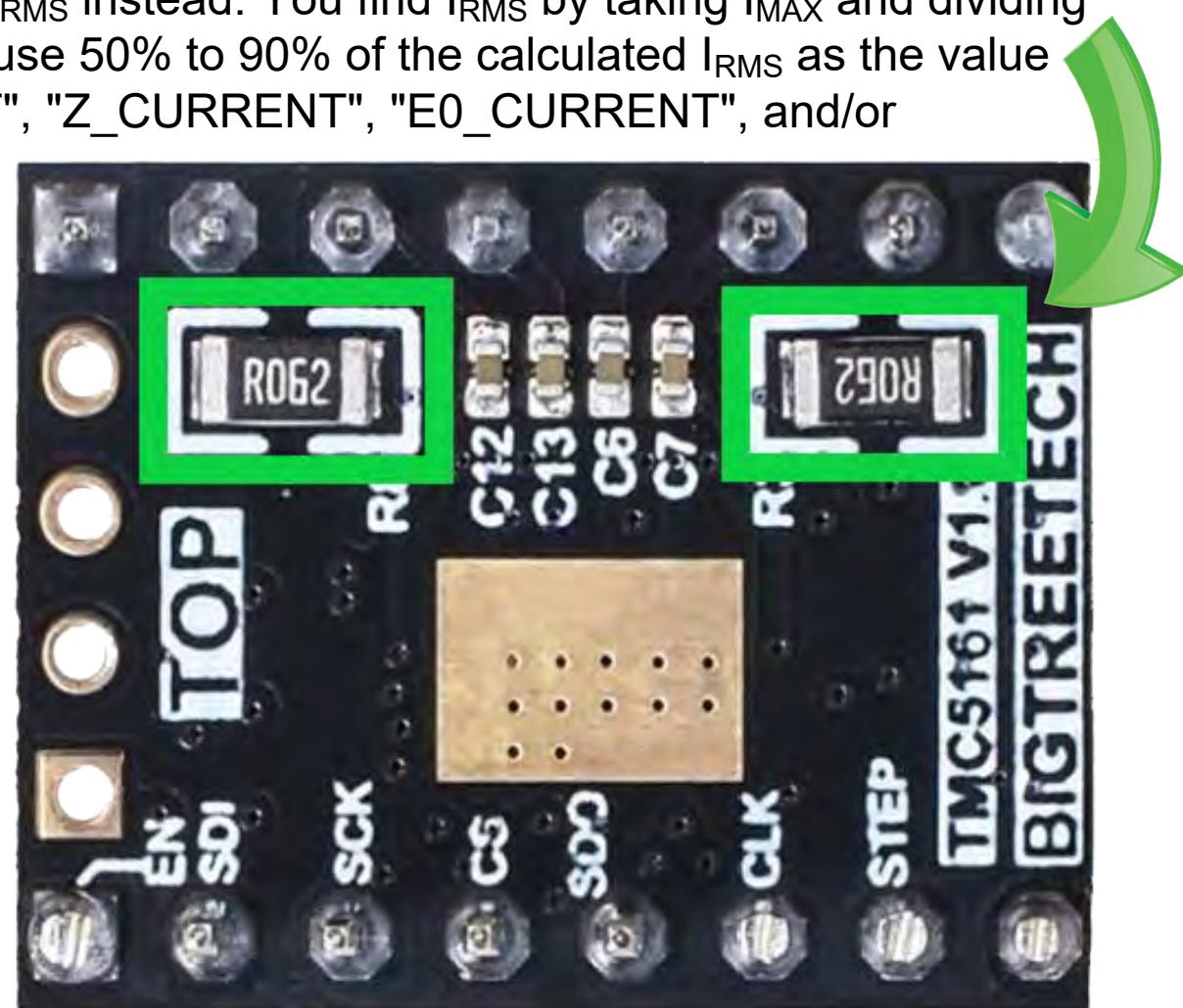
$R_s = R100$  is 0.1 Ohms

$R_s = R110$  is 0.11 Ohms

$R_s = R150$  is 0.15 Ohms

$R_s = R200$  is 0.2 Ohms

$R_s = R220$  is 0.22 Ohms



# SPI Mode

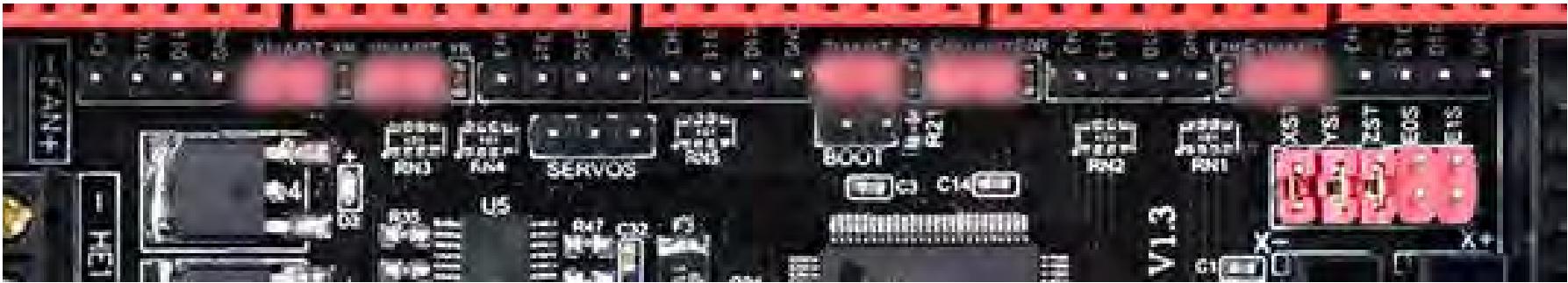
## BIGU TMC5161 V1.0

SPI Mode

|    |               |    |               |    |               |    |                 |    |                 |    |                 |   |   |   |   |   |   |   |   |
|----|---------------|----|---------------|----|---------------|----|-----------------|----|-----------------|----|-----------------|---|---|---|---|---|---|---|---|
| X- | (XST) X-DIAG1 | Y- | (YST) Y-DIAG1 | Z- | (ZST) Z-DIAG1 | X+ | (E0ST) E0-DIAG1 | X+ | (E1ST) E1-DIAG1 | Y+ | (E1ST) E1-DIAG1 |   |   |   |   |   |   |   |   |
| L  | H             | O  | -             | -  | L             | H  | O               | N  | G               | L  | H               | O | - | - | L | H | O | N | G |

Note: If you want sensor-less homing for X- (X Min), Y- (Y Min) and/or Z- (Z Min), set the appropriate JUMPER(s) ("L" for X axis,"H" for Y axis, and "O" for Z axis) on the board.

Note: If you want sensor-less homing for X+ (X Max), and Y+ (Y Max), set JUMPERS "N", and/or "G" on the board.



|    |               |    |               |    |               |    |                 |    |                 |    |                 |
|----|---------------|----|---------------|----|---------------|----|-----------------|----|-----------------|----|-----------------|
| X- | (XST) X-DIAG1 | Y- | (YST) Y-DIAG1 | Z- | (ZST) Z-DIAG1 | X+ | (E0ST) E0-DIAG1 | X+ | (E1ST) E1-DIAG1 | Y+ | (E1ST) E1-DIAG1 |
| -  | -             | -  | -             | -  | -             | -  | -               | -  | -               | -  | -               |

Note: If sensor-less homing is **not wanted** ensure the following JUMPER(s) are **empty**: "L", "H", "O", "N", and "G".

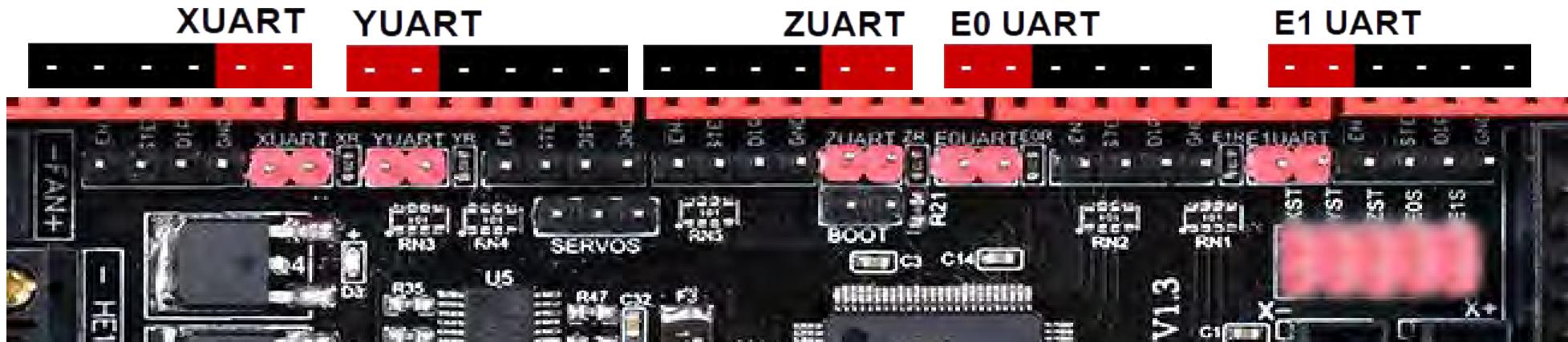


## SPI Mode

# BIQU TMC5161 V1.0

## SPI Mode

**Note:** Ensure that **ALL** the UART jumpers are **EMPTY**.



# SPI Mode

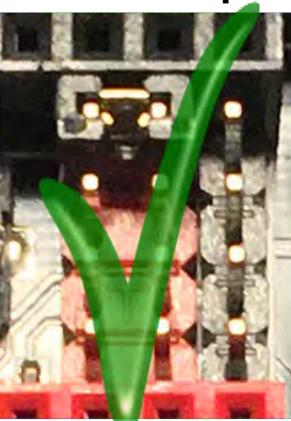
## BIQU TMC5161 V1.0 SPI Mode

### **SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode**

Example of the **WRONG** way to set the “S” jumper for SPI mode (right two columns of pins):



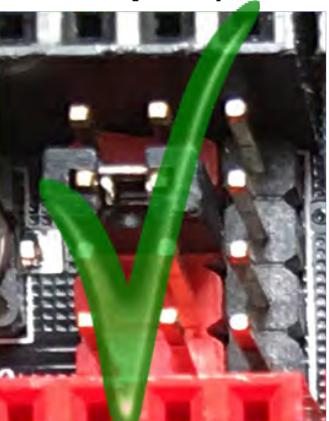
Example of the **RIGHT** way to set the “S” jumper for SPI mode (left two columns of pins):



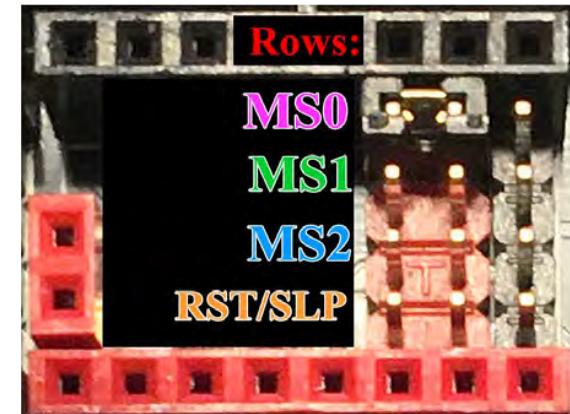
Example of the **WRONG** way to set the “P” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “P” jumper for SPI mode (left two columns of pins):



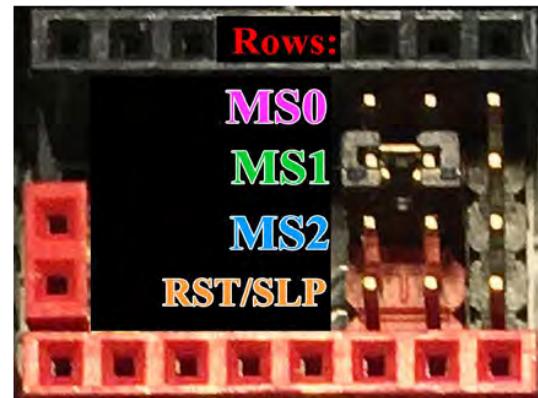
**S S** **Jumper set**



SET ‘ST’ JUMPER to enable Sensor-less Homing



**P P** **Jumper set**



SET ‘ST’ JUMPER to enable Sensor-less Homing



# SPI Mode

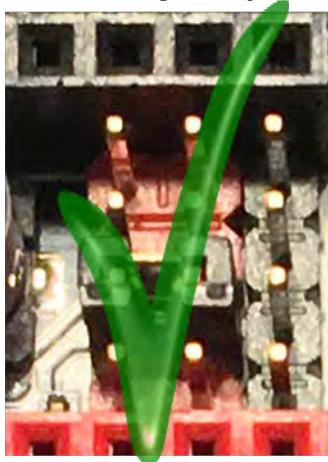
Example of the **WRONG** way to set the “I” jumper for SPI mode (right two columns of pins):



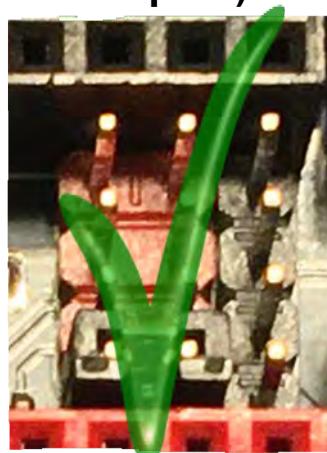
Example of the **WRONG** way to set the “M” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “I” jumper for SPI mode (left two columns of pins):



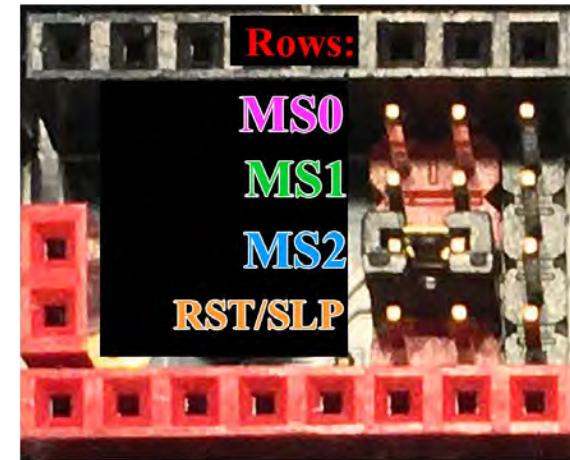
Example of the **RIGHT** way to set the “M” jumper for SPI mode (left two columns of pins):



## BIGU TMC5161 V1.0

### SPI Mode

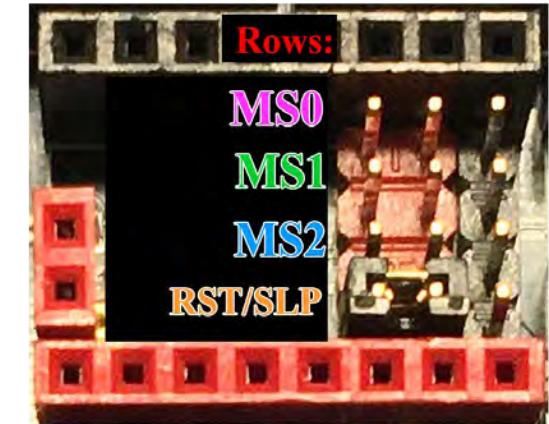
I I → **Jumper set**



SET ‘ST’ JUMPER to enable Sensor-less Homing



M M → **Jumper set**



SET ‘ST’ JUMPER to enable Sensor-less Homing



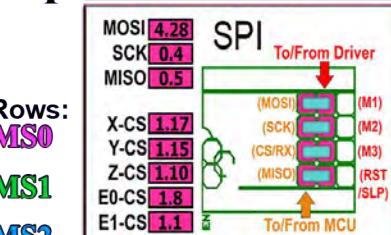
# SPI Mode

## BQU TMC5161 V1.0

### SPI Mode

## SKR V1.3 LEGEND of Driver Socket Representation for Stepper Motor Drivers in SPI Mode

### Driver Socket Representation

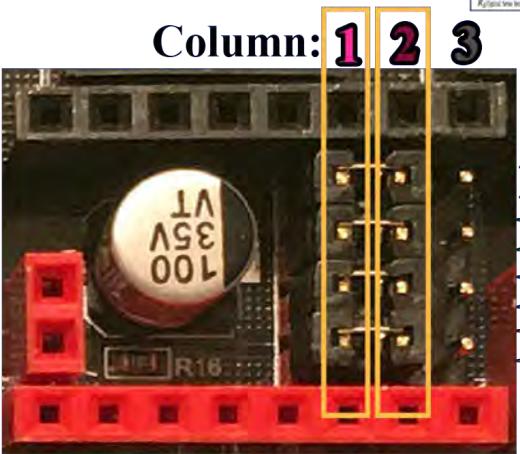
Columns: **1 2 3**

No Jumper set

**NOTE:** UART PINS must not be set if SPI mode is enabled.



## SPIM:

Column: **1 2 3**

Rows:

**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

### Driver Chip Chart:

| Driver Chip                         | Steps are set inside of your Firmware                                                |                                                                                      |
|-------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| BQU TMCXXXX                         |                                                                                      |                                                                                      |
| SPI Mode                            |                                                                                      |                                                                                      |
| Maximum 256 steps per axis          |                                                                                      |                                                                                      |
| 400 Hz, 2.5A (peak)                 |                                                                                      |                                                                                      |
| Driving Current Calculation Formula | $I_{MAX} = V_{ref}$                                                                  | $V_{ref} = I_{MAX}$                                                                  |
|                                     | See Appendix B for Use 50% to 100% of the value below:<br>$I_{MAX} = I_{max} * 0.90$ | See Appendix B for Use 50% to 100% of the value below:<br>$V_{ref} = V_{max} * 0.90$ |
|                                     |                                                                                      |                                                                                      |

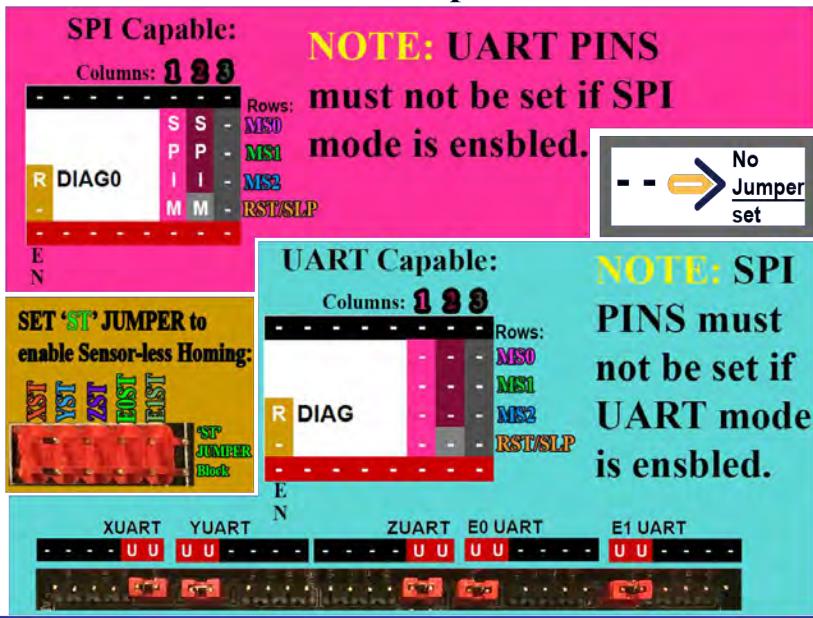
## SPI Mode

BIQU TMC5161 V1.0

## SPI Mode

# Information on Sensor-less Homing

## Driver Socket Representation



## **STALLGUARD** (Sensor-less Homing)

|    | DIAG PIN | ENDSTOP |    |
|----|----------|---------|----|
| X  | X-DIAG1  | 1.29    | X- |
| Y  | Y-DIAG1  | 1.27    | Y- |
| Z  | Z-DIAG1  | 1.25    | Z- |
| E0 | E0-DIAG1 | 1.28    | X+ |
| E1 | E1-DIAG1 | 1.26    | Y+ |

#### To/From Driver DIAG pin

**Note Concerning the TMC2209 in UART Mode ONLY:**  
If using limit switches/endstops, ensure the DIAG pin is NOT connected to the

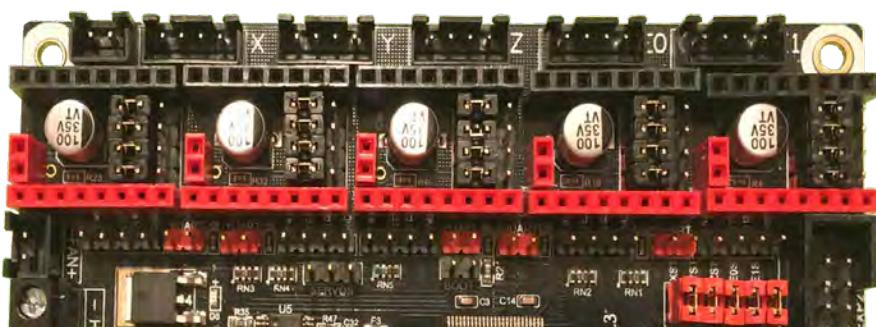
**Note:** For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to [use a filament runout sensor](#), ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the **STC\_JUMPER** is removed).

## Sensor-less Homing Capable Drivers:

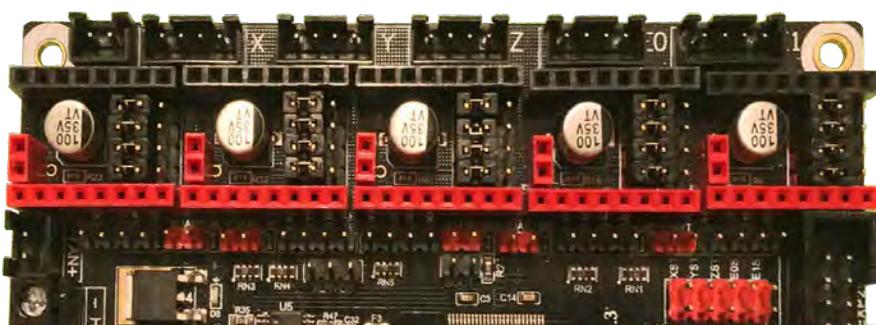
**SPI Capable Drivers:** TMC2130, TMC5160 & TMC5161

## UART Capable Drivers: TMC2209

**If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are SET for ALL Axes:**



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are REMOVED for ALL Axes:

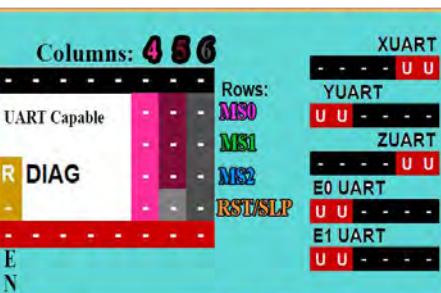
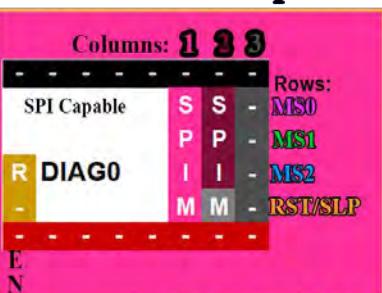


# SPI Mode

## BQU TMC5161 V1.0

### SPI Mode

#### Driver Socket Representation of 'ST' JUMPER Block



Columns: 7 8 9 10 11

Columns: 7 8 9 10 11

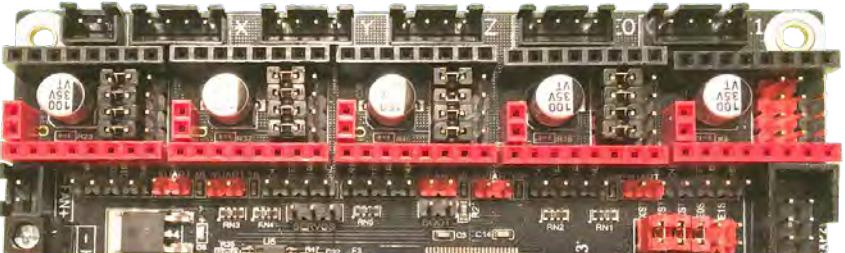
Rows:

|   |           |
|---|-----------|
| 1 | L H O - - |
| 2 | L H O - - |

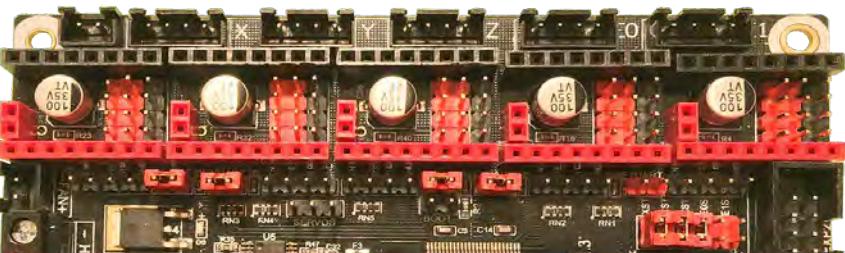
Rows:

|   |           |
|---|-----------|
| 1 | L H O N G |
| 2 | L H O N G |

#### Sensor-less Homing on X, Y, & Z via SPI mode:



#### Sensor-less Homing on X, Y, & Z via UART mode:



#### Sensor-less Homing Capable Drivers:

SPI Capable Drivers: TMC2130, TMC5160 & TMC5161

UART Capable Drivers: TMC2209



Driver  
Socket  
Representation:

Meaning:  
**set Jumper between  
row 1 and row 2  
in column 7**

Driver  
Socket  
Representation:

Meaning:  
**set Jumper between  
row 1 and row 2  
in column 8**

Driver  
Socket  
Representation:

Meaning:  
**set Jumper between  
row 1 and row 2  
in column 9**

Driver  
Socket  
Representation:

Meaning:  
**set Jumper between  
row 1 and row 2  
in column 10**

Driver  
Socket  
Representation:

Meaning:  
**set Jumper between  
row 1 and row 2  
in column 11**

# BIGTREETECH TMC5161 V1.0

## SPI Mode

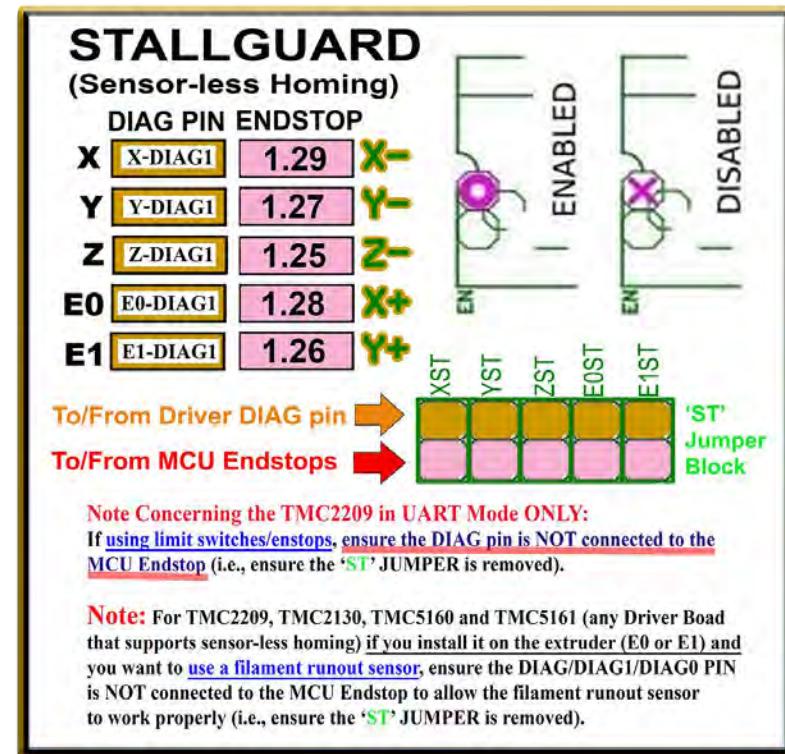
**Use the 'ST' Jumper Block**, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capability of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

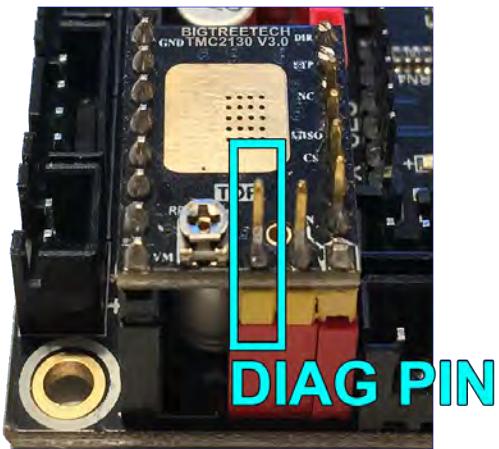
The following DO NOT have Endstops:

- Extruder Axis (E0 or E1)
- Z Axis, if a BLTouch is used to Home instead of a physical endstop

# SPI Mode

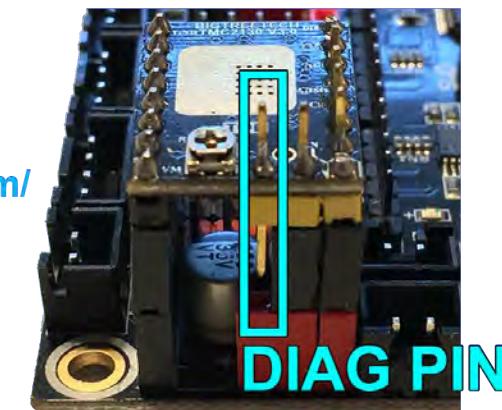
## BIQU TMC5161 V1.0 SPI Mode

To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board** **AND** the '**ST'** **JUMPER** is set in the '**'ST' JUMPER Block** for the **Axis**.



To **disable** sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the **SKR V1.3 Board** **OR** remove the '**ST**' **JUMPER** from the '**'ST' JUMPER Block**.

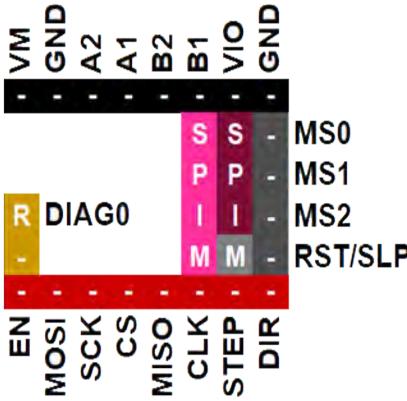
Link to stackable header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>



# SPI Mode

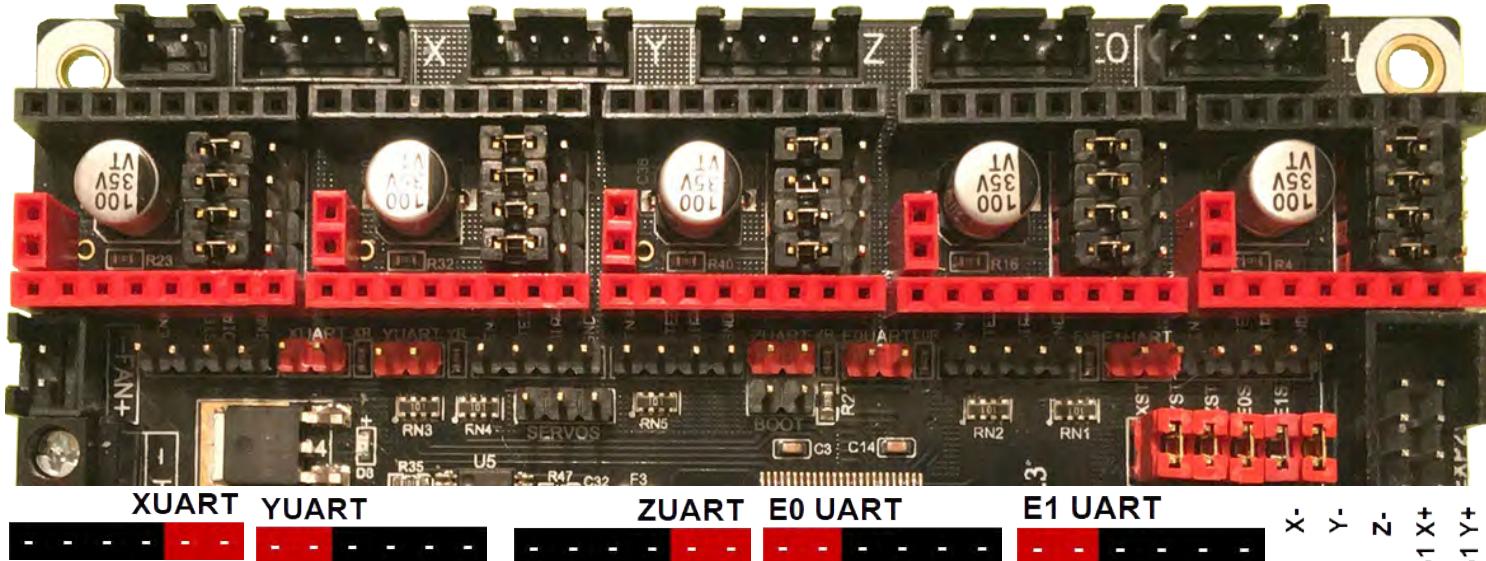
## BIQU TMC5161 V1.0 SPI Mode

**Note:** Set JUMPERS "S", "P", "I", and "M" on the board!!

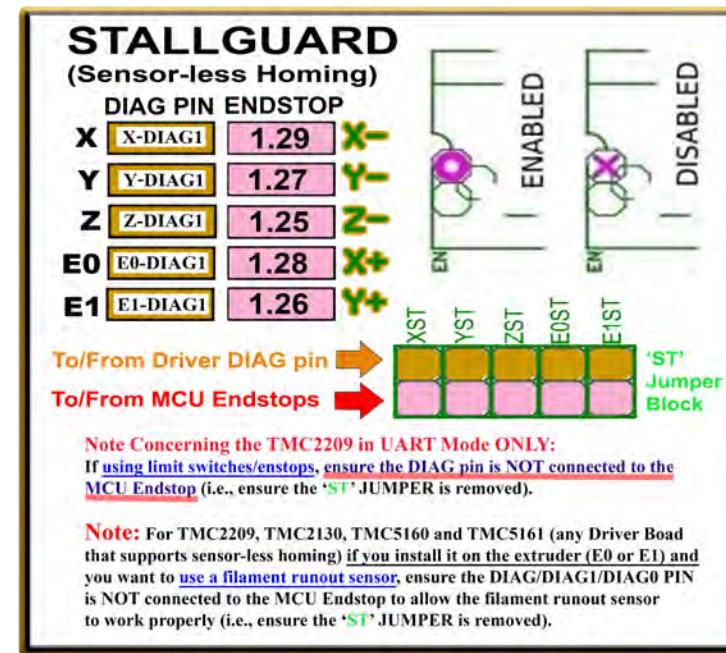
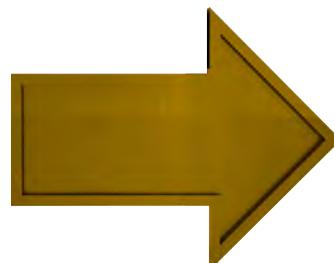


SPI

See [Appendix D](#) for legend



**Note: TMC5161 has sensor-less homing capability**



|                 |   |
|-----------------|---|
| (XST) X-DIAG1   | L |
| (YST) Y-DIAG1   | H |
| (ZST) Z-DIAG1   | L |
| (E0ST) E0-DIAG1 | H |
| (E1ST) E1-DIAG1 | L |

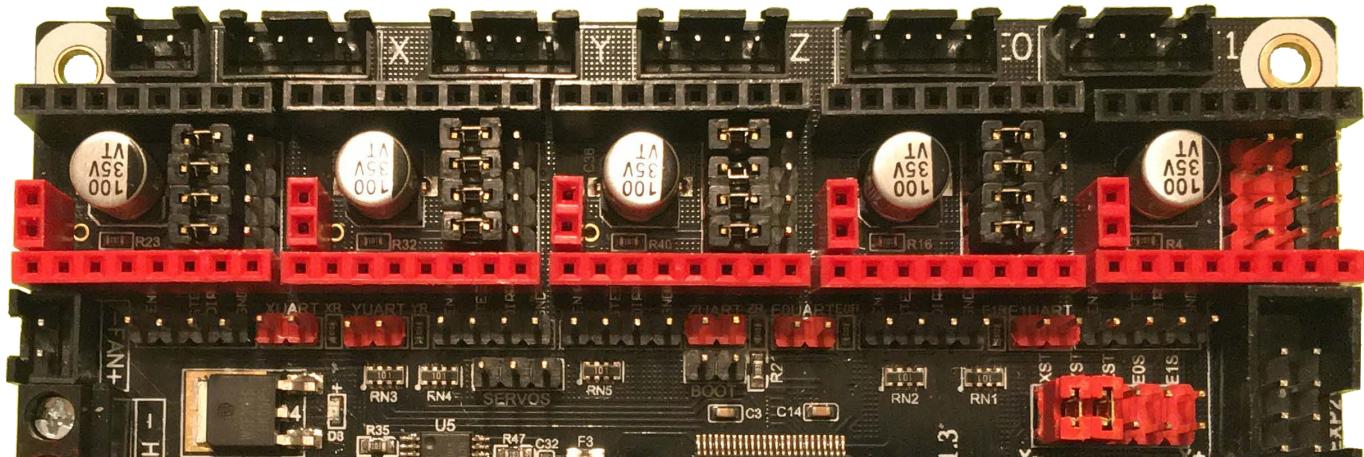
# SPI Mode

## BIQU TMC5161 V1.0 SPI Mode

### Examples of Different SPI Configurations Part 1

**X, Y, Z and E0 axes  
configured for SPI mode.**

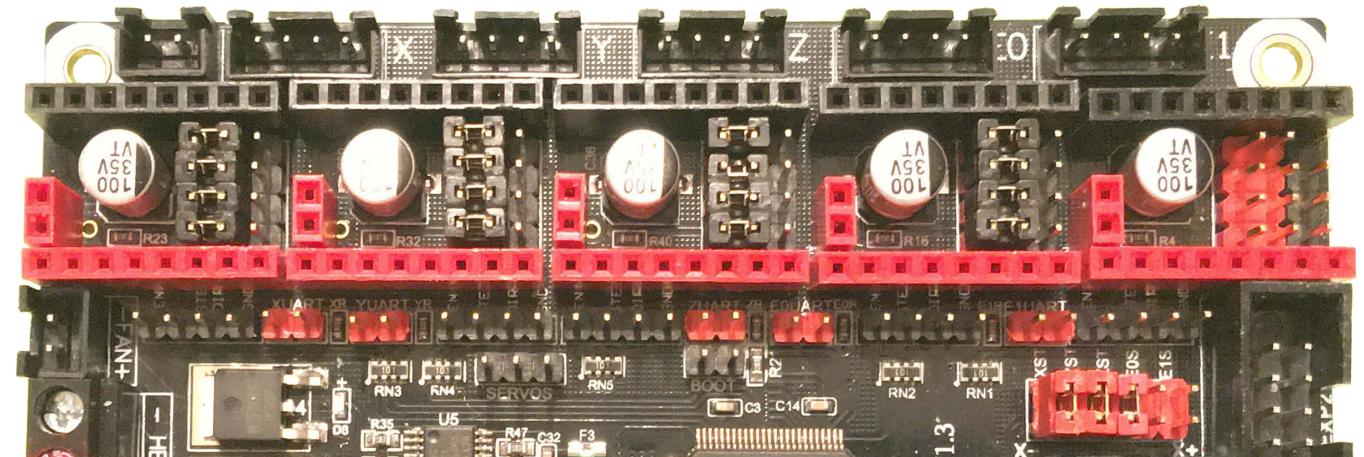
**Sensor-less homing for X  
and Y axes.**



---

**X, Y, Z and E0 axes  
configured for SPI mode.**

**Sensor-less homing for X,  
Y and Z axes.**



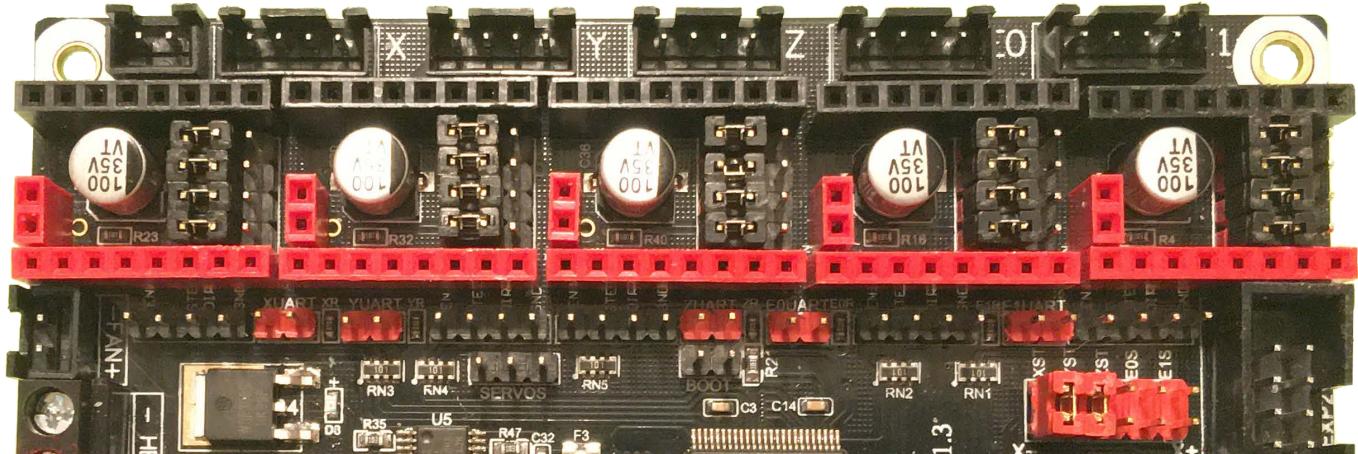
# SPI Mode

## BIQU TMC5161 V1.0 SPI Mode

### Examples of Different SPI Configurations Part 2

X, Y, Z, E0, and E1 axes  
configured for SPI mode.

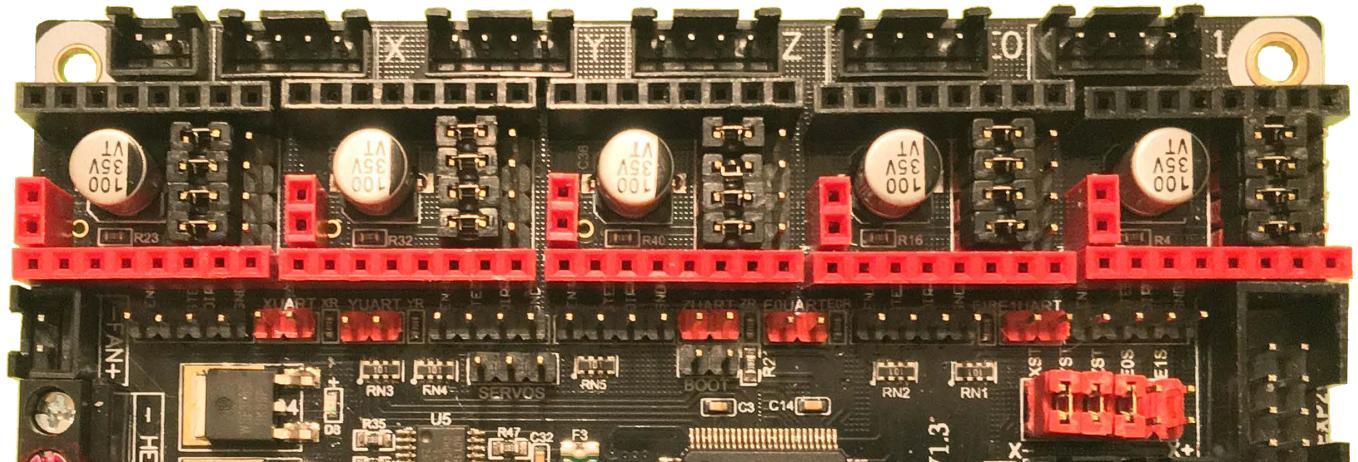
Sensor-less homing for X  
and Y axes.



---

X, Y, Z, E0 and E1 axes  
configured for SPI mode.

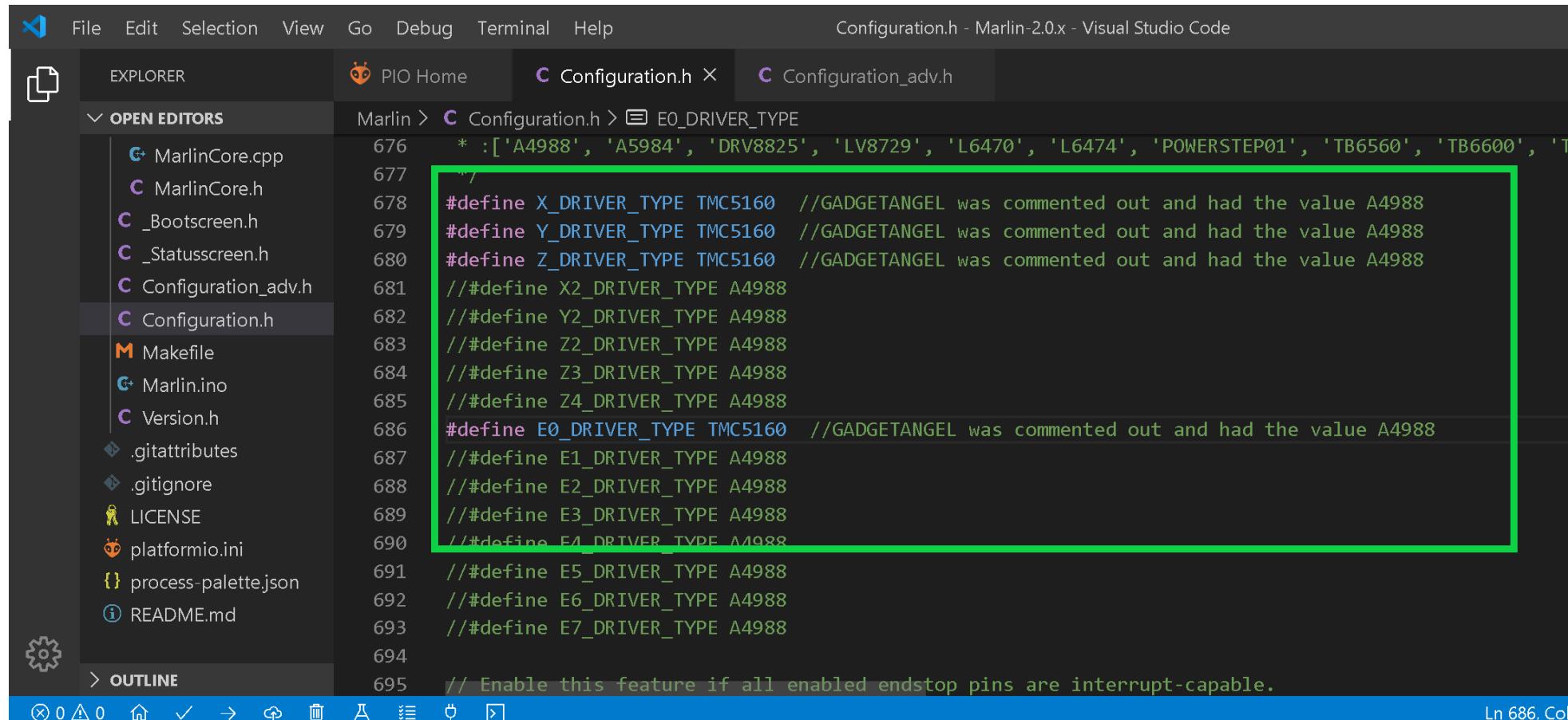
Sensor-less homing for X,  
Y and Z axes.



## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

**NOTE:** Go to Appendix C, and then come back here for the changes to Marlin for BIQU TMC5161 V1.0 stepper motor drivers in SPI mode.

- Change the stepper motor drivers so that Marlin knows you are using TMC5161 drivers in SPI mode. Change one line for each axis and one for each extruder you will be using. See the picture below for an example of how to use TMC5161 drivers in SPI mode. When two "://" appear at the beginning of a line that means that line is commented out. To un-comment a line just remove the leading two forward slashes ("//").



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home Configuration.h X Configuration_adv.h
Marlin > Configuration.h > E0_DRIVER_TYPE
676 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'T
677 "
678 #define X_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
679 #define Y_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
680 #define Z_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
681 //#define X2_DRIVER_TYPE A4988
682 //#define Y2_DRIVER_TYPE A4988
683 //#define Z2_DRIVER_TYPE A4988
684 //#define Z3_DRIVER_TYPE A4988
685 //#define Z4_DRIVER_TYPE A4988
686 #define E0_DRIVER_TYPE TMC5160 //GADGETANGEL was commented out and had the value A4988
687 //#define E1_DRIVER_TYPE A4988
688 //#define E2_DRIVER_TYPE A4988
689 //#define E3_DRIVER_TYPE A4988
690 //#define E4_DRIVER_TYPE A4988
691 //#define E5_DRIVER_TYPE A4988
692 //#define E6_DRIVER_TYPE A4988
693 //#define E7_DRIVER_TYPE A4988
694
695 // Enable this feature if all enabled endstop pins are interrupt-capable.
Ln 686, Col 1

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Next you want to set your  $V_{ref}$  in the Marlin firmware for each axis that has the TMC5161 driver, as seen in the **GREEN** box below. I changed the "X\_CURRENT" to be the calculated  $V_{ref}$  for my X-Axis, which is 760mV for an Ender 3. I changed the "Y\_CURRENT" to be the calculated  $V_{ref}$  for my Y-Axis, which is 760mV on the Ender 3.
- Ensure "X\_RSENSE" is set to 0.062. Ensure "Y\_RSENSE" is set to 0.062.
- If you **do not want to use  $V_{ref}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT", you should **use  $I_{RMS}$  instead**. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "X\_CURRENT" and/or "Y\_CURRENT".

```

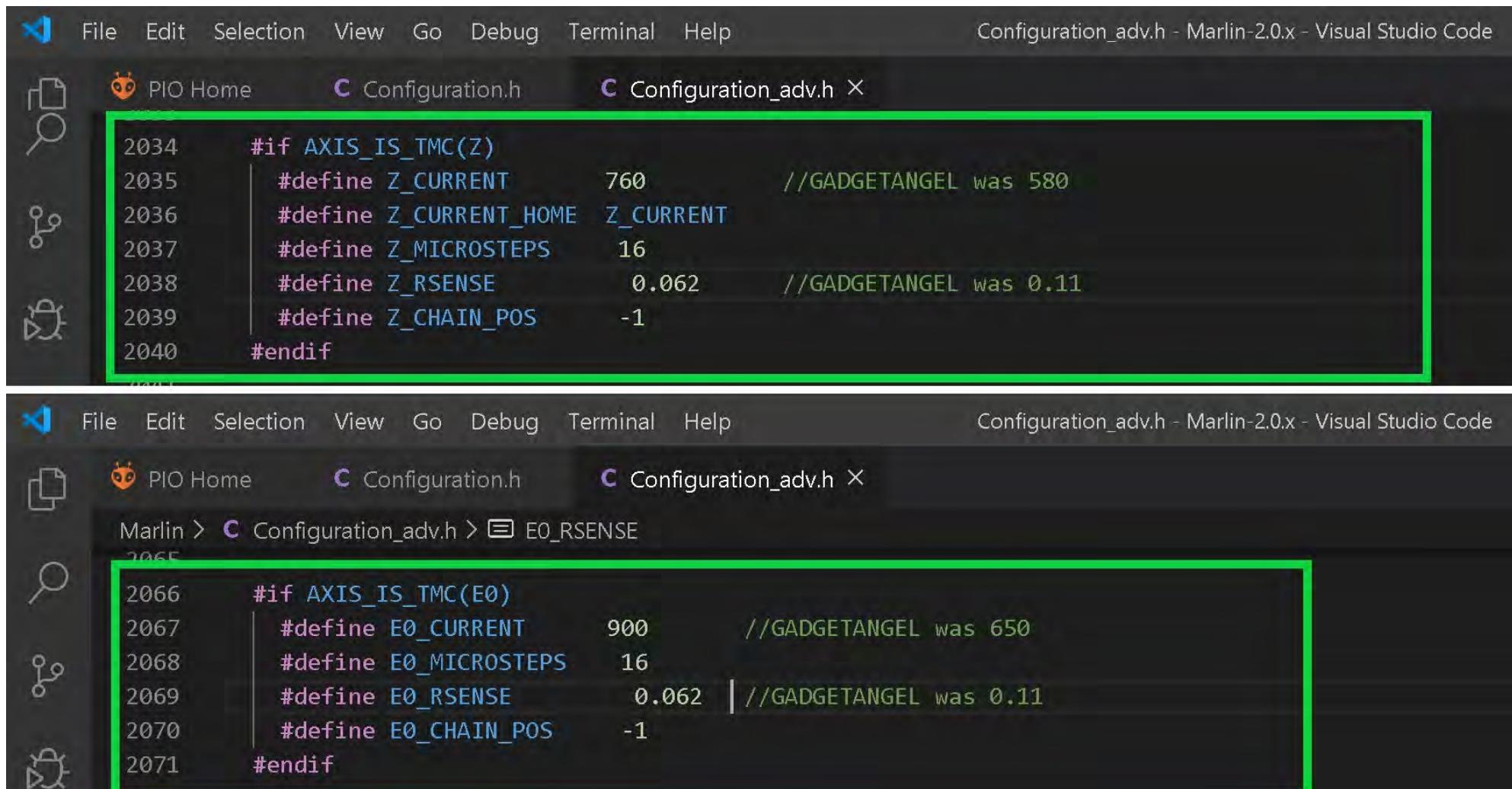
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
PIO Home Configuration.h Configuration_adv.h X
Marlin > C Configuration_adv.h > EO_RSENSE
1990 * To use the reading capabilities, also connect #_SERIAL_RX_PIN to PDN_UART without
1991 * a resistor.
1992 * The drivers can also be used with hardware serial.
1993 *
1994 * TMCStepper library is required to use TMC stepper drivers.
1995 * https://github.com/teemuatlut/TMCStepper
1996 */
1997 #if HAS_TRINAMIC
1998
1999 #define HOLD_MULTIPLIER 0.5 // Scales down the holding current from run current
2000 #define INTERPOLATE true // Interpolate X/Y/Z_MICROSTEPS to 256
2001
2002 #if AXIS_IS_TMC(X)
2003 #define X_CURRENT 760 // (mA) RMS current. Multiply by 1.414 for peak current. //GADGETANGEL was 580
2004 #define X_CURRENT_HOME X_CURRENT // (mA) RMS current for sensorless homing
2005 #define X_MICROSTEPS 16 // 0..256
2006 #define X_RSENSE 0.062 //GADGETANGEL was 0.11
2007 #define X_CHAIN_POS -1 // <=0 : Not chained, 1 : MCU MOSI connected, 2 : Next in chain, ...
2008 #endif
2009
2010 #if AXIS_IS_TMC(X2)
2011 #define X2_CURRENT 800
2012 #define X2_CURRENT_HOME X2_CURRENT
2013 #define X2_MICROSTEPS 16
2014 #define X2_RSENSE 0.11
2015 #define X2_CHAIN_POS -1
2016 #endif
2017
2018 #if AXIS_IS_TMC(Y)
2019 #define Y_CURRENT 760 //GADGETANGEL was 580
2020 #define Y_CURRENT_HOME Y_CURRENT
2021 #define Y_MICROSTEPS 16
2022 #define Y_RSENSE 0.062 //GADGETANGEL was 0.11
2023 #define Y_CHAIN_POS -1
2024 #endif

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Now, I am setting the  $V_{ref}$  for Z-Axis and the extruder, as seen in the **GREEN** boxes below. I changed the "Z\_CURRENT" to be the calculated  $V_{ref}$  for my Z-Axis, which is 760mV for an Ender 3. I changed the "E0\_CURRENT" to be the calculated  $V_{ref}$  for my Extruder, which is 900mV on the Ender 3.
- Ensure "Z\_RSENSE" is set to 0.062. Ensure "E0\_RSENSE" is set to 0.062.
- If you **do not want to use  $V_{ref}$**  as the value for "Z\_CURRENT" and/or "E0\_CURRENT", you should use  $I_{RMS}$  instead. You find  $I_{RMS}$  by taking  $I_{MAX}$  and dividing it by 1.414 ( $I_{RMS}=I_{MAX}/1.414$ ). You use **50% to 90% of the calculated  $I_{RMS}$**  as the value for "Z\_CURRENT" and/or "E0\_CURRENT".



```

2034 #if AXIS_IS_TMC(Z)
2035 #define Z_CURRENT 760 //GADGETANGEL was 580
2036 #define Z_CURRENT_HOME Z_CURRENT
2037 #define Z_MICROSTEPS 16
2038 #define Z_RSENSE 0.062 //GADGETANGEL was 0.11
2039 #define Z_CHAIN_POS -1
2040 #endif

```

```

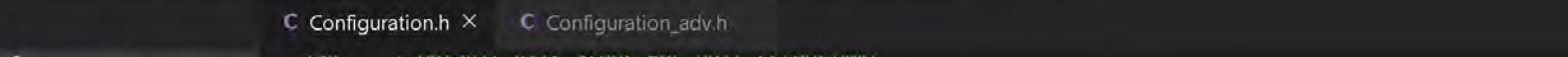
2066 #if AXIS_IS_TMC(E0)
2067 #define E0_CURRENT 900 //GADGETANGEL was 650
2068 #define E0_MICROSTEPS 16
2069 #define E0_RSENSE 0.062 //GADGETANGEL was 0.11
2070 #define E0_CHAIN_POS -1
2071 #endif

```

- Go to the next page.

The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- If you changed the "MICROSTEPS" for any of the axes then you will need to update "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to reflect your changes

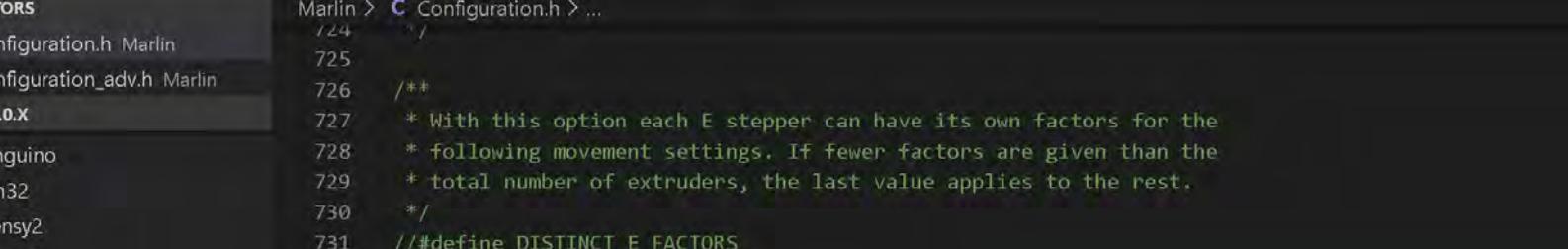


The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Configuration.h Marlin 2.0.x Visual Studio Code
- Explorer View:** Shows the project structure under MARLIN-2.0.X:
  - > sanguino
  - > stm32
  - > teensy2
  - > teensy3
  - C pins.h
- Editor View:** The Configuration.h file is open. A specific line of code is highlighted with a green box:

```
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 } //GADGETANGEL was
 // {80, 80, 400, 93} for A4988 on Ender 3|
```

- FOR EXAMPLE if you wanted to use 1/32 stepping instead of the default 1/16, you would be **doubling** your STEPS. Therefore, **we must adjust our "DEFAULT\_AXIS\_STEPS\_PER\_UNIT"** anytime our STEPS are **NOT 1/16**. So change "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" to {160, 160, 800, 186}, as seen in the **GREEN** box below.



```
File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration_adv.h

Marlin > Configuration.h > ...
724 /*
725
726 /**
727 * With this option each E stepper can have its own factors for the
728 * following movement settings. If fewer factors are given than the
729 * total number of extruders, the last value applies to the rest.
730 */
731 // #define DISTINCT_E_FACTORS
732 /**
733 *
734 * Default Axis Steps Per Unit (steps/mm)
735 * Override with M92
736 *
737 */
738 #define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 } // GADGETANGEL was
739 // {80, 80, 400, 93} for A4988 on Ender 3
740 // Double because we are going
741 // to 1/32 from 1/16
742
743 */


```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- We need to uncomment out the "TMC\_USE\_SW\_SPI" because the SKR V1.3 pins file depends on this variable to define its SPI pins

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_USE_SW_SPI
 Configuration.h Marlin 2144 * The default SW SPI pins are defined the respective pins files,
 Configuration_adv.h Marlin 2145 * but you can override or define them here.
 2146 */
 2147 #define TMC_USE_SW_SPI //GADGETANGEL was commented out
 2148 // #define TMC_SW_MOST -1

```

- By default stealthChop is enabled in the Marlin firmware. If you want spreadCycle ONLY then comment out the appropriate lines. I want **stealthChop enabled** so I want to make sure the lines are not commented out {"STEALTHCHOP\_XY", "STEALTHCHOP\_Z" and "STEALTHCHOP\_E"}. You also want to check to see if the proper "CHOPPER\_TIMING" is set for your printer. An Ender 3 is a 24VDC printer, my "CHOPPER\_TIMING" is correct.

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

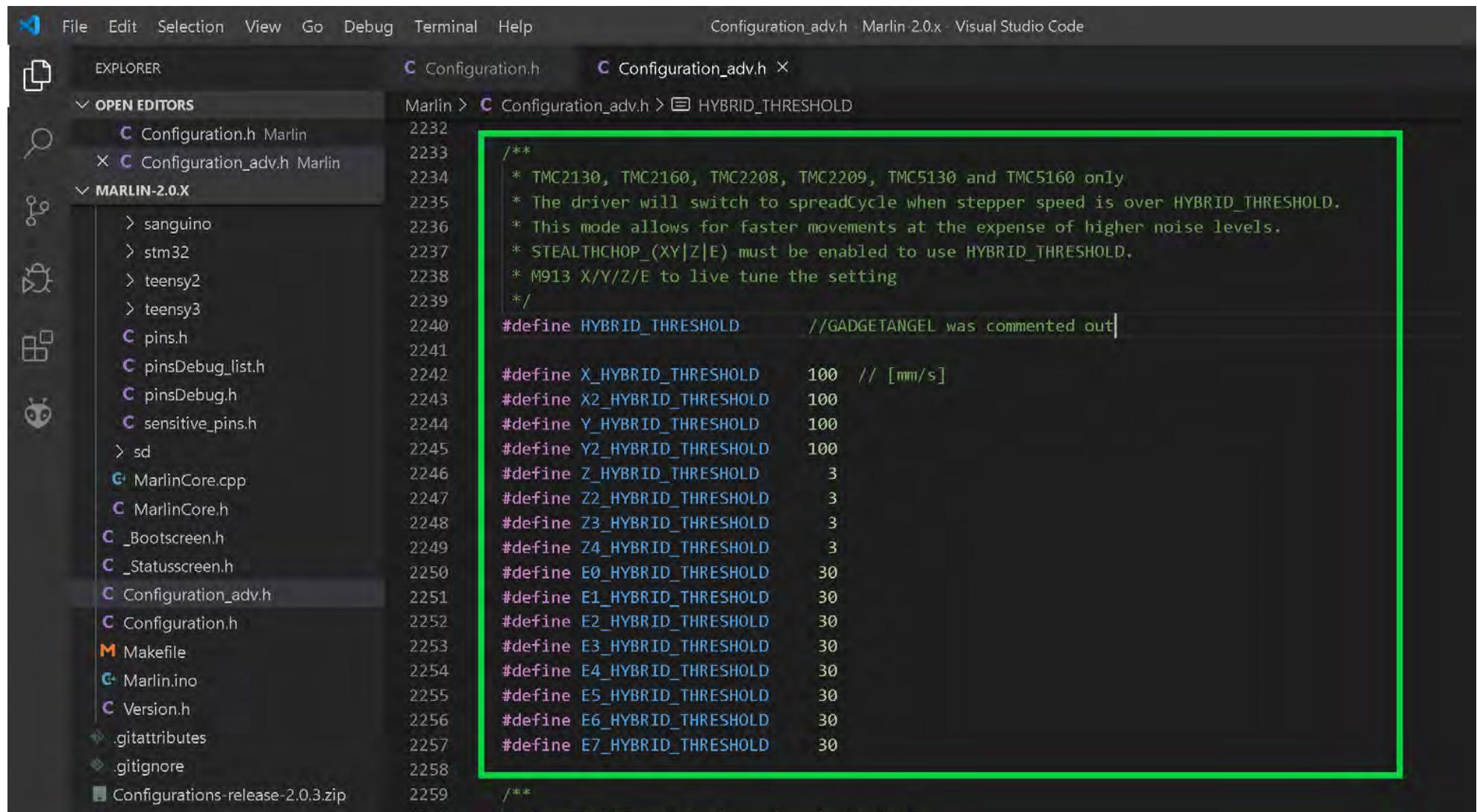
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > STEALTHCHOP_XY
 Configuration.h Marlin 2193 */
 Configuration_adv.h Marlin 2194 #define STEALTHCHOP_XY
 2195 #define STEALTHCHOP_Z
 2196 #define STEALTHCHOP_E
 2197
 2198 */
 2199 * Optimize spreadCycle chopper parameters by using predefined parameter sets
 2200 * or with the help of an example included in the library.
 2201 * Provided parameter sets are
 2202 * CHOPPER_DEFAULT_12V
 2203 * CHOPPER_DEFAULT_19V
 2204 * CHOPPER_DEFAULT_24V
 2205 * CHOPPER_DEFAULT_36V
 2206 * CHOPPER_PRUSAMK3_24V // Imported parameters from the official Prusa firmware for MK3 (24V)
 2207 * CHOPPER_MARLIN_119 // Old defaults from Marlin v1.1.9
 2208 *
 2209 * Define your own with
 2210 * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
 2211 */
 2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
 2213 */

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Now you either enable "HYBRID\_THRESHOLD" or disable it. By default, it is disabled. "HYBRID\_THRESHOLD" allows the printer to change between stealthChop and spreadCycle dynamically depending on the print speed. I want "HYBRID\_THRESHOLD" enabled so I need to remove the two leading "//", which uncomments the line in the Marlin firmware.



```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin 2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > HYBRID_THRESHOLD
2232
2233 /**
2234 * TMC2130, TMC2160, TMC2208, TMC2209, TMC5130 and TMC5160 only
2235 * The driver will switch to spreadCycle when stepper speed is over HYBRID_THRESHOLD.
2236 * This mode allows for faster movements at the expense of higher noise levels.
2237 * STEALTHCHOP_(XY|Z|E) must be enabled to use HYBRID_THRESHOLD.
2238 * M913 X/Y/Z/E to live tune the setting
2239 */
2240 #define HYBRID_THRESHOLD //GADGETANGEL was commented out
2241
2242 #define X_HYBRID_THRESHOLD 100 // [mm/s]
2243 #define X2_HYBRID_THRESHOLD 100
2244 #define Y_HYBRID_THRESHOLD 100
2245 #define Y2_HYBRID_THRESHOLD 100
2246 #define Z_HYBRID_THRESHOLD 3
2247 #define Z2_HYBRID_THRESHOLD 3
2248 #define Z3_HYBRID_THRESHOLD 3
2249 #define Z4_HYBRID_THRESHOLD 3
2250 #define E0_HYBRID_THRESHOLD 30
2251 #define E1_HYBRID_THRESHOLD 30
2252 #define E2_HYBRID_THRESHOLD 30
2253 #define E3_HYBRID_THRESHOLD 30
2254 #define E4_HYBRID_THRESHOLD 30
2255 #define E5_HYBRID_THRESHOLD 30
2256 #define E6_HYBRID_THRESHOLD 30
2257 #define E7_HYBRID_THRESHOLD 30
2258
2259 /**
2260 * H S T C D L C Y Z E

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Now I want to enable some statements that allow me access to debugging the TMC drivers. I will uncomment "MONITOR\_DRIVER\_STATUS" and "TMC\_DEBUG". "MONITOR\_DRIVER\_STATUS" will enable the following G-codes: M906, M911, and M912, "TMC\_DEBUG" will enable the M122 G-code command. You can read about these from the comments in the firmware and in [Marlin's documentation located on-line.](#)

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > MONITOR_DRIVER_STATUS
 C Configuration.h Marlin
 X C Configuration_adv.h Marlin
MARLIN-2.0.X
 > sanguino
 > stm32
 > teensy2
 > teensy3
 C pins.h
 C pinsDebug_list.h
 C pinsDebug.h
 C sensitive_pins.h
 > sd
 C MarlinCore.cpp
 C MarlinCore.h
 C Bootscreen.h
2211 */
2212 #define CHOPPER_TIMING CHOPPER_DEFAULT_24V
2213
2214 /**
2215 * Monitor Trinamic drivers for error conditions,
2216 * like overtemperature and short to ground.
2217 * In the case of overtemperature Marlin can decrease the driver current until error condition clears.
2218 * Other detected conditions can be used to stop the current print.
2219 * Relevant g-codes:
2220 * M906 - Set or get motor current in millamps using axis codes X, Y, Z, E. Report values if no axis codes given.
2221 * M911 - Report stepper driver overtemperature pre-warn condition.
2222 * M912 - Clear stepper driver overtemperature pre-warn condition flag.
2223 * M122 - Report driver parameters (Requires TMC_DEBUG)
2224 */
2225 #define MONITOR_DRIVER_STATUS //GADGETANGEL was commented out
2226
2227 #if ENABLED(MONITOR_DRIVER_STATUS)

```

```

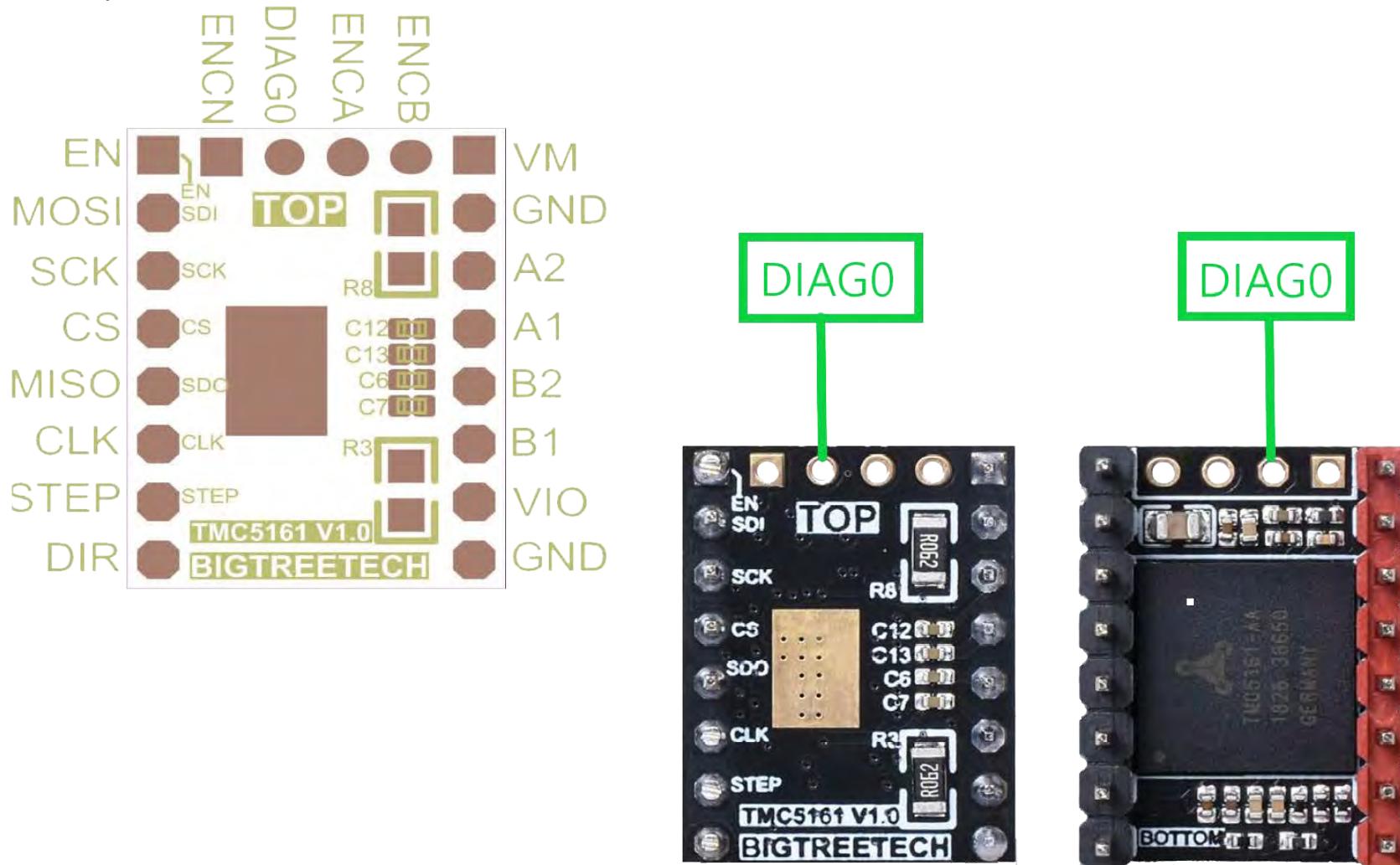
File Edit Selection View Go Debug Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > TMC_DEBUG
 C Configuration.h Marlin
 X C Configuration_adv.h Marlin
MARLIN-2.0.X
 > sanguino
 > stm32
 > teensy2
2307 */
2308
2309 /**
2310 * Enable M122 debugging command for TMC stepper drivers.
2311 * M122 S0/1 will enable continuous reporting.
2312 */
2313 #define TMC_DEBUG //GADGETANGEL was commented out

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

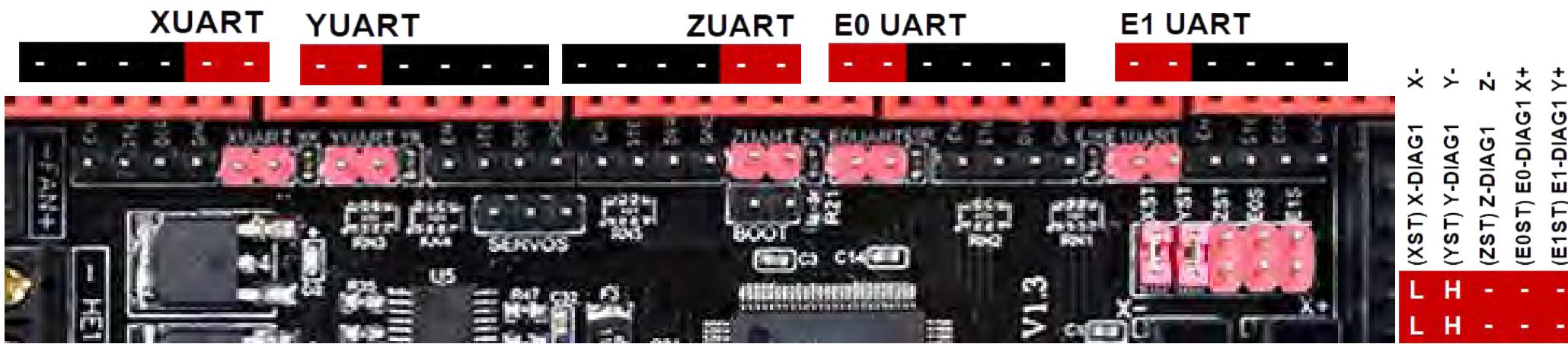
- This next section covers sensor-less homing which is available for the TMC5161 in SPI mode. I want to enable it BUT for the TMC5161 I first have to solder on the DIAG0 pin onto each TMC5161 driver that will be on an axis with sensor-less homing enabled. Therefore, I want sensor-less homing for X and Y axes only. So I need to solder in a DIAG0 pin for two TMC5161 drivers. Here is a picture of the TMC5161 V1.0 pin-out.



- The third pin position starting from the left on the top of the chip is where I need to solder in a header pin. I need it to face down so when I plug in the TMC5161 into the SKR V1.3 board the DIAG0 pin will be seated in the SKR V1.3 board. Then I can use the XST jumper to connect the DIAG0 pin to the X endstop. I can then, also, use the YST jumper to connect the DIAG0 pin to the Y endstop. Connecting the DIAG0 pins to the endstops enables sensor-less homing to work.
- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- I will be covering sensor-less homing for the X and Y axis only. I will not be using sensor-less homing on my Z axis on my Ender 3 printer. For sensor-less homing to work the DIAG0 pin on the TMC5161 driver has to be plugged into the SKR V1.3 board. Since I am not using sensor-less homing on my Z axis I will need to ensure that my DIAG0 pin on the Z axis TMC5161 is NOT connected to Z axis endstop on the SKR 1.3 board. I want X axis endstop to be connected to the DIAG0 pin of the TMC5161 for the X axis. Also, I want the Y axis endstop to be connected to the DIAG0 pin of the TMC5161 for the Y axis. Therefore my jumpers for XST, YST, ZST, EOS, E1S, XUART, YUART, ZUART, E0UART and E1UART will look like the picture below. I will ensure that the only two jumpers that are in place are the "L" and "H" jumpers.



- Sensor-less homing is commented out by default. So I remove the two leading "//" to un-comment "SENSORLESS\_HOMING"

```

File Edit Selection View Go Debug Terminal Help Configuration_adv.h Marlin 2.0.x Visual Studio Code

EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration_adv.h > SENSORLESS_HOMING
Configuration.h Marlin
Configuration_adv.h Marlin
MARLIN-2.0.X

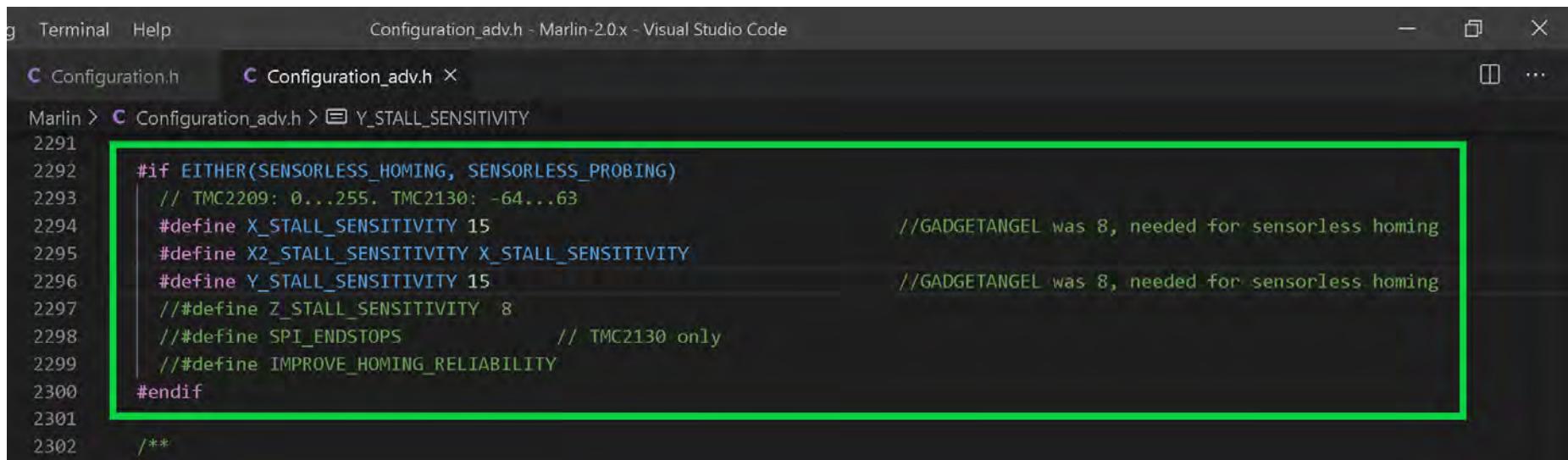
2281 */
2282 #define SENSORLESS_HOMING // StallGuard capable drivers only //GADGETANGEL was commented out
2283

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Next we set the "starting" stall sensitivity for sensor-less homing. I choose to make it 15. If the stall sensitivity is too high your motor will grind and not stop when it hits the end of travel on the axis. If the stall sensitivity is too low then the motor will barely move because it thinks it has hit the end of travel for the axis. Notice I only uncommented the "X\_STALL\_SENSITIVITY" and the "Y\_STALL\_SENSITIVITY". If you want sensor-less homing on the Z axis, then you will have to uncomment "Z\_STALL\_SENSITIVITY".

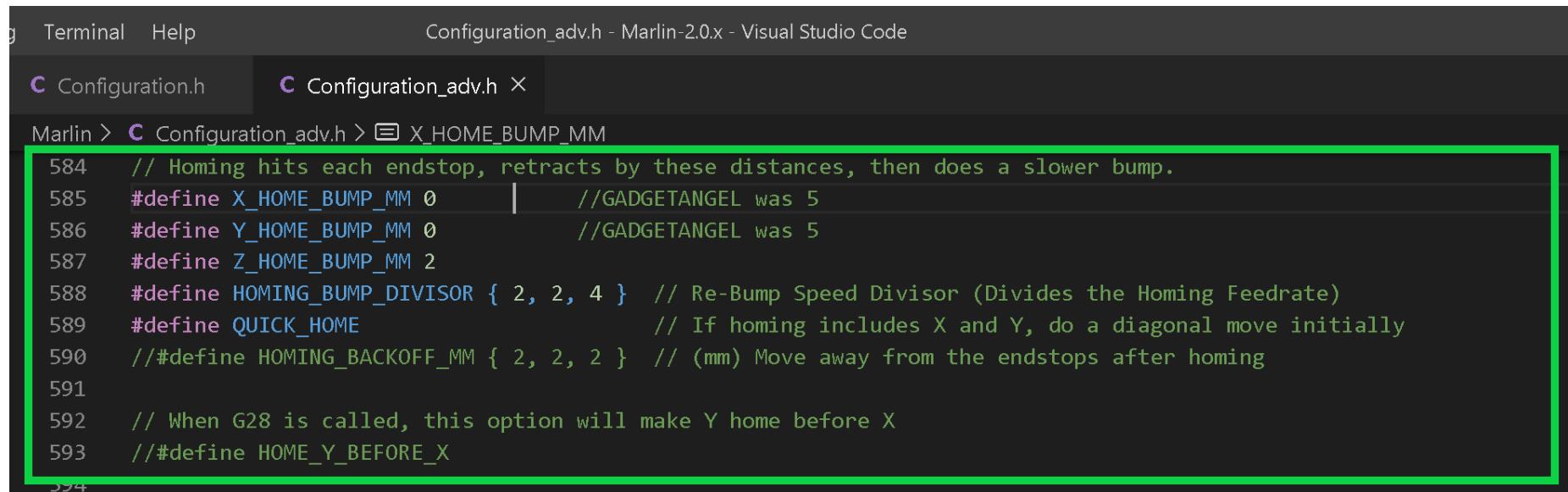


```

g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > Y_STALL_SENSITIVITY
2291
2292 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
2293 // TMC2209: 0...255. TMC2130: -64...63
2294 #define X_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2295 #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
2296 #define Y_STALL_SENSITIVITY 15 //GADGETANGEL was 8, needed for sensorless homing
2297 //#define Z_STALL_SENSITIVITY 8
2298 //">#define SPI_ENDSTOPS // TMC2130 only
2299 //">#define IMPROVE_HOMING_RELIABILITY
2300 #endif
2301
2302 /**

```

- We now have to set our home bump to 0 for each axis with sensor-less homing enabled. So I will set "X\_HOME\_BUMP\_MM" to 0 and "Y\_HOME\_BUMP\_MM" to 0. If you want sensor-less homing on Z axis then you will need to set "Z\_HOME\_BUMP\_MM" to 0.



```

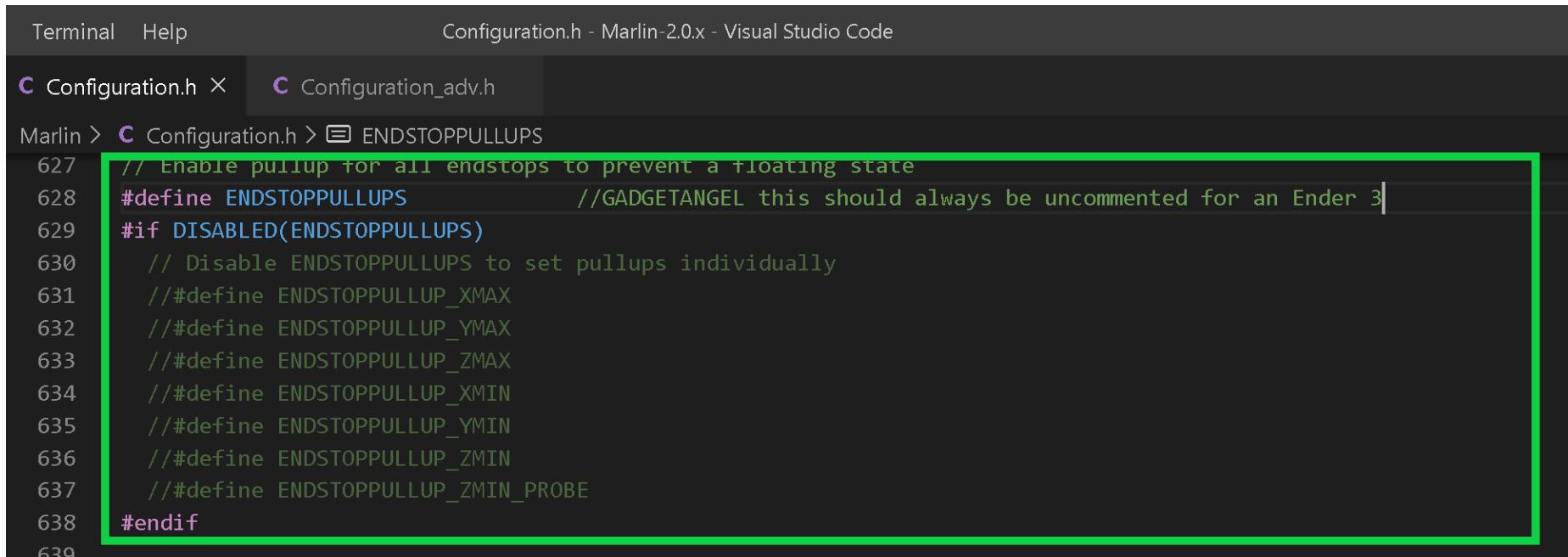
g Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > X_HOME_BUMP_MM
584 // Homing hits each endstop, retracts by these distances, then does a slower bump.
585 #define X_HOME_BUMP_MM 0 //GADGETANGEL was 5
586 #define Y_HOME_BUMP_MM 0 //GADGETANGEL was 5
587 #define Z_HOME_BUMP_MM 2
588 #define HOMING_BUMP_DIVISOR { 2, 2, 4 } // Re-Bump Speed Divisor (Divides the Homing Feedrate)
589 #define QUICK_HOME // If homing includes X and Y, do a diagonal move initially
590 //">#define HOMING_BACKOFF_MM { 2, 2, 2 } // (mm) Move away from the endstops after homing
591
592 // When G28 is called, this option will make Y home before X
593 //">#define HOME_Y_BEFORE_X
594

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- Let's check the firmware to ensure that "ENDSTOPPULLUPS" is enabled. It is by default.



```

Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

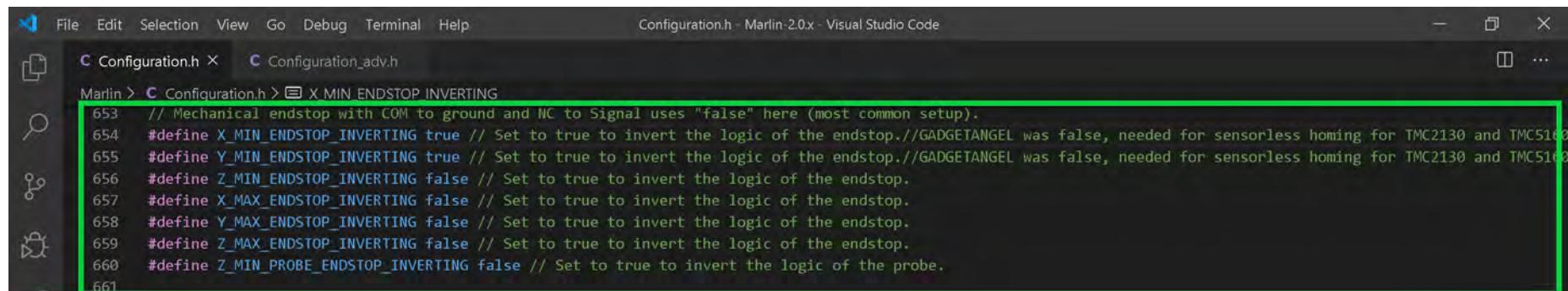
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > ENDSTOPPULLUPS

627 // Enable pullup for all endstops to prevent a floating state
628 #define ENDSTOPPULLUPS //GADGETANGEL this should always be uncommented for an Ender 3
629 #if DISABLED(ENDSTOPPULLUPS)
630 // Disable ENDSTOPPULLUPS to set pullups individually
631 //#define ENDSTOPPULLUP_XMAX
632 //#define ENDSTOPPULLUP_YMAX
633 //#define ENDSTOPPULLUP_ZMAX
634 //#define ENDSTOPPULLUP_XMIN
635 //#define ENDSTOPPULLUP_YMIN
636 //#define ENDSTOPPULLUP_ZMIN
637 //#define ENDSTOPPULLUP_ZMIN_PROBE
638 #endif
639

```

- Next to allow sensor-less homing to work (while using the BIQU TMC5161) we need to change our end stop logic. Therefore I set "X\_MIN\_ENDSTOP\_INVERTING" to true and "Y\_MIN\_ENSTOP\_INVERTING" to true. If you want sensor-less homing on the Z axis, you will need to set "Z\_MIN\_ENDSTOP\_INVERTING" to true. But since I do not want sensor-less homing on the Z axis I will leave "Z\_MIN\_ENDSTOP\_INVERTING" set to false.



```

File Edit Selection View Go Debug Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
C Configuration.h X C Configuration_adv.h

Marlin > C Configuration.h > X MIN ENDSTOP INVERTING

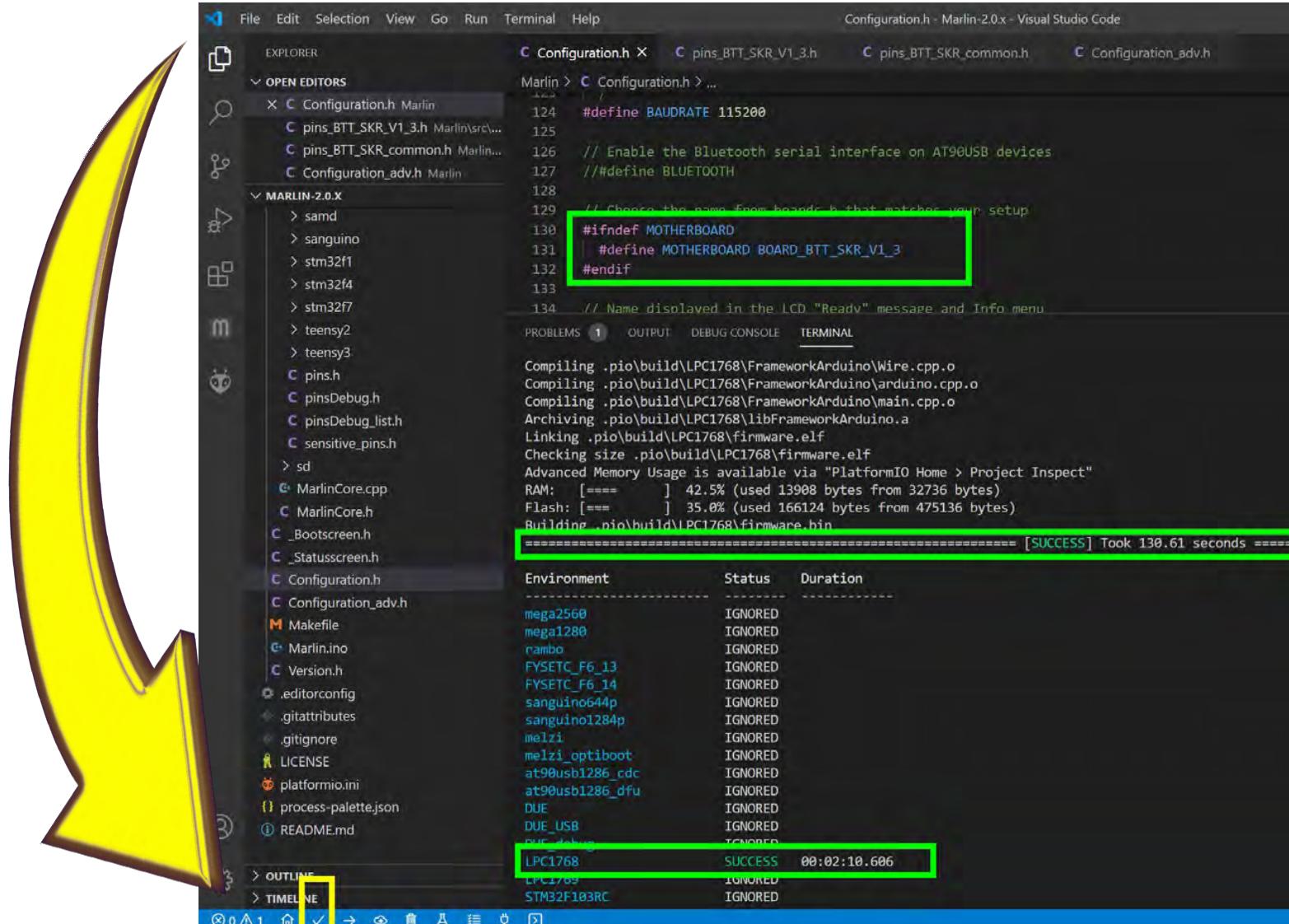
653 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
654 #define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
655 #define Y_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of the endstop.//GADGETANGEL was false, needed for sensorless homing for TMC2130 and TMC5161
656 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
657 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
658 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
659 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
660 #define Z_MIN_PROBE_ENDSTOP_INVERTING false // Set to true to invert the logic of the probe.
661

```

- Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- The end of Marlin setup for BIQU TMC5161 drivers in SPI mode. Now, compile your code by clicking on the check mark, as seen in the **YELLOW** box below. Once the firmware has compiled successfully you will see the following messages, as shown in the **GREEN** boxes below.



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows files in the project structure, including Configuration.h, pins\_BTT\_SKR\_V1\_3.h, pins\_BTT\_SKR\_common.h, Configuration\_adv.h, and various Marlin core files like MarlinCore.cpp and MarlinCore.h.
- TERMINAL:** Displays the compilation command and its output. A yellow box highlights the checkmark icon at the bottom left of the terminal window, indicating the build step. A green box highlights the successful compilation message: `[SUCCESS] Took 130.61 seconds =====`.
- OUTPUT:** Shows the build progress and memory usage. A green box highlights the success status: `LPC1768 SUCCESS 00:02:10.606`.

- With the micro SD card loaded into the SKR V1.3 board's micro SD card reader, you can compile and upload the "firmware.bin" file in one step via the VS code window. Go to the next page.

## The (latest release of) Marlin Setup for BIQU TMC5161 V1.0 Drivers in SPI Mode

- To compile **and** upload the "firmware.bin" file to the SKR V1.3 board you click the arrow, located at the bottom of the VS code window; as seen in the **RED box** below.

The screenshot shows the Visual Studio Code interface for Marlin 2.0.x. The Explorer sidebar on the left lists various Marlin source files. The main editor window displays Configuration.h with the following code snippet highlighted by a green box:

```

#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_BTT_SKR_V1_3
#endif

```

The status bar at the bottom right shows the build log: [SUCCESS] Took 130.61 seconds. Below the status bar, a table shows build results for different boards:

| Environment     | Status         | Duration            |
|-----------------|----------------|---------------------|
| mega2560        | IGNORED        |                     |
| mega1280        | IGNORED        |                     |
| rambo           | IGNORED        |                     |
| FYSETC_F6_13    | IGNORED        |                     |
| FYSETC_F6_14    | IGNORED        |                     |
| sanguino644p    | IGNORED        |                     |
| sanguino1284p   | IGNORED        |                     |
| melzi           | IGNORED        |                     |
| melzi_optiboot  | IGNORED        |                     |
| at90usb1286_cdc | IGNORED        |                     |
| at90usb1286_dfu | IGNORED        |                     |
| DUE             | IGNORED        |                     |
| DUE_USB         | IGNORED        |                     |
| SUE_L16         | IGNORED        |                     |
| <b>LPC1768</b>  | <b>SUCCESS</b> | <b>00:02:10.606</b> |
| LPC1709         | IGNORED        |                     |
| STM32F103RC     | IGNORED        |                     |

- If the upload (as seen in the **RED box** above) button does not work, you will have to find the "firmware.bin" file and manually load it onto the micro SD card. Once it has been compiled via VS code with Platformio, go to [Appendix E](#) to learn how to find the "firmware.bin" file for the SKR V1.3 board.

## How to adjust the $V_{ref}$ on a Stepper Motor Driver board using the Potentiometer<sup>1,2</sup>

Modern 3D printers usually use NEMA17 motors. The first piece of information you will need is the "Rated Current" of your NEMA17 motor. So, write down your motor's part number and pay attention to the Step Angle, Holding Torque, Rated Current, Voltage, and Inductance.

Use the Step Angle to work out your printer's "DEFAULT\_AXIS\_STEPS\_PER\_UNIT" for your firmware with:

1. <http://www.prusaprinters.org/calculator/>

## How to Tune Stepper Motor Drivers<sup>2</sup>

1. Turn power off your printer, unplug the stepper motor cables, turn power back on your printer and tune the stepper motor drivers that are already plugged into the SKR V1.3 board.
2. When done, turn power off the printer, plug in the stepper motor cables turn power back on your printer and test motor movement.

**NOTE:** Don't tune stepper motor drivers with the motors plugged in, if you accidentally set current too high you can fry the motor or the stepper motor driver.

**NOTE:** Don't plug or unplug stepper motors with the power on the printer (i.e. power on the SKR V1.3 board)

Measure DC voltage between the stepper motor driver's trimpot (POT) or " $V_{ref}$  Test point" and your PSU's (12VDC/24VDC) ground. The ground at the PSU connector to the SKR V1.3 board is fine to use. Look up the correct current for your motor part number. If you have motors with no part number, assume they have a max of 1.00 amps ( $I_{MAX}$ ) to be safe. Look up the proper formula for your stepper motor drivers (as show in this document), and find the voltage (but ONLY use 90% of the calculated  $V_{ref}$ ) which corresponds with the current you want to set. Use a ceramic screw driver to adjust the POT. A ceramic screw driver is nonconductive and if you slip while making the adjustment to the POT you could short circuit the stepper motor driver board (i.e. KILL the driver board) in the process.

Time Saver tip, but more dangerous: Get slip-on alligator clips for your multimeter. Clamp ground to a 12VDC/24VDC ground (PSU Ground) wire and clamp positive to your plastic handle screwdriver. This way you'll measure the voltage as you adjust ("live adjustment") and don't need three hands.

**Note:** See the next page for a diagram of the setup.

<sup>1</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers> and

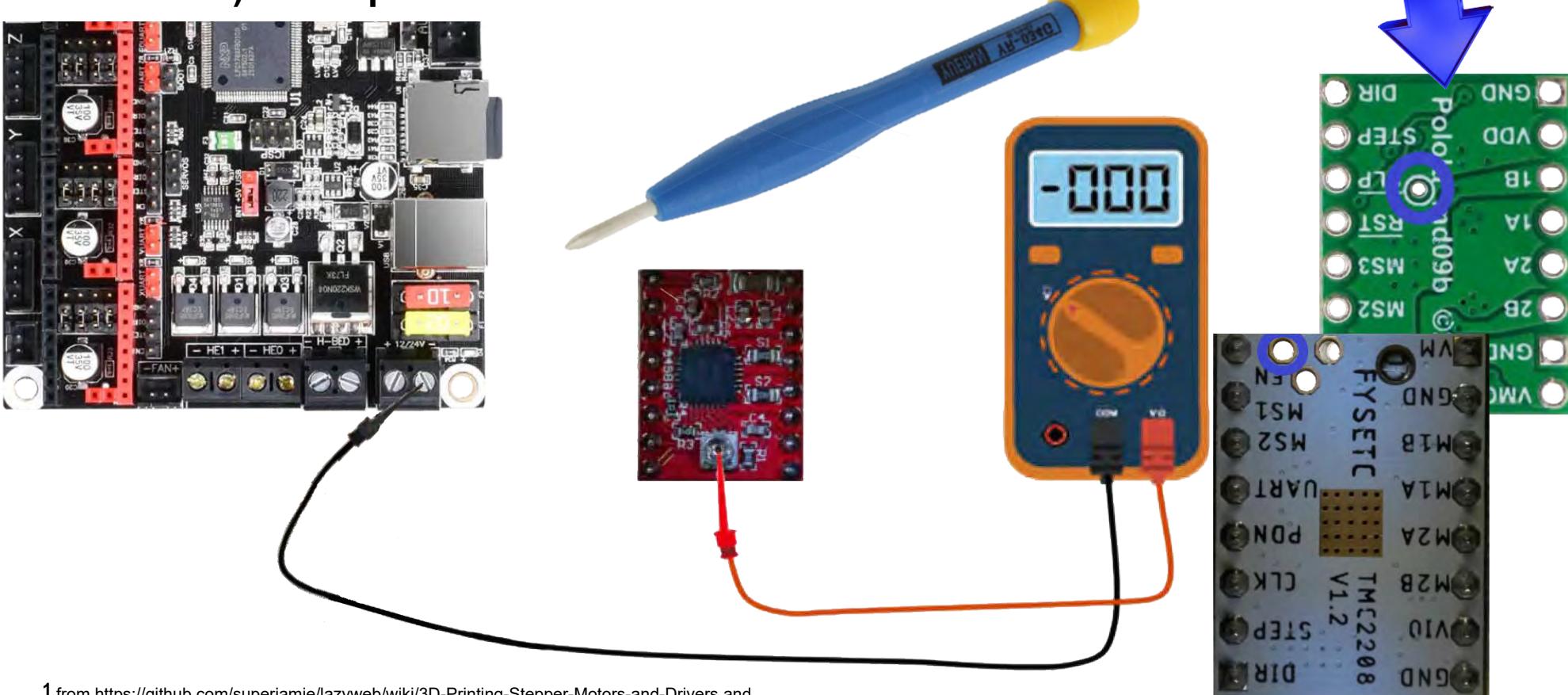
<sup>2</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

## How to adjust the $V_{ref}$ on a Stepper Motor Driver board using the Potentiometer<sup>1, 2</sup>

**Note:** A ceramic screw driver is non-conductive. If you use a plastic handle screw driver with alligator clips to your multimeter to make a "live adjustment" of  $V_{ref}$  (and you slip while making the adjustment to the POT), you could short circuit the stepper motor driver board (i.e. KILL the board) in the process.

**Note:** Some stepper motor driver boards have a " $V_{ref}$  Test point" location, shown in **BLUE**. Check bottom or top of your board for a location.

If a " $V_{ref}$  Test point" location is not available, use the potentiometer on the stepper driver board to do the  $V_{ref}$  measurement.



<sup>1</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers>

<sup>2</sup> from <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>

## APPENDIX B

### **For the TMC drivers what's the difference between stand-alone mode and ("UART" or "SPI ") modes?**

All the TMC driver chips **EXCEPT TMC5160 and TMC5161** have a stand-alone mode. **Stand-alone mode** allows you to just drop the driver into your motherboard to replace your A4988 driver. The **OTP mode** is similar, but you use some software to **PERMANENTLY** change the driver's mode of operation. The **UART or SPI modes** allow you to **dynamically** change the driver in your firmware.

### **How to Calculate $V_{ref}$ for Non-TMC Stepper Motor Drivers**

My machine is an Ender 3, the X, Y, Z stepper motor "Rated Current" is 0.84 Amps, while E (extruder) stepper motor "Rated Current" is 1 Amps.

We use the  $V_{ref}$  formula (found on the first page of each different stepper motor driver section of this document and do the  $V_{ref}$  calculation.

Remember this  $V_{ref}$  calculation is just a suggested starting point. If your stepper motors are running **too hot** you will need to adjust the  $V_{ref}$  **downward**. If your stepper motors are **skipping steps** when printing then you will need to adjust your  $V_{ref}$  **upwards**. Our goal is to find a low enough  $V_{ref}$  where our stepper motors are cool enough without the printer missing any steps. For this example, I will use the A4988 stepper motor driver.

1. So, A4988  $V_{ref}$  formula is  $V_{ref} = I_{MAX} * (8 * R_s)$ , where  $R_s = 0.1\Omega$ .
2. I take each of my Axis' "Rated Current" and plug it into that equation to get X-Axis  $V_{ref}$  is equal to  $(0.840 * 8 * 0.100) = 0.672$  volts or 672mV.
3. Now, take 90% of that for a starting point  $V_{ref}$  value of  $((0.672 * 0.90) = 0.6048)$  0.605 volts or 605mV or X-Axis  $V_{ref} = 0.605V$ . Since X, Y and Z stepper motors have the same "Rated Current" we now have the  $V_{ref}$  for X, Y and Z stepper motor drivers. Their value is 0.605 volts.
4. For E (extruder)  $V_{ref}$  the equation is  $(1.0 * 8 * 0.100) = 0.800$  volts. Now, take 90% of that, for a starting point,  $V_{ref}$  value of  $((0.800 * 0.90) = 0.720)$  0.720 volts or 720mV. We now have the  $V_{ref}$  for E (extruder stepper motor driver) which is 0.72 volts.
5. We use our multimeter and turn the POT on the top of the stepper motor driver until we see the wanted  $V_{ref}$  voltage displayed.

## APPENDIX B

### How to Calculate $V_{ref}$ for TMC Stepper Motor Drivers

My machine is an Ender 3. The X, Y, and Z stepper motors have a "Rated Current" of 0.84 Amps, while E (extruder) stepper motor "Rated Current" is 1 Amp.

We use the  $I_{MAX}$  formula and use Algebra to find the  $V_{ref}$  formula. The  $I_{MAX}$  formula for each TMC Driver is listed on the following pages. But for this example we will use TMC2100 drivers, so we will use the below equation:

TMC2100 Stand-alone Mode, (with  $R_s = 110m\Omega$ ):

$$I_{RMS} = ((V_{ref} / 2.5) * (1 / 1.41) * ((320mV / (R_s + 20m\Omega)) ))$$

Since  $I_{MAX} = (I_{RMS} * 1.41)$  is a known value then the above equation can be written as follows:

$$(I_{MAX} * (1 / 1.41)) = ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / (110 + 20))))$$

Since  $(1 / 1.41)$  is on both sides of the Algebra equation they cancel each other out leaving the equation as follows:

$$I_{MAX} = (V_{ref} / 2.5) * (2.46 \approx 2.5),$$

Therefore  $I_{MAX} = V_{ref}$ , and  $V_{ref} = I_{MAX}$ .

We use 50% to 90% of  $V_{ref}$  (i.e.  $I_{MAX}$ ) to set the current limit for TMC stepper motor driver. To take 90% we do the following:

Since  $V_{ref} = I_{MAX}$  we will use  $I_{MAX}$  instead of  $V_{ref}$ .

90% of  $V_{ref}$  for Ender 3's X, Y or Z =  $(I_{MAX} * 0.90) = (0.84 * 0.90) = 0.756$  or 756mA. Since  $I_{MAX} = V_{ref}$ , then it's also equal to 756mV.

Remember this  $V_{ref}$  calculation is just a suggested starting point. If your stepper motors are running **too hot** you will need to adjust the  $V_{ref}$  **downward**. If your stepper motors are **skipping steps** when printing then you will need to adjust your  $V_{ref}$  **upwards**. Our goal is to find a low enough  $V_{ref}$  where our stepper motors are cool enough without the printer missing any steps.

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**1. TMC2100 with  $R_s = 0.110\Omega$  (110m $\Omega$ ) :**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320mV / (R_s + 20m\Omega)) )) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((320 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.46)) \\ &= (1 / 1.41) * V_{ref} * 0.99 \\ &= (1 / 1.41) * V_{ref} * 1 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

**2. TMC2130 with  $R_s = 0.110\Omega$  (110m $\Omega$ ) :**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) )) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5)) \\ &= (1 / 1.41) * V_{ref} \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

**3. TMC2208 with  $R_s = 0.110\Omega$  (110m $\Omega$ ) for Stand-alone Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 30m\Omega)) )) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 30))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.32143)) \\ &= (1 / 1.41) * V_{ref} * 0.928572 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref} * 0.928572$$

$$I_{MAX} = V_{ref} * 0.9286$$

$$V_{ref} = I_{MAX} * 1.0769$$

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**4. TMC2208 with  $R_s = 0.110\Omega$  (110mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5)) \\ &= (1 / 1.41) * V_{ref} * 1 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

**5. TMC2209 with  $R_s = 0.110\Omega$  (110mΩ) for Stand-alone Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / (110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5)) \\ &= (1 / 1.41) * V_{ref} * 1 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**6. TMC2209 with  $R_s = 0.110\Omega$  (110mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / 110 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (2.5)) \\ &= (1 / 1.41) * V_{ref} * 1 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref}$$

$$I_{MAX} = V_{ref}$$

$$V_{ref} = I_{MAX}$$

**7. TMC2225 with  $R_s = 0.150\Omega$  (150mΩ) for UART Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 20m\Omega)) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / (150 + 20))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * (1.9118)) \\ &= (1 / 1.41) * V_{ref} * 0.7647 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * V_{ref} * 0.7647$$

$$I_{MAX} = V_{ref} * 0.7647$$

$$V_{ref} = I_{MAX} * 1.3077$$

**8. TMC5160 with  $R_s = 0.075\Omega$  (75mΩ) for SPI Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (325mV / R_s)) \\ &= ((1 / 1.41) * (325 / 75)) \\ &= (1 / 1.41) * 4.33 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$I_{MAX} * (1 / 1.41) = (1 / 1.41) * 4.333$$

$$I_{MAX} = 4.333 \text{ Amps}$$

You will use **50% to 90%** of  $I_{MAX}$  ( $4.333 * 0.50$  or  $4.333 * 0.90$ ) which is **2.1665 Amps (2167 mA)** to **3.8997 Amps (3900 mA)** for the Marlin Firmware.

See next page for other TMC stepper motor drivers

APPENDIX BDriving Current Calculation Formulas for TMC Stepper Motor Drivers**9. TMC5161 with  $R_s = 0.062\Omega$  (62mΩ) for SPI Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (325mV / R_s)) \\ &= ((1 / 1.41) * (325 / 62)) \\ &= (1 / 1.41) * 5.24194 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$\begin{aligned} I_{MAX} * (1 / 1.41) &= (1 / 1.41) * 5.24194 \\ I_{MAX} &= 5.24194 \text{ or } 5.2419 \end{aligned}$$

You will use **50% to 90%** of  $I_{MAX}$  ( $5.2419 * 0.50$  or  $5.2419 * 0.90$ ) which is **2.621 Amps (2621 mA)** to **4.7177 Amps (4718 mA)** for the Marlin Firmware.

**10. TMC2225 with  $R_s = 0.150\Omega$  (150mΩ) for Stand-alone Mode:**

$$\begin{aligned} I_{RMS} &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325mV / (R_s + 30m\Omega)) )) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * ((325 / (150 + 30)))) \\ &= ((1 / 1.41) * (V_{ref} / 2.5) * 1.8056) \\ &= (1 / 1.41) * V_{ref} * 0.7222 \end{aligned}$$

Since  $I_{RMS} = I_{MAX} * (1 / 1.41)$

Therefore,

$$\begin{aligned} I_{MAX} * (1 / 1.41) &= (1 / 1.41) * V_{ref} * 0.7222 \\ I_{MAX} &= V_{ref} * 0.7222 \\ V_{ref} &= I_{MAX} * 1.3846 \end{aligned}$$

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

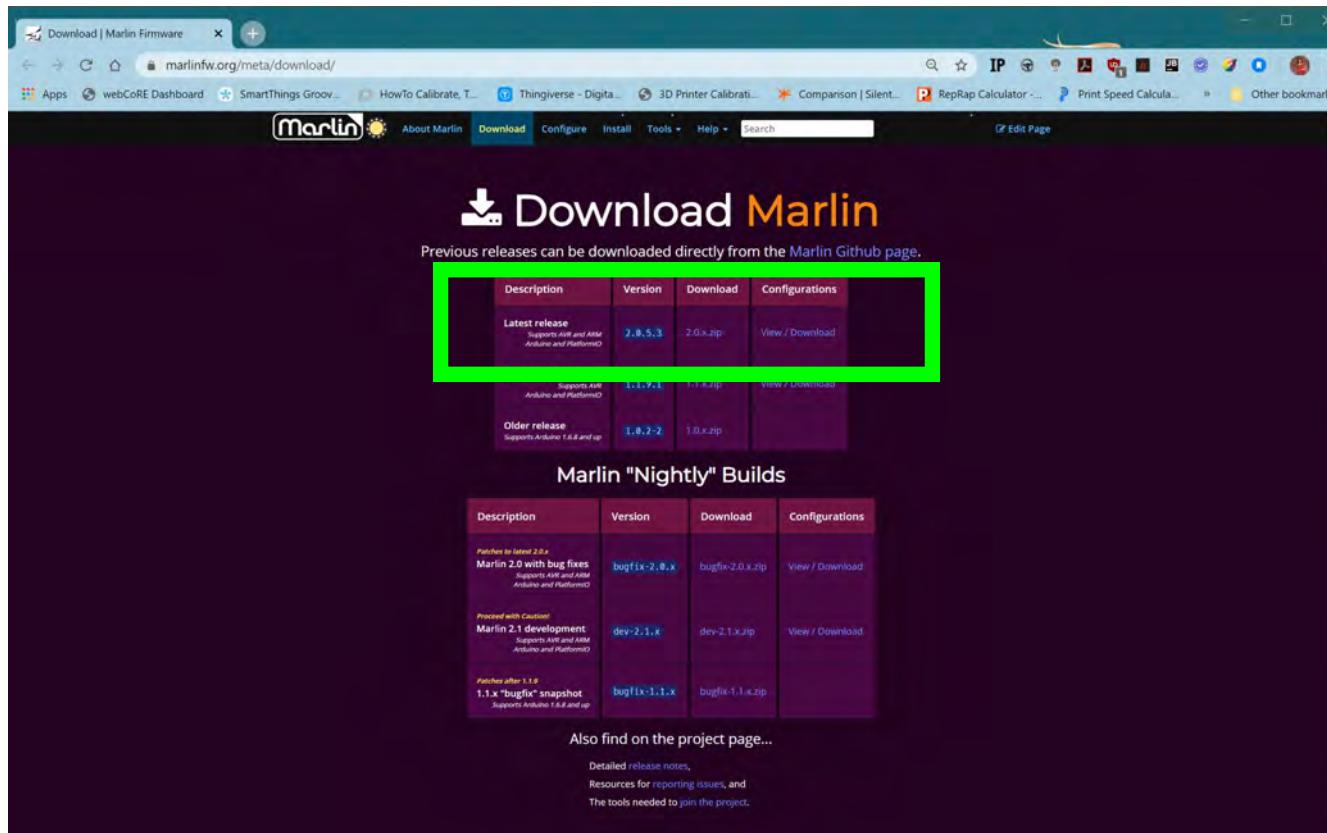
Please follow this guide to install Visual Studio Code with Platformio onto your computer. The link to the instruction are located at: [https://marlinfw.org/docs/basics/install\\_platformio.html](https://marlinfw.org/docs/basics/install_platformio.html)

**Please refer to the following documents:**

- [BIGTREETECH SKR-V1.3 Instruction Manual.pdf](#)
- [BIGTREETECH SKR-V1.3 Guide.pdf](#)

This example will use the Creality Ender 3 printer. Select the appropriate default configuration files for your specific printer!

- Download the [latest release of Marlin](https://marlinfw.org/meta/download/) from here: <https://marlinfw.org/meta/download/>
- Unzip the latest release of Marlin onto your hard drive
- Also, download the latest release of the Marlin Configuration files and then unzip the Marlin Configuration files so they reside in the same subdirectory as the Marlin files, see the pictures below for how I organize my file structure for Marlin

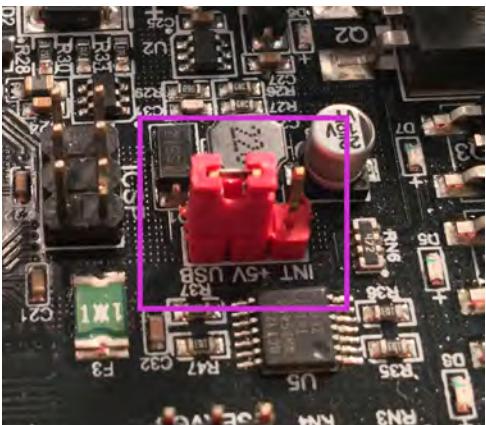


- Go to the next page.

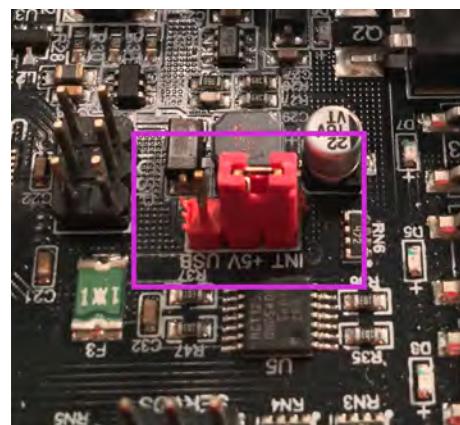
## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

To ensure that any changes I make to my Marlin Firmware are taken as I program it, I do a couple of things to ensure that the parameters are set properly. I will send an M503 by using the **Pronterface software** **BEFORE I change the Marlin Firmware** so I can **write down all the calibrations that I have previously set**.

I ensure that the "Power selection", as defined in "SKR V1.3 Instruction manual.pdf", is set to the correct power input source ②. Even though you can use the USB as the source of power when performing a firmware update, I prefer to use the 12V/24V power source, as seen below.



① USB power



② 12/24V power

### **Marlin2.0 Firmware Update Method:**

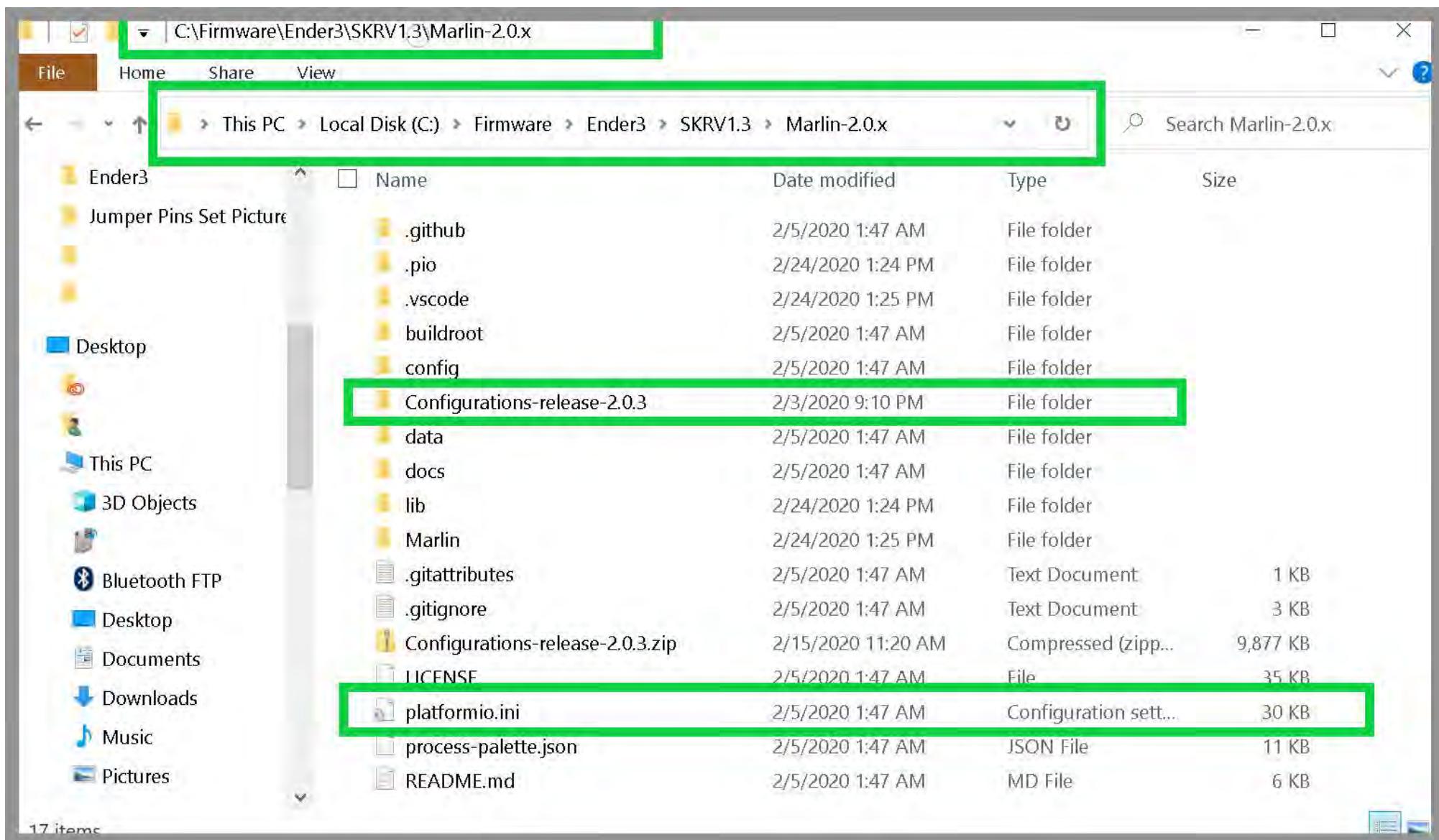
After downloading the files, use Visual Studio Code to open the project for compilation. Customize the firmware and compile it. Check for errors. If there are no errors, find the "firmware.bin" file. Copy it to the SD card and plug the SD card in the board. Reboot the board; wait for about 10 seconds before doing anything else with the board. I then use the **Pronterface software**, to send the following "G-code" commands:

- M502
- M500
- M504

M502 reset all configurable settings to their factory defaults. When you follow a M502 by a M500 the M500 will also reset settings in EEPROM. The M504 command will validate the contents of the EEPROM to ensure that the EEPROM settings have been changed to the factory defaults. If the reset does not show the correct settings, find your compiled "firmware.bin" file and copy it again to the SD card, then plug the micro SD card into the SKR V1.3 board. Reboot the board, wait for about 10 seconds and check the settings again.

**After uploading new firmware, you will need to **calibrate** your 3D printer again. Please see the following document for instructions on how to calibrate your 3D printer: [How to Calibrate your 3D printer](#)**

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

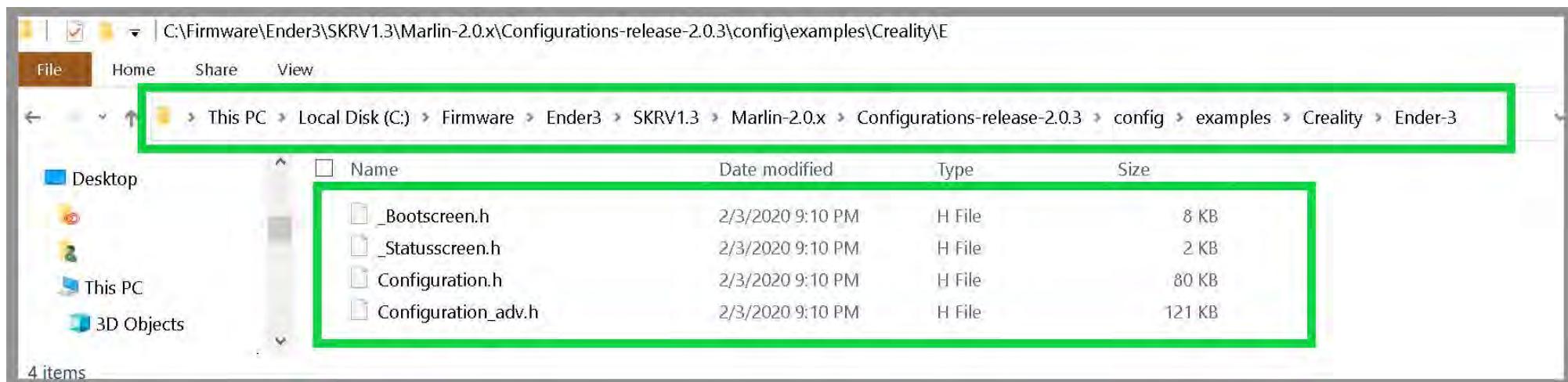


- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

**NOTE:** This example will use the Creality Ender 3 printer and uses a release of Marlin firmware which is version 2.0.3. The latest release is version 2.0.5. For version 2.0.5, all files are located in the same sub directories as version 2.0.3. The base code has been changed between these two versions of Marlin. Instead of having to replace all following pictures in this section, just insert "2.0.5" everywhere you see "2.0.3" and select the appropriate default configuration files for your specific printer!

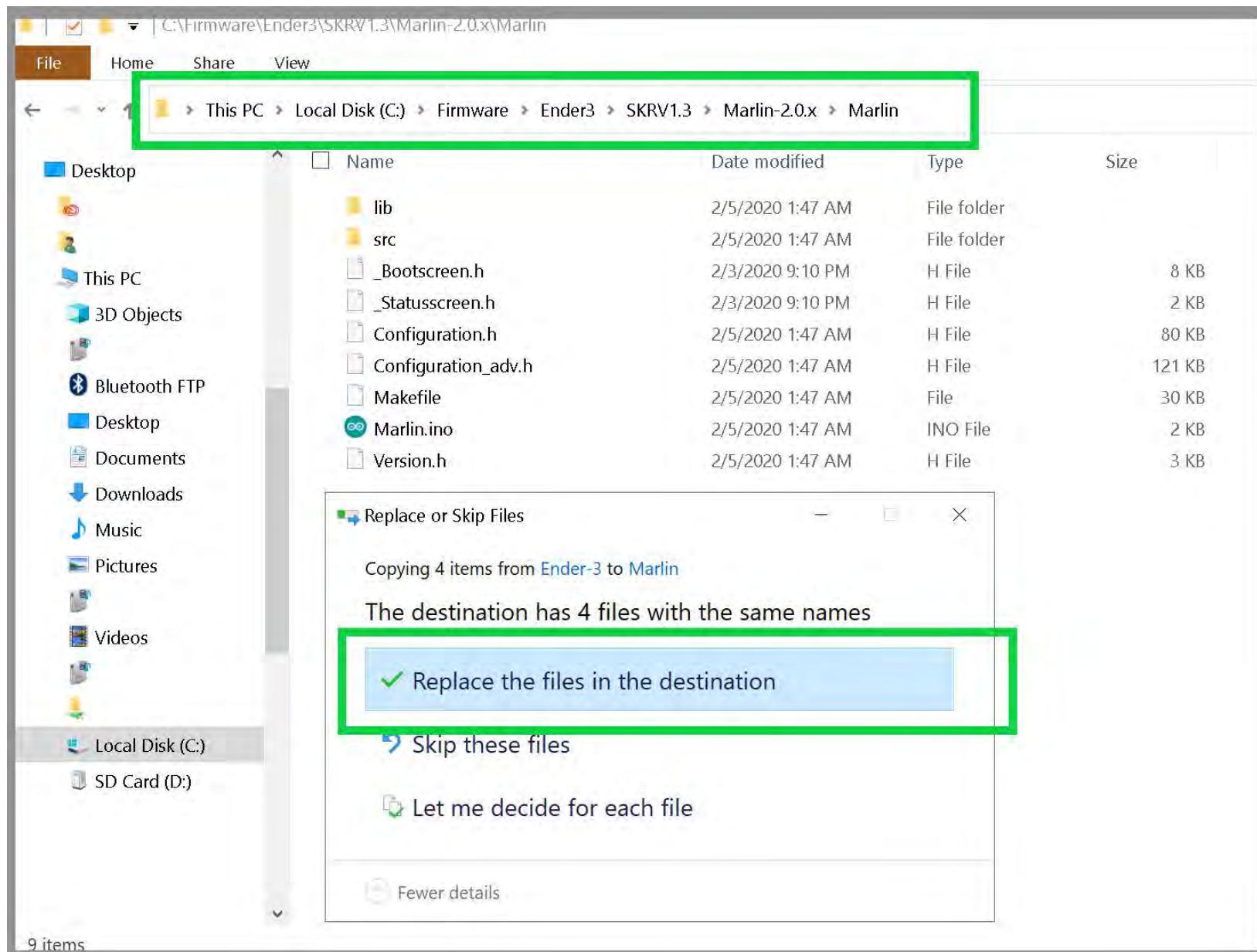
- Copy the below default configuration files (for me Ender 3 files, for you it could be another printer) to the directory where Marlin's Configuration.h and Configuration\_adv.h reside. See picture below for which files you will copy. See the next page to see where to place the files.



- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

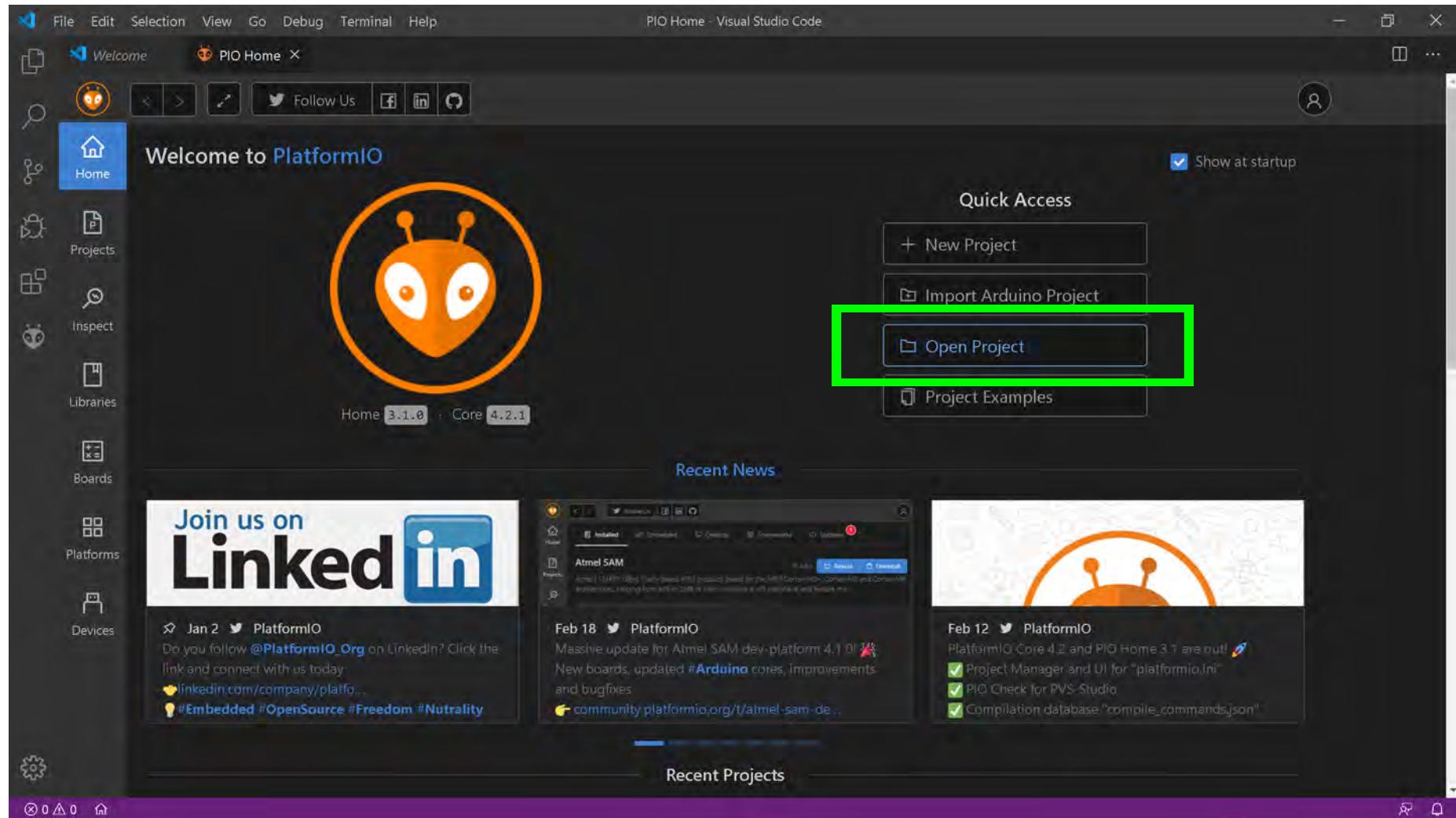
- Copy TO THIS directory so the above files reside in the same directory as Configuration.h and Configuration\_adv.h. When prompted allow the files to be overwritten!



- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

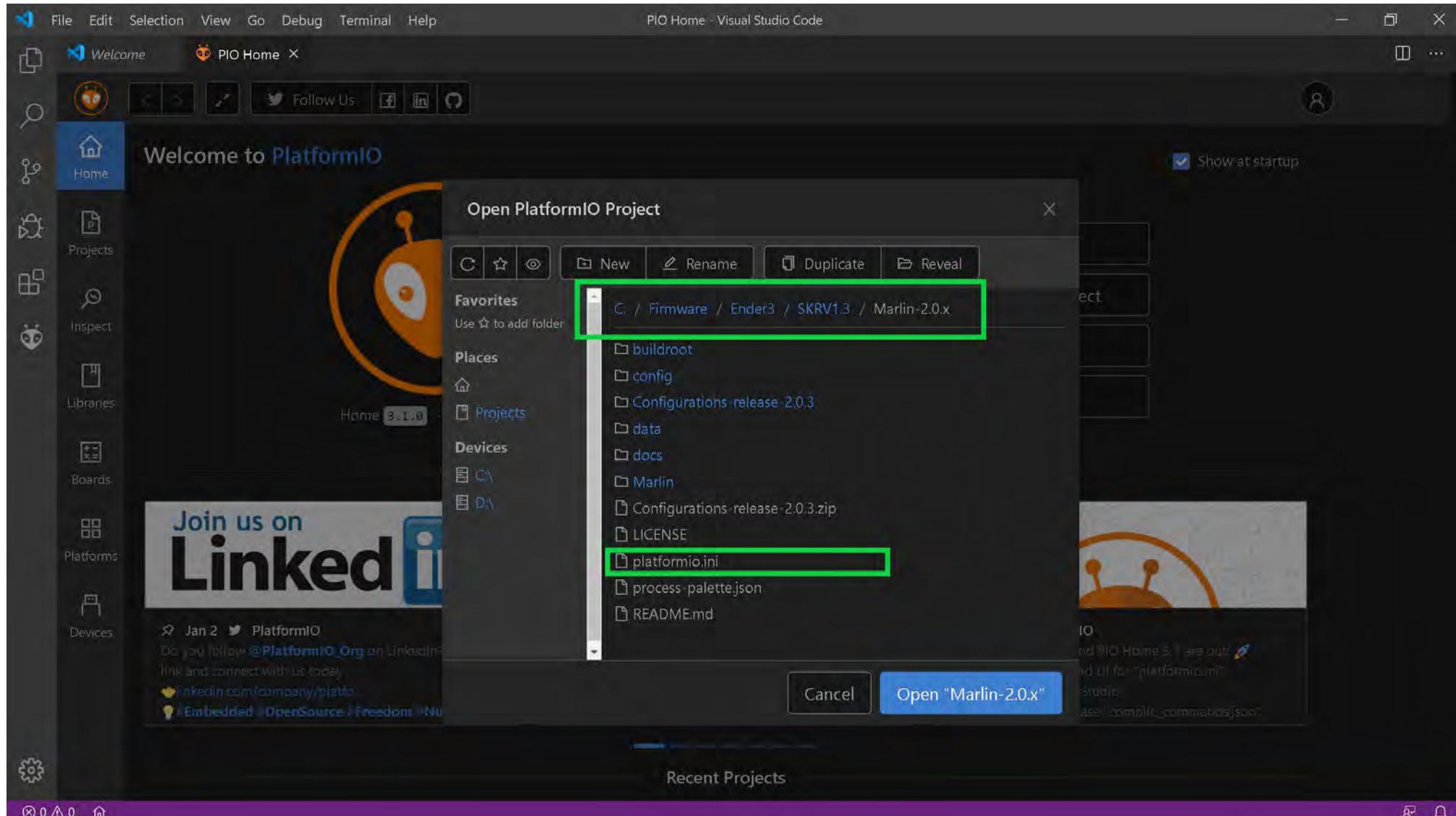
- Open VS code (see picture below) and then select "Open Project".



- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

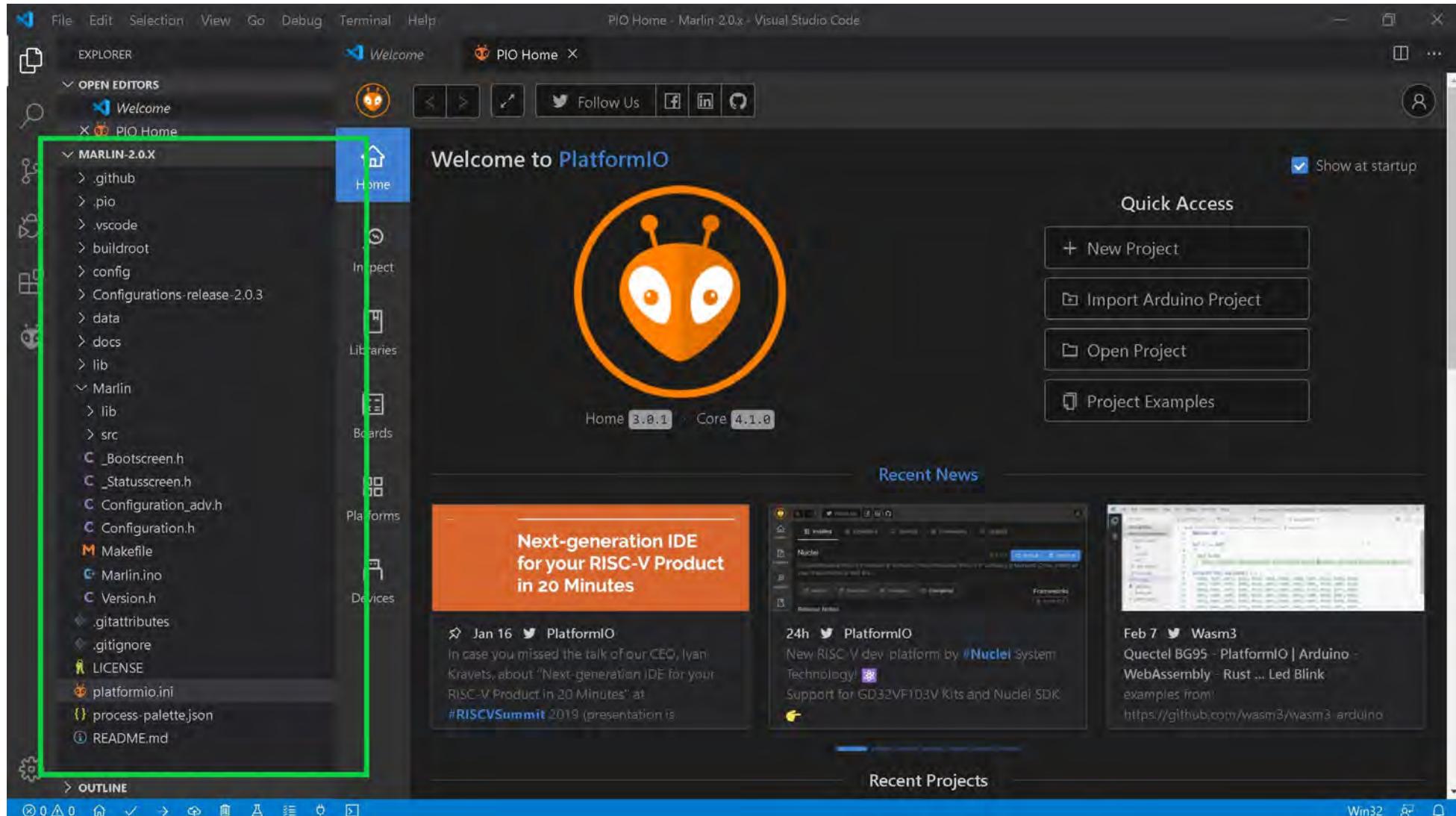
- Go to the directory where the platformio.ini file resides and open that folder (see picture below)



- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

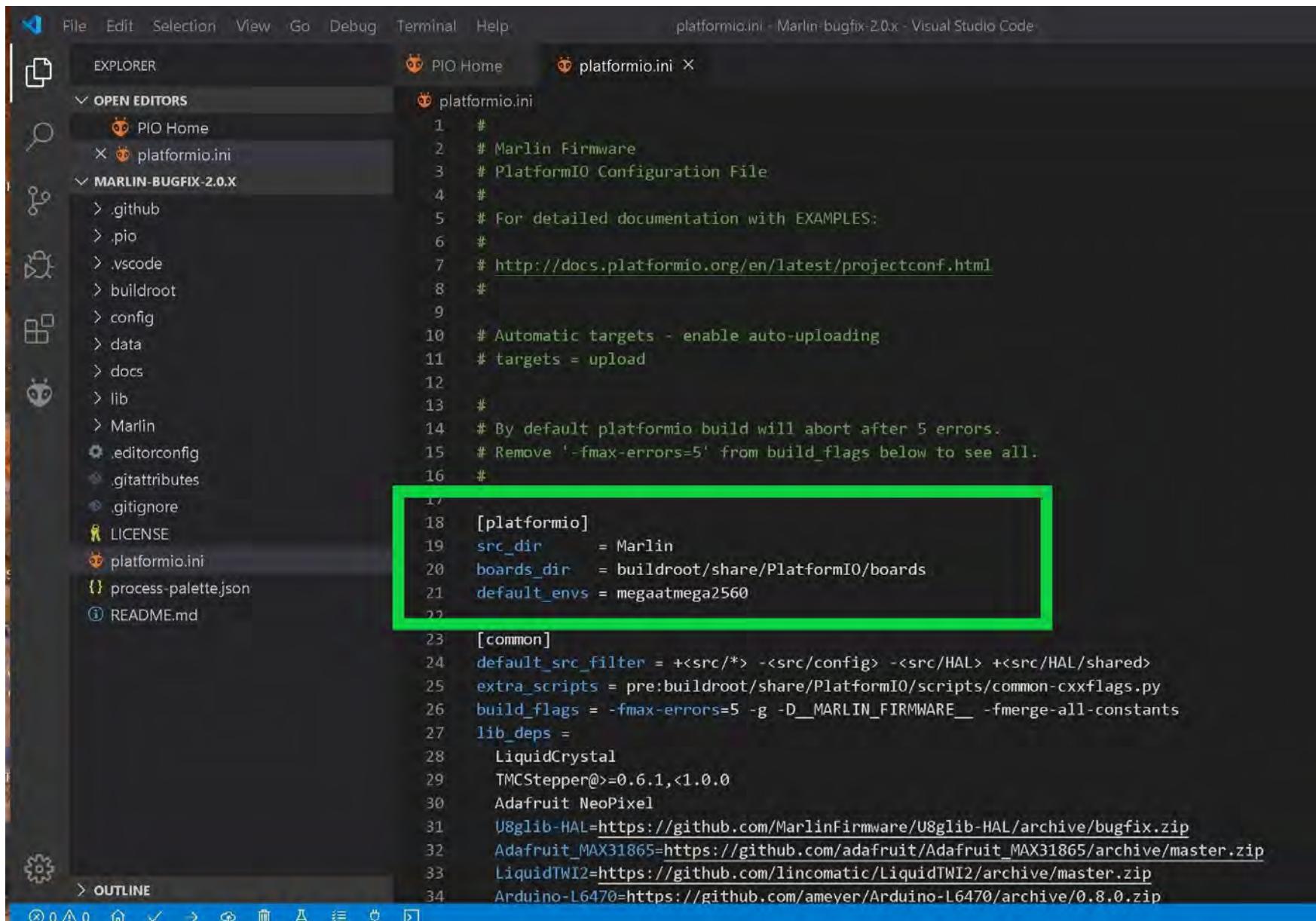
- On the left side you will see the file structure, double click on the "platformio.ini" file to open it up in the editor window.



- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- What you will see when "platformio.ini" is opened up in the VS code editor window

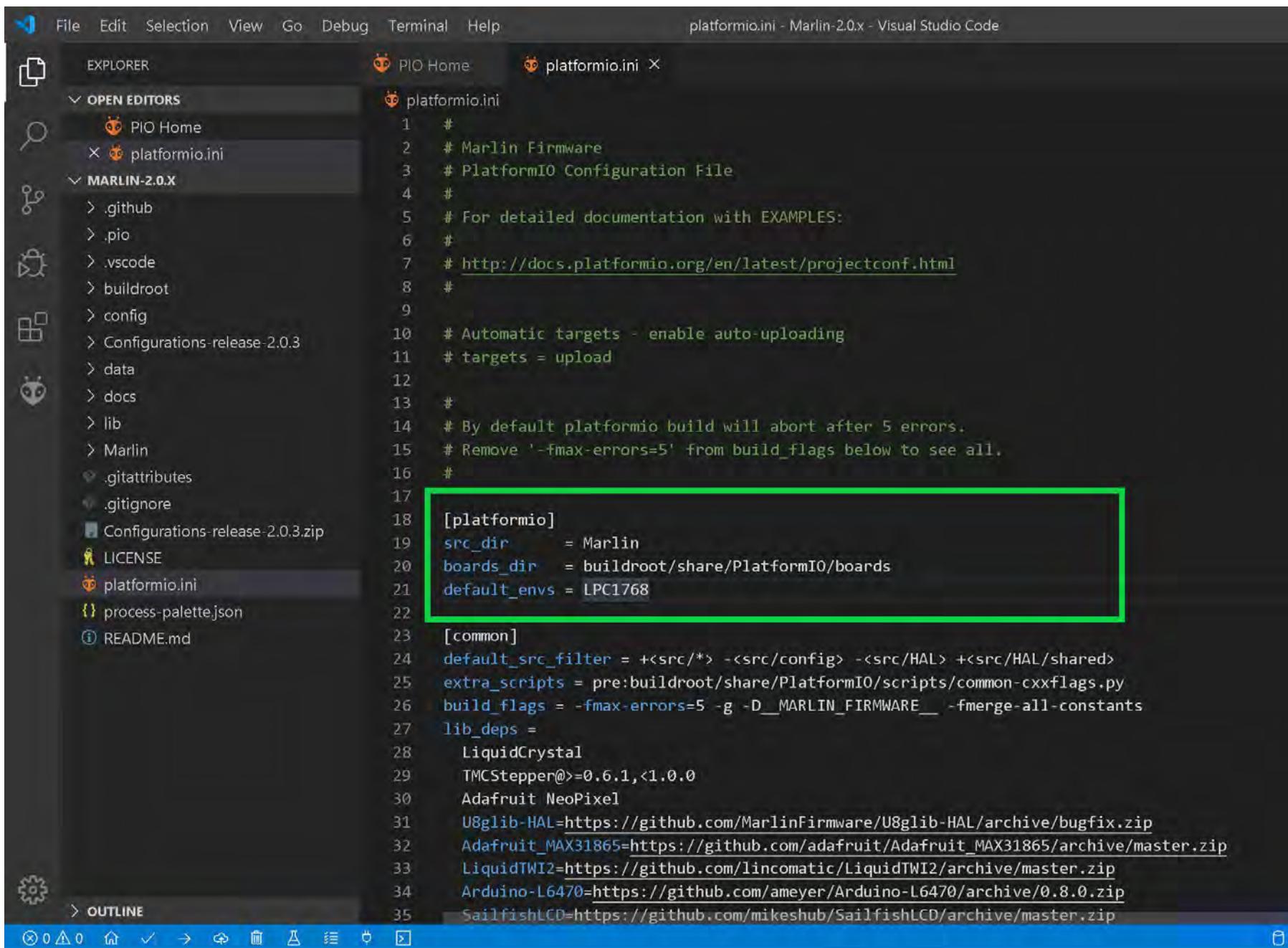


```
platformio.ini - Marlin·bugfix-2.0.x - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER PIO Home platformio.ini X
OPEN EDITORS PIO Home platformio.ini
MARLIN-BUGFIX-2.0.X
.github .pio .vscode buildroot config data docs lib Marlin .editorconfig .gitattributes .gitignore LICENSE platformio.ini process-palettejson README.md
platformio.ini
1 #
2 # Marlin Firmware
3 # PlatformIO Configuration File
4 #
5 # For detailed documentation with EXAMPLES:
6 #
7 # http://docs.platformio.org/en/latest/projectconf.html
8 #
9 #
10 # Automatic targets - enable auto-uploading
11 # targets = upload
12 #
13 #
14 # By default platformio build will abort after 5 errors.
15 # Remove '-fmax-errors=5' from build_flags below to see all.
16 #
17 [platformio]
18 src_dir = Marlin
19 boards_dir = buildroot/share/PlatformIO/boards
20 default_envs = megaatmega2560
21
22 [common]
23 default_src_filter = +<src/*> -<src/config> -<src/HAL> +<src/HAL/shared>
24 extra_scripts = pre:buildroot/share/PlatformIO/scripts/common-cxxflags.py
25 build_flags = -fmax-errors=5 -g -D__MARLIN_FIRMWARE__ -fmerge-all-constants
26 lib_deps =
27 LiquidCrystal
28 TMCStepper@>=0.6.1,<1.0.0
29 Adafruit NeoPixel
30 U8glib-HAL=https://github.com/Marlinfirmware/U8glib-HAL/archive/bugfix.zip
31 Adafruit_MAX31865=https://github.com/adafruit/Adafruit_MAX31865/archive/master.zip
32 LiquidTWI2=https://github.com/lincomatic/LiquidTWI2/archive/master.zip
33 Arduino-L6470=https://github.com/ameyer/Arduino-L6470/archive/0.8.0.zip
34
```

- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Edit the "default\_envs = megaatmega2560" line and make "default\_envs = LPC1768", as shown below in **GREEN**



```

File Edit Selection View Go Debug Terminal Help
platformio.ini - Marlin-2.0.x - Visual Studio Code

EXPLORER PIO Home platformio.ini ×
OPEN EDITORS platformio.ini
MARLIN-2.0.X
 .github
 .pio
 .vscode
 buildroot
 config
 Configurations-release 2.0.3
 data
 docs
 lib
 Marlin
 .gitattributes
 .gitignore
 Configurations-release-2.0.3.zip
 LICENSE
 platformio.ini
 process-palette.json
 README.md

[platformio]
src_dir = Marlin
boards_dir = buildroot/share/PlatformIO/boards
default_envs = LPC1768

[common]
default_src_filter = +<src/*> -<src/config> -<src/HAL> +<src/HAL/shared>
extra_scripts = pre:buildroot/share/PlatformIO/scripts/common-cxxflags.py
build_flags = -fmax-errors=5 -g -D__MARLIN_FIRMWARE__ -fmerge-all-constants
lib_deps =
 LiquidCrystal
 TMCStepper@>=0.6.1,<1.0.0
 Adafruit NeoPixel
 U8glib-HAL=https://github.com/MarlinFirmware/U8glib-HAL/archive/bugfix.zip
 Adafruit_MAX31865=https://github.com/adafruit/Adafruit_MAX31865/archive/master.zip
 LiquidTWI2=https://github.com/lincomatic/LiquidTWI2/archive/master.zip
 Arduino-L6470=https://github.com/ameyer/Arduino-L6470/archive/0.8.0.zip
 SailfishLCD=https://github.com/mikeshub/SailfishLCD/archive/master.zip

```

- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Go to the Configuration.h file and change the following three items, as seen in the **4 GREEN** boxes below.

```

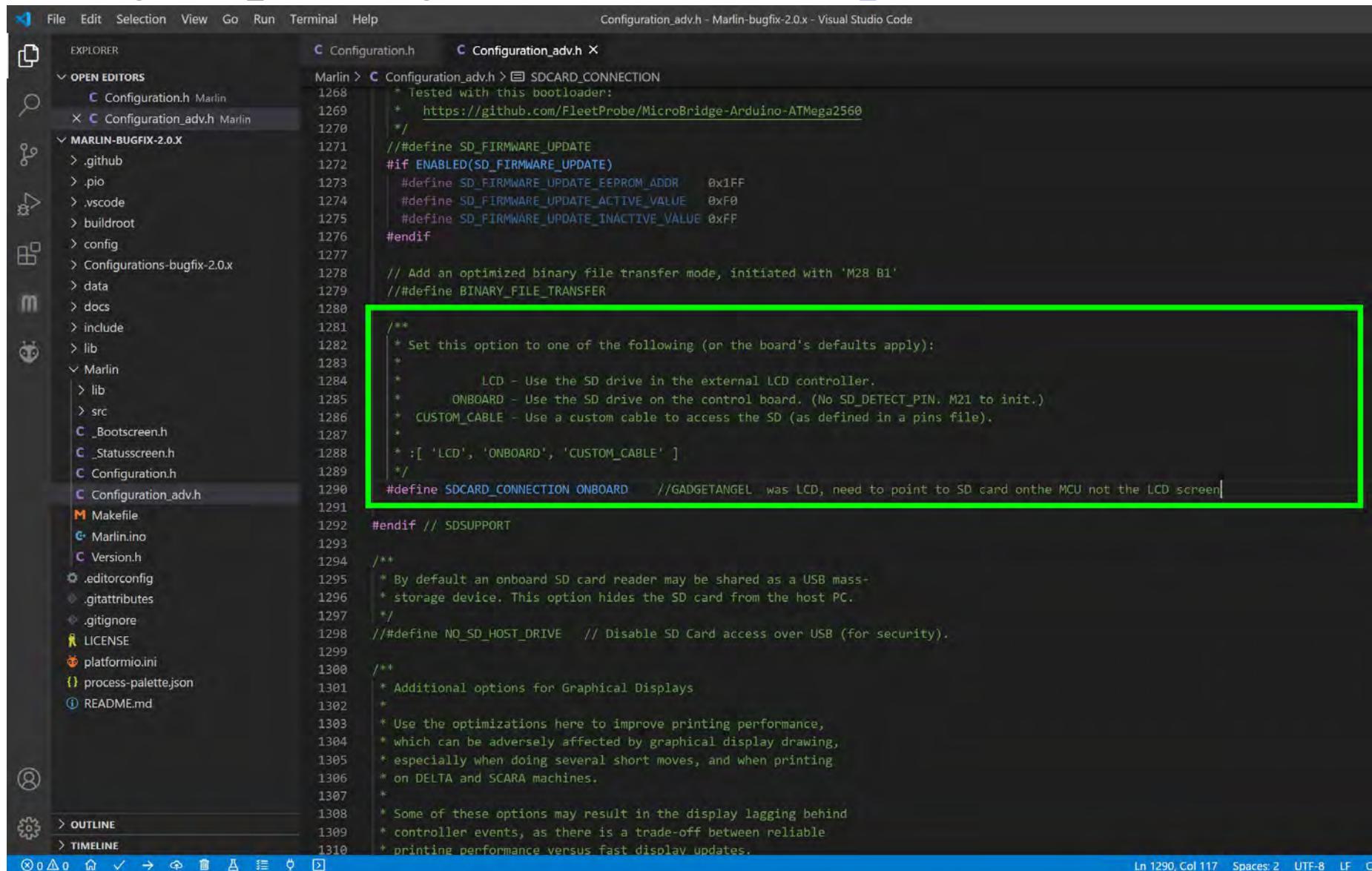
File Edit Selection View Go Run Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_common.h
OPEN EDITORS Marlin > Configuration.h > MOTHERBOARD
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
MARLIN-2.0.X
 .github
 .pio
 .vscode
 buildroot
 config
 Configurations-release-2.0.5
 data
 docs
 include
 lib
 Marlin
 lib
 src
 _Bootscreen.h
 _Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platform.ini
 process-palette.json
 README.md
OUTLINE
TIMELINE
 100 // @section machine
 101 /**
 102 * Select the serial port on the board to use for communication with the host.
 103 * This allows the connection of wireless adapters (for instance) to non-default port pins.
 104 * Serial port -1 is the USB emulated serial port, if available.
 105 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
 106 */
 107 #define SERIAL_PORT -1 //GADGETANGEL was 0
 108 /**
 109 * Select a secondary serial port on the board to use for communication with the host.
 110 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
 111 */
 112 #define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out
 113 /**
 114 * This setting determines the communication speed of the printer.
 115 *
 116 * 250000 works in most cases, but you might try a lower speed if
 117 * you commonly experience drop-outs during host printing.
 118 * You may try up to 1000000 to speed up SD file transfer.
 119 */
 120 /**
 121 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 122 */
 123 /**
 124 #define BAUDRATE 115200
 125 /**
 126 // Enable the Bluetooth serial interface on AT90USB devices
 127 //#define BLUETOOTH
 128 /**
 129 // Choose the name from boards.h that matches your setup
 130 #ifndef MOTHERBOARD
 131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
 132 #endif
 133 /**
 134 // Name displayed in the LCD "Ready" message and Info menu
 135 #define CUSTOM_MACHINE_NAME "Ender-3"
 136 /**
 137 // Printer's unique ID, used by some programs to differentiate between machines.
 138 // Choose your own or use a service like http://www.uuidgenerator.net/version4
 139 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"

```

- 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

- In Configuration\_adv.h, **enable** the Marlin variable **SDCARD\_CONNECTION** or that you **delete the two forward slashes ("//") from the beginning of the line.**
- In Configuration\_adv.h, change the Marlin variable **SDCARD\_CONNECTION** to **ONBOARD**



```

File Edit Selection View Go Run Terminal Help Configuration_adv.h - Marlin-bugfix-2.0.x - Visual Studio Code

C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > SDCARD_CONNECTION
1268 * Tested with this bootloader:
1269 * https://github.com/FleetProbe/MicroBridge-Arduino-ATMega2560
1270 */
1271 #define SD_FIRMWARE_UPDATE
1272 #if ENABLED(SD_FIRMWARE_UPDATE)
1273 #define SD_FIRMWARE_UPDATE_EEPROM_ADDR 0x1FF
1274 #define SD_FIRMWARE_UPDATE_ACTIVE_VALUE 0xF0
1275 #define SD_FIRMWARE_UPDATE_INACTIVE_VALUE 0xFF
1276#endif
1277 // Add an optimized binary file transfer mode, initiated with 'M28 B1'
1278 // #define BINARY_FILE_TRANSFER
1279
1280 /**
1281 * Set this option to one of the following (or the board's defaults apply):
1282 *
1283 * LCD - Use the SD drive in the external LCD controller.
1284 * ONBOARD - Use the SD drive on the control board. (No SD_DETECT_PIN, M21 to init.)
1285 * CUSTOM_CABLE - Use a custom cable to access the SD (as defined in a pins file).
1286 *
1287 * :['LCD', 'ONBOARD', 'CUSTOM_CABLE']
1288 */
1289#define SDCARD_CONNECTION ONBOARD //GADGETANGEL was LCD, need to point to SD card onthe MCU not the LCD screen
1290
1291#endif // SDSUPPORT
1292
1293 /**
1294 * By default an onboard SD card reader may be shared as a USB mass-
1295 * storage device. This option hides the SD card from the host PC.
1296 */
1297#define NO_SD_HOST_DRIVE // Disable SD Card access over USB (for security).
1298
1299 /**
1300 * Additional options for Graphical Displays
1301 *
1302 * Use the optimizations here to improve printing performance,
1303 * which can be adversely affected by graphical display drawing,
1304 * especially when doing several short moves, and when printing
1305 * on DELTA and SCARA machines.
1306 *
1307 * Some of these options may result in the display lagging behind
1308 * controller events, as there is a trade-off between reliable
1309 * printing performance versus fast display updates.
1310

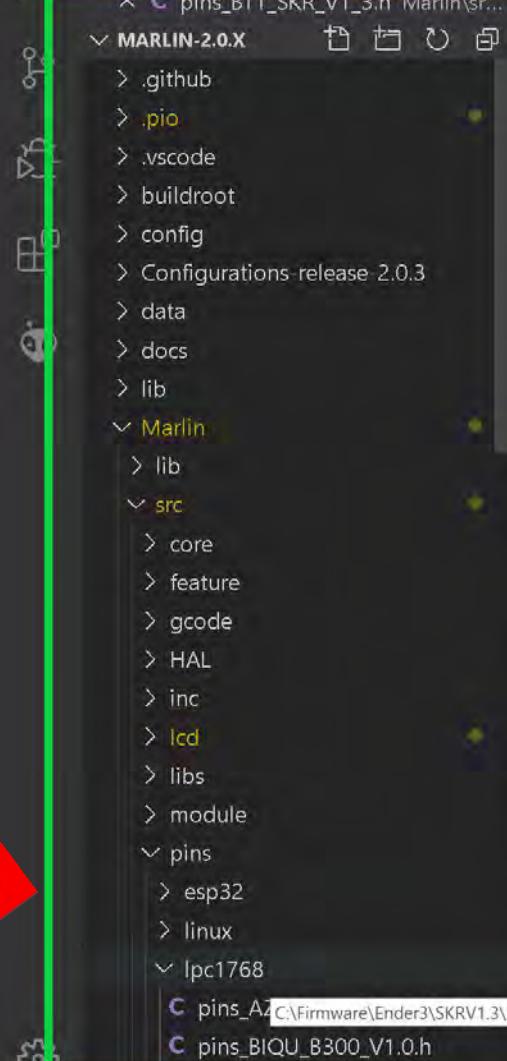
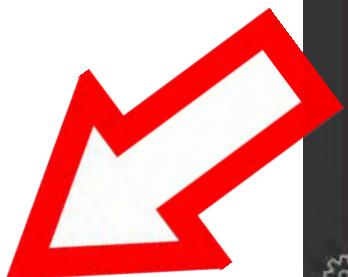
```

Ln 1290, Col 117 Spaces: 2 UTF-8 LF C

- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- Time to learn where the SKR V1.3 board's pins file is located: look at the left side and find the **C:\...\Marlin-2.0.X\Marlin\src\pins\lpc1768** subdirectory, as seen in the pictures below. Open the file, "pins\_BTT\_SKR\_V1\_3.h", by double clicking on it.



```
File Edit Selection View Go Debug Terminal Help
EXPLORER Configuration.h
OPEN EDITORS Configuration.h Marlin
pins_BTT_SKR_V1_3.h Marlin\sr...
MARLIN-2.0.X .github .pio .vscode buildroot config Configurations-release 2.0.3 data docs lib Marlin lib src core feature gcode HAL inc lcd libs module pins esp32 linux lpc1768 pins_AZ C:\Firmware\Ender3\SKR V1.3\Marlin-2.0.x\Marlin\src\pins\lpc1768 pins_BIQU_B300_V1.0.h
OUTLINE 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
```

- Go to the next page.

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

EXPLORER

OPEN EDITORS

MARLIN-2.0.X

Marlin

src

core

feature

gcode

HAL

inc

Lcd

libs

module

pins

esp32

linux

lpc1768

C pins\_BTT\_SKR\_V1\_3.h

C pins\_BIQU\_B300\_V1.0.h

C pins\_BIQU\_BQ111\_A4.h

C pins\_BTT\_SKR\_V1\_1.h

C pins\_BTT\_SKR\_V1\_3.h

C pins\_BTT\_SKR\_V1\_4.h

C pins\_BTT\_SKR.h

C pins\_GMARSH\_X6\_REV1.h

C pins\_MKS\_SBASE.h

C pins\_MKS\_SGEN\_L.h

C pins\_RAMPS\_RE\_ARM.h

```

C pins_BTT_SKR_V1_3.h
Marlin > src > pins > lpc1768 > C pins_BTT_SKR_V1_3.h ...
1 /**
2 * Marlin 3D Printer Firmware
3 * Copyright (c) 2019 MarlinFirmware [https://github.com/MarlinFirmware/Marlin]
4 *
5 * Based on Sprinter and grbl.
6 * Copyright (c) 2011 Camiel Gubbels / Erik van der Zalm
7 *
8 * This program is free software: you can redistribute it and/or modify
9 * it under the terms of the GNU General Public License as published by
10 * the Free Software Foundation, either version 3 of the License, or
11 * (at your option) any later version.
12 *
13 * This program is distributed in the hope that it will be useful,
14 * but WITHOUT ANY WARRANTY; without even the implied warranty of
15 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 * GNU General Public License for more details.
17 *
18 * You should have received a copy of the GNU General Public License
19 * along with this program. If not, see <http://www.gnu.org/licenses/>.
20 *
21 */
22 #pragma once
23
24 #define BOARD_INFO_NAME "BIGTREE SKR 1.3"
25
26 /**
27 * Trinamic Stallguard pins
28
29 #define X_DIAG_PIN P1_29 // X-
30 #define Y_DIAG_PIN P1_27 // Y-
31 #define Z_DIAG_PIN P1_25 // Z-
32 #define E0_DIAG_PIN P1_28 // X+
33 #define E1_DIAG_PIN P1_26 // Y+
34

```

C pins\_BTT\_SKR\_V1\_3.h

C pins\_BTT\_SKR\_V1\_4.h

C pins\_BTT\_SKR.h

C pins\_GMARSH\_X6\_REV1.h

```

C pins_BTT_SKR_V1_3.h
27 * Trinamic Stallguard pins
28
29 #define X_DIAG_PIN P1_29 // X-
30 #define Y_DIAG_PIN P1_27 // Y-

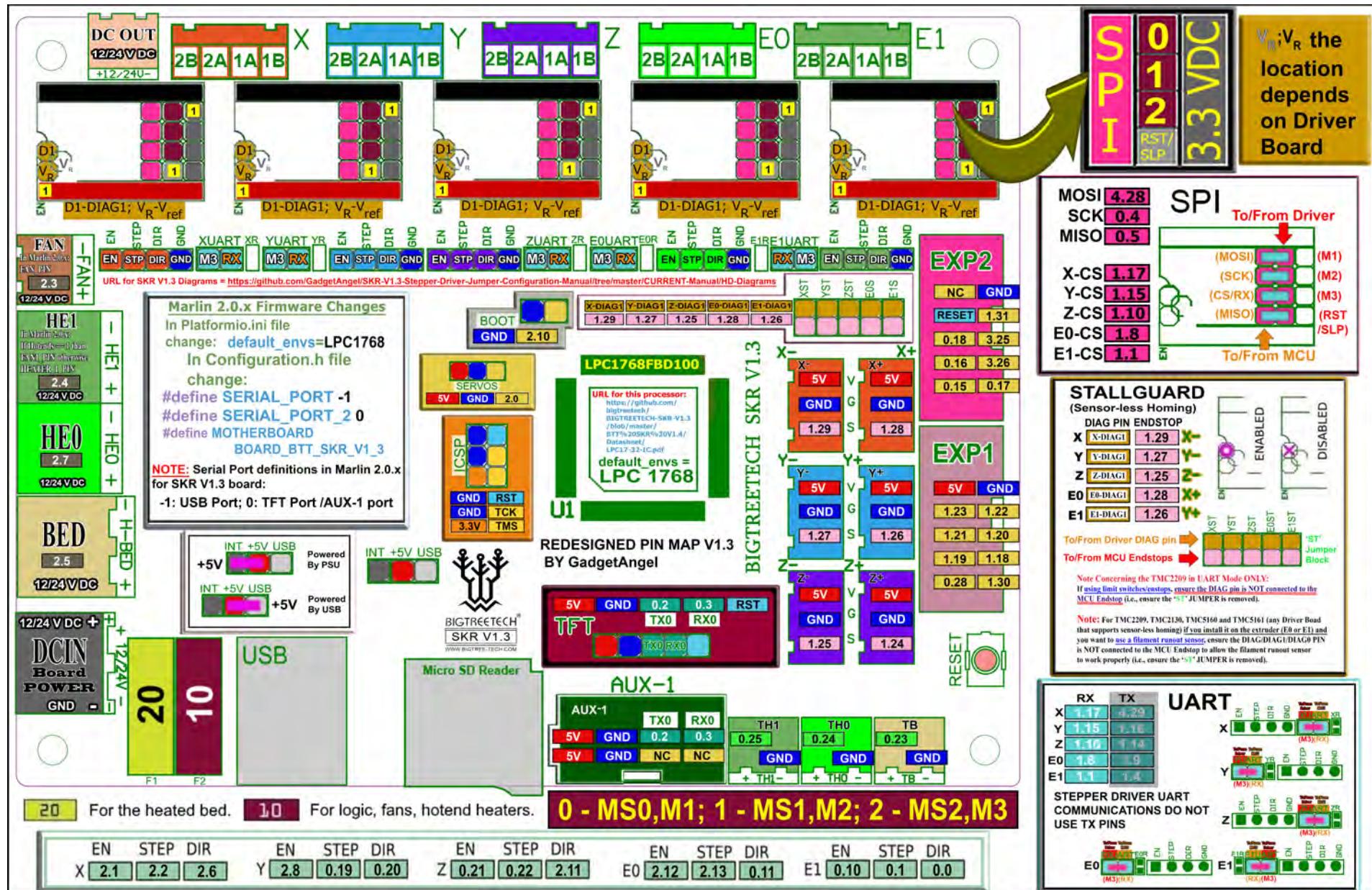
```

- Go to the next page.

## APPENDIX C

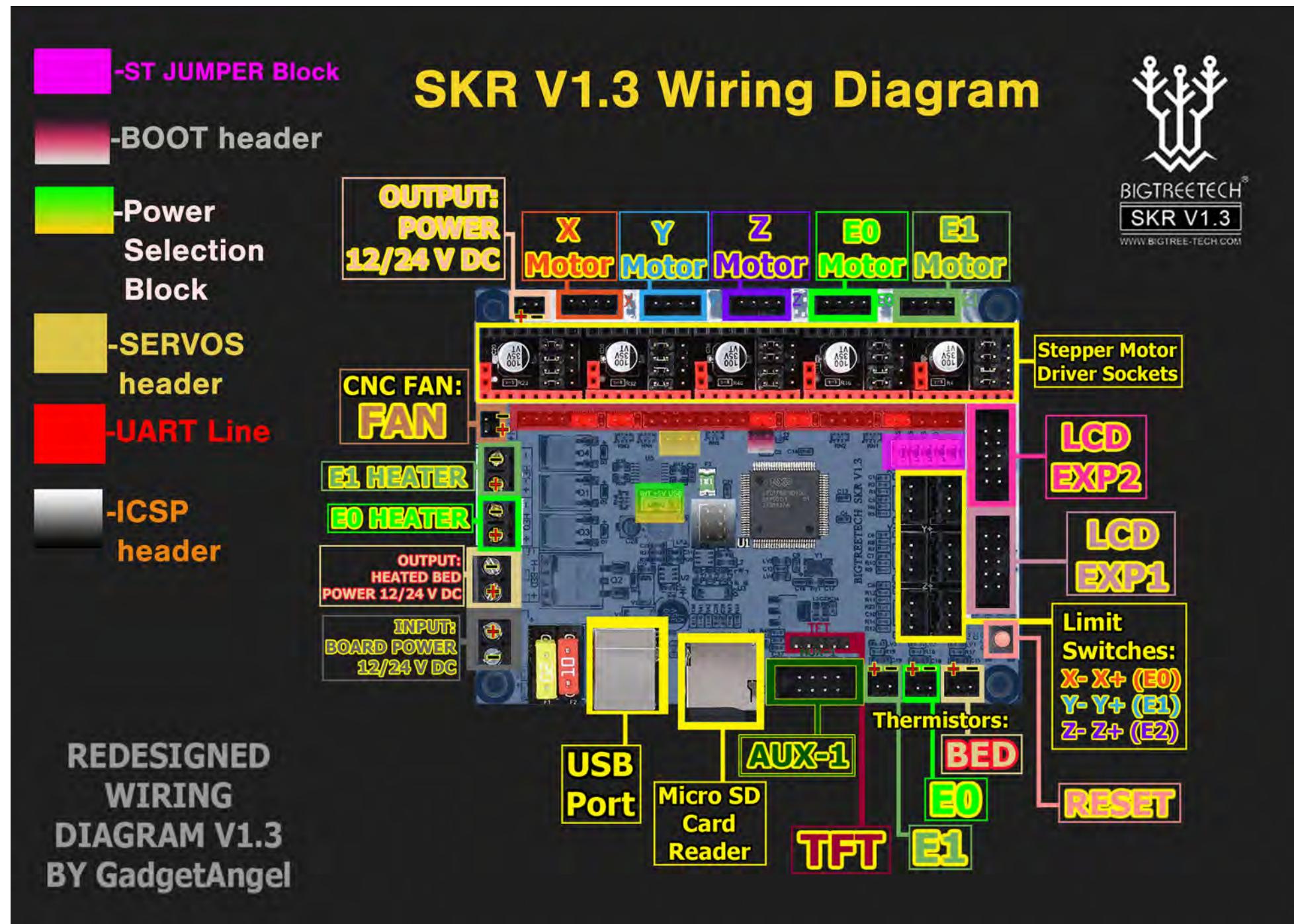
**The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers**

- We want to set the fan on the hot end to automatically turn on when the hot end starts to get hot. In the pins\_BTT\_SKR\_V1\_3.h file you will find all the pins that are defined for the board. They should all conform to the SKR V1.3 Pin diagram shown below. [See the next page](#) for a functional picture of the SKR V1.3 wiring diagram.



- We want to copy and paste the Marlin name or the actual pin number of where you hooked up the fan for your hot end and use that to set "E0\_AUTO\_FAN\_PIN" in the Configuration\_adv.h file. So, in this example I will use FAN1 port for my hot end cooling fan. In pins\_BTT\_SKR\_V1\_3.h file you will not find the FAN1 port defined, but you will see that this pins file refers to pins\_BTT\_SKR\_common.h file. You can also see that there is not a FAN1 port on the SKR V1.3 board. Since I am only using one hot end, I can use HE1 port as FAN1 port. Usually a second hot end heater port can act as a CNC FAN port if a second hot end is not attached to your printer.
- Go to the next page.

## 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

- I plan on using the main FAN port or main CNC FAN port to control the fan in the electronics box. I prefer to have the electronics box fan to run only when the electronics need cooling. The main CNC FAN port is on "FAN\_PIN" or "P2\_03", as seen in "pins\_BTT\_SKR\_common.h".
  - When you go to pins\_BTT\_SKR\_V1\_3.h file, you will see that it refers to pins\_BTT\_SKR\_common.h file. In pins\_BTT\_SKR\_common.h file you will find the pin definitions for all the FAN ports available on the SKR V1.3 board. In this example I will use the HE1 port for my hot end cooling fan. In pins\_BTT\_SKR\_common.h file we see FAN1 port (FAN1\_PIN) is defined as P2\_04. If you look at the [SKR V1.3 color pin diagram](#) you will see that HE1 port has 2.4 as its pin number (i.e. "P2\_04" in Marlin firmware). So you can choose to copy "P2\_04" or "FAN1\_PIN".

```
C:\pins_BTT_SKR_V1_3.h
Marlin > src > pins > lpc1768 > pins_BTT_SKR_V1_3.h > ...
349 // (NOT 5V tolerant)
350
351 #ifndef EYSETC_MINI_12864
352
353 #endif // !CR10_STOCKDISPLAY
354
355 #endif // HAS_SPI_LCD
356
357 //
358 // SD Support
359 //
360
361 #ifndef SDCARD_CONNECTION
362 #define SDCARD_CONNECTION LCD
363 #endif
364
365 #if SD_CONNECTION_IS(LCD)
366 #define SS_PIN EXP_A2_07_PIN
367 #endif
368
369 /**
370 * Special pins
371 * P1_30 (37) (NOT 5V tolerant)
372 * P1_31 (49) (NOT 5V tolerant)
373 * P0_27 (57) (Open collector)
374 * P0_28 (58) (Open collector)
375 */
376
377 // Include common SKR pins
378 #include "pins_BTT_SKR_common.h"

C:\pins_BTT_SKR_common.h
Marlin > src > pins > lpc1768 > pins_BTT_SKR_common.h > ...
50 // (NOT 5V tolerant)
51
52 #ifndef TEMP_BED_PIN
53 #define TEMP_BED_PIN P0_23_A0 // A0 (T0) - (67) - TEMP_BED_PIN
54 #endif
55
56 #if HOTENDS == 1 && TEMP_SENSOR_PROBE
57 #define TEMP_PROBE_PIN TEMP_1_PIN
58 #endif
59
60 //
61 // Heaters / Fans
62 //
63 #ifndef HEATER_0_PIN
64 #define HEATER_0_PIN P2_07
65 #endif
66
67 #if HOTENDS == 1
68 #ifndef FAN1_PIN
69 #define FAN1_PIN P2_04
70 #endif
71
72 #ifndef HEATER_1_PIN
73 #define HEATER_1_PIN P2_04
74 #endif
75
76 #ifndef FAN_PIN
77 #define FAN_PIN P2_03
78 #endif
79
80 #ifndef HEATER_BED_PIN
81 #define HEATER_BED_PIN P2_05
82 #endif
83
84
85
86
87
88
89
90
91
92 //
93 // LCD / Controller
94 //
95 #if HAS_SPI_LCD
96 #define BEEPER_PIN P1_30 // (37) not 5V tolerant
97 #endif
98
99 //
100 // SD Support
101 //
102 #define ONBOARD_SD_CS_PIN P0_06 // Chip select for "System" SD
```

- Go to the next page.

## **APPENDIX C**

## The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers

- To set the hot end fan, I will use FAN1\_PIN and set "E0\_AUTO\_FAN\_PIN" in the Configuration\_adv.h file to FAN1\_PIN, as seen in the picture below

The screenshot shows the Visual Studio Code interface with the following highlights:

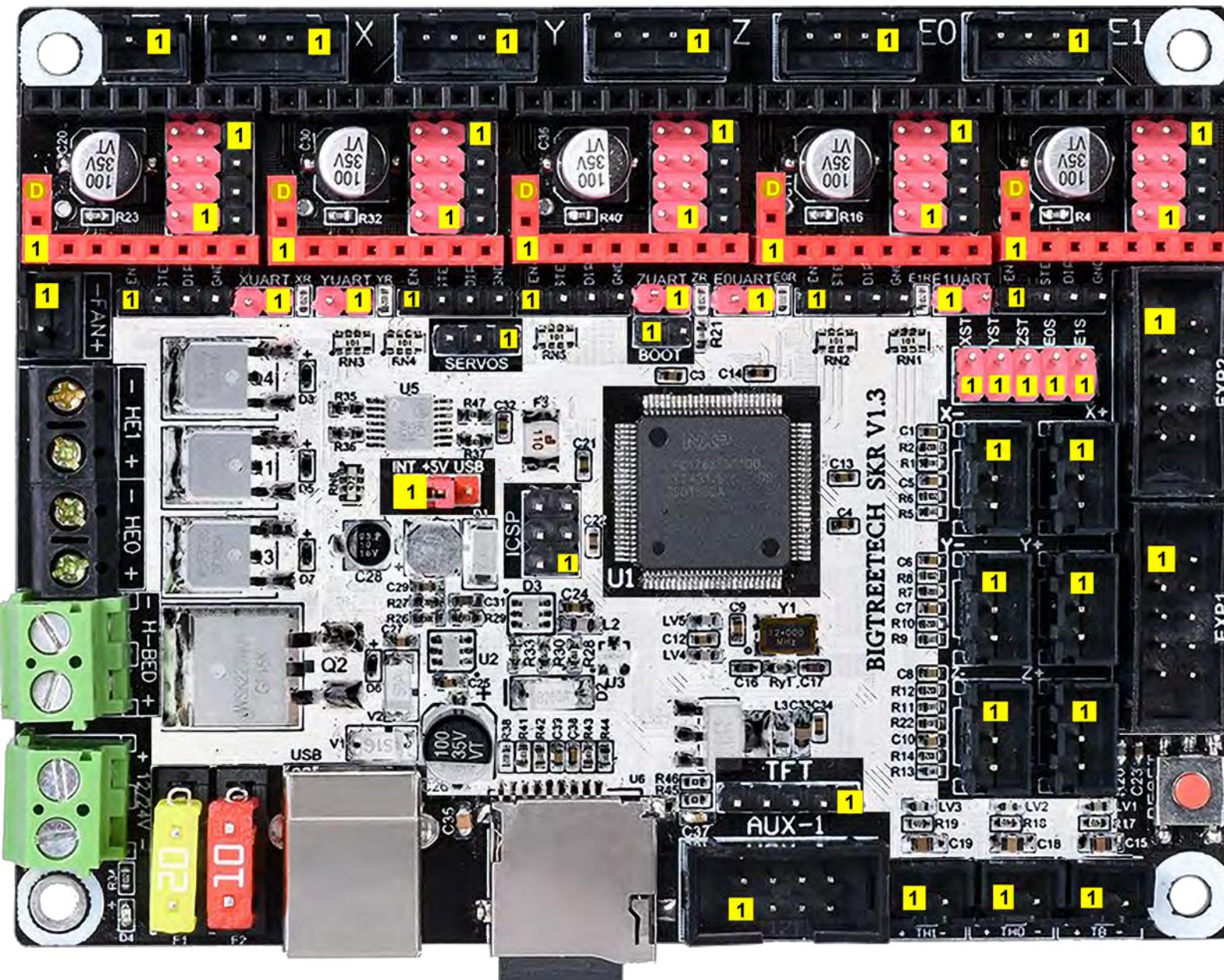
- A large green box surrounds the entire code editor area, encompassing the header files and their definitions.
- A second green box highlights the section of `Configuration_adv.h` starting at line 412, which defines fan pins for extruders E0 through E5 and a chamber fan.
- A third green box highlights the section of `Configuration_adv.h` starting at line 432, which defines temperatures and speeds for extruder and chamber fans.

```
File Edit Selection View Go Run Terminal Help Configuration_adv.h - Marlin-2.0.x - Visual Studio Code

EXPLORER
OPEN EDITORS
 Configuration.h Marlin
 pins_BTT_SKR_V1_3.h Marlin\src...
 pins_BTT_SKR_common.h Marlin...
 Configuration_adv.h Marlin
MARLIN-2.0.X
 > lpc1769
 > mega
 > rambo
 > ramps
 > sam
 > samd
 > sanguino
 > stm32f1
 > stm32f4
 > stm32f7
 > teensy2
 > teensy3
 pins.h
 pinsDebug.h
 pinsDebug.list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 Bootscreen.h
 Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 version.n
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 OUTLINE
 TIMELINE
C Configuration.h C pins_BTT_SKR_V1_3.h C pins_BTT_SKR_common.h C Configuration_adv.h
Marlin > C Configuration_adv.h => EO_AUTO_FAN_PIN
401 * USE_OCR2A_AS_TOP sacrifices duty cycle control resolution to achieve this broader
402 */
403 #if ENABLED(FAST_PWM_FAN)
404 //Define FAST_PWM_FAN_FREQUENCY 31400
405 //Define USE_OCR2A_AS_TOP
406 #endif
407
408 // @section extruder
409
410 /**
411 * Extruder cooling fans
412 *
413 * Extruder auto fans automatically turn on when their extruders'
414 * temperatures go above EXTRUDER_AUTO_FAN_TEMPERATURE.
415 *
416 * Your board's pins file specifies the recommended pins. Override those here
417 * or set to -1 to disable completely.
418 *
419 * Multiple extruders can be assigned to the same pin in which case
420 * the fan will turn on when any selected extruder is above the threshold.
421 */
422 #define E0_AUTO_FAN_PIN FAN1_PIN //GADGETANGEL was -1
423 #define E1_AUTO_FAN_PIN -1
424 #define E2_AUTO_FAN_PIN -1
425 #define E3_AUTO_FAN_PIN -1
426 #define E4_AUTO_FAN_PIN -1
427 #define E5_AUTO_FAN_PIN -1
428 #define CHAMBER_AUTO_FAN_PIN -1
429
430 #define EXTRUDER_AUTO_FAN_TEMPERATURE 50
431 #define EXTRUDER_AUTO_FAN_SPEED 255 // 255 == full speed
432 #define CHAMBER_AUTO_FAN_TEMPERATURE 30
433 #define CHAMBER_AUTO_FAN_SPEED 255
434
435 /* COOLING FAN REPLICATOR
436 */
437
438 * This feature allows you to digitally multiplex the fan output.
439 * The multiplexer is automatically switched at tool-change.
440 * Set FANMUX[012]_PINs below for up to 2, 4, or 8 multiplexed fans.
441 */
442 #define FANMUX0_PIN -1
443 #define FANMUX1_PIN -1
C:\Firmware\Ender3\SKRv1.3\Marlin-2.0.x\Marlin\Configuration_adv.h
```

- To see more Marlin setup for the latest release, [please refer to the stepper motor driver section of this document for the stepper motor driver of your choice.](#)
  - **To see my updated PIN 1 diagram for SKR V1.3, go to the next page.**

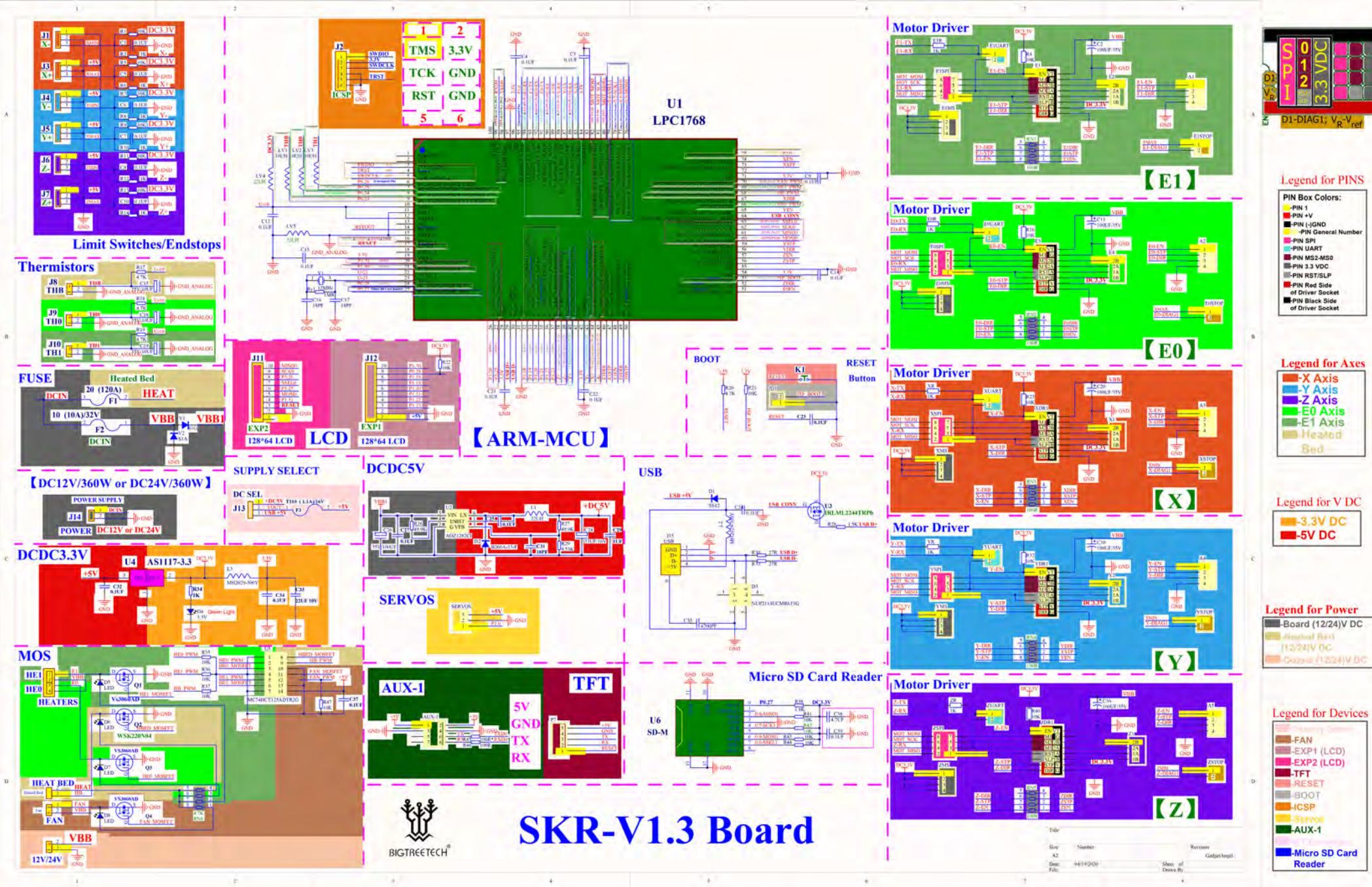
## APPENDIX C - Color PIN 1 Diagram



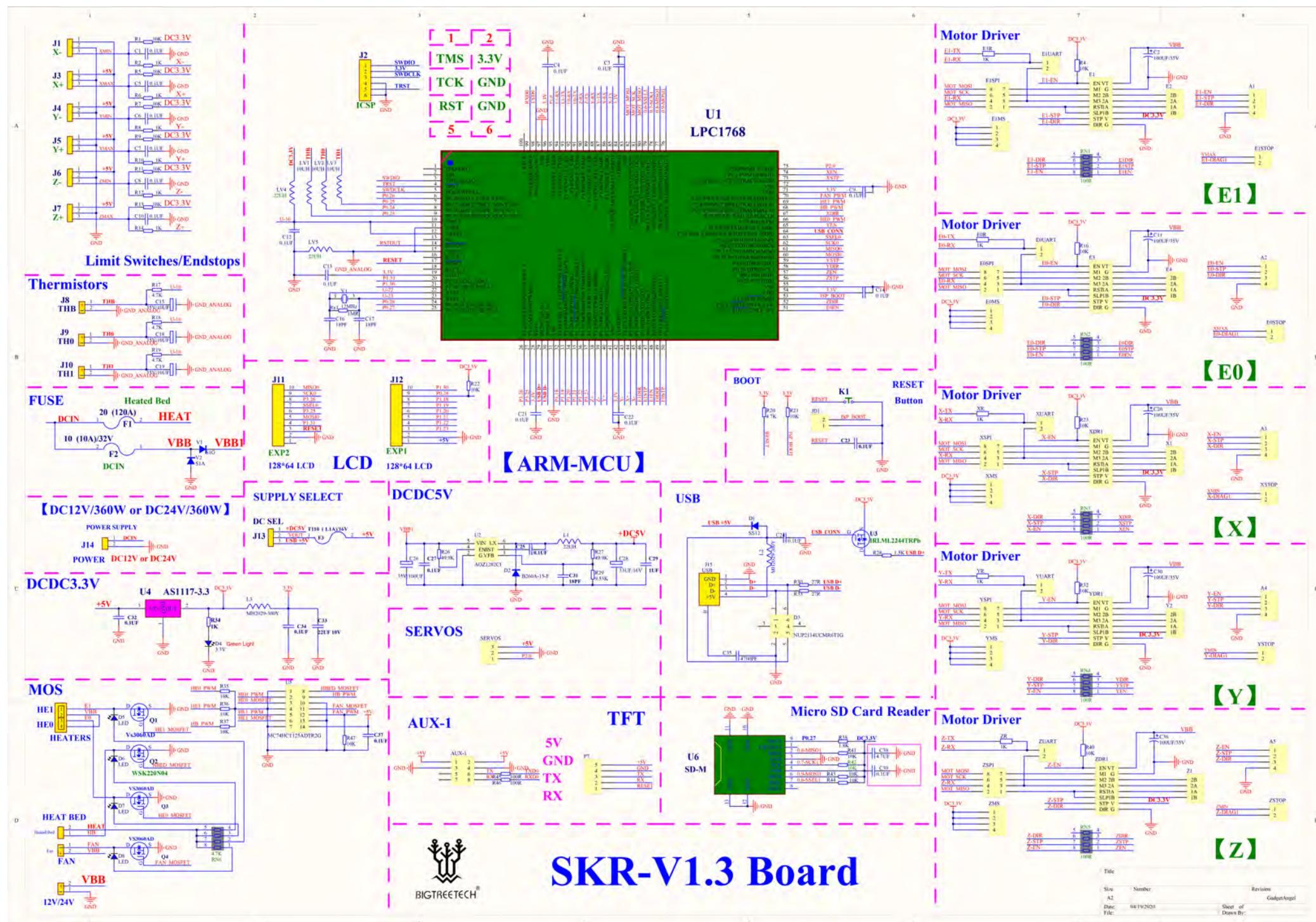
**D -DIAG PIN**

**1 -PIN 1**

- To see my updated color schematic diagram for SKR V1.3, go to the next page.



- [To see my updated uncolored schematic diagram for SKR V1.3, go to the next page.](#)

**APPENDIX C - Uncolored Schematic Diagram****SKR-V1.3 Board**

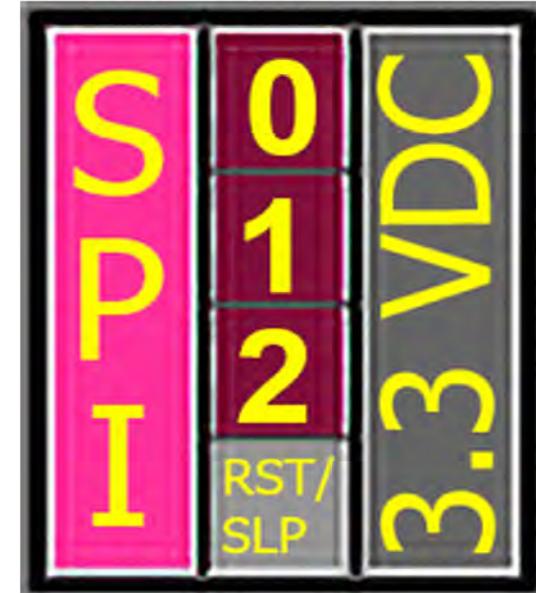
# **Legends for SKR V1.3 Stepper Driver Socket Representations**

## **Column 1 is for SPI PINS**

**Column 2 is for MS0-MS2 PINS & RST/SLP PIN**

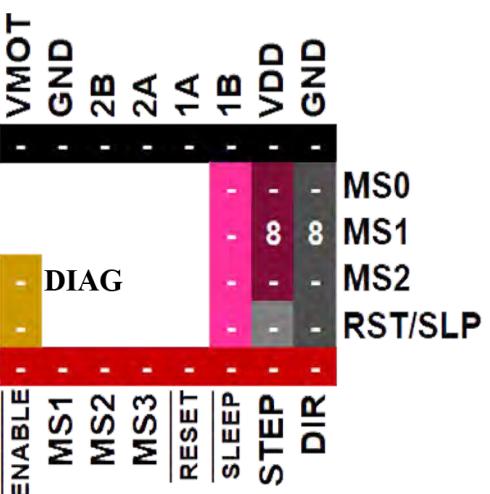
**Column 3 is for 3.3 V DC PINS**

# Columns: **1** **2** **3**



**NOTICE:** PIN labels are relative to stepper driver chip's carrier board, not the SKR V1.3 controller board. Double check driver compatibility before use.

**Number/Letter pairs denote required jumper(s). The example below indicates that you must place ONE jumper (8) across columns 2 and 3 in the MS1 row.**



## **Legend for MS0-MS2 PINS:**

**0 - MS0,M1      0 - MS1,M1  
1 - MS1,M2 OR 1 - MS2,M2  
2 - MS2,M3      2 - MS3,M3**

## Legend for Driver Socket Rep.

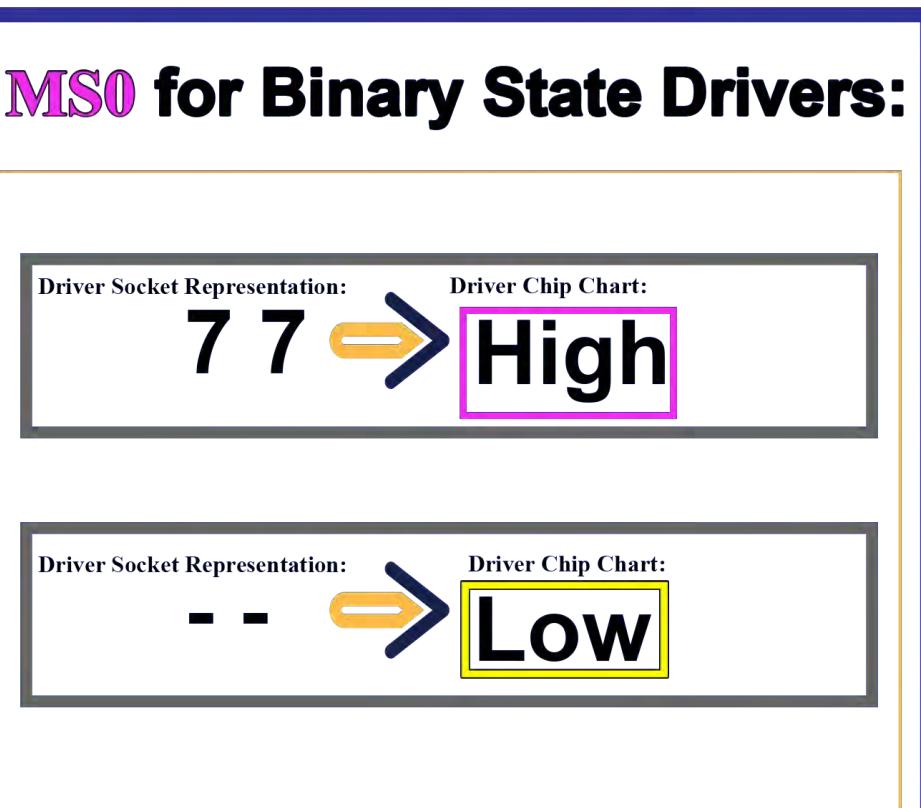
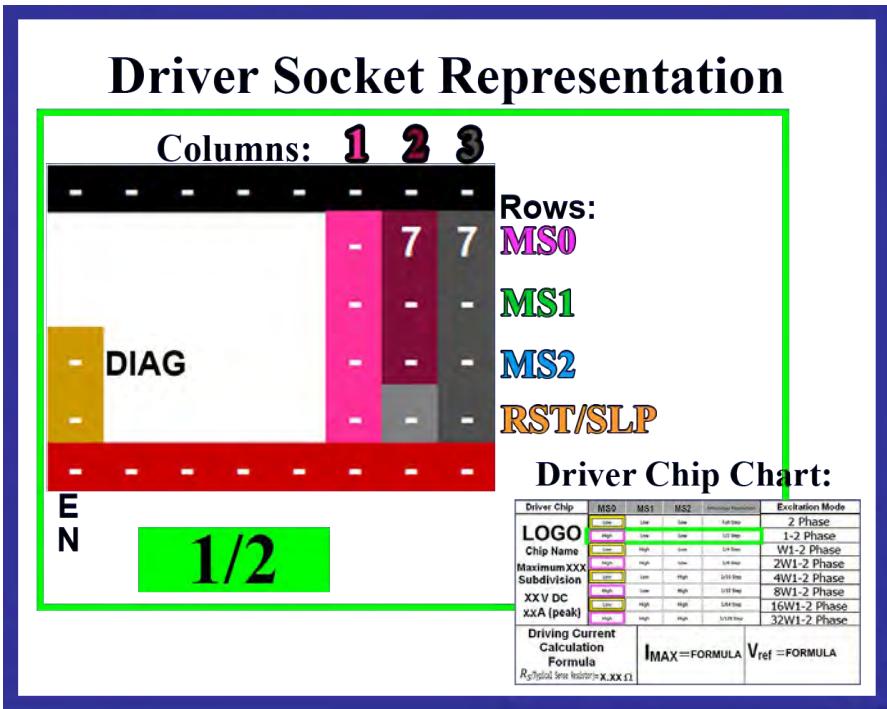
- - means: No Jumper
- 7\* - means: DuPont Jumper  
Cable to GND
- 8\* - means: DuPont Jumper  
Cable to GND

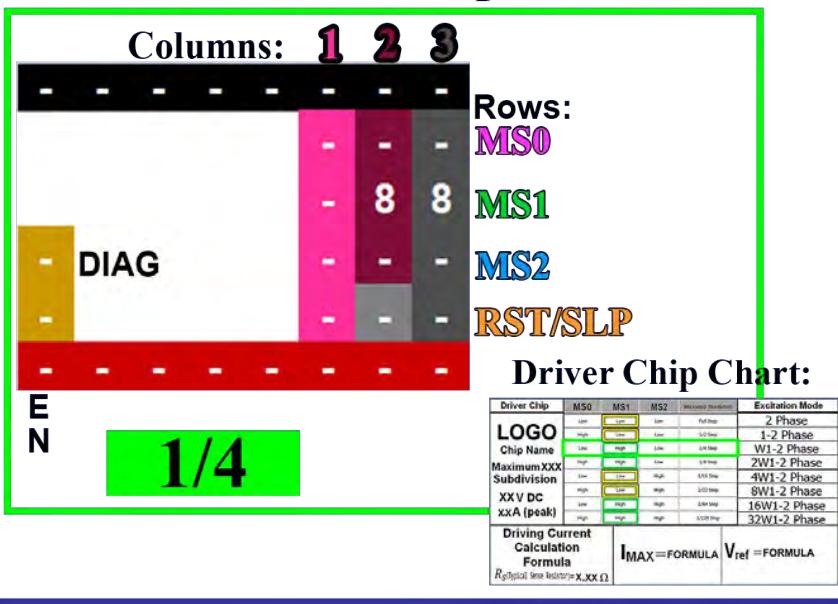
**7 7 or 8 8 or 9 9** means: Jumper set

# Columns: 1 2 3



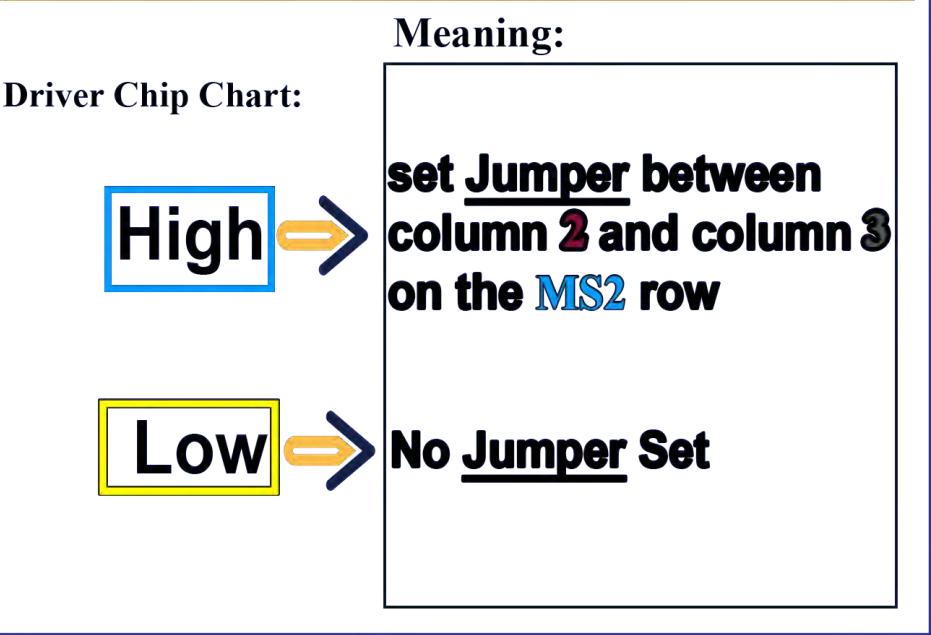
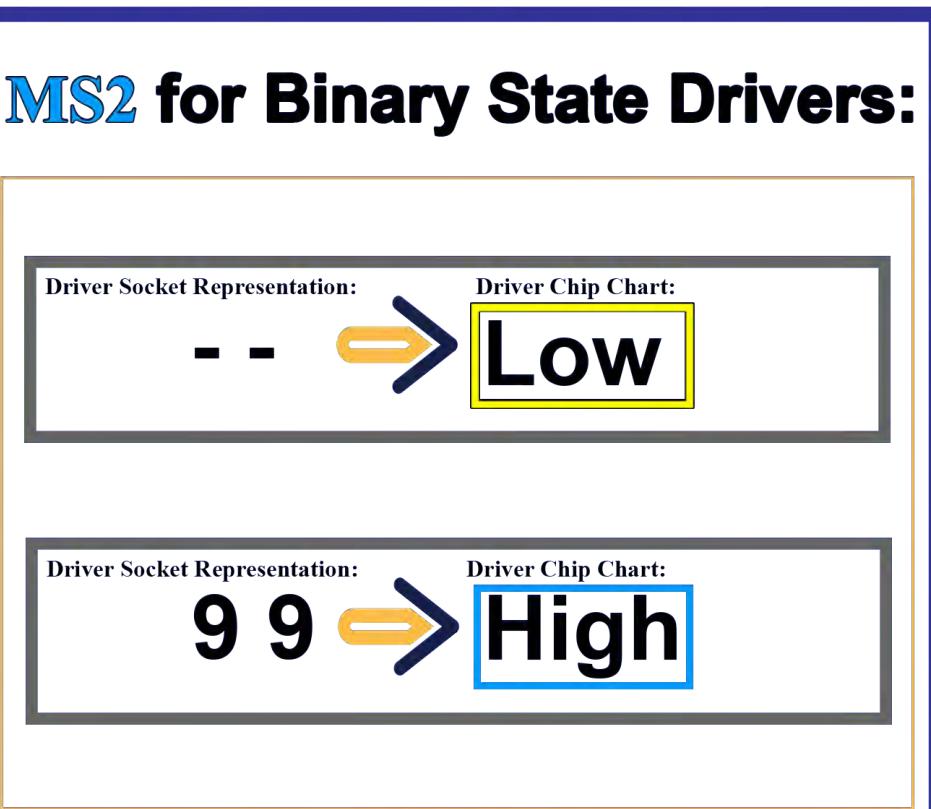
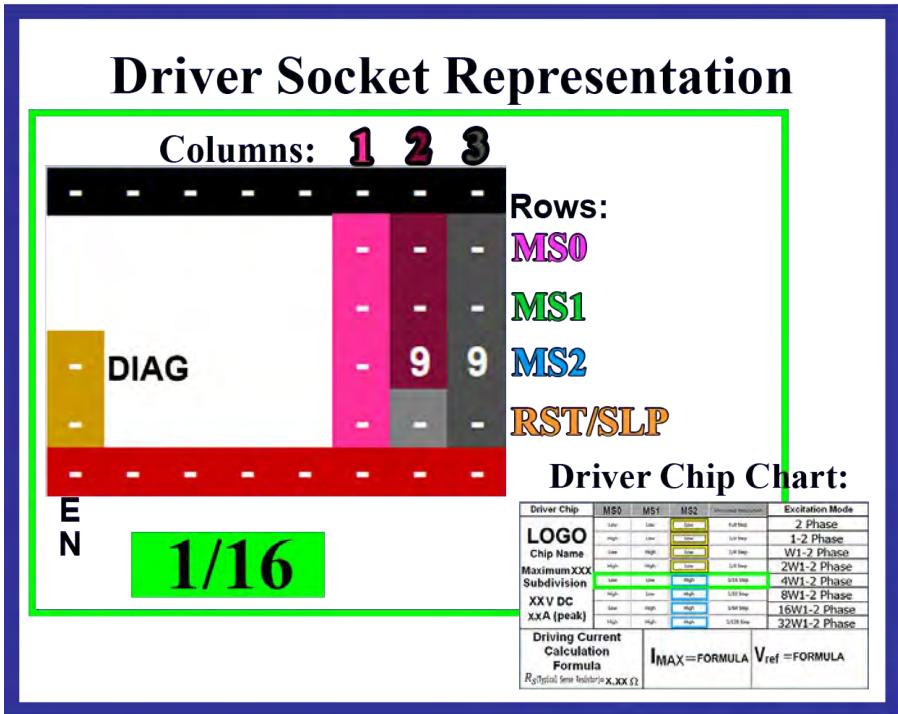
D1-DIAG1;  $V_R - V_{ref}$

**SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers**

**SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers****Driver Socket Representation****High:****MS1 for Binary State Drivers:**Driver Socket Representation: → Driver Chip Chart: **Low**Driver Socket Representation: **8 8** → Driver Chip Chart: **High**Driver Socket Representation: → Driver Chip Chart: **Low****Meaning:**

Driver Chip Chart:

**High** → **set Jumper between column 2 and column 3 on the MS1 row****Low** → **No Jumper Set**

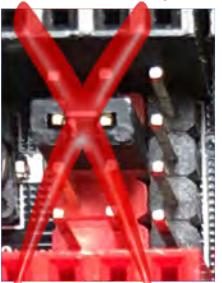
**SKR V1.3 LEGEND of Driver Socket Representation for Binary State Stepper Motor Drivers**

# SKR V1.3 LEGEND of Driver Chip Chart for Binary State Stepper Drivers

## Driver Chip Chart:

| Driver Chip                                                                  | MS0                        | MS1  | MS2                        | Microstep Resolution | Excitation Mode |
|------------------------------------------------------------------------------|----------------------------|------|----------------------------|----------------------|-----------------|
| <b>LOGO</b><br>Chip Name<br>Maximum XXX Subdivision<br>XX V DC<br>xxA (peak) | Low                        | Low  | Low                        | Full Step            | 2 Phase         |
|                                                                              | High                       | Low  | Low                        | 1/2 Step             | 1-2 Phase       |
|                                                                              | Low                        | High | Low                        | 1/4 Step             | W1-2 Phase      |
|                                                                              | High                       | High | Low                        | 1/8 Step             | 2W1-2 Phase     |
|                                                                              | Low                        | Low  | High                       | 1/16 Step            | 4W1-2 Phase     |
|                                                                              | High                       | Low  | High                       | 1/32 Step            | 8W1-2 Phase     |
|                                                                              | Low                        | High | High                       | 1/64 Step            | 16W1-2 Phase    |
|                                                                              | High                       | High | High                       | 1/128 Step           | 32W1-2 Phase    |
| Driving Current Calculation Formula<br>$R_s$ (Typical Sense Resistor)=X.XX Ω | $I_{MAX} = \text{FORMULA}$ |      | $V_{ref} = \text{FORMULA}$ |                      |                 |

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):



**Low** → No Jumper



Rows:

**MS0**

**MS1**

**MS2**

**RST/SLP**

**High** → Jumper Set

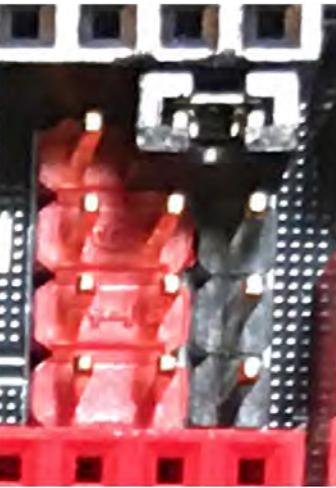
Rows:

**MS0**

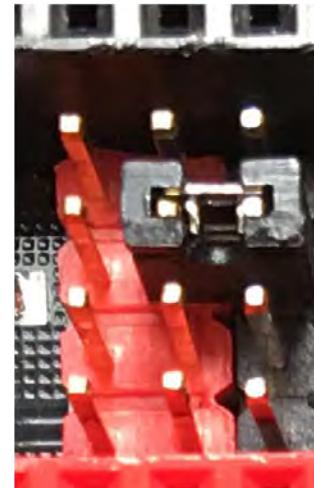
**MS1**

**MS2**

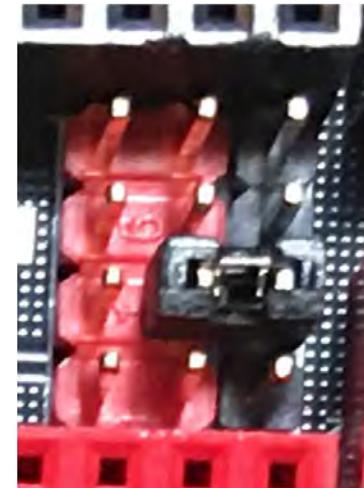
**RST/SLP**



**MS0 SET HIGH**



**MS1 SET HIGH**



**MS2 SET HIGH**

## SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers

### Driver Socket Representation

Columns: 1 2 3  
Rows: MS0 MS1 MS2 RST/SLP

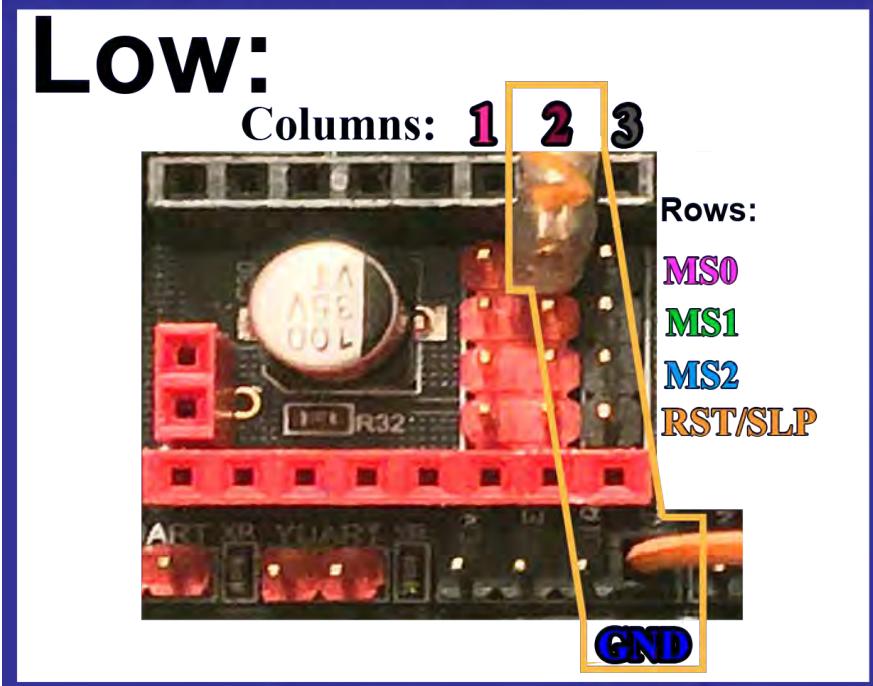
**Driver Chip Chart:**

| Driver Chip                                                                                | MS0  | MS1  | Steps | Interpolation | Mode        |
|--------------------------------------------------------------------------------------------|------|------|-------|---------------|-------------|
| <b>LOGO TMC21XX</b><br>Stand Alone Mode<br>Maximum 16 Subdivision<br>48V DC<br>2.5A (peak) | Low  | Low  | 1     | NONE          | spreadCycle |
|                                                                                            | High | Low  | 1/2   | NONE          | spreadCycle |
|                                                                                            | OPEN | Low  | 1/2   | 1/256         | spreadCycle |
|                                                                                            | LOW  | High | 1/4   | NONE          | spreadCycle |
|                                                                                            | OPEN | High | 1/4   | 1/256         | spreadCycle |
|                                                                                            | High | OPEN | 1/4   | 1/256         | stealthChop |
|                                                                                            | LOW  | High | 1/16  | NONE          | spreadCycle |
|                                                                                            | OPEN | OPEN | 1/16  | 1/256         | spreadCycle |

Driving Current Calculation Formula:  $I_{MAX} = V_{ref}$        $V_{ref} = I_{MAX}$

See Appendix B  $R_{load}$ . Use 50% to 90% as shown below:  $I_{load} = I_{MAX} * 0.90$       See Appendix B  $R_{load}$ . Use 50% to 90% as shown below:  $V_{ref} = V_{MAX} * 0.90$

**No Jumper set**



### MS0 for Tri State Drivers ONLY (TMC2100 & TMC2130):

**Driver Socket Representation:** 7 7 → **Driver Chip Chart:** High

**Driver Socket Representation:** 7\* - → **Driver Chip Chart:** Low

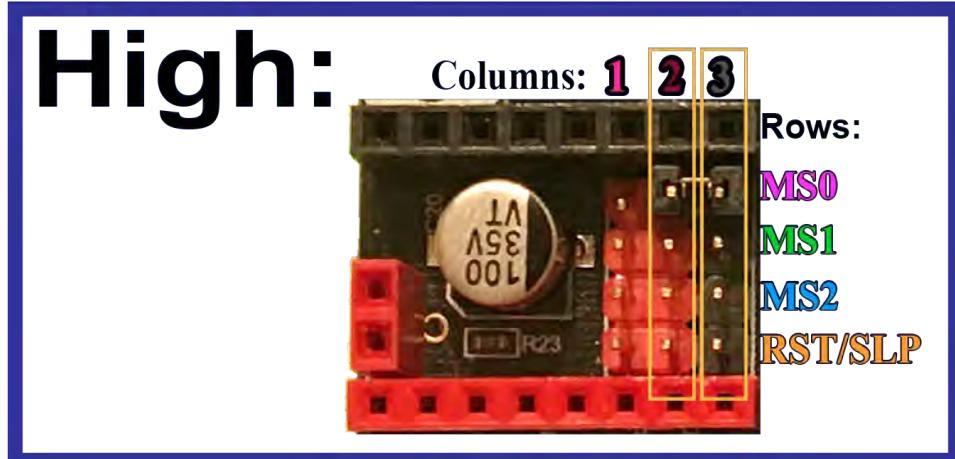
**Driver Socket Representation:** - - → **Driver Chip Chart:** Open

**Meaning:**

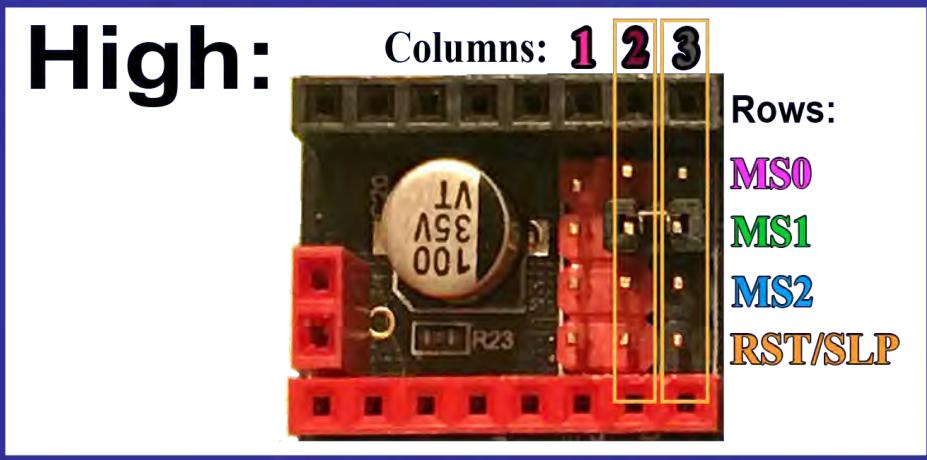
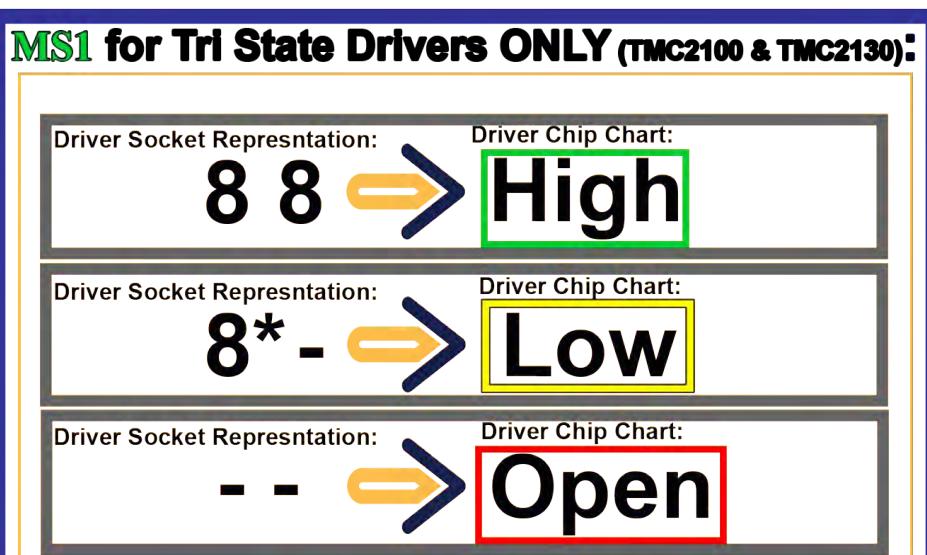
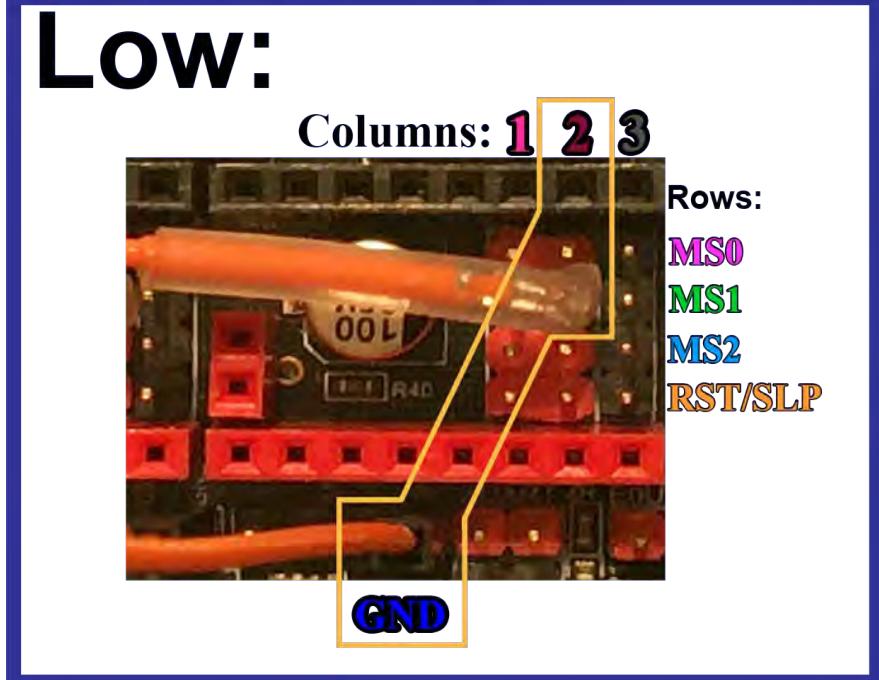
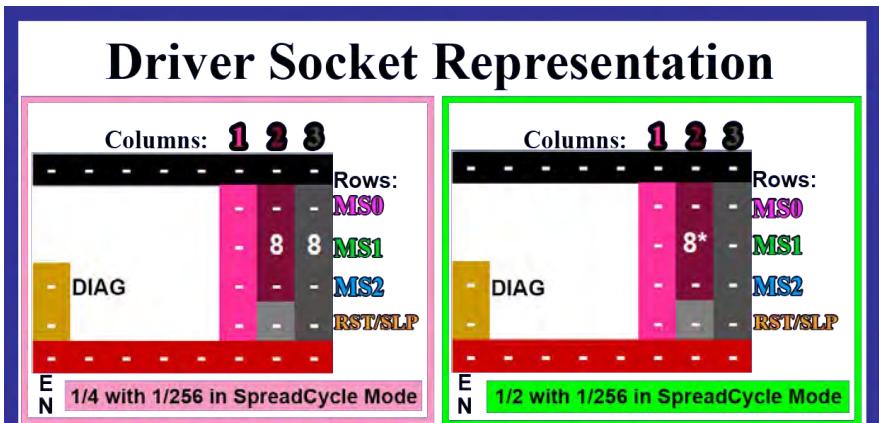
**High** → **set Jumper between column 2 and column 3 on the MS0 row**

**Low** → **Use a DuPont Jumper Cable to connect column 2 PIN from the MS0 row to ground (GND)**

**Open** → **No Jumper set**



## SKR V1.3 LEGEND of Driver Socket Representation for Tri State Stepper Motor Drivers



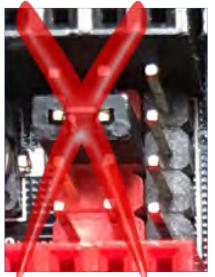
## SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers

### Driver Chip Chart:

| Driver Chip                           | MS0  | MS1  | Steps | Interpolation | Mode        |
|---------------------------------------|------|------|-------|---------------|-------------|
| LOGO                                  | Low  | Low  | 1     | NONE          | spreadCycle |
| TMCxxxx                               | High | Low  | 1/2   | NONE          | spreadCycle |
| Stand Alone Mode                      | OPEN | Low  | 1/2   | 1/256         | spreadCycle |
| Maximum 16 Subdivision                | Low  | High | 1/4   | NONE          | spreadCycle |
| 46V DC 2.5A (peak)                    | OPEN | High | 1/4   | 1/256         | spreadCycle |
| Driving Current Calculation Formula   | High | OPEN | 1/4   | 1/256         | stealthChop |
| $R_S$ (typical Sense Resistor)= 0.11Ω | High | High | 1/16  | NONE          | spreadCycle |
|                                       | Low  | OPEN | 1/16  | 1/256         | spreadCycle |
|                                       | OPEN | OPEN | 1/16  | 1/256         | stealthChop |

Driving Current Calculation Formula  
 $R_S$ (typical Sense Resistor)= 0.11Ω

Example of the **WRONG** way to set a jumper for STEP/DIR mode (left two columns of PINS):



Example of the **RIGHT** way to set a jumper for STEP/DIR mode (right two columns of PINS):



**OPEN** ➡ No Jumper



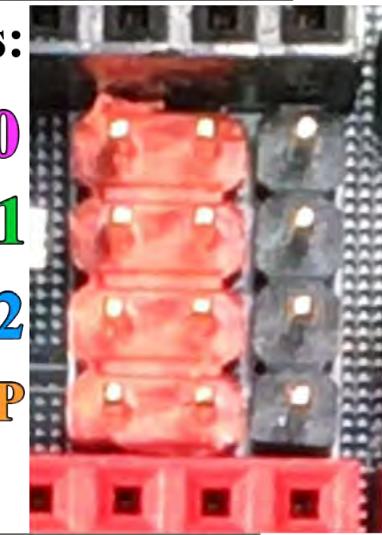
Rows:

**MS0**

**MS1**

**MS2**

**RST/SLP**



**High** ➡ Jumper Set

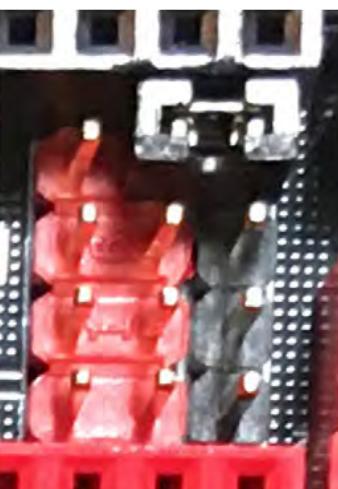
Rows:

**MS0**

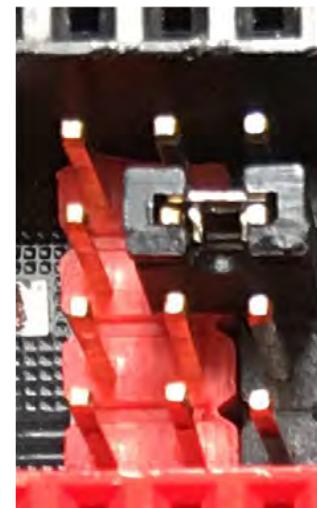
**MS1**

**MS2**

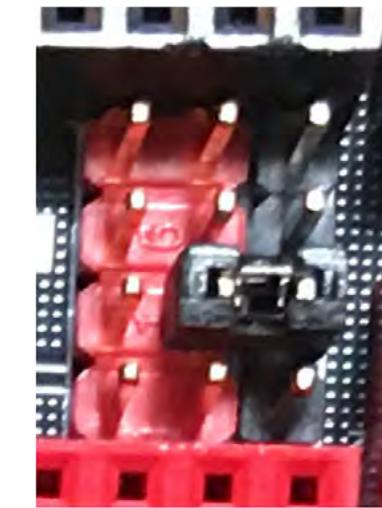
**RST/SLP**



**MS0 SET HIGH**



**MS1 SET HIGH**



**MS2 SET HIGH**

## SKR V1.3 LEGEND of Driver Chip Chart for Tri State Stepper Drivers

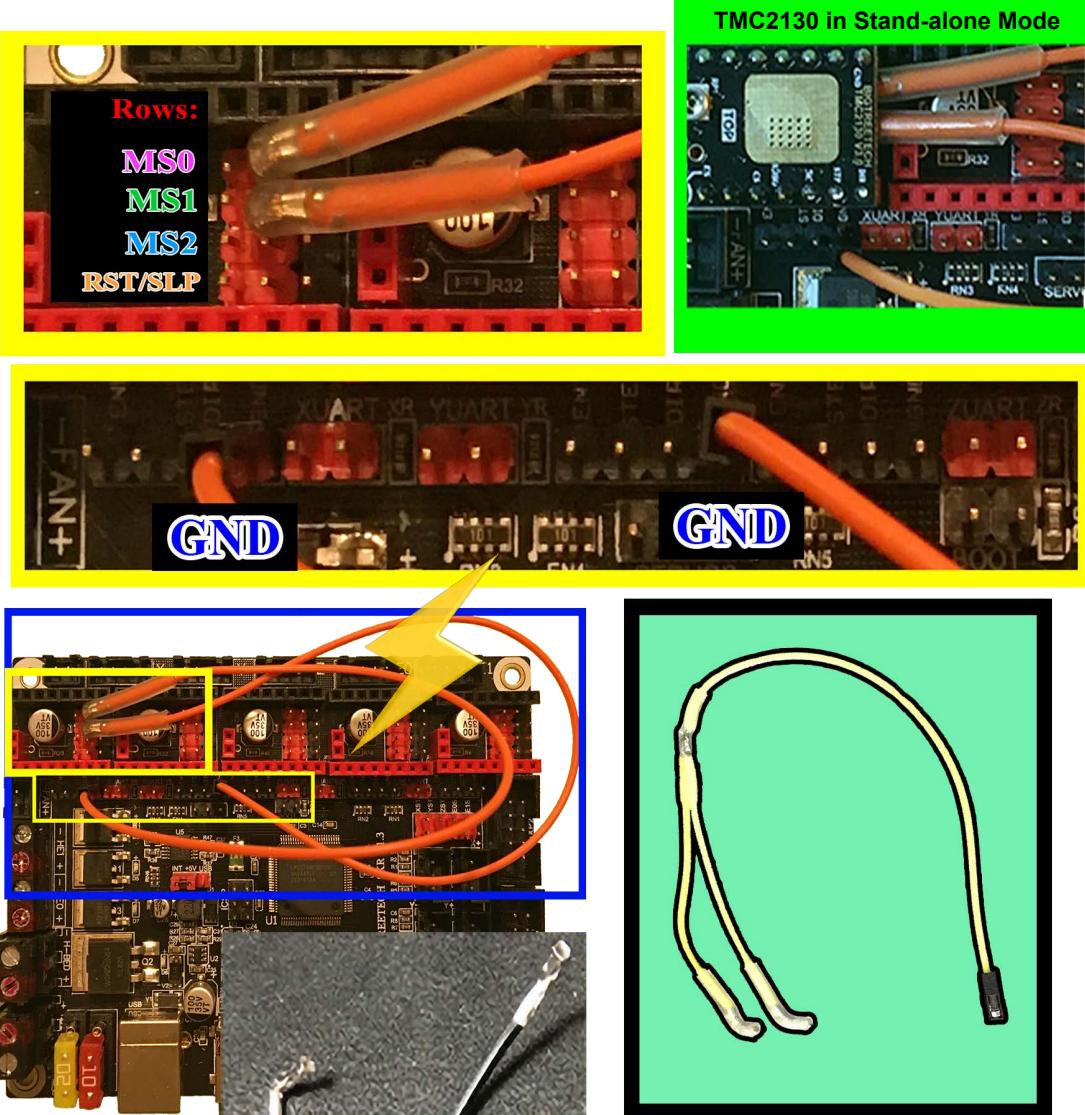
### Driver Chip Chart:

| Driver Chip                                                                                    | MS0  | MS1  | Steps | Interpolation | Mode        |
|------------------------------------------------------------------------------------------------|------|------|-------|---------------|-------------|
| <b>LOGO<br/>TMCxxxx</b><br>Stand Alone Mode<br>Maximum 16 Subdivision<br>46V DC<br>2.5A (peak) | Low  | Low  | 1     | NONE          | spreadCycle |
|                                                                                                | High | Low  | 1/2   | NONE          | spreadCycle |
|                                                                                                | OPEN | Low  | 1/2   | 1/256         | spreadCycle |
|                                                                                                | Low  | High | 1/4   | NONE          | spreadCycle |
|                                                                                                | OPEN | High | 1/4   | 1/256         | spreadCycle |
|                                                                                                | High | OPEN | 1/4   | 1/256         | stealthChop |
|                                                                                                | High | High | 1/16  | NONE          | spreadCycle |
|                                                                                                | LOW  | OPEN | 1/16  | 1/256         | spreadCycle |

**Driving Current Calculation Formula**  
 $R_S$ (Typical Sense Resistor)= 0.11Ω

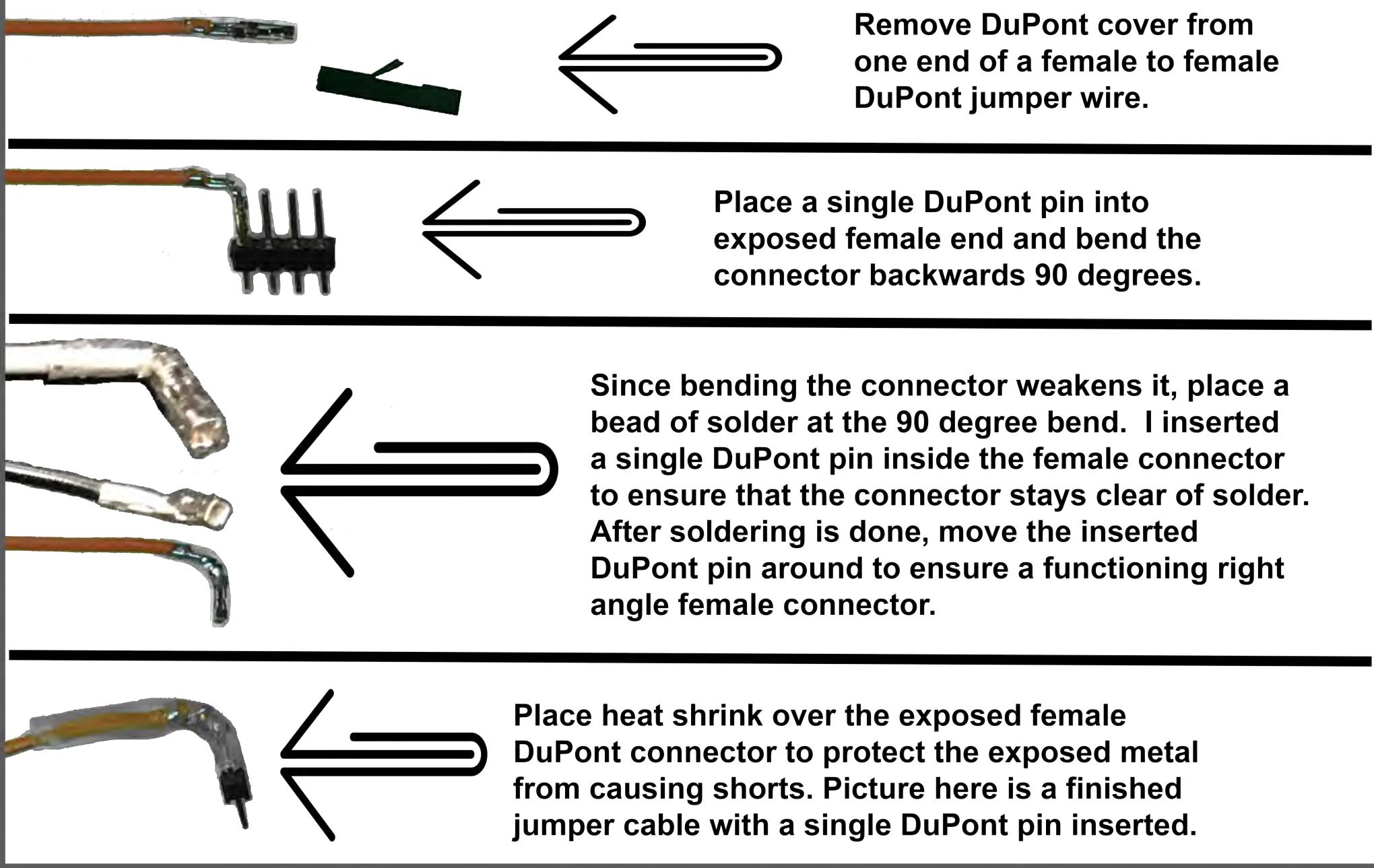
|                                                   |                                                   |
|---------------------------------------------------|---------------------------------------------------|
| $I_{MAX}=V_{ref}$                                 | $V_{ref}=I_{MAX}$                                 |
| See Appendix B #x. Use 50% to 90% as shown below: | See Appendix B #x. Use 50% to 90% as shown below: |
| $I_{MAX}=I_{MAX} * 0.90$                          | $V_{ref}=V_{ref} * 0.90$                          |

**LOW ➔ MS0 or MS1 connected to Ground(GND) via DuPont Jumper Cable**

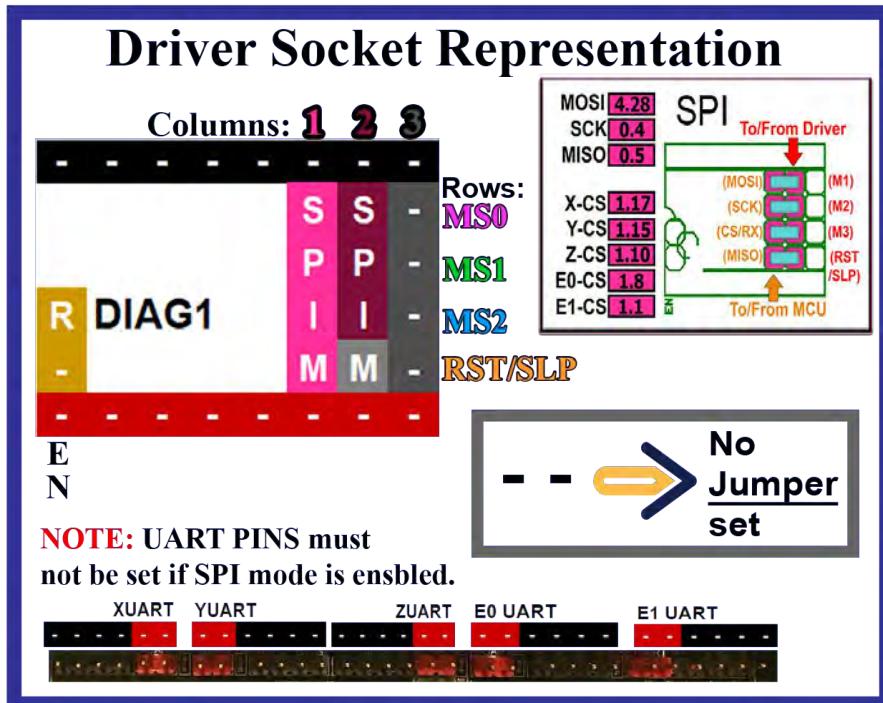


**Note:** See next page for instructions on how to create a DuPont jumper cable.

## How to Create a SKR V1.3 DuPont Jumper Cable to Use with Tri State Drivers



# SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers



## SPI Capable Drivers: TMC2130, TMC5160, and TMC5161

Driver Socket Representation:

S S  
P P  
I I  
M M

Meaning:  
**Jumpers Set**

Driver Socket Representation:

S S

Meaning:  
**set Jumper between column 1 and column 2 on the MS0 row**

Driver Socket Representation:

P P

Meaning:  
**set Jumper between column 1 and column 2 on the MS1 row**

Driver Socket Representation:

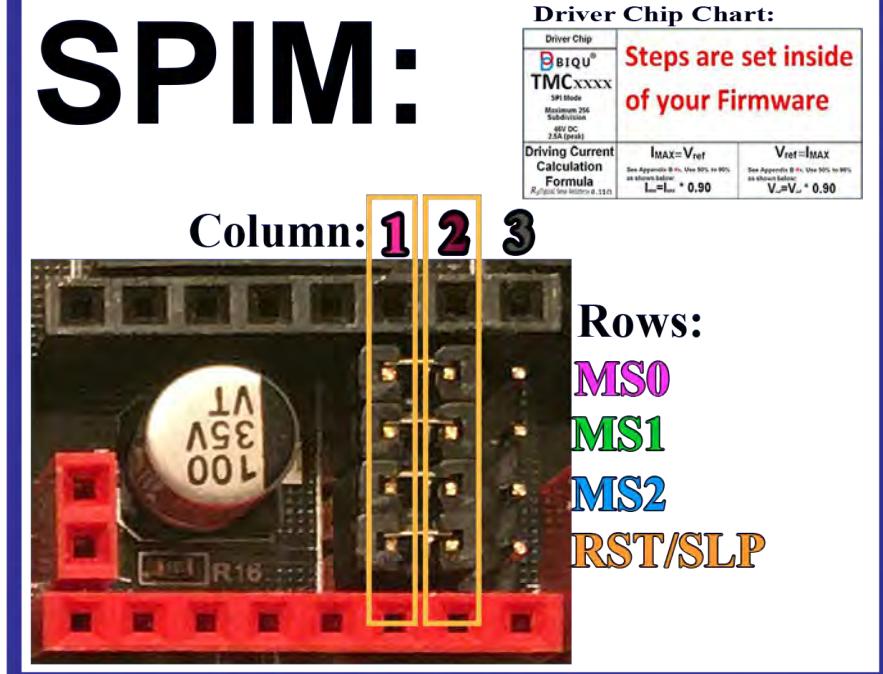
I I

Meaning:  
**set Jumper between column 1 and column 2 on the MS2 row**

Driver Socket Representation:

M M

Meaning:  
**set Jumper between column 1 and column 2 on the RST/SLP row**

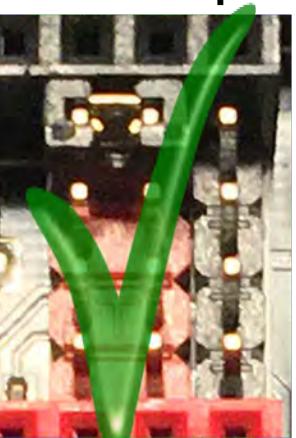


**SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers**

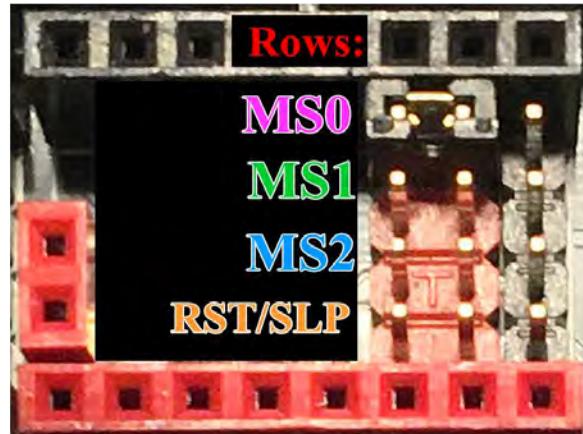
Example of the **WRONG** way to set the “S” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “S” jumper for SPI mode (left two columns of pins):



**S S → Jumper set**



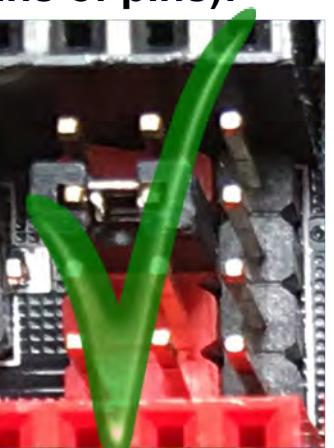
SET ‘ST’ JUMPER to enable Sensor-less Homing



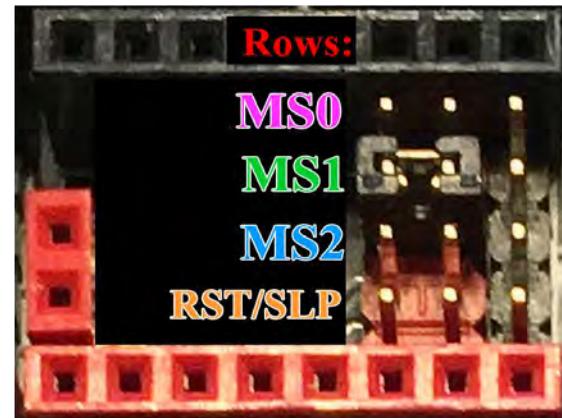
Example of the **WRONG** way to set the “P” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “P” jumper for SPI mode (left two columns of pins):



**P P → Jumper set**



SET ‘ST’ JUMPER to enable Sensor-less Homing

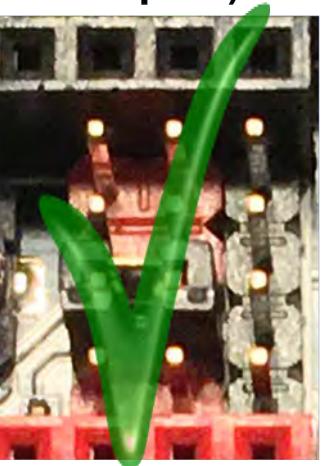


**SKR V1.3 LEGEND of Driver Socket Representation for SPI Capable Stepper Motor Drivers**

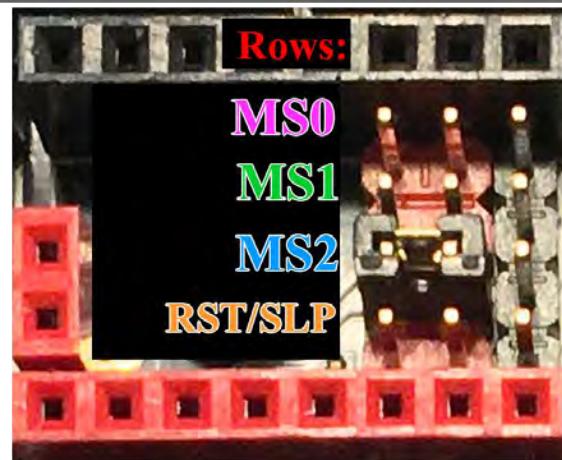
Example of the **WRONG** way to set the “**I**” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “**I**” jumper for SPI mode (left two columns of pins):



**I I → Jumper set**



SET ‘**ST**’ JUMPER to enable Sensor-less Homing



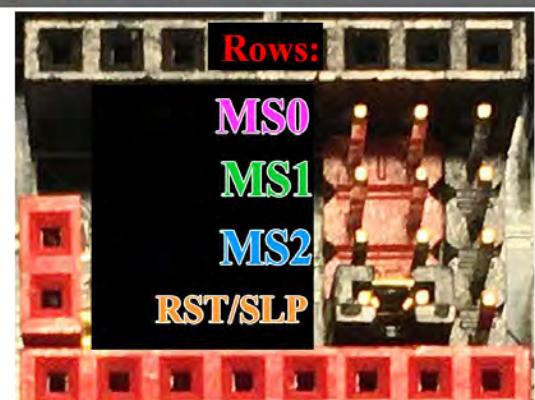
Example of the **WRONG** way to set the “**M**” jumper for SPI mode (right two columns of pins):



Example of the **RIGHT** way to set the “**M**” jumper for SPI mode (left two columns of pins):



**M M → Jumper set**

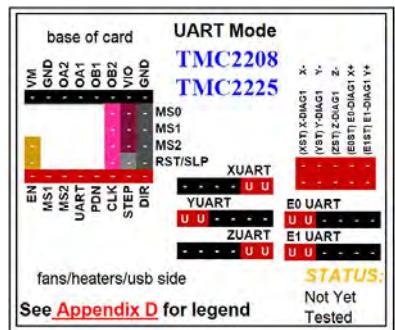


SET ‘**ST**’ JUMPER to enable Sensor-less Homing

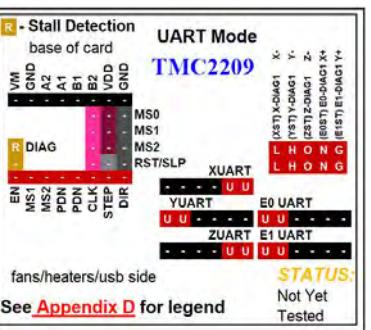


# SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers

## UART without stallGaurd™ feature



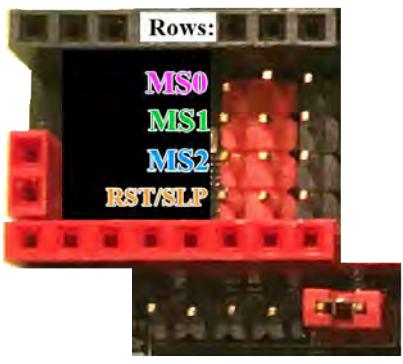
## UART with stallGaurd™ feature



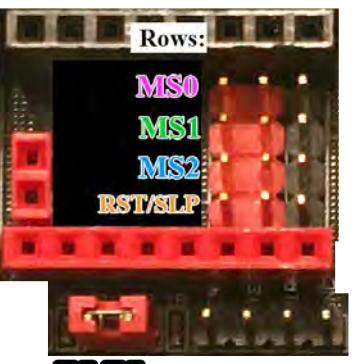
Driver Socket Representation:  
**UU** → Jumper set

**UU** → Jumper set

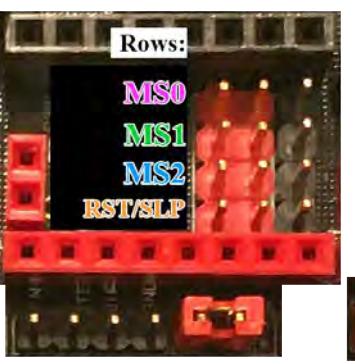
**X**



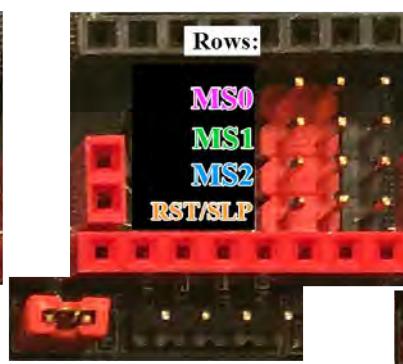
**Y**



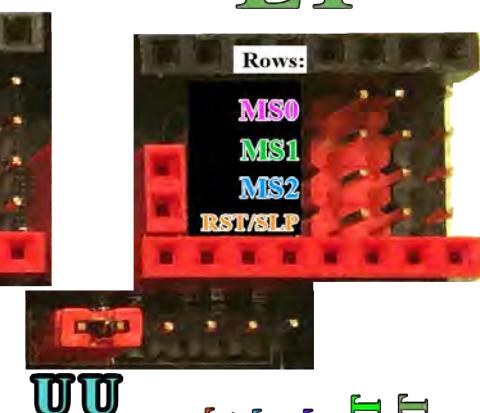
**Z**



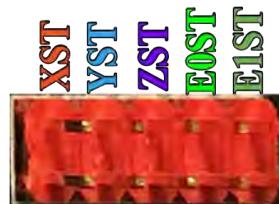
**E0**



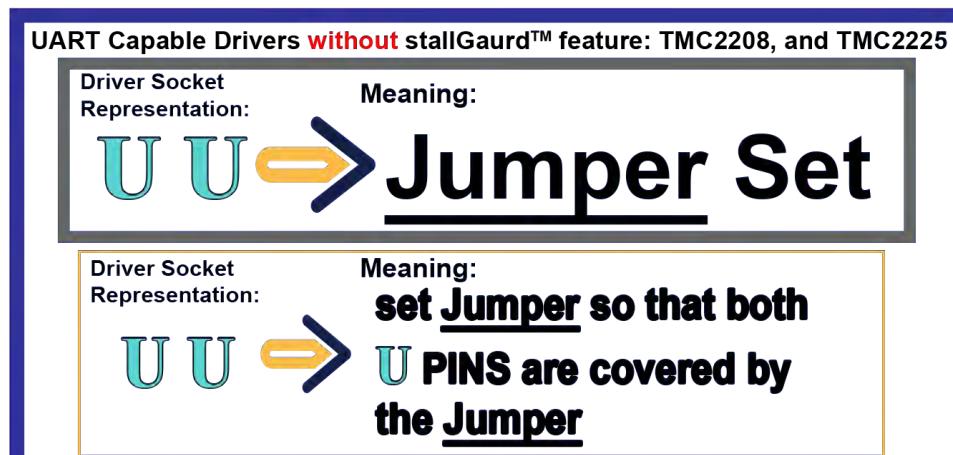
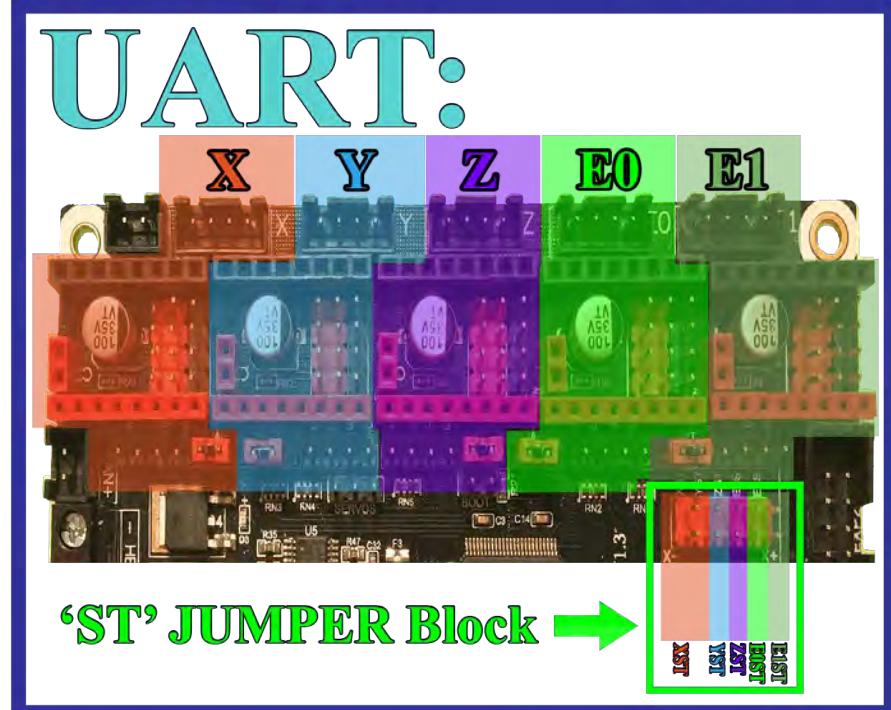
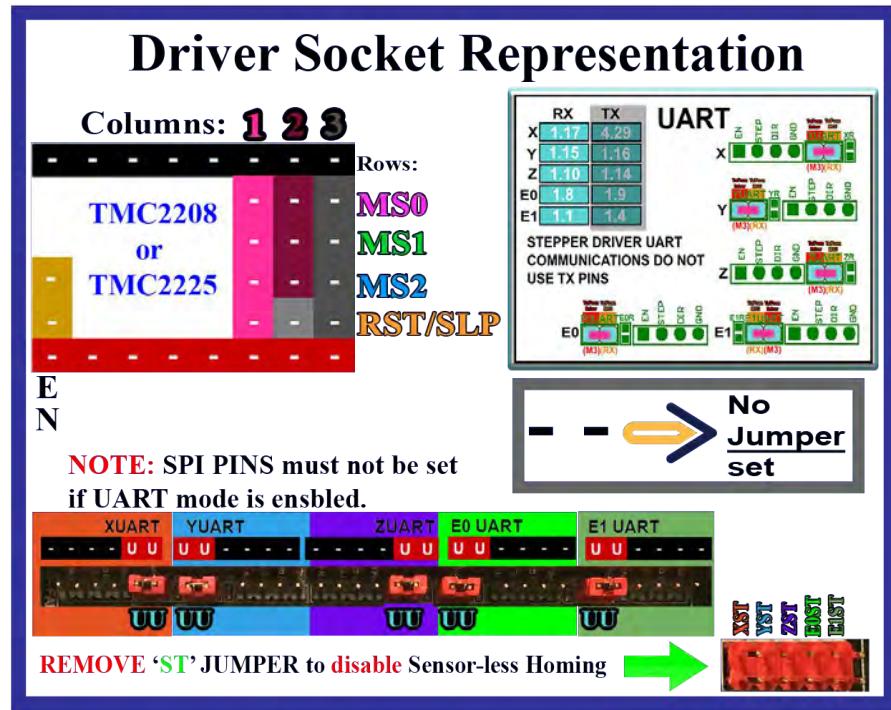
**E1**



**SET 'ST' JUMPER(s) to enable Sensor-less Homing – for drivers with stallGaurd™ feature**

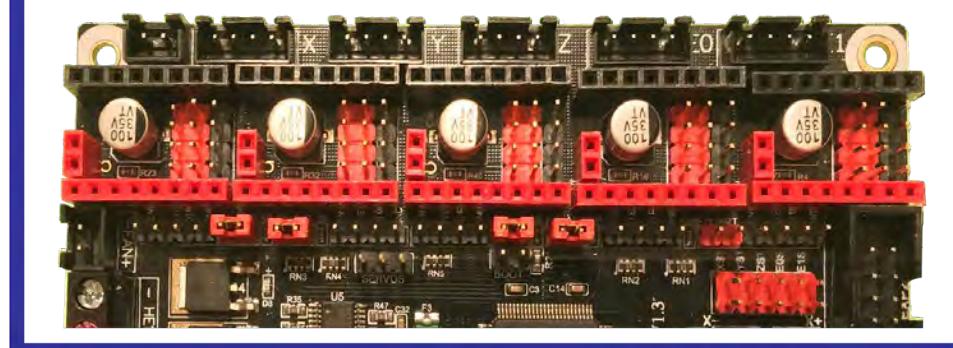


# SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers

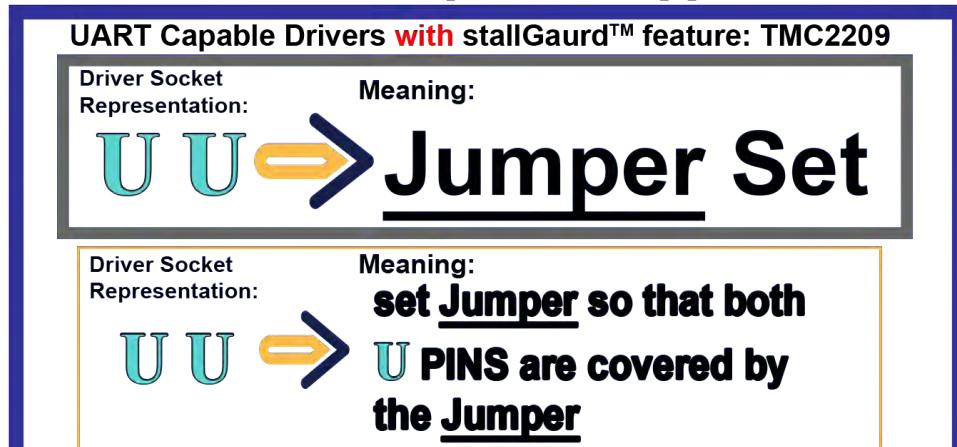
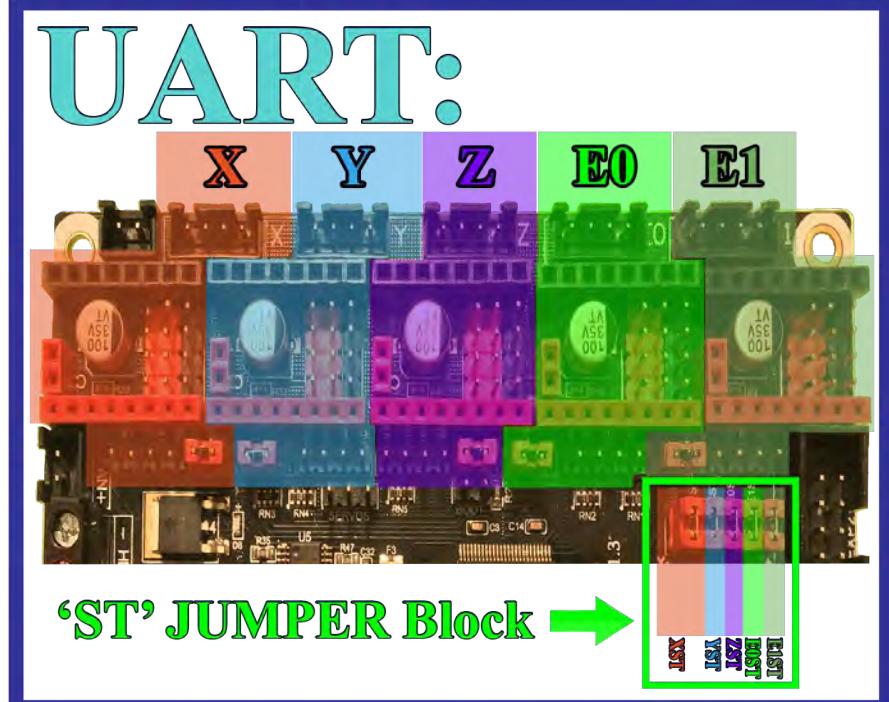
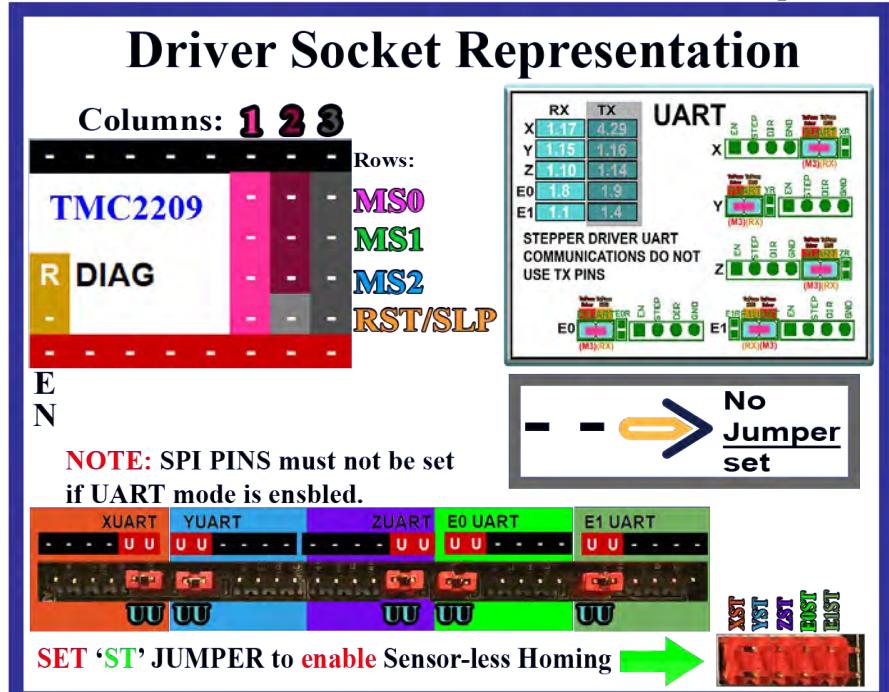


**Each Axis has its own UART PINS (UU).** You can set as many axes as you need. The **UART:** picture shows all axes having **UART enabled AND all 'ST' JUMPERS REMOVED.** All other PINS need to be empty for **UART mode to work properly.**

Here is an example of only the X, Y, Z, and E0 axes having **UART enabled AND 'ST' JUMPERS removed** for **ALL Axes:**

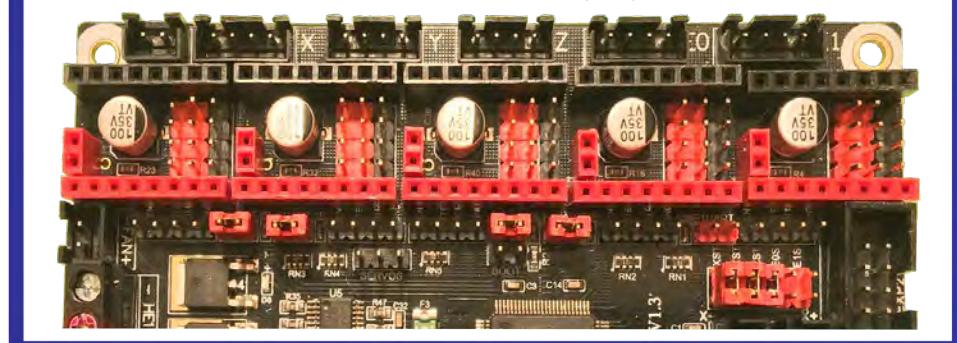


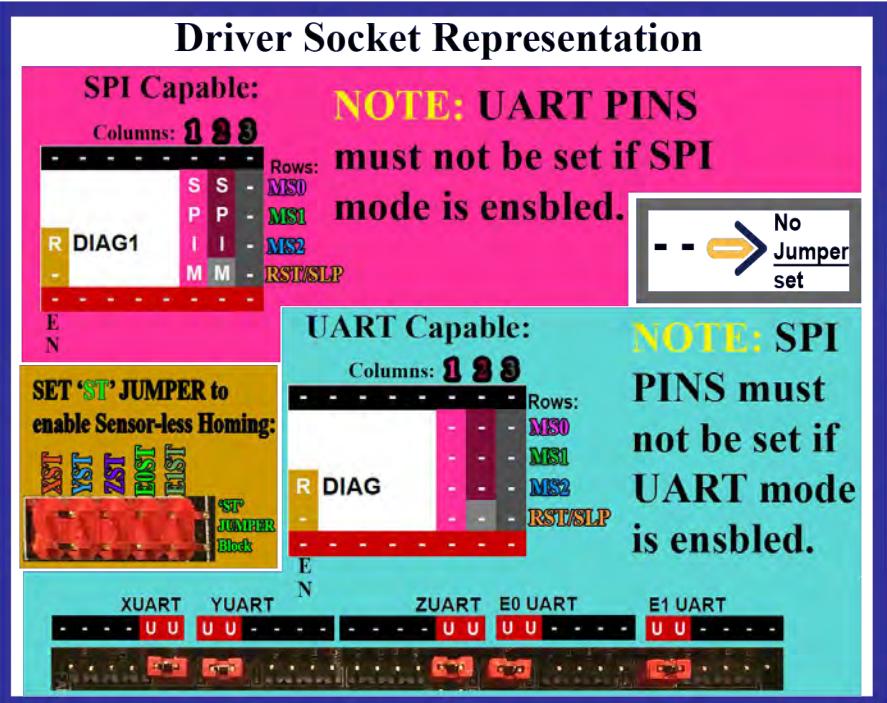
# SKR V1.3 LEGEND of Driver Socket Representation for UART Capable Stepper Motor Drivers



**Each Axis has its own UART PINS (UU).** You can set as many axes as you need. The **UART:** picture shows all axes having UART enabled **AND** all '**ST**' JUMPERS SET. All other PINS need to be empty for UART mode to work properly.

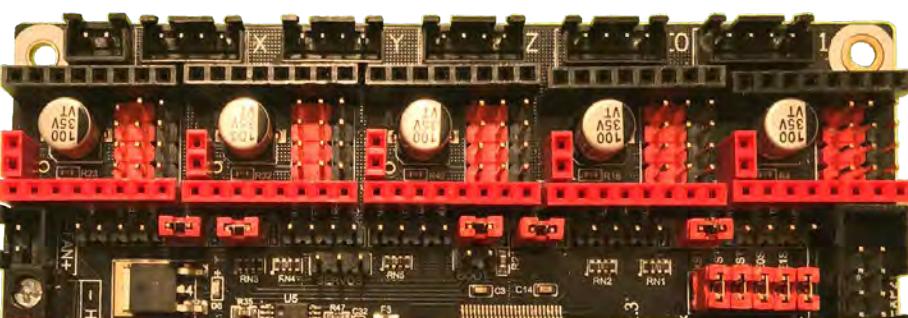
Here is an example of (**TMC2209 UART mode**) only the X, Y, Z, and E0 axes having UART enabled **AND** '**ST**' JUMPERS set for X, Y, and Z Axes:



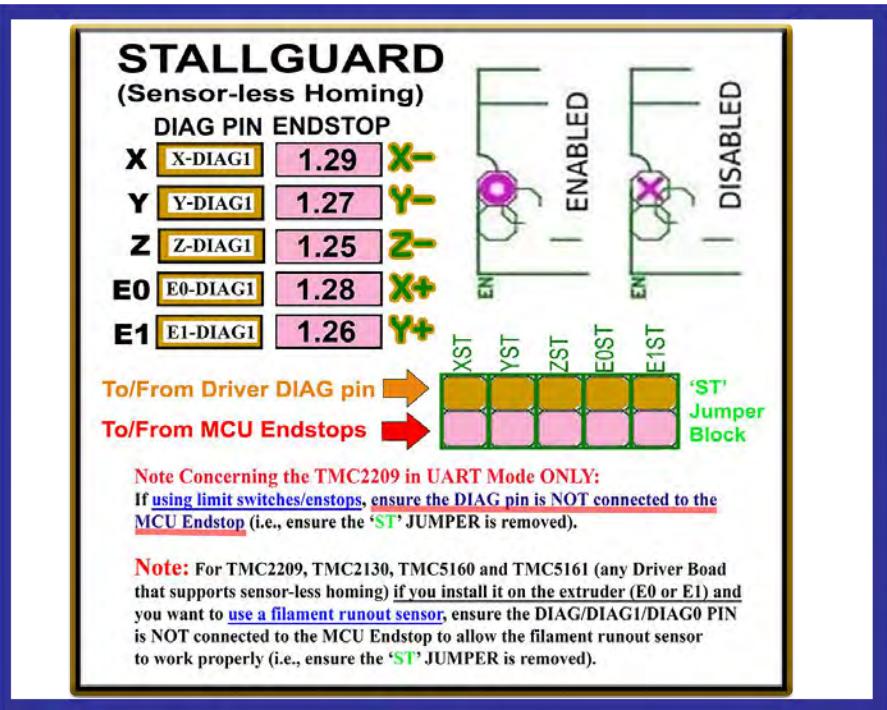
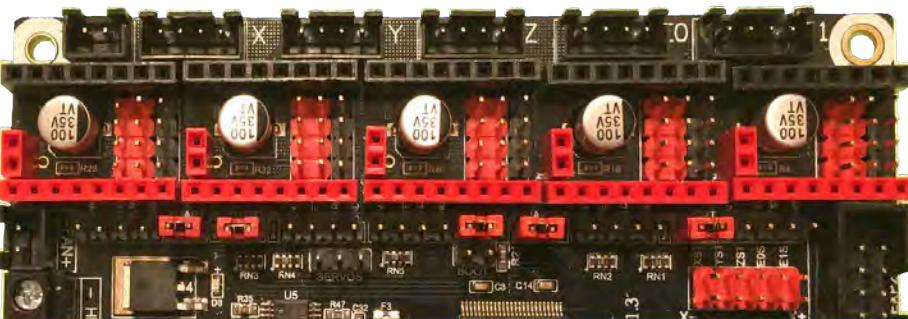
**SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers****Sensor-less Homing Capable Drivers:**

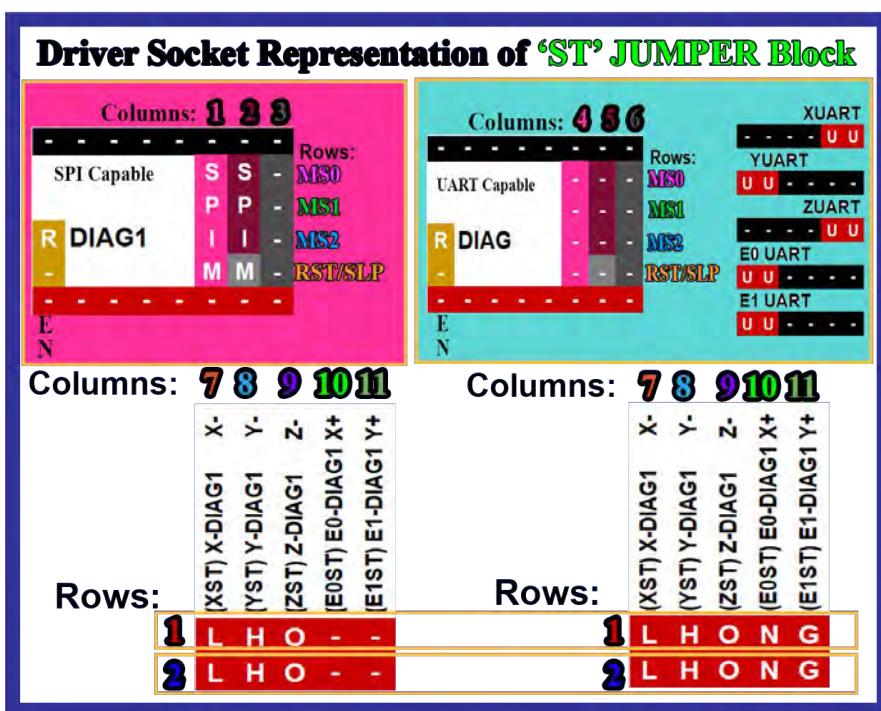
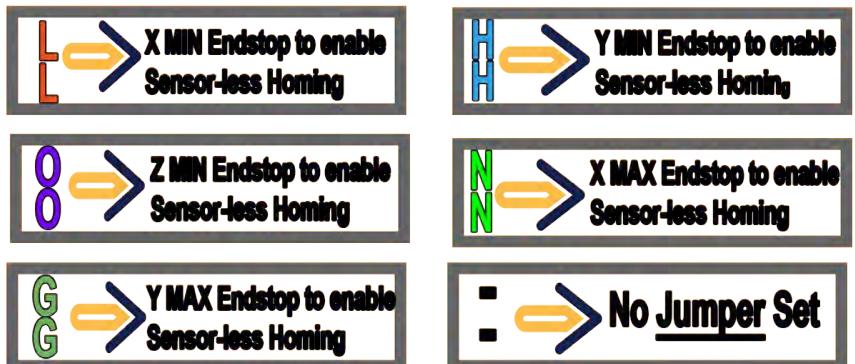
**SPI Capable Drivers:** TMC2130, TMC5160 & TMC5161  
**UART Capable Drivers:** TMC2209

If you want sensor-less homing on an axis, ensure that the 'ST' JUMPER is SET in 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are SET for ALL Axes:



If you do not want sensor-less homing (or you want to use limit switches/endstops) on any particular axis, ensure that the 'ST' JUMPER is removed from the 'ST' JUMPER Block. The example below shows ALL Axes have UART set and the 'ST' JUMPERS are REMOVED for ALL Axes:



**SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers****Sensor-less Homing Capable Drivers:****SPI Capable Drivers:** TMC2130, TMC5160 & TMC5161**UART Capable Drivers:** TMC2209

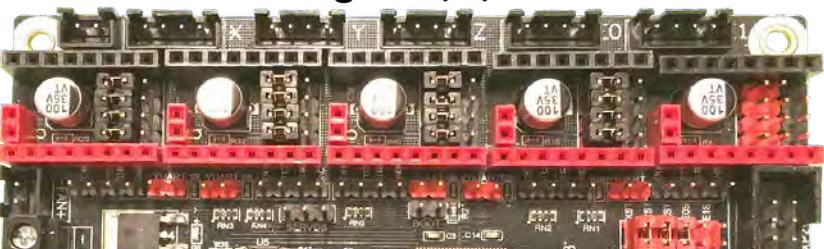
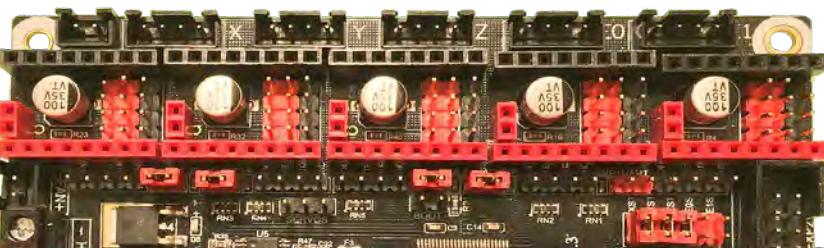
**Driver Socket Representation:** Meaning: **set Jumper between row 1 and row 2 in column 7**

**Driver Socket Representation:** Meaning: **set Jumper between row 1 and row 2 in column 8**

**Driver Socket Representation:** Meaning: **set Jumper between row 1 and row 2 in column 9**

**Driver Socket Representation:** Meaning: **set Jumper between row 1 and row 2 in column 10**

**Driver Socket Representation:** Meaning: **set Jumper between row 1 and row 2 in column 11**

**Sensor-less Homing on X, Y, & Z via SPI mode:****Sensor-less Homing on X, Y, & Z via UART mode:**

## SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers

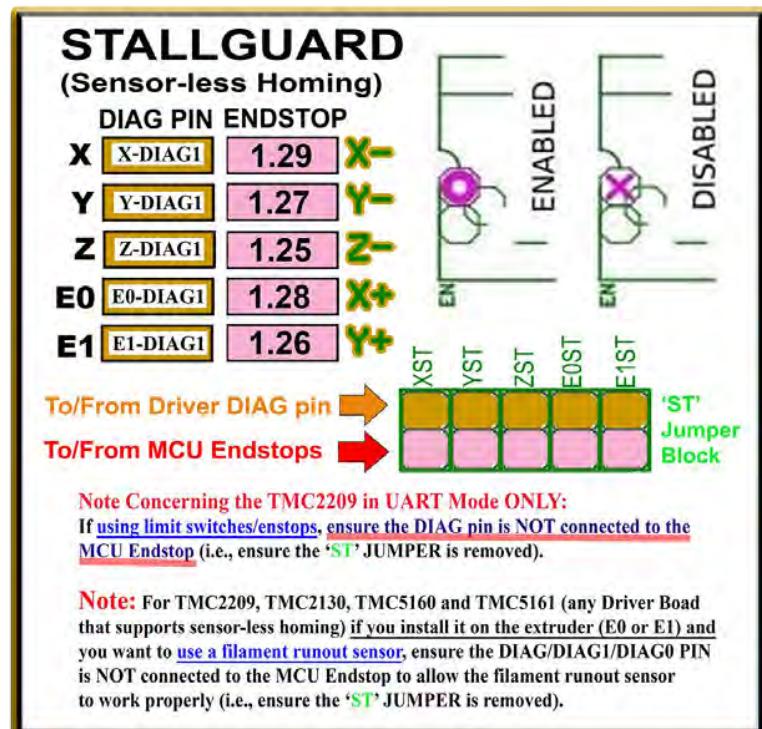
Use the 'ST' Jumper Block, when you have a stepper driver board that is capable of Sensor-less Homing (i.e., TMC2209, TMC2130, TMC5160 and TMC5161). The 'ST' Jumper Block will allow you to connect the DIAG/DIAG1/DIAG0 PIN of the stepper driver board to the MCU Endstop for that Axis. Connecting the DIAG PIN to the MCU Endstops enables the Sensor-less Homing capability of the stepper driver board (TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode and TMC5161 in SPI mode). So, if you WANT Sensor-less Homing enabled for a driver capable of Sensor-less Homing (TMC2209, TMC2130, TMC5160 or TMC5161), PLACE a 'ST' JUMPER in the 'ST' Jumper Block for that Axis.

The way you ensure the DIAG PIN is NOT connected to the MCU Endstop for the Axis is by ensuring the corresponding 'ST' JUMPER is removed from the 'ST' Jumper Block for that particular Axis.

**NOTE:** The TMC2208 in UART mode and the TMC2225 in UART mode do not have stallGuard™. Therefore, TMC2208 and TMC2225 CAN NOT be used for Sensor-less Homing. Please read the PREFACE to this manual on "Stall detection and Sensor-less Homing".

If you are using the TMC2209 in UART mode AND you still want to use limit switches/endstops on that Axis, ensure the DIAG PIN from the stepper driver board, is NOT connected to the MCU Endstop (i.e., remove the 'ST' JUMPER in the 'ST' Jumper Block for that Axis).

If you are using a TMC2209, TMC2130, TMC5160 or TMC5161 in the extruder (E0 or E1) stepper driver location AND you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that axis).



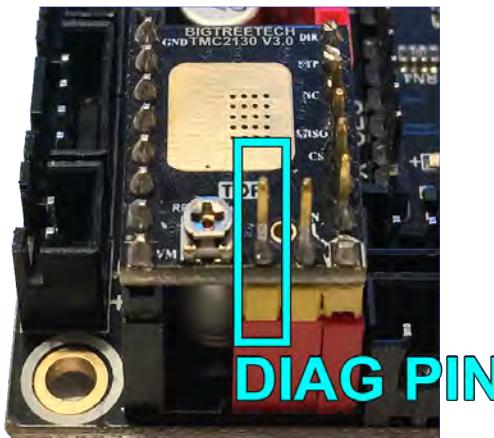
If you are using TMC2209 in UART mode, TMC2130 in SPI mode, TMC5160 in SPI mode or TMC5161 in SPI mode AND you DO NOT want to use the Sensor-less Homing capabilities of the stepper driver ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block for that Axis. This will allow you to use physical Limit switches/Endstops for the Axis. If the Axis does not have an Endstop, then ensure the 'ST' JUMPER is removed from the 'ST' Jumper Block.

The following DO NOT have Endstops:

- Extruder Axis (E0 or E1)
- Z Axis, if a BLTouch is used to Home instead of a physical endstop

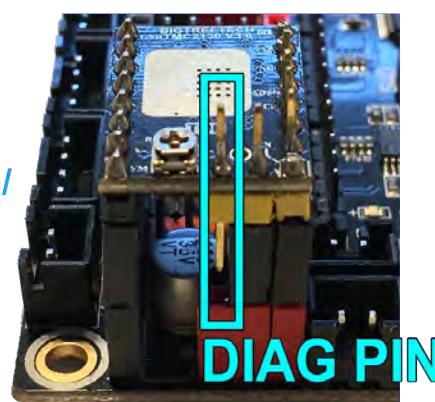
**SKR V1.3 LEGEND of Driver Socket Representation for Sensor-less Homing Capable Stepper Motor Drivers**

To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board AND** the '**ST'** **JUMPER** is set in the '**ST' JUMPER** Block for the Axis.



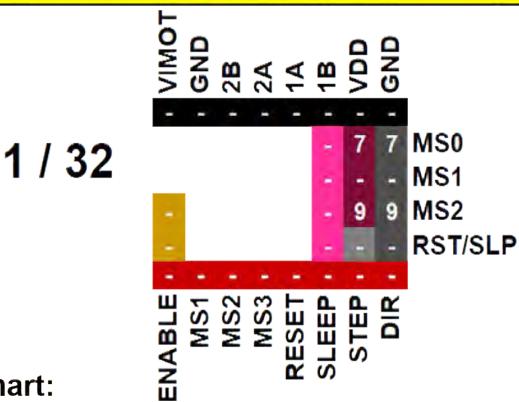
To **disable** sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the **SKR V1.3 Board OR** remove the '**ST' JUMPER** from the '**ST' JUMPER** Block.

Link to stackable header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>



APPENDIX DExamples of Driver Socket Representations

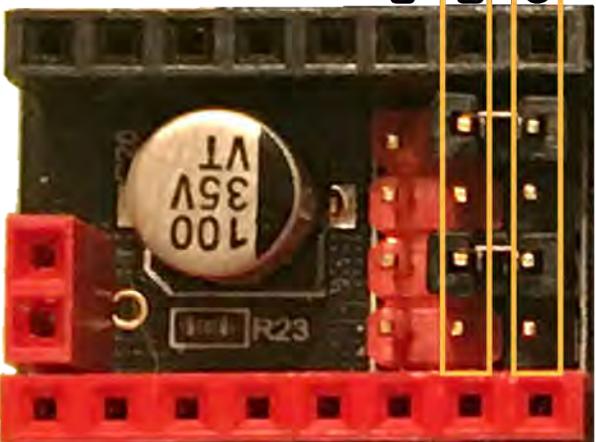
**Example 1 (LV8729 Driver Board; Binary State Driver ) for SKR V1.3 Driver Socket Representation:**



Driver Chip Chart:

| Driver Chip | MS0  | MS1 | MS2  | Microstep Resolution |
|-------------|------|-----|------|----------------------|
|             | Low  | Low | Low  | Full Step            |
| LV8729      | High | Low | Low  | 1/2 Step             |
|             | High | Low | High | 1/32 Step            |

Columns: 1 2 3

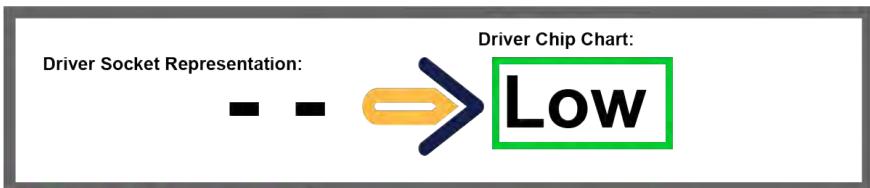


Rows:

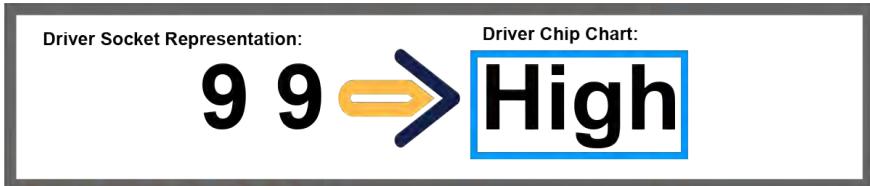
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**



**MS0 PIN:**  
High → set Jumper between column 2 and 3 on MS0 row

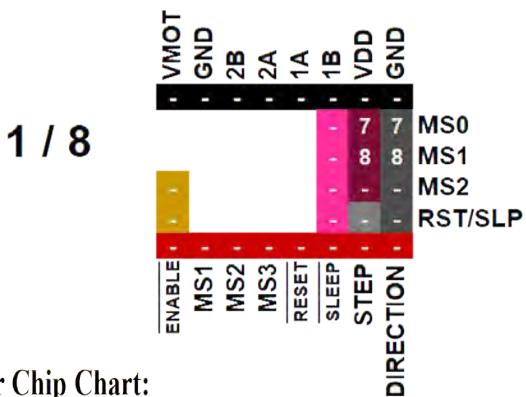


**MS1 PIN:**  
Low → No Jumper Set



**MS2 PIN:**  
High → set Jumper between column 2 and 3 on MS2 row

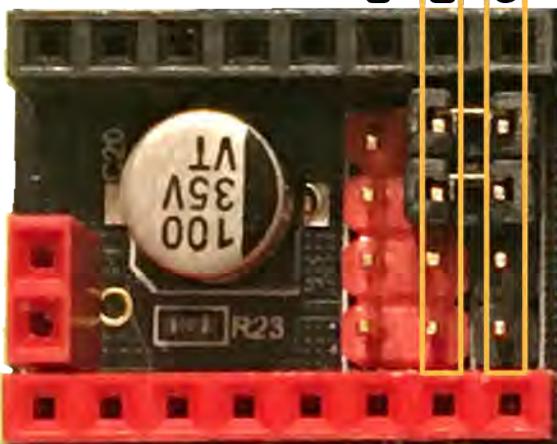
## Example 2 (A4988 Driver Board; Binary State Driver) for SKR V1.3 Driver Socket Representation:



Driver Chip Chart:

| Driver Chip                                                                        | MS0                       | MS1  | MS2                     | Microstep Resolution | Excitation Mode |
|------------------------------------------------------------------------------------|---------------------------|------|-------------------------|----------------------|-----------------|
| BIQU® A4988                                                                        | Low                       | Low  | Low                     | Full step            | 2 Phase         |
| Maximum 16 Subdivision                                                             | High                      | Low  | Low                     | Half step            | 1-2 Phase       |
| 35V DC 2A (peak)                                                                   | Low                       | High | Low                     | Quarter step         | W1-2 Phase      |
|                                                                                    | High                      | High | Low                     | Eighth step          | 2W1-2 Phase     |
|                                                                                    | High                      | High | High                    | Sixteenth step       | 4W1-2 Phase     |
| Driving Current Calculation Formula $R_S(\text{typical Sense Resistor})=0.1\Omega$ | $I_{MAX}=V_{ref}/(8*R_S)$ |      | $V_{ref}=8*I_{MAX}*R_S$ |                      |                 |

Columns: 1 2 3



Rows:  
MS0  
MS1  
MS2  
RST/SLP

Driver Socket Representation:

7 7 → High

Driver Chip Chart:

MS0 PIN: set Jumper between column 2 and 3 on MS0 row  
High → set Jumper between column 2 and 3 on MS0 row

Driver Socket Representation:

8 8 → High

Driver Chip Chart:

MS1 PIN: set Jumper between column 2 and 3 on MS1 row  
High → set Jumper between column 2 and 3 on MS1 row

Driver Socket Representation:

- - → Low

Driver Chip Chart:

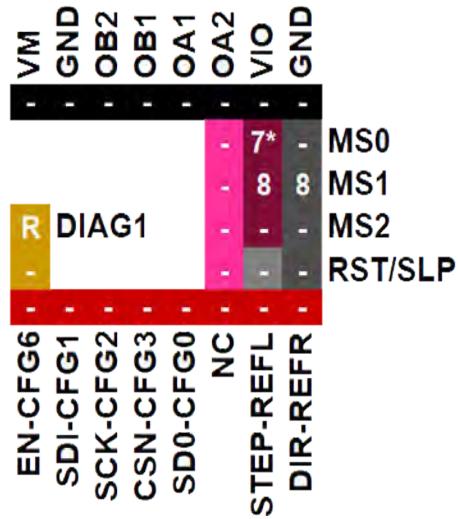
MS2 PIN:

Low → No Jumper Set

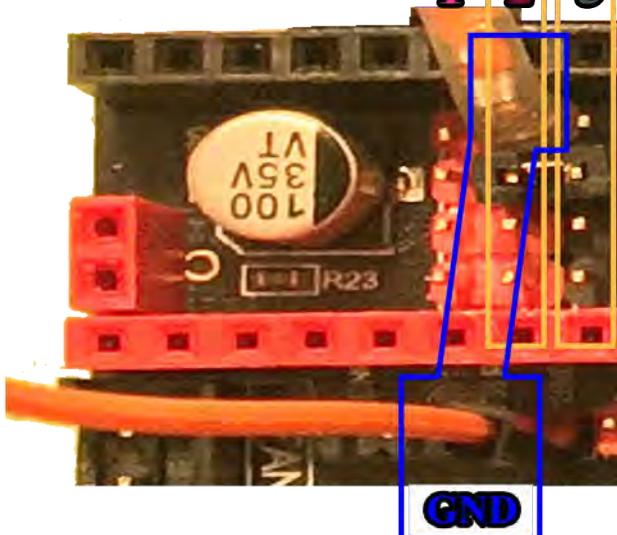
### Example 3 (TMC2130 Stand-alone Mode; Tri State Driver) for SKR V1.3 Driver Socket Representation:

Stand-alone Mode

1 / 4

Interpolation: none  
SpreadCycle

Columns: 1 2 3



Rows:  
**MS0**  
**MS1**  
**MS2**  
**RST/SLP**

Driver Socket Representation:

7\* - **LOW**

Driver Chip Chart:

7\* - **LOW**

**MS0 PIN:**

**Use a DuPont Jumper Cable**  
**LOW** **to connect column 2 PIN from the MS0 row to ground (GND)**

Driver Socket Representation:

8 8 **High**

Driver Chip Chart:

8 8 **High**

**MS1 PIN:**

**set Jumper between column 2 and 3 on MS1 row**  
**High** **High**

Driver Socket Representation:

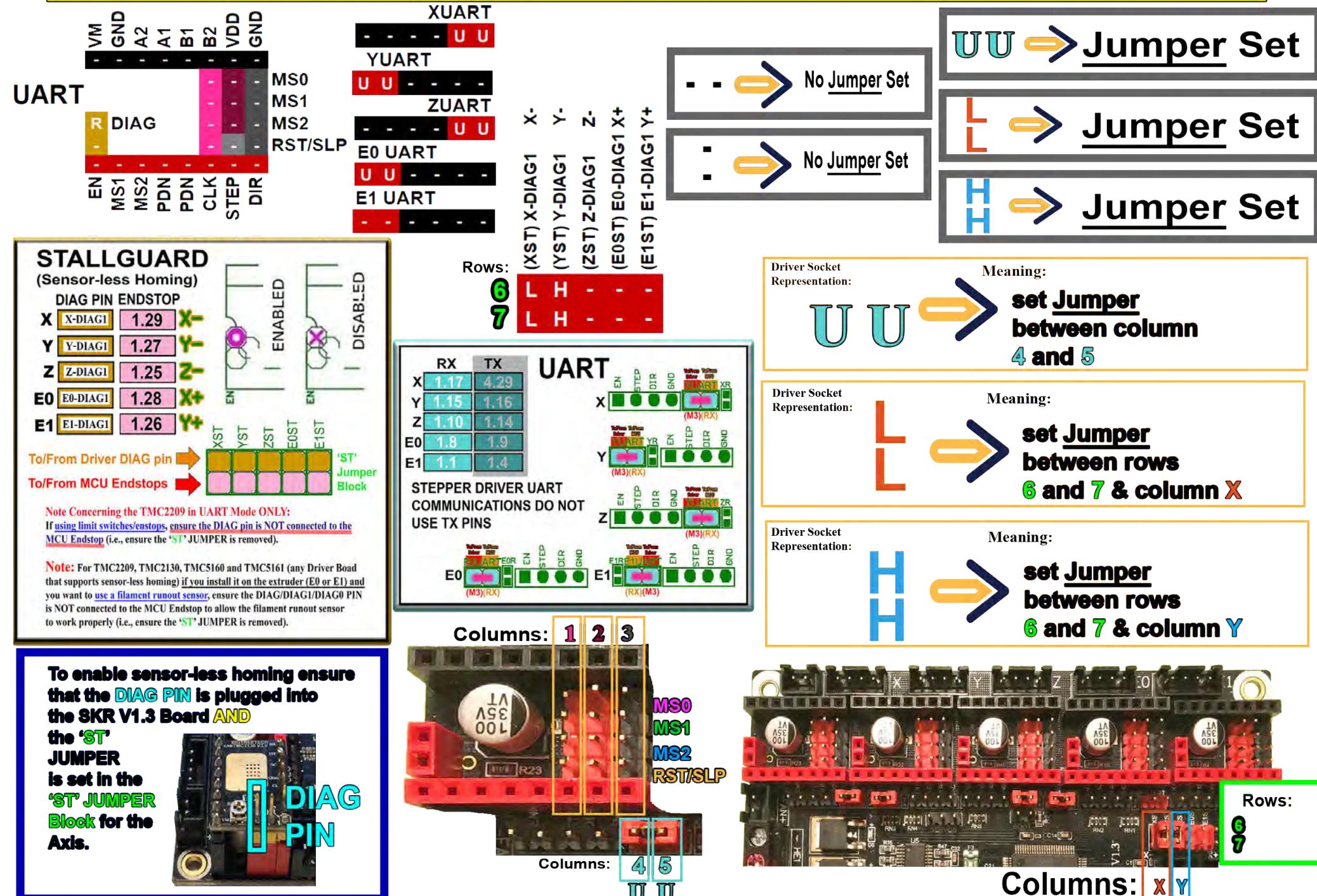
- - **No Jumper Set**

Driver Chip Chart:

- - **No Jumper Set**

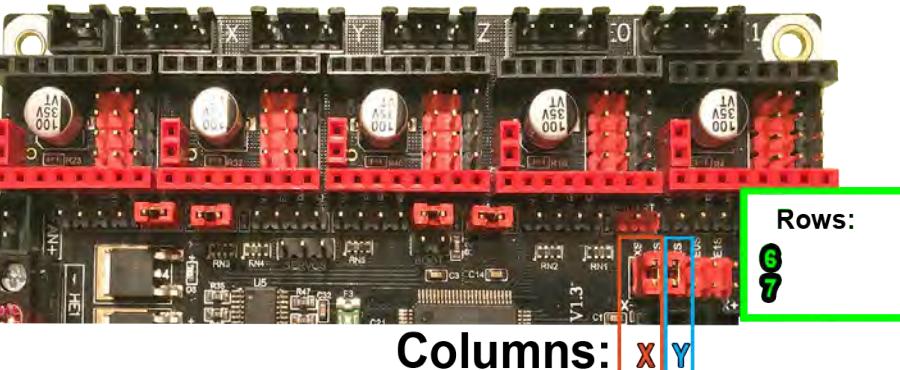
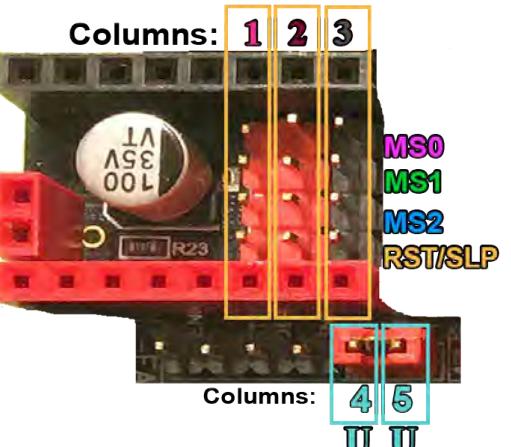
Driver Chip Chart:

| Driver Chip                        | MS0        | MS1         | Steps      | Interpolation | Mode               |
|------------------------------------|------------|-------------|------------|---------------|--------------------|
| Low                                | Low        | 1           |            | NONE          | spreadCycle        |
| High                               | Low        | 1/2         |            | NONE          | spreadCycle        |
| OPEN                               | Low        | 1/2         | 1/256      |               | spreadCycle        |
| <b>TMC2130</b><br>Stand Alone Mode | <b>Low</b> | <b>High</b> | <b>1/4</b> | <b>NONE</b>   | <b>spreadCycle</b> |
|                                    | OPEN       | High        | 1/4        | 1/256         | spreadCycle        |
|                                    | High       | OPEN        | 1/4        | 1/256         | stealthChop        |

APPENDIX D**Example 4 (TMC2209 UART with Sensor-less Homing) for SKR V1.3 Driver Socket Representation:**

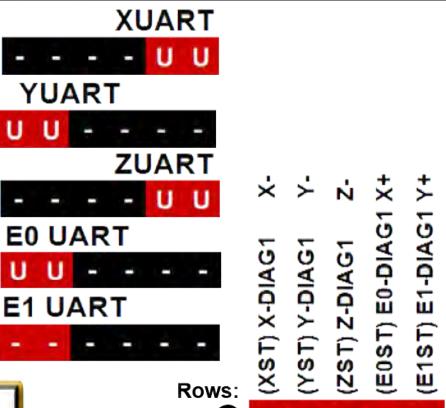
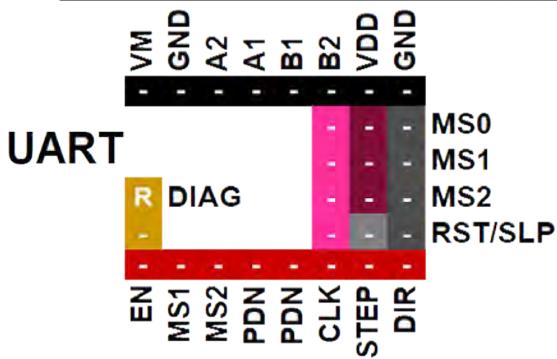
To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board AND** the **'ST'** JUMPER is set in the **'ST' JUMPER Block** for the Axis.

**DIAG PIN**



APPENDIX D

**Example 5 (TMC2209 UART WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation:**



**UU → Jumper Set**

**- → No Jumper Set**

**- - → No Jumper Set**

Driver Socket Representation:

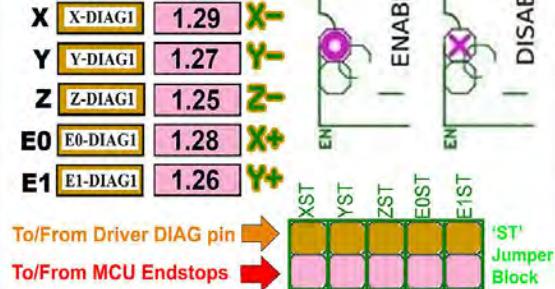
U U

Meaning:

**set Jumper between column 4 and 5**

### STALLGUARD (Sensor-less Homing)

DIAG PIN ENDSTOP



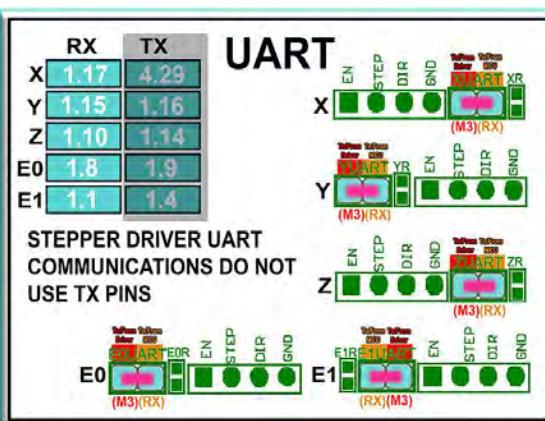
Note Concerning the TMC2209 in UART Mode ONLY:

If using limit switches/endstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

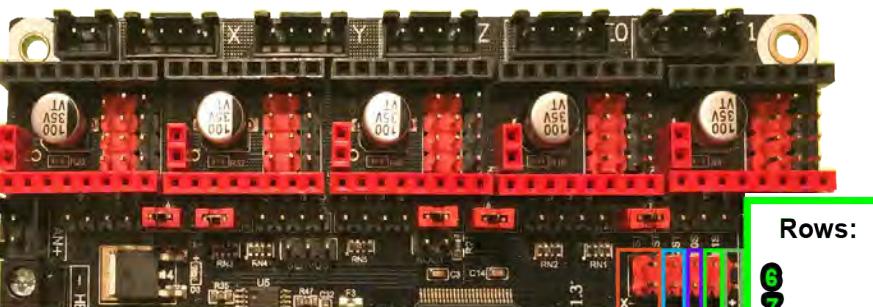
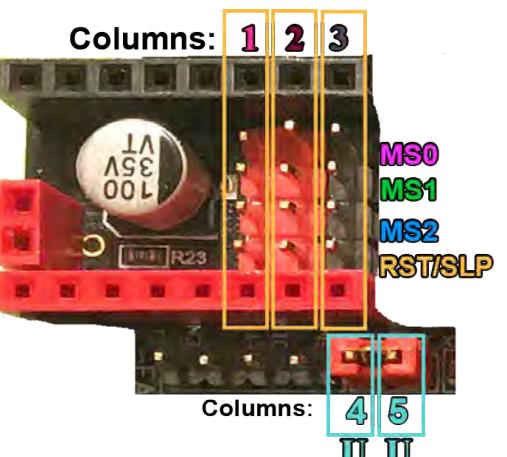
Note: For TMC2209, TMC2130, TMCS160 and TMCS161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

To disable sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the SKR V1.3 Board OR remove the '**ST**' JUMPER from the '**ST**' JUMPER Block.

Link to stackable header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>

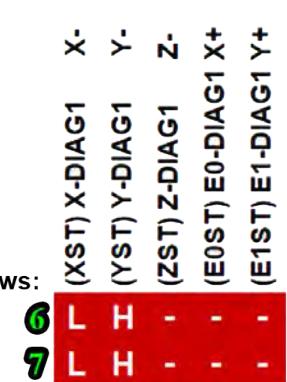
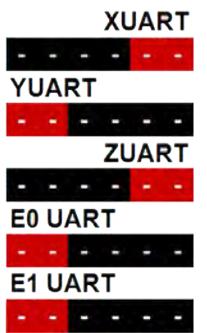
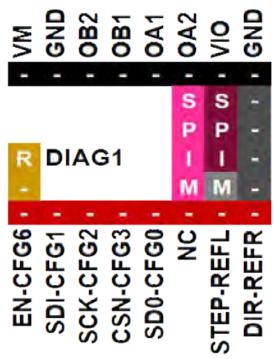


**The picture below shows ALL stepper driver sockets are setup for UART mode AND ALL 'ST' JUMPERS have been removed to DISABLE Sensor-less Homing due to the fact that the user wants to use physical endstops/limit switches on ALL Axes.**



Columns: X Y Z E0 E1

Rows: 6 7

APPENDIX D**Example 6 (TMC2130 SPI with Sensor-less Homing) for SKR V1.3 Driver Socket Representation:**

- - - → No Jumper Set

- - - → No Jumper Set

**Jumper Set**

**Jumper Set**

**Jumper Set**

**STALLGUARD**

(Sensor-less Homing)

DIAG PIN ENDSTOP

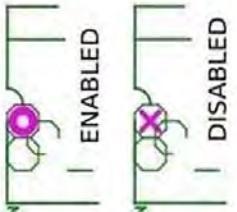
X X-DIAG1 1.29 X-

Y Y-DIAG1 1.27 Y-

Z Z-DIAG1 1.25 Z-

E0 E0-DIAG1 1.28 X+

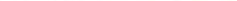
E1 E1-DIAG1 1.26 Y+



To/From Driver DIAG pin



To/From MCU Endstops

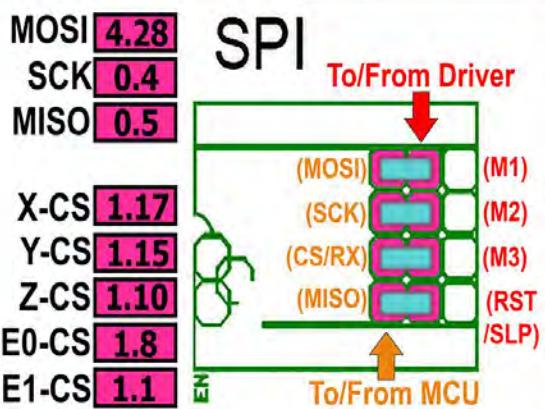
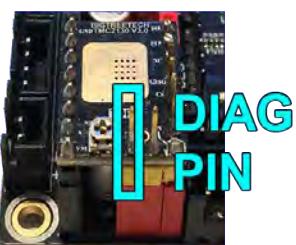


Note Concerning the TMC2209 in UART Mode ONLY:

If using limit switches/endstops, ensure the DIAG pin is NOT connected to the MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

To enable sensor-less homing ensure that the **DIAG PIN** is plugged into the **SKR V1.3 Board AND** the **'ST' JUMPER** is set in the **'ST' JUMPER Block** for the **Axes**.



Driver Socket Representation:

S S  
P P  
I I  
M M

Meaning:

**set Jumper between columns**  
**1 and 2 in rows: MS0, MS1, MS2, and RST/SLP**

Driver Socket Representation:

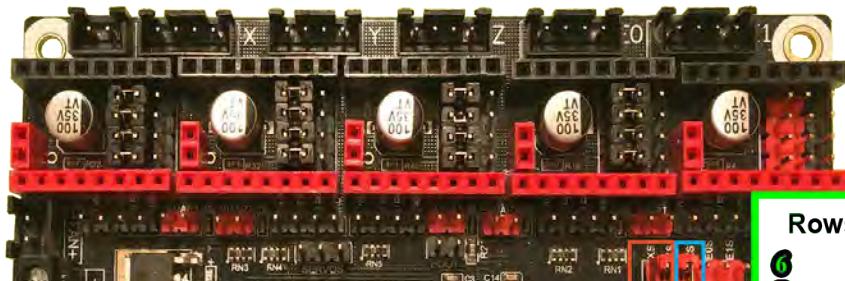
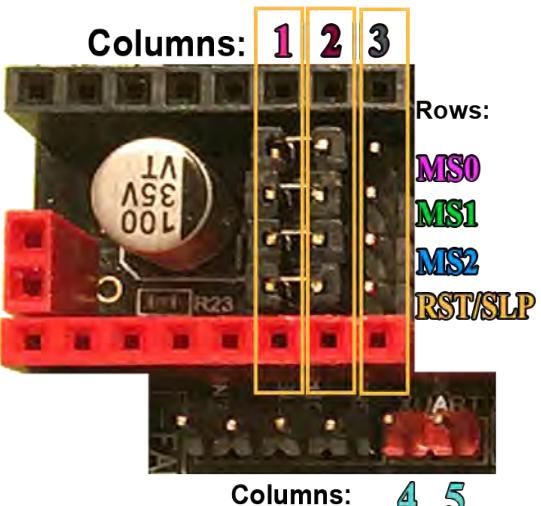
L L

Meaning:  
**set Jumper between rows**  
**6 and 7 in column X**

Driver Socket Representation:

H H

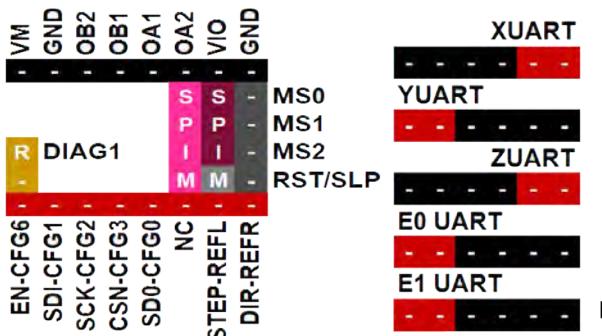
Meaning:  
**set Jumper between rows**  
**6 and 7 in column Y**



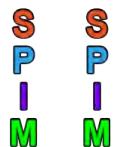
Columns: X Y  
Rows: 6 7

APPENDIX D

**Example 7 (TMC2130 SPI WITHOUT Sensor-less Homing) for SKR V1.3 Driver Socket Representation:**



Rows:  
6 (XST) X-DIAG1 X-  
7 (YST) Y-DIAG1 Y-  
6 (ZST) Z-DIAG1 Z-  
7 (E0 ST) E0-DIAG1 X+  
(E1 ST) E1-DIAG1 Y+



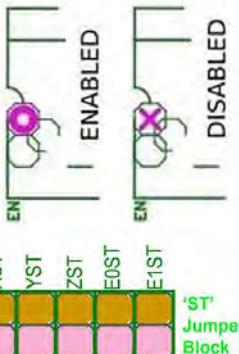
# Jumper Set

- - - → No Jumper Set

- - - → No Jumper Set

**STALLGUARD**

(Sensor-less Homing)

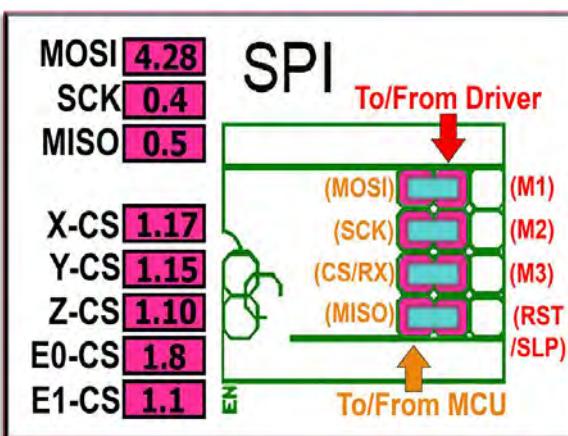


Note Concerning the TMC2209 in UART Mode ONLY:  
If using limit switches/endstops, ensure the DIAG pin is NOT connected to the  
MCU Endstop (i.e., ensure the 'ST' JUMPER is removed).

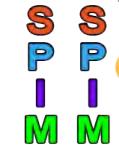
Note: For TMC2209, TMC2130, TMC5160 and TMC5161 (any Driver Board that supports sensor-less homing) if you install it on the extruder (E0 or E1) and you want to use a filament runout sensor, ensure the DIAG/DIAG1/DIAG0 PIN is NOT connected to the MCU Endstop to allow the filament runout sensor to work properly (i.e., ensure the 'ST' JUMPER is removed).

To disable sensor-less homing either ensure that the **DIAG PIN** is **NOT** plugged into the **SKR V1.3 Board** OR remove the '**ST**' JUMPER from the '**ST**' JUMPER Block.

Link to stackable header pins:  
<https://www.amazon.com/Glarks-Connector-Assortment-Stackable-Breakaway/dp/B07CWSXY7P>



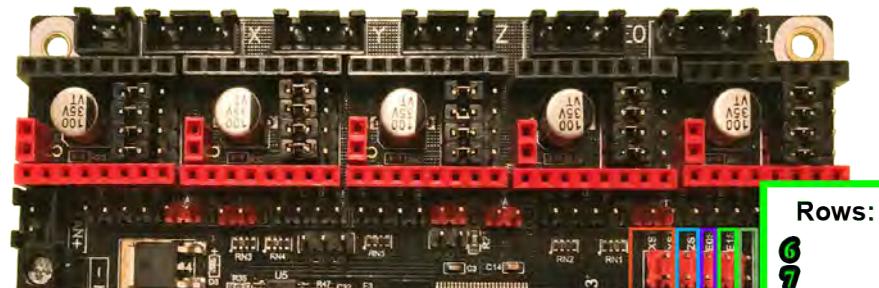
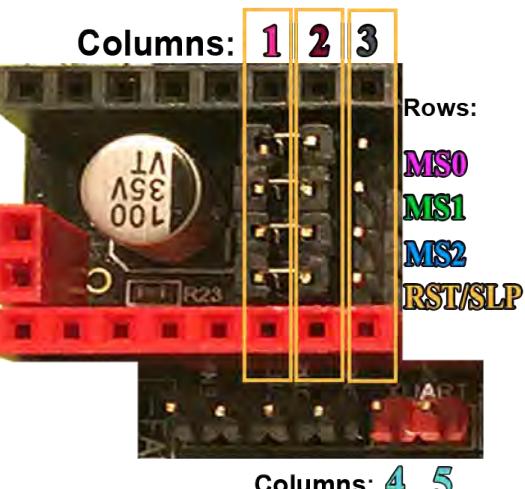
Driver Socket Representation:



Meaning:

set Jumpers between columns  
1 and 2 in rows: **MS0, MS1, MS2, and RST/SLP**

The picture below shows ALL stepper driver sockets are setup for SPI mode AND ALL 'ST' JUMPERS have been removed to DISABLE Sensor-less Homing due to the fact that the user wants to use physical endstops/limit switches on ALL Axes.

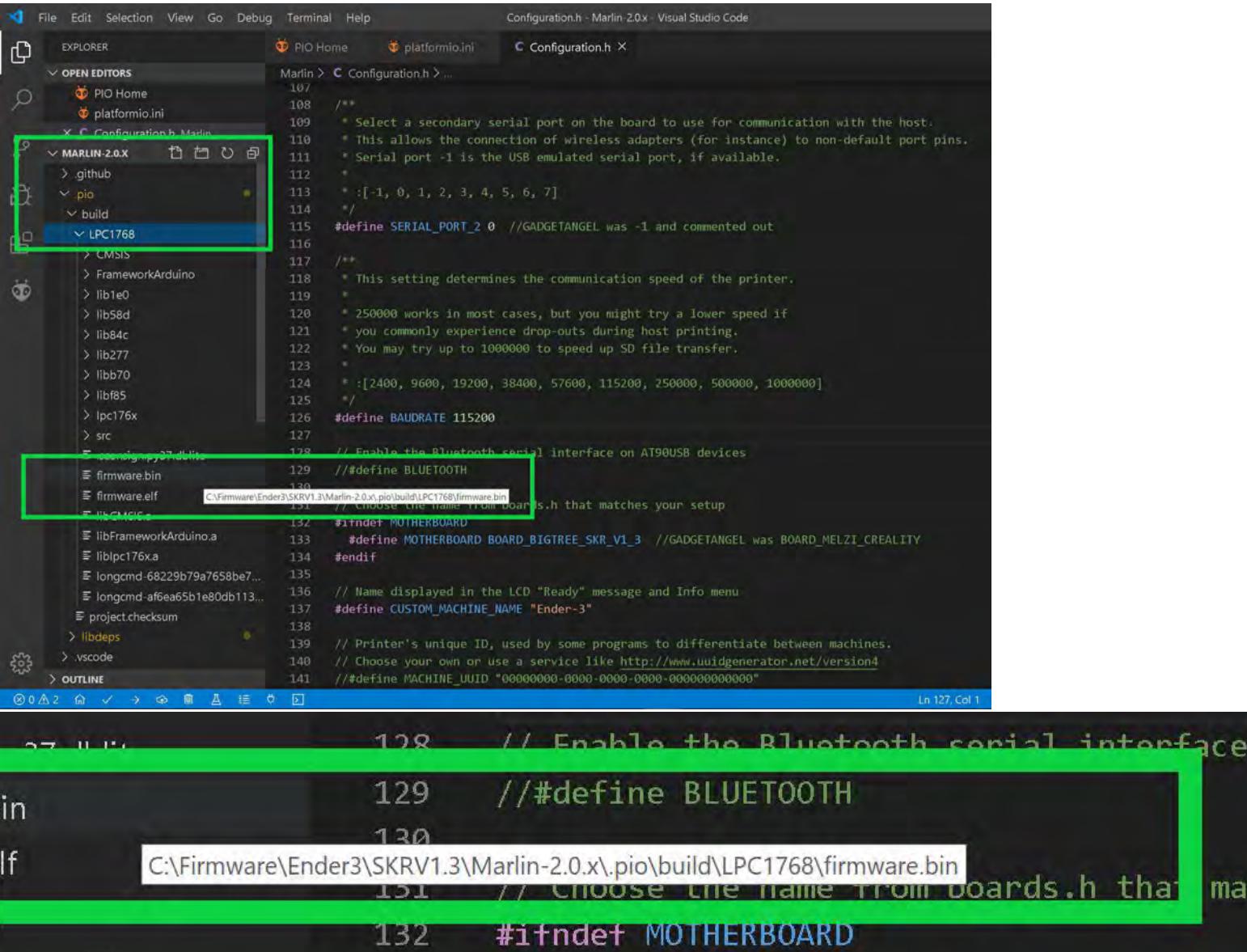


Columns: X Y Z 0 1  
Rows: 6 7

APPENDIX E

# Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

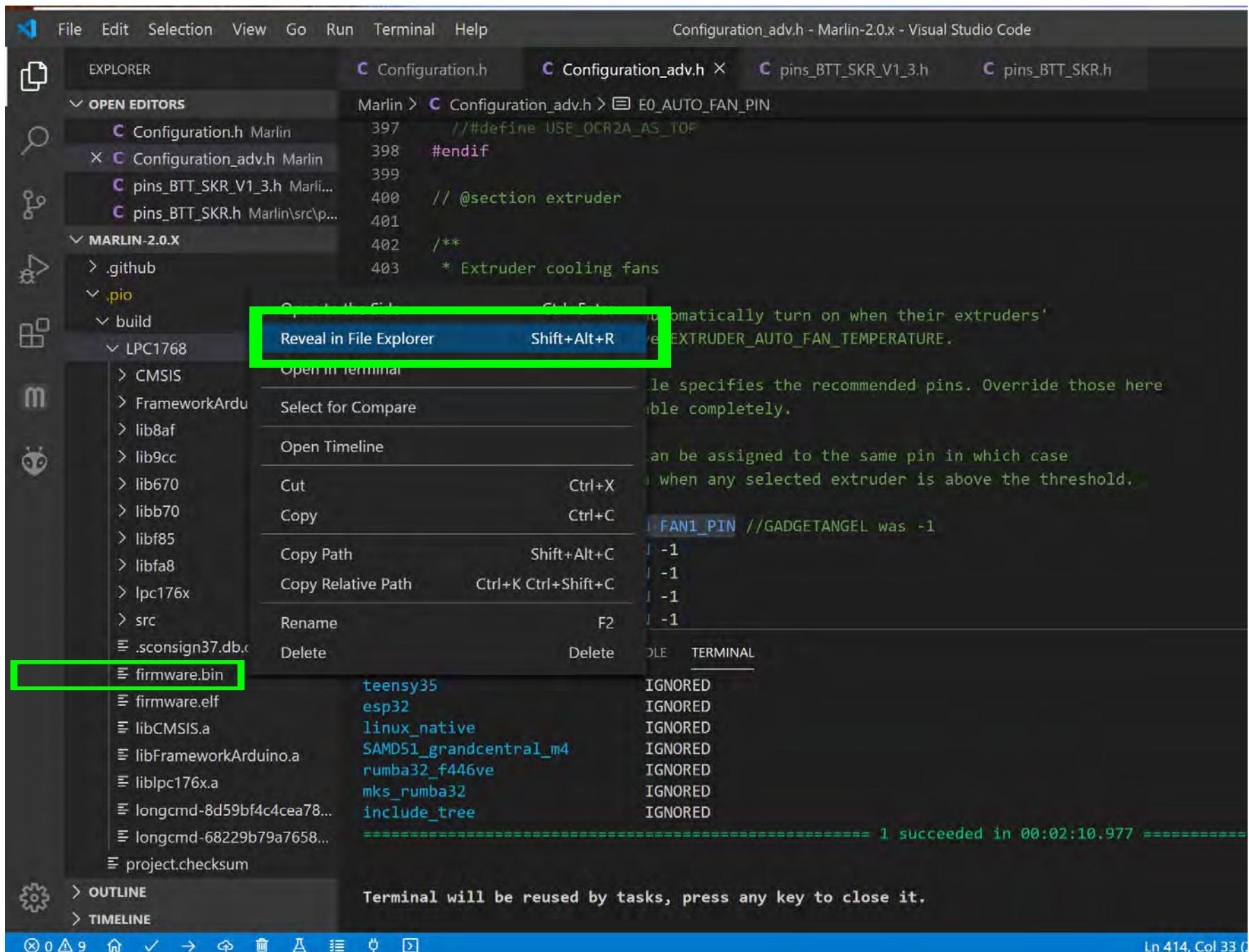
- For the SKR V1.3 board, you can save the "firmware.bin" file to the micro SD card. You can then place the micro SD card into the micro SD card reader of the SKR V1.3 board and turn on the power to your printer for the NEW firmware to be automatically applied to the SKR V1.3 board. The "firmware.bin" file can be found by using the left side of the VS code window and going to folder Marlin-2.0.X\pio\build\LPC1768\ and **right clicking** on the "firmware.bin" file name to bring up the content menu. Select "Reveal in File Explorer" to have your Windows machine open a file explorer window.



- Go to the next page.

APPENDIX E

# Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

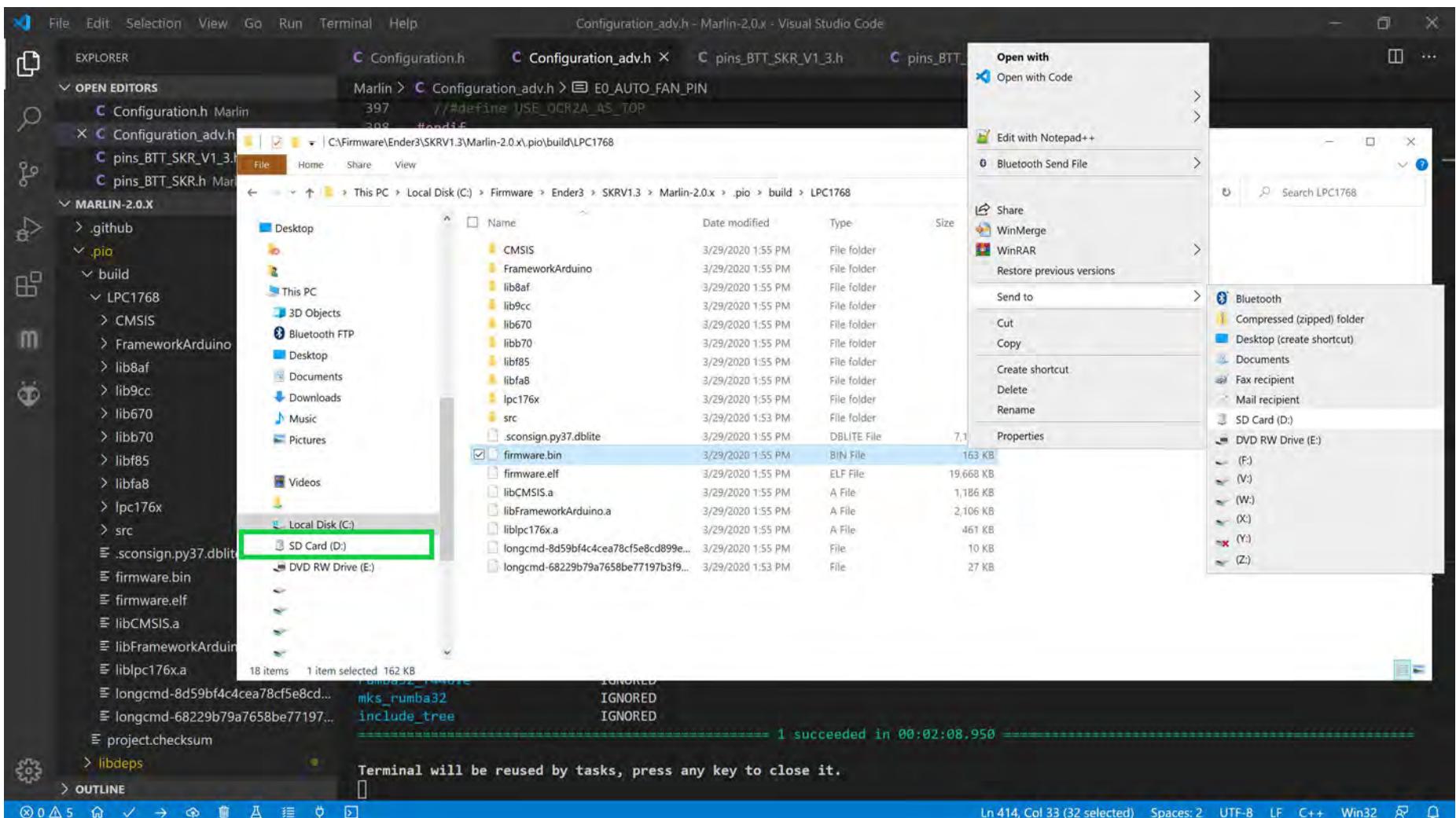


- Go to the next page.

APPENDIX E

# Location Of "firmware.bin" File from the Marlin Compilation for SKR V1.3 Board

- Ensure your micro SD card, which came with your SKR V1.3 board, is in a micro-to-SD-card adapter and plugged into your Window's SD card reader.
  - From the file explorer window **right click** on the "firmware.bin" file name and select "Send to" from the content menu. For my computer "D:" is my SD card reader. For your computer it may be a different drive letter. The "firmware.bin" file will be written to the SD card and renamed to "FIRMWARE.CUR".



- **Right click** on the SD card reader, as seen in the **GREEN** box above, and select "Eject" from the context menu. Take the micro SD card out of the adapter and place it into the micro SD card reader of the SKR V1.3 board. Turn on the power to the printer (i.e. the SKR V1.3 board) and the "FIRMWARE.CUR" file will be uploaded to the board. After the upload, the firmware file will be renamed to "firmware.bin" and stored on the micro SD card.

## APPENDIX F

### Links to Reference Material

#### Marlin Firmware Documentation

- <https://marlinfw.org/docs/configuration/configuration.html>
- <https://marlinfw.org/docs/basics/introduction.html>
- <https://marlinfw.org/meta/download/>
- <https://marlinfw.org/docs/basics/install.html>
- <https://marlinfw.org/tools/u8glib/converter.html>
- [https://marlinfw.org/tools/lin\\_advance/k-factor.html](https://marlinfw.org/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.

**DESIGNSPARK**

## Links to Reference Material

### POLOLU A4988 and BIQU A4988



- [YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware](#)
- [A4988 Stepper Motor Driver Carrier](#)



### DRV8825



- [YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware at 667s into the Video](#)
- [DRV8825 Stepper Motor Driver Carrier, High Current](#)



- Go to the next page.

## Links to Reference Material

### BIQU LV8729, FYSETC LV8729, LERDGE LV8729, and MKS LV8729



- [YouTube Channel Teching Tech: LV8729 guide + comparison with TMC drivers](#)

### BIQU LV8729

- [BIQU LV8729 Stepper Motor Driver Support 6V-36V With Heatsinks](#)



- Go to the next page.

## Links to Reference Material

### FYSETC LV8729

- [FYSETC LV8729 V1.0 Product Information](#)



### LERDGE LV8729

- [LERDGE LV8729 Stepper Motor Driver Informatin from AliExpress](#)



- Go to the next page.

## Links to Reference Material

### MKS LV8729

- [MKS LV8729 - ultra quiet stepper motor driver 1.5A Product Information](#)



### FYSETC S6128 V1.1

- [FYSETC S6128 V1.1 Product Information](#)



- Go to the next page.

## Links to Reference Material

### FYSETC ST820

- [FYSETC ST820 V1.0 Product Information](#)



### BIQU ST820

- [BIGTREETECH ST820 V1.0 Step Stepper Motor Driver Product Information from Alibaba.com](#)



- [YouTube Channel Pro3d: First Looks - ST820 NEW Stepper Drivers](#)



- <https://github.com/bigtreeTech/BIGTREETECH-ST820-V1.0>



- Go to the next page.

## Links to Reference Material

### POLOLU ST820 (STSPIN820)

- [STSPIN820 Stepper Motor Driver Carrier Product Information](#)



### POLOLU MP6500



- [YouTube Channel PW: Input signal current to drive an MP6500 Stepper driver board](#)
- [Data Sheet for MP6500](#)
- [MP6500 Stepper Motor Driver Carrier, \(Potentiometer Current Control\) Product Information](#)



- Go to the next page.

## APPENDIX F

### Links to Reference Material

#### POLOLU TB67S249FTG

- [Data sheet for TB67S249FTG Stepper Motor Driver](#) **TOSHIBA**
- [Another Data Sheet on TB67S249FTG Stepper Motor Driver from POLOLU website](#)
- [POLOLU TB67S249FTG Stepper Motor Driver Compact Carrier Product Information](#)



#### BIQU S109 and FYSETC S109

- [Data sheet TB67S109AFTG for S109 Stepper Motor Drivers](#) **TOSHIBA**

- Go to the next page.

## Links to Reference Material

### **BIQU S109**

- [BIGTREETECH S109 V1.0 Stepper Motor Driver Product Information](#)



### **FYSETC S109**

- [FYSETC S109 V1.1 Product Information](#)



- Go to the next page.

## Links to Reference Material

### Marlin Firmware Documentation Specific to TMC Drivers

- [https://marlinfw.org/docs/hardware/tmc\\_drivers.html](https://marlinfw.org/docs/hardware/tmc_drivers.html)



### Information Common to All TMC Drivers



- <https://github.com/watterott/SilentStepStick>
- Trinamic: The SilentStepStick is a stepper driver board for 2-phase motors, based on the TMC2100, TMC2130, TMC2208, TMC2209 or TMC5160
- Instructable: [Install and Configure SilentStepStick in RAMPS](#)



- Go to the next page.

## Links to Reference Material

### BIQU TMC2100 and MKS TMC2100

- [TMC2100 SILENTSTEPSTICK PIN out](#)
- [TMC2100 WIKI site](#)
- [YouTube Channel Teching Tech: TMC2100 guide - Stepper driver upgrades part 1 / How to set VREF & firmware starting at 869seconds](#)
- [TMC2100 DATASHEET](#)
- [Trinamic's Website: TMC2100 DATASHEET](#)
- [BIGTREETECH TMC2100 Stepper Motor Driver Product Information](#)



- Go to the next page.

## Links to Reference Material

### BIQU TMC2130

- [TMC 2130 Stepper Drivers - Standalone Mode - Chris's Basement](#)
- [TMC2130 Guide - stepper motors driver upgrades part 3 - Sensorless homing, starting at 34s](#)
- [TMC2130 SILENTSTEPSTICK PINout](#)
- <https://github.com/bigtreetech/BIGTREETECH-TMC2130-V3.0/blob/master/TMC2130-V3.0RM.pdf>
- [Trinamic's TMC2130 DATASHEET](#)
- [Trinamic 2130 Stepper Driver Setup - Marlin - Chris's Basement](#)
- [TMC2130 Guide - stepper motors driver upgrades part 3 - Sensorless homing, starting at 55s](#)



### Information Common to Biqu TMC2208 V3.0 and FYSETC TMC2208 V1.2

- [TMC2208 UART on BigTreeTech/Biqu SKR V1.1, V1.3, and V1.4 Controllers](#)
- [Trinamic's TMC2202, TMC22082 & TMC2224 family Datasheet](#)
- [TMC2208 SILENTSTEPSTICK PINout](#)
- [YouTube Channel Da Hai Zhu: TMC2208 3 Ways - Part 1](#)
- [TMC2208 guide - Stepper driver upgrades part 2](#)
- [TMC2208 guide - Stepper driver upgrades part 2, starting at 419s](#)
- Go to the next page.



## Links to Reference Material

### Information Common to BIQU TMC2208 V3.0 and FYSETC TMC2208 V1.2 (Continued)



- [YouTube Channel Da Hai Zhu: TMC2208 3 ways - Part 2](#)
- [YouTube Channel Keith Young: TMC2208](#)
- [YouTube Channel Da Hai Zhu: TMC2208 3 Ways - Part 3](#)
- [YouTube Channel Teching Tech: TMC2208 guide - Stepper driver upgrades part 2, starting at 510s](#)

### BIQU TMC2208 V3.0



- <https://github.com/bigtreetech/BIGTREETECH-TMC2208-V3.0/blob/master/TMC2208-V3.0%20manual.pdf>



### FYSETC TMC2208 V1.2



- [FYSETC TMC2208 Product Information](#)

- Go to the next page.

## Links to Reference Material

### Information Common to TMC2208 and BIQU TMC2225



- [Instructable: TMC2208 UART on BigTreeTech/BIQU SKR V1.1, V1.3, and V1.4 Controllers](#)

### BIQU TMC2225 V1.0



<https://github.com/bigtreeTech/BIGTREETECH-TMC2225-V1.0>



<https://github.com/bigtreeTech/BIGTREETECH-TMC2225-V1.0/blob/master/TMC2225%20V1.0%20manual.pdf>



[TMC2225 Datasheet](#)



- Go to the next page.

## APPENDIX F

### Links to Reference Material

#### **BIQU TMC2209 V1.2**

- [!\[\]\(67ce93ebab16ac9ce0763ff8c743db28\_img.jpg\) YouTube Channel Teching Tech: TMC2209 and TMC 5160: Guide for MKS Gen L and SKR V1.3](#)
- [!\[\]\(30b50109e26bc6ffa95d35a5fa3fdc67\_img.jpg\) TMC2209 Stepper Drivers - Bigtreetech - SKR 1.3 - Install - Chris's Basement](#)
- [!\[\]\(4da119d5aaf8e829da050bedcb2783c3\_img.jpg\) TMC2209 SILENTSTEPSTICK PINout](#)
- [!\[\]\(0749603f316962906d87af5cb9e5c96b\_img.jpg\) https://github.com/bigtreeTech/BIGTREETECH-TMC2209-V1.2/blob/master/manual/TMC2209-V1.2-manual.pdf](https://github.com/bigtreeTech/BIGTREETECH-TMC2209-V1.2/blob/master/manual/TMC2209-V1.2-manual.pdf)
- [!\[\]\(83747f0c8ff63abbf888cacb2ec6e78b\_img.jpg\) Trinamic's TMC2209 Datasheet](#)
- [!\[\]\(07833bc5b929df948f27c6c7741d671d\_img.jpg\) BIQU®  
BIGTREETECH®](#)

#### **BIQU TMC5160 V1.2**

- [!\[\]\(62d39fefbe9ab624a0279fe1b1e991ad\_img.jpg\) YouTube Channel Teching Tech: TMC2209 and TMC 5160: Guide for MKS Gen L and SKR V1.3](#)
- [!\[\]\(e0d7b112aeb0b652672e8da99ca979ff\_img.jpg\) TMC5160 SILENTSTEPSTICK PINout](#)
- [!\[\]\(ff75a9e3a1ed044826d5e5520767585a\_img.jpg\) Trinamic's TMC5160 / TMC5160A DATASHEET](#)
- [!\[\]\(e525bb2f5e5e451fd51ec30c68059dc5\_img.jpg\) https://github.com/bigtreeTech/BIGTREETECH-TMC5160-V1.0/blob/master/manual/BIGTREETECH%20TMC5160-V1.0%20manual.pdf](https://github.com/bigtreeTech/BIGTREETECH-TMC5160-V1.0/blob/master/manual/BIGTREETECH%20TMC5160-V1.0%20manual.pdf)
- [!\[\]\(793027b291670e510ae5b1c0189d64dd\_img.jpg\) BIQU®  
BIGTREETECH®](#)
- [!\[\]\(0abf269dffc937acc0d7eedbe025d2ed\_img.jpg\) BIGTREETECH TMC5160 V1.2 Stepper Motor Driver SPI Mode, Product Information](#)
- Go to the next page.

## APPENDIX F

### Links to Reference Material

#### **BIQU TMC5161 V1.0**

- <https://github.com/bigtreeTech/BIGTREETECH-TMC5161-V1.0/blob/master/TMC5161%20v1.0%20manual.pdf> 

- [Trinamic's TMC5161A DATASHEET](#)



#### **SKR V1.3 Board**

- <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/BigtreeTech%20SKR%20V1.3%20Case%20model>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/Wiring%20diagram>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/firmware>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/tree/master/BTT%20SKR%20V1.3/hardware>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/issues>
  - <https://github.com/bigtreeTech/BIGTREETECH-SKR-V1.3/issues?q=is%3Aissue+is%3Aclosed>
- Go to the next page.



## Links to Reference Material

### Facebook Groups

- [BIGTREETECH SKR PRO User Group](#)



- [BIGTREETECH](#)



- [Bigtreetech / Biqu SKR 32 Bit 3D printer board owners group](#)



- [MarlinFirmware](#)



- [Marlin Firmware for 3D Printers User Group](#)



- [FYSETC](#)



- [The Institute of 3D Printing](#)



- Go to the next page.



## Links to Reference Material

### Miscellaneous Information

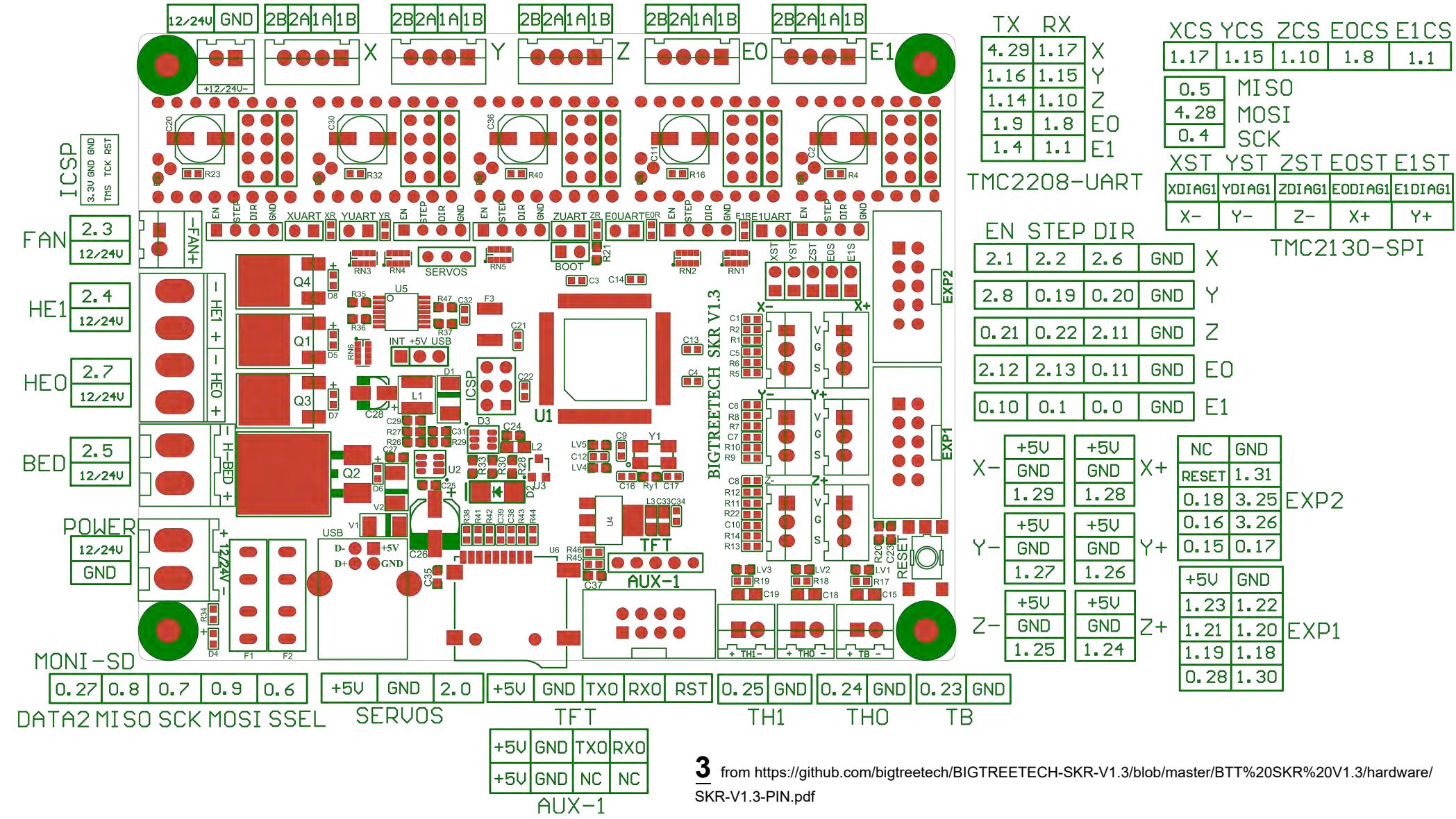
- [YouTube Channel Stelios Liver: Vref formula for common step motor drivers](#)
- [Stepper wiring WIKI](#) 
- [Watterott CONFIGURATOR FOR TMC220X](#) 
- [FYSETC's TMC2208/#to-run-the-program](#) 
- <https://marlinfw.org/meta/gcode/> 
- <https://blog.prusaprinters.org/calculator/> 
- <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers>
- <https://github.com/superjamie/lazyweb/wiki/3D-Printing-Stepper-Motors-and-Drivers#how-to-tune-stepper-drivers>
- [https://marlinfw.org/docs/basics/install\\_platformio.html](https://marlinfw.org/docs/basics/install_platformio.html) 
- <https://marlinfw.org/meta/download/> 
- <https://www.pronterface.com/> 
- [Sensorless Homing](#) 
- [Watterott's Silentstepstick Comparison](#) 
- <https://reprap.org/wiki/Category:Reference> 
- <https://reprap.org/wiki/Calibration> 
- [How does one use the PT100 Amp Board with a 3.3v controller?](#) 
- [Stepper Motor Wiring Conventions](#) 
- [BLtouch probe installation on SKR 32-bit board](#) 

## Original PIN Diagram:

**BIGTREETECH Reference Material**

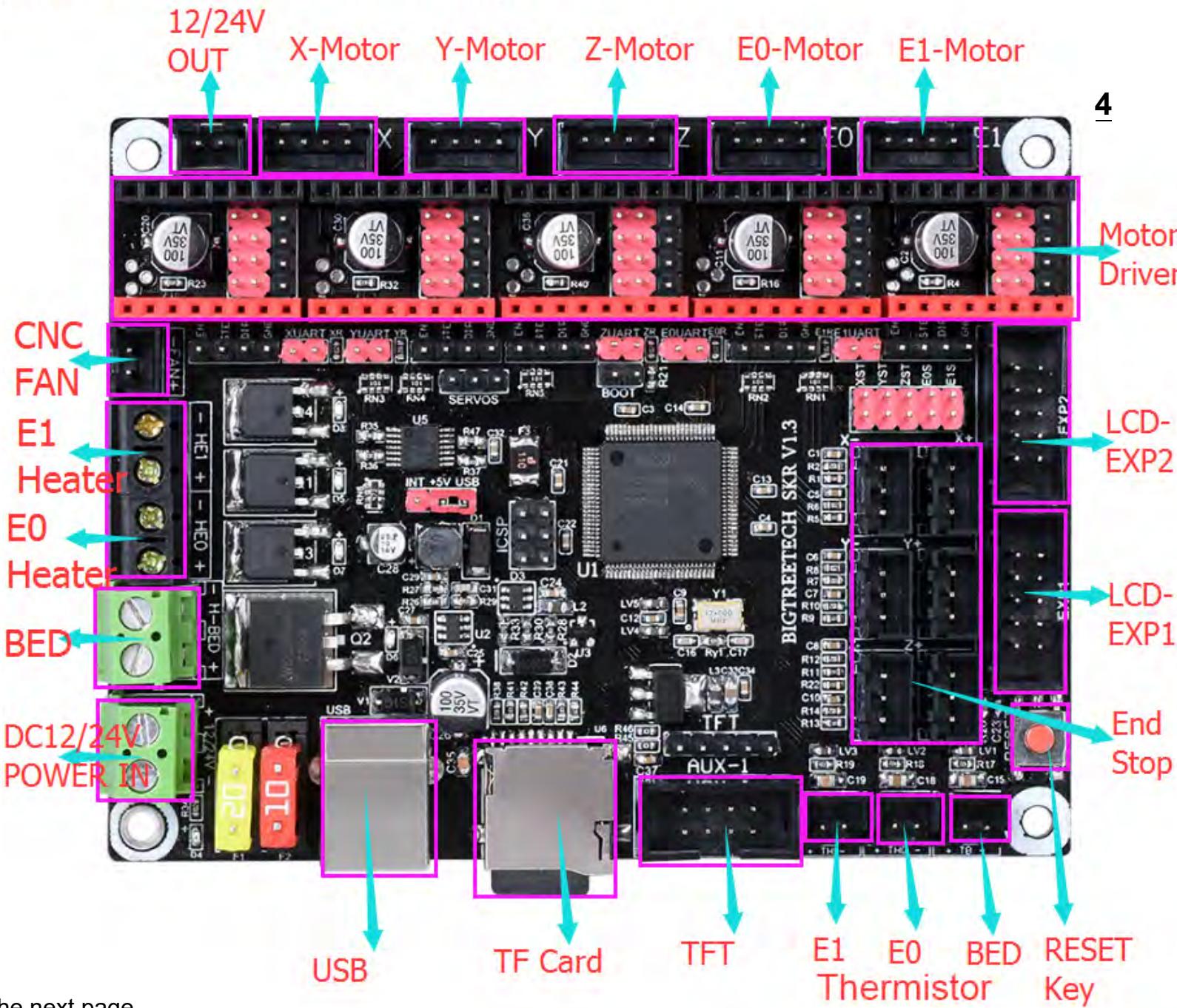
# **BIGTREETECH SKR V1.3-PIN**

**WWW.BIGTREE-TECH.COM**



- Go to the next page.

## APPENDIX G

Original Wiring Diagram 1:BIGTREETECH Reference Material**BIGTREETECH-SKR-V1.3-Connect**

- Go to the next page.

4

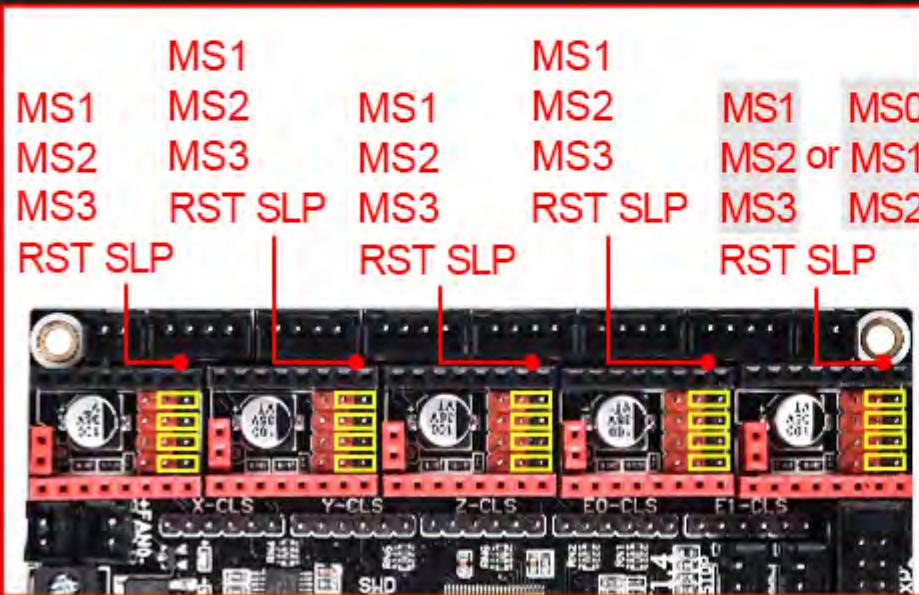
from <https://github.com/bigtreech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/Wiring%20diagram/SKR%20V1.3%20Wiring%20diagram.jpg>

Original Wiring Diagram 2  
for STEP/DIR Mode:

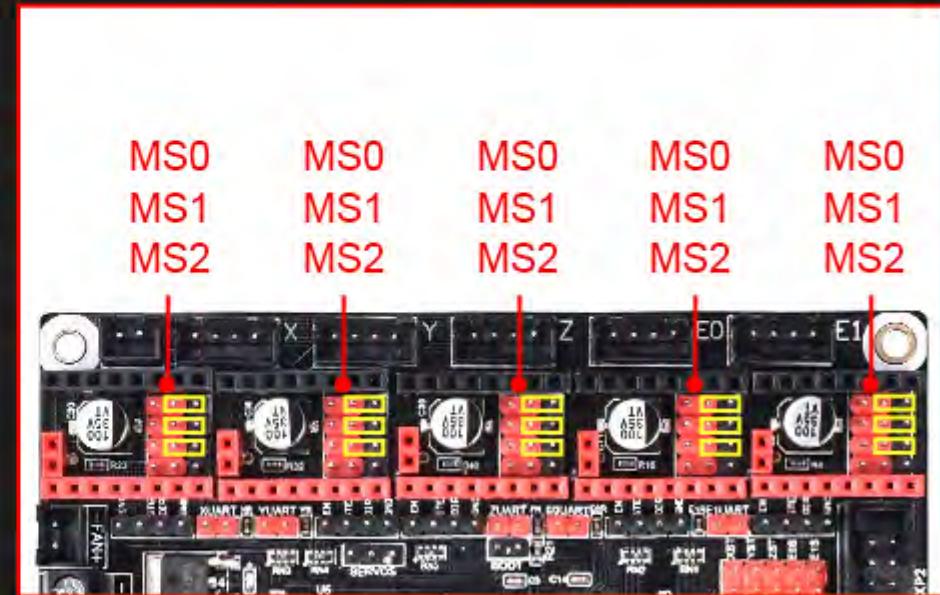
BIGTREETECH Reference Material

# The difference of driving. mode wiring between SKR V1.3 and SKR V1.4

## Driver-SIEP/DIY MODE



SKR V1.4



SKR V1.3

5 from <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/STEP%20DIR.jpg>

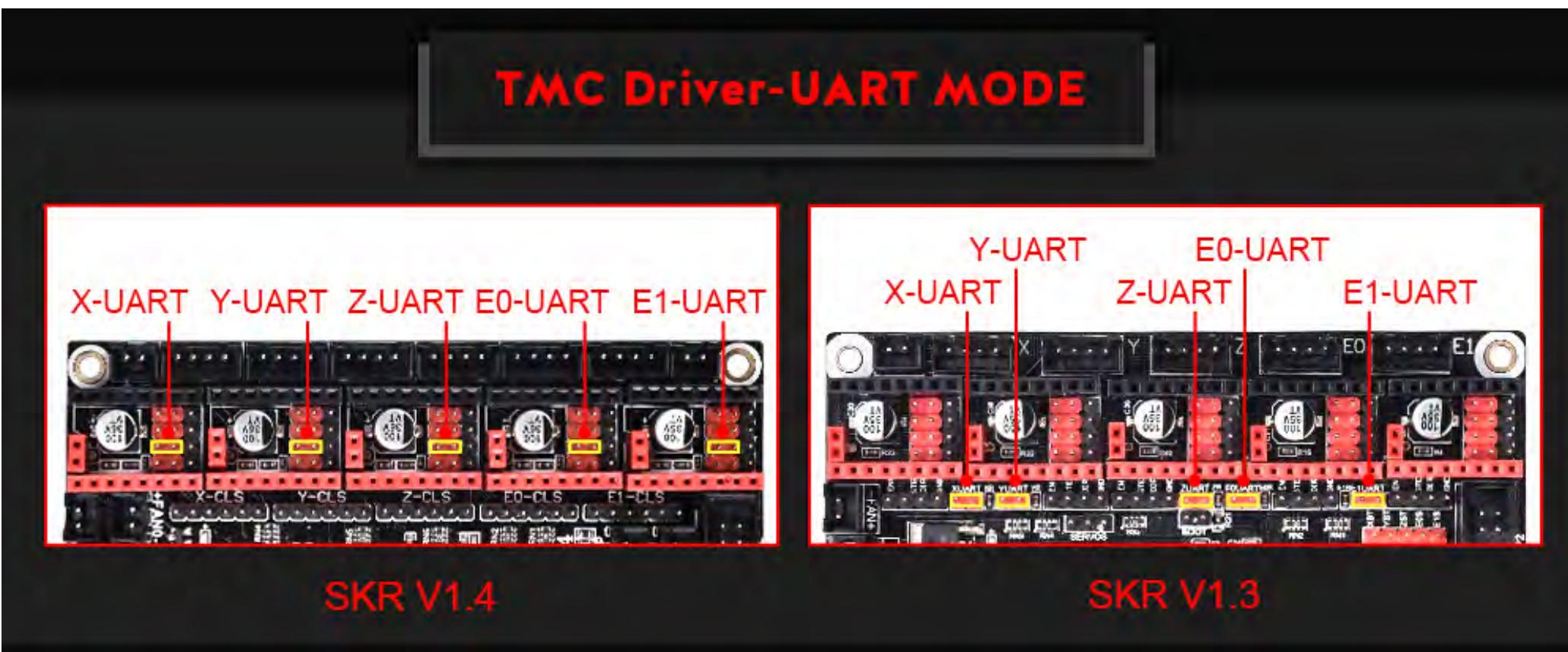
- Go to the next page.

## APPENDIX G

### BIGTREETECH Reference Material

#### Original Wiring Diagram 3 for UART Mode:

6



6 from <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/UART.jpg>

- Go to the next page.

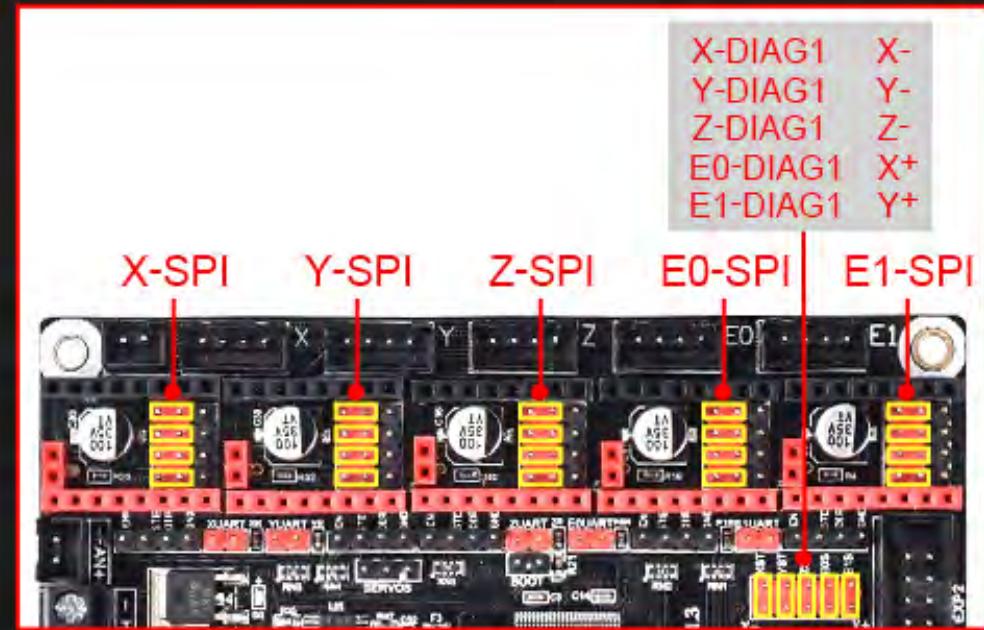
## APPENDIX G

**BIGTREETECH Reference Material****Original Wiring Diagram 4  
for SPI Mode:**

7

**SPI mode driven by TMC**

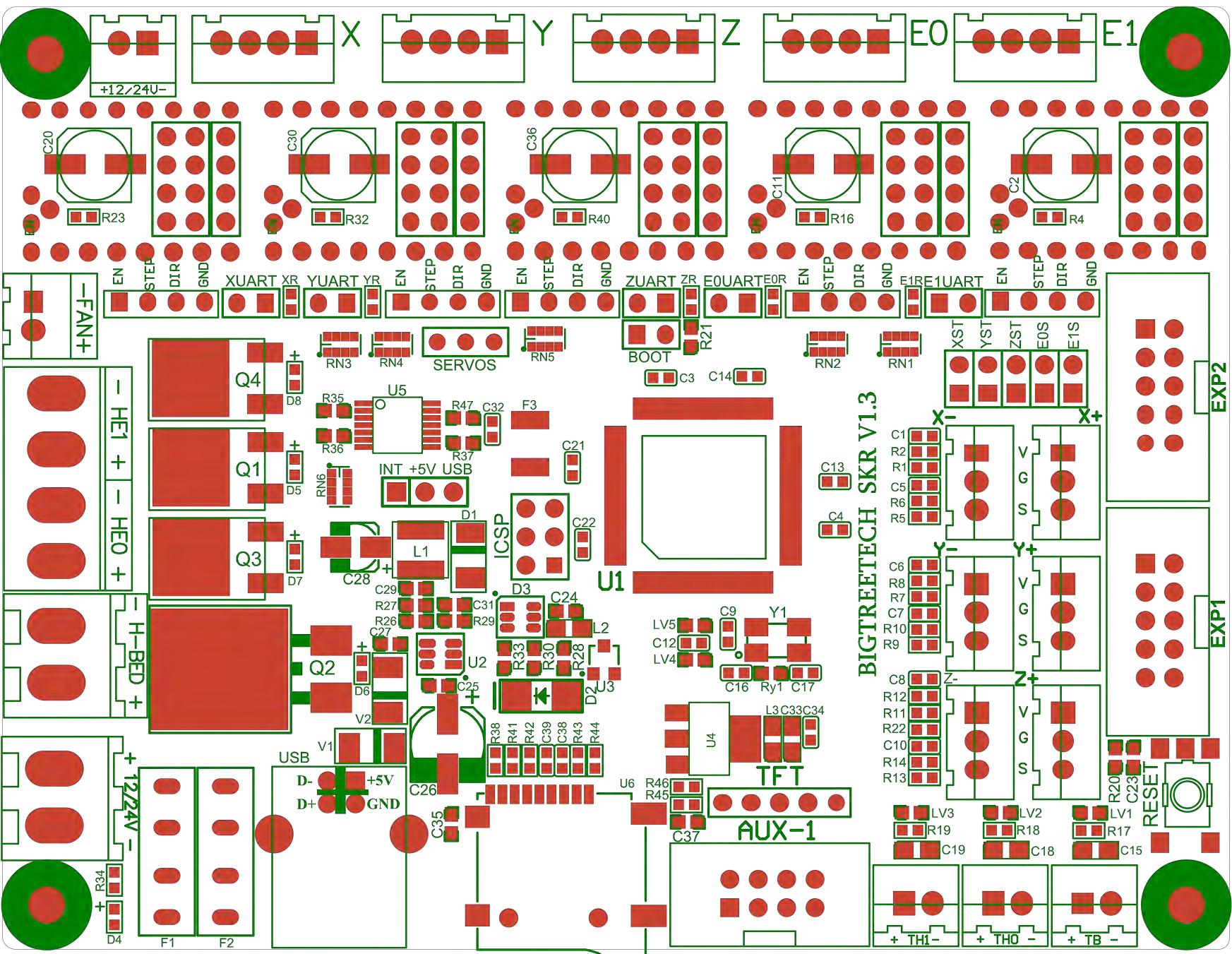
SKR V1.4



SKR V1.3

7 from <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.4/Hardware/SPI.jpg>

- Go to the next page.

**APPENDIX G****BIGTREETECH Reference Material**

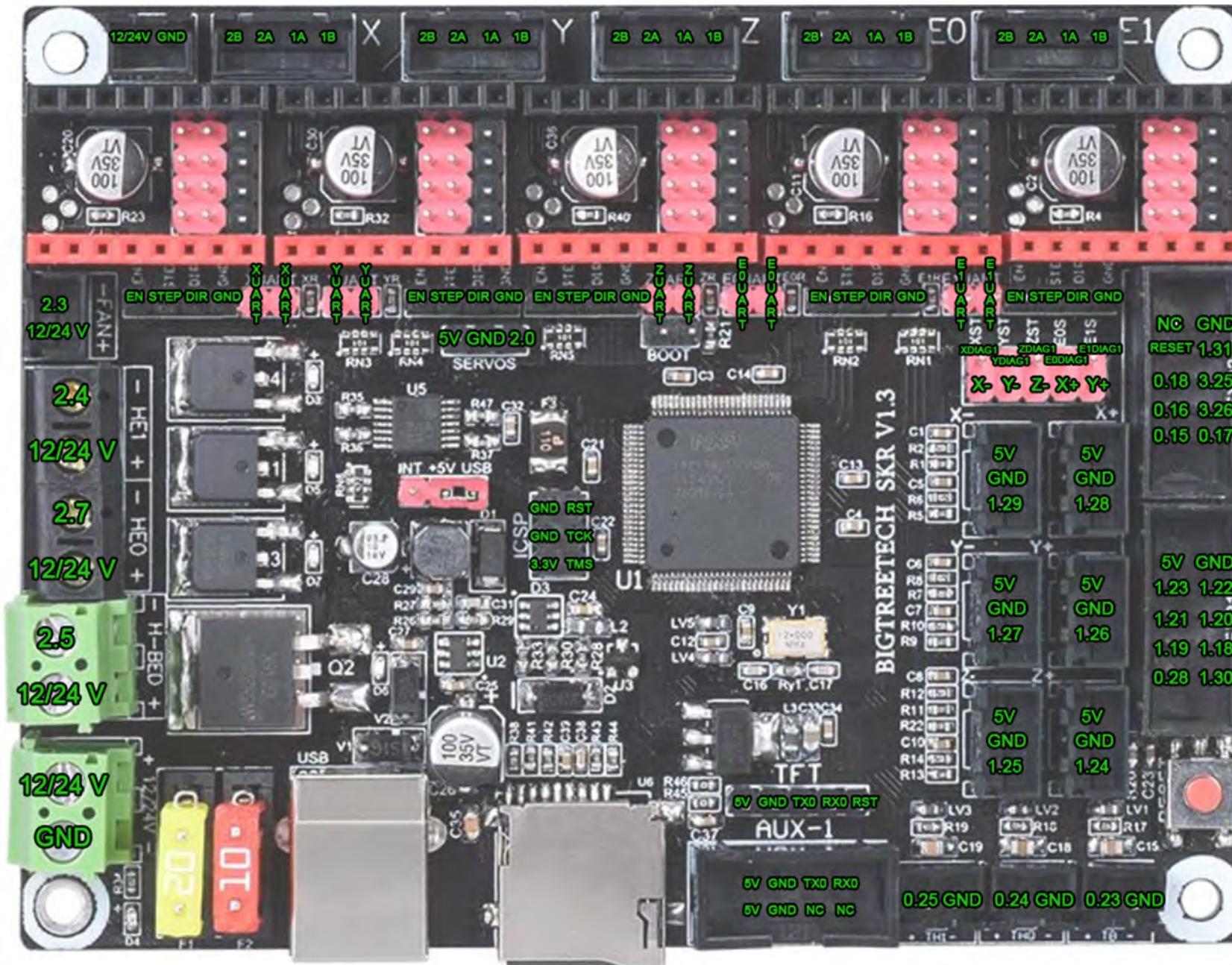
- Go to the next page.

## **APPENDIX G**

**BIGTREETECH Reference Material**

## **Additional Original Reference Material for PIN 1 Diagram:**

9

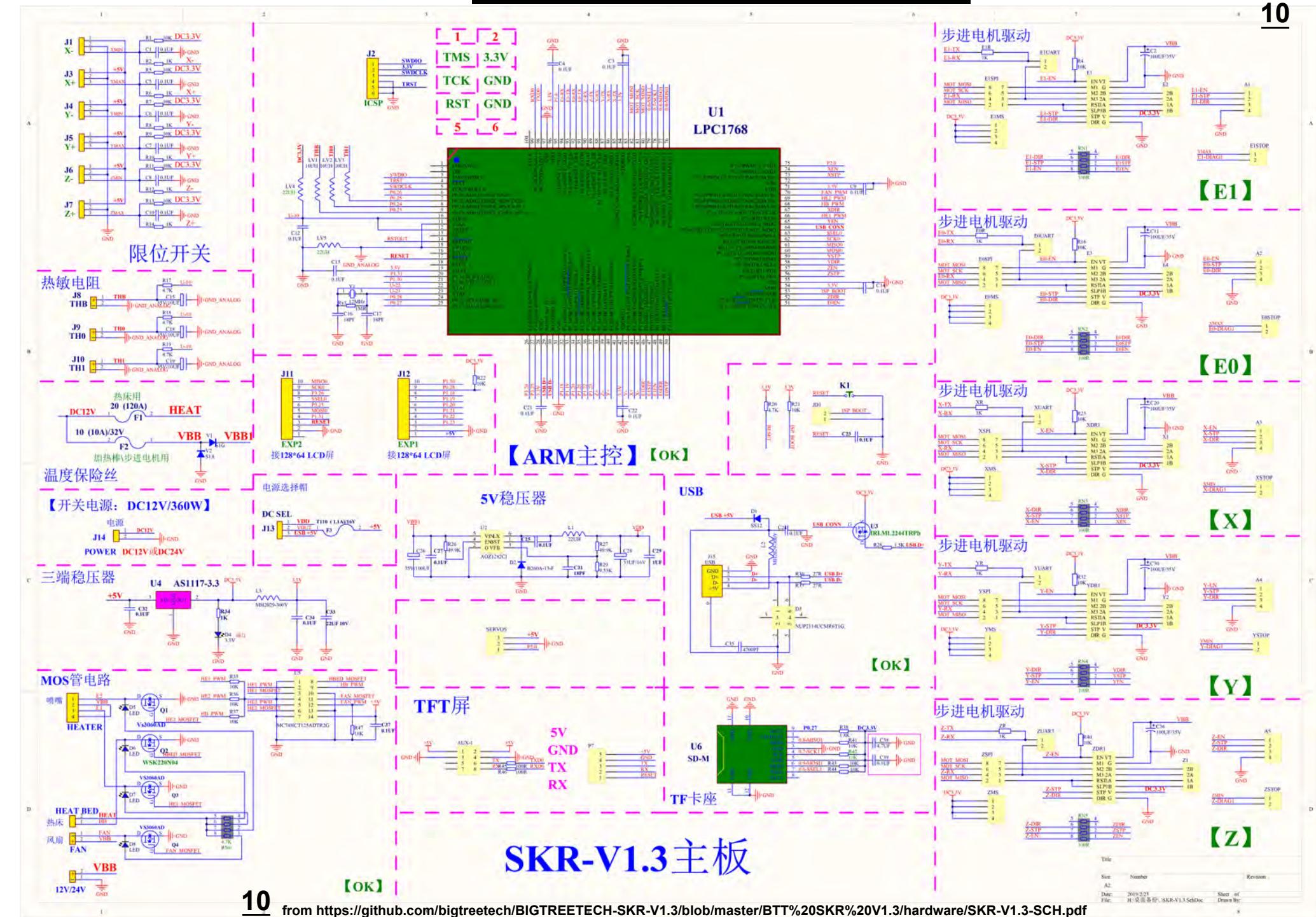


**9** from <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKRpinoutpic.jpg>

## APPENDIX G

Original Schematic Diagram:BIGTREETECH Reference Material

10



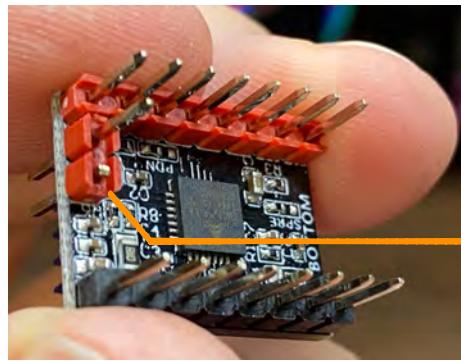
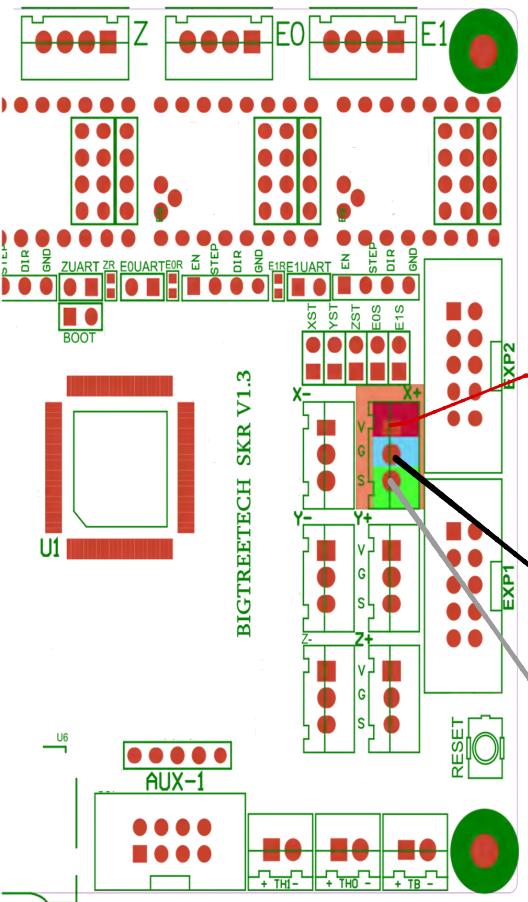
10

from <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/BTT%20SKR%20V1.3/hardware/SKR-V1.3-SCH.pdf>

APPENDIX H

# Filament Runout Sensor Wired to Limit Switch

## {X-, Y-, Z-, X+, Y+, or Z+}



The filament runout sensor will not work with the diagnostic PIN (DIAG/DIAG0/DIAG1) of the extruded stepper (or other Limit switches/Endstops) so you MUST cut it off or use a soldering iron to heat up the PIN and slide it up out of the way.

| EN STEP DIR |      |      |     |
|-------------|------|------|-----|
| 2.1         | 2.2  | 2.6  | GND |
| 2.8         | 0.19 | 0.20 | GND |
| 0.21        | 0.22 | 2.11 | GND |
| 2.12        | 2.13 | 0.11 | GND |
| 0.10        | 0.1  | 0.0  | GND |

Ground **BLACK**

Signal **WHITE (1.28)**

|      |      |
|------|------|
| +5V  | +5V  |
| GND  | GND  |
| 1.29 | 1.28 |
| +5V  | +5V  |
| GND  | GND  |
| 1.27 | 1.26 |
| +5V  | +5V  |
| GND  | GND  |
| 1.25 | 1.24 |

- Go to the next page.

# Marlin 2.0.x Setup for Filament Runout Sensor Connected to X+ Endstop Connector

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- Ensure the **FIL\_RUNOUT\_PIN** is defined in pins\_BTT\_SKR\_V1\_3.h file. The pins file for SKR V1.3 board is located in "...\\Marlin\\src\\pins\\lpc1768\\subdirectory.
- **FIL\_RUNOUT\_PIN** should be equal to **P1\_28**

```

File Edit Selection View Go Run Terminal Help pins_BTT_SKR_V1_3.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h pins_BTT_SKR_V1_3.h Configuration_adv.h
OPEN EDITORS Configuration.h Marlin pins_BTT_SKR_V1_3.h Marlin\src... pins_BTT_SKR_common.h Marlin...
MARLIN-2.0.X Configuration_adv.h Marlin
src core feature gcode HAL inc lcd libs module pins esp32 linux lpc1768
pins AZSMZ_MINI.h BQU_B300_V1.0.h BQU_BQ111_A4.h pins_BTT_SKR_common.h pins_BTT_SKR_V1.1.h pins_BTT_SKR_V1_3.h pins_BTT_SKR_V1_4.h GMARSH_X6.RE.h MKS_SBASE.h MKS_SGEN_Lh RAMPs_RE_ARM.h SELENA_COMPACT.h lpc1769 mega rambo ramps
pins_BTT_SKR_V1_3.h
#define FIL_RUNOUT_PIN P1_28
// Steppers
#define X_STEP_PIN P2_02
#define X_DIR_PIN P2_06
#define X_ENABLE_PIN P2_01
// Y_SCK_PIN
#endif
#define Y_STEP_PIN P0_19
#define Y_DIR_PIN P0_28
#define Y_ENABLE_PIN P2_08
#ifndef Y_CS_PIN
#define Y_CS_PIN P1_15
#endif
#define Z_STEP_PIN P0_22
#define Z_DIR_PIN P2_11
#define Z_ENABLE_PIN P0_21

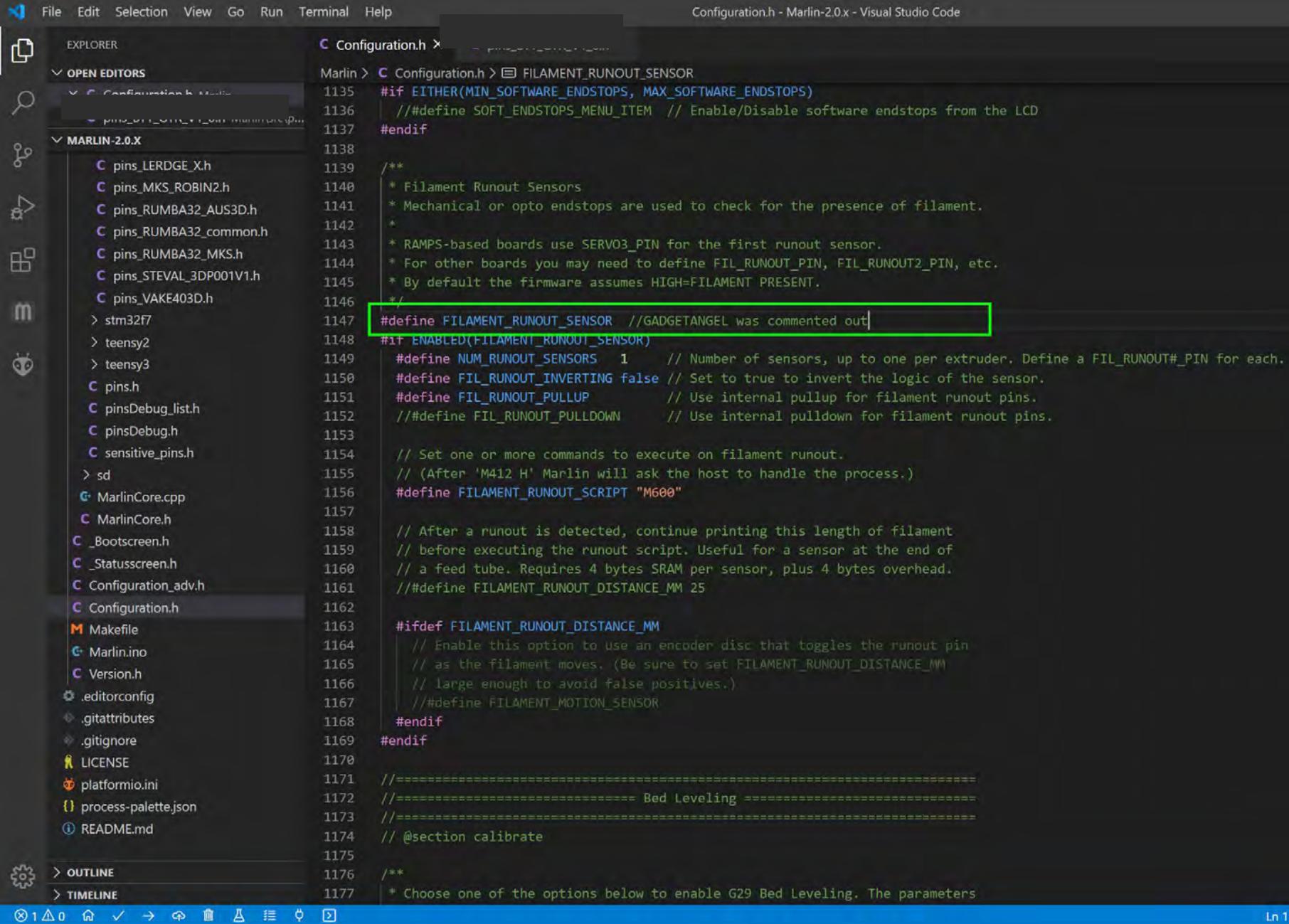
```

Ln 100, Col 49 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## Marlin 2.0.x Setup for Filament Runout Sensor

- In Configuration.h file remove the forward slashes in front of `#define FILAMENT_RUNOUT_SENSOR` line



```

File Edit Selection View Go Run Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER
OPEN EDITORS
MARLIN-2.0.X
pins_LERDGE_X.h
pins_MKS_ROBIN2.h
pins_RUMBA32_AUS3D.h
pins_RUMBA32_common.h
pins_RUMBA32_MKS.h
pins_STEVAL_3DP001V1.h
pins_VAKE403D.h
stm32f7
teensy2
teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.editorconfig
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

OUTLINE
TIMELINE
Ln 1

```

```

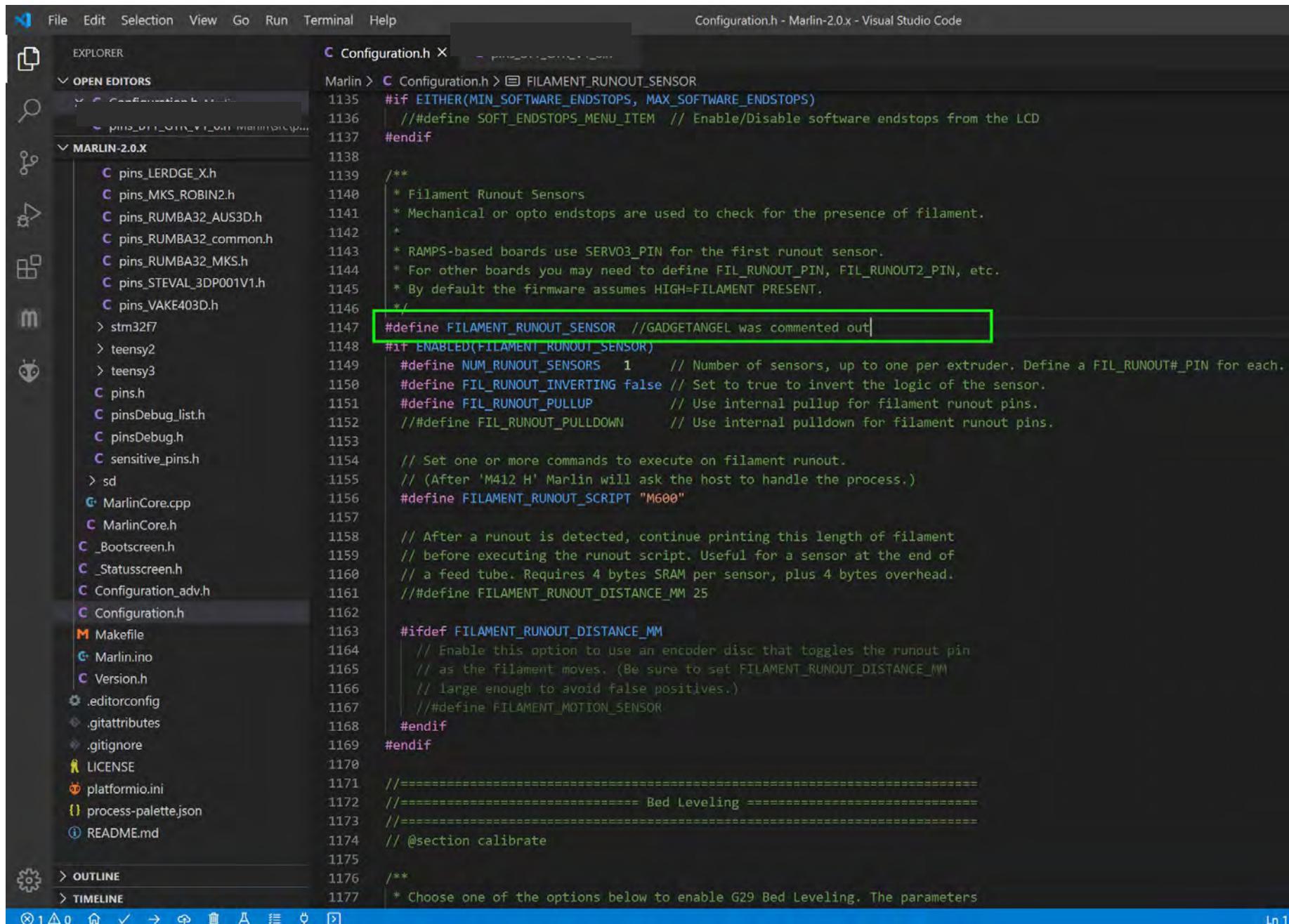
C Configuration.h x
Marlin > C Configuration.h > FILAMENT_RUNOUT_SENSOR
1135 #if EITHER(MIN_SOFTWARE_ENDSTOPS, MAX_SOFTWARE_ENDSTOPS)
1136 // #define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops from the LCD
1137 #endif
1138 /**
1139 * Filament Runout Sensors
1140 * Mechanical or opto endstops are used to check for the presence of filament.
1141 *
1142 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
1143 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN, etc.
1144 * By default the firmware assumes HIGH=FILAMENT PRESENT.
1145 */
1146 #define FILAMENT_RUNOUT_SENSOR //GADGETANGEL was commented out
1147 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1148 #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to one per extruder. Define a FIL_RUNOUT#_PIN for each.
1149 #define FIL_RUNOUT_INVERTING false // Set to true to invert the logic of the sensor.
1150 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins.
1151 // #define FIL_RUNOUT_PULLDOWN // Use internal pulldown for filament runout pins.
1152
1153 // Set one or more commands to execute on filament runout.
1154 // (After 'M412 H' Marlin will ask the host to handle the process.)
1155 #define FILAMENT_RUNOUT_SCRIPT "M600"
1156
1157 // After a runout is detected, continue printing this length of filament
1158 // before executing the runout script. Useful for a sensor at the end of
1159 // a feed tube. Requires 4 bytes SRAM per sensor, plus 4 bytes overhead.
1160 // #define FILAMENT_RUNOUT_DISTANCE_MM 25
1161
1162 #ifdef FILAMENT_RUNOUT_DISTANCE_MM
1163 // Enable this option to use an encoder disc that toggles the runout pin
1164 // as the filament moves. (Be sure to set FILAMENT_RUNOUT_DISTANCE_MM
1165 // large enough to avoid false positives.)
1166 // #define FILAMENT_MOTION_SENSOR
1167 #endif
1168
1169 #endif
1170
1171 //=====
1172 ===== Bed Leveling =====
1173 //=====
1174 // @section calibrate
1175 /**
1176 * Choose one of the options below to enable G29 Bed Leveling. The parameters
1177

```

- Go to the next page.

## Marlin 2.0.x Setup for Filament Runout Sensor

- In Configuration.h file remove the forward slashes in front of `#define FILAMENT_RUNOUT_SENSOR` line



```

File Edit Selection View Go Run Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER
OPEN EDITORS
MARLIN-2.0.X
pins_LERDGE_X.h
pins_MKS_ROBIN2.h
pins_RUMBA32_AUS3D.h
pins_RUMBA32_common.h
pins_RUMBA32_MKS.h
pins_STEVAL_3DP001V1.h
pins_VAKE403D.h
> stm32f
> teensy2
> teensy3
pins.h
pinsDebug_list.h
pinsDebug.h
sensitive_pins.h
> sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration_adv.h
Configuration.h
Makefile
Marlin.ino
Version.h
.editorconfig
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md

> OUTLINE
> TIMELINE
Ln 1

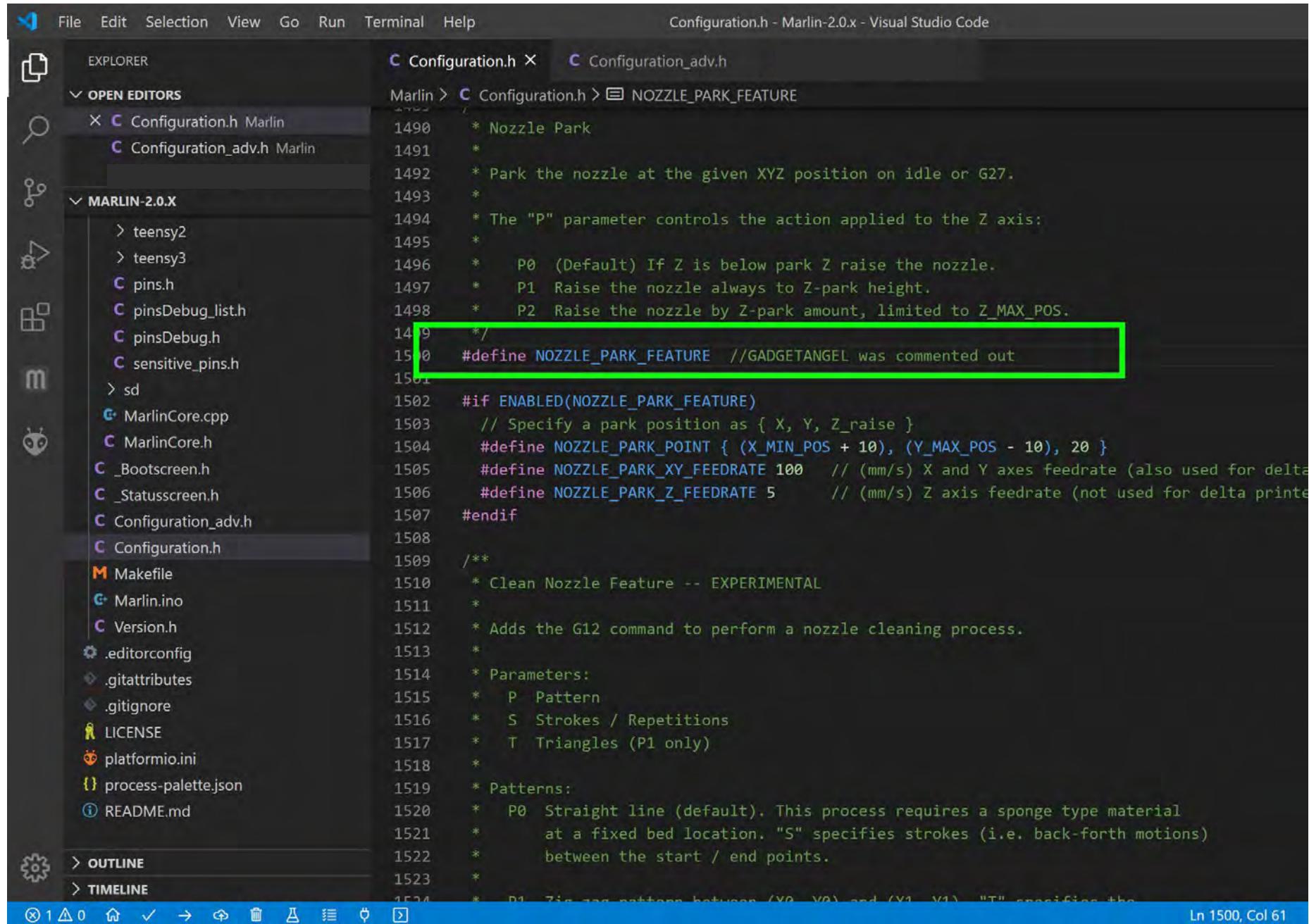
C Configuration.h X
Marlin > C Configuration.h > FILAMENT_RUNOUT_SENSOR
1135 #if EITHER(MIN_SOFTWARE_ENDSTOPS, MAX_SOFTWARE_ENDSTOPS)
1136 //##define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops from the LCD
1137 #endif
1138 /**
1139 * Filament Runout Sensors.
1140 * Mechanical or opto endstops are used to check for the presence of filament.
1141 *
1142 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
1143 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN, etc.
1144 * By default the firmware assumes HIGH=FILAMENT PRESENT.
1145 */
1146 #define FILAMENT_RUNOUT_SENSOR //GADGETANGEL was commented out
1147 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1148 #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to one per extruder. Define a FIL_RUNOUT#_PIN for each.
1149 #define FIL_RUNOUT_INVERTING false // Set to true to invert the logic of the sensor.
1150 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins.
1151 //##define FIL_RUNOUT_PULLDOWN // Use internal pulldown for filament runout pins.
1152
1153 // Set one or more commands to execute on filament runout.
1154 // (After 'M412 H' Marlin will ask the host to handle the process.)
1155 #define FILAMENT_RUNOUT_SCRIPT "M600"
1156
1157 // After a runout is detected, continue printing this length of filament
1158 // before executing the runout script. Useful for a sensor at the end of
1159 // a feed tube. Requires 4 bytes SRAM per sensor, plus 4 bytes overhead.
1160 //##define FILAMENT_RUNOUT_DISTANCE_MM 25
1161
1162 #ifdef FILAMENT_RUNOUT_DISTANCE_MM
1163 // Enable this option to use an encoder disc that toggles the runout pin
1164 // as the filament moves. (Be sure to set FILAMENT_RUNOUT_DISTANCE_MM
1165 // large enough to avoid false positives.)
1166 //##define FILAMENT_MOTION_SENSOR
1167
1168#endif
1169#endif
1170
1171 //=====
1172 //===== Bed Leveling =====
1173 //=====
1174 // @section calibrate
1175 /**
1176 * Choose one of the options below to enable G29 Bed Leveling. The parameters

```

- Go to the next page.

## Marlin 2.0.x Setup for Filament Runout Sensor

- In Configuration.h file remove the forward slashes in front of `#define NOZZLE_PARK_FEATURE` line



```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER Configuration.h X Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > NOZZLE_PARK_FEATURE
X Configuration.h Marlin
C Configuration_adv.h Marlin
MARLIN-2.0.X
 > teensy2
 > teensy3
 C pins.h
 C pinsDebug_list.h
 C pinsDebug.h
 C sensitive_pins.h
 > sd
 C MarlinCore.cpp
 C MarlinCore.h
 C _Bootscreen.h
 C _Statusscreen.h
 C Configuration_adv.h
 C Configuration.h
 M Makefile
 C Marlin.ino
 C Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md

OUTLINE
TIMELINE

Ln 1500, Col 61

```

```

1490 * Nozzle Park
1491 *
1492 * Park the nozzle at the given XYZ position on idle or G27.
1493 *
1494 * The "P" parameter controls the action applied to the Z axis:
1495 *
1496 * P0 (Default) If Z is below park Z raise the nozzle.
1497 * P1 Raise the nozzle always to Z-park height.
1498 * P2 Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
1499 */
1500 #define NOZZLE_PARK_FEATURE //GADGETANGEL was commented out
1501
1502 #if ENABLED(NOZZLE_PARK_FEATURE)
1503 // Specify a park position as { X, Y, Z_raise }
1504 #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
1505 #define NOZZLE_PARK_XY_FEEDRATE 100 // (mm/s) X and Y axes feedrate (also used for delta
1506 #define NOZZLE_PARK_Z_FEEDRATE 5 // (mm/s) Z axis feedrate (not used for delta printers)
1507 #endif
1508 /**
1509 * Clean Nozzle Feature -- EXPERIMENTAL
1510 *
1511 * Adds the G12 command to perform a nozzle cleaning process.
1512 *
1513 * Parameters:
1514 * P Pattern
1515 * S Strokes / Repetitions
1516 * T Triangles (P1 only)
1517 *
1518 * Patterns:
1519 * P0 Straight line (default). This process requires a sponge type material
1520 * at a fixed bed location. "S" specifies strokes (i.e. back-forth motions)
1521 * between the start / end points.
1522 *
1523 * P1 Zig-zag pattern between (X0, Y0) and (X1, Y1). "T" specifies the

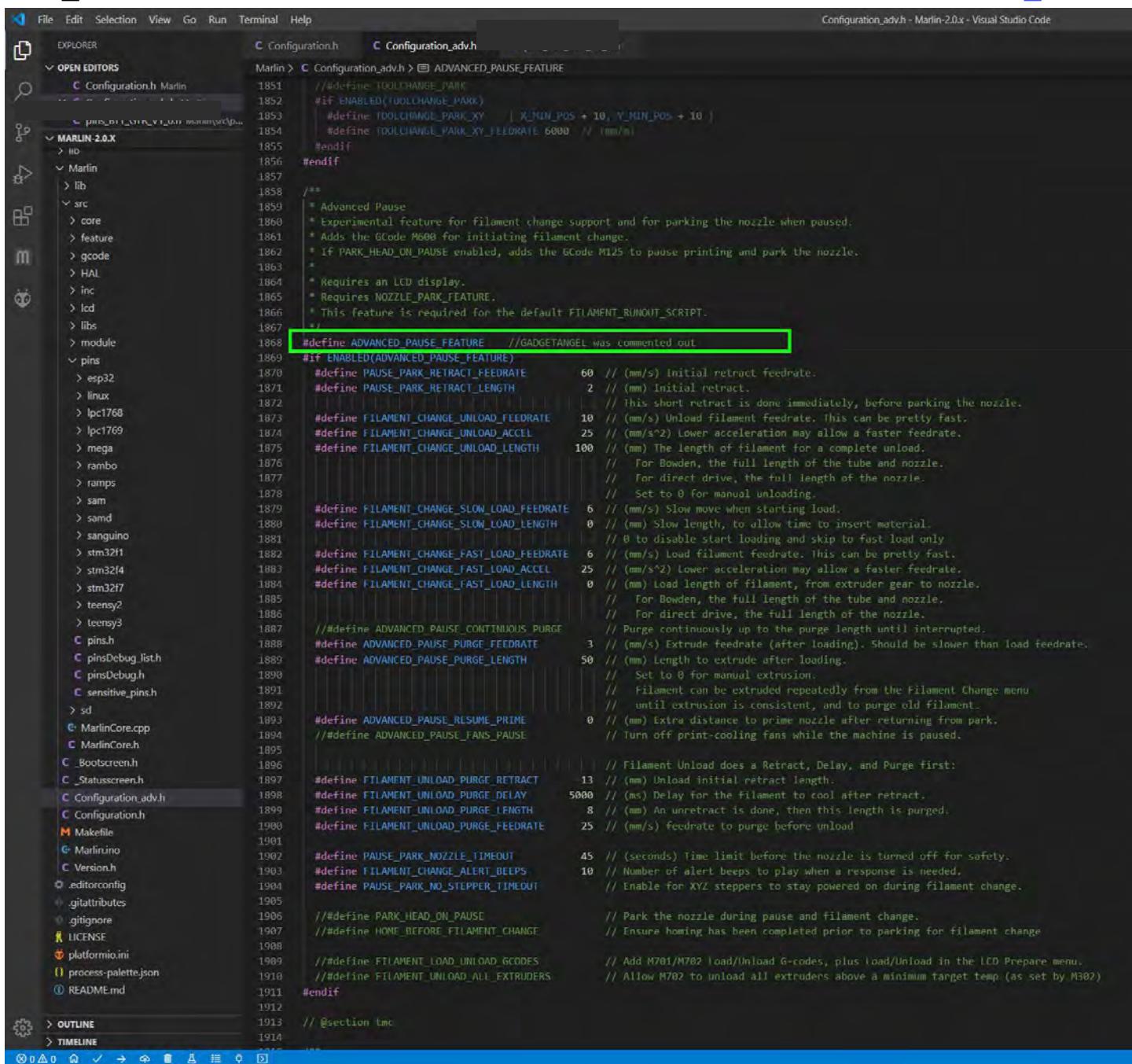
```

- Go to the next page.

## APPENDIX H

# Marlin 2.0.x Setup for Filament Runout Sensor

- In Configuration\_adv.h file remove the forward slashes before #define ADVANCED\_PAUSE\_FEATURE



```

File Edit Selection View Go Run Terminal Help
EXPLORER C Configuration.h C Configuration_adv.h
Marlin > C Configuration_adv.h > ADVANCED_PAUSE_FEATURE
1851 // #define TOOLCHANGE_PARK
1852 #if ENABLED(TOOLCHANGE_PARK)
1853 #define TOOLCHANGE_PARK_XY { X_MIN_POS + 10, Y_MIN_POS + 10 }
1854 #define TOOLCHANGE_PARK_XY_FEEDRATE 6000 // (mm/s)
1855 #endif
1856 #endif
1857 /**
1858 * Advanced Pause
1859 * Experimental feature for filament change support and for parking the nozzle when paused.
1860 * Adds the GCode M600 for initiating filament change.
1861 * If PARK_HEAD_ON_PAUSE enabled, adds the GCode M125 to pause printing and park the nozzle.
1862 *
1863 * Requires an LCD display.
1864 * Requires NOZZLE_PARK_FEATURE.
1865 * This feature is required for the default FILAMENT_RUNOUT_SCRIPT.
1866 */
1867 #define ADVANCED_PAUSE_FEATURE // GADGETANGEL was commented out.
1868 #if ENABLED(ADVANCED_PAUSE_FEATURE)
1869 #define PAUSE_PARK_RETRACT_FEEDRATE 60 // (mm/s) Initial retract feedrate.
1870 #define PAUSE_PARK_RETRACT_LENGTH 2 // (mm) Initial retract.
1871 // This short retract is done immediately, before parking the nozzle.
1872 #define FILAMENT_CHANGE_UNLOAD_FEEDRATE 10 // (mm/s) Unload filament feedrate. This can be pretty fast.
1873 #define FILAMENT_CHANGE_UNLOAD_ACCEL 25 // (mm/s2) Lower acceleration may allow a faster feedrate.
1874 #define FILAMENT_CHANGE_UNLOAD_LENGTH 100 // (mm) The length of filament for a complete unload.
1875 // For Bowden, the full length of the tube and nozzle.
1876 // For direct drive, the full length of the nozzle.
1877 // Set to 0 for manual unloading.
1878 // Set to 0 for manual unloading.
1879 #define FILAMENT_CHANGE_SLOW_LOAD_FEEDRATE 6 // (mm/s) Slow move when starting load.
1880 #define FILAMENT_CHANGE_SLOW_LOAD_LENGTH 0 // (mm) Slow length, to allow time to insert material.
1881 // 0 to disable start loading and skip to fast load only.
1882 #define FILAMENT_CHANGE_FAST_LOAD_FEEDRATE 6 // (mm/s) Load filament feedrate. This can be pretty fast.
1883 #define FILAMENT_CHANGE_FAST_LOAD_ACCEL 25 // (mm/s2) Lower acceleration may allow a faster feedrate.
1884 #define FILAMENT_CHANGE_FAST_LOAD_LENGTH 0 // (mm) Load length of filament, from extruder gear to nozzle.
1885 // For Bowden, the full length of the tube and nozzle.
1886 // For direct drive, the full length of the nozzle.
1887 // Purge continuously up to the purge length until interrupted.
1888 #define ADVANCED_PAUSE_CONTINUOUS_PURGE 3 // (mm/s) Extrude feedrate (after loading). Should be slower than load feedrate.
1889 #define ADVANCED_PAUSE_PURGE_FEEDRATE 50 // (mm) Length to extrude after loading.
1890 // Set to 0 for manual extrusion.
1891 // Filament can be extruded repeatedly from the Filament Change menu
1892 // until extrusion is consistent, and to purge old filament.
1893 #define ADVANCED_PAUSE_RESUME_PRIME 0 // (mm) Extra distance to prime nozzle after returning from park.
1894 // Turn off print-cooling fans while the machine is paused.
1895 // Filament Unload does a Retract, Delay, and Purge first:
1896 #define FILAMENT_UNLOAD_PURGE_RETRACT 13 // (mm) Unload initial retract length.
1897 #define FILAMENT_UNLOAD_PURGE_DELAY 5000 // (ms) Delay for the filament to cool after retract.
1898 #define FILAMENT_UNLOAD_PURGE_LENGTH 8 // (mm) An unretract is done, then this length is purged.
1899 #define FILAMENT_UNLOAD_PURGE_FEEDRATE 25 // (mm/s) Feedrate to purge before unload
1900 // Define PAUSE_PARK_NOZZLE_TIMEOUT
1901 #define PAUSE_PARK_NOZZLE_TIMEOUT 45 // (seconds) Time limit before the nozzle is turned off for safety.
1902 #define FILAMENT_CHANGE_ALERT_BEEPS 10 // Number of alert beeps to play when a response is needed.
1903 // Enable for XYZ steppers to stay powered on during filament change.
1904 #define PAUSE_PARK_NO_STEPPER_TIMEOUT 1905 // Park the nozzle during pause and filament change.
1906 // Ensure homing has been completed prior to parking for filament change
1907 // Add M701/M702 Load/Unload G-codes, plus Load/Unload in the LCD Prepare menu.
1908 // Allow M702 to unload all extruders above a minimum target temp (as set by M302)
1909 // #define FILAMENT_LOAD_UNLOAD_GCODES
1910 // #define FILAMENT_UNLOAD_ALL_EXTRUDERS
1911 #endif
1912 // Section tmc
1913 // Section tmc
1914

```

- Go to the next page.

## Marlin 2.0.x Setup for Filament Runout Sensor

- **keep the Marlin settings the same (as above) until the filament sensor is working. Afterwards you are free to tweak away.**
- **All Finished with setting up the filament runout sensor!!**
- **Use a spare strip of filament and feed it through the filament runout sensor. Start a print, and then pull the filament strip out of the filament runout sensor.**
- **If filament runout sensor is working, the printer will have paused and if this was not a test, you would then load in new filament.**

APPENDIX I

## Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

Connecting SKR V1.3 with Pi 3B+/4 by 3 wires, (2 wires which are serial connection and 1 wire is ground [GND]).  
With 3 wires for serial connections made you no longer need the use of USB cable.<sup>\*1</sup>

- **Raspberry Pi 3B+/4B and SKR V1.3 both use 3.3 V logic.**
- **The idea is to connect (see [Wiring Diagram](#) below)**  
**the TX and RX from the Raspberry Pi to the ARM Processor's**  
**RX and TX on the SKR V1.3 board.**

Therefore, do the following:

- Connect **TXD0** (Raspberry Pi's TX) from Raspberry Pi 3B+/4B connector **TO** the **UART0\_RX (RX0)** from SKR V1.3 board's TFT or AUX-1 connectors.
- Connect **RXD0** (Raspberry Pi's RX) from Raspberry Pi 3B+/4B connector **TO** the **UART0\_TX (TX0)** from SKR V1.3 board's TFT or AUX-1 connectors.
- Connect **GND** from Raspberry Pi 3B+/4B Connector **TO GND** from SKR 1.3 board's TFT or AUX-1 connectors.

DO NOT connect the Raspberry Pi 5V and 3.3V PINS to the SKR V1.3 board. Connecting the 5V and 3.3V PINS would bypass the fuses on the Raspberry Pi board and could cause issues to BOTH the Raspberry Pi board and the SKR V1.3 board. In fact, I recommend powering the Raspberry Pi from an independent and separate 5V power supply but connect the GND on the Raspberry Pi power supply to the GND on the SKR V1.3 board.

- Go to the next page.

# Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the areas on the Raspberry Pi board which are of interest when wanting to connect up the Raspberry Pi to the SKR V1.3 board. This is an overview. The diagram indicates the location of the TXD0 and RXD0 PINS on the Raspberry Pi.

**Pi 3B+/4B @ GPIO connector: PINS used: TXD0, RXD0 and GND (8-10-6):**



|                            |         | Pin no. |    |         |                            |
|----------------------------|---------|---------|----|---------|----------------------------|
| DC Power                   | 3.3V    | 1       | 2  | 5V      | DC Power                   |
| SDA1, I <sup>2</sup> C     | GPIO 2  | 3       | 4  | 5V      | DC Power                   |
| SCL1, I <sup>2</sup> C     | GPIO 3  | 5       | 6  | GND     |                            |
| GPIO_GCLK                  | GPIO 4  | 7       | 8  | GPIO 14 | TXD0                       |
|                            | GND     | 9       | 10 | GPIO 15 | RXD0                       |
| GPIO_GEN0                  | GPIO 17 | 11      | 12 | GPIO 18 | GPIO_GEN1                  |
| GPIO_GEN2                  | GPIO 27 | 13      | 14 | GND     |                            |
| GPIO_GEN3                  | GPIO 22 | 15      | 16 | GPIO 23 | GPIO_GEN4                  |
| DC Power                   | 3.3V    | 17      | 18 | GPIO 24 | GPIO_GEN5                  |
| SPI_MOSI                   | GPIO 10 | 19      | 20 | GND     |                            |
| SPI_MISO                   | GPIO 9  | 21      | 22 | GPIO 25 | GPIO_GEN6                  |
| SPI_CLK                    | GPIO 11 | 23      | 24 | GPIO 8  | SPI_CE0_N                  |
| I <sup>2</sup> C ID EEPROM | GND     | 25      | 26 | GPIO 7  | SPI_CE1_N                  |
|                            | DNC     | 27      | 28 | DNC     | I <sup>2</sup> C ID EEPROM |
|                            | GPIO 5  | 29      | 30 | GND     |                            |
|                            | GPIO 6  | 31      | 32 | GPIO 12 |                            |
|                            | GPIO 13 | 33      | 34 | GND     |                            |
|                            | GPIO 19 | 35      | 36 | GPIO 16 |                            |
|                            | GPIO 26 | 37      | 38 | GPIO 20 |                            |
|                            | GND     | 39      | 40 | GPIO 21 |                            |

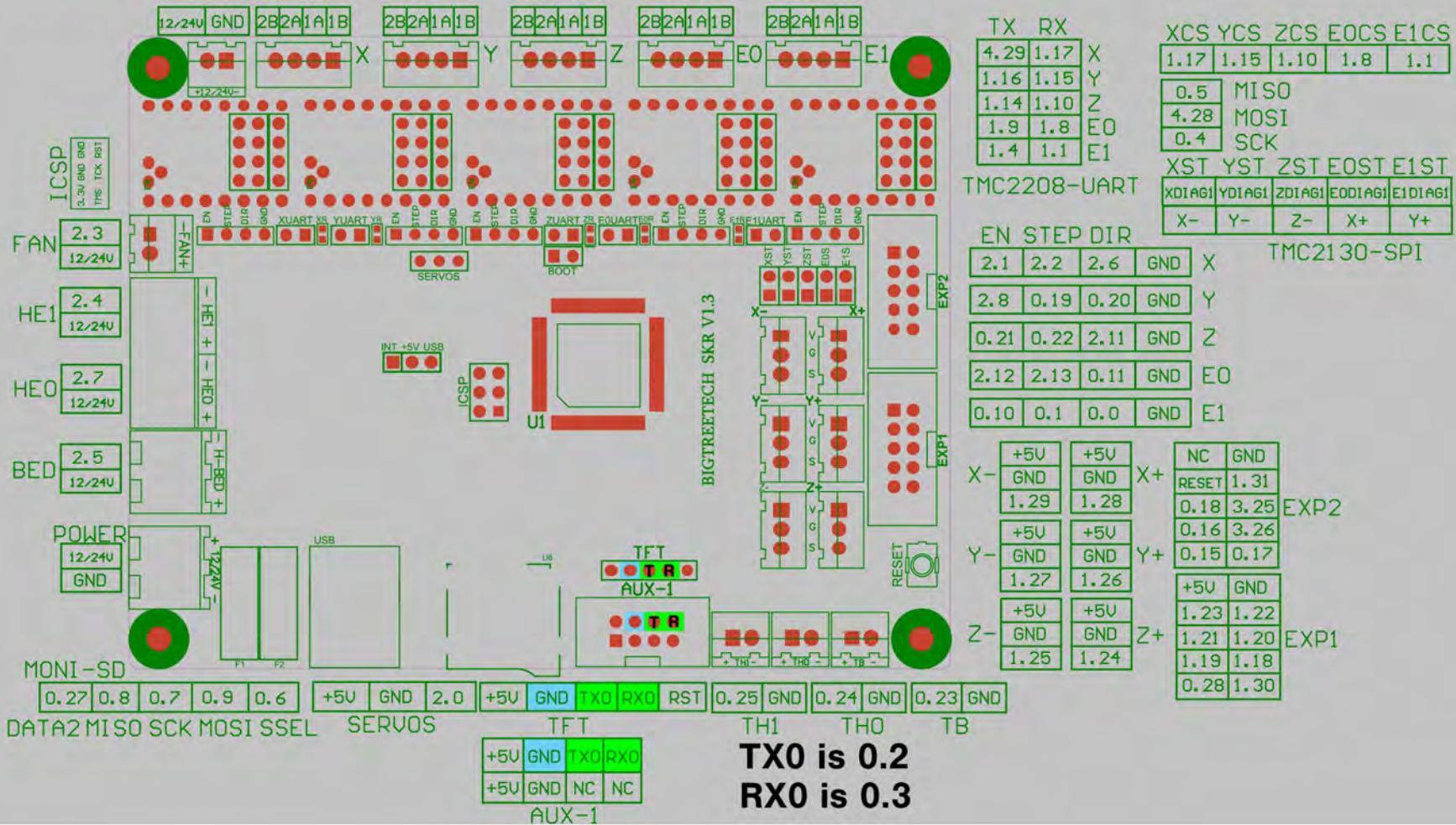
- Go to the next page.

# Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the Raspberry Pi to the SKR V1.3 board. This is an overview. The diagram indicates which PINS are involved.

SKR V1.3 @ TFT or AUX-1 connectors: TX0, RX0 and GND

## BIGTREETECH SKR V1.3-PIN WWW.BIGTREE-TECH.COM

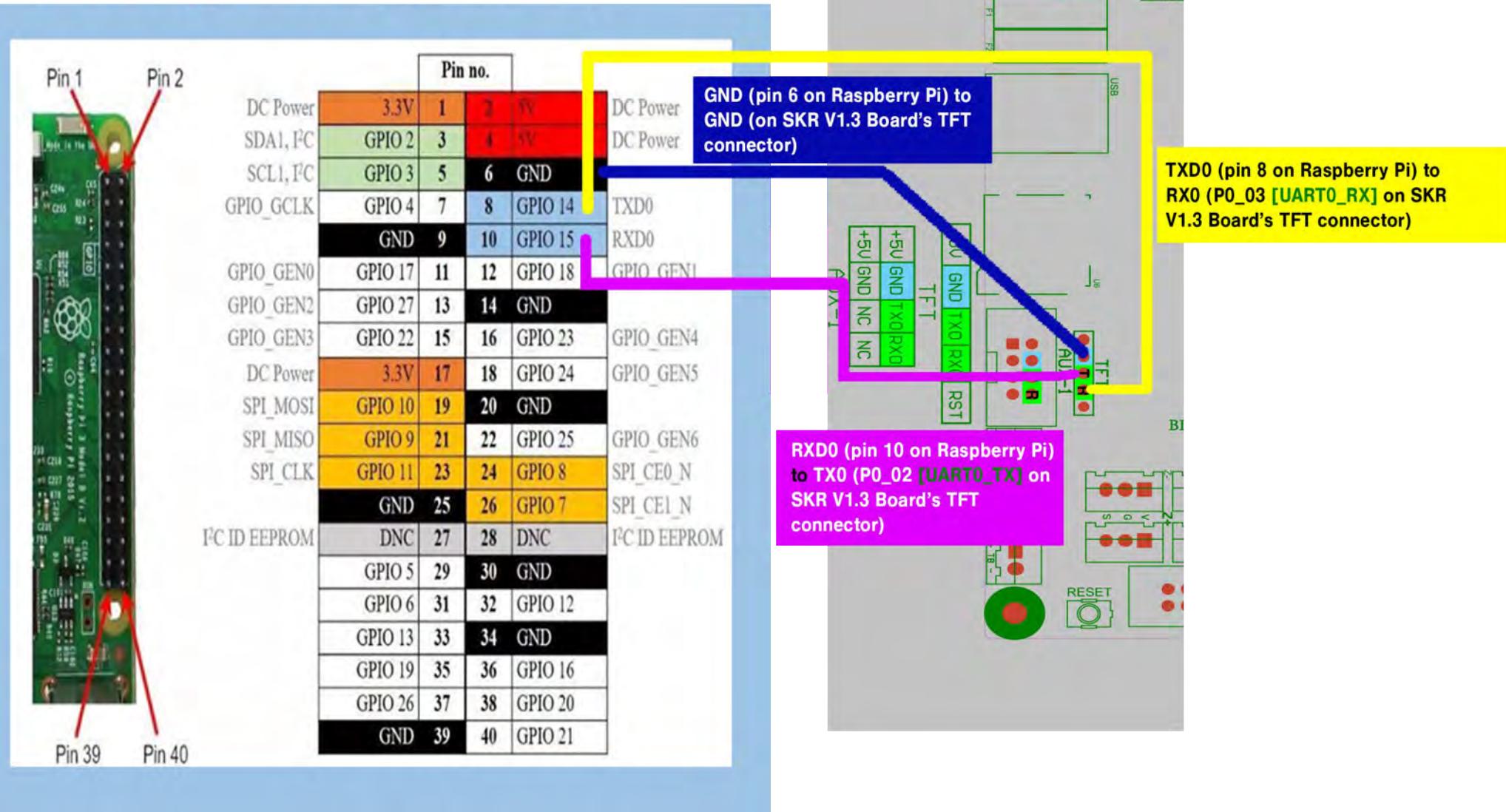


- Go to the next page.

# Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

The diagram below shows the interconnection between the Raspberry Pi and the SKR V1.3 board. This diagram indicates which PINS are involved and how to connect them together.

## Wiring Diagram:



- Go to the next page.

# Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

You also have to change some settings in the Raspberry pi 3B+/4B operating system.

## Raspberry Pi 3B+/4B Changes that need to be made:

### Login to your Raspberry Pi.

Swapping ports used by GPIO and Bluetooth

The first thing to change in the serial connection is to swap the ports used by the GPIO pins and the internal Bluetooth chip. We need to add a line in the boot config file on the boot partition.

Log in to the Pi with SSH on the IP address used.

Type in sudo nano /boot/config.txt

Move the cursor to the end of the file by cursor and add:

dtoverlay=pi3-miniuart-bt

save the file and exit the editor by control+O, hit enter then hit control+X

### Disabling the serial console

Moving to another config file, where part of the code must be deleted to disable serial console.

Type in sudo nano /boot/cmdline.txt

Look for following string (text) and delete it

console=serial0,115200

**Note: This baud rate must be the same speed set in Marlin and in Octoprint!**

save the file and exit the editor by control+O, hit enter then hit control+X

### Rebooting RPi

For all changes to take effect, please reboot your Raspberry Pi

Type in Sudo reboot

### Adding the serial port in Octoprint

Last part of the configuration is in the Octoprint web interface. Open your browser and type either "octoprint.local" or the IP address of the Pi. You might be greeted with the welcome wizard, please go through it first.

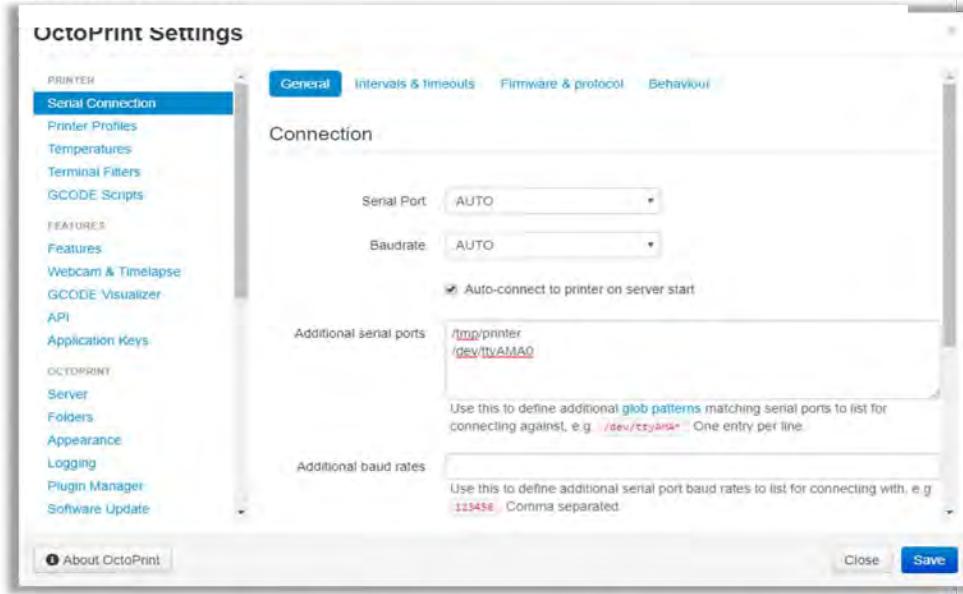
As soon as you arrive at the home screen, open "Settings" (top right), head to "Serial Connection", then "Additional serial ports" and insert following:

/dev/ttyAMA0

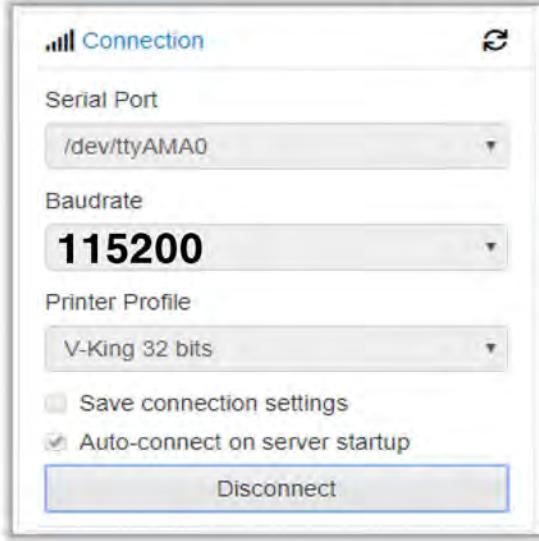
- Go to the next page.

## APPENDIX I

# Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable



Save the change and reboot OctoPrint. After reboot, select the new port and connect to your printer. Making connection might take a few seconds longer than with USB.



**Note: This baud rate must be the same speed set in Marlin and for the console speed on the Raspberry Pi!**

**\*1 This information was taken from BIGTREETECH SKR V1.3 Guide 2019 - 6 .PDF**

- Go to the next page.

## Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- SKR V1.3 board has only 2 numbers that can be used for SERIAL\_PORT or SERIAL\_PORT\_2 Marlin variables: -1 for USB port; 0 for TFT or AUX-1 port.
- The SERIAL\_PORT in Marlin is the one you want to use to get ALL errors reported. The SERIAL\_PORT\_2 is the secondary port. The secondary port will only receive errors on the connections the secondary port sends out on.
- In Configuration.h set SERIAL\_PORT to 0 and SERIAL\_PORT\_2 to -1 using these setting Marlin will use the TX0/RX0 on the TFT connector to communicate with Octoprint. The USB port (-1) will receive secondary information.

```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code

OPEN EDITORS
C Configuration.h Marlin
C pins_BTT_SKR_V1.3.h Marlin
C pins_BTT_SKR_common.h Marlin
C Configuration_adv.h Marlin
MARLIN-2.0.X
> sam
> samd
> sanguino
> stm32f1
> stm32f4
> stm32f7
> teensy2
> teensy3
C pins.h
C pinsDebug.h
C pinsDebug.listh
C sensitive_pins.h
> sd
C MarlinCore.cpp
C MarlinCore.h
C _bootscreen.h
C Statusscreen.h
C Configuration.h
C Configuration_adv.h
M Makefile
C MarlinLino
C Version.h
O editorconfig
O gitattributes
O glogignore
LICENSE
platformio.ini
I process-palette.json
I README.md
> TIMELINE
> TABLE OF CONTENTS

```

```

1 Configuration.h X C pins_BTT_SKR_V1.3.h C pins_BTT_SKR_common.h C Configuration_adv.h
Marlin > C Configuration > SERIAL_PORT_2
95 #define CUSTOM_STATUS_SCREEN_IMAGE
96 // @section machine
97
98 /**
99 * Select the serial port on the board to use for communication with the host.
100 * This allows the connection of wireless adapters (for instance) to non-default port pins.
101 * Serial port -1 is the USB emulated serial port, if available.
102 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
103 */
104
105 #define SERIAL_PORT 0 //GADGETANGEL was 0
106
107 /**
108 * Select a secondary serial port on the board to use for communication with the host.
109 * [-1, 0, 1, 2, 3, 4, 5, 6, 7]
110 */
111 #define SERIAL_PORT_2 -1 //GADGETANGEL was -1 and commented out
112
113 /**
114 * This setting determines the communication speed of the printer.
115 *
116 * 2500000 works in most cases, but you might try a lower speed if
117 * you commonly experience drop-outs during host printing.
118 * You may try up to 1000000 to speed up SD file transfer.
119 */
120 #define BAUDRATE 115200
121
122 /**
123 * Enable the Bluetooth serial interface on AlberoSD devices
124 */
125 #define BLUETOOTH
126
127 // Choose the name from boards.h that matches your setup
128 #ifndef MOTHERBOARD
129 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
130 #endif
131
132 /**
133 * Name displayed in the LCD "Ready" message and Info menu
134 */
135 #define CUSTOM_MACHINE_NAME "Ender-3"
136
137 // Printer's unique ID used by some services to differentiate between machines

```

Ln 113, Col 25 Spaces: 2 UTF-8 LF C++ Wind2 RP D

**Note:** This baud rate must be the same speed set for the console speed on the Raspberry Pi and the same speed set in Octoprint!

## Connecting SKR V1.3 to Raspberry Pi to Eliminate USB Cable

- keep the Marlin settings the same (as above) until the serial communication with the Raspberry Pi is working. Afterwards you are free to tweak away.
- All Finished with setting up the serial connection with the Raspberry Pi!!

## Connecting SKR V1.3 with BLTouch

To install a BLTouch to your printer requires three steps:

- Mount the BLTouch to your printer's print head. Search Thingiverse for a BLTouch mount for your specific printer and print head.
- Wiring the BLTouch to the MCU board (SKR V1.3 board).
- Marlin 2.0.x Firmware setup.

- Go to the next page.

## APPENDIX J

### Connecting SKR V1.3 with BLTouch

The diagram below shows the areas on the BLTouch Sensor which are of interest when wanting it up to the SKR V1.3 board. This is an overview. The diagram indicates the location of the PINS on the BLTouch that will be used when connecting it up to the SKR V1.3 board.

**ANTCLABS BLTouch @ BLTouch 3 PIN connector:** **Brown** (GND), **Red** (+5V) **Orange** (control signal)

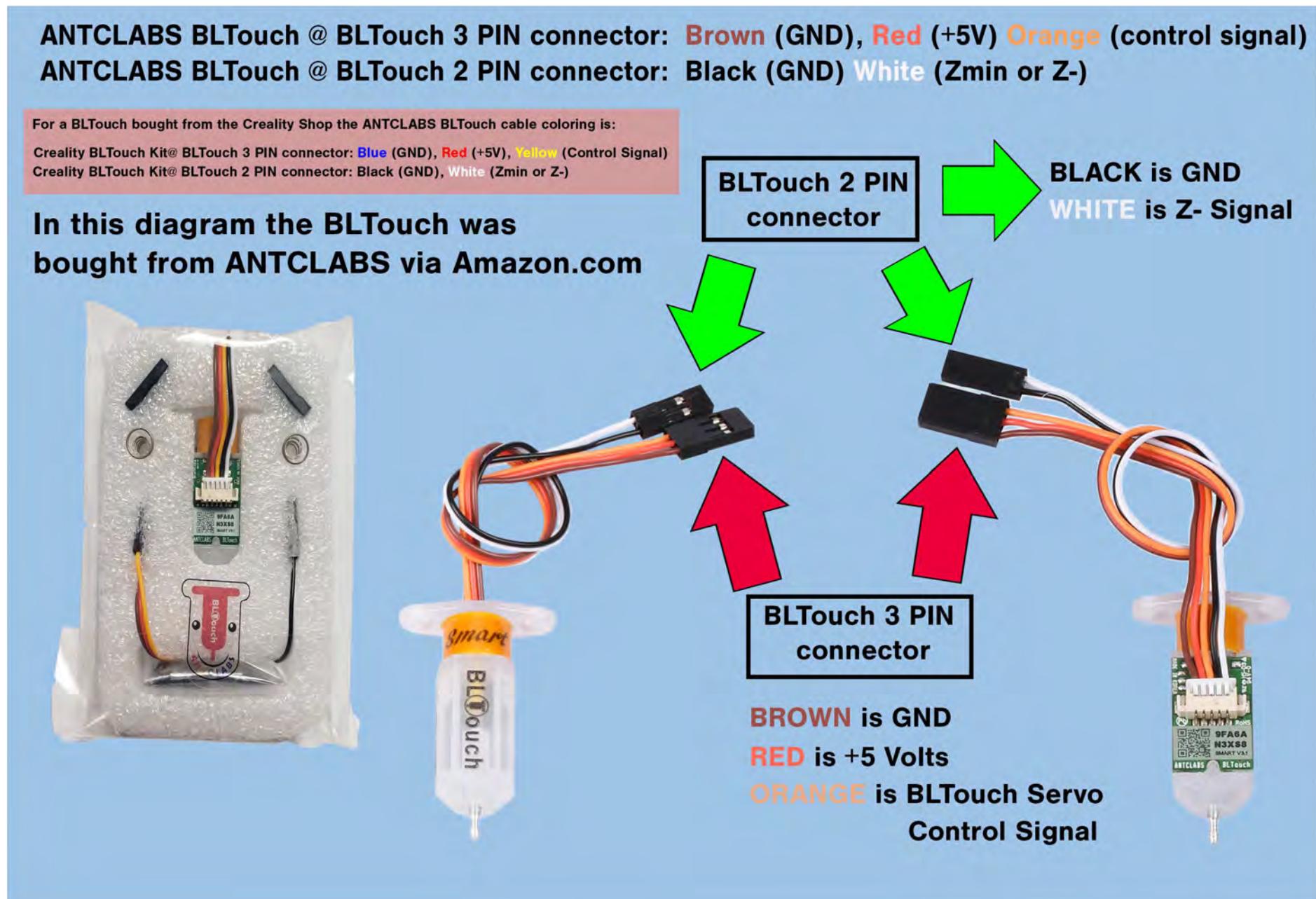
**ANTCLABS BLTouch @ BLTouch 2 PIN connector:** **Black** (GND) **White** (Zmin or Z-)

For a BLTouch bought from the Creality Shop the ANTCLABS BLTouch cable coloring is:

Creality BLTouch Kit@ BLTouch 3 PIN connector: **Blue** (GND), **Red** (+5V), **Yellow** (Control Signal)

Creality BLTouch Kit@ BLTouch 2 PIN connector: **Black** (GND), **White** (Zmin or Z-)

In this diagram the BLTouch was  
bought from ANTCLABS via Amazon.com



- Go to the next page.

## APPENDIX J

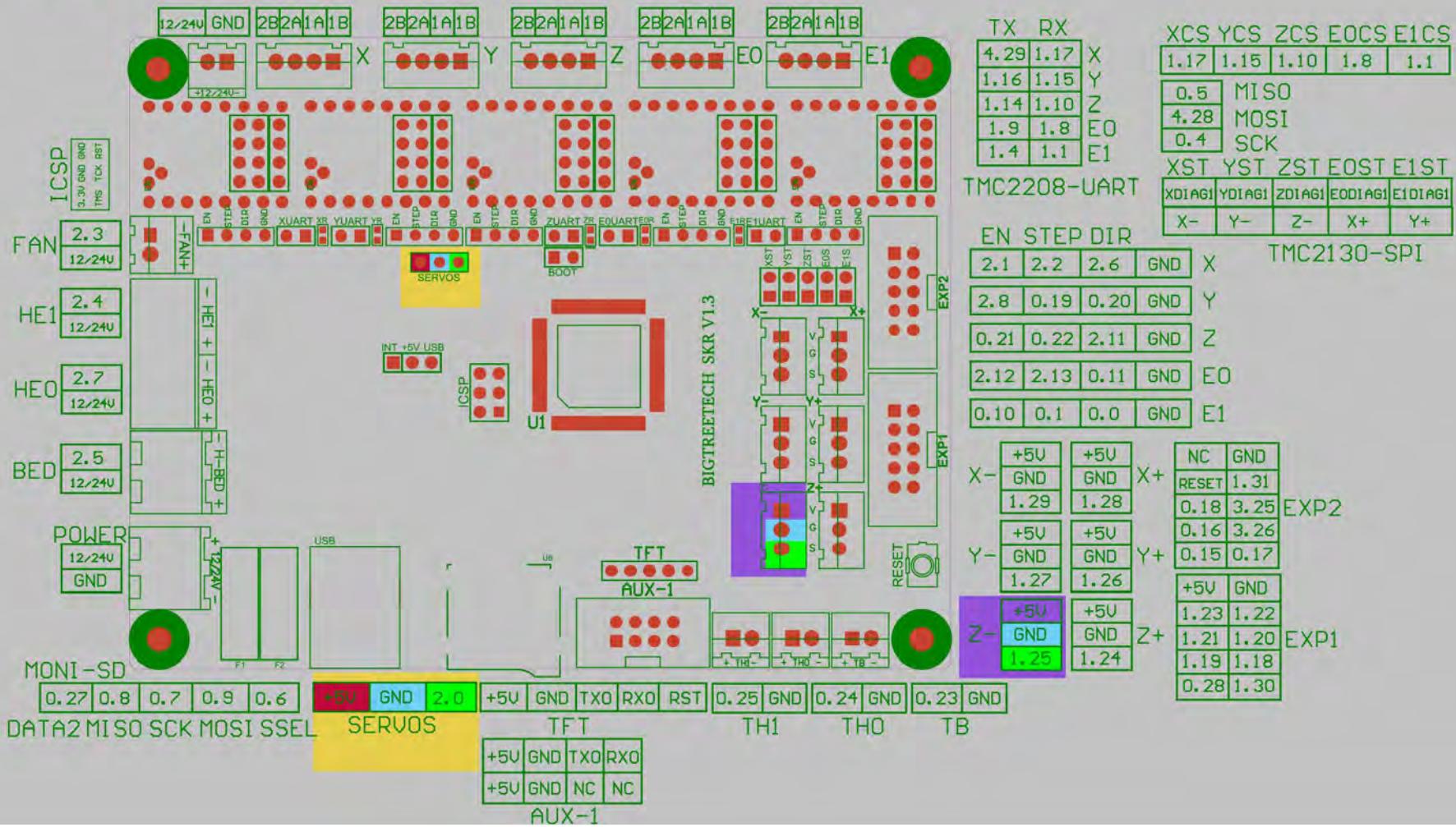
### Connecting SKR V1.3 with BLTouch

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the BLTouch to the SKR V1.3 board. This is an overview. The diagram indicates which PINS are involved.

**SKR V1.3 @ SERVOS header: +5V, GND, PIN 2.0; AND SKR V1.3 @ Z- connector: GND, PIN 1.25**

## BIGTREETECH SKR V1.3-PIN

[WWW.BIGTREE-TECH.COM](http://WWW.BIGTREE-TECH.COM)

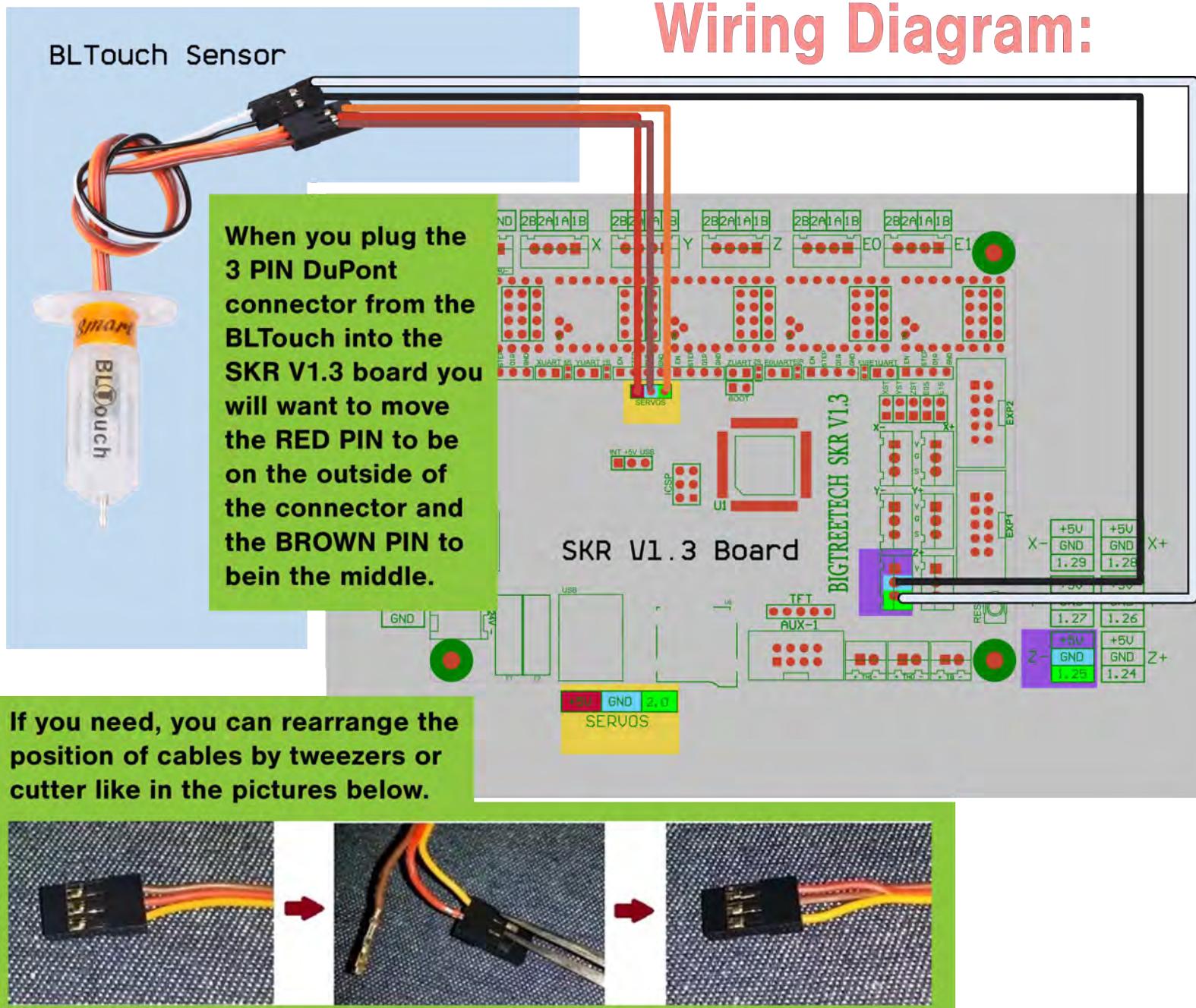


- Go to the next page.

## APPENDIX J

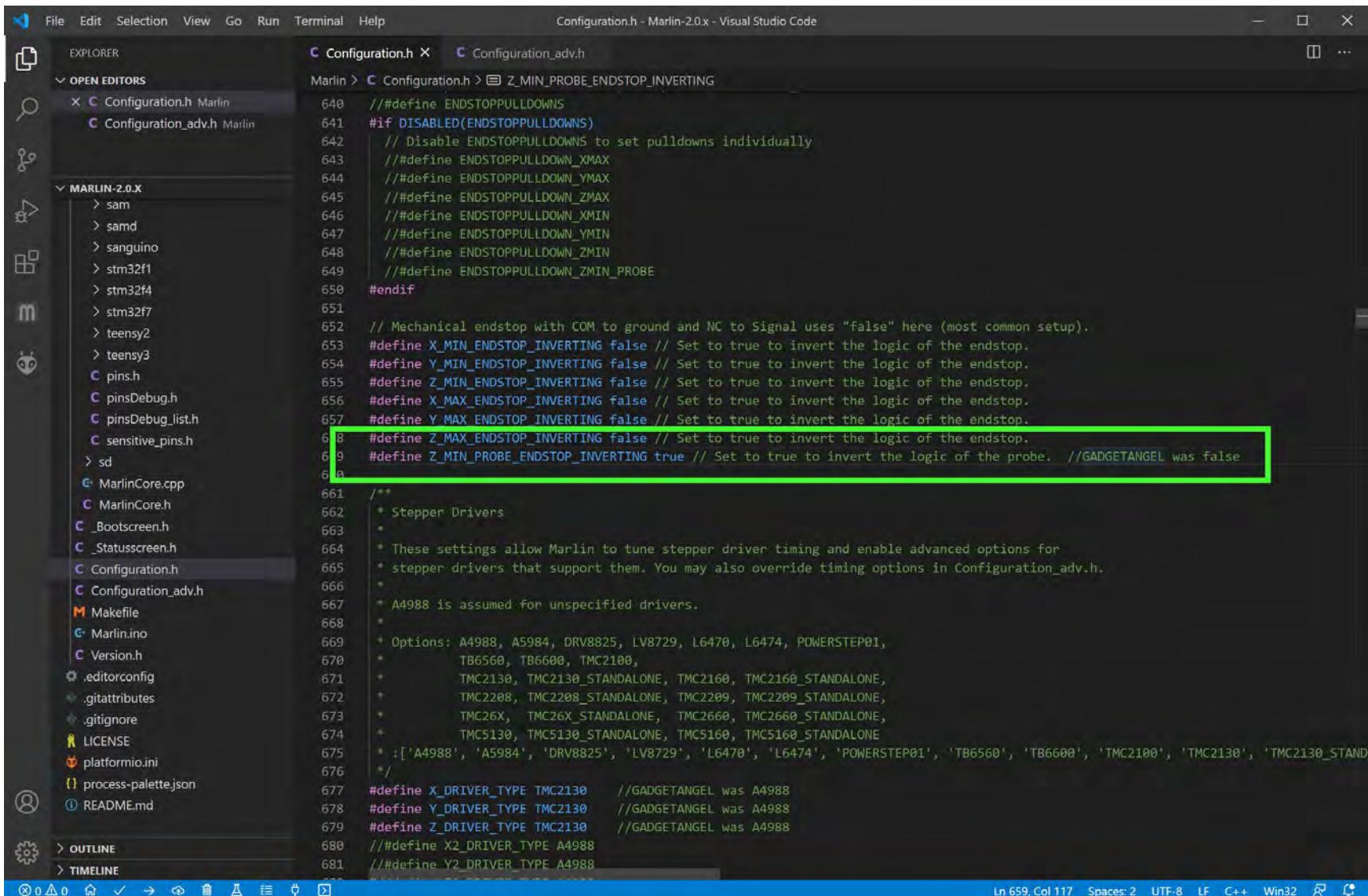
### Connecting SKR V1.3 with BLTouch

The diagram below shows the interconnection between the BLTouch and the SKR V1.3 board. This diagram indicates which PINS are involved and how to connect them together.



**APPENDIX J****Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- change the Marlin variable **Z\_MIN\_PROBE\_ENDSTOP\_INVERTING** to true



```

File Edit Selection View Go Run Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code

EXPLORER OPEN EDITORS MARLIN-2.0.X
 Configuration.h Marlin
 Configuration_adv.h Marlin
 sam
 samd
 sanguino
 stm32f1
 stm32f4
 stm32f7
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 README.md

 Configuration.h X Configuration_adv.h
 Marlin > Configuration.h > Z_MIN_PROBE_ENDSTOP_INVERTING
 640 // #define ENDSTOPPULLDOWNS
 641 #if DISABLED(ENDSTOPPULLDOWNS)
 642 // Disable ENDSTOPPULLDOWNS to set pulldowns individually
 643 // #define ENDSTOPPULLDOWN_XMAX
 644 // #define ENDSTOPPULLDOWN_YMAX
 645 // #define ENDSTOPPULLDOWN_ZMAX
 646 // #define ENDSTOPPULLDOWN_XMIN
 647 // #define ENDSTOPPULLDOWN_YMIN
 648 // #define ENDSTOPPULLDOWN_ZMIN
 649 // #define ENDSTOPPULLDOWN_ZMIN_PROBE
 650 #endif
 651
 652 // Mechanical endstop with COM to ground and NC to Signal uses "false" here (most common setup).
 653 #define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 654 #define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 655 #define Z_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 656 #define X_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 657 #define Y_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 658 #define Z_MAX_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
 659 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // Set to true to invert the logic of the probe. //GADGETANGEL was false
 660
 661 /**
 * Stepper Drivers
 *
 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 *
 * A4988 is assumed for unspecified drivers.
 *
 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 * TB6560, TB6600, TMC2100,
 * TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 * TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 * TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 * TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 * * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_STAND
 * */
 677 #define X_DRIVER_TYPE TMC2130 //GADGETANGEL was A4988
 678 #define Y_DRIVER_TYPE TMC2130 //GADGETANGEL was A4988
 679 #define Z_DRIVER_TYPE TMC2130 //GADGETANGEL was A4988
 680 // #define X2_DRIVER_TYPE A4988
 681 // #define Y2_DRIVER_TYPE A4988

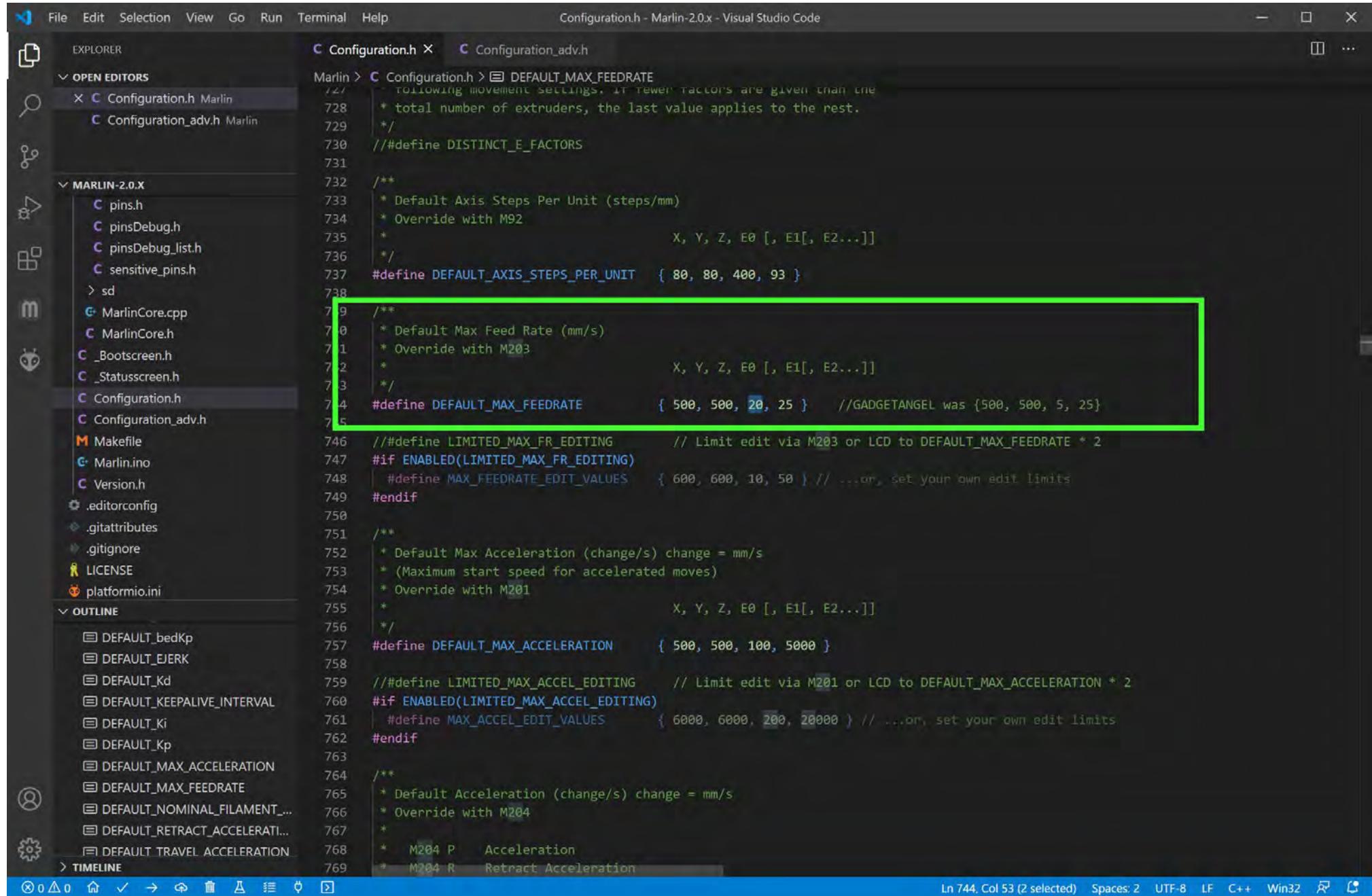
```

Ln 659, Col 117 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

# Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- change the Z value of the Marlin variable **DEFAULT\_MAX\_FEEDRATE** to 20



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x firmware configuration. A green rectangular box highlights the line:

```
#define DEFAULT_MAX_FEEDRATE { 500, 500, 20, 25 } //GADGETANGEL was {500, 500, 5, 25}
```

This line defines the default maximum feed rate for all axes (X, Y, Z, E0, E1, E2) as 500 mm/s for the first two axes and 20 mm/s for the last two axes. The original value in the comment was {500, 500, 5, 25}.

- Go to the next page.

**APPENDIX 3**  
**Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- ensure the Marlin variable `Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN` is **enabled** or that you **delete the two forward slashes ("//") from the beginning of the line.**

The screenshot shows the Visual Studio Code interface with the following details:

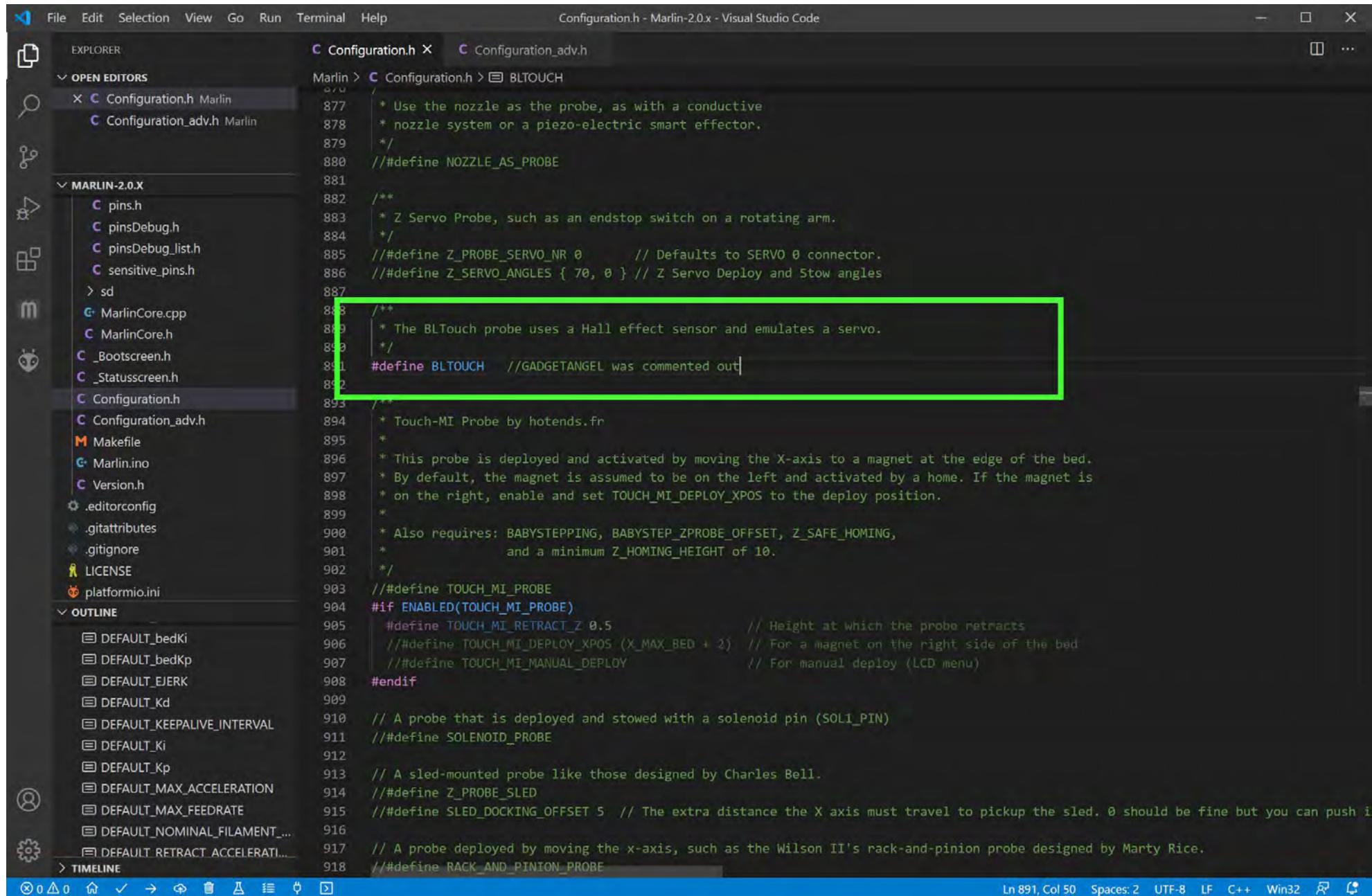
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Sidebar:** EXPLORER, OPEN EDITORS, MARLIN-2.0.X, OUTLINE, TIMELINE.
- Open Editors:** Configuration.h (Marlin), Configuration\_adv.h (Marlin).
- Code Editor:** The code for Configuration.h is displayed. A green rectangular box highlights the following section of code:

```
826 //
827 // See http://marlinfw.org/docs/configuration/probes.html
828 //
829 /**
830 * Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
831 *
832 * Enable this option for a probe connected to the Z Min endstop pin.
833 */
834 #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```
- Bottom Status Bar:** Line 835, Column 43, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **enable** the Marlin variable **BLTOUCH** or that you **delete the two forward slashes ("//") from the beginning of the line.**

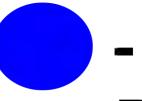


The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x firmware configuration. A green rectangular box highlights the line:

```
#define BLTOUCH //GADGETANGEL was commented out
```

This line is located in the 'BLTOUCH' section of the configuration file. The code editor's status bar at the bottom right indicates 'Ln 891, Col 50'.

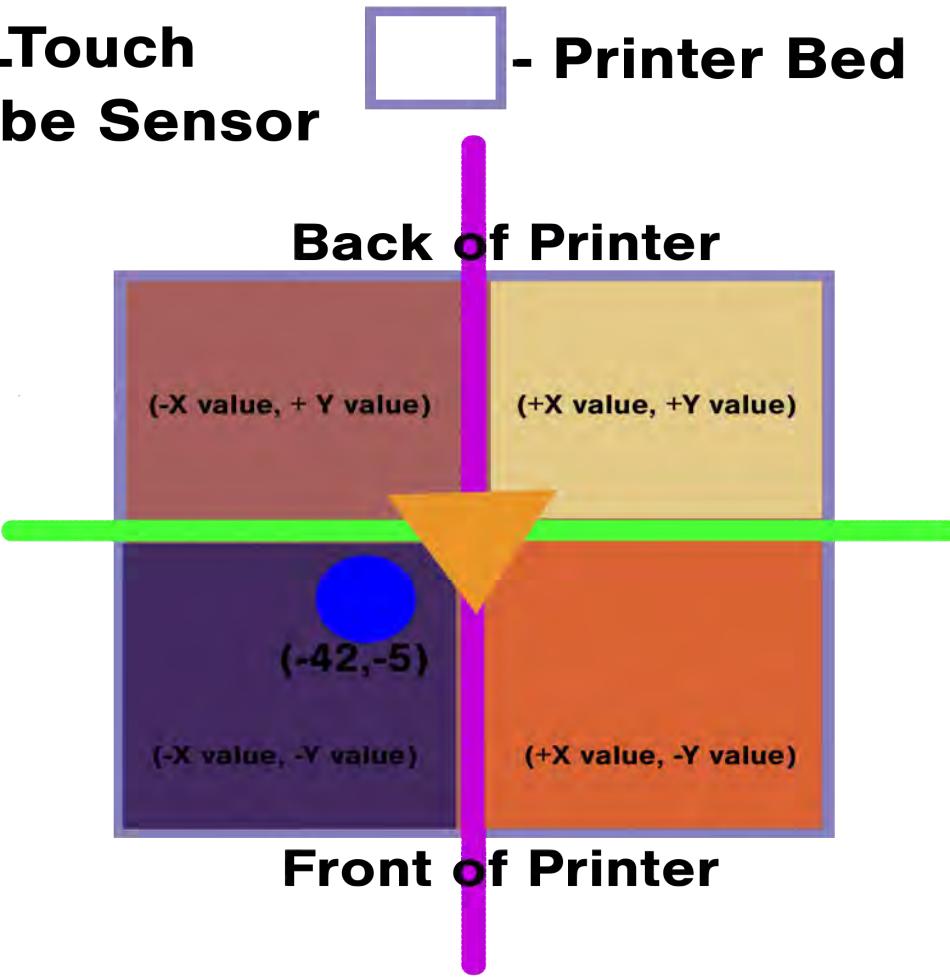
- Go to the next page.

**APPENDIX J****Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board****Understanding Marlin Firmware's NOZZLE\_TO\_PROBE\_OFFSET Setting****- NOZZLE****- BLTouch Probe Sensor****- Printer Bed**

**The BLTouch Probe Sensor can be located in one of four quadrants. The Nozzle is at the center.**

**The diagram on this page shows the X and Y values for each quadrant.**

**If the BLTouch is located in FRONT of the Green Line or in FRONT of the NOZZLE then the Y value for the Probe-to-Nozzle-Offset will be a negative (-) value. If the BLtouch is located BEHIND the Green Line or BEHIND the NOZZLE then the Y value for the Probe-to-Nozzle-Offset will be a positive (+) value. If the BLTouch is located to the LEFT of the Pink Line or to the LEFT of the NOZZLE then the X value for the Probe-to-Nozzle-Offset will be a negative (-) value. If the BLTouch is located to the RIGHT of the Pink Line or to the RIGHT of the NOZZLE then the X value for the Probe-to-Nozzle-Offset will be a positive (+) value.**

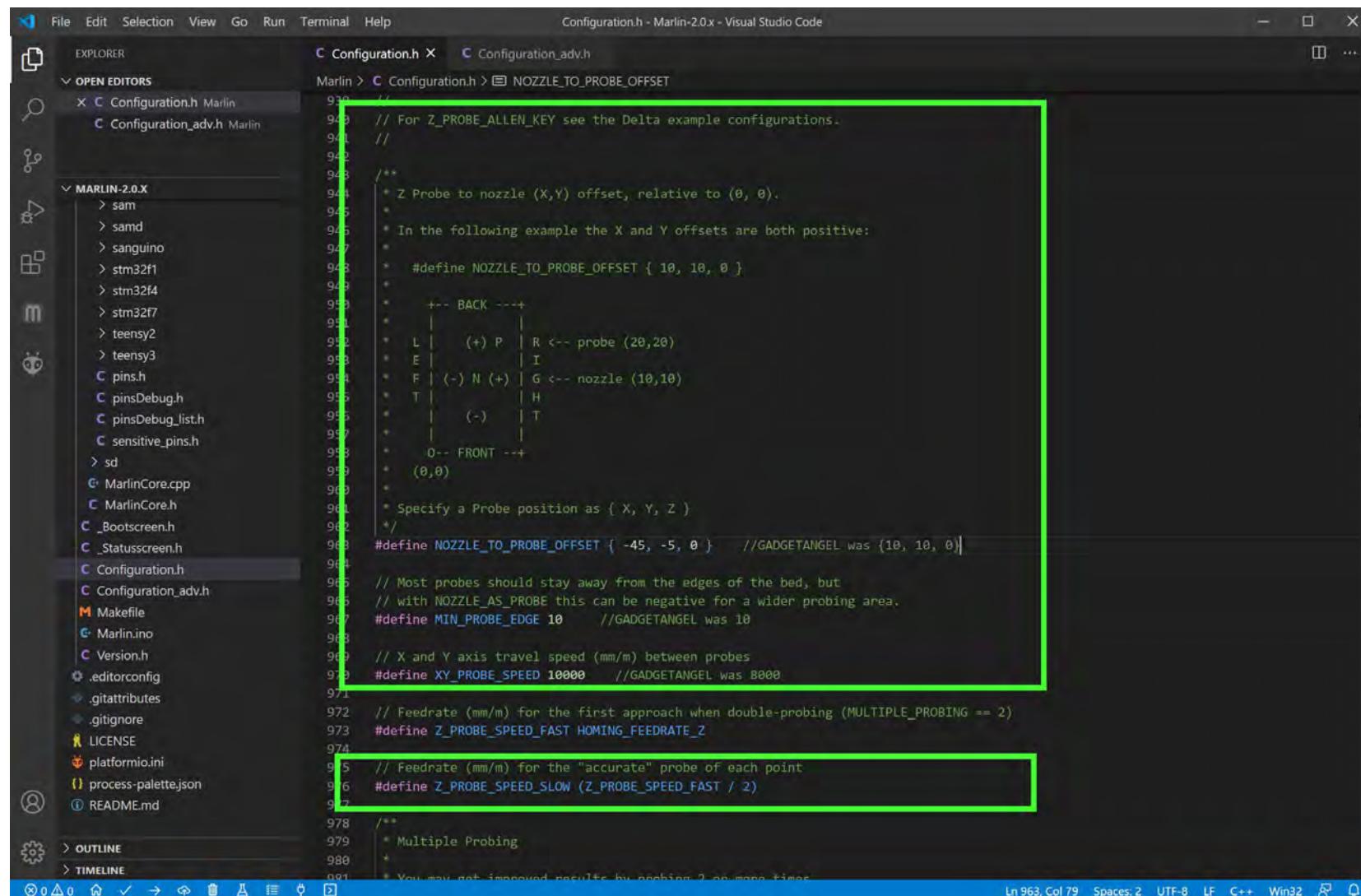


**In the above diagram the BLTouch will have a -X value and -Y value for the NOZZLE\_TO\_PROBE\_OFFSET values. In the above example, the Probe-to-Nozzle-Offset values are -42 for X and -5 for Y.**

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- change the X and Y values of the Marlin variable **NOZZLE\_TO\_PROBE\_OFFSET** to match your measurements from your 3D printer. In the example shown from the previous page, the X value is equal to -42 and the Y value is equal to -5.
- set the Marlin variable **MIN\_PROBE\_EDGE** is set for 10
- set the Marlin variable **XY\_PROBE\_SPEED** to 10000
- ensure the Marlin variable **Z\_PROBE\_SPEED\_SLOW** is equal to **Z\_PROBE\_SPEED\_FAST / 2**



```

File Edit Selection View Go Run Terminal Help
Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER OPEN EDITORS C Configuration.h X C Configuration_adv.h
Marlin > C Configuration.h > NOZZLE_TO_PROBE_OFFSET
939 // For Z_PROBE_ALLEN_KEY see the Delta example configurations.
940 //
941 /**
942 * Z Probe to nozzle (X,Y) offset, relative to (0, 0).
943 *
944 * In the following example the X and Y offsets are both positive:
945 *
946 * #define NOZZLE_TO_PROBE_OFFSET { 10, 10, 0 }
947 *
948 * +-+ BACK +-+
949 * | | (+) P | R <- probe (20,20)
950 * | | | I
951 * F | (-) N (+) | G <- nozzle (10,10)
952 * | T | | H
953 * | | (-) | T
954 * |
955 * O-- FRONT --+
956 * (0,0)
957 *
958 * Specify a Probe position as { X, Y, Z }
959 */
960 #define NOZZLE_TO_PROBE_OFFSET { -45, -5, 0 } //GADGETANGEL was {10, 10, 0}
961 //
962 // Most probes should stay away from the edges of the bed, but
963 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
964 #define MIN_PROBE_EDGE 10 //GADGETANGEL was 10
965 //
966 // X and Y axis travel speed (mm/m) between probes
967 #define XY_PROBE_SPEED 10000 //GADGETANGEL was 8000
968 //
969 // Feedrate (mm/m) for the first approach when double-probing (MULTIPLE_PROBING == 2)
970 #define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z
971 //
972 // Feedrate (mm/m) for the "accurate" probe of each point
973 #define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)
974 //
975 /**
976 * Multiple Probing
977 */
978 // You may not understand what this is doing, but it's been here since v2.0.5, so don't touch.
979
980
981

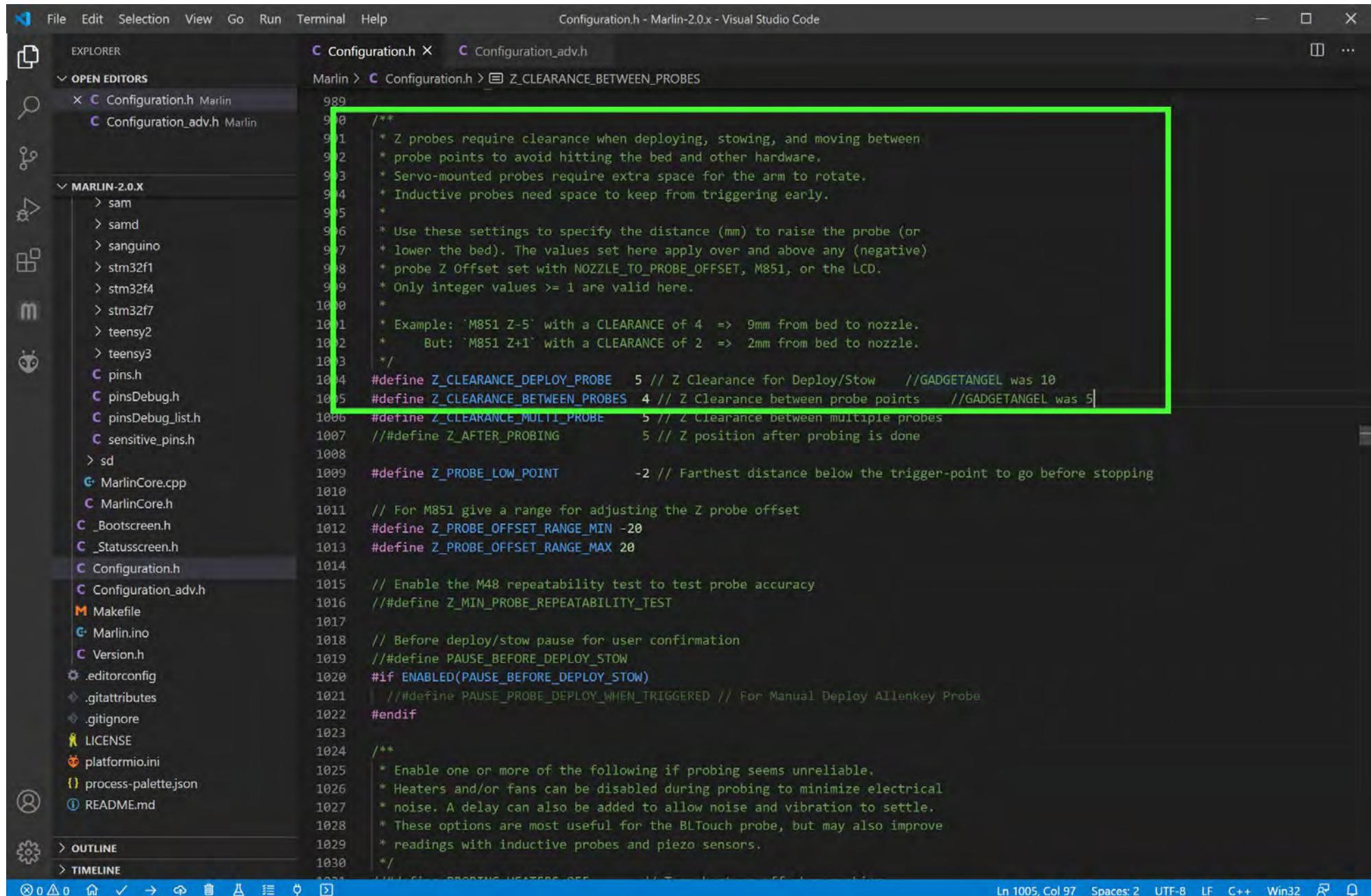
```

Ln 963, Col 79 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- change the Marlin variable **Z\_CLEARANCE\_DEPLOY\_PROBE** to **5**
- change the Marlin variable **Z\_CLEARANCE\_BETWEEN\_PROBES** to **4**



```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER Configuration.h Configuration_adv.h
OPEN EDITORS Marlin > Configuration.h > Z_CLEARANCE_BETWEEN_PROBES
989
990 /**
991 * Z probes require clearance when deploying, stowing, and moving between
992 * probe points to avoid hitting the bed and other hardware.
993 * Servo-mounted probes require extra space for the arm to rotate.
994 * Inductive probes need space to keep from triggering early.
995 *
996 * Use these settings to specify the distance (mm) to raise the probe (or
997 * lower the bed). The values set here apply over and above any (negative)
998 * probe Z Offset set with NOZZLE_TO_PROBE_OFFSET, M851, or the LCD.
999 * Only integer values >= 1 are valid here.
1000 *
1001 * Example: 'M851 Z-5' with a CLEARANCE of 4 => 9mm from bed to nozzle.
1002 * But: 'M851 Z+1' with a CLEARANCE of 2 => 2mm from bed to nozzle.
1003 */
1004 #define Z_CLEARANCE_DEPLOY_PROBE 5 // Z Clearance for Deploy/Stow //GADGETANGEL was 10
1005 #define Z_CLEARANCE_BETWEEN_PROBES 4 // Z Clearance between probe points //GADGETANGEL was 5
1006 #define Z_CLEARANCE_MULTI_PROBE 5 // Z Clearance between multiple probes
1007 //#define Z_AFTER_PROBING 5 // Z position after probing is done
1008
1009 #define Z_PROBE_LOW_POINT -2 // Farthest distance below the trigger-point to go before stopping
1010
1011 // For M851 give a range for adjusting the Z probe offset
1012 #define Z_PROBE_OFFSET_RANGE_MIN -20
1013 #define Z_PROBE_OFFSET_RANGE_MAX 20
1014
1015 // Enable the M48 repeatability test to test probe accuracy
1016 // #define Z_MIN_PROBE_REPEATABILITY_TEST
1017
1018 // Before deploy/stow pause for user confirmation
1019 // #define PAUSE_BEFORE_DEPLOY_STOW
1020 #if ENABLED(PAUSE_BEFORE_DEPLOY_STOW)
1021 // #define PAUSE_PROBE_DEPLOY_WHEN_TRIGGERED // For Manual Deploy Allenkey Probe
1022 #endif
1023
1024 /**
1025 * Enable one or more of the following if probing seems unreliable.
1026 * Heaters and/or fans can be disabled during probing to minimize electrical
1027 * noise. A delay can also be added to allow noise and vibration to settle.
1028 * These options are most useful for the BLTouch probe, but may also improve
1029 * readings with inductive probes and piezo sensors.
1030 */

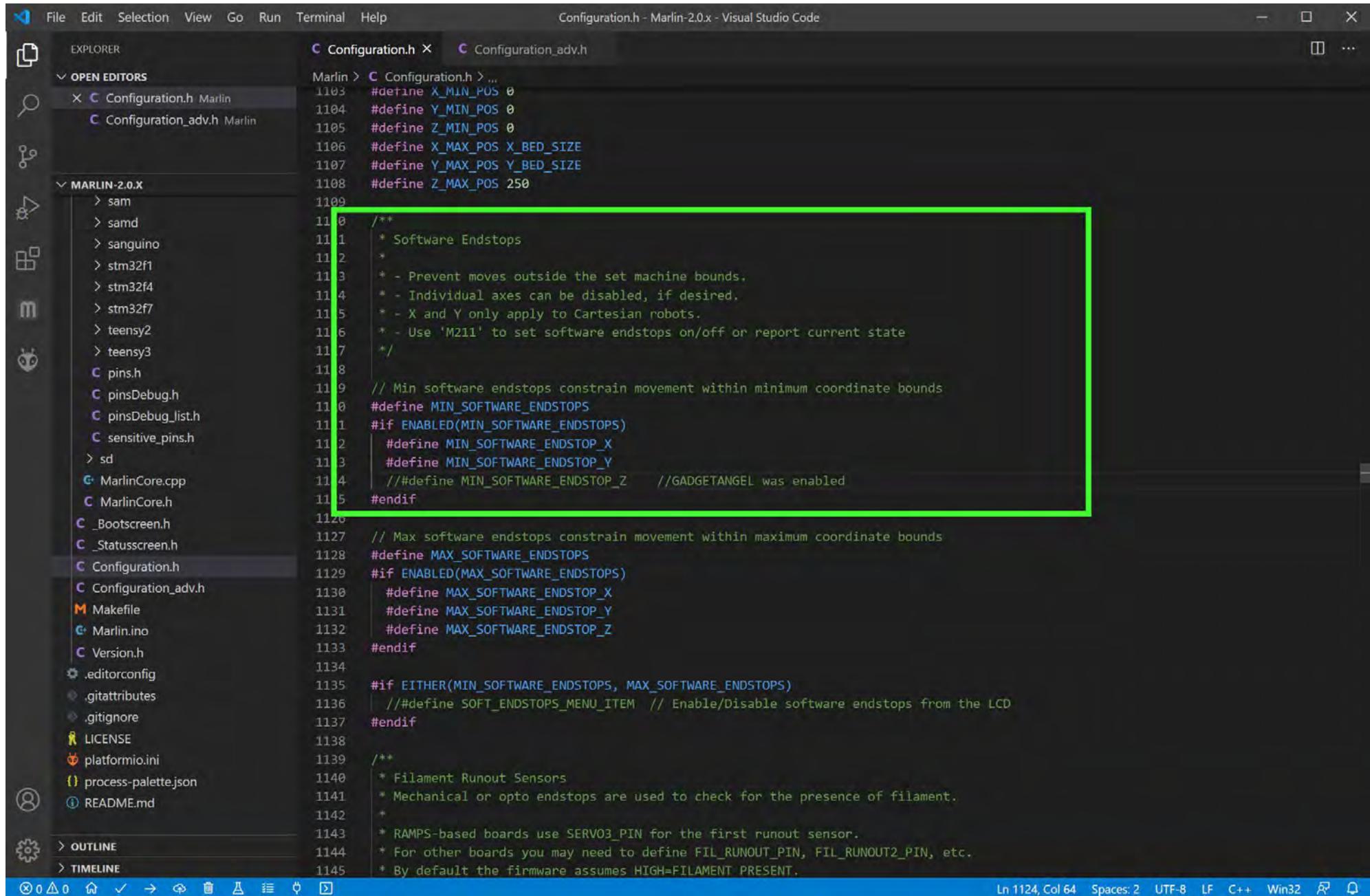
```

Ln 1005, Col 97 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- **disable** the Marlin variable **MIN\_SOFTWARE\_ENDSTOP\_Z** or that you comment out the line by inserting two forward slashes ('//') at the beginning of the line.



```

File Edit Selection View Go Run Terminal Help Configuration.h - Marlin-2.0.x - Visual Studio Code
EXPLORER OPEN EDITORS MARLIN-2.0.X Configuration.h Configuration_adv.h
Marlin > Configuration.h > ...
1103 #define X_MIN_POS 0
1104 #define Y_MIN_POS 0
1105 #define Z_MIN_POS 0
1106 #define X_MAX_POS X_BED_SIZE
1107 #define Y_MAX_POS Y_BED_SIZE
1108 #define Z_MAX_POS 250
1109
1110 /**
1111 * Software Endstops
1112 *
1113 * - Prevent moves outside the set machine bounds.
1114 * - Individual axes can be disabled, if desired.
1115 * - X and Y only apply to Cartesian robots.
1116 * - Use 'M211' to set software endstops on/off or report current state
1117 */
1118 // Min software endstops constrain movement within minimum coordinate bounds
1119 #define MIN_SOFTWARE_ENDSTOPS
1120 #if ENABLED(MIN_SOFTWARE_ENDSTOPS)
1121 #define MIN_SOFTWARE_ENDSTOP_X
1122 #define MIN_SOFTWARE_ENDSTOP_Y
1123 //##define MIN_SOFTWARE_ENDSTOP_Z //GADGETANGEL was enabled
1124 #endif
1125
1126 // Max software endstops constrain movement within maximum coordinate bounds
1127 #define MAX_SOFTWARE_ENDSTOPS
1128 #if ENABLED(MAX_SOFTWARE_ENDSTOPS)
1129 #define MAX_SOFTWARE_ENDSTOP_X
1130 #define MAX_SOFTWARE_ENDSTOP_Y
1131 #define MAX_SOFTWARE_ENDSTOP_Z
1132 #endif
1133
1134 #if EITHER(MIN_SOFTWARE_ENDSTOPS, MAX_SOFTWARE_ENDSTOPS)
1135 //##define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops from the LCD
1136 #endif
1137
1138 /**
1139 * Filament Runout Sensors
1140 * Mechanical or opto endstops are used to check for the presence of filament.
1141 *
1142 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
1143 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN, etc.
1144 * By default the firmware assumes HIGH=FILAMENT_PRESENT.
1145

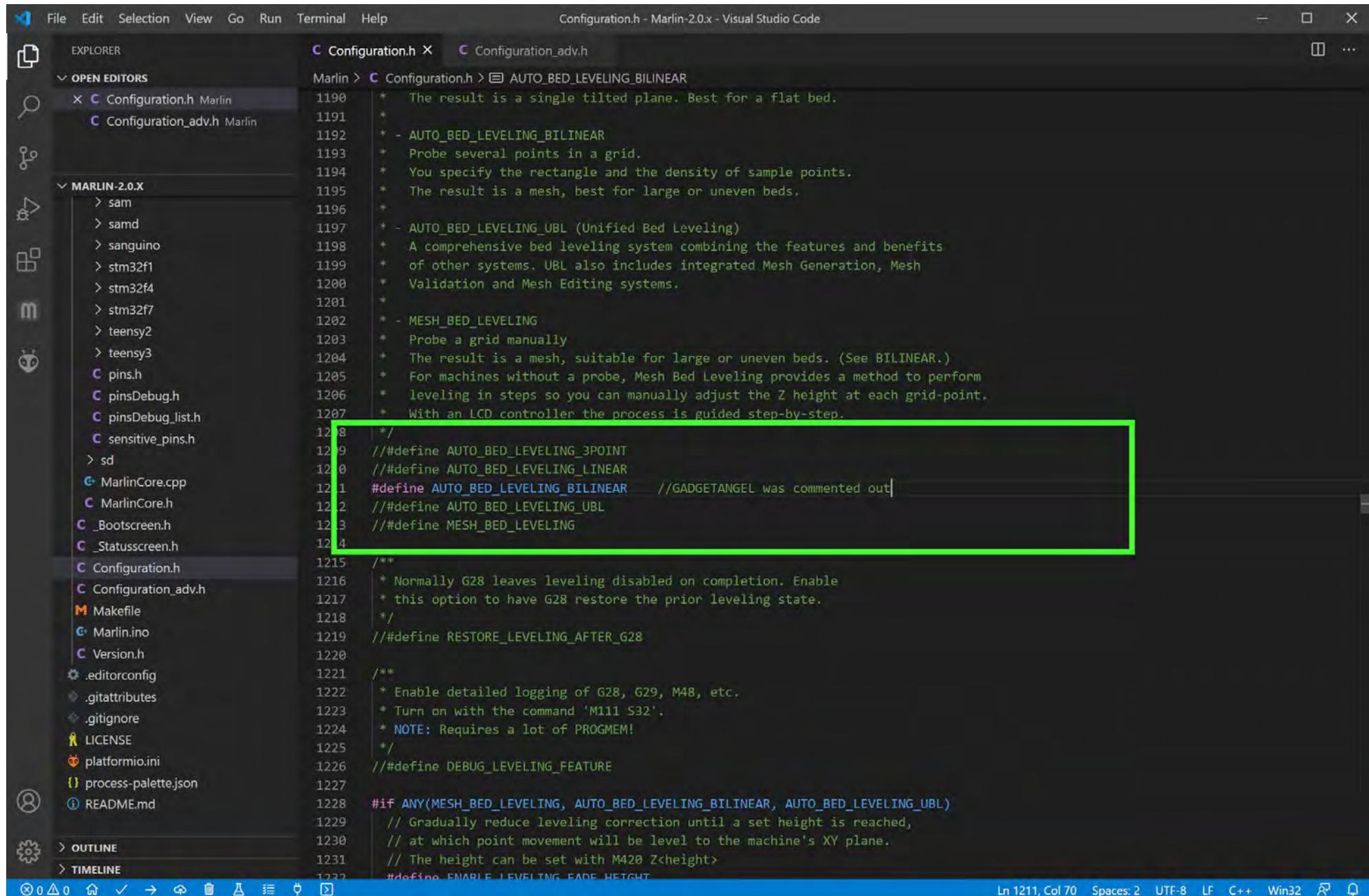
```

Ln 1124, Col 64 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- enable the Marlin variable **AUTO\_BED\_LEVELING\_BILINEAR** or that you **delete the two forward slashes ("//") from the beginning of the line.**



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x firmware's bed leveling configuration. A green rectangular box highlights the following line of code:

```
//#define AUTO_BED_LEVELING_BILINEAR //GADGETANGEL was commented out
```

This line is part of a larger block of code related to bed leveling methods. The code includes comments explaining the different methods: BILINEAR, UBL, and MESH\_BED\_LEVELING. The entire block is enclosed in a multi-line comment block starting at line 1215.

```

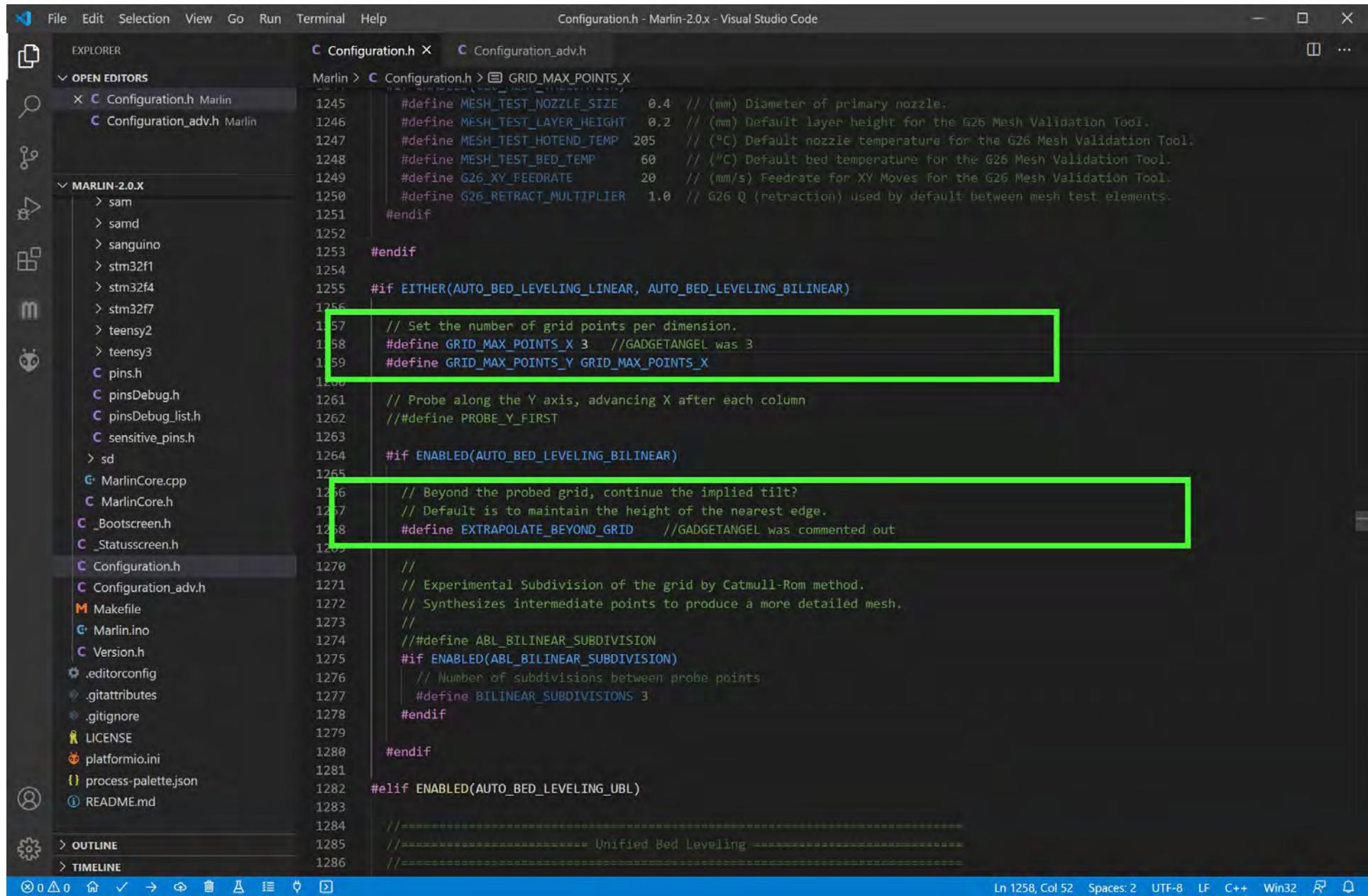
1190 * The result is a single tilted plane. Best for a flat bed.
1191 *
1192 * - AUTO_BED_LEVELING_BILINEAR
1193 * Probe several points in a grid.
1194 * You specify the rectangle and the density of sample points.
1195 * The result is a mesh, best for large or uneven beds.
1196 *
1197 * - AUTO_BED_LEVELING_UBL (Unified Bed Leveling)
1198 * A comprehensive bed leveling system combining the features and benefits
1199 * of other systems. UBL also includes integrated Mesh Generation, Mesh
1200 * Validation and Mesh Editing systems.
1201 *
1202 * - MESH_BED_LEVELING
1203 * Probe a grid manually
1204 * The result is a mesh, suitable for large or uneven beds. (See BILINEAR.)
1205 * For machines without a probe, Mesh Bed Leveling provides a method to perform
1206 * leveling in steps so you can manually adjust the Z height at each grid-point.
1207 * With an LCD controller the process is guided step-by-step.
1208 */
1209 //#define AUTO_BED_LEVELING_3POINT
1210 //#define AUTO_BED_LEVELING_LINEAR
1211 #define AUTO_BED_LEVELING_BILINEAR //GADGETANGEL was commented out
1212 //#define AUTO_BED_LEVELING_UBL
1213 //#define MESH_BED_LEVELING
1214 */
1215 /**
1216 * Normally G28 leaves leveling disabled on completion. Enable
1217 * this option to have G28 restore the prior leveling state.
1218 */
1219 //#define RESTORE_LEVELING_AFTER_G28
1220 */
1221 /**
1222 * Enable detailed logging of G28, G29, M48, etc.
1223 * Turn on with the command 'M111 S32'.
1224 * NOTE: Requires a lot of PROGMEM!
1225 */
1226 //#define DEBUG_LEVELING_FEATURE
1227 */
1228 #if ANY(MESH_BED_LEVELING, AUTO_BED_LEVELING_BILINEAR, AUTO_BED_LEVELING_UBL)
1229 // Gradually reduce leveling correction until a set height is reached,
1230 // at which point movement will be level to the machine's XY plane.
1231 // The height can be set with M420 Z<height>
1232 #define ENABLE_LEVELING_FADE_HEIGHT

```

- Go to the next page.

**APPENDIX J****Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- ensure the Marlin variable **GRID\_MAX\_POINTS\_X** is set to 3
- enable the Marlin variable **EXTRAPOLATE\_BEYOND\_GRID** or that you **delete the two forward slashes ("//") from the beginning of the line.**



The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the Marlin firmware configuration. Two specific sections of the code are highlighted with green boxes:

```

1245 #define MESH_TEST_NOZZLE_SIZE 0.4 // (mm) Diameter of primary nozzle.
1246 #define MESH_TEST_LAYER_HEIGHT 0.2 // (mm) Default layer height for the G26 Mesh Validation Tool.
1247 #define MESH_TEST_HOTEND_TEMP 205 // (°C) Default nozzle temperature for the G26 Mesh Validation Tool.
1248 #define MESH_TEST_BED_TEMP 60 // (°C) Default bed temperature for the G26 Mesh Validation Tool.
1249 #define G26_XY_FEEDRATE 20 // (mm/s) Feedrate for XY Moves for the G26 Mesh Validation Tool.
1250 #define G26_RETRACT_MULTIPLIER 1.0 // G26 Q (retraction) used by default between mesh test elements.
1251 #endif
1252
1253 #endif
1254
1255 #if EITHER(AUTO_BED_LEVELING_LINEAR, AUTO_BED_LEVELING_BILINEAR)
1256
1257 // Set the number of grid points per dimension.
1258 #define GRID_MAX_POINTS_X 3 //GADGETANGEL was 3
1259 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
1260
1261 // Probe along the Y axis, advancing X after each column
1262 // #define PROBE_Y_FIRST
1263
1264 #if ENABLED(AUTO_BED_LEVELING_BILINEAR)
1265
1266 // Beyond the probed grid, continue the implied tilt?
1267 // Default is to maintain the height of the nearest edge.
1268 #define EXTRAPOLATE_BEYOND_GRID //GADGETANGEL was commented out
1269
1270
1271 // Experimental Subdivision of the grid by Catmull-Rom method.
1272 // Synthesizes intermediate points to produce a more detailed mesh.
1273 //
1274 // #define ABL_BILINEAR_SUBDIVISION
1275 #if ENABLED(ABL_BILINEAR_SUBDIVISION)
1276 // Number of subdivisions between probe points
1277 #define BILINEAR_SUBDIVISIONS 3
1278 #endif
1279
1280 #endif
1281
1282 #elif ENABLED(AUTO_BED_LEVELING_UBL)
1283
1284 //=====
1285 //===== Unified Bed Leveling =====
1286 //=====

```

The first highlighted section contains the definition of `GRID_MAX_POINTS_X` as 3. The second highlighted section contains the definition of `EXTRAPOLATE_BEYOND_GRID` as `//GADGETANGEL was commented out`.

- Go to the next page.

## **APPENDIX** **Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- **enable** the Marlin variable **LCD\_BED\_LEVELING** or that you **delete the two forward slashes ("//")** from the beginning of the line.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Configuration.h - Marlin-2.0.x - Visual Studio Code.
- Left Sidebar:** EXPLORER, OPEN EDITORS, MARLIN-2.0.X (containing sam, samd, sanguino, stm32f1, stm32f4, stm32f7, teensy2, teensy3, pins.h, pinsDebug.h, pinsDebug.list.h, sensitive\_pins.h, sd, MarlinCore.cpp, MarlinCore.h, \_Bootscreen.h, \_Statusscreen.h, Configuration.h, Configuration\_adv.h, Makefile, Marlin.ino, Version.h, .editorconfig, .gitattributes, .gitignore, LICENSE, platformio.ini, process-palette.json, README.md).
- Central Area:** Editor showing Configuration.h with the following code snippet highlighted by a green box:

```
#define UBL_SAVE_ACTIVE_ON_M500 // Save the currently active mesh in the current slot on M500
//#define UBL_Z_RAISE_WHEN_OFF_MESH 2.5 // When the nozzle is off the mesh, this value is used
// as the Z-Height correction value.

#elif ENABLED(MESH_BED_LEVELING)
//-----
//----- Mesh -----
//-----

#define MESH_INSET 10 // Set Mesh bounds as an inset region of the bed
#define GRID_MAX_POINTS_X 3 // Don't use more than 7 points per axis, implementation limited.
#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

//#define MESH_G28_REST_ORIGIN // After homing all axes ('G28' or 'G28 XYZ') rest Z at Z_MIN_POS

#endif // BED_LEVELING

/*
 * Add a bed leveling sub-menu for ABL or MBL.
 * Include a guided procedure if manual probing is enabled.
 */
#define LCD_BED_LEVELING //GADGETANGEL was commented out

#if ENABLED(LCD_BED_LEVELING)
#define MESH_EDIT_Z_STEP 0.025 // (mm) Step size while manually probing Z axis.
#define LCD_PROBE_Z_RANGE 4 // (mm) Z Range centered on Z_MIN_POS for LCD Z adjustment
//#define MESH_EDIT_MENU // Add a menu to edit mesh points
#endif

// Add a menu item to move between bed corners for manual bed adjustment
//#define LEVEL_BED_CORNERS

#if ENABLED(LEVEL_BED_CORNERS)
#define LEVEL_CORNERS_INSET_LFRB { 30, 30, 30, 30 } // (mm) Left, Front, Right, Back insets
#define LEVEL_CORNERS_HEIGHT 0.0 // (mm) Z height of nozzle at leveling points
#define LEVEL_CORNERS_Z_HOP 4.0 // (mm) Z height of nozzle between leveling points
//#define LEVEL_CENTER_TOO // Move to the center after the last corner
#endif
```
- Bottom Status Bar:** Ln 1318, Col 60, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- enable the Marlin variable **Z\_SAFE\_HOMING** or that you delete the two forward slashes ("//") from the beginning of the line.
- set the Marlin variable **HOMING\_FEEDRATE\_XY** to **50\*60**
- set the Marlin variable **HOMING\_FEEDRATE\_Z** to **20\*60**

The screenshot shows the Visual Studio Code interface with the file `Configuration.h` open. The code editor displays the Marlin 2.0.x firmware configuration. Two specific sections of code are highlighted with green boxes:

```

1350 // #define MANUAL_X_HOME_POS 0
1351 // #define MANUAL_Y_HOME_POS 0
1352 // #define MANUAL_Z_HOME_POS 0
1353
1354 // Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
1355 //
1356 // With this feature enabled:
1357 //
1358 // - Allow Z homing only after X and Y homing AND stepper drivers still enabled.
1359 // - If stepper drivers time out, it will need X and Y homing again before Z homing.
1360 // - Move the Z probe (or nozzle) to a defined XY point before Z Homing when homing all axes (G28).
1361 // - Prevent Z homing when the Z probe is outside bed area.
1362 //
1363 #define Z_SAFE_HOMING //GADGETANGEL was commented out
1364
1365 #if ENABLED(Z_SAFE_HOMING)
1366 #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2) // X point for Z homing when homing all axes (G28).
1367 #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2) // Y point for Z homing when homing all axes (G28).
1368 #endif
1369
1370 // Homing speeds (mm/m)
1371 #define HOMING_FEEDRATE_XY (50*60) //GADGETANGEL was 20*60
1372 #define HOMING_FEEDRATE_Z (20*60) //GADGETANGEL was 4*60
1373
1374 // Validate that endstops are triggered on homing moves
1375 #define VALIDATE_HOMING_ENDSTOPS
1376
1377 // @section calibrate
1378
1379 /**
1380 * Bed Skew Compensation
1381 *
1382 * This feature corrects for misalignment in the XYZ axes.
1383 *
1384 * Take the following steps to get the bed skew in the XY plane:
1385 * 1. Print a test square (e.g., https://www.thingiverse.com/thing:2563185)
1386 * 2. For XY_DIAG_AC measure the diagonal A to C
1387 * 3. For XY_DIAG_BD measure the diagonal B to D
1388 * 4. For XY_SIDE_AD measure the edge A to D
1389 *
1390 * Marlin automatically computes skew factors from these measurements.
1391 * Skew factors may also be computed and set manually:

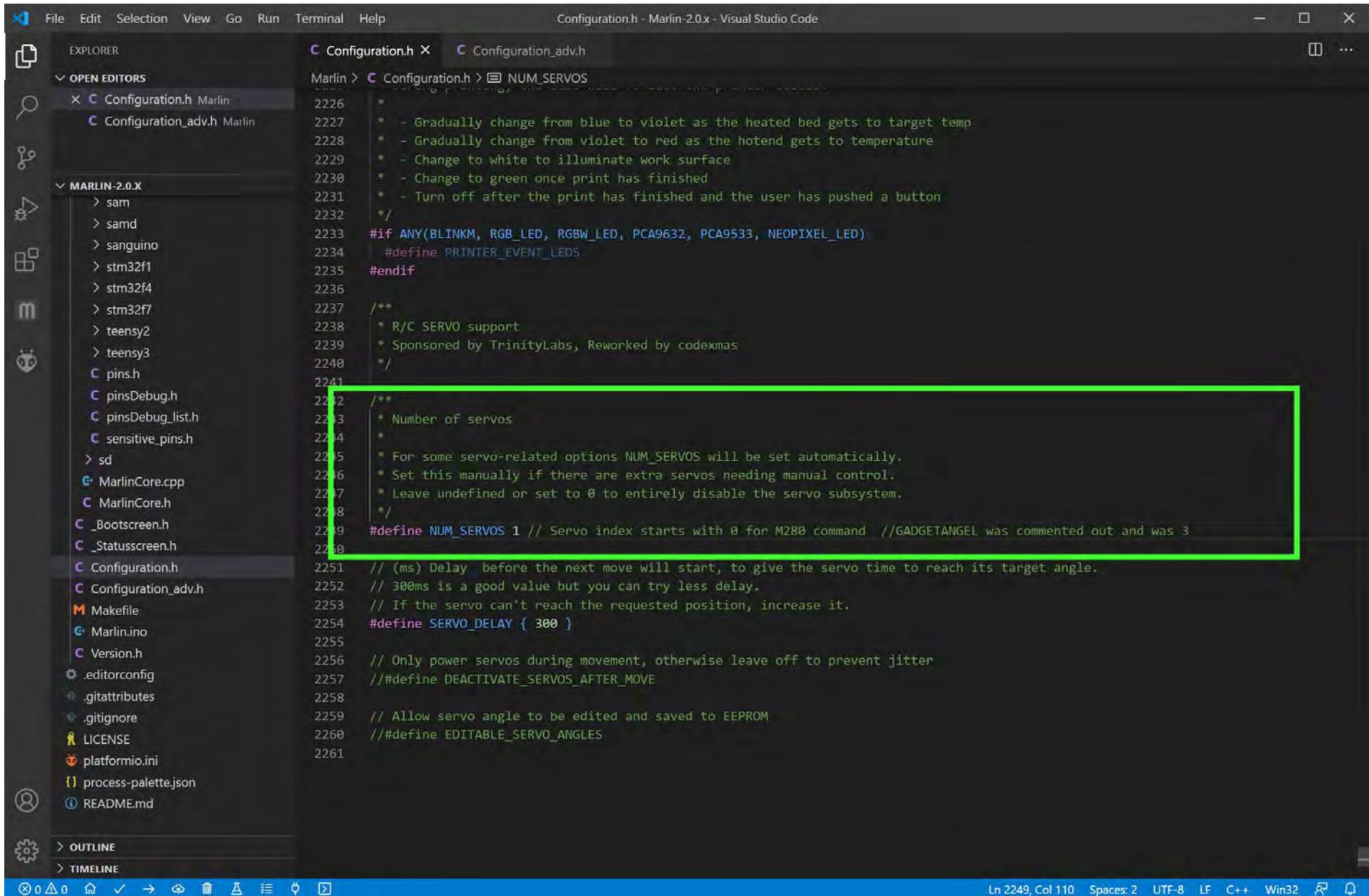
```

The first green box highlights the configuration for `Z_SAFE_HOMING`, which is currently commented out. The second green box highlights the definitions for `HOMING_FEEDRATE_XY` and `HOMING_FEEDRATE_Z`.

- Go to the next page.

**APPENDIX J****Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- **enable the Marlin variable `NUM_SERVOS` or that you delete the two forward slashes ("//") from the beginning of the line.**
- **change the Marlin variable `NUM_SERVOS` to 1**



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following snippet:

```

2226 *
2227 * - Gradually change from blue to violet as the heated bed gets to target temp
2228 * - Gradually change from violet to red as the hotend gets to temperature
2229 * - Change to white to illuminate work surface
2230 * - Change to green once print has finished
2231 * - Turn off after the print has finished and the user has pushed a button
2232 */
2233 #if ANY(BLINKM, RGB_LED, RGBW_LED, PCA9632, PCA9533, NEOPIXEL_LED)
2234 #define PRINTER_EVENT_LEDS
2235 #endif
2236 /**
2237 * R/C SERVO support
2238 * Sponsored by TrinityLabs, Reworked by codexmas
2239 */
2240 /**
2241 /**
2242 * Number of servos
2243 *
2244 * For some servo-related options NUM_SERVOS will be set automatically.
2245 * Set this manually if there are extra servos needing manual control.
2246 * Leave undefined or set to 0 to entirely disable the servo subsystem.
2247 */
2248 #define NUM_SERVOS 1 // Servo index starts with 0 for M280 command //GADGETANGEL was commented out and was 3
2249 /**
2250 * (ms) Delay before the next move will start, to give the servo time to reach its target angle.
2251 // 300ms is a good value but you can try less delay.
2252 // If the servo can't reach the requested position, increase it.
2253 /**
2254 #define SERVO_DELAY { 300 }
2255 /**
2256 * Only power servos during movement, otherwise leave off to prevent jitter
2257 // #define DEACTIVATE_SERVOS_AFTER_MOVE
2258 /**
2259 * Allow servo angle to be edited and saved to EEPROM
2260 // #define EDITABLE_SERVO_ANGLES
2261

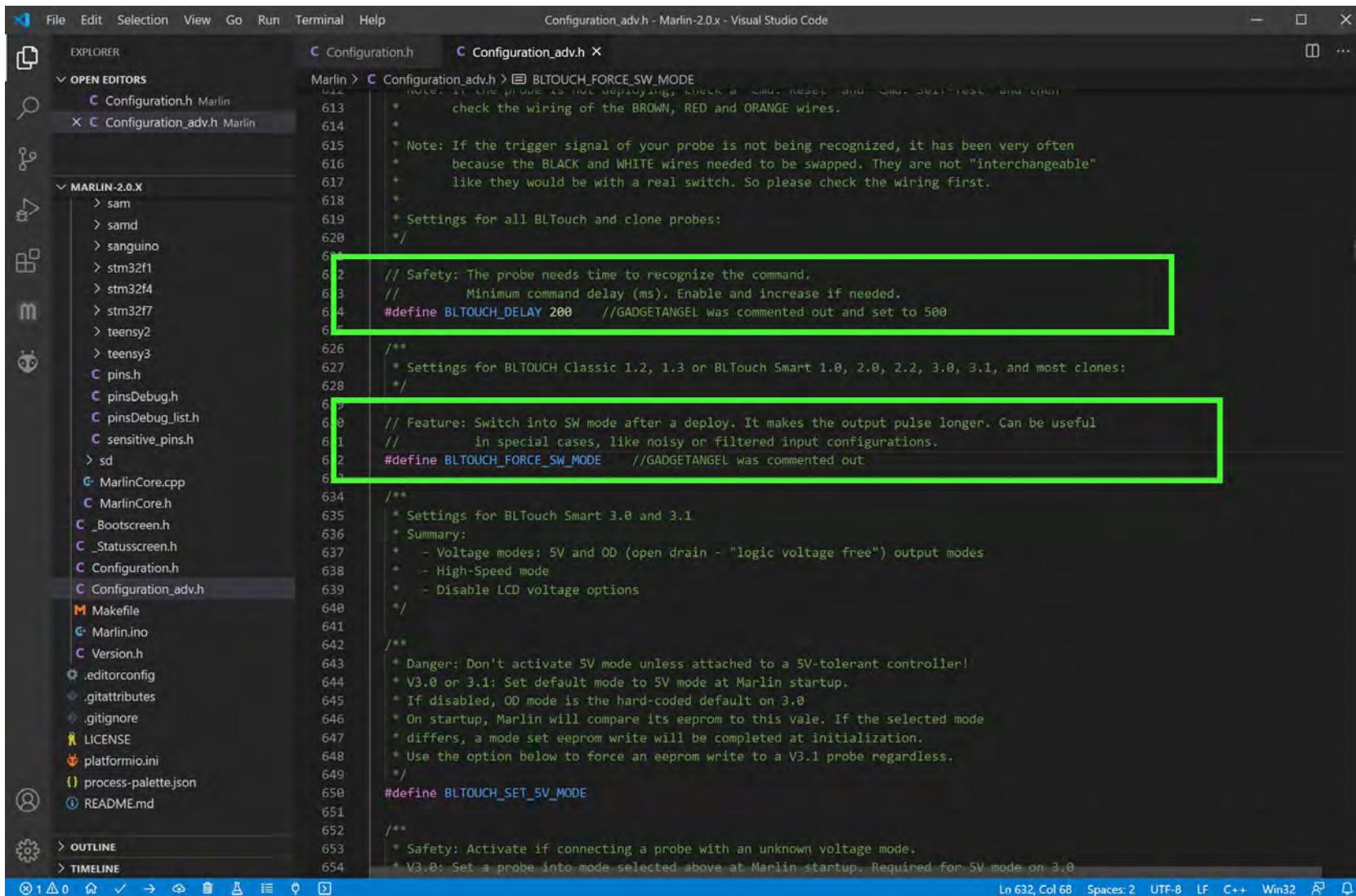
```

A green rectangular box highlights the line `#define NUM_SERVOS 1 // Servo index starts with 0 for M280 command //GADGETANGEL was commented out and was 3`. The status bar at the bottom right of the code editor shows: Ln 2249, Col 110 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board

- All the following changes are made to the Configuration\_adv.h file
- enable the Marlin variable **BLTOUCH\_DELAY** or that you delete the two forward slashes ("//") from the beginning of the line.
- change the Marlin variable **BLTOUCH\_DELAY** to 200
- enable the Marlin variable **BLTOUCH\_FORCE\_SW\_MODE** or that you delete the two forward slashes ("//") from the beginning of the line.



The screenshot shows the Visual Studio Code interface with the file "Configuration\_adv.h" open. The code editor displays the Marlin 2.0.x firmware configuration. Two specific sections of the code are highlighted with green boxes:

```

613 * check the wiring of the BROWN, RED and ORANGE wires.
614 *
615 * Note: If the trigger signal of your probe is not being recognized, it has been very often
616 * because the BLACK and WHITE wires needed to be swapped. They are not "interchangeable"
617 * like they would be with a real switch. So please check the wiring first.
618 *
619 * Settings for all BLTouch and clone probes:
620 */
621 // Safety: The probe needs time to recognize the command.
622 // Minimum command delay (ms). Enable and increase if needed.
623 #define BLTOUCH_DELAY 200 //GADGETANGEL was commented out and set to 500
624 /**
625 * Settings for BLTOUCH Classic 1.2, 1.3 or BLTouch Smart 1.0, 2.0, 2.2, 3.0, 3.1, and most clones:
626 */
627 /**
628 * Feature: Switch into SW mode after a deploy. It makes the output pulse longer. Can be useful
629 * in special cases, like noisy or filtered input configurations.
630 #define BLTOUCH_FORCE_SW_MODE //GADGETANGEL was commented out
631 */
632 /**
633 * Settings for BLTouch Smart 3.0 and 3.1
634 * Summary:
635 * - Voltage modes: 5V and OD (open drain - "logic voltage free") output modes
636 * - High-Speed mode
637 * - Disable LCD voltage options
638 */
639 /**
640 * Danger: Don't activate 5V mode unless attached to a 5V-tolerant controller!
641 * V3.0 or 3.1: Set default mode to 5V mode at Marlin startup.
642 * If disabled, OD mode is the hard-coded default on 3.0
643 * On startup, Marlin will compare its eeprom to this value. If the selected mode
644 * differs, a mode set eeprom write will be completed at initialization.
645 * Use the option below to force an eeprom write to a V3.1 probe regardless.
646 */
647 #define BLTOUCH_SET_5V_MODE
648 /**
649 * Safety: Activate if connecting a probe with an unknown voltage mode.
650 * V3.0: Set a probe into mode selected above at Marlin startup. Required for 5V mode on 3.0
651 */
652 /**
653 * Safety: Activate if connecting a probe with an unknown voltage mode.
654 * V3.0: Set a probe into mode selected above at Marlin startup. Required for 5V mode on 3.0
655 */

```

The code editor's status bar at the bottom shows: Ln 632, Col 68, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

## APPENDIX J

**Marlin 2.0.x Firmware Setup for Connecting a BLTouch to the SKR V1.3 Board**

- change the Marlin variable **BABYSTEP\_MULTIPLICATOR\_Z** to **10**
- ensure the Marlin variable **DOUBLECLICK\_FOR\_Z\_BABYSTEPPING** is **enabled** or that you **delete the two forward slashes ("//") from the beginning of the line.**
- **enable** the Marlin variable **BABYSTEP\_ZPROBE\_OFFSET** or that you **delete the two forward slashes ("//") from the beginning of the line.**

```

File Edit Selection View Go Run Terminal Help
Configuration_adv.h - Marlin-2.0.x - Visual Studio Code
EXPLORER C Configuration.h C Configuration_adv.h X
Marlin > C Configuration_adv.h > BABYSTEP_ZPROBE_OFFSET
1462 * Babystepping enables movement of the axes by tiny increments without changing
1463 * the current position values. This feature is used primarily to adjust the Z
1464 * axis in the first layer of a print in real-time.
1465 *
1466 * Warning: Does not respect endstops!
1467 */
1468 #define BABYSTEPPING
1469 #if ENABLED(BABYSTEPPING)
1470 //##define INTEGRATED_BABYSTEPPING // EXPERIMENTAL integration of babystepping into the Stepper ISR
1471 //##define BABYSTEP_WITHOUT_HOMING
1472 //##define BABYSTEP_XY // Also enable X/Y Babystepping. Not supported on DELTA!
1473 #define BABYSTEP_INVERT_Z false // Change if Z babysteps should go the other way
1474 #define BABYSTEP_MULTIPLICATOR_Z 10 // Babysteps are very small. Increase for faster motion. //GADGETANGEL was 1
1475 #define BABYSTEP_MULTIPLICATOR_XY 1
1476
1477 #define DOUBLECLICK_FOR_Z_BABYSTEPPING // Double-click on the Status Screen for Z Babystepping.
1478 #if ENABLED(DOUBLECLICK_FOR_Z_BABYSTEPPING)
1479 #define DOUBLECLICK_MAX_INTERVAL 1250 // Maximum interval between clicks, in milliseconds.
1480 //##define BABYSTEP_ALWAYS_AVAILABLE // Note: Extra time may be added to mitigate controller latency.
1481 //##define MOVE_Z_WHEN_IDLE // Allow babystepping at all times (not just during movement).
1482 #if ENABLED(MOVE_Z_WHEN_IDLE)
1483 #define MOVE_Z_IDLE_MULTIPLICATOR 1 // Jump to the move Z menu on doubleclick when printer is idle.
1484 #endif
1485 #endif
1486 #endif
1487
1488 //##define BABYSTEP_DISPLAY_TOTAL // Display total babysteps since last G28
1489
1490 #define BABYSTEP_ZPROBE_OFFSET // Combine M851 Z and Babystepping //GADGETANGEL was commented out
1491 #if ENABLED(BABYSTEP_ZPROBE_OFFSET)
1492 //##define BABYSTEP_HOTEND_Z_OFFSET // For multiple hotends, babystep relative Z offsets
1493 //##define BABYSTEP_ZPROBE_GFX_OVERLAY // Enable graphical overlay on Z-offset editor
1494 #endif
1495 #endif
1496
1497 // @section extruder
1498 /**
1499 * Linear Pressure Control v1.5
1500 *
1501 * Assumption: advance [steps] = k * (delta velocity [steps/s])
1502 * K=0 means advance disabled.
1503 */

```

Ln 1490, Col 112 Spaces: 2 UTF-8 LF C++ Win32

- Go to the next page.

## APPENDIX J

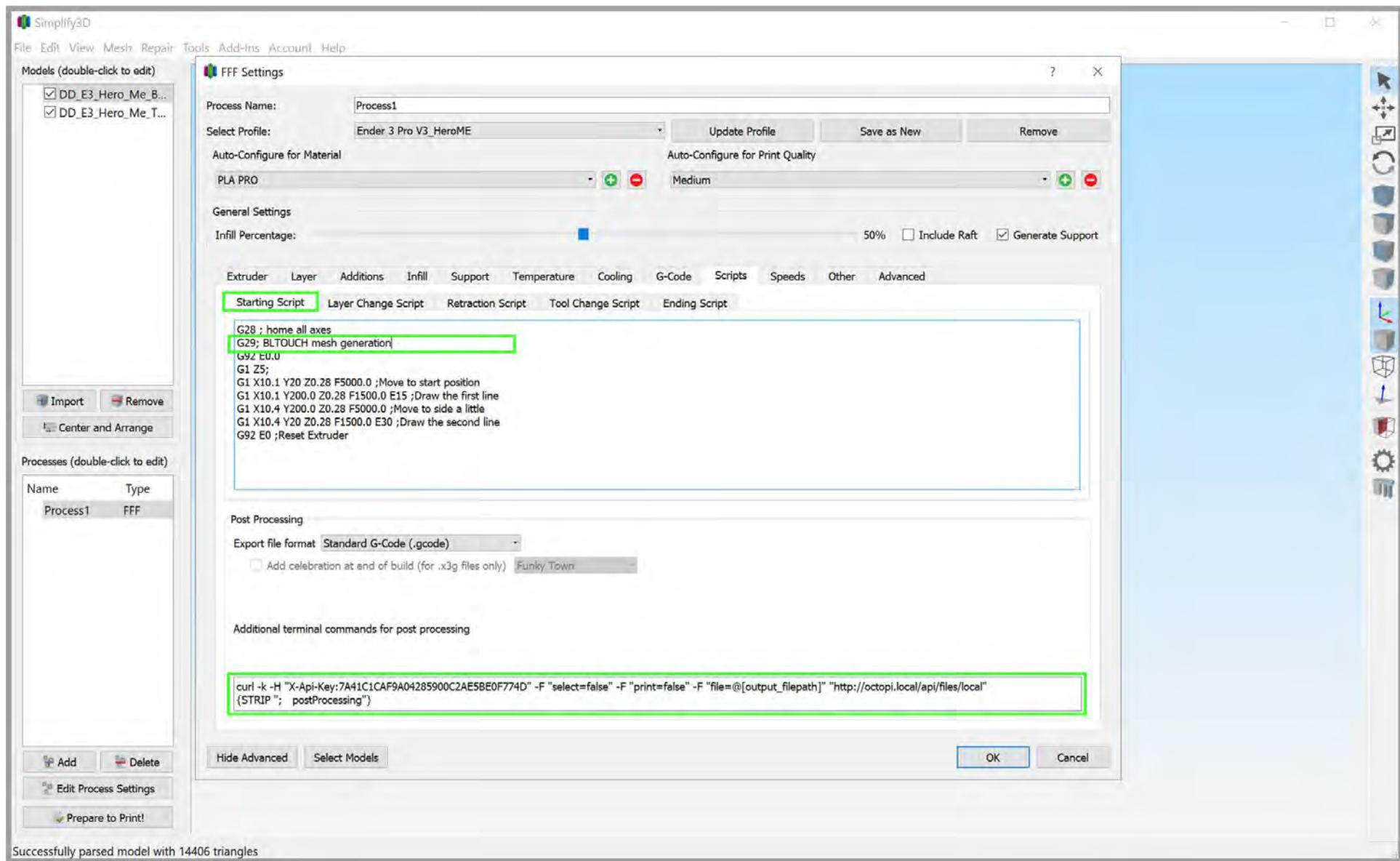
### Testing the BLTouch Setup

- Now that the BLTouch is wired to the SKR V1.3 board and the Marlin Firmware is setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.
  - we need to set the BLTouch Z probe offset on the 3D Printer.
  - Raise the 3D Printer's Z Axis up high enough so that you can get your hand under the BLTouch Probe.
  - Try to **HOME** the 3D Printer from the Printer's LCD screen.
  - As the Z Axis drops place your hand under the BLTouch so that the BLTouch Probe will hit your hand. If your 3D Printer's Z Axis raises up after the BLTouch Probe hits your hand than the BLTouch is working properly. If the Z Axis still drops, after the BLTouch hits your hand, then turn the power off your 3D printer to stop the Z Axis from hitting the bed. To correct the problem, double check your wiring of the BLTouch to the SKR V1.3 board.
  - If the BLTouch Z Axis rises after hitting your hand, then your BLTouch is wired up to the SKR V1.3 board correctly.
  - Connect to the Printer via Pronterface. Send a **G28** command to HOME all axes. After the Printer finishes homing all axes. Send a **G29** command to tell the 3D Printer to probe the bed for ABL (automatic bed leveling). Does your printer probe the bed? If so it's time to set the **Z OFFSET**.
  - Before setting the **Z OFFSET**, let's change our SLICER settings to add the **G29** command to the start G-Code script in your SLICER Settings. I use Simply3D Slicer.
  - see the next page for my Simply3D Slicer setting for the start G-Code script.
- 
- Go to the next page.

## APPENDIX J

# Testing the BLTouch Setup

- change the starting G-Code script for your slicer. You need to find the appropriate settings page in your slicer that is equivalent to the below picture, which shows the Starting G-Code script from Simplify3D.
- Add **G29;** to the starting G-Code script to your slicer. It is placed right after the **G28;** command.
- The line located in the box labeled "Additional terminal commands for post processing" is a command to allow me to send files to my 3D printer via Octoprint. This information is optional. This command will cause the generated .stl file from the slicer to be placed in a Octoprint watched folder on the Raspberry Pi.



- Go to the next page.

## APPENDIX J

### Testing the BLTouch Setup

- With your SLICER setup to do a **G-29** command automatically at the start of a 3D print job, we need a 3D object that is big but only 1 layer thick. On Thingiverse there is an object called "Bedlevel test cross (For ABL, BLTouch, 3DTouch etc)" by mdieli. You can find it here at this URL: <https://www.thingiverse.com/thing:3846283>
- Import the object into your slicer. Use a layer thickness of 0.2mm. And send the slicer generated file to your 3D Printer to print the object. Your 3D Printer should perform an ABL (Automatic Bed Leveling) Bed Probe before it starts printing the object.
- While the object is printing, go to the **MOTION** menu on your 3D Printer's LCD display and change the **Z OFFSET**. Turn the knob to the LEFT will place the Nozzle closer to the printer bed. Turn the knob to the RIGHT will raise the Nozzle away from the printer bed.
- Adjust the **Z OFFSET** with the knob until you are satisfied with your first layer.
- Save your **Z OFFSET** value by going to your 3D Printer's **CONTROL** menu and selecting "**Store Settings**" to save the **Z OFFSET** to the EEPROM of the 3D Printer. Once saved to EEPROM, the **Z OFFSET** will be saved in between reboots of your 3D Printer.

- Go to the next page.

## APPENDIX J

### Testing the BLTouch Setup

- keep the Marlin settings the same (as above) until the BLTouch is working. Afterwards you are free to tweak away.
- All Finished with setting up the BLTouch!!

## APPENDIX K

### Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

To install a PT100 temperature sensor to your 3D printer requires four steps:

- Decide which of the two techniques you will use to connect the PT100 amplifier board to the MCU board (SKR V1.3 board).
- Wire up the PT100 sensor to PT100 amplifier board.
- Wire up the PT100 amplifier board to the MCU board (SKR V1.3 board). If you chose to connect up the PT100 with **Technique #1** then you **MUST** use **Method #1 (Digital PWR)** to power your amplifier board. If you chose to connect up the PT100 with **Technique #2** than you have two methods to select from to power your amplifier board: **Method #1 (Digital PWR)** or **Method #2 (Analog PWR)**.
- Marlin 2.0.x Firmware setup.

There are two techniques you can use to wire up the PT100 amplifier board to the MCU:

- **Technique #1:** Attach the PT100 amplifier board to the TFT Port by using TX0 PIN. With this technique you lose the ability to use the TFT. The advantage is that you do not have to perform hardware hacks on the SKR V1.3 board. **ONLY** use **Method #1 (Digital PWR)** to power your amplifier board.
- **Technique #2<sup>\*1</sup>:** Attach the PT100 amplifier board to a thermistor port (TH0 or TH1). The disadvantage of this method is that you must perform hardware hacks to change the thermistor port (TH0) PIN to an analog input PIN. This requires removing a resistor and capacitor from the chosen thermistor port (TH0). The advantage of this method is that you **CAN** use the TFT because you will not be using TX0 PIN with this technique.

\*1: <https://www.thingiverse.com/thing:3603432>

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

**Technique #2<sup>\*1</sup> (continued):** If you chose to power the amplifier board by using **Method #2 (Analog PWR)** you get the advantage of noise immunity. Another advantage is that there are very few changes needed in Marlin 2.0.x Firmware. With **Technique #2** you have to decide which method you will use to power your amplifier board. There are two methods to select from: **Method #1 (Digital PWR)**, or **Method #2 (Analog PWR)**. **Method #2 is the preferred method** with this technique due to noise immunity.

**Technique #2 &Method #2 is the preferred and recommended option to connect your SKR V1.3 board up to a PT100 Sensor (with a PT100 amplifier board).**

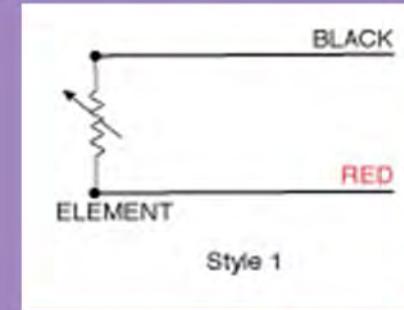
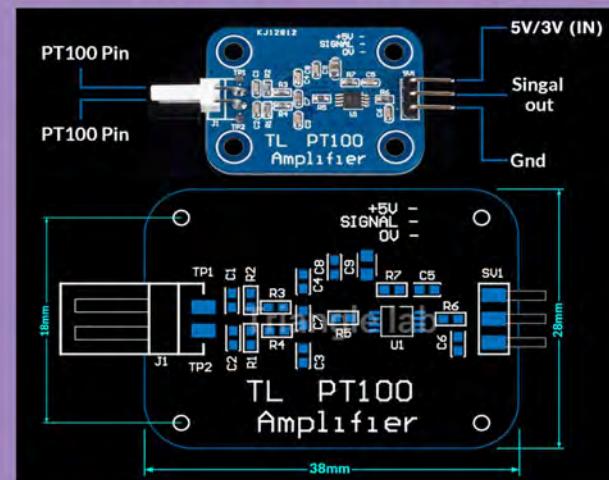
The diagram on the next page shows you the PT100 amplifier board attached to to a PT100 (RTD) temperature sensor.

It also indicates that the amplifier board will be using the ADC input on the MCU not SPI inputs. SPI inputs uses digital I/O lines while the ADC input uses an analog I/O line.

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### PT100 Sensor Hooked up to the PT100 Amplifier Board (ADC Input)



A PT100 probe is an assembly composed of an RTD element, a sheath, a lead wire, and a termination or connection. The PT100 is a temperature sensor that contains a resistor that changes resistance value as its temperature changes<sup>\*2</sup>.

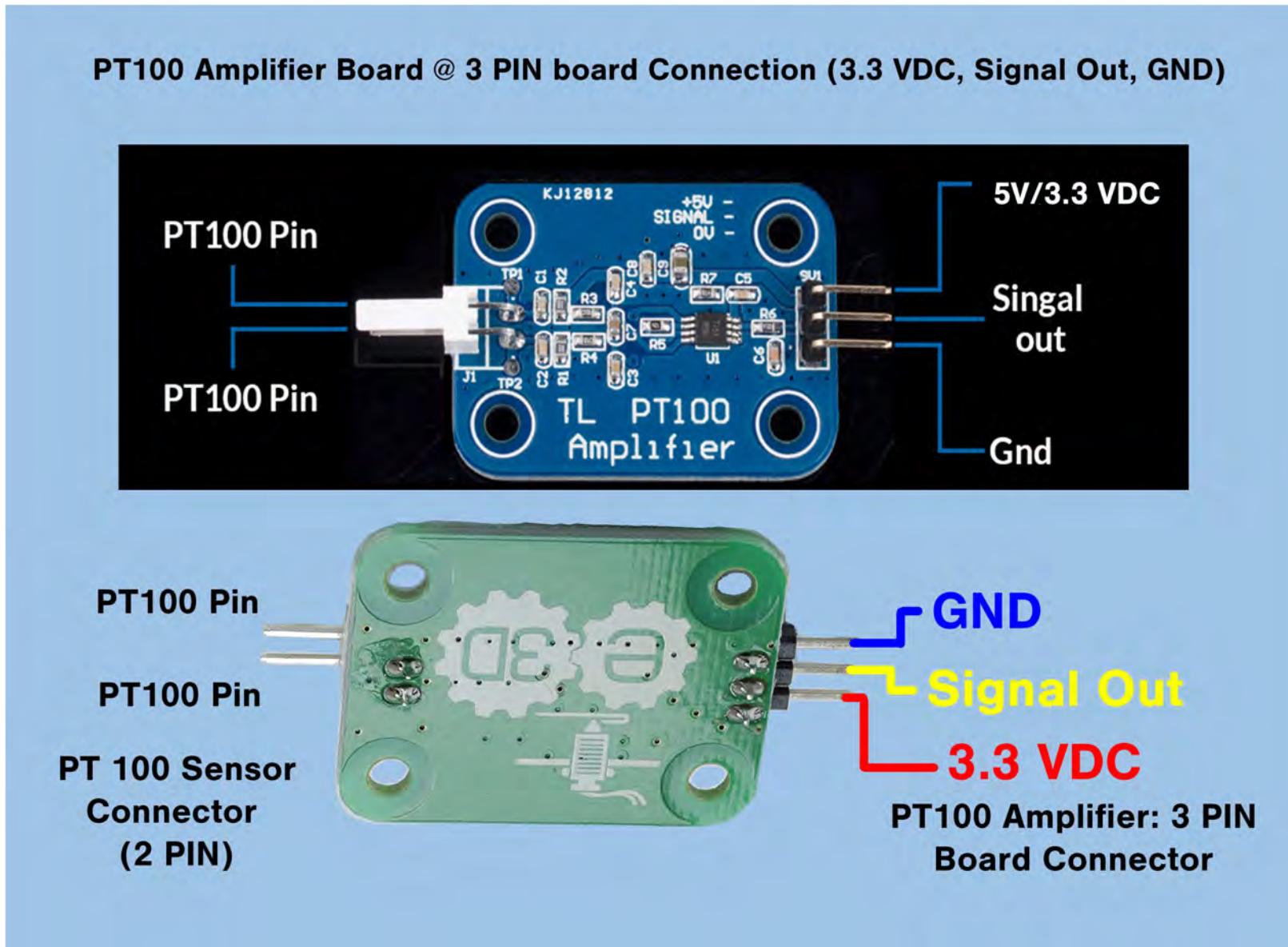
URL: for "Blog on RTD temperature sensors" at <https://blog.beamex.com/pt100-temperature-sensor>

\*2 URL: <https://www.omega.com/en-us/resources/rtd-resistance-temperature-detector-components>

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

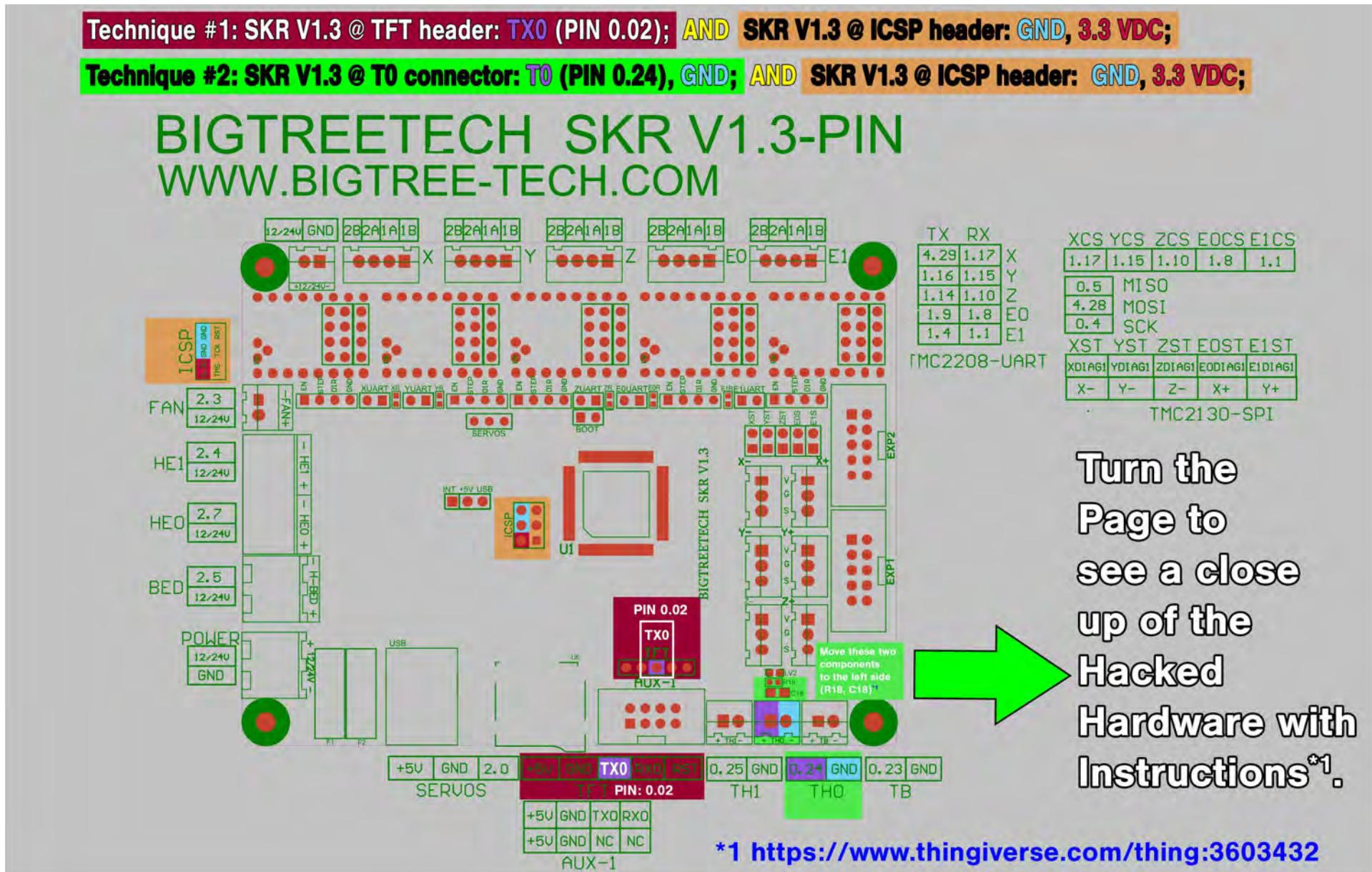
The diagram below shows the locations for **+5V/3.3VDC**, **GND**, and **Signal Out** (which is the ADC output signal that carries the information about the temperature). The diagram below also shows where the PT100 sensor pins are located. The board on top is a PT100 amplifier board made by Trianglelab while the board on the bottom of the picture is an amplifier board made by E3D. Each board's silk screen (labels) show the location for each OUTPUT PIN.



- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the PT100 temperature sensor with the PT100 amplifier board. This is an overview. The diagram indicates which PINS are involved in **Technique #1** and **Technique #2**.



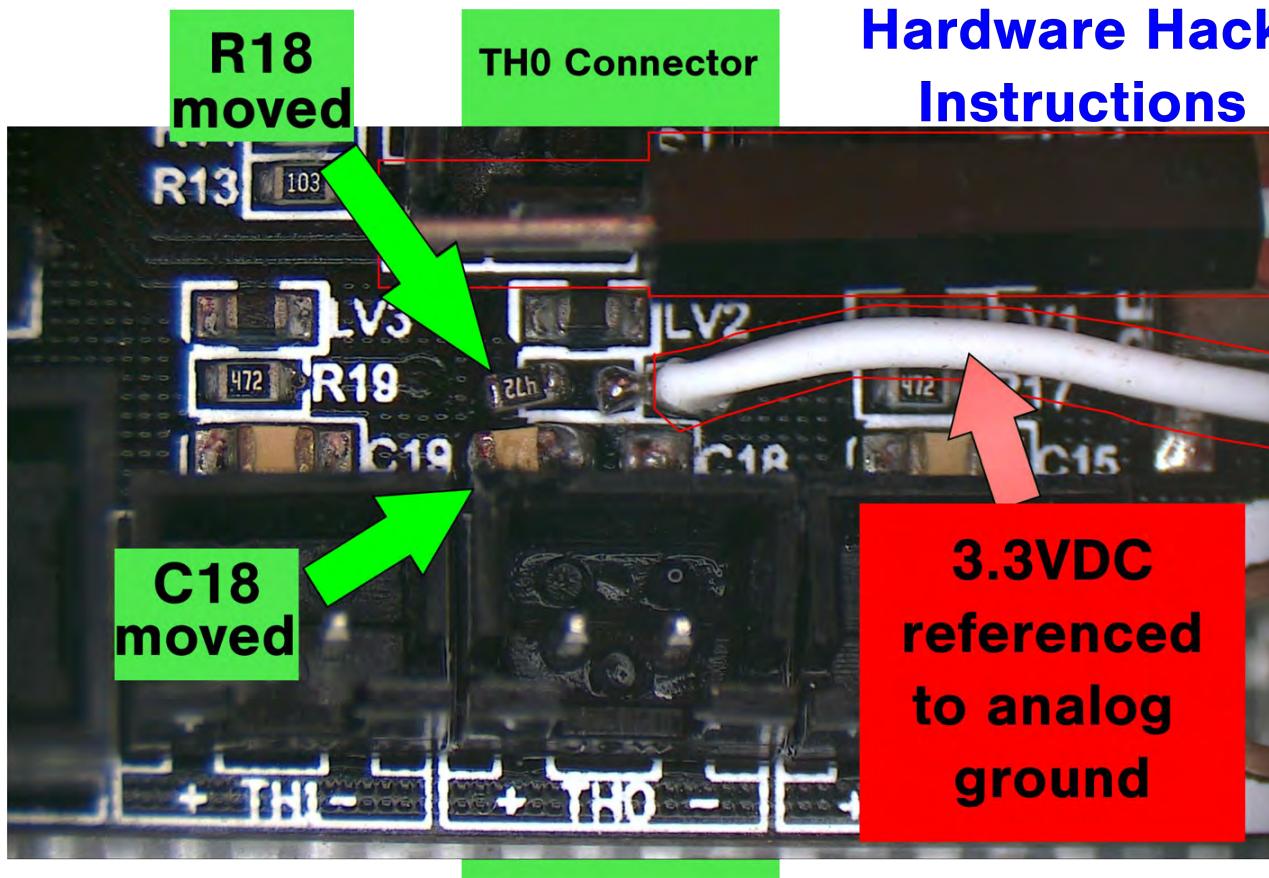
Turn the  
Page to  
see a close  
up of the  
Hacked  
Hardware with  
Instructions\*1.

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### Instructions on how to perform the SKR V1.3 hardware hack to obtain an ADC input on a Thermistor port:

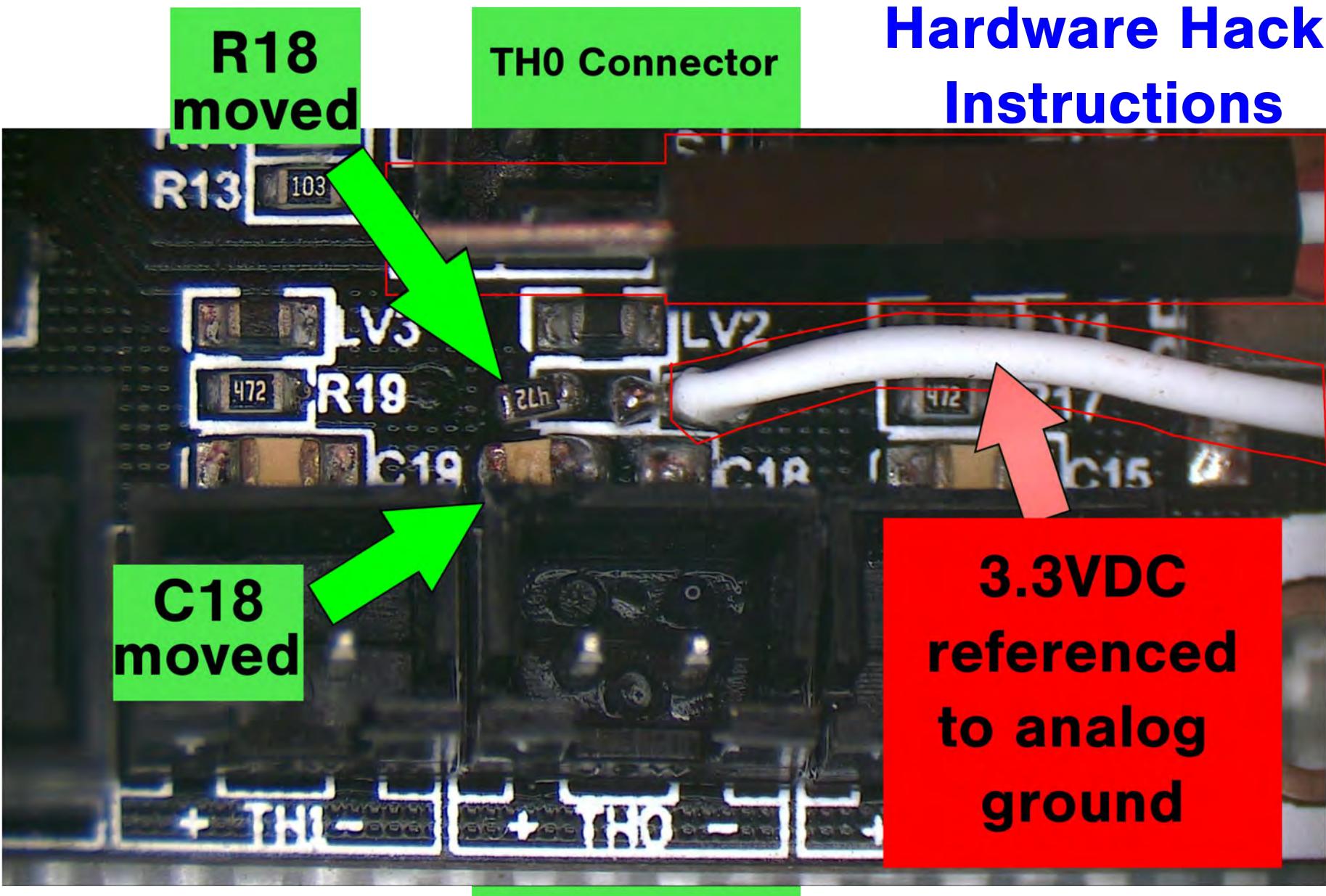
Remove R18 and C18 or shift to left-hand solder pad and leave right-hand solder pad empty. If you don't have a soldering iron you can cut them (R18, C18) with wire cutters, but that will not allow you to revert the hardware back if you want to use TH0 for a thermistor port\*1. Notice the board is oriented so the TH0 label is not upside-down.



The next page shows the above diagram again (full page size) for your convenience.

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor



- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

Now you have to decide on how you want to power the PT100 amplifier board. Where are you going to get the 3.3 VDC and ground for the PT100 amplifier board's +V and 0V PINS? There are two methods to choose from. **Method #1** does not require you to perform another hardware hack, but the power supplied to the PT100 amplifier board will only be conditioned from your 3D Printer's PSU and not conditioned for analog ADC input. While **Method #2** requires you to perform another small hardware hack, but the power will be conditioned and filtered by the electronics (for analog ADC input )found on the MCU board. **Method #2 is the preferred method for noise immunity.**

If you want to use Method #2 for powering the PT100 amplifier board do the following:

1. Obtain a male DuPont Jumper cable. Cut off one end of the Jumper cable and expose the copper wire on that end. Solder the copper end of the male DuPont Jumper cable to the R18's solder pad located on the right-hand side (the board is rotated so that the TH0 labeling is NOT upside-down).

Now its time to supply power to the PT100 amplifier board. The following pages will describe the two methods available to select from. Please chose ONE method.

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### Instructions for Suppling Power to the PT100 Amplifier Board:

#### INTRODUCTION:

If the analog ground (GND\_ANALOG on schematic) and digital ground (GND on schematic) are interconnected in any way, a ground loop is created which introduces noise into the ground circuit which causes errant temperature readings.

Therefore, there are two ways to hook up ground and the 3.3VDC for the amplifier board. **Choose ONE method ONLY. Do NOT inter-mix them!**

Method #1 (Digital PWR) uses digital ground and a 3.3VDC that is referenced to its digital ground. While Method #2 (Analog PWR) uses analog ground and a 3.3VDC that is referenced to its analog ground.

**NOTICE:** Do not pair an analog ground to a digital referenced 3.3VDC. Do not pair a digital ground to an analog referenced 3.3VDC!!

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### Method #1 for Powering the PT100 Amplifier Board (Digital PWR):

If you remove or move R18 and C18 **and DO NOT solder in a male DuPont Jumper cable** onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. the ICSP header provides you access to the 3.3VDC that is reference to the digital ground (GND on the schematic). Therefore, use the ICSP's 3.3VDC PIN to power the amplifier board. Connect the 3.3VDC input of the PT100 amplifier board (+V) to ICSP header's 3.3VDC PIN.
2. the ground located on the ICSP header is one possible digital ground (GND on schematic) available on the MCU. Connect the ground from the ICSP header to the PT100 amplifier board's ground PIN (0V).
3. **DO NOT use the ground from the TH0 connector (GND\_ANALOG on the schematic – this is analog ground you want digital ground!) for this method.**
4. if you use this method (**Method #1**) refer to:
  1. **Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header.**
  2. **Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### Method #2 for Powering PT100 Amplifier Board (Analog PWR):

If you remove or move R18 and C18 and solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. the male DuPont Jumper cable now provides you access to the 3.3VDC that is reference to the analog ground (GND\_ANALOG on schematic). Therefore, use the male DuPont Jumper cable to power the PT100 amplifier board. Place the male DuPont Jumper cable onto the 3.3VDC input of the PT100 amplifier board (+V).
2. the ground located on the TH0 connector is your analog ground (GND\_ANALOG on schematic). Connect the TH0 connector's ground to the PT100 amplifier board's ground PIN (0V).
3. **DO NOT use the ground from the ICSP header (GND on the schematic – this is digital ground you want analog ground!) for this method.**

4. if you use this method (**Method #2**) refer to:

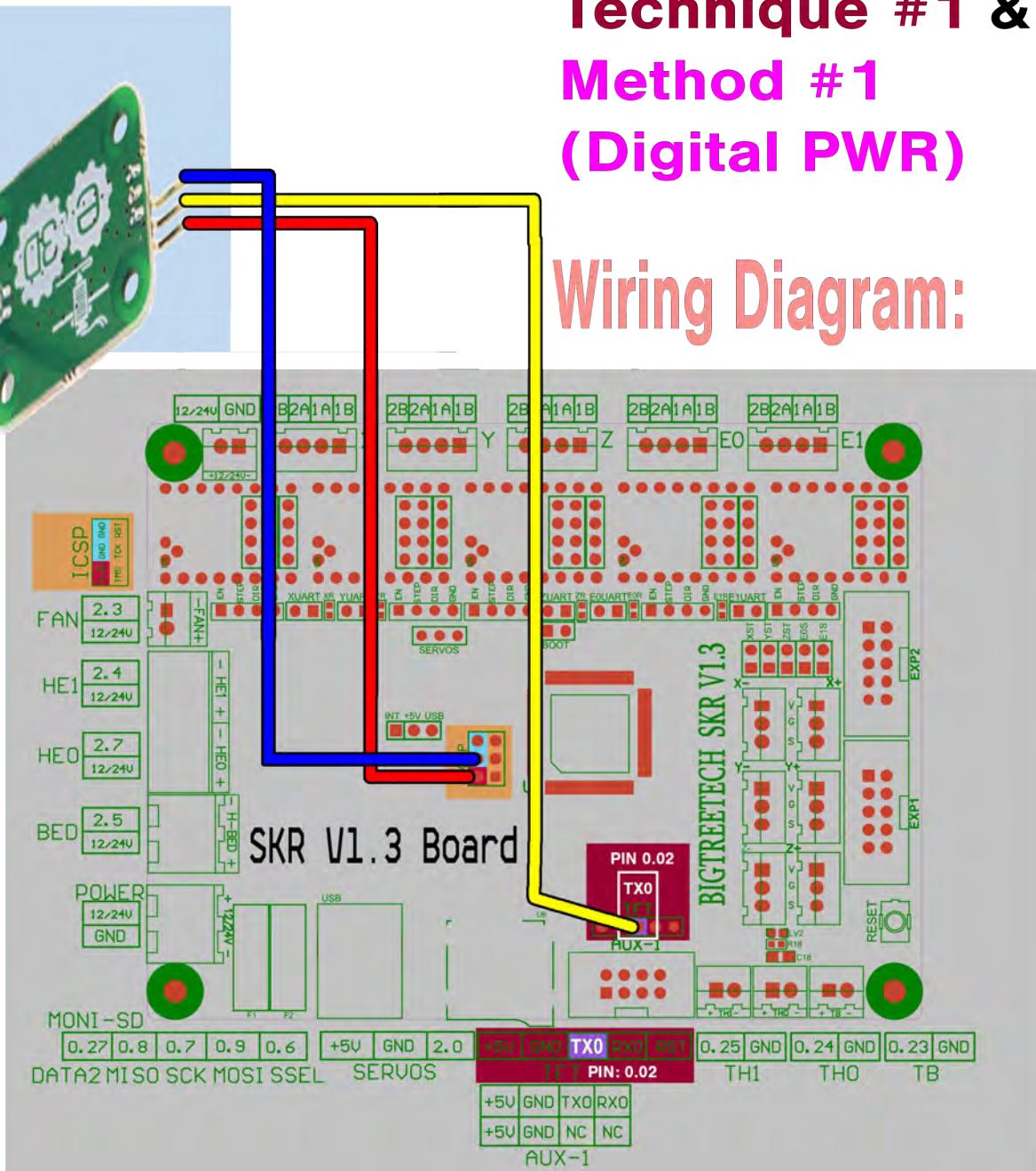
1. **Technique #2 & Method #2 (Analog PWR) wiring diagram** for connecting up your PT100 to the TH0 connector.

**This is the preferred and recommend way to connect up the PT100 amplifier board.**

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

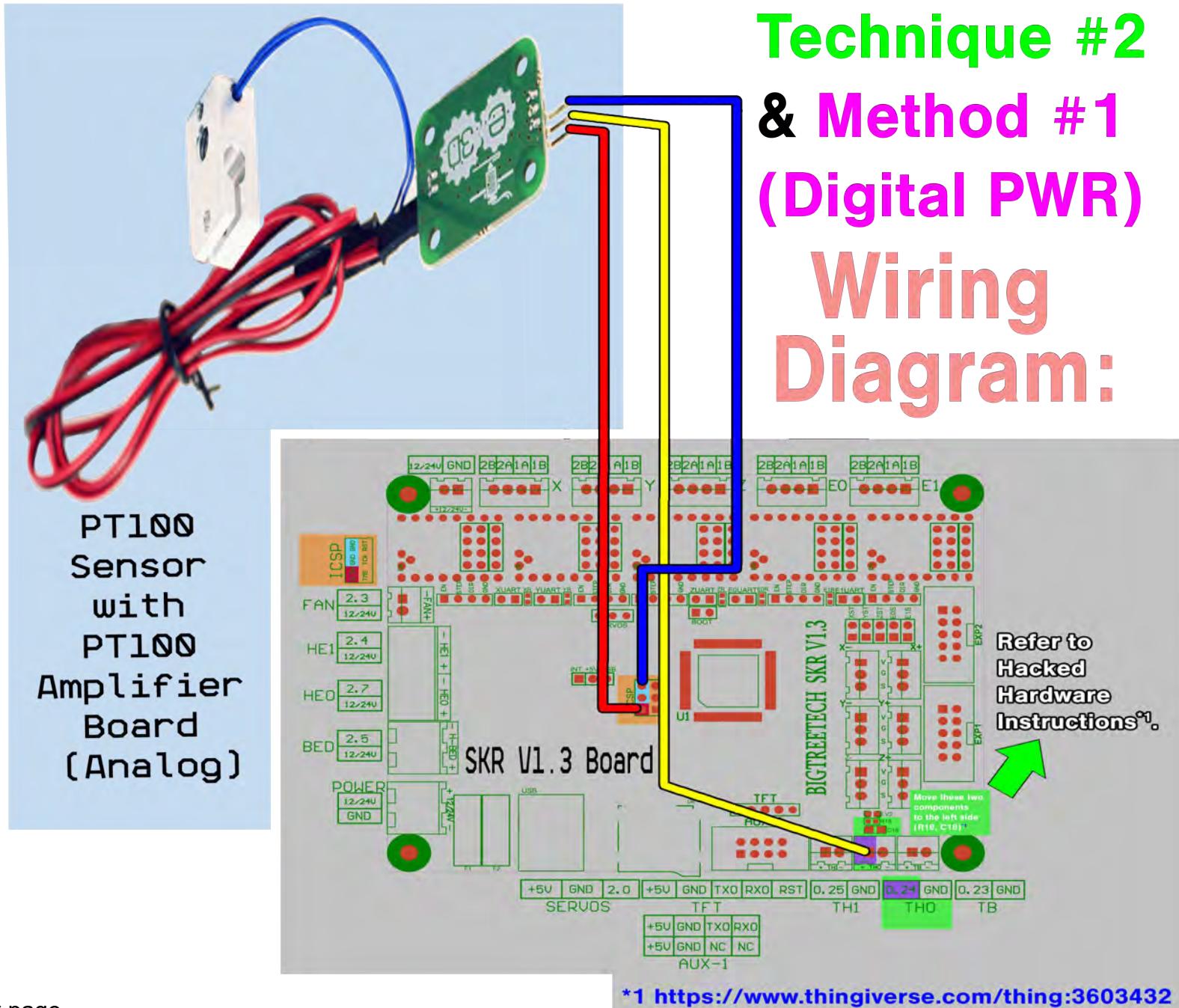
Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header:



- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

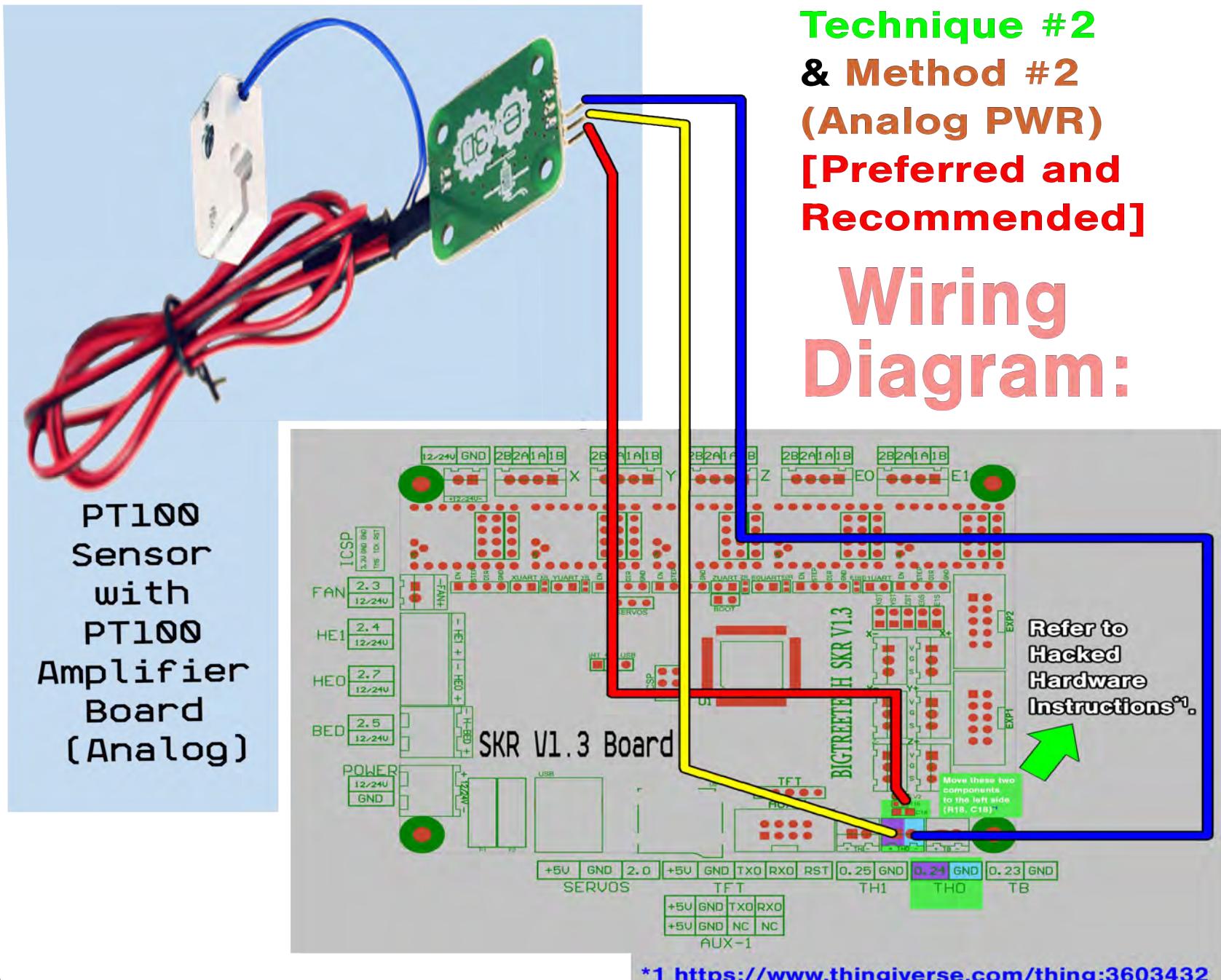
Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector:



- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

**Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your PT100 to the TH0 connector:**



- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

If you are using **Technique #2** goto the pages

titled "Marlin 2.0x Firmware Setup via **Technique #2**".

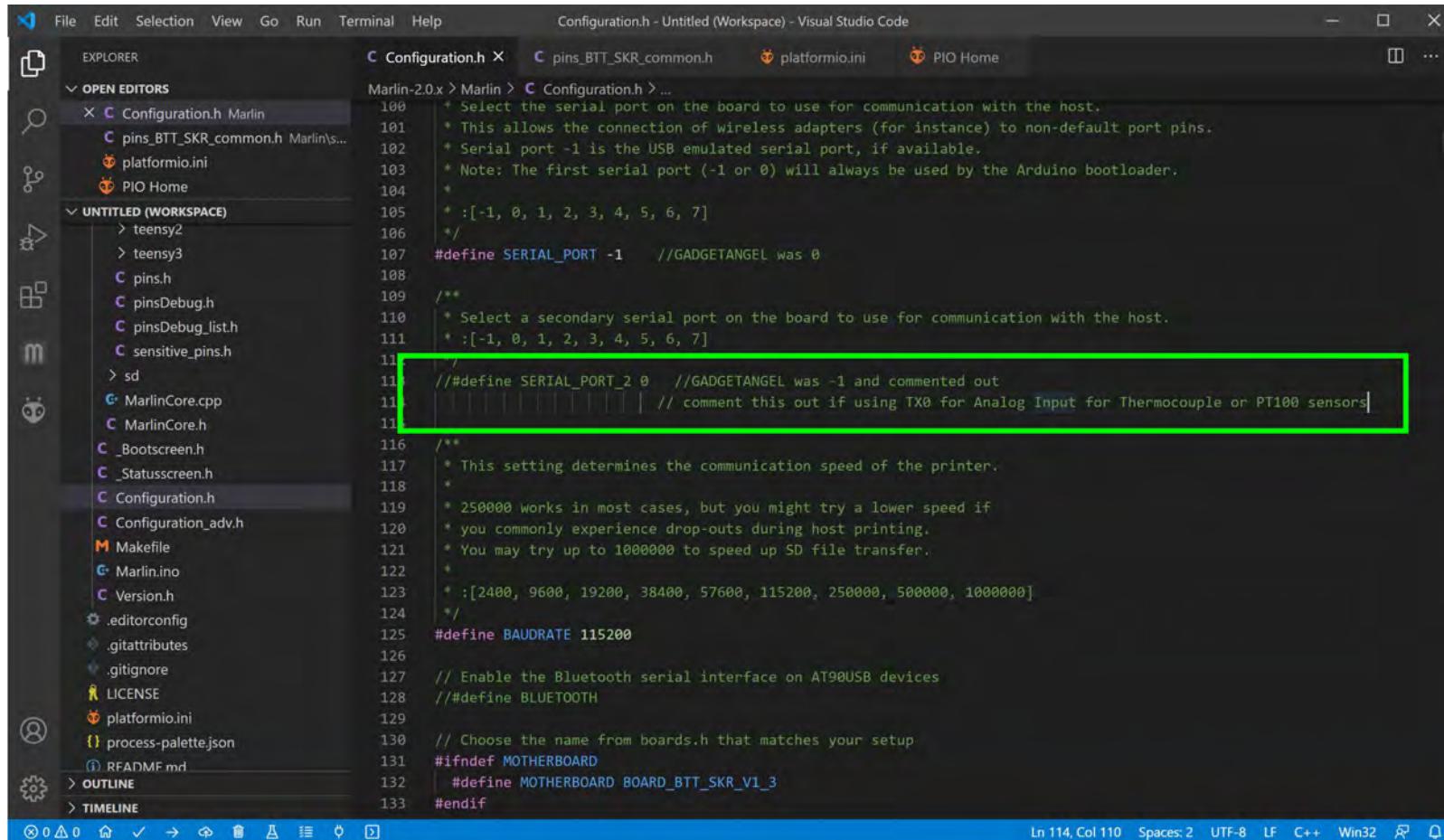
If you are using **Technique #1** goto the next page.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via **Technique #1**

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- **disable** the Marlin variable **SERIRAL\_PORT\_2** or that you comment out the line by inserting two forward slashes ('//') at the beginning of the line.



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following snippet:

```

100 * Select the serial port on the board to use for communication with the host.
101 * This allows the connection of wireless adapters (for instance) to non-default port pins.
102 * Serial port -1 is the USB emulated serial port, if available.
103 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
104 *
105 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
106 */
107 #define SERIAL_PORT -1 //GADGETANGEL was 0
108 /**
109 * Select a secondary serial port on the board to use for communication with the host.
110 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
111 */
112 //##define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out
113 // comment this out if using TX0 for Analog Input for Thermocouple or PT100 sensors
114 */
115 * This setting determines the communication speed of the printer.
116 *
117 * 250000 works in most cases, but you might try a lower speed if
118 * you commonly experience drop-outs during host printing.
119 * You may try up to 1000000 to speed up SD file transfer.
120 *
121 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
122 */
123 #define BAUDRATE 115200
124
125 // Enable the Bluetooth serial interface on AT90USB devices
126 //##define BLUETOOTH
127
128 // Choose the name from boards.h that matches your setup
129 #ifndef MOTHERBOARD
130 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
131 #endif

```

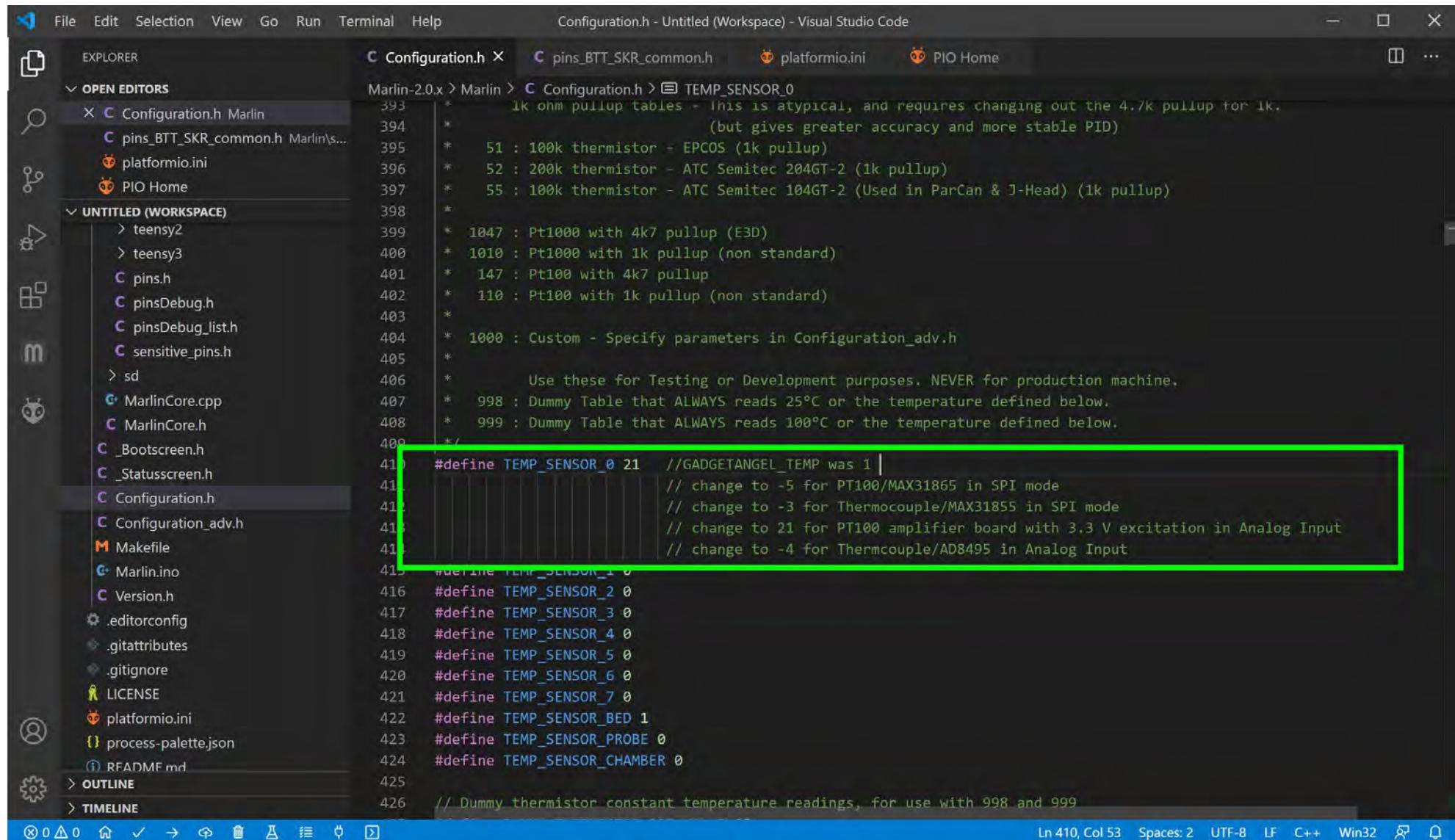
A green rectangular box highlights the line `//##define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out`. Below the code editor, the status bar shows "Ln 114, Col 110 Spaces: 2 UTF-8 LF C++ Win32".

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via Technique #1

- change the Marlin variable **TEMP\_SENSOR\_0** to **21**



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the following snippet:

```

393 * 1k ohm pullup tables - This is atypical, and requires changing out the 4.7k pullup for 1k,
394 * (but gives greater accuracy and more stable PID)
395 * 51 : 100k thermistor - EPCOS (1k pullup)
396 * 52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
397 * 55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
398 *
399 * 1047 : Pt1000 with 4k7 pullup (E3D)
400 * 1010 : Pt1000 with 1k pullup (non standard)
401 * 147 : Pt100 with 4k7 pullup
402 * 110 : Pt100 with 1k pullup (non standard)
403 *
404 * 1000 : Custom - Specify parameters in Configuration_adv.h
405 *
406 * Use these for Testing or Development purposes. NEVER for production machine.
407 * 998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
408 * 999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
409 */
410 #define TEMP_SENSOR_0 21 //GADGETANGEL_TEMP was 1 |
411 // change to -5 for PT100/MAX31865 in SPI mode
412 // change to -3 for Thermocouple/MAX31855 in SPI mode
413 // change to 21 for PT100 amplifier board with 3.3 V excitation in Analog Input
414 // change to -4 for Thermocouple/AD8495 in Analog Input
415 #define TEMP_SENSOR_1 0
416 #define TEMP_SENSOR_2 0
417 #define TEMP_SENSOR_3 0
418 #define TEMP_SENSOR_4 0
419 #define TEMP_SENSOR_5 0
420 #define TEMP_SENSOR_6 0
421 #define TEMP_SENSOR_7 0
422 #define TEMP_SENSOR_BED 1
423 #define TEMP_SENSOR_PROBE 0
424 #define TEMP_SENSOR_CHAMBER 0
425
426 // Dummy thermistor constant temperature readings, for use with 998 and 999

```

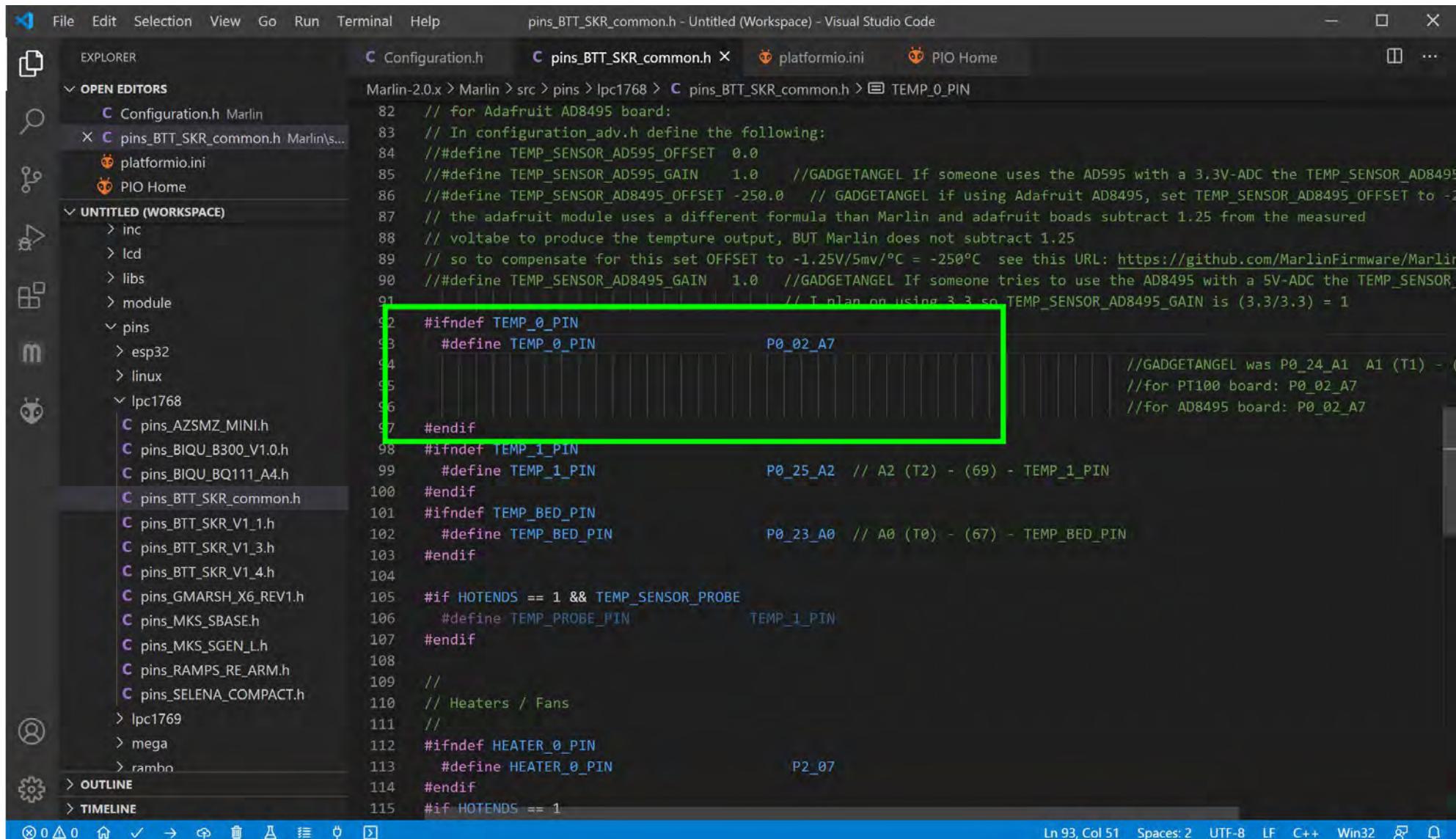
A green rectangular box highlights the line `#define TEMP_SENSOR_0 21`. The status bar at the bottom right indicates 'Ln 410, Col 53'.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via Technique #1

- change the Marlin variable **TEMP\_0\_PIN** to **P0\_02\_A7**



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Title:** pins\_BTT\_SKR\_common.h - Untitled (Workspace) - Visual Studio Code
- Explorer:**
  - OPEN EDITORS: Configuration.h, pins\_BTT\_SKR\_common.h (highlighted), platformio.ini, PIO Home
  - UNTITLED (WORKSPACE):
    - > inc
    - > lcd
    - > libs
    - > module
    - > pins
      - > esp32
      - > linux
      - > lpc1768
        - C pins\_AZSMZ\_MINI.h
        - C pins\_BIQU\_B300\_V1.0.h
        - C pins\_BIQU\_BQ111\_A4.h
        - C pins\_BTT\_SKR\_common.h (highlighted)
        - C pins\_BTT\_SKR\_V1\_1.h
        - C pins\_BTT\_SKR\_V1\_3.h
        - C pins\_BTT\_SKR\_V1\_4.h
        - C pins\_GMARSH\_X6\_REV1.h
        - C pins\_MKS\_SBASE.h
        - C pins\_MKS\_SGEN\_L.h
        - C pins\_RAMPS\_RE\_ARM.h
        - C pins\_SELENA\_COMPACT.h
      - > lpc1769
      - > mega
      - > rambo
    - > OUTLINE
    - > TIMELINE
  - Code Editor:**

```

Marlin-2.0.x > Marlin > src > pins > lpc1768 > C pins_BTT_SKR_common.h > TEMP_0_PIN
82 // for Adafruit AD8495 board:
83 // In configuration_adv.h define the following:
84 //#define TEMP_SENSOR_AD595_OFFSET 0.0
85 //#define TEMP_SENSOR_AD595_GAIN 1.0 //GADGETANGEL If someone uses the AD595 with a 3.3V-ADC the TEMP_SENSOR_AD8495
86 //#define TEMP_SENSOR_AD8495_OFFSET -250.0 // GADGETANGEL if using Adafruit AD8495, set TEMP_SENSOR_AD8495_OFFSET to -250.0
87 // the adafruit module uses a different formula than Marlin and adafruit boards subtract 1.25 from the measured
88 // voltage to produce the temperature output, BUT Marlin does not subtract 1.25
89 // so to compensate for this set OFFSET to -1.25V/5mV/°C = -250°C see this URL: https://github.com/MarlinFirmware/Marlin
90 //#define TEMP_SENSOR_AD8495_GAIN 1.0 //GADGETANGEL If someone tries to use the AD8495 with a 5V-ADC the TEMP_SENSOR_
91 // I plan on using 3.3 so TEMP_SENSOR_AD8495_GAIN is (3.3/3.3) = 1
92 #ifndef TEMP_0_PIN
93 #define TEMP_0_PIN P0_02_A7
94
95
96
97 #endif
98 #ifndef TEMP_1_PIN
99 #define TEMP_1_PIN P0_25_A2 // A2 (T2) - (69) - TEMP_1_PIN
100 #endif
101 #ifndef TEMP_BED_PIN
102 #define TEMP_BED_PIN P0_23_A0 // A0 (T0) - (67) - TEMP_BED_PIN
103 #endif
104
105 #if HOTENDS == 1 && TEMP_SENSOR_PROBE
106 #define TEMP_PROBE_PIN TEMP_1_PIN
107 #endif
108 //
109 // Heaters / Fans
110 //
111 //
112 #ifndef HEATER_0_PIN
113 #define HEATER_0_PIN P2_07
114 #endif
115 #if HOTENDS == 1

```
  - Bottom Status Bar:** Ln 93, Col 51, Spaces: 2, UTF-8, LF, C++, Win32, etc.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via Technique #1

- Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Explorer:** Shows files in the workspace, including Configuration.h, pins\_BTT\_SKR\_common.h, platformio.ini, and PIO Home.
- Editor:** Shows the content of Configuration.h with code related to the TEMP\_SENSOR\_0 pin.
- Terminal:** Shows the following output:
 

```
RAM: [====] 42.3% (used 13840 bytes from 32736 bytes)
Flash: [====] 38.8% (used 184428 bytes from 475136 bytes)
===== [SUCCESS] Took 115.30 seconds =====
```
- Task List:** Shows the task "2: Task - PlatformIO: Run" completed successfully with a duration of 00:01:55.300.
- Status Bar:** Shows the status bar with "Ln 410, Col 53" and other build-related information.

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

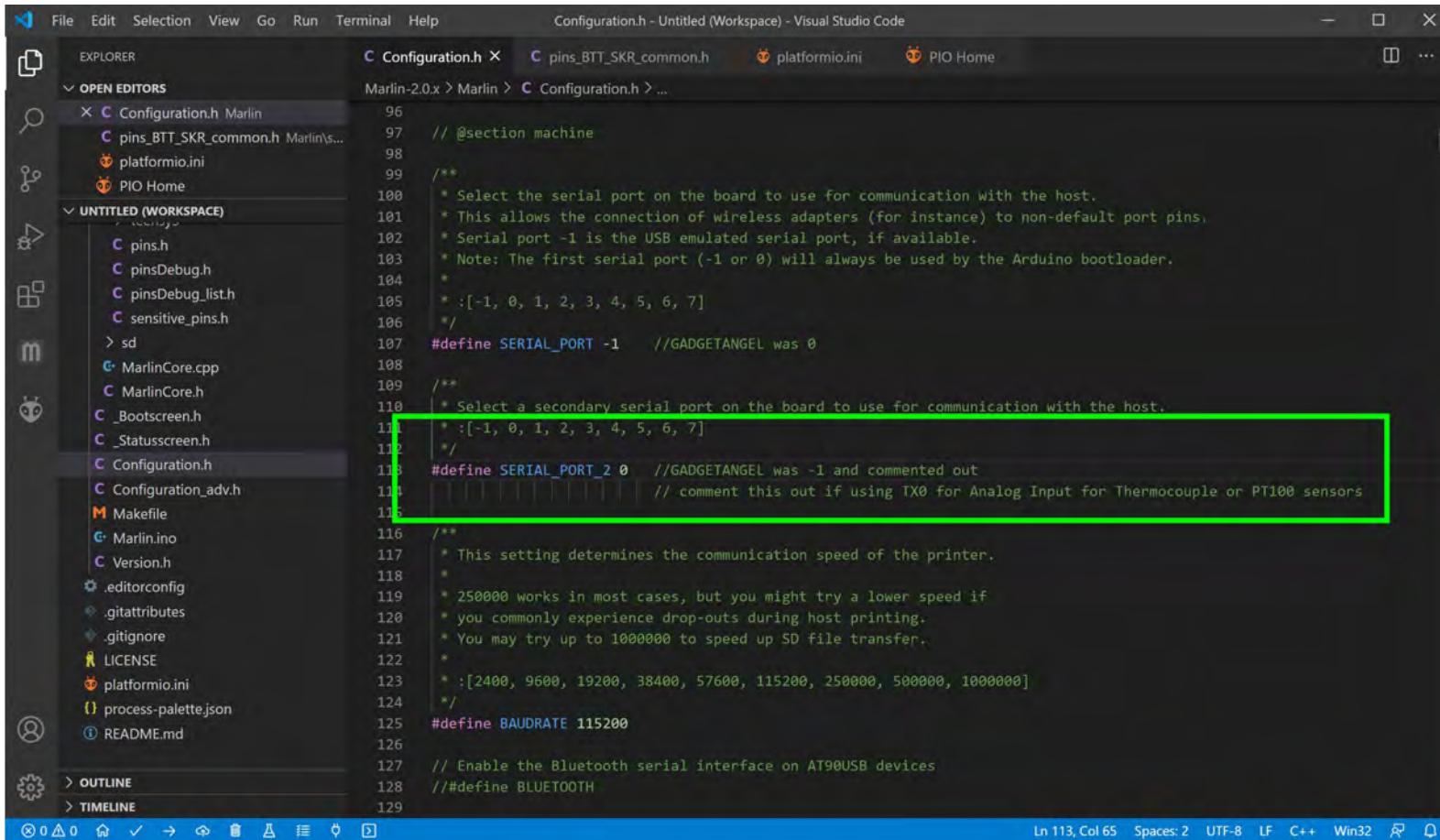
### Marlin 2.0.x Firmware Setup via [Technique #1](#)

- mggr 'vj g'O ct nk'ugwkpi u'vj g'lkco g'\*cu'cdqxg+''  
wpvkn'vj g'RV322'ugpqt 'ku'y qt mki 0'  
Chgt y ctf u'{ qw'ct g'ht gg'wq'wy gcm'cy c{ 0
- CmHpkj gf 'y kj 'ugwkpi 'wr 'vj g'RV322'  
co r nk'gt 'dqctf 'cpf 'RV322'ugpuqt using  
**Technique #1##**

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via Technique #2

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- **if disabled please enable** the Marlin variable **SERIRAL\_PORT\_2** or that you **delete the two forward slashes ("//")** from the beginning of the line.



The screenshot shows the Visual Studio Code interface with the "Configuration.h - Untitled (Workspace)" tab selected. The code editor displays the following snippet from Configuration.h:

```

96 // @section machine
97 /**
98 * Select the serial port on the board to use for communication with the host.
99 * This allows the connection of wireless adapters (for instance) to non-default port pins.
100 * Serial port -1 is the USB emulated serial port, if available.
101 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
102 *
103 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
104 */
105 #define SERIAL_PORT -1 //GADGETANGEL was 0
106 /**
107 * Select a secondary serial port on the board to use for communication with the host.
108 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
109 */
110 #define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out
111 // comment this out if using TX0 for Analog Input for Thermocouple or PT100 sensors
112 /**
113 * This setting determines the communication speed of the printer.
114 * 250000 works in most cases, but you might try a lower speed if
115 * you commonly experience drop-outs during host printing.
116 * You may try up to 1000000 to speed up SD file transfer.
117 *
118 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
119 */
120 #define BAUDRATE 115200
121 /**
122 * Enable the Bluetooth serial interface on AT90USB devices
123 // #define BLUETOOTH
124 */
125
126
127
128
129

```

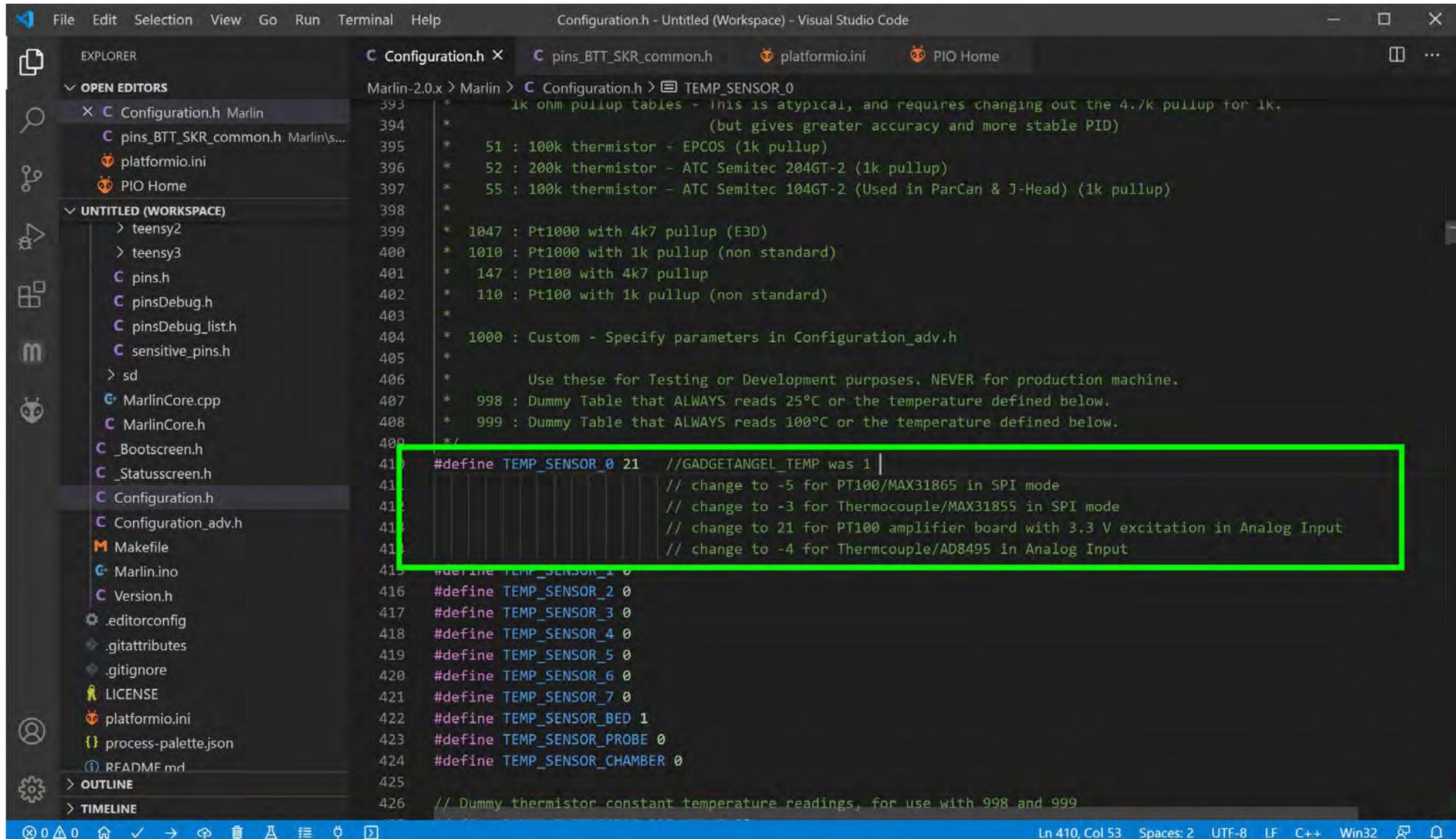
The line `#define SERIAL_PORT_2 0` is highlighted with a green rectangular selection. The status bar at the bottom right shows "Ln 113, Col 65".

- Go to the next page.

## Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

### Marlin 2.0.x Firmware Setup via [Technique #2](#)

- change the Marlin variable **TEMP\_SENSOR\_0** to **21**



The screenshot shows the Visual Studio Code interface with the 'Configuration.h' file open. The code editor displays the Marlin 2.0.x configuration header. A green rectangular box highlights the line:

```
#define TEMP_SENSOR_0 21 //GADGETANGEL_TEMP was 1 |
// change to -5 for PT100/MAX31865 in SPI mode
// change to -3 for Thermocouple/MAX31855 in SPI mode
// change to 21 for PT100 amplifier board with 3.3 V excitation in Analog Input
// change to -4 for Thermocouple/AD8495 in Analog Input
```

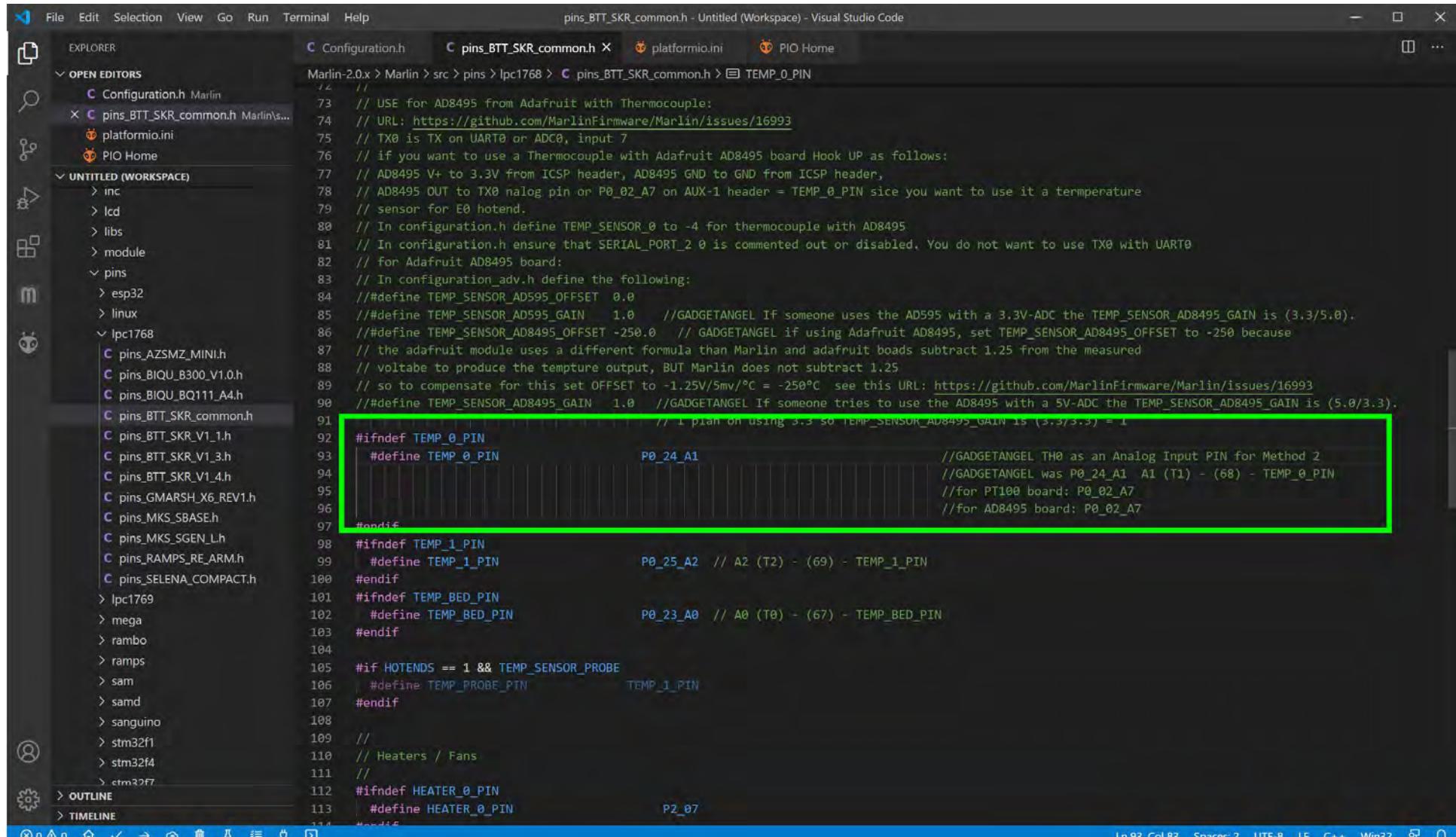
The code editor's status bar at the bottom indicates: Ln 410, Col 53, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

# Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

## Marlin 2.0.x Firmware Setup via [Technique #2](#)

- change the Marlin variable **TEMP\_0\_PIN** to **P0\_24\_A1**



The screenshot shows the Visual Studio Code interface with the file `pins_BTT_SKR_common.h` open. The code editor displays the following snippet:

```

14 //
15 // USE for AD8495 from Adafruit with Thermocouple:
16 // URL: https://github.com/MarlinFirmware/Marlin/issues/16993
17 // TX0 is TX on UART0 or ADC0, input 7
18 // if you want to use a Thermocouple with Adafruit AD8495 board Hook UP as follows:
19 // AD8495 V+ to 3.3V from ICSP header, AD8495 GND to GND from ICSP header,
20 // AD8495 OUT to TX0 analog pin or P0_02_A7 on AUX-1 header = TEMP_0_PIN since you want to use it a temperature
21 // sensor for E0 hotend.
22 // In configuration.h define TEMP_SENSOR_0 to -4 for thermocouple with AD8495
23 // In configuration.h ensure that SERIAL_PORT_2 0 is commented out or disabled. You do not want to use TX0 with UART0
24 // for Adafruit AD8495 board:
25 // In configuration_adv.h define the following:
26 // #define TEMP_SENSOR_AD595_OFFSET 0.0
27 // #define TEMP_SENSOR_AD595_GAIN 1.0 //GADGETANGEL If someone uses the AD595 with a 3.3V-ADC the TEMP_SENSOR_AD8495_GAIN is (3.3/5.0).
28 // #define TEMP_SENSOR_AD8495_OFFSET -250.0 // GADGETANGEL if using Adafruit AD8495, set TEMP_SENSOR_AD8495_OFFSET to -250 because
29 // the adafruit module uses a different formula than Marlin and adafruit boards subtract 1.25 from the measured
30 // voltage to produce the tempture output, BUT Marlin does not subtract 1.25
31 // so to compensate for this set OFFSET to -1.25V/mv/°C = -250°C see this URL: https://github.com/MarlinFirmware/Marlin/issues/16993
32 // #define TEMP_SENSOR_AD8495_GAIN 1.0 //GADGETANGEL If someone tries to use the AD8495 with a 5V-ADC the TEMP_SENSOR_AD8495_GAIN is (5.0/3.3).
33 // I plan on using 3.3 so TEMP_SENSOR_AD8495_GAIN is (3.3/3.3) = 1
34 #ifndef TEMP_0_PIN
35 #define TEMP_0_PIN P0_24_A1 //GADGETANGEL TH0 as an Analog Input PIN for Method 2
36 //GADGETANGEL was P0_24_A1 A1 (T1) - (68) - TEMP_0_PIN
37 //for PT100 board: P0_02_A7
38 //for AD8495 board: P0_02_A7
39 #endif
40 #ifndef TEMP_1_PIN
41 #define TEMP_1_PIN P0_25_A2 // A2 (T2) - (69) - TEMP_1_PIN
42 #endif
43 #ifndef TEMP_BED_PIN
44 #define TEMP_BED_PIN P0_23_A0 // A0 (T0) + (67) - TEMP_BED_PIN
45 #endif
46 #if HOTENDS == 1 && TEMP_SENSOR_PROBE
47 #define TEMP_PROBE_PIN TEMP_1_PIN
48 #endif
49 //
50 // Heaters / Fans
51 //
52 #ifndef HEATER_0_PIN
53 #define HEATER_0_PIN P2_07
54 #endif

```

A green rectangular box highlights the line `#define TEMP_0_PIN P0_24_A1`. The status bar at the bottom right shows: Ln 93, Col 83, Spaces: 2, UTF-8, LF, C++, Win32.

- Go to the next page.

Marlin 2.0.x Firmware Setup for Connecting a PT100 Sensor to the SKR V1.3 Board

## **Marlin 2.0.x Firmware Setup via [Technique #2](#)**

- Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files and folders related to the Marlin 2.0.0 project, including Configuration.h, pins\_BTT\_SKR\_common.h, platformio.ini, and various Marlin source files.
- Editor:** The main editor area displays the Configuration.h file with code for defining pins and custom machine names.
- Terminal:** A terminal window at the bottom shows the build process output:
  - Advanced Memory Usage information: RAM usage is 43.2% (14132 bytes from 32736), Flash usage is 42.0% (199632 bytes from 475136).
  - The build took 114.34 seconds and was successful.
- Environment Table:** A table below the terminal shows supported environments and their status. The row for **LPC1768** is highlighted with a green border and labeled **SUCCESS**.
- Bottom Status Bar:** Shows the current line (Ln 113), column (Col 65), and other system information like spaces used and file type.

- Go to the next page.

## Connecting SKR V1.3 with PT100 Amplifier Board and PT100 Sensor

### Marlin 2.0.x Firmware Setup via [Technique #2](#)

- mggr 'vj g'O ctnkp'ugwki u'vj g'tro g'\*cu'cdqxg+'  
wpvkn'vj g'RV322'ugpqt 'ku'y qt mki 0'  
Chgt y ctf u'{qw'ctg'lt gg'vq'wy gcmcy c{ 0
- CmHkpkuj gf 'y kj 'ugwki 'wr 'vj g'RV322''  
co r nkhkt 'dqctf 'cpf 'RV322'ugpuqt using  
**Technique #2##**

## APPENDIX L

### Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

To install a K-Type Thermocouple temperature sensor to your 3D printer requires four steps:

- Decide which of the two techniques you will use to connect the K-Type thermocouple amplifier board (AD8495) to the MCU board (SKR V1.3 board).
- Wire up the K-Type thermocouple Sensor to the Adafruit's AD8495 Amplifier board.
- Wire up the Adafruit's AD8495 amplifier board to the MCU board (SKR V1.3 board). If you chose to connect up the K-Type thermocouple with **Technique #1** then you **MUST** use **Method #1 (Digital PWR)** to power your amplifier board. If you chose to connect up the K-Type thermocouple with **Technique #2** than you have two methods to select from to power your amplifier board: **Method #1 (Digital PWR)** or **Method #2 (Analog PWR)**.
- Marlin 2.0.x Firmware setup.

There are two techniques you can use to wire up the Adafruit's AD8495 Amplifier board to the MCU:

- **Technique #1:** Attach the AD8495 amplifier board to the TFT Port by using TX0 PIN. With this technique you lose the ability to use the TFT. The advantage is that you do not have to perform hardware hacks on the SKR V1.3 board. **ONLY** use **Method #1 (Digital PWR)** to power your amplifier board.
- **Technique #2<sup>\*1</sup>:** Attach Adafruit's AD8495 amplifier board to a thermistor port (TH0 or TH1). The disadvantage of this method is that you must perform hardware hacks to change the thermistor port (TH0) PIN to an analog input PIN. This requires removing a resistor and capacitor from the chosen thermistor port (TH0). The advantage of this method is that you **CAN** use the TFT because you will not be using TX0 PIN with this technique.

\*1: <https://www.thingiverse.com/thing:3603432>

- Go to the next page.

## APPENDIX L

### Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

If you chose to power the amplifier board by using **Method #2 (Analog PWR)** you get the advantage of noise immunity. Another advantage is that there are very few changes needed in Marlin 2.0.x Firmware. With **Technique #2** you have to decide which method you will use to power your amplifier board. There are two methods to select from: **Method #1 (Digital PWR)**, or **Method #2 (Analog PWR)**. **Method #2 is the preferred method with this technique** due to noise immunity.

**Technique #2 & Method #2 is the preferred and recommended option to connect your SKR V1.3 board up to a K-Type Thermocouple Sensor (with AD8495 amplifier board).**

The diagram on the next page shows you the AD8495 amplifier board attached to a K-Type Thermocouple temperature sensor.

It also indicates that the amplifier board will be using the ADC input on the MCU not SPI inputs. SPI inputs uses digital I/O lines while the ADC input uses an analog I/O line.

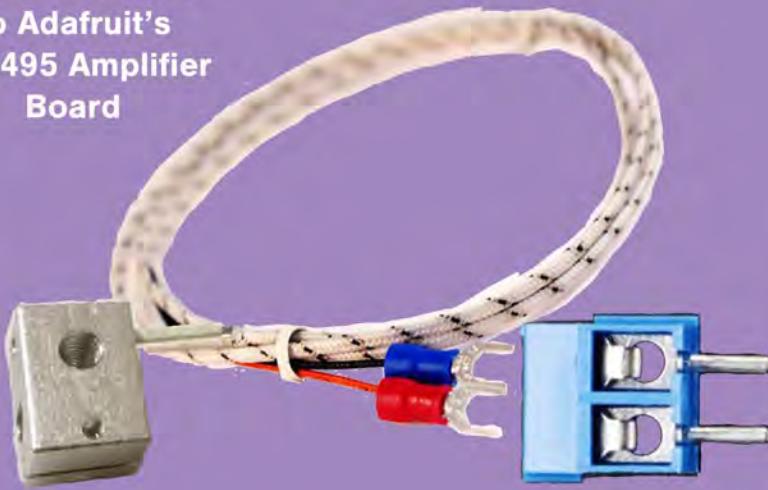
- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## K-Type Thermocouple Cartridge Hooked up to Adafruit's AD8495 Amplifier Board (ADC Input)

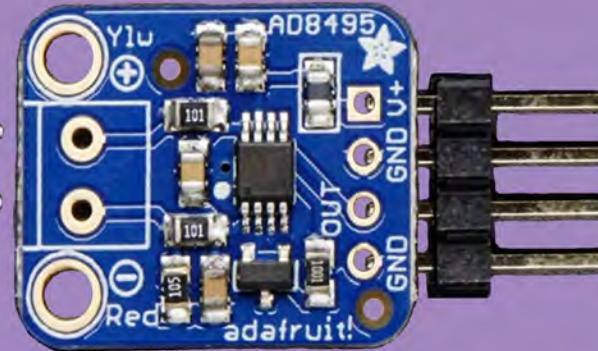
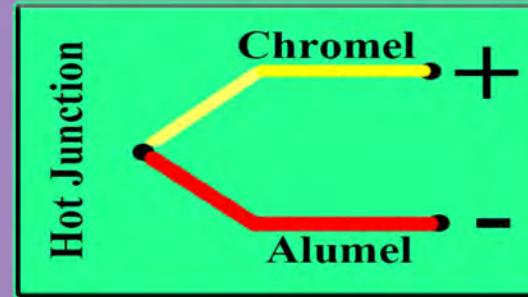
K-Type Thermocouple  
Sensor connected  
to Adafruit's  
AD8495 Amplifier  
Board



K-Type Thermocouple  
Cartridge in  
Heater  
Block

**Thermocouples:**

If you take two wires made of dissimilar metals, connect them at the two ends, and make a temperature gradient between one end and the other, a voltage potential forms and current flows. One junction is held in the environment where the temperature of interest exists. This is known as the hot junction. The other junction is referred to as the cold junction<sup>2</sup>.



\*2 URL: for “Brief Theory  
of Operation” at

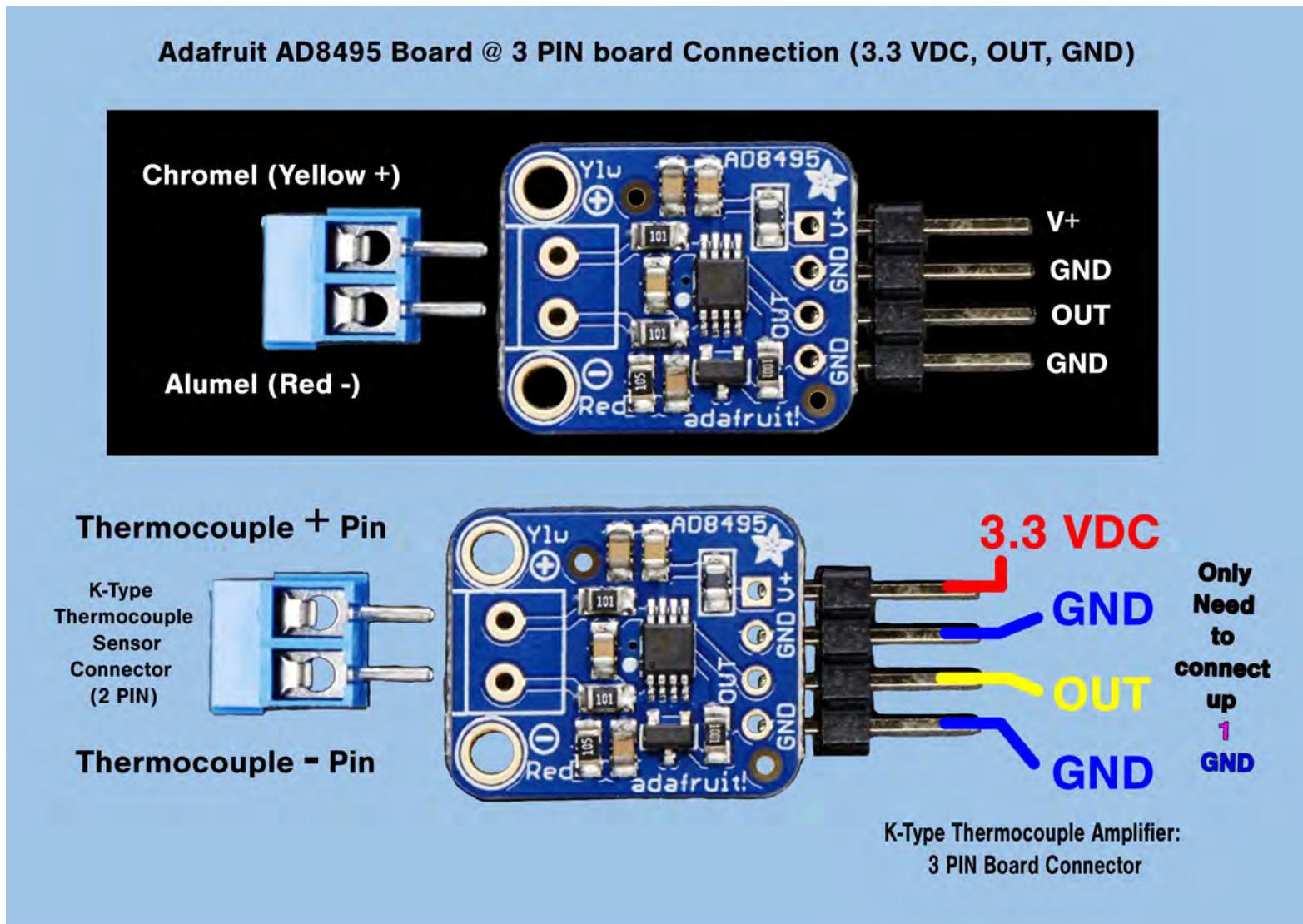
[https://learn.sparkfun.com/tutorials/  
max31855k-thermocouple-breakout-hookup-guide](https://learn.sparkfun.com/tutorials/max31855k-thermocouple-breakout-hookup-guide)

- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

The diagram below shows the locations for **+V**, **GND**, and **OUT** (which is the ADC output signal that carries the information about the temperature). The diagram below also shows where the K-Type Thermocouple sensor pins are located. The Chromel side of the thermocouple is usually indicated with a **YELLOW(+)** or **BLUE(+)** end. The Alumel side of the thermocouple is usually indicated with a **RED(-)** end. The board on top and bottom is the **AD8495 amplifier board made by Adafruit**. The board's silk screen (labels) show the location for each OUTPUT PIN.

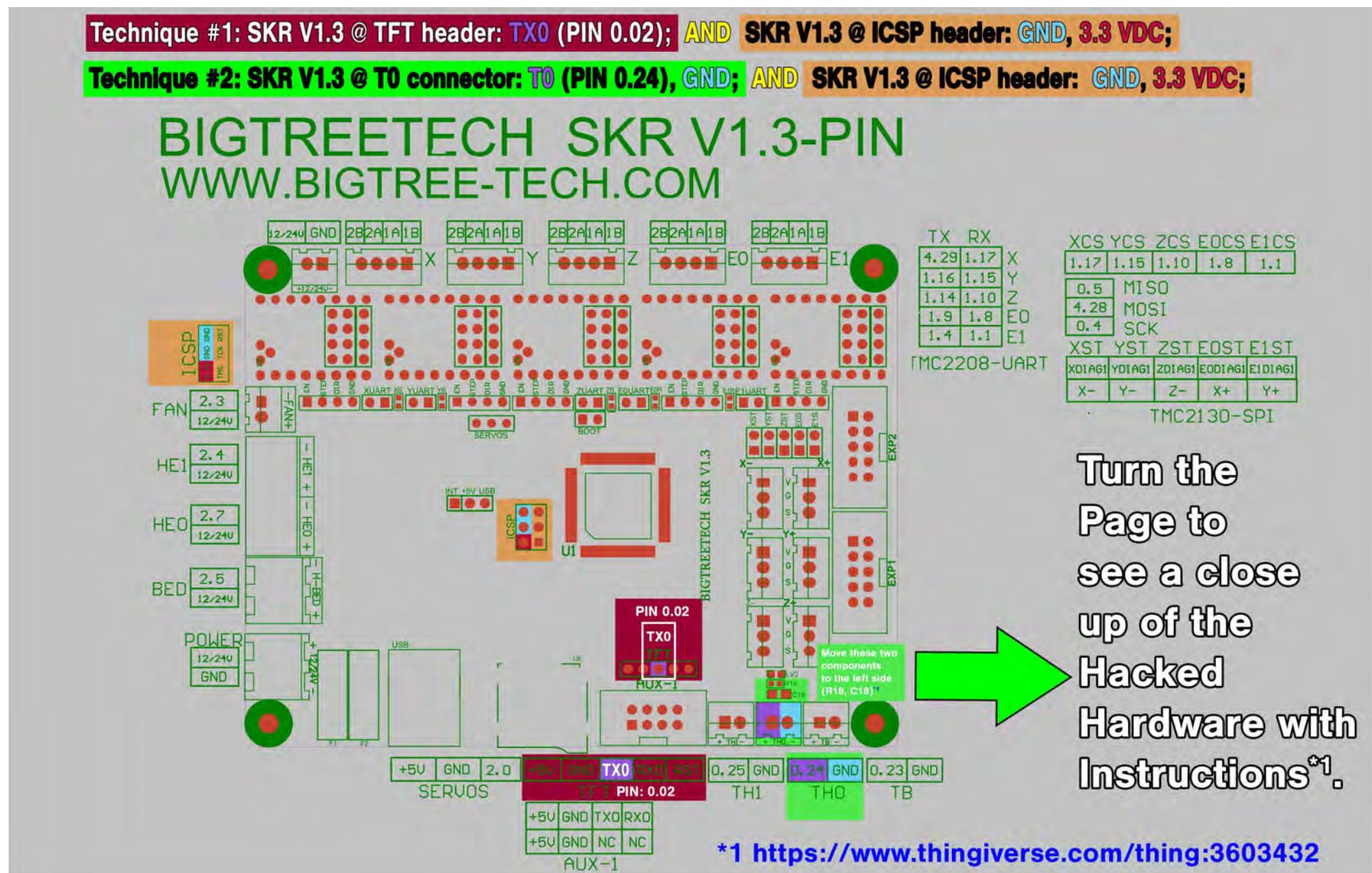


- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

The diagram below shows the areas on the SKR V1.3 board which are of interest when wanting to connect up the K-Type Thermocouple temperature sensor with the AD8495 amplifier board. This is an overview. The diagram indicates which PINS are involved in **Technique #1** and **Technique #2**.



Turn the  
Page to  
see a close  
up of the  
Hacked  
Hardware with  
Instructions\*1.

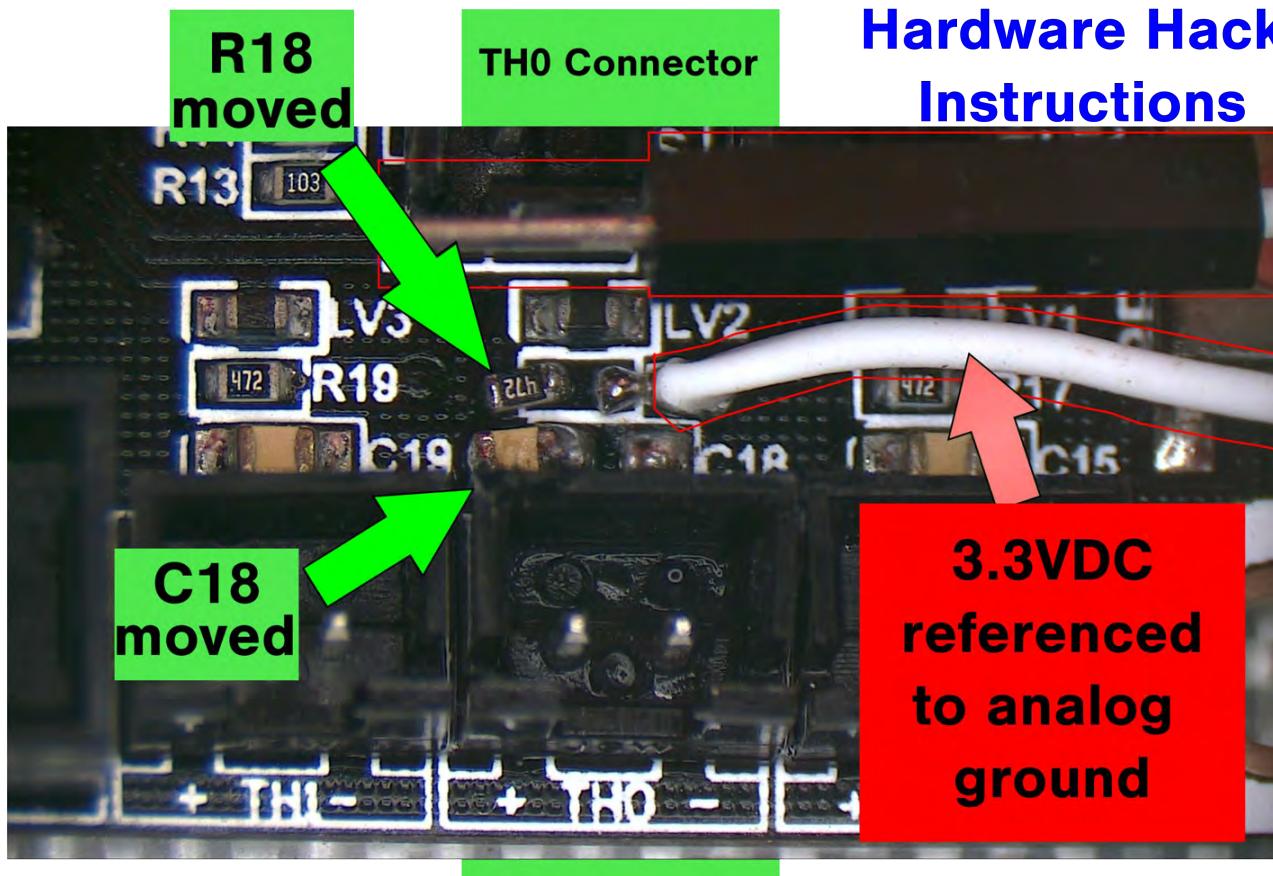
- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Instructions on how to perform the SKR V1.3 hardware hack to obtain an ADC input on a Thermistor port:

Remove R18 and C18 or shift to left-hand solder pad and leave right-hand solder pad empty. If you don't have a soldering iron you can cut them (R18, C18) with wire cutters, but that will not allow you to revert the hardware back if you want to use TH0 for a thermistor port\*1. Notice the board is oriented so the TH0 label is not upside-down.



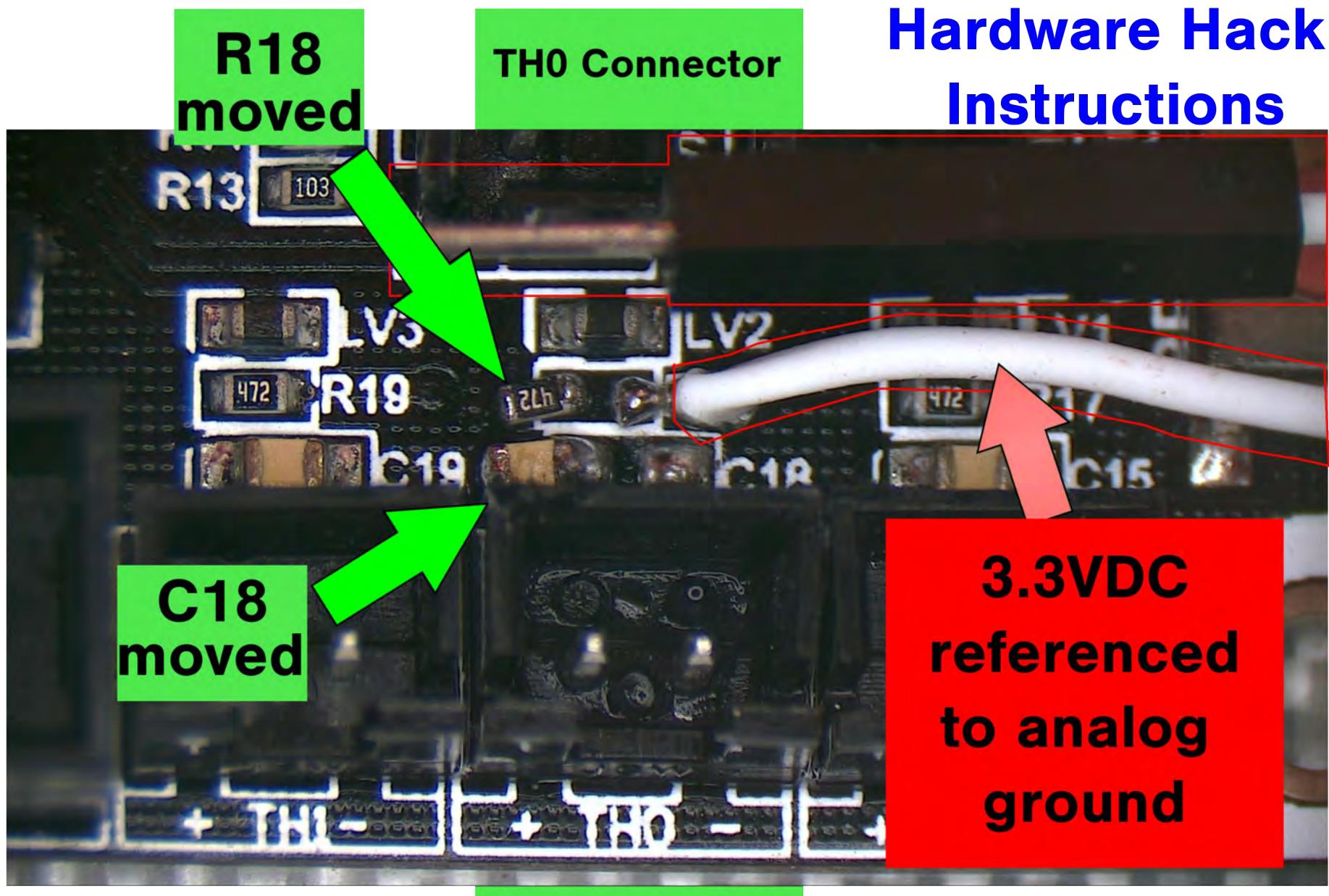
The next page shows the above diagram again (full page size) for your convenience.

- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Hardware Hack Instructions



- Go to the next page.

## APPENDIX L

### Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Now you have to decide on how you want to power the AD8495 amplifier board. Where are you going to get the 3.3 VDC and ground for the AD8495 amplifier board's +V and 0V PINS? There are two methods to choose from. **Method #1** does not require you to perform another hardware hack, but the power supplied to the AD8495 amplifier board will only be conditioned from your 3D Printer's PSU and not conditioned for analog ADC input. While **Method #2** requires you to perform another small hardware hack, but the power will be conditioned and filtered by the electronics (for analog ADC input) found on the MCU board. **Method #2 is the preferred method for noise immunity.**

If you want to use Method #2 for powering the AD8495 amplifier board do the following:

1. Obtain a male DuPont Jumper cable. Cut off one end of the Jumper cable and expose the copper wire on that end. Solder the copper end of the male DuPont Jumper cable to the R18's solder pad located on the right-hand side (the board is rotated so that the TH0 labeling is NOT upside-down).

Now its time to supply power to the AD8495 amplifier board. The following pages will describe the two methods available to select from. Please chose ONE method.

- Go to the next page.

## APPENDIX L

# Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

### Instructions for Suppling Power to the AD8495 Amplifier Board:

#### INTRODUCTION:

If the analog ground (GND\_ANALOG on schematic) and digital ground (GND on schematic) are interconnected in any way, a ground loop is created which introduces noise into the ground circuit which causes errant temperature readings.

Therefore, there are two ways to hook up ground and the 3.3VDC for the amplifier board. **Choose ONE method ONLY. Do NOT inter-mix them!**

Method #1 (Digital PWR) uses digital ground and a 3.3VDC that is referenced to its digital ground. While Method #2 (Analog PWR) uses analog ground and a 3.3VDC that is referenced to its analog ground.

**NOTICE:** Do not pair an analog ground to a digital referenced 3.3VDC. Do not pair a digital ground to an analog referenced 3.3VDC!!

- Go to the next page.

## APPENDIX L

# Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

### Method #1 for Powering the AD8495 Amplifier Board (Digital PWR):

If you remove or move R18 and C18 **and DO NOT solder in a male DuPont Jumper cable** onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

1. the ICSP header provides you access to the 3.3VDC that is reference to the digital ground (GND on the schematic). Therefore, use the ICSP's 3.3VDC PIN to power the amplifier board. Connect the 3.3VDC input of the AD8495 amplifier board (+V) to ICSP header's 3.3VDC PIN.
2. the ground located on the ICSP header is one possible digital ground (GND on schematic) available on the MCU. Connect the ground from the ICSP header to the AD8495 amplifier board's ground PIN (0V).
3. **DO NOT use the ground from the TH0 connector (GND\_ANALOG on the schematic – this is analog ground you want digital ground!) for this method.**
4. if you use this method (**Method #1**) refer to:
  1. **Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TFT header.**
  2. **Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your PT100 to the TH0 connector.**

- Go to the next page.

## APPENDIX L

### Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

#### Method #2 for Powering AD8495 Amplifier Board (Analog PWR):

If you remove or move R18 and C18 and solder in a male DuPont Jumper cable onto R18's solder pad located on the right-hand side, then do the following (board oriented so the TH0 label is not upside-down):

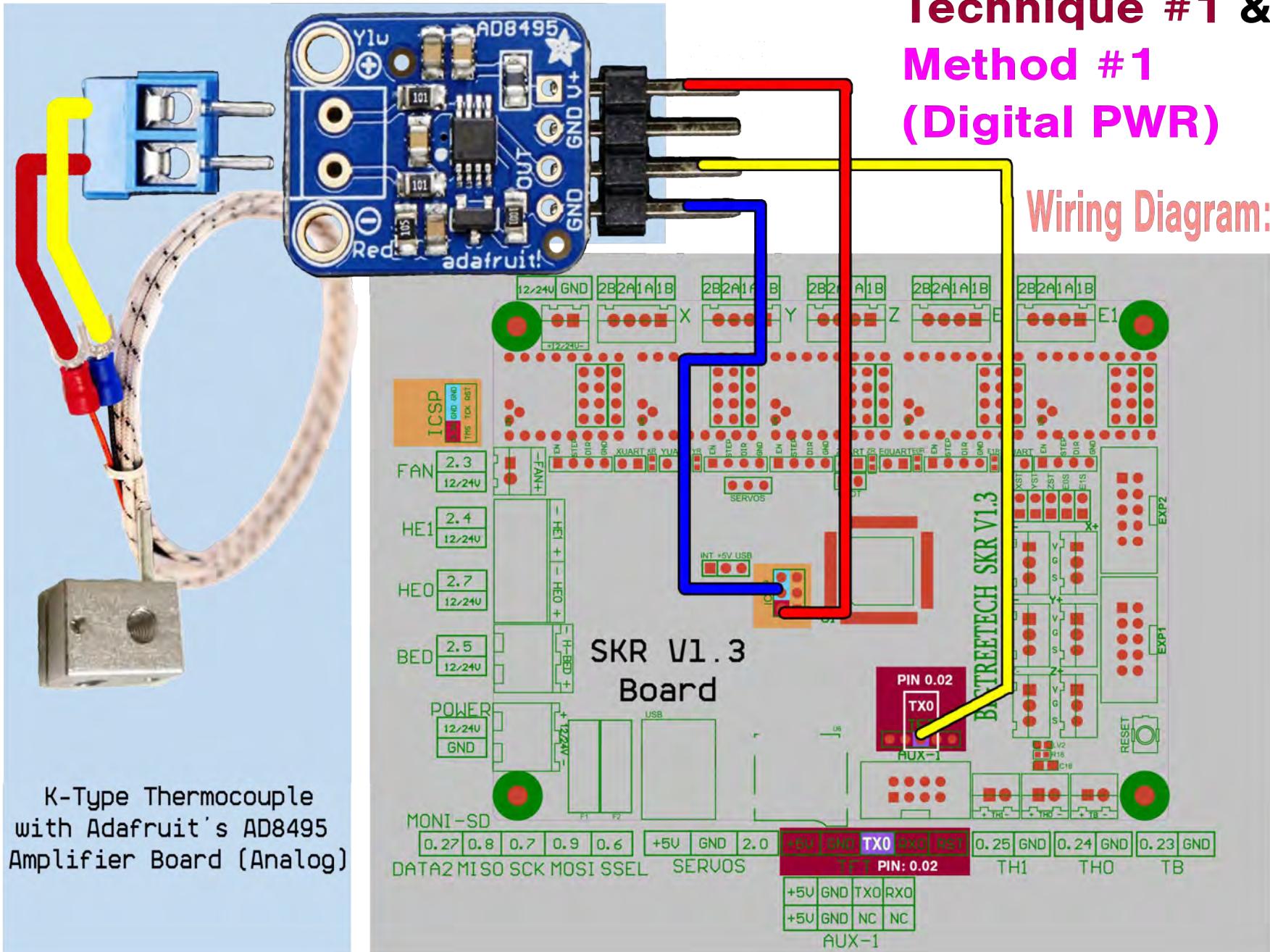
1. the male DuPont Jumper cable now provides you access to the 3.3VDC that is reference to the analog ground (GND\_ANALOG on schematic). Therefore, use the male DuPont Jumper cable to power the AD8495 amplifier board. Place the male DuPont Jumper cable onto the 3.3VDC input of the AD8495 amplifier board (+V).
2. the ground located on the TH0 connector is your analog ground (GND\_ANALOG on schematic). Connect the TH0 connector's ground to the AD8495 amplifier board's ground PIN (0V).
3. **DO NOT use the ground from the ICSP header (GND on the schematic – this is digital ground you want analog ground!) for this method.**
4. if you use this method (**Method #2**) refer to:
  1. **Technique #2 & Method #2 (Analog PWR) wiring diagram** for connecting up your PT100 to the TH0 connector.  
**This is the preferred and recommend way to connect up the AD8495 amplifier board.**

- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #1 & Method #1 (Digital PWR) wiring diagram for connecting up your K-Type Thermocouple to the TFT header:

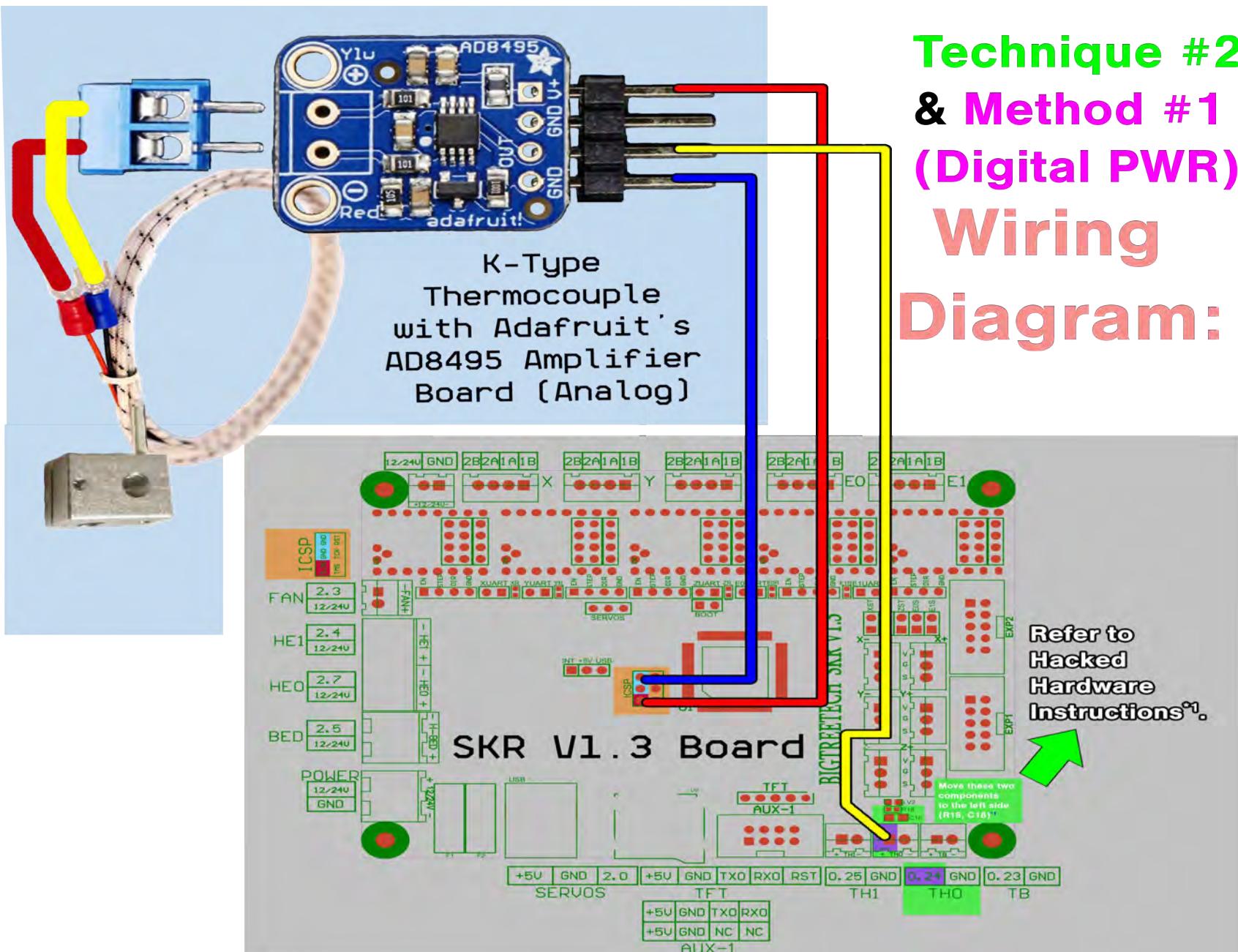


- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #2 & Method #1 (Digital PWR) wiring diagram for connecting up your K-Type Thermocouple to the TH0 connector:



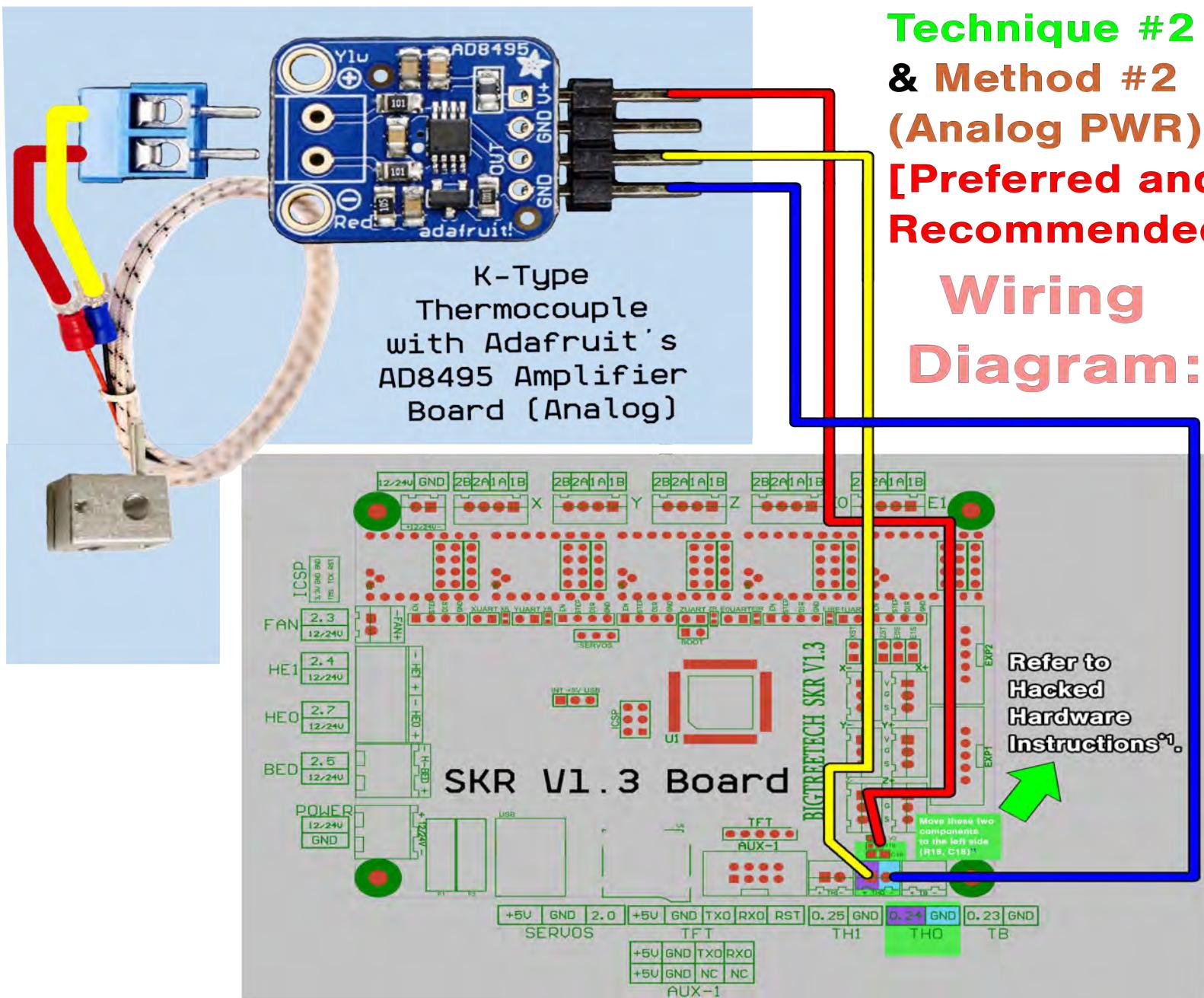
- Go to the next page.

## APPENDIX L

## Connecting SKR V1.3 with K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

Technique #2 & Method #2 (Analog PWR) wiring diagram for connecting up your K-Type Thermocouple to the TH0 connector:

This is the preferred and recommended way to connect up the amplifier board.



- Go to the next page.

\*1 <https://www.thingiverse.com/thing:3603432>

## APPENDIX L

### Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

If you are using **Technique #2** goto the pages

titled "Marlin 2.0x Firmware Setup via **Technique #2**".

If you are using **Technique #1** goto the next page.

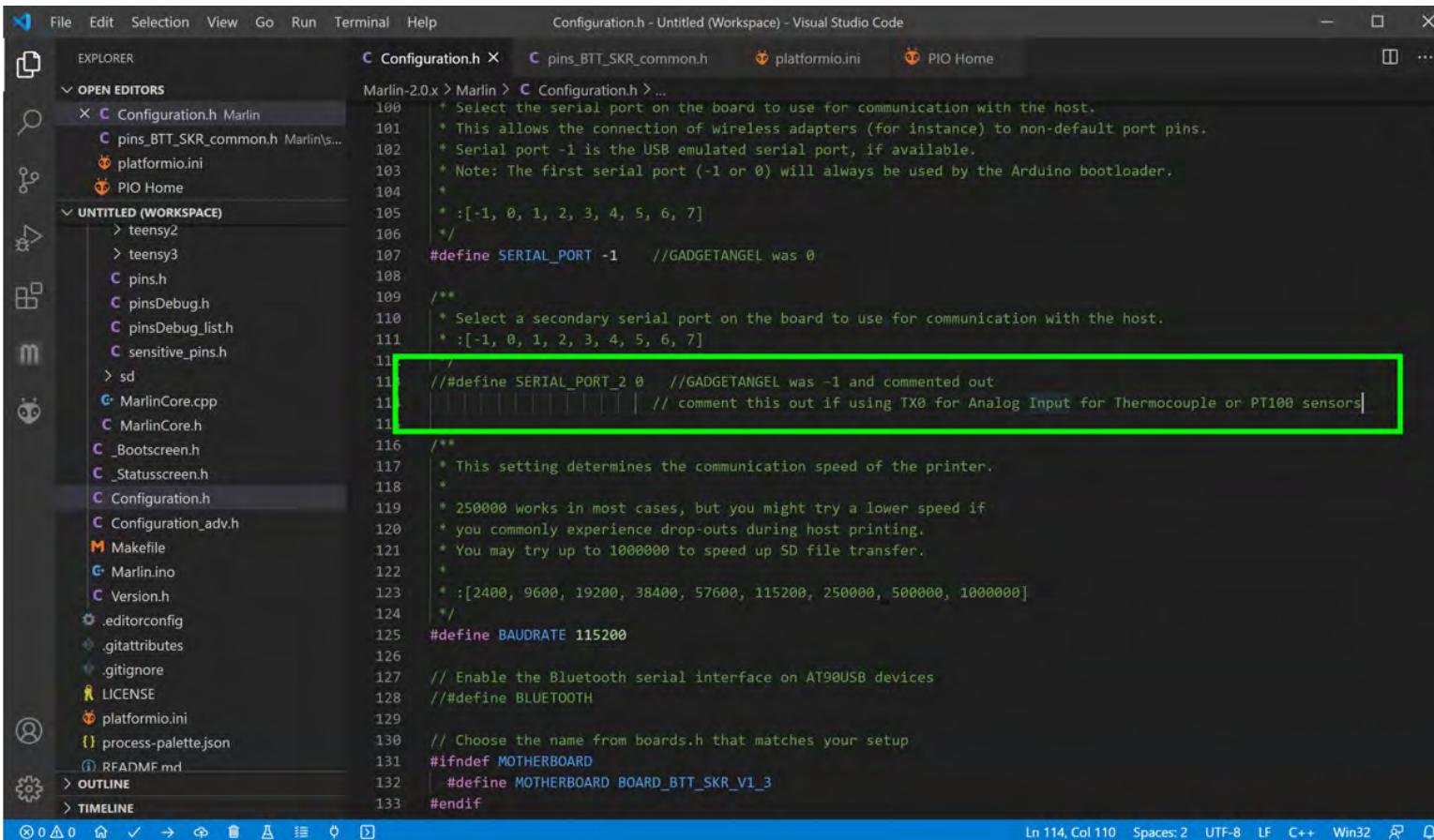
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #1](#)

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- **disable** the Marlin variable **SERIAL\_PORT\_2** or that you comment out the line by inserting two forward slashes ('//') at the beginning of the line.



The screenshot shows the Visual Studio Code interface with the "Configuration.h" file open. The code editor displays the following snippet of code:

```

100 * Select the serial port on the board to use for communication with the host.
101 * This allows the connection of wireless adapters (for instance) to non-default port pins.
102 * Serial port -1 is the USB emulated serial port, if available.
103 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
104 *
105 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
106 */
107 #define SERIAL_PORT -1 //GADGETANGEL was 0
108 /**
109 * Select a secondary serial port on the board to use for communication with the host.
110 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
111 */
112 // #define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out
113 // comment this out if using TX0 for Analog Input for Thermocouple or PT100 sensors
114 */
115 /**
116 * This setting determines the communication speed of the printer.
117 *
118 * 250000 works in most cases, but you might try a lower speed if
119 * you commonly experience drop-outs during host printing.
120 * You may try up to 1000000 to speed up SD file transfer.
121 *
122 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
123 */
124 #define BAUDRATE 115200
125
126 // Enable the Bluetooth serial interface on AT90USB devices
127 //#define BLUETOOTH
128
129 // Choose the name from boards.h that matches your setup
130 #ifndef MOTHERBOARD
131 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
132#endif
133#endif

```

A green rectangular box highlights the line "#define SERIAL\_PORT\_2 0" and the line immediately below it, which is a comment. Both lines are preceded by two forward slashes (//), indicating they are now disabled or commented out.

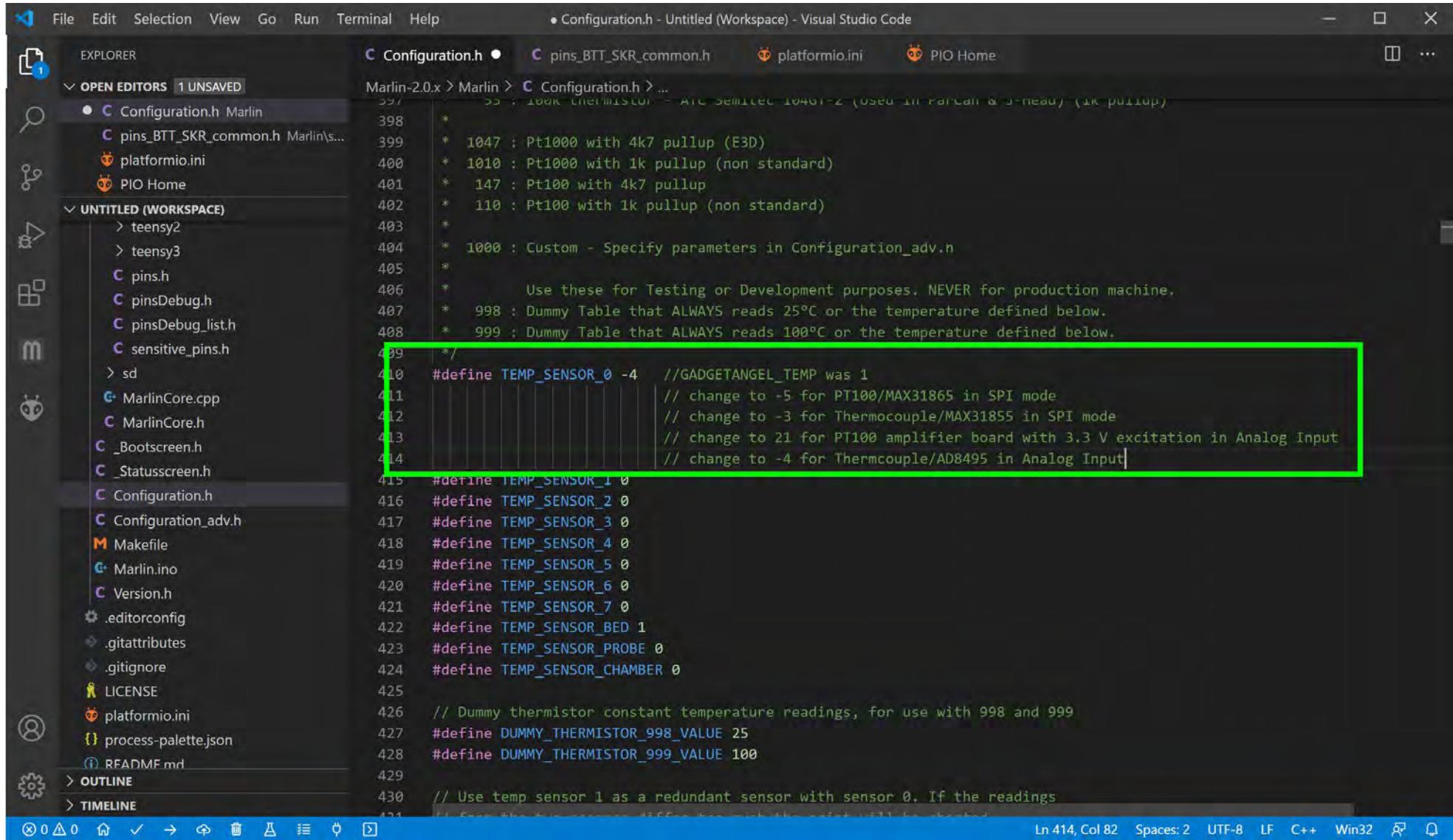
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via **Technique #1**

- change the Marlin variable **TEMP\_SENSOR\_0** to **-4**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like Configuration.h, pins\_BTT\_SKR\_common.h, platformio.ini, and PIO Home.
- Editor:** Displays the `Configuration.h` file content. A specific line of code is highlighted with a green rectangle:
 

```
#define TEMP_SENSOR_0 -4 //GADGETANGEL_TEMP was 1
```
- Status Bar:** Shows the line number (Ln 414, Col 82), spaces used (Spaces: 2), encoding (UTF-8), file type (C++), and operating system (Win32).

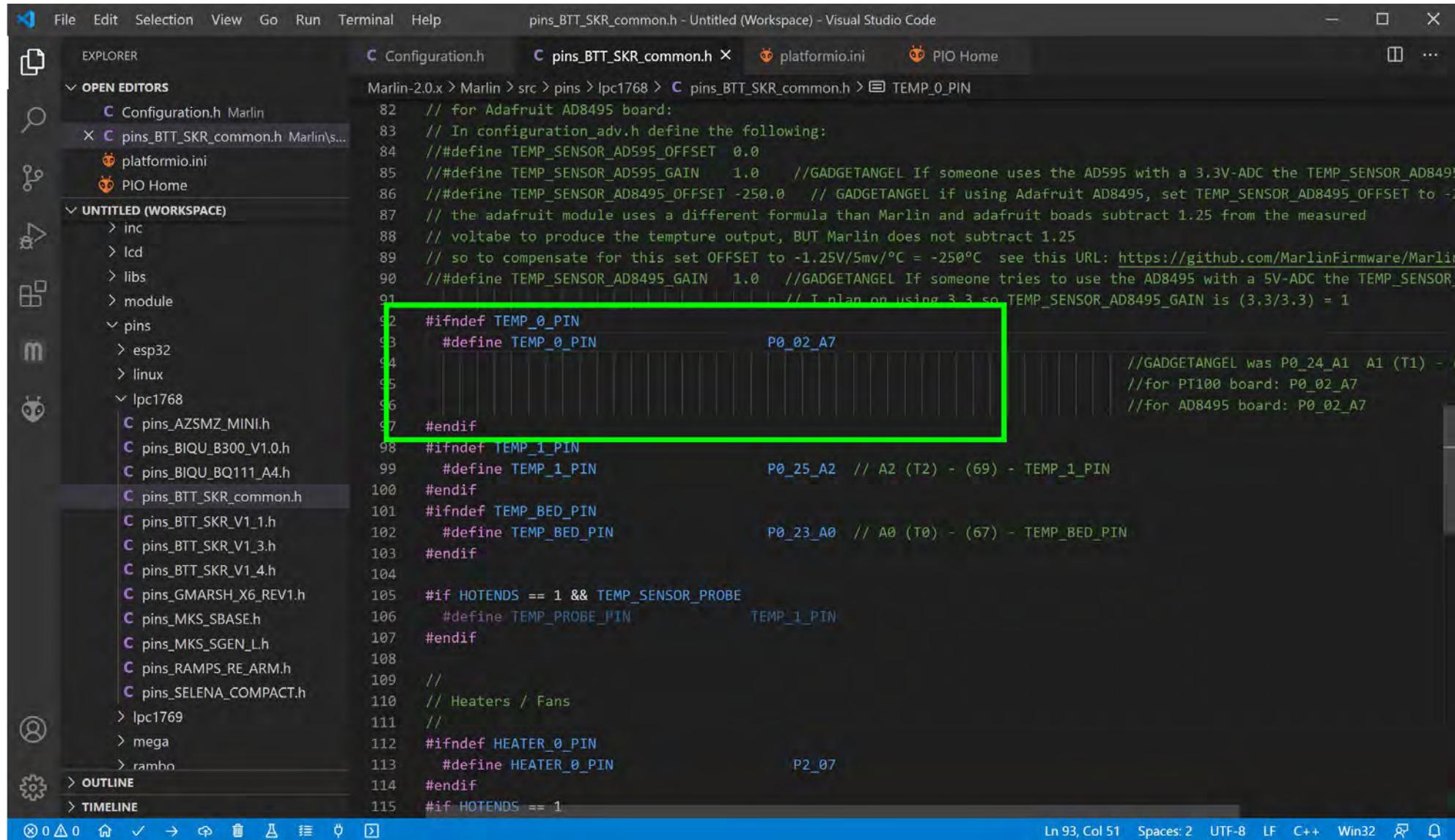
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via **Technique #1**

- change the Marlin variable **TEMP\_0\_PIN** to **P0\_02\_A7**



The screenshot shows the Visual Studio Code interface with the file `pins_BTT_SKR_common.h` open. The code editor displays the following snippet:

```

82 // for Adafruit AD8495 board:
83 // In configuration_adv.h define the following:
84 //#define TEMP_SENSOR_AD595_OFFSET 0.0
85 //#define TEMP_SENSOR_AD595_GAIN 1.0 //GADGETANGEL If someone uses the AD595 with a 3.3V-ADC the TEMP_SENSOR_AD8495
86 //#define TEMP_SENSOR_AD8495_OFFSET -250.0 // GADGETANGEL if using Adafruit AD8495, set TEMP_SENSOR_AD8495_OFFSET to -250.0
87 // the adafruit module uses a different formula than Marlin and adafruit boards subtract 1.25 from the measured
88 // voltage to produce the temperature output, BUT Marlin does not subtract 1.25
89 // so to compensate for this set OFFSET to -1.25V/5mV/°C = -250°C see this URL: https://github.com/MarlinFirmware/Marlin
90 //#define TEMP_SENSOR_AD8495_GAIN 1.0 //GADGETANGEL If someone tries to use the AD8495 with a 5V-ADC the TEMP_SENSOR_
91 // I plan on using 3.3 so TEMP_SENSOR_AD8495_GAIN is (3.3/3.3) = 1
92 #ifndef TEMP_0_PIN
93 #define TEMP_0_PIN P0_02_A7
94
95
96
97 #endif
98 #ifndef TEMP_1_PIN
99 #define TEMP_1_PIN P0_25_A2 // A2 (T2) - (69) - TEMP_1_PIN
100 #endif
101 #ifndef TEMP_BED_PIN
102 #define TEMP_BED_PIN P0_23_A0 // A0 (T0) - (67) - TEMP_BED_PIN
103 #endif
104
105 #if HOTENDS == 1 && TEMP_SENSOR_PROBE
106 #define TEMP_PROBE_PIN TEMP_1_PIN
107 #endif
108 //
109 // Heaters / Fans
110
111 //
112 #ifndef HEATER_0_PIN
113 #define HEATER_0_PIN P2_07
114 #endif
115 #if HOTENDS == 1

```

A green rectangular box highlights the line `#define TEMP_0_PIN P0_02_A7`. The status bar at the bottom of the code editor shows: Ln 93, Col 51 Spaces: 2 UTF-8 LF C++ Win32.

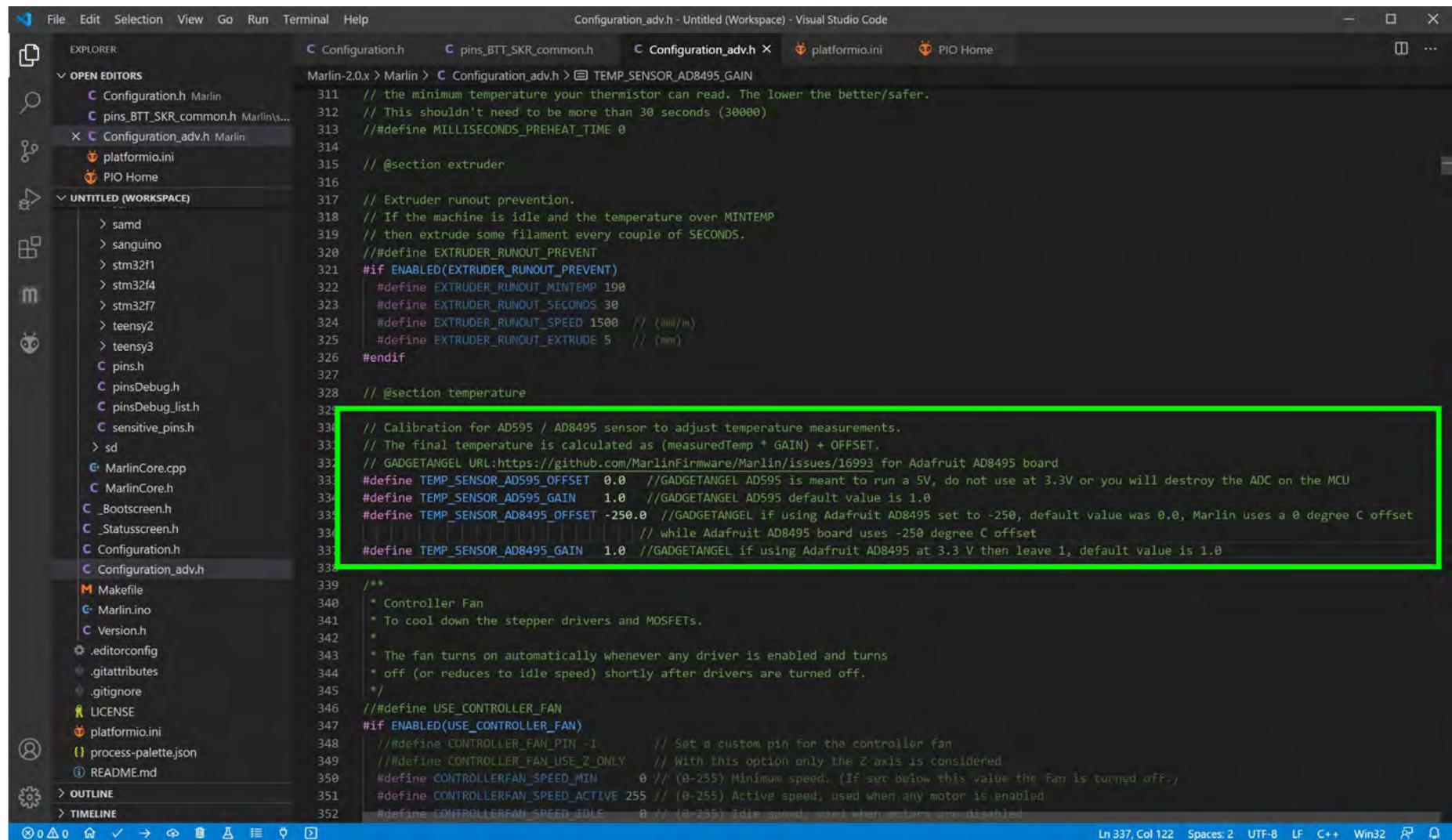
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #1](#)

- change the Marlin variable **TEMP\_SENSOR\_AD8495\_OFFSET** to **-250.0**
- ensure the Marlin variable **TEMP\_SENSOR\_AD8495\_GAIN** is set to **1.0**



The screenshot shows the Visual Studio Code interface with the file `Configuration_adv.h` open. The code editor displays configuration settings for Marlin 2.0.x. A green rectangular box highlights the following section of code:

```

336 // Calibration for AD595 / AD8495 sensor to adjust temperature measurements.
337 // The final temperature is calculated as (measuredTemp * GAIN) + OFFSET.
338 // GADGETANGEL URL: https://github.com/MarlinFirmware/Marlin/issues/16993 for Adafruit AD8495 board
339 #define TEMP_SENSOR_AD595_OFFSET 0.0 // GADGETANGEL AD595 is meant to run a 5V, do not use at 3.3V or you will destroy the ADC on the MCU
340 #define TEMP_SENSOR_AD595_GAIN 1.0 // GADGETANGEL AD595 default value is 1.0
341 #define TEMP_SENSOR_AD8495_OFFSET -250.0 // GADGETANGEL if using Adafruit AD8495 set to -250, default value was 0.0, Marlin uses a 0 degree C offset
342 // while Adafruit AD8495 board uses -250 degree C offset
343 #define TEMP_SENSOR_AD8495_GAIN 1.0 // GADGETANGEL if using Adafruit AD8495 at 3.3 V then leave 1, default value is 1.0

```

- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #1](#)

- Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows files in the workspace, including Configuration.h, pins\_BTT\_SKR\_common.h, platformio.ini, and PIO Home.
- Terminal:** Shows the command-line interface for building and uploading the firmware. The output includes:

```
RAM: [====] 42.3% (used 13840 bytes from 32736 bytes)
Flash: [====] 38.8% (used 184428 bytes from 475136 bytes)
===== [SUCCESS] Took 115.30 seconds =====
```

- Task - PlatformIO: Run:** Shows the upload status for different boards:

| Environment     | Status  | Duration     |
|-----------------|---------|--------------|
| mega2560        | IGNORED |              |
| mega1280        | IGNORED |              |
| rambo           | IGNORED |              |
| FYSETC_F6_13    | IGNORED |              |
| FYSETC_F6_14    | IGNORED |              |
| sanguino644p    | IGNORED |              |
| sanguinol284p   | IGNORED |              |
| melzi           | IGNORED |              |
| melzi_optiboot  | IGNORED |              |
| at90usb1286_cdc | IGNORED |              |
| at90usb1286_dfu | IGNORED |              |
| DUE             | IGNORED |              |
| DUE_USB         | IGNORED |              |
| DUE_debug       | IGNORED |              |
| LPC1768         | SUCCESS | 00:01:55.300 |
| LPC1769         | IGNORED |              |
| STM32F103RC     | IGNORED |              |

- Go to the next page.

## APPENDIX L

### Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

#### Marlin 2.0.x Firmware Setup via [Technique #1](#)

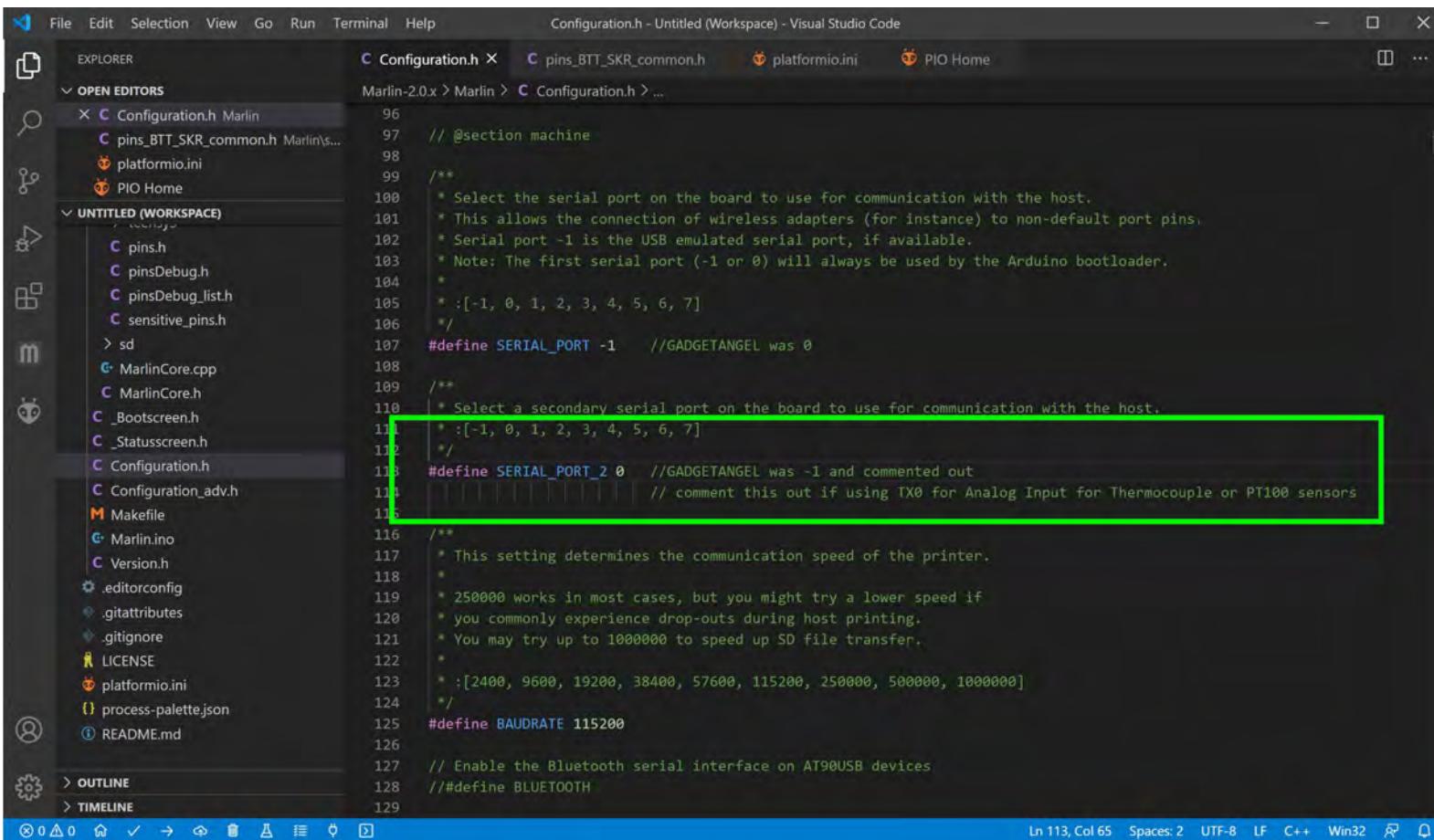
- mggr 'vj g'O ct nkpi ugwpki u'vj g'lcg g'\*cu'cdqxg+''  
wpvkn'vj g'K-Type thermocouple'gpsqt 'ku'  
y qt nkpi 0'Chgt y ctf u'{qw'ctg'htgg'wq'wy gcm'  
cy c{0
- CmHpkj gf 'y kj 'ugwpki 'wr 'vj g'AD8495''  
co r nkgt 'dqctf 'cpf 'the K-Type  
thermocouple'gpuqt using [Technique #1](#)##

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via Technique #2

- Perform the Marlin Firmware steps found in the manual called "[SKR V1.3 Stepper Driver Configuration.pdf](#)" in the section called "APPENDIX C -- The (Latest Release of) Marlin Setup That Is Common To ALL Stepper Motor Drivers". This manual can be found at <https://github.com/GadgetAngel/SKR-V1.3-Stepper-Driver-Jumper-Configuration-Manual/tree/master/CURRENT-Manual>
- **if disabled please enable** the Marlin variable **SERIAL\_PORT\_2** or that you **delete the two forward slashes ("//")** from the beginning of the line.



```

File Edit Selection View Go Run Terminal Help
Configuration.h - Untitled (Workspace) - Visual Studio Code
EXPLORER Configuration.h pins_BTT_SKR_common.h platformio.ini PIO Home
OPEN EDITORS Marlin-2.0.x > Marlin > Configuration.h ...
pins.h pinsDebug.h pinsDebug.list.h sensitive_pins.h
pins_BTT_SKR_common.h Marlin\...
platformio.ini
PIO Home
UNTITLED (WORKSPACE)
pins.h pinsDebug.h pinsDebug.list.h sensitive_pins.h
sd
MarlinCore.cpp
MarlinCore.h
_Bootscreen.h
_Statusscreen.h
Configuration.h
Configuration_adv.h
Makefile
Marlin.ino
Version.h
.editorconfig
.gitattributes
.gitignore
LICENSE
platformio.ini
process-palette.json
README.md
OUTLINE
TIMELINE
Ln 113, Col 65 Spaces: 2 UTF-8 LF C++ Win32 ⚡ ⌂
1 Configuration.h X 2 pins_BTT_SKR_common.h X 3 platformio.ini 4 PIO Home
96 // @section machine
97 /**
98 * Select the serial port on the board to use for communication with the host.
99 * This allows the connection of wireless adapters (for instance) to non-default port pins.
100 * Serial port -1 is the USB emulated serial port, if available.
101 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
102 *
103 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
104 */
105 #define SERIAL_PORT -1 //GADGETANGEL was 0
106 /**
107 * Select a secondary serial port on the board to use for communication with the host.
108 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
109 */
110 #define SERIAL_PORT_2 0 //GADGETANGEL was -1 and commented out
111 // comment this out if using TX0 for Analog Input for Thermocouple or PT100 sensors
112 /**
113 * This setting determines the communication speed of the printer.
114 * 250000 works in most cases, but you might try a lower speed if
115 * you commonly experience drop-outs during host printing.
116 * You may try up to 1000000 to speed up SD file transfer.
117 *
118 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
119 */
120 #define BAUDRATE 115200
121 /**
122 * Enable the Bluetooth serial interface on AT90USB devices
123 // #define BLUETOOTH
124
125
126
127
128
129

```

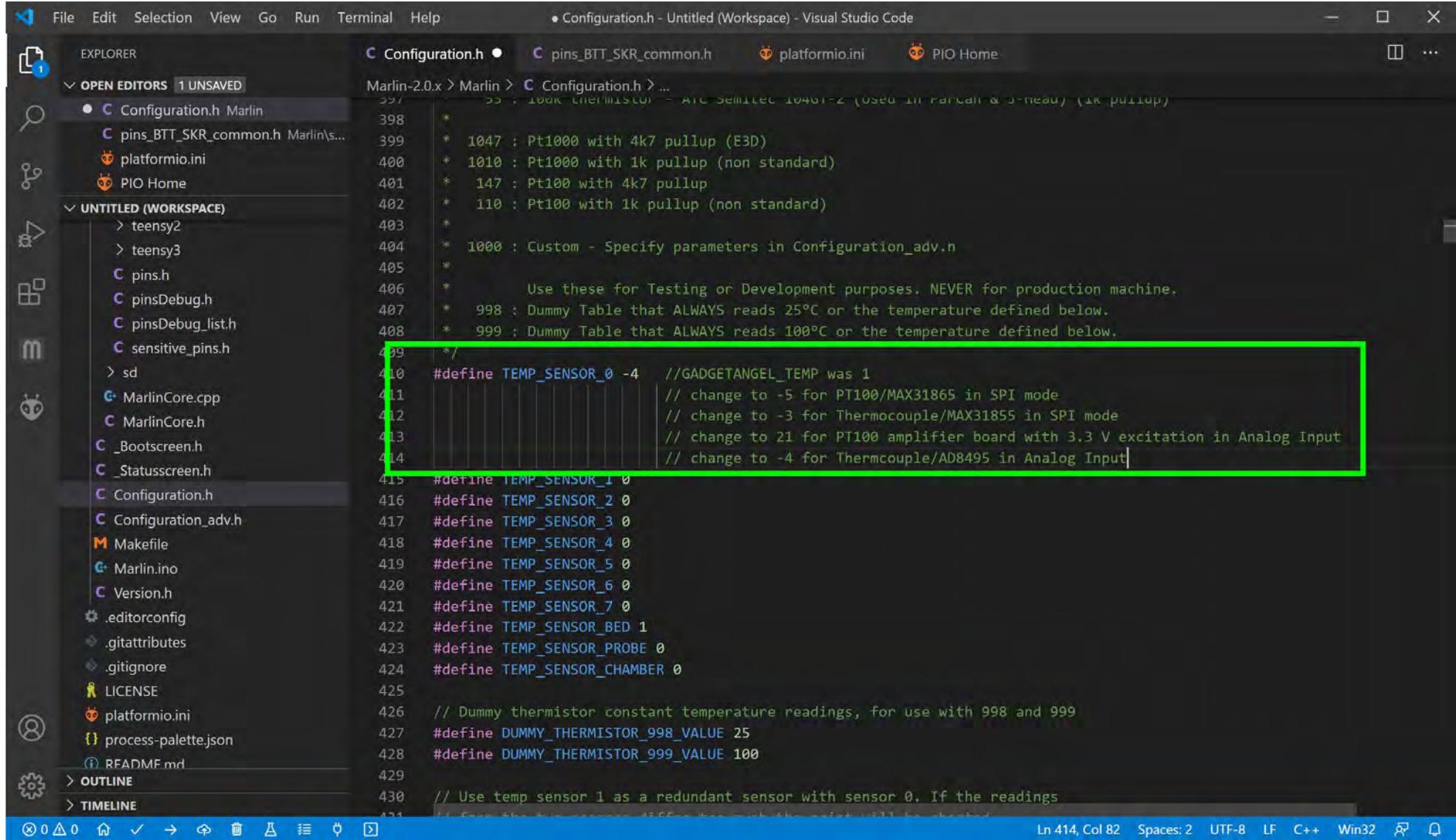
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #2](#)

- change the Marlin variable **TEMP\_SENSOR\_0** to **-4**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like Configuration.h, pins\_BTT\_SKR\_common.h, platformio.ini, and PIO Home.
- Editor:** The Configuration.h file is open, showing C code. A specific line is highlighted with a green rectangle:
 

```
#define TEMP_SENSOR_0 -4 //GADGETANGEL_TEMP was 1
```
- Status Bar:** Shows the line number (Ln 414, Col 82), character count (Spaces: 2), encoding (UTF-8), file type (LF C++), and operating system (Win32).

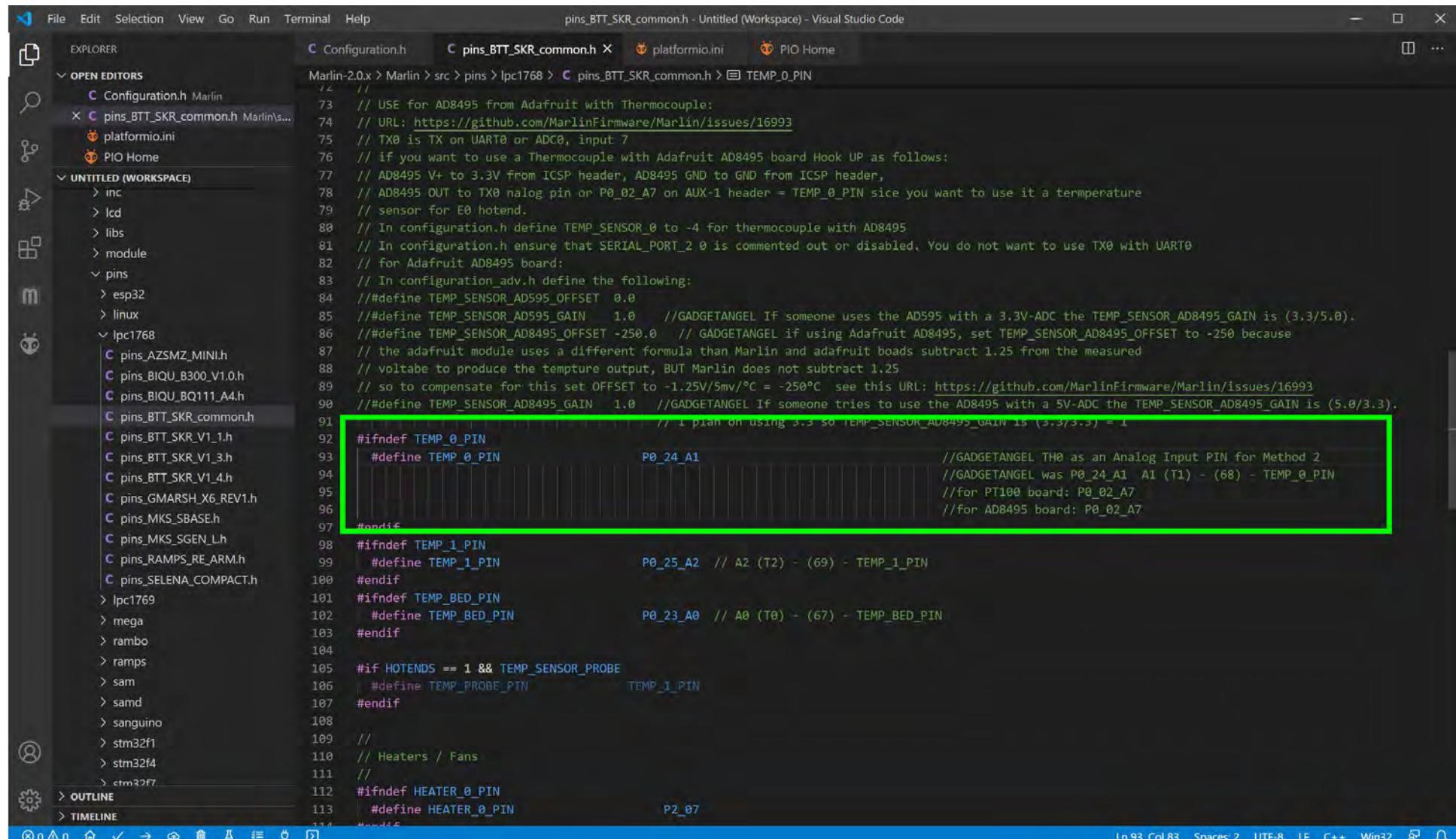
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #2](#)

- change the Marlin variable **TEMP\_0\_PIN** to **P0\_24\_A1**



```

File Edit Selection View Go Run Terminal Help pins_BTT_SKR_common.h - Untitled (Workspace) - Visual Studio Code
EXPLORER Configuration.h pins_BTT_SKR_common.h X platformio.ini PIO Home
OPEN EDITORS Marlin-2.0.x > Marlin > src > pins > lpc1768 > pins_BTT_SKR_common.h > TEMP_0_PIN
14 //
73 // USE for AD8495 from Adafruit with Thermocouple:
74 // URL: https://github.com/MarlinFirmware/Marlin/issues/16993
75 // TX0 is TX on UART0 or ADC0, input 7
76 // if you want to use a Thermocouple with Adafruit AD8495 board Hook UP as follows:
77 // AD8495 V+ to 3.3V from ICSP header, AD8495 GND to GND from ICSP header,
78 // AD8495 OUT to TX0 analog pin or P0_02_A7 on AUX-1 header = TEMP_0_PIN since you want to use it a temperature
79 // sensor for E0 hotend.
80 // In configuration.h define TEMP_SENSOR_0 to -4 for thermocouple with AD8495
81 // In configuration.h ensure that SERIAL_PORT_2 0 is commented out or disabled. You do not want to use TX0 with UART0
82 // for Adafruit AD8495 board:
83 // In configuration_adv.h define the following:
84 //#define TEMP_SENSOR_AD595_OFFSET 0.0
85 //#define TEMP_SENSOR_AD595_GAIN 1.0 //GADGETANGEL If someone uses the AD595 with a 3.3V-ADC the TEMP_SENSOR_AD8495_GAIN is (3.3/5.0).
86 //#define TEMP_SENSOR_AD8495_OFFSET -250.0 // GADGETANGEL if using Adafruit AD8495, set TEMP_SENSOR_AD8495_OFFSET to -250 because
87 // the adafruit module uses a different formula than Marlin and adafruit boards subtract 1.25 from the measured
88 // voltage to produce the tempture output, BUT Marlin does not subtract 1.25
89 // so to compensate for this set OFFSET to -1.25V/mv/°C = -250°C see this URL: https://github.com/MarlinFirmware/Marlin/issues/16993
90 //#define TEMP_SENSOR_AD8495_GAIN 1.0 //GADGETANGEL If someone tries to use the AD8495 with a 5V-ADC the TEMP_SENSOR_AD8495_GAIN is (5.0/3.3).
91 // I plan on using 3.3 so TEMP_SENSOR_AD8495_GAIN is (3.3/3.3) = 1
92 #ifndef TEMP_0_PIN
93 #define TEMP_0_PIN P0_24_A1 //GADGETANGEL TH0 as an Analog Input PIN for Method 2
94 //GADGETANGEL was P0_24_A1 A1 (T1) - (68) - TEMP_0_PIN
95 //for PT100 board: P0_02_A7
96 //for AD8495 board: P0_02_A7
97 #endif
98 #ifndef TEMP_1_PIN
99 #define TEMP_1_PIN P0_25_A2 // A2 (T2) - (69) - TEMP_1_PIN
100 #endif
101 #ifndef TEMP_BED_PIN
102 #define TEMP_BED_PIN P0_23_A0 // A0 (T0) + (67) - TEMP_BED_PIN
103 #endif
104
105 #if HOTENDS == 1 && TEMP_SENSOR_PROBE
106 #define TEMP_PROBE_PIN TEMP_1_PIN
107 #endif
108 //
109 // Heaters / Fans
110 //
111 #
112 #ifndef HEATER_0_PIN
113 #define HEATER_0_PIN P2_07
114 #
115

```

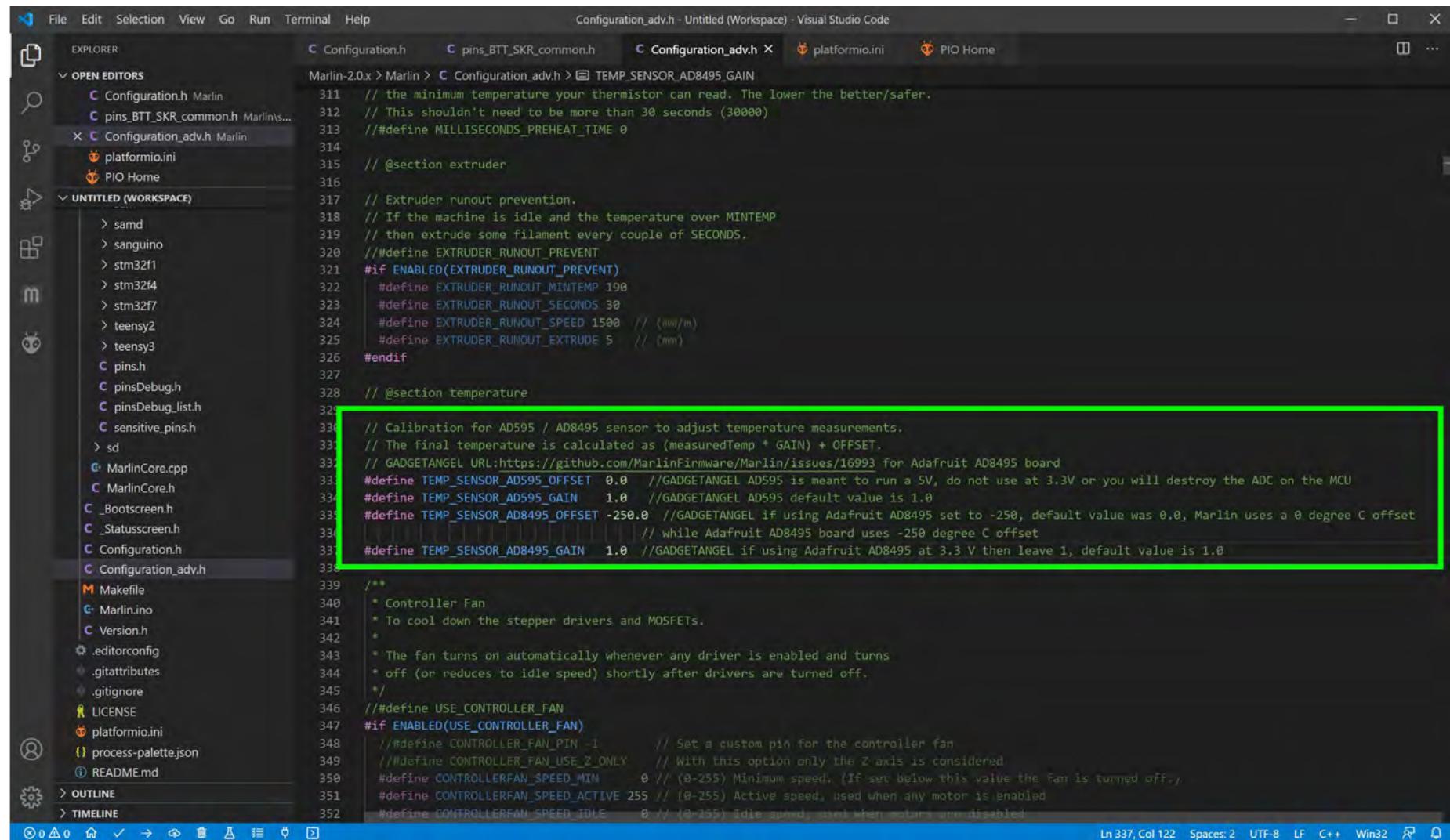
- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #2](#)

- change the Marlin variable **TEMP\_SENSOR\_AD8495\_OFFSET** to **-250.0**
- ensure the Marlin variable **TEMP\_SENSOR\_AD8495\_GAIN** is set to **1.0**



The screenshot shows the Visual Studio Code interface with the "Configuration\_adv.h" file open. The code editor displays configuration settings for the Marlin firmware. A specific section of the code is highlighted with a green rectangular selection:

```

311 // the minimum temperature your thermistor can read. The lower the better/safer.
312 // This shouldn't need to be more than 30 seconds (30000)
313 //define MILLISECONDS_PREHEAT_TIME 0
314
315 // @section extruder
316
317 // Extruder runout prevention.
318 // If the machine is idle and the temperature over MINTEMP
319 // then extrude some filament every couple of SECONDS.
320 //define EXTRUDER_RUNOUT_PREVENT
321 #if ENABLED(EXTRUDER_RUNOUT_PREVENT)
322 #define EXTRUDER_RUNOUT_MINTEMP 190
323 #define EXTRUDER_RUNOUT_SECONDS 30
324 #define EXTRUDER_RUNOUT_SPEED 1500 // (mm/m)
325 #define EXTRUDER_RUNOUT_EXTRUDE 5 // (mm)
326#endif
327
328 // @section temperature
329
330 // Calibration for AD595 / AD8495 sensor to adjust temperature measurements.
331 // The final temperature is calculated as (measuredTemp * GAIN) + OFFSET.
332 // GADGETANGEL URL:https://github.com/MarlinFirmware/Marlin/issues/16993 for Adafruit AD8495 board
333#define TEMP_SENSOR_AD595_OFFSET 0.0 //GADGETANGEL AD595 is meant to run a 5V, do not use at 3.3V or you will destroy the ADC on the MCU
334#define TEMP_SENSOR_AD595_GAIN 1.0 //GADGETANGEL AD595 default value is 1.0
335#define TEMP_SENSOR_AD8495_OFFSET -250.0 //GADGETANGEL if using Adafruit AD8495 set to -250, default value was 0.0, Marlin uses a 0 degree C offset
336// while Adafruit AD8495 board uses -250 degree C offset
337#define TEMP_SENSOR_AD8495_GAIN 1.0 //GADGETANGEL if using Adafruit AD8495 at 3.3 V then leave 1, default value is 1.0
338
339 /**
340 * Controller Fan
341 * To cool down the stepper drivers and MOSFETs.
342 *
343 * The fan turns on automatically whenever any driver is enabled and turns
344 * off (or reduces to idle speed) shortly after drivers are turned off.
345 */
346#ifndef USE_CONTROLLER_FAN
347 #if ENABLED(USE_CONTROLLER_FAN)
348 //define CONTROLLER_FAN_PIN -1 // Set a custom pin for the controller fan
349 //define CONTROLLER_FAN_USE_Z_ONLY // With this option only the Z axis is considered
350 #define CONTROLLERFAN_SPEED_MIN 0 // (0-255) Minimum speed. (If set below this value the Fan is turned off.)
351 #define CONTROLLERFAN_SPEED_ACTIVE 255 // (0-255) Active speed, used when any motor is enabled
352 #define CONTROLLERFAN_SPEED_IDLE 0 // (0-255) Idle speed, used when motors are disabled

```

- Go to the next page.

## APPENDIX L

# Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

## Marlin 2.0.x Firmware Setup via [Technique #2](#)

- Now that the PT100 temperature sensor is wired to the SKR V1.3 board and the additional Marlin Firmware statement are setup, compile the Marlin Firmware with the added changes and upload it to the SKR V1.3 board. Reboot the SKR V1.3 board to enable the new firmware to be loaded.

```

Configuration.h - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER Configuration.h X pins_BTT_SKR_common.h platformio.ini PIO Home
OPEN EDITORS Marlin-2.0.x > Marlin > Configuration.h > SERIAL_PORT_2
 Configuration.h Marlin pins_BTT_SKR_common.h Marlin\...
 pins.h platformio.ini
 PIO Home
UNTITLED (WORKSPACE)
 teensy2
 teensy3
 pins.h
 pinsDebug.h
 pinsDebug_list.h
 sensitive_pins.h
 sd
 MarlinCore.cpp
 MarlinCore.h
 _Bootscreen.h
 _Statusscreen.h
 Configuration.h
 Configuration_adv.h
 Makefile
 Marlin.ino
 Version.h
 .editorconfig
 .gitattributes
 .gitignore
 LICENSE
 platformio.ini
 process-palette.json
 RFADMF.md
 OUTLINE
 TIMELINE
 File Edit Selection View Go Run Terminal Help
 Configuration.h X pins_BTT_SKR_common.h platformio.ini PIO Home
 Marlin-2.0.x > Marlin > Configuration.h > SERIAL_PORT_2
 128 // #define BLUETOOTH
 129
 130 // Choose the name from boards.h that matches your setup
 131 #ifndef MOTHERBOARD
 132 #define MOTHERBOARD BOARD_BTT_SKR_V1_3
 133 #endif
 134
 135 // Name displayed in the LCD "Ready" message and Info menu
 136 #define CUSTOM_MACHINE_NAME "Ender-3"
 137

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Task - PlatformIO: Bu + □ ^ x
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [====] 43.2% (used 14132 bytes from 32736 bytes)
Flash: [====] 42.0% (used 199632 bytes from 475136 bytes)
===== [SUCCESS] Took 114.34 seconds =====
Environment Status Duration
mega2560 IGNORED
mega1280 IGNORED
rambo IGNORED
FYSETC_F6_13 IGNORED
FYSETC_F6_14 IGNORED
sanguino64p IGNORED
sanguino1284p IGNORED
melzi IGNORED
melzi_optiboot IGNORED
at90usb1286_cdc IGNORED
at90usb1286_dfu IGNORED
DUE IGNORED
DUE_USB IGNORED
DUE_debug IGNORED
LPC1768 SUCCESS 00:01:54.338
LPC1769 IGNORED
STM32F103RC IGNORED
Ln 113, Col 65 Spaces: 2 UTF-8 LF C++ Win32

```

- Go to the next page.

## APPENDIX L

### Marlin 2.0.x Firmware Setup for Connecting a K-Type Thermocouple Sensor and Adafruit's AD8495 Amplifier Board

#### Marlin 2.0.x Firmware Setup via [Technique #2](#)

- mggr 'vj g'Octrlp'ugwlpi u'vj g'trc o g'\*cu'cdqxg+'  
wpvkn'vj g'K-Type thermocouple'ugpsqt 'ku'  
y qt m kpi 0'Chgt y ctf u'{qw'ct g'lt gg'vq'w y gcm'  
cy c{ 0
- Cm'Hkpkuj gf 'y kj 'ugwlpi 'wr 'vj g'AD8495''  
co r nkhgt 'dqctf 'cpf 'the K-Type  
thermocouple'ugpuqt using [Technique #2](#)##