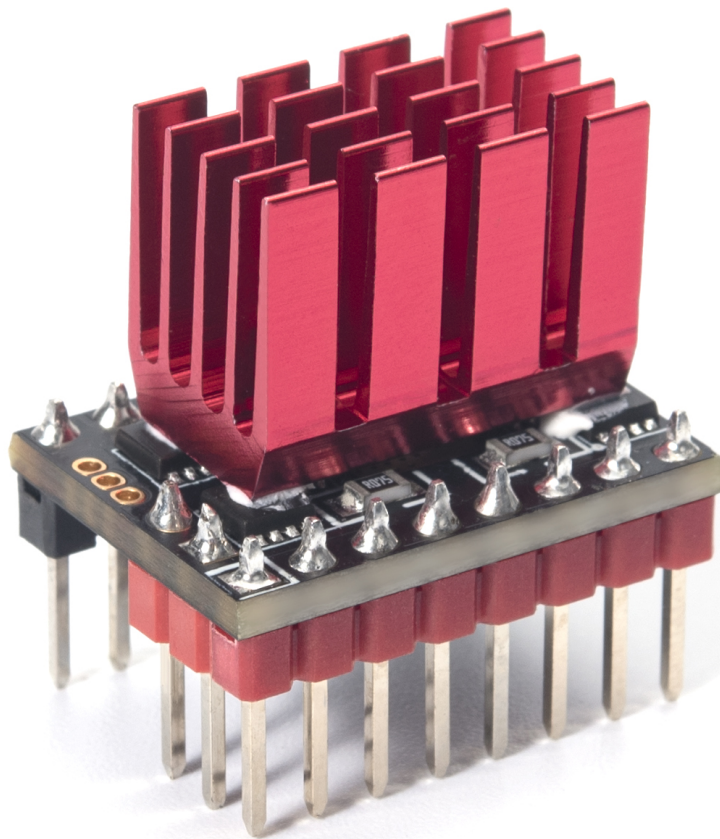


BIGTREE TECH

TMC5160T Pro V1.0

User Manual



Revision Log

Version	Date	Revisions
v1.00	October 18th, 2024	Initial Version

Table of Contents

1. Product Profile	4
1.1. Feature Highlights	4
1.2. Specifications	4
1.3. Peripheral Interface	5
1.3.1. Dimensions	5
1.3.2. Pin Description	6
2. Interface Introduction	7
2.1. Installation and Interface	7
3. Firmware Settings	8
3.1. Marlin	8
4. Cautions	14

1. Product Profile

The TMC5160 is a high-power stepper motor driver control chip that uses external power MOSFETs. It can operate at voltages up to 56V, supporting a wider range of stepper motors and offering higher adaptability.

1.1. Feature Highlights

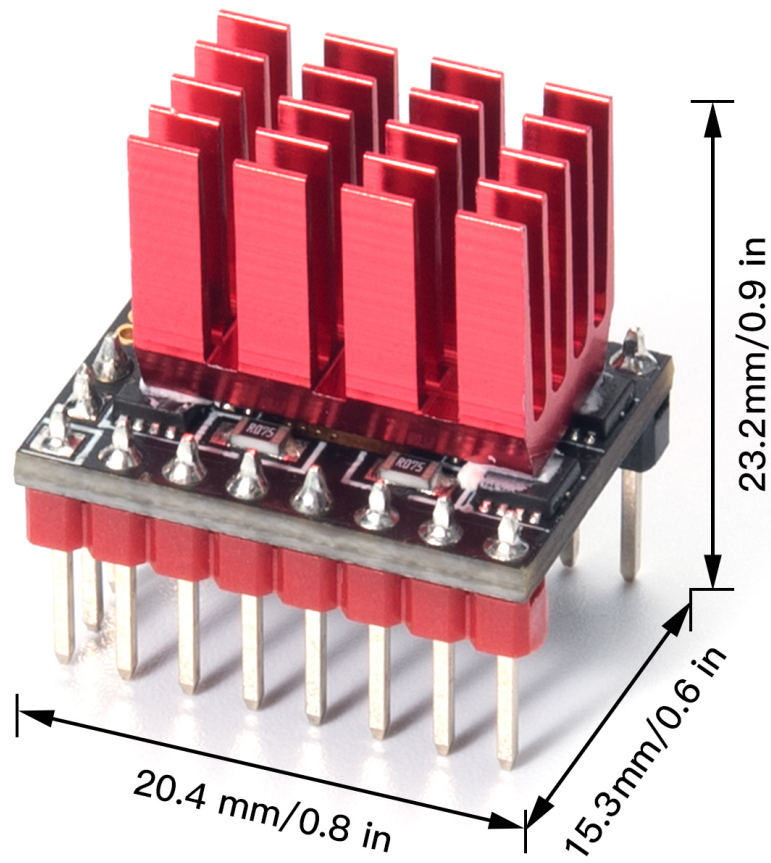
- Utilizes external power MOSFETs to support higher voltages and larger currents.
- Generates significantly less heat compared to drivers such as the 2209 and 2130.
- Delivers greater torque to prevent motor-jitter, reducing the likelihood of missing steps.
- Capable of driving 57 stepper motors.
- Adopts a universal driver board design for higher compatibility across various applications.
- Features a heatsink with enhanced fin design for improved cooling.
- Includes expansion interfaces for DIY enhancements.

1.2. Specifications

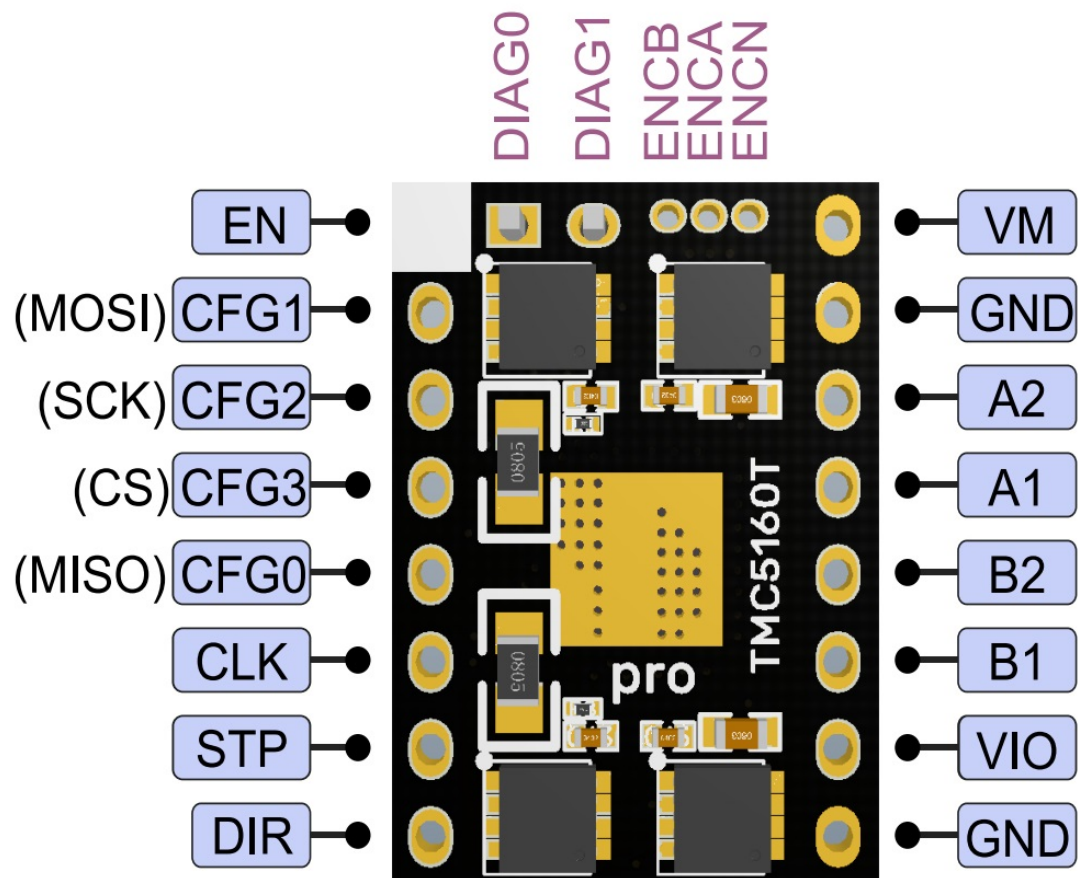
- Dimensions: 20.4mm x 15.3mm x 23.2mm
- Driver Chip: TMC5160-TA
- Input Voltage (VM): 8V-56V (TMC5160T Pro), 8V-24V (TMC5160T)
- Maximum Current: RMS 3.1A, Peak 4.4A (base capacity of 3A maximum)
- Maximum Microstepping: 256 steps
- Operating Mode: SPI

1.3. Peripheral Interface

1.3.1. Dimensions



1.3.2. Pin Description



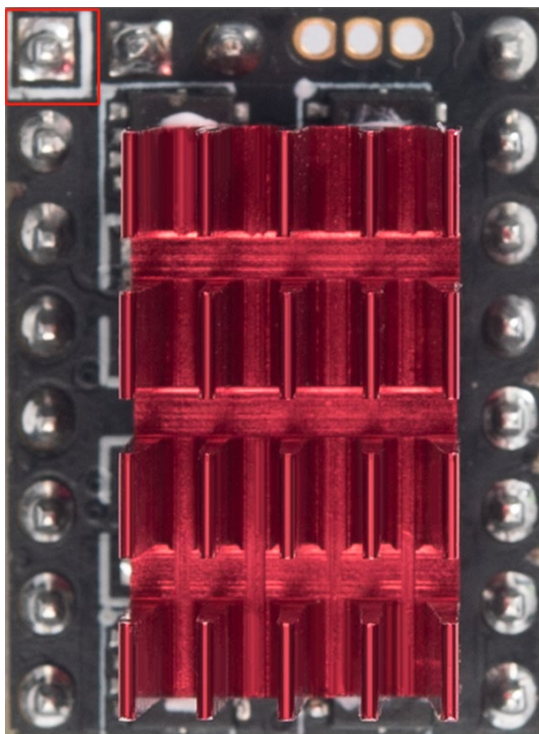
J1	Functions	J2	Functions
1	(EN) Enable	1	(VM) Motor Supply Voltage
2	(SDI/CFG1) Data	2	(GND) Ground
3	(SCK/CFG2) Clock	3	(A2) Phase A
4	(CSN/CFG3) Chip Select	4	(A1) Phase A
5	(SDO/CFG0) Data	5	(B2) Phase B
6	(CLK) External Clock Input	6	(B1) Phase B
7	(STEP) Pulse Input	7	(VIO) Logic Voltage
8	(DIR) Direction Input	8	(GND) Ground

ENCA_DCIN_CFG5	24	24	DI (pd)	Encoder A-channel input (when using internal ramp generator) or DcStep gating input for axis synchronization (SD_MODE=1, SPI_MODE=1) or Configuration input (SPI_MODE=0)
ENCN_DCO_CFG6	25	26	DIO	Encoder N-channel input (SD_MODE=0) or DcStep ready output (SD_MODE=1). With SD_MODE=0, pull to GND or VCC_IO, if the pin is not used for an encoder.
DIAG0_SWN	26	27	DIO (pu+pd)	Diagnostics output DIAG0. Interrupt or STEP output for motion controller (SD_MODE=0, SPI_MODE=1). Use external pullup resistor with 47k or less in open drain mode. Single wire I/O (negative) (only with SD_MODE=0 and SPI_MODE=0)
DIAG1_SWP	27	28	DIO (pd)	Diagnostics output DIAG1. Position compare or DIR output for motion controller (SD_MODE=0, SPI_MODE=1). Use external pullup resistor with 47k or less in open drain mode. Single wire I/O (positive) (only with SD_MODE=0 and SPI_MODE=0)

2. Interface Introduction

2.1. Installation and Interface

The Enable (EN) pin is highlighted in red in the diagram and located inside the marked white box on the driver:



3. Firmware Settings

3.1. Marlin

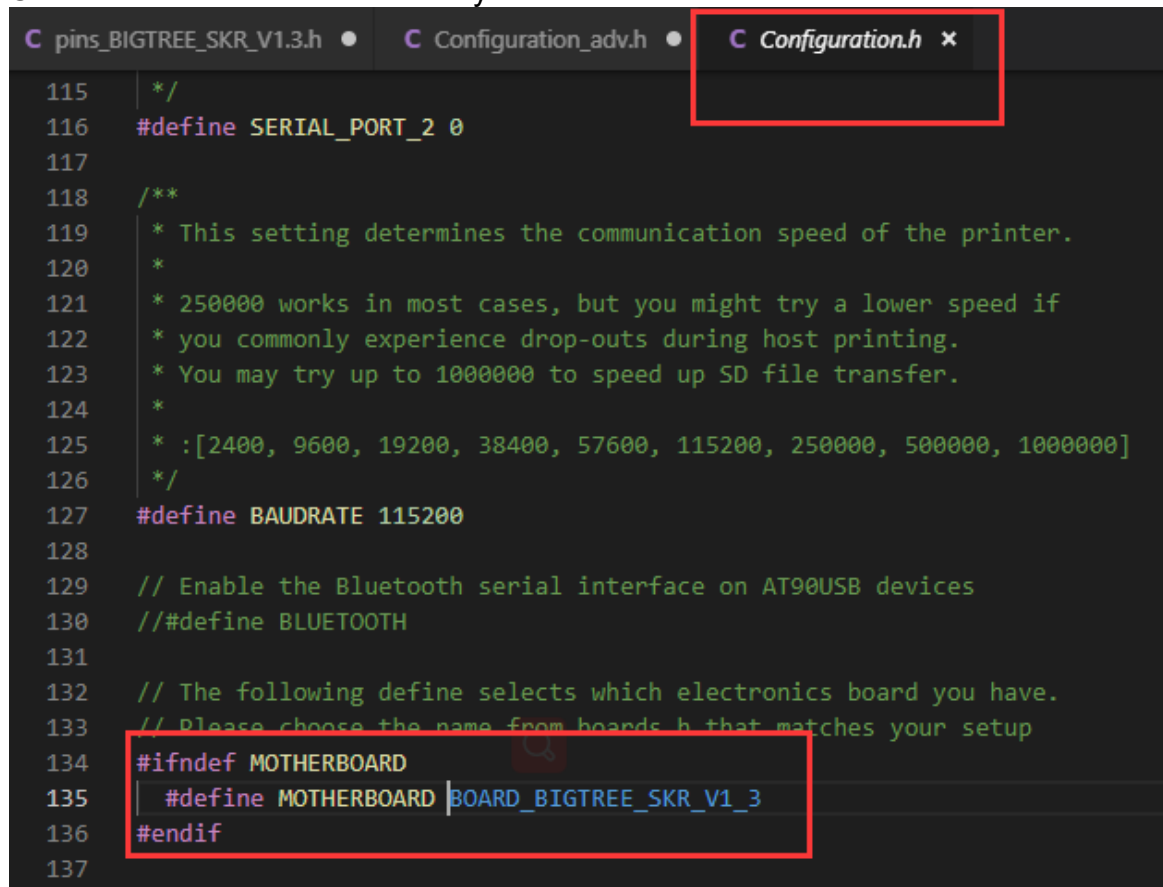
Important: Marlin firmware version 2.0 or above is required for TMC5160's SPI mode.

Step 1:

Open **Configuration.h** in your Marlin 2.0 firmware.

Find **#define MOTHERBOARD XXXXXX**.

Check the **XXXXXX** value. This is your board.



```
115  */
116  #define SERIAL_PORT_2 0
117
118  /**
119   * This setting determines the communication speed of the printer.
120   *
121   * 250000 works in most cases, but you might try a lower speed if
122   * you commonly experience drop-outs during host printing.
123   * You may try up to 1000000 to speed up SD file transfer.
124   *
125   * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
126   */
127  #define BAUDRATE 115200
128
129  // Enable the Bluetooth serial interface on AT90USB devices
130  //#define BLUETOOTH
131
132  // The following define selects which electronics board you have.
133  // Please choose the name from boards.h that matches your setup
134  #ifndef MOTHERBOARD
135    #define MOTHERBOARD BOARD_BIGTREE_SKR_V1_3
136  #endif
137
```

Step 2:

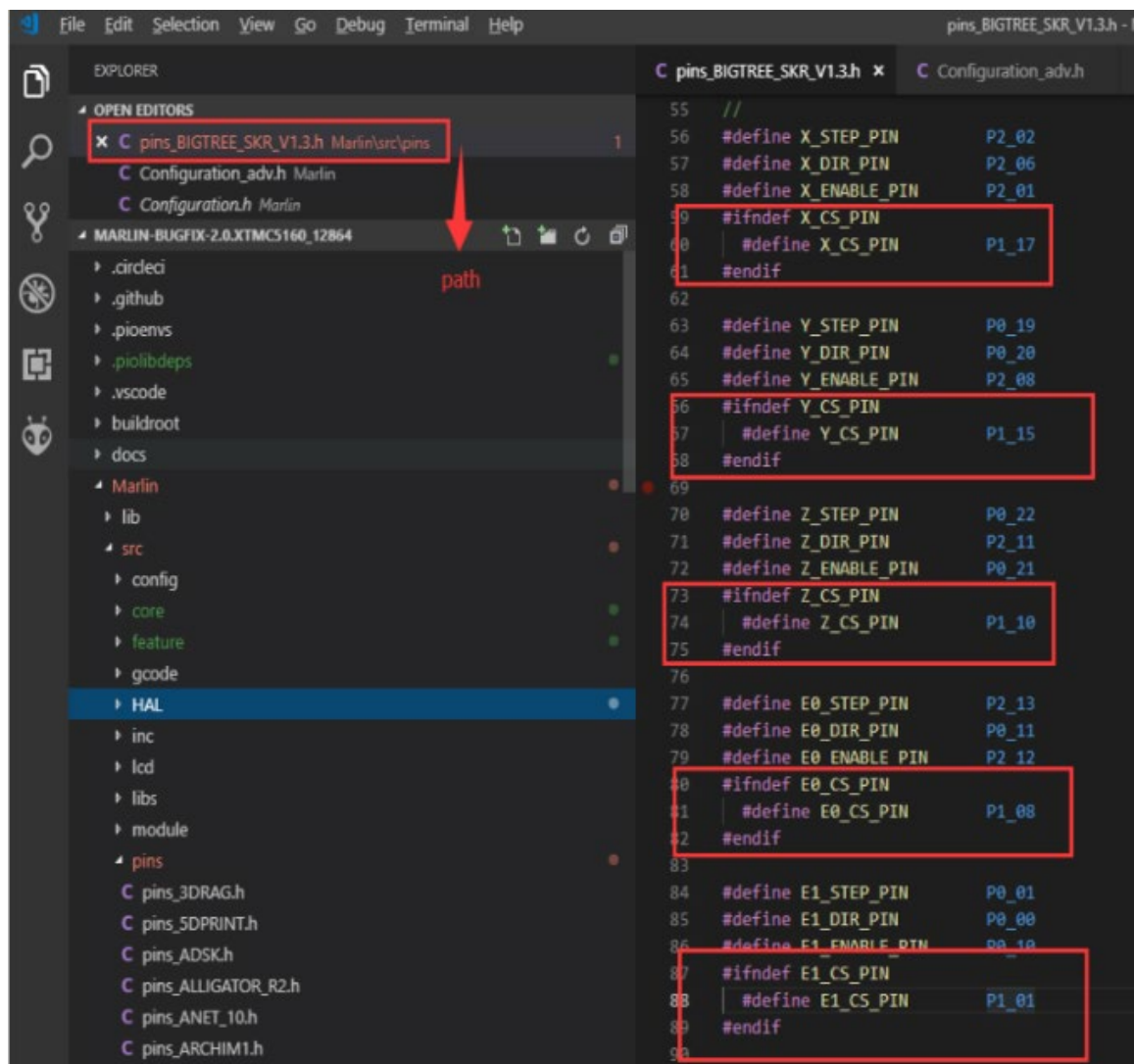
Go to the **Marlin\src\pins** directory.

Open the **pins_XXXXXX.h** file that matches your board. (Remember, XXXXXX is your board model from Step 1).

Find these lines:

- X_CS_PIN
- Y_CS_PIN
- Z_CS_PIN
- E0_CS_PIN

Change the pin numbers to the ones which you are using.



Step 3:

Stay in that same **pins_XXXXXX.h** file. (The one from Step 2).

Find these lines:

#define TMC_SW_MOSI XXX

#define TMC_SW_MISO XXX

#define TMC_SW_SCK XXX

Replace those XXX placeholders with the correct pin numbers for your setup.

```
91 //
92 // Software SPI pins for TMC2130 stepper drivers
93 //
94 #if ENABLED(TMC_USE_SW_SPI)
95     #define TMC_SW_MOSI      P4_28
96     #define TMC_SW_MISO      P0_05
97     #define TMC_SW_SCK       P0_04
98
99 #endif
100
101 /* #define TMC_SW_MISO      P4_28
102    #define TMC_SW_SCK       P0_05
103    #define TMC_SW_MOSI      P0_04
104    */
```

Step 4:

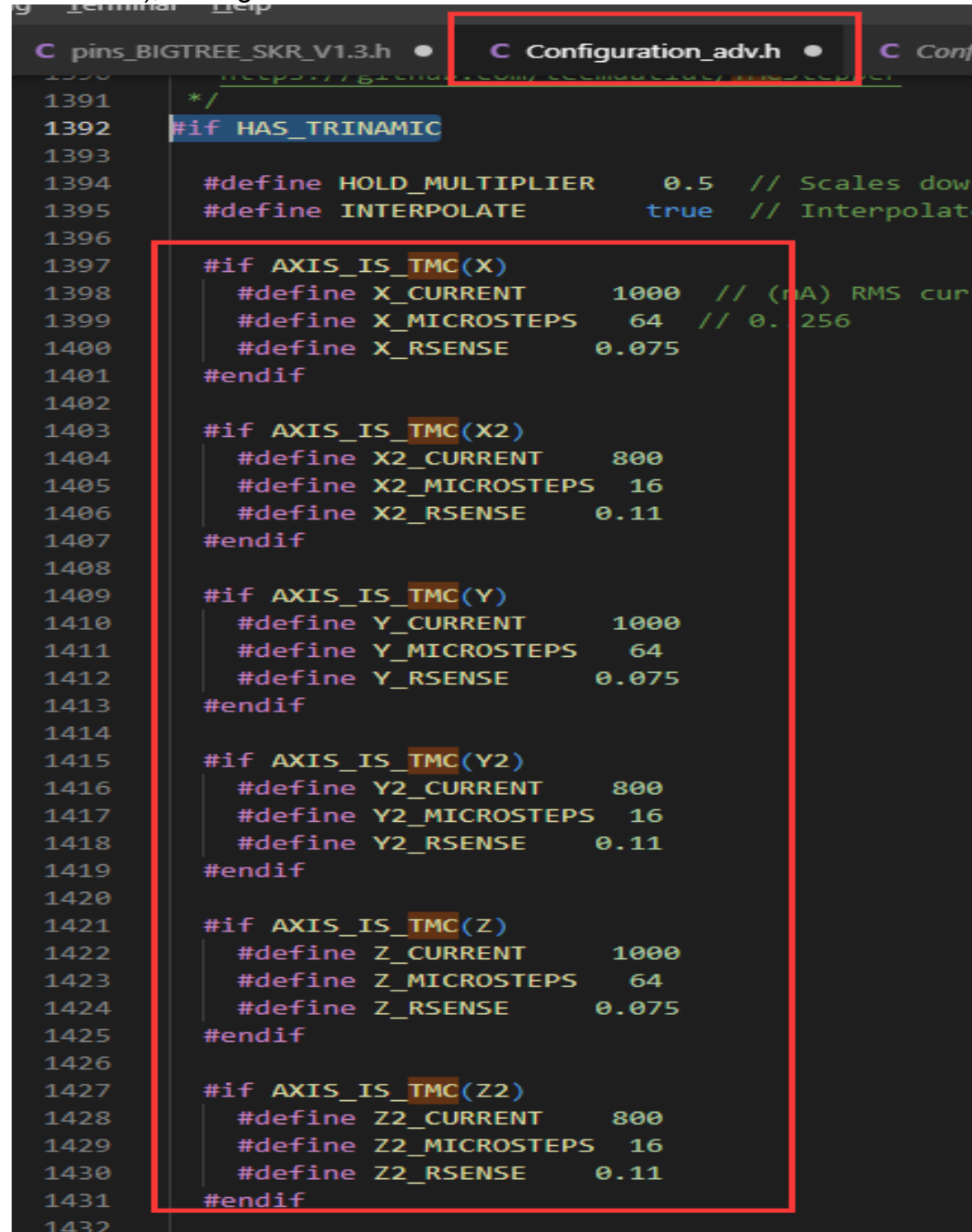
Open the **Configuration_adv.h** file. Find the line **#define TMC_USE_SW_SPI**.

Remove the **//** at the beginning of the line.

```
C pins_BIGTREE_SKR_V1.3.h • C Configuration_adv.h • C Configuration.h
1486 // #define E0_CS_PIN -1
1487 // #define E1_CS_PIN -1
1488 // #define E2_CS_PIN -1
1489 // #define E3_CS_PIN -1
1490 // #define E4_CS_PIN -1
1491 // #define E5_CS_PIN -1
1492
1493 /**
1494  * Use software SPI for TMC2130.
1495  * Software option for SPI driven drivers (TMC2130, TMC21
1496  * The default SW SPI pins are defined the respective pin
1497  * but you can override or define them here.
1498  */
1499 #define TMC_USE_SW_SPI
```

Step 5:

In `Configuration_adv.h`, find `#define X_CURRENT`, `#define X_MICROSTEPS`, `#define X_RSENSE` and modify the parameters (modifications are needed for all axes used), setting RSENSE for each axis to **0.075**.

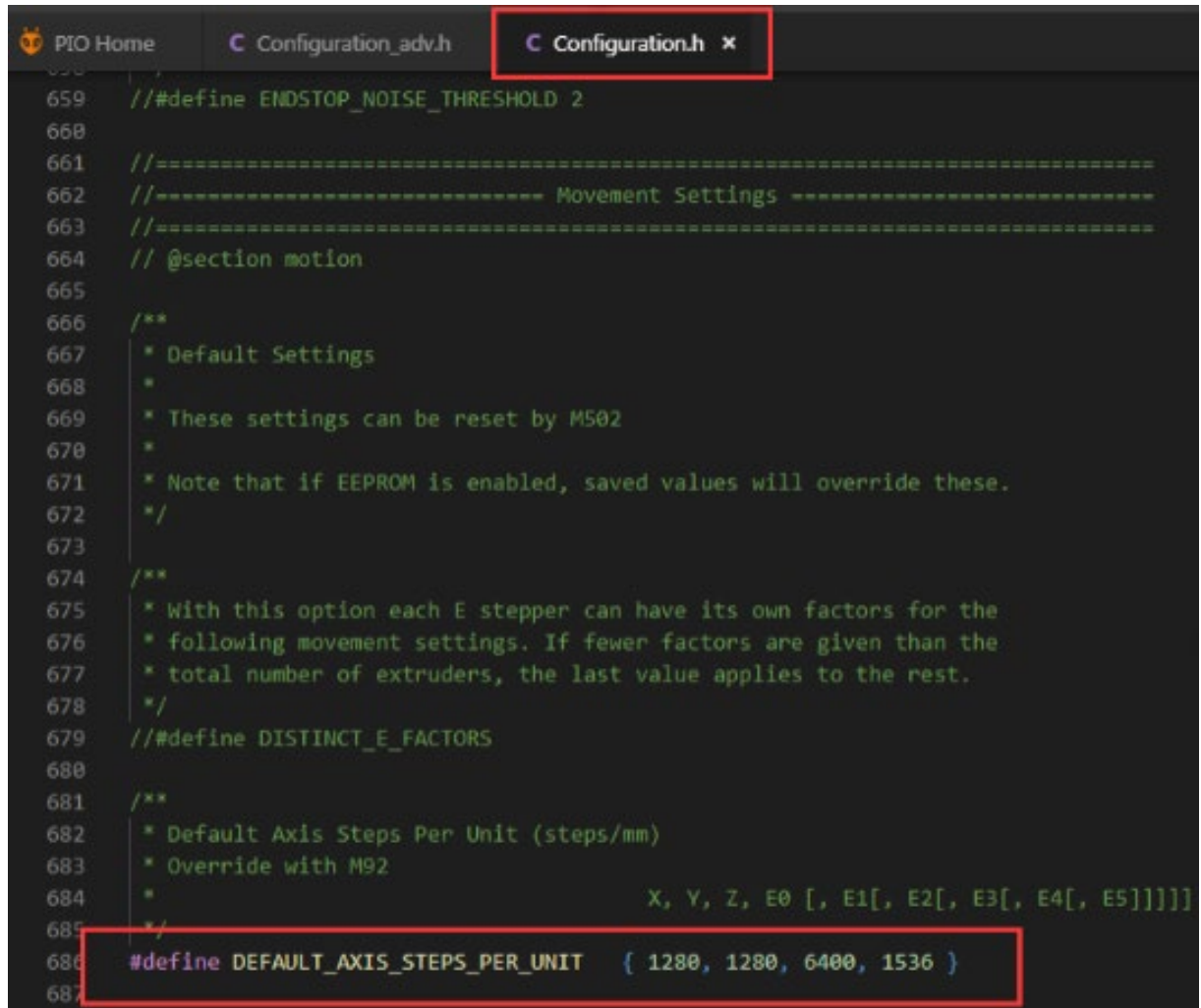


```
1390  /*
1391  */
1392  #if HAS_TRINAMIC
1393
1394      #define HOLD_MULTIPLIER    0.5  // Scales down the holding current from motor current
1395      #define INTERPOLATE        true  // Interpolates the stepper position between each stepper microstep
1396
1397      #if AXIS_IS_TMC(X)
1398          #define X_CURRENT        1000  // (mA) RMS current for each motor. If unknown, use defaults
1399          #define X_MICROSTEPS    64    // 0.256
1400          #define X_RSENSE        0.075
1401      #endif
1402
1403      #if AXIS_IS_TMC(X2)
1404          #define X2_CURRENT       800
1405          #define X2_MICROSTEPS   16
1406          #define X2_RSENSE       0.11
1407      #endif
1408
1409      #if AXIS_IS_TMC(Y)
1410          #define Y_CURRENT        1000
1411          #define Y_MICROSTEPS    64
1412          #define Y_RSENSE        0.075
1413      #endif
1414
1415      #if AXIS_IS_TMC(Y2)
1416          #define Y2_CURRENT       800
1417          #define Y2_MICROSTEPS   16
1418          #define Y2_RSENSE       0.11
1419      #endif
1420
1421      #if AXIS_IS_TMC(Z)
1422          #define Z_CURRENT        1000
1423          #define Z_MICROSTEPS    64
1424          #define Z_RSENSE        0.075
1425      #endif
1426
1427      #if AXIS_IS_TMC(Z2)
1428          #define Z2_CURRENT       800
1429          #define Z2_MICROSTEPS   16
1430          #define Z2_RSENSE       0.11
1431      #endif
1432  #endif
```

Step 6:

After completing step 5, open **Configuration.h** and locate **#define DEFAULT_AXIS_STEPS_PER_UNIT** and modify the parameters to set microstepping, ensuring it corresponds with the microstepping from step 5.

For microstepping calculation, "80,80,400,96" represents 16 microsteps, and if changed to 32 microsteps it becomes "80*(32/16), 80*(32/16), 400*(32/16), 96*(32/16)".



```
659 // #define ENDSTOP_NOISE_THRESHOLD 2
660
661 //=====
662 //----- Movement Settings -----
663 //=====
664 // @section motion
665
666 /**
667  * Default Settings
668  *
669  * These settings can be reset by M502
670  *
671  * Note that if EEPROM is enabled, saved values will override these.
672  */
673
674 /**
675  * With this option each E stepper can have its own factors for the
676  * following movement settings. If fewer factors are given than the
677  * total number of extruders, the last value applies to the rest.
678  */
679 // #define DISTINCT_E_FACTORS
680
681 /**
682  * Default Axis Steps Per Unit (steps/mm)
683  * Override with M92
684  *
685  * X, Y, Z, E0 [, E1[, E2[, E3[, E4[, E5]]]]
686  */
687 #define DEFAULT_AXIS_STEPS_PER_UNIT { 1280, 1280, 6400, 1536 }
```

The TMC5160T Pro V1.0 uses a 0.075R current sensing resistor, which sets the maximum effective RMS current to 3.1A.

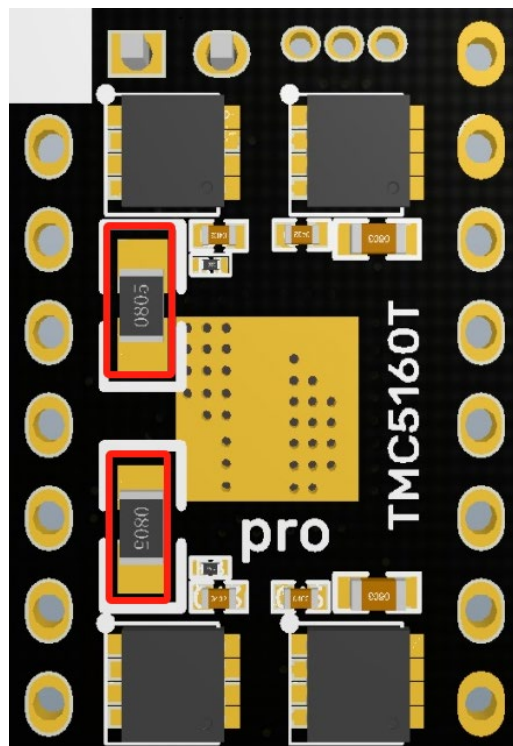
CHOICE OF R_{SENSE} AND RESULTING MAX. MOTOR CURRENT WITH $GLOBALSCALER=255$		
$R_{SENSE} [\Omega]$	RMS current [A] (CS=31)	Sine wave peak current [A] (CS=31)
0.22	1.1	1.5
0.15	1.6	2.2
0.12	2.0	2.8
0.10	2.3	3.3
0.075	3.1	4.4
0.066	3.5	5.0
0.050	4.7	6.6
0.033	7.1	10.0
0.022	10.6	15.0

If you require higher currents, it is possible to replace the current sensing resistor with a new one. Please note that you will need to source and solder it yourself.

Ensure that the replacement resistor is no less than 0.066R due to the size constraints of the module.

Note: Replacing the resistor is not recommended, but if you decide to go ahead, you'll need to take responsibility for any damage that might happen during the swap.

The location for the replacement resistor is indicated by the red box in the diagram below.



4. Cautions

Disconnect power before installing the driver to avoid damage.

Ensure proper orientation during installation to prevent malfunction.

Avoid hot-plugging the driver module to prevent damage.

If you need further resources for this product, you can find them at [GitHub](<https://github.com/bigtreetech/>). If you cannot find what you need, you may contact our after-sales support(service005@biqu3d.com).

If you encounter any other problems during use or have suggestions or feedback, please contact us. Thank you for choosing BIGTREETECH products.