

BIGTREETECH TMC5160 PRO-V1.0

User Manual

1. Instruction

TMC5160 is a control chip of high-power stepper motor with MOS power expansion, 20A maximum current and low heat generation.

StealthChop2 mode for TRINAMICs eliminates motor noise by reducing resonance. StallGuard2™ filament blockage detection enables stepper motor torque control or back to zero without a sensor, which is a safe detection of motor stopping and the replacement of mechanical stop switch. DcStep™ allows the motor to run near its load limit and speed limit, achieving 10x or higher range without any pulse loss. SpreadCycle™ is high precision chopping algorithm for highly dynamic motor motion and generating absolutely clean current waves. Low noise, low resonance and low vibration chopper. CoolStep™ current control optimizes driver performance and energy efficiency, enables smooth and silent drive, balances speed and motor torque, reduces energy consumption by 75 %.

TMC5160 is an upgrade of TMC2100, TMC2130 and TMC5130 series, with higher voltage and motor currents.

2. Product Parameters

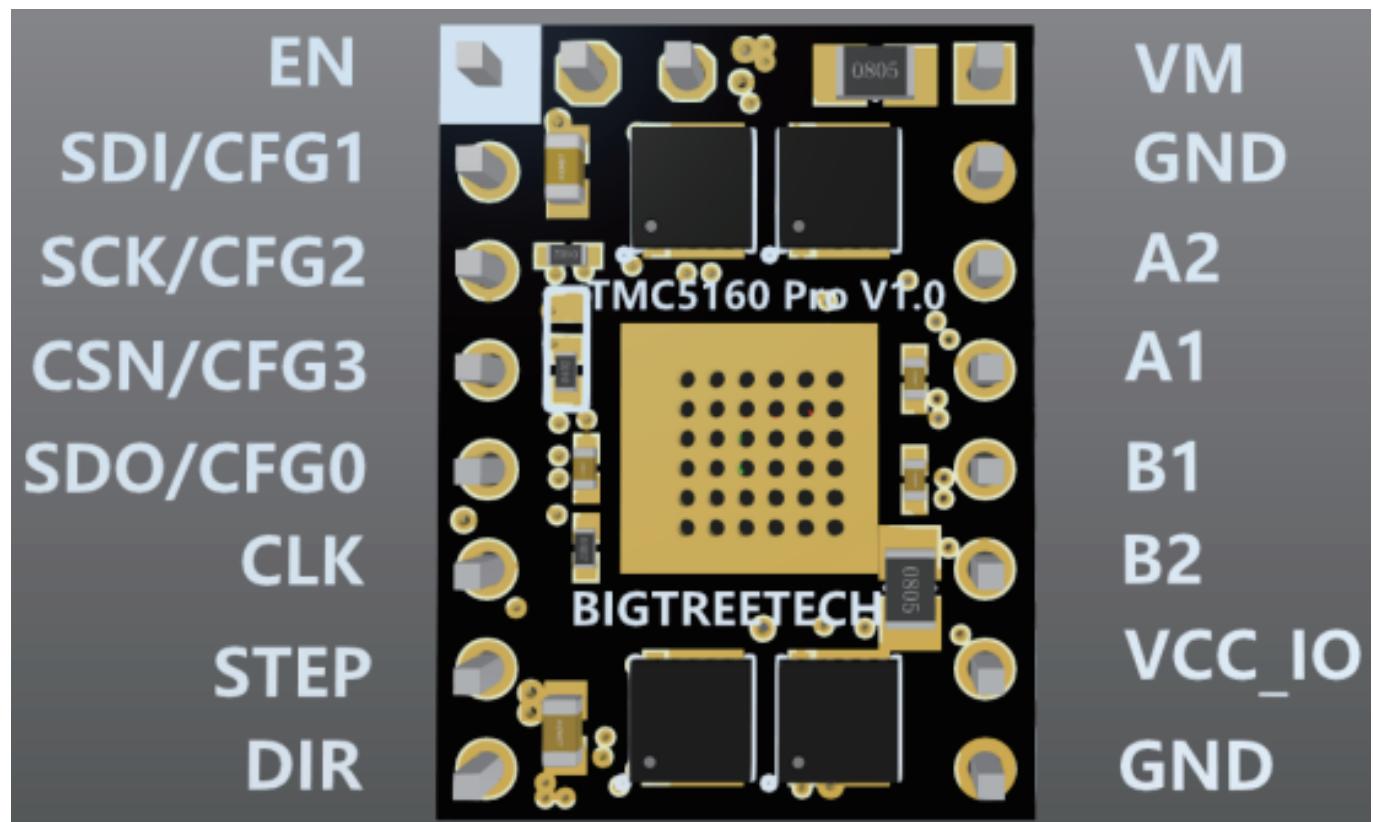
| | |
|----------------------|---|
| Driver Chip | TMC5160-WA |
| Product Size | 15.3mm*20.4mm |
| Supply Voltage | 8V--60V |
| Maximum Current | 3A (maximum current of 2.54 single-row pins-3A) |
| Maximum Segmentation | 256 |
| Working Mode | SPI Mode, SD Mode |

3. Advantages

- External power MOS tube, for higher current
- Ultra-silent mode
- Low heat generation
- Less motor jittering
- Less pulse loss
- It is able to drive 57 stepper motor

4. Pins Instruction

4.1 Names of pins

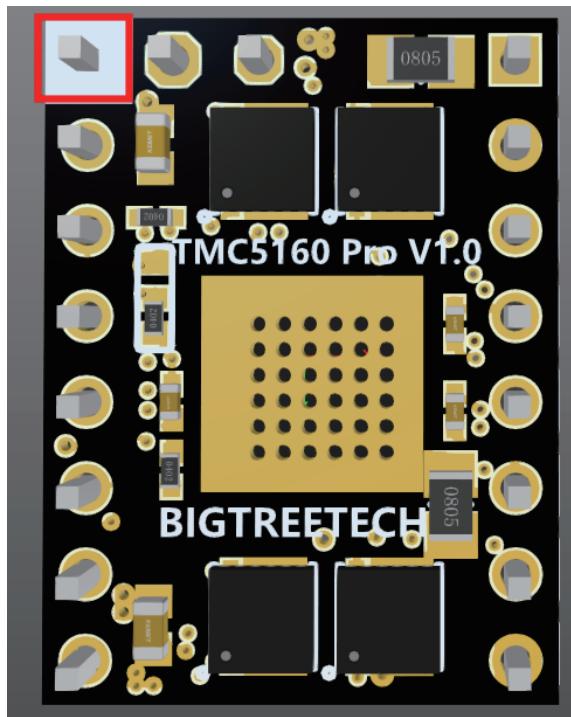


4.2 Functions of pins

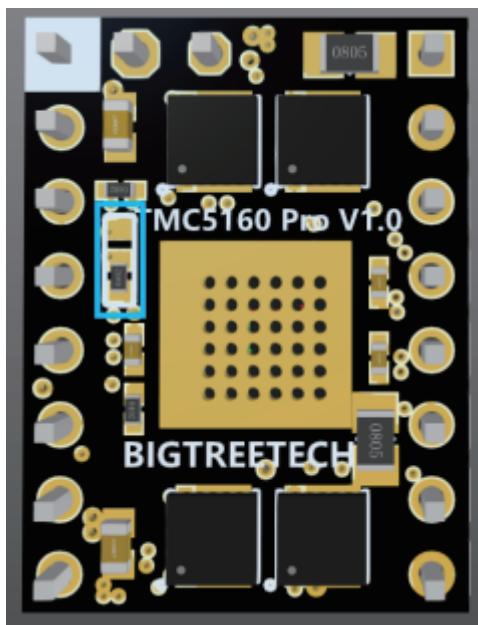
| J1 | Functions | J2 | Functions |
|----|-----------|----|-----------|
| 1 | EN | 1 | VM |
| 2 | SDI/CFG1 | 2 | GND |
| 3 | SCK/CFG2 | 3 | A2 |
| 4 | CSN/CFG3 | 4 | A1 |
| 5 | SDO/CFG0 | 5 | B1 |
| 6 | CLK | 6 | B2 |
| 7 | STEP | 7 | VCC_IO |
| 8 | DIR | 8 | GND |

5. Driver installation

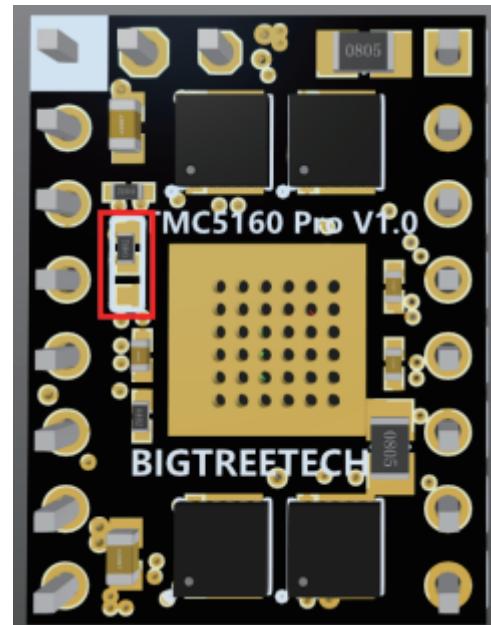
The pin with white box on the driver is enable pin(EN)



6. SD_MODE



The factory default mode $SD_MODE = 1$,
the STEP / DIR input pins control the
driver as shown



To use $SD_MODE = 0$, the step signal is made
by internal ramp generator and the resis-
tor is welded to the other side as shown

7. Firmware Configuration

7.1 Marlin

(a) Set the driver as TMC5160 in Configuration.h

```
836 /**
837 * Stepper Drivers
838 *
839 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
840 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
841 *
842 * A4988 is assumed for unspecified drivers.
843 *
844 * Use TMC2288/TMC2288_STANDALONE for TMC2225 drivers and TMC2289/TMC2289_STANDALONE for TMC2226 drivers.
845 *
846 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
847 *           TB6560, TB6600, TMC2100,
848 *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
849 *           TMC2288, TMC2288_STANDALONE, TMC2289, TMC2289_STANDALONE,
850 *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
851 *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
852 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100'],
853 */
854 #define X_DRIVER_TYPE TMC5160
855 #define Y_DRIVER_TYPE TMC5160
856 #define Z_DRIVER_TYPE TMC5160
857 // #define X2_DRIVER_TYPE A4988
858 // #define Y2_DRIVER_TYPE A4988
859 // #define Z2_DRIVER_TYPE A4988
860 // #define Z3_DRIVER_TYPE A4988
861 // #define Z4_DRIVER_TYPE A4988
862 // #define I_DRIVER_TYPE A4988
863 // #define J_DRIVER_TYPE A4988
864 // #define K_DRIVER_TYPE A4988
865 #define E0_DRIVER_TYPE TMC5160
866 // #define E1_DRIVER_TYPE A4988
867 // #define E2_DRIVER_TYPE A4988
868 // #define E3_DRIVER_TYPE A4988
869 // #define E4_DRIVER_TYPE A4988
870 // #define E5_DRIVER_TYPE A4988
871 // #define E6_DRIVER_TYPE A4988
872 // #define E7_DRIVER_TYPE A4988
```

(b) If there is an independent SPI port, set TMC_USE_SW_SPI in Configuration-adv.h

```
2774 /**
2775 * Software option for SPI driven drivers (TMC2130, TMC
2776 * The default SW SPI pins are defined the respective p
2777 * but you can override or define them here.
2778 */
2779 #define TMC_USE_SW_SPI
2780 // #define TMC_SW_MOSI -1
2781 // #define TMC_SW_MISO -1
2782 // #define TMC_SW_SCK -1
2783
```

(c) If the motherboard needs custom pins, customize the CS signal lines in the “pins_xxx.h” and the SPI signal lines in “Configuration_adv.h”

```
Marlin > src > pins > stm32f4 > C pins_BTT_OCTOPUS_V1_common.h ...  
153 //  
154 #define X_STEP_PIN PF13 // MOTOR 0  
155 #define X_DIR_PIN PF12  
156 #define X_ENABLE_PIN PF14  
157 #ifndef X_CS_PIN  
158 #define X_CS_PIN PC4  
159#endif  
160  
161 #define Y_STEP_PIN PG0 // MOTOR 1  
162 #define Y_DIR_PIN PG1  
163 #define Y_ENABLE_PIN PF15  
164 #ifndef Y_CS_PIN  
165 #define Y_CS_PIN PD11  
166#endif  
167  
168 #define Z_STEP_PIN PF11 // MOTOR 2  
169 #define Z_DIR_PIN PG3  
170 #define Z_ENABLE_PIN PG5  
171 #ifndef Z_CS_PIN  
172 #define Z_CS_PIN PC6  
173#endif  
174  
175 #define Z2_STEP_PIN PG4 // MOTOR 3  
176 #define Z2_DIR_PIN PC1  
177 #define Z2_ENABLE_PIN PA0  
178 #ifndef Z2_CS_PIN  
179 #define Z2_CS_PIN PC7  
180#endif  
181  
182 #define E0_STEP_PIN PF9 // MOTOR 4  
183 #define E0_DIR_PIN PF10  
184 #define E0_ENABLE_PIN PG2  
185 #ifndef E0_CS_PIN  
186 #define E0_CS_PIN PF2  
187#endif  
188  
189 #define E1_STEP_PIN PC13 // MOTOR 5  
190 #define E1_DIR_PIN PF0  
191 #define E1_ENABLE_PIN PF1  
192 #ifndef E1_CS_PIN  
193 #define E1_CS_PIN PE4  
194#endif  
195
```

(d) Set the sampling resistance to 0.075 (the sampling resistance value of the driver is 0.075), then set the current and subdivision according to your own needs

```

C Configuration.h 1,M C Configuration_adv.h .M X C pins_BIT_OCTOPUS_V1_common.h 8
Marlin > C Configuration_adv.h > Y2_MICROSTEPS
2577 #if HAS_TRINAMIC_CONFIG
2578
2581 /**
2582 * Interpolate microsteps to 256
2583 * Override for each driver with <driver>_INTERPOLATE settings below
2584 */
2585 #define INTERPOLATE      true
2586
2587 #if AXIS_IS_TMC(X)
2588     #define X_CURRENT      800          // (mA) RMS current. Multiply by 1.414 for peak current.
2589     #define X_CURRENT_HOME  X_CURRENT   // (mA) RMS current for sensorless homing
2590     #define X_MICROSTEPS    16           // 0..256
2591     #define X_RSENSE        0.075
2592     #define X_CHAIN_POS     -1           // -1..0: Not chained. 1: MCU MOSI connected. 2: Next in chain, ...
2593     // #define X_INTERPOLATE  true       // Enable to override 'INTERPOLATE' for the X axis
2594 #endif
2595
2596 #if AXIS_IS_TMC(X2)
2597     #define X2_CURRENT     800
2598     #define X2_CURRENT_HOME X2_CURRENT
2599     #define X2_MICROSTEPS   X_MICROSTEPS
2600     #define X2_RSENSE       0.075
2601     #define X2_CHAIN_POS    -1
2602     // #define X2_INTERPOLATE true
2603 #endif
2604
2605 #if AXIS_IS_TMC(Y)
2606     #define Y_CURRENT      800
2607     #define Y_CURRENT_HOME  Y_CURRENT
2608     #define Y_MICROSTEPS    16
2609     #define Y_RSENSE        0.075
2610     #define Y_CHAIN_POS     -1
2611     // #define Y_INTERPOLATE true
2612 #endif

```

7.2 Klipper

(a) Set the current and subdivision. For more details, please refer to

https://www.klipper3d.org/Config_Reference.html#tmc5160

```

printer.cfg X
C: > Users > Administrator > Desktop > Canbus-Toolboard > printer.cfg
359
360 [tmc5160 stepper_x]
361 cs_pin: PC4
362 sense_resistor: 0.075
363 interpolate: True
364 run_current: 1.5
365 hold_current: 0.5
366 stealthchop_threshold: 0
367 spi_bus: spi1
368 #diag1_pin: !PG6 # Pin connected to TMC DIAG1 pin (or use diag0_pin / DIAG0 pin)
369 #driver_SGT: 2 # -64 is most sensitive value, 63 is least sensitive
370 #driver_TPFD: 0
371 #driver_TOFF: 4
372 #driver_HEND: 2
373 #driver_HSTRT: 1
374 #driver_DISS2G: 12
375 #driver_DISS2VS: 12
376
377
378 [tmc5160 stepper_y]
379 cs_pin: PD11
380 sense_resistor: 0.075
381 interpolate: True
382 run_current: 1.5
383 hold_current: 0.5
384 stealthchop_threshold: 0
385 spi_bus: spi1
386 #diag1_pin: ^PG9 # Pin connected to TMC DIAG1 pin (or use diag0_pin / DIAG0 pin)
387 #driver_SGT: 2 # -64 is most sensitive value, 63 is least sensitive

```

8. Caution

- Disconnect the power supply before driver installation
- Confirm the direction of driver to avoid reverse insertion
- Do not plug and unplug the driver module when power is on to avoid damage
- Please note that the heat sink cannot contact with the pins to prevent the driver from short circuit
- TMC5160 is sensitive to static electricity, please be careful

9. Download link

<https://github.com/bigtreeetech/BIGTREETECH-Stepper-Motor-Driver>