

BIGTREETECH

EZ5160RGB V1.0

User Manual



CONTENTS

| | |
|--|----|
| Revision History | 3 |
| Product Profile | 4 |
| Feature Highlights | 4 |
| Specifications | 4 |
| Peripheral Interface | 5 |
| Dimensions | 5 |
| Connector Diagram | 6 |
| Interface Introduction | 7 |
| Installation | 7 |
| RGB Light Status | 7 |
| Firmware Settings | 8 |
| Marlin Firmware Settings | 8 |
| Klipper Firmware Settings | 11 |
| Precautions | 11 |

Revision History

| Version | Revisions | Date |
|---------|-----------|------------|
| 01.00 | Original | 2023/04/15 |

Product Profile

The EZ5160 RGB is a high-power stepper motor drive control module with 8 external power MOSFETs. The maximum voltage can reach 56V, supporting a wider range of stepper motors with higher adaptability.

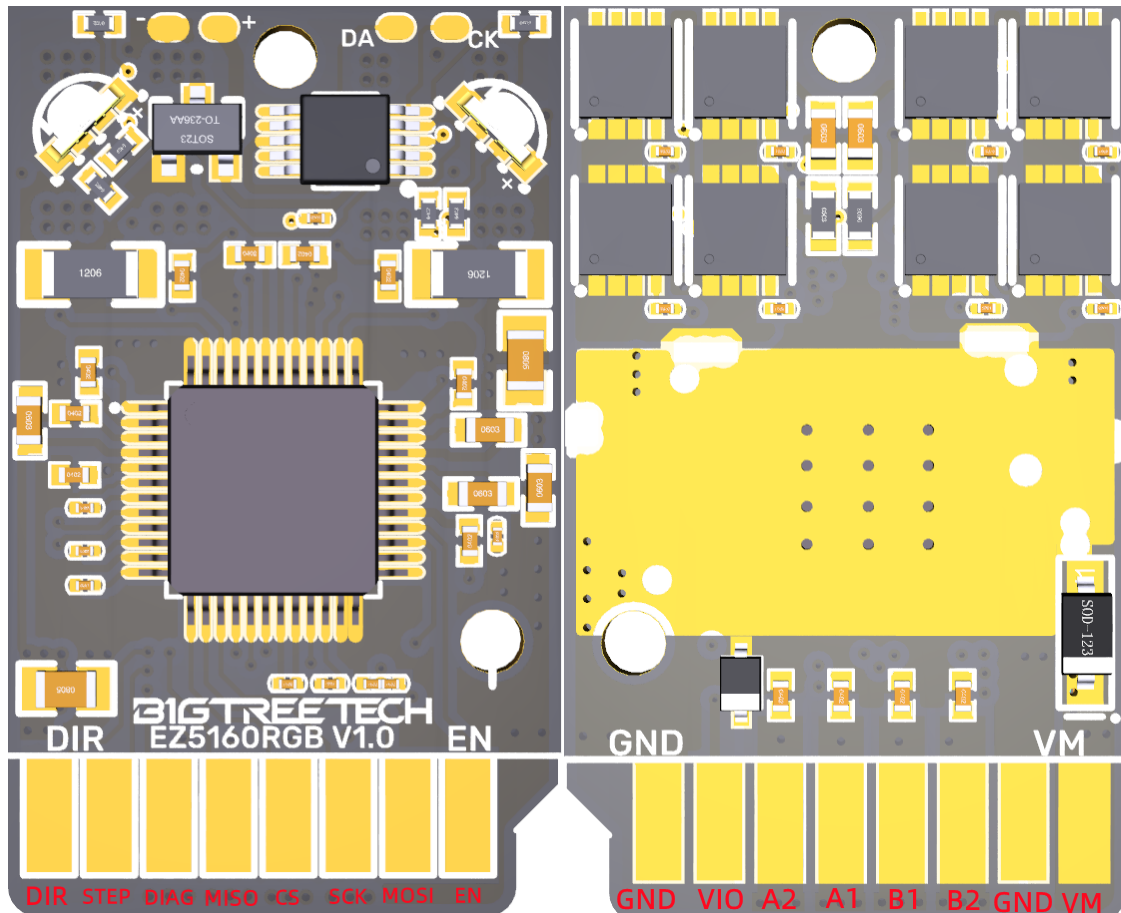
Feature Highlights

- Incorporates 8 external power MOSFETs, supports maximum voltage VM=56V, maximum current IRMS=4.7A, sine wave peak current: 6.6A;
- Ultra-quiet mode;
- Significantly less heat generation compared to 2100 and 2130 drivers;
- Helps to prevent motor jitter;
- Less prone to missed steps;
- Capable of driving 57 stepper motors;
- Utilizes a newly developed board frame and pinless connector to improve user experience and prevent injuries from old driver pins;
- Customized heat sink for enhanced heat dissipation and protection of driver chips from external damage;
- Onboard RGB lights and thermistor temperature sensing for real-time module temperature monitoring, with RGB light status indicating temperature information.

Specifications

- Dimensions: 18.5 x 30.47mm
- Driver Chip: TMC5160-TA
- Input Voltage (VM): 8V-56V
- Maximum Current: 4.7A, sine wave peak current: 6.6A
- Maximum Microsteps: 256
- Operating Mode: SPI
- R Sensor: 50mΩ

Connector Diagram



Features

- (EN) Enable
- (SDI/CFG1) Data
- (SCK/CFG2) Clock
- (CSN/CFG3) Chip Select
- (SDO/CFG0) Data
- (DIAG) Stall Detection
- (STEP) Pulse Input
- (DIR) Direction Input

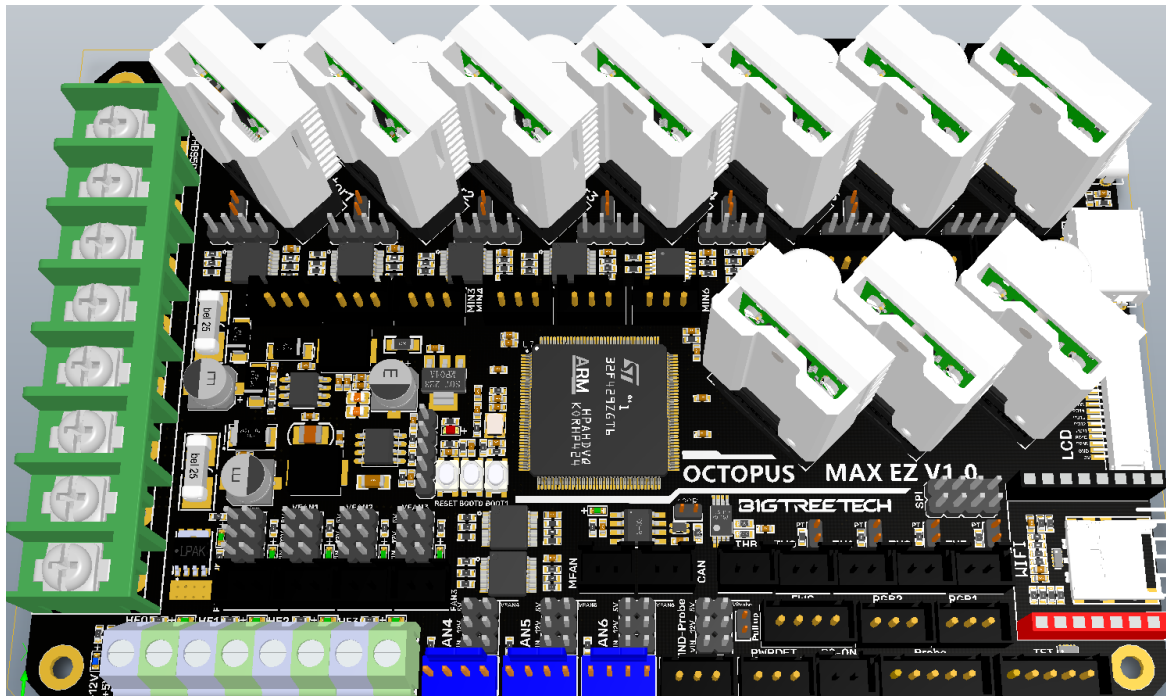
Features

- (VM) Motor Power Supply Voltage
- (GND) Ground
- (B2) B Phase
- (B1) B Phase
- (A1) A Phase
- (A2) A Phase
- (VIO) Logic Voltage
- (GND) Ground

Interface Introduction

Installation

To install the EZ driver into the EZ driver slot, follow the direction as shown in the figure below. (e.g.: Octopus MAX EZ + EZ5160RGB)



RGB Light Status

Blue: When the temperature is below 40°C.

As the temperature rises from 40°C to 75°C, the RGB blue brightness decreases while the red brightness increases. The hue transitions from blue to purple, and subsequently shifts to magenta.

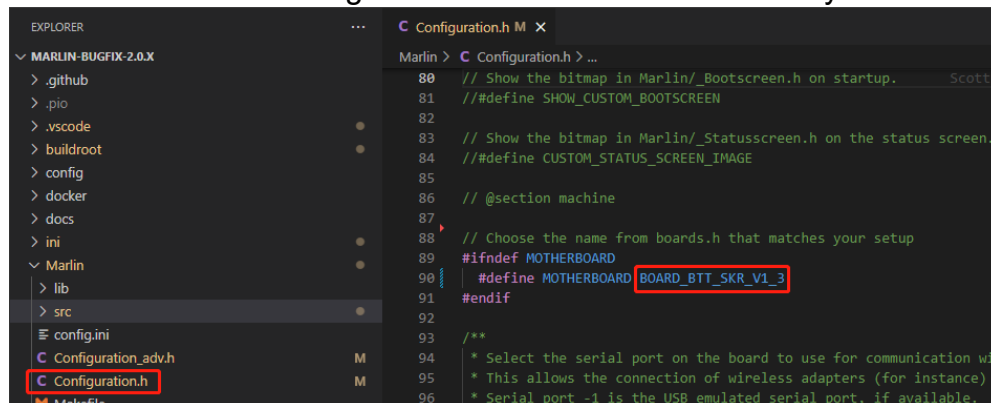
Red: When the temperature is above 75°C.

Firmware Settings

Marlin Firmware Settings

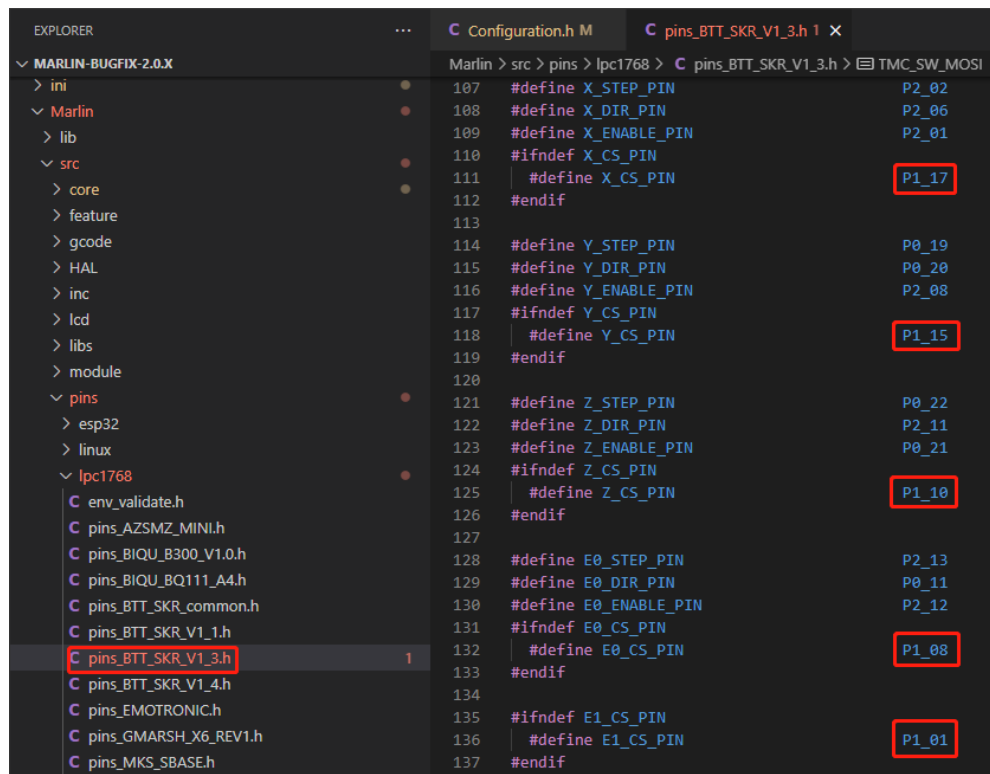
Important Note: Currently, only Marlin 2.0 and later firmware versions support TMC5160's SPI mode.

Step 1: In the Marlin 2.0 firmware, locate and open the "Configuration.h" file, then find the line "#define MOTHERBOARD XXXXXX". "XXXXXX" represents the model of the board being used. Confirm the motherboard you are using.



```
80 // Show the bitmap in Marlin/_Bootscreen.h on startup. Scott
81 // #define SHOW_CUSTOM_BOOTSCREEN
82
83 // Show the bitmap in Marlin/_Statusscreen.h on the status screen.
84 // #define CUSTOM_STATUS_SCREEN_IMAGE
85
86 // @section machine
87
88 // Choose the name from boards.h that matches your setup
89 #ifndef MOTHERBOARD
90   #define MOTHERBOARD BOARD_BTT_SKR_V1_3
91 #endif
92
93 /**
94  * Select the serial port on the board to use for communication with the host.
95  * This allows the connection of wireless adapters (for instance) to
96  * Serial port -1 is the USB emulated serial port, if available.
```

Step 2: In the Marlin\src\pins directory, find the "pins_XXXXXX.h" file corresponding to your board (XXXXXX represents the board model), and then locate "X_CS_PIN", "Y_CS_PIN", "Z_CS_PIN", and "EO_CS_PIN" within the file. Modify the pin names following these variables to the pins you are using.

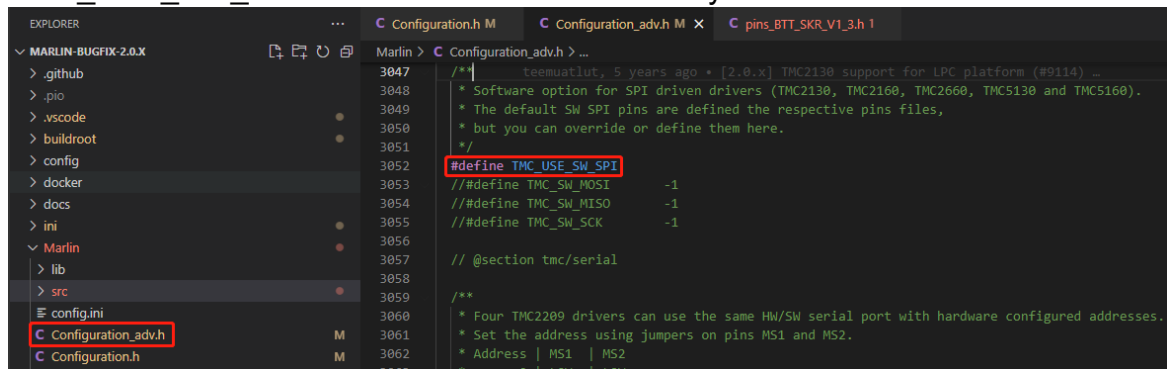


```
107 #define X_STEP_PIN P2_02
108 #define X_DIR_PIN P2_06
109 #define X_ENABLE_PIN P2_01
110 #ifndef X_CS_PIN
111   #define X_CS_PIN P1_17
112 #endif
113
114 #define Y_STEP_PIN P0_19
115 #define Y_DIR_PIN P0_20
116 #define Y_ENABLE_PIN P2_08
117 #ifndef Y_CS_PIN
118   #define Y_CS_PIN P1_15
119 #endif
120
121 #define Z_STEP_PIN P0_22
122 #define Z_DIR_PIN P2_11
123 #define Z_ENABLE_PIN P0_21
124 #ifndef Z_CS_PIN
125   #define Z_CS_PIN P1_10
126 #endif
127
128 #define E0_STEP_PIN P2_13
129 #define E0_DIR_PIN P0_11
130 #define E0_ENABLE_PIN P2_12
131 #ifndef E0_CS_PIN
132   #define E0_CS_PIN P1_08
133 #endif
134
135 #ifndef E1_CS_PIN
136   #define E1_CS_PIN P1_01
137 #endif
```


Step 3: In the file from Step 2, locate "#define TMC_SW_MOSI XXX", "#define TMC_SW_MISO XXX", and "#define TMC_SW_SCK XXX". Change "XXX" to the pins you want to use.

```
139 //
140 // Software SPI pins for TMC2130 stepper drivers
141 //
142 #if ENABLED(TMC_USE_SW_SPI)
143   #ifndef TMC_SW_MOSI
144     #define TMC_SW_MOSI P4_28
145   #endif
146   #ifndef TMC_SW_MISO
147     #define TMC_SW_MISO P0_05
148   #endif
149   #ifndef TMC_SW_SCK
150     #define TMC_SW_SCK P0_04
151   #endif
152 #endif
```

Step 4: Find and open "Configuration_adv.h", then locate "#define TMC_USE_SW_SPI" and remove the comment symbols "/*".



```
3047 /**
3048  * teemuatlati, 5 years ago * [2.0.x] TMC2130 support for LPC platform (#9114) ...
3049  * Software option for SPI driven drivers (TMC2130, TMC2160, TMC2660, TMC5130 and TMC5160).
3050  * The default SW SPI pins are defined the respective pins files,
3051  * but you can override or define them here.
3052  */
3053 #define TMC_USE_SW_SPI
3054 // #define TMC_SW_MOSI -1
3055 // #define TMC_SW_MISO -1
3056 // #define TMC_SW_SCK -1
3057
3058 // @section tmc/serial
3059
3060 /**
3061  * Four TMC2209 drivers can use the same HW/SW serial port with hardware configured addresses.
3062  * Set the address using jumpers on pins MS1 and MS2.
3063  * Address | MS1 | MS2
3064  * -----|----|----
3065  * 0 | LOW | LOW
3066  * 1 | LOW | HIGH
3067  * 2 | HIGH | LOW
3068  * 3 | HIGH | HIGH
```

Step 5: In the "Configuration_adv.h" file, find "#define X_CURRENT", "#define X_MICROSTEPS", and "#define X_RSENSE" and modify the parameters that follow (for each axis being used). The RSENSE value for each used axis should be changed to "0.050".

```

2796 #if HAS_TRINAMIC_CONFIG
2797
2798 #define HOLD_MULTIPLIER    0.5 // Scales down the holding current from run current
2799
2800 /**
2801  * Interpolate microsteps to 256
2802  * Override for each driver with <driver>_INTERPOLATE settings below
2803  */
2804 #define INTERPOLATE        true
2805
2806 #if AXIS_IS_TMC(X)
2807 #define X_CURRENT          800 // (mA) RMS current. Multiply by 1.414 for peak current.
2808 #define X_CURRENT_HOME    X_CURRENT // (mA) RMS current for sensorless homing
2809 #define X_MICROSTEPS       16 // 0..256
2810 #define X_RSENSE           0.050
2811 #define X_CHAIN_POS        -1 // -1..0: Not chained. 1: MCU MOSI connected. 2: Next in chain, ...
2812 // #define X_INTERPOLATE true // Enable to override 'INTERPOLATE' for the X axis
2813 // #define X_HOLD_MULTIPLIER 0.5 // Enable to override 'HOLD_MULTIPLIER' for the X axis
2814 #endif
2815
2816 #if AXIS_IS_TMC(X2)
2817 #define X2_CURRENT         800
2818 #define X2_CURRENT_HOME    X2_CURRENT
2819 #define X2_MICROSTEPS      X2_CURRENT
2820 #define X2_RSENSE          0.11
2821 #define X2_CHAIN_POS        -1
2822 // #define X2_INTERPOLATE true // Enable to override 'INTERPOLATE' for the X2 axis
2823 // #define X2_HOLD_MULTIPLIER 0.5 // Enable to override 'HOLD_MULTIPLIER' for the X2 axis
2824 #endif
2825
2826 #if AXIS_IS_TMC(Y)
2827 #define Y_CURRENT          800
2828 #define Y_CURRENT_HOME    Y_CURRENT
2829 #define Y_MICROSTEPS       16
2830 #define Y_RSENSE           0.050
2831 #define Y_CHAIN_POS        -1
2832 // #define Y_INTERPOLATE true // Enable to override 'INTERPOLATE' for the Y axis
2833 // #define Y_HOLD_MULTIPLIER 0.5 // Enable to override 'HOLD_MULTIPLIER' for the Y axis
2834 #endif

```

Step 6: Set the corresponding axis drive type to "TMC5160" in the "Configuration.h" file.

```

153 /**
154  * Stepper Drivers
155  *
156  * These settings allow Marlin to tune stepper driver timing and enable advanced options for
157  * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
158  *
159  * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STANDALONE for TMC2226 drivers.
160  *
161  * Options: A4988, A5984, DRV8825, LV8729, TB6560, TB6600, TMC2100,
162  *           TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
163  *           TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
164  *           TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
165  *           TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
166  * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_STANDALONE',
167  */
168 #define X_DRIVER_TYPE  TMC5160
169 #define Y_DRIVER_TYPE  TMC5160
170 #define Z_DRIVER_TYPE  TMC5160
171 // #define X2_DRIVER_TYPE A4988
172 // #define Y2_DRIVER_TYPE A4988
173 // #define Z2_DRIVER_TYPE A4988
174 // #define Z3_DRIVER_TYPE A4988
175 // #define Z4_DRIVER_TYPE A4988
176 // #define I_DRIVER_TYPE A4988
177 // #define J_DRIVER_TYPE A4988
178 // #define K_DRIVER_TYPE A4988
179 // #define U_DRIVER_TYPE A4988
180 // #define V_DRIVER_TYPE A4988
181 // #define W_DRIVER_TYPE A4988
182 #define E0_DRIVER_TYPE TMC5160
183 #define E1_DRIVER_TYPE TMC5160
184 // #define E2_DRIVER_TYPE A4988
185 // #define E3_DRIVER_TYPE A4988
186 // #define E4_DRIVER_TYPE A4988
187 // #define E5_DRIVER_TYPE A4988
188 // #define E6_DRIVER_TYPE A4988
189 // #define E7_DRIVER_TYPE A4988

```

Klipper Firmware Settings

```
[tmc5160 stepper_x]
cs_pin: P1.17
spi_software_miso_pin: P0.5
spi_software_mosi_pin: P4.28
spi_software_sclk_pin: P0.4
sense_resistor: 0.050
#diag1_pin: P1.29
run_current: 0.800
stealthchop_threshold: 999999
```

Note: The default sense_resistor in Klipper is 0.075; it needs to be set to 0.050.

Precautions

1. Be sure to disconnect the power supply before installing the driver to prevent damage;
2. Do not plug or unplug the driver module while it is powered, as this may cause damage;
3. The heat sink has been installed at the factory. Do not remove the heat sink, otherwise the heat dissipation effect will decrease without the heat-conducting material;
4. When using high voltage (greater than 48V) or high current (greater than 2A), active cooling must be adopted to ensure that the drive can work normally.

If you require additional resources for this product, please visit <https://github.com/bigtreotech/> to find them. If you cannot locate the resources you need, please contact our customer support.

If you encounter other problems during use, please feel free to contact us, and we will be more than happy to assist you. If you have any suggestions or feedback for our products, we are open to hearing them. Thank you for choosing BIGTREETECH products!