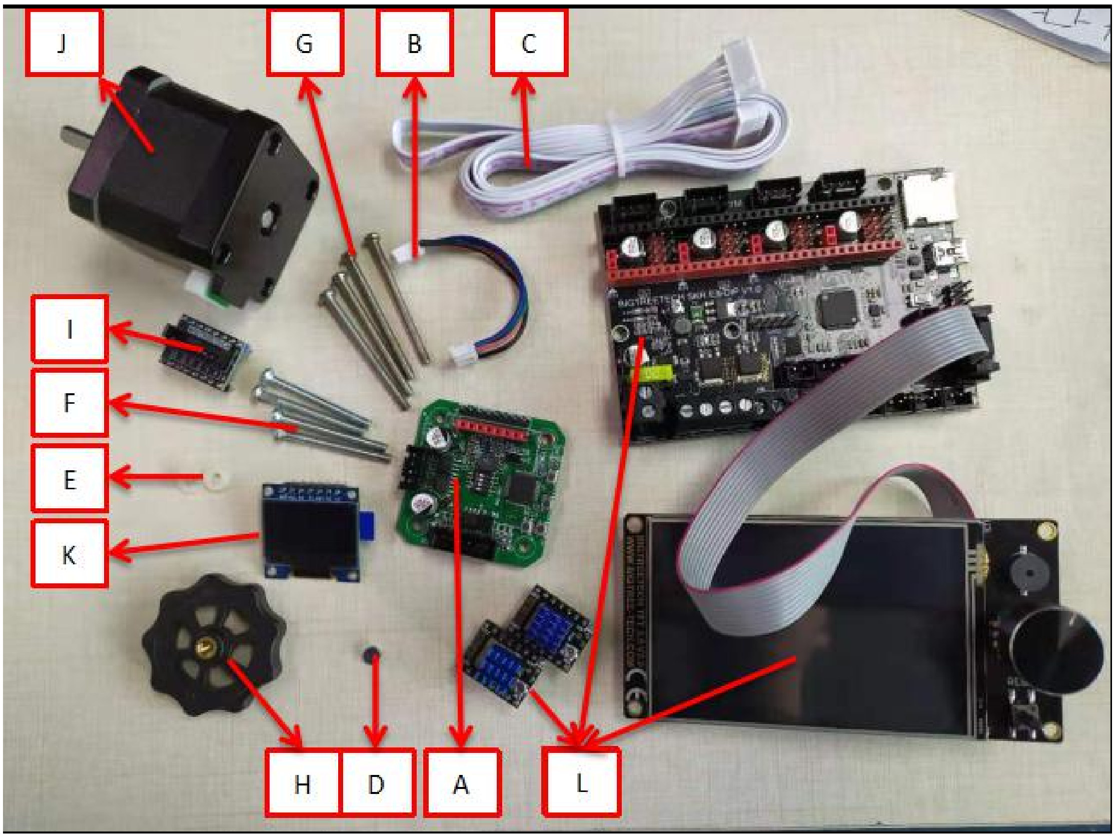
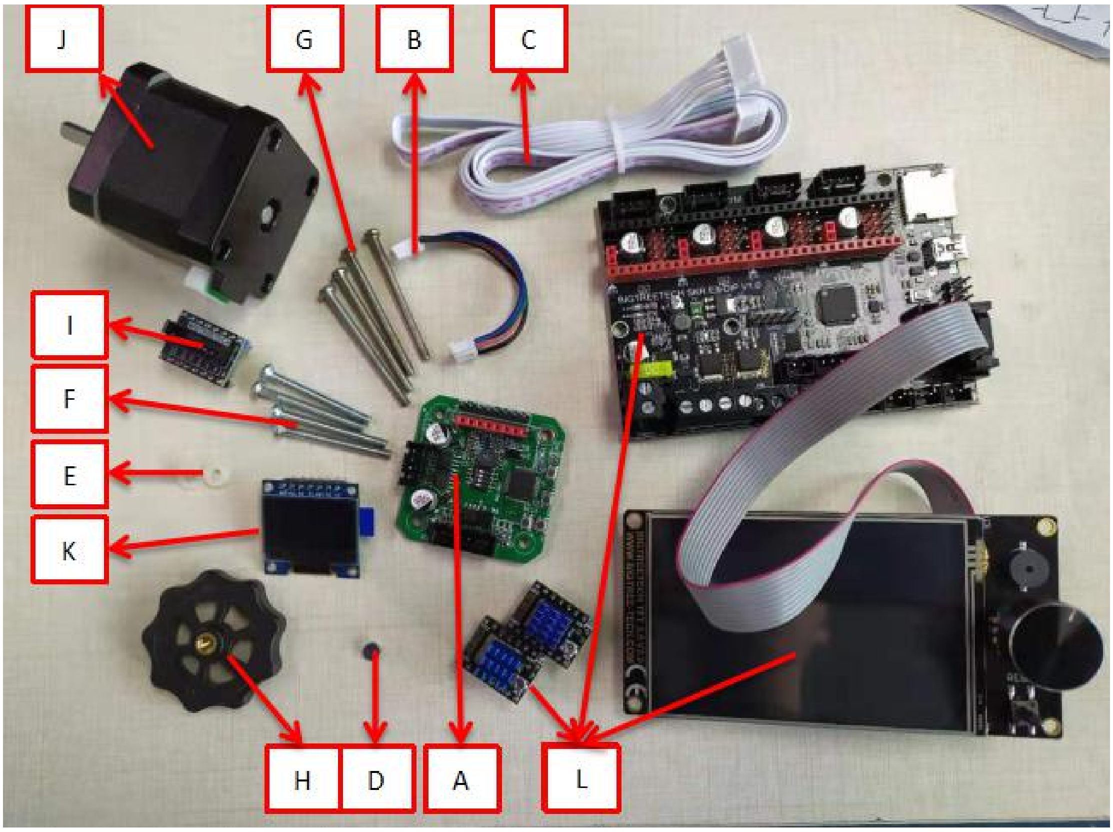
3D printer -Ender3 closed loop driver module installation instructions

Ⅰ. Closed-loop drive module installation





1. Required item :(single shaft)

   A. Closed loop driver control board S42B

   B. Connection wire between motor and closed - loop driver board, length 5cm

   C. Connection wire between closed loop driver board and main control board, length 1m

   D. Magnet: radial magnet, diameter 5mm, thickness 2mm

   E. White gasket: 1.4mm thick, M3, quantity 8

   F. Eender3 assembly screws: M3, length 34mm, quantity 4

   G. Assembly screws for other printers: M3, length: 40mm, quantity: 4

   H. Hot bed nut: aperture M4, outer diameter approx.

40mm

I. Driver circuit adapter board

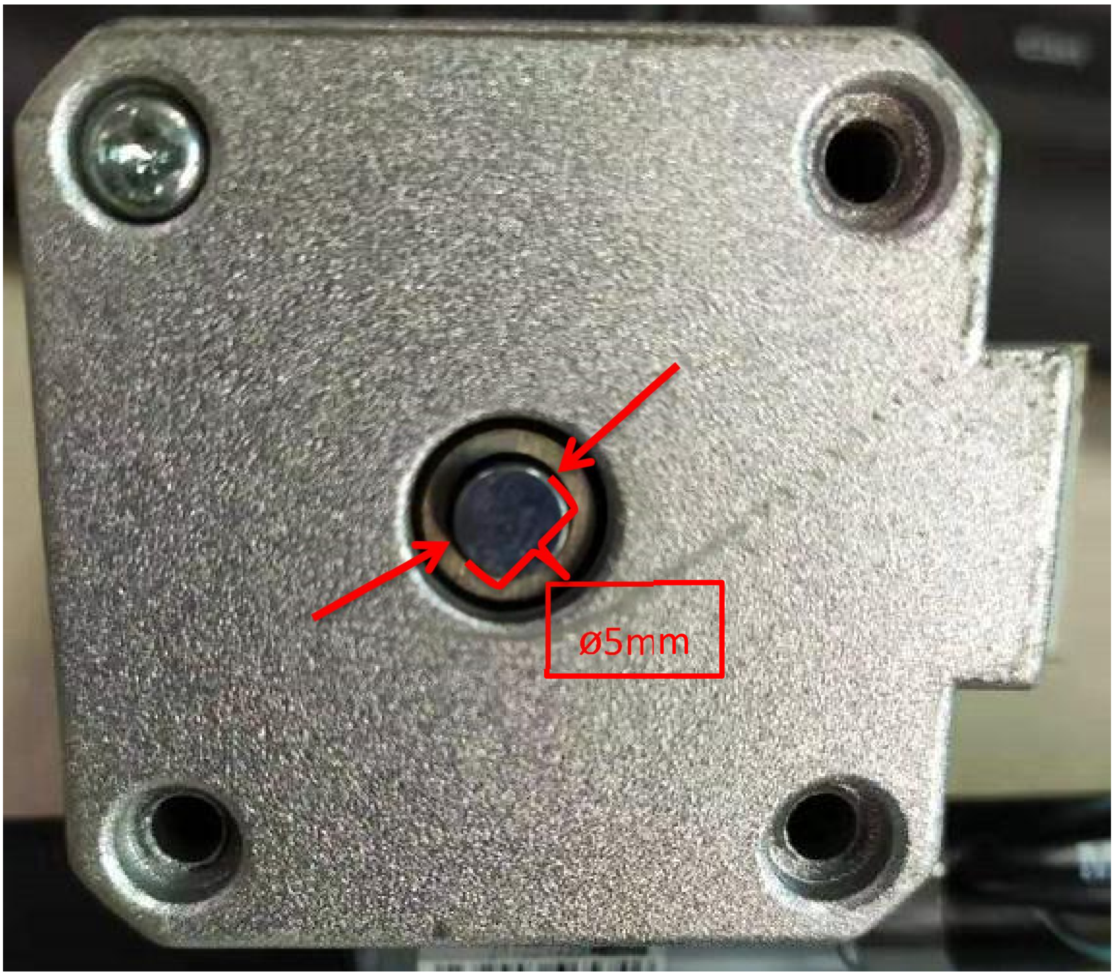
J. 42 stepper motor, step Angle 1.8°

K. OLED screen

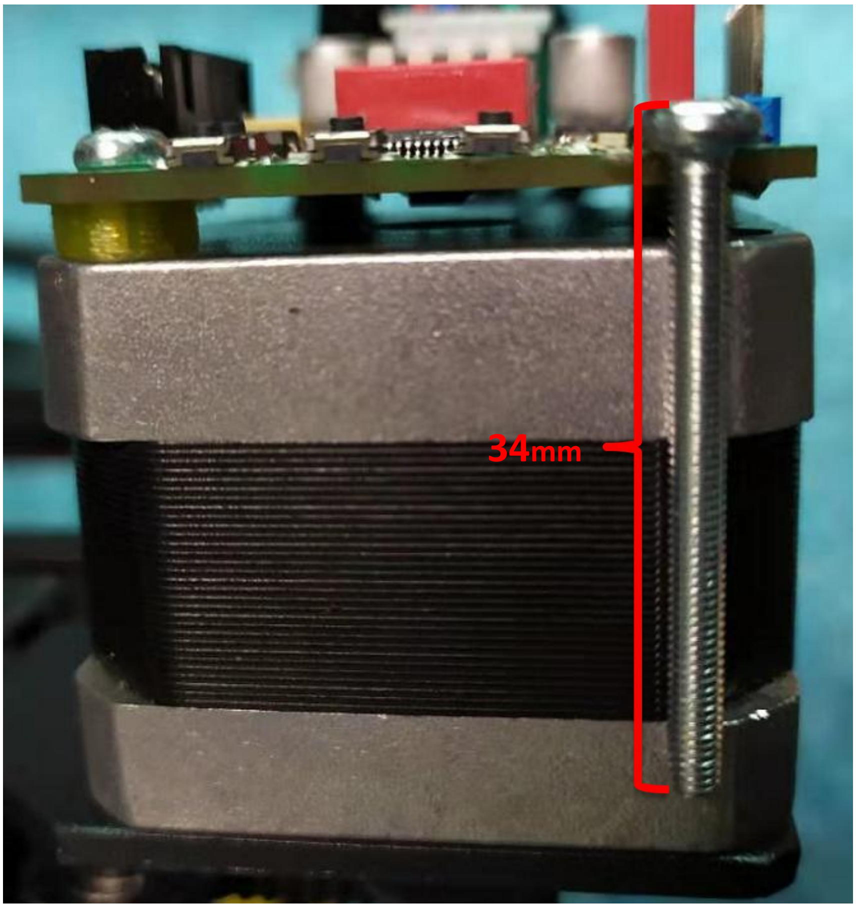
L. 3d Printer motherboard （only 1pc）： SKR E3-DIP + TMC2209（Z axis、E0）+BTT TFT35 V3.0

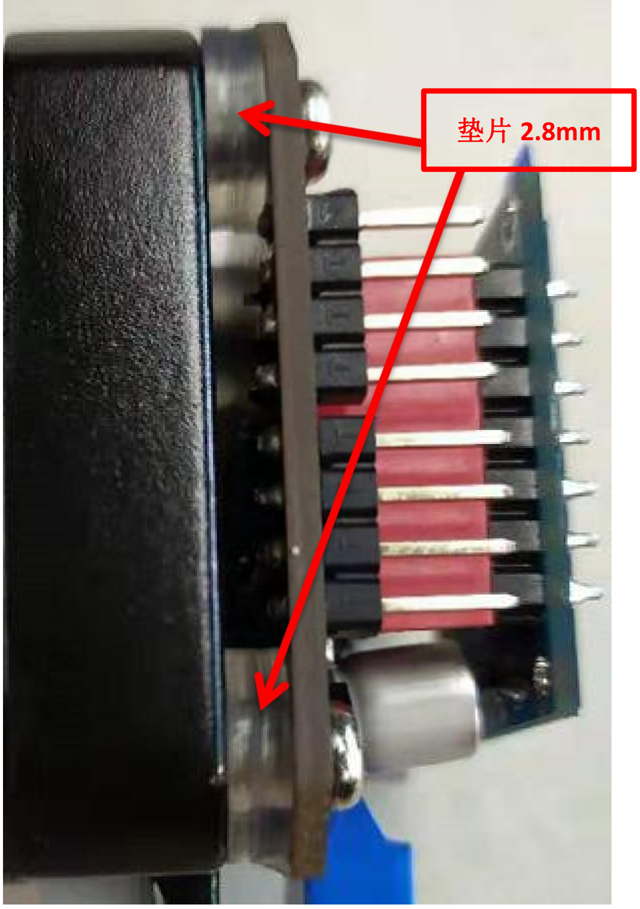
2. Magnet installation, be sure to be in the center of the motor shaft!

3. Magnet specification: radial magnet, diameter 5mm x 2mm thick. It is best to fix the magnet with glue in the center of the shaft. If you use liquid glue, be careful not to let the liquid flow into the shaft gap of the motor

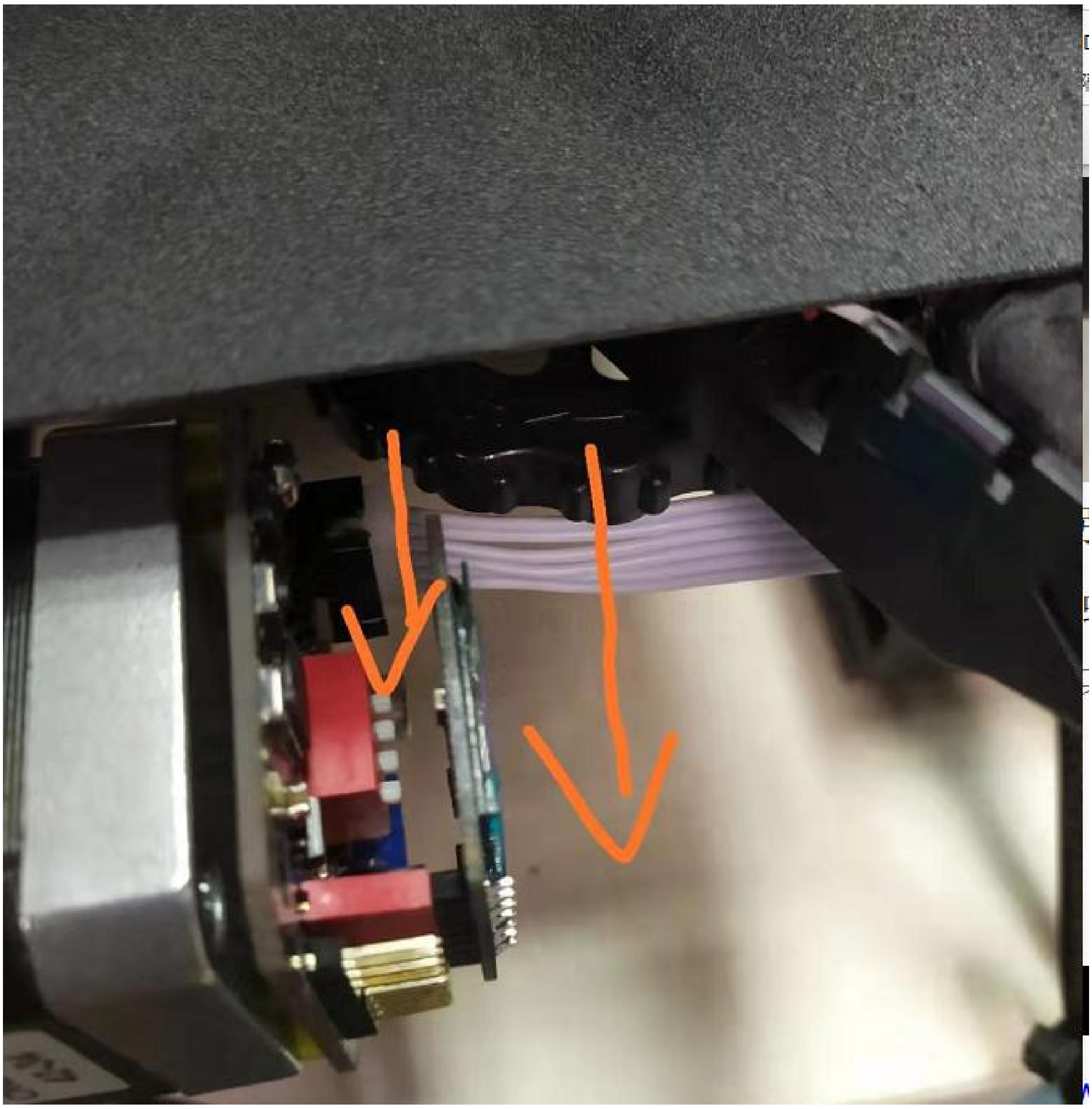


4. Gasket installation, 2 gaskets about 1.4mm thick. After adding gasket, the screw on the original motor may not be long enough. You need to replace the same type of screw, about 34mm long screw.


34mm

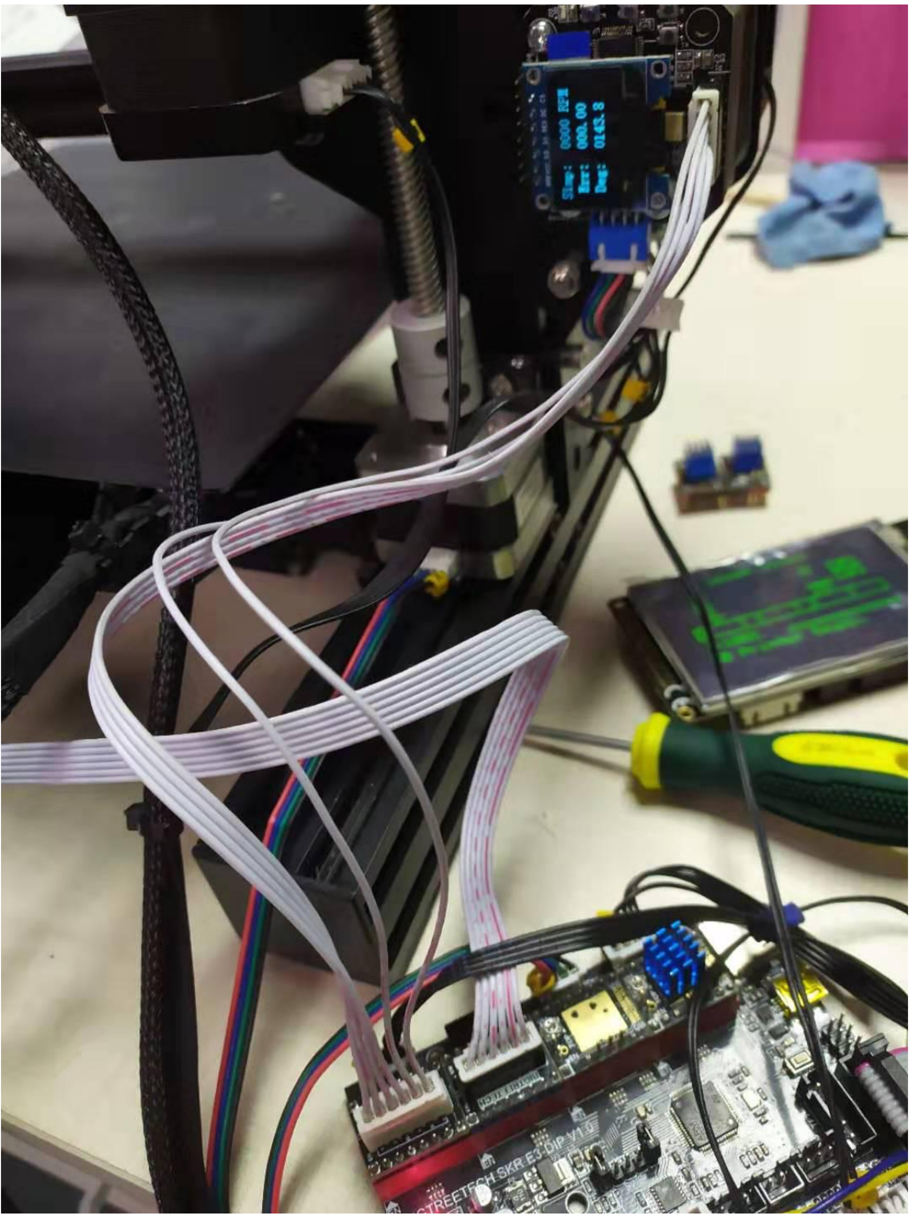

垫片 2.8mm

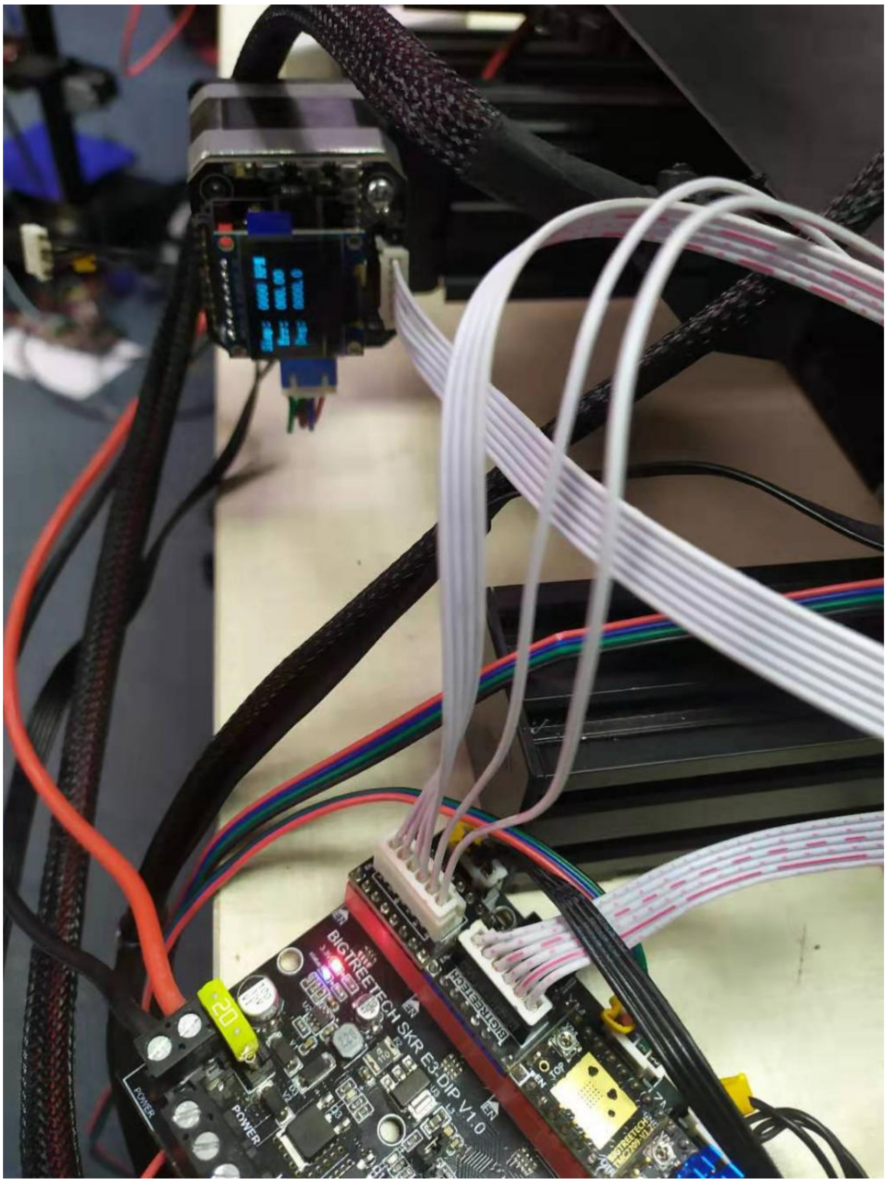
5.Special attention should be paid to the installation of the Y axis. Because the nut of the hot bed is relatively large, the nut near the corner of the motor needs to be replaced with a small nut. After the parameters are set, the screen can be removed. Avoid the phenomenon that the Y axis cannot return to the correct position during printing.



The hot bed nut shall be replaced as follows:

6. Power on after installation, as shown in the

X-axis

Y axis. After setting parameters, remove the screen.

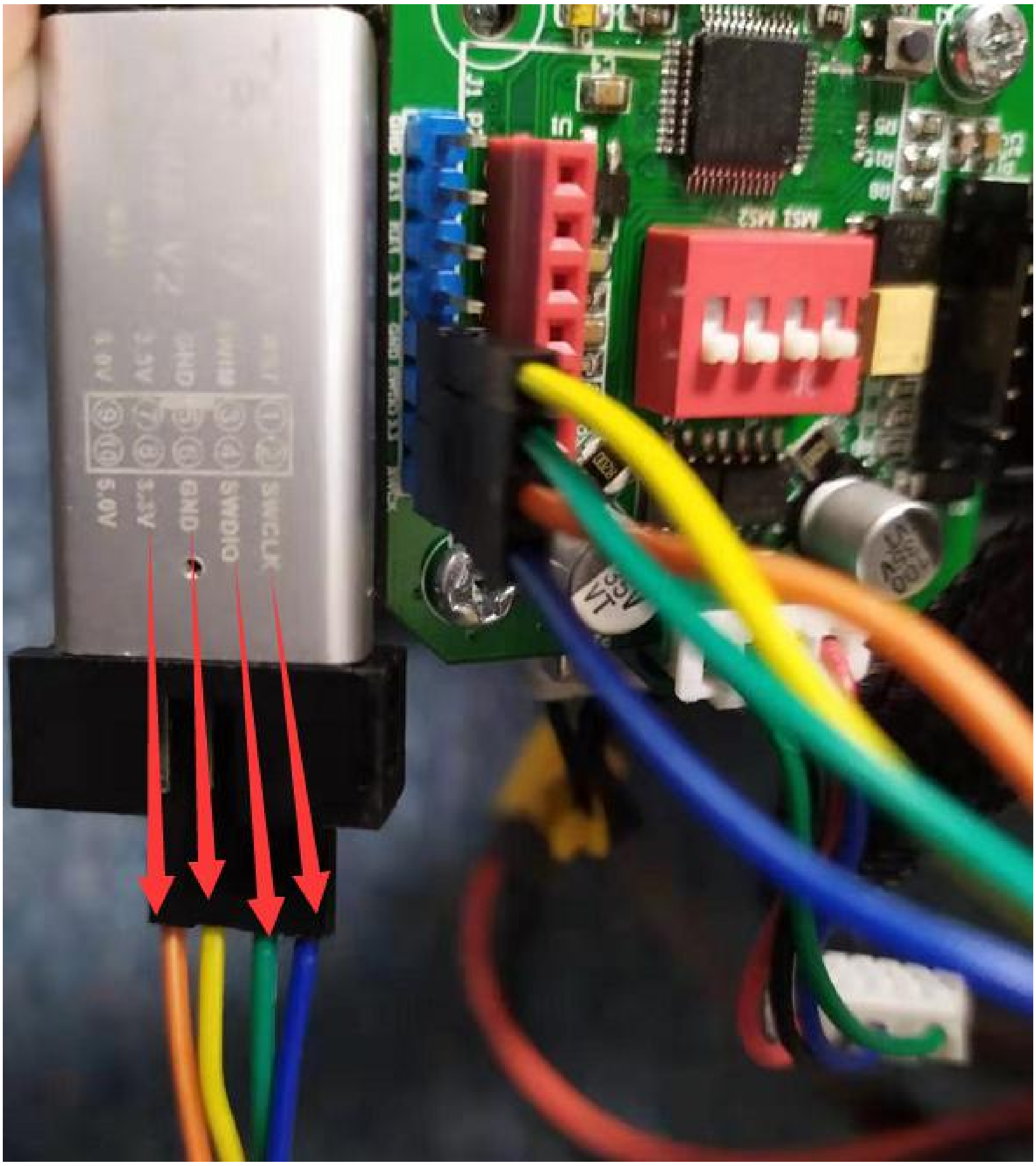
Ⅱ. Development tool installation

1) https://github.com/bigtreetech/Document/blob/master/How%20to%20install%20VScode%2BPlatformio.md

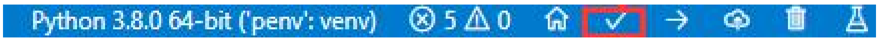
Login the above link to complete the PlatformIO tool installation.

2) After the above installation is completed, you can use an emulator (jlink or ST-Link or other emulator) to connect our closed-loop drive module for programming and online debugging! Here, we take ST-link's SWD programming simulation mode as an example, and J-link does the same.

Hardware connection, as shown in the figure:

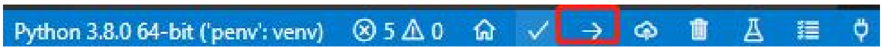


3) Download the firmware source file from https://github.com/ and open the project file in PlatformIO

4) Click 

,

Start compiling program

5) Click 

,

Complete the program burning

III. Marlin2.0 firmware configuration

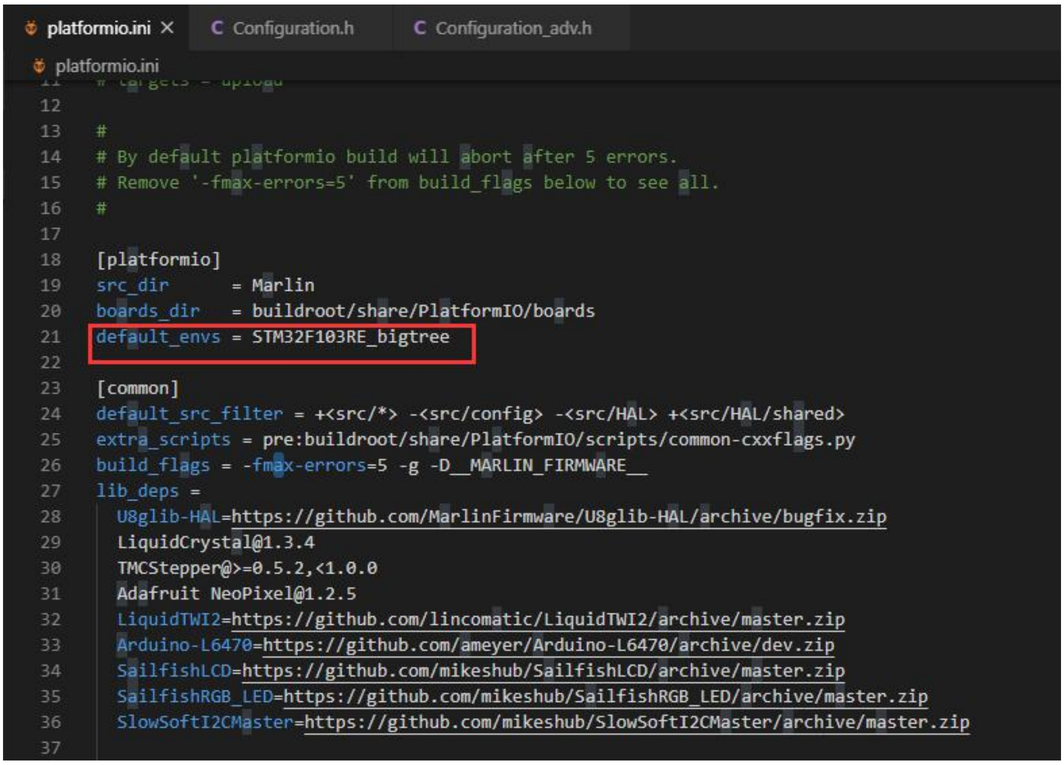
Take X, Y axis closed-loop drive module, Z axis TMC2209 UART mode, extruder E0, TMC2209 UART mode as examples, as follows:

main control board： BOARD_BIGTREE_SKR_E3_DIP

Closed-loop driver board: 42 stepper motor closed-loop driver board

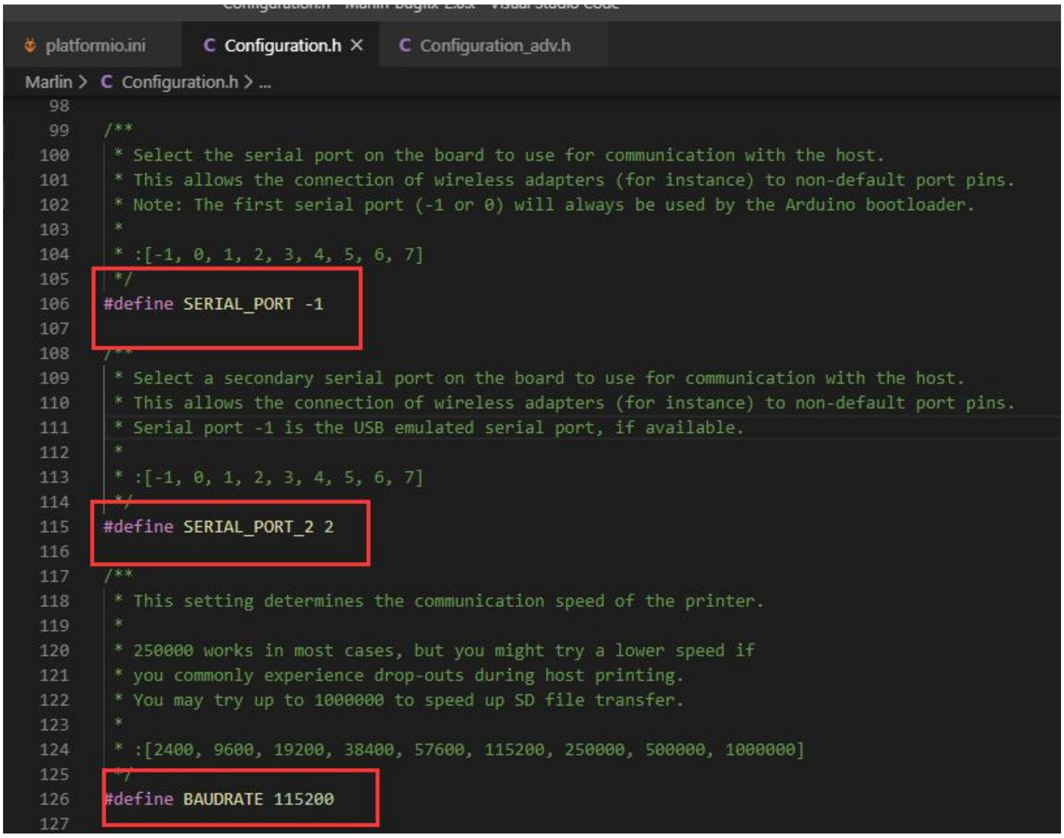
display screen： BIGTREETECH TFT35V3.0

The configuration is as follows: modify the board's environment



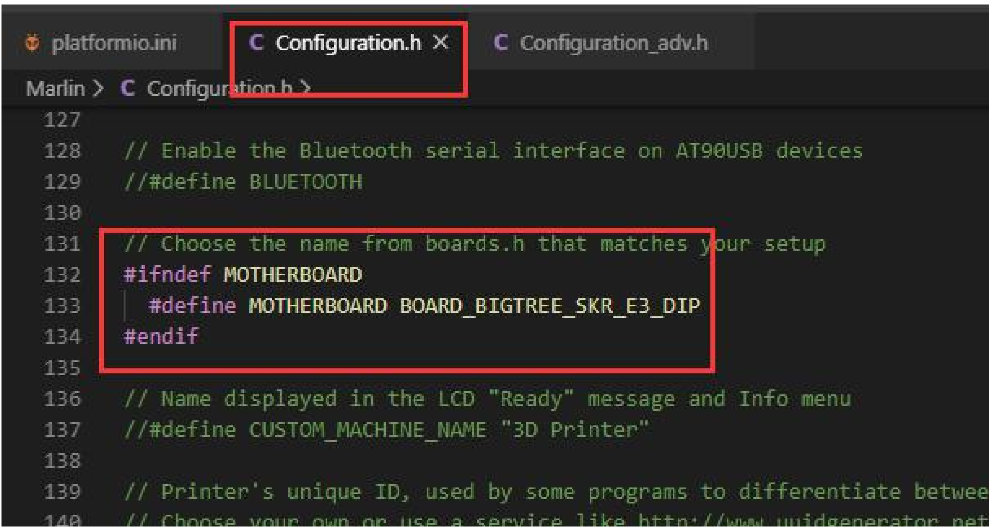
```
# targets = upload

#
# By default platformio build will abort after 5 errors.
# Remove '-fmax-errors=5' from build_flags below to see all.
#

[platformio]
src_dir      = Marlin
boards_dir   = buildroot/share/PlatformIO/boards
default_envs = STM32F103RE_bigtree

[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__
lib_deps =
  U8glib-HAL=https://github.com/MarlinFirmware/U8glib-HAL/archive/bugfix.zip
  LiquidCrystal@1.3.4
  TMCStepper@>=0.5.2,<1.0.0
  Adafruit NeoPixel@1.2.5
  LiquidTWI2=https://github.com/lincomatic/LiquidTWI2/archive/master.zip
  Arduino-L6470=https://github.com/ameyer/Arduino-L6470/archive/dev.zip
  SailfishLCD=https://github.com/mikeshub/SailfishLCD/archive/master.zip
  SailfishRGB_LED=https://github.com/mikeshub/SailfishRGB_LED/archive/master.zip
  SlowSoftI2CMaster=https://github.com/mikeshub/SlowSoftI2CMaster/archive/master.zip

```

Set serial port number and baud rate


Configuration.h - Marlin-bugfix-2.0.x - Visual Studio Code
platformio.ini
Configuration.h
Configuration_adv.h
Marlin > Configuration.h > ...

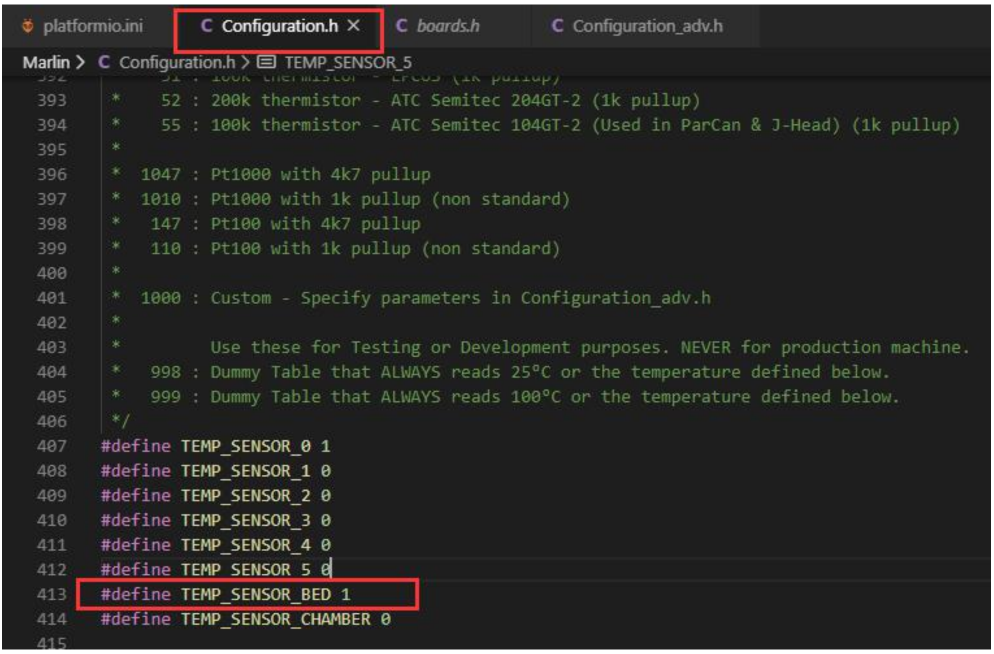
```

/**
 * Select the serial port on the board to use for communication with the host.
 * This allows the connection of wireless adapters (for instance) to non-default port pins.
 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
 *
 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
 */
#define SERIAL_PORT -1

/**
 * Select a secondary serial port on the board to use for communication with the host.
 * This allows the connection of wireless adapters (for instance) to non-default port pins.
 * Serial port -1 is the USB emulated serial port, if available.
 *
 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
 */
#define SERIAL_PORT_2 2

/**
 * This setting determines the communication speed of the printer.
 *
 * 250000 works in most cases, but you might try a lower speed if
 * you commonly experience drop-outs during host printing.
 * You may try up to 1000000 to speed up SD file transfer.
 *
 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 */
#define BAUDRATE 115200

```

Select board pin configuration file



```
// 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_BIGTREE_SKR_E3_DIP
#endif

// Name displayed in the LCD "Ready" message and Info menu
//#define CUSTOM_MACHINE_NAME "3D Printer"

// Printer's unique ID, used by some programs to differentiate betwee
// Choose your own or use a service like http://www.uuidgenerator.net
```

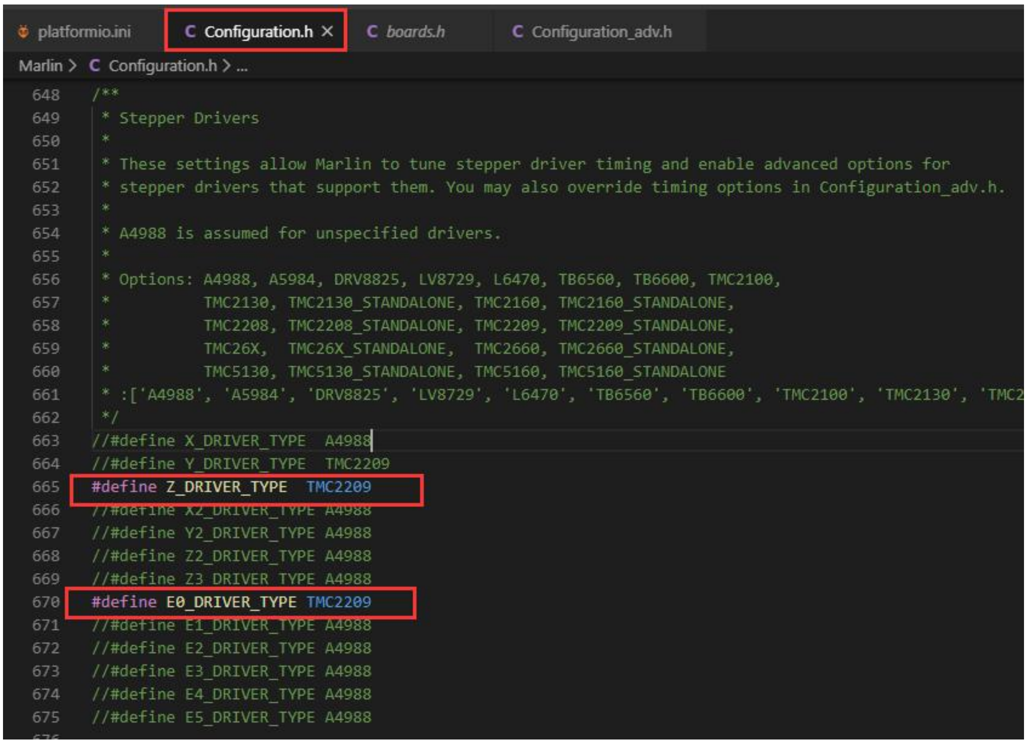
Turn on the hot bed (default heating rod 0 is turned on)



```
 *    51 : 100k thermistor - EPCOS (1k pullup)
 *    52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
 *    55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
 *
 *  1047 : Pt1000 with 4k7 pullup
 *  1010 : Pt1000 with 1k pullup (non standard)
 *   147 : Pt100 with 4k7 pullup
 *   110 : Pt100 with 1k pullup (non standard)
 *
 *  1000 : Custom - Specify parameters in Configuration_adv.h
 *
 *         Use these for Testing or Development purposes. NEVER for production machine.
 *   998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
 *   999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
 */
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_5 0
#define TEMP_SENSOR_BED 1
#define TEMP_SENSOR_CHAMBER 0
```

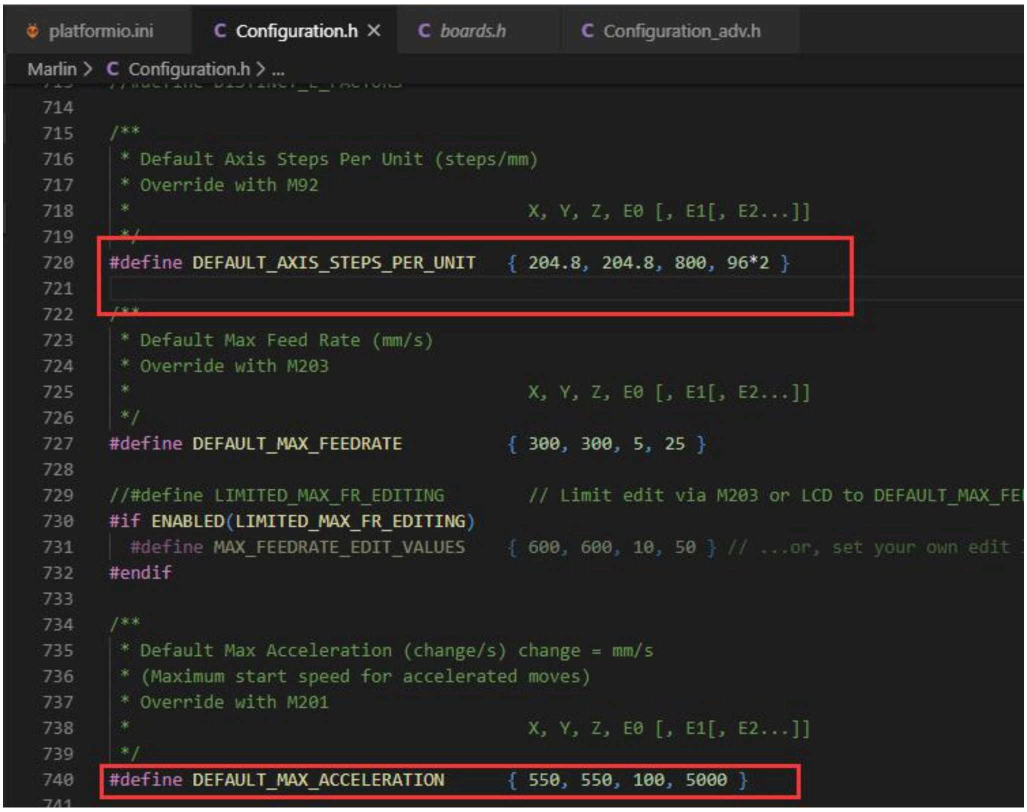
Enable TMC2209 UART

mode



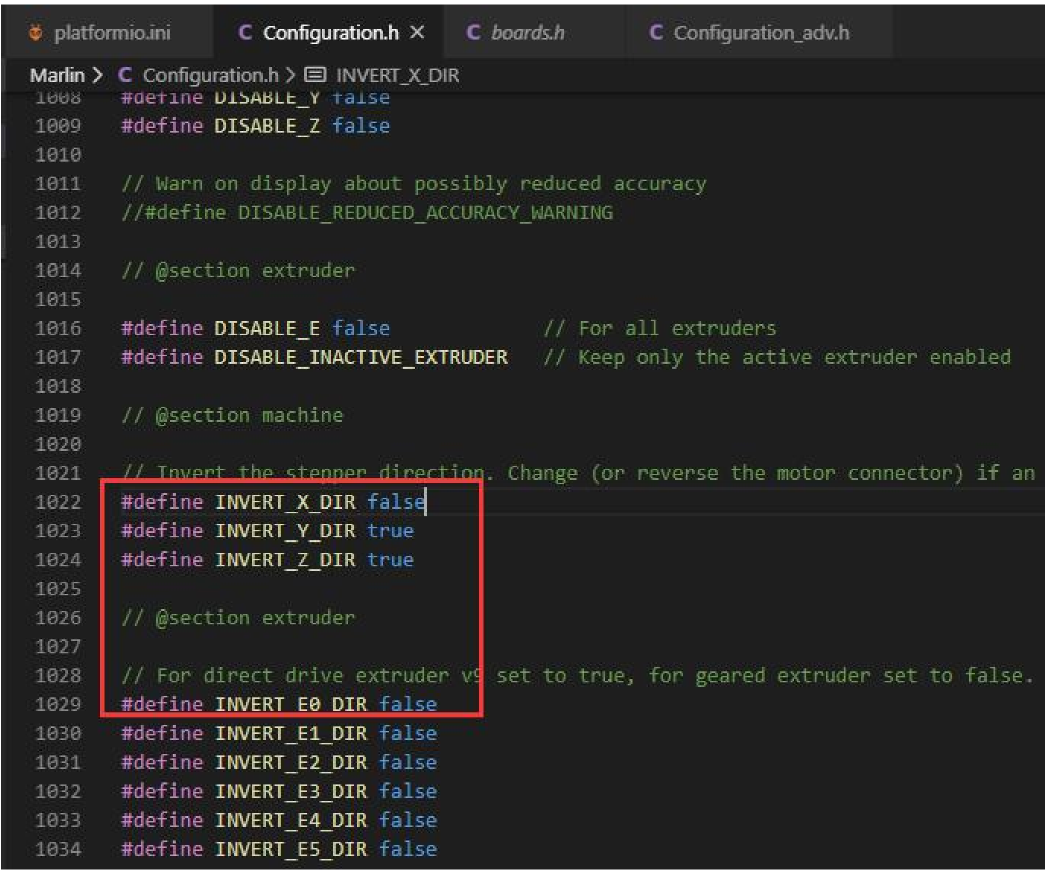
```
/**
 * Stepper Drivers
 *
 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 *
 * A4988 is assumed for unspecified drivers.
 *
 * Options: A4988, A5984, DRV8825, LV8729, L6470, 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', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2
 */
//#define X_DRIVER_TYPE  A4988
//#define Y_DRIVER_TYPE  TMC2209
#define Z_DRIVER_TYPE  TMC2209
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2209
//#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
```

Set the subdivision and maximum acceleration



```

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *                                      X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT   { 204.8, 204.8, 800, 96*2 }

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 *                                      X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE          { 300, 300, 5, 25 }

//#define LIMITED_MAX_FR_EDITING        // Limit edit via M203 or LCD to DEFAULT_MAX_FE
#if ENABLED(LIMITED_MAX_FR_EDITING)
  #define MAX_FEEDRATE_EDIT_VALUES    { 600, 600, 10, 50 } // ...or, set your own edit
#endif

/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 *                                      X, Y, Z, E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_ACCELERATION      { 550, 550, 100, 5000 }
```

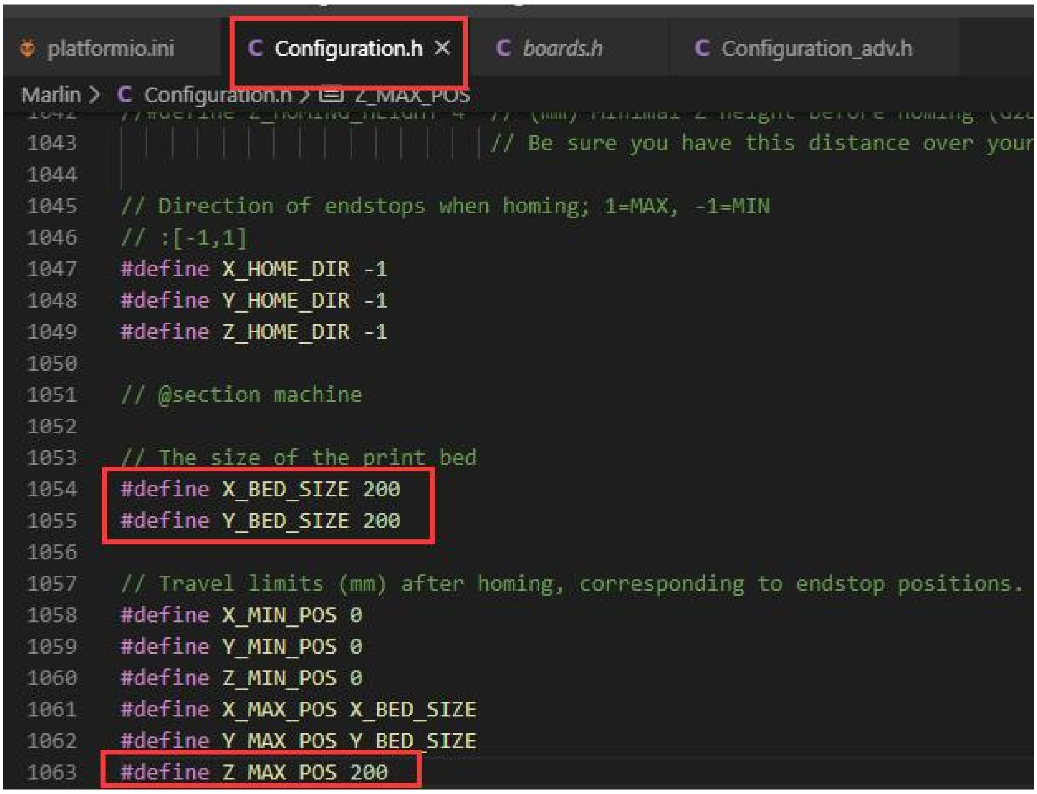
Set the direction of Z and



```
#define DISABLE_Y false
#define DISABLE_Z false

// Warn on display about possibly reduced accuracy
//#define DISABLE_REDUCED_ACCURACY_WARNING

// @section extruder

#define DISABLE_E false              // For all extruders
#define DISABLE_INACTIVE_EXTRUDER    // Keep only the active extruder enabled

// @section machine

// Invert the stepper direction. Change (or reverse the motor connector) if an
#define INVERT_X_DIR false
#define INVERT_Y_DIR true
#define INVERT_Z_DIR true

// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false
#define INVERT_E5_DIR false
```

E0

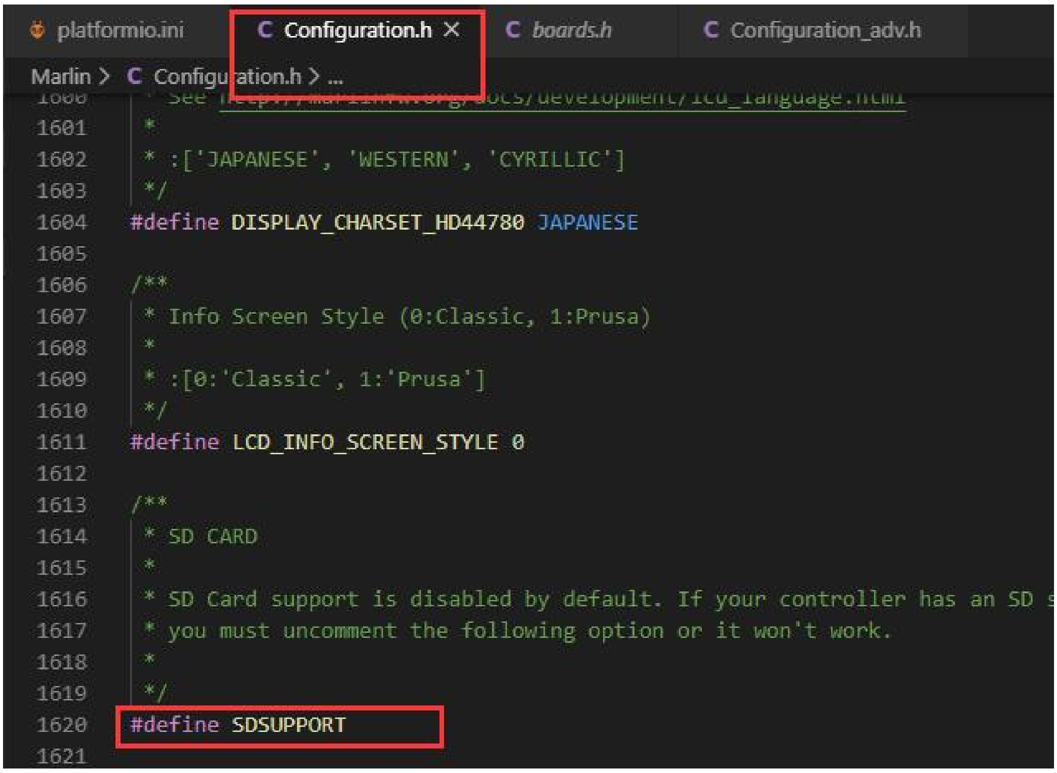
Set the size of the hot bed, by default 200mm, subject to the actual.



```
//#define Z_HOMING_HEIGHT 4  // (mm) Minimal Z height before homing (G28
                              // Be sure you have this distance over your

// Direction of endstops when homing; 1=MAX, -1=MIN
// :[-1,1]
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1

// @section machine

// The size of the print bed
#define X_BED_SIZE 200
#define Y_BED_SIZE 200

// Travel limits (mm) after homing, corresponding to endstop positions.
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS X_BED_SIZE
#define Y_MAX_POS Y_BED_SIZE
#define Z_MAX_POS 200
```

Turn on the SD card

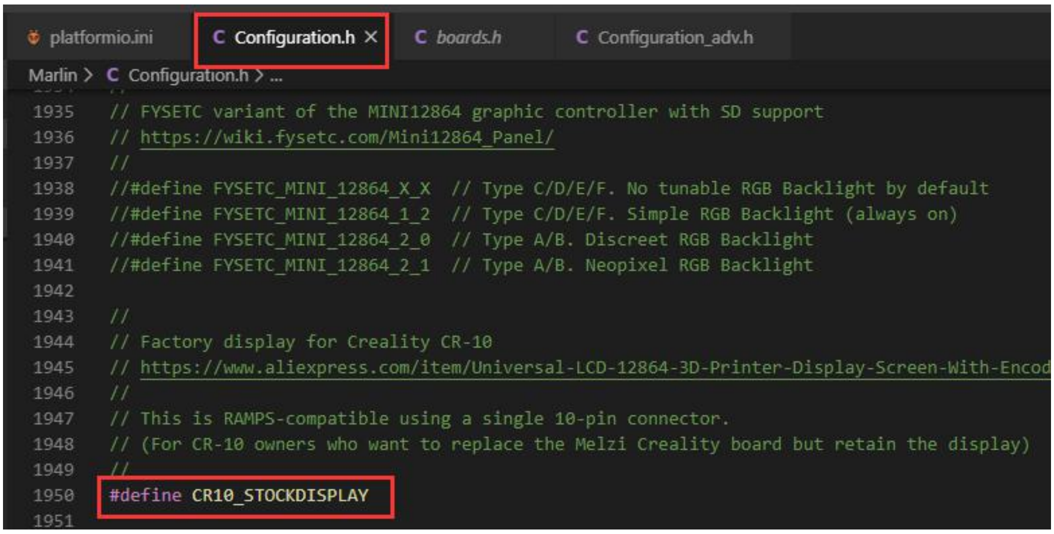
function



```
 * See http://marlinfw.org/docs/development/lcd_language.html
 *
 * :['JAPANESE', 'WESTERN', 'CYRILLIC']
 */
#define DISPLAY_CHARSET_HD44780 JAPANESE

/**
 * Info Screen Style (0:Classic, 1:Prusa)
 *
 * :[0:'Classic', 1:'Prusa']
 */
#define LCD_INFO_SCREEN_STYLE 0

/**
 * SD CARD
 *
 * SD Card support is disabled by default. If your controller has an SD s
 * you must uncomment the following option or it won't work.
 *
 */
#define SDSUPPORT
```

Open screen

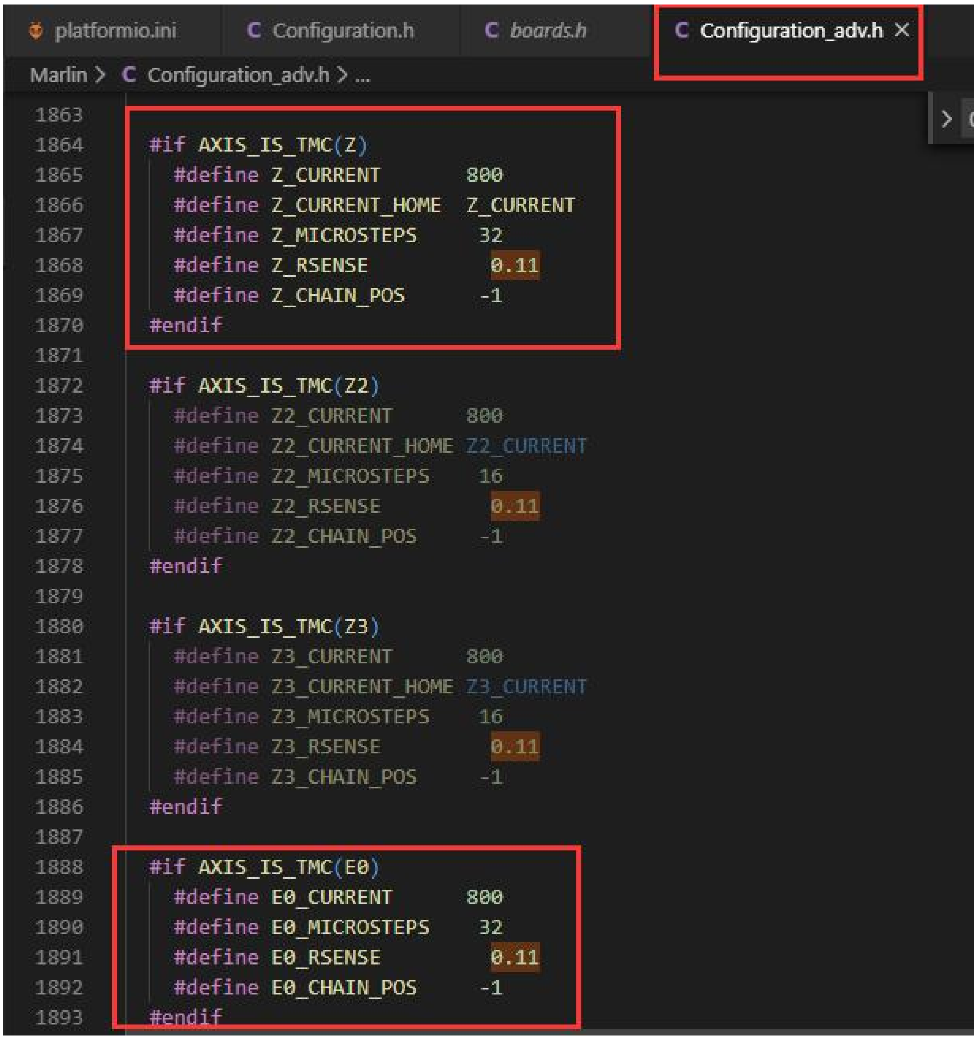
display



```
// FYSETC variant of the MINI12864 graphic controller with SD support
// https://wiki.fysetc.com/Mini12864_Panel/
//
//#define FYSETC_MINI_12864_X_X  // Type C/D/E/F. No tunable RGB Backlight by default
//#define FYSETC_MINI_12864_1_2  // Type C/D/E/F. Simple RGB Backlight (always on)
//#define FYSETC_MINI_12864_2_0  // Type A/B. Discreet RGB Backlight
//#define FYSETC_MINI_12864_2_1  // Type A/B. Neopixel RGB Backlight

//
// Factory display for Creality CR-10
// https://www.aliexpress.com/item/Universal-LCD-12864-3D-Printer-Display-Screen-With-Encod
//
// This is RAMPS-compatible using a single 10-pin connector.
// (For CR-10 owners who want to replace the Melzi Creality board but retain the display)
//
#define CR10_STOCKDISPLAY
```

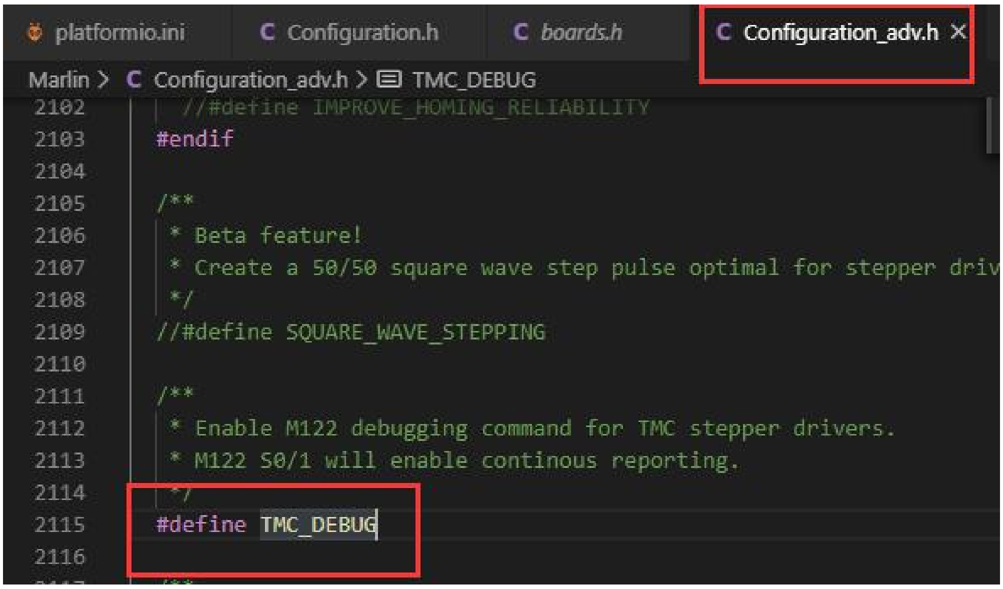
Set the Z axis and E0 current and

subdivision



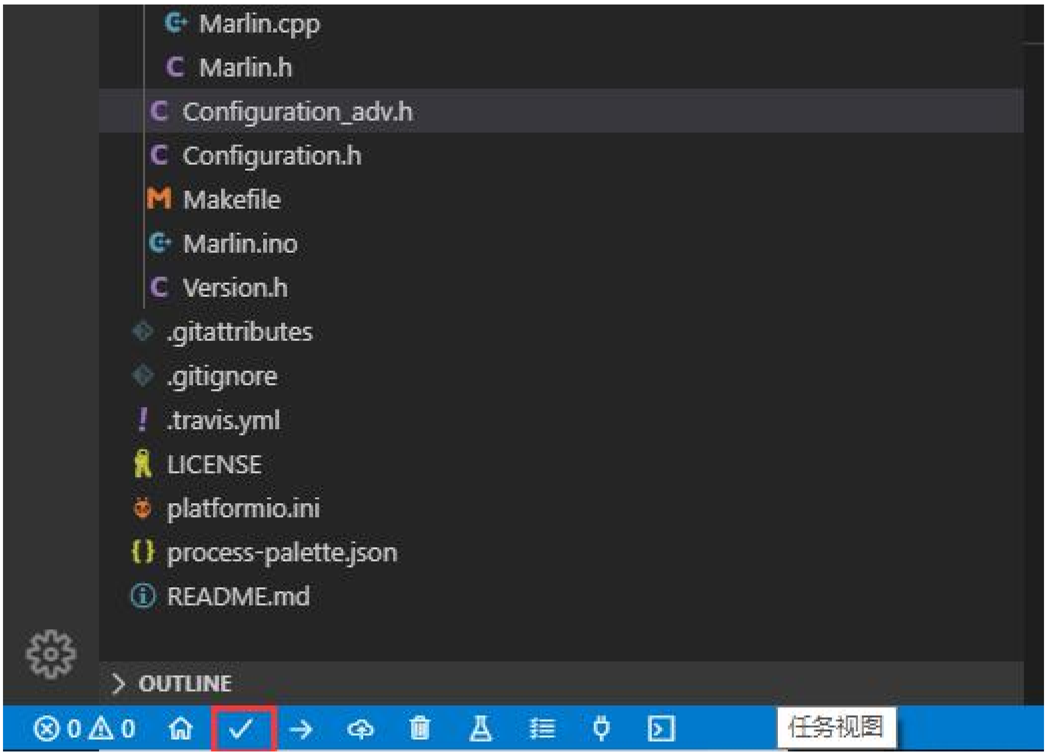
```

#if AXIS_IS_TMC(Z)
  #define Z_CURRENT       800
  #define Z_CURRENT_HOME  Z_CURRENT
  #define Z_MICROSTEPS     32
  #define Z_RSENSE          0.11
  #define Z_CHAIN_POS      -1
#endif

#if AXIS_IS_TMC(Z2)
  #define Z2_CURRENT      800
  #define Z2_CURRENT_HOME Z2_CURRENT
  #define Z2_MICROSTEPS    16
  #define Z2_RSENSE         0.11
  #define Z2_CHAIN_POS     -1
#endif

#if AXIS_IS_TMC(Z3)
  #define Z3_CURRENT      800
  #define Z3_CURRENT_HOME Z3_CURRENT
  #define Z3_MICROSTEPS    16
  #define Z3_RSENSE         0.11
  #define Z3_CHAIN_POS     -1
#endif

#if AXIS_IS_TMC(E0)
  #define E0_CURRENT      800
  #define E0_MICROSTEPS    32
  #define E0_RSENSE         0.11
  #define E0_CHAIN_POS     -1
#endif
```

Turn on the TMC_Debug function, which is convenient to view the driver exception information with the serial debugging assistant

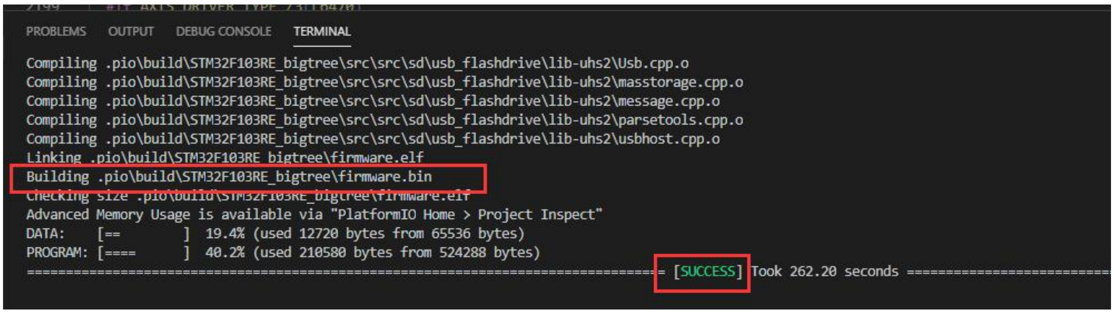


```
  //#define IMPROVE_HOMING_RELIABILITY
#endif

/**
 * Beta feature!
 * Create a 50/50 square wave step pulse optimal for stepper driv
 */
//#define SQUARE_WAVE_STEPPING

/**
 * Enable M122 debugging command for TMC stepper drivers.
 * M122 S0/1 will enable continous reporting.
 */
#define TMC_DEBUG
```

Once this is done, click the compile button in the lower left corner of the compiler to start compiling

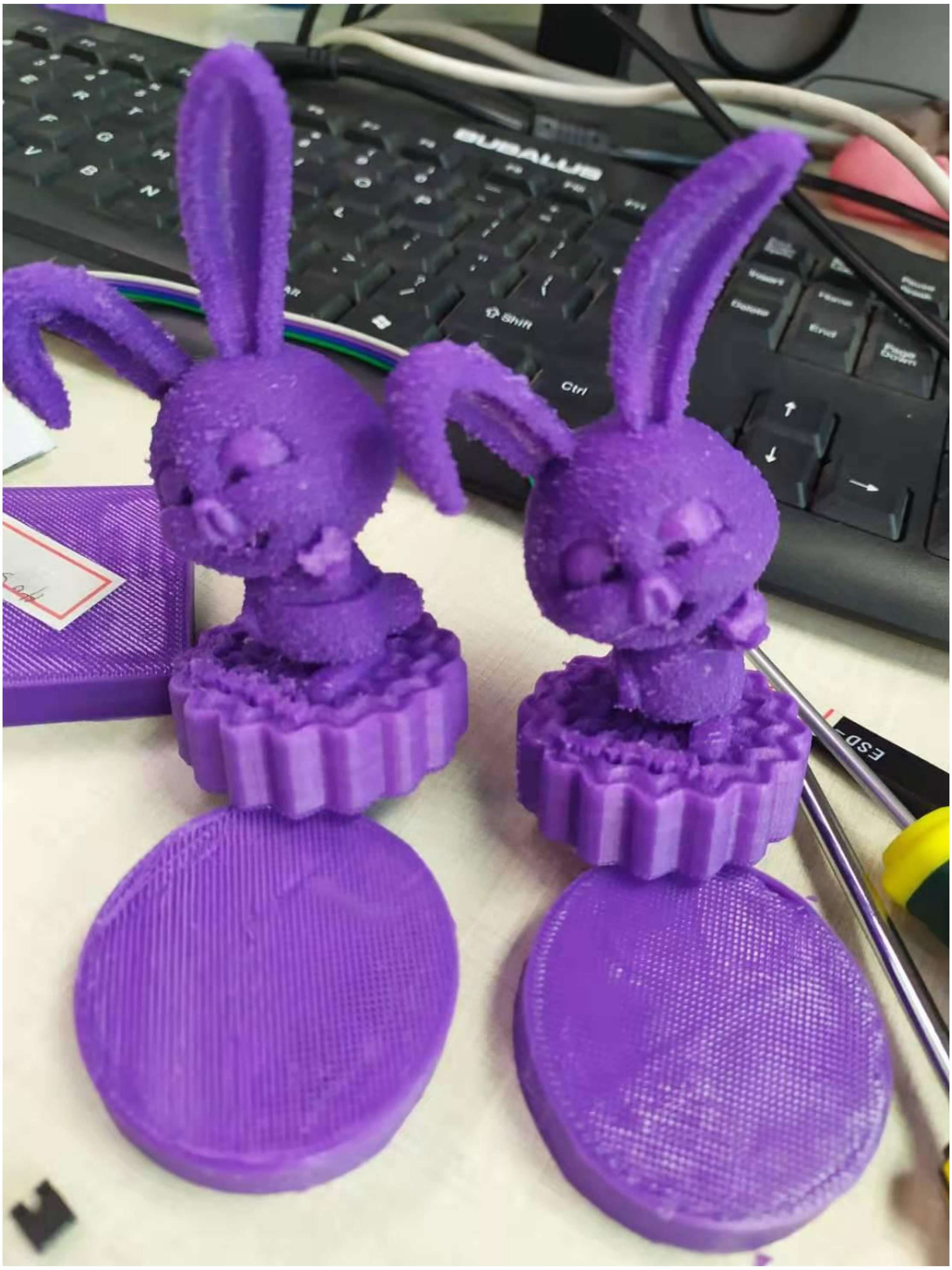


After successful compilation, directly find the firmware file in .pio\build\STM32F103RE_bigtree, copy the file into the TF card, then insert the TF card into the motherboard, press the reset key to complete firmware update.

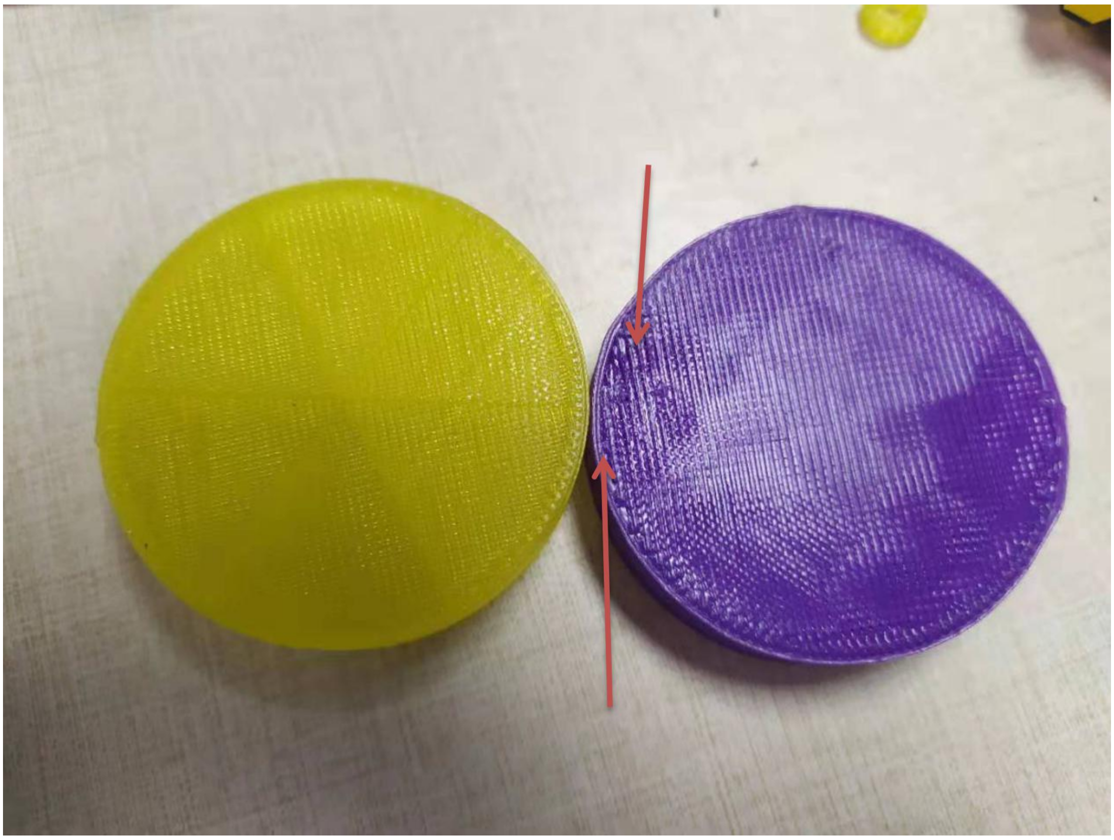


```
Compiling .pio\build\STM32F103RE_bigtree\src\src\sd\usb_flashdrive\lib-uhs2\Usb.cpp.o
Compiling .pio\build\STM32F103RE_bigtree\src\src\sd\usb_flashdrive\lib-uhs2\masstorage.cpp.o
Compiling .pio\build\STM32F103RE_bigtree\src\src\sd\usb_flashdrive\lib-uhs2\message.cpp.o
Compiling .pio\build\STM32F103RE_bigtree\src\src\sd\usb_flashdrive\lib-uhs2\parsetools.cpp.o
Compiling .pio\build\STM32F103RE_bigtree\src\src\sd\usb_flashdrive\lib-uhs2\usbhost.cpp.o
Linking .pio\build\STM32F103RE_bigtree\firmware.elf
Building .pio\build\STM32F103RE_bigtree\firmware.bin
Checking size .pio\build\STM32F103RE_bigtree\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
DATA:    [==        ]  19.4% (used 12720 bytes from 65536 bytes)
PROGRAM: [====      ]  40.2% (used 210580 bytes from 524288 bytes)
=============================================================================== [SUCCESS] Took 262.20 seconds =========================
```

Ⅳ. Bad printing example



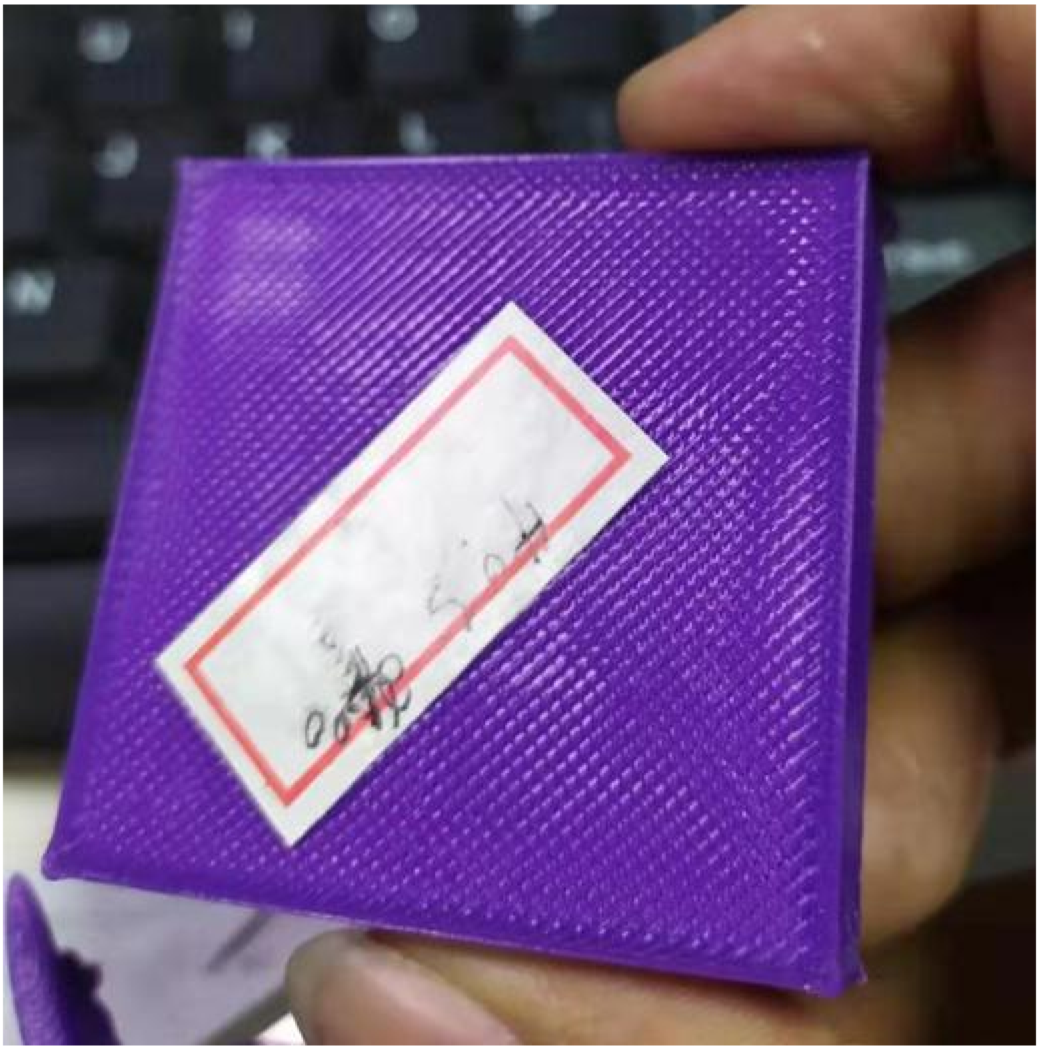
Reason: The belt is relatively loose, resulting in unstable structure and misaligned prints.



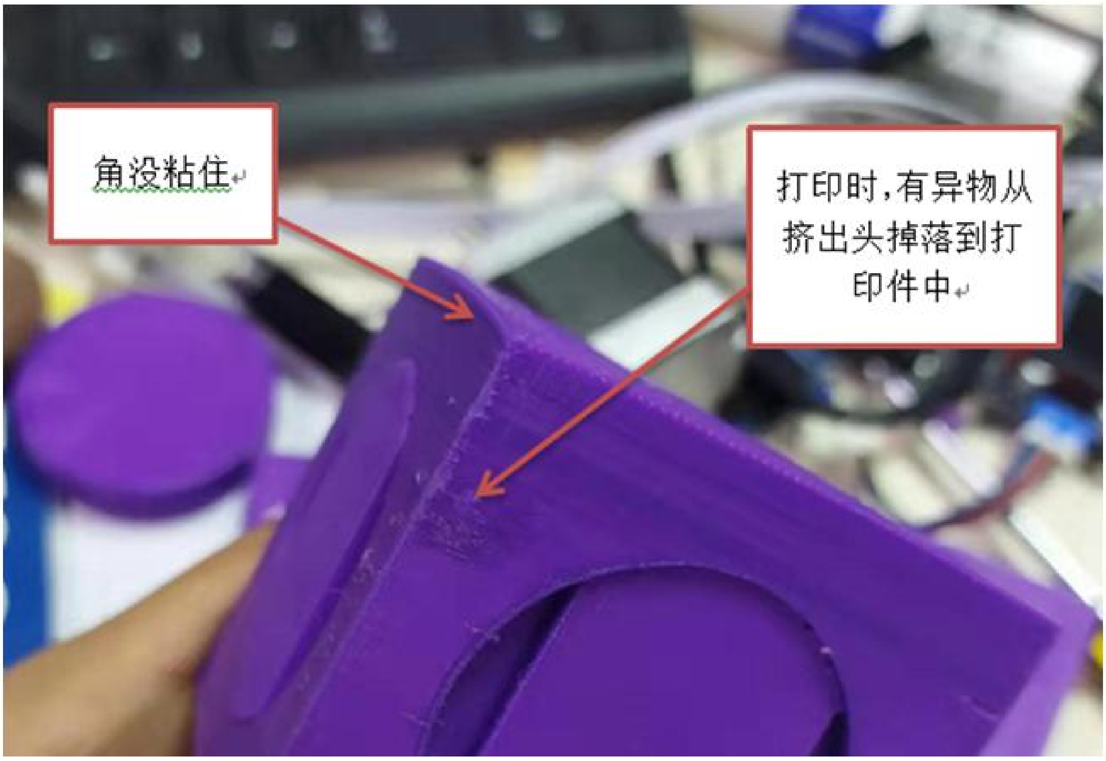
Reason: The belt is relatively loose, resulting in unstable structure and misaligned prints.

fine

printing



When printing, the corners are not stuck, or there is a foreign object falling from the extrusion head into the print.



The above problems are common problems encountered in debugging printing, mainly due to poor structure stability. If you have other questions, please contact our after-sales staff!