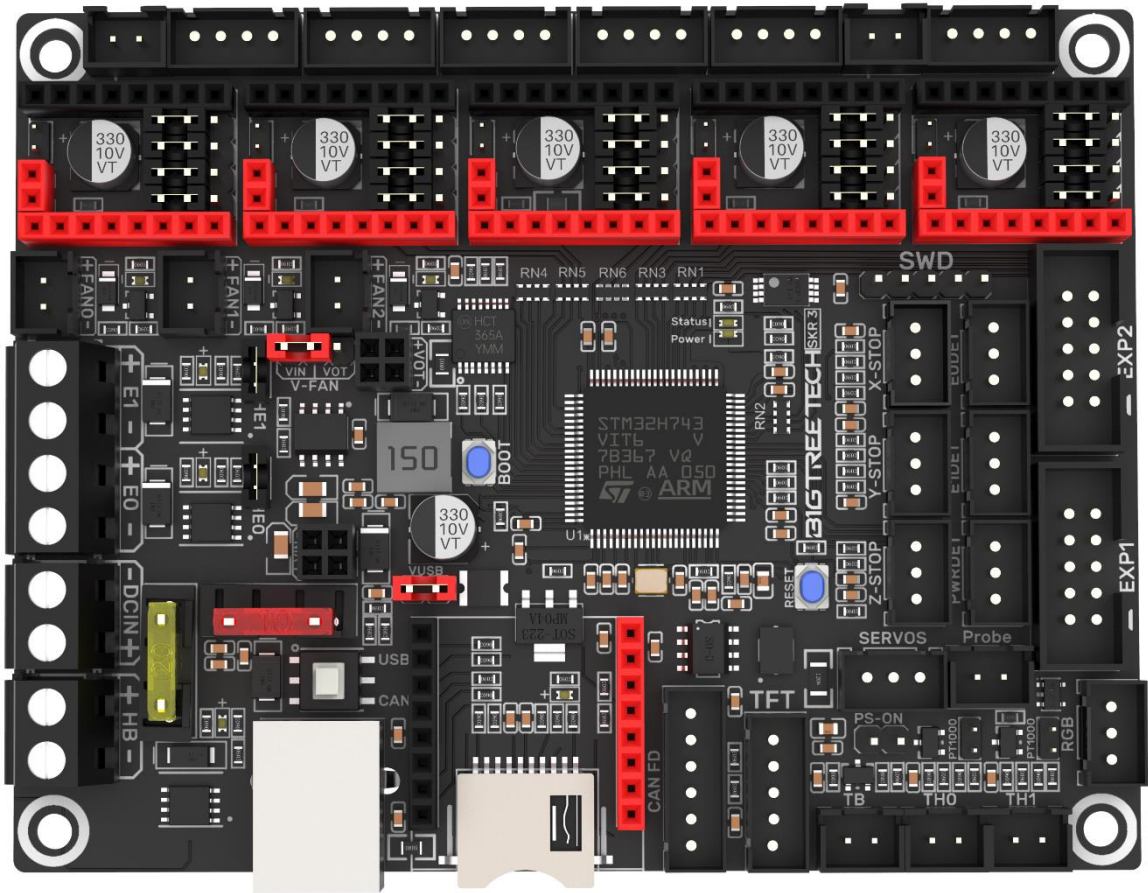


BIGTREETECH

SKR 3

User manual



Content

Content	2
Revision History	5
1. Introduction	6
1.1 Features	6
1.2 Specification	7
1.3 Firmware	8
1.4 Dimensions	8
2. Peripheral Port	9
2.1 Connector diagrams	9
2.2 Pinout diagram	9
3. Connection description	10
3.1 USB power supply	10
3.2 Stepper driver	10
3.2.1 STEP/DIR(STANDALONE) mode	10
3.2.2 UART mode of TMC driver	12
3.2.3 TMC driver SPI mode	12
3.2.4 TMC driver DIAG(Sensorless Homing)	12
3.3 USB and CAN mode	13
3.4 Voltage selection for CNC Fan	13
3.5 100K NTC or PT1000 setting	14
3.6 BLTouch wiring	14
3.7 Auto power off (Relay V1.2) wiring	15
3.8 Power loss recovery (UPS 24V V1.0) wiring	15
3.9 RGB wiring	16
3.10 Filament sensor wiring	16
3.11 Display wiring	17

3.12 Heater cartridge IO	17
4. Marlin	18
4.1 install compiling environment	18
4.2 Download Marlin firmware	18
4.3 Configure firmware	18
4.3.1 Open Marlin project	18
4.3.2 Compiling environment	19
4.3.3 Configure motherboad and serial port	19
4.3.4 Configure stepper driver	21
4.3.5 Sensorless homing	22
4.3.6 100K NTC or PT1000	23
4.3.7 BLTouch	23
4.3.8 Auto power off(Relay V1.2)	26
4.3.9 Power loss revoverly	26
4.3.10 RGB	27
4.3.11 Filament sensor	28
4.3.12 smart filament sensor(SFS V1.0 / V2.0)	29
4.3.13 ESP3D	30
4.4 Compile firmware	31
5. Klipper	32
5.1 Preparation	32
5.1.1 Download OS image	32
5.1.2 Download and install Raspberry Pi Imager	32
5.2 Write image	33
5.3 WIFI setting	35
5.4 ssh connect to raspberry pi	35
5.5 Compile firmware	37
5.6 Configure Klipper	38

6. Firmware update	39
7. Precautions	39
8. FAQ	40

Revision History

version	Note	Date
01.00	Original	2022/03/08
01.01	Add support for RRF	2022/05/21
01.02	Fix the wrong description of "1.2 Specification->12"	2022/07/19
01.03	Change "Heating rod" to "Heater cartridge"	2022/07/21
01.04	Add description of IO of Heater cartridge	2022/08/27

1. Introduction

The BIGTREETECH SKR 3 mother board is a brand new 32 bit motherboard developed by Shenzhen Big Tree Technology Co., Ltd that fixed all the problems of the V1.4/Turbo board with new features and improved functionality.

1.1 Features

1. Utilize 32bit 480MHz ARM Cortex-M7 series STM32H743VI MCU for massively improved performance
2. TPS5450-5A power supply chip, support DC12/24V power input, current output rated at 5A max continuous and 6A max instantaneous, sufficient power supply for Raspberry pi(3A requirement)
3. Onboard BOOT button to enable DFU mode to update bootloader
4. Thermistor circuit is protected to prevent MCU damage from shorted heated bed and heater cartridge connection.
5. Selectable voltage (24V, 12V, 5V by **SKR 3-DC MODE**) for CNC fan (**Note:** The voltages of the 3 * CNC fans are unified, different voltages cannot be set separately for different ports), no more need for external stepdown thus prevent board damage from user error
6. Thermistor connection supports Pull up resistance value setting using jumpers, No more extra module needed for PT1000
7. Supports all models of Serial TFT, SPI TFT, and LCD12864/2004 from BTT
8. Firmware installation from using MicroSD card, simple and efficient
9. Integrated SPI and UART mode of TMC driver and DIAG pin, easily configurable with jumpers
10. Supports power loss recovery, filament runout sensor, Auto poweroff, BLTouch, RGB led etc.
11. High efficiency MOSFET for less heat generation
12. Replaceable fuse for easy maintenance
13. WIFI module (ESP-12S, ESP-07, ESP32) supported
14. Onboard SDIO MicroSD slot for Higher transfer rate
15. Onboard EEPROM for setting storage

16. 2 method for CAN connection:USB & XH2.54 6Pin, USB port CAN/USB mode is selected by switches.

1.2 Specification

1. Dimention: 110*85mm, for details please refer to: **BIGTREETECH SKR 3-SIZE.pdf**
2. Mounting pattern: 102*76mm
3. MCU: ARM Cortex-M7 STM32H743VI
4. EEPROM: 24C32 32Kbit
5. Voltage in: DC12V-DC24V
6. Logic voltage: DC 3.3V
7. Heater connection: Heated bed (HB)、heater cartridge (E0、E1)
8. HB port max current: 10A continuous , 11A instantaneous
9. Heater cartridge max current: 5.5A continuous, 6A instantaneous
10. Fan port: 3 X CNC(selectable voltage) ,2 X always on(PSU voltage)
11. Fan port max current: 1A continuous, 1.5A instantaneous
12. The Total Current of Heating cartridges(E0, E1) + Driver + Fan: less than 10A
13. WIFI connection: ESP-12S、ESP-07S、ESP32
14. Expansion port: BLTouch (Servos、Probe)、PS-ON、PWR-DET、Fil-DET、RGB、CAN FD
15. Steppers supported: TMC5160、TMC2209、TMC2225、TMC2226、TMC2208、TMC2130、ST820、LV8729、DRV8825、A4988 etc.
16. Stepper driver mode: SPI、UART、STEP/DIR
17. Stepper motor socket: X、Y、Z (Parallel double Z-Axis)、E0、E1 5 channels in total
18. Thermistor: 2 X thermistor port(NTC/PTC selectable)
19. Display: Serial、SPI、LCD
20. PC connection: USB A

21. Supported file format: G-code

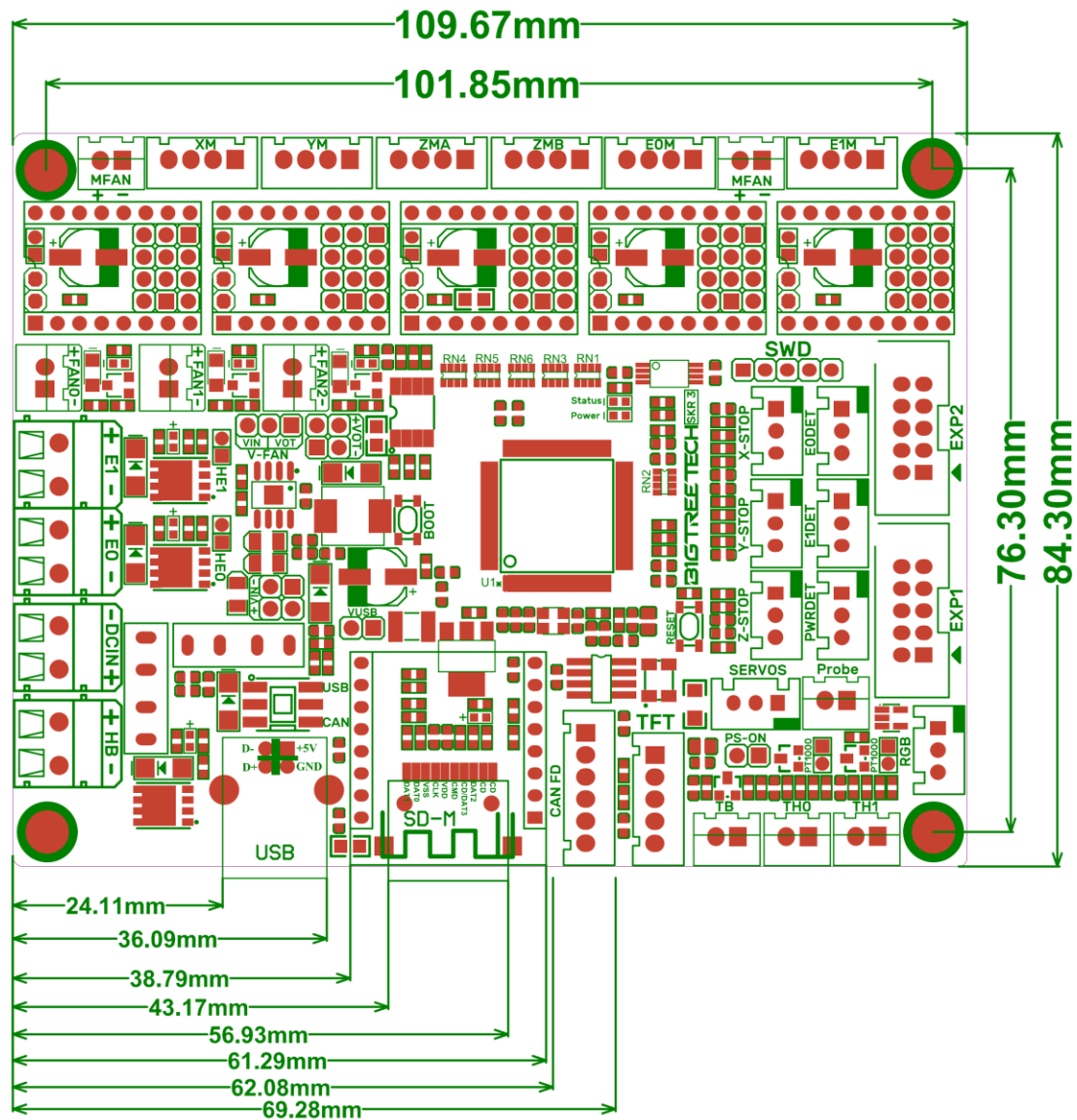
22. Supported kinematics: Cartesian、Delta、Kossel、Ultimaker、CoreXY

23. Recommended Slicer/Console: Cura、Simplify3D、Pronterface、Repetier-host、Makerware

1.3 Firmware

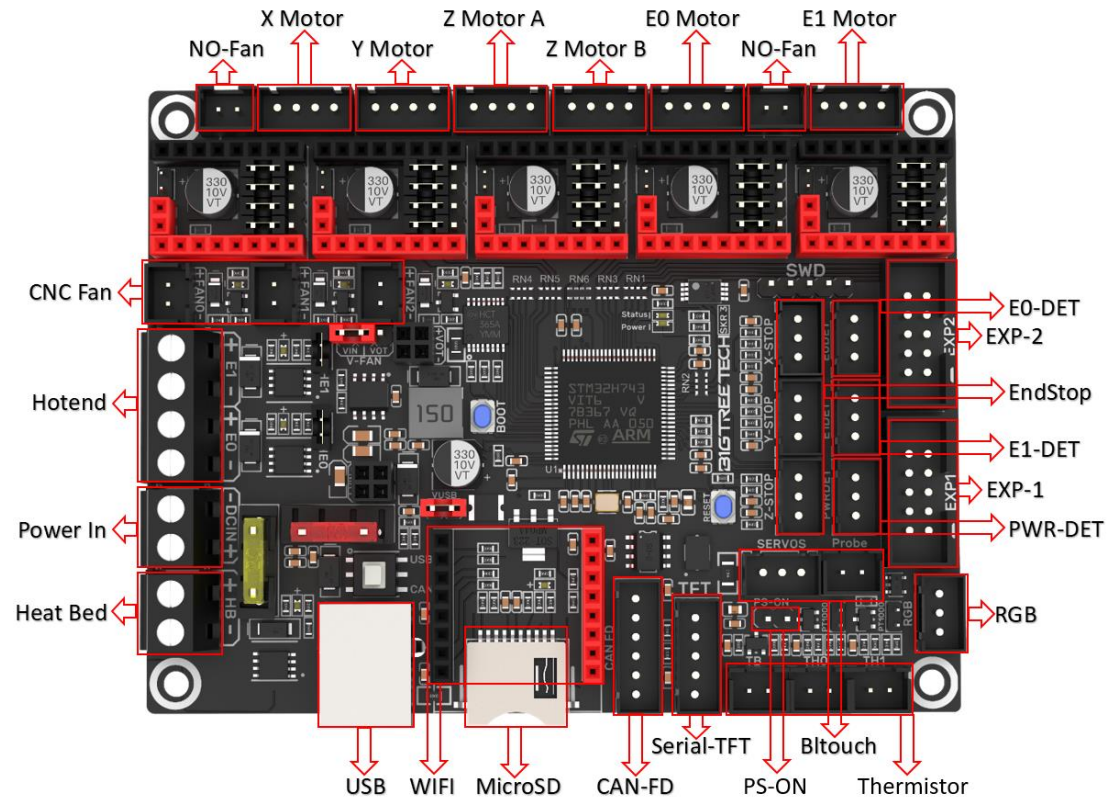
Supported Firmware: Marlin, Klipper, RRF

1.4 Dimensions

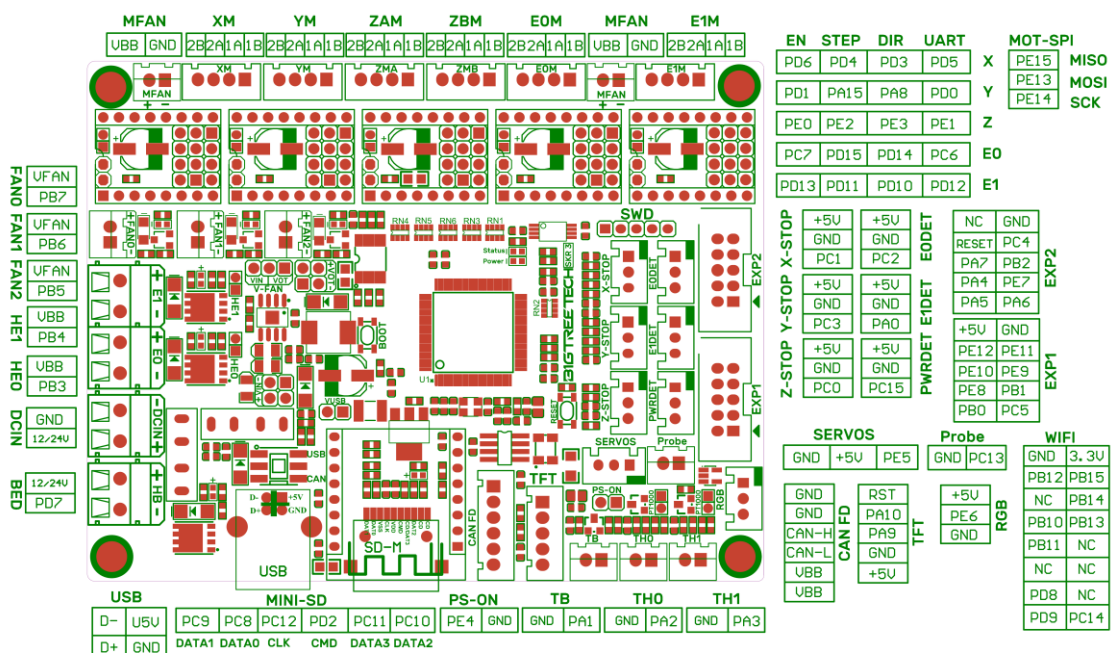


2. Peripheral Port

2.1 Connector diagrams



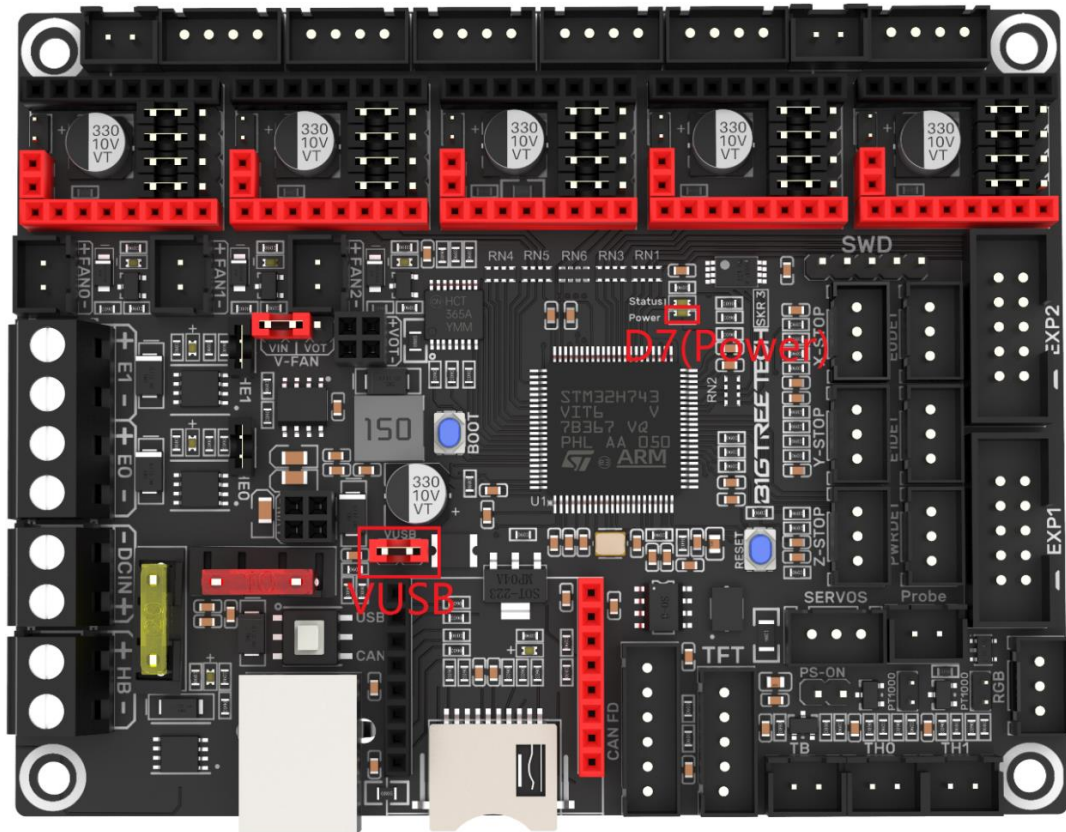
2.2 Pinout diagram



3. Connection description

3.1 USB power supply

After the SKR 3 board has been powered, the Red led D7(Power) to the upper right of the MCU will light up, indicating power on. When using only USB to power the board, Please insert the jumper cap onto the VUSB jumper.



Chips:	MS1	MS2	MS3	microstep	Excitation Mode
A4988 16 microstep max 35V 2A	L	L	L	Full Step	2 Phase
	H	L	L	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	H	H	L	1/8	2W1-2 Phase
	H	H	H	1/16	4W1-2 Phase
Current $R_S=0.1\Omega$	$I_{TripMAX} = \frac{V_{REF}}{8 * R_S}$				

Driver chips	MODE2	MODE1	MODE0	Microsteps	Excitation Mode
DRV8825 Maximum 32microsteps 8.2V-45V 2.5A at 24V T=25°C	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	
	H	L	L	1/16	
	H	L	H	1/32	
	H	H	L	1/32	
	H	H	H	1/32	
Current $R_{ISENSE}=0.1\Omega$	$I_{CHOP} = \frac{V_{(xREF)}}{5 * R_{ISENSE}}$				

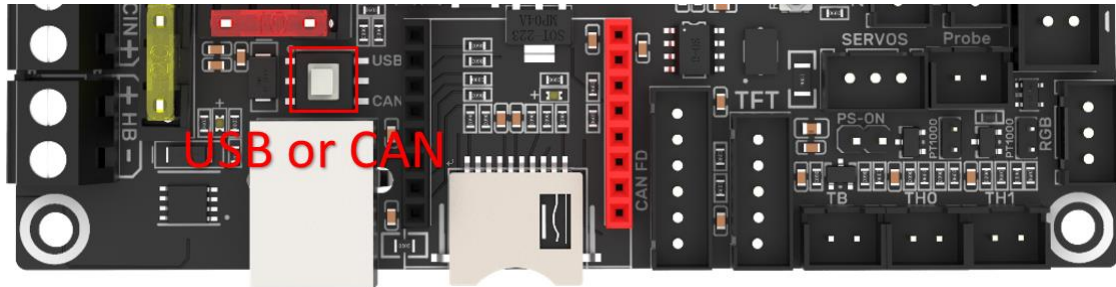
Driver chips	MD3	MD2	MD1	Microsteps	Excitation Mode
LV8729 Maximum 128microsteps 36V 1.8A	L	L	L	Full Step	2 Phase
	L	L	H	1/2	1-2 Phase
	L	H	L	1/4	W1-2 Phase
	L	H	H	1/8	2W1-2 Phase
	H	L	L	1/16	4W1-2 Phase
	H	L	H	1/32	8W1-2 Phase
	H	H	L	1/64	16W1-2 Phase
	H	H	H	1/128	32W1-2 Phase
Current $RF1=0.22\Omega$	$I_{OUT} = (V_{REF} / 5) / RF1$				

Driver chips	MS3	MS2	MS1	Microsteps
ST820 Maximum 256microsteps 45V 1.5A	L	L	L	Full Step
	L	L	H	1/2
	L	H	L	1/4
	L	H	H	1/8
	H	L	L	1/16
	H	L	H	1/32
	H	H	L	1/128
	H	H	H	1/256
Current $R_S=0.15\Omega$	$I_{peak} = \frac{V_{REF} * V_{DD}}{5 * R_S}$			

i.e: TMC2208、TMC2209、TMC2225 etc. Place jumpers according to the diagram below, microstep and current can be configured in firmware.

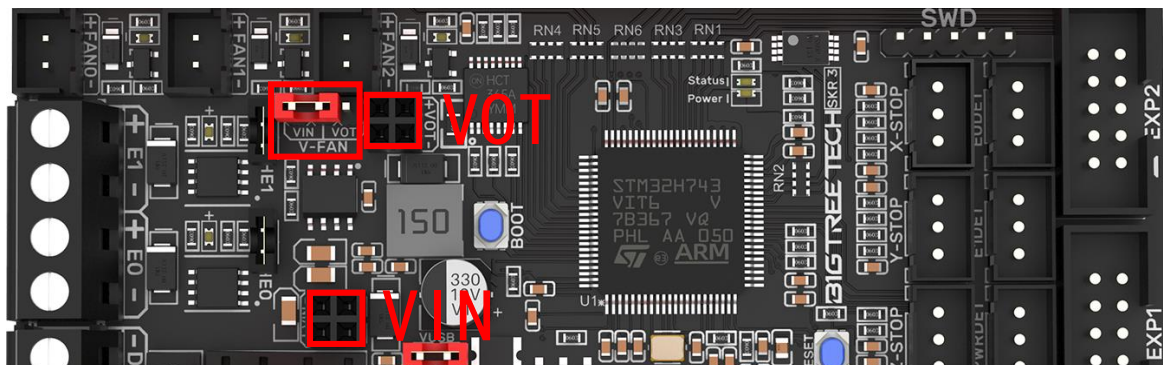
3.3 USB and CAN mode

When the button shown below is released, the board is in usb mode, when pressed down, the board is in CAN FD mode.

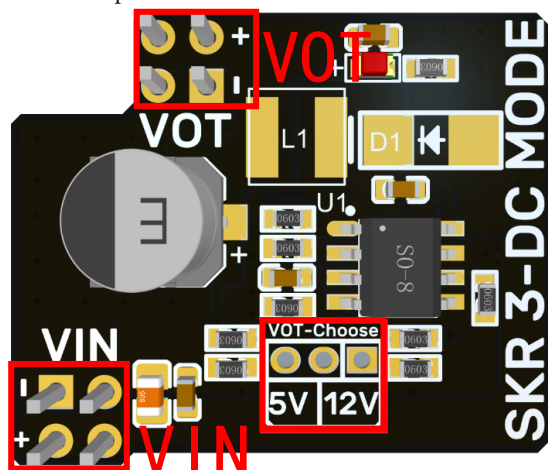


3.4 Voltage selection for CNC Fan

Connect jumper between the two pins of VIN if using DCIN as CNC fan voltage. Connect jumper between the two pins of VOT and insert SKR 3-DC MODE on to the 2*4 Pin sockets if 12V or 5V is desired (Note: The voltages of the 3*CNC fans are unified, different voltages cannot be set separately. i.e: the voltage of 3*CNC can be set to 24V, 12V or 5V at the same time, but it cannot be set to the combination of 24V+12V+5V).

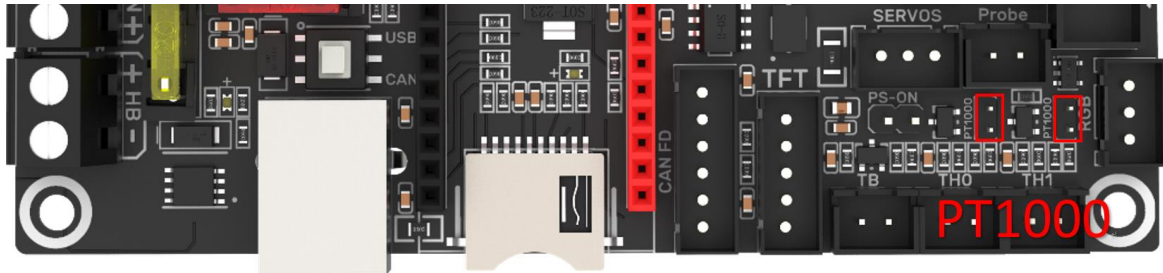


Voltage of the SKR 3-DC MODE is set by connecting jumpers on 5V or 12V on VOT-Choose pins.

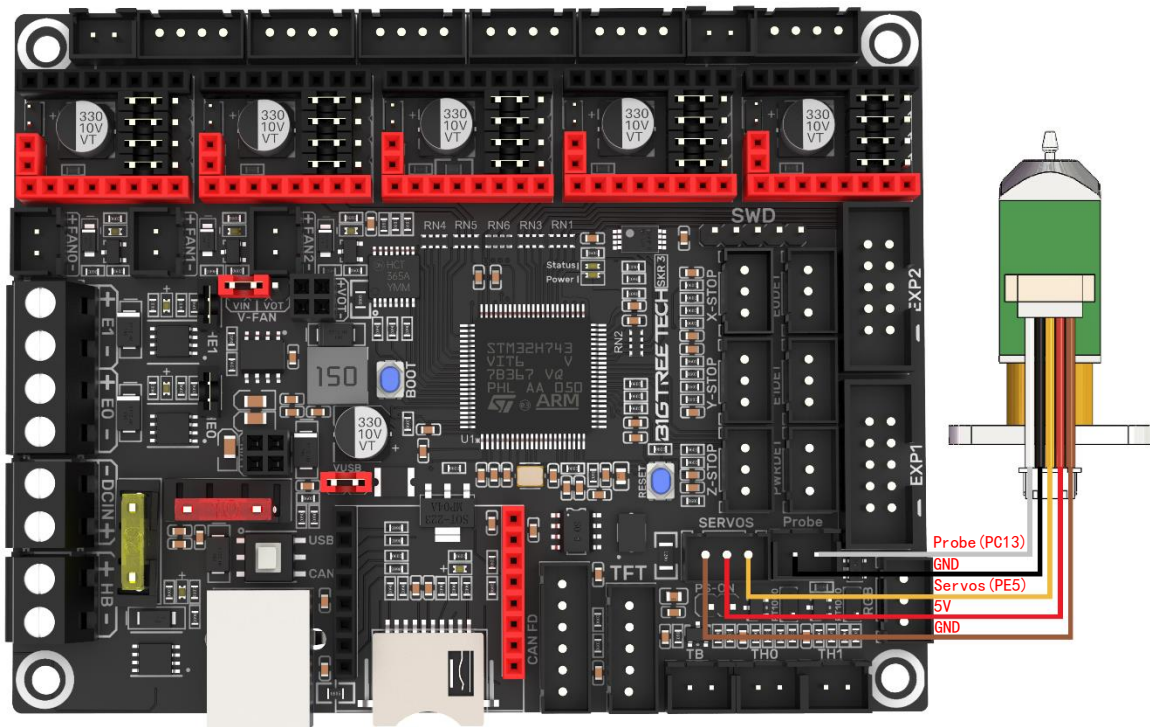


3.5 100K NTC or PT1000 setting

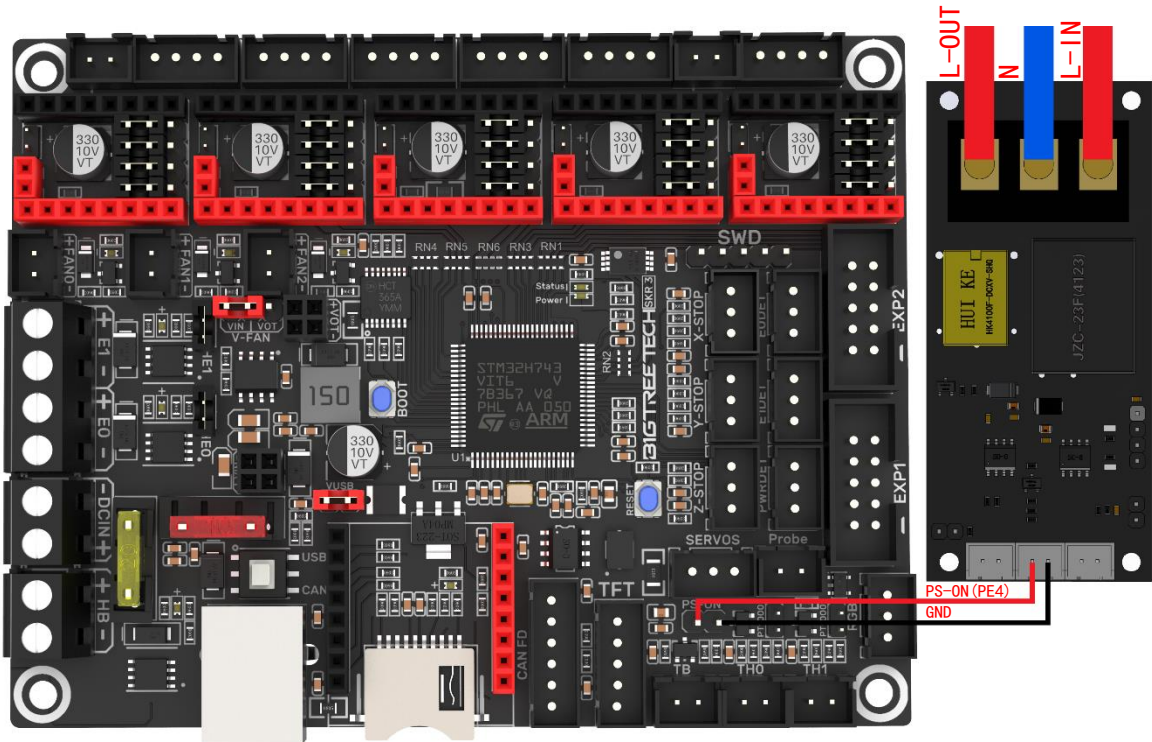
When using 100K NTC no jumpers need to be connected, the pull up resistance of TH0 & TH1 is 4.7K. When using PT1000 the jumpers indicated in the picture below needs to be connected, the pull up resistance of TH0 & TH1 is changed to 1K.



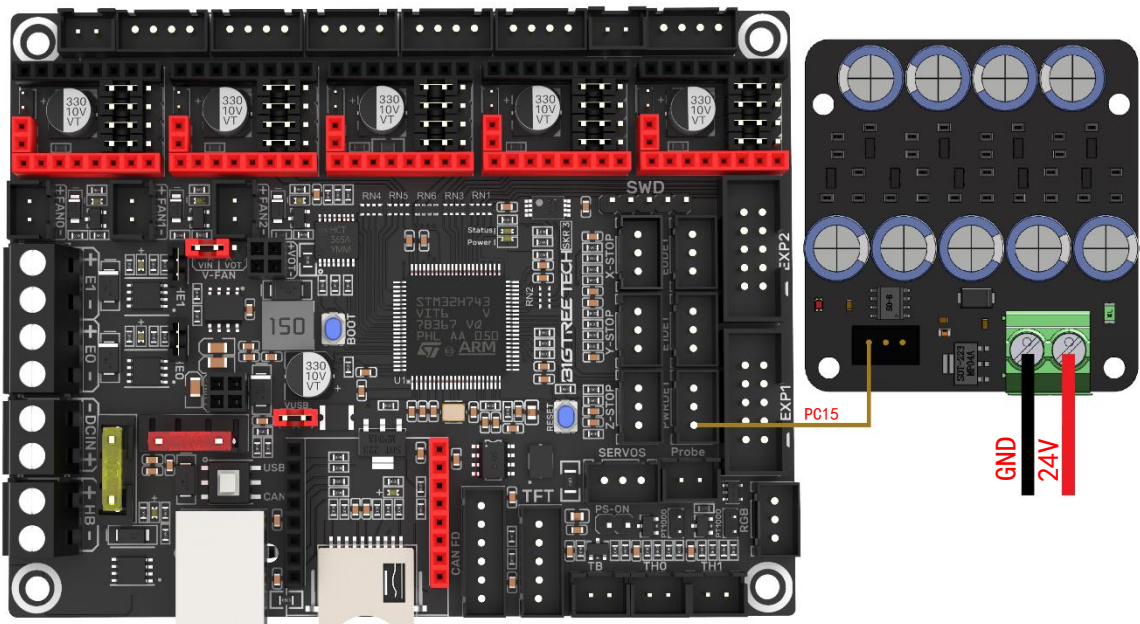
3.6 BLTouch wiring



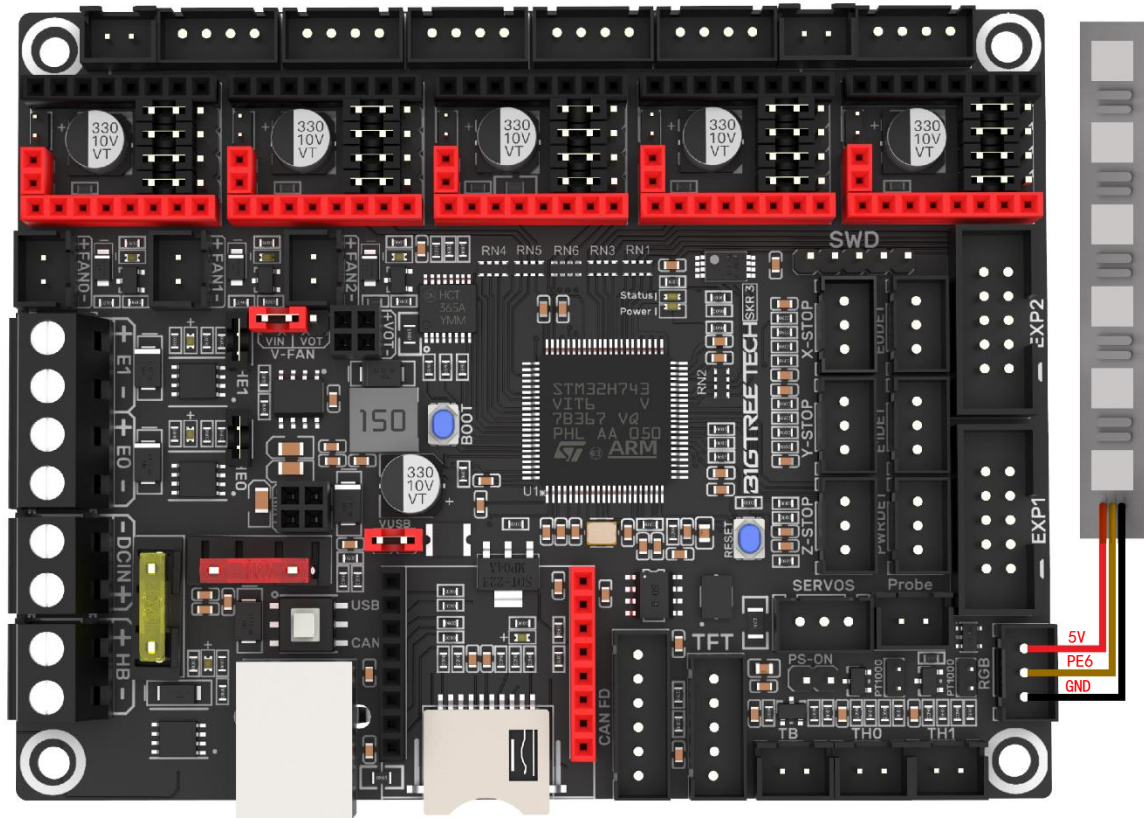
3.7 Auto power off (Relay V1.2) wiring



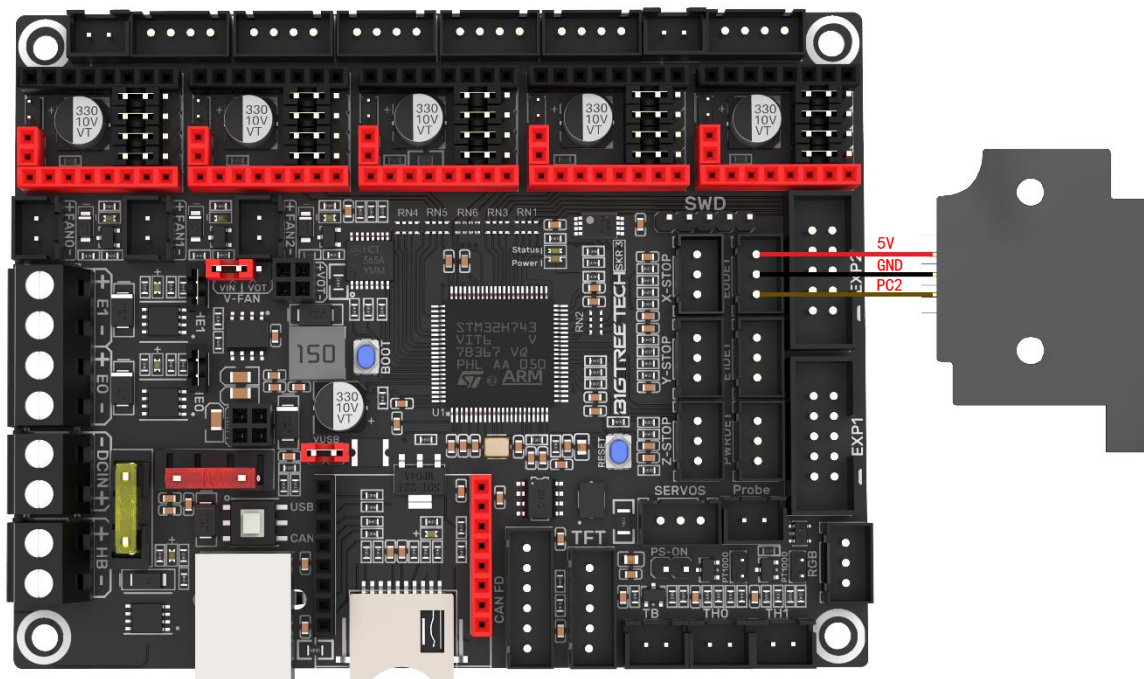
3.8 Power loss recovery (UPS 24V V1.0) wiring



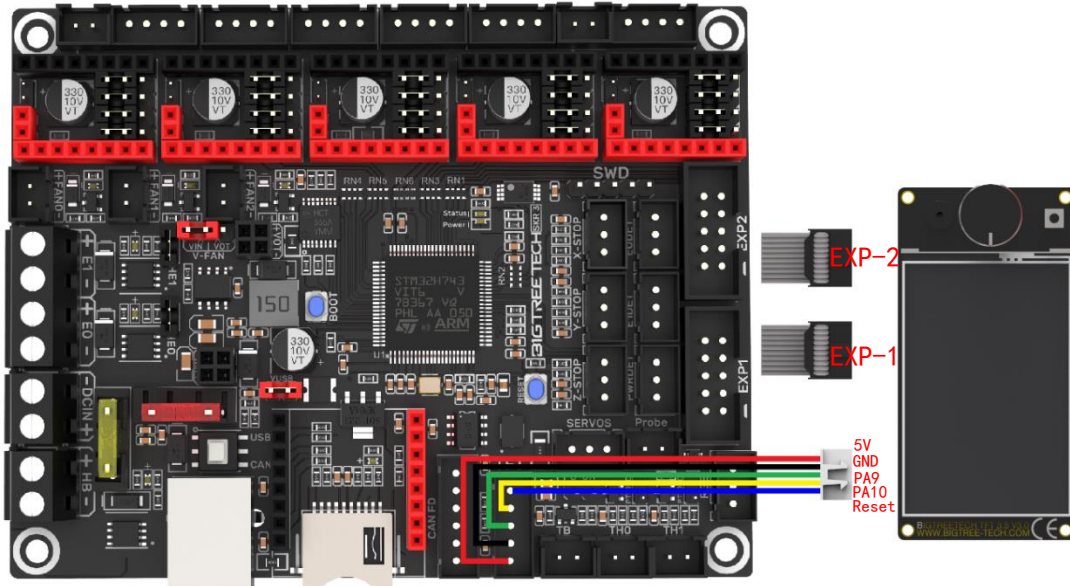
3.9 RGB wiring



3.10 Filament sensor wiring



3.11 Display wiring



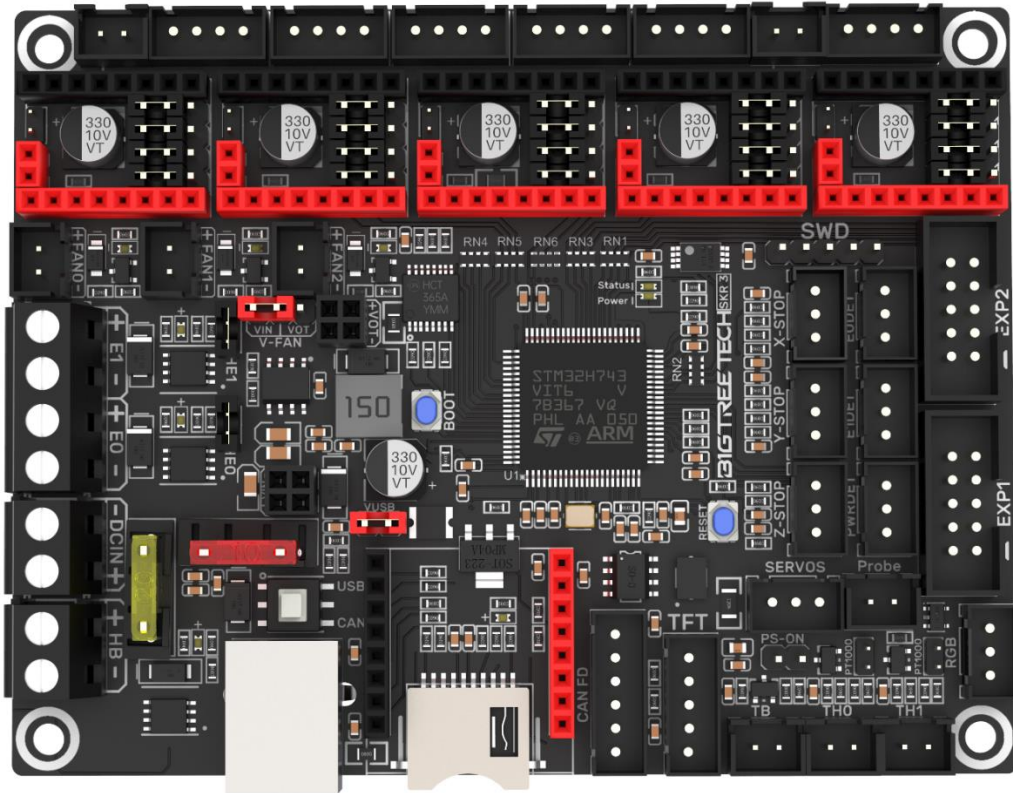
3.12 Heater cartridge IO

The IO of the SKR 3 heater cartridge is wired to the MOS by the jumper. You can remove the jumper and connect the IO to device directly if you need to use the laser or other device that need PWM. (Note: The IO passes through logic

☐ PB4

☐ PB3

conversion chip, the output high level is 5V, and cannot be used as an input)



4. Marlin

4.1 install compiling environment

<https://github.com/bigtreotech/Document/blob/master/How%20to%20install%20VScode%2BPlatformio.md>

https://marlinfw.org/docs/basics/install_platformio_vscode.html

Refer to the link above for tutorial on installing VSCode and PlatformIO plugin

4.2 Download Marlin firmware

1. Download the newest bugfix version of Marlin from official website
<https://github.com/MarlinFirmware/Marlin/tree/bugfix-2.0.x>
2. Download Pre-configured firmware from our GitHub page
<https://github.com/bigtreotech/SKR-3>

4.3 Configure firmware

4.3.1 Open Marlin project

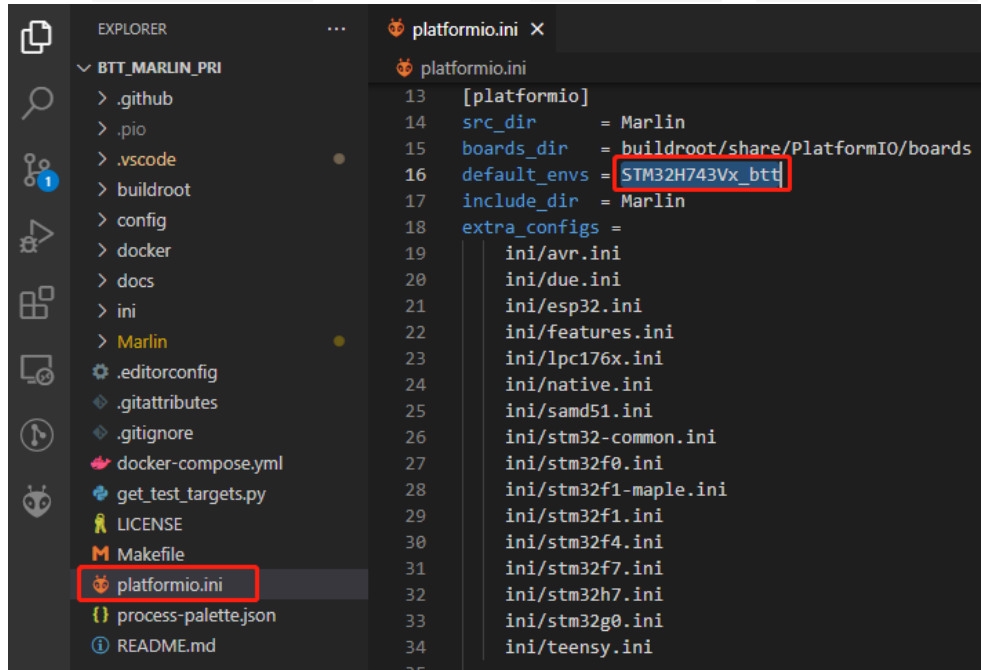
You can open Marlin in VS Code in one of several ways:

- Drag the downloaded Marlin Firmware folder onto the VScode application icon

- Use the **Open...** command in the VSCode **File** menu
- Open the PIO Home tab and click the “**Open Project**” button

4.3.2 Compiling environment

Open `platformio.ini` file and change `default_envs` to `STM32H743Vx_btt`



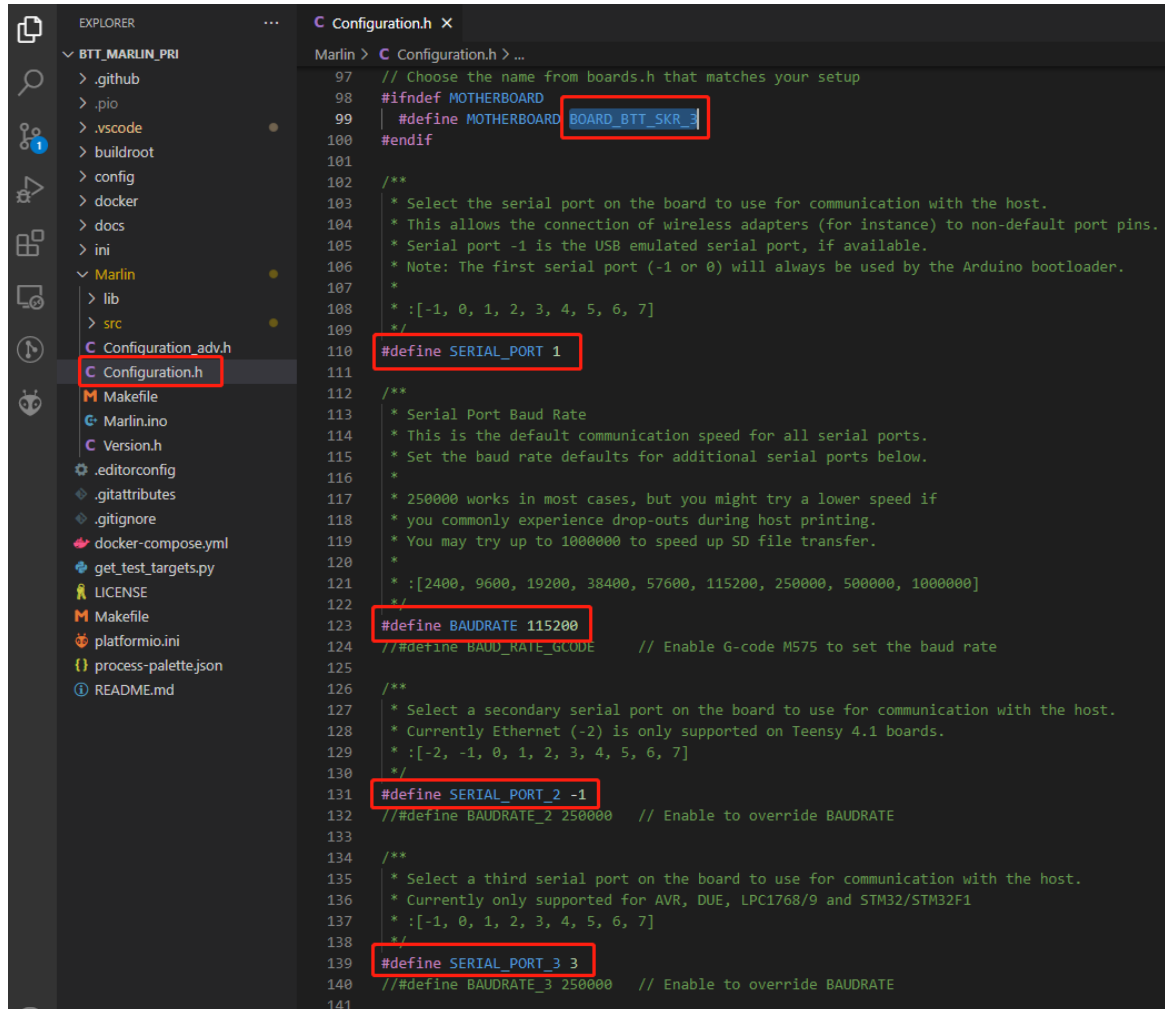
4.3.3 Configure motherboard and serial port

Set `MOTHERBOARD` to `BOARD_BTT_SKR_3`

Enable serial ports according to your needs

```
#define MOTHERBOARD BOARD_BTT_SKR_3
#define SERIAL_PORT 1          (enable TFT serial port)
#define BAUDRATE 115200       (set baudrate to the same as the communication device)
#define SERIAL_PORT_2 -1      (enable USB serial port)
```

#define SERIAL_PORT_3 3 (enable WIFI serial port)



The screenshot shows the Arduino IDE interface with the 'Configuration.h' file open. The left sidebar shows the project structure for 'BTT_MARLIN_PRI'. The main editor area displays the 'Configuration.h' file with the following lines highlighted in red boxes:

- Line 99: `#define MOTHERBOARD BOARD_BTT_SKR_3`
- Line 110: `#define SERIAL_PORT 1`
- Line 123: `#define BAUDRATE 115200`
- Line 131: `#define SERIAL_PORT_2 -1`
- Line 139: `#define SERIAL_PORT_3 3`

The code in the editor includes comments for selecting the serial port and baud rate, and defines for the MOTHERBOARD, SERIAL_PORT, BAUDRATE, and SERIAL_PORT_2.

4.3.4 Configure stepper driver

```

851  /**
852   * Stepper Drivers
853   *
854   * These settings allow Marlin to tune stepper driver timing and enable advanced options for
855   * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
856   *
857   * A4988 is assumed for unspecified drivers.
858   *
859   * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STANDALONE for TMC2226 drivers.
860   *
861   * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
862   *          TB6560, TB6600, TMC2100,
863   *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
864   *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
865   *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
866   *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
867   * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100',
868   */
869  #define X_DRIVER_TYPE  TMC2130
870  #define Y_DRIVER_TYPE  TMC2130
871  #define Z_DRIVER_TYPE  TMC2130
872  //#define X2_DRIVER_TYPE A4988
873  //#define Y2_DRIVER_TYPE A4988
874  //#define Z2_DRIVER_TYPE A4988
875  //#define Z3_DRIVER_TYPE A4988
876  //#define Z4_DRIVER_TYPE A4988
877  //#define I_DRIVER_TYPE A4988
878  //#define J_DRIVER_TYPE A4988
879  //#define K_DRIVER_TYPE A4988
880  #define E0_DRIVER_TYPE TMC2130
881  #define E1_DRIVER_TYPE TMC2130

```

When using SPI mode, `TMC_USE_SW_SPI` needs to be Uncommented in `Configuration_adv.h`

`#define TMC_USE_SW_SPI`

```

2980  /**
2981   * Software option for SPI driven drivers (TMC2130, TMC2160, TMC2660, TMC5130 and TMC5160).
2982   * The default SW SPI pins are defined the respective pins files,
2983   * but you can override or define them here.
2984   */
2985  #define TMC_USE_SW_SPI
2986  //#define TMC_SW_MOSI -1
2987  //#define TMC_SW_MISO -1
2988  //#define TMC_SW_SCK -1

```

4. 3. 5 Sensorless homing

```

3047 /**
3048  * Use StallGuard to home / probe X, Y, Z.
3049  *
3050  * TMC2130, TMC2160, TMC2209, TMC2660, TMC5130, and TMC5160 only
3051  * Connect the stepper driver's DIAG1 pin to the X/Y endstop pin.
3052  * X, Y, and Z homing will always be done in spreadCycle mode.
3053  *
3054  * X/Y/Z_STALL_SENSITIVITY is the default stall threshold.
3055  * Use M914 X Y Z to set the stall threshold at runtime:
3056  *
3057  * Sensitivity  TMC2209  Others
3058  * HIGHEST      255      -64   (Too sensitive => False positive)
3059  * LOWEST        0        63   (Too insensitive => No trigger)
3060  *
3061  * It is recommended to set HOMING_BUMP_MM to { 0, 0, 0 }.
3062  *
3063  * SPI_ENDSTOPS *** Beta feature! *** TMC2130/TMC5160 Only ***
3064  * Poll the driver through SPI to determine load when homing.
3065  * Removes the need for a wire from DIAG1 to an endstop pin.
3066  *
3067  * IMPROVE_HOMING_RELIABILITY tunes acceleration and jerk when
3068  * homing and adds a guard period for endstop triggering.
3069  *
3070  * Comment *_STALL_SENSITIVITY to disable sensorless homing for that axis.
3071  */
3072 #define SENSORLESS_HOMING // StallGuard capable drivers only
3073
3074 #if EITHER(SENSORLESS_HOMING, SENSORLESS_PROBING)
3075   // TMC2209: 0...255. TMC2130: -64...63
3076   #define X_STALL_SENSITIVITY 8
3077   #define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
3078   #define Y_STALL_SENSITIVITY 8
3079   #define Y2_STALL_SENSITIVITY Y_STALL_SENSITIVITY
3080   //#define Z_STALL_SENSITIVITY 8
3081   //#define Z2_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3082   //#define Z3_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3083   //#define Z4_STALL_SENSITIVITY Z_STALL_SENSITIVITY
3084   //#define I_STALL_SENSITIVITY 8
3085   //#define J_STALL_SENSITIVITY 8
3086   //#define K_STALL_SENSITIVITY 8
3087   //#define SPI_ENDSTOPS // TMC2130 only
3088   #define IMPROVE_HOMING_RELIABILITY
3089 #endif

```

`#define SENSORLESS_HOMING` // enable sensorless homing

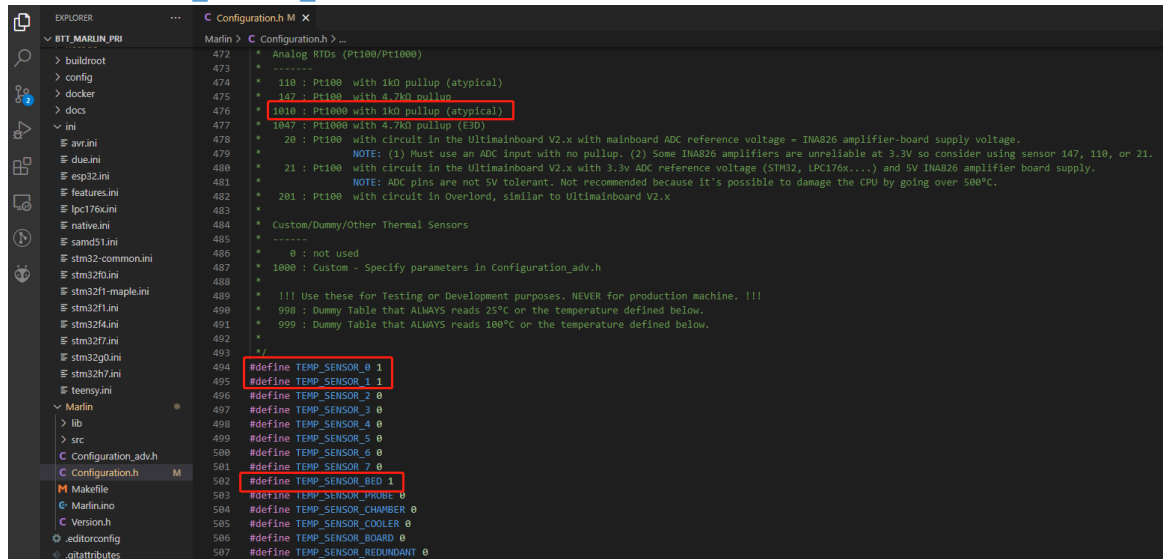
`#define xx_STALL_SENSITIVITY 8` // sensitivity setting ,TMC2209 range from 0 to 255, higher number results in more sensitive trigger threshold, sensitivity too high will cause endpoint to trigger before gantry actually move to the end, lower number results in less sensitive trigger threshold,, too low of sensitivity will cause endpoint to not trigger and gantrying continue. Other drivers ranges from 63 to -64, lower numbers results in more sensitive trigger threshold

`#define IMPROVE_HOMING_RELIABILITY` // can be used to set independent motor current for homing moves(`xx_CURRENT_HOME`) to improve homing reliability.

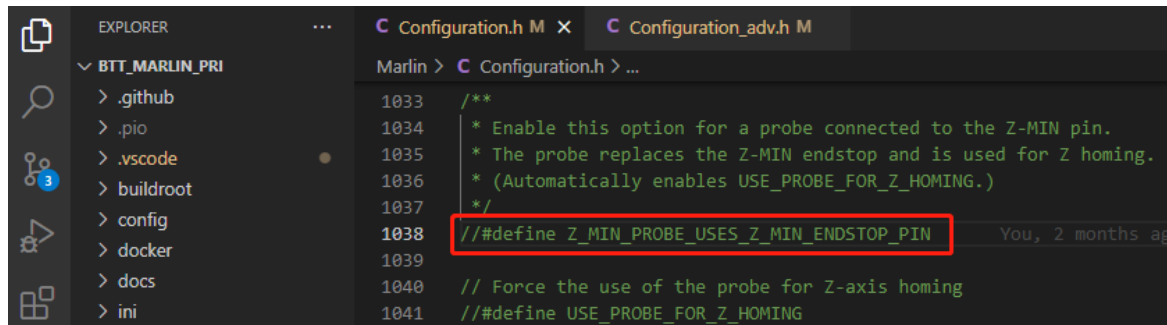
4. 3. 6 100K NTC or PT1000

When using 100K NTC, pullup resistance is 4.7K, when using Pt1000, pullup resistance is 1K, set sensor type to 1 for 100K NTC +4.7K pullup resistance , 1010 for PT1000 + 1K pullup resistance.

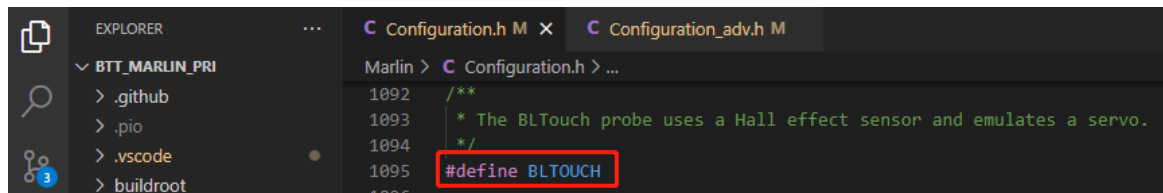
```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 1
#define TEMP_SENSOR_BED 1
```



4. 3. 7 BLTouch



```
// #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN //
```



```
#define BLTOUCH // Enable bltouch
```



```

1182 * Some examples:
1183 * #define NOZZLE_TO_PROBE_OFFSET { 10, 10, -1 } // Example "1"
1184 * #define NOZZLE_TO_PROBE_OFFSET {-10, 5, -1 } // Example "2"
1185 * #define NOZZLE_TO_PROBE_OFFSET { 5, -5, -1 } // Example "3"
1186 * #define NOZZLE_TO_PROBE_OFFSET {-15,-10, -1 } // Example "4"
1187 *
1188 * +-- BACK ---+
1189 * |   [+   |
1190 * L |       | R <-- Example "1" (right+, back+)
1191 * E | 2     | I <-- Example "2" ( left-, back+)
1192 * F |[-] N [+]| G <-- Nozzle
1193 * T |       | H <-- Example "3" (right+, front-)
1194 * | 4       | T <-- Example "4" ( left-, front-)
1195 * | [-]     |
1196 * 0-- FRONT --+
1197 */
1198 #define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 }
1199
1200 // Most probes should stay away from the edges of the bed, but
1201 // with NOZZLE_AS_PROBE this can be negative for a wider probing area.
1202 #define PROBING_MARGIN 10
1203
1204 // X and Y axis travel speed (mm/min) between probes
1205 #define XY_PROBE_FEEDRATE (133*60)
1206
1207 // Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_PROBING == 2)
1208 #define Z_PROBE_FEEDRATE_FAST (4*60)
1209
1210 // Feedrate (mm/min) for the "accurate" probe of each point
1211 #define Z_PROBE_FEEDRATE_SLOW (Z_PROBE_FEEDRATE_FAST / 2)
1212

```

`#define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 }` // set BLtouch probe offset

`#define PROBING_MARGIN 10` // set distance between probe area and print area perimeter

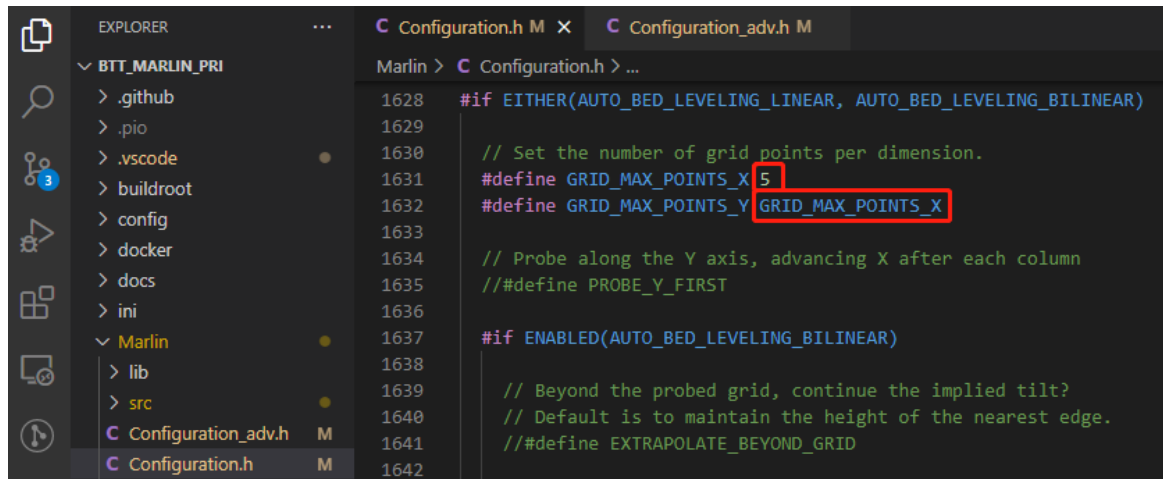
```

1562 // #define AUTO_BED_LEVELING_3POINT
1563 // #define AUTO_BED_LEVELING_LINEAR
1564 #define AUTO_BED_LEVELING_BILINEAR
1565 // #define AUTO_BED_LEVELING_UBL
1566 // #define MESH_BED_LEVELING
1567
1568 /**
1569  * Normally G28 leaves leveling disabled on completion. Enable one of
1570  * these options to restore the prior leveling state or to always enable
1571  * leveling immediately after G28.
1572  */
1573 // #define RESTORE_LEVELING_AFTER_G28
1574 #define ENABLE_LEVELING_AFTER_G28
1575
1576 /**

```

`#define AUTO_BED_LEVELING_BILINEAR` // set probe pattern

`#define RESTORE_LEVELING_AFTER_G28` // apply leveling after G28 homing command



```

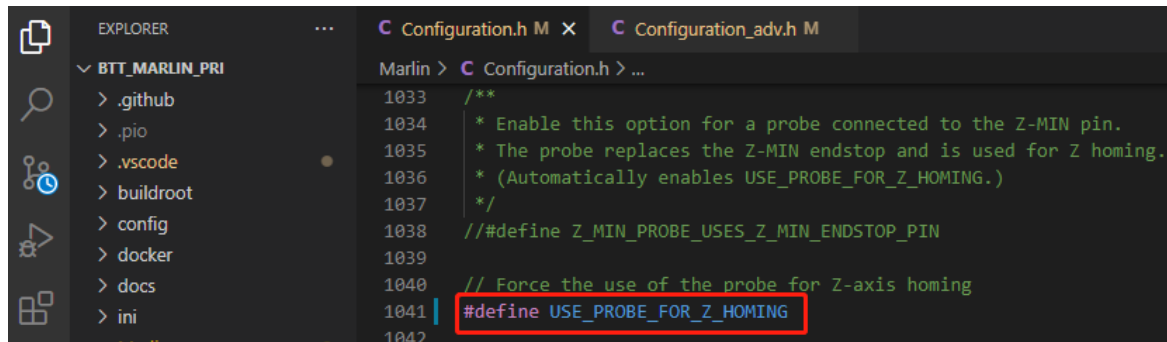
1628 #if EITHER(AUTO_BED_LEVELING_LINEAR, AUTO_BED_LEVELING_BILINEAR)
1629
1630 // Set the number of grid points per dimension.
1631 #define GRID_MAX_POINTS_X 5
1632 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
1633
1634 // Probe along the Y axis, advancing X after each column
1635 //#define PROBE_Y_FIRST
1636
1637 #if ENABLED(AUTO_BED_LEVELING_BILINEAR)
1638
1639 // Beyond the probed grid, continue the implied tilt?
1640 // Default is to maintain the height of the nearest edge.
1641 //#define EXTRAPOLATE_BEYOND_GRID
1642

```

#define GRID_MAX_POINTS_X 5 // set number of probe points for x axis, usually 5 point is sufficient

#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X // set number of probe points for Y axis to the same as X axis

If bltouch also functions as your Z homing sensor, no wiring change is needed,

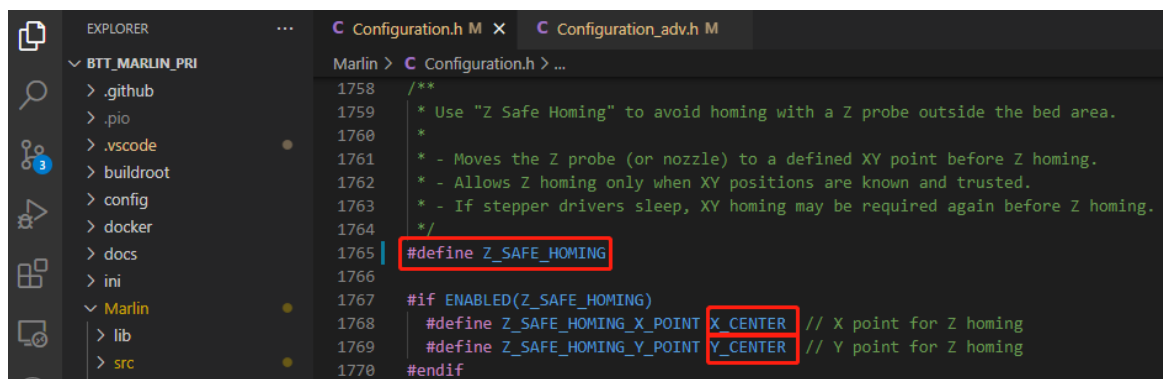


```

1033 /**
1034  * Enable this option for a probe connected to the Z-MIN pin.
1035  * The probe replaces the Z-MIN endstop and is used for Z homing.
1036  * (Automatically enables USE_PROBE_FOR_Z_HOMING.)
1037  */
1038 //#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
1039
1040 // Force the use of the probe for Z-axis homing
1041 #define USE_PROBE_FOR_Z_HOMING
1042

```

#define USE_PROBE_FOR_Z_HOMING // use Z Probe(BLtouch) for Z homing



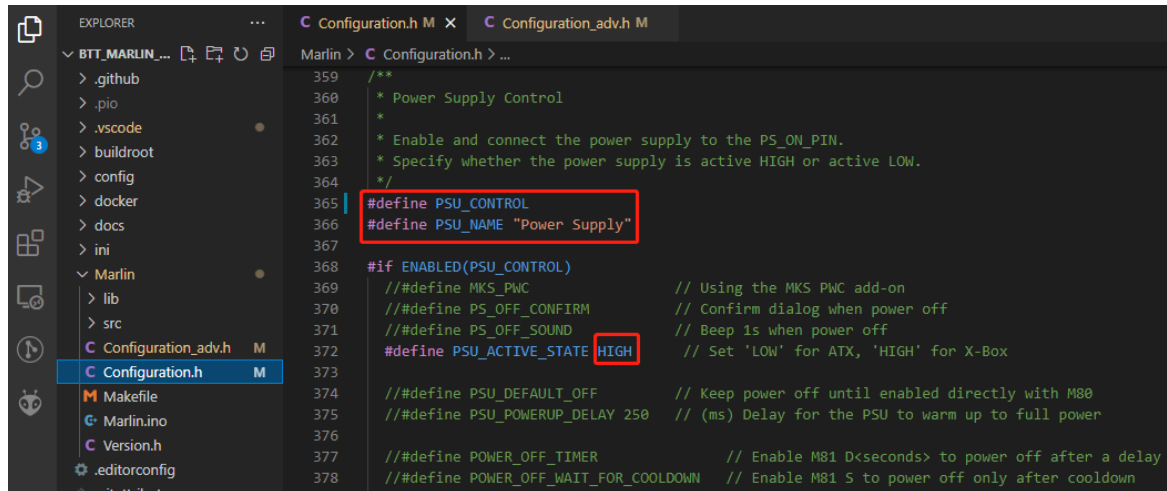
```

1758 /**
1759  * Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
1760  *
1761  * - Moves the Z probe (or nozzle) to a defined XY point before Z homing.
1762  * - Allows Z homing only when XY positions are known and trusted.
1763  * - If stepper drivers sleep, XY homing may be required again before Z homing.
1764  */
1765 #define Z_SAFE_HOMING
1766
1767 #if ENABLED(Z_SAFE_HOMING)
1768   #define Z_SAFE_HOMING_X_POINT X_CENTER // X point for Z homing
1769   #define Z_SAFE_HOMING_Y_POINT Y_CENTER // Y point for Z homing
1770 #endif

```

#define Z_SAFE_HOMING // home Z at the center of print bed to prevent probing outside of the print bed.

4.3.8 Auto power off(Relay V1.2)



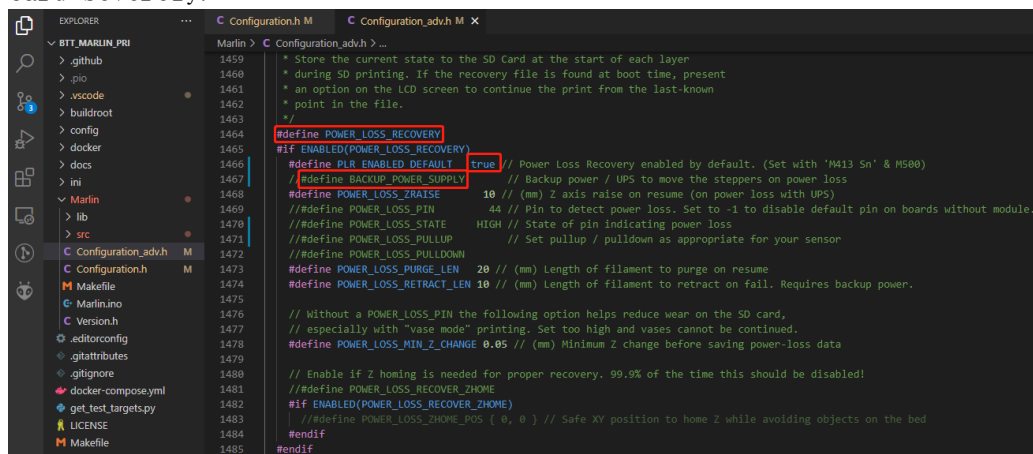
`#define PSU_CONTROL` // enable PSU control to turn on and off using M80 and M81

`#define PSU_ACTIVE_STATE HIGH` // set turn on level, Relay V1.2 is turned on with high level and turned off with low level ,so this setting needs to be HIGH.

4.3.9 Power loss recovery

There are two methods for power lost recovery

1. No extra module needed, the motherboard will write current print status to the SD card after every layer is printed, which shortens the life of the SD card severely.



`#define POWER_LOSS_RECOVERY` // enable power loss recovery
`PLR_ENABLED_DEFAULT true` // true default to power loss recovery enabled

2. external UPS 24V V1.0 module, when power is cut, the module will provide power to the board and signal the board to save current print status to SD card. This method has virtually no effect on the life of the SD card.

```

1459  * Store the current state to the SD Card at the start of each layer
1460  * during SD printing. If the recovery file is found at boot time, present
1461  * an option on the LCD screen to continue the print from the last-known
1462  * point in the file.
1463  */
1464  #define POWER_LOSS_RECOVERY
1465  #if ENABLED(POWER_LOSS_RECOVERY)
1466    #define PLR_ENABLED_DEFAULT true // Power Loss Recovery enabled by default. (Set with 'M413 Sn' & M500)
1467    #define BACKUP_POWER_SUPPLY // Backup power / UPS to move the steppers on power loss
1468    #define POWER_LOSS_ZRAISE 10 // (mm) Z axis raise on resume (on power loss with UPS)
1469    #define POWER_LOSS_PIN -1 // Pin to detect power loss. Set to -1 to disable default pin on boards without module.
1470    #define POWER_LOSS_STATE HIGH // State of pin indicating power loss
1471    #define POWER_LOSS_PULLUP // Set pullup / pulldown as appropriate for your sensor
1472    // #define POWER_LOSS_PULLDOWN
1473    #define POWER_LOSS_PURGE_LEN 20 // (mm) Length of filament to purge on resume
1474    #define POWER_LOSS_RETRACT_LEN 10 // (mm) Length of filament to retract on fail. Requires backup power.
1475
1476    // Without a POWER_LOSS_PIN the following option helps reduce wear on the SD card,
1477    // especially with "vase mode" printing. Set too high and vases cannot be continued.
1478    #define POWER_LOSS_MIN_Z_CHANGE 0.05 // (mm) Minimum Z change before saving power-loss data
1479
1480    // Enable if Z homing is needed for proper recovery. 99.9% of the time this should be disabled!
1481    // #define POWER_LOSS_RECOVER_ZHOME
1482    #if ENABLED(POWER_LOSS_RECOVER_ZHOME)
1483      // #define POWER_LOSS_ZHOME_POS { 0, 0 } // Safe XY position to home Z while avoiding objects on the bed
1484    #endif
1485  #endif

```

```

#define POWER_LOSS_RECOVERY // enable power loss recovery
#define PLR_ENABLED_DEFAULT true // true default to power loss recovery
enabled
#define POWER_LOSS_ZRAISE 10 // raise the print head by 10mm after
power loss to prevent the nozzle from touching the printed part.
#define POWER_LOSS_STATE HIGH // set signal level, UPS 24V V1.0 returns
low level when not triggered and HIGH level when power is cut, thus this
setting needs to be HIGH.

```

4. 3. 10 RGB

```

2926 // Support for Adafruit NeoPixel LED driver
2927 #define NEOPIXEL_LED
2928 #if ENABLED(NEOPIXEL_LED)
2929   #define NEOPIXEL_TYPE NEO_GRB // NEO_GRBW / NEO_GRB - four/three channel driver type (defined in Adafruit_NeoPixel.h)
2930   // #define NEOPIXEL_PIN 4 // LED driving pin
2931   // #define NEOPIXEL2_TYPE NEOPIXEL_TYPE
2932   // #define NEOPIXEL2_PIN 5
2933   #define NEOPIXEL_PIXELS 30 // Number of LEDs in the strip. (Longest strip when NEOPIXEL2_SEPARATE is disabled.)
2934   #define NEOPIXEL_IS_SEQUENTIAL // Sequential display for temperature change - LED by LED. Disable to change all LEDs at once.
2935   #define NEOPIXEL_BRIGHTNESS 255 // Initial brightness (0-255)
2936   #define NEOPIXEL_STARTUP_TEST // Cycle through colors at startup
2937
2938 // Support for second Adafruit NeoPixel LED driver controlled with M150 S1 ...
2939 #define NEOPIXEL2_SEPARATE
2940 #if ENABLED(NEOPIXEL2_SEPARATE)
2941   #define NEOPIXEL2_PIXELS 15 // Number of LEDs in the second strip
2942   #define NEOPIXEL2_BRIGHTNESS 127 // Initial brightness (0-255)
2943   #define NEOPIXEL2_STARTUP_TEST // Cycle through colors at startup
2944 #else
2945   // #define NEOPIXEL2_INSERIES // Default behavior is NeoPixel 2 in parallel
2946 #endif
2947
2948 // Use some of the NeoPixel LEDs for static (background) lighting
2949 // #define NEOPIXEL_BKGD_INDEX_FIRST 0 // Index of the first background LED
2950 // #define NEOPIXEL_BKGD_INDEX_LAST 5 // Index of the last background LED
2951 // #define NEOPIXEL_BKGD_COLOR { 255, 255, 255, 0 } // R, G, B, W
2952 // #define NEOPIXEL_BKGD_ALWAYS_ON // Keep the backlight on when other NeoPixels are off
2953 #endif

```

```

#define NEOPIXEL_LED // enable Neopixel
#define NEOPIXEL_TYPE NEO_GRB // set Neopixel type
// #define NEOPIXEL_PIN 4 // disable PIN setting, use the correct signal pin
in the pin file of the motherboard
#define NEOPIXEL_PIXELS 30 // number of leds
#define NEOPIXEL_STARTUP_TEST // the light will show red green and blue
sequentially to self-test.

```

If you are using displays like LCD2004、12864、mini12864 etc, you can also control RGB from your display directly.

```

1326  /**
1327   * LED Control Menu
1328   * Add LED Control to the LCD menu
1329   */
1330  #define LED_CONTROL_MENU
1331  #if ENABLED(LED_CONTROL_MENU)
1332    #define LED_COLOR_PRESETS           // Enable the Preset Color menu option
1333    //#define NEO2_COLOR_PRESETS        // Enable a second NeoPixel Preset Color menu option
1334    #if ENABLED(LED_COLOR_PRESETS)
1335      #define LED_USER_PRESET_RED       255 // User defined RED value
1336      #define LED_USER_PRESET_GREEN     128 // User defined GREEN value
1337      #define LED_USER_PRESET_BLUE      0  // User defined BLUE value
1338      #define LED_USER_PRESET_WHITE     255 // User defined WHITE value
1339      #define LED_USER_PRESET_BRIGHTNESS 255 // User defined intensity
1340      //#define LED_USER_PRESET_STARTUP // Have the printer display the user preset color on startup
1341    #endif
1342    #if ENABLED(NEO2_COLOR_PRESETS)
1343      #define NEO2_USER_PRESET_RED       255 // User defined RED value
1344      #define NEO2_USER_PRESET_GREEN     128 // User defined GREEN value
1345      #define NEO2_USER_PRESET_BLUE      0  // User defined BLUE value
1346      #define NEO2_USER_PRESET_WHITE     255 // User defined WHITE value
1347      #define NEO2_USER_PRESET_BRIGHTNESS 255 // User defined intensity
1348      //#define NEO2_USER_PRESET_STARTUP // Have the printer display the user preset color on startup for the second strip
1349    #endif
1350  #endif

```

#define LED_CONTROL_MENU // And led control to your menu

4.3.11 Filament sensor

Standard filament run out sensors are usually comprised of a microswitch which signals the mainboard of filament status with High or Low level signal.

```

1462  #define FILAMENT_RUNOUT_SENSOR
1463  #if ENABLED(FILAMENT_RUNOUT_SENSOR)
1464    #define FIL_RUNOUT_ENABLED_DEFAULT true // Enable the sensor on startup. Override with M412 followed by M500.
1465    #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to one per extruder. Define a FIL_RUNOUT#_PIN for each.
1466    #define FIL_RUNOUT_STATE LOW // Pin state indicating that filament is NOT present.
1467    #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins.
1468    //#define FIL_RUNOUT_PULLDOWN // Use internal pulldown for filament runout pins.
1469    //#define WATCH_ALL_RUNOUT_SENSORS // Execute runout script on any triggering sensor, not only for the active extruder.
1470    // Execute runout script on any triggering sensor, not only for the active extruder.
1471    // This is automatically enabled for MIXING_EXTRUDERS.

```

#define FILAMENT_RUNOUT_SENSOR // enable filament run out sensor

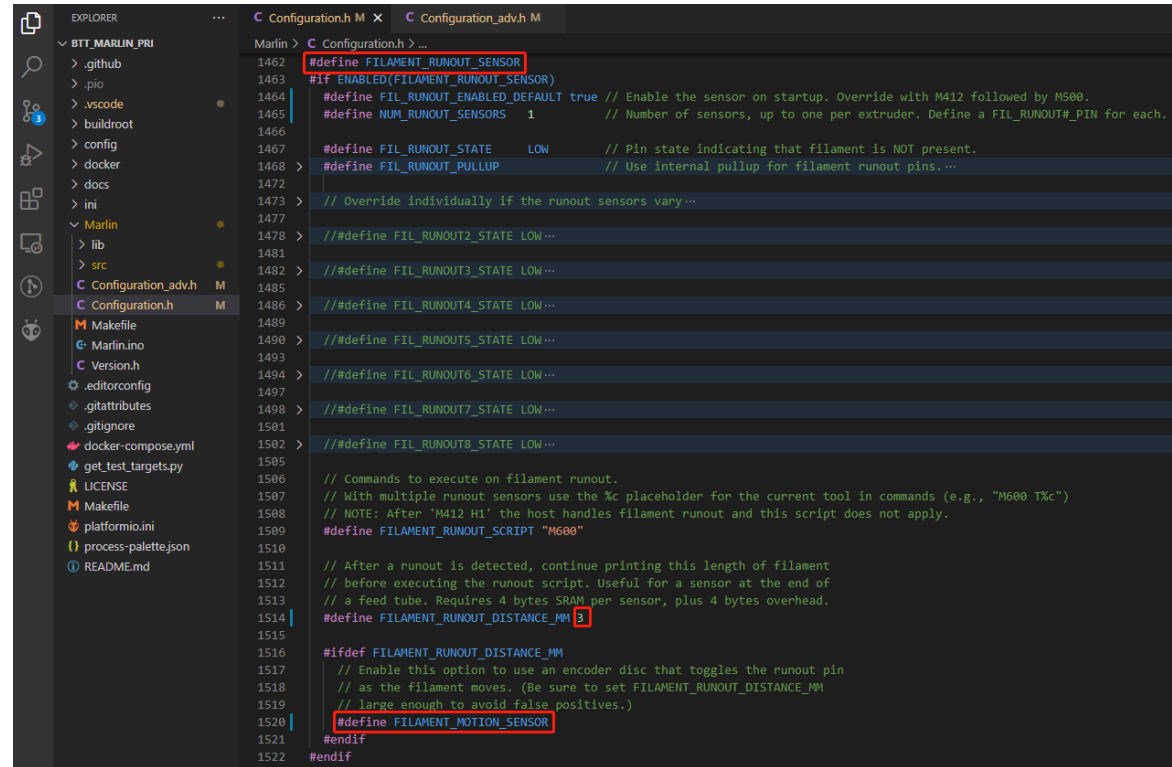
#define FIL_RUNOUT_ENABLED_DEFAULT true // true default to filament run out sensor enabled

#define NUM_RUNOUT_SENSORS 1 // number of filament run out sensor

#define FIL_RUNOUT_STATE LOW // voltage level of the filament runout sensor trigger signal.

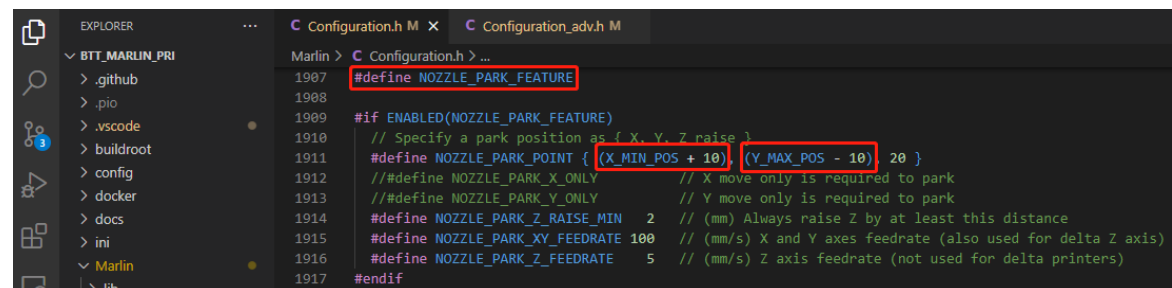
4.3.12 smart filament sensor (SFS V1.0 / V2.0)

The Smart filament sensor works by continuously sending signal to the mainboard to communicate filament status

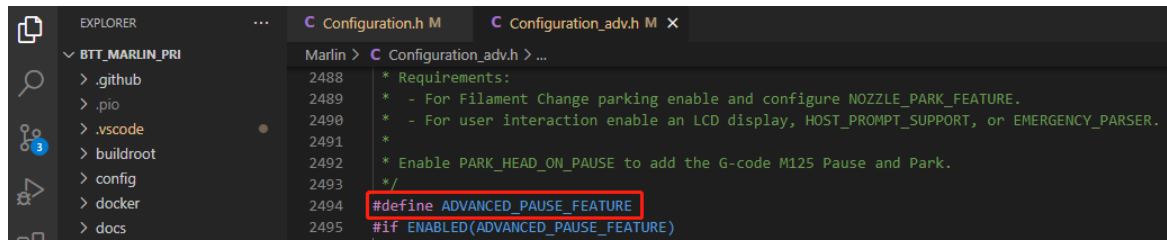


`#define FILAMENT_MOTION_SENSOR` // set encoder type
`#define FILAMENT_RUNOUT_DISTANCE_MM 3` // set sensitivity, SFS V1.0 nominal setting should be 7mm, which means if no signal of filament movement is detected after 7mm of filament travel command, filament error will be triggered, SFS V2.0 nominal setting should be 3mm

The settings below also need to be set to instruct the printer to park the nozzle after filament error is detected.



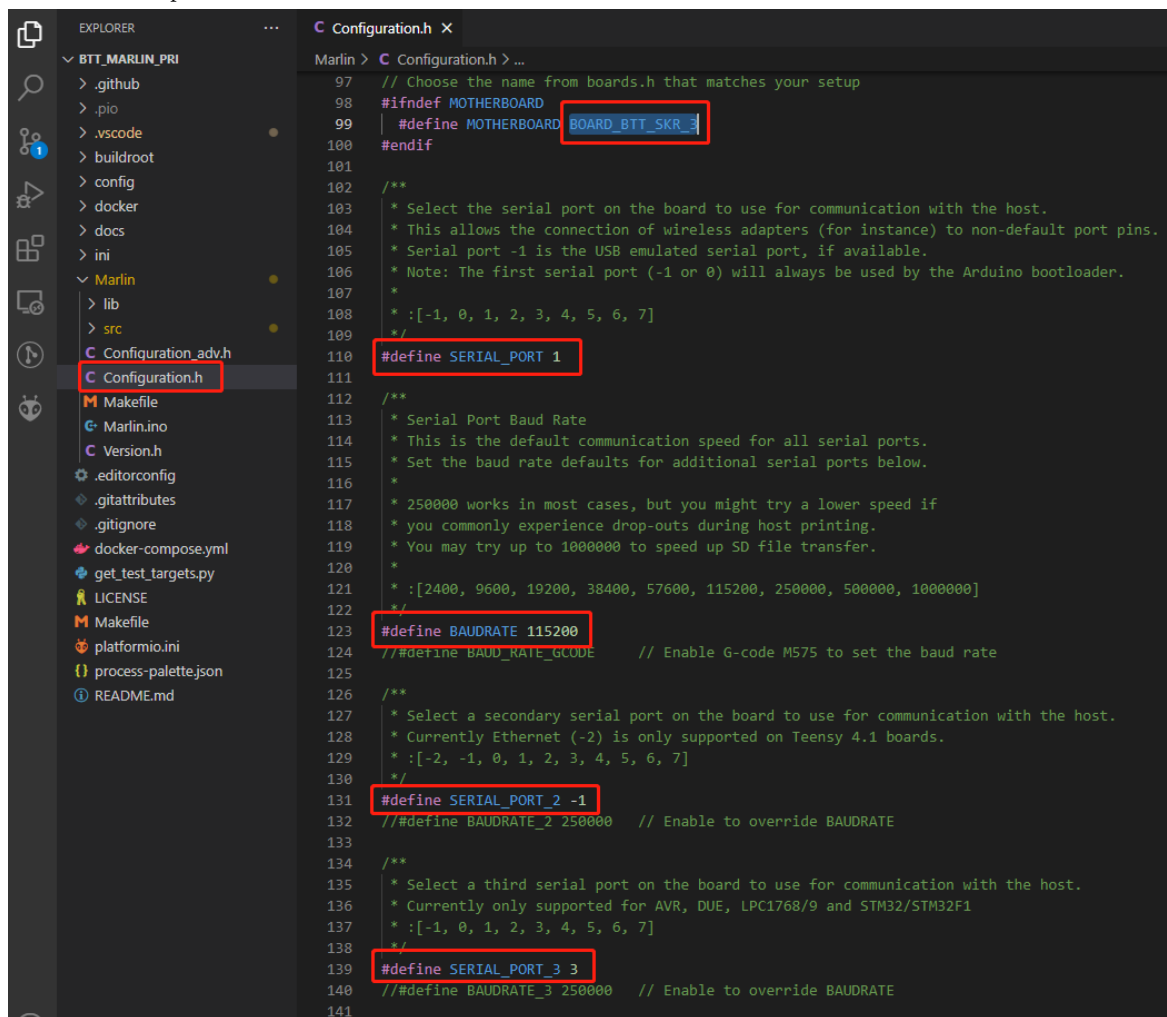
`#define NOZZLE_PARK_FEATURE` // park nozzle
`#define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }` // set the X,Y, and Z offset coordinate of the nozzle



`#define ADVANCED_PAUSE_FEATURE` // retraction setting of nozzle park movement and filament purge distance after print is resumed.

4. 3. 13 ESP3D

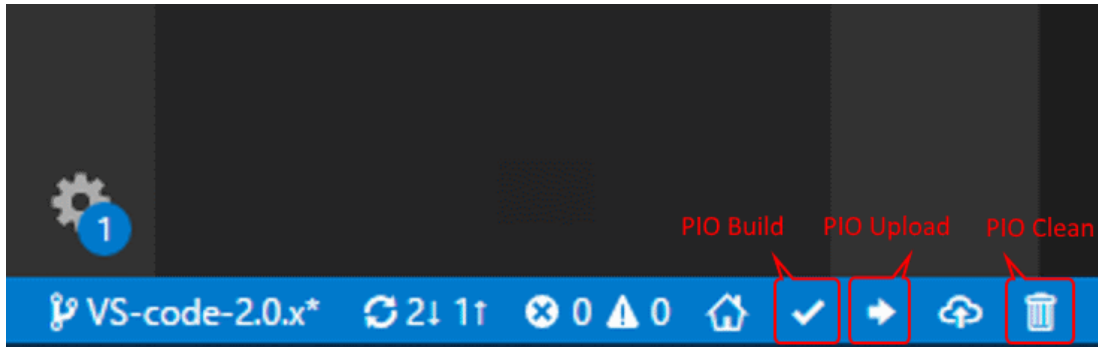
The serial port between ESP8266 and Marlin on the motherboard is UART3



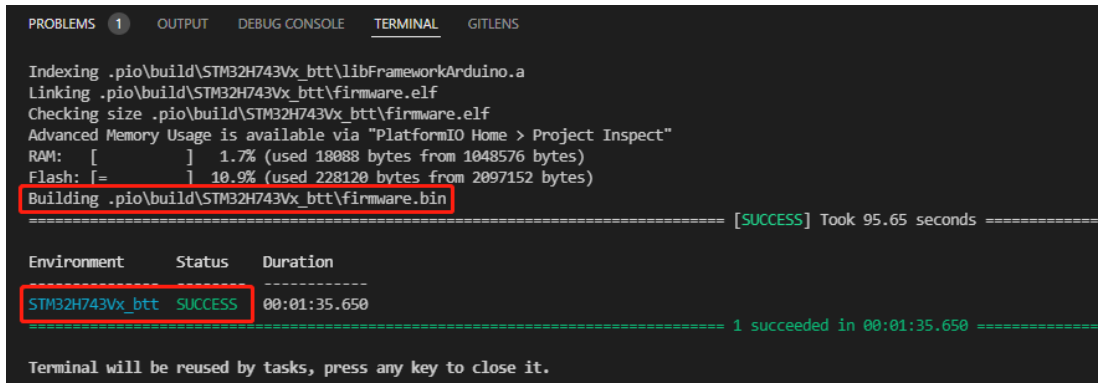
the newest ESP3D firmware can be found at <https://github.com/luc-github/ESP3D>, compile your own binary file and rename to “esp3d.bin”, copy to the root directory of the sd card, insert into the motherboard and press reset button. The bootloader will update the firmware to ESP8266 automatically. If updated successfully, the file will be renamed to ESP3D.CUR

4.4 Compile firmware

1. Click “√” to compile firmware



2. Copy the compiled “firmware.bin” to SD card and insert to motherboard to update firmware



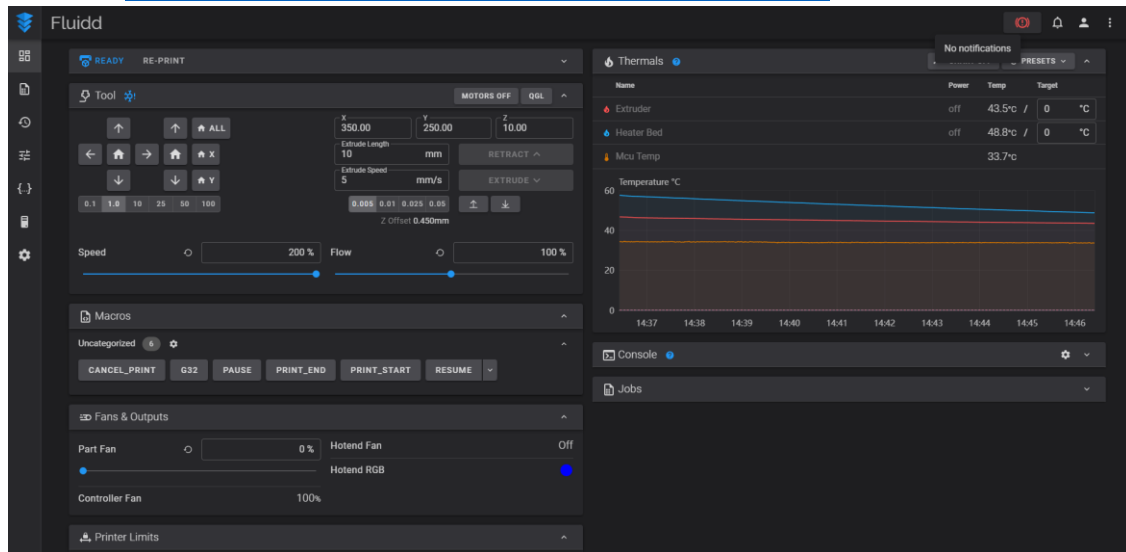
5. Klipper

5.1 Preparation

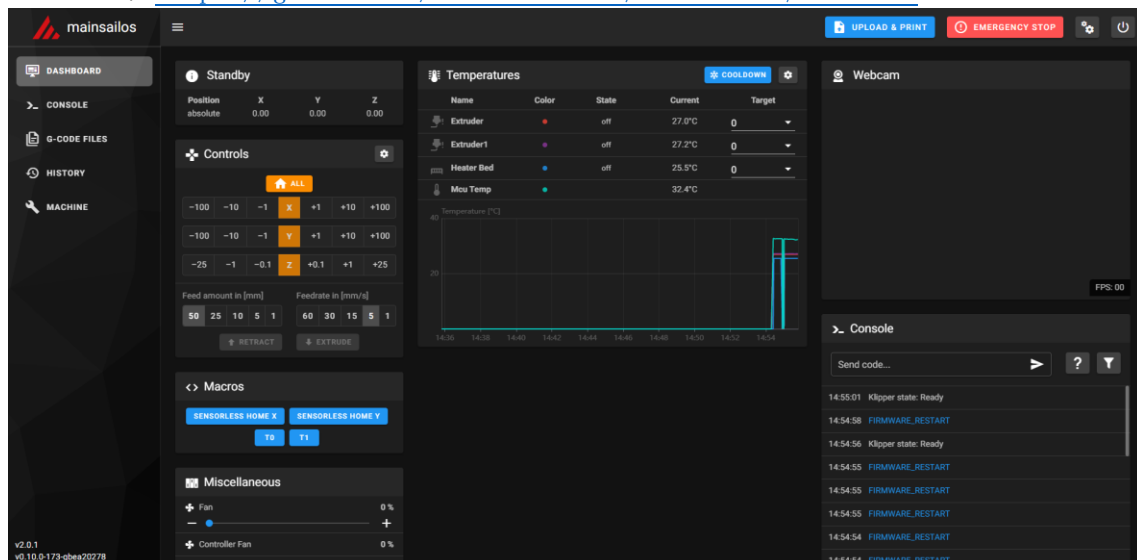
5.1.1 Download OS image

Download your preferred OS image with build in WebUI, popular choices are Fluidt, Mainsail etc.

Fluidt: <https://github.com/fluidt-core/FluidtPI/releases>



Mainsail: <https://github.com/mainsail-crew/MainsailOS/releases>



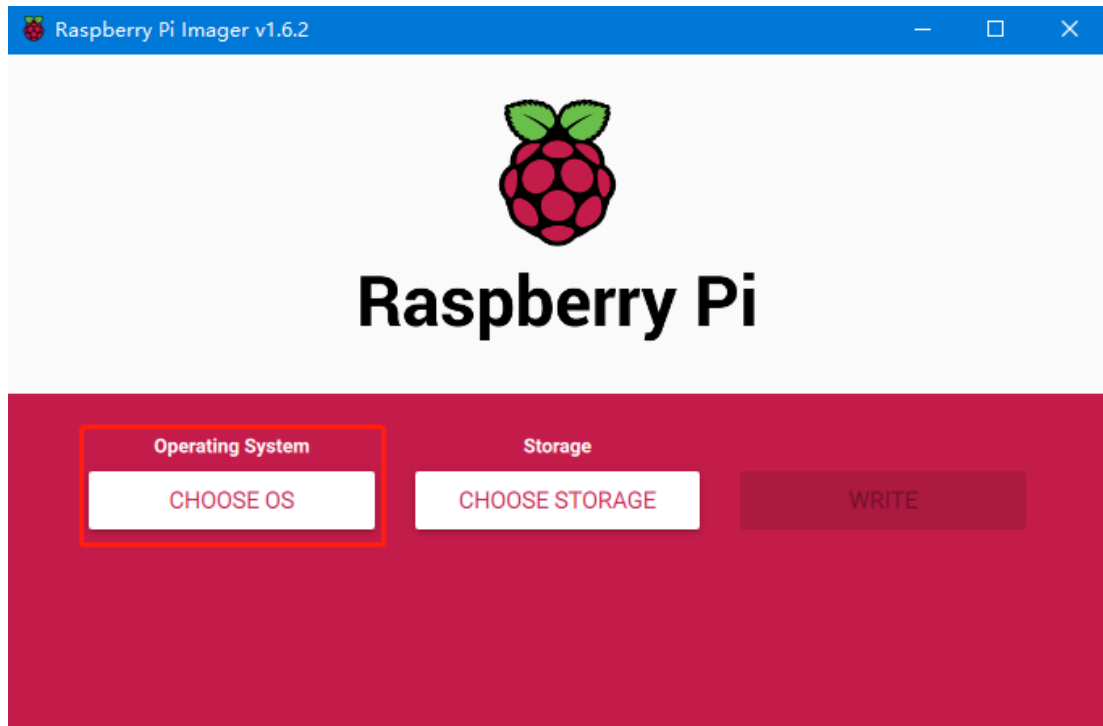
Or refer to [Klipper official installation guide](#) using Octoprint

5.1.2 Download and install Raspberry Pi Imager

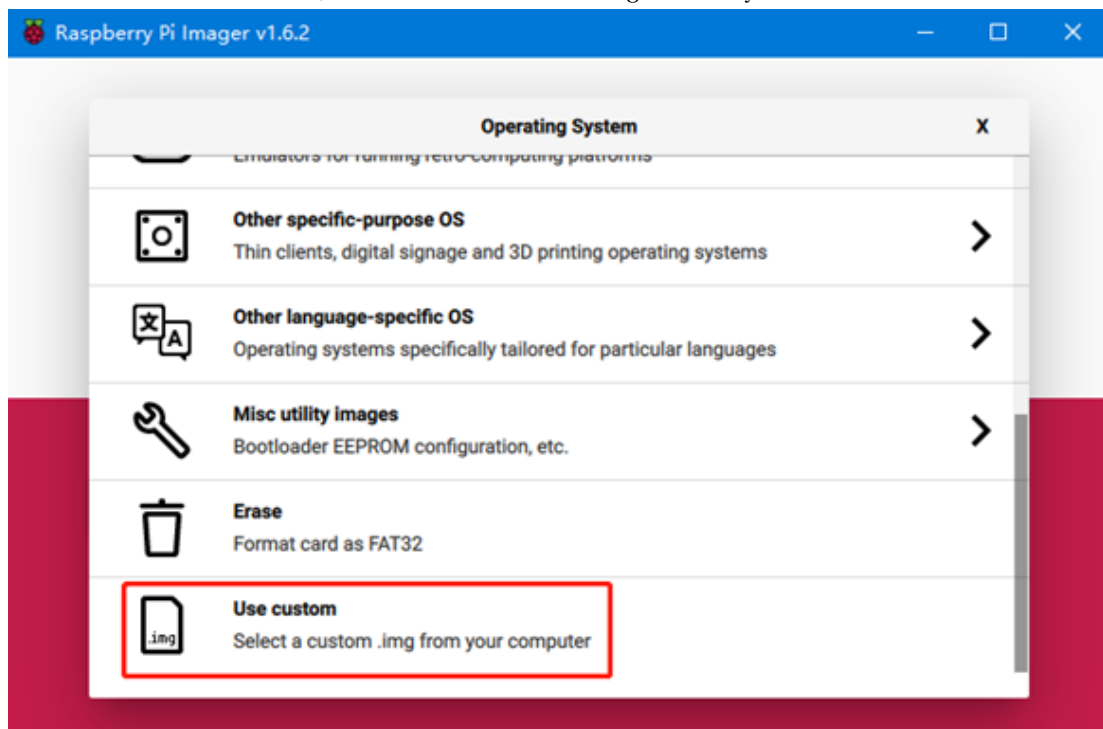
Install the official raspberry pi imager <https://www.raspberrypi.com/software/>

5.2 Write image

1. Insert Micro SD to your computer
2. Choose OS



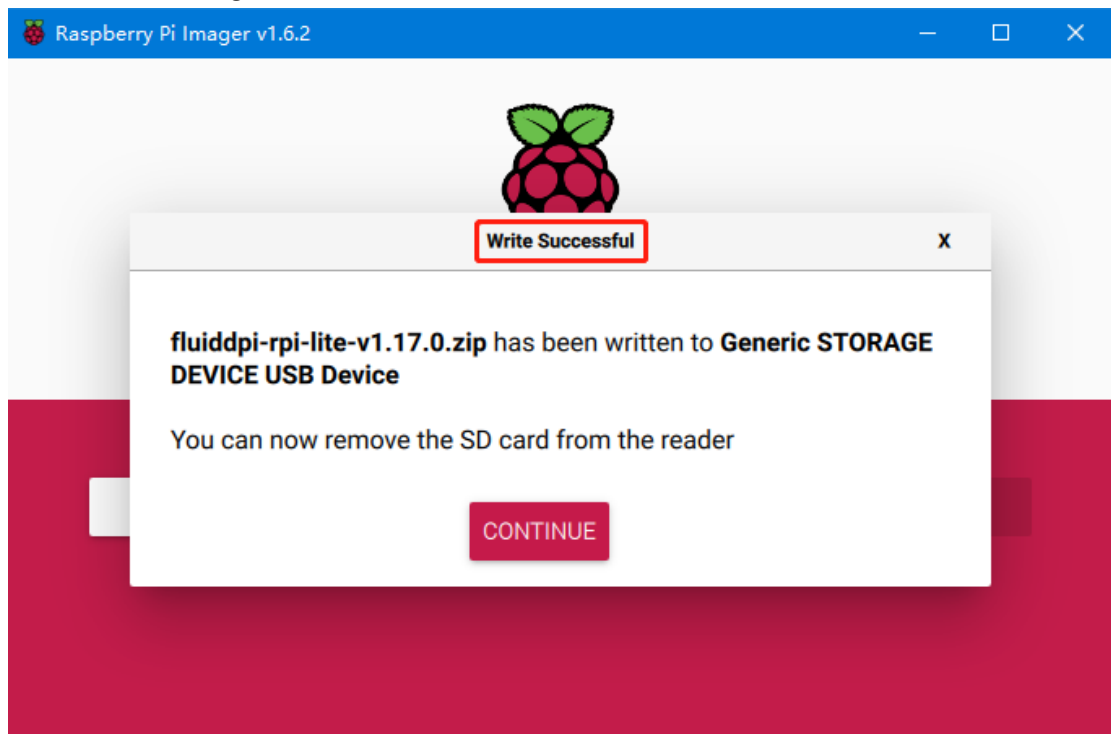
3. Select "Use custom", then select the image that you downloaded.



4. Select the SD card and click “WRITE” (WRITE the image will format the SD card. Be careful not to select the wrong storage device, otherwise the data will be formatted)



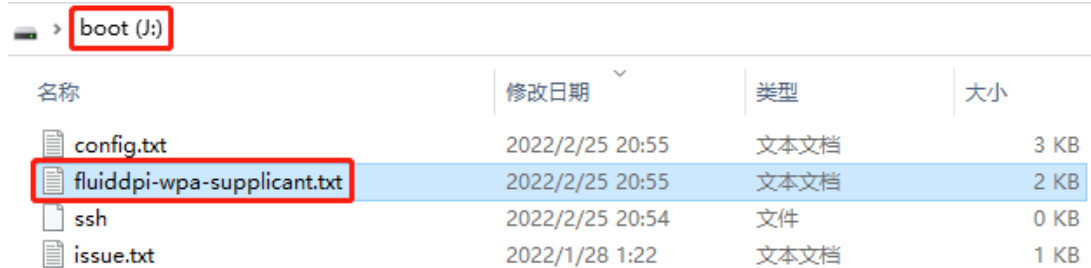
5. Wait for writing to finish



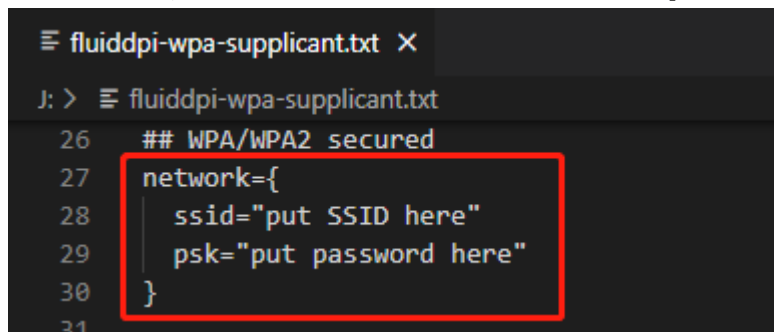
5.3 WIFI setting

note: skip this step if you are using ethernet port not using WIFI

1. Reinsert the SD card
2. Find “fluiddpi-wpa-supPLICant.txt” or “mainsail-wpa-supPLICant.txt” in the SD card root directory, open it with VSCode (do not open it with windows notepad)

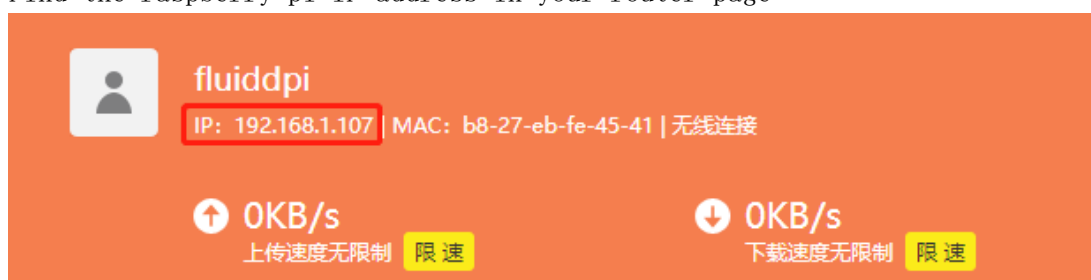


3. delete ‘#’, insert the correct wifi SSID and password then save the file,

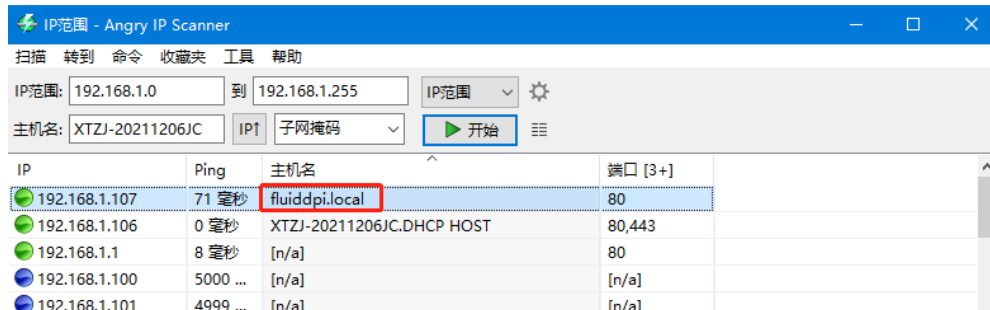


5.4 ssh connect to raspberry pi

1. Install the ssh application Mobaxterm: <https://mobaxterm.mobatek.net/download-home-edition.html>
2. Insert SD card to raspberry pi, wait for system to load after power one, aprox. 1-2min
3. The raspberry pi will automatically be assigned a IP address after successfully connected to the network
4. Find the raspberry pi IP address in your router page

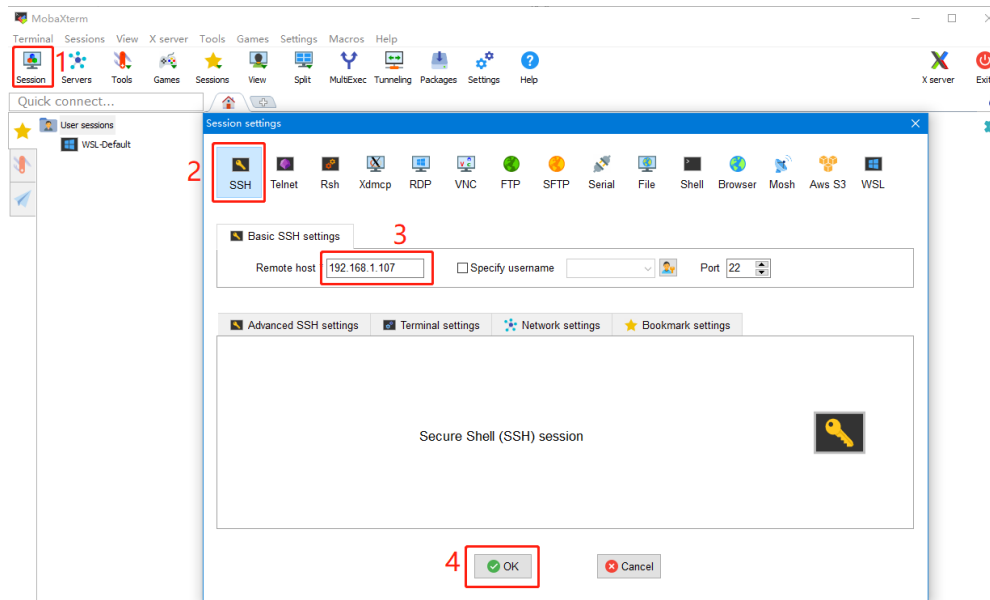


5. Or use the <https://angryip.org/> tool, scan all IP address in the current network organize by names, find the IP named Fluidd or Mailsail like shown below

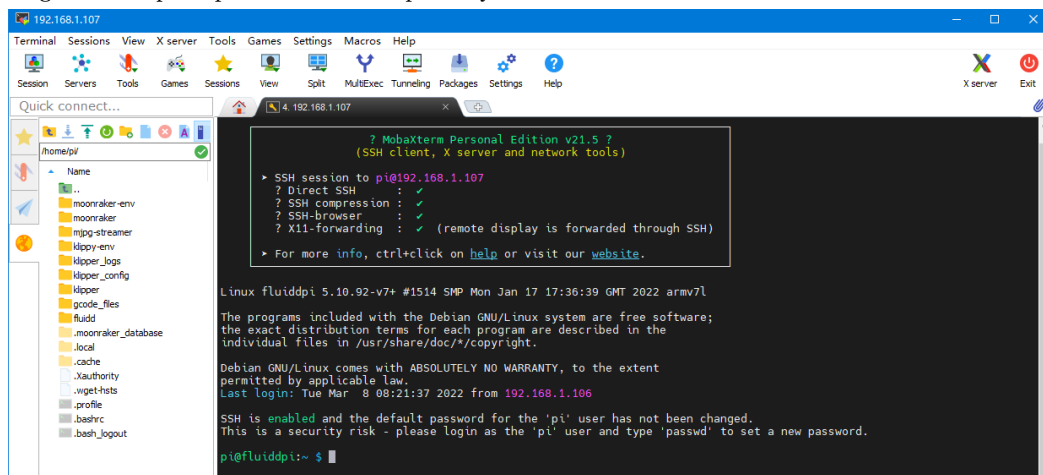


IP	Ping	主机名	端口 [3+]
192.168.1.107	71 毫秒	fluiddpi.local	80
192.168.1.106	0 毫秒	XTZJ-20211206JC.DHCP HOST	80,443
192.168.1.1	8 毫秒	[n/a]	80
192.168.1.100	5000 ...	[n/a]	[n/a]
192.168.1.101	4999 ...	[n/a]	[n/a]

6. Open MobaXterm and click “Session”, and click “SSH”, inset the raspberry pi IP into Remote host and click “OK” (note: your computer and the raspberry pi needs to be in the same network)

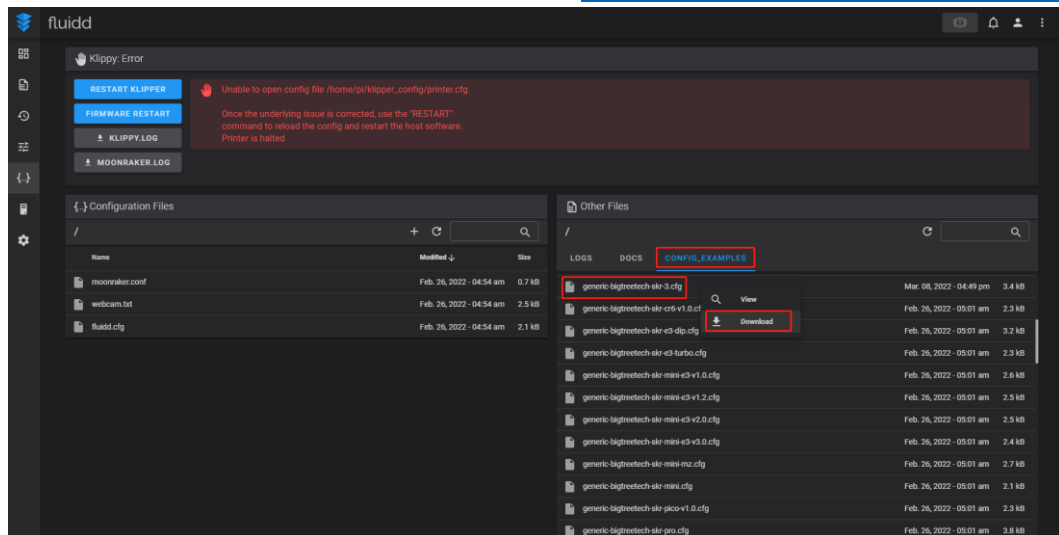


7. login as: pi password: raspberry

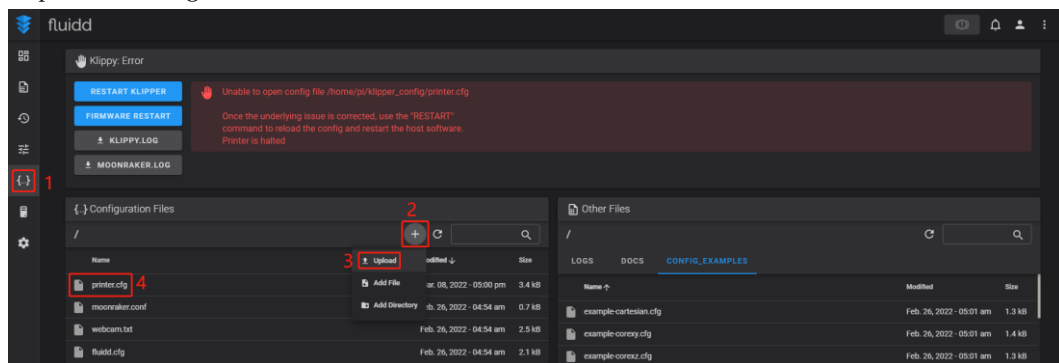


5.6 Configure Klipper

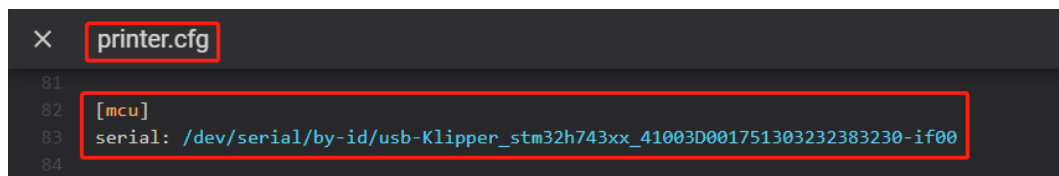
1. Enter your raspberry pi IP address into your browser to open the webUI, find the reference config for motherboard in the directory shown below, if there is no such config available, update your klipper source code to the newest version or download from github:<https://github.com/bigtreotech/SKR-3>



2. Upload your finished config file into Configuration Files, and rename to “printer.cfg”



3. Insert the correct motherboard ID



4. Refer to <https://www.klipper3d.org/Overview.html> for detailed configuration guide according to your machine type.

6. Firmware update

Update using Micro SD

1. Make sure Micro SD is formatted to FAT32
2. Rename your firmware file to “firmware.bin” (**note:** make sure your system is showing file suffix, if suffix is hided, “firmware.bin” will be shown as “firmware”)
3. Copy “firmware.bin” to the root directory of your SD card.
4. Insert Micro SD to the motherboard and power on, the bootloader will automatically update the firmware
5. The status indicator led will flash during the update process
6. When the led stops flashing and the firmware.bin file has been renamed to firmware.cur, the firmware has been successfully updated.

7. Precautions

1. When not using PT1000,do not connect any jumper in the PT1000 pins, otherwise 100K NTC will not work.
2. Maximum heated bed current is 10A, if high power heated bed is preferred, please use 24v to power the system and use a 24v heated bed.
3. CNC fan voltage selecting jumper must be inserted for CNC fan port to work correctly
4. If the board is plugged into your computer and not responding, make sure the USB/CAN selecting button is released and set to USB mode.
5. The Micro SD card slot is not spring loaded, please be careful when inserting the Micro SD card to prevent damage to the card slot. BTT is not responsible for any damage caused by forcefully inserting the Micro SD card

8. FAQ

Q: Max current of heated bed, heater cartridge, fan port?

A: heated bed: 10A continuous, 11A instantaneous

heater cartridge: 5.5A continuous, 6A instantaneous

Fan port: 1A continuous, 1.5A instantaneous

combined current of heater cartridges(E0, E1), stepper driver and fan port should not exceed 10A

Q: Can not update firmware with SD card

A: make sure your sd card is formatted to FAT32, firmware file name is “firmware.bin”, make sure your system is showing file suffix, if suffix is hided, “firmware.bin” will be shown as “firmware”