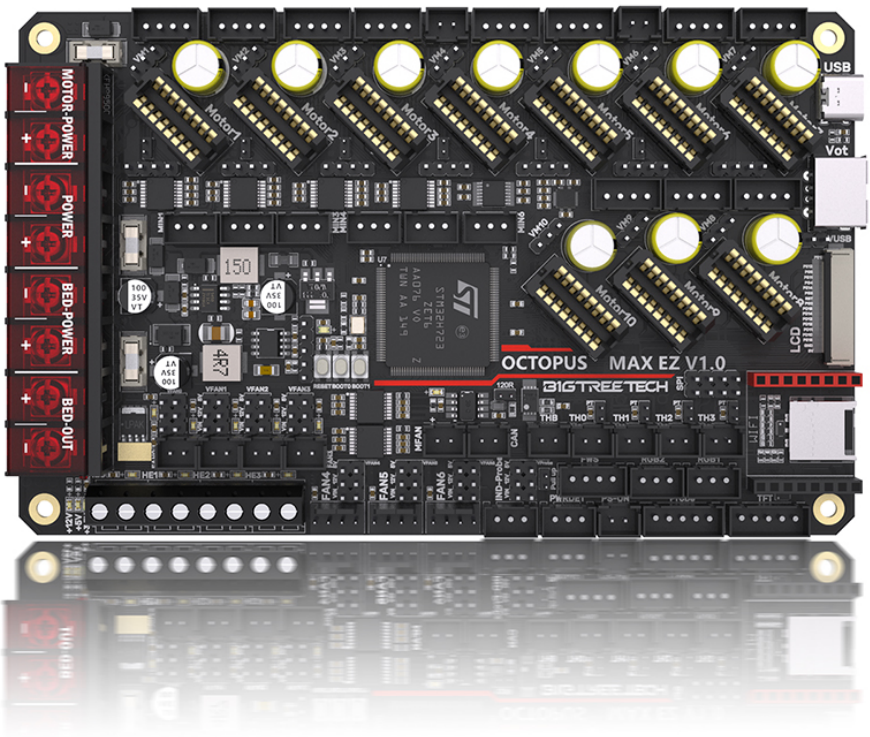
BIGTREETECH

Octopus MAX EZ V1.0

User Manual



Revision Log

Version	Date	Revisions
v1.00	6th October 2022	Initial Version
v1.01	27th September 2023	Instructions for updating Klipper firmware via DFU.

CONTENTS

Product Profile

BIGTREETECH Octopus MAX EZ is an optimized 32-bit 3D printer control board based on the Octopus Pro, using self-developed stepper motor driver sockets to enhance safety and user experience, with many new features not found on Octopus Pro to increase DIY potential.

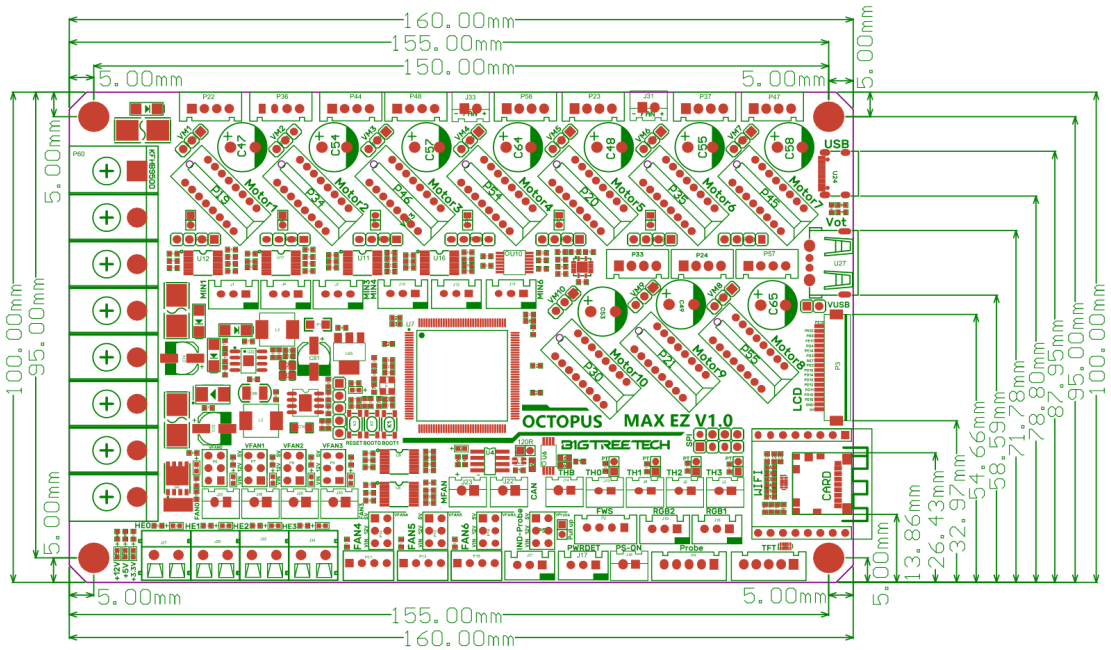
Feature Highlights

- 32 bit 550 MHz ARM Cortex-M7 series STM32H723ZET6 MCU;
- Onboard BOOT button to enable DFU mode to update bootloader;
- The thermistor circuit is protected to prevent MCU damage from shorted heated bed and heater cartridge connections;
- Selectable voltage (24V, 12V, 5V) for CNC fan, eliminating the need for external buck modules, thereby reducing the likelihood of motherboard damage.
- Upgraded with eFuse protection, which responses faster with strong protection, effectively protecting the motherboard from being damaged caused by short circuits, over-current, electric spark, etc.
- MCU firmware can be upgraded via SD card, or use DFU via Klipper's make flash command;
- 10 EZ driver sockets, working with pinless driver, safer to use; Onboard SPI and UART, can be used by simply setting in the firmware, no need for a jumper.
- Support power loss recovery, filament runout sensor, CAN, auto power-off, BLTouch, RGB, etc;
- Replaceable fuse for easy maintenance;
- 3 x 4 pins fan ports, also for connecting water cooling system;
- Onboard proximity switch port, supports NPN and PNP types, 24V, 12V, 5V voltage selectable;
- Onboard SPI interface for connecting acceleration sensor to enable Klipper's input shaping.

Specifications

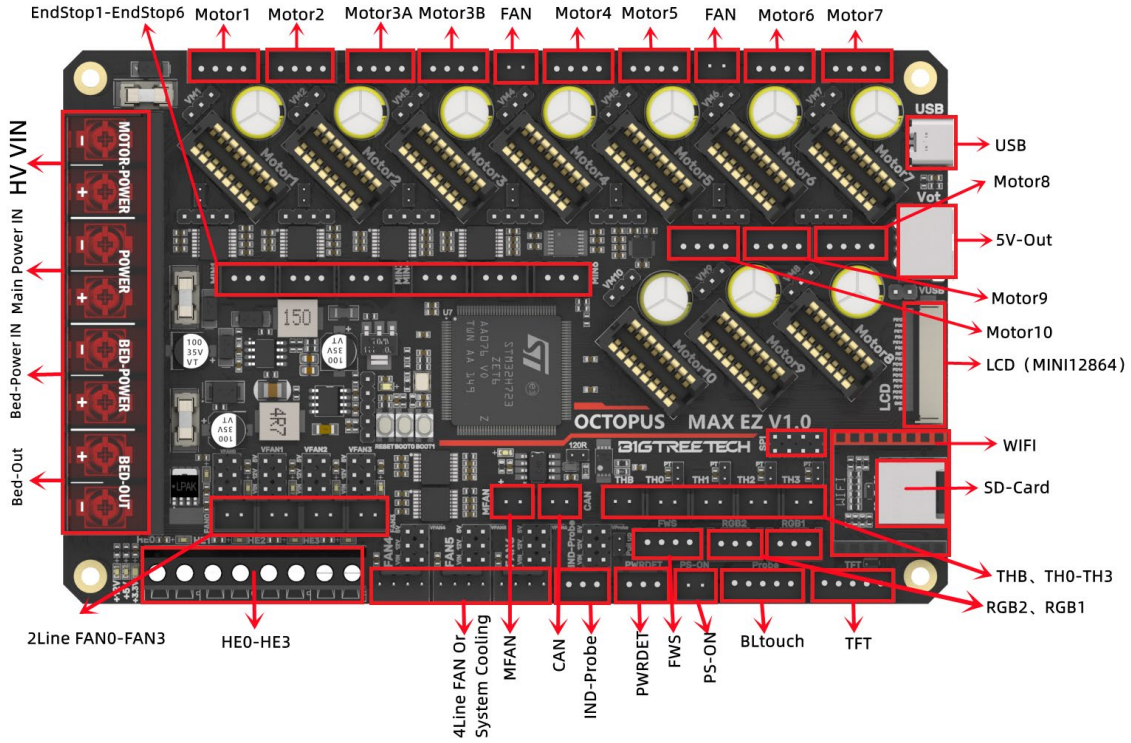
Dimensions	160mm x 100mm for details please refer to **BIGTREETECH Octopus MAX EZ V1.0-SIZE.pdf**
Mounting Size	Please refer to **BIGTREETECH Octopus MAX EZ V1.0-SIZE.pdf**
MCU	ARM Cortex-M7 STM32H723ZET6 550MHz
Driver Input Voltage	24V, HV(≤56V) Selectable
Motherboard Input Voltage	VIN=DC12V or DC24V
Heated Bed Input Voltage	BED IN=DC12V or DC24V
Logic Voltage	DC 3.3V
Heater Connection	Heated Bed (HB), Heater Cartridge (HE0, HE1, HE2, HE3)
HB Port Max Curre nt	10A Continuous, 12A Instantaneous
Heater Cartridge Max Current	5.5A Continuous, 6A Instantaneous
Fan Port	2 pins CNC Fan (FAN0, FAN1, FAN2, FAN3), 4 pins CNC Fan (FAN4, FAN5, FAN6), Always On (24V F AN x 2). CNC Fan and MFAN Voltage Selectable (5/12/24V)
Fan Port Max Curr ent	1A Continuous, 1.5A Instantaneous
Overall Max Curren t (Heater Cartridge+ Driver+All Fans)	< 12A
Expansion Port	BLTouch (Servos, Probe), PS-ON, FWS, PWRDET, RGBx2, SPI, IND-Probe, CAN, WIFI, TFT
Motor Driver	Support EZ5160, EZ2209, EZ2225, EZ2226, EZ2208, EZ 2130...
Driver Mode	SPI, UART
Motor Socket	Motor1, Motor2, Motor3 (Dual Motor Sockets), Motor4, Motor5, Motor6, Motor7, Motor8, Motor9, Motor10 10 Channels in Total
Thermistor	5 x 100K NTC, four of which are selectable for NTC and PT1000
Display	MINI12864 (FPC Connection), TFT Serial
PC Connection	Type-C
Supported Kinemat ics	Cartesian, Delta, Kossel, Ultimaker, CoreXY
Recommended Slicer/Console	Cura, Simplify3D, Pronterface, Repetier-host, Makerware

Dimensions

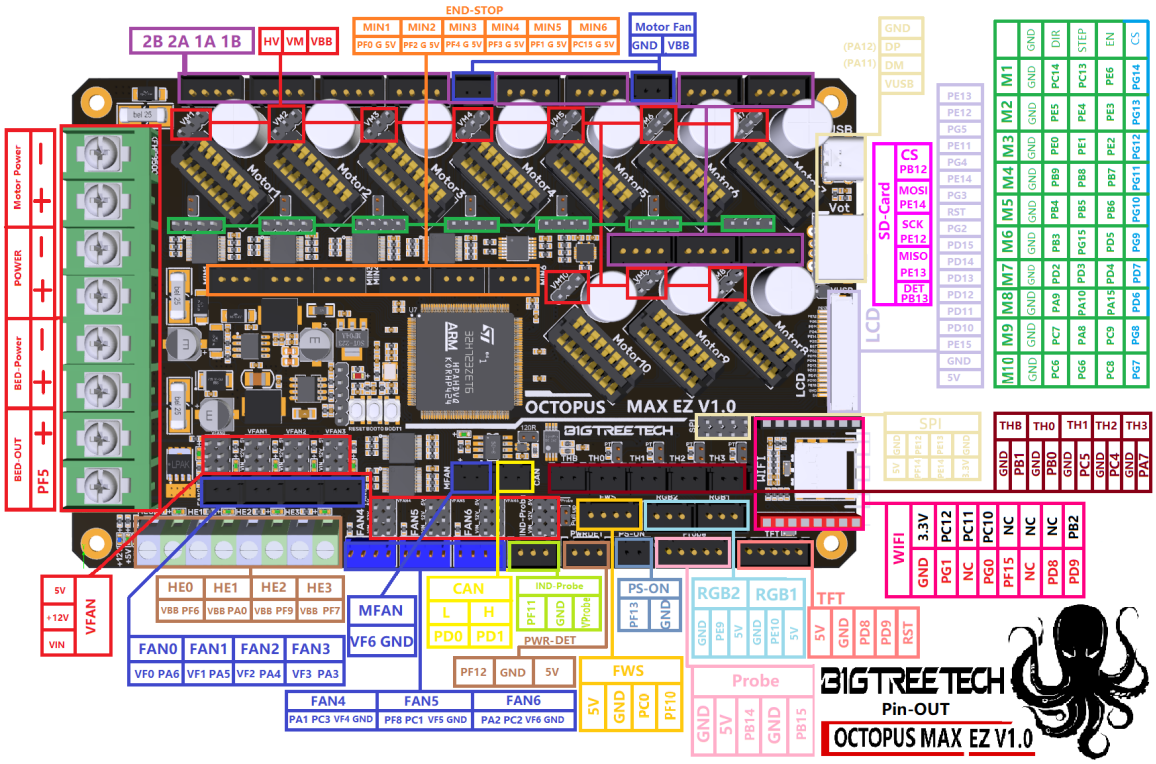

160.00mm
155.00mm
150.00mm
5.00mm
5.00mm
5.00mm
5.00mm
100.00mm
95.00mm
5.00mm
5.00mm
5.00mm
5.00mm
155.00mm
160.00mm
13.86mm
26.43mm
32.97mm
54.66mm
58.59mm
71.78mm
78.80mm
87.95mm
95.00mm
100.00mm
OCTOPUS MAX EZ V1.0
BIGTREETECH
USB
Vot
VUSB
LCD
CARD
HIFI

Peripheral Port

Connector Diagram



Pinout Diagram



Connection Description

USB Power Supply

After the Octopus MAX EZ has been powered, the Red light D32 on the left side of the MCU will light up, indicating power on. When using only USB to power the board or to supply power via USB, please insert the jumper cap onto the VUSB.

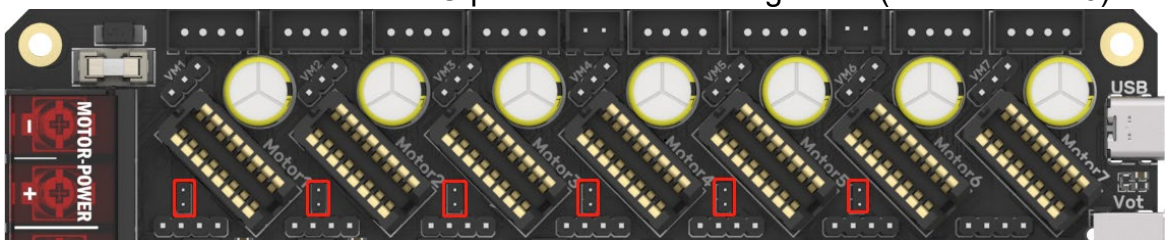
Stepper Motor Driver

UART/SPI Mode of Driver

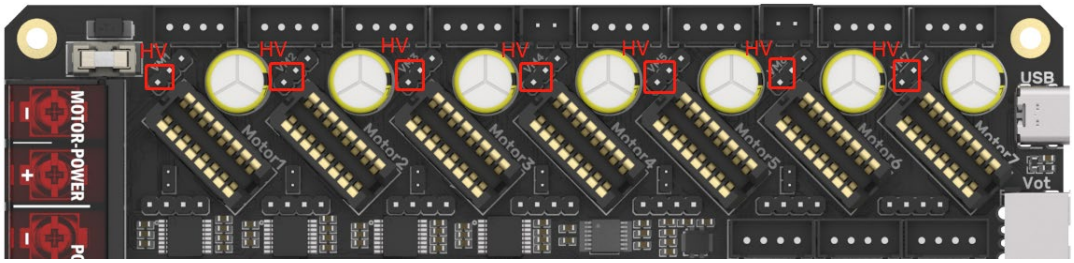
Set in the firmware, no need for a jumper.

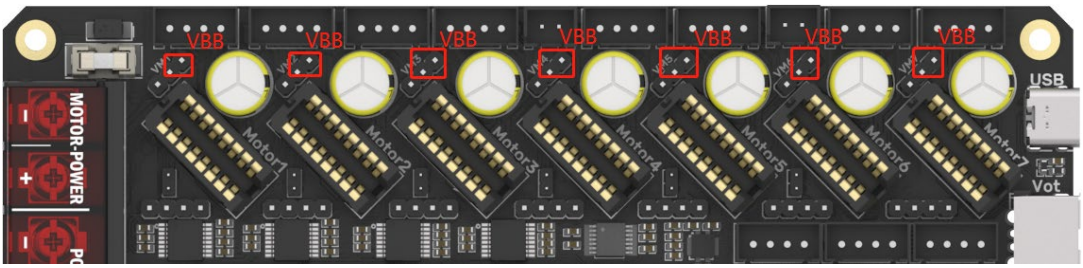
TMC Driver DIAG (Sensorless Homing)

When using sensorless homing, place jumpers according to the diagram below, there is no need to cut the DIAG pin off when not being used. (Motor1-Motor6).



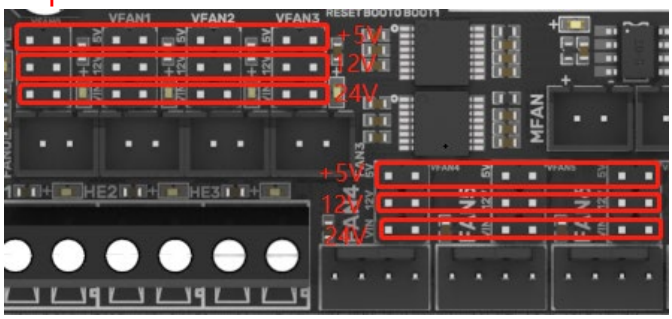
Driver Voltage Selection





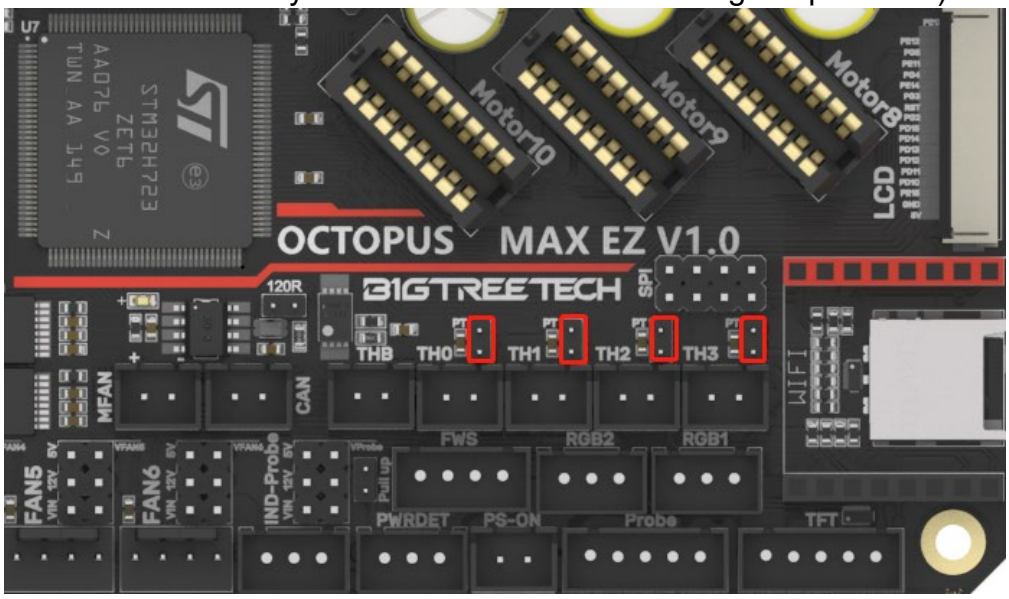
Voltage Selection for CNC Fan

The output voltage can be set to 5V, 12V or 24V through a jumper. (MFAN and FAN6 share the power supply VFAN6).
Note: Verify the fan's rated voltage before selecting to avoid damage. We are not responsible for fans burnt due to incorrect voltage selection.

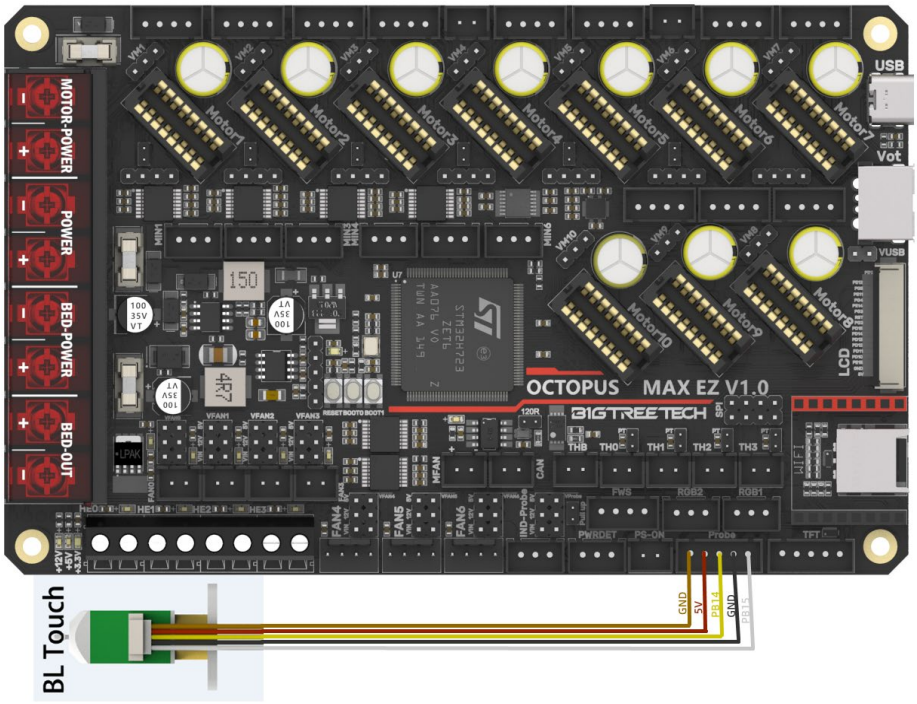


100K NTC or PT1000 Setting

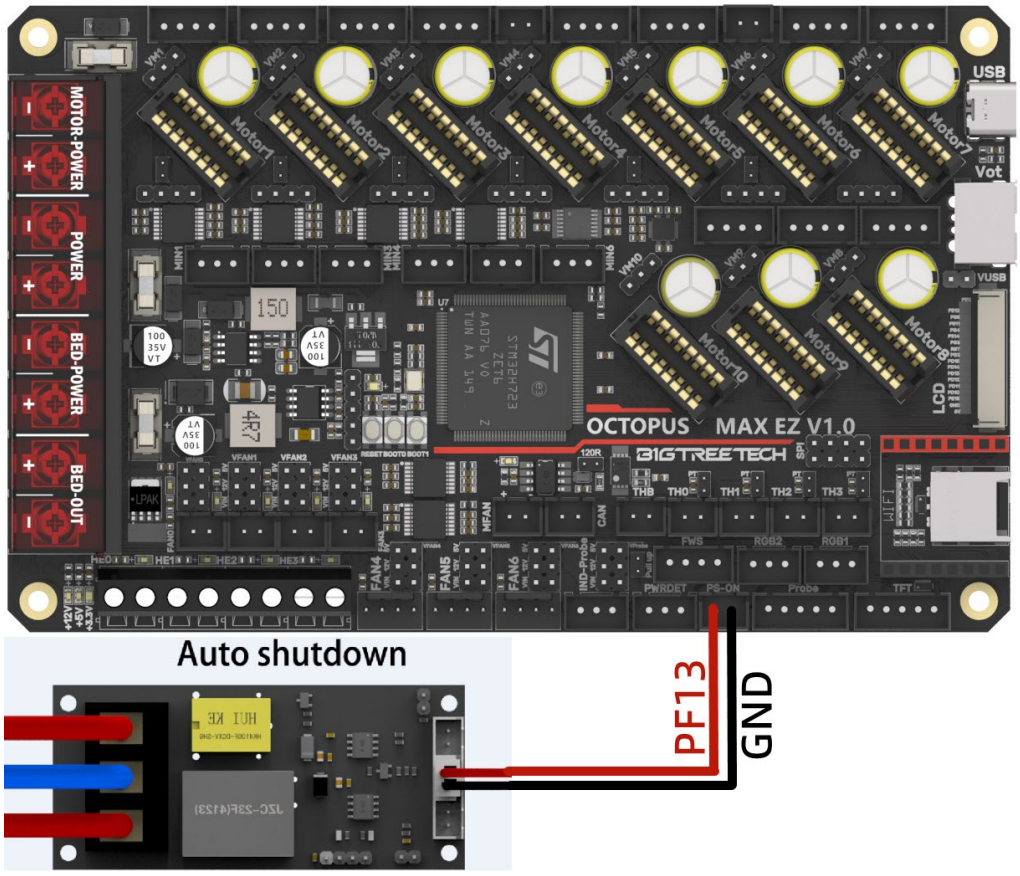
When using 100K NTC, no jumpers need to be connected, the pull-up resistance of TH0-TH3 is 4.7K 0.1%. When using PT1000, the pins indicated in the picture below need to be connected via jumpers, parallel connection of 4.12K 0.1% resistors, the pull-up resistance of TH0-TH1 is 2.2K. (Note: this method has a much lower accuracy than the MAX31865 in reading temperature.)



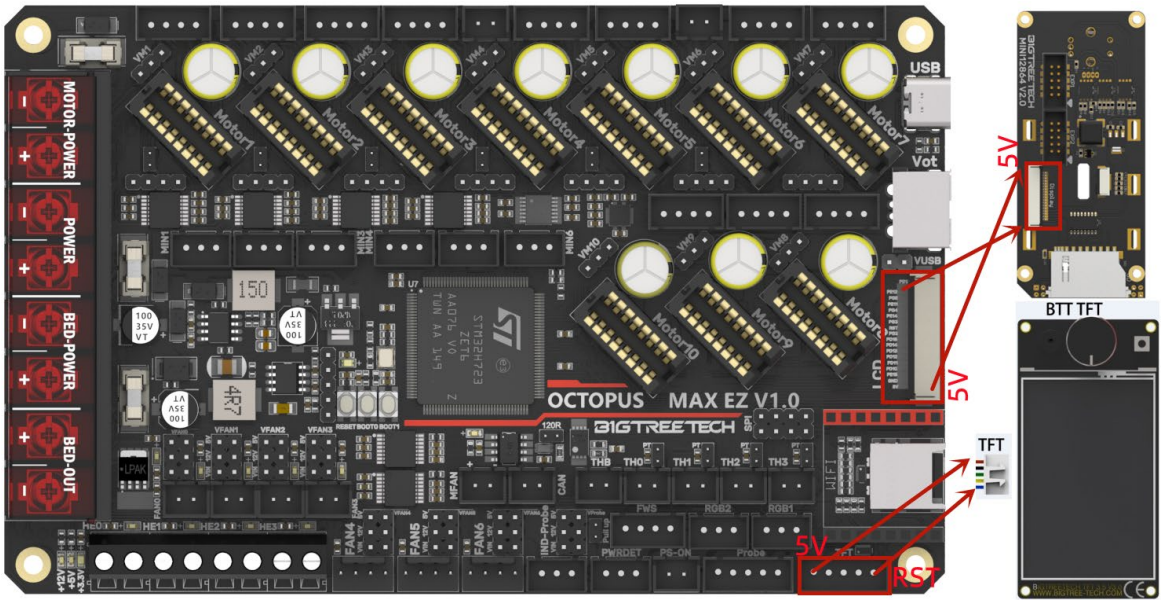
BLTouch Wiring



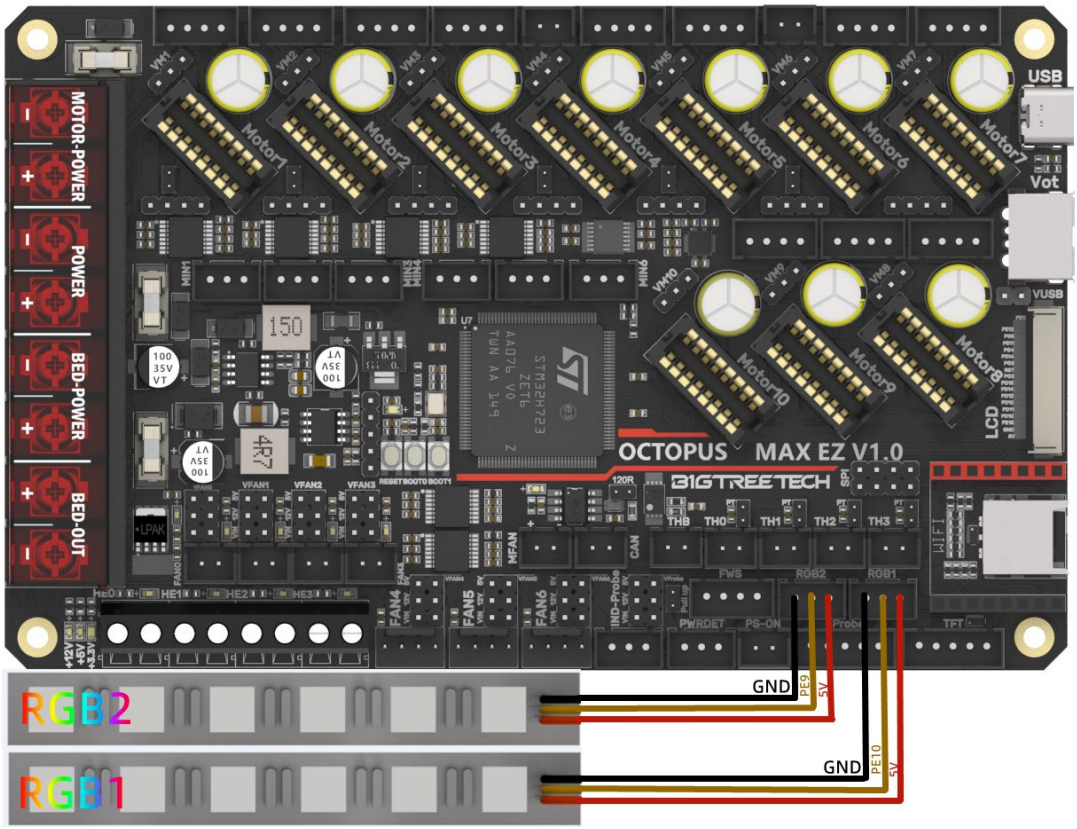
Auto Power Off (Relay V1.2) Wiring



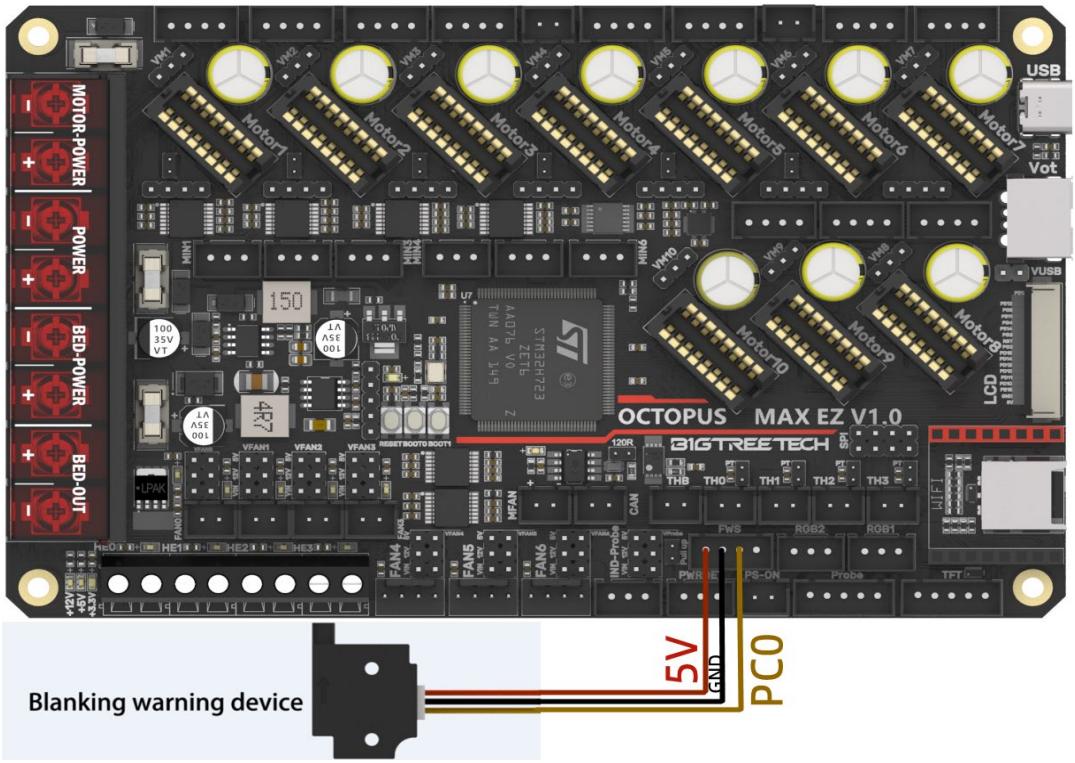
Connecting with MINI12864/TFT Screen



RGB Wiring

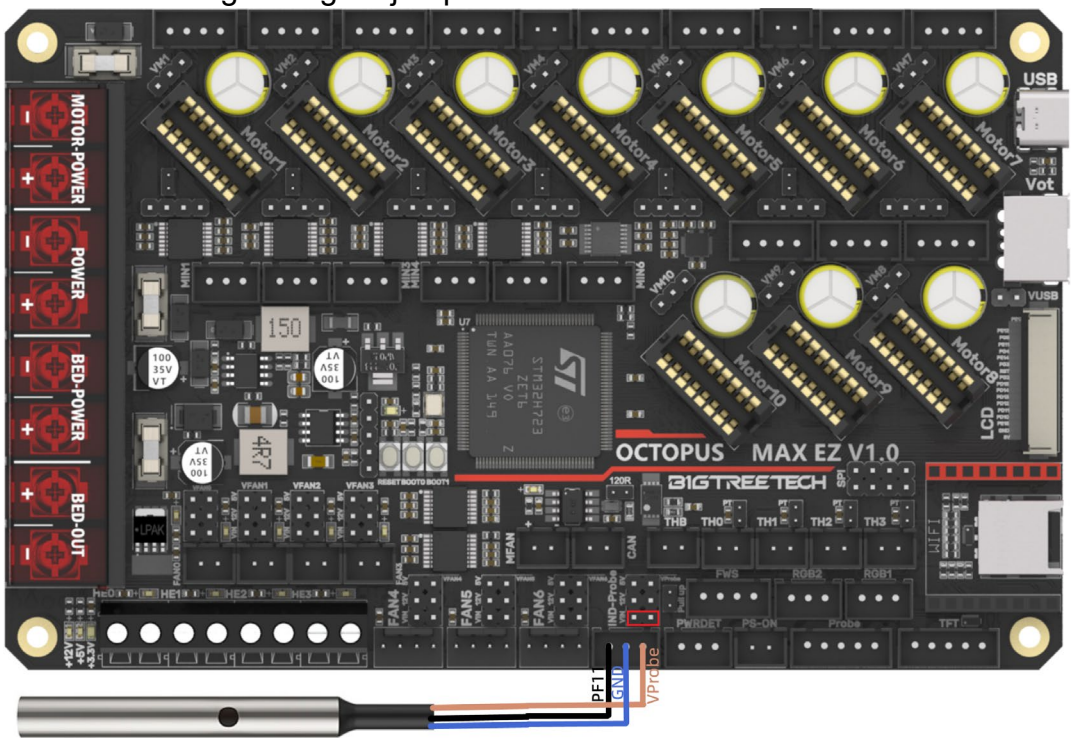


Filament Sensor Wiring

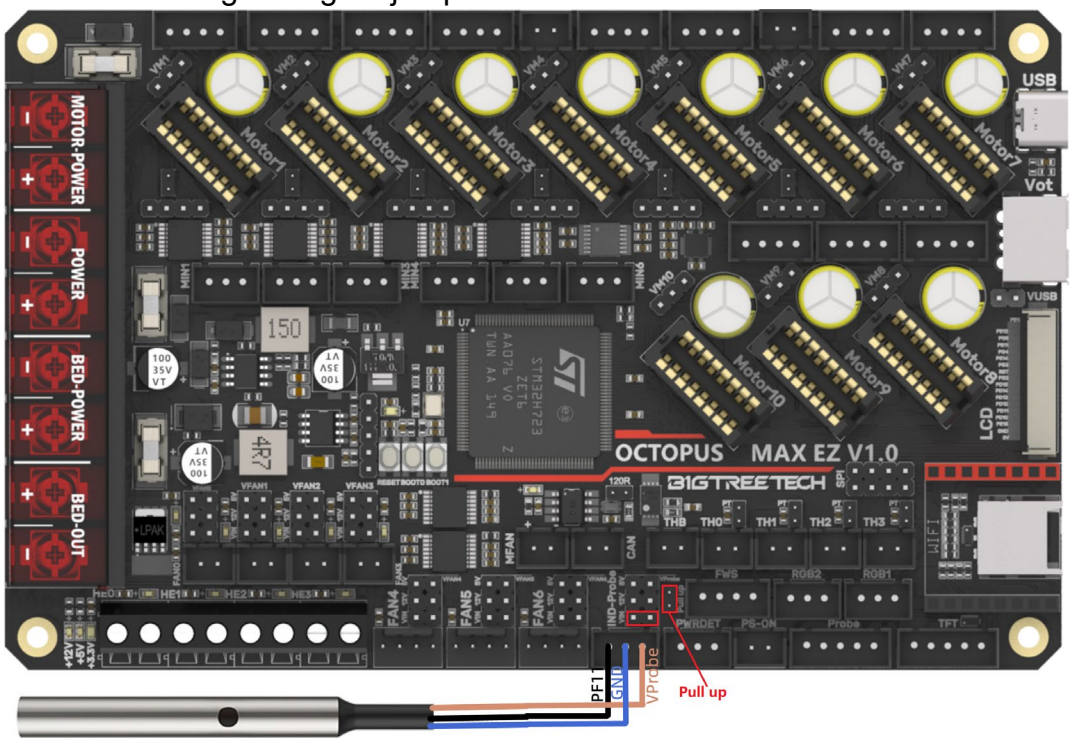


Proximity Switch Wiring

As shown in the figure below, 24V as an example, normally open (NPN type), no need for shorting through a jumper:

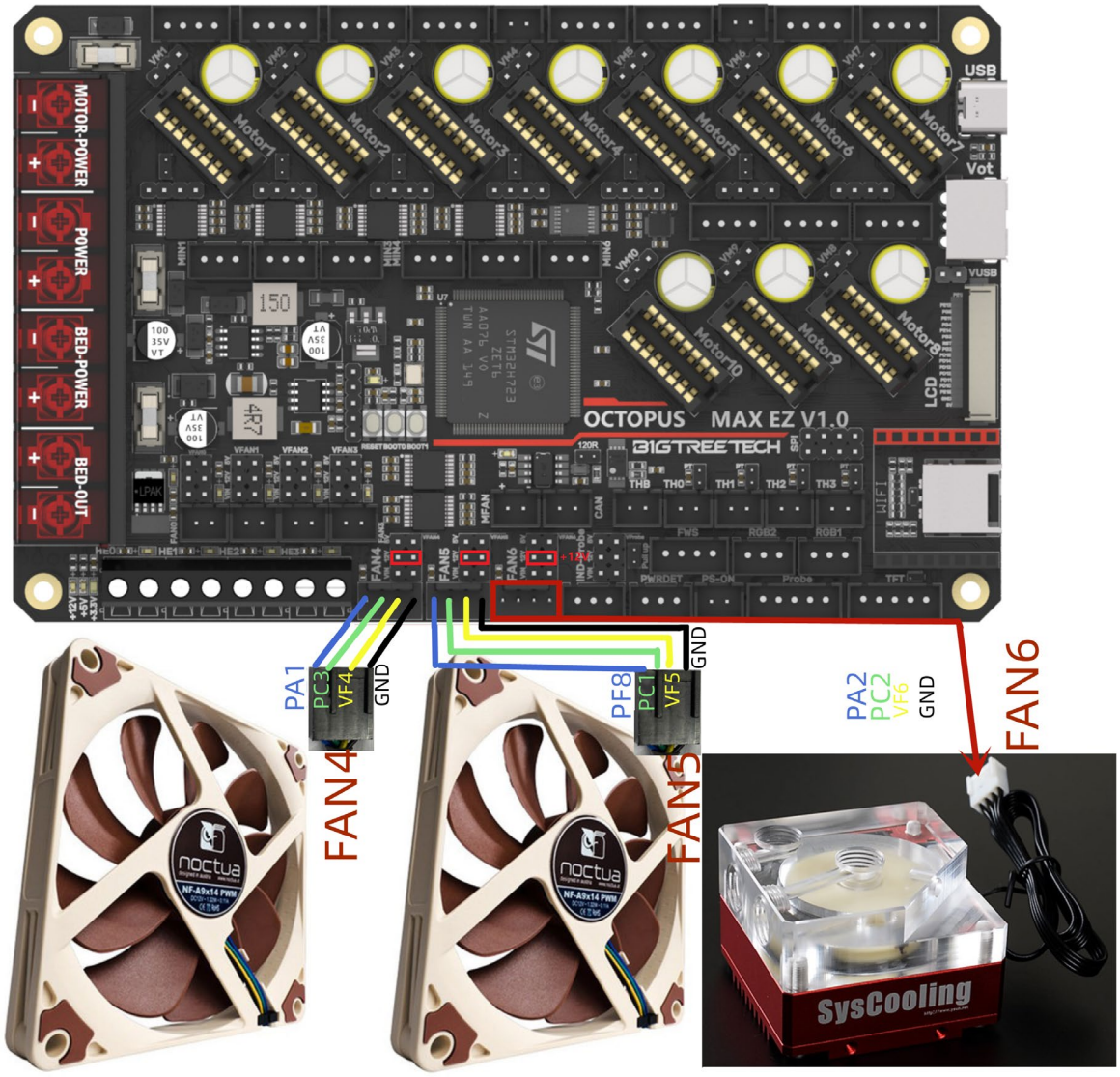


As shown in the figure below, 24V as an example, normally closed (PNP type), need for shorting through a jumper.



Wiring of 4 pins CNC Fan and Water Cooling System

(12V as an example:)



Marlin

Install Compiling Environment

https://github.com/bigtreetech/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.

Download Marlin Firmware

1. Download the newest bugfix version of Marlin from the official website: https://github.com/MarlinFirmware/Marlin/tree/bugfix-2.0.x
2. Download pre-configured firmware from our GitHub page: https://github.com/bigtreetech/BIGTREETECH-OCTOPUS-Max-EZ

Configure Firmware

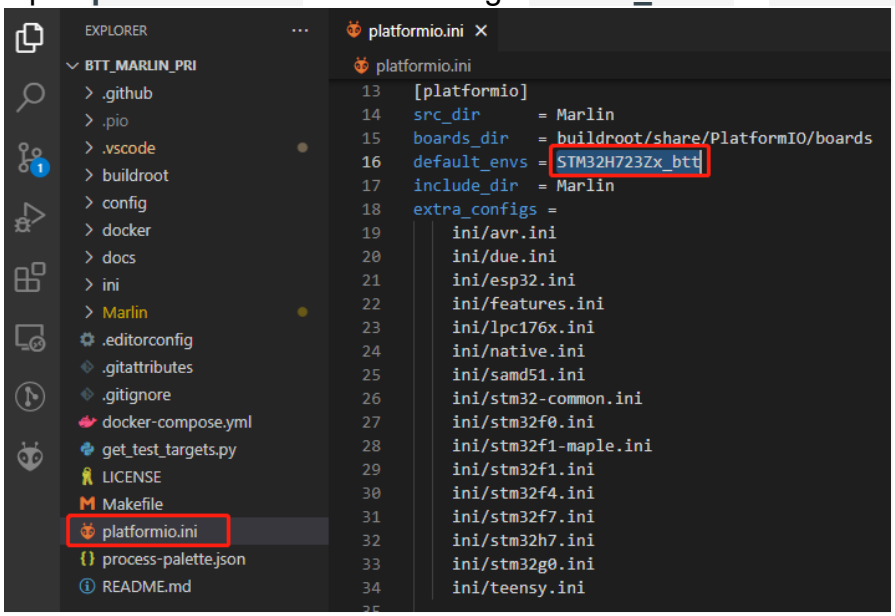
Open Marlin Project

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.

Compiling Environment

Open **platformio.ini** file and change **default_envs** to **STM32H723Zx_btt**.

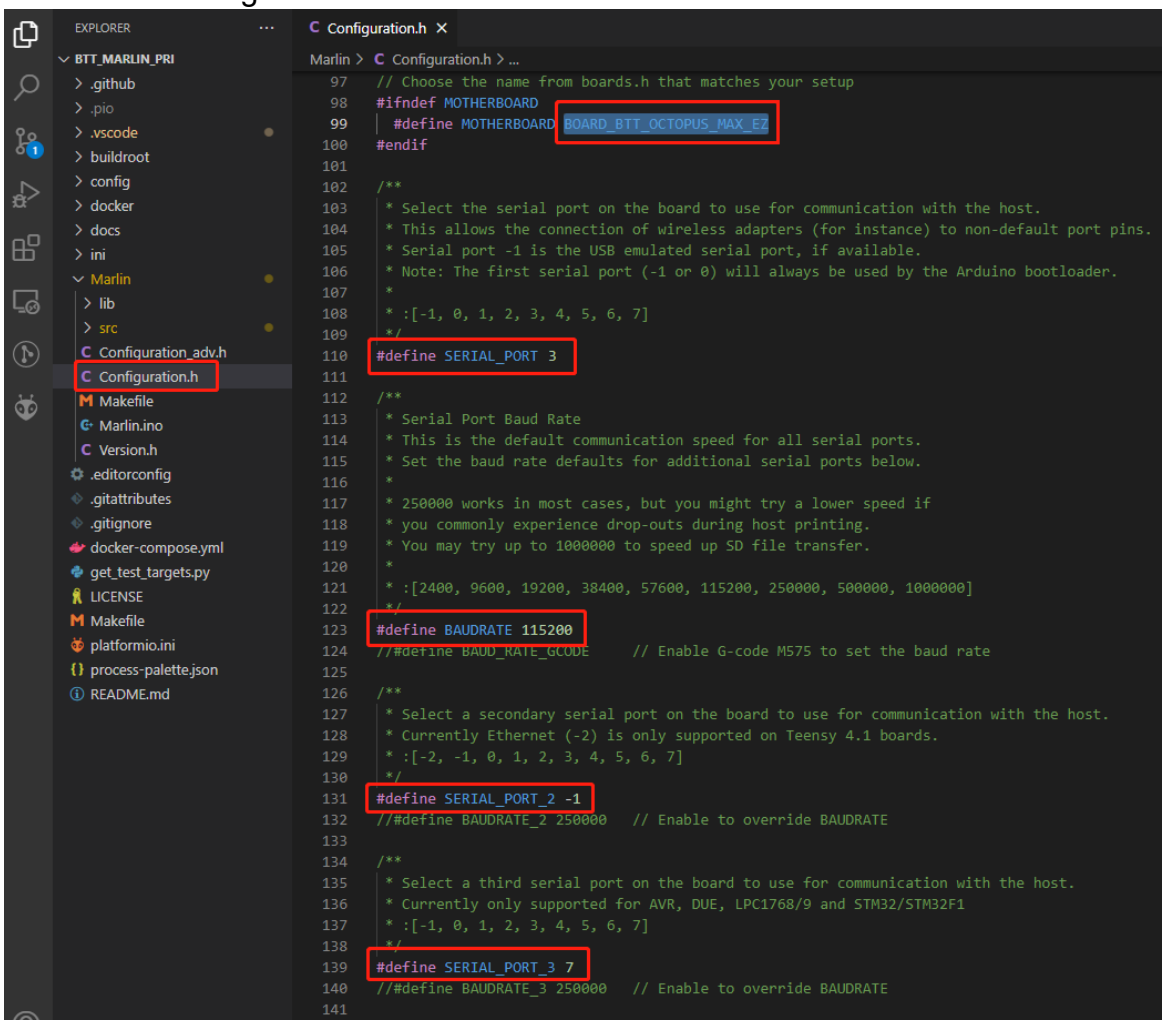


Configure Motherboard and Serial Port

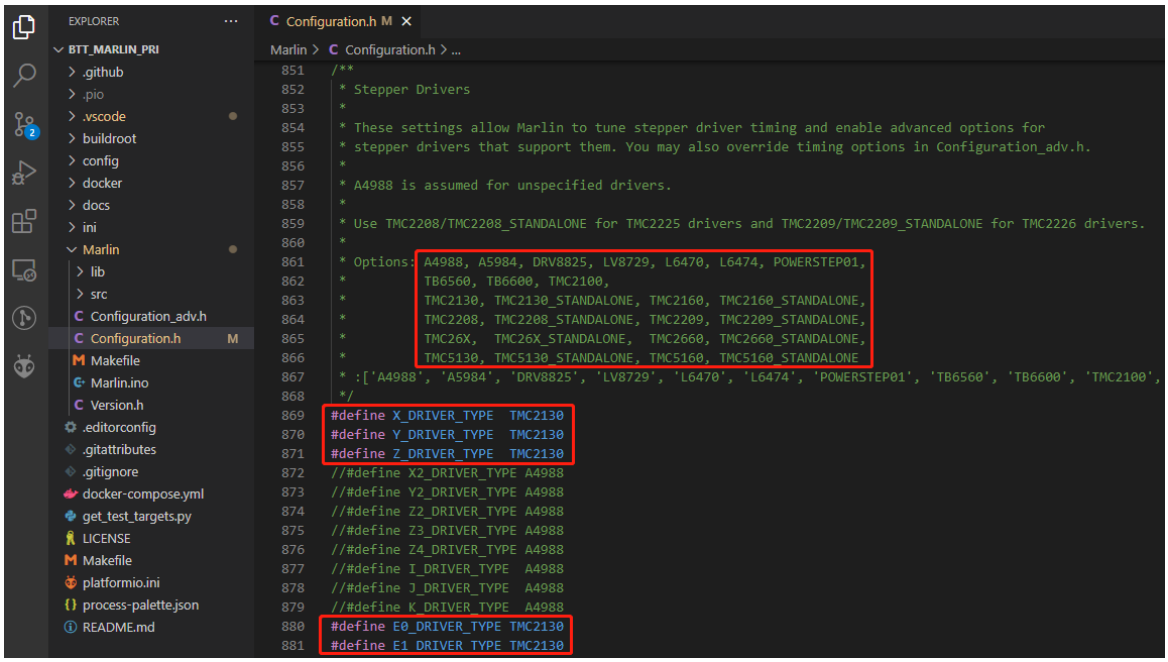
Set **MOTHERBOARD** to **BOARD_BTT_OCTOPUS_MAX_EZ**

#define MOTHERBOARD BOARD_BTT_OCTOPUS_MAX_EZ
#define SERIAL_PORT 3 (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 7 (enable WIFI serial port)

The above settings can be enabled as needed.

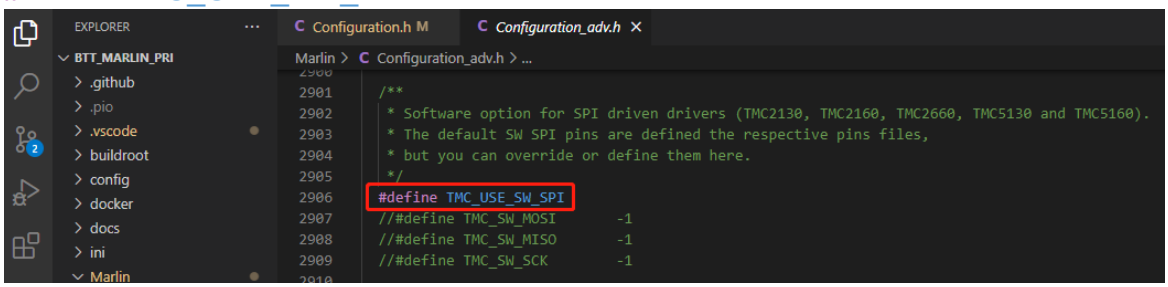


Configure Stepper Driver

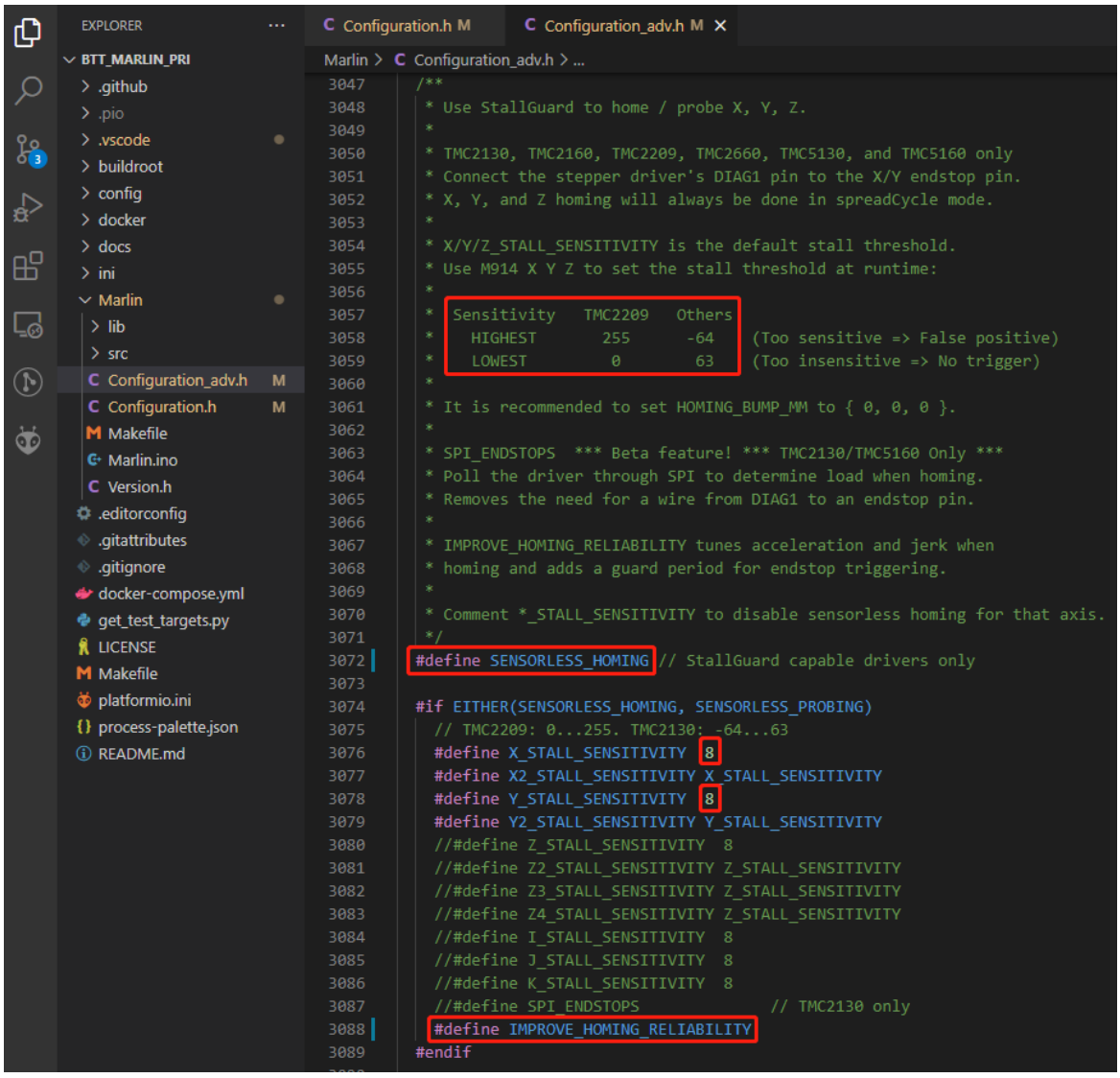


When using SPI mode, you need to enable TMC_USE_SW_SPI in Configuration_adv.h

#define TMC_USE_SW_SPI



Sensorless Homing

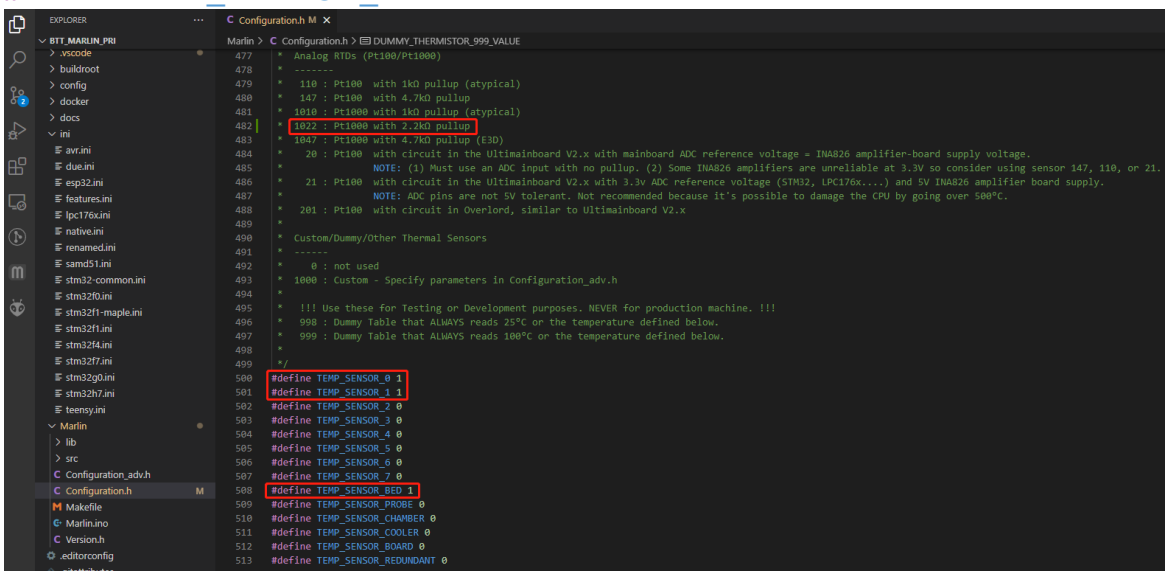


#define SENSORLESS_HOMING // enable sensorless homing
#define xx_STALL_SENSITIVITY 8 // sensitivity setting, TMC2209 range from 0 to 255, higher number results in more sensitive trigger threshold, sensitivity too high will cause endpoint to trigger before gantry actually moves to the end, lower number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

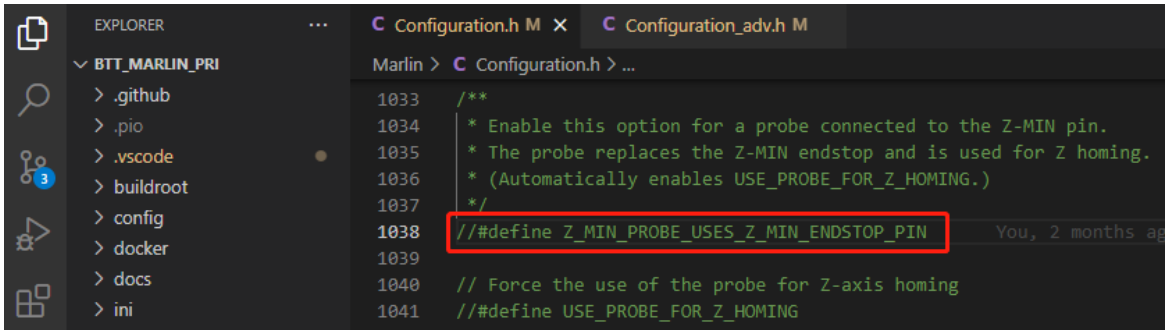
100K NTC or PT1000

Use jumpers to set the thermistor pull-up resistor to 4.7K (with 100K NTC) or 2.2K (with PT1000). In Marlin firmware, 1 represents 100K NTC + 4.7K pullup, 1022 represents PT1000 + 2.2K pullup. Note: Accuracy will be much lower than MAX31865 with this method.

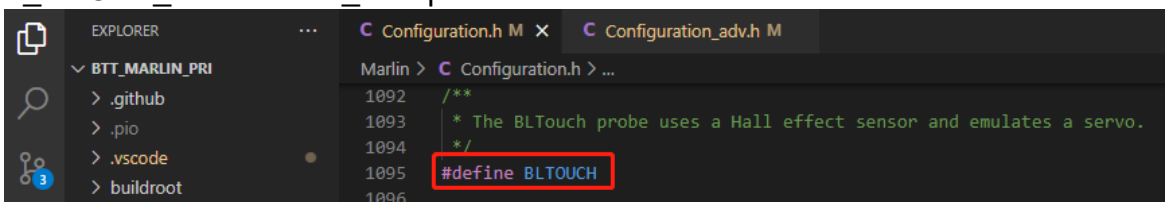
```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 1
#define TEMP_SENSOR_BED 1
```



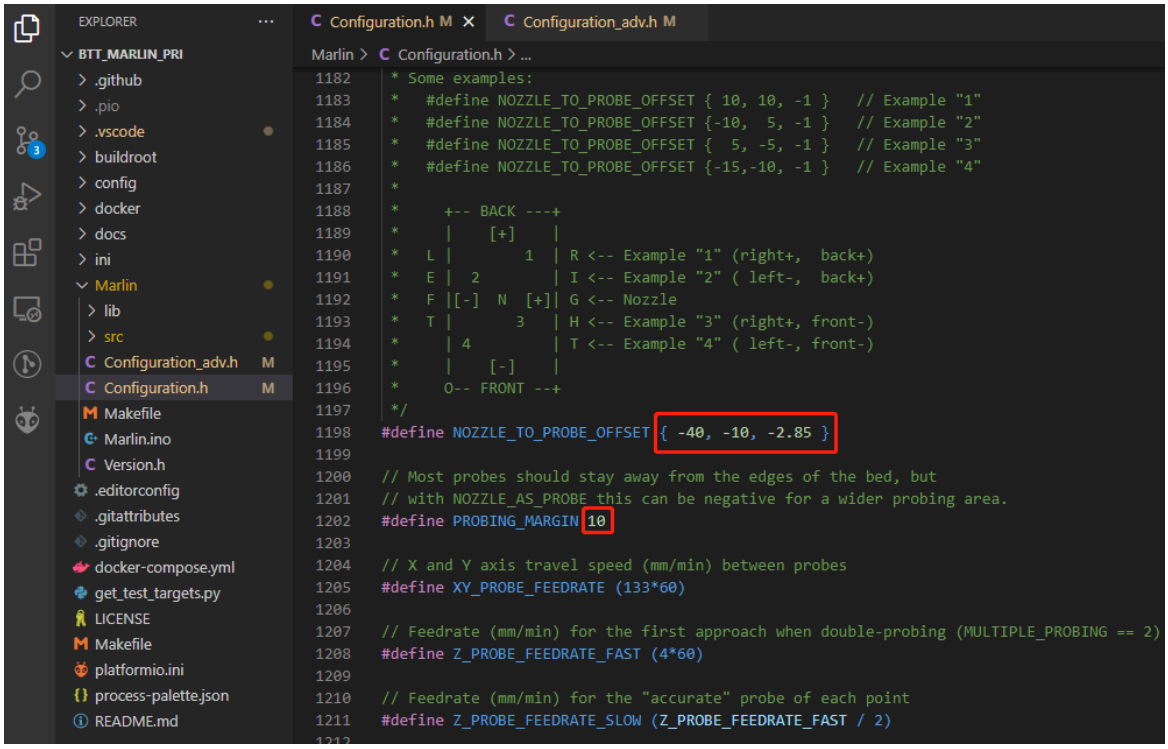
BLTouch




```
//#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN // Do not remap the Z_PROBE_PIN to the Z_MIN port.
```

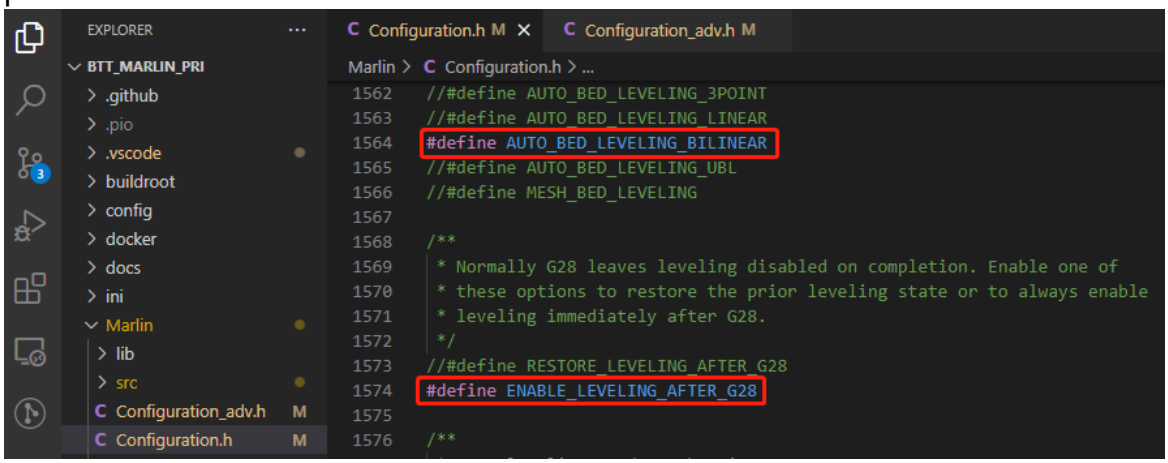



```
#define BLTOUCH  // Enable BLTouch
```



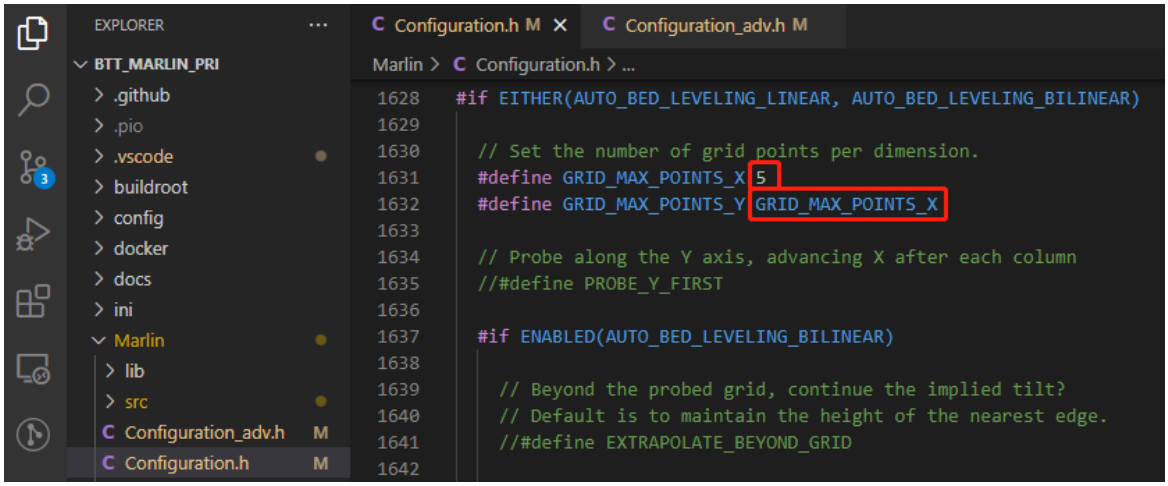
#define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 } // set BLTouch probe offset

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



#define AUTO_BED_LEVELING_BILINEAR // set probe pattern

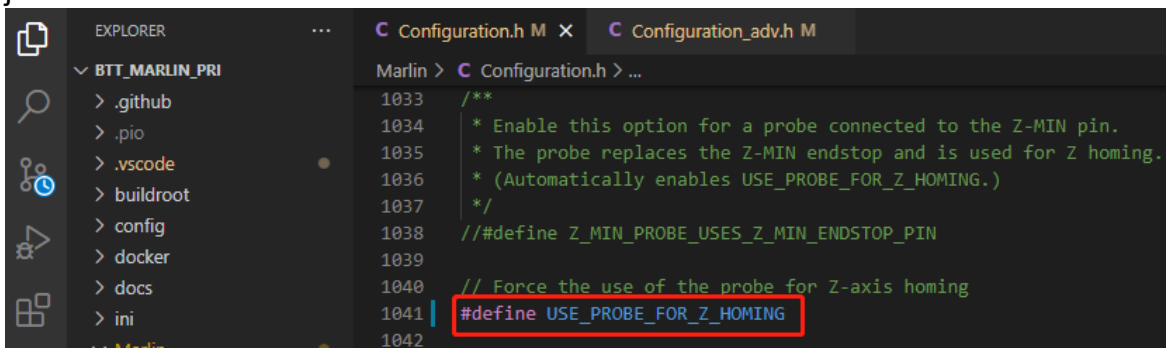
#define RESTORE_LEVELING_AFTER_G28 // apply leveling after G28 homing command



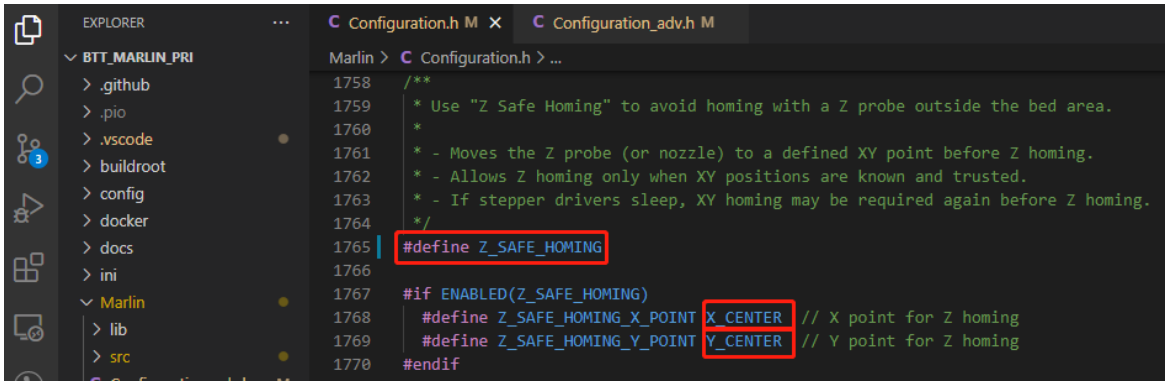
#define GRID_MAX_POINTS_X 5 // set number of probe points for X axis, usually 5 point is sufficient

#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X // set the 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, just set it in the firmware.

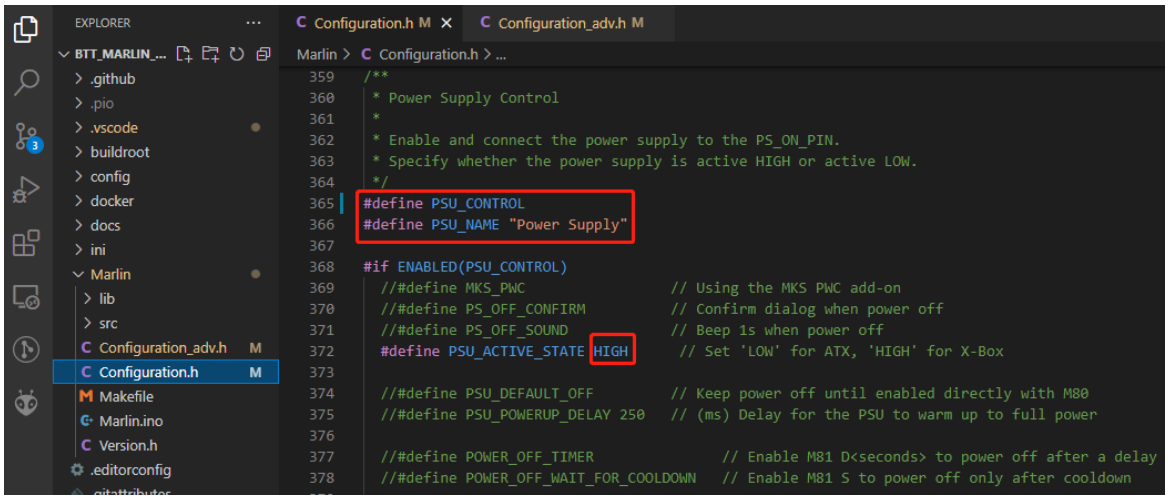


#define USE_PROBE_FOR_Z_HOMING // use Z Probe(BLTouch) for Z homing



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

Auto Power Off(Relay V1.2)



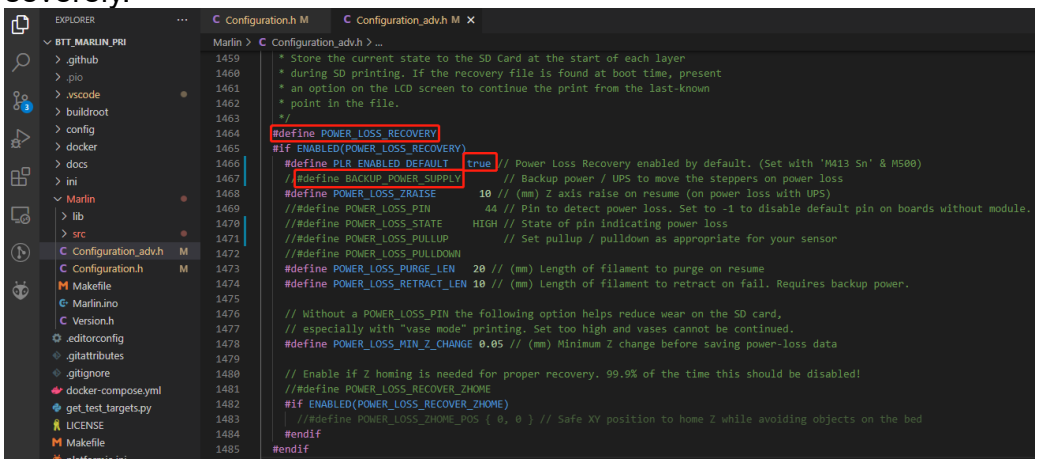
#define PSU_CONTROL // enable PSU control to turn on and off using M80 and M81

#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.

Power Loss Recovery

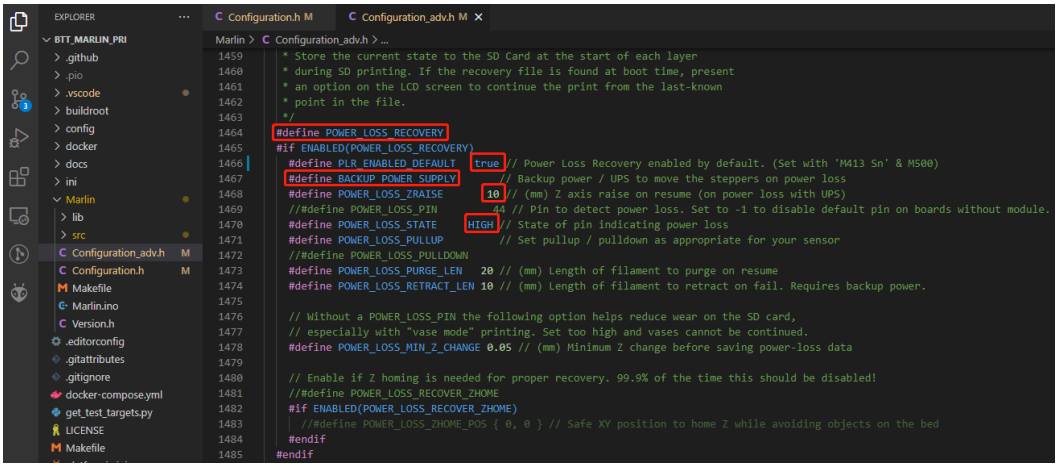
There are two methods for power loss recovery

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
#define PLR_ENABLED_DEFAULT true // true default to power loss recovery enabled

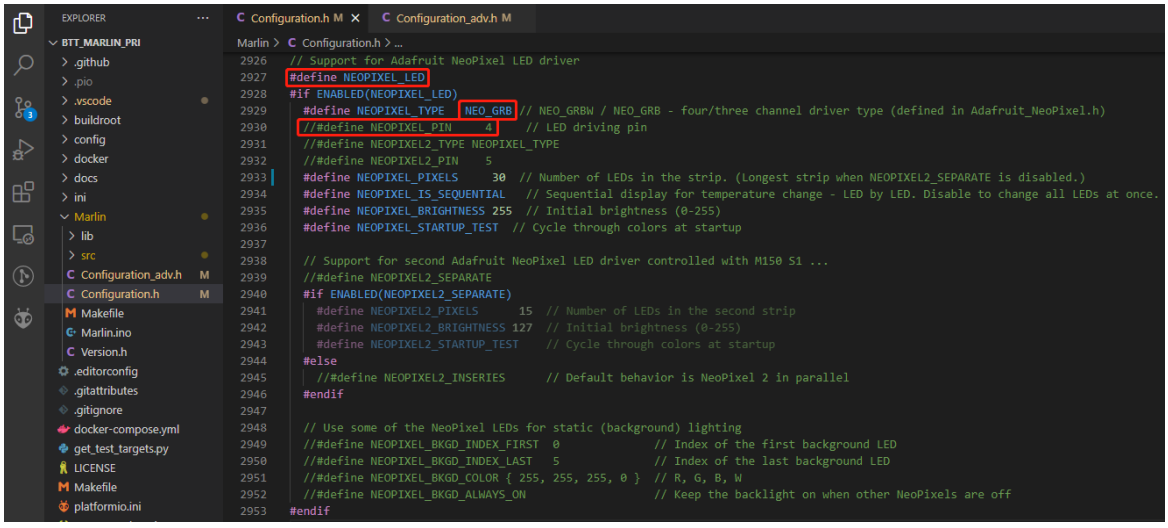
2. External UPS 24V V1.0 module, when power is cut, the module will provide power to the board and signal the board to save current print status to SD card. This method has virtually no effect on the life of the SD card.



#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

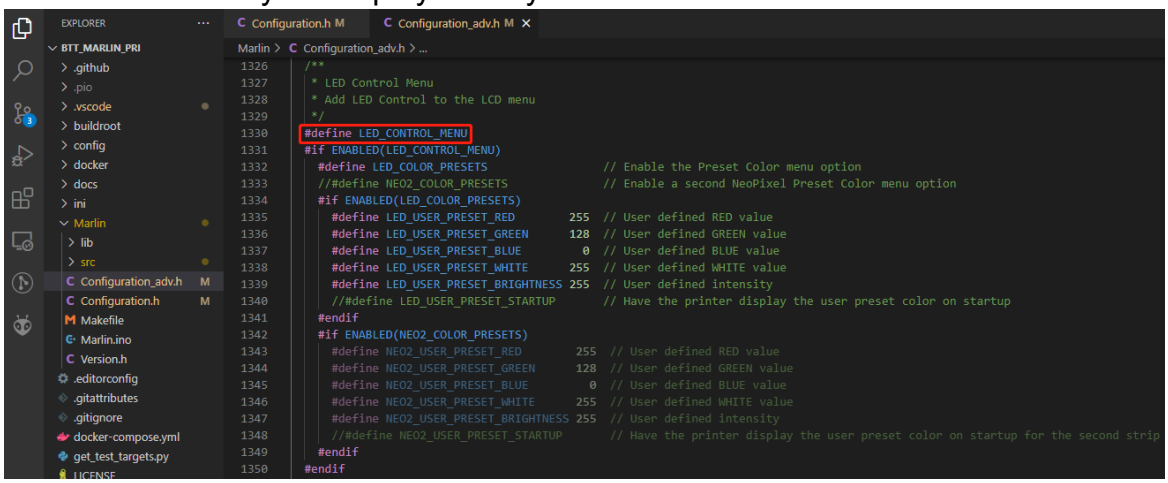
#define POWER_LOSS_STATE HIGH // set signal level, UPS 24V V1.0 returns low level when not triggered and HIGH level when power is cut, thus this setting needs to be HIGH

RGB



#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

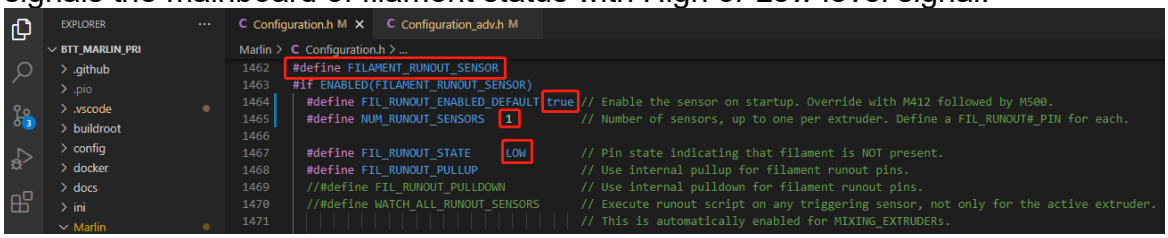
If you are using displays like LCD2004, 12864, mini12864, etc., you can also control RGB from your display directly.



#define LED_CONTROL_MENU // add led control to your menu.

Filament Sensor

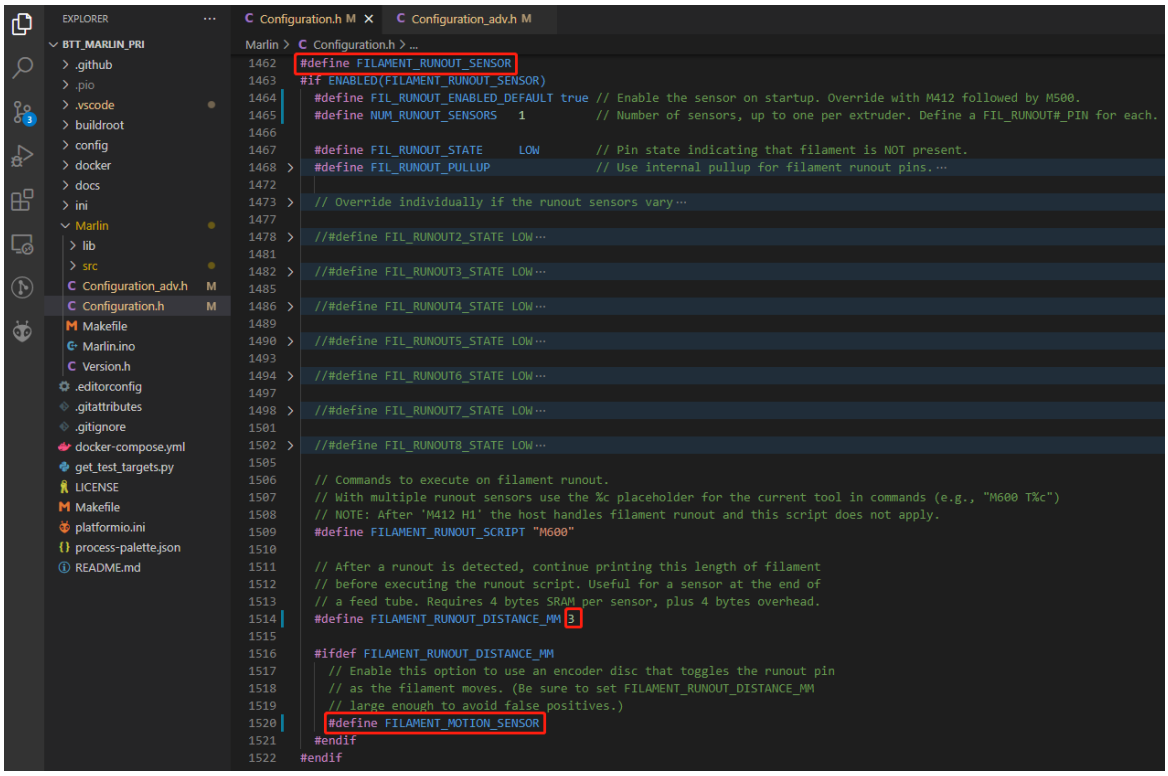
Standard filament run out sensors are usually comprised of a microswitch which signals the mainboard of filament status with High or Low level signal.



#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. Set according to the actual situation of the module. If the module sends a low level when the filament is abnormal, set it to LOW.

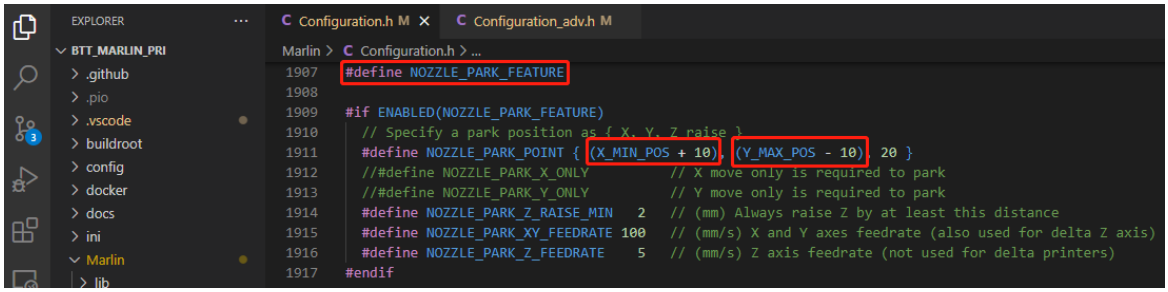
Smart Filament Sensor (SFS V1.0)

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

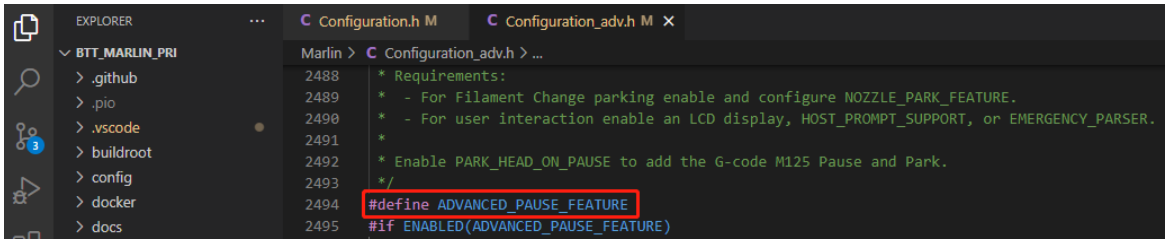


#define FILAMENT_MOTION_SENSOR // set encoder type
#define FILAMENT_RUNOUT_DISTANCE_MM 7 // set sensitivity, SFS V1.0 nominal setting should be 7mm, which means if no signal of filament movement is detected after 7mm of filament travel command, filament error will be triggered.

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



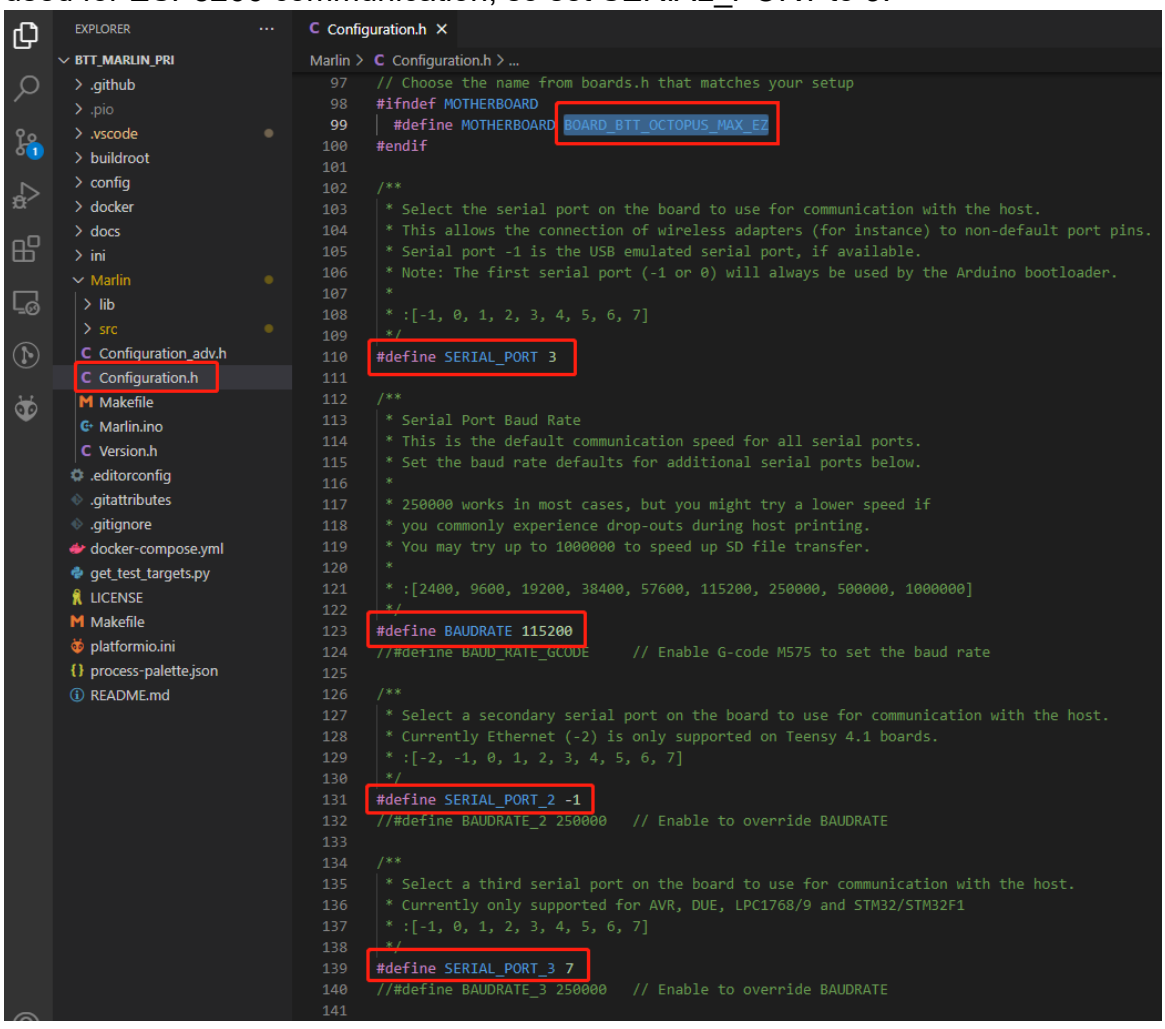
#define NOZZLE_PARK_FEATURE // park nozzle
#define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 } // set the X, Y and Z offset coordinate of the nozzle



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

ESP3D

In Marlin, simply set the correct "SERIAL_PORT" and "BAUDRATE". UART3 is used for ESP8266 communication, so set SERIAL_PORT to 3.

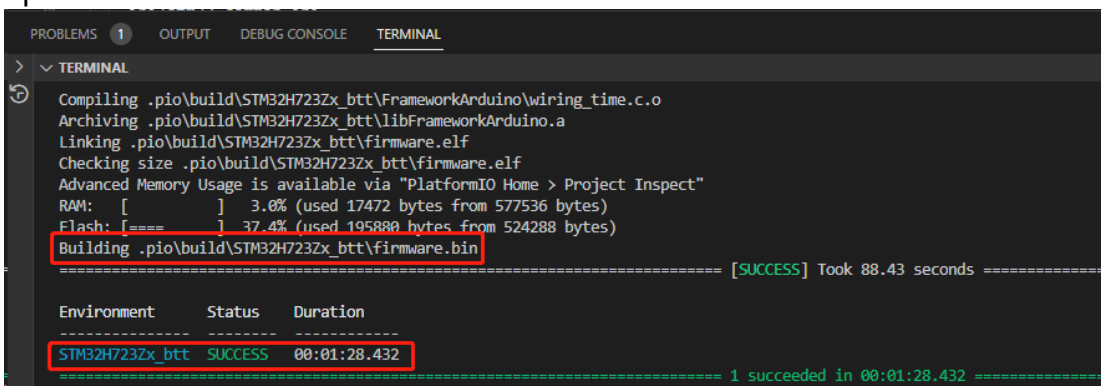


The newest ESP3D firmware can be found at https://github.com/luc-github/ESP3D, compile your own binary file and rename it to "esp3d.bin", copy it to the root directory of the SD card, insert into the motherboard and press the reset button. The bootloader will update the firmware to ESP8266 automatically.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

Compile Firmware

1. Click "√" to compile firmware.

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



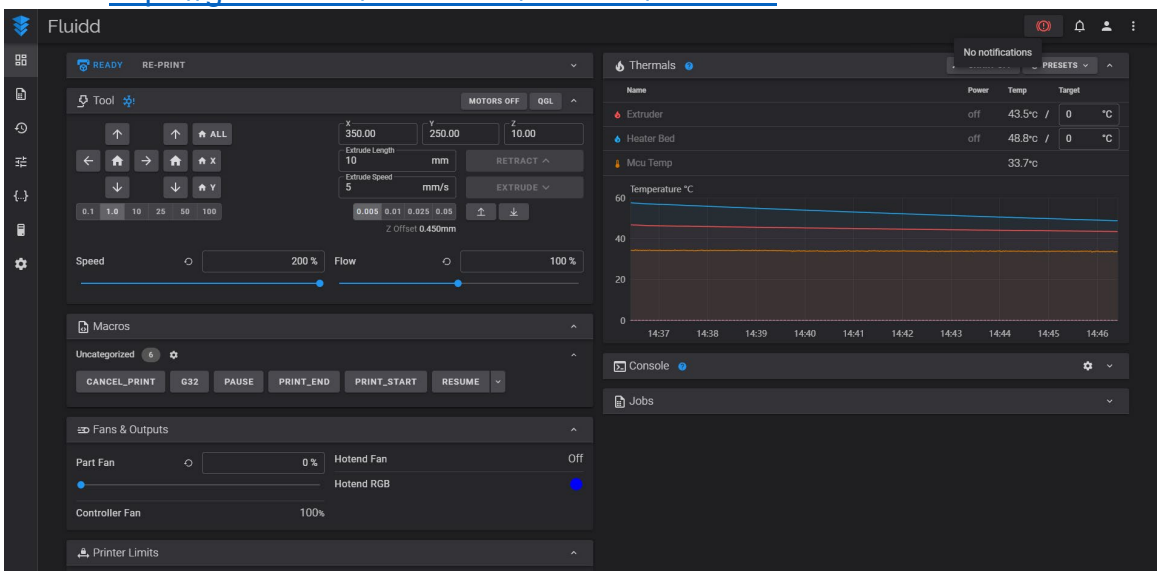
Klipper

Preparation

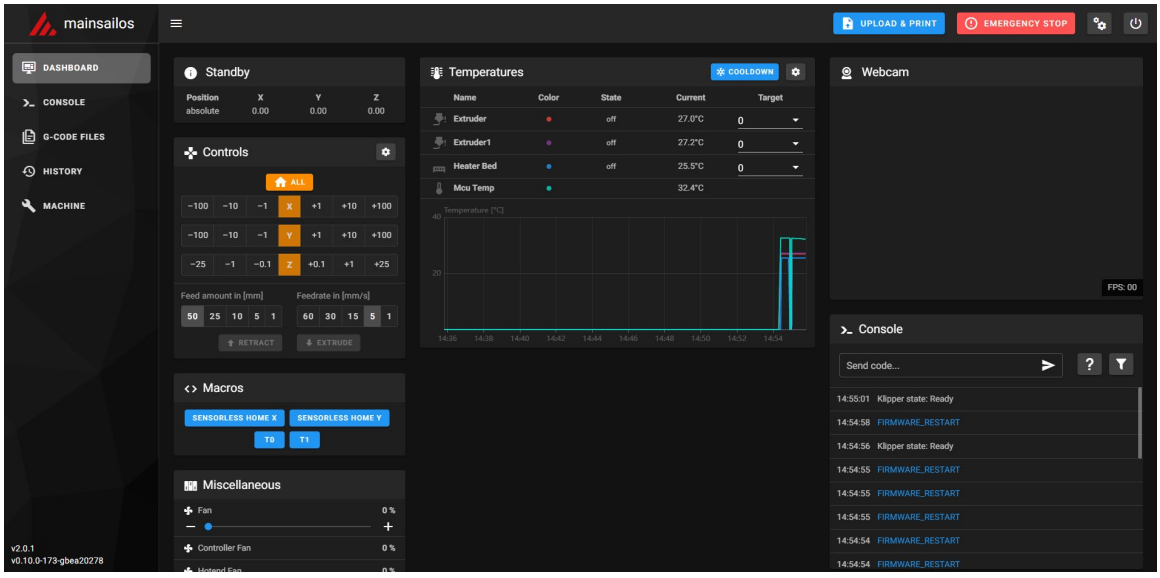
Download OS Image

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

Fluidd: https://github.com/fluidd-core/FluiddPI/releases



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



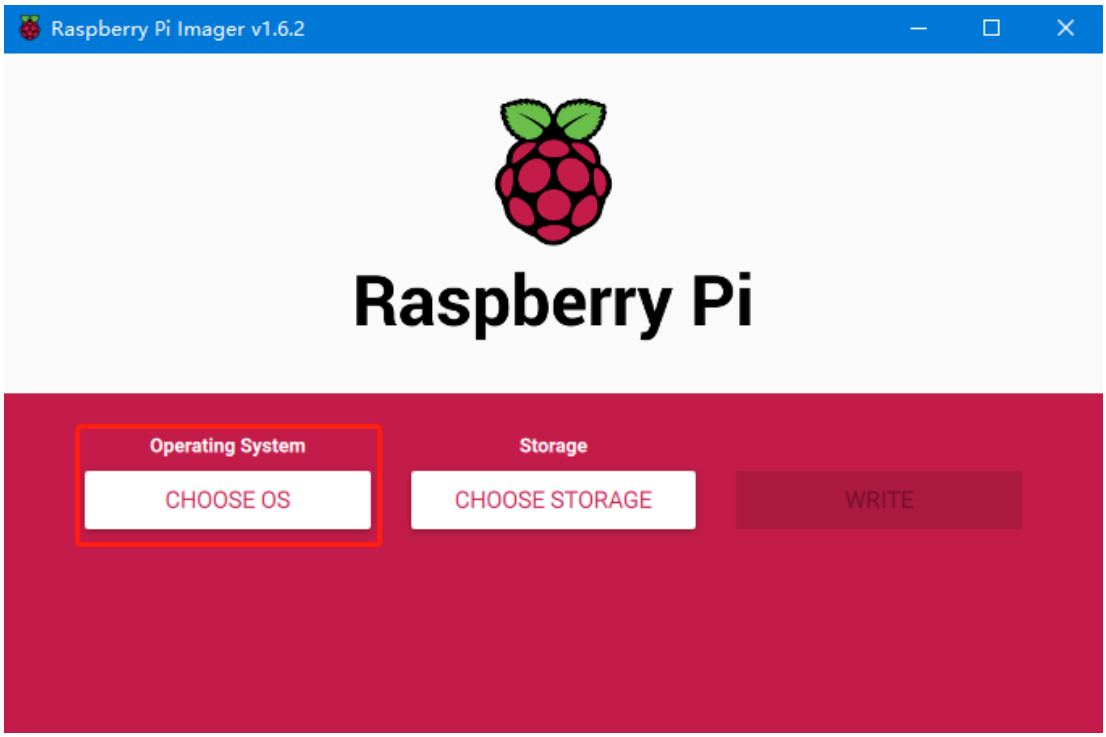
Or refer to Klipper official installation guide using Octoprint.

Download and Install Raspberry Pi Imager

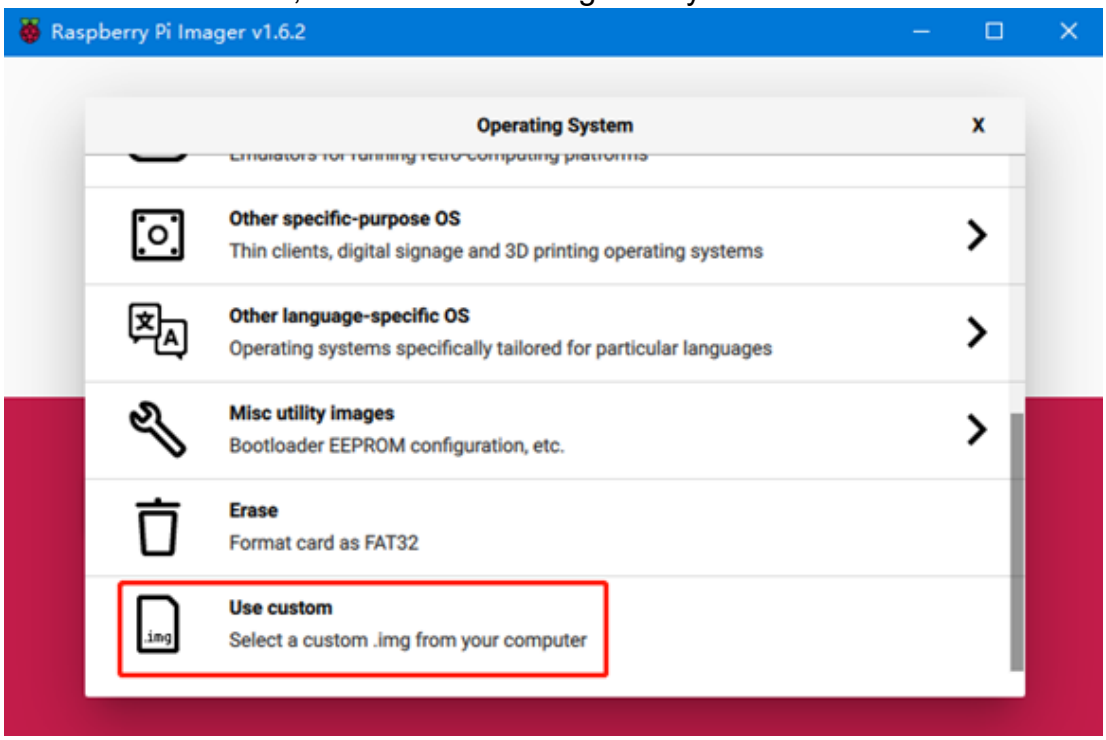
Install the official Raspberry Pi Imager https://www.raspberrypi.com/software/

Write Image

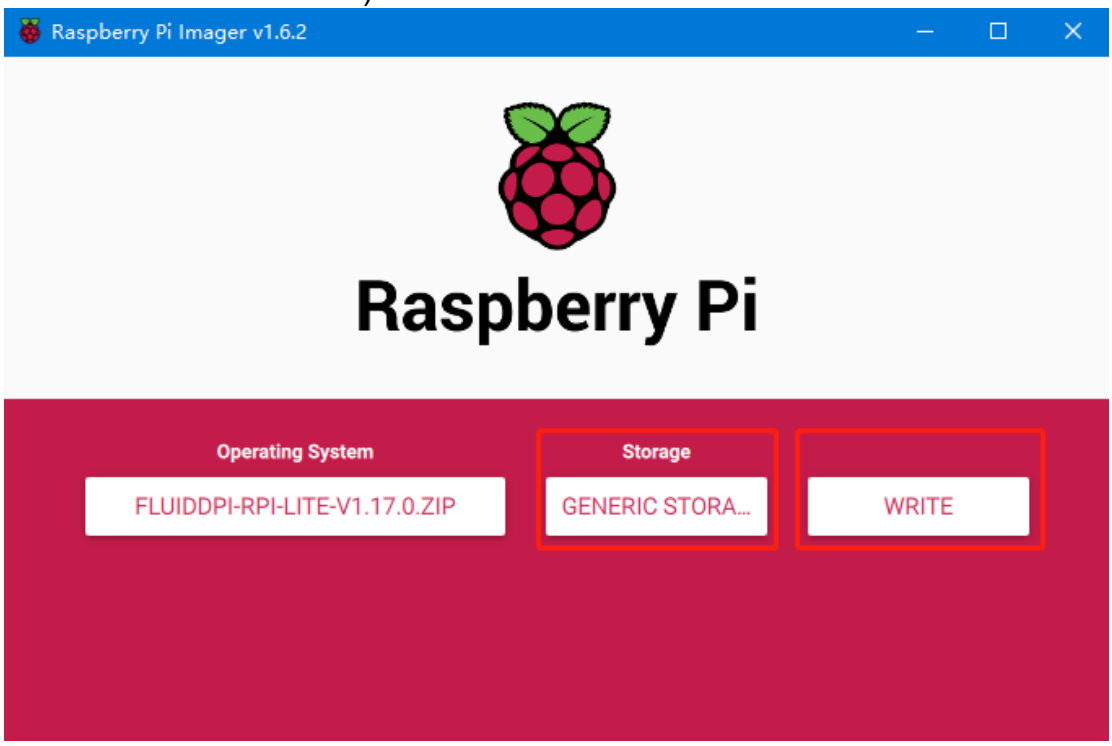
1. Insert microSD into your computer via a card reader.
2. Choose OS.



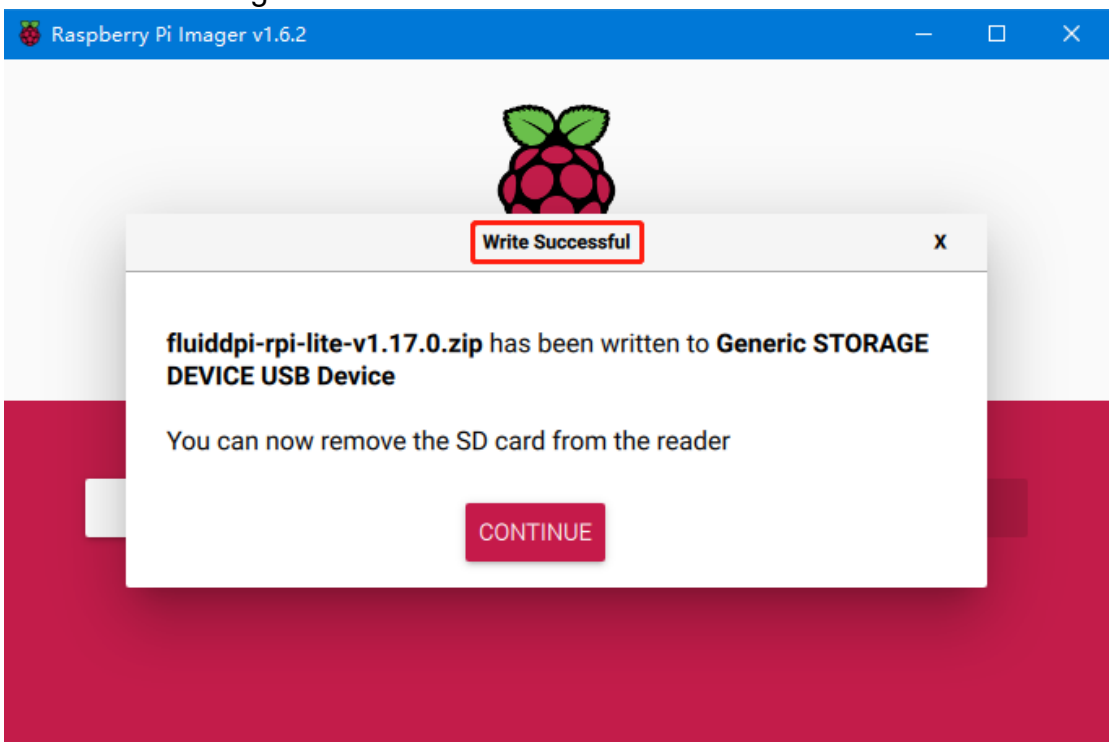
3. Select "Use custom", then select the image that you downloaded.



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



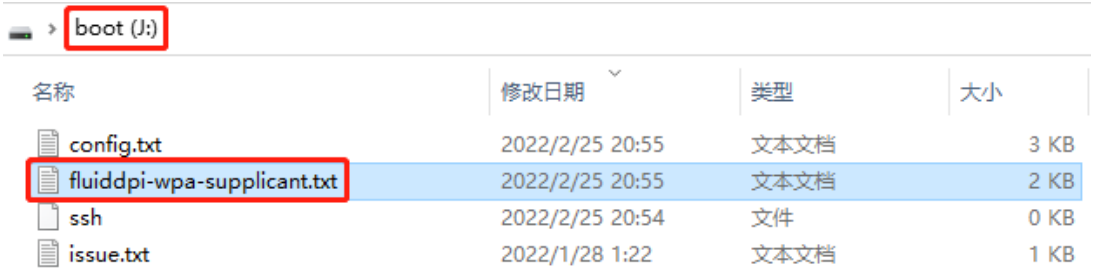
5. Wait for the writing to finish.



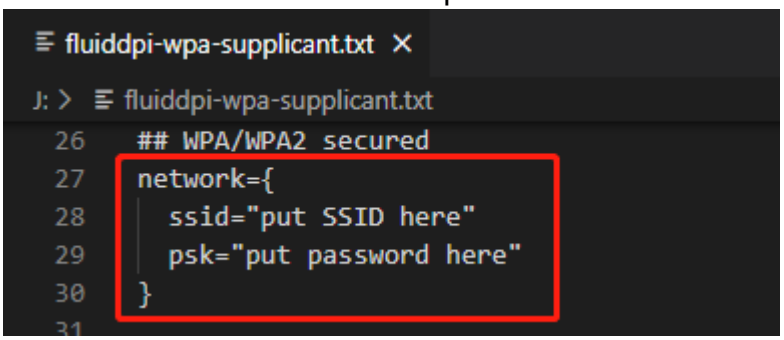
WiFi Setting

Note: this step can be skipped if you are using a network cable connection.

1. Reinsert the card reader.
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)



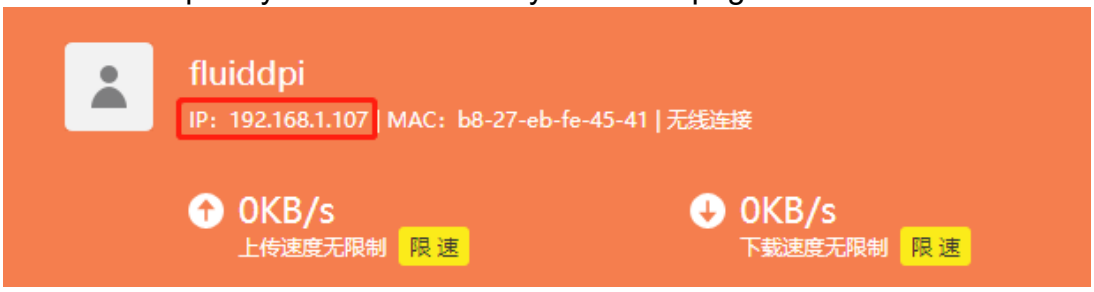
3. Remove the '#' character at the beginning of the four lines in the red box, then set the correct WIFI name and password and save.



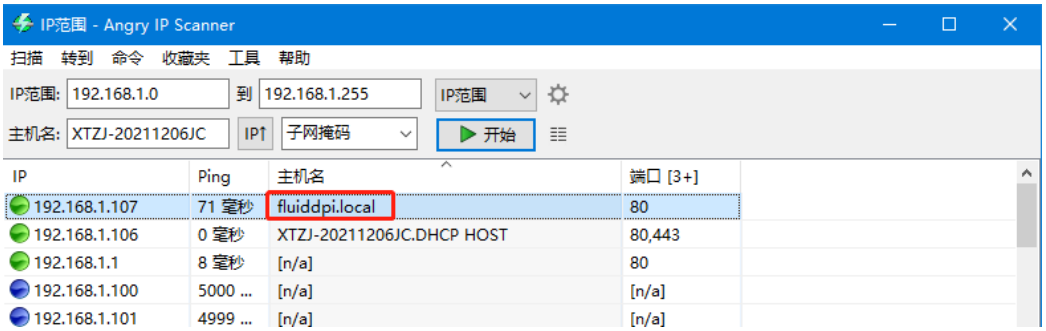
SSH Connect to Raspberry Pi

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 on, approx. 1-2min.
3. The Raspberry Pi will automatically be assigned an IP address after being successfully connected to the network.

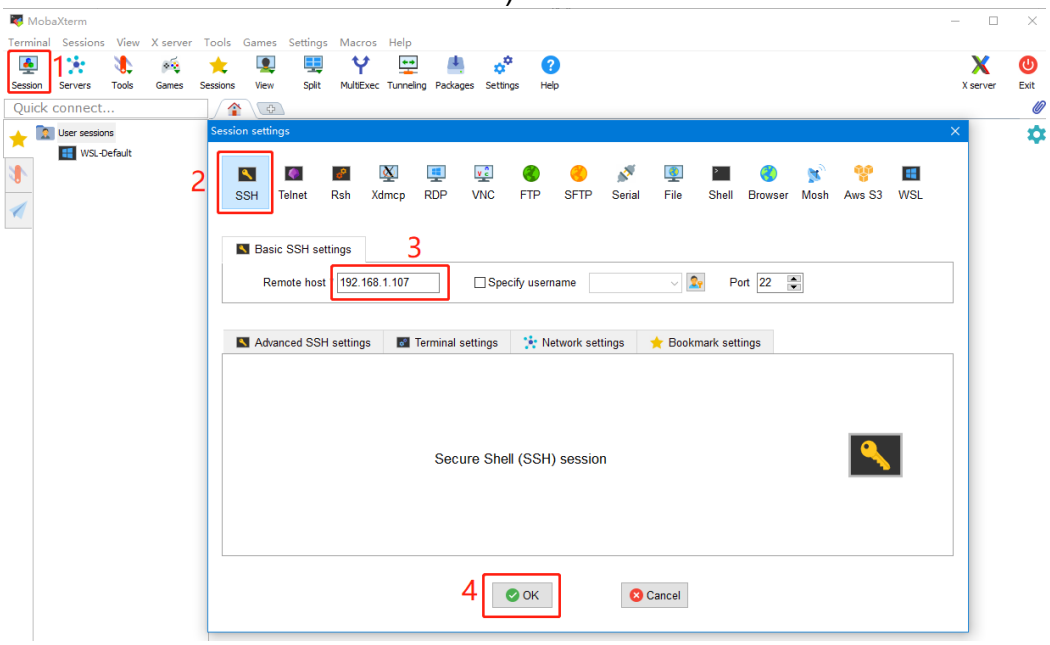
4. Find the Raspberry Pi IP address on your router page.



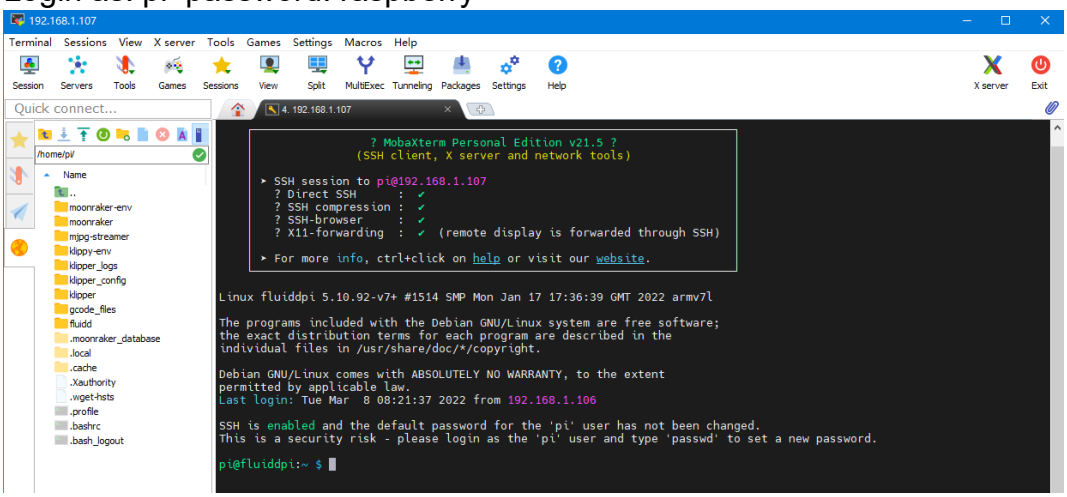
5. Or use the https://angryip.org/ tool, scan all IP addresses in the current network organize by names, and find the IP named Fluidd or Mailsail, as shown below.



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).

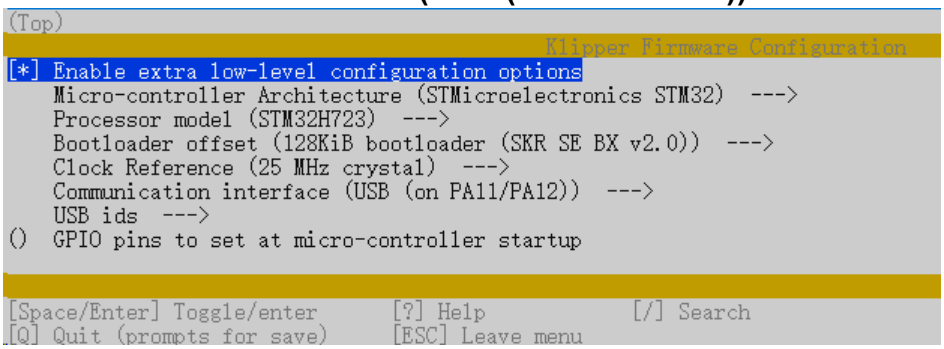


7. Login as: pi password: raspberry



Compile Firmware

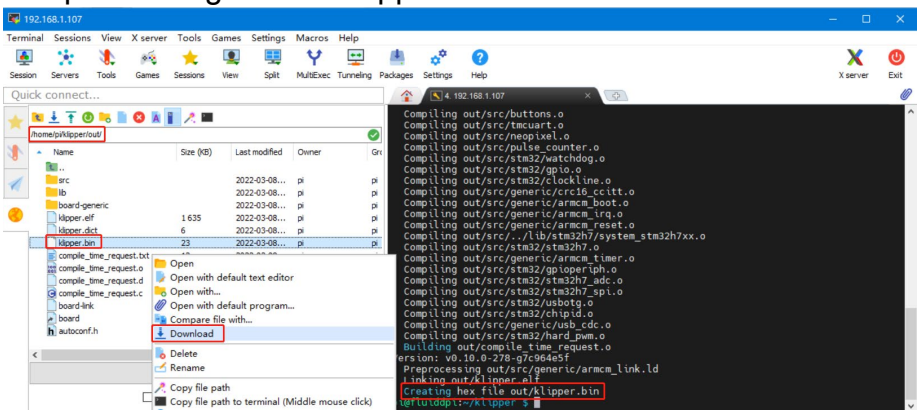
1. After SSH successfully connected to the Raspberry Pi, enter in terminal：
   cd ~/klipper/
   make menuconfig
   Compile with the configuration shown below (if the options below are not available, please update your Klipper source code to the newest version).
   *** [*] Enable extra low-level configuration options**
   *** Micro-controller Architecture (STMicroelectronics STM32) --->**
   *** Processor model (STM32H723) --->**
   *** Bootloader offset (128KiB bootloader (SKR SE BX v2.0)) --->**
   *** Clock Reference (25 MHz crystal) --->**
   *** Communication interface (USB (on PA11/PA12)) --->**



2. Press **q** to exit, and **Yes** when asked to save the configuration.

3. Run **make** to compile firmware, "klipper.bin" file will be generated in **home/pi/klipper/out** folder when **make** is finished, download it onto your

computer using the SSH application.



4. Rename klipper.bin to "firmware.bin", copy to SD card to update firmware.

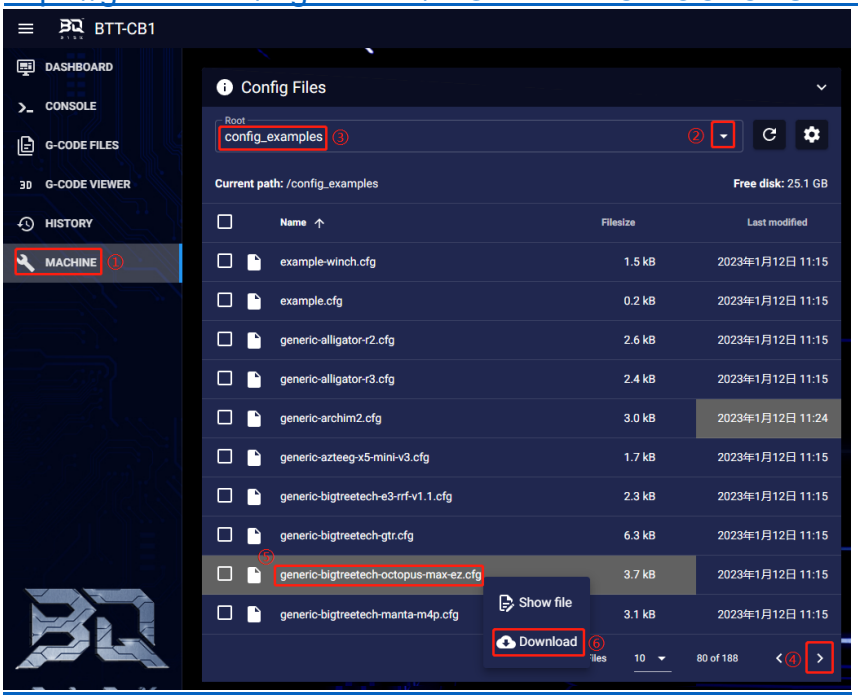
5. Enter: **ls /dev/serial/by-id/** in command line to check motherboard ID to confirm whether firmware is updated successfully, as shown below.

```
pi@fluiddpi:~/klipper $ ls /dev/serial/by-id/
usb-Klipper_stm32h723xx_41003D001751303232383230-if00
pi@fluiddpi:~/klipper $
```

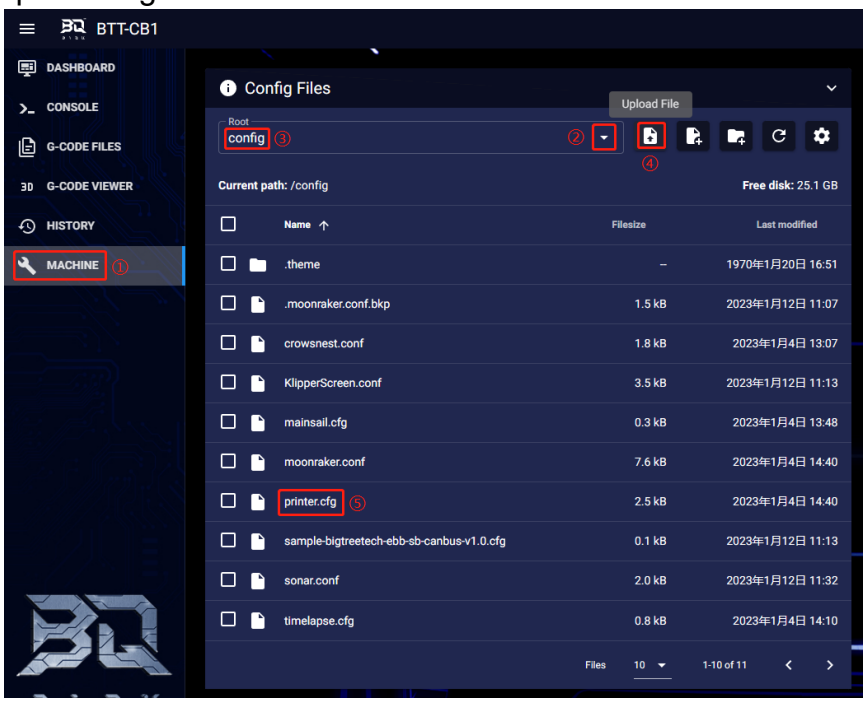
copy and save this ID, it is needed when modifying klipper config.

Configure Klipper

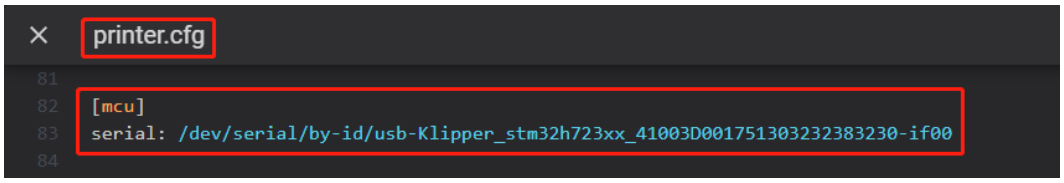
1. Enter your Raspberry Pi IP address into your browser to open the webUI, find the reference config for motherboard in the directory shown below, if there is no such config available, update your Klipper source code to the newest version or download from GitHub: https://github.com/bigtreetech/BIGTREETECH-OCTOPUS-Max-EZ



2. Upload your finished config file into Configuration Files, and rename it 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.

Firmware Updates

Updating via microSD

1. Ensure the microSD card is formatted as FAT32.
2. Rename the compiled firmware or the firmware downloaded from GitHub to "firmware.bin" (note: make sure the computer system's extension settings are clear, as some users hide the extension, and "firmware.bin" actually displays as "firmware").
3. Copy "firmware.bin" to the root directory of the microSD card.
4. Insert the microSD card into the motherboard's slot, power on the motherboard, and the bootloader will automatically update the firmware.
5. The status LED will blink during update.
6. When it stops and the file is renamed "FIRMWARE.CUR", the update is complete.

Updating Klipper via DFU

1. Run `ls /dev/serial/by-id/` to get the board ID. If Klipper is running, it will return a klipper ID.

2. With the ID, enter: `cd ~klipper`
`make flash FLASH_DEVICE= /dev/serial/by-id/usb-Klipper_stm32h712xx_41003D001751303232383230-if00`

to flash the firmware (note: replace `/dev/serial/by-id/xxx` with the actual ID found in the previous step).

```
biqu@Hurakan:~/klipper$ make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
  Building hid-flash
/bin/sh: 1: pkg-config: not found
    hid-flash requires libusb-1.0, please install with:
    sudo apt-get install libusb-1.0
  Flashing out/klipper.bin to /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
Entering bootloader on /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
Device reconnect on /sys/devices/platform/soc/5200000.usb/usb1/1-1/1-1.1/1-1.1:1.0
sudo dfu-util -p 1-1.1 -R -a 0 -s 0x8002000:leave -D out/klipper.bin

dfu-util 0.9

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release!!!
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash  "
Downloading to address = 0x08002000, size = 25264
Download        [=========================] 100%        25264 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status

Failed to flash to /dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00: Error running dfu-util

If the device is already in bootloader mode it can be flashed with the
following command:
  make flash FLASH_DEVICE=0483:df11
  OR
  make flash FLASH_DEVICE=1209:beba

If attempting to flash via 3.3V serial, then use:
  make serialflash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_190028000D50415833323520-if00
```

Ignore the dfu-util errors after successful flash.

Precautions

- All plugging and unplugging operations should be performed with the power off.
- When using a fan, make sure the voltage selection matches the fan's working voltage to prevent damage to the fan.

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.