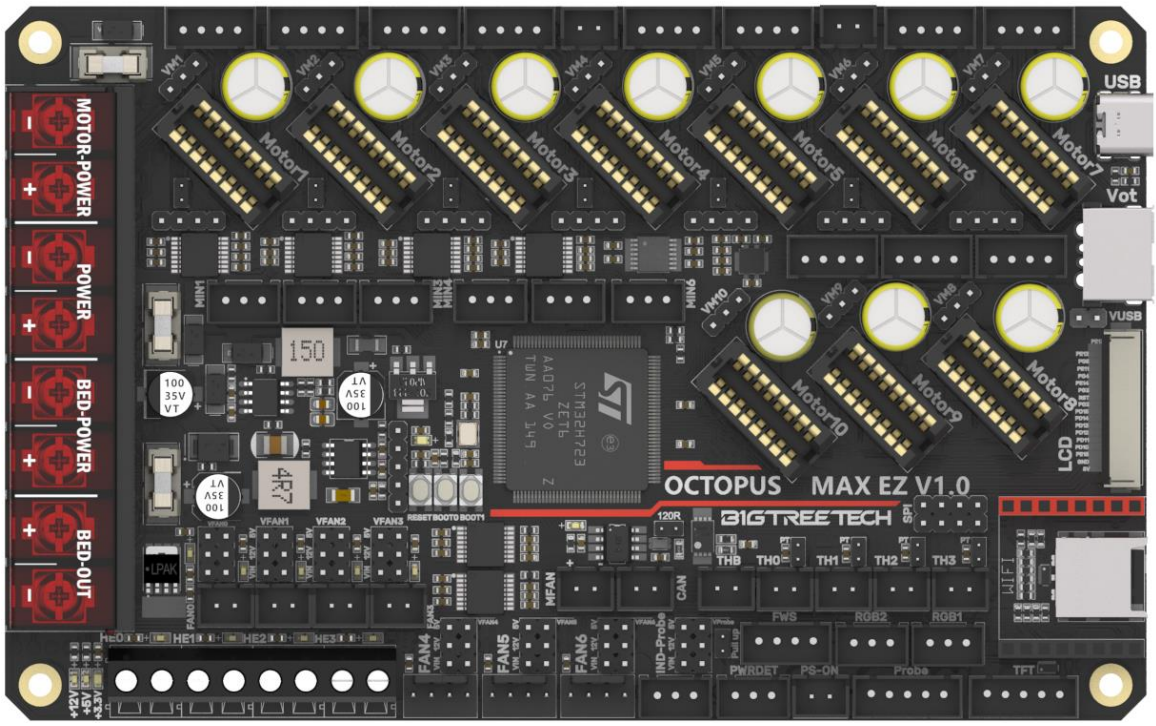
BIGTREETECH Octopus MAX EZ V1.0

使用说明



目录

修订历史

版本	修改说明	日期
01.00	初稿	2022/10/06

一、产品简介

BIGTREETECH Octopus MAX EZ 主板是深圳市必趣科技有限公司 3D 打印团队针对 Octopus Pro 优化升级的 32 位打印机主板，采用自主研发的步进电机驱动座子，增强安全性及用户体验，新增一系列 Octopus Pro 不具有的功能，加大 DIY 可能性。

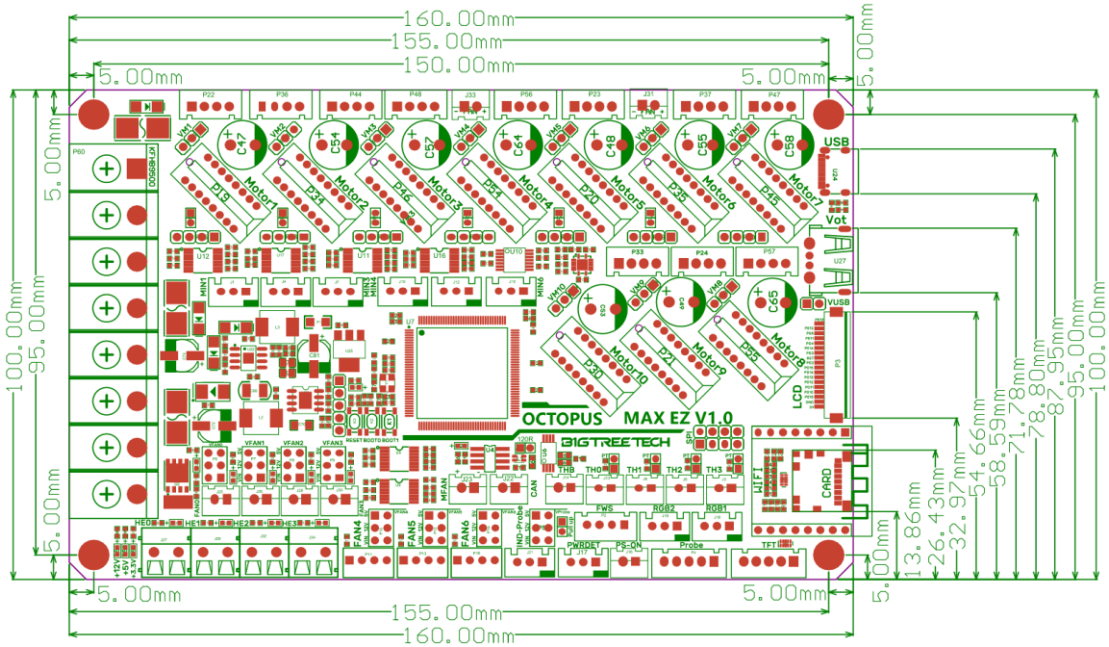
1.1 产品特点

1. 采用 32 位主频 550MHz 的 ARM Cortex-M7 系列 STM32H723ZET6 主控芯片；
2. 主板预留 BOOT 按键，用户可以通过 DFU 方式更新主板引导程序；
3. 增加热敏电阻部分的保护电路，避免因热床或者加热棒漏电导致主控芯片烧毁；
4. 数控风扇 24V、12V、5V 电压可选，省去客户外接变压模块的操作，从而减少主板损坏几率；
5. 增加 E-FUSE 保护，反应速度快，保护力强，有效避免因为短路、过流、打火等情况造成主板烧毁；
6. 可通过 SD 卡升级 MCU 固件，也可通过 Klipper 的 make flash 命令通过 DFU 更新 MCU 固件；
7. 使用十轴 EZ 驱动插座，避免手指被针扎破；板载 TMC 驱动的 SPI 和 UART 工作模式，只需通过固件设置即可使用；
8. 支持断料检测、断电续打、CAN、打完关机、BLTouch、RGB 灯等功能；
9. 采用可更换的保险丝，方便更换；
10. 预留三路四线风扇接口，且可用于连接水冷装置；
11. 预留接近开关接口，支持 NPN 和 PNP 型选择，（24V，12V，5V）电压可选；
12. 预留 SPI 拓展接口，供使用 Klipper 固件的客户外接加速度传感器来进行加速度补偿。

1.2 产品参数

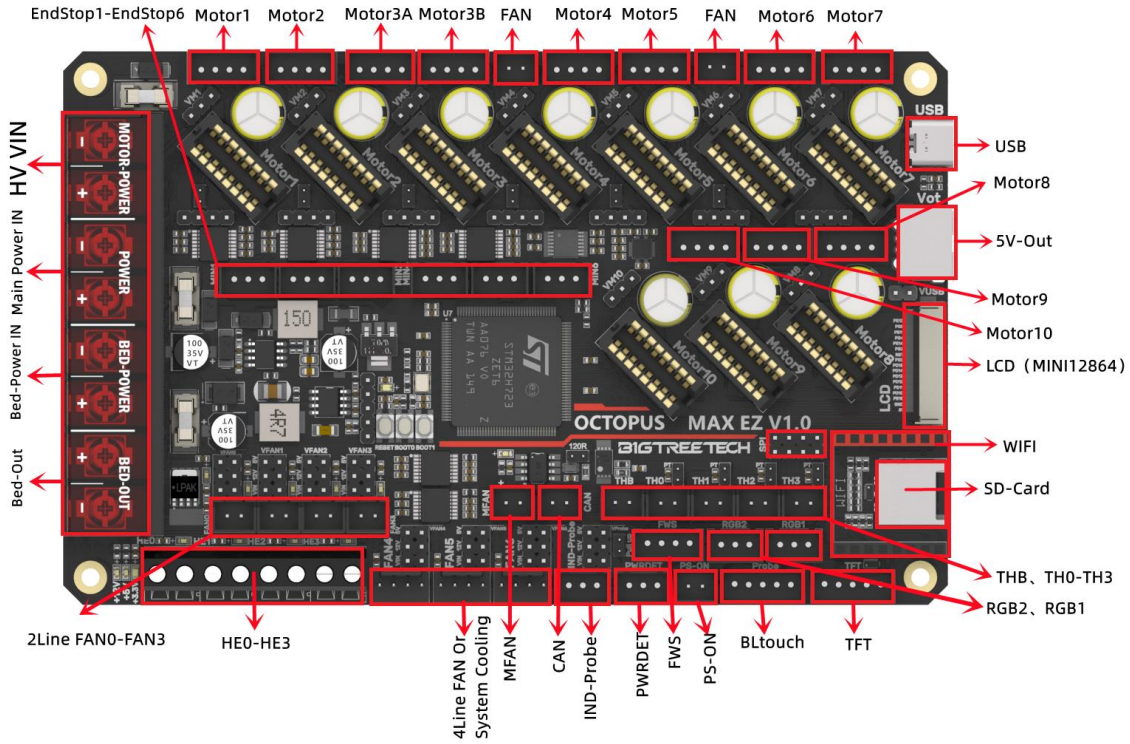
1. 外观尺寸：160mm*100mm 详情请参考：BIGTREETECH Octopus MAX EZ V1.0-SIZE.pdf
2. 安装尺寸：见 BIGTREETECH Octopus MAX EZ V1.0-SIZE.pdf
3. 微处理器：ARM Cortex-M7 STM32H723ZET6 550MHz
4. 驱动输入电压：24V，HV(≤56V)可选
5. 主板输入电压：VIN=DC12V 或 DC24V
6. 热床输入电压：BED IN=DC12V 或 DC24V
7. 逻辑电压： DC3.3V
8. 加热接口：热床（HB）、加热棒（HE0,HE1,HE2,HE3）
9. 热床端口最大输出电流：10A，峰值 12A
10. 加热棒端口最大输出电流：5.5A，峰值 6A
11. 风扇接口：两线数控风扇（FAN0，FAN1，FAN2，FAN3），四线数控风扇（FAN4，FAN5，FAN6），常开风扇（24V FAN*2），其中数控风扇和 MFAN 电压（5V，12V，24V）可选；
12. 风扇接口最大输出电流：1A，峰值 1.5A
13. 加热棒 + 驱动 + 风扇的总电流：小于 12A
14. 拓展接口：BLTouch（Servos、Probe）、PS-ON、FWS、PWRDET、RGB*2、SPI、IND-Probe、CAN、WIFI、TFT
15. 电机驱动：支持 EZ5160、EZ2209、EZ2225、EZ2226、EZ2208、EZ2130 等
16. 驱动工作模式支持：SPI、UART
17. 电机驱动接口： Motor1，Motor2，Motor3（双电机接口），Motor4，Motor5，Motor6，Motor7，Motor8，Motor9，Motor10 总共 10 路
18. 温度传感器接口：5 路 100K NTC，其中 4 路为 NTC 与 PT1000 可选端口
19. 显示屏： MINI12864（FPC 连接）、TFT 串口屏
20. PC 通信接口：Type-C，方便插拔
21. 支持机器结构：Cartesian、Delta、Kossel、Ultimaker、CoreXY
22. 推荐软件：Cura、Simplify3D、Pronterface、Repetier-host、Makerware

1.3 产品尺寸

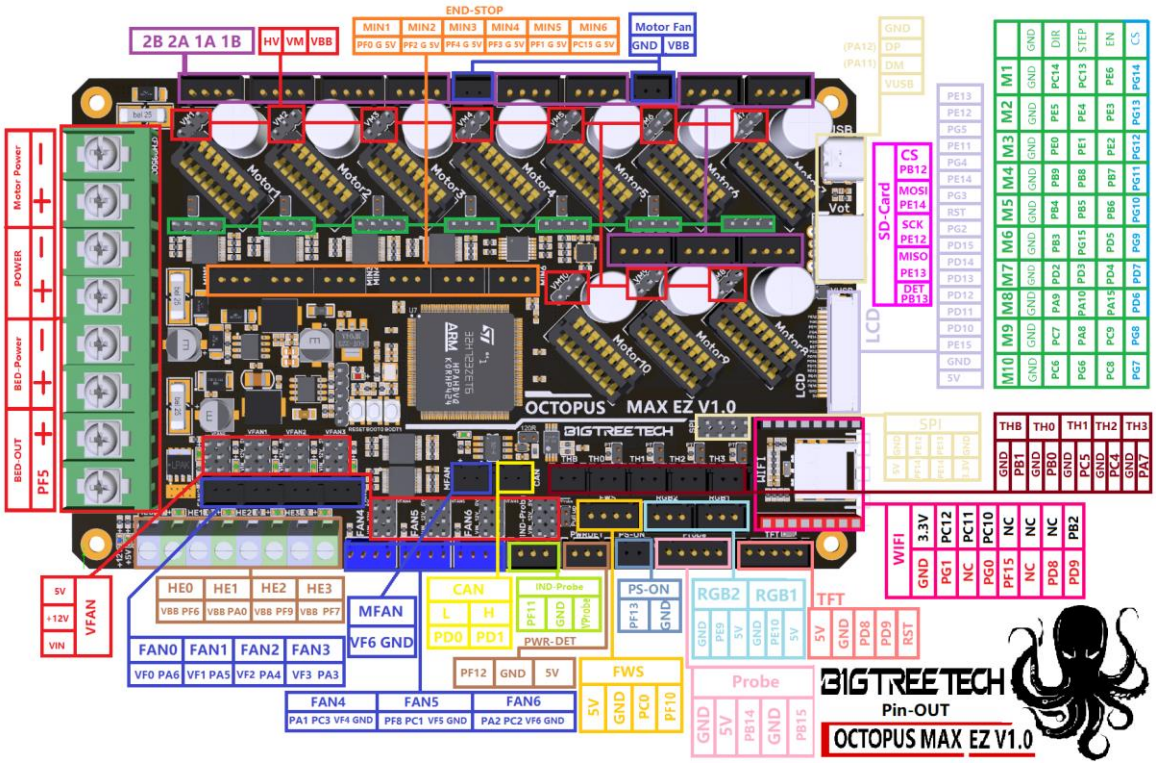


二、外设接口

2.1 接口示意图



2.2 Pin 脚说明



三、接口介绍

3.1 USB 供电

M8P 主板上电之后，MCU 左侧的 D32 红灯会亮起，表示供电正常。板子中部的 VUSB 是电源选择端，仅当使用 USB 给主板供电或需通过 USB 向外供电时，才需要使用跳帽将它短接。

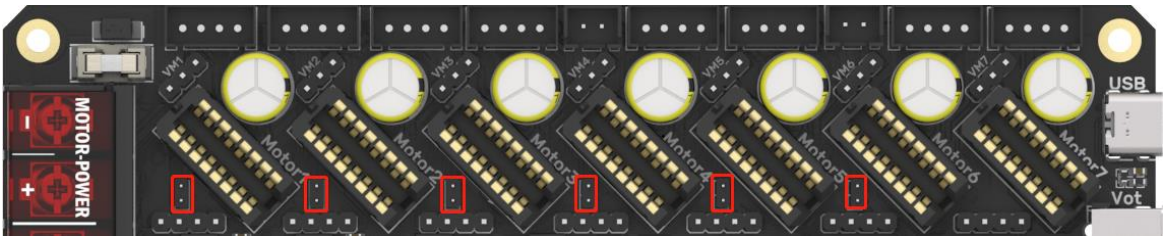
3.2 步进电机驱动

3.2.1 驱动模式支持（UART、SPI）

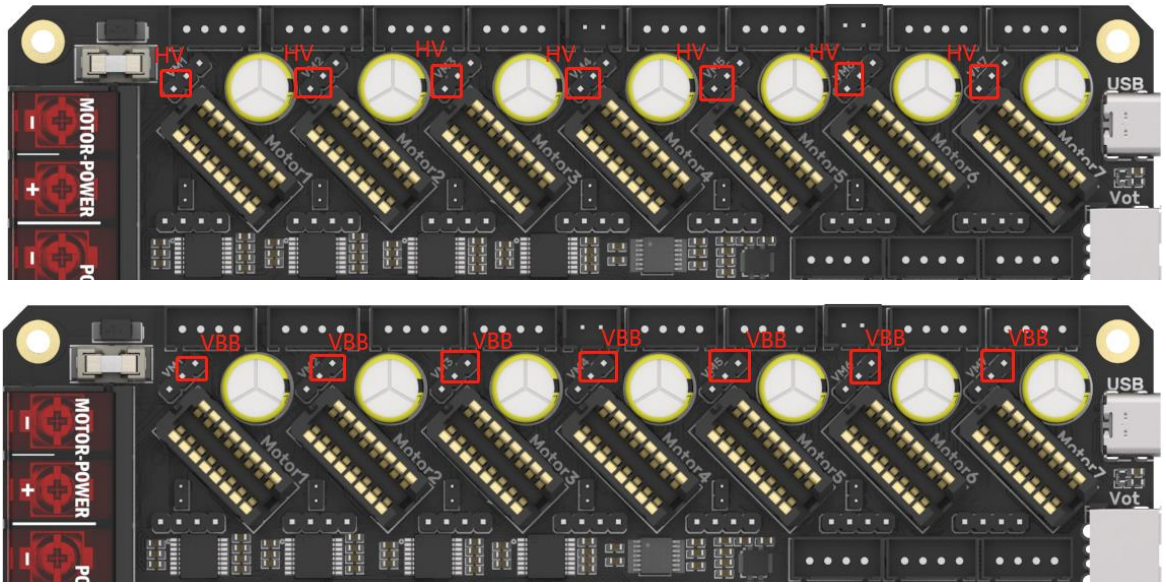
通过固件直接设置，无需手动跳线帽选择。

3.2.4 TMC 驱动的 DIAG(Sensorless Homing)

如图示位置，使用 Sensorless Homing 功能时就插上跳帽，不使用则不插，无需剪断驱动的 DIAG 引脚。（Motor1-Motor6）



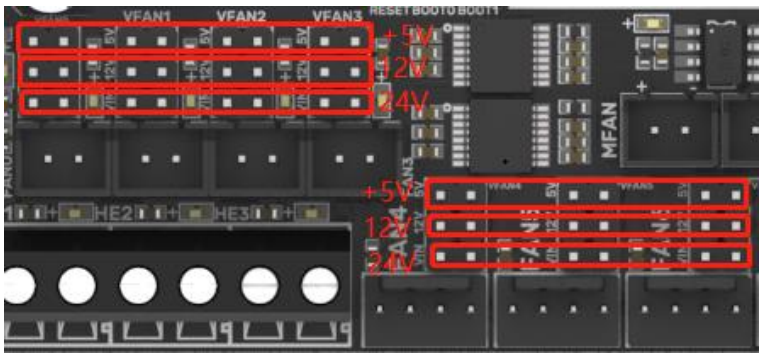
3.2.5 驱动电压选择



3.3 数控风扇的电压选择

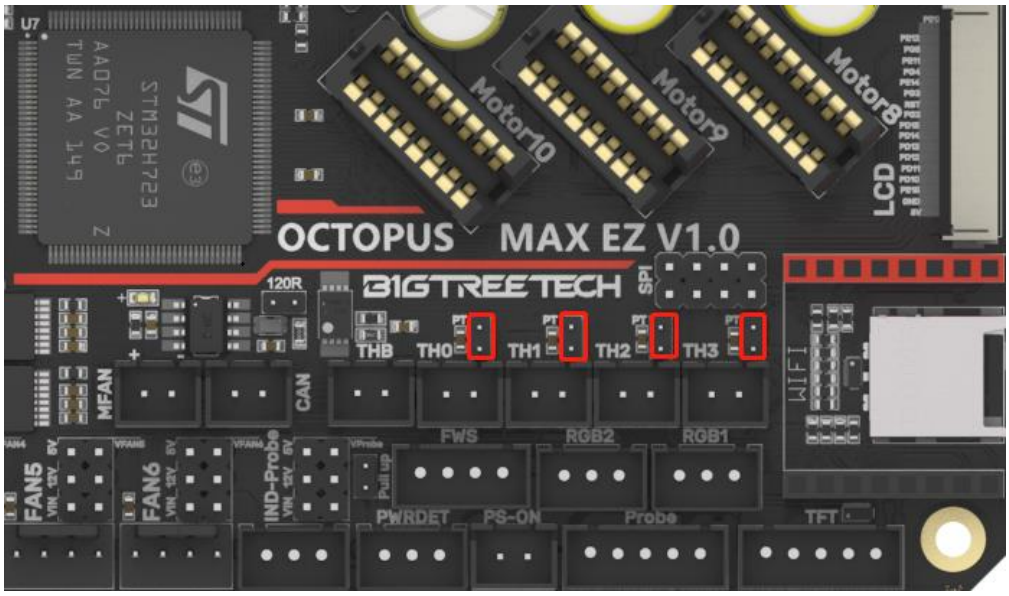
通过跳帽来设置输出电压为 5V，12V 或是 24V。（其中 MFAN 与 FAN6 共用电源 VFAN6）

注意：选择电压前请确认风扇支持的电压为哪一档，因选错导致的风扇烧毁，我司不与承责。

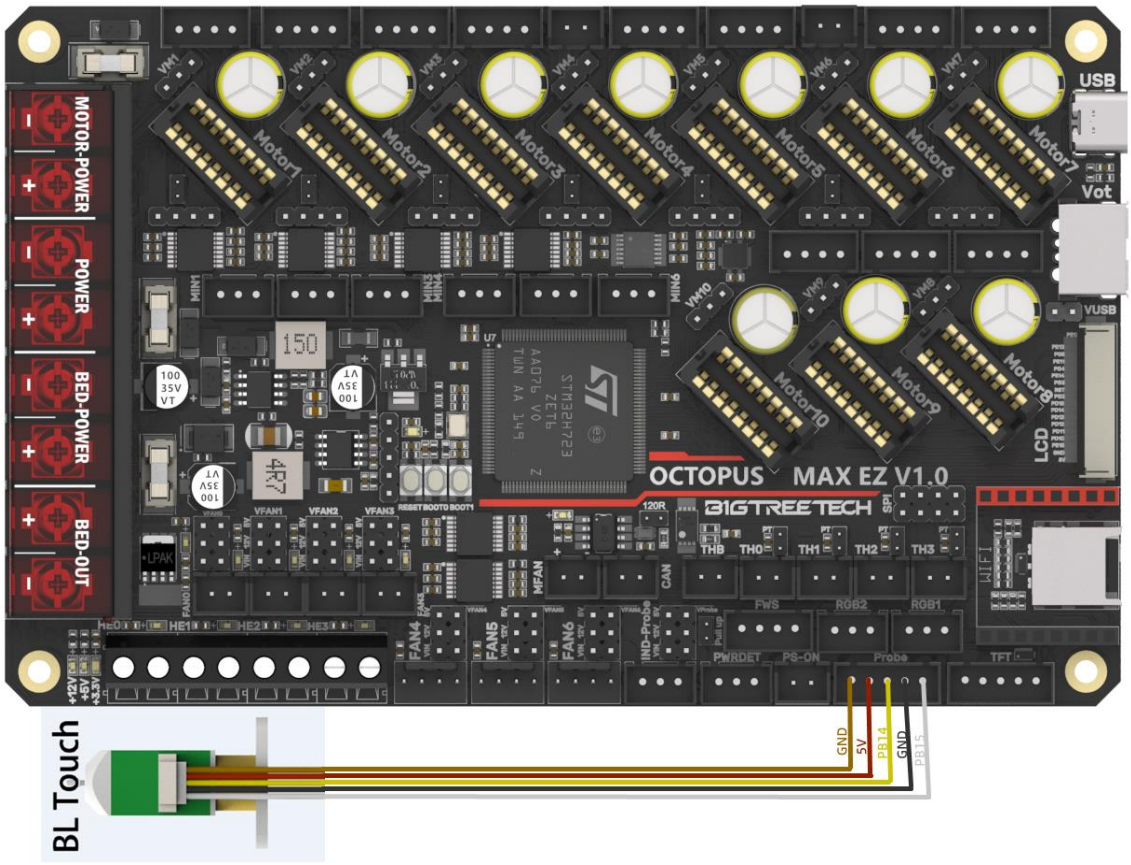


3.4 100K NTC 或 PT1000 设置

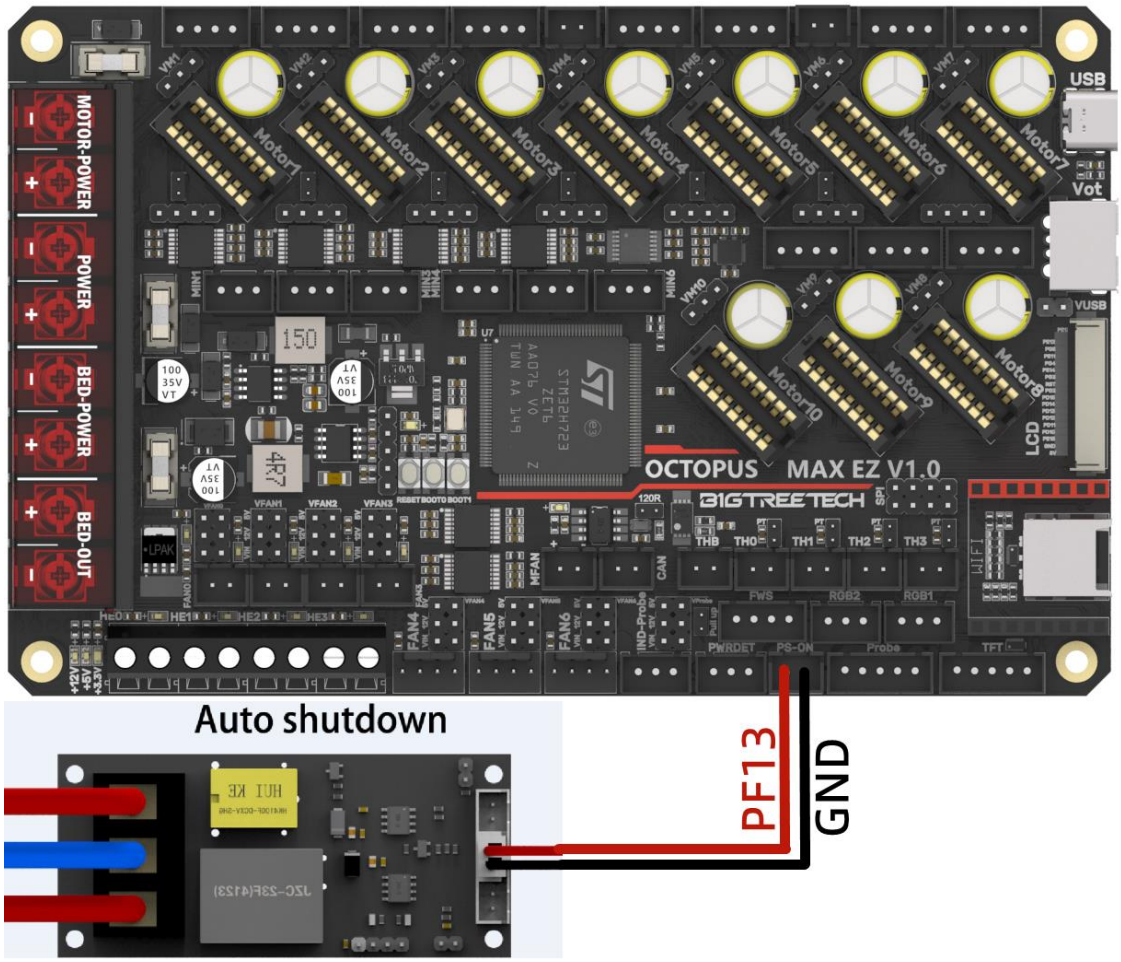
使用 100K NTC 热敏电阻时，无需插入跳线帽，此时 TH0-TH3 的上拉电阻为 4.7K 0.1%。使用 PT1000 时，需使用跳帽短接下图红框中的两 Pin,并联了 4.12K 0.1%电阻，此时 TH0-TH1 的上拉电阻为 2.2K（注意：此种方式读出的温度精度会比 MAX31865 差很多）。



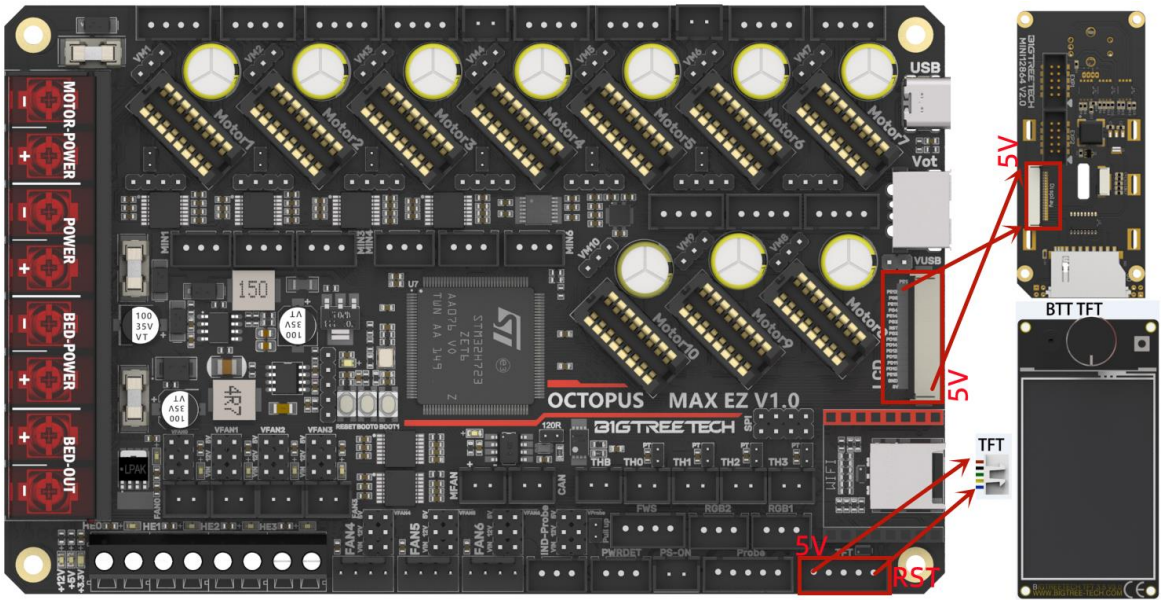
3.5 BLTouch 接线



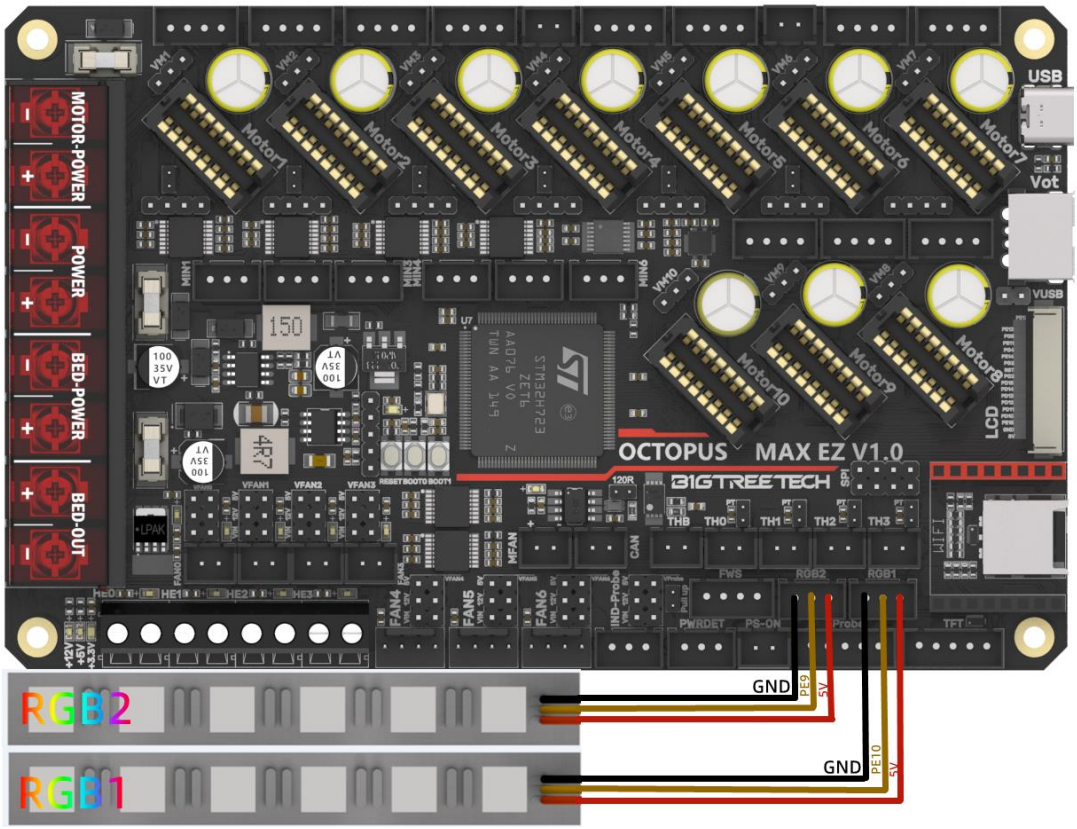
3.6 打完关机模块(Relay V1.2)接线



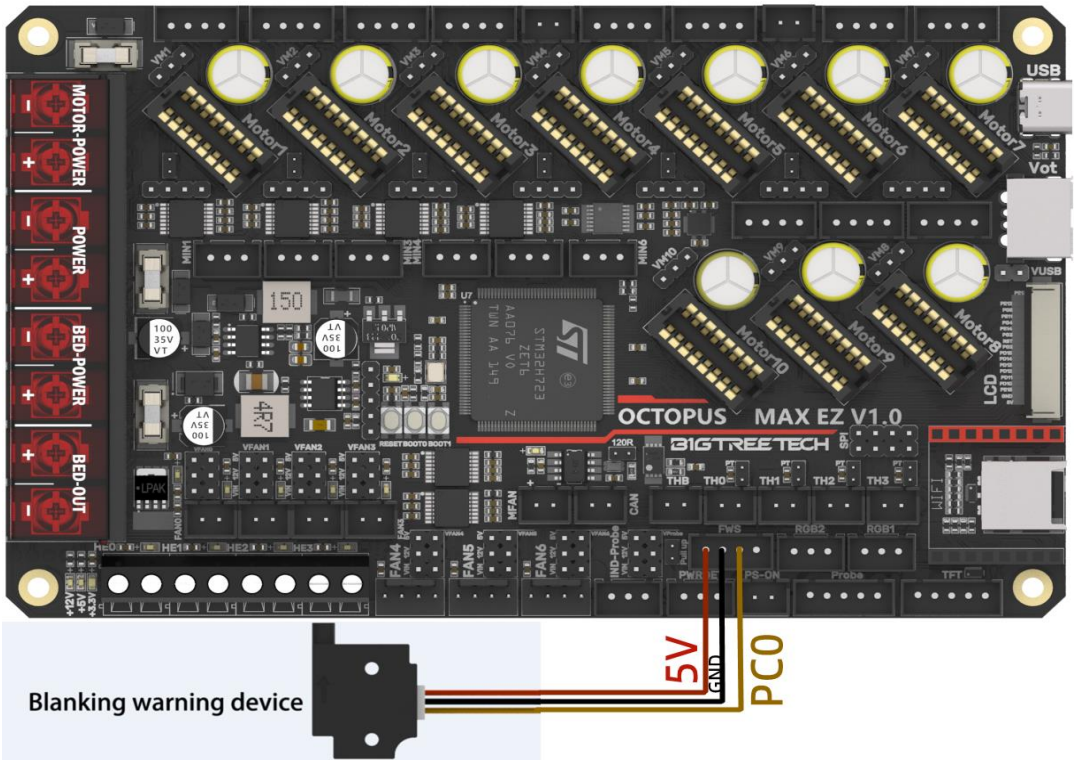
3.7 与 MINI12864 屏和 TFT 屏的接线



3.8 RGB 接线

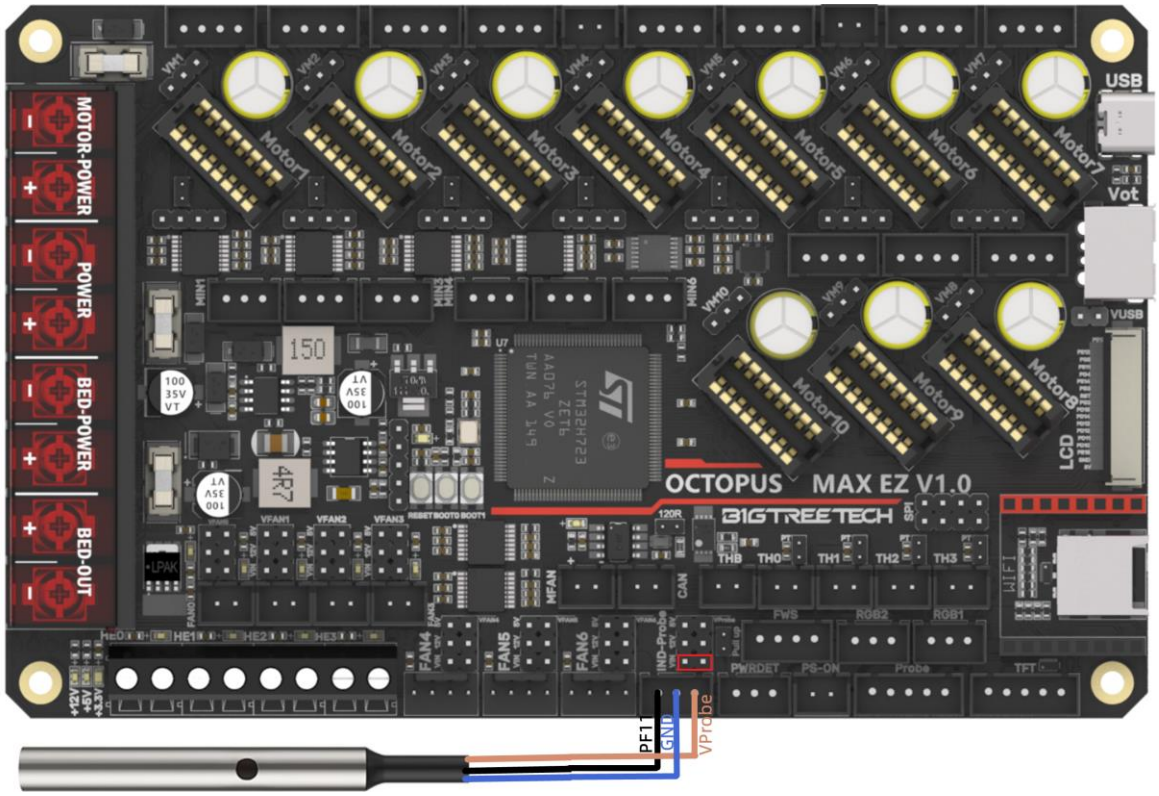


3.9 断料检测接线

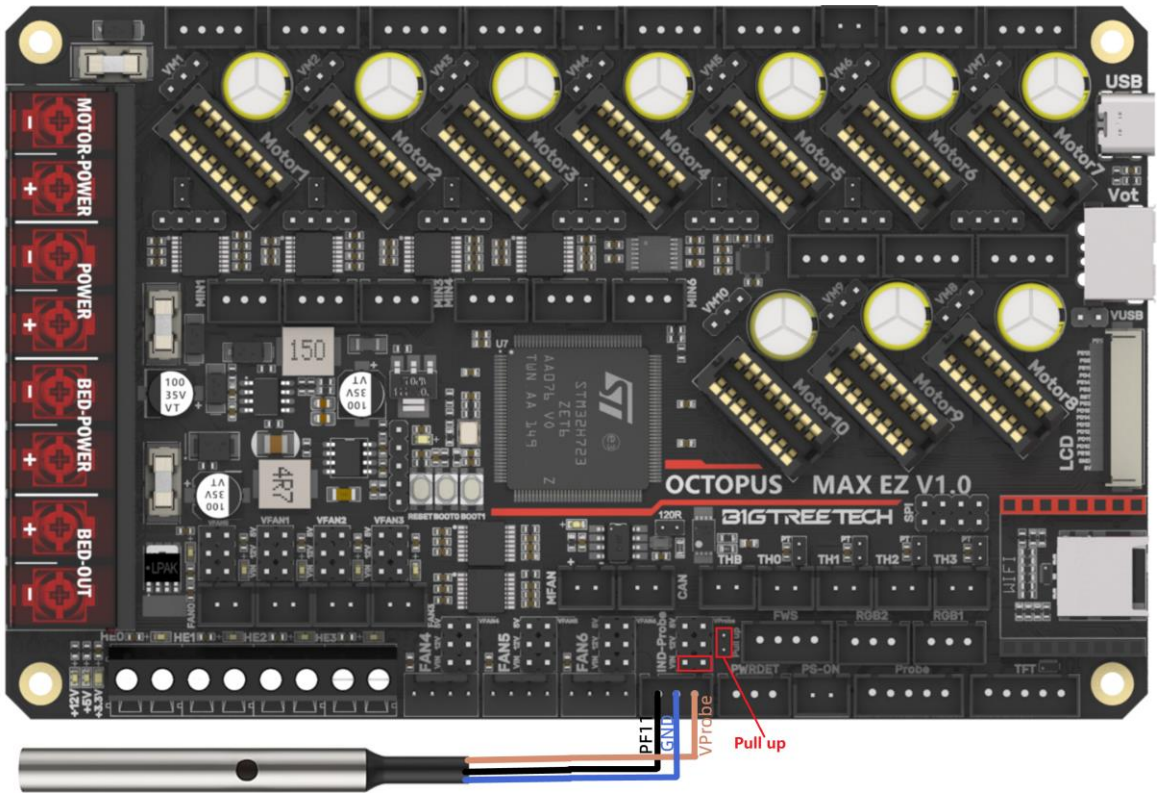


3.10 接近开关的连接

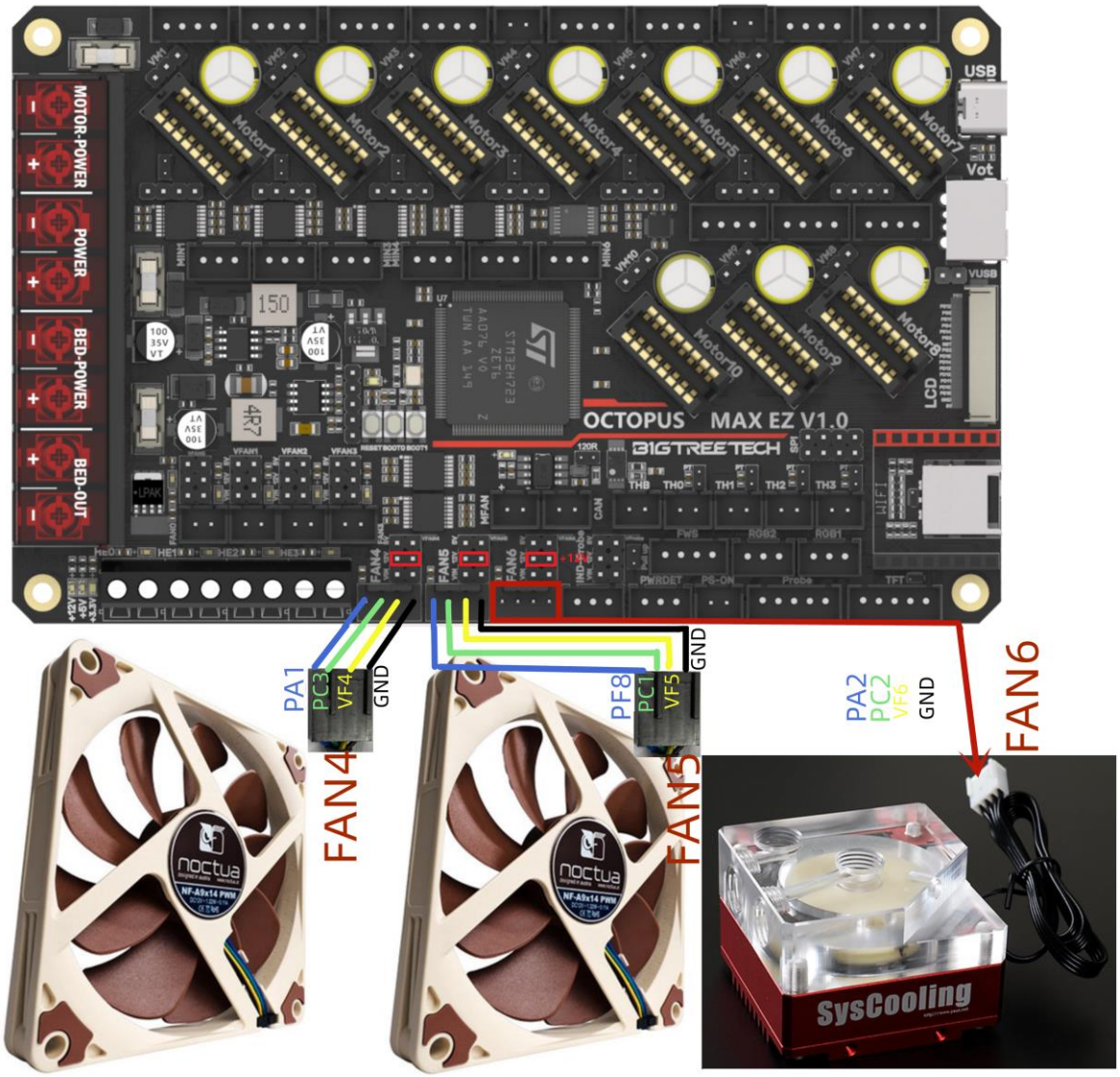
常开（NPN 型），不用通过跳线帽短接，如下图 24V 为例：



常闭（PNP 型），需要将跳线帽短接，如下图 24V 为例：



3.11 四线数控风扇及水冷装置的连接（下图以 12V 为例）



四、Marlin

4.1 安装编译环境

https://github.com/bigtreetech/Document/blob/master/How%20to%20install%20VScode%2BPlatformio.md
https://marlinfw.org/docs/basics/install_platformio_vscode.html
参考这两个链接的说明安装 VSCode 以及 PlatformIO 插件（国内的用户在线安装 PlatformIO 插件可能会很慢）

4.2 下载 Marlin 固件

1. 从 Marlin 官网下载最新版本的 bugfix 版本的固件
   https://github.com/MarlinFirmware/Marlin/tree/bugfix-2.0.x
2. 从我们 github 上下载预先配置好编译环境和主板类型的固件
   https://github.com/bigtreetech/BIGTREETECH-OCTOPUS-Max-EZ

4.3 配置固件

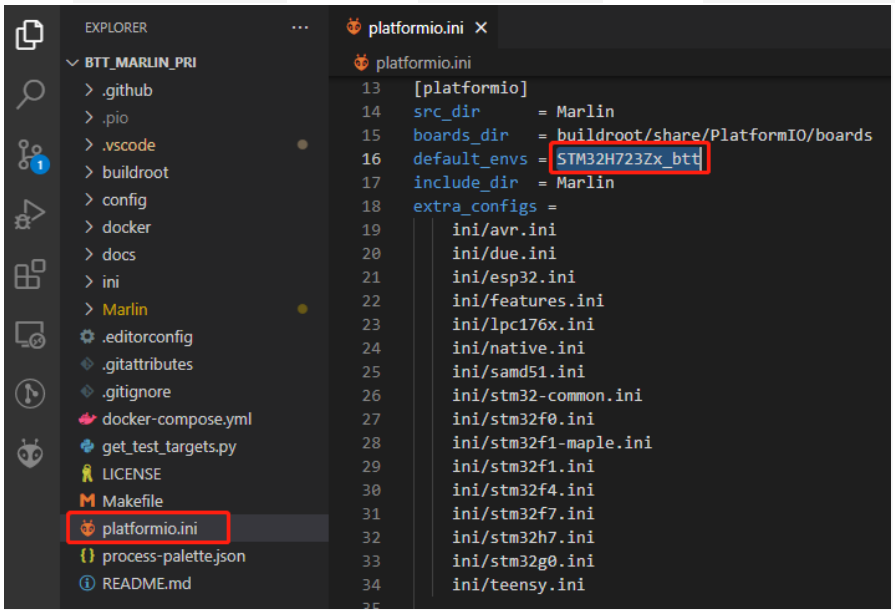
4.3.1 打开 Marlin 工程

您可以通过以下几种方式之一在 VSCode 中打开 Marlin

- 将下载的 Marlin Firmware 文件夹拖到 VSCode 应用程序图标上
- 使用 VSCode `File` 菜单中的 `Open...` 命令
- 打开 PIO Home 选项卡，然后单击“`Open Project`”按钮

4.3.2 配置编译环境

打开 `platformio.ini` 文件并将 `default_envs` 修改为 `STM32H723Zx_btt`

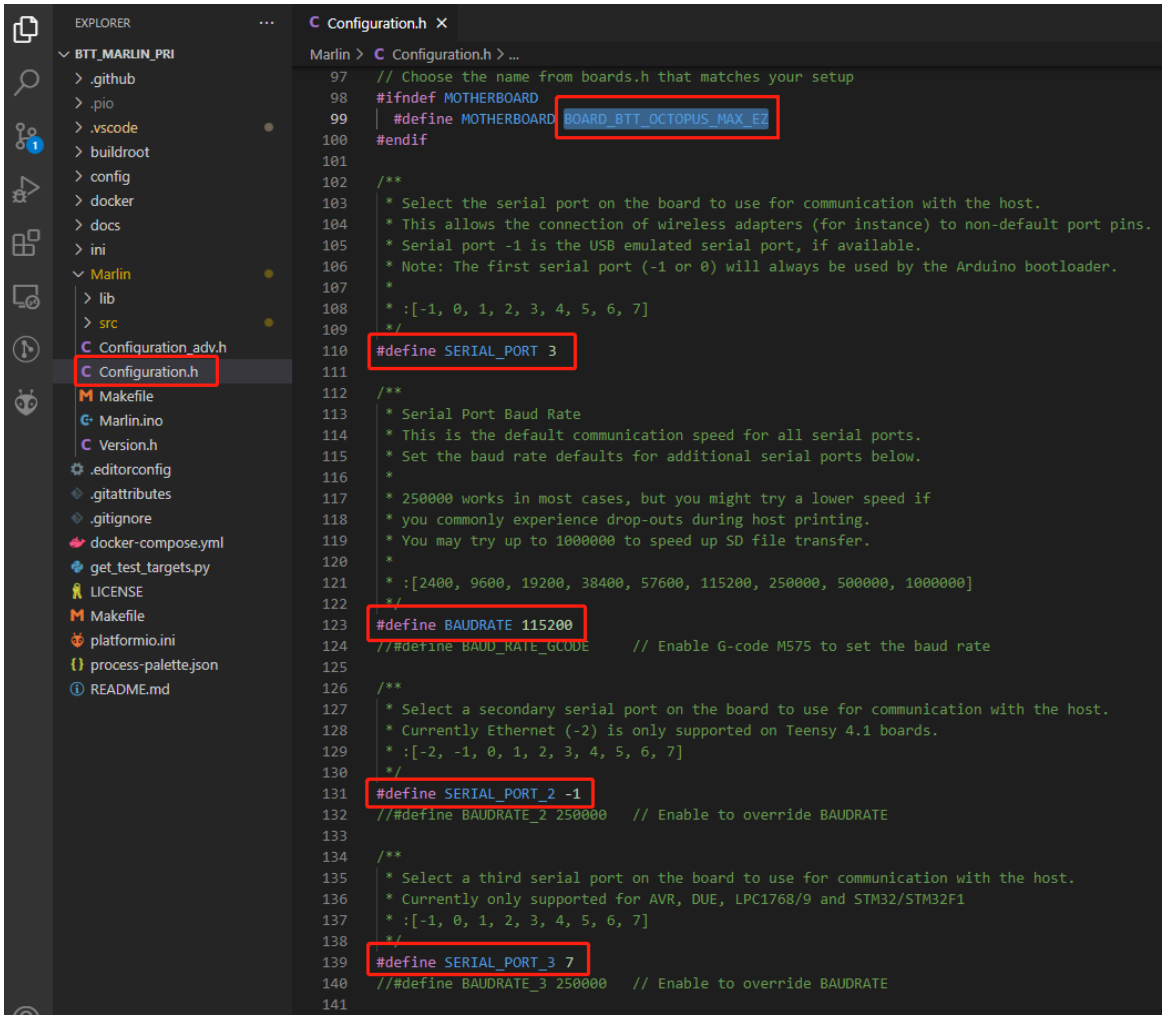


4.3.3 配置主板类型、串口号

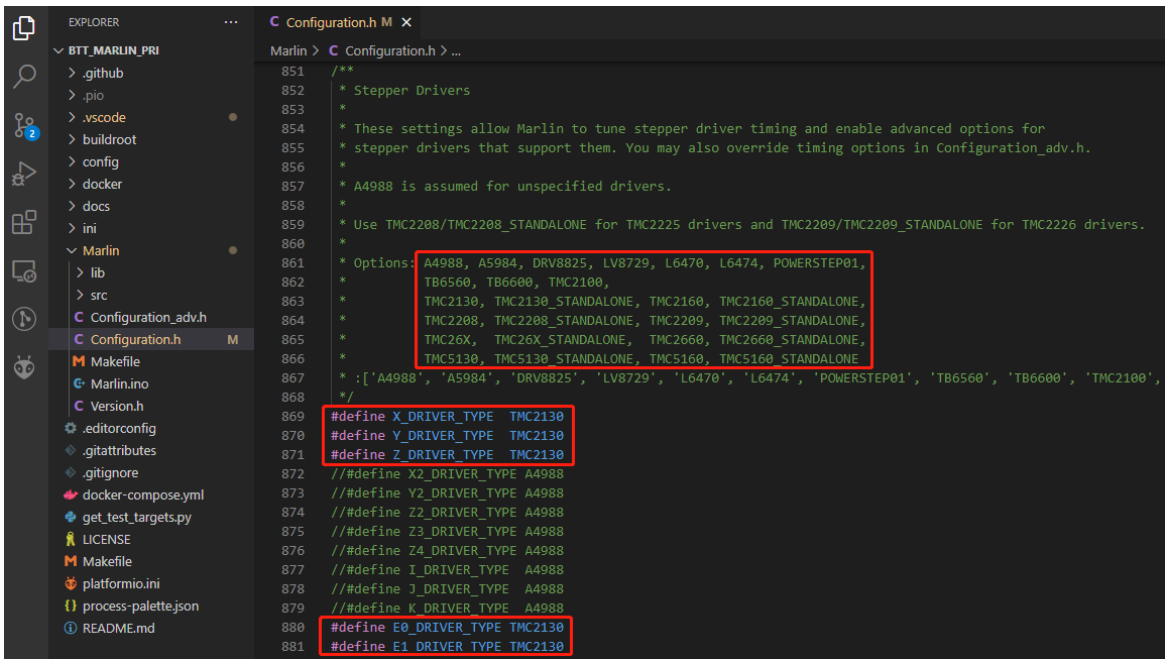
设置主板类型 MOTHERBOARD 为 BOARD_BTT_OCTOPUS_MAX_EZ

```
#define MOTHERBOARD BOARD_BTT_OCTOPUS_MAX_EZ
#define SERIAL_PORT 3        （启用 TFT 串口）
#define BAUDRATE 115200      （设置波特率，注意要跟通信的设备一致）
#define SERIAL_PORT_2 -1     （启用 USB 模拟串口）
#define SERIAL_PORT_3 7      （启用 WIFI 的串口）
```

以上的设置根据需求自行启用

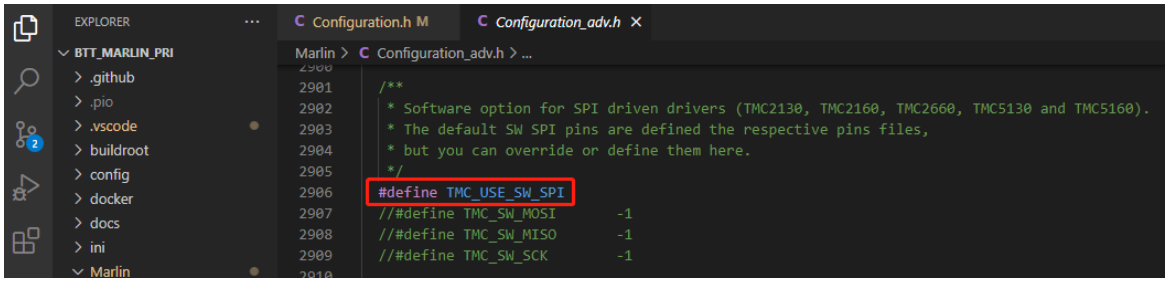


4.3.4 配置电机驱动

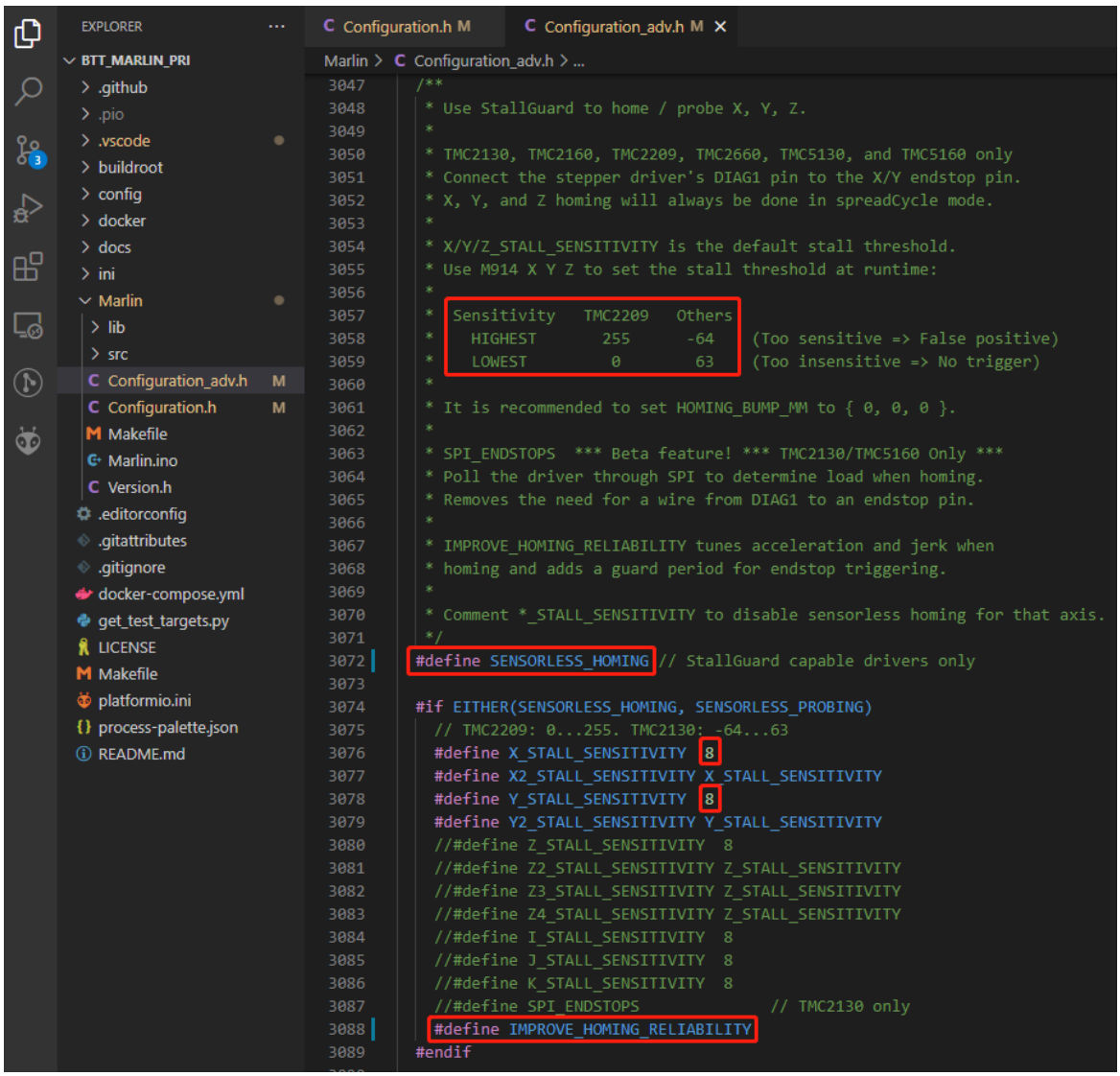


如果使用的驱动为 SPI 模式，还需在 Configuration_adv.h 文件中使能 TMC_USE_SW_SPI

```
#define TMC_USE_SW_SPI
```



4.3.5 Sensorless homing



#define SENSORLESS_HOMING // 打开驱动堵转检测作为 Home 限位开关的功能

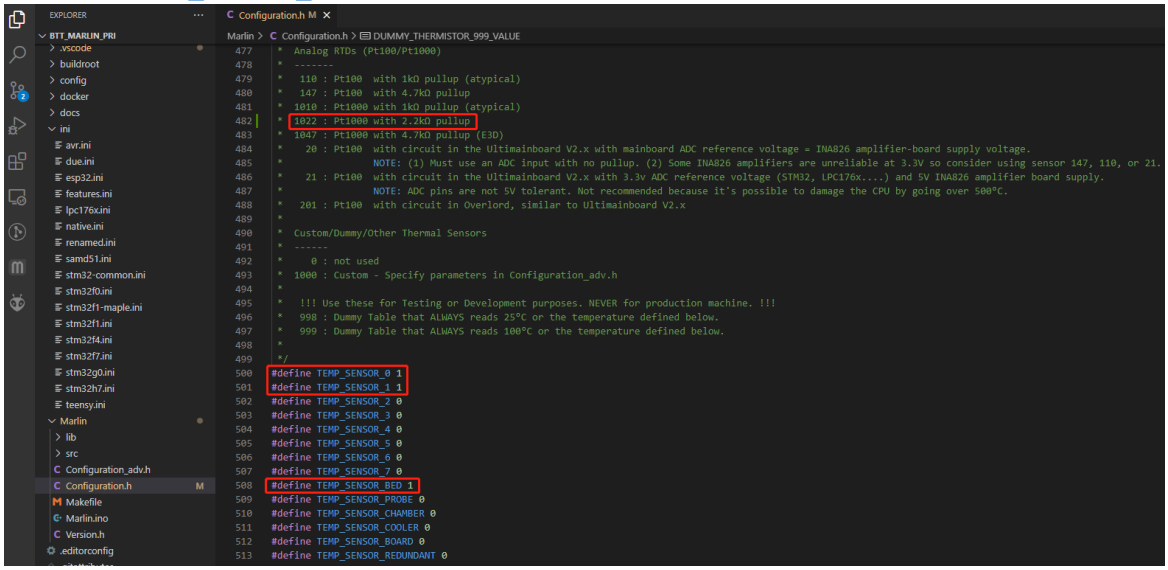
#define xx_STALL_SENSITIVITY 8 // 设置堵转检测的灵敏度。TMC2209 范围为 0~255，数值越大越灵敏，容易误触发，现象为 Home 的时候轴还没有回到原点就停下来了，数值越小越不灵敏容易不触发，现象为归零时一直撞轴发出“噔噔噔”的声音。其他驱动范围为 63~-64，数值越小越灵敏#define IMPROVE_HOMING_RELIABILITY // 可以在上面单独设置归零时的电流参数(X_CURRENT_HOME)，以便得到最好的归零效果

#define IMPROVE_HOMING_RELIABILITY // 可以在上面单独设置归零时的电流参数(xx_CURRENT_HOME)，以便得到最好的归零效果

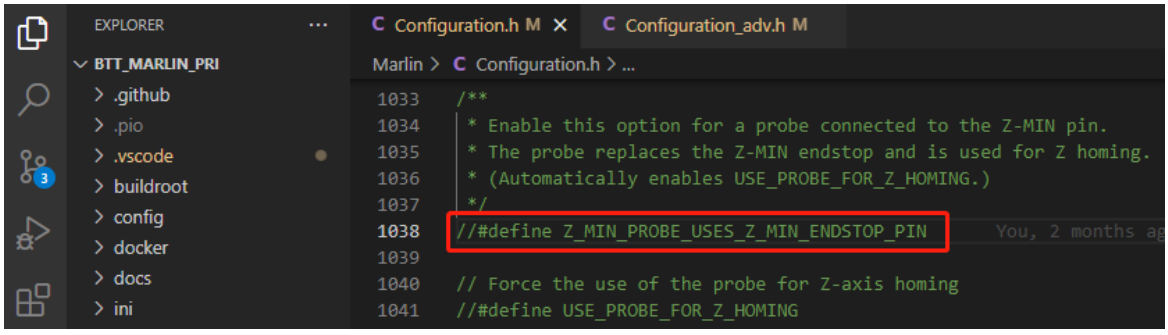
4.3.6 100K NTC 或 PT1000

通过跳帽设置热敏电阻的上拉电阻为 4.7K（搭配 100K NTC）或是 2.2K（搭配 PT1000），Marlin 固件中 1 代表 100K NTC + 4.7K 上拉电阻，1022 代表 PT1000 + 2.2K 上拉电阻（注意：此种方式读出的温度精度会比 MAX31865 差很多）。

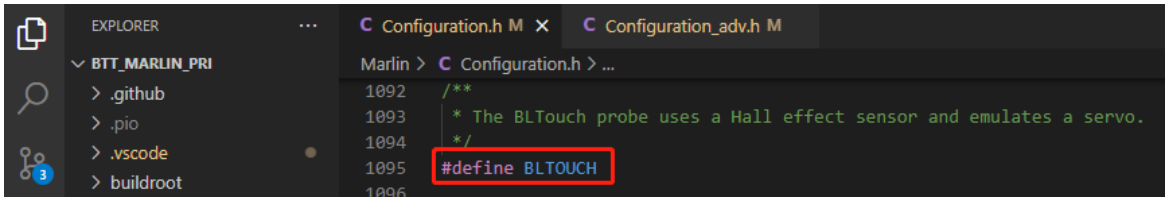
```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 1
#define TEMP_SENSOR_BED 1
```



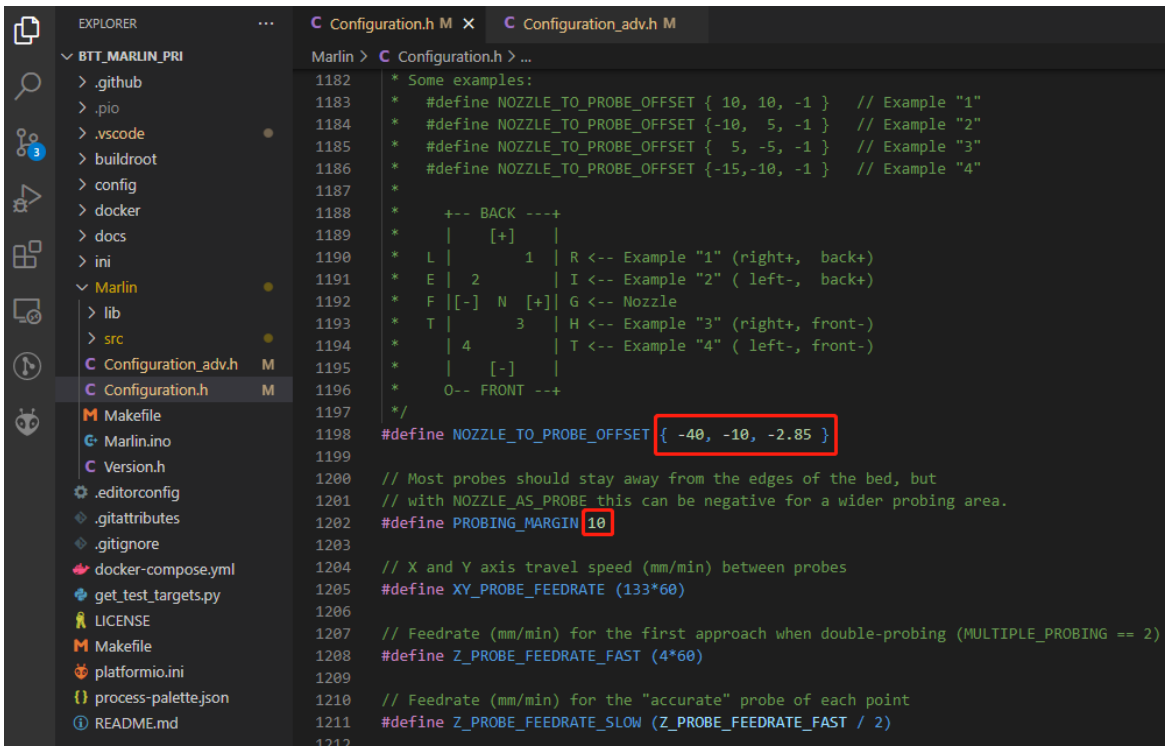
4.3.7 BLTouch




```
//#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN // 不把 Z_PROBE_PIN 重映射到 Z_MIN 端口上
```

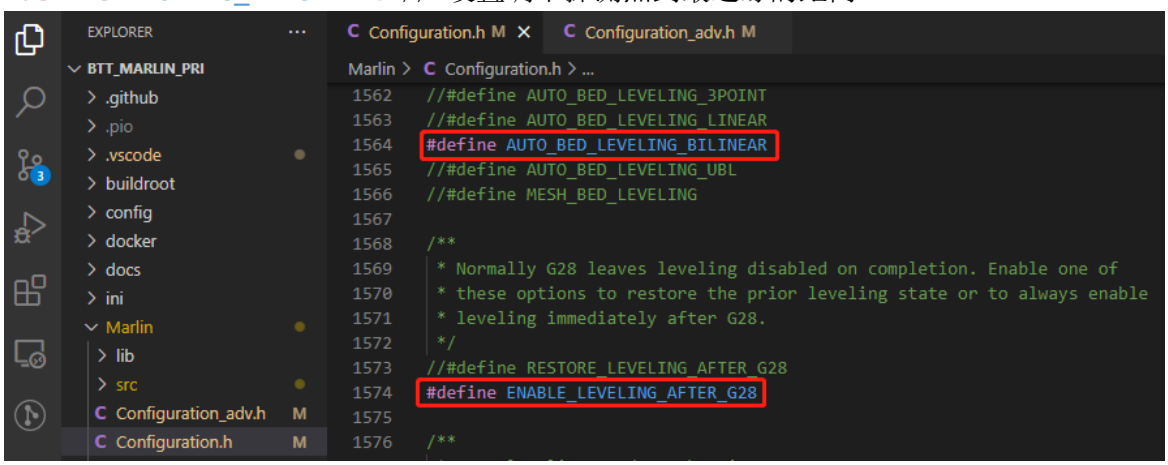



```
#define BLTOUCH  // 使能 BLtouch 功能
```



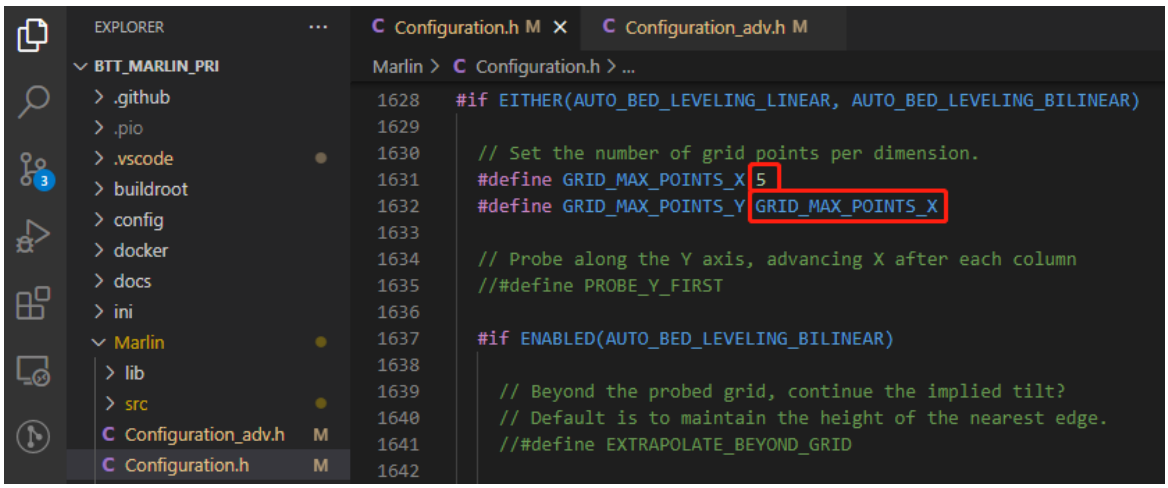
#define NOZZLE_TO_PROBE_OFFSET { -40, -10, -2.85 } // 设置 BLtouch 探针相对于喷嘴的偏移量

#define PROBING_MARGIN 10 // 设置调平探测点到最边缘的距离



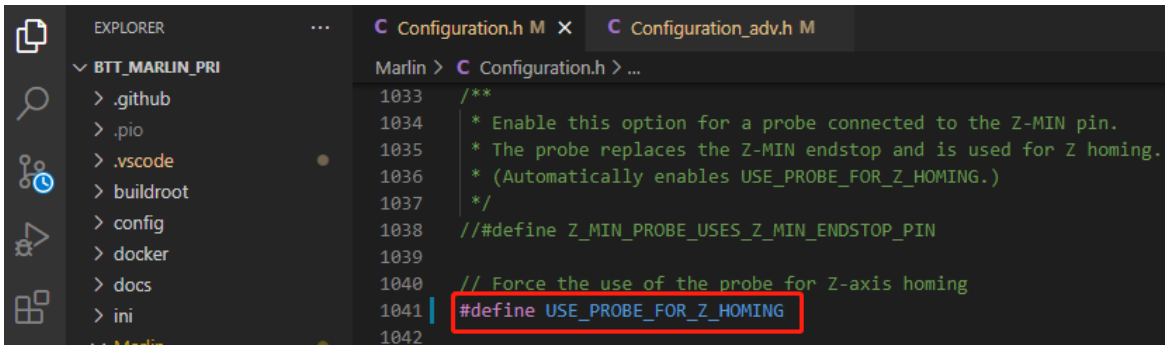
#define AUTO_BED_LEVELING_BILINEAR // 设置调平策略

#define RESTORE_LEVELING_AFTER_G28 // Home 之后自动重新加载调平补偿

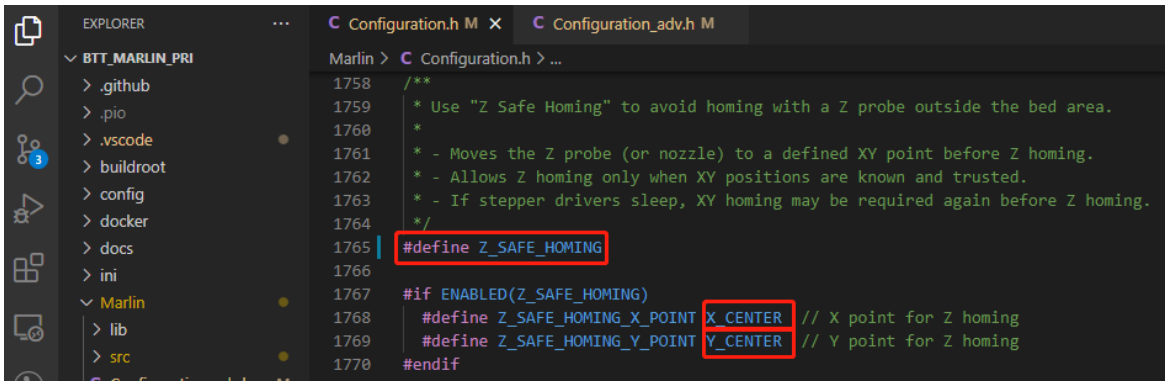


#define GRID_MAX_POINTS_X 5 // 设置调平探测的点数，X 轴探测 5 个点
#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X // Y 轴探测 5 个点

如果想用 BLtouch 作为 Z 轴限位开关，不需要更改接线，只需要修改固件的设置即可

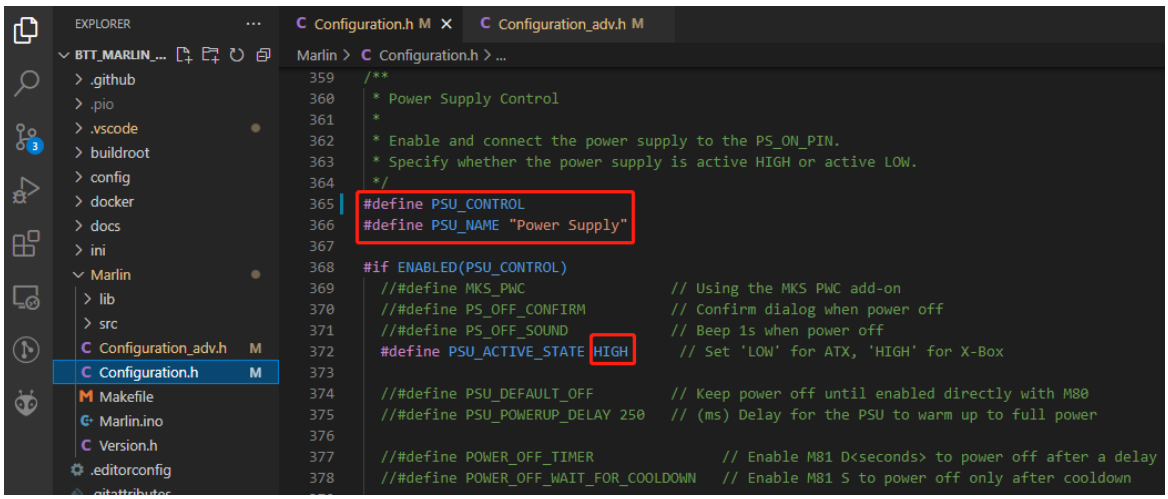


#define USE_PROBE_FOR_Z_HOMING // 使用 Z Probe(BLtouch) 作为 Z 轴 Home 限位开关



#define Z_SAFE_HOMING // Z 轴 Home 时，将 X、Y 移动到指定的坐标(通常是平台中心)，保证 Z 轴 Home 时，Z Probe(BLtouch)的探针在平台的范围内。

4.3.8 打完关机模块(Relay V1.2)



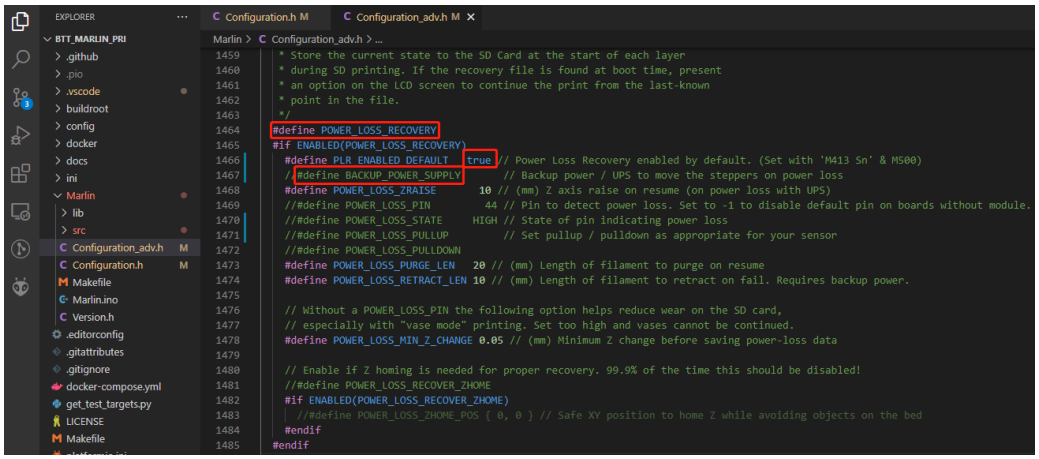
#define PSU_CONTROL // 打开控制电源功能，可以通过 M80 开机、M81 关机

#define PSU_ACTIVE_STATE HIGH // 设置开机的电平， Relay V1.2 模块是高电平开机低电平关机，所以需要设置为 HIGH

4.3.9 断电续打

断电续打目前有两种实现方式

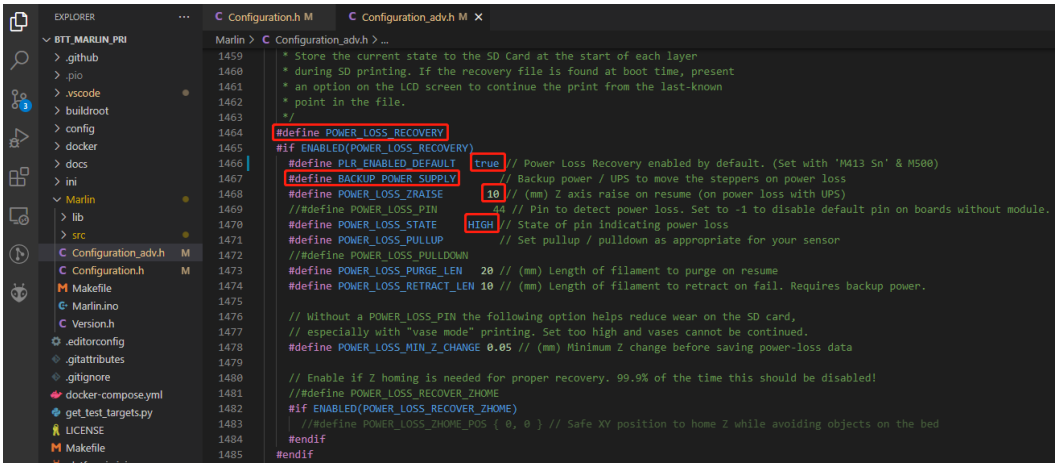
1. 无需外接模块，固件定期向 SD 卡中保存打印状态，断电重启后从 SD 卡中保存的点继续打印，这种方式的缺点就是会向 SD 卡中频繁的写入数据，非常影响 SD 卡的使用寿命。



#define POWER_LOSS_RECOVERY // 使能断电续打功能

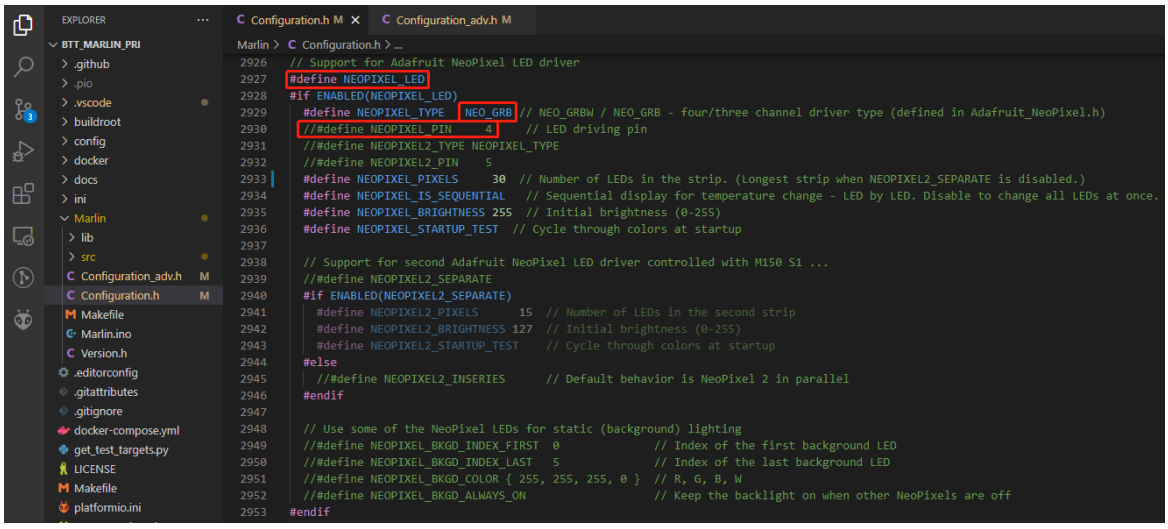
#define PLR_ENABLED_DEFAULT true // true 默认使用开启断电续打

2. 外接 UPS 24V V1.0 模块，断电时给主板提供电源并给主板发送信号，提醒主板保存打印状态，这方式只会在断电时向 SD 卡写入数据，对 SD 卡的使用寿命几乎没有影响。



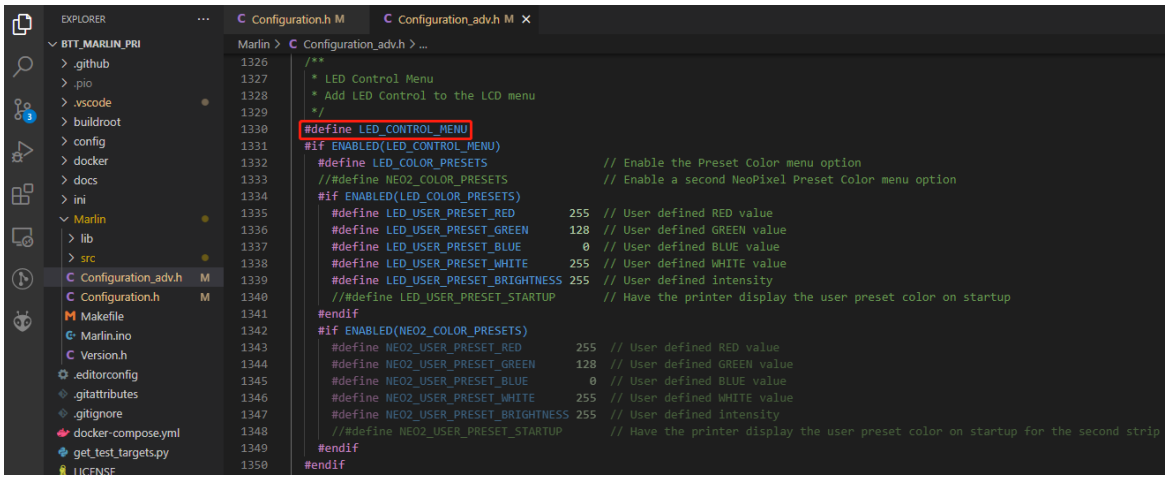

```
#define POWER_LOSS_RECOVERY  // 使能断电续打功能
#define PLR_ENABLED_DEFAULT true // true 默认使用开启断电续打
#define POWER_LOSS_ZRAISE   10 // 断电时喷头抬升 10mm 避免喷头烫坏模型
#define POWER_LOSS_STATE HIGH // 断电时模块反馈的电平，UPS 24V V1.0 正常工作时反馈低电平，断电时反馈高电平，所以设置为 HIGH
```

4.3.10 RGB 彩灯




```
#define NEOPIXEL_LED  // 使能 Neopixel 功能
#define NEOPIXEL_TYPE  NEO_GRB // 设置彩灯的类型
//#define NEOPIXEL_PIN  4 // 屏蔽 PIN 设置，使用主板 pin 文件中正确的信号线
#define NEOPIXEL_PIXELS 30 // 彩灯的数量
#define NEOPIXEL_STARTUP_TEST // 开机时会依次显示红绿蓝三种颜色便于测试
```

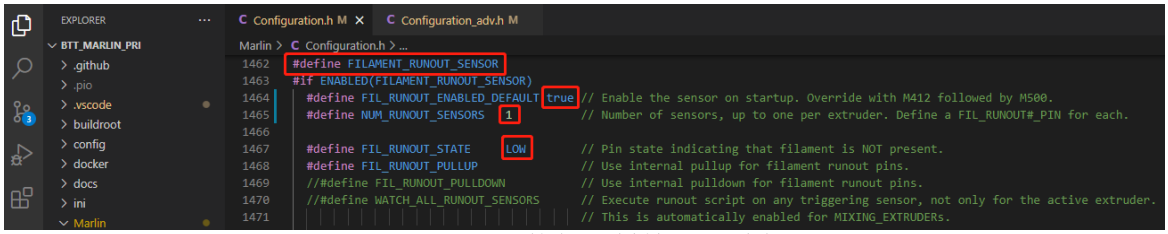
如果启用了 LCD2004、12864、mini12864 之类的显示器，还可以在界面上启用 RGB 的控制菜单



#define LED_CONTROL_MENU // 在屏幕上添加控制 LED 颜色的菜单

4.3.11 断料检测

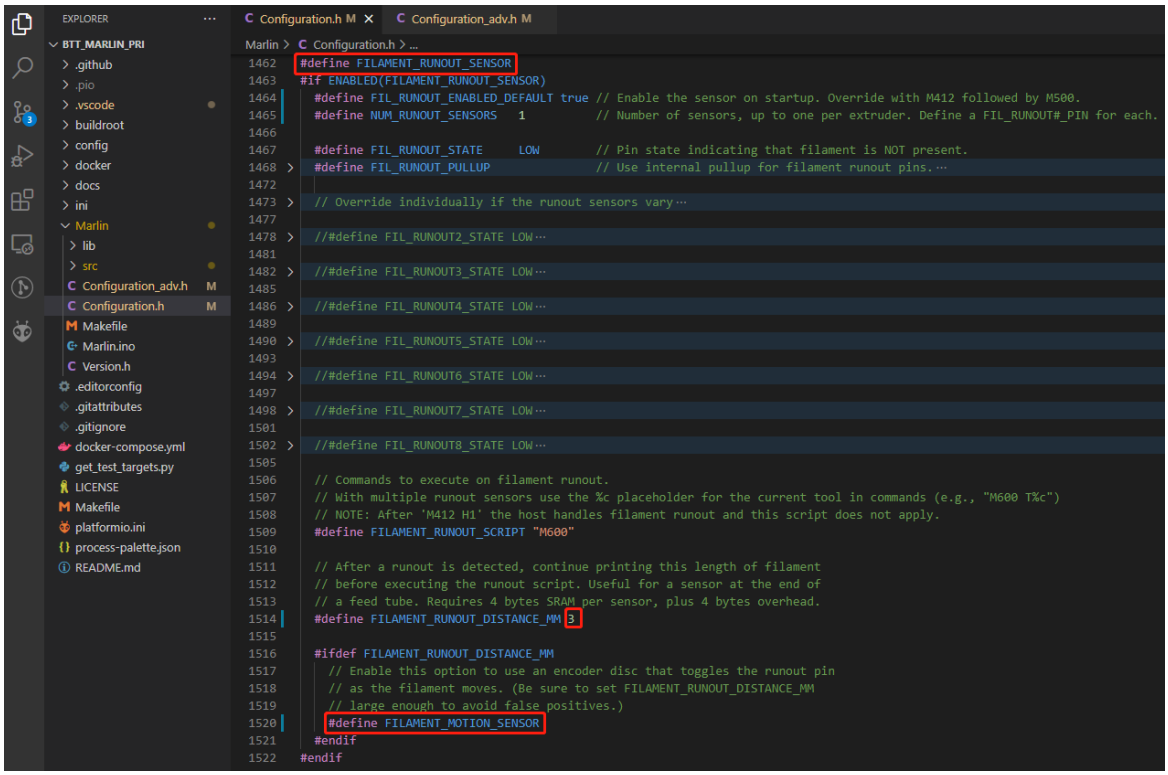
普通的断料检测模块一般是由机械开关设计而成的，模块给主板一个恒定的高低电平代表耗材的状态



#define FILAMENT_RUNOUT_SENSOR // 使能耗材检测的功能
#define FIL_RUNOUT_ENABLED_DEFAULT true // true 默认是打开的状态
#define NUM_RUNOUT_SENSORS 1 // 耗材检测传感器的数量
#define FIL_RUNOUT_STATE LOW // 耗材异常时的电平状态，根据模块实际情况设置，如果耗材异常时模块发出低电平就设置为 LOW

4.3.12 智能耗材检测(SFS V1.0)

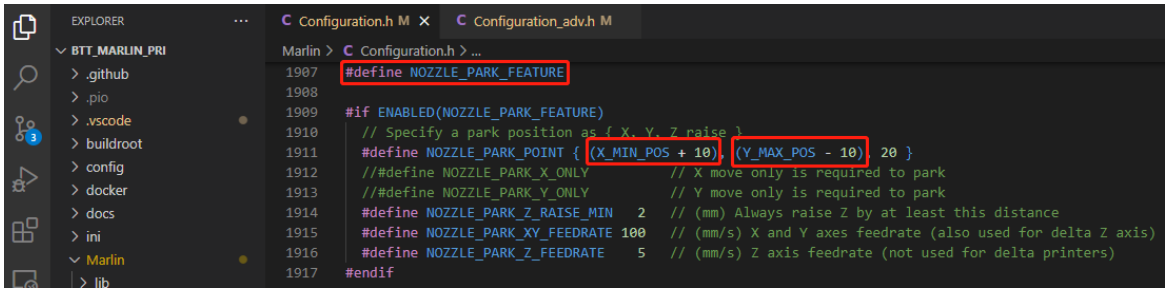
智能耗材检测模块在耗材正常通过时会不断的发出跳变的电平信号，当堵料/断料等异常情况出现，耗材无法正常的通过 SFS，模块就无法发出跳变的信号给主板，主板从而得知耗材异常。



#define FILAMENT_MOTION_SENSOR // 设置耗材传感器为编码器类型

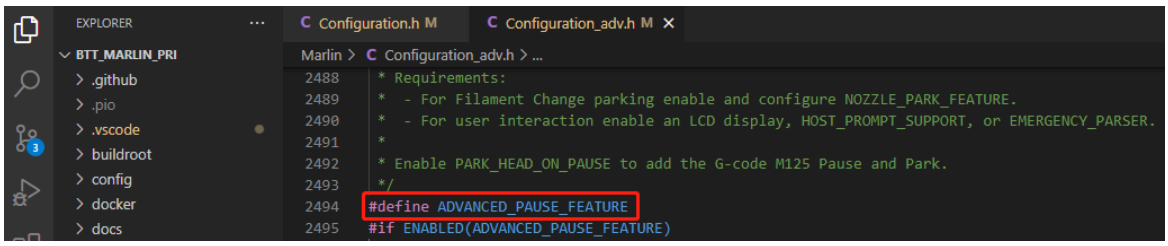
#define FILAMENT_RUNOUT_DISTANCE_MM 7 // 设置检测灵敏度，SFS V1.0 推荐设置为 7mm，耗材 7mm 内如果没有电平跳变就意味着耗材异常

耗材检测还需要通过以下两个地方来设置耗材异常暂停后的动作



#define NOZZLE_PARK_FEATURE // 喷头暂停功能

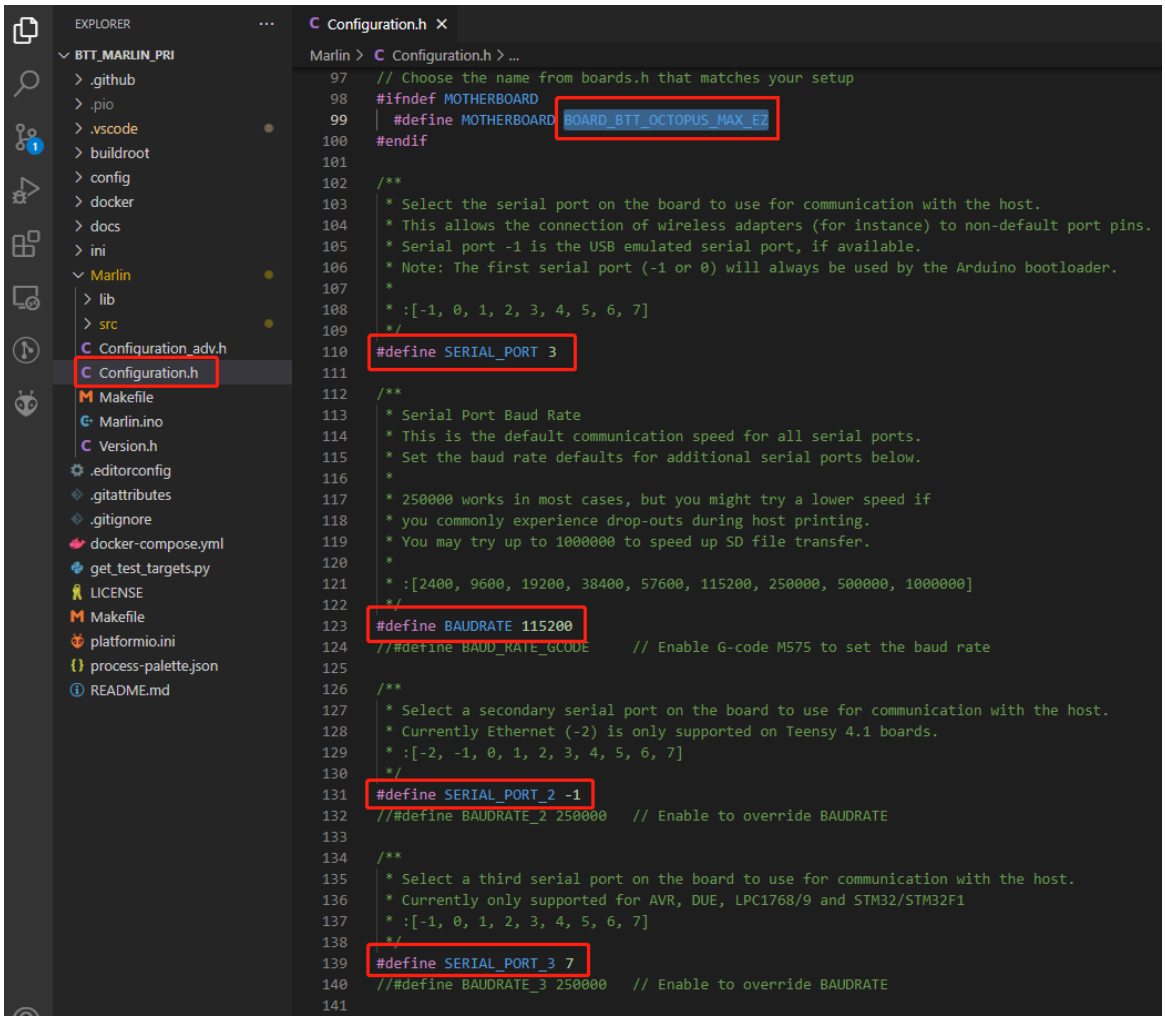
#define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 } //设置喷头暂停时的 X、Y 的坐标以及 Z 轴抬升的高度



#define ADVANCED_PAUSE_FEATURE // 可以设置暂停时耗材回抽的长度及速度，继续打印后耗材挤出的长度和速度等参数

4.3.13 ESP3D

Marlin 中只需设置正确的 “SERIAL_PORT” 和 “BAUDRATE” 即可。主板上 ESP8266 与 Marlin 通信的串口是 UART3，所以将 SERIAL_PORT 设置为 3。

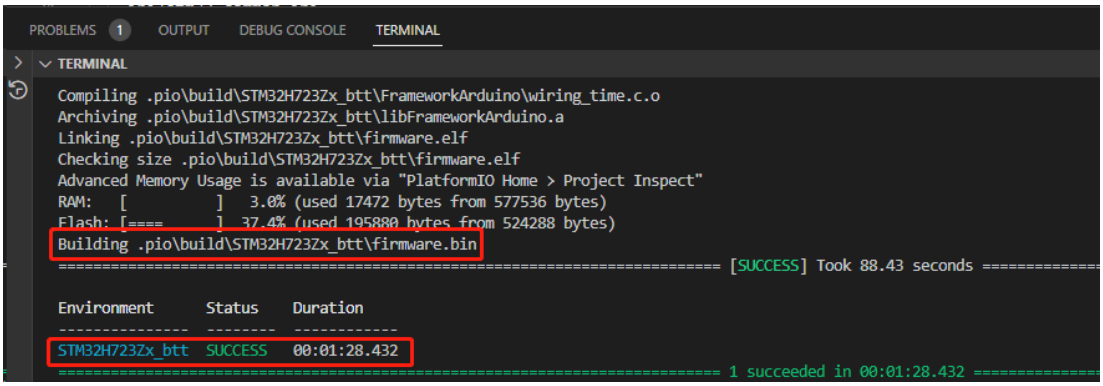


可以在 https://github.com/luc-github/ESP3D 中获取最新的 ESP3D 固件，编译出您自己的二进制文件，将其重命名为 “esp3d.bin” 然后复制到 SD 卡的根目录中，插到主板上然后 Reset，主板中的引导程序会自动将 esp3d.bin 更新到 ESP8266 中，更新完成后文件会被重命名为 “ESP3D.CUR”

4.4 编译固件

1. 点击底部状态栏中的“√”编译固件

2. 编译完成后会生成“firmware.bin”文件，复制到 SD 卡中即可更新固件



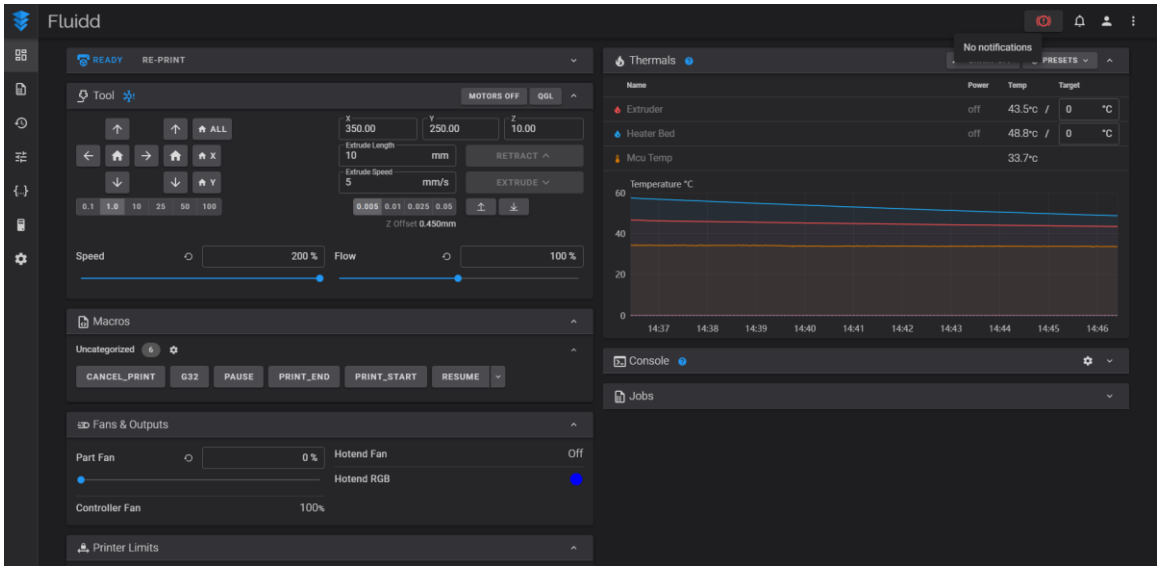
五、Klipper

5.1 准备工作

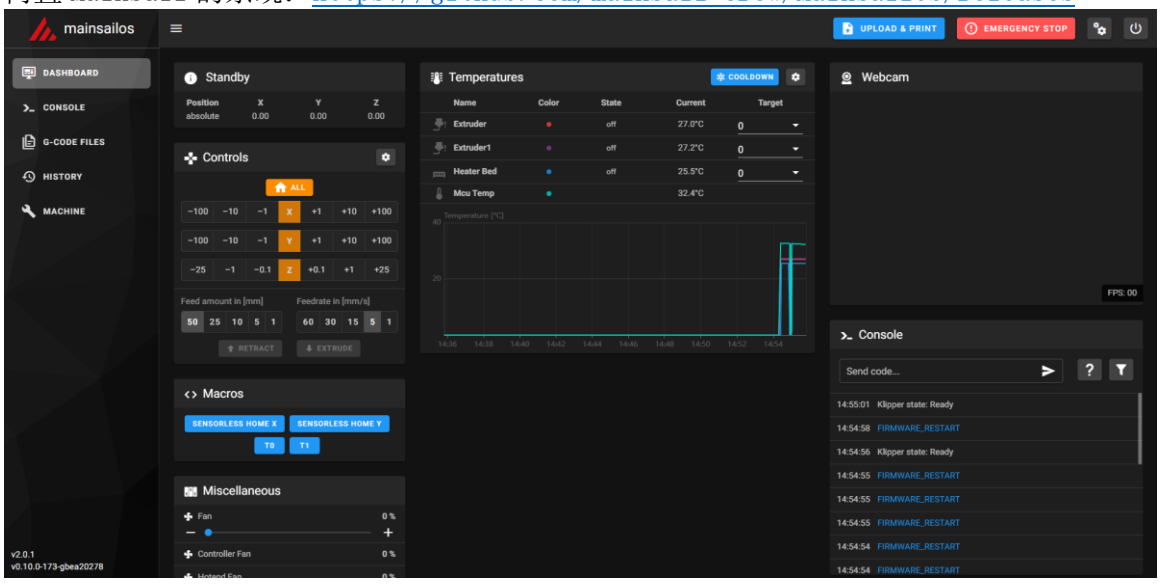
5.1.1 下载系统镜像

下载内置你喜欢的 WebUI 的系统镜像，目前主流的有 Fluidd、Mainsail 等。

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



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



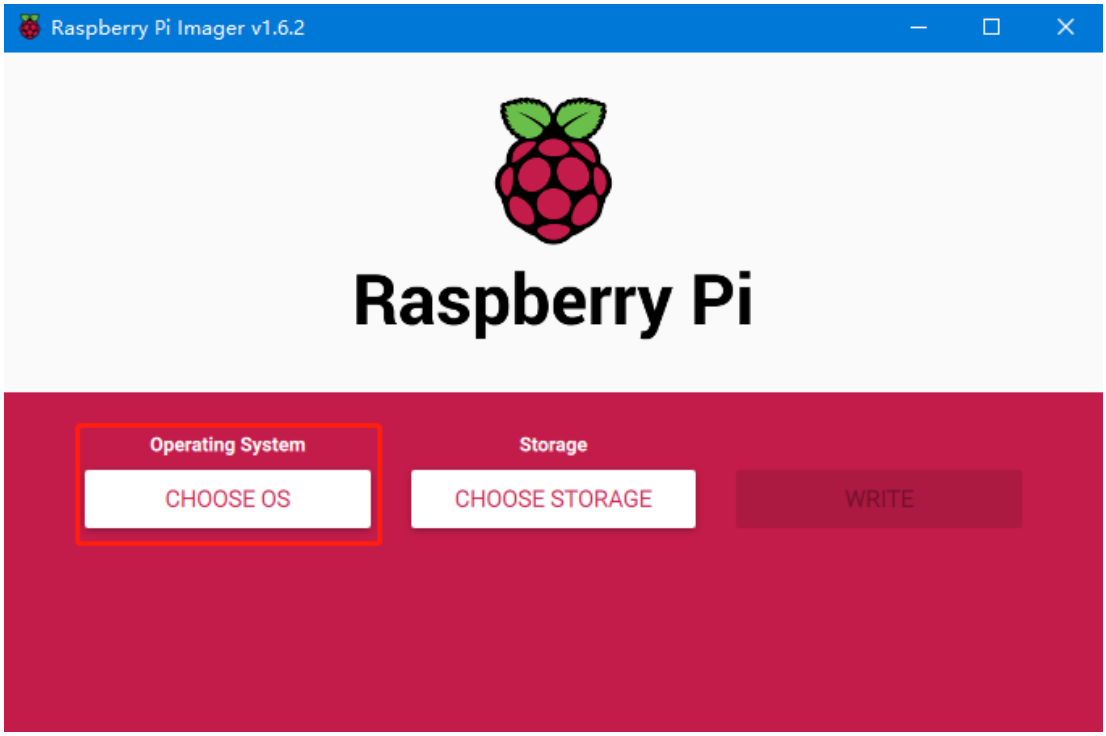
或者参考 Klipper 官方的安装说明 使用 Octoprint

5.1.2 下载并安装 Raspberry Pi Imager

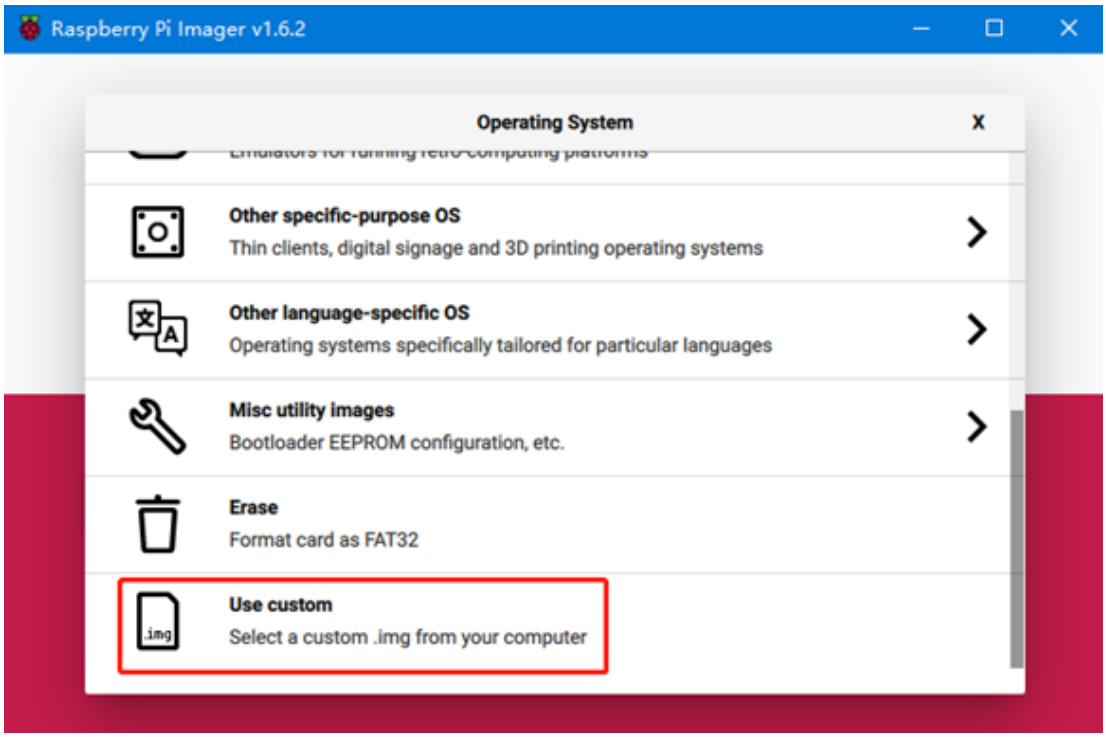
下载并安装树莓派官方的烧录软件 https://www.raspberrypi.com/software/

5.2 烧录镜像

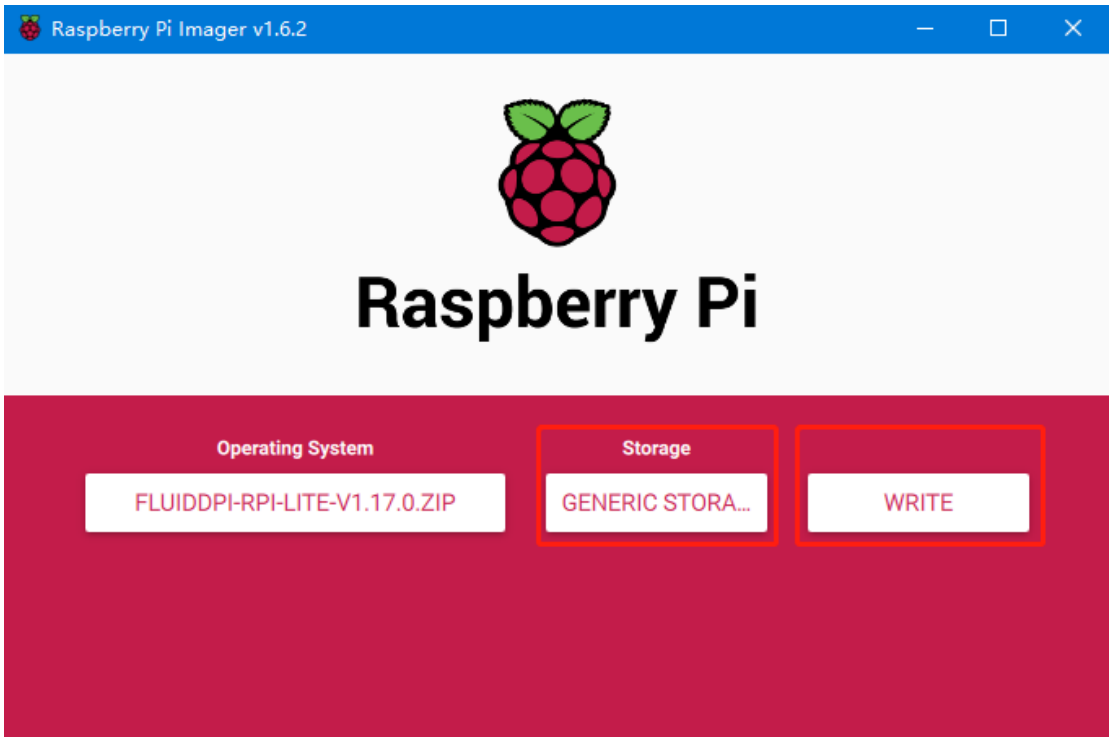
1. 将 Micro SD 卡通过读卡器插入到电脑。
2. 选择系统。



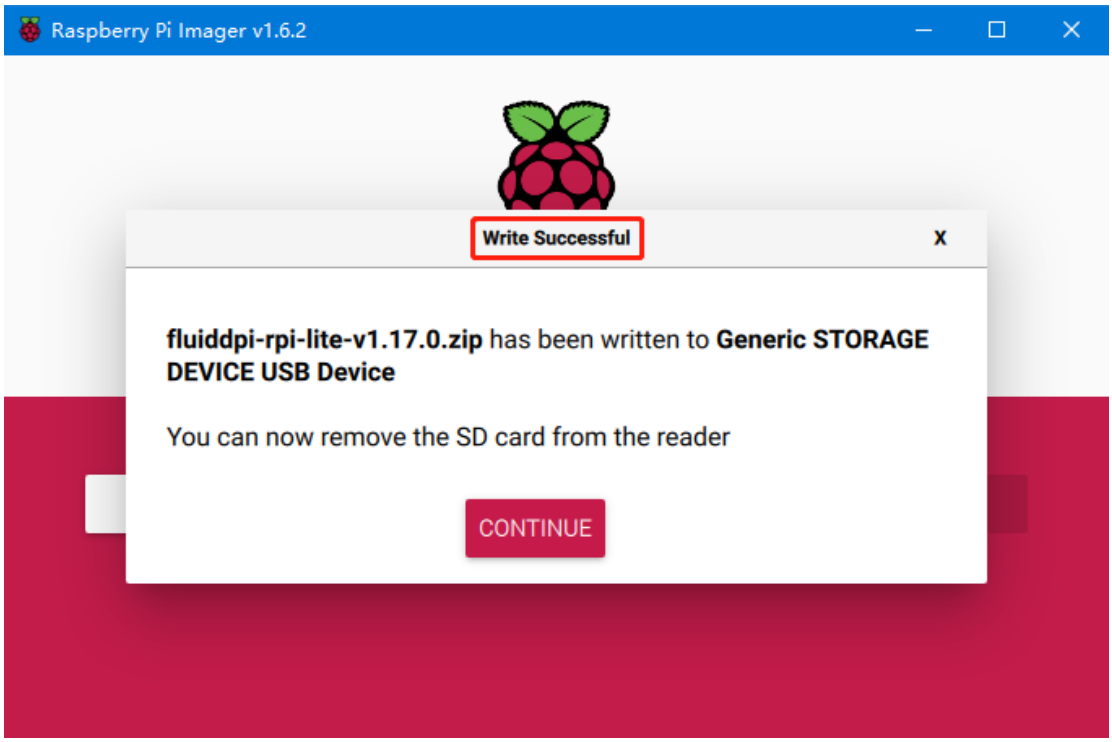
3. 选择“用户自定义”，然后选择下载到电脑中的镜像。



4. 选择待烧录的 SD 卡（烧录镜像会将 SD 卡格式化，千万注意不要选错盘符，否则会将其他存储上的数据格式化），点击“烧录”。



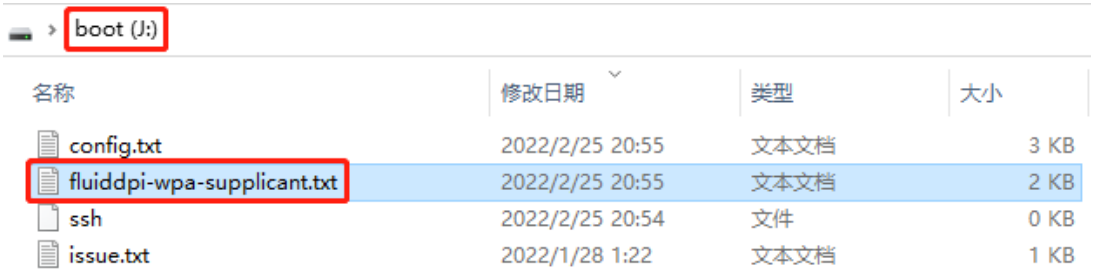
5. 等待烧录完成。



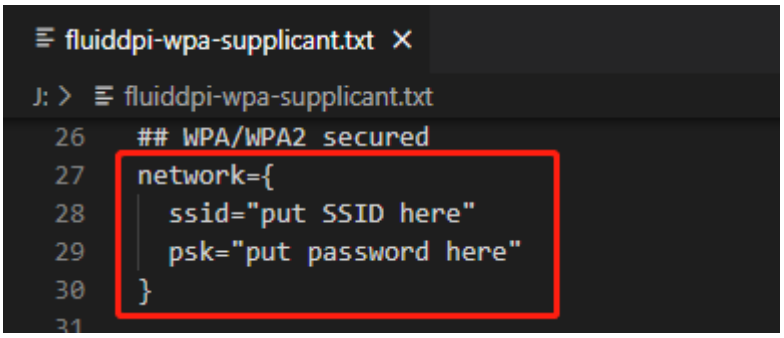
5.3 设置 WIFI

注意：如果使用网线端口而不是 WIFI，可以跳过此步

1. 重新拔插一下读卡器
2. 在 SD 卡的 boot 盘中找到“fluiddpi-wpa-supplicant.txt”或者“mainsail-wpa-supplicant.txt”文件，用 VSCode 打开（不要用 windows 自带的记事本打开）

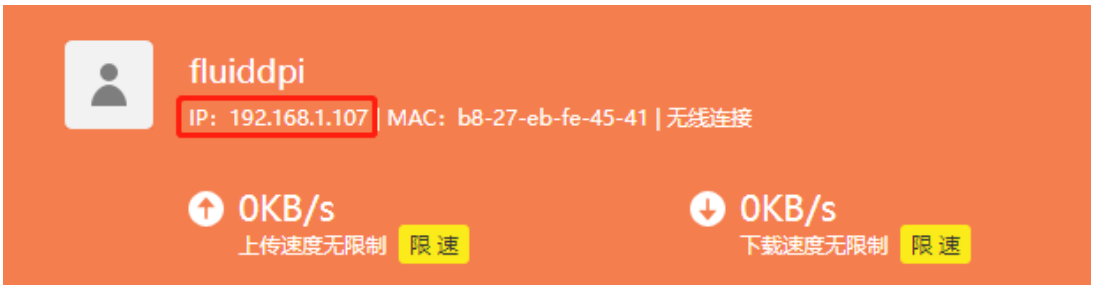


3. 删除红框中四行行首的‘#’字符，然后设置正确的 WIFI 名称和密码后保存

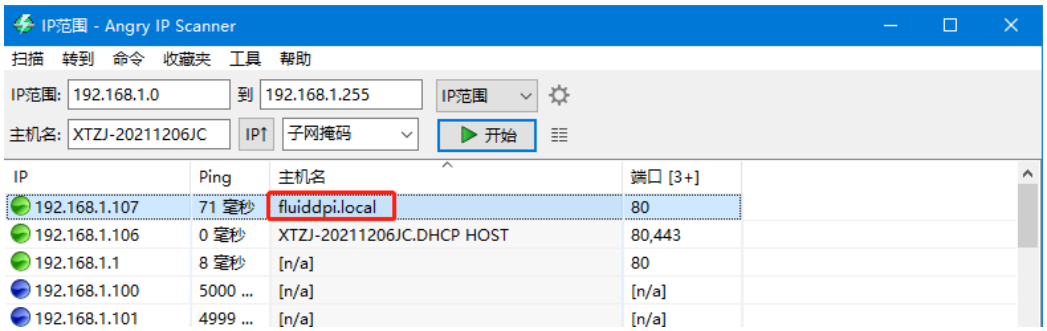


5.4 ssh 软件连接树莓派

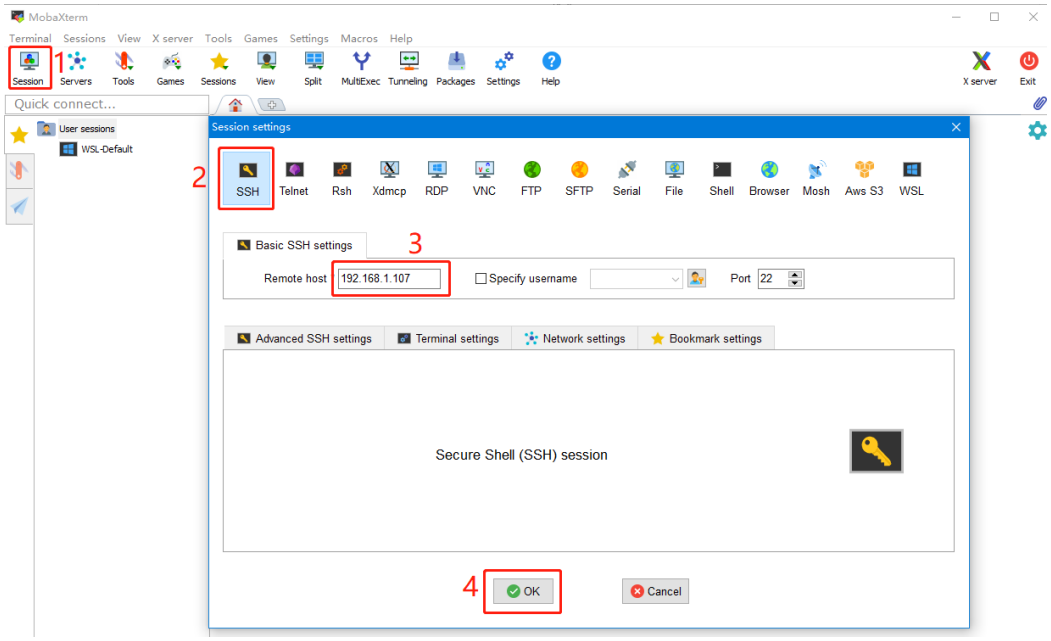
1. 安装 ssh 软件 Mobaxterm：https://mobaxterm.mobatek.net/download-home-edition.html
2. 将 SD 卡插到树莓派，通电后等待系统启动，大概 1~2 分钟
3. 树莓派连上 WIFI 或者插上网线后，会被自动分配一个 IP
4. 进入路由器管理界面找到树莓派的 IP



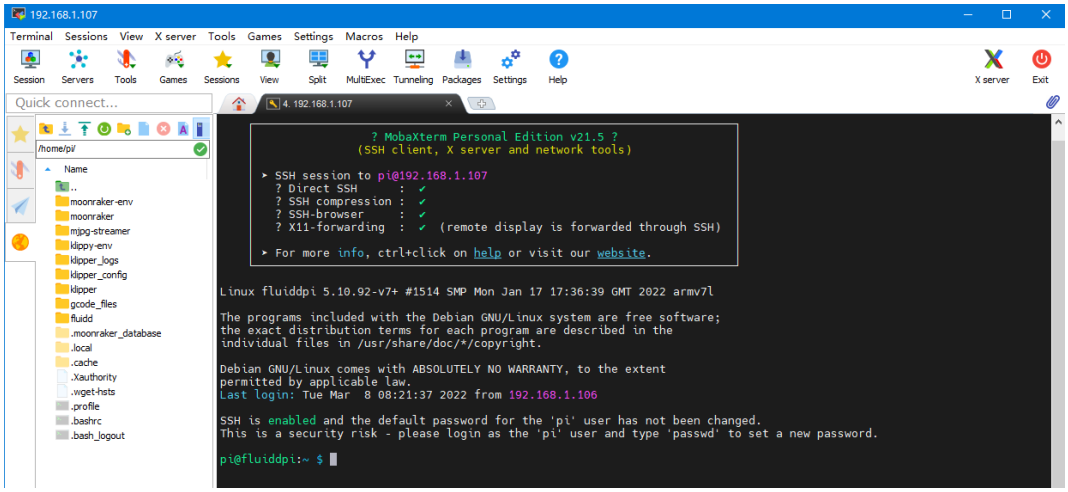
5. 或者使用 https://angryip.org/ 工具，扫描当前局域网下的所有 IP 地址，并使用主机名重新排序，找到主机名为 Fluidd 或者 Mainsail 的设备，如下图所示



6. 打开已经安装的 Mobaxterm 软件，点击“Session”，在弹出的窗口中点击“SSH”，在 Remote host 一栏中输入树莓派的 IP 地址，点击“OK”（注意：电脑和树莓派必须要在同一个局域网下）

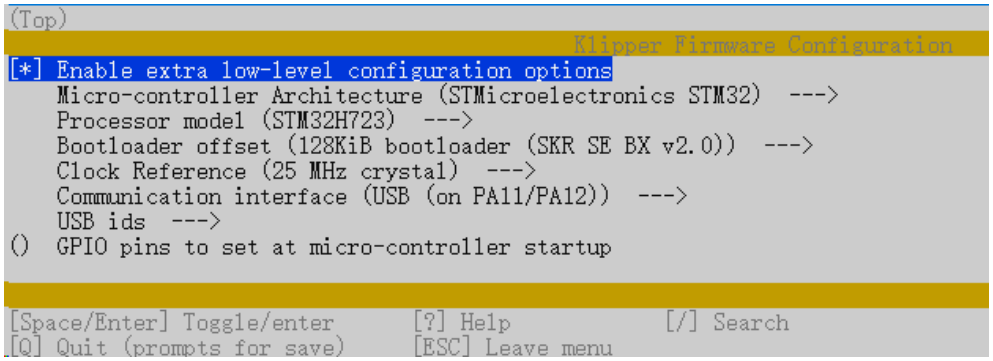


7. 输入登录名 login as：pi 登录密码：raspberry 进入 SSH 终端界面



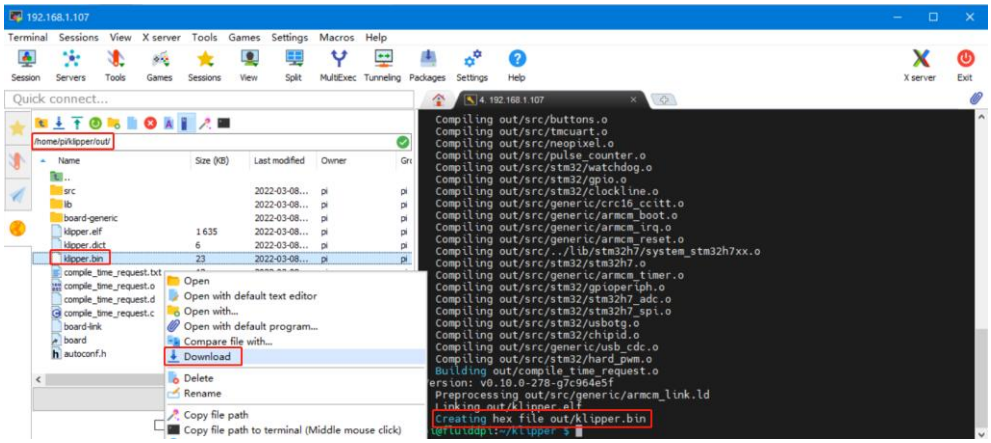
5.5 编译固件

1. ssh 连接到树莓派后，在命令行输入：
   `cd ~/klipper/`
   `make menuconfig`
   使用下面的配置编译固件(如果没有下列选项，请更新 Klipper 固件源码到最新版本)
   *** [*] 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)) --->**



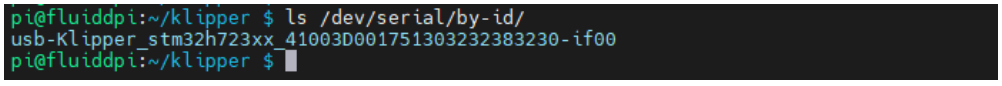
2. 配置选择完成后，输入 `q` 退出配置界面，当询问是否保存配置是选择 "Yes"

3. 输入 `make` 编译固件，当 `make` 执行完成后会在树莓派的 `home/pi/kliiper/out` 文件夹中生成我们所需要的`klipper.bin`固件，在 ssh 软件左侧可以直接下载到电脑中



4. 将 klipper.bin 重命名为“firmware.bin”，复制到 SD 卡中即可更新固件

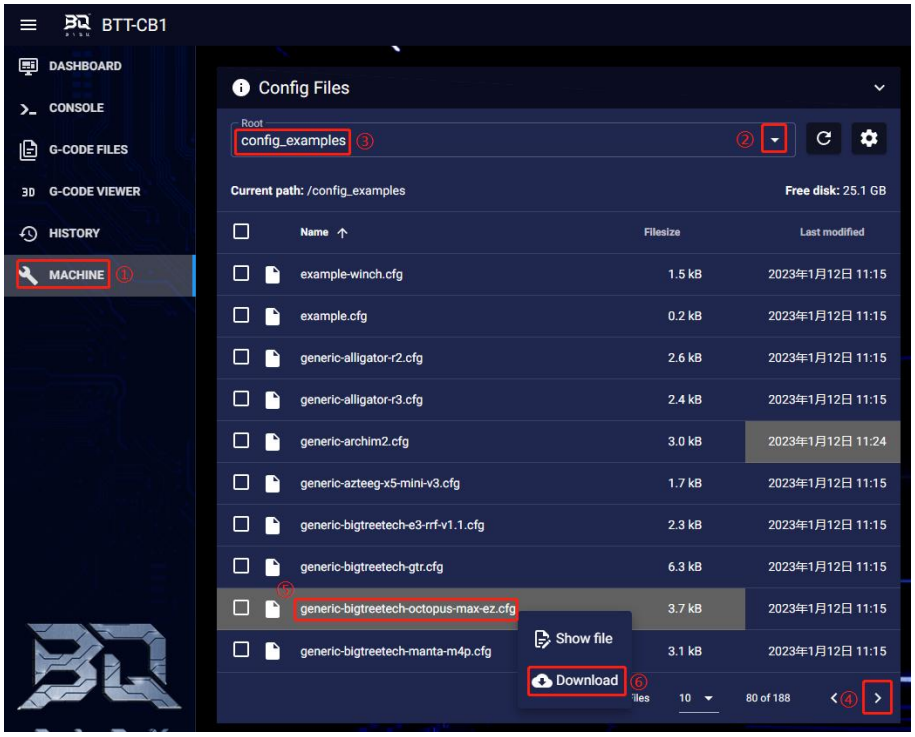
5. 在命令行输入：`ls /dev/serial/by-id/`查询主板的 ID 来确认固件是否烧录成功，如果烧录成功了会返回一个 klipper 的设备 ID，如下图所示



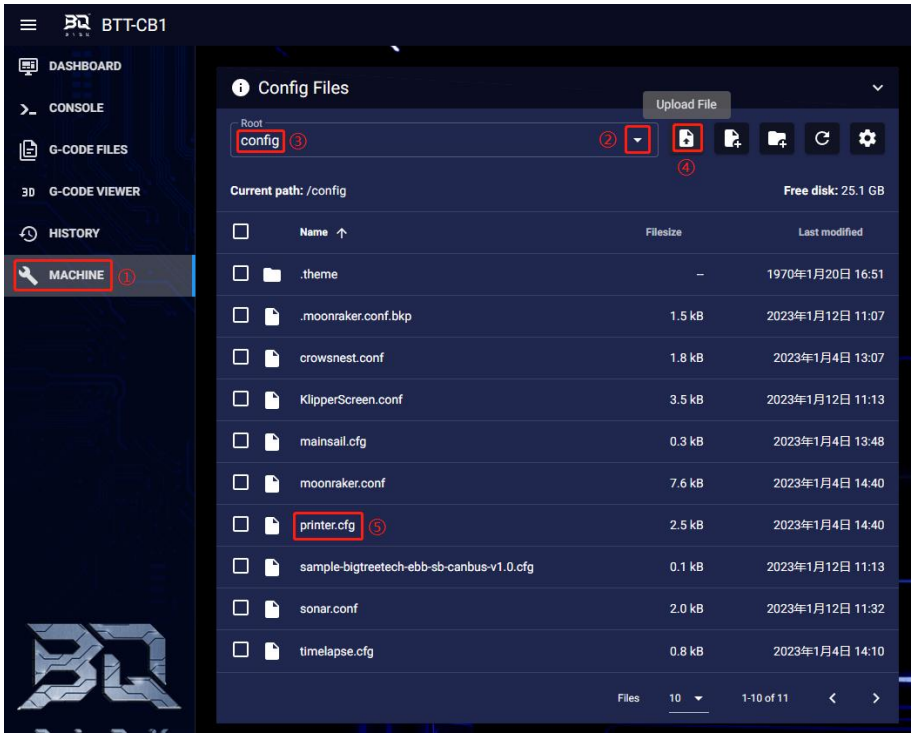
复制保存此 ID，配置文件中需要设置此 ID

5.6 配置 Klipper

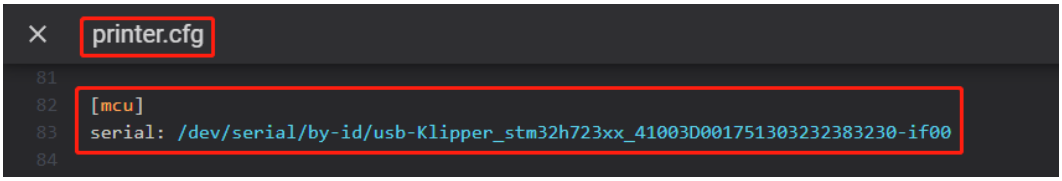
1. 在电脑的浏览器中输入树莓派的 IP 访问，如下图所示的路径中下载主板的参考配置，如果找不到此文件，请更新 Klipepr 固件源码到最新版本，或者到 github 下载 https://github.com/bigtreetech/BIGTREETECH-OCTOPUS-Max-EZ



2. 将主板的配置文件上传到 Configuration Files 中，并重命名为“printer.cfg”



3. 将配置文件中的 ID 号修改为主板实际的 ID



4. 按照 https://www.klipper3d.org/Overview.html 的说明配置机器的具体功能

六、固件更新

Micro SD 卡更新

1. 确保 Micro SD 卡已经被格式化为 FAT32 文件系统
2. 将自行编译或从 github 下载的固件重命名为“firmware.bin”（注意：明确电脑系统的扩展名设置，有部分用户隐藏了扩展名，“firmware.bin”实际显示的是“firmware”）
3. 将“firmware.bin”复制到 Micro SD 卡的根目录中
4. 将 Micro SD 卡插入主板的卡槽中，给主板重新通电，主板的引导程序会自动更新固件
5. 固件更新的过程中，主板右上角的状态指示灯会开始闪烁
6. 当状态指示灯停止闪烁并且 Micro SD 卡中的文件名被重命名为“FIRMWARE.CUR”代表固件更新成功

七、注意事项

1. 所有的拔插操作请在断电的情况下进行
2. 使用风扇时，注意电压选择必须与风扇工作电压一致，防止损坏风扇

如果您还需要此产品的其他资源，可以到 https://github.com/bigtreetech/ 上自行查找，如果无法找到您所需的资源，可以联系我们的售后支持。

若您使用中还遇到别的问题，欢迎您联系我们，我们定会细心为您解答；若您对我们的产品有什么好的意见或建议，也欢迎您回馈给我们，我们也会仔细斟酌您的意见或建议，感谢您选择 BIGTREETECH 制品，谢谢！