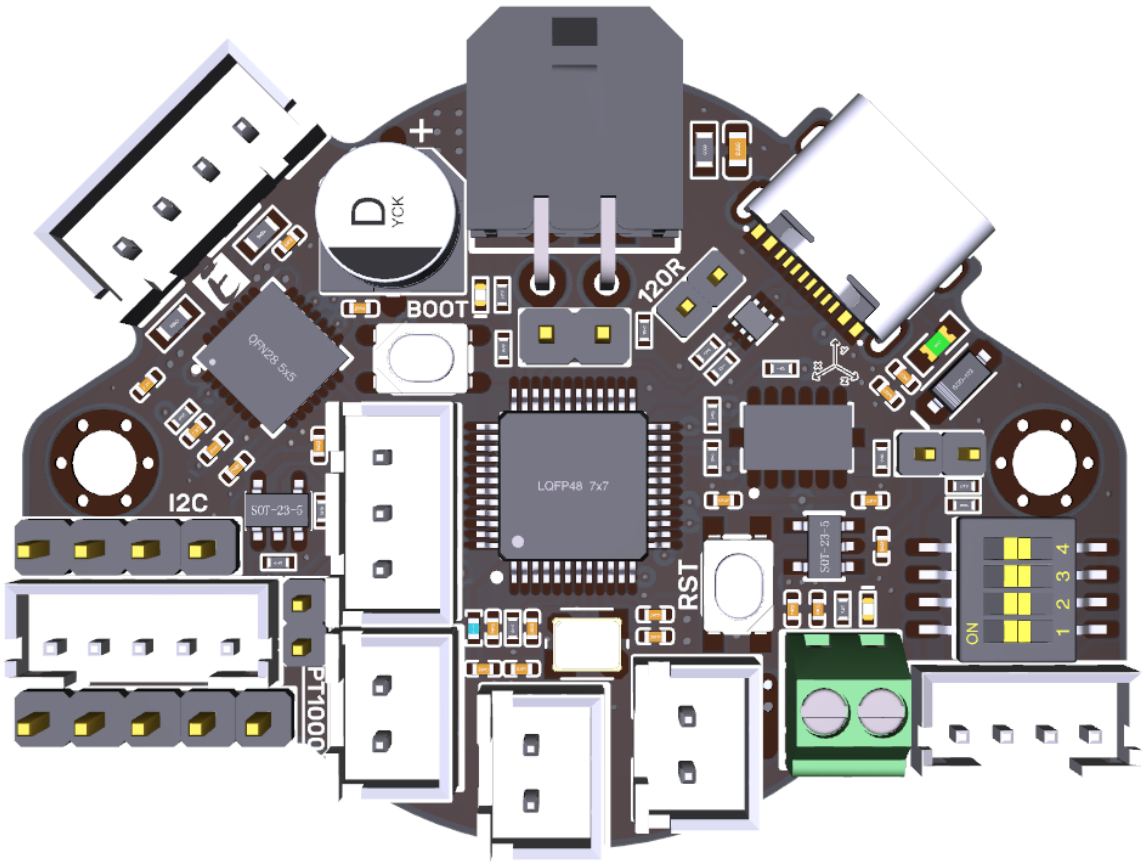
BIGTREETECH EBB36 CAN V1.1

使用说明



目录

修订历史

版本	修改说明	日期
01.00	初稿	2022/05/16
01.01	修复 pin 图加热棒端口错误的标注	2022/05/21
01.02	增加 DFU 更新固件时，加热棒注意事项	2022/05/25

一、产品简介

BIGTREETECH EBB36 CAN V1.1 是深圳市必趣科技有限公司 3D 打印团队针对 36 步进电机类挤出机制作的喷头转接板，可以通过 USB 或者 CAN 进行通讯，大大简化接线。

1.1 产品特点

1. 主板预留 BOOT 和 RESET 按键，用户可以通过 USB 进入 DFU 模式更新固件
2. 增加热敏电阻部分的保护电路，避免因加热棒漏电导致主控芯片烧毁
3. 热敏电阻可通过跳线选择上拉电阻值，以此方式支持 PT1000（2.2K 上拉电阻），方便客户 DIY 使用
4. USB 通电通过跳线帽选择，有效隔离主板 DC-DC 与 USB 5V
5. 预留 I2C 接口，此端口也可用于断料、堵料检测，或者进行其它功能的 DIY 操作
6. 感性负载接口（风扇）都加了续流二极管，保证在风扇 MOS 关断时，风扇绕组电流有续流回路，有效防止关断时绕组在 MOS 管漏极产生的高压。考虑到板子的尺寸和挤出机风扇的开关特性，采用 SOD-323 封装的肖特基二极管。
7. DCDC 降压电路串接反接二极管，防止后级电路因电源线反接而被损毁
8. 板载 MAX31865（可选功能，无 31865 版本的没有此功能，但是有预留焊盘），支持 2 线/4 线的 PT100/PT1000 选择
9. 支持 CAN 或 USB 通讯，其中 CAN 的终端电阻 120R 可通过跳线帽选择，且预留 CAN 拓展接口
10. USB 口增设 ESD 保护芯片，防止主控被 USB 口静电击穿
11. 限位开关硬件消抖电路
12. 出厂配备 DIY 所需端子，母簧片，双通螺柱及螺丝，极大地满足了客户的 DIY 需求
13. 支持 CAN 总线连接，其数据传输较远、抗噪声能力强、实时性强、可靠性高

1.2 产品参数

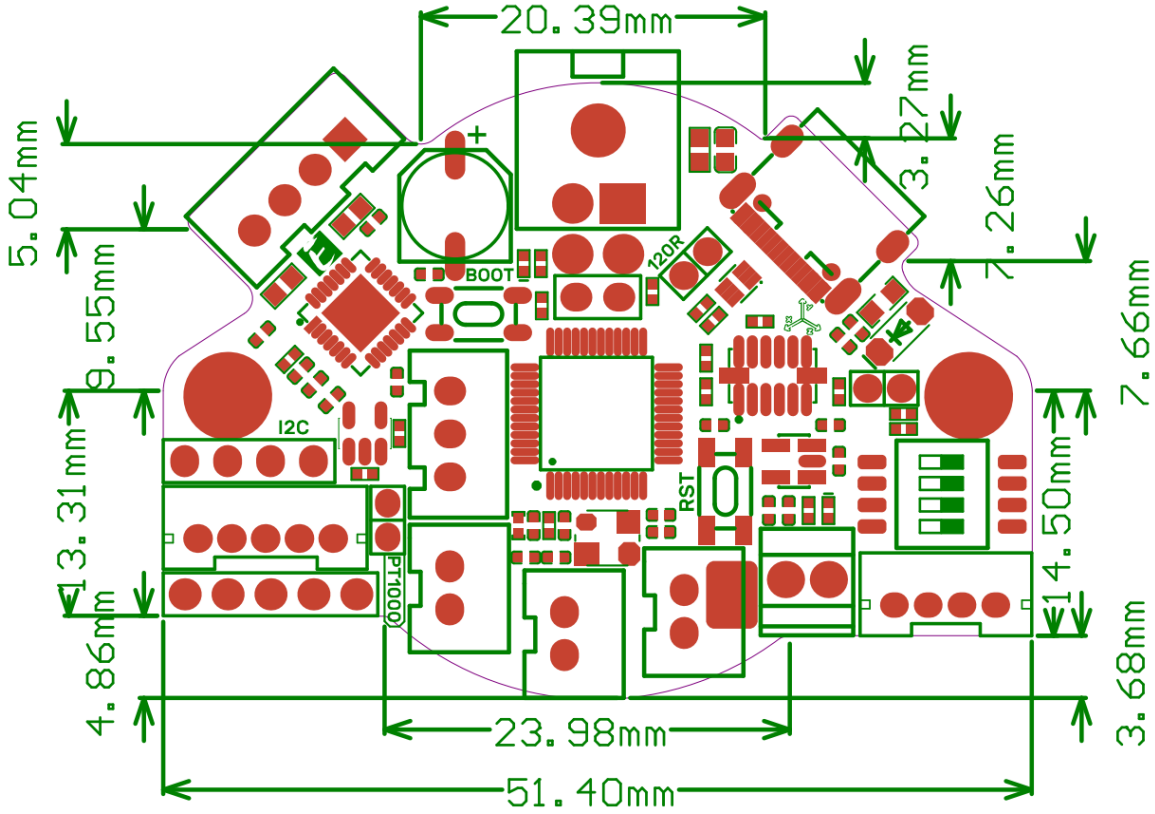
1. 外观尺寸：51.5mm*37mm 详情请参考：**BIGTREETECH EBB36 CAN V1.1-SIZE.pdf**
2. 安装尺寸：孔间距 43.85mm，M3 螺丝孔*2
3. 微处理器：ARM Cortex-M0+ STM32G0B1CBT6 64MHz
4. 输入电压：DC12V-DC24V 6A

5. 逻辑电压：DC 3.3V
6. 加热接口：加热棒（E0），最大输出电流：5A
7. 板载传感器：ADXL345
8. 风扇接口：两个数控风扇（FAN0，FAN1）
9. 风扇接口最大输出电流：1A
10. 拓展接口：EndStop，I2C，Probe，RGB，PT100/PT1000，USB 接口，CAN 接口
11. 电机驱动：板载 TMC2209，硬件地址：00，Rsense：0.11R
12. 驱动工作模式：UART
13. 步进电机接口：E
14. 温度传感器接口：1 路 100K NTC 或者 PT1000（TH0），1 路 PT100/PT1000 可选
15. USB 通信接口：USB-Type-C
16. DC-DC 5V 输出最大电流：1A

1.3 固件支持

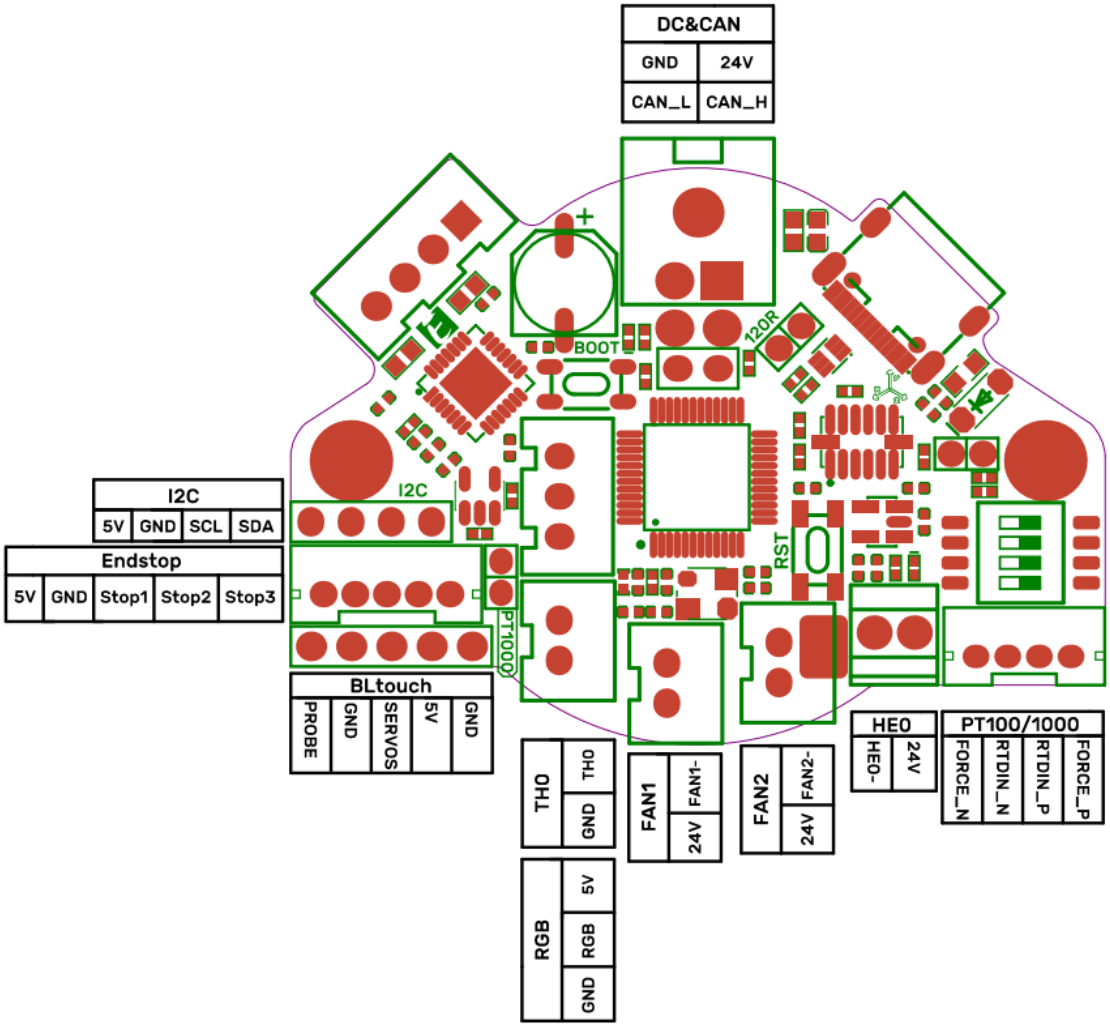
此产品当前仅支持 Klipper 固件

1.4 产品尺寸


20.39mm
3.27mm
7.26mm
7.66mm
5.04mm
9.55mm
13.31mm
14.50mm
4.86mm
3.68mm
23.98mm
51.40mm
BOOT
120R
I2C
RST
PT1000

二、外设接口

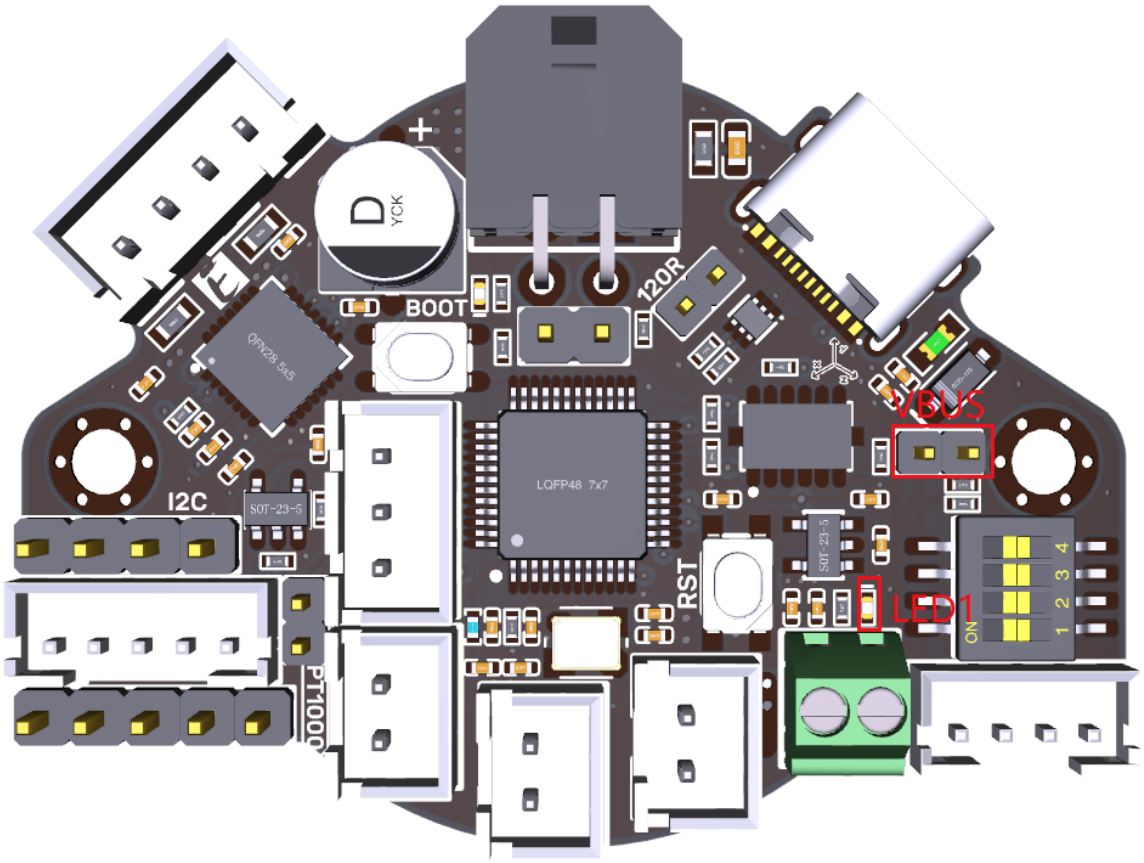
2.1 Pin 脚说明



三、接口介绍

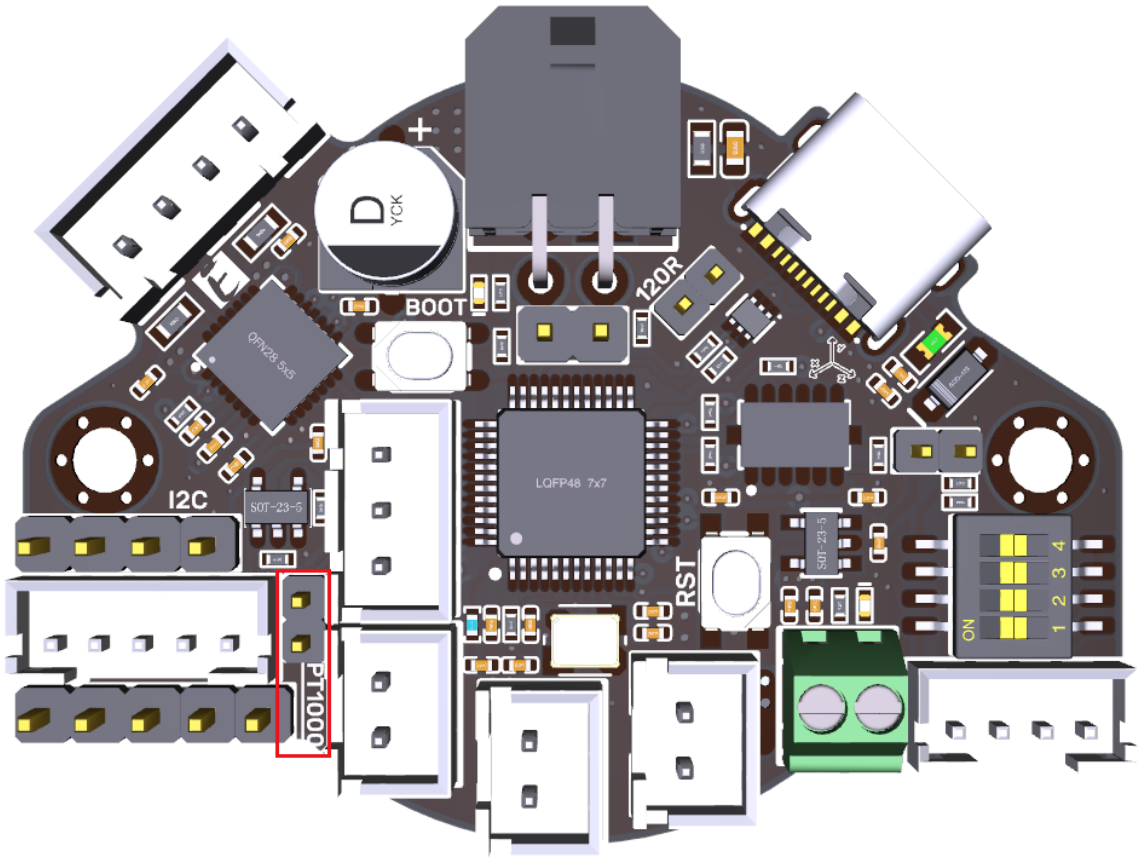
3.1 USB 供电

主板上电之后， LED1 黄绿灯会亮起，表示供电正常。板子右边的 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 VUSB 短接。



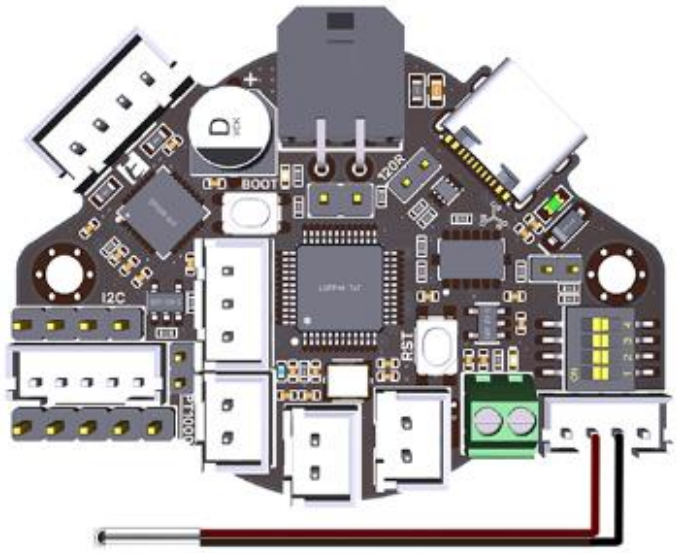
3.2 100K NTC 或 PT1000 设置

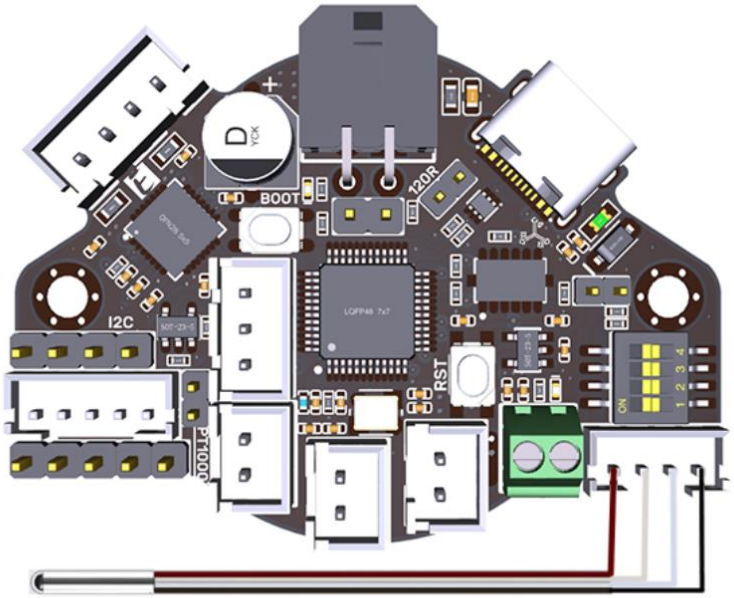
1. 不带 31865 版本：使用 100K NTC 热敏电阻时，无需插入跳线帽，此时 TH0 的上拉电阻为 4.7K。使用 PT1000 时，需使用跳帽短下图红框中的两 Pin，此时 TH0 的上拉电阻为 2.2K（注意：此种方式读出的温度精度会比 MAX31865 差很多）。



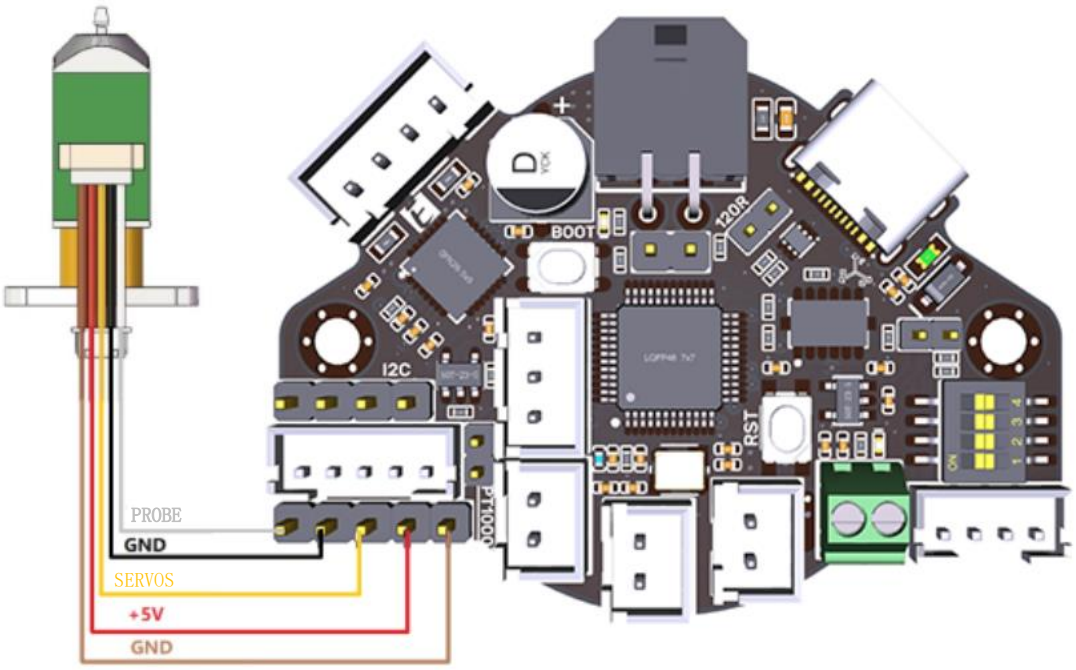
2. 带 31865 版本：通过拨码开关进行选择 PT100/PT1000，两线或者四线；

1	2	3	4	Sensor Model
ON	ON	ON	OFF	Two lines PT100
ON	ON	OFF	ON	Two linesPT1000
OFF	OFF	ON	OFF	Four-wire PT100
OFF	OFF	OFF	ON	Four-wire PT1000

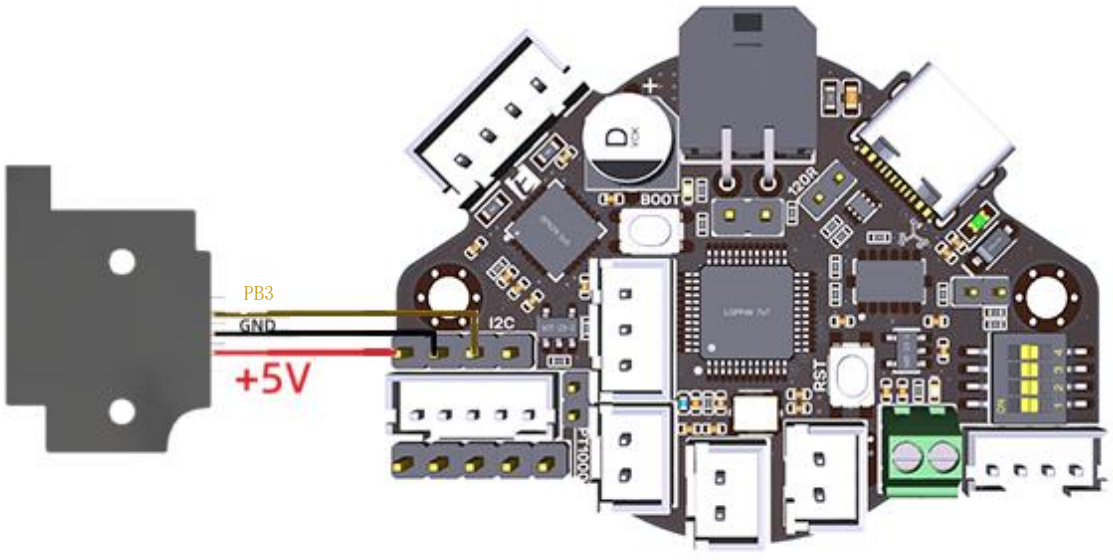

BOOT
120R
I2C
RST
PT1000


BOOT
120R
I2C
RST
PT1000

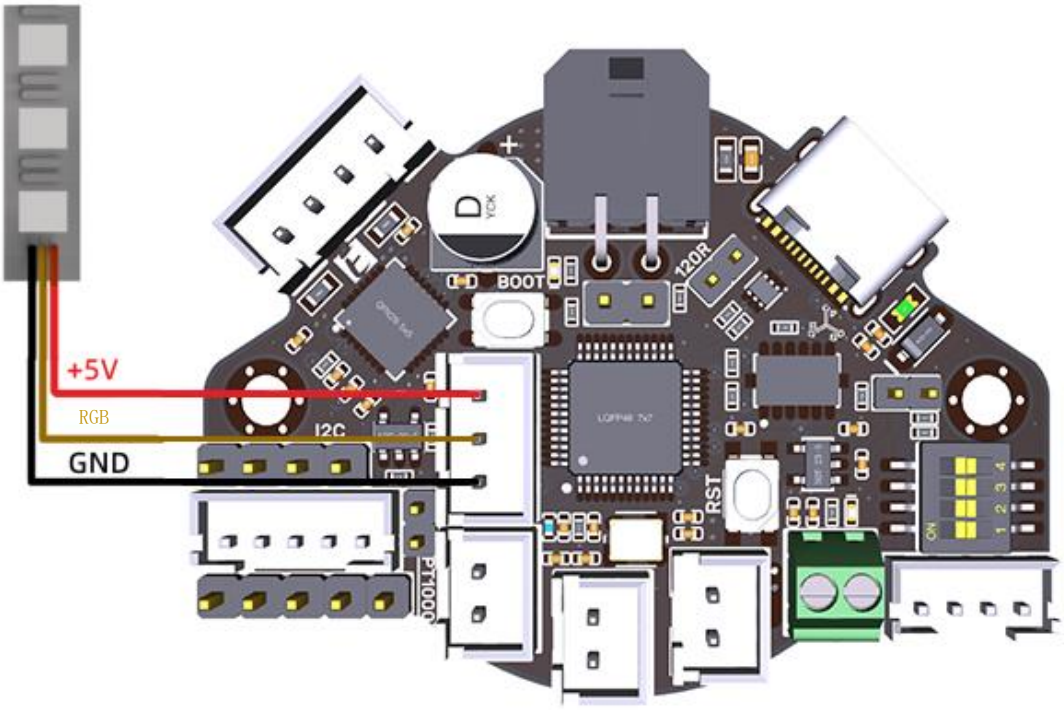
3.3 BL-Touch 接线



3.4 断料检测接线



3.5 RGB 接线



四、Klipper

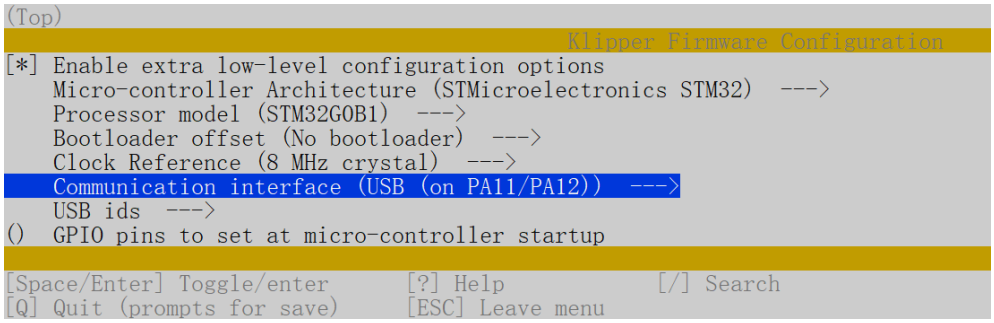
4.1 编译固件

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)

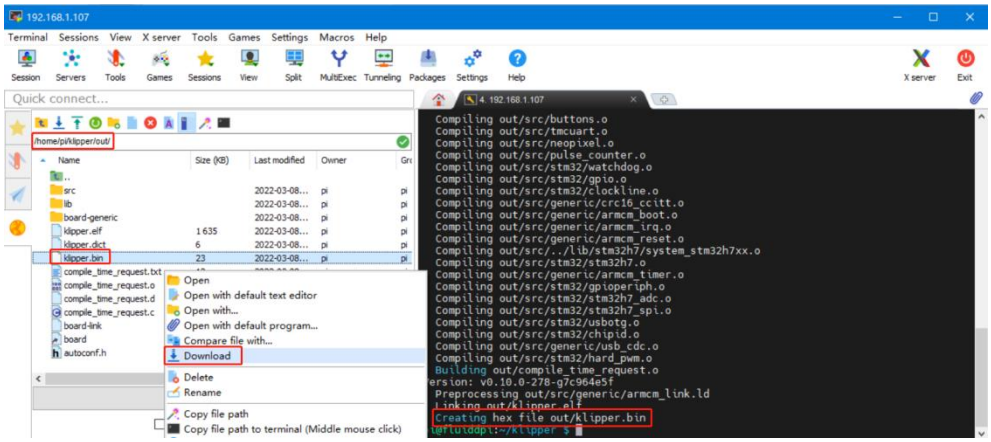
```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
  如果使用 Type-C 上的 USB 通信
    Communication interface (USB (on PA11/PA12))  --->
  如果使用 CANBus 通信
    Communication interface (CAN bus (on PB0/PB1))  --->
    (250000) CAN bus speed
```



注意：在 https://github.com/Klipper3d/klipper/pull/5488 合并到 Klipper 主分支之前，官方的固件是不支持 STM32G0B1 的 CAN bus 功能的。如果使用 CANBus 通信，可以使用我们 github 上编译好的 firmware_canbus.bin 固件，或者使用我们的源码自行编译 https://github.com/bigtreetech/klipper/tree/stm32g0b1-canbus

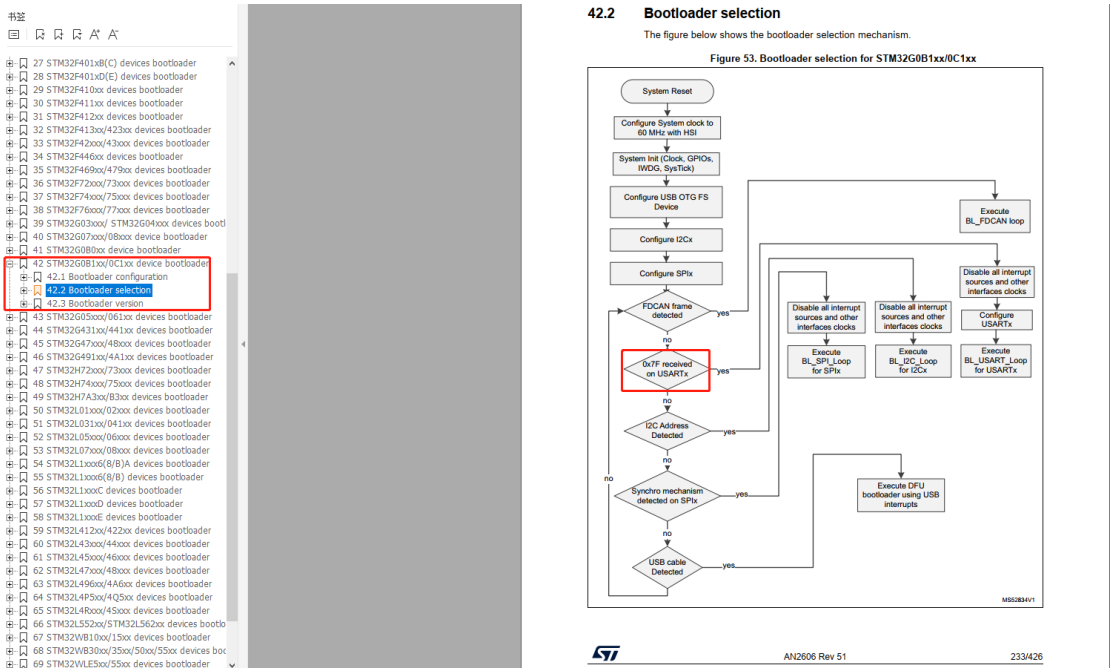
2. 配置选择完成后，输入 `q` 退出配置界面，当询问是否保存配置是选择 "Yes"

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

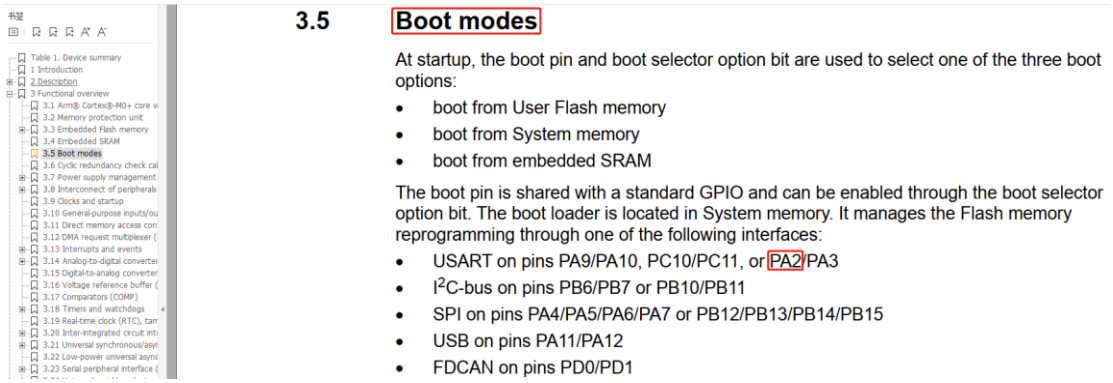


4.2 固件更新

注意：通过 Type-C 端口使用 DFU 更新固件时，STM32G0B1CB 需要跳转到 System memory 区域执行 Bootloader 程序(STMicroelectronics 出厂写死的)，参考手册 AN2606 中的描述(https://www.st.com/content/ccc/resource/technical/document/application_note/b9/9b/16/3a/12/1e/40/0c/CD00167594.pdf/files/CD00167594.pdf/jcr:content/translations/en.CD00167594.pdf)，此 Bootloader 的初始化流程如下图所示：



在进入 USB DFU 模式之前，还会初始化 USART 的 IO。参考 STM32G0B1CB 数据手册中的描述(https://www.st.com/resource/en/datasheet/stm32g0b1cb.pdf)，进入 DFU 模式后，PA2 引脚会被 System memory 区域中的 Bootloader 配置输出高电平。



3.5 Boot modes

At startup, the boot pin and boot selector option bit are used to select one of the three boot options:

- boot from User Flash memory
- boot from System memory
- boot from embedded SRAM

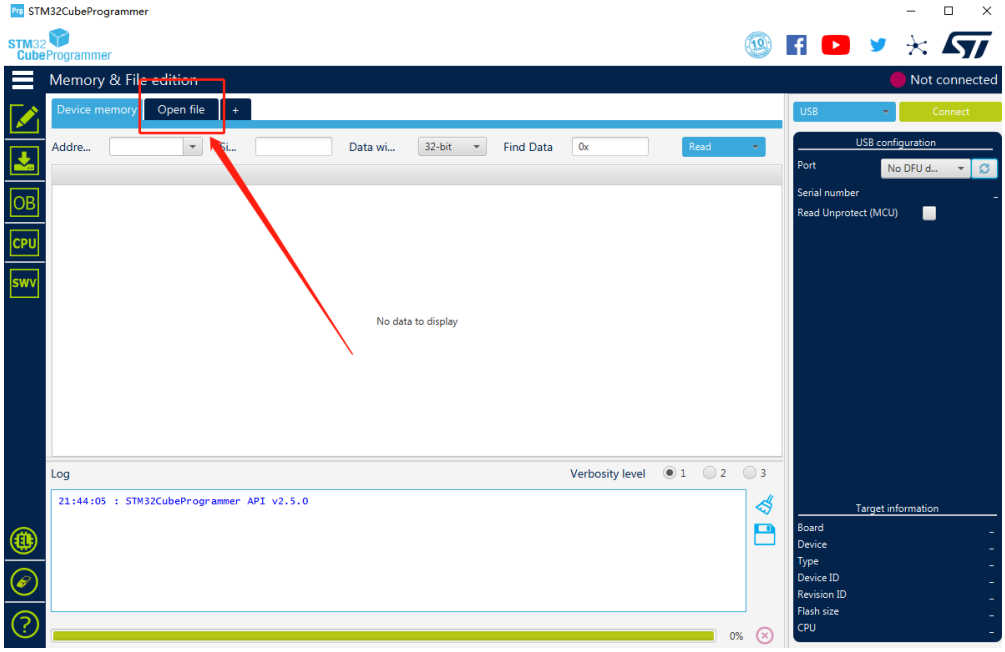
The boot pin is shared with a standard GPIO and can be enabled through the boot selector option bit. The boot loader is located in System memory. It manages the Flash memory reprogramming through one of the following interfaces:

- USART on pins PA9/PA10, PC10/PC11, or PA2/PA3
- I²C-bus on pins PB6/PB7 or PB10/PB11
- SPI on pins PA4/PA5/PA6/PA7 or PB12/PB13/PB14/PB15
- USB on pins PA11/PA12
- FDCAN on pins PD0/PD1

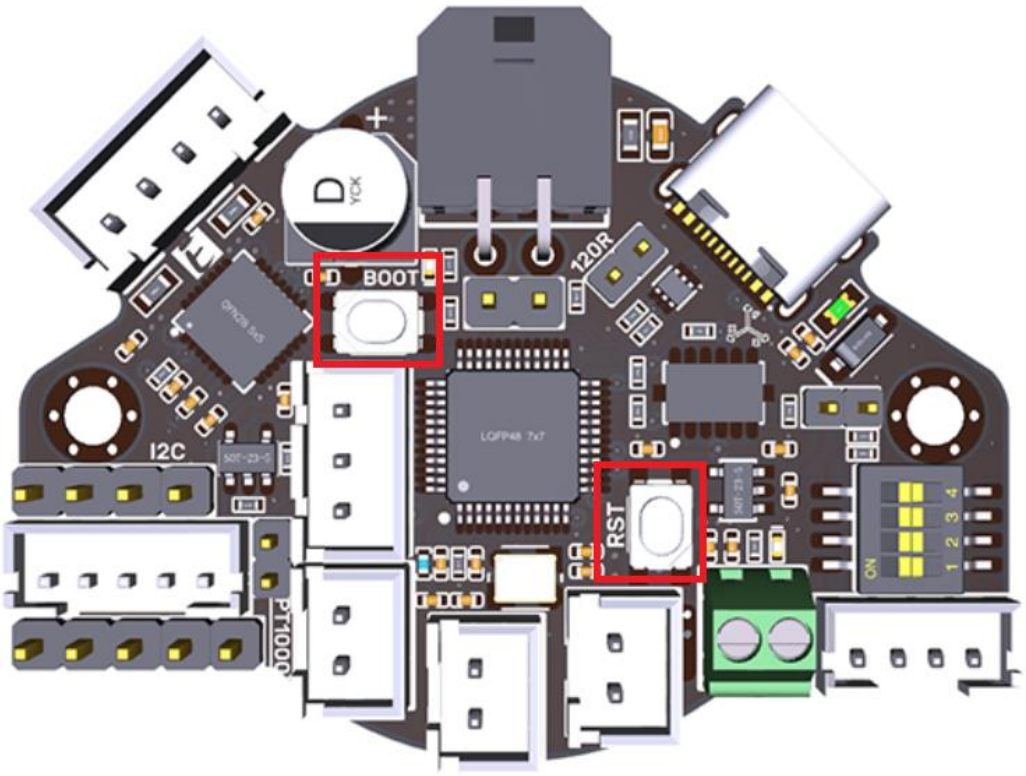
PA2 在 EBB36 CAN V1.1 和 EBB42 CAN V1.1 中被用于加热棒端口，进入 DFU 模式后的高电平会让加热棒处于加热状态，所以在使用 Type-C 端口的 DFU 更新固件时，请注意断开加热棒的主电源 Vin 或者确保固件很快更新完成，并进入正常工作的模式。万不可在主电源和加热棒都接好的情况下，使 MCU 长时间处于 DFU 模式。

使用 STM32CubeProgrammer 软件更新

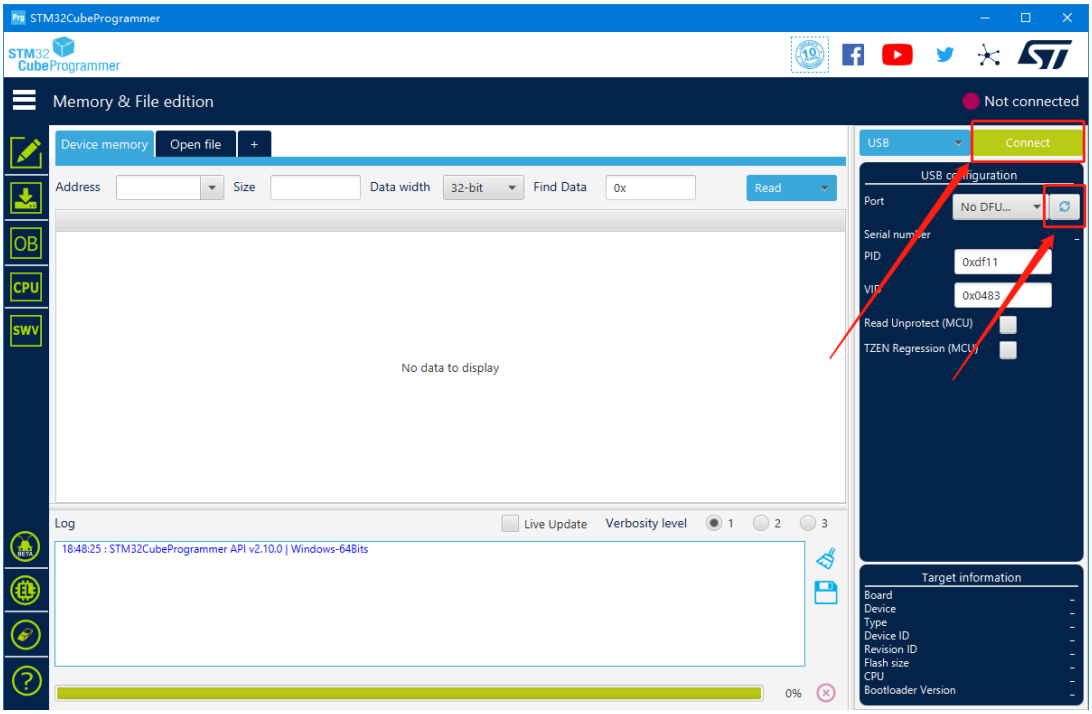
1. 打开安装好的 STM32CubeProgrammer 软件，选择要下载的固件(klipper.bin)



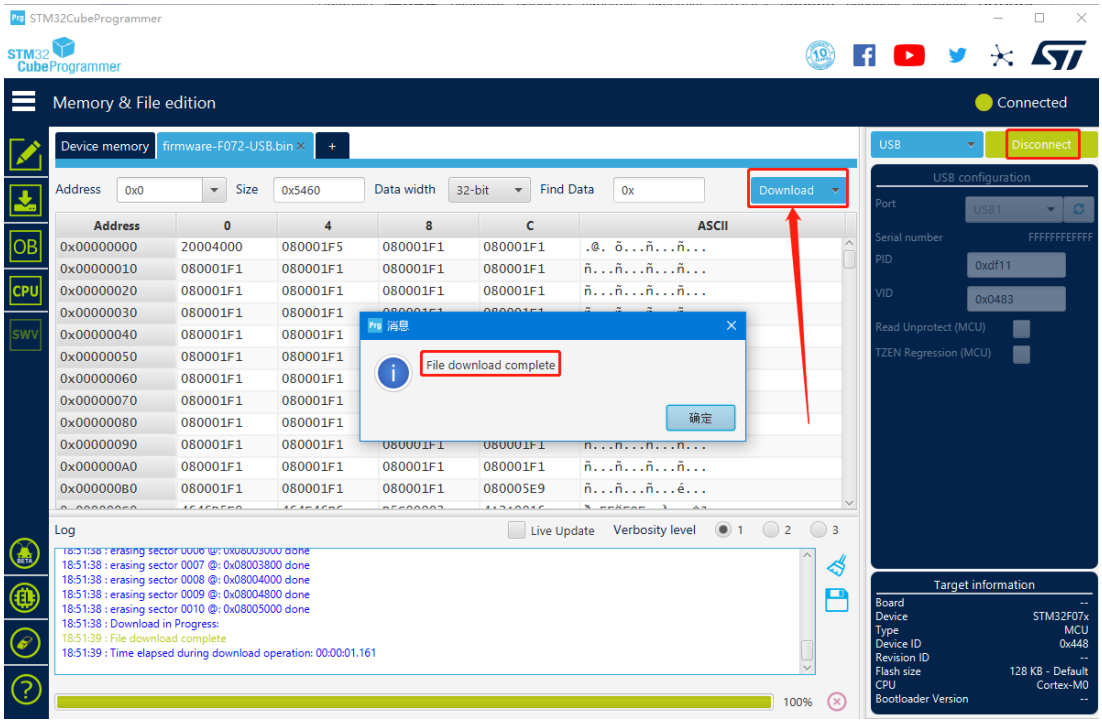
2. 按住 Boot 按钮，然后单击一下 RST 按钮进入 DFU 模式



3. 点击 STM32CubeProgrammer 软件中的“刷新”按钮，直到 Port 由”No DFU d…”变为”USB1”，然后点击 “Connect” 连接芯片

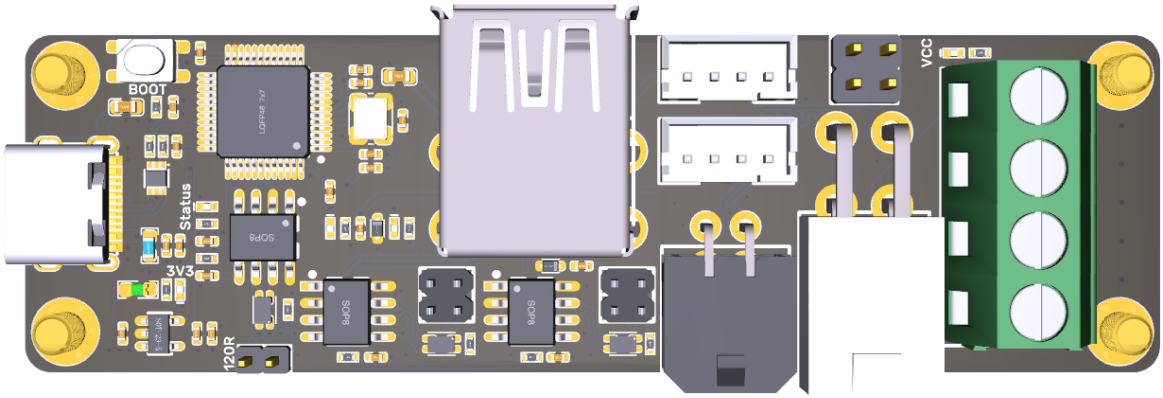


4. 连接成功后”Connect”会变成”Disconnet” ，然后点击”Download”开始下载程序，下载完成后会出现一个”File download complete”的弹窗，代表烧录成功



4.3 CANBus 配置

4.3.1 搭配 BIGTREETECH U2C 模块使用



1. 在 ssh 终端中输入 `sudo nano /etc/network/interfaces.d/can0` 命令并执行

```
auto can0
iface can0 can static
    bitrate 250000
    up ifconfig $IFACE txqueuelen 1024
```

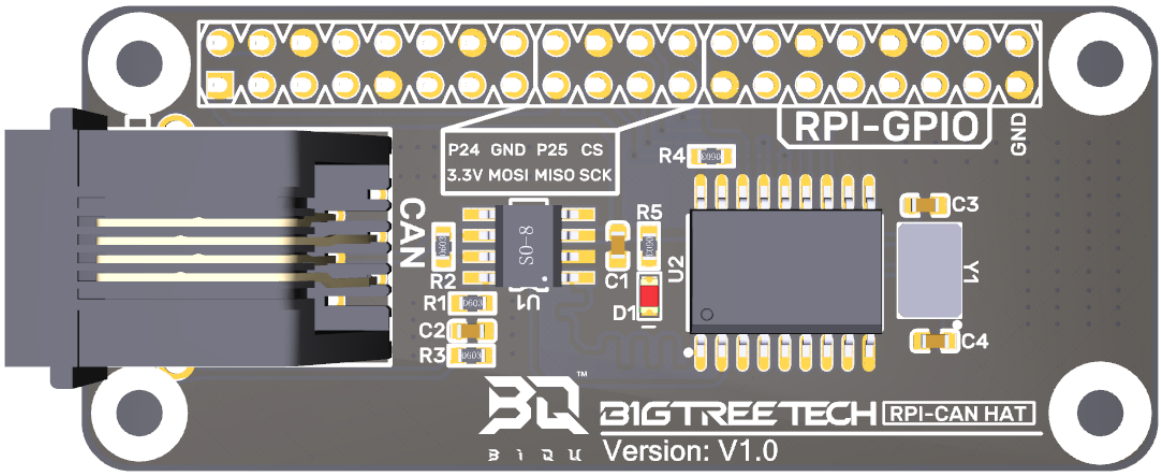
将 CANBus 速度设置为 250K（必须与固件中设置的速度一致 **(250000) CAN bus speed**），修改后保存（`Ctrl + S`）并退出（`Ctrl + X`），输入 `sudo reboot` 重启树莓派

2. CANBus 上的每个设备都会根据 MCU 的 UID 生成一个 canbus_uuid，要查找每个微控制器设备 ID，请确保硬件已通电并正确接线，然后运行：
```
~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
```

3. 如果检测到未初始化的 CAN 设备，上述命令将报告设备的 canbus_uuid：
```
Found canbus_uuid=0e0d81e4210c
```

4. 如果 Klipper 已经正常运行并且连接到此设备，那么 canbus_uuid 将不会被上报，此为正常现象

4.3.2 搭配 BIGTREETECH RPI-CAN HAT 模块使用



1. 输入并执行 `sudo nano /boot/config.txt`，然后在 config.txt 文件中添加以下内容
```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25,spimaxfrequency=1000000
```
修改后保存（Ctrl + S）并退出（Ctrl + X），输入 `sudo reboot` 重启树莓派

2. 输入并执行 `dmesg | grep -i '\(can\|spi\)'` 测试 RPI-CAN HAT 模块是否正常连接，正常的应答如下：
```
[ 8.680446] CAN device driver interface
[ 8.697558] mcp251x spi0.0 can0: MCP2515 successfully initialized.
[ 9.482332] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
```
```
pi@fluiddpi:~ $ dmesg | grep -i '\(can\|spi\)'
[    8.426216] CAN device driver interface
[    8.470380] mcp251x spi0.0 can0: MCP2515 successfully initialized.
[    9.330545] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
[   25.441341] can: controller area network core
[   25.467933] can: raw protocol
```

3. 在 ssh 终端中输入 `sudo nano /etc/network/interfaces.d/can0` 命令并执行
```
auto can0
iface can0 can static
    bitrate 250000
    up ifconfig $IFACE txqueuelen 1024
```
将 CANBus 速度设置为 250K（必须与固件中设置的速度一致 **(250000) CAN bus speed**），修改后保存（Ctrl + S）并退出（Ctrl + X），输入 `sudo reboot` 重启树莓派

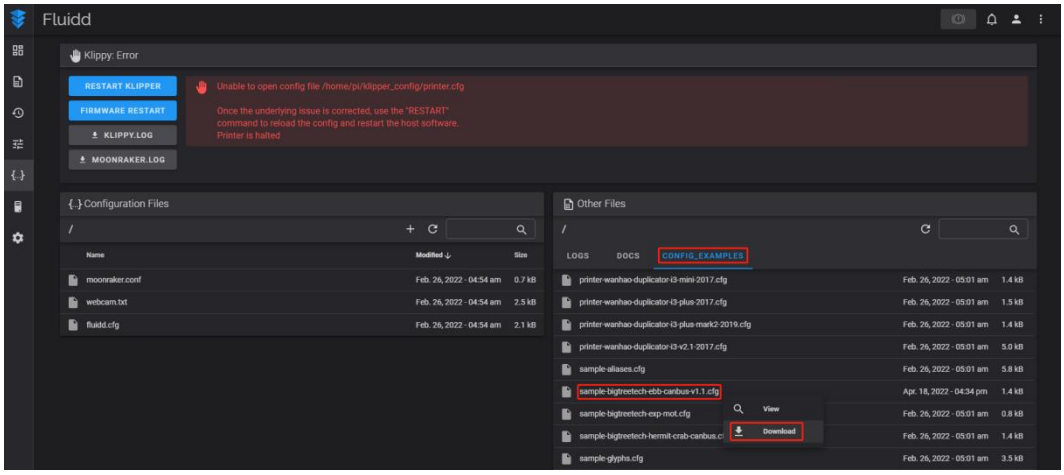
4. CANBus 上的每个设备都会根据 MCU 的 UID 生成一个 canbus_uuid，要查找每个微控制器设备 ID，请确保硬件已通电并正确接线，然后运行：
`~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`

5. 如果检测到未初始化的 CAN 设备，上述命令将报告设备的 canbus_uuid：
`Found canbus_uuid=0e0d81e4210c`

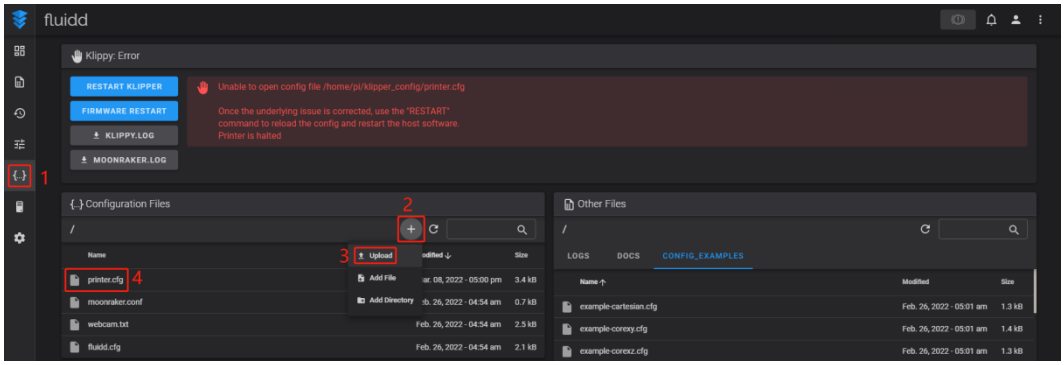
6. 如果 Klipper 已经正常运行并且连接到此设备，那么 canbus_uuid 将不会被上报，此为正常现象

4.4 配置 Klipper

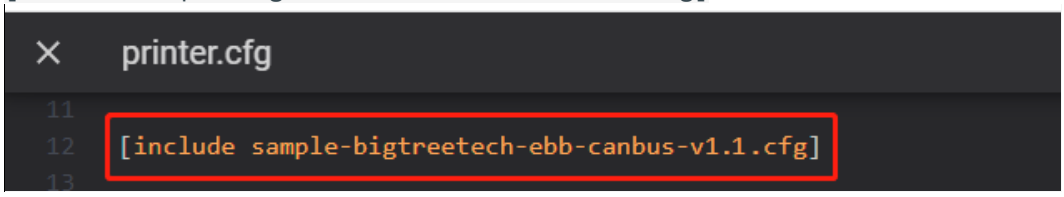
1. 在电脑的浏览器中输入树莓派的 IP 访问，如下图所示的路径中下载主板的参考配置，如果找不到此文件，请更新 Klipepr 固件源码到最新版本，或者到 github 下载 https://github.com/bigtreetech/EBB



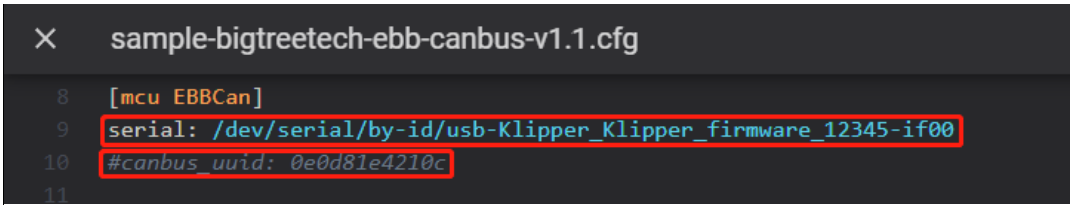
2. 将主板的配置文件上传到 Configuration Files 中



3. 并在“printer.cfg”文件中添加此主板的配置
`[include sample-bigtreetech-ebb-canbus-v1.1.cfg]`



4. 将配置文件中的 ID 号修改为主板实际的 ID（USB serial 或者 canbus）



5. 按照 https://www.klipper3d.org/Overview.html 的说明配置模块的具体功能

五、注意事项

1. THO 接口不使用 PT1000 时，不能往上面插跳线帽，否则 100K NTC 无法正常使用
2. 使用 CAN 通讯时，需要看是否用作终端，如果是终端，必须将 120R 位置插上跳线帽；
3. DIY 压线时，需注意线序，对照 Pin 图和原理图进行 DIY，避免电源线接反或者接到 CAN 信号中去，导致模块烧毁；
4. 通过 USB 端口烧录程序时，如果未外接电源，需将 VUSB 使用跳线帽短接，以便给模块提供工作电压；
5. 加热棒及风扇接口负载电流不得大于最大承受电流，以防烧坏 MOS 管。
6. 请格外注意 4.2 固件更新中的注意事项，避免主电源和加热棒都接好的情况下，使 MCU 长时间处于 DFU 模式。

六、FAQ

问：加热棒、风扇端口的最大电流

答：加热棒端口最大输出电流：5 A
风扇接口最大输出电流：1A
加热棒 + 驱动 + 风扇的总电流需小于 6A。

问：USB 接口无法更新固件

答：确保 VUSB 跳线帽有插入，主板上的电源指示灯正常亮起。