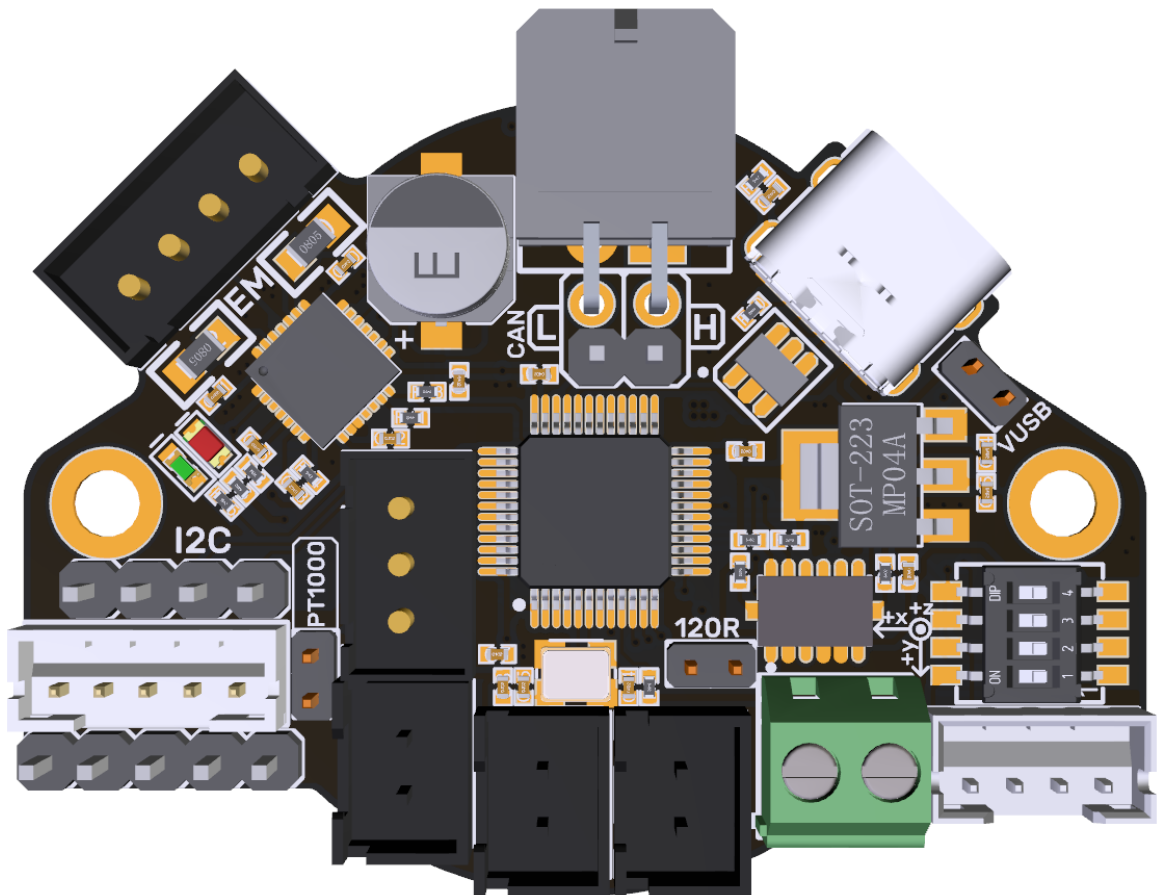


BIGTREE TECH

EBB36 CAN V1.0

使用说明



目录

目录	2
修订历史	3
一、产品简介	4
1.1 产品特点	4
1.2 产品参数	4
1.3 固件支持	5
1.4 产品尺寸	5
二、外设接口	6
2.1 Pin 脚说明	6
三、接口介绍	6
3.1 USB 供电	6
3.2 100K NTC 或 PT1000 设置	7
3.3 BL-Touch 接线	9
3.4 断料检测接线	9
3.5 RGB 接线	10
四、Klipper	11
4.1 编译固件	11
4.2 固件更新	12
4.3 CANBus 配置	14
4.3.1 搭配 BIGTREETECH U2C 模块使用	14
4.3.2 搭配 BIGTREETECH RPI-CAN HAT 模块使用	15
4.4 配置 Klipper	17
五、注意事项	18
六、FAQ	18

修订历史

版本	修改说明	日期
01.00	初稿	2022/04/18

一、产品简介

BIGTREETECH EBB36 CAN V1.0 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. 加热棒及风扇端口增加防反激二极管，有效保护 MOS 管不被反向电压烧毁
7. 电源接口有防反接保护，避免客户在 DIY 时接反电源线导致板子烧毁
8. 板载 MAX31865，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. 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

1.2 产品参数

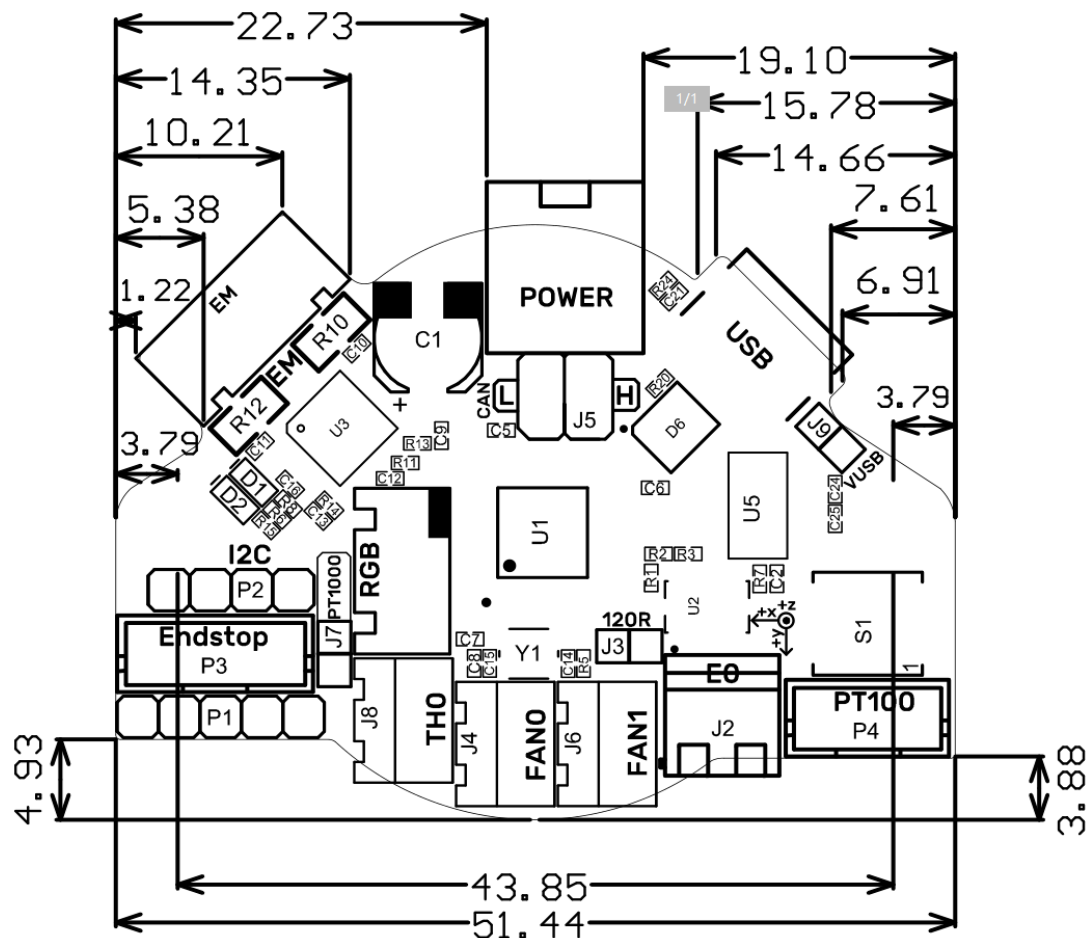
1. 外观尺寸：51.5mm*37mm 详情请参考：BIGTREETECH EBB36 CAN V1.0-SIZE. pdf
2. 安装尺寸：孔间距 43.85mm，M3 螺丝孔*2
3. 微处理器：ARM Cortex-M0 STM32F072C8T6 48MHz
4. 输入电压：DC12V-DC24V 6A
5. 逻辑电压：DC 3.3V
6. 加热接口：加热棒（E0），最大输出电流：5A
7. 板载传感器：ADXL345
8. 风扇接口：两个数控风扇（FAN0，FAN1）

9. 风扇接口最大输出电流：1A，峰值 1.5A
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
12. 驱动工作模式：UART
13. 步进电机接口：EM
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. DCDC 5V 输出最大电流：1A

1.3 固件支持

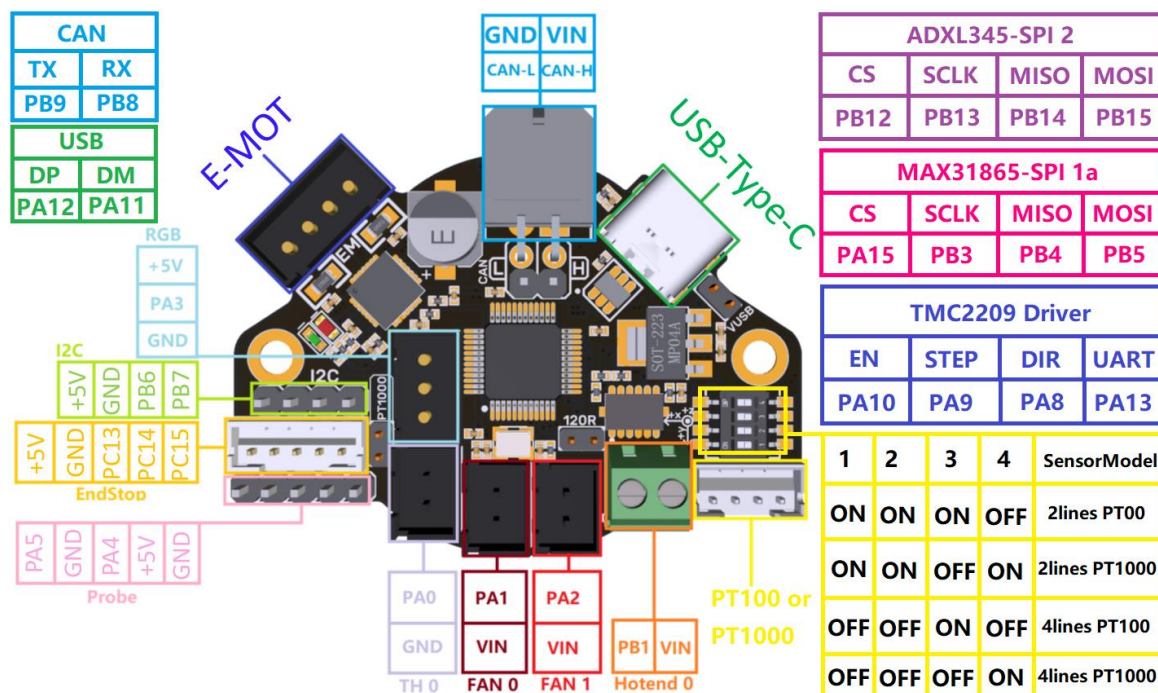
此产品当前仅支持 Klipper 固件

1.4 产品尺寸



二、外设接口

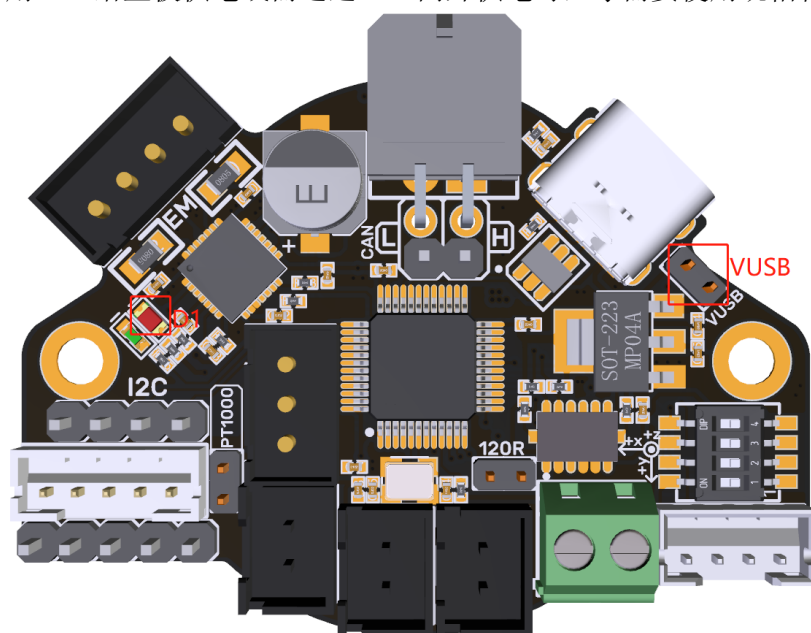
2.1 Pin 脚说明



三、接口介绍

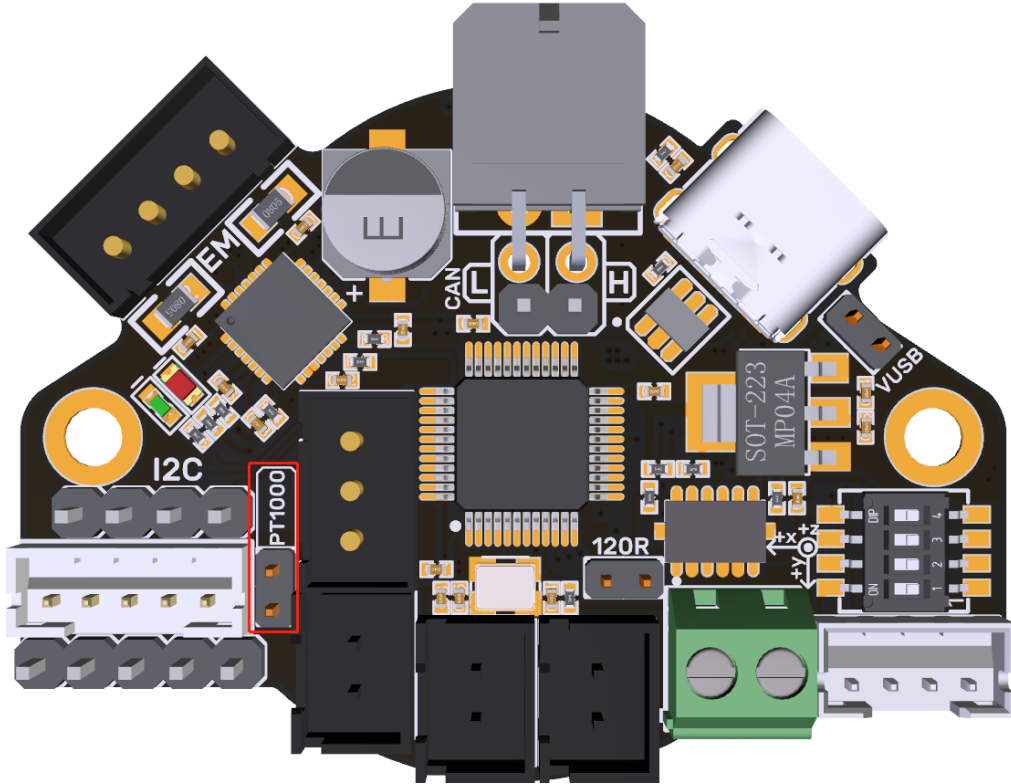
3.1 USB 供电

主板上电之后，D1 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 VUSB 短接。



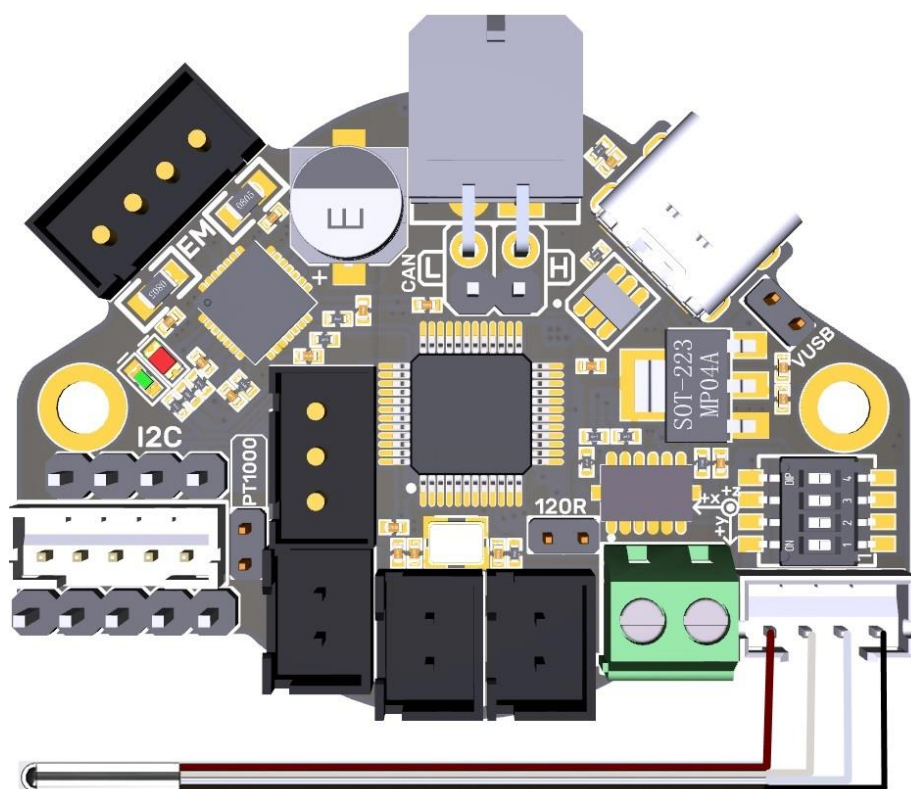
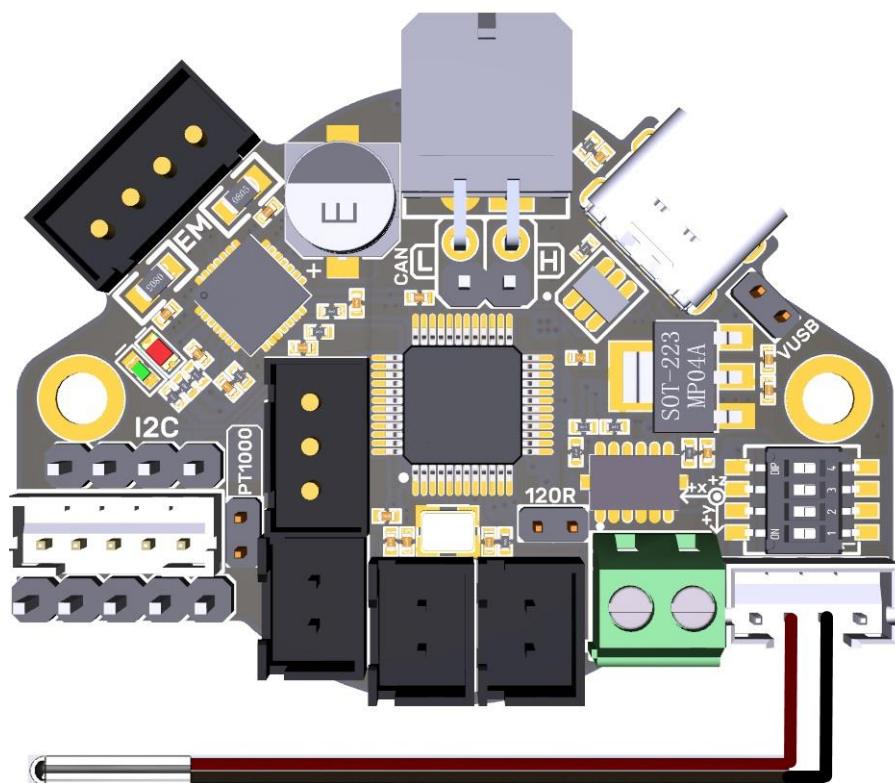
3.2 100K NTC 或 PT1000 设置

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）。

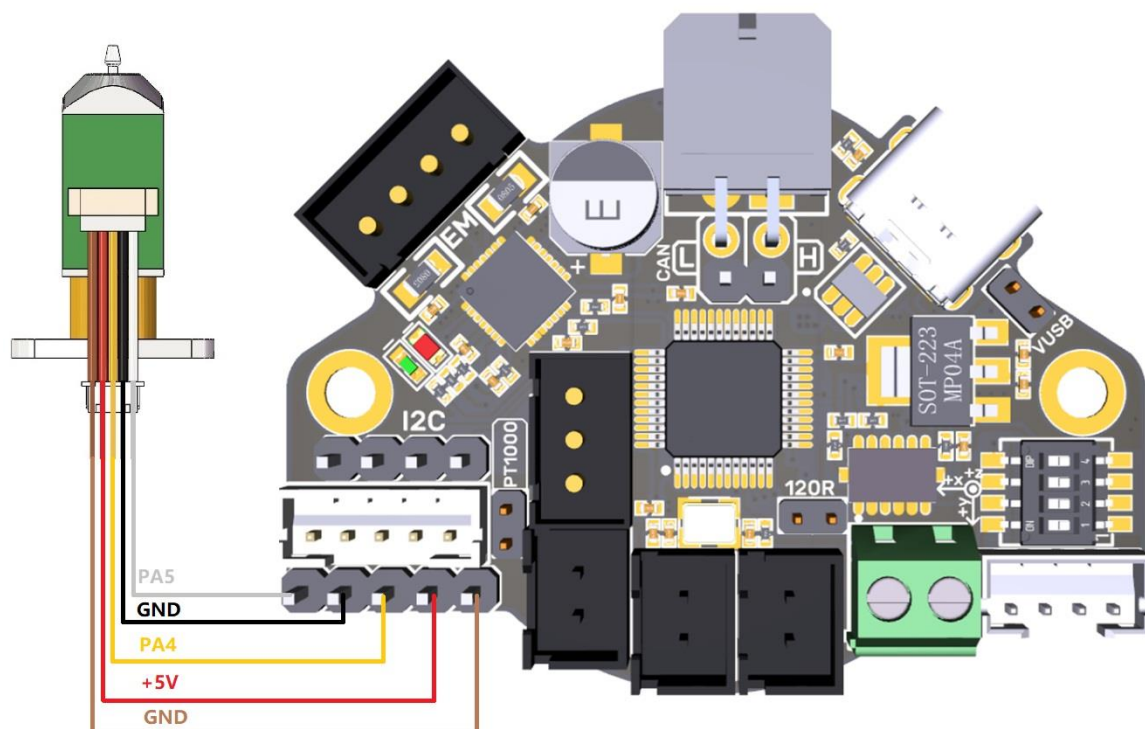


2. 带 31865 版本：通过拨码开关进行选择 PT100/PT1000，两线或者四线：

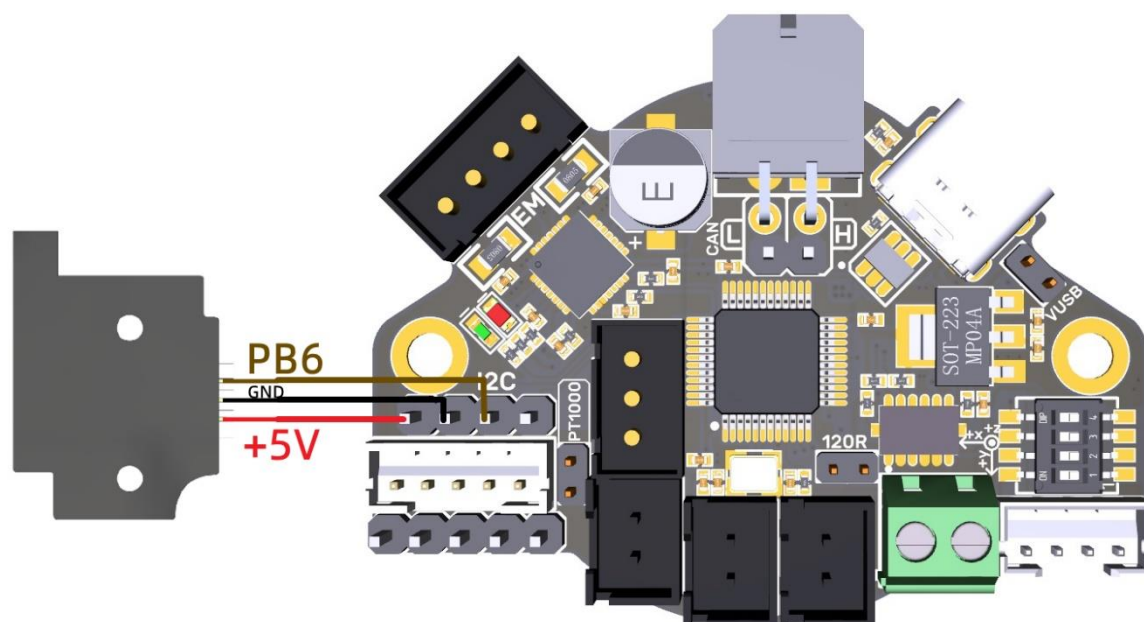
1	2	3	4	Sensor Model
ON	ON	ON	OFF	Two lines PT100
ON	ON	OFF	ON	Two linesPT1000
OFF	OFF	ON	OFF	Four-wire PT100
OFF	OFF	OFF	ON	Four-wire PT1000

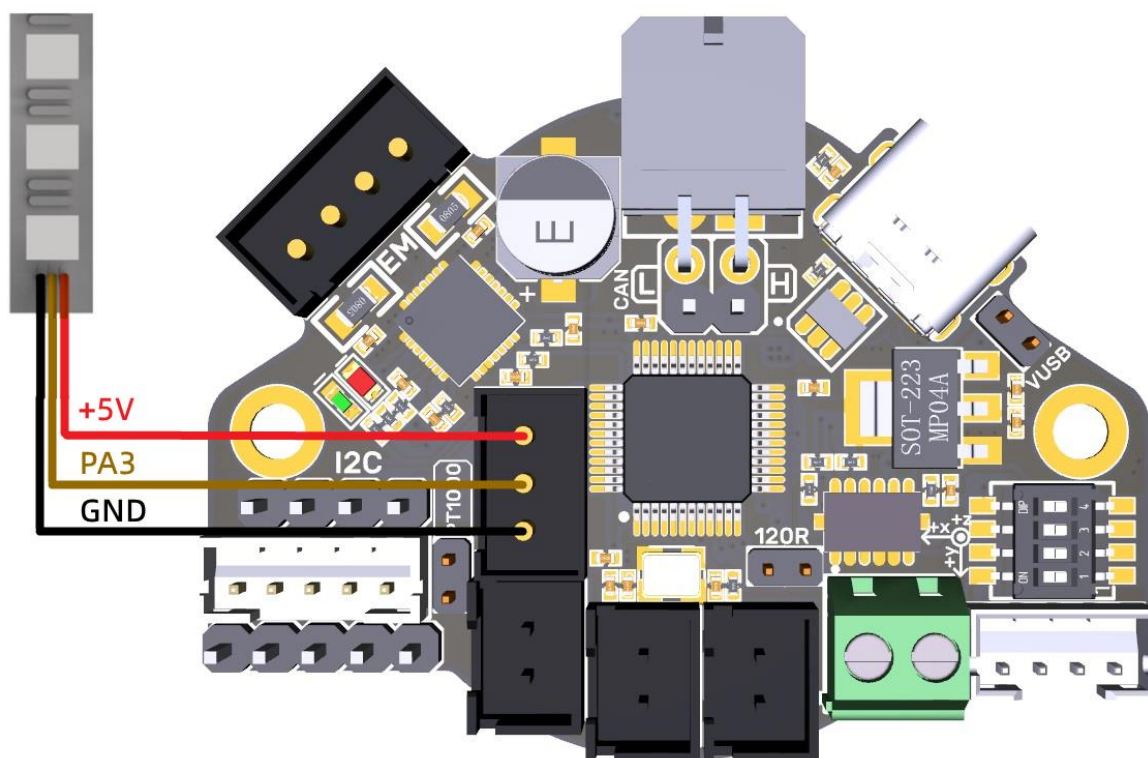


3.3 BL-Touch 接线



3.4 断料检测接线





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 (STM32F072) -->

Bootloader offset (No bootloader) -->

Clock Reference (8 MHz crystal) -->

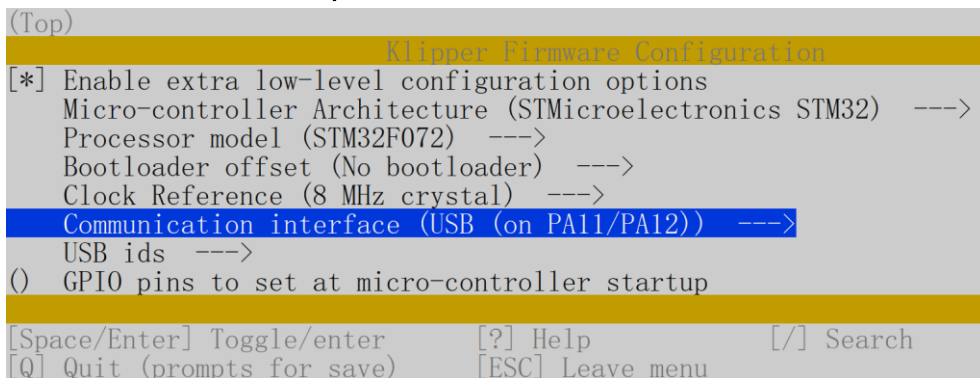
如果使用 Type-C 上的 USB 通信

Communication interface (USB (on PA11/PA12)) -->

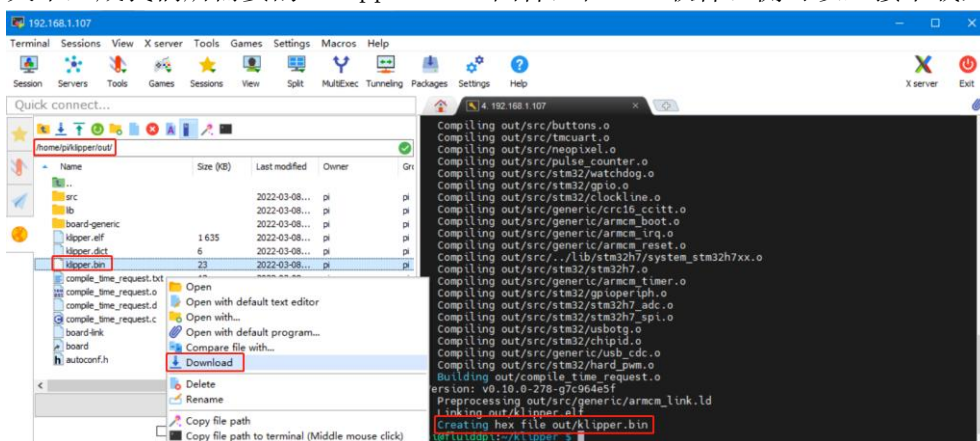
如果使用 CANBus 通信

Communication interface (CAN bus (on PB8/PB9)) -->

(250000) CAN bus speed



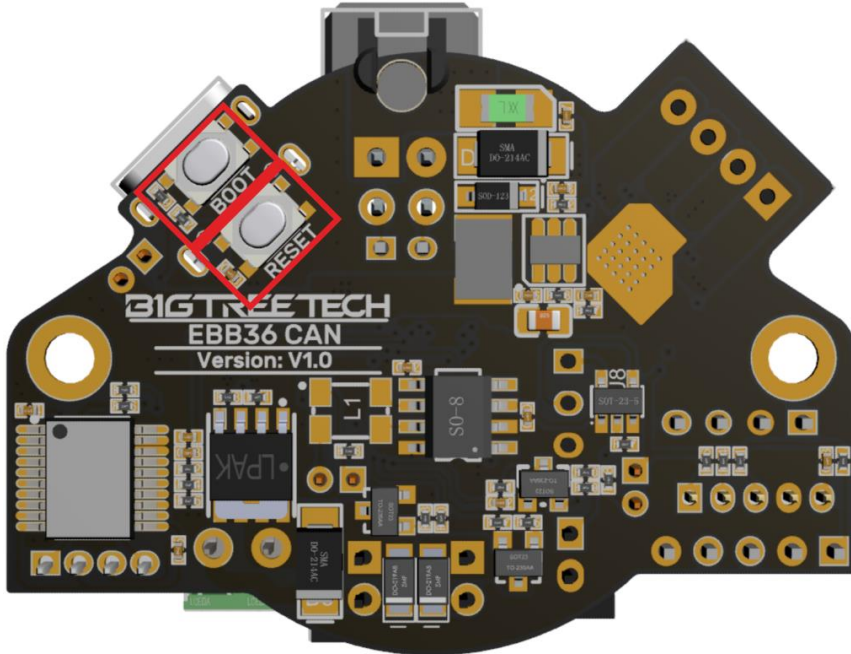
2. 配置选择完成后，输入 `q` 退出配置界面，当询问是否保存配置是选择 “Yes”
3. 输入 `make` 编译固件，当 `make` 执行完成后会在树莓派的 `home/pi/klipper/out` 文件夹中生成我们所需要的 `klipper.bin` 固件，在 ssh 软件左侧可以直接下载到电脑中



4.2 固件更新

通过树莓派直接更新（也可插到电脑上，进入 DFU 模式后，使用 STM32CubeProgrammer 软件更新）

1. 按住 Boot 按钮，然后单击一下 Reset 按钮进入 DFU 模式



2. 在 ssh 终端命令行中输入 `lsusb` 查询 DFU 设备 ID

```
pi@fluidpi:~$ lsusb
Bus 001 Device 005: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
Bus 001 Device 004: ID 1d50:6061 OpenMoko, Inc. Geschwister Schneider CAN adapter
Bus 001 Device 003: ID 0424:ec00 Microchip Technology, Inc. (formerly SMSC) SMC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Microchip Technology, Inc. (formerly SMSC) SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@fluidpi:~$
```

3. 输入 `make flash FLASH_DEVICE=0483:df11` 开始烧录固件（注意：将 0483:df11 更换为上一步中查询到的实际的设备 ID）

```
pi@fluidpi:~/klipper $ make flash FLASH_DEVICE=0483:df11
Building hid-flash
hid-flash requires libusb-1.0, please install with:
sudo apt-get install libusb-1.0
Flashing out/klipper.bin to 0483:df11
sudo dfu-util -d ,0483:df11 -R -a 0 -s 0x8000000:leave -D out/klipper.bin

[sudo] password for pi:
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 = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash "
Downloading to address = 0x08000000, size = 21728
Download [=====] 100% 21728 bytes
Download done
File downloaded successfully
Transitioning to dfuMANIFEST state
dfu-util: can't detach
Resetting USB to switch back to runtime mode
pi@fluidpi:~/klipper $
```

4. 固件烧录完成后，输入 `ls /dev/serial/by-id/` 查询设备的 Serial ID（只有通过 USB 通信的方式才会有此 ID，CANBus 方式忽略此步骤）

```
pi@fluidpi:~/klipper $ ls /dev/serial/by-id/
usb-Klipper_stm32f072xb_28002D001557434338313020-if00
pi@fluidpi:~/klipper $
```


5. 如果使用 USB 通信，第一次烧录完成之后，再次更新时无需手动按 Boot 和 Reset 按钮进入 DFU 模式，可以直接输入

```
make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32f072xb_28002D001557434338313020-if00
```

烧录固件（注意：将/dev/serial/by-id/xxx 更换为上一步中查询到的实际的 ID）

```
pi@fluidpi:~/klipper $ ls /dev/serial/by-id/
usb-Klipper_stm32f072xb_28002D001557434338313020-if00
pi@fluidpi:~/klipper $ make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32f072xb_28002D001557434338313020-if00
Building hid-flash
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_stm32f072xb_28002D001557434338313020-if00
Entering bootloader on /dev/serial/by-id/usb-Klipper_stm32f072xb_28002D001557434338313020-if00
Device reconnect on /sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.4:1.0
sudo dfu-util -p 1-1.4 -R -a 0 -s 0x80000000:leave -D out/klipper.bin

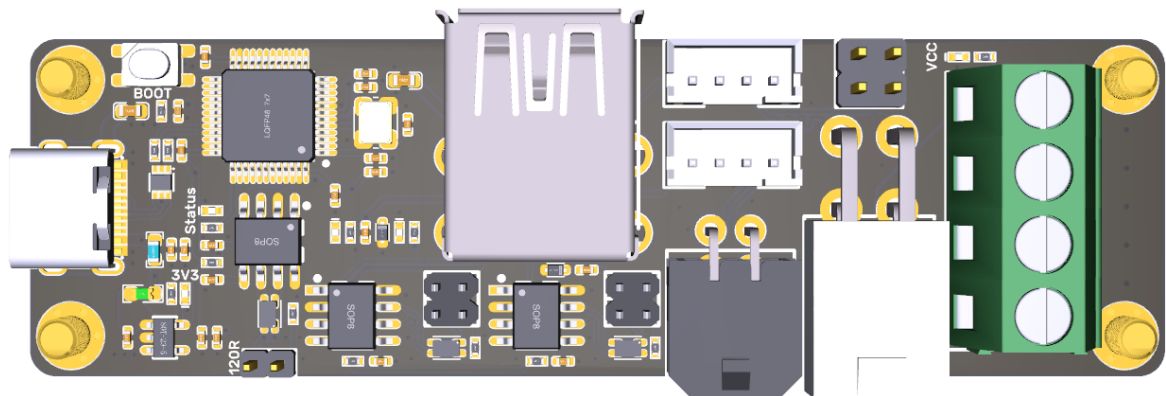
dfu-util 0.9

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 = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash "
Downloading to address = 0x08000000, size = 21728
Download [=====] 100% 21728 bytes
Download done.
File downloaded successfully
Transitioning to dfuMANIFEST state
dfu-util: can't detach
Resetting USB to switch back to runtime mode
pi@fluidpi:~/klipper $
```

4.3 CANBus 配置

4.3.1 搭配 BIGTREETECH U2C 模块使用



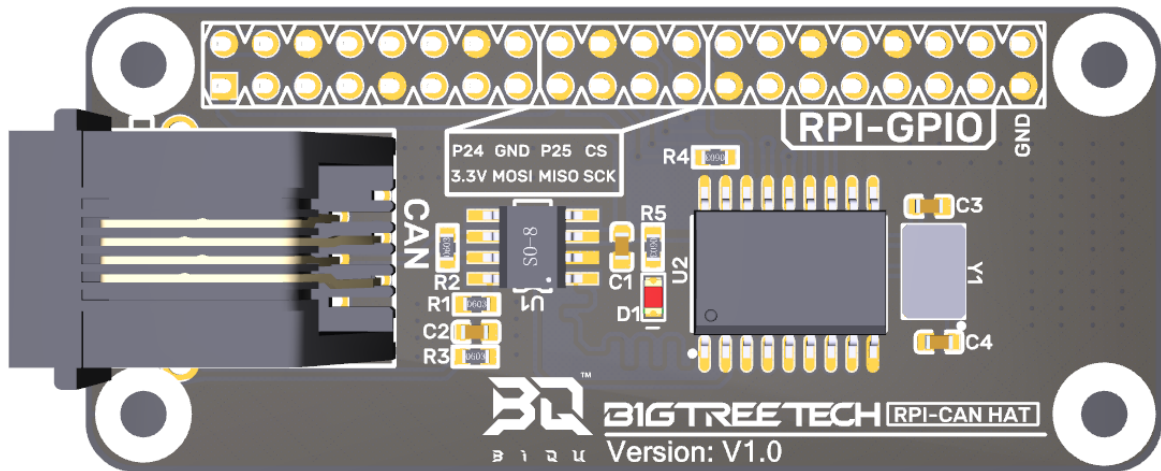
1. 在 ssh 终端中输入 `sudo nano /etc/network/interfaces.d/can0` 命令并执行


```
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
`dtoverlay=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
```

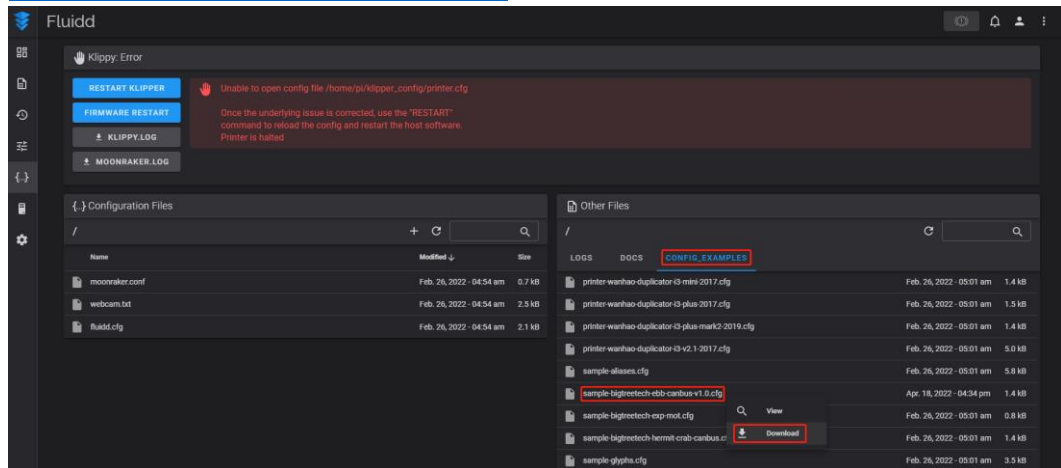

`pi@fluidpi:~$ dmesg | grep -i '\(can\|spi\)'`

```
[ 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

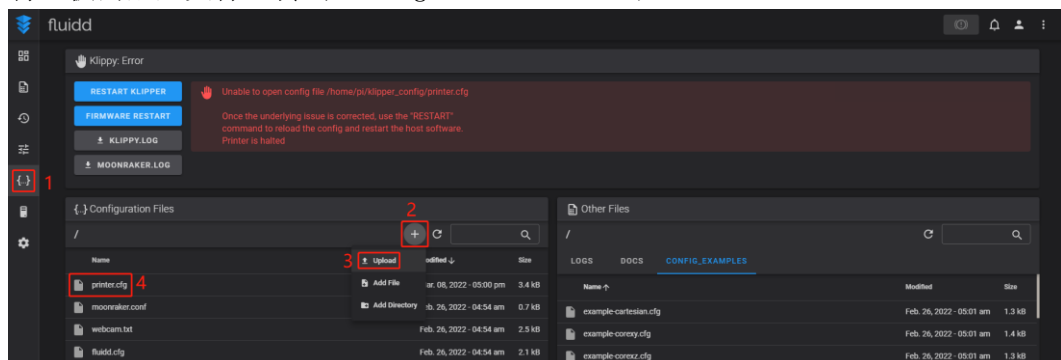
4. CANBus 上的每个设备都会根据 MCU 的 UID 生成一个 `canbus_uuid`，要查找每个微控制器设备 ID，请确保硬件已通电并正确接线，然后运行：
`~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`
5. 如果检测到未初始化的 CAN 设备，上述命令将报告设备的 `canbus_uuid`：
`Found canbus_uuid=0e0d81e4210c`
6. 如果 Klipper 已经正常运行并且连接到此设备，那么 `canbus_uuid` 将不会被上报，此为正常现象

4.4 配置 Klipper

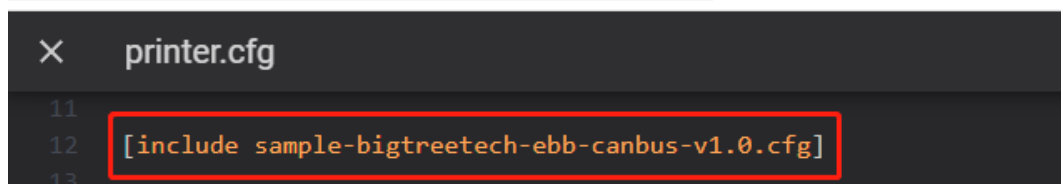
1. 在电脑的浏览器中输入树莓派的 IP 访问，如下图所示的路径中下载主板的参考配置，如果找不到此文件，请更新 Klipper 固件源码到最新版本，或者到 github 下载 <https://github.com/bigtreotech/EBB>



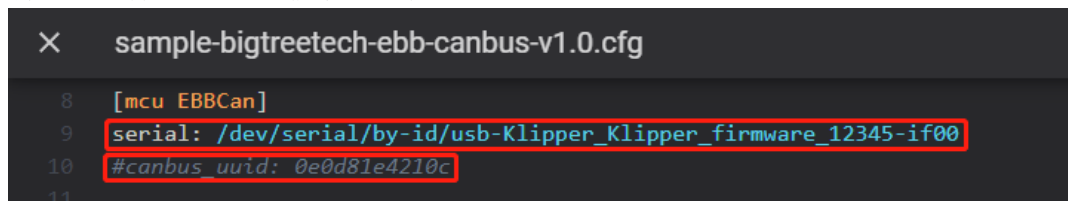
2. 将主板的配置文件上传到 Configuration Files 中



3. 并在“printer.cfg”文件中添加此主板的配置
`[include sample-bigtreotech-ebb-canbus-v1.0.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. TH0 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。

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，峰值 1.5A

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。