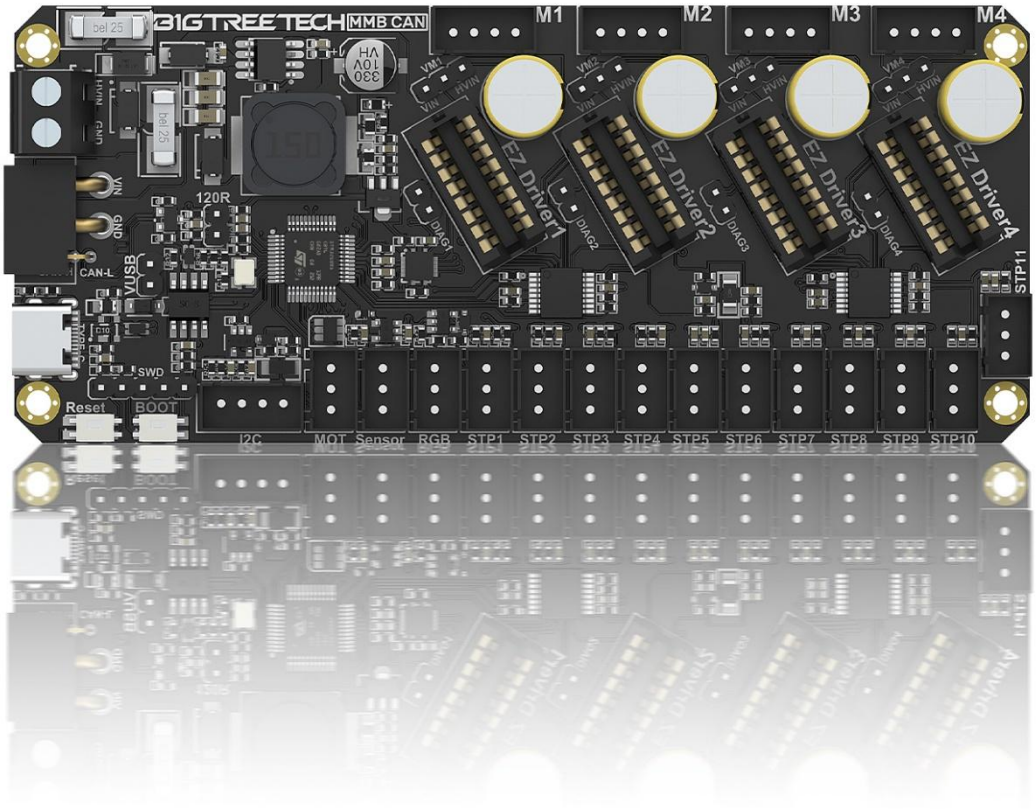
BIGTREETECH

# MMB CAN V1.0

## 用户手册



# 修订历史

| 版本 | 日期 | 修改说明 |
| --- | --- | --- |
| v1.00 | 2023/8/23 | 初稿 |

# 目录

# 一、产品简介

BIGTREETECH MMB CAN V1.0 是深圳市必趣创新科技有限公司 3D 打印团队针对多色挤出机制作的挤出控制板，可以通过 USB 或者 CAN 进行通讯，大大简化接线。

## 1.1 产品特点

- 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
- 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
- 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
- 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
- USB 口增设 ESD 保护芯片，防止主控被 USB 口静电击穿
- 采用艾迈斯接口进行 CAN 通讯及主板供电，让接线简单化
- 步进电机驱动支持高低压选择，方便客户 DIY 使用

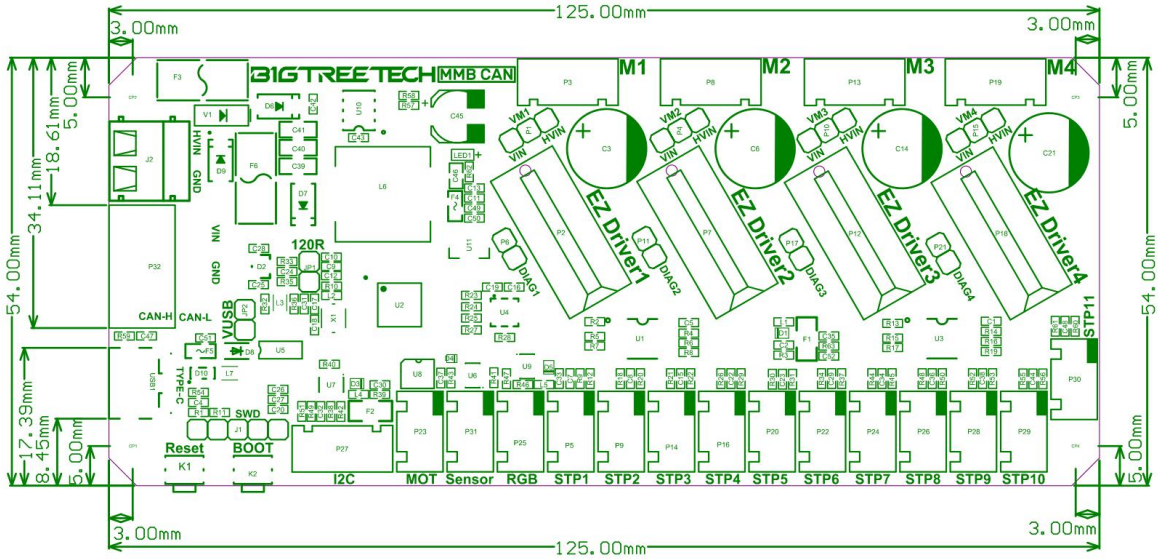
## 1.2 产品参数

| | |
|---|---|
| 外观尺寸 | 125mm x 54mm |
| 安装尺寸 | 详情请参考：BIGTREETECH MMB CAN V1.0-SIZE.pdf |
| 微处理器 | ARM Cortex-M0+ STM32G0B1CBT6 64MHz |
| 输入电压 | DC12V-DC24V 9A |
| 逻辑电压 | DC 3.3V |
| 舵机接口（MOT）最大输出 | 5V 2A，峰值 2.5A |
| 拓展接口 | STP1-STP11，I2C，RGB，Sensor（红外传感器接口），USB 接口，CAN 接口 |
| 电机驱动支持 | EZ 系列驱动（支持电压选择） |
| 驱动工作模式 | STEP/DIR、UART、SPI |
| 步进电机接口 | M1、M2、M3、M4 |
| USB 通信接口 | USB Type-C |
| DCDC 5V 输出最大电流 | 3.6A |

## 1.3 固件支持

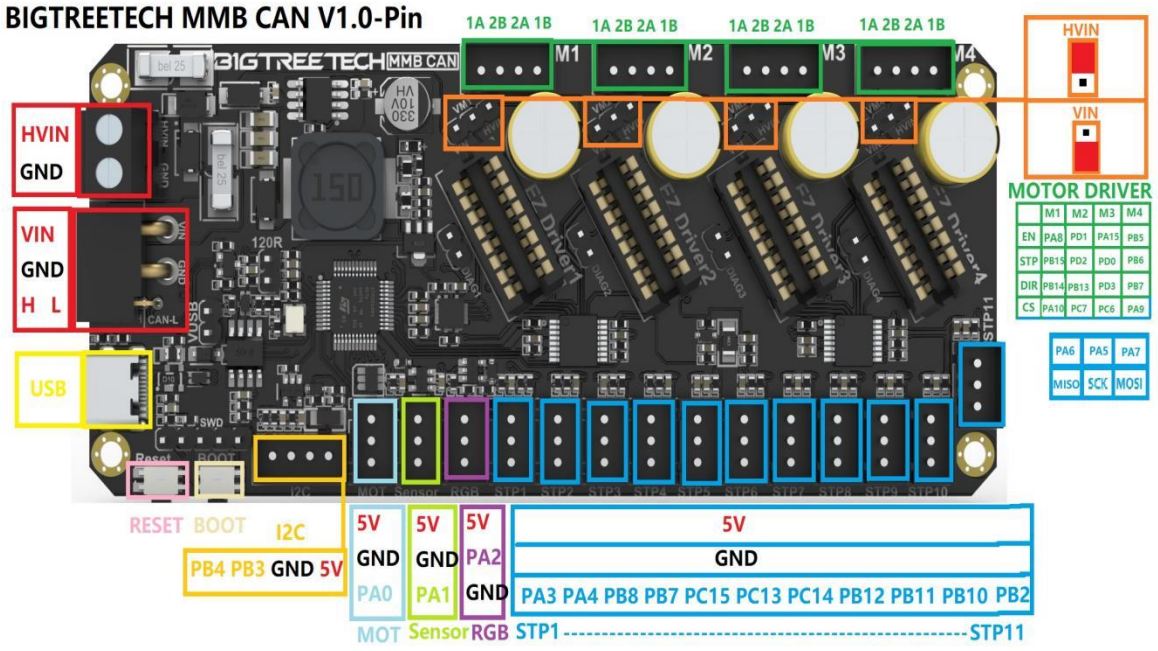
此产品当前仅支持 Klipper 固件

## 1.4 产品尺寸




# 二、外设接口

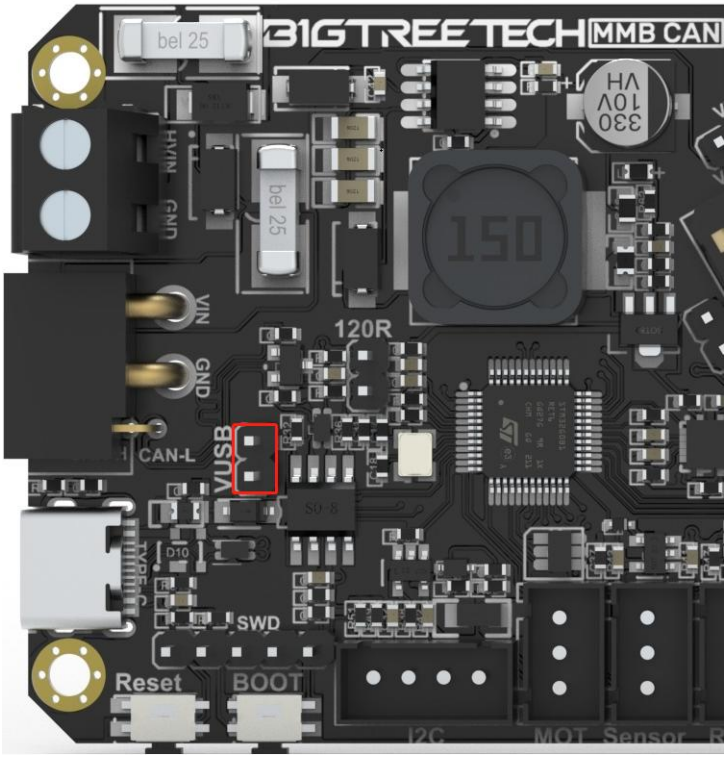
## 2.1 Pin 脚说明



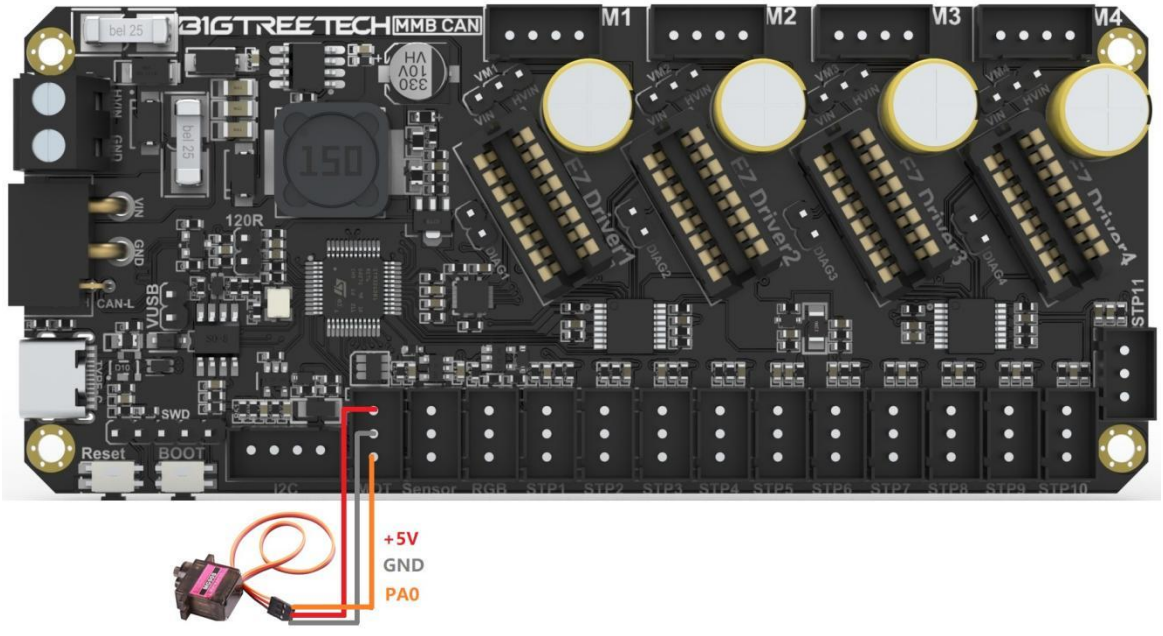
# 三、接口介绍

## 3.1 USB 供电

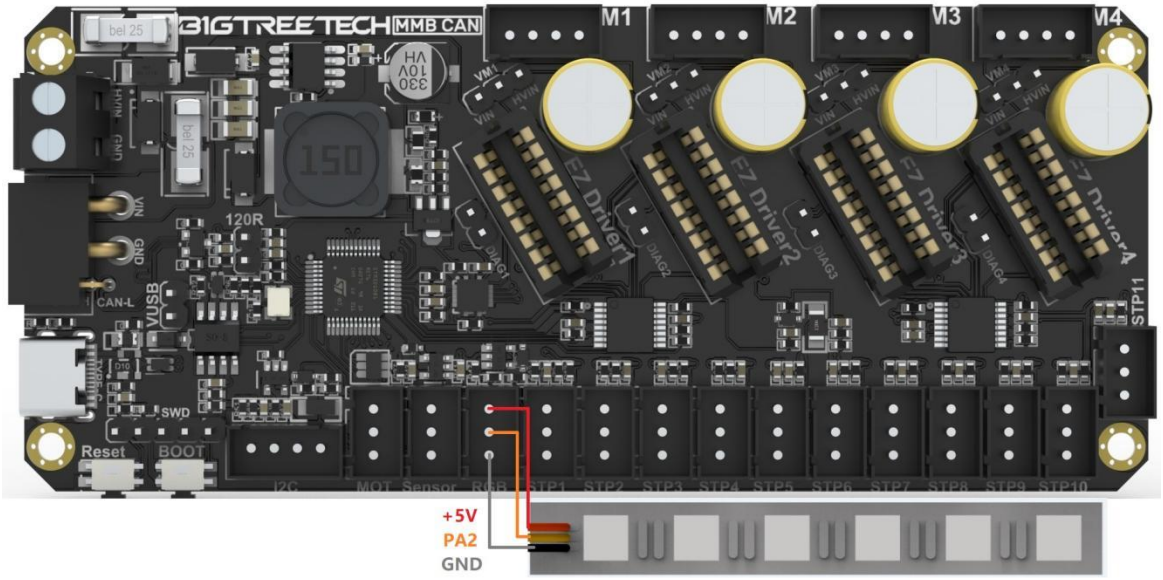
主板上电之后， 电源灯会亮起，表示供电正常。板上标识 VUSB 是电源选择端，仅当使用 USB 给主板供电时，才需要使用跳帽将 VUSB 短接。



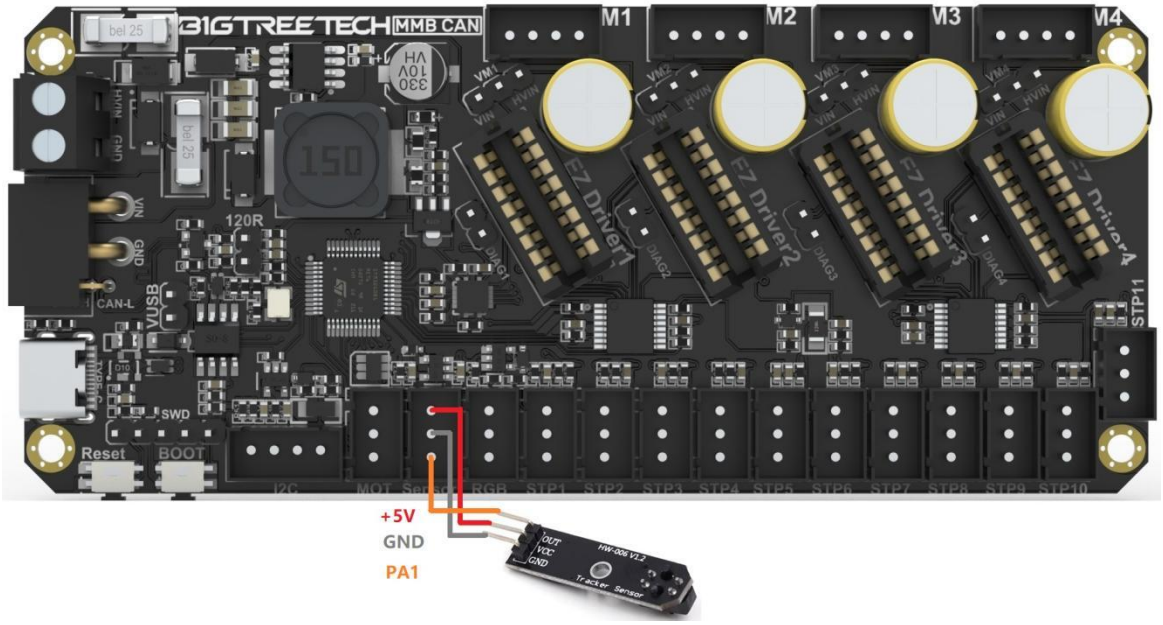

## 3.2 Servo 接线



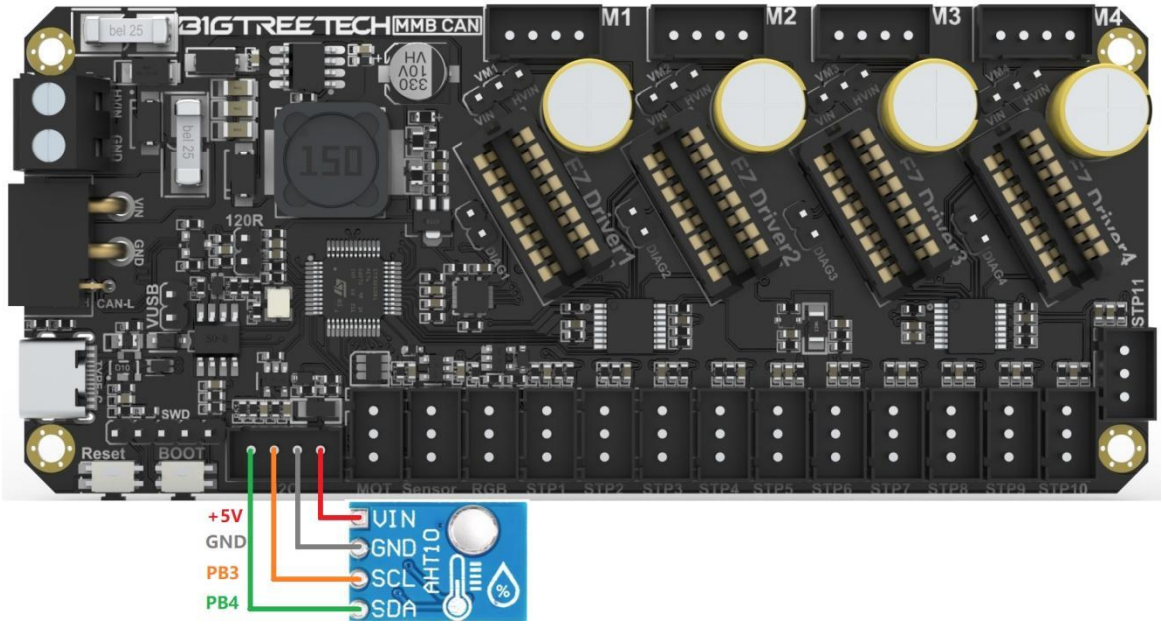
## 3.3 RGB-WS2812 接线



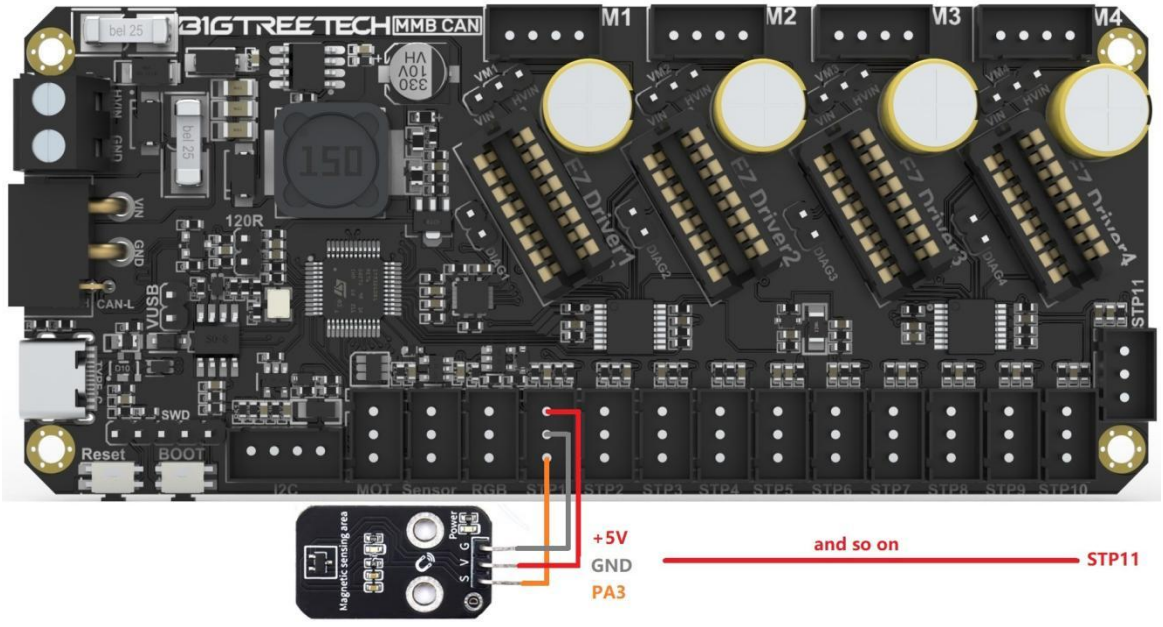

## 3.4 Sensor（如 CRT5000 红外传感器）接线



## 3.5 I2C（如 AHT10 温湿度传感器）接线




## 3.6 Endstop（如霍尔传感器）接线



# 四、Klipper 固件

## 4.1 烧录 CANBOOT

注意：CanBoot 旨在通过 CAN bus 接口直接更新 MCU 固件，若您更倾向于使用 DFU 更新方法，请跳过此步骤。

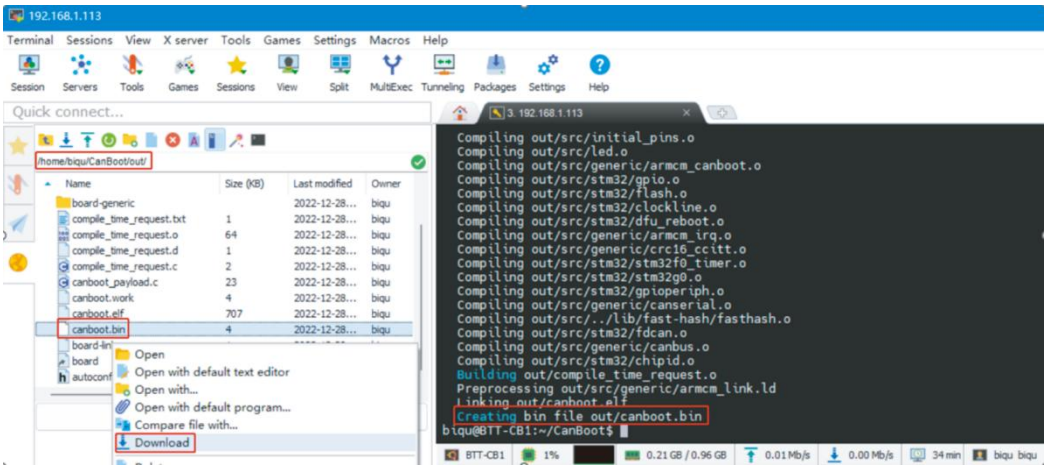
“树莓派或 CB1 烧录 CanBoot”，参考此处说明下载 CanBoot 工程
https://github.com/Arksine/CanBoot

1. 输入
   **cd ~**
   跳转到主目录，输入
   **git clone https://github.com/Arksine/CanBoot**
   下载 CanBoot 工程，然后输入
   **cd CanBoot**
   跳转到 CanBoot 目录中。

2. 输入
   **make menuconfig**
   并按照下图配置

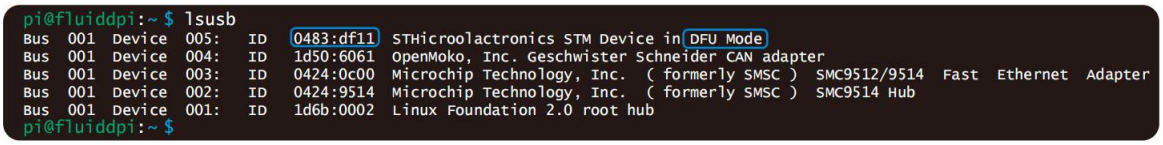
```
(Top)
                        Katapult Configuration v0.0.1-57-gabd1545
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Build Katapult deployment application (Do not build)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PB0/PB1))  --->
    Application start offset (8KiB offset)  --->
(1000000) CAN bus speed
()  GPIO pins to set on bootloader entry
[*] Support bootloader entry on rapid double click of reset button
[ ] Enable bootloader entry on button (or gpio) state
[ ] Enable Status LED

[Space/Enter] Toggle/enter      [?] Help            [/] Search
[Q] Quit (prompts for save)     [ESC] Leave menu
```

3. 输入 make 编译固件，当 make 执行完成后会在 home/biqu/CanBoot/out 文件夹中生成我们所需要的“canboot.bin”固件，在 SSH 软件左侧可以直接下载到电脑中；




4. 请按住 Boot 按钮，然后使用 Type-C 线连接至树莓派/CB1，此时芯片进入 DFU 模式

5. 在 SSH 终端命令行中输入
   lsusb
   查询 DFU 设备 ID




6. 请输入以下命令以烧录 CanBoot
   make flash FLASH_DEVICE=0483:df11
   其中“0483:df11”需替换为上一步中查询到的实际设备 ID

7. 烧录完成后，请拔下 Type-C 数据线。

## 4.2 编译 Klipper 固件

1. SSH 连接到 CB1/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)；

```
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    USB ids  --->
()  GPIO pins to set at micro-controller startup

[Space/Enter] Toggle/enter      [?] Help            [/] Search
[Q] Quit (prompts for save)     [ESC] Leave menu
```

[*] Enable extra low-level configuration optionsMicro-controller
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32G0B1) --->

如果不使用 CanBoot

Bootloader offset (No bootloader) --->

如果使用 CanBoot

Bootloader offset (8KiB bootloader) --->

如果使用 Type-C 上的 USB 通信

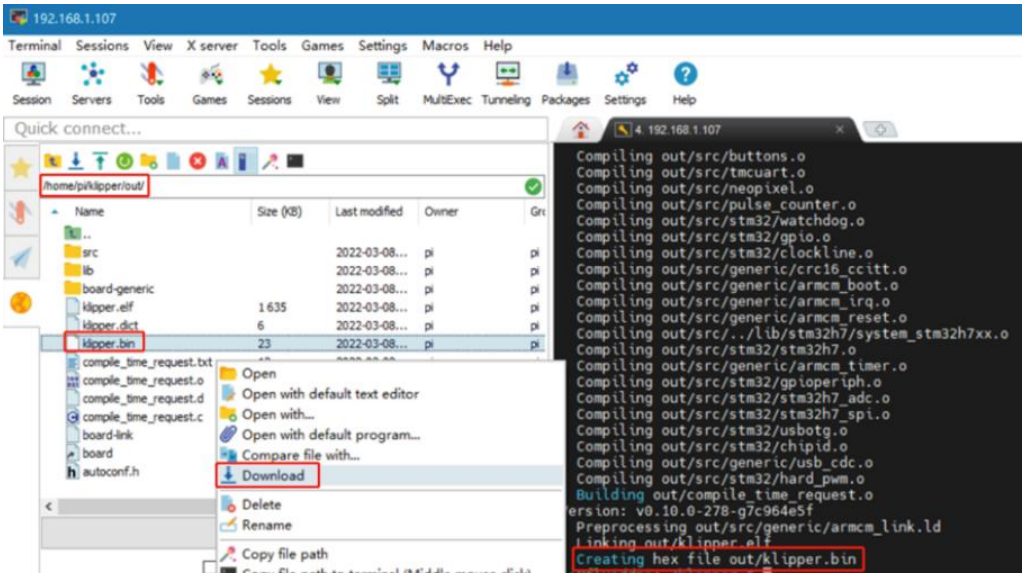
Communication interface (USB (on PA11/PA12)) --->

如果使用 CANBus 通信

Communication interface (CAN bus (on PB0/PB1)) --->
(1000000) 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时选择“Yes”；

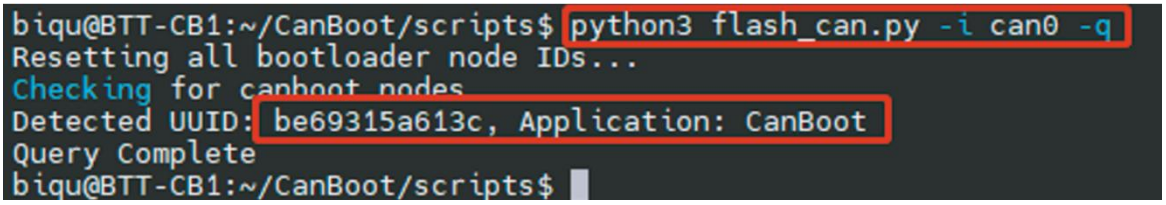
3. 输入 make 编译固件，当 make 执行完成后会在 home/pi/klipper/out 文件夹中生成我们所需要的“klipper.bin”固件，在 SSH 软件左侧可以直接下载到电脑中



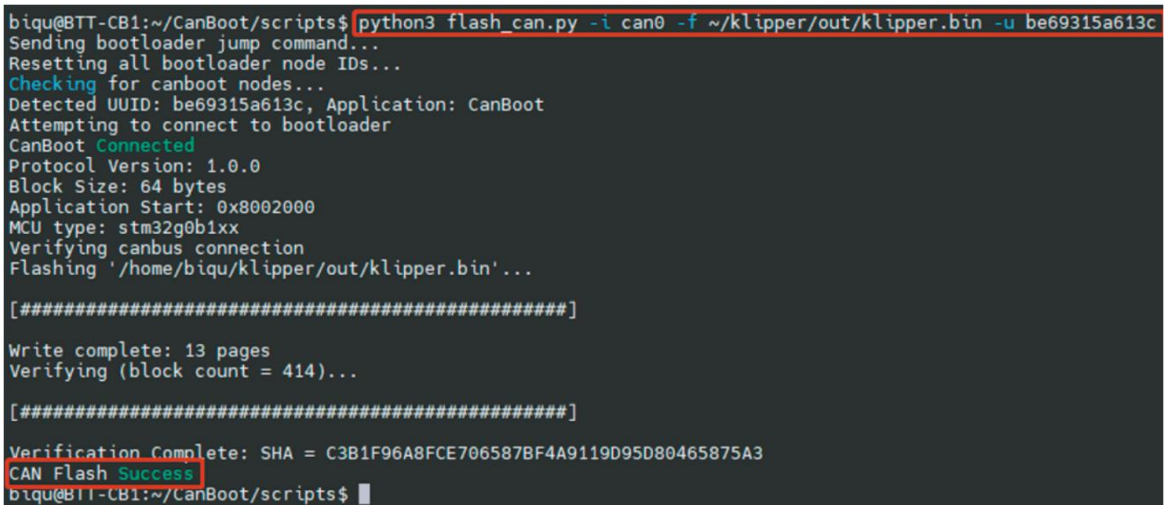
## 4.3 通过 CANBOOT 进行固件更新

1. 使用 CAN bus 需要接好 CAN bus 线缆以及插上 120R 终端电阻的跳线帽。

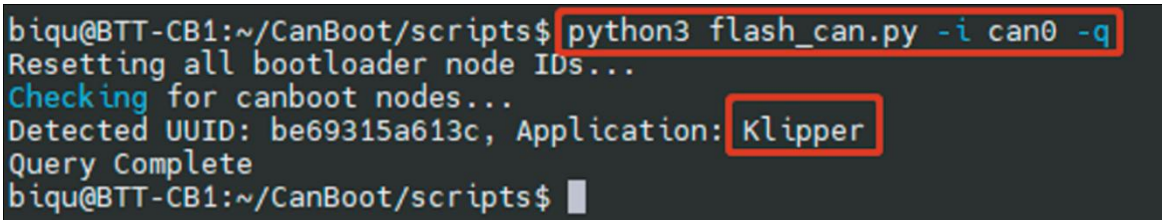
2. 输入
   cd ~/CanBoot/scripts
   然后输入
   python3 flash_can.py -i can0 -q
   查询 canbus ID （需提前接好 CAN 线并通电），如下图已找到设备的 UUID




3. 输入
   python3 flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u be69315a613c
   其中的“be69315a613c”需要替换为实际的 UUID，注意：klipper.bin 需要提前 make 生成出来，并且 CanBoot 的 Application start offset 为 8KiB offset，所以 Klipper 的 menuconfig 中 Bootloader offset 也要为 8KiB bootloader，如下图已经烧录成功。




4. 再次输入
   python3 flash_can.py -i can0 -q
   查询，此时 Application 由之前的 CanBoot 变为 Klipper，代表 Klipper 已经正常运行



## 4.4 通过 DFU 进行固件更新

树莓派或 CB1 通过 DFU 更新

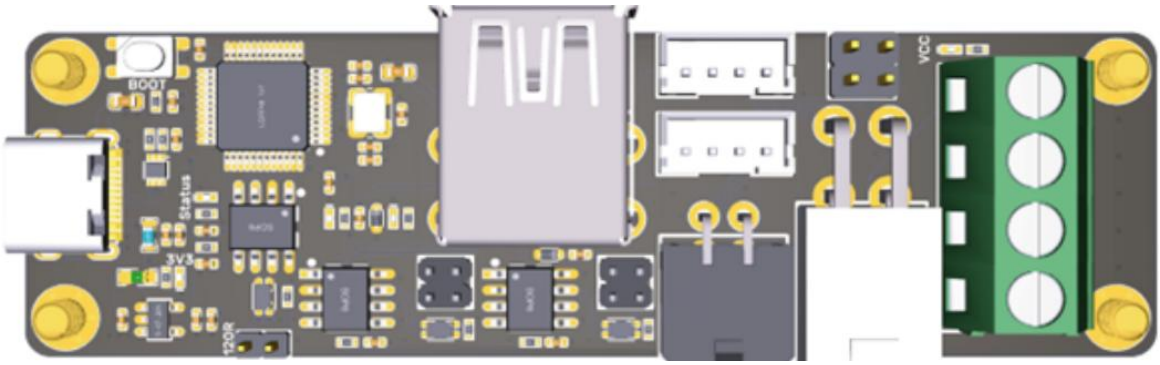
1. 请按住 Boot 按钮，然后使用 Type-C 线连接至树莓派/CB1，此时芯片进入 DFU 模式
2. 在 SSH 终端命令行中输入
   lsusb
   查询 DFU 设备 ID

```
pi@fluiddpi:~ $ lsusb
Bus 001 Device 005:  ID 0483:df11 STHicroolactronics STM Device in DFU Mode
Bus 001 Device 004:  ID 1d50:6061 OpenMoko, Inc. Geschwister Schneider CAN adapter
Bus 001 Device 003:  ID 0424:0c00 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@fluiddpi:~ $
```

3. 输入
   cd klipper
   跳转到 klipper 目录下，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）
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）。
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 按钮进入 DFU 模式，可以直接输入
   make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_4550357128922FC8-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）。
6. 如果使用 CAN bus 通信，烧录完成后，请拔下 Type-C 数据线。

## 4.5 CAN bus 配置

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,新增以下内容
   **allow-hotplug can0**
   **iface can0 can static**
   **bitrate 1000000**
   **up ifconfig $IFACE txqueuelen 1024**
   将 CAN bus 速度设置为 **1M**（必须与固件中设置的速度一致(1000000) CAN bus speed），修改后保存（Ctrl + S）并退出（Ctrl + X），输入
   **sudo reboot**
   重启树莓派。

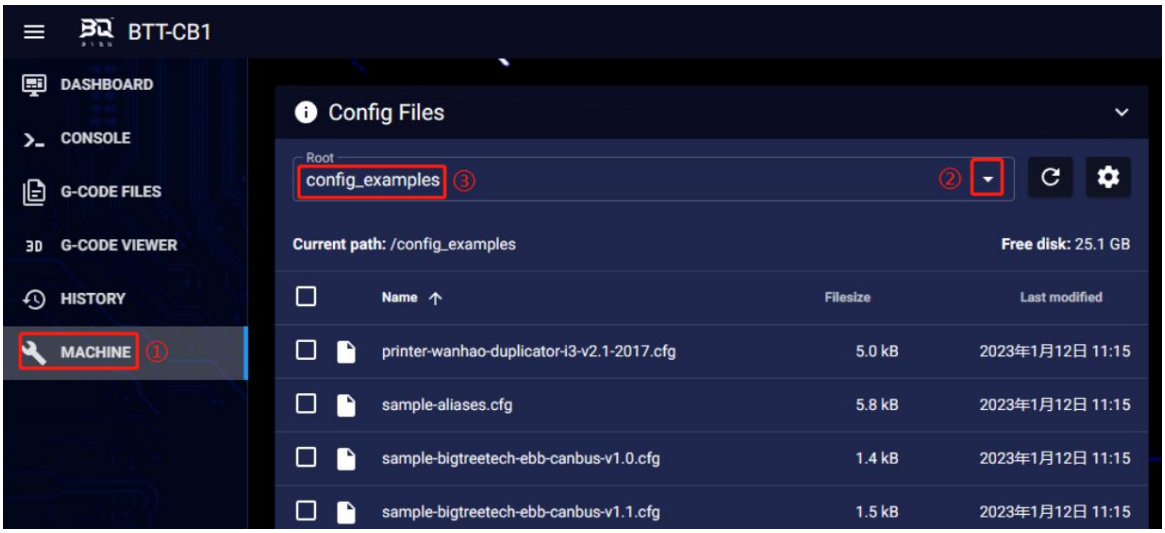
2. CANBus 上的每个设备都会根据 MCU 的 UID 生成一个 canbus_uuid，要查找每个微控制器设备 ID，请确保硬件已通电并正确接线，然后运行：
   **~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0**

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**

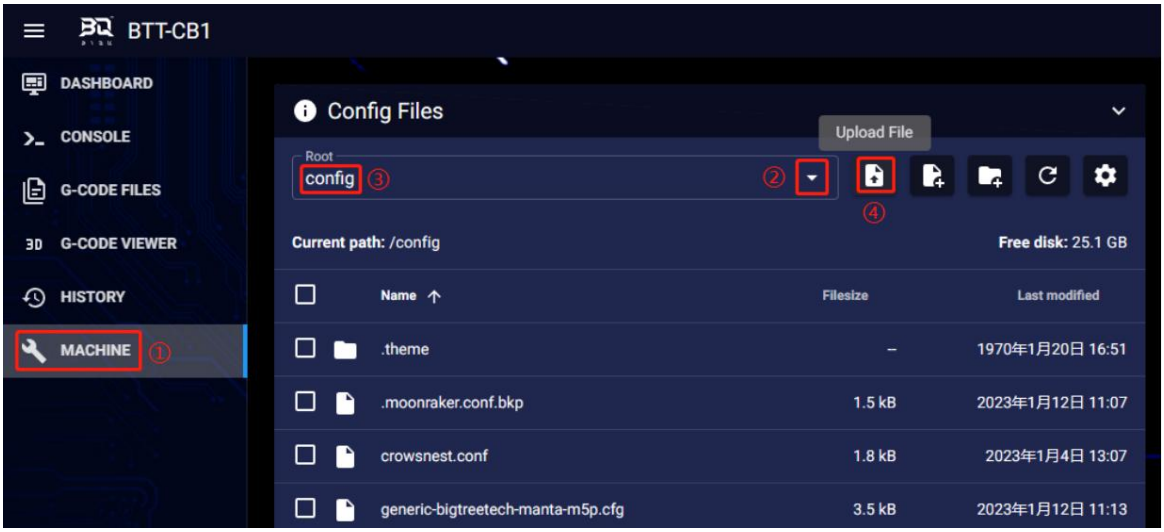
4. 如果 Klipper 已经正常运行并且连接到此设备，那么 canbus_uuid 将不会被上报，此为正常现象。

## 4.6 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名为“**sample-bigtreetech-mmb-canbus.cfg**”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/MMB



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-mmb-canbus.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. 按照下方链接的说明配置模块的具体功能：
https://www.klipper3d.org/Overview.html