

**BIGTREE TECH**

# Eddy V1.0

## 用户手册



修订历史

版本	日期	修改说明
v1.00	2024/4/19	初稿

---

## 目录

一、产品信息.....	4
二、产品优点.....	5
三、产品尺寸.....	6
四、安装指南.....	7
4.1 以 voron2.4 为例 .....	7
4.2 Eddy + MANTA M5P .....	8
4.3 Eddy + MANTA M8P V2.0 .....	9
4.4 Eddy Coil + EBB36 V1.2 .....	9
4.5 Eddy Coil + EBB42 V1.2 .....	10
五、固件.....	11
5.1 重要提示 .....	11
5.2 编译固件 .....	11
5.3 通过 DFU 进行固件更新.....	12
5.4 Klipper .....	14
5.4.1 (USB) 主要配置 .....	14
5.4.2 (coil) 主要配置 .....	15
5.4.3 bed_mesh.....	16
5.5 校准 .....	17

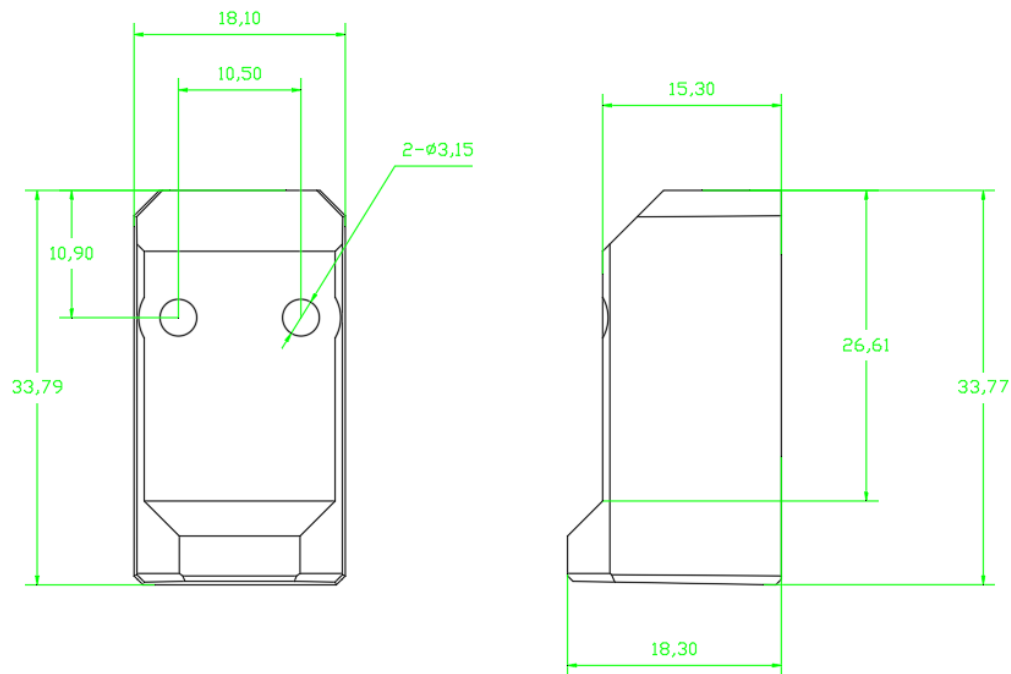
## 一、产品信息

产品名称	Eddy
产品重量	6g
电压	5V
静态电流	30mA
工作电流	30mA
电缆长度	2.5 m（USB 版），15cm（coil 版）
接线	USB: 4pin, 间距 1.5mm  Coil: 4-2.54mm 杜邦母头，一头 ZH1 5mm 4P 连接器
使用温度	≤60℃ 环境温度
标准误差	5μm
适配机型	所有使用 Klipper 的 FDM 打印机

## 二、产品优点

- 整体体积小，重量轻；
- 具备温度补偿功能；
- 高效率调平；
- 应用广泛，兼容性强；
- 精度高，稳定性强；
- 非接触式；

### 三、产品尺寸



尺寸图

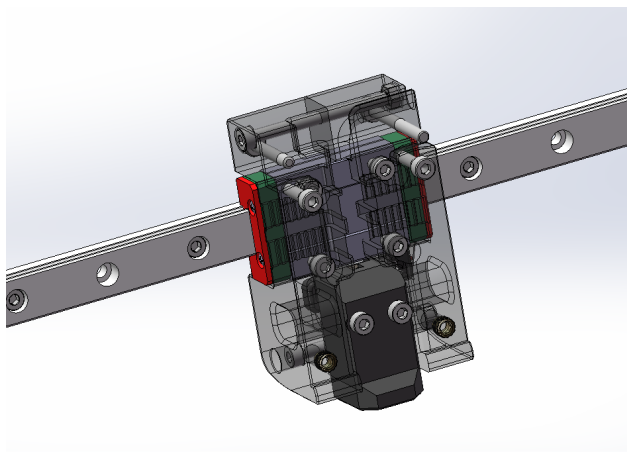
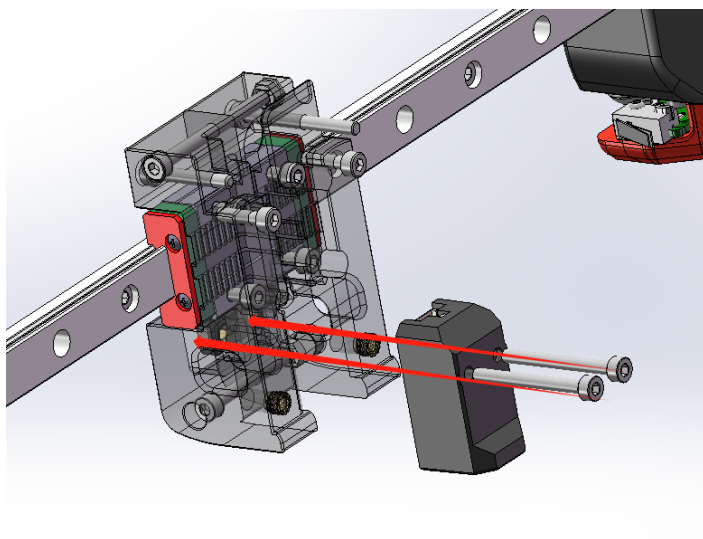
注意：在安装 Eddy 时，底部之于喷嘴的位置需至少在喷嘴上面 1-2 mm。

## 四、安装指南

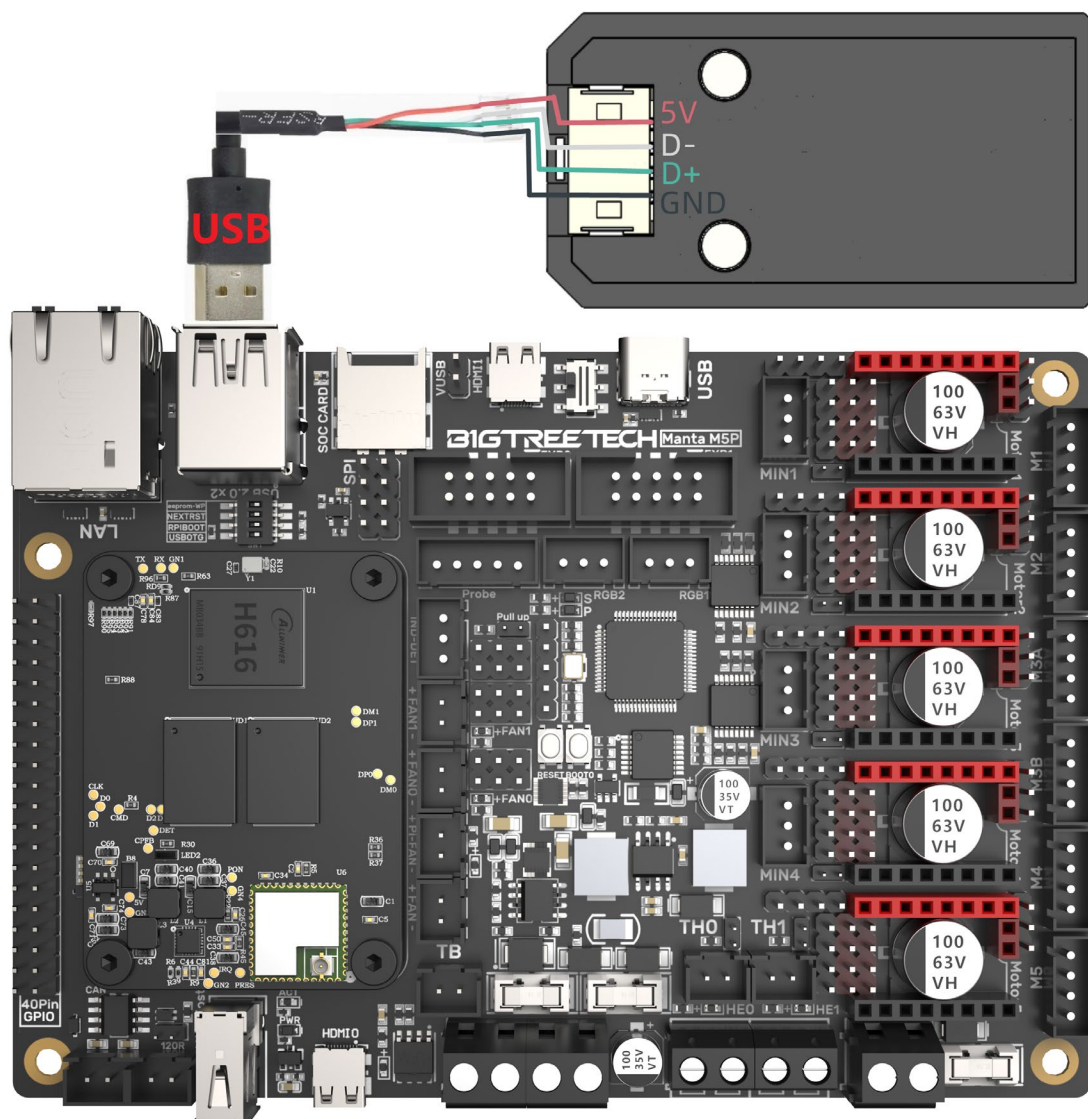
### 4.1 以 voron2.4 为例

安装位置，完全替代原有 PL-08N 安装位

使用两颗 M3\*25 螺丝（包装内附）将模块固定再 X Carriage 打印件上，如视图

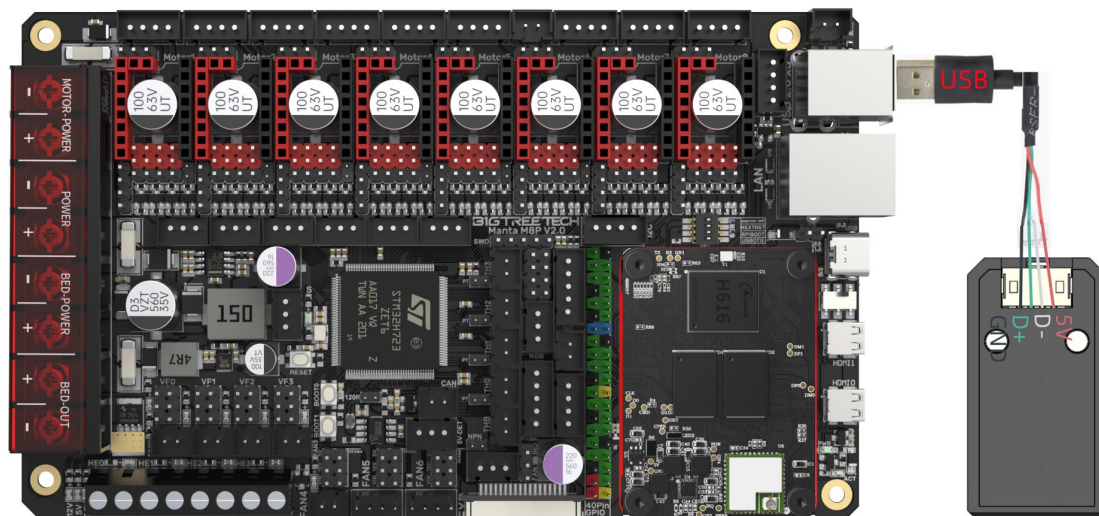


## 4.2 Eddy + MANTA M5P

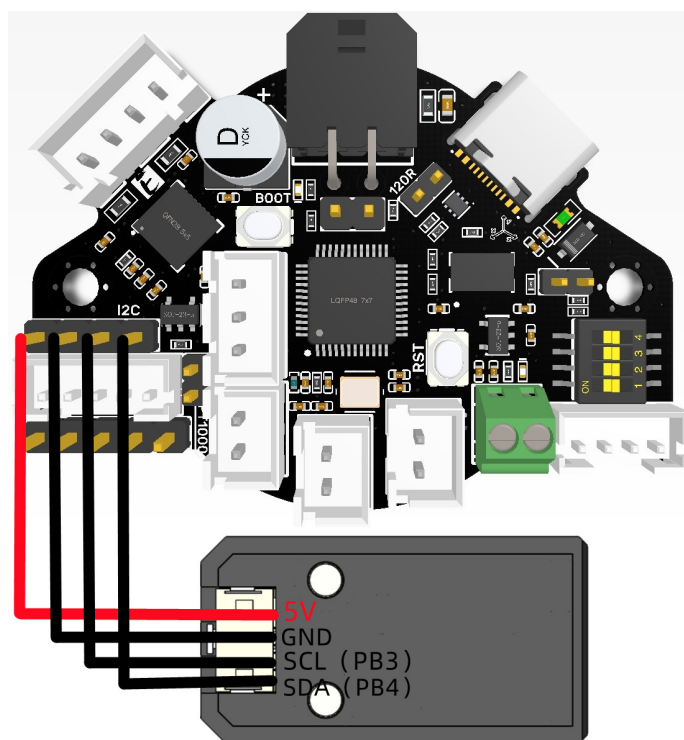




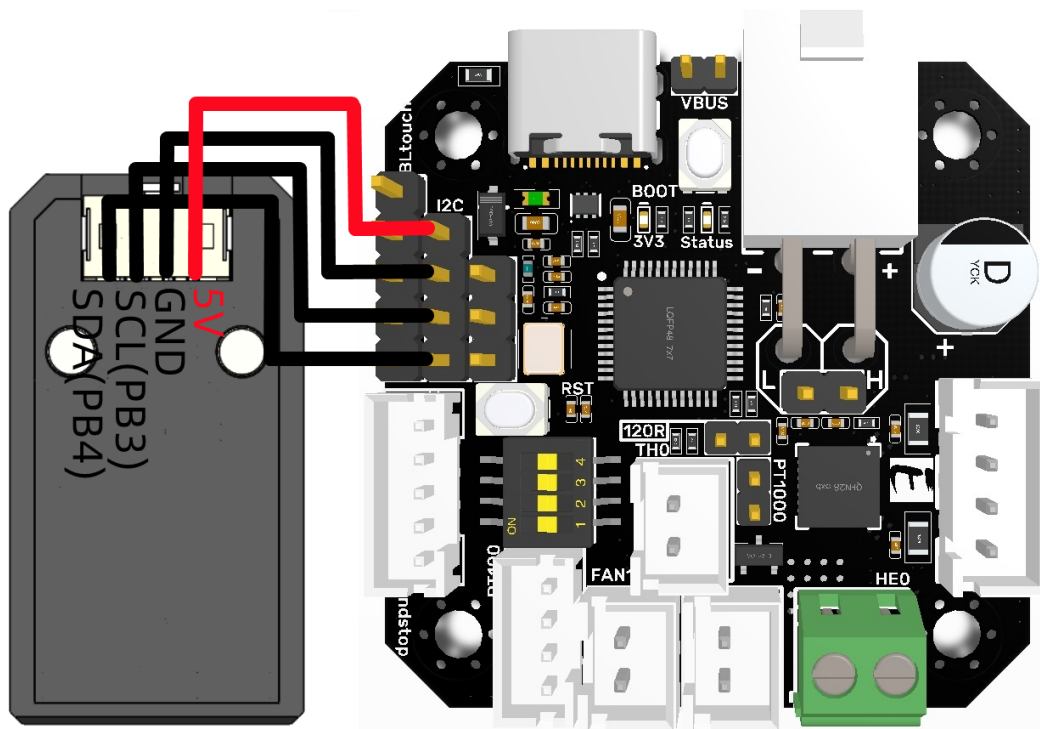
### 4.3 Eddy + MANTA M8P V2.0



### 4.4 Eddy Coil + EBB36 V1.2



## 4.5 Eddy Coil + EBB42 V1.2



## 五、固件

### 5.1 重要提示

Eddy 进行温度补偿时热床温度较高，谨防烫伤

### 5.2 编译固件

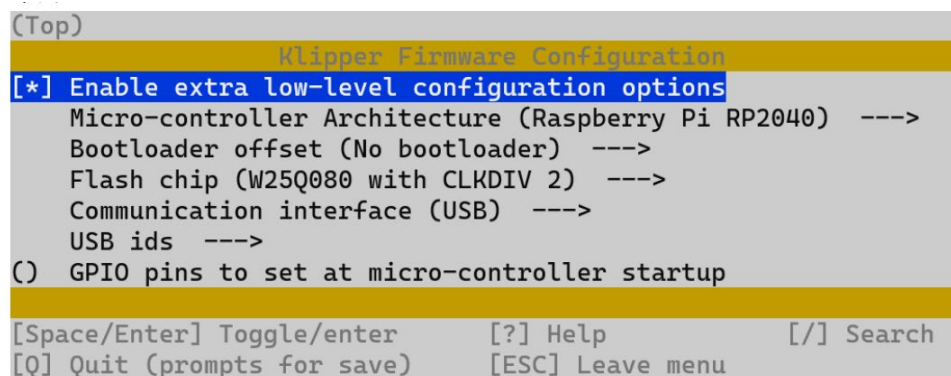
USB 版本需要更新固件，而 coil 需要根据依附使用的模块或主板更新固件。

(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)：

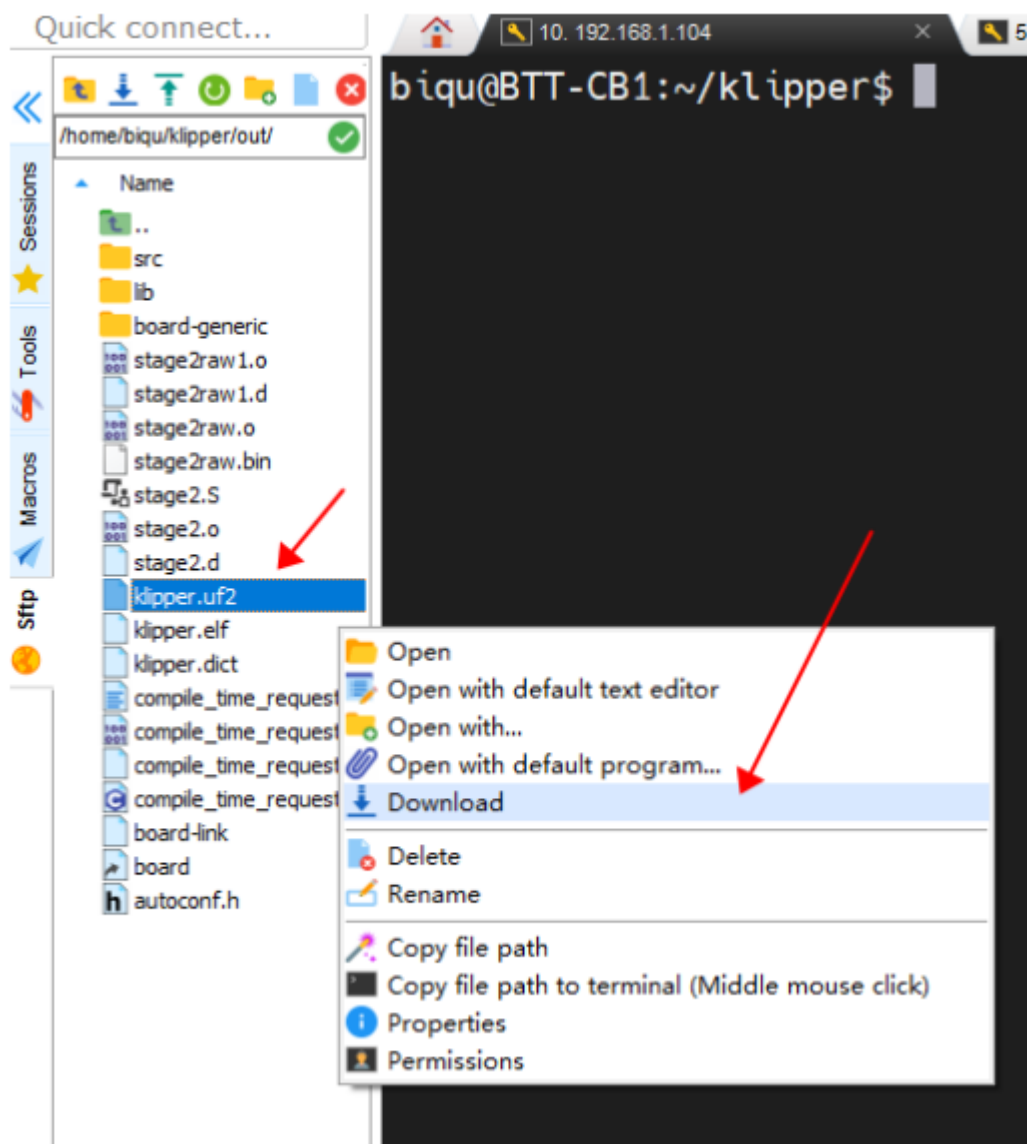


```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040) --->
    Bootloader offset (No bootloader) --->
    Flash chip (W25Q080 with CLKDIV 2) --->
    Communication interface (USB) --->
    USB ids --->
( ) GPIO pins to set at micro-controller startup
[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)  [ESC] Leave menu
```

(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.uf2’固件，在 SSH 软件左侧可以直接下载到电脑中。



### 5.3 通过 DFU 进行固件更新

1. 按住 Boot 按钮，然后接通电源进入 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 2e8a:0003 Raspberry Pi [RP2 Boot]
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@fluidpi:~$
```

3. 输入 `cd klipper`，跳转到 `klipper` 目录下

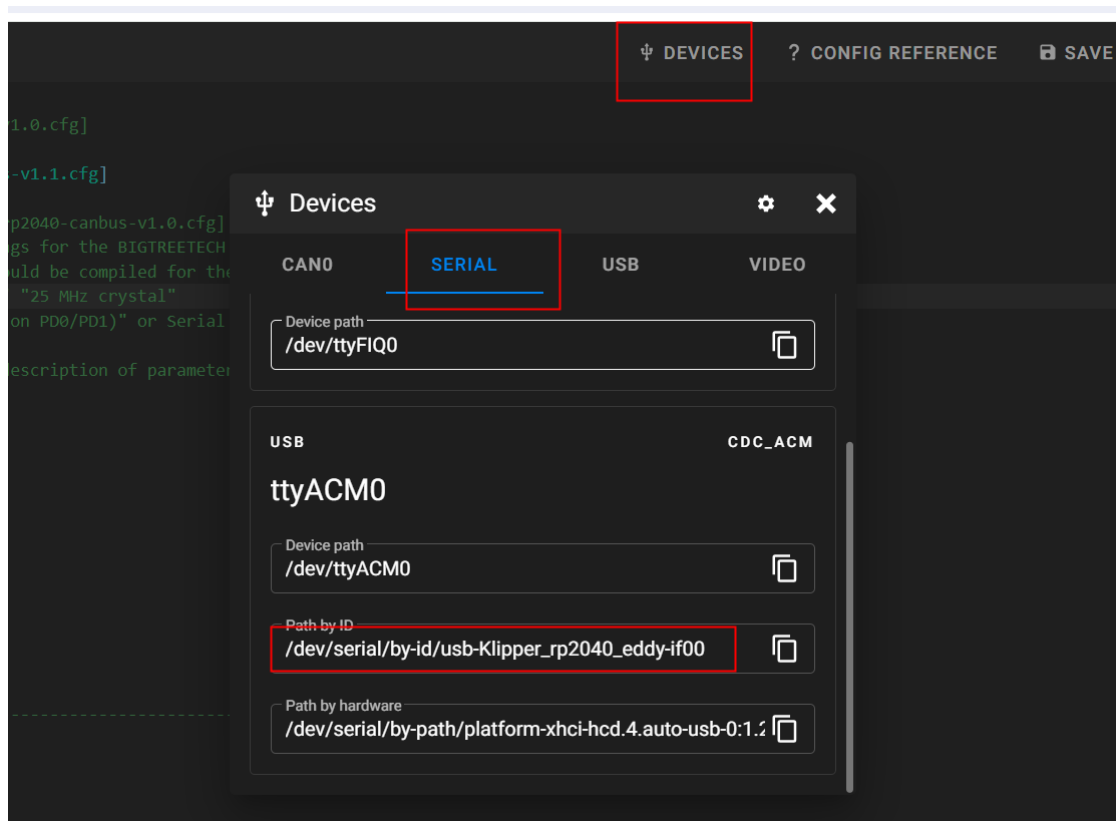
4. 输入 `make flash FLASH_DEVICE=2e8a:0003` 开始烧录固件

（注意：将 `2e8a:0003` 更换为上一步中查询到的实际的设备的 ID）

5. 固件烧录完成后，输入 `ls /dev/serial/by-id/*`，查询 USB 通信的 ID，可以将它更新到 `klipper` 的配置文件 `serial` 中。

上面的步骤完成后，USB 通信基本可以正常使用了。

查询 USB 通信 ID 也可以到 mainsail 里面直接查找



## 5.4 Klipper

### 5.4.1 (USB) 主要配置

1.在 ssh 终端中执行以下命令

```
cd ~/klipper/
```

```
git remote add eddy https://github.com/bigtreotech/klipper
```

```
git fetch eddy
```

```
git checkout eddy/eddy
```

2.在 printer.cfg 中配置 eddy

```
[mcu eddy]
```

```
serial: /dev/serial/by-id/ (根据上面的方法或 mainsail 里面查找到的实际 ID)
```

```
[temperature_sensor btt_eddy_mcu]
```

```
sensor_type: temperature_mcu
```

```
sensor_mcu: eddy
```

```
min_temp: 10
```

```
max_temp: 100
```

```
[probe_eddy_current btt_eddy]
```

```
sensor_type: ldc1612
```

```
z_offset: 1.0 # 不要设置为 0 即可
```

```
i2c_mcu: eddy
```

```
i2c_bus: i2c0f
```

```
x_offset:0 # 根据实际相对于喷嘴的偏移量设置
```

```
y_offset: 20 # 根据实际相对于喷嘴的偏移量设置
```

```
data_rate: 500
```

```
[temperature_probe btt_eddy]
```

sensor\_type: Generic 3950

sensor\_pin: eddy:gpio26

horizontal\_move\_z: 2

[bed\_mesh]

speed: 300

horizontal\_move\_z: 2

mesh\_min: 50, 40

mesh\_max: 220, 200

# probe\_count: 5,5

probe\_count: 9, 9

algorithm: bicubic

[safe\_z\_home]

home\_xy\_position:150,150

speed: 200

z\_hop: 10

z\_hop\_speed: 25

#### 5.4.2 (coil) 主要配置

1.在 ssh 终端中执行以下命令

```
cd ~/klipper/
```

```
git remote add eddy https://github.com/bigtreotech/klipper
```

```
git fetch eddy
```

```
git checkout eddy/eddy
```

2.在 printer.cfg 中配置 eddy

```
[mcu eddy]
```

```
serial: /dev/serial/by-id/usb-Klipper_stm32g0b1xx_3D0047001150425539393020-if0
```

0

`[temperature_sensor btt_eddy_mcu]``sensor_type: temperature_mcu``sensor_mcu: eddy``min_temp: 10``max_temp: 100``[probe_eddy_current btt_eddy]``sensor_type: ldc1612``z_offset: 1.0 # 不要设置为 0 即可``i2c_mcu: eddy``i2c_bus: i2c3_PB3_PB4 #根据实际使用的主板或者扩展模块的 I2C 端口的引脚写入``x_offset: 40 # 根据实际相对于喷嘴的偏移量设置``y_offset: -30 # 根据实际相对于喷嘴的偏移量设置``data_rate: 500``[bed_mesh]``speed: 300``horizontal_move_z: 5``mesh_min: 60, 10``mesh_max: 220, 190``# probe_count: 5,5``probe_count: 9, 9``algorithm: bicubic``[safe_z_home]``home_xy_position:150,150``speed: 200``z_hop: 10``z_hop_speed: 25`

### 5.4.3 bed\_mesh

`speed: 50``#校准过程中非探测移动的速度（毫米/秒）``horizontal_move_z: 5``# 在开始探测操作前,头部被命令移动到的高度(mm)`



**mesh\_min: 10, 10**

# 对于矩形热床，定义网格的最小 X, Y 坐标。此坐标相对于探测头的位置。这将是第一个探测点，最接近原点。矩形热床必须提供此参数。

**mesh\_max: 220, 220**

# 对于矩形热床，定义网格的最大 X, Y 坐标。遵循与 mesh\_min 相同的原则，但这将是离床原点最远的探测点。矩形热床必须提供此参数。

**probe\_count: 5, 5**

# 对于矩形热床，这是一对逗号分隔的整数值 X, Y，定义沿每个轴探测的点数。单个值也有效，此时该值将应用于两个轴。

参考 [https://www.klipper3d.org/Config\\_Reference.html#bed\\_mesh](https://www.klipper3d.org/Config_Reference.html#bed_mesh)

[bed\_mesh] 里的 horizontal\_move\_z 参数，建议设置为 2，使扫描平台的时候，Eddy 尽可能的靠近平台。

## 5.5 校准

1. 上述配置完成后，首先需要校准 Eddy 的驱动电流，将 Eddy 放置在平台上方 20mm 左右的地方。执行 **LDC\_CALIBRATE\_DRIVE\_CURRENT CHIP=btt\_eddy**

自动获取电流，然后 **SAVE\_CONFIG** 保存参数

2. 校准 Eddy 频率与 Z 轴高度的关系

**G28 X Y** 先 home X 和 Y 轴

**G0 X150 Y150 F6000** 将喷头移动到平台中心（注意：进行这个步骤要确保机器没有高度图）

**PROBE\_EDDY\_CURRENT\_CALIBRATE CHIP=btt\_eddy** 开始手动 z 偏移校准，校准完成后如图所示 **SAVE\_CONFIG** 保存参数

```
09:23 SAVE_CONFIG
```

```
09:23 probe_eddy_current: stddev=144.727 in 3998 queries
The SAVE_CONFIG command will update the printer config file
and restart the printer.
```

```
09:22 ACCEPT
```

对于 Voron 要进行一次 QGL（龙门架调平），防止下面进行网格扫描时喷嘴与热床摩擦此时就可以 **home all** 使所有轴归位，然后进行快速的网格扫描，执行以下命令：

**BED\_MESH\_CALIBRATE METHOD=scan SCAN\_MODE=rapid**

## SAVE\_CONFIG 保存参数





### 3. 温度补偿（Eddy coil 无温度补偿）

（1）home all 归位所有的轴

（2）执行 `SET_IDLE_TIMEOUT TIMEOUT=36000`

将机器的 idle timeout 设置长一点，避免我们升温过程的时候 timeout 了

（3）记录下常温下的 BTT eddy 温度。设定一个热床极限温度，并设置常用的工具头温度，等待 BTT eddy 温度稳定后记录这个极限的 BTT eddy 温度。

	Extruder	off	25.2°C	0 °C ▼
	Heater Bed	off	25.1°C	0 °C ▼
	Btt Eddy		36.2°C	
	Btt Eddy Mcu		39.2°C	

（这个极限 btt\_eddy 温度可以作为下面的目标温度。目标温度不能超过极限温度，最好少一两度）

（4）等待回到常温后执行

`PROBE_DRIFT_CALIBRATE PROBE=btt_eddy TARGET=50 STEP=5` 其中 TARGET=50,

TARGET=50 是指目标温度到 50°C，STEP=5 指的是每个节点的温度刻度是 5°C。

注意：热床极限温度很高，谨防烫伤。

例如当前的温度为 30°C，目标温度是 50°C，那么总的温度范围为 50-30=20°C，每个节点的刻度是 5°C，也就是会采样  $20 / 5 = 4$  个节点。

执行以上命令会立刻要求一个手动 z 偏移校准，然后手动加热热床、喷头等待温度升高，等待 BTT eddy 的温度升高，BTT eddy 在下一个节点也就是 35°C 会再要求一个手动 z 偏移校准，然后再在下一个节点也就是 40°C 再要求一个手动 z 偏移校准，以此类推。

注意：如上述的例子，第二个温度节点为 35°C，但是手动 z 偏移校准需要时间，并且此时温度可能仍然在不断的升高，如果校准完成时候的实际温度为 36°C，那么下一个节点的温度会在此基础上增加一个刻度值，也就是 41°C。